**🧾 RCA – Resposta de Erro Inconsistente (RFC 7807 / Problem Details)**

**Sistema:** Backend Spring Boot + Frontend MPA (Bootstrap + JS vanilla)  
**Endpoint envolvido:** PUT /me/senha  
**Status final:** ✅ Corrigido e padronizado

**🔍 Descrição do Problema**

Ao enviar uma requisição para alteração de senha com nova senha igual à anterior, o backend respondia com uma **string simples**, e não no formato estruturado definido pela **RFC 7807 – Problem Details for HTTP APIs**, utilizado em todo o restante da aplicação.

**🧠 Causa Raiz Identificada**

O método alterarSenha(...) no Controller capturava as exceções NegocioException e SenhaIncorretaException localmente com try/catch, e retornava diretamente a mensagem de erro como String:

java

CopiarEditar

try {

usuarioService.atualizarSenha(...);

return ResponseEntity.noContent().build();

} catch (NegocioException e) {

return ResponseEntity.badRequest().body(e.getMessage());

}

Esse catch impedia que o @ControllerAdvice (ApiExceptionHandler) tratasse a exceção conforme o padrão RFC 7807.

**🛠️ Correção Aplicada**

A estrutura try/catch foi **removida do controller**, permitindo que o ApiExceptionHandler capture as exceções normalmente:

java

CopiarEditar

@PutMapping(path = "/me/senha", produces = MediaType.APPLICATION\_JSON\_VALUE)

public ResponseEntity<Void> alterarSenha(@RequestBody @Valid UsuarioAlteraSenhaInput input,

Authentication authentication) {

String emailAutenticado = authentication.getName();

Usuario usuario = usuarioService.buscarPorEmailObrigatorio(emailAutenticado);

usuarioService.atualizarSenha(usuario, input.getNovaSenha(), input.getSenhaAtual());

return ResponseEntity.noContent().build();

}

Com isso, a exceção NegocioException (e suas filhas como SenhaIncorretaException) passa a ser processada pelo handler global:

java

CopiarEditar

@ExceptionHandler(NegocioException.class)

public ResponseEntity<?> handleNegocioException(NegocioException ex, WebRequest request) {

Problem problem = createProblemBuilder(status, type, ex.getMessage())

.userMessage(ex.getMessage())

.build();

return handleExceptionInternal(ex, problem, new HttpHeaders(), status, request);

}

**✅ Resultado**

Agora, a resposta de erro está correta e consistente com o restante da API, por exemplo:

json

CopiarEditar

{

"status": 400,

"timestamp": "2025-07-17T15:03:33.1095406-03:00",

"type": "https://fiberguardian.com.br/problema-na-requisicao",

"title": "Violação de regra de negócio",

"detail": "A nova senha não pode ser igual à anterior.",

"userMessage": "A nova senha não pode ser igual à anterior."

}

**🧩 Lições Aprendidas**

* Centralizar o tratamento de erros via @ControllerAdvice mantém coesão e evita duplicação (KISS / DRY).
* Evitar capturar exceções no controller quando já há um handler global.
* Validar as mensagens retornadas não só pelo status, mas também pela estrutura esperada (content-type, schema JSON).