

## Lista de problemas conhecidos das especificações das APIs - fase 3

Atualizada em 02/02/2022

### Orientações gerais:

- a) Utilizar o swagger como fonte primária para as especificações, isto porque o swagger é a fonte usada nos testes de conformidade.
- b) Caso o participante encontre algum das problemas nas especificações que não esteja contemplado na planilha abaixo, pedimos a gentileza que seja aberto um chamado via Service Desk (<https://servicedesk.openbankingbrasil.org.br/>).

ID	Crítico	API / Sessão	Endpoint	Campo	Como está? Qual o problema?	Como deveria ser?	Orientação até publicação do ajuste
BCLOG-F03-001	Não	Máquina de Estados Fase 3 e Guia de Experiência	Máquina de estados Fase 3 - estado REJECTED Guia de UX - Status da iniciação de pagamento Guia de UX - Status de iniciação de pagamento - Requisito #4	overdraftUsedLimitCurrency	Identificado inconsistência entre Guia de UX para o status da iniciação de pagamento e a Máquina de Estados do consentimento API Fase 3.  Na máquina de estados existe o estado REJECTED que reflete dois status diferentes no Guia de UX, (Pagamento) Cancelado - REJECTED e (Pagamento) Não efetivado - AUTHORISED (por mais de 60 min) + Payment Not Requested.	O correto é seguir os estados apresentados pela máquina de estados, unificando os dois status do Guia de UX num único estado REJECTED. As especificações do swagger estão condizentes com a máquina de estados.  Portanto, não existe o status AUTHORISED (por mais de 60 min) + Payment Not Requested presente no Guia de UX.  Além disso, o requisito #4 deve ser alterado dado que consentimentos rejeitados por regra de negócio poderiam aparecer na detentora sim, uma vez que esta é a única a saber o motivo da rejeição. É importante também definir uma recomendação de feedback para o usuário pela detentora sobre a negativa na jornada.	Seguir os estados apresentados pela máquina de estados, unificando os dois status do Guia de UX num único estado REJECTED. As especificações do swagger estão condizentes com a máquina de estados.
BCLOG-F03-002	Não	API Payments / Schema EnumRejectionReasonType	GET /payments/v1/pix/payments/{paymentId}	rejectionReason	Atualmente, não há nenhum domínio no EnumRejectionReasonType para reportar que o Pix foi rejeitado por insuficiência de saldo na conta do cliente na detentora quando validado no fluxo assíncrono.	Deveria ter uma opção no domínio para reportar essa esse motivo de rejeição.	Utilizar a opção INSUFFICIENT_FUNDS para este cenário.

BCLOG-F03-003	Não	API Payments / Schema PaymentConsent	POST /payments/v1/consents GET /payments/v1/consents	date	O campo de data de pagamento no respectivo Schema dos endpoints apontados estão com a descrição "Data do pagamento, conforme especificação RFC-3339.", o que não deixa claro qual o fuso horário a ser adotado.	Deveria estar definido explicitamente a regra para esse campo de qual o fuso horário deverá ser adotado para o campo.	Considerar o fuso horário deste campo como Horário de Brasília.
BCLOG-F03-004	Não	API Payments / Schema EnumRejectionReasonType	GET /payments/v1/pix/payments/{paymentId}	rejectionReason	Não está previsto uma opção específica no domínio de razões de rejeição para o cenário de validação de forma assíncrona da chave no DICT, quando os dados da conta do destinatário enviadas pela iniciadora não corresponderem aos dados da chave pix consultada no DICT ou chave pix inválida.	Deveria estar definido uma opção específica no domínio de Razões de Rejeição (EnumRejectionReasonType) para essa situação.	Utilizar a opção ELEMENT_CONTENT_FORMALLY_INCORRECT para a situação do problema, na validação de forma assíncrona.
BCLOG-F03-005	Não	API Payments / Schema EnumRejectionReasonType	GET /payments/v1/pix/payments/{paymentId}	rejectionReason	Não está previsto uma opção específica no domínio de razões de rejeição para o cenário de validação de forma assíncrona da chave no QR Code, quando os dados da conta do destinatário enviadas pela iniciadora não corresponderem aos dados da chave pix consultada no QR Code ou chave pix inválida	Deveria estar definido uma opção específica no domínio de Razões de Rejeição (EnumRejectionReasonType) para essa situação.	Utilizar a opção ELEMENT_CONTENT_FORMALLY_INCORRECT para a situação do problema, na validação de forma assíncrona.
BCLOG-F03-006	Não	payments	pix/payments	amount			O QR Code estático deve ser validado nos dados que o compõe. Em particular, não existindo valor financeiro, a transação assume o valor do consentimento, que será objeto de autorização posterior pelo cliente.

BCLOG-F03-007	Não	Como assinar o payload / Motor de conformidade funcional	Todos	Todos	<p>Conforme instruções na definição do campo iss do JWT, o receptor da mensagem deverá validar se o valor do campo iss coincide com o organizationId da outra parte listado no diretório. Entretanto, não é dada uma orientação de como isso mapeamos que as instituições estão realizando a validação de duas formas distintas.</p> <ol style="list-style-type: none"> <li>1. Instituição obtém o organizationId através da Cadeia Access-Token &gt; SSA &gt; Organização e em caso de divergência retorna um erro 403</li> <li>2. Instituição utiliza o iss presente no JWT para obter os certificados de assinatura do organizationId e compara o resultado da assinatura, caso o iss esteja inválido a assinatura não estará correta e será retornado um erro 400.</li> </ol>	Deveria existir instruções de como realizar a validação para evitar ambiguidades.	<p>O motor de conformidade irá aceitar as duas formas de validação, retornado um WARNING para o cenário 1 e SUCCESS para o cenário 2. Após definição de como as instituições deverão realizar a validação o motor será ajustado para aceitar apenas a forma correta.</p>
BCLOG-F03-008	Não	API Payments / Schema EnumPaymentStatusType	GET /payments/v1/pix/payments/{paymentId}	status	<p>A respeito das orientações presentes no Guia do Pix abaixo, no item Serviço de iniciação de transação de pagamento no Pix - Recomendações e OBRIGAÇÕES – Item 02 (página 75):</p> <p>“O usuário pagador deve ser notificado caso a transação seja suspeita de fraude, sempre que o PSP precise utilizar o tempo adicional para análise da transação. Durante o período em que a ordem de pagamento estiver sendo analisada, o participante prestador do serviço de iniciação de pagamento deve disponibilizar, para o usuário pagador, a opção de cancelamento da transação. Por isso, o PSP deve sempre notificar o PSI nesses casos. Mensagem obrigatória: Deve evidenciar que a transação precisa de um tempo adicional de análise para ser autorizada e dar a opção de cancelamento da transação.</p> <p>Exemplos: -Precisamos de 30 minutos adicionais para analisar sua transação. Deseja cancelar o Pix? -Sua transação precisa de aproximadamente 60 minutos para ser autorizada. Deseja cancelara transação?”</p> <p>As orientações do Guia demandam maiores esclarecimentos e a criação de um campo TEMP</p>	<p>Deve ser criado um status TEMP para transações temporizadas – comportamento desse novo status deve ser realizada em conjunto com GT PRC / Fraudes.</p>	<p>Caso a detentora temporize uma transação a mesma deve ser informada a iniciadora com o status PDNG até que a transação seja iniciada no arranjo</p> <p>No caso de rejeição:</p> <p>2.1 (Assíncrona) Caso a detentora rejeite uma transação que foi temporizada a mesma deve informar a iniciadora o status reject com o rejection reason AB11 (TIMEOUT_DEBTOR_AGENT). A iniciadora deve interpretar esse status com a mensagem descritiva: “Transação rejeitada devida análise de fraude do detentor de conta”</p> <p>2.2 (Síncrona) Caso a detentora opte por rejeitar a transação devido suas políticas de fraudes, a mesma deve informar a iniciadora o status 422 com o Code NAO_INFORMADO</p> <p>Caso a decisão seja prosseguir com a transação, o fluxo continua seguindo a máquina de estados do pagamento</p>

BCLOG-F03-009	Não	API Payments	POST/payments/v1/pix/payments	transactionIdentification	Esclarecimento sobre utilização do campo para os diferentes tipos de localInstrument.	Conforme 'orientação até publicação do ajuste' descrita.	Se localInstrument for igual a INIC, o campo transactionIdentification deve ser preenchido obrigatoriamente. Se localInstrument for igual a MANU, DICT, QRDN ou QRES, o campo transactionIdentification não deve ser preenchido. A detentora de conta deve validar se a condicionalidade do campo foi atendida pela iniciadora de pagamento.
BCLOG-F03-010	Não	API Payments	POST/payments/v1/pix/payments	422ResponseErrorCreatePixPayment	Detalhamento sobre o código de erro em caso de envio incorreto do transactionIdentification, conforme orientações contidas no problema conhecido BCLOG-F03-009.	Criação de código de erro específico para o caso citado no problema conhecido BCLOG-F03-009.	Uso do 422 com o Code NÃO_INFORMADO para o caso citado no problema conhecido BCLOG-F03-009.
BCLOG-F03-011	Não	API Payments	POST/payments/v1/pix/payments e GET/payments/v1/pix/payments/{paymentid}	transactionIdentification	Problema de incompatibilidade de tamanho de campo transactionIdentification da api payments com o regulamentado pelo arranjo pix.	Api Payments prevê 25 caracteres e o arranjo pix, de 26 a 35 caracteres - somente para QR Code dinâmico.	Considerar o campo como tamanho máximo de 35 caracteres, desconsiderando o pattern no swagger.
BCLOG-F03-012	Não	API Payments	GET/payments	rejectionReason	Não foi elaborado códigos específicos de erro no caso de validações de QR Code do arranjo Pix.	Deveria ser criado códigos de erros específicos no caso de validação do QR Code do arranjo Pix.	Deverá seguir o comportamento descrito no informe #114 onde é orientado a reutilização de código de erro no caso de erros na validação de QR code do arranjo Pix.
BCLOG-F03-013	Sim	API Payments	GET/payments/consents	status	Atualmente o campo status em caso de rejeição "REJECTED" apresenta ambiguidade tendo em vista que não é discriminado qual das partes efetuou a rejeição e sua respectiva causa.	Criação de um novo objeto no retorno da API discriminando a origem da rejeição e sua causa.	Deverá utilizar o status atualmente disponível "REJECTED" e no caso das certificações funcionais notificar o time de Sandbox deste cenário.
BCLOG-F03-014	Sim	API Payments	Todos	Todos	O Swagger não permite a inclusão de campos adicionais (atributo additionalProperties: false).	Respeitando o princípio de extensibilidade o atributo deveria ter valor true.	Consumidores considerar o atributo como valor true.