

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 Modelo de entidades:	6
6	Conversão do Modelo de entidades para o Modelo Relacional	7
7	Criação de Tabelas:	8
8	Referências	13

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, morada, telefone 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.

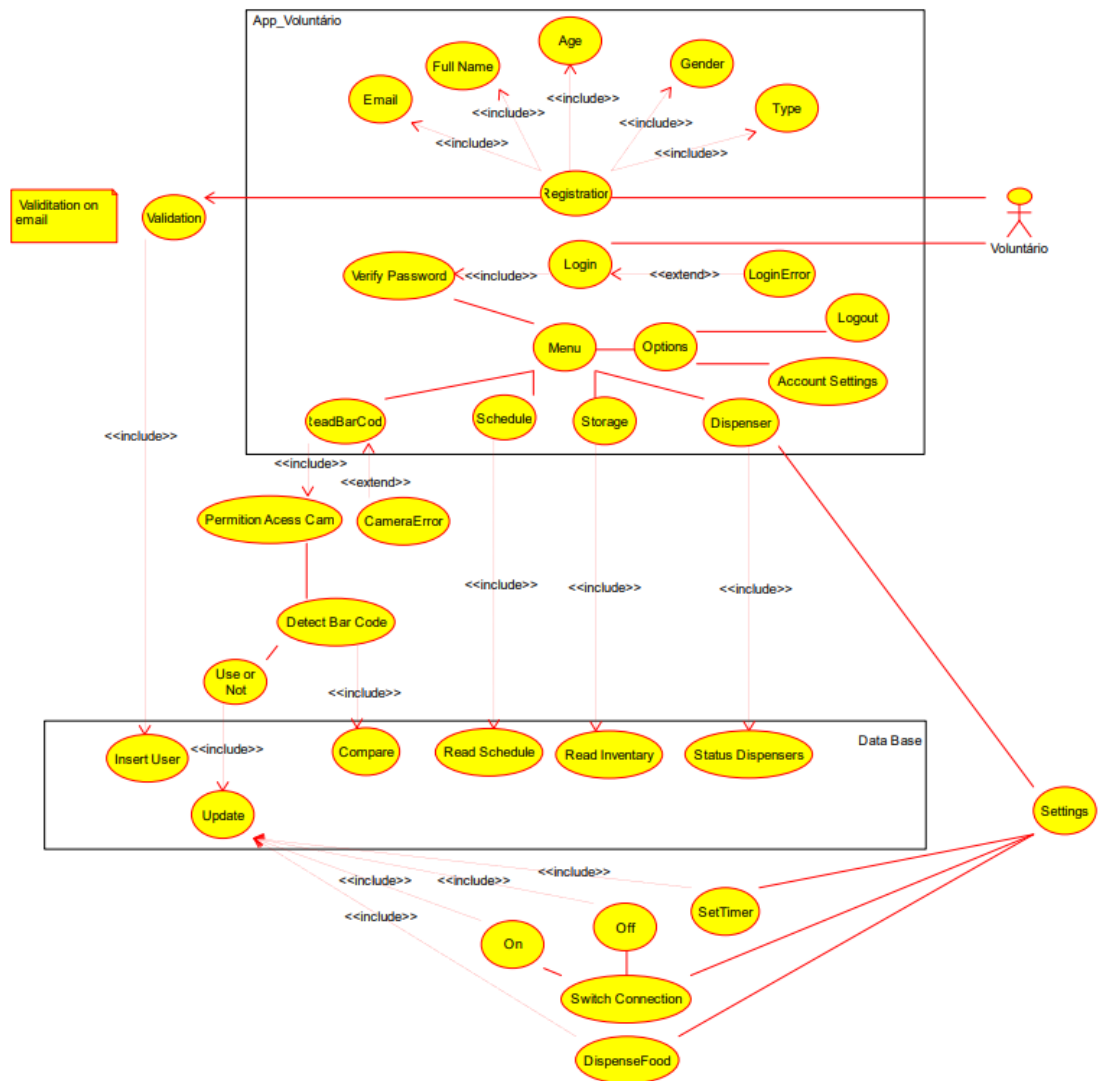
-Aceder ao calendário: O utilizador pode aceder ao calendário e ver as atividades futuras.

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

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

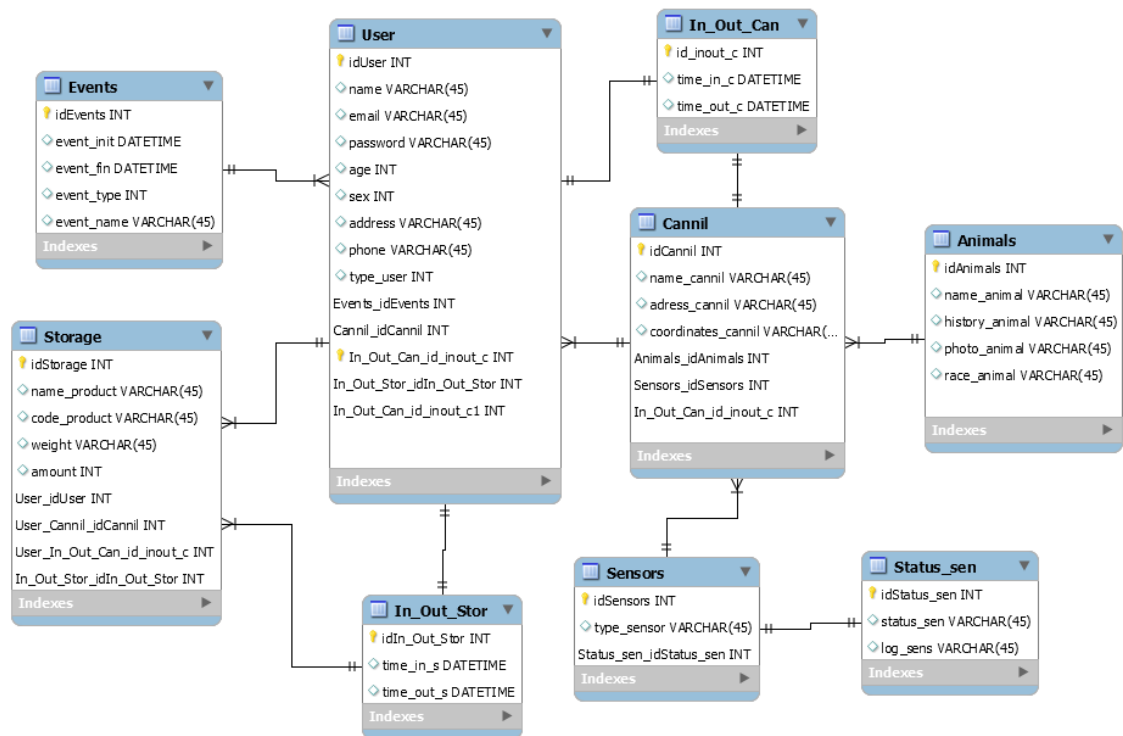
-Aceder ao dispensador: O utilizador pode controlar o dispensador a partir da aplicação.

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



5 Descrição do Modelo de entidades:

6 Conversão do Modelo de entidades para o Modelo Relacional



7 Criação de Tabelas:

De acordo com o modelo relacional é gerado o seguinte código SQL para criação das tabelas. Este processo é 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 relacional.

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

```

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

```

```

80  'Animals_idAnimals' INT NOT NULL,
81  'Sensors_idSensors' INT NOT NULL,
82  'In_Out_Can_id_inout_c' INT NOT NULL,
83  PRIMARY KEY ('idCannil', 'Animals_idAnimals', '
    Sensors_idSensors', 'In_Out_Can_id_inout_c'),
84  INDEX 'fk_Cannil_Annimals1_idx' ('Animals_idAnimals' ASC)
    VISIBLE,
85  INDEX 'fk_Cannil_Sensors1_idx' ('Sensors_idSensors' ASC)
    VISIBLE,
86  INDEX 'fk_Cannil_In_Out_Can1_idx' ('In_Out_Can_id_inout_c'
    ASC) VISIBLE,
87  CONSTRAINT 'fk_Cannil_Annimals1'
88      FOREIGN KEY ('Animals_idAnimals')
89      REFERENCES 'mydb'. 'Animals' ('idAnimals')
90      ON DELETE NO ACTION
91      ON UPDATE NO ACTION,
92  CONSTRAINT 'fk_Cannil_Sensors1'
93      FOREIGN KEY ('Sensors_idSensors')
94      REFERENCES 'mydb'. 'Sensors' ('idSensors')
95      ON DELETE NO ACTION
96      ON UPDATE NO ACTION,
97  CONSTRAINT 'fk_Cannil_In_Out_Can1'
98      FOREIGN KEY ('In_Out_Can_id_inout_c')
99      REFERENCES 'mydb'. 'In_Out_Can' ('id_inout_c')
100     ON DELETE NO ACTION
101     ON UPDATE NO ACTION)
102 ENGINE = InnoDB;
103
104
105 -----
106 -- Table 'mydb'. 'In_Out_Stor'
107 -----
108 CREATE TABLE IF NOT EXISTS 'mydb'. 'In_Out_Stor' (
109     'idIn_Out_Stor' INT NOT NULL,
110     'time_in_s' DATETIME NULL,
111     'time_out_s' DATETIME NULL,
112     PRIMARY KEY ('idIn_Out_Stor'))
113 ENGINE = InnoDB;
114
115
116 -----
117 -- Table 'mydb'. 'User'
118 -----
119 CREATE TABLE IF NOT EXISTS 'mydb'. 'User' (
120     'idUser' INT NOT NULL,
121     'name' VARCHAR(45) NULL,
122     'email' VARCHAR(45) NULL,
123     'password' VARCHAR(45) NULL,
124     'age' INT NULL,

```

```

125 'sex' INT NULL,
126 'address' VARCHAR(45) NULL,
127 'phone' VARCHAR(45) NULL,
128 'type_user' INT NULL,
129 'Events_idEvents' INT NOT NULL,
130 'Cannil_idCannil' INT NOT NULL,
131 'In_Out_Can_id_inout_c' INT NOT NULL,
132 'In_Out_Stor_idIn_Out_Stor' INT NOT NULL,
133 'In_Out_Can_id_inout_c1' INT NOT NULL,
134 PRIMARY KEY ('idUser', 'Events_idEvents', 'Cannil_idCannil',
    'In_Out_Can_id_inout_c', 'In_Out_Stor_idIn_Out_Stor', 'In_Out_Can_id_inout_c1'),
135 INDEX 'fk_User_Events1_idx' ('Events_idEvents' ASC) VISIBLE,
136 INDEX 'fk_User_Cannil1_idx' ('Cannil_idCannil' ASC) VISIBLE,
137 INDEX 'fk_User_In_Out_Stor1_idx' ('In_Out_Stor_idIn_Out_Stor' ASC) VISIBLE,
138 INDEX 'fk_User_In_Out_Can1_idx' ('In_Out_Can_id_inout_c1' ASC) VISIBLE,
139 CONSTRAINT 'fk_User_Events1'
140 FOREIGN KEY ('Events_idEvents')
141 REFERENCES 'mydb'. 'Events' ('idEvents')
142 ON DELETE NO ACTION
143 ON UPDATE NO ACTION,
144 CONSTRAINT 'fk_User_Cannil1'
145 FOREIGN KEY ('Cannil_idCannil')
146 REFERENCES 'mydb'. 'Cannil' ('idCannil')
147 ON DELETE NO ACTION
148 ON UPDATE NO ACTION,
149 CONSTRAINT 'fk_User_In_Out_Stor1'
150 FOREIGN KEY ('In_Out_Stor_idIn_Out_Stor')
151 REFERENCES 'mydb'. 'In_Out_Stor' ('idIn_Out_Stor')
152 ON DELETE NO ACTION
153 ON UPDATE NO ACTION,
154 CONSTRAINT 'fk_User_In_Out_Can1'
155 FOREIGN KEY ('In_Out_Can_id_inout_c1')
156 REFERENCES 'mydb'. 'In_Out_Can' ('id_inout_c')
157 ON DELETE NO ACTION
158 ON UPDATE NO ACTION)
159 ENGINE = InnoDB;
160
161
162 -----
163 -- Table 'mydb'. 'Storage'
164 -----
165 CREATE TABLE IF NOT EXISTS 'mydb'. 'Storage' (
166 'idStorage' INT NOT NULL AUTO_INCREMENT,
167 'name_product' VARCHAR(45) NULL,

```

```

168 'code_product' VARCHAR(45) NULL,
169 'weight' VARCHAR(45) NULL,
170 'amount' INT NULL,
171 'User_idUser' INT NOT NULL,
172 'User_Cannil_idCannil' INT NOT NULL,
173 'User_In_Out_Can_id_inout_c' INT NOT NULL,
174 'In_Out_Stor_idIn_Out_Stor' INT NOT NULL,
175 PRIMARY KEY ('idStorage', 'User_idUser', '
    User_Cannil_idCannil', 'User_In_Out_Can_id_inout_c', '
    In_Out_Stor_idIn_Out_Stor'),
176 INDEX 'fk_Storage_User1_idx' ('User_idUser' ASC, '
    User_Cannil_idCannil' ASC, 'User_In_Out_Can_id_inout_c'
    ASC) VISIBLE,
177 INDEX 'fk_Storage_In_Out_Stor1_idx' ('
    In_Out_Stor_idIn_Out_Stor' ASC) VISIBLE,
178 CONSTRAINT 'fk_Storage_User1'
179 FOREIGN KEY ('User_idUser', 'User_Cannil_idCannil', '
    User_In_Out_Can_id_inout_c')
180 REFERENCES 'mydb'. 'User' ('idUser', 'Cannil_idCannil',
    'In_Out_Can_id_inout_c')
181 ON DELETE NO ACTION
182 ON UPDATE NO ACTION,
183 CONSTRAINT 'fk_Storage_In_Out_Stor1'
184 FOREIGN KEY ('In_Out_Stor_idIn_Out_Stor')
185 REFERENCES 'mydb'. 'In_Out_Stor' ('idIn_Out_Stor')
186 ON DELETE NO ACTION
187 ON UPDATE NO ACTION)
188 ENGINE = InnoDB;
189
190
191 SET SQL_MODE=@OLD_SQL_MODE;
192 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
193 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

8 Referências