



Bruno Manuel  
de Moura Ramos

Sistema de Recolha e Armazenamento Remoto  
de Informação Sensorial de um Processo  
Industrial usando Bases de Dados Múltiplas

# DOCUMENTO PROVISORIO





**o juri/the jury**

presidente/president

**ABC**

Professor Catedratico da Universidade de Aveiro (por delegacao da Reitora da Universidade de Aveiro)

vogais/examiners committee

**DEF**

Professor Catedratico da Universidade de Aveiro (orientador)

**GHI**

Professor associado da Universidade J (co-orientador)

**KLM**

Professor Catedratico da Universidade N



agradecimientos /  
acknowledgements

ergergerg

ergergerg



## Resumo

Moldes de injeção têm uma vasta aplicabilidade no mundo industrial. Afim de melhorar a qualidade do produto final e reduzir falhas surgiu a necessidade de instrumentar e monitorizar moldes remotamente. Neste projeto desenvolveu-se uma rede de bases de dados relacionais e uma aplicação em ambiente *Web*. A primeira garante uma transferência de valores segura e permanente de forma a criar um histórico. A segunda permite ao utilizador interagir com as bases de dados desenvolvidas podendo editá-las e consultá-las de forma a gerar relatórios.

No desenvolvimento deste projeto utilizou-se *MySQL* e *C* para criar e definir a rede de bases de dados resultando numa solução simples e funcional a baixo nível. Utilizou-se *Apache*, *PHP*, e *HTML* para desenvolver e implementar a aplicação de forma a que esta seja multiplataforma e garanta um acesso remoto ao utilizador.

A solução proposta cumpre todos os objetivos definidos podendo ser já utilizada numa fase experimental apesar de necessitar alguns melhoramentos a nível de desempenho.





## **Abstract**

Nowadays, it is usual to evaluate a work . . .



# Conteúdo

<b>Conteúdo</b>	<b>i</b>
<b>Lista de Figuras</b>	<b>iii</b>
<b>Lista de Tabelas</b>	<b>v</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Estado de Arte</b>	<b>3</b>
2.1 Moldes de injeção . . . . .	3
2.2 Base de dados relacional . . . . .	5
2.2.1 Análise de requisitos . . . . .	5
2.2.2 Diagrama Entidade Relação . . . . .	5
2.2.3 Esquema Relacional . . . . .	5
2.3 SQL . . . . .	5
2.4 Software utilizado . . . . .	5
2.4.1 Linux . . . . .	5
2.4.2 MySQL . . . . .	5
2.4.3 C . . . . .	5
2.4.4 Apache . . . . .	5
2.4.5 PHP e HTML . . . . .	5
<b>3 Proposta de Solução</b>	<b>7</b>
3.1 Infraestrutura de dados . . . . .	7
3.2 Base de Dados . . . . .	7
3.2.1 Análise de Requisitos . . . . .	7
3.2.2 Desenho conceptual e esquema lógico . . . . .	7
3.2.3 Construção da base de dados . . . . .	7
3.2.4 Programa de transferência . . . . .	7
3.2.5 Gestão de <i>backups</i> . . . . .	7
3.2.6 Simulador . . . . .	7
3.2.7 Utilizadores . . . . .	7
<b>4 Aplicação</b>	<b>9</b>
4.1 Adaptação da infraestrutura . . . . .	9
4.2 Interface gráfica . . . . .	10
4.2.1 <i>Main</i> . . . . .	11

4.2.2	<i>Login</i> . . . . .	12
4.2.3	Consultas . . . . .	13
4.2.4	Administração . . . . .	16
4.2.5	Conexão local . . . . .	21
<b>5</b>	<b>Instalação do Sistema</b>	<b>25</b>
<b>6</b>	<b>Conclusões</b>	<b>27</b>
6.1	Comentários . . . . .	27
6.1.1	Infraestrutura de dados . . . . .	27
6.1.2	Aplicação . . . . .	27
6.2	Trabalhos Futuros . . . . .	28
	<b>Bibliografia</b>	<b>31</b>

# Lista de Figuras

2.1	Molde de injeção aberto com a peça realizada[1]	3
4.1	Esquema ligação Aplicação-bases de dados. A aplicação comunica com a base de dados temporária local e depois regista os valores desta nas bases de dados central e local	9
4.2	Funcionalidades da página <i>Main</i> com e sem <i>login</i>	11
4.3	Página de <i>Login</i> para iniciar sessão na base de dados central	12
4.4	Exemplos de erros retornados quando introduzidas credenciais não válidas na página <i>Login</i>	12
4.5	Página de Consultas e exemplo de resposta a uma consulta	13
4.6	Exemplos de erros retornados na página de Resposta da Consulta	15
4.7	Funcionalidades da página Administração com e sem conexão local	16
4.8	Área de Gestão de Clientes sem conexão local	17
4.9	Área de Gestão de Moldes	18
4.10	Área de Gestão de Sensores	18
4.11	Função do botão Validar, onde os valores da base de dados temporária local são transferidos de forma permanente para as bases de dados central e local	19
4.12	Área de Gestão de Clientes após conexão local onde se visualiza o botão Atualizar	20
4.13	Área Conectar Local onde se visualiza as bases de dados instaladas no sistema	21
4.14	<i>Main</i> com conexão local	22
4.15	Área Criar Local cria a base de dados para o cliente selecionado no sistema em que se acede a aplicação	22
4.16	<i>Queries</i> geradas para completar a instalação da base de dados no sistema local. Consistem nas permissões para os utilizadores que só podem ser garantidas via <i>root</i> bem como informação para completar o sistema de dados	23
4.17	Área Instalar <i>MySQL</i> para instalar o <i>software</i> com os comandos em <i>Linux</i>	24
6.1	Área de Criar Local com criação via <i>root</i> em vez de gerar <i>queries</i> para o utilizador inserir manualmente no <i>MySQL</i>	28
6.2	Exemplo de área de Registar para criar utilizadores com base no ID de trabalhador e email	29



# Lista de Tabelas





# Capítulo 1

## Introdução

arquivo e monitorização de moldes.



## Capítulo 2

# Estado de Arte

### 2.1 Moldes de injeção

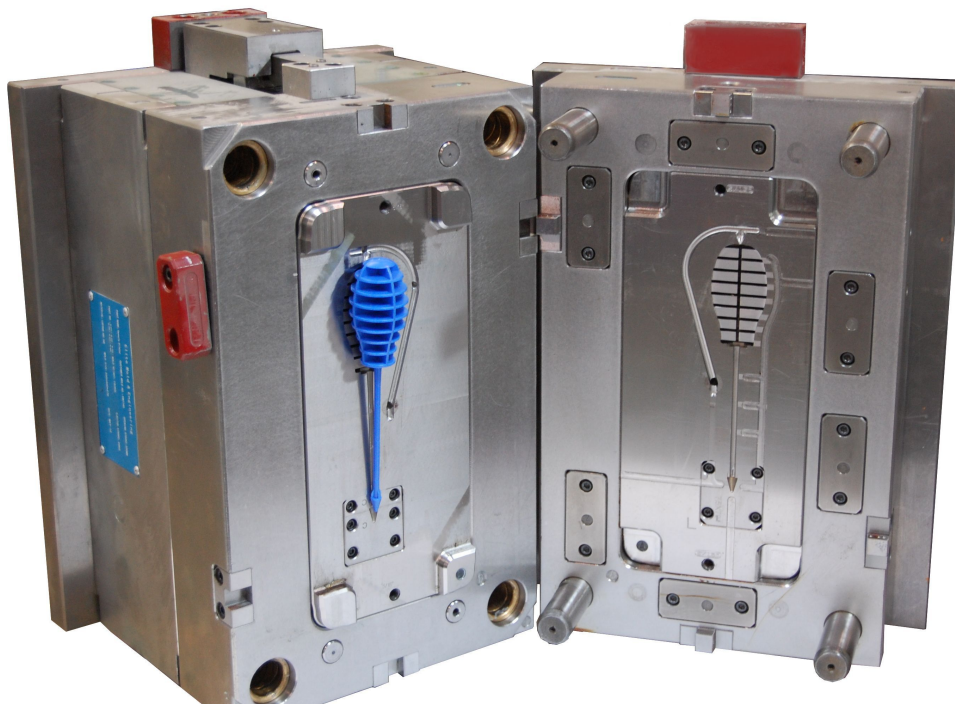


Figura 2.1: Molde de injeção aberto com a peça realizada[1]

Moldagem é o processo mecânico de dar forma a um material no estado líquido usando um molde[2, 3]. Um molde é uma ferramenta sólida oca desenvolvida para efeitos de fundição[4]. Esta é enchida com um material em estado líquido ou em pó como plástico, metal, cerâmica ou vidro[5, 6, 7, 8].

A moldagem por injeção é particularmente útil na produção em massa de peças de plástico com elevada complexidade geométrica[9, 10]. Estas podem ser encontradas em todas as áreas da indústria como por exemplo empacotamento, aviação, construção e eletrónica[11]. Neste

processo, plástico quente é forçado para dentro de um molde frio com a forma desejada. O material cobre todas as feições do molde e vai endurecendo sobre o efeito de altas pressões. O processo de injeção pode ser dividido em fases distintas[9]:

- Fecho do molde
- Enchimento
- Compactação
- Abertura do molde
- Extração

Em 1868, John Wesley Hyatt inventou uma maneira de fazer bolas de bilhar injetando celuloide para dentro de um molde[12, 13].

Em 1872 John e o seu irmão Isaiah patentearam a primeira máquina de moldes de injeção. Esta era relativamente simples comparada às que são usadas hoje em dia na indústria. Consistia de um pequeno embolo para injetar plástico num molde através de um cilindro quente[12, 14].

A indústria cresceu lentamente produzindo artigos de plástico como botões e pentes. Nos anos 40 a utilização de moldes de injeção cresceu por causa da Segunda Guerra Mundial que tinha uma grande procura de produtos baratos e produzidos em massa[12].

Em 1946, James Hendry construiu a primeira máquina de moldes de injeção com um sem fim, revolucionando a indústria dos plásticos com um design para substituir o embolo de Hyatt. Este sem fim é colocado dentro do cilindro e mistura o material a ser moldado antes de ser injetado no molde. Isto permitiu que cor ou plástico reciclado fossem adicionados à mistura[12, 15].

A qualidade de fabrico dos moldes e das peças produzidas evoluiu com o passar do tempo. A indústria investiu em técnicas sofisticadas no desenvolvimento de moldes, para que estes não contenham falhas, e no uso de materiais com qualidade elevada para evitar o desgaste da ferramenta. No entanto, dificilmente se atinge peças com a qualidade desejada unicamente através das ferramentas desenvolvidas. É necessário implementar também técnicas de monitorização de qualidade[16].

## **2.2 Base de dados relacional**

### **2.2.1 Análise de requisitos**

### **2.2.2 Diagrama Entidade Relação**

### **2.2.3 Esquema Relacional**

## **2.3 SQL**

## **2.4 Software utilizado**

### **2.4.1 Linux**

### **2.4.2 MySQL**

### **2.4.3 C**

### **2.4.4 Apache**

### **2.4.5 PHP e HTML**



## Capítulo 3

# Proposta de Solução

### 3.1 Infraestrutura de dados

### 3.2 Base de Dados

#### 3.2.1 Análise de Requisitos

#### 3.2.2 Desenho conceptual e esquema lógico

#### 3.2.3 Construção da base de dados

#### 3.2.4 Programa de transferência

#### 3.2.5 Gestão de *backups*

#### 3.2.6 Simulador

#### 3.2.7 Utilizadores





## Capítulo 4

# Aplicação

Aplicação desenvolvida em ambiente *Web* com o objetivo de ser multiplataforma, permitir acesso remoto e sem recorrer a instalação de *softwares* nos dispositivos dos utilizadores. Esta corre num servidor *Apache* e foi desenvolvida com *PHP* e *HTML*. Este capítulo descreve a adaptação da infraestrutura desenvolvida e as várias funcionalidades da aplicação.

### 4.1 Adaptação da infraestrutura

Afim de garantir uma maior integridade dos dados inseridos pela aplicação, instala-se no servidor local uma nova base de dados temporária local. Aqui os utilizadores têm a liberdade para adicionar, alterar e apagar informação sem consequências no sistema antes destas serem introduzidas nas bases de dados central e local como representado na Figura 4.1. Como referido anteriormente, esta base de dados difere das restantes, não contendo em si as tabelas fase e registos.

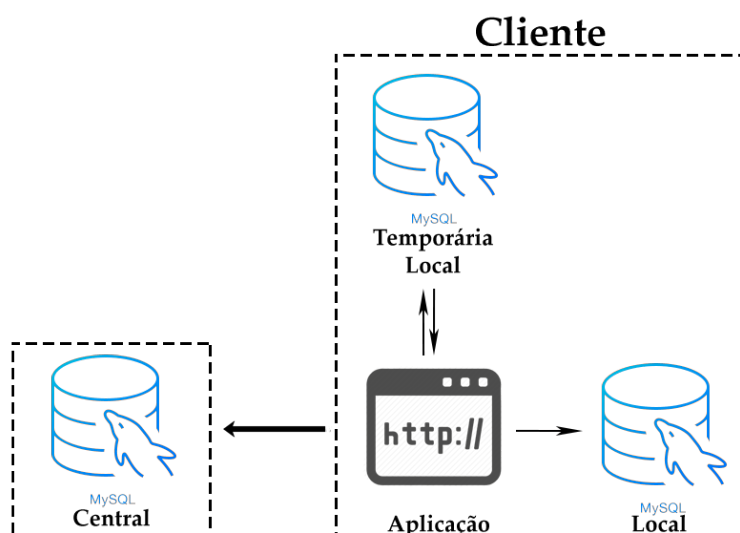


Figura 4.1: Esquema ligação Aplicação-bases de dados. A aplicação comunica com a base de dados temporária local e depois regista os valores desta nas bases de dados central e local

## 4.2 Interface gráfica

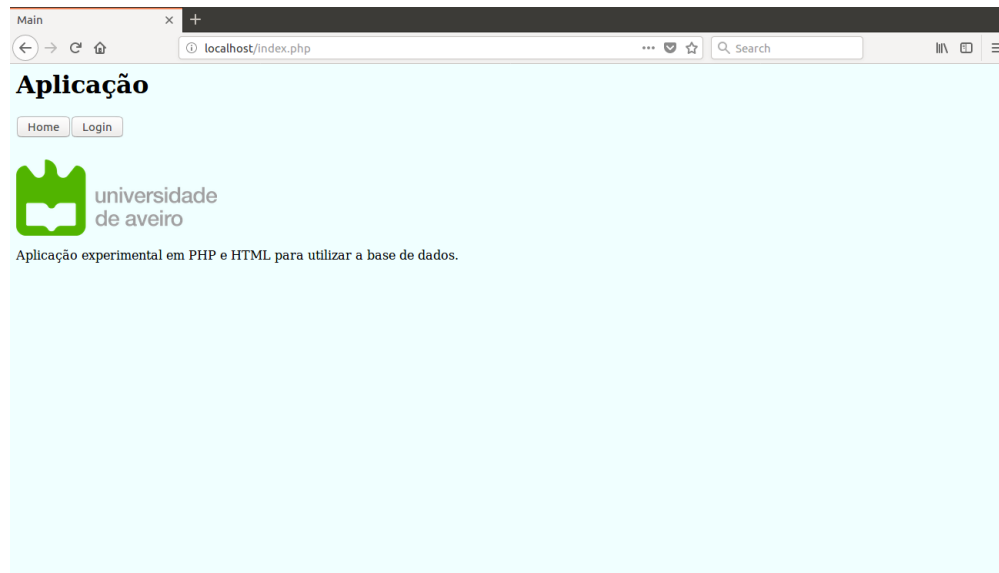
A aplicação divide-se em cinco partes distintas:

- *Main*
- *Login*
- Consultas
- Administração
- Conexão Local

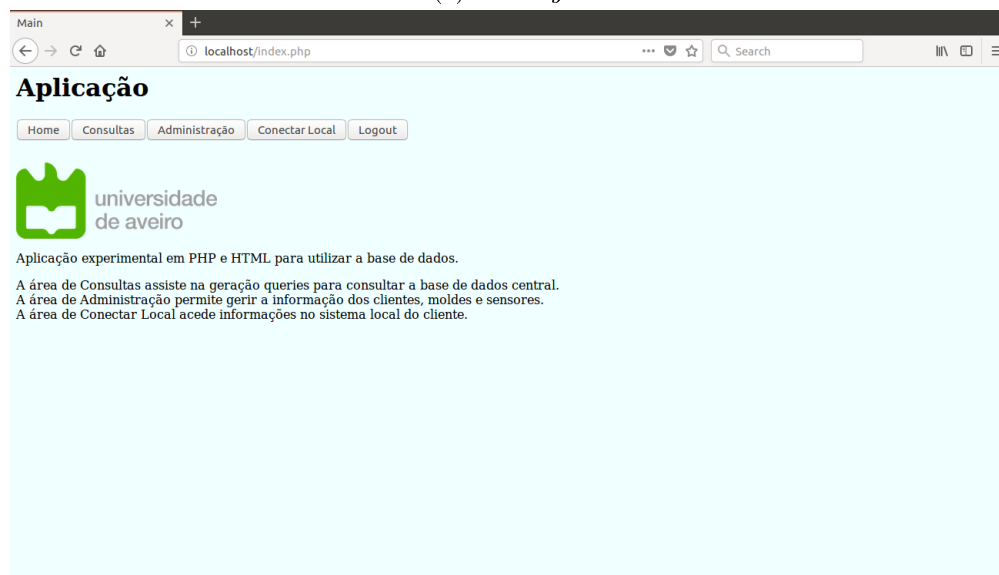
As páginas *Main*, *Login*, Consultas e parte das funcionalidades da Administração foram realizadas para uma utilização geral. As páginas Conexão Local e as restantes funcionalidades da Administração foram realizadas para uma utilização local. A primeira visa um uso a partir de qualquer dispositivo e acessível a qualquer momento e a segunda foca-se num acesso local com o objetivo de configurar e definir a informação no servidor local. Por outras palavras, para o utilizador usar as funcionalidades destas páginas tem de aceder à aplicação no sistema local que se situa no cliente.

Instalar um molde é culminar de um projeto de elevada responsabilidade, esta ideia junto com a criação da base de dados temporária local serve para melhorar a qualidade da informação introduzida no sistema e diminuir as falhas.

### 4.2.1 *Main*



(a) Sem *login*



(b) Com *login*

Figura 4.2: Funcionalidades da página *Main* com e sem *login*

*Main* serve como página principal da aplicação. Se não houver sessão iniciada todas as restantes páginas redirecionam o utilizador para aqui. Contém apenas algumas informações gerais sobre a aplicação.

Iniciar sessão na página de *Login* desbloqueia funcionalidades na aplicação, como demonstrado nas Figuras 4.2a e 4.2b. Depois de iniciada sessão navega-se com os botões para as páginas de Consultas, Administração e Conexão Local.

### 4.2.2 *Login*

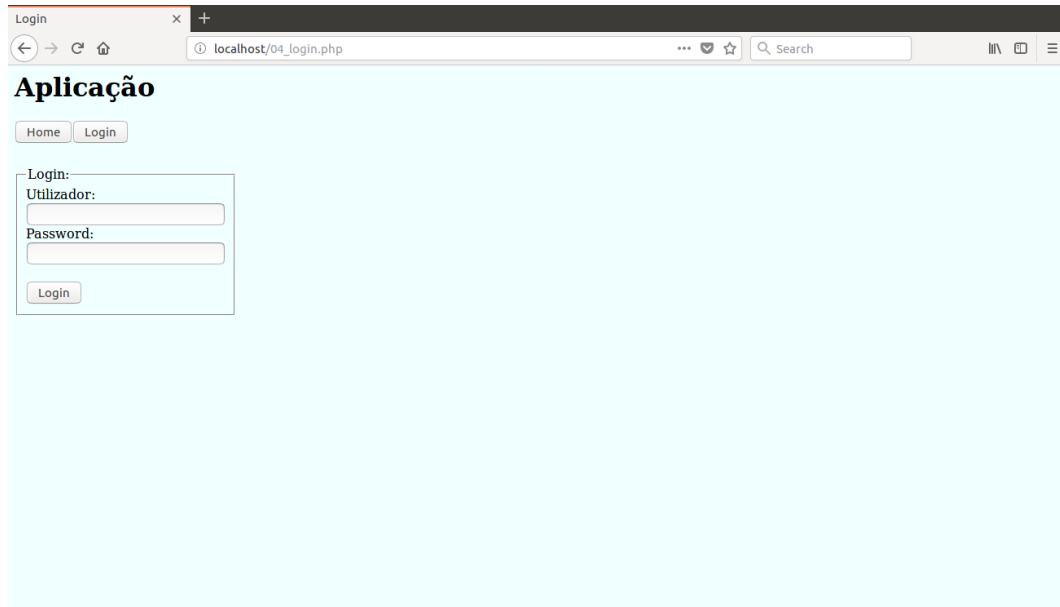
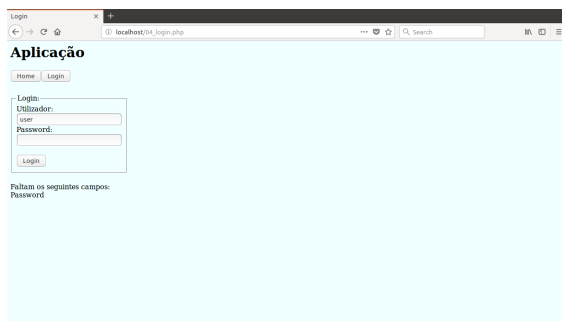


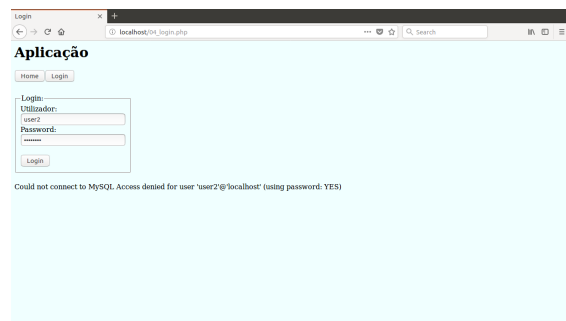
Figura 4.3: Página de *Login* para iniciar sessão na base de dados central

A página de *Login* consiste num simples formulário constituído por duas caixas de texto e um botão, como demonstrado na Figura 4.3. O botão *Login* lê as credenciais introduzidas e realiza uma conexão de teste à base de dados central validando-as diretamente com *MySQL*. Se as credenciais forem validadas com sucesso redireciona-se o utilizador para a página principal e altera-se o botão de *Login* para *Logout*. Se as credenciais introduzidas não forem suficientes ou válidas são retornados erros de forma a informar o utilizador como demonstrado nas Figuras 4.4a e 4.4b.

Quando se acede à página como *Logout* termina-se a sessão e redireciona-se o utilizador para a página principal.



(a) Exemplo de erro de falta de informação



(b) Exemplo de erro *MySQL*

Figura 4.4: Exemplos de erros retornados quando introduzidas credenciais não válidas na página *Login*

### 4.2.3 Consultas

Consultas

← → ↻ 🏠

localhost/02\_consultas.php

Search

## Aplicação

Home Consultas Administração Conectar Local Logout

Gerar Query

Escolher Atributos:

Clientes	Moldes	Sensores	Registos
<input type="checkbox"/> cl_ID	<input type="checkbox"/> m_ID	<input type="checkbox"/> s_num	<input type="checkbox"/> fase_nome
<input type="checkbox"/> cl_nome	<input type="checkbox"/> m_nome	<input type="checkbox"/> tipo_nome	<input type="checkbox"/> r_data_hora
<input type="checkbox"/> cl_morada	<input type="checkbox"/> m_descricao	<input type="checkbox"/> s_nome	<input type="checkbox"/> r_milissegundos
<input type="checkbox"/> cl_IP		<input type="checkbox"/> s_descricao	<input type="checkbox"/> r_valor
<input type="checkbox"/> cl_port			

Adicionar Condições:

Filtros:

Ordenar:

Ou:

Query:

Gerar Query

(a) Página de Consultas

Consultas Resultado

← → ↻ 🏠

localhost/resultado\_consulta.php?cl\_ID=cl\_ID&m\_ID=m\_ID&s\_num=s\_num&tipo=tipo\_nome

Search

Query anterior: SELECT cl\_ID, m\_ID, s\_num, tipo\_nome FROM clientes INNER JOIN moldes ON cl\_ID = m\_IDCliente INNER JOIN sensores ON m\_ID = s\_IDMolde INNER JOIN tipo ON s\_tipo = tipo\_ID WHERE m\_ID = 9001 OR m\_ID = 9009

10001	9001	1	Termómetro
10001	9001	2	Dinamómetro
10001	9001	3	Dinamómetro
10001	9001	4	Dinamómetro
10001	9001	5	Dinamómetro
10004	9009	1	Termómetro
10004	9009	2	Termómetro
10004	9009	3	Dinamómetro
10004	9009	4	Dinamómetro
10004	9009	5	Extensómetro
10004	9009	6	Extensómetro
10004	9009	7	Vibrómetro
10004	9009	8	Vibrómetro
10004	9009	9	Pressão
10004	9009	10	Pressão

(b) Exemplo de resposta de consulta

Figura 4.5: Página de Consultas e exemplo de resposta a uma consulta

A página de Consultas assiste utilizadores sem conhecimentos de *SQL* a criarem *queries* para consultar a base de dados central. Na Figura 4.5a observa-se várias *checkboxes* e três caixas de texto. As *checkboxes* permitem selecionar os atributos que se desejam consultar na base de dados, estes são guardados numa variável @atributos.

Quando se prime o botão *Query* gera-se uma das seguintes *queries*:

```
SELECT @atributos  
FROM clientes;
```

```
SELECT @atributos  
FROM clientes  
INNER JOIN moldes ON cl_ID = m_IDCliente;
```

```
SELECT @atributos  
FROM clientes  
INNER JOIN moldes ON cl_ID = m_IDCliente  
INNER JOIN sensores ON m_ID = s_IDMolde  
INNER JOIN tipo ON s_tipo = tipo_ID;
```

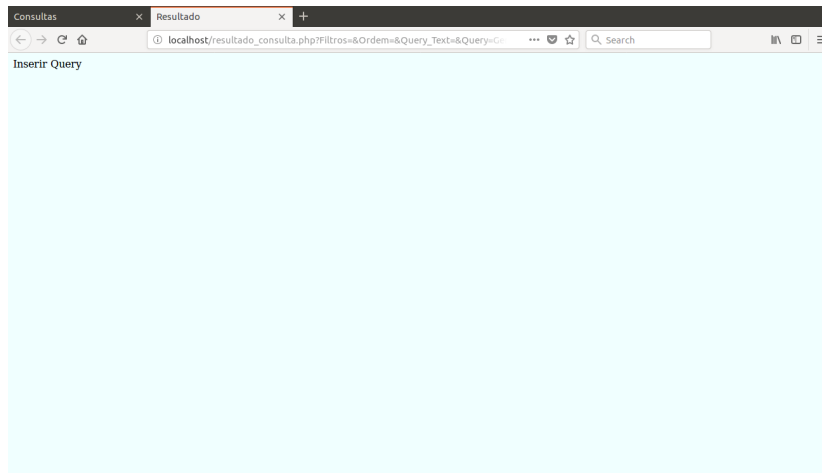
```
SELECT @atributos FROM clientes  
INNER JOIN moldes ON cl_ID = m_IDCliente  
INNER JOIN sensores ON m_ID = s_IDMolde  
INNER JOIN tipo ON s_tipo = tipo_ID  
INNER JOIN registros ON s_IDMolde = r_IDMolde  
AND s_num = r_numSensor  
INNER JOIN fase ON r_fase = fase_ID;
```

Efetua-se esta seleção com base na coluna mais à esquerda a que os atributos pertencem. Explicando melhor com um exemplo: se o utilizador desejar consultar o *cl\_ID* e o *cl\_nome* da tabela *clientes* gera-se a primeira *query* no entanto, se o utilizador desejar consultar os atributos *cl\_ID*, *m\_ID* e *s\_num* gera-se a terceira *query*.

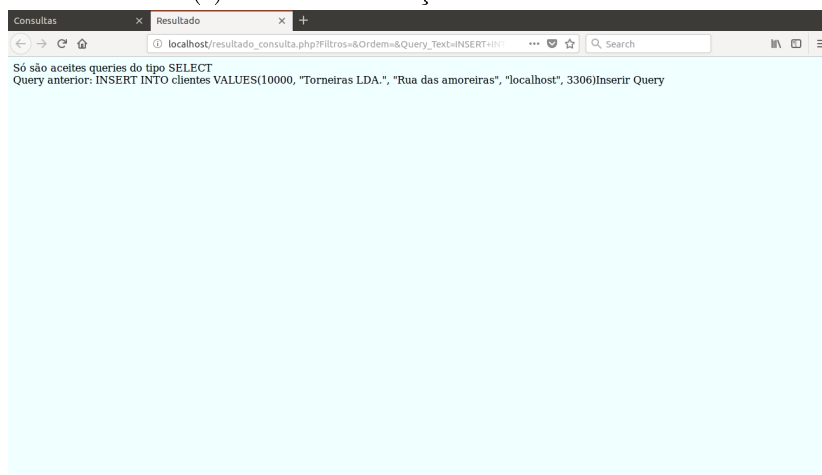
Além destas, existem três *queries* específicas quando os atributos *tipo\_nome*, *fase\_nome* e *r\_data\_hora* são selecionados sozinhos. As primeiras duas permitem consultar as opções disponíveis nos dicionários e a terceira devolve as datas e horas entre o primeiro e último registros. As caixas de texto *Filtros* e *Ordem* permitem adicionar às *queries* geradas as cláusulas *WHERE* e *ORDER BY*, respetivamente. Para os utilizadores com conhecimentos em *SQL* está disponibilizada a caixa de texto *Query* que permite a criação direta de uma *query*. Este campo está limitado apenas para *queries* do tipo *SELECT*.

Depois da *query* ser gerada retorna-se uma resposta num novo separador como demonstrado na Figura 4.5b. O *link* deste resposta contém toda a informação da *query* gerada. Este pode ser arquivado ou enviado para outro utilizador sem ser necessário gerar a *query* novamente, isto é útil para *queries* com muitas cláusulas.

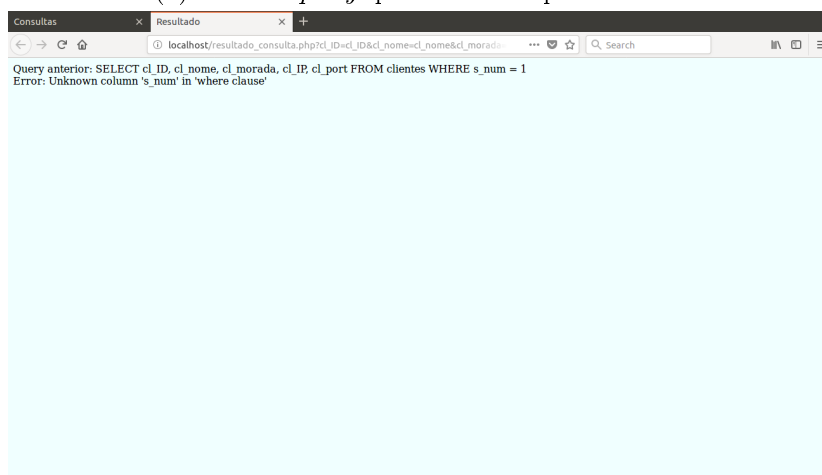
Se a *query* não for válida retorna-se um erro de forma a informar o utilizador, como demonstrado nas Figuras 4.6a, 4.6b e 4.6c.



(a) Erro de informação não introduzida



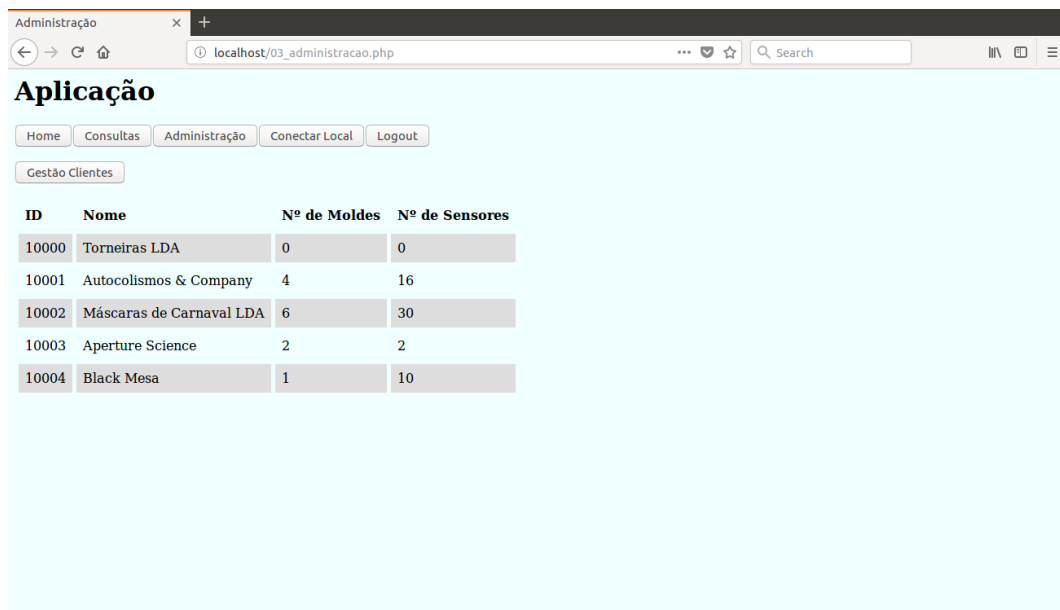
(b) Erro de *query* que não é do tipo SELECT



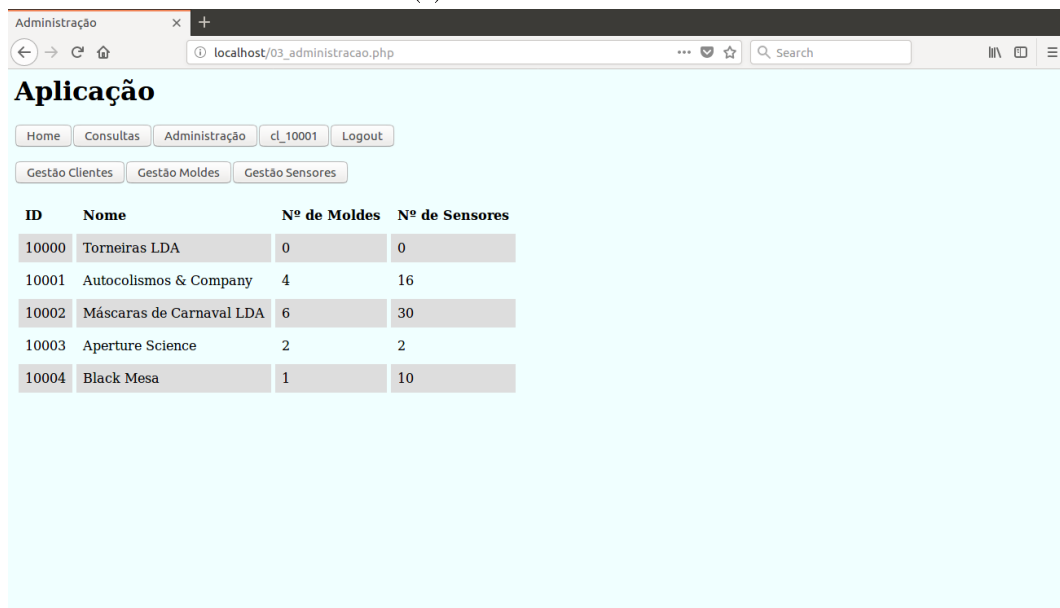
(c) Exemplo de erro *MySQL*

Figura 4.6: Exemplos de erros retornados na página de Resposta da Consulta

#### 4.2.4 Administração



(a) Sem conexão local



(b) Com conexão local

Figura 4.7: Funcionalidades da página Administração com e sem conexão local



A área de Administração permite ao utilizador alterar informações sobre os clientes, moldes e sensores. A partir de qualquer dispositivo só é possível aceder à Gestão de Clientes como demonstrado na Figura 4.7a. Nesta área a informação dos clientes pode ser alterada com o formulário demonstrado na Figura 4.8. Os botões Adicionar Cliente, Alterar Cliente e Eliminar Cliente executam *queries* do tipo INSERT, UPDATE e DELETE, respetivamente.

**Administração**

Home Consultas Administração Conectar Local Logout

Gestão Clientes

Cliente:

ID:

Nome:

Morada:

IP:

Port:

Adicionar Cliente Alterar Cliente Eliminar Cliente

Ver Clientes

ID	Nome	Morada	IP	Port	Nº de Moldes	Nº de Sensores
10000	Torneiras LDA	Rua das amoreiras	127.0.0.1	3306	0	0
10001	Autocolismos & Company	Rua das sanita, nº 2	127.0.0.1	3306	4	16
10002	Máscaras de Carnaval LDA	Rua de Elm street, nº 13	127.0.0.1	3306	6	30
10003	Aperture Science	Travessa de bolo, nº 404	127.0.0.1	3306	2	2
10004	Black Mesa	Área 51	127.0.0.1	3306	1	10

Figura 4.8: Área de Gestão de Clientes sem conexão local

Como referido anteriormente a aplicação divide-se numa utilização geral e local, todas as funcionalidades descritas até agora têm em vista uma utilização geral. As restantes funcionalidades que são descritas até ao final do capítulo visam um uso local.

Após uma conexão bem sucedida ao sistema local do cliente são desbloqueadas novas áreas de gestão como mostra a Figura 4.7b. As áreas de Gestão de Moldes e Gestão de Sensores demonstradas nas Figuras 4.9 e 4.10, permitem ao utilizador criar e apagar moldes e sensores, respetivamente. Estes dados são inseridos na base de dados temporária local, aqui o utilizador pode criar e apagar moldes e sensores sem afetar o sistema. Desta forma é possível confirmar a informação introduzida antes de a inserir no sistema. Os botões de Criar e Apagar nestes formulários realizam *queries* do tipo INSERT e DELETE, respetivamente.

**Aplicação**

Home Consultas Administração cl\_10001 Logout

Gestão Clientes Gestão Moldes Gestão Sensores

**Temporária:**

ID Cliente	ID Molde	Nome	Descrição	Nº de Sensores
10001	9000	NULL	NULL	5

Validar

Molde:

ID Cliente:

ID Molde:

Nome:

Descrição:

Adicionar Molde Eliminar Molde

Ver Clientes Ver Moldes Ver Sensores

Figura 4.9: Área de Gestão de Moldes

**Aplicação**

Home Consultas Administração cl\_10001 Logout

Gestão Clientes Gestão Moldes Gestão Sensores

**Temporária:**

ID Cliente	ID Molde	Número do Sensor	Tipo de Sensor	Nome	Descrição
------------	----------	------------------	----------------	------	-----------

Validar

Sensor:

ID Molde:

Número do Sensor:

Tipo:

Nome:

Descrição:

Adicionar Sensor Eliminar Sensor

Ver Clientes Ver Moldes Ver Sensores

Figura 4.10: Área de Gestão de Sensores

Quando a informação dos moldes e sensores estiver completa o botão Validar tenta registrar os valores presentes na base de dados temporária local nas bases de dados central e local. Se a ação não executar com sucesso é retornado um erro *MySQL* de forma a informar o utilizador. Se a ação executar com sucesso a base de dados temporária local é limpa e os valores são registados permanentemente nas bases de dados central e local, como representado nas Figuras 4.11a e 4.11b.

Depois de inseridos, moldes e sensores, não podem ser eliminados via aplicação. Esta opção foi removida da aplicação para evitar erros, dado que apagar um molde em funcionamento faz com que se percam novos registos.

The screenshot shows the 'Aplicação' interface with a navigation bar. Below the navigation bar, there is a 'Temporária:' section containing a table with the following data:

ID Cliente	ID Molde	Número do Sensor	Tipo de Sensor	Nome	Descrição
10000	5000	1	Termómetro	NULL	Temperatura ponto crítico 1
10000	5000	2	Pressão	NULL	Pressão ponto crítico 2

Below the table, there is a 'Validar' button and a form for adding a new sensor with fields for ID Molde, Número do Sensor, Tipo, Nome, and Descrição.

(a) Dados antes de serem validados

The screenshot shows the 'Aplicação' interface after validation. The 'Temporária:' section now contains a table with the following data:

ID Cliente	ID Molde	Nome	Descrição	Nº de Sensores
10000	5000	NULL	Termora A	2

The 'Validar' button is still present, and the form for adding a new sensor is also visible.

(b) Dados após serem validados

Figura 4.11: Função do botão Validar, onde os valores da base de dados temporária local são transferidos de forma permanente para as bases de dados central e local

Voltando a área de Gestão de Clientes, após a conexão local, desbloqueia-se uma nova funcionalidade como demonstra a Figura 4.12. O botão Atualizar permite reiniciar o programa de transferência de valores para que este atualize o número de clientes. Com o comando:

```
ps ax | grep transferencia
```

Obtém-se os números de processo dos programas que estão a transferir valores. Estes valores são armazenados na variável @pids. Para terminar os programas utiliza-se o seguinte comando:

```
kill -2 @pids
```

A opção -2 permite enviar para o processo escolhido o sinal SIGINT que é o sinal esperado pelo programa para que este termine as suas rotinas antes de encerrar. Para iniciar novamente o comando usar:

```
~/path/transferencia
```

É necessário garantir permissões ao servidor *Apache* no sistema central para que este possa executar estes comandos.

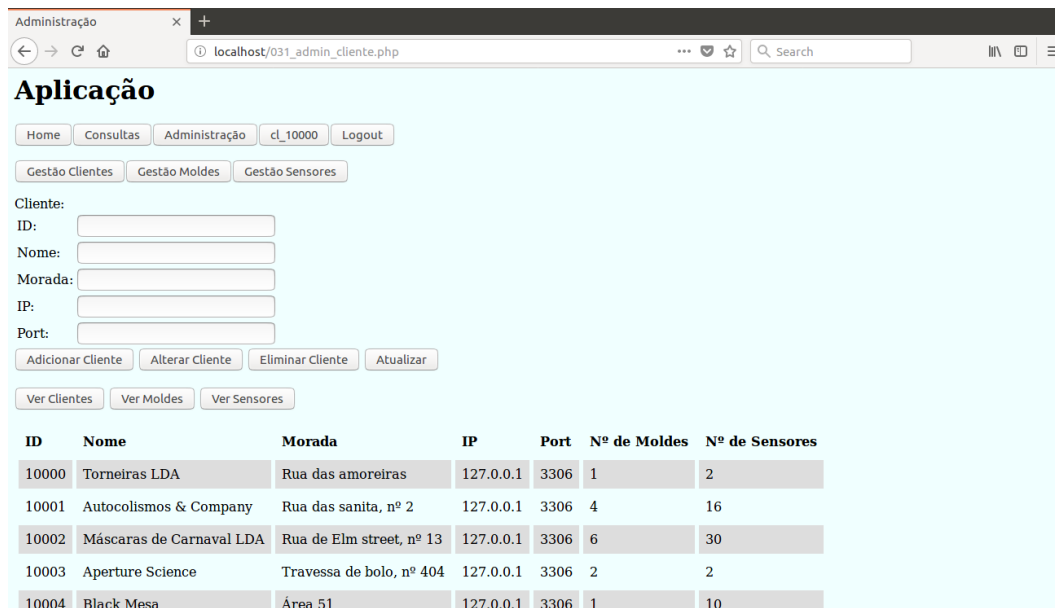


Figura 4.12: Área de Gestão de Clientes após conexão local onde se visualiza o botão Atualizar

Nas várias áreas de gestão existem os botões Ver Clientes, Ver Moldes e Ver Sensores que executam respetivamente as *queries*:

```
SELECT cl_ID , cl_nome , cl_morada , cl_IP , cl_port ,
COUNT(DISTINCT m_ID) , COUNT(DISTINCT s_IDMolde , s_num)
FROM clientes
LEFT OUTER JOIN moldes ON cl_ID = m_IDCliente
LEFT OUTER JOIN sensores ON m_ID = s_IDMolde
GROUP BY cl_ID
ORDER BY cl_ID
```

```
SELECT m_IDCliente , m_ID, m_nome, m_descricao ,
COUNT(DISTINCT s_IDMolde , s_num)
FROM clientes
INNER JOIN moldes ON cl_ID = m_IDCliente
LEFT OUTER JOIN sensores ON m_ID = s_IDMolde
GROUP BY m_ID
ORDER BY m_IDCliente , m_ID
```

```
SELECT m_IDCliente , s_IDMolde , s_num , tipo_nome ,
s_nome , s_descricao
FROM moldes
INNER JOIN sensores ON m_ID = s_IDMolde
INNER JOIN tipo ON s_tipo = tipo_id
ORDER BY m_IDCliente , s_IDMolde , s_num
```

Estas fornecem algumas informações contextuais para facilitar a navegação do utilizador.

### 4.2.5 Conexão local

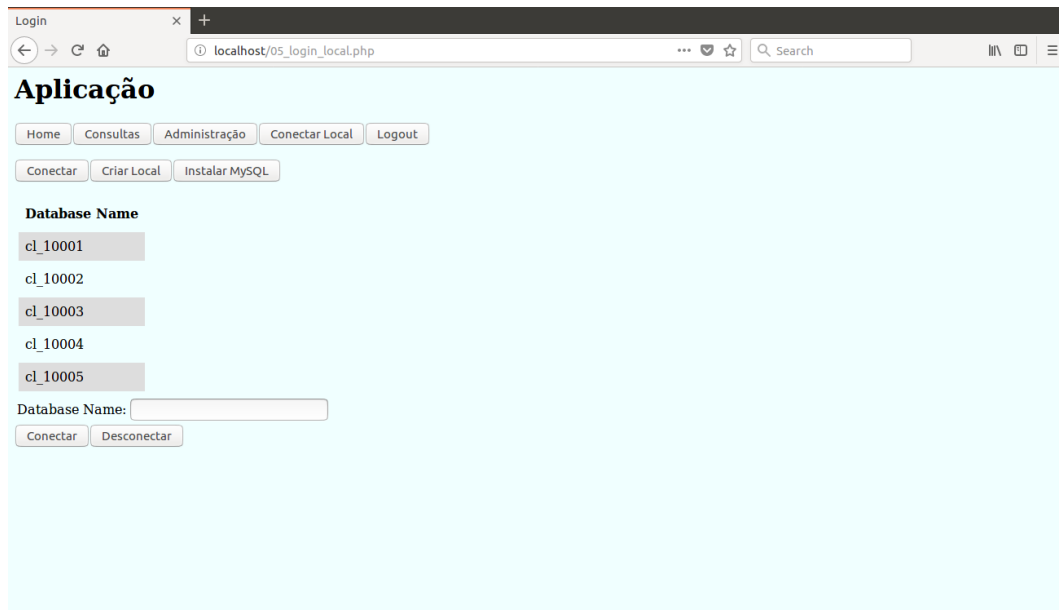


Figura 4.13: Área Conectar Local onde se visualiza as bases de dados instaladas no sistema

A área de Conectar Local na Figura 4.13 permite realizar uma conexão à base de dados local no servidor do cliente. Com recurso à *query*:

**SHOW DATABASES**

Obtém-se todas as bases de dados instaladas no servidor local. Do ponto de vista prático, cada cliente só terá uma base de dados local mas, para efeitos de desenvolvimento do projeto adotou-se esta vertente. O botão Conectar inicia sessão na base de dados local escolhida e redireciona o utilizador para a página principal como se observar na Figura 4.14. O botão Desconectar termina esta sessão e redireciona o utilizador também, para a página principal.

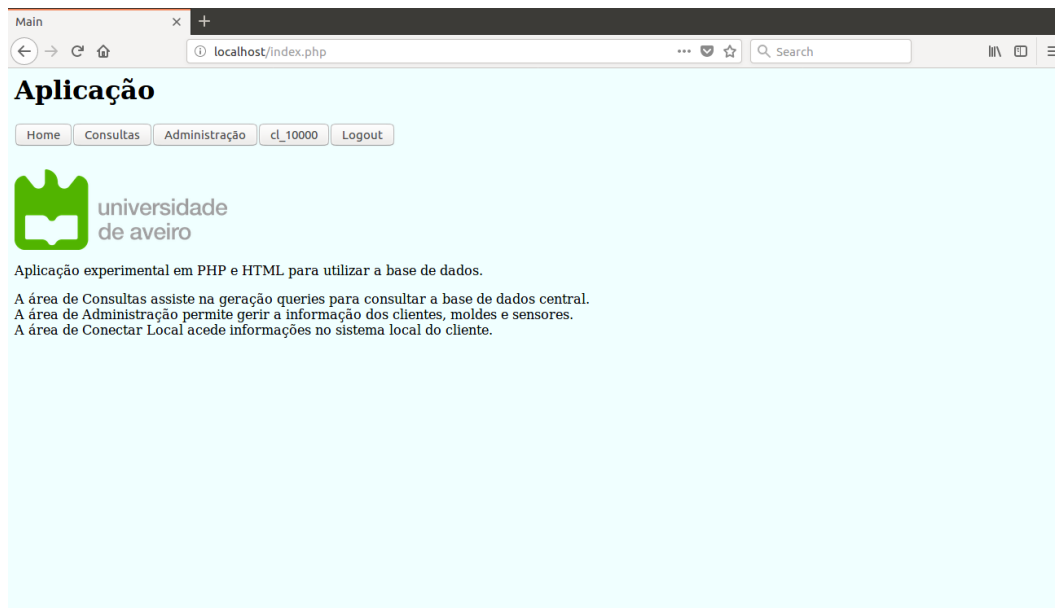


Figura 4.14: *Main* com conexão local

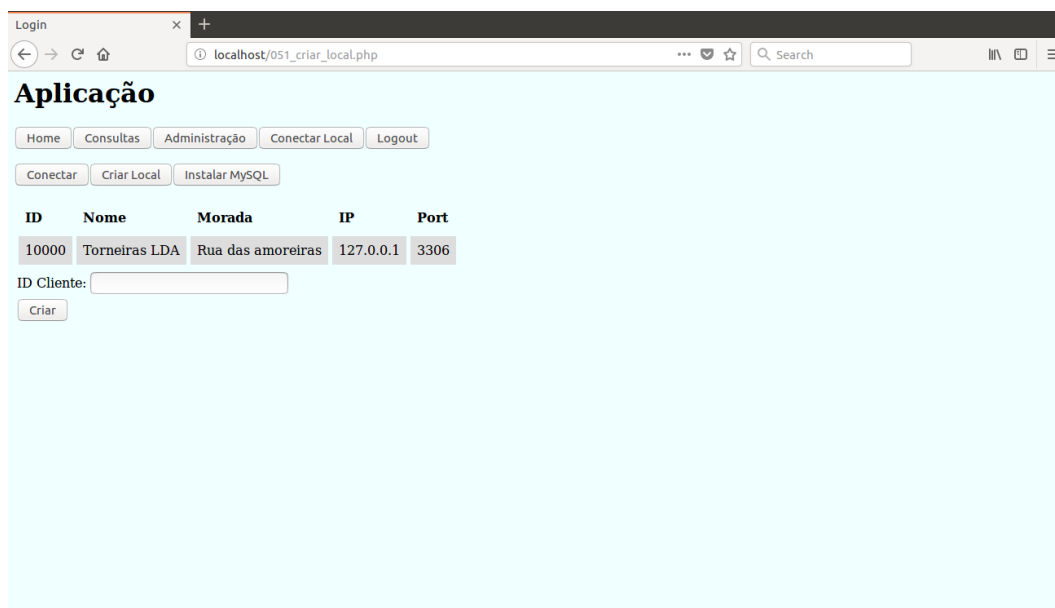


Figura 4.15: Área Criar Local cria a base de dados para o cliente seleccionado no sistema em que se acede a aplicação

A área Criar Local na Figura 4.15 permite instalar uma base de dados para um novo cliente. São considerados novos clientes todos os que não tenham moldes associados a si, esta informação obtém-se com a *query*:

```

SELECT cl_ID , cl_nome , cl_morada , cl_IP , cl_port
FROM
    (SELECT cl_ID , cl_nome , cl_morada , cl_IP , cl_port ,
    COUNT(DISTINCT m_ID) AS n_moldes
    FROM clientes
    LEFT OUTER JOIN moldes ON cl_ID = m_IDCliente
    GROUP BY cl_ID) AS contagem
WHERE n_moldes = 0

```

Escolhendo um cliente válido o botão Criar cria a base de dados local com as respetivas tabelas e gera ainda as *queries* observadas na Figura 4.16.

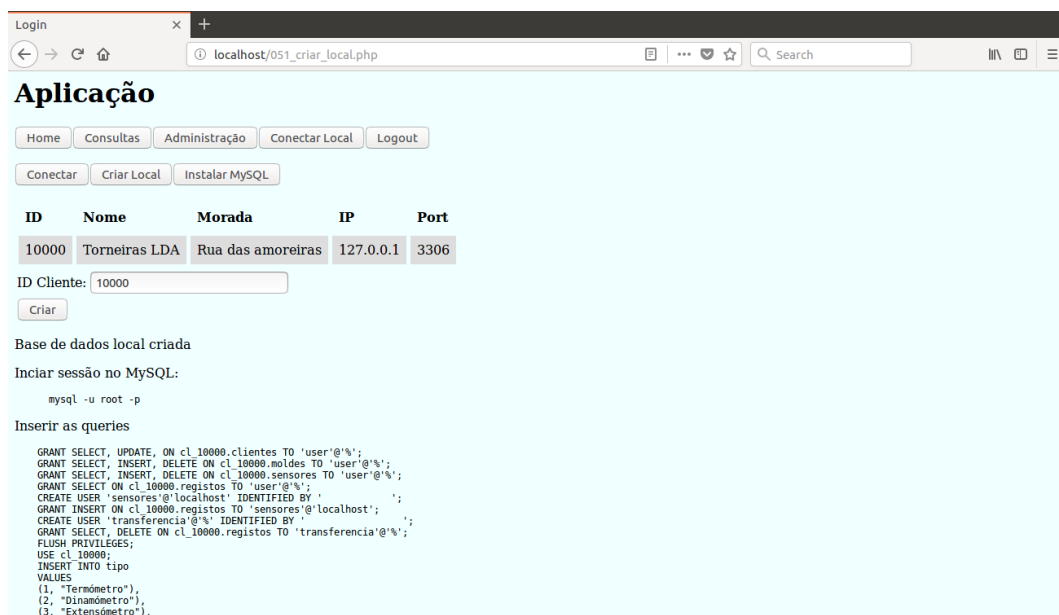


Figura 4.16: *Queries* geradas para completar a instalação da base de dados no sistema local. Consistem nas permissões para os utilizadores que só podem ser garantidas via *root* bem como informação para completar o sistema de dados

Terminado a análise das funcionalidades da aplicação com a área de Instalar *MySQL* na Figura 4.17 que contém os passos para instalar o *MySQL* num sistema *Linux*.

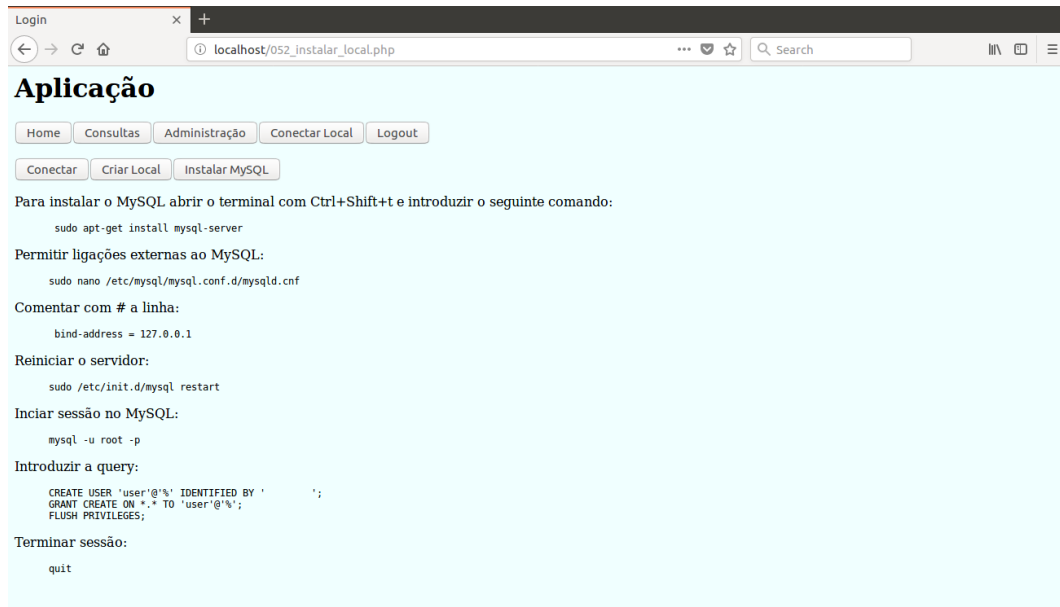


Figura 4.17: Área Instalar *MySQL* para instalar o *software* com os comandos em *Linux*



## Capítulo 5

# Instalação do Sistema



## Capítulo 6

# Conclusões

O desafio proposto consiste em monitorizar sensores remotamente. Para isto definiu-se como objetivos principais desenvolver uma rede de bases de dados e uma aplicação que permitisse interagir com estas. Estes objetivos foram concluídos com sucesso. Este capítulo contém comentários sobre o desempenho da solução desenvolvida e propostas para trabalhos futuros.

### 6.1 Comentários

#### 6.1.1 Infraestrutura de dados

Em relação à infraestrutura proposta, esta cumpre todos os requisitos propostos de garantir uma transferência segura, confidencial e permanente de valores, criando um histórico dos moldes monitorizados. Os programas desenvolvidos em *C* são simples e funcionais. No entanto, quando estão em funcionamento, o sistema executa-os sem restrições. Isto resulta num elevado consumo do processador e consequente perda de performance do sistema central. Esta perda de performance pode ser por causa do *hardware* utilizado no projeto e a utilização de um sistema devidamente dimensionado para a tarefa em mão pode resolver este problema. Outra solução seria limitar a velocidade de processamento da execução destes programas. Como definido nos objetivos esta não impõe restrições na instrumentação dos moldes. Para introduzir dados nas bases de dados locais pode ser utilizado qualquer sistema operativo e linguagem de programação desde que esta tenha protocolos de comunicação com *MySQL* e gere *queries* do tipo:

```
INSERT INTO registos
VALUES
(molde , sensor , fase , data_hora , milissegundos , valor );
```

Na realidade esta infraestrutura pode ser utilizada em qualquer contexto de monitorização remota de sensores desde que seja desenvolvido um modelo de dados apropriado.

#### 6.1.2 Aplicação

Em relação à aplicação, esta cumpre os objetivos propostos de ser multiplataforma e garantir um acesso remoto à base de dados central, bem como gerir as informações dos clientes, moldes e sensores. No entanto, o facto da aplicação ter sido desenvolvida na totalidade com *PHP* e *HTML*, causa uma perda de performance no servidor central. Isto aconteceu porque,

durante o desenvolvimento do projeto, não foi percebido na totalidade o conceito de lado do servidor e lado do cliente. De forma a melhorar o desempenho sugere-se a alteração de algumas funcionalidades desenvolvidas em *PHP* para *JavaScript*, como por exemplo, as conexões às bases de dados que são particularmente pesadas no servidor. A alteração das conexões de *PHP* para *JavaScript* permite também alterar o método de instalação da base de dados local mencionado na Subseção 4.2.5. Em vez de serem gerados comandos para o utilizador executar no *MySQL* estes podem ser enviados diretamente pela aplicação desde que seja fornecida a *password* para a *root* do sistema como sugerido na Figura 6.1.

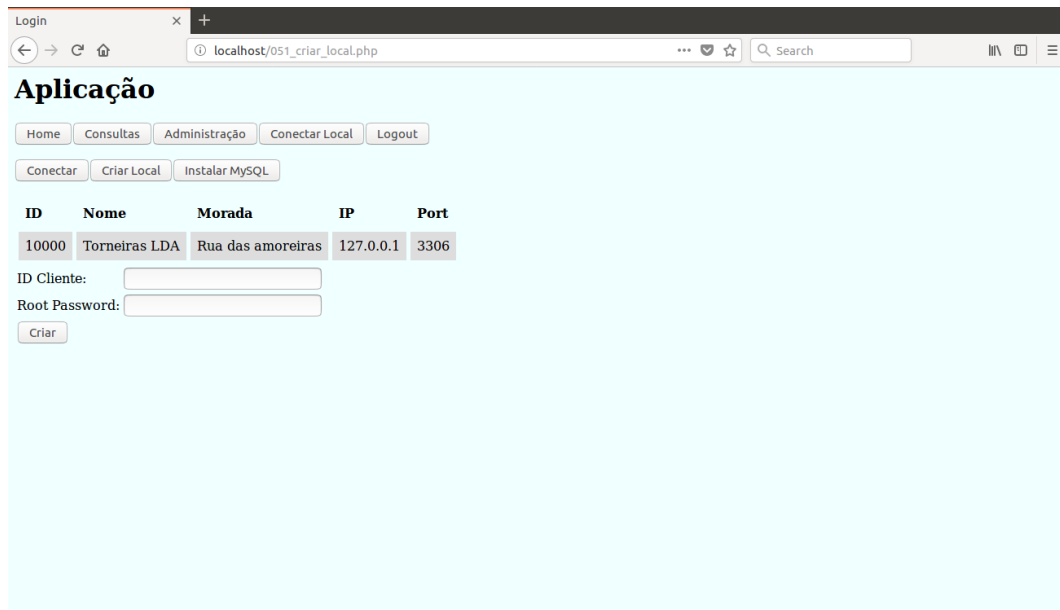


Figura 6.1: Área de Criar Local com criação via *root* em vez de gerar *queries* para o utilizador inserir manualmente no *MySQL*

Além disto é necessário realizar uma revisão de segurança à aplicação. Por exemplo, o botão Atualizar na área Gestão de Clientes que reinicia o programa de transferência de valores através de comandos no terminal. Se as permissões garantidas ao servidor *Apache* não forem bem definidas, pode constituir uma quebra de segurança. Um programador com intenções maliciosas pode acessar o sistema pela aplicação e realizar comandos no terminal de forma a comprometer o sistema. Para isto não acontecer é necessário garantir que o servidor *Apache* só tem permissões sobre o programa de transferência ou então definir um sistema de notificações entre a aplicação e o programa em *C*.

Apesar de serem necessários alguns ajustes de forma a melhorar performance, a solução proposta da infraestrutura e aplicação é completamente funcional e pode ser já implementada numa fase experimental.

## 6.2 Trabalhos Futuros

Quanto à infraestrutura dos dados não foi definido como os sensores dos moldes serão ligados ao servidor local. No desenvolvimento deste projeto assumiu-se que todos os moldes estão

ligados diretamente ao sistema local. Se esta proposta não se demonstrar viável é possível, em vez de se ter um servidor local por cliente, ter um servidor local por molde. Esta adaptação irá criar uma maior quantidade de bases de dados locais mas, isto não é problemático, se o modelo de dados e o programa de transferência forem adaptados para o efeito.

Quanto à aplicação, sugere-se que após uma apresentação inicial desta à empresa promotora, seja iniciado um processo iterativo de desenvolvimento para escolher e desenvolver novas funcionalidades que possam ser úteis e que não tenham sido abrangidas neste projeto, como por exemplo, a criação de utilizadores baseado no ID de trabalhador demonstrado na Figura 6.2. Culminando numa estilização da aplicação para que esta tenha um aspeto mais amigável ao utilizador.

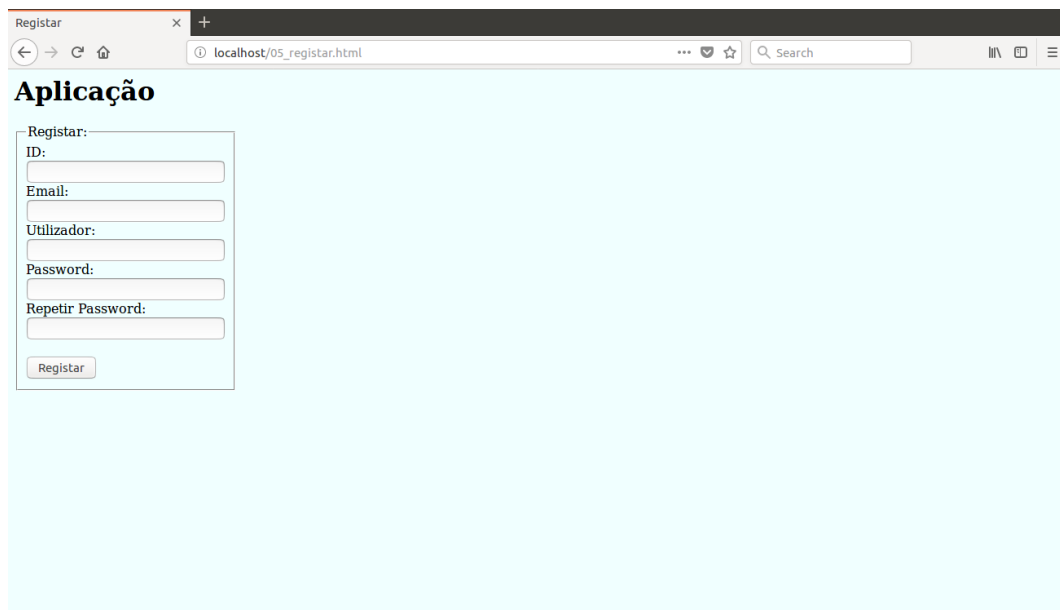
The image shows a web browser window with the title 'Registrar'. The address bar shows 'localhost/05\_registar.html'. The main content area has a light blue background and is titled 'Aplicação'. Inside this area, there is a white box containing a registration form. The form has the following fields: 'ID:', 'Email:', 'Utilizador:', 'Password:', and 'Repetir Password:'. Each field is followed by a text input box. At the bottom of the form is a button labeled 'Registrar'.

Figura 6.2: Exemplo de área de Registrar para criar utilizadores com base no ID de trabalhador e email

Além destas alterações sugere-se a criação de um sistema de notificações e monitorização automático dos moldes. É impossível para um utilizador analisar manualmente milhões de registos de um molde e concluir se este está a funcionar corretamente. Para este efeito criar um programa capaz de correr algoritmos que analisem o comportamento dos moldes. Este programa pode ser desenvolvido em *softwares* mais sofisticados, como por exemplo *MATLAB*, desde que estes tenham protocolos de comunicação com *MySQL*.

A gestão dos *backups* descrita na Subseção 3.2.5 onde se separa os registos dos moldes em vez de se realizar um *backup* geral foi realizada com este sistema de notificações em mente. Se for necessário, para efeitos de cálculo, que o programa carregue os registos de um molde armazenados em *backups*, este só necessita de carregar a informação do molde que está a ser analisado em vez de ter de carregar a informação de todos os moldes. Este programa deverá correr automaticamente no sistema de forma permanente ou com um temporizador e, no caso de ser necessário, notificar o utilizador via aplicação ou via email.



# Bibliografia

- [1] colored-part-large-mold-with-part.
- [2] moldagem in Dicionário infopédia da Língua Portuguesa [em linha]. Porto: Porto Editora, 2003-2018. [consult. 2018-05-22 18:20:35]. Disponível na Internet: <https://www.infopedia.pt/dicionarios/lingua-portuguesa/moldagem>.
- [3] moldar in Dicionário infopédia da Língua Portuguesa [em linha]. Porto: Porto Editora, 2003-2018. [consult. 2018-05-22 18:21:22]. Disponível na Internet: <https://www.infopedia.pt/dicionarios/lingua-portuguesa/moldar>.
- [4] molde in Dicionário infopédia da Língua Portuguesa [em linha]. Porto: Porto Editora, 2003-2018. [consult. 2018-05-22 16:51:33]. Disponível na Internet: <https://www.infopedia.pt/dicionarios/lingua-portuguesa/molde>.
- [5] G. Williams and H. A. Lord. Mold-filling studies for the injection molding of thermoplastic materials. Part I: The flow of plastic materials in hot- and cold-walled circular channels. *Polymer Engineering and Science*, 15(8):553–568, aug 1975.
- [6] Michael Trovant and Stavros A Argyropoulos. The implementation of a mathematical model to characterize mold metal interface effects in metal casting. *Canadian Metallurgical Quarterly*, 37(3-4):185–196, jun 1998.
- [7] TN) Janney, Mark A. (Knoxville and NG) Omatete, Ogbemi O. (Lagos. Method for molding ceramic powders using a water-based gel casting, jul 1991.
- [8] Jiwang Yan, Takashi Oowada, Tianfeng Zhou, and Tsunemoto Kuriyagawa. Precision machining of microstructures on electroless-plated NiP surface for molding glass components. *Journal of Materials Processing Technology*, 209(10):4802–4808, jun 2009.
- [9] Changyu Shen, Lixia Wang, and Qian Li. Optimization of injection molding process parameters using combination of artificial neural network and genetic algorithm method. *Journal of Materials Processing Technology*, 183(2-3):412–418, mar 2007.
- [10] K. Shelesh-Nezhad and E. Siores. An intelligent system for plastic injection molding process design. *Journal of Materials Processing Technology*, 63(1-3):458–462, jan 1997.
- [11] Babur Ozcelik and Ibrahim Sonat. Warpage and structural analysis of thin shell plastic in the plastic injection molding. *Materials & Design*, 30(2):367–375, feb 2009.
- [12] About Injection Molding | Xcentric Mold & Engineering.

- [13] *Polymer Process Engineering - R. Griskey - Google Livros.*
- [14] hyatt. nov 1872.
- [15] Injection molding machine. may 1949.
- [16] Suzanne L. B. Woll and Douglas J. Cooper. Pattern-based closed-loop quality control for the injection molding process. *Polymer Engineering & Science*, 37(5):801–812, may 1997.