

# Intelie Pipes

Uma Linguagem para Processamento Distribuído de Eventos Complexos

Juan Lopes  
13 de novembro de 2014



# Quem sou eu?

Não que credenciais importem muito, mas esse slide já é tradicional

- Pai do Miguel
- Bacharel em Ciência da Computação (UERJ)
- Mestrando em Ciências Computacionais (UERJ)
- Viciado em algoritmos e competições de programação
- Programador na Intelie
- {github.com,twitter.com}/juanplopes

# Quem sou eu?

Não que credenciais importem muito, mas esse slide já é tradicional

- Pai do Miguel
- Bacharel em Ciência da Computação (UERJ)
- Mestrando em Ciências Computacionais (UERJ)
- Viciado em algoritmos e competições de programação
- Programador na Intelie
- {github.com,twitter.com}/juanplopes
- **Palestrante ~~reserva~~ surpresa!**

# QConSP 2013

## Estruturas de dados probabilísticas



The screenshot shows a presentation slide from InfoQ. At the top, there's a navigation bar with links for Desenvolvimento, Arquitetura e Design, Processos e Práticas, Operações e Infraestrutura, and Arquitetura Corporativa. A banner for QCon Rio 2014 is visible on the right. The main content area has a heading 'Analisando e reduzindo grandes fluxos de dados em tempo real' and a sub-heading 'Agregações distribuídas'. It includes a video player showing a speaker, Juan Lopes, and some mathematical formulas for calculating moments of distributed aggregates.

Analizando e reduzindo grandes fluxos de dados em tempo real

Agregações distribuídas

$\bar{x}_n = \bar{x}_{n-1} + \frac{x_n - \bar{x}_{n-1}}{n}$

$M_{2,n} = \sum_{i=1}^n (x_i - \bar{x}_n)^2$

$M_{2,n} = M_{2,n-1} + (x_n - \bar{x}_{n-1})(x_n - \bar{x}_n)$

$n = \frac{M_{2,n}}{\bar{x}_n}$

$\bar{x}_N = \bar{x}_A + \frac{n_B}{n_X} \bar{x}_B$

$M_{2,N} = M_{2,A} + M_{2,B} + \frac{n_A n_B}{n_X}$

# QConSP 2013

## Estruturas de dados probabilísticas



The screenshot shows a presentation slide from InfoQ. The title is "Analisando o fluxo de dados em tempo real" by Juan Lopes, dated 25 Out 2013. The slide features a video thumbnail of Juan Lopes speaking, a "Slides" button, and a "Video" button. Below the video thumbnail, there is some mathematical notation related to data flow analysis:

$$Z_n = Z_{n-1} + \frac{r_n - \bar{x}_{n-1}}{\sigma_n}$$
$$M_{xx} = \sum_{n=1}^N (Z_n - \bar{Z}_x)^2$$
$$M_{zx} = M_{zx, n-1} + (\bar{x}_n - \bar{x}_{n-1})(\bar{Z}_x - \bar{Z}_{n-1})$$
$$\bar{x} = \frac{M_{zx}}{N}$$
$$\bar{Z}_x = \bar{Z}_{x-1} + \frac{r_n - \bar{x}_{x-1}}{\sigma_x}$$
$$M_{zz} = M_{zz, n-1} + \frac{r_n - \bar{x}_{n-1}}{\sigma_z^2}$$

NOTA: QCon Rio 2014 - Primeira Edição, 23-25 Setembro. Participe!

Agile Java Mobile HTML 5 JavaScript Cloud Desenvolvimento Arquitetura e Design Processos e Práticas Operações e Infraestrutura Arquitetura Corporativa

En | 中文 | 日本 | Fr | Brasil

Text to pesquisar Login

QCon  
Rio de Janeiro 2014  
23 a 25 de setembro

juanlopes.net/qconsp2013

infoq.com/br/presentations/analisando-fluxo-dados-tempo-real

# QConSP 2013

## Estruturas de dados probabilísticas



Desenvolvendo conhecimento e inovação em desenvolvimento de software corporativo.

POQ | Brasil Desenvolvimento Arquitetura e Design Processos e Práticas Operações e Infraestrutura Arquitetura Corporativa

Java Mobile HTML 5 JavaScript Cloud Computing Técnicas de Testes

NOTA: QCon Rio 2014 - Primeira Edição, 23-25 Setembro. Participe!

Slides

anlopes.net/qconsp2013

Vídeo

Set membership

# Bloom Filters

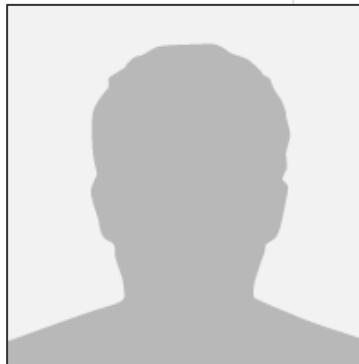
Burton H. Bloom (1970)

Download Slides 47:24

$$Z_0 = E_{Z_0} + \frac{E_{Z_0^2} - E_{Z_0}^2}{n}$$
$$M_{Z_0} = \sum_{i=1}^n (1 - Z_i)^2$$
$$M_{Z_0} = M_{Z_0, 0} + (1 - Z_0)(1 - Z_1) \dots (1 - Z_n)$$
$$\alpha = \frac{M_{Z_0}}{n}$$
$$Z_0 = E_{Z_0} + \frac{S_0}{n}$$
$$M_{Z_0} = M_{Z_0, 0} + M_{Z_0, 1} + \frac{2 - S_0 S_1}{n}$$

# QConSP 2013

## Estruturas de dados probabilísticas



Set membership

## Bloom Filters

Burton H. Bloom (1970)



Set membership

$P(\text{false positive}) = \frac{n}{m} \left(1 - e^{-\frac{n}{m}}\right)^k$

Burton H. Bloom (1970)

Multiset summarization

$P(\text{false positive}) = \frac{n}{m} \left(1 - e^{-\frac{n}{m}}\right)^k$

$M_{1,2} = M_{1,1} + M_{1,2} + M_{2,1} - M_{1,1}M_{1,2}$

$M_{1,2} = M_{1,1} + M_{1,2} + \frac{2 - n_1n_2}{m_1m_2}$

Graham Cormode, et al. (2003)

# QConSP 2013

## Estruturas de dados probabilísticas



Set membership

## Bloom Filters

Burton H. Bloom (1970)



Multiset summarization

## Count-Min

Graham Cormode, et al. (2003)



Set cardinality

## HyperLogLog

Philippe Flajolet, et al. (2007)

## Links

Não precisa correr para anotar, esse slide irá voltar

- Slides  
[juanlopes.net/weop2014](http://juanlopes.net/weop2014)
- Site e documentação  
[pipes.intelie.com](http://pipes.intelie.com)
- Tutorial interativo  
[pipes.intelie.com/tutorial](http://pipes.intelie.com/tutorial)

# Agenda

Porque ter uma agenda sempre dá uma boa impressão

1 PROCESSAMENTO DE EVENTOS 101

WHAT?

2 POR QUE MAIS UMA LINGUAGEM DE CEP?

WHY?

3 DETALHES DA LINGUAGEM

HOW?

4 DEMOS

WTF?

# Agenda

Porque ter uma agenda sempre dá uma boa impressão

1 PROCESSAMENTO DE EVENTOS 101

WHAT?

2 POR QUE MAIS UMA LINGUAGEM DE CEP?

WHY?

3 DETALHES DA LINGUAGEM

HOW?

4 DEMOS

WTF?

An event is a significant **change of state** at a particular point in **time**.



# Eventos

Onde?

- Monitoração de infraestrutura
- Análise de informações de negócio
- Algorithmic trading
- Bioinformática
- Controle de tráfego urbano
- Orquestração de software embarcado
- etc.

# Eventos

Onde?

- Monitoração de infraestrutura
- Análise de informações de negócio
- Algorithmic trading
- Bioinformática
- Controle de tráfego urbano
- Orquestração de software embarcado
- etc.

# Eventos

Onde?

- Monitoração de infraestrutura
- Análise de informações de negócio

**Não há ferramenta universal.**

Considerar: Latência × Flexibilidade × Desempenho

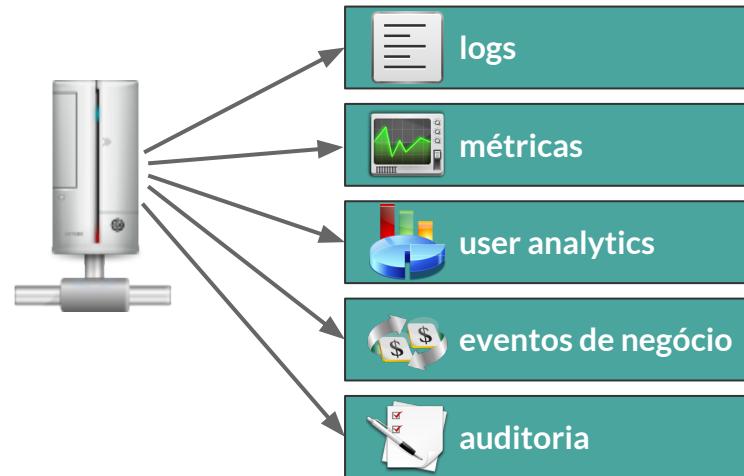
# Eventos

Sua infra está repleta deles



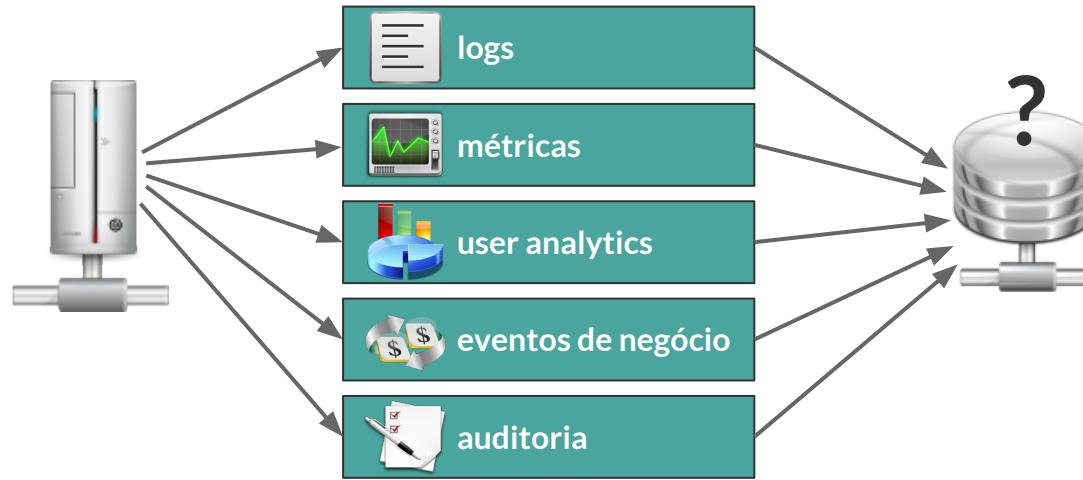
# Eventos

Sua infra está repleta deles



# Eventos

Sua infra está repleta deles



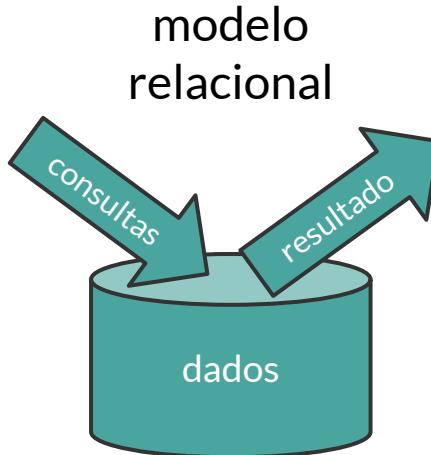
# Por que não SQL?

~~Por que não!~~

- Consultas contínuas
- Primitivas temporais
- Detecção de padrões
- Janelas de eventos
- Controle do output
- Tudo em memória

# Database Invertido

O índice fica nas consultas



foco no passado  
**consultas sem estado**  
dados são o estado

modelo baseado  
em eventos



foco no presente  
**consultas com estado**  
dados sem estado

# Ferramentas

Tem muita gente fazendo isso





```
declare Sale
    @role( event )
end

declare window Ticks
    Sale() over window:length(5)
        from entry-point MyEntryPoint
end

rule "More than 2 sale sucess in 5 events"
when
    Number($cnt : intValue,intValue > 2) from accumulate(
        Sale (saleHappened == "Y") from window Ticks, count(1)
    )
then
    System.out.println(
        "A sale has happened over " + $cnt +" events" );
end
```



```
var forward = inputStream.AlterEventStartTime(
    s => s.StartTime.AddSeconds(1));

var query = from evt in inputStream
            from prev in forward
            where prev.Value < threshold &&
                  evt.Value > threshold
            select new
            {
                Time = evt.Time,
                Low = prev.Value,
                High = evt.Value
            };
```



Complex Event Processing

```
SELECT
    count(*) as fails,
    timestamp,
    local,
    description,
FROM
    HttpMonitor(type = "error" OR
                description = "Timeout::Error")
        .std:groupwin(description)
        .win:time(15 minutes)
```

```
SELECT T.firstc, T.lastc, T.Ac1, T.Bc1, T.avgCc1, T.Dc1
FROM
    S0
MATCH_RECOGNIZE (
    MEASURES
        first(C.c2) as firstc,
        last(C.c2) as lastc,
        avg(C.c1) as avgCc1,
        A.c1 as Ac1,
        B.c1 as Bc1,
        D.c1 as Dc1
    PATTERN(A B C* D)
    DEFINE
        A as A.c1 = 30,
        B as B.c2 = 10.0,
        C as C.c1 = 7,
        D as D.c1 = 40
    ) as T
```



```
SELECT STREAM
    "SuspectLoginFailures"."accountNumber",
    "loginFailureCount",
    "transactionType",
    "amount"
FROM "SuspectLoginFailures" OVER "lastFew"
JOIN "Transactions" OVER "lastFew"
    ON "SuspectLoginFailures"."accountNumber" =
        "Transactions"."accountNumber"
WHERE ("transactionType" = 'isDebit')
WINDOW "lastFew" AS (RANGE INTERVAL '1' MINUTE
PRECEDING);
```

```
CREATE OUTPUT STREAM TickStats AS
    SELECT openval() AS StartOfTimeSlice,
    avg(NumberTicks) AS AvgTicksPerSecond,
    stdev(NumberTicks) AS StdevTicksPerSecond,
    lastval(NumberTicks) AS LastTicksPerSecond,
    FeedName
    FROM TicksPerSecond [
        SIZE 20 ADVANCE 1 ON StartOfTimeSlice
        PARTITION BY FeedName
    ]
    GROUP BY FeedName;
```



## Nomenclatura

Um pouco de bikeshedding não faz mal a ninguém

- ECA (Event-Condition-Action)
- ESP (Event Stream Processing)
- CEP (Complex Event Processing)

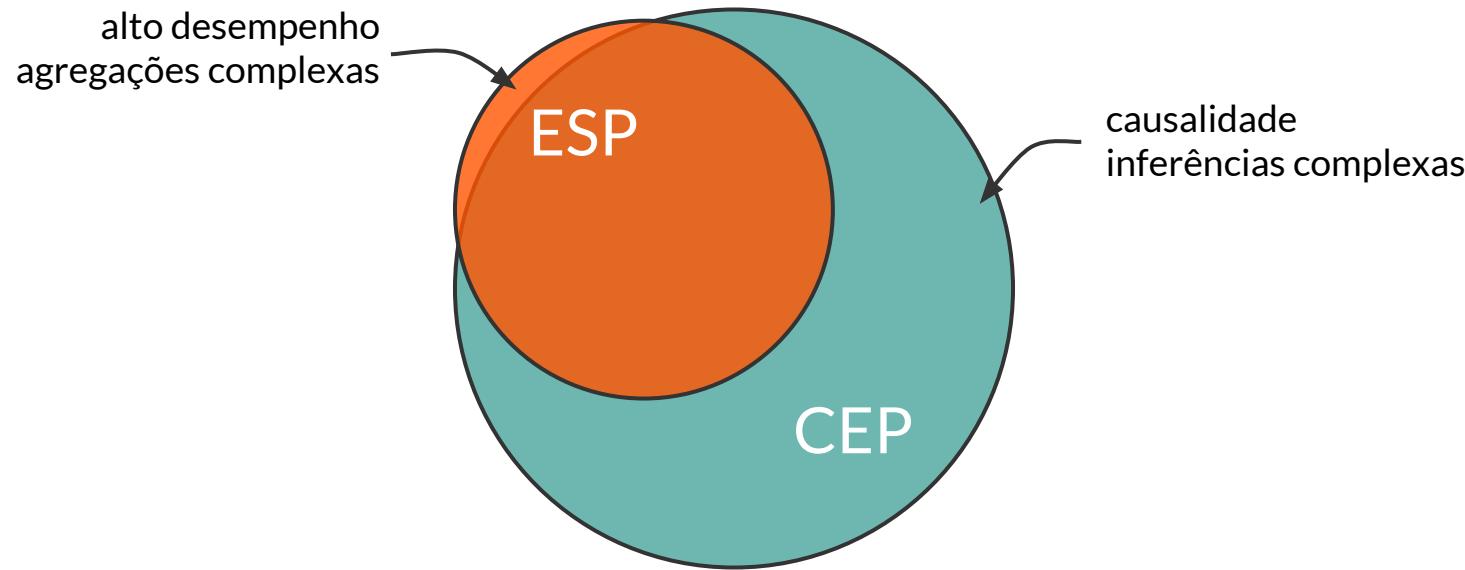
## Nomenclatura

Um pouco de bikeshedding não faz mal a ninguém

- ECA (Event-Condition-Action)
- ESP (Event Stream Processing)
- CEP (Complex Event Processing)

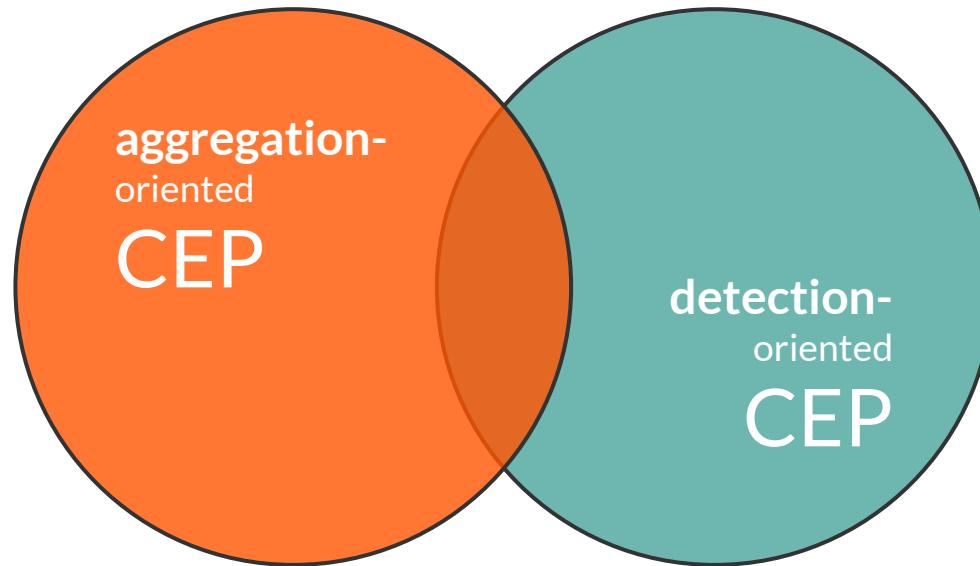
# Nomenclatura

Para falar a verdade, não há tanto consenso



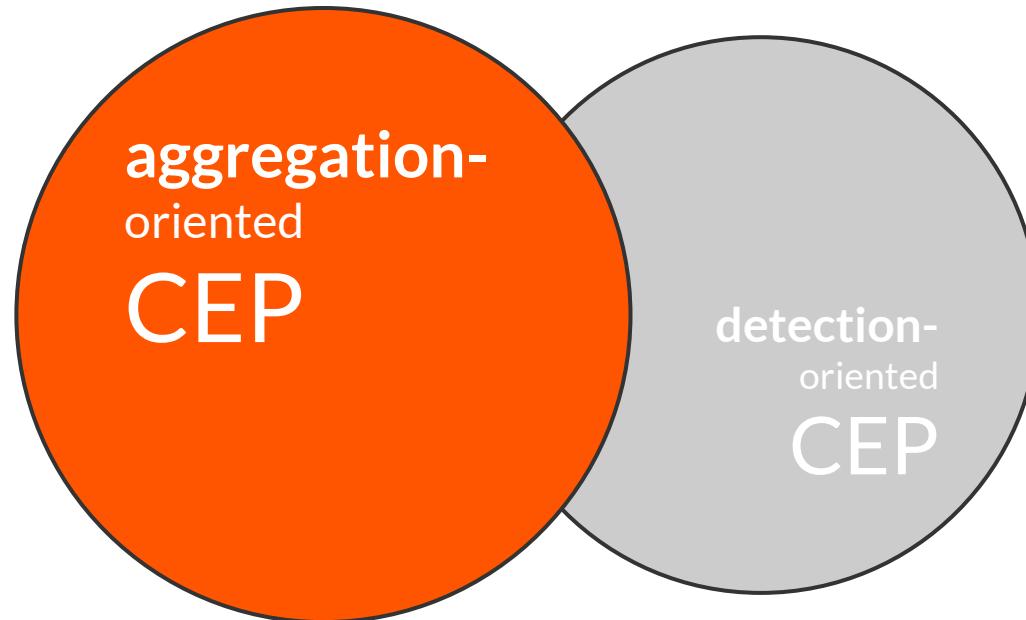
# Nomenclatura

Para falar a verdade, não há tanto consenso



# Nomenclatura

Para falar a verdade, não há tanto consenso



# Aggregações

No passado é simples

```
select count(*) from stream
```

EPL

# Aggregações

No passado é simples, mas e em tempo real?

```
select count(*) from stream
```

EPL

```
public class AggregatorCount implements AggregationMethod
{
    protected long numDataPoints;

    public void clear() {
        numDataPoints = 0;
    }

    public void enter(Object object) {
        numDataPoints++;
    }

    public void leave(Object object) {
        numDataPoints--;
    }

    public Object getValue() {
        return numDataPoints;
    }
}
```

JAVA

# Aggregações

No passado é simples, mas e em tempo real?

```
select avg(field) from stream
```

EPL

```
public class AggregatorAvg implements AggregationMethod
{
    protected double sum;
    protected long numDataPoints;

    public void clear()
    {
        sum = 0;
        numDataPoints = 0;
    }

    public void enter(Object object)
    {
        numDataPoints++;
        sum += ((Number) object).doubleValue();
    }
}
```

JAVA

```
public void leave(Object object)
{
    numDataPoints--;
    sum -= ((Number) object).doubleValue();
}

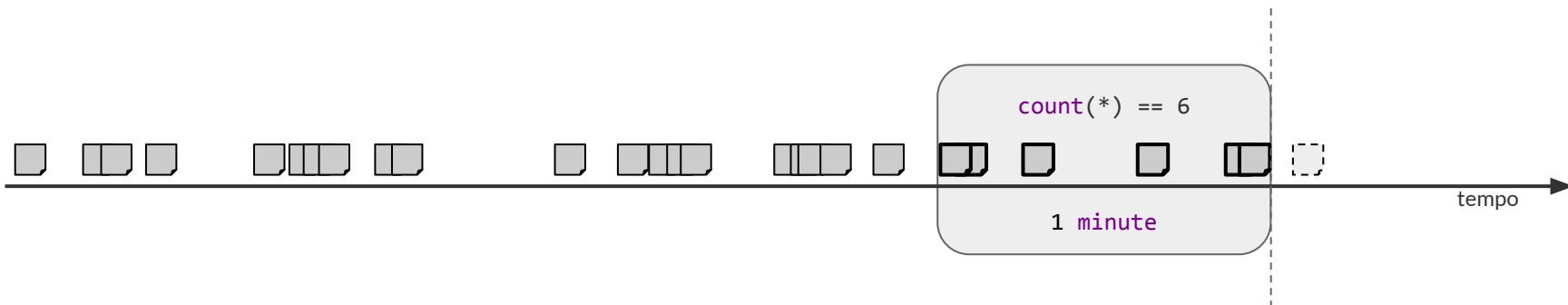
public Object getValue()
{
    if (numDataPoints == 0)
        return null;
    return sum / numDataPoints;
}
```

# Enter, leave?

A lógica em agregações sobre eventos é um pouco diferente

```
select count(*)  
from stream.win:time(1 minute)
```

EPL

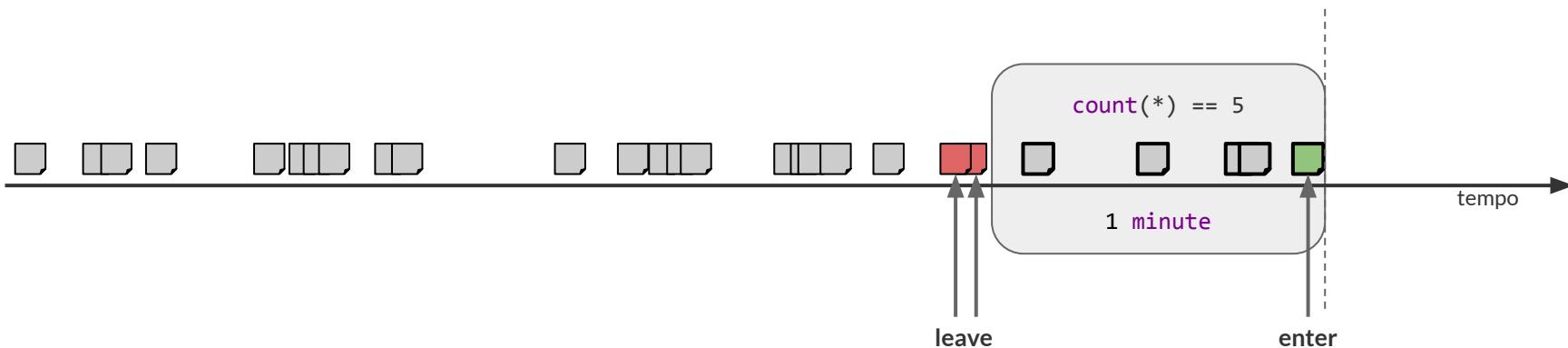


# Enter, leave?

A lógica em agregações sobre eventos é um pouco diferente

```
select count(*)  
from stream.win:time(1 minute)
```

EPL



## Janelas de eventos

Definem o conjunto de eventos sobre os quais a agregação irá executar

- Agregações **não precisam** recalcular novos eventos
- Algoritmos online são essenciais
- Estabilidade numérica é importante; exemplo ruim:

```
public class AggregatorAvg implements AggregationMethod
{
    //...
    public Object getValue()
    {
        if (numDataPoints == 0)
            return null;
        return sum / numDataPoints;
    }
    //...
}
```

JAVA

# Janelas de eventos

Definem o conjunto de eventos sobre os quais a agregação irá executar

- Agregações não precisam recalcular novos eventos
- Algoritmos online são essenciais
- Estabilidade numérica é importante; exemplo bom:

$$\bar{x}_n = \bar{x}_{n-1} + \frac{x_n - \bar{x}_{n-1}}{n}$$

$$M_{2,n} = \sum_{i=1}^n (x_i - \bar{x}_n)^2$$

$$M_{2,n} = M_{2,n-1} + (x_n - \bar{x}_{n-1})(x_n - \bar{x}_n)$$

$$\sigma_n = \frac{M_{2,n}}{n}$$

$$\delta = \bar{x}_B - \bar{x}_A$$

$$\bar{x}_X = \bar{x}_A + \delta \cdot \frac{n_B}{n_X}$$

$$M_{2,X} = M_{2,A} + M_{2,B} + \delta^2 \cdot \frac{n_A n_B}{n_X}$$

# Janelas de tempo

Difícil prever custo de memória

```
select count(*)  
from stream.win:time(1 day)  
output last every 1 second
```

EPL

# Janelas de tempo

Difícil prever custo de memória

```
select count(*)  
from stream.win:time(1 day)  
output last every 1 second
```

EPL

Bilhões de eventos por dia? OOM na certa.

**java.lang.OutOfMemoryError: GC overhead limit exceeded**

# Agenda

Porque ter uma agenda sempre dá uma boa impressão

1 PROCESSAMENTO DE EVENTOS 101

WHAT?

2 POR QUE MAIS UMA LINGUAGEM DE CEP?

WHY?

3 DETALHES DA LINGUAGEM

HOW?

4 DEMOS

WTF?

The screenshot shows a web-based log viewer titled "logit". The interface includes a search bar, filter options (scroll automático, quebra de linhas, download: texto / json, Limpar resultados, Criar gist), and a status indicator (Monitorando). The main area displays a list of log entries from "monitor-wm-rs" dated 2014-09-23 at 10:21:25. The logs show multiple failed SSH authentication attempts from the IP 46.105.24.219, with the user "root" and the message "Failed password for root from 46.105.24.219 port 54298 ssh2". Other entries include successful logins and system logs from "http-monitor". The logs are color-coded by source and grouped by "MensagensSSH". A "feedback" button is visible at the bottom right.

Exibindo 339 de um total de 14.736.171 mensagens

monitor-wm-rs 2014-09-23 10:21:25 AUTHPRIV NOTICE sshd pam\_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=ns1.dinns.net user=root  
group: MensagensSSH

monitor-wm-rs 2014-09-23 10:21:24 AUTH INFO sshd Received disconnect from 46.105.24.219: 11: Bye Bye [preauth]  
group: MensagensSSH

monitor-wm-rs 2014-09-23 10:21:22 AUTHPRIV NOTICE sshd pam\_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=ns1.dinns.net user=root  
group: MensagensSSH

monitor-wm-rs 2014-09-23 10:21:24 AUTH INFO sshd Failed password for root from 46.105.24.219 port 54298 ssh2  
username: root group: FailSSH,MensagensSSH ip: 46.105.24.219

monitor-wm-rs 2014-09-23 10:21:21 AUTH INFO sshd Received disconnect from 46.105.24.219: 11: Bye Bye [preauth]  
group: MensagensSSH

monitor-wm-rs 2014-09-23 10:21:20 SYSLOG INFO http-monitor Starting 'Escolha as opções de entrega'

monitor-wm-rs 2014-09-23 10:21:21 AUTH INFO sshd Failed password for root from 46.105.24.219 port 54124 ssh2  
username: root group: FailSSH,MensagensSSH ip: 46.105.24.219

monitor-wm-rs 2014-09-23 10:21:20 SYSLOG INFO http-monitor type:assets => current\_url:http://ads.yahoo.com/pixel?id=2457686&t=2 fullTime:296 timestamp:1411480619 networkLatency:0 loadTime:0 description:img session:42fbc7b-f48b-485f-a0ac-df3aa82900a4 monitor:monitor-wm-rs.intelie.net mobile:true type:type:assets requestTime:0 response\_time:0 service\_description:

monitor-wm-rs 2014-09-23 10:21:20 SYSLOG INFO http-monitor type:xhr => current\_url:http://m.walmart.com.br/Site/Track.aspx?um=true&mobile=2&referrer=&url=http%3A%2F%2Fm.walmart.com.br%2Fproduto%2Fintegracao-Sige%2FIntegracao-Sige%2FTaff%2FTeste-Produto%3Fuam%3Dtrue fullTime:251 timestamp:1411479260 networkLatency:0 loadTime:0 description:xmlhttprequest session:242fbcb7b-f48b-485f-a0ac-df3aa82900a4 monitor:monitor-wm-rs.intelie.net mobile:true type:type:xhr requestTime:0 response\_time:0 service\_description:

# lognit

- Usado na **Globo.com**
- Processa bilhões de linhas de log por dia
- Índice **full-text** com Lucene modificado
- Sistema completamente **distribuído**

Como processar esses logs em tempo real?

LIVE Início Dashboards Apps Configurações

RUM · globo.com · Últimos 15 minutos

**78%** satisfatória

**10%** tolerável

**12%** frustrante

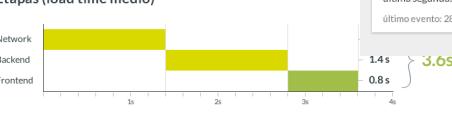
**Navegadores**

Navegador	Pageviews	Loadtime
Chrome	801.876	3.1s
Firefox	288.345	3.2s
Internet Explorer	221.460	3.2s
Safari	114.312	4.6s

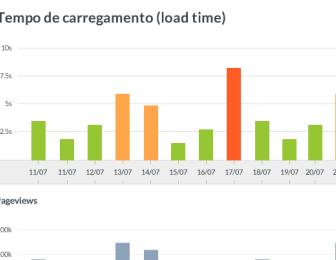
**Plataformas**

Plataforma	Pageviews	Loadtime
Desktop / notebook	801.876	3.1s
Smartphone	458.345	3.2s
Tablet	219.460	7.1s

**Etapas (load time médio)**



**Tempo de carregamento (load time)**



**Pageviews**

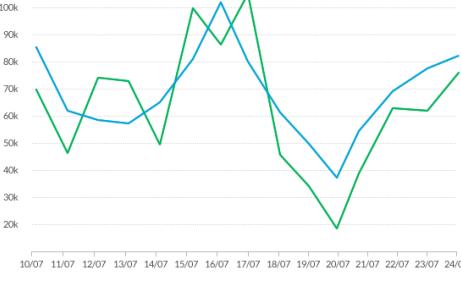


**Brasil**



- Estado
- São Paulo
- Rio de Janeiro
- Minas Gerais
- Paraná
- Bahia
- Rio Grande do Sul
- Santa Catarina
- Pernambuco
- Amazonas
- Distrito Federal
- Rio Grande do Norte

**Visitantes únicos (new X returning)**  
Nos últimos 15 dias



**New Unique Visitors** **238.617** ↑  
Mesmo horário na última segunda: 201.456  
último evento: 28/10/13 15:35

**Returning Unique Visitors** **316.987** ↓  
Mesmo horário na última segunda: 318.456  
último evento: 28/10/13 15:35

**Bounce Rate** **15%** ↓  
Mesmo horário na última segunda: 17%  
último evento: 28/10/13 15:35

**Search Results by Term**

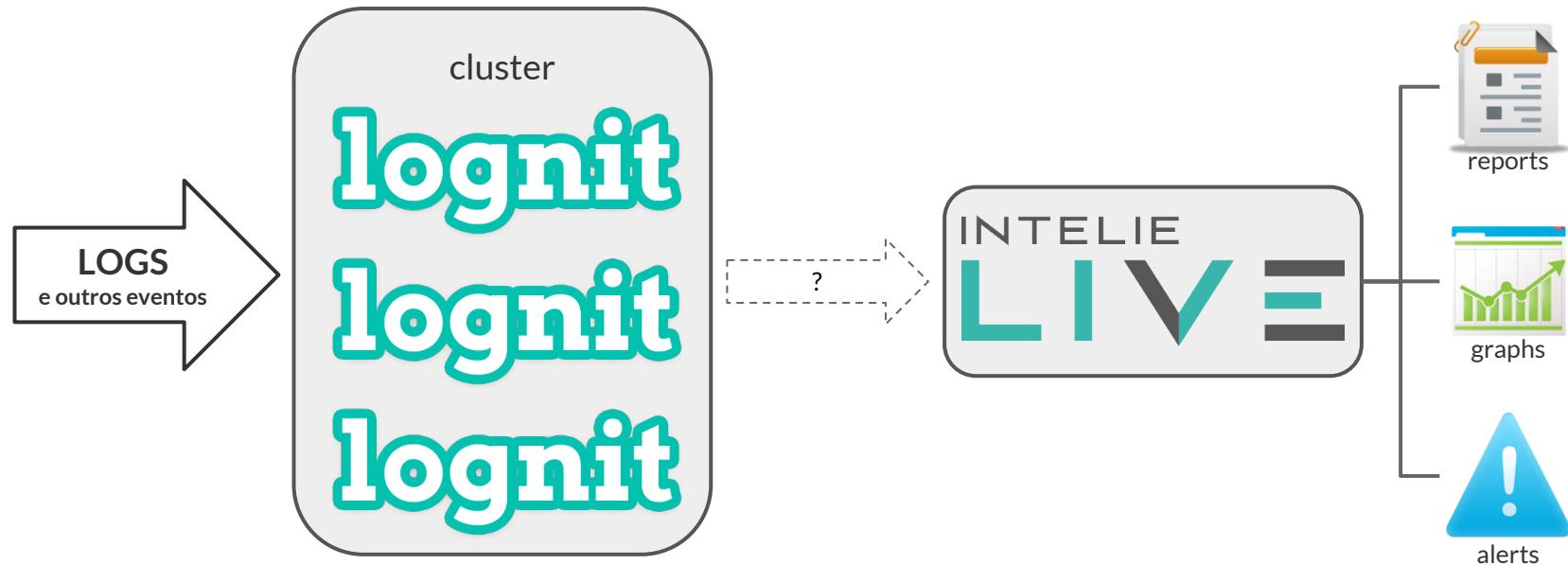
Rank	Term	Count
1	máquina de costura elgin	879
2	secador	860
3	celular galaxy s	756
4	pneus	712
5	GTA v edition colecionador para x box	543
6	martelete	501
7	camera digital	326
8	prancha de cabelo	321
9	jogos ps3	211
10	tv full hd	145
11	walmart	96

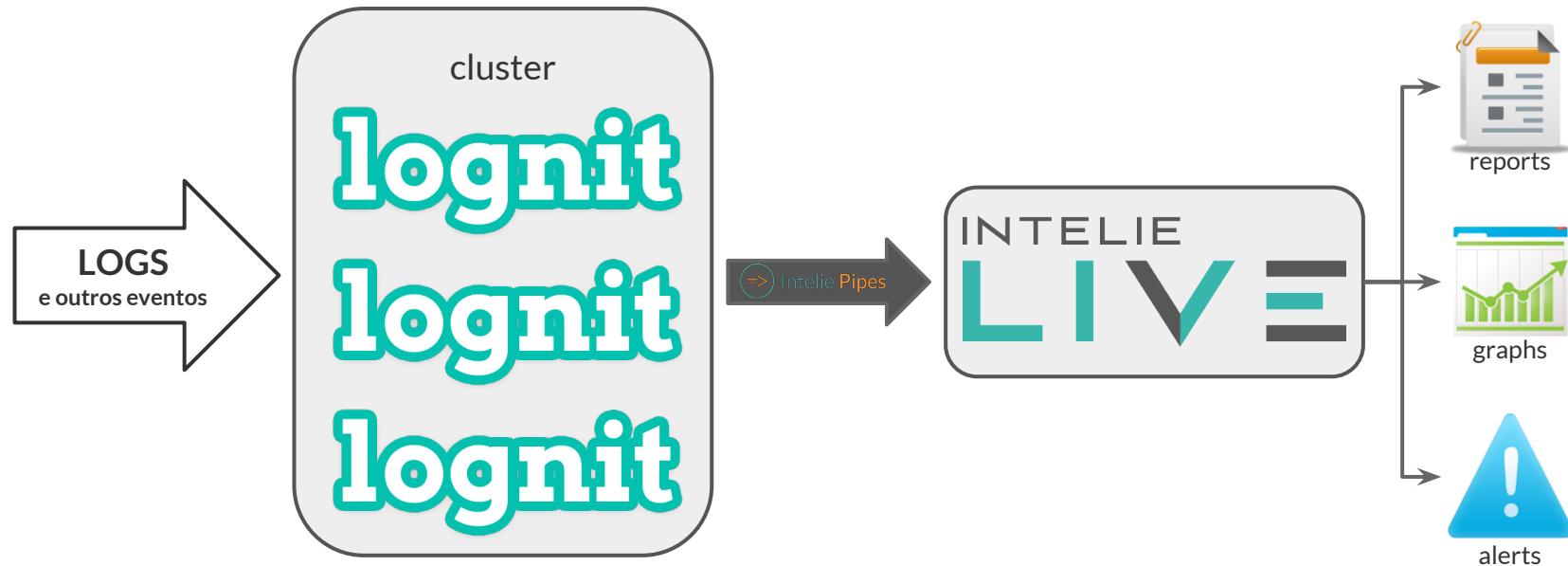
**Search Results**  
Subtítulo entra aqui **105.043** ↓  
Mesmo horário na última segunda: 106.879  
último evento: 28/10/13 15:35



- Usado na **Globo.com** e **Walmart.com**
- Processa tanto eventos de **infra** quanto de **negócio**
- Permite criar **dashboards, alertas, relatórios**, etc.
- Processa **centenas** de eventos/segundo tranquilamente
- Usa **Esper** como um dos motores de regras

Como escalar?





# Objetivos primários

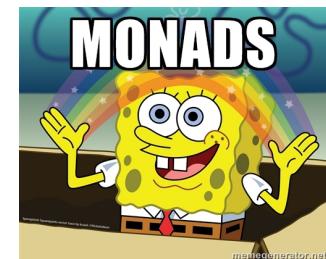
Não dava para simplesmente jogar o Esper aí

- Processar **bilhões** de eventos por dia (até 300 mil/s)
- Suportar **milhares** de queries simultâneas
- **Permitir processamento distribuído!**
- Não onerar a rede excessivamente
- Baixo consumo de memória
- Em apenas uma passagem pelos dados

# Objetivos secundários

Já que vamos criar uma linguagem nova...

- Filtros baseados na sintaxe do **Lucene**
- Poder escrever consultas em apenas **uma linha**
- Design **monádico** (esqueça SQL)
- Mesmo permitindo processamento distribuído, também fornecer ferramentas para **single-node**
- Custo de **memória** o mais **previsível** possível



# Agenda

Porque ter uma agenda sempre dá uma boa impressão

1 PROCESSAMENTO DE EVENTOS 101

WHAT?

2 POR QUE MAIS UMA LINGUAGEM DE CEP?

WHY?

3 DETALHES DA LINGUAGEM

HOW?

4 DEMOS

WTF?

# Links

Eu disse que voltaria; no final ele aparece de novo

- Slides

[juanlopes.net/weop2014](http://juanlopes.net/weop2014)

- Site e documentação

[pipes.intelie.com](http://pipes.intelie.com)

- Tutorial interativo

[pipes.intelie.com/tutorial](http://pipes.intelie.com/tutorial)

# Exemplo motivador

O mínimo que precisava ser feito

```
field:value => count() every minute
```

PIPES

filtro

agregação

# Filtros

Já que Lucene não tem real-time, implementamos o nosso

```
type: http status: 404
```

PIPES

# Filtros

Já que Lucene não tem real-time, implementamos o nosso

```
type:http status:404
```

PIPES

```
type:http status:200
```

PIPES

```
type:http status:(2?? | 3??)
```

PIPES

# Filtros

Já que Lucene não tem real-time, implementamos o nosso

```
type:http status:404
```

PIPES

```
type:http status:200
```

PIPES

```
type:http status:(2?? | 3??)
```

PIPES

# Filtros

Já que Lucene não tem real-time, implementamos o nosso

```
type:http status:404
```

PIPES

```
type:http status:200
```

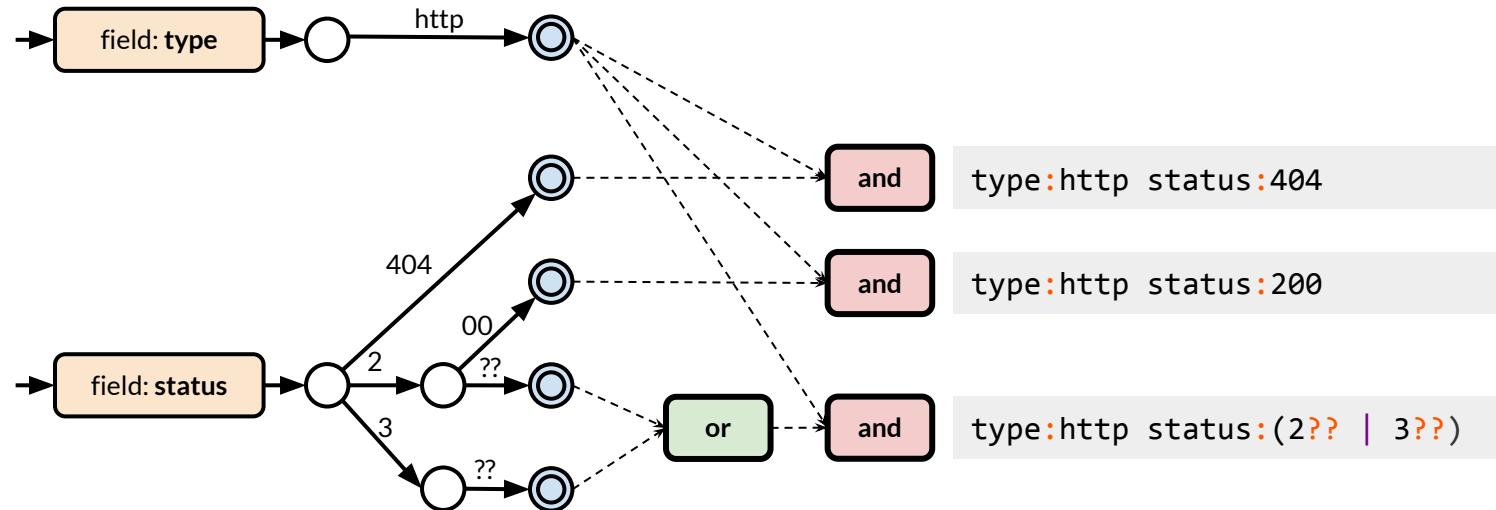
PIPES

```
type:http status:(2?? | 3??)
```

PIPES

# Autômatos de filtro

Minimizam o consumo de CPU dos filtros



# Autômatos na unha

Regex match em Java é NP-difícil (em Ruby, Python, Perl etc. também)

**Teorema 1** MATCH é NP-difícil.

*Demonstração:* 3SAT é redutível para MATCH utilizando a seguinte lógica: para resolver uma fórmula com  $n$  variáveis e  $m$  cláusulas, define-se a expressão regular

$$E = \wedge(x?)^n.*; (? : \backslash 1v_{11}|\backslash 2v_{12}|\backslash 3v_{13}), (? : \backslash 1v_{21}|\backslash 2v_{22}|\backslash 3v_{23}), \dots (? : \backslash 1v_{m1}|\backslash 2v_{m2}|\backslash 3v_{m3}), \$$$

e a string a ser reconhecida

$$S = x^n; x^m$$

onde  $v_{ij} = x$  se a variável  $j$  da cláusula  $i$  aparecer negada na fórmula, caso contrário,  $v_{ij} = \epsilon$ . Isto força o reconhecimento da *backreference*  $\backslash j$  como  $x$  ou  $\epsilon$  respectivamente, porém pelo menos uma das variáveis precisa reconhecer a substring  $x$ , definida para cada cláusula na string  $S$ . Assim, existem grupos de captura  $C = (c_1, c_2, \dots, c_n)$  se e somente se for possível reconhecer tanto as expressões dos grupos de captura ( $x$  ou  $\epsilon$  para *verdadeiro* ou *falso*, respectivamente), como as *backreferences* que representam as cláusulas da fórmula de forma consistente. Como tanto a expressão  $E$  quanto a string  $S$  podem ser construídas em  $O(n + m)$ , MATCH é NP-difícil. ■

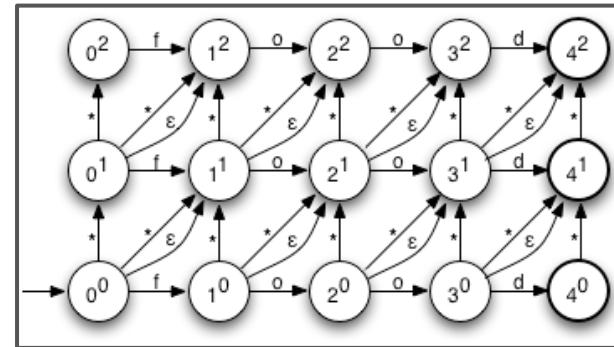
[github.com/juanplopes/pyrex](https://github.com/juanplopes/pyrex)

# Autômatos na unha

Em alguns casos regex nem ajudaria tanto

text:food~2

PIPES



[blog.notdot.net/2010/07/Damn-Cool-Algorithms-Levenshtein-Automata](http://blog.notdot.net/2010/07/Damn-Cool-Algorithms-Levenshtein-Automata)

# Tipos de filtro implementados

Nem todos os tipos de filtro do Lucene foram implementados

→ Match all:

\*

→ Term:

<field>:<term>

→ Range:

<field>:[<lower>, <upper>]

→ Fuzzy:

<field>:<term>~<number>

→ AND:

<filter> & <filter>

→ OR:

<filter> | <filter>

→ NOT:

-<filter>

# Composição de filtros

## Um pouco mais poderosa até que a do Lucene

WebAccess

PIPES

# Composição de filtros

Um pouco mais poderosa até que a do Lucene

WebAccess

PIPES

WebAccess browser:firefox

PIPES

# Composição de filtros

Um pouco mais poderosa até que a do Lucene

WebAccess

PIPES

WebAccess browser:firefox

PIPES

WebAccess browser:(firefox|chrome)

PIPES

# Composição de filtros

Um pouco mais poderosa até que a do Lucene

WebAccess

PIPES

WebAccess browser:firefox

PIPES

WebAccess browser:(firefox|chrome)

PIPES

WebAccess browser:(firefox|chrome|(IE\* product\_id#[15, 60]))

PIPES

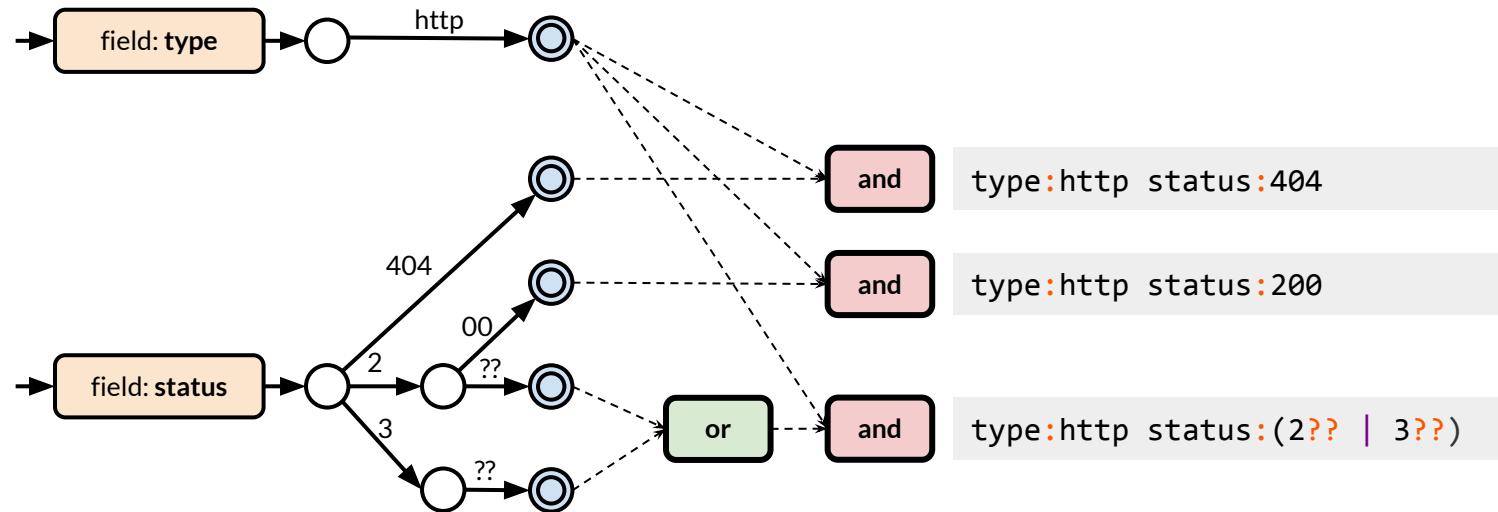
# Filtros

Um pequeno resumo

- Filtros não sabem dizer se um evento dá match ou não
- Apenas sabem criar o autômato que reconhece um valor
- Registrar um filtro significa adicionar ao autômato transições que reconheçam o valor e listeners para estados de aceite
- Os eventos são submetidos aos autômatos (um para cada campo) e os matchs são notificados e encadeados
- Código complexo (e com alta cobertura)

# Filtros

Um pequeno resumo



# Tipagem estática, mas sem schema

O máximo para evitar problemas

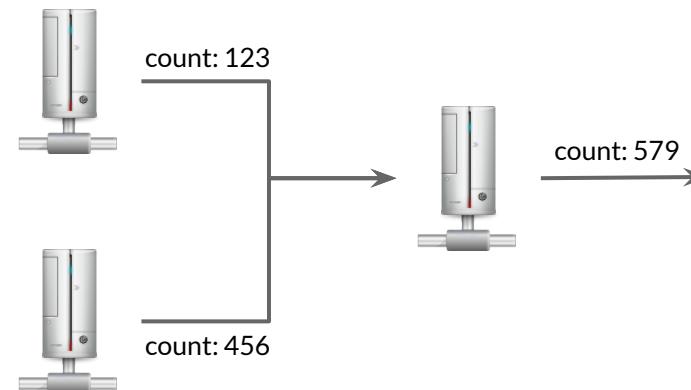
- A linguagem não assume nada sobre o tipo dos fluxos
- Tipos são codificados na consulta, não em um schema externo
- Existe apenas um fluxo global e filtros sobre ele
- Somente 8 tipos são tratados pela linguagem

# Agregações

## Uma abordagem bem diferente do tradicional

\* => `count()` every minute

PIPES

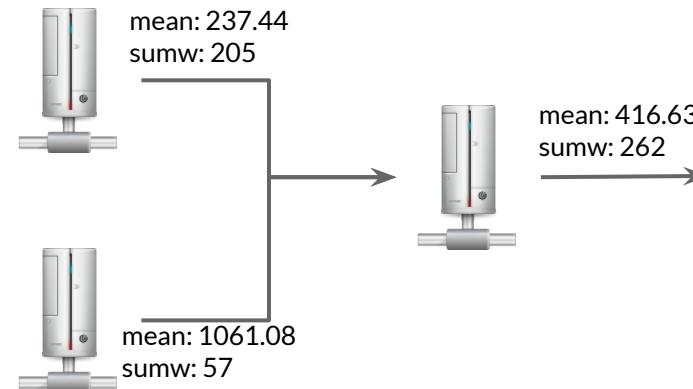


# Agregações

## Uma abordagem bem diferente do tradicional

\* => avg(field#) every minute

PIPES



# Aggregações

Cada agregação tem uma representação intermediária

```
* => avg(response_time#):describe at the end  
> {"sumw":2125.0, "mean":2.420657513400854}
```

MIXED

```
* => (variance(response_time#)*count()):describe at the end  
> {"trees": [{"sumw":2125.0, "mean":2.420657513400854, "m2":3582.7099512146638}, {"value":3481}]}  
{"sumw":2125.0, "mean":2.420657513400854, "m2":3582.7099512146638}, {"value":3481}]}
```

MIXED

```
* => median(response_time#):describe at the end  
> {"sketch": {"e": 4.147768903288338E-5, "width": 65536, "M": [...], "mask": 65535, "total": 2125.0}}
```

MIXED

# Aggregações

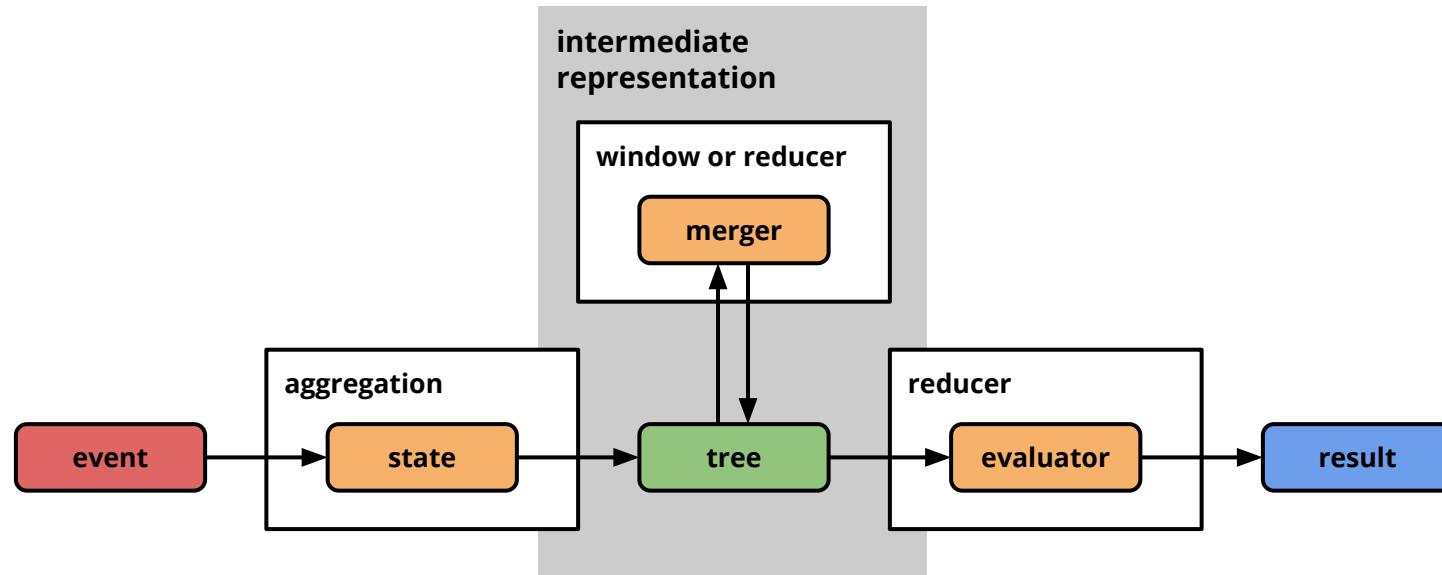
Toda agregação precisa implementar três conceitos

```
public interface Aggregation<T> extends Expression<T> {  
    State newState(int flips);  
    Merger newMerger();  
    T eval(Tree tree, WindowBounds bounds);  
}  
  
public interface State {  
    void yield(Object obj);  
    Tree flip();  
}  
  
public interface Merger {  
    void add(Tree tree);  
    void remove(Tree tree);  
    void clear();  
    Tree get();  
}
```

JAVA

# Agregações

Toda agregação precisa implementar três conceitos

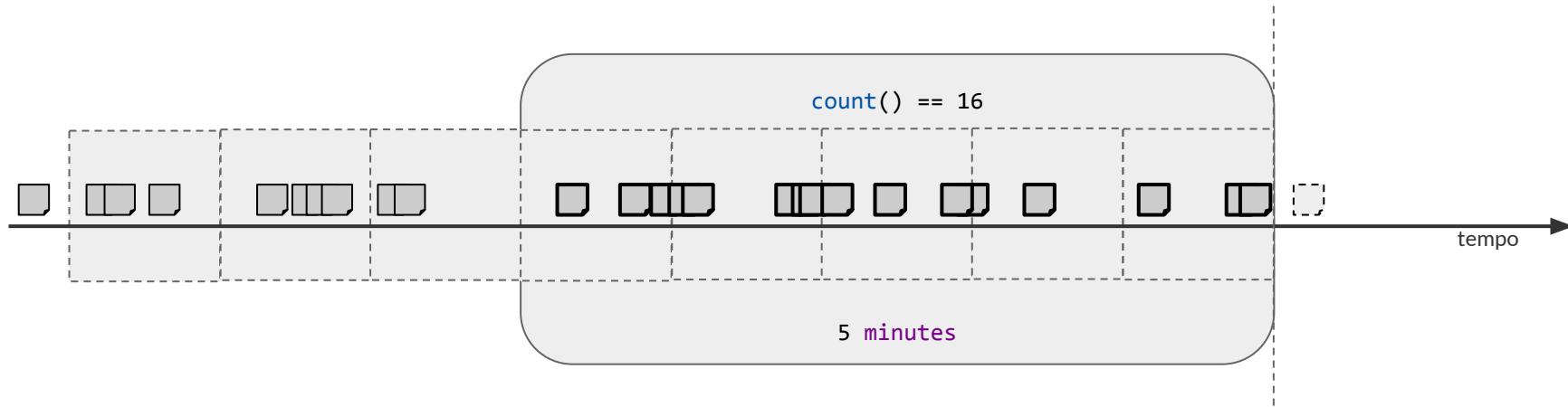


# Janelas de eventos

Usam o mesmo mecanismo para mesclar resultados

\* => `count() over last 5 minutes every minute`

PIPES

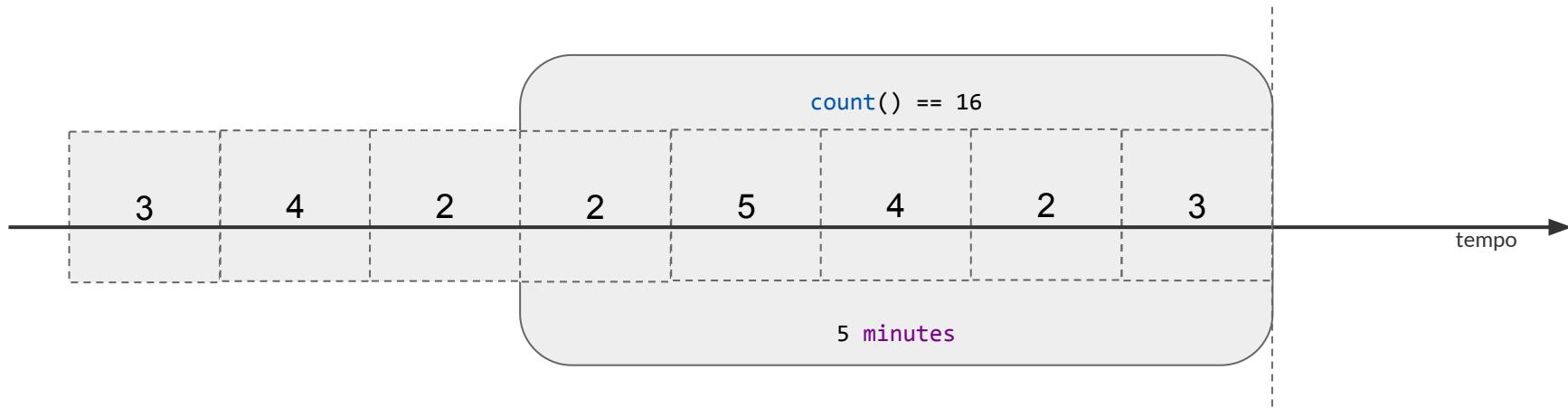


# Janelas de eventos

Usam o mesmo mecanismo para mesclar resultados

\* => `count() over last 5 minutes every minute`

PIPES



# Janelas de eventos

Usam o mesmo mecanismo para mesclar resultados

## → Vantagens

- ◆ Quantidade de memória previsível em compile-time
- ◆ Permite janelas muito maiores (mês, ano, etc.)

## → Desvantagens

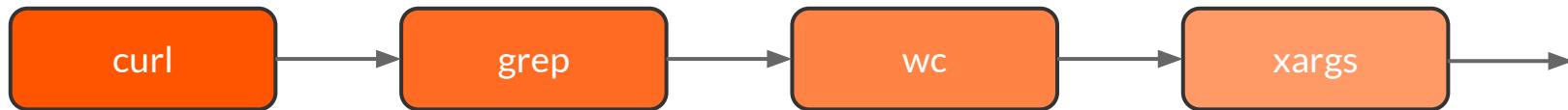
- ◆ Output precisa ser compatível com janela
- ◆ Janela precisa ter tamanho múltiplo do output

# Por que o nome é Pipes?

## Unix pipelines

```
curl http://example.com | grep somestring | wc -l | xargs echo 'Result:'
```

SHELL



# Encadeamento de queries

A saída de uma torna-se a entrada de outra

WebAccess

```
=> count() by host over last 2 hours every 5 minutes  
=> @sort _ desc
```

PIPES

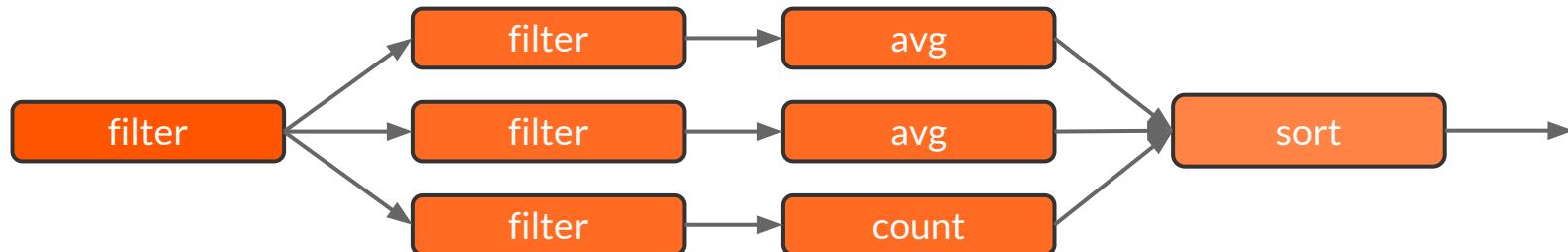


# Encadeamento de queries

A saída de uma torna-se a entrada de outra

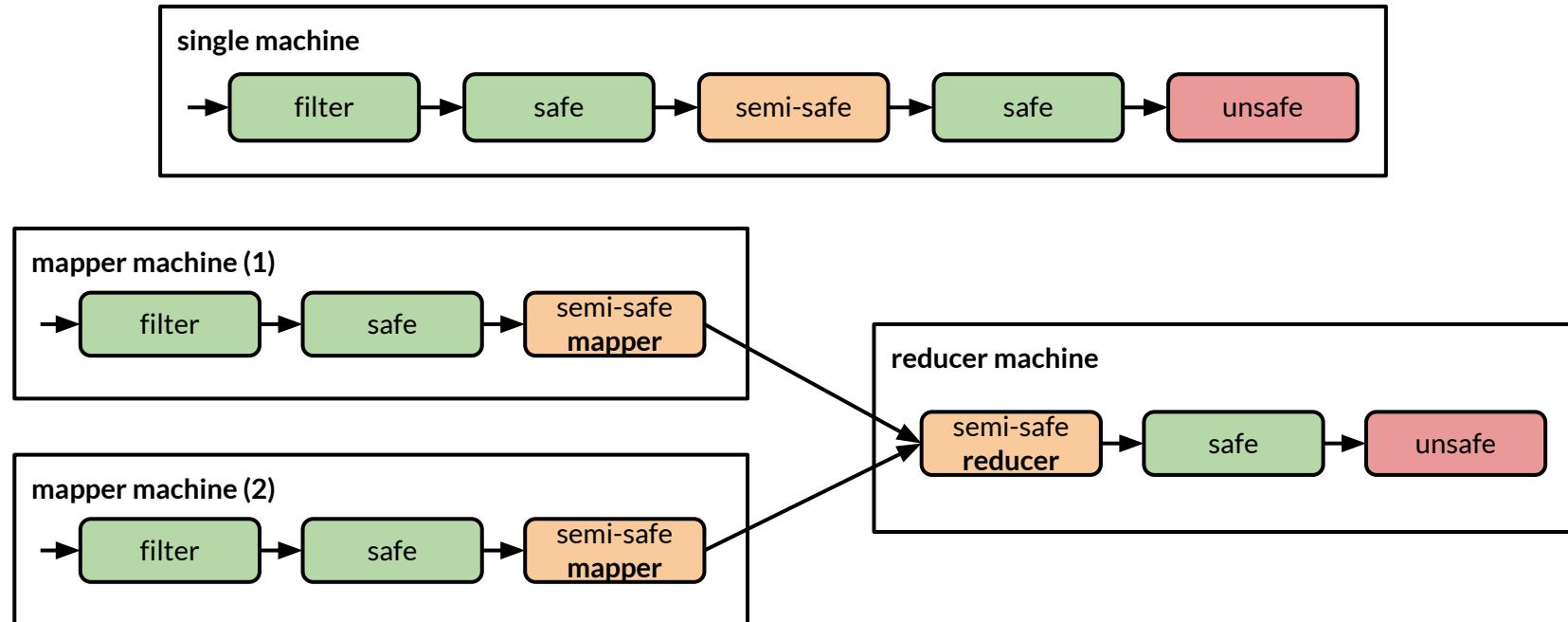
```
Metric|WebAccess
=> [
    @filter \Metric kind:memory => avg(value#):bytes as mem by host every hour
    join on host
    @filter \Metric kind:'cpu load' => avg(value#):format('0%') as cpu by host every hour
    join on host
    @filter \WebAccess => count() as requests by host every hour
]
=> @sort requests desc
```

PIPS



# Encadeamento de queries

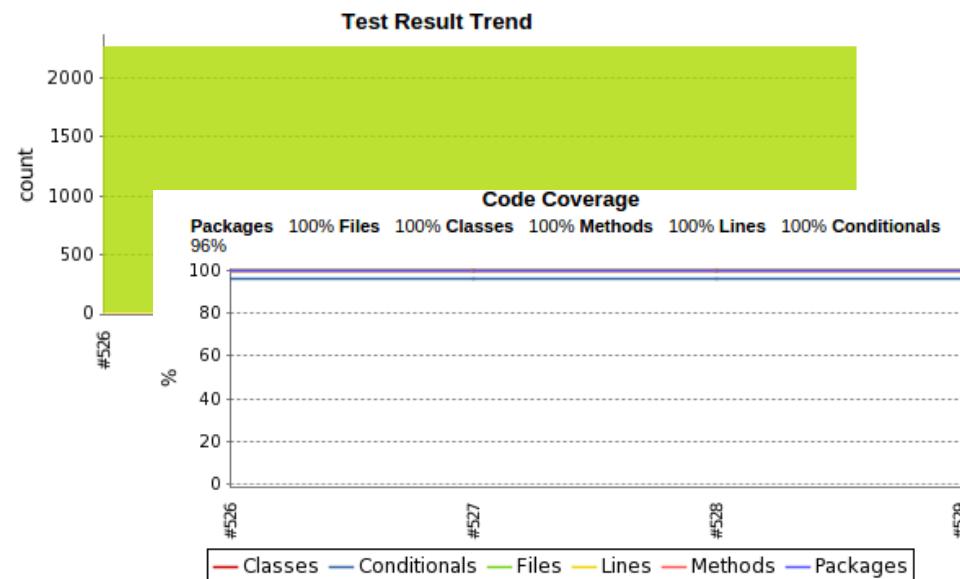
A saída de uma torna-se a entrada de outra



# Testes, testes e mais testes

Isso não deveria ser mais novidade

- 17K functional SLOC
- 21K test SLOC
- 11K generated SLOC
- 2260 test cases
- 100% line coverage
- 96% branch coverage



# Agenda

Porque ter uma agenda sempre dá uma boa impressão

1 PROCESSAMENTO DE EVENTOS 101

WHAT?

2 POR QUE MAIS UMA LINGUAGEM DE CEP?

WHY?

3 DETALHES DA LINGUAGEM

HOW?

4 DEMOS

WTF?

# Links

Agora pode anotar

- Slides

[juanlopes.net/weop2014](http://juanlopes.net/weop2014)

- Site e documentação

[pipes.intelie.com](http://pipes.intelie.com)

- Tutorial interativo

[pipes.intelie.com/tutorial](http://pipes.intelie.com/tutorial)

Obrigado!

@juanplopes