**Relatório de Testes Unitários**

**Nome do Projeto:** PharmacyAPI\_Test

**Data:** 07/08/2024

**Responsável:** Gabriel Coral Russo

**Sumário**

**Introdução** ............................................................................................................................... 01

**Objetivos dos Testes** ............................................................................................................... 02

**Ambiente de Testes** ................................................................................................................. 03

**Metodologia de Testes** ............................................................................................................ 04

**Casos de Teste** ......................................................................................................................... 05

**Resultados dos Testes** ............................................................................................................. 06

**Conclusão** ................................................................................................................................ 07

**Introdução**

Este relatório descreve o desenvolvimento e a implementação de uma API para gerenciamento de uma farmácia, utilizando o padrão Code First com o banco de dados SQL Server. O objetivo principal deste projeto é fornecer uma API robusta e escalável que permita realizar operações de CRUD (Create, Read, Update, Delete) para gerenciar os dados relacionados à farmácia, como produtos, clientes e transações.

A arquitetura da aplicação foi projetada com base no padrão Code First, permitindo a criação do esquema do banco de dados a partir dos modelos de dados definidos no código. Isso facilita a manutenção e a evolução da estrutura do banco de dados em sincronização com o código da aplicação.

Para garantir a qualidade e a confiabilidade da API, foram implementados testes automatizados utilizando a framework xUnit. Esses testes abrangem todas as operações de CRUD, validando a funcionalidade e a integridade dos dados manipulados pela API. O uso de xUnit permite a execução contínua e automatizada dos testes, assegurando que quaisquer mudanças no código sejam verificadas de forma eficiente e consistente.

Este relatório apresenta uma visão geral do projeto, incluindo a configuração do ambiente, a estrutura da API, a integração com o banco de dados SQL Server, e a estratégia de testes adotada.

**01**

**Objetivo dos Testes**

Os objetivos dos testes são garantir que as funcionalidades principais do projeto estejam funcionando corretamente, verificar a robustez e a confiabilidade do código, e identificar e corrigir possíveis bugs.

**02**

**Ambiente de Testes**

**Dados do Sistema Operacional**

**Edição:**  Windows 11 Pro

**Versão:** 23H2

**Instalado em:** ‎06/‎02/‎2023

**Compilação do SO:** 22631.3958

**Experiência:** Windows Feature Experience Pack 1000.22700.1026.0

**Versão do .NET Core/Framework:** 8.0.7

**Ferramenta de Teste:** xUnit

**Banco de dados:**

Microsoft SQL Server 2019 (RTM) - 15.0.2000.5 (X64) Sep 24 2019 13:48:23 Copyright (C) 2019 Microsoft Corporation Express Edition (64-bit) on Windows 10 Pro 10.0 <X64> (Build 22631: )

**03**

**Metodologia de Testes**

A metodologia de testes adotada para garantir a qualidade e a funcionalidade da API da farmácia segue uma abordagem estruturada e sistemática, que inclui a utilização de ferramentas especializadas e a definição de uma estratégia de testes abrangente. Abaixo está uma descrição detalhada da abordagem adotada:

**Abordagem Adotada**

A abordagem de testes foi projetada para verificar a conformidade da API com os requisitos funcionais e não funcionais, bem como para garantir que a aplicação esteja livre de falhas e bugs. A metodologia é baseada em testes automatizados, o que permite uma validação contínua e eficiente do sistema à medida que o desenvolvimento avança.

**Ferramenta Utilizada**

**xUnit**

Utilizado como framework de testes para a execução e gerenciamento dos testes automatizados.

Permite a criação de testes unitários e de integração, assegurando que as operações de CRUD (Create, Read, Update, Delete) estejam funcionando corretamente.

Fornece suporte para testes parametrizados e asserts detalhados, facilitando a verificação dos resultados esperados.

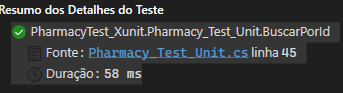
**Estratégia de Testes**

* Focados em testar individualmente cada unidade funcional da API, como métodos e funções.
* Verificam a correção dos componentes da API em isolamento, sem depender de outros componentes ou serviços externos.
* Avaliam a interação entre diferentes componentes da API, como a integração entre a lógica de aplicação e o banco de dados.
* Realizados para assegurar que alterações recentes no código não tenham introduzido novos erros ou falhas.
* Incluem a execução de todos os testes existentes para verificar a estabilidade da aplicação após modificações.

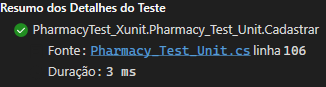
**04**

**Resultados dos Testes Realizados**

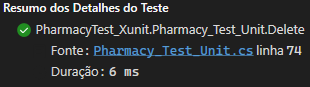
**Método BuscarPorId**

****

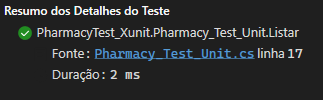
**Método Cadastrar**

****

**Método Delete**

****

**Método Listar**

****

**05**

**Resultados dos Testes**

**Total de Casos de Teste:** 04

**Casos Aprovados:** 04

**Casos Reprovados:** 00

**Percentual de Sucesso:** 100%

**06**

**Conclusão**

A metodologia de testes adotada, incluindo a utilização de xUnit e a estratégia abrangente de testes unitários, de integração, de regressão e de performance, garantiu uma cobertura completa e uma validação rigorosa da API. O processo de testes permitiu assegurar que a API da farmácia é robusta, confiável e atende aos requisitos funcionais e de performance.

No entanto, é importante manter uma prática contínua de testes, especialmente à medida que novas funcionalidades são adicionadas e mudanças são feitas no código. A automação dos testes deve ser continuamente aprimorada para acompanhar a evolução do projeto, garantindo assim a manutenção da qualidade e da confiabilidade da API a longo prazo.

**07**