**INSTITUTO POLITÉCNICO DE BEJA**

**Escola Superior de Tecnologia e Gestão**

**Curso de Engenharia Informática**

**Realização de Sistema de Gestão de Distribuições de Serviço**

**Docente – Parte II**

**Carlos Alexandre Rijo Palma**

**Beja**

**2012**

**INSTITUTO POLITÉCNICO DE BEJA**

**Escola Superior de Tecnologia e Gestão**

**Curso de Engenharia Informática**

**Realização de Sistema de Gestão de Distribuições de Serviço**

**Docente – Parte II**

**Relatório de estágio, realizado na empresa Instituto Politécnico de Beja apresentado na Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Beja**

**Elaborado por:**

**Carlos Alexandre Rijo Palma Nº5608**

**Orientado por:**

**José Jasnau Caeiro**

**Beja**

**2012**

Resumo

Este Relatório refere-se ao estágio curricular titulado, "Realização de Sistema de Gestão de Distribuições de Serviço Docente – Parte II" no âmbito da Licenciatura em Engenharia Informática. O projecto visa a realização de uma aplicação baseada na Web para a Gestão de Distribuições de Serviço Docente.

A Web Framework foi baseada na linguagem de programação python: Django, para o desenvolvimento do sistema.

O trabalho desenvolvido repartiu-se em duas partes. Numa primeira parte os esforços foram centrados na concepção dos modelos e controlo do Sistema, já na segunda parte os esforços foram centrados na concepção das vistas e controlo do Sistema.

**Índice**

Lista de figuras

Lista de abreviaturas

# Introdução

* intro
* Trabalho do actor
* Estrutura do Relatório

# Sistema de Gestão de Distribuições de Serviço Docente

## Sumário

A gestão de distribuições de serviço docente é uma função árdua no funcionamento das instituições do ensino superior. Sendo do interesse das instituições que essa função seja realizada de uma forma simples e atempadamente, de modo a diminuir os tempos de produção dos horários e tornar mais eficaz e transparente a gestão dos Recursos Humanos. Importa referir que, as distribuições de serviço docente envolvem os departamentos de recursos humanos e os departamentos de ensino das instituições, os concelhos científicos e a gestão das instituições.

## Tecnologia de desenvolvimento

Tal como definido no início do Estágio, a Aplicação web a desenvolver, é baseada na linguagem de programação Python, uma linguagem de alto nível, moderna, com um grande padrão de bibliotecas e um conjunto de ferramentas de desenvolvimento muito sofisticadas, importa ainda referir que é uma linguagem orientada a obcjetos.

Uma forma simples ao desenvolvimento de uma Aplicação Web em Python é a utilização de Frameworks. Uma Framework pode ser definida como um conjunto de classes e bibliotecas com o objetivo de facilitar o programador, no que diz respeito a economia de tempo e segurança no desenvolvimento de software, proporcionando assim a abstração de aspetos da metodologia e da Linguagem de Programação a ser utilizada, sobretudo em código partilhado por todas as aplicações, e assim apresentando uma arquitetura de software bem organizada.

No que diz respeito ao desenvolvimento de software é o ato de elaborar e implementar o um sistema computacional. Importa referir que, a arquitetura de software, mais utilizada é o MVC. O MVC (Model-View-Controller) é um padrão de desenho de software baseado na tecnologia de orientação por objetos, fazendo a separação, entre o acesso aos dados e a lógica do negócio, e entre a lógica de apresentação e a interação com o utilizador. A arquitetura da aplicação MVC está dividida em três camadas, o *Model* manipula as regras de negócio e de acesso aos dados, a *View* disponibiliza as interfaces que possibilitam a interacção com o utilizador, por fim a camada intitulada por *Controller* responsável por controlar o fluxo da aplicação entre o Model e a View, processa e responde a eventos, invoca alterações no modelo, assim sendo é o local ideal para fazer a validação e filtragem dos dados dos utilizadores.

Na aplicação desenvolvida, foi utilizada a Web Framework o Django, uma framework de alto nível desenvolvida em Python, centrada no desenvolvimento rápido e pragmático de aplicações Web e utiliza a arquitetura MVC, com uma pequena alteração.

No Django o MVC pode ser denominado de MTV (Model-Templates-View). A View deixa de fazer a interação com o utilizador e não vai conter o código responsável por criar a interface para o utilizador (HTML, CSS, JavaScript e AJAX) e passa a ser responsável pela recolha dos dados a apresentar. Já a tarefa de criar as interfaces e fazer a interação com o utilizador é da responsabilidade dos *Templates*. O Controller é reconhecido como a própria framework, que de acordo com a configuração de URL´s, encaminha para a View adequada os pedidos efetuados.

Os Templates do Django são normalmente ficheiros HTML, com uma sintaxe própria do motor de Templates do Django, facilitando o processo de desenvolvimento e atingindo o objetivo, ou seja, a divisão entre os dados que são mostrados e a forma como os mesmos são apresentados.

### Funcionamento do Django

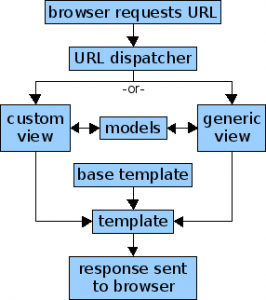


Figura 2‑1-MVT (Model-Template-View) do Django

Importa referir que, ainda foi utilizado um conjunto de bibliotecas, tituladas de Dojo Toolkit na versão 1.8, com o objetivo de facilitar o desenvolvimento da Aplicação. O Dojo Toolkit fornece um grande suporte de ferramentas, economizando tempo no processo de desenvolvimento de aplicações e sítios Web.

O Sistema de Base Dados utilizado no desenvolvimento da aplicação foi o Sqlite3. O Sqlite3 é uma biblioteca desenvolvida através da linguagem de programação C, que permite ser integrada em programas de diferentes linguagens de programação. Na prática, o Sqlite funciona como um pequeno SGBD (Sistema de Gestão de Base Dados), capaz de criar um arquivo em disco e ler e escrever diretamente sobre esse arquivo. O arquivo criado possui a extensão ".db". A Base Dados foi criada através de código implementado no ficheiro "models.py", e utilizado uma tecnologia de migrações, ou seja, sempre que existe alguma alteração feita no ficheiro "models.py" é gravada automaticamente uma migração que corresponde a alteração efetuada. A tecnologia de migração utilizada foi o South, uma tecnologia que tem como principal objetivo fornecer uma camada simples e estável, independentemente da Base Dados utilizada. Atualmente o South é considerado a mais popular ferramenta de migração de Base Dados para o Django.

No desenvolvimento da aplicação foi utilizada o Mercurial. Uma ferramenta multiplataforma de controlo de versões, concedido para programadores de software. Ferramenta que inicialmente foi criada para ser utilizada no Linux, mas atualmente já pode ser também utilizada em outros sistemas operativos, como por exemplo, o Windows e Mac. Permite que o utilizador guarde num repositório as várias versões do seu software. O "repositório central" utilizado na Aplicação desenvolvida, foi o Google Code. O Goolgle Code é um site da companhia Google, que contém código-fonte aberto e uma lista de serviços que suportam a API pública do Google.

**Funcionamento do Mercurial**

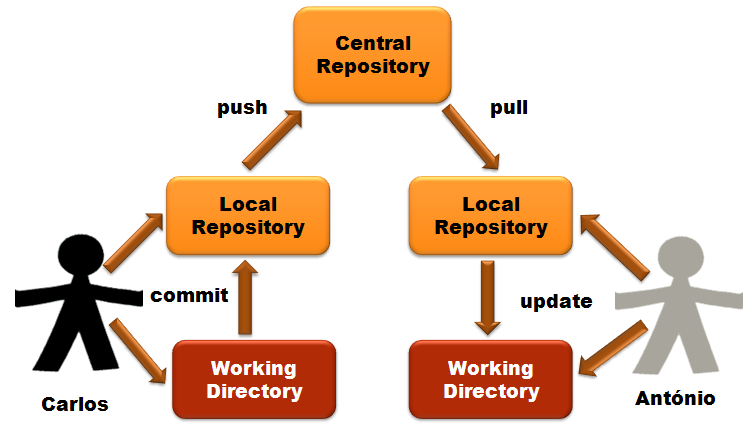
****

Figura 2‑2 - Funcionamento do Mercurial

# Fase de Implementação do Sistema

## Sumário

Este capítulo tem como principal objetivo explicar de uma forma sucinta como foi elaborada a implementação do Sistema de Gestão de Distribuições de Serviço Docente, mais propriamente duas das camadas da arquitetura MVC, sendo elas o Model e a View.

Conforme já referenciado no capítulo anterior, o Sistema foi implementado através da linguagem de programação Python utilizando a Framework do Django.

## Model

O Model corresponde a primeira camada da arquitetura MVC, que tem o objetivo de ser a camada por manipular as regras de negócio e de acesso aos dados do Sistema. Com a Framework do Django, as tabelas que vão originar a Base Dados são criadas através de Classes de Python que estendem de django.db.models.Model, em que cada atributo da classe representa um campo da tabela.

O Django disponibiliza uma API de acesso a Base Dados dinâmica que permite criar, alterar e aceder a uma Base Dados através da simples definição da mesma no Model, sem haver a necessidade ter conhecimentos de SQL.

### Base Dados do Sistema

Como já referido neste relatório, a Base Dados do Sistema de Gestão de Distribuições de Serviço Docente, foi criada através de código implementado, num módulo de Python, com o nome de "models.py". Nesse módulo foram definidas várias classes, ao qual as classes originaram as tabelas da Base Dados, já os campos das tabelas são as variáveis definidas em cada classe.

Por forma a tornar mais claro a criação de uma tabela segue um exemplo a baixo.

01 **class** **Teste(**models**.**Model**):**

02 '''

03 Teste - Class de Exemplo de criação de tabela

04 '''

05 nome **=** models**.**CharField**(**max\_length**=**80**,** unique**=**True**)**

06 Morada **=** models**.**CharField**(**max\_length**=**100**,**unique**=**True**)**

07

08 **def** \_\_unicode\_\_**(**self**):**

09 #return self.nome

10 **return** self**.**nome

Após desenvolver as classes da criação das tabelas, foi necessário fazer o preenchimento de algumas tabelas, através de um documento XLS, documento esse onde é feita a Gestão de Distribuição Docente atualmente. Para tornar o preenchimento das tabelas de uma forma rápida e precisa, foi necessário desenvolver uns métodos implementados em Python de leitura de ficheiros XLS e uns métodos de escrita em ficheiros YAML. Sendo que o preenchimento das tabelas foi efetuado através desses ficheiros YAML.

**Modelo Físico**

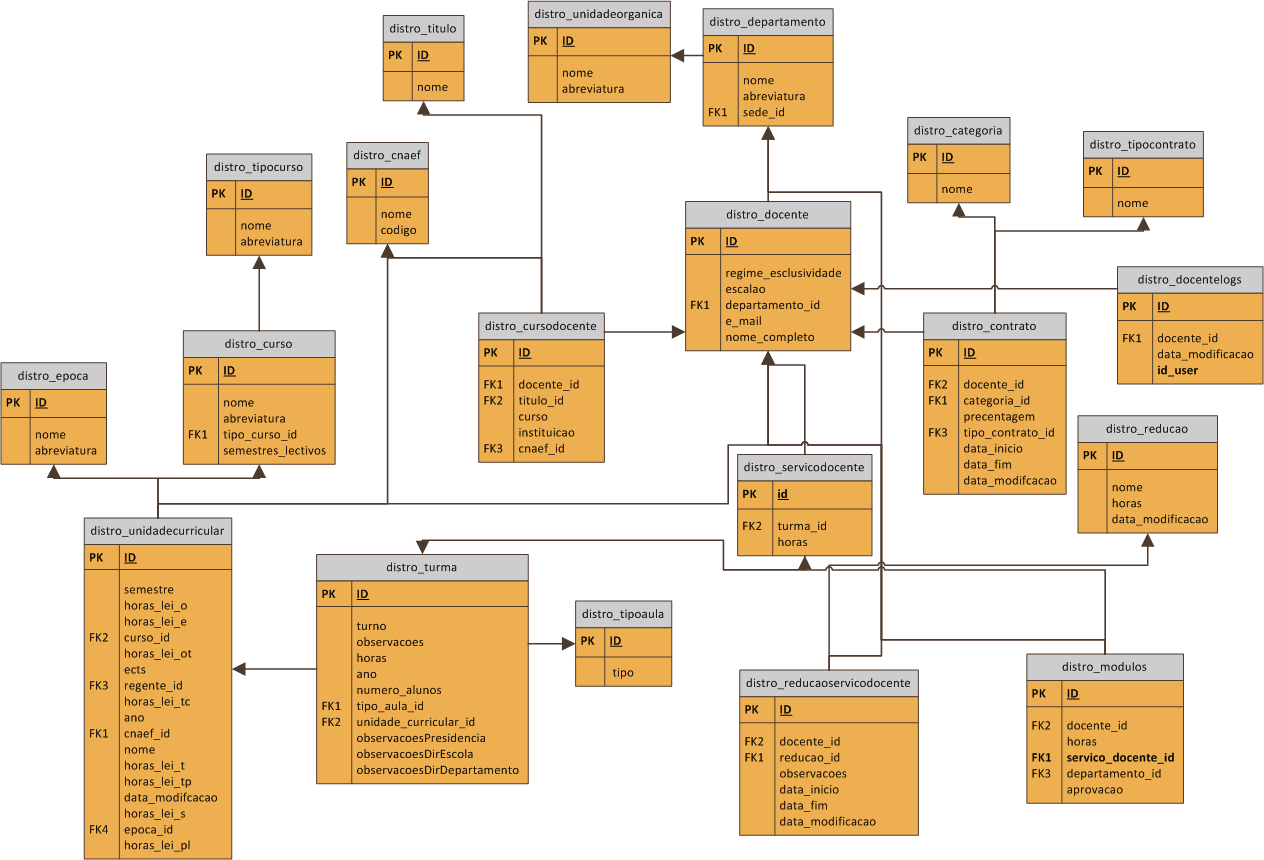


Figura 2‑3 - Modelo Físico da Base Dados do Sistema

## Base Dados de Autenticação para o Sistema

De forma a separar os dados do Sistema de Gestão de Distribuições de Serviço Docente e os dados dos utilizadores do Sistema, foi utilizada uma Base Dados de autenticação. A Base Dados de autenticação utilizada foi a do Django, uma Base Dados composta por sete tabelas as quais permitem guardar os dados dos utilizadores do sistema, dividir os utilizadores em grupos e atribuir permissões a cada grupo. Para a utilização dessa Base Dados foi necessário fazer a instalação do pacote de suporte de autenticação do Django. Onde foi necessário referir no ficheiro "settings.py" a seguinte expressão "django.contrib.auth", mais propriamente na configuração " INSTALLED\_APPS", em seguida foi necessário fazer a instalação do pacote de suporte de autenticação através da execução do comando que se segue, "python manage.py syncdb".

**Modelo Físico**

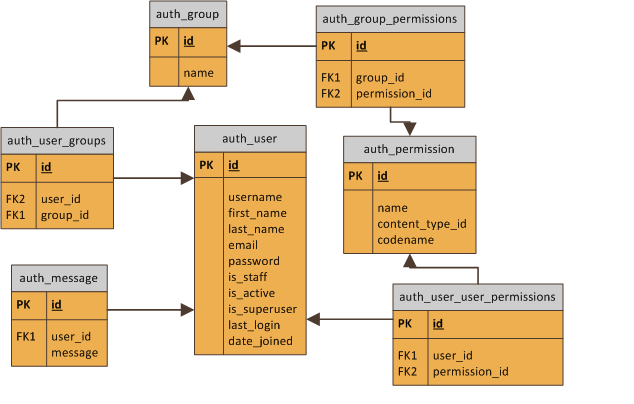


Figura 2‑4 - Modelo Físico da Base Dados de Autenticação

## View

A View corresponde a segunda camada da arquitetura MVC, que na visão do Django, tem o objetivo de ser a camada responsável pela recolha dos dados que vão ser apresentados ao utilizador, deixando assim de fazer a interação com o utilizador e também deixando de ter o código responsável por criar a interface para o utilizador (HTML, CSS, JavaScript e AJAX), como geralmente é feito em outras Frameworks.

Na implementação deste Sistema, para cada utilizador foi criado uma view diferente, ou seja, de forma a fazer uma separação do código para cada utilizador, fazendo com que este tenha uma leitura simples e explicita na visão do programador. Cada view é composta por métodos criados na linguagem Python, que normalmente recebem como parâmetro o pedido URL (request) e retornam uma resposta para o Template desejado.

VIEW\_DEPARTAMENTO - Explicar os métodos.

### Departamentos…… ->

# Guia de Utilização do Sistema

## Sumário

O Sistema de Gestão de Distribuições de Serviço Docente prevê vários tipos de utilizadores, sendo eles, docentes, recursos humanos, departamentos de ensino, diretores de escola, coordenadores de curso, entre outos. No que diz respeito ao utilizador diretores de departamento, assume o papel de criar o serviço docente, ver os docentes, ver as turmas, ver os serviços docentes atribuídos do seu departamento.

Ao aceder ao sistema, é apresentado ao utilizador uma página inicial, que permite fazer a autenticação, para o utilizador entre na sua área.



Figura 3‑1 - Ecrã de Autenticação no Sistema

## Utilizador Departamento de Engenharia

**Ecrã Principal**

Depois de o utilizador ter efectuado a autenticação com o sucesso, é apresentado a página inicial do departamento de Engenharia. No lado esquerdo da página o utilizador tem um Menu com todas as opções que lhe são permitidas efectuar, sendo elas: "Informação", "Listar Turmas", "Listar Docentes", "Listar Serviços Docentes Atribuídos", "Atribuir Serviço Docente" e por fim a "Ajuda". No Centro da Página é apresentado algumas estatísticas importantes para o utilizador.



Figura 3‑2- Ecrã Principal do Departamento Engenharia do Sistema

**Ecrã Listar Turmas no Ano 2013**

Após passar o rato sobre o "Listar Turmas" aparece uma opção com a lista de Anos que o utilizador pode escolher. Ao clicar na opção "2013" são apresentadas a informação de todas as turmas do departamento de Engenharia no Ano 2013. Importa referir que no lado esquerdo do ecrã o utilizador tem a possibilidade de filtrar as turmas por ordem alfabética e por curso, e no final do ecrã o utilizador também tem a possibilidade de mudar de página, consoante o número de turmas existentes.

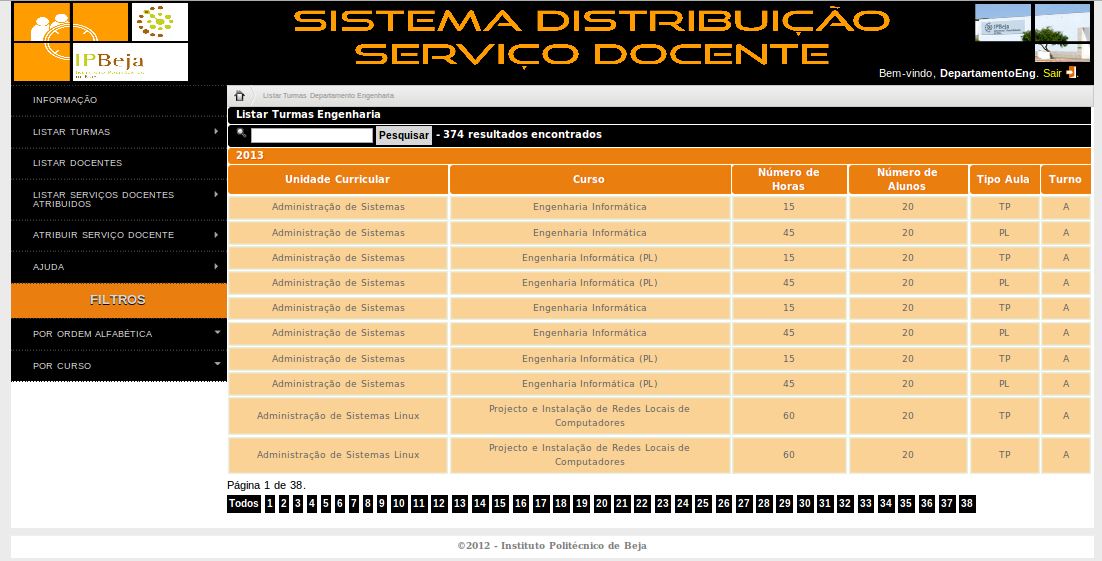


Figura 3‑3 - Ecrã Listar Turmas 2013 do Departamento Engenharia do Sistema

**Ecrã Listar Docentes**

Ao clicar na opção "Listar Docentes" é apresentado a informação dos docentes do departamento de Engenharia, mais propriamente o Nome, as horas distribuídas e o número de turmas que pertence. Neste ecrã existem também uma secção de filtros, onde pode ser filtrados os docentes por ordem alfabética e por categoria. Importa referir que os nomes dos docentes aparecem com *links* para apresentar uma nova página com a Informação do docente em questão, mais propriamente as unidades curriculares que o docente tem atribuído para lecionar.

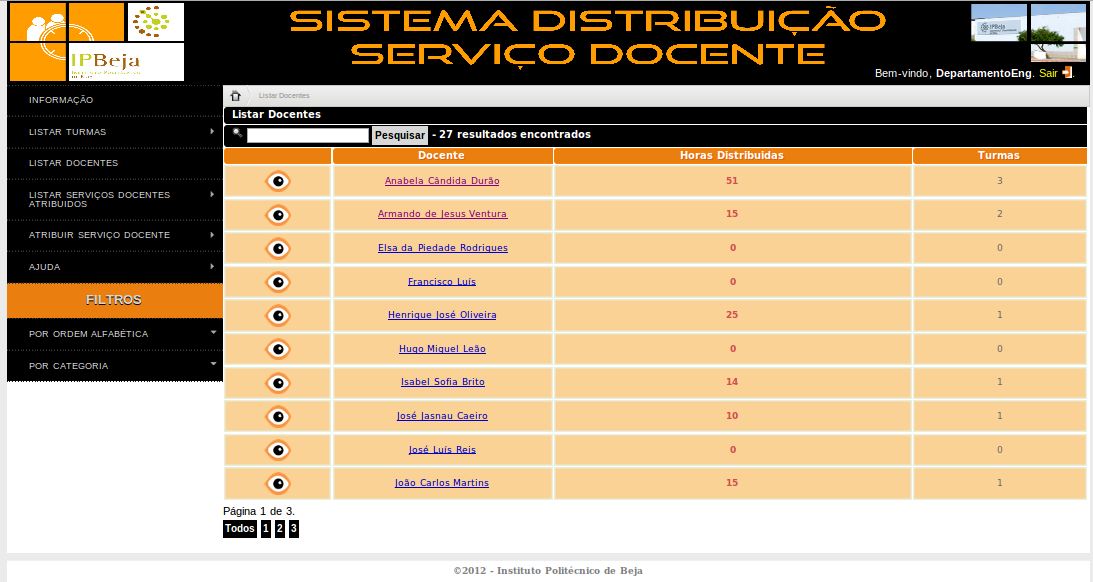


Figura 3‑4 - Ecrã Listar Docentes do Departamento Engenharia do Sistema

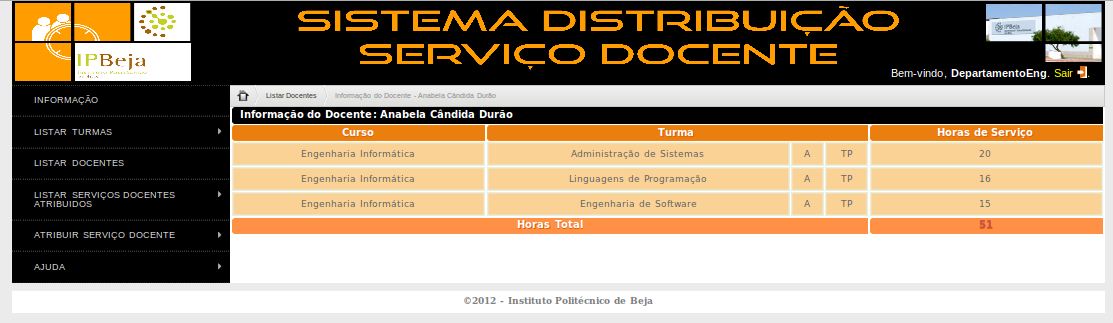


Figura 3‑5 - Ecrã Informação Docente do Departamento Engenharia do Sistema

**Ecrã Listar Serviços Docentes Atribuídos**

No Menu de opções do lado esquerdo do ecrã ao passar o rato sobre "Listar Serviço Docentes Atribuídos", aparece uma opção com a lista de Anos que o utilizador pode escolher. Ao clicar na opção "2013", são apresentadas as turmas que já têm docentes atribuídos para lecionar a mesma. Aqui também o utilizador tem a possibilidade de filtrar as turmas por ordem alfabética e por curso. No nome das unidades curriculares existem *links* para uma nova página, onde pode se pode ver o docente ou os docentes que vão lecionar a unidade curricular.

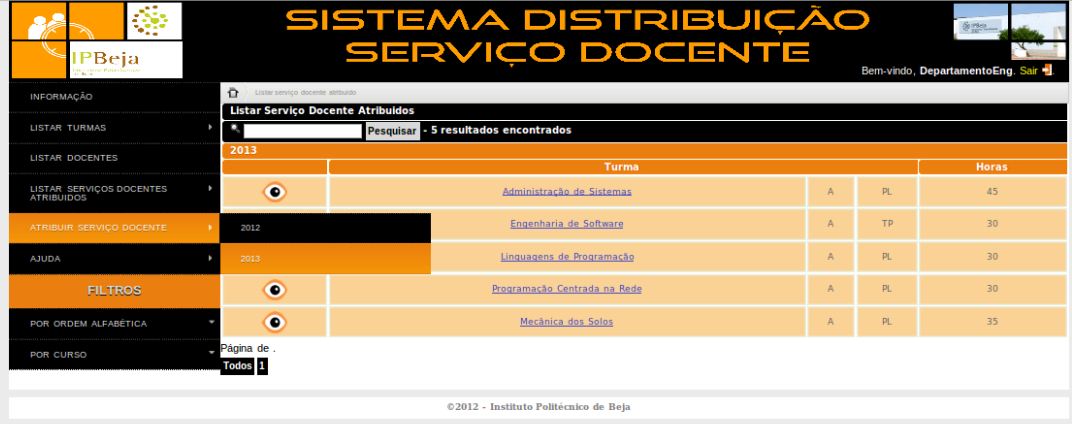


Figura 3‑6- Ecrã Listar Serviço Docente Atribuído do Departamento Engenharia do Sistema

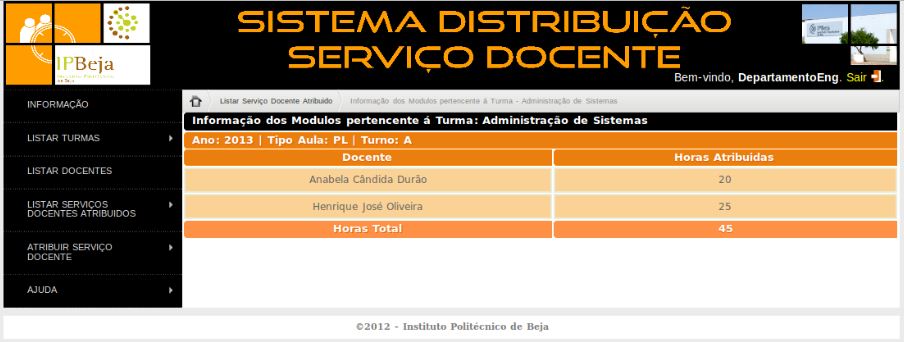


Figura 3‑7- Ecrã Informação Turma com docente atribuído do Departamento Engenharia do Sistema

**Ecrã Atribuir Serviços Docentes**

O utilizador neste ecrã pode atribuir docentes a uma turma. No Menu do lado esquerdo do ecrã ao passar o rato sobre "Atribuir Serviço Docente", aparece uma opção com a lista de Anos que o utilizador pode escolher. Ao clicar na opção "2013", são apresentadas as turmas que ainda não têm docentes atribuídos.



Figura 3‑8 - Ecrã Atribuir Serviço Docente do Departamento Engenharia do Sistema

Após clicar no *link* da turma para atribuir o docente, é apresentado ao utilizador um formulário onde o mesmo pode atribuir aos módulos os docente que são apresentados numa lista de docentes, ou delegar os módulos a outros departamentos, para que esses módulos sejam lecionados por docentes pertencentes a outros departamentos.



Figura 3‑9 - Ecrã Com o formulário de Atribuição Serviço Docente do Departamento Engenharia do Sistema

Após o utilizador efetuar as operações desejadas, pode submeter os dados, clicando na opção "Adicionar Serviço docente", onde vai aparecer uma nova página para confirmar as alterações efetuadas.



Figura 3‑10 - Ecrã Confirmação do formulário de Atribuição Serviço Docente do Departamento Engenharia do Sistema

Depois de confirmar o envio dos dados do formulário de atribuição, o Sistema informa o utilizador que os dados foram ou não submetidos com Sucesso.

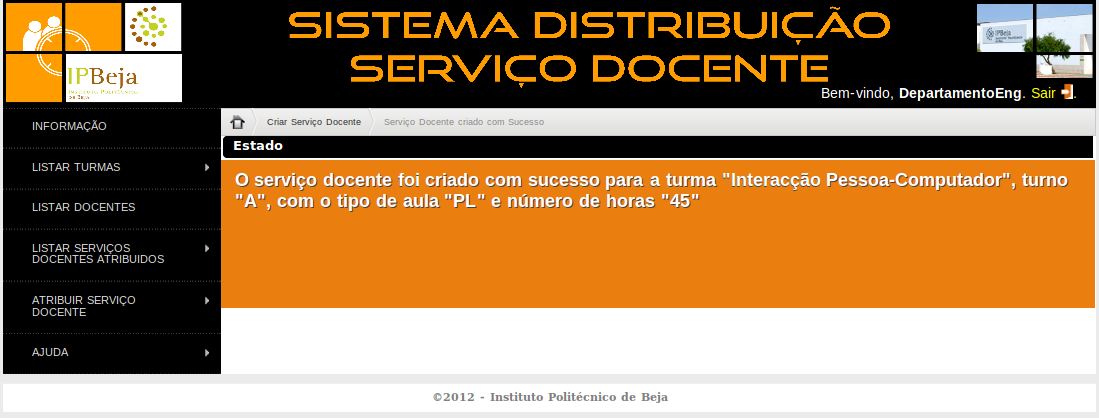


Figura 3‑11 - Ecrã dos dados Submetidos com sucesso no Sistema

# Conclusões Finais

Bibliografia