



DataSnap na Prática: De zero a 2 milhões de acessos/dia

#26

Enoque Canuto

Da sala de reunião à vida real

Embarcadero Conference 2019

Enoque Canuto

- Bacharel em Sistemas de Informação
- Especialista em Gerenciamento de Projetos
- Desenvolvedor Delphi desde 2005
- Certificado Delphi Developer
- Instrutor Oficial Embarcadero
- Coordenador de Desenvolvimento na Bauhaus Sistemas
- Consultor Técnico na WK Technology



embarcadero®

Agenda

- O Projeto “VirtualBank”
- Estrutura da Aplicação
- Tecnologias e Integrações
- Dados de Utilização
- Utilização do DataSnap
- Dificuldades e Soluções

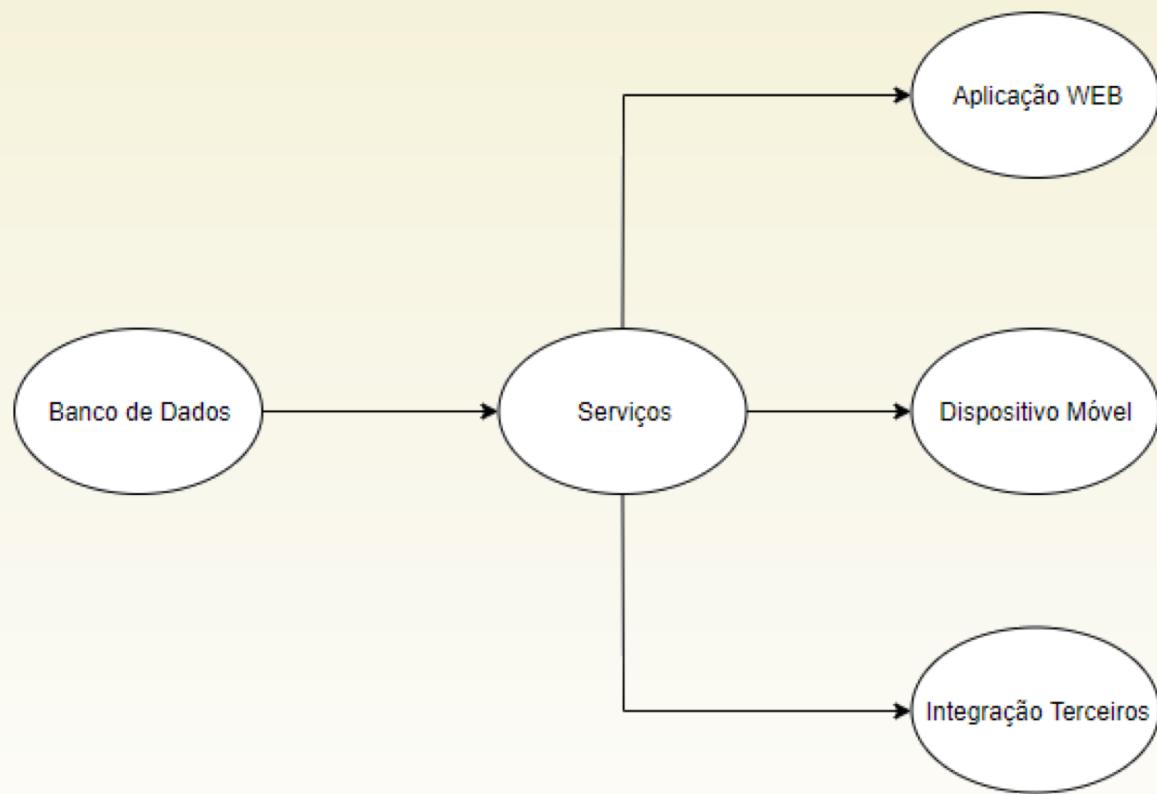
O Projeto “VirtualBank”

- Banco Virtual
- 100% On-Line
- Contas Bancárias
- Transferências
- Boletos
- Integrações

Por onde começar



Estrutura Básica



Tecnologias e Interações



PostgreSQL



ubuntu

{ JSON }

ANGULARJS

APACHE



zenvia



Amazon
RDS



Amazon
EC2

Mailgrid

RX

{ REST }



bradesco



Fast Reports

aws



CAIXA



Chilkat Software

embarcadero®

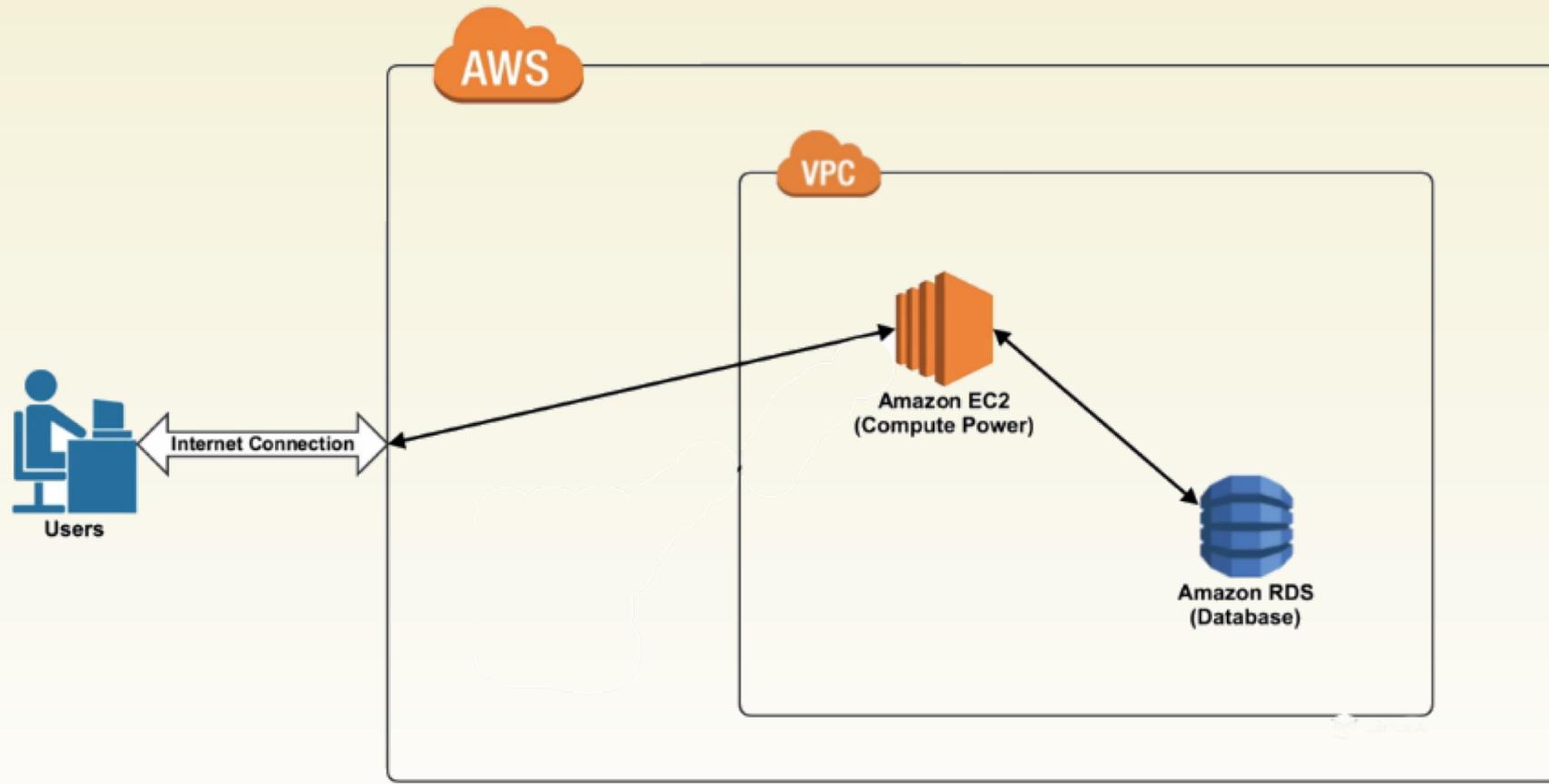
Tecnologias e Integrações

- Banco de Dados
 - PostgreSQL
- BackEnd
 - Delphi
 - DataSnap
 - TMS Aurelius
 - FastReport
 - Chilkat
 - Linux
 - Windows
 - Apache
 - REST
 - JSON
 - AWS
 - EC2
 - RDS

Tecnologias e Integrações

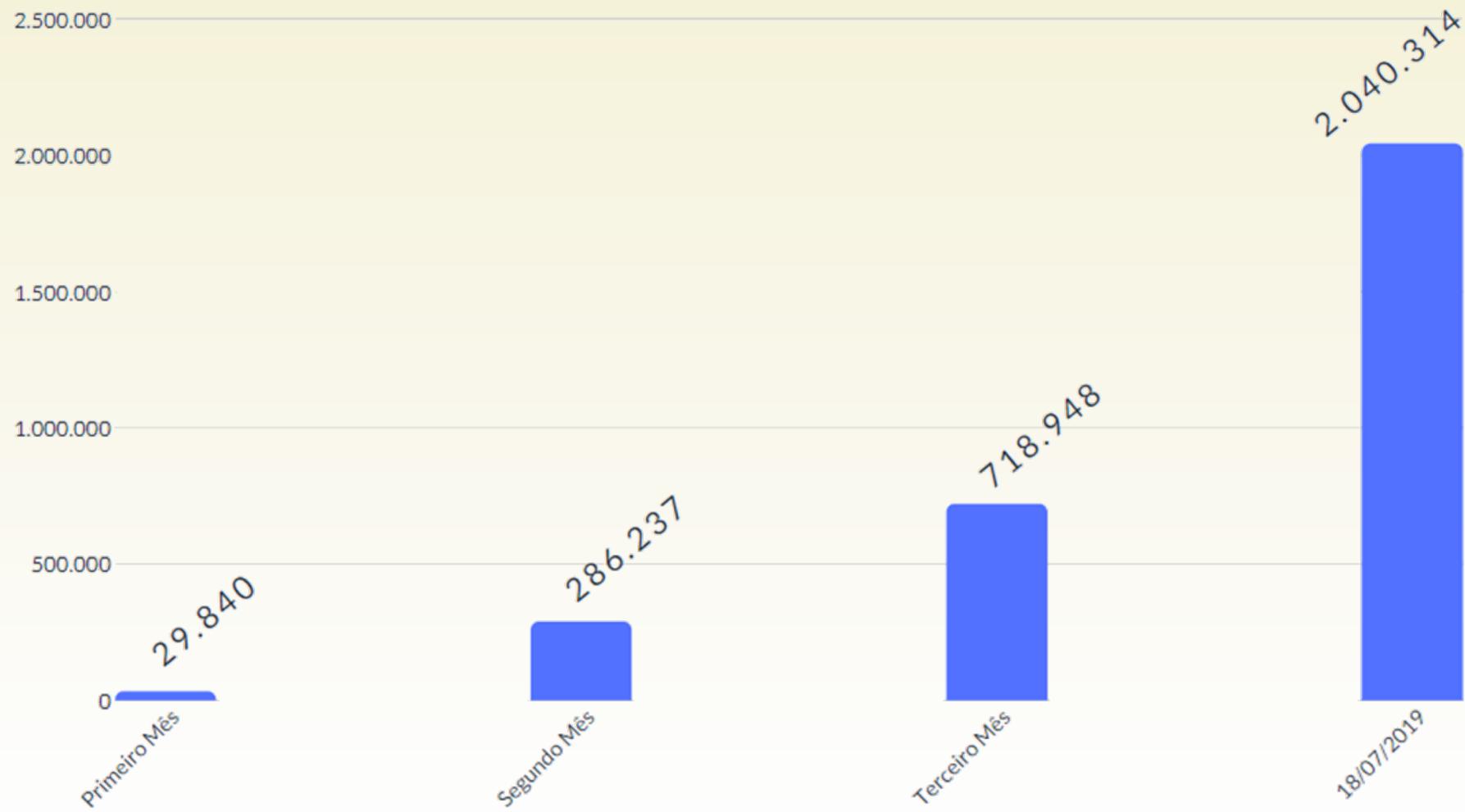
- FrontEnd
 - AngularJS
- Integrações
 - BB
 - Santander
 - CEF
 - Bradesco
 - Software Express
 - MailGrid
 - Zenvia (SMS)

Tecnologias e Integrações

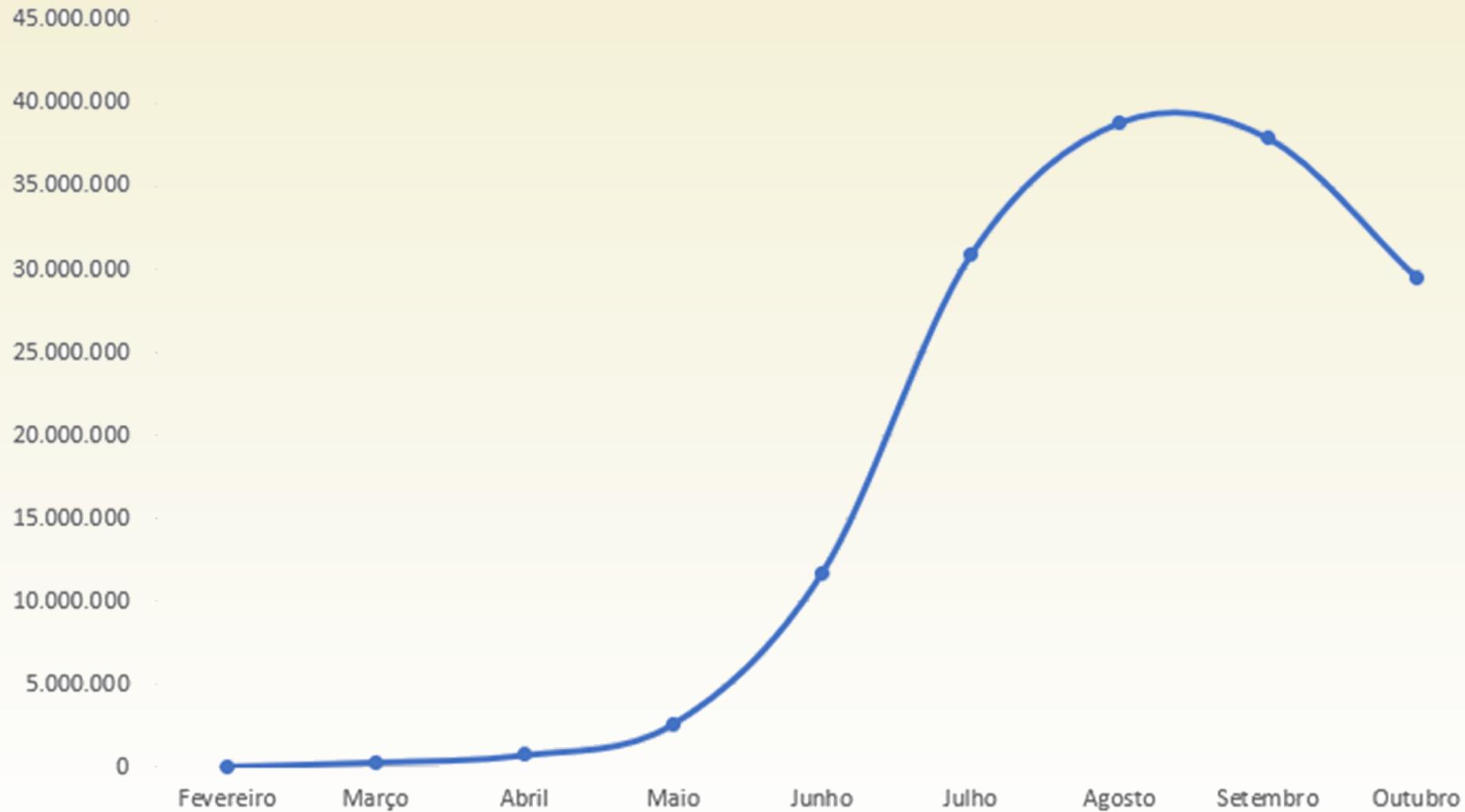


Dados de Utilização

- Início: **20/02/2019**



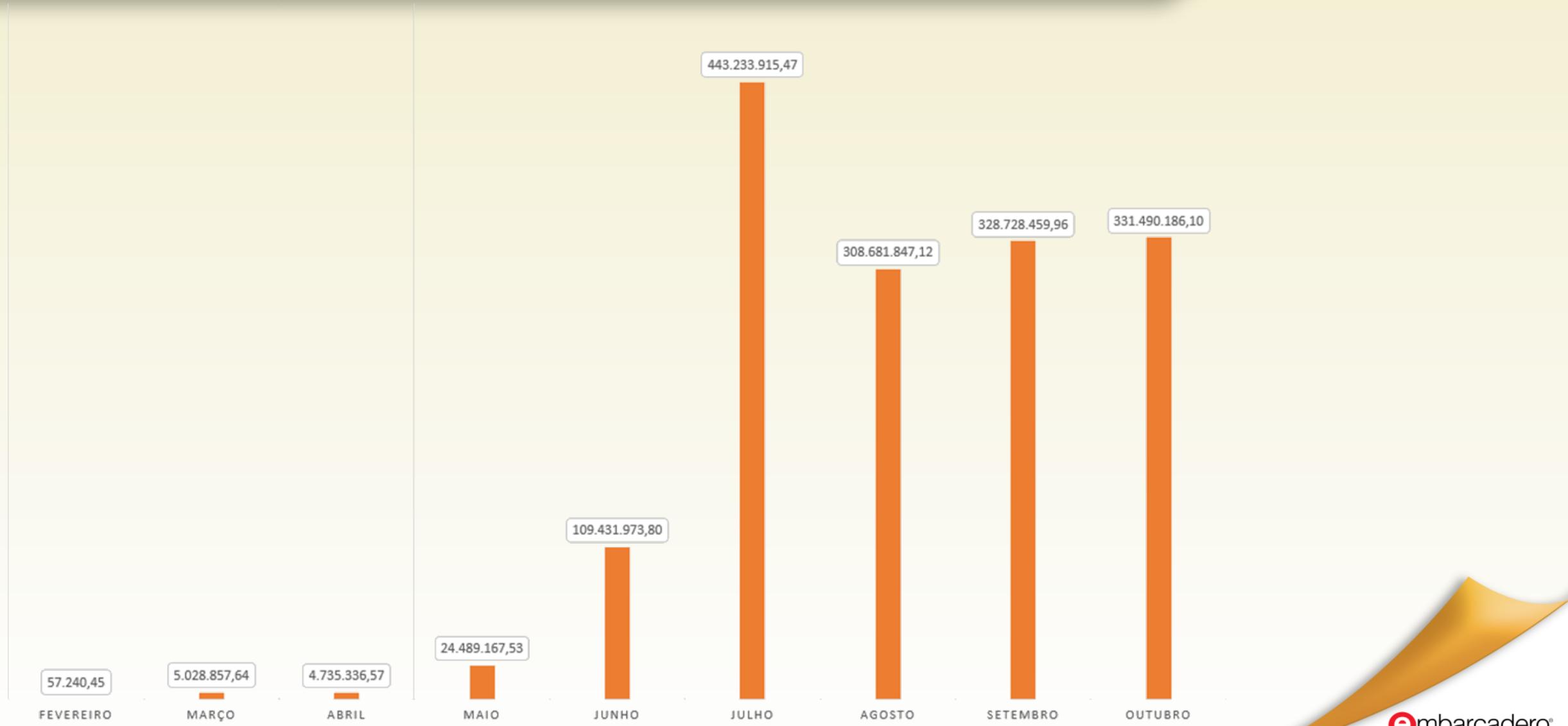
Dados de Utilização



Dados de Utilização

- Até 20/10/2019: **152.414.233 Requisições**
- $152.414.233 / 8 \text{ Meses} = \mathbf{19.051.779 \text{ a cada MÊS}}$
- $19.051.779 / 30 \text{ Dias} = \mathbf{635.059 \text{ a cada DIA}}$
- $635.059 / 24 \text{ Horas} = \mathbf{26.460 \text{ a cada Hora}}$
- $26.460 / 60 \text{ Minutos} = \mathbf{441 \text{ a cada MINUTO}}$
- $441 / 60 = \mathbf{7 \text{ a cada SEGUNDO}}$

Dados de Movimentações Financeiras



Custo da Infra

Saldo acumulado no mês atual para outubro de 2019

\$1,074.50

\$1.8K

\$1.58K

\$1.35K

\$1.07K

\$900

EC2

\$871.56

RDS

\$165.86

DataTransfer

\$36.05

Route53

\$1.03

Outros serviços

\$0.00

Imposto

\$0.00

Total

\$1,074.50

\$450

\$0

Último mês
(setembro de 2019)

Acumulados no mês
(outubro de 2019)

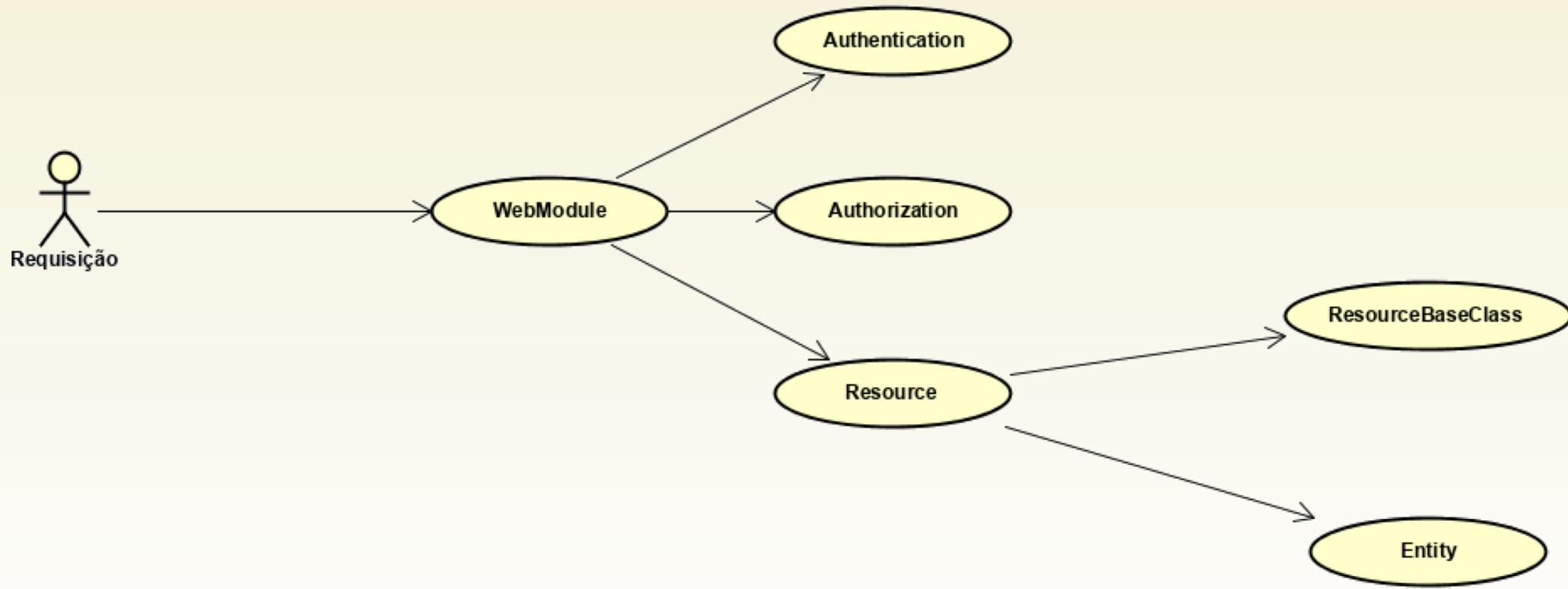
Previstos
(outubro de 2019)

Legend:

Utilização do DataSnap

- Delphi
- FireDAC
- TMS Aurelius
- FastReport
- Chilkat

Utilização do DataSnap



Entity

```
1  unit Server.Application.App.PessoasEntity;
. |
. interface
. |
. uses
.   Aurelius.Types.Nullable, Aurelius.Mapping.Attributes, Aurelius.Types.Proxy, Aurelius.Types.Blob,
.   Server.Application.App.CidadesEntity, Server.Application.App.PaisesEntity;
. |
. type
10  [Entity, Automapping]
.   [Table('pessoas', 'comum')]
.   [Sequence('comum.pessoas_id_pessoas_seq')]
.   [Id('ID', TIdGenerator.IdentityOrSequence)]
. |
.   TPessoasEntity = class
.     public
.       [Column('id_pessoas', [TColumnProp.Unique, TColumnProp.Required, TColumnProp.NoUpdate])]
.       ID: Integer;
. |
.       [Column('nm_pessoa', [TColumnProp.Required], 100)]
.       NmPessoa: String;
. |
.       [Column('nm_apelido', [], 100)]
.       NmApelido: Nullable<String>;
. |
.       [Column('tp_pessoa', [TColumnProp.Required], 1)]
.       TpPessoa: String;
. |
.       [Column('nr_cpf_cnpj', [TColumnProp.Required], 14)]
.       NrCpfCnpj: String;
30
.       [Column('nr_rg_ie', [], 20)]
.       NrRgIe: Nullable<String>;
. |
.       [Column('dt_nascimento_abertura', [])]
.       DtNascimentoAbertura: Nullable<TDate>;
. |
.       [Column('nr_cep', [], 10)]
.       NrCep: Nullable<String>;
. |
.       [Column('tp_logradouro', [], 20)]
.       TpLogradouro: Nullable<String>;
```

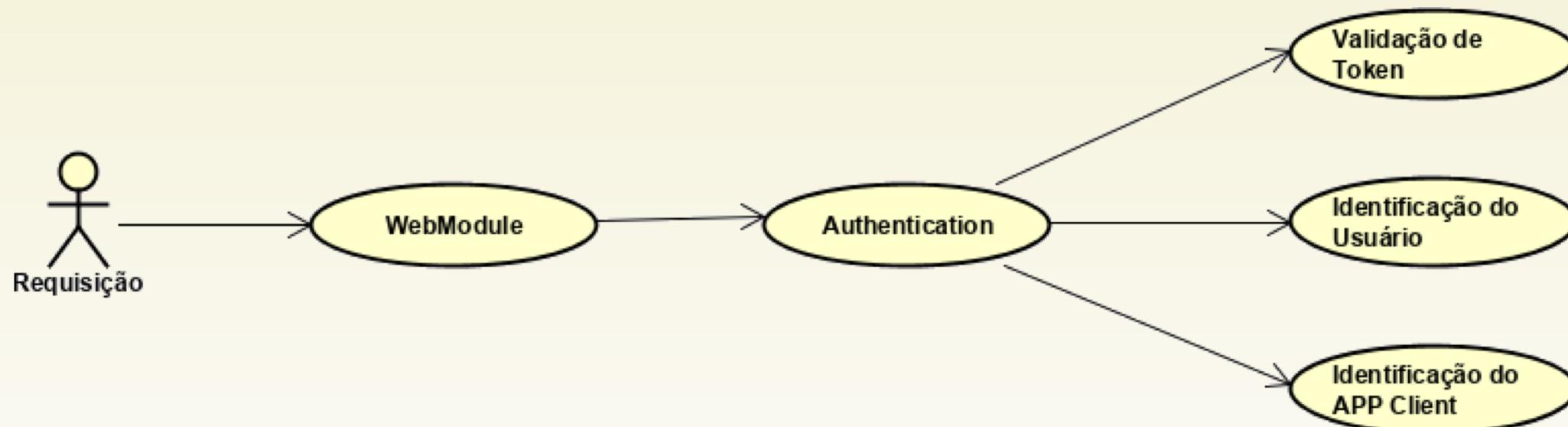
Resource

```
- □ unit Server.Application.App.PaisesResource;
- |
- □ interface
- |
-   uses
-     System.JSON, Server.Application.App.PaisesEntity, Lib.Server.ResourceBaseClass;
-   type
-     TPaisesResource = class(TResourceBaseClass)
10    public
-      function Accept(Dados: TJSONObject): TJSONObject; override;
-      function Cancel(Id: Integer): TJSONObject; override;
-      function Get(Id: Integer): TJSONObject; override;
-      function Update(Id: Integer; Dados: TJSONObject): TJSONObject; override;
-    end;
- |
- □ implementation
- |
-   { TPaisesResource }
20
-   □ function TPaisesResource.Accept(Dados: TJSONObject): TJSONObject;
-     begin
-       Result := ProcessAccept<TPaisesEntity>(Dados);
-     end;
-   |
-   □ function TPaisesResource.Cancel(Id: Integer): TJSONObject;
-     begin
-       Result := ProcessCancel<TPaisesEntity>(Id);
-     end;
30
-   □ function TPaisesResource.Get(Id: Integer): TJSONObject;
-     begin
-       Result := ProcessGet<TPaisesEntity>(Id);
-     end;
-   |
-   □ function TPaisesResource.Update(Id: Integer; Dados: TJSONObject): TJSONObject;
-     begin
-       Result := ProcessUpdate<TPaisesEntity>(Id, Dados);
-     end;
40
-   end.
```

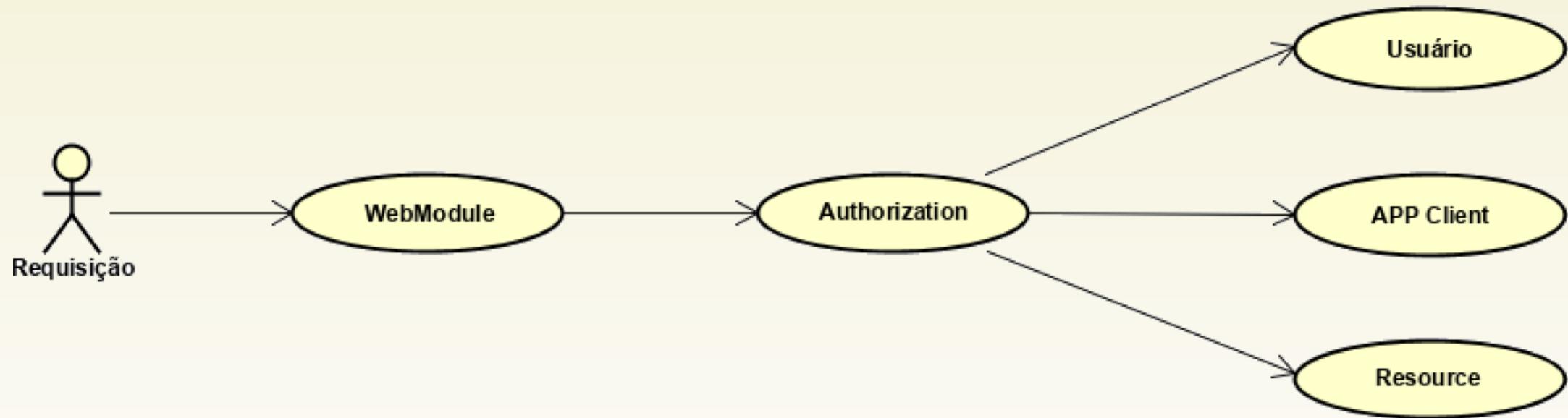
ResourceBaseClass

```
1 unit Lib.Server.ResourceBaseClass;
.
.
.
- uses ...
.
.
.
type
  TMethodKind = (TmGet, TmPost, TmPut, TmDelete);
.
.
10 TJoin = class
  private
    FField: String;
    FAlias: String;
  public
    property Field: String read FField write FField;
    property Alias: String read FAlias write FAlias;
  end;
.
.
TResourceBaseClass = class
20 strict private
  ...
  private
    ...
  protected
    FManager: TObjectManager;
    FConnection: TDConnection;
    function GetPerPageParam: Integer;
    function ProcessAccept<T: Class>(Dados: TJSONObject; SendResponse: Boolean = True): TJSONObject;
    function ProcessCancel<T: Class>(Id: Integer): TJSONObject;
    function ProcessGet<T: Class>(Id: Integer; Filter: TCustomCriterion = nil; Joins: TObjectList<TJoin> = nil): TJSONObject;
    function ProcessUpdate<T: Class>(Id: Integer; Dados: TJSONObject; SendResponse: Boolean = True; ValidateID: Boolean = True): TJSONObject;
  public
    constructor Create;
    destructor Destroy; override;
    function Get(Id: Integer): TJSONObject; virtual; abstract;
    function Accept(Dados: TJSONObject): TJSONObject; virtual; abstract;
    function Update(Id: Integer; Dados: TJSONObject): TJSONObject; virtual; abstract;
    function Cancel(Id: Integer): TJSONObject; virtual; abstract;
    function ToJSON(Obj: TObject): TJsonValue;
  end;
.
.
.
40
```

Authentication



Authorization



Mensagens de Retorno

- JSON Padronizado
- HTTP Code
- Paginação
- Mensagens de Erro e de Alerta

```
1 {  
2   "Token": "A81868F824DD31F61D073F0475F25A425239131BF13BDC2248B36D91DA461055CA9E868C9D46D935350A5  
3     8FB55F82DFB5FF253D40570A178B163C0949995FD472E48E123E17FA5958ED56CA5A9FA1CD624CAA0BE90BCBAD1C7  
4     7DA848DA08171F75EC78E04A3555C380FC7CEE7EB5ABB88382C190A3ED5EFE778C4CF329DF22222F071213283A59E  
5     72B0C52E275DF5BD37690808F8FE903",  
6   "Pages": {  
7     "RecordCount": "5571",  
8     "RecordsPerPage": "50",  
9     "PageCount": "112",  
10    "CurrentPage": "1"  
11  },  
12  "Data": [  
13    {  
14      "ID": 1100015,  
15        "NmCidade": "Alta Floresta D'Oeste",  
16        "SgUf": "RO"  
17    },  
18    {  
19      "ID": 1100023,  
20        "NmCidade": "Ariquemes",  
21        "SgUf": "RO"  
22    },  
23    {  
24      "ID": 1100031,  
25        "NmCidade": "Cabixi",  
26        "SgUf": "RO"  
27    }  
28  ]  
29}  
30}
```

Mensagens de Retorno

Pessoas - Preencha o formulário abaixo.

Tipo de Pessoa *

Pessoa Física

Nome da Pessoa *

Enoque Canuto

Apelido *

Enoque

Nº CPF/CNPJ *

012.345.678-90



Nº RG ou I.E

123456



Nº do CEP

88800000

País do Endereço *

BRASIL



Tipo Logradouro

Rua

Nome Logradouro

João da Silva



Nº Imóvel

10

Bairro

Cidadezinha



Complemento

Dificuldades e Soluções

- Amazon
 - Elastic Container Service
 - Elastic Compute Cloud
 - RDS (Relational Database Service)
- Criptografia
 - Chilkat Software
- Apache
 - Windows



Da sala de reunião à vida real

#26

embarcadero®

Obrigado



enoquecanuto@gmail.com



@enoquecanuto



@enoquecanuto



@enoquecanuto

Embarcadero Conference 2019