

BANCOS DE DADOS GEOGRÁFICOS

# Conectividade em Bancos de Dados

---

MAJOR IVANILDO BARBOSA

SEÇÃO DE ENSINO DE ENGENHARIA CARTOGRÁFICA

# *Contents*

---

- Funcionalidades
- Arquitetura ODBC
- *Drivers*
- Implementação

# Funcionalidades

---

Aplicações que interagem uma fonte de dados devem ser capazes de:

- Conectar com a fonte de dados: os parâmetros são o nome da fonte de dados (servidor), porta, usuário (*login*), senha (*password*), nome do banco de dados, etc.
- Chamar uma função da API (*Application Programming Interface*) para executar o comando.
- Processar o comando SQL: por intermédio do driver. Esse comando pode retornar resultados (SELECT), não retornar nada (INSERT, UPDATE, DELETE), ou retornar mensagens de erro.
- Processamento da resposta: se o comando retornar um *resultset*, a aplicação pode interpretar dados como número de colunas da resposta ou a confirmação da realização da transação. Se o comando causar um erro, a aplicação deve ser capaz de recuperar as informações de erro.
- Fechar a conexão quando finalizar a interação com a fonte de dados.

# Arquitetura ODBC

---

- *Microsoft Open Database Connectivity;*
- Trata-se de uma API para acesso a bancos de dados;
- Independente de SGBD
- A arquitetura ODBC tem quatro componentes:
  - Aplicação: processa e chama funções ODBC para enviar comandos (*statements*) e recuperar resultados;
  - Gerenciador de *Driver*: carrega e descarrega drivers conforme necessidade da aplicação;
  - *Driver*: processa os comando SQL recebidos e retorna resultados para o aplicativo. O driver pode aplicar alterações de sintaxe para tornar os comando compatíveis com o SGBD associado;
  - Dados: Consistem nas bases de dados a ser acessada pelo usuário, sistema operacional, SGBD e infra de rede necessária para acessar o SGBD;

# Drivers

---

- **Oracle:** O driver ODBC Oracle habilita aplicações para conectar bases de dados Oracle (<http://www.oracle.com/technetwork/database/windows/index-098976.html>)
- **SQL Server:** O Microsoft ODBC Driver 11 para SQL Server fornece conectividade nativa do Windows para o Microsoft SQL Server (<https://www.microsoft.com/pt-br/download/details.aspx?id=36434>);
- **psqlODBC:** driver ODBC oficial do PostgreSQL (<https://odbc.postgresql.org/>).
- **SQLite:** driver desenvolvido para acesso ao SGBD SQLite (<http://www.ch-werner.de/sqliteodbc/>)
- **JDBC:** API desenvolvido pela Oracle para aplicações desenvolvidas em Java (<https://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/>).

# Implementação

## Java

---

```
package connectPostgres;
```

```
import java.sql.*;
```

```
import java.sql.DriverManager;
```

```
public class ConexaoPGSQL {
```

```
    private Connection con = null;
```

```
    public Statement stmt;
```

```
    public ResultSet rs;
```

```
    private String endereco;
```

```
    private String usuario;
```

```
    private String senha;
```

```
    /** Usada para a conexao com o banco de dados */
```

```
    /** Usada para realizar as instrucoes SQL */
```

```
    /** Retorna os dados das tabelas do banco */
```

```
    /** Usada para receber o endereco da base de dados*/
```

```
    /** Usada para receber o nome do usuario do banco */
```

```
    /** Usada para receber a senha do usuario do banco */
```

# Implementação

## Java

---

```
public void listaNomes(String strEnd, String strUsuario, String strSenha) {
    try {
        Class.forName("org.postgresql.Driver");    /** Pasando o nome do Driver do PostgreSQL */
        con = DriverManager.getConnection(strEnd, strUsuario, strSenha);
        /** Obtendo a conexao com o banco de dados*/
        stmt = con.createStatement();              /** Criando o Statement */

        String sql_str = "SELECT nome FROM cadastro";
        ResultSet rs = stmt.executeQuery(sql_str);
        String name = rs.getString("nome");
    }
}
```

# Implementação

## Java

---

```
public void apagaNome(String strEnd, String strUsuario, String strSenha, String nome) {
    try {
        Class.forName("org.postgresql.Driver");    /** Pasando o nome do Driver do PostgreSQL */
        con = DriverManager.getConnection(strEnd, strUsuario, strSenha);
        /** Obtendo a conexao com o banco de dados*/
        stmt = con.createStatement();                /** Criando o Statement */

        String sql_str = "DELETE * FROM cadastro WHERE nome =" + nome;
        stmt.executeUpdate();                        /** retorna a quantidade de linhas selecionadas */
    }
}
```

# Implementação

## *Python*

---

```
from pyPgSQL import PgSQL #importar o driver
con = PgSQL.connect(host='servidor', # parâmetros de conexão
                    database='banco de dados',
                    user='usuario',
                    password='senha')

cursor = con.cursor()
cursor.execute('INSERT INTO TABELA (CAMPO1, CAMPO2, CAMPO3)
              VALUES (?, ?, ?)',
              (valor1, valor2, valor3)) # executa ordem SQL

con.commit()
```

# Implementação

## *Python*

---

```
rs = cursor.fetchone() # busca uma linha ou;  
rs = cursor.fetchall() # busca todas as linhas;  
  
print(rs[0]) # valor da primeira coluna  
  
for linha in cursor:  
    print linha[0] # valor da primeira coluna
```