

Documentação  
**WebService**  
Cliente

adapter 

# Sumário

CONTROLE DE VERSÃO.....	10
GERAL.....	11
AUTENTICAÇÃO.....	12
Gerar Token.....	12
Renovar Token.....	13
Resetar senha do App.....	14
Buscar notificações do APP.....	15
Atualizar notificação do APP.....	16
Redefinir Senha.....	17
Logar no App - Completo.....	18
Logar no App - Simplificado.....	20
Consultar última versão do APP.....	21
AUTENTICAÇÃO - Usuários.....	22
Logar no sistema (validação por login e senha).....	22
Buscar segmentos disponíveis para o usuário.....	23
Buscar empresas disponíveis para o usuário.....	24
Alterar senha de usuário da aplicação.....	25
COMERCIAL - Contratos.....	26
Informações de Contrato.....	26
Informações de Dados Rede de Contrato.....	27
Alteração de Dados Rede De Contrato.....	28
Buscar contratos e faturas.....	29
Buscar dados do contrato.....	31
Buscar total navegado pelo contrato.....	32
Buscar total de ligações agrupado por tipo de ligação.....	33
Buscar extrato de navegação.....	34

Buscar extrato de ligação.....	36
Habilitar contrato em confiança.....	38
Habilitar contrato em confiança.....	39
Solicitar atualização de dados do cliente.....	40
Verificar se o cliente pode alterar para fatura digital.....	42
Alterar forma de cobrança para fatura digital.....	43
Verificar se o cliente pode alterar o vencimento do contrato.....	44
Alterar vencimento dos contratos de um cliente.....	45
Alterar vencimento de contratos por cliente.....	46
Baixar PDF do modelo de aceite do contrato.....	47
Confirmar aceite do contrato.....	48
Buscar dados de cliente, contrato e fatura por CPF/CNPJ.....	49
Buscar dados de cliente, contrato e fatura por ID de cliente.....	51
Buscar dados do contrato e faturas correspondentes.....	53
Fazer upload de arquivos para o cadastro de cliente.....	54
Baixar PDF do modelo do contrato.....	56
Busca Clientes para PlayHub.....	57
Busca Cliente por CPF/CNPJ para PlayHub.....	58
Listar contratos para inativação temporária.....	59
Inativar contratos temporariamente.....	60
Buscar tipos de contrato.....	61
Buscar tecnologias.....	62
Buscar canais de venda ativos.....	63
Informações da Última Conexão de Equipamento.....	64
Disponibilização dos Dias Disponíveis para Alteração de Vencimento de Contratos.....	66
Alteração de Dia de Vencimento de Contratos.....	67
Buscar promoções ativas de acordo com parâmetros.....	68

Buscar planejamentos tributários de acordo com a empresa e o plano.....	70
Buscar velocidades dos planos de tabela de preço de acordo com parâmetros.....	71
Buscar área de cobertura por cidade.....	73
Buscar valores dos planos de tabela de preço de acordo com parâmetros.....	74
Buscar pacotes de voz para venda extra.....	75
Buscar canais de TV para venda extra.....	77
Buscar pacotes de canais de TV para venda extra.....	78
Cadastrar contrato de INTERNET.....	80
Cadastrar contrato de TELEFONIA.....	83
Cadastrar contrato de TV.....	86
Cadastrar contrato de demais tecnologias.....	88
Cadastrar pacotes de contratos.....	90
}.....	93
COMERCIAL / TERCEIROS - Endereços.....	94
Buscar endereço completo por CEP.....	94
Buscar cidades de um estado.....	95
Buscar bairros de uma cidade.....	96
Buscar endereços (logradouros) de um bairro.....	97
Buscar condomínios do ConnectMaster por endereço e número.....	98
Buscar condomínios por endereço e número.....	98
Buscar classificações de endereço ativas.....	100
COMERCIAL - Prospectos.....	101
Buscar tipos de prospectos.....	101
Cadastrar prospectos.....	102
Cadastrar endereço de prospecto.....	104
Buscar planos disponíveis para o prospecto por tecnologia e tipo de contrato.....	106
Buscar pacotes de planos disponíveis para o prospecto tipo de contrato.....	107

Cadastrar serviços de interesse de prospecto.....	109
VIABILIDADE - Consultar crédito interno.....	110
VIABILIDADE - Consultar crédito por endereço de prospecto.....	112
VIABILIDADE - Consultar SPC de prospecto.....	114
VIABILIDADE - Consultar Clear Sale de prospecto.....	116
VIABILIDADE - Consultar viabilidade técnica de prospecto.....	118
VIABILIDADE - Consultar viabilidade técnica por endereço.....	119
VIABILIDADE - Consultar crédito por endereço.....	121
Buscar status de vendas do vendedor.....	123
Buscar prospectos vendedor ID, nome e oportunidade.....	124
Buscar prospecto por ID.....	125
Aprovar ou reprovar prospecto.....	128
Fazer upload de arquivos para o cadastro de prospecto.....	129
Buscar arquivos de prospecto.....	130
Atualizar dados de prospectos.....	131
COMERCIAL - Clientes.....	132
Buscar CFOPs.....	132
Buscar áreas de atuação para clientes PJ.....	133
Buscar tipos de atividade para clientes PJ.....	134
Buscar tributações federais para clientes PJ.....	135
Buscar tipos de contato.....	136
Converter prospecto em cliente.....	137
Cadastrar áreas de atuação de um cliente PJ.....	139
Cadastrar dados para débito automático de cliente.....	140
Informações de Cliente.....	143
Habilitar/desabilitar notificações de cliente.....	144
Resetar senha do App.....	146

Cadastrar endereço em cliente.....	147
Buscar endereços de cliente.....	149
Buscar origens comerciais.....	151
Buscar planos disponíveis para venda por tecnologia, tipo de pessoa, tipo de contrato e cidade.....	153
Buscar pacotes de planos disponíveis para venda por tipo de contrato e cidade.....	155
Buscar dados de débito de cliente.....	157
Buscar caixas com saídas livres próximas ao endereço do cliente.....	158
Buscar planos prospectados de cliente.....	160
Buscar clientes por vendedor, ID e documento.....	161
COMERCIAL – OmniChannel.....	162
Cadastrar dados de ligação entrante na URA [MOB].....	162
Busca de Atendimentos.....	163
Disponibilização de Datas Disponíveis para Reagendamento de Visitas Técnicas.....	165
Efetivação do Reagendamento de Visitas Técnicas.....	166
Enviar Linha Digitável da Fatura por SMS ao Cliente.....	167
Agendamento de Pagamentos.....	168
Disponibilização dos Dias Disponíveis para Alteração de Vencimento de Contratos.....	169
Alteração de Dia de Vencimento de Contratos.....	170
Encaminhamento de E-mail com Regras de Mudanças de Titularidade.....	171
Reboot do Equipamento.....	172
COMERCIAL - Atendimentos.....	173
Buscar tipos de atendimentos.....	173
Buscar atendimentos.....	174
Cadastrar atendimento.....	176
Inserir documentos em atendimento.....	177
Fechar atendimento.....	179
Abrir atendimento pelo contrato.....	180

Informações de Atendimentos.....	181
Visita Técnica - Disponibilização de Datas Disponíveis para Reagendamento de Visitas Técnicas.....	183
Visita Técnica - Busca de Atendimentos.....	184
Visita Técnica - Cancelar Reagendamento de Visitas Técnicas.....	186
Visita Técnica - Efetivação do Reagendamento de Visitas Técnicas.....	187
Visita Técnica - Atribuição de Notas em Atendimentos.....	188
Visita Técnica - Busca Notas de Atendimentos.....	189
Encaminhamento de E-mail com Informações do Atendimento.....	190
Buscar dias e quantidades de visitas disponíveis por dia para agendamento de acordo com tecnologia, tipo de serviço, tipo de contrato e cidade.....	191
Buscar fechamentos de atendimentos externos por tipo de atendimento.....	192
Cadastrar log de atendimento.....	193
Reverter status de atendimento para PENDENTE.....	194
Buscar dados detalhados do atendimento.....	195
COMERCIAL – Infobip.....	197
Busca Último Registro de Log de Integração com a API da Infobip.....	197
Atualiza Registros de Log de Integração com a API da Infobip.....	198
COMERCIAL – Configurações.....	199
Buscar versões do app de vendas.....	199
FINANCEIRO.....	200
Baixar PDF da fatura.....	200
Baixar PDF da fatura 2.....	201
Baixar PDF da fatura 3.....	202
Baixar PDF de nota fiscal.....	203
Buscar notas fiscais.....	204
Buscar código de barras da fatura.....	206
Validar declaração quitação anual de débito.....	207
Baixar PDF da declaração de quitação anual de débito.....	208

Buscar vencimentos de contrato.....	209
Buscar faturas por CPF.....	210
Buscar faturas em aberto por cliente.....	211
Realizar negociação.....	212
Enviar faturas em aberto para o e-mail do cliente.....	214
Buscar linha digitável da fatura.....	215
Buscar todas as faturas de um cliente por CPF/CNPJ.....	216
Buscar faturas em aberto por CPF/CNPJ.....	218
Buscar bandeiras de cartão.....	219
Buscar todas formas de pagamento.....	220
Buscar contas financeiras de acordo com a forma de pagamento.....	221
Pagar fatura.....	222
Buscar dados simplificados de faturas por CPF.....	224
Buscar bancos.....	225
Buscar serviços disponíveis para a empresa e a cidade.....	226
Cadastrar serviço adicional.....	227
Buscar formas de cobrança por cidade e tipo.....	228
Buscar formas de cobrança ativas.....	229
REDES.....	230
Buscar dados de navegação.....	230
Buscar justificativas de alteração de caixa.....	232
}.....	232
Desprovisionar equipamento.....	233
Liberar saída de caixa de contrato.....	234
TELEFONIA.....	235
Buscar prefixos por cidade.....	235
Buscar números disponíveis para venda por prefixo.....	236



INTEGRAÇÕES PASSIVAS (SVA).....	237
Gateway NEO - Autenticação.....	237
Gateway NEO - Autorização.....	238
BIT TRAINERS - Autenticação.....	239
BIT TRAINERS - Autorização.....	240

## CONTROLE DE VERSÃO

Versão	Data	Autor	Descrição
1.0	27/09/2019	Samuel Cunha	Atualização da Documentação do Webservice Cliente
1.1	06/12/2012	Samuel Cunha	Atualização da Documentação do Webservice Cliente, adicionando informações da API de obter Token de acesso.
1.2	28/02/2020	Danielle Souza	Criação de método para negociação
1.3	18/04/2020	Danielle Souza	Criação de métodos para acesso da ura
1.4	02/07/2020	Danielle Souza	Criação de método de telefonia para inserção de dados de ligação
1.5	15/07/2020	Danielle Souza	Inserção de parâmetro para limitar o número de faturas retornados nos métodos que busca dados de cliente [MOB]
1.6	27/07/2020	Danielle Souza	Inserção de parâmetros de vencimento para limitar o número de faturas vindas nos métodos que busca dados de cliente [MOB], inserção de parâmetros para especificar quais faturas serão enviadas para o e-mail do cliente no respectivo método. Criação de método para trazer todas as faturas (pagas e não pagas) de um cliente através do CPF/CNPJ
1.7	13/08/2020	Danielle Souza	Criação de novo método que permite fazer upload de arquivos para o cadastro de cliente
1.8	21/08/2020	Danielle Souza	Criação de segundo método para baixar fatura e criação e novo método para buscar dados simplificados das faturas de um cliente
1.9	30/08/2020	Samuel Cunha	Melhoria no método de Aceite de Contrato e Criação de um novo método para o Download do Modelo de Contrato
2.0	08/03/2021	Priscilla Serpa	Revisão
2.1	31/03/2021	Danielle Souza	Criação de métodos para inativação temporária de contrato
2.2	14/05/2021	Danielle Souza	Criação de métodos para prospecto e venda
2.3	25/05/2021	Gian Honório	Criação dos métodos de integração com a API da Infobip
2.4	15/06/2021	Danielle Souza	Criação de métodos para venda de contrato e pacote
2.5	22/06/2021	Danielle Souza	Criação de métodos para integração com Gateway NEO
2.6	23/08/2021	Danielle Souza	Criação de métodos para integração com BIT TRAINERS

## GERAL

Todos os métodos presentes neste documento deverão ser acessados com a autenticação do tipo basic, utilizando o usuário e senha fornecidos pelos responsáveis pela manutenção do webservice. Abaixo, estão os possíveis status que os métodos podem retornar, clique nos links para mais detalhes

Status	Descrição
<a href="#"><u>ACCEPTED</u></a>	202 Accepted, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>BAD_REQUEST</u></a>	400 Bad Request, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>CONFLICT</u></a>	409 Conflict, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>CREATED</u></a>	201 Created, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>FORBIDDEN</u></a>	403 Forbidden, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>GONE</u></a>	410 Gone, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>INTERNAL_SERVER_ERROR</u></a>	500 Internal Server Error, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>MOVED_PERMANENTLY</u></a>	301 Moved Permanently, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>NO_CONTENT</u></a>	204 No Content, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>NOT_ACCEPTABLE</u></a>	406 Not Acceptable, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>NOT_FOUND</u></a>	404 Not Found, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>NOT_MODIFIED</u></a>	304 Not Modified, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>OK</u></a>	200 OK, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>PRECONDITION_FAILED</u></a>	412 Precondition Failed, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>SEE_OTHER</u></a>	303 See Other, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>SERVICE_UNAVAILABLE</u></a>	503 Service Unavailable, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>TEMPORARY_REDIRECT</u></a>	307 Temporary Redirect, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>UNAUTHORIZED</u></a>	401 Unauthorized, veja <a href="#">HTTP/1.1 documentation</a> .
<a href="#"><u>UNSUPPORTED_MEDIA_TYPE</u></a>	415 Unsupported Media Type, veja <a href="#">HTTP/1.1 documentation</a> .

## Gerar Token

Gera Token de acesso, com a duração definida pelo sistema. Que deverá ser passado no headers com o Name: Token

[https://\[URL\]/ws/auth/token/gerar](https://[URL]/ws/auth/token/gerar)

GET

[https://\[URL\]/ws/auth/token/gerar](https://[URL]/ws/auth/token/gerar)

```
{
    "token":
    "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI2MGE0ZDA4Zi0xYmEzLTQwYmYtYWMOYi0wYTg0MmI4NjU5YTUiLCJpc3MiOiJXZWJTZXJ2aWNLiiwicmVuIjowLCJleHAiOjE1NzU2NDMzMjI2NDgsIm1hdCI6MTU3NTYOMDEyMjY0OCwiYWxpIjojcCwicXRkUmVudjowfQ._3_OQ9AMQRUIseRFsIp8G_P3vCuWSJlWZahQGGr7DU9Z4"
}
```

## Renovar Token

### Descrição

Renova Token de acesso, por mais um período definido pelo sistema.

### URL

[https://\[URL\]/ws/auth/token/renovar](https://[URL]/ws/auth/token/renovar)

### Método

PUT

### Entrada

HeaderParam: Name = "Token", Value = Token gerado pela api de Gerar Token

### Saída

JSON contendo Token gerado para uso.

**Envio:** [https://\[URL\]/ws/auth/token/gerar](https://[URL]/ws/auth/token/gerar)

### Retorno:

```
{
  "token":
"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI2MGE0ZDA4Zi0xYmEzLTQwYmYtYWM0Yi0wYTg0MmI4NjU5YTYiLCJpc3MiOiJXZWJ0ZXJ2aWNlIiwicmVuIjowLCJleHAiOiJlNzU2NDM3MjI2NDgsIm1hdCI6MTU3NTY0MDEyMjY0OCwiYW50b290IjoxMCwicXVzUmVudjowfQ.3_OQ9AMQRUiSeRFsIp8G_P3vCuwSJ1WZaHqGr7DU9Z4"
}
```

## Resetar senha do App

### Descrição

Reseta a senha do cliente.

### URL

[https://\[URL\]/ws/comercial/resetar\\_senha](https://[URL]/ws/comercial/resetar_senha)

### Método

POST

### Entrada

JSON contendo o usuário para reseta senha

### Saída

Resposta informando se o atendimento foi criado.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{  
  "usuario": "12312312312"  
}
```

#### Retorno:

Em caso de sucesso:

```
{  
  "Success": "Senha resetada com sucesso, foi enviado um SMS com a  
nova senha para o telefone cadastrado."  
}
```

Ou em caso de erro:

```
{  
  "error": "Nao foi possivel encontrar um cliente com esses dados de  
acesso"  
}
```

Ou:

```
{  
  "error": "Não foi possivel reseta a senha.",  
  "Motivo": "Ocorreu um erro ao enviar o SMS ou salvar a nova senha: "  
}
```

## Buscar notificações do APP

### Descrição

Obtém as notificações do cliente informado.

### URL

[https://\[URL\]/ws/comercial/avisos\\_app/cliente/{idCliente}](https://[URL]/ws/comercial/avisos_app/cliente/{idCliente})

### Método

POST

### Entrada

Pelo PathParam passar o código do cliente e JSON contendo usuário e senha

### Saída

JSON contendo listas de notificações para o cliente.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

```
[
  {
    "Status": "PENDENTE",
    "DataEnvio": 1510315302000,
    "Descricao": "TESTE",
    "IDLogAvisoApp": 1,
    "Mensagem": "BATATA",
    "Titulo": "BATATA"
  }
]
```

## Atualizar notificação do APP

### Descrição

Atualiza o status da notificação do app.

### URL

[https://\[URL\]/ws/comercial/avisos\\_app/{id}](https://[URL]/ws/comercial/avisos_app/{id})

### Método

POST

### Entrada

Pelo PathParam passar o código da notificação a ser atualizada e JSON contendo usuário e senha

### Saída

Retorna a mensagem informando.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

"OK": A notificação teve o seu status atualizado para "ENVIADO".



## Redefinir Senha

### Descrição

Redefine a senha do cliente

### URL

[https://\[URL\]/ws/comercial/redefinir\\_senha?senhaAntiga={senhaAntiga}&novaSenha={novaSenha}](https://[URL]/ws/comercial/redefinir_senha?senhaAntiga={senhaAntiga}&novaSenha={novaSenha})

### Método

POST

### Entrada

Por QueryParam passar a senha antiga e a nova senha e JSON contendo usuário e senha

### Saída

JSON contendo a resposta.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

Em caso de sucesso:

```
{
  "success": "Senha redefinida com sucesso"
}
```

Ou em caso de erro:

```
{
  "error": "Nao foi possivel encontrar um cliente com esses dados de acesso"
}
```

Se a senha atual do cliente for diferente da cadastrada no banco:

```
{
  "error": "Senha atual digitada, diferente da senha cadastrada"
}
```

## Logar no App - Completo

### Descrição

Realiza login no Aplicativo.

## URL

https://[URL]/ws/comercial/logar

## Método

POST

## Entrada

## JSON contendo usuário e senha

## Saída

JSON contendo informações básicas do cliente, assim como contratos, faturas em aberto e número de atendimentos abertos

## Campos da Entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

## Exemplo

**Envio:**

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

**Retorno:**

[illegible]

[illegible]

## Logar no App - Simplificado

### Descrição

Realiza login no Aplicativo.

### URL

[https://\[URL\]/ws/comercial/clientes/logar](https://[URL]/ws/comercial/clientes/logar)

### Método

POST

### Entrada

JSON contendo usuário e senha

### Saída

JSON contendo informações básicas do cliente, assim como contratos, faturas em aberto e número de atendimentos abertos

### Campos da Entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

```
{
  "Celular": "31999999999",
  "NomeCompleto": "CLIENTE TESTE",
  "Email": "teste@teste.com",
  "CodigoCliente": 81593,
  "NomeTratamento": "CLIENTE",
  "TelefoneFixo": "3133333333",
  "DataNascimento": 574398000000,
  "CPF": "12312312312",
  "Contratos": [
    {
      "ValorContrato": 114.9,
      "Tecnologia": "FIBRA",
      "PlanoContrato": "FIBRA ÓPTICA 30MB",
      "StatusContrato": "HABILITADO",
      "CodContrato": 61049,
      "IDVencimento": 1
    }
  ]
}
```

## Consultar última versão do APP

### Descrição

Obtém a última versão do App.

### URL

[https://\[URL\]/ws/comercial/configuracoes\\_app](https://[URL]/ws/comercial/configuracoes_app)

### Método

GET

### Entrada

Não Possui

### Saída

JSON contendo um Map com os valores das chaves de configuração.

### Exemplo

### Retorno:

```
{  
  "NIVEL_ATUALIZACAO": "1",  
  "ULTIMA_VERSAO_APP": "1.0"  
}
```

## AUTENTICAÇÃO - Usuários

### Logar no sistema (validação por login e senha)

#### Descrição

Logar no sistema através de usuário e senha

#### URL

[https://\[URL\]/ws/auth/login](https://[URL]/ws/auth/login)

#### Método

POST

#### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: JSON contendo os dados de usuário e senha

#### Saída

JSON contendo a ID do usuário ou JSON contendo mensagem de erro

#### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
usuario*	body	String	
senha*	body	String	

\* Parâmetro obrigatório

#### Exemplo entrada:

```
{
  "usuario": "joao.souza",
  "senha": "abc123"
}
```

#### Exemplo retorno:

##### Sucesso:

```
{
  "idUsuario": 285,
  "idPerfil": 1,
  "vendedor": true,
  "podeAprovarReprovarProspecto": true
}
```

##### Falha:

```
{
  "error": "Usuário e senha informados não pertencem a um usuário ativo no sistema."
}
```

## Buscar segmentos disponíveis para o usuário

### Descrição

Busca os segmentos disponíveis para o usuário

### URL

[https://\[URL\]/ws/comercial/segmentos/usuario/{idUserio}](https://[URL]/ws/comercial/segmentos/usuario/{idUserio})

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

PathParam: idUsuario

### Saída

JSON contendo lista de segmentos disponíveis para aquele usuário ou JSON contendo mensagem de erro

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/segmentos/usuario/111](https://[URL]/ws/comercial/segmentos/usuario/111)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "descricao": "RESIDENCIAL"
    },
    {
      "id": 2,
      "descricao": "CORPORATIVO"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar empresas disponíveis para o usuário

### Descrição

Busca as empresas disponíveis para o usuário

### URL

[https://\[URL\]/ws/terceiros/empresas/usuario/{idUsuario}](https://[URL]/ws/terceiros/empresas/usuario/{idUsuario})

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

PathParam: idUsuario

### Saída

JSON contendo lista de empresas disponíveis para aquele usuário ou JSON contendo mensagem de erro

### Exemplo

**Envio:** [https://\[URL\]/ws/terceiros/empresas/usuario/111](https://[URL]/ws/terceiros/empresas/usuario/111)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "nome": "EMPRESA A",
      "nomeFantasia": "EMPRESA A",
      "cnpj": "11111111111111"
    },
    {
      "id": 2,
      "nome": "EMPRESA B",
      "nomeFantasia": "EMPRESA B",
      "cnpj": "22222222222222"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```



## Alterar senha de usuário da aplicação

### Descrição

Alterar senha de usuário

### URL

[https://\[URL\]/ws/auth/alterar\\_senha](https://[URL]/ws/auth/alterar_senha)

### Método

PUT

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: JSON contendo os dados de usuário e nova senha

### Saída

JSON contendo a ID do usuário ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	String	
senhaAntiga*	body	String	
novaSenha*	body	String	

**\* Parâmetro obrigatório**

### Exemplo entrada:

```
{
  "idUsuario": 1,
  "senhaAntiga": "123456",
  "novaSenha": "789456"
}
```

### Exemplo retorno:

#### Sucesso:

```
{
  "success": "Senha alterada."
}
```

#### Falha:

```
{
  "error": "A senha antiga está incorreta."
}
```

## COMERCIAL - Contratos

### Informações de Contrato

#### Descrição

Busca informações de contratos de acordo com os filtros passados por parâmetro.

#### URL

https://{{url}}/ws/area/cliente/contrato/{idContrato: [0-9]+}

#### Método

GET

#### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

PathParam: *idContrato*

#### Saída

JSON contendo os dados solicitados.

#### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idContrato*	path	Number	

\* Parâmetro obrigatório

#### Exemplo retorno:

```
{
  "data": {
    "descricaoStatus": "HABILITADO",
    "vencimentoFidelidade": 1639061038000,
    "enderecos": [
      "PADRAO - RUA TESTE, 669 - TESTE -TESTE -560000000"
    ],
    "descricaoVelocidade": "BLS 1MB",
    "descricaoFormaCobranca": "SPMX-BOLETO SANTANDER CR - 5724031",
    "valor": 9.99,
    "nomePlano": "SUPER LIGEIRO EMPRESARIAL - PRE",
    "id": 311974,
    "diaVencimento": "5"
  },
  "status": "OK"
}
```

## Informações de Dados Rede de Contrato

### Descrição

Busca informações de dados rede de contratos de acordo com os filtros passados por parâmetro.

### URL

https://{{url}}/ws/area/cliente/contrato/rede/{idContrato: [0-9]+}

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

PathParam: *idContrato*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idContrato*	path	Number	

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": {
    "senhaWiFi": null,
    "fornecedorEquipamento": null,
    "criptografiaRede": null,
    "serial": null,
    "loginWiFi": null
  },
  "status": "OK"
}
```

## Alteração de Dados Rede De Contrato

### Descrição

Realiza a alteração de dados rede de contratos de acordo com as informações passadas por parâmetro.

### URL

https://{{url}}/ws/area/cliente/alterar/dados/rede

### Método

PUT

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: *idContrato*, *loginWiFi*, *senhaWiFi*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idContrato*	body	Number	
loginWiFi*	body	String	
senhaWifi*	body	String	

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": "O login e a senha do Wi-Fi do contrato de ID 'X' foram atualizadas com sucesso.",
  "status": "OK"
}
```

## Buscar contratos e faturas

### Descrição

Obtém os contratos e faturas do cliente informado.

### URL

[https://\[URL\]/ws/comercial/contratos/cliente/{idCliente}](https://[URL]/ws/comercial/contratos/cliente/{idCliente})

### Método

POST

### Entrada

Pelo PathParam passar o código do cliente e JSON contendo usuário e senha

### Saída

JSON contendo o status da edição (sucesso/erro)

### Campos da Entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

```
{
  "Celular": "31999994444",
  "NomeCompleto": "DANIELLE SCHINNIGER ASSUN SOUZA",
  "DataNascimento": 574398000000,
  "TelefoneFixo": "3130312596",
  "Email": "teste@vero.net",
  "faturas": [
    {
      "IDFatura": 25,
      "CodigoCliente": 81593,
      "nroBoleto": 35,
      "nroDocumento": "0008177049",
      "Vencimento": "2017-08-01",
      "Valor": 90,
      "DebitoAutomaico": "FALSE"
    },
    {
      "IDFatura": 26,
      "CodigoCliente": 81593,
      "nroBoleto": 27,
      "nroDocumento": "0008177048",
      "Vencimento": "2017-06-15",
      "Valor": 150,
      "DebitoAutomaico": "FALSE"
    }
  ],
}
```

```
"CodigoCliente": 81593,
"NomeTratamento": "DANIELLE",
"contratos": [
  {
    "Tecnologia": "FIBRA",
    "LoginRadius": null,
    "Franquia": 240,
    "PlanoContrato": "INTERNET FIBRA 10 MB",
    "StatusContrato": "HABILITADO",
    "UnidadeMedida": "GB",
    "CodContrato": 6976,
    "tipoEndereco": "PADRAO",
    "GrupoTecnologia": "INTERNET",
    "EnderecoInstalacao": "FRANCISCA MALVINA DE RESENDE, 444",
    "IDVencimento": "1"
  },
  {
    "Tecnologia": "RADIO",
    "LoginRadius": null,
    "Franquia": 270,
    "PlanoContrato": "RADIO 10 MB",
    "StatusContrato": "HABILITADO",
    "UnidadeMedida": "GB",
    "CodContrato": 7058,
    "tipoEndereco": "PADRAO",
    "GrupoTecnologia": "INTERNET",
    "EnderecoInstalacao": "ENIDIO JOSE, 966",
    "IDVencimento": "1"
  }
],
"atendimentosAbertos": 666
}
```

## Buscar dados do contrato

### Descrição

Busca dados gerais do contrato

### URL

[https://\[URL\]/ws/comercial/contrato/dados?somenteContrato=s](https://[URL]/ws/comercial/contrato/dados?somenteContrato=s)

### Método

POST

### Entrada

JSON contendo o ID do contrato

### Saída

JSON contendo os dados do contrato

### Campos do JSON da entrada

Tag	Tipo	Observação
idContrato*	Long	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "idContrato": 1283321
}
```

#### Retorno:

```
{
  "ValorContrato": 114.9,
  "Tecnologia": "FIBRA",
  "TipoContrato": "OUTROS",
  "StatusContrato": "HABILITADO",
  "IDContrato": 61049,
  "DescricaoPlano": "FIBRA PROMO ION 10MBPS/30MBPS FRAN. 360GB",
  "Enderecos": [
    {
      "Logradouro": "RUA PEDRO RIBEIRO DE REZENDE",
      "UF": "MG",
      "Numero": 102,
      "Tipo": "PADRAO",
      "Complemento": "CASA A",
      "Bairro": "NOVO HORIZONTE",
      "Cidade": "CONSELHEIRO LAFAIETE",
      "CEP": "36402157"
    }
  ],
  "TipoTecnologia": "INTERNET",
  "IDVencimento": 1
}
```

## Buscar total navegado pelo contrato

### Descrição

Obtém o total que o cliente navegou de acordo com o contrato.

### URL

[https://\[URL\]/ws/comercial/total\\_navegacao/contrato/{idContrato}](https://[URL]/ws/comercial/total_navegacao/contrato/{idContrato})

### Método

POST

### Entrada

Pelo PathParam passar o código do contrato e JSON contendo usuário e senha

### Saída

Retorna a mensagem informando.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

```
{
  "navegadoGB": 5.068077605217695,
  "limiteGB": "360GB"
}
```



## Buscar total de ligações agrupado por tipo de ligação

### Descrição

Obtém o tempo total das ligações agrupado por tipo de ligação.

### URL

[https://\[URL\]/ws/comercial/ligacoes/contrato/{idContrato}](https://[URL]/ws/comercial/ligacoes/contrato/{idContrato})

### Método

POST

### Entrada

Pelo PathParam passar o código do contrato e JSON contendo usuário e senha

### Saída

Retorna a mensagem informando.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

```
{
  "ligacoes": [
    {
      "descricao": "FIXO DDD",
      "duracao": 5
    },
    {
      "descricao": "MOVEL DDD",
      "duracao": 15
    }
  ]
}
```

## Buscar extrato de navegação

### Descrição

Obtém o extrato de navegação de acordo com Contrato e período informado

### URL

[https://\[URL\]/ws/comercial/contrato/{idContrato}/extrato/navegacao?dataInicio={dataInicio}&dataFim={dataFim}](https://[URL]/ws/comercial/contrato/{idContrato}/extrato/navegacao?dataInicio={dataInicio}&dataFim={dataFim})

### Método

POST

### Entrada

Pelo PathParam passar o código do contrato,  
Por QueryParam passar a Data Inicial e a Data Final no formato (dd-MM-yyyy) e  
JSON contendo usuário e senha

### Saída

JSON contendo o extrato de navegação filtradas de acordo com o ID do contrato e o período fornecido.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

```
[
  {
    "id": 1,
    "dataColeta": 1499264996000,
    "idRadacct": 76984667,
    "loginRadius": "luziarsantos",
    "ipNas": "10.0.10.33",
    "inicioSessao": 1492725782000,
    "fimSessao": 1471958076000,
    "duracaoSessao": -20767706,
    "download": 391787917,
    "upload": 33658349,
    "macCliente": "00:0C:43:C2:8F:8F",
    "ipCliente": "187.73.84.95",
    "isFaturado": false,
    "circuitID": null,
    "motivoDesconexao": null,
    "duracaoFormatada": "0-5768:0-48:0-26",
    "downloadFormatado": 0.3648809315636754
  },
  {
    "id": 2,
    "dataColeta": 1499265138000,
    "idRadacct": 81463731,
    "loginRadius": "luziarsantos",
    "ipNas": "10.0.10.8",
  }
]
```

```
"inicioSessao": 1492957543000,  
"fimSessao": 1492974406000,  
"duracaoSessao": 16863,  
"download": 9342519,  
"upload": 670931,  
"macCliente": "00:0C:43:D5:17:23",  
"ipCliente": "186.235.195.228",  
"isFaturado": false,  
"circuitID": null,  
"motivoDesconexao": null,  
"duracaoFormatada": "04:41:03",  
"downloadFormatado": 0.008700898848474026
```

```
}
```

```
]
```

## Buscar extrato de ligação

### Descrição

Obtém o extrato de ligações de acordo com Contrato e período informado

### URL

[https://\[URL\]/ws/comercial/contrato/{idContrato}/extrato/ligacao?dataInicio={dataInicio}&dataFim={dataFim}](https://[URL]/ws/comercial/contrato/{idContrato}/extrato/ligacao?dataInicio={dataInicio}&dataFim={dataFim})

### Método

POST

### Entrada

Pelo PathParam passar o código do contrato,  
Por QueryParam passar a Data Inicial e a Data Final no formato (dd-MM-yyyy) e  
JSON contendo usuário e senha

### Saída

JSON contendo o extrato de ligações filtradas de acordo com o ID do contrato e o período fornecido.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

```
[
  {
    "id": 1,
    "dataColeta": 1499355958000,
    "isFaturado": false,
    "idServico": null,
    "origem": "553112341234",
    "valor": 0,
    "batch": "0800 Camara Municipal Conselheiro Lafaiete",
    "dataLigacao": 1390405381000,
    "descricao": "Brasil - MG",
    "idTipoLigacao": null,
    "descricaoTipoLigacao": null,
    "idChamada": 1,
    "destino": "553135353535",
    "duracao": 0,
    "controlNumber": 2
  },
  {
    "id": 2,
    "dataColeta": 1499355958000,
    "isFaturado": false,
    "idServico": null,
    "origem": "553112341234",
    "valor": 0,
    "batch": "0800 Camara Municipal Conselheiro Lafaiete",
```

```
"dataLigacao": 1390405454000,  
"descricao": "Brasil - MG",  
"idTipoLigacao": null,  
"descricaoTipoLigacao": null,  
"idChamada": 2,  
"destino": "553135353535",  
"duracao": 0,  
"controlNumber": 2  
}  
]
```

## Habilitar contrato em confiança

### Descrição

Habilita o plano em confiança

### URL

[https://\[URL\]/ws/comercial/contrato/{idContrato}/webservice/alterar/status?descricaoStatus={descricaoStatus}&acessoCentral={s}](https://[URL]/ws/comercial/contrato/{idContrato}/webservice/alterar/status?descricaoStatus={descricaoStatus}&acessoCentral={s})

### Método

POST

### Entrada

Por PathParam passar o código do contrato

Por QueryParam passar a descrição do status e o acesso central e

JSON contendo usuário e senha

### Saída

JSON contendo um Map com o retorno do resultado de execução do método.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

Em caso de sucesso:

```
{
  "success": "Contrato Alterado, DataHabilitacaoConfiancaFim"
}
```

Ou em caso de erro na alteração do status:

```
{
  "error": "Erro ao habilitar o contrato"
}
```

Se o contrato já tiver sido habilitado em confiança recentemente:

```
{
  "error": " Não é possível habilitar o contrato pois já foi realizada uma habilitação em confiança nos últimos {parâmetro com o valor máximo de dias cadastrados em banco} dias"
}
```

## Habilitar contrato em confiança

### Descrição

Habilita um contrato em confiança

### URL

[https://\[URL\]/ws/comercial/contrato/habilitar\\_confianca](https://[URL]/ws/comercial/contrato/habilitar_confianca)

### Método

PUT

### Entrada

JSON contendo o ID do contrato

### Saída

JSON contendo a resposta (FALHA OU SUCESSO) e a data fim da habilitação em confiança

### Campos do JSON da entrada

Tag	Tipo	Observação
idContrato*	Long	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "idContrato": 1283321
}
```

#### Retorno:

##### Se sucesso:

```
{
  "dataFim": "20/04/2020",
  "resposta": "SUCESSO"
}
```

##### Se falha:

```
{
  "resposta": "FALHA"
}
```

## Solicitar atualização de dados do cliente

### Descrição

Envia uma solicitação para realizar a atualização dos Dados do cliente.

### URL

[https://\[URL\]/ws/comercial/alterarPerfil](https://[URL]/ws/comercial/alterarPerfil)

### Método

POST

### Entrada

JSON no corpo da requisição

### Saída

JSON contendo um Map com o retorno do resultado de execução do método.

### Campos do JSON da entrada

Tag	Tipo	Observação
IDCliente	String	
NomeCompleto	String	
RG	String	
DataNascimento	String	Formato: dd-mm-yyyy
Email	String	
TelefoneFixo	String	
Celular	String	
Telefone1	String	
Telefone2	String	
Contratos	List<>	Lista com os dados do contrato a ser alterado.
Plano	String	Nome do Plano.
CodContrato	String	
TipoEndereco	String	
TipoLogradouro	String	
Logradouro	String	
Numero	String	
Complemento	String	
Bairro	String	
Cidade	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "IDCliente": "66499",
  "NomeCompleto": "LUCIANA ALEIXO TAVARES",
  "RG": "MG-10.111.3",
  "DataNascimento": "05-02-1983",
  "Email": "lucyana.aleixo@email.com",
  "TelefoneFixo": "(31) 8888-0000",
  "Celular": "(31) 11258-00000",
  "Telefone1": "",
  "Telefone2": "",
  "Contratos": [
```



```
{
  "Plano": "FIBRA ÓPTICA 5MB",
  "CodContrato": "10568",
  "TipoEndereco": "INSTALACAO",
  "TipoLogradouro": "RUA",
  "Logradouro": "DONA MARIANA GUIMARAES",
  "Numero": "513",
  "Complemento": "",
  "Bairro": "QUELUZ",
  "Cidade": "CONSELHEIRO LAFAIETE"
},
{
  "Plano": "VIAFONE FALA MAIS ILIMIT",
  "CodContrato": "83102",
  "TipoEndereco": "PADRAO",
  "TipoLogradouro": "AVENIDA",
  "Logradouro": "MARIANA GUIMARAES",
  "Numero": "513",
  "Complemento": "",
  "Bairro": "QUELUZ",
  "Cidade": "CONSELHEIRO LAFAIETE"
}
]
```

**Retorno:**

Em caso de sucesso:

```
{
  "A solicitacao foi enviada ao setor responsavel, que realizara as
  atualizacoes dos dados!"
}
```

Ou em caso de erro na alteração do perfil:

```
{
  "Exceção na execução do método"
}
```

## Verificar se o cliente pode alterar para fatura digital

### Descrição

Faz a verificação se o cliente pode alterar a forma da Fatura para Fatura digital

### URL

[https://\[URL\]/ws/comercial/cliente/{idCliente}/verificar/fatura\\_digital](https://[URL]/ws/comercial/cliente/{idCliente}/verificar/fatura_digital)

### Método

POST

### Entrada

Pelo PathParam passar o código do cliente,  
JSON contendo usuário e senha

### Saída

String informando se pode ou não alterar para fatura digital.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

Caso a resposta seja "NOT\_ACCEPTABLE", a alteração não poderá ser feita. Caso seja "ACCEPTED", está permitida a alteração.

## Alterar forma de cobrança para fatura digital

### Descrição

Alterar a forma de cobrança do cliente.

### URL

[https://\[URL\]/ws/comercial/cliente/{idCliente}/alterar/fatura\\_digital](https://[URL]/ws/comercial/cliente/{idCliente}/alterar/fatura_digital)

### Método

POST

### Entrada

Pelo PathParam passar o código do cliente,  
JSON contendo usuário e senha

### Saída

"OK" se for alterado algum contrato do cliente para a nova forma de cobrança, nesse caso o cliente receberá um e-mail informando que a alteração para fatura digital foi feita com sucesso. Caso não aconteça alteração em nenhum contrato (por não ter nenhum habilitado que não seja débito e nem isento ou por não ter forma de cobrança do tipo fatura digital ativa) a resposta será "NOT\_MODIFIED".

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

##### Se sucesso:

```
{
  "Status": "OK",
  "Contratos": [
    355568,
    355589
  ],
  "Mensagem": "Foi alterado para Fatura Digital!"
}
```

##### Se sucesso, sem modificação:

```
{
  "Status": "NOT_MODIFIED",
  "Mensagem": "Não foi possível alterar para Fatura Digital!"
}
```

##### Se falha:

```
{
  "Status": "INTERNAL_SERVER_ERROR",
  "Erro": "Mensagem informado o tipo do Erro",
  "Mensagem": "Não foi possível alterar para Fatura Digital!"
}
```

## Verificar se o cliente pode alterar o vencimento do contrato

### Descrição

Faz a verificação se o cliente pode alterar o vencimento do contrato

### URL

[https://\[URL\]/ws/comercial/cliente/{idCliente}/verificar/vencimento\\_contrato?vencimentoAtual{idVencimentoContrato}](https://[URL]/ws/comercial/cliente/{idCliente}/verificar/vencimento_contrato?vencimentoAtual{idVencimentoContrato})

### Método

POST

### Entrada

Pelo PathParam passar o código do cliente,  
Por QueryParam id do vencimento atual do contrato.  
JSON contendo usuário e senha

### Saída

String informando se pode ou não alterar o vencimento. Caso a resposta seja "NOT\_ACCEPTABLE", o vencimento já foi solicitado, porém será efetuado no próximo faturamento. Caso seja "ACCEPTED", está permitida a alteração.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

String informando se pode ou não alterar o vencimento. Caso a resposta seja "NOT\_ACCEPTABLE", o vencimento já foi solicitado, porém será efetuado no próximo faturamento. Caso seja "ACCEPTED", está permitida a alteração.

## Alterar vencimento dos contratos de um cliente

### Descrição

Altera para o novo vencimento todos os contratos de um determinado cliente que possuem o vencimento atual selecionado

### URL

[https://\[URL\]/ws/comercial/cliente/{idCliente}/alterar/vencimento\\_contrato?vencimentoAtual={idVencimentoContrato}&novoVencimento={idVencimento}](https://[URL]/ws/comercial/cliente/{idCliente}/alterar/vencimento_contrato?vencimentoAtual={idVencimentoContrato}&novoVencimento={idVencimento})

### Método

POST

### Entrada

Pelo PathParam passar o código do cliente,

Por QueryParam id do vencimento atual do contrato e id do novo vencimento.

JSON contendo usuário e senha

### Saída

"OK" se for alterado algum contrato do cliente para o novo vencimento, nesse caso o cliente receberá um e-mail informando que a alteração do vencimento foi feita com sucesso. Caso a alteração de vencimento venha a acontecer depois do próximo faturamento, o cliente também é informado no e-mail. Caso não aconteça alteração em nenhum contrato a resposta será "NOT\_MODIFIED".

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

"OK" se for alterado algum contrato do cliente para o novo vencimento, nesse caso o cliente receberá um e-mail informando que a alteração do vencimento foi feita com sucesso. Caso a alteração de vencimento venha a acontecer depois do próximo faturamento, o cliente também é informado no e-mail. Caso não aconteça alteração em nenhum contrato a resposta será "NOT\_MODIFIED".

## Alterar vencimento de contratos por cliente

### Descrição

Altera para um novo vencimento todos os contratos de um cliente que possuem o vencimento atual selecionado

### URL

[https://\[URL\]/ws/comercial/cliente/{idCliente}/alterar/vencimentos](https://[URL]/ws/comercial/cliente/{idCliente}/alterar/vencimentos)

### Método

PUT

### Entrada

PathParam com o ID do cliente

JSON contendo o ID do vencimento atua e ID do novo vencimento

### Saída

String de FALHA OU SUCESSO

### Campos do JSON da entrada

Tag	Tipo	Observação
idVencimentoAtual*	Long	
idNovoVencimento*	Long	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "idVencimentoAtual": "2",
  "idNovoVencimento": "4"
}
```

**Retorno:** "SUCESSO" ou "FALHA"

## Baixar PDF do modelo de aceite do contrato

### Descrição

Faz download do Modelo de aceite de contrato informado

### URL

[https://\[URL\]/ws/comercial/contratos/modelo\\_contrato/{idContrato}](https://[URL]/ws/comercial/contratos/modelo_contrato/{idContrato})

### Método

POST

### Entrada

Pelo PathParam passar o código do contrato,  
JSON contendo usuário e senha

### Saída

PDF com o modelo de aceite de contrato.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

PDF com o modelo de aceite de contrato.

## Confirmar aceite do contrato

### Descrição

Faz a confirmação do Aceite.

### URL

[https://\[URL\]/ws/comercial/contratos/aceite/{idContrato}](https://[URL]/ws/comercial/contratos/aceite/{idContrato})

### Método

GET

### Entrada

Pelo PathParam passar o código do contrato,  
E pelo Header o Token gerado no método Gera Token

### Saída

OK, caso o aceite tenha sido dado com sucesso, ou INTERNAL SERVER caso haja algum problema ao alterar o aceite.

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

```
{
  "Status": "INTERNAL_SERVER_ERROR",
  "Mensagem": "Não foi possível realizar o aceite."
}
```

```
{
  "Status": "OK",
  "Mensagem": "Aceite realizado com sucesso."
}
```



## Buscar dados de cliente, contrato e fatura por CPF/CNPJ

### Descrição

Busca dados gerais dos contratos e faturas de um cliente

### URL

[https://\[URL\]/ws/comercial/cliente/dados?dataInicioVencFatura=01-07-2020&dataFimVencFatura=30-07-2020](https://[URL]/ws/comercial/cliente/dados?dataInicioVencFatura=01-07-2020&dataFimVencFatura=30-07-2020)

### Método

POST

### Entrada

JSON contendo o cpf ou cnpj, QueryParam("dataInicioVencFatura") (opcional) e QueryParam("dataFimVencFatura") (opcional) para limitar quais vencimentos de fatura trará.

**OBS.:** Caso não queira limitar os vencimentos das faturas na resposta, é só não enviar o parâmetro os parâmetros de data.

### Saída

JSON contendo os dados do cliente

### Campos do JSON da entrada

Tag	Tipo	Observação
cpfcnpj*	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "cpfcnpj": "12312312312"
}
```

#### Retorno:

```
{
  "TipoPessoa": "PessoaFisica",
  "ClienteEncontrado": true,
  "NomeCliente": "CLIENTE TESTE",
  "ExisteFaturaSemPagamento": true,
  "IDCliente": 81593,
  "Contratos": [
    {
      "ValorContrato": 105.49,
      "StatusContrato": "HABILITADO",
      "IDContrato": 1283299,
      "DescricaoPlano": "FIBRA RES 10MBPS FRANQUIA 360GB",
      "TipoContrato": "RESIDENCIAL",
      "TipoTecnologia": "INTERNET",
      "IDVencimento": 4
    },
    {
      "ValorContrato": 9.9,
      "StatusContrato": "HABILITADO EM CONFIANCA",
      "IDContrato": 1283303,
      "DescricaoPlano": "LOCACAO DE EQUIPAMENTOS",
      "TipoContrato": "RESIDENCIAL",
      "TipoTecnologia": "INTERNET",
      "IDVencimento": 4
    }
  ],
  "ExisteAtendimentoAberto": true,
  "Faturas": [
    {
      "idsContratos": [1283299,1283303],
      "DebitoAutomatico": false,
      "CodigoCliente": 81593,
      "ValorMulta": 0.0,
      "linhaDigitavel": "11111.11111 11111.11111 11111.11111 1 11111111111111",
      "IDFatura": 90,
      "ValorJuros": 0.0,
    }
  ]
}
```

$\},$ 

}

 $\{$ 

}

**OU** "FALHA"

## Buscar dados de cliente, contrato e fatura por ID de cliente

### Descrição

Busca dados gerais dos contratos e faturas de um cliente

### URL

[https://\[URL\]/ws/comercial/cliente/{idCliente}/dados?dataInicioVencFatura=01-07-2020&dataFimVencFatura=30-07-2020](https://[URL]/ws/comercial/cliente/{idCliente}/dados?dataInicioVencFatura=01-07-2020&dataFimVencFatura=30-07-2020)

### Método

GET

### Entrada

PathParam contendo ID do cliente, QueryParam("dataInicioVencFatura") (opcional) e QueryParam("dataFimVencFatura") (opcional) para limitar quais vencimentos de fatura trará.

**OBS.:** Caso não queira limitar os vencimentos das faturas na resposta, é só não enviar o parâmetro os parâmetros de data.

### Saída

JSON contendo os dados do cliente

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/cliente/81593/dados](https://[URL]/ws/comercial/cliente/81593/dados)

### Retorno:

```
{
  "TipoPessoa": "PessoaFisica",
  "ClienteEncontrado": true,
  "NomeCliente": "CLIENTE TESTE",
  "ExisteFaturaSemPagamento": true,
  "IDCliente": 81593,
  "Contratos": [
    {
      "ValorContrato": 105.49,
      "StatusContrato": "HABILITADO",
      "IDContrato": 1283299,
      "DescricaoPlano": "FIBRA RES 10MBPS FRANQUIA 360GB",
      "TipoContrato": "RESIDENCIAL",
      "TipoTecnologia": "INTERNET",
      "IDVencimento": 4
    },
    {
      "ValorContrato": 9.9,
      "StatusContrato": "HABILITADO EM CONFIANCA",
      "IDContrato": 1283303,
      "DescricaoPlano": "LOCACAO DE EQUIPAMENTOS",
      "TipoContrato": "RESIDENCIAL",
      "TipoTecnologia": "INTERNET",
      "IDVencimento": 4
    }
  ],
  "ExisteAtendimentoAberto": true,
  "Faturas": [
    {
      "idsContratos": [1283299,1283303],
      "DebitoAutomatico": false,
      "CodigoCliente": 81593,
      "ValorMulta": 0.0,
      "linhaDigitavel": "11111.11111 11111.11111 11111.11111 1 11111111111111",
      "IDFatura": 90,
      "ValorJuros": 0.0,
      "nroBoleto": "00000085425P",
      "Valor": 315.98,
      "nroDocumento": "0320283295",

```

**OU**

```
{ "ClienteEncontrato": false }
```

**OU**

"FALHA"

## Buscar dados do contrato e faturas correspondentes

### Descrição

Busca dados gerais dos contratos e faturas correspondentes

## URL

[https://\[URL\]/ws/comercial/contrato/dados](https://[URL]/ws/comercial/contrato/dados)

## Método

POST

## Entrada

JSON contendo o ID do contrato

## Saída

JSON contendo os dados do contrato

### Campos do JSON da entrada

Tag	Tipo	Observação
idContrato*	Long	

\*Campos obrigatórios

## Exemplo

**Envio:**

```
{
  "idContrato": 1283321
}
```

**Retorno:**

```
{  
    "ValorContrato": 66.99,  
    "StatusContrato": "HABILITADO",  
    "IDContrato": 1283321,  
    "DescricaoPlano": "FIBRA RES 2/6MBPS ILIM.",  
    "TipoContrato": "RESIDENCIAL",  
    "TipoTecnologia": "INTERNET",  
    "IDVencimento": 4  
    "Faturas": [  
        {  
            "idsContratos": [  
                1283322,  
                1283321  
            ],  
            "DebitoAutomatico": false,  
            "CodigoCliente": 81593,  
            "ValorMulta": 0.0,  
            "linhaDigitavel": "11111.11111 11111.111111 11111.111111 1 111111111111111",  
            "IDFatura": 90,  
            "ValorJuros": 0.0,  
            "nroBoleto": "000000085425P",  
            "Valor": 315.98,  
            "nroDocumento": "0320283295",  
            "codigoBarras": "1111111111111111111111111111111111111111",  
            "Vencimento": "2021-03-09"  
        }  
    ]  
}
```

## Fazer upload de arquivos para o cadastro de cliente

### Descrição

Faz o upload de uma lista de arquivos para o cadastro de um cliente

### URL

[https://\[URL\]/ws/comercial/clientes/anexar\\_arquivos](https://[URL]/ws/comercial/clientes/anexar_arquivos)

### Método

POST

### Entrada

HeaderParam("Token") com o value contendo um token válido, gerado pelo método "Gerar Token".

JSON contendo a lista de arquivos que deseja fazer o upload.

**OBS.:** a lista não deve conter mais que 5 arquivos e os arquivos (model) não podem ter tamanho maior que 5MB.

### Saída

String de "SUCESSO" ou o erro encontrado.

### Campos do JSON da entrada

Tag	Tipo	Observação
idCliente*	Long	
tipo*	String	Nesse campo só serao permitidos os seguintes valores, exatamente como descritos: COMPROVANTE_PAGAMENTO, DOCUMENTO, CONTRATO, ALTERACAO_TITULARIDADE, OUTROS
model*	Map<String,Object>	

\*Campos obrigatórios

**OBS.:** o Map **model** é onde serão enviados os dados do arquivo (imagem, pdf, txt, etc) em base64. Todos os campos do mesmo são obrigatórios e devem ser dos seguintes tipos:

base64 (String),  
filename (String),  
filesize (String),  
filetype (String)

### Exemplo

#### Envio:

```
[{
  "idCliente": 81593,
  "tipo": "DOCUMENTO",
  "model": {
    "base64": "QzpcVXNlcnNcUHJpc2NpbGxhXERvd25sb2Fkc1xURVNURV9TSU5URUdSQSAoMjMpL0cmli...",
    "filename": "foto.jpg",
    "filesize": 106743,
    "filetype": "image/jpeg"
  },
},
{
  "idCliente": 81593,
  "tipo": "CONTRATO",
  "model": {
    "base64": "QzpcVXNlcnNcUHJpc2NpbGxhXERvd25sb2Fkc1xURVNURV9TSU5URUdS4dDtDb250cmli...",
    "filename": "cac.pdf",
    "filesize": 106743,
    "filetype": "application/pdf"
  }
}]
```

#### Retorno:

"SUCESSO"

#### ou

```
{
  "error": "O tamanho da imagem foto.jpg excede o limite de 5MB."
}
```

ou

```
{  
  "error": "A quantidade de arquivos excede o número máximo de 5."  
}
```

## Baixar PDF do modelo do contrato

### Descrição

Faz download do Modelo do contrato informado

### URL

[https://\[URL\]/ws/comercial/contratos/modelo/{idContrato}](https://[URL]/ws/comercial/contratos/modelo/{idContrato})

### Método

GET

### Entrada

Pelo PathParam passar o código do contrato,  
E pelo Header o Token gerado no método Gera Token

### Saída

PDF com o modelo de aceite de contrato.

### Exemplo

### Retorno:

PDF com o modelo de aceite de contrato.



## Busca Clientes para PlayHub

### Descrição

Busca os clientes para Integração com a PlayHub de acordo com a Data de Habilitação.

### URL

[https://:\[URL\]/ws/playhub/data](https://:[URL]/ws/playhub/data)

### Método

POST

### Entrada

Header: Name: Token, Value: Token gerado pela api de Gerar Token  
JSON contendo usuário, senha e a Data de Habilitação dos Contratos

### Saída

JSON contendo informações do cliente para a PlayHub

### Formato da Entrada

Tag	Tipo	Observação
Usuario *	String	
Senha *	String	
DataHabilitacao *	String	Padrão: yyyy-mm-dd

\* Todos os itens são obrigatórios

### Exemplo

#### Envio:

```
{
  "Usuario": "12312312312",
  "Senha": "12312312312",
  "DataHabilitacao": "2018-08-20"
}
```

#### Retorno:

```
{
  "Clientes": [
    {
      "Celular": "33999777888",
      "FoxEsport": false,
      "Email": "teste@yahoo.com.br",
      "uol": false,
      "WatchTv": true,
      "CPF_CNPJ": "111111111111",
      "EsporteIterativo": false,
      "McAfee": true
    },
    {
      "Celular": "33999777888",
      "FoxEsport": false,
      "Email": "teste@yahoo.com.br",
      "uol": true,
      "WatchTv": false,
      "CPF_CNPJ": "11111111111",
      "EsporteIterativo": true,
      "McAfee": false
    }
  ]
}
```

## Busca Cliente por CPF/CNPJ para PlayHub

### Descrição

Busca o cliente para Integração com a PlayHub de acordo com o CPF ou CNPJ.

### URL

[https:// ://\[URL\]/ws/comercial/playhub/cpf\\_cnpj](https:// ://[URL]/ws/comercial/playhub/cpf_cnpj)

### Método

POST

### Entrada

Header: Name: Token, Value: Token gerado pela api de Gerar Token

JSON contendo usuário, senha e o CPF ou CNPJ

### Saída

JSON contendo o status da edição (sucesso/erro)

### Formato da Entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	
CPF_CNPJ *	String	Sem ponto e traço.

\* Todos os itens são obrigatórios

### Exemplo

#### Envio:

```
{
  "Usuario": "teste.teste",
  "Senha": "testel23",
  "CNPJ": "12345678"
}
```

#### Retorno:

```
{
  "Cliente":
  {
    "Celular": "33999777888",
    "FoxEsport": false,
    "Email": "teste@yahoo.com.br",
    "uol": false,
    "WatchTv": true,
    "CPF_CNPJ": "111111111111",
    "EsporteIterativo": false,
    "McAfee": true
  }
}
```

## Listar contratos para inativação temporária

### Descrição

Busca informações de contratos habilitados para realizar a inativação temporária

### URL

https://{{url}}/ws/comercial/cliente/{{idCliente}}/lista/contratos/inativar/temporariamente

### Método

GET

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

PathParam: idCliente

### Saída

List de JSON contendo os dados dos contratos.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idCliente*	path	Number	

\* Parâmetro obrigatório

### Exemplo retorno:

```
[
  {
    "tecnologia": "FIBRA",
    "possuiPendenciaFinanceira": "SIM",
    "valor": 105.49,
    "plano": "FIBRA RES 10MBPS FRANQUIA 360GB",
    "inativadoMenosUmAno": "SIM",
    "idContrato": 1283294,
    "status": "HABILITADO"
  },
  {
    "tecnologia": "CONSULTORIA",
    "possuiPendenciaFinanceira": "SIM",
    "valor": 55.0,
    "plano": "ASASAS",
    "inativadoMenosUmAno": "NAO",
    "idContrato": 1283392,
    "status": "HABILITADO"
  },
  {
    "tecnologia": "RADIO",
    "possuiPendenciaFinanceira": "NAO",
    "valor": 99.9,
    "plano": "WIRELESS VEROSUPER 2M",
    "inativadoMenosUmAno": "SIM",
    "idContrato": 1283459,
    "status": "HABILITADO"
  },
  {
    "tecnologia": "RADIO",
    "possuiPendenciaFinanceira": "NAO",
    "valor": 99.9,
    "plano": "WIRELESS VEROSUPER 2M",
    "inativadoMenosUmAno": "NAO",
    "idContrato": 1283460,
    "status": "HABILITADO"
  }
]
```

## Inativar contratos temporariamente

### Descrição

Inativa temporariamente um contrato

### URL

https://{{url}}/ws/comercial/contratos/inativar/temporariamente

### Método

PUT

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

Json contendo os dados para inativação

### Saída

String com os valores SUCESSO ou FALHA.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idContrato*	body	Number	
quantidadeDias*	body	Number	
dataInicio*	body	String	Formato: dd-MM-yyyy

\* Parâmetro obrigatório

### Exemplo entrada:

```
{
  "idContrato": 1283519,
  "dataInicio": "05-04-2021",
  "quantidadeDias": 60
}
```

### Exemplo retorno:

Em caso de sucesso retornará a string: "SUCESSO"

Em caso de falha retornará a string: "FALHA"

## Buscar tipos de contrato

### Descrição

Busca todos os tipos de contratos

### URL

[https://\[URL\]/ws/comercial/contratos/tipos](https://[URL]/ws/comercial/contratos/tipos)

### Método

GET

### Entrada

Header: *Token* `{{token}}`, *Authorization Basic* `{{authorization}}`

### Saída

JSON contendo lista de tipos de contrato ou JSON contendo mensagem de erro

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/contratos/tipos](https://[URL]/ws/comercial/contratos/tipos)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "nome": "RESIDENCIAL"
    },
    {
      "id": 2,
      "nome": "CORPORATIVO"
    },
    {
      "id": 3,
      "nome": "LICITAÇÃO"
    },
    {
      "id": 4,
      "nome": "OUTROS"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar tecnologias

### Descrição

Busca todas as tecnologias

### URL

[https://\[URL\]/ws/comercial/tecnologias](https://[URL]/ws/comercial/tecnologias)

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

### Saída

JSON contendo lista de tecnologias ou JSON contendo mensagem de erro

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/tecnologias](https://[URL]/ws/comercial/tecnologias)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 2,
      "descricao": "RADIO",
      "tipo": "INTERNET"
    },
    {
      "id": 4,
      "descricao": "FIBRA",
      "tipo": "INTERNET"
    },
    {
      "id": 10,
      "descricao": "TELEFONIA IP",
      "tipo": "TELEFONIA"
    },
    {
      "id": 11,
      "descricao": "TELEFONIA FIXA",
      "tipo": "TELEFONIA"
    },
    {
      "id": 12,
      "descricao": "TV POR SATELITE",
      "tipo": "TV"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar canais de venda ativos

### Descrição

Busca todos os canais de venda ativos

### URL

[https://\[URL\]/ws/comercial/canais/venda/ativos](https://[URL]/ws/comercial/canais/venda/ativos)

### Método

GET

### Entrada

Header: *Token* `{{token}}`, *Authorization Basic* `{{authorization}}`

### Saída

JSON contendo lista de canais de venda ativos ou JSON contendo mensagem de erro

### Exemplo

#### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "descricao": "A",
      "descricoesRD": null,
      "permiteQualityCall": false,
      "ativo": true
    },
    {
      "id": 2,
      "descricao": "B",
      "descricoesRD": null,
      "permiteQualityCall": false,
      "ativo": true
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Informações da Última Conexão de Equipamento

### Descrição

Busca informações da última conexão dos equipamentos dos contratos de internet de clientes com status diferente de "CANCELADO", de acordo com os filtros passados por parâmetro.

### URL

https://{{url}}/ws/area/cliente/{idCliente: [0-9]+}/equipamentos/ultima\_conexao

### Método

GET

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

PathParam: *idCliente*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idCliente*	path	Number	

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": [
    {
      "senhaWiFi": null,
      "fornecedorEquipamento": null,
      "criptografiaRede": null,
      "infoUltimaConexao": {},
      "serial": null,
      "loginWiFi": null
    },
    {
      "senhaWiFi": null,
      "fornecedorEquipamento": null,
      "criptografiaRede": null,
      "infoUltimaConexao": {
        "acctuniqueid": "e02aa96a3e91e7afb17789bdc42559da",
        "framedprotocol": "PPP",
        "connectinfo_start": "",
        "nasipaddress": "201.33.240.4",
        "realm": "",
        "acctinterval": null,
        "acctupdatetime": 1493331094000,
        "connectinfo_stop": "",
        "groupname": "",
        "callingstationid": "74:CC:39:87:25:93",
        "acctsessionid": "8192f896",
        "nasporttype": "Ethernet",
        "radacctid": 2,
        "username": "SAL.farinha.311622",
        "calledstationid": "PPPoE1-Tag",
        "acctoutputoctets": 238,
        "acctstarttime": 1493331094000,
        "acctstoptime": null,
        "acctsessiontime": 24,
        "nasportid": "0 0/0/0:4096.1003 LAFAIETE-4/0/0/11/0/1/FHTT-09872590 GP"
      }
    }
  ],
}
```



```
        "acctterminatecause": "User-Request",
        "acctinputoctets": 908,
        "acctauthentic": "RADIUS",
        "servicetype": "Framed-User",
        "framedipaddress": "172.16.20.252"
    },
    "serial": null,
    "loginWiFi": null
}
],
"status": "OK"
}
```

## Disponibilização dos Dias Disponíveis para Alteração de Vencimento de Contratos

### Descrição

Busca informações dos vencimentos ativos presentes no sistema para alteração de contratos.

### URL

https://{{url}}/ws/area/cliente/vencimentos/ativos

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

QueryParam: *isPrePago*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
isPrePago*	query	String	Valores permitidos: "s" e "n".

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": [
    {
      "id": 2,
      "dia": 5,
      "diaFaturamentoInicial": 6,
      "diaFaturamentoFinal": 5,
      "ativo": true,
      "prePago": false,
      "faturaMesmoMesVencimento": true
    },
    {
      "id": 4,
      "dia": 10,
      "diaFaturamentoInicial": 11,
      "diaFaturamentoFinal": 10,
      "ativo": true,
      "prePago": false,
      "faturaMesmoMesVencimento": true
    }
  ],
  "status": "OK"
}
```

## Alteração de Dia de Vencimento de Contratos

### Descrição

Realiza a alteração do dia de vencimento das faturas de um contrato de acordo com as informações passadas por parâmetro.

### URL

https://{{url}}/ws/area/cliente/contrato/alterar/vencimento\_contrato

### Método

PUT

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

Body: *idContrato, idVencimentoAtual, idVencimentoNovo*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idContrato*	body	Number	
idVencimentoAtual*	body	Number	
idVencimentoNovo*	body	Number	

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": "A data de vencimento foi alterada do ID 'X' para o ID 'Y' no(s) contrato(s)
) selecionado(s) com sucesso.",
  "status": "OK"
}
```

## Buscar promoções ativas de acordo com parâmetros

### Descrição

Buscar promoções ativas de acordo com parâmetros

### URL

[https://\[URL\]/ws/comercial/contratos/promoco/es/ativas](https://[URL]/ws/comercial/contratos/promoco/es/ativas)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Lista de Json contendo os dados necessários para a busca

### Saída

JSON contendo as promoções

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idFormaCobranca*	body	Number	
idTipoContrato*	body	Number	
idCidade*	body	Number	
idBairro*	body	Number	
idPerfil*	body	Number	IDPerfil do usuário (vendedor)
estado*	body	String	
cep*	body	String	Valores: PADRAO, FISCAL, COBRANCA, INSTALACAO
prePago*	body	Number	Tipo da forma de cobrança. 1 = SIM / 2 = NAO
tipoPessoa*	body	String	Valores: PessoaFisica, PessoaJuridica
idPlano	body	Number	
idTecnologia	body	String	Obrigatório se a informação idPlano for preenchida
idPacote	body	String	
idsTecnologias	body	String	Obrigatório se a informação idPacote for preenchida. Deverá ser preenchido com os ids das tecnologias de cada plano do pacote separados por '-'

**\* Parâmetro obrigatório**

**OBS.:** O JSON deverá conter obrigatoriamente o campo IDPlano ou IDPacote preenchido.

Os campos não obrigatórios quando não possuem valor, podem ser enviados no JSON com vazio ("") ou nem serem enviados no JSON

### Exemplo entrada:

#### PLANO:

```
{
  "idFormaCobranca": 65,
  "idPlano": 787,
  "idTipoContrato": 1,
  "idCidade": 100,
```

```

    "idBairro": 15372,
    "idPerfil": 1,
    "estado": "MG",
    "cep": "36402-157",
    "prePago": 0,
    "tipoPessoa": "PessoaFisica",
    "idTecnologia": 4
  }
}

```

#### **PACOTE:**

```

{
  "idFormaCobranca": 65,
  "idTipoContrato": 1,
  "idCidade": 100,
  "idBairro": 15372,
  "idPerfil": 1,
  "estado": "MG",
  "cep": "36402-157",
  "prePago": 0,
  "tipoPessoa": "PessoaFisica",
  "idsTecnologias": "4-14",
  "idPacote": 81
}

```

#### **Exemplo retorno:**

##### **Sucesso:**

```

{
  "resposta": [
    {
      "TipoDesconto": "MENSALIDADE",
      "Descricao": "DESCONTO DE 100% NA PRIMEIRA MENSALIDADE DE FIBRA ÓPTICA",
      "Desconto": 100.0,
      "NumeroMensalidades": 1,
      "Id": 41,
      "DescontoPorcentagem": true
    },
    {
      "TipoDesconto": "MENSALIDADE",
      "Descricao": "30% DESCONTO NA MENSALIDADE 12 MESES FUNCIONARIO",
      "Desconto": 30.0,
      "NumeroMensalidades": 12,
      "Id": 283,
      "DescontoPorcentagem": true
    },
    {
      "TipoDesconto": "MENSALIDADE",
      "Descricao": "R$5,00 BOLETO DIGITAL",
      "Desconto": 5.0,
      "NumeroMensalidades": 24,
      "Id": 306,
      "DescontoPorcentagem": false
    }
  ]
}

```

##### **Falha:**

```

{
  "error": "Internal Server Error"
}

```

## Buscar planejamentos tributários de acordo com a empresa e o plano

### Descrição

Busca os planejamentos tributários de acordo com empresa e plano

### URL

[https://\[URL\]/ws/comercial/contratos/planejamentos\\_tributarios/plano/{idPlano}/empresa/{idEmpresa}](https://[URL]/ws/comercial/contratos/planejamentos_tributarios/plano/{idPlano}/empresa/{idEmpresa})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idPlano, idEmpresa*

### Saída

JSON contendo planejamentos tributários

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/contratos/planejamentos\\_tributarios/plano/787/empresa/1](https://[URL]/ws/comercial/contratos/planejamentos_tributarios/plano/787/empresa/1)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "Descricao": "PLANEJAMENTO 1",
      "ServicoInstalacao": {
        "Descricao": "CL - FIBRA - ALTERAÇÃO DE ENDEREÇO",
        "NumeroMaximoParcelas": 10,
        "Id": 6,
        "PermiteParcela": true,
        "Valor": 200.0
      },
      "Id": 1
    },
    {
      "Descricao": "PLANEJAMENTO 2",
      "ServicoInstalacao": null,
      "Id": 3
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar velocidades dos planos de tabela de preço de acordo com parâmetros

### Descrição

Buscar velocidades dos planos de tabela de preço de acordo com parâmetros

### URL

[https://\[URL\]/ws/comercial/contratos/tabela/precos/velocidades](https://[URL]/ws/comercial/contratos/tabela/precos/velocidades)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Lista de Json contendo os dados necessários para a busca

### Saída

JSON contendo o as velocidades

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idPlano*	body	Number	
idTipoContrato*	body	Number	
idAreaCobertura*	body	Number	
tipoPessoa*	body	String	Valores: PessoaFisica, PessoaJuridica

**\* Parâmetro obrigatório**

### Exemplo entrada:

```
{
  "idPlano": 8,
  "idTipoContrato": 1,
  "idAreaCobertura": 2150,
  "tipoPessoa": "PessoaFisica"
}
```

### Exemplo retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 67,
      "descricao": "600/800K Residencial",
      "taxaDownloadGenerico": "600",
      "taxaUploadGenerico": "300",
      "velocidadesConexaoRadius": null
    },
    {
      "id": 4,
      "descricao": "2MB RESIDENCIAL",
      "taxaDownloadGenerico": "2048",
      "taxaUploadGenerico": "2048",
      "velocidadesConexaoRadius": null
    }
  ]
}
```

#### Falha:

```
{
```

```
    "error": "Internal Server Error"  
}
```



## Buscar área de cobertura por cidade

### Descrição

Busca área de cobertura por cidade

### URL

[https://\[URL\]/ws/comercial/contratos/areas\\_cobertura/cidade/{idCidade}](https://[URL]/ws/comercial/contratos/areas_cobertura/cidade/{idCidade})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idCidade*

### Saída

JSON contendo a área de cobertura

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/contratos/areas\\_cobertura/cidade/100](https://[URL]/ws/comercial/contratos/areas_cobertura/cidade/100)

### Retorno:

#### Sucesso:

```
{
  "resposta": {
    "Descricao": "MG",
    "Id": 2124
  }
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar valores dos planos de tabela de preço de acordo com parâmetros

### Descrição

Buscar valores dos planos de tabela de preço de acordo com parâmetros

### URL

[https://\[URL\]/ws/comercial/contratos/tabela/precos/valores](https://[URL]/ws/comercial/contratos/tabela/precos/valores)

### Método

POST

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

Body: Lista de Json contendo os dados necessários para a busca

### Saída

JSON contendo os valores

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idPlano*	body	Number	
idTipoContrato*	body	Number	
idAreaCobertura*	body	Number	
tipoPessoa*	body	String	Valores: PessoaFisica, PessoaJuridica
idVelocidade	Body	Number	Obrigatório quando o plano for de internet

**\* Parâmetro obrigatório**

### Exemplo entrada:

```
{
  "idPlano": 8,
  "idTipoContrato": 1,
  "idAreaCobertura": 2150,
  "tipoPessoa": "PessoaFisica",
  "idVelocidade": 67
}
```

### Exemplo retorno:

#### Sucesso:

```
{
  "resposta": {
    "Velocidade": "600/800K Residencial",
    "Valor": 80.0,
    "Plano": "VERO 600KBPS ILIM."
  }
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar pacotes de voz para venda extra

### Descrição

Busca os pacotes de voz disponíveis para venda extra no contrato

### URL

[https://\[URL\]/ws/comercial/contratos/pacotes\\_voz/extras](https://[URL]/ws/comercial/contratos/pacotes_voz/extras)

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

### Saída

JSON contendo lista dos pacotes de voz ou JSON contendo mensagem de erro

### Exemplo

Envio: [https://\[URL\]/ws/comercial/contratos/pacotes\\_voz/extras](https://[URL]/ws/comercial/contratos/pacotes_voz/extras)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "descricao": "PACOTE VOZ 1",
      "valor": 50.0,
      "franquias": [
        {
          "id": 1,
          "descricao": "100 LIVRES FIXO DDD ",
          "tipoLigacao": {
            "id": 2,
            "descricao": "FIXO DDD",
            "idServico": 135
          },
          "minutos": 100
        },
        {
          "id": 2,
          "descricao": "100 LIVRES FIXO LOCAL",
          "tipoLigacao": {
            "id": 1,
            "descricao": "FIXO LOCAL",
            "idServico": 134
          },
          "minutos": 100
        }
      ]
    },
    {
      "id": 2,
      "descricao": "PACOTE VOZ 2",
      "valor": 20.0,
      "franquias": [
        {
          "id": 3,
          "descricao": "200 LIVRES FIXO LOCAL",
          "tipoLigacao": {
            "id": 1,
            "descricao": "FIXO LOCAL",
            "idServico": 134
          },
          "minutos": 200
        }
      ]
    }
  ]
}
```

```

    },
    {
      "id": 4,
      "descricao": "300 LIVRES FIXO DDD ",
      "tipoLigacao": {
        "id": 2,
        "descricao": "FIXO DDD",
        "idServico": 135
      },
      "minutos": 300
    }
  ]
}

```

**Falha:**

```

{
  "error": "Internal Server Error"
}

```

## Buscar canais de TV para venda extra

### Descrição

Busca os canais de TV disponíveis para venda extra no contrato

### URL

[https://\[URL\]/ws/comercial/contratos/canais\\_tv/extras](https://[URL]/ws/comercial/contratos/canais_tv/extras)

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

### Saída

JSON contendo lista dos canais de TV ou JSON contendo mensagem de erro

### Exemplo

Envio: [https://\[URL\]/ws/comercial/contratos/canais\\_tv/extras](https://[URL]/ws/comercial/contratos/canais_tv/extras)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "descricao": "ION TV",
      "valor": 0.0,
      "idCanalIntegracaoMinerva": null,
      "permiteALaCarte": true
    },
    {
      "id": 2,
      "descricao": "TV ALTEROSA BH SD",
      "valor": 10.0,
      "idCanalIntegracaoMinerva": null,
      "permiteALaCarte": true
    },
    {
      "id": 5,
      "descricao": "REDE TV! HD",
      "valor": 0.0,
      "idCanalIntegracaoMinerva": null,
      "permiteALaCarte": true
    },
    {
      "id": 6,
      "descricao": "GLOBO MINAS HD",
      "valor": 0.0,
      "idCanalIntegracaoMinerva": null,
      "permiteALaCarte": true
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar pacotes de canais de TV para venda extra

### Descrição

Busca os pacotes de canais de TV disponíveis para venda extra no contrato

### URL

[https://\[URL\]/ws/comercial/contratos/pacotes\\_tv/extras](https://[URL]/ws/comercial/contratos/pacotes_tv/extras)

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

### Saída

JSON contendo lista dos pacotes de canais de TV ou JSON contendo mensagem de erro

### Exemplo

Envio: [https://\[URL\]/ws/comercial/contratos/pacotes\\_tv/extras](https://[URL]/ws/comercial/contratos/pacotes_tv/extras)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 3,
      "descricao": "TELECINE HD",
      "valor": 44.9,
      "canais": [
        {
          "id": 43,
          "descricao": "CURTA",
          "valor": 0.0,
          "idCanalIntegracaoMinerva": null,
          "permiteALaCarte": null
        },
        {
          "id": 45,
          "descricao": "MIX TV",
          "valor": 0.0,
          "idCanalIntegracaoMinerva": null,
          "permiteALaCarte": null
        }
      ],
      "idPacoteIntegracaoMinerva": null
    },
    {
      "id": 4,
      "descricao": "PREMIERE FC",
      "valor": 74.9,
      "canais": [
        {
          "id": 14,
          "descricao": "REDE MINAS",
          "valor": 0.0,
          "idCanalIntegracaoMinerva": null,
          "permiteALaCarte": null
        }
      ],
      "idPacoteIntegracaoMinerva": null
    }
  ]
}
```

```
]
}
```

**Falha:**

```
{
  "error": "Internal Server Error"
}
```

## Cadastrar contrato de INTERNET

### Descrição

Cadastrar contrato de internet

### URL

[https://\[URL\]/ws/comercial/contratos/cadastrar](https://[URL]/ws/comercial/contratos/cadastrar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados necessários para o cadastro

### Saída

JSON contendo os dados do contrato cadastrado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Long	
idPlano*	body	Long	
idCliente*	body	Long	
valorContrato*	body	Double	O valor do contrato deverá ser somente o valor do plano ou o valor cadastrado na tabela de preço.
idTipoContrato*	body	Long	
idsEnderecos*	body	List<Long>	A lista deverá ser preenchida com o IDEnderecoCliente. É obrigatório pelo menos um do tipo PADRAO ou um do tipo INSTALACAO e um do tipo COBRANCA
idOrigemComercial*	body	Long	
idVencimento*	body	Long	
idFormaCobranca*	body	Long	
idCanalVenda*	body	Long	
podeSuspender*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
naoCancelaPorInadimplencia*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
fidelizacao*	body	String	Valores: MESES_24, SEM, MESES_36, MESES_12, MESES_4
idPlanejamentoTributario	body	Long	Obrigatório se o plano tiver o atributo <b>PossuiPlanejamentoTributario = true</b>
idDadosDebito	body	Long	Obrigatório caso a forma de cobrança seja do tipo débito automático. Deverá ser informado o



			IDDadosDebitoCliente.
idsPromocoões	body	List<String>	
observacaoFiscal	body	String	
observacaoFatura	body	String	
idServicoInstalacao	body	Long	Obrigatório se possuir taxa de instalação
parcelasContratadasServicoInstalacao	body	int	Obrigatório se possuir taxa de instalação. Se não for informado, será considerado 1
idPlanoProspectado	body	Long	Informar quando o contrato for originado de um serviço de interesse do prospecto (plano prospectado)
loginRadius	body	String	Obrigatório para planos que não geram login automaticamente
senhaRadius	body	String	Obrigatório para planos que não geram senha automaticamente
idSaidaCaixa	body	Long	Obrigatório se for fibra e se o plano tiver o atributo <b>reservaSaida = true</b>
idCaixa	body	Long	Obrigatório se for fibra e se o plano tiver o atributo <b>reservaSaida = true</b>
idLocationCM	body	Long	Obrigatório se for fibra, se o plano tiver o atributo <b>reservaSaida = true</b> e se a empresa possuir integração com ConnectMaster
idVelocidade	body	Long	Obrigatório para planos que possuem o atributo <b>UtilizaTabelaPreco = true</b>

\* Parâmetro obrigatório

**OBS.:** os campos não obrigatórios, podem ser enviados no JSON com vazio ("") ou nem serem enviados no JSON

#### Exemplo entrada:

```
{
  "idPlano": 1159,
  "idTipoContrato": 1,
  "idCliente": 81593,
  "idUsuario": 285,
  "idsEnderecos": [481018],
  "idVencimento": 1,
  "idFormaCobranca": 65,
  "valorContrato": 83.1,
  "idOrigemComercial": 1,
  "podeSuspender": 1,
  "idCanalVenda": 1,
  "naoCancelaPorInadimplencia": 0,
  "fidelizacao": "MESES_24",
  "idPlanejamentoTributario": 1,
  "idServicoInstalacao": 17,
  "parcelasContratadasServicoInstalacao": 2,
  "idSaidaCaixa": 225,
  "idCaixa": 24,
  "idLocationCM": 1,
}
```

```
"idVelocidade": 1,  
"idsPromocoes": ["20", "21"]  
}
```

**Exemplo retorno:****Sucesso:**

```
{  
  "resposta": {  
    "cliente": "81593 - CLIENTE TESTE",  
    "valorContrato": 83.1,  
    "tecnologia": "FIBRA",  
    "plano": "1159 - FIBRA MAIS - 15M",  
    "id": 1283603,  
    "status": "AGUARDANDO INSTALACAO"  
  }  
}
```

**Falha:**

```
{  
  "error": "O idCliente informado não pertence a nenhum cliente cadastrado no  
sistema."  
}
```

## Cadastrar contrato de TELEFONIA

### Descrição

Cadastrar contrato de telefonia

### URL

[https://\[URL\]/ws/comercial/contratos/cadastrar](https://[URL]/ws/comercial/contratos/cadastrar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados necessários para o cadastro

### Saída

JSON contendo os dados do contrato cadastrado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Long	
idPlano*	body	Long	
idCliente*	body	Long	
valorContrato*	body	Double	O valor do contrato deverá ser somente o valor do plano ou o valor cadastrado na tabela de preço.
idTipoContrato*	body	Long	
idsEnderecos*	body	List<Long>	A lista deverá ser preenchida com o IDEnderecoCliente. É obrigatório pelo menos um do tipo PADRAO ou um do tipo INSTALACAO e um do tipo COBRANCA
idOrigemComercial*	body	Long	
idVencimento*	body	Long	
idFormaCobranca*	body	Long	
idCanalVenda*	body	Long	
podeSuspender*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
naoCancelaPorInadimplencia*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
fidelizacao*	body	String	Valores: MESES_24, SEM, MESES_36, MESES_12, MESES_4
idPlanejamentoTributario	body	Long	Obrigatório se o plano tiver o atributo <b>PossuiPlanejamentoTributario = true</b>
idDadosDebito	body	Long	Obrigatório caso a forma de cobrança seja do tipo débito automático. Deverá ser

			informado o IDDadosDebitoCliente.
idsPromocoies	body	List<String>	
observacaoFiscal	body	String	
observacaoFatura	body	String	
idServicoInstalacao	body	Long	Obrigatório se possuir taxa de instalação
parcelasContratadasServicoInstalacao	body	int	Obrigatório se possuir taxa de instalação. Se não for informado, será considerado 1
idPlanoProspectado	body	Long	Informar quando o contrato for originado de um serviço de interesse do prospecto (plano prospectado)
idNumero*	body	Long	
bloquearLigacaoCobrar*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
bloquearLigacaoDDD*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
bloquearLigacaoCelular*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
bloquearLigacaoInternacional*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
exibeListaTelefonica*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
idsPacotesVozExtra	body	List<String>	

**\* Parâmetro obrigatório**

**OBS.:** os campos não obrigatórios, podem ser enviados no JSON com vazio ("") ou nem serem enviados no JSON

#### Exemplo entrada:

```
{
  "idPlano": 673,
  "idTipoContrato": 1,
  "idCliente": 81593,
  "idUsuario": 285,
  "idsEnderecos": [481018],
  "idVencimento": 1,
  "idFormaCobranca": 65,
  "valorContrato": 83.1,
  "idOrigemComercial": 1,
  "podeSuspende": 1,
  "idCanalVenda": 1,
  "naoCancelaPorInadimplencia": 0,
  "fidelizacao": "MESES_24",
  "idPlanejamentoTributario": 1,
  "idServicoInstalacao": 17,
  "parcelasContratadasServicoInstalacao": 2,
  "idsPromocoies": ["20", "21"]
  "idNumero": 178,
  "bloquearLigacaoCobrar": 1,
  "bloquearLigacaoDDD": 0,
  "bloquearLigacaoCelular": 0,
  "bloquearLigacaoInternacional": 1,
  "exibeListaTelefonica": 0,
  "idsPacotesVozExtra": ["1"]
}
```

**Exemplo retorno:****Sucesso:**

```
{
  "resposta": {
    "cliente": "81593 - CLIENTE TESTE",
    "valorContrato": 133.1,
    "tecnologia": "TELEFONIA FIXA",
    "plano": "673 - FALA BEM ILIM",
    "id": 1283604,
    "status": "AGUARDANDO INSTALACAO"
  }
}
```

**Falha:**

```
{
  "error": "O idCliente informado não pertence a nenhum cliente cadastrado no sistema."
}
```

## Cadastrar contrato de TV

### Descrição

Cadastrar contrato de telefonia

### URL

[https://\[URL\]/ws/comercial/contratos/cadastrar](https://[URL]/ws/comercial/contratos/cadastrar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados necessários para o cadastro

### Saída

JSON contendo os dados do contrato cadastrado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Long	
idPlano*	body	Long	
idCliente*	body	Long	
valorContrato*	body	Double	O valor do contrato deverá ser somente o valor do plano ou o valor cadastrado na tabela de preço.
idTipoContrato*	body	Long	
idsEnderecos*	body	List<Long>	A lista deverá ser preenchida com o IDEnderecoCliente. É obrigatório pelo menos um do tipo PADRAO ou um do tipo INSTALACAO e um do tipo COBRANCA
idOrigemComercial*	body	Long	
idVencimento*	body	Long	
idFormaCobranca*	body	Long	
idCanalVenda*	body	Long	
podeSuspender*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
naoCancelaPorInadimplencia*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
fidelizacao*	body	String	Valores: MESES_24, SEM, MESES_36, MESES_12, MESES_4
idPlanejamentoTributario	body	Long	Obrigatório se o plano tiver o atributo <b>PossuiPlanejamentoTributario = true</b>
idDadosDebito	body	Long	Obrigatório caso a forma de cobrança seja do tipo débito automático. Deverá ser

			informado o IDDadosDebitoCliente.
idsPromoco	body	List<String>	
observacaoFiscal	body	String	
observacaoFatura	body	String	
idServicoInstalacao	body	Long	Obrigatório se possuir taxa de instalação
parcelasContratadasServicoInstalacao	body	int	Obrigatório se possuir taxa de instalação. Se não for informado, será considerado 1
idPlanoProspectado	body	Long	Informar quando o contrato for originado de um serviço de interesse do prospecto (plano prospectado)
idsCanaisExtras	body	List<String>	
idsPacotesCanaisExtras	body	List<String>	

**\* Parâmetro obrigatório**

**OBS.:** os campos não obrigatórios, podem ser enviados no JSON com vazio ("") ou nem serem enviados no JSON

**Exemplo entrada:**

```
{
  "idPlano": 692,
  "idTipoContrato": 1,
  "idCliente": 81593,
  "idUsuario": 285,
  "idsEnderecos": [481018],
  "idVencimento": 1,
  "idFormaCobranca": 65,
  "valorContrato": 100.0,
  "idOrigemComercial": 1,
  "podeSuspender": 1,
  "idCanalVenda": 1,
  "naoCancelaPorInadimplencia": 0,
  "fidelizacao": "MESES_24",
  "idPlanejamentoTributario": 1,
  "idServicoInstalacao": 17,
  "parcelasContratadasServicoInstalacao": 2,
  "idsPromoco": ["20", "21"],
  "idsCanaisExtras": ["2", "5"],
  "idsPacotesCanaisExtras": ["3"]
}
```

**Exemplo retorno:**

**Sucesso:**

```
{
  "resposta": {
    "cliente": "81593 - CLIENTE TESTE",
    "valorContrato": 154.9,
    "tecnologia": "TV POR SATELITE",
    "plano": "692 - TV BRONZE HD",
    "id": 1283605,
    "status": "AGUARDANDO INSTALACAO"
  }
}
```

**Falha:**

```
{
  "error": "O idCliente informado não pertence a nenhum cliente cadastrado no sistema."
}
```

## Cadastrar contrato de demais tecnologias

### Descrição

Cadastrar contrato de telefonia

### URL

[https://\[URL\]/ws/comercial/contratos/cadastrar](https://[URL]/ws/comercial/contratos/cadastrar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados necessários para o cadastro

### Saída

JSON contendo os dados do contrato cadastrado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Long	
idPlano*	body	Long	
idCliente*	body	Long	
valorContrato*	body	Double	O valor do contrato deverá ser somente o valor do plano ou o valor cadastrado na tabela de preço.
idTipoContrato*	body	Long	
idsEnderecos*	body	List<Long>	A lista deverá ser preenchida com o IDEnderecoCliente. É obrigatório pelo menos um do tipo PADRAO ou um do tipo INSTALACAO e um do tipo COBRANCA
idOrigemComercial*	body	Long	
idVencimento*	body	Long	
idFormaCobranca*	body	Long	
idCanalVenda*	body	Long	
podeSuspender*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
naoCancelaPorInadimplencia*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
fidelizacao*	body	String	Valores: MESES_24, SEM, MESES_36, MESES_12, MESES_4
idPlanejamentoTributario	body	Long	Obrigatório se o plano tiver o atributo <b>PossuiPlanejamentoTributario = true</b>
idDadosDebito	body	Long	Obrigatório caso a forma de cobrança seja do tipo débito automático. Deverá ser



			informado o IDDadosDebitoCliente.
idsPromocoies	body	List<String>	
observacaoFiscal	body	String	
observacaoFatura	body	String	
idServicoInstalacao	body	Long	Obrigatório se possuir taxa de instalação
parcelasContratadasServicoInstalacao	body	int	Obrigatório se possuir taxa de instalação. Se não for informado, será considerado 1
idPlanoProspectado	body	Long	Informar quando o contrato for originado de um serviço de interesse do prospecto (plano prospectado)

**\* Parâmetro obrigatório**

**OBS.:** os campos não obrigatórios, podem ser enviados no JSON com vazio ("") ou nem serem enviados no JSON

**Exemplo entrada:**

```
{
  "idPlano": 786,
  "idTipoContrato": 1,
  "idCliente": 81593,
  "idUsuario": 285,
  "idsEnderecos": [481018],
  "idVencimento": 1,
  "idFormaCobranca": 65,
  "valorContrato": 100.0,
  "idOrigemComercial": 1,
  "podeSuspender": 1,
  "idCanalVenda": 1,
  "naoCancelaPorInadimplencia": 0,
  "fidelizacao": "MESES_12",
  "idServicoInstalacao": 17,
  "parcelasContratadasServicoInstalacao": 1,
  "idPlanoProspectado": 2366
}
```

**Exemplo retorno:**

**Sucesso:**

```
{
  "resposta": {
    "cliente": "81593 - CLIENTE TESTE",
    "valorContrato": 100.0,
    "tecnologia": "LOCACAO",
    "plano": "786 - LOCACAO WI-FI",
    "id": 1283606,
    "status": "HABILITADO"
  }
}
```

**Falha:**

```
{
  "error": "O idCliente informado não pertence a nenhum cliente cadastrado no sistema."
}
```

## Cadastrar pacotes de contratos

### Descrição

Cadastrar pacotes de contrato

OBS.: O JSON enviado para cadastro de pacote é o mesmo para cadastro de contrato único, com a diferença que o usuário não enviará o **idPlano**, e sim um campo com **idPacote** e outro campo que será um JSON informando dados referentes aos planos pertencentes ao pacote, como valor do contrato (já com o desconto referente ao item do pacote), idPlanejamentoTributario, serviço de instalação e promoções de cada item. Os dados específicos para cada tecnologia deverão ser enviados no corpo do JSON normalmente como nos exemplos anteriores.

### URL

[https://\[URL\]/ws/comercial/contratos/cadastrar](https://[URL]/ws/comercial/contratos/cadastrar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados necessários para o cadastro

### Saída

JSON contendo os dados do contrato cadastrado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Long	
idCliente*	body	Long	
idTipoContrato*	body	Long	
idsEnderecos*	body	List<Long>	A lista deverá ser preenchida com o IDEnderecoCliente. É obrigatório pelo menos um do tipo PADRAO ou um do tipo INSTALACAO e um do tipo COBRANCA
idOrigemComercial*	body	Long	
idVencimento*	body	Long	
idFormaCobranca*	body	Long	
idCanalVenda*	body	Long	
podeSuspender*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
naoCancelaPorInadimplencia*	body	int	Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
fidelizacao*	body	String	Valores: MESES_24, SEM, MESES_36, MESES_12, MESES_4
idDadosDebito	body	Long	Obrigatório caso a forma de cobrança seja do tipo débito automático. Deverá ser informado o IDDadosDebitoCliente.
observacaoFiscal	body	String	
observacaoFatura	body	String	

idPacote*	body	Long	
dadosPlanosPacote*	body	List<JSON>	<b>Detalhamento do JSON abaixo da tabela</b>
loginRadius	body	String	Se o pacote possuir plano de <b>INTERNET</b> . Obrigatório para planos que não geram login automaticamente
senhaRadius	body	String	Se o pacote possuir plano de <b>INTERNET</b> . Obrigatório para planos que não geram senha automaticamente
idSaidaCaixa	body	Long	Se o pacote possuir plano de <b>INTERNET</b> . Obrigatório se for fibra e se o plano tiver o atributo <b>reservaSaida = true</b>
idCaixa	body	Long	Se o pacote possuir plano de <b>INTERNET</b> . Obrigatório se for fibra e se o plano tiver o atributo <b>reservaSaida = true</b>
idLocationCM	body	Long	Se o pacote possuir plano de <b>INTERNET</b> . Obrigatório se for fibra, se o plano tiver o atributo <b>reservaSaida = true</b> e se a empresa possuir integração com ConnectMaster
idVelocidade	body	Long	Se o pacote possuir plano de <b>INTERNET</b> . Obrigatório para planos que possuem o atributo <b>UtilizaTabelaPreco = true</b>
idNumero*	body	Long	Se o pacote possuir plano de <b>TELEFONIA</b> .
bloquearLigacaoCobrar*	body	int	Se o pacote possuir plano de <b>TELEFONIA</b> . Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
bloquearLigacaoDDD*	body	int	Se o pacote possuir plano de <b>TELEFONIA</b> . Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
bloquearLigacaoCelular*	body	int	Se o pacote possuir plano de <b>TELEFONIA</b> . Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
bloquearLigacaoInternacional*	body	int	Se o pacote possuir plano de <b>TELEFONIA</b> . Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
exibeListaTelefonica*	body	int	Se o pacote possuir plano de <b>TELEFONIA</b> . Valores: 0 ou 1 (1 = SIM/ 0 = NÃO)
idsPacotesVozExtra	body	List<String>	Se o pacote possuir plano de <b>TELEFONIA</b> .
idsCanaisExtras	body	List<String>	Se o pacote possuir plano de <b>TV</b> .
idsPacotesCanaisExtras	body	List<String>	Se o pacote possuir plano de <b>TV</b> .

**\* Parâmetro obrigatório**

Detalhamento dos campos da lista de JSON enviado no campo **dadosPlanosPacote**:

**IMPORTANTE:** cada item do pacote deve ser representado por um JSON nessa lista de **dadosPlanosPacote**.

Nome	Tipo dado	Observações
idPlano*	Long	
valorContrato*	Double	
idServicoInstalacao	Long	Obrigatório se possuir taxa de instalação
parcelasContratadasServicoInstalacao	Long	Obrigatório se possuir taxa de instalação. Se não for informado, será considerado 1
idsPromocoes	List<String>	
idPlanejamentoTributario	Long	Obrigatório se o plano tiver o atributo <b>PossuiPlanejamentoTributario = true</b>

**\* Parâmetro obrigatório**

**OBS.:** os campos não obrigatórios, podem ser enviados no JSON com vazio ("") ou nem serem enviados no JSON.

#### Exemplo entrada:

```
{
  "idTipoContrato": 1,
  "idCliente": 81593,
  "idUsuario": 285,
  "idsEnderecos": [481018],
  "idVencimento": 1,
  "idFormaCobranca": 65,
  "valorContrato": 100.0,
  "idOrigemComercial": 1,
  "podeSuspender": 1,
  "idCanalVenda": 1,
  "naoCancelaPorInadimplencia": 0,
  "fidelizacao": "MESES_12",
  "idPacote": 23,
  "dadosPlanosPacote": [{
    "idPlano": 673,
    "valorContrato": 100,
    "idServicoInstalacao": 17,
    "parcelasContratadasServicoInstalacao": 1,
    "idsPromocoes": ["20"]
  },
  {
    "idPlano": 145,
    "valorContrato": 20,
    "idsPromocoes": ["21"]
  }
  ],
  "idSaidaCaixa": 165,
  "idCaixa": 21,
  "idNumero": 311,
  "bloquearLigacaoCobrar": 1,
  "bloquearLigacaoDDD": 0,
  "bloquearLigacaoCelular": 0,
  "bloquearLigacaoInternacional": 1,
  "exibeListaTelefonica": 0
}
```

#### Exemplo retorno:

**Sucesso:**

```
{
  "resposta": [
    {
      "cliente": "81593 - CLIENTE TESTE",
      "valorContrato": 100.0,
      "tecnologia": "TELEFONIA FIXA",
      "plano": "673 - FALA BEM ILIM",
      "id": 1283609,
      "status": "AGUARDANDO INSTALACAO"
    },
    {
      "cliente": "81593 - CLIENTE TESTE",
      "valorContrato": 20.0,
      "tecnologia": "FIBRA",
      "plano": "145 - FIBRA 10MBPS FRANQUIA 360GB",
      "id": 1283610,
      "status": "AGUARDANDO INSTALACAO"
    }
  ]
}
```

**Falha:**

```
{
  "error": "O idCliente informado não pertence a nenhum cliente cadastrado no sistema."
}
```

## COMERCIAL / TERCEIROS - Endereços

### Buscar endereço completo por CEP

#### Descrição

Busca o endereço correspondente ao CEP informado

#### URL

[https://\[URL\]/ws/terceiros/enderecos/cep/{cep}](https://[URL]/ws/terceiros/enderecos/cep/{cep})

#### Método

GET

#### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

PathParam: cep

#### Saída

JSON contendo o endereço correspondente ao cep informado ou JSON contendo mensagem de erro

#### Exemplo

**Envio:** [https://\[URL\]/ws/terceiros/enderecos/cep/36404199](https://[URL]/ws/terceiros/enderecos/cep/36404199)

#### Retorno:

##### Sucesso:

**Em caso de um único endereço com o cep:**

```
{
  "id": 587374,
  "endereco": "MARILIA BERNARDINA LOBO",
  "bairro": "ROCHEDO",
  "cidade": "CONSELHEIRO LAFAIETE",
  "estado": "MG",
  "cep": "36404-199",
  "tipo": "RUA",
  "codigoCidadeIBGE": 3118304,
  "nomeCidadeIBGE": "Conselheiro Lafaiete",
  "enderecoAbreviado": "MARILIA B. LOBO",
  "bairroAbreviado": "ROCHEDO",
  "tipoAbreviado": "R.",
  "uf": "MG",
  "costingCodeSap": "2000",
  "consultaSPC": true,
  "abreviaturaCidade": null,
  "nomeCondominioCM": null,
  "nomeBlocoCM": null,
  "condominiosCM": null,
  "idCidade": 100
}
```

**Em caso de mais de um endereço com o cep:**

```
{
  "ufs": [
    "MG"
  ],
  "maisDeUmEnderecoMesmoCep": 1
}
```

##### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar cidades de um estado

### Descrição

Busca as cidades de um estado

### URL

[https://\[URL\]/ws/terceiros/cidades/uf/{uf}?cep={cep}](https://[URL]/ws/terceiros/cidades/uf/{uf}?cep={cep})

### Método

GET

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

PathParam: uf

QueryParam: cep (não obrigatório)

### Saída

JSON contendo lista de cidades de um determinado estado ou JSON contendo mensagem de erro

### Exemplo

Envio: [https://\[URL\]/ws/terceiros/cidades/uf/MG](https://[URL]/ws/terceiros/cidades/uf/MG)

**OBS.: O valor do parametro {uf} deve ser a abreviatura do nome do estado**

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "nome": "ABRE CAMPO",
      "uf": "MG"
    },
    {
      "id": 6,
      "nome": "ALFENAS",
      "uf": "MG"
    },
    {
      "id": 26,
      "nome": "ARAXA",
      "uf": "MG"
    },
    {
      "id": 33,
      "nome": "BARBACENA",
      "uf": "MG"
    },
    {
      "id": 44,
      "nome": "BELO HORIZONTE",
      "uf": "MG"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar bairros de uma cidade

### Descrição

Busca os bairros de uma determinada cidade

### URL

[https://\[URL\]/ws/terceiros/bairros/cidade/{idCidade}?cep={cep}](https://[URL]/ws/terceiros/bairros/cidade/{idCidade}?cep={cep})

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

PathParam: idCidade

QueryParam: cep (não obrigatório)

### Saída

JSON contendo lista de bairros de uma determinada cidade ou JSON contendo mensagem de erro

### Exemplo

**Envio:** [https://\[URL\]/ws/terceiros/bairros/cidade/6](https://[URL]/ws/terceiros/bairros/cidade/6)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 26337,
      "nome": "VILA TEIXEIRA"
    },
    {
      "id": 320368,
      "nome": "CENTRO"
    },
    {
      "id": 322362,
      "nome": "JARDIM SAO CARLOS"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```



## Buscar endereços (logradouros) de um bairro

### Descrição

Busca os endereços (logradouros) de um determinado bairro

### URL

[https://\[URL\]/ws/terceiros/enderecos/bairro/{idBairro}?cep={cep}](https://[URL]/ws/terceiros/enderecos/bairro/{idBairro}?cep={cep})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: idBairro

QueryParam: cep (não obrigatório)

### Saída

JSON contendo lista de endereços (logradouros) de um determinado bairro ou JSON contendo mensagem de erro

**OBS.: o campo ID desses endereços retornados é que serão utilizados no campo idEndereco dos endpoints que exigem esse dado de entrada**

### Exemplo

**Envio:** [https://\[URL\]/ws/terceiros/enderecos/bairro/26337](https://[URL]/ws/terceiros/enderecos/bairro/26337)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 712251,
      "logradouro": "AFRANIO PEIXOTO",
      "cep": "37137-166",
      "tipoLogradouro": "RUA",
      "abreviatura": "AFRANIO PEIXOTO"
    },
    {
      "id": 712252,
      "logradouro": "ANTONIO MARIA",
      "cep": "37137-167",
      "tipoLogradouro": "RUA",
      "abreviatura": "ANTONIO MARIA"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar condomínios do ConnectMaster por endereço e número

**OBS.: ESSE ENDPOINT SÓ DEVE SER UTILIZADO POR EMPRESAS QUE POSSUEM INTEGRAÇÃO COM O CONNECT MASTER**

### Descrição

Busca os condomínios através do endereço e número

### URL

[https://\[URL\]/ws/terceiros/condominiosCM/endereco/{idEndereco}/numero/{numero}](https://[URL]/ws/terceiros/condominiosCM/endereco/{idEndereco}/numero/{numero})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: idEndereco, numero (sem caracteres especiais)

### Saída

JSON contendo lista de condomínios do CM de um determinado endereço e número ou JSON contendo mensagem de erro

### Exemplo

**Envio:** [https://\[URL\]/ws/terceiros/condominiosCM/endereco/8302/numero/5](https://[URL]/ws/terceiros/condominiosCM/endereco/8302/numero/5)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "nomeCondominio": "TESTE CONDOMINIO CM",
      "numeroCondominio": "5",
      "statusCM": "A",
      "tipoCondominio": "VERTICAL",
      "blocos": {
        "id": 2,
        "idBloco": 2,
        "nomeBloco": "BLOCO TESTE"
      }
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar condomínios por endereço e número

### Descrição

Busca os condomínios através do endereço e número

#### URL

[https://\[URL\]/ws/redes/condominios/endereco/{idEndereco}/numero/{numero}](https://[URL]/ws/redes/condominios/endereco/{idEndereco}/numero/{numero})

#### Método

GET

#### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

PathParam: idEndereco, numero (sem caracteres especiais)

#### Saída

JSON contendo lista de condomínios de um determinado endereço e número ou JSON contendo mensagem de erro

#### Exemplo

**Envio:** [https://\[URL\]/ws/redes/condominios/endereco/48160/numero/145](https://[URL]/ws/redes/condominios/endereco/48160/numero/145)

#### Retorno:

##### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "descricao": "AV. JOAO PESSOA, N° 145",
      "idEndereco": 48160,
      "numero": "145",
      "latitude": "-20.6622846",
      "longitude": "-43.79039820000003",
      "numeroLacre": "13530"
    }
  ]
}
```

##### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar classificações de endereço ativas

### Descrição

Busca todas as classificações de endereço ativas

### URL

[https://\[URL\]/ws/comercial/enderecos/classificacoes](https://[URL]/ws/comercial/enderecos/classificacoes)

### Método

GET

### Entrada

Header: *Token* `{{token}}`, *Authorization Basic* `{{authorization}}`

### Saída

JSON contendo lista de classificações de endereço ativas ou JSON contendo mensagem de erro

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/enderecos/classificacoes](https://[URL]/ws/comercial/enderecos/classificacoes)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "descricao": "TESTE",
      "ativo": true,
      "condominio": false,
      "shopping": false
    },
    {
      "id": 2,
      "descricao": "A",
      "ativo": true,
      "condominio": false,
      "shopping": false
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## COMERCIAL - Prospectos

### Buscar tipos de prospectos

#### Descrição

Busca todos os tipos de prospectos

#### URL

[https://\[URL\]/ws/comercial/prospectos/tipos](https://[URL]/ws/comercial/prospectos/tipos)

#### Método

GET

#### Entrada

Header: *Token* `{{token}}`, *Authorization Basic* `{{authorization}}`

#### Saída

JSON contendo lista de tipos de prospecto ou JSON contendo mensagem de erro

#### Exemplo

##### Retorno:

##### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "descricao": "TESTE"
    },
    {
      "id": 2,
      "descricao": "TESTE 2"
    }
  ]
}
```

##### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Cadastrar prospectos

### Descrição

Cadastra um prospecto

### URL

[https://\[URL\]/ws/comercial/prospectos/cadastrar](https://[URL]/ws/comercial/prospectos/cadastrar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados do prospecto

### Saída

JSON contendo o ID do prospecto cadastrado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
tipoPessoa*	body	String	Valores: PessoaFisica ou PessoaJuridica
nome*	body	String	
telefoneFixo*	body	String	Somente números. Tamanho: 10 ou 11 caracteres obrigatoriamente
telefoneMovei*	body	String	Somente números. Tamanho: 10 ou 11 caracteres obrigatoriamente
idVendedor*	body	Number	
idSegmentoProspecto*	body	Number	
idTipoProspecto*	body	Number	
idCanalVenda*	body	Number	
idEmpresa*	body	Number	
observacaoProspecto	body	String	
email	body	String	
cpf	body	String	Preencher somente se o tipoPessoa for PessoaFisica. Campo não obrigatório. Pode ser enviado com ou sem pontuação.
dataNascimento	body	Date	Formato: dd-MM-yyyy. Campo obrigatório caso o tipoPessoa for PessoaFisica
rg	body	String	Preencher somente se o tipoPessoa for PessoaFisica. Campo não obrigatório.
nomeMae	body	String	Preencher somente se o tipoPessoa for PessoaFisica. Campo não obrigatório.
sexo	body	String	Preencher somente se o tipoPessoa for PessoaFisica. Campo não obrigatório. Valores: MASCULINO ou FEMININO
cnpj	body	String	Preencher somente se o tipoPessoa for PessoaJuridica. Campo não obrigatório. Pode ser enviado com ou sem pontuação.
inscricaoEstadual	body	String	Preencher somente se o tipoPessoa for PessoaJuridica. Campo não obrigatório.
inscricaoMunicipal	body	String	Preencher somente se o tipoPessoa for PessoaJuridica. Campo não obrigatório.
nomeFantasia	body	String	Preencher somente se o tipoPessoa for PessoaJuridica. Campo não obrigatório.

\* Parâmetro obrigatório

**OBS.:** os campos não obrigatórios quando não possuírem valor, podem ser enviados no JSON com vazio ("" ) ou nem serem enviados no JSON

#### **Exemplo entrada:**

##### **Pessoa física:**

```
{
  "tipoPessoa": "PessoaFisica",
  "nome": "TESTE WS",
  "telefoneFixo": "32999998888",
  "telefoneMovel": "32999998888",
  "dataNascimento": "15-03-1988",
  "idVendedor": 1,
  "idSegmentoProspecto": 1,
  "idTipoProspecto": 1,
  "idCanalVenda": 1,
  "idEmpresa": 1,
  "email": "teste@teste.com",
  "observacaoProspecto": "TESTE",
  "cpf": "111.111.111-11",
  "rg": "",
  "nomeMae": "TESTE MAE",
  "sexo": "",
  "cnpj": "",
  "inscricaoEstadual": "",
  "inscricaoMunicipal": "",
  "nomeFantasia": ""
}
```

##### **Pessoa jurídica:**

```
{
  "tipoPessoa": "PessoaJuridica",
  "nome": "TESTE PESSOA JURIDICA",
  "telefoneFixo": "32999998888",
  "telefoneMovel": "32999998888",
  "dataNascimento": "",
  "idVendedor": 1,
  "idSegmentoProspecto": 1,
  "idTipoProspecto": 1,
  "idCanalVenda": 1,
  "idEmpresa": 1,
  "email": "teste@teste.com",
  "observacaoProspecto": "TESTE",
  "cpf": "",
  "rg": "",
  "nomeMae": "",
  "sexo": "",
  "cnpj": "22222222222222",
  "inscricaoEstadual": "123",
  "inscricaoMunicipal": "ISENTO",
  "nomeFantasia": "TESTE PJ"
}
```

#### **Exemplo retorno:**

##### **Sucesso:**

```
{
  "idProspecto": 220335
}
```

##### **Falha:**

```
{
  "error": "Internal Server Error"
}
```

## Cadastrar endereço de prospecto

### Descrição

Cadastra um endereço em um prospecto

### URL

[https://\[URL\]/ws/comercial/prospectos/enderecos/cadastrar](https://[URL]/ws/comercial/prospectos/enderecos/cadastrar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados do endereço do prospecto

### Saída

JSON contendo o ID do prospecto cadastrado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Number	
idProspecto*	body	Number	
idEndereco*	body	Number	
numero*	body	Number	
tipoEndereco*	body	String	Valores: PADRAO, FISCAL, COBRANCA, INSTALACAO
idClassificacaoEndereco*	body	Number	
complemento	body	String	
referencia	body	String	
idCondominio	body	Number	
idBloco	body	Number	Utilizado somente quando há integração com ConnectMaster
andar	body	String	Utilizado somente quando há integração com ConnectMaster
numeroCasa	body	String	Utilizado somente quando há integração com ConnectMaster
apto	body	String	Utilizado somente quando há integração com ConnectMaster

**\* Parâmetro obrigatório**

**OBS.:** os campos não obrigatórios quando não possuírem valor, podem ser enviados no JSON com vazio ("") ou nem serem enviados no JSON

### Exemplo entrada:

```
{
  "idProspecto": 220329,
  "idEndereco": 11907,
  "numero": 45,
  "complemento": "casa A",
  "referencia": "",
  "tipoEndereco": "PADRAO",
  "idClassificacaoEndereco": 1,
  "idCondominio": 1,
  "idBloco": "",
  "andar": "",
```



```
"numeroCasa": "",  
"apto": ""  
}
```

**Exemplo retorno:****Sucesso:**

```
{  
  "idEnderecoProspecto": 64424  
}
```

**Falha:**

```
{  
  "error": "O idProspecto informado nao pertence a nenhum prospecto cadastrado no  
sistema."  
}
```

## Buscar planos disponíveis para o prospecto por tecnologia e tipo de contrato

### Descrição

Busca os planos disponíveis para o prospecto por tecnologia e tipo de contrato

### URL

[https://\[URL\]/ws/comercial/planos/tecnologia/{idTecnologia}/tipo\\_contrato/{idTipoContrato}?idProspecto={idProspecto}](https://[URL]/ws/comercial/planos/tecnologia/{idTecnologia}/tipo_contrato/{idTipoContrato}?idProspecto={idProspecto})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idTecnologia, idTipoContrato*

QueryParam: *idProspecto*

### Saída

JSON contendo lista de planos disponíveis para o prospecto ou JSON contendo mensagem de erro

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/planos/tecnologia/4/tipo\\_contrato/1?idProspecto=220329](https://[URL]/ws/comercial/planos/tecnologia/4/tipo_contrato/1?idProspecto=220329)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "Tecnologia": "RADIO",
      "DescricaoPlano": "RADIO 256KBPS",
      "Valor": null,
      "UtilizaTabelaPreco": true,
      "TipoTecnologia": "INTERNET",
      "IdPlano": 5
    },
    {
      "Tecnologia": "FIBRA",
      "DescricaoPlano": "FIBRA 100M",
      "Valor": 135.1,
      "UtilizaTabelaPreco": false,
      "TipoTecnologia": "INTERNET",
      "IdPlano": 1158
    },
    {
      "Tecnologia": "FIBRA",
      "DescricaoPlano": "FIBRA 15M",
      "Valor": 83.1,
      "UtilizaTabelaPreco": false,
      "TipoTecnologia": "INTERNET",
      "IdPlano": 1159
    }
  ]
}
```

#### Falha:

```
{
  "error": "O idProspecto informado nao pertence a nenhum prospecto cadastrado no sistema."
}
```

## Buscar pacotes de planos disponíveis para o prospecto tipo de contrato

### Descrição

Busca os pacotes de planos disponíveis para o prospecto tipo de contrato

### URL

[https://\[URL\]/ws/comercial/pacotes/tipo\\_contrato/{idTipoContrato}?idProspecto={idProspecto}](https://[URL]/ws/comercial/pacotes/tipo_contrato/{idTipoContrato}?idProspecto={idProspecto})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idTipoContrato*

QueryParam: *idProspecto*

### Saída

JSON contendo lista de pacotes de planos disponíveis para o prospecto ou JSON contendo mensagem de erro

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/pacotes/tipo\\_contrato/1?idProspecto=220329](https://[URL]/ws/comercial/pacotes/tipo_contrato/1?idProspecto=220329)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "Descricao": "TESTE DANI",
      "IdPacote": 1261,
      "PlanosPacote": [
        {
          "DescontoEmPorcentagem": false,
          "DescontoPacote": 5.0,
          "ValorPlano": 50.0,
          "Plano": "HOSPEDAGEM WEB ASP - LT"
        },
        {
          "DescontoEmPorcentagem": false,
          "DescontoPacote": 0.0,
          "ValorPlano": 50.0,
          "Plano": "LOCACAO AVULSA WI-FI"
        }
      ]
    },
    {
      "Descricao": "PACOTE DANI",
      "IdPacote": 1260,
      "PlanosPacote": [
        {
          "DescontoEmPorcentagem": false,
          "DescontoPacote": 5.0,
          "ValorPlano": null,
          "Plano": "VIAFONE FALA BEM ILIMIT"
        },
        {
          "DescontoEmPorcentagem": false,
          "DescontoPacote": 2.0,
          "ValorPlano": null,
          "Plano": "LOCACAO WI-FI"
        }
      ]
    }
  ]
}
```

```
    }  
  ]  
}  
]
```

**Falha:**

```
{  
  "error": "O idProspecto informado nao pertence a nenhum prospecto cadastrado no  
sistema."  
}
```

## Cadastrar serviços de interesse de prospecto

### Descrição

Cadastra um serviço de interesse em um prospecto

### URL

[https://\[URL\]/ws/comercial/prospectos/servicos\\_interesse/cadastrar](https://[URL]/ws/comercial/prospectos/servicos_interesse/cadastrar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados do serviço de interesse

### Saída

JSON contendo o ID do serviço de interesse cadastrado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Number	
idProspecto*	body	Number	
idPacote	body	Number	Obrigatório se o serviço de interesse for um pacote
idPlano	body	Number	Obrigatório se o serviço de interesse for um plano

**\* Parâmetro obrigatório**

**OBS.:** os campos não obrigatórios quando não possuírem valor, podem ser enviados no JSON com vazio (""), ou nem serem enviados no JSON

### Exemplo entrada:

#### Plano:

```
{
  "idProspecto": 220330,
  "idPlano": 5
}
```

#### Pacote:

```
{
  "idProspecto": 220330,
  "idPacote": 15
}
```

### Exemplo retorno:

#### Sucesso:

```
{
  "idServicoInteresse": 140130
}
```

#### Falha:

```
{
  "error": "O idProspecto informado nao pertence a nenhum prospecto cadastrado no sistema."
}
```

## VIABILIDADE - Consultar crédito interno

### Descrição

Realizar consulta de crédito interno do prospecto

### URL

[https://\[URL\]/ws/comercial/prospectos/credito\\_interno/consultar](https://[URL]/ws/comercial/prospectos/credito_interno/consultar)

### Método

POST

### Entrada

Header: *Token* `{{token}}`, *Authorization Basic* `{{authorization}}`

Body: Json contendo os dados para a consulta

### Saída

JSON contendo se o CPF/CNPJ do prospecto possui dívida ou não

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Number	
idProspecto*	body	Number	

\* Parâmetro obrigatório

### Exemplo entrada:

```
{
  "idProspecto": 220330,
  "idUsuario": 5
}
```

### Exemplo retorno:

#### Se não possuir dívida:

```
{
  "resposta": {
    "possuiDívida": false
  }
}
```

#### Se não possuir dívida:

```
{
  "resposta": {
    "faturas": [
      {
        "idCliente": 81593,
        "numeroBoleto": "21",
        "dataVencimento": "05-04-2021",
        "valor": 99.9,
        "valorPago": null,
        "id": 12205372,
        "numeroDocumento": "0421183459"
      },
      {
        "idCliente": 81593,
        "numeroBoleto": "000000000003P",
        "dataVencimento": "07-05-2021",
        "valor": 40.0,
        "valorPago": null,
        "id": 12205373,
        "numeroDocumento": "0521283459"
      }
    ]
  }
}
```

```
        }  
    ],  
    "possuiDivida": true  
}  
}
```

**Falha:**

```
{  
    "error": "O idProspecto informado nao pertence a nenhum prospecto cadastrado no  
sistema."  
}
```

## VIABILIDADE - Consultar crédito por endereço de prospecto

### Descrição

Realizar consulta de crédito por endereço do prospecto

### URL

[https://\[URL\]/ws/comercial/prospectos/credito\\_endereco/consultar](https://[URL]/ws/comercial/prospectos/credito_endereco/consultar)

### Método

POST

### Entrada

Header: *Token* `{{token}}`, *Authorization Basic* `{{authorization}}`

Body: Json contendo os dados para a consulta

### Saída

JSON contendo se o endereço do prospecto possui dívida ou não

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Number	
idProspecto*	body	Number	

**\* Parâmetro obrigatório**

### Exemplo entrada:

```
{
  "idProspecto": 220330,
  "idUsuario": 5
}
```

### Exemplo retorno:

#### Se não possuir dívida:

```
{
  "resposta": {
    "possuiDívida": false
  }
}
```

#### Se possuir dívida:

```
{
  "resposta": {
    "faturas": [
      {
        "idCliente": 81593,
        "numeroBoleto": "21",
        "dataVencimento": "05-04-2021",
        "valor": 99.9,
        "valorPago": null,
        "id": 12205372,
        "numeroDocumento": "0421183459"
      },
      {
        "idCliente": 81593,
        "numeroBoleto": "000000000003P",
        "dataVencimento": "07-05-2021",
        "valor": 40.0,
        "valorPago": null,
        "id": 12205373,
        "numeroDocumento": "0521283459"
      }
    ]
  }
}
```



```
        }  
      ],  
      "possuiDivida": true  
    }  
  }  
}
```

**Falha:**

```
{  
  "error": "O idProspecto informado nao pertence a nenhum prospecto cadastrado no  
sistema."  
}
```

## VIABILIDADE - Consultar SPC de prospecto

### Descrição

Realizar consulta no SPC

### URL

[https://\[URL\]/ws/comercial/prospectos/spc/consultar](https://[URL]/ws/comercial/prospectos/spc/consultar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados para a consulta

### Saída

JSON contendo os dados retornados do SPC e as validações de análise de crédito

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Number	
idProspecto*	body	Number	

\* Parâmetro obrigatório

### Exemplo entrada:

```
{
  "idProspecto": 220330,
  "idUsuario": 5
}
```

### Exemplo retorno:

```
{
  "resposta": {
    "id": 19140,
    "dataConsulta": 1621016934427,
    "idUsuario": 5,
    "tipoPessoa": null,
    "cpf": "11122211122",
    "dataNascimento": 574398000000,
    "idade": 33,
    "nome": "TESTE",
    "nomeMae": null,
    "rg": null,
    "sexo": null,
    "enderecoLogradouro": "R TESTE",
    "numeroEndereco": "400",
    "bairro": "TESTE",
    "cep": "36280000",
    "cidade": "CARANDAI",
    "uf": "MG",
    "telefone": "3233334444",
    "totalRestricoes": 0,
    "valorTotalRestricoes": null,
    "dataUltimaRestricao": null,
    "cnpj": null,
    "nomeFantasia": null,
    "inscricaoEstadual": null,
    "detalhesLojistas": [],
  }
}
```

```

"analisesCredito": [
  {
    "id": 75328,
    "dataAnalise": 1621016934529,
    "criterioAprovacaoCredito": {
      "id": 1,
      "ativo": true,
      "pergunta": "Quantidade de negativas",
      "operador": "<=",
      "valorReferencia": "4",
      "nomeCampoComparacao": "totalRestricoes",
      "tipoPessoa": null
    },
    "valorCriterio": "totalRestricoes",
    "aprovado": true,
    "idUsuarioAprovacao": 285
  },
  {
    "id": 75327,
    "dataAnalise": 1621016934529,
    "criterioAprovacaoCredito": {
      "id": 3,
      "ativo": true,
      "pergunta": "Restrições acima de R$ 500,00",
      "operador": "<=",
      "valorReferencia": "500",
      "nomeCampoComparacao": "valorTotalRestricoes",
      "tipoPessoa": null
    },
    "valorCriterio": "valorTotalRestricoes",
    "aprovado": true,
    "idUsuarioAprovacao": 285
  },
  {
    "id": 75326,
    "dataAnalise": 1621016934529,
    "criterioAprovacaoCredito": {
      "id": 5,
      "ativo": true,
      "pergunta": "Possui mais de 18 anos",
      "operador": ">=",
      "valorReferencia": "18",
      "nomeCampoComparacao": "idade",
      "tipoPessoa": "PessoaFisica"
    },
    "valorCriterio": "idade",
    "aprovado": true,
    "idUsuarioAprovacao": 285
  }
],
"dataPrimeiraRestricao": null,
"diasEmRestricao": null,
"totalPontosCredito": null,
"resultadoPontuacao": null,
"score": 485,
"probabilidade": 31.15,
"classe": "D",
"aprovado": false
}
}

Falha:
{
  "error": "Erro ao verificar credito"
}

```

## VIABILIDADE - Consultar Clear Sale de prospecto

### Descrição

Realizar consulta DataTrust – Clear Sale

### URL

[https://\[URL\]/ws/comercial/prospectos/clear\\_sale/consultar](https://[URL]/ws/comercial/prospectos/clear_sale/consultar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados para a consulta

### Saída

JSON contendo resultados da consulta DataTrust – Clear Sale

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Number	
idProspecto*	body	Number	

**\* Parâmetro obrigatório**

### Exemplo entrada:

```
{
  "idProspecto": 220330,
  "idUsuario": 5
}
```

### Exemplo retorno:

```
{
  "resposta": {
    "id": 22,
    "documento": 11123222255,
    "tipoDocumento": null,
    "ddd": 32,
    "telefone": 333333333,
    "email": null,
    "idSessao": null,
    "tipo": 0,
    "dataReferencia": null,
    "enviaOpcao": null,
    "verificaEmail": false,
    "verificaTelefone": false,
    "endereco": null,
    "informacoesAdicionais": null,
    "dataCriacao": null,
    "resultado": {
      "ponto": {
        "valor": 90.99,
        "razao": "Confirmação do TokenSMS",
        "data": "2019-06-07T12:45:00.000",
        "linhasTempo" : [
          {
            "valor": 80.0,
            "Reason": "Confirmação do TokenSMS",
            "data": "2018-06-07T12:45:00.000"
          }
        ]
      }
    }
  }
}
```

```

    }
  ],
  },
  "validacao": {
    "verificacaoSMS": {
      "enviado": {
        "texto": "Ola Joao Silva! Você está participando do teste do
XTrust? 1 - SIM, 2 - NAO, 3 - Nao sou Joao Silva",
        "data": "2018-06-05T12:00:00.000"
      },
      "respondido": {
        "texto": "1) SIM",
        "data": "2018-06-05T12:45:00.000"
      },
      "regraCondicional" : null
    },
    "verificacaoEmail": {
      "enviado": {
        "texto": "Ola Joao Silva! Você está participando do teste do
XTrust?",
        "data": "2018-06-05T12:00:00.000"
      },
      "respondido": {
        "texto": "1",
        "data": "2018-06-05T12:45:00.000"
      }
    }
  },
  "avaliacoes": [{
    "valor": 3,
    "relacionadoA": [
      "Document",
      "Phone"
    ]
  }],
  "intuicoes": [{
    "codigo": "TEL001",
    "descricao": "O Celular informado foi visto nos ultimos 3 meses.",
    "tipo": "consulta",
    "categoria": "fraude",
    "relevancia": "Positivo",
    "relacionadoA": [
      "Document",
      "Phone"
    ]
  }]
],
},
"sucesso": true,
"tempoResposta": 11,
"mensagem": "TESTE"
}
}

```

#### Falha:

```

{
  "error": "Internal Server Error"
}

```

## VIABILIDADE - Consultar viabilidade técnica de prospecto

### Descrição

Realizar consulta de viabilidade técnica no prospecto

### URL

[https://\[URL\]/ws/comercial/prospectos/viabilidade\\_tecnica/consultar](https://[URL]/ws/comercial/prospectos/viabilidade_tecnica/consultar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados para a consulta

### Saída

JSON contendo resultados da viabilidade técnica

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Number	
idProspecto*	body	Number	

**\* Parâmetro obrigatório**

### Exemplo entrada:

```
{
  "idProspecto": 220330,
  "idUsuario": 5
}
```

### Exemplo retorno:

```
{
  "resposta": {
    "resultado": "APROVADO",
    "distancia": 0.13157409087606006
  }
}
```

### Falha:

```
{
  "error": "Internal Server Error"
}
```

## VIABILIDADE - Consultar viabilidade técnica por endereço

### Descrição

Realizar consulta de viabilidade técnica por endereço

### URL

[https://\[URL\]/ws/comercial/viabilidade\\_tecnica/consultar](https://[URL]/ws/comercial/viabilidade_tecnica/consultar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados para a consulta

### Saída

JSON contendo resultados da viabilidade técnica

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idEndereco*	body	Number	
numero*	body	Number	
latitude*	body	String	
longitude*	body	String	
idSegmentoProspecto*	body	Number	
idCondominioCM	body	Number	Enviar somente se a empresa possuir integração com ConnectMaster
idBloco	body	Number	Enviar somente se a empresa possuir integração com ConnectMaster
andar	body	Number	Enviar somente se a empresa possuir integração com ConnectMaster
numeroCasa	Body	Number	Enviar somente se a empresa possuir integração com ConnectMaster

**\* Parâmetro obrigatório**

### Exemplo entrada:

```
{
  "idEndereco": 481018,
  "numero": 85,
  "latitude": "-20.6560787",
  "longitude": "-43.756296599999985",
  "idSegmentoProspecto": 1,
  "idCondominioCM": null,
  "idBloco": null,
  "andar": null,
  "numeroCasa": null
}
```

### Exemplo retorno:

```
{
  "resposta": {
    "resultado": "APROVADO",
    "distancia": 0.13157409087606006
  }
}
```

**Falha:**

```
{  
  "error": "Internal Server Error"  
}
```



## VIABILIDADE - Consultar crédito por endereço

### Descrição

Realizar consulta de crédito por endereço

### URL

[https://\[URL\]/ws/comercial/credito\\_endereco/consultar](https://[URL]/ws/comercial/credito_endereco/consultar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados para a consulta

### Saída

JSON contendo se o endereço possui dívida ou não

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idEndereco*	body	Number	
numero*	body	Number	
idCondominioCM	body	Number	Enviar somente se a empresa possuir integração com ConnectMaster
idBloco	body	Number	Enviar somente se a empresa possuir integração com ConnectMaster
andar	body	Number	Enviar somente se a empresa possuir integração com ConnectMaster
numeroCasa	Body	Number	Enviar somente se a empresa possuir integração com ConnectMaster

**\* Parâmetro obrigatório**

### Exemplo entrada:

```
{
  "idEndereco": 481018,
  "numero": 85,
  "idCondominioCM": null,
  "idBloco": null,
  "andar": null,
  "numeroCasa": null
}
```

### Exemplo retorno:

#### Se não possuir dívida:

```
{
  "resposta": {
    "possuiDívida": false
  }
}
```

#### Se possuir dívida:

```
{
  "resposta": {
    "faturas": [
      {
        "idCliente": 81593,
        "numeroBoleto": "21",
        "dataVencimento": "05-04-2021",
        "valor": 99.9,
      }
    ]
  }
}
```

```

        "valorPago": null,
        "id": 12205372,
        "numeroDocumento": "0421183459"
    },
    {
        "idCliente": 81593,
        "numeroBoleto": "00000000003P",
        "dataVencimento": "07-05-2021",
        "valor": 40.0,
        "valorPago": null,
        "id": 12205373,
        "numeroDocumento": "0521283459"
    }
],
"possuiDivida": true
}
}

```

**Falha:**

```

{
    "error": "Internal Server Error"
}

```

## Buscar status de vendas do vendedor

### Descrição

Busca status de vendas dos vendedores

### URL

[https://\[URL\]/ws/comercial/prospectos/vendas/status/{idVendedor}](https://[URL]/ws/comercial/prospectos/vendas/status/{idVendedor})

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

PathParam: *idVendedor*

### Saída

JSON contendo os dados das vendas

### Exemplo

#### Retorno:

#### Sucesso:

```
{
  "resposta": {
    "REALIZADAS": 4,
    "EM_ANDAMENTO": 16,
    "RECUSADAS": 0
  }
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar prospectos vendedor ID, nome e oportunidade

### Descrição

Busca prospectos de acordo com os parâmetros passados

### URL

[https://\[URL\]/ws/comercial/prospectos/buscar/vendedor/{idVendedor}?id={id}&nome={nome}&oportunidade={oportunidade}](https://[URL]/ws/comercial/prospectos/buscar/vendedor/{idVendedor}?id={id}&nome={nome}&oportunidade={oportunidade})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idVendedor*

QueryParam: *id (opcional), nome (opcional), oportunidade (opcional – valores: ABERTA, PERDEU, GANHOU)*

### Saída

JSON contendo os prospectos

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/prospectos/buscar/vendedor/285?](https://[URL]/ws/comercial/prospectos/buscar/vendedor/285?id=220317&nome=TESTE&oportunidade=ABERTA)

[id=220317&nome=TESTE&oportunidade=ABERTA](https://[URL]/ws/comercial/prospectos/buscar/vendedor/285?id=220317&nome=TESTE&oportunidade=ABERTA)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 220311,
      "nome": "TESTE A",
      "cpf": "66655544488",
      "cnpj": "",
      "oportunidade": "ABERTA"
    },
    {
      "id": 220317,
      "nome": "CLIENTE TESTE 1",
      "cpf": "33366655544",
      "cnpj": null,
      "oportunidade": "ABERTA"
    },
    {
      "id": 220323,
      "nome": "CLIENTE TESTE 1",
      "cpf": "11122255566",
      "cnpj": null,
      "oportunidade": "ABERTA"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar prospecto por ID

### Descrição

Busca prospecto pelo ID

### URL

[https://\[URL\]/ws/comercial/prospectos/{id}](https://[URL]/ws/comercial/prospectos/{id})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *id*

### Saída

JSON contendo os dados do prospecto

### Exemplo

Envio: [https://\[URL\]/ws/comercial/prospectos/220315](https://[URL]/ws/comercial/prospectos/220315)

### Retorno:

#### Sucesso:

```
{
  "resposta": {
    "id": 220330,
    "tipoPessoa": "PessoaFisica",
    "email": "teste@teste.com",
    "nome": "TESTE WS2",
    "telefoneFixo": "32999998888",
    "telefoneMovel": "32999998888",
    "inscricaoEstadual": null,
    "inscricaoMunicipal": null,
    "nomeFantasia": null,
    "cpf": "76970141063",
    "cnpj": null,
    "dataNascimento": 574398000000,
    "rg": "9898989898",
    "nomeMae": "TESTE",
    "sexo": "FEMININO",
    "senhaApp": null,
    "senhaCentral": null,
    "dataCadastro": 1617621469000,
    "dataUltimaAtualizacao": 1628277389000,
    "idUsuarioUltimaAlteracao": 285,
    "observacaoProspecto": null,
    "isExterno": true,
    "motivoReprovacaoProspecto": null,
    "funilProspecto": {
      "id": 2,
      "descricao": "SEGUNDO",
      "exibeKanban": true,
      "inicial": false,
      "ordem": 2,
      "fluxoAutomatico": true,
      "segmentoProspecto": {
        "id": 1,
        "descricao": "RESIDENCIAL",
        "descricaoRDStation": null,
        "tipoContrato": {
          "id": 1,
```

```

        "nome": "RESIDENCIAL"
    },
    },
    "qualificacao": "CONVERTER_CLIENTE",
    "permiteReanalise": false
},
"dataUltimaAlteracaoStatus": 1628277389000,
"tipoProspecto": {
    "id": 1,
    "descricao": "TESTE"
},
"canalVenda": {
    "id": 1,
    "descricao": "A",
    "descricoesRD": null,
    "permiteQualityCall": false,
    "ativo": true
},
"observacaoQualificacoes": null,
"complementoMotivoReprovacao": null,
"idVendedor": 285,
"segmentoProspecto": {
    "id": 1,
    "descricao": "RESIDENCIAL",
    "descricaoRDStation": null,
    "tipoContrato": {
        "id": 1,
        "nome": "RESIDENCIAL"
    }
},
"oportunidade": "ABERTA",
"temperatura": null,
"idEmpresa": 1,
"integrarRDStation": null,
"enderecosProspecto": [
    {
        "andar": null,
        "cidade": "CONSELHEIRO LAFAIETE",
        "estado": "MG",
        "numero": 96,
        "idCondominio": null,
        "idEndereco": 587374,
        "bairro": "ROCHEDO",
        "latitude": "-20.6647539",
        "apto": null,
        "numeroCasa": null,
        "idCidade": 100,
        "idBloco": null,
        "idBairro": 19298,
        "complemento": null,
        "tipoDeEndereco": "PADRAO",
        "logradouro": "R. MARILIA BERNARDINA LOBO",
        "classificacaoEndereco": {
            "id": 1,
            "descricao": "TESTE",
            "ativo": true,
            "condominio": false,
            "shopping": false
        },
        "id": 64427,
        "referencia": null,
    }
]

```

```

        "idCondominioCM": null,
        "longitude": "-43.7741393"
    }
],
"planosProspectados": [
    {
        "id": 140135,
        "idPlano": 1162,
        "idProspecto": 1123321,
        "descricaoPlano": "FIBRA VERO MAIS - 35M (FIBRA)",
        "idPacote": null,
        "descricaoPacote": null,
        "idTecnologia": 4,
        "descricaoTecnologia": "FIBRA",
        "tipoTecnologia": "INTERNET",
        "idTipoContrato": 1,
        "descricaoTipoContrato": "RESIDENCIAL",
        "status": "PENDENTE"
    }
],
"convertido": 1
}
}
}

```

#### **Falha:**

```

{
    "error": "Internal Server Error"
}

```

## Aprovar ou reprovar prospecto

### Descrição

Realizar aprovação ou reprovação de prospecto

### URL

[https://\[URL\]/ws/comercial/prospectos/aprovar/reprovar](https://[URL]/ws/comercial/prospectos/aprovar/reprovar)

### Método

PUT

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

Body: Json contendo os dados para a consulta

### Saída

JSON contendo o prospecto alterado

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Number	
idProspecto*	body	Number	
acao*	body	String	Valores: aprovar, reprovar

**\* Parâmetro obrigatório**

### Exemplo entrada:

```
{
  "idUsuario": 285,
  "idProspecto": 220334,
  "acao": "reprovar"
}
```

### Exemplo retorno:

```
{
  "resposta": 220333
}
```

### Falha:

```
{
  "error": "Internal Server Error"
}
```



## Fazer upload de arquivos para o cadastro de prospecto

### Descrição

Faz o upload de uma lista de arquivos para o cadastro de um prospecto

### URL

[https://\[URL\]/ws/comercial/prospectos/anexar\\_arquivos](https://[URL]/ws/comercial/prospectos/anexar_arquivos)

### Método

POST

### Entrada

HeaderParam("Token") com o value contendo um token válido, gerado pelo método "Gerar Token".

JSON contendo a lista de arquivos que deseja fazer o upload.

**OBS.:** a lista não deve conter mais que 5 arquivos e os arquivos (model) não podem ter tamanho maior que 5MB.

### Saída

String de "SUCESSO" ou o erro encontrado.

### Campos do JSON da entrada

Tag	Tipo	Observação
Token*	header	
Authorization*	header	
idProspecto*	Long	
model*	Map<String, Object>	

\*Campos obrigatórios

**OBS.:** o Map **model** é onde serão enviados os dados do arquivo (imagem, pdf, txt, etc) em base64. Todos os campos do mesmo são obrigatórios e devem ser dos seguintes tipos:

base64 (String),

filename (String),

filesize (String),

filetype (String)

### Exemplo

#### Envio:

```
[{
  "idProspecto": 81593,
  "model": {
    "base64": "QzpcVXNlcnNcUHJpc2NpbGxhXERvd25sb2Fkc1xURVNURV9TSU5URUdSQSAoMjMpL0cmli...",
    "filename": "foto.jpg",
    "filesize": 106743,
    "filetype": "image/jpeg"
  }
},
{
  "idProspecto": 81593,
  "model": {
    "base64": "QzpcVXNlcnNcUHJpc2NpbGxhXERvd25sb2Fkc1xURVNURV9TSU5URUdS4dDb250cmli...",
    "filename": "cac.pdf",
    "filesize": 106743,
    "filetype": "application/pdf"
  }
}]
```

#### Retorno:

Em caso de sucesso:

```
{
  "success": "OK"
}
```

#### Em caso de falha:

```
{
  "error": "O tamanho da imagem foto.jpg excede o limite de 5MB."
}
```

## Buscar arquivos de prospecto

### Descrição

Busca os arquivos de um prospecto

### URL

[https://\[URL\]/ws/comercial/prospectos/{id}/arquivos](https://[URL]/ws/comercial/prospectos/{id}/arquivos)

### Método

GET

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

PathParam: *id*

### Saída

JSON contendo os dados dos arquivos do prospecto

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/prospectos/220333/arquivos](https://[URL]/ws/comercial/prospectos/220333/arquivos)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 8,
      "tipo": "text/plain",
      "nome": "texto_240821162339.txt",
      "dataVinculo": 1629833019000,
      "idProspecto": 220333
    },
    {
      "id": 9,
      "tipo": "text/plain",
      "nome": "teste_240821162548.txt",
      "dataVinculo": 1629833149000,
      "idProspecto": 220333
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Atualizar dados de prospectos

### Descrição

Atualiza dados de um prospecto

### URL

[https://\[URL\]/ws/comercial/prospectos/atualizar](https://[URL]/ws/comercial/prospectos/atualizar)

### Método

PUT

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados para atualizar um prospecto

### Saída

JSON contendo o ID do prospecto atualizado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idProspecto*	body	String	
idVendedor*	body	String	
obs	body	String	
email	body	String	
inscricaoEstadual	body	String	
inscricaoMunicipal	body	String	
nomeMae	body	String	
nomeFantasia	body	String	
sexo	body	String	Valores: MASCULINO ou FEMININO

**\* Parâmetro obrigatório**

**OBS.:** os campos não obrigatórios quando não possuírem valor, podem ser enviados no JSON com vazio (""), ou nem serem enviados no JSON

### Exemplo entrada:

```
{
  "idProspecto": "219168",
  "obs": "TESTE TESTE TESTE TESTE",
  "email": "teste@teste.com",
  "inscricaoEstadual": "5454545",
  "inscricaoMunicipal": "isento",
  "idVendedor": "285"
}
```

### Exemplo retorno:

#### Sucesso:

```
{
  "idProspecto": 219168
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## COMERCIAL - Clientes

### Buscar CFOPs

#### Descrição

Busca todos os cfops

#### URL

[https://\[URL\]/ws/comercial/cfops](https://[URL]/ws/comercial/cfops)

#### Método

GET

#### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

#### Saída

JSON contendo lista de cfops ou JSON contendo mensagem de erro

#### Exemplo

##### Retorno:

##### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "descricao": "Prestacao de servico de comunicacao a nao contribuinte",
      "numero": 5307
    },
    {
      "id": 2,
      "descricao": "Prestacao de servico de comunicacao a produtor rural ",
      "numero": 5306
    },
    {
      "id": 3,
      "descricao": "Prestacao de servico de comunicacao a distribuidora energia",
      "numero": 5305
    }
  ]
}
```

##### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar áreas de atuação para clientes PJ

### Descrição

Busca todos as áreas de atuação (segmentos) para cadastro de clientes PJ

### URL

[https://\[URL\]/ws/comercial/areas\\_atuacao](https://[URL]/ws/comercial/areas_atuacao)

### Método

GET

### Entrada

Header: *Token* `{{token}}`, *Authorization Basic* `{{authorization}}`

### Saída

JSON contendo lista de áreas de atuação (segmento) ou JSON contendo mensagem de erro

### Exemplo

#### Retorno:

##### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "descricao": "ADVOCACIA"
    },
    {
      "id": 2,
      "descricao": "AGROPECUÁRIA"
    },
    {
      "id": 3,
      "descricao": "ACADEMIA DE GINÁSTICA"
    }
  ]
}
```

##### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar tipos de atividade para clientes PJ

### Descrição

Busca todos tipos de atividade para cadastro de clientes PJ

### URL

[https://\[URL\]/ws/comercial/atividades/tipos](https://[URL]/ws/comercial/atividades/tipos)

### Método

GET

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

### Saída

JSON contendo lista de tipos de atividade ou JSON contendo mensagem de erro

### Exemplo

#### Retorno:

##### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "descricao": "COMERCIO",
      "codigoConvenio": "01"
    },
    {
      "id": 2,
      "descricao": "INDUSTRIA",
      "codigoConvenio": "02"
    },
    {
      "id": 3,
      "descricao": "PREST. DE SERVIÇOS",
      "codigoConvenio": "99"
    }
  ]
}
```

##### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar tributações federais para clientes PJ

### Descrição

Busca todas as tributações federais para cadastro de clientes PJ

### URL

[https://\[URL\]/ws/comercial/tributacoes\\_federais](https://[URL]/ws/comercial/tributacoes_federais)

### Método

GET

### Entrada

Header: *Token* `{{token}}`, *Authorization Basic* `{{authorization}}`

### Saída

JSON contendo lista de tributações federais ou JSON contendo mensagem de erro

### Exemplo

#### Retorno:

##### Sucesso:

```
{
  "resposta": [
    {
      "id": 5,
      "descricao": "ISENTO"
    },
    {
      "id": 3,
      "descricao": "MEI"
    },
    {
      "id": 1,
      "descricao": "NORMAL"
    },
    {
      "id": 2,
      "descricao": "SIMPLES"
    }
  ]
}
```

##### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar tipos de contato

### Descrição

Busca todos os tipos de contato para cadastro de contatos de clientes

### URL

[https://\[URL\]/ws/comercial/clientes/contatos/tipos](https://[URL]/ws/comercial/clientes/contatos/tipos)

### Método

GET

### Entrada

Header: *Token* `{{token}}`, *Authorization Basic* `{{authorization}}`

### Saída

JSON contendo lista de tipos de contato ou JSON contendo mensagem de erro

### Exemplo

#### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 2,
      "descricao": "ADMINISTRATIVO"
    },
    {
      "id": 1,
      "descricao": "COMERCIAL"
    },
    {
      "id": 4,
      "descricao": "FINANCEIRO"
    },
    {
      "id": 6,
      "descricao": "OUTORGADO"
    },
    {
      "id": 5,
      "descricao": "SINDICO"
    },
    {
      "id": 3,
      "descricao": "TECNICO"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```



## Converter prospecto em cliente

### Descrição

Converte um prospecto já cadastrado em cliente

### URL

[https://\[URL\]/ws/comercial/prospectos/converter/cliente](https://[URL]/ws/comercial/prospectos/converter/cliente)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados necessários para a conversão

### Saída

JSON contendo o ID do cliente cadastrado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Number	
idProspecto*	body	Number	
emiteNotaFiscal*	body	Number	0 = NÃO, 1 = SIM
emiteNotaUnica*	body	Number	0 = NÃO, 1 = SIM
enviaNotaFiscalEmail*	body	Number	0 = NÃO, 1 = SIM
emiteFaturaUnica*	body	Number	0 = NÃO, 1 = SIM
formadorOpiniao*	body	Number	0 = NÃO, 1 = SIM
funcionario*	body	Number	0 = NÃO, 1 = SIM
parceiro*	body	Number	0 = NÃO, 1 = SIM
recebeSMS*	body	Number	0 = NÃO, 1 = SIM
nomePai	body	String	Somente para Pessoa Física
idTipoAtividade*	body	Number	Somente para Pessoa Jurídica
idTributacaoFederal	body	Number	Somente para Pessoa Jurídica
idCfop*	body	Number	Somente para Pessoa Jurídica

**\* Parâmetro obrigatório**

**OBS.:** os campos não obrigatórios quando não possuírem valor, podem ser enviados no JSON com vazio ("" ) ou nem serem enviados no JSON

### Exemplo entrada:

```
{
  "idProspecto": 220329,
  "emiteNotaFiscal": 1,
  "emiteNotaUnica": 1,
  "enviaNotaFiscalEmail": 1,
  "emiteFaturaUnica": 1,
  "formadorOpiniao": 1,
  "funcionario": 1,
  "parceiro": 1,
  "recebeSMS": 1,
  "nomePai": "TESTE PAI"
}
```

**Exemplo retorno:****Sucesso:**

```
{  
  "idCliente": 428738  
}
```

**Falha:**

```
{  
  "error": "O idProspecto informado nao pertence a nenhum prospecto cadastrado no  
sistema."  
}
```

## Cadastrar áreas de atuação de um cliente PJ

### Descrição

Cadastrar áreas de atuação (segmentos) de um cliente PJ

### URL

[https://\[URL\]/ws/comercial/clientes/areas\\_atuacao/cadastrar](https://[URL]/ws/comercial/clientes/areas_atuacao/cadastrar)

### Método

POST

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

Body: Lista de Json contendo os dados necessários para o cadastro

### Saída

JSON contendo a lista de IDs dos vínculos de área de atuação e cliente cadastrado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Number	
idCliente*	body	Number	
idAreaAtuacao*	body	Number	
tipo*	body	String	Valores: PRIMARIO, SECUNDARIO

**\* Parâmetro obrigatório**

**OBS.:** os campos não obrigatórios quando não possuírem valor, podem ser enviados no JSON com vazio ("") ou nem serem enviados no JSON

### Exemplo entrada:

```
[{
  "idCliente": 220329,
  "idAreaAtuacao": 1,
  "tipo": "PRIMARIO"
},
{
  "idCliente": 220329,
  "idAreaAtuacao": 3,
  "tipo": "SECUNDARIO"
}]
```

### Exemplo retorno:

#### Sucesso:

```
{
  "idsAreasAtuacaoCliente": [11613,11614]
}
```

#### Falha:

```
{
  "error": "O idAreaAtuacao informado não pertence a nenhuma área de atuação cadastrada no sistema."
}
```

## Cadastrar dados para débito automático de cliente

### Descrição

Cadastrar dados para débito automático de cliente

### URL

[https://\[URL\]/ws/comercial/clientes/dados\\_debito/cadastrar](https://[URL]/ws/comercial/clientes/dados_debito/cadastrar)

### Método

POST

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

Body: Lista de Json contendo os dados necessários para o cadastro

### Saída

JSON contendo a lista de IDs dos dados de débito cadastrados ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Number	
idCliente*	body	Number	
idBanco*	body	Number	
nomeTitular*	body	String	
agencia*	body	String	
dvAgencia	body	String	
contaCorrente*	body	String	
dvContaCorrente*	body	String	

**\* Parâmetro obrigatório**

**OBS.:** os campos não obrigatórios quando não possuírem valor, podem ser enviados no JSON com vazio ("") ou nem serem enviados no JSON

### Exemplo entrada:

```
[{
  "idCliente": 220329,
  "idBanco": 2,
  "nomeTitular": "TESTE 1",
  "agencia": "1111",
  "dvAgencia": "",
  "contaCorrente": "55555",
  "dvContaCorrente": "6"
},
{
  "idCliente": 220329,
  "idBanco": 1,
  "nomeTitular": "TESTE 2",
  "agencia": "4444",
  "dvAgencia": "5",
  "contaCorrente": "555444",
  "dvContaCorrente": "7"
}]
```

### Exemplo retorno:

**Sucesso:**

```
{  
  "idsDadosDebitoCliente": [6935,6936]  
}
```

**Falha:**

```
{  
  "error": "O idCliente informado não pertence a nenhum cliente cadastrado no  
sistema."  
}
```

## Cadastrar contatos de um cliente

**Descrição**

Cadastrar contatos de um cliente

**URL**

[https://\[URL\]/ws/comercial/clientes/contatos/cadastrar](https://[URL]/ws/comercial/clientes/contatos/cadastrar)

**Método**

POST

**Entrada**

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Lista de Json contendo os dados necessários para o cadastro

**Saída**

JSON contendo a lista de IDs dos contatos cadastrados ou JSON contendo mensagem de erro

## Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Number	
idCliente*	body	Number	
nomeOutorgado*	body	Number	
idTipoContato*	body	Number	
email	body	String	
celular	body	String	Somente números. Tamanho: 10 ou 11 caracteres obrigatoriamente
telefone	body	String	Somente números. Tamanho: 10 ou 11 caracteres obrigatoriamente
cargo	body	String	
dataNascimento	body	String	Formato: dd-MM-yyyy
cpf	body	String	
rg	body	String	

**\* Parâmetro obrigatório**

**OBS.:** os campos não obrigatórios quando não possuírem valor, podem ser enviados no JSON com vazio ("") ou nem serem enviados no JSON

### Exemplo entrada:

```
[{
  "idCliente": 220329,
  "nomeOutorgado": "TESTE 1",
  "email": "teste@teste.com",
  "celular": "32333333333",
  "telefone": "32444444444",
  "dataNascimento": "01-05-1982",
  "idTipoContato": 1
},
{
  "idCliente": 220329,
  "nomeOutorgado": "TESTE 2",
  "email": "teste@teste2.com",
  "celular": "32333333333",
  "cpf": "111222333222",
  "rg": "1232212222",
  "cargo": "TESTE",
  "telefone": "32444444444",
  "dataNascimento": "01-05-1982",
  "idTipoContato": 2
}]
```

### Exemplo retorno:

#### Sucesso:

```
{
  "idsContatosCliente": [388653,388654]
}
```

#### Falha:

```
{
  "error": "O idCliente informado não pertence a nenhum cliente cadastrado no sistema."
}
```

## Informações de Cliente

### Descrição

Busca informações de clientes de acordo com os filtros passados por parâmetro.

### URL

https://{{url}}/ws/area/cliente/{idCliente: [0-9]+}

### Método

GET

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

PathParam: *idCliente*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idCliente*	path	Number	

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": {
    "telefoneFixo": "9999999999",
    "rg": "11111111",
    "telefoneMovei": "88888888888",
    "cpf": "12345678900",
    "nome": "TESTE TESTE",
    "contratos": [
      {
        "descricaoPlano": "PLANO INTERNET TESTE - PRE",
        "id": 311622,
        "status": "HABILITADO"
      },
      {
        "descricaoPlano": "PLANO TELEFONIA TESTE - PRE",
        "id": 311623,
        "status": "CANCELADO"
      }
    ],
    "dataNascimento": 656218800000,
    "email": "teste.teste@teste.teste",
    "nomeMae": "TESTE TESTE"
  },
  "status": "OK"
}
```

## Habilitar/desabilitar notificações de cliente

### Descrição

Realiza o processo de habilitação/desabilitação de notificações de clientes de acordo com as informações passadas por parâmetro.

### URL

https://{{url}}/ws/area/cliente/alterar/configuracao/notificacoes

### Método

PUT

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: *idAtendimento*, *recebeNotificacoes*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idAtendimento*	body	Number	
recebeNotificacoes*	body	Boolean	.

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": "O campo que define o recebimento de notificações do cliente de ID 'X' foi atualizado com sucesso.",
  "status": "OK"
}
```



## Informações de Aviso de Aplicativo

### Descrição

Busca informações de aviso do aplicativo cujo status seja igual a "PENDENTE", de acordo com os filtros passados por parâmetro.

### URL

https://{{url}}/ws/area/cliente/{idCliente: [0-9]+}/avisos\_app

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

PathParam: *idCliente*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idCliente*	path	Number	

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": [
    {
      "id": 1,
      "descricao": "TESTE",
      "dataEnvio": 1611671985000,
      "titulo": "TESTE",
      "mensagem": "TESTE",
      "idCliente": 308261,
      "status": "PENDENTE",
      "idFatura": 1,
      "idRotina": 1
    }
  ],
  "status": "OK"
}
```

## Resetar senha do App

### Descrição

Reseta a senha do cliente.

### URL

[https://\[URL\]/ws/area/cliente/resetar\\_senha](https://[URL]/ws/area/cliente/resetar_senha)

### Método

PUT

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

JSON contendo o CPF/CNPJ ou IDCliente para reseta senha

### Saída

Resposta informando se o atendimento foi criado.

### Campos do JSON da entrada

Tag	Tipo	Observação
CPFCNPJ*	String	
IDCliente*	Long	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "CPFCNPJ": "12312312312"
}
```

#### Retorno:

#### Sucesso :

```
{
  "Sucesso": " Senha resetada com sucesso.",
  "Senha": 020202
}
```

#### Falha :

```
{
  "Error": "Não foi possivel encontrar um cliente com CPF/CNPJ informado."
}
```

#### Ou :

```
{
  "Error": " Não foi possivel reseta a senha.",
  "Motivo": "Ocorreu um erro ao gerar a nova senha: Descricao do motivo."
}
```

## Cadastrar endereço em cliente

### Descrição

Cadastrar endereços em um cliente

### URL

[https://\[URL\]/ws/comercial/clientes/enderecos/cadastrar](https://[URL]/ws/comercial/clientes/enderecos/cadastrar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Lista de Json contendo os dados necessários para o cadastro

### Saída

JSON contendo o IDEnderecoCliente cadastrado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Number	
idCliente*	body	Number	
idEndereco*	body	Number	
numero*	body	String	
complemento	body	String	
referencia	body	String	
tipoEndereco*	body	String	Valores: PADRAO, FISCAL, COBRANCA, INSTALACAO
idClassificacaoEndereco*	body	Number	
idCondominio	body	Number	Opcional e só deve ser preenchido se houver integração com ConnectMaster
idBloco	body	Number	Opcional e só deve ser preenchido se houver integração com ConnectMaster
andar	body	String	Opcional e só deve ser preenchido se houver integração com ConnectMaster
numeroCasa	body	String	Opcional e só deve ser preenchido se houver integração com ConnectMaster
apto	body	String	Opcional e só deve ser preenchido se houver integração com ConnectMaster

**\* Parâmetro obrigatório**

**OBS.:** os campos não obrigatórios quando não possuírem valor, podem ser enviados no JSON com vazio ("") ou nem serem enviados no JSON

### Exemplo entrada:

```
{
  "idCliente": 81593,
  "idEndereco": 645337,
  "numero": 70,
  "complemento": "",
  "referencia": "",
  "tipoEndereco": "PADRAO",
  "idClassificacaoEndereco": 1,
  "idCondominio": 1,
```

```
"idBloco": "",  
"andar": "",  
"numeroCasa": "",  
"apto": "",  
"idUsuario": 285  
}
```

**Exemplo retorno:****Sucesso:**

```
{  
  "idEnderecoCliente": 934251  
}
```

**Falha:**

```
{  
  "error": "O idCliente informado não pertence a nenhum cliente cadastrado no  
sistema."  
}
```

## Buscar endereços de cliente

### Descrição

Busca os endereços de um cliente

### URL

[https://\[URL\]/ws/comercial/clientes/{idCliente}/enderecos](https://[URL]/ws/comercial/clientes/{idCliente}/enderecos)

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idCliente*

### Saída

JSON contendo os endereços do cliente

### Exemplo

Envio: [https://\[URL\]/ws/comercial/clientes/81593/enderecos](https://[URL]/ws/comercial/clientes/81593/enderecos)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "numero": 85,
      "idCondominio": null,
      "idEndereco": 645831,
      "bairro": "NOVO HORIZONTE",
      "nomeCidade": "CONSELHEIRO LAFAIETE",
      "latitude": "-20.6560787",
      "idCidade": 100,
      "CEP": "36402157",
      "UF": "MG",
      "idBairro": 15372,
      "complemento": "CASA A",
      "tipoDeEndereco": "PADRAO",
      "logradouro": "R. PEDRO RIBEIRO DE REZENDE",
      "problemaEntregaBoleto": false,
      "classificacaoEndereco": "1 - TESTE",
      "id": 481018,
      "referencia": null,
      "longitude": "-43.756296599999985"
    },
    {
      "numero": 45,
      "idCondominio": 3783,
      "idEndereco": 645337,
      "bairro": "LUZIA AUGUSTA",
      "nomeCidade": "CONSELHEIRO LAFAIETE",
      "latitude": "-20.6669315",
      "idCidade": 100,
      "CEP": "36404199",
      "UF": "MG",
      "idBairro": 26913,
      "complemento": null,
      "tipoDeEndereco": "PADRAO",
      "logradouro": "R. MARILIA BERNARDINA LOBO",
      "problemaEntregaBoleto": null,
      "classificacaoEndereco": "1 - TESTE",

```

```
        "id": 934249,  
        "referencia": null,  
        "longitude": "-43.7731363"  
    }  
]  
}
```

**Falha:**

```
{  
    "error": "Internal Server Error"  
}
```

## Buscar origens comerciais

### Descrição

Busca as origens comerciais cadastradas no sistema

### URL

[https://\[URL\]/ws/comercial/contratos/origens\\_comerciais](https://[URL]/ws/comercial/contratos/origens_comerciais)

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

### Saída

JSON contendo os endereços do cliente

### Exemplo

#### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "descricao": "INDICAÇÃO DE CLIENTE",
      "isIndicacao": true,
      "promocao": {
        "id": 87,
        "descricao": "INDICAÇÃO 50% EM UMA MENSALIDADE",
        "inicioVigencia": "01/08/2018",
        "fimVigencia": "29/12/2020",
        "desconto": 50.0,
        "valorPorcentagem": true,
        "numMensalidades": 1,
        "tipoDesconto": "MENSALIDADE",
        "tipoItemPromocional": "TECNOLOGIA",
        "permitePrePago": true,
        "precisaConfirmar": false,
        "cumulativa": true,
        "transfereDesconto": true,
        "ordem": 12,
        "valorBase": "VALOR_LIQUIDO",
        "tipoPromocao": null,
        "diasCarencia": null
      }
    },
    {
      "id": 3,
      "descricao": "PROMOÇÃO RADIO",
      "isIndicacao": false,
      "promocao": null
    },
    {
      "id": 4,
      "descricao": "FACEBOOK",
      "isIndicacao": false,
      "promocao": null
    },
    {
      "id": 5,
```

```
        "descricao": "OUTROS",  
        "isIndicacao": false,  
        "promocao": null  
    }  
]  
}
```

**Falha:**

```
{  
    "error": "Internal Server Error"  
}
```



## Buscar planos disponíveis para venda por tecnologia, tipo de pessoa, tipo de contrato e cidade

### Descrição

Busca os planos disponíveis para venda de acordo com parâmetros

### URL

[https://\[URL\]/ws/comercial/contratos/planos/tecnologia/{idTecnologia}/tipo\\_contrato/{idTipoContrato}/tipo\\_pessoa/{tipoPessoa}/cidade/{idCidade}](https://[URL]/ws/comercial/contratos/planos/tecnologia/{idTecnologia}/tipo_contrato/{idTipoContrato}/tipo_pessoa/{tipoPessoa}/cidade/{idCidade})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idTecnologia, idTipoContrato, tipoPessoa, idCidade*

### Saída

JSON contendo os planos disponíveis

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/contratos/planos/tecnologia/14/tipo\\_contrato/1/tipo\\_pessoa/PessoaFisica/cidade/100](https://[URL]/ws/comercial/contratos/planos/tecnologia/14/tipo_contrato/1/tipo_pessoa/PessoaFisica/cidade/100)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "ServicoInstalacao": {
        "Descricao": "CL - IVR - TAXA DE HABILITAÇÃO",
        "NumeroMaximoParcelas": 6,
        "Id": 37,
        "PermiteParcela": true,
        "Valor": 250.0
      },
      "Tecnologia": "FIBRA",
      "DadosVoz": null,
      "DescricaoPlano": "RADIO 256KBPS",
      "PossuiPlanejamentoTributario": false,
      "DadosRede": {
        "LimiteFranquia": 5,
        "SenhaPadraoRadius": null,
        "ReservaSaidaCaixa": true,
        "Velocidade": "3MB Residencial",
        "GeraSenhaRadiusAutomatico": true,
        "QuantidadeIPPermitida": 1,
        "DescricaoFranquia": "ILIMITADO",
        "GeraLoginRadiusAutomaticoCidade": null,
        "Id": 504,
        "GeraLoginRadiusAutomatico": true,
        "LoginPadraoRadius": null
      },
      "Valor": null,
      "UtilizaTabelaPreco": true,
      "TipoTecnologia": "INTERNET",
      "IdPlano": 5,
      "DadosTV": null
    },
    {
      "ServicoInstalacao": {
```

```

        "Descricao": "BCA-HABILITACAO",
        "NumeroMaximoParcelas": 10,
        "Id": 223,
        "PermiteParcela": true,
        "Valor": 0.0
    },
    "Tecnologia": "FIBRA",
    "DadosVoz": null,
    "DescricaoPlano": "FIBRA CITY MAIS - 100M",
    "PossuiPlanejamentoTributario": false,
    "DadosRede": {
        "LimiteFranquia": 5,
        "SenhaPadraoRadius": null,
        "ReservaSaidaCaixa": false,
        "Velocidade": "100MB Residencial",
        "GeraSenhaRadiusAutomatico": false,
        "QuantidadeIPPermitida": 1,
        "DescricaoFranquia": "ILIMITADO",
        "GeraLoginRadiusAutomaticoCidade": null,
        "Id": 1269,
        "GeraLoginRadiusAutomatico": false,
        "LoginPadraoRadius": null
    },
    "Valor": 135.1,
    "UtilizaTabelaPreco": false,
    "TipoTecnologia": "INTERNET",
    "IdPlano": 1158,
    "DadosTV": null
    }
}

```

**Falha:**

```

{
    "error": "Internal Server Error"
}

```

## Buscar pacotes de planos disponíveis para venda por tipo de contrato e cidade

### Descrição

Busca os pacotes disponíveis para venda de acordo com parâmetros

### URL

[https://\[URL\]/ws/comercial/contratos/pacotes/tipo\\_contrato/{idTipoContrato}/cidade/{idCidade}](https://[URL]/ws/comercial/contratos/pacotes/tipo_contrato/{idTipoContrato}/cidade/{idCidade})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idTipoContrato, idCidade*

### Saída

JSON contendo os pacotes disponíveis

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/contratos/pacotes/tipo\\_contrato/1/cidade/100](https://[URL]/ws/comercial/contratos/pacotes/tipo_contrato/1/cidade/100)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "Descricao": "PACOTE TESTE",
      "IdPacote": 1260,
      "PlanosPacote": [
        {
          "DescontoEmPorcentagem": false,
          "Tecnologia": "TELEFONIA FIXA",
          "DescontoPacote": 5.0,
          "PossuiPlanejamentoTributario": false,
          "ValorPlano": null,
          "UtilizaTabelaPreco": true,
          "TipoTecnologia": "TELEFONIA",
          "IdPlano": 673,
          "Plano": "VIAFONE FALA BEM ILIMIT"
        },
        {
          "DescontoEmPorcentagem": false,
          "Tecnologia": "LOCACAO",
          "DescontoPacote": 2.0,
          "PossuiPlanejamentoTributario": false,
          "ValorPlano": null,
          "UtilizaTabelaPreco": true,
          "TipoTecnologia": "OUTROS",
          "IdPlano": 786,
          "Plano": "LOCACAO WI-FI"
        }
      ]
    },
    {
      "Descricao": "TESTE II",
      "IdPacote": 1261,
      "PlanosPacote": [
        {
          "DescontoEmPorcentagem": false,
          "Tecnologia": "HOSTING",
          "DescontoPacote": 5.0,
```

```

        "PossuiPlanejamentoTributario": true,
        "ValorPlano": 50.0,
        "UtilizaTabelaPreco": false,
        "TipoTecnologia": "HOSPEDAGEM",
        "IdPlano": 579,
        "Plano": "HOSPEDAGEM WEB ASP - LT"
    },
    {
        "DescontoEmPorcentagem": false,
        "Tecnologia": "LOCACAO",
        "DescontoPacote": 0.0,
        "PossuiPlanejamentoTributario": true,
        "ValorPlano": 50.0,
        "UtilizaTabelaPreco": false,
        "TipoTecnologia": "OUTROS",
        "IdPlano": 787,
        "Plano": "LOCACAO AVULSA WI-FI"
    }
]
}

```

**Falha:**

```

{
  "error": "Internal Server Error"
}

```

## Buscar dados de débito de cliente

### Descrição

Busca os dados de débito do cliente

### URL

[https://\[URL\]/ws/comercial/clientes/{idCliente}/dados\\_debito](https://[URL]/ws/comercial/clientes/{idCliente}/dados_debito)

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idCliente*

### Saída

JSON contendo os endereços do cliente

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/clientes/81593/dados\\_debito](https://[URL]/ws/comercial/clientes/81593/dados_debito)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 5,
      "agencia": "8888",
      "numeroConta": "99999",
      "titularConta": "TESTE",
      "idBanco": 2,
      "dvConta": "0",
      "dvAgencia": "7"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar caixas com saídas livres próximas ao endereço do cliente

### Descrição

Busca as caixas e saídas livres próximas ao endereço do cliente

### URL

[https://\[URL\]/ws/comercial/clientes/enderecos/{idEnderecoCliente}/caixas/proximas/tipo\\_contrato/{idTipoContrato}](https://[URL]/ws/comercial/clientes/enderecos/{idEnderecoCliente}/caixas/proximas/tipo_contrato/{idTipoContrato})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idEnderecoCliente, idTipoContrato*

### Saída

JSON contendo os endereços do cliente

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/clientes/enderecos/481018/caixas/proximas/tipo\\_contrato/1](https://[URL]/ws/comercial/clientes/enderecos/481018/caixas/proximas/tipo_contrato/1)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "DistanciaRoute": null,
      "Descricao": "NOVO HORIZONTE - CAIXA 1617",
      "Saidas": [
        {
          "Status": "LIVRE",
          "Descricao": "SAIDA 8",
          "Id": 95184
        }
      ],
      "DuracaoRoute": null,
      "Id": 8614,
      "QtdeSaidas": 8,
      "IdLocationCM": null,
      "NumeroCaixa": "1617"
    },
    {
      "DistanciaRoute": null,
      "Descricao": "NOVO HORIZONTE - CAIXA 1640",
      "Saidas": [
        {
          "Status": "LIVRE",
          "Descricao": "SAIDA 3",
          "Id": 95424
        }
      ],
      "DuracaoRoute": null,
      "Id": 8647,
      "QtdeSaidas": 8,
      "IdLocationCM": null,
      "NumeroCaixa": "1640"
    }
  ]
}
```

#### Falha:

```
{  
  "error": "Internal Server Error"  
}
```

## Buscar planos prospectados de cliente

### Descrição

Busca os planos prospectados de um cliente

### URL

[https://\[URL\]/ws/comercial/clientes/{idCliente}/planos\\_prospectados](https://[URL]/ws/comercial/clientes/{idCliente}/planos_prospectados)

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idCliente*

### Saída

JSON contendo os planos prospectados

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/clientes/428740/planos\\_prospectados](https://[URL]/ws/comercial/clientes/428740/planos_prospectados)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 140135,
      "idPlano": 1162,
      "idProspecto": 1123321,
      "descricaoPlano": "FIBRA VERO MAIS - 35M (FIBRA)",
      "idPacote": null,
      "descricaoPacote": null,
      "idTecnologia": 4,
      "descricaoTecnologia": "FIBRA",
      "tipoTecnologia": "INTERNET",
      "idTipoContrato": 1,
      "descricaoTipoContrato": "RESIDENCIAL",
      "status": "PENDENTE"
    },
    {
      "id": 140136,
      "idPlano": 1164,
      "idProspecto": 1123321,
      "descricaoPlano": "FIBRA VERO MAIS - 80M (FIBRA)",
      "idPacote": null,
      "descricaoPacote": null,
      "idTecnologia": 4,
      "descricaoTecnologia": "FIBRA",
      "tipoTecnologia": "INTERNET",
      "idTipoContrato": 1,
      "descricaoTipoContrato": "RESIDENCIAL",
      "status": "PENDENTE"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```



## Buscar clientes por vendedor, ID e documento

### Descrição

Busca clientes de acordo com os parâmetros passados

### URL

[https://\[URL\]/ws/comercial/clientes/buscar/vendedor/{idVendedor}?id={id}&documento={documento}](https://[URL]/ws/comercial/clientes/buscar/vendedor/{idVendedor}?id={id}&documento={documento})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idVendedor*

QueryParam: *id (opcional), documento (opcional)*

### Saída

JSON contendo os clientes

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/clientes/buscar/vendedor/285?id=220317&documento=38885763006](https://[URL]/ws/comercial/clientes/buscar/vendedor/285?id=220317&documento=38885763006)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 428731,
      "nome": "TESTE",
      "documento": "38885763006"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## COMERCIAL – OmniChannel

### Cadastrar dados de ligação entrante na URA [MOB]

#### Descrição

Cadastra um evento de ligação vindo da URA

#### URL

[https://\[URL\]/ws/telefonica/ligacoes/cadastrar](https://[URL]/ws/telefonica/ligacoes/cadastrar)

#### Método

POST

#### Entrada

JSON contendo o ID do cliente, ramal atual, fila atual e número de origem.

#### Saída

String de FALHA OU SUCESSO

#### Campos do JSON da entrada

Tag	Tipo	Observação
idCliente*	Long	
numeroOrigem*	String	
ramal	String	
fila	String	

\*Campos obrigatórios

#### Exemplo

##### Envio:

```
{
  "idCliente": 81593,
  "numeroOrigem": "888889999",
  "ramal": "5858",
  "fila": "125"
}
```

**Retorno:** "SUCESSO" ou "FALHA"

## Busca de Atendimentos

### Descrição

Busca informações dos atendimentos de um determinado cliente de acordo com os filtros passados por parâmetro.

### URL

[https://{{url}}/ws/integracao/omnichannel/atendimentos/cliente/{idCliente: \[0-9\]+}](https://{{url}}/ws/integracao/omnichannel/atendimentos/cliente/{idCliente: [0-9]+})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idCliente*

QueryParam: *idContrato, statusContrato, tipoServico, statusAtendimento*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idCliente*	path	Number	
idContrato	query	Number	
statusContrato	query	String	Valores permitidos: AGUARDANDO INSTALACAO, HABILITADO, INATIVO TEMPORARIAMENTE, SUSPENSO PARCIALMENTE, SUSPENSO, HABILITADO EM CONFIANCA, EM NEGOCIACAO, CANCELADO, AGUARDANDO PAGAMENTO e AGUARDANDO AMPLIACAO.
tipoServico	query	String	Valores permitidos: INSTALACAO, SUPORTE, RETIRADA, MIGRACAO, ALTERACAO_ENDERECO, INTERNO, SOLICITACAO_SERVICO e UPGRADE_DOWNGRADE.
statusAtendimento	query	String	Valores permitidos: PENDENTE, VISITA_AGENDADA, AGUARDANDO_ANALISE, SOLUCIONADO, SEM_SOLUCAO, POS_ATENDIMENTO e CONTATO_CLIENTE.

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": [
    {
      "idAtendimento": 3395,
      "numeroProtocolo": "2020062214034302",
      "statusAtendimento": "SEM_SOLUCAO",
      "dataAbertura": 1592845423000,
      "dataAgendamentoVisita": 1592823600000,
      "gerouVisita": true,
      "turno": "MANHA",
      "idTipoAtendimento": 18,
      "descricaoTipoAtendimento": "MUDANÇA DE PONTO",
      "tipoServico": "SUPORTE",
    }
  ]
}
```

```

        "idContrato": 311622,
        "statusContrato": "CANCELADO",
        "idTecnicoVisita": 10019
    },
    {
        "idAtendimento": 3414,
        "numeroProtocolo": "2020062411000854",
        "statusAtendimento": "SOLUCIONADO",
        "dataAbertura": 1593007210000,
        "dataAgendamentoVisita": null,
        "gerouVisita": false,
        "turno": null,
        "idTipoAtendimento": 3,
        "descricaoTipoAtendimento": "SOLICITAÇÃO DE ALTERAÇÃO DE VENCIMENTO",
        "tipoServico": "INTERNO",
        "idContrato": 311622,
        "statusContrato": "CANCELADO",
        "idTecnicoVisita": 10019
    }
],
"status": "OK"
}

```

## Disponibilização de Datas Disponíveis para Reagendamento de Visitas Técnicas

### Descrição

Busca informações das datas disponíveis para reagendamento de visita técnica de acordo com os filtros passados por parâmetro.

### URL

[https://{{url}}/ws/integracao/omnichannel/agenda\\_tecnica/configuracoes/atendimento/{idAtendimento:\[0-9\]+}/data/{data:\[0-9\]{8}}/turno/{turno:\[A-Z\]+}](https://{{url}}/ws/integracao/omnichannel/agenda_tecnica/configuracoes/atendimento/{idAtendimento:[0-9]+}/data/{data:[0-9]{8}}/turno/{turno:[A-Z]+})

### Método

GET

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

PathParam: *idAtendimento*, *data*, *turno*

QueryParam: *qtdeDiasLimite*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idAtendimento*	path	Number	
data*	query	String	Este parâmetro precisa ser enviado obrigatoriamente no formato "ddMMyyyy".
turno*	query	String	Valores permitidos: MANHA e TARDE.
qtdeDiasLimite	query	Number	Caso este parâmetro não seja informado ou ele seja menor que 0 ou maior que 365, o limite considerado será igual a 7.

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": [
    {
      "id": 11781,
      "tecnicosEmCampo": 3,
      "quantidadeOSAgendada": 1,
      "turno": "MANHA",
      "dataProgramacao": 1607482800000
    }
  ],
  "status": "OK"
}
```

## Efetivação do Reagendamento de Visitas Técnicas

### Descrição

Realiza a efetivação do reagendamento da visita técnica de acordo com as informações passadas por parâmetro.

### URL

[https://{{url}}/ws/integracao/omnichannel/agenda\\_tecnica/configuracoes/efetivar](https://{{url}}/ws/integracao/omnichannel/agenda_tecnica/configuracoes/efetivar)

### Método

PUT

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: *idAtendimento*, *idItemConfiguracaoAgendaTecnica*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idAtendimento*	body	Number	
idItemConfiguracaoAgendaTecnica*	body	Number	

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": "Reagendamento da visita técnica do tipo de serviço 'X'
do atendimento de ID 'Y' realizado com sucesso.",
  "status": "OK"
}
```

## Enviar Linha Digitável da Fatura por SMS ao Cliente

### Descrição

Realiza o envio da linha digitável da fatura por SMS de acordo com as informações passadas por parâmetro.

### URL

[https://{{url}}/ws/integracao/omnichannel/fatura/linha\\_digitavel/enviar](https://{{url}}/ws/integracao/omnichannel/fatura/linha_digitavel/enviar)

### Método

POST

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

Body: *idFatura*

QueryParam: *tituloSMS*, *conteudoSMS*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idFatura*	body	Number	
tituloSMS	query	String	Caso este parâmetro não seja informado, o SMS será enviado sem título.
conteudoSMS	query	String	Caso este parâmetro não seja informado, o SMS será enviado com um conteúdo de mensagem padrão.

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": {
    "Lote": 0,
    "StatusMensagem": "Mensagens enviadas",
    "Status": 1,
    "CreditosGastos": 1,
    "Destinatarios": [
      {
        "Celular": "32999134412",
        "Sequencia": 679871597
      }
    ]
  },
  "status": "OK"
}
```

## Agendamento de Pagamentos

### Descrição

Realiza o agendamento do pagamento de fatura de acordo com as informações passadas por parâmetro.

### URL

<https://{{url}}/ws/integracao/omnichannel/agendar/pagamento>

### Método

POST

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

Body: *idFatura*, *dataAgendamentoPagamento*, *idTipoContatoInadimplencia*, *idMotivoInadimplencia*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idFatura*	body	Number	
dataAgendamentoPagamento*	body	String	Este parâmetro precisa ser enviado obrigatoriamente no formato "ddMMyyyy".
idTipoContatoInadimplencia*	body	Number	
idMotivoInadimplencia*	body	Number	

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": "Agendamento de pagamento realizado com sucesso. IDFaturaDadosNegociacao [X], IDContatoInadimplencia [X]",
  "status": "OK"
}
```



## Disponibilização dos Dias Disponíveis para Alteração de Vencimento de Contratos

### Descrição

Busca informações dos vencimentos ativos presentes no sistema para alteração de contratos.

### URL

<https://{{url}}/ws/integracao/omnichannel/vencimentos/ativos>

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

QueryParam: *isPrePago*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
isPrePago*	query	String	Valores permitidos: "s" e "n".

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": [
    {
      "id": 2,
      "dia": 5,
      "diaFaturamentoInicial": 6,
      "diaFaturamentoFinal": 5,
      "ativo": true,
      "prePago": false,
      "faturaMesmoMesVencimento": true
    },
    {
      "id": 4,
      "dia": 10,
      "diaFaturamentoInicial": 11,
      "diaFaturamentoFinal": 10,
      "ativo": true,
      "prePago": false,
      "faturaMesmoMesVencimento": true
    }
  ],
  "status": "OK"
}
```

## Alteração de Dia de Vencimento de Contratos

### Descrição

Realiza a alteração do dia de vencimento das faturas de um contrato de acordo com as informações passadas por parâmetro.

### URL

[https://{{url}}/ws/integracao/omnichannel/contrato/alterar/vencimento\\_contrato](https://{{url}}/ws/integracao/omnichannel/contrato/alterar/vencimento_contrato)

### Método

PUT

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: *idContrato*, *idVencimentoAtual*, *idVencimentoNovo*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idContrato*	body	Number	
idVencimentoAtual*	body	Number	
idVencimentoNovo*	body	Number	

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": "A data de vencimento foi alterada do ID 'X' para o ID 'Y' no(s) contrato(s) selecionado(s) com sucesso.",
  "status": "OK"
}
```

## Encaminhamento de E-mail com Regras de Mudanças de Titularidade

### Descrição

Realiza o envio de um e-mail com regras de mudança de titularidade para um cliente de acordo com as informações passadas por parâmetro.

### URL

https://{{url}}/ws/integracao/omnichannel/encaminhamento\_email/mudanca\_titularidade

### Método

POST

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

Body: *idCliente*

QueryParam: *tituloEmail*, *conteudoEmail*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idCliente*	body	Number	
tituloEmail	query	String	Caso este parâmetro não seja informado, o e-mail será enviado com um título de mensagem padrão.
conteudoEmail	query	String	Caso este parâmetro não seja informado, o e-mail será enviado com um conteúdo de mensagem padrão.

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": "E-mail informando as regras de alteração de titularidade enviado para o endereço 'teste@teste.teste' com sucesso.",
  "status": "OK"
}
```

## Reboot do Equipamento

### Descrição

Realiza o reboot de um equipamento de um cliente de acordo com as informações passadas por parâmetro.

### URL

<https://{{url}}/ws/integracao/omnichannel/reiniciar/equipamento>

### Método

POST

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

Body: idContrato

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idContrato*	body	Number	

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": "O equipamento do contrato de ID 'X' foi reiniciado com sucesso.",
  "status": "OK"
}
```

## COMERCIAL - Atendimentos

### Buscar tipos de atendimentos

#### Descrição

Obtém todos os tipos de atendimentos filtrados de acordo com a tecnologia do Contrato informado

#### URL

[https://\[URL\]/ws/comercial/atendimentos/tipos/contrato/{idContrato}](https://[URL]/ws/comercial/atendimentos/tipos/contrato/{idContrato})

#### Método

POST

#### Entrada

Pelo PathParam passar o código do contrato e JSON contendo usuário e senha

#### Saída

JSON contendo a lista dos tipos de atendimentos filtrados de acordo com a tecnologia vinculada ao plano do contrato escolhido, e que tenham um destino default parametrizado

#### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

#### Exemplo:

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

```
[
  {
    "CodTipoAtendimento": 1,
    "ManutencaoInterna": true,
    "TipoServico": "SUPORTE",
    "IDDestino": 7,
    "Tecnologia": "INTERNET",
    "TipoAtendimento": "ÁREA DE MANUTENÇÃO",
    "DestinoAtendimento": "Suporte"
  },
  {
    "CodTipoAtendimento": 2,
    "ManutencaoInterna": false,
    "TipoServico": "SUPORTE",
    "IDDestino": 7,
    "Tecnologia": "INTERNET",
    "TipoAtendimento": "HOTSPOT - GERAL",
    "DestinoAtendimento": "Suporte"
  }
]
```

## Buscar atendimentos

### Descrição

Obtém todos os atendimentos filtrados pelo contrato informado.

### URL

[https://\[URL\]/ws/comercial/atendimentos/contrato/{idContrato}](https://[URL]/ws/comercial/atendimentos/contrato/{idContrato})

### Método

POST

### Entrada

Pelo PathParam passar o código do contrato e JSON contendo usuário e senha

### Saída

JSON contendo a lista dos atendimentos filtrados de acordo com o contrato.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo:

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

```
{
  "atendimentos": [
    {
      "CodTipoAtendimento": 4,
      "NroProtocolo": "2016120706925342",
      "DescricaoProblema": "ACESSO LENTO",
      "CodContrato": 1,
      "GerouVisita": "Nao",
      "DestinoAtendimento": "Suporte",
      "TempoEstimado": "1-DIAS",
      "Interrupcao": false,
      "StatusAtend": "SOLUCIONADO",
      "DataVisita": "Nao gerou",
      "CodigoCliente": 81593,
      "DataAbertura": "2016-12-07"
    },
    {
      "CodTipoAtendimento": 7,
      "NroProtocolo": "2016100606871889",
      "DescricaoProblema": "SEM ACESSO",
      "CodContrato": 1,
      "GerouVisita": "Nao",
      "DestinoAtendimento": "Suporte",
      "TempoEstimado": "1-DIAS",
      "Interrupcao": false,
      "StatusAtend": "SOLUCIONADO",
      "DataVisita": "Nao gerou",

```

```
    "CodigoCliente": 81593,  
    "DataAbertura": "2016-10-06"  
  }  
]  
}
```

## Cadastrar atendimento

### Descrição

Cadastra um novo atendimento

### URL

[https://\[URL\]/ws/comercial/atendimento](https://[URL]/ws/comercial/atendimento)

### Método

POST

### Entrada

JSON contendo as informações para criar o atendimento

### Saída

Resposta informando se o atendimento foi criado.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	
idContrato	Long	
idCentroCustoOrigem	Integer	
idCentroCustoDestino	Integer	
status	String	
idTipoAtendimento	Integer	
descricao	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312",
  "idContrato": 6861,
  "idCentroCustoOrigem": 3,
  "idCentroCustoDestino": 3,
  "status": "PENDENTE",
  "idTipoAtendimento": 4,
  "descricao": "TESTE"
}
```

#### Retorno:

"CREATED": O atendimento foi criado.



## Inserir documentos em atendimento

### Descrição

Anexa documentos ao atendimento informado

### URL

https://{{url}}/ws/area/cliente/atendimento/anexar\_documento

### Método

POST

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
IDAtendimento*	body	Number	
Documentos*	body	List	
tipoDocumento*	body	String	Valores: PADRAO, INSTALACAO, DOCUMENTO, CAIXA OS, EQUIPAMENTO, VELOCIDADE, POTENCIA_SINAL, NUMERO_CLIENTES LOCALIZACAO_CAIXA
model*	body	Object	
base64*	body	Number	Base64 do arquivo
filename*	Body	String	
filesize*	Body	Number	
filetype*	Body	String	

\* Parâmetro obrigatório

### Exemplo envio:

```
{
  "IDAtendimento":4091,
  "Documentos":[
    {
      "tipoDocumento":"DOCUMENTO",
      "model":{
        "base64":
"QzpcVXNlcnNcUHJpc2NpbGxhXERvd25sb2Fkc1xURVNURV9TSU5URUdSQSAoMjMpL0cmli",
        "filename": "foto2.jpg",
        "filesize": 524280,
        "filetype": "image/jpeg"
      }
    }
  ]
}
```

### Exemplo retorno:

```
{
```

```
"Status": "0",
"Mensagem": "Documentos Anexados com sucesso.",
"Sucesso": [
  {
    "IDAtendimento": 4091,
    "Name": "foto2.jpg"
  },
  {
    "IDAtendimento": 4091,
    "Name": "foto.jpg"
  }
]
}
```

## Fechar atendimento

### Descrição

Fecha um atendimento através do protocolo

### URL

[https://\[URL\]/ws/comercial/atendimento/fechar](https://[URL]/ws/comercial/atendimento/fechar)

### Método

PUT

### Entrada

JSON contendo o protocolo do atendimento

### Saída

String de FALHA OU SUCESSO

### Campos do JSON da entrada

Tag	Tipo	Observação
protocolo*	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "protocolo": "2020012718255111"
}
```

**Retorno:** "SUCESSO" ou "FALHA"

## Abrir atendimento pelo contrato

### Descrição

Abre um atendimento através do ID do contrato

### URL

[https://\[URL\]/ws/comercial/atendimentos/cadastrar](https://[URL]/ws/comercial/atendimentos/cadastrar)

### Método

POST

### Entrada

JSON contendo o ID do contrato e a descrição do atendimento

### Saída

JSON contendo o status da chamada e o número de protocolo ou a mensagem de erro

### Campos do JSON da entrada

Tag	Tipo	Observação
idContrato*	Long	
descricao*	String	
idTipoAtendimento	Long	
idCentroCustoOrigem	Long	
idCentroCustoDestino	Long	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "idContrato": 1283367,
  "descricao": "TESTE",
  "idTipoAtendimento": 1
}
```

#### Retorno:

```
{
  "numeroProtocolo": "2020070719330228",
  "status": "CREATED"
}
```

## Informações de Atendimentos

### Descrição

Busca informações de atendimentos de acordo com os filtros passados por parâmetro.

### URL

https://{{url}}/ws/area/cliente/{idCliente: [0-9]+}/atendimentos

### Método

GET

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

PathParam: *idCliente*

QueryParam: *idContrato*, *statusContrato*, *tipoServico*, *statusAtendimento*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idContrato*	path	Number	
idContrato	query	String	
statusContrato	query	String	Valores permitidos: AGUARDANDO INSTALACAO, HABILITADO, INATIVO TEMPORARIAMENTE, SUSPENSO PARCIALMENTE, SUSPENSO, HABILITADO EM CONFIANCA, EM NEGOCIACAO, CANCELADO, AGUARDANDO PAGAMENTO e AGUARDANDO AMPLIACAO.
tipoServico	query	String	Valores permitidos: INSTALACAO, SUPORTE, RETIRADA, MIGRACAO, ALTERACAO_ENDERECO, INTERNO, SOLICITACAO_SERVICO e UPGRADE_DOWNGRADE.
statusAtendimento	query	String	Valores permitidos: PENDENTE, VISITA_AGENDADA, AGUARDANDO_ANALISE, SOLUCIONADO, SEM_SOLUCAO, POS_ATENDIMENTO e CONTATO_CLIENTE.

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": [
    {
      "idAtendimento": 4092,
      "numeroProtocolo": "2020102809461371",
      "statusAtendimento": "VISITA_AGENDADA",
      "dataAbertura": 1603889173000,
      "dataFechamento": null,
      "dataAgendamentoVisita": 1603882800000,
      "gerouVisita": true,
      "turno": "MANHA",
      "idTipoAtendimento": 21,
      "descricaoTipoAtendimento": "ATIVACÃO INICIAL",
      "tipoServico": "INSTALACAO",
      "idContrato": 311962,
      "nomePlano": "SUPER LIGEIRO TOP - PRE",
    }
  ]
}
```

```
        "statusContrato": "HABILITADO",  
        "idTecnicoVisita": null  
    },  
    ],  
    "status": "OK"  
}
```

## Visita Técnica - Disponibilização de Datas Disponíveis para Reagendamento de Visitas Técnicas

### Descrição

Busca informações das datas disponíveis para reagendamento de visita técnica de acordo com os filtros passados por parâmetro.

### URL

https://{{url}}/ws/servicos/agendamento/agenda\_tecnica/configuracoes/atendimento/{idAtendimento: [0-9]+}/data/{data: [0-9]{8}}/turno/{turno: [A-Z]+}

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idAtendimento, data, turno*

QueryParam: *qtdeDiasLimite*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idAtendimento*	path	Number	
data*	path	String	Este parâmetro precisa ser enviado obrigatoriamente no formato "ddMMyyyy".
turno*	path	String	Valores permitidos: MANHA e TARDE.
qtdeDiasLimite	query	Number	Caso este parâmetro não seja informado ou ele seja menor que 0 ou maior que 365, o limite considerado será igual a 7.

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": [
    {
      "id": 11781,
      "tecnicosEmCampo": 3,
      "quantidadeOSApendada": 1,
      "turno": "MANHA",
      "dataProgramacao": 1607482800000
    }
  ],
  "status": "OK"
}
```

## Visita Técnica - Busca de Atendimentos

### Descrição

Busca informações dos atendimentos com status de visita agendada de um determinado cliente de acordo com os filtros passados por parâmetro.

### URL

https://{{url}}/ws/servicos/agendamento/atendimentos/cliente/{idCliente: [0-9]+}

### Método

GET

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

PathParam: *idCliente*

QueryParam: *tipoServico*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idCliente*	path	Number	
tipoServico	query	String	Valores permitidos: INSTALACAO, SUPORTE, RETIRADA, MIGRACAO, ALTERACAO_ENDERECO, INTERNO, SOLICITACAO_SERVICO e UPGRADE_DOWNGRADE.

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": [
    {
      "idAtendimento": 3395,
      "numeroProtocolo": "2020062214034302",
      "statusAtendimento": "VISITA_AGENDADA",
      "dataAbertura": 1592845423000,
      "dataAgendamentoVisita": 1592823600000,
      "gerouVisita": true,
      "turno": "MANHA",
      "idTipoAtendimento": 18,
      "descricaoTipoAtendimento": "MUDANÇA DE PONTO",
      "tipoServico": "SUPORTE",
      "idContrato": 311622,
      "statusContrato": "CANCELADO",
      "idTecnicoVisita": 10019
    },
    {
      "idAtendimento": 3414,
      "numeroProtocolo": "2020062411000854",
      "statusAtendimento": "VISITA_AGENDADA",
      "dataAbertura": 1593007210000,
      "dataAgendamentoVisita": null,
      "gerouVisita": false,
      "turno": null,
      "idTipoAtendimento": 3,
    }
  ]
}
```



```
        "descricaoTipoAtendimento": "SOLICITAÇÃO DE ALTERAÇÃO DE VENCIMENTO",
        "tipoServico": "INTERNO",
        "idContrato": 311622,
        "statusContrato": "CANCELADO",
        "idTecnicoVisita": 10019
    }
],
    "status": "OK"
}
```

## Visita Técnica - Cancelar Reagendamento de Visitas Técnicas

### Descrição

Realiza o cancelamento do reagendamento da visita técnica de acordo com as informações passadas por parâmetro.

### URL

https://{{url}}/ws/servicos/agendamento/agenda\_tecnica/configuracoes/cancelar

### Método

PUT

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

Body: *idAtendimento*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idAtendimento*	body	Number	

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": "Reagendamento da visita técnica do tipo de serviço 'X'
do atendimento de ID 'Y' cancelado com sucesso.",
  "status": "OK"
}
```

## Visita Técnica - Efetivação do Reagendamento de Visitas Técnicas

### Descrição

Realiza a efetivação do reagendamento da visita técnica de acordo com as informações passadas por parâmetro.

### URL

https://{{url}}/ws/servicos/agendamento/agenda\_tecnica/configuracoes/efetivar

### Método

PUT

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

Body: *idAtendimento, idItemConfiguracaoAgendaTecnica*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idAtendimento*	body	Number	
idItemConfiguracaoAgendaTecnica*	body	Number	

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": "Reagendamento da visita técnica do tipo de serviço 'X'
do atendimento de ID 'Y' realizado com sucesso.",
  "status": "OK"
}
```

## Visita Técnica - Atribuição de Notas em Atendimentos

### Descrição

Realiza a atribuição de notas em atendimentos de acordo com as informações passadas por parâmetro.

### URL

https://{{url}}/ws/servicos/agendamento/atendimento/atribuir\_nota

### Método

POST

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

Body: *idAtendimento*, *notaTecnico*, *notaResolucao*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idAtendimento*	body	Number	
notaTecnico*	body	Number	Valor em ponto flutuante com duas casas decimais separado por ponto. Ex: 4.20
notaResolucao*	body	Number	Valor em ponto flutuante com duas casas decimais separado por ponto. Ex: 6.69

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": [
    {
      "message": "Pontuações atribuídas ao atendimento de ID 'X' com sucesso."
    },
    {
      "id": 13752,
      "dataInsercao": 1608578303858,
      "idUserario": 102160,
      "texto": "Pontuações do atendimento: \n\n - Nome do técnico: Teste Teste \n - Nota do técnico: 4,20\n - Nota da resolução: 6,69"
    }
  ],
  "status": "OK"
}
```

## Visita Técnica - Busca Notas de Atendimentos

### Descrição

Busca informações das notas dos atendimentos de acordo com os filtros passados por parâmetro.

### URL

https://{url}/ws/servicos/agendamento/notas/atendimento/{idAtendimento: [0-9]+}

### Método

GET

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

PathParam: *idAtendimento*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idAtendimento*	path	Number	

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": [
    {
      "id": 13752,
      "dataInsercao": 1608578303858,
      "idUsuario": 102160,
      "texto": "Pontuações do atendimento: \n\n - Nome do técnico: Teste Teste \n - Nota do técnico: 4,20\n - Nota da resolução: 6,69",
      "nomeTecnico": "Teste Teste",
      "notaTecnico": "4,20",
      "notaResolucao": "6,69",
      "pontuacaoAtendimento": true
    }
  ],
  "status": "OK"
}
```

## Encaminhamento de E-mail com Informações do Atendimento

### Descrição

Realiza o encaminhamento de um e-mail com os dados do atendimento de acordo com as informações passadas por parâmetro.

### URL

https://{{url}}/ws/area/cliente//protocolo/atendimento/enviar

### Método

POST

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

Body: *idAtendimento*

QueryParam: *tituloEmail*, *conteudoEmail*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idAtendimento*	body	Number	
tituloEmail	query	String	Caso este parâmetro não seja informado, o e-mail será enviado com título padrão.
conteudoEmail	query	String	Caso este parâmetro não seja informado, o e-mail será enviado com um conteúdo de mensagem padrão.

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": "Os dados do atendimento de ID '4106' foi enviado com sucesso para o endereço 'gian.honorio@adaptersolucoes.com'.",
  "status": "OK"
}
```

## Buscar dias e quantidades de visitas disponíveis por dia para agendamento de acordo com tecnologia, tipo de serviço, tipo de contrato e cidade

### Descrição

Busca os dias e quantidade de visitas disponíveis para agendamentos de acordo com parâmetros

### URL

[https://\[URL\]/ws/servicos/agendamento/dias\\_disponiveis/tipo\\_servico/{tipoServico}/tecnologia/{idTecnologia}/cidade/{idCidade}/tipo\\_contrato/{idTipoContrato}](https://[URL]/ws/servicos/agendamento/dias_disponiveis/tipo_servico/{tipoServico}/tecnologia/{idTecnologia}/cidade/{idCidade}/tipo_contrato/{idTipoContrato})

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

PathParam: tipoServico, idTecnologia, idCidade, idTipoContrato

### Saída

JSON contendo lista dos dias disponíveis para agendamento ou JSON contendo mensagem de erro

### Exemplo

**Envio:** [https://\[URL\]/ws/servicos/agendamento/dias\\_disponiveis/tipo\\_servico/INSTALACAO/tecnologia/4/cidade/100/tipo\\_contrato/1](https://[URL]/ws/servicos/agendamento/dias_disponiveis/tipo_servico/INSTALACAO/tecnologia/4/cidade/100/tipo_contrato/1)

**OBS.: Parâmetro “tipoServico” deverá receber um dos valores: INSTALACAO, SUPORTE, RETIRADA, MIGRACAO, ALTERACAO\_ENDERECO, INTERNO, SOLICITACAO\_SERVICO, UPGRADE\_DOWNGRADE**

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 286434,
      "dataProgramacao": 1623812400000,
      "tecnicosEmCampo": 5,
      "quantidadeOSAgendada": 0,
      "tipoServico": "INSTALACAO",
      "turno": "MANHA",
      "idTipoContrato": 1,
      "idTecnologia": 4,
      "descricaoTecnologia": "FIBRA"
    },
    {
      "id": 286437,
      "dataProgramacao": 1623898800000,
      "tecnicosEmCampo": 5,
      "quantidadeOSAgendada": 0,
      "tipoServico": "INSTALACAO",
      "turno": "MANHA",
      "idTipoContrato": null,
      "idTecnologia": 4,
      "descricaoTecnologia": "FIBRA"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar fechamentos de atendimentos externos por tipo de atendimento

### Descrição

Busca os fechamentos de atendimento externos pelo tipo de atendimento

### URL

[https://\[URL\]/ws/comercial/atendimentos/fechamentos?idTipoAtendimento={id}](https://[URL]/ws/comercial/atendimentos/fechamentos?idTipoAtendimento={id})

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

QueryParam: idTipoAtendimento (obrigatório)

### Saída

JSON contendo lista dos dados ou JSON contendo mensagem de erro

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/atendimentos/fechamentos?idTipoAtendimento=999](https://[URL]/ws/comercial/atendimentos/fechamentos?idTipoAtendimento=999)

### Retorno:

#### Sucesso:

```
[
  {
    "id": 217,
    "descricao": "2097 - PROBLEMA NÃO INDENTIFICADO - RELATAR",
    "assinaturaDigitalObrigatoria": true,
    "anexaxaoImagensConclusaoVisitaObrigatoria": true,
    "solucionado": true
  },
  {
    "id": 338,
    "descricao": "3053 - RESOLVIDO SEM INTERVENÇÃO",
    "assinaturaDigitalObrigatoria": false,
    "anexaxaoImagensConclusaoVisitaObrigatoria": false,
    "solucionado": false
  },
  {
    "id": 13150,
    "descricao": "5007 - SEM SOLUCAO",
    "assinaturaDigitalObrigatoria": false,
    "anexaxaoImagensConclusaoVisitaObrigatoria": false,
    "solucionado": false
  }
]
```

#### Sem resultado:

```
{
  "error": "Não há fechamentos de atendimentos externos vinculados ao tipo de atendimento informado."
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```



## Cadastrar log de atendimento

### Descrição

Cadastra um log de atendimento

### URL

[https://\[URL\]/ws/comercial/atendimentos/log/cadastrar](https://[URL]/ws/comercial/atendimentos/log/cadastrar)

### Método

POST

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

Body: Json contendo os dados do log

### Saída

JSON contendo o ID do log cadastrado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Long	
acao*	body	String	
tipoAcao*	body	String	Valores: CADASTRO, ATUALIZAR
idAtendimento*	body	Long	

**\* Parâmetro obrigatório**

### Exemplo entrada:

```
{
  "idUsuario": 285,
  "acao": "teste",
  "tipoAcao": "CADASTRO",
  "idAtendimento": 4049282
}
```

### Exemplo retorno:

#### Sucesso:

```
{
  "idLogAtendimento": 3603258
}
```

#### Em caso de atendimento não encontrado:

```
{
  "error": "Atendimento nao encontrado"
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Reverter status de atendimento para PENDENTE

### Descrição

Reverte o status de um atendimento para PENDENTE

### URL

[https://\[URL\]/ws/comercial/atendimentos/reverter](https://[URL]/ws/comercial/atendimentos/reverter)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados para alteração do atendimento

### Saída

JSON contendo o ID do atendimento alterado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Long	
idCentroCusto*	body	String	
idAtendimento*	body	Long	

**\* Parâmetro obrigatório**

### Exemplo entrada:

```
{
  "idUsuario": 285,
  "idAtendimento": 4049282,
  "idCentroCusto": 1
}
```

### Exemplo retorno:

#### Sucesso:

```
{
  "idAtendimento": 4049282
}
```

#### Em caso de atendimento não encontrado:

```
{
  "error": "Atendimento nao encontrado"
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar dados detalhados do atendimento

### Descrição

Busca dados detalhados do atendimento

### URL

[https://\[URL\]/ws/comercial/atendimentos/{id}](https://[URL]/ws/comercial/atendimentos/{id})

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

PathParam: id

### Saída

JSON contendo os dados ou JSON contendo mensagem de erro

### Exemplo

Envio: [https://\[URL\]/ws/comercial/atendimentos/4049332](https://[URL]/ws/comercial/atendimentos/4049332)

### Retorno:

#### Sucesso:

```
{
  "IDAtendimento": 4049332,
  "Descricao": "Fidelidade: 12 MESES - SERVIÇO: NÃO POSSUI\nDATA: 16/09/2021",
  "NumeroProtocolo": "2021091615531377",
  "IDTipoAtendimento": 51,
  "Status": "PENDENTE",
  "DataAbertura": 1631818394000,
  "DataAgendamentoVisita": null,
  "IDCentroCustoDestino": 3,
  "Contrato": {
    "IDContrato": 1283658,
    "IDTipoContrato": 1,
    "Cliente": {
      "IDCliente": 81593,
      "Nome": "CLIENTE TESTE",
      "Telefone": "3133331111",
      "Celular": "31988884444"
    }
  },
  "Plano": "FIBRA VERO MAIS - 15M",
  "Velocidade": "15MB Residencial",
  "DadosRede": {
    "SerialONU": null,
    "MACONU": null
  },
  "DadosCaixa": {
    "DescricaoCaixa": "NOVO HORIZONTE - CAIXA 1640",
    "DescricaoSaida": "SAIDA 3",
    "CodSaida": 95424,
    "CodCaixa": 8647,
    "Mapeamento": "[OLT LAFA 12 <i class='fa fa-chevron-right' aria-hidden='true'></i> PONCARD 11 <i class='fa fa-chevron-right' aria-hidden='true'></i> PORTA 2, DIO 72 FO PROGRESSO ...",
    "IDComponenteCM": 22,
    "IDLocationCM": 1,
    "IDPortaComponenteCM": 5
  },
  "Enderecos": [
    {
```

```

        "IDEnderecoContrato": 1291299,
        "Complemento": "APTO 1",
        "Numero": 152,
        "tipoDeEndereco": "PADRAO",
        "Latitude": "-20.6560787",
        "Longitude": "-43.756296599999985",
        "Referencia": null,
        "IDClassificacaoEndereco": 1,
        "ClassificacaoEndereco": "TESTE",
        "NomeBairro": "NOVO HORIZONTE",
        "IDBairro": 15372,
        "NomeCidade": "CONSELHEIRO LAFAIETE",
        "IDCidade": 100,
        "Estado": "MG",
        "Endereco": "R. PEDRO RIBEIRO DE REZENDE",
        "CEP": "36402157",
        "Condominio": {
            "Descricao": "TESTE CONDOMINIO CM",
            "Numero": "5",
            "IDBloco": 2,
            "Apto": null,
            "NumeroCasa": null,
            "IDCondominio": 1,
            "Andar": null
        }
    }
}
]
}

```

**Em caso de atendimento não encontrado:**

```

{
    "error": "Atendimento nao encontrado"
}

```

**Falha:**

```

{
    "error": "Internal Server Error"
}

```

## COMERCIAL – Infobip

### Busca Último Registro de Log de Integração com a API da Infobip

#### Descrição

Busca informações do último registro de log de integração com a API da Infobip, cuja data/hora de verificação esteja dentro da última hora, de acordo com os filtros passados por parâmetro.

#### URL

https://{{url}}/ws/integracao/infobip/ultimo/{telefone: [0-9]+}

#### Método

GET

#### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

PathParam: *telefone*

#### Saída

JSON contendo os dados solicitados.

#### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
telefone*	path	Number	

\* Parâmetro obrigatório

#### Exemplo retorno:

```
{
  "data": {
    "id": 1,
    "idAplicacao": "DE0342851CD1D22053FB62D054DAE3E3",
    "idMensagem": "E871BD63E69816FC17C5A72D10D1B47F",
    "idPin": "EEA2D9A0F876B52EED8D599CEF1A03F3",
    "pin": 1425,
    "telefone": "5532988882222",
    "statusNC": "NC_DESTINATION_REACHABLE",
    "statusSMS": "MESSAGE_SENT",
    "tentativasRestantes": 0,
    "verificado": true,
    "erroPin": null,
    "erroRequisicao": null,
    "dataVerificacao": 1621887553000,
    "dataLog": 1621886963000,
    "idUsuario": 1
  },
  "status": "OK"
}
```

## Atualiza Registros de Log de Integração com a API da Infobip

### Descrição

Realiza a atualização de registros de log de integração com a API da Infobip de acordo com os filtros passados por parâmetro.

### URL

https://{url}/ws/integracao/infobip/verificar/pin/{pin: [0-9]+}

### Método

PUT

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

PathParam: *pin*

Body: *pinId*, *msisdn*, *verified*, *attemptsRemaining*, *pinError*, *requestError*

### Saída

JSON contendo os dados solicitados.

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
pin*	path	Number	
pinId*	body	Number	
msisdn	body	String	
verified*	body	Boolean	
attemptsRemaining	body	Number	
requestError	body	Object	No seguinte formato: requestError: {serviceExceptionResponse: {"messageId": "", "text": ""}}

\* Parâmetro obrigatório

### Exemplo retorno:

```
{
  "data": {
    "id": 1,
    "idAplicacao": "DE0342851CD1D22053FB62D054DAE3E3",
    "idMensagem": "E871BD63E69816FC17C5A72D10D1B47F",
    "idPin": "EEA2D9A0F876B52EED8D599CEF1A03F3",
    "pin": 6969,
    "telefone": "5532999134412",
    "statusNC": "NC_DESTINATION_REACHABLE",
    "statusSMS": "MESSAGE_SENT",
    "tentativasRestantes": 0,
    "verificado": true,
    "erroPin": null,
    "erroRequisicao": null,
    "dataVerificacao": 1621972623196,
    "dataLog": 1621886963000,
    "idUsuario": 1
  },
  "status": "OK"
}
```

## COMERCIAL – Configurações

### Buscar versões do app de vendas

#### Descrição

Busca as informações das versões do app de vendas

#### URL

https://{{url}}/ws/comercial/app\_vendas/versao

#### Método

GET

#### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

#### Saída

JSON contendo os dados solicitados.

#### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	

\* Parâmetro obrigatório

#### Exemplo retorno:

```
{  
  "VERSAO_FINAL_APP_VENDAS": "1.0",  
  "VERSAO_FINAL_NOME_APP_VENDAS": "teste"  
}
```

## FINANCEIRO

### Baixar PDF da fatura

#### Descrição

Faz download da fatura.

#### URL

[https://\[URL\]/ws/financeiro/fatura/{idFatura}?isChatbot=s](https://[URL]/ws/financeiro/fatura/{idFatura}?isChatbot=s)

#### Método

POST

#### Entrada

Pelo PathParam passar o código da fatura,

Por QueryParam passa se for utilizar o Chatbot, se não for informa não precisa passar

JSON contendo usuário e senha

#### Saída

PDF da fatura.

#### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

#### Exemplo

##### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

Caso o parâmetro "isChatbot" seja enviado o corpo da requisição deverá ter o seguinte formato:

```
{
  "usuario": "12312312312"
}
```

##### Retorno:

PDF da fatura.



## Baixar PDF da fatura 2

### Descrição

Faz download da fatura.

### URL

[https://\[URL\]/ws/financeiro/fatura/{idFatura}/pdf?isChatbot=s](https://[URL]/ws/financeiro/fatura/{idFatura}/pdf?isChatbot=s)

### Método

POST

### Entrada

Pelo PathParam passar o código da fatura,

Por QueryParam passa se for utilizar o Chatbot, se não for informa não precisa passar

JSON contendo usuário e senha

### Saída

PDF da fatura.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

Caso o parâmetro "isChatbot" seja enviado o corpo da requisição deverá ter o seguinte formato:

```
{
  "usuario": "12312312312"
}
```

#### Retorno:

PDF da fatura.

## Baixar PDF da fatura 3

### Descrição

Faz download da fatura.

### URL

[https://\[URL\]/ws/financeiro/fatura/{idFatura}/imprimir](https://[URL]/ws/financeiro/fatura/{idFatura}/imprimir)

### Método

GET

### Entrada

Pelo PathParam passar o código da fatura

### Saída

PDF da fatura.

### Exemplo

#### Envio:

[https://\[URL\]/ws/financeiro/fatura/123456/imprimir](https://[URL]/ws/financeiro/fatura/123456/imprimir)

#### Retorno:

PDF da fatura.

## Baixar PDF de nota fiscal

### Descrição

Obtém a nota fiscal em pdf

### URL

[https://\[URL\]/ws/financeiro/notasfiscais/{idNotaFiscal}](https://[URL]/ws/financeiro/notasfiscais/{idNotaFiscal})

### Método

POST

### Entrada

Pelo PathParam passar o código da nota fiscal,  
JSON contendo usuário e senha

### Saída

PDF da nota fiscal escolhida.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

PDF da nota fiscal escolhida.

## Buscar notas fiscais

### Descrição

Buscas as notas fiscais por um período informado

### URL

[https://\[URL\]/ws/financeiro/notasfiscais/cliente/{idCliente}?dataInicio={dataInicio}&dataFim={dataFim}](https://[URL]/ws/financeiro/notasfiscais/cliente/{idCliente}?dataInicio={dataInicio}&dataFim={dataFim})

### Método

POST

### Entrada

Pelo PathParam passar o código do cliente,

Por QueryParam passar a data inicial e data final no formato dd-MM-yyyy.

JSON contendo usuário e senha

### Saída

JSON contendo a lista de notas fiscais filtradas de acordo com o ID do cliente.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

```
[
  {
    "id": 51,
    "idCliente": 3,
    "vencimento": 1491015600000,
    "dataEmissao": 1490711456000,
    "valor": 132,
    "numeroNF": 18,
    "idEmpresa": 4,
    "modeloNotaFiscal": {
      "tipoModelo": "OFFLINE",
      "id": 1,
      "descricao": "MODELO 22",
      "nomeArquivoJasper": "MODELO_NF_OFFLINE_1.jasper",
      "idServidorIntegracaoHomologacao": 1,
      "idServidorIntegracaoProducao": null,
      "serie": "A22",
      "codigoModelo": null,
      "arquivoJasper": null,
      "convenio115": false,
      "prefeitura": false,
      "sefaz": false,
      "offline": true
    }
  }
]
```

```

},
"dataCancelamento": null,
"notaUnica": null,
"itensNota": [
  {
    "id": 63,
    "descricao": "SCM(01/03/17 a 31/03/17)",
    "valorItem": 66,
    "baseICMS": 66,
    "valorICMS": 17.82,
    "valorIsentoICMS": 0,
    "baseISS": 0,
    "valorISS": 0,
    "baseIRRF": 0,
    "basePIS": 0,
    "valorPIS": 0,
    "baseCOFINS": 0,
    "valorCOFINS": 0,
    "baseCSLL": 0,
    "valorCSLL": 0,
    "aliquotaICMS": 27,
    "aliquotaPIS": 0,
    "aliquotaCOFINS": 0,
    "valorOutrosICMS": null,
    "valorIRRF": 0
  }
],
"situacao": "N",
"chaveCodificacaoDigital": "dbee783e95f55e9e0e727a6c772a85bd",
"descricaoModelo": "MODELO 22"
}
]

```

## Buscar código de barras da fatura

### Descrição

Buscas as notas fiscais por um período informado

### URL

[https://\[URL\]/ws/financeiro/fatura/{idFatura}/linha\\_digitavel](https://[URL]/ws/financeiro/fatura/{idFatura}/linha_digitavel)

### Método

POST

### Entrada

Pelo PathParam passar o código da fatura,  
JSON contendo usuário e senha

### Saída

JSON contendo o código de barras e o valor da fatura de acordo com o ID da fatura.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

```
{
  "linhaDigitavel": "03399.31479 64900.000239 56580.201012 8 71550000018170",
  "valor": 181.7
}
```

## Validar declaração quitação anual de débito

### Descrição

Valida declaração quitação anual de débito

### URL

[https://\[URL\]/ws/financeiro/faturamento/valida/quitacao\\_anual\\_debito?  
dataInicio={dataInicio}&dataFim={dataFim}&idCliente={idCliente}](https://[URL]/ws/financeiro/faturamento/valida/quitacao_anual_debito?dataInicio={dataInicio}&dataFim={dataFim}&idCliente={idCliente})

### Método

POST

### Entrada

Pelo QueryParam passar o código do cliente e a data inicial e data final no formato dd-MM-yyyy, JSON contendo usuário e senha

### Saída

JSON contendo um Map com a validação de débitos do cliente.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

Em caso de sucesso:

```
{
  "success": "Nao existem debitos em aberto no periodo selecionado"
}
```

Ou em caso de erro na alteração do status:

```
{
  "error": "Existem debitos em aberto no periodo selecionado"
}
```

## Baixar PDF da declaração de quitação anual de débito

### Descrição

Declaração quitação anual de débito

### URL

[https://\[URL\]/ws/financeiro/faturamento/quitacao\\_anual\\_debito?  
dataInicio={dataInicio}&dataFim={dataFim}&idCliente={idCliente}](https://[URL]/ws/financeiro/faturamento/quitacao_anual_debito?dataInicio={dataInicio}&dataFim={dataFim}&idCliente={idCliente})

### Método

POST

### Entrada

Pelo QueryParam passar o código do cliente e a data inicial e data final no formato dd-MM-yyyy, JSON contendo usuário e senha

### Saída

PDF da declaração anual de débito para o período escolhido.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{  
  "usuario": "12312312312",  
  "senha": "12312312312"  
}
```

#### Retorno:

PDF da declaração anual de débito para o período escolhido.



## Buscar vencimentos de contrato

### Descrição

Retorna a lista de vencimentos liberados

### URL

[https://\[URL\]/ws/financeiro/buscar/vencimentos](https://[URL]/ws/financeiro/buscar/vencimentos)

### Método

POST

### Entrada

JSON contendo usuário e senha

### Saída

Map contendo as informações de todos os vencimentos possíveis para a alteração do contrato.

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

#### Retorno:

```
[
  {
    "id": 1,
    "dia": 1,
    "diaFaturamentoInicial": 6,
    "diaFaturamentoFinal": 5,
    "ativo": true,
    "prePago": false
  },
  {
    "id": 2,
    "dia": 5,
    "diaFaturamentoInicial": 10,
    "diaFaturamentoFinal": 9,
    "ativo": true,
    "prePago": false
  }
]
```

## Buscar faturas por CPF

### Descrição

## Retorna as Faturas por CPF

## URL

[https://\[URL\]/ws/financeiro/faturas/cliente](https://[URL]/ws/financeiro/faturas/cliente)

## Método

## POST

## Entrada

## JSON contendo usuário e senha

## Saída

Map contendo as informações sobre as faturas abertas do cliente.

## Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	

\*Campos obrigatórios

## Exemplo

**Envio:**

```
{
  "usuario": "12312312312"
}
```

**Retorno:**

```
{
  "Contratos": [
    {
      "ValorContrato": 14.9,
      "IDContrato": 111222,
      "DescricaoPlano": "LOCACAO DE EQUIPAMENTOS"
    },
    {
      "ValorContrato": 154.1,
      "IDContrato": 222333,
      "DescricaoPlano": "FIBRA 300MB ILIMITADO"
    }
  ],
  "Faturas": [
    {
      "idsContratos": [111222, 222333],
      "DebitoAutomatico": false,
      "CodigoCliente": 8768696,
      "linhaDigitavel": "11111.11111 11111.11111 11111.11111 1 11111111111111",
      "IDFatura": 8768696,
      "Valor": 125.0,
      "codigoBarras": "1111111111111111111111111111111111111111",
      "Vencimento": "2020-03-10"
    }
  ],
  "Cliente": "CLIENTE TESTE"
}
```

## Buscar faturas em aberto por cliente

### Descrição

Retorna as faturas sem pagamento do cliente

### URL

[https://\[URL\]/ws/financeiro/faturas/cliente/{{idCliente}}](https://[URL]/ws/financeiro/faturas/cliente/{{idCliente}})

### Método

GET

### Entrada

PathParam contendo o ID do cliente

### Saída

List de map contendo as informações sobre as faturas em aberto do cliente.

### Exemplo

#### Envio:

[https://\[URL\]/ws/financeiro/faturas/cliente/366890](https://[URL]/ws/financeiro/faturas/cliente/366890)

#### Retorno:

```
[
  {
    "VencimentoFormatado": "10/05/2020",
    "CodigoCliente": 366890,
    "IDFatura": 89,
    "Valor": 112.4,
    "Contratos": [1282441]
  },
  {
    "VencimentoFormatado": "22/06/2020",
    "CodigoCliente": 366890,
    "IDFatura": 8,
    "Valor": 109.9,
    "Contratos": [1282441]
  },
  {
    "VencimentoFormatado": "22/07/2020",
    "CodigoCliente": 366890,
    "IDFatura": 9,
    "Valor": 109.9,
    "Contratos": [1282441]
  }
]
```

## Realizar negociação

### Descrição

Realiza a negociação de faturas em aberto, através de dados das parcelas fornecidos

### URL

[https://\[URL\]/ws/financeiro/negociar](https://[URL]/ws/financeiro/negociar)

### Método

PUT

### Entrada

JSON contendo usuário e senha, e lista com os dados das parcelas da negociação

### Saída

List contendo os dados das novas faturas de negociação geradas

### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	
dadosNegociacao*	List<Map<String,Object>	

\*Campos obrigatórios

**OBS.:** o Map dos dados de negociação deverá conter os seguintes campos nos seguintes formatos:

numeroParcela (int),  
valorTotalParcela (Double),  
valorMultaParcela (Double),  
valorJurosParcela (Double),  
valorDescontoParcela (Double),  
vencimentoParcela (String – formato dd-mm-aaaa),  
idsFaturasNegociadas (List<Long>)

### Exemplo

#### Entrada:

```
{  
  "usuario": "12312312312",  
  "senha": "12312312312",  
  "dadosNegociacao": [  
    {  
      "numeroParcela": 1,  
      "valorTotalParcela": 112.4,  
      "valorMultaParcela": 0.5,  
      "valorJurosParcela": 2.0,  
      "vencimentoParcela": "10-04-2020",  
      "valorDescontoParcela": 0.0,  
      "idsFaturasNegociadas": [4,3]  
    },  
    {  
      "numeroParcela": 2,  
      "valorTotalParcela": 112.4,  
      "valorMultaParcela": 0.5,  
      "valorJurosParcela": 2.0,  
      "vencimentoParcela": "10-05-2020",  
      "valorDescontoParcela": 0.0,  
      "idsFaturasNegociadas": [4,3]  
    }  
  ]  
}
```

**Retorno:**

Em caso de sucesso:

[illegible]

Em caso de falha:

```
{
  "error": "Erro ao realizar negociação",
  "internalError": null
}
```

## Enviar faturas em aberto para o e-mail do cliente

### Descrição

Envia faturas em aberto para o e-mail do cliente

### URL

[https://\[URL\]/ws/financeiro/faturas/abertas/{idCliente}/enviar](https://[URL]/ws/financeiro/faturas/abertas/{idCliente}/enviar)

### Método

POST

### Entrada

JSON contendo a lista de ids das faturas que deseja enviar e ID do cliente passado na URL como PathParam

### Campos do JSON da entrada

Tag	Tipo	Observação
idsFaturas*	List<Long>	

\*Campos obrigatórios

### Exemplo

#### Envio:

```
{  
  "idsFaturas": [90, 91]  
}
```

### Saída

String de “FALHA” ou “SUCESSO”

## Buscar linha digitável da fatura

### Descrição

Busca linha digitável através do ID da fatura

### URL

[https://\[URL\]/ws/financeiro/fatura/aberta/{IdFatura}/linha\\_digitavel](https://[URL]/ws/financeiro/fatura/aberta/{IdFatura}/linha_digitavel)

### Método

GET

### Entrada

ID da fatura passado na URL como PathParam

### Saída

String de com a linha digitável ou “FALHA”

### Exemplo

#### Envio:

[https://\[URL\]/ws/financeiro/fatura/aberta/40/linha\\_digitavel](https://[URL]/ws/financeiro/fatura/aberta/40/linha_digitavel)

#### Retorno:

##### Se sucesso:

"11111.11111 11111.11111 11111.11111 1 11111111111111"

##### Se falha:

"FALHA"





**OU**

**OU "FALHA"**

## Buscar faturas em aberto por CPF/CNPJ

### Descrição

Retorna as faturas sem pagamento do cliente

### URL

[https://\[URL\]/ws/financeiro/faturas/cliente/documento/{cpfcpnpj}](https://[URL]/ws/financeiro/faturas/cliente/documento/{cpfcpnpj})

### Método

GET

### Entrada

PathParam contendo o CPF ou CNPJ

### Saída

List de Map contendo as informações sobre as faturas encontradas

### Exemplo

#### Envio:

[https://\[URL\]/ws/financeiro/faturas/cliente/documento/1231231232](https://[URL]/ws/financeiro/faturas/cliente/documento/1231231232)

#### Retorno:

```
[
  {
    "VencimentoFormatado": "15/07/2020",
    "CodigoCliente": 308254,
    "ValorMulta": 1.8112000000000001,
    "IDFatura": 366189,
    "Contratos": [
      311649
    ],
    "ValorJuros": 14.3718720000000002,
    "nroBoleto": "12241008",
    "Valor": 90.56,
    "nroDocumento": "0720311597",
    "DebitoAutomatico": false
  }
]
```

## Buscar bandeiras de cartão

### Descrição

Busca todas as bandeiras de cartão

### URL

[https://\[URL\]/ws/financeiro/bandeiras\\_cartao](https://[URL]/ws/financeiro/bandeiras_cartao)

### Método

GET

### Entrada

HeaderParam("Token") com o value contendo um token válido, gerado pelo método "Gerar Token".  
QueryParam("adquirente") (opcional) para filtrar por adquirente. Caso não seja informado, o sistema trará todas as bandeiras.

### Saída

JSON contendo a lista de bandeiras

### Exemplo

Envio: [https://\[URL\]/ws/financeiro/bandeiras\\_cartao?adquirente=STONE](https://[URL]/ws/financeiro/bandeiras_cartao?adquirente=STONE)

### Retorno:

```
[
  {
    "id": 3,
    "descricao": "MASTERCARD",
    "idEmpresa": 1,
    "nomeEmpresa": "LAFAIETE",
    "codigoCreditoSap": "8",
    "codigoDebitoSap": "8",
    "adquirente": "STONE"
  }
]
```

OU "FALHA"

## Buscar todas formas de pagamento

### Descrição

Busca todas as formas de pagamento

### URL

[https://\[URL\]/ws/financeiro/formas\\_pagamento](https://[URL]/ws/financeiro/formas_pagamento)

### Método

GET

### Entrada

HeaderParam("Token") com o value contendo um token válido, gerado pelo método "Gerar Token".

### Saída

JSON contendo a lista de formas de pagamento

### Exemplo

Envio: [https://\[URL\]/ws/financeiro/formas\\_pagamento](https://[URL]/ws/financeiro/formas_pagamento)

### Retorno:

```
[
  {
    "id": 1,
    "descricao": "BOLETO",
    "isCartao": false,
    "isDinheiro": false,
    "isTransferencia": false,
    "debito": false
  },
  {
    "id": 2,
    "descricao": "CARTAO DEBITO",
    "isCartao": true,
    "isDinheiro": false,
    "isTransferencia": false,
    "debito": true
  },
  {
    "id": 3,
    "descricao": "CARTAO CREDITO",
    "isCartao": true,
    "isDinheiro": false,
    "isTransferencia": false,
    "debito": false
  },
  {
    "id": 4,
    "descricao": "DEPOSITO",
    "isCartao": false,
    "isDinheiro": false,
    "isTransferencia": false,
    "debito": false
  }
]
```

OU "FALHA"

## Buscar contas financeiras de acordo com a forma de pagamento

### Descrição

Busca as contas financeiras vinculadas a uma determinada forma de pagamento

### URL

[https://\[URL\]/ws/financeiro/contas\\_financeiras/forma\\_pagamento/{idFormaPagamento}](https://[URL]/ws/financeiro/contas_financeiras/forma_pagamento/{idFormaPagamento})

### Método

GET

### Entrada

HeaderParam("Token") com o value contendo um token válido, gerado pelo método "Gerar Token".  
ID da formad e cobrança na URL passado como PathParam("idFormaPagamento")

### Saída

JSON contendo a lista de contas financeiras

### Exemplo

Envio: [https://\[URL\]/ws/financeiro/contas\\_financeiras/forma\\_pagamento/1](https://[URL]/ws/financeiro/contas_financeiras/forma_pagamento/1)

### Retorno:

```
[
  {
    "ativo": true,
    "banco": "CAIXA ECONOMICA",
    "id": 31,
    "descricao": "CEF 280637"
  },
  {
    "ativo": true,
    "banco": "BANCO DO BRASIL",
    "id": 32,
    "descricao": "BANCO DO BRASIL - OPERA SERVIÇOS"
  },
  {
    "ativo": true,
    "banco": "BRADESCO",
    "id": 33,
    "descricao": "BRADESCO CITY10"
  }
]
```

OU "FALHA"

## Pagar fatura

### Descrição

Insere pagamento de forma manual na fatura

### URL

[https://\[URL\]/ws/financeiro/faturas/pagar](https://[URL]/ws/financeiro/faturas/pagar)

### Método

PUT

### Entrada

HeaderParam("Token") com o value contendo um token válido, gerado pelo método "Gerar Token".  
JSON no formato especificado contendo os dados de pagamento

### Campos da Entrada

Tag	Tipo	Observação
IDFatura*	Long	
IDContaFinanceira*	Long	
IDFormaPagamento	Long	
ValorPago*	Double	
ValorMulta*	Double	
ValorJuros*	Double	
ValorDesconto*	Double	
DataPagamento*	String	Formato: dd-MM-yyyy
PagamentoParcial*	String	"0" = pagamento total "1" = pagamento parcial
CodigoAutorizacaoCartao	String	
TIDCartao	String	
NSUCartao	String	
IDBandeiraCartao	Long	

\*Campos obrigatórios

### Saída

JSON contendo a resposta "SUCESSO" ou o erro encontrado.

### Exemplo

#### Envio:

```
{
  "IDFatura": 90,
  "IDFormaPagamento": 2,
  "IDContaFinanceira": 1,
  "DataPagamento": "04-08-2020",
  "ValorMulta": 1.0,
  "ValorJuros": 0.0,
  "ValorDesconto": 0.0,
  "ValorPago": 315.98,
  "IDBandeiraCartao": 1,
  "CodigoAutorizacaoCartao": "44444",
  "NSUCartao": "55555",
  "TIDCartao": "",
  "PagamentoParcial": "0"
}
```

#### Retorno:

"SUCESSO"

**OU**

```
{  
  "error": "A fatura informada já se encontra inteiramente quitada."  
}
```

**OU**

```
{  
  "error": "(CAMPO ESPECÍFICO) inválido."  
}
```

**OU**

```
{  
  "error": "INTERNAL_SERVER_ERROR"  
}
```

## Buscar dados simplificados de faturas por CPF

### Descrição

Retorna as Faturas por CPF

### URL

[https://\[URL\]/ws/financeiro/faturas/{cpfcnpj}/simplificado](https://[URL]/ws/financeiro/faturas/{cpfcnpj}/simplificado)

### Método

GET

### Entrada

PathParam contendo o CPF/CNPJ do cliente

### Saída

Map contendo as informações sobre as faturas em aberto do cliente.

### Exemplo

#### Envio:

[https://\[URL\]/ws/financeiro/faturas/11122211122/simplificado](https://[URL]/ws/financeiro/faturas/11122211122/simplificado)

#### Retorno:

```
{
  "Faturas": [
    {
      "CodigoCliente": 8768696,
      "IDFatura": 8768696,
      "Valor": 125.0,
      "VencimentoFormatado": "10/03/2020"
    }
  ],
  "Cliente": "CLIENTE TESTE",
  "Celular": 98999999999
}
```



## Buscar bancos

### Descrição

Busca todos os bancos cadastrados

### URL

[https://\[URL\]/ws/financeiro/bancos](https://[URL]/ws/financeiro/bancos)

### Método

GET

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

### Saída

JSON contendo lista dos bancos ou JSON contendo mensagem de erro

### Exemplo

#### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "codigo": "033",
      "nome": "BANCO SANTANDER",
      "id": 1
    },
    {
      "codigo": "001",
      "nome": "BANCO DO BRASIL",
      "id": 2
    },
    {
      "codigo": "104",
      "nome": "CAIXA ECONOMICA",
      "id": 3
    },
    {
      "codigo": "237",
      "nome": "BRADESCO",
      "id": 5
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar serviços disponíveis para a empresa e a cidade

### Descrição

Busca os serviços disponíveis para a empresa e cidade informados

### URL

[https://\[URL\]/ws/financeiro/servicos/cidade/{idCidade}/empresa/{idEmpresa}](https://[URL]/ws/financeiro/servicos/cidade/{idCidade}/empresa/{idEmpresa})

### Método

GET

### Entrada

Header: *Token* {{token}}, *Authorization Basic* {{authorization}}

PathParam: idCidade, idEmpresa

### Saída

JSON contendo lista de serviços disponíveis para aquela cidade e empresa ou JSON contendo mensagem de erro

### Exemplo

**Envio:** [https://\[URL\]/ws/financeiro/servicos/cidade/100/empresa/1](https://[URL]/ws/financeiro/servicos/cidade/100/empresa/1)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "permiteParcelar": true,
      "isTaxaInstalacao": false,
      "valor": 200.0,
      "numeroMaximoParcelas": 10,
      "id": 6,
      "descricao": "CL - FIBRA - ALTERAÇÃO DE ENDEREÇO"
    },
    {
      "permiteParcelar": false,
      "isTaxaInstalacao": false,
      "valor": 40.0,
      "numeroMaximoParcelas": 0,
      "id": 10,
      "descricao": "CL - FIBRA - ALTERAÇÃO FÍSICA"
    },
    {
      "permiteParcelar": false,
      "isTaxaInstalacao": false,
      "valor": 40.0,
      "numeroMaximoParcelas": 0,
      "id": 19,
      "descricao": "CL - FIBRA - VISITA TÉCNICA"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Cadastrar serviço adicional

### Descrição

Cadastra um serviço adicional para um contrato

### URL

[https://\[URL\]/ws/financeiro/servicos\\_adicionais/cadastrar](https://[URL]/ws/financeiro/servicos_adicionais/cadastrar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados do serviço adicional

### Saída

JSON contendo o ID do serviço adicional cadastrado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idUsuario*	body	Number	
idCliente*	body	Number	
idServico*	body	Number	
idContrato*	body	Number	
obs*	body	String	
qtdeParcelas*	body	Number	

**\* Parâmetro obrigatório**

### Exemplo entrada:

```
{
  "idCliente": 81593,
  "idUsuario": 285,
  "idServico": 133,
  "idContrato": 9999999,
  "obs": "A",
  "qtdeParcelas": 2
}
```

### Exemplo retorno:

#### Sucesso:

```
{
  "idFaturamentoServico": 646225
}
```

#### Falha:

```
{
  "error": "O idCliente informado não pertence a nenhum cliente cadastrado no sistema."
}
```

## Buscar formas de cobrança por cidade e tipo

### Descrição

Busca as formas de cobrança de acordo com a cidade e o tipo informado

### URL

[https://\[URL\]/ws/financeiro/formas\\_cobranca/cidade/{idCidade}/tipo/{tipo}](https://[URL]/ws/financeiro/formas_cobranca/cidade/{idCidade}/tipo/{tipo})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

PathParam: *idCidade e tipo (Valores: DIGITAL, CARNE, DEBITO, PREPAGO, BOLETO)*

### Saída

JSON contendo as formas de cobrança disponíveis

### Exemplo

**Envio:** [https://\[URL\]/ws/financeiro/formas\\_cobranca/cidade/100/tipo/BOLETO](https://[URL]/ws/financeiro/formas_cobranca/cidade/100/tipo/BOLETO)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "Descricao": "TESTE",
      "DebitoAutomatico": false,
      "PermitePacote": true,
      "Carne": false,
      "Digital": false,
      "IdEmpresa": 1,
      "Id": 1,
      "Isento": false,
      "Ativa": true,
      "PrePago": false
    },
    {
      "Descricao": "TESTE II",
      "DebitoAutomatico": false,
      "PermitePacote": true,
      "Carne": false,
      "Digital": false,
      "IdEmpresa": 1,
      "Id": 2,
      "Isento": false,
      "Ativa": true,
      "PrePago": false
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar formas de cobrança ativas

### Descrição

Busca as formas de cobrança ativas

### URL

[https://\[URL\]/ws/financeiro/formas\\_cobranca](https://[URL]/ws/financeiro/formas_cobranca)

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

### Saída

JSON contendo as formas de cobrança ativas

### Exemplo

Envio: [https://\[URL\]/ws/financeiro/formas\\_cobranca](https://[URL]/ws/financeiro/formas_cobranca)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "Descricao": "LAFAIETE BANCO SANTANDER - LT",
      "DebitoAutomatico": false,
      "PermitePacote": true,
      "Carne": false,
      "Digital": false,
      "IdEmpresa": 1,
      "Id": 1,
      "Isento": false,
      "Ativa": true,
      "PrePago": false
    },
    {
      "Descricao": "LAFAIETE DÉBITO SANTANDER",
      "DebitoAutomatico": true,
      "PermitePacote": true,
      "Carne": false,
      "Digital": false,
      "IdEmpresa": 1,
      "Id": 3,
      "Isento": false,
      "Ativa": true,
      "PrePago": false
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## REDES

### Buscar dados de navegação

#### Descrição

Retorna as Faturas por CPF

#### URL

[https://\[URL\]/ws/redes/graficos\\_navegacao?login={loginRadius}&dataInicio={dataInicio}&dataFim={dataFim}](https://[URL]/ws/redes/graficos_navegacao?login={loginRadius}&dataInicio={dataInicio}&dataFim={dataFim})

#### Método

POST

#### Entrada

Login Radius do cliente, data início do intervalo de busca e data fim do intervalo de busca no formato dd/MM/yyyy HH:mm como query param (todos obrigatórios) e JSON

**OBS:** O intervalo entre a data início e a data fim deve ser inferior a 40 dias.

#### Saída

Map contendo uma lista de contadores de tráfego de navegação ao longo do tempo.

#### Campos do JSON da entrada

Tag	Tipo	Observação
usuario *	String	
senha *	String	

\*Campos obrigatórios

#### Exemplo

##### Envio:

```
{
  "usuario": "12312312312",
  "senha": "12312312312"
}
```

##### Retorno:

```
[
  {
    "_id": {
      "year": 2018,
      "month": 1,
      "day": 9,
      "hour": 17,
      "minute": 8,
      "userName": "2adimarapnogueira"
    },
    "in": 5842016016,
    "out": 65092063096
  },
  {
    "_id": {
      "year": 2018,
      "month": 1,
      "day": 9,
```

```

    "hour": 17,
    "minute": 18,
    "userName": "2adimarapnogueira"
  },
  "in": 5848510368,
  "out": 65199855336
}
]

```

**OBS:** Os valores in e out são contadores de tráfego e precisam ser tratados para exibir corretamente o consumo de banda naquele momento de tempo, para isso é necessário fazer:

$$download_i = \frac{(out_i - out_{i-1})}{600}$$

$$upload_i = \frac{(\square_i - \square_{i-1})}{600}$$

Onde 600 é o intervalo de tempo em segundos entre cada leitura. O resultado é dado em bits/segundo.

## Buscar justificativas de alteração de caixa

### Descrição

Busca as justificativas de alteração de caixa

### URL

[https://\[URL\]/ws/redes/alterar\\_caixa/justificativas](https://[URL]/ws/redes/alterar_caixa/justificativas)

### Método

GET

### Entrada

Header: *Token* `{{token}}`, *Authorization Basic* `{{authorization}}`

### Saída

JSON contendo lista dos dados ou JSON contendo mensagem de erro

### Exemplo

**Envio:** [https://\[URL\]/ws/redes/alterar\\_caixa/justificativas](https://[URL]/ws/redes/alterar_caixa/justificativas)

### Retorno:

#### Sucesso:

```
[
  {
    "Descricao": "ERRO - HA OUTRO CLIENTE CONECTADO",
    "Tipo": "ERRO",
    "ID": "1"
  },
  {
    "Descricao": "SAIDA NAO CONECTA",
    "Tipo": "MANUTENCAO",
    "ID": "2"
  },
  {
    "Descricao": "EXISTE UMA CAIXA MAIS PRÓXIMA DO CLIENTE",
    "Tipo": "OUTROS",
    "ID": "3"
  }
]
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```



## Desprovisionar equipamento

### Descrição

Desprovisiona um equipamento

### URL

[https://\[URL\]/ws/redes/equipamento/desprovisionar](https://[URL]/ws/redes/equipamento/desprovisionar)

### Método

POST

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

Body: Json contendo os dados para desprovisionamento

### Saída

JSON contendo se foi desprovisionado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
IdContrato*	body	Long	
SerialEquipamento*	body	String	
IdPlano*	body	Long	
IdCliente*	body	Long	
IdCidade*	body	Long	Cidade de instalação
LoginPPoE*	body	String	
TipoEquipamento*	body	String	Valores: ZTE, NOKIA ,HUAWEI, FIBERHOME
isTelefonia*	body	Boolean	

**\* Parâmetro obrigatório**

### Exemplo entrada:

```
{
  "IdContrato": 1
  "SerialEquipamento": "554545"
  "IdPlano": 1
  "IdCliente": 1
  "IdCidade": 1
  "LoginPPoE": "teste"
  "TipoEquipamento": "ZTE"
  "isTelefonia": false
}
```

### Exemplo retorno:

#### Sucesso:

```
{
  "desprovisionou": true
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Liberar saída de caixa de contrato

### Descrição

Libera a saída da caixa e desvincula do contrato

### URL

[https://\[URL\]/ws/redes/liberar/caixa/saida](https://[URL]/ws/redes/liberar/caixa/saida)

### Método

POST

### Entrada

Header: *Token* `{{token}}`, *Authorization Basic* `{{authorization}}`

Body: Json contendo os dados do contrato

### Saída

JSON contendo o ID do contrato alterado ou JSON contendo mensagem de erro

### Formato da Entrada

Nome	Tipo parâmetro	Tipo dado	Observações
Token*	header	String	
Authorization*	header	String	
idContrato*	body	Long	

**\* Parâmetro obrigatório**

### Exemplo entrada:

```
{
  "idContrato": 285
}
```

### Exemplo retorno:

#### Sucesso:

```
{
  "idContratoAtualizado": 285
}
```

#### Em caso de atendimento não encontrado:

```
{
  "error": "Contrato nao encontrado"
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## TELEFONIA

### Buscar prefixos por cidade

#### Descrição

Busca os prefixos de acordo com a cidade e se será para cadastro de um número que será portado

#### URL

[https://\[URL\]/ws/telefonica/prefixos/cidade/{idCidade}/portabilidade/{isPortabilidade}](https://[URL]/ws/telefonica/prefixos/cidade/{idCidade}/portabilidade/{isPortabilidade})

#### Método

GET

#### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

PathParam: idCidade, isPortabilidade

#### Saída

JSON contendo lista de prefixos da cidade ou JSON contendo mensagem de erro

#### Exemplo

Envio: [https://\[URL\]/ws/telefonica/prefixos/cidade/100/portabilidade/0](https://[URL]/ws/telefonica/prefixos/cidade/100/portabilidade/0)

**OBS.: Parâmetro “isPortabilidade” deverá receber os valores 0 ou 1 (0 = prefixo próprio – não é portabilidade / 1 = prefixo de terceiros – é portabilidade)**

#### Retorno:

##### Sucesso:

```
{
  "resposta": [
    {
      "id": 1,
      "codigoPais": "55",
      "codigoArea": "31",
      "prefixo": "3939",
      "idCidade": 100,
      "nomeCidade": "CONSELHEIRO LAFAIETE",
      "isTerceiro": false
    }
  ]
}
```

##### Falha:

```
{
  "error": "Internal Server Error"
}
```

## Buscar números disponíveis para venda por prefixo

### Descrição

Busca os números disponíveis para venda de acordo com o prefixo

### URL

[https://\[URL\]/ws/telefonica/numeros/livres/prefixo/{idPrefixo}](https://[URL]/ws/telefonica/numeros/livres/prefixo/{idPrefixo})

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

PathParam: idPrefixo

### Saída

JSON contendo lista de números livres do prefixo ou JSON contendo mensagem de erro

### Exemplo

Envio: [https://\[URL\]/ws/telefonica/numeros/livres/prefixo/1](https://[URL]/ws/telefonica/numeros/livres/prefixo/1)

### Retorno:

#### Sucesso:

```
{
  "resposta": [
    {
      "id": 2742,
      "numero": "2741",
      "status": "LIVRE",
      "prefixo": "3939",
      "codigoArea": "31",
      "codigoPais": "55"
    },
    {
      "id": 2743,
      "numero": "2742",
      "status": "LIVRE",
      "prefixo": "3939",
      "codigoArea": "31",
      "codigoPais": "55"
    },
    {
      "id": 2746,
      "numero": "2745",
      "status": "LIVRE",
      "prefixo": "3939",
      "codigoArea": "31",
      "codigoPais": "55"
    }
  ]
}
```

#### Falha:

```
{
  "error": "Internal Server Error"
}
```

## INTEGRAÇÕES PASSIVAS (SVA)

### Gateway NEO - Autenticação

#### Descrição

Autentica usuário para acesso aos SVAs da NEO

#### URL

[https://\[URL\]/ws/comercial/clientes/integracao/authentication?username={usuario}&password={senha}](https://[URL]/ws/comercial/clientes/integracao/authentication?username={usuario}&password={senha})

#### Método

GET

#### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

QueryParam: *username, password*

**OBS.:** O usuário e senha informados deverão ser os mesmos para acessar a área do assinante

#### Saída

JSON contendo se o acesso foi permitido ou não

#### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/clientes/integracao/authentication?username=11122233344&password=11122233344](https://[URL]/ws/comercial/clientes/integracao/authentication?username=11122233344&password=11122233344)

#### Retorno:

##### Sucesso:

##### Acesso permitido:

```
{
  "access": true,
  "subscriber_id": 81593
}
```

##### Acesso negado ou falha:

```
{
  "access": false
}
```

## Gateway NEO - Autorização

### Descrição

Autoriza usuário para acesso aos SVAs da NEO

### URL

[https://\[URL\]/ws/comercial/clientes/integracao/authorization?](https://[URL]/ws/comercial/clientes/integracao/authorization?subscriber_id={subscriber_id}&resource_id={resource_id}&country_code=BR&action_id=VIEW&ip_address={ip})

[subscriber\\_id={subscriber\\_id}&resource\\_id={resource\\_id}&country\\_code=BR&action\\_id=VIEW&ip\\_address={ip}](https://[URL]/ws/comercial/clientes/integracao/authorization?subscriber_id={subscriber_id}&resource_id={resource_id}&country_code=BR&action_id=VIEW&ip_address={ip})

### Método

GET

### Entrada

Header: *Token {{token}}, Authorization Basic {{authorization}}*

QueryParam: subscriber\_id (obrigatório), resource\_id (obrigatório), country\_code, action\_id, ip\_address

### Saída

JSON contendo se foi autorizado ou não

### Exemplo

**Envio:** [https://\[URL\]/ws/comercial/clientes/integracao/authorization?](https://[URL]/ws/comercial/clientes/integracao/authorization?subscriber_id=81593&resource_id=123&country_code=BR&action_id=VIEW&ip_address=192.168.1.2)

[subscriber\\_id=81593&resource\\_id=123&country\\_code=BR&action\\_id=VIEW&ip\\_address=192.168.1.2](https://[URL]/ws/comercial/clientes/integracao/authorization?subscriber_id=81593&resource_id=123&country_code=BR&action_id=VIEW&ip_address=192.168.1.2)

### Retorno:

#### Sucesso:

#### Acesso permitido:

```
{
  "access": true
}
```

#### Acesso negado ou falha:

```
{
  "access": false
}
```

## BIT TRAINERS - Autenticação

### Descrição

Autentica usuário para acesso aos SVAs da Bit Trainers

### URL

[https://\[URL\]/ws/integracao/bittrainers/authentication?username={username}&password={password}](https://[URL]/ws/integracao/bittrainers/authentication?username={username}&password={password})

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

QueryParam: username, password

**OBS.:** O usuário e senha informados deverão ser os mesmos para acessar a área do assinante

### Saída

JSON contendo se o acesso foi permitido ou não

### Exemplo

**Envio:** [https://\[URL\]/ws/integracao/bittrainers/authentication?username=11122233344&password=12345678](https://[URL]/ws/integracao/bittrainers/authentication?username=11122233344&password=12345678)

### Retorno:

#### Sucesso:

##### Acesso permitido:

```
{
  "access": true,
  "subscriber_id": "kVGLCfGeGjWzbSzm7Sv46Q=="
}
```

##### Acesso negado ou falha:

```
{
  "access": false
}
```

## BIT TRAINERS - Autorização

### Descrição

Autoriza usuário para acesso aos SVAs da Bit Trainers

### URL

[https://\[URL\]/ws/integracao/bittrainers/authorization?subscriber\\_id=kVGLCfGeGjWzbSzm7Sv46Q==&resource\\_id=11](https://[URL]/ws/integracao/bittrainers/authorization?subscriber_id=kVGLCfGeGjWzbSzm7Sv46Q==&resource_id=11)

### Método

GET

### Entrada

Header: *Token {{token}}*, *Authorization Basic {{authorization}}*

QueryParam: *subscriber\_id*, *resource\_id*

### Saída

JSON contendo se foi autorizado ou não

### Exemplo

**Envio:** [https://\[URL\]/ws/integracao/bittrainers/authorization?subscriber\\_id=kVGLCfGeGjWzbSzm7Sv46Q==&resource\\_id=11](https://[URL]/ws/integracao/bittrainers/authorization?subscriber_id=kVGLCfGeGjWzbSzm7Sv46Q==&resource_id=11)

### Retorno:

#### Sucesso:

#### Acesso permitido:

```
{
  "access": true
}
```

#### Acesso negado ou falha:

```
{
  "access": false
}
```