

Universidade de Aveiro
Departamento de Electrónica, Telecomunicações e Informática

DigiPeet
Base De Dados

Professor: Pedro Fonseca

Alunos: João Lima 48019
Rafael Fonseca 93355

Março
2021

Conteúdo

1	Introdução:	1
2	Descrição Geral:	1
3	Tipo de Utilizadores	2
3.1	Administrador:	2
3.2	Voluntário:	2
4	Descrição dos Use-Cases:	3
4.1	Todos os Utilizadores:	3
4.2	Administrador:	3
4.3	Voluntário:	4
4.3.1	Registo na app:	4
4.3.2	Menu:	4
5	Descrição do Diagrama de Classes:	5
6	Conversão do Diagrama de Classes para o Modelo Relacional	6
7	Descrição do Modelo Físico:	7
8	Criação de Tabelas:	8
9	Referências	14

1 Introdução:

Este relatório tem como objetivo demonstrar como foi implementada a base de dados para aplicação DigiPeet. A aplicação é uma ferramenta para o utilizador, que tem como objetivo facilitar a sua experiência. Nela o utilizador poderá criar uma conta pessoal, fazer gestão do armazém/stock, aceder a um calendário de atividades e de vacinas e configurar o Feed eet.

2 Descrição Geral:

A base de dados tem com função guardar e atualizar autonomamente todos os dados provenientes da aplicação e do dispensador, garantindo assim o seu bom funcionamento.

3 Tipo de Utilizadores

3.1 Administrador:

- Tem a função criar ou editar os utilizadores da aplicação;

3.2 Voluntário:

- Utilizadores da aplicação.

4 Descrição dos Use-Cases:

4.1 Todos os Utilizadores:

- Login: O utilizador entra na aplicação, o sistema regista a hora de entrada e apresenta ao utilizador a sua área de trabalho (Cada tipo de utilizador tem um login diferente).

- Logout: O utilizador sai da aplicação, o sistema regista a hora de saída, e é fechada a área de trabalho.

4.2 Administrador:

- Aprovar Contas: Aprovar o registo de novas contas na aplicação. Só depois deste passo é que as contas são adicionadas a base de dados.

- Editar Contas: Permite activar/desactivar as contas existentes, bem como alterar o nome e a password

- Lista das Contas: Obter uma listagem com todas as contas de utilizadores.

4.3 Voluntário:

4.3.1 Registo na app:

- Criar Conta: Preenchimento de um formulário com nome completo, email, idade, género e tipo de utilizador. Também é necessária a escolha de um username e uma password para fazer o login na aplicação.

4.3.2 Menu:

-Ler código de barras: O voluntário pode ler códigos de barras de produtos através da camera do smartphone, para isso o utilizador precisa aceitar as permissões de acesso à camera.

-Adicionar produtos: Após a leitura do código de barras, caso esse produto não exista na aplicação, o utilizador pode adicioná-lo.

-Aceder ao calendário: O utilizador pode aceder ao calendário e ver as atividades futuras.

-Ver inventário: O utilizador pode ver os produtos que existem em armazém.

-Atualizar stock: O utilizador pode adicionar/retirar produtos ao stock.

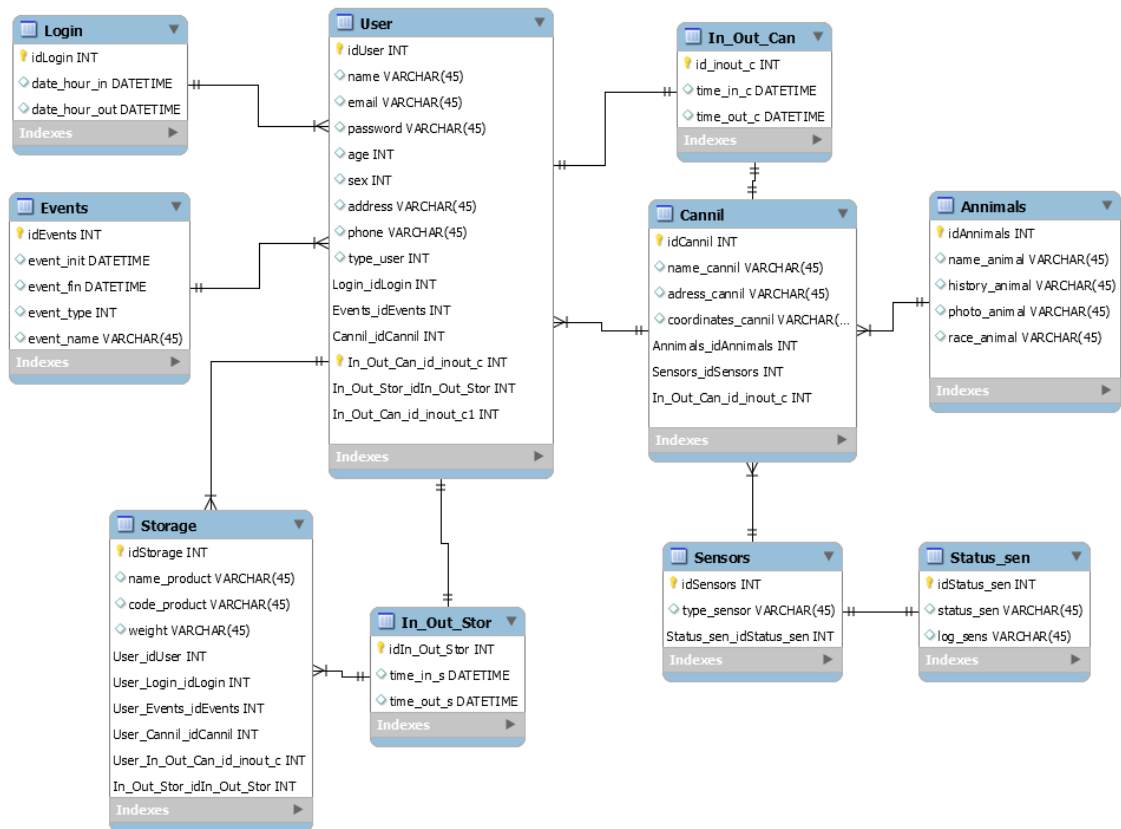
-Aceder ao dispensador: O utilizador pode controlar o dispensador a partir da aplicação.

-Alterar definições: O utilizador pode alterar as definições da aplicação.

5 Descrição do Diagrama de Classes:

6 Conversão do Diagrama de Classes para o Modelo Relacional

7 Descrição do Modelo Físico:



8 Criação de Tabelas:

De acordo com o modelo físico é gerado o seguinte código SQL para criação das tabelas. Este processo foi automatizado através do software Mysql Workbench. Nestas tabelas são utilizadas chaves primárias e chaves estrangeiras de modo a podermos aceder às várias tabelas conforme o especificado no modelo físico.

```
1  -----
2  -- Schema mydb
3  -----
4  CREATE SCHEMA IF NOT EXISTS 'mydb' DEFAULT CHARACTER SET utf8
5  ;
6  USE 'mydb' ;
7  -----
8  -- Table 'mydb'.'Login'
9  -----
10 CREATE TABLE IF NOT EXISTS 'mydb'.'Login' (
11   'idLogin' INT NOT NULL,
12   'date_hour_in' DATETIME NULL,
13   'date_hour_out' DATETIME NULL,
14   PRIMARY KEY ('idLogin'))
15 ENGINE = InnoDB;
16
17
18 -----
19 -- Table 'mydb'.'Events'
20 -----
21 CREATE TABLE IF NOT EXISTS 'mydb'.'Events' (
22   'idEvents' INT NOT NULL,
23   'event_init' DATETIME NULL,
24   'event_fin' DATETIME NULL,
25   'event_type' INT NULL,
26   'event_name' VARCHAR(45) NULL,
27   PRIMARY KEY ('idEvents'))
28 ENGINE = InnoDB;
29
30
31 -----
```

```

32 -- Table 'mydb'. 'Animals'
33 -----
34 CREATE TABLE IF NOT EXISTS 'mydb'. 'Animals' (
35     'idAnimals' INT NOT NULL AUTO_INCREMENT,
36     'name_animal' VARCHAR(45) NULL,
37     'history_animal' VARCHAR(45) NULL,
38     'photo_animal' VARCHAR(45) NULL,
39     'race_animal' VARCHAR(45) NULL,
40     PRIMARY KEY ('idAnimals'))
41 ENGINE = InnoDB;
42
43
44 -----
45 -- Table 'mydb'. 'Status_sen'
46 -----
47 CREATE TABLE IF NOT EXISTS 'mydb'. 'Status_sen' (
48     'idStatus_sen' INT NOT NULL AUTO_INCREMENT,
49     'status_sen' VARCHAR(45) NULL,
50     'log_sens' VARCHAR(45) NULL,
51     PRIMARY KEY ('idStatus_sen'))
52 ENGINE = InnoDB;
53
54
55 -----
56 -- Table 'mydb'. 'Sensors'
57 -----
58 CREATE TABLE IF NOT EXISTS 'mydb'. 'Sensors' (
59     'idSensors' INT NOT NULL AUTO_INCREMENT,
60     'type_sensor' VARCHAR(45) NULL,
61     'Status_sen_idStatus_sen' INT NOT NULL,
62     PRIMARY KEY ('idSensors', 'Status_sen_idStatus_sen'),
63     INDEX 'fk_Sensors_Status_sen1_idx' ('
        Status_sen_idStatus_sen' ASC) VISIBLE,
64     CONSTRAINT 'fk_Sensors_Status_sen1'
65         FOREIGN KEY ('Status_sen_idStatus_sen')
66         REFERENCES 'mydb'. 'Status_sen' ('idStatus_sen')
67         ON DELETE NO ACTION
68         ON UPDATE NO ACTION)
69 ENGINE = InnoDB;
70
71
72 -----
73 -- Table 'mydb'. 'In_Out_Can'
74 -----
75 CREATE TABLE IF NOT EXISTS 'mydb'. 'In_Out_Can' (
76     'id_inout_c' INT NOT NULL AUTO_INCREMENT,
77     'time_in_c' DATETIME NULL,
78     'time_out_c' DATETIME NULL,
79     PRIMARY KEY ('id_inout_c'))

```

```

80 ENGINE = InnoDB;
81
82
83 -----
84 -- Table 'mydb`.`Cannil`
85 -----
86 CREATE TABLE IF NOT EXISTS 'mydb`.`Cannil` (
87   'idCannil' INT NOT NULL AUTO_INCREMENT,
88   'name_cannil' VARCHAR(45) NULL,
89   'adress_cannil' VARCHAR(45) NULL,
90   'coordinates_cannil' VARCHAR(45) NULL,
91   'Animals_idAnimals' INT NOT NULL,
92   'Sensors_idSensors' INT NOT NULL,
93   'In_Out_Can_id_inout_c' INT NOT NULL,
94   PRIMARY KEY ('idCannil', 'Animals_idAnimals', '
95     Sensors_idSensors', 'In_Out_Can_id_inout_c'),
96   INDEX 'fk_Cannil_Animals1_idx' ('Animals_idAnimals' ASC)
97     VISIBLE,
98   INDEX 'fk_Cannil_Sensors1_idx' ('Sensors_idSensors' ASC)
99     VISIBLE,
100   INDEX 'fk_Cannil_In_Out_Can1_idx' ('In_Out_Can_id_inout_c'
101     ASC) VISIBLE,
102   CONSTRAINT 'fk_Cannil_Animals1'
103     FOREIGN KEY ('Animals_idAnimals')
104     REFERENCES 'mydb`.`Animals' ('idAnimals')
105     ON DELETE NO ACTION
106     ON UPDATE NO ACTION,
107   CONSTRAINT 'fk_Cannil_Sensors1'
108     FOREIGN KEY ('Sensors_idSensors')
109     REFERENCES 'mydb`.`Sensors' ('idSensors')
110     ON DELETE NO ACTION
111     ON UPDATE NO ACTION,
112   CONSTRAINT 'fk_Cannil_In_Out_Can1'
113     FOREIGN KEY ('In_Out_Can_id_inout_c')
114     REFERENCES 'mydb`.`In_Out_Can' ('id_inout_c')
115     ON DELETE NO ACTION
116     ON UPDATE NO ACTION)
117 ENGINE = InnoDB;
118
119 -----
120 -- Table 'mydb`.`In_Out_Stor`
121 -----
122 CREATE TABLE IF NOT EXISTS 'mydb`.`In_Out_Stor` (
123   'idIn_Out_Stor' INT NOT NULL,
124   'time_in_s' DATETIME NULL,
125   'time_out_s' DATETIME NULL,
126   PRIMARY KEY ('idIn_Out_Stor'))
127 ENGINE = InnoDB;

```

```

125
126
127 -----
128 -- Table 'mydb'.'User'
129 -----
130 CREATE TABLE IF NOT EXISTS 'mydb'.'User' (
131     'idUser' INT NOT NULL,
132     'name' VARCHAR(45) NULL,
133     'email' VARCHAR(45) NULL,
134     'password' VARCHAR(45) NULL,
135     'age' INT NULL,
136     'sex' INT NULL,
137     'address' VARCHAR(45) NULL,
138     'phone' VARCHAR(45) NULL,
139     'type_user' INT NULL,
140     'Login_idLogin' INT NOT NULL,
141     'Events_idEvents' INT NOT NULL,
142     'Cannil_idCannil' INT NOT NULL,
143     'In_Out_Can_id_inout_c' INT NOT NULL,
144     'In_Out_Stor_idIn_Out_Stor' INT NOT NULL,
145     'In_Out_Can_id_inout_c1' INT NOT NULL,
146     PRIMARY KEY ('idUser', 'Login_idLogin', 'Events_idEvents',
147         'Cannil_idCannil', 'In_Out_Can_id_inout_c', '
148         In_Out_Stor_idIn_Out_Stor', 'In_Out_Can_id_inout_c1'),
149     INDEX 'fk_User_Login_idx' ('Login_idLogin' ASC) VISIBLE,
150     INDEX 'fk_User_Events1_idx' ('Events_idEvents' ASC) VISIBLE
151     ,
152     INDEX 'fk_User_Cannil1_idx' ('Cannil_idCannil' ASC) VISIBLE
153     ,
154     INDEX 'fk_User_In_Out_Stor1_idx' ('
155         In_Out_Stor_idIn_Out_Stor' ASC) VISIBLE,
156     INDEX 'fk_User_In_Out_Can1_idx' ('In_Out_Can_id_inout_c1'
157         ASC) VISIBLE,
158     CONSTRAINT 'fk_User_Login'
159         FOREIGN KEY ('Login_idLogin')
160         REFERENCES 'mydb'.'Login' ('idLogin')
161         ON DELETE NO ACTION
162         ON UPDATE NO ACTION,
163     CONSTRAINT 'fk_User_Events1'
164         FOREIGN KEY ('Events_idEvents')
165         REFERENCES 'mydb'.'Events' ('idEvents')
166         ON DELETE NO ACTION
167         ON UPDATE NO ACTION,
168     CONSTRAINT 'fk_User_Cannil1'
169         FOREIGN KEY ('Cannil_idCannil')
170         REFERENCES 'mydb'.'Cannil' ('idCannil')
171         ON DELETE NO ACTION
172         ON UPDATE NO ACTION,
173     CONSTRAINT 'fk_User_In_Out_Stor1'

```

```

168 FOREIGN KEY ('In_Out_Stor_idIn_Out_Stor')
169 REFERENCES 'mydb'.'In_Out_Stor' ('idIn_Out_Stor')
170 ON DELETE NO ACTION
171 ON UPDATE NO ACTION,
172 CONSTRAINT 'fk_User_In_Out_Can1'
173 FOREIGN KEY ('In_Out_Can_id_inout_c1')
174 REFERENCES 'mydb'.'In_Out_Can' ('id_inout_c')
175 ON DELETE NO ACTION
176 ON UPDATE NO ACTION)
177 ENGINE = InnoDB;
178
179
180 -----
181 -- Table 'mydb'.'Storage'
182 -----
183 CREATE TABLE IF NOT EXISTS 'mydb'.'Storage' (
184 'idStorage' INT NOT NULL AUTO_INCREMENT,
185 'name_product' VARCHAR(45) NULL,
186 'code_product' VARCHAR(45) NULL,
187 'weight' VARCHAR(45) NULL,
188 'User_idUser' INT NOT NULL,
189 'User_Login_idLogin' INT NOT NULL,
190 'User_Events_idEvents' INT NOT NULL,
191 'User_Cannil_idCannil' INT NOT NULL,
192 'User_In_Out_Can_id_inout_c' INT NOT NULL,
193 'In_Out_Stor_idIn_Out_Stor' INT NOT NULL,
194 PRIMARY KEY ('idStorage', 'User_idUser', '
    User_Login_idLogin', 'User_Events_idEvents', '
    User_Cannil_idCannil', 'User_In_Out_Can_id_inout_c', '
    In_Out_Stor_idIn_Out_Stor'),
195 INDEX 'fk_Storage_User1_idx' ('User_idUser' ASC, '
    User_Login_idLogin' ASC, 'User_Events_idEvents' ASC, '
    User_Cannil_idCannil' ASC, 'User_In_Out_Can_id_inout_c'
    ASC) VISIBLE,
196 INDEX 'fk_Storage_In_Out_Stor1_idx' ('
    In_Out_Stor_idIn_Out_Stor' ASC) VISIBLE,
197 CONSTRAINT 'fk_Storage_User1'
198 FOREIGN KEY ('User_idUser', 'User_Login_idLogin', '
    User_Events_idEvents', 'User_Cannil_idCannil', '
    User_In_Out_Can_id_inout_c')
199 REFERENCES 'mydb'.'User' ('idUser', 'Login_idLogin', '
    Events_idEvents', 'Cannil_idCannil', '
    In_Out_Can_id_inout_c')
200 ON DELETE NO ACTION
201 ON UPDATE NO ACTION,
202 CONSTRAINT 'fk_Storage_In_Out_Stor1'
203 FOREIGN KEY ('In_Out_Stor_idIn_Out_Stor')
204 REFERENCES 'mydb'.'In_Out_Stor' ('idIn_Out_Stor')
205 ON DELETE NO ACTION

```

```
206         ON UPDATE NO ACTION)
207 ENGINE = InnoDB;
208
209
210 SET SQL_MODE=@OLD_SQL_MODE;
211 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
212 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

9 Referências