

# Relatorio Final do Projeto

---

## Autores

- Aricarlo Alberto
- Edson Peixoto
- Kelvin Lamas

**Universidade:** IADE

**Curso:** Licenciatura em Engenharia Informática

**Cadeira:** Projeto Mobile

**Docente:** Pedro Rosa

**Data de Entrega:** 12/01/2025

---

## § Nome do Projeto

**IADE Social** - Plataforma de comunicação académica para alunos.

---

## § Enquadramento do Projeto

Ideia:

A **IADE Social** é uma plataforma digital destinada aos alunos da faculdade IADE, com o objetivo de fornecer informações essenciais sobre a vida académica, como eventos e anúncios da associação de estudantes. A ideia é criar um espaço centralizado de comunicação, onde os alunos possam visualizar e fazer postagens no mural público, com foco na disseminação de informações, sem funcionalidades de chat ou mensagens privadas.

Pesquisa sobre o Contexto:

Atualmente, muitas universidades não possuem um canal centralizado e eficiente para a divulgação de informações essenciais para os alunos. Aplicativos como portais de faculdades são voltados principalmente para aspectos administrativos e académicos (aulas, notas), deixando de lado a parte social e comunicativa. A **IADE Social** busca preencher essa lacuna, integrando informações académicas e sociais em uma única plataforma.

Objetivos:

- **Centralizar** a comunicação entre a faculdade e seus alunos.
- **Fornecer** um espaço para a associação de estudantes divulgar eventos e novidades.
- **Facilitar** o acesso rápido e eficiente a informações relevantes sobre a vida académica e social.

Público-Alvo:

Alunos da faculdade IADE que buscam um meio ágil e centralizado para se informar sobre novidades, eventos e comunicados importantes.

Pesquisa sobre Aplicações Existentes:

- **UniNow:** Um aplicativo popular entre universidades para centralizar dados académicos, mas com foco mais administrativo do que social.
  - **MyCampus:** Outra plataforma que visa organizar a vida académica dos alunos, porém sem ênfase na comunicação social ou em eventos estudantis.
  - **ClassApp:** Voltada para comunicação entre alunos e professores, mas sem recursos de compartilhamento de eventos ou informações institucionais.
- 

## § Guiões de Teste (Definições Completas)

### Guia 1: Registro e Primeira Utilização

1. O aluno abre o aplicativo pela primeira vez e seleciona a opção "Registrar-se".
2. Ele preenche as informações obrigatórias (nome, email, senha) e pode adicionar uma foto de perfil opcional.
3. O sistema verifica se os dados estão corretos e envia um email de confirmação.
4. Após a confirmação, o aluno é redirecionado para a página inicial.

### Guia 2: Login e Recuperação de Senha

1. O aluno já registrado abre o aplicativo e escolhe a opção "Entrar".
2. Ele insere seu email e senha.
3. Caso tenha esquecido a senha, clica em "Esqueceu a senha?" e segue as instruções de verificação (via email).
4. Se as credenciais estiverem corretas, o aluno é redirecionado para a página inicial.

### Guia 3: Publicar no Mural

1. O aluno faz login no aplicativo.

2. Na página inicial, seleciona "Fazer Postagem".
3. Ele insere o conteúdo (texto, imagem ou vídeo) e clica em "Publicar".
4. A postagem é exibida no mural público.

#### Guia 4: Interação com Posts

1. O aluno acessa o mural e visualiza os posts.
2. Ele pode clicar em um post para curtir, comentar ou visualizar comentários existentes.
3. O sistema atualiza as interações em tempo real.

---

## § Caso de Utilização "Core"

### Mural de Postagens

- **Descrição:** O principal caso de uso da **IADE Social** é o mural de postagens, onde todas as informações importantes, como eventos e anúncios, são centralizadas. Os alunos podem visualizar essas postagens e se manter informados sobre o que acontece na faculdade.

---

## § Outros 2 Casos de Utilização

### Caso de Uso 1: Publicação de Postagens

- **Descrição:** "Como aluno, quero fazer postagens no mural para compartilhar informações importantes com os outros alunos."

### Caso de Uso 2: Seguimento de Páginas

- **Descrição:** "Como aluno, quero seguir a página da associação de estudantes para acompanhar todas as suas publicações e eventos diretamente no meu feed."

---

## § Plano de Trabalhos

O projeto foi dividido em várias fases, conforme abaixo:

### Fase 1: Definir Design e Mockups

- Criar o design inicial e os mockups de todas as páginas da aplicação.

### Fase 2: Desenvolvimento

- Implementar o código da aplicação e criar a base de dados, seguindo o mockup desenvolvido.

### Fase 3: Testes

- Realizar testes funcionais para garantir que todas as funcionalidades estão operando corretamente.

### Fase 4: Apresentação à Associação de Estudantes

- Após o sucesso nos testes, apresentar a aplicação à associação de estudantes para feedback e testes.

### Fase 5: Entrega do Projeto e Lançamento

- Entregar o projeto e lançar a aplicação para uso pelos alunos.

### Fase 6: Divulgação

- Divulgar a aplicação aos estudantes da faculdade.

**Prazo de Entrega:** O projeto será entregue no dia **12/01/2025**.

---

## § Project Charter e WBS

### Project Charter:

Documento que define o escopo, os principais objetivos e as partes interessadas no projeto.

### WBS (Work Breakdown Structure):

Estrutura de divisão do trabalho, organizando as tarefas e subtarefas em entregas específicas do projeto, como design, desenvolvimento, testes, e lançamento.

## § Requisitos Funcionais e Não Funcionais

### Requisitos Funcionais:

1. Os alunos devem poder se registrar, fazer login e recuperar senha.
2. O mural deve exibir todas as postagens mais recentes de forma centralizada.
3. O sistema deve permitir que os alunos façam postagens e sigam páginas.

### Requisitos Não Funcionais:

1. A aplicação deve ser responsiva e funcionar em diferentes dispositivos e tamanhos de tela.
2. Os dados dos alunos devem ser armazenados de forma segura, garantindo privacidade e criptografia.
3. O carregamento das informações deve ser rápido, proporcionando uma boa experiência de usuário.

---

## § Modelo do Domínio

O modelo do domínio inclui as seguintes entidades principais:

- **Aluno:** Representa os usuários da aplicação.
- **Postagem:** As postagens feitas no mural.
- **Mural:** O espaço onde todas as postagens são exibidas.

---

## § Mockups e Interfaces

- **Tela de Login/Registro:** Tela inicial com campos de email, senha, e botão de registro ou recuperação de senha.
- **Página Inicial (Mural):** Exibe as postagens recentes e opções para seguir páginas.
- **Perfil de Usuário:** Página onde o aluno pode visualizar e editar suas informações pessoais.

---

## § Planificação (Gráfico de Gantt)

# IADE Social - Project Development

### Timeline (Gantt Chart)

## Cronograma do Projeto (Atualizado)

Semana	1	2	3	4	5	6	7	8	9	10	11	12
Design e Mockups	<div></div>	<div></div>										
Desenvolvimento			<div></div>	<div></div>	<div></div>	<div></div>						
Testes												
Apresentação e Lançamento								<div></div>				
Entrega e Lançamento								<div></div>				

### Resumo das Alterações

- **Design e Mockups:** Concluído nas semanas 1 e 2.
- **Desenvolvimento:** Estendido até a semana 6.
- **Testes:** Não realizados.
- **Apresentação e Lançamento:** Apresentação marcada para o dia 14, na semana 8.
- **Entrega e Lançamento:** Entrega realizada no dia 12, também na semana 8.
- **Promoção:** Removida do cronograma.

---

## § Documentação REST

### Estrutura dos Endpoints Identificados

#### CommentController (/api/comments)

- **GET /api/comments:** Retorna a lista de todos os comentários.

- **GET** `/api/comments/{id}`: Retorna os detalhes de um comentário específico.
- **POST** `/api/comments`: Cria um novo comentário.
- **PUT** `/api/comments/{id}`: Atualiza os dados de um comentário existente.
- **DELETE** `/api/comments/{id}`: Remove um comentário pelo ID.

#### FollowerController (`/api/Followers`)

- **GET** `/api/Followers`: Retorna a lista de todos os seguidores.
- **GET** `/api/Followers/{id}`: Retorna os detalhes de um seguidor específico.
- **POST** `/api/Followers`: Cria um novo seguidor.
- **PUT** `/api/Followers/{id}`: Atualiza os dados de um seguidor existente.
- **DELETE** `/api/Followers/{id}`: Remove um seguidor pelo ID.

#### UserController (`/api/users`)

- **GET** `/api/users`: Retorna a lista de todos os usuários.
- **GET** `/api/users/{id}`: Retorna os detalhes de um usuário específico.
- **POST** `/api/users`: Cria um novo usuário.
- **PUT** `/api/users/{id}`: Atualiza os dados de um usuário existente.
- **DELETE** `/api/users/{id}`: Remove um usuário pelo ID.

#### ProfileController (`/api/profiles`)

- **GET** `/api/profiles`: Retorna a lista de perfis.
- **GET** `/api/profiles/{id}`: Retorna os detalhes de um perfil.
- **POST** `/api/profiles`: Cria um novo perfil.
- **PUT** `/api/profiles/{id}`: Atualiza um perfil existente.
- **DELETE** `/api/profiles/{id}`: Remove um perfil pelo ID.

#### PostController (`/api/posts`)

- **GET** `/api/posts`: Retorna a lista de posts.
- **GET** `/api/posts/{id}`: Retorna os detalhes de um post.
- **POST** `/api/posts`: Cria um novo post.
- **PUT** `/api/posts/{id}`: Atualiza um post existente.
- **DELETE** `/api/posts/{id}`: Remove um post pelo ID.

#### LikeController (`/api/likes`)

- **GET** `/api/likes`: Retorna a lista de "likes".
- **GET** `/api/likes/{id}`: Retorna os detalhes de um "like".
- **POST** `/api/likes`: Cria um novo "like".
- **PUT** `/api/likes/{id}`: Atualiza um "like" existente.
- **DELETE** `/api/likes/{id}`: Remove um "like" pelo ID.

---

## § Guia de Dados

### 1. Tabela: Users

- **Descrição:** Armazena informações básicas dos utilizadores.
- **Campos:**
  - **UserID** (INT, PK, AUTO\_INCREMENT): Identificador único do utilizador.
  - **Username** (VARCHAR): Nome de utilizador escolhido.
  - **Email** (VARCHAR): Endereço de email do utilizador.
  - **Password** (VARCHAR): Senha para autenticação.
  - **Name** (VARCHAR): Nome completo do utilizador.
  - **StudentID** (INT): Número de estudante associado ao utilizador.

### 2. Tabela: Profiles

- **Descrição:** Contém informações adicionais sobre os utilizadores, como perfil e biografia.
- **Campos:**
  - **ProfileID** (INT, PK, AUTO\_INCREMENT): Identificador único do perfil.
  - **UserID** (INT, FK): Referência ao utilizador associado (ligado a **Users.UserID**).
  - **Name** (VARCHAR): Nome do perfil (geralmente igual ao nome do utilizador).
  - **Bio** (TEXT): Biografia ou descrição do utilizador.
  - **ProfilePicture** (VARCHAR): Caminho ou URL da imagem de perfil.

### 3. Tabela: Posts

- **Descrição:** Armazena os posts (publicações) criados pelos utilizadores.
- **Campos:**
  - **PostID** (INT, PK, AUTO\_INCREMENT): Identificador único do post.
  - **UserID** (INT, FK): Referência ao autor do post (ligado a **Users.UserID**).
  - **Content** (TEXT): Conteúdo da publicação.

### 4. Tabela: Comments

- **Descrição:** Armazena os comentários feitos nos posts.
- **Campos:**
  - **CommentID** (INT, PK, AUTO\_INCREMENT): Identificador único do comentário.
  - **PostID** (INT, FK): Referência ao post comentado (ligado a **Posts.PostID**).
  - **UserID** (INT, FK): Referência ao autor do comentário (ligado a **Users.UserID**).
  - **Content** (TEXT): Texto do comentário.

### 5. Tabela: Likes

- **Descrição:** Regista os likes feitos em posts por utilizadores.
- **Campos:**
  - **LikeID** (INT, PK, AUTO\_INCREMENT): Identificador único do like.
  - **PostID** (INT, FK): Referência ao post que recebeu o like (ligado a **Posts.PostID**).
  - **UserID** (INT, FK): Referência ao utilizador que deu o like (ligado a **Users.UserID**).

### 6. Tabela: Followers

- **Descrição:** Regista as relações de seguidores entre os utilizadores.
- **Campos:**
  - **FollowerID** (INT, PK, AUTO\_INCREMENT): Identificador único da relação de seguidores.
  - **UserID** (INT, FK): Utilizador que está a ser seguido (ligado a **Users.UserID**).
  - **FollowerUserID** (INT, FK): Utilizador que está a seguir (ligado a **Users.UserID**).

---

## § Relacionamentos Entre Tabelas

1. **Users → Profiles:** Um utilizador (**Users.UserID**) pode ter apenas um perfil associado (**Profiles.UserID**).
2. **Users → Posts:** Um utilizador (**Users.UserID**) pode criar múltiplos posts (**Posts.UserID**).
3. **Posts → Comments:** Um post (**Posts.PostID**) pode ter vários comentários (**Comments.PostID**).
4. **Posts → Likes:** Um post (**Posts.PostID**) pode receber múltiplos likes (**Likes.PostID**).
5. **Users → Comments:** Um utilizador (**Users.UserID**) pode fazer vários comentários (**Comments.UserID**).
6. **Users → Likes:** Um utilizador (**Users.UserID**) pode dar múltiplos likes (**Likes.UserID**).
7. **Users → Followers:** Um utilizador (**Users.UserID**) pode seguir ou ser seguido por vários utilizadores (**Followers.UserID** e **Followers.FollowerUserID**).

---

## § Diagrama de Classes

### Classe: User

- **Atributos:**
  - **userID:** int
  - **email:** String
  - **senha:** String
  - **nomeUsuario:** String

---

### Classe: Profile

- **Atributos:**
  - **profileID:** int
  - **userID:** int
  - **fotoPerfil:** String

---

### Classe: Post

- **Atributos:**
  - **postID:** int
  - **userID:** int
  - **conteudo:** String

- **dataCriacao**: DateTime

---

Classe: Comment

- **Atributos:**
  - **commentID**: int
  - **postID**: int
  - **userID**: int
  - **conteudo**: String
  - **dataCriacao**: DateTime

---

Classe: Like

- **Atributos:**
  - **likeID**: int
  - **postID**: int
  - **userID**: int

---

Classe: Follower

- **Atributos:**
  - **followerID**: int
  - **userID**: int
  - **followerUserID**: int

---

## Relações

- **User (1) ↔ (1) Profile**
- **User (1) ↔ (\*) Post**
- **Post (1) ↔ (\*) Comment**
- **Post (1) ↔ (\*) Like**
- **User () ↔ () Follower**

---

## Manual de Utilizador

### Bem-vindo à IADE Social!

#### Passo 1: Registro e Primeira Utilização

1. Abra o aplicativo **IADE Social**.
2. Clique na opção **"Registrar-se"**.
3. Preencha as informações necessárias:
  - Nome
  - Email
  - Senha
  - (Opcional) Carregue uma foto de perfil.
4. Clique em **"Concluir Registro"** para criar sua conta.
5. Após o registro, você será redirecionado para a página inicial.

---

#### Passo 2: Login

1. Na tela inicial, clique em **"Entrar"**.
2. Insira seu **email** e **senha** cadastrados.
3. Caso tenha esquecido a senha, clique em **"Esqueceu a senha?"** e siga as instruções para recuperá-la.
4. Ao fazer login, você será redirecionado para a página inicial.

---

#### Passo 3: Interação na Página Inicial

1. Na página inicial, você pode:
  - Fazer **publicações no mural**.
  - **Pesquisar informações**.
  - **Seguir páginas** da associação de estudantes.
2. Clique em **"Perfil"** para editar suas informações pessoais, como nome de usuário e foto de perfil.

---

#### Passo 4: Participar de Eventos e Workshops

1. Explore as publicações do mural para descobrir **eventos**, **workshops** e **atividades relevantes**.
  2. Clique nos posts para mais informações e interaja com curtidas ou comentários.
- 

## Aplicações Utilizadas

---

As aplicações utilizadas que nos ajudaram no desenvolvimento da nossa aplicação foram:

- **Android Studio/Code**
  - **Postman/API**
  - **MySQL/BD**
  - **MAMP/BD**
  - **Visual Studio/API**
  - **Figma/Design e Mockup**
- 

## Contributo das Cadeiras

---

- **Matemática Discreta:**  
Ajudou-nos a estruturar raciocínios lógicos fundamentais para a construção de algoritmos e modelos de dados.
- **Competências Comunicacionais:**  
Foi essencial para aprimorarmos a apresentação das ideias, através do pitch, da criação de vídeos e da definição de personas.
- **Bases de Dados:**  
Aprendemos a organizar e gerenciar as informações que alimentam a aplicação.
- **Programação Orientada a Objetos:**  
Foi crucial para a criação de uma arquitetura robusta e escalável.
- **Programação de Dispositivos Móveis:**  
Nos forneceu as bases técnicas para desenvolver uma aplicação funcional e intuitiva.