

```

2
3 -- João Pedro Soares da Franca PE3021114
4 -- Exercício 1
5 CREATE OR REPLACE FUNCTION abntextendido(nome VARCHAR(100))
6     returns varchar(60)
7     LANGUAGE 'plpgsql'
8 AS
9 $$
10 DECLARE
11     abntado text[];
12     tamanho int;
13     correto varchar;
14 begin
15     abntado := string_to_array(nome, ' ');
16     tamanho := array_length(abntado, 1);
17     correto := Concat(upper(abntado[tamanho]), ',');
18     for i in 1 .. (tamanho - 1) loop
19         correto := concat(correto, ' ', abntado[i]);
20     end loop;
21     return correto;
22 end;
23 $$;

```

```

27 -- João Pedro Soares da Franca PE3021114
28 -- Exercício 2
29 CREATE OR REPLACE FUNCTION abnt(nome VARCHAR(100))
30     returns varchar(60)
31     LANGUAGE 'plpgsql'
32 AS
33 $$
34 DECLARE
35     abntado text[];
36     tamanho int;
37     correto varchar;
38     palavra varchar;
39 begin
40     abntado := string_to_array(nome, ' ');
41     tamanho := array_length(abntado, 1);
42     correto := Concat(upper(abntado[tamanho]), ',');
43     for i in 1 .. (tamanho - 1) loop
44         palavra := lower(abntado[i]);
45         if (palavra not in ('da', 'do', 'das', 'dos')) then
46             palavra := concat(upper(left(palavra, 1)), '.');
47         end if;
48         correto := concat(correto, ' ', palavra);
49     end loop;
50     return correto;
51 end;
52 $$;

```

```

56 -- João Pedro Soares da Franca PE3021114
57 ✓ -- Exercício 3
58 CREATE OR REPLACE FUNCTION abrevia(nome VARCHAR(100))
59     returns varchar(60)
60     LANGUAGE 'plpgsql'
61 AS
62 $$
63 DECLARE
64     abntado text[];
65     tamanho int;
66     correto varchar;
67     palavra varchar;
68 begin
69     abntado := string_to_array(nome, ' ');
70     tamanho := array_length(abntado, 1);
71     correto := abntado[1];
72     for i in 2 .. (tamanho - 1) loop
73         palavra := lower(abntado[i]);
74         if (palavra not in ('da', 'do', 'das', 'dos')) then
75             palavra := concat(upper(left(palavra, 1)), '.');
76         end if;
77         correto := concat(correto, ' ', palavra);
78     end loop;
79
80     correto := Concat(correto, ' ', abntado[tamanho]);
81     return correto;
82 end;
83 $$;

```

```

87 -- João Pedro Soares da Franca PE3021114
88 ✓ -- Exercício 4
89 CREATE OR REPLACE FUNCTION retornaNome(posicao int, nome VARCHAR(100))
90     returns varchar(60)
91     LANGUAGE 'plpgsql'
92 AS
93 $$
94 DECLARE
95     abntado text[];
96     tamanho int;
97 begin
98     abntado := string_to_array(nome, ' ');
99     tamanho := array_length(abntado, 1);
100     if posicao > tamanho then
101         raise exception 'Posição maior do que a quantidade de palavras';
102     end if;
103     return abntado[posicao];
104 end;
105 $$;

```

```

109 -- João Pedro Soares da Franca PE3021114
110 ✓ -- Exercício 5
111 create or replace function contvogais(nome varchar(60)) returns int
112     language plpgsql
113 as
114 $$
115 declare
116     retorno int default 0;
117     i int default 0;
118 begin
119     for i in 0 .. length(nome) loop
120         IF LOWER(SUBSTRING(nome, i, 1)) IN ('a', 'e', 'i', 'o', 'u') THEN
121             retorno := retorno + 1;
122         end if;
123     end loop;
124     return retorno;
125 end;
126 $$;

```

```

130 -- João Pedro Soares da Franca PE3021114
131 ✓ -- Exercício 6
132 CREATE OR REPLACE FUNCTION autores(codmusica int)
133     RETURNS varchar
134     LANGUAGE plpgsql
135 AS $$
136 DECLARE
137     retorno varchar;
138     codigo_musica int default 0;
139 BEGIN
140     select musica.codmus from musica where codmus = codmusica into codigo_musica;
141
142     if (codigo_musica is not null) then
143         SELECT string_agg(upper(a.nomeaut), ', ')
144         INTO retorno
145         FROM autor a
146             INNER JOIN musicaautor m 1<->1..n: ON a.codaut = m.codaut
147         WHERE m.codmus = autores.codmusica;
148
149         RETURN retorno;
150     end if;
151
152     raise exception 'Musica não encontrada, informações inválidas';
153 END;
154 $$;

```