



# **Instituto Politécnico de Bragança**

Escola Superior de tecnologia e gestão

## **Estágio curricular em Mestrado Sistemas Informação**

Leandro Ismael Pereira Alexandre

2015/2016



# **Instituto politécnico de Bragança**

Escola Superior de tecnologia e gestão

## **Estágio curricular em Mestrado Sistemas Informação**

Projeto orientado pelo Professor João Paulo Pereira e José Adriano Pires

2015/2016

Certifico que li este relatório e que na minha opinião, é adequado no seu conteúdo e forma como demonstrador do trabalho desenvolvido no âmbito da UC de Projeto.

---

José Paulo Ribeiro Pereira (Orientador)

Certifico que li este relatório e que na minha opinião, é adequado no seu conteúdo e forma como demonstrador do trabalho desenvolvido no âmbito da UC de Projeto.

---

José Adriano Gomes Pires (Co-Orientador)

Certifico que li este relatório e que na minha opinião, é adequado no seu conteúdo e forma como demonstrador do trabalho desenvolvido no âmbito da UC de Projeto.

---

Nome Arguente (Arguente)

## Agradecimentos

O desempenho no desenvolvimento deste projeto de estágio não teria sido o mesmo sem a ajuda e apoio de algumas pessoas, às quais gostaria de aqui expressar o meu reconhecimento.

Assim, começaria por agradecer ao orientador Professor João Paulo Pereira, pelo seu apoio e acompanhamento.

Um agradecimento especial ao Professor José Adriano Pires pelo apoio, liderança e pela forma como me integrou na equipa da empresa JAGP, Lda. Um agradecimento a todos os colegas da equipa JAGP Lda. com a qual tive o privilégio de trabalhar durante os meses de estágio: David Dias e Micael Vinhas pela disponibilidade e companheirismo demonstrados ao longo do projeto.

À minha família, em especial à minha mãe, pai e irmã, pela forma como sempre me apoiaram e ajudaram a ultrapassar as barreiras que foram surgindo.

Finalmente, um agradecimento muito especial à minha namorada pela compreensão, carinho, paciência e incondicional apoio que sempre demonstrou. A todos os que estão e não estão aqui indicados, o meu obrigado.

## Resumo

A Tecnologia da Informação (TI) pode ser definida como o conjunto de todas as atividades e soluções providas por recursos computacionais que visam permitir a obtenção, o armazenamento, o acesso, a gestão e o uso de informação. Na verdade, as aplicações para TI são tantas - estão ligadas às mais diversas áreas - que há várias definições para a expressão e nenhuma delas consegue determiná-la por completo.

Sendo a informação um patrimônio, um bem que agrega valor e dá sentido às atividades que a utilizam, é necessário fazer uso de recursos de TI de maneira apropriada, ou seja, é preciso utilizar ferramentas, sistemas ou outros meios que façam das informações um diferencial. Além disso, é importante a procura de novas soluções que tragam resultados realmente relevantes, isto é, que permitam transformar as informações em algo com valor maior, sem deixar de considerar o aspeto do menor custo possível [1].

Este documento descreve o trabalho realizado no âmbito da disciplina Projeto em Mestrado de Sistemas de informação do Instituto Politécnico de Bragança.

O trabalho consistiu na realização de um estágio curricular em ambiente de empresa, assim foi proposto ao estagiário o desenvolvimento de uma plataforma de emprego para suporte das autarquias e instituições de ensino onde se podem destacar como principais funcionalidades a criação de ofertas por partes das empresas registadas e o utilizador comum pode efetuar candidaturas a essas mesmas ofertas. À frente será pormenorizado os fundamentos e os objetivos da plataforma a ser desenvolvida de modo a facilitar a perceção e compreensão deste documento e todas as funcionalidades inerentes desenvolvidas.

## Abstract

The Information Technology (IT) can be defined as the set of all activities and solutions provided by computing resources that are designed to enable the collection, storage, access, management and use of information. In fact, applications for IT are many - are linked to several areas - there are several definitions for the term, and none of them can determine it completely.

Because information an asset, an asset that adds value and gives meaning to activities that use it, you need to make use of IT resources properly, ie, you must use tools, systems or other means to make the information a differential . In addition, the search for new solutions that bring truly relevant results is important, that is, allowing transform information into something with greater value while considering the aspect of the lowest possible cost.

This document describes the work carried out under the Project course in Master of Information Systems at the Polytechnic Institute of Bragança.

The work consisted in carrying out a traineeship in business environment, so it was offered to trainee development of an employment platform for support of local authorities and educational institutions where they can stand out as key features to create offers for parts of registered companies and the average user can make applications for these same offers. Ahead will be detailed the basics and the platform aims to be developed in order to facilitate the perception and understanding of this document and all developed inherent features.

# Conteúdo

<b>Lista de Figuras.....</b>	<b>10</b>
<b>Lista de Tabelas.....</b>	<b>11</b>
<b>Introdução .....</b>	<b>12</b>
1.1    Âmbito.....	12
1.2    Instituição de acolhimento .....	12
1.2.1    Descrição da empresa .....	12
1.3    Organização do documento.....	14
<b>Objetivos e contexto .....</b>	<b>15</b>
2.1 Projeto Global .....	15
2.2 Portal do emprego .....	17
<b>Metodologia e calendarização .....</b>	<b>18</b>
3.1 Metodologia .....	18
3.2 Calendarização .....	20
<b>Tecnologias e ferramentas utilizadas .....</b>	<b>22</b>
4.1 PHP.....	22
4.2 Apache.....	24
4.2.1 Características do servidor Apache .....	26
4.3 MySql.....	28
4.4 JavaScript .....	29
4.5 HTML5.....	30
4.6 CSS.....	31
4.7 Yii.....	32
4.7.1 História .....	33
4.7.2 MVC .....	34
4.8 Git.....	37
4.8.1 Gitlab .....	38
4.9 IDE .....	38
4.9.1 Eclipse .....	40
4.10 Wiki.....	41
4.11 Inkscape.....	41

<b>Trabalhos realizados.....</b>	<b>43</b>
5.1 Plataforma “e-emprego”.....	43
5.2 Configuração do Ambiente de Desenvolvimento .....	45
5.3 Criação da base de dados .....	47
5.4 Instalação da framework yii2 .....	47
5.4.1 Repositório git .....	49
5.4.2 Usermanagement yii2.....	50
5.4.3 Criação CRUD.....	55
5.4.4 Implementação de funcionalidades e demonstração do trabalho .....	57
5.5 Outras tarefas.....	64
<b>Conclusão e trabalho futuro .....</b>	<b>65</b>
<b>Referências.....</b>	<b>67</b>
<b>Anexos .....</b>	<b>68</b>
Estrutura da base de dados .....	68

# Lista de Figuras

Figura 1-Logo JAGP Lda.....	13
Figura 2 - e-serviços.....	16
Figura 3 - Logo e-emprego .....	17
Figura 4- Modelo em cascata.....	18
Figura 5 - Logo open-source.....	22
Figura 6 - PHP.....	24
Figura 7 - Servidor Apache.....	27
Figura 8 - Mysql.....	29
Figura 9 - JavaScript .....	29
Figura 10 – Html.....	30
Figura 11 - Css .....	32
Figura 12 - Yii2.....	33
Figura 13- MVC.....	34
Figura 14 - Estrutura estática aplicação yii.....	35
Figura 15 - Fluxo de execução yii.....	36
Figura 16 - Git.....	38
Figura 17 - Gitlab.....	38
Figura 18 - Eclipse .....	41
Figura 19 - Inkscape.....	42
Figura 20 - Instalação da Yii.....	48
Figura 21 - Acesso à página models e CRUD generator .....	55
Figura 22 - Model generator .....	56
Figura 23 - CRUD.....	57
Figura 24 - Página principal da plataforma .....	58
Figura 25 - Registo de empresa .....	59
Figura 26 - Perfil de empresa.....	60
Figura 27 - Primeiro passo criação de oferta .....	61
Figura 28 - Oferta detalhada .....	62
Figura 29 - Áreas de interesse do utilizador .....	63
Figura 30 - Design Plataforma e-serviços.....	64

## **Lista de Tabelas**

Tabela 1-Calendarização.....	20
------------------------------	----

# **Capítulo 1**

## **Introdução**

### **1.1 Âmbito**

O projeto de estágio desenvolvido pelo aluno, Leandro Ismael Pereira Alexandre, na instituição, JAGP Lda., enquadra-se no âmbito da disciplina de Projeto Mestrado de Sistemas de Informação (MSI) da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Bragança.

### **1.2 Instituição de acolhimento**

#### **1.2.1 Descrição da empresa**

A empresa JAGP Lda é uma start-Up do Instituto Politécnico de Bragança que foi criada em Janeiro de 2013 por dois docentes e dois alunos de mestrado da Escola Superior de Tecnologia e Gestão de Bragança.

Na sua fase de arranque a empresa estava vocacionada para o comércio eletrónico de produtos tradicionais da região de Trás-os-Montes e Alto Douro, tendo por base a exploração da marca comercial RuralNet.

A par da vertente do comércio eletrónico a empresa assegurava as tarefas técnicas relacionadas com a gestão e manutenção da plataforma de vendas da Ruralnet que podia ser acedida no endereço web [www.ruralnet.pt](http://www.ruralnet.pt).

Em Outubro de 2014, a empresa JAGP, Lda vendeu todos os ativos relacionados com a marca comercial RuralNet, passando a direcionar a sua atividade, exclusivamente para o desenvolvimento de software.

Do portefólio de produtos da empresa constam trabalhos desenvolvidos para o Núcleo Empresarial da Região de Bragança, Portugal Foods, Old Care e para o consórcio de Universidades Europeias que gere o Projecto INTACT.



*Figura 1-Logo JAGP Lda.*

De momento a empresa continua a dar suporte os projetos já desenvolvidos bem como encontra-se a desenvolver novas funcionalidades para os mesmos, consoante a necessidades dos clientes, mas não deixa de procurar novas ocasiões de negócio pois o mercado atual é exigente e existe um leque enorme de oportunidades que podem ser aproveitadas ao apresentar um produto com qualidade e que satisfaça os requisitos pretendidos.

### 1.3 Organização do documento

Este documento está organizado da seguinte forma:

- Capítulo 2 – Objetivos e contexto – apresentação dos objetivos do projeto e dos objetivos do estágio no contexto da instituição. Será apresentado neste capítulo todo o contexto do projeto, às suas origens, o porquê do projeto e o potencial.
- Capítulo 3 – Metodologia e calendarização – apresenta as metodologias usadas no desenvolvimento do trabalho realizado e a sua calendarização. Num desenvolvimento de um projeto é necessária uma metodologia de trabalho bem como uma divisão de tarefas ao longo do tempo, embora a questão temporal seja sempre bastante volátil, assim neste capítulo irá ser descrito estas duas questões fundamentais.
- Capítulo 4 – Tecnologias e ferramentas – apresentação sucinta tecnologias e ferramentas utilizadas. As tecnologias e ferramentas representam o núcleo duro do desenvolvimento de software e é necessário ter um conhecimento aprofundado das mesmas e é isso que será apresentado neste capítulo.
- Capítulo 5 – Trabalho realizado – descrição detalhada do trabalho realizado e das tecnologias utilizadas durante o estágio. O trabalho desenvolvido pelo estagiário será apresentado de uma forma global de modo a obter-se uma visão geral do trabalho desenvolvido ao longo do estágio curricular.
- Capítulo 6 – Sumários e conclusões – considerações sobre o trabalho desenvolvido e possível trabalho futuro. Após o trabalho desenvolvido é necessário fazer um sumário e tirar conclusões pois deve-se olhar aquilo que correu bem e aquilo que correu menos bem e pode ser melhorado.

# Capítulo 2

## Objetivos e contexto

### 2.1 Projeto Global

O trabalho de estágio desenvolvido pelo estagiário na empresa inseriu-se num projeto chamado “**e-serviços**”.

O projeto “**e-serviços**” consiste num conjunto de plataformas de apoio a um aglomerado de municípios de modo a dinamizar e a centralizar a informação regional. De entre o conjunto de plataformas pode-se destacar o portal da cultura, das bibliotecas ou ainda do comércio.

Assim os principais objetivos do “**e-serviços**” são:

- Informação centralizada;

Com o conjunto de plataformas será possível centralizar a informação de uma região num único sítio. Por exemplo na plataforma de comércio toda a região poderá aceder numa única plataforma aos produtos regionais disponíveis para venda isto não só traz divulgação aos produtos como dinamiza a venda de produtos regionais.

- Cooperação entre municípios;

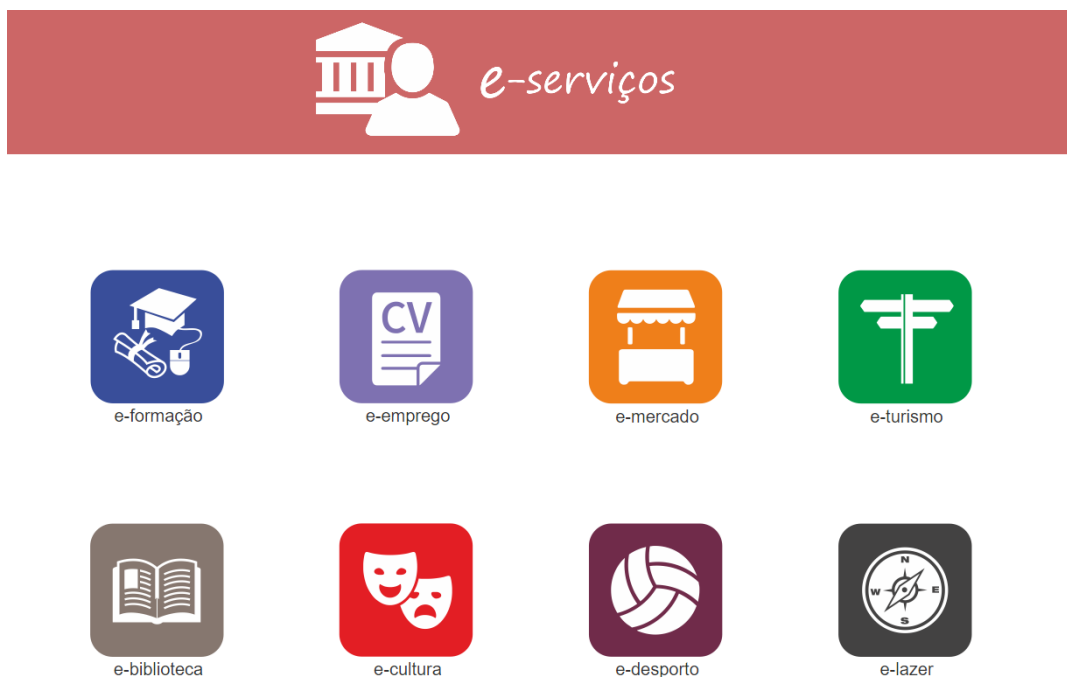
A questão da informação centralizada obriga os municípios a cooperar entre si para dinamizar as plataformas pertencentes ao “**e-serviços**” o que pressupõe ainda uma maior vantagem para todos os municípios.

- Criação de emprego;

A criação de emprego representa uma enorme fragilidade a nível nacional, maior parte das ofertas de emprego que surgem estão aliadas a grandes cidades e maioritariamente associadas ao litoral do país, assim surgiu a ideia da criação de um portal de emprego (a ser desenvolvido pelo estagiário) que possa ser usado pelos municípios bem como pelas instituições de ensino locais de modo a criar uma rede de contactos regional e dinamizar o emprego local. Como já foi referido esta plataforma irá ser desenvolvida pelo estagiário e é com base nestes pressupostos que a plataforma irá assentar.

- Divulgação de eventos cultura, desporto, lazer.

Outro aspeto importante é a divulgação tanto de eventos de cultura, de desporto ou de lazer numa dada área o que permite ainda mais reforçar laços numa região.



*Figura 2 - e-serviços*

## 2.2 Portal do emprego

Inserido no projeto “e-serviços” foi proposto ao estagiário que desenvolve-se uma plataforma de emprego para ser inserido no projeto já referido. Assim sendo este portal de emprego tem como principal objetivo a divulgação de propostas de emprego regionais para que estas possam chegar ao maior número de pessoas na região inserida. Assim sendo podemos destacar os principais objetivos do portal:

- Dinamizar o emprego regional;

O principal objetivo desta plataforma, como é obvio é dinamizar o emprego regional, para isso é necessário criar parcerias e divulgar e dinamizar a plataforma junto do público-alvo.

- Incentivar a criação de empresas;

Para existirem ofertas de emprego é necessário que existem empresas ou entidades que tenham propostas de trabalho para oferecer. Assim sendo este portal também como objetivo incentivar a criação de empresas.

- Cooperação com instituições de ensino locais.

Outro aspeto muito importante é a cooperação com instituições de ensino locais quer seja do ensino superior quer seja do secundário, este público-alvo pode desencadear interesse junto das empresas locais.



*Figura 3 - Logo e-emprego*

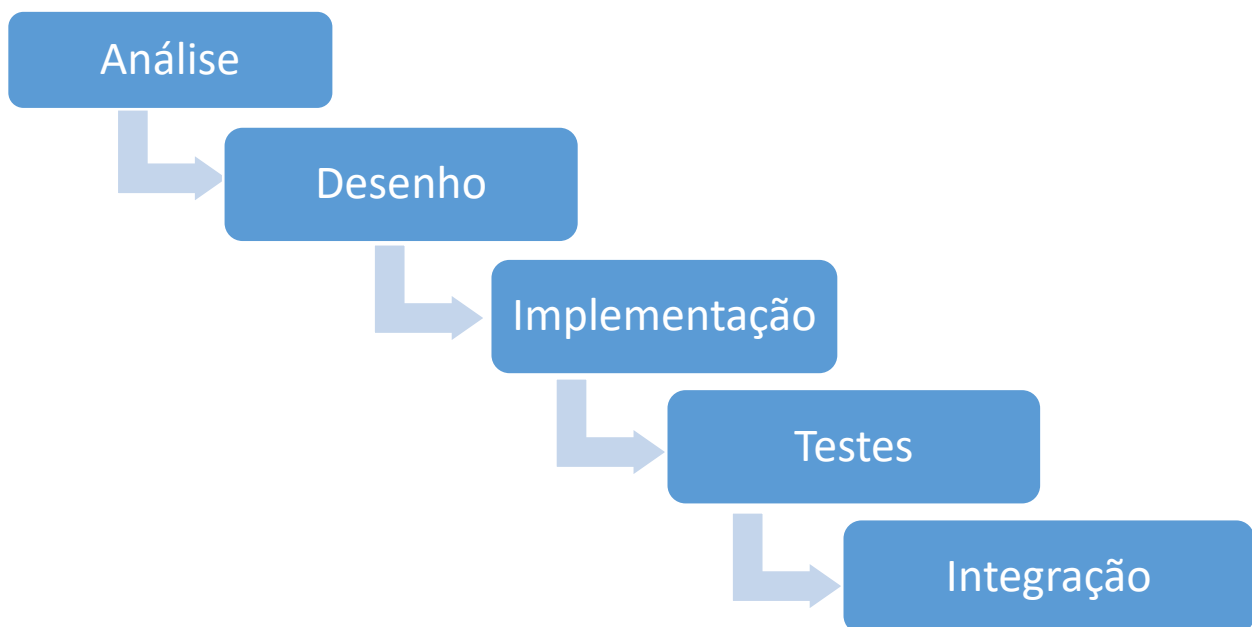
# Capítulo 3

## Metodologia e calendarização

### 3.1 Metodologia

A utilização de uma metodologia de desenvolvimento, também denominado de processo de software, é um fator primordial para o sucesso das empresas de desenvolvimento de software. Um processo de software pode ser entendido como um conjunto estruturado de atividades para desenvolver um sistema de software.

A metodologia de desenvolvimento adotada, na execução das tarefas que envolveram o desenvolvimento deste trabalho de estágio, foi a do modelo em cascata. Este processo é constituído por cinco etapas, ver Figura 4: análise; desenho; implementação; testes; e integração, sendo gerada no final de cada etapa a documentação necessária.



*Figura 4- Modelo em cascata*

A fase de análise é caracterizada pela identificação dos requisitos da aplicação, o que consiste usualmente nos serviços que se devem fornecer, limitações e objetivos do software. Na fase de desenho são desenvolvidos modelos conceptuais e é analisada a melhor solução de implementação dos componentes da aplicação, tomando-se decisões fundamentais na conceção de novos componentes ou das alterações a fazer aos componentes já existentes.

A fase de implementação consiste na codificação da aplicação numa linguagem de programação. Na fase de testes são executados os planos de teste definidos para a aplicação, de forma a solucionar erros de implementação e assegurar que são produzidos resultados reais que coincidam com os requisitos especificados. Nesta fase é produzida toda a documentação relativa à aplicação.

Finalmente, a última fase consiste na integração da aplicação no sistema, ou seja, corresponde à entrega do produto. O modelo em cascata segue uma abordagem top-down e tem a vantagem que só avança para a tarefa seguinte quando o cliente valida e aceita os produtos finais da tarefa atual. Contudo, apresenta como desvantagens a dificuldade em responder a mudanças nos requisitos depois do processo se ter iniciado e o alto custo de correção das especificações quando nas fases de teste e integração. Assim como este modelo de desenvolvimento é necessário que cada passo seja dado firmemente, ou seja, cada fase de desenvolvimento deve ser bem pensada e trabalhada antes de avançar para o seguinte passo. Assim este projeto foi sempre desenvolvido com esta questão em mente de modo a desenvolver um produto com qualidade e acima de tudo com funcionalidades consolidadas.

### 3.2 Calendarização

No início do estágio foi elaborado em reunião com o orientador e com o supervisor, o seguinte plano de orientação de trabalho:

Tarefas		Duração
<b>Integração na equipa de trabalho</b>		1 semana
<b>Projeto – Portal Emprego</b>	Instalação das ferramentas necessárias	1 semana
	Desenho base de dados e testes	1 semana
	Criação de CRUD e User management	1 semana
	Implementação das funcionalidades	2 meses
	Testes à plataforma e integração	4 semanas
<b>Tarefas</b>	Suporte e manutenção	Sem data
	Design	Sem data

*Tabela 1-Calendarização*

Durante o período de integração e adaptação à equipa, o estagiário teve a oportunidade de realizar as seguintes tarefas:

- Adaptação com as tecnologias e ferramentas a usar;

Uma fase muito importante durante o estágio foi a adaptação as tecnologias a utilizar no desenvolvimento pois com algumas delas teve-se o primeiro contacto.

- Leitura de documentação necessária para o desenvolvimento do projeto;

Para um melhor entendimento de algumas tecnologias foi necessária a leitura de alguma documentação necessária para uma melhor compreensão e para uma

melhor inserção na equipa de trabalho, a documentação em relação a *framework yii2* que irá ser apresentada mais à frente foi uma das mais estudadas.

- Familiarização com o conceito de trabalho cooperativo;

O conceito de trabalho cooperativo cada vez mais é importante no seio das empresas. A principal característica que define um trabalho em equipa é, sem dúvida, a existência de objetivos comuns, coletivos, a serem alcançados através do desempenho de todos os participantes. O esforço conjunto de todos os membros é indispensável para a consecução desses objetivos. Num trabalho em equipa, também, todos os membros desejam aprender ou melhorar os processos de todos os trabalhos do grupo, independentemente de sua função individual. Todos sabem fazer e conhecem a fundo todos os trabalhos da equipa. Basicamente, sempre que houver um objetivo comum que dependa de todos para ser alcançado, esse grupo torna-se numa equipa. O trabalho cooperativo surgiu da necessidade humana de unir esforços para atingir objetivos que seriam mais difíceis ou impossíveis de serem alcançados individualmente. Essa estratégia de ação vem sendo estimulada em todas as atividades humanas, para melhorar a efetividade do trabalho e o nível de satisfação dos trabalhadores.

- Adaptação as metodologias de trabalho.

Cada equipa de trabalho adota uma metodologia de trabalho onde é necessário perceber como deve ser efetuado cada passo do desenvolvimento de um projeto.

# Capítulo 4

## Tecnologias e ferramentas utilizadas

Neste capítulo irão ser descritas detalhadamente as tecnologias e ferramentas utilizadas ao longo de todo o estágio. De salientar que todas as tecnologias e ferramentas utilizadas são open-source.

O open-source significa Software de código aberto (do inglês open source software ou OSS) é o software de computador com o seu código fonte disponibilizado e licenciado com uma licença de código aberto no qual o direito autoral fornece o direito de estudar, modificar e distribuir o software de graça para qualquer um e para qualquer finalidade [2].



*Figura 5 - Logo open-source*

### 4.1 PHP

O **PHP** foi a principal linguagem utilizada ao longo do desenvolvimento de toda a plataforma.

O PHP (um acrônimo recursivo para PHP: Hypertext Preprocessor) é uma linguagem de script open-source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML.

O que distingue o PHP de algo como o Javascript no lado do cliente é que o código é executado no servidor, gerando o HTML que é então enviado para o browser. O

PHP é focado principalmente nos scripts do lado do servidor, portanto, pode fazer qualquer coisa que outro programa CGI pode fazer: receber dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber cookies [3].

Alguns dos maiores campos onde os scripts PHP podem ser utilizados:

- Scripts no lado do servidor (server-side). Este é o mais tradicional e principal campo de atuação do PHP. Existem três ferramentas para correr o PHP. O interpretador do PHP (CGI ou módulo do servidor), um servidor web e um navegador web.
- Scripts de linha de comando. Pode-se executar um script PHP sem um servidor ou navegador, a única coisa necessária é o interpretador PHP. Esse tipo de uso é ideal para script executados usando o cron (Unix, Linux) ou o gestor de Tarefas (no Windows). Esses scripts podem ser usados também para rotinas de processamento de texto simples.

O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes do Unix (como HP-UX, Solaris e OpenBSD), Microsoft Windows, Mac OS X, RISC OS e provavelmente outros. O PHP também é suportado pela maioria dos servidores web atualmente. Isso inclui o Apache, o IIS e muitos outros. Também qualquer servidor web que pode utilizar o binário FastCGI do PHP, como o lighttpd e o nginx. O PHP trabalha tanto como módulo quanto como um processador CGI.

Com o PHP, portanto, existe uma liberdade de escolha tanto do sistema operacional como do servidor web. Do mesmo modo, pode-se optar por uma programação estruturada ou programação orientada a objeto (POO), ou ainda uma mistura das duas.

Uma das características mais fortes e mais significativas do PHP é o seu suporte a uma ampla variedade de base de dados. Criar uma página web para consultar uma base de dados é incrivelmente simples usando uma das extensões específicas da base de dados (e.g., mysql), ou usando uma camada de abstração como o PDO ou conectar a qualquer base de dados que suporte o padrão "Open Database Connection" usando a extensão ODBC.

O PHP também tem suporte para comunicação com outros serviços utilizando protocolos como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (em Windows) e incontáveis outros [4].



*Figura 6 - PHP*

## **4.2 Apache**

O Apache foi o servidor web escolhido para todo o desenvolvimento da plataforma pois fornece o conjunto de serviços necessários para o desenvolvimento e além disso é o servidor web mais usado do mundo.

Ao acessar qualquer site, há um servidor por trás de um determinado endereço responsável por disponibilizar as páginas e todos os demais recursos. Assim, por exemplo, quando se envia um e-mail através de um formulário, coloca-se uma mensagem num fórum de discussão, faz-se uma compra on-line, etc., um servidor Web (ou um conjunto de servidores) é responsável por processar todas essas informações.

Portanto, um servidor Web é um computador que processa solicitações HTTP (Hyper-Text Transfer Protocol), o protocolo padrão da Web. Ao aceder um navegador de internet para aceder a um site, este faz as solicitações devidas ao servidor Web do site através de HTTP e então recebe o conteúdo correspondente. No caso do Apache, ele não só executa o HTTP, como outros protocolos, tais como o HTTPS (O HTTP combinado com a camada de segurança **SSL - Secure Socket Layer**), o **FTP (File Transfer Protocol)**, entre outros.

Como servidor Web, o Apache é o mais conhecido e usado. Os motivos incluem a sua excelente performance, segurança, compatibilidade com diversas plataformas e todos os seus recursos.

O servidor Apache (ou Apache Server) surgiu no **National Center of Supercomputing Applications (NCSA)** através do trabalho de *Rob McCool*. Ao sair da **NCSA**, *McCool* parou de trabalhar no software (que nessa época recebia justamente a denominação **NCSA**) e então várias pessoas e grupos passaram a adaptar o servidor Web às suas necessidades. No entanto, foram *Brian Behlendorf* e *Cliff Skolnick* os principais responsáveis pela retomada do projeto, contando logo em seguida com o apoio de *Brandon Long* e *Beth Frank*. Estes últimos tinham a tarefa de continuar com o desenvolvimento do servidor, mas pela **NCSA**. Não demorou muito para que eles se juntassem ao **Apache Group**.

A origem da Fundação Apache (Apache Foundation) vem daí: *Brian Behlendorf* e *Cliff Skolnick* passaram a controlar "patches" (adição de recursos ou correções) atrás de uma lista de discussão. Dessa lista foi formado o grupo, constituído inicialmente por 8 pessoas. O nome da fundação parece ter sido baseado nessa característica (uso de patches), já que pode ser interpretado como um trocadilho com a expressão em inglês "a patchy". No entanto, segundo os desenvolvedores do servidor, o nome é baseado numa tribo americana de nome Apache.

O servidor Apache teve sua primeira versão oficial (0.6.2) lançada em 1995. De 1995 até hoje houve várias mudanças no projeto. Talvez, a mais importante tenha sido o *Shambhala*, uma arquitetura para o Apache desenvolvida por *Robert Thau*. O **Shambhala** agradou tanto na época (principalmente pela melhor gestão de memória) que acabou a ser incorporado definitivamente no servidor, tirando de cena a base existente até então. Surgiu a versão 0.8.8, que tinha, além do **Shambhala**, recursos da versão 0.7.x. Após uma série de aperfeiçoamentos, testes e de uma documentação nova (feita por *David Robinson*), o Apache 1.0 foi lançado em dezembro de 1995 [5].

#### 4.2.1 Características do servidor Apache

O Apache Server é um software livre, o que significa, como já foi referido a cima, que qualquer um pode estudar ou alterar seu código-fonte, além de poder utilizá-lo gratuitamente. É graças a essa característica que o software é melhorado ao passar dos anos. Graças ao trabalho muitas vezes voluntário de vários desenvolvedores.

Além de estar disponível para o Linux (e para outros sistemas operacionais baseados no Unix), o Apache também conta com versões para o Windows. O servidor Apache é capaz de executar código em PHP, Perl, Shell Script e até em ASP e pode atuar como servidor FTP, HTTP, entre outros. A sua utilização mais conhecida é a que combina o Apache com a linguagem PHP e a base de dados MySQL.

De seguida é apresentado um resumo com as principais características:

- Possui suporte a scripts cgi usando linguagens como Perl, PHP, Shell Script, ASP, etc;

- Suporte a autorização de acesso podendo ser especificadas restrições de acesso separadamente para cada endereço/arquivo/diretório consultado no servidor;
- Autenticação requerendo um nome de utilizador e password válidos para acesso a alguma página/subdiretória/arquivo (suportando criptografia via Crypto e MD5);
- Negociação de conteúdo, permitindo a exibição da página Web no idioma requisitado pelo Cliente Navegador;
- Personalização de logs;
- Mensagens de erro;
- Suporte a virtual hosting (é possível servir 2 ou mais páginas com endereços/portas diferentes através do mesmo processo ou usar mais de um processo para controlar mais de um endereço);
- Suporte a IP virtual hosting;
- Suporte a name virtual hosting;
- Suporte a servidor Proxy ftp e http, com limite de acesso, caching (todas flexivelmente configuráveis);
- Suporte a proxy e redireccionamentos baseados em URLs para endereços Internos;
- Suporte a criptografia via SSL, Certificados digitais;
- Módulos DSO (Dynamic Shared Objects) permitem adicionar/remover funcionalidades e recursos sem necessidade de recompilação do programa [6].



*Figura 7 - Servidor Apache*

## 4.3 MySQL

O **MySQL** é um sistema de gestão de base de dados (**SGBD**), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) como interface. É atualmente uma das bases de dados mais populares, com mais de 10 milhões de instalações pelo mundo inteiro.

O MySQL é um dos sistemas de gestão de base de dados mais populares que existe e, por ser otimizado para aplicações Web, é amplamente utilizado na internet. É muito comum encontrar serviços de hospedagem de sites que oferecem o MySQL e a linguagem PHP, justamente porque ambos trabalham muito bem em conjunto.

Outro fator que ajuda na popularidade do MySQL é sua disponibilidade para praticamente qualquer sistema operacional, como Linux, FreeBSD (e outros sistemas baseados em Unix), Windows e Mac OS X. Além disso, o MySQL é um software livre (sob licença GPL), o que significa que qualquer um pode estudá-lo ou alterá-lo conforme a necessidade.

Entre as características técnicas do SGBD MySQL, estão:

- Alta compatibilidade com linguagens como PHP, Java, Python, C#, Ruby e C/C++;
- Baixa exigência de processamento (em comparação como outros SGBD);
- Vários sistemas de armazenamento de dados (database engine), como MyISAM, MySQL Cluster, CSV, Merge, InnoDB, entre outros;
- Recursos como transactions (transações), conectividade segura, indexação de campos de texto, replicação, etc;

- Instruções em SQL [7].



*Figura 8 - Mysql*

#### **4.4 JavaScript**

JavaScript é uma linguagem de programação interpretada. Foi originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente e interagissem com o utilizador sem a necessidade deste script passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido.

É atualmente a principal linguagem para programação client-side em navegadores web. Começa também a ser bastante utilizada do lado do servidor através de ambientes como o node.js. Foi concebida para ser uma linguagem script com orientação a objetos baseada em protótipos e funções de primeira classe. Possui suporte à programação funcional e apresenta recursos como funções de alta ordem comumente indisponíveis em linguagens populares como Java e C++ [8].



*Figura 9 - JavaScript*

## 4.5 HTML5

*HTML* é a sigla de HyperText Markup Language, expressão inglesa que significa "Linguagem de Marcação de Hipertexto". Consiste numa linguagem de marcação utilizada para produção de páginas web, que permite a criação de documentos que podem ser lidos em praticamente qualquer tipo de computador e transmitidos pela Internet. Para escrever documentos HTML não é necessário mais do que um editor de texto simples e conhecimento dos códigos que compõem a linguagem. Os códigos (conhecidos como *tags*) servem para indicar a função de cada elemento da página Web. Os *tags* funcionam como comandos de formatação de textos, formulários, links (ligações), imagens, tabelas, entre outros. Os browsers (navegadores) identificam as *tags* e apresentam a página conforme está especificada. A linguagem HTML foi criada por **Tim Barners Lee** na década de 1990. As especificações da linguagem são controladas pela W3C (World Wide Web Consortium). As versões de evolução da HTML incluem XHTML (uma linguagem com sintaxe mais rigorosa, baseada em XML) e HTML5 (quinta versão da HTML que traz novos recursos, principalmente a manipulação de conteúdo gráfico e de multimédia) [9].



*Figura 10 – Html*

## 4.6 CSS

O *cascading Style Sheets* (CSS) é uma "folha de estilo" composta por "camadas" e utilizada para definir a apresentação (aparência) das páginas da internet que adotam para o desenvolvimento linguagens de marcação como XML, HTML e XHTML. O CSS define como serão exibidos os elementos contidos no código de uma página de internet e sua maior vantagem é efetuar a separação entre o formato e o conteúdo de um documento.

Com a evolução dos recursos de programação as páginas da internet estavam a adotar cada vez mais estilos e variações para deixá-las mais elegantes e atrativas para os utilizadores. Com isto, linguagens de marcação simples como o HTML, que era destinada para apresentar os conteúdos também precisaram de ser aprimoradas.

Foram criadas novas *tags* e atributos de estilo para o HTML e em resumo ele passou a exercer tanto a função de estruturar o conteúdo quanto de apresentá-lo para o utilizador final. Entretanto, isto começou a trazer um problema para os desenvolvedores, pois não havia uma forma de definir, por exemplo, um padrão para todos os cabeçalhos ou conteúdos em diversas páginas. Ou seja, as alterações teriam que ser feitas manualmente, uma a uma.

A partir destas complicações, nasceu o CSS. Primariamente, foi desenvolvido para habilitar a separação do conteúdo e formato de um documento (na linguagem de formatação utilizada) da apresentação, incluindo elementos como cores, formatos de fontes e layout. Esta separação proporcionou uma maior flexibilidade e controlo na especificação de como as características serão exibidas, permitiu uma partilha de formato e reduziu a repetição no conteúdo estrutural de uma página.

Com isto, as linguagens de marcação passaram novamente a exercer a função de marcar e estruturar o conteúdo de uma página enquanto o CSS encarregou-se da

aplicação dos estilos necessários para a aparência dela. Isto é feito por meio da criação de um arquivo externo que contém todas as regras aplicadas e, com isto, é possível fazer alterações de estilo em todas as páginas de um site de forma fácil e rápida.

O CSS também permite que as mesmas marcações de uma página sejam apresentadas em diferentes estilos, conforme os métodos de renderização. A maioria dos menus em cascata, estilos de cabeçalho e rodapé de páginas da internet, por exemplo, atualmente são desenvolvidos em CSS [10].



*Figura 11 - Css*

## **4.7 Yii**

Yii é um framework de alta performance em PHP que utiliza componentes para o desenvolvimento de grandes aplicações Web. Yii permite a máxima reutilização de códigos na programação Web e pode acelerar significativamente o processo de desenvolvimento. O nome Yii (pronunciado gi) representa as palavras fácil (easy), eficiente (efficient) e extensível (extensible).

O Yii destaca-se de outras frameworks PHP na medida em que é eficiente, rica em recursos e bem documentada. A Yii é cuidadosamente projetada para se ajustar a aplicações Web desde o seu início [11].

Outra enorme vantagem da yii é a existência de imensas extensões e widgets que podem ser instaladas separadamente que facilitam ainda mais o desenvolvimento através desta ferramenta.



*Figura 12 - Yii2*

#### **4.7.1 História**

A Yii foi criada por **Qiang Xue**, que iniciou este projeto a 1 de janeiro de 2008. Antes deste projeto, **Qiang Xue** já havia trabalhado num outro, **Prado**, uma framework não tão potente, mas com o mesmo fundamento da Yii. Os seus anos de experiência a programar permitiram com que fosse capaz de desenvolver uma framework de raiz capaz de ultrapassar os maiores obstáculos que enfrentou no passado. A 3 de dezembro de 2008 lançou a Yii 1.0. A sua performance impressionante, quando comparada com outras frameworks PHP, garantiram imediatamente atenções por parte de programadores web, e a sua popularidade nunca parou de crescer desde então.

Em outubro de 2014 saiu a versão 2.0.0, que foi, de resto, uma alteração profunda a primeira versão de forma a garantir que a Yii se manteria como uma framework ideal para quem quer utilizar todas as funcionalidades das versões mais recentes do PHP [11].

## 4.7.2 MVC

A Yii implementa o padrão de desenvolvimento model-view-controller (MVC) que é amplamente adotado na programação Web. O MVC visa separar a lógica de interação da interface com o utilizador, assim os programadores podem mudar facilmente cada parte, sem afetar as outras. No padrão MVC, o **model** representa as informações (os dados) e as regras de interação, a **view** contém elemento de interface com o utilizador, como textos, formulários, e o **controller** gere a comunicação entre o **model** e a **view**.

Além de MVC, a Yii também introduz um controlo de frente, chamado aplicação (application), que representa o contexto de execução dos processos requisitados. A aplicação recebe a solicitação do utilizador e a envia para um **controller** adequado para ser processada.

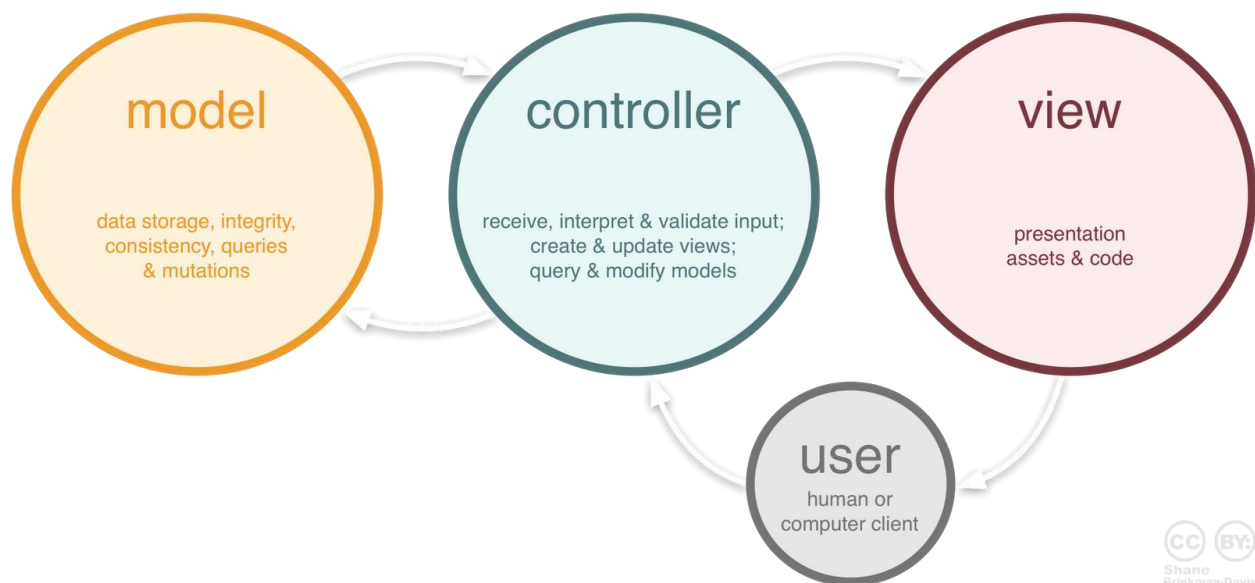
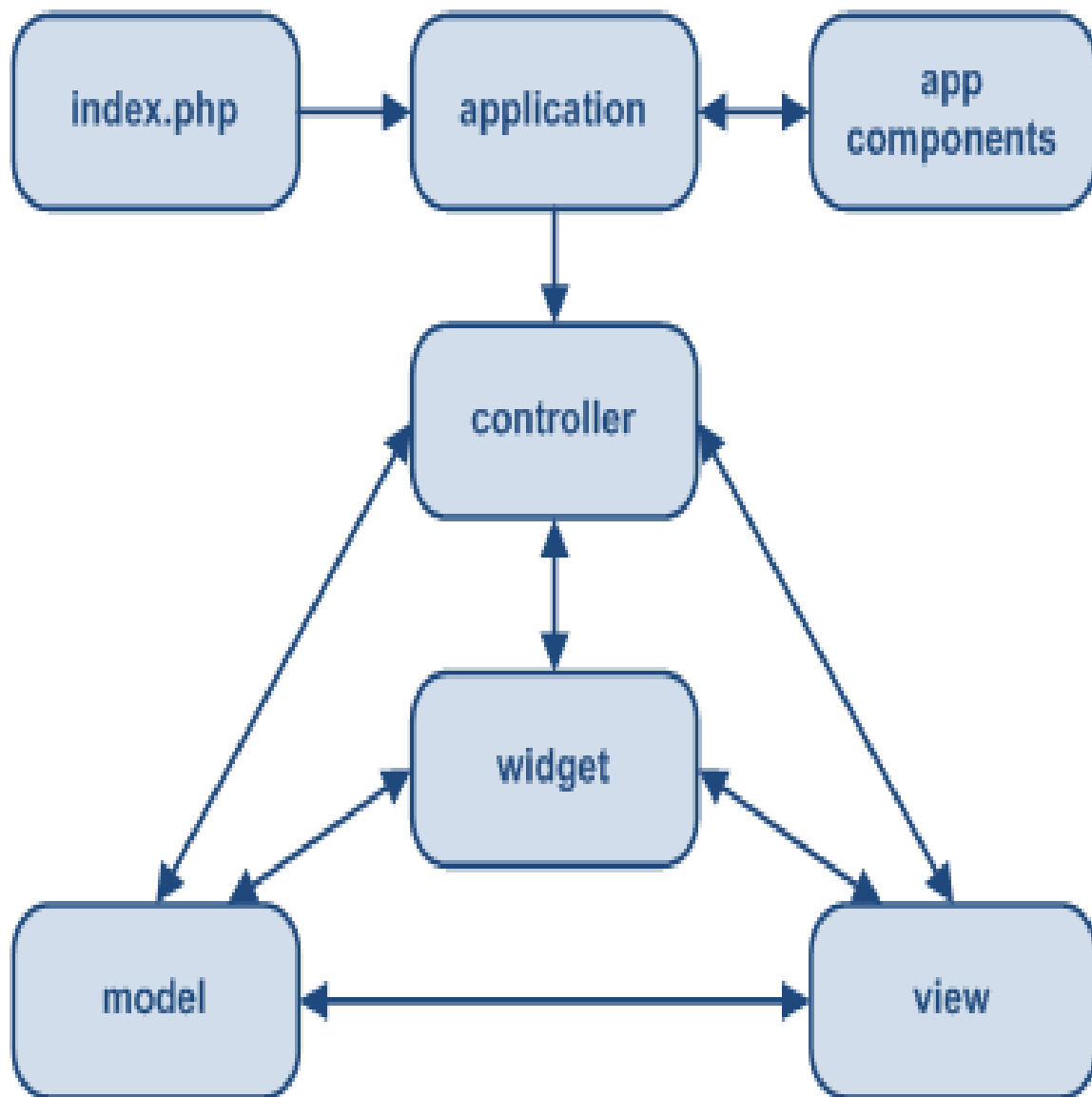


Figura 13- MVC

O diagrama seguinte mostra a estrutura estática de uma aplicação Yii:

1) Estrutura estática de uma aplicação Yii:



*Figura 14 - Estrutura estática aplicação yii*

De seguida irá ser apresentado um típico fluxo de execução de uma aplicação em Yii.

2) Fluxo de execução:

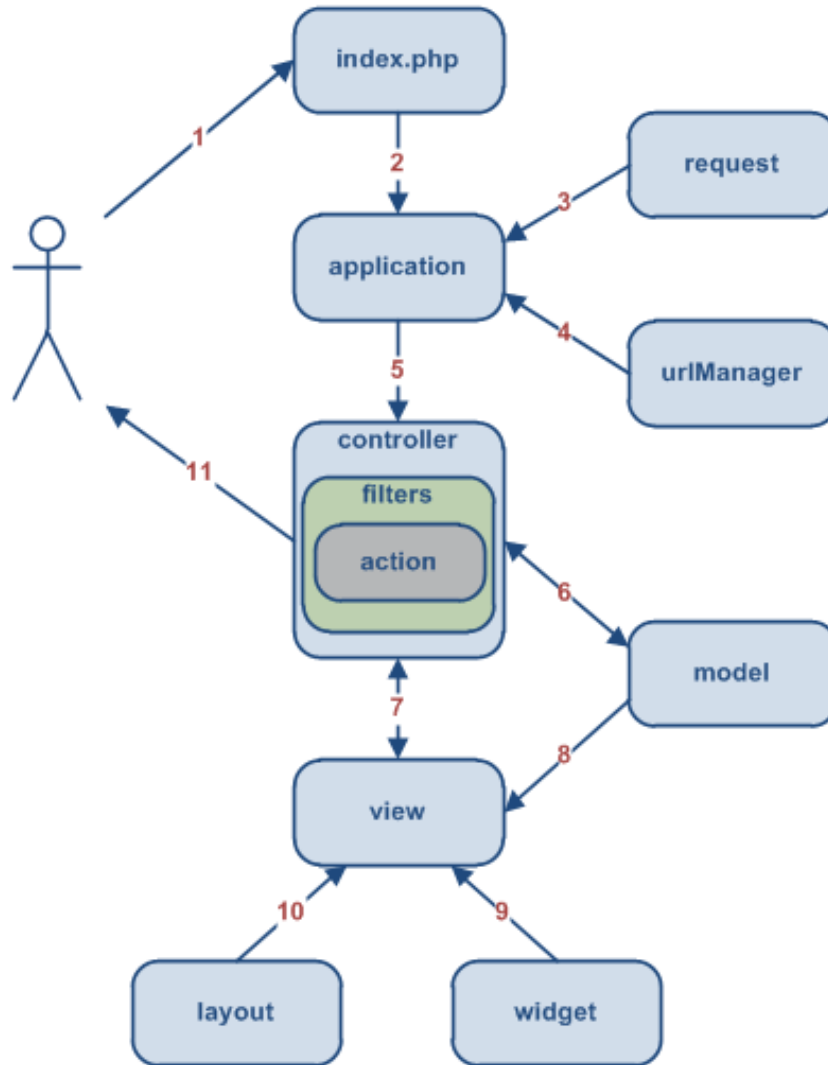


Figura 15 - Fluxo de execução yii

- 1) O utilizador faz uma solicitação com a URL `http://www.exemplo.com/index.php?r=post/show&id=1` e o servidor Web processa o pedido executando o script de bootstrap `index.php`.

- 2) O script de bootstrap cria uma instância da aplicação (application) e executa-a.
- 3) A aplicação obtém as informações detalhadas da solicitação de um componente da aplicação chamado request.
- 4) A aplicação determina o controler e a ação requerida com a ajuda do componente chamado urlManager. Para este exemplo, o controler é post que se refere à classe PostController e a ação é show cujo significado real é determinado no controler.
- 5) A aplicação cria uma instância do controler solicitado para poder lidar com a solicitação do utilizador. O controler determina que a ação show refere-se a um método chamado actionShow no controler da classe. Em seguida, cria e executa filtros (por exemplo, o controler de acesso, benchmarking) associados a esta ação. A ação só é executada se permitida pelos filtros.
- 6) A ação lê um modelo Post cujo ID é 1 na Base de Dados.
- 7) A ação processa a view chamada show, com o Post.
- 8) A view apresenta os atributos do modelo Post.
- 9) A view executa alguns widgets.
- 10) O resultado do processamento da view é embutido num layout.
- 11) A ação conclui o processamento da view e exhibe o resultado ao utilizador [12].

## 4.8 Git

O *Git* é um sistema de controlo de versões distribuído e um sistema de gestão de código fonte. O *Git* foi inicialmente projetado e desenvolvido por **Linus Torvalds** para o desenvolvimento do kernel Linux, mas foi adotado por muitos outros projetos. Cada diretório de trabalho do *Git* é um repositório com um histórico completo e

habilidade total de acompanhamento das revisões, não dependente de acesso a uma rede ou a um servidor central sendo uma ferramenta extremamente importante no desenvolvimento de software em equipa. O *Git* é um software livre, distribuído sob os termos da versão 2 da GNU General Public License [13].



*Figura 16 - Git*

#### **4.8.1 Gitlab**

O *GitLab* não é nada mais que um gestor de repositórios de software e controlo de versões. O *GitLab* é similar por exemplo ao *GitHub*, mas o *GitLab* permite que os desenvolvedores armazenem o código em servidores próprios, ao invés de servidores operados pelo *GitHub*.



*Figura 17 - Gitlab*

#### **4.9 IDE**

IDE, do inglês Integrated Development Environment ou Ambiente de Desenvolvimento Integrado, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo.

Geralmente os IDE's facilitam a técnica de RAD (de Rapid Application Development, ou "Desenvolvimento Rápido de Aplicativos"), que visa a maior produtividade dos desenvolvedores.

As características e ferramentas mais comuns encontradas nos IDEs são:

- Editor - edita o código-fonte do programa escrito na(s) linguagem(s) suportada(s) pela IDE;
- Compilador (compiler) - compila o código-fonte do programa, editado numa linguagem específica e a transforma em linguagem máquina;
- Linker - liga (linka) os vários "pedaços" de código-fonte, compilados em linguagem de máquina, em um programa executável que pode ser executado em um computador ou outro dispositivo computacional;
- Depurador (debugger) - auxilia no processo de encontrar e corrigir defeitos no código-fonte do programa, na tentativa de aprimorar a qualidade de software;
- Modelagem (modeling) - criação do modelo de classes, objetos, interfaces, associações e interações dos artefactos envolvidos no software com o objetivo de solucionar as necessidades-alvo do software final;
- Geração de código - característica mais explorada em Ferramentas CASE, a geração de código também é encontrada em IDEs, contudo com um escopo mais direcionado a templates de código comumente utilizados para solucionar problemas rotineiros. Todavia, em conjunto com ferramentas de modelagem, a geração pode gerar praticamente todo o código-fonte do programa com base no modelo proposto, tornando muito mais rápido o processo de desenvolvimento e distribuição do software;
- Distribuição (deploy) - auxilia no processo de criação do instalador do software, ou outra forma de distribuição;

- Testes Automatizados (automated tests) - realiza testes no software de forma automatizada, com base em scripts ou programas de testes previamente especificados, gerando um relatório, assim auxiliando na análise do impacto das alterações no código-fonte. Ferramentas deste tipo mais comuns no mercado são chamadas robôs de testes;
- Refatoração (refactoring) - consiste na melhoria constante do código-fonte do software, seja na construção de código mais otimizado, mais limpo e/ou com melhor entendimento pelos envolvidos no desenvolvimento do software. A refatoração, em conjunto com os testes automatizados, é uma poderosa ferramenta no processo de erradicação de "bugs", tendo em vista que os testes "garantem" o mesmo comportamento externo do software ou da característica sendo reconstruída [14].

#### **4.9.1 Eclipse**

O eclipse foi o IDE devido à familiarização já presente no estagiário.

Eclipse é um IDE para desenvolvimento principalmente para Java, porém suporta várias outras linguagens a partir de plugins como C/C++, PHP, ColdFusion, Python, Scala e plataforma Android. Está desenvolvido em Java e segue o modelo open source de desenvolvimento de software. O projeto Eclipse foi iniciado na IBM que desenvolveu a primeira versão do produto e doou-o como software livre para a comunidade. Hoje, o Eclipse é o IDE Java mais utilizado no mundo, possui como característica marcante o uso da SWT e não do Swing como biblioteca gráfica, a forte orientação ao desenvolvimento baseado em plug-ins e o amplo suporte ao desenvolvedor com centenas de plug-ins que procuram atender as diferentes necessidades de diferentes programadores.



*Figura 18 - Eclipse*

## **4.10 Wiki**

Durante todo o estágio foi utilizada uma política de registo de informação à medida que se iam efetuando novas atividades ou quando surgiam problemas em atividades já efetuadas, com este registo de informação é possível a melhor cooperação em equipa e também uma maior eficiência na elaboração de projetos dentro da empresa.

## **4.11 Inkscape**

Inkscape é um software livre para edição eletrónica de imagens e documentos vetoriais, com base numa versão mais avançada do antigo Sodipodi no qual teve origem. Trata-se assim de um fork considerado de sucesso.

Utiliza o método vetorial, ou seja, gera imagens a partir de um caminho de pontos definindo suas coordenadas, de forma transparente ao utilizador. Imagens vetoriais têm maior aplicação em desenho técnico ou artístico e são, geralmente, mais leves e não perdem a qualidade ao sofrer o facto dos formatos vetoriais ainda não possuem capacidade direta para lidar com captação de fotografias em tempo real, pelo que na maior parte das aplicações tecnológicas de captação de imagem, os formatos bitmap ainda são considerados standard.

O Inkscape trabalha nativamente com o formato SVG (Scalable Vectorial Graphics), um formato aberto de imagens vetoriais, nomeadamente, uma subdefinição (DTD)

da linguagem XML definido pela W3C. O aplicativo também exporta para o popular formato da Internet PNG e importa vários formatos vetoriais ou bitmap, como por exemplo: TIFF, GIF, JPG, AI, PDF, PS, entre outros.



*Figura 19 - Inkscape*

# Capítulo 5

## Trabalhos realizados

Ao longo deste capítulo irão ser apresentados alguns dos passos mais importantes no desenvolvimento da plataforma “*e-emprego*” no âmbito do estágio curricular. Também serão apresentadas tarefas menores que iam surgindo, inerente à necessidade da empresa.

### 5.1 Plataforma “e-emprego”

Como já foi referenciado o estagiário teve como principal tarefa o desenvolvimento de uma plataforma on-line intitulada de “e-emprego”. O principal objetivo da plataforma é a possibilidade de empresas e alunos/utilizadores poderem criar contatos de modo a promover o emprego. O principal objetivo desta aplicação será ao nível das instituições de ensino e os seus alunos.

Assim sendo, os papéis mais importantes no seio desta plataforma serão:

- **Utilizador** – que poderá pesquisar ofertas de trabalho e candidatar-se as mesmas;
- **Empresa** – que efetuará a inserção de ofertas de trabalho e poderá consultar os candidatos relativos a cada uma.
- **Gestor** – gere todas as estatísticas em relação a plataforma, número de ofertas, número de candidaturas, número de empresas registadas.
- **Admin** – O admin é o perfil que gere toda a plataforma desde as permissões que cada um dos outros perfis podem ter, bem como criar e eliminar utilizadores.

Foi a pensar nestes quatro papéis que se desenvolveu toda a plataforma e foi feito o levantamento de requisitos. De seguida será apresentado as funcionalidades inerentes a casa perfil.

### **Perfil de utilizador:**

- Efetuar o registo na plataforma;
- Perfil editável;
- Possibilidade de anexar ficheiros ao perfil (CV, carta de recomendação, etc.);
- Candidatar-se a qualquer oferta desde que cumpra os requisitos mínimos.
- Selecionar áreas de interesse inerentes às ofertas de trabalho de modo a receber notificações sobre essas mesmas ofertas.

### **Perfil de empresa:**

- Efetuar o registo na plataforma;
- Perfil editável;
- Criação de ofertas de emprego;
- Possibilidade de adicionar documentos a uma determinada oferta (mais detalhes/informações);
- Possibilidade de fechar uma oferta a qualquer momento;
- Atribuir a(s) oferta(s) ao(s) candidato(s), dependendo do número de vagas.

### **Perfil de gestor:**

- Visualização estatísticas (número de empresas registadas, número de ofertas ...);
- Possibilidade de criação de perfis de empresa e aluno.

## **Perfil de Admin:**

O perfil de “**admin**” é onde é gerida toda a plataforma desde permissões, a papéis desempenhado por cada tipo de utilizadores aqui chamado de “**roles**”, portanto é importante que este perfil seja só usado por administradores do sistema pois qualquer configuração errada pode resultar em falta de acesso aos utilizadores ou ainda pior serem permitido aceder a páginas que estes não deviam poder aceder.

Posto isto, avançou-se então para a configuração do ambiente de trabalho.

## **5.2 Configuração do Ambiente de Desenvolvimento**

Antes de começar o processo de desenvolvimento foi necessário preparar o ambiente de desenvolvimento. A primeira tarefa foi instalar o ubuntu 14.04 (Sistema operativo baseado em Unix) numa máquina com as seguintes características;

- Processador Intel i7 4700-HQ;
- 8 GB RAM;
- 256 Disco SSD;
- Placa gráfica AMD R9-265x

Tendo o sistema operativo operacional deu-se a instalação das ferramentas necessárias para o desenvolvimento em PHP.

As principais ferramentas necessárias para este tipo de desenvolvimento pode ser chamado como stack **LAMP**.

LAMP é uma combinação de softwares livres e de código aberto. O acrónimo LAMP refere-se as primeiras letras de:

- Linux (sistema operacional),
- Apache (servidor web),

- MariaDB ou MySQL (software de base de dados) e
- PHP (linguagens de programação) ou Python.

As instruções necessárias através de linha de comando para instalar este stack são as seguintes:

1) Instalação stack LAMP;

```
sudo apt-get update

sudo apt-get install apache2

sudo apt-get install mysql-server libapache2-mod-auth-mysql
php5-mysql

sudo mysql_install_db

sudo /usr/bin/mysql_secure_installation

sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt

sudo service apache2 restart
```

De seguida procedeu-se a instalação do Ambiente IDE que neste caso foi o *eclipse Mars 2*.

Findado este processo de configuração do ambiente de trabalho podemos então dar início ao processo de desenvolvimento. O próximo passo do desenvolvimento da plataforma foi sem duvida um dois minuciosos do projeto pois para uma base consistente de trabalho é necessário criar uma base de dados robusta de modo a evitar problemas durante o desenvolvimento de funcionalidades.

### 5.3 Criação da base de dados

Para o desenvolvimento deste projeto era importante o desenho consistente de uma base de dados que suportasse todas as funcionalidades necessárias, para isso usou-se a ferramenta *phpmyadmin* que vem associado a instalação do *apache2*. Assim sendo despendeu-se algum no tempo desenho e testes de consistência da base de dados. Poderá consultar a base de dados completa no Anexo.

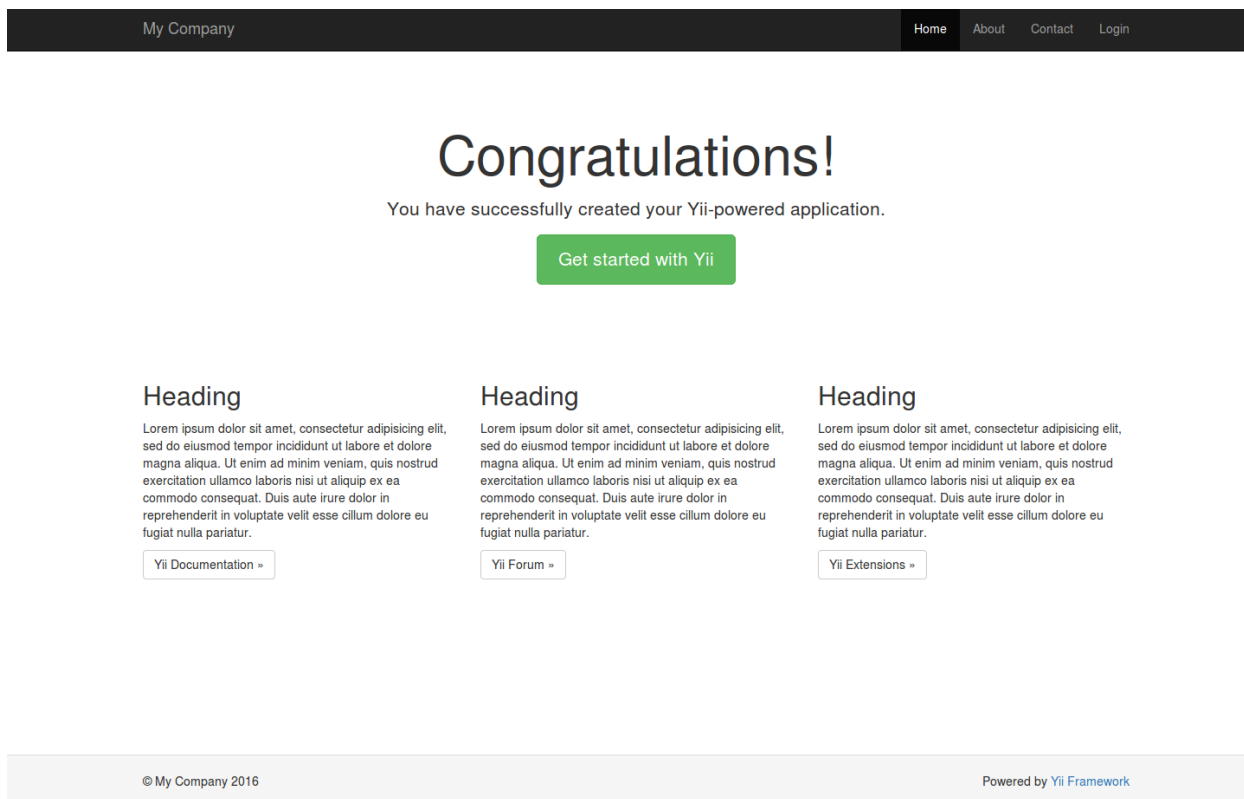
### 5.4 Instalação da framework yii2

Como já foi referido antes a *yii2* é uma framework de desenvolvimento *PHP*, dadas as características foi esta a framework escolhida para o desenvolvimento da plataforma.

1) Instalação da yii2 via linha de comandos;

```
# curl -sS https://getcomposer.org/installer | php
# mv composer.phar /usr/local/bin/composer
# composer global require "fxp/composer-asset-
plugin:^1.2.0"
# composer create-project --prefer-dist yiisoft/yii2-app-
basic e-emprego
```

Pode-se então observar ao aceder ao servidor local que a instalação da yii2 foi efetuado com sucesso.



*Figura 20 - Instalação da Yii*

Quando é apresentada a imagem na figura .20 significa que a framework foi instalada com sucesso e que ocorreu sem erros. Assim a partir daqui pode-se desenvolver a aplicação web, neste caso a plataforma de emprego a nível de código. Para uma melhor gestão do desenvolvimento do projeto vai-se usar um controlo de versões que irá manter sempre em dia todas as alterações de código efetuadas para poder haver um controlo rigoroso sobre casa passo, e evitar que possam acontecer problemas na conceção da plataforma. Como já foi referido vai ser utilizado o **git** como controlo de versões e é isso que irá ser apresentado com detalhe de seguida.

### 5.4.1 Repositório git

Para o controlo de versões, ou seja para gerir cada fase de evolução do código do projeto, usou-se o git como já foi referido e para isso basta iniciar o git na linha de comandos através do comando *git init* na pasta do projeto onde deseja ter o controlo de versões.

De seguida irão ser apresentados alguns dos principais comandos associados ao *git*.

#### Iniciar o git:

- crie uma nova pasta, abra-a e execute o comando *git init* para criar um novo repositório.

#### Obter um repositório:

- cria-se uma cópia do trabalho num repositório local executando o comando *git clone /caminho/para/o/repositório* quando usar um servidor remoto, o comando será *git clone utilizador@servidor:/caminho/para/o/repositório*.

#### Fluxo de trabalho:

- os repositórios locais consistem em três "árvores" mantidas pelo *git*. A primeira delas é a **Working Directory** que contém os arquivos vigentes. A segunda é o **Index** que funciona como uma área temporária e finalmente a **HEAD** que aponta para o último commit (confirmação) que foi feita.

#### Adicionar e confirmar alterações:

- Pode-se propor mudanças (adicioná-las ao **Index**) usando *git add <arquivo>* *git add .* este é o primeiro passo no fluxo de trabalho básico do *git*. Para realmente confirmar estas mudanças (isto é, fazer um *commit*), usa-se *git*

*commit -m "comentários das alterações"* Agora o arquivo é enviado para o **HEAD**, mas ainda não para o repositório remoto.

### **Enviar as alterações:**

- As alterações agora estão no HEAD da cópia de trabalho local. Para enviar estas alterações para o repositório remoto, executa-se *git push origin master*. Altera-se master para qualquer ramo (*branch*) desejado, enviando as alterações.

## **5.4.2 Usermanagement yii2**

Aquando da descrição da ferramenta *yii2* ficou-se a perceber que esta ferramenta possui um número grande de extensões que podem ser instaladas e integradas após a instalação da *yii2*. Uma das extensões usadas e provavelmente a mais importante foi o *user management module*.

Esta extensão basicamente controla a gestão de utilizadores, *gestão de acessos* e *gestão de roles* e *permissions* também conhecido por **RBAC**. O **RBAC (role-based access control)** é um mecanismo de controlo de acesso que é definida em termos de funções e privilégios. Os componentes do RBAC, tais como role, premissions, user-role, role-role, podem ser atribuídos a diferentes utilizadores.

Os perfis utilizados nesta plataforma acima descritos derivaram do uso do “*user managent*”.

- 1) Instalação da extensão user managent:

```
# composer require webvimark/module-user-management
```

Após a instalação é necessário efetuar algumas configurações.

## 2) Configuração no config/web.php:

```
'components'=>[
    'user' => [
        'class' =>
'webvimark\modules\UserManagement\components\UserConfig',

        // Comment this if you don't want to record user
logins
        'on afterLogin' => function($event) {

'webvimark\modules\UserManagement\models\UserVisitLog::newVi
sitor($event->identity->id);
        }
    ],
],

'modules'=>[
    'user-management' => [
        'class' =>
'webvimark\modules\UserManagement\UserManagementModule',

        // 'enableRegistration' => true,
```

```

        // Add regexp validation to passwords. Default
pattern does not restrict user and can enter any set of
characters.

        // The example below allows user to enter :
// any set of characters
// (?=\S{8,}): of at least length 8
// (?=\S*[a-z]): containing at least one lowercase
letter
// (?=\S*[A-Z]): and at least one uppercase letter
// (?=\S*[\d]): and at least one number
// $: anchored to the end of the string

        // 'passwordRegexp' => '^\\S*(?=\S{8,})(?=\S*[a-
z])(?=\S*[A-Z])(?=\S*[\d])\\S*$',

        // Here you can set your handler to change layout
for any controller or action
// Tip: you can use this event in any module
'on beforeAction'=>function(yii\base\ActionEvent
$event) {
            if ( $event->action->uniqueId == 'user-
management/auth/login' )
            {
                $event->action->controller->layout =
'loginLayout.php';

```

```
        }  
    },  
],  
],
```

3) Configuração config/console.php:

```
'modules'=>[  
    'user-management' => [  
        'class' =>  
'webvimark\modules\UserManagement\UserManagementModule',  
  
'controllerNamespace'=>'vendor\webvimark\modules\UserManagem  
ent\controllers', // To prevent yii help from crashing  
    ],  
],
```

4) Fazer as migrações:

```
./yii migrate --migrationPath=vendor/webvimark/module-user-  
management/migrations/
```

As migrações consistem na ligação com a base de dados da plataforma e a criação de tabelas associadas a instalação do “**user management**”.

## 5) Configuração do base controller:

```
public function behaviors()
{
    return [
        'ghost-access' => [
            'class' =>
'webvimark\modules\UserManagement\components\GhostAccessCont
rol',
        ],
    ];
}
```

Uma fase crucial da configuração do “**user management module**” é a criação de uma chave estrangeira nas tabelas de utilizador e empresa (*user\_id*) que vai ser utilizada pela tabela *user* do user-managent, sem este passo é impossível fazer a gestão de utilizadores.

Como já foi referido em cima a tabela “**user**” derivada da migração feita aquando da instalação do “**user management module**”.

Com esta pequena configuração é então possível usufruir de todas as funcionalidades inerentes a esta extensão que é extremamente importante no desenvolvimento de toda a plataforma.

## 5.4.3 Criação CRUD

Outra grande vantagem do uso da framework yii2 é a geração de **CRUD** automático.

O **CRUD** (acrónimo de Create, Read, Update e Delete na língua Inglesa) para as quatro operações básicas utilizadas em bases de dados relacionais (RDBMS) ou em interface para utilizadores para criação, consulta, atualização e eliminação de dados.

Para efetuar isto, basta apenas aceder ao link **/index.php?r=gii/**.

The screenshot shows the Gii code generator interface. At the top, there is a navigation bar with the Yii logo and the text 'yii code generator'. Below this, the main heading reads 'Welcome to Gii a magical tool that can write code for you'. The page lists six code generators, each with a description and a 'Start »' button:

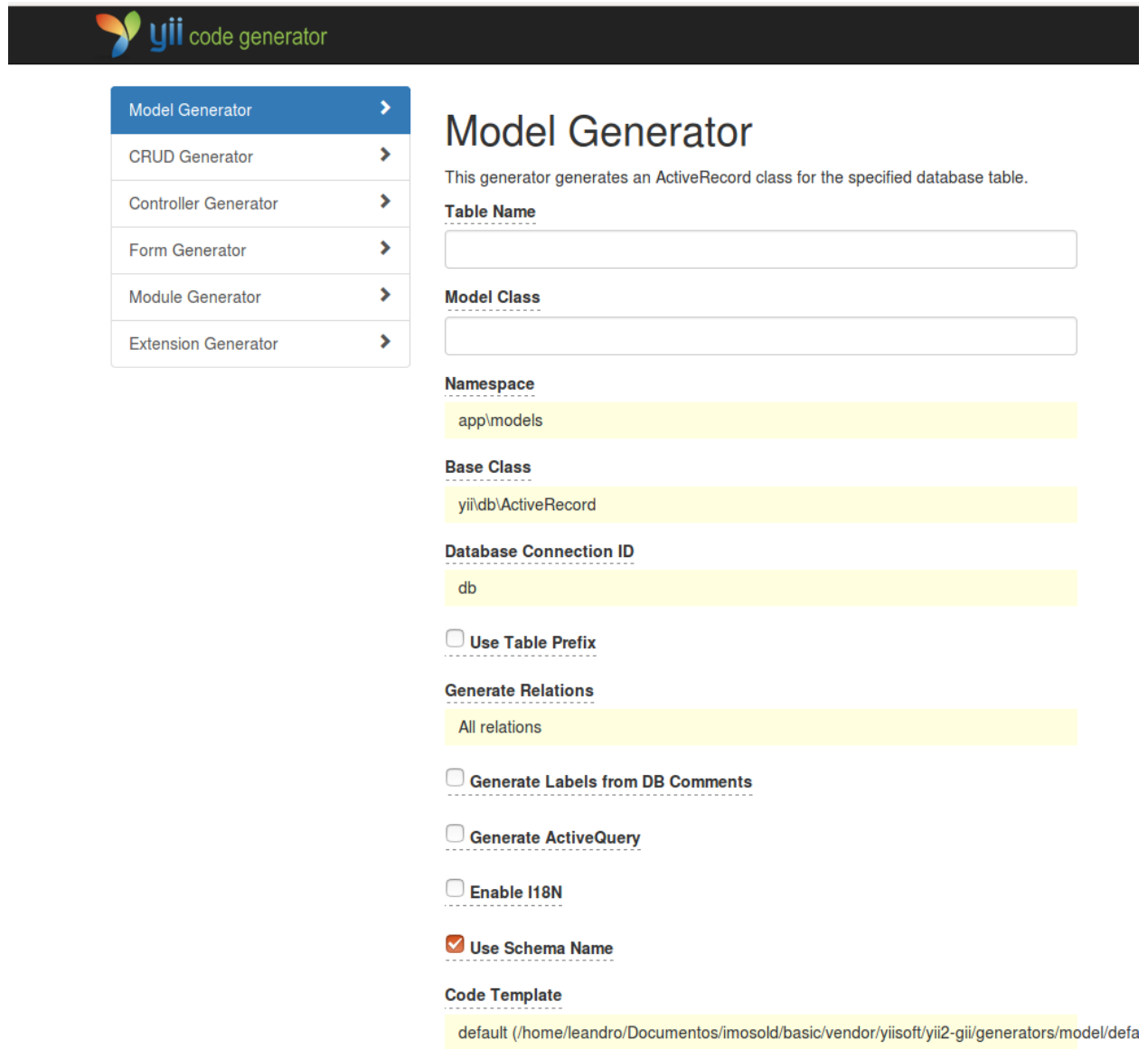
- Model Generator**: This generator generates an ActiveRecord class for the specified database table.
- CRUD Generator**: This generator generates a controller and views that implement CRUD (Create, Read, Update, Delete) operations for the specified data model.
- Controller Generator**: This generator helps you to quickly generate a new controller class with one or several controller actions and their corresponding views.
- Form Generator**: This generator generates a view script file that displays a form to collect input for the specified model class.
- Module Generator**: This generator helps you to generate the skeleton code needed by a Yii module.
- Extension Generator**: This generator helps you to generate the files needed by a Yii extension.

At the bottom left, there is a green button labeled 'Get More Generators'. The footer contains the text 'A Product of Yii Software LLC' and 'Powered by Yii Framework'.

*Figura 21 - Acesso à página models e CRUD generator*

De seguida é necessário gerar os modelos a partir das tabelas da base dados que irão permitir definir várias regras para cada tabela e conseqüentemente cada campo da tabela.

Abaixo é apresentado a geração de um modelo.



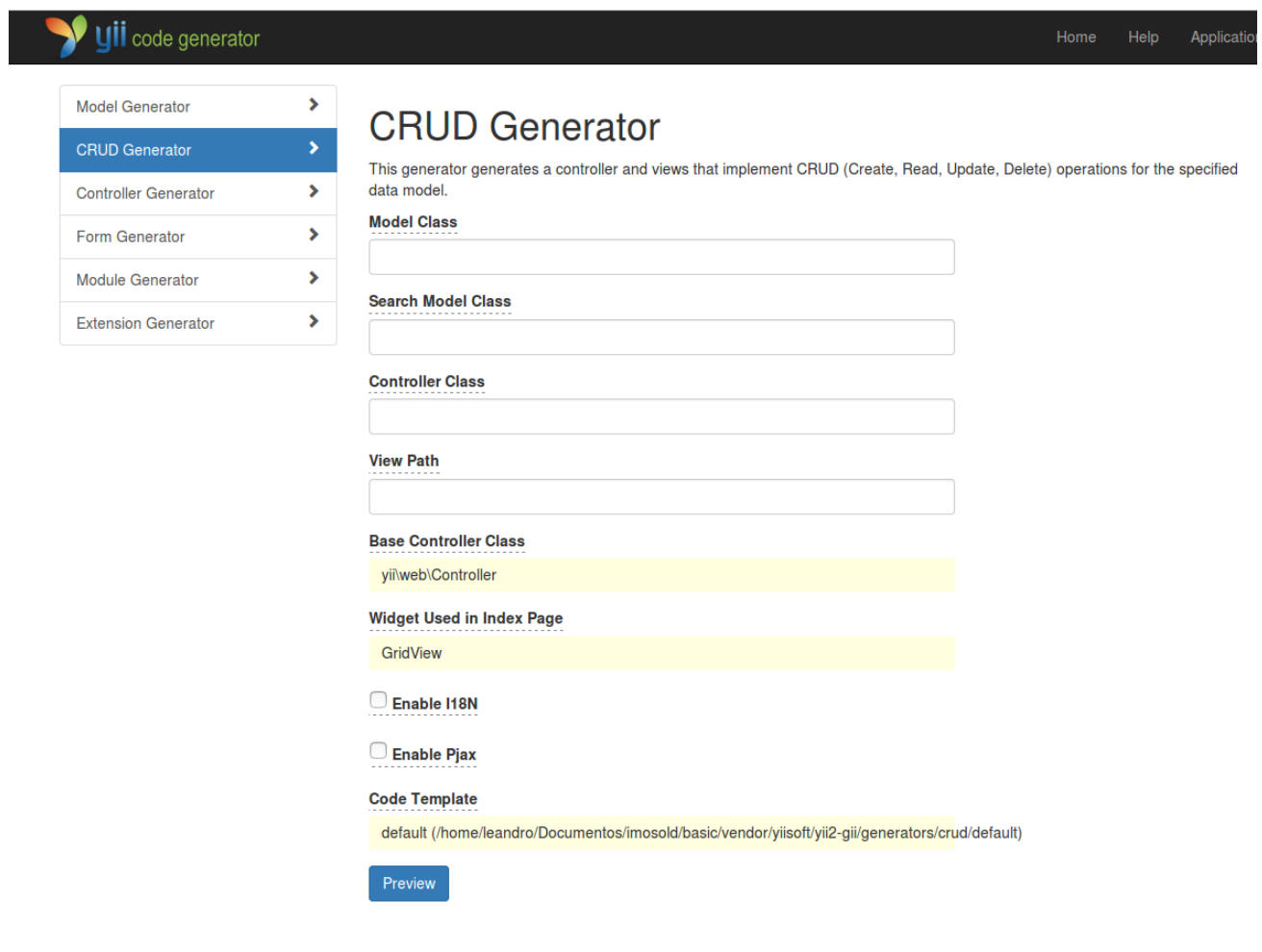
The image shows the 'Model Generator' interface of the Yii Code Generator. On the left is a sidebar menu with options: Model Generator (selected), CRUD Generator, Controller Generator, Form Generator, Module Generator, and Extension Generator. The main area is titled 'Model Generator' and contains the following fields and options:

- Table Name:** An empty text input field.
- Model Class:** An empty text input field.
- Namespace:** A text input field containing 'app\models'.
- Base Class:** A text input field containing 'yii\db\ActiveRecord'.
- Database Connection ID:** A text input field containing 'db'.
- Use Table Prefix:** An unchecked checkbox.
- Generate Relations:** A dropdown menu set to 'All relations'.
- Generate Labels from DB Comments:** An unchecked checkbox.
- Generate ActiveQuery:** An unchecked checkbox.
- Enable I18N:** An unchecked checkbox.
- Use Schema Name:** A checked checkbox.
- Code Template:** A text input field containing 'default (/home/leandro/Documentos/imosold/basic/vendor/yiisoft/yii2-gii/generators/model/defa'.

Figura 22 - Model generator

Gerados os modelos pode-se então de seguida gerar os **controllers** e **views** que são responsáveis pelo chamado **CRUD** como já foi referido anteriormente.

De seguida é então apresentado um exemplo de como são gerados automaticamente após preencher apenas alguns campos.



The screenshot shows the 'yii code generator' interface. On the left is a sidebar menu with options: Model Generator, **CRUD Generator** (highlighted), Controller Generator, Form Generator, Module Generator, and Extension Generator. The main content area is titled 'CRUD Generator' and includes a description: 'This generator generates a controller and views that implement CRUD (Create, Read, Update, Delete) operations for the specified data model.' Below this are several form fields: 'Model Class' (empty), 'Search Model Class' (empty), 'Controller Class' (empty), 'View Path' (empty), 'Base Controller Class' (pre-filled with 'yii\web\Controller'), 'Widget Used in Index Page' (pre-filled with 'GridView'), 'Enable I18N' (checkbox), 'Enable Pjax' (checkbox), and 'Code Template' (pre-filled with a file path). A 'Preview' button is located at the bottom.

*Figura 23 - CRUD*

#### 5.4.4 Implementação de funcionalidades e demonstração do trabalho

Após todas estas instalações e configurações é altura de aplicar as principais funcionalidades já anteriormente identificadas. A partir deste momento foi feito o

trabalho de programador propriamente dito, através de elaboração de código *PHP*, assim sendo doravante serão apresentadas algumas figuras da plataforma com algumas das principais funcionalidades desenvolvidas.

Vai-se então apresentar de seguida a página principal da plataforma onde se pode efetuar login, registo e ainda a apresentação de um gráfico com algumas estatísticas do site.

É de salientar que foi necessário efetuar dois tipos de registos diferentes pois existem dois perfis distintos da plataforma como já foi referido.



Figura 24 - Página principal da plataforma

Abaixo é apresentada a área de registo para o perfil de “empresa” que é diferente do perfil de “aluno/utilizador”.

Os outros dois perfis da plataforma (“**Gestor, Admin**”) existem por defeito, ou seja, o “**admin**” é criado automaticamente, e o perfil de “**gestor**” é uma role derivada do “**admin**” só que com permissões mais limitadas pois como já foi referido acima o “**admin**” só deve ser usado pelo administrador do sistema.

### Registo de empresa

<b>E-mail/Login</b>	<input type="text"/>
<b>Password</b>	<input type="password"/>
<b>Repita a password</b>	<input type="password"/>
<b>Nome da empresa</b>	<input type="text"/>
<b>NIF</b>	<input type="text"/>
<b>Ramo de negócio</b>	<input type="text"/>
<b>Morada</b>	<input type="text"/>

*Figura 25 - Registo de empresa*

Aqui pode-se ver o registo de empresa e os campos necessários para poder-se efetuar o registo de uma empresa, como já foi referido a existência de dois perfis distintos neste caso o conceito de “empresa” e por outro lado o conceito de “utilizador” levou a necessidade de criação de dois tipos de registos diferentes e consequentemente com campos de preenchimento também diferentes assim o aspeto do registo de utilizador será idêntico ao da figura .24 mas como campos de preenchimento diferentes.

Após o registo a empresa é redirecionada para o seu perfil onde pode alterar dados e preencher outros que ache pertinentes.

The screenshot shows a web application interface for updating a company profile. The header is a purple bar with the text 'e-emprego' on the left and navigation links 'Perfil Empresa', 'Criar oferta', 'As minhas ofertas', 'As minhas ofertas fechadas', and 'Logout' on the right. Below the header is a breadcrumb trail: 'Home / Innovtek / Atualizar'. The main heading is 'Atualizar Empresa: Innovtek'. The form consists of several sections:

- Nome da Empresa:** A text input field containing 'Innovtek'.
- Ramo Negócio:** A text input field containing 'Desenvolvimento de Software'.
- Nif:** A text input field containing '510521762'.
- CAE:** An empty text input field.
- Número de empregados:** A text input field containing '1'.
- Pais:** A dropdown menu with the placeholder text 'Indique o país'.
- Distrito:** A dropdown menu with the placeholder text 'Indique o distrito'.
- Concelho:** A dropdown menu with the placeholder text 'Indique o concelho'.
- Morada:** A text input field containing 'Campus Santa Apolonia Edificio Estig Gabinete Ruralnet'.
- Codigo Postal:** A text input field containing '5300-253'.
- Telefone:** A text input field containing '273303186'.
- Pessoa Contacto:** A text input field containing 'David Dias'.
- Cargo Pessoa Contacto:** A text input field containing 'Programador'.
- E-mail:** A text input field containing 'innovtek@innovtek.net'.
- Web:** An empty text input field.
- Apresentação Empresa:** A large empty text area for a company description.

At the bottom left of the form is a blue button labeled 'Atualizar'.

*Figura 26 - Perfil de empresa*

Na figura .25 pode-se ver que uma empresa ao efetuar um registo é depois reencaminha para uma página para atualizar os seus dados de modo a poder pormenorizar mais detalhadamente todas as suas informações que não foram pedidas aquando do registo. O mesmo é aplicável para o perfil de “utilizador”.

Após completar o perfil a empresa pode então começar a inserção de ofertas de trabalho, gerir as suas ofertas ativas e pode também encerrar o estado da(s) oferta(s) para fechado. De seguida irá ser apresentado o primeiro passo da criação de uma oferta.

The screenshot shows a web interface for creating a job offer. At the top, there is a purple navigation bar with the text 'e-emprego' on the left and several menu items: 'Perfil Empresa', 'Criar oferta' (highlighted in black), 'As minhas ofertas', 'As minhas ofertas fechadas', and 'Logout'. Below the navigation bar is a progress bar indicating 33% completion. The main form area contains several input fields and dropdown menus: 'Título da Oferta' (text input), 'Estado da Oferta' (dropdown menu with 'Aberta' selected), 'Descrição da Oferta' (text area), 'Número de vagas' (text input), 'Local de Trabalho' (text input), 'Horário de Trabalho' (text input), and 'Regime' (dropdown menu with 'Tempo Inteiro' selected). At the bottom left of the form is a blue button labeled 'Próximo Passo'.

*Figura 27 - Primeiro passo criação de oferta*

Após a criação, podemos gerir as ofertas e ter vista detalhada sobre uma determinada oferta e ter opções como atualizar a oferta, fechar oferta, associar anexos à oferta e consultar esses mesmos documentos. Também se pode ver os candidatos que efetuaram candidatura à oferta em questão. Por outro lado, o “utilizador” irá também ter a visão detalhada da oferta bem como a possibilidade de se candidatar oferta se assim o pretender e ainda consultar eventuais ficheiros que a oferta tenha associados. Isto permite obter o maior número de informação acerca de uma proposta de emprego.

## Programador WEB

1 - Descrição da Oferta			
Data da Oferta	Jun 24, 2016	Número de vagas	1
Local de Trabalho	Bragança	Horário de Trabalho	09-18
Resumo da Oferta	Procura-se programador com experiência em PHP		
2 - Condições Necessárias			
Requisitos Mínimos	Licenciatura		
Requisitos Pretendidos	Licenciatura		
3 - Condições Oferecidas			
Tipo de contrato	Termo	Duração do Contrato	1 Ano
		Regime	Tempo Inteiro
Salário	1000.00	Comissões e incentivos	0%

[Atualizar Oferta](#)
[Fechar Oferta](#)
[Anexar Documentos à oferta](#)
[Documentos](#)

## Candidatos

#	Nome	Grau	E-mail/Login	Situação Profissional
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
No results found.				

*Figura 28 - Oferta detalhada*

Como se pode observar na figura .27 tem-se uma visão geral sobre a oferta em questão desde a descrição de oferta, condições necessárias e condições oferecidas. Tentou-se agrupar toda esta informação num único sitio de modo a ser possível visualizar com clareza todos os dados pertinentemente pois hoje em dia uma das principais características das plataformas web são a navegabilidade e a disponibilização clara de informação.

As figuras até então exibidas enaltecem as principais funcionalidades inerentes ao perfil “empresa”, não se irá mostrar as mesmas para o perfil “utilizador” pois como já foi referido, são em muitos semelhantes sendo diferente apenas que o perfil “utilizador” procura ofertas e este pode efetuar uma candidatura a uma determinada oferta, contudo salienta-se a funcionalidade de o perfil “utilizador” poder selecionar áreas de interesse no seu perfil para assim estar sempre a par das últimas ofertas inseridas do seu interesse. É isso que mostra a figura seguinte.

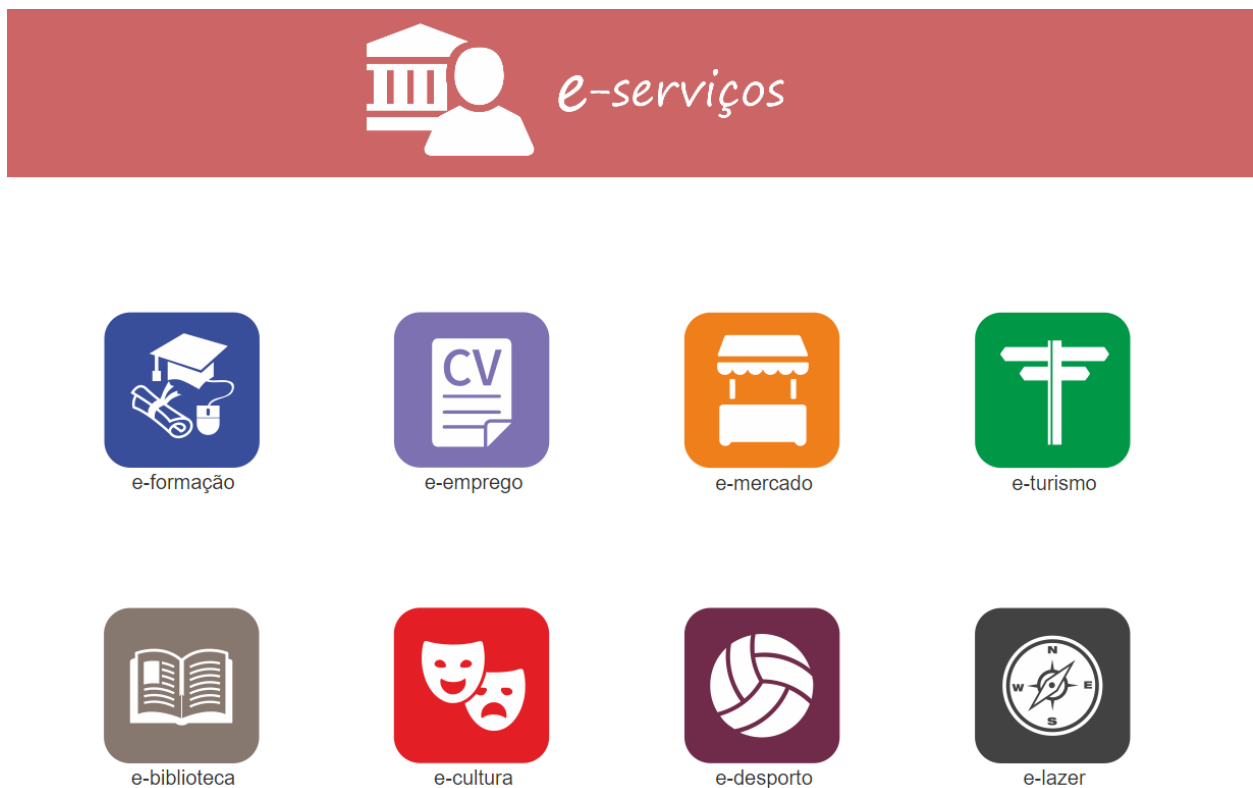


*Figura 29 - Áreas de interesse do utilizador*

Na Figura .28 pode-se observar uma importante funcionalidade da plataforma, um “utilizador” pode na sua área de perfil selecionar as suas preferências de interesse o que irá permitir este estar sempre a par das últimas ofertas com as suas áreas de interesse pois foi implementado o envio de e-mails automáticos aquando a criação de ofertas por partes das empresas.

## 5.5 Outras tarefas

Durante o estágio curricular foram desenvolvidas outras tarefas menores que iam surgindo consoante a necessidade da empresa, principalmente na área do design, como a desenvolvimento de alguns símbolos e layouts. Para isso a principal ferramenta utilizada foi o *inkscape* que já foi descrita anteriormente. Um exemplo deste trabalho desenvolvido pode ser visto na figura abaixo onde é apresentada uma página principal de uma plataforma da empresa.



*Figura 30 - Design Plataforma e-serviços*

Inicialmente foi desenhado cada símbolo no “*inkscape*” e depois o layout foi montado usando *css* e *javascript* pois os símbolos reagem assim que passamos o cursor por cima o que não pode ser visível na figura 29.

## Capítulo 6

### Conclusão e trabalho futuro

Neste período de estágio, o contacto com novas tecnologias, a integração numa equipa de trabalho e a experiência de desenvolver projetos inseridos em ambientes empresariais são algumas das competências adquiridas que completam a formação académica.

O trabalho desenvolvido permitiu interiorizar vários dos conhecimentos adquiridos durante toda o caminho académico, permitindo a execução de todas as fases de um projeto, como a análise e especificação de requisitos, desenho, implementação e testes, acompanhada da respetiva elaboração da documentação necessária e relatório. Relativamente a trabalhos futuros, devido ao aumento das necessidades provenientes do negócio de cada empresa e à sua adaptação aos mercados onde se encontram, podem surgir novos pedidos de funcionalidades, de modo a tornar o Portal do emprego cada vez mais útil e funcional. No entanto os objetivos inicialmente propostos foram largamente atingidos, e a plataforma inicialmente pensada ficou funcional.

A maior dificuldade durante todo este processo de estágio foi sem dúvida a realização deste relatório principalmente no capítulo número 5 que corresponde à demonstração do trabalho realizado, pois como sendo desenvolvimento software fica complicado mostrar realmente o trabalho realizado por detrás da cada figura apresentada da plataforma. No entanto considero que o ponto fulcral da realização deste estágio curricular foi trabalhar num seio de uma empresa que se revelou extremamente enriquecedor tanto a nível pessoal como profissional porque serviu

como uma preparação para o mercado de trabalho e não podia ter sido uma melhor preparação.

Uma conclusão final é a de que numa profissão cada vez mais competitiva, existem duas competências que podem fazer a diferença: a especialização e a proatividade. Foi nesse sentido que este projeto foi desenvolvido. Obtenção de know-how específico e bastante requisitado no mercado das TI e desenvolver a capacidade de iniciativa, rentabilizando o esforço e otimizando a produtividade, e assim transformar estas competências em mais-valias quer pessoais quer profissionais.

# Referências

- [1] (10/9/2016). <http://study.com/academy/lesson/what-are-information-systems-definition-types-quiz.html>
- [2] (10/9/2016). [https://en.wikipedia.org/wiki/Open-source\\_software](https://en.wikipedia.org/wiki/Open-source_software)
- [3] (10/9/2016). <https://secure.php.net/manual/en/intro-what-is.php>
- [4] (13/9/2016). <https://secure.php.net/manual/en/intro-what-cando.php>
- [5] (17/9/2016). [https://httpd.apache.org/ABOUT\\_APACHE.html](https://httpd.apache.org/ABOUT_APACHE.html)
- [6] (18/9/2016). <http://www.infowester.com/servapach.php>
- [7] (30/9/2016). <http://www.infowester.com/postgremysql.php>
- [8] (1/10/2016). <https://en.wikipedia.org/wiki/JavaScript>
- [9] (10/10/2016). <https://en.wikipedia.org/wiki/HTML>
- [10] (10/10/2016). [https://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets)
- [11] (10/10/2016). <http://www.yiiframework.com/about/>
- [12] (10/10/2016). <http://www.yiiframework.com/doc/guide/1.1/en/basics.mvc>
- [13] (10/10/2016). <https://en.wikipedia.org/wiki/Git>
- [14] (10/10/2016). [https://en.wikipedia.org/wiki/Integrated\\_development\\_environment](https://en.wikipedia.org/wiki/Integrated_development_environment)

# Anexos

## Estrutura da base de dados

```
CREATE TABLE `aluno` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `id_curso` int(11) DEFAULT NULL,  
  `concelho_id` int(11) DEFAULT NULL,  
  `id_escola` int(11) DEFAULT NULL,  
  `nome` varchar(90) DEFAULT NULL,  
  `grau` varchar(45) DEFAULT NULL,  
  `anoIngresso` int(11) DEFAULT NULL,  
  `contingente` varchar(12) DEFAULT NULL,  
  `dataNascimento` date DEFAULT NULL,  
  `telemovel` varchar(15) DEFAULT NULL,  
  `email` varchar(45) DEFAULT NULL,  
  `codigo` int(11) DEFAULT NULL,  
  `dataConclusaoCurso` date DEFAULT NULL,  
  `linkedin` varchar(120) DEFAULT NULL,  
  `facebook` varchar(120) DEFAULT NULL,  
  `situacao` varchar(45) DEFAULT NULL,  
  `loginUtilizador` varchar(60) DEFAULT NULL,  
  `divulgaInformacao` varchar(15) DEFAULT NULL,  
  `morada` varchar(90) DEFAULT NULL,  
  `codigoPostal` varchar(30) DEFAULT NULL,  
  `bi` int(11) DEFAULT NULL,  
  `nif` int(11) DEFAULT NULL,  
  `password` varchar(45) DEFAULT NULL,  
  `carta` varchar(900) DEFAULT NULL,  
  `id_utilizador` int(11) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `fkCurso_idx` (`id_curso`),  
  KEY `fkConcelho_idx` (`concelho_id`),  
  KEY `fkEscola_idx` (`id_escola`),  
  CONSTRAINT `fk_concelho` FOREIGN KEY (`concelho_id`) REFERENCES `concelho`  
  (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT `fk_curso` FOREIGN KEY (`id_curso`) REFERENCES `curso` (`id`) ON  
  DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT `fk_escola` FOREIGN KEY (`id_escola`) REFERENCES `escola` (`id`)  
  ON DELETE NO ACTION ON UPDATE NO ACTION  
  ) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8;  
  
CREATE TABLE `area_profissional` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `nome` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
  ) ENGINE=InnoDB AUTO_INCREMENT=36 DEFAULT CHARSET=utf8;
```

```

CREATE TABLE `area_profissional_aluno` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `aluno_id` int(11) DEFAULT NULL,
  `area_profissional_id` int(11) DEFAULT NULL,
  `grau` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_aluno_idx` (`aluno_id`),
  KEY `fk_area_profissional_aluno_1_idx` (`area_profissional_id`),
  CONSTRAINT `fk_aluno_area` FOREIGN KEY (`aluno_id`) REFERENCES `aluno`
(`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_area_profissional_aluno_1` FOREIGN KEY
(`area_profissional_id`) REFERENCES `area_profissional` (`id`) ON DELETE NO
ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `area_profissional_oferta` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `oferta_id` int(11) DEFAULT NULL,
  `area_profissional_id` int(11) DEFAULT NULL,
  `grau` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_oferta_idx` (`oferta_id`),
  KEY `fk_area_profissional_oferta_1_idx` (`area_profissional_id`),
  CONSTRAINT `fk_area_profissional_oferta_1` FOREIGN KEY
(`area_profissional_id`) REFERENCES `area_profissional` (`id`) ON DELETE NO
ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_oferta_area` FOREIGN KEY (`oferta_id`) REFERENCES `oferta`
(`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=387 DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `auth_assignment` (
  `item_name` varchar(64) NOT NULL,
  `user_id` int(11) NOT NULL,
  `created_at` int(11) DEFAULT NULL,
  PRIMARY KEY (`item_name`,`user_id`),
  KEY `user_id` (`user_id`),
  CONSTRAINT `auth_assignment_ibfk_1` FOREIGN KEY (`item_name`) REFERENCES
`auth_item` (`name`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `auth_assignment_ibfk_2` FOREIGN KEY (`user_id`) REFERENCES
`user` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `auth_item` (
  `name` varchar(64) NOT NULL,
  `type` int(11) NOT NULL,
  `description` text,
  `rule_name` varchar(64) DEFAULT NULL,
  `data` text,
  `created_at` int(11) DEFAULT NULL,

```

```

`updated_at` int(11) DEFAULT NULL,
`group_code` varchar(64) DEFAULT NULL,
PRIMARY KEY (`name`),
KEY `rule_name` (`rule_name`),
KEY `idx-auth_item-type` (`type`),
KEY `fk_auth_item_group_code` (`group_code`),
CONSTRAINT `auth_item_ibfk_1` FOREIGN KEY (`rule_name`) REFERENCES
`auth_rule` (`name`) ON DELETE SET NULL ON UPDATE CASCADE,
CONSTRAINT `fk_auth_item_group_code` FOREIGN KEY (`group_code`) REFERENCES
`auth_item_group` (`code`) ON DELETE SET NULL ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `auth_item_child` (
  `parent` varchar(64) NOT NULL,
  `child` varchar(64) NOT NULL,
  PRIMARY KEY (`parent`,`child`),
  KEY `child` (`child`),
  CONSTRAINT `auth_item_child_ibfk_1` FOREIGN KEY (`parent`) REFERENCES
`auth_item` (`name`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `auth_item_child_ibfk_2` FOREIGN KEY (`child`) REFERENCES
`auth_item` (`name`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `auth_item_group` (
  `code` varchar(64) NOT NULL,
  `name` varchar(255) NOT NULL,
  `created_at` int(11) DEFAULT NULL,
  `updated_at` int(11) DEFAULT NULL,
  PRIMARY KEY (`code`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `auth_rule` (
  `name` varchar(64) NOT NULL,
  `data` text,
  `created_at` int(11) DEFAULT NULL,
  `updated_at` int(11) DEFAULT NULL,
  PRIMARY KEY (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `candidatura` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_aluno` int(11) DEFAULT NULL,
  `id_oferta` int(11) DEFAULT NULL,
  `data` date DEFAULT NULL,
  `motivo` varchar(900) DEFAULT NULL,
  `estado` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_aluno_idx` (`id_aluno`),
  KEY `fk_oferta_idx` (`id_oferta`),

```

```

    CONSTRAINT `fk_aluno` FOREIGN KEY (`id_aluno`) REFERENCES `aluno` (`id`) ON
DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT `fk_oferta` FOREIGN KEY (`id_oferta`) REFERENCES `oferta` (`id`)
ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `colocado_oferta` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `oferta_id` int(11) DEFAULT NULL,
  `aluno_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_colocado_oferta_oferta_idx` (`oferta_id`),
  KEY `fk_colocado_oferta_aluno_idx` (`aluno_id`),
  CONSTRAINT `fk_colocado_oferta_aluno` FOREIGN KEY (`aluno_id`) REFERENCES
`aluno` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_colocado_oferta_oferta` FOREIGN KEY (`oferta_id`) REFERENCES
`oferta` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=25 DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `concelho` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nome` varchar(45) DEFAULT NULL,
  `codigo` varchar(45) NOT NULL,
  `distrito_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `unique_concelho` (`codigo`,`distrito_id`),
  KEY `fk_concelho_distrito1_idx` (`distrito_id`),
  CONSTRAINT `fk_concelho_distrito1` FOREIGN KEY (`distrito_id`) REFERENCES
`distrito` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=38333 DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `curso` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nome` varchar(120) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `distrito` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nome` varchar(45) NOT NULL,
  `codigo` varchar(45) NOT NULL,
  `pais_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `unique_distrito` (`codigo`,`pais_id`),
  KEY `fk_distrito_pais1_idx` (`pais_id`),
  CONSTRAINT `fk_distrito_pais1` FOREIGN KEY (`pais_id`) REFERENCES `pais`
(`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=3915 DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `empresa` (

```

```

`id` int(11) NOT NULL AUTO_INCREMENT,
`nif` int(11) DEFAULT NULL,
`designacao` varchar(55) DEFAULT NULL,
`ramo_negocio` varchar(200) DEFAULT NULL,
`cae` varchar(20) DEFAULT NULL,
`num_empregados` int(11) DEFAULT NULL,
`concelho_id` int(11) DEFAULT NULL,
`morada` varchar(1000) DEFAULT NULL,
`codigo_postal` varchar(100) DEFAULT NULL,
`pessoa_contacto` varchar(500) DEFAULT NULL,
`cargo_pessoa_contacto` varchar(100) DEFAULT NULL,
`telefone` varchar(30) DEFAULT NULL,
`e_mail` varchar(100) DEFAULT NULL,
`web` varchar(100) DEFAULT NULL,
`data_registo` date DEFAULT NULL,
`apresentacao_empresa` varchar(2000) DEFAULT NULL,
`id_utilizador` int(11) DEFAULT NULL,
`login_utilizador` varchar(100) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `concelho_id` (`concelho_id`),
CONSTRAINT `empresa_ibfk_1` FOREIGN KEY (`concelho_id`) REFERENCES
`concelho` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;

CREATE TABLE `escola` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `escola` varchar(5) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `ficheiro` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) DEFAULT NULL,
  `ficheiro` varchar(45) DEFAULT NULL,
  `nome` varchar(45) DEFAULT NULL,
  `public` int(1) DEFAULT '0',
  `id_oferta` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_ficheiro_user_idx` (`user_id`),
  KEY `fk_ficheiro_oferta_idx` (`id_oferta`),
  CONSTRAINT `fk_ficheiro_aluno` FOREIGN KEY (`user_id`) REFERENCES `user`
(`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_ficheiro_oferta` FOREIGN KEY (`id_oferta`) REFERENCES
`oferta` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=193 DEFAULT CHARSET=latin1;

CREATE TABLE `inquerito` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_aluno` int(11) DEFAULT NULL,
  `codigo_aluno` int(11) DEFAULT NULL,

```

```

`q1` varchar(45) DEFAULT NULL,
`q2` varchar(15) DEFAULT NULL,
`q21` varchar(45) DEFAULT NULL,
`q22` varchar(15) DEFAULT NULL,
`q23` varchar(45) DEFAULT NULL,
`q3` varchar(45) DEFAULT NULL,
`q4` varchar(15) DEFAULT NULL,
`q41` varchar(30) DEFAULT NULL,
`q42` varchar(45) DEFAULT NULL,
`q5` varchar(20) DEFAULT NULL,
`ano` int(11) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `fk_aluno_idx` (`id_aluno`),
KEY `fk_codigo_aluno_idx` (`codigo_aluno`),
CONSTRAINT `fk_aluno_inquerito` FOREIGN KEY (`id_aluno`) REFERENCES `aluno`
(`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `fk_codigo_aluno` FOREIGN KEY (`codigo_aluno`) REFERENCES
`aluno` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `inqueritoEmpresa` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `ano` int(11) DEFAULT NULL,
  `id_empresa` int(11) DEFAULT NULL,
  `login_empresa` varchar(45) DEFAULT NULL,
  `contactoAlunos` int(11) DEFAULT NULL,
  `recrutamento1` int(11) DEFAULT NULL,
  `satisfeito` int(11) DEFAULT NULL,
  `recrutamento2` int(11) DEFAULT NULL,
  `recrutamento3` int(11) DEFAULT NULL,
  `criatividade` int(11) DEFAULT NULL,
  `flexibilidade` int(11) DEFAULT NULL,
  `aprendizagem` int(11) DEFAULT NULL,
  `conceptualizar` int(11) DEFAULT NULL,
  `equipa` int(11) DEFAULT NULL,
  `motivacao` int(11) DEFAULT NULL,
  `diversidade` int(11) DEFAULT NULL,
  `etica` int(11) DEFAULT NULL,
  `lideranca` int(11) DEFAULT NULL,
  `linguas` int(11) DEFAULT NULL,
  `tecnologias` int(11) DEFAULT NULL,
  `conhecimento` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_empresa_idx` (`id_empresa`),
  CONSTRAINT `fk_empresa` FOREIGN KEY (`id_empresa`) REFERENCES `empresa`
(`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `inqueritoNovoAluno` (
  `id` int(11) NOT NULL AUTO_INCREMENT,

```

```

`id_aluno` int(11) DEFAULT NULL,
`codigo_aluno` varchar(15) DEFAULT NULL,
`q1` varchar(100) DEFAULT NULL,
`q2` varchar(100) DEFAULT NULL,
`ano` int(11) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `fk_aluno_idx` (`id_aluno`),
CONSTRAINT `fk_aluno_inqueritoNovoAluno` FOREIGN KEY (`id_aluno`)
REFERENCES `aluno` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `inqueritoSatisfacao` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`id_aluno` int(11) DEFAULT NULL,
`codigo_aluno` int(11) DEFAULT NULL,
`q1` int(11) DEFAULT NULL,
`q2` int(11) DEFAULT NULL,
`q3` int(11) DEFAULT NULL,
`q4` int(11) DEFAULT NULL,
`q5` int(11) DEFAULT NULL,
`q6` int(11) DEFAULT NULL,
`q7` int(11) DEFAULT NULL,
`ano` int(11) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `fk_aluno_idx` (`id_aluno`),
KEY `fk_codigo_aluno_idx` (`codigo_aluno`),
CONSTRAINT `fk_aluno_inqueritoSatisfacao` FOREIGN KEY (`id_aluno`)
REFERENCES `aluno` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `fk_codigo_aluno_inqueritoSatisfacao` FOREIGN KEY
(`codigo_aluno`) REFERENCES `aluno` (`id`) ON DELETE NO ACTION ON UPDATE NO
ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `migration` (
`version` varchar(180) NOT NULL,
`apply_time` int(11) DEFAULT NULL,
PRIMARY KEY (`version`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `oferta` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`id_empresa_oferta` int(11) DEFAULT NULL,
`dataOferta` date DEFAULT NULL,
`estadoOferta` varchar(200) DEFAULT NULL,
`descricaoOferta` varchar(1000) DEFAULT NULL,
`tituloOferta` varchar(500) DEFAULT NULL,
`numVagas` int(11) DEFAULT '1',
`requisitosMinimos` varchar(1000) DEFAULT NULL,
`requisitosPretendidos` varchar(1000) DEFAULT NULL,
`duracao_contrato` varchar(100) DEFAULT NULL,

```

```

`comissoesIncentivos` varchar(1000) DEFAULT NULL,
`cursosOferta1` varchar(400) DEFAULT NULL,
`tipoContrato` varchar(200) DEFAULT NULL,
`localTrabalho` varchar(200) DEFAULT NULL,
`categoriaProfissional` varchar(200) DEFAULT NULL,
`salario` decimal(10,2) DEFAULT NULL,
`horarioTrabalho` varchar(1000) DEFAULT NULL,
`regime` varchar(200) DEFAULT NULL,
`cursosOferta2` varchar(400) DEFAULT NULL,
`cursosOferta3` varchar(400) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `fk_empresa_oferta_idx` (`id_empresa_oferta`),
CONSTRAINT `fk_empresa_oferta` FOREIGN KEY (`id_empresa_oferta`) REFERENCES
`empresa` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=56 DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `ofertaObservatorio` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`id_empresa_oferta` int(11) DEFAULT NULL,
`dataOferta` date DEFAULT NULL,
`tituloOferta` varchar(500) DEFAULT NULL,
`numVagas` int(11) DEFAULT NULL,
`estudosMinimos` varchar(1000) DEFAULT NULL,
`experienciaMinima` varchar(1000) DEFAULT NULL,
`requisitosMinimos` varchar(1000) DEFAULT NULL,
`requisitosPretendidos` varchar(1000) DEFAULT NULL,
`duracaoContrato` varchar(100) DEFAULT NULL,
`comissoesIncentivos` varchar(1000) DEFAULT NULL,
`cursosOferta` varchar(1000) DEFAULT NULL,
`tipoContrato` varchar(200) DEFAULT NULL,
`localTrabalho` varchar(200) DEFAULT NULL,
`categoriaProfissional` varchar(200) DEFAULT NULL,
`salario` int(11) DEFAULT NULL,
`descricaoOferta` varchar(1000) DEFAULT NULL,
`horarioTrabalho` varchar(1000) DEFAULT NULL,
`regime` varchar(200) DEFAULT NULL,
`estadoOferta` varchar(200) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `fk_empresa_oferta_idx` (`id_empresa_oferta`),
CONSTRAINT `fk_empresa_ofertaObservatorio` FOREIGN KEY
(`id_empresa_oferta`) REFERENCES `empresa` (`id`) ON DELETE NO ACTION ON
UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `pais` (
`id` int(11) NOT NULL,
`nome` varchar(45) DEFAULT NULL,
`codigo` varchar(45) DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `trabalhador` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `ano` int(11) DEFAULT NULL,
  `escola` varchar(15) DEFAULT NULL,
  `numTrabalhadores` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) NOT NULL,
  `auth_key` varchar(32) NOT NULL,
  `password_hash` varchar(255) NOT NULL,
  `confirmation_token` varchar(255) DEFAULT NULL,
  `status` int(11) NOT NULL DEFAULT '1',
  `superadmin` smallint(6) DEFAULT '0',
  `created_at` int(11) NOT NULL,
  `updated_at` int(11) NOT NULL,
  `registration_ip` varchar(15) DEFAULT NULL,
  `bind_to_ip` varchar(255) DEFAULT NULL,
  `email` varchar(128) DEFAULT NULL,
  `email_confirmed` smallint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=27 DEFAULT CHARSET=utf8;

CREATE TABLE `user_visit_log` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `token` varchar(255) NOT NULL,
  `ip` varchar(15) NOT NULL,
  `language` char(2) NOT NULL,
  `user_agent` varchar(255) NOT NULL,
  `user_id` int(11) DEFAULT NULL,
  `visit_time` int(11) NOT NULL,
  `browser` varchar(30) DEFAULT NULL,
  `os` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `user_id` (`user_id`),
  CONSTRAINT `user_visit_log_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES
`user` (`id`) ON DELETE SET NULL ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=635 DEFAULT CHARSET=utf8;

```