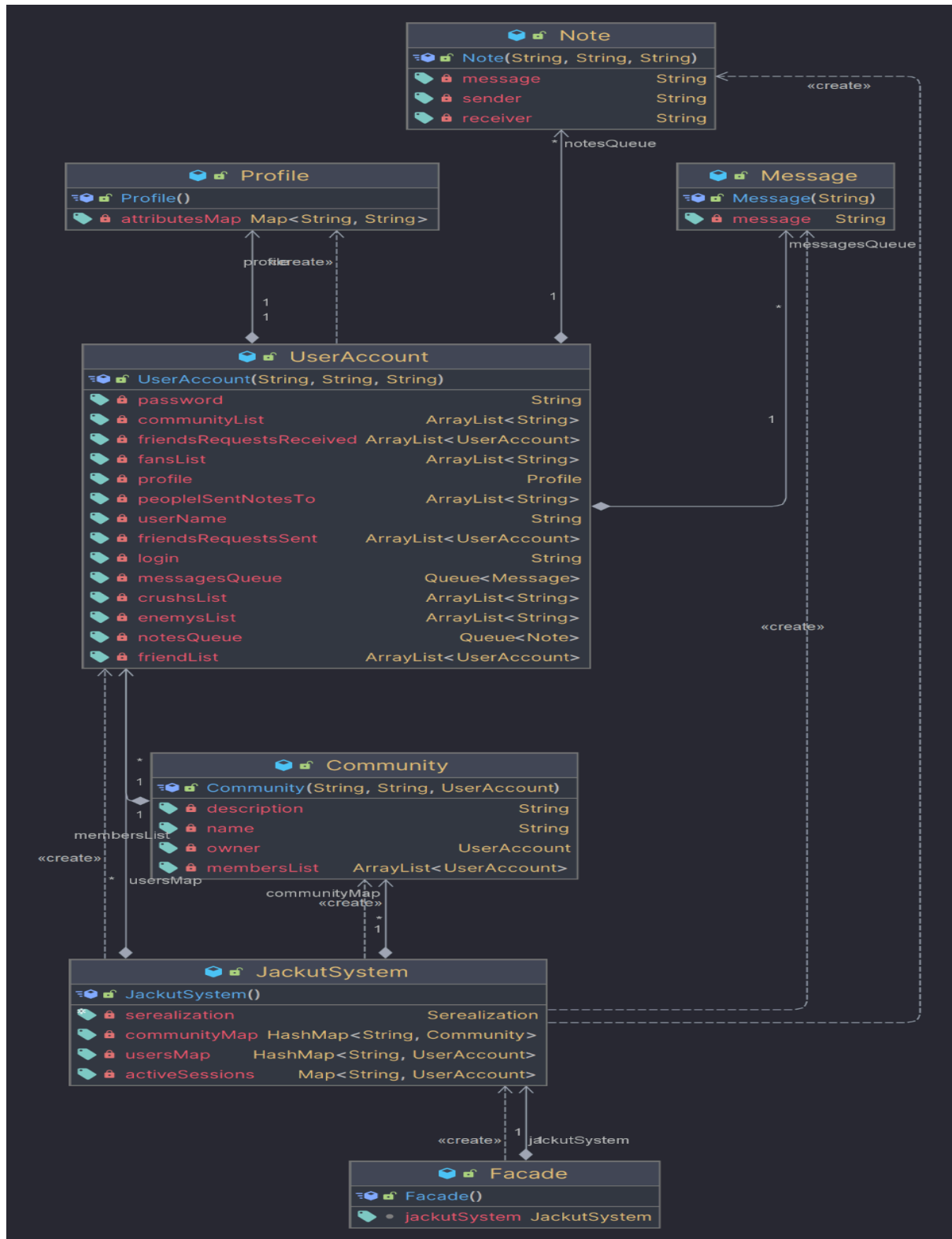


# Relatório - Jackut

Aluno: Jhordan Borges Almeida de Lacerda

Diagrama UML:



## Introdução:

Este relatório apresenta as modificações realizadas no sistema Jackut para o Milestone 2. A estrutura geral do projeto foi mantida, uma vez que não houve necessidade de incorporar novos padrões de projeto além do já utilizado *Facade*. Além disso, foram realizadas algumas refatorações no código, as quais serão detalhadas ao longo do documento.

## Novas classes criada:

As classes *Community* e *Message* foram criadas para satisfazer as exigências desse novo milestone. A classe *Community* foi criada no intuito de representar uma comunidade dentro do sistema, armazenando seu nome, descrição, criador e lista de membros. Já a classe *Message* foi criada para representar a troca de mensagens entre os membros participantes da comunidade, onde ela apenas armazena um conteúdo textual.

## Modificações e Melhorias nas classes existentes:

- *Serialização:*

A função de serialização foi modificada para serializar não apenas usuários (*UserAccount*), mas também as comunidades (*Community*). Isso garantiu que a minha persistência de dados realmente funcionasse.

- *UtilsString:*

A função que faz a formatação de *ArrayLists* para *Strings* foi generalizada para que pudesse receber listas de qualquer tipo de objeto, não só listas de strings.

- *UserAccount:*

Tivemos a adição da propriedade *enemyList* que é responsável por armazenar todos os inimigos do usuário e isso impede que outro usuário que foi adicionado a essa lista envie solicitação de amizade, adicione como paquera ou ídolo.

Também criamos a *Lista de destinatários de recados (note)* que é responsável por facilitar a remoção completa de dados ao excluir uma conta, garantindo que os recados enviados também sejam limpos dos destinatários.

Outras alterações foram a adição de alguns outros atributos, como: lista de fãs e paqueras e comunidades que o usuários faz parte e uma fila para armazenamento de mensagens recebidas através da comunidade.

## Regras de negócios Adicionadas:

- Relação de Inimizade:  
A relação de inimizade fez com que um usuário possa marcar outro usuário como inimigo, e com isso o usuário marcado como inimigo não pode adicionar como amigo, paquera ou tornar-se fã do usuário que o adicionou como inimigo e essa relação me fez alterar os métodos *addFriend* e *sendNote*, pois antes de enviar a solicitação de amizade, o sistema tem que verificar se essa relação inimizade existe, e caso exista ele não realiza o envio da solicitação e o mesmo se aplica para o método *sendNote*.
- Exclusão de conta:  
Agora, ao excluir um *UserAccount*, o sistema percorre a lista de destinatários dos recados enviados e remove as mensagens associadas. Isso previne referências "fantasmas" a usuários excluídos e melhora a integridade do sistema.