

**UCONNECT**  
**Documento de Arquitetura de Software**

**Versão 1.0**

## Histórico da Revisão

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
12/10/2025	1.0	Versão 1.0 final	Todos os integrantes do grupo

## Índice Analítico

<b>Introdução</b>	4
Finalidade	4
Escopo	4
Definições, Acrônimos e Abreviações	4
Referências	4
<b>Requisitos e Restrições da Arquitetura</b>	4
<b>Visão de Casos de Uso</b>	5
<b>Visão Lógica</b>	5
Visão Geral	5
Pacotes de Design Significativos do Ponto de Vista da Arquitetura	6
<b>Visão de Processos (opcional)</b>	6
<b>Visão de Implantação</b>	7
<b>Visão da Implementação (opcional)</b>	7
<b>Visão de Dados (opcional)</b>	7
<b>Volume e Desempenho</b>	7
<b>Qualidade</b>	7

# Documento de Arquitetura de Software

## 1. Introdução - Daniel

O documento tem como propósito definir e consolidar a arquitetura de software do sistema UCONNECT, servindo de base para as futuras fases do projeto.

### 1.1 Finalidade - Daniel

Este documento oferece uma visão geral arquitetural abrangente do sistema, usando diversas visões arquiteturais para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquitetônicas significativas que foram tomadas em relação ao sistema. O documento irá adotar uma estrutura baseada na visão “4+1” de modelo de arquitetura, conforme (Kruchten, 1994).

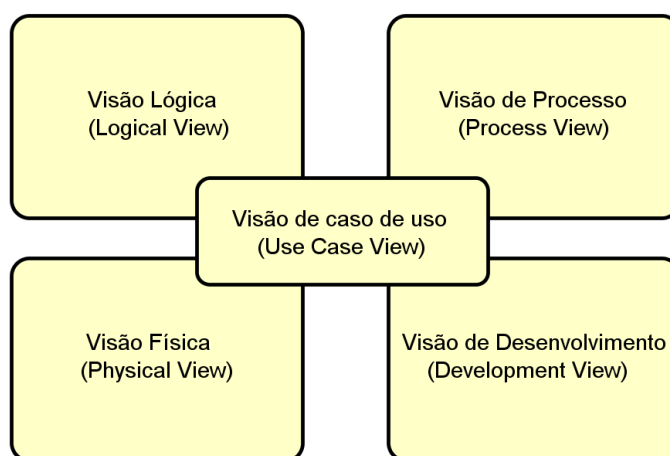


Figura 1 - Visões 4+1 da arquitetura de software (Kruchten, 1994)

### 1.2 Escopo - Paulo

O UCONNECT é uma plataforma web desenvolvida para instituições de ensino superior, com o objetivo de centralizar e unificar a comunicação acadêmica e a gestão de atividades em um único ambiente digital. Sua principal proposta é melhorar a experiência de alunos, professores e coordenadores, proporcionando uma forma mais eficiente e intuitiva de interagir com a instituição e com seus colegas. A plataforma visa eliminar a fragmentação das ferramentas atuais, como WhatsApp, Telegram, AVA e e-mail institucional, oferecendo uma solução única e acessível de qualquer dispositivo com internet.

O UCONNECT proporciona uma comunicação simplificada e centralizada, permitindo o envio de mensagens, publicação de avisos e gerenciamento de eventos acadêmicos em um único local. O sistema facilita o acompanhamento de atividades, prazos e informações importantes, além de permitir que as instituições possam organizar e gerenciar cursos, turmas e comunicados de maneira prática e eficiente.

### 1.3 Definições, Acrônimos e Abreviações - Gabriel

- **JWT (JSON Web Token)** – Padrão de autenticação baseado em tokens, usado para validar sessões de usuários.
- **OAuth 2.0** – Protocolo de autorização que permite acesso seguro a recursos sem expor credenciais do usuário.
- **RBAC (Role-Based Access Control)** – Controle de acesso baseado em papéis, no qual permissões são atribuídas a funções (aluno, professor, coordenador, admin).
- **OIDC (OpenID Connect)** – Camada de autenticação construída sobre o OAuth 2.0, usada para verificar

identidade de usuários.

- **MySQL** – Sistema de gerenciamento de banco de dados relacional escolhido para persistência do sistema.
- **React** – Biblioteca JavaScript utilizada para o desenvolvimento do front-end.
- **JavaScript** – Linguagem de programação usada no front-end.
- **Python** – Linguagem de programação que pode ser utilizada em componentes de backend ou serviços auxiliares.
- **SaaS (Software as a Service)** – Modelo em que o sistema é disponibilizado como serviço em nuvem, acessado via navegador.
- **TLS (Transport Layer Security)** – Protocolo de criptografia usado em comunicações seguras (HTTPS/TLS 1.3).
- **HTTPS (Hypertext Transfer Protocol Secure)** – Protocolo para comunicação segura pela web, usado no acesso ao sistema.
- **SLA (Service Level Agreement)** – Acordo de nível de serviço que define garantias de disponibilidade .

#### 1.4 Referências - Andressa

- Documento de Requisitos de Software – UCONNECT, Universidade Católica de Brasília (UCB).
- Modelo 4+1 de Arquitetura de Software – Kruchten, P. (1994).

## 2. Requisitos e Restrições da Arquitetura - João

Esta seção apresenta os principais requisitos e restrições que orientaram as decisões arquiteturais do sistema. Foram considerados aspectos como linguagem de programação, plataforma de execução, mecanismos de segurança, persistência de dados e internacionalização. A Tabela 1 resume as soluções adotadas para cada requisito identificado.

<i>Requisito</i>	<i>Solução</i>
<b>Linguagem</b>	<i>Python para backend (API REST), JavaScript com React para frontend.</i>
<b>Plataforma</b>	<i>WEB, acessível via navegador. Foco em responsividade e compatibilidade cross-browser.</i>
<b>Segurança</b>	<i>Autenticação via JWT e OAUTH 2.0, controle de acesso com RBAC, suporte a OIDC para integração com provedores externos.</i>
<b>Persistência</b>	<i>MySQL como banco de dados relacional, por ser robusto, amplamente suportado e adequado ao modelo de dados.</i>
<b>Internacionalização</b>	<i>A aplicação deve funcionar em português do Brasil (PT-BR), com suporte a tradução de textos e formatação local (datas, moeda).</i>

### 3. Visão de Casos de Uso - Daniel

D	Título	Valor de Negócio (Detalhado)	Prioridade (MoSCoW)	Estimativa (Story Points)	Dependências
S01	Acessar o sistema (Login)	<b>Crítico:</b> permite autenticar usuários, sem ele não há como acessar funcionalidades.	Must	5	Base de usuários
S02	Criar parametrizador de acessos	<b>Alto:</b> garante que a gestão de permissões seja flexível (admins, coordenadores, professores, alunos).	Must	8	US01
S03	Criar acesso	<b>Alto:</b> novos usuários só entram no sistema se puderem ter credenciais criadas.	Must	5	US02
S04	Editar acesso	<b>Médio:</b> permite manter dados de login atualizados (e-mails trocados, mudanças de perfil).	Should	3	US02
S05	Excluir acesso	<b>Médio:</b> remove acessos indevidos, mas não impacta no funcionamento base.	Should	3	US02
S06	Criar parametrizador de grupos	<b>Alto:</b> organiza a estrutura acadêmica (cursos/turmas) que é base para todas as interações.	Must	8	US01
S07	Criar grupo	<b>Alto:</b> usuários só fazem sentido no sistema quando estão em grupos (turmas/disciplinas).	Must	5	US06
S08	Editar grupo	<b>Médio:</b> ajustes de nomes ou composição de turmas aumentam flexibilidade, mas não travam o sistema.	Should	3	US06

S09	Excluir grupo	<b>Médio:</b> ajuda na manutenção da base, mas não é essencial para rodar o MVP.	Should	3	US06
S10	Criar publicação	<b>Alto:</b> centraliza a comunicação dos professores com os alunos (avisos, atividades).	Must	5	US01, US07
S11	Editar publicação	<b>Médio:</b> garante qualidade e correções em avisos, importante mas não vital.	Should	3	US10
S12	Excluir publicação	<b>Médio:</b> evita manter conteúdo incorreto, mas pode ser substituído por edição.	Should	3	US10
S13	Criar comunicado	<b>Alto:</b> canal oficial de comunicação institucional, crítico para confiança no sistema.	Must	5	US07
S14	Visualizar comunicado	<b>Alto:</b> sem isso, alunos não recebem informações oficiais.	Must	5	US13
S15	Visualizar calendário	<b>Alto:</b> principal recurso para organização de eventos acadêmicos.	Must	5	US01
S16	Visualizar evento	<b>Médio:</b> melhora a experiência, mas não é impeditivo.	Should	3	US15
S17	Criar evento	<b>Alto:</b> coordenadores/professores precisam agendar provas, aulas extras, prazos.	Must	5	US15, US07
S18	Editar evento	<b>Médio:</b> corrige ajustes de data/descrição, não é crítico no MVP inicial.	Should	3	US17

S19	Excluir evento	<b>Médio:</b> remove erros, mas pode ser substituído por edição no início.	Should	3	US17
S20	Visualizar perfil	<b>Alto:</b> dá identidade ao usuário no sistema e contexto de seus grupos.	Must	5	US01
S21	Visualizar outros perfis	<b>Médio:</b> facilita interação, mas não essencial no MVP.	Should	3	US20
S22	Receber notificações	<b>Muito Alto:</b> experiência em tempo real, aumenta engajamento e confiabilidade.	Must	8	US10, US13, US17
S23	Criar canal	<b>Alto:</b> base do chat em grupo, essencial para colaboração.	Must	5	US07
S24	Criar subcanais	<b>Médio:</b> melhora organização das conversas, mas não essencial no início.	Should	3	US23
S25	Editar canal/subcanal	<b>Baixo:</b> apenas ajuste administrativo.	Could	3	US23
S26	Excluir canal/subcanal	<b>Baixo:</b> manutenção do ambiente, não crítico no início.	Could	3	US23
S27	Enviar mensagem no chat	<b>Alto:</b> comunicação em tempo real é central ao produto.	Must	5	US23
S28	Editar mensagens	<b>Baixo:</b> conveniência, mas não essencial.	Could	3	US27
S29	Excluir mensagens	<b>Baixo:</b> recurso de moderação, importante mais tarde.	Could	3	US27



## 4. Visão Lógica - Vitor

### Divisão em Subsistemas/Pacotes

O sistema pode ser organizado em 4 camadas/pacotes lógicos:

#### Autenticação e Acessos

- Controle de login, sessões, permissões e perfis de usuário.
- Regras de negócio: RN-001, RN-002, RN-003.

#### Comunicação e Colaboração

- Chats, canais, subcanais, mural, notificações.
- Regras de negócio: RN-004, RN-007, RN-008.

#### Gestão Acadêmica

- Grupos (cursos, matérias, turmas), calendário, eventos, publicações, comunicados.
- Regras de negócio: RN-003, RN-004, RN-005.

#### Infraestrutura e Serviços

- Monitoramento, segurança, auditoria, failover, escalabilidade.
- Requisitos não funcionais: RNF001 a RNF021.

### Pacotes e Classes Significativas

#### 1. Pacote Autenticação e Acessos

##### Classe User (Usuário)

- **Atributos:** id, registration (matrícula), name (nome), email, passwordHash (senhaHash), role (perfil), accessStatus (acessoStatus);
- **Operações:** authenticate() (autenticar), updateProfile() (atualizarPerfil), checkPermission() (verificarPermissao);
- **Responsabilidade:** representar qualquer usuário do sistema.

##### Classe Session (Sessão)

- **Atributos:** token, startDate (dataInicio), expirationDate (dataExpiracao), user (usuario);
- **Operações:** startSession() (iniciarSessao), validateSession() (validarSessao), endSession() (encerrarSessao);
- **Responsabilidade:** gerenciar sessões ativas.

##### Classe AccessManager (GestorAcesso)

- **Atributos:** accessList (listaAcessos);
- **Operações:** createAccess() (criarAcesso), editAccess() (editarAcesso), deleteAccess() (excluirAcesso), listAccesses() (listarAcessos);

- **Responsabilidade:** administrar credenciais e permissões.

## 2. Pacote Comunicação e Colaboração

### Classe Channel (Canal)

- **Atributos:** id, name (nome), classGroup (turma), creator (criador), subchannelsList (listaSubcanais);
- **Operações:** createChannel() (criarCanal), editChannel() (editarCanal), deleteChannel() (excluirCanal);
- **Responsabilidade:** agrupar conversas de uma turma.

### Classe Subchannel (Subcanal)

- **Atributos:** name (nome), parentChannel (canalPai);
- **Operações:** createSubchannel() (criarSubcanal), deleteSubchannel() (excluirSubcanal);
- **Responsabilidade:** organização de conversas específicas.

### Classe Message (Mensagem)

- **Atributos:** id, content (conteudo), timestamp (dataHora), author (autor);
- **Operações:** send() (enviar), edit() (editar), delete() (excluir);
- **Responsabilidade:** comunicação em tempo real.

### Classe Notification (Notificacao)

- **Atributos:** id, type (tipo), content (conteudo), timestamp (dataHora), targetUser (usuarioDestino), readStatus (statusLida);
- **Operações:** sendNotification() (enviarNotificacao), markAsRead() (marcarComoLida);
- **Responsabilidade:** informar usuários sobre eventos relevantes.

## 3. Pacote Academic Management

### Classe AcademicGroup (GrupoAcademico)

- **Atributos:** id, course (curso), classGroup (turma), subject (disciplina), usersList (listaUsuarios);
- **Operações:** createGroup() (criarGrupo), editGroup() (editarGrupo), deleteGroup() (excluirGrupo);
- **Responsabilidade:** organizar usuários por curso/turma.

### Classe Post (Publicação)

- **Atributos:** id, title (titulo), content (conteudo), author (autor), date (data);
- **Operações:** create() (criar), edit() (editar), delete() (excluir);

- **Responsabilidade:** avisos, provas, atividades e oportunidades.

#### **Classe Announcement (Comunicado)** (especialização de Publicacao)

- **Atributos:** targetClasses (turmasDestino);
- **Operações:** publishAnnouncement() (publicarComunicado);
- **Responsabilidade:** transmitir informações oficiais para turmas específicas.

#### **Classe Event (Evento)**

- **Atributos:** id, title (titulo), description (descricao), timestamp (dataHora), classGroup (turma), creator (criador);
- **Operações:** createEvent() (criarEvento), editEvent() (editarEvento), deleteEvent() (excluirEvento);
- **Responsabilidade:** gerir calendário acadêmico.

#### **Classe Calendar (Calendario)**

- **Atributos:** eventsList (listaEventos);
- **Operações:** viewEvents() (visualizarEventos), addEvent() (adicionarEvento);  
v
- **Responsabilidade:** centralizar os eventos por usuário.

### **4. Pacote Infrastructure and Services**

#### **Classe SecurityService (ServicoSeguranca)**

- **Operações:** encryptData() (criptografarDados), validateCertificate() (validarCertificado), auditAccess() (auditarAcesso);
- **Responsabilidade:** manter confidencialidade e integridade.

#### **Classe ScalingService (ServicoEscalabilidade)**

- **Operações:** balanceLoad() (balancearCarga), startNewInstance() (iniciarNovaInstancia), monitorUsage() (monitorarUso);
- **Responsabilidade:** garantir desempenho em cenários de pico.

#### **Classe AvailabilityService (ServicoDisponibilidade)**

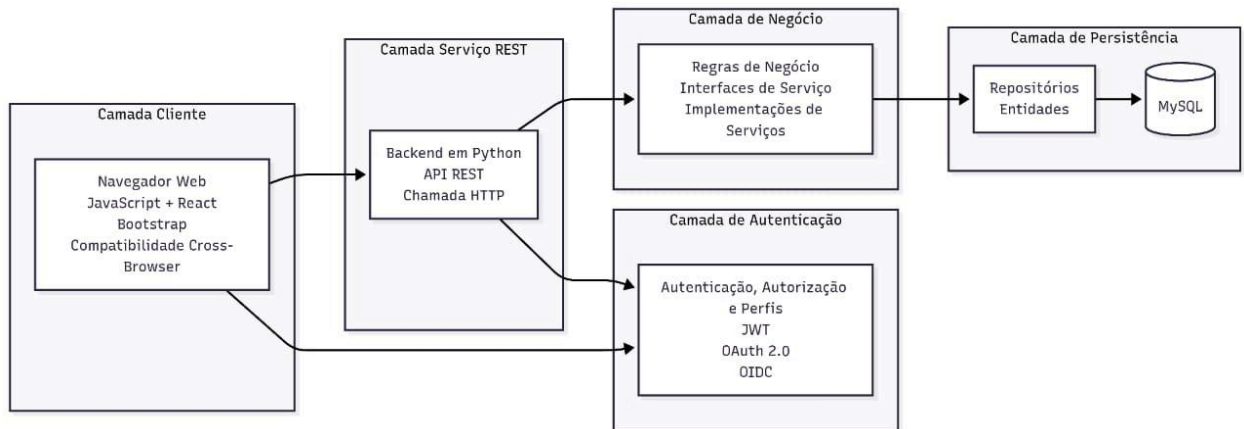
- **Operações:** activateFailover() (ativarFailover), monitorAvailability() (monitorarDisponibilidade);
- **Responsabilidade:** manter o SLA de 99,9%.

#### **Relacionamentos Relevantes**

- Usuario 1..\* — pertence a → Grupo

- Usuario 1..\* — participa de → Canal/Subcanal
- Usuario 1..\* — cria → Mensagem, Publicacao, Evento
- Publicacao — gera → Notificacao
- Calendario — contém → Evento
- Canal — contém → Subcanal
- Mensagem — pertence → Canal/Subcanal

#### 4.1 Visão Geral - João



#### 4.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura - Paulo

A arquitetura do UCONNECT foi organizada em quatro pacotes principais, que refletem as responsabilidades centrais do sistema. Cada pacote concentra classes que representam funcionalidades críticas, estando diretamente relacionadas aos casos de uso e regras de negócio definidos no Documento de Requisitos de Software.

##### 4.2.1. Pacote Autenticação e Acessos

Este pacote concentra os mecanismos de autenticação, autorização e gestão de perfis de usuário. Ele garante que apenas usuários válidos e devidamente autorizados tenham acesso às funcionalidades do sistema.

- **Classes principais:**
  - Usuario (User): representa os perfis cadastrados (aluno, professor, coordenador, administrador).
  - Sessao (Session): gerencia tokens de acesso e sessões ativas.
  - GestorAcesso (AccesManager): administra credenciais e permissões.
- **Regras de negócio relacionadas:** RN-001 (Autenticação), RN-002 (Gestão de Acessos).

##### 4.2.2. Pacote Comunicação e Colaboração

Responsável pelos recursos de interação e troca de informações em tempo real, reunindo ferramentas de chat, mural e notificações.

- **Classes principais:**
  - Canal e Subcanal (Channel / Subchannel): organizam as conversas por turma ou tema.
  - Mensagem (Message): permite envio, edição e exclusão de mensagens.
  - Notificacao (Notification): dispara alertas e mantém usuários informados sobre eventos relevantes.
- **Regras de negócio relacionadas:** RN-004 (Publicações), RN-007 (Notificações), RN-008 (Chat e Canais).

#### **4.2.3. Pacote Gestão Acadêmica**

Abrange funcionalidades de organização e controle de cursos, turmas, eventos e publicações. Ele possibilita a comunicação oficial entre professores, coordenadores e alunos, bem como o gerenciamento da agenda acadêmica.

- **Classes principais:**
  - Grupo (AcademicGroup): organiza usuários em cursos, disciplinas e turmas.
  - Publicacao e Comunicado (Post / Announcement): viabilizam a divulgação de avisos, atividades e comunicados oficiais.
  - Evento e Calendario (Event / Calendar): permitem a criação, edição e visualização de eventos acadêmicos.
- **Regras de negócio relacionadas:** RN-003 (Gestão de Grupos), RN-004 (Publicações e Comunicados), RN-005 (Calendário Acadêmico).

#### **4.2.4. Pacote Infraestrutura e Serviços**

Reúne os componentes que suportam requisitos não funcionais, como desempenho, disponibilidade, segurança e escalabilidade. Esse pacote garante a confiabilidade e robustez do sistema.

- **Classes principais:**
  - ServicoSeguranca (SecurityService): trata da criptografia de dados, validação de certificados e auditoria de acessos.
  - ServicoEscalabilidade (ScalingService): realiza balanceamento de carga e monitora uso de recursos.
  - ServicoDisponibilidade (AvailablilityService): assegura mecanismos de failover e manutenção do SLA de 99,9%.
- **Requisitos não funcionais relacionados:** RNF001 a RNF021.

#### **4.2.4 Relacionamentos entre Pacotes**

- O **Pacote Autenticação e Acessos** é a porta de entrada do sistema, liberando ou restringindo acesso aos demais pacotes.
- O **Pacote Comunicação e Colaboração** depende da autenticação para permitir interações e está integrado ao **Pacote Gestão Acadêmica**, recebendo publicações e comunicados como insumos para chats e notificações.
- O **Pacote Gestão Acadêmica** gera eventos, comunicados e publicações que são distribuídos através do **Pacote Comunicação e Colaboração**.
- O **Pacote Infraestrutura e Serviços** dá suporte transversal a todos os demais, garantindo segurança, disponibilidade e escalabilidade.

## **5. Visão de Implantação - Andressa**

O sistema UCONNECT será implantado em arquitetura baseada em nuvem, com suporte a escalabilidade horizontal e alta disponibilidade.

### **Configuração dos nós físicos:**

#### **Cliente**

- Dispositivos: desktops, notebooks, tablets e smartphones.
- Acesso via navegadores compatíveis (Chrome, Firefox, Edge, Safari).
- Comunicação estabelecida via HTTPS/TLS 1.3.

#### **Servidor Web**

- Responsável por hospedar toda a aplicação UCONNECT.
- Disponibiliza a interface ao usuário (React/JavaScript) e processa a lógica de negócio (autenticação, gestão acadêmica, comunicação, notificações).
- Implantado em contêineres, com suporte a escalabilidade horizontal para atender a picos de acesso (RNF003).
- Inclui mecanismos de segurança: autenticação JWT, OAuth 2.0, RBAC, OIDC; criptografia de dados em trânsito e repouso (RNF002, RNF008, RNF010).

#### **Servidor de Banco de Dados**

- MySQL 8.0, configurado em replicação.
- Armazena dados transacionais do sistema (mensagens, publicações, comunicados, eventos, perfis, notificações).
- Suporta estratégias de particionamento e arquivamento de longo prazo para históricos de anos letivos.

#### **Rede e Interconexões**

- Comunicação entre clientes, servidor web e banco de dados em rede privada segura (LAN virtual em nuvem).
- Balanceamento de carga entre instâncias do servidor web.

## 6. Visão de Dados - Vitor

### Pacote: Autenticação e Controle de Acesso

**Tabela User**

Nome do Campo	Tipo de Dado	Restrições	Descrição
id	INT	PRIMARY KEY, AUTO_INCREMENT	Identificador numérico único para cada usuário.
registration	VARCHAR(50)	UNIQUE, NOT NULL	Número de matrícula ou identificador de registro único do usuário.
name	VARCHAR(100)	NOT NULL	Nome completo do usuário.
email	VARCHAR(100)	UNIQUE, NOT NULL	Endereço de e-mail único do usuário, usado para login e comunicação.
passwordHash	VARCHAR(255)	NOT NULL	Hash da senha do usuário para armazenamento seguro.
role	ENUM(...)	NOT NULL	Perfil do usuário: <i>admin</i> , <i>coordinator</i> , <i>teacher</i> , <i>student</i> , <i>guest</i> .
accessStatus	ENUM(...)	DEFAULT 'active'	Status da conta: <i>active</i> , <i>inactive</i> , <i>suspended</i> .
createdAt	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Data e hora em que o registro foi criado.
updatedAt	TIMESTAMP	DEFAULT ... ON UPDATE ...	Data e hora da última atualização do registro.

---

**Tabela Session**

Nome do Campo	Tipo de Dado	Restrições	Descrição
token	VARCHAR(255)	PRIMARY KEY	Token de autenticação (ex: JWT) que identifica a sessão.
startDate	DATETIME	NOT NULL	Data e hora de início da sessão.
expirationDate	DATETIME	NOT NULL	Data e hora em que a sessão expira.
userId	INT	NOT NULL, FOREIGN KEY	Referência ao id da tabela <b>User</b> .

**Pacote: Comunicação e Colaboração**

**Tabela Channel**

Nome do Campo	Tipo de Dado	Restrições	Descrição
id	INT	PRIMARYKEY, AUTO_INCREMENT	Identificador único do canal.
name	VARCHAR(100)	NOT NULL	Nome do canal (ex: "Cálculo I - Turma A").
classGroup	VARCHAR(50)	-	Código da turma associada (opcional).
creatorId	INT	FOREIGN KEY, ON DELETE SET NULL	Usuário que criou o canal.
createdAt	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Data e hora de criação.



**Tabela Subchannel**

Nome do Campo	Tipo de Dado	Restrições	Descrição
id	INT	PRIMARYKEY, AUTO_INCREMENT	Identificador único do subcanal.
name	VARCHAR(100)	NOT NULL	Nome do subcanal (ex: "Avisos", "Trabalhos").
parentChannelId	INT	NOT NULL, FOREIGN KEY	Referência ao canal principal ( <b>Channel</b> ).

**Tabela Message**

Nome do Campo	Tipo de Dado	Restrições	Descrição
id	INT	PRIMARYKEY, AUTO_INCREMENT	Identificador único da mensagem.
content	TEXT	NOT NULL	Conteúdo textual da mensagem.
timestamp	DATETIME	DEFAULT CURRENT_TIMESTAMP	Data e hora do envio.
authorId	INT	NOT NULL, FOREIGN KEY	Usuário autor da mensagem.
channelId	INT	FOREIGN KEY	Canal onde foi enviada (nulo se em subcanal).
subchannelId	INT	FOREIGN KEY	Subcanal onde foi enviada (nulo se em canal).

**Tabela Notification**

Nome do Campo	Tipo de Dado	Restrições	Descrição
id	INT	PRIMARY KEY, AUTO_INCREMENT	Identificador da notificação.
type	ENUM(...)	NOT NULL	Tipo: <i>info</i> , <i>warning</i> , <i>alert</i> .
content	TEXT	NOT NULL	Conteúdo da notificação.
timestamp	DATETIME	DEFAULT CURRENT_TIMESTAMP	Data e hora de criação.
targetUserId	INT	NOT NULL, FOREIGN KEY	Usuário que receberá a notificação.
readStatus	BOOLEAN	DEFAULT FALSE	Lida (TRUE) ou (FALSE).

---

**Tabela Calendar**

Nome do Campo	Tipo de Dado	Restrições	Descrição
id	INT	PRIMARY KEY, AUTO_INCREMENT	Identificador do calendário.
userId	INT	NOT NULL, FOREIGN KEY	Usuário dono do calendário.
year	INT	NOT NULL	Ano do calendário.

---

**Tabela Calendar\_Event**

Nome do Campo	Tipo de Dado	Restrições	Descrição
calendarId	INT	PRIMARY KEY, FOREIGN KEY	Referência ao calendário.
eventId	INT	PRIMARY KEY, FOREIGN KEY	Referência ao evento.

**Pacote: Gestão Acadêmica**

**Tabela AcademicGroup**

Nome do Campo	Tipo de Dado	Restrições	Descrição
id	INT	PRIMARYKEY, AUTO_INCREMENT	Identificador único do grupo acadêmico.
course	VARCHAR(100)	NOT NULL	Nome do curso.
classGroup	VARCHAR(50)	NOT NULL	Código da turma.
subject	VARCHAR(100)	NOT NULL	Nome da disciplina.

**Tabela AcademicGroup\_User**

Nome do Campo	Tipo de Dado	Restrições	Descrição
groupId	INT	PRIMARY KEY, FOREIGN KEY	Referência ao grupo acadêmico.
userId	INT	PRIMARY KEY, FOREIGN KEY	Referência ao usuário.

**Tabela Post**

Nome do Campo	Tipo de Dado	Restrições	Descrição
id	INT	PRIMARY KEY, AUTO_INCREMENT	Identificador da publicação.
title	VARCHAR(200)	NOT NULL	Título da publicação.
content	TEXT	NOT NULL	Conteúdo principal.
authorId	INT	NOT NULL, FOREIGN KEY	Autor da publicação.
date	DATETIME	DEFAULT CURRENT_TIMESTAMP	Data e hora da publicação.

**Tabela Announcement**

Nome do Campo	Tipo de Dado	Restrições	Descrição
id	INT	PRIMARY KEY, FOREIGN KEY	Referência à tabela <b>Post</b> (1 para 1).
targetClasses	VARCHAR(255)	-	Turmas de destino (ex: "CC5P, SI3N").

**Tabela Announcement\_AcademicGroup**

Nome do Campo	Tipo de Dado	Restrições	Descrição
announcementId	INT	PRIMARY KEY, FOREIGN KEY	Referência à tabela <b>Announcement</b> (1 para 1).
groupId	INT	PRIMARY KEY, FOREIGN KEY	Referência à tabela <b>AcademicGroup</b> (1 para 1).

Tabela Event

Nome do Campo	Tipo de Dado	Restrições	Descrição
id	INT	PRIMARYKEY, AUTO_INCREMENT	Identificador do evento.
title	VARCHAR(200)	NOT NULL	Título (ex: "Prova P1 de Algoritmos").
description	TEXT	-	Descrição detalhada.
timestamp	DATETIME	NOT NULL	Data e hora do evento.
dateTime	DATETIME	NOT NULL	Data do evento.
startTime	DATETIME	NOT NULL	Horário de início do evento.
endTime	DATETIME	NOT NULL	Horário de término do evento.
academicGroupI d	VARCHAR(50)	-	Código da turma (opcional).
creatorId	INT	FOREIGN KEY, ON DELETE SET NULL	Usuário criador do evento.

## 7. Volume e Desempenho - Vitor

### Volume

- **Número estimado de usuários:** Sistema voltado para alunos, professores, coordenadores e demais funcionários, resultando em milhares de usuários.
- **Acessos simultâneos:** Devido a chats, mural e notificações em tempo real, o sistema deve suportar picos de uso durante horários de aula e prazos de entrega/provas.
- **Sessão de usuário:** Usuários permanecem logados durante longos períodos (aulas, acompanhamento de atividades, chat). Sessões de 1 a 3 horas são esperadas.
- **Volume de dados transacionais:** Mensagens de chat, comunicados, eventos de calendário, publicações e notificações em tempo real geram grande quantidade de registros no banco.
- **Notificações disparadas:** Cada publicação/evento pode gerar centenas ou milhares de notificações simultâneas.
- **Histórico de dados:** O sistema deve manter registros de chats, comunicados, eventos e perfis por anos

letivos, exigindo estratégias de arquivamento ou particionamento de banco de dados.

- **Armazenamento de mídia:** Anexos em comunicados, publicações ou chats (PDFs, imagens, vídeos curtos) aumentam a demanda de espaço em disco e largura de banda.

### Desempenho

- **Tempo de resposta do feed:** Atualização em menos de 1 segundo (RNF001).
- **Notificações em tempo real:** Devem ser entregues de forma instantânea (push).
- **Login:** Autenticação deve ser rápida, com resposta em até 2 segundos.
- **Criação/edição de eventos, publicações e mensagens:** Operação deve refletir no sistema em tempo quase imediato (<2 segundos).
- **Disponibilidade:** 99,9% do tempo, exceto manutenções programadas (RNF004).
- **Escalabilidade:** Suporte a crescimento via instâncias paralelas (horizontal scaling) (RNF003).
- **Continuidade de serviço:** Failover e load balancing para evitar indisponibilidade (RNF007).
- **Tempo de abertura do calendário/mural:** Exibição em até 2 segundos para até algumas centenas de eventos.
- **Latência de mensagens no chat:** Envio/recebimento em tempo real (<1 segundo).
- **Consumo de recursos:** Sistema deve ser eficiente para não sobrecarregar dispositivos móveis ou conexões de internet mais lentas.

## 8. Qualidade - Gabriel

<i>Item</i>	<i>Descrição</i>	<i>Solução</i>
<b>Escalabilidade</b>	Capacidade do sistema SaaS de crescer em número de usuários e dados sem perda de desempenho.	Arquitetura em nuvem que permite escalabilidade em diferentes aspectos (processamento, armazenamento e conexões), garantindo adaptação a variações de demanda.
<b>Confiabilidade</b>	Garantia de funcionamento correto mesmo em situações de falha ou alto uso.	Utilização de mecanismos que asseguram continuidade do serviço e recuperação em caso de falhas.
<b>Disponibilidade</b>	Garantia de que o sistema	Organização da arquitetura para

	permaneça acessível quando necessário pelos usuários.	reduzir interrupções e manter o serviço ativo de forma contínua.
<b><i>Portabilidade</i></b>	Possibilidade de uso em diferentes dispositivos e navegadores.	Aplicação Web responsiva, acessível exclusivamente por navegadores.
<b><i>Segurança</i></b>	Proteger dados sensíveis e controlar acessos de forma adequada.	Adoção de protocolos de autenticação, autorização e criptografia, garantindo a confidencialidade e integridade das informações.
<b><i>Usabilidade</i></b>	Facilidade de uso por diferentes perfis de usuários (alunos, professores, coordenadores).	Interface intuitiva, navegação clara e design responsivo, facilitando a interação com o sistema.