

### 0.0.1 Classe abstracta Event

Classe com o conceito mais abstracto de Evento, contém as variáveis:

- *String name*;
- *String tipoActivity*;
- *String location*;
- *int maxParticipants*;
- *int participants*;
- *GregorianCalendar deadline*;
- *GregorianCalendar date*;
- *double duration*;
- *TreeSet<User> participantsList*;
- *TreeSet<Ranking> ranking*;
- *TreeSet<Ranking> desistentes*;
- *TreeSet<Simulacao> simula*;

respectivos *getters* e *setters* e os vários construtores. Ainda tem métodos auxiliares para, adicionar um *User*, *Ranking* (desistente ou não) e *Simulacao* aos respectivos *Sets* e para mostrar a classificação geral do evento.

### 0.0.2 Tipo de Evento

Subclasses de Evento (Marathon, HalfMarathon, MarathonBTT e Trail), todas estas contem mais uma variável *distance*, que nos casos de Marathon e HalfMarathon são variáveis *final*, porque este tipo de eventos tem distâncias especificas. Não tem métodos auxiliares para além de *getDistance()*.

### 0.0.3 Classe abstracta Activity

Esta é a classe mais abstracta que contem o conceito de actividade. Contém variáveis comuns a todas as actividades:

- *String name*;
- *GregorianCalendar date*;
- *double timeSpent*;
- *double calories*;

tal como os construtores, *getters* e *setters*.

### 0.0.4 Classe abstracta Person

Classe geral para todo tipo de utilizador. As suas variáveis são:

- *String email* - Email do utilizador;
- *String password* - Password da conta;
- *String name* - Nome do utilizador;
- *char gender* - Género do utilizador;
- *GregorianCalendar dateOfBirth* - Data de nascimento do utilizador;

e contém os métodos construtores *getters* e *setters*

### 0.0.5 Classes User e Admin

As subclasses de Person referem-se a dois possíveis tipos de utilizador; utilizador normal ou utilizador com privilégios de administrador.

A classe Admin não tem métodos ou variáveis adicionais, visto que este tipo de utilizador apenas opera sobre a base de dados da aplicação.

A classe User adiciona as seguintes variáveis:

- *int height*;
- *double weight*;
- *String favoriteActivity*;
- *TreeSet<Activity> userActivities* - Actividades realizadas pelo utilizador;
- *TreeSet<String> friendsList* - Lista dos amigos do utilizador;
- *TreeMap<String, ListRecords> records* - Lista dos seus recordes pessoais;
- *TreeSet<String> messageFriend* - Lista de pedidos de amizade;

Respectivos métodos *getters* e *setters*, construtores e métodos auxiliares para gerir os seus amigos, recordes, as suas actividades e estatísticas relevantes. Ainda contém funções auxiliares para a simulação de eventos.

### 0.0.6 Comparador

O tipo Person tem apenas um comparador:

- ComparePersonByName - que ordena por ordem alfabética do seu nome.

### 0.0.7 Statistics

A classe Statistics é usada para mostrar ao utilizador dados relevantes das suas actividades, estes podem ser discriminados por um dado mês ou por um ano. As suas variáveis são:

- *double timeSpend* ;
- *double calories*;
- *double distance*;

contém os respectivos métodos *getters* e *setters* e construtores.

### 0.0.8 Classe abstracta Record

Esta classe representa todos os recordes que o utilizador pode bater. Contém apenas uma variável:

- *String name*;

métodos construtores, *getName()* e *isEmpty()* que verifica se esse recorde existe ou não.

### 0.0.9 DistancePerTime e TimePerDistance

Estas classes simboliza os dois diferentes tipos de recordes.

DistancePerTime é um recorde em que o objectivo é fazer a maior distância para um dado tempo. As suas variáveis são:

- *double recordTime* - Tempo do recorde;

- *double distance* - Distância registrada;

Enquanto que TimePerDistance representa um recorde de menor tempo para uma certa distância. As suas variaveis sao:

- *double recordDistance* - Distância do recorde;
- *double time* - Tempo registrado;