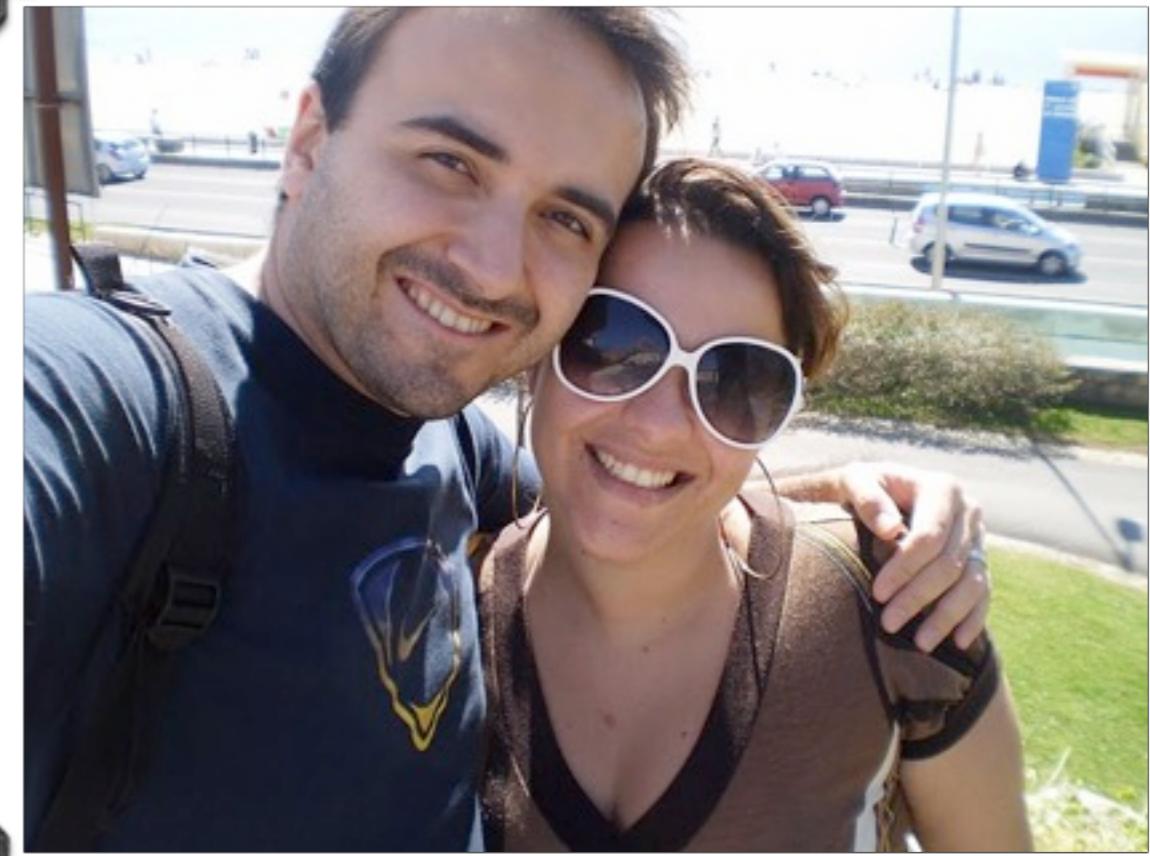


GRAILS





Bruno Gama Catão

Pai em tempo integral

Professor da Facisa

Sistemas de Informação

Jogos Digitais

Coordenador do LTI

Tento levar alguns projetos open source:

github.com/brunogamacatao



GRAILS

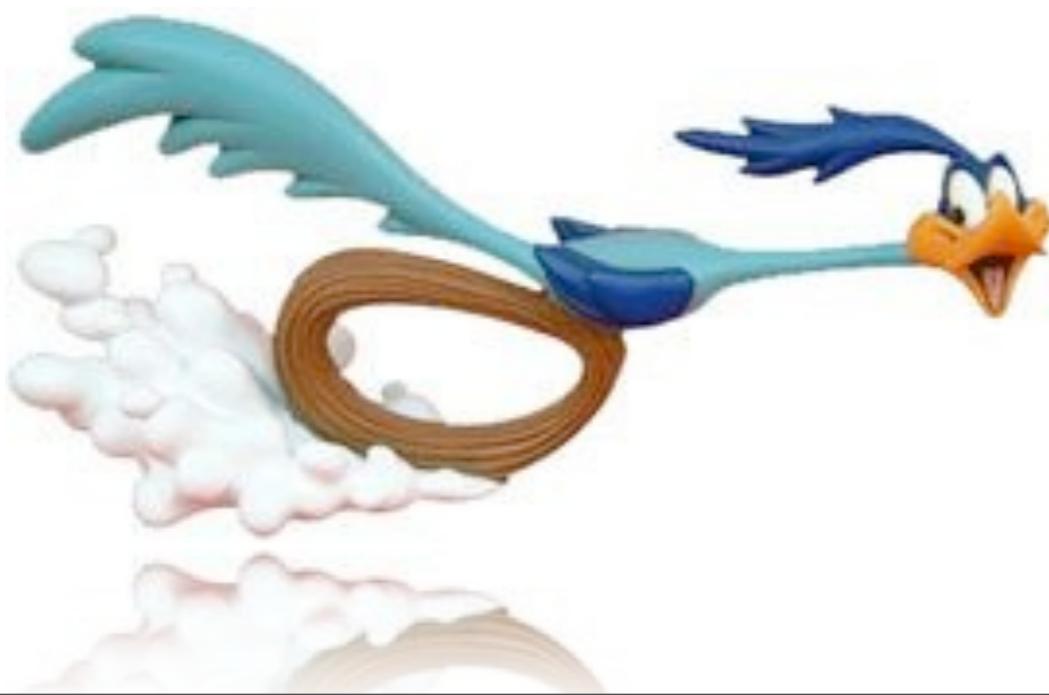
the search is over.



grails.org

O que é ?

- Tecnologia para construção **RÁPIDA** e **FÁCIL** de aplicações para a Internet (web)





GRAILS = GROOVY + RAILS



GROOVY ?

Groovy

Java



Código Java

```
public class AloMundo {  
    private String nome;  
  
    public String getName() {  
        return nome;  
    }  
  
    public void setName(String nome) {  
        this.nome = nome;  
    }  
  
    public String saudacao() {  
        return "Oi " + nome;  
    }  
  
    public static void main(String[] args) {  
        AloMundo alo = new AloMundo();  
        alo.setName("Fulano");  
        System.out.println(alo.saudacao());  
    }  
}
```

Código Groovy

```
public class AloMundo {  
    private String nome;  
  
    public String getName() {  
        return nome;  
    }  
  
    public void setName(String nome) {  
        this.nome = nome;  
    }  
  
    public String saudacao() {  
        return "Oi " + nome;  
    }  
  
    public static void main(String[] args) {  
        AloMundo alo = new AloMundo();  
        alo.setName("Fulano");  
        System.out.println(alo.saudacao());  
    }  
}
```

É a mesma coisa ?!

... então, qual a vantagem disso ?



Código Groovy

Pra quê ; ?

Todas as classes e métodos são públicos até que prove o contrário !

```
class AloMundo {  
    private String nome  
  
    String getName() {  
        return nome  
    }  
  
    void setName(String nome) {  
        this.nome = nome  
    }  
  
    String saudacao() {  
        return "Oi " + nome  
    }  
  
    static void main(String[] args) {  
        AloMundo alo = new AloMundo()  
        alo.setName("Fulano")  
        System.out.println(alo.saudacao())  
    }  
}
```

Código Groovy

```
class AloMundo {  
    String nome  
  
    String saudacao() {  
        return "Oi " + nome  
    }  
  
    static void main(String[] args) {  
        AloMundo alo = new AloMundo()  
        alo.setNome("Fulano")  
        System.out.println(alo.saudacao())  
    }  
}
```

Atributos já tem
get e set
automático
(propriedades)

Código Groovy

Linguagens
fortemente
tipadas estão
fora de moda !

```
class AloMundo {  
    def nome  
  
    def saudacao() {  
        return "Oi " + nome  
    }  
  
    static main(args) {  
        def alo = new AloMundo()  
        alo.setNome("Fulano")  
        println(alo.saudacao())  
    }  
}
```

Ao invés de
System.out.println use
apenas *println* ;)

Código Groovy

Versão final !

Olha que forma
legal de construir
Strings !

```
class AloMundo {  
    def nome  
  
    def saudacao() {  
        return "Oi ${nome}"  
    }  
  
    static main(args) {  
        def alo = new AloMundo()  
        alo.nome = "Fulano"  
        println alo.saudacao()  
    }  
}
```

Não precisa
chamar get/set



- Resultado:
 - 21 linhas Java contra 13 linhas Groovy
 - Limpamos o lixo do código !

Java - Lendo um Arquivo

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

public class LeArquivo {
    public static void main(String[] args) {
        try {
            BufferedReader arquivo = new BufferedReader(new FileReader("/etc/passwd"));
            String line = null;
            while ((line = arquivo.readLine()) != null) {
                System.out.println(line);
            }
            arquivo.close();
        } catch (FileNotFoundException ex) {
            ex.printStackTrace();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}
```

Groovy - Lendo um Arquivo

Tá quase bom ...

```
class LeArquivo {  
    static main(args) {  
        new File("/etc/passwd").eachLine({line ->  
            println(line)  
        })  
    }  
}
```

Podemos passar um
bloco de código, closure,
como parâmetro !

Groovy - Lendo um Arquivo

Pra quê uma classe, se ela só tem
um método main ?

Versão final !

```
new File("/etc/passwd").eachLine {line -> println line}
```

Pense bem ... nem todos os
parêntesis são necessários !

Entendendo Melhor Groovy

Variáveis

```
def nomeDaVariavel = valorDaVariavel
```

Ex:

```
def idade = 28
```

```
def nome = "Bruno"
```

Strings 1/2

```
def nome = 'José da Silva'  
def cidade = "Campina Grande"  
def texto = '''Oi,  
como vai você ?  
Tudo bem ?'''  
def expr = /Contra-barras são bem vindas !/
```



Strings 2/2

- GStrings (Groovy Strings) podem ser construídos de duas formas:
 - Concatenação: +
 - Expressões: \${expressão}

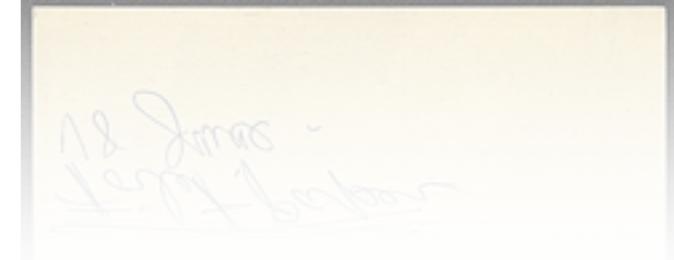
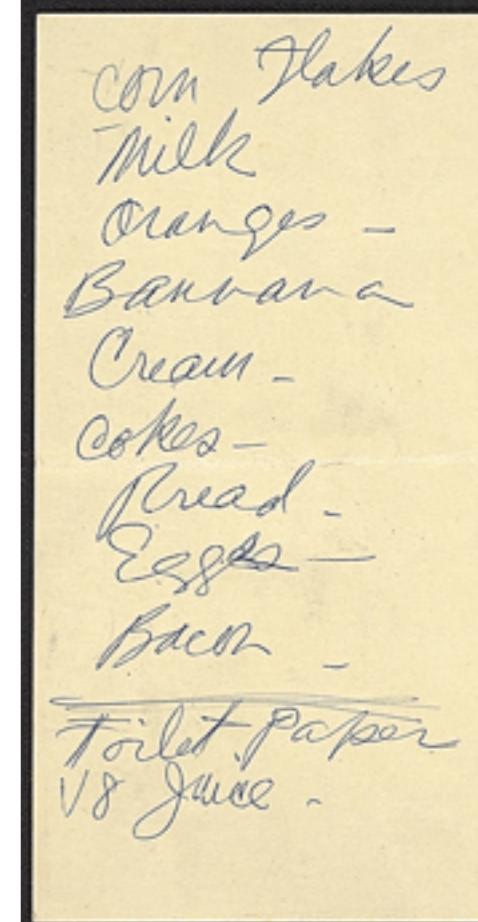
```
def primeiroNome = 'José'  
def ultimoNome = 'Silva'  
def nomeCompleto = primeiroNome + ' ' + ultimoNome  
  
//Expressões devem vir em Strings com aspas duplas !  
def nomeCompleto = "${primeiroNome} ${ultimoNome}"
```

Listas

```
def listaVazia = []  
def frutas = ['Banana', 'Maçã']  
frutas << 'Abacate'  
  
print listaVazia.size() // 0  
print frutas[2] // Maçã
```

// O trecho abaixo dá no mesmo ...

```
def nomes = new ArrayList()  
nomes.add('Bruno')  
  
// Porém não é muito Groovy
```



Maps

```
def mapaVazio = [:]
```

```
def preco = [cocaCola: 2.00, coxinha: 2.50]
```

```
print preco[pipoca] // 0.50
```

```
preco[guarana: 1.50]
```



Métodos

```
def saudacao(nome) {  
    print "Oi ${nome}, como vai você ?"  
}  
  
def soma(a, b) {  
    a + b // Não precisa do return  
}
```

Closures 1/3

- É um novo tipo de dado;
- Representa um trecho de código;
- Pode ser passado como argumento em métodos e armazenado em variáveis.



Closures 2/3

```
def saudacao = {print "Oi, como vai?"}
```

```
saudacao() // "Oi como vai?"
```

// Parâmetro implícito: it

```
def saudacao2 = {  
    print "Oi ${it} como vai?"
```

```
}
```

```
saudacao("Bruno") // "Oi Bruno como vai?"
```

```
def soma = { a, b ->
```

```
    a + b // Lembrem-se, não precisa do return
```

```
}
```

```
soma(2, 3) // 5
```

Closures 3/3

```
def executaOperacao(a, b, operacao) {  
    operacao(a, b)  
}  
  
def multiplicacao = {a, b -> a * b}  
def subtracao = {a, b => a - b}  
  
executaOperacao(5, 2, multiplicacao) // 10  
executaOperacao(5, 2, subtracao) // 3
```

Novos Operadores

- Spread: *.
- Elvis: ?:
- Safe navigation: ?.
- Contém: in
- Conversão: as
- Identidade: is





E o Grails ?

TODO o Código de uma Aplicação Web

Contato.groovy

```
class Contato {  
    String nome  
    String telefone  
}
```

Este código gera as tabelas do banco de dados, consultas, e páginas em 13 idiomas !

ContatoController.groovy

```
class ContatoController {  
    def scaffold = Contato  
}
```

Isso é bom ?

$1+1=3$



Vantagens

- Mais fácil de manter
- Mais fácil de testar
- Resultados rápidos
- Aumenta a diversão





E a infra-estrutura ?



Tudo é Java

- Você só está digitando menos !
- Tudo vai para um arquivo WAR
- Base sólida:
 - JPA (Hibernate)
 - Spring
 - Sitemesh ...





ALGUNS CONCEITOS



- Convenção ao invés de codificação
- Scaffolding (andaime)
- Templates
- Boas ferramentas



MÃO NA MASSA !!!

