

# Um Aplicativo Móvel para Execução de Jogos Móveis Locativos com Realidade Aumentada

Messias Junior\*

Universidade Federal do Ceará  
Campus do PICI, CEP 60451-970  
Fortaleza, Ceará, Brasil  
messiaslima.03@gmail.com

Windson Viana

Universidade Federal do Ceará  
Campus do PICI, CEP 60451-970  
Fortaleza, Ceará, Brasil  
windsonviana@great.ufc.br

Fernando A.M.Trinta

Universidade Federal do Ceará  
Campus do PICI, CEP 60451-970  
Fortaleza, Ceará, Brasil  
fernando.trinta@great.ufc.br

Carleandro Nolêto

Universidade Federal do Ceará  
Campus do PICI, CEP 60451-970  
Fortaleza, Ceará, Brasil  
carleandrofederal@gmail.com

Luis Fernando Maia

Instituto Federal do Maranhão  
Campus Caxias, CEP 65600-000  
Caxias, Maranhão, Brasil  
luisfmss@gmail.com

## RESUMO

Desenvolver jogos é uma tarefa complexa, principalmente para pessoas não possuem conhecimento sobre programação de computadores. Ferramentas de autoria de jogos procuram fornecer recursos e abstrações para reduzir essa complexidade. Dentro desse contexto, esse trabalho de iniciação científica apresenta um aplicativo móvel para a interpretação e execução de jogos pervasivos baseados em localização e realidade aumentada. O aplicativo Android concebido interpreta jogos gerados por uma ferramenta de autoria Web. A ferramenta permite a modelagem de jogos locativos usando um modelo baseado em missões. O artigo também apresenta uma prova de conceito realizada com o aplicativo móvel.

## Palavras-chave

Jogos Móveis Locativos, Ferramenta de autoria, Realidade Aumentada.

## 1. INTRODUÇÃO

A popularização do *smartphones* fez com que uma quantidade cada vez maior de pessoas tenha acesso a esses aparelhos, e conseqüentemente, a um número cada vez maior de aplicativos móveis disponíveis no mercado [2]. Muitos desses aplicativos são jogos móveis que exploram em seus *gameplays* recursos multimídia dos *smartphones*. (e.g., sensores de luminosidade, acelerômetro e GPS).

Mecânicas de jogos são definidas como “como qualquer parte do sistema de regras de um jogo que abrange um, e somente um possível tipo de interação que ocorre durante o jogo, seja ela geral ou específica”[7] e os recursos citados

\*Bolsista de iniciação científica do PIBIC

podem ser usados para criar mecânicas ou aperfeiçoar as mecânicas de jogos já existentes.

Quando se usa informações de contexto do jogador [8], como localização do usuário em conjunto com conceitos da Computação Ubíqua/Pervasiva (e.g., interfaces naturais, adaptabilidade) para complementar ou basear mecânicas de jogos, temos a ideia de um jogo pervasivo. Computação ubíqua se refere a capacidade da computação de se desprender dos micro-computadores pessoais e computadores de grande porte para aderir a dispositivos eletrônicos menores, fazendo com que seja possível portá-la e transportá-la mais facilmente, tornando-se praticamente imperceptível ao usuário[10]. Assim, um jogo pervasivo é aquele cuja jogabilidade mescla elementos físicos e eletrônicos e que “pode borrar as fronteiras entre realidade e ficção com base no uso das novas mídias, usando o espaço urbano como pano de fundo para suas ações.”[3].Dentre os estilos de jogos pervasivos[5], destacam-se os jogos móveis locativos (JML) ou *location-based games*.

“Os JML são jogos urbanos que utilizam tecnologias e serviços baseados em localização nos quais o lugar é parte integrante das regras e das ações dos jogos.”[6]

O trabalho proposto neste artigo, descreve um dos módulos de uma ferramenta de autoria de JMLs. Pesquisadores colaboradores desenvolveram um sistema que permite a especificação de JMLs. O objetivo desse projeto é permitir que pessoas com pouco ou nenhum conhecimento prévio em linguagens de programação, como C ou Java, possam desenvolver seus próprios jogos, de uma maneira fácil, rápida e sem o auxílio de desenvolvedores ou analistas de *softwares* móveis. Essa ferramenta de criação é baseada na *web*, e os jogos criados por ela são interpretados por um aplicativo para a plataforma Android, que é o objeto principal desse trabalho de iniciação científica.

## 2. ARQUITETURA DA FERRAMENTA DE AUTORIA

A Figura 1 apresenta a arquitetura do ferramenta de autoria de JML que é dividida em três módulos: a ferramenta Web de criação<sup>1</sup>, o servidor dos jogos e o aplicativo interpre-

<sup>1</sup>Vídeo de demonstração da ferramenta: <https://youtu>.

tador de jogos. A comunicação entre eles ocorre utilizando o protocolo HTTP sob o paradigma cliente-servidor. A ferramenta fornece interface de modelagem e configuração, o servidor gerencia a execução dos jogos e dos jogadores e o aplicativo provê interface de interação com o jogo.

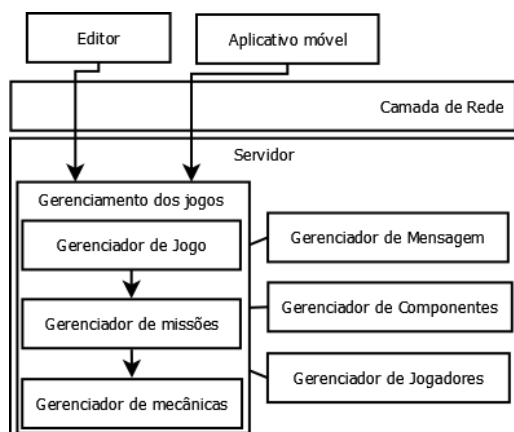


Figura 1: Arquitetura da Ferramenta de Autoria de JML.[9]

Os jogos modelados são formados por configurações de missões, missões e mecânicas. Estas últimas podem ser simples ou compostas por outras mecânicas. Existe uma relação hierárquica entre esses itens. As mecânicas são as ações dos jogadores, o que terão que executar para prosseguir no jogo. Conjunto de mecânicas formam missões, e o conjunto delas, as configurações de missão.

### 3. INTERPRETADOR DO JOGO

A aplicação<sup>2</sup> foi desenvolvida para a plataforma Android, utilizando a linguagem Java. Foi criado para interpretar todas as configurações de jogos gerados pela ferramenta desenvolvida anteriormente, dando suporte as mecânicas definidas pela ferramenta, obedecendo a ordem delas, a quantidade de equipes, definições e as restrições definidas pelo *game designer*. A Figura 2 apresenta os principais componentes do aplicativo, que são: I- Visualizador de mapas; II- Visualizador de realidade aumentada baseada em localização; III- Visualizador de imagens, áudio e vídeo; III- Visualizador de imagens, áudio e vídeo; IV- Cliente de câmera para a captura de fotos e vídeos; IV- Módulo de aquisição de contexto; e V- Módulo de comunicação e notificação.

#### 3.1 Mecânicas e recursos

Além de um inventário de itens, as mecânicas que podem ser executadas pelo aplicativo:

- Visualizar, capturar e deixar no mapa, arquivos de mídia como imagens, áudios, vídeos e textos.
- Ir a um local específico: O jogador realiza a missão ao chegar em um local assinalado no mapa.
- Capturar outro jogador: Se um jogador estiver próximo o bastante de outro, poderá tocar no marcador que o representa e retirar pontos dele.

be/8KmFmyFnEqw

<sup>2</sup>Video de demonstração do aplicativo: <https://youtu.be/PIuRA3n0cH8>

- Compartilhar itens do inventário com outros jogadores. Um item pode ser passado para o inventário de outro jogador.
- Visualizar e deixar no mapa objetos de realidade aumentada.. O jogador pode visualizar por meio da câmera, um objeto 3d

O aplicativo também permite que membros da mesma equipe se comuniquem por meio de um chat. Para as mecânicas que utilizam mídias, utiliza-se recursos de hardware presentes nos *smartphones*, câmera, microfone, alto-falantes e teclado.

Para as que necessitam ter como parâmetro a posição de um ou mais jogadores, utiliza-se o sensor de GPS e para a visualização de objetos de realidade aumentada é necessário a análise de dados de vários sensores ao mesmo tempo, GPS, acelerômetro e giroscópio.

#### 3.2 Fluxo da aplicação

A figura 3 mostra o fluxo de execução da aplicação.

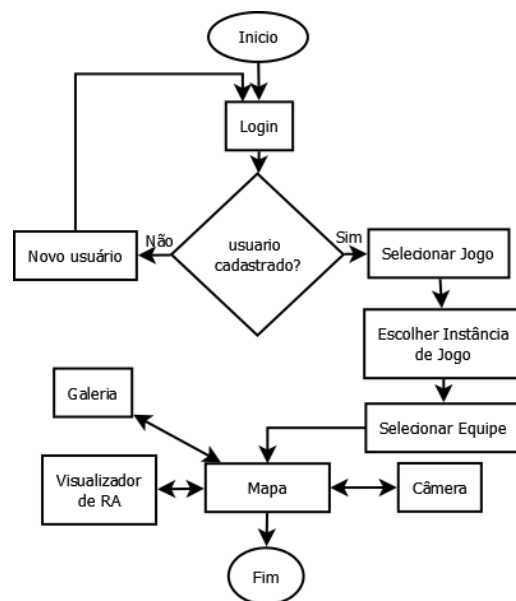


Figura 3: Fluxo de utilização do aplicativo.

##### 3.2.1 Seleção de jogo, instância e equipe

São listados os jogos disponíveis que foram criados anteriormente na ferramenta (Figura 4b). É possível personalizar a lista efetuando um filtro de busca pela distância do jogo ao jogador. Na tela seguinte, são mostradas as instâncias do jogo em execução atualmente no servidor. Essas instâncias são independentes umas das outras, e funcionam como “salas”. Se dois grupos de pessoas participarem do mesmo jogo mas de instâncias diferentes, pessoas do primeiro grupo não interagem com pessoas do segundo. O usuário pode optar entre entrar em uma em andamento ou iniciar uma nova além de ser possível participar de mais em uma.

O jogador deve selecionar em qual equipe deseja jogar. O número e o nome dos grupos é definido pelo *game designer* no momento da concepção do jogo. Após a seleção das equipes, é realizada uma requisição HTTP para o servidor a fim de recuperar as mecânicas referentes a equipe se-

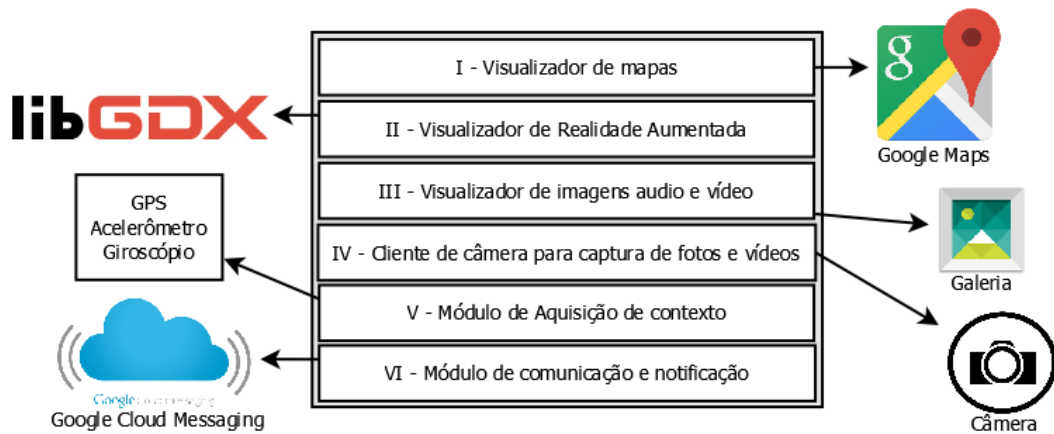


Figura 2: Componentes do aplicativo interpretador do jogo

lecionada e a lista dos arquivos relacionados a elas para que seja realizado seu download (imagens, áudio, vídeo, objetos 3d). Quando o processo cliente recebe todas as informações necessárias ao início da partida, a tela de mapa é mostrada.

### 3.2.2 Mapa

Utiliza-se da Google Maps API v2 para mostrar um mapa. Nele é possível ver marcadores que representam o jogador, os colegas de equipe, usuários de grupos adversários e aliados, e as mecânicas (Figura 3a). Estas últimas são representadas por bandeiras de cores diferentes e cada cor correspondente a um estado: Verde a mecânica está disponível, Azul ela está realizada e Vermelha ela está bloqueada.

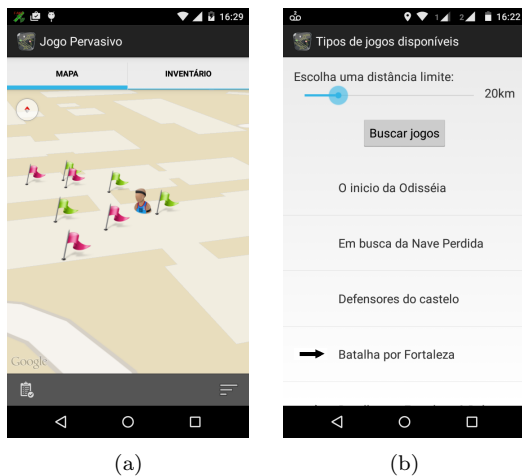


Figura 4: Mapa do jogo com mecânicas (a) e tela para escolha do jogo(b).

Vale lembrar que a habilitação dessas mecânicas no mapa depende da configuração de visibilidade feita pelo *game designer* no momento da criação. Ele define se esses marcadores são mostrados ou não.

No momento da configuração da mecânica, o *game designer* pode optar entre quatro opções de visibilidade. São elas: *Nunca*, *Quando não realizada*, *Quando desbloqueada* e *Sempre*.

Quando a visibilidade é definida como *Nunca*, marcador nunca é mostrado e a mecânica é executada automática-

mente quando o jogador atinge uma distancia mínima do local definido. Quando é definida como *Quando não realizada*, o marcador é mostrado até o momento em que a mecânica é realizada. Depois disso o marcador é removido. Definindo como *Quando desbloqueada*, o marcador se torna visível quando a mecânica passa do estado “bloqueada” para “desbloqueada”. Definindo como *Sempre*, o marcador é sempre visível. Existe um recurso de filtro de marcadores. O jogador pode optar por ver apenas os marcadores de mecânicas de uma missão específica.

Essa *activity*<sup>3</sup> também possui uma aba para acesso rápido para o inventário do jogador, como o mostrado na figura 4(a). Objetos no inventário do jogador podem ser compartilhados com os jogadores que estiverem próximos.

### 3.2.3 Vídeo e imagem

Para executar as mecânicas que envolvem vídeo, o sistema móvel se utiliza de recursos do próprio sistema operacional. Para capturar vídeo faz-se uma chamada, por meio de uma *intent*<sup>4</sup> para o aplicativo nativo de câmera. O mesmo se aplica à captura de fotos. Para visualizar um vídeo, usa-se o aplicativo configurado como *player* padrão do sistema.

### 3.2.4 Realidade aumentada (RA)

Para introduzir mecânicas de RA foi necessário reunir informações de mais de um sensor ao mesmo tempo para projetar na tela do dispositivo, um objeto modelado em 3d no mundo real.

Um módulo de visualização de objetos de RA foi desenvolvido por um aluno de doutorado do grupo de pesquisa, em paralelo ao projeto principal. O visualizador consistem em uma *activity*, com imagem captada pela câmera no plano de fundo. Sobre ela, é projetada um objeto 3d, como pode ser visto na figura 5(b). Isso é feito utilizando a biblioteca LibGDX[1].

A manipulação desse objeto ocorre em tempo real e usa como parâmetros os valores captados pelo módulo de aquisição de contexto. A distância do objeto é definida pela distância entre a localização do jogador e o local onde foi definida a mecânica de visualização. Assim, a medida que o jogador se

<sup>3</sup>Activity: Classe Java que representa uma tela de aplicativo, no sistema Android

<sup>4</sup>Intent: Classes Java que manipula chamadas dentro do aplicativo e chamadas para aplicativos externos

aproxima do local do objeto, este aumenta de tamanho e a medida que a distância aumenta, o seu tamanho diminui. Da mesma maneira, a posição na tela pode ser definida pela movimentação captada pelo giroscópio e a inclinação fornecida pelo acelerômetro.



Figura 5: Inventário com um item de objeto 3d (a) e tela de visualização de realidade aumentada(b).

### 3.2.5 Módulo de comunicação e notificação

Entre o aplicativo e o servidor, comunica-se por mensagens no padrão JSON (JavaScript Object Notation), e para isso, o protocolo de transferência de hipertexto (HTTP) é utilizado. A escolha do JSON foi deve-se a sua forma de representar um objeto e prover documentos estruturados mais leves, as mensagens serem pequenas e consequentemente menos dados trafegarem pela rede [4].

Por outro lado, o jogador também precisa ser informado quando eventos que afetam o estado do jogo de um modo geral acontecem, caso contrário, a integridade das informações mostradas na interface seria comprometida. Para contornar esse problema sem que fosse necessário fazer requisições periódicas ao servidor, o aplicativo conta com o serviço de push-notifications do Google, o Google Cloud Messaging (GCM). Dessa maneira, os clientes são avisados quando existe a necessidade sincronizar novamente os dados com o servidor.

Esse recurso é usado quando algum jogador realiza uma mecânica. Os outros precisam saber qual foi realizada ou se novas foram desbloqueadas, por exemplo. O chat também se baseia nesse serviço.

## 4. PROVA DE CONCEITO

“Em busca da nave perdida” foi um jogo foi concebido para ilustrar o funcionamento do aplicativo. O jogo consiste em 4 mecânicas, um texto de introdução, visualização de uma imagem com pistas de onde a nave está, visualizar uma nave em RA e um vídeo de conclusão do jogo. As figuras 4 e 5 são referentes a essa prova de conceito.

O editor anteriormente citado gera configurações de jogos, que devido ao número de parâmetros mutáveis, podem ser muito diferentes uns dos outros. Esse foi um dos desafios encontrados na fase de implementação do aplicativo, pois quando o universo de possíveis configurações aumenta,

proporcionalmente cresce a quantidade variações de jogos consequentemente de eventos inesperados. Ainda na fase de desenvolvimento, a visualização de RA foi um desafio. Por causa da grande variedade de aparelhos, não existe a garantia de que o aparelho que executará a aplicação possui todos os sensores necessários para uma boa experiência de visualização de RA.

Uma outra limitação encontrada foi a qualidade da internet móvel fornecida atualmente pelas operadoras de telefonia. Uma conexão lenta poderia dificultar ou até impossibilitar a realização de partidas.

## 5. CONCLUSÃO

O uso de ferramentas que facilitem o desenvolvimento de JMLs tem o potencial para popularizar esse tipo de jogo. JMLs podem ser usados de muitas formas, como por exemplo em aplicações turísticas, educativas e publicitárias. Entretanto, quanto mais recursos forem disponibilizados pelas ferramentas de autoria de JMLs, mais flexível deve ser a aplicação responsável por interpretar os jogos gerados, sendo, portanto fundamental para a execução destes jogos.

Neste trabalho apresentamos uma aplicativo móvel que serve como módulo para interpretação dos JMLs desenvolvidos na ferramenta proposta em [9] por usuários que não possuem conhecimento sobre programação.

## 6. REFERENCIAS

- [1] LibGDX Introduction. <https://libgdx.badlogicgames.com/documentation.html>. Acessado em: 2015-08-20.
- [2] Number of available applications in the Google Play Store from December 2009 to July 2015 . <http://goo.gl/X1EJIf>. Acessado em: 2015-08-20.
- [3] L. A. Andrade. Jogos pervasivos: Educação, cultura e cidade digital. *Revista Opara*, 3(1), 2013.
- [4] D. Crockford. The application/json Media Type for JavaScript Object Notation (JSON). <https://tools.ietf.org/html/rfc4627>.
- [5] V. Kasapakis and D. Gavalas. Pervasive gaming: Status, trends and design principles. *Journal of Network and Computer Applications*, 55:213 – 236, 2015.
- [6] A. Lemos. Jogos móveis locativos: Cibercultura, espaço urbano e mídia locativa. *Revista USP*, (86):54–65, 2010.
- [7] S. Lundgren and S. Bjork. Game mechanics: Describing computer-augmented games in terms of interaction. In *Proceedings of TIDSE*, volume 3, 2003.
- [8] M. E. F. Maia, A. Fonteles, B. Neto, R. Gadelha, W. Viana, and R. M. C. Andrade. Loccam - loosely coupled context acquisition middleware. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13*, pages 534–541, New York, NY, USA, 2013. ACM.
- [9] C. Nolêto, W. Viana, and F. Trinta. Uma ferramenta de autoria para o desenvolvimento de jogos pervasivos baseados em realidade aumentada. In *Anais do Workshop de Teses e Dissertações, XX Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, 2014.
- [10] M. Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991.