

ESCOLA TÉCNICA ETEC ZONA LESTE

**Tratamento de Exceções**

São Paulo

2025

## **RESUMO**

GUEDES, Ryan; **Sistemas Elétricos Trifásicos** Quantidade de folhas 15. Pesquisa Técnica (Técnico em Mecatrônica) – Escola e Faculdade de Tecnologia SENAI “Roberto Simonsen”, São Paulo, 2025.

Esta pesquisa técnica sobre tratamento de exceções em Java permitiu reunir e analisar informações essenciais sobre os conceitos fundamentais, classificações, funcionamento interno do mecanismo de exceções, estruturas de tratamento (`try`, `catch`, `finally`, `throw` e `throws`) e boas práticas recomendadas para desenvolvimento seguro e eficiente. A partir do levantamento teórico e das referências consultadas, foi possível compreender como o tratamento adequado de exceções contribui para a confiabilidade, a estabilidade e a manutenção de programas, evitando falhas inesperadas e interrupções indesejadas na execução.

O estudo reforça que a correta implementação do tratamento de exceções — aliada ao uso apropriado das exceções verificadas (`checked`), não verificadas (`unchecked`) e personalizadas — contribui diretamente para o controle de erros, segurança da informação e robustez de sistemas críticos. A aplicação adequada dessas técnicas atende às demandas modernas de desenvolvimento, qualidade e sustentabilidade tecnológica.

**Palavras-chave:** Exceções; Java; Programação; Tratamento de Erros; Robustez de Sistemas.

# PESQUISA TÉCNICA SOBRE TRATAMENTO DE EXCEÇÕES EM JAVA

## SUMÁRIO

<b>1 INTRÓITO .....</b>	04
1.1 Finalidade .....	04
1.2 Introdução .....	04
1.3 Conceito de tratamento de exceções .....	04
1.4 Diferença entre erros e exceções .....	04
1.5 Histórico e importância no contexto da programação .....	04
<b>2 DESENVOLVIMENTO .....</b>	05
2.1 Tipos de exceções em Java .....	05
2.1.1 Exceções verificadas (checked) .....	05
2.1.2 Exceções não verificadas (unchecked) .....	05
2.1.3 Classe Throwable e hierarquia .....	05
2.2 Estrutura de tratamento em Java .....	05
2.2.1 Bloco try/catch .....	05
2.2.2 Bloco finally .....	05
2.2.3 Instrução throw .....	05
2.3 Boas práticas no tratamento de exceções .....	05
2.3.1 Capturas específicas .....	05
2.3.2 Uso adequado de logs .....	05
2.3.3 Evitar uso de exceções para controle de fluxo .....	05
2.4 Exemplos de aplicação em programas reais .....	05
2.4.1 Leitura e conversão de dados .....	05
2.4.2 Divisão entre números .....	06
2.4.3 Manipulação de arquivos .....	06
2.5 Importância na confiabilidade dos sistemas .....	06
2.5.1 Prevenção de falhas críticas .....	06
2.5.2 Melhorias na manutenibilidade .....	06
2.5.3 Segurança e consistência do software .....	06
<b>3 CONCLUSÃO .....</b>	06
<b>4 ENCERRAMENTO .....</b>	06
<b>5 REFERÊNCIAS .....</b>	06

# **PESQUISA TÉCNICA SOBRE TRATAMENTO DE EXCEÇÕES EM JAVA**

## **1. INTRÓITO**

O presente trabalho técnico tem como objetivo descrever e analisar os conceitos fundamentais relacionados ao tratamento de exceções na linguagem Java, abordando desde sua definição, finalidade e tipos, até sua aplicação prática no desenvolvimento de sistemas. São apresentados os principais blocos e estruturas de tratamento, além de boas práticas amplamente recomendadas.

### **1.1 FINALIDADE**

A finalidade desta pesquisa é oferecer uma compreensão sólida acerca do funcionamento das exceções em Java, mostrando como elas auxiliam na prevenção de erros durante a execução e na manutenção do fluxo seguro de um programa. O estudo reúne fundamentos teóricos necessários para desenvolver aplicações mais robustas, capazes de lidar com situações inesperadas de forma adequada.

### **1.2 INTRODUÇÃO**

Com o crescimento do uso de sistemas informatizados e a necessidade crescente de confiabilidade nas aplicações, torna-se essencial compreender métodos eficazes de controle de erros. Em ambientes reais, falhas podem ocorrer por diversos motivos: entrada inadequada, divisões inválidas, acessos a arquivos inexistentes, entre outros.

Java, uma das linguagens mais utilizadas no mundo, oferece uma estrutura extremamente completa para lidar com esses cenários, evitando encerramentos abruptos e protegendo a integridade do software. Assim, conhecer o tratamento de exceções é indispensável para desenvolvedores e profissionais de tecnologia.

### **1.3 CONCEITO DE TRATAMENTO DE EXCEÇÕES**

O tratamento de exceções consiste em capturar eventos inesperados que ocorrem durante a execução de um programa, evitando que ele seja interrompido de forma inadequada. Uma exceção representa uma situação anormal, como erros de cálculo, dados inválidos ou impossibilidade de acessar recursos.

### **1.4 DIFERENÇA ENTRE ERROS E EXCEÇÕES**

**Erros (Error):** problemas graves que afetam a JVM, como falta de memória, e não devem ser tratados pelo programa.

**Exceções (Exception):** representam situações previstas que podem ocorrer durante a execução e devem ser tratadas pelo desenvolvedor para garantir a continuidade da aplicação.

### **1.5 HISTÓRICO E IMPORTÂNCIA**

O tratamento estruturado de exceções ganhou importância com o avanço de linguagens orientadas a objetos e sistemas que exigem maior confiabilidade. Java adotou um modelo robusto que permitiu padronizar e melhorar a forma como falhas são tratadas.

## **2. DESENVOLVIMENTO**

### **2.1 TIPOS DE EXCEÇÕES EM JAVA**

#### **2.1.1 Exceções Verificadas (Checked)**

São verificadas durante a compilação, obrigando o programador a tratá-las. Ex.: `IOException`, `SQLException`.

#### **2.1.2 Exceções Não Verificadas (Unchecked)**

Ocorrem em tempo de execução e não exigem tratamento obrigatório. Ex.: `ArithmeticException`, `NumberFormatException`.

#### **2.1.3 Classe Throwable e Hierarquia**

Toda exceção deriva da classe `Throwable`, que se divide em:

- `Error`
- `Exception`
  - `RuntimeException`

## **2.2 ESTRUTURA DE TRATAMENTO EM JAVA**

### **2.2.1 Bloco try/catch**

O bloco `try` envolve o código que pode gerar uma exceção, enquanto o `catch` captura e trata o evento.

### **2.2.2 Bloco finally**

Usado para executar uma ação independente da ocorrência de erro, como fechar arquivos ou liberar recursos.

### **2.2.3 Instrução throw**

Permite lançar manualmente uma exceção dentro do programa para indicar problemas específicos.

## **2.3 BOAS PRÁTICAS NO TRATAMENTO DE EXCEÇÕES**

- Capturar exceções específicas sempre que possível.
- Evitar capturar exceções genéricas como `Exception`.
- Utilizar `finally` para garantir limpeza de recursos.
- Registrar erros usando logs para facilitar manutenção.
- Não usar exceções como fluxo de controle.

## **2.4 EXEMPLOS REAIS DE APLICAÇÃO**

### **2.4.1 Leitura e Conversão de Dados**

Conversões incorretas de tipos podem gerar `NumberFormatException`.

## 2.4.2 Divisão entre Números

Operações matemáticas inválidas, como divisão por zero, geram `ArithmeticException`.

## 2.4.3 Manipulação de Arquivos

Abertura de arquivos inexistentes lança `FileNotFoundException`.

## 2.5 IMPORTÂNCIA NA CONFIABILIDADE DO SISTEMA

### 2.5.1 Prevenção de falhas críticas

Exceções bem tratadas evitam encerramentos inesperados.

### 2.5.2 Melhoria da manutenção

Eros registrados ajudam a identificar e corrigir problemas.

### 2.5.3 Segurança e consistência

Evita que dados sensíveis ou processos críticos sejam comprometidos.

## 3. CONCLUSÃO

O tratamento de exceções em Java é fundamental para a criação de sistemas robustos, seguros e estáveis. A compreensão dos tipos de exceções, da finalidade do tratamento e da estrutura de blocos como `try`, `catch`, `finally` e `throw` permite que o desenvolvedor crie aplicações capazes de lidar com falhas inesperadas sem comprometer o funcionamento do programa. As boas práticas apresentadas demonstram a importância de um tratamento adequado e consistente, contribuindo significativamente para a qualidade do software.

## 4. ENCERRAMENTO

Nada mais havendo a registrar, encerra-se a presente Pesquisa Técnica, composto por 15 (quinze) páginas, elaborado com base em evidências técnicas e documentação oficial da linguagem Java, com vistas à segurança e conformidade das atividades analisadas.

**São Paulo, 25 de outubro de 2025.**

**Seu Nome Completo**

Técnico em \_\_\_\_\_

## 5. REFERÊNCIAS

ORACLE. *Java Platform, Standard Edition – Documentation*. Disponível em:  
<https://docs.oracle.com/javase/>. Acesso em: 25 out. 2025.

DEITEL, Paul; DEITEL, Harvey. *Java: Como Programar*. 10. ed. São Paulo: Pearson, 2016.

HORSTMANN, Cay S. *Core Java – Volume I: Fundamentals*. 11. ed. Prentice Hall, 2018.

SIERRA, Kathy; BATES, Bert. *Use a Cabeça! Java*. Rio de Janeiro: Alta Books, 2014.

DEV MEDIA. Tratamento de Exceções em Java. Disponível em: <https://www.devmedia.com.br>. Acesso em: 25 out. 2025.