

## ARTIGO: 11374

### Transações

#### Introdução

Nesse artigo veremos como trabalhar com transações através do Veloster Framework.

#### Uso

As transações estão disponíveis na interface de serviço `Veloster<T>`, através do método

Java Code

```
void runTrans(TransactionalOperation operation)
```

O método `runTrans` recebe uma operação transacional, que é definida através da interface

Java Code

```
public interface TransactionalOperation {  
  
    /**  
     * wrapping transactional method  
     *  
     */  
    void execute();  
}
```

#### Exemplo:

Java Code

```
final Veloster<Person> managerP = VelosterRepository.getORM(Person.class);  
final Veloster<PersonGroup> managerG = VelosterRepository.getORM(PersonGroup.class);  
  
final PersonGroup personGroup = new PersonGroup();  
final Person person = new Person();  
person.setGroup(personGroup);  
  
managerG.runTrans(new TransactionalOperation() {  
    @Override  
    public void execute() {  
        managerG.save(personGroup);  
        managerP.save(person);  
    }  
});
```

Todo código que estiver dentro do método `execute` em `TransactionalOperation` é executado em um contexto transacional. Se algum erro ocorrer tudo é desfeito (`rollback`), caso contrário tudo é enviado ao banco de dados (`commit`).