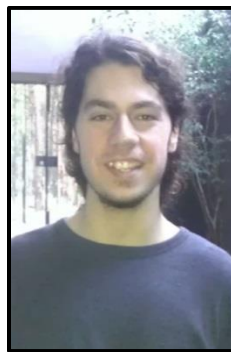


**Instituto Politécnico de Beja**

**Projecto de Análise de Inscrições no  
Ensino Superior desenvolvido em Python**



**Jorge Emanuel Candeias Coveiro, nº 10873**

**Carlos Eduardo Rosário, nº 12794**

**02 de Dezembro de 2012**

# 1. Introdução

Este primeiro trabalho foi realizado no âmbito da unidade curricular de Linguagens de Programação do primeiro semestre do segundo ano do curso de Engenharia Informática da ESTIG, pelos alunos Jorge Coveiro e Carlos Rosário.

Contextualiza-se no prosseguimento da aprendizagem de duas linguagens de programação: Python e C#, tendo esta primeira parte sido desenvolvida na primeira linguagem referida com o objectivo analisar a variação de inscrições nas diversas áreas e níveis de estudo ao longo dos passados anos lectivos (de 1995 a 2011).

O relatório está repartido por uma breve introdução teórica, parte experimental e conclusão. Na parte teórica fazemos uma pequena abordagem da linguagem Python entre as diferentes linguagens de programação. Na parte experimental abordamos a realização do trabalho, os sistemas adoptados e os resultados práticos. Finalmente na conclusão fazemos uma retrospectiva de todo o trabalho e discutem-se melhorias do mesmo.

## 2. Teoria

Todas as linguagens de programação têm o mesmo poder computacional de uma *Máquina de Turing*, o que difere é a forma de como o código é escrito.

Estas são caracterizadas por uma forma (*síntaxe*) e significado (*semântica*). A *síntaxe* é usada para referir o estudo das regras que regem o comportamento de sistemas matemáticos como a lógica ou as linguagens de programação, enquanto a *semântica* estuda o significado para se expressar através de uma linguagem. A *semântica* pode ser *estática* ou *dinâmica*. *Estática* é o que acontece em tempo de compilação. *Dinâmica* é o que acontece em tempo de execução.

Em linguagens de programação começamos por abordar os diferentes *paradigmas computacionais*: imperativo (e.g. C), lógico ou declarativo (e.g. Prolog), funcionais (e.g. Haskell, Lisp) e object oriented. No entanto muitas suportam vários paradigmas e são assim chamadas de multiparadigmáticas (e.g. Java, Python).

A programação imperativa descreve a computação como acções, enunciados ou comandos que mudam o estado de um programa. Os paradigmas lógicos e funcionais situam-se muito perto das matemáticas enquanto as object oriented são muito parecidas às imperativas.

Numa linguagem lógica parece mais que se está a descrever um problema e as suas propriedades do que propriamente dizer ao computador como realizar os cálculos. Em vez de “tira um elemento da lista x”, diz-se “existe uma lista que é igual à lista x, mas tem menos um elemento”.

Nas linguagens funcionais abdica-se de tudo aquilo que forme efeitos colaterais, ou seja, altere a estrutura dos dados originais.

## 3. Parte Experimental

### 3.1 Realização Experimental

Neste trabalho usamos a linguagem Python porque não tivemos opção de escolha.

Utilizamos também os seguintes portáteis com o seu respectivo sistema operativo:

- ➔ Laptop Acer Intel i3-370M@2,4GHz + Ubuntu Linux (32bit)
- ➔ Laptop Acer AMD AthlonX2@1,2GHz + Linux Mint Cinnamon (64bit)

Para programar utilizamos vários ambientes de desenvolvimento e editores de texto entre os quais: Ninja-IDE, Sublime-Text-2, Qt4 Designer,

### 3.2 Sistema Experimental

O trabalho é constituído por 3 packages: o core, o dataplot e o gui e também um ficheiro main.py que é o cerne do programa que utiliza estes packages e o ficheiro Inscritos\_2010-2011.xls.

O package core contém 3 módulos: o dbhandler.py, o xlshandler.py e o curso.py.

O módulo dbhandler.py contém a classe DbHandler que opera sobre a base de dados. O módulo xlshandler.py contém a classe XlsHandler que opera sobre o ficheiro Inscritos\_2010-2011.xls. Por último o módulo curso.py que contém a classe Curso que é utilizada pela classe DbHandler para o uso posterior na criação dos gráficos.

O package dataplot contém também 3 módulos: o matplotlib\_bars.py, o matplotlib\_bars3d.py e o matplotlib\_lines.py. Estes 3 módulos criam gráficos de barras ou linhas com a informação obtida em arrays de objectos da classe Curso.

Finalmente o package gui contém módulos para a interface gráfica do nosso projecto.

### 3.3 Resultados Experimentais