

Brincando com o Lock Pessimista

Uma estratégia mais radical de Lock é impedir, de fato, que os conflitos ocorram por meio de uma trava do registro. Podemos usar o **Lock Pessimista** para conseguir essa trava limitando com que apenas um cliente de cada vez tenha acesso ao registro.

Vamos testar o Lock Pessimista, mas antes vamos inserir um produto no banco de dados:

```
insert into Produto(nome) values('leonardo');
```

Agora, crie uma classe com método `main` para testar o Lock Pessimista fora da aplicação web:

```
public class TesteLockPessimista {
    public static void main(String[] args) {
        AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(Jpa
        EntityManagerFactory factory = context.getBean(EntityManagerFactory.class);

        EntityManager em1 = factory.createEntityManager();

    }
}
```

Vamos pedir um **lock** para um produto e em seguida simular uma alteração em um dos campos:

```
public class TesteLockPessimista {
    public static void main(String[] args) {
        AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(Jpa
        EntityManagerFactory factory = context.getBean(EntityManagerFactory.class);

        EntityManager em1 = factory.createEntityManager();
        EntityManager em2 = factory.createEntityManager();

        em1.getTransaction().begin();
        em2.getTransaction().begin();

        Produto produtoDoEM1 = em1.find(Produto.class, 1);
        em1.lock(produtoDoEM1, LockModeType.PESSIMISTIC_WRITE);

        produtoDoEM1.setNome("Maria");

    }
}
```

Agora, imagine que outro cliente (um outro `EntityManager`) tente pedir o lock desse produto para editar o nome enquanto o primeiro `EntityManager` ainda está usando.

```
public class TesteLockPessimista {
    public static void main(String[] args) {
        AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(Jpa
```

```
EntityManagerFactory factory = context.getBean(EntityManagerFactory.class);

EntityManager em1 = factory.createEntityManager();
EntityManager em2 = factory.createEntityManager();

em1.getTransaction().begin();
em2.getTransaction().begin();

Produto produtoDoEM1 = em1.find(Produto.class, 1);
em1.lock(produtoDoEM1, LockModeType.PESSIMISTIC_WRITE);

produtoDoEM1.setNome("Maria");

Produto produtoDoEM2 = em2.find(Produto.class, 1);
em2.lock(produtoDoEM2, LockModeType.PESSIMISTIC_WRITE);

    }
}
```



O que acontece? Por que não podemos utilizar na nossa aplicação web?