

### 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*;
- *String password*;
- *String name*;
- *char gender*;
- *GregorianCalendar dateOfBirth*;

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*;
- *TreeSet<String> friendsList*; *TreeMap<String, ListRecords> records*;
- *TreeSet<String> messageFriend*;

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 Comparators

O tipo Person tem apenas um comparator:

- 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.