



Instituto Superior de Economia e Gestão

UNIVERSIDADE TÉCNICA DE LISBOA

DESDE 1911

# **MESTRADO**

## **GESTÃO DE SISTEMAS DE INFORMAÇÃO**

### **TRABALHO FINAL DE MESTRADO**

#### **TRABALHO DE PROJETO**

**DESENHO E IMPLEMENTAÇÃO DE UMA APLICAÇÃO WEB DE GESTÃO  
DE OPORTUNIDADES**

**SECORÁULIO PÔNCIO GOMES**

**SETEMBRO 2014**

**MESTRADO**  
**GESTÃO DE SISTEMAS DE INFORMAÇÃO**

**TRABALHO FINAL DE MESTRADO**  
**TRABALHO DE PROJETO**

**DESENHO E IMPLEMENTAÇÃO DE UMA APLICAÇÃO WEB DE GESTÃO  
DE OPORTUNIDADES**

**SECORÁULIO PÔNCIO GOMES**

**ORIENTAÇÃO:**

**PROF. DOUTOR ANTÓNIO MARIA PALMA DOS REIS**  
**TELMO MARAVILHA**

**SETEMBRO 2014**

## RESUMO

O desenvolvimento das tecnologias de informação tem influenciado muito a forma como as empresas gerem e tratam volumes de informação, elemento chave no crescimento do negócio. A forma de gerir e tratar esses dados tem um impacto direto nos benefícios e nas vantagens que este conjunto de informação pode oferecer. Por isso, as empresas procuram sempre soluções informáticas que possam auxiliar e proporcionar uma gestão eficiente deste volume de dados.

Infelizmente muitas das soluções encontradas no mercado para além de terem um custo elevado, têm característica de ser genéricas.

A solução desenvolvida neste trabalho tenta responder de forma simples às necessidades da empresa S24Group.

O projeto iHR consiste na análise e implementação de uma aplicação web de gestão de oportunidades.

O iHR foi desenvolvido em duas fases:

1. Fase de análise

Apontar as necessidades da equipa de gestão de oportunidades, delimitar o âmbito aplicacional e definir as operações que o sistema deverá efetuar são os pontos fortes desta fase do projeto. Foi necessário recorrer as ferramentas UML: para ter uma visão ampla dos processos e dos atores utilizamos o diagrama de caso. Para perceber o encadeamento desses processos utilizamos os diagramas de atividade e para ter uma visão estática do sistema e dos objetos que formam o seu ecossistema utilizamos os diagramas de classes. Por fim recorreremos ao modelo relacional para representar toda essa informação de forma clara e coerente.

## 2. Fase de implementação

A principal preocupação da equipa do projeto foi a utilização das mais recentes e boas práticas no desenvolvimento de aplicações web. De forma a alcançar a maior eficiência, performance e agilidade foram utilizados os melhores modelos e ferramentas.

A arquitetura 3-tiers junto ao modelo MVC, em HTML, PHP, JQuery e JavaScript suportado com uma base de dados MySQL resulta numa aplicação acessível na web, fluido com o maior desempenho.

**Palavras-chave:** Desenvolvimento de *software*, Aplicação web, Análise de requisitos do sistema, UML, PHP e CodeIgniter.

## **ABSTRACT**

The development of information technology has influenced strongly the way companies manage and treat volume of information, a key element in business growth. The way to manage and treat such data has a direct impact on benefits and advantages that this set of information can offer. So companies are always looking for IT solutions that can assist and provide efficient management of data sets.

Unfortunately, in many solutions found in the market pair besides having a high cost, have a characteristic of being generic.

The solution developed in this paper attempts to answer in a simple way the needs of S24Group company.

IHR project consists in analysis and implementation of web application for managing opportunities.

IHR was developed in two phases:

### 1. Analysis

Pointing the needs of the opportunities management team, defining the application context and defining operations that the future system should perform, are the strong points of this phase of the project. It was necessary to use UML tools: to take a broad view of processes and actors use case diagram were used, to realize the linkage of these processes we use activity diagrams, to have a static view of the system and objects that form its ecosystem we use class diagrams and finally resorted to the relational model to represent all this information clearly and coherently.

### 2. Implementation

Using the latest and best practices in web application development, has been the concern of the project team. The best model, tools that deliver performance and agility have been used to achieve greater efficiency.

The 3-tiers architecture with the MVC model, HTML, PHP, JavaScript and JQuery supported with a MySQL data base results in an accessible web application, fluid with higher performance.

**Keywords:** Software development, Analysis of system requirements, Web application, UML, PHP and CodeIgniter.

## ÍNDICE DAS FIGURAS

Figura 1-Diagrama de caso de utilização

Figura 2-Diagrama de atividade «Login»

Figura 3-Diagrama de atividade «Novo candidato»

Figura 4-Diagrama de atividade «Alterar dados do candidato»

Figura 5-Diagrama de atividade «Criar atividade ao candidato»

Figura 6-Diagrama de atividade «Inserir novo cliente»

Figura 7-Diagrama de atividade «Criar novo utilizador»

Figura 8-Diagrama de classes

Figura 9-Módulos da aplicação

Figura 10-Arquitetura MVC

Figura 11-Arquitetura física

Figura 12-Diagrama de Gantt

Figura 13 - Teste «Login»

Figura 14 - Teste «Dashboard»

Figura 15 - Teste «Lista dos candidatos»

Figura 16 – Teste «Novo candidato»

Figura 17 – Teste «Novo candidato com erros»

Figura 18 – Teste «Ficha do candidato»

Figura 19 – Teste «Tarefas & Atividades ao candidato»

Figura 20 – Teste «Anexos do candidato»

Figura 21 – Teste «Lista dos clientes»

Figura 22 – Teste «Lista das campanhas do cliente»

Figura 23 – Teste «Gestão dos utilizadores do sistema»

Figura 24 - Configuração da conexão a base de dados «database.php»

Figura 25 - Configuração do Base Path «config.php»

Figura 26 - *Controller* «verify\_login.php»

Figura 27 - *Model* «Novo candidato» Candidates\_model.php

## **ÍNDICE DAS TABELAS**

Tabela 1 - Comparação de 5 *frameworks* PHP *opensource*

Tabela 2 - Lista dos atores e suas ações no sistema

## NOTAÇÃO E GLOSSÁRIO

MVC	<i>Model View Controller</i>
REQ	Requisito Funcional
RNF	Requisito Não Funcional
UML	<i>Unified Modeling Language</i>
TI	Tecnologia de Informação
iHR	Solução implementada ( <i>International Human Resource</i> )

# ÍNDICE

<b>1. INTRODUÇÃO</b>	<b>1</b>
1.1. APRESENTAÇÃO DO PROJETO	1
1.2. A EMPRESA S24 GROUP	2
1.3. ENQUADRAMENTO TEÓRICO	2
1.4. CONCLUSÃO	5
<b>2. FASE DE ANÁLISE</b>	<b>5</b>
2.1. ANÁLISE DE REQUISITOS	5
2.1.1. Requisitos funcionais	5
2.1.2. Requisitos não funcionais	6
2.2. PLANO DE TESTES	7
2.2.1. Testes unitários	7
2.2.2. Testes funcionais	7
2.2.3. Testes por cenário de utilização	7
2.3. MODELAGEM EM UML	8
2.3.1. Diagramas de caso de utilização	8
2.3.2. Diagramas de atividade	15
2.3.3. Diagrama de classes	20
2.3.4. Modelo relacional	21
2.4. CONCLUSÃO	22
<b>3. FASE DE IMPLEMENTAÇÃO</b>	<b>23</b>
3.1. ARQUITETURA DA SOLUÇÃO E TECNOLOGIAS UTILIZADAS	23
3.2. PROCESSO DE IMPLEMENTAÇÃO	25
3.2.1. Plano de desenvolvimento das tarefas	25
3.2.2. Processo de coordenação de contributos dos programadores	27
3.3. CONFIGURAÇÃO DO PC - SERVIDOR	27
3.4. FASE DE TESTE POR CENÁRIO DE UTILIZAÇÃO	27
3.5. CONCLUSÃO	28
<b>4. CONCLUSÃO GERAL</b>	<b>28</b>
4.1. OBJETIVOS REALIZADOS	28
4.2. LIMITAÇÕES E TRABALHO FUTURO	28
4.3. APRECIÇÃO FINAL	29
<b>5. BIBLIOGRAFIA</b>	<b>30</b>
<b>6. ANEXOS</b>	<b>31</b>
6.1. CAPTURA DE ECRÃ DA FASE DE TESTE POR CENÁRIO DE UTILIZAÇÃO	31
6.2. FRAGMENTOS DE CÓDIGO	37

# 1. INTRODUÇÃO

Para gerir eficientemente oportunidades e estar preparados a responder atempadamente aos pedidos do cliente, empresas hoje em dia recorrem não só as técnicas tradicionais de gestão mais também as novas tecnologias de informação e telecomunicação, nomeadamente aplicações informática. Estas aplicações proporcionam melhor gestão das oportunidades, o que resulta numa forma mais eficiente de responder as expectativas do cliente e assim fazer crescer o negócio. Existem varias soluções no mercado que tentam responder a esta problemática, mas em muitos casos são soluções genéricas que por vezes não se enquadram na realidade da empresa.

Neste ponto de vista, faz todo o sentido desenvolver internamente uma aplicação sobre medida que será desenhado para corresponder com os objetivos definidos pela gestão de topo.

Com o desenvolvimento da rede internet, aplicações deste género são baseadas na web. Tipicamente o utilizador acede aos dados alojados num servidor de base de dados através de um servidor web.

Este projeto manifestou-se da vontade da S24 Group, empresa que vamos apresentar no ponto 1.2, a gerir mais eficientemente as suas oportunidades. O projeto foi realizado trabalhando em conjunto com a equipa em cargo da gestão de oportunidades da empresa. Foram recolhidos requisitos, e foram feitos “*brainstormings*” a fim de desenhar um sistema o mais eficiente possível.

## 1.1. Apresentação do projeto

Com o crescimento da atividade da empresa, o aumento do volume de clientes e projetos associados à atividade tal como a internacionalização da empresa, nasce a necessidade de ter uma aplicação que permite a gestão eficiente destes dados.

O projeto “iHR”, termo que daqui em frente ira designar a solução applicativa que será implementada, consiste em análise de requisitos e desenvolvimento duma aplicação de gestão de oportunidades.

O “iHR” será composto pelos seguintes módulos:

- Gestão da base de dados de profissionais para os projetos, tarefa efetuada pelos consultores e managers: registo e manipulação de dados dos candidatos (profissionais), dados que serão posteriormente analisados de forma a selecionar os melhores perfis para uma determinada campanha (oportunidade);
- Gestão de clientes e oportunidades associadas, tarefa efetuada pelos managers: registo e manipulação de dados dos clientes, definição de requisitos do cliente para cada campanha, dados que irão facilitar a seleção de perfis consoante os critérios do cliente;
- Monitorização das atividades e tarefas a realizar ao longo do processo de recrutamento para as campanhas.

Gestão dos utilizadores do sistema: aqui poderão ser criados utilizadores com as respetivas palavra passe e o nível, que permite apresentar só os menus em que o utilizador tem acesso.

### *1.2. A empresa S24 Group*

O S24 Group é um grupo nacional especializado na entrega de Serviços na área da Saúde, que surgiu em 2005 e é líder na colocação de profissionais nos mais diversos serviços em território nacional.

As principais áreas de negócio do grupo são o recrutamento, a formação técnica e a gestão de unidades hospitalares, e o seu portfólio de clientes inclui hospitais, centros de saúde, o sistema prisional português, a rede nacional de santa casa da misericórdia, empresas públicas e privadas, câmaras municipais.

### *1.3. Enquadramento teórico*

Com o avanço da internet é possível hoje em dia desenvolver aplicações que corram no navegador web, o que permite um acesso total em toda a empresa através de um computador, um tablete ou um simples smartphone, isso sem instalação prévia.

Uma aplicação web é definida como “um sistema web composto de servidor web, rede informática, protocolo HTTP e de navegador web, em que a interação do utilizador (navegação e entrada de dados) afeta o estado do negócio ”(Conallen, 1999).

O desenvolvimento de aplicação web evoluiu muito nos últimos anos, vimos aparição de várias arquiteturas e modelos de desenvolvimento:

- Arquitetura **multi-tiers**, (muitas vezes referido como arquitetura n-tiers) é uma arquitetura cliente-servidor em que a apresentação, os processos e a gestão dos dados são logicamente separados (Thakare et al, 2012).
- Modelo **MVC**, permite desenvolver uma aplicação separando a camada logica (modelo) que define os dados e os métodos de acesso da camada apresentação (view) que apresenta ao utilizador informações provenientes do modelo, e dos processos (controller) que respondem as interações do utilizador interrogando o modelo. Esta separação facilita o desenvolvimento e a manutenção da aplicação (Utpatadevi et al, 2012).
- Arquitetura **SOA**, basicamente é uma coleção de serviços (El-Sharkawi et al, 2013). Estes serviços comunicam-se um aos outros para trocar informações. O conceito de SOA é baseado em 3 componentes: *service provider*, *service registry* e *service requester*.

No desenvolvimento de *software* existem 3 fatores chave para o sucesso do projeto: o tempo, os custos e a funcionalidade. O tempo é diretamente influenciado pela forma como os programadores realizam as tarefas. Utilizam-se ferramentas como *frameworks* para ganhar tempo no desenvolvimento de *software* (Hustinawati et al, 2014).

Uma *framework* ([www.techterms.com](http://www.techterms.com)) é uma plataforma de desenvolvimento de *software*, composto de bibliotecas com classes predefinidas e funções, ajuda no desenvolvimento de aplicações. Por exemplo uma *framework* pode incluir métodos e funções para acesso e extração de informações da base de dados.

Por questões de princípios a empresa S24Group privilegia a utilização de ferramentas *opensource*. A tabela 1 apresenta um estudo comparativo sobre 5 *frameworks open-source* mais utilizadas no desenvolvimento de aplicações web.

Tabela I

Comparação de 5 *frameworks* PHP *opensource*

	<b>Comunidade</b>	<b>Documentação</b>	<b>Instalação e Configuração</b>	<b>PHP</b>
<b>Yii Framework v1</b>	SIM	Excelente	Complexidade média	5.1
<b>CodeIgniter v2</b>	FORUM, Wiki, Blogs	Excelente	Complexidade baixa	5.1.6
<b>Zend Framework v1</b>	SIM	Boa	Complexidade alta	5
<b>Cake PHP</b>	SIM	Normal	Complexidade baixa	5.2.6
<b>Symfony v2</b>	SIM	Normal	Complexidade alta	5.3.3

Fonte: <http://josemmsimo.wordpress.com/2013/12/02/php-frameworks-comparison-zend-symfony-codeigniter-yii-and-cake-php/>

A equipa de desenvolvimento optou pela utilização do CodeIgniter ([www.ellislab.com](http://www.ellislab.com)) dado que dispõe duma grande e diversa comunidade e também por ser fácil de instalar e de configurar.

O CodeIgniter utiliza o método de desenvolvimento Model-View-Controller (MVC) e disponibiliza várias funções que ajudam bastante no desenvolvimento.

A primeira fase no desenvolvimento de software é a análise a seguir vem a modelagem. UML é considerado como uma linguagem standard de modelização na área de desenvolvimento de

software orientado por objetos (Lee, 2012). Ajuda na especificação, na visualização e na conceção da estrutura do software de acordo com os requisitos do projeto (Kothamasu et al, 2012).

#### *1.4. Conclusão*

O primeiro capítulo permitiu-nos ter uma visão ampla do projeto e do seu contexto. Após descrição do projeto podemos caminhar com segurança para a fase de análise que consiste em anotar as funcionalidades chaves do futuro sistema.

## **2. FASE DE ANÁLISE**

Nesta parte vamos recolher os requisitos que nos permitem delimitar o âmbito e os recursos necessários para o nosso projeto, definir o plano de testes e proceder à modelagem do sistema utilizando a linguagem UML.

### *2.1. Análise de requisitos*

Após uma reunião com a equipa do projeto definimos os seguintes requisitos funcionais e não funcionais:

#### *2.1.1. Requisitos funcionais*

[REQ-01] - O sistema deve permitir a gestão das tarefas e atividades a realizar (adicionar e alterar).

[REQ-02] - O sistema deve permitir a gestão dos candidatos (adicionar e alterar).

[REQ-03] - O sistema deve apresentar uma ficha por candidato com todos os seus dados. Nesta ficha poderemos também ver todas as tarefas e atividades associados ao candidato, assim como as campanhas em que o candidato está associado e os seus documentos.

[REQ-04] - O sistema deve oferecer a possibilidade de fazer pesquisas a lista dos candidatos por «nome», «telefone», «email» ou «ID number».

[REQ-05] - O sistema deve oferecer possibilidade de fazer filtros a lista dos candidatos por categoria, nacionalidade, estado do processo, estado da tarefa e por intervalo de data de inserção no sistema.

[REQ-06] - O sistema deve permitir o envio de «email» ao candidato.

[REQ-07] - O sistema deve permitir a gestão dos clientes (adicionar e alterar).

[REQ-08] - O sistema deve oferecer a possibilidade de fazer pesquisas a lista dos clientes por nome, telefone, email ou ID number.

[REQ-09] - O sistema deve apresentar uma ficha por cliente com todos os seus dados. Nesta ficha poderemos também gerir todas as tarefas e atividades associados ao cliente, assim como as suas campanhas e os seus documentos.

[REQ-10] - O sistema deve permitir o envio de email ao cliente.

[REQ-11] – O sistema deve permitir a gestão dos utilizadores do sistema.

[REQ-12] – O sistema só apresenta ao utilizador os menus em que tem acesso.

[REQ-13] - Antes de ter acesso ao sistema todos os utilizadores devem ser devidamente autenticados.

### *2.1.2. Requisitos não funcionais*

[RNF-01] – Interface Amigável, fácil de utilizar e mais intuitivo possível.

[RNF-02] – Utilizadores Simultâneos, o sistema deve permitir a conexão de vários utilizadores em simultâneo.

[RNF-03] – Garantir o “não repúdio”

[RNF-04] – Linguagens de Programação - o sistema deve ser desenvolvido numa linguagem de programação que permite uma fluidez na comunicação entre o sistema e a base de dados.

[RNF-05] – Utilizar ferramentas Open-Source

[RNF-06] – Servidor de dados devem ter capacidade suficiente para garantir um bom desempenho das tarefas dos consultores.

[RNF-07] – Não repudição, o sistema deve manter um registo de todas as ações realizadas pelos utilizadores.

[RNF-08] – O sistema de gestão de base de dados deve ser MySQL.

## *2.2. Plano de testes*

Tendo em conta que no final do projeto vamos ter que testar o produto final para garantir que está em conformidade com os requisitos, julgamos necessário definir um plano de teste que seguiremos ao longo do desenvolvimento do sistema.

Definimos três tipos de testes : testes unitários, teste funcionais e testes por cenário de utilização.

### *2.2.1. Testes unitários*

Os testes unitários são feitos após desenvolvimento de pequenas tarefas como por exemplo a inserção de informações do candidato na base de dados ou extração das mesmas, para verificar se efetivamente a tarefa faz exatamente o que pretende.

### *2.2.2. Testes funcionais*

Os testes funcionais são realizados sobre um conjunto de tarefas por exemplo para testar se a redirecção a pagina do login com mensagem de erro após de inserir uma palavra passe incorreta faz se normalmente.

### *2.2.3. Testes por cenário de utilização*

Testes por cenário permitem testar a aplicação final em condições reais ou quase reais. Com cenários que representam a utilização diária da aplicação, permitem:

- Verificar se a aplicação faz bem o que pretende.
- Detetar erros de conceção com fim de os corrigir.
- Corrigir erros de nomenclatura que possam ter sido esquecidos.

### 2.3. Modelagem em UML

UML em sua versão 2 disponibiliza vários diagramas que podem ser utilizados para a descrição do sistema (Gabay, 2008).

No caso do nosso projeto utilizamos os seguintes diagramas: o diagrama de caso de utilização, o diagrama de atividade e o diagrama de classes.

#### 2.3.1. Diagramas de caso de utilização

O diagrama de caso de utilização tem como objetivo representar as necessidades dos utilizadores perante o sistema. De acordo com os requisitos do sistema, identificamos 2 atores: o consultor e o manager. A tabela II apresenta os atores e as suas respetivas ações.

Tabela II

Lista dos diferentes atores e suas respetivas ações.

<b>Consultor</b>	<b>Manager</b>
Gerir candidato	Gerir candidato
Consultar informações do cliente	Gerir cliente
Login	Gerir utilizador do sistema
	Login

Depois de identificar os atores e suas ações no sistema, apresentamos na figura 1 o diagrama dos diferentes casos de utilização.

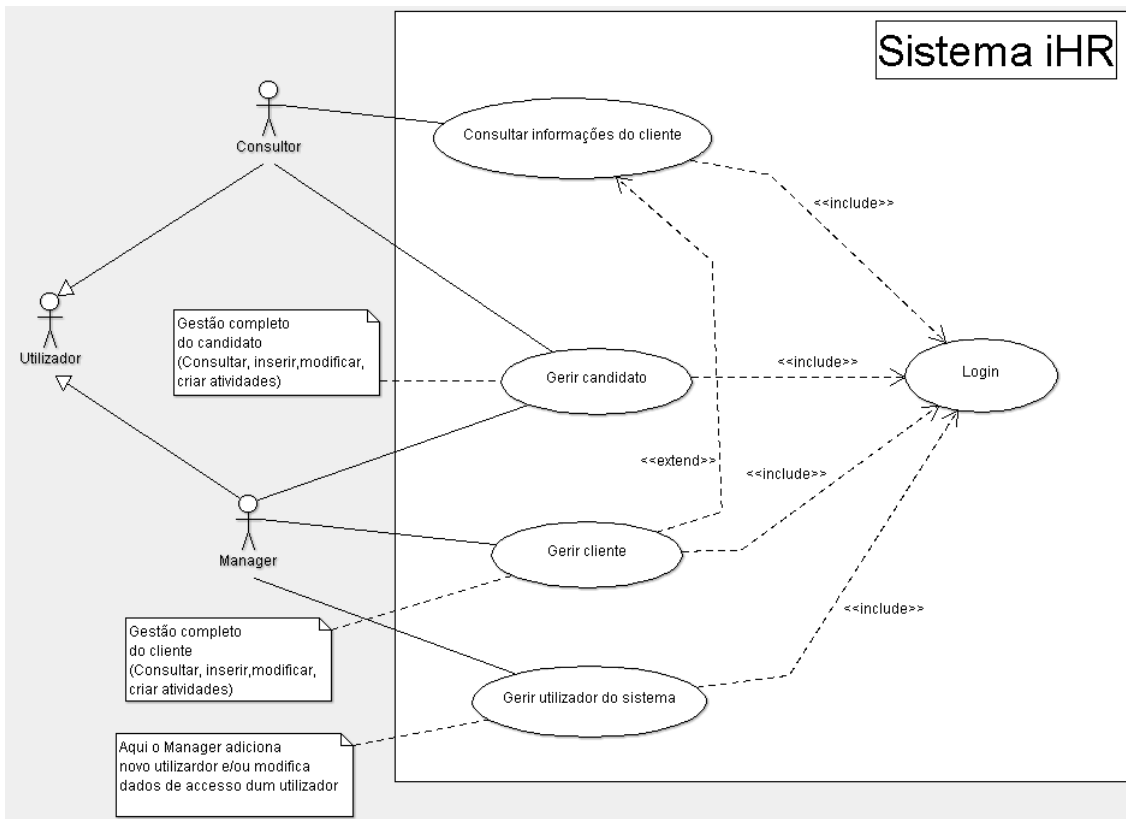


Figura 1-Diagrama de caso de utilização

A seguir apresentamos uma descrição do funcionamento dos diferentes casos de utilização:

a. Caso de utilização «Login»

- **Objetivo** – Dar acesso ao utilizador no sistema
- **Ator** – utilizador do sistema
- **Pré condição** – Estar ligado a internet e dispor de dados de acesso
- **Cenário nominal:**
  - 1- Abrir a página de início de sessão
  - 2- O sistema devolve o formulário de início de sessão
  - 3- O utilizador insere os seus dados de conexão
  - 4- O sistema verifica se os dados inseridos são válidos
  - 5- O sistema devolve a página de início com o menu específico ao nível do utilizador
- **Cenários alternativos**

### **3-) Erro, dados de conexão incorreto**

- O sistema indica mensagem de erro
- O caso de utilização volta no ponto 3 do cenário nominal

#### **b. Caso de utilização «Gerir candidato»**

- **Objetivo** – Permite ao utilizador inserir novo candidato, consultar ou alterar informações, enviar email, gerir documentos do candidato, gerir atividade em relação ao candidato.

- **Ator** – utilizador do sistema

- **Pré condição** – Estar ligado a internet e dispor de dados de acesso

- **Cenário nominal:** Inserir novo candidato

1-O utilizador seleciona o menu «candidatos».

2-O sistema devolve a lista dos candidatos caso existirem na base de dados.

3-O utilizador seleciona o ícone-menu «Novo Candidato» ou prima «alt+n».

4-O sistema devolve o formulário de inscrição de novo candidato.

5-O utilizador insere os dados e clique no botão «guardar».

6-O sistema verifica se todos os campos são coerentes e que os obrigatórios estão preenchidos.

7-O sistema guarda os dados na base de dados.

8-O sistema fecha o formulário e atualiza a lista dos candidatos.

- **Cenários alternativos**

#### **3-a) Alterar dados de candidato existente**

-O utilizador clique no «Nome» ou «ID Number» do candidato.

-O sistema devolve o formulário com dados do candidato.

-O caso de utilização continua no ponto 5 do cenário nominal.

#### **3-b) Anular a operação**

-O utilizador clique em cancelar.

-O caso de utilização continua no ponto 8 do cenário nominal.

### **3-c) Pesquisar um candidato**

-O utilizador introduz a palavra de pesquisa.

-O sistema faz a pesquisa na base de dados.

-O sistema atualiza a lista dos candidatos conforme a pesquisa.

### **3-d) Filtrar candidatos**

-O utilizador seleciona o ícone-menu «Filtro».

-O sistema devolve o formulário de filtragem.

-O utilizador introduz critérios por qual filtrar e clique no botão «Filtrar».

- O sistema fecha o formulário e apresenta a lista dos candidatos filtrados.

### **3-a-2) Consultar tarefas e atividades do candidato**

-O utilizador seleciona o submenu «Tarefas & Atividades».

-O sistema devolve a lista das tarefas e atividades em relação a este candidato com opções: adicionar.

-Fim do caso de utilização.

### **3-a-3) Gerir documentos do candidato**

-O utilizador seleciona o submenu «Anexo».

-O sistema devolve a lista dos documentos do candidato com opções: alterar e adicionar.

-Fim do caso de utilização.

### **7-a) Erros ou incoerências detetados no preenchimento do formulário**

-O sistema mostra o formulário com os dados inseridos e indica os campos errados em vermelha.

-O utilizador corrige os erros e clique no botão «Save».

-O caso de utilização volta no ponto 6 do cenário nominal.

c. Caso de utilização «Gerir cliente»

- **Objetivo** – Permite ao manager inserir novo cliente, consultar ou alterar informações, enviar email, gerir documentos, definir campanhas, gerir atividade em relação ao cliente.
- **Ator** – manager
- **Pré condição** – Estar ligado a internet e dispor de dados de acesso
- **Senário nominal:** Inserir novo cliente
  - 1-O manager seleciona o menu «clientes».
  - 2-O sistema devolve a lista dos clientes caso existirem na base de dados.
  - 3-O manager seleciona o ícone-menu «Novo Cliente» ou prima «alt+n».
  - 4-O sistema devolve o formulário de inscrição de novo cliente.
  - 5-O manager insere os dados e clique no botão «guardar».
  - 6-O sistema verifica se todos os campos são coerentes e que os obrigatórios estão preenchidos.
  - 7-O sistema guarda os dados na base de dados.
  - 8-O sistema fecha o formulário e atualiza a lista dos clientes.
- **Cenários alternativos**
  - 3-a) Alterar dados de cliente existente**
    - O manager clica no «Nome» ou «ID Number» do cliente.
    - O sistema devolve o formulário com dados do cliente.
    - O caso de utilização continua no ponto 5 do cenário nominal.
  - 3-a-1) Consultar campanhas do cliente**
    - O manager seleciona o submenu «Campanhas».
    - O sistema devolve a lista das campanhas do cliente com opções: adicionar.
    - Fim do caso de utilização.
  - 3-a-2) Consultar detalhes duma campanhas do cliente**
    - O manager seleciona o submenu «Campanhas».

- O sistema devolve a lista das campanhas do cliente
- O manager seleciona a campanha que quer ver os detalhes
- O sistema abre a ficha da campanha com todos os detalhes
- Fim do caso de utilização.

### **3-a-3) Gerir documentos do candidato**

- O utilizador seleciona o submenu «anexo».
- O sistema devolve a lista dos documentos do cliente com opções: eliminar e adicionar.
- Fim do caso de utilização.

### **3-b) Anular a operação**

- O manager clica no botão cancelar.
- O caso de utilização continua no ponto 8 do cenário nominal.

### **3-c) Pesquisar um cliente**

- O manager introduz a palavra de pesquisa.
- O sistema faz a pesquisa na base de dados.
- O sistema atualiza a lista dos clientes conforme a pesquisa.
- Fim do caso de utilização.

### **7-a) Erros ou incoerências detetados no preenchimento do formulário**

- O sistema mostra o formulário com os dados inseridos e indica os campos errados em vermelha.
- O utilizador corrige os erros e clique no botão «Save».
- O caso de utilização volta no ponto 6 do cenário nominal.

#### **d. Caso de utilização «Gerir utilizador do sistema»**

- **Objetivo** – Permite ao manager alterar dados dos utilizadores e criar novo utilizador do sistema
- **Ator** – manager

- **Pré condição** – Estar ligado a internet, ter o nível de administrador e dispor de dados de acesso
- **Cenário nominal:** Adicionar um novo utilizador
  - 1-O manager seleciona o menu «Admin».
  - 2-O sistema devolve a lista dos utilizadores.
  - 3-O manager seleciona o ícone-menu «Novo utilizador».
  - 4-O sistema devolve o formulário para inserir dados do novo utilizador.
  - 5-O manager insere dados do novo utilizador e clique no botão «Save».
  - 6-O sistema verifica se todos os campos são coerentes, que os obrigatórios estão preenchidos.
  - 7-O sistema guarda os dados na base de dados.
  - 8-O sistema fecha o formulário e atualiza a lista dos utilizadores.
- **Cenários alternativos**
  - 3-a) Alterar dados de utilizador existente**
    - O manager clique no ícone-menu «Alterar» frente ao utilizador.
    - O sistema devolve o formulário com dados do utilizador.
    - O caso de utilização continua no ponto 5 do cenário nominal.
  - 7-a) Erros ou incoerências detetados no preenchimento do formulário**
    - O sistema mostra o formulário com os dados inseridos e indica os campos errados em vermelha.
    - O consultor corrige os erros e clique no botão «Save».
    - O caso de utilização volta no ponto 6 do cenário nominal.

### 2.3.2. Diagramas de atividade

O diagrama de atividade permite ter uma visão do encadeamento das atividades de uma operação ou dum caso de utilização. Apresentamos aqui diagramas de atividade dos principais casos de utilização:

#### a. Diagrama de atividade «Login»

Este diagrama descreve as diferentes etapas a seguir no para ter acesso ao sistema. Depois de abrir a aplicação, o sistema pede o *username* e a *password*, se estes não são validos o sistema volta a pedir de novo, se foram validos o sistema redireciona o utilizador a pagina inicial.

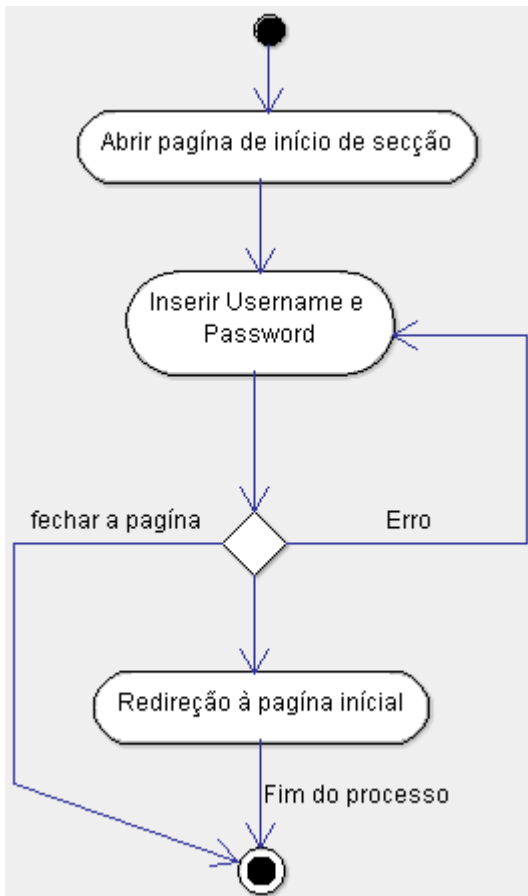


Figura 2-Diagrama de atividade «Login»

b. Diagrama de atividade «Novo candidato»

Os passos necessários para registar um novo candidato são descritas no seguinte diagrama: Depois de entrar no menu dos candidatos, o utilizador seleciona o icon-menu «novo candidato», insere os dados, se não houver erros ou incoerência, os dados são guardados.

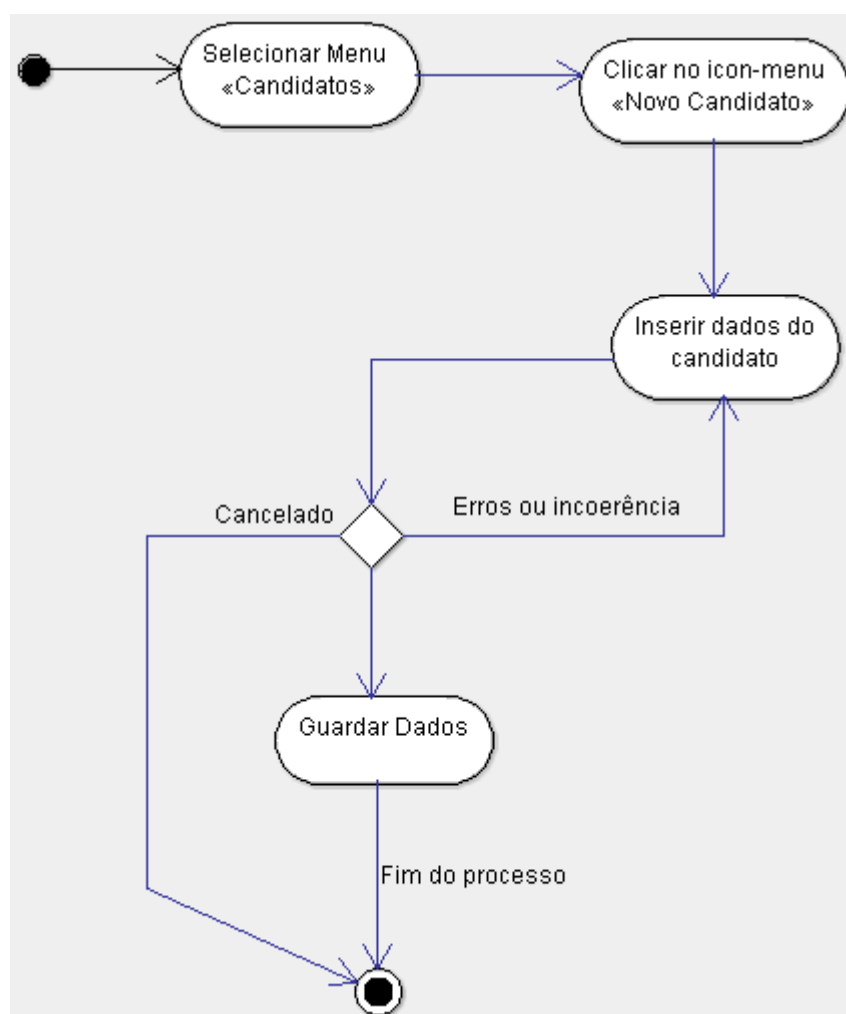


Figura 3-Diagrama de atividade «Novo candidato»

c. Diagrama de atividade «Alterar dados do candidato»

Este diagrama ilustra os diferentes passos para alterar dados do candidato. O utilizador seleciona o menu «candidatos», clica no nome ou no ID do candidato, altera os dados e clica em guarda. Em caso de erros ou incoerência o utilizador volta a inserir os dados, no caso contrário os dados ficam guardados.

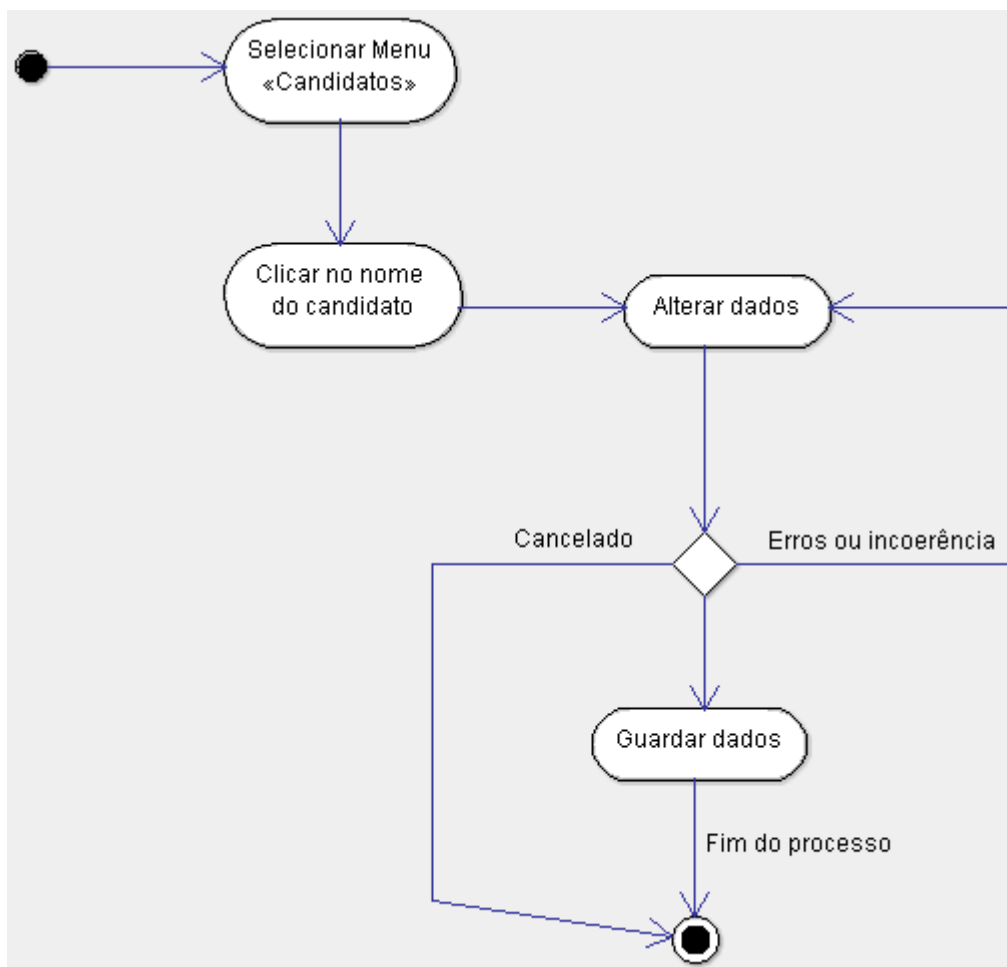


Figura 4-Diagrama de atividade «Alterar dados do candidato»

d. Diagrama de atividade «Criar atividade ao candidato»

As etapas necessárias para criar uma atividade ao candidato são descritas neste diagrama. O utilizador seleciona o menu «candidatos», clica no nome ou ID do candidato, seleciona o sub-menu «tarefas & atividades», seleciona o icon-menu «nova atividade», insere dados da atividade e clica em guardar. Em caso de erros ou incoerência o utilizador volta a inserir os dados, no caso contrário os dados ficam guardados.

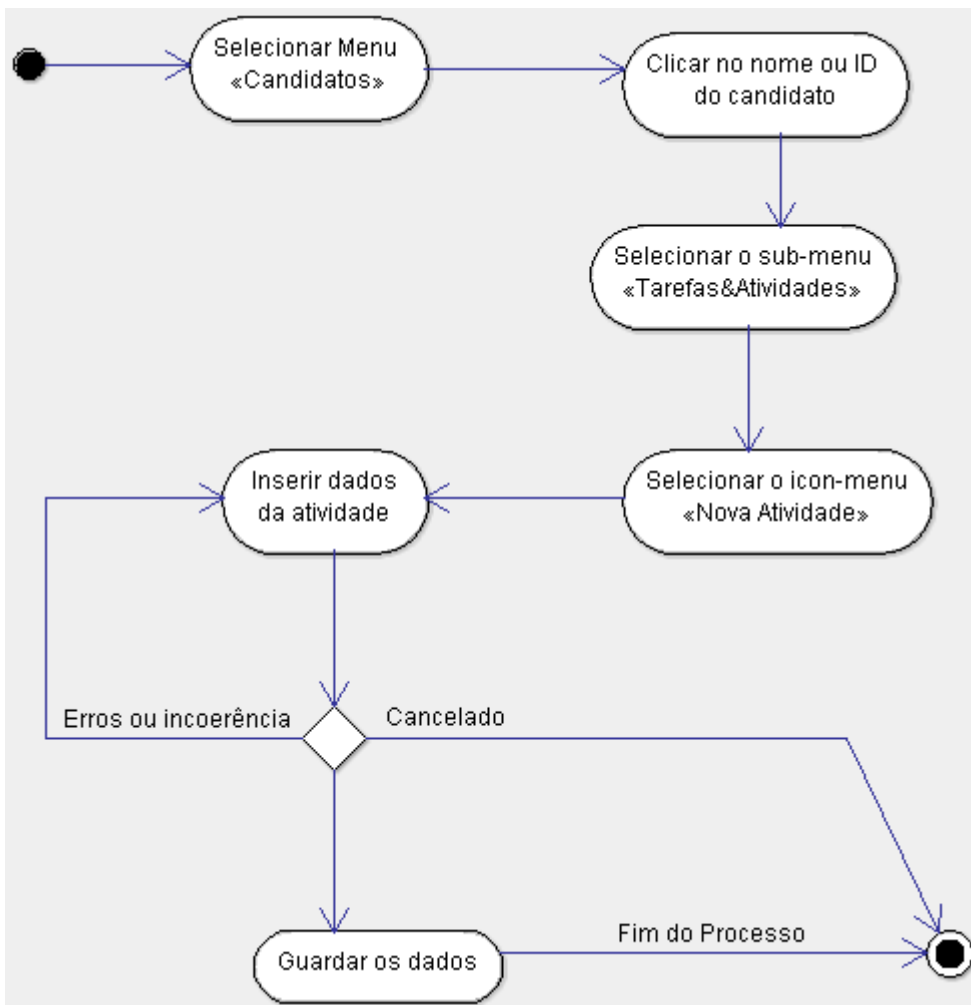


Figura 5-Diagrama de atividade «Criar atividade ao candidato»

e. Diagrama de atividade «Inserir novo cliente»

Este diagrama descreve os passos a seguir na inserção de novo cliente. O manager seleciona o menu «clientes», clica no icon-menu «novo cliente», insere os dados do cliente e clica em guardar. Em caso de erros ou incoerência o manager volta a inserir os dados, no caso contrário os dados ficam guardados.

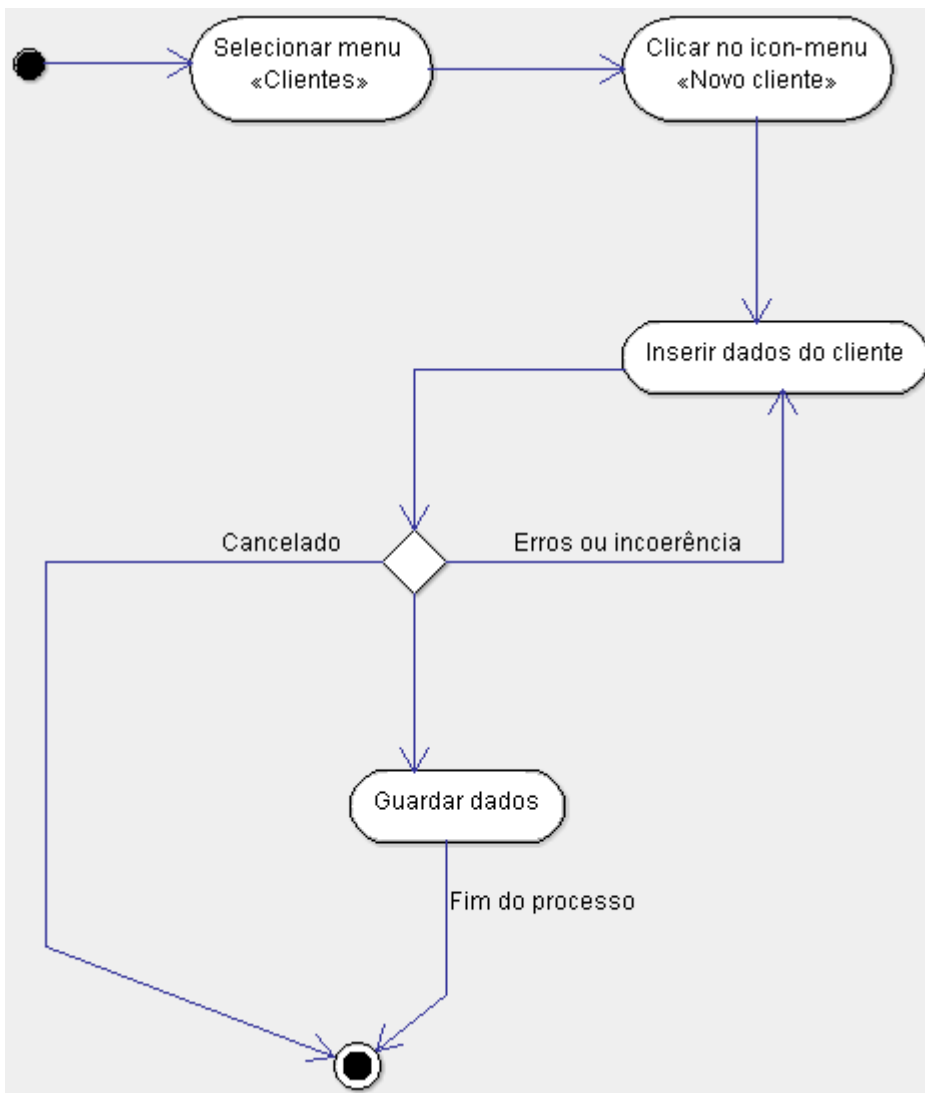


Figura 6-Diagrama de atividade «Inserir novo cliente»

f. Diagrama de atividade «Criar novo utilizador»

Com este diagrama descrevemos as etapas para criar um novo utilizador do sistema. O manager seleciona o menu «admin», clica no icon-menu «novo utilizador», insere os dados e guarda. Em caso de erros ou incoerência o manager volta a inserir os dados, no caso contrário os dados ficam guardados.

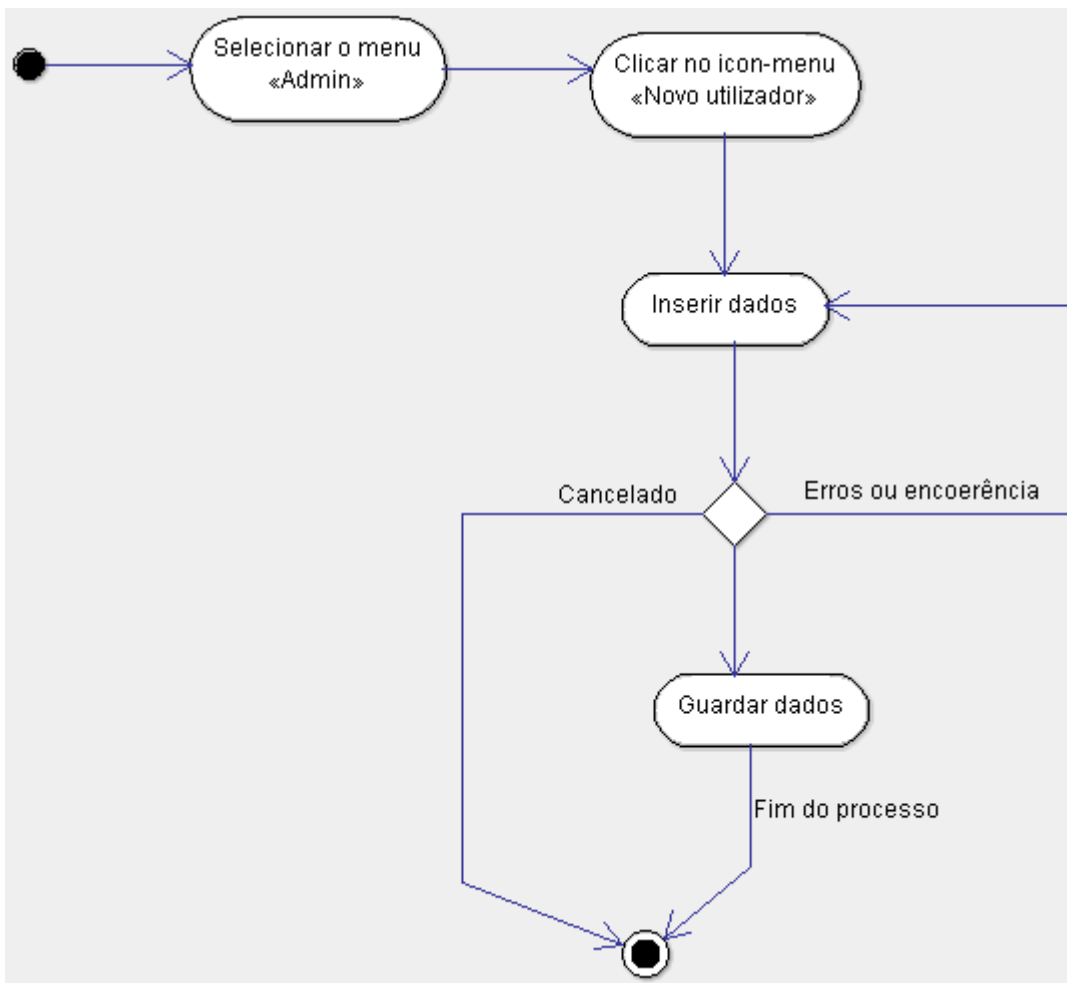


Figura 7-Diagrama de atividade «Criar novo utilizador»

### 2.3.3. Diagrama de classes

Enquanto o diagrama de caso de utilização nos dá uma visão dos atores e das suas ações, o diagrama de classe apresenta a descrição estática do sistema. Permite fazer uma representação abstrata dos objetos do futuro sistema que irão interagir em conjunto para realizar os casos de utilização.

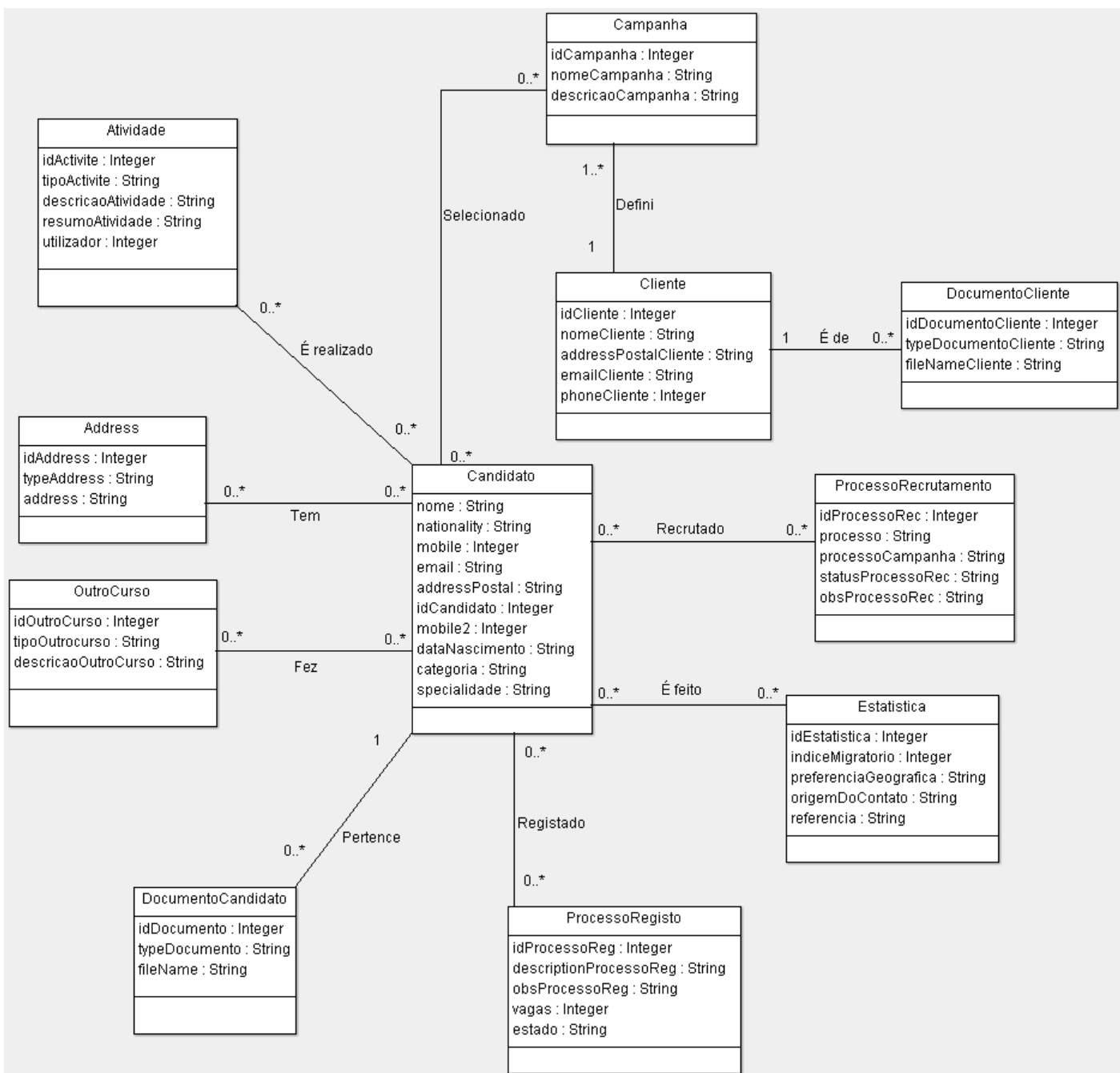


Figura 8-Diagrama de classes

### 2.3.4. Modelo relacional

A seguir podemos elaborar o modelo relacional que permite representar toda a informação do diagrama de classes de forma completa, isto é, sem incoerências e redundâncias, recorrendo à aplicação de regras de transformação de classes em tabelas.

Candidato (idCandidato, nome, nationality, mobile, mobile2, email, addressPostal, dataNascimento, categoria, especialidade)

Cliente (idCliente, nomeCliente, addressPostalCliente, emailCliente, phoneCliente)

Campanha (idCampanha, nomeCampanha, descricaoCampanha, vagas, estado, *idCliente*)

Selecionado (idCandidato, idCampanha)

Atividade (idAtividade, tipoActivite, descricaoAtividade, resumoAtividade, utilizador)

Realizado (idAtividade, idCandidato)

Address (idAddress, typeAddress, address)

Tem (idAddress, idCandidato)

OutroCurso (idOutroCurso, tipoOutrocurso, descricaoOutroCurso)

Fez (idOutroCurso, idCandidato)

DocumentoCandidato (idDocumento, typeDocumento, fileName, *idCandidato*)

DocumentoCliente (idDocumentoCliente, typeDocumentoCliente, fileNameCliente, *idCliente*)

ProcessoRegisto (idProcessoReg, descriptionProcessoReg, obsProcessoReg)

Registado (idProcessoReg, idCandidato)

ProcessoRecrutamento (idProcessoRec, processo, processoCampanha, statusProcessoRec, obsProcessoRec)

Recrutado (idProcessoRec, idCandidato)

Estatistica (idEstatistica, indiceMigratorio, preferenciaGeografica, origemDoContato, referencia)

Feito (idEstatistica, idCandidato)

#### *2.4. Conclusão*

Ao fim deste capítulo temos informação suficiente para passar à fase de implementação, onde falaremos da arquitetura do sistema e soluções utilizadas, do processo de implementação e proceder a alguns testes.

### 3. FASE DE IMPLEMENTAÇÃO

Neste nível começa a fase de programação. Após o trabalho de análise definimos 4 módulos que irão compor o menu principal do nosso sistema:

- **Dashboard** – página inicial da aplicação, apresenta todas as atividades realizadas e pendentes.
- **Candidatos** – menu de gestão dos candidatos.
- **Clientes** – menu de gestão dos clientes.
- **Admin** – menu de gestão de utilizadores

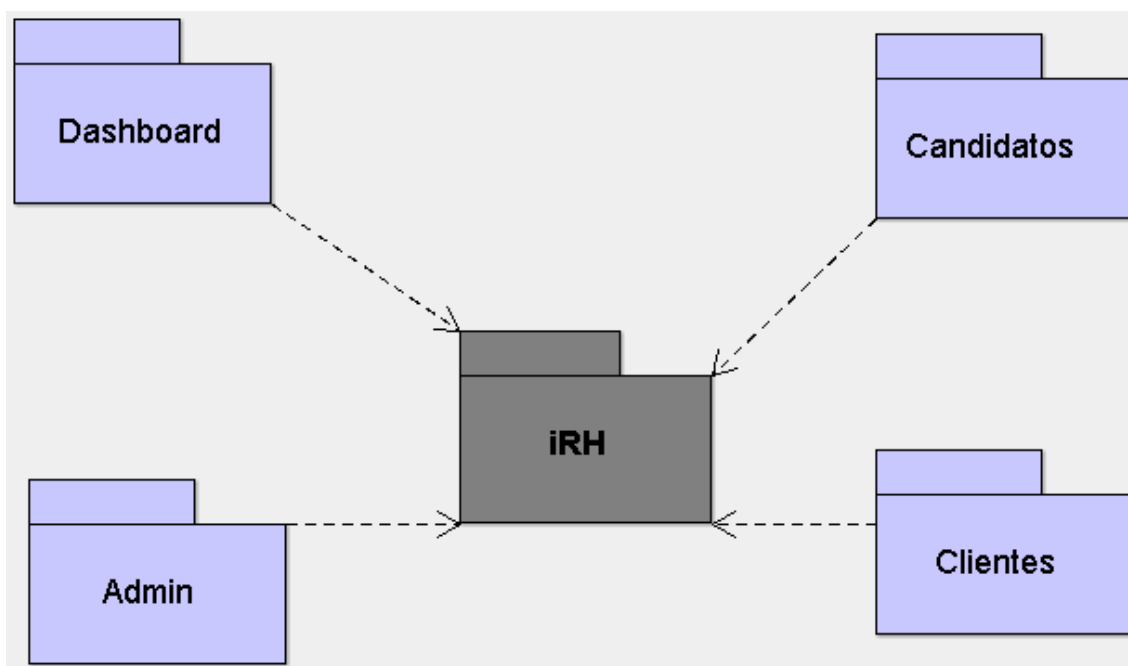


Figura 9 - Módulos da aplicação

#### 3.1. Arquitetura da solução e tecnologias utilizadas

A nível de arquitetura lógica utilizamos a arquitetura 3-tiers implementando o modelo MVC, cada módulo tem o seu respetivo *view*, *controller* e *model*. A figura 11 ilustra o modo de funcionamento da arquitetura MVC.

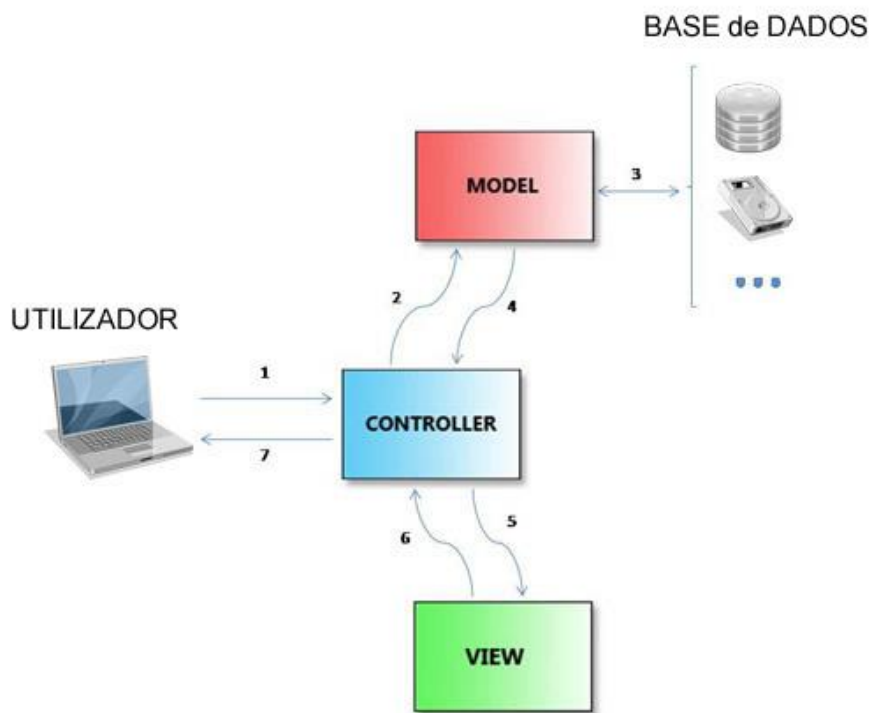


Figura 10-Arquitetura MVC

1- O utilizador faz um pedido no sistema.

2- O *controller* a cargo deste pedido solicita ao *model* que vai buscar as informações na base de dados.

3- O *model* recupera as informações na base de dados.

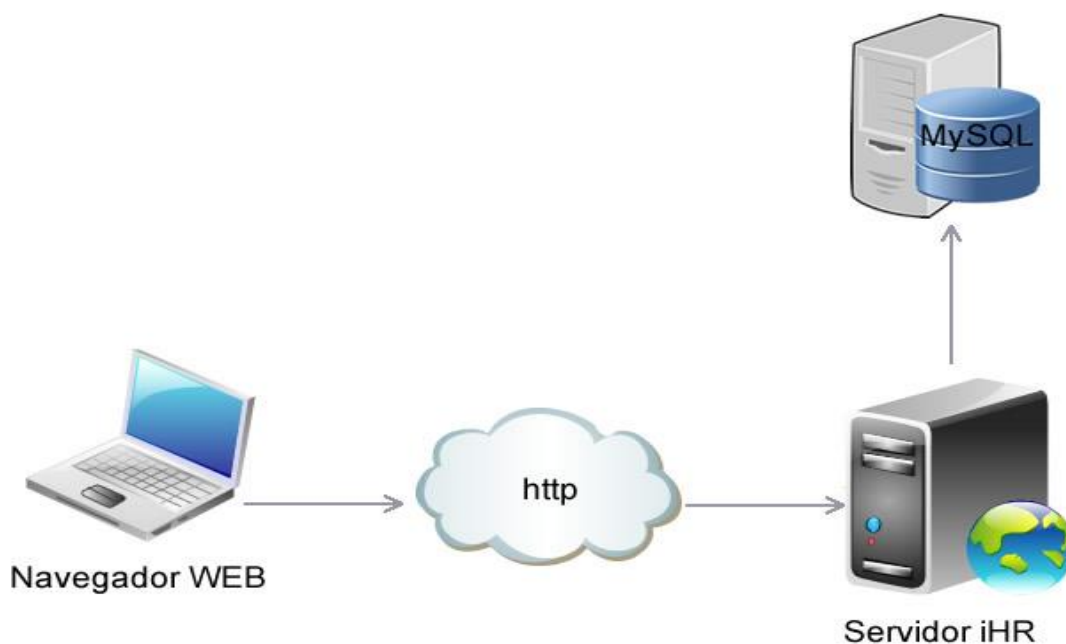
4- O *model* devolve as informações recolhidas ao controlador.

5- O *controller* seleciona a *view* que vai integrar os dados no HTML

6- A *view* integra os dados e espera o controlador para a apresentação

7- O *controller* apresenta os dados (a *view*) ao utilizador

A nível da arquitetura física, a figura 12 ilustra os equipamentos que garantam o funcionamento da nossa aplicação.



*Figura 11-Arquitetura física*

A nível de tecnologias utilizadas, como mencionado em cima utilizamos **Codeigniter** como *framework*, **HTML** e **PHP** como principais linguagens de programação. Também utilizamos **Ajax** para gerir pedidos do utilizador de forma fluido e **JQuery** para a boa apresentação. Como editor de texto utilizamos **Sublime Text**, um editor que suporta o modelo MVC.

### *3.2. Processo de implementação*

O projeto foi dividido em módulos considerados tarefas e foi definido um plano de desenvolvimento das tarefas. Com equipa de desenvolvimento composto de dois programadores, também foi definido um processo de coordenação de contributos dos programadores.

#### *3.2.1. Plano de desenvolvimento das tarefas*

Aqui baseado sobre a experiencia, o tempo útil dedicado ao desenvolvimento das tarefas, apresentamos um diagrama de *Gantt* que mostra uma estimativa do tempo para a realização das diferentes tarefas.

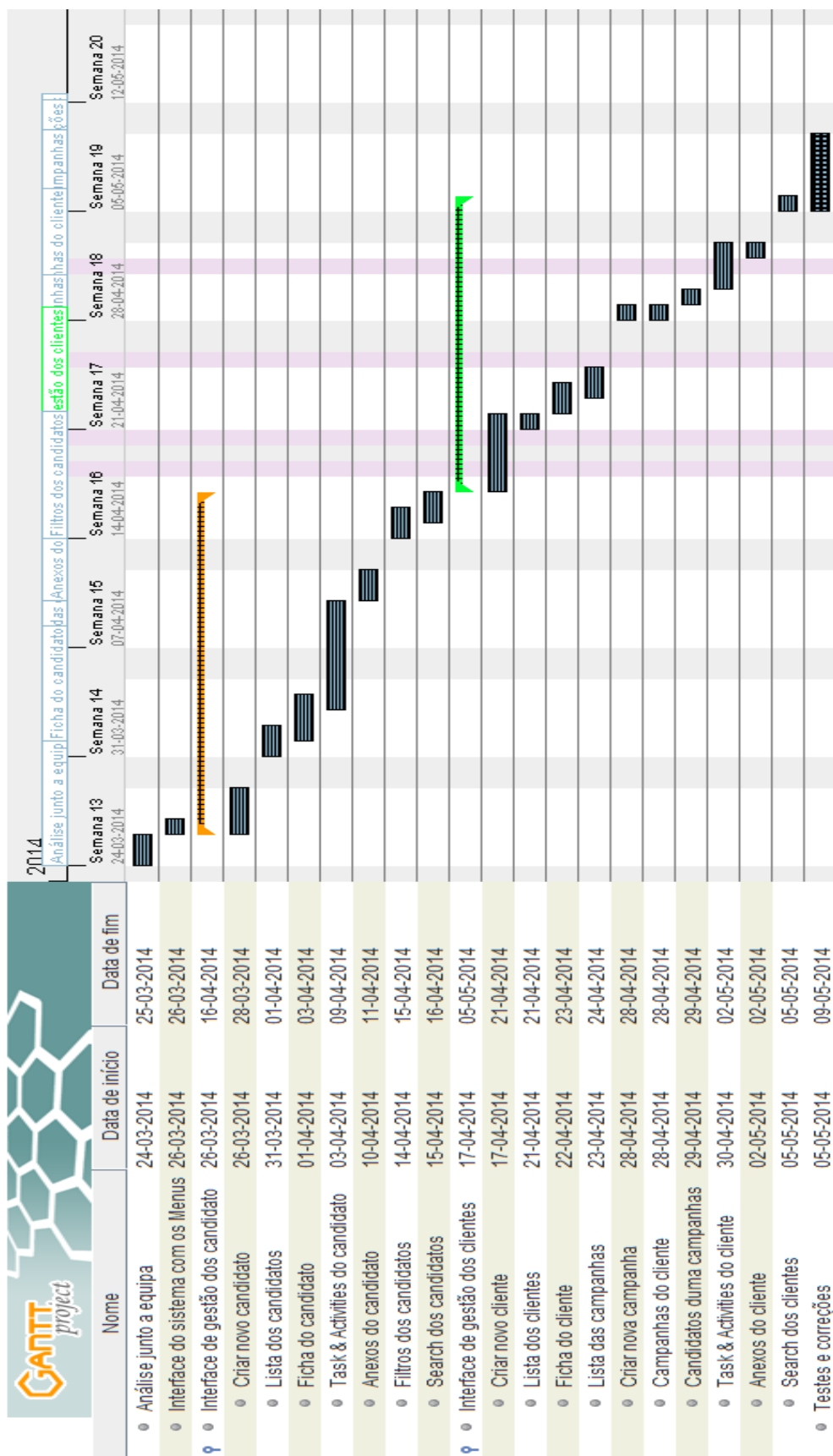


Figura 12 Diagrama de Gantt

### 3.2.2. Processo de coordenação de contributos dos programadores

Para nos permitir de trabalhar de forma independente, utilizamos TortoiseSVN ([www.tortoisesvn.net](http://www.tortoisesvn.net)), uma ferramenta que permite-nos fazer ambos atualização no servidor de forma independente mantendo sempre o código da aplicação atualizado, fazendo *update* e *commit*. O TortoiseSVN permite também controlar as versões da aplicação.

### 3.3. Configuração do PC - Servidor

Para o iHR funcionar corretamente devem estar devidamente instalados e configurados um conjunto de tecnologias. Os seguintes passos ilustram como pode ser feita a configuração. Esta foi feita no sistema Windows 7 e utilizando o servidor Apache XAMPP que facilita bastante a instalação por suportar PHP e MySQL:

- Instalar xampp v.1.8.3 – servidor apache que suporta PHP (5.5.9) e MySQL (v.5.6.16);
- Criar a base de dados no phpmyadmin (<http://localhost/phpmyadmin/>) a partir do modelo relacional e do diagrama de classes;
- Copiar o código fonte (pasta iHR) para `\xampp\htdocs\`
- Configurar o ficheiro `config.php` e `database.php` no `\xampp\htdocs\ihr\application\config\`
- Abrir no browser <http://localhost/ihr/>

Podem ainda ser feitas outras alterações como o número da porta (caso a 80 que é a porta padrão não esta disponível).

### 3.4. Fase de teste por cenário de utilização

Como previsto no ponto 2, efetuamos alguns testes para garantir que o sistema responde aos requisitos. No anexo 1 apresentamos algumas capturas de teste dos processos de login, inserção de novo candidato, consulta da ficha do candidato, planificação duma atividade.

### 3.5. Conclusão

Esta última parte neste documento parece ser curta mas na prática é a fase mais longa. Pois é aqui que se desenvolvem todas as tarefas uma a uma efetuando no final de cada tarefa e os testes que validam a passagem para a tarefa seguinte.

## 4. CONCLUSÃO GERAL

Sistemas de Informação para negócios juntam funções de negócio e módulos de informação para estabelecer canais de comunicação eficazes que são úteis para a tomada de decisões oportunas e precisas. Neste ponto de vista, o S24Group decidiu desenvolver uma aplicação que ira auxiliar a equipa em cargo de gestão das oportunidades. Antes a equipa desempenhava as suas tarefas utilizando Excel o que dificultava a análise dos dados. Agora com a utilização da aplicação iHR, a equipa ganhou eficiência em relação a seguimento dos processos de recrutamento e a gestão dos documentos dos candidatos.

### 4.1. Objetivos realizados

Como definido na fase de análise, todos os requisitos são cumpridos, a aplicação foi testado com sucesso e posto em produção. Nomeadamente os módulos de gestão dos candidatos, de gestão dos clientes, o dashboard que permite o controlo das tarefas e atividades e o módulo de gestão dos utilizadores funcionam como deve ser.

### 4.2. Limitações e trabalho futuro

Em termos de limitações, no início do projeto tive dificuldades a configurar a base de dados MySQL com o *framework* CodeIgniter por ser a primeira vez a utiliza-lo, mas depois de ler alguns tutoriais foi ultrapassado. Para trabalhos futuros fica o desenvolvimento do portal de acesso ao cliente, recomendamos a implementação dum sistema de alerta vermelha no «Dashboard» para atividades pendentes. Ainda recomendamos também o desenvolvimento de um módulo dedicado a gestão

das campanhas onde poderão aparecer gráficos indicando o estado de uma determinada campanha.

#### *4.3. Apreciação final*

O desenvolvimento deste projeto permitiu-me consolidar conceitos adquiridos ao longo do meu curso. A fase de análise confirmou-me que na área de programação aprende-se todos os dias, com ferramentas e métodos, na área da informática em geral.

## 5. BIBLIOGRAFIA

- Conallen, J. (1999). Modeling Web application architectures with UML. *Communications of the ACM* 42 (10), 63-70.
- Thakare, R. Kakade, S. Sapre, B. e Meshram, B. (2012). Spring MVC Framework for Web 2.0. *International Journal of Engineering Innovation & Research* 1 (2), 242-247.
- Utpatadevi, N. Sudana, A. e Cahyawan, A. (2012). Implementation of MVC (Model-View-Controller) Architectural to Academic Management Information System with Android Platform Base. *International Journal of Computer Applications* 57 (8), 1-6.
- El-Sharkawi, A. Shouman, A. e Lasheen, S. (2013). Service Oriented Architecture for Remote Sensing Satellite Telemetry Data Implemented on Grid Computing. *International Journal of Computer Applications* 78 (14), 42-51.
- Hustinawati, Himawan, A. e Latifah (2014). Performance Analysis Framework Codeigniter and CakePHP in Website Creation. *International Journal of Computer Applications* 94 (20), 6-11.
- Simo, J. (2013) josemmsimo/PHP FRAMEWORKS COMPARISON: ZEND, SYMFONY, CODEIGNITER, YII AND CAKE PHP. Disponível em:  
<http://josemmsimo.wordpress.com/2013/12/02/php-frameworks-comparison-zend-symfony-codeigniter-yii-and-cake-php/> Acesso em: 24/05/2014.
- Lee, S. (2012). Unified Modeling Language (UML) for Database Systems and Computer Applications. *International Journal of Database Theory and Application* 5 (1), 157-163.
- Kothamasu, D. Gao, X. e Jiang, Z. [cs.wcupa.edu/](http://www.cs.wcupa.edu/) Web Application Development Using UML Student Grading System. Disponível em: <http://www.cs.wcupa.edu/~zjiang/IndependentStudy6.pdf> Acesso em: 12/06/2014.
- Gabay, J. (2008). *Uml 2, Analyse Et Conception*, Ed. 2008. Paris : Dunod.

## 6. ANEXOS

### 6.1. Captura de ecrã da fase de teste por cenário de utilização

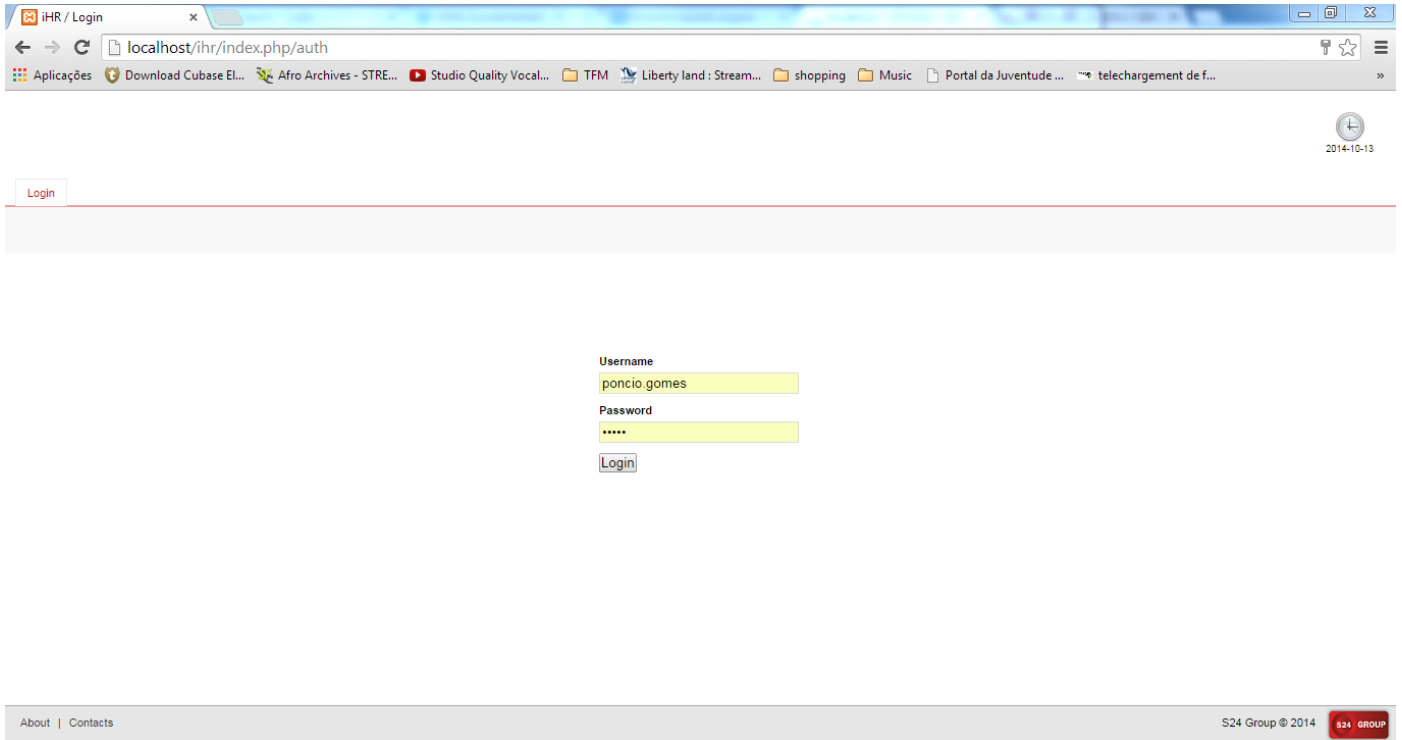


Figura 13 - Teste «Login»

Browser: iHR / Dashboard | localhost:1285/ihr/index.php/home

Navigation: Dashboard, Candidatos, Clientes, Admin, 2014-12-31, Logout

### Dashboard

#### My Activities

Date	Type	Responsible	Professional	Summary	Status
2014-12-04 09:30:00	Telephone	poncio.gomes	<a href="#">José Fernandes</a>	Marcar entrevista para a campanha em Angola	Done
2014-12-04 16:30:00	Presential	poncio.gomes	<a href="#">José Fernandes</a>	Entrevista de pré-selecção para a campanha em Ang	Done
2014-12-05 09:57:00	Telephone	poncio.gomes	<a href="#">Maria de Nascimento</a>	Marcar entrevista para o dia 9/12/2014	Past Event
2014-12-05 10:19:00	Telephone	poncio.gomes	<a href="#">Vanderson Jesus</a>	Marcar entrevista para o dia 11/12/2014	Past Event
2014-12-05 17:04:00	Telephone	poncio.gomes	<a href="#">Rui Silva</a>	Marcar entrevista para o dia 10/12/2014	Past Event

Page 1 of 1 | View 1 - 5 of 5

#### All Activities

Date	Type	Responsible	Professional	Summary	Status
2014-12-04 10:55:00	Telephone	manuel.sousa	<a href="#">Marisa Santos</a>	Marcar entrevista	Done
2014-12-04 11:04:00	Telephone	manuel.sousa	<a href="#">Rui Silva</a>	Marcar entrevista para a campanha CLCH	Done
2014-12-04 11:07:00	Telephone	manuel.sousa	<a href="#">Marta Da Silva</a>	Marcar entrevista para a campanha HPP Faro	Done
2014-12-05 11:35:00	Telephone	tmaraviha	<a href="#">Ricardo Maquinhães</a>	Marcar entrevista para conhecer a campanha CLCH	Done
2014-12-09 10:24:00	Email	manuel.sousa	<a href="#">Necia Gomes Fernandes</a>	Marcar uma entrevista por Skype	Waiting

Page 1 of 1 | View 1 - 11 of 11

About | Contacts | S24 Group © 2014 | S24 GROUP

Figura 14 - Teste «Dashboard»

Browser: iHR / Candidatos | localhost/ihr/index.php/candidates/

Navigation: Dashboard, Candidatos, Clientes, 2014-10-13, Logout

### Candidatos

Search: Search by Name, Phone, Email or ID Number

ID Number	Category	Name	Nationality	Mobile	Email	Task Date	Task Summary	Task Status
13	01 Nursing	<a href="#">André Pereira</a>	Portuguese	936547821	<a href="#">andrep@gmail.com</a>			
9	03 HCPC	<a href="#">Eugenio Dias</a>	Portuguese	964587253		2014-10-13 01:05:00	Pedido de certificado de habilitações	Done
12	01 Nursing	<a href="#">Filipa Da Costa</a>	Portuguese	914786235		2014-10-14 10:00:00	Marcar entrevista	Waiting
8	02 Doctor	<a href="#">Isabel Gomes</a>	Portuguese	914587632				
2	02 Doctor	<a href="#">Jorge Sampaio</a>	Portuguese	936547825		2014-10-13 09:00:00	Marcar uma entrevista	Waiting
7	02 Doctor	<a href="#">José Fernandes</a>	Portuguese	9205784563				
4	02 Doctor	<a href="#">Luisa Silva</a>	Portuguese	915286452		2014-10-17 09:00:00	Entrevista de preselecção	Waiting
6	03 HCPC	<a href="#">Mamadou Bá</a>	Portuguese	932587456				
1	01 Nursing	<a href="#">Maria Semedo</a>	Portuguese	965478521		2014-10-13 10:00:00	Ligar para marcar uma entrevista	Waiting
10	03 HCPC	<a href="#">Paula Correia</a>	Portuguese	967896523				
11	01 Nursing	<a href="#">Solange Ndiaye</a>	Senegalese	96857231		2014-10-14 11:00:00	Marcar entrevista	Waiting
5	02 Doctor	<a href="#">Teresa Sampaio</a>	Portuguese	926874582		2014-10-14 10:00:00	Marcar entrevista	Waiting
3	01 Nursing	<a href="#">Tiaqo Mota</a>	Portuguese	914588795		2014-10-13 10:50:00	Pedir para enviar CV, Certificados de habilitações	Waiting

Page 1 of 1 | View 1 - 13 of 13

About | Contacts | S24 Group © 2014 | S24 GROUP

Figura 15 - Teste «Lista dos candidatos»

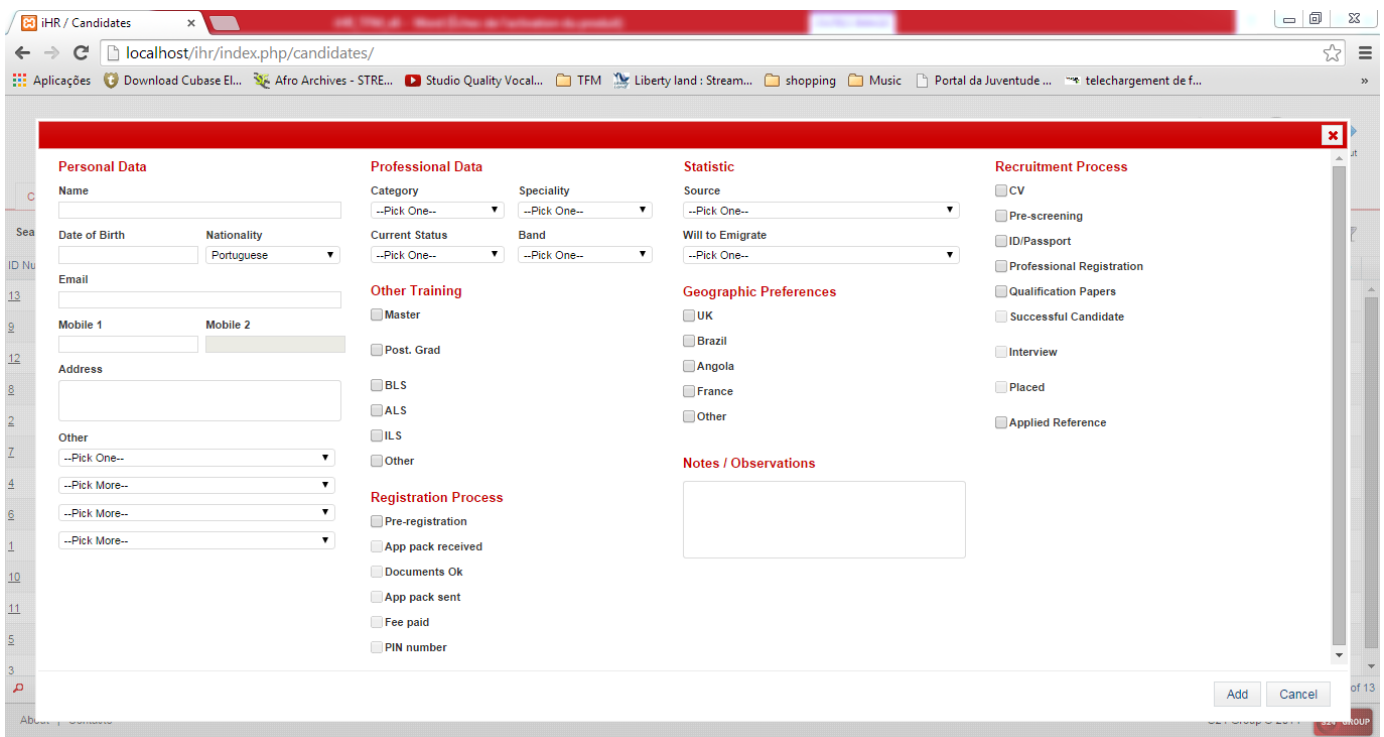


Figura 16 – Teste «Novo candidato»

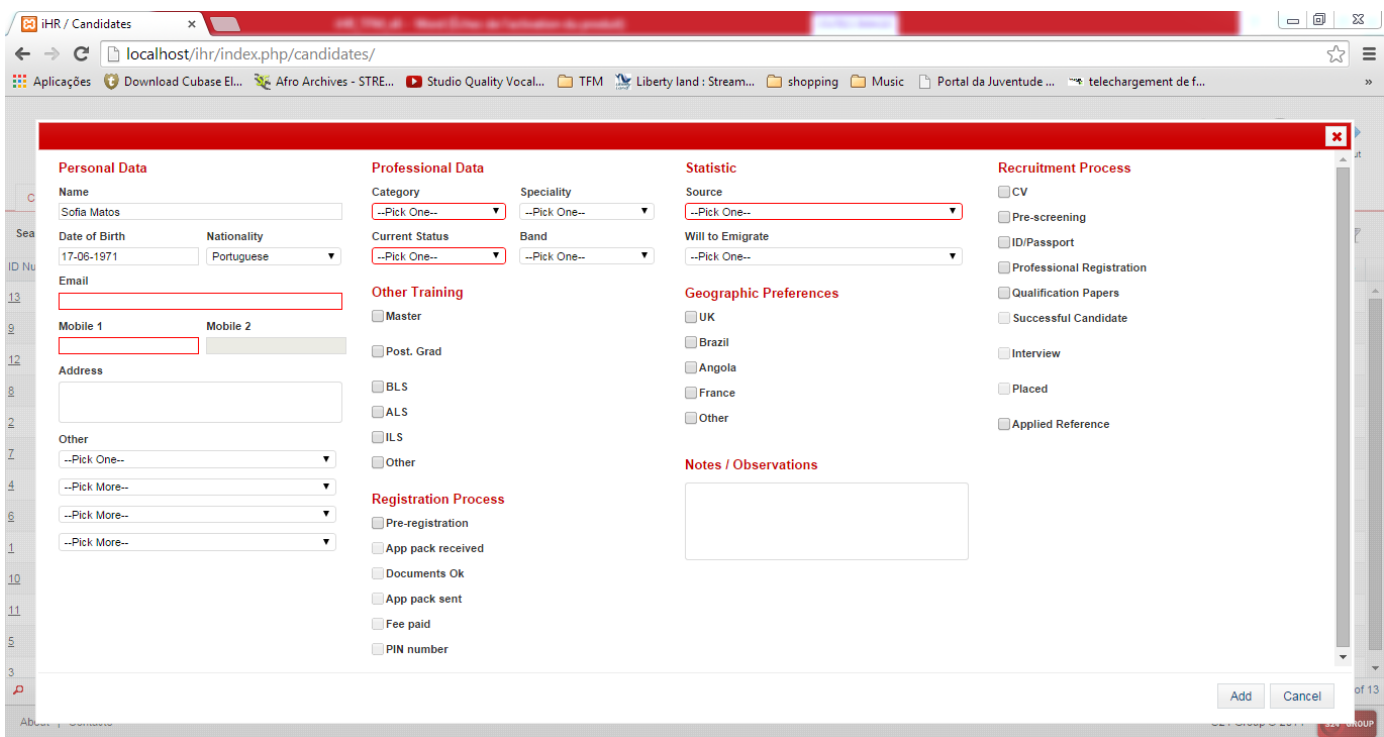


Figura 17 – Teste «Novo candidato com erros»

iHR / Candidates - Google Chrome  
localhost/ihr/index.php/candidates/info/1

**Maria Semedo**  
05 Saúde Infantil e Pediátrica ( 01 Nursing )

Candidato Tarefas & Atividades Anexos

**Personal Data**  
 Name: Maria Semedo  
 Date of Birth: 16-05-1977 Age: 37  
 Nationality: Portuguese  
 Mobile 1: 965478521 Mobile 2:  
 Email:  
 Address: Rua António Jesus nº24 - 1ªEsq, Lisboa  
 Other: --Pick One--  
 --Pick More--  
 --Pick More--  
 --Pick More--

**Professional Data**  
 Category: 01 Nursing Speciality: 05 Saúde Infantil e Pr  
 Current Status: Working Band: 5  
 Will to Emigrate: 7

**Other Training**  
 Master  
 Post. Grad  
 BLS  
 ALS  
 ILS  
 Other

**Registration Process**  
 Pre-registration  
 App pack received  
 Documents Ok  
 App pack sent  
 Fee paid  
 PIN number

**Statistic**  
 Source: LinkedIn  
 Will to Emigrate: 7

**Geographic Preferences**  
 UK  
 Brazil  
 Angola  
 France  
 Other

**Recruitment Process**  
 CV  
 Pre-screening  
 CLCH Enfermagem  
 Valid  
 Not Valid  
 ID/Passport  
 Professional Registration  
 Qualification Papers  
 Successful Candidate  
 CLCH Enfermagem  
 Yes  
 No  
 Interview  
 Placed  
 Applied Reference

**Notes / Observations**

Figura 18 – Teste «Ficha do candidato»

iHR / Candidates - Google Chrome  
localhost/ihr/index.php/candidates/info/4

**Luisa Silva**  
Cardiology ( 02 Doctor )

Candidato Tarefas & Atividades Anexos

Date	Type	Responsible	Summary	Description	Status
2014-10-13 09:00:0	Email	poncio.gomes	Pedido de documentos	CV atualizado Habilitações	Waiting
2014-10-17 09:00:0	Presential	poncio.gomes	Entrevista de preseleção	Entrevista no dia 20 para conhecer a campanha e as condições	Waiting

Page 1 of 1  
View 1 - 2 of 2

Next events:  
 13/10/2014 09:00  
 Email  
 Pedido de documentos  
 17/10/2014 09:00  
 Presential  
 Entrevista de preseleção

October 2014  
 mon tue wed thu fri sat sun  
 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 31

localhost/ihr/index.php/candidates/info/4#tasks  
 Page 1 of 1  
 No records to view

Figura 19 – Teste «Tarefas & Atividades ao candidato»

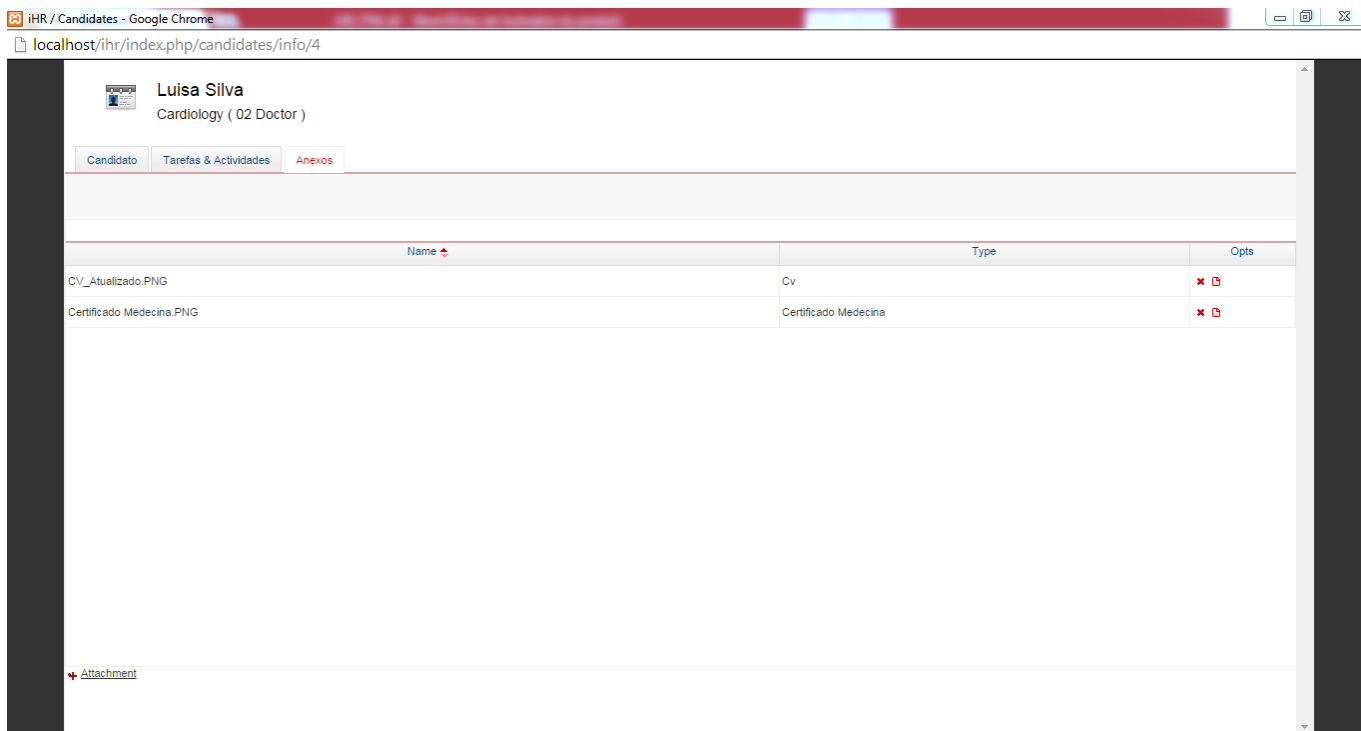


Figura 20 – Teste «Anexos do candidato»

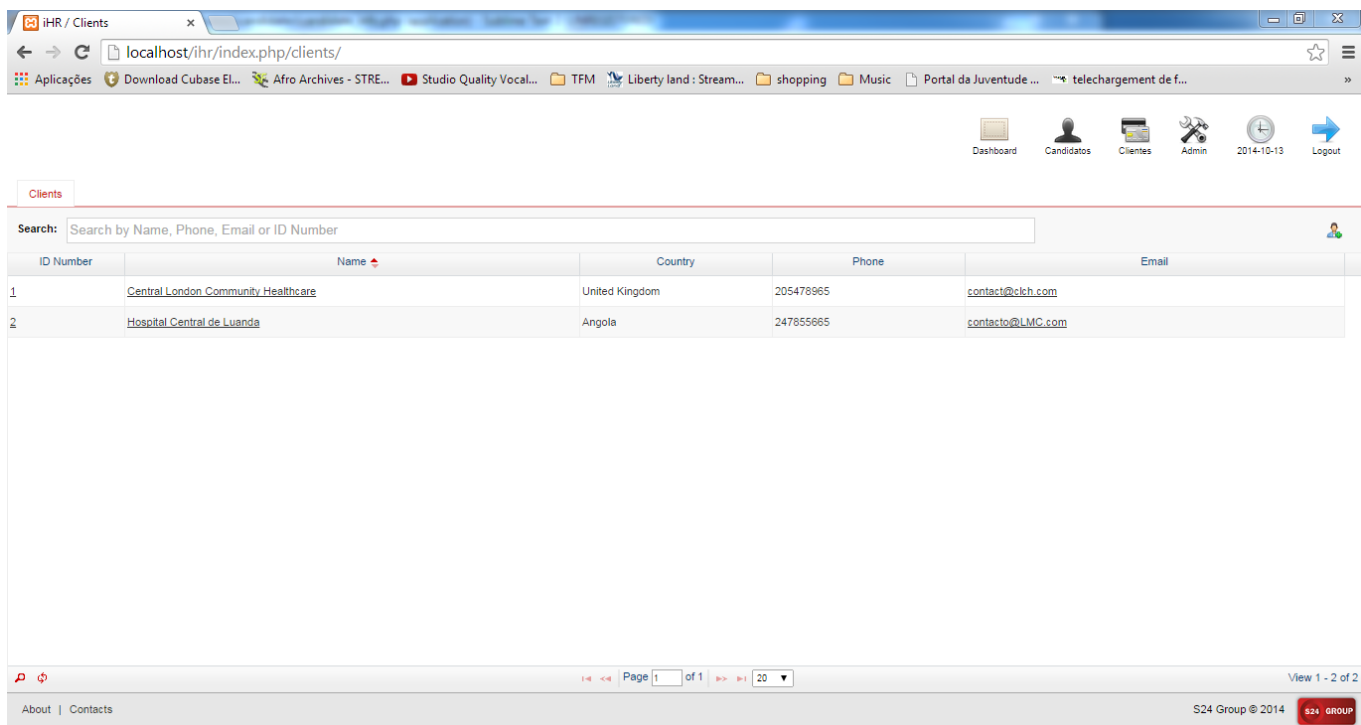


Figura 21 – Teste «Lista dos clientes»

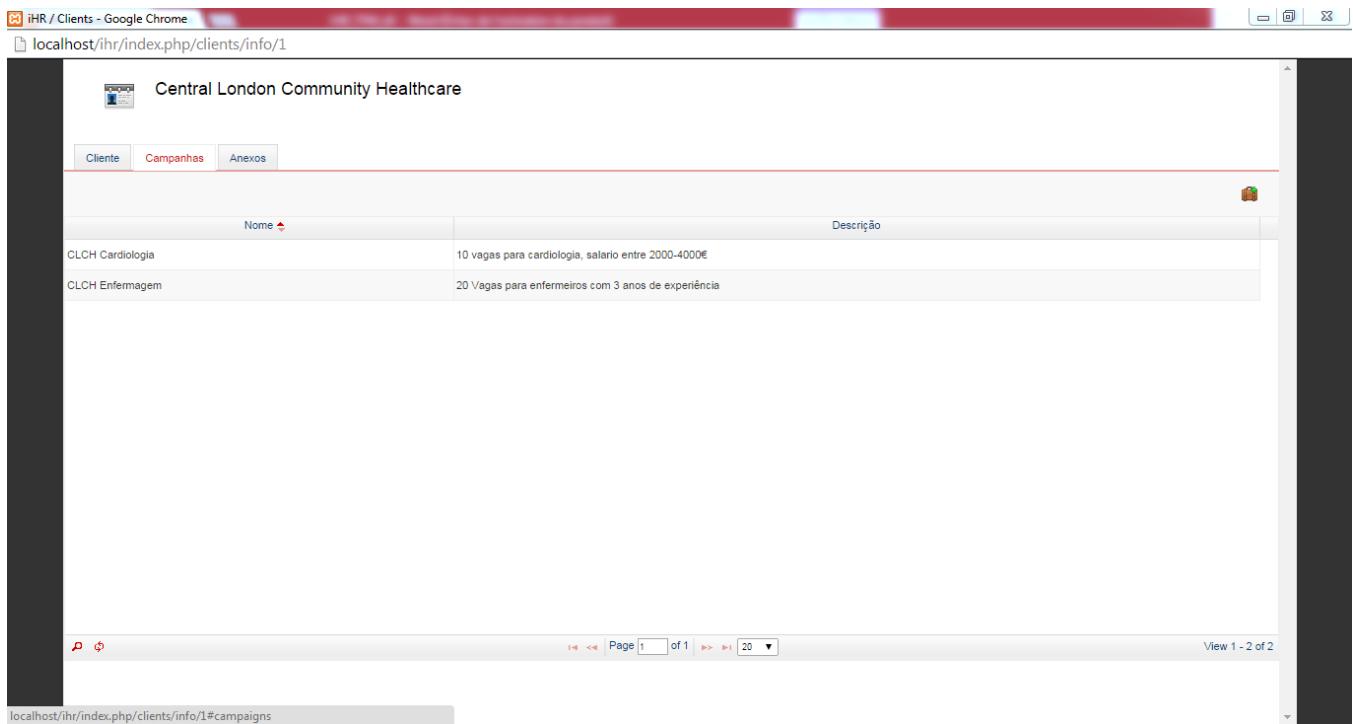


Figura 22 – Teste «Lista das campanhas do cliente»

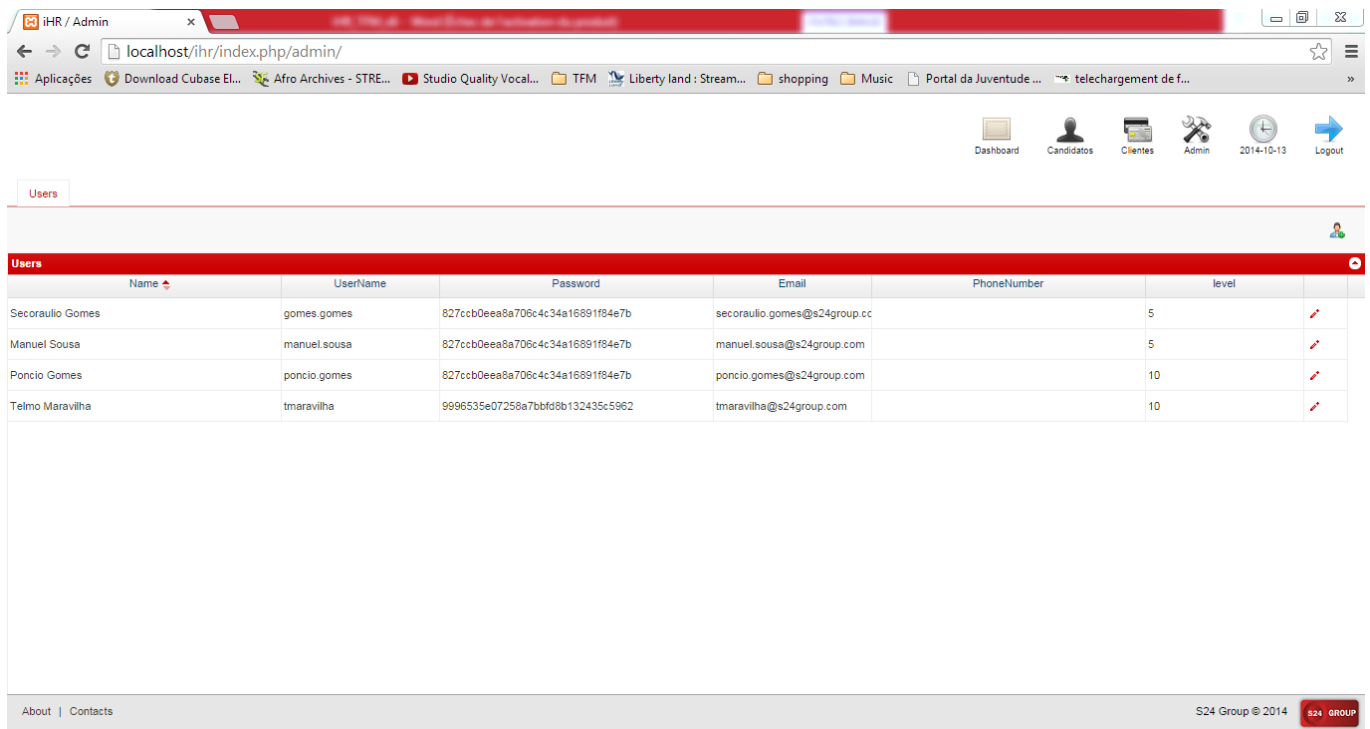
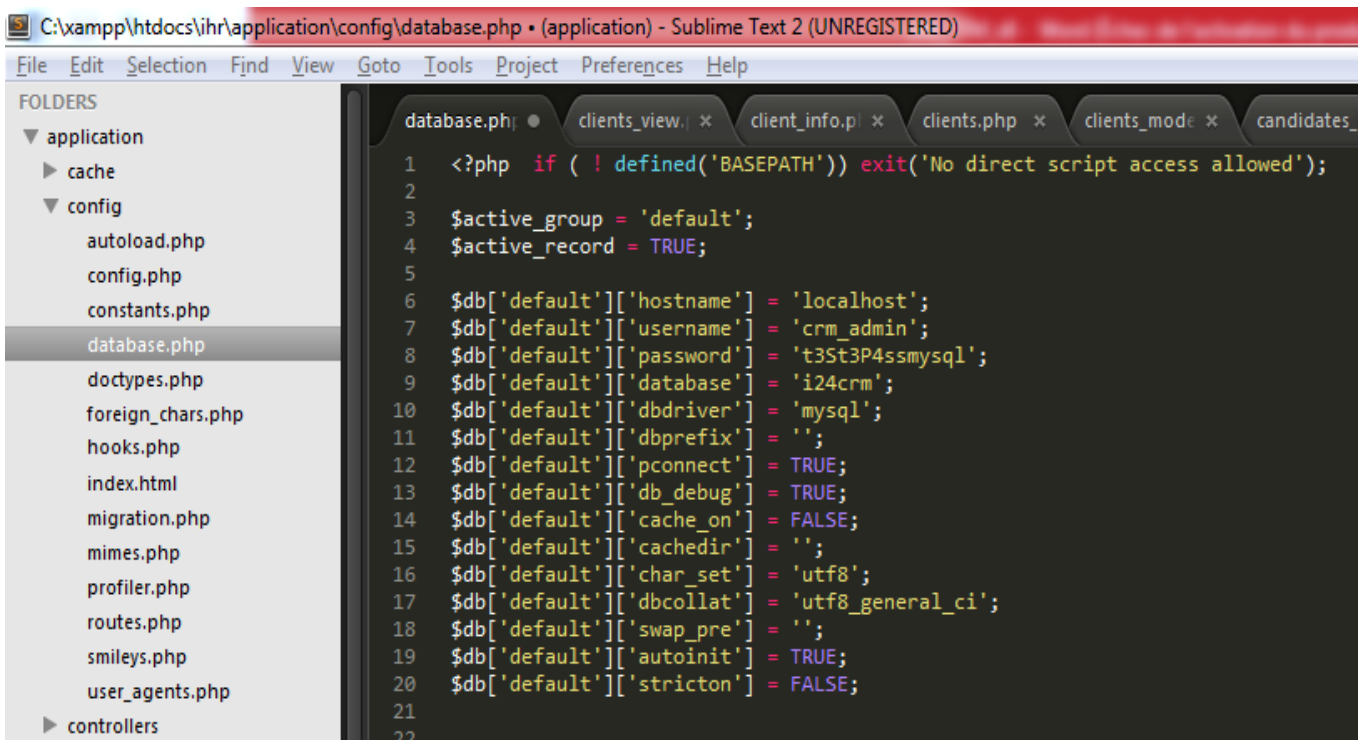


Figura 23 – Teste «Gestão dos utilizadores do sistema»

## 6.2. Fragmentos de código



The image shows a screenshot of a code editor window titled "C:\xampp\htdocs\application\config\database.php • (application) - Sublime Text 2 (UNREGISTERED)". The editor displays the following PHP code:

```
1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3 $active_group = 'default';
4 $active_record = TRUE;
5
6 $db['default']['hostname'] = 'localhost';
7 $db['default']['username'] = 'crm_admin';
8 $db['default']['password'] = 't3St3P4ssmysql';
9 $db['default']['database'] = 'i24crm';
10 $db['default']['dbdriver'] = 'mysql';
11 $db['default']['dbprefix'] = '';
12 $db['default']['pconnect'] = TRUE;
13 $db['default']['db_debug'] = TRUE;
14 $db['default']['cache_on'] = FALSE;
15 $db['default']['cachedir'] = '';
16 $db['default']['char_set'] = 'utf8';
17 $db['default']['dbcollat'] = 'utf8_general_ci';
18 $db['default']['swap_pre'] = '';
19 $db['default']['autoinit'] = TRUE;
20 $db['default']['stricton'] = FALSE;
21
22
```

The left sidebar shows a folder structure with "application" expanded to "config", where "database.php" is selected. Other files in the "config" folder include autoload.php, config.php, constants.php, doctypes.php, foreign\_chars.php, hooks.php, index.html, migration.php, mimes.php, profiler.php, routes.php, smileys.php, and user\_agents.php.

Figura 24 - Configuração da conexão a base de dados «database.php»

```
1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3
4 /*
5 |-----|
6 | Base Site URL
7 |-----|
8 |
9 | URL to your CodeIgniter root. Typically this will be your base URL,
10 | WITH a trailing slash:
11 |
12 | http://example.com/
13 |
14 | If this is not set then CodeIgniter will guess the protocol, domain and
15 | path to your installation.
16 |-----|
17 */
18 $config['base_url'] = 'http://localhost/ihr';
19
20 /*
21 |-----|
22 | Index File
23 |-----|
24 |
25 | Typically this will be your index.php file, unless you've renamed it to
26 | something else. If you are using mod_rewrite to remove the page set this
27 | variable so that it is blank.
28 |-----|
29 */
30 $config['index_page'] = 'index.php';
```

Figura 25 - Configuração do Base Path «config.php»

```

10 class Verify_Login extends CI_Controller {
11
12     /**
13      *
14      * @access public
15      * @param array
16      * @return mixed
17      */
18
19     public function __construct( ) {
20
21         parent::__construct();
22         $this->load->model( 'users_model' );
23     }
24
25     /**
26     *
27     * @access public
28     * @param array
29     * @return mixed
30     */
31
32     public function index() {
33
34         $this->load->library('form_validation');
35
36         $this->form_validation->set_rules('username', 'Username', 'trim|required|xss_clean');
37         $this->form_validation->set_rules('password', 'Password', 'trim|required|xss_clean|callback_check_database');
38
39         if( $this->form_validation->run() == FALSE ){
40
41             $this->load->view('auth/login_view');
42
43         } else {
44             redirect('home', 'refresh');
45         }
46     }
47

```

Figura 26 - Controller «verify\_login.php»

```

/**
 *
 * Add novo candidato
 *
 */
public function create_profile($pd_nameTXT,$pd_dobTXT,$pd_nationalityTXT,$pd_mobile1TXT,$pd_mobile2TXT,
    $pd_emailTXT,$pd_addressTXT,$pd_current_statusTXT,$pd_bandTXT,$pd_notes,$id_campanha_selected,$user_test){

    $data = array(
        'name' => $pd_nameTXT,
        'date_of_birth' => $pd_dobTXT,
        'mobile1' =>$pd_mobile1TXT,
        'mobile2' => $pd_mobile2TXT,
        'email' => $pd_emailTXT,
        'nationality' => $pd_nationalityTXT,
        'address' => $pd_addressTXT,
        'current_status' => $pd_current_statusTXT,
        'band' => $pd_bandTXT,
        'notes'=>$pd_notes,
        'id_campanha' =>$id_campanha_selected,
        'created_by'=>$user_test
    );

    if ($pd_emailTXT!="" || $pd_mobile1TXT!="" || $pd_mobile2TXT!="") {
        $sql = "SELECT * FROM candidate WHERE name='$pd_nameTXT' AND email='$pd_emailTXT' AND nationality='$pd_nationalityTXT'";
        $rs = $this->db->query( $sql );
        $rsNumRows = $rs->num_rows();
        if ($rsNumRows==0) {
            $this->db->insert( 'candidate', $data );
            return $this->db->insert_id();
        }elseif ($rsNumRows==1) {
            return "exist";
        }
    }
}
}

```

Figura 27 - Model «Novo candidato» Candidates\_model.php