

# Information Systems and Databases



## **PROJECT ASSIGNMENT**

### **PART III**

#### **Group 32**

75572 - Orlando Bastos Vaz

75637 - Inês de Miranda de Matos Lourenço

75988 - João Pedro Beirão

## Introdução

Para a realização da 3ª parte deste laboratório foi utilizada uma vez mais a base de dados criada na parte anterior. Após a criação das tabelas, foram inseridos alguns dados, estando os mais relevantes descritos nos pontos em baixo.

- Na tabela *patient*, foram inseridos 4 pacientes do centro de saúde como mostrado na Figura 1, através dos seguintes comandos:

```
insert into patient values ('P-001', 'Cristina Gomes', '1990-01-01', 'Lisboa');
```

```
insert into patient values ('P-002', 'Joana Silva', '1990-01-02', 'Porto');
```

```
insert into patient values ('P-003', 'Andrea Miranda', '1985-07-22', 'Faro');
```

```
insert into patient values ('P-004', 'Joana Barbosa', '1981-11-12', 'Faro');
```

```
mysql> select * from patient;
+-----+-----+-----+-----+
| patient_id | name_pat      | birthday   | address |
+-----+-----+-----+-----+
| P-001      | Cristina Gomes | 1990-01-01 | Lisboa  |
| P-002      | Joana Silva    | 1990-01-02 | Porto   |
| P-003      | Andrea Miranda | 1985-07-22 | Faro    |
| P-004      | Joana Barbosa  | 1981-11-12 | Faro    |
+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

Figura 1 - Pacientes introduzidos na base de dados do centro de saúde.

- Na tabela *doctor* da Figura 2, foram inseridos 3 doutores disponíveis para dar consultas:

```
insert into doctor values ('D-001', 'Orlanda', 'Cardiologia');
```

```
insert into doctor values ('D-002', 'Ricardo', 'Radiologia');
```

```
insert into doctor values ('D-003', 'Fernanda', 'Imagiologia');
```

```
mysql> select * from doctor;
+-----+-----+-----+
| doctor_id | name_doc | specialty   |
+-----+-----+-----+
| D-001     | Orlanda  | Cardiologia |
| D-002     | Ricardo  | Radiologia  |
| D-003     | Fernanda | Imagiologia |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Figura 2 - Médicos introduzidos na base de dados do centro de saúde.

- Também uma primeira consulta foi já criada, na tabela “appointment” na Figura 3.

*insert into appointment values ('P-001', 'D-001', '2016-10-04 16:00:01', 'A1');*

```
mysql> select * from appointment;
+-----+-----+-----+-----+
| patient_id | doctor_id | date_ap           | office |
+-----+-----+-----+-----+
| P-001      | D-001     | 2016-10-04 16:00:01 | A1     |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Figura 3 - Consultas introduzidas na base de dados do centro de saúde.

## Tarefa 1

Quando um paciente liga para o centro de saúde com o objectivo de marcar uma consulta médica, o recepcionista da secretaria pode aceder à base de dados do centro de saúde através do seu browser, e deparar-se-á com a seguinte página inicial, implementada no ficheiro “proj3.php” e representada na Figura 4.

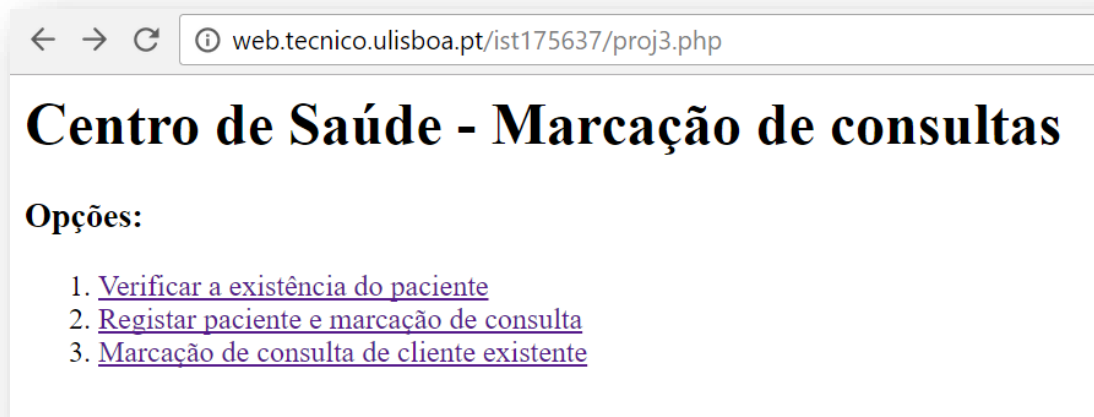


Figura 4 - Página inicial.

Neste momento, o recepcionista tem a possibilidade de fazer uma das 3 acções inseridas em cada um dos links. Note-se que, daqui em diante, no final de cada página Web haverá um link que permitirá ao utilizador regressar a esta página inicial, de modo a que este possa realizar uma outra operação. Seguindo o funcionamento descrito no enunciado, caso se deseje saber se o paciente já existe na base de dados do centro de saúde, o recepcionista deve carregar no primeiro link, e será reencaminhado para a uma outra página (Figura 5), onde poderá pesquisar o nome do paciente, e verificar se este já se encontra inserido na base de dados.

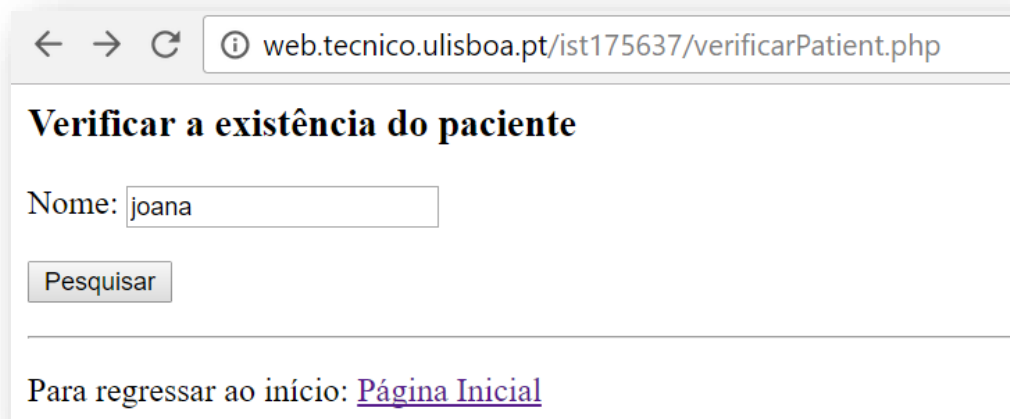


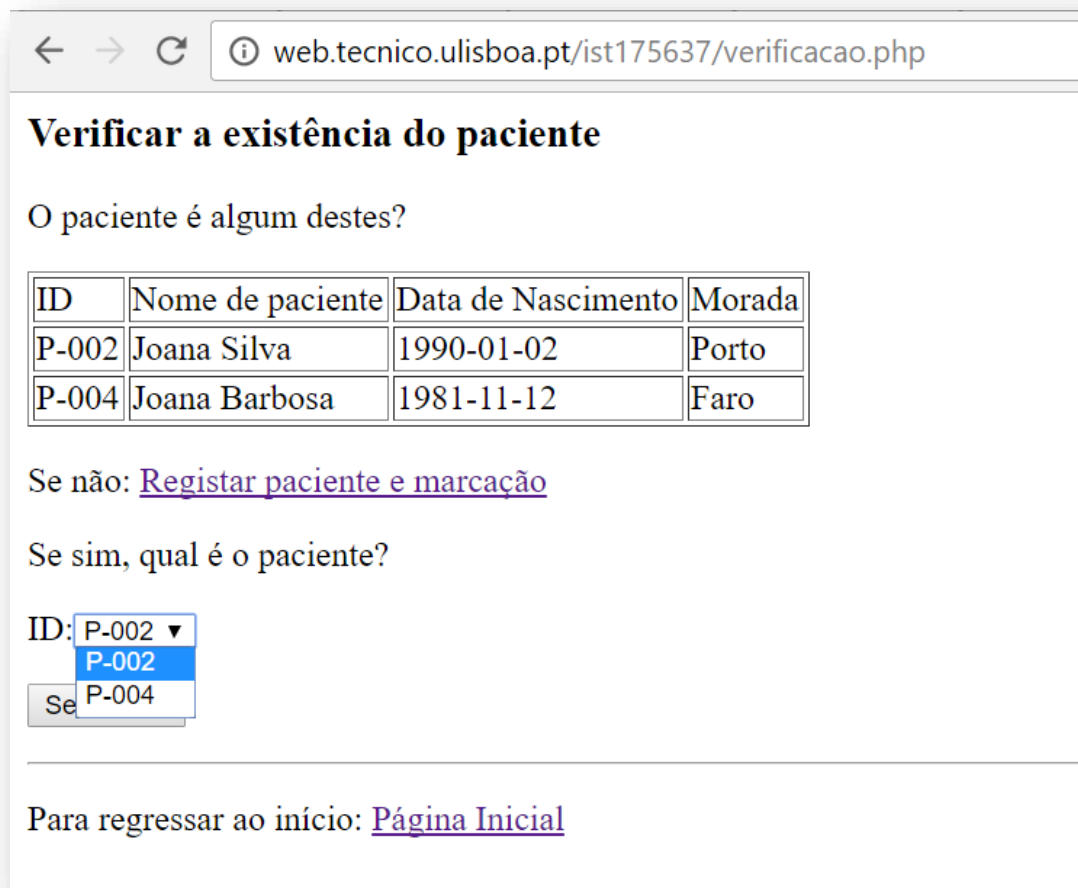
Figura 5 - Pesquisa de paciente por nome.

Escrevendo o nome “joana” (note-se que, de modo a facilitar a utilização e minimizar os erros de procura, o programa não é sensível a maiúsculas e minúsculas), e carregando no botão “Pesquisar”, uma nova página Web é carregada com os resultados do query realizado:

```
$sql = "SELECT * FROM patient WHERE name_pat like '%$patient_name'";
```

Esta página é gerada através do ficheiro “verificacao.php”, que testa a existência de nomes de pacientes na base de dados.

Neste caso, se já existirem na base de dados pacientes cujo nome contenha a palavra “joana”, esses resultados serão apresentados numa tabela como representado na Figura 6.



**Verificar a existência do paciente**

O paciente é algum destes?

ID	Nome de paciente	Data de Nascimento	Morada
P-002	Joana Silva	1990-01-02	Porto
P-004	Joana Barbosa	1981-11-12	Faro

Se não: [Registrar paciente e marcação](#)

Se sim, qual é o paciente?

ID:

Se

Para regressar ao início: [Página Inicial](#)

Figura 6 - Verificação da existência de pacientes com o nome pesquisado.

Para evitar que o rececionista escolha o paciente errado de entre vários pacientes com o mesmo nome, é apresentada informação adicional: ID, data de nascimento e morada.

Duas situações podem ocorrer:

- 1) O paciente está na lista apresentada. Quando o rececionista confirmar, através da informação fornecida, qual dos pacientes é o desejado, o mesmo deve escolher na *drop-down list* qual é o ID do paciente em questão, e em seguida pressionar o botão “Seleccionar”.

- 2) O paciente em questão não existe na base de dados. Neste caso, um novo registo terá de ser criado constituindo uma nova entrada na base de dados do centro de saúde. Mais especificamente, as informações de um novo paciente terão que ser introduzidas como uma nova linha na tabela "Patient", criada no laboratório anterior.

## Tarefa 2

No caso da situação 1) acima descrita, em que o paciente em questão é um dos que já existe na base de dados, depois de o seleccionar, o recepcionista é reencaminhado para uma nova página Web, representada na Figura 7 e descrita no ficheiro “marcarAppPatient.php”. Esta página recebe como *hidden input* os dados do paciente (ID, nome, data de nascimento e morada) seleccionado na página anterior. Por outro lado, apresenta um *form* que permite a inserção dos dados necessários à criação de uma nova consulta. De notar que a data e hora da consulta são do tipo *date* e *time*, respectivamente; o consultório é seleccionado através de *radio buttons*; e o médico é escolhido a partir de uma *drop-down list* que apresenta todos os médicos existentes na base de dados por ordem alfabética.

← → ↻ ⓘ web.tecnico.ulisboa.pt/ist175637/marcarAppPatient.php

### Marcar consulta

**Dados do paciente:**

ID	Nome de paciente	Data de Nascimento	Morada
P-002	Joana Silva	1990-01-02	Porto

**Marcação da Consulta:**

Data:

Hora:

Consultório:

☐ A1 ☐ A2 ☐ A3 ☒ A4 ☐ A5 ☐ A6 ☐ A7 ☐ A8

Médico:

Fernanda D-003

Orlinda D-001

Ricardo D-002

Para regressar ao início: [Página Inicial](#)

Figura 7 - Formulário de marcação de consulta para paciente já existente.

Após o preenchimento de todos os campos, o rececionista pode submeter o formulário. Os dados são introduzidos na base de dados através do ficheiro “inserirPatientApp.php”. Para isso, as variáveis 'patient\_id', 'name\_pat', 'birthday' e 'address' são definidas como *hidden inputs* dado que não foram introduzidos pelo utilizador nesta página.

Para que a operação de marcação da consulta para este paciente seja realizada com sucesso, todos os campos têm obrigatoriamente de ser preenchidos.

Um pormenor importante a ter em conta aquando da marcação da consulta é a validade da informação da data da mesma. Desta forma, são impostas duas restrições na escolha da data:

- A data escolhida para a consulta (Figura 8) não pode ser igual nem anterior à data actual. No limite, poderá ser no próprio dia, pelo menos 1 minuto depois da hora actual. Se o rececionista introduzir uma data que não cumpra esta restrição, a mensagem da Figura 9 será gerada numa nova página após a submissão da marcação da consulta.

**Marcação da Consulta:**

Data:

Hora:

Constu

☐ A1

Médico

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

Para regressar ao início: [Página Inicial](#)

Figura 8 - Escolha da data da consulta.

← → ↻

**Registar o paciente e marcar consulta**

Preenchimento inválido.

A data da consulta tem de ser posterior ao dia de hoje.

Retroceda e tente novamente.

Para regressar ao início: [Página Inicial](#)

Figura 9 - Mensagem de erro quando as datas não são introduzidas corretamente.



- A consulta não pode ter lugar durante o fim-de-semana, seja sábado ou domingo. Caso o dia do mês escolhido calhe em qualquer um destes dias da semana, a mensagem de erro é a descrita na Figura 10.

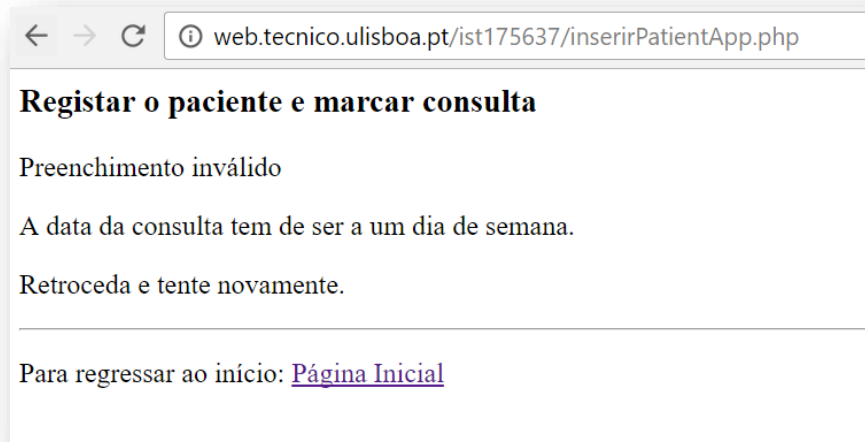


Figura 10 - Mensagem de erro apresentada quando o dia da consulta coincide com um fim de semana.

Apesar de tudo, a introdução dos dados na *query* é sujeita a uma validação prévia que tem como objectivo garantir que não são introduzidas linhas na base de dados com campos em falta. Por exemplo, uma consulta na qual não é introduzido o ID do paciente não é introduzida, visto que existe uma condição que verifica se algum campo foi deixado em branco (Figura 11).

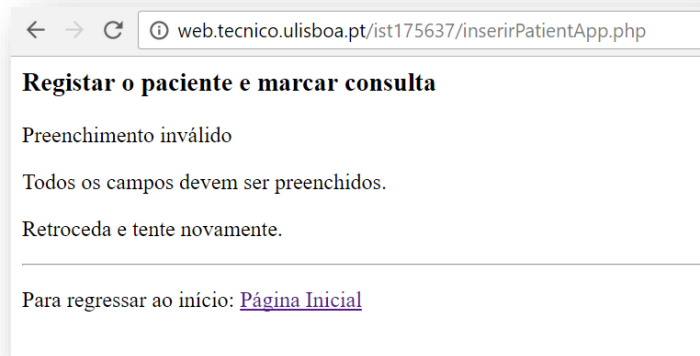


Figura 11 - Mensagem de erro apresentada quando pelo menos um campo do *form* não é preenchido.

As *queries* associadas à introdução dos dados do paciente e da consulta na base de dados são realizadas a partir de uma *transaction* que é controlada por uma *flag*. Esta *flag* foi concebida com o objetivo de usar um ficheiro único para introdução de apenas um paciente ou de um paciente e uma consulta em simultâneo. Neste caso, o paciente já se encontra na base de dados e, por isso, a *flag* tem o valor 0. Assim, o *insert* na tabela *appointment* é efetivamente realizado através do comando *commit*, e pode ser abortado (em caso de erro) a partir do comando *rollback*. Em todos os casos de erro descritos neste documento, o algoritmo faz *rollback* e nenhum dado é introduzido nas tabelas.

Com o objectivo de evitar que o programa seja *hackeado*, resolveu adicionar-se uma proteção contra *SQL injection* nos formulários, que consiste em dizer ao *script* o local exato para onde irá cada frase inserida nos *inputs* do formulário, da seguinte maneira:

```
$stmt = $connection->prepare("INSERT INTO appointment VALUES (:patient_id,  
:doctor_id, :totaldate, :office)");
```

```
$stmt->bindParam(':patient_id', $patient_id);
```

```
$stmt->bindParam(':doctor_id', $doctor_id);
```

```
$stmt->bindParam(':totaldate', $totaldate);
```

```
$stmt->bindParam(':office', $office);
```

```
$stmt->execute();
```

Por fim, caso todas as restrições sejam satisfeitas e caso todos os campos sejam preenchidos, o algoritmo faz *commit* e as informações são inseridas nas respetivas tabelas da base de dados.

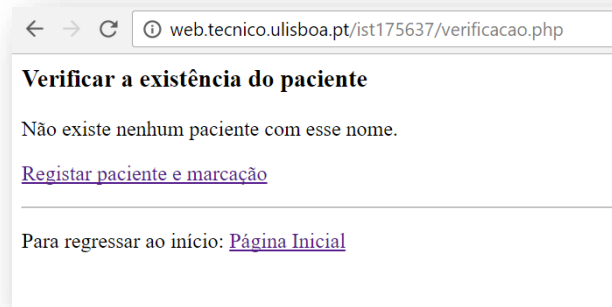
Desta forma, a Figura 12 representa a introdução de uma nova linha na tabela *appointment* na base de dados.

```
mysql> select * from appointment;  
+-----+-----+-----+-----+  
| patient_id | doctor_id | date_ap          | office |  
+-----+-----+-----+-----+  
| P-001      | D-001     | 2016-10-04 16:00:01 | A1     |  
| P-002      | D-002     | 2016-12-21 15:30:00 | A5     |  
+-----+-----+-----+-----+  
2 rows in set (0.01 sec)
```

Figura 12 - Tabela *appointment* com as novas consultas introduzidas.

### Tarefa 3

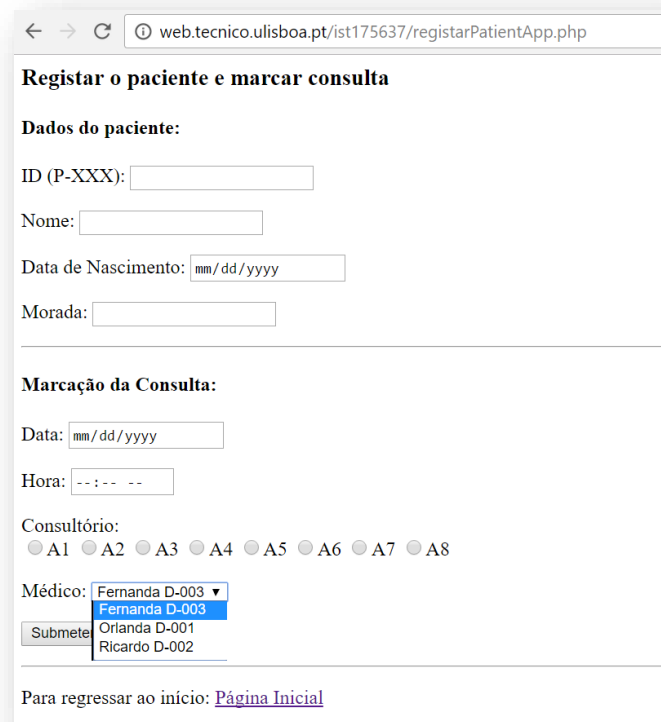
Caso o rececionista seja deparado com a situação 2), em que o paciente não existe na base de dados (como é o caso da pesquisa para o nome 'joão'), o *script* "verificacao.php" identifica que não há nenhuma linha da tabela *patient* que cumpra os requisitos do *query*, retornando a mensagem apresentada na Figura 13.



A screenshot of a web browser window. The address bar shows the URL "web.tecnico.ulisboa.pt/ist175637/verificacao.php". The page title is "Verificar a existência do paciente". The main content area displays the message "Não existe nenhum paciente com esse nome." followed by a blue underlined link "Registar paciente e marcação". At the bottom, it says "Para regressar ao início: [Página Inicial](#)".

Figura 13 - Mensagem apresentada caso o paciente não exista na base de dados.

Isto significa que o paciente em causa ainda não teve nenhuma consulta no centro de saúde, pelo que antes de ter a primeira deve ser registado na base de dados. Este procedimento é feito na página seguinte, para onde o rececionista é redireccionado a partir do link [Registar paciente e marcação](#), gerada pelo scrip "registarPatientApp.php".



A screenshot of a web browser window. The address bar shows the URL "web.tecnico.ulisboa.pt/ist175637/registarPatientApp.php". The page title is "Registar o paciente e marcar consulta". The form is divided into two main sections: "Dados do paciente:" and "Marcação da Consulta:". Under "Dados do paciente:", there are input fields for "ID (P-XXX):", "Nome:", "Data de Nascimento:" (with a date picker showing mm/dd/yyyy), and "Morada:". Under "Marcação da Consulta:", there are input fields for "Data:" (with a date picker showing mm/dd/yyyy) and "Hora:" (with a time picker showing --:--:--). Below these, there is a "Consultório:" section with radio buttons for A1 through A8. The "Médico:" section has a dropdown menu with "Fernanda D-003" selected, and a "Submeter" button. At the bottom, it says "Para regressar ao início: [Página Inicial](#)".

Figura 14 - Formulário de inscrição do paciente e da consulta (em simultâneo) na base de dados.

Neste caso, o *form* a preencher inclui os campos relacionados com o paciente (ID, nome, data de nascimento e morada), não necessitando de transmitir essas informações por *hidden inputs*. As restrições referidas anteriormente para o preenchimento do formulário relativo apenas à consulta, são igualmente válidas neste caso. Todo o processo referido acima repete-se uma vez que o ficheiro utilizado para a validação do formulário e posterior introdução na base de dados é o mesmo: “*inserirPatientApp.php*”. A única diferença consiste na variável *flag*, que assume o valor 1. Apenas neste caso a utilização do procedimento *transaction* é uma mais valia, visto que, neste caso existe a necessidade de inserir dados em duas tabelas diferentes: *patient* e *appointment*; e este método permite executar ou abortar várias ações em simultâneo.

## Outras funcionalidades

No caso da 2ª opção da Página Inicial (Figura 15) o link Registar paciente e marcação de consulta redireciona para a página relativa ao *script* “registarPatientApp.php” (Figura 14), onde podem ser adicionados à base de dados do centro de saúde um paciente e uma consulta. A 3ª opção, Marcação de consulta de cliente existente, introduz uma maneira de procurar determinado paciente a partir da lista completa de todos os pacientes existentes na base de dados, isto é, na tabela *patient*. Esta página está representada na Figura 15 e diz respeito ao *script* “marcarAppSemPatient.php”.

**Escolher paciente existente**

Pacientes:

ID	Nome de paciente	Data de Nascimento	Morada
P-003	Andrea Miranda	1985-07-22	Faro
P-001	Cristina Gomes	1990-01-01	Lisboa
P-004	Joana Barbosa	1981-11-12	Faro
P-002	Joana Silva	1990-01-02	Porto

Escolher paciente:

ID: P-001 ▼  
P-001  
P-002  
P-003  
P-004

Para regressar ao início: [Página Inicial](#)

Figura 15 - Seleção de um paciente a partir da listagem de todos os pacientes existentes na base de dados.

Ao analisar a lista de todos os pacientes, o rececionista pode escolher o ID daquele a quem quer marcar uma consulta. Neste *script*, implementou-se o mesmo formulário que no *script* “verificacao.php”, representado na Figura 6, sendo que a sua submissão redireciona tal como antes o rececionista para a página correspondente ao *script* “marcarAppPatient.php”, da Figura 7.