



Universidade Católica de Brasília - Ceilândia
Análise e Desenvolvimento de Sistemas
Qualidade de Software e Governança
Wérklis Marques Almeida

Relatório de Análise de Código - Sistema de Pet Shop

Brasília, novembro, 2024

1. Introdução

O objetivo deste trabalho foi criar e corrigir um código C para um sistema simples de **Pet Shop**, permitindo realizar operações como cadastrar, listar e editar dados de pets. Durante o desenvolvimento do código, foram introduzidos **erros de sintaxe** de forma proposital para que fossem identificados, analisados e corrigidos.

2. Descrição do Código Original (com erros)

O código foi inicialmente elaborado com os seguintes objetivos:

1. **Cadastrar novos animais (pets)**, armazenando informações como nome, espécie e idade.
2. **Listar os animais cadastrados.**
3. **Editar as informações de um animal já cadastrado.**

O código original apresenta as seguintes falhas de sintaxe e lógica:

- **Erro 1: Uso incorreto do scanf**
Na função `cadastrarPet` e `editarPet`, o uso do `scanf` para ler a variável `idade` do pet estava correto em termos de sintaxe (`scanf("%d", &pet->idade);`), mas no momento de elaborar o código, cometi uma possível confusão, sugerindo que seria um erro. A instrução está correta e precisa do operador `&` para passar o endereço de memória da variável `idade`.
 - **Erro 2: Falta de controle de índices**
O código original não fazia um controle adequado do número de pets cadastrados, o que poderia levar a um comportamento inesperado, como tentar cadastrar mais de 5 pets. Esse controle foi corrigido na versão final através da variável `indice`.
 - **Erro 3: Verificação de entrada inválida**
Na função de edição (`editarPet`), o código não verificava corretamente se o índice do pet a ser editado estava dentro dos limites. Isso foi corrigido garantindo que o índice informado pelo usuário fosse válido antes de tentar editar.
-

3. Descrição da Versão Corrigida

A versão corrigida do código resolve os problemas de sintaxe e melhora a lógica do programa. As correções realizadas são:

- **Correção no uso do scanf:**
O código foi mantido com o uso correto do operador `&` no `scanf` para garantir que o valor da idade fosse corretamente passado para a variável `pet->idade`.

- **Controle de índices:**

Para evitar que o programa tentasse cadastrar mais de 5 pets, foi introduzido o controle da variável `indice`, que controla o número de animais cadastrados. Essa variável é incrementada sempre que um novo pet é cadastrado.

- **Validação de índices na edição de pets:**

Ao tentar editar um pet, o código agora verifica se o índice fornecido pelo usuário está dentro dos limites válidos (entre 0 e o número de pets cadastrados - 1). Isso evita o erro de tentar acessar um pet inexistente.

4. Análise dos Erros e Acertos

Erros Identificados e Corrigidos

- **Uso incorreto do `scanf`:**

Embora o `scanf` estivesse tecnicamente correto no código original, sugeri que havia um erro quando, na verdade, não havia. O operador `&` é necessário para passar o endereço de memória das variáveis e estava sendo usado corretamente.

- **Controle de quantidade de pets cadastrados:**

No código original, não havia controle de quantos pets poderiam ser cadastrados. Isso foi corrigido através da variável `indice`, que agora controla o número de pets e impede que o programa cadastre mais do que 5 animais.

- **Verificação inadequada de índices:**

Na versão original, ao editar um pet, não havia verificação rigorosa de que o índice estava dentro do intervalo válido. Isso foi corrigido ao adicionar uma verificação que garante que o índice fornecido pelo usuário não exceda o número de pets cadastrados.

Acertos e Pontos Positivos

- **Estrutura do código:**

O uso de funções como `cadaststrarPet`, `listarPet` e `editarPet` ajuda a modularizar o código, tornando-o mais organizado e fácil de entender.

- **Simplicidade do código:**

A lógica implementada é simples e eficiente para o objetivo de um sistema básico de cadastro e edição de dados de animais. Isso facilita a manutenção e a extensão do código no futuro.

- **Tratamento de erros de entrada:**

A versão corrigida agora possui verificações para garantir que o índice fornecido pelo usuário seja válido, o que evita erros durante a execução do programa.

5. Conclusão

O exercício de analisar e corrigir o código de um sistema de **Pet Shop** foi uma ótima oportunidade para aplicar conceitos de análise de código, controle de fluxo e tratamento de

erros. A versão final do código agora está mais robusta, com verificações adequadas e um controle de fluxo melhorado. Além disso, a estrutura modular do código facilita a manutenção e futura expansão.

6. Referências

- **Documentação da Linguagem C:** Para referências sobre funções como `scanf`, verificação de erros, e manipulação de strings em C.
 - **Artigos e livros sobre boas práticas em C:** Para aprofundamento sobre a estruturação de programas e controle de fluxo.
-