



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS

Universidade Federal do Ceará - UFC
Curso de Bacharelado em Ciência da Computação
Projeto Integrador III

Maria Fernanda Aquino Freitas Scarcela
Giovanna Maria Veríssimo Xavier
Stephany Rodrigues Gonçalves
Emerson de Sousa Carvalho
Hitalecio Gomes Mesquita

RELATÓRIO PARA O PROJETO INTEGRADOR III - ANUNCIAUFC

Crateús - Ceará
07 de Agosto de 2025

Índice

1. Introdução.....	3
1.1. Propósito do documento.....	3
1.2. Apresentação Geral do Sistema.....	3
1.3. Organização da documentação.....	4
1) Análise de Requisitos.....	4
2) Modelagem de Diagramas.....	4
2. Análise de Requisitos.....	9
2.1. Requisitos Funcionais.....	9
2.2. Requisitos Não Funcionais.....	10
2.3. Requisitos de Domínio.....	11
3. Diagramas.....	11
3.1. Diagrama de Casos de Uso.....	11
3.2. Diagrama de Classes.....	18
3.3. Diagramas de Sequência.....	21
3.4. Diagramas de Pacotes.....	22
4. 6. Protótipos de Interface de Usuário.....	33

1. Introdução

1.1. Propósito do documento

Este relatório apresenta os princípios que orientaram a análise de requisitos e a modelagem do sistema AnunciaUFC, desenvolvido na disciplina Projeto Integrador III da UFC (2025.1). Ele descreve as principais funcionalidades, interfaces e comportamentos esperados, servindo de base para o desenvolvimento, alinhamento da equipe e futuras manutenções.

1.2. Apresentação Geral do Sistema

Problemática

A principal forma de ingresso na Universidade Federal do Ceará é pelo Sistema de Seleção Unificada (SiSU), que permite que estudantes de diversas regiões do país concorram a vagas em diferentes estados. Diante disso, a universidade deve estar preparada para acolher esses discentes, facilitando sua adaptação ao novo ambiente acadêmico.

Ao ingressar, muitos calouros buscam moradias na região por meio de plataformas na internet, geralmente procurando dividir aluguel com outros estudantes. No entanto, essas oportunidades geralmente são compartilhadas apenas em grupos fechados ou entre amigos, dificultando o acesso a informações sobre moradias disponíveis.

Além disso, durante a graduação, é comum que os alunos adquiram livros, calculadoras, computadores e outros materiais acadêmicos, que, em algum momento, desejam vender, doar ou trocar. Um exemplo é no final do curso que, ao retornarem para suas cidades de origem, muitos estudantes precisam se desfazer de móveis e eletrodomésticos, que poderiam ser úteis para novos alunos em busca de itens de segunda mão a preços acessíveis.

Apresentação do sistema

Pensando nessas dificuldades e no acolhimento dos discentes, o **AnunciaUFC** foi desenvolvido como uma iniciativa do **Programa de Educação Tutorial (PET) da UFC - campus Crateús**. O **AnunciaUFC** é um sistema de anúncios desenvolvido exclusivamente para discentes e servidores da Universidade Federal do Ceará (UFC). Seu propósito é oferecer um ambiente seguro e direcionado para que os usuários possam anunciar a venda de itens novos ou usados, além de divulgar serviços, facilitando a negociação entre os membros da comunidade acadêmica e tornando sua comunicação e intercâmbio mais acessíveis e organizados.

Para garantir um sistema moderno, responsivo e eficiente, o AnunciaUFC foi desenvolvido utilizando **React** e **Tailwind CSS** no front-end, com **TypeScript** para maior segurança no código. No back-end, utilizamos **Flask**, garantindo uma API robusta e escalável, e **MySQL** como banco de dados para armazenar as informações dos anúncios e usuários.

1.3. Organização da documentação

Este documento apresenta as principais etapas adotadas na concepção e desenvolvimento do sistema AnunciaUFC, com foco na **Análise de Requisitos** e na **Modelagem de Diagramas**, seguindo os princípios fundamentais da engenharia de software. Além de documentar o processo técnico, também registra a distribuição de atividades entre os membros da equipe.

1) Análise de Requisitos

Após a definição dos objetivos do sistema, a **análise de requisitos** torna-se um passo essencial para garantir que o desenvolvimento atenda às expectativas e necessidades dos usuários finais. Essa etapa desempenha um papel crucial na estruturação do projeto, permitindo antecipar desafios, organizar a implementação de forma eficiente, reduzir falhas e retrabalho.

No contexto deste projeto, a identificação dos requisitos do sistema ocorre por meio de reuniões com os stakeholders, em especial a **coordenadora do PET**, que é a principal cliente e usuária desse sistema. A partir dessas interações, as necessidades da comunidade acadêmica são compreendidas e documentadas corretamente, garantindo que o sistema seja funcional, eficiente e útil para discentes e servidores da UFC. Na seção 2, são descritos os três tipos de requisitos identificados durante a análise:

1. **Requisitos funcionais:** Descrevem o que o sistema deve fazer, ou seja, suas funcionalidades principais.
2. **Requisitos não funcionais:** Definem restrições e qualidades do sistema, como desempenho, segurança e usabilidade.
3. **Requisitos de domínio:** Especificam regras e particularidades do contexto em que o sistema será utilizado.

2) Modelagem de Diagramas

Após a análise e definição dos requisitos, a **modelagem de diagramas** também desempenha um papel essencial no desenvolvimento do sistema. Essa etapa permite representar visualmente a estrutura e o comportamento do sistema, facilitando a compreensão tanto para a equipe técnica quanto para os stakeholders. A utilização de diagramas é especialmente importante neste projeto, pois a **coordenadora do PET**, nossa principal cliente, não possui formação em TI. Assim, os diagramas funcionam como uma ponte entre a concepção do sistema e sua implementação, tornando mais claro o fluxo de informações, as interações entre os usuários e a organização dos componentes do sistema.

Para garantir uma visão clara do sistema, alguns diagramas fundamentais aplicados são descritos na seção 3 tais como:

1. **Diagrama de Casos de Uso** – Representa as principais funcionalidades do sistema e como os usuários interagem com elas.
2. **Diagrama de Classes** – Estrutura os componentes do sistema, demonstrando as relações entre as entidades e seus atributos.
3. **Diagrama de Sequência** – Detalha o fluxo das interações entre os usuários e o sistema ao longo do tempo.
4. **Diagrama de Estados** – Define os diferentes estados pelos quais uma entidade pode passar durante seu ciclo de vida.

1.4. Divisão das tarefas

As responsabilidades foram distribuídas entre os membros da seguinte forma:

Front-end

Protótipo no Figma e Modelagem do sistema: Maria Fernanda Aquino Freitas Scarcela e Stephany Rodrigues Gonçalves

Desenvolvimento do front-end: Emerson de Sousa Carvalho e Maria Fernanda Aquino Freitas Scarcela.

Back-end

Funções da API: Giovanna Maria Veríssimo Xavier e Hitalécio Gomes Mesquita

Conexão com o banco de dados: Hitalécio Gomes Mesquita

2. Análise de Requisitos

2.1. Requisitos Funcionais

ID	Descrição	Prioridade	Rastreabilidade
RF01	O sistema deve permitir que o usuário faça autenticação.	Alta	Caso de uso: CU01
RF02	O sistema deve permitir que o usuário se cadastre.	Alta	Caso de uso: CU02
RF03	O sistema deve permitir que o usuário recupere sua senha.	Alta	Caso de uso: CU03
RF04	O sistema deve permitir que o usuário altere suas informações informadas no ato do cadastro, com exceções do e-mail.	Média	Caso de uso: CU04
RF05	O sistema deve permitir que o usuário exclua definitivamente sua conta.	Alta	Caso de uso: CU05
RF06	O sistema deve permitir que o usuário redefina sua senha de acesso.	Alta	Caso de uso: CU06
RF07	O sistema deve permitir que o usuário crie anúncios.	Alta	Caso de uso: CU07
RF08	O sistema deve permitir ao usuário editar seus anúncios.	Alta	Caso de uso: CU08
RF09	O sistema deve permitir que o administrador autorize anúncios que estão dentro das normas de segurança.	Alta	Caso de uso: CU09
RF10	O sistema deve permitir que o administrador rejeite anúncios que não estão dentro das normas de segurança.	Alta	Caso de uso: CU10
RF11	O sistema deve permitir que o usuário exclua anúncios permanentemente.	Alta	Caso de uso: CU11
RF12	O sistema deve permitir que o usuário visualize detalhes de anúncios disponíveis na plataforma, criados por outros usuários.	Alta	Caso de uso: CU12

Ps: RF = Requisito Funcional.

2.2. Requisitos Não Funcionais

ID	Descrição	Critérios de Aceitação
RNF01	O sistema deve armazenar as senhas dos usuários utilizando o bcrypt, garantindo que não fiquem acessíveis em texto puro no banco de dados.	Verificação de armazenamento no banco de dados e testes de autenticação.
RNF02	A interface deve ser responsiva, se adaptando a diferentes tamanhos de tela, tanto em dispositivos móveis quanto em desktops.	Testes de usabilidade devem ser conduzidos.
RNF03	A interface do sistema deve ser simples, intuitiva e permitir fácil navegação.	Testes de usabilidade devem ser realizados.
RNF04	O sistema deve utilizar tokens JWT para autenticação de usuários, garantindo que o token tenha um tempo de expiração (2 horas) e seja armazenado de forma segura.	Testes de segurança devem confirmar que tokens expirados são rejeitados e que a comunicação do token está protegida.
RNF05	O sistema deve ser capaz de processar requisições de autenticação em menos de 1 segundo, mesmo sob carga elevada de usuários simultâneos.	Simular requisições de autenticação (login) com até 200 usuários simultâneos e garantir que o tempo médio de resposta para cada autenticação não ultrapasse 1 segundo.
RNF06	O sistema deve ser compatível com os navegadores Chrome, Edge e Opera, garantindo total funcionalidade.	Testes de funcionalidade nos navegadores Google Chrome, Mozilla Firefox e Microsoft Edge (últimas versões estáveis), devem ser feitos garantindo que a interface seja renderizada corretamente, os componentes funcionem conforme esperado e não haja erros críticos no console do desenvolvedor.

Ps: RNF = Requisito não funcional.

2.3. Requisitos de Domínio

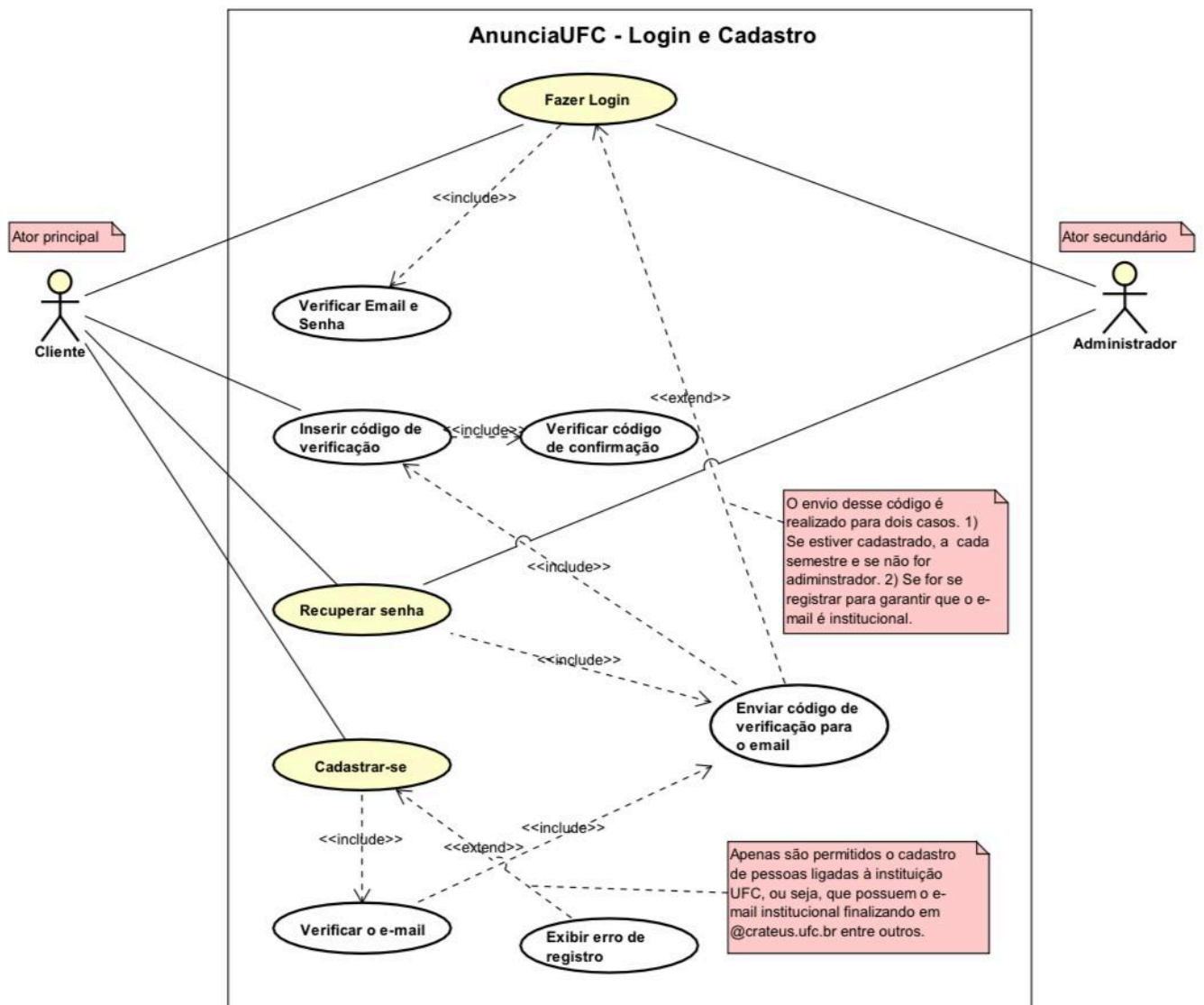
ID	Descrição
RD01	O sistema deve permitir o acesso apenas a usuários que possuam um e-mail institucional válido (@alu.ufc.br ou @crateus.ufc.br), impedindo o cadastro e login com endereços de e-mail externos.
RD02	Durante o processo de cadastro, o sistema deve enviar um código de verificação único para o e-mail institucional informado pelo usuário. O cadastro só será concluído após a inserção e validação deste código no sistema.
RD03	Todos os anúncios cadastrados no sistema devem ser revisados por um administrador antes da publicação, garantindo que não contenham conteúdos que violem diretrizes da plataforma, incluindo preconceito, discriminação ou qualquer forma de discurso ofensivo.
RD04	Caso um anúncio seja recusado pelo administrador, ele deve ser excluído do sistema, e um e-mail deve ser enviado ao anunciante informando o motivo da recusa.
RD05	Para garantir que o e-mail institucional do usuário ainda é válido, o sistema deve exigir, a cada seis meses, a confirmação do endereço de e-mail por meio de um código de verificação enviado para ele. O acesso só será concedido após a validação do código.
RD06	O sistema deve permitir apenas a publicação de anúncios, sem intermediar vendas ou transações financeiras. Caso um usuário se interesse por um anúncio, ele será direcionado para entrar em contato com o anunciante via ligação telefônica ou WhatsApp.
RD07	Os anúncios devem ser classificados em categorias predefinidas pelo sistema para facilitar a busca e organização.

Ps: RD = Requisito de Domínio.

3. Diagramas

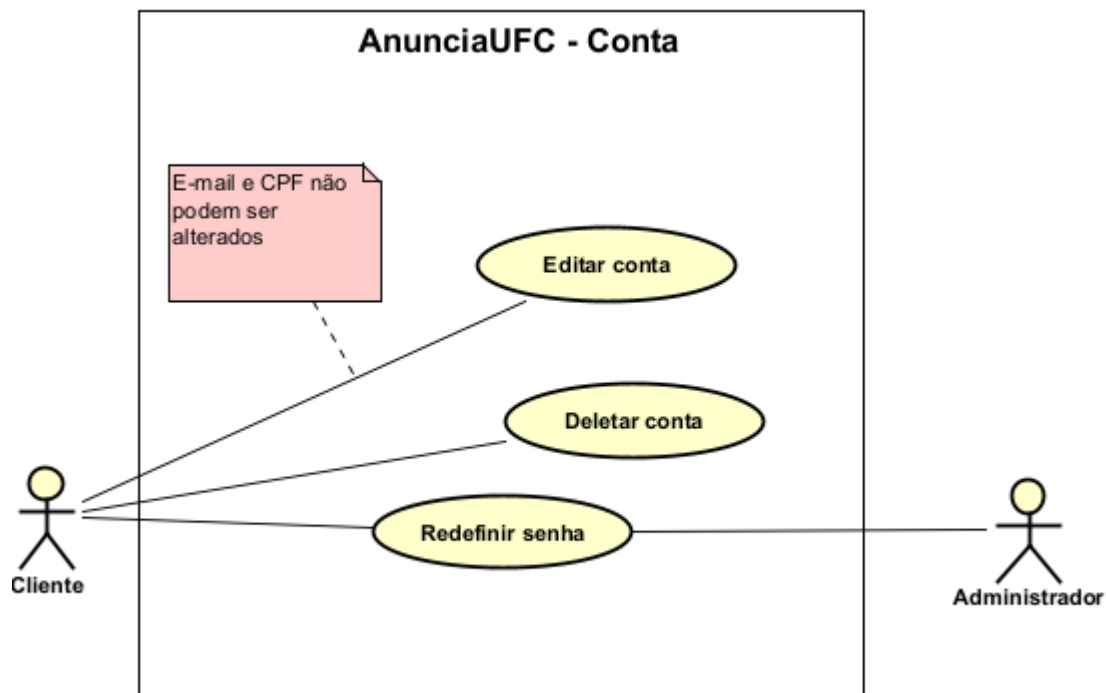
3.1. Diagrama de Casos de Uso

Os atores são: **Clientes** e **Administrador**. Ambos são uma especialização de **Usuário**.



Fazer Login (CU01)
Fluxo Principal
<ol style="list-style-type: none"> 1. O sistema solicita o e-mail e a senha de acesso do usuário. 2. O usuário fornece o e-mail e a senha. 3. O sistema valida as credenciais. 4. O sistema reconhece o usuário e concede o acesso.
Fluxos Alternativos
<ol style="list-style-type: none"> 1. O sistema envia um código de verificação semestralmente ao fazer login (apenas para clientes).

<ol style="list-style-type: none"> O cliente insere o código de verificação. Se o código for válido, o sistema reconhece o usuário e concede o acesso. Caso contrário, exibe uma mensagem de erro.
Se o e-mail ou a senha estiverem inválidos, o sistema exibe uma mensagem de erro.
Cadastrar-se (CU02)
Fluxo Principal
<ol style="list-style-type: none"> O sistema solicita o e-mail institucional do usuário. O sistema verifica se o e-mail pertence ao domínio da Universidade Federal do Ceará e envia um código de verificação. O usuário insere o código de verificação enviado. O sistema solicita as informações pessoais do cliente (nome, telefone, CPF, campus, sexo e senha). Com essas informações, o sistema cadastra o usuário.
Fluxo Alternativo
Se o e-mail ou o código de verificação estiverem inválidos, o sistema exibe uma mensagem de erro.
Recuperar senha (CU03)
Fluxo Principal
<ol style="list-style-type: none"> O sistema solicita o e-mail cadastrado do usuário. O sistema envia um código de verificação para esse e-mail para recuperar a senha. O cliente insere o código de verificação e a nova senha. O sistema redefine a senha e permite o login.
Fluxo Alternativo
Se o e-mail ou o código de verificação estiverem inválidos, o sistema exibe uma mensagem de erro.



Editar conta (CU04)	
Fluxo Principal	
<ol style="list-style-type: none"> 1. O usuário informa os dados que deseja modificar (exceto o e-mail institucional e CPF, que são imutáveis). 2. O sistema valida as informações fornecidas pelo usuário e atualiza as informações no perfil do usuário. 3. O sistema confirma que as alterações foram salvas com sucesso. 	
Fluxo Alternativo	
Caso as informações preenchidas forem inválidas, o sistema rejeita a edição e exibe uma mensagem de erro.	
Deletar conta (CU05)	
Fluxo Principal	
<ol style="list-style-type: none"> 1. O sistema disponibiliza a opção de deleção de conta para o usuário. 2. O usuário deseja realizar essa operação. 3. O sistema solicita uma confirmação do usuário para garantir que ele realmente deseja excluir a conta. 	

4. Após a confirmação, o sistema exclui a conta do usuário e exibe uma mensagem de sucesso, informando que a conta foi deletada com sucesso.

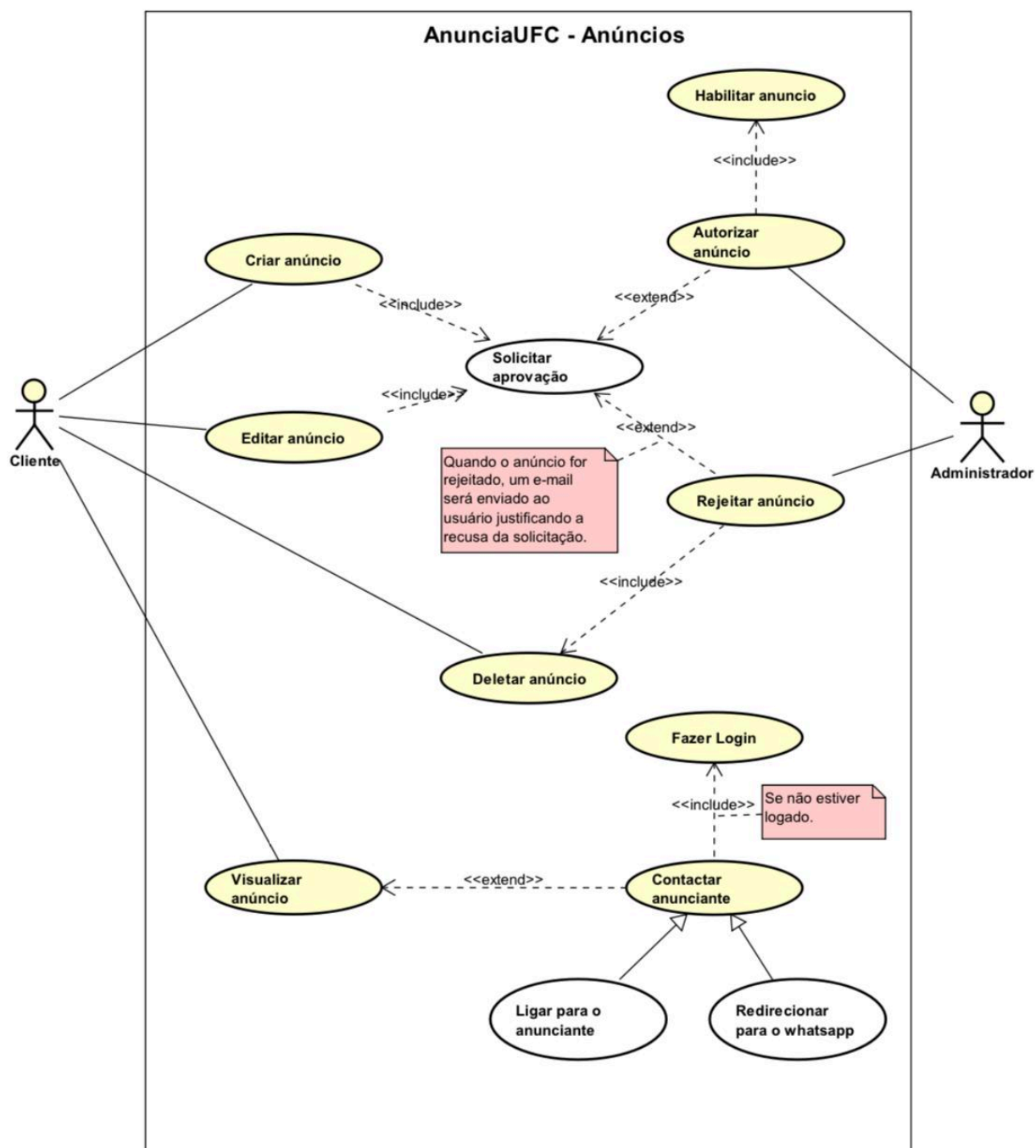
Redefinir senha (CU06)

Fluxo Principal

1. O sistema disponibiliza a opção de redefinição de senha para o usuário.
2. O usuário deseja realizar essa operação.
3. O sistema solicita ao usuário que forneça a senha atual e a nova senha.
4. O usuário insere a senha atual e a nova senha.
5. O sistema confirma que a senha foi redefinida com sucesso e exibe uma mensagem de sucesso.

Fluxo Alternativo

Caso a senha atual esteja incorreta, o sistema rejeita a alteração e exibe uma mensagem de erro.



Criar anúncio (CU07)
Fluxo Principal
<ol style="list-style-type: none"> 1. O sistema possibilita criar um anúncio e solicita ao cliente as informações necessárias para a criação do anúncio (como título, descrição, categoria, preço, fotos, etc.). 2. O usuário submete corretamente as informações.

3. O sistema envia o anúncio para a análise e exibe uma mensagem informando ao usuário que o anúncio será analisado pela equipe responsável para aprovar ou rejeitar a solicitação.

Editar anúncio (CU08)

Fluxo Principal

1. O sistema possibilita editar um anúncio e solicita ao cliente as informações necessárias para a criação anúncio (como título, descrição, categoria, preço, fotos, etc.).
2. O usuário submete corretamente as informações que deseja modificar.
3. O sistema solicita uma confirmação do usuário para garantir que ele realmente deseja alterar aquelas informações.
4. Após a confirmação, o anúncio é enviado para o sistema para a análise e exibe uma mensagem informando ao usuário que o anúncio será analisado pela equipe responsável para aprovar ou rejeitar a solicitação de edição.

Autorizar anúncio (CU09)

Fluxo Principal

1. O sistema exibe **apenas** para o administrador todos os anúncios que foram enviados para análise, aguardando aprovação.
2. O administrador visualiza cada anúncio e confere se as informações fornecidas são válidas e não impróprias.
3. Se o administrador verificar que todas as informações atendem aos requisitos, o anúncio pode ser autorizado.
4. O sistema atualiza o status do anúncio para "**Aprovado**" e o disponibiliza no sistema.

Rejeitar anúncio (CU10)

Fluxo Principal

1. O sistema exibe **apenas** para o administrador todos os anúncios que foram enviados para análise, aguardando aprovação.
2. O administrador visualiza cada anúncio e identifica os anúncios que contém alguma informação inadequada para serem rejeitados.
3. O sistema solicita ao administrador que forneça uma justificativa para a rejeição do anúncio e envia ao usuário um email com a justificativa.

4. O sistema remove o anúncio do banco de dados, impedindo que ele seja visualizado por outros usuários.

Deletar anúncio (CU11)

Fluxo Principal

1. O sistema possibilita ao cliente a opção de deletar um anúncio de sua autoria.
2. O usuário seleciona essa opção.
3. O sistema solicita uma confirmação do usuário para garantir que ele realmente deseja deletar o anúncio.
4. Após a confirmação, o sistema remove o anúncio do banco de dados e exibe uma mensagem informando que o anúncio foi deletado com sucesso.

Visualizar anúncio k(CU12)

Fluxo Principal

1. O cliente seleciona um anúncio disponível para visualizar seus detalhes.
2. O sistema exibe as informações do anúncio e as opções de contato **"Ligar para o anunciante"** e **"Enviar mensagem no WhatsApp"**.
3. O cliente deve estar logado para entrar em contato com o anunciante.
4. O cliente é redirecionado para a ligação ou para o WhatsApp, conforme a opção escolhida.

Fluxo Alternativo

1. Caso o cliente não esteja logado, o sistema o redireciona para entrar em sua conta.
2. Após autenticação bem-sucedida, o usuário retorna automaticamente à visualização do anúncio e pode prosseguir com o contato.

Ps: CU = Caso de Uso.

3.2. Diagrama de Classes

1) Classes

1. **Usuário:** Classe que define os atributos e funcionalidades comuns a todos os usuários do sistema.
2. **Cliente:** Representa os membros da comunidade acadêmica que utilizam o sistema. É uma subclasse de Usuário.
3. **Administrador:** Responsável por autorizar ou rejeitar anúncios solicitados pelos clientes. Também é uma subclasse de Usuário.
4. **Anúncio:** Classe que armazena as informações de um anúncio, além de oferecer funcionalidades relacionadas à sua publicação e edição.
5. **Imagem:** Representa as imagens associadas a um anúncio, armazenando informações como formato e URL.

2) Relacionamentos

1. **Cliente e Administrador → Usuário (Generalização/Especialização)**

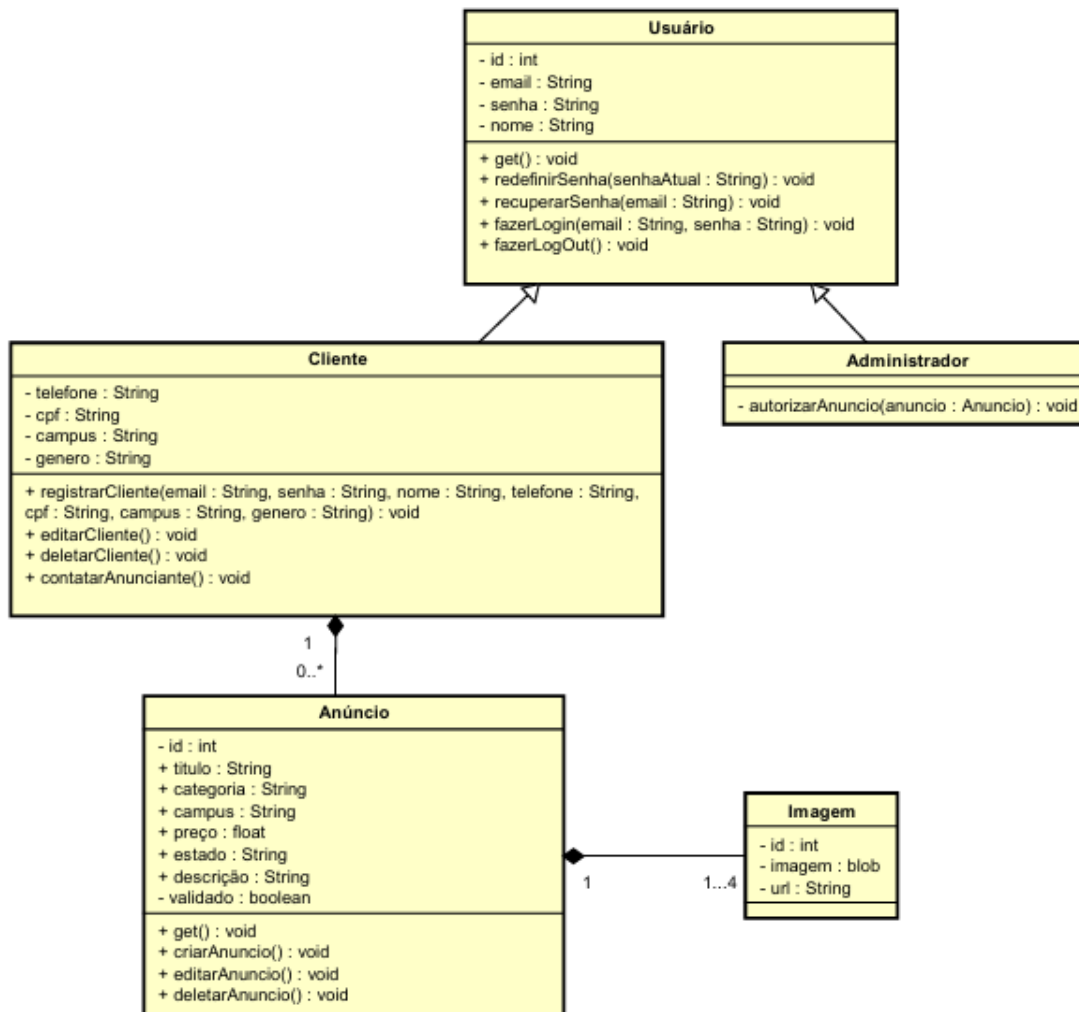
O Cliente e o Administrador herdam os atributos da classe usuário, pois ambos compartilham funcionalidades básicas, como login e cadastro.

2. **Cliente → Anúncio (Associação de composição - 0..*)**

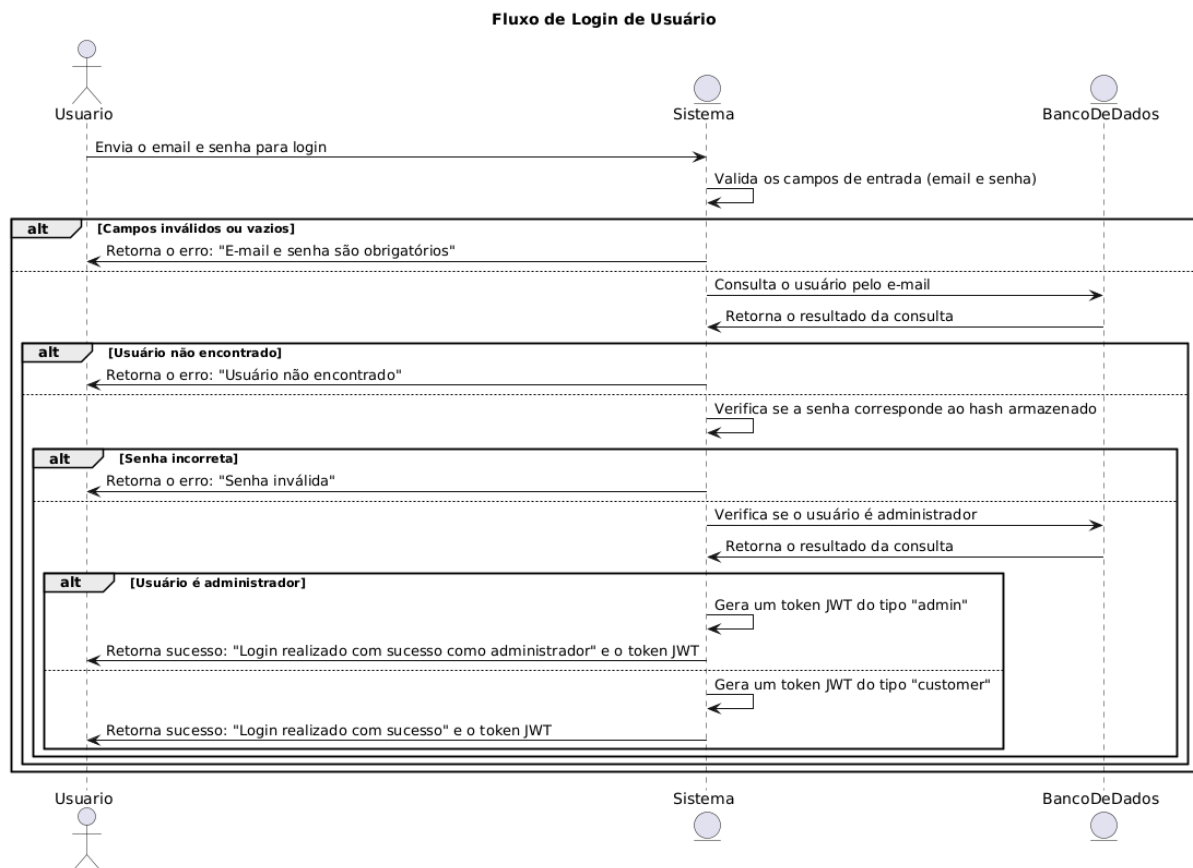
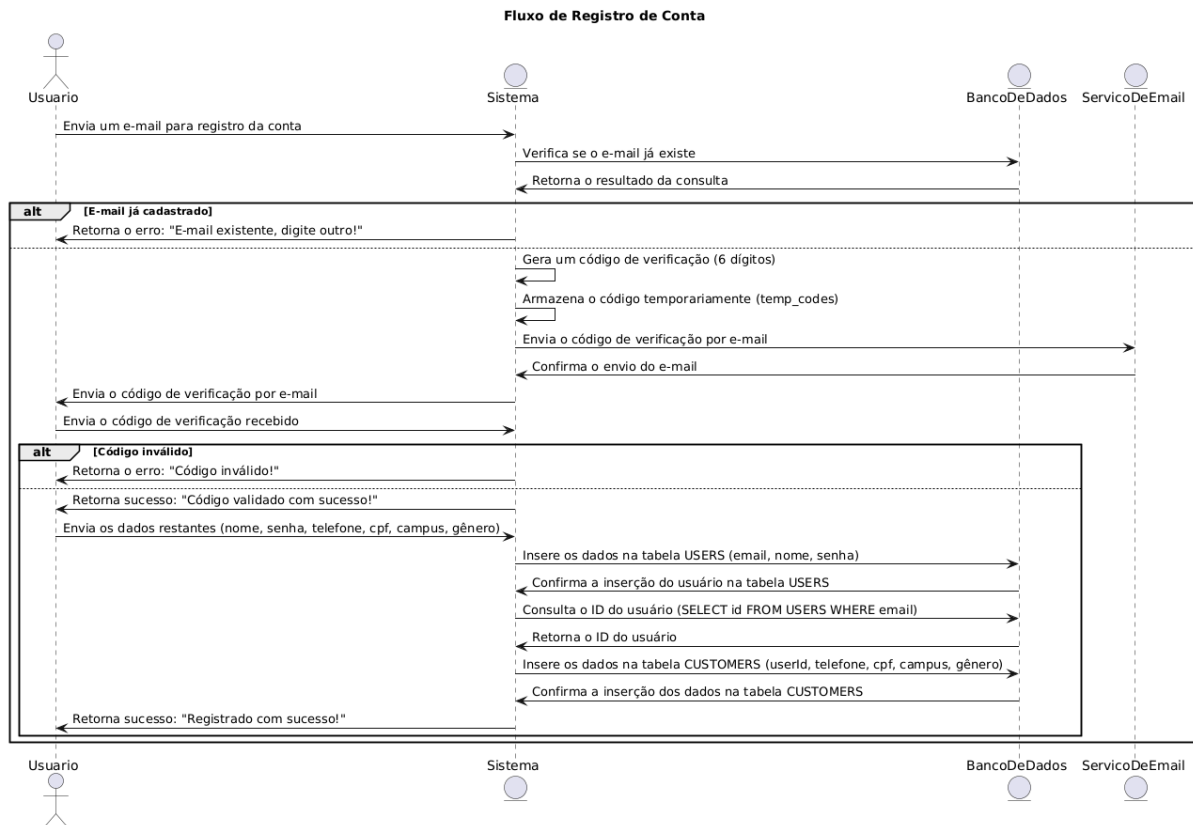
Um Cliente pode criar 0 ou muitos anúncios, mas cada anúncio pertence a apenas um cliente.

3. **Anúncio → Imagem (Associação de composição - 1..4)**

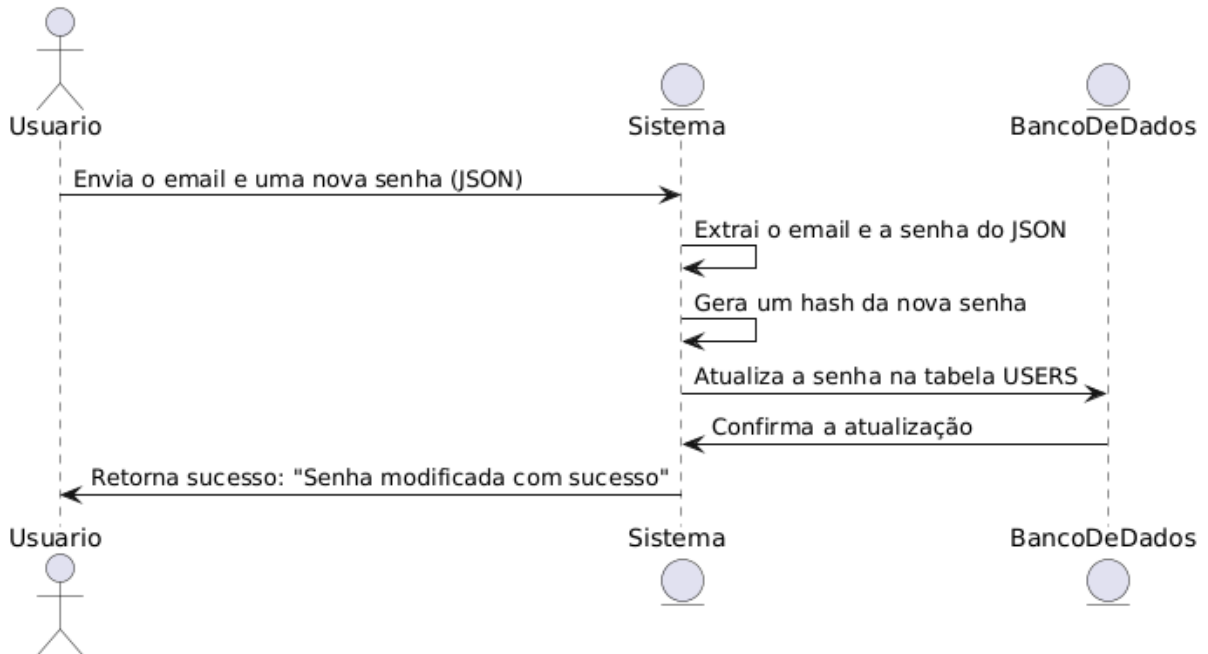
Um Anúncio deve conter pelo menos 1 e até no máximo 4 imagens, e cada imagem pertence a apenas um anúncio.



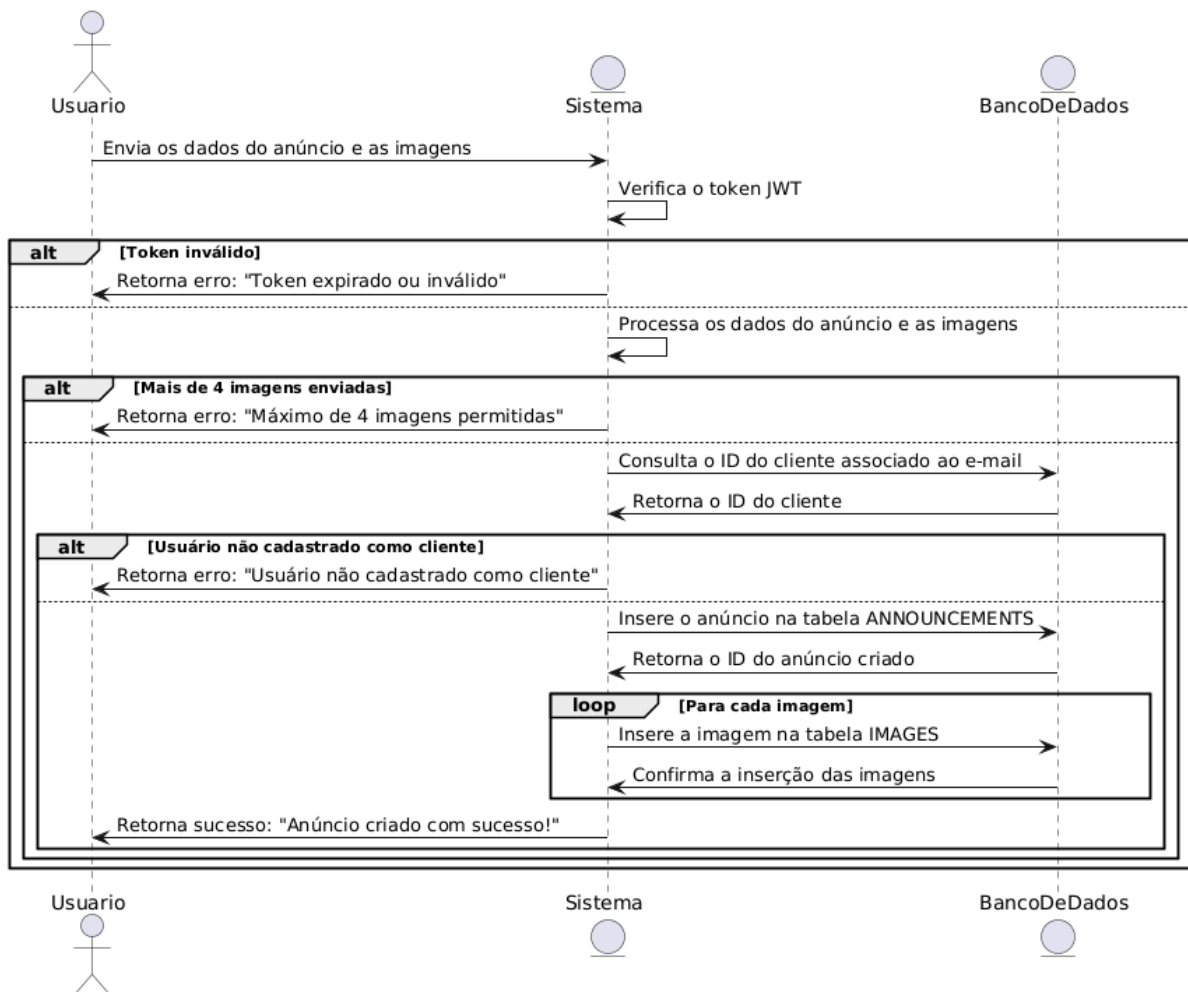
3.3. Diagramas de Sequência



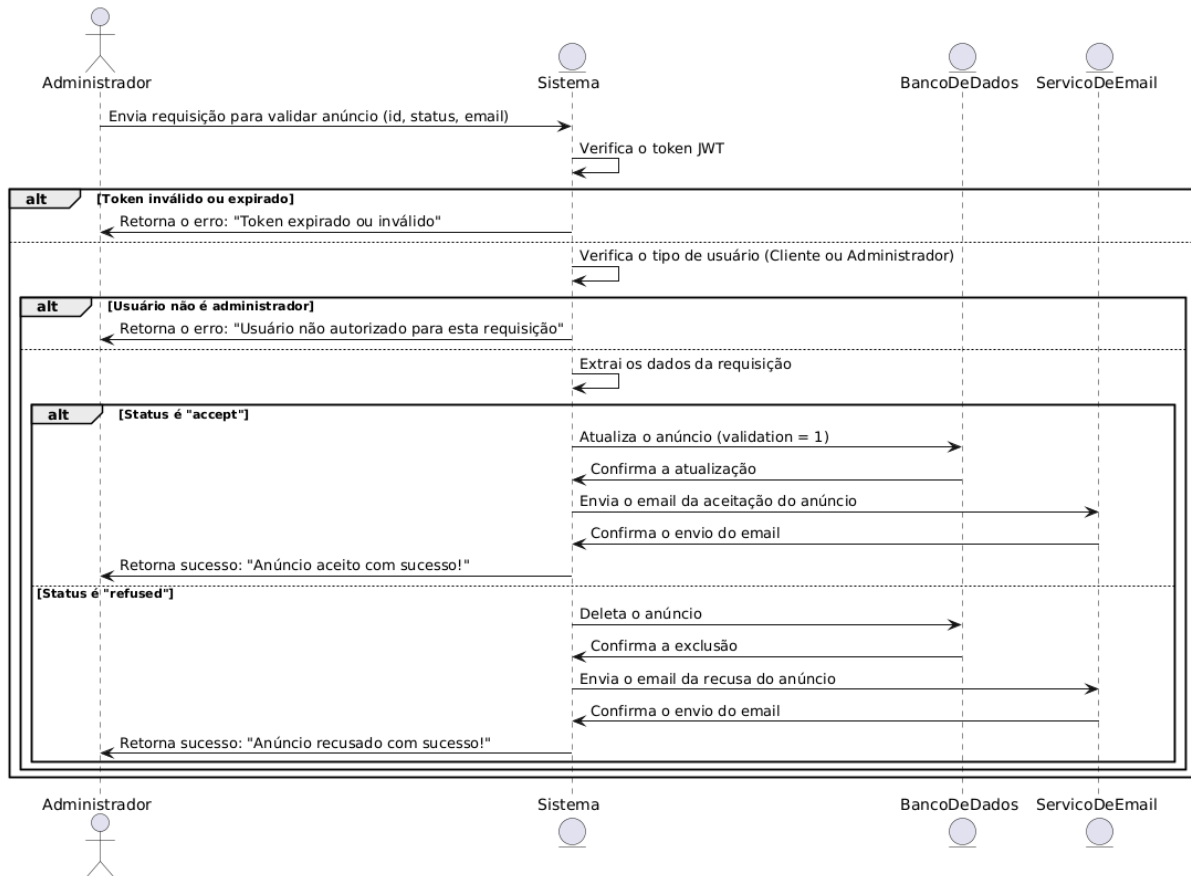
Fluxo de Recuperação de Senha



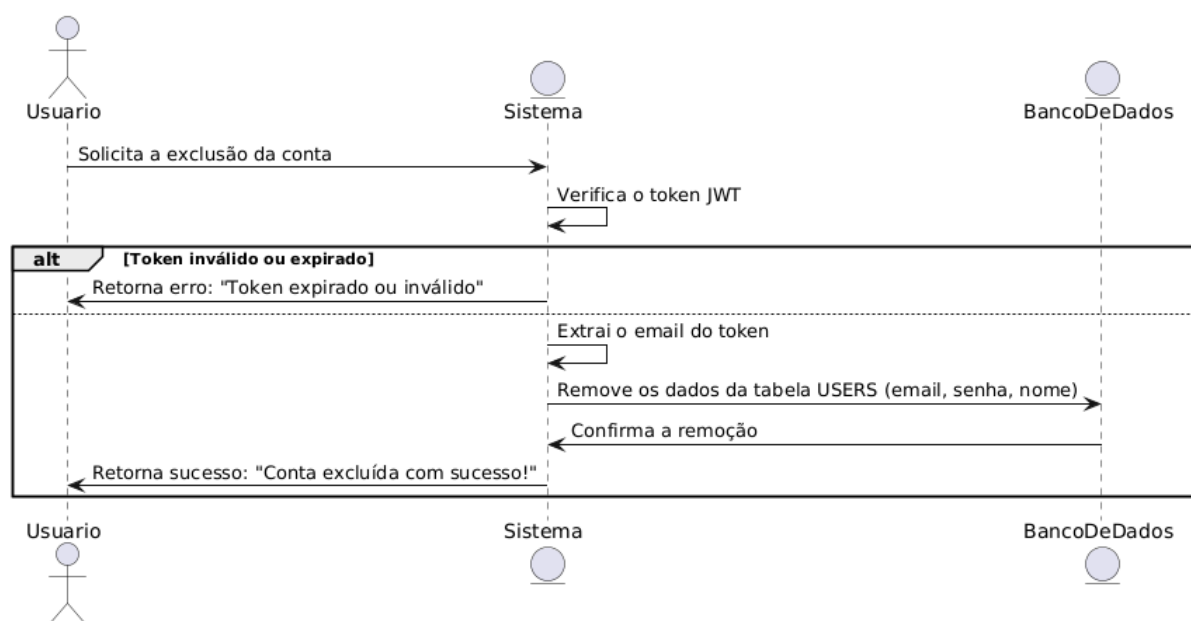
Fluxo de Criação de Anúncio



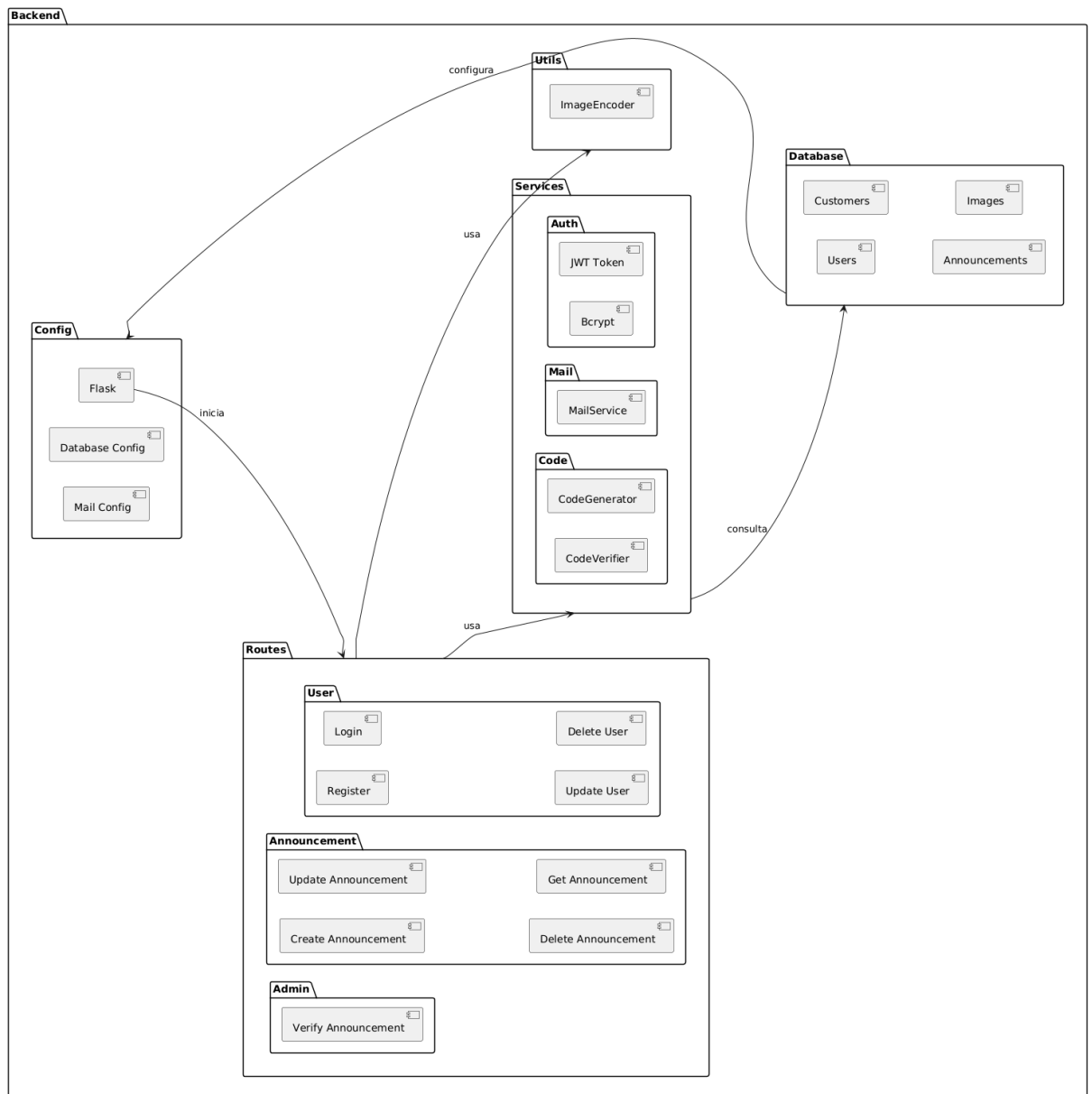
Fluxo de Validação de Anúncio pelo Administrador



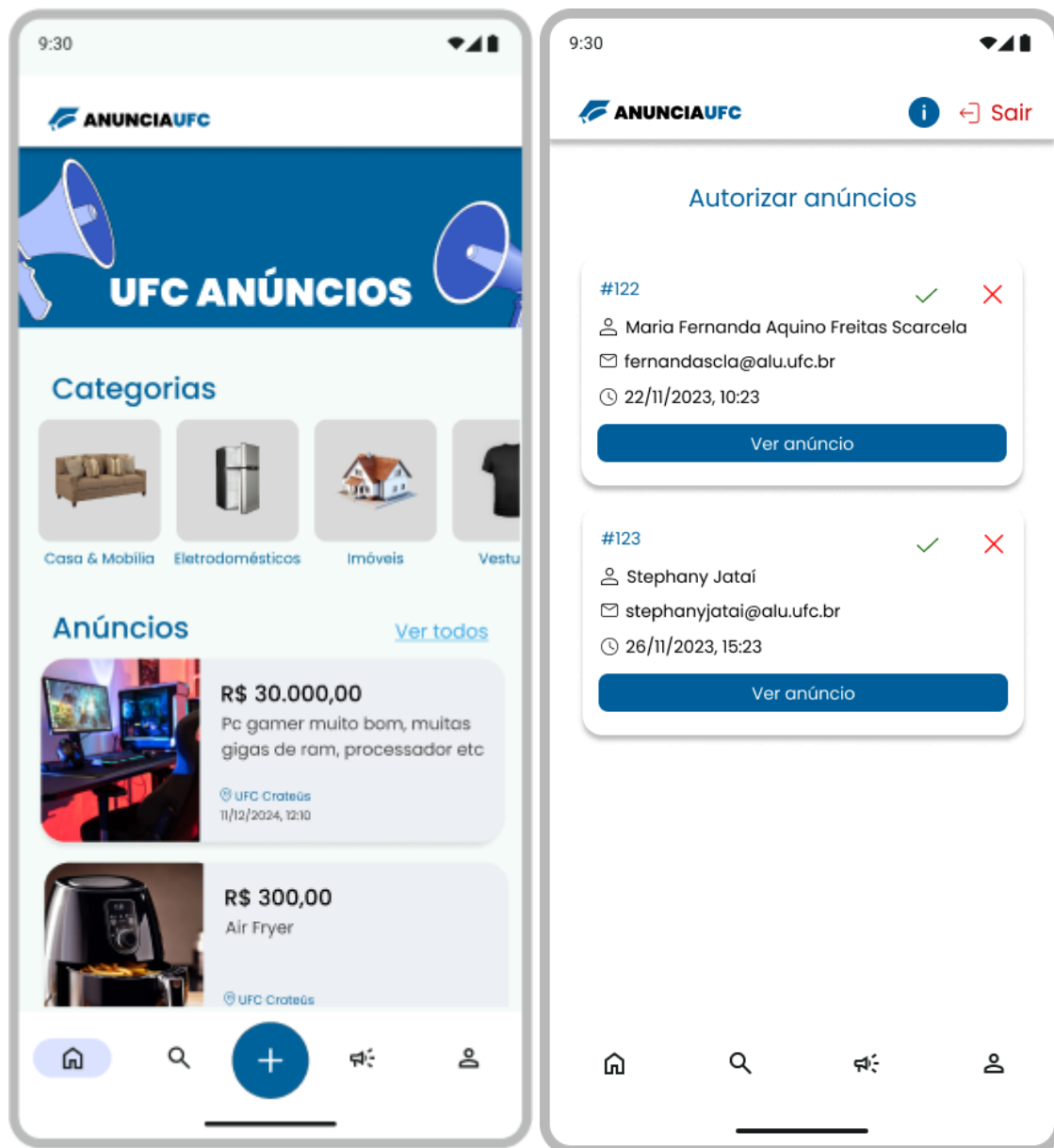
Fluxo de Exclusão de Conta



3.4. Diagramas de Pacotes



6. Protótipos de Interface de Usuário



← Casaco Rosa Muito Lindo



R\$15.50

Casaco Rosa confortável, usado poucas vezes, ideal para o clima frio das aulas do Wellington. Feito de fibra nobre de algodão, com um desing impecável semelhante aos da channel.

 Mandar uma mensagem

 Fazer uma ligação

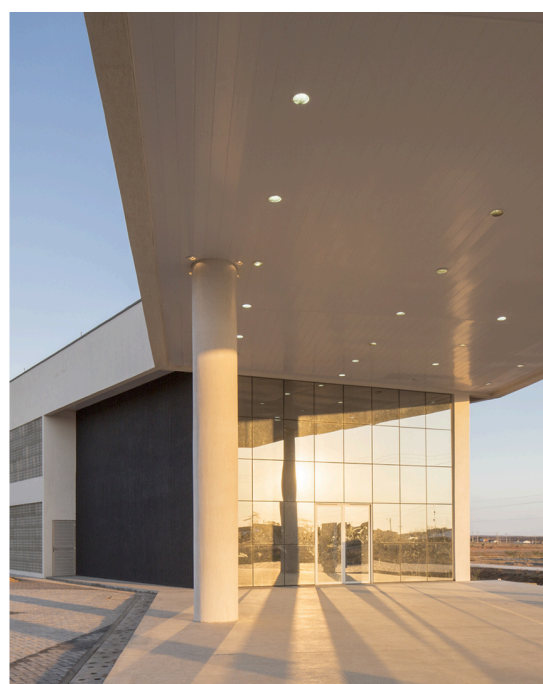


Cadastrar-se

Continue preenchendo suas informações para usufruir da nossa plataforma! Certifique-se que seu número de celular está correto, pois ele será utilizado para as negociações.

Nome e Sobrenome*

Número de Celular*



7. Considerações Finais

O desenvolvimento do AnunciaUFC superou expectativas técnicas, integrando teoria acadêmica e demandas práticas da comunidade universitária. A equipe enfrentou desafios complexos, desde a modelagem de diagramas até a integração de tecnologias heterogêneas, sempre buscando equilibrar robustez técnica e usabilidade. A aplicação de padrões de projeto foi essencial: o Singleton garantiu eficiência na gestão de conexões com o banco de dados, o Observer preparou o sistema para notificações em tempo real, e o Adapter permitiu futuras migrações para serviços em nuvem sem impactar o núcleo do sistema.

Os requisitos de domínio, como validação de e-mails institucionais e moderação de anúncios, foram tratados com rigor. O uso do Command para ações administrativas automatizou processos e reforçou a segurança, garantindo a publicação de conteúdo alinhado às diretrizes da UFC. A integração entre front-end (React com TypeScript) e back-end (Flask) demandou atenção meticulosa, com revisões iterativas de diagramas de sequência e testes contínuos para alinhar expectativas técnicas com a experiência do usuário final.

Para o futuro, o AnunciaUFC tem potencial expansivo. A inclusão de notificações por WhatsApp e SMS, via padrão Decorator, e estratégias de busca personalizadas, baseadas no padrão Strategy, poderiam melhorar a usabilidade e a relevância dos anúncios. A escalabilidade do código permite adaptações para outros campi da UFC, ampliando seu impacto. O sistema também pode evoluir com novos módulos, como um marketplace para estágios, e reforço de medidas de acessibilidade.

O sucesso técnico do projeto está intrinsecamente ligado à sua capacidade de resolver problemas concretos, como facilitar a busca por moradias, venda de materiais e moderação de conteúdo. O AnunciaUFC é uma prova de que a engenharia de software, aliada à empatia, pode transformar comunidades, tornando a vida acadêmica mais justa e colaborativa.