

# Uma Ferramenta de Autoria Web para Edição de Jogos Móveis Baseados em Localização

Carleandro Nolêto  
Fernando Trinta  
Universidade Federal do Ceará  
Campus do PICI, CEP 60451-970  
Fortaleza, Ceará, Brasil

Luis Fernando Maia  
Instituto Federal do Maranhão  
Campus Caxias, CEP 65600-000  
Caxias, Maranhão, Brasil

Windson Viana  
Messias Lima  
Universidade Federal do Ceará  
Campus do PICI, CEP 60451-970  
Fortaleza, Ceará, Brasil

## RESUMO

Este artigo têm como objetivo apresentar uma ferramenta de autoria para a criação de jogos móveis baseados em localização com recursos de realidade aumentada. Jogos móveis baseados em localização são uma subclasse de jogos pervasivos em que a jogabilidade tem uma forte dependência da localização dos jogadores. A ferramenta em questão oferece um ambiente integrado tanto para edição de jogos na Web, quanto para sua execução. Jogadores utilizam um aplicativo móvel desenvolvido sobre a plataforma Android para obtenção de sua localização via sensor GPS e execução dos jogos. Um diferencial de nossa proposta é a criação de jogos com missões compartilhadas entre equipes, além da possibilidade de se ordenar as missões dos jogos. Por fim, a ferramenta permite o uso de recursos de realidade aumentada como forma de aumentar a imersão do jogador no ambiente do jogo.

## Palavras-chave

Jogo Baseados em Localização, Realidade Aumentada, Ferramenta de Autoria

## 1. INTRODUÇÃO

Nos últimos anos Jogos Móveis Baseados em Localização (JMBLs) têm ganhado atenção na academia e indústria. Alguns exemplos desses jogos são: MusA[8], Geocaching[6], Urban Match[4] e Ingress<sup>1</sup>. JMBLs são jogos onde a localização do jogador tem um papel fundamental na narrativa do jogo. Jogadores utilizam dispositivos móveis como smartphones dotados de sensores de GPS para obter a localização do jogador no mundo real, e utilizar esta informação no jogo. Nesse tipo de jogo, ações do mundo real são mapeadas para o mundo virtual, e jogadores devem mover-se pelo ambiente para que tarefas sejam realizadas. Em geral, estes jogos determinam um conjunto de tarefas, onde de acordo com

<sup>1</sup>[https://play.google.com/store/apps/details?id=com.nianticproject.ingress&hl=pt\\_BR](https://play.google.com/store/apps/details?id=com.nianticproject.ingress&hl=pt_BR)

as regras especificadas em seu projeto, participantes devem deslocar-se pelo mundo real para interagir com objetos virtuais, como imagens, textos, vídeos e áudios.

O fato dos JMBLs permitirem a coleta de informações do contexto do jogador, como seu deslocamento no mundo real, potencializa seu uso em áreas como turismo e educação. Observa-se que, nestes jogos a ordem em que as tarefas são executadas pode afetar a interação e a experiência dos jogadores. Por exemplo, em jogos educacionais desafios podem ser resolvidos sequencialmente conforme o grau de dificuldade. Esta ordem pode ser planejada por profissionais, como especialistas em turismo ou educadores que, em geral, não têm conhecimento sobre desenvolvimento de aplicações. Uma alternativa é propor uma ferramenta de autoria que permita o desenho e implementação de JMBLs de maneira simples e intuitiva. É neste contexto que a presente ferramenta surge como uma solução para estas questões.

Este artigo apresenta uma ferramenta de autoria para JMBLs, onde pessoas sem conhecimento de linguagens de programação e desenvolvimento de aplicativos móveis podem criar e executar jogos. Com o objetivo de tornar mais simples e intuitivo o processo de criação dos jogos, optou-se por uma notação gráfica para projetar os JMBLs. Esta notação fundamenta-se no conceito de missões e mecânicas como objetos constituintes dos jogos. Missões representam um conjunto de mecânicas que os jogadores devem executar para alcançar seus objetivos. São exemplos de mecânicas: mover-se para um local específico, visualizar um objeto virtual e assim por diante.

Para concepções de cenários e mecânicas de JMBLs, foi realizado um levantamento na literatura e utilizada a técnica de grupos focais, empregada com sucesso no design de jogos [3]. No levantamento foram investigados requisitos para JMBLs e estudados jogos criados em outras ferramentas de autoria, como Aris Game[1], ALRA [5] e Tidy City Scout [9]. A partir do estudo destes trabalhos, cenários de aplicações foram detalhados, e nos forneceram requisitos para a idealização de um modelo de aplicação desejado para os JMBLs. Com base neste modelo, foi projetada e implementada uma arquitetura de software para suporte e execução de JMBLs, segundo cenários estabelecidos.

As próximas seções deste trabalho estão divididas como se segue. Na seção 2 é descrito o modelo de jogo suportado por essa ferramenta. A seção 3 apresenta a arquitetura da ferramenta proposta. As principais funcionalidades da ferramenta de autoria são descritas na seção 4. Por fim, a seção 5 conclui o artigo.

## 2. MODELO DE JOGO

Para obtenção dos cenários para os JMBLs, além do levantamento bibliográfico, foi utilizada uma técnica de coleta de dados qualitativos, chamada de Grupos Focais. Esta técnica tem por objetivo identificar as percepções, sentimentos, atitudes e ideias de um grupo de indivíduos a respeito de um determinado assunto, produto ou atividade [2]. Nos grupos focais buscou-se respostas para assuntos relevantes à esta pesquisa, tais como os tipos de desafios que deveriam ser propostos aos jogadores e como aplicar a realidade aumentada nesta modalidade de jogo.

Baseado nos cenários de JMBLs extraídos dos Grupos Focais e no levantamento bibliográfico, foi projetada uma ferramenta de autoria que desse suporte a esse tipo de jogo. Para esse trabalho tais jogos são concebidos como um conjunto de missões a serem cumpridas por um único jogador ou por um grupo de jogadores. Estas missões podem ser projetadas para serem executadas em uma ordem pré-definida ou aleatoriamente. A ferramenta de autoria fornece suporte para criação de jogos com missões ordenadas e não ordenadas, ou mesmo parcialmente ordenadas, onde apenas algumas missões têm ordem definida.

No projeto de jogo cada missão representa uma ou muitas ações que um ou vários jogadores devem executar, e cuja realização no mundo real tem reflexo no mundo virtual. Como mencionado previamente, essas ações incluem a visualização e captura de objetos virtuais por jogadores, de acordo com o objetivo do jogo. Cada ação representa uma atividade básica de um JMBL, chamada em nosso projeto de *mecânica*. De acordo com [7], uma mecânica é qualquer parte de um sistema de regras de um jogo que abrange um, e apenas um, possível tipo de interação que ocorre durante o jogo, seja ela geral ou específica. Dentro do modelo proposto na ferramenta, uma mecânica representa o tipo mais básico de interação que um jogador pode realizar. A partir do resultado dos grupos focais, foi definido um conjunto de mecânicas básicas que podem ser suportadas pela ferramenta, são elas: ir a um local específico no mapa, visualizar, capturar ou deixar um determinado objeto. Objetos variam entre textos, imagens, vídeos, sons ou elementos em 3D, onde estes últimos são objetos de realidade aumentada. Uma mecânica pode ser combinada de maneira a produzir interações mais complexas chamadas de mecânicas compostas. Um exemplo de uma mecânica composta seria ir a um determinado ponto em um mapa, capturar um objeto 3D e deixar uma mensagem de texto vinculada ao local.

O modelo do jogo ainda possibilita que um conjunto de missões seja exclusivo de um único grupo de jogadores ou compartilhado entre vários grupos. Dessa forma podem ser gerados jogos com todas as missões compartilhadas, jogos com parte das missões compartilhadas e jogos sem missões compartilhadas. Com esse modelo de distribuição de missões entre os grupos é possível criar jogos competitivos e colaborativos. Jogos competitivos são aqueles em que grupos competem para realizar um conjunto de missões mais rapidamente que outros grupos, enquanto que nos jogos colaborativos grupos colaboram para realizar missões de interesse comum.

## 3. ARQUITETURA

Com base no modelo de JMBL proposto, uma arquitetura de suporte a estes jogos foi elaborada. Esta arquitetura de

software tem por objetivo gerenciar o ciclo de vida de um JMBL (segundo o modelo proposto), desde sua criação até sua execução no dispositivo do jogador. A arquitetura está dividida em três componentes principais: (i) editor, (ii) aplicação móvel e (iii) servidor do jogo. Esta arquitetura é baseada no modelo cliente/servidor, onde o editor do jogo e o aplicativo móvel são clientes do servidor do jogo. A Figura 1 apresenta a arquitetura da ferramenta proposta.

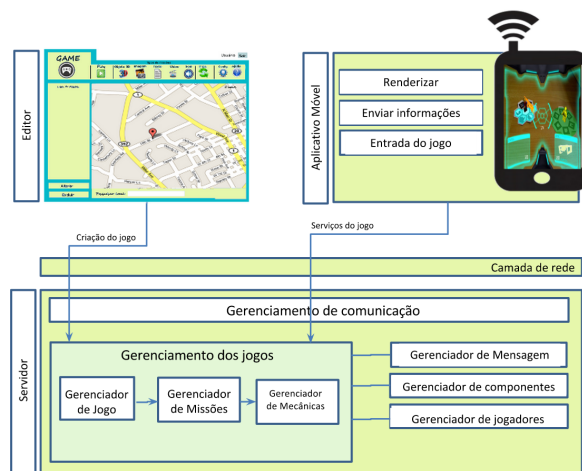


Figura 1: Arquitetura da Ferramenta de Autoria para JMBLs.

O editor é uma aplicação Web que permite aos seus usuários configurar, por meio de notações gráficas, as missões e mecânicas a ser realizadas em um jogo. Para a construção de cada jogo, os autores definem em um mapa o conjunto de missões e mecânicas com base nos locais onde estas devam ser executadas.

Para ter acesso aos jogos publicados no editor, os jogadores devem utilizar um aplicativo móvel desenvolvido na plataforma Android, que acessa as informações dos jogos disponíveis e envia para o servidor a localização do jogador com base no sensor de GPS de seu dispositivo móvel. Ao receber os dados de um jogo, o aplicativo móvel representa o estado atual de suas missões através de ícones pré-definidos, no caso, bandeiras coloridas.

Cada bandeira representa um local no mapa onde está localizada uma mecânica, assim os jogadores devem se deslocar até este local para executar a mecânica correspondente. Sendo que, uma bandeira verde representa uma mecânica que ainda não foi executada, uma bandeira vermelha significa uma mecânica bloqueada (existem outras mecânicas que precisam ser concluídas para desbloqueá-la), e as mecânicas finalizadas são representadas por bandeiras azuis. Esta estratégia permite que a mesma aplicação móvel possa ser usada nos diferentes jogos gerados pela ferramenta. O aplicativo móvel informa ao servidor através da Internet quais ações foram executadas pelo jogador.

O terceiro componente é o servidor do jogo, que é responsável pela gestão de informações sobre os jogos e os jogadores. O servidor é composto por seis componentes, sendo eles: (i) Gerenciador de comunicação - responsável pela troca de informações entre os dispositivos dos jogadores e o servidor, (ii) Gerenciador de jogos - componente que coordena instâncias dos jogos e seu estado de execução, re-

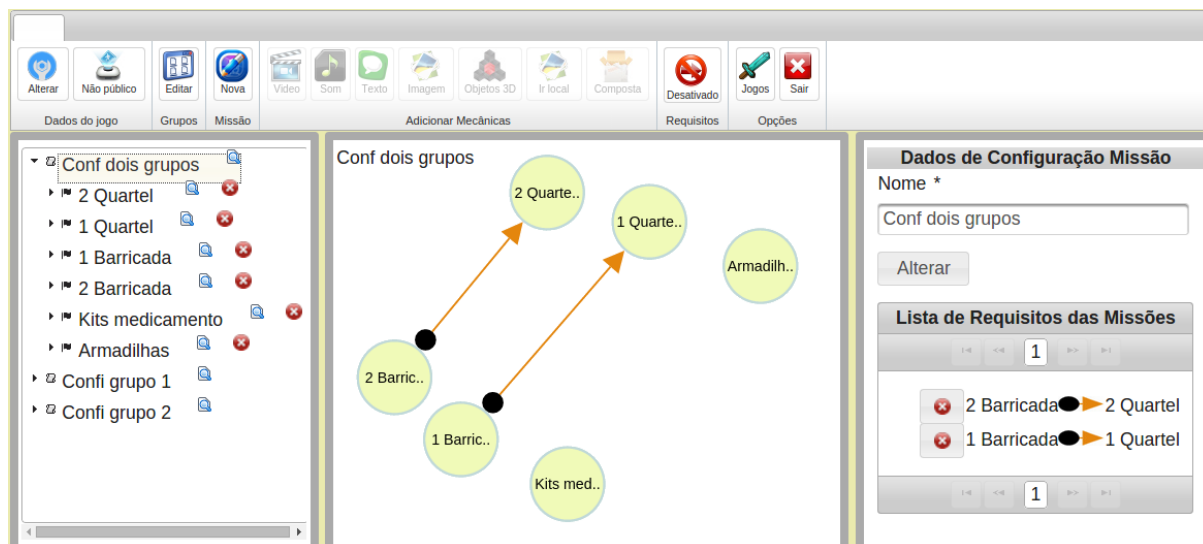


Figura 2: Tela - Menu principal da Ferramenta de Autoria

tornando o estado atual do jogo aos jogadores sempre que solicitado, (iii) Gerenciador de missões - responsável por gerenciar as missões e suas pré-condições, de acordo com cada jogo, (iv) Gerenciador de mecânicas - que controla a sequência de execução das mecânicas, (v) Gerenciador de mensagens - cuja função é permitir a troca de mensagens de texto entre jogadores durante o jogo, e (vi) Gerenciador de jogadores - cujo objetivo é controlar o perfil e o status dos jogadores nos jogos que eles participam.

O servidor utiliza um sistema gerenciador de banco de dados relacional para armazenar todos os dados sobre os jogos, suas configurações, objetos, assim como jogadores e autores. Arquivos de mídia, como áudio ou vídeo são armazenados em sistema de arquivos, sendo que o banco de dados mantém apenas as informações sobre os diretórios onde estão guardados cada documento.

#### 4. PRINCIPAIS FUNCIONALIDADES

Ao acessar o editor do jogo, autores tem acesso à lista de jogos criados previamente, podendo editá-los ou então, desenvolver um novo. O primeiro passo para criação de um novo jogo é definir o número de grupos e o lugar onde o mesmo ocorrerá. O segundo passo é configurar os grupos de jogadores, bem como os conjuntos de missões que cada grupo deve realizar. Por padrão, haverá sempre um conjunto de configurações e uma equipe, sendo que é possível compartilhar uma configuração (conjunto de missões) entre mais de um grupo.

Um jogo que apresente mais de um grupo e um conjunto de missões diferentes, implicará em um jogo competitivo. Neste caso o grupo de jogadores que concluir primeiramente as missões designadas será considerado o grupo vencedor. Porém, também é possível criar jogos com missões compartilhadas, ou mesmo um modelo híbrido, em que parte das missões são compartilhadas, enquanto outras são específicas a cada grupo do jogo.

Na definição dos grupos de um jogo, a ferramenta impõe que os jogadores sejam classificados em tipos pré-definidos. Estes tipos variam entre: “neutro”, que indica que o jogador não pode capturar e nem ser capturado por outros; “cap-

turável”, que pode ser capturado mas nunca poderá capturar outro jogador; “capturador”, que pode capturar não pode ser capturado e o por último, “híbrido”, que é capaz de capturar e ser capturado.

O terceiro passo é a definição dos nomes das missões para cada configuração, além das mecânicas que fazem parte de cada uma destas configurações, como apresentado na Figura 2. Os conjuntos de configurações e cada missão desses conjuntos são exibidos no lado direito da tela do editor. Uma barra de ferramentas com as principais ações que os autores do jogo podem executar é exibida na parte superior do editor. No lado esquerdo são apresentadas informações específicas da missão que está sendo editada, como a sua lista de requisitos.

Missões, assim como as mecânicas, são ilustradas como elipses, que podem ser conectadas através de setas para indicar a ordem em que as mesmas devem ser executadas. Esta forma de ordenar as missões é definida na ferramenta como uma lista de requisitos, que permite ao usuário criar jogos flexíveis quanto à ordem de execução. Por exemplo, em jogos que não possuem uma ordem de execução estabelecida, não existe lista de requisitos de cada missão, logo, todas as missões podem ser executadas em qualquer momento, em jogos totalmente ordenados, a lista de requisitos de cada missão contém as missões que precisam ser executadas previamente, isto faz com que os jogadores sigam uma sequência pré-estabelecida. É possível também criar jogos parcialmente ordenados, bastando que se atribua uma lista de requisitos às missões que se deseja ordenar. Na Figura 2 são mostradas as missões kit de medicamento e armadilhas, ambas sem uma ordem para sua execução, já as missões *Barricada* somente podem ser executadas após as missões *Quartel*, pois *Quartel* encontram-se em suas listas de requisitos. As listas de requisitos também foram implementadas em mecânicas, permitindo também a definição de ordens de execução.

Uma atribuição importante que pode ser feita às missões na ferramenta é quanto aos grupos que têm permissão de realizá-las. Se uma missão é compartilhada por mais de um grupo mas apenas um pode realizá-la, os outros grupos apenas visualizarão o estado desta missão. Esta característica

é fundamental para criação de um jogo colaborativo, com missões que são executadas por todos os grupos e missões que são executadas somente por grupos específicos.

Na parte superior do editor, uma barra de ferramentas oferece a inclusão de missões, mecânicas, bem como alterações de dados das configurações já realizadas previamente. Cada botão da barra de ferramentas oferece um botão para a inclusão de mecânicas relacionadas aos diferentes tipos de mídia suportados pela ferramenta, a saber, texto, áudio, vídeo e objetos 3D. Na barra de ferramentas também há uma opção para configurar o deslocamento do jogador a qualquer ponto no mapa. Escolhendo uma destas opções, uma janela é aberta, onde devem ser fornecidas informações adicionais para completar a configuração da mecânica. Por exemplo, a Figura 3 mostra a configuração para a visualização de um objeto de realidade aumentada.

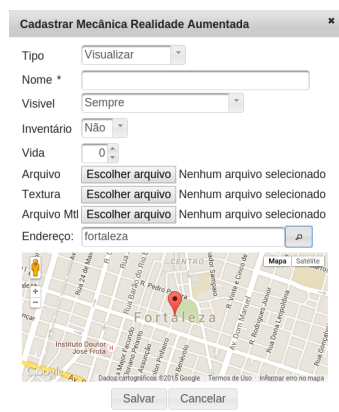


Figura 3: Tela - Adicionar uma mecânica de RA

As opções de inventário e vida indicam, respectivamente, se os objetos ligados à mecânica em questão serão anexados ou removidos do inventário do jogador, e se haverá ganho ou perda de pontos, quando a mecânica é realizada. A opção de visibilidade permite quatro tipos: (i) “sempre”, quando a mecânica estará sempre visível, não importa em que estado a mesma se encontre, (ii) “nunca visível”, quando a mecânica só será ativada quando um jogador se aproximar do local associado, porém sem nunca se tornar visível, (iii) “desbloqueado”, que significa que uma mecânica será visível apenas quando a mesma estiver no estado “pronta”, ou seja, todas as mecânicas que estavam na sua lista de requisitos foram realizadas e ela já pode ser executada, e (iv) “ao realizar”, indica que uma mecânica deve se tornar invisível quando mudar para o estado de finalizada, ou seja, ela é executada quando o jogador se aproximar e permanece visível ao ser finalizada.

A barra de ferramentas também oferece opções para criar mecânicas compostas, que são concebidas pela união entre duas ou mais mecânicas, chamadas então de sub-mecânicas. Esse tipo de mecânica permite aos usuários criar, dentro de uma missão, um conjunto de tarefas mais complexas, pois é necessário que o jogador execute todas as sub-mecânicas inseridas nas compostas. As sub-mecânicas contidas na mecânica composta podem ser visualizadas como um modelo comum de missão, ajudando os autores a economizar espaço físico para a execução de atividades e fornecendo uma alternativa para diagramas complexos.

Ao finalizar a construção de um jogo, o autor tem disponível

na barra de ferramentas, um botão para publicação do jogo. Esta ação disponibilizará o jogo para ser acessado por qualquer usuário, a partir do aplicativo instalado em seu dispositivo móvel.

## 5. CONCLUSÃO

Este artigo apresentou uma ferramenta web para construção de jogos móveis baseados em localização. Uma contribuição fundamental deste trabalho é o desenvolvimento de uma ferramenta de autoria que pode ser usada por usuários sem experiência em programação, como professores, designers de jogos, etc, para projetar uma grande variedade de jogos desta modalidade. Além disso, a ferramenta permite a utilização de conteúdos digitais múltiplos, incluindo objetos de realidade aumentada, que podem ser inseridos no jogo, aumentando assim a imersão do jogador.

Uma característica que diferencia esta ferramenta de autoria das demais é a maneira como as missões são ordenadas, permitindo aos seus usuários criarem diversos tipos de missões, ordenadas ou não. Uma outra peculiaridade da ferramenta é permitir a criação de jogos colaborativos e competitivos, enfatizando que essa possibilidade torna o domínio de jogos gerados por ela maior que as demais encontradas nesse segmento. Um vídeo no YouTube<sup>2</sup> encontra-se disponível esclarecendo as principais funcionalidades dessa ferramenta.

## 6. REFERÊNCIAS

- [1] Basic Structure of ARIS - ARIS - The Manual. <http://manual.arisgames.org/>. Accessed: 2014-02-12.
- [2] I. M. Aschidamini and R. Saube. Grupo focal: estratégia metodológica qualitativa: um ensaio teórico. *Cogitare enferm*, 9(1):9-14, 2004.
- [3] R. Ballagas and S. Walz. Rexplorer: Using player-centered iterative design techniques for pervasive game development. *Pervasive Gaming Applications*, 2, 2007.
- [4] I. Celino, S. Contessa, M. Corubolo, D. Dell’Aglia, E. Della Valle, S. Fumeo, T. Krüger, and T. Krüger. Urbanmatch-linking and improving smart cities data. In *LDOW*, 2012.
- [5] J. do Prado Rafalski, O. L. dos Santos, and C. S. de Menezes. Um editor colaborativo para descrição de aventuras pedagógicas locais com realidade ampliada. pages 283-288, 2013.
- [6] P. Ihamäki and M. Luimula. Let’s go geocaching: Understanding users enjoyment in geocentric application. 2013.
- [7] S. Lundgren and S. Bjork. Game mechanics: Describing computer-augmented games in terms of interaction. In *Proceedings of TIDSE*, volume 3, 2003.
- [8] I. Rubino, J. Xhembulla, A. Martina, A. Bottino, and G. Malnati. Musa: Using indoor positioning and navigation to enhance cultural experiences in a museum. *Sensors*, 13(12):17445-17471, 2013.
- [9] R. Wetzell, L. Blum, and L. Oppermann. Tidy city: a location-based game supported by in-situ and web-based authoring tools to enable user-created content. In *Proceedings of the international conference on the foundations of digital games*, pages 238-241. ACM, 2012.

<sup>2</sup><https://youtu.be/8KmFmyFnEqw>