

FIAP

ANALISE E DESENVOLVIMENTO DE SISTEMAS

FELIPE DE LIMA SANTIAGO

MATHEUS RAMOS DE PIERRO

THIAGO GYORGY TEIXEIRA DE CASTRO

VICTOR SHIMADA SERETE

GABRIEL TRICERRI ANDRÉ NIACARIS

Database Application Development



Introdução:

O aumento das demandas relacionadas à saúde mental ressalta a necessidade de soluções inovadoras e acessíveis. O aplicativo PsyCheck é uma resposta estratégica da Notredame para oferecer suporte eficaz à saúde mental, promovendo a prevenção, tornando a assistência médica mais acessível e precisa, e fomentando a saúde geral dos usuários. Tudo isso de forma acessível.

2. Descrição do Problema/Foco de Problema a Resolver

O cenário atual de saúde mental apresenta desafios significativos, com o aumento das taxas de estresse, ansiedade e depressão. A busca por assistência muitas vezes é dificultada pela falta de acesso a recursos adequados. Além disso, a detecção precoce e a prevenção eficaz são essenciais para reduzir o impacto das doenças mentais.

3. Descrição da Solução Idealizada

O PsyCheck é um aplicativo gratuito de saúde mental que integra tecnologias emergentes para oferecer uma abordagem abrangente e inovadora. As principais características incluem:

3.1. Detecção Precoce por meio de IA:

- Utilização de algoritmos de inteligência artificial (IA) para analisar padrões de comportamento e identificar sinais precoces de problemas de saúde mental.
- Análise contínua dos dados de uso do aplicativo para personalizar as intervenções e fornecer suporte personalizado.

3.2. Automação de Suporte:

- Implementação de chatbots e assistentes virtuais para oferecer suporte emocional e fornecer informações úteis de forma instantânea.
- Recursos de automação para enviar lembretes personalizados, atividades de relaxamento e estratégias de enfrentamento.

3.3. Análise de Dados para Personalização:

- Coleta de dados sobre o usuário, como padrões de sono, níveis de atividade física e interações no aplicativo.
- Utilização de análise de dados para personalizar recomendações e intervenções, garantindo uma abordagem adaptativa.

3.4. Acessibilidade Aprimorada:

- Interface intuitiva e design centrado no usuário para garantir acessibilidade a todas as faixas etárias.
- Disponibilidade em múltiplos idiomas e integração com tecnologias de acessibilidade para atender a diversos perfis de usuários.

3.5. Colaboração com Profissionais de Saúde:

- Facilitação da comunicação entre usuários e profissionais de saúde mental.
- Compartilhamento seguro de dados relevantes para permitir uma abordagem colaborativa no tratamento.

4. Benefícios Esperados

- **Prevenção Eficiente:** Identificação precoce de problemas de saúde mental para intervenção rápida.
- **Acessibilidade:** Disponibilidade a qualquer momento, em qualquer lugar, promovendo a acessibilidade a uma ampla gama de usuários.
- **Personalização:** Oferta de suporte personalizado com base em dados e padrões específicos de cada usuário.
- **Colaboração Integrada:** Facilitação da colaboração entre usuários e profissionais de saúde mental.

5. Considerações Éticas e de Segurança

- **Privacidade:** Implementação de protocolos robustos para proteção de dados do usuário.
- **Consentimento Informado:** Garantia de que os usuários tenham informações claras e deem consentimento informado para o uso de dados.
- **Intervenção Humana:** Manutenção de canais para intervenção humana em casos de emergência ou quando a IA não pode oferecer suporte adequado.

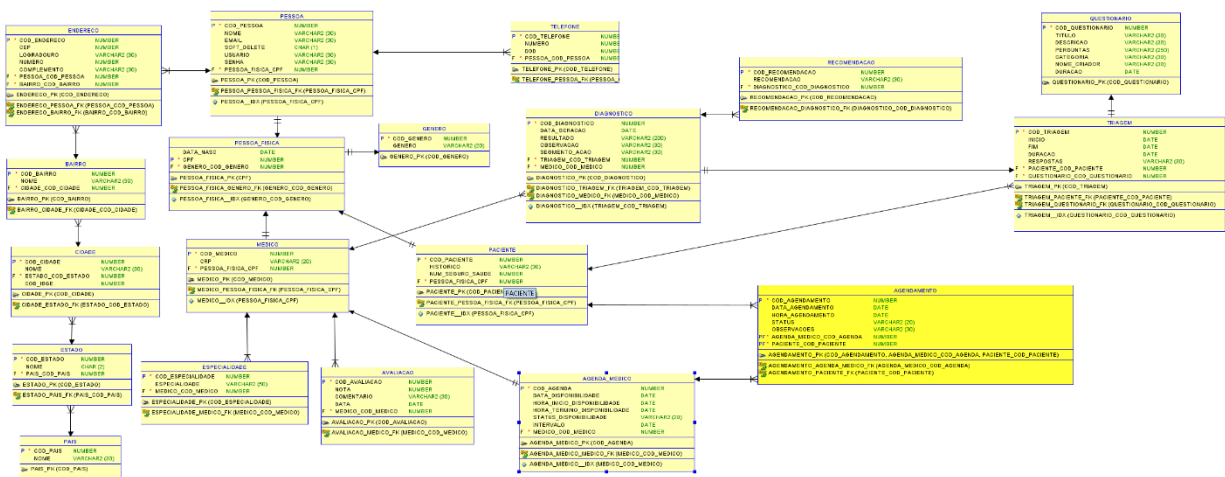
5.1 Forma de Monetização

- Através do intermédio de agendamento de consultas entre os paciente e os profissionais, será aplicada uma taxa para a consulta gerando um percentual de lucro para nossa plataforma.

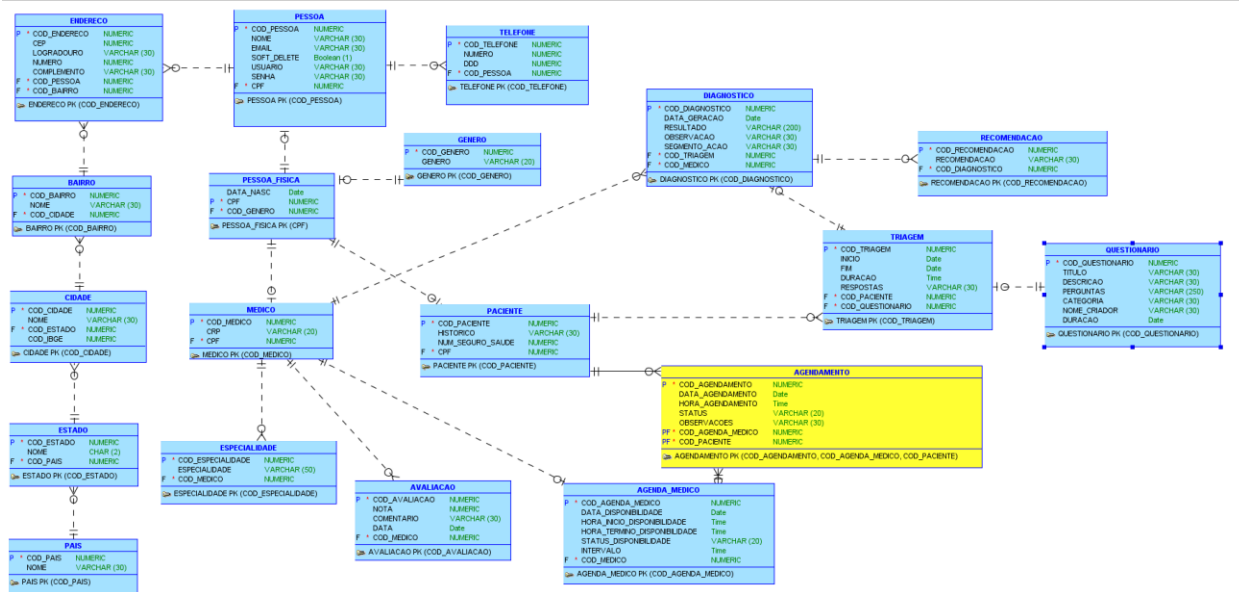
6. Conclusão

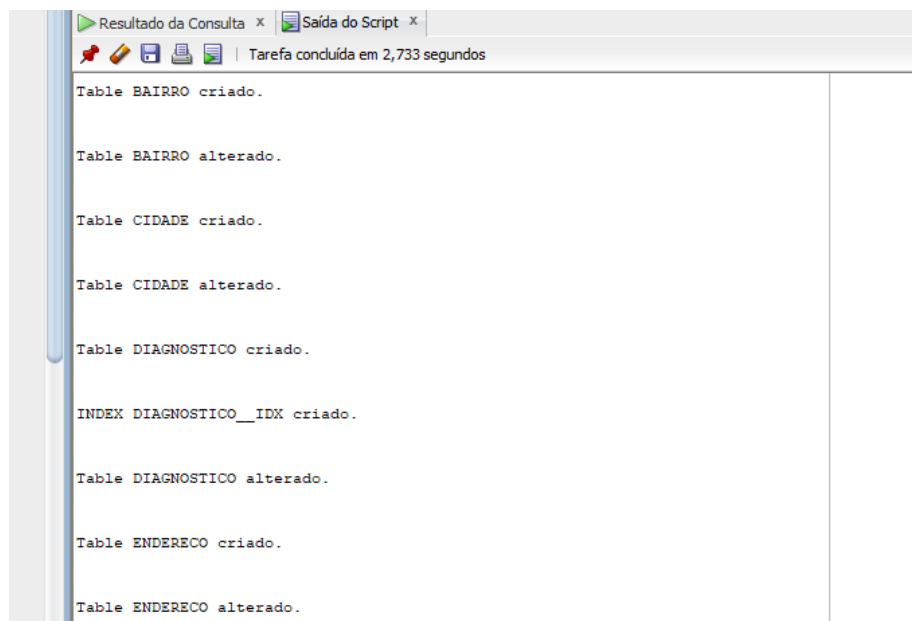
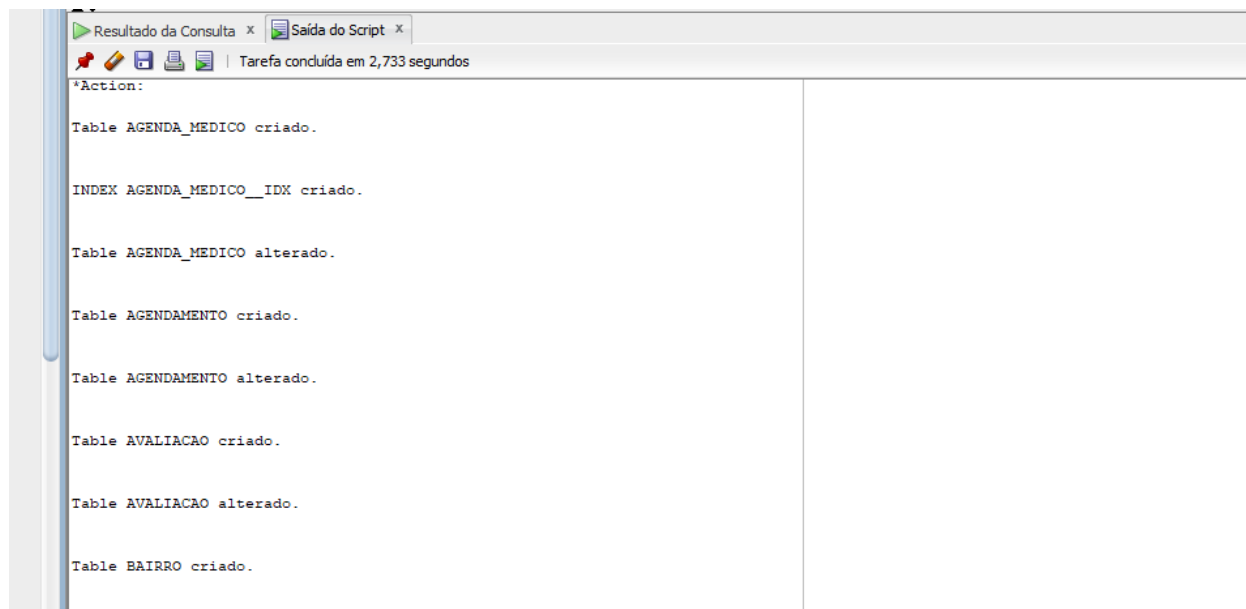
O PsyCheck representa um avanço inovador no cuidado com a saúde mental, oferecendo uma abordagem holística e acessível. Ao integrar tecnologias emergentes, o aplicativo visa promover a prevenção, tornar a assistência médica mais acessível e precisa, e incentivar a saúde mental de maneira geral, alinhando-se aos objetivos estratégicos da Notredame.

Modelo Relacional:



Modelo Físico





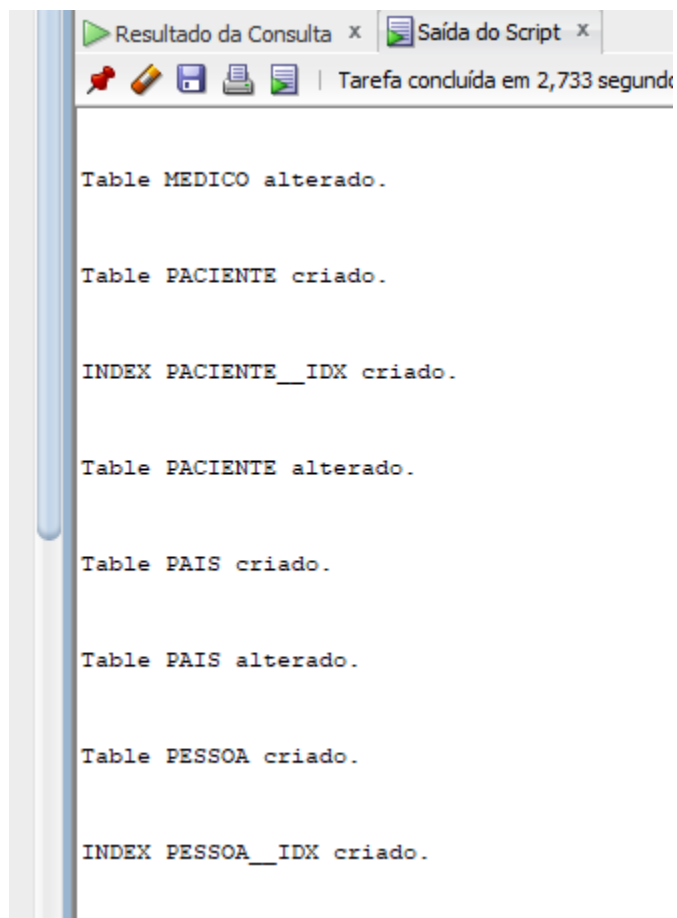
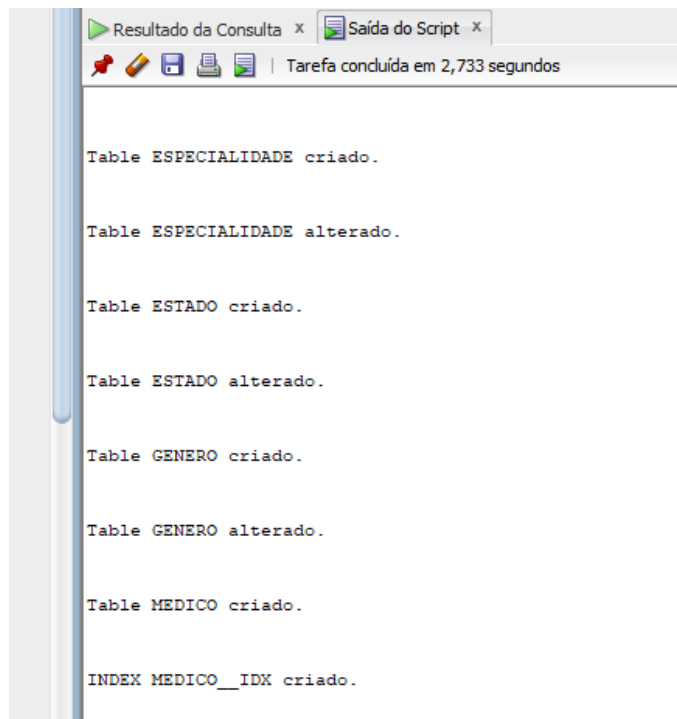


Tabela Genero

```
--Genero--
DECLARE
    COD_GENERO    NUMBER(10) := &COD_GENERO;
    GENERO        VARCHAR2(20) := '& GENERO';
BEGIN
    BEGIN
        INSERT INTO genero
            VALUES (COD_GENERO, GENERO );

        DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

Script Output x

Task completed in 15.928 seconds

```
DBMS_OUTPUT.PUT_LINE('Registro de farmacia inserido com sucesso.');
```

```
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');
```

```
END;
END;
```

PL/SQL procedure successfully completed.

Tabela Pessoa Fisica

```
--Pessoa_fisica--
DECLARE
    DATA_NASC    DATE := '& DATA_NASC';
    CPF           NUMBER := &CPF;
    GENERO_COD_GENERO NUMBER := &GENERO_COD_GENERO;
BEGIN
    BEGIN
        INSERT INTO pessoa_fisica
            VALUES (DATA_NASC, CPF, GENERO_COD_GENERO);

        DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

Script Output x Query Result x

Task completed in 9.809 seconds

```
DBMS_OUTPUT.PUT_LINE('Registro de farmacia inserido com sucesso.');
```

```
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');
```

```
END;
END;
```

PL/SQL procedure successfully completed.

Tabela Pessoa

The screenshot displays the SQL Developer interface with the 'Query Builder' tab active. The main editor shows a PL/SQL procedure named 'Pessoa' that declares variables for personal information and inserts a record into the 'pessoa' table. The 'Script Output' window at the bottom shows the successful execution of the procedure.

```
--Pessoa--  
DECLARE  
  COD_PESSOA NUMBER := &COD_PESSOA;  
  NOME        VARCHAR2(30) := '&NOME';  
  EMAIL       VARCHAR2(30) := '&EMAIL';  
  SOFT_DELETE CHAR(1) := '&SOFT_DELETE';  
  USUARIO     VARCHAR2(30) := '&USUARIO';  
  SENHA       VARCHAR2(30) := '&SENHA';  
  PESSOA_FISICA_CPF NUMBER := &PESSOA_FISICA_CPF;  
  
BEGIN  
  BEGIN  
    INSERT INTO pessoa  
    VALUES (COD_PESSOA, NOME, EMAIL, SOFT_DELETE, USUARIO, SENHA, PESSOA_FISICA_CPF );  
  
    DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');  EXCEPTION  
    WHEN DUP_VAL_ON_INDEX THEN  
      DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');  END;  
END;
```

Script Output x
Task completed in 45.126 seconds

```
DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
 DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');

END;
END;

PL/SQL procedure successfully completed.

Tabela Telefone

The screenshot displays the SQL Developer interface with the 'Query Builder' tab active. The main editor shows a PL/SQL procedure named 'Telefone' that declares variables for phone information and inserts a record into the 'telefone' table. The 'Script Output' window at the bottom shows the successful execution of the procedure.

```
DECLARE  
  COD_TELEFONE NUMBER := &COD_TELEFONE;  
  NUMERO        NUMBER := &NUMERO;  
  DDD           NUMBER := &DDD;  
  PESSOA_COD_PESSOA NUMBER := &PESSOA_COD_PESSOA;  
  
BEGIN  
  BEGIN  
    INSERT INTO telefone  
    VALUES (COD_TELEFONE, NUMERO, DDD, PESSOA_COD_PESSOA);  
  
    DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');  EXCEPTION  
    WHEN DUP_VAL_ON_INDEX THEN  
      DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');  END;  
END;
```

Script Output x
Task completed in 8.052 seconds

```
DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
 DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');

END;
END;

PL/SQL procedure successfully completed.

Tabela Pais

```
--Pais--
DECLARE
    COD_PAIS    NUMBER := &COD_PAIS ;
    NOME        VARCHAR2(30) := '&NOME' ;
BEGIN
    BEGIN
        INSERT INTO pais
        VALUES (COD_PAIS, NOME);

        DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

Script Output x

Task completed in 6.208 seconds

```
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');
```

```
END;
END;
```

```
DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

```
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');
```

```
END;
END;
```

PL/SQL procedure successfully completed.

Tabela Estado;

```
--ESTADO--
DECLARE
    COD_ESTADO  NUMBER := &COD_ESTADO ;
    NOME        VARCHAR2(30) := '&NOME' ;
    PAIS_COD_PAIS NUMBER := &PAIS_COD_PAIS ;
BEGIN
    BEGIN
        INSERT INTO estado
        VALUES (COD_ESTADO, NOME, PAIS_COD_PAIS);

        DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

Worksheet Query Builder

Script Output x Query Result x

Task completed in 4.197 seconds

```
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');
```

```
END;
END;
```

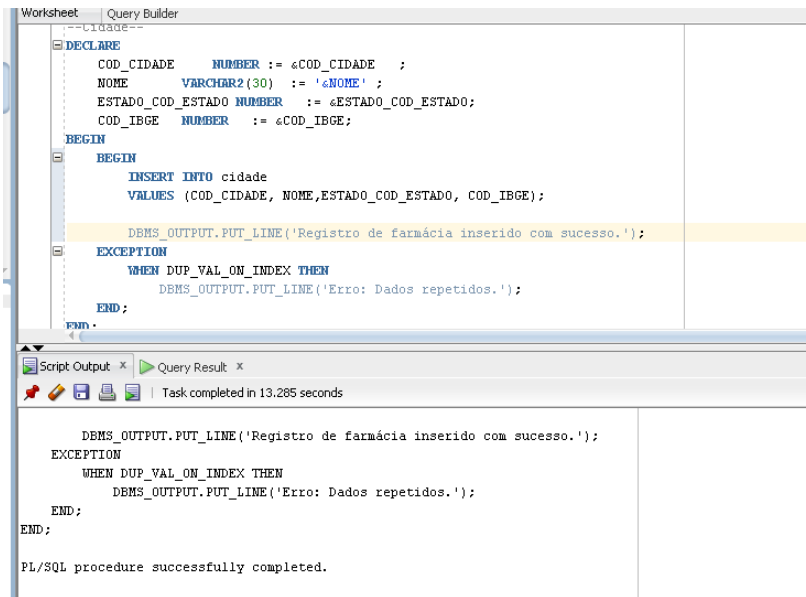
```
DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

```
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');
```

```
END;
END;
```

PL/SQL procedure successfully completed.

Tabela Cidade;



```
--Cidade--
DECLARE
  COD_CIDADE      NUMBER := &COD_CIDADE ;
  NOME            VARCHAR2(30) := '&NOME' ;
  ESTADO_COD_ESTADO NUMBER := &ESTADO_COD_ESTADO;
  COD_IBGE        NUMBER := &COD_IBGE;
BEGIN
  BEGIN
    INSERT INTO cidade
    VALUES (COD_CIDADE, NOME, ESTADO_COD_ESTADO, COD_IBGE);

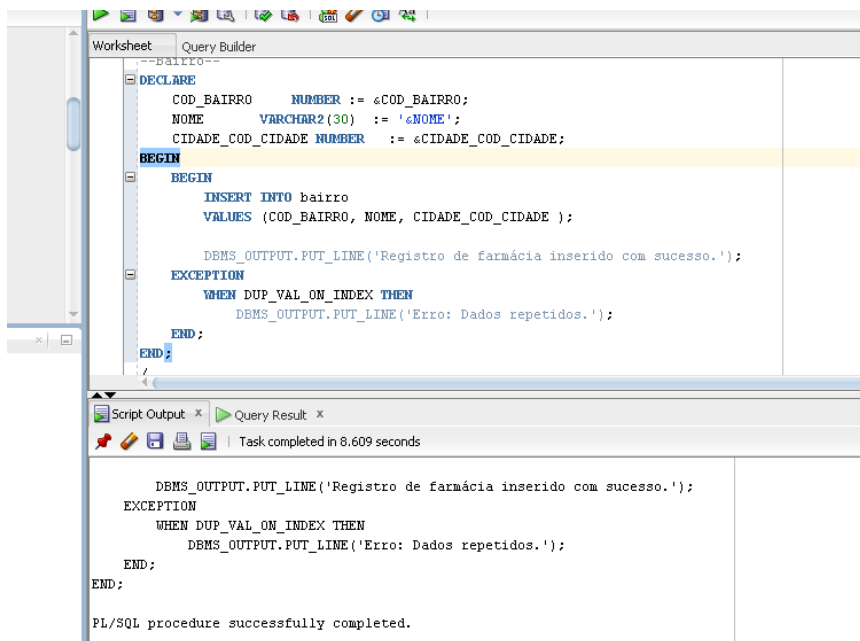
    DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

Task completed in 13.285 seconds

```
END;
END;
```

PL/SQL procedure successfully completed.

Tabela Bairro



```
--Bairro--
DECLARE
  COD_BAIRRO      NUMBER := &COD_BAIRRO;
  NOME            VARCHAR2(30) := '&NOME' ;
  CIDADE_COD_CIDADE NUMBER := &CIDADE_COD_CIDADE;
BEGIN
  BEGIN
    INSERT INTO bairro
    VALUES (COD_BAIRRO, NOME, CIDADE_COD_CIDADE );

    DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

Task completed in 8.609 seconds

```
END;
END;
```

PL/SQL procedure successfully completed.

Tabela Endereco

```
--Endereco--
DECLARE
    COD_ENDERECO    NUMBER := &COD_ENDERECO ;
    CEP             NUMBER := &CEP;
    LOGRADOURO      VARCHAR2(30) := '&LOGRADOURO';
    NUMERO          NUMBER := &NUMERO;
    COMPLEMENTO     VARCHAR2(30) := '&COMPLEMENTO';
    PESSOA_COD_PESSOA NUMBER := &PESSOA_COD_PESSOA;
    BAIRRO_COD_BAIRRO NUMBER := &BAIRRO_COD_BAIRRO;
BEGIN
    BEGIN
        INSERT INTO endereco
        VALUES (COD_ENDERECO, CEP, LOGRADOURO, NUMERO, COMPLEMENTO, PESSOA_COD_PESSOA, BAIRRO_COD_BAIRRO );

        DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

Script Output x Query Result x

Task completed in 11.921 seconds

```
        DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

```
    EXCEPTION
        WHEN DUP_VAL_ON_INDEX THEN
            DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');
```

```
    END;
END;
```

PL/SQL procedure successfully completed.

Tabela medico

```
--Medico--
DECLARE
    COD_MEDICO      NUMBER := &COD_MEDICO;
    CRP             VARCHAR2(20) := '&CRP';
    PESSOA_FISICA_CPF NUMBER := &PESSOA_FISICA_CPF;
BEGIN
    BEGIN
        INSERT INTO medico
        VALUES ( COD_MEDICO, CRP, PESSOA_FISICA_CPF );

        DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

Script Output x Query Result x

Task completed in 9.137 seconds

```
    EXCEPTION
        WHEN DUP_VAL_ON_INDEX THEN
            DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');
```

```
    END;
END;
```

PL/SQL procedure successfully completed.

Tabela Especialidade

```
--Especialidade--
DECLARE
    COD_ESPECIALIDADE    NUMBER := &COD_ESPECIALIDADE;
    ESPECIALIDADE        VARCHAR2(20) := '&ESPECIALIDADE';
    MEDICO_COD_MEDICO    NUMBER := &MEDICO_COD_MEDICO;
BEGIN
    BEGIN
        INSERT INTO especialidade
        VALUES (COD_ESPECIALIDADE,ESPECIALIDADE ,MEDICO_COD_MEDICO);

        DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

Task completed in 8.752 seconds

```
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');
```

PL/SQL procedure successfully completed.

Tabela avaliacao

```
--Avaliacao--
DECLARE
    COD_AVALIACAO    NUMBER := &COD_AVALIACAO ;
    NOTA            NUMBER := &NOTA;
    COMENTARIO      VARCHAR2(30) := '&COMENTARIO';
    DATA date      := '&DATA';
    MEDICO_COD_MEDICO    NUMBER := &MEDICO_COD_MEDICO;
BEGIN
    BEGIN
        INSERT INTO avaliacao
        VALUES (COD_AVALIACAO, NOTA, COMENTARIO, DATA , MEDICO_COD_MEDICO );

        DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

Task completed in 9.946 seconds

```
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');
```

PL/SQL procedure successfully completed.

Tabela paciente

```
--Paciente--
DECLARE
    COD_PACIENTE      NUMBER := &COD_PACIENTE;
    HISTORICO          VARCHAR2(30) := '&HISTORICO ';
    NUM_SEGURO_SAUDE   NUMBER := '&NUM_SEGURO_SAUDE';
    PESSOA_FISICA_CPF  NUMBER := '&PESSOA_FISICA_CPF';
BEGIN
    BEGIN
        INSERT INTO paciente
            VALUES (COD_PACIENTE, HISTORICO, NUM_SEGURO_SAUDE, PESSOA_FISICA_CPF );

        DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

Script Output x Query Result x

Task completed in 6.597 seconds

```
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');
```

END;

PL/SQL procedure successfully completed.

Tabela Agendamento

```
--Agendamento--
DECLARE
    COD_AGENDAMENTO    NUMBER := &COD_AGENDAMENTO;
    DATA_AGENDAMENTO  date := '&DATA_AGENDAMENTO';
    HORA_AGENDAMENTO   date := '&HORA_AGENDAMENTO ';
    STATUS              VARCHAR2(20) := '&STATUS';
    OBSERVACOES         varchar(30) := '&OBSERVACOES';
    AGENDA_MEDICO_COD_AGENDA number := &AGENDA_MEDICO_COD_AGENDA;
    PACIENTE_COD_PACIENTE number := &PACIENTE_COD_PACIENTE;
BEGIN
    BEGIN
        INSERT INTO agendamento
            VALUES (COD_AGENDAMENTO, DATA_AGENDAMENTO, HORA_AGENDAMENTO, STATUS, OBSERVACOES,
```

Saída do Script x

Tarefa concluída em 30,243 segundos

```
        DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

EXCEPTION

```
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');
```

END;

END;

Tabela questionario

```
--Questionario--
DECLARE
    COD_QUESTIONARIO    NUMBER := &COD_QUESTIONARIO;
    TITULO               VARCHAR2(30) := '&TITULO';
    DESCRICAO            VARCHAR2(30) := '&DESCRICAO';
    PERGUNTAS            VARCHAR2(250) := '&PERGUNTAS';
    CATEGORIA            varchar(30) := '&CATEGORIA';
    NOME_CRIADOR         VARCHAR2(30) := '&NOME_CRIADOR';
    DURACAO              date := '&DURACAO';
BEGIN
    BEGIN
        INSERT INTO questionario
        VALUES (COD_QUESTIONARIO,TITULO, DESCRICAO, PERGUNTAS, CATEGORIA, NOME_C
        DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

Tarefa concluída em 29,231 segundos

```
END;
END;
```

Procedimento PL/SQL concluído com sucesso.

Tabela Triagem

```
--Triagem--
DECLARE
    COD_TRIAGEM         NUMBER := &COD_TRIAGEM;
    INICIO              date := '&INICIO';
    FIM                 date := '&FIM';
    DURACAO             date := '&DURACAO';
    RESPOSTAS           varchar(30) := '&RESPOSTAS';
    PACIENTE_COD_PACIENTE number := &PACIENTE_COD_PACIENTE;
    QUESTIONARIO_COD_QUESTIONARIO number := &QUESTIONARIO_COD_QUESTIONARIO;
BEGIN
    BEGIN
        INSERT INTO triagem
        VALUES (COD_TRIAGEM,INICIO ,FIM, DURACAO, RESPOSTAS, PACIENTE_COD_PACIENTE, QUESTIONARIO_COD_Q
        DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso.');
```

Tarefa concluída em 27,754 segundos

```
END;
END;
```

Procedimento PL/SQL concluído com sucesso.

Diagnostico

```
--Diagnostico--
DECLARE
  COD_DIAGNOSTICO  NUMBER := &COD_DIAGNOSTICO;
  DATA_GERACAO    date   := '&DATA_GERACAO';
  RESULTADO        varchar2(100) := '&RESULTADO ';
  OBSERVACAO       varchar2(30) := '&OBSERVACAO ';
  SEGMENTO_ACAO    varchar2(30) := '&SEGMENTO_ACAO ';
  TRIAGEM_COD_TRIAGEM number := &TRIAGEM_COD_TRIAGEM;
  MEDICO_COD_MEDICO number := &MEDICO_COD_MEDICO ;
BEGIN
  BEGIN
    INSERT INTO diagnostico
    VALUES (COD_DIAGNOSTICO, DATA_GERACAO, RESULTADO, OBSERVACAO, TRIAGEM_COD_TRIAGEM, MEDICO_COD_MEDICO);

    DBMS_OUTPUT.PUT_LINE('Registro de farmácia inserido com sucesso');
  EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
      DBMS_OUTPUT.PUT_LINE('Erro: Dados repetidos.');
```

Procedimento PL/SQL concluído com sucesso.

Relatorio 1:

```
--Relatorio 1 com Cursor--
DECLARE
  -- Declaração de variáveis para cursor
  v_cod_especialidade NUMBER;
  v_especialidade      VARCHAR2(50);
  v_cod_medico         NUMBER;

  -- Declaração do cursor
  CURSOR especialidade_cur IS
    SELECT es.cod_especialidade, es.especialidade, es.medico_cod_medico
    FROM especialidade es;
BEGIN
  -- Execução do cursor
  OPEN especialidade_cur;
  LOOP
    FETCH especialidade_cur INTO v_cod_especialidade, v_especialidade, v_cod_medico;
    EXIT WHEN especialidade_cur%NOTFOUND;
    -- Processamento dos dados
    DBMS_OUTPUT.PUT_LINE('Código da Especialidade: ' || v_cod_especialidade ||
      'Especialidade: ' || v_especialidade ||
      'Código do Médico: ' || v_cod_medico);
  END LOOP;
  CLOSE especialidade_cur;
END;
```

Procedimento PL/SQL concluído com sucesso.

Código da Especialidade: 1
Especialidade: padeiro
Código do Médico: 1

Relatorio 2

The screenshot displays the Oracle SQL Developer environment. The top toolbar includes icons for running, saving, and editing. The main window is titled 'Query Builder' and contains the following PL/SQL code:

```
v_cod_medico          NUMBER;  
v_crp                 VARCHAR2(20);  
v_pessoa_fisica_cpf   NUMBER;  
v_cod_diagnostico     NUMBER;  
v_data_geracao        DATE;  
v_resultado           VARCHAR2(200);  
v_observacao          VARCHAR2(30);  
v_segmento_acao       VARCHAR2(30);  
v_triagem_cod_triagem NUMBER;  
  
-- Declaração do cursor  
CURSOR medico_diagnostico_cur IS  
SELECT m.COD_MEDICO,
```

Below the code editor, the 'DBMS_OUTPUT' window is open, showing the results of the script execution. The output is as follows:

```
CPF: 523232873  
CPF da Pessoa Física: 523232873  
Código do Diagnóstico: 1  
Data do Diagnóstico: 23/11/2023  
Resultado do Diagnóstico: CANCER  
Observação: CANCER  
Segmento de Ação: tomar remedio  
Código da Triagem: 1  
-----  
Procedimento PL/SQL concluído com sucesso.
```