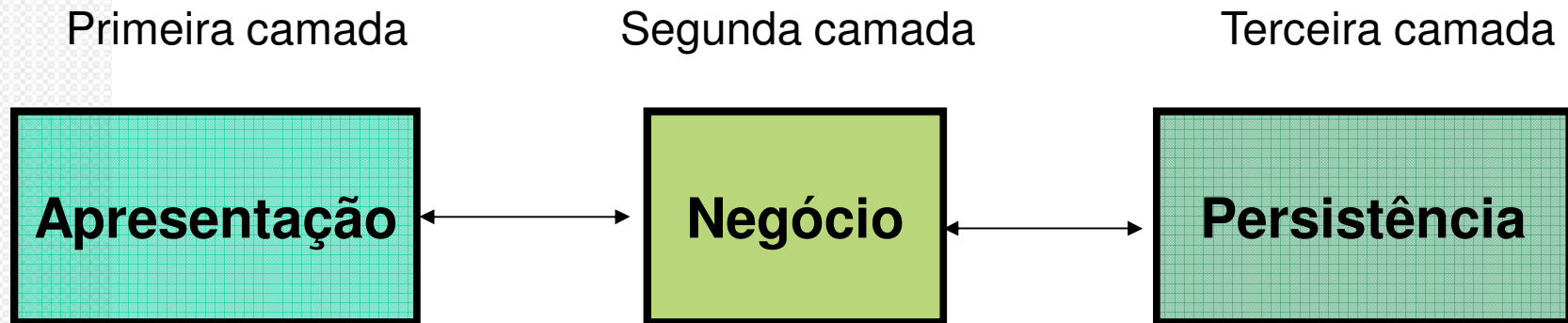




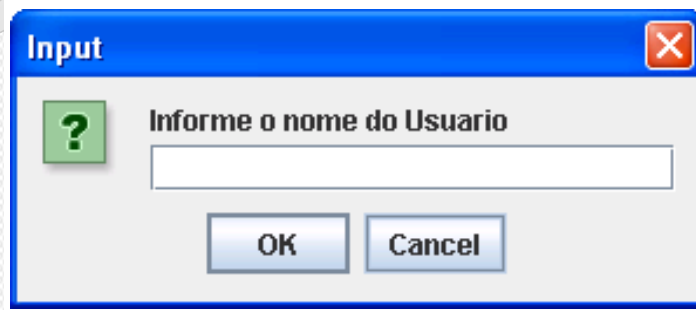
# Camada de Persistência

# Arquitetura para Desenvolvimento



Aplicações em três camadas é o padrão de desenvolvimento para aplicativo !!!

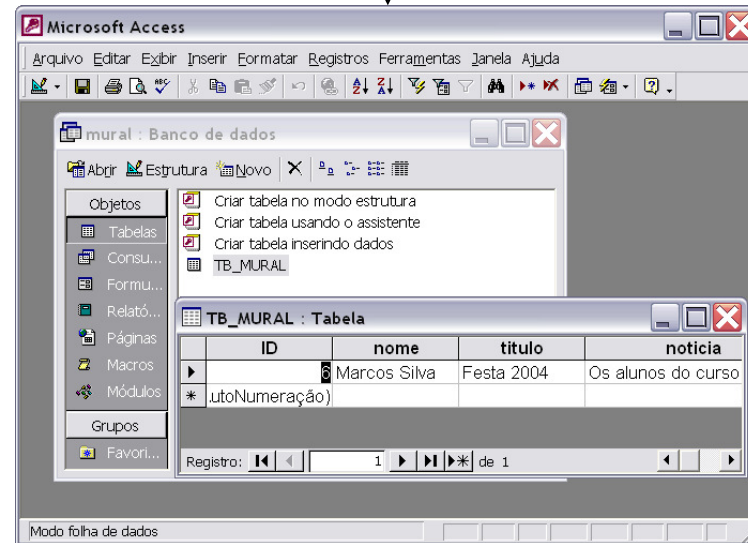
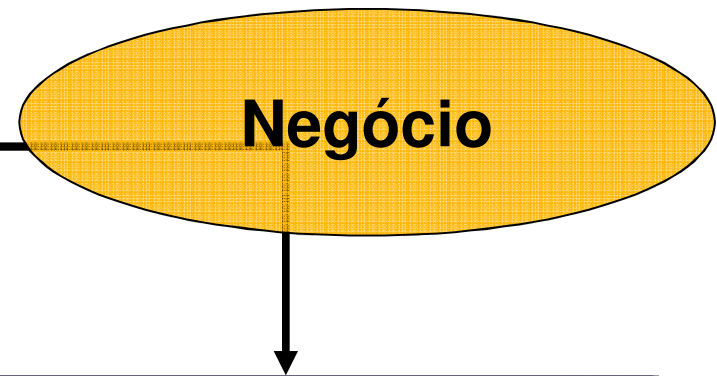
# Aplicação



A screenshot of a Windows-style dialog box titled "Input". It contains a green question mark icon, the text "Informe o nome do Usuario", a text input field, and "OK" and "Cancel" buttons.

## Apresentação

## Persistência



A screenshot of the Microsoft Access application window. The main window shows a database named "mural : Banco de dados". A table named "TB\_MURAL" is selected, and its structure is displayed in a separate window. The table has four columns: ID, nome, titulo, and noticia. A single record is shown with the values: Marcos Silva, Festa 2004, Os alunos do curso.

| ID | nome         | titulo     | noticia            |
|----|--------------|------------|--------------------|
| 1  | Marcos Silva | Festa 2004 | Os alunos do curso |

# Camada de Persistência



Camada de Persistência

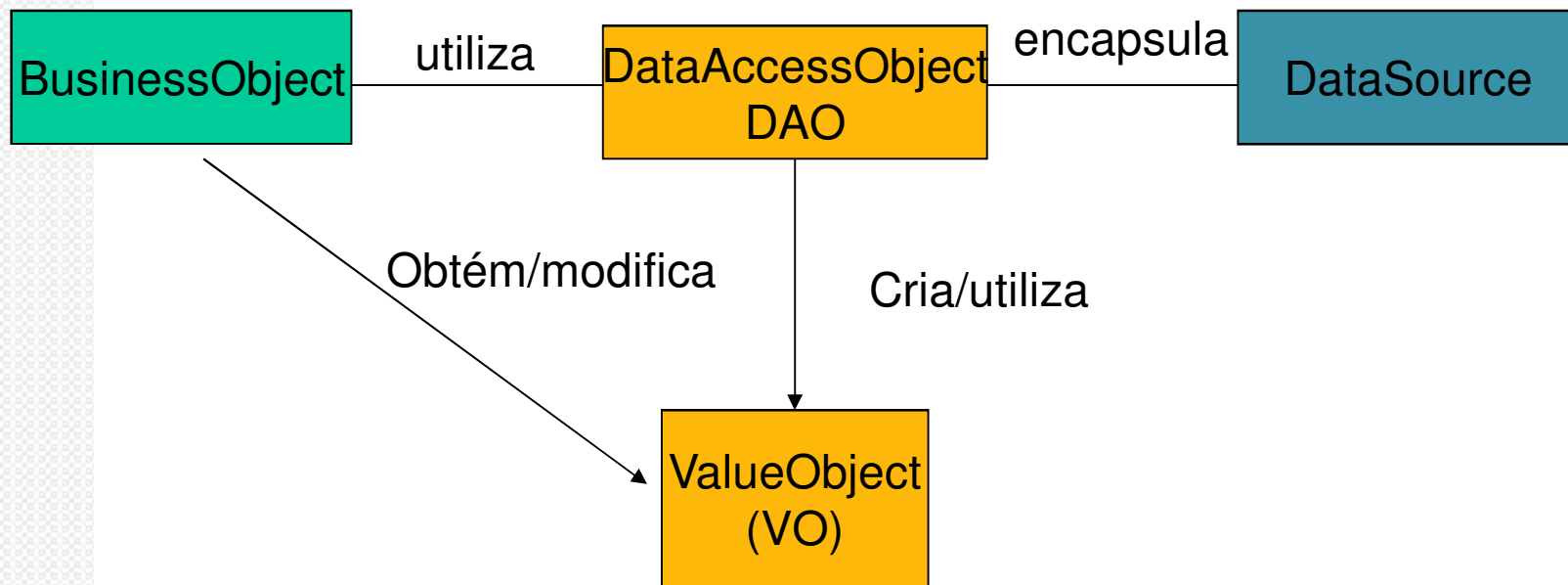
| TB_USUARIO : Tabela |               |
|---------------------|---------------|
| Nome do campo       | Tipo de dados |
| id                  | Número        |
| nome                | Texto         |
| cpf                 | Texto         |
| email               | Texto         |



## O Modelo DAO

- DAO (Data Access Object) é utilizado para extrair e encapsular todos os acessos a origem dos dados.
- O DAO gerencia a conexão com a origem dos dados para manipulação dos dados selecionados
- Utilizado para construir a camada de persistência.

# Estrutura do DAO





## O Modelo DAO

- BusinessObject: o cliente do dados
- DataAccessObject (DAO): encapsula o acesso a fonte de dados
- DataSource: a fonte de dados
- ValueObject (VO): o objeto que representa os dados

# Implementando um DAO

```
public class ObjetoDAO{
    public boolean inserirUmObjeto(Objeto){
        // INSERT
    }

    public boolean apagarObjeto(Objeto){
        //DELETE
    }

    public boolean apagarObjeto(chavePrimaria){
        //DELETE
    }

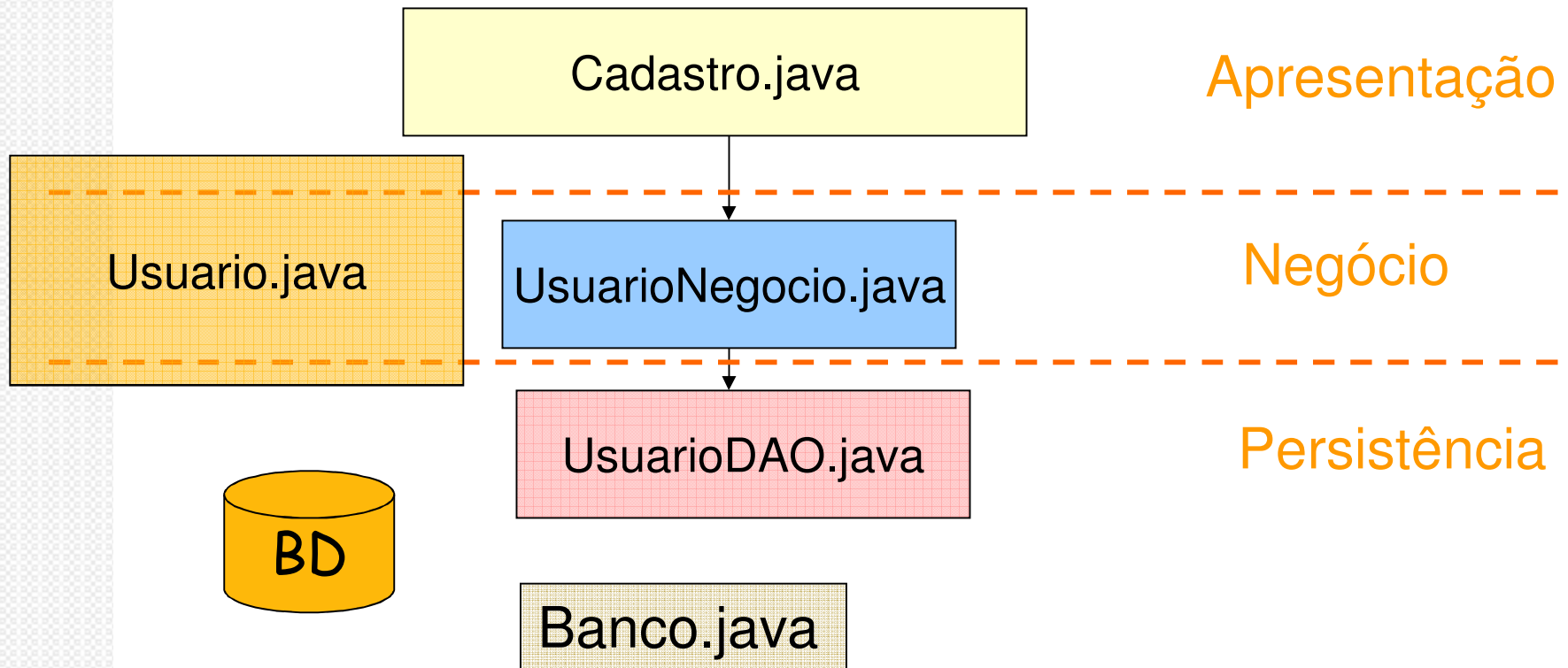
    public boolean atualizarObjeto(Objeto){
        //UPDATE
    }

    public Objeto buscarPorCHavePrimariaObjeto(chavePrimaria){
        // SELECT
    }

    public Colecao buscarTodosObjeto(){
        // SELECT
    }
}
```



# Estudo de Caso





# Usuario.java

```
public class Usuario{

    public Usuario() {
    }

    public Usuario(int id, String nome, String endereco, String email){
        this.setId(id);
        this.setNome(nome);
        this.setEndereco(endereco);
        this.setEmail(email);
    }

    private int id;
    private String nome;
    private String endereco;
    private String email;
```



# Usuario.java

```
public String getNome() {  
    return nome;  
}  
public void setNome(String newNome) {  
    nome = newNome;  
}  
  
...  
public void setEmail(String newEmail) {  
    email = newEmail;  
}  
public String getEmail() {  
    return email;  
}  
}
```

# UsuarioDAO.java

```
import java.sql.*;
...
public class UsuarioDAO {
    public UsuarioDAO(){
        Banco banco – new Banco();
    }
    public void insereUsuario (Usuario usu){
        try {
            Connection con = Banco.getConexao();
            Statement stmt = con.createStatement();
            String comandoSQL = "INSERT INTO TB_USUARIO(NOME, ENDERECO,
EMAIL)" +
                " VALUES ('"+ usu.getNome() + "','" + usu.getEndereco() +
                "','" + usu.getEmail() + "'" ) ";
            stmt.executeUpdate(comandoSQL );
            stmt.close();
            con.commit();
            con.close();
        } catch (SQLException e){
            System.out.println("Problemas ao abrir a conexão com o BD");
        }
    }
    ...
}
```

# UsuarioDAO.java

```
public Usuario consultaUsuario(int id) {

    try{
        Connection con = Banco.getConexao();
        Usuario usu = new Usuario();
        Statement stat = con.createStatement();

        ResultSet res = stat.executeQuery("SELECT * FROM TB_USUARIO where id = "+id);

        if(res.next()) {
            usu.setCpf(res.getInt("ID"));
            usu.setNome(res.getString("NOME"));
            usu.setEndereco(res.getString("ENDERECO"));
            usu.setCep(res.getString("EMAIL"));
        }

        return usu;
    } catch(SQLException e) {
        System.out.println("Erro = "+e.getMessage());
        return null;
    } }
```

# UsuarioDAO.java

```
public Vector consultaUsuarios(){
    Vector listaUsuarios = new Vector();
    Usuario usu = null;
    try {
        Connection con = Banco.getConexao();
        Statement stmt = con.createStatement();
        String query = "SELECT * FROM TB_USUARIO" ;
        ResultSet res = stmt.executeQuery(query);

        while (res.next() ){
            usu = new Usuario(res.getInt("ID"), res.getString("NOME"),
                res.getString("ENDERECO"), res.getString("EMAIL"));
            listaUsuarios.add(usu);
        }
        stmt.close();
        con.close();
    } catch (SQLException e){
        System.out.println("Problemas ao abrir a conexao com o BD"); }

    return listaUsuarios; }
```

# UuarioDAO.java

```
public int deleteUuario(int id) {  
  
    try{  
        Connection con = Banco.getConexao();  
  
        Statement stat;= con.createStatement();  
  
        stat.executeUpdate("Delete FROM Uuario WHERE id = "+id);  
        stat.close();  
        return(0);  
    }  
  
    catch(SQLException e)  
    {  
        return(1);  
    }  
}
```

# Banco.java

```
import java.sql.*;

public class Banco {
    // Access
    private static String DRIVER_BD = "sun.jdbc.odbc.JdbcOdbcDriver";
    private static String URL_BD = "jdbc:odbc:agenda";
    private static String usuario = "";
    private static String senha = "";
    private static Connection con;

    public Banco() {
        try{
            Class.forName(DRIVER_BD);
        } catch (ClassNotFoundException e){
            System.out.println("Problemas ao carregar o driver");
        }
    }

    public static Connection getConexao(){
        try {
            if ((con == null) || (con.isClosed()))
                con = DriverManager.getConnection(URL_BD, usuario, senha);
        } catch (SQLException e){
            System.out.println("Problemas ao abrir a conexao com o
BD");
        }
        return con;
    }
}
```





# UsuarioNegocio.java

```
public class UsuarioNegocio {  
  
    public UsuarioNegocio() {  
    }  
  
    public void salva(Usuario usu){  
        UsuarioDAO usuBD = new UsuarioDAO();  
        usuBD.inserirUsuario(usu);  
    }  
  
    ....  
}
```



# Cadastro.java

```
public class Cadastro {
```

```
public static void main(String[] args) {
```

```
    String nome = JOptionPane.showInputDialog("Informe o nome do  
    Usuario");
```

```
    ...
```

```
    Usuario usu = new Usuario();  
    usu.setNome(nome);
```

```
    ...
```

```
    UsuarioNegocio usuNeg = new UsuarioNegocio();  
    usuNeg.salva(usu);
```

```
}}
```