

Curso Superior de Desenvolvimento de Software Multiplataforma

Carlos Eduardo da Silva, 3011392323015

Matheus Prusch, 3011392323025

Raphael Cremer Albiero, 3011392323028

Samuel Porcer Pregnoatto, 3011392323042

Scripts do Projeto Interdisciplinar

Ada Company

Profº Jones Artur Gonçalves

Votorantim

Junho, 2024

RESUMO

O projeto é focado em criar um software baseado no modelo CRM (Customer Relationship Management) tanto em versão desktop quanto web. O software deve ser capaz de gerar orçamentos, acompanhar os status dos pedidos e fornecer uma solução de análise aos administradores sobre os serviços prestados. No caso da web, pretende-se implementar também um ChatBot no próximo semestre para responder dúvidas mais frequentes.

A empresa foi pensada para promover a acessibilidade digital por meio da prestação de serviços e consultoria na área de desenvolvimento, principalmente web. Pois foi identificado por meio de pesquisas e discussões a necessidade de pensar nos públicos mais negligenciados, como por exemplo, os idosos.

Dessa forma, o modelo de negócio é a prestação de serviço na área de desenvolvimento de sites e software para outras empresas que precisem melhorar sua experiência de usuário, tornando-a mais acessível a diferentes públicos.

SUMÁRIO

1. RESUMO DO PROJETO	1
1.1. Proposta do Software	1
1.2. Justificativa	2
2. DIAGRAMAS DO PROJETO	3
2.1. Modelo Conceitual	3
2.2. Modelo Lógico	4
2.1. Diagrama de Classes.....	5
3. SCRIPTS DO PROJETO	6
3.1. SCRIPTS DE CRIAÇÃO DO BANCO DE DADOS	6
3.2. SCRIPTS DE INSERÇÃO.....	8
3.3. SCRIPTS DE SELECT	11
3.4. SCRIPTS DE UPDATE	12
3.5. SCRIPTS DE DELETE	Erro! Indicador não definido.
4. OUTROS	Erro! Indicador não definido.

LISTA DE FIGURAS

Figura 1 - Modelo Conceitual.....	3
Figura 2 - Modelo Lógico	4
Figura 3 - Diagrama de Classe.....	5

1. RESUMO DO PROJETO

1.1. Proposta do Software

O presente documento mostra o projeto de um website responsivo projetado para atender ao objetivo de reduzir as desigualdades por meio da promoção da acessibilidade. O site foi cuidadosamente desenvolvido para garantir que todas as pessoas, independentemente de suas habilidades ou dispositivos que utilizam, possam navegar e interagir facilmente. Ao visitá-lo, os usuários encontrarão uma experiência inclusiva desde o primeiro momento. Foram utilizadas práticas de design acessível, como cores contrastantes e fontes legíveis, para tornar o conteúdo facilmente perceptível por todos. Além disso, foram implementados recursos de áudio e texto alternativo para garantir que pessoas com deficiência visual ou auditiva possam acessar todas as informações de maneira eficaz. As páginas foram organizadas de forma intuitiva, com uma navegação simplificada para facilitar a busca por informações relevantes. Foram disponibilizados recursos educativos sobre a importância da acessibilidade e como as pessoas podem contribuir para a redução das desigualdades em suas próprias comunidades.

A missão do projeto é criar uma plataforma online onde a acessibilidade seja uma prioridade, demonstrando que a inclusão é possível em todas as facetas da vida digital. Ao construir pontes de igualdade, estamos criando um espaço onde todos são valorizados e possuem oportunidade de contribuir plenamente para uma sociedade mais justa e equitativa.

1.2. Justificativa

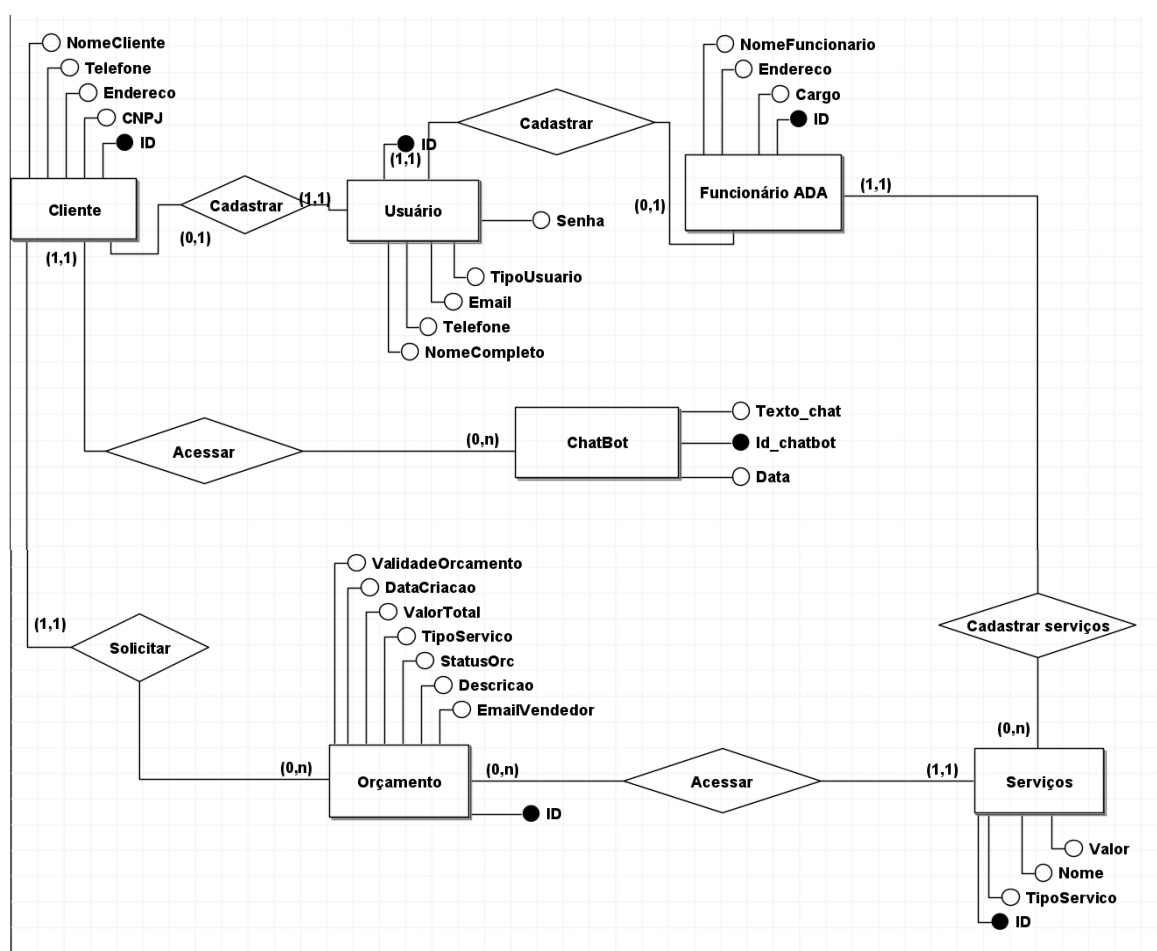
Atualmente há uma grande parcela da sociedade que possui grandes dificuldades com a utilização de ferramentas digitais, tanto para navegação pessoal quanto para utilização em ambiente de trabalho. Dessa forma, buscando uma solução para o problema, foi criado a Ada Company com o objetivo de facilitar a navegação web desses públicos com específicos dificuldades por meio da prestação de serviços de consultoria e desenvolvimento.

2. DIAGRAMAS DO PROJETO

2.1. Modelo Conceitual

Foi utilizado a ferramenta brModelo para elaboração do modelo conceitual e lógico.

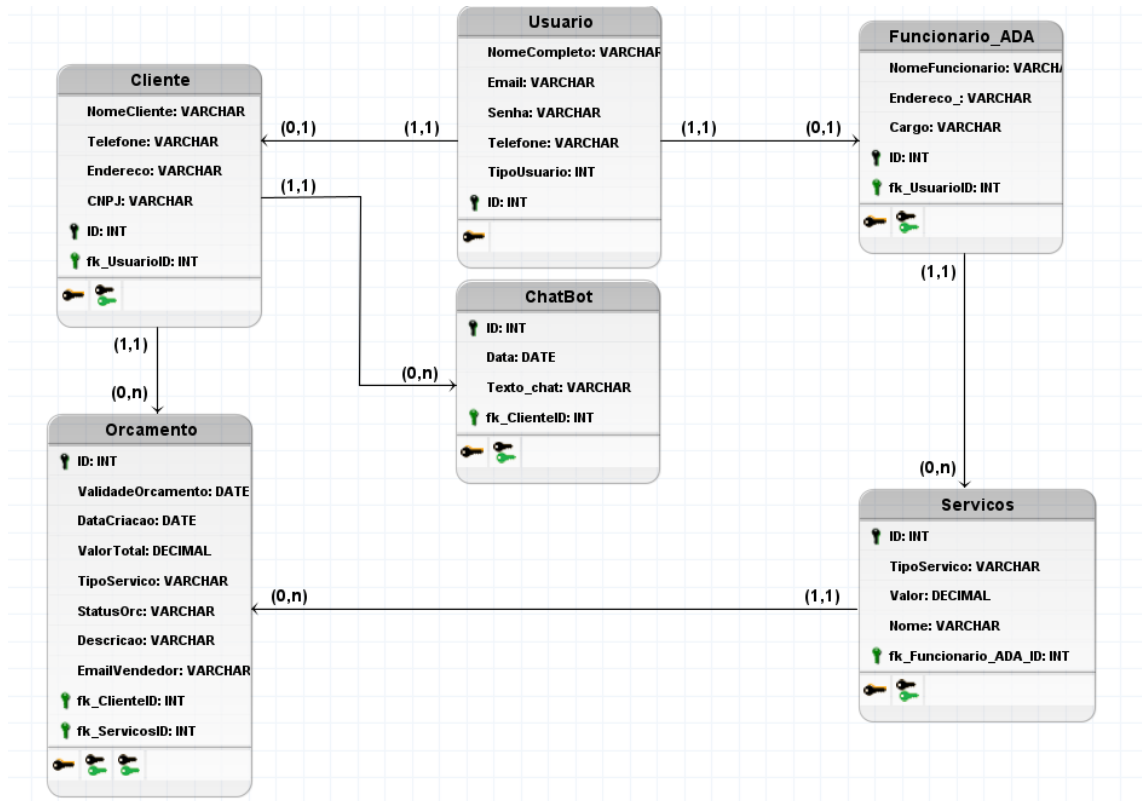
Figura 1 - Modelo Conceitual



Fonte: elaboração própria com a ferramenta brModelo (2024)

2.2. Modelo Lógico

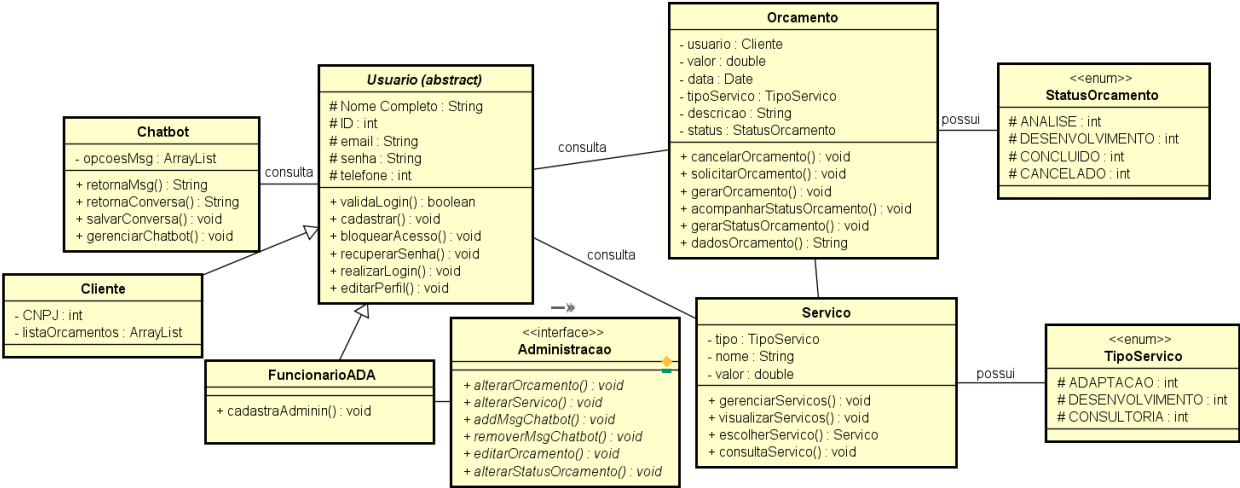
Figura 2 - Modelo Lógico



Fonte: elaboração própria com a ferramenta brModelo (2024)

2.1. Diagrama de Classes

Figura 3 - Diagrama de Classe



Fonte: elaboração própria com a ferramenta brModelo (2024)

3. SCRIPTS DO PROJETO

3.1. SCRIPTS DE CRIAÇÃO DO BANCO DE DADOS

```
CREATE DATABASE IF NOT EXISTS ADACompany;

USE ADACompany;

CREATE TABLE Usuario (
  ID INT PRIMARY KEY AUTO_INCREMENT,
  NomeCoIDmpleto VARCHAR(100),
  Email VARCHAR(50),
  Senha VARCHAR(50),
  Telefone VARCHAR(11),
  TipoUsuario BIT /*0 CLIENTE E 1 FUNCIONARIO*/
);

CREATE TABLE Cliente (
  ID INT PRIMARY KEY AUTO_INCREMENT,
  Telefone VARCHAR(11),
  Endereco VARCHAR(100),
  NomeCliente VARCHAR(100),
  CNPJ VARCHAR(14),
  fk_UsuarioID INT,
  FOREIGN KEY (fk_UsuarioID) REFERENCES Usuario(ID)
);

CREATE TABLE Funcionario (
  ID INT PRIMARY KEY AUTO_INCREMENT,
  NomeFuncionario VARCHAR(100),
  Cargo VARCHAR(50),
  fk_UsuarioID INT,
  FOREIGN KEY (fk_UsuarioID) REFERENCES Usuario(ID)
);

CREATE TABLE Orcamento (
  ID INT PRIMARY KEY AUTO_INCREMENT,
  ValidadeOrcamento DATE,
  DataCriacao DATE,
  ValorTotal DECIMAL(10,2),
  TipoServico VARCHAR(20),
  StatusOrc VARCHAR(15),
  Descricao VARCHAR(500),
  EmailVendedor VARCHAR(100),
  fk_IDCliente INT,
```

```
fk_IDCServico INT,  
FOREIGN KEY (fk_IDServico) REFERENCES Servico(ID),  
FOREIGN KEY (fk_IDCliente) REFERENCES Cliente(ID)  
);  
  
CREATE TABLE Servico (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    TipoServico VARCHAR(20) NOT NULL,  
    Valor DECIMAL(10, 2),  
    Nome VARCHAR(20)  
);  
  
/* Mudanças 23/05: */  
  
-- syntax do ID para AUTO_INCREMENT (mySQL);  
-- mudança tipo de CNPJ e Telefone para VARCHAR;  
-- adequação datas para o tipo DATE (Java tipo sql.Date);  
  
/* Mudanças 07/06: */  
-- adição de nova coluna na tabela Usuário: TipoUsuario INT;  
  
/* Mudanças 09/06: */  
-- alteração de campo: TipoUsuario INT para BIT.
```

3.2. SCRIPTS DE INSERÇÃO

```
-- Funcionários
INSERT INTO Usuario (NomeCompleto, Email, Senha, Telefone, TipoUsuario)
VALUES
('Matheus Prusch', 'matheus.prusch@example.com', 'senhaMatheus',
'11987654321', 1),
('Samuel', 'samuel@example.com', 'senhaSamuel', '11987654322', 1),
('Rafael', 'rafael@example.com', 'senhaRafael', '11987654323', 1),
('Carlos', 'carlos@example.com', 'senhaCarlos', '11987654324', 1);

INSERT INTO Funcionario (NomeFuncionario, Cargo, fk_UsuarioID) VALUES
('Matheus Prusch', 'Administrador', 1),
('Samuel', 'Administrador', 2),
('Rafael', 'Pedreiro', 3),
('Carlos', 'Administrador', 4);

-- Clientes
INSERT INTO Usuario (NomeCompleto, Email, Senha, Telefone, TipoUsuario)
VALUES
('Softech Solutions', 'contact@softechsolutions.com', 'senhaSoftech',
'11987654325', 0),
('Refrigerante Rossi', 'info@refrigeranterossi.com', 'senhaRossi',
'11987654326', 0),
```

Fonte: elaboração própria (2024)

```
('Alpha Networks', 'sales@alphanetworks.com', 'senhaAlpha',
'11987654327', 0),
('Beta Technologies', 'support@betatech.com', 'senhaBeta', '11987654328',
0),
('Gamma Enterprises', 'contact@gammaenterprises.com', 'senhaGamma',
'11987654329', 0),
('Delta Innovations', 'info@deltainnovations.com', 'senhaDelta',
'11987654330', 0),
('Epsilon Motors', 'sales@epsilonmotors.com', 'senhaEpsilon',
'11987654331', 0),
('Zeta Pharmaceuticals', 'support@zetapharma.com', 'senhaZeta',
'11987654332', 0),
('Eta Communications', 'contact@etacommunications.com', 'senhaEta',
'11987654333', 0),
('Theta Logistics', 'info@thetalogistics.com', 'senhaTheta',
'11987654334', 0);

INSERT INTO Cliente (Telefone, Endereco, NomeCliente, CNPJ, fk_UsuarioID)
VALUES
('11987654325', 'Rua A, 123', 'Softech Solutions', '11111111000111', 5),
```

```

('11987654326', 'Rua B, 456', 'Refrigerante Rossi', '2222222000122', 6),
('11987654327', 'Rua C, 789', 'Alpha Networks', '3333333000133', 7),
('11987654328', 'Rua D, 101', 'Beta Technologies', '4444444000144', 8),
('11987654329', 'Rua E, 202', 'Gamma Enterprises', '5555555000155', 9),
('11987654330', 'Rua F, 303', 'Delta Innovations', '6666666000166', 10),
('11987654331', 'Rua G, 404', 'Epsilon Motors', '7777777000177', 11),
('11987654332', 'Rua H, 505', 'Zeta Pharmaceuticals', '8888888000188',
12),
('11987654333', 'Rua I, 606', 'Eta Communications', '9999999000199',
13),
('11987654334', 'Rua J, 707', 'Theta Logistics', '0000000000100', 14);

-- Servico
INSERT INTO Servico (TipoServico, Nome, Valor) VALUES
('ADAPTACAO', 'Adaptacao', 5000),
('DESENVOLVIMENTO', 'Desenvolvimento', 7000),
('CONSULTORIA', 'Consultoria', 4000);

-- Orcamento
INSERT INTO Orcamento (ValidadeOrcamento, DataCriacao, ValorTotal,
TipoServico, StatusOrc, Descricao, EmailVendedor, fk_IDCliente) VALUES
('2024-10-11', '2024-02-11', 5000.00, 'ADAPTACAO', 'ANALISE',
'Solicitação de adaptação de layout para tornar o site mais acessível ao
público idoso.', '', 5),
('2024-11-15', '2024-03-15', 7000.00, 'DESENVOLVIMENTO', 'ANALISE',
'Solicitação de desenvolvimento de site do zero, com foco em
acessibilidade para o público infantil.', '', 6),
('2024-12-20', '2024-04-20', 4000.00, 'CONSULTORIA', 'ANALISE',
'Solicitação de consultoria na área de desenvolvimento de site voltado
para o público idoso.', '', 7),
('2024-11-15', '2024-03-15', 2250.00, 'ADAPTACAO', 'DESENVOLVIMENTO',
'Adaptação de layout para tornar o site mais acessível ao público
idoso.', 'matheus.prusch@example.com', 8),
('2024-06-11', '2024-06-11', 7700.00, 'DESENVOLVIMENTO', 'CONCLUIDO',
'Desenvolvimento de aplicativo educacional com foco em acessibilidade
para crianças com deficiência visual.', 'samuel@example.com', 9),
('2024-09-30', '2024-01-30', 3500.00, 'DESENVOLVIMENTO', 'CANCELADO',
'Adaptação de infraestrutura para tornar o site mais acessível ao público
idoso.', 'matheus.prusch@example.com', 10),
('2024-07-05', '2024-03-05', 4400.00, 'CONSULTORIA', 'DESENVOLVIMENTO',
'Consultoria em marketing digital com foco em promover a acessibilidade
em websites corporativos.', 'matheus.prusch@example.com', 11),
('2024-06-11', '2024-06-11', 6000.00, 'CONSULTORIA', 'CONCLUIDO',
'Desenvolvimento de site para empresa de entretenimento infantil, com
ênfase em tornar o conteúdo acessível para crianças com deficiência.',
'samuel@example.com', 12),
('2024-06-02', '2024-03-02', 3850.00, 'CONSULTORIA', 'CANCELADO',
'Solicitação de consultoria técnica em desenvolvimento de sistemas para

```

```
promover a acessibilidade em plataformas de comércio eletrônico.',  
'matheus.prusch@example.com', 13);
```

3.3. SCRIPTS DE SELECT

Selects de exemplo que poderiam ser utilizados para inteligência comercial e tomada de decisão.

```
-- Select do valor total de orçamentos concluídos por cliente.
SELECT c.NomeCliente, SUM(o.ValorTotal) AS ValorTotalConcluido
FROM Cliente c
INNER JOIN Orcamento o ON c.ID = o.fk_IDCliente
WHERE o.StatusOrc = 'CONCLUIDO'
GROUP BY c.NomeCliente;

-- Select do valor total de orçamentos pendentes por cliente (que podem
ser concluídos).
SELECT c.NomeCliente, SUM(o.ValorTotal) AS
ValorTotalAnaliseDesenvolvimento
FROM Cliente c
INNER JOIN Orcamento o ON c.ID = o.fk_IDCliente
WHERE o.StatusOrc IN ('ANALISE', 'DESENVOLVIMENTO')
GROUP BY c.NomeCliente;

-- orçamentos concluidos para cada Tipo de Serviço.
-- Verifica rentabilidade.
SELECT s.TipoServico,
       SUM(CASE WHEN o.StatusOrc = 'CONCLUIDO' THEN o.ValorTotal ELSE 0 END)
AS ValorTotalConcluido
FROM Servico s
LEFT JOIN Orcamento o ON s.TipoServico = o.TipoServico
GROUP BY s.TipoServico;

--Select que mostra os tipos de serviços mais solicitados pelos clintes.
SELECT
    o.TipoServico,
    COUNT(*) AS Quantidade
FROM
    Orcamento o
GROUP BY
    o.TipoServico;
```

3.4. SCRIPTS DE UPDATE

```
-- Atualização de +10% do preço dos serviços.  
UPDATE Servico  
SET Valor = Valor * 1.10;  
  
-- Alteração do status dos orçamentos em análise.  
SET StatusOrc = 'CANCELADO'  
WHERE ID = 1;  
  
UPDATE Orcamento  
SET StatusOrc = 'CANCELADO'  
WHERE ID = 2;
```


4. SCRIPTS USANDO JDBC

Nosso projeto já possui uma parte da implementação com o banco de dados, utilizando JDBC_(Java Database Connectivity).

4.1. UsuarioDAO

Exemplo de dois métodos da classe UsuarioDAO: cadastrar um usuário (realiza um insert) e do método que consulta todos os usuários do sistema (realiza um select *).

```
public void cadastrar(Usuario user) throws ExceptionDAO {

    String sqlUsuario = "insert into Usuario(NomeCompleto,
Email, Senha, Telefone, TipoUsuario) value (?, ?, ?, ?, ?)";
    String sqlCliente = "insert into Cliente(NomeCliente, Telefone,
Endereco, CNPJ, fk_UsuarioID) value (?, ?, ?, ?, ?)";
    String sqlFuncionario = "insert into Funcionario(NomeFuncionario,
Cargo, fk_UsuarioID) VALUES (?, ?, ?)";

    Connection connection = null;
    PreparedStatement pStatementUsuario = null;
    PreparedStatement pStatementCliente = null;
    PreparedStatement pStatementFuncionario = null;
    ResultSet generatedKeys = null;

    try {
        connection = new ConexaoBD().getConnection();
        pStatementUsuario = connection.prepareStatement(sqlUsuario,
PreparedStatement.RETURN_GENERATED_KEYS);
        pStatementUsuario.setString(1, user.getNomeCompleto());
        pStatementUsuario.setString(2, user.getEmail());
        pStatementUsuario.setString(3, user.getSenha());
        pStatementUsuario.setString(4, user.getTelefone());

        // TipoUsuario BIT no BD.
        if (user instanceof Cliente) {
            pStatementUsuario.setBoolean(5, false); // Cliente
        } else {
            pStatementUsuario.setBoolean(5, true); // Admin
        }
    }
```

```

        pStatementUsuario.executeUpdate();

// O ID do user deve ser referenciado na tabel Cliente/Funcionario
como chave estrangeira.

        generatedKeys = pStatementUsuario.getGeneratedKeys();
        if (generatedKeys.next()) {
            int userID = generatedKeys.getInt(1);

            if (user instanceof Cliente) {
                Cliente cliente = (Cliente) user;
                pStatementCliente =
connection.prepareStatement(sqlCliente);
                pStatementCliente.setString(1,
cliente.getNomeCompleto());
                pStatementCliente.setString(2,
cliente.getTelefone());
                pStatementCliente.setString(3,
cliente.getEndereco());
                pStatementCliente.setString(4, cliente.getCNPJ());
                pStatementCliente.setInt(5, userID);
                pStatementCliente.executeUpdate();

            }

            else if (user instanceof Funcionario) {
                Funcionario funcionario = (Funcionario) user;
                pStatementFuncionario =
connection.prepareStatement(sqlFuncionario);
                pStatementFuncionario.setString(1,
funcionario.getNomeCompleto());
                pStatementFuncionario.setString(2,
funcionario.getCargo());
                pStatementFuncionario.setInt(3, userID);
                pStatementFuncionario.executeUpdate();

            }
        }
    } catch... fechamentos dos statements no bloco finally.
}

    public ArrayList<Usuario> consultarUsuarios(String nome, int
tipoConsulta) throws ExceptionDAO {
        String sqlCliente = "SELECT u.*, c.Telefone AS
clienteTelefone, c.Endereco, c.NomeCliente, c.CNPJ "
            + "FROM Usuario u JOIN Cliente c ON u.ID =
c.fk_UsuarioID " + "WHERE ";

        String sqlFuncionario = "SELECT u.*, f.NomeFuncionario, f.Cargo "

```

```

        + "FROM Usuario u JOIN Funcionario f ON u.ID =
f.fk_UsuarioID " + "WHERE ";
    if (tipoConsulta == 1) {
        sqlCliente += "u.NomeCompleto LIKE ?";
        sqlFuncionario += "u.NomeCompleto LIKE ?";

    } else { // consulta no email
        sqlCliente += "u.Email LIKE ?";
        sqlFuncionario += "u.Email LIKE ?";
    }

    PreparedStatement pStatementCliente = null;
    PreparedStatement pStatementFuncionario = null;
    Connection connection = null;
    ArrayList<Usuario> usuarios = new ArrayList<>();

    try {
        connection = new ConexaoBD().getConnection();

        // cliente
        pStatementCliente = connection.prepareStatement(sqlCliente);
        pStatementCliente.setString(1, "%" + nome + "%");
        ResultSet rsCliente = pStatementCliente.executeQuery();
        while (rsCliente.next()) {
            Cliente cliente = new Cliente();

            cliente.setNomeCompleto(rsCliente.getString("NomeCompleto"));
            cliente.setEmail(rsCliente.getString("Email"));

            cliente.setTelefone(rsCliente.getString("clienteTelefone"));
            cliente.setEndereco(rsCliente.getString("Endereco"));
            cliente.setCNPJ(rsCliente.getString("CNPJ"));
            usuarios.add(cliente);
        }

        // funcionario
        pStatementFuncionario =
connection.prepareStatement(sqlFuncionario);
        pStatementFuncionario.setString(1, "%" + nome + "%");
        ResultSet rsFuncionario =
pStatementFuncionario.executeQuery();
        while (rsFuncionario.next()) {
            Funcionario funcionario = new Funcionario();

            funcionario.setNomeCompleto(rsFuncionario.getString("NomeCompleto"
));
            funcionario.setEmail(rsFuncionario.getString("Email"));

            funcionario.setTelefone(rsFuncionario.getString("Telefone"));

```

```

        funcionario.setCargo(rsFuncionario.getString("Cargo"));
        usuarios.add(funcionario);
    }

} catch... fechamentos dos statements no bloco finally.
    return usuarios;
}

```

4.1. ServicoDAO

Exemplo de métodos da classe ServicosDAO. Utilizado visualizarServicos() que realiza um select na tabela serviços

```

public ArrayList<Servico> visualizarServicos(String input)
throws ExceptionDAO, SQLException {
    String sql = "SELECT * FROM SERVICOS WHERE Nome LIKE ?";
    PreparedStatement pStatement = null;
    Connection connection = null;
    ArrayList<Servico> servicos = new ArrayList<>();
    try {
        connection = new ConexaoBD().getConnection();
        pStatement = connection.prepareStatement(sql);
        pStatement.setString(1, "%" + input + "%");
        ResultSet rs = pStatement.executeQuery();

        while (rs.next()) {
            Servico serv = new
Servico(TipoServico.valueOf(rs.getString("TipoServico")),
rs.getString("Nome"), rs.getDouble("Valor"));

servicos.add(serv);
        }

    } finally {
        try {
            if (pStatement != null)
{pStatement.close();}
        } catch (SQLException e) {
            throw new ExceptionDAO("Erro ao fechar o
statement: " + e);
        }
    }
}

```

```

        if (connection != null)
    {connection.close();}
    } catch (SQLException e) {
        throw new ExceptionDAO("Erro ao fechar a
conexao do BD:" + e);
    }
}
return servicos;
}

public Servico consultaServico(String tipoServ) throws
ExceptionDAO {
    String sql = "SELECT TipoServico, Nome, Valor FROM
SERVICO WHERE TipoServico = ?";
    PreparedStatement pStatement = null;
    Connection connection = null;
    Servico serv = null;
    try {
        connection = new ConexaoBD().getConnection();
        pStatement = connection.prepareStatement(sql);
        pStatement.setString(1, tipoServ);
        ResultSet rs = pStatement.executeQuery();

        if (rs.next()) {
            serv = new
Servico(TipoServico.valueOf(rs.getString("TipoServico")),
            rs.getString("Nome"),
rs.getDouble("Valor"));
        }
    } catch... fechamento dos statements no bloco finally.

    return serv;
}

```

```

public void alterarServico(String tipoServ, Double
novoValor) throws ExceptionDAO {
    String sql = "UPDATE SERVICOS SET VALOR = ? WHERE
TipoServico = ?";
    PreparedStatement pStatement = null;
    Connection connection = null;
    try {
        connection = new ConexaoBD().getConnection();
        pStatement = connection.prepareStatement(sql);
        pStatement.setDouble(1, novoValor);
        pStatement.setString(2, tipoServ);

        int rowsAffected = pStatement.executeUpdate();

        if (rowsAffected == 0) {
            throw new ExceptionDAO("Nenhum serviço foi
alterado. Tipo de serviço não encontrado: " + tipoServ);
        }
    } catch (...) {
        // fechamento dos statements no bloco
        finally.
    }
}

```