

**UNIVERSIDADE DO MINDELO**

**DEPARTAMENTO DE ENGENHARIA E CIÊNCIAS DO MAR**

**CURSO DE LICENCIATURA EM**

**ENGENHARIA INFORMÁTICA E SISTEMA COPUTACIONAIS**

**RELATORIO**

**Mindelo, 2023**

**Tema: Criar base dados conectados no java**

**Autor:Anifa Pinheiro N.º 5062**

**Professor: Frederico Soares**

**universidadE dO MINDELo**

**DEPARTAMENTO de ENGENHARIA E CIÊNCIAS DO MAR**

**curso de licenciatura em**

**ENGENHARIA INFORMÁTICA E SISTEMAS COMPUTACIONAIS**

**Relatorio**

**Tema: Criar base dados conectados no java**

**Autor: Anifa Pinheiro, Nº 5062**

**Professor: Frederico Soares**

**Mindelo, 2023**

O trabalho sugerido seria criar duas tabelas relacionais para armazenar informaçoes de clientes e trasações em um sistema java. Para isso, foi desenvolvida uma solução que envolve duas classes: User e trasações.

A classe User tem informaçoes sobre o cliente, incluindo nome, endereço e id do cliente, o valor a transferir.

Para criar a relação entre as tabelas, foram utilizadas chaves primarias e estrangeiras. A tabela users possui uma chave primaria "id", que referencia na tabela transação como chave estrangeira "user\_id", assim cada transação fica vinculada a um cliente especifico.

A solução foi implementada em java, as informções dos clientes e transações podem ser armazenadas em banco de dados(mysql).

Portanto, a solução desenvolvida tem como objetivo criar duas tabelas que armazenam informações de clientes e transações em sistema java.

Foi criado uma tabela no banco de dados usando o comando CREATE TABLE, especificando as colunas a utelizar, para se comunicar com o banco de dados MySQL em no código Java, utilizei a API JDBC (Java Database Connectivity), que nos permitiu estabelecer uma conexão com o banco de dados e executar comandos SQL.

Para inserir dados em na tabela MySQL a partir do código Java, usei um objeto PreparedStatement para preparar uma instrução SQL com espaços reservados para os valores a serem inseridos. Em seguida, usei métodos set para fornecer os valores dos espaços reservados, e executeUpdate para executar a instrução. Para recuperar dados da tabela My\_SQL a partir do código Java, usei um objeto Statement para criar uma instrução SQL simples. Em seguida, o método executeQuery para executar a instrução e recuperar um objeto ResultSet, que contém os resultados da consulta.

Para lidar com transações em nosso código Java, foi utelizado o métodos setAutoCommit(false) e commit() para desativar o modo autocommit e confirmar a transação.

O trabalho solicitado foi bastante interessante, eu pessoalmente passei a conhecer e a esclarecer muitos receios que tinha acerca da ligação base de dados e código fonte, despertei a curiosidade em criar mais uma tabela, adicionar mais atributos para que a mensagem final possa exibir para que conta de quem o valor esta a ser transferido assim como também incluir a hora e a data do acontecimento e claro mais atributos na tabela do user/cliente.