Relatório do Projecto Prático de Programação Orientada aos Objectos (2013/2014)

Grupo 50:
64315 - Filipe Ribeiro
64361 - Pedro Matos
64288 - Nelson Gomes

Jun. 2014

(a) Filipe Ribeiro

(b) Pedro Matos

(c) Nelson Gomes

Figura 1: Grupo 50

Resumo

O projecto da unidade curricular Programação Orientada aos Objectos consiste no desenvolvimento de uma aplicação que permita registar e simular actividades desportivas chamada FitnessUM. O processo de implementação do mesmo encontra-se devidamente detalhado neste relatório.

Conteúdo

Lista de Figuras

1 Introdução

O tema deste projecto era o desenvolvimento de uma aplicação de simulação de actividades desportivas. De forma a conseguir isso o grupo recorreu ao ambiente de desenvolvimento BlueJ. Este relatório descreve detalhadamente como se encontra organizado esta mesma aplicação, para isso encontra-se dividido em seis fases descrevendo cada uma delas uma etapa do projecto.

A primeira fase do projecto consiste na implementação e organização das actividades desportivas, de modo a ser fácil a expansão das mesmas futuramente.

A segunda fase descreve a implementação da classe *Amigo* e sua organização, sendo esta futuramente utilizada pela classe *Utilizador* .

A terceira fase descreve a classe mais importante, a classe *Utilizador*, esta utilizando as duas classes criadas nas duas primeiras fases.

A quarta fase consiste em informação relacionada com estatística e records de um dado utilizador.

Na quinta fase encontrasse todos os menus de interface com o utilizador presentes nesta aplicação.

Por ultimo, a sexta fase mostra as restantes classes criadas na realização desta aplicação que não foram abordadas anteriormente.

2 Diagrama Geral

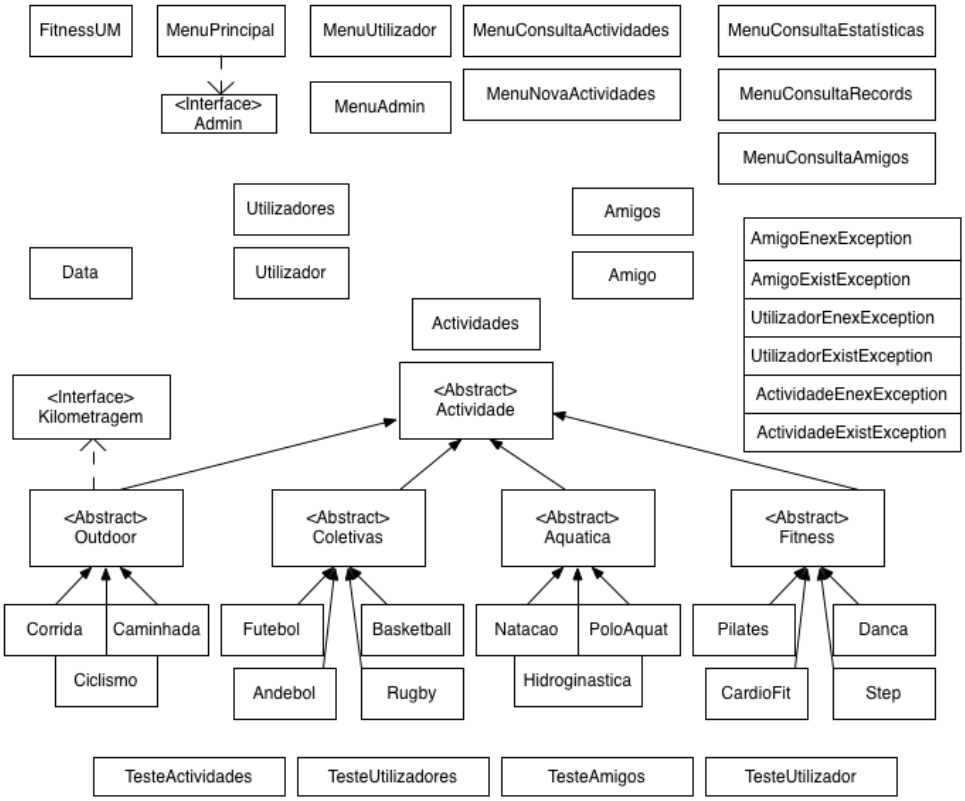

Figura 2: Diagrama geral da aplicação no BlueJ

3 Actividades

3.1 Organização das Actividades

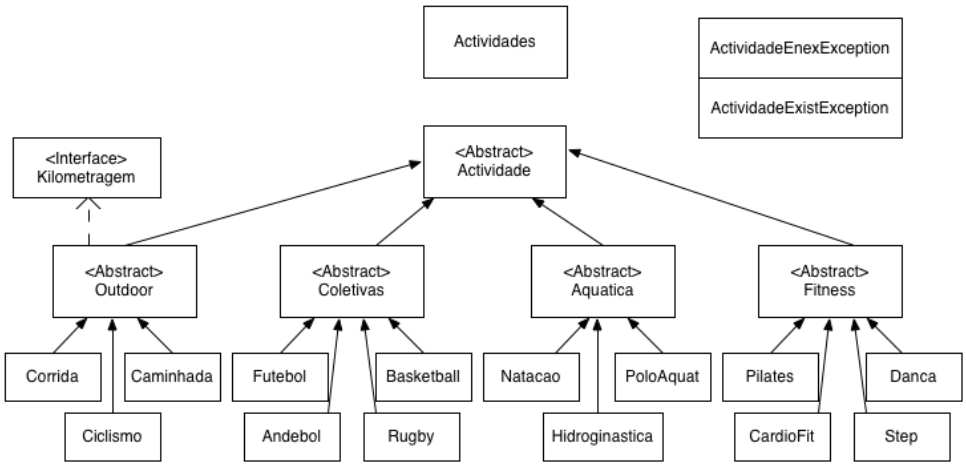

Figura 3: Organização das Actividades

Como mostra na figura 2, a classe *Actividade* esta declarada com <*abstract*> tendo como objectivo a reutilização do código através de herança. As actividades estão divididas em outras quatro classes, *Outdoor* , *Coletivas*, *Aquatica* e *Fitness* , também estas declaradas como classes abstractas pelo mesmo objectivo que a anterior. Cada uma destas subclasses têm actividades desportivas associadas, sendo que com este tipo de organização é possível inserir outras actividades facilmente devido mais uma vez à herança de classes. Nesta figura é possível ver também a classe *Kilometragem* declarada como <*Interface*> tendo como objectivo obrigar a implementação de um método que devolve o total de quilômetros nas actividades, neste caso, todas do tipo *Outdoor*. Existe também uma classe *Actividades*, esta classe foi criada essencialmente para a organização de código, contendo um conjunto de actividades organizadas de uma certa forma. Alem dessa existe mais duas classes *ActividadeEnexException* e *ActividadeExistException*, sendo cada uma delas exceções para funções implementadas na classe *Actividades*.

3.2 Estrutura Escolhida

A estrutura escolhida para guardar a informação de cada actividade foi um *HashMap*:

```
private HashMap<Integer,Actividade> l_actividades;
```

esta estrutura é declarada na classe *Actividades*, escolhida pela eficiência, sendo a chave de inserção um identificador de cada actividade e como objecto a informação da actividade. Esta estrutura nesta situação é bastante eficiente devido ao facto de existir um identificador único para cada actividade.

4 Amigos

4.1 Organização de Amigos

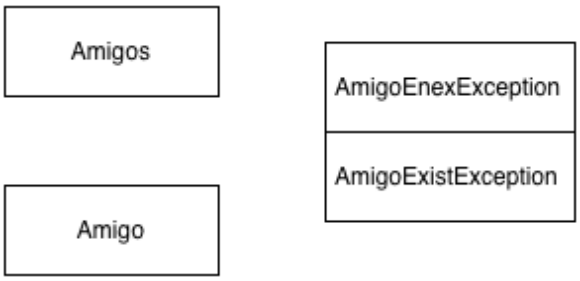

Figura 4: Organização dos Amigos

Como mostra na figura existe a classe *Amigo*, esta contendo a informação de cada amigo. Tal como a classe *Actividades*, existe também uma classe *Amigos* com o mesmo objectivo, contendo um conjunto de amigos.

Alem dessas existe mais duas classes *AmigoEnexException* e *AmigoExistException*, sendo cada uma delas exceções para funções implementadas na classe *Amigos*.

4.2 Estrutura Escolhida

A estrutura escolhida para guarda a informação de cada amigo foi um *ArrayList*:

```
private ArrayList<Amigo> lista;
```

esta estrutura é declarada na classe *Amigos*, sendo escolhida pela simplicidade, sendo o a classe *Amigos* uma classe simples com apenas duas variáveis do tipo *String* uma estrutura simples é o ideal para usar.

5 Utilizadores

5.1 Organização de Utilizadores

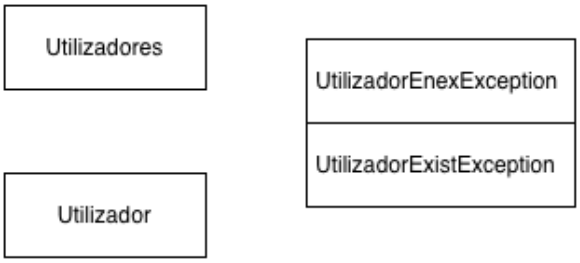

Figura 5: Organização dos Utilizadores

Tal como na classe Amigos, como mostra na figura existe uma classe Utilizador, contendo a informação de casa utilizador, existe também uma classe *Utilizadores* contendo um conjunto de utilizadores.

Mais uma vez, alem dessas existe mais duas classes *UtilizadorEnexException* e *UtilizadorExistException*, sendo também cada uma delas exceções para funções implementadas na classe *Utilizadores*.

5.2 Estrutura Escolhida

A estrutura escolhida esta na classe *Utilizadores* sendo escolhida para guardar informação de cada utilizador tal como anteriormente foi um *HashMap*:

```
private HashMap<String,Utilizador> users;
```

esta estrutura foi escolhida pelo facto de só existir um e-mail distinto para cada utilizador, sendo assim esta estrutura baste eficaz.

6 Estatística / Records

6.1 Estatística

O menu que lista as estatísticas será acedido a partir do menu principal, apresentado mais a frente. Ao lista as estatísticas será possível aceder a três opções:

- Estatísticas gerais total;
- Estatísticas gerais ultimo mês;
- Estatísticas gerais Anual;

cada uma delas mostra a mesma quantidade de informação, a primeira opção tal como o nome indica, mostra as estatísticas de todas as actividades já praticadas, a segunda do último mês e a terceira opção, pedindo ao utilizador um ano, mostra a respectiva estatística desse mesmo ano.

- Cada uma das opções mostrará:
  - Numero total de actividades praticadas;
  - Duração total despendidas em todas as actividades;
  - Total de Calorias queimadas em todas as actividades;
  - Total de quilômetros em todas as actividades (só contribui para este item as actividades que respondem a classe *Kilometragem* já referido em anteriormente);

6.2 Recordes

O menu de recordes tal como o de estatística será acedido do menu principal. Este menu acaba por ser parecido com o de estatística, tratando-se apenas de uma lista de records simples. Ao listar será possível aceder a uma única opção:

- Recordes Gerais
  - Lista total de actividades praticadas;
  - Máxima duração numa actividade física;
  - Máximo de calorias queimas em uma só actividade;

a opção apresenta três resultados relacionados as actividades realizadoras por um dado utilizador.

7 Organização de Menus

7.1 FitnessUM

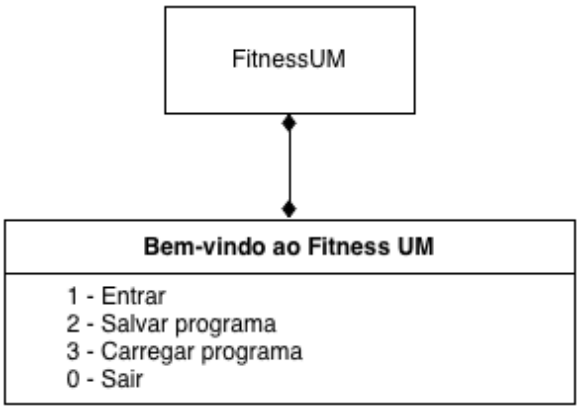

Figura 6: Classe e Menu FitnessUM

Este menu é o primeiro da aplicação contendo quatro opções relevantes, a primeira para entrar na verdadeiramente na aplicação, a segunda para guardar a informação uma vez introduzida, para futuramente usar a terceira opção para voltar a carregar essa respectiva informação. Por último, tal como o nome indica a quarta opção sai da aplicação.

7.2 Menu Principal

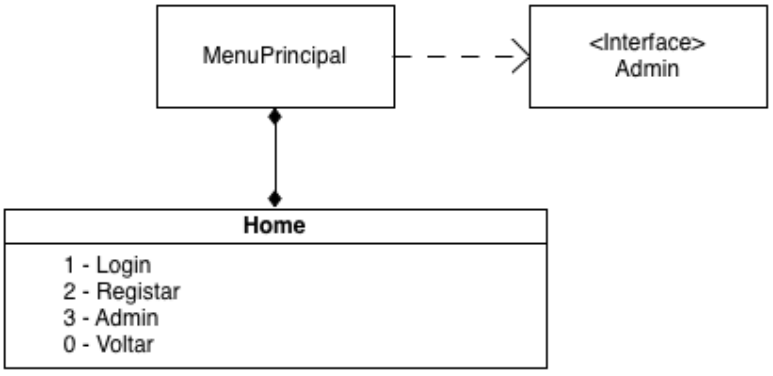

Figura 7: Classe e Menu Principal

- Login

Uma vez que um utilizador já se encontra no sistema, basta fazer login com o seu respectivo e-mail e password, para assim poder aceder as suas actividades.

- Entrar

Se o utilizador ainda não estiver registado no sistema, tem com esta opção a possibilidade de se registar, introduzindo os seus dados conforme lhe seja pedido pelo sistema.

- Amin

Esta é uma opção é relativamente igual a primeira, mas ao contrário da mesma, esta opção requer dados de administração. Na figura 6 esta representada a classe *Admin* , declarada com <*Interface*> que contem declarados os dados de login de administração.

- Sair

Opção que permite voltar par o menu anterior.

7.3 Menu de Administração

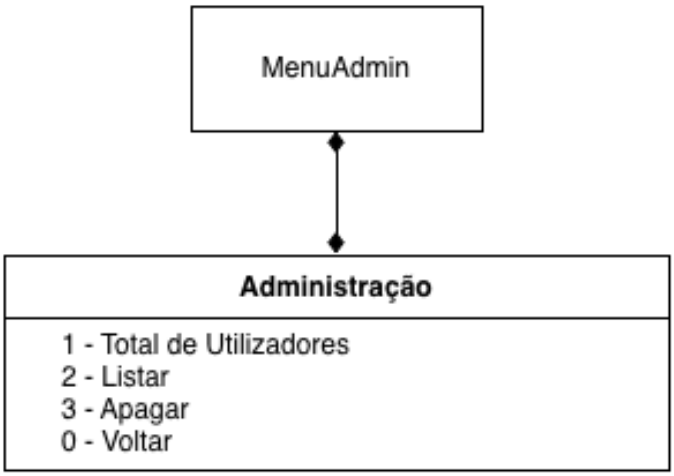

Figura 8: Classe e Menu Admin

- Total de Utilizadores

Esta opção simples mostra simplesmente o número total de utilizadores registados no sistema.

- Listar

Opção que não faz mais do que listar toda a informação de cada um dos utilizadores registados.

- Apagar

Como o nome indica, possibilita apagar um dado utilizador, introduzindo o seu e-mail.

- Voltar

Opção para voltar ao menu anterior, menu principal.

7.4 Menu Utilizador

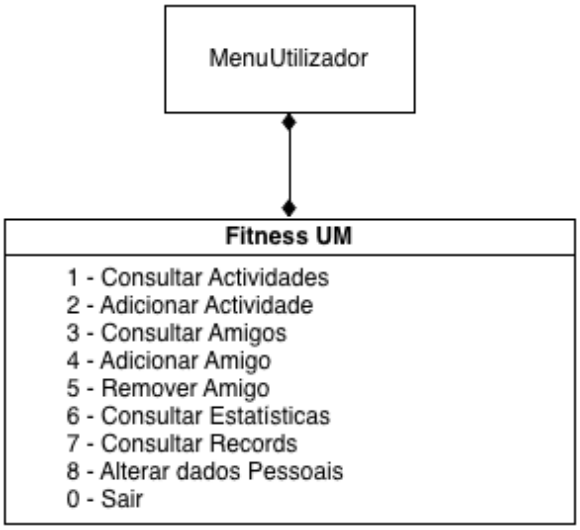

Figura 9: Classe e Menu Utilizador

Este menu é o menu mais importante do sistema, aonde o utilizador pode consultar e alterar tudo que ele tem acesso, é também um menu intermediário aonde a partir do mesmo se acesa outros menus, todos eles apresentados mais tarde.

- Consultar Actividades

Este é um dos casos que ao escolher esta opção irá para outro menu mais detalhado - *"MenuConsultaActividades"*.

- Adicionar Actividade

A semelhança da opção anterior, este também redireciona para outro menu - *"MenuNovaActividade"*.

- Consultar Amigos

Mais uma das opção de redirecionamento desta vez para o menu - *MenuConsulta-Amigos*.

- Adicionar Amigo

Esta opção ao contrário das três anteriores não chama outro menu, pedindo ela mesmo ao utilizador o e-mail e nome ou *nickname* do utilizador que quer adicionar como amigo. É de salientar que se o utilizador quiser inserir um outro utilizador que não esta registado no sistema ou que já se encontra na lista de amigos surgirá uma exceção para cada um dos casos.

- Remover Amigo

Esta opção semelhante à anterior, pedindo ao utilizador o e-mail do utilizador que quer remover da lista de amigos. Também neste caso é de salientar que se o utilizador quiser remover um outro utilizador que não esta registado na sua lista de amigos surgirá uma excepção para esse caso.

- Consultar Estatísticas

Sendo esta opção já abordada anteriormente, mas é mais uma das que redireciona para outro menu - *MenuConsultaEstatisticas*

- Consultar Records

Tal como a opção anterior, opção já abordada anteriormente, sendo também esta mais uma das que redireciona para outro menu - *MenuConsultaRecordes*

- Alterar dados pessoais

Esta opção consiste na possibilidade de alterar alguns dos dados do utilizador como, password, desporto favorito, altura e peso. Sendo cada informação pedida ao utilizador.

- Sair

Opção que permite sair deste menu, voltando para o menu anterior.

7.5 Menu Consulta Actividades

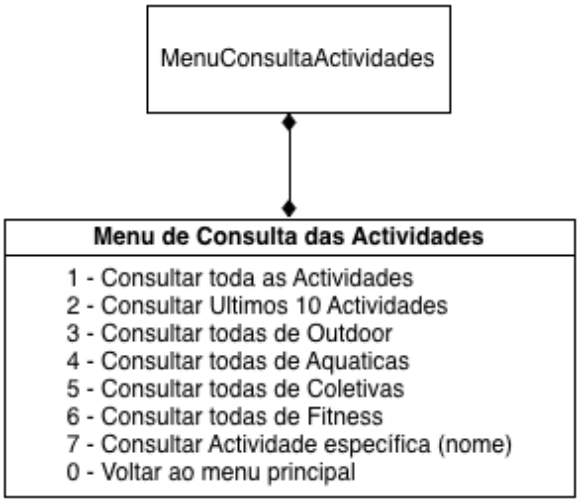

Figura 10: Classe e Menu Consulta Actividade

Estas opções são todas relativamente parecidas, servindo todas para consultas de actividades, todas, ultimas dez, possibilidade de consultar por tipo de actividade ou pelo nome especifico da mesma são o tipo de consultas que aqui é mostrado.

7.6 Menu Nova Actividade

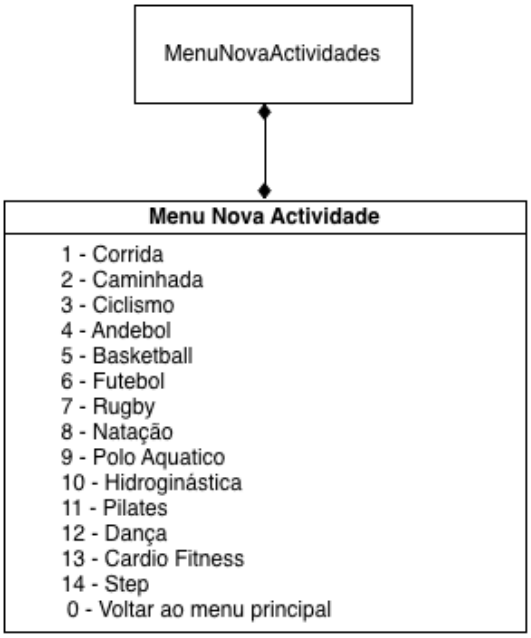

Figura 11: Classe e Menu Nova Actividade

Menu com a possibilidade de adicionar novas actividades das que existem na aplicação, ao escolher alguma delas, será pedido informação ao utilizador correspondente à actividade selecionada.

7.7 Menu Consulta Amigos

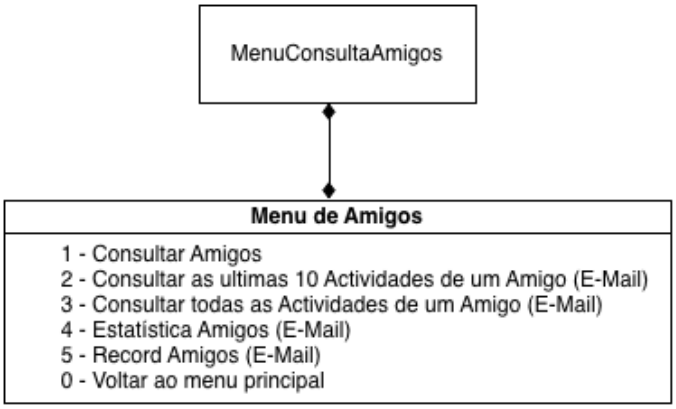

Figura 12: Classe e Menu Consulta Amigos

Menu que lista tudo sobre os amigos adicionados na lista de amigos do utilizador, mostrar todos, consultar todas ou as ultimas 10 actividades, estatísticas e recordes dado o e-mail do amigo a querer saber são as opções de consulta presentes neste menu.

7.8 Menu Consulta Estatística

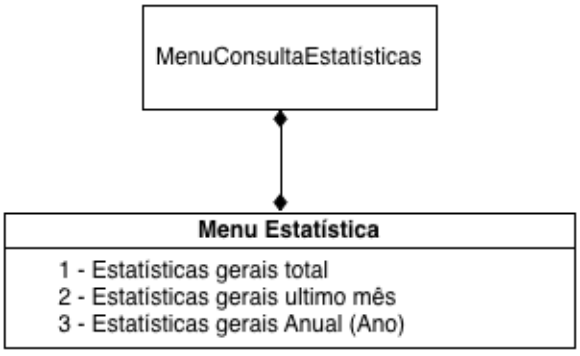

Figura 13: Classe e Menu Consulta Estatísticas

Este menu já foi referido anteriormente, senso responsável por mostrar a estatística de em dado utilizador. Estatísticas gerais, relacionadas com todas as actividades, do ultimo mês, ou também a possibilidade de consultar as estatísticas de um dado ano, pedido ao utilizador.

7.9 Menu Consulta Recordes

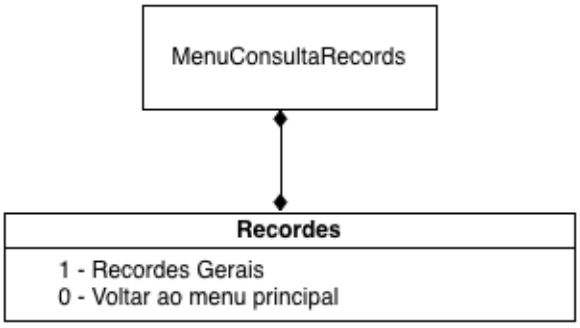

Figura 14: Classe e Menu Consulta Recordes

Este menu tal como o anterior já foi referido anteriormente, sendo responsável por mostrar básicos recordes do utilizador. Total de actividades praticadas, máxima duração numa única actividade e máximo de calorias queimadas numa só actividade são a informação que a opção deste menu mostra.

8 Outras Classes

Figura 15: Classes de teste

Classes ainda não referidas anteriormente mas indispensáveis na realização deste projecto, classes de teste, para um utilizador, e um conjunto deles, mas também para um conjunto de actividades bem como de amigos.

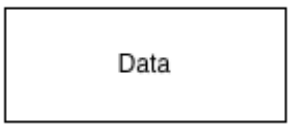

Figura 16: Classe Data

A criação desta classe surgiu com o aparecimento de dificuldades que o grupo teve com a classe *Gregorian Calendar* já pertencente ao java, tendo esta classe como objectivo, de alguma modo substituir essa mesma classe.

9 Conclusão

Numa primeira abordagem ao projecto tentamos definir classes genéricas para a implementação de actividades físicas, acabando por perceber que não era o mais correcto, a partir dai durante todo o resto do desenvolvimento do trabalho, as nossas principais preocupações foram de definir as variáveis de instância como sendo o mais genérico possível possibilitando um maior leque de possibilidades na definição dos vários métodos e definir classes abstractas que no futuro permitissem implementar/acrescentar novas actividades facilmente, sem ser necessário a redefinição do código existente.

É de salientar que além das estruturas de dados apresentadas neste relatório, foram usadas também *TreeMap* e *TreeSet* essencialmente para métodos de ordenação, permitindo logo à partida, por comparadores definidos nas classes, que é o caso das classes *Actividade* e *Data*, mas também, estando já definido, o comparador de *Strings* uma ordenação na inserção.