Classes Implementadas:

Classe User:

Esta é a classe que define as características de um utilizador. Estas características são: o nome de utilizador, o endereço de correio electrónico (e-mail) e a password respectiva. O utilizador necessita da sua password de cada vez que pretender utilizar o software.

Classe Software:

É esta a classe usada para definir um software que conste da base de dados. Esta possuí três variáveis: id (que é um valor único e é atribuído a cada software novo que é adicionado), nome e o link do seu website. Depois também possui uma estrutura, nomeadamente um Dictionary, que contém as suas características. A chave do Dictionary é o id da Característica (ver Classe Característica) e o seu valor respectivo.

Classe Characteristic:

A classe Characteristic é uma classe abstracta (pois nunca existirão instâncias dela) e também uma super-classe dos 3 tipos de características: características qualitativas, quantitativas e booleanas (ver QualitativeCharacteristic, NumericCharacteristic e YesNoCharacteristic, respectivamente). Esta define, portanto, o que é que os 3 tipo de características possuem em comum. Estas possuem um valor númerico denominado id, valor único a cada característica e o seu nome.

Classe QualitativeCharacteristic:

Esta classe define o tipo de característica que é constituído por valores qualitativos (como por exemplo: Mau, Razoável, Muito Bom). A cada palavra é atribuído o valor respectivo (definido na classe Value) e esses dados são armazenados num Dictionary denominado values.

Classe NumericCharacteristic:

As características do tipo quantitativo são definidas por esta classe. Uma dessas características pode ser, por exemplo, o preço de cada software. A variável única desta característica é, portanto, o valor dela, definido por um inteiro na variável value.

Classe YesNoCharacteristic:

As características booleanas (de apenas Sim ou Não), são definidas na classe YesNoCharacteristic. Posssuem como única variável, a variável state, que é um um vakir booleano que indica o estado da característica (True se Sim, False se Não). Este tipo de características é útil quando para saber se um Software possui ou não alguma propriedade (como por exemplo se possui um módulo de Ajuda).

Classe Value:

A classe Value é utilizada pela classe QualitativeCharacteristic para definir a relação entre o valor qualitativo (definido por uma palavra) e um valor númerico (para ser usado no processo de decisão). É através do valor númerico presente nesta classe, que é definida a ordem de valor das características qualitativas na classe Qualitative Characteristic.

Classe DataBaseUser:

Esta classe associa um utilizador (ver classe User) a um conjunto de Softwares (ver classe Software) e também a um conjunto de características (ver classe Characteristic e respectivas sub-classes). Um utilizador possui, na sua base de dados, um conjunto de softwares e de características personalizado, permitindo que estes possuam características exclusivas, que não são acedidas por outros utilizadores que possuam os mesmos softwares na sua base de dados. Tanto os Softwares como as Características são armazenados numa estrutura Dictionary, em que a chave é o seu Id e e o valor é um apontador para o objecto respectivo.

Os dados do utilizador a quem está associado essa base de dados são acedidos através de uma variável do tipo User, que apontará para o objecto respectivo.

Classe DataBaseManagement:

A classe DataBaseManagement é utilizada para gerir a classe DataBaseUser. Esta, na variável tableSW possui um Dictionary, em que, a cada chave (que corresponde ao nome de um software), está associado um segundo Dictionary, sendo a chave o nome da característica e o valor respectivo. Está também associada um objecto do tipo DataBaseUser, na variável db.