Tecnologia em Gestão

P&D

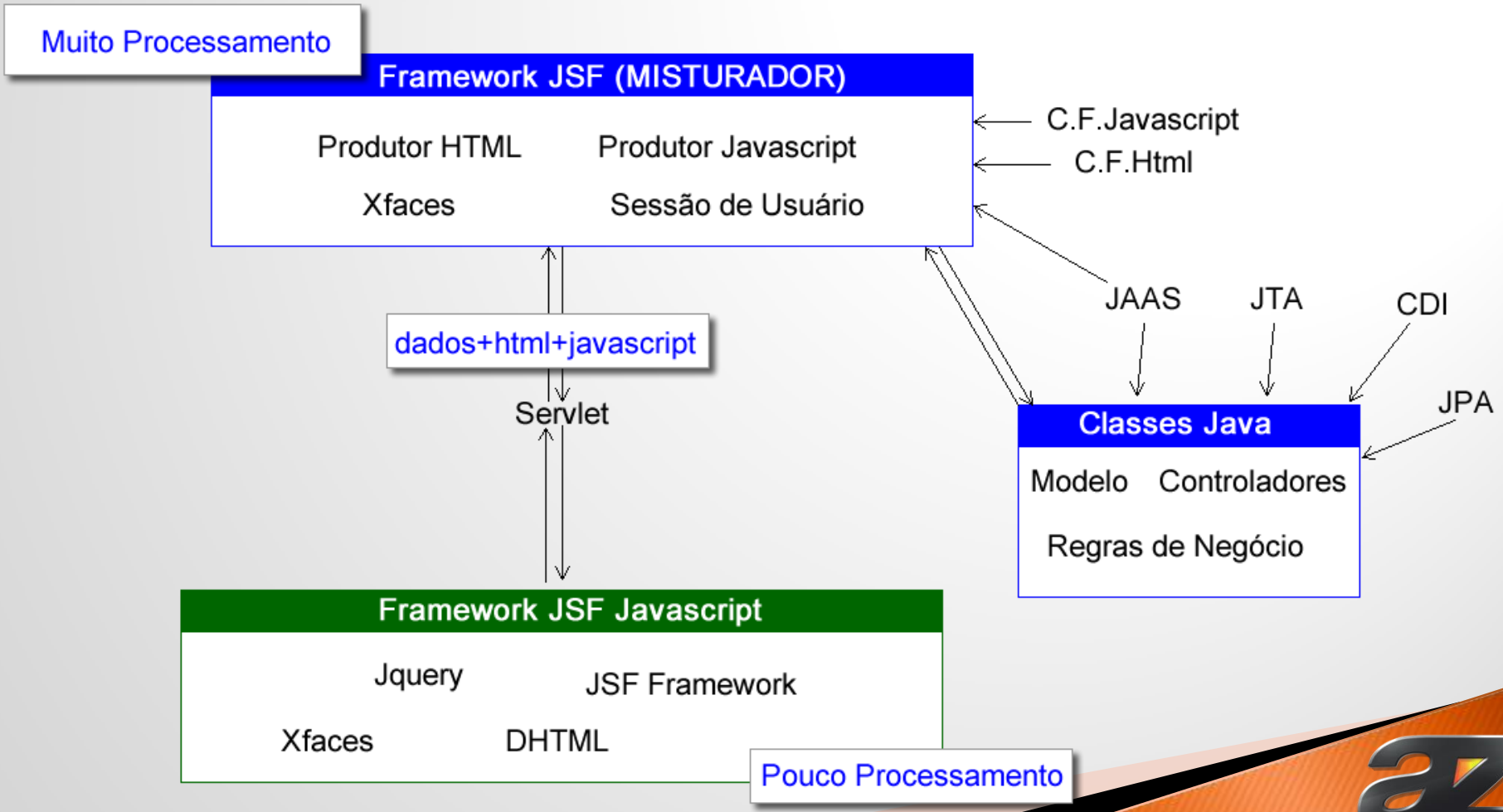
Pesquisa Desenvolvimento e Inovação

Java com AngularJS

- Visão Geral
- Aplicação Exemplo (contatos)
- JSON
- Comunicação por HTTP
  - REQUEST
  - RESPONSE
- Restful
- SPA (Single Page Application)

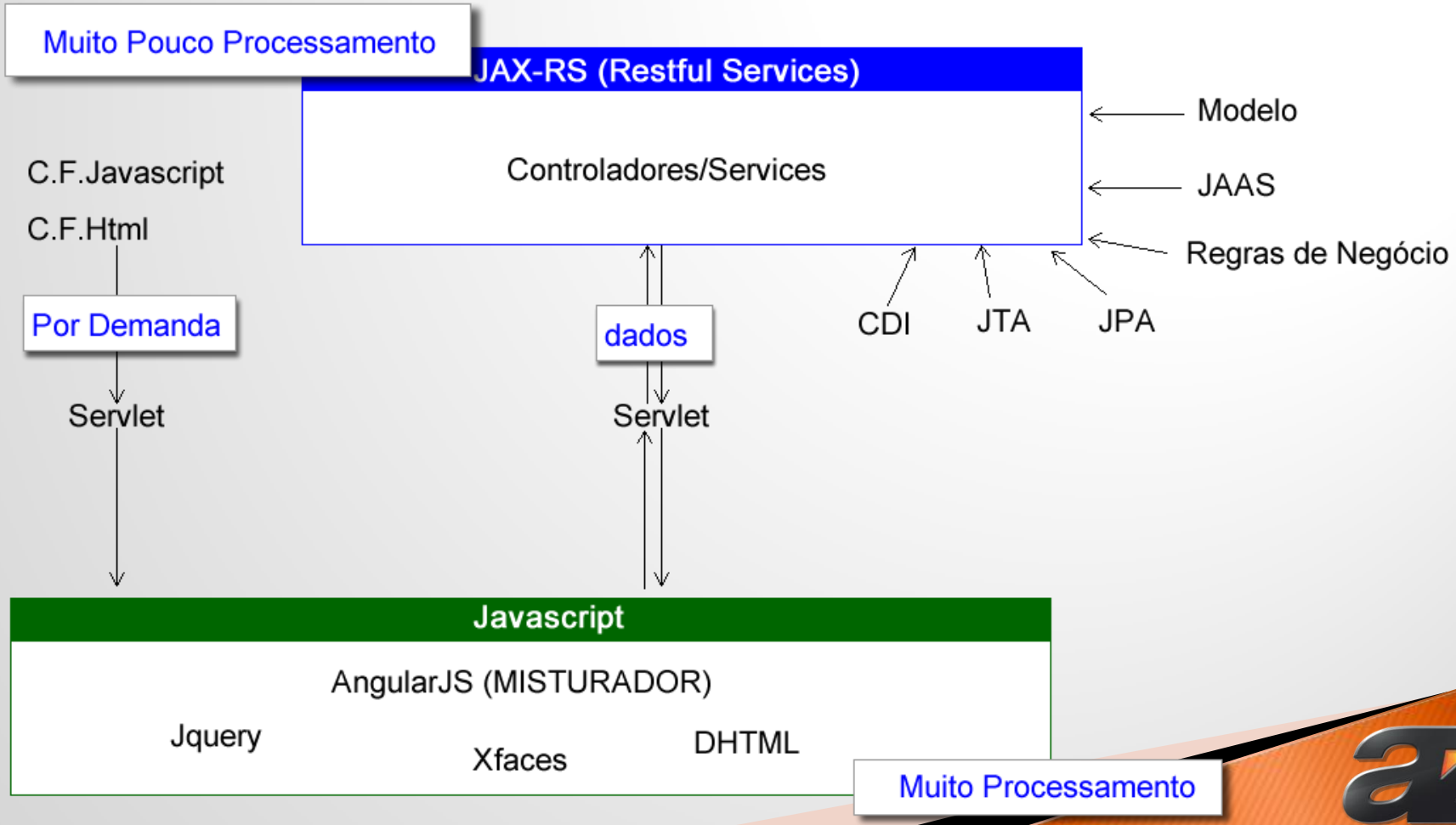
Visão Geral

Round-trip

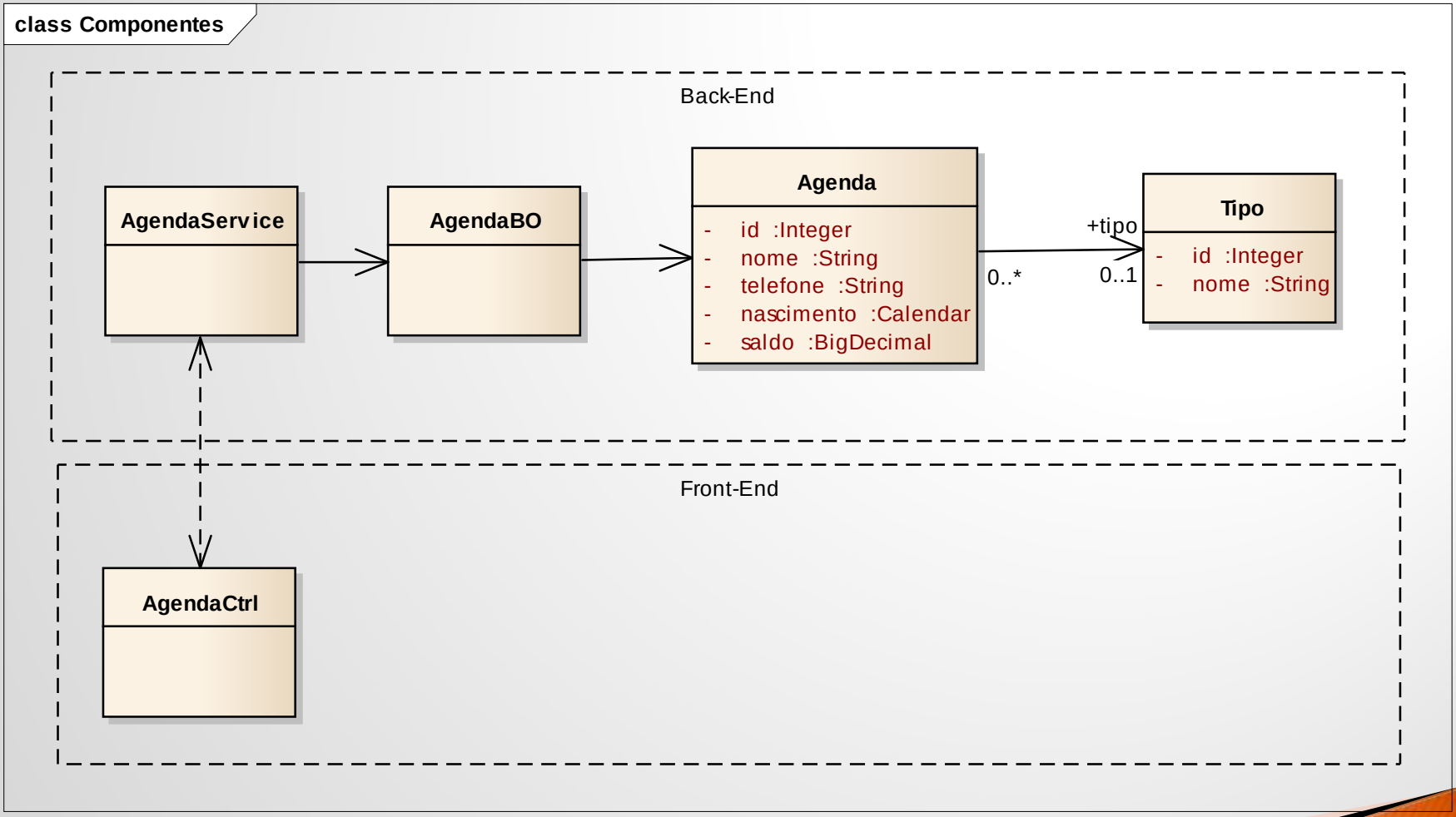


Visão Geral

SPA (Simple Page Application)



Aplicação Exemplo



JSON (Javascript Object Notation)

- Criado por Douglas Crockford e definido pela RFC 4627
- https://tools.ietf.org/html/rfc4627
- http://json.org/

- Exemplo:

{“nome”:“joao”,“idade”:22,“salario”:2342.11,“interesses”:[“musica”,”arte”],”dependentes”:[{“nome”:“camila”,“idade”:6},{“nome”:“joana”,“idade”:21}]}

JSON (Javascript Object Notation)

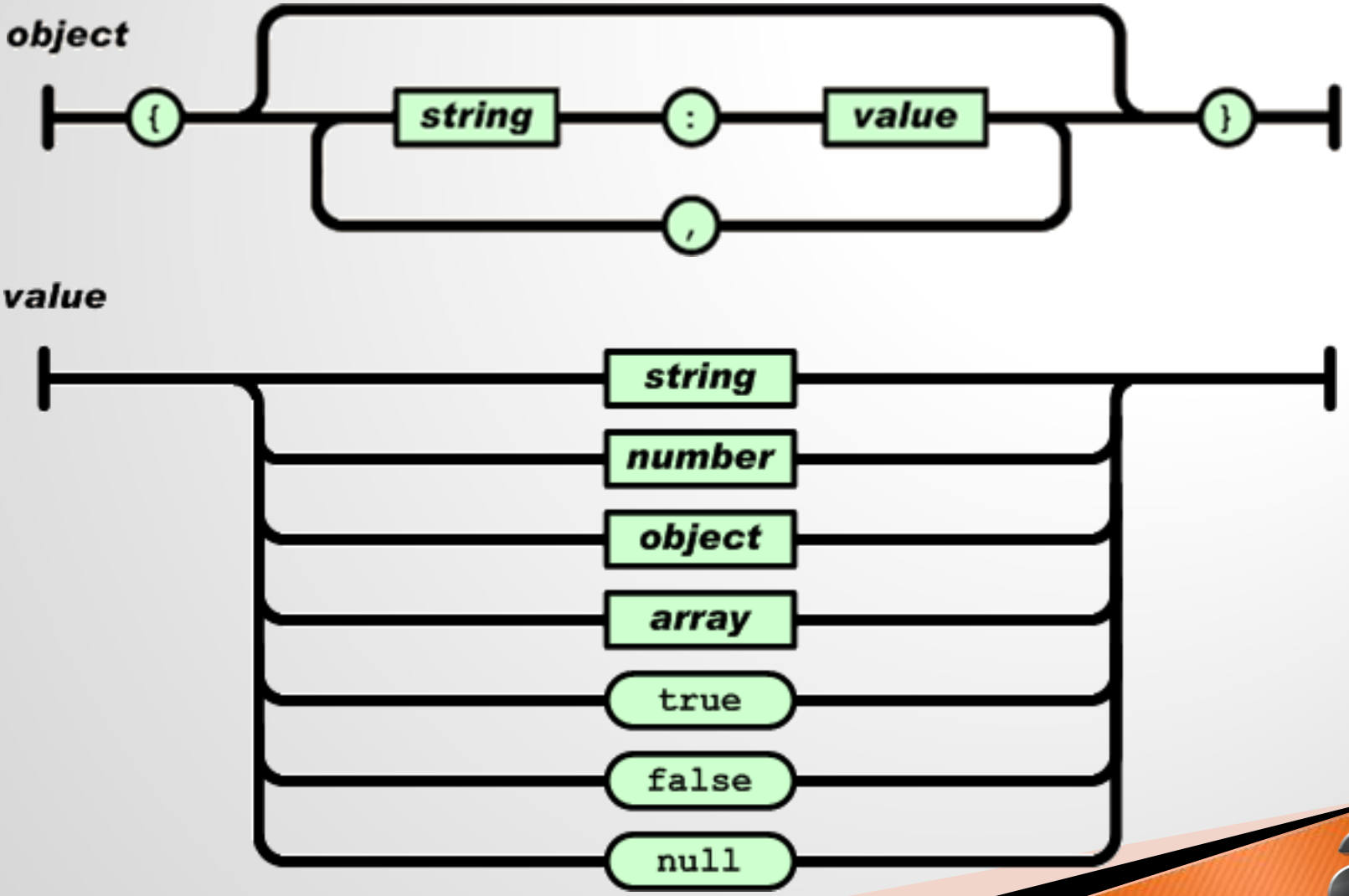
```
{
        “nome”:“joao”,
        “idade”:22,
        “salario”:2342.11,
        “interesses”:[
                “musica”,
                ”arte”
        ],
        ”dependentes”:[
                {“nome”:“camila”,“idade”:6},
                {“nome”:“joana”, “idade”:21}
        ]
}
```

JSON (Javascript Object Notation)

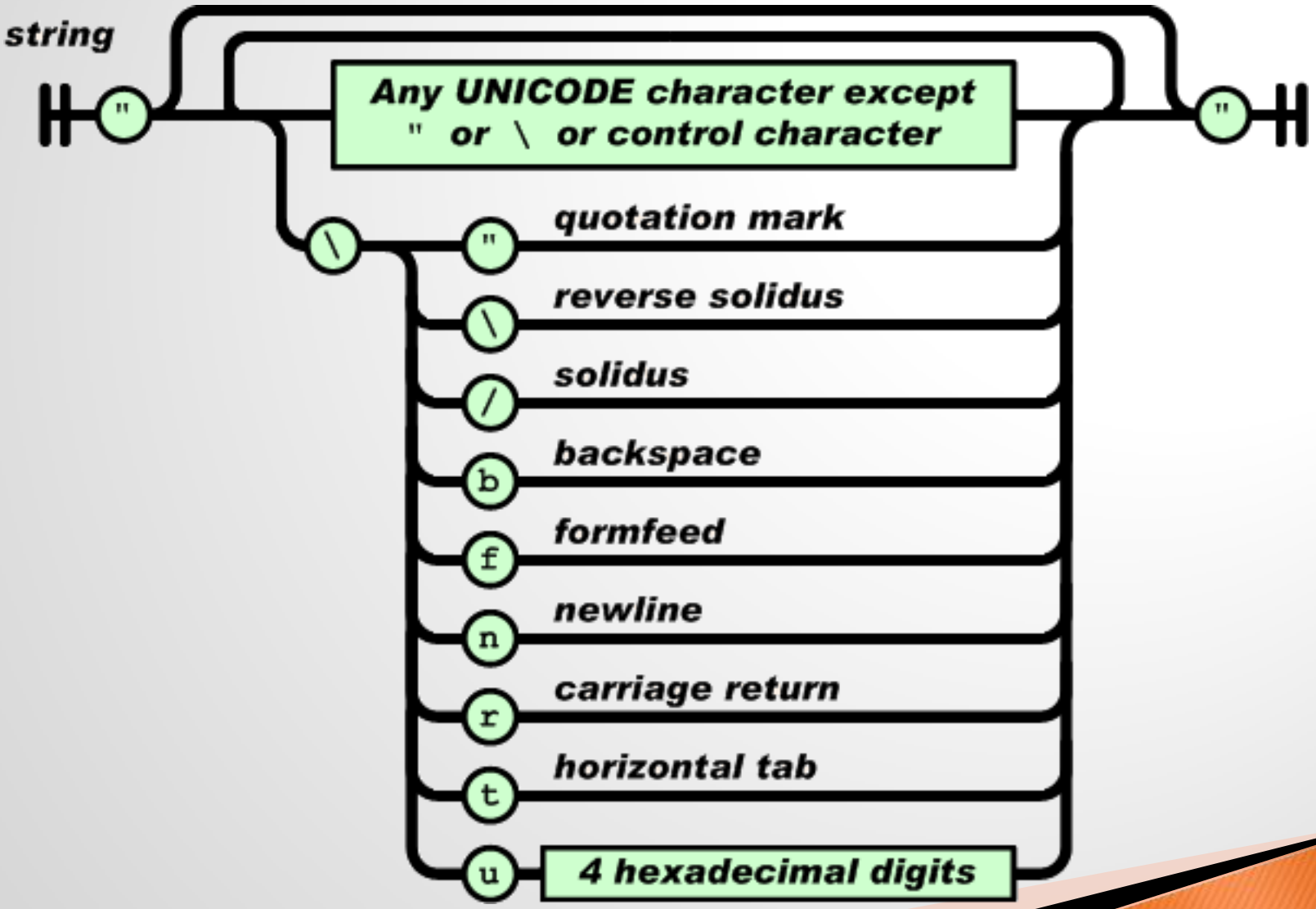
- Tipos de Dados Representados

  - String/Literal
  - Número
  - Vetor (array)
  - Objeto
  - true (verdadeiro)
  - false (falso)
  - null

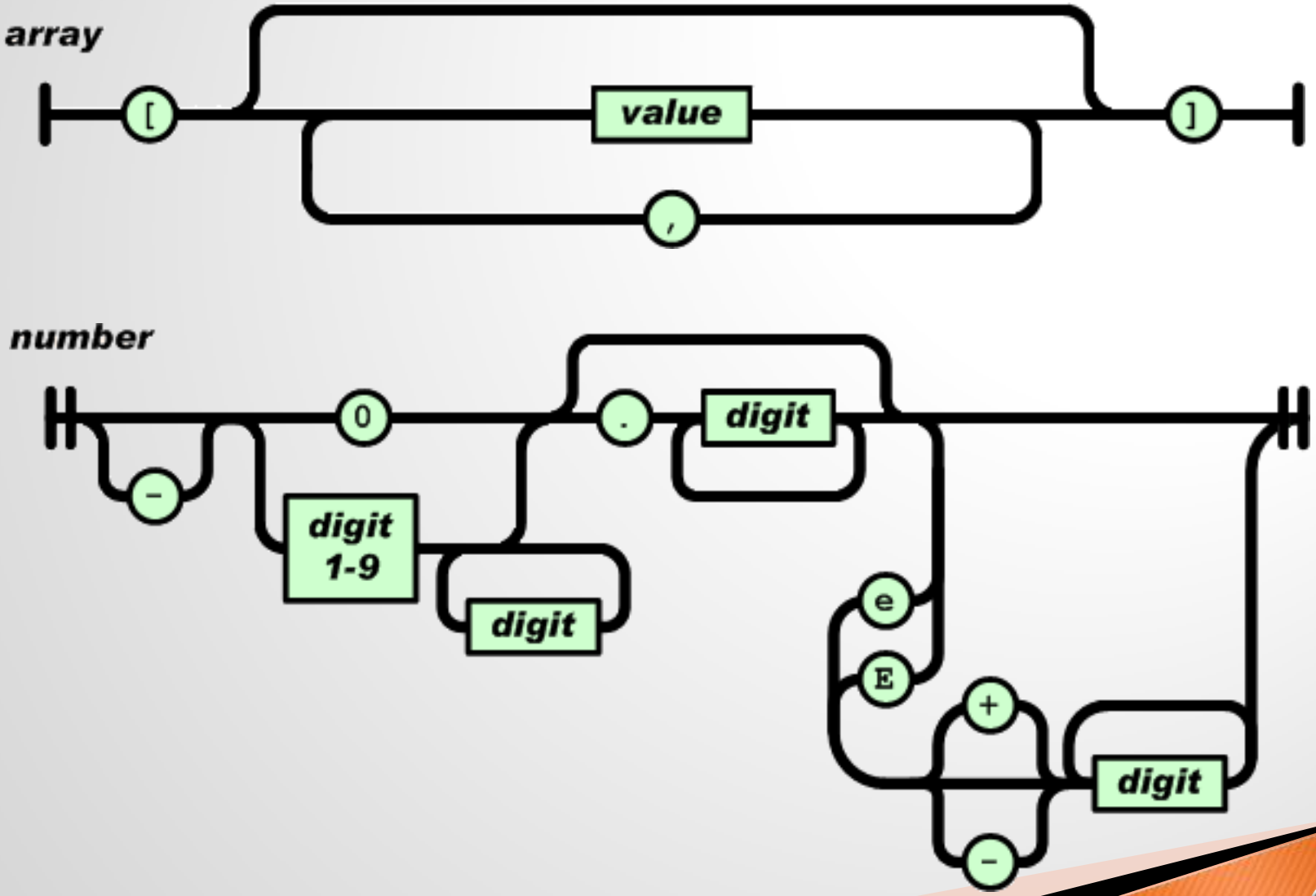
JSON (Javascript Object Notation)



JSON (Javascript Object Notation)



JSON (Javascript Object Notation)



Tecnologia em Gestão

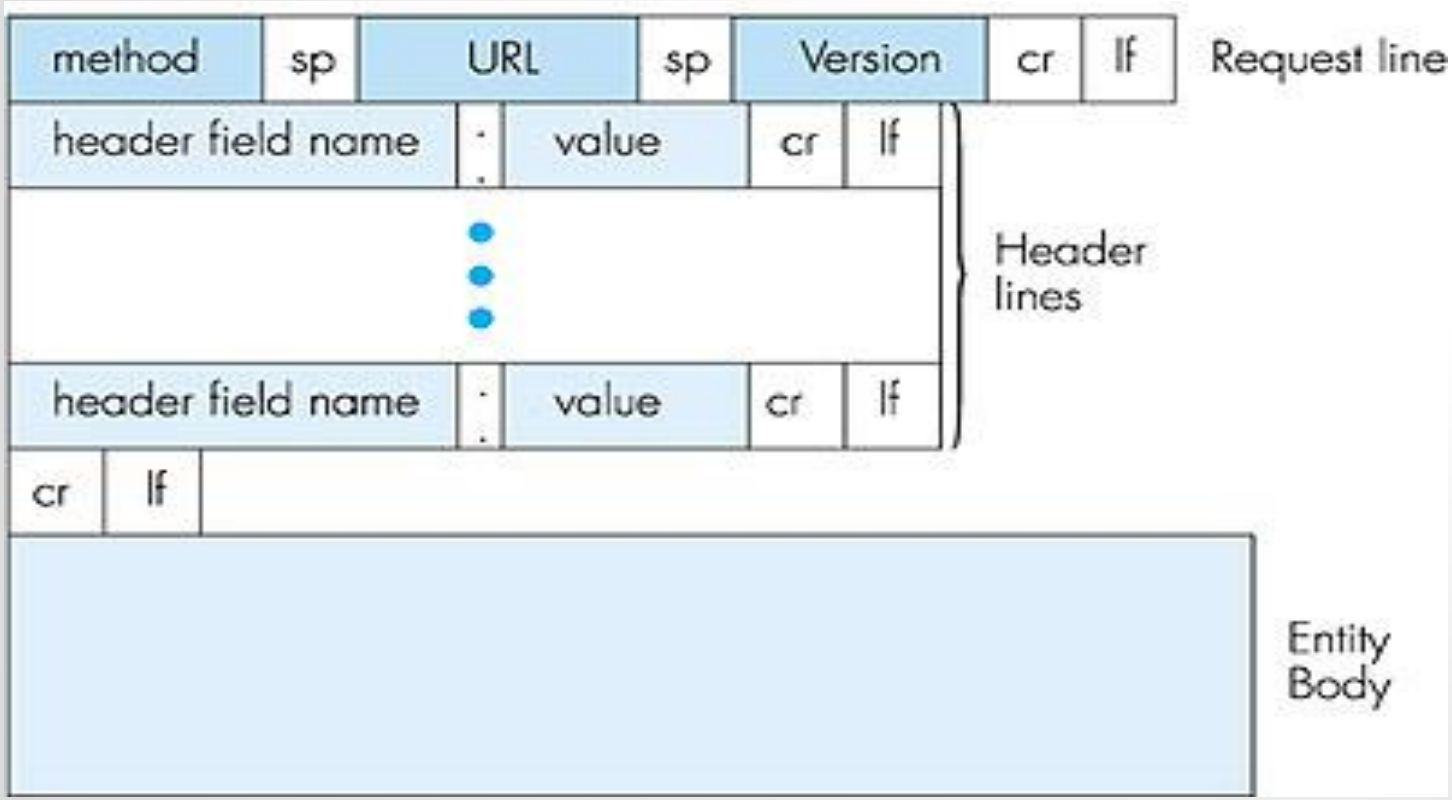
Comunicação por Http

Usamos o protocolo http para realizar a comunicação entre o java e o angular.

Enviamos uma mensagem pela rede (Request) e recebemos uma resposta (Response), ambas em formato de texto.

Comunicação por Http

Request (Requisição)



Comunicação por Http

Metodos Http

GET – Obter informações
POST – Enviar Informações
PUT – Alterar Informações
DELETE – Remover
HEAD – Solicitar Ações
OPTIONS – Solicitar Informações

Comunicação por Http

Request (Requisição)

```
POST /foo/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.5)
Gecko/20091102 Firefox/3.5.5 (.NET CLR 3.5.30729)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Content-Type: application/x-www-form-urlencoded
Content-Length: 43

first_name=John&last_name=Doe&action=Submit
```

Comunicação por Http

Request (Requisição)

```
POST /blog/posts HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.5)
Gecko/20091102 Firefox/3.5.5 (.NET CLR 3.5.30729)
Accept: application/json
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Content-Type: application/json
Content-Length: 57

{"title":"Hello World!","body":"This is my first post!"}
```

Comunicação por Http

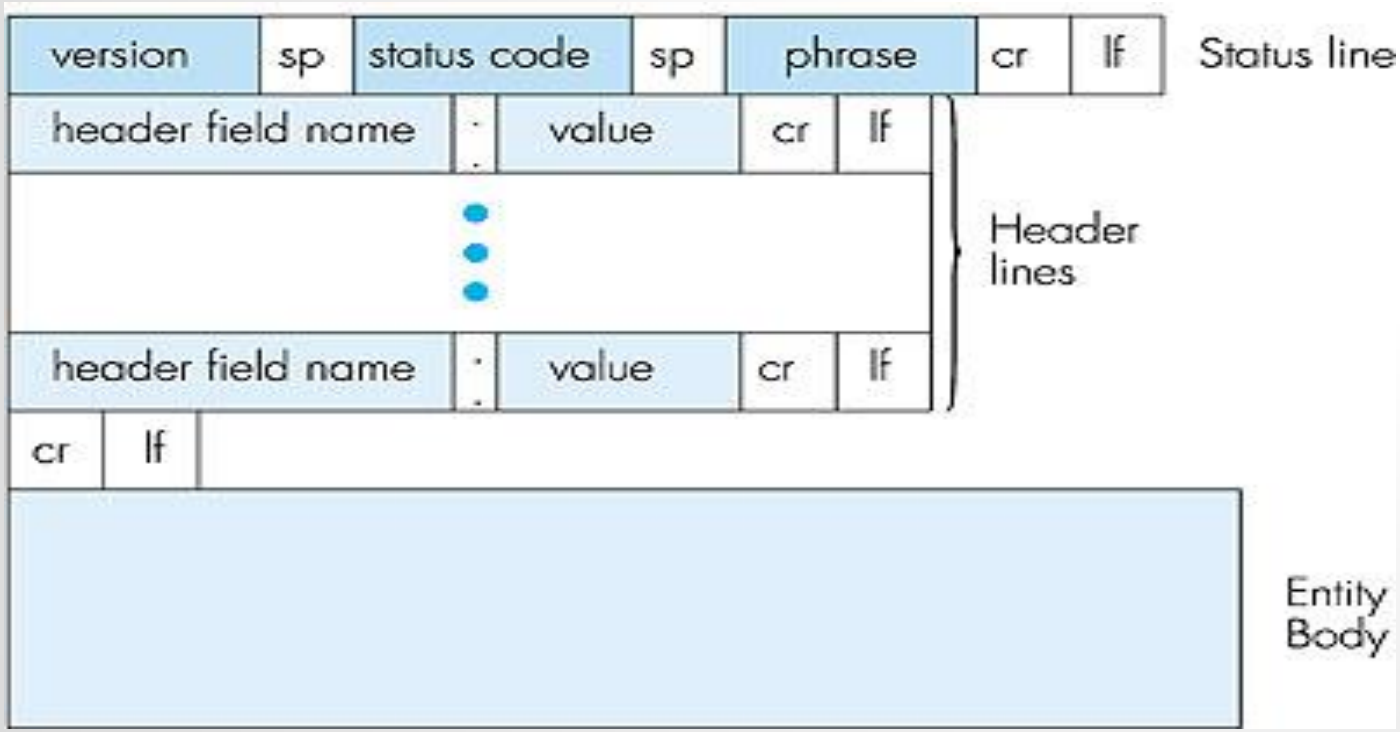
Request (Requisição)

```
POST /cgi-bin/qtest HTTP/1.1
Host: aram
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.10) Gecko/2009042316 Firefox/3.0.10
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Referer: http://aram/~martind/banner.htm
Content-Type: multipart/form-data; boundary=-287032381131322
Content-Length: 582

---287032381131322
Content-Disposition: form-data; name="datafile1"; filename="r.gif"
Content-Type: image/gif

GIF87a.....,.....D..;
---287032381131322
Content-Disposition: form-data; name="datafile2"; filename="g.gif"
Content-Type: image/gif

GIF87a.....,.....D..;
---287032381131322
Content-Disposition: form-data; name="datafile3"; filename="b.gif"
Content-Type: image/gif

GIF87a.....,.....D..;
---287032381131322--
```

Tecnologia em Gestão

Comunicação por Http

Response (Resposta)



Comunicação por Http

Códigos de Status

- 1xx Informativa
- 2xx Sucesso
- 3xx Redirecionamento
- 4xx Erro de cliente
- 5xx outros erros

Comunicação por Http

1xx Informativa

100 Continuar
101 Mudando protocolos
122 Pedido-URI muito longo

Comunicação por Http

2xx Sucesso

200 OK
201 Criado
202 Aceito
203 não-autorizado (desde HTTP/1.1)
204 Nenhum conteúdo
205 Reset
206 Conteúdo parcial
207-Status Multi (WebDAV) (RFC 4918)

Comunicação por Http

3xx Redirecionamento

300 Múltipla escolha
301 Movido
302 Encontrado
304 Não modificado
305 Use Proxy (desde HTTP/1.1)
306 Proxy Switch
307 Redirecionamento temporário (desde HTTP/1.1)

Comunicação por Http

4xx Erro de cliente

400 Requisição inválida
401 Não autorizado
402 Pagamento necessário
403 Proibido
404 Não encontrado
405 Método não permitido
406 Não Aceitável
407 Autenticação de proxy necessária
408 Tempo de requisição esgotou (Timeout)
409 Conflito
410 Gone
411 comprimento necessário

Comunicação por Http

4xx Erro de cliente

412 Pré-condição falhou
413 Entidade de solicitação muito grande
414 Pedido-URI Too Long
415 Tipo de mídia não suportado
416 Solicitada de Faixa Não Satisfatória
417 Falha na expectativa
418 Eu sou um bule de chá
422 Entidade improcessável (WebDAV) (RFC 4918)
423 Fechado (WebDAV) (RFC 4918)
424 Falha de Dependência (WebDAV) (RFC 4918)
425 coleção não ordenada (RFC 3648)
426 Upgrade Obrigatório (RFC 2817)
450 bloqueados pelo Controle de Pais do Windows
499 cliente fechou Pedido (utilizado em ERPs/VPSA)

Comunicação por Http

5xx outros erros

500 Erro interno do servidor (Internal Server Error)
501 Não implementado (Not implemented)
502 Bad Gateway
503 Serviço indisponível (Service Unavailable)
504 Gateway Time-Out
505 HTTP Version not supported

Comunicação por Http

Response (Resposta)

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 65
Connection: close

{"id":"1","title":"Hello World!","body":"This is my first post!"}
```

Content-Type

Normalmente o método post envia dados no formato **application/x-www-form-urlencoded**

- Control names e valores são codificados. Espaços são substituidos por ’+’, e caracteres reservados são substituidos, de acordo com a [RFC1738], seção 2.2: Caracteres não-alphanumericos são trocados por ’%HH’, um sinal de percentagem e dois dígitos hexadecimais que representam o código ASCII do caractere. Quebras de linha são representadas por "CR LF" par (i.e., ’%0D%0A’).
- Os control names/valores são listados na ordem que aparecem no documento. A propriedade “name” é separada do valor por ’=‘ e os pares nome/valor são separadas entre sí por ‘&’.

Content-Type

Outro formato muito usado é **multipart/form-data**

- Uma mensagem "multipart/form-data" contem várias partes, Cada uma representando uma successful control. As partes são enviadas na mesma ordem que aparecem do documento. Part boundarie não pode aparecer em nenhum dos dados;
- Como ocorre com todos os multipart MIME types, cada parte tem um cabeçalho opcional "Content-Type" cujo default é "text/plain". O cabelhado deve ser fornecido com "Content-Type" acompanhado do parêmetro "charset".
- Para cada parte é esperado que contenha:
  - Atributo "Content-Disposition" cujo valor é "form-data".
  - Atributo “name” especificando o control name do controle correspondente. Control names originalmente codificadod em não-ASCII character sets podem ser codificados usando o método descrito pela [RFC2045].

multipart/form-data

```
POST /cgi-bin/qtest HTTP/1.1
Host: aram
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.10) Gecko/2009042316 Firefox/3.0.10
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Referer: http://aram/~martind/banner.htm
Content-Type: multipart/form-data; boundary=-287032381131322
Content-Length: 582

---287032381131322
Content-Disposition: form-data; name="submit-name"

Larry
---287032381131322
Content-Disposition: form-data; name="datafile1"; filename="g.gif"
Content-Type: image/gif

GIF87a.....,.....D..;
---287032381131322
Content-Disposition: form-data; name="datafile1"; filename="b.gif"
Content-Type: image/gif

GIF87a.....,.....D..;
---287032381131322--
```

Restful

- JAX-RS (Java API for Restful Web Services)
- Exemplos

  www.oracle.com/technetwork/articles/java/jaxrs20-1929352.html
- Especificação (API)

  https://jax-rs-spec.java.net/nonav/2.0-rev-a/apidocs/index.html

Restful – Requisitos

- A Uniforme,constrained interface
- Representation-Oriented
- Communicate Statelessly
- Hypermedia As The Engine Of Application State (HATEOAS)

Restful

- **A Uniforme,constrained interface**

Use os métodos que já existem para http

- GET – Obter um item(passando id) ou uma lista deles
- POST – Criar um novo item
- PUT – Atualizar um item (passando id)
- DELETE – remover um item
- HEAD – Executar uma ação para obter um cabeçalho e http response code
- OPTIONS – obter informações do Webservice

Restful

- **Representation-Oriented**

Diferentes plataformas precisam de diferentes formatos.

  - Browser usa html
  - Javascript usa json
  - Java usa XML etc.

Restful

- **Communicate Statelessly**

Aplicações sem estado (sem beans de sessão) são mais fáceis de escalar.

Restful

- **Hypermedia As The Engine Of Application State (HATEOAS)**

Hipermídia como motor do estado do aplicativo.

Use Links para realizar ações de vão modificar o estado do seu aplicativo.

Restful

- **Hypermedia As The Engine Of Application State (HATEOAS)**

Hipermídia como motor do estado do aplicativo.

Use Links para realizar ações de vão modificar o estado do seu aplicativo.

Restful

- Exemplo de Restful

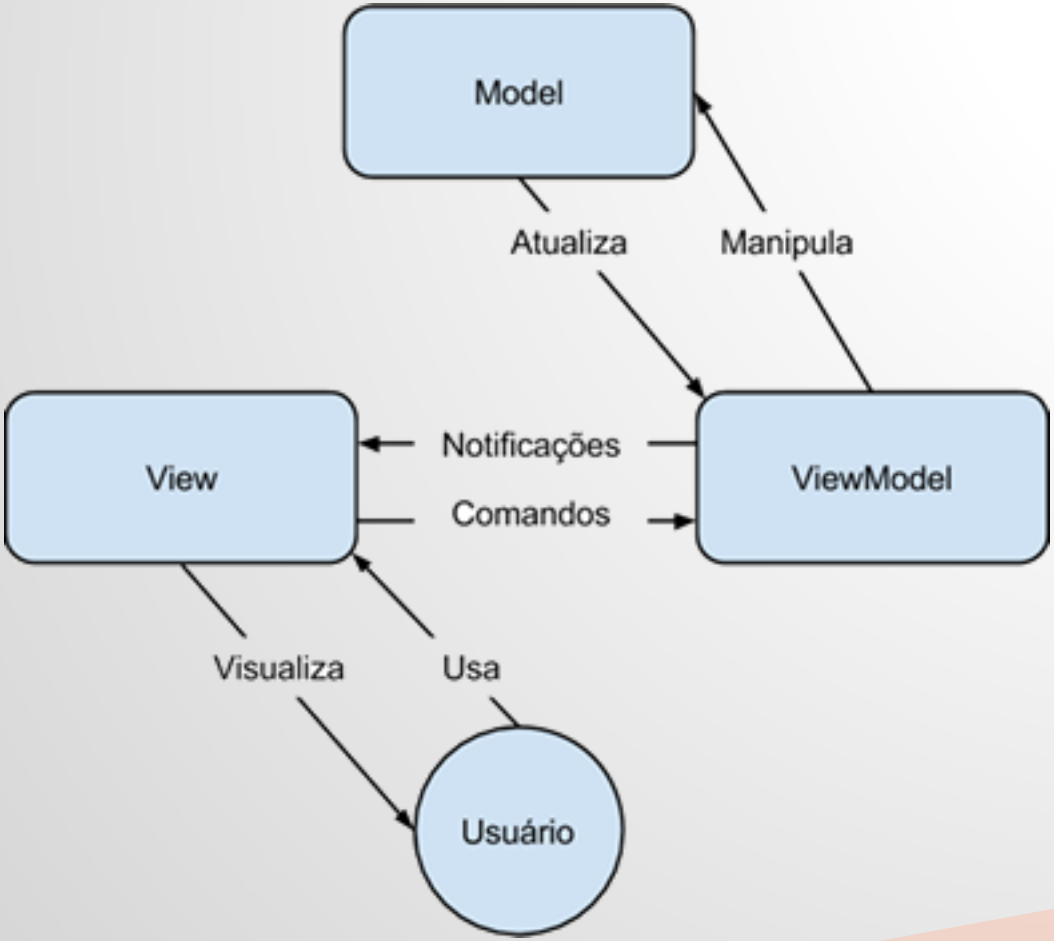
Ação	Metodo	URL	Envia	Recebe
Listar	GET	/prod	-	Lista os Produtos
Novo	POST	/prod	Produto	Produto Salvo
Carregar	GET	/prod/{id}	-	Produto
Atualizar	PUT	/prod	Produto	Produto Atualizado
remover	DELETE	/prod/{id}	Produto	-

SPA (Single Page Application)

- Tudo numa página soh! Como o desktop!
- Carga de html,css e javascript por demanda.
- Exemplos:
  - AngularJS
  - Ember.JS
  - ExtJS
  - React
  - Beckbone

SPA (Single Page Application)

MVVM Model-View-ViewModel



Tecnologia em Gestão

Fim

Obrigado pela participação.

Tecnologia em Gestão