

UNIVERSIDADE TIRADENTES – UNIT
Curso de Ciência da Computação

PLATAFORMA DE GERENCIAMENTO DE CURSOS E TREINAMENTOS – CIDADE DE VÉRIDIA

MÓDULO DE NOTIFICAÇÕES

RELATÓRIO TÉCNICO FINAL – UNIDADES I E II

Disciplina: Projeto de Programação

Professora: Layse Santos

Autores:

Davi Lima dos Santos Filho

João Arthur Lima Rodrigues

Murilo Souza de Barros

Victor Evangelista Castro Rodrigues

Victor Gabriel Reis Oliveira

Aracaju – SE
2025

Sumário

Resumo	2
1 Introdução	3
2 Objetivos	3
2.1 Objetivo Geral	3
2.2 Objetivos Específicos	3
3 Desenvolvimento do Módulo	4
3.1 Escopo e Funcionalidades	4
3.2 Arquitetura e Tecnologias	4
4 Modelagem do Banco de Dados	5
5 Diagramas UML	5
5.1 Casos de Uso	5
5.2 Diagrama de Classes	5
6 Interface e API REST	6
7 Testes e Resultados	6
8 Conclusão	7
Referências	7

Resumo

O presente relatório apresenta o desenvolvimento do **Módulo de Notificações** da Plataforma de Gerenciamento de Cursos e Treinamentos da cidade de Véridia. Esse módulo tem como principal objetivo otimizar a comunicação entre alunos, instrutores e administradores da plataforma, centralizando as mensagens e alertas automáticos gerados pelos demais módulos do sistema.

A implementação foi realizada utilizando a linguagem **Java 17** e o **framework Spring Boot**, com banco de dados **PostgreSQL** hospedado na Supabase. O módulo foi estruturado de forma integrada à arquitetura da aplicação, permitindo o envio, o registro e o controle de notificações personalizadas e administrativas.

O projeto proporcionou aprendizado significativo para o grupo, permitindo aplicar conceitos de arquitetura de software, desenvolvimento em camadas e modelagem de banco de dados. O módulo foi planejado para ser escalável, seguro e de fácil manutenção, contribuindo diretamente para a qualidade geral da plataforma.

Palavras-chave: Notificações; Comunicação; Spring Boot; PostgreSQL; Plataforma Educacional.

1. Introdução

Com o avanço da tecnologia e a expansão das plataformas digitais voltadas à educação, cresce a necessidade de sistemas capazes de integrar comunicação, gestão e acompanhamento de usuários. Nesse contexto, a **cidade de Véridia** manifestou o interesse em modernizar a forma como cursos e treinamentos são organizados, ofertados e acompanhados por alunos e instituições.

A cidade de Véridia deseja digitalizar e organizar a oferta de cursos e treinamentos oferecidos por instituições públicas e privadas. Para isso, a prefeitura contratou o nosso grupo para desenvolver uma **Plataforma de Gerenciamento de Cursos e Treinamentos**, que permita aos alunos se inscreverem em cursos, aos instrutores gerenciarem aulas e aos administradores acompanham toda a operação do sistema.

O sistema contempla diferentes perfis de usuários (alunos, instrutores e administradores) e possibilita o gerenciamento completo de cursos, controle de inscrições com pré-requisitos, processamento de pagamentos, agendamento de aulas, envio de notificações automáticas e geração de relatórios detalhados de desempenho e participação.

Dentro desse contexto, o **Grupo 6** foi responsável pelo desenvolvimento do **Módulo de Notificações**, que tem como papel garantir que todos os usuários sejam informados sobre eventos importantes de forma rápida e organizada. O módulo foi projetado para funcionar como o centro de comunicação da plataforma, garantindo que nenhum aviso relevante passe despercebido.

2. Objetivos

2.1. Objetivo Geral

Desenvolver um módulo de notificações automatizado e integrado à plataforma Véridia, garantindo comunicação eficiente e rastreável entre os diferentes usuários do sistema.

2.2. Objetivos Específicos

- Centralizar e organizar todas as notificações da plataforma;
 - Enviar alertas automáticos sobre eventos importantes;
 - Permitir o envio de comunicados administrativos;
 - Oferecer personalização de preferências de notificação;
 - Armazenar o histórico de mensagens enviadas;
 - Integrar o módulo com outros componentes da plataforma.
-

3. Desenvolvimento do Módulo

3.1. Escopo e Funcionalidades

O módulo de notificações foi desenvolvido com o intuito de oferecer uma comunicação mais dinâmica e eficiente dentro da plataforma. Ele é responsável por registrar, armazenar e exibir mensagens automáticas e administrativas, de forma personalizada para cada tipo de usuário.

Entre as principais funcionalidades implementadas, destacam-se:

- Envio automático de notificações sempre que um evento ocorre na plataforma (como nova aula, cancelamento ou confirmação de pagamento);
 - Envio de comunicados gerais por parte dos administradores;
 - Registro e exibição de histórico de notificações;
 - Personalização de canais de envio, como e-mail, pop-up ou aplicativo;
 - Marcação de notificações como lidas e exclusão individual ou em lote.
-

3.2. Arquitetura e Tecnologias

O sistema foi desenvolvido em **Java 17**, utilizando o **Spring Boot 3.x** para a estrutura do backend. O banco de dados escolhido foi o **PostgreSQL**, hospedado na Supabase, por oferecer robustez e excelente desempenho.

A arquitetura segue o padrão **Model-View-Controller (MVC)**, garantindo separação de responsabilidades e facilitando futuras manutenções.

Camadas Principais

- **Model:** Entidades do sistema (Notificação, Mensagem e Preferências);
 - **Repository:** Manipulação dos dados com Spring Data JPA;
 - **Service:** Implementação das regras de negócio e envio de mensagens;
 - **Controller:** Endpoints REST de comunicação com o frontend;
 - **View:** Interface visual desenvolvida em HTML, CSS e JavaScript.
-

4. Modelagem do Banco de Dados

O banco de dados foi modelado para garantir a integridade e a rastreabilidade das notificações enviadas.

- **Tabela de Notificações:** Armazena todas as notificações individuais com informações sobre tipo, prioridade, status e data de envio.
- **Tabela de Mensagens do Sistema:** Guarda comunicados gerais criados pelos administradores.
- **Tabela de Preferências:** Armazena as configurações de cada usuário, como canal de notificação preferido e categorias de alerta.

Essas tabelas possuem relacionamentos diretos com a tabela de usuários, garantindo integridade e rastreabilidade das mensagens enviadas.

—

5. Diagramas UML

5.1. Casos de Uso

Os principais casos de uso contemplam:

- Receber notificações automáticas;
- Consultar histórico de notificações;
- Marcar mensagens como lidas;
- Configurar preferências de recebimento;
- Enviar comunicados administrativos.

5.2. Diagrama de Classes

O diagrama de classes representa a estrutura principal dos módulos integrados:

- **Usuário:** Relaciona-se diretamente com notificações e preferências;
- **Curso/Aula:** Dispara eventos que geram notificações;
- **Pagamento:** Atualiza status e envia alertas automáticos;
- **Serviço de Notificação:** Controla todo o fluxo de envio e registro.

—

6. Interface e API REST

A interface visual foi criada com HTML5, CSS3 e JavaScript, buscando simplicidade e boa experiência de uso. As notificações são exibidas em forma de cards, com ícones, cores e legendas que indicam tipo e prioridade.

A API REST oferece endpoints para:

- Criar e listar notificações;
 - Marcar mensagens como lidas;
 - Consultar preferências e mensagens ativas;
 - Excluir notificações antigas.
-

7. Testes e Resultados

Durante os testes, o grupo simulou diferentes situações para garantir que o módulo funcionasse de forma estável. Foram testados:

- Envio automático de notificações a partir de eventos reais do sistema;
- Armazenamento e recuperação de mensagens no banco de dados;
- Filtros e contagem de notificações não lidas;
- Exibição correta de mensagens na interface.

Os resultados foram satisfatórios. O módulo funcionou conforme o esperado e se integrou com sucesso aos outros módulos da plataforma Véridia.

Plano de Implementação – Divisão de Módulos

A equipe definiu uma divisão de tarefas para otimizar o desenvolvimento do módulo e assegurar que cada etapa fosse concluída de forma organizada e colaborativa. Cada integrante foi responsável por uma parte específica, conforme descrito a seguir:

- **Davi Lima** – Modelagem, implementação de classes, construção do código;
- **João Arthur** – Casos de uso, documentação, construção do código, implementações;
- **Murilo Souza** – Modelagem, implementação, refatoração, banco de dados e junção dos códigos;
- **Victor Evangelista** – Modelagem, implementação de classes e testes;
- **Victor Gabriel** – Histórico de notificações, modelagem da interface.

8. Conclusão

O desenvolvimento do módulo de notificações foi um desafio importante dentro do projeto Véridia. Ele permitiu que a equipe aplicasse na prática conceitos de programação orientada a objetos, APIs REST, arquitetura MVC e manipulação de dados em banco relacional.

O módulo foi essencial para o sistema, pois garante uma comunicação clara entre todos os usuários da plataforma. Além disso, contribui para a transparência e a organização do fluxo de informações.

Como grupo, aprendemos muito sobre o processo de desenvolvimento em equipe, a importância da documentação e o cuidado com a integração entre módulos. O trabalho foi fundamental para nosso amadurecimento técnico e profissional.

Referências

- SPRING FRAMEWORK. *Spring Boot Documentation*. Disponível em: <https://spring.io/>. Acesso em: nov. 2025.
- POSTGRESQL. *PostgreSQL Official Documentation*. Disponível em: <https://www.postgresql.org/docs>. Acesso em: nov. 2025.
- SUPABASE. *Supabase Docs*. Disponível em: <https://supabase.com/docs>. Acesso em: nov. 2025.
- PRESSMAN, Roger S. *Engenharia de Software*. 8. ed. Porto Alegre: AMGH, 2016.
- SOMMERVILLE, Ian. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2019.