

Integrantes

Leonardo Paganini RM96562

Matheus Leite RM96893

Jhonn Brandon RM97305

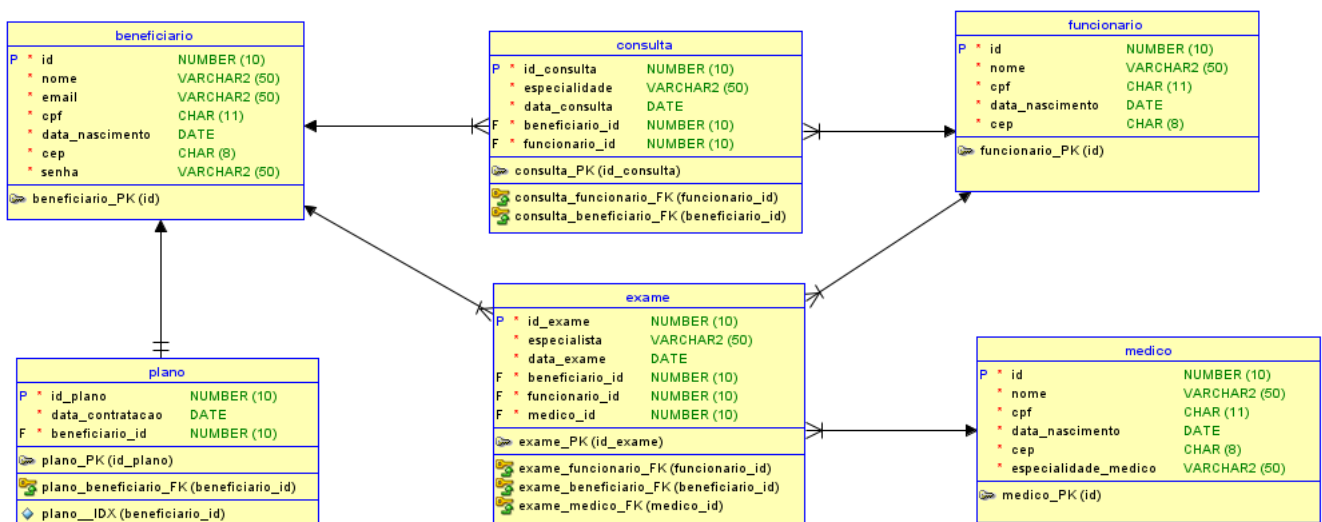
Regina Pompeo RM97032

Flow Life

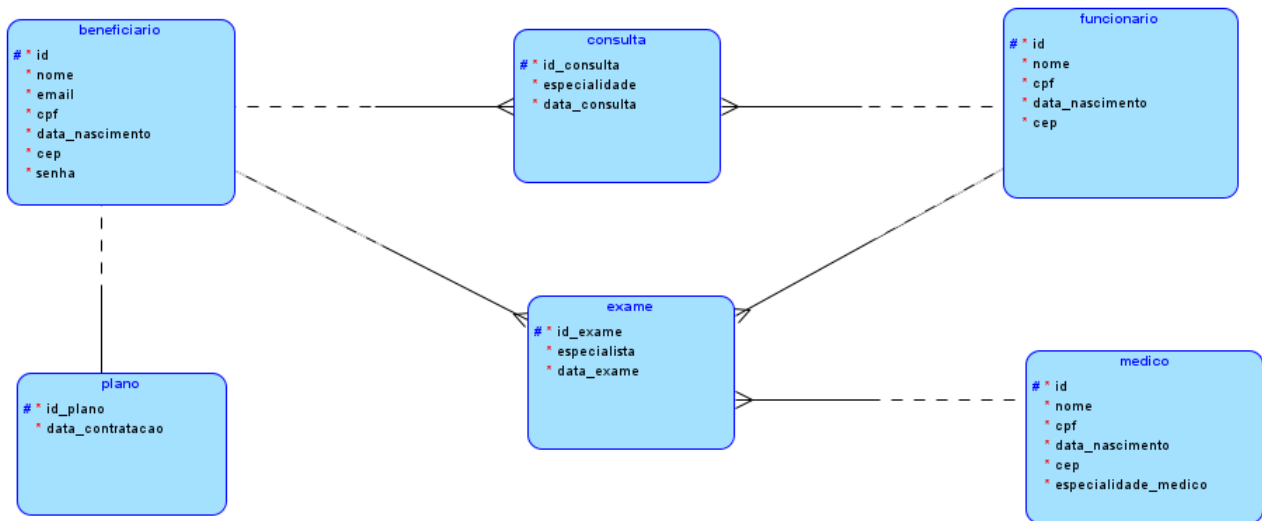
Descrição do Projeto

A nossa proposta inovadora visa revolucionar o acesso à saúde, proporcionando aos pacientes uma experiência simplificada e eficiente no momento da consulta. Através do nosso aplicativo intuitivo, os beneficiários podem agendar consultas, acompanhar o status da consulta e verificar os exames realizados. Eliminamos barreiras ao oferecer automação no processo de marcação, conectando os pacientes aos profissionais de saúde de forma rápida e precisa. Além disso, ao consolidar informações e resultados de exames em um só lugar, proporcionamos uma jornada de cuidados de entrada no paciente quando o mesmo comparece na consulta. Nosso foco em prevenção, automação e precisão não apenas otimiza o sistema hospitalar, mas também capacita os beneficiários a gerenciar proativamente sua saúde.

Modelagem Física



Modelagem Lógica



Criação dos objetos no Banco

```
12 CREATE TABLE beneficiario (
13     id          NUMBER(10) NOT NULL,
14     nome        VARCHAR2(50) NOT NULL,
15     email       VARCHAR2(50) NOT NULL,
16     cpf         CHAR(11) NOT NULL,
17     data_nascimento DATE NOT NULL,
18     cep         CHAR(8) NOT NULL,
19     senha       VARCHAR2(50) NOT NULL
20 );
21
22 ALTER TABLE beneficiario ADD CONSTRAINT beneficiario_pk PRIMARY KEY ( id );
23
24 CREATE TABLE consulta (
25     id_consulta    NUMBER(10) NOT NULL,
26     especialidade  VARCHAR2(50) NOT NULL,
27     data_consulta  DATE NOT NULL,
28     beneficiario_id NUMBER(10) NOT NULL,
29     funcionario_id NUMBER(10) NOT NULL
30 );
31
32 ALTER TABLE consulta ADD CONSTRAINT consulta_pk PRIMARY KEY ( id_consulta );
```

```

34 CREATE TABLE exame (
35     id_exame          NUMBER(10) NOT NULL,
36     especialista      VARCHAR2(50) NOT NULL,
37     data_exame        DATE NOT NULL,
38     beneficiario_id   NUMBER(10) NOT NULL,
39     funcionario_id     NUMBER(10) NOT NULL,
40     medico_id         NUMBER(10) NOT NULL
41 );
42
43 ALTER TABLE exame ADD CONSTRAINT exame_pk PRIMARY KEY ( id_exame );
44
45 CREATE TABLE funcionario (
46     id                NUMBER(10) NOT NULL,
47     nome              VARCHAR2(50) NOT NULL,
48     cpf               CHAR(11) NOT NULL,
49     data_nascimento   DATE NOT NULL,
50     cep               CHAR(8) NOT NULL
51 );
52
53 ALTER TABLE funcionario ADD CONSTRAINT funcionario_pk PRIMARY KEY ( id );
54
55 CREATE TABLE medico (
56     id                NUMBER(10) NOT NULL,
57     nome              VARCHAR2(50) NOT NULL,
58     cpf               CHAR(11) NOT NULL,
59     data_nascimento   DATE NOT NULL,
60     cep               CHAR(8) NOT NULL,
61     especialidade_medico VARCHAR2(50) NOT NULL
62 );
63
64 ALTER TABLE medico ADD CONSTRAINT medico_pk PRIMARY KEY ( id );
65
66 CREATE TABLE plano (
67     id_plano          NUMBER(10) NOT NULL,
68     data_contratacao  DATE NOT NULL,
69     beneficiario_id   NUMBER(10) NOT NULL
70 );
71
72 CREATE UNIQUE INDEX plano__idx ON
73     plano (
74         beneficiario_id
75     ASC );
76
77 ALTER TABLE plano ADD CONSTRAINT plano_pk PRIMARY KEY ( id_plano );

```

```

79 ALTER TABLE consulta
80     ADD CONSTRAINT consulta_beneficiario_fk FOREIGN KEY ( beneficiario_id )
81     REFERENCES beneficiario ( id );
82
83 ALTER TABLE consulta
84     ADD CONSTRAINT consulta_funcionario_fk FOREIGN KEY ( funcionario_id )
85     REFERENCES funcionario ( id );
86
87 ALTER TABLE exame
88     ADD CONSTRAINT exame_beneficiario_fk FOREIGN KEY ( beneficiario_id )
89     REFERENCES beneficiario ( id );
90
91 ALTER TABLE exame
92     ADD CONSTRAINT exame_funcionario_fk FOREIGN KEY ( funcionario_id )
93     REFERENCES funcionario ( id );
94
95 ALTER TABLE exame
96     ADD CONSTRAINT exame_medico_fk FOREIGN KEY ( medico_id )
97     REFERENCES medico ( id );
98
99 ALTER TABLE plano
100     ADD CONSTRAINT plano_beneficiario_fk FOREIGN KEY ( beneficiario_id )
101     REFERENCES beneficiario ( id );

```

Table BENEFICIARIO criado.

Table BENEFICIARIO alterado.

Table CONSULTA criado.

Table CONSULTA alterado.

Table EXAME criado.

Table EXAME alterado.

Table FUNCIONARIO criado.

Table FUNCIONARIO alterado.

Table MEDICO criado.

Table MEDICO alterado.

Table PLANO criado.

INDEX PLANO__IDX criado.

Table PLANO alterado.

Table CONSULTA alterado.

Table CONSULTA alterado.

Table EXAME alterado.

Table EXAME alterado.

Table EXAME alterado.

Table PLANO alterado.

Evidências da carga de dados

Beneficiário:

107 DECLARE

108 v_id NUMBER(10);

109 v_nome VARCHAR2(50);

110 v_email VARCHAR2(50);

111 v_cpf CHAR(11);

112 v_data_nascimento DATE;

113 v_cep CHAR(8);

114 v_senha VARCHAR2(50);

115 BEGIN

116 -- Inserindo dados para a tabela beneficiario

117 v_id := 5;

118 v_nome := 'Tiago Nicolas Thiago Teixeira';

119 v_email := 'tiago.nicolas.teixeira@focustg.com.br';

120 v_cpf := '95443312472';

121 v_data_nascimento := TO_DATE('19931017', 'YYYYMMDD');

122 v_cep := '54330551';

123 v_senha := 'kbUuFd6yi0';

124

125 -- Tratando exceções para evitar inserção duplicada na tabela beneficiario

126 BEGIN

127 INSERT INTO beneficiario (id, nome, email, cpf, data_nascimento, cep, senha)

128 VALUES (v_id, v_nome, v_email, v_cpf, v_data_nascimento, v_cep, v_senha);

129 EXCEPTION

130 WHEN DUP_VAL_ON_INDEX THEN

131 DBMS_OUTPUT.PUT_LINE('Erro: Beneficiario com ID ' || v_id || ' já existe.');

132 END;

133 COMMIT;

134 END;

Saída do Script x Resultado da Consulta x

Todas as Linhas Extraídas: 5 em 0,015 segundos

ID	NOME	EMAIL	CPF	DATA_NASCIMENTO	CEP	SENHA
1	1 Márcio Bento Silveira	marcio_bento_silveira@mucooucah.com.br	03171905124	16/10/49	79906074	bp4ebpbeRb
2	2 Enrico César Ribeiro	enrico_cesar_ribeiro@dgh.com.br	82366355971	02/01/94	88134441	O6JN1yZNMU
3	3 Theo Anderson Moreira	theoandersonmoreira@baltico.com.br	04980439403	14/09/94	88132738	WRWW3MHcIH
4	4 Osvaldo Julio Peixoto	osvaldojuliopeixoto@gmail.com	94170139984	26/06/45	72410615	USUTklRxmi
5	5 Tiago Nicolas Thiago Teixeira	tiago.nicolas.teixeira@focustg.com.br	95443312472	17/10/93	54330551	kbUuFd6yi0

Funcionário:

```
138 DECLARE
139     v_id          NUMBER(10);
140     v_nome        VARCHAR2(50);
141     v_cpf         CHAR(11);
142     v_data_nascimento DATE;
143     v_cep         CHAR(8);
144 BEGIN
145     -- Inserindo dados para a tabela funcionario
146     v_id := 5;
147     v_nome := 'Nathan Jorge Moraes';
148     v_cpf := '40506004635';
149     v_data_nascimento := TO_DATE('19510607', 'YYYYMMDD');
150     v_cep := '81710100';
151
152     -- Tratando exceções para evitar inserção duplicada na tabela funcionario
153     BEGIN
154         INSERT INTO funcionario (id, nome, cpf, data_nascimento, cep)
155         VALUES (v_id, v_nome, v_cpf, v_data_nascimento, v_cep);
156     EXCEPTION
157         WHEN DUP_VAL_ON_INDEX THEN
158             DBMS_OUTPUT.PUT_LINE('Erro: Funcionario com ID ' || v_id || ' já existe.');
```

Saída do Script

Resultado da Consulta

SQL

Todas as Linhas Extraídas: 5 em 0,016 segundos

ID	NOME	CPF	DATA_NASCIMENTO	CEP
1	1 Bryan Felipe Novaes	31255300914	20/07/54	69088492
2	2 Matheus Leonardo Thales Viana	64905174503	08/05/49	75805055
3	3 Bento Luan César Galvão	81015521703	12/07/58	29161787
4	4 Kaique Heitor César Oliveira	06477492202	19/10/49	64027424
5	5 Nathan Jorge Moraes	40506004635	07/06/51	81710100

Médico:

165

DECLARE

166

v_id

NUMBER(10);

167

v_nome

VARCHAR2(50);

168

v_cpf

CHAR(11);

169

v_data_nascimento

DATE;

170

v_cep

CHAR(8);

171

v_especialidade_medico

VARCHAR2(50);

172

BEGIN

173

-- Inserindo dados para a tabela medico

174

v_id := 5;

175

v_nome := 'Kauê Lorenzo Sebastião Pires';

176

v_cpf := '84890442367';

177

v_data_nascimento := TO_DATE('19501103', 'YYYYMMDD');

178

v_cep := '79037887';

179

v_especialidade_medico := 'Infectologia';

180

181

-- Tratando exceções para evitar inserção duplicada na tabela medico

182

BEGIN

183

INSERT INTO medico (id, nome, cpf, data_nascimento, cep, especialidade_medico)

184

VALUES (v_id, v_nome, v_cpf, v_data_nascimento, v_cep, v_especialidade_medico);

185

EXCEPTION

186

WHEN DUP_VAL_ON_INDEX THEN

187

DBMS_OUTPUT.PUT_LINE('Erro: Medico com ID ' || v_id || ' já existe.');

188

END;

189

COMMIT;

190

END;

Saída do Script

Resultado da Consulta

Todas as Linhas Extraídas: 5 em 0,018 segundos

ID	NOME	CPF	DATA_NASCIMENTO	CEP	ESPECIALIDADE_MEDICO
1	1 Márcio Luan Ian Drumond	11349631221	05/11/46	65092039	Cardiologista
2	2 Márcio Fernando Arthur Silva	08794705933	07/04/58	59075053	Acupuntura
3	3 Murilo Julio Assunção	81039810608	20/04/74	94065462	Coloproctologia
4	4 Manoel Daniel Lucca Rodrigues	90162244401	01/06/69	68970970	Geriatria
5	5 Kauê Lorenzo Sebastião Pires	84890442367	03/11/50	79037887	Infectologia

Plano:

```
250 DECLARE
251     v_id_plano          NUMBER(10);
252     v_data_contratacao  DATE;
253     v_beneficiario_id   NUMBER(10);
254 BEGIN
255     -- Inserindo dados para a tabela plano
256     v_id_plano := 5;
257     v_data_contratacao := TO_DATE('20230909', 'YYYYMMDD');
258     v_beneficiario_id :=1;
259
260     -- Tratando exceções para evitar inserção duplicada na tabela plano
261     BEGIN
262         INSERT INTO plano (id_plano, data_contratacao, beneficiario_id)
263             VALUES (v_id_plano, v_data_contratacao, v_beneficiario_id);
264     EXCEPTION
265         WHEN DUP_VAL_ON_INDEX THEN
266             DBMS_OUTPUT.PUT_LINE('Erro: Plano com ID ' || v_id_plano || ' já existe.');
```

Saída do Script x Resultado da Consulta x

SQL | Todas as Linhas Extraídas: 5 em 0,012 segundos

ID_PLANO	DATA_CONTRATACAO	BENEFICIARIO_ID
1	1 01/08/23	5
2	2 19/02/23	4
3	3 09/05/23	3
4	4 12/12/23	2
5	5 09/09/23	1

Consulta:

```
194 DECLARE
195     v_id_consulta      NUMBER(10);
196     v_especialidade    VARCHAR2(50);
197     v_data_consulta    DATE;
198     v_beneficiario_id  NUMBER(10);
199     v_funcionario_id   NUMBER(10);
200 BEGIN
201     -- Inserindo dados para a tabela consulta
202     v_id_consulta := 5;
203     v_especialidade := 'Infectologia';
204     v_data_consulta := TO_DATE('20230324', 'YYYYMMDD');
205     v_beneficiario_id := 5;
206     v_funcionario_id := 1;
207
208     -- Tratando exceções para evitar inserção duplicada na tabela consulta
209     BEGIN
210         INSERT INTO consulta (id_consulta, especialidade, data_consulta, beneficiario_id, funcionario_id)
211         VALUES (v_id_consulta, v_especialidade, v_data_consulta, v_beneficiario_id, v_funcionario_id);
212     EXCEPTION
213         WHEN DUP_VAL_ON_INDEX THEN
214             DBMS_OUTPUT.PUT_LINE('Erro: Consulta com ID ' || v_id_consulta || ' já existe.');
```

Saída do Script x
Resultado da Consulta x

Todas as Linhas Extraídas: 5 em 0,012 segundos

ID_CONSULTA	ESPECIALIDADE	DATA_CONSULTA	BENEFICIARIO_ID	FUNCIONARIO_ID
1	1 Cardiologia	01/01/23	1	5
2	2 Acupuntura	11/05/23	2	4
3	3 Coloproctologia	21/04/23	3	3
4	4 Geriatria	07/08/23	4	2
5	5 Infectologia	24/03/23	5	1

Exame:

```

221 | DECLARE
222 |     v_id_exame            NUMBER(10);
223 |     v_especialista        VARCHAR2(50);
224 |     v_data_exame          DATE;
225 |     v_beneficiario_id     NUMBER(10);
226 |     v_funcionario_id      NUMBER(10);
227 |     v_medico_id           NUMBER(10);
228 | BEGIN
229 |     -- Inserindo dados para a tabela exame
230 |     v_id_exame := 5;
231 |     v_especialista := 'Cardiologista';
232 |     v_data_exame := TO_DATE('20231125', 'YYYYMMDD');
233 |     v_beneficiario_id := 5;
234 |     v_funcionario_id := 1;
235 |     v_medico_id := 4;
236 |
237 |     -- Tratando exceções para evitar inserção duplicada na tabela exame
238 |     BEGIN
239 |         INSERT INTO exame (id_exame, especialista, data_exame, beneficiario_id, funcionario_id, medico_id)
240 |         VALUES (v_id_exame, v_especialista, v_data_exame, v_beneficiario_id, v_funcionario_id, v_medico_id);
241 |     EXCEPTION
242 |         WHEN DUP_VAL_ON_INDEX THEN
243 |             DBMS_OUTPUT.PUT_LINE('Erro: Exame com ID ' || v_id_exame || ' já existe.');
```

Saída do Script x Resultado da Consulta x

Todas as Linhas Extraídas: 5 em 0,013 segundos

ID_EXAME	ESPECIALISTA	DATA_EXAME	BENEFICIARIO_ID	FUNCIONARIO_ID	MEDICO_ID
1	1 Infectologia	01/02/23	1	5	3
2	2 Geriatria	11/09/23	2	4	1
3	3 Coloproctologia	29/01/23	3	3	2
4	4 Acupuntura	20/12/23	4	2	5
5	5 Cardiologista	25/11/23	5	1	4

Evidências da execução das duas pesquisas

Relatório Consulta:

```
DECLARE
    CURSOR consulta_cursor IS
        SELECT c.id_consulta,
               c.especialidade,
               c.data_consulta,
               b.nome AS beneficiario_nome,
               f.nome AS funcionario_nome
        FROM consulta c
        JOIN beneficiario b ON c.beneficiario_id = b.id
        JOIN funcionario f ON c.funcionario_id = f.id;
BEGIN
    DBMS_OUTPUT.PUT_LINE('ID_CONSULTA | ESPECIALIDADE | DATA_CONSULTA | BENEFICIARIO_NOME | FUNCIONARIO_NOME');
    DBMS_OUTPUT.PUT_LINE('-----');

    FOR consulta_rec IN consulta_cursor
    LOOP
        DBMS_OUTPUT.PUT_LINE(consulta_rec.id_consulta || ' | ' ||
                               consulta_rec.especialidade || ' | ' ||
                               TO_CHAR(consulta_rec.data_consulta, 'DD/MM/YYYY') || ' | ' ||
                               consulta_rec.beneficiario_nome || ' | ' ||
                               consulta_rec.funcionario_nome);
    END LOOP;
END;
```

Saída do Script x Resultado da Consulta x

Tarefa concluída em 0,085 segundos

ID_CONSULTA	ESPECIALIDADE	DATA_CONSULTA	BENEFICIARIO_NOME	FUNCIONARIO_NOME
1	Cardiologia	01/01/2023	Márcio Bento Silveira	Nathan Jorge Moraes
2	Acupuntura	11/05/2023	Enrico César Ribeiro	Kaique Heitor César Oliveira
3	Coloproctologia	21/04/2023	Theo Anderson Moreira	Bento Luan César Galvão
4	Geriatrics	07/08/2023	Oswaldo Julio Peixoto	Matheus Leonardo Thales Viana
5	Infectologia	24/03/2023	Tiago Nicolas Thiago Teixeira	Bryan Felipe Novaes

Procedimento PL/SQL concluído com sucesso.

Relatório Exame:

```
DECLARE
    CURSOR exame_cursor IS
        SELECT e.id_exame,
               e.especialista,
               e.data_exame,
               b.nome AS beneficiario_nome,
               f.nome AS funcionario_nome,
               m.nome AS medico_nome
        FROM exame e
        JOIN beneficiario b ON e.beneficiario_id = b.id
        JOIN funcionario f ON e.funcionario_id = f.id
        JOIN medico m ON e.medico_id = m.id;
BEGIN
    DBMS_OUTPUT.PUT_LINE('ID_EXAME | ESPECIALISTA | DATA_EXAME | BENEFICIARIO_NOME | FUNCIONARIO_NOME | MEDICO_NOME');
    DBMS_OUTPUT.PUT_LINE('=====');
    FOR exame_rec IN exame_cursor
    LOOP
        DBMS_OUTPUT.PUT_LINE(exame_rec.id_exame || ' | ' ||
                               exame_rec.especialista || ' | ' ||
                               TO_CHAR(exame_rec.data_exame, 'DD/MM/YYYY') || ' | ' ||
                               exame_rec.beneficiario_nome || ' | ' ||
                               exame_rec.funcionario_nome || ' | ' ||
                               exame_rec.medico_nome);
    END LOOP;
END;
```

Salida do Script x Resultado da Consulta x

Tarefa concluída em 0,061 segundos

ID_EXAME	ESPECIALISTA	DATA_EXAME	BENEFICIARIO_NOME	FUNCIONARIO_NOME	MEDICO_NOME
2	Geriatria	11/09/2023	Enrico César Ribeiro	Kaique Heitor César Oliveira	Márcio Luan Ian Drumond
3	Coloproctologia	29/01/2023	Theo Anderson Moreira	Bento Luan César Galvão	Márcio Fernando Arthur Silva
1	Infectologia	01/02/2023	Márcio Bento Silveira	Nathan Jorge Moraes	Murilo Julio Assunção
5	Cardiologista	25/11/2023	Tiago Nicolas Thiago Teixeira	Bryan Felipe Novaes	Manoel Daniel Lucca Rodrigues
4	Acupuntura	20/12/2023	Osvaldo Julio Peixoto	Matheus Leonardo Thales Viana	Kauê Lorenzo Sebastião Pires

Procedimento PL/SQL concluído com sucesso.