DataFlow

Como Fazer Fluxos de Dados Facilmente em Perl

Alexei “Russo” Znamensky
russoz@cpan.org

História

“Hack de Dados Públicos” como idéia de tema para o Equinócio de Março/2011 da São Paulo Perl Mongers

Projeto OpenData-BR

Idéias transformadas em código

OpenData: O Começo

Scripts para buscar, transformar e publicar dados públicos

Portal da Transparência do Gov.BR

Módulos para facilitar a criação de scripts

OpenData: O Fim

Prover um framework que promova os conceitos de “OpenData”

Permita às pessoas analisar e manipular dados públicos

E o meio?

Várias APIs e frameworks para várias tarefas diferentes

Novos projetos sempre levam a muito (re-)trabalho de codificação

- Isso pode ser bom
- Isso pode ser ruim

E o meio?

Várias APIs e frameworks para várias tarefas diferentes

Novos projetos sempre levam a muito (re-)trabalho de codificação

- Isso pode ser bom **(APROVEITAR!)**
- Isso pode ser ruim **(EVITAR!)**

E o meio?

A seqüência de passos usual é muito parecida na maior parte das vezes, mas ...

... não queremos “engessar” o modelo, é preciso ter flexibilidade para mudarmos as coisas, afinal ...

“there is more than one way to do it” (TIMTOWTDI => Tim Toady!), mas

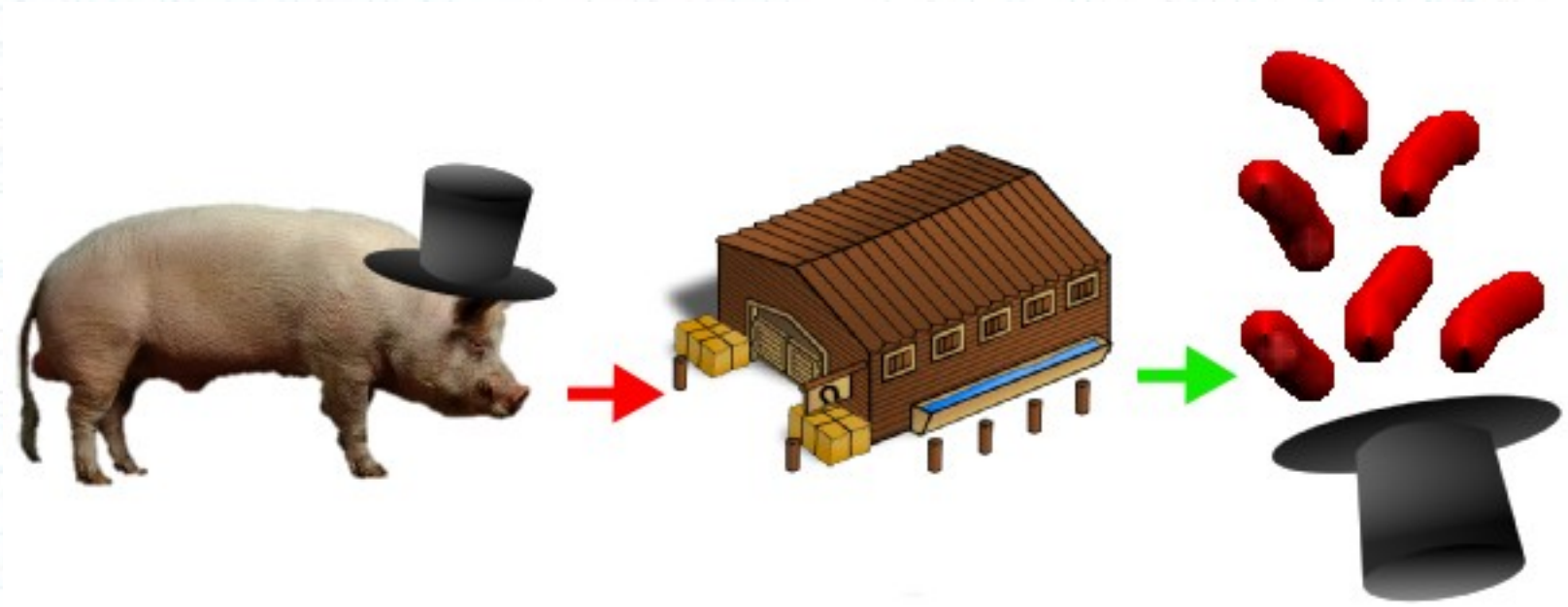
“but sometimes consistency is not a bad thing either” (BSCINABTE => Bicarbonate!)

O Meio Precisa

Prover um modelo de tratamento de dados que seja:

- Simples
- (Facilmente) Customizável
- (Facilmente) Extensível
- (Facilmente) Utilizável
- (Facilmente) Re-utilizável

O que por no Meio?



O que precisamos?

Algo Simples

Algo que:

Simplesmente funcione

Permita códigos sofisticados com pouco tempo de aprendizado

Permita códigos poderosos com poucas linhas de código

Possua várias funcionalidades já prontas para serem usadas

Algo Customizável

Permita encaixar peças, como no Linux
(`cat file | grep foo | sort | uniq`)

Permita montar “scripts”, como no Linux

“Defaults” razoáveis, mas que possam ser manipulados

Algo Extensível

Permita criar peças novas

Para caso único, específico, ou ...

... um componente novo (“black box”) que possa ser usado pelos outros

Algo Utilizável

Não seja necessário “programar”, no sentido test/loop/variáveis da palavra

Criação de fluxos definidos por arquivos de configuração ...

... que poderiam ser criados por interfaces gráficas/web, por leigos

Algo Re-Utilizável

Fluxos de dados que possam ser utilizados em outros fluxos (sub-fluxos?)

Fluxos disponibilizados como:

- Programas (linha de comando)
- Sistemas acessados pela Web
- Web Services? <Buzzword> Services?
- Consumidores/Produtores de filas de mensagens?

Caixinhas Mutantes

Projeto OpenData-BR [Dez '10]

- pacotes OpenData::AZ::Box*
- pacotes OpenData::Flow::*

Projeto DataFlow

- pacotes DataFlow::Node::* [Fev '10]
- pacotes DataFlow::* [Mar '10]

Projeto EM ANDAMENTO!!!

Em Perl

Projeto nasceu na comunidade Perl

Perl sempre foi a melhor linguagem para a manipulação de textos

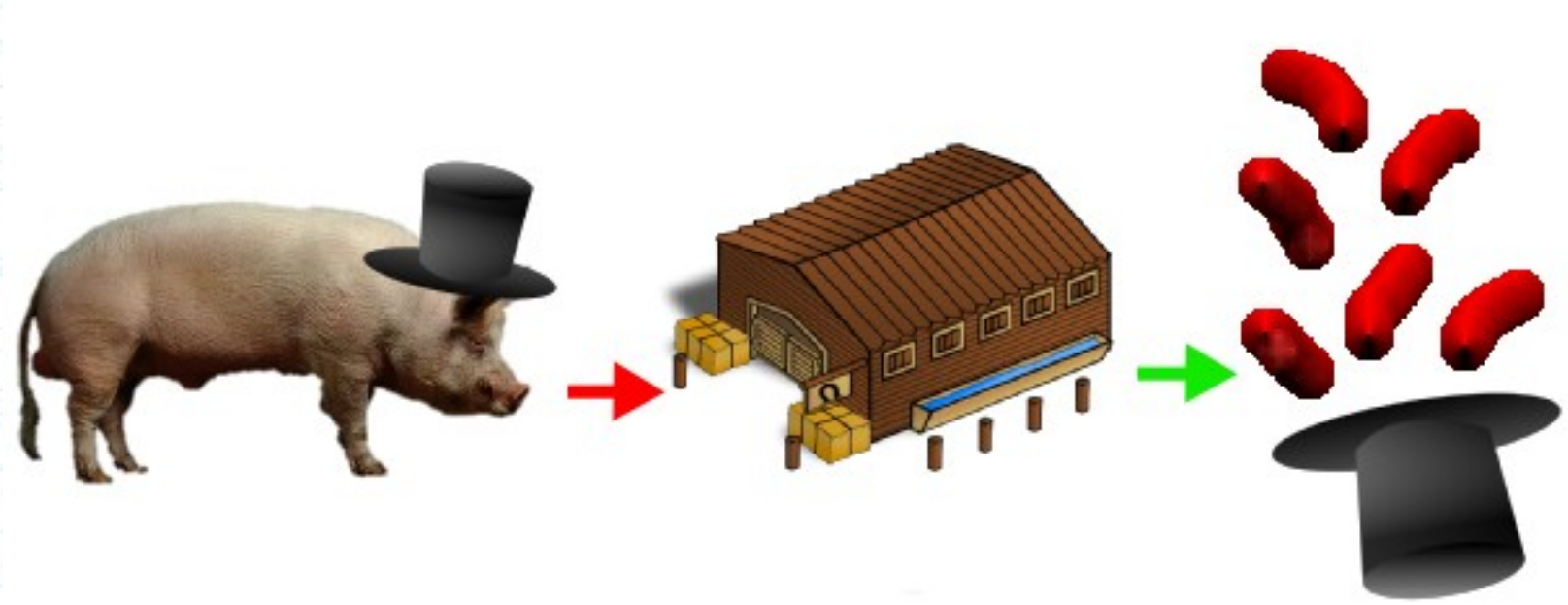
Boas práticas de desenvolvimento

Cultura de testes muito forte

Comunidade

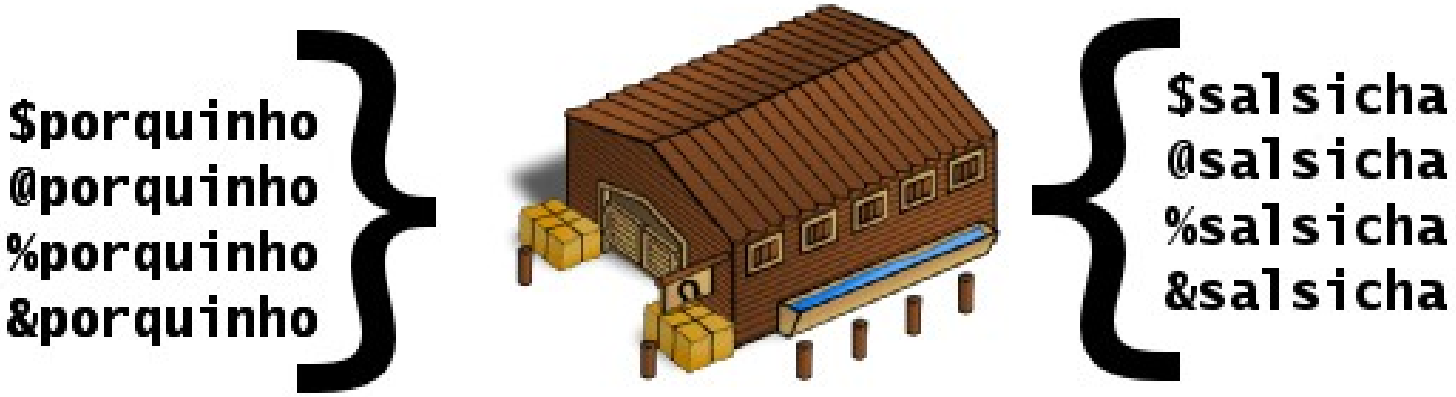
É divertido :-)

DataFlow



DataFlow

Em termos de Perl, precisamos



Show me The Code!!

Um Fluxo Simples

Um fluxo de dados

```
my $flow = DataFlow->new(
    procs => [
        sub {
            return uc(shift);
        },
        sub {
            return scalar reverse(shift);
        },
    ],
);
```

Um Fluxo Simples

Usando:

```
$flow->input('batatas', 'bananas');
print $flow->output();
print $flow->output();

# ou, de outra forma

print join “\n”,
    $flow->process('batatas', 'bananas');
```

Resultado:

```
SATATAB
SANANAB
```

Mais Que Subs

Cada “sub” do fluxo torna-se um objeto do tipo **DataFlow::Proc**, um “Processador”

```
my $flow = DataFlow->new(
    procs => [
        DataFlow::Proc->new(
            p => sub {
                return uc(shift);
            },
        ),
        DataFlow::Proc->new(
            p => sub {
                return scalar reverse(shift);
            },
        ),
    ],
);
```

Debaixo Dos Caracóis

DataFlow::Proc

- Recebe um *scalar* e devolve uma lista (com zero ou mais elementos)

DataFlow

- Array de Procs, com filas entre si, para "bufferizar" o consumo dos dados

Em Moose: um sistema de objetos pós-moderno para Perl

Criando Processadores

Quando:

- Uma mesma “sub” é utilizada com muita freqüência
- Se quer fazer uma “sub” parametrizada e que possa ser reutilizada
- Existe uma necessidade semântica de dar um nome a um tipo de rotina

Criando Processadores

My/Proc/CSV.pm

```
package My::Proc::CSV;

use Moose; extends 'DataFlow::Proc';

has 'csv' => ( isa => 'Text::CSV' );

has '+p' => (
    'default' => sub {
        my $self = shift;
        return sub {
            return $self->csv->parse(shift);
        }
    },
);
```

E Fluxos

My/Flow/Weird.pm

```
package My::Flow::Weird;

use Moose; extends 'DataFlow';

has '+procs' => (
    'default' => sub {
        return [
            sub { return uc(shift) },
            sub { scalar reverse(shift) },
        ];
    },
);
```

Para Dentro!

DataFlow dentro de um Proc ou de outro Flow, e vice-versas:

```
my $flow = DataFlow->new(
    procs => [
        My::Flow::Weird->new(),     # um flow
        My::Proc::CSV->new(),       # um proc
    ],
);
```

(Alguns dos) Procs Prontos

Alguns Procs disponíveis:

CSV

processar “de” e “para” conteúdo .csv

URLRetriever

busca páginas/arquivos da web

HTMLFilter

filtro para conteúdo HTML

SimpleFileInput/SimpleFileOutput

leitura/escrita de arquivos

Encoding (em finalização)

mudar codificações de caracteres

Extração de Dados (Scrape)

Exemplo no repositório do DataFlow, no arquivo **ceis.pl**:

- http://bit.ly/kyJLXx

Próximas Melhorias

Meta-dados no fluxo

Mais Procs Prontas

- Split e Join de fluxos
- Acesso a banco de dados
- Execução de comando externo
- OCR (reconhecimento de caracteres)
- XML => RDF
- JSON

Próximas Melhorias

Construção de DataFlow a partir de uma configuração em texto:

- YAML, JSON, XML, etc...

Paralelismo

- Processos, threads

Mecanismo de log de mensagens

Referências DataFlow

Repositório

http://github.com/russoz/DataFlow

CPAN

http://search.cpan.org/perldoc?DataFlow

Testers Matrix

http://www.cpantesters.org/distro/D/DataFlow

Referências OpenData

OpenData-BR

- http://opendatabr.org/

Portal da Transparência do Gov.BR

- http://www.portaltransparencia.gov.br/

Ref. Desenvolvimento

- git
  - http://git-scm.com/
  - http://github.com
- Moose
  - http://www.iinteractive.com/moose/
- Dist::Zilla
  - http://dzil.org/
- git flow
  - http://bit.ly/9bUvdC

Muito Obrigado!

Thank You!

Большое Спасибо!

Customização & Extensão

- Podemos montar “flows” arbitrariamente grandes e...
- ... colocar coisas arbitrariamente complexas nos “procs”.
- Podemos estender ou agregar um (ou mais) **DataFlow** a outro código.

Demo

- ceis.pl