

Documentação de Erros Injetados - Sistema de Acervo

Introdução

Este documento detalha os erros intencionalmente injetados e outros defeitos presentes no código-fonte de um sistema de acervo pessoal. O objetivo é servir como referência para a validação de estratégias de teste de caixa preta.

Detalhamento dos Erros Injetados e Outros Defeitos

Erro 1: Retorno Inconsistente na Autenticação

- **Localização:** UsuarioController, método autenticar.
- **Descrição:** O método autenticar deve retornar duas variáveis (um booleano indicando sucesso e o objeto Usuario). No entanto, no bloco `except DoesNotExist` (quando o usuário ou senha estão incorretos), ele retorna apenas o valor booleano `False`. A interface (menuLogin) sempre espera dois valores (autenticado, usuario = ...).
- **Comportamento Injetado (Incorreto):** `return False`.
- **Comportamento Esperado (Correto):** `return False, None`.
- **Impacto:** Causa um erro fatal (`TypeError: cannot unpack non-iterable bool object`) quando o usuário tenta fazer login com credenciais inválidas, impedindo o fluxo normal e travando essa parte da interação.

Erro 2: Falha na Exclusão do Livro

- **Localização:** LivroController, método excluirLivro (e seu uso em menuPrincipal).
- **Descrição:** Em menuPrincipal, ao buscar o livro, ele usa `l.id` para exibir e, presumivelmente, `livro_escolhido.id` para passar para `excluirLivro`. No entanto, o modelo Livro define a chave primária como `id_livro`. Se `livro_escolhido.id` não existir ou não for o valor esperado por `Livro.get_by_id(livro_id)`, a busca pelo livro a ser excluído falhará (provavelmente levantando `Livro.DoesNotExist` ou `AttributeError`), impedindo a exclusão. *Mesmo que .id funcione*, o comentário sugere que a exclusão falha por algum motivo.
- **Impacto:** A funcionalidade de excluir um livro não funciona como esperado. O usuário tenta excluir, mas o livro permanece no acervo.

Erro 3: Omissão do Primeiro Livro na Listagem

- **Localização:** menuPrincipal, opção 4 (Listar Livros).
- **Descrição:** Ao iterar sobre a lista de livros para exibição, o código usa a fatia `livros[1:]`. Esta sintaxe ignora o primeiro elemento da lista (índice 0).
- **Comportamento Injetado (Incorreto):** `for livro in livros[1:]`:
- **Comportamento Esperado (Correto):** `for livro in livros`:

- **Impacto:** O primeiro livro do acervo do usuário nunca é exibido na listagem. Se houver apenas um livro, a listagem aparecerá vazia.

Erro 4: Retorno Incorreto na Criação do Livro

- **Localização:** LivroController, método criarLivro.
- **Descrição:** Após criar a instância do livro (`Livro.create(...)`), o método retorna a classe Livro em vez da instância livro que acabou de ser criada.
- **Comportamento Injetado (Incorreto):** `return Livro`.
- **Comportamento Esperado (Correto):** `return livro`.
- **Impacto:** Em menuPrincipal, a linha `livro = LivroController.visualizarLivro(livro.id_livro)` tentará acessar `Livro.id_livro`, o que causará um `AttributeError`, pois a classe Livro não possui um atributo `id_livro`. Isso causa um crash logo após a criação (aparente) do livro.

Erro 5: Parâmetro Incorreto na Edição do Livro

- **Localização:** LivroController, método editarLivro e seu uso em menuPrincipal.
- **Descrição:** O método editarLivro espera receber uma *instância* de Livro como primeiro argumento (`livro: Livro`). No entanto, em menuPrincipal, ele é chamado com um *inteiro* (`id_livro = int(input(...))`).
- **Comportamento Injetado (Incorreto):** `livro = LivroController.editarLivro(id_livro, **dados)`.
- **Comportamento Esperado (Correto):** Seria necessário primeiro buscar o livro pelo ID e *depois* passar a instância para editarLivro, ou editarLivro deveria ser reescrito para aceitar o ID.
- **Impacto:** Quando editarLivro tentar acessar `livro.id_livro` (ou mesmo antes, dependendo da checagem de tipos), ocorrerá um `AttributeError`, pois um `int` não tem o atributo `id_livro`. Isso causa um crash durante a tentativa de edição.