

**Versão:** <1.0>

## Glossário

**Data:** 24/03/2025

---

### Histórico da Revisão

Data	Versão	Descrição	Autor
24/03/2025	1.0	Preenchimento inicial	Victor Cottar Marçal Silva

---

### Índice

1. Introdução
  - 1.1 Objetivo
  - 1.2 Escopo
  - 1.3 Referências
  - 1.4 Visão Geral
2. Termos Técnicos
  - 2.1 API
  - 2.2 Backend
  - 2.3 Docker
  - 2.4 Frontend
  - 2.5 Git
  - 2.6 Github
  - 2.7 Hardware
  - 2.8 NestJS
  - 2.9 NextJS
  - 2.10 Nodejs
  - 2.11 Postgres
  - 2.12 Postman
  - 2.13 React
  - 2.14 Shadcn

- 2.15 Sistema operacional
- 2.16 Tailwind
- 2.17 TypeORM
- 2.18 Visual Studio Code

### 3. Estereótipos UML

---

## Introdução

Este documento tem como objetivo apresentar e definir todos os termos e siglas que serão utilizados ao longo do desenvolvimento do projeto. Ele serve como referência para padronizar a comunicação entre os envolvidos e garantir um entendimento claro e consistente das terminologias aplicadas. Além disso, este glossário abrange informações essenciais relacionadas a especificações abordadas em outros documentos, como o documento de visão, especificação de casos de uso, especificação complementar, requisitos do investidor e plano de projeto.

---

## Objetivo

Este documento tem a finalidade de definir a terminologia específica do domínio do problema, esclarecendo os termos utilizados nas descrições de casos de uso e em outros documentos do projeto. Ele funciona como um dicionário informal de dados, registrando definições importantes para que os demais documentos possam concentrar-se na funcionalidade do sistema e no tratamento das informações.

---

## Escopo

Este glossário está inicialmente associado aos documentos: documento de visão, especificação de casos de uso, especificação complementar, requisitos do investidor e plano de projeto, que compõem o desenvolvimento do sistema Controle Gerencial Londricostura.

---

## Referências

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML, guia do usuário. 2. ed. Rio de Janeiro: Elsevier, c2006. 474 p.

---

## Visão Geral

**Este glossário contém as definições dos termos, acrônimos e abreviações referenciados e utilizados nos diversos documentos do projeto do software do sistema proposto.**

### # Termos Técnicos

#### API

Interface que permite a comunicação entre diferentes sistemas, possibilitando o consumo e fornecimento de dados de forma padronizada.

---

#### Backend

Parte do sistema responsável pela lógica de negócio, processamento de dados e comunicação com o banco de dados, geralmente executada no servidor.

---

#### Docker

Plataforma que permite criar, implantar e executar aplicativos em contêineres, facilitando a consistência entre ambientes de desenvolvimento e produção.

---

#### Frontend

Interface do usuário e camada de apresentação de um sistema, onde ocorrem interações diretas com o usuário, geralmente em navegadores web.

---

## **Git**

Sistema de controle de versão distribuído que permite rastrear alterações em arquivos e colaborar de forma eficiente no desenvolvimento de software.

---

## **Github**

Plataforma para hospedagem e gerenciamento de código-fonte, que utiliza o sistema Git para controle de versões e facilita a colaboração entre desenvolvedores.

---

## **Hardware**

Componentes físicos de um sistema de computação, incluindo processadores, memória, dispositivos de armazenamento e periféricos.

## **NestJS**

Framework Node.js progressivo e modular para construção de aplicações escaláveis e eficientes, usando TypeScript e conceitos de programação orientada a objetos.

---

## **NextJS**

Framework React que facilita o desenvolvimento de aplicações com renderização do lado do servidor, geração de páginas estáticas e funcionalidades de SEO.

## **Nodejs**

Ambiente de execução para JavaScript no lado do servidor, permitindo a criação de aplicações escaláveis e de alta performance.

---

### **Postgres**

Sistema gerenciador de banco de dados relacional de código aberto, conhecido por sua robustez, conformidade com padrões e extensibilidade.

---

### **Postman**

Ferramenta utilizada para testar e desenvolver APIs, permitindo a criação, execução e gerenciamento de requisições HTTP.

---

### **React**

Biblioteca JavaScript para construção de interfaces de usuário interativas e reutilizáveis, focada em componentes.

---

### **Shadcn**

Conjunto de componentes UI modernos e personalizáveis, muitas vezes integrados com Tailwind CSS, para construir interfaces consistentes e acessíveis.

---

### **Sistema operacional**

Software que gerencia os recursos do hardware, facilitando a execução de aplicações e a interação com o usuário.

---

### **Tailwind**

Framework CSS utilitário que permite a construção rápida de interfaces customizadas através de classes pré-definidas, sem sair do HTML.

---

## TypeORM

Biblioteca ORM (Object-Relational Mapping) para Node.js, que facilita a interação com bancos de dados utilizando TypeScript ou JavaScript.

---

## Visual Studio Code

Editor de código-fonte leve e extensível, que oferece suporte para múltiplas linguagens e integrações para facilitar o desenvolvimento.

---

## Estereótipos UML

SA - System Actor (Ator do Sistema)

---

## Confidencialidade

© Londricostura, 2025