

Mãos a obra: Interceptors

Vamos criar uma classe para interceptar no nosso projeto Alura-lib :

```
@Interceptor
@Transactional
public GerenciadorDeTransacao implements Serializable{

    @Inject
    private EntityManager em;

    public Object efetuaComTransacao(InvocationContext context){
        em.getTransaction().begin();

        try{
            Object resultado = context.proceed();
            em.getTransaction().commit();
            return resultado;
        } catch( Exception e){
            em.getTransaction().rollback();
            throw new RuntimeException(e);
        }
    }
}
```

Crie também a anotação @Transactional

```
@InterceptorBinding
@Target({ElementType.METHOD, ElementType.TYPE})
@Retention(RetentionPolicy.RUNTIME)
public @interface Transacional{}
```

- Na classe DAO apague todas as declarações que abrem e fazem o commit de uma transação.

Instale seu projeto no repositório local: Clique com o botão direito sobre o projeto Alura-lib e selecione Run as > Maven Install

Agora vamos usar o interceptor no projeto Livraria .

Primeiramente vamos atualizar o projeto: Clique com o botão direito do mouse sobre o projeto Livraria e selecione Maven > Update project...

Anotem o método gravar e remover das classes LivroBean e AutorBean com a anotação @Transactional .

Para habilitar nosso Interceptor vamos alterar o arquivo beans.xml e adicionar as seguintes tag s:

```
<interceptor>  
  <class>br.com.alura.alura_lib.tx.GerenciadorDeTransacao</class>  
</interceptor>
```