



API **BRADESCO**

MANUAL E *LAYOUT*

**API DE REGISTRO DE
BOLETO COM QR CODE**
COBRANÇA BRADESCO

Versão 1 – novembro/22

SUMÁRIO

1.	OBJETIVO	3
2.	PRÉ-REQUISITOS	4
3.	CONECTIVIDADE	5
3.1.	CERTIFICADO PÚBLICO	5
3.2.	MENSAGENS DE POSSÍVEIS ERROS	6
3.3.	AMBIENTE DE HOMOLOGAÇÃO	6
3.4.	AMBIENTE DE PRODUÇÃO	8
3.5.	ENVIO DOS DADOS CADASTRAIS	8
3.6.	CONSUMO DAS APIS	8
3.7.	OBTER <i>ACCESS-TOKEN</i>	9
3.8.	<i>JWT</i>	9
3.9.	<i>HEADER</i>	9
3.10.	<i>PAYLOAD</i>	10
3.11.	GERAÇÃO <i>JWT</i>	10
3.12.	GERAÇÃO <i>JWS</i>	11
3.13.	REQUISIÇÃO <i>HTTP ACCESS-TOKEN</i>	13
3.14.	CONSUMO <i>ENDPOINT</i>	15
3.15.	ASSINATURA	15
3.16.	FORMATAÇÃO	155
3.17.	GERAÇÃO DE ASSINATURA	200
3.18.	REQUISIÇÃO <i>HTTP ASSINATURA</i>	211
3.19.	PRODUÇÃO	22
4.	REGISTO DE BOLETO COM <i>QR CODE</i> VIA API	244
5.	<i>LAYOUT</i> DE DESENVOLVIMENTO E <i>URLs</i>	266
6.	<i>LAYOUT</i> DE COMUNICAÇÃO - ENTRADA	311
7.	<i>LAYOUT</i> DE COMUNICAÇÃO - RETORNO	37
8.	MENSAGENS DE RETORNO	42
9.	TABELA DE CÓDIGO DE ESPÉCIE DE TÍTULOS	455
10.	FAQ	46
11.	SUORTE	51

1. OBJETIVO

Este manual apresentará o modelo de acesso às APIs - Interface de Programação de Aplicativos da Organização Bradesco, demonstrando a regra de negócio e orientando o passo a passo para automatizar o uso das APIs na integração da API de registro de título de Cobrança com *QR Code*.

Nesse modelo, a autorização de acesso considerará os recursos acessados pertencentes à aplicação servidora, e o *token* de acesso será emitido para a própria aplicação, e não para um usuário final.

O padrão de autorização adotado será o *JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants*.

2. PRÉ-REQUISITOS

Para promover a integração sistêmica de registro *on-line* de boletos com QR Code Bradesco, é necessário:

1. Possuir Contrato de Cobrança Bradesco ativo;
2. Definir a Chave Pix que será usada na geração dos boletos com QR Code;
3. O Gerente deve habilitar o uso da funcionalidade do indicador “175 - Registro via Webservice”, assim como o indicador “182 - Cobrança com QR Code”;
4. Efetuar o desenvolvimento de conexão de acordo com o manual do desenvolvedor.

3. CONECTIVIDADE

Para geração do ID da aplicação e iniciar a comunicação com o Bradesco, o Gerente deve enviar os dados da empresa (nome, CNPJ, dados para contato), formalizando o pedido de interesse na utilização por meio do *e-mail* suporte.api@bradesco.com.br.

O cadastro será por meio do certificado público digital fornecido, sendo um para cada ambiente, homologação e produção. Seguem as características de construção e formato esperado:

O acesso às APIs Bradesco, nos ambientes de homologação e produção, utiliza o protocolo TLS 1.2 com os algoritmos (cifras):

- ✓ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, e/ou
- ✓ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384;

Importante:

Informamos que desabilitaremos as versões 1.0 e 1.1 do TLS e os algoritmos (cifras) diferentes das apresentadas acima.

3.1. CERTIFICADO PÚBLICO

O envio deve ser feito de toda a cadeia de certificados - raiz, intermediários e domínio, sendo extensão “.PEM” (base64) e padrão X.509 no formato requisitado. Na composição dos dados, é importante a identificação da empresa, referenciando o CNPJ e Razão Social.

O arquivo a ser disponibilizado no formato de texto deve começar com “BEGIN CERTIFICATE”, conforme exemplo abaixo.

Quadro 01 - Exemplo Arquivo Certificado Público, Formato X.509

```
Exemplo: manual.teste.com.cert.pem
-----BEGIN CERTIFICATE-----
MIIDQzCCAiugAwIBAgIUcByYmH6Btz/Fk3p5QMvedrO4+fgwDQYJKoZIhvcNAQELBQAwMTELMakG A
1UEBhMCQlIx CzAJBgNVBAgMAIBSMRUwEwYDVQQKDAxUZXR0ZSByYW51YW...
-----END CERTIFICATE-----
```

A Chave Privada, relacionada ao certificado compartilhado, é de responsabilidade de sua empresa e, deve ser armazenada de forma segura, nunca sendo fornecida a terceiros.

3.2. MENSAGENS DE POSSÍVEIS ERROS

a) Mensagem de erro utilizando uma versão do TLS diferente da 1.2. Para exemplo foi utilizada a versão 1.0 do TLS.

Erro:

java.security.NoSuchAlgorithmException: TLSv1.0 SSLContext not available

b) Mensagem de erro utilizando algoritmos (cifras) diferentes do recomendado.

TLS_RSA_WITH_AES_256_CBC_SHA256 e

TLS_RSA_WITH_AES_256_CBC_SHA384

Erro:

Exception in thread "main" java.lang.IllegalArgumentException: Cannot support TLS_RSA_WITH_AES_256_CBC_SHA256 with currently installed providers

3.3. AMBIENTE DE HOMOLOGAÇÃO

Antes do início dos testes, deve ser enviado um certificado digital, sendo possível o uso do tipo autoassinado. Para tal, a geração pode ser por meio da biblioteca *openssl*.

Observação:

O *openssl* é distribuído de maneira nativa na maioria das distros *Linux*. Caso esteja fazendo uso de algum *OS* sem essa biblioteca, pode ser usado o “*Git Bash*”, ao qual é instalado com o “*Git Client*”, ou em caso de *OS Windows*, algum *WSL (Windows Subsystem for Linux)*.

Sendo assim, crie no diretório de execução o arquivo “*template.txt*” com o conteúdo ilustrado a seguir, substituindo os valores em negrito conforme necessidade. Por exemplo “<estado>”, tornando-se “São Paulo”. Nesse caso, a Razão Social e o CNPJ estarão no *Common Name (CN)* do certificado X.509.

Quadro 02 - Arquivo *template.txt* para Diretrizes do Certificado

```
[ req_distinguished_name ] stateOrProvinceName
                             = <estado>
organizationName             = <razão social>
commonName                   = <razão social> : <CNPJ>
countryName                   = BR

# =====

[ v3_req ]
keyUsage                     = critical,digitalSignature subjectKeyIdentifier
hash extendedKeyUsage
                             = clientAuth, serverAuth

[ req ]
default_bits                  = 2048
default_md                    = sha256
default_keyfile                = parceiro.homologacao.key.pem
utf8                          = yes
distinguished_name            = req_distinguished_name
req_extensions                 = v3_req prompt
                             = no
```

Como boas práticas, na geração do nome do arquivo, definido no campo “*default_keyfile*”, assim como no comando a seguir, use no lugar de “parceiro” o nome associado à sua organização. A expiração do certificado deve ser de, no mínimo, um ano.

Execute o seguinte comando para fazer a geração do par de arquivos:

Quadro 03 - Geração Certificado

Comando:
 openssl req -new -x509 -config template.txt -nodes -out parceiro.homologacao.cer.pem -days 1080

A execução gerará dois arquivos:

- ✓ **Chave Privada (*parceiro.homologacao.key.pem*);**
- ✓ **Certificado Público (*parceiro.homologacao.cer.pem*).**

Cada execução do comando com o mesmo nome do arquivo destino (*default_keyfile*) gera uma nova Chave Privada em um novo certificado. Dessa forma, uma execução no mesmo diretório sobrescreverá os arquivos já existentes.

3.4. AMBIENTE DE PRODUÇÃO

Visando agregar maior segurança ao processo, o uso de certificados emitidos por uma autoridade certificadora é recomendável desde que na estrutura contenha os dados da Razão Social e CNPJ, podendo ser utilizado um certificado autoassinado. O formato do arquivo modelo para produção é fornecido no item **20 - Produção**.

3.5. ENVIO DOS DADOS CADASTRAIS

O Certificado Público, e somente este, deve ser enviado por *e-mail* de forma compactada e protegido com senha. Em um segundo *e-mail*, deverá ser anexado o arquivo formato “.txt”, que contém a senha para descompactação. Além disso, informar no mesmo *e-mail* os dados abaixo:

- ✓ Razão Social e CNPJ;
- ✓ Breve descrição/uso da aplicação consumidora;
- ✓ Dois *e-mails* de contatos de referência para renovações de certificados e avisos.

Para o envio dos dados, o contato deve ser direcionado para o *e-mail* suporte.api@bradesco.com.br. Após a resposta com o fornecimento do ID de acesso, o uso do ambiente de homologação Bradesco já pode ser iniciado.

3.6. CONSUMO DAS APIS

Para consumir as Open APIs, são necessárias duas requisições:

- ✓ Obtenção do *access-token*;
- ✓ Consultar o *endpoint* do serviço.

Os passos seguintes descreverão como fazer as duas requisições de forma manual, usando as ferramentas de requisições HTTP, *Postman* e a biblioteca *openssl*. Os fluxos descritos devem ser desenvolvidos para funcionar de forma automatizada nos sistemas consumidores.

Os ambientes disponíveis e suas respectivas *URLs* são:

- ✓ **Homologação:** <https://proxy.api.prebanco.com.br>;
- ✓ **Produção:** <https://openapi.bradesco.com.br>.

3.7. OBTENHA O ACCESS-TOKEN

O *token* de acesso às APIs é concebido após o fornecimento de um *JWS* (após assinar digitalmente o *JWT*, ele torna-se um *JWS*), informando o ID da aplicação em um de seus *claims*, tal valor é gerado no passo anterior após o compartilhamento do certificado público.

No *response* da requisição estará presente o *token* a ser usado na autenticação do tipo *Bearer*.

3.8. JWT

O desenvolvedor deverá gerar um *JWT* de acesso e assinar o conteúdo com sua Chave Privada. O *JWT* é uma estrutura *JSON*, formada por duas partes, sendo *header* e *payload*. Ao fazer a assinatura, torna-se um *JWS*.

Para Boletão com QR Code é necessário que o Token seja gerado na URL:
<https://proxy.api.prebanco.com.br/auth/server/v1.2/token>

O Header e Payload permanecem iguais.

Para funcionar o access token em homologação, deve ser obtido na URL v1.2, mas os campos do Payload devem permanecer com 1.1, dessa forma:

Autenticação

clientID (campo "sub" do JWT)

URL geração access-token:

<https://proxy.api.prebanco.com.br/auth/server/v1.2/token>

3.9. HEADER

A estrutura do *header* será:

Estrutura:

```
{
  "alg": "<algoritmo utilizado>",
  "typ": "JWT"
}
```

Exemplo:

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

3.10. PAYLOAD

Para esse modelo de autenticação, o *endpoint* a ser consumido para geração do *access-token* é o `"/auth/server/v1.1/token"`. Assim, deve ser fornecida a *URL* completa no Campo *Audience*, acrônimo *"aud"*.

Quadro 05 - Estrutura e Exemplo de Preenchimento do *Payload*

Estrutura:
<pre>{ "aud" : "<URL do serviço de geração de token>", "sub" : "<ID aplicação / client key>", "iat" : "<data de geração atual, formato Unix timestamp (segundos)>", "exp" : "<data de expiração, formato Unix timestamp (segundos)>", "jti" : "<nonce - numérico de no máximo dezoito dígitos, valor sem repetição>", "ver" : <versão> }</pre>
Exemplo:
<pre>{ "aud" : "https://proxy.api.prebanco.com.br/auth/server/v1.1/token", "sub" : "bc7ccf09-8a85-4be6-y67e-82bf11737994", <id cliente fornecido pelo banco> "iat" : "1612899472", <data atual em segundos> "exp" : "1612903071", <data atual adicionando uma hora à frente, em segundos> "jti" : "1574094116000", <data atual em milissegundos> "ver" : "1.1" }</pre>

A data de expiração do *token* é controlada pelo claim *"exp"*. A expiração pode ser definida entre um segundo e uma hora, podendo ser escolhida conforme necessidade da aplicação consumidora.

3.11. GERAÇÃO JWT

Para realizar a geração do *JWT*, será necessário fazer o *encode* dos dois *JSONs* citados para *base64 url encoded*, além da remoção de todos os espaços e quebras de linha, operação conhecida como *"stringify"*.

Quadro 06 - *Encode* e formatação *JSON* para o *JWT*

Comando:
<pre>echo -n "\$(cat <arquivo>)" tr -d '[:space:]' base64 tr -d '[:space:]' tr '+/' '-_'</pre>

O comando `"echo -n "$(cat <arquivo>)"` realizará a leitura do arquivo. A Opção *"-n"* é para que seja feita a leitura e propagação da *string* sem quebra de linha no final. Por sua vez, o Comando `"tr -d '[:space:]'"` faz a operação *"stringify"*, removendo caracteres especiais (espaço, *line feed* – `'LF'` ou `'\n'` – e *carriage return* – `'CR'` ou `'\r'`) do arquivo lido. Dessa forma, realiza-se o *encode* para *base64*. O último comando, `"tr -d '[:space:]' | tr '+/' '-_'"`, formata para *URL encoded* e

facilita o *copy and paste*, fazendo com o que o *output* não tenha quebras de linha.

Quadro 07 - Exemplo Geração *Header* e *Payload* para Uso no *JWT*

Exemplo (header.json):

```
echo -n "$(cat header.json)" | tr -d '[:space:]' | base64 | tr -d '[:space:]' | tr '+/' '-_'
```

Exemplo (payload.json):

```
echo -n "$(cat payload.json)" | tr -d '[:space:]' | base64 | tr -d '=' | tr '+/' '-_'  
eyJhdWQiOiJodHRwczovL3Byb3h5LmFwaS5wcmViYW5jbYy5jY20uYnlvYXV0aC9zZXZlZXIvdjEuMS9-  
0btltbilslN1Yil6ImJnN2NjZjA5LTlhODUtNGJlNi15NjdILTgyYmYxMTczNzk5NCIsImhhbmhlcGl6eE1NzQwOTQ-  
xMXYtLCJleHAoOiwkNTc2Njg2MTE2liwiianRpljoimTU3NDANDEENjAwMCIsInZici6l6l6lEuMSJ9
```

Junte os dois resultados, separando-os com o caractere "." (ponto). O formato será:

Header (base64url encoded) + "." + Payload (base64url encoded)

Quadro 08 - Exemplo JWT

[eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJodHRwczovLjB3YyB3h5LmFwaS5wcmViYW5jbyB5jb20uYnVlYXV0aC9zZXZlZXIvdjEuMS90b2t1biIsInN1Yil6ImJnN2NjZjA5LTlhODUtNGJJNi15NjdldjEgYmYxMTczNzk5NCIsImh0dCI6IjE1NzQwOTQxMTYiLCJleHAiOiIxNTc2Njg2MTE2IiwianRpIjoiaMTU3NDAA5NDExNjAwMCIsInZlcnI6IjEuMSJ9](#)

3.12. GERAÇÃO JWS

O *JWS* (*JSON Web Signature*) é uma estrutura que contém o conteúdo do *JWT* adicionado da assinatura gerada após o seu *digest*. Para gerar a assinatura, use o seguinte comando:

Quadro 09 – Exemplo Geração Assinatura

Comando:

```
echo -n "$(cat <arquivo_assinar>)" | openssl dgst -sha256 -keyform pem -sign <chave_privada.pem> | base64 | tr -d '[:space:]' | tr '+/' '- '
```

```
Exemplo (arquivo "jwt.txt" e chave "manual.teste.com.key.pem"): echo -n "$(cat jwt.txt)" | openssl
dgst -sha256 -keyform pem -sign suporteAPI.teste.com.key.pem | base64 | tr -d
'=:space:]' | tr '+' '-'
```

HldrEd626l8GTPdulKteaopAYBuk_Yzp9oq9_GYko6ikKlgK3ezB-
y0yxfbzRNdUKH77c57t_re8j8EBHSyeauVRJdGfdJssU3dXH9rdFQ7yLTc7PH489oQ1x1CaC9HRJPOFL
6T
yq5pWfQyDBl4d9b777sriAx25oq8Lvr8pKszMkCDBccfFP0cZxFN5FkoBw8ynRe3FSUsVVzQTUzQ7
Kr1jL3dQctBPXHJ_84qfv9tUbX6k7RmYqyOAuqEC5JiqFTZzkjzGS8kbaZX0l4cDF7vKnbmhefiLIqc7P
ooWmw65o4zqpu6iME1hgOeYkwx5JlAO HdPR46S-lkH-Wg

Concatene a assinatura com a *string* usada, separando-a com caractere "." (ponto). O formato final será o *JWS* abaixo:

Header+ "." + Payload + "." + Assinatura (base64url encoded)

Quadro 10 - Exemplo JWS

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJodHRwczovL3Byb3h5LmFwaS5wcmViYW5jb2Y5LjB2OUYnIiwiaWF0eC9zZXZlcnVudjEuMS90b2tlibiIsInN1YiI6ImJnN2NjZjA5LTlhODUtNGJlNi15NjdllTgyYmYxMTczNzk5NCIsImhhdCI6IjE1NzQwOTQxMTYiLCJleHAiOiIxNTc2Njg2MTE2IiwianRpIjoimTU3NDA5NDExNjAwMCIsInZlcil6IjEuMSJ9.HldrEd626l8GTPdulKteaopAYBuk_Yzp9oq9_GYko6ikKIgk3ezBy0yxfbzRNdUK h77c57t_re8j8EBHSyeauVRJdGfdJssU3dXH9rdFQ7yLTc7PH489oQ1x1CaC9HJRJPofL6Tyq5pWfQyDBl4d9b777sriAx25oq8Lvr8pKszMkCDBccfFP0cZxFN5FKoBw8ynRe3FSUsVVzQTUzQ7Kr1jL3d

Q

ctBPXHJ_84qfv9tUbX6k7RmYqyOAuqEC5JiqFTZkjzGS8kbaZXO14cDF7vKnbmhefiLIqc7PooWmw65o4zqp
u6 iME1hqOEykwX5JIAO HdPR46S-lkh-Wg

Para uma geração mais rápida do *JWT* e, posteriormente, o *JWS*, tenha os 2 arquivos – *header* e *payload* – e execute a sequência abaixo.

Quadro 11 - Exemplo Geração Direta JWT

Comando:

```
stringToSign="$(echo -n "$(echo -n "$(cat header.json)" | tr -d '[:space:]' | base64)". "$(echo -n  

"$(cat payload.json)" | tr -d '[:space:]' | base64)" | tr -d '[:space:]' | tr '+/' - ' ')"
```

Não será retornado nenhum dado, o *JWT* foi gravado na variável *"stringToSign"*. O comando a seguir gerará a assinatura a partir do dado (*JWT*) gravado e, assim, retornar o *JWS*.

Quadro 12 - Exemplo Geração Direta JWS

Comando:

```
echo -n $stringToSign."$(echo -n $stringToSign | openssl dgst -sha256 -keyform pem -
sign <chave_privada.pem> | base64 | tr -d '[:space:]' | tr '+/' '- ')"
```

Exemplo (chave “manual.teste.com.key.pem”):

```
echo -n $stringToSign."$(echo -n $stringToSign | openssl dgst -sha256 -keyform pem sign
manual.teste.com.key.pem | base64 | tr -d '[:space:]' | tr '+/' '-_')"
eyJhbGciOiJIUzU1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJodHRwczovL29wZW5hcGkuYnJhZGV
yZ28uY29t
LmJyL2F1dGgvc2VydmVyL3YxLjEvdG9rZW4iLCJzdWlil0i0MWYxZWYwZC0yZDk4LTQxZjI
tOThjNC0xY
zQyMWI2YmM1ZmliLCJpYXQiOiIxNjEwOTI0NTU1IiwiaXhwaWoiMTYxMzUxNjU1NSIsImp0aSI6
IjE2MTA
5MjQ1NTUxNTMiLCJ2ZXliOiIxLjEifQ.4vA0Tz9bnuoVYjI4aTGLDdsSZhn6lI1erX1UsLchnSND_
L9nASwsH
HmJi-a7oixePQnY27dQayxN2VT-
6uVEfML8Rj7vS1wh2wa2y2CFDN79Dr2iSriBlimXlrPQ8gOjcSk48kpdYqynZuOMuZjvppMu
6kUJ8uz4Pmx0B5erCN4KNV19mSuNLXhDGbbXPhrDYOV7GRBlrEaRHFD0
RT5uxQDFL28C8ENUCoC17eSOmoBzPnrYle9wal28VxHg4g2GOLvbtIxfigmWhn5clMiBC8YStCpb0iA
UdYGucIXH1KfF94lpXloZTekS78zkgAmPvoPSutovHcsWUlr9-4w
```

3.13. REQUISIÇÃO *HTTP ACCESS-TOKEN*

Na descrição abaixo, foi usado o *Postman*, o qual deverá conter as informações de “método” e “headers” para obtenção do *token*.

Quadro 13 - Requisição Geração *Access-Token*

Method POST
URL: https://<endereço_do_ambiente>/auth/server/v1.1/token
Body: selecionar “x-www-form-urlencoded”
Key: grant_type = urn:ietf:params:oauth:grant-type:jwt-bearer Key: assertion = <JWS gerado>

O *Postman* adicionará o header “*Content-Type*” automaticamente, “*application/x-www-form-urlencoded*”.

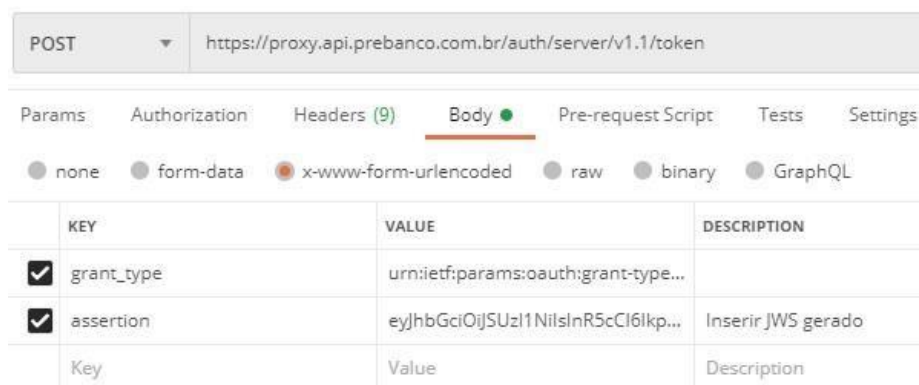


Figura 01 - Exemplo Requisição *Access-Token*, Ambiente de Homologação.

Após enviar a requisição ao serviço, o *access-token* gerado será retornado da seguinte forma:

Quadro 14 - Retorno Geração Access-Token

[illegible]

O Campo "*expires_in*" traz como valor por quantos segundos o *token* é válido. Um *token*, mesmo com duração maior que 3600 como retorno no campo, será impedido no consumo do serviço.

A geração de um novo *access-token* somente deve ser feita após a expiração, conforme valor recebido no retorno no atributo *"expires_in"*. O efeito colateral de gerações contínuas é se deparar com o erro abaixo.

Quadro 15 - Retorno Geração Contínua Access-Token

```
{
  "code": 0,
  "message": "unexpected error",
  "details": [
    {
      "name": "internalCode",
      "value": "FRWK0103"
    },
    {
      "name": "internalMessage",
      "value": "EXCEDIDO O LIMITE DE SESSÕES ABERTAS SIMULTANEAS PARA ESTE USUÁRIO. PARA ABRIR UMA NOVA SESSÃO, FECHER A SESSÃO QUE ESTÁ ATIVA NO TERMINAL APIFRONT OU UMA SESSÃO ATIVA EM OUTRO TERMINAL."
    }
  ]
}
```

3.14. CONSUMO ENDPOINT

Nesse manual, usaremos o *endpoint* de exemplo *"/v1.1/jwt-service"*. A resposta do recurso é "API acessada com sucesso!" e pode ser usada para o teste de autenticação na camada de segurança.

3.15. ASSINATURA

Um dos passos é gerar uma *string* a ser assinada pela chave privada. Nesse manual, será criado um arquivo com o nome *request.txt*, porém em uma aplicação é necessário somente construir a *string* no formato correto, não sendo necessária a criação de um arquivo.

3.16. FORMATAÇÃO

Nessa *string*, deverão ser incluídas algumas informações que serão usadas na chamada do HTTP, tais como: método, URI, parâmetros, *endpoint* e o *access-token* obtido anteriormente. As informações no arquivo devem ser inseridas assumindo que cada informação nova assumirá uma nova linha. O

arquivo *request.txt* deverá estar no mesmo diretório que o certificado público e a chave privada. O formato está descrito abaixo.

Quadro 16 – Formato Geração Assinatura Consumo *Endpoint*

VERBO[1] – linha 1
<URI da chamada>[2] – linha 2
<Parâmetros que estão sendo utilizados na URL>[3] – linha 3 <body>[4] – linha 4...
<Valor do access-token>[5]
<Nonce>[6]
<Timestamp>[7]
<Algoritmo que está sendo utilizado>[8]

É possível que os parâmetros e *token* ocupem várias linhas do arquivo, assim uma nova informação deverá ser colocada logo abaixo da outra. A ordem de informações no arquivo deve permanecer exatamente conforme o modelo acima. O arquivo “*request.txt*” após ser preenchido e estar devidamente assinado, será usado na chamada ao *endpoint* de desejo.

Segue, abaixo, um exemplo de preenchimento do arquivo que será assinado e como ficará a chamada no *Postman*.

- ✓ **Linha 1:** método utilizado no *Postman*.

POST^[1] - linha 1 do arquivo

Colocaremos esse mesmo verbo na chamada *Postman*:



Figura 02 – Inserção Método *HTTP*.

- ✓ **Linha 2:** URI da chamada (*endpoint* à API consultada)

/v1.1/jwt-service^[2] - linha 2 do arquivo

E no *Postman*:

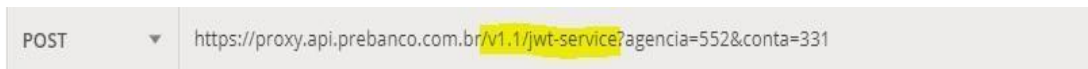


Figura 03 – Inserção *Endpoint*.

- ✓ **Linha 3:** exemplo de parâmetros da chamada
agencia=552&conta=331^[3] - linha 3 do arquivo

Então, com os parâmetros que estão sendo usados no *Postman*, podemos inserir esses valores na URI após o “?” ou na aba “*params*” do *Postman*.

Parâmetros:



Figura 04 - Inserção *query string*.

Aba *params*:

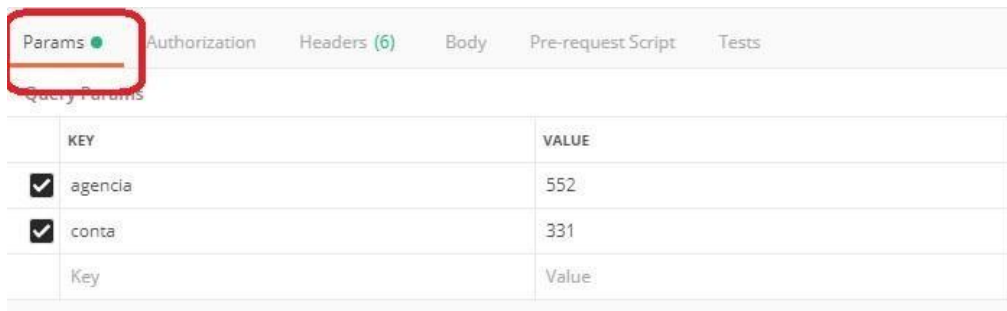


Figura 05 - Inserção *query string*.

✓ **Linha 4:** *body* da chamada

`{"teste":"valor"}[4]`

O *body* dessa chamada deve ser “raw”, “JSON” no *Postman*.

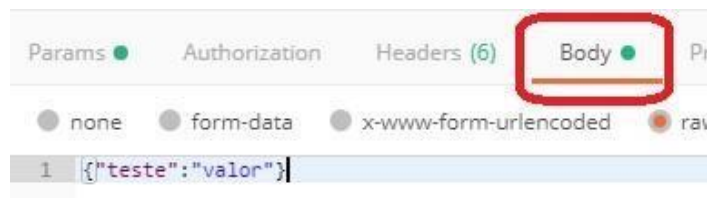


Figura 06 - Inserção *JSON body*.

No caso de um *endpoint* específico não possuir “parâmetros” ou “*body*” na requisição, a linha do arquivo *request.txt* correspondente deverá ficar em branco.

✓ **Linha 5:** *access-token* gerado nos passos anteriores.

`eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.ew0KICJ...[5]`

No *Postman*, devemos inserir esse valor no cabeçalho da chamada, que será referente à Chave “*Authorization*”, e no Campo Valor deve iniciar com “*Bearer*[espaço] valor do *access-token*” da seguinte forma:



Figura 07 - Inserção *access-token*.

Lembramos que deve ser considerado no Header da requisição o CNPJ do cliente, devendo ser preenchido para homologação conforme abaixo:

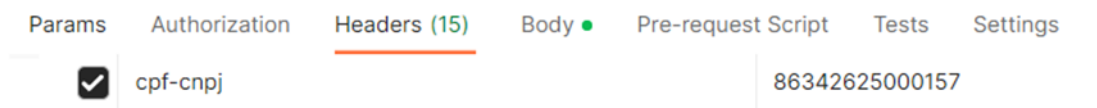


Figura 08 – Inserção cpf-cnpj

- ✓ **Linha 6:** *nonce* (valor numérico (18) aleatório que poderá ser usado uma única vez para cada chamada. Nesse caso, está sendo utilizada a data atual em milissegundos).

1574693951000^[6]

Esse mesmo valor deve ser inserido no *Postman*, equivalente à Chave “XBrad-Nonce”, no “*header*” da chamada, ficando conforme a imagem abaixo.

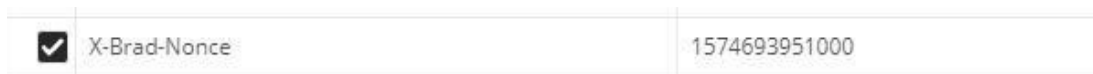


Figura 09 - Inserção *nonce*.

- ✓ **Linha 7:** *timestamp* (refere-se à data e hora que está sendo efetuada a chamada para o *endpoint*).

2019-11-25T11:23:00-03:00^[7]

Formato “AAAA-MM-DDThh:mm:ss-00:00”, sendo:

- AAAA = ano com quatro caracteres, exemplo “2019”, referindo-se ao ano atual;
- MM = mês com dois caracteres, exemplo “11”, referindo-se ao mês de novembro;
- DD = dia com dois caracteres, exemplo “25”, referindo-se ao dia 25;
- **T = texto fixo;**

- hh = hora com dois caracteres, exemplo “11”, referindo-se às 11 da manhã;
- mm = minutos com dois caracteres, exemplo “23”, referindo-se aos 23 minutos daquela hora;
- ss = segundos com dois caracteres, exemplo “00”;
- -03:00 = diferença para o fuso horário UTC 0:00. UTC -3:00 (fuso horário Brasília).

Esse mesmo valor deve ser inserido no cabeçalho da chamada, sendo a sua Chave “X-BradTimestamp”, ficando no padrão ilustrado abaixo:

<input checked="" type="checkbox"/>	X-Brad-Timestamp	2020-03-10T12:17:06-00:00
-------------------------------------	------------------	---------------------------

Figura 10 - Inserção *timestamp*.

- ✓ **Linha 8:** algoritmo que está sendo usado.

SHA256^[8]

No cabeçalho da chamada, o valor é correspondente à Chave: “X-BradAlgorithm”, conforme na imagem abaixo:

<input checked="" type="checkbox"/>	X-Brad-Algorithm	SHA256
-------------------------------------	------------------	--------

Figura 11 - Inserção *algorithm*.

Então, o arquivo para assinatura ficará da seguinte forma:

Quadro 17 - String geração assinatura consumo *endpoint*.

```
POST
/v1.1/jwt-service agencia=552&conta=331
{"teste":"valor"} eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.ew0KICJ...
1574693951000
2019-11-25T11:23:00-00:00
SHA256
```

No exemplo acima, estamos preenchendo tanto *body* quanto os parâmetros da requisição, porém, caso não seja usado *body* ou parâmetros, deve-se deixar a linha referente ao que não está sendo usado em branco.

Abaixo, ilustramos como deve ser feito um arquivo de assinatura quando não está sendo passado o *body* na chamada. Note que, nesse caso, deixamos a linha referente ao *body* em branco.

Quadro 18 - String geração assinatura consumo *endpoint*, sem *body*.

```
GET
/v1.1/jwt-service
agencia=552&conta=331

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.ew0KICJ...
1574693951000
2019-11-25T11:23:00-00:00 SHA256
```

É importante considerar a quebra de linha do padrão *Unix* (\n: LF – *LineFeed*) no arquivo *request*. O padrão de quebra de linha do *Windows* (\r\n CR – *CarriageReturn*, LF- *LineFeed*) ou de algum outro sistema (\r CR – *CarriageReturn*) causarão uma assinatura inválida.

Ou seja, o arquivo deve ser salvo no padrão de quebra linha do *Linux*. Caso seja salvo em outro padrão de quebra linha, será retornado na chamada um erro de assinatura inválida.

3.17. GERAÇÃO DE ASSINATURA

Então, agora que o arquivo de assinatura está completo, será necessário assiná-lo. Para exemplo, chamaremos esse arquivo de “*request.txt*”. Para assinar esse arquivo, é necessário salvá-lo no mesmo diretório da chave privada. Execute o comando em um ambiente *Linux* para gerar assinatura.

A assinatura deve estar no padrão do algoritmo usado, nesse caso, SHA256. O comando será o seguinte:

Quadro 19 - Exemplo geração de assinatura - *digest*.

```
Comando: echo -n "$(cat Arquivo_de_assinatura.txt)" | openssl dgst -sha256 -keyform pem -sign
```

```
<chave_privada.pem> | base64 | tr -d '[:space:]' | tr '+/' '-_'
```

Exemplo (arquivo “request.txt” e chave “manual.teste.com.key.pem”): \$ echo -n "\$(cat request.txt)" | openssl dgst -sha256 -keyform pem -sign manual.teste.com.key.pem | base64 | tr -d '[:space:]' | tr '+/' '-_'

```
hAj1J6HFztwqpJaTnt4YQcUlrRuUBb-
Uhr0N2J3TBgdxGo_UDGM3jUk4Ql6KmosNHKULfDc4SxcClpeAuYKocdD758EHwaBNuSW rddjq3hv2y
gFbqcVocT_wf6shsJiGqcT0QnuE12QZXszalP60oloS6H4MvYXnhNJmTGZ5q6gcFnXyl6zpY
G0Cci5Sme mmWMT8f0e04-wrDD-
XA02CrstbAovSKFc75JlRyLWcWQ6IMKMhfzmuD0Te59Wjg70mTfswJjXUehs9DNJNZS
_9TFX5gPWHp5pzMx_qv5jgjugtvYL31TECK5bNMWfTF8P2hnxFaAnEQMwGx8RgVA
```

A *string* gerada será usada para fazer a chamada, sendo o valor para o *header* “X-BradSignature” na requisição.

<input checked="" type="checkbox"/> X-Brad-Signature	U1QKL2p3dC1zZXJ2aWNlCmFnZW5jaWE9NTUyJmNvb...
--	--

Figura 12 - Inserção *signature*.

Antes de colar a *string* no campo do valor do “X-Brad-Signature”, é necessário deixá-lo em uma única linha; caso contrário, sua chamada retornará erro de assinatura inválida.

3.18. REQUISIÇÃO HTTP ASSINATURA

Quadro 20 - Exemplo *cURL* gerado pelo *Postman*.

```
curl --location --request POST 'https://proxy.api.prebanco.com.br/v1.1/jwt-service' \
--header 'Authorization: Bearer <access-token>' \
--header 'X-Brad-Signature:
BPao15xM4pjA6OVOo4o0muCgb22AV1iMwZiGj4Oyzxsx5brOFzTCgFNqiVR4p3arSCU4Z
0qqAlNP1fV mrNkmfHJeCaoB3Xxbw3qlz885M5Cc3s46vJQR4vvaNZ-uS-
NXP7Vj986t_oKNORCJaPrh7fsHGuN91RFnNIJ1WrFHFfdT8eMKR_qjT7_iroB4LWiB1f7smq7
D30OFCB398HGCBYdLbaEpK1-
mc_yxBPc7wXW0X6WtvAppZpDUkmmuC1d7sMXil1hJkAVdUHafqHSZvxopmBXTz1kyLT HF9vt23x4C
KYE_ykV9thmsZnsjO431mRQLL_k_q8-8Wba_eWGg'\
--header 'X-Brad-Nonce: 1610929465406' \
--header 'X-Brad-Timestamp: 2021-01-17T21:24:25-00:00' \ --header 'X-Brad-Algorithm: SHA256' \
```

Tabela 01 - Explicação cada *header* utilizado.

Chave	Valor
Authorization	Bearer access-token
X-Brad-Signature	Valor da assinatura em base 64, em linha

	única
<i>X-Brad-Nonce</i>	Valor do <i>nonce</i> que foi inserido no arquivo de assinatura
<i>X-Brad-Timestamp</i>	<i>Timestamp</i> inserido no arquivo de assinatura
<i>X-Brad-Algorithm</i>	Algoritmo utilizado

Após preenchidos os valores, a chamada à API “/v1.1/jwt-service” retornará a resposta “API acessada com sucesso.”.

Quadro 21 - Response.

```
HTTP/1.1 200 OK
Content-Type: text/plain;charset=UTF-8 Content-Length: 42
```

3.19. PRODUÇÃO

Após a conclusão dos testes em ambiente de homologação, é necessário seguir os passos abaixo, para que seja disponibilizado o acesso no ambiente.

- ✓ URL de Produção: <https://openapi.bradesco.com.br>

Em outro diretório de execução, crie o arquivo “*template.txt*” com o seguinte conteúdo e substitua os valores em negrito conforme necessidade (por exemplo: “<estado>”, tornando-se “São Paulo”).

Quadro 22 - Arquivo *template.txt* para diretrizes do certificado.

```
[ req_distinguished_name ] stateOrProvinceName
    = <estado> organizationName
    = <razão social>
commonName          = <razão social> : <CNPJ>
countryName          = BR

# =====

[ v3_req ]
keyUsage              = critical,digitalSignature subjectKeyIdentifier
                     = hash extendedKeyUsage              = clientAuth, serverAuth

[ req ]
default_bits = 2048 default_md = sha256
default_keyfile          = parceiro.producao.key.pem utf8
                        = yes
distinguished_name = req_distinguished_name req_extensions
```

```
= v3_req
prompt                = no
```

Como boas práticas na geração do nome do arquivo, definido no Campo “*default_keyfile*”, assim como no comando a seguir, use no lugar de “parceiro” o nome associado da sua organização. A expiração do certificado deve ser de, no mínimo, um ano.

Execute o seguinte comando para gerar o par de arquivos:

Quadro 23 – Geração de certificado.

```
Comando: openssl req -new -x509 -config template.txt -nodes -out
parceiro.producao.cer.pem -days 1080
```

O certificado público, e somente este, deve ser enviado, por *e-mail*, de forma compactada e protegido com senha. Em um segundo *e-mail*, deverá ser anexado um arquivo formato “.txt” com a senha para descompactação. Além disso, informar no mesmo *e-mail* os dados abaixo:

- ✓ Razão Social e CNPJ;
- ✓ Breve descrição/uso da aplicação consumidora;
- ✓ Evidência (*LOG HTTP status code 2xx*) de uma requisição à API;
- ✓ Dois *e-mails* de contatos de referência para renovações de certificados e avisos.

Para o envio dos dados, o contato deve ser direcionado para o *e-mail*: suporte.api@bradesco.com.br.

O processo de cadastro dos certificados em produção é para atender à demanda em três dias úteis. Ou seja, submissões feitas até as 18h de dias úteis serão implantadas em três dias úteis após a solicitação. A figura 12 ilustra exemplos da dinâmica.

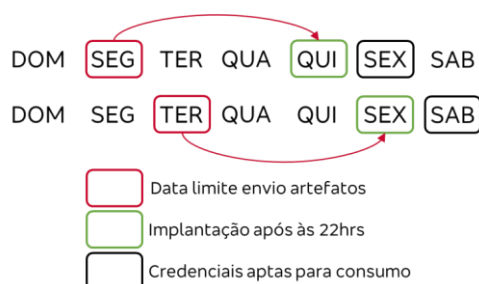


Figura 12 – Exemplos de credenciamento em produção.

4. REGISTO DE BOLETO COM QR CODE VIA API

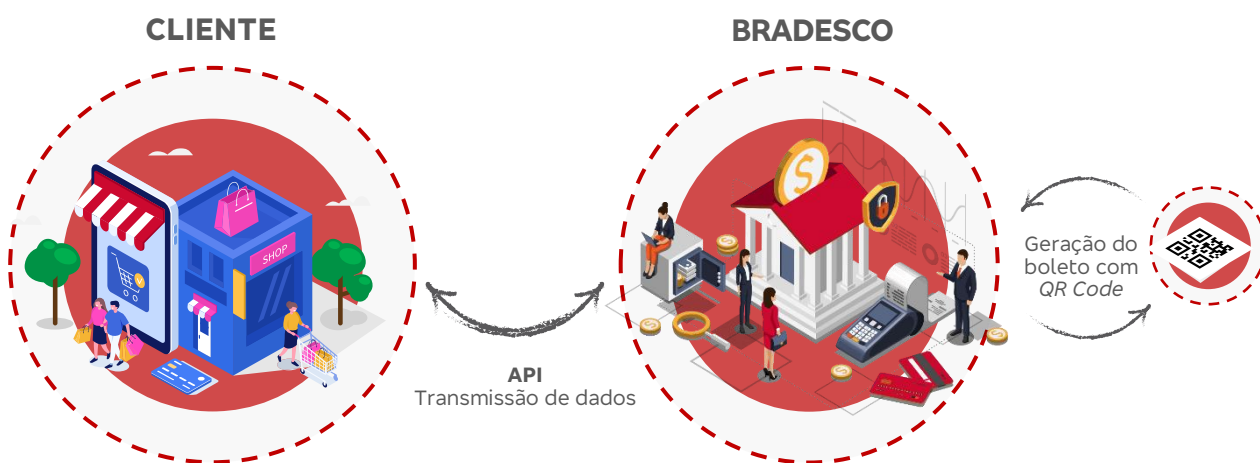
Para uma experiência completa, disponibilizamos a API de registro de boleto com *QR Code*. Com ela, você não precisará se preocupar em associar seus boletos com o *QR Code*, o Bradesco faz para você com segurança, praticidade e rapidez.

Para utilizar os benefícios disponíveis na API, primeiramente é necessário desenvolver a interface em sua plataforma de acordo com o *layout* de desenvolvimento disponibilizado pelo Bradesco (pág. 05) e manual do desenvolvedor, assim sendo possível a conectividade de transmissão de dados do Sistema Bradesco e a utilização do serviço de registro de boleto com *QR Code*.

Após concluir a conectividade, basta enviar a requisição com os dados do boleto de cobrança (ex.: dados do pagador, valor, desconto, juros etc.), a plataforma Bradesco consistirá as informações recepcionadas, e estando com os dados preenchidos de acordo com a formatação válida, será registrado na base Bradesco e CIP. Consulte os campos obrigatórios disponibilizados no **Layout de Comunicação - Entrada** (pág. 10).

A plataforma Bradesco enviará o retorno de registro com sucesso, de forma *on-line*, com os dados do boleto bancário e do *QR Code* gerado para pagamento, a qualquer momento, em todas instituições financeiras.

Para consultar e/ou realizar alterações nos boletos registrados, deve ser realizada via Net Empresa (em desenvolvimento*) ou por arquivo-remessa *layout* 240 e 400 posições.



Nota: esse produto está em fase de normatização pelo BACEN, podendo ocasionar ajustes sistêmicos e regras futuras.

* Para mais informações e disponibilidade do serviço no Canal Net Empresa, consulte o seu Gerente.

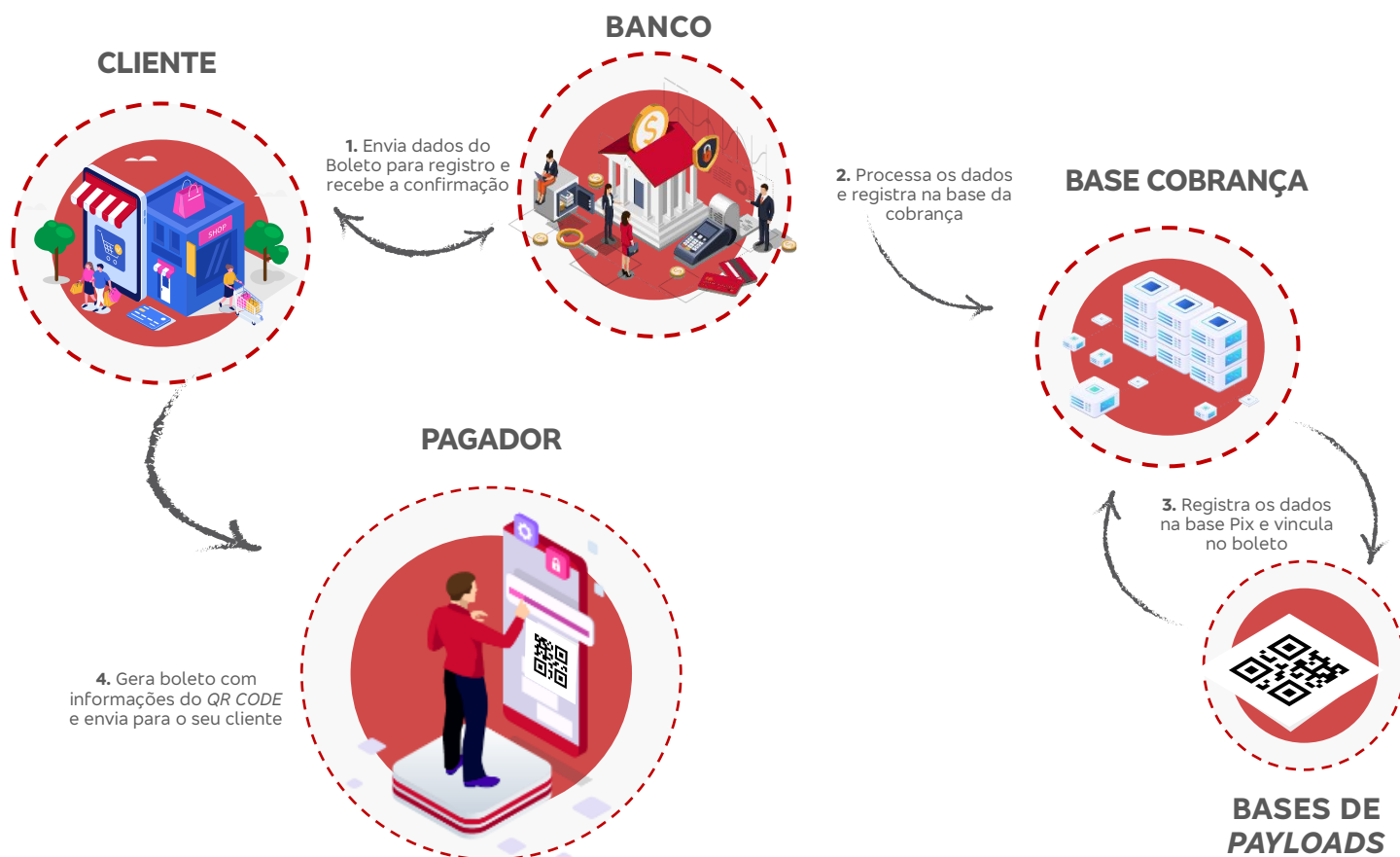
Retorno com Erro

Caso os dados recepcionados pela plataforma Bradesco estejam incorretos ou com formatação indevida, o processo será interrompido e você será notificado no mesmo instante com o código de erro (pág. 21).

Será necessário efetuar a correção dos dados correspondentes ao erro e retransmitir a requisição do registro.

Lembre-se que o desenvolvimento da solução para registro *on-line* via API não contempla a apresentação (imagem) do boleto com *QR Code*, para isso, deverá ser desenvolvida uma solução apartada.

Veja como é simples o fluxo para emissão:



5. LAYOUT DE DESENVOLVIMENTO E URLs

Os ambientes disponíveis e suas respectivas *URLs* são:

- ❖ Homologação: <https://proxy.api.prebanco.com.br/v1/boleto/registrarBoleto>
- ❖ Produção: <https://openapi.bradesco.com.br/v1/boleto/registrarBoleto>

O exemplo abaixo é apenas como referência para formatação dos campos (**JSON**):

Entrada

```
{
  "registrarTitulo": 0,
  "codUsuario": "String",
  "nroCpfCnpjBenef": 0,
  "filCpfCnpjBenef": 0,
  "digCpfCnpjBenef": 0,
  "tipoAcesso": 0,
  "cpssoaJuridContr": 0,
  "ctpoContrNegoc": 0,
  "nseqContrNegoc": 0,
  "cidtfdProdCobr": 0,
  "cnegocCobr": 0,
  "filler": "",
  "codigoBanco": 0,
  "eNseqContrNegoc": 0,
  "tipoRegistro": 0,
  "cprodtServcOper": 0,
  "ctitloCobrCdent": 0,
  "ctitloCliCdent": 0,
  "demisTitloCobr": 0,
  "dvctoTitloCobr": 0,
  "cidtfdTpoVcto": 0,
  "cindcdEconmMoeda": 0,
  "vnmnalTitloCobr": 0,
  "qmoedaNegocTitlo": 0,
  "cespceTitloCobr": 0,
  "cindcdAceitSacdo": 0,
  "ctpoProteTitlo": 0,
  "ctpoPrzProte": 0,
  "ctpoProteDecurs": 0,
  "ctpoPrzDecurs": 0,
  "cctrlPartcTitlo": 0,
  "cformaEmisPplta": 0,
  "cindcdPgtoParcial": 0,
  "filler1": "",
  "ptxJuroVcto": 0,
  "vdiaJuroMora": 0,
  "pmultaAplicVcto": 0,
  "vmultaAtrsoPgto": 0,
  "qdialnicMultas": 0,
  "pdescBonifPgto01": 0,
  "vdescBonifPgto01": 0,
  "dlimDescBonif1": 0,
```

```
"pdescBonifPgto02": 0,
"vdescBonifPgto02": 0,
"dlimDescBonif2": 0,
"pdescBonifPgto03": 0,
"vdescBonifPgto03": 0,
"dlimDescBonif3": 0,
"ctpoPrzCobr": 0,
"pdescBonifPgto": 0,
"vdescBonifPgto": 0,
"dlimBonifPgto": 0,
"vabtmTtloCobr": 0,
"viofPgtoTtlo": 0,
"filler2": "",
"elogdrSacdoTtlo": "string",
"enroLogdrSacdo": "string",
"ecomplLogdrSacdo": "string",
"ccepSacdoTtlo": 0,
"ccomplCepSacdo": 0,
"ebairoLogdrSacdo": "String",
"csglUfSacdo": "String",
"nroCpfCnpjSacdo": 0,
"renderEletrSacdo": "String",
"cdddFoneSacdo": 0,
"cfoneSacdoTtlo": 0,
"agenciaDeb": 0,
"agenciaDebDv": 0,
"contaDeb": 0,
"agenciaDvCentPr": 0,
"elogdrSacdrAvals": "String",
"enroLogdrSacdr": "String",
"ecomplLogdrSacdr": "String",
"ccepSacdrTtlo": 0,
"ccomplCepSacdr": 0,
"ebairoLogdrSacdr": "String",
"csglUfSacdr": "String",
"nroCpfCnpjSacdr": 0,
"renderEletrSacdr": "String",
"cdddFoneSacdr": 0,
"filler3": "",
"Fase": 0,
"cindcdCobrMisto": "String",
"caliasAdsaoCta": "0",
"wqrcdPdraoMerced": "String",
"filler4": "",
"qtdePgtoParcial": 0,
"qdiaInicJuro": 0,
"isacdoTtloCobr": "String",
"imunSacdoTtlo": "String",
"indCpfCnpjSacdo": 0,
"bancoDeb": 0,
"bancoCentProt": 0,
"isacdrAvalsTtlo": "String",
"imunSacdrAvals": "String",
"indCpfCnpjSacdr": 0,
"cfoneSacdrTtlo": 0,
"ialiasAdsaoCta": "String",
```

```
"iconcPgtoSpi": "String",  
"ilinkGeracQrCd": "String"  
}
```

Exemplo de Resposta (JSON)

```
{  
  "cidtFdProdCobr": 0,  
  "cnegocCobr": 0,  
  "cpssoaJuridContr": 0,  
  "ctpoContrNegoc": 0,  
  "nseqContrNegoc": 0,  
  "cprodtservcOper": 0,  
  "ctitloCobrCdent": 0,  
  "tp08Reg1": 0,  
  "agencCred10": 0,  
  "ctaCred10": 0,  
  "digCred10": 0,  
  "cip10": 0,  
  "codStatus10": 0,  
  "status10": "String",  
  "cedente10": "String",  
  "endCedente10": "String",  
  "nroEndCed10": "String",  
  "comEndCed10": "String",  
  "baiCedente10": "String",  
  "cepEndCed10": 0,  
  "cepcCedente10": 0,  
  "cidCedente10": "String",  
  "ufCedente10": "String",  
  "razCred10": 0,  
  "nomeSacado10": "String",  
  "cnpjSacado10": 0,  
  "endSacado10": "String",  
  "baiSacado10": "String",  
  "cidSacado10": "String",  
  "ufSacado10": "String",  
  "cepSacado10": 0,  
  "cepcSacado10": "String",  
  "cebp10": "String",  
  "debitoAuto10": "String",  
  "aceite10": "String",  
  "enderecoEma10": "String",  
  "nomeSacador10": "String",  
  "cnpjSacador10": 0,  
  "endSacador10": "String",  
  "cidSacador10": "String",  
  "ufSacador10": "String",  
  "cepSacador10": 0,  
  "cepcSacador10": 0,  
  "sFiller6": "",  
  "tp08Reg2": 0,  
  "cense10": 0,  
  "agenOper10": 0,  
  "bcoDepos10": 0,  
  "agenDepos10": 0,  
}
```

```
"snumero10": "String",
"dataReg10": "String",
"especDocto10": "String",
"descrEspec10": "",
"valorIof10": "String",
"dataEmis10": "String",
"especMoeda10": "String",
"qtdeMoeda10": 0,
"qtdeCas10": 0,
"dataVencto10": "String",
"descrMoeda10": "String",
"valMoeda10": 0,
"valAbat10": 0,
"dataInstr10": "String",
"diasProt10": 0,
"dataCartor10": "String",
"numCartor10": "String",
"numProtoc10": "String",
"dataPedSus10": "String",
"dataSust10": "String",
"dataMulta10": "String",
"valMulta10": 0,
"qtdeCasMul10": 0,
"codValMul10": 0,
"descrMulta10": "String",
"dataPerm10": "String",
"valPerm10": 0,
"dataDesc110": "String",
"valDesc110": 0,
"qtdeCasDe110": 0,
"codValDe110": 0,
"descrDesc110": "String",
"dataDesc210": "String",
"valDesc210": 0,
"qtdeCasDe210": 0,
"codValDe210": -84902961,
"descrDesc210": "String",
"dataDesc310": "String",
"valDesc310": 0,
"qtdeCasDe310": 0,
"codValDe310": 0,
"descrDesc310": "String",
"diasMulta10": 0,
"diasJuros10": 0,
"codBarras10": 0,
"linhaDig10": 0,
"despCart10": 008465,
"bcoCentr10": 0,
"ageCentr10": 0,
"acessEsc10": 0,
"tipEndo10": "String",
"oriProt10": 0,
"corige3510": "String",
"ctpoVencto10": 0,
"codInscrProt10": 0,
"codDecurPrz10": 0,
```

```
"qtdDDecurPrz10": 0,
"ctpoAbat10": 0,
"codComisPerm10": 0,
"ctpoDesc110": 0,
"ctpoDesc210": 0,
"ctpoDesc310": 0,
"ctrlPartic10": "String",
"diasComisPerm10": 0,
"codComisPerm101": 0,
"qmoedaComisPerm": 0,
"cnpjCpfCedente10": 0,
"valorMoedaBol10": 0,
"dataVencBol10": "String",
"indTitParcel10": "String",
"indParcelaPrin10": "String",
"indBoletoDda10": "String",
"dataLimitePgt10": "String",
"dataImpressao10": 0,
"horImpressao10": 0,
"identTitDda10": 0,
"exibeLinDig10": "String",
"permPgtoParcial": "String",
"qtdePgtoParcial": 0,
"filler5": "",
"sFase": 0,
"cindcdCobrMisto": "String",
"aliasAdsaoCta": "String",
"iconcPgtoSpi": "String",
"caliasAdsaoCta": "String",
"ilinkGeracQrcd": "String",
"wqrcdPdraoMerced": "String",
"sInfoAdicNome": "String",
"sInfoAdicCpfCnpj": "String",
"validadeAposVencimento": "String",
"qFiller6": ""
}
```

6. LAYOUT DE COMUNICAÇÃO - ENTRADA

Nome do Campo	Tipo	Descrição	Qtde.	Campo Obrigatório
registrarTitulo	Numérico	Registrar título 1 = Registrar o título 2 = Somente consistir dados do título	1	Sim
codUsuario	Alfanumérico	Código do Usuário responsável	9	Sim Fixo = APISERVIC
nroCpfCnpjBenef	Numérico	Número de Inscrição do CNPJ ou CPF do Beneficiário (Cedente)	9	Sim
filCpfCnpjBenef	Numérico	Número da Filial do CNPJ do Beneficiário (Cedente) Obs.: incluir zeros quando se tratar de CPF	4	Sim
digCpfCnpjBenef	Numérico	Dígitos verificadores do CNPJ ou CPF do Beneficiário (Cedente)	2	Sim
tipoAcesso	Numérico	Tipo de acesso desejado	1	Sim Fixo = 2
cpssoaJuridContr	Numérico	Código da pessoa jurídica do contrato	10	Sim Fixo = 0
ctpoContrNegoc	Numérico	Tipo do Contrato	3	Sim Fixo = 0
nseqContrNegoc	Numérico	Número do Contrato	10	Sim Fixo = 0
cidtfdProdCobr	Numérico	Identificador do Produto Cobrança (Carteira)	2	Sim
cnegocCobr	Numérico	Número do Contrato (Negociação Agência + Conta)	18	Sim
codigoBanco	Numérico	Código do Banco	3	Sim Fixo = 237
filler	Alfanumérico	Implementações futuras		Manter em branco
eNseqContrNegoc	Numérico	Número do Contrato	10	Sim Fixo = 0
tipoRegistro	Numérico	Tipo de registro do título: 1 = Título 2 = Título com Instrução de Protesto 3 = Título com Instrução de Protesto Falimentar	3	Não
cprodtServcOper	Numérico	Código do Produto Cobrança	8	Sim Fixo = 0
ctitloCobrCdent	Numérico	Nosso Número	11	Não
ctitloCliCdent	Alfanumérico	Identificador do título pelo beneficiário (Seu Número)	25	Sim
demisTitloCobr	Alfanumérico	Data de emissão do título. Deve ser informada como o exemplo 01.01.2001	10	Sim
dvctoTitloCobr	Alfanumérico	Data de vencimento do título Deve ser informada como o exemplo 01.01.2001	10	Sim
cidtfdTpoVcto	Numérico	Identificador do tipo de vencimento 0 = Data fixa 1 = Contra-apresentação 2 = À vista	1	Sim Fixo = 0

Nome do Campo	Tipo	Descrição	Qtde.	Campo Obrigatório
cindcdEconmMoeda	Numérico	Identificador da moeda do título (BACEN)	5	Sim
vnmnalTitloCobr	Numérico	Valor nominal do título	17	Sim
qmoedaNegocTitlo	Numérico	Quantidade de moeda do título	17	Sim Fixo = 0
cespceTitloCobr	Numérico	Código da espécie do título Ex. 02 = DM	2	Sim
cindcdAceitSacdo	Alfanumérico	Identificador de aceite do devedor (Sacado)	1	Sim Fixo = N
ctpoProteTitlo	Numérico	Tipo de protesto automático do título 1 = Dias corridos 2 = Dias úteis	2	Não
ctpoPrzProte	Numérico	Quantidade de dias após o vencimento, para protesto automático	2	Sim, caso informado ctpoProteTitlo
ctpoProteDecurs	Numérico	Tipo decurso de protesto 1 = Dias corridos 2 = Dias úteis	2	Sim, caso informado ctpoProteTitlo
ctpoPrzDecurs	Numérico	Quantidade de dias para decurso de protesto	2	Sim Fixo = 0
cctrlPartcTitlo	Numérico	Controle do participante	25	Não
cformaEmisPplta	Numérico	Forma de emissão do boleto (Papeleta) 01 = Banco emite 02 = Cliente emite 03 = Banco envia e-mail 04 = Banco envia sms	2	Sim Fixo = 2
cindcdPgtoParcial	Alfanumérico	Indicador de pagamento parcial S = Sim N = Não	1	Sim Fixo = Não
qtdePgtoParcial	Numérico	Quantidade de pagamento parcial de 001 a 099	3	Sim Fixo = 000
filler1	Alfanumérico	Implementações futuras	18	Manter em branco
ptxJuroVcto	Numérico	Percentual de juros após vencimento Deve ser informado com 5 decimais (Exemplo: 2% = 2.00000 0,05% = 0.05000)	6	Não
vdiaJuroMora	Numérico	Valor diário de juros após vencimento Deve ser informado com 2 decimais (Exemplo: R\$ 2,00 = 200 R\$ 0,05 = 005)	17	Não
qdiaInicJuro	Numérico	Quantidade de dias após o vencimento, para incidência de juros	2	Sim, caso informado valor ou percentual de Juros

Nome do Campo	Tipo	Descrição	Qtde.	Campo Obrigatório
pmultaAplicVcto	Numérico	Percentual de multa após vencimento Deve ser informado com 5 decimais (Exemplo: 2% = 2.00000 0,05% = 0.05000)	6	Não
vmultaAtrsoPgto	Numérico	Valor da multa após vencimento Deve ser informado com 2 decimais (Exemplo: R\$ 2,00 = 200 R\$ 0,05 = 005)	17	Não
qdiaInicMulta	Numérico	Quantidade de dias após o vencimento, para incidência de multa	3	Sim, caso informado valor ou percentual de Multa
pdescBonifPgto01	Numérico	1° - Percentual de desconto Deve ser informado com 5 decimais (Exemplo: 2% = 2.00000 0,05% = 0.05000)	6	Não
vdscBonifPgto01	Numérico	1° - Valor de desconto Deve ser informado com 2 decimais (Exemplo: R\$ 2,00 = 200 R\$ 0,05 = 005)	17	Não
dlimDescBonif1	Alfanumérico	1° - Data-limite de desconto Deve ser informada como o exemplo 01.01.2001	10	Sim, caso informado valor ou percentual do 1° desconto/ bonificação
pdescBonifPgto02	Numérico	2° - Percentual de desconto Deve ser informado com 5 decimais (Exemplo: 2% = 2.00000 0,05% = 0.05000)	6	Não
vdscBonifPgto02	Numérico	2° - Valor de desconto Deve ser informado com 2 decimais (Exemplo: R\$ 2,00 = 200 R\$ 0,05 = 005)	17	Não
dlimDescBonif2	Alfanumérico	2° - Data-limite de desconto Deve ser informada como o exemplo 01.01.2001	10	Sim, caso informado valor ou percentual do 2° desconto
pdescBonifPgto03	Numérico	3° - Percentual de desconto Deve ser informado com 5 decimais (Exemplo: 2% = 2.00000 0,05% = 0.05000)	6	Não
vdscBonifPgto03	Numérico	3° - Valor de desconto Deve ser informado com 2 decimais (Exemplo: R\$ 2,00 = 200 R\$ 0,05 = 005)	17	Não
dlimDescBonif3	Alfanumérico	3° - Data-limite de desconto Deve ser informada como o exemplo 01.01.2001	10	Sim, caso informado valor ou percentual do 3° desconto
ctpoPrzCobr	Numérico	Tipo de prazo desconto/bonificação 1 = Dias corridos 2 = Dias úteis	2	Sim, caso informado valor ou percentual de desconto/ bonificação

Nome do Campo	Tipo	Descrição	Qtde.	Campo Obrigatório
pdescBonifPgto	Numérico	Percentual de desconto Deve ser informado com 5 decimais (Exemplo: 2% = 2.00000 0,05% = 0.05000)	6	Não
vdescBonifPgto	Numérico	Valor de bonificação Deve ser informado com 2 decimais (Exemplo: R\$ 2,00 = 200 R\$ 0,05 = 005)	17	Não
dlimBonifPgto	Alfanumérico	Data-limite de bonificação Deve ser informada como o exemplo 01.01.2001	10	Sim, caso informado valor ou percentual de bonificação
vabtmTitoCobr	Numérico	Valor de abatimento do título	17	Não
viofPgtoTito	Numérico	Valor de IOF do título	17	Não, usar apenas para seguradoras
filler2	Alfanumérico	Implementações futuras	0	Manter em branco
isacdoTitoCobr	Alfanumérico	Nome do devedor (Sacado)	70	Sim
elogdrSacdoTito	Alfanumérico	Logradouro do devedor (Sacado)	40	Sim
enroLogdrSacdo	Numérico	Número do logradouro do devedor (Sacado)	10	Sim
ecomplLogdrSacdo	Alfanumérico	Complemento do logradouro do devedor (Sacado)	15	Não
ccepSacdoTito	Numérico	CEP do devedor (Sacado)	5	Sim
ccomplCepSacdo	Numérico	Complemento do CEP do devedor (Sacado)	3	Sim
ebairoLogdrSacdo	Alfanumérico	Bairro do logradouro do devedor (Sacado)	40	Sim
imunSacdoTito	Alfanumérico	Município do devedor (Sacado)	30	Sim
csglUfSacdo	Alfanumérico	Sigla da Unidade Federativa do devedor (Sacado)	2	Sim
indCpfCnpjSacdo	Numérico	Indicador de CPF ou CNPJ do devedor (Sacado) 1 = CPF 2 = CNPJ	1	Sim
nroCpfCnpjSacdo	Numérico	Número do CPF ou CNPJ do devedor (Sacado)	14	Sim
renderEletrSacdo	Alfanumérico	Endereço eletrônico do devedor - e-mail (Sacado)	70	Não
cdddFoneSacdo	Numérico	DDD do telefone do devedor (Sacado)	3	Não
cfoneSacdoTito	Numérico	Número do telefone do devedor (Sacado)	11	Não
bancoDeb	Numérico	Código do Banco para débito automático	3	Não
agenciaDeb	Numérico	Número da Agência para débito automático	5	Não
agenciaDebDv	Numérico	Dígito verificador da Agência para débito automático	1	Sim, caso informado agenciaDeb
contaDeb	Numérico	Número da conta para débito automático	13	Sim, caso informado agenciaDeb

Nome do Campo	Tipo	Descrição	Qtde.	Campo Obrigatório
bancoCentProt	Numérico	Código do Banco de protesto	3	Sim Fixo = 0
agenciaDvCentPr	Numérico	Número da Agência de protesto	5	Sim Fixo = 0
isacdrAvalsTitlo	Alfanumérico	Nome do sacador avalista (Beneficiário final)	40	Não
elogdrSacdrAvals	Alfanumérico	Logradouro do sacador avalista (Beneficiário final)	40	Sim, caso informado dados do sacador avalista (Beneficiário final)
enroLogdrSacdr	Numérico	Número do logradouro do sacador avalista (Beneficiário final)	10	Sim, caso informado dados do sacador avalista (Beneficiário final)
ecomplLogdrSacdr	Alfanumérico	Complemento do logradouro do sacador avalista (Beneficiário final)	15	Sim, caso informado dados do sacador avalista (Beneficiário final)
ccepSacdrTitlo	Numérico	CEP do sacador avalista	5	Sim, caso informado dados do sacador avalista (Beneficiário final)
ccomplCepSacdr	Numérico	Complemento do CEP do sacador avalista	3	Sim, caso informado dados do sacador avalista (Beneficiário final)
ebairroLogdrSacdr	Alfanumérico	Bairro do logradouro do sacador avalista	40	Sim, caso informado dados do sacador avalista (Beneficiário final)
imunSacdrAvals	Alfanumérico	Município do sacador avalista	40	Sim, caso informado dados do sacador avalista (Beneficiário final)
csglUfSacdr	Alfanumérico	Sigla da Unidade Federativa do sacador avalista	2	Sim, caso informado dados do sacador avalista (Beneficiário final)
indCpfCnpjSacdr	Numérico	Indicador de CPF ou CNPJ do sacador avalista 1 = CPF 2 = CNPJ	1	Sim, caso informado dados do sacador avalista (Beneficiário final)
nroCpfCnpjSacdr	Numérico	Número do CPF ou CNPJ do sacador avalista	14	Sim, caso informado dados do sacador avalista (Beneficiário final)

Nome do Campo	Tipo	Descrição	Qtde.	Campo Obrigatório
renderEletrSacdr	Alfanumérico	Endereço eletrônico do sacador avalista - <i>e-mail</i>	70	Não
cdddFoneSacdr	Numérico	DDD do telefone do sacador avalista	3	Não
cfoneSacdrTitlo	Numérico	Número do telefone do sacador avalista	11	Não
filler3	Alfanumérico	Implementações futuras	0	Sim Fixo = 0
fase	Numérico	Fase de atualização do <i>QR Code</i> : 1 = Registro do título e envio ao BSPI 2 = Vinculação do título com <i>QR Code</i>	1	Sim Fixo = 1
cindcdCobrMisto	Alfanumérico	Indicador do registro de título com <i>QR Code</i> S = Sim N = Não	1	Sim Fixo = S
ialiasAdsaoCta	Alfanumérico	Chave Pix do beneficiário	100	Manter em branco
iconcPgtoSpi	Alfanumérico	TXID do título	35	Manter em branco
caliasAdsaoCta	Alfanumérico	Códigos de erro na geração do <i>QR Code</i> pelo BSPI	15	Manter em branco
ilinkGeracQrcd	Alfanumérico	Identificação do <i>location</i> do <i>QR Code</i> gerado pelo BSPI	77	Manter em branco
wqrcdPdraoMerced	Alfanumérico	Código EMV do <i>QR Code</i> gerado pelo BSPI	500	Manter em branco
validadeAposVencimento	Alfanumérico	Quantidade de dias após vencimento, que o título é válido para pagamento via Pix	4	Manter em branco
filler4	Alfanumérico	Implementações futuras	0	Manter em branco

Observações sobre o preenchimento dos campos:

1. Todos os campos devem ser informados conforme *layout*, entretanto, para os tipos **não obrigatórios**, devem ser preenchidos com zeros para campo numérico, ou espaços para campo alfanumérico.
2. Nos campos referentes à multa, juros, bonificação ou desconto, devem ser preenchidos com valor ou percentual, nunca os dois campos mutuamente.

7. LAYOUT DE COMUNICAÇÃO - RETORNO

Nome do Campo	Tipo	Descrição	Qtde.
cidtfdProdCobr	Numérico	Identificador do Produto Cobrança (Carteira)	2
cnegocCobr	Numérico	Número do Contrato (Negociação Agência + Conta)	18
cpssoaJuridContr	Numérico	Código da pessoa jurídica do contrato	10
ctpoContrNegoc	Numérico	Tipo do Contrato	3
nseqContrNegoc	Numérico	Número do Contrato	10
cprodtServcOper	Numérico	Código do Produto Cobrança	8
ctitloCobrCdent	Numérico	Nosso Número	11
tp08Reg1	Numérico	Tipo de registro 1 - Dados do Sacado / Beneficiário	1
agencCred10	Numérico	Número da Agência de crédito do beneficiário	5
ctaCred10	Numérico	Número da conta de crédito do beneficiário	13
digCred10	Numérico	Dígito verificador da conta de crédito do beneficiário	2
cip10	Numérico	Código da cartela de instrução permanente	3
codStatus10	Numérico	Status do título Ex. 01	2
status10	Alfanumérico	Descrição do status do título Ex. A Vencer/Vencido	40
cedente10	Alfanumérico	Descrição do Campo Nome do Cedente Ex. Nome do Cedente	40
endCedente10	Alfanumérico	Descrição do endereço do cedente Ex. Endereço Cedente	40
nroEndCed10	Numérico	Número do logradouro do beneficiário (Cedente)	7
comEndCed10	Alfanumérico	Complemento do logradouro do beneficiário (Cedente)	20
baiCedente10	Alfanumérico	Descrição do Campo Bairro do Cedente Ex. Bairro Cedente	20
cepEndCed10	Numérico	CEP do beneficiário (Cedente)	5
cepcCedente10	Numérico	Complemento do CEP do beneficiário (Cedente)	3
cidCedente10	Alfanumérico	Município do beneficiário (Cedente)	50
ufCedente10	Alfanumérico	Sigla da Unidade Federativa do beneficiário (Cedente)	2
razCred10	Alfanumérico	Razão de crédito	6
nomeSacado10	Alfanumérico	Nome do devedor (Sacado)	40
cnpjSacado10	Numérico	CPF ou CNPJ do devedor (Sacado)	15
endSacado10	Alfanumérico	Endereço do devedor (Sacado)	40
baiSacado10	Alfanumérico	Bairro do logradouro do devedor (Sacado)	20
cidSacado10	Alfanumérico	Município do devedor (Sacado)	40
ufSacado10	Alfanumérico	Sigla da Unidade Federativa do devedor (Sacado)	2

Nome do campo	Tipo	Descrição	Qtde.
cepSacado10	Numérico	CEP do devedor (Sacado)	5
cepcSacado10	Numérico	Complemento do CEP do devedor (Sacado)	3
cebp10	Alfanumérico	Identificador de rateio de crédito S = Sim N = Não	1
debitoAuto10	Alfanumérico	Identificador de débito automático	1
aceite10	Alfanumérico	Identificador de aceite do devedor (Sacado) S = Sim N = Não	1
enderecoEma10	Alfanumérico	Endereço eletrônico do devedor - <i>e-mail</i> (Sacado)	50
nomeSacador10	Alfanumérico	Nome do sacador avalista	40
cnpjSacador10	Numérico	CPF ou CNPJ do sacador avalista	15
endSacador10	Alfanumérico	Endereço do sacador avalista	40
cidSacador10	Alfanumérico	Município do sacador avalista	40
ufSacador10	Alfanumérico	Sigla da Unidade Federativa do sacador avalista	2
cepSacador10	Numérico	CEP do sacador avalista	5
cepcSacador10	Numérico	Complemento do CEP do sacador avalista	3
filler4	Alfanumérico	Implementações futuras	28
tp08Reg2	Numérico	Tipo de registro 2 - Dados do título pendente	1
cense10	Alfanumérico	Identificador do subcentro	5
agenOper10	Alfanumérico	Número da Agência operadora	5
bcoDepos10	Numérico	Código do Banco depositário	3
agenDepos10	Numérico	Número da Agência depositária	5
snumero10	Alfanumérico	Seu Número	15
dataReg10	Numérico	Data de registro do título	8
especDocto10	Alfanumérico	Sigla da Espécie do título	3
descrEspec10	Alfanumérico	Descrição da espécie do título	40
valorIof10	Numérico	Valor do IOF	15
dataEmis10	Numérico	Data de emissão do título (DD.MM.AAAA)	8
especMoeda10	Alfanumérico	Sigla da moeda do título Ex. R\$	5
qtdeMoeda10	Numérico	Quantidade de moeda do título	15
qtdeCas10	Numérico	Quantidade de casas decimais da moeda do título	1
dataVenc10	Numérico	Data de vencimento do título (DD.MM.AAAA)	10
descrMoeda10	Alfanumérico	Sigla do indicador econômico (moeda) Ex. R\$	5
valMoeda10	Numérico	Valor nominal do título	15
valAbat10	Numérico	Valor do abatimento do título	15
dataInstr10	Numérico	Data de instrução de protesto	8
diasProt10	Numérico	Quantidade de dias para protesto automático	3
dataCartor10	Numérico	Data de envio ao cartório	8

Nome do Campo	Tipo	Descrição	Qtde.
numCartor10	Numérico	Número do cartório de protesto	5
numProtoc10	Numérico	Número do protocolo de protesto	15
dataPedSus10	Numérico	Data da solicitação de sustação do protesto	8
dataSust10	Numérico	Data da efetivação da sustação de protesto	8
dataMulta10	Numérico	Data da multa após vencimento	8
qtdeCasMul10	Numérico	Quantidade de casas decimais do valor da multa	1
codValMul10	Numérico	Código do valor da multa 1 = Valor 2 = Percentual	1
descrMulta10	Alfanumérico	Descrição da multa Ex. Valor Fixo ou Taxa Mensal ou Isento	40
dataPerm10	Numérico	Data da comissão de permanência após vencimento (juros)	8
valPerm10	Numérico	Valor da comissão de permanência após vencimento (juros)	15
dataDesc110	Numérico	Data-limite do primeiro desconto	8
valDesc110	Numérico	Valor do primeiro desconto	15
qtdeCasDe110	Numérico	Quantidade de casas decimais do valor do primeiro desconto	1
codValDe110	Numérico	Código do valor do primeiro desconto 1 = Valor 2 = Percentual	1
descrDesc110	Alfanumérico	Descrição do primeiro desconto Ex. Valor Fixo por Antecipação até a Data Percentual por Antecipação até a Data	40
dataDesc210	Numérico	Data-limite do segundo desconto	8
valDesc210	Numérico	Valor do segundo desconto	15
qtdeCasDe210	Numérico	Quantidade de casas decimais do valor do segundo desconto	1
codValDe210	Numérico	Código do valor do segundo desconto 1 = Valor 2 = Percentual	1
descrDesc210	Numérico	Descrição do segundo desconto Ex. Valor Fixo por Antecipação até a Data Percentual por Antecipação até a Data	40
dataDesc310	Numérico	Data-limite do terceiro desconto	8
valDesc310	Numérico	Valor do terceiro desconto	15
qtdeCasDe310	Numérico	Quantidade de casas decimais do valor do terceiro desconto	1
codValDe310	Numérico	Código do valor do terceiro desconto 1 = Valor 2 = Percentual	1

Nome do Campo	Tipo	Descrição	Qtde.
numCartor10	Numérico	Número do cartório de protesto	5
descrDesc310	Alfanumérico	Descrição do terceiro desconto Ex. Valor Fixo por Antecipação até a Data Percentual por Antecipação até a Data	40
diasMulta10	Numérico	Quantidade de dias após o vencimento, para início da incidência de multa	3
diasJuros10	Numérico	Quantidade de dias após o vencimento, para início da incidência de juros	3
codBarras10	Numérico	Código de barras do título	112
linhaDig10	Numérico	Linha digitável do título	54
despCart10	Numérico	Valor das despesas cartorárias	15
bcoCentr10	Numérico	Código do Banco de protesto	3
ageCentr10	Numérico	Número da Agência de protesto	5
acessEsc10	Numérico	Número do acessório escritural	18
tipEndo10	Numérico	Tipo de endosso do título	1
tipEndo10	Numérico	Tipo de endosso do título	1
oriProt10	Numérico	Origem do protesto	2
corige3510	Numérico	Código de origem	1
ctpoVenc10	Numérico	Tipo de vencimento	1
codInscrProt10	Numérico	Código da instrução de protesto do título 1 = Dias corridos 2 = Dias úteis	1
codDecurPrz10	Numérico	Código para decurso de prazo 1 = Dias corridos 2 = Dias úteis	1
qtdDDecurPrz10	Numérico	Quantidade de dias para decurso de prazo do título após o vencimento	3
ctpoAbat10	Numérico	Tipo de abatimento	1
codComisPerm10	Numérico	Código da comissão de permanência (juros)	1
ctpoDesc110	Numérico	Tipo do primeiro desconto	1
ctpoDesc210	Numérico	Tipo do segundo desconto	1
ctpoDesc310	Numérico	Tipo do terceiro desconto	1
ctrlPartic10	Alfanumérico	Controle do participante	25
diasComisPerm10	Numérico	Quantidade de dias para incidência da comissão de permanência após vencimento (juros)	3
codComisPerm101	Numérico	Código da comissão de permanência (juros)	1
qmoedaComisPerm	Numérico	Quantidade de moeda da comissão de permanência (juros)	15
cnpjCpfCedente10	Numérico	CPF ou CNPJ do beneficiário (Cedente)	15
valorMoedaBol10	Numérico	Valor do boleto em moeda vigente	15
dataVencBol10	Numérico	Data de vencimento do título (DD.MM.AAAA)	10
indTitParcel10	Alfanumérico	Indicador de título parcelado	1
indParcelaPrin10	Numérico	Indicador da primeira parcela	1
indBoletoDda10	Alfanumérico	Indicador de boleto DDA S = Sim N = Não	1
dataLimitePgt10	Numérico	Data-limite para pagamento do título	10

Nome do Campo	Tipo	Descrição	Qtde.
dataImpressao10	Numérico	Data da impressão do boleto	8
horaImpressao10	Numérico	Hora da impressão do boleto	6
identTitDda10	Numérico	Identificação do título DDA	17
exibeLinDig10	Numérico	Indicador de exibição da linha digitável S = Sim N = Não	1
permPgtoParcial	Alfanumérico	Indicador de pagamento parcial S = Sim N = Não	1
qtdePgtoParcial	Numérico	Quantidade de pagamento parcial	3
filler5	Alfanumérico	Implementações futuras	32
sFase	Numérico	Fase de atualização do <i>QR Code</i> : 1 = Registro do título e envio ao BSPI 2 = Vinculação do título com <i>QR Code</i>	1
cindcdCobrMisto	Alfanumérico	Indicador do registro de título com <i>QR Code</i> S = Sim N = Não	1
ialiasAdsaoCta	Alfanumérico	Chave Pix do beneficiário	100
iconcPgtoSpi	Alfanumérico	TXID do título	35
caliasAdsaoCta	Alfanumérico	Códigos de erro na geração do <i>QR Code</i> pelo BSPI	15
ilinkGeracQrcd	Alfanumérico	Identificação do <i>location</i> do <i>QR Code</i> gerado pelo BSPI	77
wqrcdPdrazMercd	Alfanumérico	Código EMV do <i>QR Code</i> gerado pelo BSPI	500
validadeAposVencimento	Numérico	Quantidade de dias após o vencimento, que o título é válido para pagamento via Pix	4
qFiller4	Alfanumérico	Implementações futuras	13

8. MENSAGENS DE RETORNO

DESCRIÇÃO DAS MENSAGENS DE RETORNO

Status-Header	Descrição
-99	Serviço indisponível no momento. Tente novamente mais tarde.
-4	Tamanho do campo inválido
-3	Tipo do campo inválido
-2	Contrato não encontrado
-1	Contrato não aprovado
0	Solicitação atendida
1	Solicitação não encontrada
2	Erro Genérico - sistema indisponível
5	Inclusão efetuada
6	Dados inconsistentes
10	Erro Acesso Sub-rotina
12	Cliente/Negociação Bloqueado
13	Usuário não Autorizado
14	Espécie Título Inválida
15	Tipo/Número Inscrição Inválido
16	Informe todos os campos para decurso de Prazo
17	Nome do Pagador Especial não Informado
18	Endereço Inválido
19	CEP Inválido
20	Agência Depositária Inválida
21	Informe todos os campos para Instrução de Protesto
22	Banco Inválido
23	Seu Número Inválido
24	Informe todos os campos para Abatimento
25	Valor dos Juros maior que o Valor do Título
26	Data de Emissão maior que a Data de Vencimento
27	Documento do Sacador Avalista Inválido
28	Informe todos os campos para Desconto
29	Informe todos os campos para Sacador Avalista
30	Data Vencimento menor ou igual Data Emissão
31	Data Desconto menor ou igual Data Emissão
32	Data Desconto maior que Data Vencimento
33	Valor Desconto/Bonificação maior ou igual Valor Título
34	Tipo informado deve ser 1, 2 ou 3
35	Valor Abatimento maior que o Valor do Título
36	CEP Inválido
37	Data Emissão Inválida
38	Data Vencimento Inválida
39	Percentual informado maior ou igual 100,00
40	Número CGC/CPF inválido
41	Protesto Automático x Decurso de Prazo Incompatível
42	Banco/Agência Depositária Inválido
43	Espécie de Documento inválido
44	Informe 1-Contra-apresentação ou 2-À vista
45	Código da instrução de protesto inválido
46	Dias para instrução de protesto inválido
47	Código para desconto inválido
48	Código para multa inválido

49	Código para comissão permanência dia inválido
50	Espécie Documento exige CGC para Sacador Avalista
51	CEP e/ou Banco/Agência Depositária Inválido
52	Data Emissão maior ou igual Data Vencimento
53	Data Desconto Inválida
54	Data emissão maior Data Registro
55	Percentual multa informado maior que o permitido
56	Percentual comissão permanência informado maior que o permitido
57	Percentual Bonificação informado maior que o permitido
58	Prazo para Protesto inválido
59	Informe a data ou tipo do vencimento
60	Valor do IOF não permitido para produtos 05,15,43 ou 44
61	Abatimento já cadastrado para o título
62	Abatimento não cadastrado para o título
63	Não é permitida mais de uma bonificação para o título
64	Não são permitidas datas de desconto/bonificação iguais
65	Negociação inexistente
66	Cliente inexistente
67	CNPJ/CPF inválido
68	N. Número não pode ser informado quando <i>status</i> 4
69	Título já cadastrado
70	Data e tipo de vencimento incompatíveis
71	Data de vencimento não pode ser posterior a 10 anos
72	Dias para instrução inferior ao padrão
73	Dias para instrução antecipa data de protesto
74	Valor IOF obrigatório
75	Valor IOF incompatível com Id produto
76	Tipo de abatimento inválido
77	<i>Status</i> Inválido
78	Registro <i>on-line</i> não permite Banco diferente de 237
79	Carta para protesto não recebida
80	Tipo de vencimento inválido
81	Valor acumulado desconto/bonificação maior ou igual valor título
82	Datas desconto/bonificação fora de sequência
83	Informe todos os campos para multa
84	Código comissão permanência inválido
85	Informe todos os campos para comissão permanência
86	Registro duplicado na tabela de ocorrências
87	Solicitação de protesto já existente
88	Registro duplicado na base de atualização sequencial
89	Sacador avalista já cadastrado

Status-Header	Descrição
90	Indicador CIP inexistente
91	Moeda negociada inexistente
92	Banco/Agência operadora inexistente
93	Acessório escritural negociado inexistente
94	Polo de serviço inexistente para Banco/Agência
95	Banco/Agência centralizadora não cadastrada para Banco/Agência depositária
96	Título não encontrado pelo módulo CBON8230
97	Valor IOF maior ou igual valor título
98	Data Inválida
99	Id Prod/Cta não cadastrados

DESCRIÇÃO DAS MENSAGENS DE RETORNO

Erro de geração de boleto com QR CODE
Título Incluído com Sucesso sem <i>Qr Code</i>
Título Incluído com Sucesso com <i>Qr Code</i>
Cliente com Cobrança Mista não Negociada
Chave Pix Inválida
Chave Pix não Cadastrada
Chave Pix não Compatível com CPF/CNPJ
Chave Pix não pode ser maior que 77 caracteres
Txid Inválido
Txid em Duplicidade
Dados do <i>Qr Code</i> Inconsistentes

9. TABELA DE CÓDIGO DE ESPÉCIE DE TÍTULOS

CÓDIGO	SIGLA	DESCRIÇÃO
1	CH	Cheque
2	DM	Duplicata de Venda Mercantil
3	DMI	Duplicata Mercantil por Indicação
4	DS	Duplicata de Prestação de Serviços
5	DSI	Duplicata Prest. Serviços por Indicação
6	DR	Duplicata Rural
7	LC	Letra de Câmbio
8	NCC	Nota de Crédito Comercial
9	NCE	Nota de Crédito Exportação
10	NCI	Nota de Crédito Industrial
11	NCR	Nota de Crédito Rural
12	NP	Nota Promissória
13	NPR	Nota Promissória Rural
14	TM	Triplicata de Venda Mercantil
15	TS	Triplicata de Prestação de Serviços
16	NS	Nota de Serviço
17	RC	Recibo
18	FAT	Fatura
19	ND	Nota de Débito
20	AP	Apólice de Seguro
21	ME	Mensalidade Escolar
22	PC	Parcela de Consórcio
23	DD	Documento de Dívida
24	CCB	Cédula de Crédito Bancário
25	FI	Financiamento
26	RD	Rateio de Despesas
27	DRI	Duplicata Rural Indicação
28	EC	Encargos Condominiais
29	ECI	Encargos Condominiais por Indicação
31	CC	Cartão de Crédito. Obs.: não disponível no momento para QR Code
32	BDP	Boleto de Proposta
99	OUT	Outros

10. FAQ

a. Regras de Conexão da API

1. Qual o procedimento para consumo das APIs?

Para consumir as *Open APIs*, são necessárias duas requisições:

- ✓ Obtenção do *access-token*;
- ✓ Consultar o *endpoint* do serviço.

Os passos seguintes descreverão como fazer as duas requisições de forma manual, usando as ferramentas de requisições *HTTP*, *Postman* e a biblioteca *openssl*. Os fluxos descritos devem ser desenvolvidos para funcionar, de forma automatizada, nos sistemas consumidores.

2. Como obter o *access-token*?

O *token* de acesso às APIs é concebido após o fornecimento de um *JWS* (após assinar digitalmente o *JWT*, ele torna-se um *JWS*), informando o ID da aplicação em um de seus *claims*, tal valor é gerado no passo anterior após o compartilhamento do certificado público.

No *response* da requisição estará presente o *token* a ser usado na autenticação do tipo *Bearer*.

3. Como obter o *JWT*?

Para realizar a geração do *JWT*, será necessário fazer o *encode* dos dois *JSONs* citados para *base64 url encoded*, além da remoção de todos os espaços e quebras de linha, operação conhecida como “*stringify*”.

Comando: `echo -n "$(cat <arquivo>)" | tr -d '[:space:]' | base64 | tr -d '[:space:]' | tr '+/' '-_'`

O comando “`echo -n "$(cat <arquivo>)"`” fará a leitura do arquivo, a Opção “`-n`” é para que seja feita a leitura e propagação da *string* sem quebra de linha no final. Por sua vez, o Comando “`tr -d '[:space:]'`” faz a operação “*stringify*”, removendo caracteres especiais (espaço, *line feed* – ‘`LF`’ ou ‘`\n`’- e *carriage return* – ‘`CR`’ ou ‘`\r`’-) do arquivo lido. Dessa forma, realiza-se o *encode* para *base64*. O último comando, “`tr -d '[:space:]' | tr '+/' '-_'`”, formata para *URL encoded* e facilita o *copy and paste*, o que faz com o que o *output* não tenha quebras de linha.

Junte os dois resultados, separando-os com o caractere "." (ponto). O formato será:

Header (base64url encoded) + "." + Payload (base64url encoded)

4. Qual o tipo de Certificado utilizado?

Certificado do tipo público.

5. Qual o tipo de extensão do arquivo para a assinatura?

O envio desse certificado deve ser feito de toda a cadeia de certificados - raiz, intermediários e domínio, sendo extensão ".PEM" (base64) e padrão X.509 no formato requisitado. Na composição dos dados, é importante a identificação da empresa, referenciando o CNPJ e Razão Social. O arquivo a ser disponibilizado no formato de texto deve começar com "*Begin Certificate*".

Nota: a chave privada, relacionada ao certificado compartilhado, é de responsabilidade de sua empresa, e deve ser armazenada de forma segura, nunca sendo fornecida a terceiros.

6. Certificado digital em arquivo (.pem), token, instalado ou uma leitora no servidor?

A chave privada do certificado .pem ou .cer deve ficar cadastrada onde a aplicação dele consiga acessar.

7. O serviço de API de pagamento de arrecadação utiliza SOAP ou REST?

Utiliza REST.

8. Como funciona o acesso às APIs Bradesco nos ambientes de homologação e produção?

O acesso utiliza o protocolo TLS 1.2 com os algoritmos (cifras) abaixo:

- ✓ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, e/ou
- ✓ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384.

9. O certificado é emitido pelo Bradesco ou pela empresa certificadora digital?

Pelo Bradesco.

10. Há alteração do certificado?

Sim, quando o mesmo vencer.

11. Temos várias empresas que operam com a arrecadação, e possuímos Certificados Digitais ICP Brasil para cada uma delas. Existe a possibilidade de utilizarmos o mesmo certificado para todas? Alguma parametrização no Bradesco que indique que o certificado vale para todas as empresas, ou devemos acessar utilizando um certificado por empresa?

Cada empresa deve utilizar um certificado exclusivo para ela e um clientID gerado pelo Banco.

12. Necessita de certificados de segurança no nosso domínio (SSL)?

Se o cliente possuir uma configuração de confiabilidade de acesso a *URLs* externas no servidor dele, sim.

13. Existe certificado homologado (válido) para testes?

Cada ambiente deve utilizar um certificado exclusivo para o ambiente utilizado.

No caso de homologação, o cliente deve gerar um certificado conforme o manual e solicitar a implantação no ambiente de homologação. O mesmo deve ocorrer para o ambiente de produção.

O padrão de certificado é igual para qualquer ambiente, mas o certificado gerado deve ser diferente em cada ambiente.

14. O que é origem do *request*?

É o pedido encaminhado pelo cliente ao Banco para o processamento de alguma transação ou consulta para determinado pagamento.

15. Para realizar a homologação dos pagamentos e assinar a mensagem, é obrigatório o uso do certificado digital da empresa cadastrada no Banco ou posso utilizar o certificado de uma outra empresa que não está cadastrada no Banco?

Para acesso a qualquer API, é necessário que o usuário, ou quem for testar, tenha um certificado que foi implantado no nosso *gateway* e um clientID gerado para ele mesmo.

16. Se, na comunicação com o Bradesco, a empresa não conseguir receber a mensagem de retorno, como deveremos proceder?

O pagamento deverá ser consultado por meio da API de consulta pelo ID transação.

b. Regras de Negócio

1. Qual o tempo médio entre a solicitação de registro do boleto com QR Code e a efetivação do registro na CIP?

1 segundo.

2. Se, por algum motivo, o serviço *on-line* fique indisponível por tempo indeterminado ou por elevado tempo impedindo o registro do boleto com QR Code, temos alguma contingência por parte do Bradesco?

O cliente pode fazer o registro via arquivo-remessa.

3. No campo “nuCliente” que informação devo passar?

Essa informação é de responsabilidade do cliente, não é consistida pelo Banco, é o equivalente a “Seu Número”, normalmente utilizada para informar o número da nota fiscal.

4. Posso utilizar qualquer tipo de Chave Pix para o boleto com QR Code?

Sim, podem ser usadas as Chaves CNPJ, Telefone, *e-mail* ou aleatória, desde que esteja atrelada ao mesmo CNPJ do contrato de Cobrança.

5. É possível registrar boleto com QR Code com desconto, abatimento, juros e multa?

Sim, os dados do boleto serão replicados no QR Code.

6. Qual o prazo máximo para vencimento do boleto com QR Code?

Para registros via API, o vencimento máximo é de 3 anos. Em breve, será ampliado para 6 anos (consulte seu Gerente para mais informações).

7. Posso registrar boleto com QR Code usando os serviços de protesto e negativação?

Sim, porém quando o boleto entrar em situação de protesto, o QR Code será baixado.

8. Qual tipo de QR Code poderá ser usado nesse produto?

Somente QR Code dinâmico.

9. É possível registrar boleto com QR Code que permita recebimento de valor divergente?

Não. Nessa modalidade, deve ser pago exatamente o valor calculado no momento do pagamento. Está em desenvolvimento a permissão de pagamento em qualquer valor. Para mais informações, consulte seu Gerente.

10. Qual decurso de prazo será permitido para boleto com QR Code?

Será respeitado o decurso de prazo parametrizado no contrato de Cobrança do cliente para o boleto, porém o QR Code ficará em aberto, no máximo, por 60 dias.

11. Se o pagamento for realizado via Pix, o que acontece com o boleto?

Quando o pagamento ocorrer via Pix, o boleto será baixado no processamento noturno.

12. Se o pagamento for realizado via boleto, o que acontece com o QR Code?

Quando o pagamento ocorrer pelo boleto, o QR Code será baixado no processamento noturno.

13. Em caso de alteração dos dados do boleto, o QR Code será alterado?

Sim, o boleto e o QR Code sempre terão a mesma informação.

14. Será permitido pagamento parcial no boleto com QR Code?

Não. Essa solução está em desenvolvimento. Para mais informações, procure o seu Gerente.

15. No Boleto com QR Code, como fica a cobrança de juros em feriados quando pago via Pix?

No Pix, aplica-se feriados locais e nacionais, porém com base no domicílio do pagador registrado. Para QR Code com vencimento em dia não útil, se o pagamento for realizado no dia útil subsequente, não serão cobrados encargos. Lembramos que as regras de pagamento de boleto não sofreram alteração.

16. Como saber se o boleto foi gerado com QR Code?

Quando constar o EMV na resposta da requisição, significa que o boleto foi gerado com QR Code.

17. Será disponibilizado ambiente para teste?

Sim. Abaixo, destacamos as URLs que devem ser usadas.

Homologação: <https://proxy.api.prebanco.com.br>;

Produção: <https://openapi.bradesco.com.br>.

11. SUPORTE

Regras Técnicas da API: em caso de dúvidas técnicas relacionadas à conectividade e ao consumo da API, entrar em contato com a nossa Central de Suporte pelo *e-mail*: suporte.api@bradesco.com.br.

Importante enviar as seguintes informações:

- ✓ O **"request"** e o **"response"** completos da chamada/requisição;
- ✓ O CNPJ e Nome da Empresa que contratou o serviço junto ao Bradesco.