Engenharia de Software: Calculadora em 3 camadas - V4

Breno Keller

Universidade Federal de Ouro Preto

kellerbrenons@gmail.com

Introdução

- Trabalho prático para exemplificar o desenvolvimento de uma aplicação Cliente/Servidor utilizando APIs do QT;
- Objetivo: Evoluir a arquitetura do projeto;
- Código fonte disponível em: `https://bitbucket.org/KellerBreno/calculadora/`
  - Versão 4 - Uso de Patterns.

Controle de Acesso

Existem diversos patterns para controle de acesso:

- ACL - Access Control List;
- Actor-Role;
- RSBAC - User Set Based Access Control.

Actor-Role

- No mundo real, pessoas não confundem atores com papéis;
  - A não ser no Brasil.

Actor-Role

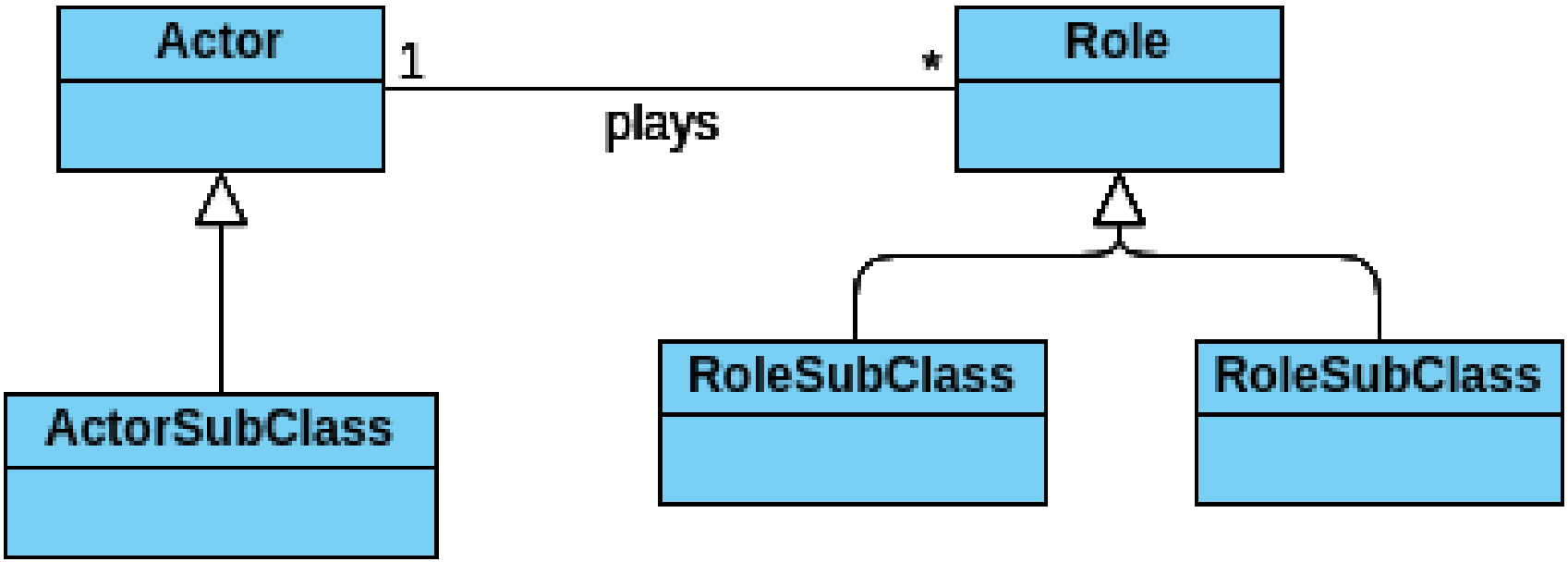

Figura: Actor-Role

Actor-Role

- Desacoplar atores e papéis;
- Permitir mudança de comportamento em tempo de execução;
- Variação de comportamento não é fixada pela hierarquia das classes;
- Também pode ser visto como parte do pattern Agent-Behaviour.

Actor-Role Implementação

Variações de Implementação:

- Como representar diversos papéis;
- Como representar generalização;
- Como lidar com tipo dinâmico de um objeto.

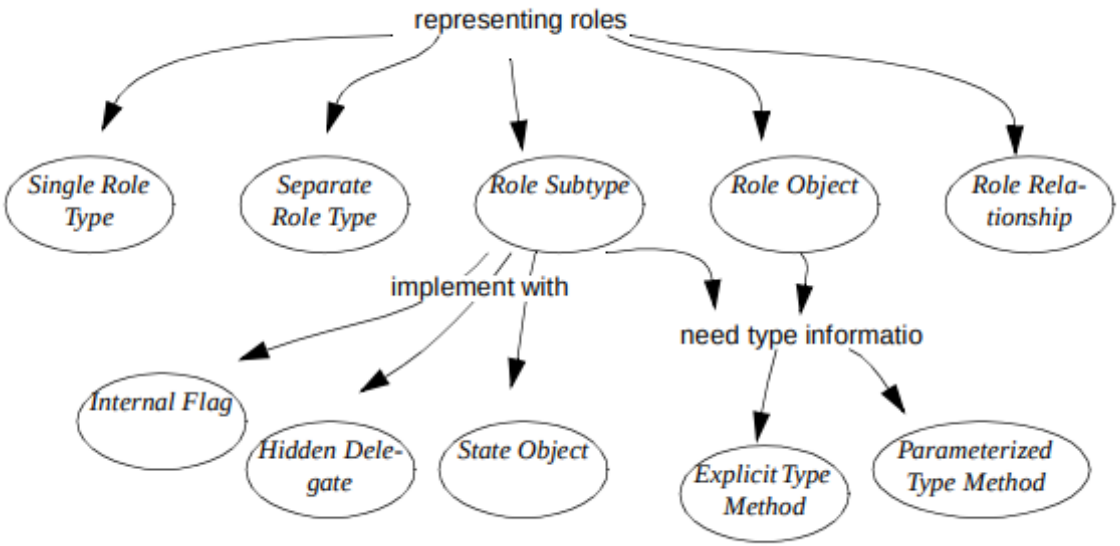

Figura: Fonte: FOWLER, Martin. Dealing with roles. In: Proceedings of PLoP. 1997.

Role Subtype

Uma classe para cada papel e comportamento comum na superclasse.
Vantagens:

- Conceitualmente simples;
- Interfaces simples.

Desvantagens:

- Cada novo papel gera alterações na superclasse.

Role Subtype

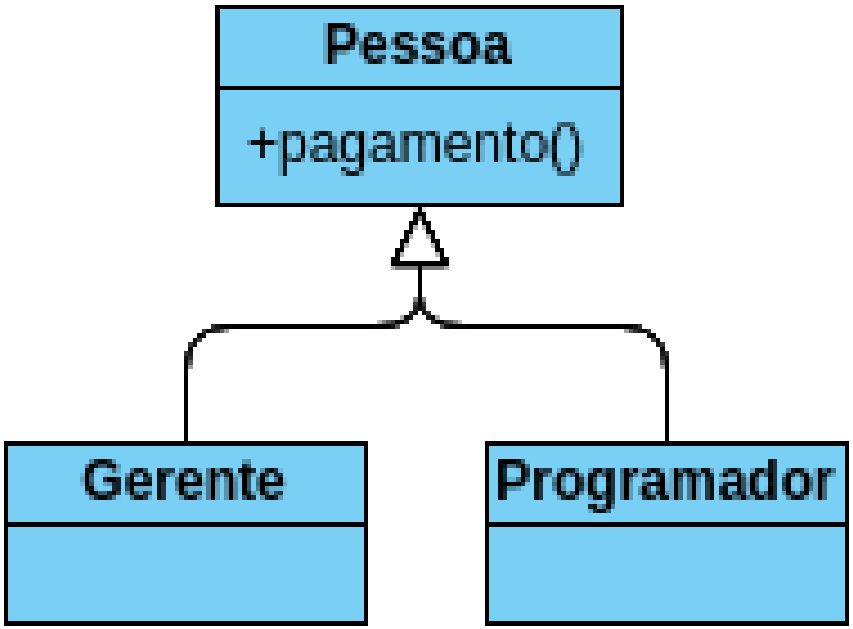

Figura: Role Subtype

Hidden Delegate

Coloque as variações em uma classe privada e delega a ela a resolução das requisições.
Vantagens:

- Suporta múltiplas classificações dinâmicas;
- Comportamento está definido na delegação.

Desvantagens:

- Seleção de método é feita pela classe pública.

Hidden Delegate

```
class Pessoa{
        virtual double pagamento() = 0;
}

class Gerente: public Pessoa{
        Actor *actor;

        double pagamento() override{
                return actor->pagamento();
        }
}
```

Arquitetura

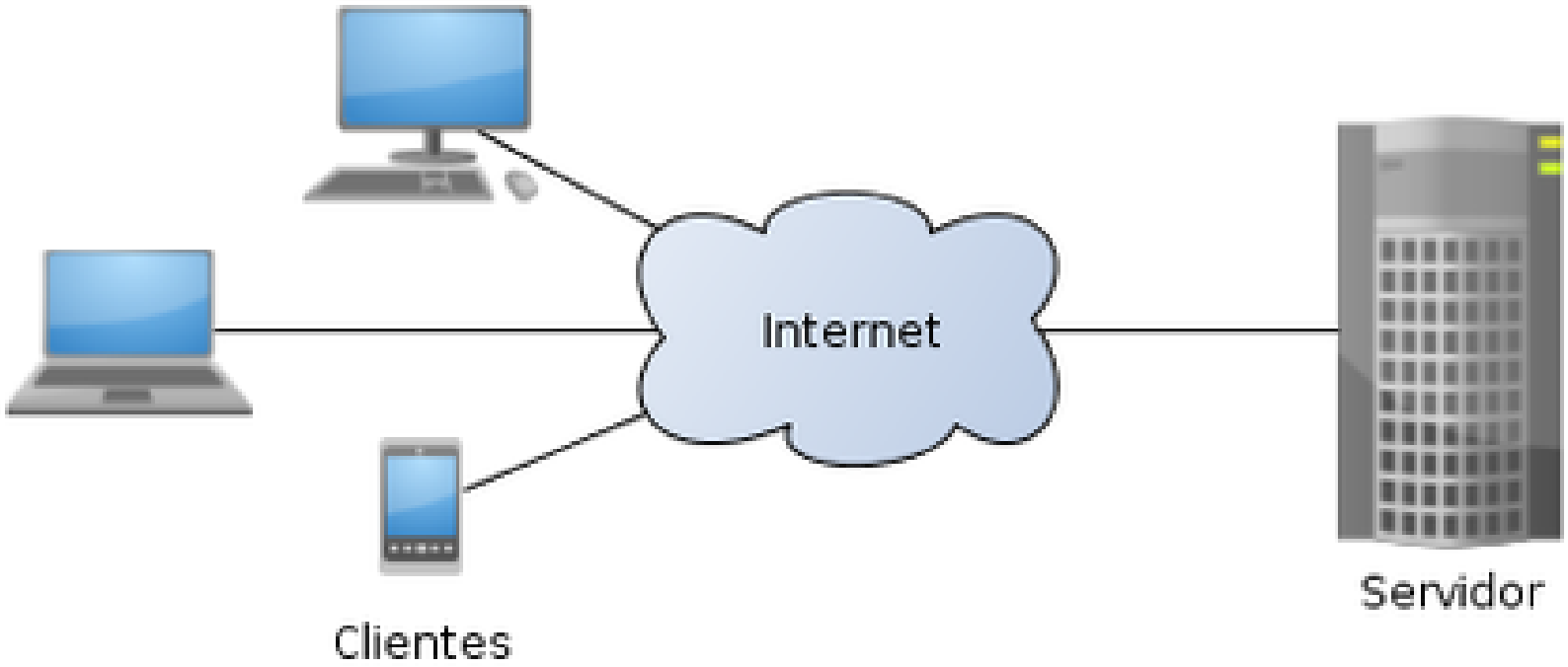

Figura: Arquitetura da Aplicação

Componente: Client em V3a

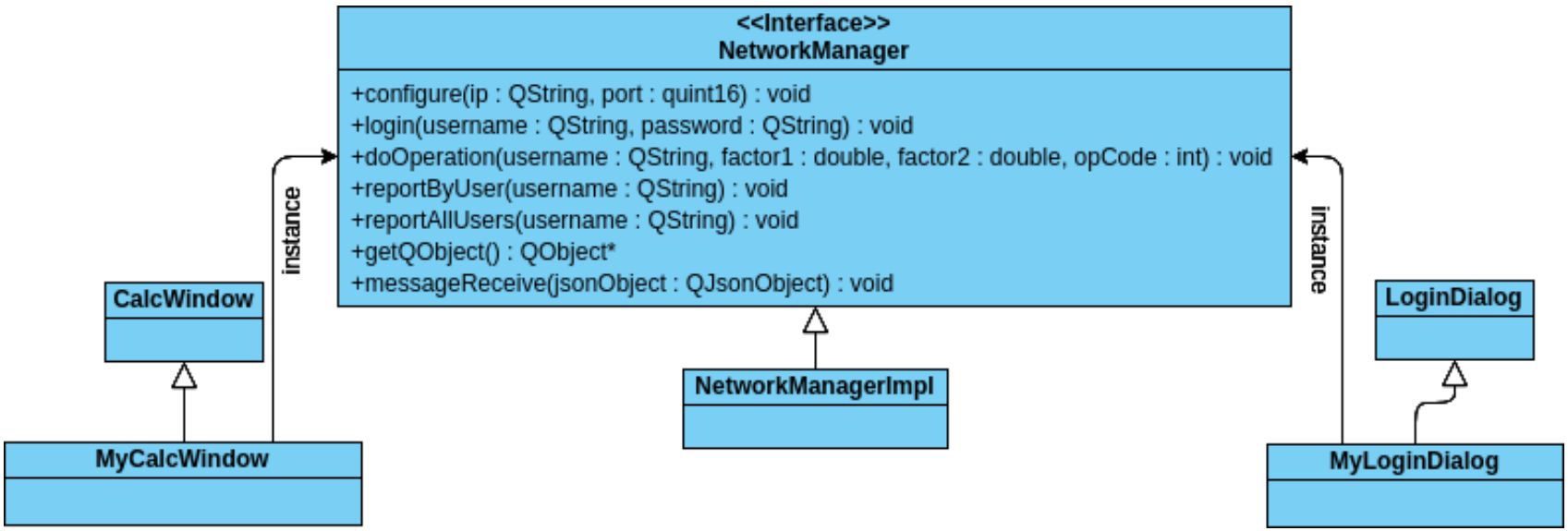

Figura: Diagrama de Classes: Client em V3a

Componente: Client em V4

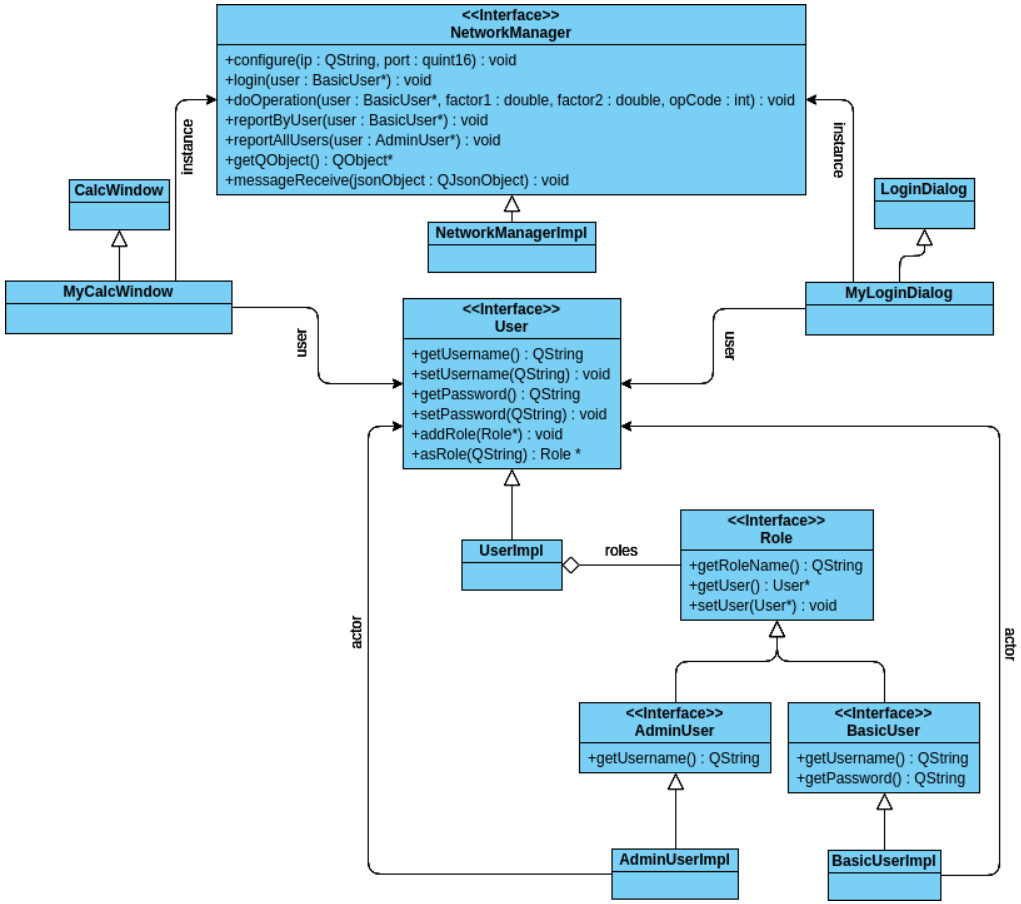

Figura: Diagrama de Classes: Client em V4

Componente: Client em V4

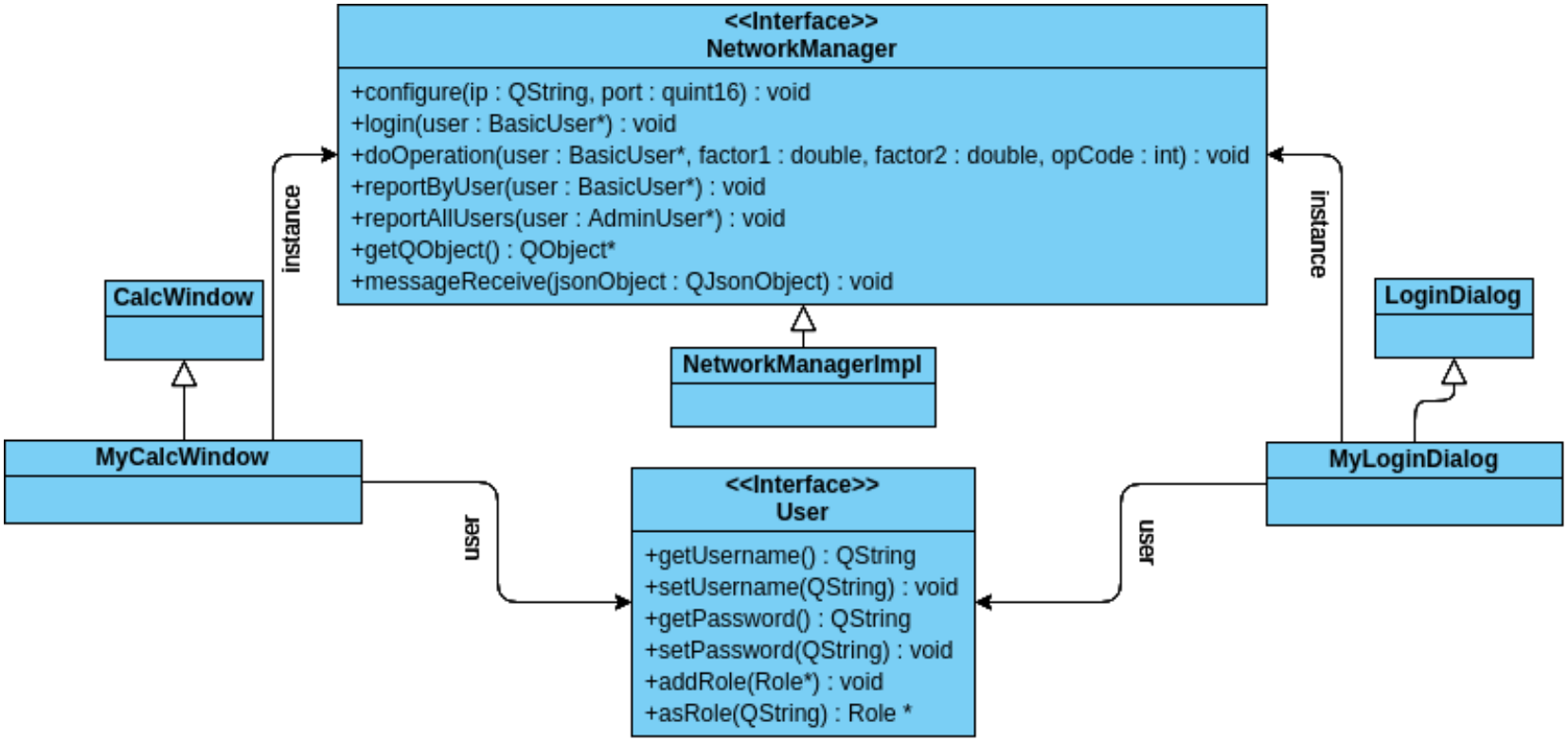

Figura: Diagrama de Classes: User

Componente: Client em V4

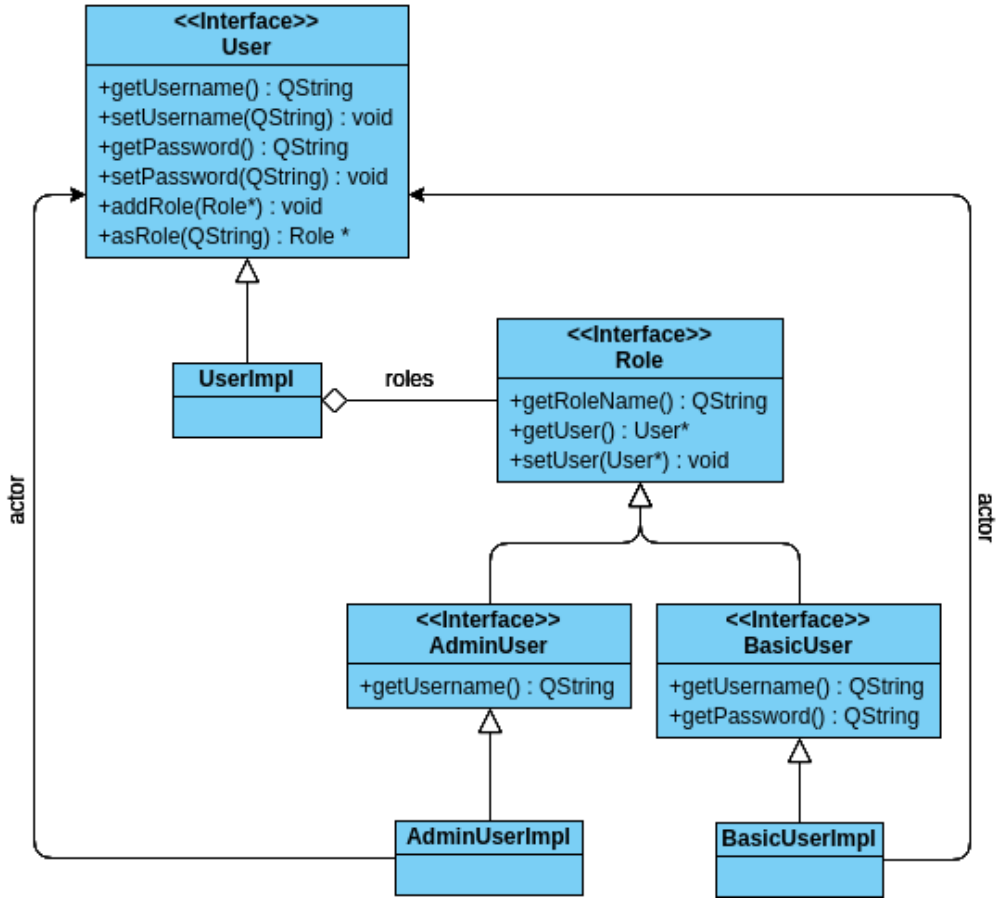

Figura: Diagrama de Classes: User Role

Componente: Server em V3a

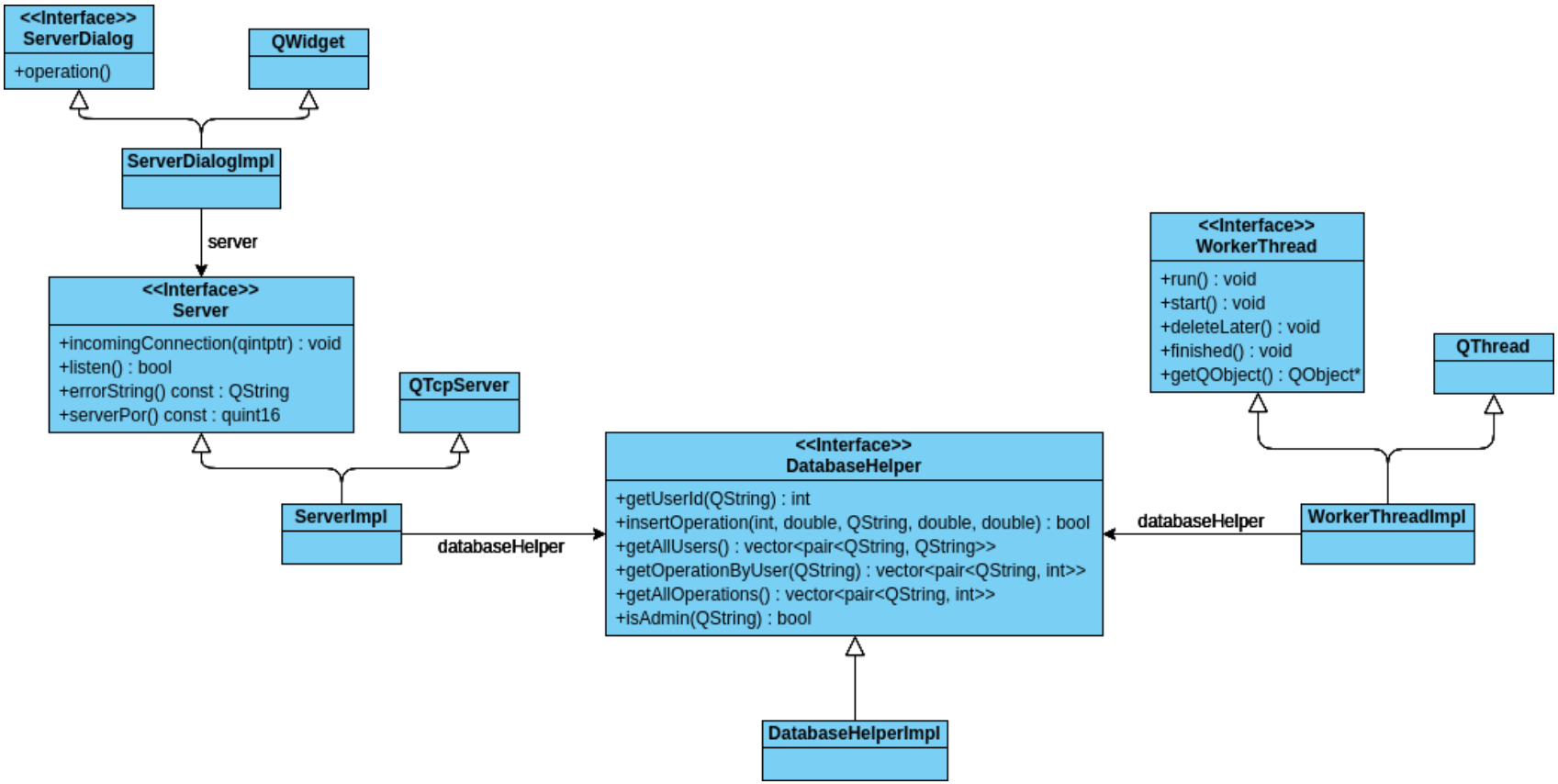

Figura: Diagrama de Classe

Diagrama de Sequência em V3a

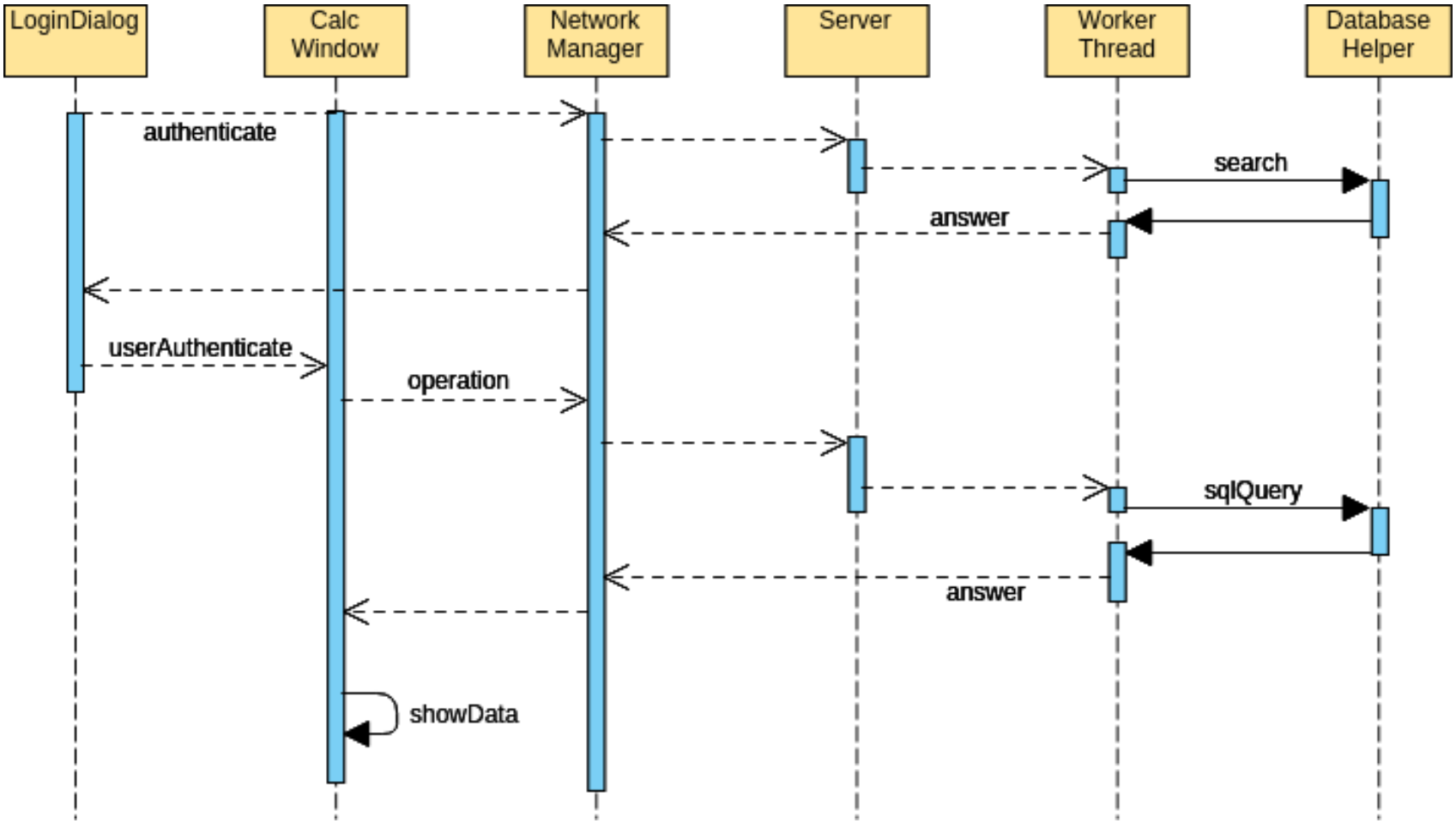

Figura: Diagrama de Sequência

Diagrama de Sequência em V4

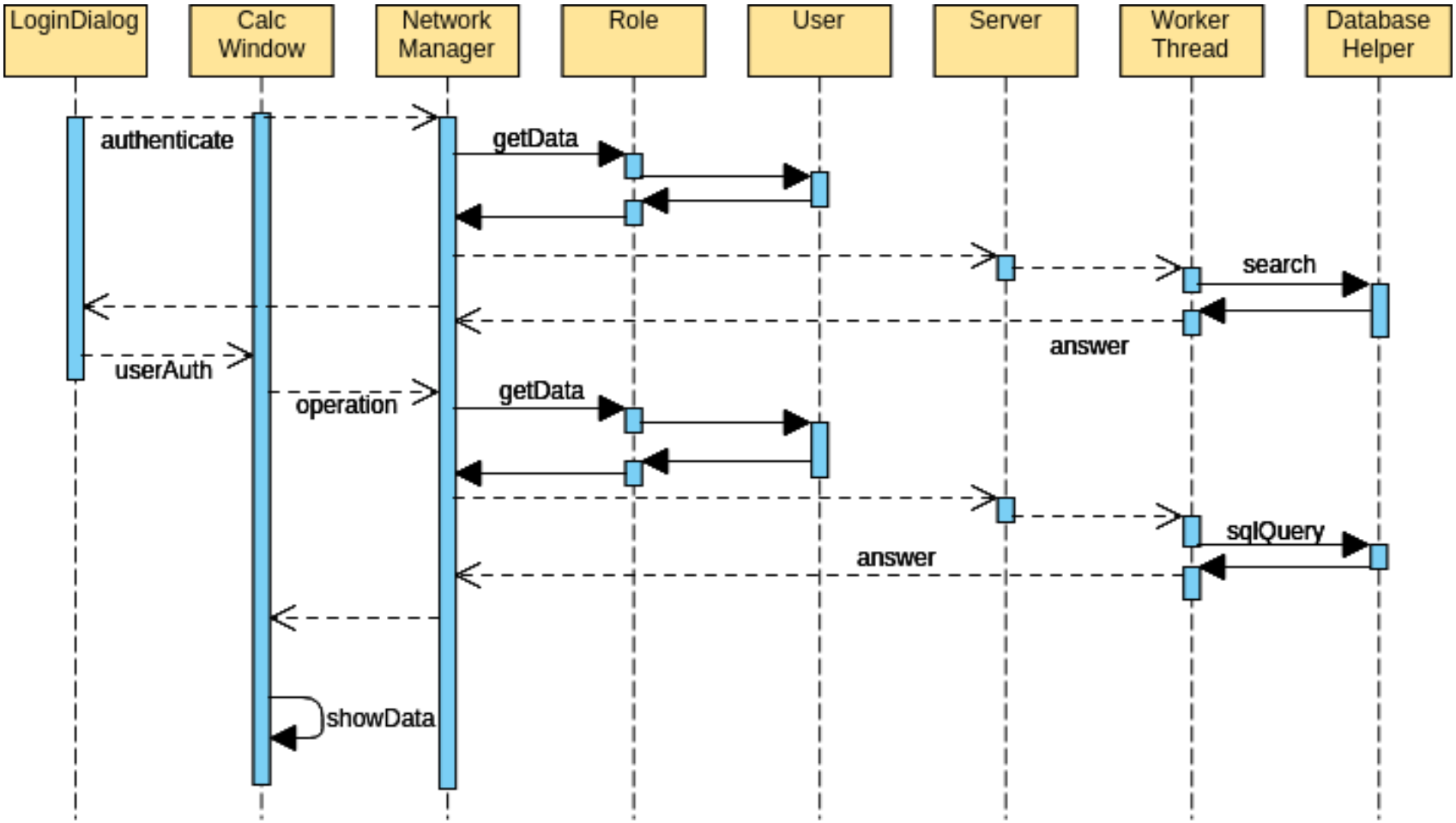

Figura: Diagrama de Sequência

Fim