



**pix**  
powered by Banco Central

# Manual de Padrões para Iniciação do Pix

Versão 2.3.0

## Histórico de revisão

Data	Versão	Descrição das alterações
11/8/2020	1.0	Versão inicial
23/9/2020	1.1	<ul style="list-style-type: none"> <li>Ajustes nas definições nos campos do <i>payload</i>, especialmente na semântica do campo <i>txid</i>.</li> <li>Incluídos os Anexos I e II, tratando dos Conceitos de Negócio da API Pix e das suas especificações técnicas.</li> </ul>
13/10/2020	2.0	<ul style="list-style-type: none"> <li>Seção 1.5.2: incluídas explicações sobre caracteres permitidos no campo <i>txid</i>.</li> <li>Seção 1.6.1: incluído texto para deixar claro que o QR Code dinâmico pode ser gerado por meio de aplicativo.</li> <li>Seção 1.6.6: excluído texto para deixar claro que o QR Code dinâmico não precisa ser necessariamente gerado por meio da API Pix.</li> <li>Seção 1.6.8: ajuste na explicação do campo “calendario.vencimento” e exclusão do campo “calendario.recebivelAposVencimento”.</li> <li>Seção 1.6.17: inclusão de nota de rodapé no campo 62 “Additional Data Field”.</li> <li>Seção 1.6.18: incluído texto para deixar claro que o QR Code dinâmico pode ser gerado por meio de aplicativo.</li> <li>Incluída a seção 1.7, que trata sobre a funcionalidade “Pix Copia e Cola”.</li> <li>Seção 6.3.3 do Anexo I: correção da função associada à alteração da cobrança via API Pix.</li> </ul>
06/11/2020	2.1	<ul style="list-style-type: none"> <li>Seção 1.6: atualização dos campos do Payload JSON, com a inclusão de campos referentes às funcionalidades de cobrança para pagamentos com vencimento (juros, multa, abatimento, desconto e correlatos); reorganização das subseções para refletir as diferenças de campos entre as cobranças para pagamentos imediatos e pagamentos com vencimento;</li> <li>Adicionados ao Anexo I casos de uso relacionados ao “Reuso de Location”, cenários incluindo cobrança para pagamentos com vencimento e geração de cobranças em lotes.</li> <li>Inclusão do Anexo III tratando sobre criação, atualização e cálculo de cobranças para pagamentos com vencimento.</li> </ul>
09/12/2020	2.2	<ul style="list-style-type: none"> <li>Seção 1.6.6.2: inserção na tabela que exhibe a estrutura do payload JSON para cobranças com vencimento da informação <b>calendario.validadeAposVencimento</b>.</li> <li>Anexo II – Seção 3.1: inserção de recomendação relativa ao uso de certificados nos webhooks.</li> </ul>
12/02/2021	2.2.1	<ul style="list-style-type: none"> <li>Seção 1.6.6.1: adicionada explicação para a regra de incrementos do campo.</li> <li>Seção 1.6.6.2: adicionados esclarecimentos sobre o funcionamento do campo <b>calendário.validadeAposVencimento</b></li> </ul>

		<ul style="list-style-type: none"> <li>Seção 2.1: alterado de obrigatório para opcional o preenchimento dos campos logradouro, cidade, UF e CEP do campo “devedor” na criação de uma cobrança com vencimento.</li> <li>Seção 1.6.1: Excluído o trecho que erroneamente se referia a um possível valor “0” no campo de valor EMV.</li> </ul>
22/03/2021	2.3.0	<ul style="list-style-type: none"> <li>Seção 1.4: removidas notas de rodapé e adicionadas explicações para explicitar que a regra de formatação das chaves Pix segue as determinações constantes do Manual Operacional do DICT.</li> <li>Seção 1.6.3: removido o fragmento que indica a versão do <i>location</i>.</li> <li>Seção 1.6.6: ajuste redacional para esclarecer que o código do município a ser informado pelo PSP do pagador deve corresponder à informação cadastral de endereço do usuário pagador.</li> <li>Seção 1.6.6.1: adicionado o campo modalidadeAlteracao no objeto “valor” para Pix Cobrança para pagamentos imediatos.</li> <li>Seção 1.6.6.2: inseridos esclarecimentos sobre o comportamento da data de vencimento e da validade após vencimento em caso de fim de semana e de feriado para o usuário pagador.</li> <li>Seção 1.6.7: exemplo revisado: valor (EMV, opcional) retirado; <i>Ref.Label</i> (txid) modificado (ênfatiza que vale o <i>payload</i> da cobrança); fragmento ‘v2’ (tornado opcional) retirado da <i>location</i> (url da cobrança).</li> <li>Anexo III: inseridos esclarecimentos sobre o comportamento da data de vencimento em caso de fim de semana e de feriado para o usuário pagador e sobre os consequentes impactos nos campos que façam referência a esta data (<i>validadeAposVencimento</i>; <i>desconto</i>; <i>juros e multa</i>).</li> </ul>

## Sumário

<b>1. INICIAÇÃO DO PIX POR QR CODE .....</b>	<b>1</b>
1.1. INICIAÇÃO DO PIX COM EMV-MPM: <i>MERCHANT PRESENTED MODE</i> .....	1
1.2. TIPOS DE QR CODES .....	1
1.2.1. QR Code Estático.....	1
1.2.2. QR Code Dinâmico .....	2
1.3. DEFINIÇÕES GERAIS EM RELAÇÃO AO BR CODE .....	2
1.4. FORMATAÇÃO DAS CHAVES DO DICT NO BR CODE .....	3
1.4.1. E-mail .....	3
1.4.2. CPF ou CNPJ .....	3
1.4.3. Número de telefone celular.....	3
1.4.4. Chave aleatória.....	3
1.5. INICIAÇÃO DO PIX VIA QR CODE ESTÁTICO .....	4
1.5.1. Campos Chave e InfoAdicional.....	4
1.5.2. Identificador de transação: txid no QR Code estático.....	5
1.5.3. Mapeamento pacs.008 .....	5
1.5.4. Exemplo de QR Code estático .....	5
1.5.5. O QR Code Estático na API Pix .....	7
1.6. INICIAÇÃO DO PIX VIA QR CODE DINÂMICO .....	7
1.6.1. A API Pix .....	7
1.6.2. O QR Code dinâmico .....	8
1.6.3. A URL hospedada no PSP do recebedor: "Location" .....	9
1.6.4. URL do QR Code dinâmico.....	10
1.6.5. Sobre cobranças concluídas .....	10
1.6.6. O payload JSON.....	10
1.6.7. Exemplo de QR Code Dinâmico .....	18
1.6.8. O QR Code Dinâmico na API Pix .....	20
1.7. PIX COPIA E COLA.....	21
<b>2. MAPEAMENTO PARA MENSAGENS ISO 20022 .....</b>	<b>22</b>
2.1. MAPEAMENTO DO QR CODE ESTÁTICO PARA PACS.008.....	22
2.2. MAPEAMENTO PAYLOAD JSON PARA PACS.008.....	23
<b>ANEXO I – API PIX: CONCEITOS DE NEGÓCIO.....</b>	<b>24</b>
<b>1. INTRODUÇÃO.....</b>	<b>24</b>
<b>2. DOCUMENTAÇÃO DA API PIX.....</b>	<b>24</b>
<b>3. CONTEXTO DA API PIX .....</b>	<b>25</b>
<b>4. CONCEITOS GERAIS.....</b>	<b>26</b>
<b>5. FUNCIONALIDADES DA API PIX .....</b>	<b>27</b>
5.1. DEFINIÇÕES DAS ENTIDADES.....	27
5.2. CARDINALIDADE ENTRE AS ENTIDADES.....	28
5.3. CICLO DE VIDA DO TRANSACTIONID (TXID).....	28
5.3.1 txid no contexto das Cobranças .....	28
5.3.2 txid no contexto dos QR Codes Estáticos .....	29
5.4. GRUPOS DE FUNCIONALIDADES.....	29

<b>6. CASOS DE USO .....</b>	<b>30</b>
6.1 QR CODE ESTÁTICO .....	30
6.1.1 Pagamento no ato da compra com QR Code Estático fixo e sem valor definido. ....	30
6.1.2 Pagamento no ato da compra com QR Code Estático fixo com valor definido.....	31
6.1.3 Pagamento no ato da compra com QR Code Estático gerado no ato da compra .....	32
6.2 QR CODE DINÂMICO .....	32
6.2.1 Pagamento imediato (no ato da compra) com QR Code Dinâmico .....	32
6.2.2 QR Code dinâmico para pagamentos com vencimento.....	33
6.2.3 Lote de Cobranças com vencimento .....	34
6.2.4 O QR Code dinâmico “impresso”.....	35
6.3 OUTROS CASOS DE USO.....	36
6.3.1 Efetuar uma devolução.....	36
6.3.2 Remover uma Cobrança.....	36
6.3.3 Alterar uma cobrança .....	37
6.3.4 Configuração de Webhooks.....	37
<b>ANEXO II – API PIX: ESPECIFICAÇÃO TÉCNICA .....</b>	<b>38</b>
<b>1. INTRODUÇÃO.....</b>	<b>38</b>
<b>2. PROTOCOLOS E TECNOLOGIAS.....</b>	<b>38</b>
<b>3. SEGURANÇA.....</b>	<b>38</b>
3.1 REQUISITOS DE SEGURANÇA OBRIGATÓRIOS .....	38
3.2 RECOMENDAÇÕES DE SEGURANÇA .....	39
3.3 JORNADA DE ADESAO .....	40
<b>ANEXO III – COBRANÇAS PARA PAGAMENTOS COM VENCIMENTO: CRIAÇÃO, ALTERAÇÃO E CÁLCULO .....</b>	<b>41</b>
<b>1. INTRODUÇÃO.....</b>	<b>41</b>
<b>2. CRIANDO UMA COBRANÇA PARA PAGAMENTO COM VENCIMENTO .....</b>	<b>41</b>
<b>2.1 ESTRUTURA PARA CRIAÇÃO E ATUALIZAÇÃO DE UMA COBRANÇA PARA PAGAMENTO COM VENCIMENTO .....</b>	<b>41</b>
<b>2.2 CÁLCULO DO VALOR DA COBRANÇA .....</b>	<b>47</b>
2.2.1 Cálculo do valor de abatimento ( $V_a$ ) .....	47
2.2.2 Cálculo do valor de desconto ( $V_d$ ) .....	48
2.2.3 Cálculo do valor de juros ( $V_j$ ).....	52
2.2.4 Cálculo do valor de multa ( $V_m$ ).....	54

## 1. Iniciação do Pix por QR Code

O QR Code dinâmico e o QR Code estático, enquanto mecanismos para envio ou disponibilização prévia de informações para fins de iniciação de um Pix, seguirão o padrão do BR Code, nos termos do Manual do BR Code<sup>1</sup>. Nesses casos, o usuário recebedor disponibiliza os dados de pagamento em um QR Code, para ser capturado por imagem pelo usuário pagador.

Podem atuar na emissão de QR Code o PSP do recebedor, a Secretaria do Tesouro Nacional (STN) e, em casos de uso específicos, órgãos do governo federal<sup>2</sup>, enquanto usuários finais do Pix, desde que possuam contrato firmado com o BCB para fins de utilização do Sisbacen e que apresentem a certificação de segurança requerida para essa finalidade<sup>3</sup>.

O aplicativo do prestador de serviços de pagamento (PSP) do usuário pagador, que deve estar instalado em seu telefone celular e que é utilizado para a leitura do QR Code, acessará o *backend*<sup>4</sup> do PSP do pagador, que gera a ordem de pagamento.

### 1.1. Iniciação do Pix com EMV-MPM: *Merchant Presented Mode*

O BR Code adota o padrão EMV para uso de QR Codes em sistemas de pagamento (EMV-QRCPS), o qual consiste em dois casos de uso (modos) distintos: apresentado pelo recebedor (*Merchant Presented Mode - MPM*)<sup>5</sup> ou apresentado pelo pagador (*Consumer Presented Mode - CPM*)<sup>6</sup>. O BR Code trata do caso EMV-MPM.

### 1.2. Tipos de QR Codes

#### 1.2.1. QR Code Estático

O QR Code estático apresenta um rol de funcionalidades compacto. São apenas quatro opções de configuração. Primeiramente, o usuário do QR Code estático precisa necessariamente configurá-lo com uma chave válida no Diretório de Identificadores de Contas Transacionais (DICT), nos termos do Manual Operacional do DICT<sup>7</sup>. As outras três configurações são opcionais: identificador da transação<sup>8</sup>, campo de texto livre e valor do pagamento.

---

<sup>1</sup> Disponível em <<https://www.bcb.gov.br/estabilidadefinanceira/arranjosintegrantesspb>>.

<sup>2</sup> Exclusivamente para realizar recolhimentos relativos às suas atividades típicas.

<sup>3</sup> Para fins de simplificação, doravante neste documento a expressão “PSP do recebedor” contemplará também a STN e os órgãos do governo federal que poderão emitir QR Codes no âmbito do Pix.

<sup>4</sup> O *backend* do PSP do pagador é o servidor do PSP do pagador que está conectado ao Sistema de Pagamentos Instantâneos (SPI) via interface ICOM na Rede do Sistema Financeiro Nacional. O aplicativo do PSP do pagador está conectado ao SPI indiretamente por meio desse servidor.

<sup>5</sup> Disponível em <<https://www.emvco.com/terms-of-use/?u=/wp-content/uploads/documents/EMVCo-Merchant-Presented-QR-Specification-v1-1.pdf>>.

<sup>6</sup> Disponível em <<https://www.emvco.com/terms-of-use/?u=/wp-content/uploads/documents/EMVCo-Consumer-Presented-QR-Specification-v1-1.pdf>>.

<sup>7</sup> Um QR Code estático pode potencialmente ser gerado com uma chave inválida, mas será um QR Code inválido, de forma que nenhum leitor conseguirá processar o QR Code, porque não haverá como rotear o pagamento.

<sup>8</sup> O identificador da transação é uma informação que abre a possibilidade de conciliação para o usuário recebedor. O PSP do recebedor recebe essa informação no processo de liquidação do valor apresentado por meio do QR Code estático, por meio da mensagem pacs.002.

### 1.2.2. QR Code Dinâmico

O QR Code dinâmico dispõe de um rol de funcionalidades abrangente, tais como conciliação via identificador da transação, configuração de valor e de campos livres estruturados<sup>9</sup>. O QR Code dinâmico também deve ser configurado para apresentar uma chave do DICT.

A característica que define o QR Code dinâmico é sua flexibilidade. O QR Code dinâmico, em sua estrutura interna, é configurado com uma URL que é acessada no momento de sua leitura. Essa funcionalidade abre diversas possibilidades de uso, dado que as informações trazidas pela URL podem variar em função de diversos parâmetros.

A URL também cumpre o papel de reduzir a quantidade de dados codificados diretamente na imagem<sup>10</sup>. O QR Code dinâmico contém somente as informações básicas do usuário receptor. O restante das informações é obtido em um *webservice* do PSP do receptor, com base nessa URL.

### 1.3. Definições gerais em relação ao BR Code

Conforme especificado no Manual do BR Code, o Pix precisa definir seu GUI (identificador único do arranjo) para ser utilizado ao longo dos IDs raiz 26-51:

GUI do PIX	Valor	Tamanho
GUI - <i>Globally Unique Identifier</i>	<b>br.gov.bcb.pix</b> <sup>11</sup>	<b>14 caracteres</b>

O Manual do BR Code apresenta a possibilidade de abrigar informações específicas do arranjo na faixa 26..51 e, como complemento, na faixa 80..99. A tabela abaixo apresenta a estrutura do Pix dentro dos campos extensíveis do BR Code.

ID	Nome EMV	Tamanho	Uso <sup>12</sup>	Descrição				
26..51 <sup>13</sup>	<i>Merchant Account Information</i>	23..99	M	"26" – indica arranjo específico; "00" (GUI) obrigatório:				
				ID	Nome	Tam	Uso	Descrição
				00	GUI	14	M	br.gov.bcb.pix
				01..99	conforme PIX			
80..99 <sup>14</sup>	<i>Unreserved Templates</i>	23..99	O	ID	Nome	Tam	Uso	Descrição
				00	GUI	14	M	br.gov.bcb.pix
				01..99	conforme PIX			

O detalhamento dos dados dos objetos tem interpretação específica definida nas seções a seguir, conforme os casos QR Code estático ou dinâmico, quando o GUI corresponder a **br.gov.bcb.pix**.

<sup>9</sup> O PSP do receptor pode configurar uma lista estruturada de campos livres. Essa funcionalidade possibilita uma apresentação de maior quantidade e qualidade de informações para o pagador no momento da confirmação de um pagamento.

<sup>10</sup> Um QR Code com muitas informações pode dificultar ou inviabilizar a decodificação dos dados a partir da imagem, que se torna muito densa.

<sup>11</sup> O GUI – DNS reverso – é *case insensitive*.

<sup>12</sup> M – Mandatório; O – opcional.

<sup>13</sup> O ID pode ser qualquer um dos identificadores dentro da faixa 26-51.

<sup>14</sup> O ID pode ser qualquer um dos identificadores dentro da faixa 80-99.



É importante ressaltar que o padrão BR Code possui outros campos nativos e opcionais, que podem ser utilizados na iniciação de pagamento. As possibilidades de mapeamento desses campos nativos EMV®, além dos campos específicos do arranjo proposto, para os dados de iniciação da ordem de pagamento que será emitida pelo PSP do pagador, são apresentadas em seção específica neste Manual, em que se descreve o mapeamento dos dados de pagamento para mensagens ISO 20022.

#### **1.4. Formatação das chaves do DICT no BR Code**

A regra para formatação das chaves Pix no BR Code com trilho Pix segue estritamente as regras definidas no Manual Operacional do DICT.

##### *1.4.1. E-mail*

O e-mail será codificado no seguinte formato:

e-mail: fulano\_da\_silva.recebedor@example.com

##### *1.4.2. CPF ou CNPJ*

O CPF e o CNPJ serão codificados no seguinte formato:

CPF: 12345678900

CNPJ: 00038166000105

##### *1.4.3. Número de telefone celular*

O telefone será codificado seguindo o formato internacional:

+5561912345678

Em que:

+55: código do país.

61: código do território ou estado.

912345678: número do telefone celular.

##### *1.4.4. Chave aleatória*

A chave aleatória será codificada juntamente com a pontuação, como segue:

123e4567-e12b-12d1-a456-426655440000



## 1.5. Iniciação do Pix via QR Code Estático

O QR Code estático no Pix conterá o seguinte conjunto de informações:

QR Code Estático		
#	Campo	Tipo
1	Chave Pix	Obrigatório
2	Valor	BR Code ID54 <sup>15</sup>
3	Conjunto livre de caracteres, com limite de tamanho (infoAdicional)	Opcional
4	Identificador da transação "TransactionIdentification <TxId>"	BR Code ID62-05 <sup>16</sup>

O mapeamento para o QR Code gerado pelo recebedor, segundo o manual do BR Code, utiliza os campos a seguir:

ID	Merchant Account Information				
26..51	ID	Nome	Tamanho	Uso	Descrição
	00	GUI	14	M	br.gov.bcb.pix
	01	chave	01..77 <sup>17</sup>	M	chave Pix
	02	infoAdicional <sup>18</sup>	01..72 <sup>19</sup>	O	conjunto livre de caracteres com limite de tamanho.

### 1.5.1. Campos Chave e InfoAdicional

O campo *chave* é limitado pelo tamanho máximo do *template* 26-51, conforme padronizado pela especificação EMV® para QR Codes: 99 caracteres. Mais especificamente, 77 caracteres quando o campo *infoAdicional* não for utilizado. Não é possível que os campos *chave* e *infoAdicional* cheguem simultaneamente a seus tamanhos máximos potenciais. Um exemplo ilustra melhor a situação. Supondo uma chave de tamanho 9, temos:

- ID + tamanho da GUI: 4
- ID + tamanho da chave: 4
- Tamanho da GUI: 14
- Tamanho da chave: 9
- $4 + 4 + 14 + 9 = 31$ .

Sobram, no exemplo acima, portanto, 68 caracteres para o *campo* texto livre ( $99 - 31 = 68$ ), incluindo o ID '02' e tamanho (2 caracteres)<sup>20</sup>. Se a chave fosse maior, menor seria o espaço destinado ao texto livre. Em outras palavras, o texto livre e a chave disputam o espaço de 99 caracteres do ID raiz (na faixa 26-51).

<sup>15</sup> Transaction Amount.

<sup>16</sup> Additional Data Field – Reference Label, sempre presente em um BRCode.

<sup>17</sup> Máximo Chave:  $99 - 8$  (ID e tamanho de GUI e Chave) – 14 (tamanho de GUI): 77

<sup>18</sup> Esse item será mostrado para o pagador no momento do rastreamento do QR. Pode servir, imagina-se, como uma pequena "descrição" do item. O limite de caracteres é um fator limitador que deve ser levado em consideração.

<sup>19</sup> Máx. Referência:  $99 - 27$  ( GUI + Chave Mínima + ID e tamanho do info adicional): 72

<sup>20</sup> Portanto, 64 caracteres de texto livre, no máximo, neste exemplo.

#### 1.5.2. Identificador de transação: txid no QR Code estático

O objeto primitivo EMV 62-05 *Reference Label*, conforme especificado no manual do BR Code, é **limitado a 25 caracteres** e, quando em efeito<sup>21</sup>, deve ser utilizado para conciliar pagamentos. Trata-se de um identificador de transação que deve ser retransmitido intacto pelo PSP do pagador ao gerar a ordem de pagamento. Essa informação permitirá ao recebedor identificar e correlacionar a transferência, quando recebida, com a apresentação das instruções ao pagador.

Os caracteres permitidos no contexto do Pix para o campo txid (EMV 62-05) são:

- Letras minúsculas, de 'a' a 'z'
- Letras maiúsculas, de 'A' a 'Z'
- Dígitos decimais, de '0' a '9'

#### 1.5.3. Mapeamento pacs.008

O mapeamento entre os dados de iniciação de pagamento apresentados pelo PSP do recebedor via QR Code (sejam campos nativos EMV/BR Code ou campos específicos do Pix) e as ordens de pagamento que devem ser geradas pelo PSP do pagador após a confirmação do usuário pagador são apresentados em seção específica deste Manual.

#### 1.5.4. Exemplo de QR Code estático

No exemplo abaixo, define-se um *Merchant Name* (nome do recebedor) fictício, obrigatório no contexto EMV®/BR Code. O valor da transação não é fornecido. Portanto, ele deverá ser solicitado ao pagador. O campo BR Code 62-05 (ID da transação) não é utilizado. O campo *3-infoAdicionais* (campo livre) não é utilizado. O QR Code foi gerado com uma chave aleatória.

Nome do Recebedor	: Fulano de Tal <sup>22</sup>
Chave Pix	: 123e4567-e12b-12d1-a456-426655440000
Valor	: não informado

---

<sup>21</sup> Conforme EMV QRCPS-MPM QR Codes for Payment Systems – Merchant Presented Mode, seção 4.8.1.2: “If present, the content of the data object value for IDs “01” to “08” shall be either “\*\*\*\*” or a value defined by the merchant. The presence of “\*\*\*\*” indicates that the mobile application is responsible for obtaining the necessary information”. Conclui-se que, se o gerador do QR optar por não utilizar um transactionID, o valor ‘\*\*\*\*’ deverá ser usado para indicar essa escolha.

<sup>22</sup> O nome a apresentar ao pagador será necessariamente o nome retornado na consulta ao DICT. O *MerchantName* será ignorado pelo pagador.

ID	Nome BR Code	Tam	Valor			
00	<i>Payload Format Indicator</i>	02	01			
26	<i>Merchant Account Information</i> <sup>23</sup>	58	ID	Nome	Tam	Valor
			00	GUI	14	br.gov.bcb.pix
			01	chave	36	123e4567-e12b-12d1-a456-426655440000
52	<i>Merchant Category Code</i>	04	0000 (não informado)			
53	<i>Transaction Currency</i>	03	986 (R\$)			
58	<i>Country Code</i>	02	BR			
59	<i>Merchant Name</i>	13	Fulano de Tal			
60	<i>Merchant City</i>	08	BRASILIA			
62	<i>Additional Data Field Template</i>	07	ID	Nome	Tam	Valor
			05	txid	03	***
63	<i>CRC16</i> <sup>24</sup>	04	0x1D3D – incluindo “6304” (ID 63 e tamanho 04)			

A sequência de caracteres correspondente ao *payload* do QR Code no padrão BR Code, grifada na tabela, fica evidenciada abaixo, com quebras de linha adicionais para favorecer a compreensão:

```

000201
2658
  0014br.gov.bcb.pix
  0136123e4567-e12b-12d1-a456-426655440000
52040000
5303986
5802BR
5913Fulano de Tal
6008BRASILIA
6207
  0503***
63041D3D

```

O respectivo QR Code estático está abaixo:

<sup>23</sup> Um ou mais campos *Merchant Account Information* (IDs 02 a 51) podem estar presentes no QR Code; identificando os diferentes arranjos de pagamento que podem ser utilizados.

<sup>24</sup> Polinômio 0x1021, C.I. 0xFFFF. A ordem dos objetos modifica o CRC.



00020126580014br.gov.bcb.pix0136123e4567-e12b-12d1-a456-426655440000  
5204000053039865802BR5913Fulano de Tal6008BRASILIA62070503\*\*\*63041D3D

#### 1.5.5. O QR Code Estático na API Pix

O QR estático pode ser criado diretamente pela automação do usuário recebedor, ou por algum outro elemento de software em função dos parâmetros: valor, txid, chave Pix, infoAdicional. A criação<sup>25</sup> do QR Code estático **NÃO** é uma funcionalidade que faz parte da API Pix.

Apesar de a criação do QR Code estático não fazer parte das funcionalidades da API Pix, os Pix recebidos que tenham sido iniciados via QR Codes estáticos que estejam associados a um txid podem ser consultados via API Pix. Maiores detalhes sobre essa funcionalidade podem ser encontrados no repositório oficial da API Pix no perfil do BCB no github: <https://github.com/bacen/pix-api>.

## 1.6. Iniciação do Pix via QR Code Dinâmico

### 1.6.1. A API Pix

O QR Code dinâmico no Pix deve ser gerado pelo usuário recebedor por meio da API Pix, exceto no fluxo de geração de QR dinâmico via app mobile do PSP<sup>26</sup>. A API Pix é uma API padronizada pelo BCB com o objetivo de facilitar o processo de integração ao arranjo Pix por parte das soluções de automação, ampliar a concorrência no setor e possibilitar menores custos aos usuários finais.

A versão vigente da API Pix contempla a especificação das funcionalidades e campos relativos aos casos de negócio focados em recebimento de pagamentos imediatos, a exemplo de pontos de venda em

---

<sup>25</sup> Ou, em outras palavras, a montagem da *string* específica que representa o QR Code, no Padrão BR Code, que segue o EMV para QR codes *Merchant presented Mode*, *string* esta que pode ser “renderizada” ou transformada visualmente como um QR code, está fora do escopo definido pela API Pix.

<sup>26</sup> A API específica utilizada pelo aplicativo do PSP do recebedor para interagir com o *backend* do PSP do recebedor encontra-se fora do escopo deste documento.

lojas físicas e de soluções para comércio eletrônico, ou com data de vencimento específica, incorporando ainda os cálculos referentes a descontos e abatimentos, juros e multas, quando aplicáveis. A API Pix continuará evoluindo para agregar novas funcionalidades.

A especificação e detalhamento da API Pix está disponível nos seguintes documentos:

- Anexo I “API Pix – Definições de Negócio”;
- Anexo II “API Pix: Especificação Técnica”;
- Anexo III “Cobranças para pagamentos com vencimento: Criação, alteração e cálculo”; e
- “**Especificação Técnica Detalhada**”, apresentada no formato OpenAPI 3.0, disponível em <https://github.com/bacen/pix-api>
- 

#### 1.6.2. O QR Code dinâmico

O QR Code dinâmico conterá as seguintes informações:

QR Code Dinâmico		
#	Campo	Tipo
1	Valor	BR Code ID54 <sup>27</sup>
2	Identificador da transação “TransactionIdentification <TxId>”	BR Code ID62-05 <sup>28</sup>
3	Link URL	Obrigatório

O campo *Link URL* representa uma URL que será utilizada para recuperação dos dados que fazem parte do pagamento. O formato dessa URL, bem como os demais detalhes sobre segurança relacionados ao QR Code, está detalhado no Manual de Segurança do SFN.

Os campos *Valor* e *Identificador da Transação (txid)* não devem ser preenchidos no QR Code Dinâmico. Se preenchidos, seu conteúdo deve ser ignorado, prevalecendo sempre os campos obtidos através da URL (*payload JSON*).

Esses campos podem ter os mesmos valores do *payload JSON*<sup>29</sup>, quando possível<sup>30</sup>.

---

<sup>27</sup> Transaction Amount

<sup>28</sup> Additional Data Field – Reference Label

<sup>29</sup> A consistência de campos entre múltiplos arranjos de pagamento com BR Code está fora do escopo desta especificação.

<sup>30</sup> No Pix, como um mesmo BR Code dinâmico pode ser reutilizado para vários pagamentos, Valor e txid - obtidos via URL - podem mudar a cada transação. Assim, esses campos BR Code devem ser ignorados pelo pagador em qualquer caso de uso de QR Code dinâmico.

O mapeamento do QR Code dinâmico para o padrão BR Code utiliza os campos a seguir:

ID	Merchant Account Information				
26..51	ID	Nome	Tam	Uso	Descrição
	00	GUI	14	M	br.gov.bcb.pix
	25	3-URL	01..77	M	Link para payload JSON

### 1.6.3. A URL hospedada no PSP do recebedor: "Location"

A URL presente no QR Code dinâmico, também chamada de "location", não deve incluir prefixo de protocolo. O acesso deverá ser realizado após<sup>31</sup> validações<sup>32</sup>, incluindo o domínio válido autorizado pelo PSP do recebedor para geração de QR Codes dinâmicos, exclusivamente via HTTPS<sup>33</sup>.

Além disto, a URL permitirá diferenciar os QR Codes dinâmicos associados a cobranças para pagamento imediato daqueles associados a pagamentos com vencimento, conforme exposto a seguir, observando-se a estrutura do fragmento "pixEndpoint".

Respeitadas as regras de formação de URL<sup>34</sup>, a URL terá esse layout:

**{fdqnPspRecebedor}/{pixEndpoint}/{pixUrlAccessToken}**

Onde:

- **fdqnPspRecebedor:** Full Qualified Domain Name do Provedor de Serviços de Pagamento da ponta Recebedora.

**pixEndpoint:** será utilizado pelo PSP recebedor para organizar suas URLs, definir a versão da API Pix e definir o tipo de cobrança que está sendo apresentada. Está estruturado da seguinte maneira:

- **{endpointOpcional}/{fragmentoTipoCobrança}**

Onde:

**endpointOpcional:** fragmento opcionalmente utilizado pelo PSP recebedor para fins de organização de URLs em um domínio. Por exemplo "/pix/123/". Pode ser vazio.

**fragmentoTipoCobrança:** define o tipo de cobrança. Na presente versão, diferencia entre cobrança para pagamento imediato ou cobrança para pagamento com vencimento. Quando vazio, significa que a cobrança é para pagamento imediato. Se apresentado o valor "/cobv/", significa que se trata de uma cobrança com vencimento.

- **pixUrlAccessToken:** fragmento que incorpora uma característica de segurança à *location*, tornando-a difícil de ser pressuposta<sup>35</sup>.

<sup>31</sup> A URL obtida no campo #3 deve ser completamente avaliada quanto à presença de caracteres inválidos e suas partes constituintes (prefixo, domínio, aplicação e demais fragmentos) antes de qualquer tentativa de acesso.

<sup>32</sup> Todos os campos da estrutura JSON associada ao acesso à URL, incluindo os opcionais, devem estar *nulos* e só devem ser preenchidos depois que o acesso for realizado e todas as verificações de segurança forem positivas.

<sup>33</sup> Restrições adicionais de segurança e outros detalhes estão definidos do Manual de Segurança do SFN.

<sup>34</sup> Ver <<https://www.ietf.org/rfc/rfc3986.txt>>.

<sup>35</sup> Em outras palavras, torna a *location* uma "URL de capacidade". Maiores informações sobre esta questão encontram-se no Manual de Segurança do SFN.

O tamanho máximo da URL completa (sem o prefixo de protocolo) é de 77 caracteres.

Maiores informações a respeito de segurança e demais detalhes podem ser encontrados diretamente no Manual de Segurança do SFN.

#### 1.6.4. URL do QR Code dinâmico

Qualquer PSP pode hospedar URLs de QR Codes dinâmicos em seus respectivos domínios.

O PSP que atua como participante responsável ou liquidante de outro participante do Pix pode ofertar ao PSP contratante o serviço de hospedagem de *payloads* de QR Codes dinâmicos.

#### 1.6.5. Sobre cobranças concluídas

A critério do PSP receptor, *payloads* que representem cobranças já concluídas, expiradas ou removidas podem retornar um http status que apresente semântica adequada<sup>36</sup>.

#### 1.6.6. O payload JSON

O *payload JSON* é o conteúdo recuperado a partir da chamada à URL, lida a partir do QR Code dinâmico (ou a partir do “Pix Copia e Cola”) e representa uma cobrança. A estrutura do payload JSON varia de acordo com o tipo de cobrança associada ao QR Code, seja ela para pagamentos imediatos ou para pagamentos com vencimento.

No caso da cobrança para **pagamento imediato**, o PSP do pagador irá acessar a URL, não havendo necessidade de envio de parâmetros. O PSP do receptor devolve diretamente no payload os dados relativos à cobrança para pagamento.

No caso da cobrança referente a **pagamento com vencimento**, o PSP do pagador irá acessar a URL e enviar a informação do código do município<sup>37</sup> (*codMun*) do usuário pagador e a Data de Pagamento Pretendida (*DEPP*). Estas informações são importantes para que o PSP do receptor possa calcular corretamente o valor a ser pago, considerando feriados estaduais e municipais que possam impactar o valor da cobrança. A informação referente ao código do município do usuário pagador poderá não ser enviada somente nas hipóteses em que, em decorrência de regulamentação específica relativa ao tipo de conta do usuário final, o PSP do pagador não possuir esta informação.

Caso a informação referente ao município não seja enviada, o PSP do receptor assumirá que não existem feriados estaduais e municipais no período em questão.

Caso a informação referente à Data de Pagamento Pretendida (*DEPP*) não seja enviada, o PSP do receptor procederá da seguinte forma para o cálculo da cobrança:

---

<sup>36</sup> Por exemplo, o PSP receptor pode optar por retornar o HTTP status 410 ‘gone’ ou ainda 404 ‘not found’.

<sup>37</sup> Esta informação corresponde ao município que consta na informação cadastral de endereço do usuário pagador.



- Cobrança não está vencida (data da consulta é menor ou igual à data de vencimento da cobrança): PSP do recebedor deve assumir a data de vencimento como a Data de Pagamento Pretendida; ou
- Cobrança está vencida (data da consulta é posterior à data de vencimento da cobrança): PSP do recebedor deve assumir a data da consulta como a Data de Pagamento Pretendida (DPP).

Para mais detalhes técnicos, consultar a documentação da API Pix<sup>38</sup>.

O aplicativo do PSP do pagador utilizará as informações contidas no *payload* JSON para apresentar os elementos da cobrança ao usuário pagador.

#### 1.6.6.1. Payload para cobrança para pagamentos imediatos

A estrutura do *payload* JSON para cobranças para pagamento imediato é a seguinte:

#	Campo	Mult.	Nome Campo JSON	Tipo	OU
1	Revisão da cobrança	[1..1]	revisao	Number	
2.1	Timestamp de criação da cobrança associada ao QR Code	[1..1]	calendario.criacao	String	
2.2	Timestamp de apresentação da cobrança associada ao QR Code	[1..1]	calendario.apresentacao	String	
2.3	Expiração da cobrança associada ao QR Code em segundos	[0..1]	calendario.expiracao	Number	
3.1	CPF do usuário devedor	[0..1]	devedor.cpf	String	OU( )
3.2	CNPJ do usuário devedor	[0..1]	devedor.cnpj	String	
3.3	Nome do usuário devedor	[0..1]	devedor.nome	String	
4.1	Valor original do documento	[1..1]	valor.original	String	
4.2	Modalidade de alteração de valor	[0..1]	valor.modalidadeAlteracao	Number	
5	Chave Pix do recebedor	[1..1]	chave	String	
6	Identificador da transação	[1..1]	txid	String	
7	Solicitação ao Pagador	[0..1]	solicitacaoPagador	String	
8	Conjunto livre de caracteres, com limite de tamanho	[0..1]	infoAdicionais	Array[InfoAdicional]	
9	Assinatura	[1..1]	–	<JWS Signature> <sup>39</sup>	
10	Situação da cobrança	[1..1]	status	String	

A seguir, apresenta-se uma breve explanação sobre os principais aspectos dos campos listados no payload acima. Para maiores detalhes técnicos, a API Pix é a referência indicada<sup>40</sup>.

#### • Revisão

O tipo do campo **revisao** é um número, começando em zero e que varia em acréscimos de 1.

O campo **revisao** adiciona rastreabilidade ao *payload*. Uma vez que se recomenda que o *payload* assinado seja armazenado pelo PSP do pagador em seus registros, fica facilitada a comunicação entre

<sup>38</sup> Disponível em <<https://github.com/bacen/pix-api>>.

<sup>39</sup> Detalhes de segurança estão definidos no Manual de Segurança do SFN.

<sup>40</sup> Disponível em <<https://github.com/bacen/pix-api>>

PSPs acerca de qual *payload* especificamente está se tratando, no contexto de resolução de possíveis problemas.

Adicionalmente, via API Pix<sup>41</sup>, é possível acessar as revisões anteriores da cobrança.

O campo **revisao** deve ser incrementado somente quando forem realizadas alterações na cobrança, via endpoints **PATCH** ou **PUT**, conforme especificados na API Pix.

- **Calendário**

Os campos aninhados sob o identificador **calendario** organizam informações a respeito de controle de tempo da cobrança. Os campos *criacao*, *expiracao* e *apresentacao*, definem elementos temporais do *payload*. Os campos *criacao* e *apresentacao* (*timestamps*) devem usar como base o tempo conforme entendido pelo PSP do recebedor. Todos os *timestamps* devem respeitar UTC<sup>42</sup>. O campo *expiração* é um inteiro apresentado em segundos.

Expostas estas explicações, segue uma descrição detalhada de cada campo do objeto **calendário**:

- **calendario.criacao**: [obrigatório] *timestamp* que indica o momento em que foi criada a cobrança. Respeita o formato definido na **RFC 3339**.
- **calendario.apresentacao**: [obrigatório] *timestamp* que indica o momento em que o *payload* JSON que representa a cobrança foi recuperado. Ou seja, idealmente, é o momento em que o usuário realizou a captura do QR Code para verificar os dados de pagamento. Respeita o formato definido na RFC 3339.
- **calendario.expiracao**: [opcional – *default* 86400] *duração* que indica limite, com granularidade de segundos, para que o pagamento da cobrança possa ser realizado, a partir da data-hora de criação. Se não for informado, assume-se a duração de 86400 segundos, que corresponde a 24 horas. Exemplo: 3600 (indica validade de 1 hora).

- **Devedor**

Os campos aninhados sob o objeto **devedor** são opcionais na cobrança para pagamentos imediatos e identificam a pessoa ou a instituição a quem a cobrança está endereçada. Não identifica, necessariamente, quem irá efetivamente realizar o pagamento. Em outras palavras, o CPF/CNPJ do devedor não é necessariamente o mesmo do CPF/CNPJ do pagador.

Apresenta-se, a seguir, a descrição detalhada de cada campo do objeto **devedor**:

- **devedor.cpf**: [opcional] determina o CPF do devedor.
- **devedor.cnpj**: [opcional] determina o CNPJ do devedor.
- **devedor.nome**: [opcional] nome da instituição ou pessoa a quem a cobrança está endereçada. O preenchimento do campo **devedor.nome** é obrigatório se o campo **devedor.cpf** ou o campo **devedor.cnpj** estiver preenchido.

---

<sup>41</sup> Disponível em <<https://github.com/bacen/pix-api>>

<sup>42</sup> UTC: *Coordinated Universal Time*. Mais detalhes em <[https://en.wikipedia.org/wiki/Coordinated\\_Universal\\_Time](https://en.wikipedia.org/wiki/Coordinated_Universal_Time)>.

- **Valor**

O objeto **valor** organiza os elementos que compõem o valor da cobrança. No caso da cobrança para pagamento imediato, são os seguintes os campos do objeto valor:

- **valor.original**: [obrigatório] valor do documento para cobrança para pagamento imediato. Deve apresentar valores maiores do que zero, exceto no caso de o campo **modalidadeAlteracao** apresentar valor 1.
- **valor.modalidadeAlteracao**: [opcional] indica a modalidade de alteração aplicada para a cobrança em questão. Se ausente, assume-se que a modalidade aplicada é a modalidade 0, que significa que não se pode alterar o valor da cobrança. Caso o campo **modalidadeAlteracao** apresente o valor 1, então o usuário pagador poderá alterar o valor da cobrança. Ainda no caso de o campo **modalidadeAlteracao** apresentar o valor 1, o campo **valor.original** pode funcionar como um “valor sugerido”, de forma que, caso o “valor sugerido” atribuído seja zero, o usuário pagador será obrigado a digitar um valor válido, ou seja, um valor maior que do que zero<sup>43</sup>.

Todos os campos que indicam valores monetários obedecem ao formato do ID 54 da especificação EMV/BR Code para QR Codes. O separador decimal é o caractere **ponto**. Não é aplicável utilizar separador de milhar. Exemplos aderentes ao padrão: “0.10”, “.10”, “1.”, “1”, “123.99”, “123456789.23”.

- **Chave**

O campo **chave**, obrigatório, determina a chave Pix registrada no DICT que será utilizada para a cobrança. Essa chave será lida pelo aplicativo do PSP do pagador para consulta ao DICT, que retornará a informação que identificará o recebedor da cobrança.

- **txid**

O campo **txid**, obrigatório, determina o identificador da transação. O objetivo desse campo é ser um elemento que possibilite ao PSP do recebedor apresentar ao usuário recebedor a funcionalidade de conciliação de pagamentos (possibilitando associar a transação Pix à cobrança correlata).

Na pacs.008, é referenciado como TransactionIdentification <TxId> ou idConciliacaoRecebedor. O campo **txid**, no caso do QR Dinâmico, deve ter, no mínimo, **26 caracteres e, no máximo, 35 caracteres**.

Em termos de fluxo de funcionamento, o **txid** é lido pelo aplicativo do PSP do pagador e, depois de confirmado o pagamento, se o usuário recebedor tem conta em outro PSP, é enviado para o SPI via pacs.008. Por sua vez, o SPI envia uma pacs.008 ao PSP do recebedor, contendo, além de todas as informações usuais do pagamento, o **txid**. Ao receber uma mensagem com o campo **txid**

---

<sup>43</sup> O PSP do pagador não deve permitir a confirmação de um pagamento cujo valor seja zero.

preenchido, o PSP do recebedor está apto a se comunicar com o usuário recebedor, informando que um pagamento específico foi liquidado.

O **txid**, no contexto da API Pix, é criado pelo usuário recebedor e encontra-se sob sua responsabilidade<sup>44</sup>. O **txid**, no contexto de representação de uma cobrança, seja ela para pagamento imediato ou com vencimento, é único por CPF/CNPJ do usuário recebedor e PSP. Cabe ao PSP do recebedor validar essa regra na API Pix.

Mais informações sobre o campo **txid** podem ser encontradas no Anexo I (“API Pix: Conceitos de Negócio”, seção 4).

- **Solicitação ao Pagador**

O campo **solicitacaoPagador**, opcional, determina um texto a ser apresentado ao pagador para que este possa digitar uma informação correlata, em formato livre, a ser enviada ao usuário recebedor. Esta informação correlata<sup>45</sup> será preenchida, na pacs.008, pelo PSP do pagador, no campo *RemittanceInformation* <RmtInf>. O tamanho do campo <RmtInf> na pacs.008 está limitado a 140 caracteres.

- **Informações Adicionais**

O campo **infoAdicionais**, se estiver presente, se refere a uma lista em que cada elemento deve utilizar o esquema abaixo:

Subcampo JSON (infoAdicionais)	Presença	Tipo JSON	Propósito
nome	Obrigatório	String	Nome do campo
valor	Obrigatório	String	Dados do campo

Os limites relativos ao tamanho de cada campo e à quantidade de elementos da lista estão tratados na especificação da API Pix<sup>46</sup>.

Cada respectiva informação adicional contida na lista (*nome* e *valor*) deve ser apresentada ao pagador. Exemplo: campo *infoAdicionais* no *JWSPayload*:

```
infoAdicionais[0].nome: "campo1"
infoAdicionais[0].valor: "informação adicional 01"
infoAdicionais[1].nome: "campo2"
infoAdicionais[1].valor: "informação adicional 02"...
```

- **Assinatura**

<sup>44</sup> No caso das cobranças para pagamento imediato, a criação do txid poderá ser delegada ao PSP do pagador.

<sup>45</sup> Importante destacar que é a informação correlata digitada livremente pelo pagador que é enviada na pacs.008, e não o texto apresentado no campo *solicitacaoPagador*.

<sup>46</sup> Disponível em <<https://github.com/bacen/pix-api>>.

Todos os campos, exceto o campo **assinatura**, compõem um *payload* JSON assinado. Maiores detalhes sobre a assinatura desse *payload* JSON podem ser encontrados no Manual de Segurança do SFN.

- **Status**

O campo **status** representa a situação da cobrança para pagamentos imediatos, podendo assumir os seguintes estados: “ATIVA”, “CONCLUIDA”, “REMOVIDA\_PELo\_USUARIO\_RECEBEDOR”, “REMOVIDA\_PELo\_PSP”

#### 1.6.6.2. Payload para cobrança para pagamentos com vencimento

Quando o PSP do pagador, por meio da URL encaminhada no QR Code dinâmico, também chamada de “location”, ou no “Pix Copia e Cola”, acessa o payload de uma cobrança para **pagamentos com vencimento**, ele dá início ao cálculo do valor da cobrança pelo PSP do recebedor, que enviará os campos do payload devidamente preenchidos.

É importante notar que se a obrigação não estiver vencida e o PSP do pagador não enviar a data desejada, o PSP do recebedor irá assumir a data de vencimento como a data a ser utilizada no cálculo do valor a ser pago.

Por outro lado, se a obrigação já estiver vencida e o PSP do pagador não enviar a data desejada, o PSP do recebedor irá assumir a data da consulta como aquela a ser utilizada no cálculo do valor a ser pago.

Vale lembrar que cabe ao PSP do pagador enviar o código do município do pagador, a fim de que o PSP do recebedor possa calcular corretamente o valor a ser pago, considerando feriados estaduais e municipais aplicáveis. Nas hipóteses em que o PSP do pagador não possuir essa informação e, portanto, não enviá-la (naqueles casos em que a regulação não exige que o PSP do pagador tenha, em seu cadastro, o endereço do usuário pagador), o PSP do recebedor assumirá que não existem feriados estaduais e municipais no período em questão. Para mais detalhes técnicos, consultar a documentação da API Pix<sup>47</sup>.

A estrutura do *payload* JSON para cobranças para pagamentos com vencimento é:

#	Campo	Mult.	Nome Campo JSON	Tipo	OU
1	Revisão da cobrança	[1..1]	revisao	Number	
2.1	Timestamp de criação da cobrança associada ao QR Code	[1..1]	calendario.criacao	String	
2.2	Timestamp de apresentação da cobrança associada ao QR Code	[1..1]	calendario.apresentacao	String	
2.3	Data de vencimento do pagamento	[1..1]	calendario.dataDeVencimento	String	
2.4	Validade da cobrança após vencimento	[1..1]	Calendario.validadeAposVencimento	Number	

<sup>47</sup> Disponível em <<https://github.com/bacen/pix-api>>.

3.1	CPF do usuário devedor	[0..1]	devedor.cpf	String	OU(
3.2	CNPJ do usuário devedor	[0..1]	devedor.cnpj	String	)
3.3	Nome do usuário devedor	[1..1]	devedor.nome	String	
4.1	Valor original do documento	[0..1]	valor.original	String	
4.2	Valor do abatimento	[0..1]	valor.abatimento	String	
4.3	Valor dos descontos	[0..1]	valor.desconto	String	
4.4	Valor dos juros	[0..1]	valor.juros	String	
4.5	Valor da multa	[0..1]	valor.multa	String	
4.6	Valor final do documento	[1..1]	valor.final	String	
5	Chave Pix do recebedor	[1..1]	chave	String	
6.1	CPF do recebedor	[0..1]	recebedor.cpf	String	OU(
6.2	CNPJ do recebedor	[0..1]	recebedor.cnpj	String	)
6.3	Nome ou razão social do recebedor	[1..1]	recebedor.nome	String	
6.4	Nome de fantasia do recebedor	[0..1]	recebedor.nomeFantasia	String	
6.5	Logradouro do recebedor	[0..1]	recebedor.logradouro	String	
6.6	Cidade do recebedor	[0..1]	recebedor.cidade	String	
6.7	Unidade da federação do recebedor	[0..1]	recebedor.uf	String	
6.8	CEP do recebedor	[0..1]	recebedor.cep	String	
7	Identificador da transação	[1..1]	txid	String	
8	Solicitação ao Pagador	[0..1]	solicitacaoPagador	String	
9	Conjunto livre de caracteres, com limite de tamanho	[0..1]	infoAdicionais	Array[InfoAdicional]	
10	Assinatura	[1..1]	-	<JWS Signature> <sup>48</sup>	
11	Situação da cobrança	[1..1]	status	String	

A seguir apresenta-se uma breve explanação sobre os campos do payload de cobranças para pagamentos com vencimento que não são utilizados na cobrança para pagamentos imediatos ou cuja explicação precise ser adequada para o contexto das cobranças para pagamentos com vencimento. Para maiores detalhes técnico, a API Pix é a referência indicada<sup>49</sup>.

## • Calendário

Campos do objeto **calendário**:

- **calendario.criacao**: idem à cobrança para pagamentos imediatos
- **calendario.apresentacao**: idem à cobrança para pagamentos imediatos
- **calendario.dataDeVencimento**: [obrigatório] trata-se de uma data, no formato 'yyyy-mm-dd', segundo ISO 8601. É a data de vencimento da cobrança, **que pode ser paga em qualquer horário do dia**. Exemplo: 2020-10-19.<sup>50</sup>

<sup>48</sup> Detalhes de segurança estão definidos no Manual de Segurança do SFN.

<sup>49</sup> Disponível em <<https://github.com/bacen/pix-api>>

<sup>50</sup> Sempre que a data de vencimento cair em um fim de semana ou em um feriado para o usuário pagador, ela deve ser automaticamente prorrogada para o primeiro dia útil subsequente.

- **calendario.validadeAposVencimento: (int32)** Trata-se da quantidade de **dias corridos** após **calendario.dataDeVencimento** em que a cobrança poderá ser paga. Aplica-se o valor deste campo sobre o vencimento original da cobrança acrescentando-se o número de dias corridos nos quais a cobrança ainda poderá ser paga, após vencida<sup>51</sup>.

- **Devedor**

Diferentemente da cobrança para pagamentos imediatos, no caso de uma cobrança para **pagamentos com vencimento**, o objeto **devedor** é obrigatório. As demais observações sobre o objeto e seus campos permanecem válidas.

- **Valor**

O objeto **valor** organiza os elementos que compõem o valor da cobrança: juros, multa, desconto, abatimento, entre outros elementos correlatos.

Todos os campos que indicam valores monetários obedecem ao formato do ID 54 da especificação EMV/BR Code para QR Codes. O separador decimal é o caractere **ponto**. Não é aplicável utilizar separador de milhar. Exemplos de valores aderentes ao padrão: **"0.00"**, **"1.00"**, **"123.99"**, **"123456789.23"**.

Apresenta-se, a seguir, a descrição detalhada de cada campo do objeto **valor**:

- **valor.original:** [opcional] valor original do documento, antes de possíveis multas, juros, descontos ou abatimentos.
- **valor.abatimento:** [opcional] valor do abatimento aplicado à cobrança. Não será enviado se for igual a 0 (zero);
- **valor.desconto:** [opcional] valor do desconto aplicado à cobrança em virtude de pagamento antecipado. Não será enviado se for igual a 0 (zero);
- **valor.juros:** [opcional] valor dos juros aplicados à cobrança em atraso. Não será enviado se for igual a 0 (zero);
- **valor.multa:** [opcional] valor da multa aplicada à cobrança em atraso. Não será enviado se for igual a 0 (zero);
- **valor.final:** [obrigatório] valor final da cobrança, considerados abatimentos, desconto, juros e multa. Ressalvado o campo **original**, se todos os demais campos estiverem zerados, o App do PSP do pagador deve exibir apenas o campo **final**.

- **Recebedor**

O objeto **Recebedor** organiza as informações sobre o credor da cobrança. Apresenta-se, a seguir, a descrição detalhada de cada campo do objeto **Recebedor**:

---

<sup>51</sup> Sempre que a data de validade após o vencimento cair em um fim de semana ou em um feriado para o usuário pagador, ela deve ser automaticamente prorrogada para o primeiro dia útil subsequente.



- **recebedor.cpf**: CPF do recebedor, se pessoa natural, conforme cadastro da conta transacional associada à chave Pix.
- **recebedor.cnpj**: CNPJ do recebedor, se ele estiver inscrito no Cadastro Nacional da Pessoa Jurídica, mantido junto à Receita Federal, conforme cadastro da conta transacional associada à chave Pix. Se este campo estiver preenchido, o campo **recebedor.cpf** não deverá ser preenchido.
- **recebedor.nome**: [obrigatório] nome ou razão social do recebedor, conforme cadastro da conta transacional associada à chave Pix.
- **recebedor.nomeFantasia**: [opcional] nome fantasia, no caso de recebedor inscrito no Cadastro Nacional da Pessoa Jurídica, conforme cadastro da conta transacional associada à chave Pix.
- **recebedor.logradouro**: [obrigatório] logradouro do recebedor, conforme cadastro da conta transacional associada à chave Pix;
- **recebedor.cidade**: [obrigatório] cidade do recebedor, conforme cadastro da conta transacional associada à chave Pix;
- **recebedor.uf**: [obrigatório] unidade da federação (estado ou DF) do recebedor, conforme cadastrado na conta transacional associada à chave Pix;
- **recebedor.cep**: [obrigatório] Código de endereçamento postal (CEP) do recebedor, conforme cadastrado na conta transacional associada à chave Pix.

#### 1.6.7. Exemplo de QR Code Dinâmico

No exemplo abaixo, para uma cobrança para pagamentos imediatos, define-se um *Merchant Name* (nome do recebedor) fictício, obrigatório no contexto do BR Code. O valor da transação deve ser obtido no payload da cobrança. A URL deverá ser validada e consultada após a leitura do QR Code para apresentar ao pagador as informações completas do beneficiário e o contexto do pagamento. O campo BR Code 01 (*Point of Initiation Method*), opcional, está presente para indicar que não deve ser iniciado mais de um pagamento com este mesmo QR Code.

Nome EMV do recebedor <sup>52</sup> : Fulano de Tal  
 Valor : R\$ 123,45  
 URL do PSP do recebedor : pix.example.com/8b3da2f39a4140d1a91abd93113bd441

ID	Nome EMV	Tam	Valor			
00	Payload Format Indicator	02	01			
01	Point of Initiation Method	02	12 (não deve ser utilizado mais de uma vez) <sup>53</sup>			
26	Merchant Account Information	70	ID	Nome	Tam	Valor
			00	GUI	14	br.gov.bcb.pix

<sup>52</sup> O nome que deve ser apresentado ao pagador é o nome do usuário recebedor conforme registrado no DICT.

<sup>53</sup> Favorece melhor controle da experiência do usuário, evitando tentar a iniciação de pagamento de um QR Code que já foi processado com sucesso (por exemplo, baseado em Instituição e *Reference Label*), evitando assim acesso à URL e o envio de mensagens desnecessárias (que serão provavelmente rejeitadas).

			25	URL	48	pix.example.com/8b3da2f39a4140d1a91abd93113bd441
52	Merchant Category Code	04	0000 (não informado)			
53	Transaction Currency	03	986 (R\$)			
58	Country Code	02	BR			
59	Merchant Name	13	Fulano de Tal			
60	Merchant City	08	BRASILIA			
62	Additional Data Field	07	ID	Nome	Tam	Valor
			05	Reference Label	03	***
63	CRC16	04	0x64E4			

A sequência de caracteres correspondente ao *payload* do QR Code dinâmico no padrão BR Code gerado pelo recebedor, grifada na tabela, fica evidenciada abaixo:

```

000201
010212
2670
  0014br.gov.bcb.pix
  2548pix.example.com/8b3da2f39a4140d1a91abd93113bd441
52040000
5303986
5802BR
5913Fulano de Tal
6008BRASILIA
6207
  0503***
630464E4

```

O respectivo QR Code dinâmico está abaixo:



```

00020101021226700014br.gov.bcb.pix2548pix.example.com/8b3da2f39a4140d1a91abd93113bd44
15204000053039865802BR5913Fulano de Tal6008BRASILIA62070503***630464E4

```

Recupera-se a URL do *payload* JSON com os dados do pagamento:

**pix.example.com/8b3da2f39a4140d1a91abd93113bd441**

Verificado o domínio “example.com” como autorizado<sup>54</sup> e obtida a chave pública de assinatura (**PSPRECPUBKEY**), recupera-se o *payload* JSON:

HTTPS Request:

<https://pix.example.com/8b3da2f39a4140d1a91abd93113bd441>

HTTPS Response:

*JWSHeader.JWSPayload.JWSSignature*

Se (e somente se) **JWS-AssinaturaVálida**<sup>55</sup> (*JWSHeader, JWSPayload, JWSSignature, PSPRECPUBKEY*), procede-se ao *parsing* dos dados da estrutura JSON:

JSONPayload = base64url<sup>56</sup>-decode (*JWSPayload*).

Um exemplo de *payload* JSON pode ser encontrado na API Pix<sup>57</sup>

Exemplo de apresentação para confirmação do Pagador:

#### **PAGAMENTO - 31/03/2020 15:20:30**

**\*\* ATENÇÃO – Verifique os dados do pagamento \*\***

Valor : R\$ 123,45

Para : FULANO DE TAL EIRELI

CNPJ 00.123.456/7891-23

PSP EXAMPLE.COM

Expira em: 31/03/2020 17:00:00<sup>58</sup>

Detalhes do Pagamento: **Informação Adicional do PSP do Recebedor**  
**Confirma?**

Obtido por consulta ao DICT pela chave do  
recebedor:  
4b7b4eb3-d426-48f1-8ecf-998bda62c0a1

#### **1.6.8. O QR Code Dinâmico na API Pix**

O QR Code Dinâmico, em um contexto de integração automatizada<sup>59</sup>, não pode ser criado diretamente pelo usuário recebedor. É preciso que o usuário recebedor, em seu software de automação comercial / TEF / Gateway configure o *endpoint*<sup>60</sup> que será responsável por retornar as informações relativas à cobrança que o QR Code Dinâmico busca representar. A API Pix não monta a *string* para a renderização do QR Code nem retorna um arquivo com o QR Code gerado; desta forma, estas funcionalidades

<sup>54</sup> Conforme Manual de Segurança do SFN.

<sup>55</sup> Conforme Manual de Segurança do SFN.

<sup>56</sup> RFC 4648, disponível em <<https://tools.ietf.org/html/rfc4648>>. *base64url* corresponde à codificação base64 alterada para evitar o uso de “+”, “/” e “=”, pois esses caracteres causariam efeitos adversos em URLs.

<sup>57</sup> <https://github.com/bacen/pix-api>

<sup>58</sup> Note-se que o *payload* utiliza UTC.

<sup>59</sup> Em outras palavras, fora do cenário de criação do QR Dinâmico por meio do aplicativo do PSP recebedor.

<sup>60</sup> Uma URL que ao acessada retorna como resposta um conjunto de informações, também chamada de “location”.



poderão ser ofertadas pelo PSP do recebedor ou providas diretamente pela solução de automação comercial adotada pelo usuário recebedor ou ainda via outro elemento de software.

Para maiores informações a respeito da dinâmica de gerenciamento de cobranças, pode-se consultar a API Pix<sup>61</sup>.

### **1.7. Pix Copia e Cola**

Eventualmente, em cenários de iniciação de um Pix, ler o QR Code pode ser impraticável, como, por exemplo, navegar em uma loja online via web browser mobile, comprar em aplicativos instalados no telefone celular e enviar um QR Code via aplicativo de mensagens. Nessas situações, como o telefone celular é o dispositivo usado para ler o QR Code, não é possível utilizá-lo para ler um QR Code disponibilizado em sua própria tela. A funcionalidade “Pix Copia e Cola” foi pensada para ser uma alternativa nesses casos.

Para utilizar a funcionalidade “Pix Copia e Cola”, o usuário recebedor deve poder oferecer ao pagador uma maneira de copiar o inteiro teor da sequência de caracteres que representa o BR Code. Em outras palavras, deve ser copiada exatamente a mesma sequência de caracteres que seria lida pelo leitor de QR Code no momento da leitura do QR Code em questão. Essa funcionalidade é possível tanto para o QR Code estático quanto para o QR Code dinâmico.

---

<sup>61</sup> <https://github.com/bacen/pix-api>

## 2. Mapeamento para mensagens ISO 20022

Alguns campos de iniciação de Pix apresentados pelo PSP do recebedor e pelo ente governamental por meio de QR Code devem ser mapeados pelo PSP do pagador nas mensagens de pagamento ISO 20022, como especificado abaixo. Os demais campos do pagamento (por exemplo, o `valor.final - IntrBkSttlmAmt` - da transação), devem seguir a lógica de negócio desenhada no Catálogo de Serviços do SFN.

### 2.1. Mapeamento do QR Code estático para pacs.008

Mapeamento de QR Code estático:

ID	Nome no Arranjo Nome EMV	ISO 20022 pacs.008		
26-51	Merchant Account Information	ID	Nome no Arranjo Nome EMV	pacs.008
		01	chave	CreditorAccount <CdtrAcct> Proxy <Prxy> Identification <Id>
62	Additional Data Field	ID	Nome no Arranjo Nome EMV	pacs.008
		05	txid <sup>62</sup> Reference Label	PaymentIdentification <PmtId> TransactionIdentification <TxId>

**CdtrAcct.Prxy.Id** – A chave Pix que identifica o recebedor do pagamento sempre deve ser mapeada para esse campo.

**PmtId.TxId** – Quando diferente de '\*\*\*', deve ser retransmitido intacto pelo PSP do pagador ao gerar a ordem de pagamento.

<sup>62</sup> Quando em efeito (ou seja, diferente de '\*\*\*').

## 2.2. Mapeamento payload JSON para pacs.008

Os dados do recebedor, provenientes do *payload* JSON, que devem ser mapeados pelo PSP do pagador para a mensagem de pagamento, são definidos abaixo:

Campo	Uso	campo JSON	pacs.008
Chave Pix do recebedor	M	chave	CreditorAccount <CdtrAcct> Proxy <Prxy> Identification <Id>
Identificador da Transação	M	txid	PaymentIdentification <PmtId> TransactionIdentification <TxId>
Informações Adicionais pelo Pagador	O	(resposta ao campo <i>solicitacaoPagador</i> <sup>63</sup> )	RemittanceInformation <RmtInf>

**CdtrAcct.Prxy.Id** – A chave Pix que identifica o recebedor do pagamento sempre deve ser mapeada para esse campo.

**PmtId.TxId** – Deve ser retransmitido intacto pelo PSP do pagador. O campo presente no QR Code é ignorado, mesmo que diferente de ‘\*\*\*’, quando o QR Code for do tipo dinâmico.

**RmtInf** – O campo JSON *solicitacaoPagador*, opcional, conforme consta na API Pix<sup>64</sup>, especifica um texto a ser apresentado ao pagador para que ele possa digitar a informação correlata, em formato livre, a ser enviada ao recebedor. O campo <RmtInf> na pacs.008 é limitado a 140 caracteres.

**ATENÇÃO:** observar que o campo “txid” no JSON é expresso em minúsculas, enquanto o campo “TxId” da pacs.008 alterna maiúsculas e minúsculas.

<sup>63</sup> Destaca-se que não é o conteúdo do campo “solicitacaoPagador” que deve ser preenchido no campo <RmtInf> da pacs.008, mas sim a resposta digitada pelo usuário pagador em função de “solicitacaoPagador”. O usuário pagador é instado a preencher alguma informação decorrente do texto que conste em “solicitacaoPagador”, e é exatamente esse texto que o usuário pagador escreveu que deve ser preenchido no campo <RmtInf> da pacs.008.

<sup>64</sup> Disponível em: <<https://github.com/bacen/pix-api>>

## ANEXO I – API Pix: Conceitos de Negócio

### 1. Introdução

O presente anexo tem por objetivo apresentar os conceitos de negócio e os casos de uso envolvendo a utilização da API Pix, sendo complementado pelas especificações técnicas da API, conforme explicitado na seção 2 deste documento.

Esta versão da API Pix contempla as funcionalidades necessárias para viabilizar o recebimento de cobranças em casos de negócio focados em pagamentos imediatos, a exemplo de pontos de venda em lojas físicas e de soluções para comércio eletrônico, ou com data de vencimento específica, incorporando ainda os cálculos referentes a descontos e abatimentos, juros e multas, quando aplicáveis. A API Pix continuará evoluindo para agregar novas funcionalidades.

### 2. Documentação da API Pix

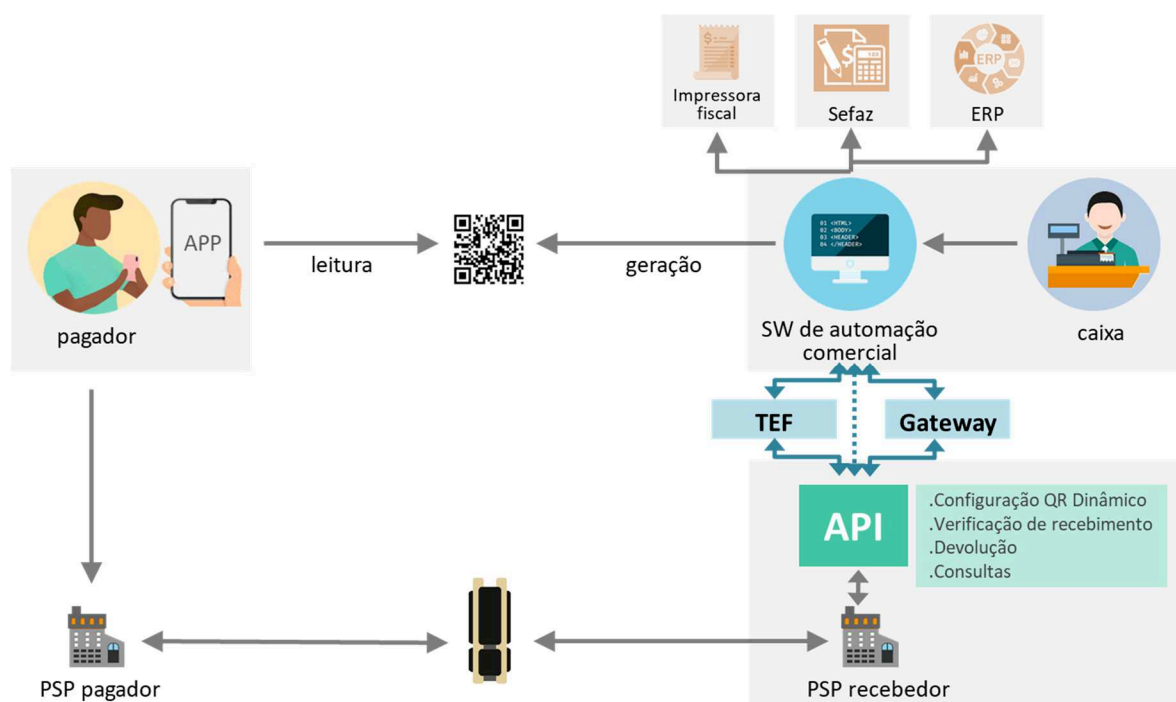
O conjunto completo de documentação da API Pix é composto pelos seguintes artefatos:

- (1) **Anexo I: API Pix: Conceitos de Negócio** – [este anexo](https://www.bcb.gov.br/estabilidadefinanceira/pagamentosinstantaneos), disponível em <https://www.bcb.gov.br/estabilidadefinanceira/pagamentosinstantaneos>
- (2) **Anexo II: API Pix: Especificação técnica** – disponível em <https://www.bcb.gov.br/estabilidadefinanceira/pagamentosinstantaneos>
- (3) **Anexo III: Cobranças para pagamentos com data de vencimento: criação, alteração e cálculo** – disponível em <https://www.bcb.gov.br/estabilidadefinanceira/pagamentosinstantaneos>
- (4) **API Pix: Especificação técnica detalhada** – apresentada no formato OpenAPI 3.0 e disponível em <https://github.com/bacen/pix-api>.

Para acessar a documentação técnica constante no item 3, deve ser recuperada a última versão disponível. Para cada versão, é gerada uma “tag” no Github contendo o seu número, no formato “major.minor.patch” (exemplo: 2.1.0).



### 3. Contexto da API Pix



**Figura 1:** A API Pix

A API Pix é o componente do arranjo que visa possibilitar que o usuário pagador ou receptor, no contexto P2B<sup>65</sup> ou B2B<sup>66</sup>, possa automatizar a interação com seu prestador de serviços de pagamento (PSP), a fim de realizar ou receber transações no âmbito do arranjo Pix.

Nesse contexto, a presente versão da API Pix busca automatizar a interação do usuário receptor com seu prestador de serviços de pagamento (PSP), a fim de gerar cobranças e confirmar o recebimento do pagamento dessas cobranças por meio do Pix. Na **figura 1**, pode-se visualizar possíveis caminhos de integração dos sistemas do usuário receptor com a API Pix do PSP.

O usuário receptor poderá, via API Pix:

- I – gerar cobranças que poderão ser pagas via QR Code pelos seus clientes;
- II – alterar dados da cobrança;
- III – remover dados da cobrança, em caso de necessidade de cancelamento;
- IV – verificar a liquidação da cobrança por meio de Pix recebidos;
- V – realizar a conciliação dos pagamentos de maneira facilitada;

<sup>65</sup> “Person to Business”, trata-se de uma interação de Pessoa para Negócio/empresa/estabelecimento.

<sup>66</sup> “Business to Business”, trata-se de interações entre negócios/empresas/estabelecimentos.

VI – suportar o processo de devolução de valores, que pode ser acionado em função, por exemplo, da devolução de uma compra.

A seguir são detalhados os aspectos gerais que dizem respeito à API Pix.

## 4. Conceitos gerais

Para os fins deste documento, as expressões e os termos relacionados são assim definidos:

**I - client\_ID:** componente de acesso a API Pix que identifica um elemento de **software cliente** do usuário. Para acessar a API, é necessário utilizar, por exemplo, o Client\_ID e um segredo, ambos fornecidos pelo PSP do usuário no processo de cadastramento. Pode existir mais de um elemento software cliente na infraestrutura do usuário e, portanto, para um usuário pode existir mais de um client\_ID.

**II - escopos:** definem as autorizações associadas a cada um dos serviços da API. Por sua vez, os Client\_IDs possuem acesso a um ou mais escopos, o que definirá quais serviços podem ser acessados por cada Client\_ID;

**III – payload JSON:** conteúdo recuperado a partir da chamada à URL, lida a partir do QR Code dinâmico e que representa uma cobrança;

**IV - PSP Pagador:** participante do Pix no qual o usuário pagador possui uma conta transacional;

**V – PSP Recebedor:** participante do Pix no qual o usuário recebedor possui uma conta transacional que será usada para recebimentos de Pix. O PSP Recebedor que quiser disponibilizar a seus clientes uma solução de integração automatizada com o arranjo Pix deve fazê-lo por meio da API Pix, seguindo as especificações de negócio e técnicas definidas pelo Banco Central do Brasil neste documento e nos outros documentos trazidos na seção 2.

**VI - transactionId (txid):** identificador da transação, na perspectiva do usuário recebedor. Esse número é gerado pelo usuário recebedor e repassado ao PSP recebedor na chamada da API Pix, no momento da criação da cobrança, a fim de identificar unicamente aquela cobrança<sup>67</sup>. Assim, o txid é utilizado pelo usuário recebedor, em seus processos de conciliação de pagamentos recebidos por meio de Pix. No caso das cobranças criadas por meio da API Pix, o PSP recebedor deve garantir que o txid seja único para um dado usuário recebedor (CPF/CNPJ).

**VII - usuário pagador:** aquele que efetua o pagamento de uma cobrança por meio do Pix.

**VIII - usuário recebedor:** pessoa natural ou jurídica que deseja receber cobranças – para pagamentos imediatos ou com vencimento – por meio do Pix e se vale da API Pix para automação dos seus processos de geração dessas cobranças e para conciliação de pagamentos recebidos por meio do Pix.

---

<sup>67</sup> Nas cobranças para pagamento imediato, é possível ao usuário recebedor delegar a geração do txid para seu PSP. Para mais detalhes, veja a seção 5.3.1

## 5. Funcionalidades da API Pix

### 5.1 Definições das entidades

A API Pix está estruturada em torno de algumas entidades de negócio, que agrupam conjuntos de atributos, conforme definido abaixo:

**I - Cobrança (/cob e /cobv):** representa cada uma das cobranças geradas por meio da API Pix, a fim de permitir que o usuário pagador efetue um pagamento identificado para o usuário recebedor. A cobrança é caracterizada por um conjunto de informações que são utilizadas para que o usuário pagador execute um pagamento por meio do Pix, geralmente, em função de acordo comercial entre o usuário pagador e o usuário recebedor, sem se confundir com o pagamento Pix em si. A cobrança se subdivide em duas espécies: cobranças para pagamento imediato e cobranças para pagamento com vencimento.

Estados da cobrança:

- a) **ATIVA:** indica que a cobrança foi gerada e pronta para ser paga;
- b) **CONCLUÍDA:** indica que a cobrança já foi paga e, por conseguinte, não pode acolher outro pagamento<sup>68</sup>;
- c) **REMOVIDO\_PELU\_USUARIO\_RECEBEDOR:** indica que o usuário recebedor solicitou a remoção da cobrança; e
- d) **REMOVIDO\_PELU\_PSP:** indica que o PSP Recebedor solicitou a remoção da cobrança.

**II - Pix (/pix):** representa um pagamento recebido por meio do arranjo de pagamentos Pix.

**III - Devolução (devolução):** representa uma solicitação de devolução de um Pix realizado, cujos fundos já se encontrem disponíveis na conta transacional do usuário recebedor.

Estados da devolução:

- a) **EM\_PROCESSAMENTO:** indica que a devolução foi solicitada, mas ainda está em processamento no SPI;
- b) **DEVOLVIDO:** indica que a devolução foi liquidada pelo SPI; e
- c) **NAO\_REALIZADO:** indica que a devolução não pode ser realizada em função de algum erro durante a liquidação (exemplo: saldo insuficiente).

**IV – Webhook(/webhook):** é um recurso técnico que permite que o PSP Recebedor informe diretamente o usuário recebedor quando um Pix associado a um txid foi creditado na sua conta transacional. Nesse caso, a lógica do processo é invertida em relação ao funcionamento padrão da API, para garantir uma melhor performance ao processo. O usuário recebedor deixa de consultar o PSP Recebedor a todo momento (*polling*) e passa a ser informado na ocorrência de uma liquidação. Somente Pix associados a um txid serão informados via a funcionalidade webhook.

**V – PayloadLocation(/loc):** é um recurso que permite ao PSP Recebedor reusar uma URL, retornando diferentes cobranças (payloads JSON) ao longo do tempo, , mas apenas uma por vez. Tipicamente, é utilizado quando o usuário recebedor precisa apresentar um QR Code impresso, mas que seja dinâmico.

---

<sup>68</sup> Importante observar que o estado CONCLUIDA refere-se à Cobrança gerada e não à obrigação associada. Dessa forma, esse estado não indica a liquidação da obrigação em si, mas apenas que aquela Cobrança não admite novos pagamentos.

## 5.2 Cardinalidade entre as entidades

- I - Uma **Cobrança** pode estar associada a um ou mais **Pix** (mesmo txid);
- II - um **Pix** pode estar associado a uma única **Cobrança**. O **Pix**, no entanto, pode existir independentemente da existência de uma **Cobrança**;
- III - um **Pix** pode ter uma ou mais **Devoluções** associadas a ele. Uma **Devolução** está sempre associada a um **Pix**.
- IV – Uma **Cobrança** somente pode estar associada a um **PayloadLocation** e, num determinado momento, o PayloadLocation só pode estar associado a uma cobrança.

## 5.3 Ciclo de vida do TransactionID (txid)

Há duas situações de uso para o campo TransactionId (txid), que envolvem regras distintas para seu preenchimento, conforme tratado a seguir.

### 5.3.1 txid no contexto das Cobranças

As cobranças para pagamentos imediatos ou com vencimento criadas por meio da API Pix são identificadas unicamente por meio de um txid. O txid pode ser enviado pelo usuário recebedor<sup>69</sup> quando da geração da cobrança, e não poderá se repetir entre cobranças distintas<sup>70</sup>, a fim de garantir a correta conciliação dos pagamentos. Alternativamente, no caso de cobranças para pagamento imediato, é possível que o usuário recebedor delegue a geração do txid para o seu PSP<sup>71</sup>.

Uma vez que seja solicitada a criação de uma cobrança por meio da API Pix, o PSP Recebedor deve assegurar que não exista repetição do txid para o mesmo usuário recebedor, seja ele enviado pelo usuário ou gerado pelo próprio PSP. Assim, o conjunto CNPJ ou CPF e txid deve ser único para um dado PSP.

O txid deve ter, no mínimo, 26 caracteres e, no máximo, 35 caracteres de tamanho. Os caracteres aceitos neste contexto são: A-Z, a-z, 0-9.

---

<sup>69</sup> A importância de que o txid seja enviado pelo usuário recebedor está relacionada à garantia da idempotência da API Pix. Dessa forma, evita-se a criação de múltiplas cobranças para uma mesma obrigação original, devido a uma falha na chamada da API que cause incerteza quanto a se a cobrança já teria sido criada ou não.

<sup>70</sup> Cabe enfatizar, inclusive, que não pode existir uma cobrança para pagamento imediato e uma cobrança para pagamento com vencimento que utilizem o mesmo txid.

<sup>71</sup> Caberá ao PSP do usuário recebedor explicar que, nesse caso, a idempotência não poderá ser garantida. Sendo assim, podem ser geradas cobranças que nunca serão retornadas ao usuário recebedor. O modelo de negócios do usuário recebedor deve ser analisado pelo seu PSP, a fim de garantir a adequação desse modelo (em que o PSP gera o txid) ao modelo de negócios praticado pelo usuário recebedor.

### 5.3.2 txid no contexto dos QR Codes Estáticos

Um aspecto importante a ser observado é que o PSP Recebedor não tem a possibilidade de assegurar que o txid não se repita em QR Codes estáticos, uma vez que eles não são gerados por meio da API Pix e, portanto, não são previamente registrados pelo PSP recebedor.

Além disso, há casos de uso em que a repetição de um txid em QR Codes estáticos é desejável (ex. 6.1.1) e outros em que deve ser evitada (ex. 6.1.3). Desta forma, a consistência dos pagamentos realizados por meio de QR Codes estáticos fica totalmente a cargo do usuário recebedor<sup>72</sup>.

Assim, em relação a QR Codes estáticos, a única funcionalidade atualmente prevista na API Pix refere-se a retornar as transações Pix para o usuário recebedor com o txid em questão, possibilitando algum nível básico de confirmação/conciliação de pagamentos relacionados a essa modalidade de QR Code.

## 5.4 Grupos de funcionalidades

As funcionalidades da API Pix, na versão atual, estão definidas em seis grupos:

I – Gerenciamento de Cobranças com pagamento imediato (Cob), que trata do ciclo de vida da entidade **Cobrança**:

Criação e atualização de cobrança	PUT /cob/{txid}
Criação de uma cobrança sem passar txid	POST /cob
Consulta de uma cobrança	GET /cob/{txid}
Consulta de lista de cobranças	GET /cob/
Alteração de uma cobrança	PATCH /cob/{txid}

II – Gerenciamento de Cobranças para pagamento com vencimento (CobV), que trata do ciclo de vida da entidade **Cobrança** para **pagamentos com vencimento**:

Criação e atualização de cobrança	PUT /cobv/{txid}
Consulta de uma cobrança	GET /cobv/{txid}
Consulta de lista de cobranças	GET /cobv/
Alteração de uma cobrança	PATCH /cobv/{txid}

III – Gerenciamento de lote de Cobranças para pagamento com vencimento (LoteCobV), que trata do ciclo de vida da entidade **Lote de cobranças com vencimento**:

Criação de lote de cobrança	PUT /lotecobv/{id}
Consulta de um lote de cobrança	GET /lotecobv/{id}
Consulta lista de lotes de cobranças	GET /lotecobv/

---

<sup>72</sup> O contexto explorado aqui é a API Pix. O preenchimento do txid em um cenário de geração de QR Codes diretamente via o aplicativo do PSP do recebedor fica a cargo do PSP do recebedor, desde que seguidos os preceitos mínimos estabelecidos no Manual de requisitos Mínimos para a Experiência do Usuário.

IV – Gerenciamento de Pix Recebidos (Pix), que trata do ciclo de vida das entidades **Pix** e **Devolução**:

Consulta de um Pix específico	GET /pix/{e2eid}
Consulta de lista de Pix	GET /pix/
Solicitação de devolução	PUT /pix/{e2eid}/devolucao/{id}
Consulta de devolução	GET /pix/{e2eid}/devolucao/{id}

V – Gerenciamento de “Localizações”<sup>73</sup>, que trata da entidade **PayloadLocation**:

Criação de uma nova “location”	POST /loc
Consulta de lista de “locations”	GET /loc
Consulta de uma location específica	GET /loc/{id}
Desvincular uma cobrança de uma location	DELETE /loc/{id}/txid

VI – Gerenciamento de Notificações (*Webhook*), que trata da entidade **Webhook**:

Associa URL ao webhook	PUT /webhook/{chave}
Consulta o status da webhook	GET /webhook/{chave}
Remove o webhook	DELETE /webhook/{chave}
Consulta webhooks cadastrados	GET /webhook

## 6. Casos de Uso

O objetivo dessa seção é trazer exemplos de como a API Pix pode ser utilizada na automação das interações entre usuários recebedores e seus respectivos PSPs em transações associadas ao Pix. Esses casos de uso **NÃO** pretendem esgotar as formas de utilização ou as funções disponibilizadas pela API Pix.

### 6.1 QR Code Estático

Para o contexto de QR Code Estático, a API Pix será estruturada em torno dos casos de uso listados a seguir<sup>74</sup>:

#### 6.1.1 Pagamento no ato da compra com QR Code Estático fixo e sem valor definido.

**Aplicação:** aplicável em diversos cenários, especialmente em negócios com pequeno volume de transações.

**Premissa:** a conciliação de pagamento, se necessária, só pode ser feita de forma bem simplificada, sem a possibilidade de tratar transações em paralelo.

<sup>73</sup> Chamamos aqui de “localização” as URLs que servem payloads JSON que representam uma cobrança.

<sup>74</sup> Vale lembrar que a API Pix não é utilizada para a geração de QR Codes estáticos.

1. O usuário recebedor imprime um QR Code com a sua Chave Pix, um identificador de transação (txid) e sem valor definido. Como todas as transações terão o mesmo txid, a capacidade de conciliação posterior é limitada;
2. O usuário recebedor afixa o QR Code impresso em um local visível no seu estabelecimento, por exemplo, próximo ao caixa;
3. O usuário pagador, ao realizar a compra ou no *checkout*, lê o QR Code impresso por meio do App de seu PSP, informa o valor e efetua o pagamento;
4. O software de automação do usuário recebedor consulta, por meio da API Pix, os últimos Pix recebidos com aquele txid:

*Serviço invocado: GET /pix?txid={txid} → Podem ser informados parâmetros para limitar a consulta aos últimos minutos, por exemplo (parâmetros “início” e “fim” podem ser usados).*

5. O usuário recebedor verifica a lista de pagamentos que foram creditados, observando sempre o último pagamento e o valor pago; e
6. O cliente é liberado, após a confirmação do pagamento, e pode seguir com a mercadoria ou sair do estabelecimento.

Este caso de uso pode ser implementado sem a necessidade de acesso à API Pix. Nesse caso, suprime-se o passo 4, mantendo o passo 5 com a confirmação da efetivação do crédito pelo usuário recebedor por meio do App provido pelo PSP Recebedor.

#### 6.1.2 Pagamento no ato da compra com QR Code Estático fixo com valor definido

**Aplicação:** similar ao caso anterior, para cenários envolvendo a venda de produtos padronizados, vendidos a preço fixo.

**Premissa:** conciliação de pagamento, se necessária, só pode ser feita de forma bem simplificada, sem a possibilidade de tratar transações em paralelo.

1. O usuário recebedor imprime um ou mais QR Codes com a sua Chave Pix, um identificador da transação (txid) e com um valor definido. Em caso de mais de um produto sendo vendido, a alternativa é a criação de vários QR Codes impressos, cada um com um txid diferente, associado, por exemplo, a um código de um produto distinto;
2. O usuário recebedor afixa o QR Code impresso em um local visível no seu estabelecimento;
3. O usuário pagador, ao realizar a compra ou o checkout, lê o QR Code impresso por meio do App de seu PSP e efetua o pagamento;
4. O software de automação do usuário recebedor consulta, por meio da API Pix, os últimos Pix recebidos:

*Serviço invocado: GET /pix?txid={txid} → Podem ser informados parâmetros para limitar a consulta aos últimos minutos, por exemplo (parâmetros “início” e “fim” podem ser usados).*

5. O usuário recebedor verifica a lista de pagamentos que foram creditados, observando sempre o último pagamento e o valor pago; e
6. O cliente é liberado, após a validação, e pode seguir com a mercadoria ou sair do estabelecimento.



Este caso de uso pode ser implementado sem a necessidade de acesso à API Pix. Nesse caso, suprime-se o passo 4, mantendo o passo 5 com a confirmação da efetivação do crédito pelo usuário recebedor por meio do App provido pelo PSP Recebedor.

Um eventual uso do txid como identificação do produto permite algum nível de controle posterior (e.g., quantos produtos de cada tipo foram vendidos em uma determinada data ou período), por meio de consultas ao extrato ou à própria API Pix.

### 6.1.3 Pagamento no ato da compra com QR Code Estático gerado no ato da compra

**Aplicação:** cenários diversos, em especial envolvendo negócios de pequeno porte, mas que exigem um processo de conciliação dos recebimentos.

**Premissa:** software de automação que saiba gerar QR Codes Estáticos.

1. O usuário recebedor, por meio do software de automação, imprime um QR Code estático com a sua chave, e um identificador da transação (txid) único (i.e., que não tenha sido usado anteriormente);
2. O usuário pagador, ao realizar a compra ou no *checkout*, lê o QR Code impresso por meio do App de seu PSP e efetua o pagamento;
3. O software de automação do usuário recebedor consulta, por meio da API Pix, se foi recebido um Pix com aquele txid:

*Serviço invocado: GET /pix?txid={txid} → Deve ser informado o parâmetro “txid” para limitar a consulta.*

4. O usuário recebedor verifica a liquidação do recebimento; e
5. O cliente é liberado, após a validação, e pode seguir com a mercadoria ou sair do estabelecimento.

Alternativamente, o passo 3 pode ser realizado com o uso de *webhooks* configurados no serviço correspondente. Nesse caso, o usuário recebedor seria informado pelo PSP Recebedor do crédito de um Pix associado a um txid na sua conta transacional.

## 6.2 QR Code Dinâmico

### 6.2.1 Pagamento imediato (no ato da compra) com QR Code Dinâmico

**Aplicação:** Comerciantes com volumes de vendas médios ou altos. Comércio *online*.

1. O usuário pagador, ao realizar a compra, informa que deseja pagar com Pix;
2. O software de automação utilizado pelo usuário recebedor acessa a API Pix para criação de uma cobrança e, com os dados recebidos como resposta, gera um QR Code Dinâmico, que é apresentado em um dispositivo de exibição qualquer:
  - a. em uma compra presencial, tipicamente uma tela próxima ao caixa ou mesmo um POS;
  - b. nas compras *online*, no dispositivo em uso pelo pagador.

*Serviço invocado: PUT /cob/{txid} → Devem ser informados todos os dados necessários para criação do payload da cobrança, conforme especificação detalhada.*

*Alternativamente, se o usuário recebedor não quiser identificar a cobrança imediata com seu próprio número {txid}, pode-se optar por utilizar o método POST /cob<sup>75</sup>.*

3. O usuário pagador lê, a seguir, o QR Code com o App do seu PSP e efetua o pagamento;
4. O usuário recebedor, de forma automatizada, por meio de nova consulta à API Pix, verifica se o pagamento foi realizado:

*Serviço invocado: GET /cob/{txid}*

5. O usuário recebedor libera os produtos para o usuário pagador ou, no caso das compras *online*, confirma o recebimento do pagamento.

Alternativamente, o passo 4 pode ser realizado com o uso de *webhooks* configurados no serviço correspondente. Nesse caso, o usuário recebedor seria informado pelo PSP Recebedor do crédito de um Pix associado a um txid na sua conta transacional.

### 6.2.2 QR Code dinâmico para pagamentos com vencimento

**Aplicação:** cobranças com prazo para pagamento pelo usuário pagador; cobranças de mensalidades referentes a serviços cobrados no modelo de assinatura; outras situações correlatas.

1. O usuário recebedor, por meio do software de automação utilizado por ele, acessa a API Pix para criação de uma cobrança, informando, entre outros dados, a data de vencimento, o valor da obrigação, descontos, multa e juros aplicáveis à cobrança. Com os dados recebidos da API Pix, gera um QR Code Dinâmico, que é enviado ao usuário pagador (por exemplo, por e-mail)

*Serviço invocado: PUT /cobv/{txid} → Devem ser informados todos os dados necessários para criação do payload da cobrança, conforme especificação detalhada.*

2. O usuário pagador lê, a seguir, o QR Code dinâmico com o App do seu PSP.
3. O APP do PSP extrai a “location” do Qr Code dinâmico.
4. O APP do PSP pagador, depois de efetuar as validações de segurança, acessa a *location* incorporando os parâmetros: município do pagador e DPP configurada para a data de vencimento.
5. O PSP pagador recupera os elementos de configuração de cobrança, em particular, o valor final.
6. O usuário pagador confirma as informações e agenda o pagamento para a data de vencimento.
7. Com o pagamento tendo sido agendado, na data agendada, o PSP pagador escolhe o melhor momento, dentro do dia especificado, para realizar a transação Pix.
8. O usuário recebedor, a fim de confirmar o recebimento de suas cobranças, pode realizar uma consulta na API do PSP Recebedor:

---

<sup>75</sup> A consequência de se utilizar POST /cob no lugar de PUT /cob/{txid} é que o método POST /cob não apresenta idempotência, enquanto o método PUT /cob/{txid} apresenta. A idempotência está presente em PUT /cob/{txid} porque no evento de um erro (por exemplo, uma falha na comunicação entre o usuário recebedor e seu PSP) pode acontecer de não se saber se a cobrança foi ou não criada de fato; nesse caso, a chamada poderia ser simplesmente repetida sem nenhum problema ou efeito adverso: caso a cobrança já tenha sido criada, a nova chamada PUT /cob/{txid} retornaria a mesma informação que deveria ter sido inicialmente retornada. Não é o caso com POST /cob: em caso de problemas, não se sabe se a cobrança foi criada e não há um identificador previamente conhecido, que permita verificar o status da cobrança. Assim, a chamada, ao ser repetida (POST /cob), ensejará a criação de uma nova cobrança.

*Serviço invocado: GET /cobv/{txid}.*

Alternativamente, o usuário pode ser notificado de que a cobrança foi honrada, por meio de serviço de notificação previamente configurado:

*Serviço invocado: PUT /Webhook/{chave}*

9. O usuário recebedor dá quitação à obrigação, liberando, por exemplo, a continuação do uso do produto ou serviço por parte do usuário pagador.

### 6.2.3 Lote de Cobranças com vencimento

**Aplicação:** usuários recebedores que disponham de uma quantidade expressiva de cobranças para serem criadas, de maneira que seja mais eficiente solicitar a criação de um conjunto grande de cobranças de uma só vez.

1. O usuário recebedor, por meio do software de automação utilizado por ele, acessa a API Pix para criação de um lote de cobranças, informando, para cada cobrança, entre outros dados, a data de vencimento, o valor da obrigação, descontos, multa e juros aplicáveis à cobrança, juntamente com seu identificador txid.

*Serviço invocado: PUT /lotecobv/{id} → devem ser informados todos os dados necessários para criação dos payloads das cobranças, conforme especificação detalhada.*

2. Como retorno, o usuário recebedor recebe uma mensagem de sucesso significando que a solicitação foi recebida e está em processamento.
3. O usuário recebedor, quando achar conveniente, valendo-se do “id” utilizado para criar o lote de cobranças, verifica o status de sua requisição para identificar o status de criação de cada cobrança individual:

*Serviço invocado: GET /lotecobv/{id} → deve ser informado o “id” utilizado para criação do lote.*

4. O usuário recebedor, como retorno do endpoint “GET /lotecobv/{id}”, recebe uma lista de status correspondentes a cada solicitação de criação de cobranças.
5. Quando o usuário recebedor achar conveniente, pode acessar as informações específicas das cobranças criadas com sucesso:

*Serviço invocado: GET /cobv?lotecobvid={id} → deve ser informado o “id” que foi utilizado para criação do lote.*

6. Como retorno, o usuário recebedor recebe uma consulta paginada contendo todas as cobranças criadas com sucesso associadas ao lote em questão.
7. O usuário recebedor utiliza as informações obtidas no passo anterior para efetuar a criação de seus QR Codes ou “Pix Copia e Cola”.
8. O usuário recebedor envia as cobranças criadas com sucesso em lote aos seus devedores correspondentes usando o meio que achar mais adequado.

#### 6.2.4 O QR Code dinâmico “impresso”

**Aplicação:** estabelecimentos comerciais cujos caixas são conectados ao sistema de automação comercial, mas não dispõem de dispositivo para exibição de QR Codes; cenários variados de uso.

1. Para configurar a API Pix para esse caso de uso, o primeiro passo é criar uma “location” (a URL que exibirá as cobranças, e que será configurada no QR Code dinâmico:

*Serviço invocado: POST /loc → será retornado o identificador da “location” e a URL a qual será utilizada para exibir as cobranças. Deve ser informado se a location servirá cobranças imediatas (cob) ou cobranças para pagamento com vencimento (cobv). No presente exemplo, será uma location que servirá cobranças imediatas.*

2. O usuário recebedor, em seguida, por meio de seus softwares de automação, imprime um QR Code Dinâmico configurado com a “location” obtida e o afixa em um local apropriado, próximo a seu caixa (podem ser gerados e impressos QR Codes distintos, cada um com sua própria “location”, caso existam múltiplos caixas na loja).
3. Quando o primeiro cliente comparece ao caixa para realizar o *checkout*, o caixa, via automação, cria uma cobrança estabelecendo que será utilizado a “location” previamente criada (o software deverá associar a cobrança à “location” configurada no QR Code impresso que fica próximo ao caixa em que o *checkout* está sendo realizado):

*Serviço invocado: PUT /cob/{txid} → Devem ser informados todos os dados necessários para criação do payload da cobrança, conforme especificação detalhada. **Em particular**, é informado o parâmetro “location” (loc.id) detalhando que esta cobrança específica utilizará a “location” previamente criada.*

4. O usuário pagador, usando o aplicativo mobile de seu PSP pagador escolhido, realiza a leitura do QR Code impresso, confere as informações e confirma o pagamento.
5. A automação comercial verifica que o pagamento foi liquidado:

*Serviço invocado: GET /cob/{txid} → deve ser utilizado o mesmo ‘txid’ passado na criação do payload da cobrança.*

6. O cliente é liberado, após a validação, e pode seguir com a mercadoria ou sair do estabelecimento;
7. Uma vez que o cliente foi liberado, a automação desvincula o QR Code Dinâmico impresso desta cobrança já liquidada, comandando a “location” a não exibir nenhuma cobrança:

*Serviço invocado: DELETE /loc/{id}/txid → O identificador da location, {id}, a ser informado, é o identificador obtido ao se estabelecer uma “location” (loc.id) para o caixa em questão.*

8. O caixa encontra-se, neste ponto, preparado para receber o próximo cliente. O QR Code impresso, neste ponto, não exibirá nenhuma cobrança ao ser lido.

## 6.3 Outros casos de uso

### 6.3.1 Efetuar uma devolução

**Aplicação:** várias (devolução de produto, erro na cobrança, indisponibilidade do produto em estoque etc.)

**Premissa:** um pagamento foi recebido via Pix.

1. O usuário pagador solicita ao usuário recebedor, via algum meio de comunicação adequado, a devolução total ou parcial de um pagamento realizado;
2. O usuário recebedor concorda e identifica o pagamento original realizado pelo Pix. Há duas situações possíveis:

- a. Quando o Pix está associado a uma Cobrança:

*Serviço invocado: GET /cob/{txid} → Como resultado, será recebida uma entidade Cobrança que contém uma relação dos Pix recebidos, cada um com a sua identificação (EndToEndId).*

- b. Quando o Pix não está associado a uma Cobrança. Nesse caso, é necessário saber, por outros meios, o EndToEndId do Pix original. Alternativamente, pode ser uma consulta ampla, trazendo a relação dos Pix recebidos.

*Serviço invocado: GET /pix/ → Podem ser informados parâmetros para limitar a consulta temporalmente (parâmetros início e “fim” podem ser usados). Além disso, pode-se limitar a busca a um usuário pagador específico, por meio do CNPJ/CPF do pagador.*

3. O software de automação do usuário recebedor aciona a API Pix para realizar a devolução.  
*Serviço invocado: PUT /pix/{e2eid}/devolucao/{id} → No caso, “id” é um código gerado pelo sistema do usuário recebedor que identifica a devolução associada ao Pix original. Observar, que um Pix pode ter várias devoluções associadas a ele, desde que o montante das devoluções não ultrapasse o valor do Pix original. O “id” deve ser único por EndToEndID Pix.*

4. O software de automação do usuário recebedor aciona a API Pix para verificar se a devolução foi liquidada:

*Serviço invocado: GET /pix/{e2eid}/devolucao/{id}*

5. O usuário pagador recebe um Pix com o valor de devolução acordado.

### 6.3.2 Remover uma Cobrança

**Aplicação:** várias (por exemplo: quando um QR Code foi gerado, mas não é mais válido, pois o cliente desistiu da compra).

**Premissa:** há uma Cobrança gerada e válida.

1. O usuário recebedor percebe que, por alguma razão, precisa remover uma cobrança que foi anteriormente gerada;
2. O usuário recebedor solicita a remoção da cobrança via API Pix;

*Serviço invocado: PATCH /cob[v]/{txid} → Deve ser atribuído o Status para REMOVIDO\_PELO\_USUARIO\_RECEBEDOR.*

3. A cobrança é removida e, se mesmo assim, o usuário pagador ler o QR Code, o PSP Pagador deverá indicar que a cobrança foi excluída;
4. Não se pode alterar e nem remover uma cobrança cujo status esteja em CONCLUÍDA. O Status CONCLUÍDA é **final**;
5. Uma cobrança terá seu status alterado para CONCLUÍDA quando um Pix associado ao txid da cobrança for recebido.

### 6.3.3 Alterar uma cobrança

**Aplicação:** várias (por exemplo: quando uma cobrança foi gerada, mas o cliente solicitou um produto a mais; ou quando se percebe um erro no valor total).

**Premissa:** há uma cobrança gerada e válida.

1. O usuário recebedor percebe que, por alguma razão, precisa alterar uma cobrança que foi anteriormente gerada;
2. O usuário recebedor solicita a alteração da cobrança via API Pix;  
*Serviço invocado: PATCH/cob[v]/{txid}<sup>76</sup> → Devem ser passadas todas as informações corrigidas, conforme especificação detalhada.*
3. A cobrança é alterada e, numa nova leitura do QR Code, as informações estarão atualizadas;
4. Não se pode alterar e nem remover uma cobrança cujo status esteja em CONCLUÍDA. O Status CONCLUÍDA é **final**;
5. Uma cobrança terá seu status alterado para CONCLUÍDA quando um Pix associado ao txid da cobrança for recebido.

### 6.3.4 Configuração de Webhooks

**Aplicação:** para usuários recebedores que trabalham com grande número de recebimentos via Pix e que desejam evitar um oneroso processo de *polling* na API Pix de seu PSP recebedor.

**Premissa:** o usuário recebedor possui uma infraestrutura de TI apta para tratar os *webhooks*.

1. O usuário recebedor aciona a API Pix, indicando a URL na qual deseja receber as informações sobre Pix recebidos.  
*Serviço invocado: PUT /webhook/{chave} → Deve ser atribuída a URL base que receberá os callbacks enviados pelo PSP Recebedor ao usuário recebedor. Cada webhook está associado a uma chave Pix de maneira que os Pix informados a um webhook estão associados à chave Pix que também está associada ao webhook em questão. Se um Pix recebido estiver associado a uma chave que não esteja associado a um webhook, não haverá notificação.*
2. A partir desse momento, o usuário Recebedor receberá chamadas nesse *endpoint* com a lista de Pix recebidos associados a um txid, e associados à respectiva chave Pix, conforme esses Pix forem sendo liquidados. Pix recebidos que não estejam associados a um txid não gerarão chamadas.

---

<sup>76</sup> Pode-se alterar tanto uma cobrança imediata quanto uma cobrança com vencimento.



## ANEXO II – API Pix: Especificação Técnica

### 1. Introdução

O presente anexo tem por objetivo apresentar a especificação da **API Pix**, que normatiza a comunicação entre usuários e respectivos PSPs nas operações relacionadas ao Pix. Para as definições que seguem, o Banco Central do Brasil (BC) valeu-se de subsídios colhidos junto ao mercado por meio de consultas previamente enviadas e de reuniões bilaterais com participantes e associações, seguindo o modelo de criação conjunta, adotado desde o começo dos trabalhos no âmbito do Fórum Pagamentos Instantâneos.

### 2. Protocolos e tecnologias

A API Pix adotará os seguintes protocolos e tecnologias:

**Definição da API:** A API Pix está detalhada no formato OpenAPI 3.0<sup>77</sup>.

**Formato:** O formato de dados utilizados é o JSON<sup>78</sup>.

**Protocolo:** a automação do usuário recebedor interage com a API utilizando *web services baseados em REST*<sup>79</sup> sobre HTTPS.

### 3. Segurança

Os PSPs devem desenvolver e implementar a API seguindo boas práticas de segurança, atendendo aos requisitos obrigatórios abaixo e às recomendações detalhadas nesta seção.

#### 3.1 Requisitos de segurança obrigatórios

O PSP deve obrigatoriamente observar os seguintes requisitos:

1. A conexão à API deve ser criptografada utilizando o protocolo *TLS* versão 1.2 ou superior, permitindo apenas *cipher suites* que atendam ao requisito de *forward secrecy*<sup>80</sup>.
2. O PSP deve implementar o framework *OAuth 2.0 (RFC 6749)*<sup>81</sup> com *TLS* mútuo (*mTLS – RFC 8705*<sup>82</sup>) para autenticação na API, conforme especificações abaixo:

---

<sup>77</sup> <http://spec.openapis.org/oas/v3.0.3>

<sup>78</sup> JSON: JavaScript Object Notation.

[https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\\_Objects/JSON](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/JSON)

<sup>79</sup> Os webservices da API Pix serão baseados no estilo arquitetural REST, usando métodos HTTP para realizar as chamadas à API. Mais detalhes em: [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)

<sup>80</sup> <https://tools.ietf.org/html/bcp195#section-6.3>

<sup>81</sup> <https://tools.ietf.org/html/rfc6749>

<sup>82</sup> <https://tools.ietf.org/html/rfc8705>



- a. Os certificados digitais dos clientes da API poderão ser emitidos pelo próprio PSP ou por ACs externas, conforme definido por cada PSP. Não deverão ser aceitos certificados auto-assinados pelo cliente.
  - b. Cada PSP deve possuir seu próprio *Authorization Server* e *Resource Server* associado à API Pix, e ambos devem implementar *TLS* mútuo.
  - c. O *Authorization Server* do PSP deve implementar a técnica de vinculação do certificado do cliente aos *access tokens* emitidos ("*Client Certificate-Bound Access Tokens*"), conforme seção 3 da *RFC 8705*.
  - d. O *Resource Server* do PSP deve confirmar que o *thumbprint* do certificado associado ao *access token* apresentado pelo cliente é o mesmo do utilizado na autenticação *TLS* (*proof-of-possession*).
  - e. O fluxo *OAuth* a ser utilizado é o "*Client Credentials Flow*".
  - f. Os escopos *OAuth* serão definidos na especificação *Open API 3.0* da API Pix e permitirão associar diferentes perfis de autorização ao *software* cliente.
3. O processo de cadastro/*onboarding* do cliente para acesso à API deve ser realizado em ambiente logado no PSP, e deve incluir um canal seguro para envio das credenciais ao usuário, de forma a permitir a rastreabilidade das ações executadas.
  4. A API deve suportar múltiplos níveis de autorização ou papéis, segregando as funcionalidades de acordo com perfis (escopos *OAuth*) dos usuários clientes.
  5. O PSP deve implementar tecnologia que permita garantir a alta disponibilidade da API.
  6. A API deve garantir a confidencialidade e a integridade das informações dos usuários e de suas transações, tanto em trânsito como em repouso.
  7. O PSP deve manter logs de auditoria dos acessos à API pelo período mínimo de 1 ano.
  8. A credencial de acesso utilizada na autenticação (*Client\_ID*) deve ser vinculada ao CNPJ ou CPF do usuário recebedor e deve permitir acesso a recursos apenas de contas transacionais de titularidade do CNPJ ou CPF associado.
  9. Para a funcionalidade de webhooks, as notificações oriundas do PSP recebedor ao usuário recebedor trafegarão utilizando um canal *mTLS*.
    - a. Recomenda-se que os certificados utilizados para autenticação mútua no canal *TLS* do webhook sejam os mesmos da API Pix. De todo modo, não há objeção quanto à utilização de outros certificados, mediante acordo entre o PSP e o usuário recebedor.

## 3.2 Recomendações de segurança

É recomendado ao PSP:

1. Implementar múltiplos fatores de autenticação para o processo de cadastro/*onboarding* na API.
2. Desenvolver e implementar a API seguindo boas práticas de segurança, de forma a eliminar/reduzir ao máximo os riscos de segurança conforme última versão dos guias *OWASP API Security Top Ten*<sup>83</sup> e *CWE Top 25 Software Weaknesses*<sup>84</sup>.

---

<sup>83</sup> <https://owasp.org/www-project-api-security/>

<sup>84</sup> [https://cwe.mitre.org/top25/archive/2020/2020\\_cwe\\_top25.html](https://cwe.mitre.org/top25/archive/2020/2020_cwe_top25.html)

3. Possuir processo periódico de análise de vulnerabilidades, tanto estática como dinâmica da API.
4. Assegurar a segurança do desenvolvimento do *software* cliente<sup>85</sup> da API, mesmo que desenvolvido por terceiros. Sugere-se que o PSP institua e mantenha processo de homologação dos *softwares* clientes, estabelecendo critérios mínimos de segurança para que eles sejam autorizados a interagir com a API. Nesse caso, a API deve negar tentativas de comunicação de clientes não homologados.
5. Os usuários recebedores, como clientes da API, são um elo importante na segurança do sistema, e, portanto, recomenda-se que o PSP tome ações para mitigar os riscos do ambiente computacional dos seus usuários, uma vez que caso um risco se materialize em um incidente, o próprio PSP poderá ser afetado. Ações recomendadas (sem prejuízo para outras ações que o PSP julgar importantes):
  - a. Instituir e acompanhar programa de melhoria contínua da segurança dos usuários recebedores que utilizam a API;
  - b. Realizar campanhas de conscientização e compartilhamento de informações de segurança junto aos usuários;
  - c. Definir uma política de troca periódica do certificado, senha e outras credenciais utilizadas no acesso à API;
  - d. Validar a segurança do ambiente computacional dos usuários nos aspectos de infraestrutura, implementação e configuração do *software* cliente da API.
  - e. Exigir que as empresas e instituições que utilizem a API tenham uma Política de Segurança da Informação formalmente instituída.

O BC entende que os PSPs poderão adotar as tecnologias e soluções de segurança para a API que mais acharem apropriados, desde que sejam atendidos os requisitos obrigatórios de segurança e, sempre que possível, as recomendações descritas acima, com atenção também aos elementos listados nos tópicos a seguir.

### 3.3 Jornada de Adesão

Por jornada de adesão, entende-se o processo por meio do qual um usuário recebedor passa a utilizar os serviços de um PSP específico. Do ponto de vista da API Pix, tal processo deve incluir o fornecimento de credenciais de acesso (Client\_IDs e senhas) e de certificados ao usuário recebedor. Cada PSP terá autonomia para definir a jornada de Adesão para os seus clientes, utilizando os canais que julgar mais adequados.

No processo de adesão, o Client\_ID disponibilizado pelo PSP deve possuir um conjunto de escopos que determinarão as funcionalidades às quais o Usuário Recebedor terá acesso. Os critérios de autorização nos escopos são de responsabilidade do PSP, que pode criar critérios diferenciados em função das características do Usuário Recebedor.

Dessa forma, é possível, por exemplo, que determinadas funcionalidades estejam acessíveis apenas por usuários que cumpram requisitos adicionais de segurança estipulados por cada PSP.

---

<sup>85</sup> No contexto, o termo “cliente” é utilizado para designar o hardware e o software utilizado pelo Usuário Recebedor, ou por terceiro por ele selecionado, para interagir com a API Pix provida por um PSP.

## ANEXO III – Cobranças para pagamentos com vencimento: Criação, alteração e cálculo

### 1. Introdução

O presente anexo tem por objetivo apresentar como são criadas e atualizadas as cobranças com data de vencimento. Descreve, ainda, como o **PSP do recebedor** deve calcular, a partir das informações apresentadas pelo usuário recebedor ao criar e/ou atualizar uma cobrança para pagamentos com vencimento, o valor final de uma cobrança para pagamento com vencimento, de forma a incorporar as componentes relacionadas aos abatimentos, descontos, juros e multas que podem incidir sobre essas cobranças.

### 2. Criando uma cobrança para pagamento com vencimento

Para criar uma cobrança para pagamentos com vencimento, é necessário que o usuário recebedor conecte seus sistemas (ou de terceiros por ele contratados) à API Pix provida pelo PSP por ele contratado.

#### 2.1 Estrutura para criação e atualização de uma cobrança para pagamento com vencimento

Para cada cobrança a ser criada, o usuário recebedor deverá enviar as seguintes informações:

#	Campo	Mult.	Nome Campo JSON	Tipo	OU
1.1	Data de vencimento do pagamento	[1..1]	calendario.dataDeVencimento	String	
1.2	Validade Após Vencimento (em dias corridos)	[1..1]	calendario.validadeAposVencimento	Integer	
2.1	CPF do usuário devedor	[0..1]	devedor.cpf	String	OU(
2.2	CNPJ do usuário devedor	[0..1]	devedor.cnpj	String	
2.3	Nome do usuário devedor	[1..1]	devedor.nome	String	
2.4	e-Mail do usuário devedor	[0..1]	devedor.email	String	
2.5	Logradouro do devedor	[0..1]	devedor.logradouro	String	
2.6	Cidade do devedor	[0..1]	devedor.cidade	String	
2.7	Unidade da federação do devedor	[0..1]	devedor.uf	String	
2.8	CEP do devedor	[0..1]	devedor.cep	String	
3	Identificador da localização do <i>payload</i> associado à cobrança	[0..1]	loc.id	Integer	
4.1	Valor original do documento	[1..1]	valor.original	String	
4.2.1	Modalidade de abatimentos ou outras deduções, conforme tabela de domínios	[0..1]	valor.abatimento.modalidade	Integer	

4.2.2	Abatimentos ou outras deduções aplicadas ao documento, em valor absoluto ou percentual do valor original do documento	[0..1]	valor.abatimento.valorPerc	String	
4.3.1	Modalidade de desconto, conforme tabela de domínios.	[0..1]	valor.desconto.modalidade	Integer	
4.3.2	Descontos por pagamento antecipado, com data fixa. Matriz com até três elementos, sendo que cada elemento é composto por um par "data e valorPerc", para estabelecer descontos percentuais ou absolutos, até aquela data de pagamento.	[0..1]	valor.desconto.descontoDataFixa	Array	OU(
4.3.3	Desconto em valor absoluto ou percentual por dia, útil ou corrido, conforme valor.desconto.modalidade	[0..1]	valor.desconto.valorPerc	String	)
4.4.1	Modalidade de juros, conforme tabela de domínios.	[0..1]	valor.juros.modalidade	Integer	
4.4.2	Valor absoluto ou percentual dos juros, a ser utilizado no cálculo dos juros, conforme "valor.juros.modalidade"	[0..1]	valor.juros.valorPerc	String	
4.5.1	Modalidade da multa, conforme tabela de domínios.	[0..1]	valor.multa.modalidade	Integer	
4.5.2	Multa em valor absoluto ou percentual, conforme "valor.multa.modalidade"	[0..1]	valor.multa.valorPerc	String	
5	Chave Pix do recebedor	[1..1]	chave	String	
6	Identificador da transação	[1..1]	txid	String	
7	Solicitação ao Pagador	[0..1]	solicitacaoPagador	String	
8	Conjunto livre de caracteres, com limite de tamanho	[0..1]	infoAdicionais	Array[InfoAdicional]	

A seguir, apresenta-se uma breve explanação sobre os principais campos listados para a criação de uma cobrança para pagamento com vencimento. Para informações em maior detalhe técnico, a API Pix é a referência indicada<sup>86</sup>.

<sup>86</sup> Disponível em <<https://github.com/bacen/pix-api>> (ver "Cobv: PUT" para criação de uma única cobrança; e "LoteCobV: PUT" para criação de um lote de cobranças).

- **Calendário**

Campos do objeto **calendário**:

- **calendario.dataDeVencimento**: [obrigatório] trata-se de uma data, no formato 'yyyy-mm-dd', segundo ISO 8601. É a data de vencimento da cobrança, que pode ser paga em qualquer horário do dia. Exemplo: 2020-10-19.



Sempre que a data de vencimento cair em um fim de semana ou em um feriado para o usuário pagador, ela deve ser automaticamente prorrogada para o primeiro dia útil subsequente. Todos os campos que façam referência a esta data (validadeAposVencimento; desconto; juros e multa) devem assumir essa prorrogação, quando for o caso.

- **calendario.validadeAposVencimento**: [opcional] (int32) Trata-se da quantidade de dias corridos após `calendario.dataDeVencimento` em que a cobrança poderá ser paga. Aplica-se o valor deste campo sobre o vencimento original da cobrança acrescentando-se o número de dias corridos nos quais a cobrança ainda poderá ser paga, após vencida.



Sempre que a data de validade após o vencimento cair em um fim de semana ou em um feriado para o usuário pagador, ela deve ser automaticamente prorrogada para o primeiro dia útil subsequente.

- **valor.abatimento.modalidade**: indica como o valor do abatimento deve ser calculado. Pode assumir os seguintes valores:

Descrição	Domínio
Valor Fixo	1
Percentual	2

- **valor.abatimento.valorPerc**: valor ou percentual do abatimento aplicado à cobrança. É obrigatório se modalidade for preenchida.
- **valor.desconto.modalidade**: [opcional] indica como o valor do desconto deve ser calculado. Pode assumir os seguintes valores:

Descrição	Domínio
Valor Fixo até a data informada	1
Percentual até a data informada	2
Valor por antecipação dia corrido	3
Valor por antecipação dia útil	4
Percentual por antecipação dia corrido	5
Percentual por antecipação dia útil	6

- **valor.desconto.descontoDataFixa:** [opcional] descontos no valor da cobrança se pago até determinada data. Este objeto é composto por uma matriz com até três elementos, sendo que cada elemento é composto por um par “data e valorPerc”, para estabelecer descontos percentuais ou absolutos, até a data de pagamento em questão. Só deve ser preenchido se o campo modalidade for preenchido e assumir o valor “1” ou “2”. Exemplos de preenchimento:

```
valor.desconto.modalidade = 1
valor.desconto.descontoDataFixa[0].data = "2020-10-13"
valor.desconto.descontoDataFixa[0].valorPerc = "200.00"
valor.desconto.descontoDataFixa[1].data = "2020-10-19"
valor.desconto.descontoDataFixa[1].valorPerc = "100.00"
```



Sempre que a data limite para desconto cair em um fim de semana ou em um feriado para o usuário pagador, ela deve ser automaticamente prorrogada para o primeiro dia útil subsequente.

- **valor.desconto.valorPerc:** [opcional] valor absoluto ou percentual a ser utilizado no cálculo do desconto, conforme a modalidade de desconto escolhida. Só deve ser preenchido se o campo modalidade for preenchido e assumir o valor “3” a “6”.
- **valor.juros.modalidade:** [opcional] indica como o valor dos juros aplicáveis ao pagamento da cobrança em atraso deve ser calculado. Pode assumir os seguintes valores:

Descrição	Domínio
Valor (dias corridos)	1
Percentual ao dia (dias corridos)	2
Percentual ao mês (dias corridos)	3
Percentual ao ano (dias corridos)	4
Valor (dias úteis)	5
Percentual ao dia (dias úteis)	6
Percentual ao mês (dias úteis)	7
Percentual ao ano (dias úteis)	8

- **valor.juros.valorPerc:** [opcional] valor absoluto ou percentual a ser utilizado no cálculo do desconto, conforme a modalidade de desconto escolhida. Se modalidade tiver sido preenchida, então este campo também deve ser preenchido.
- **valor.multa.modalidade:** [opcional] indica como o valor da multa aplicável ao pagamento da cobrança em atraso deve ser calculado. Pode assumir os seguintes valores:

Descrição	Domínio
Valor fixo	1
Percentual	2

- **valor.multa.valorPerc:** [opcional] valor absoluto ou percentual a ser utilizado no cálculo da multa, conforme a modalidade de escolhida. Se modalidade tiver sido preenchida, então este campo também deve ser preenchido.

- **Chave**

O campo **chave**, obrigatório, determina a chave Pix registrada no DICT que será utilizada para a cobrança, identificando o usuário recebedor, bem como os dados da conta transacional à qual a cobrança deve estar atrelada.

- **txid**

O **txid**, no contexto de uma cobrança para pagamento com vencimento, é criado pelo usuário recebedor e encontra-se sob sua responsabilidade, devendo ser único para um mesmo PSP por ele contratado, seja a cobrança para pagamentos imediatos ou com vencimento (único por CPF/CNPJ do usuário recebedor e PSP). Cabe ao PSP recebedor validar essa regra na API Pix.

O objetivo desse campo é possibilitar ao usuário recebedor realizar a conciliação de pagamentos referentes a suas cobranças. Na pacs.008, o PSP do Pagador deve enviar essa informação como *TransactionIdentification* <TxId>.<sup>87</sup> O campo **txid** deve ter, no mínimo, **26 caracteres e, no máximo, 35 caracteres**.

- **Solicitação ao Pagador**

O campo **solicitacaoPagador**, opcional, determina um texto a ser apresentado ao pagador para que ele possa digitar uma informação correlata, em formato livre, a ser enviada ao recebedor. Esse texto<sup>88</sup> será preenchido, na pacs.008, pelo PSP do pagador, no campo *RemittanceInformation* <RmtInf>. O tamanho do campo <RmtInf> na pacs.008 está limitado a 140 caracteres.

- **Informações Adicionais**

O campo **infoAdicionais**, se estiver presente, se refere a uma lista em que cada elemento deve utilizar o esquema abaixo:

Subcampo JSON (infoAdicionais)	Presença	Tipo JSON	Propósito
nome	Obrigatório	String	Nome do campo
valor	Obrigatório	String	Dados do campo

<sup>87</sup> Vale lembrar que, no JSON, “txid” é escrito todo em minúsculas, enquanto a tag “TxId” da pacs.008 alterna maiúsculas e minúsculas.

<sup>88</sup> Importante destacar que é o texto digitado livremente pelo pagador que é enviado na pacs.008, e não o texto apresentado no campo *solicitacaoPagador*.



Os limites relativos ao tamanho de cada campo e à quantidade de elementos da lista estão tratados na especificação da API Pix<sup>89</sup>.

Cada respectiva informação adicional contida na lista (*nome* e *valor*) deve ser apresentada ao pagador. Exemplo: campo *infoAdicionais* no *JWSPayload*:

```
infoAdicionais[0].nome: "campo1"  
infoAdicionais[0].valor: "informação adicional 01"  
infoAdicionais[1].nome: "campo2"  
infoAdicionais[1].valor: "informação adicional 02"
```

Esse campo deve ser utilizado para indicar as regras que regem o cálculo do valor da cobrança (descontos, juros e multas), entre outras informações que o usuário recebedor queira transmitir ao usuário pagador.

---

<sup>89</sup> Disponível em <<https://github.com/bacen/pix-api>>.

## 2.2 Cálculo do valor da cobrança

O valor final da Cobrança, a ser calculado pelo PSP recebedor, deve ser calculado de acordo com a fórmula a seguir:

$$V_f = V_o - V_a - V_d + V_j + V_m$$

*Equação 1*

Onde:

$V_f$ : valor final da cobrança, que deve pago pelo usuário pagador;

$V_o$ : valor original da cobrança;

$V_a$ : valor de abatimento aplicável à cobrança;

$V_d$ : valor de desconto aplicável à cobrança;

$V_j$ : valor de juros aplicável pelo atraso no pagamento da cobrança;

$V_m$ : valor de multa aplicável pelo atraso no pagamento da cobrança.

Na apresentação do valor a ser pago pelo usuário pagador, as componentes com valor zero não devem ser apresentadas ao usuário pagador. As componentes do valor final da cobrança com valor não nulo devem ser apresentadas de forma individualizada. Se o valor final da cobrança é igual ao valor nominal da cobrança, então apenas o valor nominal da cobrança deve ser apresentado.

### 2.2.1. Cálculo do valor de abatimento ( $V_a$ )

Por padrão, uma Cobrança Pix é criada sem abatimentos. O abatimento deve ser acordado entre o devedor e o beneficiário da cobrança, que é quem pode atualizar o valor da cobrança junto a seu prestador de serviços de pagamento.

O abatimento pode ser realizado em valor absoluto ou em percentual do valor original, conforme modalidade do abatimento. Se a modalidade escolhida pelo recebedor for por valor fixo, então o valor a ser abatido é o próprio valor informado pelo recebedor. Se for informado o valor percentual, então o valor de abatimento será calculado conforme a fórmula a seguir:

$$V_a = V_o \times \frac{I_a}{100}$$

*Equação 2*

Onde:

$V_a$ : valor de abatimento aplicável à cobrança;

$V_o$ : valor original da cobrança;

$I_a$ : percentual de abatimento, informado com 2 casas decimais.

O valor calculado para  $V_d$  deve ser truncado com duas casas decimais.

Campos JSON referenciados:

**valor.original** ( $V_o$ )

**valor.abatimento.modalidade**

**valor.abatimento.valorPerc** ( $I_a$ )

O campo **valor.abatimento.modalidade** pode assumir os seguintes valores:

Descrição	Domínio
Valor Fixo	1
Percentual	2

## 2.2.2. Cálculo do valor de desconto ( $V_d$ )

### 2.2.2.1. Fixo até a data informada<sup>90</sup>

Nesse caso, o usuário recebedor que indicar descontos, até uma determinada data. O usuário recebedor poderá fixar até três datas antes do vencimento. Além disso, ele poderá indicar se o desconto será um valor absoluto ou um percentual do valor do documento. Se o valor do desconto for dado em percentual, será calculado pela fórmula a seguir:

$$V_d = (V_o - V_a) \times \left( \frac{I_d}{100} \right)$$

*Equação 3*

Onde:

$V_d$ : valor do desconto;

$V_o$ : valor original da cobrança;

$V_a$ : valor de abatimento aplicável à cobrança;

$I_d$ : percentual de desconto, informado com 2 casas decimais.

Podem ser informadas até 3 datas para aplicação de desconto antes da data de vencimento. A data de cálculo deve ser comparada com a data mais antiga (elemento1) na matriz. Se a data for menor ou igual a essa data; aplica-se o valor de desconto (absoluto ou percentual) associado a esse elemento. Se a data for maior, compara-se com a data do segundo elemento e assim por diante.



Sempre que a data limite para desconto cair em um fim de semana ou em um feriado para o usuário pagador, ela deve ser automaticamente prorrogada para o primeiro dia útil subsequente.

<sup>90</sup> Modalidades de desconto com domínio "1" ou "2".



O valor calculado para  $V_d$  deve ser truncado com duas casas decimais.

Campos JSON referenciados:

**valor.original** ( $V_o$ )

**valor.desconto.modalidade**

**valor.desconto.descontoDataFixa[n].data**

**valor.desconto.descontoDataFixa[n].valorPerc** ( $I_d$ )

O valor de “n”, em destaque amarelo nos campos relacionados a data fixa, varia de 0 a 2.

Veja-se o exemplo:

`valor.desconto.modalidade = 1` (“Valor Fixo até a data informada”)

`valor.desconto.descontoDataFixa[0].data= “2020-12-10”`

`valor.desconto.descontoDataFixa[0].valorPerc = “300.00”`

Nesse exemplo, caso o pagamento seja efetuado até o final do dia 10 de dezembro de 2020, o usuário terá um desconto de R\$ 300,00 (trezentos reais) no valor a ser pago.

#### 2.2.2.2. Desconto por dia de antecipação<sup>91</sup>

O desconto pode, alternativamente, ser informado em valor absoluto ou percentual, por dia corrido ou útil antes da data de vencimento.

Se o desconto foi informado em valor absoluto por dia de antecipação, deverá ser calculado pela fórmula a seguir:

$$V_d = V_{dd} \times D_\alpha$$

*Equação 4*

Onde:

$V_d$ : valor do desconto;

$V_{dd}$ : valor do desconto por dia de antecipação;

$D_\alpha$ : dias úteis ou corridos, conforme modalidade

Se dias corridos:  $D_\alpha = D_c$

Se dias úteis:  $D_\alpha = D_u$

$D_c$  número de **dias corridos** entre o dia de realização do cálculo e a data de vencimento. Se o resultado da subtração for menor que zero, então  $D_c$  deve ser zero:

$$D_c = \max(0; (Data_{Venc} - Data_{Calc}))$$

*Equação 5*

---

<sup>91</sup> Modalidades de desconto com domínio “3” a “6”.

Onde:

$Data_{Venc} - Data_{Calc}$ : diferença entre a data de vencimento e a data do cálculo da cobrança, em dias corridos.



Em se tratando de **dias corridos**,  $Data_{Venc}$  deverá considerar a **data original**, independentemente de ela cair em fim de semana ou feriado para o usuário pagador.

$D_u$ : número de dias úteis entre o dia de realização do cálculo e a data de vencimento. Se o resultado da subtração for menor que zero, então  $D_u$  deve ser zero:

$$D_u = \max(0; (Data_{Venc} - Data_{Calc}))$$

Equação 6

Onde:

$Data_{Venc} - Data_{Calc}$ : diferença entre a data de vencimento e a data de cálculo da cobrança, em dias úteis.

Se o desconto foi informado por percentual por dia de antecipação, deverá ser calculado pela fórmula a seguir:

$$V_d = (V_o - V_a) \times \left( \frac{I_{dd}}{100} \right) \times D_\alpha$$

Equação 7

Onde:

$V_d$ : valor do desconto;

$V_o$ : valor original da cobrança;

$V_a$ : valor de abatimento aplicável à cobrança;

$I_{dd}$ : Percentual de desconto por dia de antecipação, informado com 2 casas decimais;

$D_\alpha$ : dias úteis ou corridos, conforme modalidade

Se dias corridos:  $D_\alpha = D_c$  (vide Equação 5)

Se dias úteis:  $D_\alpha = D_u$  (vide Equação 6)

Campos JSON referenciados:

`calendario.dataDeVencimento`

`valor.original` ( $V_o$ )

`valor.desconto.modalidade`

`valor.desconto.valorPerc`

Veja-se o exemplo a seguir:

```
valor.desconto.modalidade = 3    ("Valor por antecipação dia corrido")
valor.desconto.valorPerc = "100.00"
```

Nesse exemplo, imagine que a data de vencimento é 10 de dezembro de 2020 e que o usuário realiza o pagamento em 7 de dezembro. O número de dias corridos antecipados foi 3 (10 – 7), tendo o usuário direito a um desconto de R\$ 300,00 (3 x 100 reais) no valor a ser pago. Por outro lado, se o usuário paga na data de vencimento, não houve antecipação e, portanto, o usuário não terá direito a desconto.

O campo **valor.desconto.modalidade** pode assumir os seguintes valores:

Descrição	Domínio
Valor Fixo até a data informada	1
Percentual até a data informada	2
Valor por antecipação dia corrido	3
Valor por antecipação dia útil	4
Percentual por antecipação dia corrido	5
Percentual por antecipação dia útil	6

### 2.2.3. Cálculo do valor de juros ( $V_j$ )

Os juros podem ser informados em valor absoluto, por dia corrido ou útil; podem ainda ser referenciados em percentual ao dia, mês ou ano.

#### 2.2.3.1. Valor absoluto diário:

Se os juros forem informados em valor absoluto, deverá ser calculado pela fórmula a seguir:

$$V_j = V_{jd} \times D_\alpha$$

Equação 8

Onde:

$V_j$ : valor de juros aplicável pelo atraso no pagamento da cobrança;

$V_{jd}$ : valor de juros por dia de atraso no pagamento da cobrança;

$D_\alpha$ : dias úteis ou corridos, conforme modalidade

Se dias corridos:  $D_\alpha = D_c$

Se dias úteis:  $D_\alpha = D_u$

$D_c$  número de **dias corridos** entre a data de vencimento e o dia de realização do cálculo informado pelo recebedor. Se o resultado da subtração for menor que zero, então  $D_c$  deve ser zero:

$$D_c = \max(0; (Data_{calc} - Data_{venc}))$$

Equação 9



Onde:

**$Data_{calc} - Data_{venc}$** : diferença entre a data de cálculo e a data do vencimento da cobrança, em dias corridos

**$D_u$** : número de **dias úteis** entre a data de vencimento e o dia de realização do cálculo informado pelo recebedor. Se o resultado da subtração for menor que zero, então  **$D_u$**  deve ser zero:

$$D_u = \max(0; (Data_{calc} - Data_{venc}))$$

Equação 10

Onde:

**$Data_{calc} - Data_{venc}$** : diferença entre a data de cálculo e a data do vencimento da cobrança, em dias úteis

Campos JSON referenciados:

**calendario.dataDeVencimento** ( **$Data_{venc}$** )

**valor.juros.modalidade**

**valor.juros.valorPerc** ( **$V_{jd}$** )



Se  **$Data_{venc}$**  cair em um fim de semana ou em um feriado para o usuário pagador, ela deve ser automaticamente prorrogada para o primeiro dia útil subsequente.

#### 2.2.3.2. Percentual de juros diário, mensal ou anual:

Se os juros forem informados em percentual, o valor de juros deverá ser calculado pela fórmula a seguir:

$$V_j = (V_o - V_a) \times \left( \frac{\left( \frac{I_j}{100} \right)}{n} \right) \times D_a$$

Equação 11

Onde:

**$V_j$** : valor de juros aplicável pelo atraso no pagamento da cobrança;

**$V_o$** : valor original da cobrança;

**$V_a$** : valor de abatimento aplicável à cobrança;

**$I_j$** : taxa de juros (percentual), informada com 2 casas decimais;

Se modalidade for “dias corridos”, então:

$n$ : 1, 30 ou 360, conforme a taxa de juros seja diária, mensal ou anual;

$D_{\alpha} = D_c$  (vide Equação 9).

Se modalidade for “dias úteis”, então:

$n$ : 1, 21 ou 252, conforme a taxa de juros seja diária, mensal ou anual;

$D_{\alpha} = D_u$ : (vide Equação 10).

Campos JSON referenciados:

**calendario.dataDeVencimento** ( $Data_{venc}$ )

**valor.juros.modalidade**

**valor.juros.valorPerc** ( $I_j$ )

**valor.original** ( $V_o$ )



Se  $Data_{venc}$  cair em um fim de semana ou em um feriado para o usuário pagador, ela deve ser automaticamente prorrogada para o primeiro dia útil subsequente.

O valor calculado para  $V_j$  deve ser truncado com duas casas decimais.

O campo **valor.juros.modalidade** pode assumir os seguintes valores:

Descrição	Domínio
Valor (dias corridos)	1
Percentual ao dia (dias corridos)	2
Percentual ao mês (dias corridos)	3
Percentual ao ano (dias corridos)	4
Valor (dias úteis)	5
Percentual ao dia (dias úteis)	6
Percentual ao mês (dias úteis)	7
Percentual ao ano (dias úteis)	8

#### 2.2.4. Cálculo do valor de multa ( $V_m$ )

A multa pode ser informada em valor absoluto ou como percentual da obrigação. Se a modalidade escolhida pelo recebedor for por valor fixo, então o valor da multa é o próprio valor informado pelo recebedor.

Se for informado o valor absoluto, então o valor da multa será calculado conforme a fórmula a seguir:

$$V_m = V_m \times k$$

Equação 12

Onde:

$V_m$ : valor da multa

$k(D_\alpha)$ : assume os seguintes valores, em função de  $D_\alpha$

$$k = \frac{D_\alpha}{\max(1, D_\alpha)}$$

*Equação 13*

Onde:

$D_\alpha$ : calculado conforme descrito na Equação 9 ou na Equação 10, conforme os dias sem juros sejam dias corridos ou dias úteis.

Assim, K assume o valor 0 (zero) se  $D_\alpha$  for zero; e assume o valor 1 (um), para valores positivos não nulos de  $D_\alpha$ .

Se for informado o valor percentual, então o valor da multa será calculado conforme a fórmula a seguir:

$$V_m = (V_o - V_a) \times \frac{I_m}{100} \times k$$

*Equação 14*

Onde:

$V_m$ : valor da multa;

$V_o$ : valor original da cobrança;

$V_a$ : valor de abatimento aplicável à cobrança;

$I_m$ : percentual de multa, informado com 2 casas decimais;

$k(D_a)$ : ver Equação 13.

Campos JSON referenciados:

**calendario.dataDeVencimento** ( $Data_{venc}$ )

**valor.multa.modalidade**

**valor.multa.valorPerc** ( $V_m$  ou  $I_m$ , conforme a modalidade da multa)



Se  $Data_{venc}$  cair em um fim de semana ou em um feriado para o usuário pagador, ela deve ser automaticamente prorrogada para o primeiro dia útil subsequente.

O valor calculado para  $V_m$  deve ser truncado com duas casas decimais.

O campo **valor.multa.modalidade** pode assumir os seguintes valores:

Descrição	Domínio
Valor fixo	1
Percentual	2