



**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

Potato

ergergergerg

bergergerg

Abstract

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

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Lista de Tabelas	v
1 Introdução	1
2 Estado de Arte	3
3 Proposta de Solução	5
3.1 Infraestrutura de dados	5
3.2 Base de Dados	5
3.2.1 Análise de Requisitos	5
3.2.2 Desenho conceptual e esquema lógico	5
3.2.3 Construção da base de dados	5
3.2.4 Programa de transferência	5
3.2.5 Gestão de <i>backups</i>	5
3.2.6 Simulador	5
3.2.7 Utilizadores	5
4 Aplicação	7
4.1 Adaptação da infraestrutura	7
4.2 Interface gráfica	8
4.2.1 <i>Main</i>	8
4.2.2 <i>Login</i>	9
4.2.3 Consultas	10
4.2.4 Administração	12
4.2.5 Conexão local	15
5 Instalação do Sistema	17
6 Conclusões	19
6.1 Comentários	19
6.1.1 Infraestrutura de dados	19
6.1.2 Aplicação	19
6.2 Trabalhos Futuros	20

Lista de Figuras

4.1	Esquema ligação temporária	7
4.2	Main	8
4.3	Esquema ligação temporária	9
4.4	Main	9
4.5	Main	10
4.6	Main	11
4.7	Main	12
4.8	Esquema ligação temporária	12
4.9	Main	13
4.10	Main	15
4.11	Esquema ligação temporária	16
4.12	Esquema ligação temporária	16
6.1	Esquema ligação temporária	20
6.2	Esquema ligação temporária	21

Lista de Tabelas

Capítulo 1

Introdução

arquivo e monitorização de moldes.

Capítulo 2

Estado de Arte

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.

CAPAZ DE CONSULTAR E INTRODUIZIR VALORES NAS BASES DE DADOS

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 temporária

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*

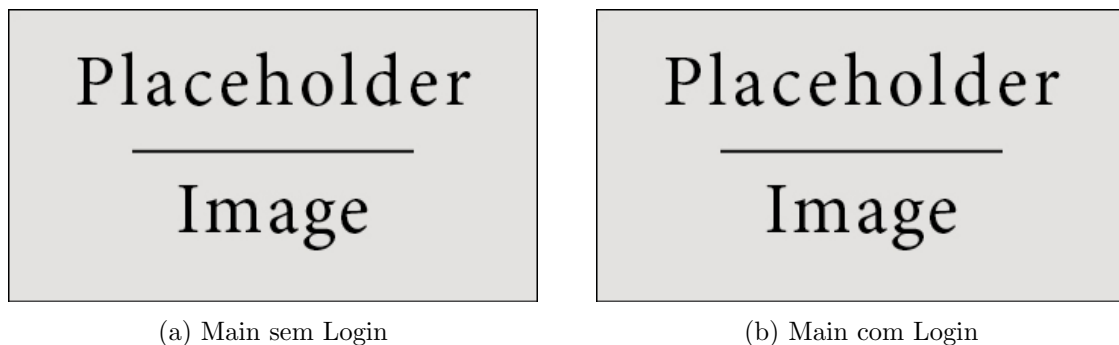


Figura 4.2: Main

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.

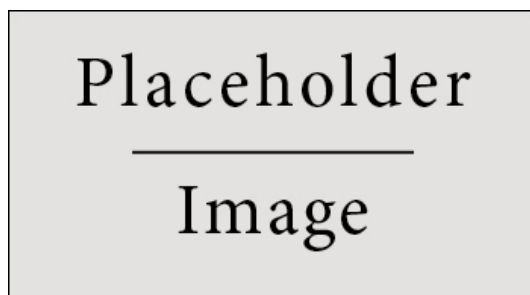
4.2.2 *Login*



Figura 4.3: Esquema ligação temporária

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) Main sem Login



(b) Main com Login

Figura 4.4: Main

4.2.3 Consultas

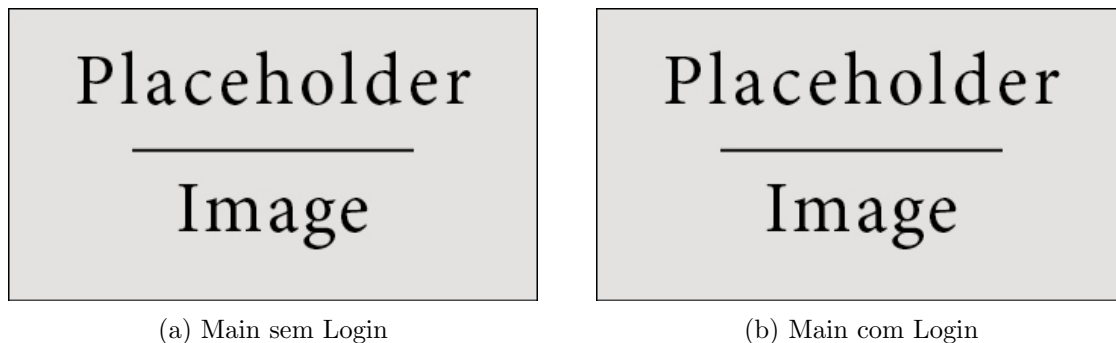


Figura 4.5: Main

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 registos 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 seleccionados sozinhos. As primeiras duas permitem consultar as opções disponíveis nos dicionários e a terceira devolve as datas e horas entre o primeiro e último registos. 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.

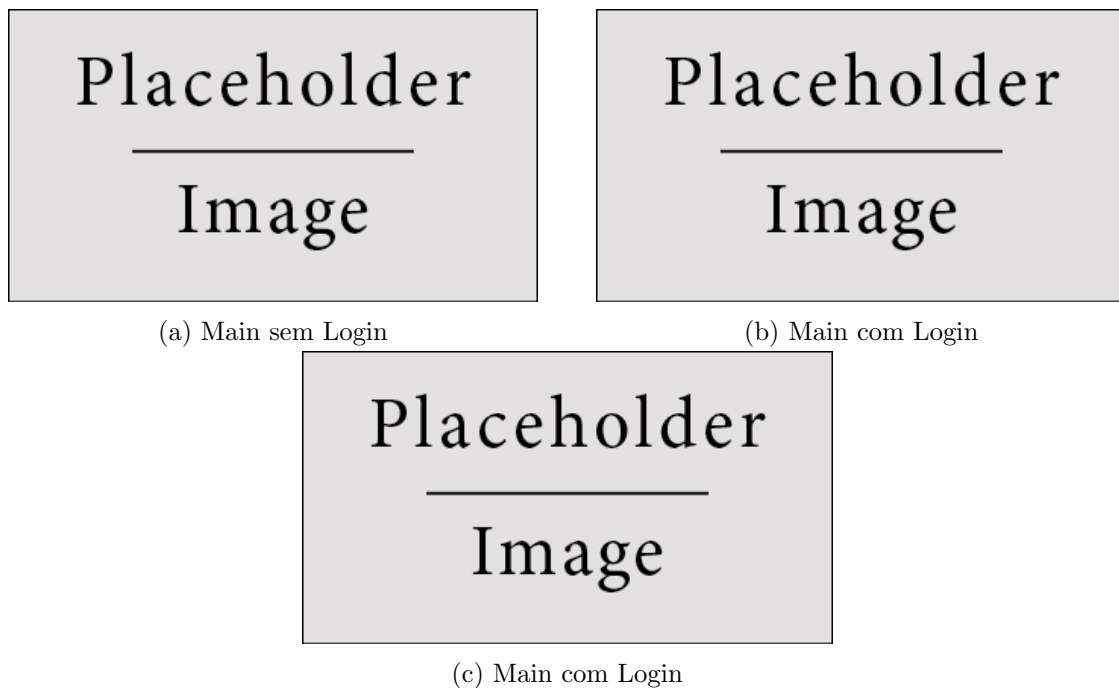


Figura 4.6: Main

4.2.4 Administração

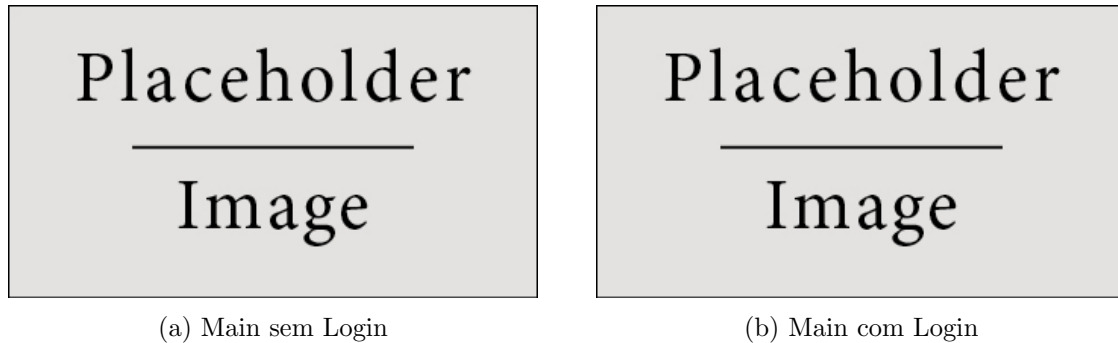


Figura 4.7: Main

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.



Figura 4.8: Esquema ligação temporária

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 servidor 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 4.9b e 4.9c, 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.

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.

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.

Voltando a área de Gestão Clientes, após a conexão local, desbloqueia-se uma nova funcionalidade como demonstra a Figura 4.9a. O botão Atualizar permite reiniciar o programa de transferência de valores para que este atualize o número de clientes. Com o comando:

`grep`

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.

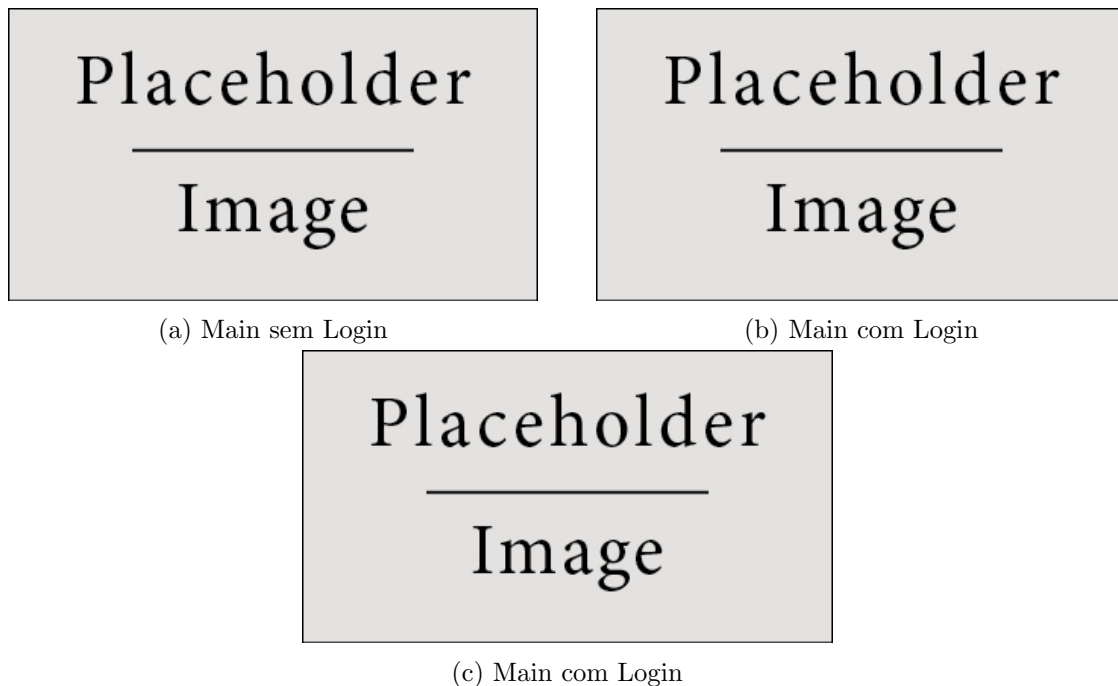


Figura 4.9: Main

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

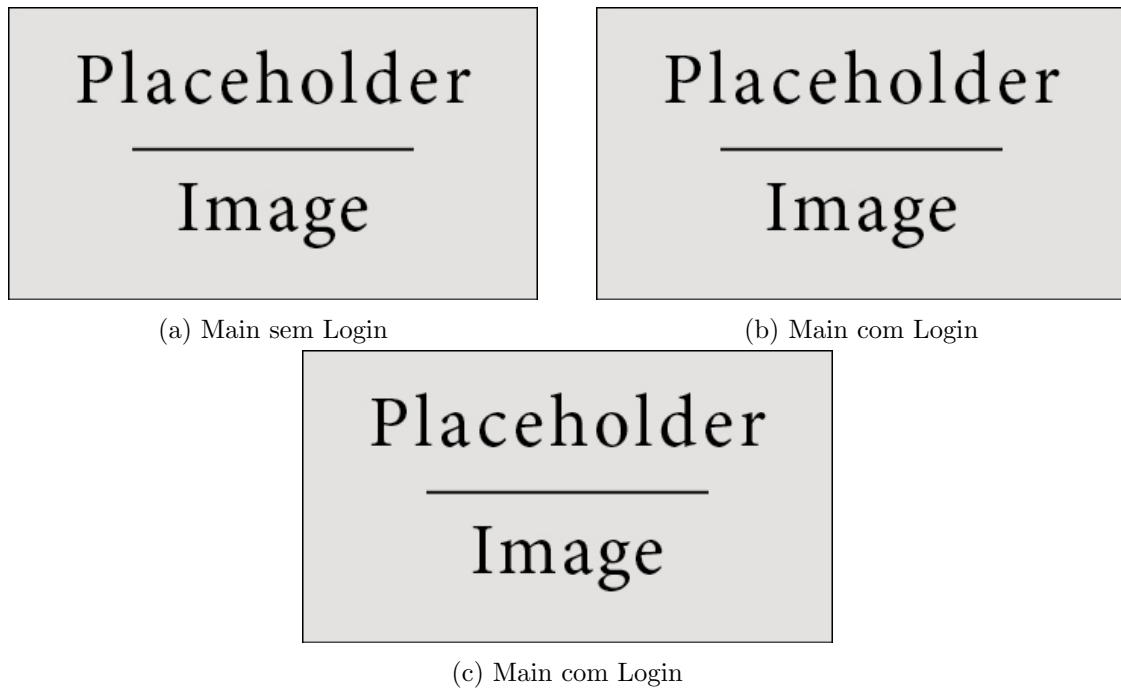


Figura 4.10: Main

A área de Conectar Local na Figura 4.10a 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 4.11. O botão Desconectar termina esta sessão e redireciona o utilizador também, para a página principal.



Figura 4.11: Esquema ligação temporária

A área Criar Local na Figura 4.10b 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 4.12.



Figura 4.12: Esquema ligação temporária

Terminado a análise das funcionalidades da aplicação com a área de Instalar *MySQL* na Figura 4.10c que contém os passos para instalar o *MySQL* num sistema *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);
```

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: Esquema ligação temporária

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.



Figura 6.2: Esquema ligação temporária

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] Shalom Eliahou. The $3x + 1$ problem: New lower bounds on nontrivial cycle lengths. *Discrete Mathematics*, 118(1–3):45–56, 1993.
- [2] Lynn E. Garner. On the collatz $3n + 1$ algorithm. *Proceedings of the American Mathematical Society*, 82(1):19–22, May 1981.

