

# JogginGo!

Luís Carlos Moreira Dias

Faculdade de Engenharia  
da Universidade do Porto  
R. Dr. Roberto Frias, 4200-465 Porto  
ei08094@fe.up.pt

Luís Filipe Castanheira Gomes

Faculdade de Engenharia  
da Universidade do Porto  
R. Dr. Roberto Frias, 4200-465 Porto  
ei08169@fe.up.pt

28 de Abril de 2013

## Resumo

Com uma interface limpa e amigável do utilizador, o JogginGo! é uma aplicação Web que permite a gestão de todas as corridas feitas por qualquer utilizador registado. Cada corrida é tratada como um conjunto de coordenadas GPS (*Global Positioning System*) recolhidas com recurso aos sensores de um dispositivo móvel. A cada minuto, intervalo de tempo definido, é recolhida a posição em que o atleta se encontra, e o conjunto de pontos é assim representado visualmente na interface web. É depois possível também competir contra outros utilizadores, através de contra-relógio, em percursos pré-definidos pela plataforma. Utilizadores que façam jogging no seu dia a dia podem assim aproveitar a evolução dos dispositivos móveis e dos sistemas GPS para melhorarem a sua performance, ao mesmo tempo que aproveitam a componente social para maior diversão.

## 1 Introdução

Os sistemas GPS (*Global Positioning System*) são responsáveis pela recolha, em tempo real, da posição de um determinado dispositivo no mundo. Actualmente um sistema GPS é altamente fiável, garantindo com enorme exactidão a posição, em formato de coordenadas, num mapa previamente inserido. Para além da posição, através de um GPS é possível calcular dados como velocidade média, velocidade instantânea ou até o relevo. Por outro lado, assistimos a um evoluir sistemático da utilização de dispositivos móveis, tais como os *smartphones*. A cada dia, a sua utilização é mais massificada, já que são dispositivos com alta mobilidade e performance, que permitem aceder à internet em qualquer lado, a qualquer altura. Aproveitando as suas capacidades técnicas, uma percentagem altamente elevada destes dispositivos incluem sen-

sos GPS. É através deste princípio que surge o "JogginGo!". Utilizando as capacidades de um dispositivo móvel Android, um *jogger* inicia o seu percurso, e automaticamente a aplicação móvel recolhe, em intervalos de tempo definidos, a sua posição actual. Esta informação é guardada em formato XML, e no final do percurso existe assim um conjunto estruturado de informação que é depois sincronizada com o *webservice*. Recorrendo à interface web, o jogger pode ver todos os seus percursos e respectivos tempos e outra informação como, por exemplo, a velocidade média calculada com base na distância percorrida. Para além desta introdução, onde se caracterizou o problema abordado por este projecto, refere-se na secção ?? o estado da arte, onde são descritos os trabalhos relacionados com a captura de coordenadas em dispositivos móveis e a sua gestão e visualização na Web para melhor enquadramento do leitor nesta temática.

## 2 Estado da Arte

AQUI VAI O ESTADO DA ARTE

## 3 Requisitos do Utilizador

Para o uso da plataforma, qualquer utilizador necessita de ter um email válido, para que se possa registar. Após o registo estar concluído, o utilizador tem acesso a uma série de funcionalidades. Nesta arquitectura do JogginGo! existem três níveis de utilizadores.

- Utilizador – Este tipo de utilizador é o utilizador final. Ou seja, o cliente alvo para o JogginGo!. A este tipo de utilizador é permitido fazer as suas próprias corridas, competir em corridas pré-definidas e sugerir que alguma(s) da(s) sua(s) corrida(s) passem a

pré-definidas, pertencentes à funcionalidade *Test your limits*.

- **Moderador** – Os moderadores têm a função de garantir o bom funcionamento da plataforma, assim como prestar auxílio aos utilizadores na utilização do sistema. Para além disso, os moderadores têm como responsabilidade a aprovação, ou não, do pedido de um utilizador a que a sua corrida seja tida em conta como pertencente à funcionalidade "*Test your limits*". Para tal, o moderador deve avaliar alguns factores, nomeadamente a dificuldade do percurso ou a área involvente (se há ou não alguma corrida pré-definida nas proximidades, se a paisagem é agradável, etc.).
- **Administrador** – O Administrador existe para actuar ou resolver algum conflito que saia da área de acção do moderador. A reter, o administrador pode banir um utilizador, apagar alguma corrida, introduzir alguma corrida pré-definida, etc.

## 4 Arquitectura da solução

Tal como temos vindo a mostrar ao longo do desenvolvimento do projecto, um dos principais problemas prende-se na forma como os dados irão ser armazenados, a sua estrutura, para que sejam facilmente interpretáveis. Para responder a este problema, optámos por utilizar uma linguagem de marcação (XML) como base para a estruturação dos dados. No entanto, cada documento XML, que contém a informação de cada corrida, tem que guardar dados relativos ao utilizador, como o seu id, e dados da própria localização da corrida. De seguida, apresentamos a nossa solução para este problema.

### INTRODUZIR O XML LINDO AQUI

Explicando melhor, este documento é um exemplo de uma representação de um percurso feito por um utilizador, neste caso, o utilizador com o *user\_id* igual a 1. Há outros campos que decodificamos facilmente o seu significado, como *city*, que representa a cidade onde foi feita a corrida, o *name*, que é o nome que o utilizador deu à sua corrida, ou mesmo os campos referentes ao tempo, que tem tempo inicial e final. No entanto, viremos a nossa atenção para os campos *approved* e *private*, com o valor *false* e *true* respectivamente. Por defeito, qualquer corrida criada por um utilizador tem o valor tem esses valores para os campos referidos anteriormente. Isto

foi uma implementação para diferenciar os trilhos que são privados, os que são privados à espera de aprovação para serem públicos, e os públicos. Ou seja, quando um user cria um trilho, *approved* e *private* tomam os valores *false* e *true*, respectivamente. Após isso, o utilizador pode pretender que o seu trilho seja passado para a funcionalidade *Test you limits*. Assim, *private* passa a valer *true*. No entanto, *approved* continua com o valor booleano para falso. Isto porque esse trilho ainda necessita de aprovação de um moderador para que passe a público. Se for aprovado, *approved* passa a *true* e o trilho passa a ser público. Caso contrário *approved* mantém o valor *false*, e *private* é alterado para *true*, exactamente igual ao momento da sua criação.

Como facilmente conseguimos perceber, esse *user* tem um *email*, um *id* um nome e um *username* associado. Após estas informações, segue a *tag* **<track>**. A partir daqui, são passadas todas as informações necessárias para que o *web-service* identifique da melhor maneira o trilho pretendido. Como podemos ver, cada percurso tem uma cidade, um país, um *id*, um nome e um *array* de pontos. Cada um desses pontos é caracterizado pela morada, pelo seu id, e pelas coordenadas GPS já referidas anteriormente (latitude e longitude). A morada fica guardada pois a API utilizada no googlemaps, necessita de uma morada para determinar o ponto em questão. Com esta estruturação, facilmente transferimos a informação entre os dois terminais, de uma forma simples, estruturada e de fácil leitura.

## 5 Conclusões

Neste artigo abordou-se o desenvolvimento de uma aplicação que permita a monitorização de um *jogger* ao longo de um percurso. Aliado à sua vertente social, esta plataforma permite fazer a ponte entre a individualidade de fazer *jogging* com a possibilidade de competir saudavelmente contra outros atletas. Posto toda a problemática, resolvemos dividir o desenvolvimento em três fases distintas:

- **Recolha de requisitos** – Nesta etapa foi feito um levantamento de todos os requisitos necessários para a correcta implementação desta aplicação. Foi feito um estudo de todas as tecnologias existentes a fim de escolher aquela que melhor preenchesse o requisitos da aplicação.
- **Recolha dos dados** – Apesar de já existir o

modelo do documento XML que vai vigorar na aplicação, ainda não foi feita a recolha da informação. Isto deve-se ao facto de o JogginGo! não utilizar nenhum dataset existente, mas sim, usar a sua aplicação móvel para a recolha dos próprios elementos. Assim, essa recolha será feita aquando da finalização do desenvolvimento da aplicação *Android*.

- **Análise dos dados e desenvolvimento da plataforma** – Nesta terceira etapa pretende-se fazer a análise dos documentos XML recebidos em ambos os terminais, e desenvolver a plataforma para que possa ser usada pelo utilizador final.

No fim destas três etapas, pretendemos ter uma plataforma pronta a usar em qualquer parte do Mundo, usando documentos XML como base para a troca de informação entre *webservice* e os dispositivos móveis.

## Referências

- [1] W3C World Wide Web Consortium. XML Tutorial. <http://www.w3schools.com/xml/>.
- [2] Wikipedia. XML, What is it? <http://en.wikipedia.org/wiki/XML>.
- [3] Benjamin Roth. Google Maps for Rails. <https://github.com/apneadiving/Google-Maps-for-Rails>.
- [4] Google. Google Maps API. <https://developers.google.com/maps/>.