



FCI – Faculdade de Computação e Informática

PROJETO LABORATÓRIO DE ENGENHARIA DE SOFTWARE

PRIMEIRA ENTREGA

<<TINTIN >>

Nome: Diego Sugai
RA: 10417329

Nome: Kaue Alves
RA: 10417894

Nome: Leonardo Moreira
RA: 10417555

Nome: Marcos Arambasic Rebelo da Silva
RA: 10443260

Nome: Matheus Alonso Varjão
RA: 10417888

Nome: Victor Maki
RA: 10419861



Sumário

1.	Apresentação / Definição do Produto	3
2.	Definição da Demanda	3
2.1	Problema ou Oportunidade Percebida	3
2.2	Razão ou Justificativa para esta Demanda	3
2.3	Descrição Sucinta do Produto de Software	3
2.4	Clientes, Usuários e Demais Envolvidos/Impactados	3
2.5	Principais Etapas para Construção do Produto	3
2.6	Principais Critérios de Qualidade	4
3.	Requisitos Funcionais e Não Funcionais	4
3.1	Requisitos Não Funcionais	4
3.2	Requisitos Funcionais	5
4.	Modelagem	6
4.1	Diagrama de Caso de Uso	6
4.2	Diagrama de Classe	7
4.3	Diagrama de Sequência	8
5.	Arquitetura de Software	10
6.	Wireframe do protótipo	11



1. Apresentação / Definição do Produto

Nosso projeto é um aplicativo inspirado no Tinder, mas focado em conectar pessoas que querem aprender e ensinar ao mesmo tempo. A ideia é simples: cada usuário pode tanto ensinar quanto aprender sobre um assunto específico. Ao navegar pelo app, os usuários dão “swipe” em perfis de outras pessoas que possuem conhecimentos ou interesses que desejam trocar. Quando duas pessoas dão “match”, é aberto um chat onde elas podem conversar, trocar informações, tirar dúvidas e aprender juntas. É uma forma dinâmica, prática e social de expandir conhecimentos, unindo aprendizado com interação!

2. Definição da Demanda

2.1 Problema ou Oportunidade Percebida

Com a crescente busca por aprendizado informal e personalizado, muitas pessoas encontram dificuldades para acessar conteúdos e professores de forma dinâmica e social. Ao mesmo tempo, há uma demanda por plataformas que facilitem o encontro entre quem quer ensinar e quem quer aprender, sem burocracias e com interação direta. A oportunidade está em criar uma solução que combine a praticidade do modelo de “swipe” para conectar pessoas com interesses em comum, tornando o aprendizado mais acessível, engajante e colaborativo.

2.2 Razão ou Justificativa para esta Demanda

A crescente popularização de apps de relacionamento mostra que o modelo de “match” é eficiente para conectar pessoas com interesses similares. Aplicar essa lógica ao aprendizado pode facilitar trocas de conhecimento, motivar o ensino colaborativo e democratizar o acesso a conteúdos diversificados. Além disso, permite que o usuário assuma o papel de professor e aluno, promovendo uma comunidade ativa e engajada.

2.3 Descrição Sucinta do Produto de Software

O produto será um aplicativo móvel que permite aos usuários criar perfis detalhando temas que desejam aprender ou ensinar. Usando um sistema de “swipe”, os usuários poderão visualizar perfis de outros participantes com interesses educacionais compatíveis. Quando ocorre um “match”, é aberto um chat privado para que ambos possam conversar, trocar informações e realizar atividades de ensino/aprendizado.

2.4 Clientes, Usuários e Demais Envolvidos/Impactados

- Clientes: Instituições educacionais, organizações que incentivam educação contínua, e o público geral interessado em aprendizado colaborativo.
- Usuários: Pessoas que desejam aprender algo novo ou compartilhar seus conhecimentos, sem restrição de idade ou formação.
- Demais envolvidos: Desenvolvedores, equipe de suporte, e parceiros educacionais que possam fornecer conteúdos complementares.

2.5 Principais Etapas para Construção do Produto

- Levantamento detalhado dos requisitos junto aos usuários potenciais.



- Design da interface e experiência do usuário (UI/UX), priorizando simplicidade e engajamento.
- Desenvolvimento do backend para gerenciar perfis, sistema de “match” e chat.
- Implementação do aplicativo móvel (iOS e Android).
- Testes de usabilidade, segurança e desempenho.
- Lançamento beta para coleta de feedback e ajustes.
- Publicação oficial e monitoramento contínuo para melhorias.

2.6 Principais Critérios de Qualidade

- Usabilidade: Interface intuitiva e fácil navegação para todos os perfis de usuários.
- Confiabilidade: Sistema estável, sem falhas críticas no “match” e chat.
- Segurança: Proteção de dados pessoais e controle sobre interações entre usuários.
- Desempenho: Resposta rápida nas operações de swipe, busca e mensagens.
- Escalabilidade: Capacidade de suportar aumento gradual do número de usuários sem perda de qualidade.
- Engajamento: Recursos que incentivem a participação ativa e troca contínua de conhecimento

3. Requisitos Funcionais e Não Funcionais

3.1 Requisitos Não Funcionais

Tipo	Requisito	Descrição
RNF1	Desempenho	A plataforma deve suportar no mínimo 2 usuário simultâneo, com tempo de resposta inferior a 10 segundos.
RNF2	Escalabilidade	A aplicação deve ser escalável horizontalmente e suportar auto scaling na AWS com Docker.
RNF3	Segurança	Login e Senha
RNF4	Interface Responsiva	A interface deve se adaptar a diferentes dispositivos (smartphones, tablets, desktops).
RNF5	Usabilidade e Design	A plataforma deve ser intuitiva e simples, com um design de fácil navegação.
RNF6	Compatibilidade com Navegadores	A aplicação deve ser compatível com navegadores populares (Chrome, Firefox, Safari, Edge).



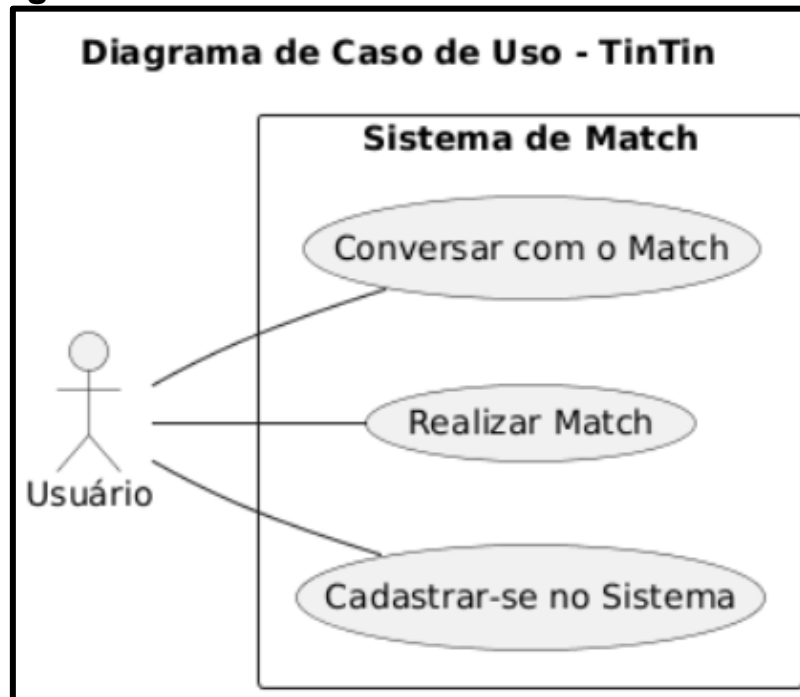
3.2 Requisitos Funcionais

Tipo	Requisito	Descrição
RF1	Cadastro de Usuário	O usuário pode se cadastrar usando login e senha
RF2	Criação de Perfil	O usuário cria um perfil com nome, foto, habilidades (ensinar ou aprender) e descrição.
RF3	Interação (Deslizar)	O usuário “clica” se teve “match” ou se não teve “match”
RF4	Match e Chat	Quando há um match, os usuários podem iniciar uma conversa via chat (mensagens de texto).
RF5	Sistema de Avaliação	Os usuários podem avaliar mutuamente após a troca de conhecimento (de 1 a 5 estrelas).
RF6	Mensagens	Permitir troca de mensagens entre usuários após o match.



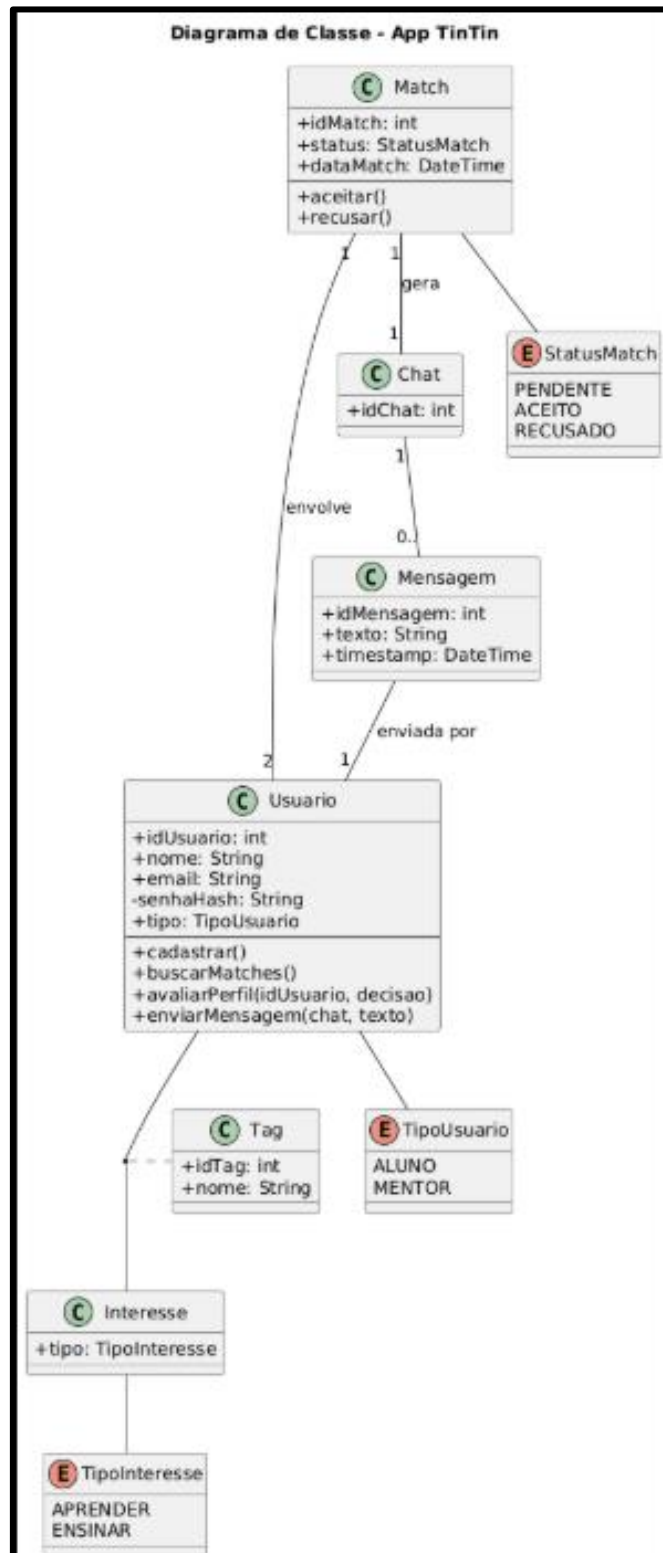
4. Modelagem

4.1 Diagrama de Caso de Uso





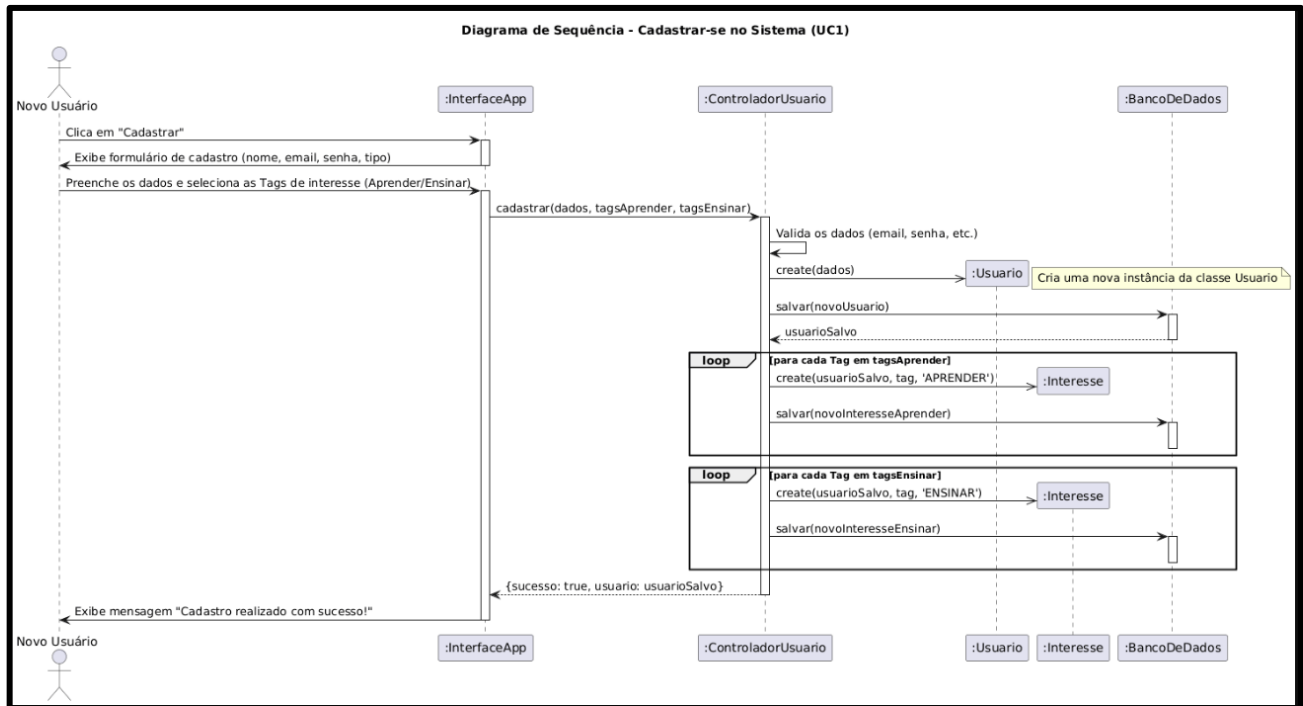
4.2 Diagrama de Classe



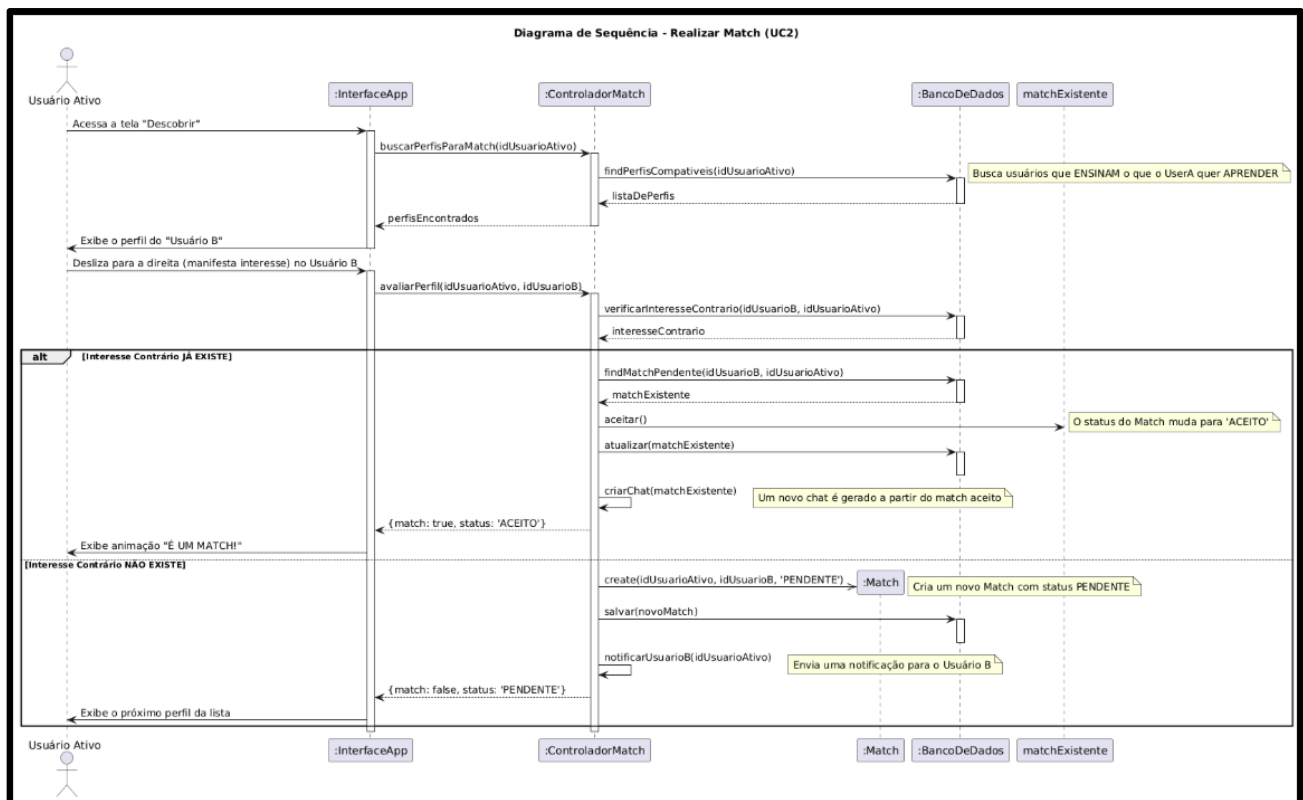


4.3 Diagrama de Sequência

4.3.1 Cadastrar-se no Sistema

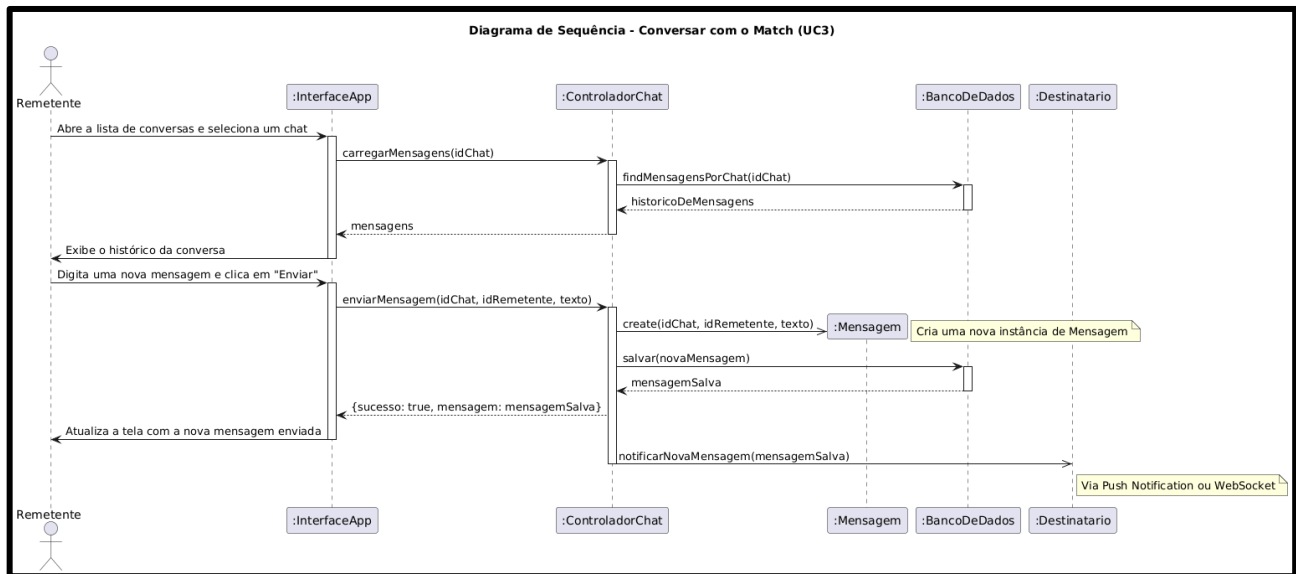


4.3.2 Realizar Match





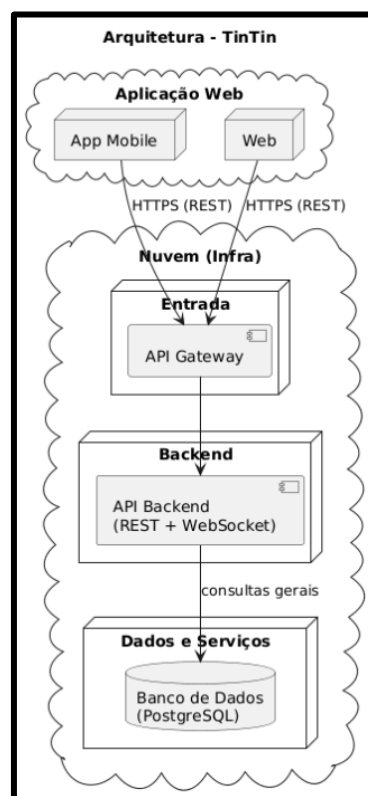
4.3.3 Conversar com o Match





5. Arquitetura de Software

O grupo decidiu adotar uma arquitetura composta por front-end Web (React) e Mobile (React Native), que acessam o sistema via API Gateway em HTTPS. As requisições são encaminhadas pelo gateway para o back-end em Java, responsável pela lógica de negócio envolvendo usuários, matches e chat em tempo real. O grupo optou por utilizar REST como padrão de comunicação entre cliente e servidor para garantir interações dinâmicas. Para persistência, escolheu-se o banco PostgreSQL, visando consistência e segurança dos dados. Essa divisão em camadas foi escolhida por simplificar a manutenção, ampliar a escalabilidade e facilitar futuras evoluções do projeto.





6. Wireframe do protótipo

O Wireframe do projeto foi desenvolvido no Figma, com o apoio da inteligência artificial da própria plataforma. Utilizando os recursos de IA, foi possível criar um protótipo dinâmico e interativo, permitindo a navegação fluida entre as telas e a simulação da experiência do usuário. Para visualizar e interagir com o wireframe, basta acessar o link abaixo:

[LINK_WIREFRAME](#)