Universidade Federal de Alagoas
Curso de Ciência da Computação

Projeto de Engenharia de Software 1 (1ª Etapa)
Simples Repositório de Animes

Eduardo Vieira, Iago Conrado, Vinícius Barbosa

Arapiraca, 18 – Fevereiro

Índice

1. Introdução

O principal objetivo do sistema é armazenar informações sobre diversos animes e, usando de avaliações dos usuários gerar uma avaliação média sobre cada anime adicionado. Também pode ser útil agindo como uma lista pessoal para cada usuário. Sendo destinado a essa função simples de armazenar e organizar, o escopo do sistema, pelo menos inicialmente, não abrange funções mais complexas como recomendações e extração de sentimento de avaliações escritas(apesar delas estarem presentes). O sistema pode ajudar qualquer usuário interessado em manter uma lista pessoal e/ou avaliar ou ver avaliações de seus animes favoritos.

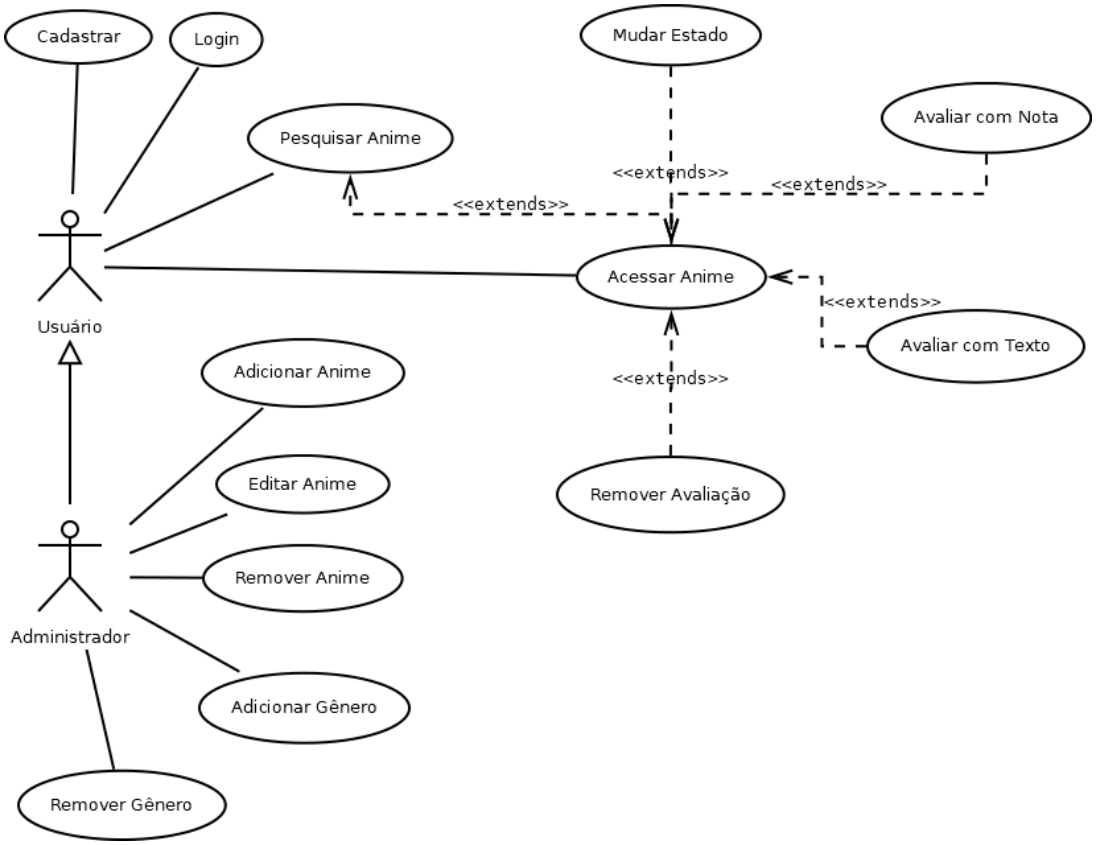
2. Requisitos

2.1. Elicitação dos requisitos

Como a equipe é bem familiarizada com o domínio do sistema, não foi necessário o uso de formas mais robustas de elicitação. Um simples e rápido brainstorm entre os membros foi suficiente para o levantamento inicial de requisitos. Tendo como referência sistemas já existentes (como o myanimelist.net), a equipe separou os requisitos mais cruciais para a solução do problema.

2.2. Requisitos funcionais

Diagrama 1 – casos de uso.



Caso de uso:	Acessar Anime
Ator(es):	Usuário, Administrador.
Pré-condições:	Usuário/Administrador logado, pelo menos um anime adicionado.
Fluxo padrão:	O Usuário/Administrador acessa algum anime. <<extend>> Mudar estado, podendo mudar o estado do mesmo entre: 1= “completo”, 2= “assistindo”, 3= “interessado” e 4= “sem interesse”. <<extend>> Avaliar com Nota/Texto, podendo avaliar o anime com uma nota (1 a 10) ou com texto, avaliações subsequentes sobrescreveram as antigas. <<extend>> Remover Avaliação, podendo remover alguma avaliação feita.
Fluxo de exceção 1:	Anime inexistente.
Pós-condições:	Algum anime foi acessado, podendo ou não ter novas informações.
Prioridade:	Alta.

Caso de Uso:	Acessar Anime
Fluxo de Exceção 1:	Anime inexistente.
Descrição:	Caso o Usuário/Administrador tente acessar um anime que não exite uma mensagem de erro será exibida e a operação não poderá ser feita.

Caso de uso:	Adicionar Anime
Ator:	Administrador.
Pré-condições:	Administrador logado.
Fluxo padrão:	O Administrador pode adicionar um anime, com seus atributos, ao repositório.
Fluxo de exceção 1:	Anime já existente.
Pós-condições:	Um anime foi adicionado.
Prioridade:	Alta.

Caso de Uso:	Adicionar Anime
Fluxo de Exceção 1:	Anime já existente.
Descrição:	Caso o Usuário/Administrador tente adicionar um anime que já exite uma mensagem de erro será exibida e a operação não será concluída.

2.3. Requisitos não funcionais

Usabilidade:

RNF1 –

Meta: Retornar animes com nomes parecidos com a busca caso a mesma não seja exata.

Modo de verificação: A partir de qualquer busca que não tenha correspondência exata no bando de dados, o sistema deve retornar os resultados mais próximos.

RNF2 –

Meta: Exibir nota média de cada anime.

Modo de verificação: A partir da lista de notas dadas, cada anime deve apresentar sua nota média (média aritmética).

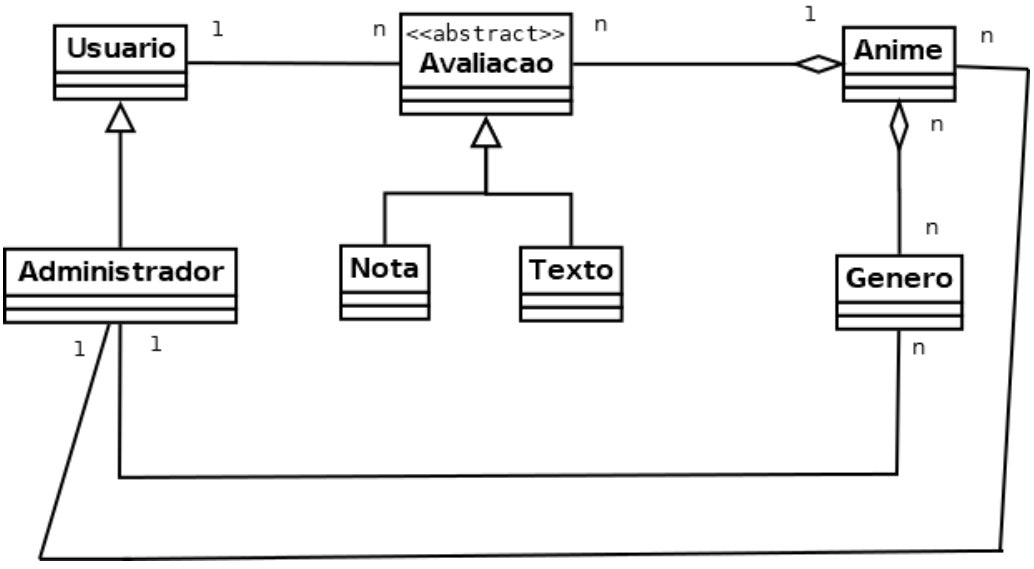
2.4. Validação dos requisitos

Após a elicitação inicial e durante a criação dos diagramas, a equipe naturalmente refinou os requisitos; adicionando, removendo ou editando. Até chegarmos em um consenso sobre o sistema e se seus requisitos realmente resolvem o problema. Devido ao tamanho do projeto, a validação não se mostrou um importante passo, pois após a própria elicitação a equipe já tinha uma boa ideia do que seria feito.

3. Projeto

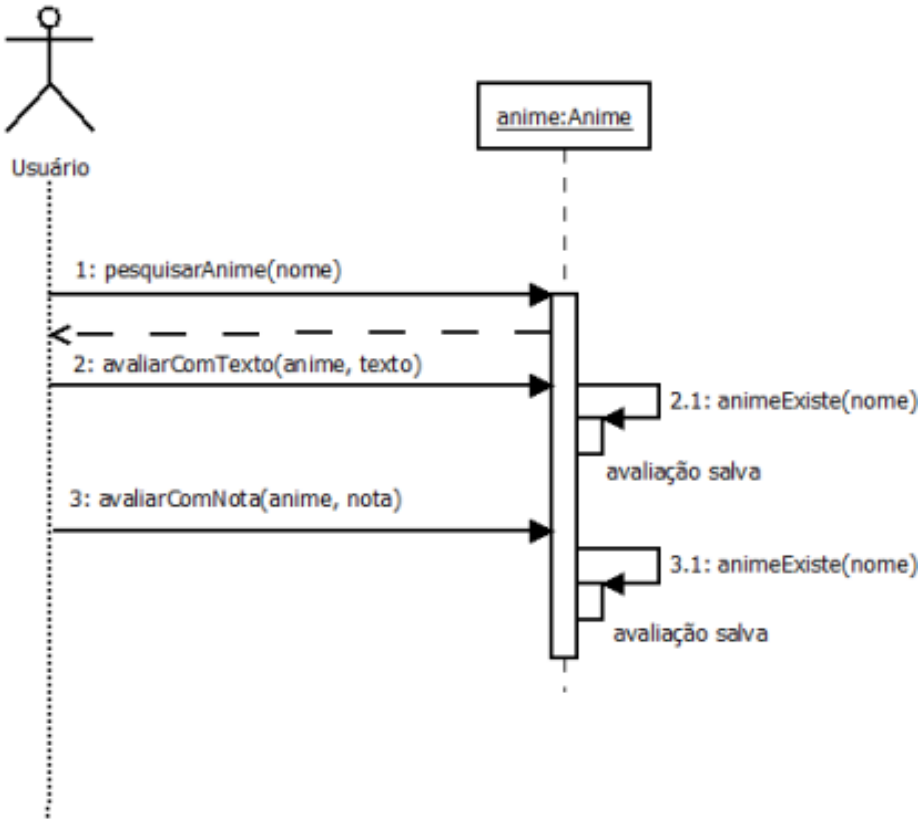
3.1. Projeto orientado a objetos

Diagrama 2 – classes.



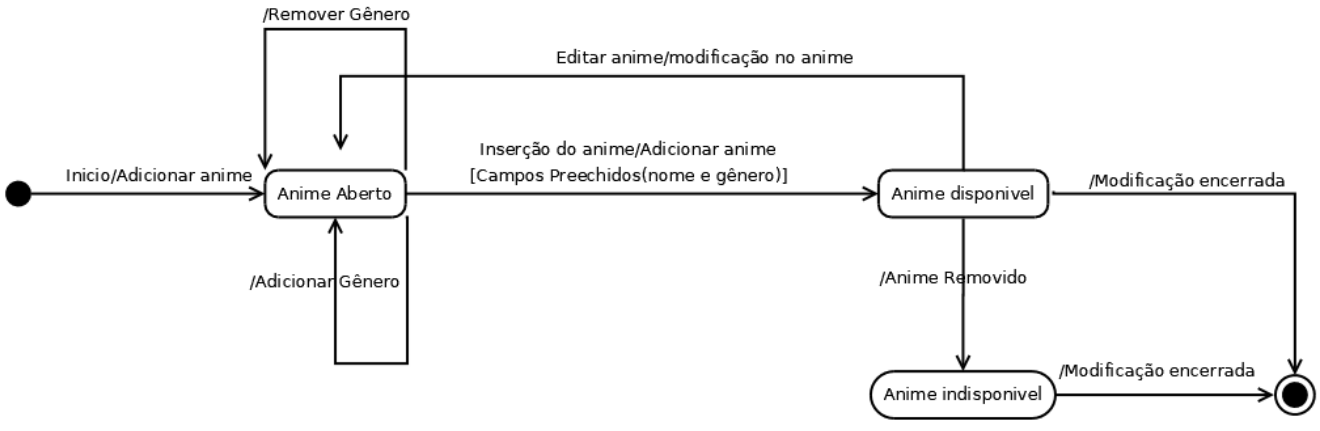
Descrição do diagrama 2 - classes: Diagrama base para as classes, com métodos e atributos abstraídos, seguindo o seguindo o determinado esquema, onde a classe “Administrador” herda atributos e métodos da classe “Usuario”, esta classe é responsável por criar objetos das classes “Anime” e “Genero”. Seguindo o esquema a interação entre “Usuario” e “Anime” acontece pelo relacionamento com a classe abstrata “Avaliacao”, “Nota” e “Texto” como subclasses da mesma, tendo o seguinte fluxo: Cada anime pode ter **n** avaliações e pode abranger **n** gêneros, cada usuário pode fazer várias avaliações, lembrando que cada anime só pode ter uma avaliação de cada tipo (Nota ou Texto) de cada usuário.

Diagrama 3 – sequência.



Descrição do diagrama 3 - sequência: O usuário pode buscar pelos animes existentes utilizando o método pesquisarAnime(), podendo atribuir a ele uma avaliação textual avaliarComTexto() ou uma avaliação com nota avaliarComNota(), desde que o anime esteja presente na lista de animes, logo após isso a avaliação será inserida e salva na lista de avaliações do anime.

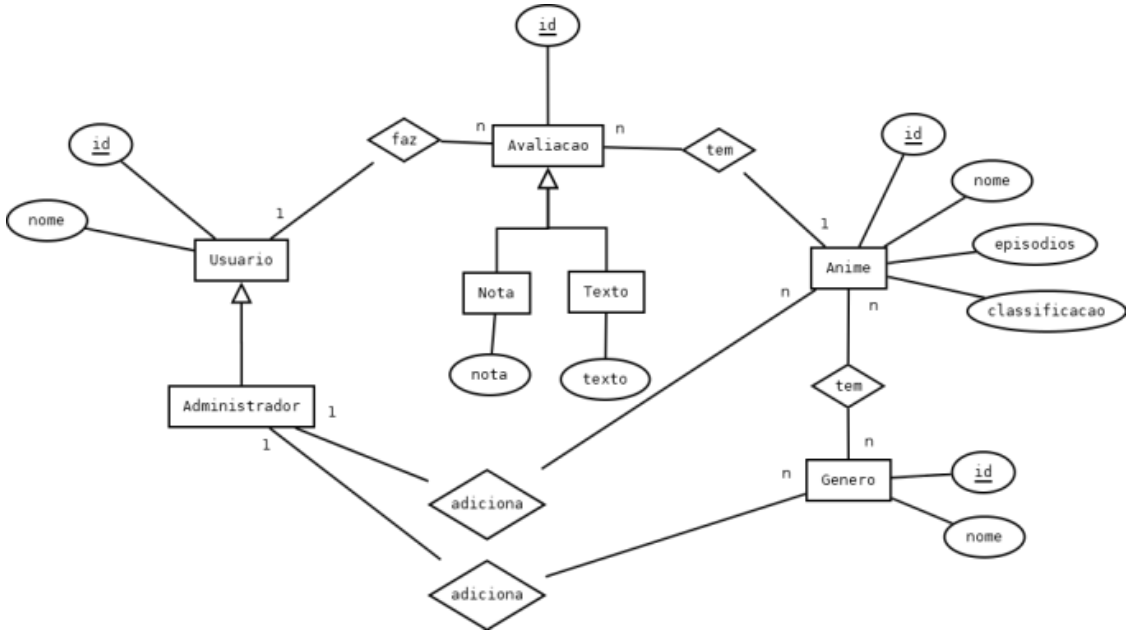
Diagrama 4 – estados.



Descrição do diagrama 4 - estados: Diagrama que segue os estados do objeto “anime” e seus estados na determinada inserção do mesmo. Fluxo: Administrador inicia a inserção do anime onde o mesmo fica em estado aberto até os campos “Nome” e “Gênero” serem preenchidos, como mostrado, o anime então segue para estado disponível, onde pode ser modificado ou removido. Modificações encerradas, fim de processo.

3.2 Projeto de banco de dados

Diagrama 5 – entidade-relacionamento.



Descrição do diagrama 5 – entidade-relacionamento: Diagrama que contem o relacionamento entre as entidades do projeto, seguindo o seguinte esquema: A entidade “Administrador” herda atributos da classe “Usuario”, a entidade por sua vez pode adicionar objetos das entidades “Anime” e “Genero”. A entidade “Usuario” pode se relacionar com as demais por meio da entidade “Avaliacao”, que é mãe de mais duas entidades “Nota” e “Texto”, assim um usuário pode fazer uma avaliação para um determinado anime.

4. Planejamento de entregas:

Interação 1 - versão base: Será entregue uma versão inicial com funcionalidades implementadas porém com riscos de instabilidade e bugs.

Interação 2: Correção de instabilidades e incrementos novos ainda não planejados podem ser adicionados.

Nota: (Novas interações podem ser adicionadas caso sejam necessárias).