

**FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA**

**TURMA 2TDSPS**

**ECORAL**

**JHONN BRANDON CABRERA TACACHIRI - RM97305**

**LEONARDO PAGANINI - RM96562**

**REGINA CÉLIA POMPEO BATISTA ALVES - RM97032**

**PROJETO E DESENVOLVIMENTO DE UM APP MOBILE PARA O  
MONITORAMENTO E TURISMO DE RECIFES DE CORAIS**

**SÃO PAULO**

**2024**

## **INTRODUÇÃO**

Este projeto representa uma abordagem inovadora e multifacetada para a interação com os ecossistemas marinhos, combinando tecnologia de ponta com a conservação ambiental. Ao integrar educação, monitoramento e turismo responsável, visa não apenas aumentar a conscientização sobre as condições dos recifes de corais, mas também promover a participação ativa na preservação desses preciosos habitats marinhos. Através de uma aplicação interativa, os usuários podem acessar em tempo real uma variedade de parâmetros aquáticos, fornecendo não apenas dados valiosos para pesquisa, mas também uma oportunidade única de aprendizado sobre a importância dos recifes de corais e as ameaças que enfrentam. Além disso, o projeto oferece uma experiência de mergulho educacional, onde os participantes não apenas desfrutam da beleza submarina, mas também contribuem para a restauração dos recifes por meio de técnicas inovadoras de fragmentação controlada. Este esforço conjunto de educação, monitoramento e conservação destaca o compromisso com a sustentabilidade e a preservação dos ecossistemas marinhos, ao mesmo tempo em que oferece uma experiência enriquecedora e transformadora para os envolvidos.

## **SOBRE**

O projeto em questão apresenta uma abordagem multifacetada para a interação com ecossistemas marinhos, combinando tecnologia e conservação ambiental. A aplicação proposta serve como uma ferramenta educacional e de monitoramento, oferecendo aos usuários a capacidade de verificar em tempo real uma variedade de parâmetros aquáticos, como temperatura, turbidez, condutividade elétrica, pH, nível de água e composição de H<sub>2</sub>O, todos medidos por sensores avançados. Esta funcionalidade não apenas aumenta a conscientização sobre as condições ambientais, mas também fornece dados valiosos para pesquisas e esforços de conservação.

Além disso, o aplicativo visa educar o público sobre a importância dos recifes de corais, abordando tópicos como biodiversidade, serviços ecossistêmicos e as ameaças que enfrentam, incluindo as pressões do turismo e as consequências da atividade humana. As páginas informativas são projetadas para informar os usuários sobre as melhores práticas para a preservação dos recifes de corais e os esforços de restauração em andamento no Brasil.

A proposta mais inovadora do projeto é a integração de uma experiência de turismo responsável com ações de conservação ativa. Os participantes do programa de mergulho receberão instruções detalhadas sobre a ecologia dos recifes e a

importância da preservação antes de sua imersão. Durante o mergulho, terão a oportunidade de contribuir para a restauração dos recifes através de um processo de fragmentação controlada, onde um pequeno fragmento de coral é coletado e, posteriormente, cultivado para regenerar e formar novos corais. Este método de fragmentação e regeneração é utilizada

para ajudar a acelerar a recuperação de ecossistemas de corais danificados, promovendo a diversidade genética e a resiliência dos recifes.

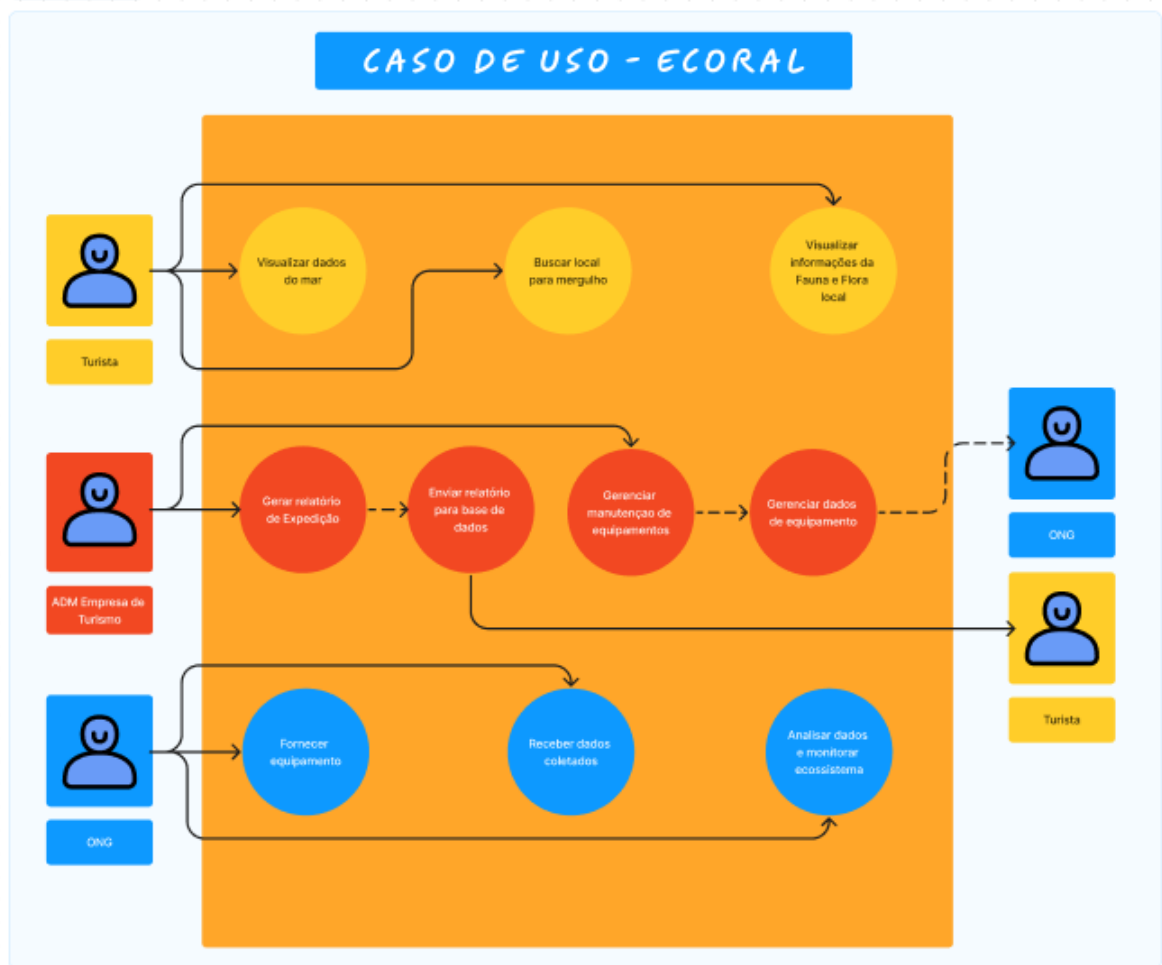
### **OBJETIVO**

O objetivo do ECORAL é promover a conservação e regeneração dos recifes de corais no Brasil. Para isso, integramos tecnologia de monitoramento ambiental com práticas de turismo sustentável. Nosso projeto visa proteger esses ecossistemas vitais, educar turistas e a comunidade local sobre a importância dos corais e envolvê-los diretamente na regeneração dos recifes, garantindo sua saúde e biodiversidade para as futuras gerações.

### **ARQUITETURA**

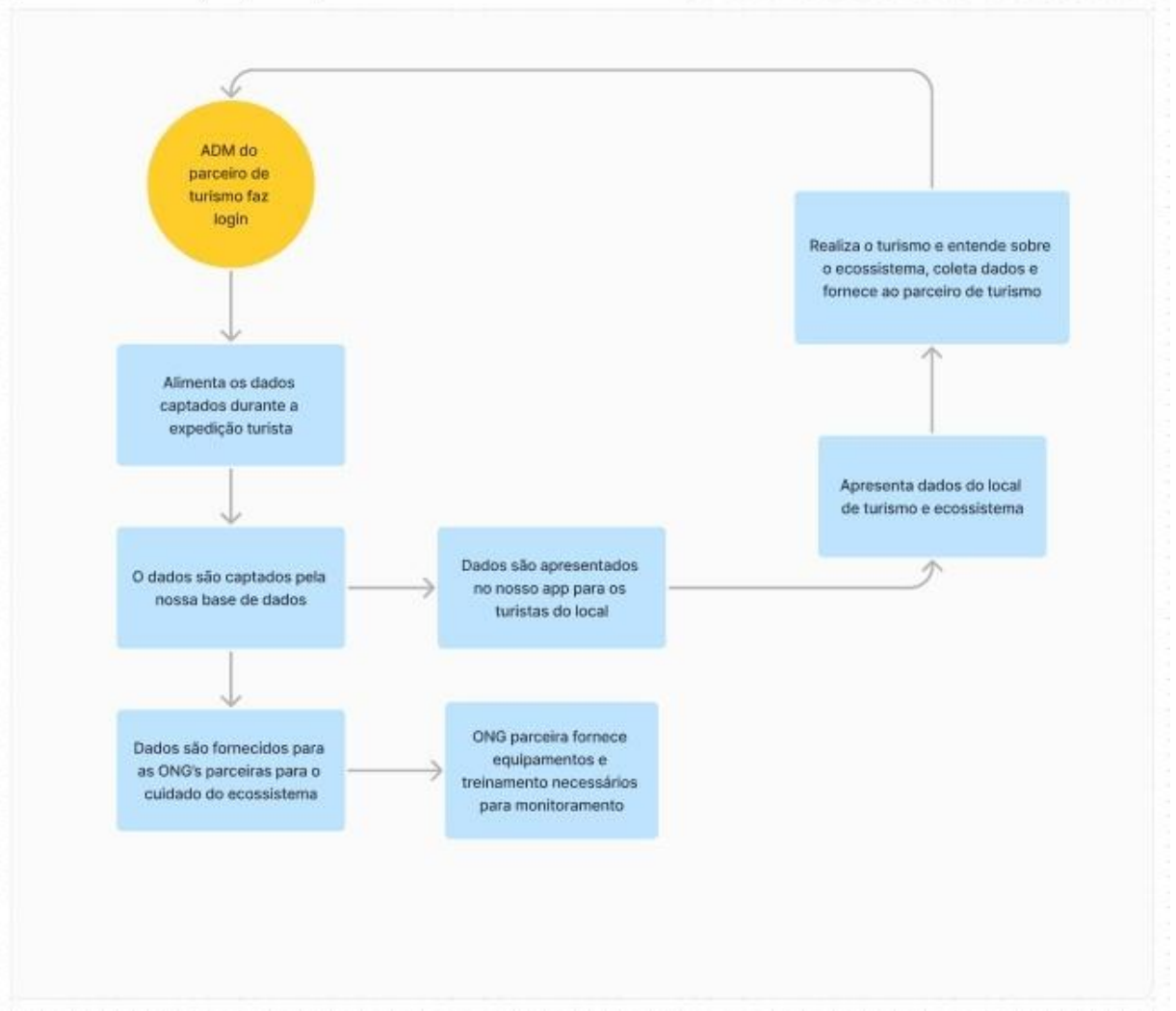
1. Documentação do Projeto:

## Use case

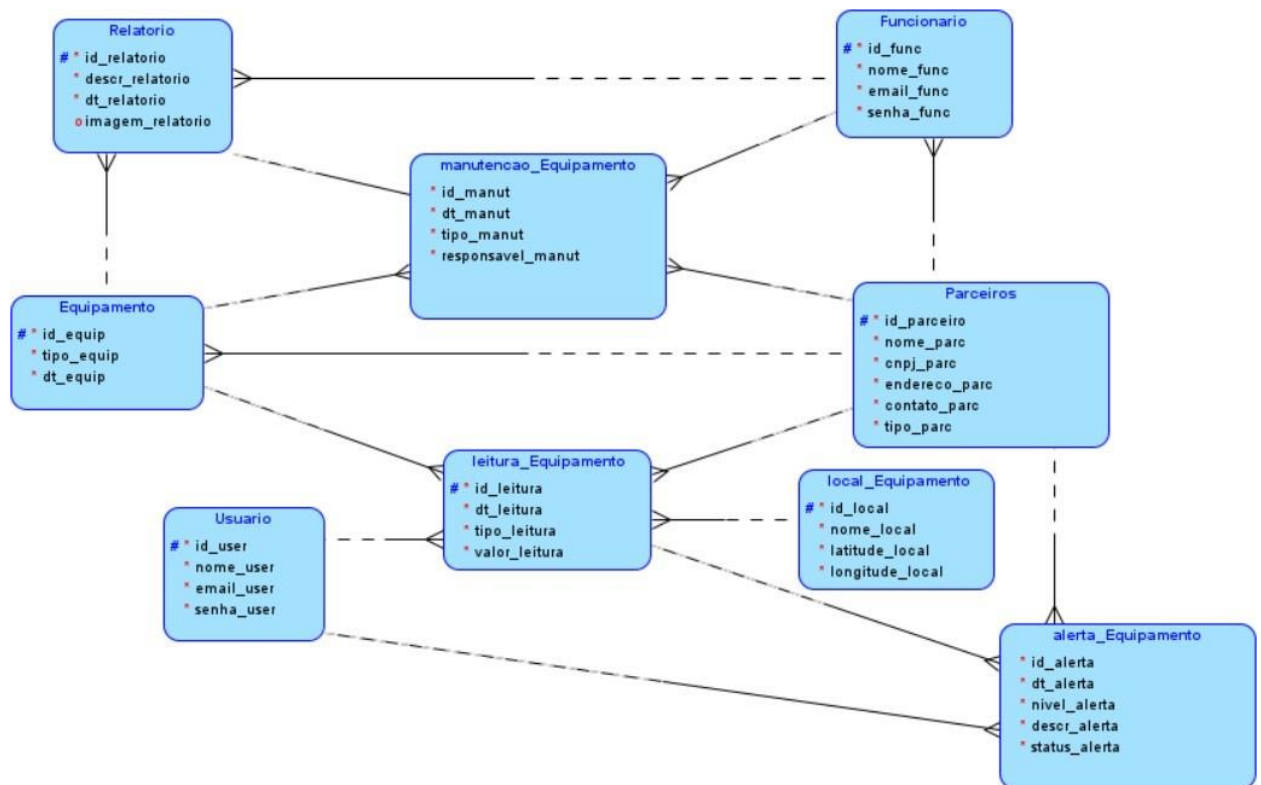


## 2. Fluxograma do projeto:

### ECOral: Qual o propósito para melhorar o ecossistema?



### 3. Modelo lógico do banco de dados:



## Descrição das entidades:

**Usuário** = Turista que pode utilizar o app para verificar dados do mar no local, verificar locais de turismo e ser direcionado para informações dos parceiros.

**Parceiros** = Empresas de Turismo e ONG's parceiras do projeto. Empresas de turismo tem seus funcionários que fazem os relatórios das expedições e realizam manutenções e trocas nos/dos equipamentos. ONG's tem propriedade dos equipamentos, então estão totalmente relacionados com informações de onde o equipamento está localizado e quem realizou a manutenção.

**Relatório** = Entidade onde o funcionário da empresa de turismo registra dados dos equipamentos, tais como local, data, qual tipo de manutenção realizada, quem realizou a manutenção e registra dados importantes para as ONG's acessarem.

**Equipamento** = Entidade que possui dados do equipamento fornecido pela ONG.

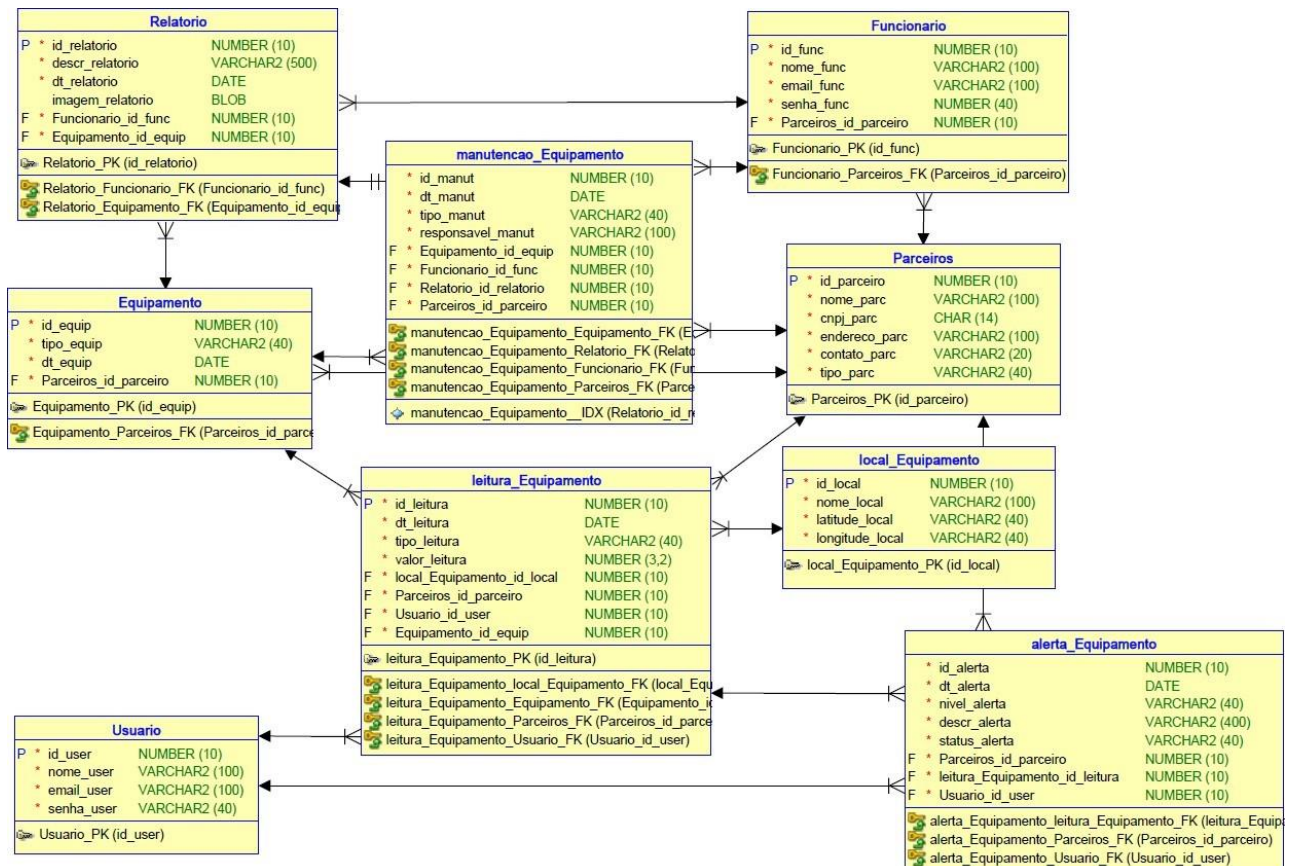
**Manutenção Equipamento** = Entidade que possui dados da manutenção realizada nas expedições.

**Leitura Equipamento** = Entidade que possui dados de monitoramento dos equipamentos, como temperatura, salinidade, o valor das medidas e datas da leitura.

**Local Equipamento** = Entidade que guarda dados do local onde o equipamento está instalado, com nome do local, longitude e latitude.

**Alerta Equipamento** = Entidade que está relacionada com a leitura do equipamento e que envia alerta dos dados após ser registrado, seria uma notificação informando a ultrapassagem dos limites de temperatura, por exemplo.

#### 4. Modelo Relacional:



## TABELAS E DADOS DE RELACIONAMENTO:

```
DROP TABLE ALERTA_EQUIPAMENTO cascade constraints;  
DROP TABLE EQUIPAMENTO CASCADE CONSTRAINTS;  
DROP TABLE FUNCIONARIO CASCADE CONSTRAINTS;  
DROP TABLE LEITURA_EQUIPAMENTO CASCADE CONSTRAINTS;  
DROP TABLE LOCAL_EQUIPAMENTO CASCADE CONSTRAINTS;  
DROP TABLE MANUTENCAO_EQUIPAMENTO CASCADE CONSTRAINTS;  
DROP TABLE PARCEIROS CASCADE CONSTRAINTS;  
DROP TABLE RELATORIO CASCADE CONSTRAINTS;  
DROP TABLE USUARIO CASCADE CONSTRAINTS;  
DROP TABLE AUDIT_LOG CASCADE CONSTRAINTS;|
```

```
CREATE TABLE alerta_equipamento (  
    id_alerta          NUMBER(10) NOT NULL,  
    dt_alerta          DATE NOT NULL,  
    nivel_alerta        VARCHAR2(40) NOT NULL,  
    descr_alerta        VARCHAR2(400) NOT NULL,  
    status_alerta       VARCHAR2(40) NOT NULL,  
    parceiros_id_parceiro NUMBER(10) NOT NULL,  
    leitura_equipamento_id_leitura NUMBER(10) NOT NULL,  
    usuario_id_user     NUMBER(10) NOT NULL  
);|
```

```
CREATE TABLE equipamento (  
    id_equip          NUMBER(10) NOT NULL,  
    tipo_equip        VARCHAR2(40) NOT NULL,  
    dt_equip          DATE NOT NULL,  
    parceiros_id_parceiro NUMBER(10) NOT NULL  
);  
  
ALTER TABLE equipamento ADD CONSTRAINT equipamento_pk PRIMARY KEY (id_equip);
```

```
CREATE TABLE funcionario (  
    id_func          NUMBER(10) NOT NULL,  
    nome_func        VARCHAR2(100) NOT NULL,  
    email_func       VARCHAR2(100) NOT NULL,  
    senha_func       VARCHAR2(40) NOT NULL,  
    parceiros_id_parceiro NUMBER(10) NOT NULL  
);  
  
ALTER TABLE funcionario ADD CONSTRAINT funcionario_pk PRIMARY KEY (id_func);
```



```
CREATE TABLE leitura_equipamento (  
    id_leitura          NUMBER(10) NOT NULL,  
    dt_leitura          DATE NOT NULL,  
    tipo_leitura        VARCHAR2(40) NOT NULL,  
    valor_leitura       NUMBER(3, 2) NOT NULL,  
    local_equipamento_id_local NUMBER(10) NOT NULL,  
    parceiros_id_parceiro NUMBER(10) NOT NULL,  
    usuario_id_user     NUMBER(10) NOT NULL,  
    equipamento_id Equip NUMBER(10) NOT NULL  
);
```

```
ALTER TABLE leitura_equipamento ADD CONSTRAINT leitura_equipamento_pk PRIMARY KEY (id_leitura);
```

```
CREATE TABLE local_equipamento (  
    id_local          NUMBER(10) NOT NULL,  
    nome_local        VARCHAR2(100) NOT NULL,  
    latitude_local    VARCHAR2(40) NOT NULL,  
    longitude_local   VARCHAR2(40) NOT NULL  
);
```

```
ALTER TABLE local_equipamento ADD CONSTRAINT local_equipamento_pk PRIMARY KEY (id_local);
```

```
CREATE TABLE manutencao_equipamento (  
    id_manut          NUMBER(10) NOT NULL,  
    dt_manut          DATE NOT NULL,  
    tipo_manut        VARCHAR2(40) NOT NULL,  
    responsavel_manut VARCHAR2(100) NOT NULL,  
    equipamento_id Equip NUMBER(10) NOT NULL,  
    funcionario_id_func NUMBER(10) NOT NULL,  
    relatorio_id_relatorio NUMBER(10) NOT NULL,  
    parceiros_id_parceiro NUMBER(10) NOT NULL  
);
```

```
CREATE UNIQUE INDEX manutencao_equipamento_idx ON manutencao_equipamento (relatorio_id_relatorio ASC);
```

```
CREATE TABLE parceiros (  
    id_parceiro      NUMBER(10) NOT NULL,  
    nome_parceiro    VARCHAR2(100) NOT NULL,  
    cnpj_parceiro     CHAR(14) NOT NULL,  
    endereco_parceiro VARCHAR2(100) NOT NULL,  
    contato_parceiro VARCHAR2(20) NOT NULL,  
    tipo_parceiro     VARCHAR2(40) NOT NULL  
);
```

```
ALTER TABLE parceiros ADD CONSTRAINT parceiros_pk PRIMARY KEY (id_parceiro);
```

```
CREATE TABLE relatorio (  
    id_relatorio      NUMBER(10) NOT NULL,  
    descr_relatorio   VARCHAR2(500) NOT NULL,  
    dt_relatorio      DATE NOT NULL,  
    imagem_relatorio  BLOB,  
    funcionario_id_func NUMBER(10) NOT NULL,  
    equipamento_id_equip NUMBER(10) NOT NULL  
);  
  
ALTER TABLE relatorio ADD CONSTRAINT relatorio_pk PRIMARY KEY (id_relatorio);
```

```
CREATE TABLE usuario (  
    id_user      NUMBER(10) NOT NULL,  
    nome_user    VARCHAR2(100) NOT NULL,  
    email_user   VARCHAR2(100) NOT NULL,  
    senha_user   VARCHAR2(40) NOT NULL  
);  
  
ALTER TABLE usuario ADD CONSTRAINT usuario_pk PRIMARY KEY (id_user);
```

```
CREATE TABLE audit_log (  
    id_log      NUMBER(10) PRIMARY KEY,  
    table_name  VARCHAR2(50) NOT NULL,  
    operation   VARCHAR2(10) NOT NULL,  
    altered_by  VARCHAR2(100) NOT NULL,  
    altered_on  DATE NOT NULL,  
    old_data    VARCHAR2(4000),  
    new_data    VARCHAR2(4000)  
);
```

```
DROP SEQUENCE SEQ_ID_USER;  
DROP SEQUENCE SEQ_ID_PARC;  
DROP SEQUENCE SEQ_ID_LEITURA_EQUIP;  
DROP SEQUENCE SEQ_ID_LOCAL_EQUIP;  
DROP SEQUENCE SEQ_ID_EQUIP;  
DROP SEQUENCE SEQ_ID_FUNC;  
DROP SEQUENCE SEQ_ID_RELAT;  
DROP SEQUENCE SEQ_ID_MANUT;  
DROP SEQUENCE SEQ_ID_ALERTA;  
DROP SEQUENCE SEQ_AUDIT_LOG;
```

```
CREATE SEQUENCE SEQ_ID_USER START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE SEQ_ID_PARC START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE SEQ_ID_LEITURA_EQUIP START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE SEQ_ID_LOCAL_EQUIP START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE SEQ_ID_EQUIP START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE SEQ_ID_FUNC START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE SEQ_ID_RELAT START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE SEQ_ID_MANUT START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE SEQ_ID_ALERTA START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE SEQ_AUDIT_LOG START WITH 1 INCREMENT BY 1;
```

```
ALTER TABLE alerta_equipamento
  ADD CONSTRAINT alerta_leitura_fk FOREIGN KEY (leitura_equipamento_id_leitura)
  REFERENCES leitura_equipamento (id_leitura);
```

```
ALTER TABLE alerta_equipamento
  ADD CONSTRAINT alerta_parceiro_fk FOREIGN KEY (parceiros_id_parceiro)
  REFERENCES parceiros (id_parceiro);
```

```
ALTER TABLE alerta_equipamento
  ADD CONSTRAINT alerta_usuario_fk FOREIGN KEY (usuario_id_user)
  REFERENCES usuario (id_user);
```

```
ALTER TABLE equipamento
  ADD CONSTRAINT equip_parceiro_fk FOREIGN KEY (parceiros_id_parceiro)
  REFERENCES parceiros (id_parceiro);
```

```
ALTER TABLE funcionario
  ADD CONSTRAINT func_parceiro_fk FOREIGN KEY (parceiros_id_parceiro)
  REFERENCES parceiros (id_parceiro);
```

```
ALTER TABLE leitura_equipamento
  ADD CONSTRAINT leitura Equip_fk FOREIGN KEY (equipamento_id_equip)
    REFERENCES equipamento (id_equip);

ALTER TABLE leitura_equipamento
  ADD CONSTRAINT leitura local_fk FOREIGN KEY (local_equipamento_id_local)
    REFERENCES local_equipamento (id_local);

ALTER TABLE leitura_equipamento
  ADD CONSTRAINT leitura parceiro_fk FOREIGN KEY (parceiros_id_parceiro)
    REFERENCES parceiros (id_parceiro);

ALTER TABLE leitura_equipamento
  ADD CONSTRAINT leitura usuario_fk FOREIGN KEY (usuario_id_user)
    REFERENCES usuario (id_user);

ALTER TABLE manutencao_equipamento
  ADD CONSTRAINT manut Equip_fk FOREIGN KEY (equipamento_id_equip)
    REFERENCES equipamento (id_equip);
```

```
ALTER TABLE manutencao_equipamento
  ADD CONSTRAINT manut func_fk FOREIGN KEY (funcionario_id_func)
    REFERENCES funcionario (id_func);

ALTER TABLE manutencao_equipamento
  ADD CONSTRAINT manut parceiro_fk FOREIGN KEY (parceiros_id_parceiro)
    REFERENCES parceiros (id_parceiro);

ALTER TABLE manutencao_equipamento
  ADD CONSTRAINT manut relatorio_fk FOREIGN KEY (relatorio_id_relatorio)
    REFERENCES relatorio (id_relatorio);

ALTER TABLE relatorio
  ADD CONSTRAINT relatorio Equip_fk FOREIGN KEY (equipamento_id_equip)
    REFERENCES equipamento (id_equip);

ALTER TABLE relatorio
  ADD CONSTRAINT relatorio func_fk FOREIGN KEY (funcionario_id_func)
    REFERENCES funcionario (id_func);
```

## PROCEDURES:

```
CREATE OR REPLACE PACKAGE pkg_usuario AS
  PROCEDURE inserir_usuario (
    p_id_user      NUMBER,
    p_nome_user    VARCHAR2,
    p_email_user   VARCHAR2,
    p_senha_user   VARCHAR2
  );
END pkg_usuario;
/
CREATE OR REPLACE PACKAGE BODY pkg_usuario AS
  PROCEDURE inserir_usuario (
    p_id_user      NUMBER,
    p_nome_user    VARCHAR2,
    p_email_user   VARCHAR2,
    p_senha_user   VARCHAR2
  ) IS
  BEGIN
    INSERT INTO usuario (
      id_user, nome_user, email_user, senha_user
    ) VALUES (
      p_id_user, p_nome_user, p_email_user, p_senha_user
    );
  EXCEPTION
    WHEN OTHERS THEN
      ROLLBACK;
      RAISE_APPLICATION_ERROR(-20009, 'Erro ao inserir_usuario: ' || SQLERRM);
  END inserir_usuario;
```

```
-- Pacote para a tabela PARCEIROS
CREATE OR REPLACE PACKAGE pkg_parceiros AS
  -- PROCEDURE QUE INSERE DADOS NA TABELA PARCEIROS
  PROCEDURE inserir_parceiros (
    p_id_parceiro  NUMBER,
    p_nome_parc    VARCHAR2,
    p_cnpj_parc    CHAR,
    p_endereco_par VARCHAR2,
    p_contato_par  VARCHAR2,
    p_tipo_parc    VARCHAR2
  );
END pkg_parceiros;
/
```

```

CREATE OR REPLACE PACKAGE BODY pkg_parceiros AS
  -- PROCEDURE QUE INSERE DADOS NA TABELA MANUTENCAO-EQUIPAMENTO
  PROCEDURE inserir_parceiros (
    p_id_parceiro    NUMBER,
    p_nome_parc      VARCHAR2,
    p_cnpj_parc      CHAR,
    p_endereco_parc  VARCHAR2,
    p_contato_parc   VARCHAR2,
    p_tipo_parc      VARCHAR2
  ) IS
  BEGIN
    INSERT INTO parceiros (
      id_parceiro, nome_parc, cnpj_parc, endereco_parc, contato_parc, tipo_parc
    ) VALUES (
      p_id_parceiro, p_nome_parc, p_cnpj_parc, p_endereco_parc, p_contato_parc, p_tipo_parc
    );
  EXCEPTION
    WHEN OTHERS THEN
      ROLLBACK;
      RAISE_APPLICATION_ERROR(-20007, 'Erro ao inserir parceiros: ' || SQLERRM);
  END inserir_parceiros;
END pkg_parceiros;
/

```

```

-- Pacote para a tabela LOCAL-EQUIPAMENTO
CREATE OR REPLACE PACKAGE pkg_local_equipamento AS
  -- PROCEDURE QUE INSERE DADOS NA TABELA LOCAL-EQUIPAMENTO
  PROCEDURE inserir_local_equipamento (
    p_id_local        NUMBER,
    p_nome_local       VARCHAR2,
    p_latitude_local   VARCHAR2,
    p_longitude_local  VARCHAR2
  );
END pkg_local_equipamento;
/

```

```

CREATE OR REPLACE PACKAGE BODY pkg_local_equipamento AS
    -- PROCEDURE QUE INSERE DADOS NA TABELA LOCAL_EQUIPAMENTO
    PROCEDURE inserir_local_equipamento (
        p_id_local          NUMBER,
        p_nome_local        VARCHAR2,
        p_latitude_local    VARCHAR2,
        p_longitude_local   VARCHAR2
    ) IS
    BEGIN
        INSERT INTO local_equipamento (
            id_local, nome_local, latitude_local, longitude_local
        ) VALUES (
            p_id_local, p_nome_local, p_latitude_local, p_longitude_local
        );
    EXCEPTION
        WHEN OTHERS THEN
            ROLLBACK;
            RAISE_APPLICATION_ERROR(-20005, 'Erro ao inserir local_equipamento: ' || SQLERRM);
    END inserir_local_equipamento;
END pkg_local_equipamento;
/

```

```

-- Pacote para a tabela EQUIPAMENTO
CREATE OR REPLACE PACKAGE pkg_equipamento AS
    -- PROCEDURE QUE INSERE DADOS NA TABELA EQUIPAMENTO
    PROCEDURE inserir_equipamento (
        p_id_equip          NUMBER,
        p_tipo_equip        VARCHAR2,
        p_dt_equip          DATE,
        p_parceiros_id_parceiro NUMBER
    );
END pkg_equipamento;
/

```

```

CREATE OR REPLACE PACKAGE BODY pkg_equipamento AS
    -- PROCEDURE QUE INSERE DADOS NA TABELA EQUIPAMENTO
    PROCEDURE inserir_equipamento (
        p_id_equip          NUMBER,
        p_tipo_equip        VARCHAR2,
        p_dt_equip          DATE,
        p_parceiros_id_parceiro NUMBER
    ) IS
    BEGIN
        INSERT INTO equipamento (
            id_equip, tipo_equip, dt_equip, parceiros_id_parceiro
        ) VALUES (
            p_id_equip, p_tipo_equip, p_dt_equip, p_parceiros_id_parceiro
        );
    EXCEPTION
        WHEN OTHERS THEN
            ROLLBACK;
            RAISE_APPLICATION_ERROR(-20002, 'Erro ao inserir equipamento: ' || SQLERRM);
    END inserir_equipamento;
END pkg_equipamento;
/

```

```

-- Pacote para a tabela LEITURA_EQUIPAMENTO
CREATE OR REPLACE PACKAGE pkg_leitura_equipamento AS
    -- PROCEDURE QUE INSERE DADOS NA TABELA LEITURA EQUIPAMENTO
    PROCEDURE inserir_leitura_equipamento (
        p_id_leitura          NUMBER,
        p_dt_leitura          DATE,
        p_tipo_leitura        VARCHAR2,
        p_valor_leitura        NUMBER,
        p_local_equipamento_id_local NUMBER,
        p_parceiros_id_parceiro NUMBER,
        p_usuario_id_user      NUMBER,
        p_equipamento_id_equip NUMBER
    );
END pkg_leitura_equipamento;
/

```



```

CREATE OR REPLACE PACKAGE BODY pkg_leitura_equipamento AS
-- PROCEDURE QUE INSERE DADOS NA TABELA LEITURA EQUIPAMENTO
PROCEDURE inserir_leitura_equipamento (
    p_id_leitura          NUMBER,
    p_dt_leitura          DATE,
    p_tipo_leitura        VARCHAR2,
    p_valor_leitura        NUMBER,
    p_local_equipamento_id_local NUMBER,
    p_parceiros_id_parceiro NUMBER,
    p_usuario_id_user      NUMBER,
    p_equipamento_id Equip NUMBER
) IS
BEGIN
    INSERT INTO leitura_equipamento (
        id_leitura, dt_leitura, tipo_leitura, valor_leitura, local_equipamento_id_local, parceiros_id_parceiro, usuario_id_user, equipamento_id_equip
    ) VALUES (
        p_id_leitura, p_dt_leitura, p_tipo_leitura, p_valor_leitura, p_local_equipamento_id_local, p_parceiros_id_parceiro, p_usuario_id_user, p_equipamento_id_equip
    );
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE_APPLICATION_ERROR(-20004, 'Erro ao inserir leitura-equipamento: ' || SQLERRM);
END inserir_leitura_equipamento;
END pkg_leitura_equipamento;
/

```

```

-- Pacote para a tabela FUNCIONARIO
CREATE OR REPLACE PACKAGE pkg_funcionario AS
-- PROCEDURE QUE INSERE DADOS NA TABELA FUNCIONARIO
PROCEDURE inserir_funcionario (
    p_id_func            NUMBER,
    p_nome_func          VARCHAR2,
    p_email_func         VARCHAR2,
    p_senha_func         NUMBER,
    p_parceiros_id_parceiro NUMBER
);
END pkg_funcionario;
/

```

```

CREATE OR REPLACE PACKAGE BODY pkg_funcionario AS
  -- PROCEDURE QUE INSERE DADOS NA TABELA FUNCIONARIO
  PROCEDURE inserir_funcionario (
    p_id_func          NUMBER,
    p_nome_func        VARCHAR2,
    p_email_func       VARCHAR2,
    p_senha_func       NUMBER,
    p_parceiros_id_parceiro NUMBER
  ) IS
  BEGIN
    INSERT INTO funcionario (
      id_func, nome_func, email_func, senha_func, parceiros_id_parceiro
    ) VALUES (
      p_id_func, p_nome_func, p_email_func, p_senha_func, p_parceiros_id_parceiro
    );
  EXCEPTION
    WHEN OTHERS THEN
      ROLLBACK;
      RAISE_APPLICATION_ERROR(-20003, 'Erro ao inserir funcionario: ' || SQLERRM);
  END inserir_funcionario;
END pkg_funcionario;
/

```

```

CREATE OR REPLACE PACKAGE pkg_relatorio AS
  -- PROCEDURE QUE INSERE DADOS NA TABELA RELATORIO
  PROCEDURE inserir_relatorio(
    p_id_relatorio      NUMBER,
    p_descr_relatorio   VARCHAR2,
    p_dt_relatorio      DATE,
    p_imagem_relatorio  BLOB,
    p_funcionario_id_func NUMBER,
    p_equipamento_id_equip NUMBER
  );
END pkg_relatorio;
/

```

```

CREATE OR REPLACE PACKAGE BODY pkg_relatorio AS
-- PROCEDURE QUE INSERE DADOS NA TABELA RELATORIO
PROCEDURE inserir_relatorio (
    p_id_relatorio      NUMBER,
    p_descr_relatorio   VARCHAR2,
    p_dt_relatorio      DATE,
    p_imagem_relatorio  BLOB,
    p_funcionario_id_func NUMBER,
    p Equipamento_id Equip NUMBER
) IS
BEGIN
    INSERT INTO relatorio (
        id_relatorio, descr_relatorio, dt_relatorio, imagem_relatorio, funcionario_id_func, equipamento_id Equip
    ) VALUES (
        p_id_relatorio, p_descr_relatorio, p_dt_relatorio, p_imagem_relatorio, p_funcionario_id_func, p Equipamento_id Equip
    );
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE_APPLICATION_ERROR(-20008, 'Erro ao inserir_relatorio: ' || SQLERRM);
END inserir_relatorio;
END pkg_relatorio;
/

```

```

-- Pacote para a tabela MANUTENCAO_EQUIPAMENTO
CREATE OR REPLACE PACKAGE pkg_manutencao_equipamento AS
-- PROCEDURE QUE INSERE DADOS NA TABELA MANUTENCAO_EQUIPAMENTO
PROCEDURE inserir_manutencao_equipamento (
    p_id_manut      NUMBER,
    p_dt_manut      DATE,
    p_tipo_manut    VARCHAR2,
    p_responsavel_manut VARCHAR2,
    p_equipamento_id Equip NUMBER,
    p_funcionario_id_func NUMBER,
    p_relatorio_id_relatorio NUMBER,
    p_parceiros_id_parceiro NUMBER
);
END pkg_manutencao_equipamento;
/

```

```

CREATE OR REPLACE PACKAGE BODY pkg_manutencao_equipamento AS
-- PROCEDURE QUE INSERE DADOS NA TABELA MANUTENCAO_EQUIPAMENTO
PROCEDURE inserir_manutencao_equipamento (
    p_id_manut      NUMBER,
    p_dt_manut      DATE,
    p_tipo_manut    VARCHAR2,
    p_responsavel_manut VARCHAR2,
    p_equipamento_id Equip NUMBER,
    p_funcionario_id_func NUMBER,
    p_relatorio_id_relatorio NUMBER,
    p_parceiros_id_parceiro NUMBER
) IS
BEGIN
    INSERT INTO manutencao_equipamento (
        id_manut, dt_manut, tipo_manut, responsavel_manut, equipamento_id Equip, funcionario_id_func, relatorio_id_relatorio, parceiros_id_parceiro
    ) VALUES (
        p_id_manut, p_dt_manut, p_tipo_manut, p_responsavel_manut, p_equipamento_id Equip, p_funcionario_id_func, p_relatorio_id_relatorio, p_parceiros_id_parceiro
    );
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE_APPLICATION_ERROR(-20006, 'Erro ao inserir_manutencao_equipamento: ' || SQLERRM);
END inserir_manutencao_equipamento;
END pkg_manutencao_equipamento;
/

```

```

-- Pacote para a tabela ALERTA-EQUIPAMENTO
CREATE OR REPLACE PACKAGE pkg_alerta_equipamento AS
    -- PROCEDURE QUE INSERE DADOS NA TABELA ALERTA-EQUIPAMENTO
    PROCEDURE inserir_alerta_equipamento (
        p_id_alerta          NUMBER,
        p_dt_alerta          DATE,
        p_nivel_alerta       VARCHAR2,
        p_descr_alerta       VARCHAR2,
        p_status_alerta      VARCHAR2,
        p_parceiros_id_parceiro NUMBER,
        p_leitura_equipamento_id_leitura NUMBER,
        p_usuario_id_user    NUMBER
    );
END pkg_alerta_equipamento;
/

```

```

CREATE OR REPLACE PACKAGE BODY pkg_alerta_equipamento AS
    -- PROCEDURE QUE INSERE DADOS NA TABELA ALERTA-EQUIPAMENTO
    PROCEDURE inserir_alerta_equipamento (
        p_id_alerta          NUMBER,
        p_dt_alerta          DATE,
        p_nivel_alerta       VARCHAR2,
        p_descr_alerta       VARCHAR2,
        p_status_alerta      VARCHAR2,
        p_parceiros_id_parceiro NUMBER,
        p_leitura_equipamento_id_leitura NUMBER,
        p_usuario_id_user    NUMBER
    ) IS
    BEGIN
        INSERT INTO alerta_equipamento (
            id_alerta, dt_alerta, nivel_alerta, descr_alerta, status_alerta, parceiros_id_parceiro, leitura_equipamento_id_leitura, usuario_id_user
        ) VALUES (
            p_id_alerta, p_dt_alerta, p_nivel_alerta, p_descr_alerta, p_status_alerta, p_parceiros_id_parceiro, p_leitura_equipamento_id_leitura, p_usuario_id_user
        );
    EXCEPTION
        WHEN OTHERS THEN
            ROLLBACK;
            RAISE_APPLICATION_ERROR(-20001, 'Erro ao inserir alerta-equipamento: ' || SQLERRM);
    END inserir_alerta_equipamento;
END pkg_alerta_equipamento;
/

```

## TRIGGERS:

```
-- TRIGGER MONITORANDO INSERTS DA TABELA FUNCIONARIO
CREATE OR REPLACE TRIGGER trg_audit_funcionario
AFTER INSERT OR UPDATE OR DELETE ON funcionario
FOR EACH ROW
DECLARE
    v_old_data VARCHAR2(4000);
    v_new_data VARCHAR2(4000);
BEGIN
    IF INSERTING THEN
        v_new_data := 'id_func=' || :NEW.id_func ||
            ', nome_func=' || :NEW.nome_func ||
            ', email_func=' || :NEW.email_func ||
            ', senha_func=' || :NEW.senha_func ||
            ', parceiros_id_parceiro=' || :NEW.parceiros_id_parceiro;

        INSERT INTO audit_log (
            id_log, table_name, operation, altered_by, altered_on, old_data, new_data
        ) VALUES (
            SEQ_AUDIT_LOG.NEXTVAL, 'funcionario', 'INSERT', 'system_user', SYSDATE, NULL, v_new_data
        );
    ELSIF UPDATING THEN -- DEIXAMOS OS OUTROS METODOS CRUD PARA MELHORIAS, POREM NAO SERAO USADOS AGORA
        v_old_data := 'id_func=' || :OLD.id_func ||
            ', nome_func=' || :OLD.nome_func ||
            ', email_func=' || :OLD.email_func ||
            ', senha_func=' || :OLD.senha_func ||
            ', parceiros_id_parceiro=' || :OLD.parceiros_id_parceiro;

        v_new_data := 'id_func=' || :NEW.id_func ||
            ', nome_func=' || :NEW.nome_func ||
            ', email_func=' || :NEW.email_func ||
            ', senha_func=' || :NEW.senha_func ||
            ', parceiros_id_parceiro=' || :NEW.parceiros_id_parceiro;

        INSERT INTO audit_log (
            id_log, table_name, operation, altered_by, altered_on, old_data, new_data
        ) VALUES (
            SEQ_AUDIT_LOG.NEXTVAL, 'funcionario', 'UPDATE', 'system_user', SYSDATE, v_old_data, v_new_data
        );
    ELSIF DELETING THEN -- DEIXAMOS OS OUTROS METODOS CRUD PARA MELHORIAS, POREM NAO SERAO USADOS AGORA
        v_old_data := 'id_func=' || :OLD.id_func ||
            ', nome_func=' || :OLD.nome_func ||
            ', email_func=' || :OLD.email_func ||
            ', senha_func=' || :OLD.senha_func ||
            ', parceiros_id_parceiro=' || :OLD.parceiros_id_parceiro;

        INSERT INTO audit_log (
            id_log, table_name, operation, altered_by, altered_on, old_data, new_data
        ) VALUES (
            SEQ_AUDIT_LOG.NEXTVAL, 'funcionario', 'DELETE', 'system_user', SYSDATE, v_old_data, NULL
        );
    END IF;
END;
```

```

CREATE OR REPLACE TRIGGER trg_audit_usuario
AFTER INSERT OR UPDATE OR DELETE ON usuario
FOR EACH ROW
DECLARE
    v_old_data VARCHAR2(4000);
    v_new_data VARCHAR2(4000);
BEGIN
    IF INSERTING THEN
        v_new_data := 'id_user=' || :NEW.id_user ||
            ', nome_user=' || :NEW.nome_user ||
            ', email_user=' || :NEW.email_user ||
            ', senha_user=' || :NEW.senha_user;
        INSERT INTO audit_log (
            id_log, table_name, operation, altered_by, altered_on, old_data, new_data
        ) VALUES (
            SEQ_AUDIT_LOG.NEXTVAL, 'usuario', 'INSERT', 'system_user', SYSDATE, NULL, v_new_data
        );
    ELSIF UPDATING THEN
        v_old_data := 'id_user=' || :OLD.id_user ||
            ', nome_user=' || :OLD.nome_user ||
            ', email_user=' || :OLD.email_user ||
            ', senha_user=' || :OLD.senha_user;
        v_new_data := 'id_user=' || :NEW.id_user ||
            ', nome_user=' || :NEW.nome_user ||
            ', email_user=' || :NEW.email_user ||
            ', senha_user=' || :NEW.senha_user;
        INSERT INTO audit_log (
            id_log, table_name, operation, altered_by, altered_on, old_data, new_data
        ) VALUES (
            SEQ_AUDIT_LOG.NEXTVAL, 'usuario', 'UPDATE', 'system_user', SYSDATE, v_old_data, v_new_data
        );
    ELSIF DELETING THEN
        v_old_data := 'id_user=' || :OLD.id_user ||
            ', nome_user=' || :OLD.nome_user ||
            ', email_user=' || :OLD.email_user ||
            ', senha_user=' || :OLD.senha_user;
        INSERT INTO audit_log (
            id_log, table_name, operation, altered_by, altered_on, old_data, new_data
        ) VALUES (
            SEQ_AUDIT_LOG.NEXTVAL, 'usuario', 'DELETE', 'system_user', SYSDATE, v_old_data, NULL
        );
    END IF;
END;
/

```