



**AVISTA S/A ADMINISTRADORA DE CARTOES DE CRÉDITO
NGI – NÚCLEO DE GESTÃO DA INFORMAÇÃO**

DOCUMENTAÇÃO DO WEBSERVICE PARA O APLICATIVO “FAZER CARTÃO”

Vitória, 15 Outubro de 2014.

HISTÓRICO DE REVISÕES

Versão	Autores	Descrição da Versão	Data
1.0	Rodrigo Tosta Trancoso	Versão Inicial	03/11/2014

VISÃO GERAL

Este documento visa apresentar o funcionamento do WebService para o aplicativo mobile “fazer cartão”. Serão demonstrados os métodos disponibilizados pelo WebService.

Vale ressaltar que os objetivos deste documento são voltados para o analista/programador e também para a empresa responsável pelo aplicativo. Dentre os objetivos, os principais são:

- Explicar a arquitetura em que o WebService foi desenvolvido;
- Apresentar a forma de consumir o mesmo;
- Apresentar a forma de usar os dados retornados pelo mesmo;

1 – Introdução	4
1.1 – JSON	4
1.1.1 – Objeto	4
1.1.2 – Array	5
2 – Módulos	6
2.1 – Sou Lojista	6
2.1.1 – Funcionalidades	6
2.1.1.1 – Login Unidade Credenciada	7
2.1.1.2 – Cadastro de Celular	8
2.1.1.3 – Consultar CPF	9
2.1.1.4 – Consultar CEP	11
2.1.1.5 – Consultar CEP Validador	12
2.1.1.6 – Consultar CEP Logradouro	13
2.1.1.7 – Consultar Escolaridade	14
2.1.1.8 – Consultar Estado Civil	15
2.1.1.9 – Aprovar	16
2.1.1.10 – Enviar Documentos	18
2.1.1.11 – Verificar Aprovação	19
2.1.1.12 – Vincular Cartão Temporário	20
2.1.1.13 – Consultar Estados	20
2.1.1.14 – Consultar Cidades	22
2.1.1.15 – Consultar Bairros	23
2.1.1.16 – Consultar Lojas	25
2.2 – Não Sou Cliente	26
Apêndice A	27

1 – Introdução

O WebService foi desenvolvido para ser consumido pelo aplicativo mobile “fazer cartão”. Para o desenvolvimento do mesmo, foi definido a utilização da tecnologia RESTFull, essa de tecnologia possibilita transformar uma aplicação web em um WebService, e possibilita tanto consumir quanto produzir dados na forma de XML e de JSON, a mesma também suporta todos os métodos de requisições HTTP (GET, POST, PUT, DELETE), no entanto foi definido a utilização do JSON como estrutura de tráfego e que todas as requisições utilizarão o método POST.

1.1 – JSON

JSON é um formato leve para transmissão de dados computacionais, é um subconjunto da notação de objeto de JavaScript, mas seu uso não requer JavaScript propriamente dito.

Um dos principais motivos pela escolha do JSON é o fato de ser muito mais fácil escrever, ler e analisar as estruturas montadas.

Em JSON, podemos apresentar os dados da seguinte forma:

- Objeto;
- Array;

1.1.1 – Objeto

Um objeto JSON é um conjunto desordenado de pares atributo/valor. Um objeto começa com “{” (chave de abertura) e termina com “}” (chave de fechamento). Cada nome é seguido por “:” (dois pontos) e os pares nome/valor são seguidos por “,” (vírgula).

No exemplo abaixo, temos um objeto em JSON que contém uma estrutura capaz de armazenar os dados de um determinado funcionário.

```
1 {  
2   "NOME": "RODRIGO",  
3   "SOBRENOME": "TOSTA TRANCOSO",  
4   "IDADE": "22 ANOS",  
5   "EMPRESA": "AVISTA",  
6   "CARGO": "PROGRAMADOR"  
7 }
```

Observe que a estrutura contém a chave de abertura, os dois pontos que separam o nome do atributo e seu valor, a vírgula para separar os conjuntos de atributo/valor e por fim, a chave de fechamento.

1.1.2 – Array

Um array é uma coleção de valores ou de objetos. O array começa com "[" (colchete de abertura) e termina com "]" (colchete de fechamento). Os seus valores ou objetos também são separados por "," (vírgula).

No exemplo abaixo, temos um array em JSON que contém uma estrutura capaz de armazenar uma lista de funcionários.

```
1 {  
2   "FUNCIONÁRIOS": [  
3     {  
4       "NOME": "RODRIGO",  
5       "SOBRENOME": "TRANCOSO",  
6       "IDADE": "22 ANOS",  
7       "EMPRESA": "AVISTA",  
8       "CARGO": "PROGRAMADOR"  
9     },  
10    {  
11      "NOME": "CICLANO",  
12      "SOBRENOME": "DE ALGUMA COISA",  
13      "IDADE": "27 ANOS",  
14      "EMPRESA": "AVISTA",  
15      "CARGO": "PROGRAMADOR"  
16    },  
17    {  
18      "NOME": "BELTRANO",  
19      "SOBRENOME": "DE OUTRA COISA",  
20      "IDADE": "31 ANOS",  
21      "EMPRESA": "AVISTA",  
22      "CARGO": "PROGRAMADOR"  
23    }  
24  ]  
25 }
```

Observe que a estrutura contém a chave de abertura, os dois pontos que separam o nome do atributo e seu valor, o colchete de abertura, a vírgula para separar os conjuntos de atributo/valor, o colchete de fechamento e por fim, a chave de fechamento.

2 – Módulos

O WebService contempla três módulos, os quais são nomeados como:

- Sou Lojista;
- Sou Cliente;
- Não Sou Cliente;

Descrevendo rapidamente esses módulos, podemos chegar as seguintes definições.

O módulo “Sou Lojista” é direcionado às lojas credenciadas ao Avista, para que os mesmos possam criar novas propostas de cartões de crédito.

O módulo “Sou Cliente” é direcionado às pessoas que já possuem o cartão avista e querem gerenciar seus dados e acessar diversas outras funcionalidades.

O módulo “Não Sou Cliente” é direcionado às pessoas que ainda não possuem um cartão de crédito Avista e desejam fazer o seu.

Cada módulo deste WebService contem funcionalidades específicas e cada uma com sua regra de negócio. Essas mesmas funcionalidades serão descritas dentro de cada módulo.

2.1 – Sou Lojista

Como dito anteriormente, o módulo “Sou Lojista” é para as lojas que são credenciadas ao Avista, dessa forma é possível às mesmas consumirem o WebService para criarem novas proposta de crédito através do aplicativo.

2.1.1 – Funcionalidades

As funcionalidades disponibilizadas pelo WebService para o acesso e consumo do aplicativo são:

- Login Unidade Credenciada;
- Cadastro de Celular;
- Consultar CPF;
- Consultar CEP;
- Consultar CEP Validador;
- Consultar CEP Por Logradouro;
- Consultar Escolaridades;
- Consultar Estado Civil;
- Aprovar;
- Enviar Documentos;
- Verificar Aprovação

- Vincular Cartão Temporário
- Ativar Blister
- Consultar Estados;
- Consultar Cidades;
- Consultar Bairros;
- Consultar Lojas;

Vale ressaltar que cada uma destas funcionalidades é acessada através de requisições HTTP, onde cada uma possui um endereço (URL) específico e único.

2.1.1.1 – Login Unidade Credenciada

A funcionalidade “login unidade credenciada” realiza a autenticação de uma unidade credenciada.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/login/unidade/credenciada/"
```

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o login e a senha da unidade credenciada, para que o mesmo possa ser autenticado em nossa base de dados. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para o login.

```
1 {  
2   "PARAM_LOGIN_UNIDADE_CREDENCIADA": "161392",  
3   "PARAM_SENHA_UNIDADE_CREDENCIADA": "960417"  
4 }
```

Quando a autenticação foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "DADOS": {
3     "PROMOTORES": [
4       [
5         11519,
6         "CAMILA CLEICIANE DA SILVA"
7       ],
8       [
9         11498,
10        "CAROLINE OLIVEIRA"
11      ],
12      [
13        11754,
14        "DEBORA DE OLIVEIRA"
15      ],
16      [
17        11567,
18        "JESSICA CAROLINE DA SILVA"
19      ],
20      [
21        11517,
22        "VANESSA RODRIGUES DA SILVA"
23      ]
24    ],
25    "CODIGO": 161392,
26    "VENCIMENTOS": [
27      1,
28      5,
29      10,
30      11,
31      12,
32      15,
33      20,
34      25
35    ],
36    "NOME": "AVISTA SHOPPING ESTACAO BH"
37  },
38  "RETORNO": 0
39 }

```

No caso de falha, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Usuário e/ou senha inválidos."
4 }

```

2.1.1.2 – Cadastro de Celular

A funcionalidade “cadastro de celular” realiza a primeira etapa do cadastro.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/cadastrar/celular/"
```

Obs.: Onde está escrito "endereco.webservice" deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o número de celular do cliente, para que o mesmo possa ser cadastrado em nossa base de dados. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para o cadastro de celular.

```
1 {
2   "PARAM_NUMERO_CELULAR": "27999437851"
3 }
```

Vale ressaltar que o número de telefone pode ser enviado **com ou sem máscara**. A mesma será tratada no WebService.

Outro ponto a ressaltar é que a validação do código segurança será realizada no próprio aplicativo, e o mesmo será devolvido no JSON de retorno.

Quando o cadastro foi bem sucedido, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```
1 {
2   "DADOS": {
3     "CODIGO": "70710"
4   },
5   "RETORNO": 0,
6   "MENSAGEM": "O número de celular que você informou precisa estar
em suas mãos, pois você receberá um SMS com o código de segurança
para finalizar o seu cadastro."
7 }
```

No caso de falha no cadastro, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```
1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }
```

2.1.1.3 – Consultar CPF

A funcionalidade "consultar CPF" tem a finalidade, como o próprio nome já diz, de devolver os dados vinculados ao CPF informado.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/consultar/cpf/"
```

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o Número de Celular, o CPF do cliente e uma informação para saber se a consulta está sendo feita pelo módulo lojista ou não, para que seja possível consultar os dados do mesmo. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para a consulta de dados do CPF.

```
1 {
2   "PARAM_NUMERO_CELULAR": "27999437851",
3   "PARAM_NUMERO_CPF": "13805236727",
4   "PARAM_LOJISTA": "0"
5 }
```

Vale ressaltar que o número de CPF pode ser enviado **com ou sem máscara**. A mesma será tratada no WebService.

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```
1 {
2   "DADOS": {
3     "VENCIMENTOS": [
4       1,
5       5,
6       10,
7       11,
8       12,
9       15,
10      20,
11      25
12    ],
13    "ACAO": 0,
14    "DATA_NASCIMENTO": "",
15    "NOME": ""
16  },
17  "RETORNO": 0
18 }
```

Vale ressaltar que o array “VENCIMENTOS” só estará preenchido quando o parâmetro “PARAM_LOJISTA” estiver como “0”, caso contrário será enviado vazio.

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```
1 {
2   "RETORNO":1,
3   "MENSAGEM":"Erro ocorrido durante o processo da funcionalidade."
4 }
```

2.1.1.4 – Consultar CEP

A funcionalidade “consultar CEP” tem a finalidade, como o próprio nome já diz, de devolver os dados vinculados ao CEP informado.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/consultar/cep/"
```

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o CEP do cliente, para que seja possível consultar os dados do mesmo. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para a consulta de dados do CEP.

```
1 {
2   "PARAM_NUMERO_CEP": "29145120"
3 }
```

Vale ressaltar que o número de CEP pode ser enviado **com ou sem máscara**. A mesma será tratada no WebService.

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```
1 {
2   "DADOS": {
3     "BAIRRO": "SANTA BARBARA",
4     "LOGRADOURO": "FRANCISCA DE PAULA BONADIMAN",
5     "UF": "ES",
6     "CIDADE": "CARIACICA"
7   },
8   "RETORNO": 0
9 }
```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO":1,
3   "MENSAGEM":"Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.1.1.5 – Consultar CEP Validador

A funcionalidade “consultar CEP validador” tem a finalidade, como o próprio nome já diz, de devolver os dados vinculados ao CEP informado validando o cliente.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/consultar/cep/validador"
```

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o número de celular, o CPF do cliente e o CEP do cliente, para que seja possível consultar os dados do mesmo. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para a consulta de dados do CEP.

```

1 {
2   "PARAM_NUMERO_CELULAR": "(27) 99943-1234",
3   "PARAM_NUMERO_CPF": "123.123.123-23",
4   "PARAM_NUMERO_CEP": "12123-123"
5 }

```

Vale ressaltar que o número de CEP pode ser enviado **com ou sem máscara**. A mesma será tratada no WebService.

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "DADOS": {
3     "ACAO": 0,
4     "BAIRRO": "ALPHAVILLE INDUSTRIAL",
5     "LOGRADOURO": "GRAJAU",
6     "UF": "SP",
7     "CIDADE": "BARUERI"
8   },
9   "RETORNO": 0
10 }

```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO":1,
3   "MENSAGEM":"Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.1.1.6 – Consultar CEP Logradouro

A funcionalidade “consultar CEP” tem a finalidade, como o próprio nome já diz, de devolver os dados vinculados ao CEP informado.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/consultar/cep/logradouro/"
```

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.



Esta funcionalidade espera receber um JSON contendo o estado, a cidade, o bairro e o logradouro para que seja possível consultar os dados do mesmo. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para a consulta de dados do CEP.

```

1 {
2   "PARAM_UF": "ES",
3   "PARAM_CIDADE": "VITORIA",
4   "PARAM_BAIRRO": "CENTRO",
5   "PARAM_LOGRADOURO": "ALGUMA COISA"
6 }

```

~~Vale ressaltar que o número de CEP pode ser enviado com ou sem máscara. A mesma será tratada no WebService.~~

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "DADOS": [
3     [
4       "ES",
5       "VITORIA",
6       "EUGENIO NETTO",
7       "PRAIA DO CANTO",
8       "29055275"
9     ],
10    [
11      "ES",
12      "VITORIA",
13      "EUGENIO NETTO",
14      "SANTA LUCIA",
15      "29056235"
16    ],
17    [
18      "ES",
19      "VITORIA",
20      "EUGENIO NETTO",
21      "PRAIA DO CANTO",
22      "29055270"
23    ],
24    [
25      "ES",
26      "VITORIA",
27      "EUGENIO NETTO",
28      "SANTA LUCIA",
29      "29056240"
30    ],
31    [
32      "ES",
33      "VITORIA",
34      "RUA EUGENIO NETTO 488",
35      "PRAIA DO CANTO",
36      "29055915"
37    ]
38  ],
39  "RETORNO": 0
40 }

```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.1.1.7 – Consultar Escolaridade

A funcionalidade “consultar Escolaridade” tem a finalidade, como o próprio nome já diz, de devolver os diferentes tipos de escolaridades disponíveis.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/consultar/escolaridade/"
```

Obs.: Onde está escrito "endereco.webservice" deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade não requer nenhum parâmetro para ser consumida.

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```
1 {
2   "DADOS": [
3     [
4       1,
5       "Ensino Fundamental"
6     ],
7     [
8       2,
9       "Ensino Médio"
10    ],
11    [
12      3,
13      "Ensino Superior"
14    ]
15  ],
16  "RETORNO": 0
17 }
```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```
1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }
```

2.1.1.8 – Consultar Estado Civil

A funcionalidade “consultar Escolaridade” tem a finalidade, como o próprio nome já diz, de devolver os diferentes tipos de escolaridades disponíveis.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/consultar/estado/civil/"
```

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade não requer nenhum parâmetro para ser consumida.

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "DADOS": [
3     [
4       "S",
5       "Solteiro"
6     ],
7     [
8       "C",
9       "Casado"
10    ],
11    [
12      "V",
13      "Viúvo"
14    ],
15    [
16      "D",
17      "Divorciado"
18    ]
19  ],
20  "RETORNO": 0
21 }

```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.1.1.9 – Aprovar

A funcionalidade “cadastrar” tem a finalidade, como o próprio nome já diz, de cadastrar o cliente e com isso fazer o seu cartão.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/aprovar/"
```

Obs.: Onde está escrito “endereço.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o número de celular, o número de CPF, o nome do cliente, dentre outros, para que seja possível realizar a aprovação do mesmo. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para o cadastro.

```

1 {
2   "PARAM_CODIGO_UNIDADE_CREDENCIADA": "161392",
3   "PARAM_CODIGO_PROMOTER": "11519",
4   "PARAM_NUMERO_CELULAR": "27999437851",
5   "PARAM_NUMERO_TELEFONE": "2732425262",
6   "PARAM_NUMERO_CPF": "13805236727",
7   "PARAM_NOME": "RODRIGO TOSTA TRANCOSO",
8   "PARAM_DATA_NASCIMENTO": "19/09/1992",
9   "PARAM_ESCOLARIDADE": "3",
10  "PARAM_ESTADO_CIVIL": "S",
11  "PARAM_SEXO": "M",
12  "PARAM_DADOS_ALTERADOS": "false",
13  "PARAM_RENDA_MENSAL": "1500.0",
14  "PARAM_VENCIMENTO": "10",
15  "PARAM_NUMERO_CEP": "29145120",
16  "PARAM_UF": "ES",
17  "PARAM_CIDADE": "CARIACICA",
18  "PARAM_BAIRRO": "SANTA BARBARA",
19  "PARAM_LOGRADOURO": "FRANCISCA DE PAULA BONADIMAN",
20  "PARAM_NUMERO": "55",
21  "PARAM_COMPLEMENTO": "CASA"
22 }
```

Quando a aprovação foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

<p>Crédito</p> <pre> { "DADOS": { "TARIFA": " ", "TERMO": " ", "CODIGO": " ", "APROVADO": 0, "MENSAGEM": " " }, "RETORNO": 0 }</pre>	<p>Pré pago</p> <pre> { "DADOS": { "TARIFA": " ", "TERMO": " ", "APROVADO": 1, "MENSAGEM": " " }, "RETORNO": 0 }</pre>
--	--

No caso de falha, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.1.1.10 – Enviar Documentos

A funcionalidade “enviar documentos” tem a finalidade, de receber as imagens do cliente e dos documentos do cliente, para anexa-las à proposta.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/enviar/documentos/"
```

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o código da proposta, o as imagens, para que seja possível anexar às mesmas. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para o processo.

```

1 {
2   "PARAM_CODIGO_PROPOSTA": "318273",
3   "PARAM_IMAGEM_ASSINATURA": "...",
4   "PARAM_IMAGEM_CLIENTE": "...",
5   "PARAM_IMAGEM_DOC_FRENTE": "...",
6   "PARAM_IMAGEM_DOC_VERSO": "iVBORw0KGgoAAAANSUgAAQAA..."
7 }

```

Vale ressaltar que as imagens devem ser enviadas codificadas na base64. Ressalto também que as mesmas podem ser enviadas tanto no formato “DataURL” quanto somente o hash codificados.

Quando a configuração foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.



```

1 {
2   "RETORNO": 0,
3   "MENSAGEM": "O documentos foram anexados com sucesso."
4 }

```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO":1,
3   "MENSAGEM":"Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.1.1.11 – Verificar Aprovação

A funcionalidade “verificar aprovação” tem a finalidade de, como o próprio nome já diz, verificar se a proposta do cliente já foi aprovada.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/verificar/aprovacao/"
```

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o código da proposta, para que seja possível realizar a consulta. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para o processo.

```

1 {
2   "PARAM_CODIGO_PROPOSTA": "123456"
3 }

```

Quando a configuração foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "DADOS":{
3     "PROCESSADO":1,
4     "APROVADO":1
5   },
6   "RETORNO":0
7 }

```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO":1,
3   "MENSAGEM":"Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.1.1.12 – Vincular Cartão Temporário

A funcionalidade “vincular cartão temporário” tem a finalidade de, como o próprio nome já diz, vincular um cartão temporário a um cliente.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/vincular/temporario/"
```

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o número do plástico, o número de celular, o CPF e o CEP, para que seja possível realizar a consulta. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para o processo.

```
1 {
2   "PARAM_NUMERO_PLASTICO": "1234 1234 1234 1234",
3   "PARAM_NUMERO_CELULAR": "27999437853",
4   "PARAM_NUMERO_CPF": "687.342.267-37",
5   "PARAM_NUMERO_CEP": "06454050"
6 }
```

Quando a configuração foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```
1 {
2   "RETORNO": 0,
3   "MENSAGEM": "Cartão vinculado com sucesso."
4 }
```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```
1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }
```

2.1.1.13 – Consultar Estados

A funcionalidade “consultar estados” tem a finalidade, como o próprio nome já diz, de devolver a lista de estados disponíveis no sistema.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/consultar/uf/"
```

Obs.: Onde está escrito "endereco.webservice" deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade não requer nenhum parâmetro para ser consumida.

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```
1 {
2   "RETORNO": 0,
3   "DADOS": [
4     [
5       "AC", "AC"
6     ],
7     [
8       "AL", "AL"
9     ],
10    [
11      "AM", "AM"
12    ],
13    [
14      "AP", "AP"
15    ],
16    [
17      "BA", "BA"
18    ],
19    [
20      "CE", "CE"
21    ],
22    [
23      "DF", "DF"
24    ],
25    [
26      "ES", "ES"
27    ]
28  ]
29 }
```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```
1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }
```

2.1.1.14 – Consultar Cidades

A funcionalidade “consultar cidades” tem a finalidade, como o próprio nome já diz, de devolver a lista de cidades disponíveis no sistema.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/consultar/cidade/"
```

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o estado, para que seja possível realizar a consulta de cidades. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para o cadastro.

```
1 {  
2   "PARAM_UF": "ES"  
3 }
```

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO": 0,
3   "DADOS": [
4     [
5       "ALEGRE",
6       "ALEGRE"
7     ],
8     [
9       "ALFREDO CHAVES",
10      "ALFREDO CHAVES"
11     ],
12     [
13       "VIANA",
14       "VIANA"
15     ],
16     [
17       "VILA VELHA",
18       "VILA VELHA"
19     ],
20     [
21       "VITORIA",
22       "VITORIA"
23     ]
24   ]
25 }

```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.1.1.15 – Consultar Bairros

A funcionalidade “consultar bairros” tem a finalidade, como o próprio nome já diz, de devolver a lista de bairros disponíveis no sistema.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/consultar/bairro/"
```

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o estado e a cidade, para que seja possível realizar a consulta de bairros. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para o cadastro.

```

1 {
2   "PARAM_UF": "ES",
3   "PARAM_CIDADE": "CARIACICA"
4 }

```

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO": 0,
3   "DADOS": [
4     [
5       "ALICE COUTINHO",
6       "ALICE COUTINHO"
7     ],
8     [
9       "ALTO BOA VISTA",
10      "ALTO BOA VISTA"
11     ],
12     [
13       "ALTO LAJE",
14       "ALTO LAJE"
15     ],
16     [
17       "ANTONIO FERREIRA BORGES",
18       "ANTONIO FERREIRA BORGES"
19     ],
20     [
21       "BANDEIRANTES",
22       "BANDEIRANTES"
23     ],
24     [
25       "BELA AURORA",
26       "BELA AURORA"
27     ],
28     [
29       "CAMPO GRANDE",
30       "CAMPO GRANDE"
31     ]
32   ]
33 }

```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.1.1.16 – Consultar Lojas

A funcionalidade “consultar lojas” tem a finalidade, como o próprio nome já diz, de devolver a lista de lojas disponíveis no sistema.

O endereço de acesso a esta funcionalidade é:

```
"http://endereco.webservice/BossRest/nao/cliente/consultar/lojas/"
```

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o estado, a cidade e o bairro, para que seja possível realizar a consulta de lojas. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para o cadastro.

```
1 {
2   "PARAM_UF": "ES",
3   "PARAM_CIDADE": "CARIACICA",
4   "PARAM_BAIRRO": "CAMPO GRANDE"
5 }
```

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```
1 {
2   "RETORNO": 0,
3   "DADOS": [
4     {
5       "CODIGO": 12246,
6       "TIPO_CARTAO": "ACTITUDE MODAS",
7       "ENDERECO": "RUA PIO XII - CAMPO GRANDE - CARIACICA - (27)3226-8100",
8       "NOME": "ACTITUDE MODAS-12246-CARIA"
9     },
10    {
11      "CODIGO": 1171,
12      "TIPO_CARTAO": "ADEMAR CUNHA",
13      "ENDERECO": "AV EXPEDITO GARCIA, 159 - CAMPO GRANDE - CARIACICA - (27)3336-5016",
14      "NOME": "ADEMAR CUNHA-1171-CARIAC"
15    }
16  ]
17 }
```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```
1 {  
2   "RETORNO":1,  
3   "MENSAGEM":"Erro ocorrido durante o processo da funcionalidade."  
4 }
```

2.2 – Não Sou Cliente

Por hora seguir os mesmo parâmetros e URL's do módulo "Sou Lojista".

Apêndice A

Endereço do Servidor: 177.125.181.172.