

O programa foi implementado na linguagem de programação Java em sua versão 10. Seu desenvolvimento foi feito no Sistema operacional Windows 10 e não foi testado em outros sistemas, então não podemos garantir um bom funcionamento da aplicação caso seja executada em um outro sistema.

Em relação ao banco de dados, primeiramente precisamos criar uma 'Database' local pelo programa postgresQL, tal base deve ser nomeada 'ecommerce'. Para a base ainda precisaremos criar um usuário chamado 'trabalhofbd', com senha 'senhafinalFBD'.

```
CREATE USER trabalhofbd WITH PASSWORD 'senhafinalFBD';
```

Após a criação do usuário devemos garantir que ele terá acesso a todas as informações e seja possível modificar as tabelas, para isso devemos realizar "Grant" de SELECT/INSERT/DELETE/UPDATE para ele.

```
GRANT SELECT ON ALL TABLES IN SCHEMA public TO trabalhofbd;  
GRANT INSERT ON ALL TABLES IN SCHEMA public TO trabalhofbd;  
GRANT UPDATE ON ALL TABLES IN SCHEMA public TO trabalhofbd;  
GRANT DELETE ON ALL TABLES IN SCHEMA public TO trabalhofbd;
```

Para conectar ao banco de dados, utilizamos o método criado como mostra a figura abaixo, em que seleciona o banco de dados (URL do mesmo), o usuário e a senha. Podemos notar que é importante criarmos o usuário corretamente como descrito acima.

```
public abstract class postgresSQLConnection {  
    private static Connection conn = null;  
  
    static final String JDBC_DRIVER = "org.postgresql.Driver";  
    static final String DB_URL = "jdbc:postgresql://localhost:5432/ecommerce";  
    static final String USER = "trabalhofbd";  
    static final String PASSWORD = "senhafinalFBD";  
}
```

```

public static void conectar(){
    try{
        Class.forName(JDBC_DRIVER);
        conn = DriverManager.getConnection(DB_URL, USER, PASSWORD);
    } catch (SQLException e) {
        e.printStackTrace(System.err);
    } catch (Exception e) {
        e.printStackTrace(System.err);
    }
}

```

O programa também desconecta da base ao final de sua execução para não permitir acessos inesperados. Para se desconectar executa-se o código mostrado na imagem abaixo:

```

public static void desconectar(){
    try{
        conn.close();
    } catch (SQLException e) {
        e.printStackTrace(System.err);
    }
}

```

Para ilustrar o acesso à base de dados, mostraremos um dos exemplos que realizamos em nossa implementação.

```

public static ResultSet consultal(int quantidadeProdutoAnalise){
    String s = "SELECT nomeParceiro "
        + "FROM PARCEIRO JOIN PRODUTO ON(cnpj = cnpjVendedor) NATURAL JOIN COMPRA "
        + "GROUP BY nomeParceiro "
        + "HAVING COUNT(DISTINCT codProduto) >= ?;";

    PreparedStatement stmt = null;

    try{
        stmt = postgresSQLConnection.getConn().prepareStatement(s);

        stmt.setInt( parameterIndex 1, quantidadeProdutoAnalise);

        ResultSet rs = stmt.executeQuery();

        return rs;
    } catch (SQLException e) {
        e.printStackTrace(System.err);
    }

    return null;
}

```