# Manual de Padrões para Iniciação do Pix

Versão 2.10.0

Acesse **aqui** a versão **2.11.0-rc1**

BANCO CENTRAL DO BRASIL

# Sumário


1. INTRODUÇÃO ..... 6
2. INICIAÇÃO POR QR CODE ..... 6
2.1. Informações gerais sobre iniciação via QR Code Pix ..... 6
2.2. Atores habilitados a apresentar QR Code Pix ..... 6
2.3. Conteúdo de um QR Code Pix ..... 6
2.4. Tipos de QR Codes ..... 7
*2.4.1. QR Code Estático ..... 7*
*2.4.2. QR Code Dinâmico ..... 7*
*2.4.3. QR Code Composto ..... 7*
2.5. Definições gerais em relação ao BR Code ..... 8
*2.5.1. Formatação das chaves do DICT no BR Code ..... 9*
*2.5.2. Formatação das URLs presentes nos QR Codes ..... 10*
2.6. Iniciação via QR Code Estático ..... 11
*2.6.1. Campos chave, infoAdicional e fss ..... 11*
*2.6.2. Identificador de transação: txid no QR Code estático ..... 12*
*2.6.3. Exemplo de QR Code estático ..... 12*
*2.6.4. QR Code Estático na API Pix ..... 14*
2.7. Iniciação via QR Code Dinâmico ..... 14
*2.7.1. O payload JSON ..... 15*
*2.7.2. Exemplo de QR Code Dinâmico ..... 26*
*2.7.3. Sobre cobranças concluídas ..... 28*
*2.7.4. QR Code Dinâmico na API Pix ..... 28*
2.8. Iniciação via QR Code Composto ..... 29
*2.8.1. QR Code composto apenas com os dados de recorrência ..... 29*
*2.8.2. QR Code composto com dados estáticos e recorrência ..... 29*
*2.8.3. QR Code composto com dados dinâmicos (pagamento imediato ou com vencimento) e recorrência ..... 29*
*2.8.4. O payload JSON ..... 30*
*2.8.5. Exemplos de QR Code Composto ..... 31*
3. OUTRAS FORMAS DE INICIAÇÃO ..... 38
3.1. Pix Copia e Cola ..... 38
3.2. Serviço de Iniciação de Transação de Pagamento ..... 38
3.3. Pix Automático ..... 39
*3.3.1. Jornadas de autorização ..... 39*
*3.3.2. QR Codes no Pix Automático ..... 39*
3.4. Pix por aproximação ..... 40
/ANEXO I – API PIX: CONCEITOS DE NEGÓCIO ..... 41
1. INTRODUÇÃO ..... 41
2. DOCUMENTAÇÃO DA API PIX ..... 41
3. CONTEXTO DA API PIX ..... 42
4. CONCEITOS GERAIS ..... 43
5. FUNCIONALIDADES DA API PIX ..... 44

5.1 Definições das entidades ..... 44
5.2 Cardinalidade entre as entidades ..... 47
5.3 Ciclo de vida do TransactionID (txid) ..... 47
*5.3.1 txid no contexto das Cobranças ..... 48*
*5.3.2 txid no contexto dos QR Codes Estáticos ..... 48*
5.4 Grupos de funcionalidades ..... 48
*5.4.1 Funcionalidades obrigatórias por produtos ..... 48*

**6. CASOS DE USO ..... 49**

6.1 QR Code Estático ..... 49
*6.1.1 Pagamento no ato da compra com QR Code Estático fixo e sem valor definido. ..... 49*
*6.1.2 Pagamento no ato da compra com QR Code Estático fixo com valor definido ..... 50*
*6.1.3 Pagamento no ato da compra com QR Code Estático gerado no ato da compra ..... 51*
6.2 QR Code Dinâmico ..... 51
*6.2.1 Pagamento imediato (no ato da compra) com QR Code Dinâmico ..... 51*
*6.2.2 QR Code dinâmico para pagamentos com vencimento ..... 52*
*6.2.3 Lote de Cobranças com vencimento ..... 53*
*6.2.4 O QR Code dinâmico "impresso" ..... 54*
6.3 QR Code Composto ..... 55
*6.3.1 Utilização de QR Code Composto apenas para ofertar uma recorrência ..... 55*
*6.3.2 Pagamento de QR Composto com dados estáticos e recorrência ..... 56*
*6.3.3 Pagamento imediato via QR Code Composto com dados dinâmicos e recorrência ..... 56*
*6.3.4 Pagamento com vencimento via QR Code Composto com dados dinâmicos e recorrência ..... 57*
6.4 Pix Automático ..... 58
*6.4.1 Inclusão de recorrências ..... 58*
*6.4.2 Envio de solicitação de confirmação de recorrência ..... 59*
*6.4.3 Agendamento de cobrança recorrente ..... 60*
*6.4.4 Solicitação de retentativa de pagamento de cobrança após o vencimento ..... 60*
*6.4.5 Cancelamento de agendamento de pagamento de cobrança recorrente ..... 61*
*6.4.6 Cancelamento de recorrência ..... 61*
6.5 Outros casos de uso ..... 62
*6.5.1 Efetuar uma devolução ..... 62*
*6.5.2 Remover uma Cobrança ..... 63*
*6.5.3 Alterar uma cobrança ..... 63*
*6.5.4 Configuração de Webhooks ..... 63*
*6.5.5 Configuração de Webhooks de Cobranças Recorrentes ..... 64*
*6.5.6 Configuração de Webhooks de Recorrências ..... 64*

**ANEXO II – API PIX: ESPECIFICAÇÃO TÉCNICA ..... 65**

**1. INTRODUÇÃO ..... 65**

**2. PROTOCOLOS E TECNOLOGIAS ..... 65**

**3. SEGURANÇA ..... 65**

3.1 Requisitos de segurança obrigatórios ..... 65
3.2 Recomendações de segurança ..... 66
3.3 Jornada de Adesão ..... 67

**ANEXO III – COBRANÇAS PARA PAGAMENTOS COM VENCIMENTO: CRIAÇÃO, ALTERAÇÃO E CÁLCULO ..... 68**

**1. INTRODUÇÃO ..... 68**

**2. CRIANDO UMA COBRANÇA PARA PAGAMENTO COM VENCIMENTO ..... 68**

**3. ESTRUTURA PARA CRIAÇÃO E ATUALIZAÇÃO DE UMA COBRANÇA PARA PAGAMENTO COM VENCIMENTO ..... 68**

**4. CÁLCULO DO VALOR DA COBRANÇA ..... 73**

4.1. CÁLCULO DO VALOR DE ABATIMENTO ($V_A$) ..... 73
4.2. CÁLCULO DO VALOR DE DESCONTO ($V_D$) ..... 74
*4.2.1. Fixo até a data informada ..... 74*
*4.2.2. Desconto por dia de antecipação ..... 75*
4.3. CÁLCULO DO VALOR DE JUROS ($V_J$) ..... 77
*4.3.1. Valor absoluto diário: ..... 77*
*4.3.2. Percentual de juros diário, mensal ou anual: ..... 78*
4.4. CÁLCULO DO VALOR DE MULTA ($V_M$) ..... 80

**ANEXO IV – PIX AUTOMÁTICO ..... 82**

**1. INTRODUÇÃO ..... 82**

**2. ENTIDADES ..... 82**

2.1. REC ..... 82
*2.1.1. Atributo idRec ..... 82*
*2.1.2. Atributo recebedor ..... 83*
*2.1.3. Atributo calendario ..... 83*
*2.1.4. Atributo atualizacao ..... 83*
*2.1.5. Atributo politicaRetentativa ..... 83*
*2.1.6. Atributo encerramento ..... 83*
*2.1.7. Atributo ativacao ..... 84*
2.2. SOLICREC ..... 84
*2.2.1. Atributo idSolicRec ..... 84*
*2.2.2. Atributo destinatario ..... 85*
*2.2.3. Atributo atualizacao ..... 85*
*2.2.4. Atributo recPayload ..... 85*
*2.2.5. Atributo encerramento (NR) ..... 85*
2.3. COBR ..... 85
*2.3.1. Atributo infoAdicional ..... 85*
*2.3.2. Atributo calendario ..... 85*
*2.3.3. Atributo recebedor ..... 86*
*2.3.4. Atributo politicaRetentativa ..... 86*
*2.3.5. Atributo encerramento ..... 86*
*2.3.6. Atributo devedor ..... 86*
*2.3.7. Atributo tentativas ..... 86*
*2.3.8. Atributo status x tentativas.status ..... 87*
*2.3.9. Atributo ajusteDiaUtil ..... 87*

**3. ESTADOS DAS ENTIDADES ..... 88**

3.1. ESTADOS DA REC ..... 88
3.2. ESTADOS DA SOLICREC ..... 89
3.3. ESTADOS DA COBR ..... 90
*3.3.1. Cobrança paga na primeira tentativa (data normal) ..... 90*
*3.3.2. Cobrança paga em uma retentativa ..... 91*
*3.3.3. Cobrança não paga (sem retentativas) ..... 92*
*3.3.4. Cobrança não paga (com retentativas) ..... 92*
*3.3.5. Agendamento rejeitado pelo PSP Pagador ..... 93*
*3.3.6. Cobrança cancelada pelo recebedor ..... 94*

*3.3.7. Agendamento de cobrança cancelado pelo usuário pagador ..... 95*

**4. REGRAS DE NEGÓCIO ..... 96**

4.1. Recorrência ..... 96
*4.1.1. Alteração de recorrência ..... 96*
*4.1.2. Confirmação de recorrência ..... 96*
*4.1.3. Cancelamento de recorrência ..... 96*
*4.1.4. Recorrências com retentativas ..... 96*
*4.1.5. Confirmação da recorrência na Jornada 3 ..... 97*
4.2. Solicitação de confirmação de recorrência ..... 97
*4.2.1. Alteração de solicitação de confirmação de recorrência ..... 97*
4.3. Cobrança recorrente ..... 97
*4.3.1. Data de início dos ciclos no Pix Automático ..... 97*
*4.3.2. Alteração de cobrança recorrente ..... 98*
*4.3.3. Retentativa intradia por erro na liquidação ..... 99*
*4.3.4. Retentativa após o vencimento (de acordo com a política de retentativa) ..... 99*
4.4. QR Code composto ..... 99

**5. TAGS ESPECÍFICAS PARA O PIX AUTOMÁTICO ..... 100**

**ANEXO V – MAPEAMENTO PARA MENSAGENS ISO 20022 ..... 101**

**1. INTRODUÇÃO ..... 101**

**2. MAPEAMENTO DE QR CODES PARA MENSAGEM PACS.008 ..... 102**

2.1. Mapeamento do QR Code estático para pacs.008 ..... 102
2.2. Mapeamento do QR Code dinâmico para pacs.008 ..... 103
2.3. Mapeamento do Serviço de Iniciação de Transação de Pagamento para pacs.008 ..... 105

**3. MAPEAMENTO PARA AS MENSAGENS DO PIX AUTOMÁTICO ..... 106**

3.1. Mapeamento dos dados de recorrência para pain.009 ..... 107
3.2 . Mapeamento dos dados de recorrência para pain.012 ..... 108
3.3 . Mapeamento dos dados do cancelamento da recorrência ou da solicitação de confirmação da recorrência (pain.009) para pain.011 ..... 110
3.4 . Mapeamento de uma cobrança recorrente para pain.013 ..... 112
3.5 . Mapeamento da resposta à solicitação de um agendamento (pain.014) ..... 114
3.6 . Mapeamento de um cancelamento de um agendamento (camt.055) ..... 114
3.7 Mapeamento da resposta a um cancelamento de um agendamento (camt.029) ..... 115

**ANEXO VI – ARQUIVO PADRONIZADO DO PIX AUTOMÁTICO ..... 117**

**ANEXO VII – PRAZOS PARA IMPLEMENTAÇÃO DAS FUNCIONALIDADES ..... 118**

# 1. Introdução

Este documento, parte integrante do Regulamento do Pix, lista e detalha as funcionalidades de iniciação de pagamentos do arranjo de pagamentos Pix.

# 2. Iniciação por QR Code

## 2.1. Informações gerais sobre iniciação via QR Code Pix

O BR Code é o padrão usado para geração de QR Codes no Brasil. Adota o padrão EMV para uso de QR Codes em sistemas de pagamento (EMV-QRCPS).

O padrão EMV trabalha com dois fluxos distintos: o QR Code apresentado pelo recebedor (*Merchant Presented Mode - MPM*)[1] e o QR Code apresentado pelo pagador (*Consumer Presented Mode - CPM*)[2]. O BR Code trata apenas do **primeiro**.

O QR Code Pix, na qualidade de mecanismo para envio ou disponibilização prévia de informações para fins de iniciação de um Pix, seguirá o padrão do BR Code, nos termos do Manual do BR Code[3].

## 2.2. Atores habilitados a apresentar QR Code Pix

Em se tratando de emissão de QR Code Pix, há alguns atores habilitados:

1) O Prestador de Serviços de Pagamento (PSP) do recebedor; e
2) A Secretaria do Tesouro Nacional (STN).

## 2.3. Conteúdo de um QR Code Pix

Em um QR Code Pix, o usuário recebedor pode apresentar até dois tipos de informação: dados de pagamento ou dados de recorrência (contendo parâmetros para pagamentos recorrentes). Posteriormente, o usuário pagador pode capturar o QR Code e seguir para os fluxos correlatos aos dados capturados.

Instalado no dispositivo móvel e utilizado para a leitura do QRCode, o aplicativo do PSP do usuário pagador acessará o serviço de *backend*[4] disponibilizado pelo PSP do recebedor. Este é responsável por gerar a ordem de pagamento, a solicitação de recorrência, ou ambos.

---

[1] Disponível em <https://www.emvco.com/terms-of-use/?u=/wp-content/uploads/documents/EMVCo-Merchant-Presented-QR-Specification-v1-1.pdf>.
[2] Disponível em <https://www.emvco.com/terms-of-use/?u=/wp-content/uploads/documents/EMVCo-Consumer-Presented-QR-Specification-v1-1.pdf>.
[3] Disponível em <https://www.bcb.gov.br/estabilidadefinanceira/arranjosintegrantesspb>.
[4] 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.

## 2.4. Tipos de QR Codes

Abaixo listamos de forma resumida cada um dos tipos de QR Codes existentes no âmbito do Pix, e nas seções seguintes descreveremos em detalhes cada um deles com exemplos.

### *2.4.1. QR Code Estático*

O QR Code estático apresenta um rol de funcionalidades restrito. Neste tipo de QR Code, todos os dados necessários à iniciação do pagamento (configurações) estão completamente contidos dentro do próprio QR Code.

São apenas cinco opções para configuração. A primeira configuração é obrigatória, o usuário que gera o QR Code estático precisa informar uma chave válida no Diretório de Identificadores de Contas Transacionais (DICT), nos termos do Manual Operacional do DICT[5]. As outras quatro configurações são opcionais: i) identificador da transação[6], ii) campo de texto livre, iii) valor do pagamento, e iv) identificação do facilitador de serviço de saque, e são preenchidas de acordo com o caso de uso desejado.

### *2.4.2. 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[7]. O QR Code dinâmico também deve, necessariamente, 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[8]. O QR Code dinâmico contém somente as informações básicas do usuário recebedor. O restante das informações é obtido no *endpoint* do PSP do recebedor indicado por esta URL.

### *2.4.3. QR Code Composto*

O QR Code composto também dispõe de um rol de funcionalidades abrangente. Além das funcionalidades presentes no QR Code Estático e QR Code dinâmico, à exceção do Pix Saque e Pix Troco, também está presente uma definição estruturada dos elementos relacionados aos parâmetros

---

[5] 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.

[6] 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.008.

[7] O PSP do recebedor 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.

[8] Um QR Code com muitas informações pode dificultar ou inviabilizar a decodificação dos dados a partir da imagem, que se torna muito densa.

para pagamentos recorrentes[9]. A característica que distingue claramente o QR Code composto é a existência de uma URL adicional, em campo específico no BR Code, cujo *payload* contém as informações relativas aos aspectos de configuração de recorrência de pagamentos oferecido pelo usuário recebedor ao usuário pagador.

O QR Code composto, em sua estrutura interna, além da possibilidade de indicar uma URL relacionada a um pagamento, como acontece no QR Code dinâmico, ou das informações de chave DICT, como é o caso do QR Estático, também é configurado com outra URL que aponta para os dados de configuração da recorrência. Essa funcionalidade abre diversas possibilidades de uso, dado que as informações trazidas pela URL de pagamento, quando presente, e das configurações de recorrência podem variar em função de diversos parâmetros conforme os casos de uso desejados.

Pode-se considerar o QR Code composto como uma extensão dos QR Codes já conhecidos adicionando-se a característica de repetição, incluindo a dimensão **recorrência**. Entretanto é importante observar que o QR Code composto, como veremos adiante, permite também apenas a apresentação da recorrência, sem necessidade de preenchimentos da chave DICT ou da URL de *payload*, como ocorrem no QR Code estático e QR Code dinâmico, respectivamente.

A URL com os parâmetros da recorrência, assim como a URL do pagamento, cumpre o papel de reduzir a quantidade de dados codificados diretamente na imagem (no QR Code). O QR Code composto também contém as informações básicas do usuário recebedor. O restante das informações é obtido nos *endpoints* do PSP do recebedor indicados pelas URLs do pagamento, dos parâmetros da recorrência ou por ambas.

Os formatos dessas URLs (os dois casos são similares) estão detalhados na seção 2.5.2 deste manual. Os detalhes sobre segurança relacionados ao QR Code estão detalhados no Manual de Segurança do SFN.

## 2.5. 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 - *Globally Unique Identifier* | **br.gov.bcb.pix**[10] | **14 caracteres** |

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.

[9] O PSP do recebedor 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 uma solicitação de recorrência.

[10] O GUI – DNS reverso – é *case insensitive*.

| ID | Nome EMV | Tamanho | Uso[11] | Descrição | | | | |
|---|---|---|---|---|---|---|---|---|
| 26..51[12] | *Merchant Account Information* | 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[13] | *Unreserved Templates* | 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, dinâmico ou composto), quando o *GUI* corresponder a **br.gov.bcb.pix.** Em especial, veremos que para o QR Code composto o grupo de *ids* 80..99 do BR Code deixa de ser opcional e torna-se obrigatório. Em particular, apresenta o *endpoint* para os dados com os parâmetros de recorrência.

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 20.022.

### *2.5.1. 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, conforme os exemplos abaixo.

| Tipo de Chave | Exemplo |
|---|---|
| *E-mail* | codificado no seguinte formato:<br>fulano_da_silva.recebedor@example.com |
| *CPF ou CNPJ* | codificados nos seguintes formatos:<br>CPF: 12345678900<br>CNPJ: 00038166000105<br>CNPJ: 12ABC34501DE35 |
| *Número de telefone celular* | codificado seguindo o formato internacional:<br>+5561912345678<br>em que:<br>+55: código do país,<br>61: código do território ou estado,<br>912345678: número do telefone celular. |
| *Chave aleatória* | codificada juntamente com a pontuação, como segue:<br>123e4567-e12b-12d1-a456-426655440000 |

[11] M – Mandatório; O – opcional.
[12] O ID pode ser qualquer um dos identificadores dentro da faixa 26-51.
[13] O ID pode ser qualquer um dos identificadores dentro da faixa 80-99.

### *2.5.2. Formatação das URLs presentes nos QR Codes*

As URLs presentes nos QR Codes, também chamadas de "*locations*", não devem incluir prefixo de protocolo. O acesso deverá ser realizado após[14] validações[15], incluindo o domínio válido autorizado pelo PSP do recebedor para geração de QR Codes, exclusivamente via HTTPS[16].

Veremos que temos *locations* para os dados de cobrança, imediata ou com vencimento, no grupo 26.25 do QR Code dinâmico, e para os parâmetros de recorrência, no grupo 80.25 do QR Code composto.

Respeitadas as regras de formação de URL[17], a URL terá esse layout:

**{fqdnPspRecebedor}/{pixEndpoint}/{urlAccessToken}**

Onde:

- **fqdnPspRecebedor:** *Fully Qualified Domain Name* do Provedor de Serviços de Pagamento da ponta Recebedora.
- **pixEndpoint:** será utilizado pelo PSP recebedor para organizar[18] suas URLs e definir o tipo de recurso que está sendo apresentado. Está estruturado da seguinte maneira:
  **- {endpointOpcional}/{fragmentoTipoRecurso}**
  Onde:
  - **endpointOpcional:** fragmento opcionalmente utilizado pelo PSP recebedor para fins de organização de URLs em um domínio. Por exemplo "/pix/123/". **Pode ser vazio**;
  - **fragmentoTipoRecurso:** define o tipo de recurso acessado. Temos recursos acessíveis relacionados às cobranças e relacionados aos parâmetros de recorrência. Desta forma temos as seguintes convenções:
    - **sem este fragmento, ou com "/cob/" (opcional):** leva a *payloads* de cobranças imediatas;
    - **"/cobv/":** leva a *payloads* de cobranças com vencimento;
    - **"/rec/":** leva a *payloads* de parâmetros de recorrência.
- **urlAccessToken:** fragmento que incorpora uma característica de segurança à *location*, tornando-a difícil de ser pressuposta[19]. Na especificação da Api Pix nos referimos a esse trecho como *pixUrlAccessToken* para as *locations* de cobranças, e *recUrlAccessToken* para as *locations* de recorrência.

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

---

[14] Todas as URLs referenciadas nos QR Codes (i.e. as de cobrança e as de recorrência) devem ser completamente avaliadas quanto à presença de caracteres inválidos e suas partes constituintes (prefixo, domínio, aplicação e demais fragmentos) antes de qualquer tentativa de acesso.

[15] 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.

[16] Restrições adicionais de segurança e outros detalhes estão definidos do Manual de Segurança do SFN.

[17] Ver <https://www.ietf.org/rfc/rfc3986.txt>.

18 Futuramente, a versão da API Pix também constará nesse fragmento.

[19] Em outras palavras, torna a *location* uma "URL de capacidade". Mais informações sobre esta questão encontram-se no Manual de Segurança do SFN.

Qualquer PSP pode hospedar URLs de QR Codes 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.

Para mais informações a respeito de segurança e para mais detalhes, pode-se consultar o Manual de Segurança do SFN.

## 2.6. Iniciação 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*[20] |
| 3 | Conjunto livre de caracteres, com limite de tamanho (infoAdicional) | Opcional |
| 4 | *Identificador da transação "TransactionIdentification <TxId>"* | *BR Code ID62-05*[21] |
| 5 | Facilitador de serviço de saque | Opcional |

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 [22] | M | Chave Pix |
| | 02 | infoAdicional[23] | 01..72 [24] | O | Conjunto livre de caracteres com limite de tamanho. |
| | 03 | fss | 08 | O | ISPB do facilitador de serviço de saque |

### *2.6.1. Campos chave, infoAdicional e fss*

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 os campos *infoAdicional* e *fss* não forem utilizados. Não é possível que os campos *chave* e *infoAdicional* cheguem simultaneamente a seus tamanhos máximos potenciais, menos ainda se também houver preenchimento do campo *fss*, uma vez que esse campo consome 12 (2+2+8) caracteres. Um exemplo ilustra melhor a situação. Supondo uma chave de tamanho 9, temos:

---

[20] *Transaction Amount*.
[21] *Additional Data Field – Reference Label,* sempre presente em um BR Code.
[22] Máximo Chave: 99 – 8 (Identificador ID[2] + indicador de tamanho de GUI[2] + Identificador ID[2] + indicador do tamanho de chave[2]) – 14 (tamanho do valor em GUI): 77
[23] 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.
[24] Máx. Referência: 99 – 27 (GUI[4+14] + Chave Mínima[5] + ID[2] e tamanho do *infoAdicional*[2]): 72

- Identificador ID + indicador de tamanho da GUI: 4
- Identificador ID + indicador de tamanho da *chave*: 4
- Tamanho do valor no campo GUI: 14
- Tamanho do valor no campo *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) para o campo *infoAdicional*, restando, desta forma, 64 caracteres de texto livre. Se a chave fosse maior, menor seria o espaço destinado ao texto livre. Em outras palavras, os campos: *chave*, *infoAdicional* e *fss* disputam o espaço de 99 caracteres do ID raiz (na faixa 26-51).

Convém observar que no exemplo acima, se houver o uso do campo *fss*, restariam apenas 68 – 12 (identificador do campo *fss* [2] + indicador tamanho do campo [2] + valor do campo [8]) = 56 caracteres para o campo *infoAdicional* incluindo o ID e tamanho, ou seja, 56 – 4 = 52 caracteres para o valor em *infoAdicional*.

A presença do campo *fss*, com um ISPB válido junto às regras do arranjo Pix, no QR Code, indica que esse é um QR Code para Pix Saque. **Não há funcionalidade de Pix Troco para QR Codes estáticos**, apenas para QR Codes dinâmicos.

### *2.6.2. 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[25], 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'

### *2.6.3. 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 campo *fss* também não é utilizado. O QR Code foi gerado com uma chave aleatória.

[25] 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.

**Nome do Recebedor** : Fulano de Tal [26]
**Chave Pix** : 123e4567-e12b-12d1-a456-426655440000
**Valor** : não informado

| ID | Nome BR Code | Tam | Valor | | | |
|---|---|---|---|---|---|---|
| **00** | *Payload Format Indicator* | **02** | **01** | | | |
| **26** | *Merchant Account Information*[27] | **58** | **ID** | **Nome** | **Tam** | **Valor** |
| | | | **00** | *GUI* | **14** | **br.gov.bcb.pix** |
| | | | **01** | chave | **36** | **123e4567-e12b-12d1-a456-426655440000** |
| **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 Template* | **07** | **ID** | **Nome** | **Tam** | **Valor** |
| | | | **05** | **txid** | **03** | *** |
| **63** | *CRC16*[28] | **04** | 0x***1D3D*** – 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:

```
00 02 01
26 58
      00 14 br.gov.bcb.pix
      01 36 123e4567-e12b-12d1-a456-426655440000
52 04 0000
53 03 986
58 02 BR
59 13 Fulano de Tal
60 08 BRASILIA
62 07
      05 03 ***
63 04 1D3D
```

[26] O nome a apresentar ao pagador será necessariamente o nome retornado na consulta ao DICT. O *MerchantName* será ignorado pelo pagador.
[27] 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.
[28] Polinômio 0x1021, C.I. 0xFFFF. A ordem dos objetos modifica o CRC.

O respectivo QR Code estático está abaixo:



00020126580014br.gov.bcb.pix0136123e4567-e12b-12d1-a456-426655440000520400005303986580 2BR5913Fulano de Tal6008BRASILIA62070503***63041D3D

### *2.6.4. QR Code Estático na API Pix*

O QR Code 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* e *fss*. A criação[29] 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 e que estejam associados a um *txid* podem ser consultados via API Pix. Mais 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.

## 2.7. Iniciação via QR Code Dinâmico

O QR Code dinâmico conterá as seguintes informações:

| QR *Code* Dinâmico | | |
|---|---|---|
| # | Campo | Tipo |
| *1* | *Valor* | *BR Code ID54*[30] |
| *2* | *Identificador da transação "TransactionIdentification <TxId>"* | *BR Code ID62-05*[31] |
| 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.

---

[29] 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.

[30] *Transaction Amount*

[31] *Additional Data Field – Reference Label*

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[32], quando possível. 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 | **URL** | 01..77 | M | Link para *payload* JSON |

### *2.7.1. 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.

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

#### 2.7.1.1. *Payload* da cobrança para **pagamentos imediatos**

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 recebedor devolve diretamente no *payload* os dados relativos à cobrança para pagamento.

[32] A consistência de campos entre múltiplos arranjos de pagamento com BR Code está fora do escopo desta especificação.

A estrutura do *payload* JSON de 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 | |
| 4.3.1 | Valor do saque | [0..1] | `valor.retirada.saque.valor` | String | OU( saque |
| 4.3.2 | Modalidade de alteração de valor do saque | [0..1] | `valor.retirada.saque.modalidadeAlteracao` | Number | saque |
| 4.3.3 | ISPB do facilitador de serviço de saque | [0..1] | `valor.retirada.saque.prestadorDoServicoDeSaque` | String | saque |
| 4.3.4 | Modalidade do agente | [0..1] | `valor.retirada.saque.modalidadeAgente` | String | saque |
| 4.4.1 | Valor do troco | [0..1] | `valor.retirada.troco.valor` | String | troco |
| 4.4.2 | Modalidade de alteração de valor do troco | [0..1] | `valor.retirada.troco.modalidadeAlteracao` | Number | troco |
| 4.4.3 | ISPB do facilitador de serviço de saque | [0..1] | `valor.retirada.troco.prestadorDoServicoDeSaque` | String | troco |
| 4.4.4 | Modalidade do agente | [0..1] | `valor.retirada.troco.modalidadeAgente` | String | troco ) |
| 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>*[33] | |
| 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 mais detalhes técnicos, **a API Pix[34] é a referência indicada**.

[33] Detalhes de segurança estão definidos no Manual de Segurança do SFN.
[34] Disponível em <https://github.com/bacen/pix-api>

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

O campo **revisao** deve ser incrementado somente quando forem realizadas alterações na cobrança, via *endpoints* **PATCH** ou **PUT**, conforme especificados na API Pix. Adicionalmente, via API Pix, é possível acessar as revisões anteriores da cobrança.

- **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[35]. 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.

[35] UTC: *Coordinated Universal Time*. Mais detalhes em <https://en.wikipedia.org/wiki/Coordinated_Universal_Time>.

- **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.

- **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. Este campo obedece às seguintes restrições:
  - Para cobranças imediatas que não envolvam saque ou troco: deve apresentar valores maiores do que zero, exceto no caso de o campo **valor.modalidadeAlteracao** apresentar valor 1;
  - Para cobranças imediatas que representem um saque: deve apresentar o valor 0.00 (zero). O valor do saque será preenchido no campo '**valor.retirada.saque.valor**';
  - Para cobranças imediatas que sejam com troco: deve apresentar necessariamente valor maior que 0.00 (zero) referente à compra; o valor do troco será preenchido no campo '**valor.retirada.troco.valor**'.
- **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 **valor.modalidadeAlteracao** apresente o valor 1, então o usuário pagador poderá alterar o valor da cobrança. Ainda no caso de o campo **valor.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 zero[36].
- **valor.retirada.saque**: [opcional] indica o valor do saque a ser realizado, no campo **valor.retirada.saque.valor** e, se é possível ou não sua modificação pelo usuário quando da leitura do QR Code, no campo **valor.retirada.saque.modalidadeAlteracao**. O campo **valor.retirada.saque.modalidadeAlteracao** admite dois valores: 0 (zero) quando a alteração não for permitida ou 1 (um) quando a alteração for permitida; na ausência do campo nesta estrutura considera-se que **valor.retirada.saque.modalidadeAlteracao** assume o valor 0 (zero). Convém observar que a estrutura **valor.retirada.saque** é opcional mas, **uma vez preenchida (valor ou modalidade)**, as regras seguintes se aplicam:
  - **É obrigatório** o preenchimento dos campos:
    - **valor.retirada.saque.modalidadeAgente:** indica a modalidade do agente por meio da qual se dá a facilitação do serviço de saque, quais sejam, via estabelecimento comercial (AGTEC), outra espécie de pessoa jurídica que tenha como atividade principal ou secundária a

[36] O PSP do pagador não deve permitir a confirmação de um pagamento cuja soma do valor.original + (valor.retirada.saque.valor[se existir] ou valor.retirada.troco.valor[se existir]) seja zero.

prestação de serviços auxiliares a serviços financeiros ou afins ou correspondente no País (AGTOT) ou diretamente pelo facilitador de serviço de saque (AGPSS[37]), conforme especificado na API;
        - **`valor.retirada.saque.prestadorDoServicoDeSaque`:** indica o ISPB do facilitador de serviço de saque.
    - O valor em **`valor.original`** deve ser **sempre igual a 0.00 (zero)**;
    - O valor em **`valor.modalidadeAlteracao`** deve **ser sempre igual a 0 (zero)** – sem possibilidade de alteração - explicitamente, ou implicitamente 0 (zero) quando não preenchido;
    - O campo **`valor.retirada.saque.valor`**:
        - Deve ser maior que 0.00 (zero) se **`valor.retirada.saque.modalidadeAlteracao`** for 0 (zero). Neste cenário o usuário não pode alterar o **`valor.retirada.saque.valor`** informado no QR Code, portanto ele deve ser indicado previamente;
        - Pode ser 0.00 (zero) se **`valor.retirada.saque.modalidadeAlteracao`** for 1, ou ter um valor preenchido diferente de 0.00 (zero) que serve como valor sugerido inicial de saque e pode ser alterado pelo usuário, assim como explicado para o campo **`valor.original`**. A restrição da *app mobile* de evitar preenchimento com 0.00 (zero) continua não permitindo saques com valor 0.00 (zero).

- **`valor.retirada.troco`**: [opcional] que indica o valor do troco a ser realizado, no campo **`valor.retirada.troco.valor`** e, se é possível ou não sua modificação pelo usuário quando da leitura do QR Code, no campo **`valor.retirada.troco.modalidadeAlteracao`**. O campo **`valor.retirada.troco.modalidadeAlteracao`** admite dois valores: 0 (zero) quando a alteração não for permitida ou 1 (um) quando a alteração for permitida; na ausência do campo nesta estrutura considera-se que **`valor.retirada.troco.modalidadeAlteracao`** assume o valor 0 (zero). Convém observar que a estrutura **`valor.retirada.troco`** é opcional mas, **uma vez preenchida (valor ou modalidade)**, as regras seguintes se aplicam:
    - **É obrigatório** o preenchimento dos campos:
        - **`valor.retirada.troco.modalidadeAgente`:** indica a modalidade do agente por meio da qual se dá a facilitação do serviço de saque; no caso do Pix Troco, estará sempre restrita ao estabelecimento comercial (AGTEC) ou ao correspondente no País (AGTOT), conforme especificado na API Pix;
        - **`valor.retirada.troco.prestadorDoServicoDeSaque`:** indica o ISPB do facilitador de serviço de saque.
    - O valor em **`valor.original`** deve ser **sempre maior que 0.00 (zero)**;
    - O valor em **`valor.modalidadeAlteracao`** deve ser **sempre igual a 0 (zero)** - sem possibilidade de alteração - explicitamente ou implicitamente 0 (zero) quando não preenchido;
    - O campo **`valor.retirada.troco.valor`**:

[37] Observação: no mapeamento para o campo 'modalidadeAgente', da pacs.008, esse valor deve ser substituído por AGFSS.

- Deve ser maior que 0.00 (zero) se **`valor.retirada.troco.modalidadeAlteracao`** for 0 (zero). Neste cenário, o usuário não pode alterar o **`valor.retirada.troco.valor`** informado no QR Code, portanto ele deve ser indicado previamente;
- Pode ser 0.00 (zero) se **`valor.retirada.troco.modalidadeAlteracao`** for 1, ou ter um valor preenchido diferente de 0.00 (zero) que serve como sugestão inicial e pode ser alterado pelo usuário assim como explicado para o campo **`valor.original`**. A restrição da *app mobile* de evitar preenchimento com 0.00 (zero) continua, não permitindo trocos com valor 0.00 (zero).

Sobre as estruturas **`valor.retirada.saque`** e **`valor.retirada.troco`** não é possível que ambas sejam fornecidas em conjunto, são estruturalmente e mutuamente excludentes, quando há "**`saque`**" não há "**`troco`**" e vice-versa.

Todos os campos que indicam valores monetários obedecem ao pattern ou regex **\d{1,10}\.\d{2}**. Exemplos de valores aderentes ao padrão seriam: "0.00", "1.00", "123.99", "123456789.23".

Em síntese:

| | | **Valor** | | | | | |
|---|---|---|---|---|---|---|---|
| | | original | modalidade Alteracao | Retirada | | | |
| | | | | Saque | | Troco | |
| | | | | valor | modalidade Alteracao | valor | modalidade Alteracao |
| **Pix Cobrança para pagamento imediato** | Com alteração de valor da cobrança / compra | 0 ou valor sugerido | 1 | - | - | - | - |
| | Sem alteração de valor da cobrança / compra | Valor da cobrança ou compra | 0 | - | - | - | - |
| **Pix Saque** | Com alteração do valor do saque | 0 | 0 | 0 ou valor sugerido do saque | 1 | - | - |
| | Sem alteração do valor do saque | 0 | 0 | Valor do saque | 0 | - | - |
| **Pix Troco (não admite alteração de valor da compra)** | Com alteração do valor do troco | Valor da compra | 0 | - | - | 0 ou valor sugerido do troco | 1 |
| | Sem alteração do valor do troco | Valor da compra | 0 | - | - | Valor do troco | 0 |

Convém observar que na estrutura da tabela acima, por simplicidade, os campos **`modalidadeAgente`** e **`prestadorDoServicoDeSaque`**, que são obrigatórios, foram omitidos.

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

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** 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, normalmente[38] é criado pelo usuário recebedor e encontra-se sob sua responsabilidade. 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 tópico Ciclo de vida do TransactionID (txid), do Anexo I.

- **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[39] 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.

[38] No caso das cobranças para pagamento imediato, a criação do *txid* poderá ser delegada ao PSP do recebedor.

[39] 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*.

- **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[40].

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**

Todos os campos, excluído o campo **`assinatura`**, compõem um *payload* JSON assinado. Mais 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", "CONCLUÍDA", "REMOVIDO_PELO_USUARIO_RECEBEDOR" e "REMOVIDO_PELO_PSP".

#### 2.7.1.2. *Payload* para cobrança para **pagamentos com vencimento**

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[41] (`codMun`) do usuário pagador e a Data de Pagamento Pretendida (`DPP`). Estas informações são importantes para que o PSP do recebedor 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.

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

[41] Esta informação corresponde ao município que consta na informação cadastral de endereço do usuário pagador, baseado na Tabela de Códigos de Municípios do IBGE, que apresenta a lista dos municípios brasileiros associados a um código composto por 7 dígitos, sendo os dois primeiros referentes ao código da Unidade da Federação.

Pontos de atenção:

1. Caso a informação referente ao município não seja enviada; o PSP do recebedor assumirá que não existem feriados estaduais e municipais no período em questão;
2. Caso a informação referente à Data de Pagamento Pretendida (DPP) não seja enviada; o PSP do recebedor procederá da seguinte forma para o cálculo da cobrança:
   - 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).

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.

A estrutura do *payload* JSON de 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 em dias corridos | [1..1] | `calendario.validadeAposVencimento` | 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 | [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 | [1..1] | `recebedor.logradouro` | String | |

| | | | | | |
|---|---|---|---|---|---|
| 6.6 | Cidade do recebedor | [1..1] | `recebedor.cidade` | String | |
| 6.7 | Unidade da federação do recebedor | [1..1] | `recebedor.uf` | String | |
| 6.8 | CEP do recebedor | [1..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>*[42] | |
| 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 mais detalhes técnicos, **a API Pix[43] é a referência indicada**.

- **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`**[44];
- **`calendario.validadeAposVencimento`**: [obrigatório] (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[45].

- **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.

[42] Detalhes de segurança estão definidos no Manual de Segurança do SFN.

[43] Disponível em <https://github.com/bacen/pix-api>

[44] 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.

[45] 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**

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 pattern ou regex **\d{1,10}\.\d{2}**. Exemplos de valores aderentes ao padrão seriam: “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**:

- **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.

- **Status** (NR)

O campo **`status`** representa a situação da cobrança para pagamentos com vencimento, podendo assumir os seguintes estados: "ATIVA", "CONCLUÍDA", "REMOVIDO_PELO_USUARIO_RECEBEDOR", "REMOVIDO_PELO_PSP".

### *2.7.2. 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** [46] : Fulano de Tal
**URL do PSP do recebedor** : pix.example.com/8b3da2f39a4140d1a91abd93113bd441
**Valor** : R$123,45[47]

| 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) [48] | | | |
| **26** | *Merchant Account Information* | **70** | **ID** | **Nome** | **Tam** | **Valor** |
| | | | **00** | *GUI* | **14** | **br.gov.bcb.pix** |
| | | | **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** | 0x***64E4*** | | | |

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

[46] O nome que deve ser apresentado ao pagador é o nome do usuário recebedor conforme registrado no DICT.
[47] Este valor será lido dentro do *payload* indicado pela URL da cobrança.
[48] 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).

```
00 02 01
01 02 12
26 70
      00 14 br.gov.bcb.pix
      25 48 pix.example.com/8b3da2f39a4140d1a91abd93113bd441
52 04 0000
53 03 986
58 02 BR
59 13 Fulano de Tal
60 08 BRASILIA
62 07
      05 03 ***
63 04 64E4
```

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[49] com os dados do pagamento:

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

Verificado o domínio "pix.example.com" como autorizado[50] 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*

[49] Deve-se obedecer aos padrões definidos pela RCF 7515. Em especial, respeitando-se o tipo de conteúdo como sendo '*application/jose*' <https://datatracker.ietf.org/doc/html/rfc7515#section-9.2.1>

[50] Conforme Manual de Segurança do SFN.

Se (e somente se) **JWS-AssinaturaVálida**[51] **(*JWSHeader, JWSPayload, JWSSignature, PSPRECPUBKEY*)**, procede-se ao *parsing* dos dados da estrutura JSON:

JSONPayload = base64url[52]-decode **(*JWSPayload*)**.

Um exemplo de payload JSON pode ser encontrado na API Pix[53].

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
Expira em: 31/03/2020 17:00:00[54]
Detalhes do Pagamento: **Informação Adicional do PSP do Recebedor**
**Confirma?**

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

### *2.7.3. Sobre cobranças concluídas*

A critério do PSP recebedor, *payloads* que representem cobranças já concluídas, expiradas ou removidas podem retornar um HTTP *status* que apresente semântica adequada. Por exemplo, o PSP recebedor pode optar por retornar o HTTP status 410 '*gone*' ou ainda 404 '*not found*'.

### *2.7.4. QR Code Dinâmico na 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[55]. 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.
O QR Code Dinâmico, em um contexto de integração automatizada[56], 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*[57] que será responsável por retornar as informações relativas à cobrança que o QR Code Dinâmico busca representar.

A API Pix **NÃO** retorna um arquivo com o QR Code gerado; desta forma, estas funcionalidades 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.

---

[51] Conforme Manual de Segurança do SFN.
[52] 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.
[53] https://github.com/bacen/pix-api
[54] Note-se que o *payload* utiliza UTC.
[55] 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.
[56] Em outras palavras, fora do cenário de criação do QR Dinâmico por meio do aplicativo do PSP recebedor.
[57] Uma URL que ao acessada retorna como resposta um conjunto de informações, também chamada de "location".

Para mais informações a respeito da dinâmica de gerenciamento de cobranças, pode-se consultar a API Pix[58].

## 2.8. Iniciação via QR Code Composto

O QR Code composto pode apresentar três configurações, a depender do caso de uso, que são descritas a seguir.

### *2.8.1. QR Code composto apenas com os dados de recorrência*

| QR *Code* Composto | | |
|---|---|---|
| **#** | **Campo** | **Tipo** |
| *1* | *Identificador da transação "TransactionIdentification <TxId>"* | *BR Code ID62-05*[59] |
| *2* | *Link URL dos parâmetros de recorrência* | Obrigatório |

Neste cenário de uso o QR Code composto só precisa apresentar a URL com os parâmetros de recorrência.

### *2.8.2. QR Code composto com dados estáticos e recorrência*

| QR *Code* Composto | | |
|---|---|---|
| **#** | **Campo** | **Tipo** |
| *1* | *Valor* | *BR Code ID54*[60] |
| *2* | *Identificador da transação "TransactionIdentification <TxId>"* | *BR Code ID62-05*[61] |
| *3* | *Link URL dos parâmetros de recorrência* | Obrigatório |

Neste cenário, o QR Code composto pode ser visto como um QR Code estático estendido com as funcionalidades de recorrência.

### *2.8.3. QR Code composto com dados dinâmicos (pagamento imediato ou com vencimento) e recorrência*

| QR *Code* Composto | | |
|---|---|---|
| **#** | **Campo** | **Tipo** |
| *1* | *Identificador da transação "TransactionIdentification <TxId>"* | *BR Code ID62-05*[62] |
| *2* | *Link URL do pagamento* | Obrigatório |
| *3* | *Link URL dos parâmetros de recorrência* | Obrigatório |

Neste cenário, o QR Code composto pode ser visto como um QR Code dinâmico estendido, onde o campo *Link URL do pagamento* representa uma URL que será utilizada para recuperação dos dados

---

[58] https://github.com/bacen/pix-api
[59] *Additional Data Field – Reference Label*
[60] *Transaction Amount*
[61] *Additional Data Field – Reference Label*
[62] *Additional Data Field – Reference Label*

que fazem parte do pagamento e o campo *Link URL dos parâmetros de recorrência* representa uma URL que será utilizada para recuperação dos dados que parametrizam a recorrência. Nestes casos, com duas URLs presentes no QR Code Composto, temos a restrição adicional de que o *fqdnPspRecebedor* (vide seção 2.5.2) destas duas URLs deve ser idêntico, o que reduz as chances de problemas relacionados aos aspectos de segurança, como discrepância de certificados.

Observe-se que no arranjo Pix, como um mesmo BR Code composto pode ser reutilizado para vários pagamentos, *valor* e *txid* obtidos via *Link URL do pagamento* podem mudar a cada transação. Assim, esses campos BR Code devem ser ignorados pelo pagador em qualquer caso de uso de QR Code composto que tenha URL de pagamento. Da mesma forma, em cada reutilização do QR Code composto, os dados presentes em *Link URL dos parâmetros de recorrência* também podem mudar.

É importante destacar, portanto, que quaisquer dados retornados pelos *Links de recorrência e pagamento* terão prioridade sobre os campos específicos do formato EMV, como *Valor* ou *Identificador da Transação.* Esses campos podem ter os mesmos valores do *payload* JSON[63], quando possível.

Além do mapeamento de campos já conhecidos para o QR Code estático e QR Code dinâmico, o QR Code composto utiliza o campo a seguir:

| **ID** | ***Merchant Account Information*** | | | | |
|---|---|---|---|---|---|
| | **ID** | **Nome** | **Tam** | **Uso** | **Descrição** |
| **80..99** | 00 | *GUI* | 14 | M | **br.gov.bcb.pix** |
| | 25 | **URL** | 01..77 | **M** | Link para *payload* JSON dos parâmetros de recorrência. |

O grupo de IDs 80..99 será o responsável por informar o *endpoint* do *payload* JSON relacionado ao QR Code composto quanto aos parâmetros de recorrência. Para esta nova URL aplicam-se todos os requisitos de segurança relativos à URL do grupo 26..51 já descrito para o QR Code dinâmico.

### *2.8.4. O payload JSON*

Para o QR Code composto, dois *payloads* podem ser lidos: i) o relativo ao pagamento, quando existir, e ii) o relativo à configuração de recorrências. Abaixo veremos cada um deles em detalhes.

#### 2.8.4.1. O *payload* JSON do pagamento

O *payload* JSON retornado pelo *endpoint* indicado em *Link URL do pagamento* obedece à estrutura já definida na seção 'O payload JSON' correlata do QR Code dinâmico, para pagamentos imediatos e com vencimento. Entretanto, aqui, temos a expressa restrição de que para pagamentos imediatos, iniciados por QR Codes compostos, não é permitida a presença da estrutura `valor.retirada`, que caracteriza os produtos Pix Saque e Pix Troco.

[63] A consistência de campos entre múltiplos arranjos de pagamento com BR Code está fora do escopo desta especificação.

Em outras palavras, **os pagamentos imediatos iniciados por QR Codes compostos**, neste momento, **NÃO permitem saque ou troco**. Todos os demais elementos do *payload* JSON descritos para pagamentos seguem as regras já definidas.

#### 2.8.4.2. O *payload* JSON dos parâmetros de recorrência

A estrutura do *payload* JSON de parâmetros de recorrência é:

| # | Campo | Mult. | Nome Campo JSON | Tipo | OU |
|---|---|---|---|---|---|
| 1.1 | Identificador da recorrência | [1..1] | `idRec` | String | |
| 2.1 | Identificador do objeto de vínculo. | [0..1] | `vinculo.objeto` | String | |
| 2.2.1 | CPF do usuário devedor. | [0..1] | `vinculo.devedor.cpf` | String | OU( |
| 2.2.2 | CNPJ do usuário devedor. | [0..1] | `vinculo.devedor.cnpj` | String | ) |
| 2.2.3 | Nome do usuário devedor. | [1..1] | `vinculo.devedor.nome` | String | |
| 2.3 | Identificador do contrato. | [1..1] | `vinculo.contrato` | String | |
| 3.1 | Data estimada de primeiro pagamento. | [1..1] | `calendario.dataInicial` | String | |
| 3.2 | Data final para os pagamentos. | [0..1] | `calendario.dataFinal` | String | |
| 3.3 | Periodicidade. | [1..1] | `calendario.periodicidade` | String | |
| 4.1 | Valor da recorrência. | [0..1] | `valor.valorRec` | String | |
| 4.2 | Valor mínimo da recorrência aceito pelo recebedor. | [0..1] | `valor.valorMinimoRecebedor` | String | |
| 5.1 | CPNJ do usuário recebedor. | [1..1] | `recebedor.cnpj` | String | |
| 5.2 | Nome do usuário recebedor. | [1..1] | `recebedor.nome` | String | |
| 5.3 | ISPB do usuário recebedor. | [1..1] | `recebedor.ispbParticipante` | String | |
| 6.1 | Política de retentativas a ser aplicada. | [1..1] | `politicaRetentativa` | String | |
| 7.1 | Estado da atualização da recorrência. | [1..1] | `atualizacao.status` | String | Array[1..N] ( |
| 7.2 | Data correspondente ao estado da recorrência. | [1..1] | `atualizacao.data` | String | ) |

Que, em linhas gerais, mostra: i) quais as informações sobre o vínculo do usuário devedor com o usuário recebedor; ii) qual seria a periodicidade desejada, com datas de início e fim (opcional); iii) qual o valor da recorrência e qual valor mínimo aceito pelo recebedor (opcionais); (iv) os dados do usuário recebedor; v) a política de retentativas a ser aplicada e por fim vi) uma lista com as mudanças de estado da recorrência correlata.
Para mais detalhes técnicos, **a API Pix[64] é a referência indicada**, lá estão os padrões usados em cada um dos campos acima.

### *2.8.5. Exemplos de QR Code Composto*

#### 2.8.5.1. QR Codes composto apenas com parâmetros de recorrência

Neste caso apenas o *Link URL dos parâmetros de recorrência* é fornecida no QR Code composto, ou seja, não temos o preenchimento do *Link URL do pagamento* ou chave DICT associada ao campo 26,

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

sendo este composto apenas pelo identificador do arranjo Pix. O campo 26, nesse caso, é obrigatório porque segundo a especificação EMVQR-MPM, pelo menos um MAI[65] deve estar presente.

**Nome EMV do recebedor** [66] : Fulano de Tal
**URL parâm. de recorrência** : pix.example.com/rec/2353c790eefb11eaadc10242ac120002

| ID | Nome EMV | Tam | Valor | | | |
|---|---|---|---|---|---|---|
| **00** | *Payload Format Indicator* | **02** | **01** | | | |
| **26** | *Merchant Account Information* | **18** | **ID** | **Nome** | **Tam** | **Valor** |
| | | | **00** | *GUI* | **14** | **br.gov.bcb.pix** |
| **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** | *** |
| **80** | *Unreserved Templates* | **74** | **ID** | **Nome** | **Tam** | **Valor** |
| | | | **00** | *GUI* | **14** | **br.gov.bcb.pix** |
| | | | **25** | URL | **52** | **pix.example.com/rec/2353c790eefb11eaadc10242ac120002** |
| **63** | *CRC16* | **04** | 0x***F2DA*** | | | |

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

```
00 02 01
26 18
      00 14 br.gov.bcb.pix
52 04 0000
53 03 986
58 02 BR
59 13 Fulano de Tal
60 08 BRASILIA
62 07
      05 03 ***
80 74
      00 14 br.gov.bcb.pix
      25 52 pix.example.com/rec/2353c790eefb11eaadc10242ac120002
63 04 F2DA
```

O respectivo QR Code composto está representado abaixo:

[65] Merchant Account Information, faixas 02-51. Ver EMV® QR Code Specification for Payment Systems (EMV QRCPS), sec. 4.7.9

[66] O nome que deve ser apresentado ao pagador é o nome do usuário recebedor conforme registrado no DICT.



00020126180014br.gov.bcb.pix52040000530398658O2BR5913Fulano de Tal6008BRASILIA62070503***80740014br.gov.bcb.pix2552pix.example.com/rec/2353c790eefb11eaadc10242ac1200026304F2DA

O processo de recuperação do *payload* JSON das configurações de recorrência:

**`pix.example.com/rec/2353c790eefb11eaadc10242ac120002`**

deve seguir todo o rigor de verificação de segurança já descrito na validação de URLs do exemplo do BC Code dinâmico.

O fluxo subsequente com a exibição oportuna dos parâmetros de recorrência para o usuário pagador seguirá os padrões definidos no manual de Requisitos Mínimos para a Experiência do Usuário.

2.8.5.2. QR Code composto com dados estáticos e parâmetros de recorrência.

Neste cenário a recorrência dos pagamentos é ofertada em conjunto com um pagamento via QR Estático. As informações gerais são similares ao exemplo dado em QR Code estático, complementadas com a existência da URL de configuração das recorrências ("/rec/") no PSP Recebedor.

**Nome EMV do recebedor** [67] : Fulano de Tal
**Valor** : R$ 100,50
**URL parâm. de recorrência** : pix.example.com/rec/2353c790eefb11eaadc10242ac120002

| ID | Nome EMV | Tam | Valor | | | |
|---|---|---|---|---|---|---|
| **00** | *Payload Format Indicator* | **02** | **01** | | | |
| **26** | *Merchant Account Information* | **70** | **ID** | **Nome** | **Tam** | **Valor** |
| | | | **00** | *GUI* | **14** | **br.gov.bcb.pix** |
| | | | **01** | Chave | **48** | `123e4567-e12b-12d1-a456-426655440000` |
| **52** | *Merchant Category Code* | **04** | **0000** (não informado) | | | |

[67] O nome que deve ser apresentado ao pagador é o nome do usuário recebedor conforme registrado no DICT.

<table>
<tr><td>53</td><td>Transaction Currency</td><td>03</td><td colspan="4">986 (R$)</td></tr>
<tr><td>54</td><td>Transaction Amount</td><td>06</td><td colspan="4">100.50</td></tr>
<tr><td>58</td><td>Country Code</td><td>02</td><td colspan="4">BR</td></tr>
<tr><td>59</td><td>Merchant Name</td><td>13</td><td colspan="4">Fulano de Tal</td></tr>
<tr><td>60</td><td>Merchant City</td><td>08</td><td colspan="4">BRASILIA</td></tr>
<tr><td rowspan="2">62</td><td rowspan="2">Additional Data Field</td><td rowspan="2">07</td><td>ID</td><td>Nome</td><td>Tam</td><td>Valor</td></tr>
<tr><td>05</td><td>Reference Label</td><td>03</td><td>***</td></tr>
<tr><td rowspan="3">80</td><td rowspan="3">Unreserved Templates</td><td rowspan="3">74</td><td>ID</td><td>Nome</td><td>Tam</td><td>Valor</td></tr>
<tr><td>00</td><td>GUI</td><td>14</td><td>br.gov.bcb.pix</td></tr>
<tr><td>25</td><td>URL</td><td>52</td><td>pix.example.com/rec/2353c790eefb11eaadc10242ac120002</td></tr>
<tr><td>63</td><td>CRC16</td><td>04</td><td colspan="4">0x2875</td></tr>
</table>

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

```
00 02 01
26 58
      00 14 br.gov.bcb.pix
      01 36 123e4567-e12b-12d1-a456-426655440000
52 04 0000
53 03 986
54 06 100.50
58 02 BR
59 13 Fulano de Tal
60 08 BRASILIA
62 07
      05 03 ***
80 74
      00 14 br.gov.bcb.pix
      25 52 pix.example.com/rec/2353c790eefb11eaadc10242ac120002
63 04 2875
```

O respectivo QR Code composto está abaixo:



00020126580014br.gov.bcb.pix0136123e4567-e12b-12d1-a456-426655440000520400005303986540610 0.505802BR5913Fulano de Tal6008BRASILIA62070503***80740014br.gov.bcb.pix2552pix.example.com/rec/2353c790eefb11eaadc10242ac12000263042875

O processo de recuperação dos *payloads* JSON dos parâmetros de recorrência:

**`pix.example.com/rec/2353c790eefb11eaadc10242ac120002`**

deve seguir todo o rigor de verificação de segurança já descrito na validação de URLs do exemplo do BC Code dinâmico.

O fluxo subsequente com a exibição oportuna dos dados de pagamentos e dos parâmetros de recorrência para o usuário pagador seguirá os padrões definidos no manual de Requisitos Mínimos para a Experiência do Usuário.

2.8.5.3. QR Code composto com dados dinâmicos (pagamento imediato ou com vencimento[68]) e parâmetros de recorrência.

Neste cenário a recorrência dos pagamentos é ofertada em conjunto com um pagamento imediato (ou com vencimento). As informações gerais são similares ao exemplo dado em QR Code dinâmico, complementadas com a existência da URL de configuração das recorrências (“/rec/”) no PSP Recebedor.

**Nome EMV do recebedor** [69] : Fulano de Tal
**URL do pagamento** : pix.example.com/8b3da2f39a4140d1a91abd93113bd441
**URL parâm. de recorrência** : pix.example.com/rec/2353c790eefb11eaadc10242ac120002

[68] Neste caso a URL do pagamento deve conter o path ‘/cobv’.
[69] O nome que deve ser apresentado ao pagador é o nome do usuário recebedor conforme registrado no DICT.

| 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) [70] | | | |
| **26** | *Merchant Account Information* | **70** | **ID** | **Nome** | **Tam** | **Valor** |
| | | | **00** | *GUI* | **14** | **br.gov.bcb.pix** |
| | | | **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** | *** |
| **80** | *Unreserved Templates* | **74** | **ID** | **Nome** | **Tam** | **Valor** |
| | | | **00** | *GUI* | **14** | **br.gov.bcb.pix** |
| | | | **25** | URL | **52** | `pix.example.com/rec/2353c790eefb11eaadc10242ac120002` |
| **63** | *CRC16* | **04** | 0x***FB42*** | | | |

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

```
00 02 01
01 02 12
26 70
      00 14 br.gov.bcb.pix
      25 48 pix.example.com/8b3da2f39a4140d1a91abd93113bd441
52 04 0000
53 03 986
58 02 BR
59 13 Fulano de Tal
60 08 BRASILIA
62 07
      05 03 ***
80 74
      00 14 br.gov.bcb.pix
      25 52 pix.example.com/rec/2353c790eefb11eaadc10242ac120002
63 04 FB42
```

[70] 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).

O respectivo QR Code composto está abaixo:



00020101021226700014br.gov.bcb.pix2548pix.example.com/8b3da2f39a4140d1a91abd93113bd44
1520400005303986580 2BR5913Fulano de
Tal6008BRASILIA62070503***80740014br.gov.bcb.pix2552pix.example.com/rec/2353c790eefb11ea
adc10242ac1200026304FB42

O processo de recuperação dos *payloads* JSON com os dados do pagamento:

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

e dos parâmetros de recorrência:

**`pix.example.com/rec/2353c790eefb11eaadc10242ac120002`**

devem seguir todo o rigor de verificação de segurança já descrito na validação de URLs do exemplo do BC Code dinâmico, incluindo-se o respeito à tipagem de conteúdo definido pela RFC 7515. Reforçando-se também que, para este cenário, temos a obrigatoriedade de que o fqdnPspRecebedor (vide seção 2.5.2) deve ser o mesmo para as duas URLs.

O fluxo subsequente com a exibição oportuna dos dados de pagamentos e dos parâmetros de recorrência para o usuário pagador seguirá os padrões definidos no manual de Requisitos Mínimos para a Experiência do Usuário.

# 3. Outras formas de iniciação

## 3.1. 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 para o QR Code estático, o QR Code dinâmico e, também, para o QR Code composto.

## 3.2. Serviço de Iniciação de Transação de Pagamento

Um Pix pode ser iniciado através do serviço de iniciação de transação de pagamento vinculado à implementação do Sistema Financeiro Aberto, tanto pelo participante iniciador, quanto pelo participante provedor de conta transacional.

A iniciação de um Pix através do serviço de iniciação de transação de pagamento acima referido deve conter, minimamente, o conjunto de dados listados abaixo.

**Envio das Informações pelo Prestador de Serviço de Iniciação para o Detentor da Conta**

| Nome do Campo | Obrigatoriedade | | | | | |
|---|---|---|---|---|---|---|
| | Inserção Manual | Chave Pix | Usuário Recebedor com Chave | QR Code Estático | QR Code Dinâmico | Pix Automático (NR) |
| Tipo de Iniciação | MANU | DICT | INIC | QRES | QRDN | AUTO (****) |
| Iniciador de pagamento (CNPJ) | Sim | Sim | Sim | Sim | Sim | Sim |
| EndToEndId (*) | Sim | Sim | Sim | Sim | Sim | Sim |
| QrCode (Pix Copia e Cola) | Não | Não | Não | Sim | Sim | Não |
| Código da cidade IBGE (CodMun) | Não | Não | Não | Não | Não (***) | Não |
| Identificação do recebedor (CPF/CNPJ) | Sim | Sim | Sim | Sim | Sim | Sim |
| Instituição do recebedor participante do Pix | Sim | Sim | Sim | Sim | Sim | Sim |
| Agência do recebedor | Não | Não | Não | Não | Não | Não |
| Conta do recebedor | Sim | Sim | Sim | Sim | Sim | Sim |
| Tipo de Conta do recebedor | Sim | Sim | Sim | Sim | Sim | Sim |
| Valor | Sim | Sim | Sim | Sim | Sim | Sim |
| Moeda | Sim | Sim | Sim | Sim | Sim | Sim |
| Campo de descrição | Não (**) | Não (**) | Não (**) | Não (**) | Não (**) | Não (**) |
| Chave Pix | Não | Sim | Sim | Sim | Sim | Não |
| Código de Conciliação | Não | Não | Sim | Não | Sim | Não |

(*) A regra de obrigatoriedade de geração do <EndToEndId> pelo iniciador entrou em produção em 25/09/22. Anteriormente, admitia-se que o <EndToEndId> fosse gerado pelo participante direto, pelo participante indireto ou pelo iniciador de pagamento.
(**) É obrigatório o envio da informação da descrição quando o usuário preencher o campo.
(***) No caso de QR Code Dinâmico com vencimento, deve ser considerado o código do município do usuário pagador informado pelo prestador do serviço de iniciação.
(****) As informações contidas na instrução de pagamento do Pix Automático serão providas a partir da API do Open Finance. (NR)

## 3.3. Pix Automático

O Pix Automático é a solução que permite o pagamento de uma cobrança recorrente, de forma automática, mediante a concessão de uma permissão do usuário pagador ao usuário recebedor, utilizando-se da estrutura do Pix.

Uma vez concedida a permissão pelo usuário pagador, o usuário recebedor enviará periodicamente as informações das cobranças recorrentes ao seu PSP, via API Pix ou arquivo padronizado. O PSP Recebedor providenciará a instrução de pagamento correspondente e a enviará ao PSP Pagador, que por sua vez realizará o agendamento do débito e, posteriormente, sua liquidação na data prevista, de forma automática. A data prevista para liquidação de uma cobrança recorrente deve ser igual à data de vencimento. Caso a data de vencimento seja um dia não útil, a data prevista para liquidação pode corresponder ao próximo dia útil, a critério do usuário recebedor.

Mais detalhes sobre esta funcionalidade e as regras para a sua implementação estão descritos no ANEXO I – API Pix: Conceitos de Negócio e no ANEXO IV – Pix Automático.

### *3.3.1. Jornadas de autorização*

As jornadas de autorização possíveis para a confirmação do Pix Automático correspondem aos seguintes artigos do Regulamento do Pix, inseridos pela Resolução BCB nº 402, de 22.7.2024:

| Jornada de autorização | Regulamento Pix |
|---|---|
| Jornada 1 | art. 11-Q, § 1º, inciso VII, alínea a |
| Jornada 2 | art. 11-Q, § 1º, inciso VII, alínea b |
| Jornada 3 | art. 11-Q, § 1º, inciso VII, alínea c |
| Jornada 4 | art. 11-Q, § 1º, inciso VII, alínea d |
| Jornada via Open Finance[71] | art. 11-Q, § 1º, inciso VII, alínea e |

### *3.3.2. QR Codes no Pix Automático*

No Pix Automático, utilizamos o QR Code composto para as jornadas de autorização 2, 3, e 4.

Diferentemente das outras cobranças do Pix, **não há geração de QR Code para as cobranças recorrentes**.

É importante frisar que caso a leitura de um QR Code composto para a jornada de autorização 3 apresente erro em qualquer dos seus dois links URL, o PSP pagador deve exibir uma mensagem de erro para o usuário pagador, interrompendo o processo de pagamento/autorização.

No caso em que a leitura de um QR Code composto para a jornada de autorização 4 encontre erro no link ou nos dados da cobrança (inclusive em razão de a cobrança já ter sido paga) o PSP pagador também deve exibir uma mensagem de erro para o pagador, independentemente do resultado da leitura do link da recorrência. No entanto, se o erro estiver no link da recorrência, o PSP pagador deve

[71] Regras estão definidas no arcabouço normativo do Open Finance.

permitir que o usuário pagador continue com o processo de pagamento normal de um QR Code estático ou dinâmico com vencimento, sem a oferta do Pix Automático ao final da jornada.

## 3.4. Pix por aproximação

Um Pix pode ser iniciado por meio da tecnologia *Near Field Communication* (NFC)[72]. Essa forma de iniciação tem muitas semelhanças com a iniciação por leitura de QR Code.

Para que a iniciação de um Pix por aproximação funcione, é necessário que:

- o dispositivo do recebedor[73] possua antena NFC;
- o dispositivo do pagador[74] também tenha a tecnologia NFC; e
- ambos os dispositivos estejam fisicamente próximos e conectados à internet.

Atendidas as condições e uma vez combinado entre pagador e recebedor que a transação será realizada via Pix por aproximação, o dispositivo recebedor emitirá um "*payload*", correspondente a um "Pix Copia e Cola" codificado em uma URI. A partir da obtenção do *payload* do QR Code pelo dispositivo do usuário pagador, o fluxo da transação segue o mesmo caminho de um pagamento via leitura de QR Code, inclusive no que se refere às etapas de confirmação dos dados de pagamento.

Assim que obtido o *payload* via NFC, a URL contida no "Pix Copia e Cola" será resolvida, as checagens de segurança serão realizadas e o *payload* JSON[75] será recuperado. O *payload* JSON inclui, entre outros elementos, as informações sobre valor e recebedor. Após a confirmação pelo usuário pagador, o Pix será enviado e a transação será concluída da mesma forma que ocorre no cenário de leitura de QR Codes do Pix.

Observa-se, portanto, que a única diferença entre o "Pix por aproximação" e a leitura convencional de um QR Code do Pix está na forma de obtenção do *payload*: em vez de se utilizar uma câmera para a leitura do QR Code, é utilizada uma antena NFC.

É importante ressaltar que **todas as providências e verificações de segurança** aplicáveis no fluxo de leitura de um QR Code por um aplicativo de um PSP pagador também se aplicam no Pix por aproximação.

Os detalhes técnicos da comunicação NFC realizada entre o dispositivo do recebedor e o dispositivo do pagador, em ambiente Android, estão descritos no documento **"Especificações do Pix por aproximação para Android"**, disponível em https://www.bcb.gov.br/content/estabilidadefinanceira/pix/especificacoes_pix_aproximacao_android.pdf. Trata-se de documento complementar a este Manual, de observância obrigatória pelos participantes. A opção por mantê-lo como documento apartado justifica-se pela necessidade de maior conveniência na sua atualização, tendo em vista que os sistemas operacionais utilizados nos dispositivos móveis, mantidos por terceiros independentes, podem sofrer alterações técnicas que demandem ajustes na especificação — o que exige adaptações técnicas que não alterem o conteúdo normativo da funcionalidade, preservando a clareza regulatória e a aderência às boas práticas de versionamento.

---

[72] Verificar Padrões ECMA-340 e ISO/IEC 18092.
[73] Com frequência, um terminal POS (as "maquininhas de cartão"), porém pode ser qualquer dispositivo que emita um sinal NFC.
[74] Geralmente um aparelho "smartphone".
[75] O Pix Copia e Cola pode ser um QR Estático. Nesse caso, não há resolução de payload JSON.

# /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.

A API Pix contempla as funcionalidades necessárias para viabilizar:

a) 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 **pagamentos com data de vencimento** específica, incorporando ainda os cálculos referentes a descontos e abatimentos, juros e multas, quando aplicáveis;
b) as transações Pix, com finalidade de **saque** ou de **troco**, iniciadas por meio de QR Code dinâmico; e
c) o recebimento de **cobranças recorrentes** via Pix Automático, incluindo a solicitação e gestão de permissões para pagamentos recorrentes.

A API Pix continuará evoluindo para agregar novas funcionalidades.

## 2. Documentação da API Pix

Além desta documentação, com seus anexos, que descrevem informações gerais da API Pix e complementam detalhes específicos dos produtos e casos de uso correlatos, a documentação da API Pix também é composta pela especificação técnica detalhada apresentada no formato OpenAPI 3.0 e disponível em https://github.com/bacen/pix-api.

Para acessar a documentação técnica correspondente à versão deste documento, deve-se recuperar a versão da especificação técnica de mesmo identificador. Para cada versão deste documento, é gerada uma "*tag*" correlata no Github contendo o seu número, no formato "major.minor.patch" (exemplo: 2.1.0), disponível na capa deste documento. Desta forma, as descrições aqui realizadas estarão sincronizadas e compatíveis com as informações obtidas na versão OpenAPI correspondente.

# 3. Contexto da API Pix

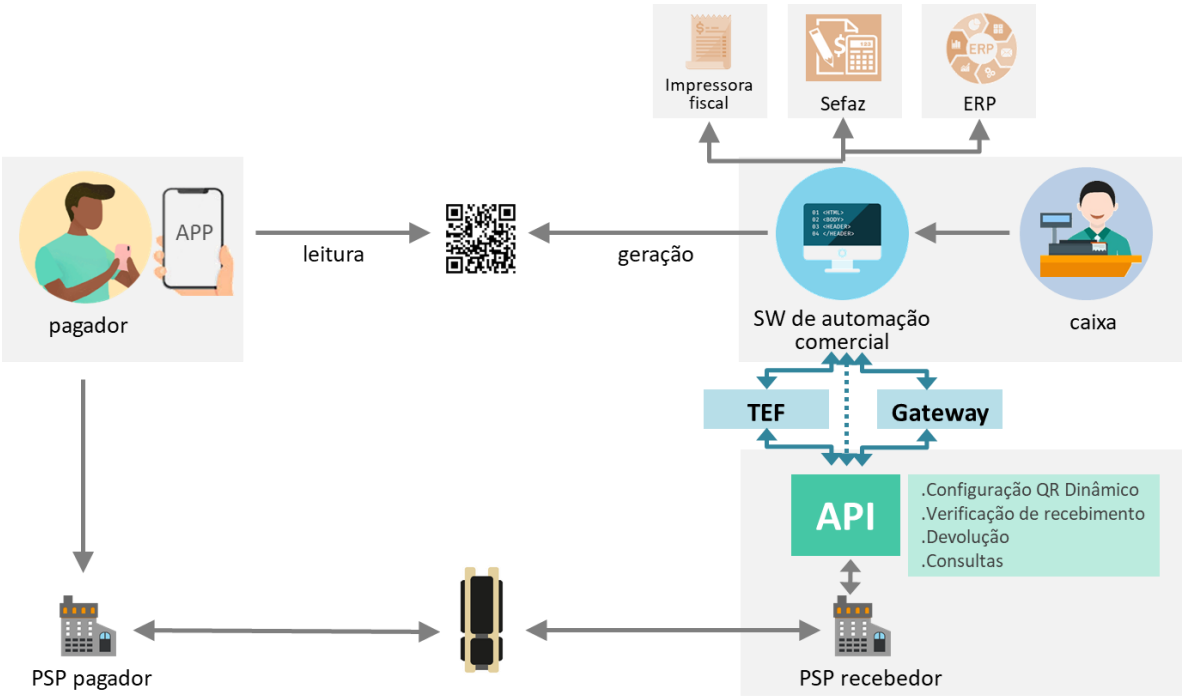



**Figura 1:** A API Pix

A API Pix é o componente do arranjo que visa possibilitar que o usuário pagador ou recebedor, no contexto P2B[76] ou B2B[77], 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 recebedor 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, bem como iniciar um processo de cobranças recorrentes, o Pix Automático. Na **figura 1,** pode-se visualizar possíveis caminhos de integração dos sistemas do usuário recebedor com a API Pix do PSP.

O usuário recebedor 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;

[76] "Person to Business", trata-se de uma interação de Pessoa para Negócio/empresa/estabelecimento.
[77] "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;

VII – criar e gerenciar recorrências, necessárias para a realização de pagamentos recorrentes do Pix Automático;

VIII – enviar solicitações de confirmação de recorrência de pagamentos aos clientes;

IX – criar e gerenciar cobranças recorrentes.

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, ou composto, e que representa uma cobrança ou dados de recorrência;

**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[78]. 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

[78] 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

automação dos seus processos de geração dessas cobranças e para conciliação de pagamentos recebidos por meio do Pix.

**IX – facilitador de serviço de saque (fss)**: participante do Pix, que se enquadre na modalidade provedor de conta transacional e que seja autorizado a funcionar pelo Banco Central do Brasil, que, em caráter facultativo, venha a facilitar o serviço de saque, diretamente, ou por meio de agente de saque, mediante estabelecimento de relação contratual para essa finalidade.

**X – recorrência:** contém informações gerais sobre o usuário recebedor e o débito recorrente, no contexto do Pix Automático. As informações da recorrência serão utilizadas para validar e permitir o pagamento automático, após autorização pelo usuário pagador.

**XI – solicitação de confirmação de recorrência:** solicitação encaminhada ao usuário pagador para que ele confirme se autoriza que o PSP Pagador realize os pagamentos recorrentes automaticamente.

**XII – autorização:** permissão concedida pelo usuário pagador ao seu PSP, para que este possa realizar débitos em sua conta transacional com a finalidade de pagar cobranças recorrentes, em conformidade com determinadas variáveis, que são os dados da recorrência e os parâmetros definidos pelo próprio usuário junto ao seu PSP (funcionalidades).

**XIII – cobrança recorrente:** cobrança gerada periodicamente pelo usuário recebedor e enviada pelo PSP Recebedor ao PSP Pagador, para que seja agendada e paga automaticamente, após a autorização para recorrência que foi concedida pelo usuário pagador.

**XIV – retentativa intradia por falta de saldo**: dever de o PSP Pagador tentar novamente realizar o pagamento ao longo do dia, em caso de ausência de saldo suficiente na conta do usuário pagador. Esta retentativa fica no controle exclusivo do PSP Pagador (não há comunicação com o PSP Recebedor).

**XV – retentativa intradia por erro na liquidação:** possibilidade de enviar uma nova instrução de pagamento com vencimento no mesmo dia, caso tenha ocorrido algum erro operacional ou de comunicação que tenha impedido a liquidação da transação após o envio da ordem de pagamento.

**XVI – retentativa após o vencimento:** possibilidade de realizar novas tentativas de pagamento nos dias subsequentes caso o pagamento da cobrança recorrente não tenha sido realizado na data de agendamento original por ausência de recursos suficientes ou de limite transacional disponível ou por falha operacional que impeça o envio da ordem de pagamento para liquidação. Eventuais ajustes calculados pelo pagamento efetivado após a data do vencimento só podem ser aplicados à próxima cobrança

# 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[79];
c) **REMOVIDO_PELO_USUARIO_RECEBEDOR**: indica que o usuário recebedor solicitou a remoção da cobrança; e
d) **REMOVIDO_PELO_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.

**VI – CobPayload:** utilizado pelo software do PSP pagador para recuperar o payload JSON que representa uma cobrança imediata ou com vencimento.

**VII – Recorrência (/rec):** contém informações da recorrência, utilizada no contexto do Pix Automático.

Estados da recorrência:

**a) CRIADA:** indica que a recorrência foi criada pelo usuário recebedor;
**b) APROVADA:** indica que a recorrência foi aceita pelo usuário pagador;
**c) REJEITADA:** indica que a recorrência foi rejeitada pelo usuário pagador (ocorre apenas na Jornada 1 de autorização);

[79] Importante observar que o estado CONCLUÍDA 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.

d) **EXPIRADA:** indica que a data final da recorrência já passou;
e) **CANCELADA:** indica que a recorrência foi cancelada pelo usuário recebedor ou pelo usuário pagador.

**VIII – Solicitação de confirmação de recorrência (/solicrec):** representa uma solicitação de confirmação de recorrência do Pix Automático, enviada nos casos da Jornada 1 de autorização.

Estados da solicitação de confirmação de recorrência:

a) **CRIADA:** indica que a solicitação foi criada pelo usuário recebedor;
b) **ENVIADA:** indica que o PSP Recebedor enviou a solicitação para o PSP Pagador;
c) **RECEBIDA:** indica que o PSP Pagador recebeu a solicitação de confirmação de recorrência e ela está aguardando a ação do usuário pagador;
d) **REJEITADA:** indica que a solicitação foi rejeitada pelo PSP Pagador por algum erro (como conta inexistente) ou que foi rejeitada pelo usuário pagador no momento da oferta de adesão ao Pix Automático;
e) **ACEITA:** indica que o usuário pagador aceitou a solicitação de confirmação de recorrência;
f) **EXPIRADA:** indica que a data de expiração da solicitação de confirmação de recorrência já passou, sem que ela tenha tido uma resposta do usuário pagador;
g) **CANCELADA**: indica que a solicitação foi cancelada por algum dos seguintes motivos:
1 – a recorrência à qual ela está associada foi cancelada;
2 – a recorrência à qual ela está associada foi confirmada por alguma outra jornada de adesão ao Pix Automático;
3 – o recebedor solicitou o cancelamento por algum erro, caso ela ainda não tenha sido confirmada ou rejeitada pelo usuário pagador; ou
4 – PSP recebedor solicita cancelamento por timeout: o PSP recebedor não recebeu a confirmação do recebimento da mensagem com a solicitação dentro do prazo limite de 1 minuto.

**IX – Cobrança recorrente (/cobr):** representa uma cobrança recorrente do Pix Automático.

Estados da cobrança recorrente:

a) **CRIADA:** indica que a cobrança foi criada pelo usuário recebedor;
b) **ATIVA:** indica que a cobrança foi enviada pelo PSP Recebedor ao PSP Pagador;
c) **CONCLUIDA:** indica que a cobrança foi paga;
d) **EXPIRADA:** indica que a cobrança não foi quitada após expiradas todas as tentativas de pagamento permitidas pela política vigente;
e) **REJEITADA:** indica que a cobrança foi rejeitada pelo PSP Pagador;
f) **CANCELADA:** indica que a cobrança foi cancelada pelo usuário recebedor ou pelo usuário pagador.

**X – WebhookRec (/webhookrec):** é um recurso técnico que permite que o PSP Recebedor informe diretamente ao usuário recebedor informações sobre as recorrências. Assim como no Webhook para as cobranças, aqui se aplicam as notificações como forma de evitar *polling* em cima dos dados de recorrência.

**XI – WebhookCobR (/webhookcobr):** é um recurso técnico que permite que o PSP Recebedor informe diretamente ao usuário recebedor informações sobre as cobranças recorrentes. Assim como no

Webhook para as cobranças, aqui se aplicam as notificações como forma de evitar *polling* em cima dos dados das cobranças recorrentes.

**XII – PayloadLocationRec (/locRec):** é um recurso que permite ao PSP Recebedor reusar uma URL, retornando diferentes parâmetros de recorrência (*payloads* JSON) ao longo do tempo, mas apenas uma por vez. É utilizado quando o usuário recebedor precisa gerar um novo QR Code composto ou reutilizar um existente.

**XIII – RecPayload:** utilizado pelo software do PSP pagador para recuperar o payload JSON que representa uma recorrência.

## 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, em um determinado momento, o **PayloadLocation** só pode estar associado a uma **Cobrança**;

V – Uma **Recorrência** pode estar associada a uma ou mais **Solicitações de Recorrência**;

VI – Uma **Solicitação de Recorrência** está associada a uma única **Recorrência**;

VII – Uma **Recorrência** pode estar associada a várias **Cobranças Recorrentes**;

VIII – Uma **Cobrança Recorrente** está associada a uma única **Recorrência**;

IX – Uma **Cobrança Recorrente** pode estar associada a um ou mais **Pix** (mesmo txid);

X – Uma **Recorrência** somente pode estar associada a um **PayloadLocationRec** e, em um determinado momento, o **PayloadLocationRec** só pode estar associado a uma **Recorrência**.

## 5.3 Ciclo de vida do TransactionID (txid)

Há 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[80] quando da geração da cobrança, e não poderá se repetir entre cobranças distintas[81], 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[82].

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 ser sempre único, não podendo ser repetido, ainda que a cobrança original tenha sido cancelada ou baixada.

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.

### *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. Pagamento no ato da compra com QR Code Estático fixo e sem valor definido.) e outros em que deve ser evitada (ex. Pagamento no ato da compra com QR Code Estático gerado no ato da compra). Desta forma, a consistência dos pagamentos realizados por meio de QR Codes estáticos fica totalmente a cargo do usuário recebedor[83].

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 disponíveis na API correspondem aos *endpoints* e respectivas ações criados em cada tag.

### *5.4.1 Funcionalidades obrigatórias por produtos*

---

[80] 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.

[81] Cabe enfatizar, inclusive, que não pode existir uma cobrança para pagamento imediato e uma cobrança para pagamento com vencimento que utilizem o mesmo txid.

[82] 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.

[83] 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.

No contexto da API Pix, devem ser implementadas as funcionalidades das tags abaixo, para cada produto:

| Tags | Produtos | | | |
|---|---|---|---|---|
| | Pix Cobrança | | Pix Saque e Pix Troco | Pix Automático |
| | *Pagamento Imediato* | *Pagamento com Vencimento* | | |
| **Cob** | x | | x | x[1] |
| **CobV** | | x | | x[2] |
| **LoteCobV** | | x | | |
| **Pix** | x | x | x | x |
| **PayloadLocation** | x | x | x | x[3] |
| **PayloadLocationRec** | | | | x[4] |
| **CobPayload** | x | x | x | x[3] |
| **Webhook** | x | x | x | x[1] |
| **WebhookRec** | | | | x |
| **WebhookCobR** | | | | x |
| **Rec** | | | | x |
| **CobR** | | | | x |
| **SolicRec** | | | | x[5] |
| **RecPayload** | | | | x[4] |

1 – Apenas para a Jornada 3 de autorização.
2 – Apenas para a Jornada 4 de autorização.
3 – Apenas para as jornadas 3 e 4 de autorização.
4 - Para as jornadas 2, 3 e 4 de autorização.
5 - Apenas para a jornada 1 de autorização.

# 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[84]:

*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.

[84] Vale lembrar que a API Pix não é utilizada para a geração de QR Codes estáticos.

**Premissa**: a 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 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. Comércios *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[85].*

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.

---

[85] 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.

7. Com o pagamento tendo sido agendado, na data agendada, o PSP Pagador envia a ordem de pagamento, dentro do dia especificado (na primeira janela, entre 0 h e 8 h, se há recursos na conta do usuário pagador e na segunda janela, entre 18h e 21h, se não houve liquidação anterior por falta de saldo), 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:

   *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 umas 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 QR Code Composto

### *6.3.1 Utilização de QR Code Composto apenas para ofertar uma recorrência*

**Aplicação**: Prestadores de serviço que desejam ofertar o pagamento de cobranças recorrentes via Pix Automático. Corresponde à Jornada 2 de autorização.

**Premissa:** deve existir uma recorrência já criada e ainda não autorizada.

1. Para configurar a API Pix para esse caso de uso, o primeiro passo é criar uma "location", a URL que exibirá os parâmetros da recorrência, e que será configurada no QR Code composto:

   *Serviços invocados:*

   ** POST /locrec → será retornado o identificador da "location" e a URL que será utilizada para exibir os parâmetros da recorrência.*

   ** POST /rec → antes de mostrar o QR Code Composto ao usuário pagador, o usuário recebedor cria a recorrência com o identificador da "location" recebido do serviço anterior com todos os parâmetros da recorrência.*

2. O usuário pagador (cliente) recebe uma oferta do usuário recebedor (fornecedor) para aderir ao Pix Automático via um QR Code composto com dados de recorrência;

   *Serviços invocados:*

   ** GET /rec/{idRec} → após a definição da location e criação da recorrência, nos endpoints anteriores, o usuário recebedor acessa este endpoint para obtenção dos dados de recorrência com o campo 'dadosQR' preenchido com a informação de jornada 2 e o Pix Copia e Cola correspondente ao QR Composto desejado para a exibição ao usuário pagador.*

3. O cliente lê o QR Code composto por meio do app do seu PSP;

   *Serviço invocado: GET /rec/{recUrlAccessToken} → o PSP Pagador lê os dados da recorrência para controle e apresentação ao usuário pagador para aceite.*

4. O cliente conclui a jornada 2, confirmando a autorização do Pix Automático para pagamentos àquele fornecedor;
5. As próximas cobranças serão enviadas pelo usuário recebedor ao PSP pagador, passando pelo PSP recebedor, por meio de instruções de pagamento;

   *Serviço invocado: POST /cobr → cria a cobrança recorrente com os dados pertinentes sendo gerado um txid para conciliação; ou PUT /cobr/{txid} → cria a cobrança recorrente com os dados pertinentes já informando um txid para conciliação.*

6. O PSP pagador realiza o pagamento das cobranças de forma automática, a partir das instruções de pagamento recebidas.

   *Serviço invocado: GET /cobr/{txid} → a qualquer momento este endpoint retorna os dados da cobrança com o Pix de pagamento correlato.*

### *6.3.2 Pagamento de QR Composto com dados estáticos e recorrência*

**Aplicação**: Prestadores de serviço que precisam gerar QR Codes de cobrança de forma *offline* mas que desejam ofertar o pagamento via Pix Automático aos seus clientes. Corresponde à Jornada 4 de autorização.

**Premissa:** deve existir uma recorrência já criada e ainda não autorizada.

1. Para configurar a API Pix para esse caso de uso, o primeiro passo é criar uma "location", a URL que exibirá os parâmetros da recorrência, e que será configurada no QR Code composto. Esse passo é repetido para criação de todas as recorrências necessárias para os diferentes clientes.

   *Serviços invocados:*

   ** POST /locrec → será retornado o identificador da "location" e a URL que será utilizada para exibir os parâmetros da recorrência.*

   ** POST /rec → antes de mostrar o QR Code Composto ao usuário pagador, usuário recebedor cria a recorrência com o identificador da "location" recebido do serviço anterior com todos os parâmetros da recorrência.*

2. Quando for gerar, de forma *offline*, um QR Code estático contendo a cobrança de determinado cliente, o usuário recebedor insere também a *location* associada à recorrência previamente criada para aquele cliente, tornando-o um QR Code composto[86];
3. Quando o usuário pagador utilizar o *app* do seu PSP para ler aquele QR Code, é apresentada a ele a jornada de pagamento de um QR Code estático;
4. Após a conclusão dessa jornada, é ofertada ao usuário pagador a possibilidade de autorizar o pagamento das próximas cobranças por meio do Pix Automático;[87]
5. Caso o usuário pagador aceite, é apresentada a ele a jornada de autorização do Pix Automático, com base nos dados da recorrência obtidos por meio da *location* contida no QR Code composto que ele leu.

   *Serviço invocado: GET /rec/{recUrlAccessToken} → o Psp Pagador lê os dados da recorrência para controle e apresentação ao usuário pagador para aceite.*

### *6.3.3 Pagamento imediato via QR Code Composto com dados dinâmicos e recorrência*

**Aplicação**: Prestadores de serviço, tipicamente de assinatura, em que o acesso ao serviço é disponibilizado imediatamente após o primeiro pagamento e as cobranças seguintes são realizadas periodicamente. Utilizado na Jornada 3 de autorização do Pix Automático.

**Premissa:** deve existir uma recorrência já criada e ainda não autorizada.

---

[86] A construção do QR Composto pelo PSP Recebedor neste cenário não está contemplado na API Pix, que não engloba o tratamento de QRs estáticos, sendo a geração deste QR Composto responsabilidade do usuário recebedor.

[87] Se por qualquer razão o PSP pagador não conseguir obter os dados da recorrência por meio da *location* contida no QR Code, esta etapa (e a seguinte) não deve ocorrer. Ou seja, a jornada apresentada ao usuário pagador deve ser a jornada de pagamento de um QR Code estático.

1. O usuário pagador deseja contratar um serviço de assinatura e entra em contato com o fornecedor informando que deseja realizar o pagamento das cobranças periódicas por meio do Pix Automático;
2. O fornecedor gera um QR Code composto, contendo duas *locations*: uma para a cobrança imediata inicial e outra para os parâmetros da recorrência de cobranças futuras;

    *Serviços invocados:*

    ** Endpoints para criação de cobrança imediata, já exemplificados, de forma a haver a location para a cobrança com um txid que será compartilhado com a recorrência.*

    ** POST /locrec → será retornado o identificador da "location" e a URL que será utilizada para exibir os parâmetros da recorrência.*

    ** POST /rec → antes de mostrar o QR Code Composto ao usuário pagador, o usuário recebedor cria a recorrência com o identificador da "location" recebido do serviço anterior com todos os parâmetros da recorrência informando também o txid, via campo 'ativacao.dadosJornada.txid' no payload, que será conciliado com o txid idêntico da cobrança imediata a ser paga.*

    ** GET /rec/{idRec}?txid={txid} → após a definição da location e criação da recorrência, nos endpoints anteriores, o usuário recebedor acessa este endpoint para obtenção dos dados de recorrência com o campo 'dadosQR' preenchido com a informação de jornada 3 e o Pix Copia e Cola correspondente ao QR Composto desejado para a exibição ao usuário pagador.*

3. O usuário pagador lê esse QR Code no *app* do seu PSP e dá início a uma jornada em que se realizam, simultaneamente, o pagamento imediato inicial e a autorização para o pagamento das cobranças futuras por meio do Pix Automático;
4. Ao receber o primeiro pagamento imediato, o fornecedor disponibiliza instantaneamente o acesso ao serviço ao usuário pagador;

    *Serviço habilitados: POST /cobr ou PUT /cobr/{txid} → passam a ficar disponíveis para o usuário recebedor disparar as futuras cobranças recorrentes.*

5. As demais cobranças periódicas são pagas automaticamente por meio do Pix Automático.

### *6.3.4 Pagamento com vencimento via QR Code Composto com dados dinâmicos e recorrência*

**Aplicação**: Prestadores de serviço cuja cobrança se dá de forma periódica desejam ofertar a seus clientes a possibilidade de pagamento dessas cobranças por meio do Pix Automático. Corresponde à Jornada 4 de autorização.

**Premissa:** deve existir uma recorrência já criada e ainda não autorizada.

1. Prestador de serviço que realiza cobranças de forma periódica por meio de QR Codes dinâmicos com vencimento, passa a gerar QR Codes compostos, contendo a *location* dos dados da cobrança com vencimento usual mais a *location* dos parâmetros da recorrência;

*Serviços invocados:*

** Endpoints para criação de cobrança com vencimento, já exemplificados, de forma a haver a location para a cobrança.*

** POST /locrec → será retornado o identificador da "location" e a URL que será utilizada para exibir os parâmetros da recorrência.*

** POST /rec → antes de mostrar o QR Code Composto ao usuário pagador, usuário recebedor cria a recorrência com o identificador da "location" recebido do serviço anterior com todos os parâmetros da recorrência.*

** GET /rec/{idRec}?txid={txid} → após a definição da location e criação da recorrência, nos endpoints anteriores, o usuário recebedor acessa este endpoint para obtenção dos dados de recorrência com o campo 'dadosQR' preenchido com a informação de jornada 4 e o Pix Copia e Cola correspondente ao QR Composto desejado para a exibição ao usuário pagador.*

2. O usuário pagador, ao ler esse QR Code composto utilizando o *app* do seu PSP, percorrerá a jornada usual de pagamento da cobrança com vencimento;
3. Ao final dessa jornada, o usuário pagador será questionado acerca do seu interesse em utilizar o Pix Automático para o pagamento das cobranças futuras referentes ao serviço em questão;[88]
4. Caso tenha interesse, será apresentada ao pagador a jornada de autorização do Pix Automático, contendo todos os parâmetros da recorrência;

   *Serviço invocado: GET /rec/{recUrlAccessToken} → o Psp Pagador lê os dados da recorrência para controle e apresentação ao usuário pagador para aceite.*

5. Confirmada a autorização pelo pagador, notifica-se o usuário recebedor e as cobranças seguintes são pagas automaticamente por meio do Pix Automático.

## 6.4 Pix Automático

Além dos casos previstos no item QR Code Composto, utilizados durante as jornadas de adesão do Pix Automático, há outros casos de uso específicos deste produto.

### *6.4.1 Inclusão de recorrências*

**Aplicação:** usuários recebedores que desejam ofertar o pagamento via Pix Automático para seus clientes. A inclusão das recorrências é pré-requisito para o envio, aos clientes, de solicitações de autorização de pagamento usando o Pix Automático.

1. O usuário recebedor, por meio do software de automação utilizado por ele, acessa a API Pix para a criação de uma recorrência, informando, dentre outras coisas, dados sobre o contrato vinculado às cobranças, o recebedor, a data de início das cobranças recorrentes, a periodicidade, e se a recorrência permite ou não retentativas após o vencimento.

[88] Se por qualquer razão o PSP pagador não conseguir obter os dados da recorrência por meio da *location* contida no QR Code, esta etapa (e as seguintes) não deve ocorrer. Ou seja, a jornada apresentada ao usuário pagador deve ser a jornada de pagamento/agendamento de uma cobrança com vencimento.

*Serviços invocados:*

** POST /locrec → será retornado o identificador da "location" e a URL que será utilizada para exibir os parâmetros da recorrência.*

** POST /rec → antes de mostrar o QR Code Composto ao usuário pagador, usuário recebedor cria a recorrência com o identificador da "location" recebido do serviço anterior com todos os parâmetros da recorrência.*

2. Uma vez criada a recorrência, o usuário recebedor poderá enviar ao cliente solicitações de autorização de pagamento usando o Pix Automático por meio de QR Codes compostos (vide casos de uso da seção anterior) ou por meio do envio de uma solicitação de confirmação de recorrência.

   *Serviço invocado: POST /solicRec → o usuário recebedor envia uma solicitação de confirmação de recorrência informando o identificador da recorrência entre outros dados.*

### *6.4.2 Envio de solicitação de confirmação de recorrência*

**Aplicação:** usuários recebedores que tenham acordado diretamente com seus clientes que a forma de pagamento utilizada será o Pix Automático e desejam enviar o pedido de autorização para que o usuário pagador confirme pelo site ou app da sua instituição de relacionamento. Caracteriza a Jornada 1 de autorização do Pix Automático.

**Premissa:** deve existir uma recorrência já criada e ainda não autorizada.

1. O usuário recebedor, por meio do software de automação utilizado por ele, acessa a API Pix para a criação de uma solicitação de confirmação de recorrência, informando o identificador da recorrência criada previamente, a data de expiração da solicitação, a identificação do usuário pagador e os dados completos da conta transacional para a qual a solicitação de confirmação de recorrência será enviada.

   *Serviço invocado: POST /solicRec → o usuário recebedor envia uma solicitação de confirmação de recorrência informando o identificador da recorrência entre outros dados.*

2. O PSP Recebedor envia ao PSP Pagador a solicitação correspondente, via mensageria.
3. O PSP Pagador valida os dados do cliente e, caso estejam corretos, armazena a solicitação na lista de solicitações pendentes do usuário pagador e o notifica.
4. O PSP Pagador apresenta ao usuário pagador, juntamente com alguns parâmetros adicionais de configuração, a solicitação de autorização de pagamentos recorrentes referentes àquela recorrência.
5. O usuário pagador define as configurações desejadas e autoriza o pagamento das cobranças do contrato em questão via Pix Automático.
6. O PSP Pagador armazena os dados da autorização e avisa o PSP Recebedor que a recorrência foi confirmada.

*Serviço invocado: GET /solicRec/{idSolicRec} → a qualquer momento deste fluxo após o passo 1, o usuário recebedor pode consultar junto ao PSP Recebedor (fornecedor da API) a situação da solicitação.*

### *6.4.3 Agendamento de cobrança recorrente*

**Aplicação:** usuários recebedores que necessitam enviar regularmente ao seu PSP as cobranças recorrentes para pagamento por meio do Pix Automático.

**Premissas:** as recorrências correspondentes às cobranças precisam ter sido autorizadas pelos usuários pagadores e estarem vigentes na data de envio da cobrança recorrente.

1. O usuário recebedor, por meio do software de automação utilizado por ele, acessa a API Pix para a criação de uma cobrança recorrente, informando, dentre outras coisas, o identificador da recorrência, a data para pagamento, o valor do débito, os dados da conta transacional do recebedor e as informações do pagador.

   *Serviço invocado: POST /cobr → cria a cobrança recorrente com os dados pertinentes sendo gerado um txid para conciliação; ou PUT /cobr/{txid} → cria a cobrança recorrente com os dados pertinentes já informando um txid para conciliação.*

2. O PSP Recebedor valida as informações da cobrança com as informações da recorrência e, caso estejam coerentes, armazena os dados das cobranças recorrentes.
3. O PSP Recebedor verifica diariamente quais cobranças podem ser enviadas para o PSP Pagador, de acordo com a janela de agendamento. Ao chegar na data permitida para agendamento, o PSP Recebedor envia os dados da cobrança recorrente para o PSP Pagador.
4. O PSP Pagador valida as informações da cobrança com os dados da autorização concedida pelo usuário pagador e, caso esteja tudo certo, faz o agendamento da cobrança.
5. Após realizado o agendamento pelo PSP Pagador, a cobrança recorrente aparece nos extratos de lançamentos futuros do usuário pagador.

### *6.4.4 Solicitação de retentativa de pagamento de cobrança após o vencimento*

**Aplicação:** usuários recebedores que necessitam enviar retentativas de cobrança cujo pagamento não foi realizado na data prevista por falta de saldo ou insuficiência do limite transacional disponível relativos à conta do usuário pagador, ou ainda por falha operacional que impeça o envio da ordem de pagamento para liquidação.

**Premissas:** as recorrências correspondentes às cobranças devem permitir retentativas após o vencimento e a retentativa deve estar dentro dos parâmetros especificados na política de retentativa correspondente.

1. O usuário recebedor, por meio do software de automação utilizado por ele, acessa a API Pix para solicitar uma nova tentativa de cobrança, informando o txId da cobrança original e a nova data prevista para pagamento.

*Serviço invocado: POST /cobr/{txid}/retentativa/{data} → solicita a retentativa informando o txid da cobrança e a data prevista para liquidação da ordem de pagamento correspondente.*

2. O PSP Recebedor valida as informações para confirmar se a retentativa está dentro dos requisitos da política de retentativas.
3. O PSP Recebedor envia os dados da cobrança com a data de pagamento atualizada para o PSP Pagador.
4. O PSP Pagador valida as informações da cobrança com os dados da autorização concedida pelo usuário pagador e, caso esteja tudo certo, faz o agendamento da cobrança.

*Serviço invocado: GET /cobr/{txid} → a qualquer momento este endpoint retorna os dados da cobrança com os Pix de pagamentos correlatos.*

### *6.4.5 Cancelamento de agendamento de pagamento de cobrança recorrente*

**Aplicação:** usuários recebedores que desejam cancelar uma cobrança recorrente que foi enviada indevidamente.

**Premissas:** a solicitação de cancelamento deve ser feita dentro do horário máximo permitido (o PSP pagador deve receber a mensagem de cancelamento do débito agendado até as 22h00 do dia anterior à data prevista para o pagamento).

1. O usuário recebedor, por meio do software de automação utilizado por ele, acessa a API Pix para solicitar o cancelamento de uma cobrança recorrente que esteja com o pagamento agendado, informando o identificador da cobrança e o status Cancelada.

*Serviço invocado: PATCH /cobr/{txid} → informando o status para 'CANCELADA'.*

2. O PSP Recebedor verifica se a solicitação de cancelamento está dentro do prazo permitido e, caso esteja, cancela a cobrança.
3. Caso a cobrança já tenha sido enviada para agendamento do pagamento no PSP Pagador, o PSP Recebedor envia mensagem avisando sobre o cancelamento.
4. O PSP Pagador faz as validações necessárias e, caso esteja tudo certo, faz o cancelamento do agendamento.

*Serviço invocado: GET /cobr/{txid} → a qualquer momento este endpoint retorna os dados da cobrança.*

### *6.4.6 Cancelamento de recorrência*

**Aplicação:** usuários recebedores que desejam cancelar uma recorrência porque o contrato com o usuário pagador foi rescindido ou porque o usuário pagador gostaria de usar outra forma de pagamento.

**Premissas:** deve existir uma recorrência ativa vigente.

1. O usuário recebedor, por meio do software de automação utilizado por ele, acessa a API Pix para solicitar o cancelamento de uma recorrência, informando o identificador da recorrência e o status Cancelada.

   *Serviço invocado: PATCH /rec/{idRec} → informando o status para 'CANCELADA'.*

2. O PSP Recebedor cancela a recorrência.
3. O PSP Recebedor envia informação sobre o cancelamento da recorrência ao PSP Pagador.
4. O PSP Pagador cancela a autorização vinculada à recorrência cancelada.

## 6.5 Outros casos de uso

### *6.5.1 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.

### *6.5.2 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.

### *6.5.3 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}[89] → 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.

### *6.5.4 Configuração de Webhooks*

[89] Pode-se alterar tanto uma cobrança imediata quanto uma cobrança com vencimento.

**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.

### *6.5.5 Configuração de Webhooks de Cobranças Recorrentes*

**Aplicação**: para usuários recebedores que trabalham com grande número de cobranças recorrentes 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 /webhookcobr → Deve ser atribuída a URL base que receberá os callbacks enviados pelo PSP Recebedor ao usuário recebedor. Cada webhook está associado a um usuário recebedor de maneira que sejam informadas as devidas atualizações de cobranças recorrentes.*

2. A partir desse momento, o usuário Recebedor receberá chamadas nesse *endpoint* com a lista de respectivas cobranças recorrentes atualizadas.

### *6.5.6 Configuração de Webhooks de Recorrências*

**Aplicação**: para usuários recebedores que trabalham com grande número de recorrências 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 /webhookrec → Deve ser atribuída a URL base que receberá os callbacks enviados pelo PSP Recebedor ao usuário recebedor. Cada webhook está associado a um usuário recebedor de maneira que sejam informadas as devidas atualizações de recorrências.*

2. A partir desse momento, o usuário Recebedor receberá chamadas nesse *endpoint* com a lista de respectivas recorrências atualizadas.

# 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[90].
**Formato**: O formato de dados utilizados é o JSON[91].
**Protocolo**: a automação do usuário recebedor interage com a API utilizando *web services baseados em REST*[92] 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*[93].
2. O PSP deve implementar o framework *OAuth* 2.0 (*RFC* 6749)[94] com *TLS* mútuo (*mTLS – RFC* 8705[95]) para autenticação na API, conforme especificações abaixo:
    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.

---

[90] http://spec.openapis.org/oas/v3.0.3
[91] JSON: JavaScript Object Notation.
https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/JSON
[92] 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
[93] https://tools.ietf.org/html/bcp195#section-6.3
[94] https://tools.ietf.org/html/rfc6749
[95] https://tools.ietf.org/html/rfc8705

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*[96] e *CWE Top 25 Software Weaknesses*[97].
3. Possuir processo periódico de análise de vulnerabilidades, tanto estática como dinâmica da API.
4. Assegurar a segurança do desenvolvimento do *software* cliente[98] 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

[96] https://owasp.org/www-project-api-security/
[97] https://cwe.mitre.org/top25/archive/2020/2020_cwe_top25.html
[98] 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.

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.

# 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.

## 3. 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 | [0..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 *payload* 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 | [0..1] | `valor.abatimento.valorPerc` | String | |

| | | | | | |
|---|---|---|---|---|---|
| | valor absoluto ou percentual do valor original do documento | | | | |
| 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**[99].

- **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`**.

[99] 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).



| | |
|---|---|
| ⚠ | 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**

Além do campo `valor.original` a ser preenchido na criação da cobrança, os demais campos deverão seguir as instruções abaixo referenciadas. Como regra, não devem ser geradas cobranças cujo valor do desconto possa ser superior ao valor da cobrança original.

- **`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 dos juros, conforme a modalidade de juros escolhida. Se modalidade tiver sido preenchida, então este campo também deve ser preenchido. (NR)
- **`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>.[100] 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[101] 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 |

Os limites relativos ao tamanho de cada campo e à quantidade de elementos da lista estão tratados na especificação da API Pix[102].

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

---

[100] Vale lembrar que, no JSON, "txid" é escrito todo em minúsculas, enquanto a tag "TxId" da pacs.008 alterna maiúsculas e minúsculas.

[101] Importante destacar que é o texto digitado livremente pelo pagador que é enviado na pacs.008, e não o texto apresentado no campo *solicitacaoPagador*.

[102] Disponível em https://github.com/bacen/pix-api

```
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.

# 4. 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.

## 4.1. 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 $V_a$ 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 |

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

### *4.2.1. Fixo até a data informada*[103]

Nesse caso, o usuário recebedor terá que indicar descontos, até uma determinada data. O usuário recebedor poderá fixar até três datas menores ou iguais à data 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 até a data de vencimento, inclusive. 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.

[103] Modalidades de desconto com domínio "1" ou "2".

| | |
|---|---|
| ⚠ | 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. |

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.

### *4.2.2. Desconto por dia de antecipação*[104]

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: $D_\alpha = D_u$

[104] Modalidades de desconto com domínio "3" a "6".

$D_c$ 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*

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 |

## 4.3. 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.

*4.3.1. Valor absoluto diário:*

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

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

*Equação 8*

Onde:

$\boldsymbol{V_j}$: valor de juros aplicável pelo atraso no pagamento da cobrança;

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

$\boldsymbol{D_\alpha}$: 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.

*4.3.2. 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:

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

*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;

$\left(\frac{\left(\frac{I_j}{100}\right)}{n}\right) \times D_a$: fator de juros, a ser calculado com precisão de 6 casas decimais sem arredondamento (truncado);

$I_j$: 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 |

## 4.4. 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;

$k(D_\alpha)$: 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 |

# ANEXO IV – Pix Automático

## 1. Introdução

O presente anexo tem por objetivo apresentar as especificidades da API Pix para o Pix Automático.

## 2. Entidades

A especificação completa das entidades criadas para o Pix Automático encontra-se na especificação técnica detalhada da API Pix, disponível em https://github.com/bacen/pix-api. Nesta seção explicaremos alguns pontos específicos, visando facilitar o entendimento.

### 2.1. Rec

No Pix Automático, o usuário recebedor deve enviar ao seu PSP as informações de recorrência, que identificam o recebedor, o objeto do débito recorrente, a periodicidade e a data de início das cobranças, dentre outras informações. Essas informações serão apresentadas pelo PSP Pagador ao usuário pagador para que este possa autorizá-lo a realizar em sua conta débitos que estejam em conformidade com as informações da recorrência.

#### *2.1.1. Atributo idRec*

A regra de formação do identificador da recorrência é:

`RRxxxxxxxxxyyyyMMddkkkkkkkkkk` (29 caracteres alfanuméricos; “case sensitive”), sendo:
• R ou C – fixo (1 caractere). “R” caso a recorrência tenha sido criada dentro do Pix, ou "C" caso tenha sido criada pela trilha do Open Finance;
• R ou N – fixo (1 caractere). “R” caso a recorrência permita retentativas de realização do débito em data posterior à data de agendamento original, ou “N” caso não permita retentativas.
• xxxxxxxx – identificação do agente que presta serviço para o usuário recebedor que gerou o <Id>, podendo ser: o ISPB do participante direto, o ISPB do participante indireto ou os 8 primeiros caracteres do CNPJ do prestador de serviço de iniciação (8 caracteres alfanuméricos [0-9A-Z]{8});
• yyyyMMdd – data (8 caracteres) de criação da recorrência;
• kkkkkkkkkk – sequencial criado pelo agente que gerou o <Id> (11 caracteres alfanuméricos [a-z|A-Z|0-9]). Deve ser único dentro de cada “yyyyMMdd”.

Assim, o ID da recorrência deverá ter um dos quatro padrões abaixo:
• `RRxxxxxxxxxyyyyMMddkkkkkkkkkk`; para recorrência criada dentro do Pix e que permite retentativas após o vencimento;
• `RNxxxxxxxxxyyyyMMddkkkkkkkkkk`; para recorrência criada dentro do Pix e que não permite retentativas após o vencimento;
• `CRxxxxxxxxxyyyyMMddkkkkkkkkkk`; para recorrência criada pela trilha do Open Finance e que permite retentativas após o vencimento; ou
• `CNxxxxxxxxxyyyyMMddkkkkkkkkkk`; para recorrência criada pela trilha do Open Finance e que não permite retentativas após o vencimento.

Convém ressaltar, entretanto, que as recorrências criadas e gerenciadas por intermédio da API Pix devem ter seus identificadores iniciados pela letra ‘R’, ou seja, identificadores das recorrências criadas via a API Pix não poderão ser iniciados com o a letra ‘C’, para recorrências originadas no Open Finance. Portanto, as alterações e consultas via API Pix de recorrências cujos identificadores são iniciados com ‘C’ não serão permitidas.

#### *2.1.2. Atributo recebedor*

Este atributo contém a identificação do usuário recebedor e do convênio entre o usuário recebedor e seu PSP (opcional). A identificação do usuário recebedor não será informada pelo próprio usuário no momento de criação da recorrência. O PSP Recebedor deverá preenchê-la de acordo com os dados cadastrais do usuário que está se conectando com ele via API Pix.

#### *2.1.3. Atributo calendario*

O calendário contém datas relativas à configuração da recorrência, como data inicial e data final de vigência.

#### *2.1.4. Atributo atualizacao*

Este atributo contém o histórico das alterações de status da recorrência. Todas as mudanças de status feitas deverão ser refletidas aqui.

#### *2.1.5. Atributo politicaRetentativa*

Este atributo guarda a informação sobre a habilitação ou não de retentativas após o vencimento.

O usuário recebedor deve escolher uma opção, dentre as disponíveis:

- NAO_PERMITE : Não permite retentativas.
- PERMITE_3R_7D: Permite até 3 retentativas em dias diferentes no intervalo de até 7 dias corridos contados a partir da data de liquidação prevista na instrução de pagamento original (acrônimo 3R_7D).

#### *2.1.6. Atributo encerramento*

Contém informações sobre a rejeição ou o cancelamento de uma recorrência.

A rejeição acontece apenas na Jornada 1, no momento da solicitação de autorização, caso o usuário pagador rejeite a recorrência ou na tentativa de solicitar recorrências para participantes que não oferecem o Pix Automático. (NR) As informações relativas à rejeição estão dentro da estrutura “encerramento.rejeicao”.

O cancelamento da recorrência pode ser solicitado pelo usuário ou PSP recebedores ou pelo usuário ou PSP pagadores. As informações sobre o cancelamento estão dentro da estrutura “encerramento.cancelamento”.

Conforme descrito acima, as duas opções são excludentes. Portanto, o atributo “encerramento” é do tipo “oneOf”.

Os atributos “codigo" devem corresponder aos mesmos utilizados nas mensagens de rejeição e de cancelamento de recorrência. O PSP Recebedor deve utilizar a tabela de domínios do Catálogo de Mensagens do SPI para preencher o valor do atributo “descricao”.

Nem todos os códigos recebidos nas mensagens geram rejeição ou cancelamento (alguns podem dizer respeito a erros de comunicação, por exemplo). Assim, os campos “codigo" da API contêm apenas os domínios que correspondem à rejeição ou ao cancelamento de fato das entidades.

O atributo cancelamento.solicitante contém o tipo de pessoa que solicitou o cancelamento, seja usuários finais ou PSPs. Ele deverá ser preenchido pelo PSP de acordo com o contexto da solicitação.

#### *2.1.7. Atributo ativacao*

Este atributo contém informações sobre a jornada de autorização que foi utilizada para a confirmação da recorrência.

O campo tipoJornada será preenchido de acordo com o código da mensagem de confirmação da autorização (PAIN.012), que identificará se foi Jornada 1, Jornada 2, Jornada 3 ou Jornada 4. O PSP Recebedor não deve fazer nenhuma validação sobre este campo. Ele deve aceitar a informação enviada pelo PSP Pagador. Na criação da recorrência este campo deverá ser preenchido com o valor AGUARDANDO_DEFINICAO.

O campo dadosJornada varia de acordo com a jornada usada e deve ser preenchido apenas se for aplicável. Atualmente há somente o campo “txId”, que armazenará o identificador da transação que corresponde ao pagamento imediato realizado no momento da autorização da recorrência. Ele deverá ser preenchido pelo usuário recebedor que deseje utilizar a Jornada 3 de autorização.

## 2.2. SolicRec

A solicitação de confirmação de recorrência é específica para a Jornada 1 de autorização do Pix Automático.

#### *2.2.1. Atributo idSolicRec*

O identificador da solicitação de recorrência deve ser gerado pelo PSP Recebedor e tem a seguinte regra de formação:

SCxxxxxxxxyyyyMMddkkkkkkkkkkk (29 caracteres alfanuméricos; “case sensitive”), sendo:
• SC – fixo (2 caracteres);
• xxxxxxxx – ISPB do agente que envia a mensagem pain.009 de solicitação de confirmação da recorrência (8 caracteres alfanuméricos [0-9A-Z]{8});
• yyyyMMdd – data (8 caracteres) de criação da mensagem pain.009 de solicitação de confirmação da recorrência;
• kkkkkkkkkkk – sequencial criado pelo agente que gerou a mensagem de solicitação de confirmação da recorrência (11 caracteres alfanuméricos [a-z|A-Z|0-9]). Deve ser único dentro de cada “yyyyMMdd”.

#### *2.2.2. Atributo destinatario*

Este campo contém a identificação da pessoa (CPF ou CNPJ) e da respectiva conta para a qual a solicitação será enviada. No contexto da Jornada 1 de autorização, é preciso informar os dados bancários para que o PSP Recebedor possa encaminhar a solicitação de confirmação de recorrência para a conta correta, uma vez que o Pix Automático não utiliza chaves Pix.

#### *2.2.3. Atributo atualizacao*

Este atributo contém o histórico das alterações de status da solicitação de confirmação de recorrência. Todas as mudanças de status feitas deverão ser refletidas aqui.

#### *2.2.4. Atributo recPayload*

Este atributo contém as mesmas informações do payload da recorrência usado na leitura do QR Code composto para as outras jornadas de autorização. Quando se trata de consulta a uma solicitação de confirmação de recorrência não faz sentido apresentar informações da recorrência posteriores à confirmação de autorização, por isso o uso do atributo recPayload e não os campos da própria rec neste contexto.

#### *2.2.5. Atributo encerramento (NR)*

Este atributo contém as informações sobre o encerramento de uma solicitação de recorrência. O objetivo deste campo é fornecer informações, com o nível de abstração apropriado, para que o PSP Recebedor possa indicar ao usuário da API Pix eventuais ações corretivas que precisem ser realizadas por ele, como corrigir as informações de agência e conta da solicitação.

### 2.3. CobR

Uma vez que o usuário pagador conceda a autorização, as cobranças recorrentes passarão a ser enviadas automaticamente pelo usuário recebedor, com a periodicidade acordada, e pagas automaticamente mediante débitos na conta do pagador realizados pelo PSP Pagador, na data especificada, desde que a cobrança esteja em conformidade com os parâmetros da respectiva autorização.

O PSP Pagador só deverá realizar o agendamento de pagamentos de cobranças recorrentes que estejam em conformidade com os parâmetros da recorrência e com os demais parâmetros definidos pelo usuário pagador no momento da autorização.

#### *2.3.1. Atributo infoAdicional*

Este campo contém informações específicas da fatura em questão. Será informado pelo usuário recebedor ao usuário pagador.

#### *2.3.2. Atributo calendario*

Como nas outras entidades, este atributo reúne as datas relacionadas à cobrança em questão.

### *2.3.3. Atributo recebedor*

Contém informações bancárias, que deverão ser informadas em todas as cobranças. A razão para essas informações estarem nas cobranças recorrentes e não na recorrência é dar mais flexibilidade ao usuário recebedor, caso ele queira mudar a conta de recebimento. O Pix Automático não utiliza chaves Pix em nenhuma das pontas, portanto é necessário ter sempre os dados bancários completos do recebedor e do pagador.

O PSP Recebedor deverá validar se o recebedor é o titular da conta informada. Caso não seja, o PSP Recebedor não deve permitir a criação da cobrança recorrente.

### *2.3.4. Atributo politicaRetentativa*

Este atributo corresponde ao valor do atributo politicaRetentativa da Rec vigente para a cobrança recorrente. O objetivo é guardar um histórico da política de retentativa que se aplicava à cobrança à época em que ela foi paga.

### *2.3.5. Atributo encerramento*

Contém informações detalhadas sobre a rejeição ou o cancelamento de uma cobrança recorrente. Por exemplo: qual o código de erro da rejeição, quem solicitou o cancelamento etc.

Os atributos “codigo" devem corresponder aos mesmos utilizados nas mensagens de rejeição e de cancelamento de cobrança recorrente. O PSP Recebedor deve utilizar a tabela de domínios do Catálogo de Mensagens do SPI para preencher o valor do atributo “descricao”.

Nem todos os códigos recebidos nas mensagens geram rejeição ou cancelamento (alguns podem dizer respeito a erros de comunicação, por exemplo). Assim, os campos “codigo" da API contêm apenas os domínios que correspondem à rejeição ou ao cancelamento de fato das entidades.

O atributo cancelamento.solicitante contém o tipo de pessoa que solicitou o cancelamento, seja usuários finais ou PSPs. Ele deverá ser preenchido pelo PSP de acordo com o contexto da solicitação.

### *2.3.6. Atributo devedor*

Este atributo contém informações complementares sobre o devedor da cobrança. A identificação do devedor consta da Rec.

### *2.3.7. Atributo tentativas*

Este atributo contém informações sobre todas as tentativas de cobrança realizadas. Isso inclui a tentativa original de pagamento na data de liquidação prevista, as retentativas intradia por erro na liquidação, se houver, e as retentativas que ocorreram após o vencimento, caso existam. Assim, cada registro no array “tentativas” contém as informações detalhadas sobre um agendamento específico.

O atributo “atualizacao" contém o histórico de alterações de status que ocorrem em uma tentativa (ver mais detalhes no tópico Atributo status x tentativas.status abaixo).

### *2.3.8. Atributo status x tentativas.status*

A COBR apresenta dois níveis de status:

- status da própria cobrança (atributo status): indica os estados pelos quais a COBR passa, sem detalhar os estados intermediários entre o agendamento e o pagamento. Este detalhamento está no atributo status da respectiva tentativa de cobrança.
- status da tentativa (atributo tentativas.status): a partir do envio de uma instrução de pagamento para agendamento (seja o envio original ou uma retentativa), os status transitórios até que haja alguma definição sobre o pagamento ou não da cobrança ficam dentro do atributo tentativas.status.

As transições de estado da COBR e das tentativas serão mais bem compreendidas no tópico Estados da CobR.

### *2.3.9. Atributo ajusteDiaUtil*

Este atributo deve ser definido pelo usuário recebedor no momento da criação da cobrança. Por *default*, seu valor é *true*. Isso significa que, caso a data de vencimento da cobrança seja um dia não útil, o PSP recebedor deve ajustar sua data prevista para liquidação para o próximo dia útil. Esse ajuste deve considerar eventuais feriados locais do usuário pagador com base no seu código de município, que foi informado no momento da confirmação da recorrência.

Se o usuário recebedor não desejar que esse ajuste seja feito, ou seja, se ele quiser que a data prevista para liquidação da cobrança corresponda à sua data de vencimento, independentemente de ela ser um dia útil ou não, deve alterar o valor desse campo para *false*. Nesse caso, salientamos que é dever do usuário recebedor se certificar de que possui amparo legal para realizar tentativas de liquidação de cobranças em dias não úteis.

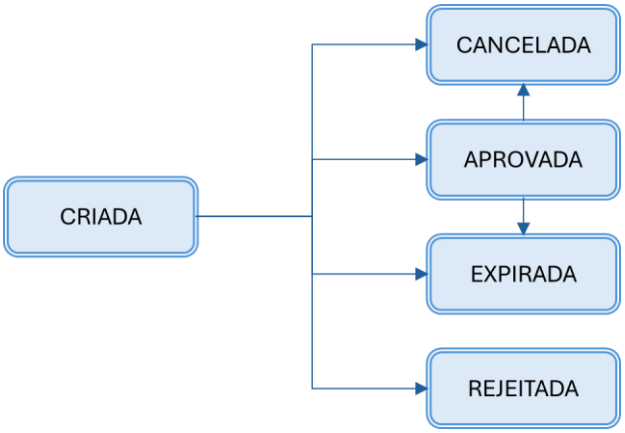
# 3. Estados das entidades

O objetivo desta seção é esclarecer as transições de estados possíveis para as entidades do Pix Automático. A definição dos estados está descrita na seção Definições das entidades.

## 3.1. Estados da Rec

As transições de estados possíveis para a Rec são:

DIAGRAMA DE ESTADOS – REC



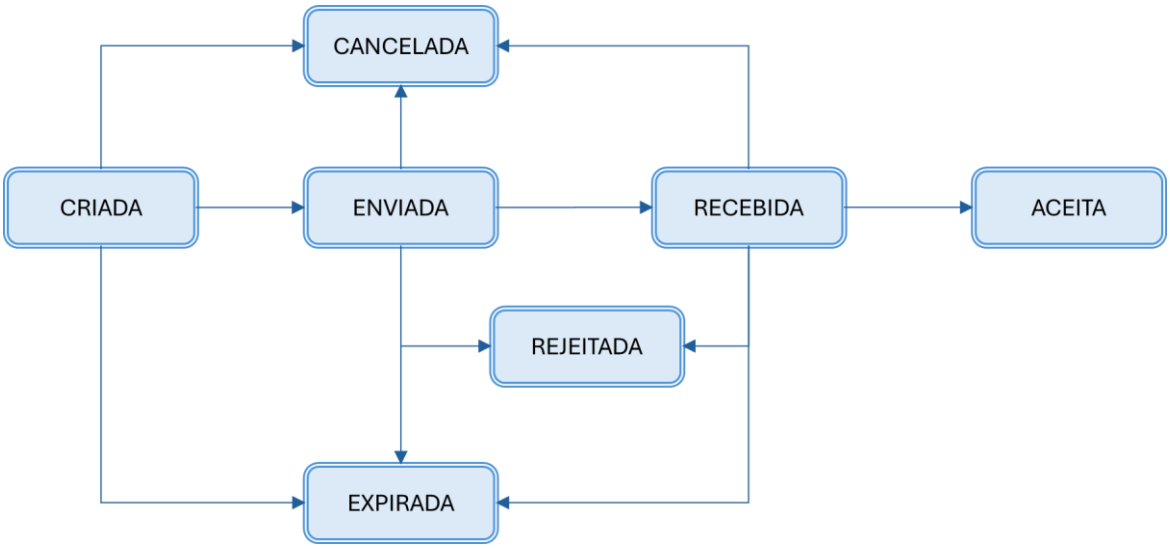

O estado Rejeitada ocorre apenas na Jornada 1. As jornadas de autorização via QR Code retornam apenas a informação de confirmação da recorrência, se houver. Não há mensagem de retorno de negativa de recorrência em jornadas envolvendo QR Code.

O estado Expirada ocorre após a data final da recorrência (atributo calendario.dataFinal). Essa alteração de estado deve ser feita pelo PSP Recebedor ao consultar uma recorrência criada ou aprovada e verificar que a sua validade já passou, mas o estado ainda não havia sido alterado.

## 3.2. Estados da SolicRec

As transições de estado permitidas para a SolicRec são:

DIAGRAMA DE ESTADOS – SOLICREC




Uma solicitação de confirmação de recorrência pode ser cancelada caso a recorrência à qual ela está vinculada seja cancelada ou confirmada por alguma outra jornada de adesão. Nesses casos, quando ocorre o cancelamento ou a confirmação de uma recorrência, o PSP Recebedor deve cancelar qualquer solicitação que ainda esteja pendente de autorização, se houver. Ao cancelar uma solicitação, o PSP Recebedor deve comunicar o PSP Pagador (conforme fluxo de mensagens para cancelamento de solicitação de confirmação de recorrência), para que a solicitação seja excluída da lista de autorizações pendentes do usuário pagador.

Há ainda outro cenário de cancelamento de uma solicitação de confirmação de recorrência: na jornada 1, o limite máximo de tempo que o PSP do recebedor deve aguardar para o recebimento da mensagem pain.012 confirmando o recebimento da solicitação de permissão (mensagem pain.009) pelo PSP do pagador é de 1 minuto. Dentro desse tempo, o PSP do Recebedor pode reenviar a pain.009 quantas vezes entender necessário. Após esse tempo, configura-se o timeout, a SolicRec associada deve ser cancelada pelo PSP do Recebedor, que deve providenciar o envio de uma pain.011 ao PSP pagador.

Uma solicitação pode ser rejeitada pelo PSP Pagador por algum problema (como no caso de conta inexistente) ou pelo usuário pagador, no caso de não reconhecimento do usuário recebedor, por exemplo.

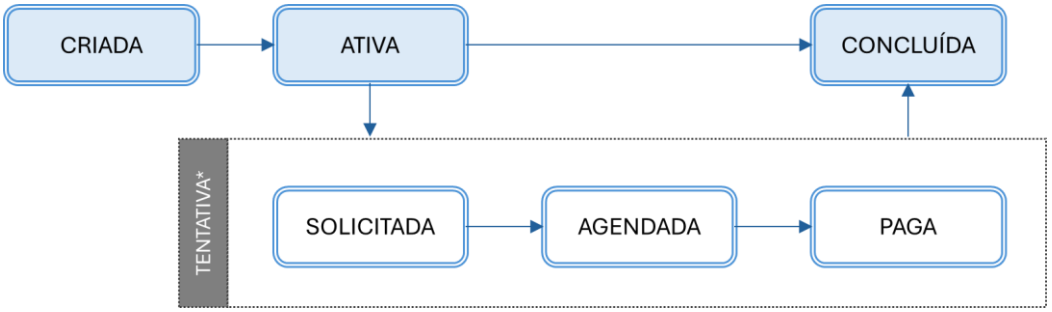
## 3.3. Estados da CobR

Conforme descrito no tópico Atributo status x tentativas.status, para visualizar o estado atual da COBR precisamos considerar também o estado da última tentativa de cobrança, constante do atributo "tentativas".

Visando facilitar a compreensão, especificamos as transições de estado da cobrança recorrente de acordo com cada cenário possível.

### *3.3.1. Cobrança paga na primeira tentativa (data normal)*

DIAGRAMA DE ESTADOS – COBR PAGA NA DATA NORMAL




(*) cada tentativa corresponde a um elemento dentro do array 'cobr.tentativas'. Esse array contém tanto a primeira tentativa de cobrança quanto as retentativas.

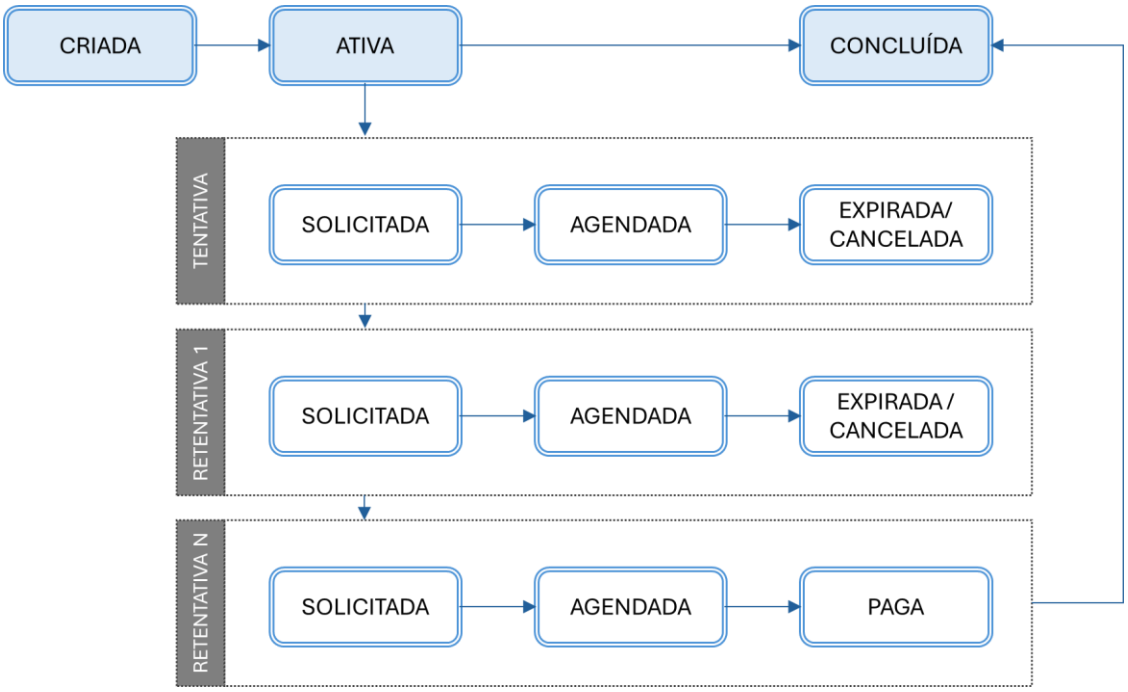
Após a cobrança ser enviada para agendamento do pagamento, o estado passa a ser atualizado no atributo tentativas.status. O estado da CobR será atualizado apenas ao final, caso a tentativa seja paga com sucesso ou expire.

Ao ser enviada para o PSP Pagador, a tentativa passa para o estado SOLICITADA. Após a confirmação de agendamento, ela fica no estado AGENDADA.

O pagamento de uma tentativa de cobrança gera a atualização da COBR para o estado CONCLUIDA.

### *3.3.2. Cobrança paga em uma retentativa*

DIAGRAMA DE ESTADOS – COBR PAGA, COM RETENTATIVA




Uma cobrança pode ser paga com retentativa nas seguintes situações:

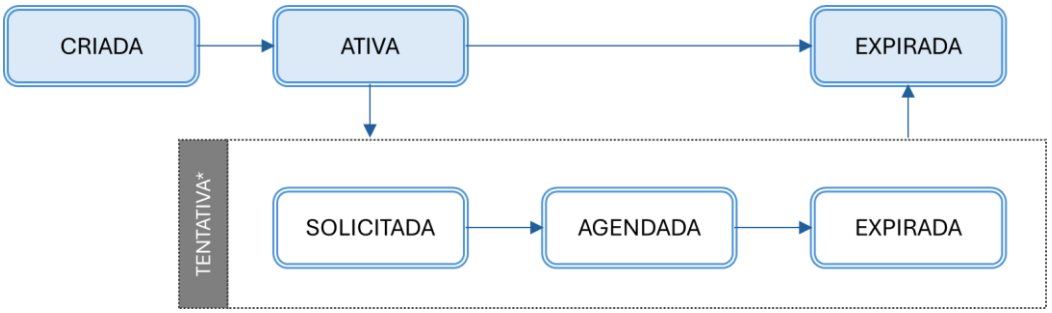
- enquanto o prazo estipulado na política de retentativas não expira e o pagamento não é realizado, a COBR continua no estado ATIVA, aguardando pagamento, e novas tentativas de cobrança após o vencimento podem ser feitas.
- caso ocorra falha na liquidação e seja enviada uma nova retentativa intradia por erro na liquidação.

O fluxo de estados das retentativas é o mesmo da tentativa original.

O PSP Recebedor deve alterar o estado da tentativa para EXPIRADA ao verificar que o atributo “dataLiquidacao” já passou, mas o estado ainda não havia sido atualizado.

### *3.3.3. Cobrança não paga (sem retentativas)*

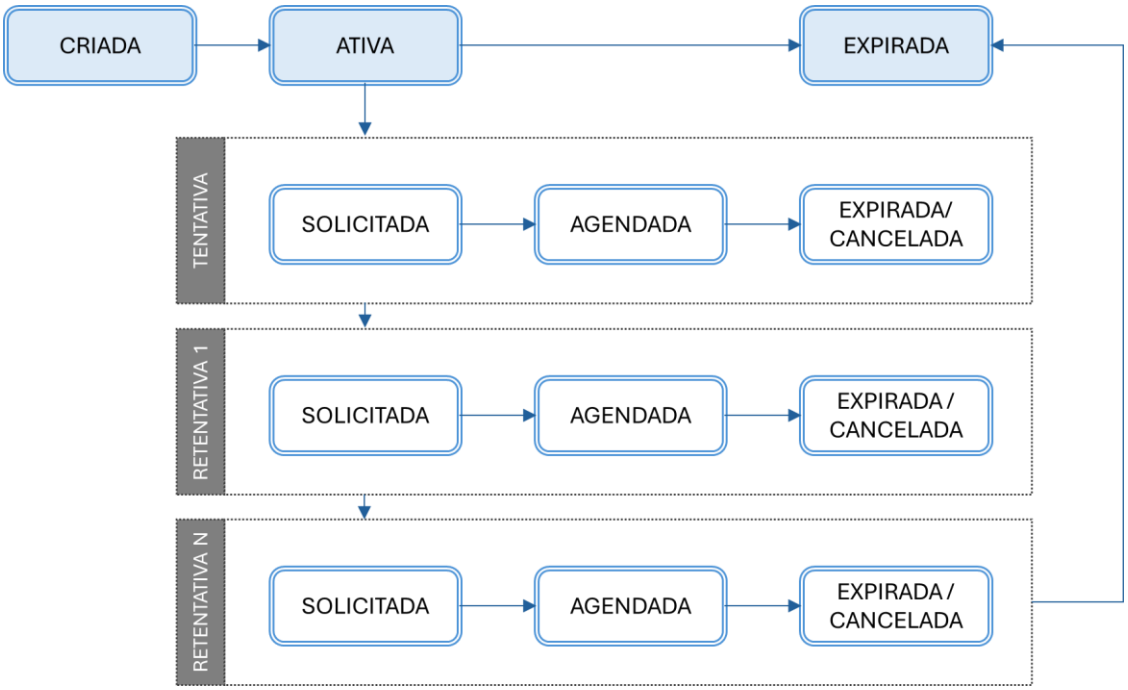
DIAGRAMA DE ESTADOS – COBR NÃO PAGA, SEM RETENTATIVA




Caso a tentativa original de uma cobrança expire, se a recorrência não permitir retentativas após o vencimento a COBR deverá expirar também.

### *3.3.4. Cobrança não paga (com retentativas)*

DIAGRAMA DE ESTADOS – COBR NÃO PAGA, COM RETENTATIVA

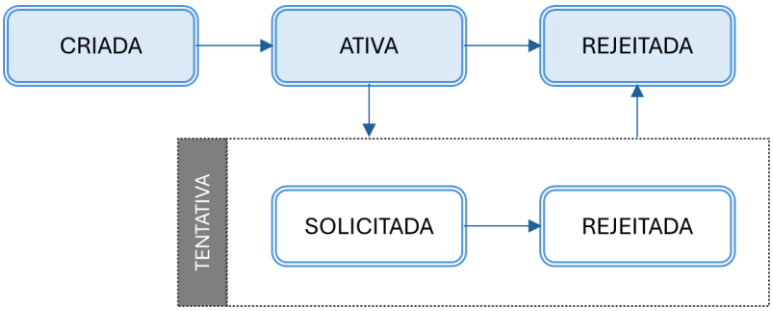


Caso a recorrência permita retentativas após o vencimento, a COBR deverá ter seu estado alterado para EXPIRADA caso todas as retentativas tenham expirado (ou o prazo permitido ou a quantidade possível de retentativas tenham acabado).

A cobrança também expira caso ocorra falha em todas as retentativas intradia enviadas por erro na liquidação.

### *3.3.5. Agendamento rejeitado pelo PSP Pagador*

DIAGRAMA DE ESTADOS – COBR REJEITADA PELO PSP PAGADOR




A rejeição de uma tentativa de agendamento de pagamento da cobrança nem sempre gera a rejeição da cobrança. A depender do código de rejeição informado na pain.014, altera-se o status da cobr ou da tentativa de agendamento conforme tabela abaixo. O atributo COBR.encerramento.rejeicao.codigo contém a lista de códigos que geram a rejeição da cobrança.

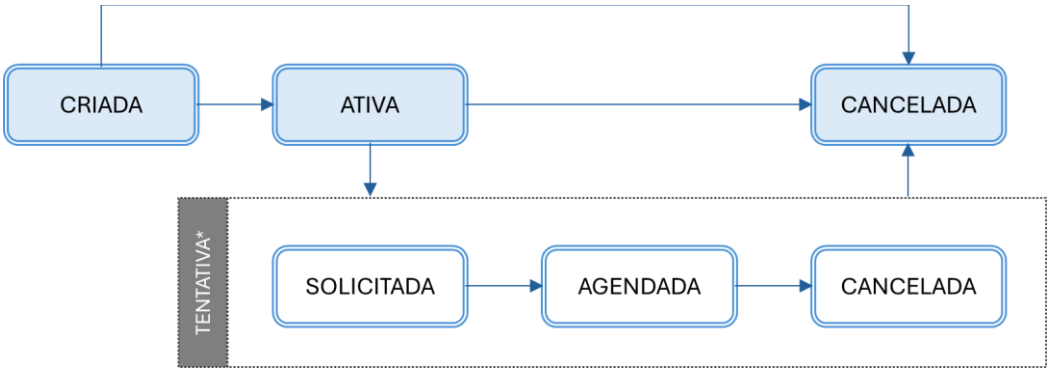
CÓDIGOS DE ERRO DA PAIN.014 – IMPACTOS NO STATUS DA COBR E TENTATIVA

| Código de Erro | Impacto na COBR | Impacto na Tentativa |
|---|---|---|
| AB10 | - | status = REJEITADA |
| AC05 | status = REJEITADA | status = REJEITADA |
| AC06 | - | status = REJEITADA |
| AM02 | - | status = REJEITADA |
| AM09 | status = REJEITADA | status = REJEITADA |
| DENC | status = REJEITADA | status = REJEITADA |
| DS27 | status = REJEITADA | status = REJEITADA |
| DTED | status = REJEITADA | status = REJEITADA |
| DTNT | - | status = REJEITADA |
| FBRD | - | status = REJEITADA |
| IRNT | - | status = REJEITADA |
| MIDI | status = REJEITADA | status = REJEITADA |
| MSUC | status = REJEITADA | status = REJEITADA |

| | | |
|---|---|---|
| NIEC | - | status = REJEITADA |
| NIPA | - | status = REJEITADA |
| NITX | status = REJEITADA | status = REJEITADA |
| QUNT | - | status = REJEITADA |
| RC09 | status = REJEITADA | status = REJEITADA |
| UDEI | status = REJEITADA | status = REJEITADA |

### *3.3.6. Cobrança cancelada pelo recebedor*

DIAGRAMA DE ESTADOS – COBR CANCELADA PELO PSP PAGADOR

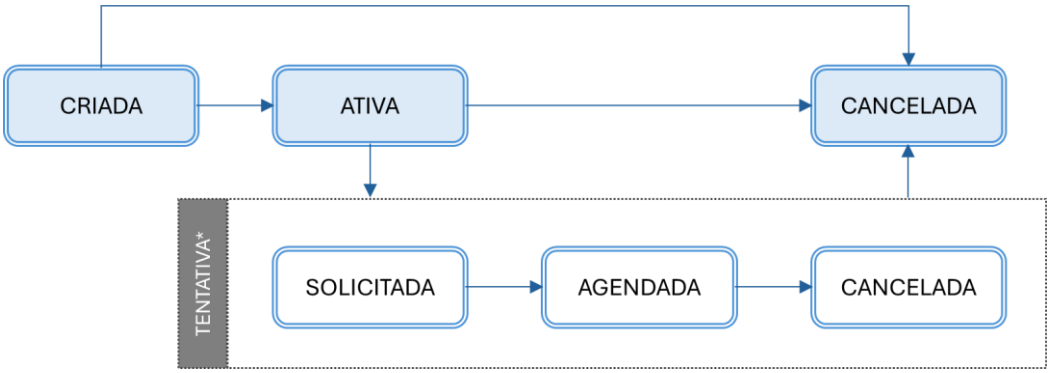



O recebedor pode cancelar a cobrança até a véspera da data prevista para liquidação. Após essa data, não é mais possível o cancelamento pela ponta recebedora, ainda que não tenha ocorrido o pagamento e estejamos no período das retentativas após o vencimento.

O cancelamento da cobrança gera o cancelamento da tentativa de cobrança atual.

### *3.3.7. Agendamento de cobrança cancelado pelo usuário pagador*

DIAGRAMA DE ESTADOS – COBR CANCELADA PELO PAGADOR




Antes da data de liquidação original, a tentativa de pagamento pode ser cancelada pelo usuário pagador. Ele não pode cancelar o agendamento de uma retentativa após o vencimento.

O cancelamento do agendamento do pagamento da cobrança (tentativa) gera o cancelamento da COBR.

# 4. Regras de negócio

## 4.1. Recorrência

### *4.1.1. Alteração de recorrência*

Após a criação de uma recorrência, o usuário recebedor poderá alterar apenas as informações constantes do PATCH. Não é permitido alterar os demais campos da recorrência. Caso o usuário recebedor queira alterar algum parâmetro que não esteja no PATCH, ele deverá cancelar a recorrência e solicitar nova confirmação do usuário pagador, com os parâmetros atualizados.

Dentre os campos que podem ser alterados pelo usuário recebedor, os campos calendario.dataInicial e ativacao.dadosJornada.txid só podem ser modificados se a recorrência ainda não tiver sido confirmada pelo usuário pagador. Após a confirmação, somente os campos loc e vinculo.devedor.nome podem ser objeto de alterações pelo usuário recebedor.

### *4.1.2. Confirmação de recorrência*

Sempre que alterar o status de uma recorrência para “aprovada”, o PSP Recebedor deve verificar se existe uma solicitação de confirmação de recorrência pendente de autorização. Se houver, ele deve cancelá-la e deve comunicar o PSP Pagador (conforme fluxo de mensagens para cancelamento de solicitação de confirmação de recorrência), para que a solicitação seja excluída da lista de autorizações pendentes do usuário pagador.

### *4.1.3. Cancelamento de recorrência*

Ao cancelar uma recorrência que estava pendente de autorização, o PSP Recebedor deve cancelar também qualquer solicitação de confirmação de recorrência que esteja pendente de autorização, se houver. O PSP Recebedor deve comunicar o PSP Pagador (conforme fluxo de mensagens para cancelamento de solicitação de confirmação de recorrência), para que a solicitação seja excluída da lista de autorizações pendentes do usuário pagador.

### *4.1.4. Recorrências com retentativas*

As retentativas após o vencimento são novas tentativas de cobrança em datas posteriores à data prevista para liquidação da tentativa original, caso o pagamento não tenha sido realizado na data prevista, por falta de saldo na conta, insuficiência do limite transacional ou falha operacional que impeça o envio da ordem de pagamento para liquidação.

O usuário recebedor poderá informar, na criação de uma recorrência, se ela permite ou não retentativas após o vencimento. Esta informação será apresentada ao usuário pagador, no momento da autorização. O usuário pagador não tem opção de negar apenas esta opção. Caso ele não aceite esta condição, ele deverá recusar toda a recorrência.

Atualmente há apenas uma regra possível, na qual todas as recorrências com retentativas após o vencimento devem se enquadrar: até 3 novas tentativas em dias diferentes dentro do intervalo de 7 dias corridos após a data prevista para liquidação da tentativa original.

O PSP Recebedor deverá negar solicitações de retentativa após o vencimento para recorrências que não possuam esta opção ou que extrapolem a regra citada acima.

O usuário recebedor deverá enviar o pedido de retentativa no prazo definido pelo PSP Recebedor, de forma que o PSP pagador o receba até as vinte e três horas e cinquenta e nove minutos do dia imediatamente anterior ao dia da liquidação informado para a retentativa.

#### *4.1.5. Confirmação da recorrência na Jornada 3*

Na Jornada 3 de autorização, que é composta pelo pagamento imediato da primeira cobrança juntamente com a autorização de pagamento recorrente, o PSP Recebedor só deve permitir a confirmação da recorrência caso o pagamento da cobrança imediata tenha sido confirmado. A forma de iniciação do pagamento da cobrança imediata deve ser identificada, pelo PSP Pagador, como 'QRDN' na mensagem PACS.008.

Caso o pagamento tenha sido efetuado e tenha ocorrido erro na autorização, o usuário recebedor deverá utilizar outra jornada de adesão para obter a autorização do usuário pagador. O recebedor poderá também enviar novamente o pedido de autorização com o pagamento imediato no período seguinte, caso deseje utilizar a Jornada 3 apenas.

## 4.2. Solicitação de confirmação de recorrência

#### *4.2.1. Alteração de solicitação de confirmação de recorrência*

A solicitação pode ser cancelada, por meio da alteração de status.

Após recebida pelo PSP Pagador, a solicitação de confirmação de recorrência não pode ser corrigida/alterada no PSP Recebedor. Se o usuário recebedor informou algum dado incorreto na recorrência ou na solicitação, ele deve cancelar a recorrência e criar uma nova, juntamente com uma nova solicitação.

Ao cancelar uma recorrência, o PSP Recebedor deve cancelar também a solicitação de recorrência que esteja pendente de decisão pelo usuário final.

## 4.3. Cobrança recorrente

#### *4.3.1. Data de início dos ciclos no Pix Automático*

O critério para definir a data de vencimento de uma cobrança quando a data esperada não existir deve constar do contrato firmado entre os usuários pagador e recebedor. Por exemplo, em uma recorrência mensal, se a cobrança vence todo dia 30, o contrato deve dizer se o vencimento referente ao mês de fevereiro ocorrerá no dia 28/fev ou 1º/mar.

No âmbito do Pix Automático, o critério a ser utilizado para se determinar o início de cada ciclo, caso a data esperada não exista, é o da data existente imediatamente anterior. Dessa forma, para uma recorrência mensal com primeiro pagamento previsto para o dia 30/dez/2024, teríamos os seguintes ciclos:

De 30/dez/2024 a 29/jan/2025;
De 30/jan/2025 a 27/fev/2025;
De 28/fev/2025 a 29/mar/2025;
De 30/mar/2025 a 29/abr/2025;
...

Esse critério permite que as cobranças com vencimento esperado para o dia 30/fev, que é uma data inexistente, possam apresentar vencimento no dia 28/fev[105], 1º/mar ou em qualquer outra data do terceiro ciclo acima, desde que a data prevista para liquidação não ultrapasse o dia 29/mar.

Lembramos que é permitida apenas uma cobrança por ciclo, ou seja, devem ser rejeitadas novas tentativas de agendamento de pagamento de cobranças para uma mesma recorrência no mesmo ciclo, excetuadas aquelas referentes às instruções de pagamento (pain.013) com finalidadeDoAgendamento igual a RIFL (Reenvio da instrução de pagamento devido a erro na liquidação) ou NTAG (Agendamento de nova tentativa de pagamento pós vencimento).

#### *4.3.2. Alteração de cobrança recorrente*

A única alteração permitida pelo usuário recebedor em uma cobrança recorrente é o seu cancelamento. Só é permitido o cancelamento de cobranças até um dia antes da data prevista para liquidação da primeira tentativa. Após esse prazo o cancelamento não é permitido, devendo assim a cobrança expirar pelo decorrer do prazo.

Caso o usuário recebedor deseje corrigir alguma informação na cobrança, ele deve cancelar a cobrança gerada anteriormente e criar uma nova, com um novo txId. Caso a cobrança cancelada já tenha sido enviada para agendamento, o PSP Recebedor deverá enviar ao PSP Pagador a mensagem de cancelamento e fazer um novo agendamento, com a nova cobrança.

Caso o recebedor crie uma nova cobrança para corrigir a anterior, ele deve observar se a data de vencimento está dentro da janela de agendamento permitida. Caso necessário, ele deve postergar a data de vencimento para que a nova cobrança respeite os prazos mínimos definidos para o agendamento.

[105] Em se tratando de um ano bissexto, deve-se considerar o dia 29/fev ao invés de 28/fev.

#### *4.3.3. Retentativa intradia por erro na liquidação*

Caso receba uma mensagem de cancelamento do agendamento por erro na liquidação, o PSP Recebedor deve criar uma nova tentativa do tipo RIFL, com um novo endToEndId, e enviar a nova mensagem de agendamento intradia por erro na liquidação.

#### *4.3.4. Retentativa após o vencimento (de acordo com a política de retentativa)*

Caso a recorrência permita retentativas após o vencimento da cobrança, o usuário recebedor pode solicitar que o PSP Recebedor faça uma nova tentativa de cobrança para uma data futura, caso o pagamento não tenha sido realizado na data original prevista para a liquidação por falta de saldo na conta do usuário pagador, insuficiência de limite transacional ou falha operacional que tenha impedido o envio da ordem de pagamento para liquidação.

Neste caso, o recebedor deve acessar o endpoint *POST /cobr/{txid}/retentativa/{data}.*

O PSP Recebedor deve validar se a retentativa solicitada está dentro das regras permitidas pela política de retentativa vigente para a cobrança.

## 4.4. QR Code composto

O QR Composto é usado nas jornadas de autorização 2, 3 e 4.

O usuário recebedor deverá criar o QR Composto das seguintes formas, considerando a jornada de autorização desejada:

| Jornada de autorização | Composição do QR Composto |
|---|---|
| Jornada 2 | QR Code apenas com location da recorrência |
| Jornada 3 | QR Code com location da cobrança imediata (Cob) + location da recorrência |
| Jornada 4 | QR Code com:<br>• location da cobrança com vencimento (CobV) + location da recorrência; ou<br>• dados do QR Code estático + location da recorrência |

# 5. Tags específicas para o Pix Automático

Tags criadas para a implementação do Pix Automático na API Pix:

| Tag | Função |
|---|---|
| **Rec** | Criação do payload da recorrência e gestão de recorrências. |
| **RecPayload** | Utilizados pelo PSP pagador para recuperar o payload JSON que representa uma recorrência. |
| **PayloadLocationRec** | Gestão de locations relacionadas aos payloads de recorrência. |
| **SolicRec** | Gestão de solicitações de recorrências. |
| **CobR** | Gestão de cobranças recorrentes. |
| **WebhookRec** | Notificações das ações relacionadas às recorrências. |
| **WebhookCobR** | Notificações das ações relacionadas às cobranças recorrentes. |

# ANEXO V – Mapeamento para mensagens ISO 20022

## 1. Introdução

As mensagens utilizadas no Pix seguem o padrão ISO 20022.

A tabela abaixo contém as mensagens relacionadas com as várias formas de iniciação de Pix:

| Mensagem | Finalidade | Contexto |
|---|---|---|
| PACS.008 | Liquidação | Todos |
| PAIN.009 | Solicitação de confirmação da recorrência. | Pix Automático |
| PAIN.012 | Na jornada 1 de autorização: i. resposta ao recebimento da solicitação de confirmação da recorrência (pain.009); ii. comunicação da permissão ou da rejeição do usuário pagador para os pagamentos recorrentes.<br>Em todas as jornadas de autorização: i. comunicação da permissão do usuário pagador para os pagamentos recorrentes e, ii. confirmação do recebimento da pain.012 anterior.<br>Também utilizada como resposta a uma solicitação de cancelamento da permissão para pagamentos recorrentes ou como resposta a uma solicitação de cancelamento de uma pain.009 (pain.011). | Pix Automático |
| PAIN.011 | Cancelamento de permissão de pagamento recorrente ou de solicitação de confirmação da recorrência (pain.009). | Pix Automático |
| PAIN.013 | Instrução de pagamento referente a cobrança recorrente. | Pix Automático |
| PAIN.014 | Resposta à instrução de pagamento referente a cobrança recorrente. | Pix Automático |
| CAMT.055 | Cancelamento de cobrança recorrente. | Pix Automático |
| CAMT.029 | Resposta ao cancelamento de cobrança recorrente. | Pix Automático |

# 2. Mapeamento de QR Codes para mensagem pacs.008

Alguns campos de iniciação de Pix apresentados pelo PSP do recebedor ou pela STN 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. Outros exemplos: os *Purp.Cd* e *RmtInf.Strd.RfrdDocInf.AdjstmntAmtAndRsn* variam de acordo com a natureza do Pix realizado:

- cobrança ou transferência sem saque/troco;
- saque com QR Estático (id 03: *fss* presente com um ISPB);
- saque com QR Dinâmico (estrutura **`valor.retirada.saque`** presente);
- troco com QR Dinâmico (estrutura **`valor.retirada.troco`** presente)

O catálogo de mensagens[106] tem exemplos de como deve ser feito o preenchimento em cada um dos casos (sem saque/troco, com saque e com troco).

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

Os dados a serem mapeados pelo PSP do pagador do QR Code estático para a mensagem de pagamento são:

Campos EMV

| ID | Nome no Arranjo<br>*Nome EMV* | ISO 20022 pacs.008 |
|---|---|---|
| 54 | *Transaction Amount*[107] | Se `Purp.Cd for "IPAY"` então<br>**InterbankSettlementAmount <IntrBkSttlmAmt> = transaction amount**<br>**Se Purp.Cd for "OTHR" então**<br>**RemittanceInformation <RmtInf>**<br>**Structured <Strd>**<br>**ReferredDocumentAmount <RfrdDocAmt>**<br>**AdjustmentAmountAndReason <AdjstmntAmtAndRsn>**<br>**Amount <Amt> = transaction amount**<br>**Reason <Rsn> = "VLDN"** |

**IntrBkSttlmAmt** – O valor da transação ou do saque, quando informado em um QR code estático.

Quando o QR Code estático tiver o propósito de Pix Saque:

**RmtInf.Strd.RfrdDocAmt.AdjstmntAmtAndRsn.Amt** – Deve ser utilizado para informar o valor do saque.
**RmtInf.Strd.RfrdDocAmt.AdjstmntAmtAndRsn.Rsn** – Deve sempre ser preenchido com o valor `VLDN`.
**Purp.Cd** – Deve sempre ser preenchido com o valor `OTHR`.
Campos EMV Especiais

---

[106] Comunicação eletrônica de dados no sistema financeiro (bcb.gov.br)
[107] De preenchimento não obrigatório, mas quando preenchido, deve ser utilizado como valor da transação.

| ID | Nome no Arranjo<br>*Nome EMV* | ISO 20022 pacs.008 | | |
|---|---|---|---|---|
| 26-51 | *Merchant Account Information* | **ID** | **Nome no Arranjo**<br>***Nome EMV*** | **pacs.008** |
| | | 01 | chave | `CreditorAccount <CdtrAcct>`<br>`Proxy <Prxy>`<br>`Identification <Id>` |
| | | 03 | fss | `RemittanceInformation <RmtInf>`<br>`Structured <Strd>`<br>`ReferredDocumentInformation <RfrdDocInf>`<br>`Type <Tp>`<br>`Issuer <Issr>` |
| 62 | *Additional Data Field* | **ID** | **Nome no Arranjo**<br>***Nome EMV*** | **pacs.008** |
| | | 05 | txid [108]<br>*Reference Label* | `PaymentIdentification <PmtId>`<br>`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.

Quando o QR Code estático tiver o propósito de Pix Saque (id 03: *fss* presente com um ISPB):

**RmtInf.Strd.RfrdDocInf.Tp.Issr** – Deve ser utilizado para indicar o ISPB do facilitador de serviço de saque (id 03: *fss*).
**RmtInf.Strd.RfrdDocInf.Tp.CdOrPrtry.Prtry** – Deve sempre ser preenchido com o valor `AGTEC`. Caso a modalidade do agente de saque seja Agente Outra Espécie de Pessoa Jurídica que tenha como atividade principal ou secundária a prestação de serviços auxiliares a serviços financeiros ou afins ou correspondente no País (valor `AGTOT`) ou Agente Facilitador de Serviço de Saque (valor `AGFSS`), deve ser utilizado o QR Code dinâmico.

## 2.2. Mapeamento do QR Code dinâmico 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:

Campos Comuns nos Produtos Pix

| Campo | Uso | campo JSON | ISO 20022 pacs.008 |
|---|---|---|---|
| Chave Pix do recebedor | M | Chave | `CreditorAccount <CdtrAcct>`<br>`Proxy <Prxy>`<br>`Identification <Id>` |
| Identificador da Transação | M | Txid | `PaymentIdentification <PmtId>`<br>`TransactionIdentification <TxId>` |
| Informações Adicionais pelo Pagador | O | *(resposta ao campo solicitacaoPagador[109])* | `RemittanceInformation <RmtInf>` |

[108] Quando em efeito (ou seja, diferente de '***').
[109] 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.

**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[110], especifica um texto descritivo a ser exibido em tela ao usuário pagador para que este possa digitar a informação correlata, em formato livre, a ser enviada ao recebedor. O campo *<RmtInf>* na pacs.008 é limitado a 140 caracteres. O conteúdo deste campo deve ser negociado com o PSP.

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

Campos de Valor para Cobrança para Pagamento Imediato, Saque e Troco

| Campo | Uso | campo JSON | ISO 20022 pacs.008 |
|---|---|---|---|
| Valor da cobrança ou da compra | M | valor.original | Se Purp.Cd for "IPAY" então<br>InterbankSettlementAmount <IntrBkSttlmAmt> = valor.original<br><br>Se Purp.Cd for "GSCB" então<br>RemittanceInformation <RmtInf><br>Structured <Strd><br>ReferredDocumentAmount <RfrdDocAmt><br>AdjustmentAmountAndReason <AdjstmntAmtAndRsn><br>Amount <Amt> = valor.original<br>Reason <Rsn> = "VLCP" |
| Valor do saque | O | valor.retirada.saque.valor | Se Purp.Cd for "OTHR" então<br>RemittanceInformation <RmtInf><br>Structured <Strd><br>ReferredDocumentAmount <RfrdDocAmt><br>AdjustmentAmountAndReason <AdjstmntAmtAndRsn><br>Amount <Amt> = valor.retirada.saque.valor<br>Reason <Rsn> = "VLDN" |
| Valor do troco | O | valor.retirada.troco.valor | Se Purp.Cd for "GSCB" então<br>RemittanceInformation <RmtInf><br>Structured <Strd><br>ReferredDocumentAmount <RfrdDocAmt><br>AdjustmentAmountAndReason <AdjstmntAmtAndRsn><br>Amount <Amt> = valor.retirada.troco.valor<br>Reason <Rsn> = "VLDN" |

**IntrBkSttlmAmt** – O valor informado no campo *`InterbankSettlementAmount`* equivale à soma dos três campos (**`valor.original`** + **`valor.retirada.saque.valor`** + **`valor.retirada.troco.valor`**).
**RmtInf.Strd.RfrdDocAmt.AdjstmntAmtAndRsn.Amt** – Deve ser utilizado para informar valor de saque/troco ou compra, a depender do motivo.
**RmtInf.Strd.RfrdDocAmt.AdjstmntAmtAndRsn.Rsn** – Deve ser utilizado para informar o motivo do detalhamento (se saque/troco, ou se compra).

[110] Disponível em: <https://github.com/bacen/pix-api>

**Purp.Cd** – Deve ser utilizado para informar se a transação se refere a um Pix Saque (valor `OTHR`), Pix Troco (valor `GSCB`), ou outra transação Pix (valor `IPAY`).

Campos Adicionais para os Produtos Pix Saque e Pix Troco

| Campo | Uso | campo JSON | ISO 20022 pacs.008 |
|---|---|---|---|
| Modalidade do agente de saque | O | valor.retirada.(saque ou troco).modalidadeAgente | `RemittanceInformation <RmtInf>`<br>`Structured <Strd>`<br>`ReferredDocumentInformation <RfrdDocInf>`<br>`Type <Tp>`<br>`CodeOrProprietary <CdOrPrtry>`<br>`Proprietary <Prtry>` |
| ISPB do facilitador de serviço de saque | O | valor.retirada.(saque ou troco). prestadorDoServicoDeSaque | `RemittanceInformation <RmtInf>`<br>`Structured <Strd>`<br>`ReferredDocumentInformation <RfrdDocInf>`<br>`Type <Tp>`<br>`Issuer <Issr>` |

Quando **Purp.Cd=**`OTHR` (Pix Saque) ou **Purp.Cd=**`GSCB` (Pix Troco), além dos aspectos relacionados aos valores acima citados, os respectivos campos de **`modalidadeAgente`** e **`prestadorDoServicoDeSaque`** serão de preenchimento obrigatório.
**RmtInf.Strd.RfrdDocInf.Tp.CdOrPrtry.Prtry** – Deve ser utilizado para indicar a modalidade do agente, com um dos valores possíveis do catálogo: Agente Estabelecimento Comercial (valor `AGTEC`), Agente Outra Espécie de Pessoa Jurídica que tenha como atividade principal ou secundária a prestação de serviços auxiliares a serviços financeiros ou afins ou correspondente no País (valor `AGTOT`) ou Agente Facilitador de Serviço de Saque (valor `AGFSS`[111]). Quando se tratar de um Pix Troco (**Purp.Cd=**`GSCB`), a modalidade do agente poderá ser `AGTEC ou AGTOT`[112].
**RmtInf.Strd.RfrdDocInf.Tp.Issr** – Deve ser utilizado para indicar o ISPB do facilitador de serviço de saque.

Campos de Valor para Cobrança para Pagamento com Vencimento

| Campo | Uso | campo JSON | ISO 20022 pacs.008 |
|---|---|---|---|
| Valor do pagamento | M | valor.final | **`InterbankSettlementAmount <IntrBkSttlmAmt>`** |

**IntrBkSttlmAmt** – O valor informado no campo *`InterbankSettlementAmount`* equivale ao valor final transacionado (**`valor.original`** + **`valor.multa`** + **`valor.juros`** – **`valor.abatimento`** – **`valor.desconto`**).

## 2.3. Mapeamento do Serviço de Iniciação de Transação de Pagamento para pacs.008

Nos casos em que participantes do Pix prestadores do serviço de iniciação se utilizam do serviço de iniciação, um conjunto de dados é transmitido desse prestador de serviço de iniciação ao PSP detentor

[111] Observação: no mapeamento do campo da API Pix (AGPSS) para o campo 'modalidadeAgente', da pacs.008, o domínio a ser considerado é AGFSS.
[112] No caso do Pix Troco, a modalidade do agente será AGTOT apenas quando se tratar de correspondente no País. Não é permitida a disponibilização do Pix Troco por Agente Outra Espécie de Pessoa Jurídica que tenha como atividade principal ou secundária a prestação de serviços auxiliares a serviços financeiros ou afins.

da conta. A tabela abaixo refere-se ao mapeamento dos dados transitados no serviço de iniciação para a ordem de pagamento que deve ser enviada pelo PSP ao SPI na pacs.008.

Campos do Serviço de Iniciação de Transação de Pagamento

| Nome do Campo | ISO 20022 pacs.008 |
|---|---|
| Tipo de Iniciação | MandateRelatedInformation <MndtRltdInf><br>Type <Tp><br>LocalInstrument <LclInstrm><br>Proprietary <Prtry> |
| Identificação do iniciador de pagamento (CNPJ) | InitiatingParty <InitgPty><br>Identification <Id><br>OrganisationIdentification <OrgId><br>Other <Othr><br>Identification <Id> |
| Identificação do recebedor (CPF/CNPJ) | Creditor <Cdtr><br>Identification <Id><br>PrivateIdentification <PrvtId><br>Other <Othr><br>Identification <Id> |
| Instituição do recebedor participante do Pix | CreditorAgent <CdtrAgt><br>FinancialInstitutionIdentification <FinInstnId><br>ClearingSystemMemberIdentification <ClrSysMmbId<br>MemberIdentification <MmbId> |
| Agência do recebedor | CreditorAccount <CdtrAcct><br>Identification <Id><br>Other <Othr><br>Issuer <Issr> |
| Conta do recebedor | CreditorAccount <CdtrAcct><br>Identification <Id><br>Other <Othr><br>Identification <Id> |
| Tipo de Conta do recebedor | CreditorAccount <CdtrAcct><br>Type <Tp><br>Code <Cd> |
| Valor | InterbankSettlementAmount <IntrBkSttlmAmt> |
| Moeda | InterbankSettlementAmount <IntrBkSttlmAmt Ccy="" > |
| Campo de descrição | RemittanceInformation <RmtInf> |
| Chave Pix | CreditorAccount <CdtrAcct><br>Proxy <Prxy> |
| Código de Conciliação | PaymentIdentification <PmtId><br>TransactionIdentification <TxId> |

# 3. Mapeamento para as mensagens do Pix Automático

Para os fluxos de autorização e de cancelamento da autorização do Pix Automático, assim como para os fluxos de agendamento e de cancelamento do agendamento de um pagamento recorrente foram mapeados um conjunto de dados que devem ser transmitidos entre os participantes dos usuários envolvidos. As subseções abaixo tratam do batimento entre os dados que devem transitar em cada fluxo, via mensageria, e sua correspondência com a API Pix.

O detalhamento dos formatos dos campos, dos domínios, bem como a estrutura completa das mensagens descritas nessa seção está contemplado nos arquivos .xlsx, que fazem parte do Catálogo de Mensagens. Exemplos para as mensagens, envolvendo situações de uso, sua aceitação ou rejeição por erros possíveis, foram elaborados e fazem parte dos anexos dos arquivos .xls disponibilizados.

## 3.1. Mapeamento dos dados de recorrência para pain.009

A mensagem pain.009 será utilizada exclusivamente na jornada 1 de autorização, quando o participante do usuário recebedor envia uma mensagem contendo os parâmetros da recorrência ao participante do usuário pagador, por meio da SolicRec, para validação das informações relativas ao usuário pagador e posterior confirmação da autorização. A mensagem deverá conter as informações necessárias para que o participante do usuário pagador seja capaz de armazenar em seus sistemas internos os dados que comporão a autorização do Pix Automático, que será submetida à confirmação do usuário pagador.

Campos da pain.009 e API Pix:

| Campo | Uso | Campo da Rec | Campo da SolicRec | ISO 20022 pain.009 |
|---|---|---|---|---|
| Id da Recorrência | M | idRec | idRec | Mandate <Mndt><br>MandateIdentification <MndtId> |
| Id da Solicitação Recorrência | M | | idSolicRec | Mandate <Mndt><br>MandateRequestIdentification <MndtReqId> |
| Tipo de frequência | M | calendario.periodicidade | | Occurrences <Ocrncs><br>Frequency <Frqcy><br>Type <Tp> |
| Data inicial da recorrência | M | calendario.dataInicial | | Occurrences <Ocrncs><br>FirstCollectionDate <FrstColltnDt> |
| Data final da recorrência | O | calendario.dataFinal | | Occurrences <Ocrncs><br>FinalCollectionDate <FnlColltnDt> |
| Valor | O | valor.valorRec | | CollectionAmount <ColltnAmt> |
| Piso do valor máximo | O | valor.valorMinimoRecebedor | rec.valor.valorMinimoRecebedor | Adjustment <Adjstmnt><br>Amount <Amt> |
| Nome do recebedor | M | recebedor.nome | | Creditor <Cdtr><br>Name <Nm> |
| Identificação do recebedor (CPF/CNPJ) | M | recebedor.cnpj | | Creditor <Cdtr><br>Identification <Id><br>PrivateIdentification <PrvtId><br>Other <Othr><br>Identification <Id> |
| Identificação do usuário pagador (CPF/CNPJ) | M | | destinatario.cpf ou destinatario.cnpj | Debtor <Dbtr><br>Identification <Id><br>PrivateIdentification <PrvtId><br>Other <Othr><br>Identification <Id> |
| Conta do pagador | M | | destinatario.conta | DebtorAccount <DbtrAcct><br>Identification <Id><br>Other <Othr><br>Identification <Id> |
| Agência do pagador | O | | destinatario.agencia | DebtorAccount <DbtrAcct><br>Identification <Id><br>Other <Othr><br>Issuer <Issr> |
| Instituição participante do pagador | M | | destinatario.ispbParticipante | DebtorAgent <DbtrAgt><br>FinancialInstitutionIdentification <FinInstnId><br>ClearingSystemMemberIdentification <ClrSysMmbId><br>MemberIdentification <MmbId> |
| Identificação do devedor (nome) | O* | vinculo.devedor.nome | | UltimateDebtor <UltmtDbt><br>Name <Nm> |
| Identificação do devedor (CPF/CNPJ) | O* | vinculo.devedor.cpf ou vinculo.devedor.cnpj | | UltimateDebtor <UltmtDbtr><br>Identification <Id><br>PrivateIdentification <PrvtId><br>Other <Othr><br>Identification <Id> |
| Número do contrato | M | vinculo.contrato | | ReferredDocument <RfrdDoc><br>Number <Nb> |

| | | | | |
|---|---|---|---|---|
| Descrição do objeto/contrato | O | vinculo.objeto | | ReferredDocument <RfrdDoc><br>CreditorReference <CdtrRef> |
| Criação da Recorrência** | M | atualizacao.status | | MandateProcessingDetails <MndtPrcgDtls><br>MandateProcessingType <MndtPrcgTp> |
| Data/hora da criação da recorrência | M | atualizacao.data | | MandateProcessingDetails <MndtPrcgDtls><br>ProcessingDateTime <PrcgDtTm> |
| Criação da solicitação da recorrência*** | M | | atualizaçao.status | MandateProcessingDetails <MndtPrcgDtls><br>MandateProcessingType <MndtPrcgTp> |
| Data/hora da criação da solicitação da recorrência | M | | atualizaçao.data | MandateProcessingDetails <MndtPrcgDtls><br>ProcessingDateTime <PrcgDtTm> |
| Data/hora da expiração da solicitação da recorrência | M | | calenario.dataExpiracaoSolicitacao | MandateProcessingDetails <MndtPrcgDtls><br>ProcessingDateTime <PrcgDtTm> |

* Campo opcional na mensageria apenas se o devedor for o próprio destinatário da solicitação de confirmação de recorrência.
**A ser preenchido com o domínio 'CRTN' associado ao campo de data-hora da criação da recorrência.
*** A ser preenchido com o domínio 'CRAT' associado ao campo de data-hora da criação da solicitação de confirmação da recorrência.

## 3.2 . Mapeamento dos dados de recorrência para pain.012

A mensagem pain.012 será utilizada em todos os fluxos de autorização e no cancelamento da autorização do Pix Automático, tanto nos casos de sucesso quanto nos casos de rejeição. Será utilizada também nos casos de cancelamento de uma solicitação de confirmação da recorrência (pain.009), em resposta à pain.011. Para as jornadas de autorização, sejam aquelas iniciadas pelo recebedor, sejam aquelas iniciadas pela leitura do QR Code contendo os parâmetros de recorrência, a mensagem deverá disponibilizar um conjunto de dados necessários para viabilizar o pagamento periódico, se autorizado, e para confirmar um recebimento de uma pain.012 anterior, para sincronização de informações entre os sistemas dos participantes.

A estrutura proposta para a mensagem é construída, basicamente, a partir dos mesmos campos da mensagem pain.009 correlata, o que permite seu uso em todos os cenários. A estrutura traz as informações:

- campo status da mensagem, a ser preenchido com 'true' ou 'false'. Se for 'false', poderá indicar a rejeição da recorrência pelo usuário pagador (NotRecognizedByDebtor ou RejectedByDebtor), refletindo no status da recorrência;
- campo motivo de rejeição, a ser preenchido com uma das opções contidas na tabela de domínios da mensagem, somente no caso do campo status ser igual a 'false', indica que ocorreu um erro na etapa imediatamente anterior do fluxo de autorização ou de cancelamento da autorização;
- o código de município do usuário pagador, necessário para se determinar o próximo dia útil, caso a data de vencimento da cobrança seja um dia não útil e o usuário recebedor tenha optado por esse ajuste, e para os cálculos de eventuais multas e juros por atraso em pagamentos efetivados após a data de vencimento;
- as informações de conta e agência do usuário pagador, que trafegarão na mensagem, mas que ficarão restritas aos controles do PSP recebedor (não deverão ser disponibilizadas ao usuário recebedor via API Pix ou Arquivo Padronizado);
- o campo suplementar de autorização da recorrência, a ser preenchido com AUT1, AUT2, AUT3 ou AUT4, se a autorização para os pagamentos periódicos for aceita pelo usuário pagador, indicando qual a jornada utilizada, dentre as disponíveis (1, 2, 3 ou 4);

- o campo suplementar status da recorrência, que poderá indicar que a recorrência está pendente de confirmação, confirmada pelo usuário pagador ou cancelada. Caso o status da mensagem seja 'true', o status da recorrência poderá assumir os valores:
  - 'PDNG': quando a pain.012 for resposta à pain.009, de solicitação de confirmação da recorrência (exclusivo à jornada 1);
  - 'CFDB: quando a pain.012 comunicar a permissão concedia pelo usuário pagador para os pagamentos recorrentes;
  - CCLD: quando a pain.012 for resposta à pain.011, informando o cancelamento de uma recorrência ou de uma solicitação de confirmação pendente de autorização.

Campos da pain.012 e API Pix:

| Campo | Uso | Campo da Rec | Campo da SolicRec***** | ISO 20022 pain.012 |
|---|---|---|---|---|
| Id da Recorrência | M | idRec | idRec | OriginalMandate <OrgnlMndt><br>Mandate <Mndt><br>MandateIdentification <MndtId> |
| Tipo de frequência | M | calendario.periodicidade | | Occurrences <Ocrncs><br>Frequency <Frqcy><br>Type <Tp> |
| Data inicial da recorrência | M | calendario.dataInicial | | Occurrences <Ocrncs><br>FirstCollectionDate <FrstColltnDt> |
| Data final da recorrência | O | calendario.dataFinal | | Occurrences <Ocrncs><br>FinalCollectionDate <FnlColltnDt> |
| Valor | O | valor.valorRec | | CollectionAmount <ColltnAmt> |
| Nome do recebedor | M | recebedor.nome | | Creditor <Cdtr><br>Name <Nm> |
| Identificação do recebedor (CPF/CNPJ) | M | recebedor.cnpj | | Creditor <Cdtr><br>Identification <Id><br>PrivateIdentification <PrvtId><br>Other <Othr><br>Identification <Id> |
| Código município IBGE | O | pagador.CodMun | | Debtor <Dbtr><br>PostalAddress <PstlAdr><br>TownName <TwnNm> |
| Identificação do usuário pagador (CPF/CNPJ) | M | pagador.cpf ou pagador.cnpj | destinatario.cpf ou destinatario.cnpj | Debtor <Dbtr><br>Identification <Id><br>PrivateIdentification <PrvtId><br>Other <Othr><br>Identification <Id> |
| Conta do pagador | M | | destinatario.conta | DebtorAccount <DbtrAcct><br>Identification <Id><br>Other <Othr><br>Identification <Id> |
| Agência do pagador | O | | destinatario.agencia | DebtorAccount <DbtrAcct><br>Identification <Id><br>Other <Othr><br>Issuer <Issr> |
| Instituição participante do pagador | M | pagador.ispbParticipante | destinatario.ispbParticipante | DebtorAgent <DbtrAgt><br>FinancialInstitutionIdentification <FinInstnId><br>ClearingSystemMemberIdentification <ClrSysMmbId><br>MemberIdentification <MmbId> |
| Identificação do devedor (nome) | O | vinculo.devedor.nome | | UltimateDebtor <UltmtDbt><br>Identification <Id> |
| Identificação do devedor (CPF/CNPJ) | O | vinculo.devedor.cpf ou vinculo.devedor.cnpj | | UltimateDebtor <UltmtDbtr><br>Identification <Id><br>PrivateIdentification <PrvtId><br>Other <Othr><br>Identification <Id> |

| | | | | |
|---|---|---|---|---|
| Número do contrato | M | vinculo.contrato | | ReferredDocument <RfrdDoc><br>Number <Nb> |
| Descrição do objeto/contrato | O | vinculo.objeto | | ReferredDocument <RfrdDoc><br>CreditorReference <CdtrRef> |
| Criação da Recorrência* | O | atualizacao.status | | MandateProcessingDetails <MndtPrcgDtls><br>MandateProcessingType <MndtPrcgTp> |
| Data/hora da criação da recorrência | O | atualizacao.data | | ProcessingDateTime <PrcgDtTm> |
| Atualização da recorrência** | O | atualizacao.status | | MandateProcessingDetails <MndtPrcgDtls><br>MandateProcessingType <MndtPrcgTp> |
| Data/hora da atualização da recorrência | O | atualizacao.data | | ProcessingDateTime <PrcgDtTm> |
| Autorização da recorrência*** | O | atualizacao.status | | MandateProcessingDetails <MndtPrcgDtls><br>MandateProcessingType <MndtPrcgTp> |
| Data/hora da autorização da recorrência | O | atualizacao.data | | ProcessingDateTime <PrcgDtTm> |
| Status da recorrência **** | O | status | | MandateStatus <MndtSts> |

*A ser preenchido com o domínio 'CRTN' associado ao campo de data-hora da criação da recorrência.
** A ser preenchido com o domínio 'UPDT', associado ao campo de data-hora da última atualização no status da recorrência.
*** A ser preenchido com o domínio 'AUTn' associado ao campo de data-hora em que o usuário pagador concedeu a permissão para os pagamentos periódicos, onde 'n' indica a jornada utilizada (n= 1, 2, 3 ou 4).
****Quando o status da recorrência assumir os valores 'PNDG' ou 'CCLD', os domínios "CRTN" e "UPDT" devem ser preenchidos, com as data-horas correspondentes. Quando o status for 'CFDB', os domínios "CRTN", "UPDT" e "AUTn" devem ser preenchidos, com as data-horas correspondentes.
***** As informações contidas em "Campo da SolicRec" só serão usadas para a pain.012 que responde uma pain.011 de cancelamento de uma SolicRec (pain.009).

## 3.3. Mapeamento dos dados do cancelamento da recorrência ou da solicitação de confirmação da recorrência (pain.009) para pain.011

Tanto o usuário e o PSP pagador quanto o usuário e o PSP recebedor poderão solicitar, unilateralmente, o cancelamento de uma recorrência do Pix Automático. Os participantes envolvidos devem acatar as solicitações de seus usuários e transmitir as informações de cancelamento ao outro participante, para sincronização das informações entre as partes pagadora e recebedora. A mensagem pain.011 contém todos os campos necessários para que os participantes consigam identificar qual recorrência está sendo cancelada, o motivo do cancelamento, independente de quem fez a solicitação. Sua estrutura é similar à da mensagem pain.012, diferenciando-se pelos seguintes acréscimos:
- Identificação do solicitante do cancelamento, que permitirá a identificação do proponente do cancelamento, seja o usuário pagador, o usuário recebedor ou os respectivos participantes.
- Motivo do cancelamento, que deve identificar a razão do cancelamento, dentro de uma tabela de domínios proprietários;
- Campo para registro da data/hora do cancelamento da recorrência.
A pain.011 também pode ser utilizada na situação em que se deseja cancelar uma solicitação de confirmação de recorrência ainda pendente de resposta do pagador. Isso ocorre quando há algum erro em alguma das informações contidas na recorrência (já que não podem ser alteradas), ou quando a recorrência objeto daquela solicitação de confirmação for confirmada por meio de outra jornada de autorização (QR Code). Nesse caso, como se trata de cancelamento da pain.009, que é enviada pelo PSP recebedor, esse fluxo será iniciado somente pelo PSP recebedor.

Campos da pain.011 e API Pix:

| Campo | Uso | campo da Rec | campo da SolicRec | ISO 20022 pain.011 |
|---|---|---|---|---|
| Motivo do cancelamento | M | encerramento.cancelamento.codigo | | CancellationReason <CxlRsn><br>Reason <Rsn><br>Proprietary <Prtry> |
| Id da Recorrência | M | idRec | idRec | OriginalMandate <OrgnlMndt><br>Mandate <Mndt><br>MandateIdentification <MndtId> |
| Tipo de frequência | M | calendario.periodicidade | | Occurrences <Ocrncs><br>Frequency <Frqcy><br>Type <Tp> |
| Data inicial da recorrência | M | calendario.dataInicial | | Occurrences <Ocrncs><br>FirstCollectionDate <FrstColltnDt> |
| Data final da recorrência | O | calendario.dataFinal | | Occurrences <Ocrncs><br>FinalCollectionDate <FnlColltnDt> |
| Valor | O | valor.valorRec | | CollectionAmount <ColltnAmt> |
| Nome do recebedor | M | recebedor.nome | | Creditor <Cdtr><br>Name <Nm> |
| Identificação do recebedor (CPF/CNPJ) | M | recebedor.cnpj | | Creditor <Cdtr><br>Identification <Id><br>PrivateIdentification <PrvtId><br>Other <Othr><br>Identification <Id> |
| Identificação do usuário pagador (CPF/CNPJ) | M | pagador.cpf ou pagador.cnpj | destinatario.cpf ou destinatario.cnpj | Debtor <Dbtr><br>Identification <Id><br>PrivateIdentification <PrvtId><br>Other <Othr><br>Identification <Id> |
| Conta do pagador | M | | destinatario.conta | DebtorAccount <DbtrAcct><br>Identification <Id><br>Other <Othr><br>Identification <Id> |
| Agência do pagador | O | | destinatario.agencia | DebtorAccount <DbtrAcct><br>Identification <Id><br>Other <Othr><br>Issuer <Issr> |
| Instituição participante do pagador | M | pagador.ispbParticipante | destinatario.ispbParticipante | DebtorAgent <DbtrAgt><br>FinancialInstitutionIdentification <FinInstnId><br><br>ClearingSystemMemberIdentification <ClrSysMmbId><br>MemberIdentification <MmbId> |
| Identificação do devedor (nome) | O | vinculo.devedor.nome | | UltimateDebtor <UltmtDbt><br>Name <Nm> |
| Identificação do devedor (CPF/CNPJ) | O | vinculo.devedor.cpf ou vinculo.devedor.cnpj | | UltimateDebtor <UltmtDbtr><br>Identification <Id><br>PrivateIdentification <PrvtId><br>Other <Othr><br>Identification <Id> |
| Número do contrato | M | vinculo.contrato | | ReferredDocument <RfrdDoc><br>Number <Nb> |
| Descrição do objeto/contrato | O | vinculo.objeto | | ReferredDocument <RfrdDoc><br>CreditorReference <CdtrRef> |
| Criação da Recorrência * | M | atualizacao.status | | MandateProcessingDetails <MndtPrcgDtls><br>MandateProcessingType <MndtPrcgTp> |
| Data/hora da criação da recorrência | M | atualizacao.data | | MandateProcessingDetails <MndtPrcgDtls><br>ProcessingDateTime <PrcgDtTm> |
| Cancelamento da recorrência ** | M | atualizacao.status | atualizacao.status | MandateProcessingDetails <MndtPrcgDtls><br>MandateProcessingType <MndtPrcgTp> |

| | | | | |
|---|---|---|---|---|
| Data-hora do cancelamento da recorrência | M | atualizacao.data | atualizacao.data | MandateProcessingDetails <MndtPrcgDtls> ProcessingDateTime <PrcgDtTm> |

* A ser preenchido com o domínio 'CRTN' associado ao campo de data-hora da criação da recorrência.
** A ser preenchido com o domínio 'CLTN', associado ao campo de data-hora do cancelamento da recorrência ou da solicitação de confirmação da recorrência (pain.009).

## 3.4. Mapeamento de uma cobrança recorrente para pain.013

A mensagem pain.013 será usada no fluxo de envio de uma instrução de pagamento do Pix Automático. Depois de satisfeitas as condições para que os pagamentos periódicos possam ser realizados por meio do Pix Automático, o usuário recebedor estará apto a gerar instruções de pagamento, as quais deverão ser enviadas ao PSP pagador pelo PSP recebedor por meio da mensagem pain.013, após feitas as verificações de compatibilidade entre os parâmetros da recorrência e os dados das instruções de pagamento enviadas para agendamento. Para que a pain.013 possa ser enviada, assume-se como pressuposto a existência de uma recorrência com o status 'aprovada'.

Considerando que a recorrência pode ou não admitir retentativas após a data do vencimento, a pain.013 será enviada pelo participante do usuário recebedor nas situações:

a) para que o participante do usuário pagador agende o débito para a data prevista, sendo atribuído ao campo finalidadeDoAgendamento o valor "AGND" (domínio de agendamento inicial);
b) para que o participante do usuário pagador tente efetivar o débito em datas posteriores caso a liquidação não tenha acontecido na data prevista, por motivos relacionados ao usuário pagador (ex.: insuficiência de recursos, limite Pix Automático indisponível etc.), sendo atribuído ao campo finalidadeDoAgendamento o valor "NTAG" (domínio de novas tentativas de agendamento); ou
c) para que o participante do usuário pagador possa tentar enviar nova ordem de pagamento para liquidação, na mesma data originalmente prevista, caso tenha ocorrido falha no fluxo de liquidação após o envio da ordem de pagamento original.

O envio da pain.013 na situação "b" deve ocorrer apenas se a recorrência para os pagamentos periódicos assim o identificar, conforme regra de formação do campo idRecorrencia descrita na mensagem pain.012, que reproduz o idRecorrencia atribuído por ocasião da criação da pain.009 (jornada 1) ou do payload do QR Code contendo os dados da autorização nas demais jornadas:

> *R ou N – fixo (1 caractere). "R" caso a recorrência permita novas tentativas de agendamento de cobrança, ou "N" caso não permita novas tentativas.*

Campos da pain.013 e API Pix:

| Campo | Uso | Campo da CobR | Campo da Rec | ISO 20022 pain.013 |
|---|---|---|---|---|
| Id de conciliação do recebedor | M | txid | | PaymentInformationIdentification <PmtInfId> |
| Data hora de recebimento da cobrança pelo participante do usuário recebedor | M | calendario.dataCriacao | | RequestedExecutionDate <ReqdExctnDt> DateTime <DtTm> |
| Data de Vencimento | M | calendario.dataDeVencimento | | ExpiryDate <XpryDt> Date <Dt> |

| | | | | |
|---|---|---|---|---|
| Identificação do usuário pagador (CPF/CNPJ) | M | | pagador.cpf ou pagador.cnpj | Debtor <Dbtr><br>Identification <Id><br>PrivateIdentification <PrvtId><br>Other <Othr><br>Identification <Id> |
| Instituição participante do pagador | M | | pagador.ispbParticipante | DebtorAgent <DbtrAgt><br>FinancialInstitutionIdentification <FinInstnId><br>ClearingSystemMemberIdentification <ClrSysMmbId><br>MemberIdentification <MmbId> |
| Nome do devedor | M | | vinculo.devedor.nome | UltimateDebtor <UltmtDbt><br>Name <Nm> |
| Identificação do devedor (CPF/CNPJ) | M | | vinculo.devedor.cpf ou vinculo.devedor.cnpj | UltimateDebtor <UltmtDbtr><br>Identification <Id><br>PrivateIdentification <PrvtId><br>Other <Othr><br>Identification <Id> |
| idFimAFim | M | tentativas.endToEndId | | CreditTransferTransaction <CdtTrfTx><br>PaymentIdentification <PmtId><br>EndToEndIdentification <EndToEndId> |
| Valor | M | valor.original | | CreditTransferTransaction <CdtTrfTx><br>Amount <Amt><br>InstructedAmount <InstdAmt> |
| Id da Recorrência | M | idRec | idRec | MandateRelatedInformation <MndtRltdInf><br>MandateIdentification <MndtId> |
| Identificação do recebedor (CNPJ) | M | | recebedor.cnpj | Creditor <Cdtr><br>Identification <Id><br>PrivateIdentification <PrvtId><br>Other <Othr><br>Identification <Id> |
| Conta do recebedor | M | recebedor.conta | | CreditorAccount <CdtrAcct><br>Identification <Id><br>Other <Othr><br>Identification <Id> |
| Agência do recebedor | O | recebedor.agencia | | CreditorAccount <CdtrAcct><br>Identification <Id><br>Other <Othr><br>Issuer <Issr> |
| Tipo de conta do recebedor | M | recebedor.tipoConta | | CreditorAccount <CdtrAcct><br>Type <Tp><br>Code <Cd |
| Finalidade do agendamento | M | tentativas.tipo | | Purpose <Purp><br>Proprietary <Prtry> |
| Informações entre usuários | O | infoAdicional | | RemittanceInformation <RmtInf><br>Unstructured <Ustrd> |

Conforme previsto na situação “c” acima, a pain.013 poderá ser usada quando, devido a erro no fluxo de liquidação ou indisponibilidade do PSP recebedor, há necessidade de envio de nova instrução de pagamento relativa a uma cobrança recorrente que não foi paga.

O campo ‘finalidadeDoAgendamento’ da pain.013 irá definir se a mensagem se refere ao agendamento da primeira tentativa de pagamento da cobrança, ao agendamento de uma nova tentativa de pagamento pós vencimento ou ao reenvio da instrução de pagamento, devido a erro na liquidação do pagamento original.

As regras de negócio para a realização das novas tentativas, assim como para a sua operacionalização, estão definidas nos normativos que tratam do Pix Automático, disponíveis em https://www.bcb.gov.br/estabilidadefinanceira/participantespix?modalAberto=regulamentacao_pix.

## 3.5 . Mapeamento da resposta à solicitação de um agendamento (pain.014)

Depois de recebida a pain.013, o PSP pagador terá um prazo para verificar a compatibilidade entre os dados da cobrança periódica e a autorização concedida pelo usuário pagador e informar o sucesso ou insucesso do agendamento ao PSP recebedor, por meio do envio de uma mensagem pain.014. Nos casos de insucesso, a mensagem pain.014 deve conter as informações relativas ao motivo do erro.

O conteúdo da mensagem, que inclui o id da recorrência e o id de conciliação do recebedor (txid) da pain.013 original, permitirá ao recebedor a conciliação dos pagamentos de forma inequívoca.

Campos da pain.014 e API Pix:

| Campo | Uso | campo da CobR | Campo da Rec | ISO 20022 pain.014 |
|---|---|---|---|---|
| Id de conciliação do recebedor | M | txid | | OriginalPaymentInformationIdentification <OrgnlPmtInfId> |
| IdFimAFimOriginal | M | tentativas.endToEndId | | TransactionInformationAndStatus <TxInfAndSts> OriginalEndToEndIdentification <OrgnlEndToEndId> |
| Situação do agendamento | M | tentativas.status e/ou atualizaçao.status | | TransactionInformationAndStatus <TxInfAndSts> TransactionStatus <TxSts> |
| codigoDeErro | | encerramento.rejeicao .codigo | | TransactionInformationAndStatus <TxInfAndSts> StatusReasonInformation <StsRsnInf> Reason <Rsn> Proprietary <Prtry> |
| Data-hora da aceitação ou rejeição do agendamento pelo PSP Pagador | M | tentativas.atualizacao. dataStatus e/ou atualizacao.data | | TransactionInformationAndStatus <TxInfAndSts> DebtorDecisionDateTime <DbtrDcsnDtTm> |
| Identificação do recebedor (CNPJ) | M | | recebedor.cnpj | OriginalTransactionReference <OrgnlTxRef> Creditor <Cdtr> Identification <Id> PrivateIdentification <PrvtId> Other <Othr> Identification <Id> |

## 3.6 . Mapeamento de um cancelamento de um agendamento (camt.055)

O Pix automático prevê a possibilidade de cancelamento de uma instrução ou agendamento do pagamento antes de sua liquidação, a ser solicitado por qualquer uma das partes envolvidas, desde que respeitadas as condições negociais e os prazos regulamentares. Na mensagem camt.055 encontram-se as informações sobre o cancelamento, como a identificação da transação, o motivo – conforme opções da tabela de domínios - e a identificação do solicitante do cancelamento, dentre outras informações.

Nas situações em que ocorrer o cancelamento de uma recorrência, caberá ao PSP pagador operacionalizar o cancelamento das eventuais ordens de pagamento agendadas relativas à recorrência cancelada. O PSP pagador deverá enviar a camt.055, informando motivo do cancelamento ‘CCLD’ para informar o PSP recebedor.

Campos da camt.055 e API Pix:

| Campo | Uso | campo da CobR | campo da Rec | ISO 20022 camt.055 |
|---|---|---|---|---|
| Identificação do participante que solicita o cancelamento do agendamento | M | | ISPB do PSP recebedor (caso o solicitante seja o PSP ou usuário recebedor) | `Assigner <Assgnr>`<br>`  Agent <Agt>`<br>`  FinancialInstitutionIdentification <FinInstnId>`<br>`  ClearingSystemMemberIdentification <ClrSysMmbId>`<br>`  MemberIdentification <MmbId>` |
| Identificação do participante destinatário do cancelamento | M | | pagador.ispbParticipante (caso o solicitante seja o PSP ou usuário recebedor) | `Assignee <Assgne>`<br>`  Agent <Agt>`<br>`  FinancialInstitutionIdentification <FinInstnId>`<br>`  ClearingSystemMemberIdentification <ClrSysMmbId>`<br>`  MemberIdentification <MmbId>` |
| Id de conciliação do recebedor original | M | txid | | `OriginalPaymentInformationAndCancellation <OrgnlPmtInfAndCxl>`<br><br>`OriginalPaymentInformationIdentification <OrgnlPmtInfId>` |
| Motivo do cancelamento | M | encerramento.cancelamento.codigo | | `OriginalPaymentInformationAndCancellation <OrgnlPmtInfAndCxl>`<br>`  CancellationReasonInformation <CxlRsnInf>`<br>`  Reason <Rsn>`<br>`  Proprietary <Prtry>` |
| idFimAFimOriginal | M | tentativas.endToEndId | | `TransactionInformation <TxInf>`<br>`  OriginalEndToEndIdentification <OrgnlEndToEndId>` |
| Data-hora da solicitação do cancelamento pelo PSP solicitante | M | atualizacao.data e/ou tentativas.atualizacao.data | | `<SplmtryData>`<br>`  <Envlp>`<br>`  <CxlPrcgDtls>`<br>`  ProcessingDateTime <PrcgDtTm>` |

## 3.7 Mapeamento da resposta a um cancelamento de um agendamento (camt.029)

O PSP que recebeu a mensagem camt.055 confirma seu recebimento enviando ao outro PSP, por intermédio do SPI, uma mensagem camt.029. Nos fluxos de cancelamento de um agendamento de pagamento iniciados pela parte pagadora, a camt.029 atuará como uma notificação de que o cancelamento, já efetivado, foi registrado pelo PSP recebedor em seus sistemas internos. Nos fluxos de cancelamento da instrução de pagamento iniciados pela parte recebedora, o PSP pagador enviará a camt.029 para confirmar que o débito referente à instrução de pagamento foi cancelado também em seus sistemas internos. Só depois de recebida essa confirmação é que o PSP recebedor irá notificar seu usuário confirmando o cancelamento da instrução de pagamento.

Campos da camt.029 e API Pix:

| Campo | Uso | campo da CobR | campo da Rec | ISO 20022 camt.029 |
|---|---|---|---|---|
| Participante que atualiza a solicitação do cancelamento | M | | ISPB do PSP do recebedor (caso o PSP recebedor esteja enviando a camt.029) | Assigner <Assgnr><br>Agent <Agt><br>FinancialInstitutionIdentification <FinInstnId><br>ClearingSystemMemberIdentification <ClrSysMmbId><br>MemberIdentification <MmbId> |
| Participante que recebe a atualização do cancelamento | M | | pagador.ispbParticipante (caso o PSP recebedor esteja enviando a camt.029) | Assignee <Assgne><br>Agent <Agt><br>FinancialInstitutionIdentification <FinInstnId><br>ClearingSystemMemberIdentification <ClrSysMmbId><br>MemberIdentification <MmbId> |
| Id de conciliação do recebedor original | M | txid | | OriginalPaymentInformationAndStatus <OrgnlPmtInfAndSts><br>OriginalPaymentInformationIdentification <OrgnlPmtInfId> |
| Status do pedido de cancelamento | M | status e/ou tentativas.status | | OriginalPaymentInformationAndStatus <OrgnlPmtInfAndSts><br>PaymentInformationCancellationStatus<PmtInfCxlSts> |
| idFimAFimOriginal | M | tentativas.endToEndId | | TransactionInformationAndStatus <TxInfAndSts><br>OriginalEndToEndIdentification <OrgnlEndToEndId> |
| Data-hora da aceitação ou rejeição do cancelamento pelo participante | M | atualizacao.data e/ou tentativas.atualizacao.data | | <SplmtryData><br><Envlp><br>CancellationProcessingDetails <CxlPrcgDtls><br>ProcessingDateTime <PrcgDtTm> |

# ANEXO VI – Arquivo padronizado do Pix Automático

A geração e o processamento de lotes no âmbito do Pix Automático, quando não for realizada via API Pix, poderá ser realizada por meio da troca eletrônica de arquivos entre os participantes e seus usuários recebedores. O Arquivo Padronizado é uma alternativa ao uso da API Pix para viabilizar a implementação do Pix Automático por parte de empresas que utilizam sistemas legados. Ele foi desenvolvido com base no formato CNAB 750, modelo amplamente utilizado no mercado financeiro para troca de informações padronizadas, e foi adaptado para incluir as informações necessárias ao processamento do Pix Automático.

O Arquivo Padronizado segue uma lógica estruturada por posições fixas. Cada arquivo possui exatamente 750 posições por linha, onde campos específicos ocupam faixas pré-determinadas, garantindo uniformidade e fácil processamento por sistemas automatizados. Essa solução é ideal para empresas que já possuem infraestrutura integrada a arquivos no padrão CNAB e preferem continuar utilizando fluxos de troca de arquivos em lote para automatizar operações relacionadas ao Pix Automático.

O leiaute completo e as instruções para utilização do Arquivo Padronizado podem ser encontrados nos links:

- Orientações sobre o Arquivo Padronizado para o Pix Automático
- Leiaute do arquivo de remessa
- Leiaute do arquivo de retorno

# ANEXO VII – Prazos para implementação das funcionalidades

O presente anexo tem por objetivo apresentar quadro informativo com os prazos para implementação das funcionalidades de oferta obrigatória no âmbito da API Pix e do Arquivo Padronizado do Pix Automático.

Data da atualização: 19/08/2026

| Prazo para implementação | Funcionalidade |
|---|---|
| 16/06/2025 | Pix Automático |
| | |

# Histórico de revisão

| Data | Versão | Descrição das alterações |
|---|---|---|
| 11/8/2020 | 1.0 | Versão inicial |
| 23/9/2020 | 1.1 | • Ajustes nas definições nos campos do *payload*, especialmente na semântica do campo *txid*.<br>• Incluídos os Anexos I e II, tratando dos Conceitos de Negócio da API Pix e das suas especificações técnicas. |
| 13/10/2020 | 2.0 | • Seção 1.5.2: incluídas explicações sobre caracteres permitidos no campo *txid*.<br>• Seção 1.6.1: incluído texto para deixar claro que o QR Code dinâmico pode ser gerado por meio de aplicativo.<br>• Seção 1.6.6: excluído texto para deixar claro que o QR Code dinâmico não precisa ser necessariamente gerado por meio da API Pix.<br>• Seção 1.6.8: ajuste na explicação do campo "calendario.vencimento" e exclusão do campo "calendario.recebivelAposVencimento".<br>• Seção 1.6.17: inclusão de nota de rodapé no campo 62 "Additional Data Field".<br>• Seção 1.6.18: incluído texto para deixar claro que o QR Code dinâmico pode ser gerado por meio de aplicativo.<br>• Incluída a seção 1.7, que trata sobre a funcionalidade "Pix Copia e Cola".<br>• Seção 6.3.3 do Anexo I: correção da função associada à alteração da cobrança via API Pix. |
| 06/11/2020 | 2.1 | • 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;<br>• 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.<br>• Inclusão do Anexo III tratando sobre criação, atualização e cálculo de cobranças para pagamentos com vencimento. |
| 09/12/2020 | 2.2 | • Seção 1.6.6.2: inserção na tabela que exibe a estrutura do payload JSON para cobranças com vencimento da informação calendario.**validadeAposVencimento.**<br>• Anexo II – Seção 3.1: inserção de recomendação relativa ao uso de certificados nos webhooks. |
| 12/02/2021 | 2.2.1 | • Seção 1.6.6.1: adicionada explicação para a regra de incrementos do campo.<br>• Seção 1.6.6.2: adicionados esclarecimentos sobre o funcionamento do campo **calendário.validadeAposVencimento** |

| | | |
|---|---|---|
| | | • 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.<br>• Seção 1.6.1: Excluído o trecho que erroneamente se referia a um possível valor “0” no campo de valor EMV. |
| 22/03/2021 | 2.3.0 | • 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.<br>• Seção 1.6.3: removido o fragmento que indica a versão do *location*.<br>• 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.<br>• Seção 1.6.6.1: adicionado o campo modalidadeAlteracao no objeto “valor” para Pix Cobrança para pagamentos imediatos.<br>• 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.<br>• Seção 1.6.7: exemplo revisado: valor (EMV, opcional) retirado; *Ref.Label* (txid) modificado (enfatiza que vale o *payload* da cobrança); fragmento ‘v2’ (tornado opcional) retirado da *location* (url da cobrança).<br>• 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 (validadeAposVencimento; desconto; juros e multa). |
| 22/07/2021 | 2.4.0 | • Seção 1: Generalização de 'celular' para 'dispositivo móvel';<br>• Seção 2.4.2: Reforçando obrigatoriedade da chave;<br>• Seção 1.6.2. Nota de rodapé promovida para evidenciar reuso de QR Codes;<br>• Seção 1.6.3. Correção de fdqnPspRecebdor para fqdnPspRecebedor;<br>• Seção 2.7.1. Estruturando pontos de atenção;<br>• Seção 2.7.1.1: na descrição do campo valor, texto alterado para refletir que o campo segue a regex especificada na API Pix: \d{1,10}\.\d{2};<br>• Seção 2.7.1.2: na descrição do campo valor, texto alterado para refletir que o campo segue a regex especificada na API Pix: \d{1,10\|\.\d{2}; corrigida a obrigatoriedade dos campos logradouro, cidade, uf e cep, pertencentes ao objeto `recebedor`. Estes campos estavam constando erroneamente como opcionais.<br>• Seção 2.7.1.2: Remoção da obrigatoriedade do calendario.validadeAposVencimento;<br>• Seção 1.8: Inclusão de campos para o serviço de iniciação de transação de pagamento |

| | | |
|---|---|---|
| | | • Seção 0: Informações gerais sobre como mapear os campos do serviço de iniciação de transação de pagamento<br>• Anexo III: Seção 2.1: Obrigatoriedade do campo calendario.validadeAposVencimento removida.<br>• Anexo III: Seção 2.3.3.2: Definição da precisão a ser utilizada no cálculo do fator de juros |
| 26/08/2021 | 2.5.0 | • Seção 1: Esclarecimentos no texto<br>• Seção 1.2 e 1.5: Inclusão do novo campo de FSS relativos ao Pix Saque no QR Estático<br>• Seção 1.5.1: Inclusão de situação em que o campo *fss* é utilizado<br>• Seção 1.6: Inclusão dos campos relativos ao Pix Saque e Pix Troco no QR Dinâmico<br>• Seção 1.6.6.2: Adequação da descrição e da obrigatoriedade do campo calendario.validadeAposVencimento à especificação da API Pix<br>• Seção 1.8: Alteração no quadro com as informações obrigatórias sobre iniciação através do serviço de iniciação de transação de pagamento<br>• Seção 2: Inclusão dos campos nas mensagens de pagamento relativos ao Pix Saque e Pix Troco<br>• Anexo I: Seção 1: Inclusão das funcionalidades relacionadas ao Pix Saque e ao Pix Troco dentre as contempladas pela API Pix<br>• Anexo I: Seção 4: Inclusão da definição de FSS<br>• Anexo I: Seção 5.4.2: Inclusão das funcionalidades obrigatórias por produto ofertado<br>• Anexo III: Seção 2.1: Adequação da obrigatoriedade do campo calendario.validadeAposVencimento à especificação da API Pix<br>• Anexo 4: Inclusão do cronograma de implementação das funcionalidades obrigatórias |
| 17/09/2021 | 2.6.0 | • Seção 2: Adequação das informações contidas nos campos das mensagens de pagamento e inclusão de orientações referentes a um Pix Saque via QR Code estático |
| 29/10/2021 | 2.6.1 | • Seção 1.6.1: Alteração de texto da nota de rodapé 35 sobre o código do município<br>• Seção 1.6.6.1: Correção de AGTET para AGTEC<br>• Seção 1.6.6.1: Adequação sobre o conteúdo do campo valor.retirada.troco. modalidadeAgente para o Pix Troco<br>• Seção 1.6.6.2: Reforço da obrigatoriedade do campo calendario.validadeAposVencimento no retorno<br>• Seção 1.8: Inclusão de campos para o serviço de iniciação de transação de pagamento<br>• Seção 2.3: Inclusão de campo do serviço de iniciação de transação de pagamento para pacs.008 |

| | | |
|---|---|---|
| 09/12/2021 | 2.6.2 | • Seção 1: Adequação das terminologias relacionadas ao Pix Saque e Pix Troco<br>• Seção 1.5.4, 1.6.6.1 e 2.2: Inclusão dos correspondentes bancários como agente de Saque (modalidade AGTOT) |
| 30/08/2022 | 2.6.3 | • Seção 1.5: Alteração na denominação do campo *pss* para *fss* no QR Code estático, com semântica equivalente<br>• Seções 1 e 2: Adequação das terminologias relacionadas ao Pix Saque e Pix Troco, em relação ao Facilitador de Serviço de Saque<br>• Seção 1.8: Alteração no quadro com as informações obrigatórias sobre iniciação através do serviço de iniciação de transação de pagamento, com adequação da data de obrigatoriedade da geração do código <EndToEndId> pelo iniciador e inclusão de informação sobre o codMun do usuário pagador<br>• Anexo III. Seção 2.1: Inseridos esclarecimentos sobre os campos valor e valor do desconto, na composição do valor da cobrança<br>• Anexo III. Seção 2.3.2: Ajuste no cálculo do valor do desconto, na cobrança com vencimento, podendo ser aplicado para datas menores ou iguais à data de vencimento, conforme especificação da API Pix. |
| 31/10/2024 | 2.7.0 | Reorganização do documento:<br>• Criação da seção “1. Introdução”<br>• Seção anterior “1. Iniciação do Pix por QR Code” foi dividida em duas seções: “2. Iniciação por QR Code” e “3. Outras formas de iniciação”<br>• Seção anterior “2. Mapeamento para Mensagens ISO 20022” foi transferida para o “ANEXO V - Mapeamento para Mensagens ISO 20022”<br>• O “ANEXO IV – Prazos para implementação das funcionalidades” foi alterado para “ANEXO VI – Prazos para implementação das funcionalidades”<br><br>Inclusão do produto Pix Automático:<br>• Criação da seção “2.8 Iniciação via QR Code Composto”<br>• Criação da seção “3.3 Pix Automático”<br>• Inclusão de conteúdo relativo ao Pix Automático e ao QR Code Composto no “ANEXO I – API Pix: Conceitos de negócio”<br>• Criação do “ANEXO IV – Pix Automático”<br>• Inclusão das mensagens utilizadas no Pix Automático no “ANEXO V - Mapeamento para Mensagens ISO 20022” |
| 29/11/2024 | 2.8.0 | • Anexo IV, Seção 2.2.1: Inclusão de esclarecimento sobre os identificadores das recorrências.<br>• Inclusão do ANEXO VI sobre o Arquivo Padronizado do Pix Automático. |

| | | |
|---|---|---|
| | | • Reorganização no documento: O “ANEXO VI – Prazos para implementação das funcionalidades” foi alterado para “ANEXO VII – Prazos para implementação das funcionalidades”. |
| 17/03/2025 | 2.8.1 | • Anexo IV, Seção 2.1.2: Ajuste na redação e inclusão da referência ao convênio no atributo recebedor da recorrência.<br>• Anexo IV, Seção 2.3.9: Inclusão da referência ao atributo ajusteDiaUtil na cobrança recorrente.<br>• Anexo IV, Seção 3.3.5: Inclusão da tabela de códigos de rejeição das tentativas de agendamento de cobranças recorrentes que provocam a rejeição da cobrança recorrente correlata.<br>• Anexo IV, Seção 4.3.1: Ajuste na regra para definição da data de início dos ciclos de cobrança, quando a data esperada da cobrança não existe. |
| 05/06/25 | 2.8.2 | • Seção 2.8.4.2: Ajuste na coluna Mult. da tabela de forma a evidenciar a opcionalidade do campo vinculo.objeto dos parâmetros da recorrência.<br>• Inclusão de seção 3.4 sobre forma de iniciação “Pix por aproximação” |
| 05/09/25 | 2.9.0 | • Seção 2.5.1: inclusão de exemplo de chave CNPJ alfanumérica.<br>• Anexo IV, Seção 2.1, Seção 2.2: adaptação das regras de formação do idRec e idSolicRec ao CNPJ alfanumérico. |
| 19/08/26 | 2.10.0 | • Seção 3.2: Inclusão da forma de iniciação AUTO no quadro com as informações obrigatórias sobre iniciação através do serviço de iniciação de transação de pagamento e ajuste na informação sobre o codMun do usuário pagador no QR Code dinâmico com vencimento<br>• Anexo III, Seção 3: Ajustes textuais sobre o campo valor.juros.valorPerc<br>• Anexo IV, Seção 2.1.6: Ajuste para inclusão de nova informação de encerramento de uma recorrência<br>• Anexo IV, Seção 2.2.5: Inclusão da seção com informações de encerramento de uma solicitação de recorrência. |