

# Engenharia de Software: Calculadora em 3 camadas - V5

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/>
  - V5 - Outros Patterns.

# Active Record × Data Mapper

- Patterns arquiteturais;
- Utilizados para gerenciar a comunicação entre lógica de negócios e banco de dados.

# Active Record

Cada tipo de dados conhece como armazená-lo. De forma que cada instância corresponda a uma linha da tabela no banco de dados.

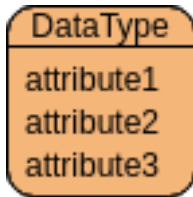
Vantagens:

- Simplicidade;
- Útil para CRUDs.

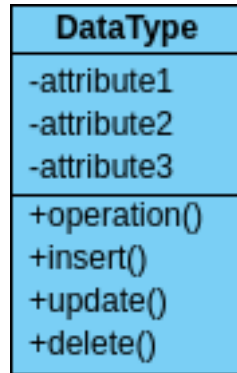
Desvantagens:

- Mudar o SGBD é uma tarefa árdua;
- Fere o princípio de responsabilidade;
- Inviável para concorrência.

# Active Record



(a) ER



(b) Active Record

# Data Mapper

Utilização de uma camada de separação entre a lógica de negócios e o banco de dados.

Vantagens:

- Mudança simples entre SGBDs;
- Respeita o princípio de responsabilidade.

Desvantagens:

- Complexidade.

# Data Mapper

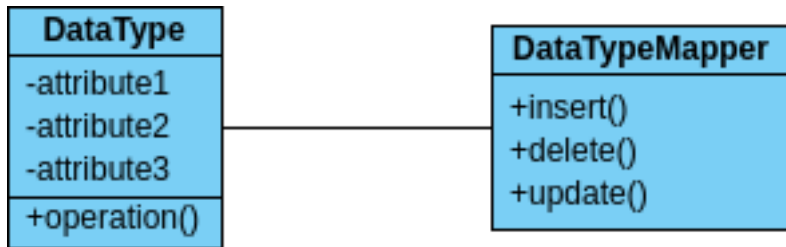
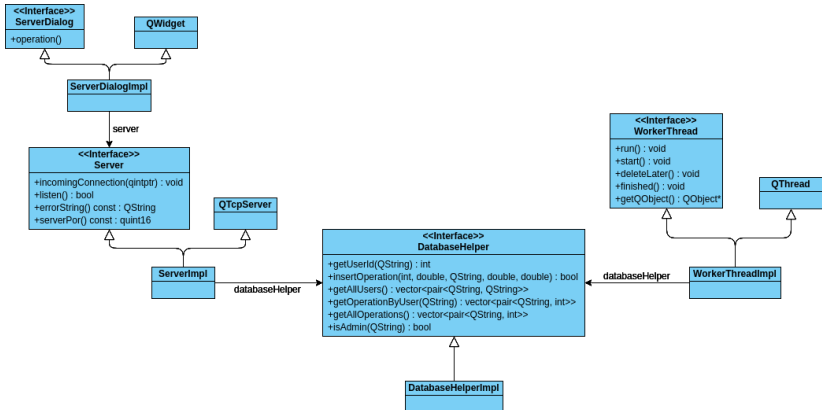


Figura: Data Mapper

# Calculadora - Server





# Database Helper

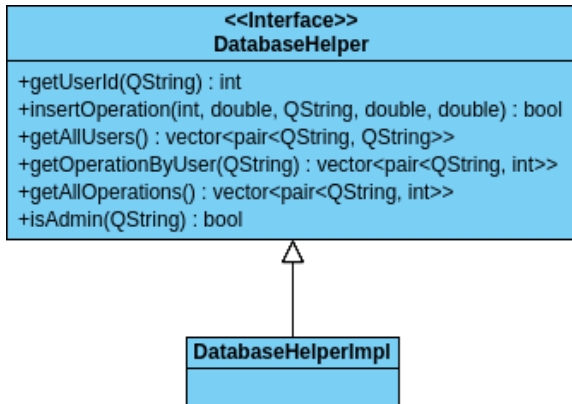


Figura: Database Helper

# Pools

- Gerenciamento eficiente de memória;
- Ganho de desempenho.

# Object Pool

Utiliza um conjunto de objetos pré-instanciados, para evitar o custo de alocação e remoção de novos objetos.

Vantagens:

- Ganho de desempenho;

Desvantagens:

- Complexidade no tempo de vida do objeto;

# Object Pool

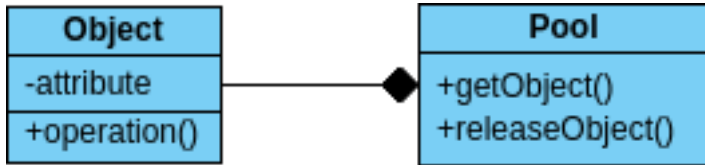


Figura: Object Pool

# Thread Pool

Caso "específico" de Object Pool. Gerencia a execução de tarefas associado a um conjunto de threads disponíveis. Usualmente a quantidade de threads é um múltiplo da quantidade de processadores.

Vantagens:

- Redução do custo de troca de contexto;
- Ajuste em tempo de execução.

Desvantagens:

- Tempo de execução (fila);

# Thread Pool

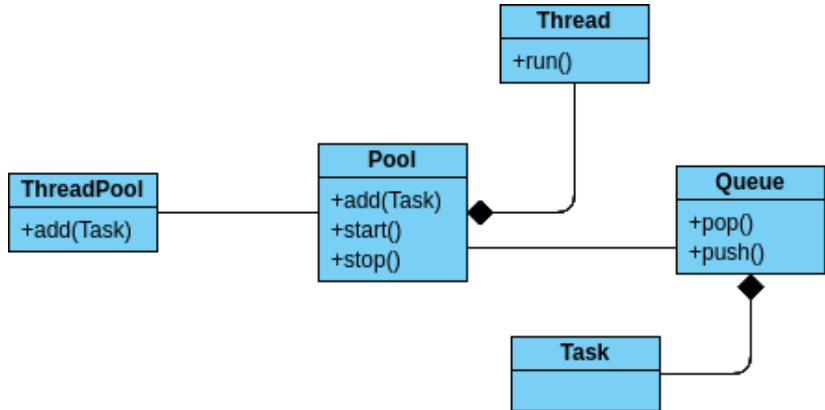
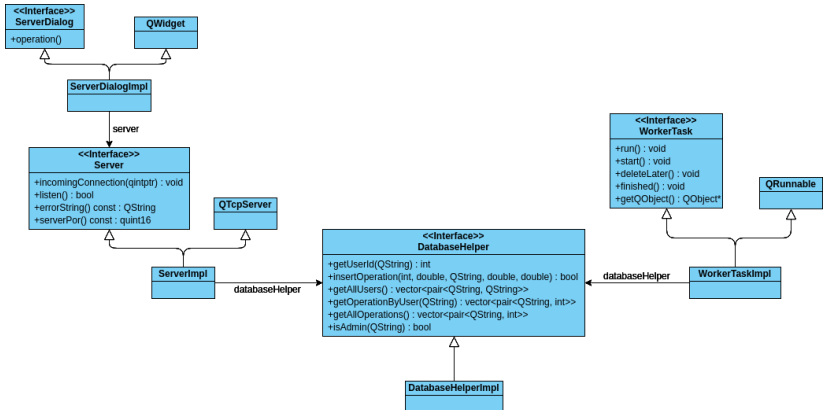
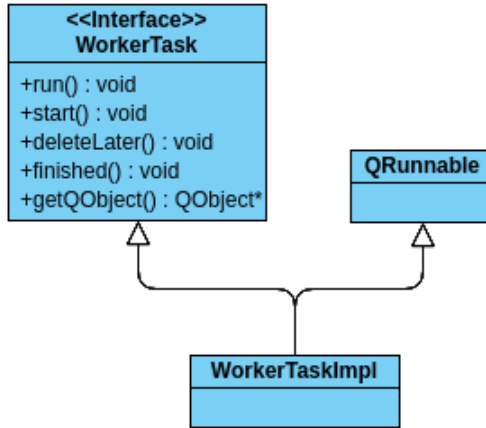


Figura: Thread Pool

# Calculadora - Server



# WorkerTask





# QT Thread

API do QT para o uso de threads, já utilizado pelos widgets para se ter processamento assíncrono. Disponível no core do QT.

- **QRunnable:** Classe base para todos os objetos executáveis.
- **QThreadPool:** Classe para gerenciar um conjunto de threads, dado o pattern ThreadPool.

# QRunnable

```
class HelloWorldTask : public QRunnable{  
    void run() override {  
        qDebug() << "Hello world from thread" <<  
            QThread::currentThread();  
    }  
};
```

# QThreadPool

```
HelloWorldTask *hello = new HelloWorldTask();  
// QThreadPool se torna dona da task e  
// apaga ela automaticamente  
QThreadPool::globalInstance()->start(hello);
```

# Fim