

Paulo Koji Ribeiro Serita

**UM AMBIENTE DE AUTORIA DE JOGOS
EDUCACIONAIS EPISTÊMICOS**

Vitória, ES

2019

Paulo Koji Ribeiro Serita

UM AMBIENTE DE AUTORIA DE JOGOS EDUCACIONAIS EPISTÊMICOS

Dissertação submetida ao Programa de PósGraduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Informática.

Orientador: Prof. Dr. Crediné Silva de Menezes

Vitória, ES

2019

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Setorial Tecnológica,
Universidade Federal do Espírito Santo, ES, Brasil)

S485a Serita, Paulo Koji Ribeiro, 1988-
Um ambiente de autoria de jogos educacionais epistêmicos / Paulo
Koji Ribeiro Serita. – 2019.
87 f. : il.

Orientador: Crediné Silva de Menezes.
Dissertação (Mestrado em Informática) – Universidade Federal do
Espírito Santo, Centro Tecnológico.

1. Jogos educativos. 2. Jogos eletrônicos. 3. Software educacional. 4.
Jogos para computador. 5. Jogos no ensino de matemática. 6. Jogos
para celular. I. Menezes, Crediné Silva de. II. Universidade Federal
do Espírito Santo. Centro Tecnológico. III. Título.

CDU: 004

Paulo Koji Ribeiro Serita

UM AMBIENTE DE AUTORIA DE JOGOS EDUCACIONAIS EPISTÊMICOS

Dissertação submetida ao Programa de PósGraduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Informática.

Aprovada em 07 de fevereiro de 2019.

COMISSÃO EXAMINADORA

Prof. Dr. Crediné Silva de Menezes
Universidade Federal do Espírito Santo (UFES)
(Orientador)

Prof. Dr. Orivaldo de Lira Tavares
Universidade Federal do Espírito Santo (UFES)
(Membro Interno)

Prof. Sergio Crespo Coelho da Silva Pinto
Universidade Federal Fluminense (UFF)
(*Membro Externo*)

Aos meus pais, Nobuyuki e Ana Maria, e minha irmã, Rafaela.

AGRADECIMENTOS

Agradeço primeiramente a minha família, os meus pais que acreditaram em mim muito mais do eu mesmo, me dando o suporte e incentivo que foram fundamentais nessa parte de minha jornada. E a minha irmã a quem sempre ocupava o tempo quando precisava explicar algo para entendimento próprio.

Agradeço ao meu orientador Prof. Dr. Crediné Silva de Menezes, que teve a paciência de guiar alguém como quase nenhuma experiência acadêmica assim como fez durante os momentos de dificuldade que passei durante este período.

Fico grato aos amigos e colegas estudantes cuja genialidade e disposição me serviram de inspiração para alcançar meus próprios objetivos.

RESUMO

O desenvolvimento de um bom jogo requer do autor - além de expertise da temática abordada - entender o público alvo, ser capaz de produzir uma arte adequada a proposta e possuir conhecimentos em design de jogos. Requisitos ainda mais relevantes se tratando de jogos educacionais e o design instrucional, onde a má adequação do jogo há um conteúdo educacional, pode, além de descaracterizar um jogo, não atingir os objetivos educacionais esperados. Este trabalho apresenta a proposta de um ambiente de autoria para jogos educacionais com base nos conceitos gerais de design de jogos e da teoria dos Jogos Epistêmicos como base para produção de jogos educacionais.

Tal teoria apresenta o conceito de Quadros Epistêmicos, que descreve o processo de recorte de uma cultura, ambiente, atividades e ferramentas que compõem uma profissão. A organização desses a fazer sentido quanto a uma atividade educacional e as aprendizagens esperadas em um contexto que não o habitual dos participantes. O processo de transformação do recorte em jogo resulta no Jogo Epistêmico, portanto, não se limita ao design instrucional, mas também ao planejamento dos componentes à serem definidos no design do jogo.

O ambiente de autoria proposto foi dividido em três aplicações, seguindo o modelo cliente servidor com uma estrutura interna organizada por módulos e base. A base dita modelos de comportamento aos módulos estabelecendo a comunicação entre si. As funcionalidades do ambiente foram obtidas de análises da teoria, jogos epistêmicos produzidos, outros ambientes existentes e do design de jogos. Cada módulo pode representar uma atividade através de um objeto ou personagem, no caso dos jogos epistêmicos, as atividades do recorte de uma profissão.

O resultado final é um ambiente capaz de gerar jogos epistêmicos de forma mais simples e barata, não exigindo do autor conhecimentos em programação ao simplificar o acesso a recursos tecnológicos como o motor de jogo e o ambiente de desenvolvimento ao custo da liberdade criativa e em prol da produção de um jogo que siga o modelo dos jogos epistêmicos.

Palavras-chave: jogo epistêmico. ambiente de autoria. quadro epistêmico. design de jogo.

ABSTRACT

The development of a good game requires from the author - besides the expertise of the subject - to understand the target audience, to be able to produce an art appropriate to the proposal and have knowledge in game design. Even more relevant are the requirements for educational games and instructional design, where the poor choice of the game's educational content, may, in addition to mischaracterizing a game, not achieve the expected educational objectives. This paper presents a proposal of an authoring environment for educational games based on the general concepts of game design and the Epistemic Games theory as a basis for the development of educational games.

This theory presents the concept of Epistemic Frames, which describes the process of making a snippet of a culture, environment, activities and tools that make up a profession. The organization of these to make sense as an educational activity and the expected learning in a context other than which the participants are used. The process of transforming the snippet into a game results in the Epistemic Game, therefore, it is not limited to the instructional design, but also to the planning of the components to be defined in the game design.

The proposed authoring environment was divided into three applications, following the client server model with an internal structure organized by modules and base. The base dictates behavioral to the modules by establishing communication with each other. The functionalities of the environment were obtained from analyzes of theory, epistemic games produced, other existing environments and game design. Each module can represent an activity through an object or character, in the case of epistemic games, the activities of a profession snippet.

The end result is an environment capable of generating epistemic games in a simpler and cheaper way, not requiring from the author programming knowledge by simplifying his access to technological resources such as game engine and development environment at the cost of creative freedom for the production of a game that follows the model of epistemic games.

Keywords: epistemic game. authoring environment. epistemic frame. game design.

SUMÁRIO DE FIGURAS

Figura 1: Telas dos editores: ITystudio (à dir.) e VTS (à esq.).	27
Figura 2: Tela do MATCHS.	28
Figura 3: Tela do jogo criado com o plugin.	29
Figura 4: Tela de edição do eAdventure.	30
Figura 5: Tela do SodaConstructor.	35
Figura 6: Tela do Land Science.	35
Figura 7: Tela da aplicação especialista usada no RescuShell.	36
Figura 8: Tela da aplicação especialista usada no Nephrotex.	37
Figura 9: Diagrama do ciclo de um jogo epistêmico.	40
Figura 10: Elementos dos jogos epistêmicos agrupados em conjuntos.	41
Figura 11: Fluxo para desenvolvimento de um jogo epistêmico.	42
Figura 12: Diagrama das atividades do jogo.	46
Figura 13: Diagrama final das atividades do jogo mesclado com jogos epistêmicos.	47
Figura 14: Organização editor/jogo.	48
Figura 15: Organização editor, núcleo e cliente	49
Figura 16: Montagem de um jogo no editor.	52
Figura 17: Caso de uso Aplicação Núcleo.	56
Figura 18: Diagrama de classe da Aplicação Núcleo.	57
Figura 19: Diagrama entidade relacional da Aplicação Núcleo.	58
Figura 20: Caso de uso Aplicação Editor.	60
Figura 21: Diagrama de classe da Aplicação Editor.	61
Figura 22: Diagrama entidade relacional da Aplicação Editor.	62
Figura 23: Fluxo da Aplicação Cliente.	63
Figura 24: Caso de uso Aplicação Cliente.	64
Figura 25: Diagrama de classe da Aplicação Cliente.	65
Figura 26: Modelo do comportamento básico da Aplicação Cliente.	66
Figura 27: Tela de login da aplicação cliente.	67
Figura 28: Tela de salas (jogos criados) da aplicação cliente.	67
Figura 29: Caso de uso Aplicação Cliente, Base.	69
Figura 30: Diagrama de classe da Aplicação Cliente, Base.	70
Figura 31: Menu da tela de jogo.	70
Figura 32: Caso de uso Aplicação Cliente, TileMap.	72
Figura 33: Diagrama de classe da Aplicação Cliente, TileMap.	72
Figura 34: TileMap, Editor (à esq.) e Cliente (à dir.).	73
Figura 35: Caso de uso Aplicação Cliente, Módulo Expositor.	74
Figura 36: Diagrama de classe da Aplicação Cliente, Módulo Expositor.	75
Figura 37: Tela: menu de expositor.	76
Figura 38: Caso de uso Aplicação Cliente, Módulo Fornecedor.	77
Figura 39: Diagrama de classe da Aplicação Cliente, Módulo Fornecedor.	77

Figura 40: Tela: menu de fornecedor.	78
Figura 41: Caso de uso Aplicação Cliente, Módulo Cliente.	79
Figura 42: Diagrama de classe da Aplicação Cliente, Módulo Cliente.	80
Figura 43: Tela: informações de clientes.	81
Figura 44: Tela: mensagem inicial.	82
Figura 45: Tela: informações de clientes simples.	83
Figura 46: Tela: requisição de informações ao jogador.	84

SUMÁRIO

1. INTRODUÇÃO	13
1.1. Motivação	16
1.2. Problemática e Hipóteses	17
1.3. Objetivos.....	18
1.4. Metodologia.....	19
1.5. Organização	20
2. JOGOS DIGITAIS E EDUCAÇÃO.....	22
2.1. Desenvolvimento de jogos	22
2.2. Tempo de desenvolvimento e equipes	24
2.3. Ambientes de autoria	25
2.4. Outros ambientes de autoria	26
2.5. Análise dos ambientes de autoria	30
3. JOGOS EPISTÊMICOS	32
3.1. Exemplos de jogos epistêmicos	34
3.2. Análise dos trabalhos correlatos	37
4. MODELO PARA DESIGN INSTRUCIONAL DE JOGOS EPISTÊMICOS.....	39
5. CONCEPTUALIZAÇÃO DE UM JOGO EPISTÊMICO	44
6. PROPOSTA DE UM AMBIENTE DE AUTORIA	48
6.1. Aplicação Núcleo.....	49
6.2. Aplicação cliente	50
6.2.1. Arquivos de configuração	53
6.3. Aplicação Editor.....	53
7. PROTOTIPAÇÃO DO AMBIENTE	55
7.1. Protótipo da aplicação núcleo	55
7.1.1. Requisitos funcionais	55
7.1.2. Modelo de classe.....	57
7.1.3. Modelo entidade relacional	58
7.1.4. Implementação	58
7.2. Protótipo da aplicação editor.....	59
7.2.1. Requisitos funcionais	59
7.2.2. Modelagem.....	60
7.2.3. Implementação	61
7.2.4. Editor TileMap	62
7.3. Protótipo da aplicação cliente	63

7.4.1. Requisitos funcionais	63
7.4.2. Modelagem.....	64
7.4.3. Implementação	65
7.5. Base	68
7.5.1. Requisitos funcionais	68
7.5.2. Modelagem.....	69
7.5.3. Implementação	70
7.6. Módulo TileMap	71
7.6.1. Requisitos funcionais	71
7.6.2. Modelagem.....	72
7.6.3. Implementação	73
7.7. Módulo Expositor.....	73
7.7.1. Requisitos funcionais	73
7.7.2. Modelagem.....	74
7.7.3. Implementação	75
7.8. Módulo Fornecedor	76
7.8.1. Requisitos funcionais	76
7.8.2. Modelagem.....	77
7.8.3. Implementação	78
7.9. Módulo Cliente.....	78
7.9.1. Requisitos funcionais	78
7.9.2. Modelagem.....	79
7.9.3. Implementação	80
8. DESENVOLVIMENTO DO JOGO NO AMBIENTE.....	82
9. CONSIDERAÇÕES FINAIS	85
9.1. Trabalhos futuros	86
10. REFERÊNCIAS BIBLIOGRÁFICAS.....	88

1. INTRODUÇÃO

No projeto de desenvolvimento de um jogo eletrônico, uma das figuras de destaque é o game designer, designer de jogo, profissional cujas decisões afetam todo o processo de desenvolvimento. Responsável por definir os elementos componentes de um jogo, este se especializa em diversas áreas de conhecimento a fim de transformar ideias em um projeto de jogo.

A profissão de designer de jogos é recente, considerando que os próprios jogos eletrônicos comerciais só agora estão alcançando os 50 anos de existência. Do Odyssey [Magnavox, 1972] em 1972 até os tempos de hoje o desenvolvimento de jogos assim como ocorrido em outras áreas tecnológicas sofreu e se beneficiou das constantes evoluções. Nesse período os jogos passaram de abstrações limitadas a quadros de 256 por 240 pixels e 432 cores a ambientes tridimensionais capazes de calcular comportamentos físicos de elementos como vento, água e neve.

A evolução tecnológica deu origem não somente a ferramentas mais poderosas mas também simplificou o desenvolvimento dos jogos, a exemplo, em 1983 no NES [Nintendo, 1983], Nintendo Entertainment System, o artista possuía uma limitação da quantidade de pixels no qual poderia produzir suas criações e o programador um formato e quantidade de memória limitada para o armazenamento das artes, obrigando um intenso trabalho em equipe para que conseguisse tirar o maior proveito possível do hardware disponível. Hoje o mesmo conjunto de operações pode se resumir a importar uma imagem e posicionar onde está deve aparecer nas telas do jogo.

Retomando à época do lançamento do NES, em 1984, Chris Crawford [Crawford, 1997] produz uma publicação no qual realiza uma compilação dos principais jogos lançados até a dada data, fornecendo uma análise dos estilos de jogos, suas características, como foram desenvolvidos. Podendo ser caracterizado como uma tentativa inicial de se obter um padrão no desenvolvimento de jogos.

Entretanto, as tecnologias evoluíram em passos acelerados, dando origem a novas formas de desenvolvimento, tornando difícil a popularização e formalização de um modelo para projeto de jogos até meados do fim da primeira década do século 21. Deste ponto em diante ocorre uma desaceleração na criação de novas tecnologias

e uma popularização de ferramentas para produção de jogos conhecidas como game engine, motor de jogos. Até então, o desenvolvimento de um bom jogo dependia do bom senso, bagagem de conhecimento, vivência e aptidão artística da equipe envolvida em sua produção. Entretanto, cada jogo desenvolvido até então contribuiu para a construção de uma cultura de jogos eletrônicos, e seus casos de sucessos e fracassos serviram como referência para as formalizações ocorridas. De 2003 a 2007, Paul Gee [Gee, 2003] e Raph Koster [Koster, 2004], produzem, individualmente, livros que demonstram a relação dos jogos, o lúdico e a aprendizagem. Exemplificadas por teorias cognitivas como as de Piaget e Vygotsky, em situações onde um jogo gera desequilíbrio no jogador ao buscar formas de provocá-lo a resolver um problema ou em situações onde diferentes jogadores interagem para superar um desafio, seja de forma cooperativa quanto colaborativa [Castorina, 1990]. Experiências pessoais dos autores em suas respectivas áreas e vivências, relacionando-os também a exemplos de jogos, seus diferentes estilos e os diferentes processos de aprendizagem no decorrer de um jogo, como e por que tal aprendizagem ocorre de forma lúdica, recompensadora e muitas vezes imperceptível.

Neste mesmo período ocorrem diferentes publicações de livros focados na formalização do processo de produção de um jogo eletrônico, especificando profissões, composição e relacionamento de equipes, documentações, da psicologia de uma atividade recompensadora à frustração e ética, padrões de interface gráfica, ciclos e etapas de desenvolvimento até a publicação e publicidade.

Ocorre também o aumento no volume de informação disponível sobre desenvolvimento de jogos e o barateamento do custo de produção, resultado do surgimento de novas e sofisticadas game engines de baixo custo, que excluem a exigência de um programador e capazes de produzir jogos de qualidade similar ao de grandes empresas do mercado de jogos eletrônicos.

Tal acesso à informação e tecnologia fomentou a partir de 2009 o movimento dos “jogos indie”, jogos independentes, e uma nova onda de amadorismo. O sucesso de alguns jogos indies produzidos por individuais ou pequeno grupos, como de sua característica, popularizou o mercado do desenvolvimento de jogos. E como

resultado do aumento no volume de jogos produzidos, ocorreu igualmente um aumento dos jogos de baixa qualidade.

Em paralelo, os jogos educacionais, que não se deve confundir com o uso de jogos na educação, avançaram quanto a teorias e modelos de desenvolvimento de jogos. A exemplo, a teoria dos jogos e quadros epistêmicos [Shaffer, 2006], baseado em outras teorias como a aprendizagem situada [Brown, 1989] do qual define o processo do aprendizado a partir da prática, anterior a teoria, seguida de reflexões realizadas feitas durante e posterior a atividade. Comunidade de práticas [Lave, 1991], na qual dentro de um contexto colaborativo se inicia como aprendiz e através da reprodução das práticas por membros mais experientes e troca de informações vai-se evoluindo até alcançar o domínio de uma habilidade ou conhecimento.

As ilhas de especialização [Crowley, 2002] e o uso de tópicos nos quais ocorre um profundo interesse propiciando uma aprendizagem relevante. E pedagógica praxis [Shaffer, 2004a] a qual entende que os conhecimentos e habilidades aprendidos através da prática de atividades autênticas podem ser transferidos ou aproveitados por outras áreas de conhecimento. Propõem na prática um modelo não para a criação de um jogo, mas de um conteúdo, temática, contexto, dialético e ambientação, sobre o qual pode ser desenvolvido um jogo eletrônico.

O conteúdo de um jogo, em geral, pode variar de situações do dia a dia para aventuras fantasiosas ou séries de desafios intelectuais. Não há regras para o que pode ser transformado em um jogo, entretanto, considerado o tempo a ser despendido a aprender uma nova temática e transformá-la em um jogo, pequenos desenvolvedores tendem limitar-se a situações que lhe são familiares, conhecimentos já existentes ou histórias fantasiosas. Em casos mais complexos como um jogo cuja temática sejam fatos históricos, a presença de especialistas, como historiadores, se faz essencial para uma reprodução o mais fiel possível da passagem histórica.

Os jogos educacionais possuem um objetivo pré-definido, a prática de uma habilidade ou a aprendizagem de um conteúdo, este sobrepõe todos os demais elementos de um jogo e deve ser tratada como uma constante no decorrer de uma partida, as ações do jogador devem ser significativas e direcionadas a geração de conhecimento.

O jogo não somente deve ensinar como utilizar as ferramentas entregues para resolução dos desafios requeridos para a completude do próprio, mas também como fazer tirar proveito destas aprendizagens para aquisição de conhecimentos no mundo real. Para tal é importante no desenvolvimento de um jogo educacional o design de conteúdo educacional, e, portanto, de conhecimento em design instrucional e da área de conhecimento ao qual serão feitas as aprendizagens. Para que, em teoria, seja elaborado um bom jogo que consiga transmitir de forma adequada o conteúdo que se almeja ser aprendido pelo jogador.

Nesse contexto a adoção de um modelo bem definido para a criação de conteúdo educacional conforme proposto pela teoria dos jogos epistêmicos poderia suprir a ausência de um designer instrucional, profissional responsável por produzir conteúdo educacional.

Adicionar este modelo a um ambiente, que simplifique a implementação de um jogo, proporciona ao especialista do conteúdo a ser ensinado - seja ele um professor, aluno, hobbista ou entusiasta - a capacidade de produzir um jogo educacional independente dos demais profissionais, em um menor tempo e a um menor custo.

Este trabalho propõe um ambiente de autoria para jogos educacionais utilizando o modelo de criação de conteúdo educacional proposto pela teoria dos jogos epistêmicos e o adaptando à conceitos de design de jogos.

1.1. Motivação

Antes da grande popularidade da cultura dos jogos eletrônicos do século 21, já se notava o interesse no uso dos jogos eletrônicos na educação, ainda que na forma de entretenimento focado no público infantil com representações digitais de brinquedos educacionais já existentes. No demais, a grande maioria dos jogos produzidos se resumem a mascarar um conteúdo a ser aprendido em estilo de jogo popular e de baixo custo ou em uma coletânea de minijogos no intuito apenas de usufruir dos benefícios observados nos jogos para entretenimento.

Entretanto é vital que observar a disparidade entre entretenimento e arte [Koster, 2012] e como afetam a experiência do jogador em relação ao jogo, enquanto

elementos como dificuldade reduzida, repetições, conservadorismo, normas sociais, acessibilidade, prazer, baixas cargas cognitivas aderem ao entretenimento, intensa dificuldade, riscos, inusitado, empatia, interpretação, fracasso e alta carga cognitiva aderem a arte.

A arte neste caso se refere às nuances que podem tornar estes elementos como condições de rejeição do jogador em relação ao jogo, um jogo de longos textos e linguagem extremamente formal e polida poderia tornar um jogo monótono, maçante ou até mesmo de difícil compreensão para o jogador. No entanto, reduzido o grau de formalidade e aplicado um contexto apropriado jogos como *Child of The Light* [Ubisoft, 2014] e *Ace Attorney* [Capcom, 2001] conseguiram conquistar o público com estilos de jogos apenas possíveis de se apreciar dada atenção aos textos de diálogos.

Partindo dessa configuração de entretenimento e arte é possível assimilar entretenimento às mecânicas do jogo e a adequação do conteúdo como arte. O que então poderia contribuir para a redução da disparidade entre entretenimento e arte em um cenário de amadorismo, onde o criador do jogo teria o conhecimento do conteúdo, mas pouca ou nenhuma noção de design instrucional ou de jogos? Que modelo de ambiente seria o ideal para atingir tal objetivo? O quanto poderia ser incluído na suposta ambiente sem que se limite demasiadamente a criatividade do autor? Para entender que tipo de ambiente de autoria para jogos educacionais atenderia melhor aos questionamentos realizados foram revisados outros ambientes e modelos de jogo educacionais sem que se encontrasse algo satisfatório. Desse levantamento inicial originou-se a proposta da concepção de um ambiente.

1.2. Problemática e Hipóteses

O desenvolvimento de um jogo por vias tradicionais requer além de conhecimentos em design de jogos, aptidão artística e proficiência em programação. Considerando que existem outras formas menos complexas que a clássica para o desenvolvimento de um jogo, quais foram aplicadas para a composição de jogos epistêmicos? Caso exista, seria a menos custosa para o autor?

Observando os jogos declarados como epistêmicos é possível notar uma similaridade no que se trata do estilo de jogo, roteiro, atividades e personagens, no que é chamado de estágio virtual. Evitando outros estilos de jogos que a princípio aparentam poder ser concebidos a partir da teoria dos jogos epistêmicos. Seriam outros estilos de jogos inviáveis ou apenas não tão eficazes?

1.3. Objetivos

Este trabalho apresenta um ambiente de autoria para jogos educacionais conceitual que isenta o autor da necessidade de possuir conhecimentos de design instrucional, design de jogos, programação e arte, através da adoção do modelo proposto na teoria dos jogos epistêmicos para produção de conteúdo, adequação do modelo aos conceitos modernos de design de jogos, a adição de uma camada sobre um motor de jogo que simplifique o desenvolvimento e o fornecimento de material gráfico.

Resultando em um projeto de ambiente de autoria para jogos educacionais epistêmicos e a confecção de um protótipo a fim de demonstrar a viabilidade do ambiente. E por fim a produção de um jogo no protótipo do qual se espera observar a adequação dos elementos do modelo optado no design instrucional com o jogo. Em ordem de dependência, são objetivos específicos do trabalho:

- Verificar similaridades entre todo processo que envolve um jogo epistêmico e o design de jogos.
- Analisar a viabilidade de aplicar a teoria dos jogos epistêmicos como modelo para criação de jogos.
- Observar a adequação do modelo como ambiente de autoria para desenvolvimento de jogos educacionais epistêmicos.
- Verificar se o uso do ambiente se limita apenas ao desenvolvimento de jogos educacionais epistêmicos.

1.4. Metodologia

Foi realizado inicialmente uma pesquisa em livros, bases de artigos e periódicos por trabalhos relacionados aos tópicos: jogos educacionais, jogos epistêmicos, motores de jogo e ambientes de autoria. Em um primeiro momento buscou-se exclusivamente por trabalhos dentro do tópico jogos epistêmicos, no qual seria baseado o ambiente de autoria. Aprofundado entendimento do que são os jogos epistêmicos foi observado que de forma geral esses jogos se assimilam a simuladores de profissões, o que expandiu as buscas por trabalhos que ligassem jogos a aprendizagem de uma profissão e simuladores.

Em outro momento foram pesquisados trabalhos voltados ao desenvolvimento de ambientes de autoria para jogos educacionais, durante as pesquisas foram encontrados trabalhos dos quais o produto final não era considerado ambiente de autoria, mas motor de jogos. É de entendimento que há uma grande distinção entre esses e que, em geral, um ambiente de autoria utiliza-se de um motor para a geração do jogo. Entretanto, diversos deste trabalham buscavam simplificar a atividade de desenvolvimento de jogos direto no motor e não através de um componente terceiro, e, portanto, poderiam vir a apresentar conceitos e ideias relevantes também à produção de um ambiente de autoria.

O material encontrado foi classificado em um dos tópicos citados, exceto jogos educacionais, este fora utilizado para abrigar trabalhos que não se enquadraram nos demais tópicos. Os trabalhos foram analisados em duas etapas: desenvolvimento do modelo de design instrucional e desenvolvimento do ambiente de autoria.

No desenvolvimento do modelo de design instrucional foram analisados os trabalhos dentro dos tópicos, jogos epistêmicos, jogos educacionais e design de jogos (incluindo educacionais). Foi realizado um mapeamento das características encontradas em jogos também vistas nos jogos epistêmicos, em seguida foi realizado um comparativo dentre os jogos epistêmicos e os demais jogos a fim de obter o que justifica uma classificação distinta dos jogos epistêmicos quanto a jogos dos demais encontrados. Os resultados foram transformados em um modelo de desenvolvimento para jogos educacionais com base nos jogos epistêmicos.

No desenvolvimento do ambiente de autoria, foram analisados os trabalhos classificados como motores de jogo e ambientes de autoria, identificando sucessos e fracassos e utilizando-os para guiar o levantamento de requisitos e concepção das funcionalidades que serviriam para adequar ao ambiente conceitual ao modelo de desenvolvimento proposto.

Do ambiente conceitual foi realizado um recorte de suas principais funcionalidades para a implementação de um protótipo no intuito de comprovar a viabilidade do ambiente. No protótipo foi então projetado um jogo com base no modelo definido e este comparado a outros jogos epistêmicos com objetivo de identificar futuras adições e alterações ao ambiente conceitual não observados previamente

1.5. Organização

A dissertação está organizada em # capítulos com a seguinte distribuição:

1. Introdução - apresentação do tema da dissertação, o contexto, motivação, objetivos e a metodologia utilizada.
2. Jogos digitais e educação - uma breve introdução ao design de jogos e trabalhos relacionados à aprendizagem em jogos; motores e ambientes de autoria para jogos educacionais; exemplos de outros ambientes de autoria correlacionados.
3. Jogos epistêmicos - fundamentação teoria dos jogos epistêmicos, quadros epistêmicos e práxis educacional. E uma revisão de jogos epistêmicos.
4. Modelo para design instrucional de jogos epistêmicos - concepção de um modelo a partir de conceitos já existentes do design de jogos e da teoria dos jogos epistêmicos.
5. Proposta de um ambiente de autoria - uma proposta de ambiente de autoria para jogos.
6. Prototipação do ambiente - descrição do processo de implementação inicial do ambiente a partir do recorte do ambiente projetada.
7. Concepção de um jogo no ambiente - design de um jogo no ambiente prototipado utilizando como temática atividades comuns a um minimercado.

8. Considerações finais - contribuições, limitações do ambiente e adições ou alterações que podem representar futuros trabalhos.
9. Referências - referência bibliográfica do conteúdo apresentado na dissertação.

2. JOGOS DIGITAIS E EDUCAÇÃO

Neste capítulo serão abordados alguns conceitos básicos do desenvolvimento de jogos, o tempo de desenvolvimento em relação a equipe e tamanho do escopo. Terminando pela descrição e exemplos de ambientes de autoria com aplicações semelhantes ao da proposta e uma breve descrição de conceitos de objetos de aprendizagem aproveitados no desenvolvimento do ambiente.

2.1. Desenvolvimento de jogos

Para se desenvolver um bom jogo é necessário acima de tudo conhecer o público alvo. O conceito de bom quando tratando-se de jogos varia bruscamente, enquanto um jogo simples como o Flapbird [GEARS Studio, 2013] - composto por um pássaro que segue para direita de forma constante, sendo a única interação com o jogador o pressionar de um botão que controla a movimentação vertical para desviar de obstáculos - se tornou excepcionalmente popular atingindo a faixa de 50 milhões de jogadores. O mesmo se avaliado pelos critérios para identificação de um bom jogo, definidos por Gee [Gee, 2007], mal atenderia a um dos requisitos. Esta capacidade de entreter os jogadores com pouco, é uma dentre várias características dos jogos que mais atraem a atenção de pesquisadores, desenvolvedores e professores quanto ao uso dos jogos na educação. Tal capacidade de entreter promove o foco no jogo por parte do jogador, o que é similar ao requerido pela teoria do fluxo educacional. Ainda que fórmulas simples como a citada sejam capazes de prender a atenção do jogador. Cada aluno é um ser individual que possui preferências quanto ao estilo de aprendizagem, estilo de jogador e tipo de jogo preferido.

Na intenção de agradar diferentes públicos é comum que se venha a produzir algo que atenda ao mínimo de cada um, entretanto, o caminho para se conseguir atender a diferentes públicos é complexo. Schell [Schell, 2008] descreve uma abordagem passando por três áreas: psicologia, antropologia e design. Sugerindo um pensar que aplique tanto o behaviorismo quanto a fenomenologia, dado que jogos, assim como uma música, um livro, uma pintura ou qualquer outra forma de

arte, dois jogos similares podem provocar diferentes reações de uma mesmo jogador.

Não há uma fórmula para um clássico, mas compreender as possíveis respostas do jogador a certos estímulos e as imprevisibilidades resultante de fatores como intuição ou do instinto e empatia. Entender o ser humano permite o designer prever as respostas dos jogadores e preparar contramedidas, tornando o ato de jogar não apenas uma atividade, mas uma experiência. Para tal é necessário que o designer valorize a empatia e se desprenda de seus interesses pessoais, assim como descrito por Adams [Adams, 2010], dar ênfase ao jogador em cada etapa de criação do jogo, se colocar na sua posição e imaginar como este agiria diante do que se pretende adicionar ao jogo.

Antes de se começar a planejar um jogo deve-se conhecer o assunto central, a partir desse conhecimento é definida a temática, elemento crucial para o desenvolvimento do jogo que, em geral, é o elemento ao qual todos os demais serão definidos. Podem ser fantasiosas ou realistas, exigir conhecimento de diferentes culturas, fatos históricos ou científicos, desafios lógicos, biologia, química, física dentre outros. A temática influencia elementos como a narrativa e diálogos, arte gráfica, arte sonora, regras e mecânicas.

Assim como um livro ou um filme, a história de um jogo pode ser contada de diferentes formas, por diferentes observadores e diferentes estilos de diálogos. A arte gráfica é influenciada diretamente pelo estilo do jogo: podendo variar de um ambiente bidimensional à um tridimensional; de uma aparência realista à algo surreal; a posição da câmera, do ponto de visão do personagem controlado pelo jogador, seguindo o personagem a distância, fornecendo uma visão superior do cenário ou de movimento livre dentro do cenário. A arte sonora afeta como o jogo será percebido pelo jogador. Dos efeitos sonoros ao estilo musical, a escolha correta deste elemento afeta positivamente o nível de imersão do jogador.

As regras transformam uma representação ou trabalho de arte em um jogo. Aqui são definidas as ações suas reações e consequências, os limites do jogador, a navegação do jogador e às diferentes interações jogo-jogo, jogo-jogadores e jogadores-jogadores. É nesta etapa que os elementos artísticos ganham uma definição técnica para que possam ser desenvolvidos, tornando-se mecânicas. A

implementação das mecânicas costuma ser feita sobre um motor de jogos, uma ferramenta responsável por agrupar componentes essenciais na implementação de um jogo, tornando esta tarefa menos complexa do que seria sem esse suporte. Os processos de renderização de imagens e áudio de acordo com o dispositivo onde será executado, a interação entre os elementos e seus comportamentos, cálculos físicos de propagação do som e luz, gravidade, aceleração, atrito, podendo inclusive ter comportamento diferente do mundo real, são algumas das complexidades comumente resolvidas por um motor de jogos.

Em geral, um motor de jogos costuma ser apresentado como um conjunto de bibliotecas de programação para abstração da criação de jogos, composta por algoritmos de renderização gráfica, cálculos de física, gerenciamento da memória, controle de áudio e de adaptadores de entrada. Alguns motores, no entanto, possuem uma interface gráfica, seja para uma melhor visualização do jogo a ser desenvolvido quanto também para fornecer ao desenvolvedor estilos de implementação diferentes da programação clássica. Alguns exemplos desses estilos são:

- Clicar e soltar, neste o desenvolvedor pode compor o jogo posicionando os objetos dentro do cenário e em seguida configurar cada um deles em grupo ou individualmente. Tal configuração pode ser feita através de uma tela de inputs, ou através dos outros exemplos a seguir.
- Blocos lógicos, é feita a configuração das regras e objetos do jogo através de uma organização sequencial de blocos lógicos substituindo a programação clássica.
- Scripts, permite que códigos sejam anexados diretamente a objetos e outros elementos dos jogos, dando a flexibilidade da programação clássica acrescida de uma forma de organização visual.

2.2. Tempo de desenvolvimento e equipes

Como já descrito, as equipes de desenvolvimento de jogos costumam ser formadas por profissionais de diferentes domínios. Todavia, esta não é uma obrigatoriedade,

e o mesmo válido para o número de participantes, podendo variar de um único membro até centenas de profissionais.

Importante notar que quanto maior e melhor qualificada a equipe, maior e mais complexo poderá ser o escopo do jogo produzido em um definido período de tempo. Outro ponto importante é a forma como são produzidos os jogos por pequenas equipes, ainda que não exclusivamente reservada a elas.

Em vários projetos muito do que é planejado advém de experiências dos profissionais. Experiência com o contexto ao qual se passará o jogo e experiências com diferentes jogos, seja na posição de desenvolvedor ou de jogador.

No caso de desenvolvedores solos, é comum que não seja produzida alguma das partes do jogo: arte visual, som, programação e parte do sistema. Porém, raros são os casos onde o sistema por inteiro não é criado pelo desenvolvedor, casos onde são utilizados motores de jogos especializados a um único estilo de jogo.

O tempo de produção de um desenvolvedor solo pode variar de acordo com o escopo do jogo. Enquanto um jogo sem história e de mecânica e conteúdo simples como Flappy Bird [GEARS Studio, 2013] pode ser produzido em um dia, algo pequeno mas complexo, com uma trama, ambientação, artes e diferentes objetivos e feedbacks como “Papers, Please” [3909 LLC, 2013] demorou 9 meses para ser desenvolvidos. Ambos produzidos por desenvolvedores experientes em jogos e programação.

2.3. Ambientes de autoria

Ambientes de autoria propiciam a produção de conteúdo em um elevado grau de abstração ao fornecer um ambiente com grau complexidade abaixo do comumente esperado. O que permite a produção de conteúdo especializado sem que o criador possua o conhecimento em programação ou tenha em sua equipe um programador. Essa facilidade, entretanto, vem a um custo, quanto maior o grau de abstração exigido para que se simplifique uma atividade, maior o custo de desenvolvimento do ambiente.

Quando considerado o universo que é o desenvolvimento de jogos, simplificar todos os processos para os diferentes tipos de jogos existentes exigiria um esforço

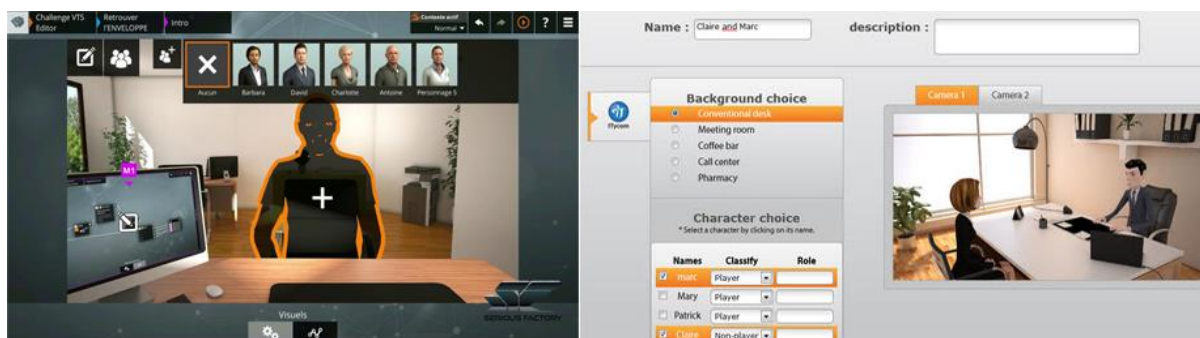
e custo e suas inúmeras possibilidades, a opção por abstrair um único estilo de jogo reduz os custos de implementação. Em paralelo, há a necessidade de se anexar um design instrucional ao ambiente de autoria, amplificando a complexidade da implementação ao produzir novos requisitos e ampliando o escopo do ambiente para que se acomode o conteúdo educacional. A ampliação do escopo neste caso não representa maior liberdade ao autor, ao contrário, a parte de jogos do ambiente agora será delimitada também pela parte educacional, reduzindo ainda mais as possibilidades de desenvolvimento do autor.

2.4. Outros ambientes de autoria

Das pesquisas sobre ambientes de autoria foram encontrados alguns trabalhos relacionados e ambientes comerciais: ITystudio, VTS, Machs, Plugin para jogos investigativo e uAdventure. Destes buscou-se identificar suas características, os pontos positivos, negativos e o que poderia ser aproveitado no projeto do ambiente de autoria proposto neste trabalho.

ITystudio [ITycom] e VTS [Serious Factory], Virtual Training Suite, são ambientes online para produção de jogos sérios com foco na análise comportamental do jogador a partir de diálogos. Dos ambientes relacionados são os únicos de caráter comercial, permitem a criação de um estilo de jogo, uma variação do point & click, onde o jogador através de um ponteiro interage com o jogo selecionando objetos, caixas de diálogo ou qualquer outro elemento presente na tela do jogo. A configuração dos jogos é feita em uma ambiente web de interface similar a um website. Conforme representado na Figura 1, são disponibilizadas configurações de árvores de diálogos, aparência de npc's, imagens de fundo e a história na qual o jogador participará.

Figura 1: Telas dos editores: ITystudio (à dir.) e VTS (à esq.).



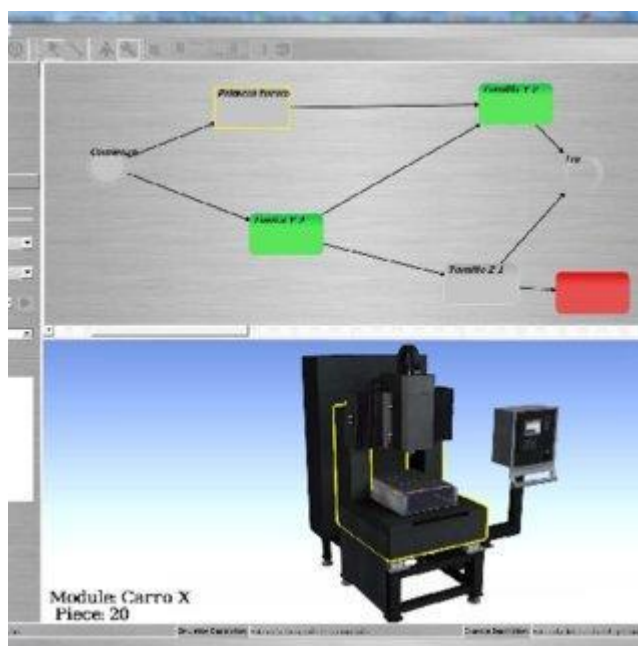
Fonte: ITycom, 2018. Serious Factory, 2018.

Durante a montagem do jogo o desenvolvedor pode pontuar as ações do jogador e marcá-las com características, como empatia, egoísmo, ganância entre outros. E ao fim de uma partida estes valores são coletados e reportados ao responsável pela avaliação comportamental do jogador.

Os jogos montados no ambiente podem ser exportados para diferentes plataformas web, pc e mobile.

O Machs [Mujika, 2011] é um ambiente de autoria para produção de jogos sérios para treinamento no manuseio e reparo de maquinário industrial. Dívida em duas aplicações, a editor de curso onde a partir de um conjunto de modelos previamente disponibilizados o autor pode configurar cursos, configurar defeitos, planejar atividades e definir resultados esperados a partir de um grupo de ações pré-definidas. A segunda aplicação é a jogo, Figura 2, esta recebe o curso produzido no editor em um modelo de arquivo XML e inicia a simulação para o aluno. Os jogos criados são representações 3D de máquinas industriais capazes de fornecer feedback a partir das interações do jogador, simulando o comportamento real das máquinas.

Figura 2: Tela do MATCH.



Fonte: Mujika, 2011.

Plugin para jogos investigativos [Carmosino, 2016] como o título é utilizada para criação de jogos de investigação, apesar do nome, não se resumem a detetives e crimes, mas qualquer temática de caráter investigativo. Um biólogo tentando descobrir os motivos para redução de uma espécie de animal ou um historiador em busca de fatos para um acontecimento histórico são exemplos de jogos a serem criados no ambiente.

O plugin fornece uma interface de desenvolvimento complexa em relação a uma interface de desenvolvimento gráfica, porém, simples se comparada a programação clássica. Através da linguagem xml, o autor define os elementos do jogo, como: linhas de diálogo, cenário e desafios, condições de vitória. O xml em seguida é lido por analisador que identifica os itens montando assim o jogo da Figura 3. A aplicação jogo por sua vez, foi montada utilizando um motor de jogos Unity e os demais recursos como objetos 3D, imagens e sons foram pré-definidos no ambiente, limitando a personalização do jogo, mas reduzindo custos de produção.

Figura 3: Tela do jogo criado com o plugin.

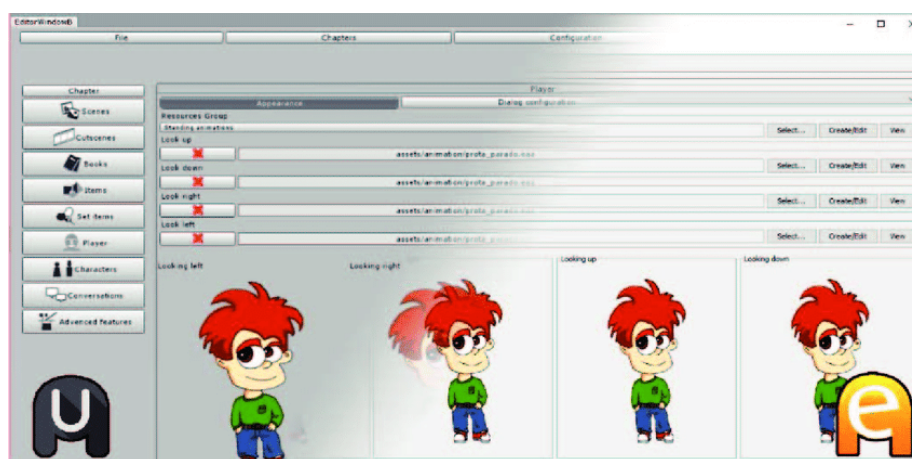


Fonte: Carmosino, 2016.

uAdventure [Perez, 2017], uma versão atualizada do <e-Adventure> [Torrente, 2010], é um ambiente de autoria para jogos investigativos anteriormente desenvolvida em java e nessa versão atualizada para compilar utilizando o motor de jogos Unity. O desenvolvimento do jogo ocorre em uma aplicação editor, Figura 4, onde o autor pode configurar variáveis de rastreo das atividades do jogador a serem coletadas durante a partida, como tomadas de decisões do jogador ou negar a ajuda de um personagem do jogo.

Além de já se tratar de um ambiente maduro se comparado aos demais trabalhos, o principal diferencial do ambiente é a possibilidade de se conectar com Sistemas de Gerenciamento de Aprendizado (LMS). Ao se conectar com um LMS, o jogo desenvolvido formata as informações coletadas e as transfere para uma qualquer ferramenta que realize o acompanhamento do aprendizado.

Figura 4: Tela de edição do eAdventure.



Fonte: Torrente, 2010.

Dos ambientes descritos foi observado que apenas possibilitam a criação de um único estilo de jogo e em geral o estilo investigativo. Foi possível observar, também, algumas características em comum como a disponibilização de recursos audiovisuais pré-definidos, reduzindo o esforço na autoria de jogos. E no caso do Machs, essa adição foi inevitável pela exigência de objetos complexos (máquina). Outra característica foi a adoção de um motor de jogos já existente, reduzindo assim os custos de implementação e manutenção em troca da dependência para com o motor escolhido.

Também foi observado a divisão entre editor e jogo como aplicações separadas. No caso do editor houve uma divergência entre os ambientes comerciais e os trabalhos pela tecnologia optada, enquanto as comerciais optaram por uma aplicação web os trabalhos optaram por um software desktop. Por fim à parte da uAdventure nenhum outro trabalho apresentou a possibilidade de coletar informações da partida e a julgar pelo grau de maturidade dos ambientes pode-se imaginar que é um ponto ao qual um ambiente de autoria para conteúdo educacional a ser atingido em algum momento na evolução do ambiente.

2.5. Análise dos ambientes de autoria

Dos ambientes de autoria selecionados foi possível observar a divisão dos ambientes em editor e jogo, onde em um se é produzido o jogo e no outro

disponibilizado para o jogador. Em alguns casos integrados na mesma ferramenta, se comunicando através de arquivos de configuração ou base de dados compartilhada e em outros casos se conectados por uma rede, nos exemplos a internet, em um modelo que aparenta o de cliente-servidor onde ambas as aplicações (editor e jogo) se conectam a uma terceira que controla a troca de informações.

Dependendo de como for implementada este modelo cliente-servidor pode viabilizar a integração de diferentes aplicações editor e jogo com um baixo custo de manutenção se comparada a solução integrada. No entanto o custo inicial de implantação se torna superior por exigir uma estrutura mais complexa com cabeamento de rede e instalação de servidor ao contrário da integrada com um simples instalador.

Observou-se também dos ambientes a adoção de um motor de jogos já desenvolvido na aplicação jogo, reduzindo significamente o custo e a complexidade de implementação não só da aplicação, mas dos futuros módulos a serem integrados. A escolha adequada de um motor de jogos simplifica o processo ao já fornecer soluções a questões como renderização das artes gráficas, integração de som, cálculos de física e até a compilação multiplataforma, permitindo que o mesmo jogo possa ser exportado tanto para o computador quanto os celulares e até videogames.

Por fim notou-se a importância de poder acompanhar ou ao menos registrar os passos do jogador durante sua partida visto que suas decisões podem permitir que sejam traçados perfis de aprendizagem ou avaliado o conhecimento, seja, apenas por questões avaliativas ou para adaptação do conteúdo.

3. JOGOS EPISTÊMICOS

Um jogo epistêmico, partindo do entendimento geral do termo, pode ser descrito como um jogo para a aquisição ou construção de conhecimento. Duas teorias utilizam deste termo para definir modelos para desenvolvimento de jogos educacionais, quadros epistêmicos [Shaffer, 2004b] e formas epistêmicas [Collins, 1993]. As formas epistêmicas de Collins não serão tratados neste trabalho, mas a critério de compreensão das diferenças estas trabalham com a ideia da construção de conhecimento através de interfaces gráficas ou como o autor define, formas epistêmicas, podendo ser listas, grafos, mapas conceituais ou quaisquer outras representações visuais do conhecimento. Com base nessas formas propõe-se a criação de regras e objetivos para atividades colaborativas ou cooperativas nas quais os participantes interagem na montagem da representação. Atividade da qual se espera que estimule dos participantes a construção de conhecimento, com base no assunto central ao qual será construída a representação.

Quadros epistêmicos é um framework voltado para o desenvolvimento de jogos digitais educacionais, de forma resumida, podem ser descritos como jogos que, através da reprodução de um contexto e suas atividades, visa desenvolver habilidades e conhecimentos passíveis de transferência em um momento posterior para diferentes contextos. Similar às formas epistêmicas a proposta visa a construção do conhecimento pelo participante, entretanto os meios dos quais se utiliza diferem da proposta de Collins.

Para melhor compreender os jogos epistêmicos a partir dos quadros epistêmicos é necessário passar pelas teorias que compõem este modelo, que são: comunidade de práticas, ilhas de especialização, o práxis pedagógico e quadros epistêmicos.

A comunidade de práticas [Lave, 1991], na qual dentro de um contexto colaborativo se inicia como aprendiz e através da reprodução das práticas por membros mais experientes e troca de informações vai-se evoluindo até alcançar o domínio de uma habilidade ou conhecimento. Para ser considerada uma comunidade de práticas devem estar presentes três características: domínio, comunidade e prática. O domínio é a temática na qual ocorre um compartilhamento de interesse pelos membros e por consequência um interesse na resolução de problemas existentes

no domínio. A coletividade no interesse por diferentes pessoas e as interações decorrentes desse interesse formam uma comunidade. As técnicas, processos e atividades aplicados na resolução dos problemas pode ser classificada como práticas.

As ilhas de especialização [Crowley, 2002] são tópicos nos quais ocorre um profundo interesse propiciando uma aprendizagem relevante. Com base no público mais jovem, a teoria propõe a identificação ou geração de tópicos de interesses e deste definir um ponto de partida na construção de uma gama de conhecimento devida a fácil absorção de novas informações.

O pedagogical praxis [Shaffer, 2004a] entende que os conhecimentos e habilidades aprendidos através da prática de atividades autênticas podem ser transferidos ou aproveitados por outras áreas de conhecimento, como a acadêmica. Para tal os seguintes passos foram propostos:

1. Estudo inicial: conduza um estudo piloto no qual os alunos se envolvam em um protótipo das práticas de aprendizagem através das quais os profissionais de um determinado campo são treinados, explorando como os alunos se relacionam com as formas profissionais de pensar e trabalhar.
2. Etnografia da prática: realize estudos etnográficos das práticas de aprendizagem através das quais os profissionais são treinados em um determinado campo.
3. Desenvolvimento de tecnologias: com base na etnografia da prática e no estudo inicial, desenvolver ou adaptar tecnologias que levem o domínio da prática ao alcance de alunos.
4. Ambiente de aprendizagem: crie um ambiente de aprendizagem ou adaptações de tecnologias para ajudar os alunos a aprender com base nas pedagogias descobertas na etnografia da prática.
5. Medidas de resultado e processo: com base na etnografia da prática e no estudo inicial, identifique, desenvolva ou adapte as medidas de resultado que avaliem o progresso no domínio alvo ou domínios de pensamento, bem como procedimentos de observação que documentam como a aprendizagem ocorre no contexto da prática profissional.

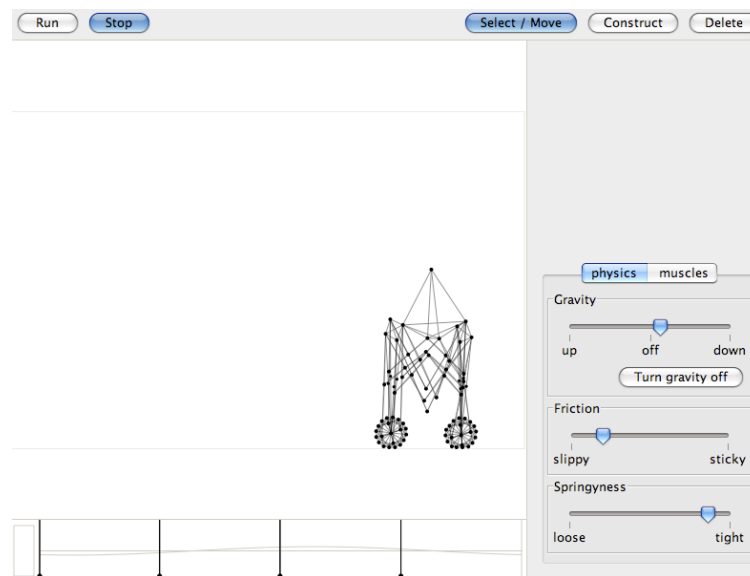
Os jogos epistêmicos gerados através do uso de quadros epistêmicos são voltados ao público mais jovem dado que seu objetivo principal não é a aprendizagem de uma profissão, mas a abstração de conhecimentos acadêmicos adaptados ao contexto profissional e o retorno desse conhecimento alterado para o acadêmico.

3.1. Exemplos de jogos epistêmicos

Para melhor entender a proposta e o que os diferencia de outros jogos similares foram avaliados jogos epistêmicos existentes, no caso, os jogos desenvolvidos pelo grupo GAPS, Games and Professional Simulations: Urban Science, Digital Zoo, Land Science, RescuShell e Nephrotex. Em Urban Science [Shaffer, 2006] o jogador assume o papel de um urbanista na tarefa de replanejar um centro comercial. São fornecidos ao jogador o orçamento disponível, cartas de requisições da comunidade e em seguida é disponibilizado um mapa interativo da região no qual os alunos podem verificar os resultados das diferentes ações possíveis. Ao fim da partida os alunos são convidados a escrever uma proposta de revitalização justificando as razões das decisões tomadas em relação às diferentes requisições. O jogador assume o papel de um urbanista na tarefa de replanejar um centro comercial. São fornecidos ao jogador o orçamento disponível, cartas de requisições da comunidade e em seguida é disponibilizado um mapa interativo da região no qual os alunos podem verificar os resultados das diferentes ações possíveis.

Em Digital Zoo [Svarovsky, 2007] o jogador assume o papel de um engenheiro biomecânico na tarefa de desenvolver criaturas virtuais em um simulador, desenvolvendo e aprendendo conceitos de física e engenharia. Os alunos são apresentados ao SodaConstructor, Figura 5, uma ferramenta na qual podem simular diferentes estruturas compostas por consolas (ou cantilever), estruturas de sustentação cinética comumente vistos em pontes e torres, nela os alunos deverão buscar criar estruturas que se assemelham a animais. A cada ciclo de desenvolvimento da criatura os alunos são convidados a apresentar as mudanças realizadas.

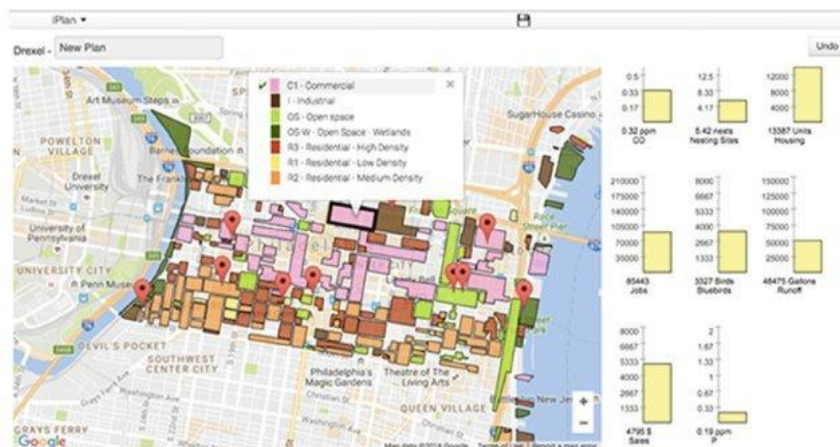
Figura 5: Tela do SodaConstructor.



Fonte: Svarovsky, 2007.

Em Land Science [Beckett, 2005] o jogador assume o papel de um estagiário em uma empresa de urbanismo com a tarefa de zoneamento de uma cidade. Para a tarefa são fornecidas as requisições dos diversos grupos comunitários existentes na região e um mapa interativo, Figura 6, onde pode avaliar os impactos das possíveis decisões. Durante as sessões de jogo os alunos contam com o auxílio de tutores para que tirem dúvidas. No fim os alunos são convidados a apresentarem a proposta final argumentando possíveis decisões conflitantes.

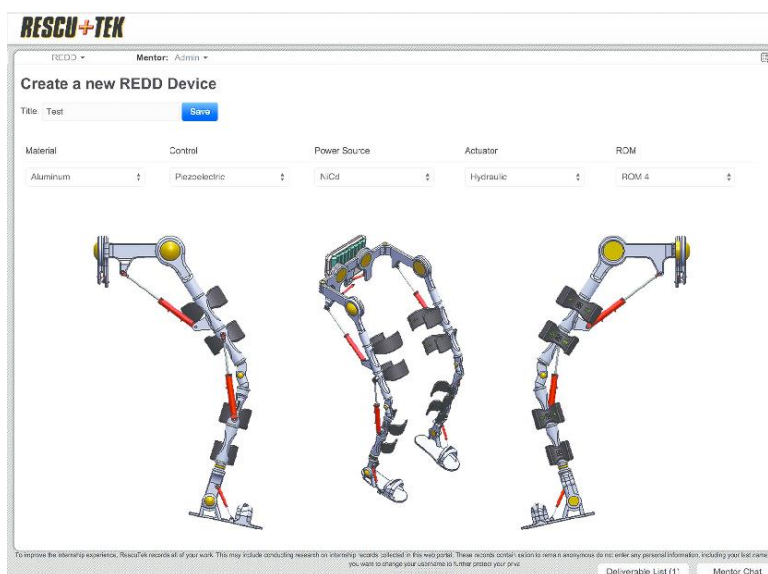
Figura 6: Tela do Land Science.



Fonte: Beckett, 2005.

Em RescuShell [Arastoopour, 2015] o jogador assume o papel de um estagiário em uma empresa de robótica e é dada a tarefa de desenvolver uma perna exoesqueleto. São fornecidos relatórios técnicos baseados em informações e experimentos reais, em seguida os jogadores recebem acesso a um simulador, Figura 7, onde recebem acesso a diferentes modelos de exoesqueletos os quais alterações resultam em uma série de informações de desempenho. Durante as sessões os alunos recebem diferentes feedback de setores da empresa apresentando interesses conflitantes em relação ao produto final. Ao fim os alunos apresentam o modelo final o justificando em relação aos interesses dos diferentes setores.

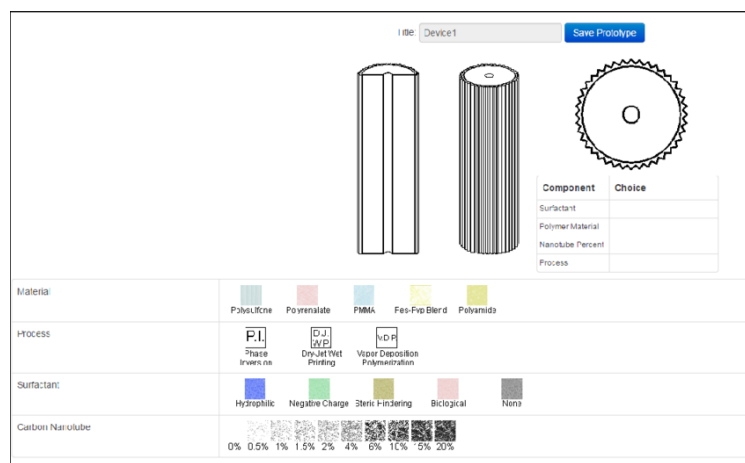
Figura 7: Tela da aplicação especialista usada no RescuShell.



Fonte: Arastoopour, 2015.

E em Nephrotex [D'Angelo, 2011] o jogador assume o papel de um estagiário em engenharia biomédica com a tarefa de inventar uma membrana do sistema de diálise renal com uso de nanotecnologia. São fornecidos relatórios técnicos baseados em informações e pesquisas reais, em seguida os jogadores recebem acesso a um simulador, Figura 8, para testar os modelos e obterem feedback. A cada tentativa os alunos são convidados a refletirem as hipóteses geradas e os resultados alcançados com o grupo e seu tutor.

Figura 8: Tela da aplicação especialista usada no Nephrotex.



Fonte: D'Angelo, 2011.

Dos jogos observados foi possível observar um padrão seguido em todos: o jogador como um aprendiz em um ambiente profissional munido de informações básicas e uma ferramenta de feedback com o objetivo de propor uma solução para um dado problema. Em alguns casos os jogadores são convidados a trabalhar em equipe e durante toda a partida é disponibilizado um tutor para atender a possíveis questionamentos

3.2. Análise dos trabalhos correlatos

De início pode ser observado dos jogos epistêmicos relacionados a repetição do seguinte roteiro: o jogador assume o papel de um estagiário ou aprendiz em uma empresa onde recebe informações relacionadas a uma atividade específica e a missão de desenvolver uma proposta de solução para um problema relacionado às informações fornecidas. Além das informações, para que o jogador possa realizar inferências, é disposto uma ferramenta de feedback na qual é possível testar suas afirmações e suposições.

Tal modelo de jogo é referenciado como estágio virtual. Um dos pontos que os diferenciam de outros jogos sérios, como os criados nos ambientes de autoria relacionados (exceção do Machs), é a presença de uma ferramenta de feedback. Um ambiente de autoria para jogos epistêmicos deve, portanto, ser capaz de

incorporar ou facilitar a adição de ferramentas especializadas capazes de fornecer um feedback adequado à aprendizagem objetivada.

Além do feedback técnico é também uma constante em um jogo epistêmico a presença de um instrutor responsável por responder às possíveis dúvidas dos participantes em relação ao contexto geral do jogo ou a pontos quais a máquina de feedback falha em suprir.

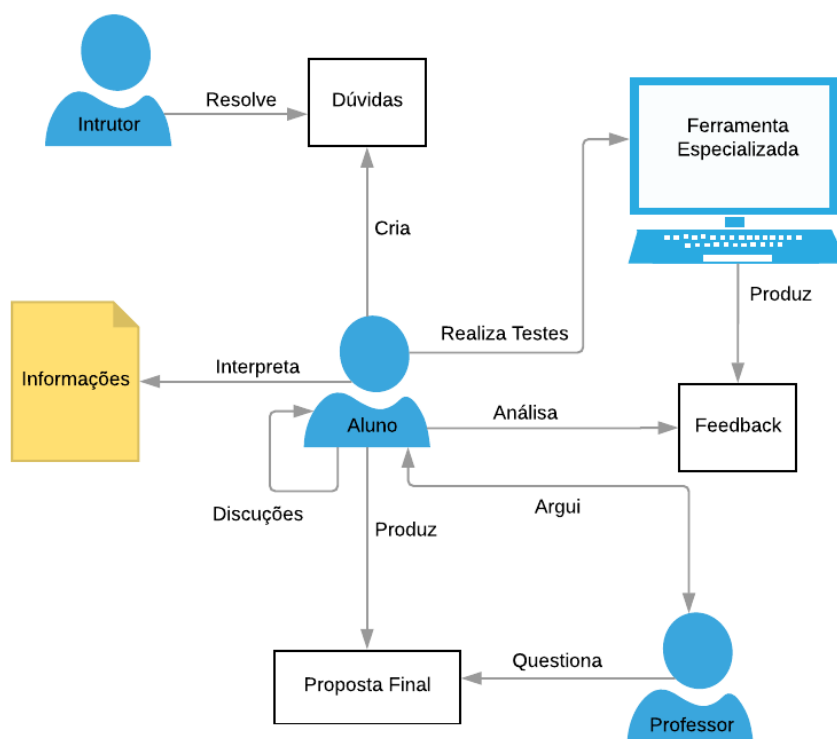
Por último há o professor cuja tarefa é a de questionar o resultado final produzido pelo jogador, a solução para o problema inicial, estimulando a capacidade argumentativa do jogador aluno. Esses elementos devem ser considerados e trabalhados quando planejado produzir algo que não fuja ao modelo de um jogo epistêmico, mas que não descaracterize demais dos jogos digitais como conhecidos. O meio termo, portanto, será essencial quando planejado o ambiente e os possíveis jogos a serem criados no mesmo.

4. MODELO PARA DESIGN INSTRUCCIONAL DE JOGOS EPISTÊMICOS

De início buscou-se um modelo de desenvolvimento a ser seguido durante todo o processo de criação do ambiente, não apenas para os jogos a serem criados, mas para entender quais e qual o comportamento das funcionalidades do ambiente a ser projetado. Foram então observadas as atividades dos jogos epistêmicos e seus ciclos. Das observações, foram notadas características em comum a todos os jogos pesquisados: cenário, aplicação, composição de uma situação, tarefas a se realizar em prol de um objetivo final e apresentação de uma solução à uma problemática proposta. Em todos os jogos descritos, também foi possível constatar a presença de professores ou tutores para auxiliar no caso de dificuldades, devido ao grau de complexidade da tarefa apresentada.

Identificou-se nos jogos epistêmicos os seguintes elementos: documentos apresentando informações essenciais para entendimento do problema; ferramenta de feedback ou o ambiente virtual de simulação; suporte ao qual os jogadores poderão recorrer durante a partida; um produto resultado da conclusão, podendo ser um relatório ou uma apresentação. Para melhor visualizar o fluxo das atividades, os elementos e suas aplicações, foi elaborado o diagrama da Figura 9, que representa os ciclos ocorridos durante uma partida em um jogo epistêmico.

Figura 9: Diagrama do ciclo de um jogo epistêmico.



Fonte: O autor.

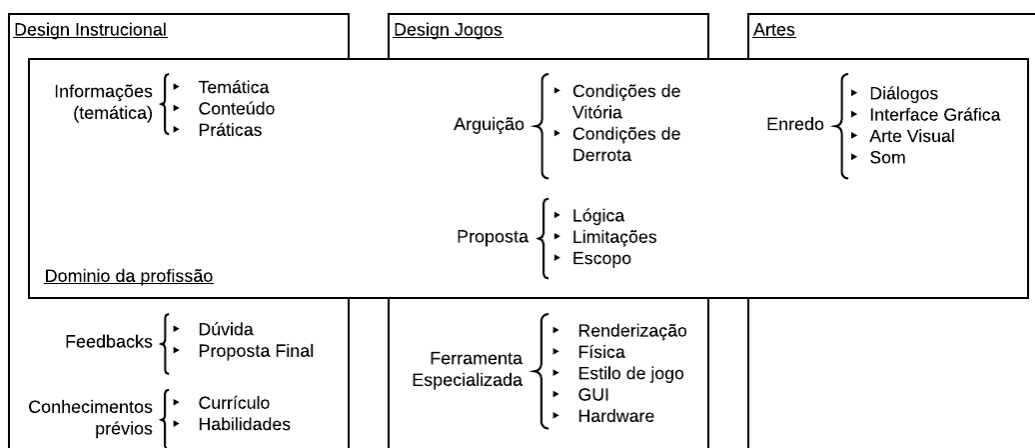
Em seguida, dos elementos e atividade definidos, buscou-se por aqueles relacionados também aos jogos em um contexto geral. Das análises, foi acrescentado aos objetivos, a busca por alternativas que permitisse acrescentar o maior número disponível de elementos educacionais (jogos epistêmicos) à proposta de um jogo para o ambiente de autoria a ser desenvolvido, de modo que o jogador necessite durante uma partida do mínimo de intervenções externas possível. Para melhor visualização, os elementos foram distribuídos em: design instrucional, todos elementos relacionados à aprendizagem e não ligado as mecânicas; design de jogos para tudo que influencia as regras e mecânicas do jogo; e as artes como os demais elementos.

Dos grupos, foi observado no design instrucional uma dependência quanto a temática e conteúdo. Nos jogos epistêmicos, a temática representa o ambiente, e o conteúdo, o domínio da profissão a ser retratada apresentando variações de acordo com as competências a serem praticadas. Pela mesma razão, no design de jogos podem ocorrer alterações no diálogo, advindas da ontologia da comunidade correspondente, transformação do cenário para uma ambientação adequada,

definição de regras que adaptem suas práticas e ferramentas, dentre outras mudanças derivadas da profissão a ser representada.

Esse conjunto de características que formam a representação de uma profissão foi separado como um terceiro grupo denotado como domínio da profissão que representa as áreas de conhecimento externas das quais serão feitos recortes que servirão de base para os jogos criados. Desse novo grupo foi elaborado o diagrama da Figura 10, representando os conjuntos e sobrepondo os domínios da profissão sobre os elementos agrupados.

Figura 10: Elementos dos jogos epistêmicos agrupados em conjuntos.



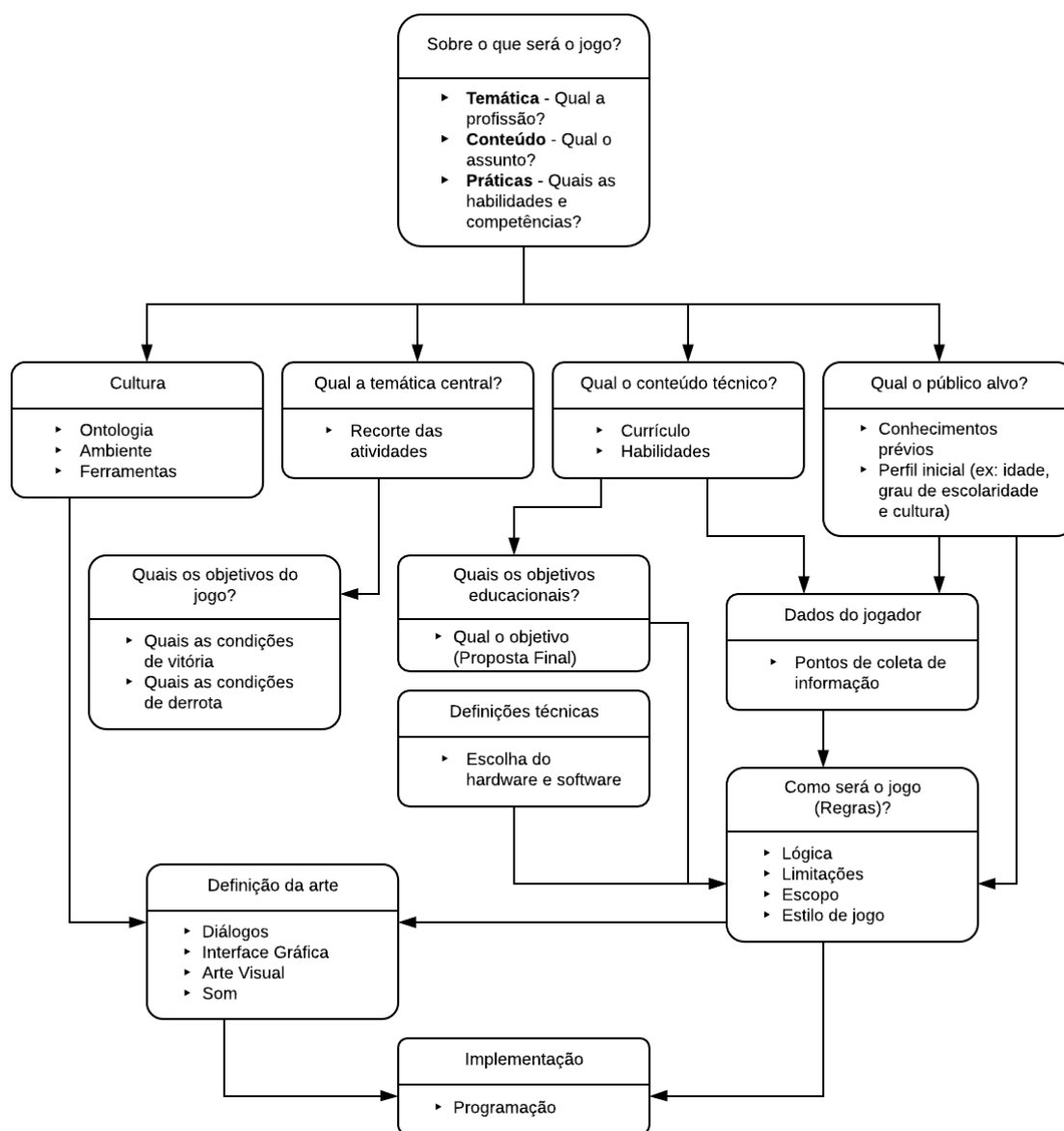
Fonte: O autor.

Em uma descrição mais detalhada, os elementos que compõem o processo de produção de um jogo foram categorizados em atividades relacionadas ou não à programação, uma vez que a programação está ligada às mecânicas. No caso das práticas exclusivamente criativas como produção da arte gráfica e sonora, confecção de personagens, enredo, diálogos e conteúdo, o produto resultado é utilizado pelas demais sem grandes influências. Outras escolhas como a definição do estilo de jogo, gráficos visuais, jogabilidade, pontuação, inventário, conquistas e condicionais (vitória e derrota, mudança de cenário, aquisição ou perda de itens e conquistas) não estão diretamente relacionadas a programação, mas afetam diretamente as regras do jogo a ser produzido.

As regras, junto da definição do hardware, determinam as mecânicas do jogo. No caso específico dos jogos educacionais, devem ser respeitados os objetivos

educacionais que guiarão o jogador na utilização do jogo da forma adequada alcançando a aprendizagem desejada e, acima das regras, os domínios da temática e conteúdo delimitadores do escopo do jogo. Dos elementos destacados foi definido um modelo, Figura 11, a ser seguido tanto no desenvolvimento do ambiente quanto no desenvolvimento de jogos no ambiente, sendo a segunda implícita ao aperfeiçoamento do ambiente, no que tange em como o ambiente deve se portar para que o autor siga o modelo proposto intuitivamente.

Figura 11: Fluxo para desenvolvimento de um jogo epistêmico.



Fonte: O autor.

Inicia-se pela definição do quadro epistêmico, o subdividindo em elementos culturais que formam a profissão representada no quadro, as atividades envolvidas, o que se espera ser aprendido ou desenvolvido e quem serão os aprendizes. A partir desses itens são definidos os demais elementos que serão utilizados no desenvolvimento do jogo.

Dos elementos culturais é possível identificar o dialeto, a ontologia e demais nuances que compõem as interações sociais, a ambientação, os elementos visuais e a sonoridade propiciada pelo ambiente. Esses elementos refinados ao ponto que seria adequado integrá-los no jogo pode ser tratado como arte.

Um recorte das atividades pode vir a representar uma rotina à qual um profissional percorre para a produção de algum resultado final esperado. Tal rotina, objetivos, expectativas e possíveis percalços encontrados se tratados adequadamente, delimitando início e fim, pontos de retorno, obstáculos e condições de sucesso e fracasso, podem dar origem a fluxos dentro de um jogo.

As expectativas de aprendizagem refletem em parte o conteúdo que se pretende ensinar, saber que tipo de competência se pretende desenvolver influencia diretamente nas mecânicas e regras do jogo. As mecânicas por vez se limitam também pela tecnologia a ser adotada.

Conhecer o público alvo é fundamental para que se definam as regras do jogo, tanto em sua complexidade quanto no quão compassivo será aos erros do jogador. Também permite identificar em que pontos do jogo seria adequado realizar a coleta de informações, vez que em pontos onde se é de esperar que o jogador não encontre dificuldades a coleta de informações pode se tornar irrelevante.

Definido como será o jogo, é possível identificar a arte adequada, e o mínimo necessário, dentro do escopo pensado para que se tenha um jogo adequado visto que na ausência de estímulos externos a arte é o único elemento responsável por atrair a atenção do jogador nesta interação homem máquina.

5. CONCEPTUALIZAÇÃO DE UM JOGO EPISTÊMICO

Seguindo o modelo baseado na teoria dos jogos epistêmicos foi conceptualizado um jogo, ocorrida antes da produção do ambiente para facilitar na definição de módulos a serem implementados que permitiriam a criação do jogo no ambiente. A princípio foi definida a temática e as competências a serem trabalhadas. Adotou-se das diretrizes do MEC [Brasil, 2018] para ensino da matemática, dos objetivos gerais as competências a serem trabalhadas:

- Saber identificar padrões matemáticos e adequá-los a problemas diversos.
- Relacionar elementos quantitativos e qualitativos.
- Interpretar e deduzir situações-problema.
- Ser capaz de transferir os conhecimentos matemáticos a diferentes temáticas.

Foi optado como temático do jogo as atividades do varejo, em específico a de pequenas lojas ou mercados. Compra de mercadorias - como roupas, alimentos e brinquedos - seja por visita a lojas físicas ou online são atividades à quais podem ser familiares a alunos dos anos iniciais do ensino fundamental. E onde é possível observar atividades de profissões ligadas ao varejo mesmo na ausência do mesmo como é o caso das lojas online.

Da temática foi feito um levantamento das atividades, personagens, objetos e ambientes básicos que compõem uma loja, resultando no diagrama da Figura 12.

- Atividades: exposição e organização de produtos, venda, compra, controle de estoque, controle de caixa, limpeza, propaganda.
- Personagens: gerente, vendedor, caixa (função), expositor (função), cliente e fornecedor.
- Objetos: produtos, carrinho/cesta de compras, lista de compras, carteira, dinheiro, expositor (objeto), itens de limpeza, caixa (objeto).
- Ambientes: loja, escritório, estoque.

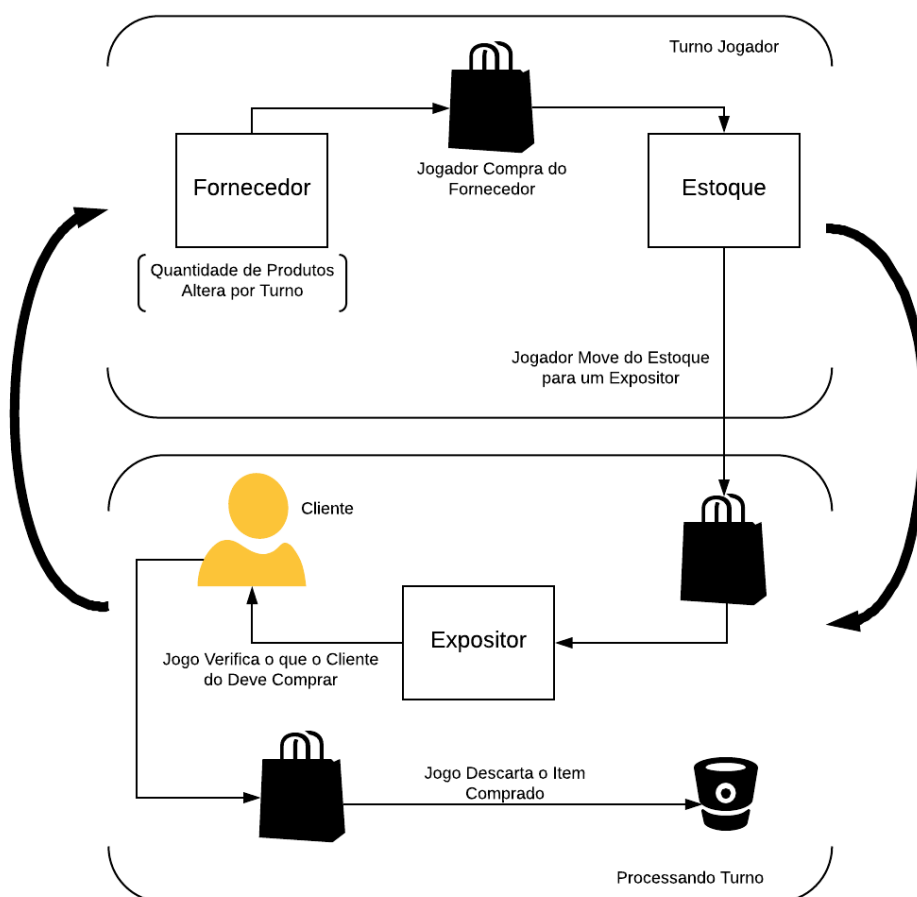
Do levantamento buscou-se um ambiente e personagem que permitisse o maior número possível de relacionamentos com as atividades que promovessem as competências selecionadas. Foi decidido usar o ambiente de uma loja por representar melhor a temática proposta. Para o personagem, no entanto, não havia

um capaz de atingir uma quantidade satisfatória de atividades. O gerente cobre as atividades superficialmente dado que a participação dele nas atividades ocorre no plano do planejamento, em um jogo, no entanto, não se faz necessário que se siga à risca o funcionamento do mundo real.

Foi definido como personagem o gerente e omitido os demais personagens que representam profissões gerando feedback imediato às ações de planejamento. Resultando na adaptação do ambiente como cenário e das interfaces gráficas para melhor representar as atividades e feedbacks.

As atividades seriam melhor representadas por uma visão ampla do mercado seguida de informações similar ao que poderia ser encontrado em sistemas de gestão. Entretanto, para que se tornem funcionalidades no ambiente as atividades precisam ser implementadas via módulos, objeto ou interface. Com base no conceito de personagem ao qual o jogador será representado e o ambiente a ser reproduzido no módulo cenário, foi gerado um modelo das atividades, seus relacionamentos, personagens, objetos e ambientes.

Figura 12: Diagrama das atividades do jogo.



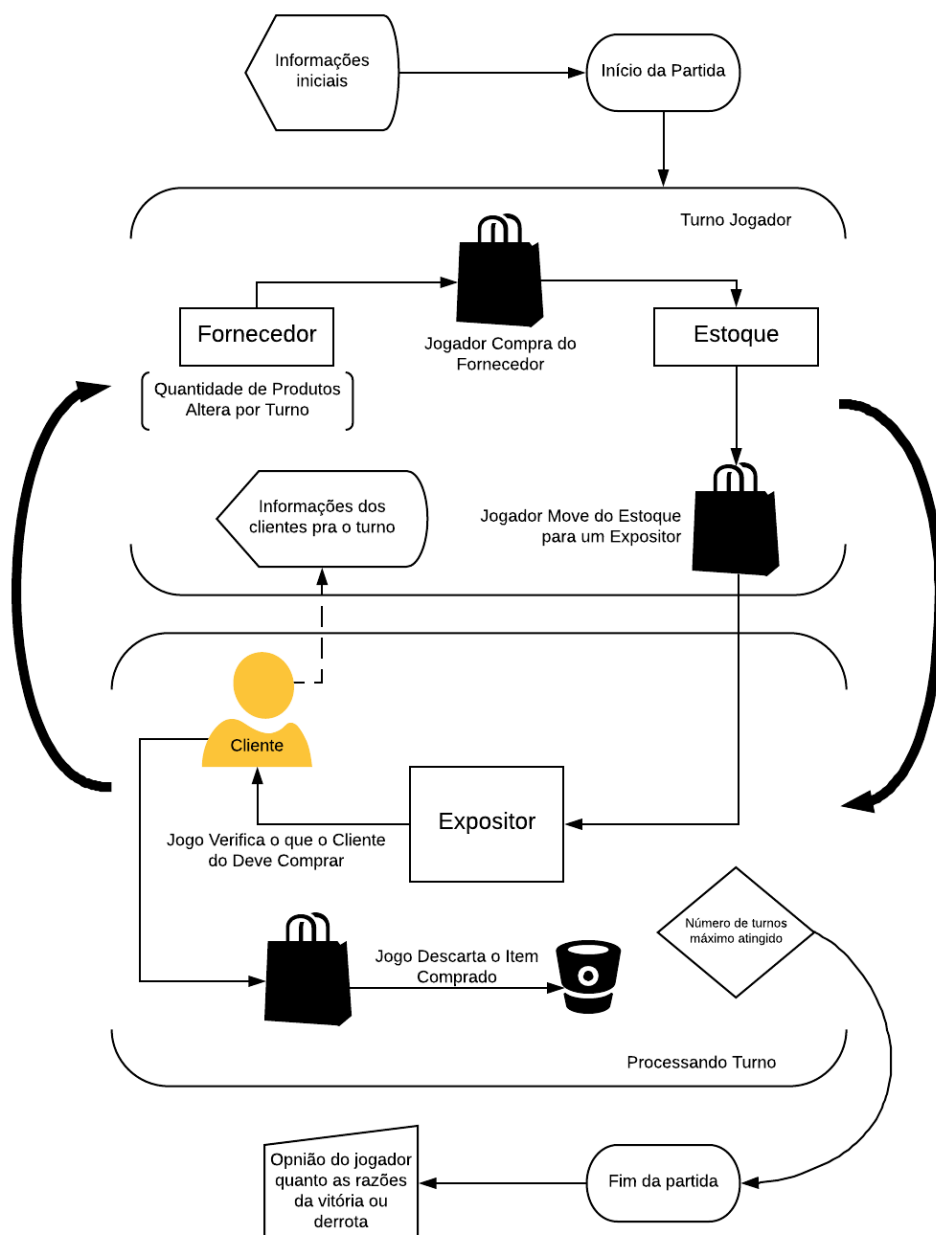
Fonte: O autor.

As atividades e seus elementos foram tirados de seu contexto e buscou-se qual dos objetos envolvidos na atividade poderia vir a representá-la, quais os outros objetos fundamentais para a atividade e destes os que poderiam se tornar elementos deste objeto. As atividades restantes foram analisadas para identificar a possibilidade de estas serem representadas por um módulo interface e o restante foi descartado do escopo do jogo.

Como resultado foram concentradas as atividades no ciclo do produto: do fornecedor ao estoque, do estoque ao expositor e do expositor ao cliente. E os seguintes módulos: expositor, módulo objeto; fornecedor, módulo interface; cliente, módulo interface. Os produtos ficaram definidos como itens e o estoque como inventário do jogador, ambos suprido pela base do ambiente, portanto, os módulos deverão interagir com os itens e inventário.

Por último o modelo das atividades foi adequado ao modelo de jogo epistêmico resultado em um modelo final da, Figura 13, de fluxo do jogo a ser produzido.

Figura 13: Diagrama final das atividades do jogo mesclado com jogos epistêmicos.

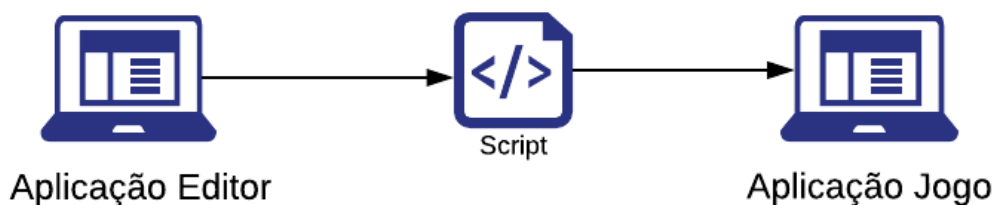


Fonte: O autor.

6. PROPOSTA DE UM AMBIENTE DE AUTORIA

Do modelo de desenvolvimento foi elaborada uma proposta de ambiente de autoria para jogos, aplicando o modelo elaborado a partir da interpretação dos processos de um jogo epistêmico como design instrucional e aproveitando de características dos ambientes de autoria observadas. Uma das características mais presente dentre os ambientes é a divisão em duas aplicações: editor, onde os jogos são criados; cliente, onde os jogos são executados. Nessa organização, representada na Figura 14, o autor produz uma configuração de jogo que é enviada do editor para a cliente que a interpreta e instância os objetos formando o jogo.

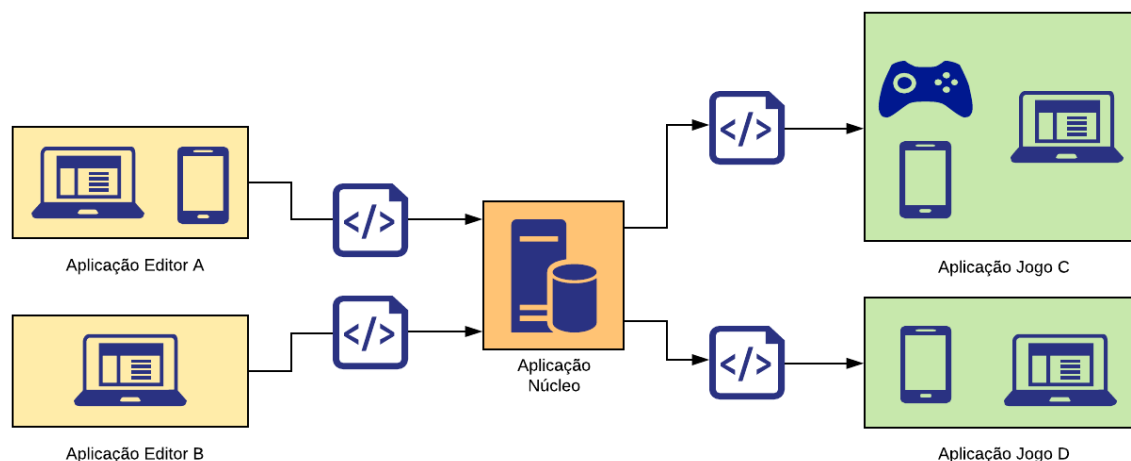
Figura 14: Organização editor/jogo.



Fonte: O autor.

Para melhor adequar as tecnologias atuais sem deixar de pensar no futuro foi decidido pela divisão da ferramenta, conforme Figura 15, em três: editor, cliente e núcleo. Seguindo um modelo cliente-servidor, a aplicação núcleo atuaria como uma camada de serviço, ficando responsável por organizar os jogos criados no editor e sua disponibilização para a aplicação jogo. Além da nova divisão foi acrescentado a adoção de um padrão de arquivo para a configuração dos jogos, dessa forma mesmo que as tecnologias se tornem obsoletas basta que a nova aplicação seja capaz de interpretar os jogos criados.

Figura 15: Organização editor, núcleo e cliente.



Fonte: O autor.

A adição da aplicação núcleo como um intermediário torna as aplicações editor e cliente independentes, possibilitando a existência de diversos editores e cliente compartilhando a mesma aplicação núcleo e tudo que está aplicação venha a oferecer.

6.1. Aplicação Núcleo

Responsável por fazer o intermédio das comunicações dentre as aplicações, internas ou externas, centraliza as vias de comunicação entre as aplicações, controlando e validando a comunicação e a formatação do que trafega no ambiente. As comunicações ocorrem por três rotas controladas pelo núcleo: para com o editor, o jogo e aplicações externas. Cada uma com objetivos e funções diferentes. Esta aplicação só pode ser acessada diretamente por um único tipo de usuário classificado de administrador do sistema, os demais só farão acesso através de outras aplicações.

As comunicações com o editor são realizadas para as trocas de informações relevante a configuração de módulos - armazenamento e validação dos modelos de configuração - com base em modelos pré-definidos pelos desenvolvedores dos módulos e enviados pelo administrador para registro no núcleo e futuras validações das configurações enviadas pelo editor. Das comunicações com o cliente, o editor

disponibiliza as informações adquiridas com o editor e registra informações de partida advindas da aplicação cliente.

A aplicação núcleo, além da circulação das informações, realiza a gerência dos usuários do sistema: seus acessos, histórico de uso, permissões e privacidade. E a gerência dos jogos criados: nome e descrição de jogo, criador, versão, datas de criação e acesso, resultados de partida e demais informações relativas ao jogo. Ficam a critério das aplicações cliente como serão representados os jogos criados para o jogador, se fará proveito da base de usuários e autores ou se haverá ou não coleta de informações. Observado que ainda que seja opção da aplicação jogo, estas devem ainda obedecer aos níveis de permissão e privacidade configurados pelo autor no editor.

6.2. Aplicação cliente

Com base na organização do ambiente, foi pensado um modelo de aplicação cliente que siga os mesmos preceitos de reusabilidade e durabilidade, considerando também que as principais funções desta aplicação são: interpretar as configurações de jogos e coletar dados da partida. Foi elaborado uma organização de módulos e base. Sendo a base composta por funcionalidades não relacionadas diretamente às do jogo:

- Organização dos jogos criados para o ambiente.
- Informações de alunos registrados.
- Armazenamento dos dados coletados durante as partidas e sua leitura.
- Administração de acessos e permissões para os diferentes tipos de usuários: autor, jogador e administrador.

E de funcionalidades comuns aos estilos de jogos escolhidos para compor o escopo do ambiente como:

- Gerência de módulos com formas de identificação do que estão ativos no jogo e o acesso às suas funcionalidades.
- Gerência de fases ou cenários, para o caso de jogos com mais de um módulo cenário cadastrado.

- Gerência de recursos, o registro de todos os recursos, como: itens, tempo, turnos, dinheiro. Como rastrear eles durante uma partida e suas interações.
- Gerência de condições ou regras, eventos a serem iniciados a partir do estado de outros elementos como os de recurso.

Tais características são variáveis que possam vir a influenciar o jogo, um exemplo seria a limitação do número de tentativas, comum a diversos jogos e popularmente descrito como vidas, e caso venha a ser necessário ao ambiente, este deve ser incluído a base por promover uma alteração significativa as mecânicas. Eventos de fracasso devem ser disponibilizado às demais funcionalidades do jogo, em caso de fracasso ou atingido o limite do número de fracassos outros eventos devem ser ativados.

As funcionalidades base podem ser implementadas como uma tela única ou representações na interface do jogo. Todas funcionalidades e elementos implementados na base devem ser incluídas no arquivo modelo de configuração. Considerando que a ausência deste em uma outra aplicação cliente pode inviabilizar a execução de um jogo que faça uso dessa funcionalidade.

Os módulos são implementações de um conjunto de funcionalidades cujo uso na montagem de um jogo é opcional, devem, no entanto, estar ligadas as funcionalidades base. No escopo do ambiente estes foram divididos em módulos do tipo cenário, objeto e interface de acordo com estilo de representação que melhor se adeque a funcionalidade. Módulos do tipo cenário são essencialmente representações de ambiente onde ocorrerá o jogo em um exato momento. Cada jogo produzido pode conter vários módulos de tipo cenário, gerenciados pela base, porém, cada cenário deve conter somente um módulo deste tipo. O conteúdo do módulo fica a critério da temática e objetivo para o qual este fora tal planejado. Entretanto, algumas características devem estar presentes:

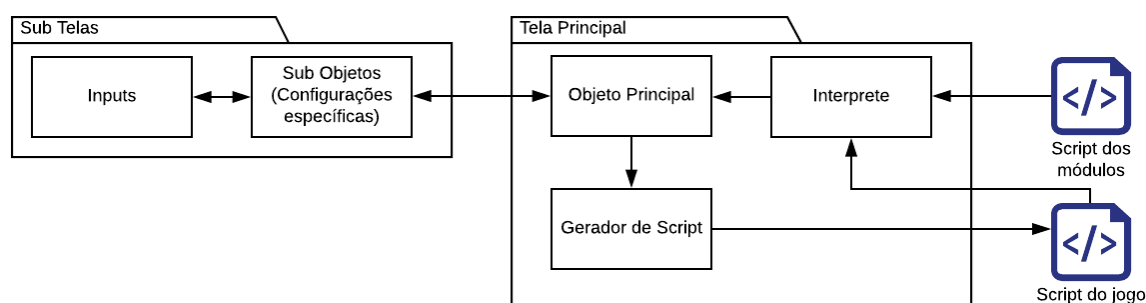
- Permitir o jogador a alcançar outros cenários a partir deste.
- Permitir ao jogador interação com os demais objetos que compõem o cenário.
- Suportar outros módulos objeto de categoria similar.
- Ser redimensionável, capaz de suportar qualquer resolução de tela suportado pela base.

- Deve possuir pontos de coleta de informações da partida.

Os módulos do tipo objeto são representações de objetos que podem vir a ocupar espaços nos módulos do tipo cenários e devem, além de possuir as mesmas características do módulo cenário, serem capazes de trocar informações entre si e com o cenário. Tais trocas devem ser reguladas pela base, através dos gerenciadores de inventário, conquistas ou de forma genérica, vez que para que seja mantida a modularidade não pode haver comunicação direta entre módulos. Os módulos do tipo interface só diferem do tipo objeto quanto a representação, estes ficam expostos junto aos outros componentes de menu do jogo.

Tanto a base quanto todos os tipos de módulos devem ser produzidos em conjunto com um arquivo template de configuração que informa a aplicação núcleo informações que permitam o reconhecimento do módulo. Em geral é esperado que os módulos sejam projetados em conjunto, mas que, entretanto, sejam projetados para um possível reaproveitamento que permita a produção de jogos além dos inicialmente imaginados. Desta descrição foi concluída um diagrama de fluxo, Figura 16, da montagem de jogos no editor com base na organização dos módulos implementados na aplicação cliente (jogo).

Figura 16: Montagem de um jogo no editor.



Fonte: O autor.

As comunicações com o editor são realizadas para as trocas de informações relevante a configuração de módulos - armazenamento e validação dos modelos de configuração - com base em modelos pré-definidos pelos desenvolvedores dos módulos e enviados pelo administrador para registro no núcleo e futuras validações das configurações enviadas pelo editor. Das comunicações com o cliente, o editor

disponibiliza as informações adquiridas com o editor e registra informações de partida advindas da aplicação cliente.

6.2.1. Arquivos de configuração

Como já descrito, na montagem do jogo há uma comunicação entre as aplicações realizada pelo envio de arquivos de configuração e que estes se distinguem em configuração de um jogo e informações de aplicações. Seguindo o fluxo de montagem do editor é possível visualizar que as informações de configuração de input podem ser requisitadas conforme é feita a opção pelo uso de específicos módulos na montagem do jogo.

Para que se mantenha a compatibilidade entre os módulos e para com a aplicação jogo, os arquivos de configuração devem possuir neles informações que proporcionem uma validação adequada no ato de interpretação e montagem do jogo. Informações relevantes podem incluir, para ambos: nome, versão, descrição. E para os módulos qual as bases suportadas, permitindo que o autor seja informado com antecedência quais aplicações existentes seriam capazes de executar o jogo criado através da combinação de tais módulos. Cada módulo pode, além do arquivo de informações, fornecer também um arquivo de configuração do módulo com os tipos dos inputs necessários para sua configuração.

6.3. Aplicação Editor

Responsável por fornecer ao autor uma interface simplificada para produção de jogos, adequada aos conhecimentos previstos dos usuários do ambiente. Em geral nos ambientes observados optou-se por modelos de interface comum à um maior número de usuários. Como um dos pontos principais do ambiente será a inclusão de novas aplicações cliente e de seus módulos a qualquer momento, uma aplicação editor ideal deve ser capaz de se adaptar à diferentes configurações.

Propõe-se que esta aplicação seja estruturada por um conjunto de inputs de interface fixos equivalentes as configurações base comuns a todos jogos a serem criados acrescidos por agrupamentos de inputs de interface de acordo com as

possíveis configurações dos módulos que compõem a aplicação jogo. Ou seja, para cada módulo criado para uma aplicação cliente deve existir um conjunto de inputs de configuração em um editor para que se torne possível a criação dos jogos.

Para tal a aplicação editor deve fornecer a descrição dos possíveis inputs à aplicação núcleo que então irá selecionar quais módulos e bases podem ser configurados pelo editor. Diferentes aplicações editores podem interagir com a mesma aplicação núcleo mas configurarem diferentes módulos. Para que sejam evitados erros na comunicação entre as aplicações, casos de inputs similares ou que realizam funções diferentes da esperada, cada descrição dos inputs esperados por um módulo deve ser validada com a listagem existente na aplicação núcleo.

7. PROTOTIPAÇÃO DO AMBIENTE

A fim de averiguar a aplicabilidade e melhor descrever o modelo proposto foi elaborado um protótipo do ambiente e um conjunto de módulos para um suposto jogo a ser criado no ambiente. Nesta seção serão descritas a modelagem do protótipo, tecnologias escolhidas para o desenvolvimento e complicações encontradas. O capítulo foi organizado em: descrição do protótipo da aplicação núcleo; descrição do protótipo da aplicação editor; descrição do protótipo da aplicação cliente, o comportamento dos módulos e coleta de dados. E cada tópico em: descrição da aplicação, requisitos funcionais e modelagem de dados.

7.1. Protótipo da aplicação núcleo

Como se trata de um intermediário das comunicações dentre as aplicações, características de um serviço, optou-se por uma estrutura cliente-servidor, tendo a aplicação núcleo como responsável pelo fornecimento de serviços às demais aplicações.

Para tal foram empregados o uso das tecnologias Python e Django, como também há a necessidade de se armazenar informações foi utilizado um banco de dados relacional SQL, o SQLite por ser menos complexo, versátil e principalmente pelo fato de que questões como eficiência e performance não serem de grande relevância nessa etapa.

Como já descrito apenas a aplicação núcleo fará armazenamento de informação, sendo a única que obrigatoriamente deve possuir uma base de dados, caso do protótipo descrito. Também são funções da aplicação o controle de usuários, arquivos e jogos criados.

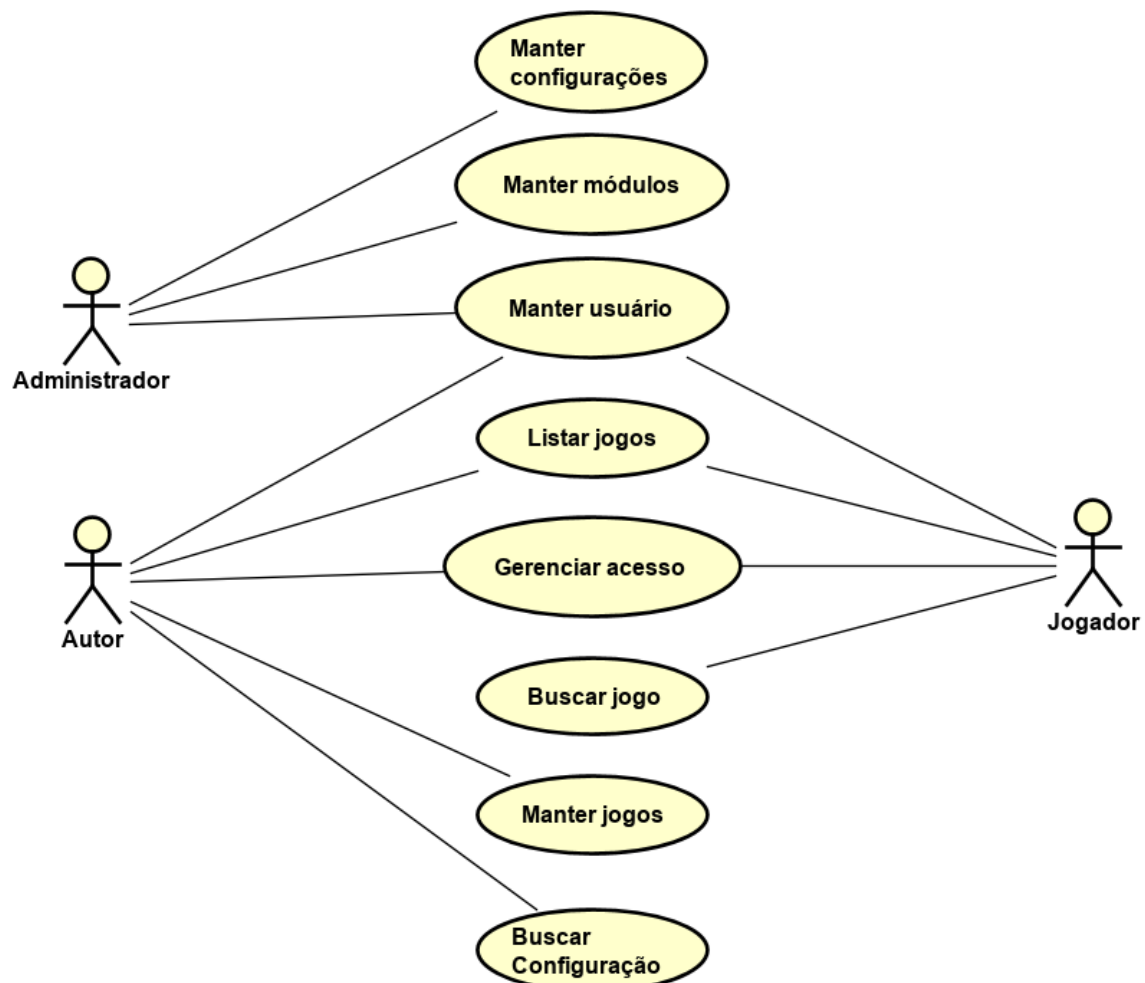
7.1.1. Requisitos funcionais

São funcionalidades da aplicação núcleo conforme Figura 17:

- Cadastrar e buscar usuários dos tipos cliente e editor, alterar senha e gerar, renovar e validar tokens de acesso.

- Cadastrar, remover, buscar e listar as salas (jogos) criados por usuários do tipo editor, as buscas podem ser feitas pelo código ou nome da sala e podem retornar as informações da sala ou o arquivo de configuração. Cadastro e remoção só podem ser realizados pelo usuário editor.
- Cadastrar, remover, buscar e listar módulos criados por desenvolvedores, as buscas podem ser feitas pelo código ou nome do módulo. Cadastro e remoção só podem ser realizados pelo usuário administrador.
- Cadastrar, remover e listar arquivos de configuração da aplicação pelo usuário administrador.

Figura 17: Caso de uso Aplicação Núcleo.

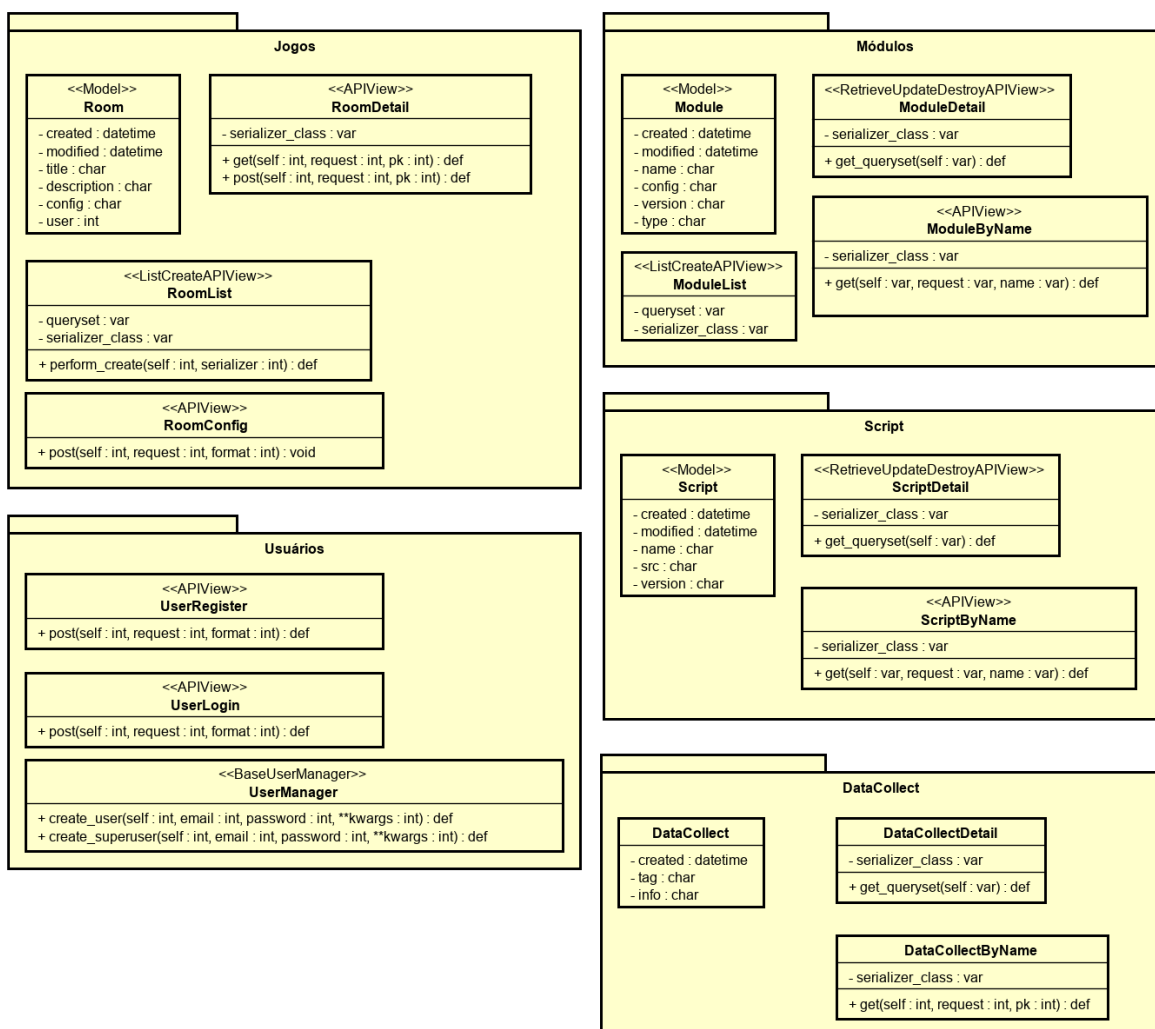


Fonte: O autor.

7.1.2. Modelo de classe

Foi utilizada o padrão Unified Modeling Language, UML, na representação das classes da aplicação, Figura 18, adaptando a estrutura do Django/Python e omitindo objetos genéricos.

Figura 18: Diagrama de classe da Aplicação Núcleo.

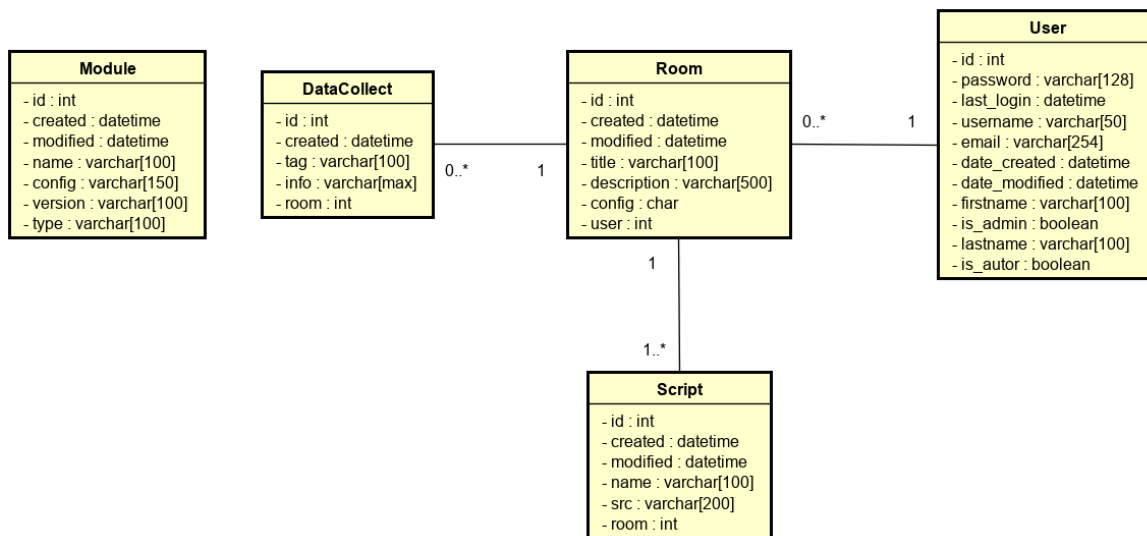


Fonte: O autor.

7.1.3. Modelo entidade relacional

Modelo da base de dados, Figura 19, como apenas a aplicação núcleo possui uma base as demais não constarão deste modelo.

Figura 19: Diagrama entidade relacional da Aplicação Núcleo.



Fonte: O autor.

7.1.4. Implementação

O controle de usuários foi estabelecido por tokens de acesso, antes que um usuário consiga consumir qualquer serviço é necessário que realize um acesso com login e senha do qual recebe um token. Portado do token, o usuário poderá acessar as demais funcionalidades do núcleo. O controle dos arquivos e jogos segue o modelo de uso do token, com exceção da inserção de arquivos de configuração realizados pelo administrador. Estes não utilizam do token, o administrador deve inserir os dados de acesso junto aos demais no consumo dos serviços exclusivos deste usuário.

Além dos usuários, foram adicionados rooms (salas) como representação dos jogos criados. Destes elementos foi feita a modelagem relacional da aplicação para criação da base de dados.

Em seguida foi modelada a aplicação com base na proposta, acrescido da gerência de usuários e de tokens de acesso. Das funcionalidades propostas foram então implementados serviços RestFul, o qual adota a arquitetura Rest que define modelos de nomenclatura para os serviços que os tornem mais intuitivos quanto a suas características, facilitando a implementação de outras aplicações que venham a consumir tais serviços.

7.2. Protótipo da aplicação editor

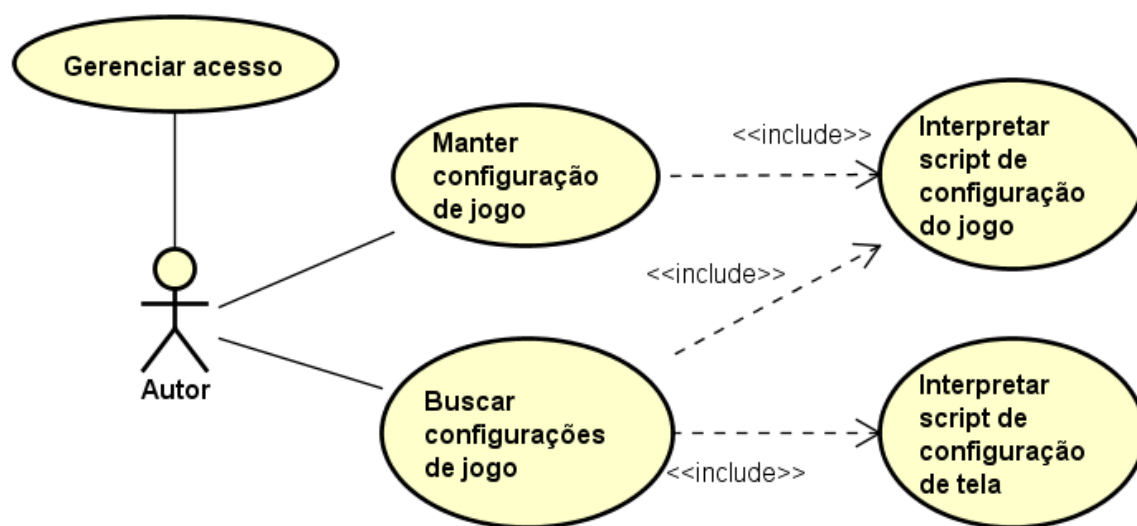
Para que não se eleve excessivamente a curva de aprendizagem no uso do ambiente foi optado pelo uso de uma interface comum a uma grande fatia de usuários atuais, páginas web. Como descrito as aplicações editor devem ser flexíveis e não dependentes de formas de armazenamento de dados, todas informações processadas pela aplicação serão advindas da aplicação núcleo. Foi optado então pelo uso do HTML (HyperText Markup Language ou Linguagem de Marcação de Hipertexto) com JQuery, uma biblioteca JavaScript, para o desenvolvimento da parte gráfica do editor e do ASP.NET MVC, uma estrutura de aplicativo web da Microsoft.

7.2.1. Requisitos funcionais

São funcionalidades da aplicação editor e exclusivas para usuários do tipo editor conforme Figura 20:

- Acessar através de um usuário do tipo editor.
- Cadastrar, editar, buscar e listar salas (jogos). Cadastrar e editar consistem de um conjunto de telas para cada elemento do jogo.

Figura 20: Caso de uso Aplicação Editor.

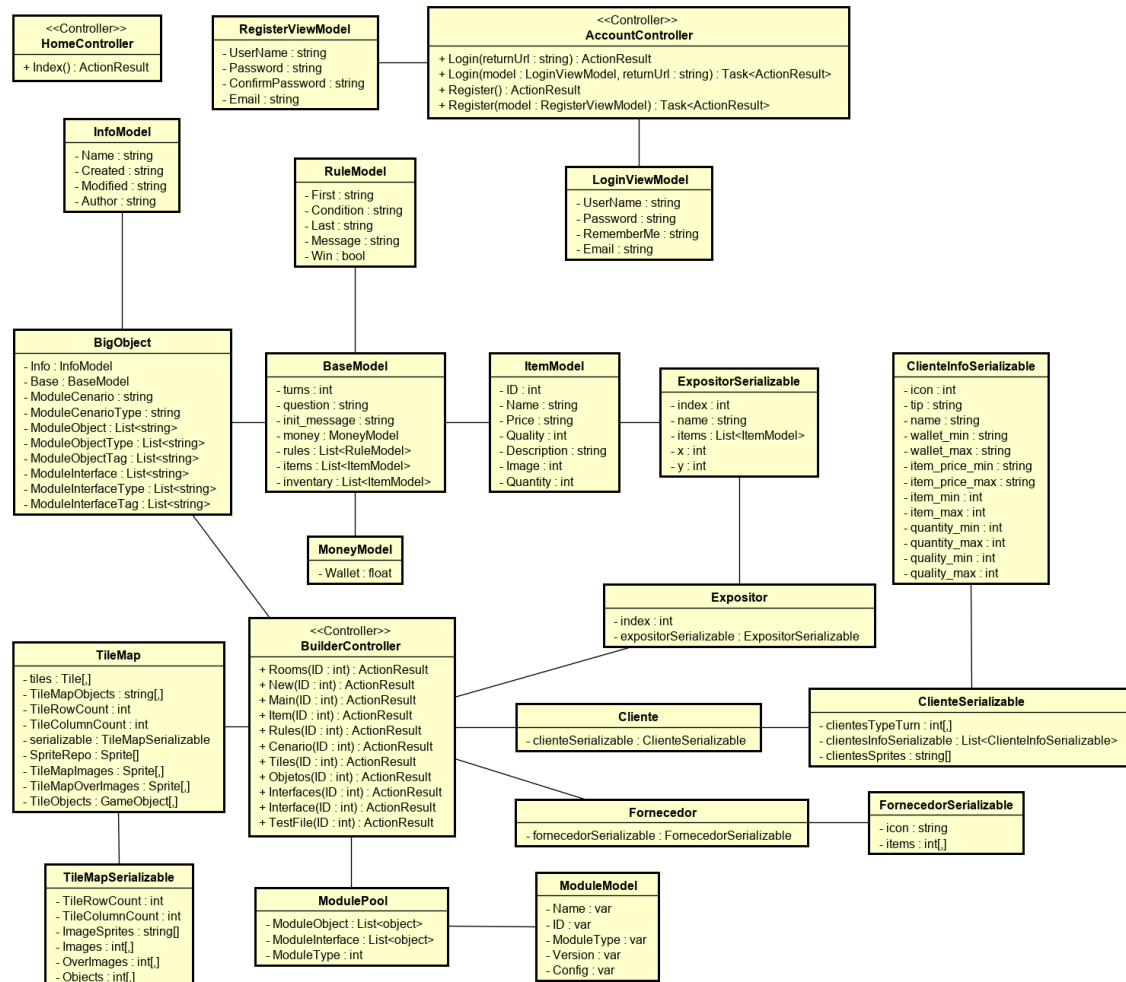


Fonte: O autor.

7.2.2. Modelagem

O javascript, apesar de ser orientado a funções, permite ser visualizado seguindo o paradigma de orientação à objetos permitindo facilmente a adequação ao padrão UML conforme Figura 21.

Figura 21: Diagrama de classe da Aplicação Editor.



Fonte: O autor.

7.2.3. Implementação

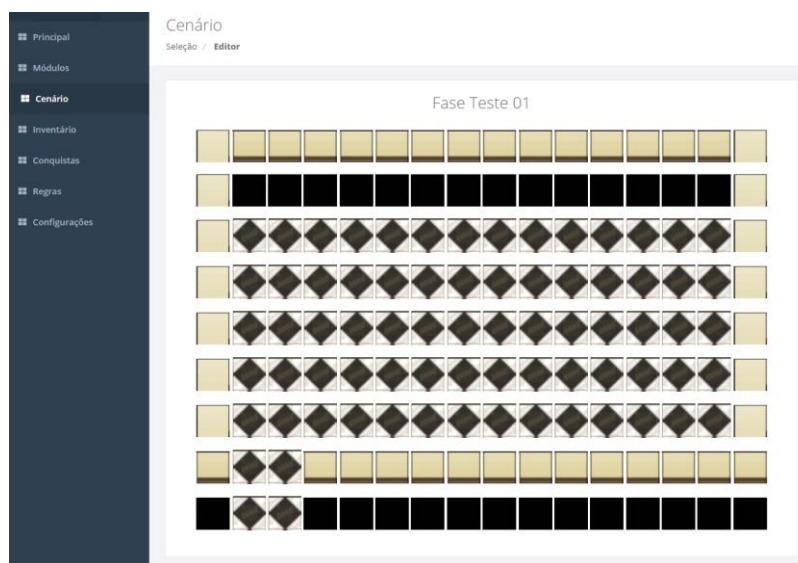
Os inputs de configuração seguem o padrão dos já existentes no desenvolvimento web: input, select, option, textarea, button e datalist. A maior parte das telas de configuração seguiu o padrão comumente encontrado em outras aplicações web composta por um formulário e um conjunto de inputs, a exemplo das telas de cenário composta por uma lista de dados de cenários da qual o autor pode selecionar para edição ou da tela de criação/edição de cenário composta por um conjunto de inputs.

Algumas configurações, no entanto, exigiram componentes adaptados para tornar seu uso mais intuitivo, a exemplo da configuração dos objetos em um cenário 2D usando o componente TileMap abordado na seção seguinte. Além dos inputs na montagem do jogo há uma hierarquia de elementos globais a serem seguidas: inventário, conquistas, cenários e módulos. No protótipo não foi dada a opção de enviar recursos personalizados, todos recursos visualizados foram pré inseridos tanto no editor quanto na aplicação cliente.

7.2.4. Editor TileMap

Nomenclatura comum para definir ferramentas em jogos onde o cenário é mapeado em uma matriz de quadrados de valor único. Utilizado para montagem do cenário ao adicionar imagens aos espaços mapeados semelhante a um mosaico, ilustrado na Figura 22. Seguindo esta ideia, o input seria utilizado por módulos cenário para a montagem gráfica dos mesmos, ao fornecer um conjunto de imagens e tipos de tiles o autor poderá montar o cenário a critério.

Figura 22: Diagrama entidade relacional da Aplicação Editor.



Fonte: O autor.

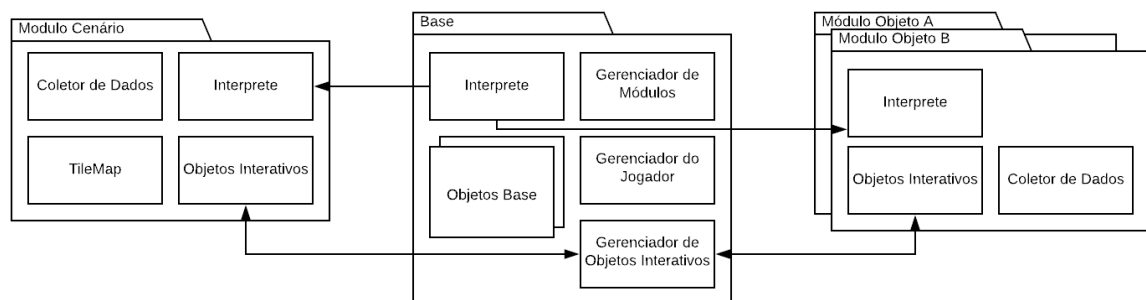
Em cada tile o autor pode definir além da imagem, se o mesmo poderá ser ocupado por um módulo objeto, se será espaço de interação e quais as possíveis interações com estes.

Ao se definir o uso de um input do tipo TileMap deve ser atribuído também o mapeamento em unidades, como 64x32 (64 tiles por 32 tiles) ou 128x64 (128 tiles por 64 tiles) e as imagens que irão ocupar os tiles para que o editor possa gerar a interface de tamanho adequado.

7.3. Protótipo da aplicação cliente

O desenvolvimento do protótipo da aplicação cliente foi dividido em três etapas, conforme Figura 23: funcionalidades básicas, módulo base e demais módulos (cenário, objetos e interfaces). A etapa dos módulos cenário, objetos e interfaces ocorreu após a conceptualização do jogo. No desenvolvimento da base fora planejado as funcionalidades relacionadas ao jogo - inventário, conquistas, navegação - e as funcionalidades relacionadas aos módulos. As funcionalidades relacionadas ao núcleo - gerência de usuários e jogos - ficaram como da aplicação.

Figura 23: Fluxo da Aplicação Cliente.



Fonte: O autor.

É importante frisar que, para cada funcionalidade relacionada ao jogo e que seja configurável, deve haver um conjunto de inputs na aplicação editor.

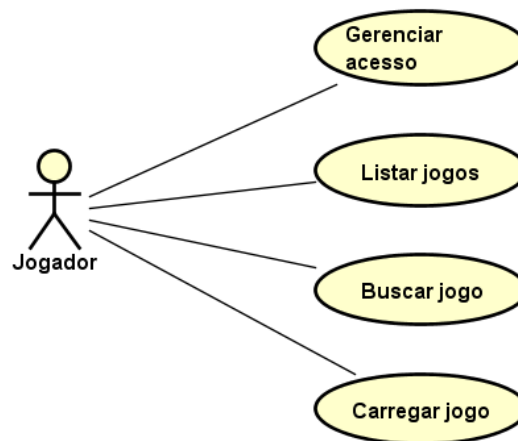
7.4.1. Requisitos funcionais

São funcionalidades da aplicação cliente conforme Figura 24:

- Acessar através de um usuário do tipo editor ou cliente.

- Buscar e listar as salas (jogos) criados por usuários do tipo editor, as buscas podem ser feitas pelo código ou nome da sala e podem retornar as informações da sala ou o arquivo de configuração.
- Montar os jogos instanciando os módulos, as buscas podem ser feitas pelo código ou nome do módulo.

Figura 24: Caso de uso Aplicação Cliente.

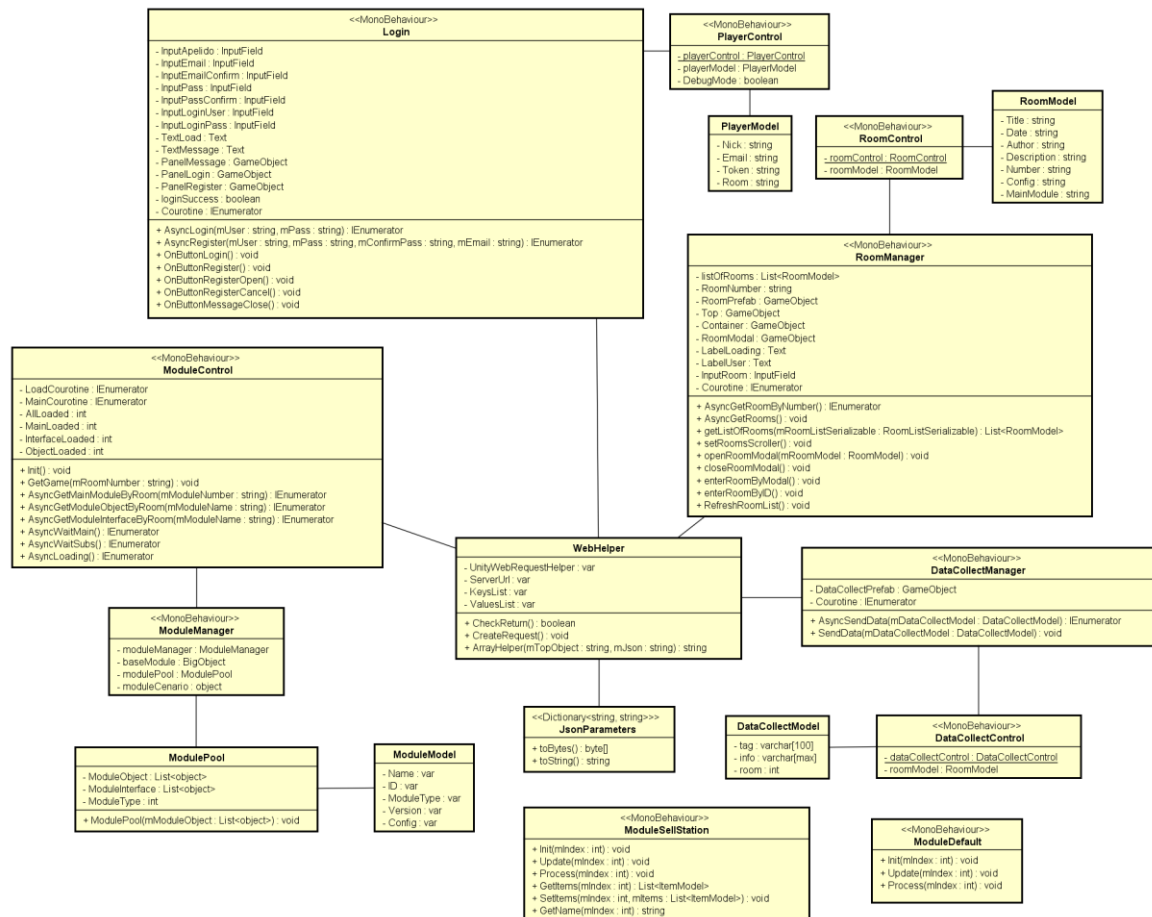


Fonte: O autor.

7.4.2. Modelagem

A programação no Unity mescla orientação a eventos e orientação à objetos, na modelagem, Figura 25, foi seguido a à objetos adaptando partes da orientação a eventos para melhor visualização.

Figura 25: Diagrama de classe da Aplicação Cliente.

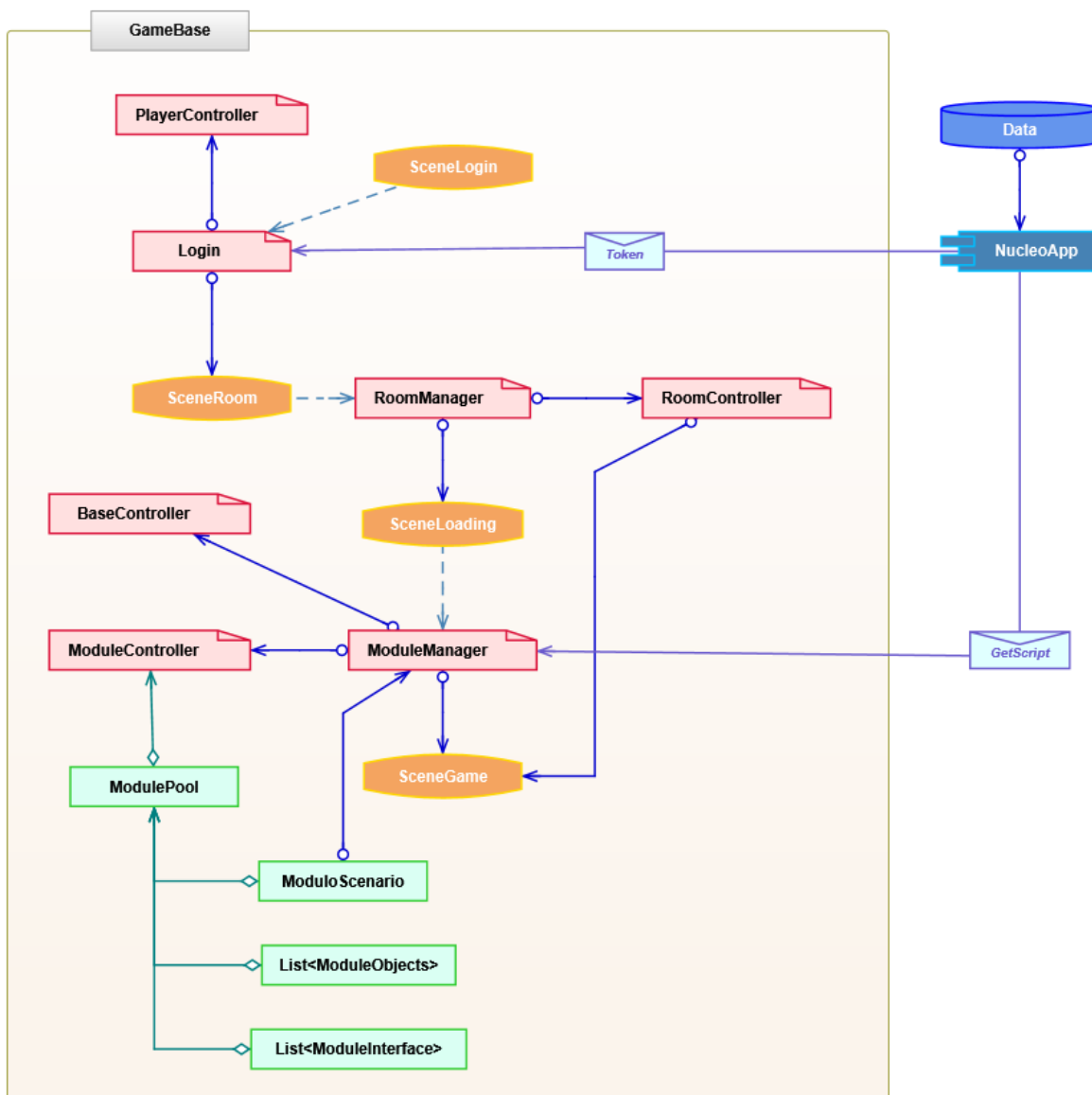


Fonte: O autor.

7.4.3. Implementação

Seguindo as observações das ferramentas existentes foi definido o uso de um motor de jogos já estabelecido, o Unity, como base para o desenvolvimento da aplicação e seus módulos. No Unity tem-se a possibilidade de desenvolvimento em C# e javascript, ambas podem ser trabalhadas seguindo a orientação a objetos, ainda que mesclado com modelo de cenas, objetos e scripts. Seguindo essa mescla, representado na Figura 26, foi criado um modelo do comportamento base da aplicação cliente.

Figura 26: Modelo do comportamento básico da Aplicação Cliente.



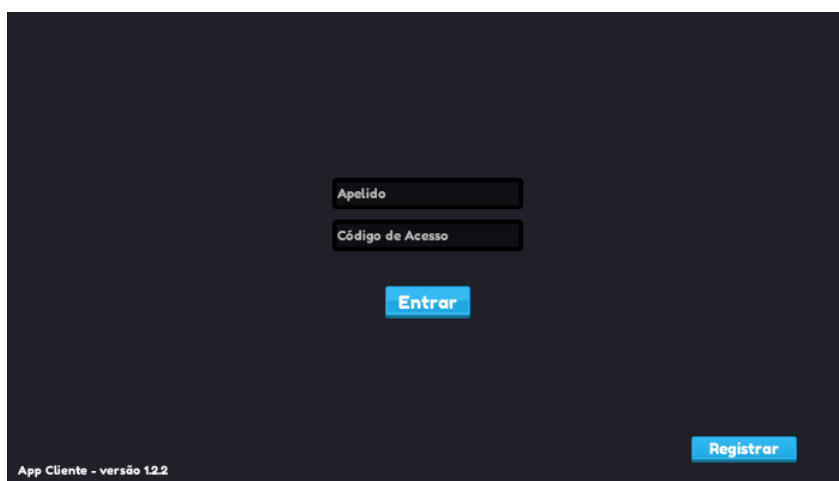
Fonte: O autor.

Da modelo base foi iniciada a modelagem do módulo que iria acompanhar essa primeira versão da aplicação cliente, o módulo cenário intérprete do TileMap. Na aplicação cliente este módulo realiza uma operação inversa do editor, mapeando a tela do jogo e preenchendo com o conteúdo configurado, cada tile ou quadrado é formado da imagem e módulo objeto definidos e configurados em camadas: a imagem em segundo plano e o objeto a frente.

Foram definidas quatro cenas no Unity em ordem de navegação: Login, Busca, Loading e Jogo. Na cena Login, Figura 27, é realizada as chamadas de acesso do

usuário ao núcleo, recuperação de acesso ou cadastro de novo usuário. Um acesso bem-sucedido resulta no token que fica armazenado durante toda atividade do usuário e o transfere para a cena de Busca, Figura 28, onde requisitados ao núcleo e listados os jogos existentes para a aplicação com informações gerais, são fornecidos também ferramentas de busca por nome ou título do jogo e acesso direto pelo código do jogo.

Figura 27: Tela de login da aplicação cliente.



Fonte: O autor.

Figura 28: Tela de salas (jogos criados) da aplicação cliente.



Fonte: O autor.

Selecionada uma sala o usuário é transferido a cena Loading onde será feita uma requisição da configuração do cliente ao núcleo, o arquivo recebido será interpretado gerando um objeto de controle dos módulos e base. O objeto de controle é enviado a cena de jogo onde será utilizado para a instanciação de todos

os objetos e interface do jogo, após instanciados a tela da cena de jogo é liberada ao usuário dando início a partida.

7.5. Base

A base é um componente da aplicação cliente responsável pela estrutura mínima do jogo, contendo os elementos comuns a diferentes jogos como itens, inventário e dinheiro, responsável também por gerir configurações técnicas como resolução, renderização e som. E por ser composta de outros elementos que venham a afetar a estrutura básica de um jogo como jogabilidade e alertas.

Foi planejada apenas um tipo de jogabilidade para o protótipo, a baseada em turnos, a cada rodada o jogador pode realizar interações e ao finalizar o turno o jogo produz os feedbacks para as ações tomadas. Será permitido ao autor configurar a duração da partida através da configuração do número de rodadas e condições de vitória ou derrota, essa a fará uso dos elementos quantitativos: itens, dinheiro, total de itens no inventário ou quantidade de um item em específico no inventário.

Como já informado o ambiente consta de itens e inventário, partes da base, os itens requerem que sejam criados pelo autor e podem ou não iniciar no inventário do jogador. Os elementos referidos como da base devem estar disponibilizados programaticamente aos módulos que vierem a ser agregados ao ambiente, tanto na aplicação cliente como no editor. Essa disponibilização ocorre, pois, a base faz o controle dos módulos.

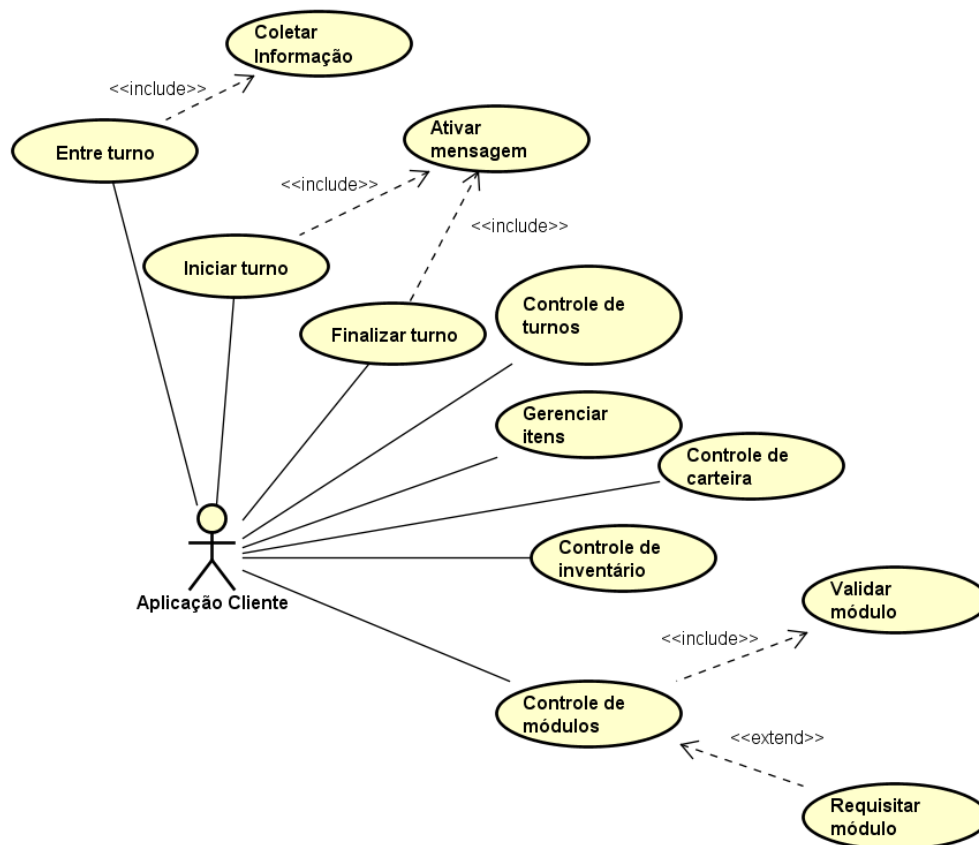
7.5.1. Requisitos funcionais

São funcionalidades da base conforme Figura 29:

- Gestão de itens.
- Controle da carteira, adição e subtração de valores.
- Controle de inventário, fluxo de itens possuídos pelo jogador ou outros objetos e suas quantidades.

- Controle de turnos, contagem dos turnos, limites e eventos como fim, início e entre turnos.
- Mensagens de início e fim da partida.
- Controle dos módulos, inicialização do módulo cenário e instanciação dos módulos interface.

Figura 29: Caso de uso Aplicação Cliente, Base.

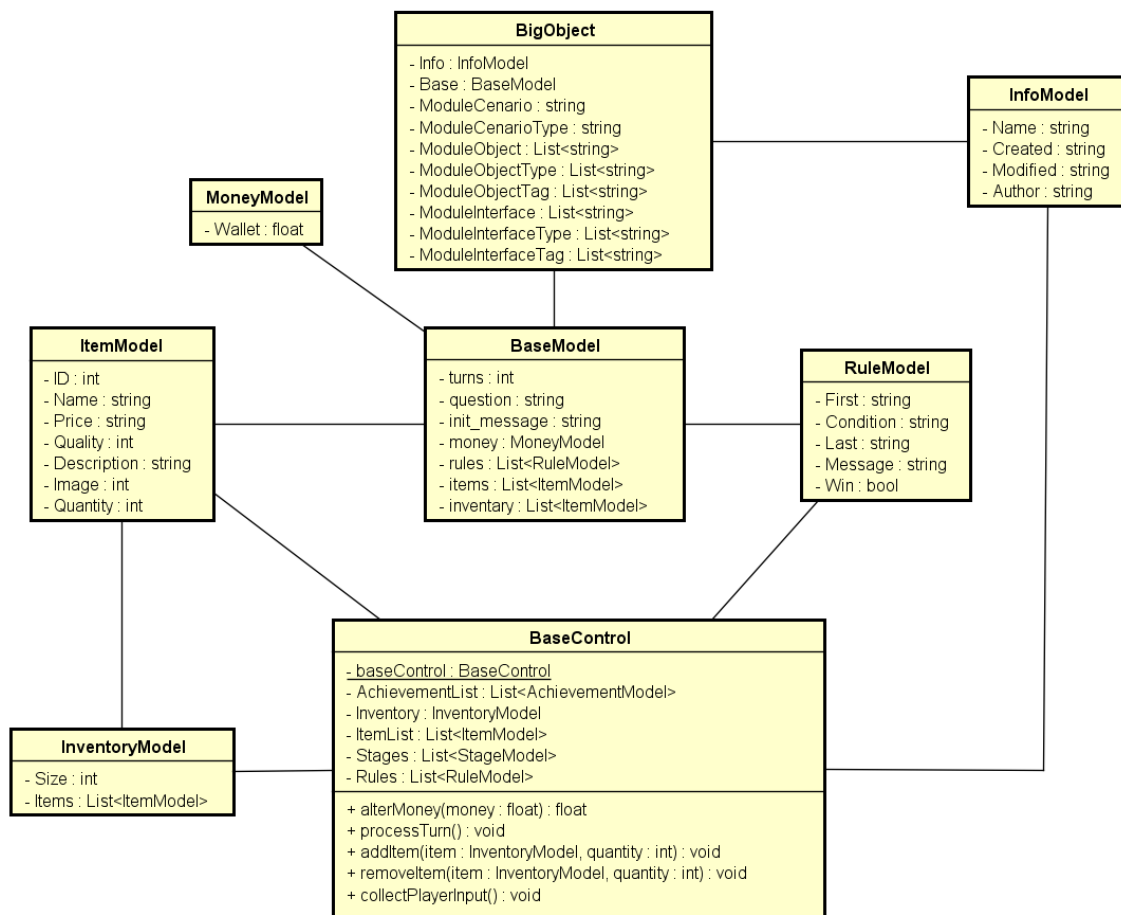


Fonte: O autor.

7.5.2. Modelagem

Seguindo os mesmos critérios da modelagem da Aplicação Cliente para a Base, representada na Figura 30, foram adaptadas as partes da orientação a eventos para melhor visualização.

Figura 30: Diagrama de classe da Aplicação Cliente, Base.



Fonte: O autor.

7.5.3. Implementação

Visualmente os elementos da base são representados pela interface de menu do jogo, Figura 31, entretanto elementos que não possuem nenhum tipo de interação com o jogador, como os itens cadastrados, devem ser invisíveis ao jogador.

Figura 31: Menu da tela de jogo.



Fonte: O autor.

No demais são criados os objetos carteira, inventário, turno e coleções de elementos como módulos, itens e condições. E em seguida apontados por uma classe de acesso às quais os outros módulos terão acesso.

Devido a estrutura na qual o Unity opera, orientado a eventos via scripts, apenas as classes que estendem `MonoBehaviour` e estão anexados a um `GameObject` serão automaticamente executados pelo motor em tempo de execução. Como tal estrutura vem das bibliotecas do motor de jogos foi criado uma classe base, nomeada de `ModuleStarter`, para todos os módulos para que se tenha padronizadas alguns métodos visto que diferentes módulos podem se comportar de formas diferente.

```
public abstract class ModuleStarter : MonoBehaviour
{
    public abstract void Init(int mIndex);
    public abstract void Update(int mIndex);
}
```

Ao estender do `MonoBehavior`, a classe `ModuleStarter` herda todas suas características permitindo que sejam executados pelo motor e acrescentando novas chamadas que serão utilizadas para iniciar e atualizar um módulo.

7.6. Módulo TileMap

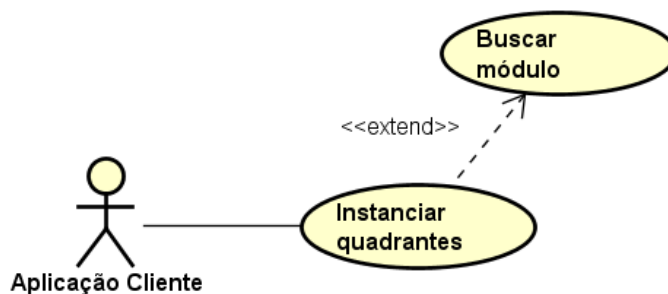
No cliente a funcionalidade do módulo `TileMap` é a de gerar o cenário criado pelo input `TileMap` do editor. Quando iniciada a montagem da cena no Unity este criará as instâncias das posições que formam a tela com as respectivas imagens e objetos (módulos) configurados no editor.

7.6.1. Requisitos funcionais

São funcionalidades do módulo conforme Figura 32:

- Instanciar os quadrantes e suas imagens.
- Instanciar os objetos módulos na posição a qual foi configurado.

Figura 32: Caso de uso Aplicação Cliente, TileMap.

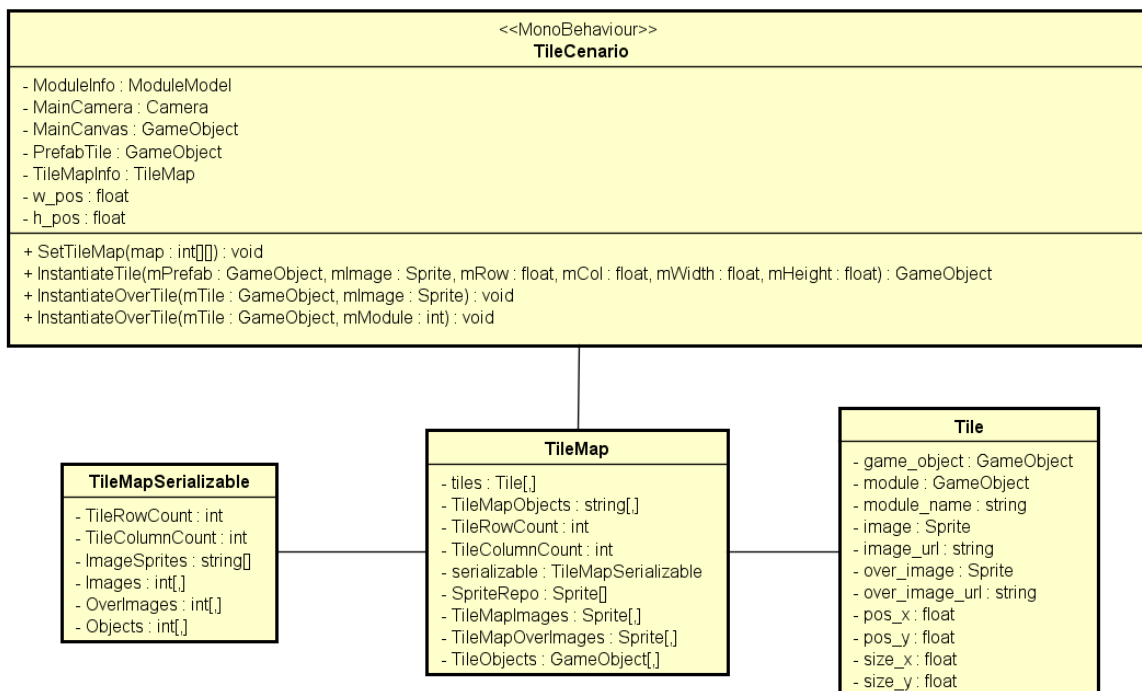


Fonte: O autor.

7.6.2. Modelagem

Seguindo os mesmos critérios da modelagem da aplicação cliente o modelo do TileMap, Figura 33, foi adaptado às partes da orientação a eventos para melhor visualização.

Figura 33: Diagrama de classe da Aplicação Cliente, TileMap.

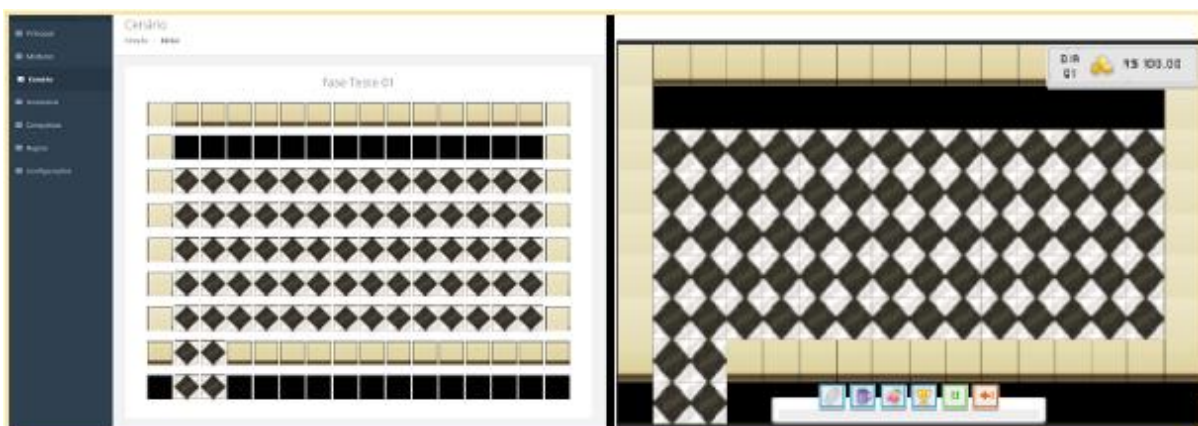


Fonte: O autor.

7.6.3. Implementação

O módulo é chamado pela base através de um método de inicialização, dando partida a instanciação de objetos prefab - objetos já referenciados em memória - que compõem a tela, ilustrado na Figura 34, na resolução definida no editor. Nesses objetos são carregadas as suas respectivas imagens e os objetos dos módulos registrados gerando uma instância do mesmo na coleção de módulos da base.

Figura 34: TileMap, Editor (à esq.) e Cliente (à dir.).



Fonte: O autor.

7.7. Módulo Expositor

Módulo objeto capaz de armazenar itens instanciados na tela a partir de um módulo cenário. Interagem com o jogador acionando uma tela de itens armazenados fornecendo as opções de remover itens ou adicionar novos itens e definir prioridade de acesso. Devem seguir algumas regras:

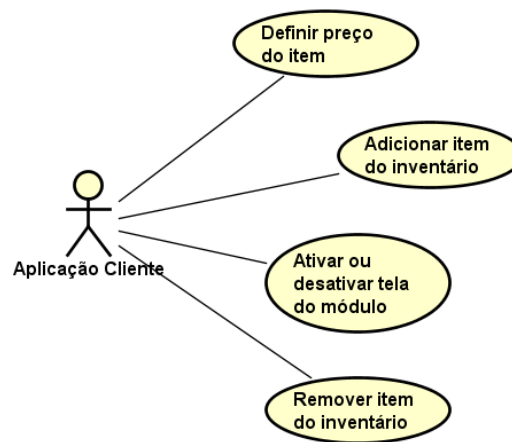
- Só são permitidos a adição de itens presentes no inventário do jogador.
- A prioridade de acesso define a posição do módulo e a ordem que estes serão encontrados pelos demais módulos ou base.

7.7.1. Requisitos funcionais

São funcionalidades do módulo conforme Figura 35:

- Ativar ou desativar a tela de controle dos itens no módulo.
- Adicionar item do inventário no módulo.
- Definir preço do item.
- Remover item do módulo para o inventário.

Figura 35: Caso de uso Aplicação Cliente, Módulo Expositor.

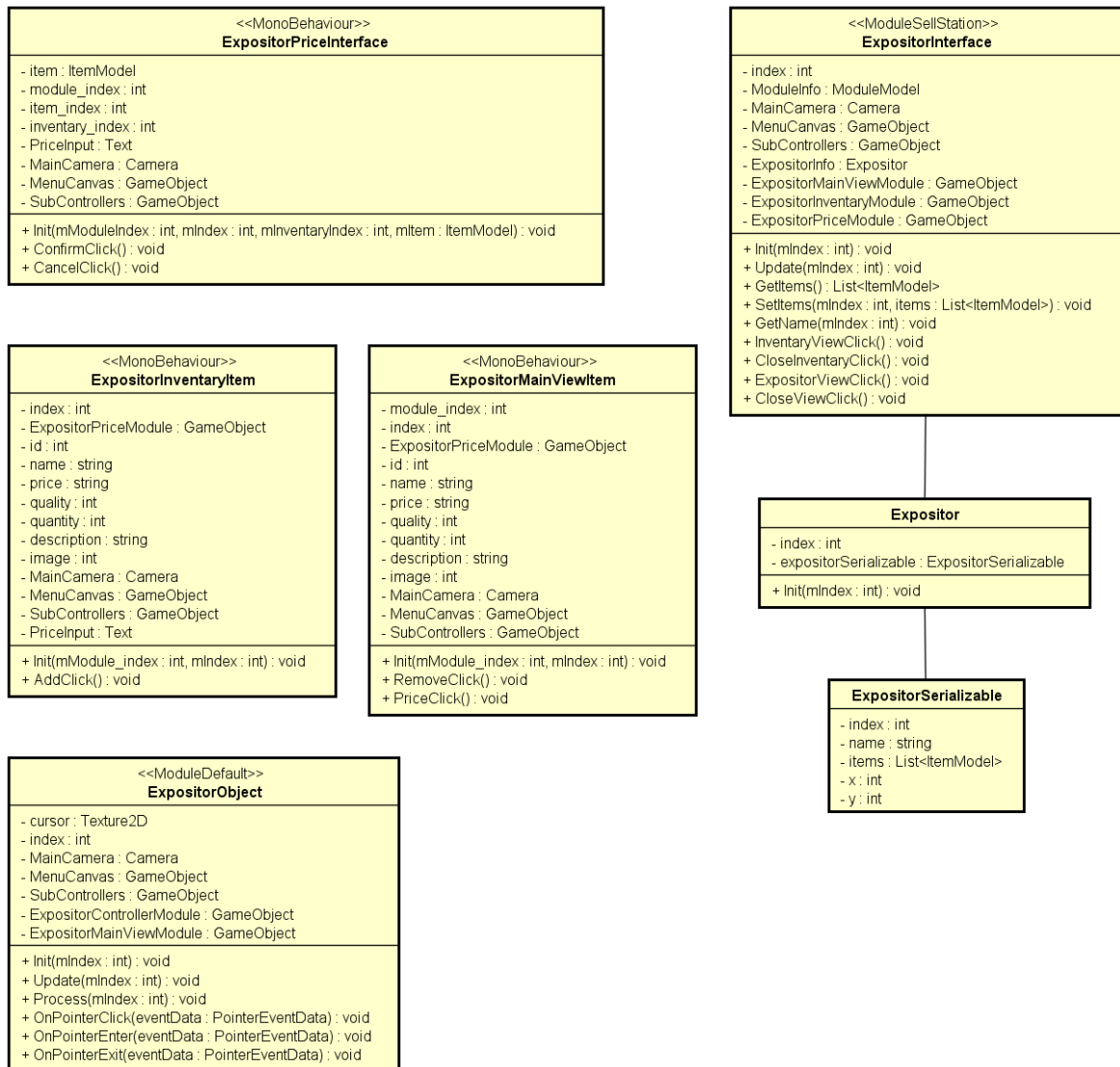


Fonte: O autor.

7.7.2. Modelagem

Seguindo os mesmos critérios da modelagem da aplicação cliente foi adaptado às partes da orientação a eventos para melhor visualização conforme Figura 36.

Figura 36: Diagrama de classe da Aplicação Cliente, Módulo Expositor



Fonte: O autor.

7.7.3. Implementação

O módulo é chamado pela base através de um método de inicialização, dando partida a instanciação de objetos prefab - objetos já referenciados em memória - que compõem a tela, Figura 37, na resolução definida no editor. Nesses objetos são carregadas as suas respectivas imagens e os objetos dos módulos registrados gerando uma instância do mesmo na coleção de módulos da base. Os módulos são identificados pela chave “NpcBuyStation” disponibilizando os métodos:

- “buyItem(string itemTag, float totalValue)” para remoção de item dentro do limite de valor “totalValue”.
- “addItem(string itemTag, int itemQty)” para adição de um item.
- “removeItem(string itemTag, int itemQty)” para remoção de um item;

Figura 37: Tela: menu de expositor.



Fonte: O autor.

7.8. Módulo Fornecedor

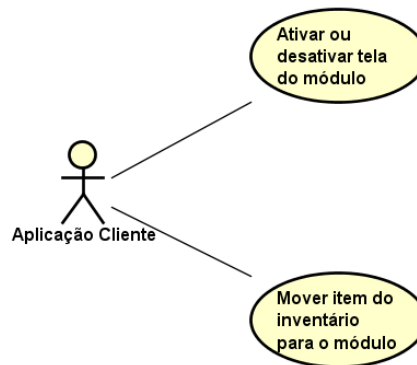
Módulo interface que disponibiliza ao jogador uma tela onde poderá, a cada turno, realizar a aquisição de diferentes itens em troca de dinheiro. Os itens que aparecerão a cada turno ficam a critério do autor.

7.8.1. Requisitos funcionais

São funcionalidades do módulo conforme Figura 38:

- Ativar ou desativar a tela de compra de itens.
- Mover os itens adquiridos para o inventário subtraindo seus valores da carteira.

Figura 38: Caso de uso Aplicação Cliente, Módulo Fornecedor.

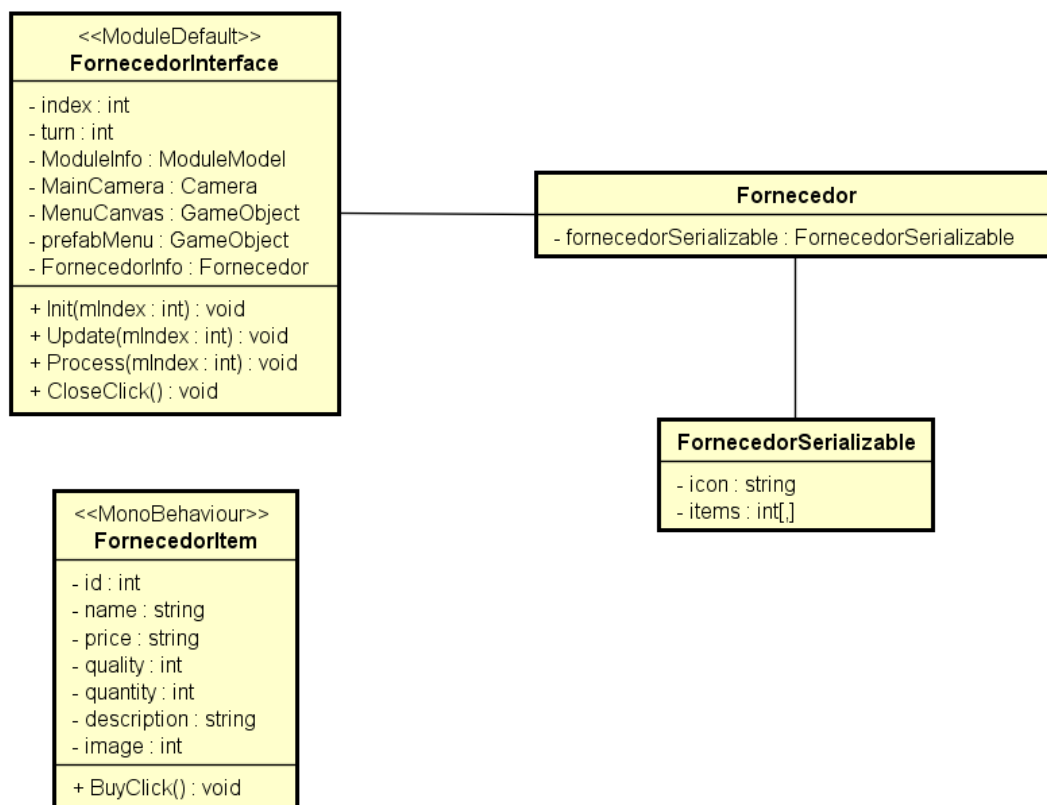


Fonte: O autor.

7.8.2. Modelagem

Seguindo os mesmos critérios da modelagem da aplicação cliente foi adaptado às partes da orientação a eventos para melhor visualização conforme Figura 39.

Figura 39: Diagrama de classe da Aplicação Cliente, Módulo Fornecedor.



Fonte: O autor.

7.8.3. Implementação

Foi criada uma listagem de itens com todas as informações disponíveis e um botão de compra representado na Figura 40. Ao clicar no botão é ativado o evento de compra. No editor só poderão ser registrados itens já criados.

Figura 40: Tela: menu de fornecedor.



Fonte: O autor.

7.9. Módulo Cliente

Um módulo interface criado especificamente para interagir com o módulo expositor retirando itens atribuídos ao módulo, nele pode ser criado clientes capazes de adquirir os produtos dos expositores. Podem ser definidos no editor clientes para um específico turno munidos de dinheiro e uma lista de itens que pretendem comprar, essas informações ficam disponíveis no turno ao qual estão registrados em um formato textual também definido pelo autor.

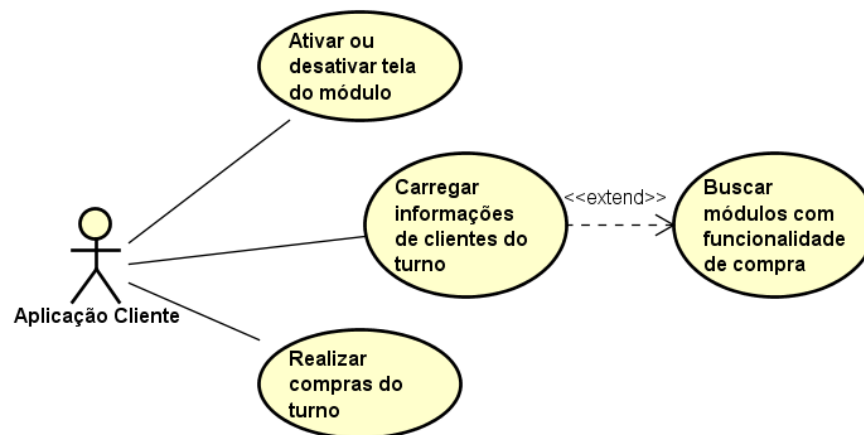
7.9.1. Requisitos funcionais

São funcionalidades do módulo conforme Figura 41:

- Executar a lista de clientes de um turno específico.

- Exibir informações de clientes do próximo turno.
- Navegar a lista de módulos buscando por módulos com métodos compatíveis, executando-os em caso positivo.
- Verificar se item da lista do cliente existe no módulo, remover até que não exista mais itens da lista nos módulos.

Figura 41: Caso de uso Aplicação Cliente, Módulo Cliente.

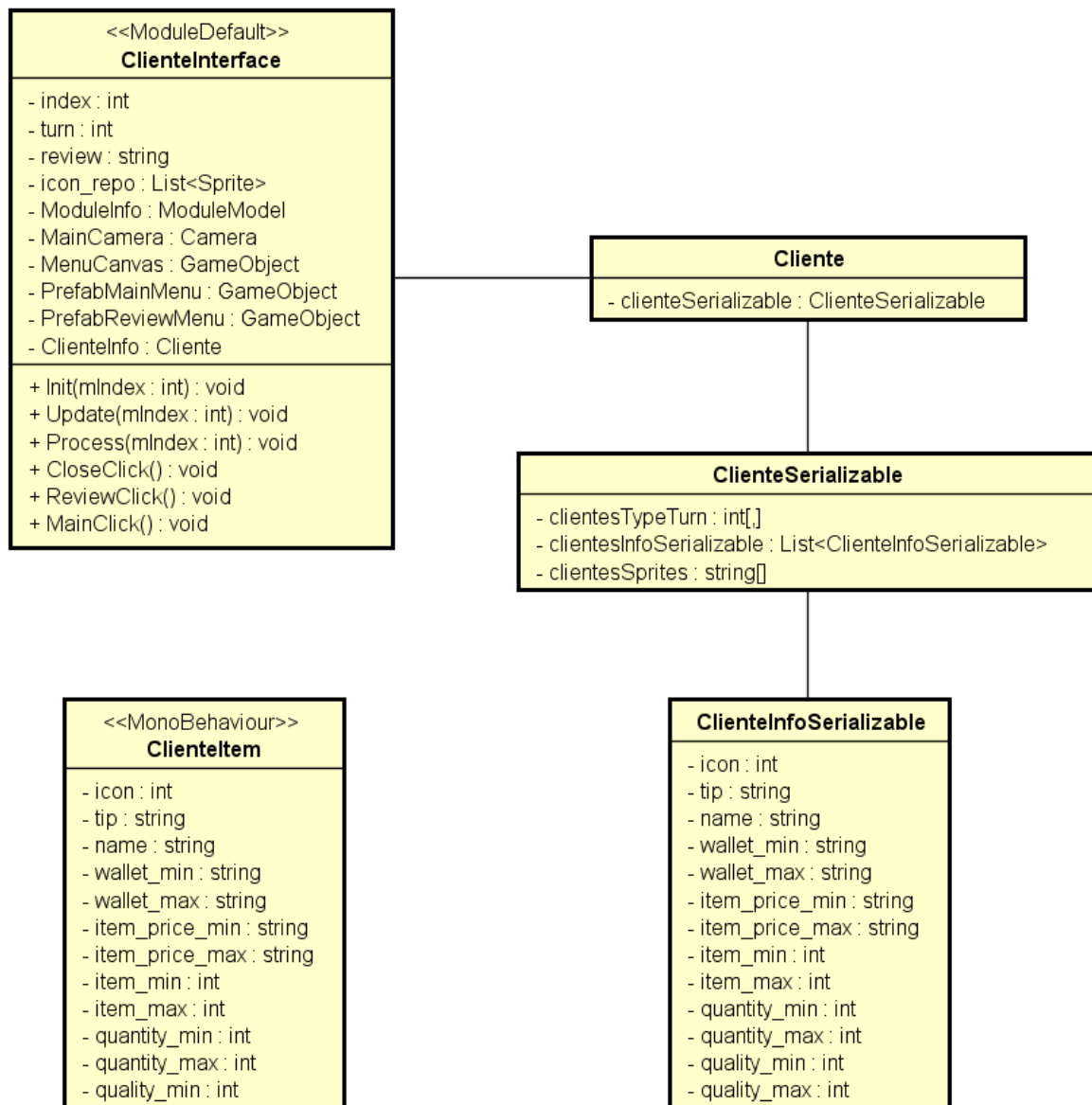


Fonte: O autor.

7.9.2. Modelagem

Seguindo os mesmos critérios da modelagem da aplicação cliente foi adaptado às partes da orientação a eventos para melhor visualização conforme Figura 42.

Figura 42: Diagrama de classe da Aplicação Cliente, Módulo Cliente.



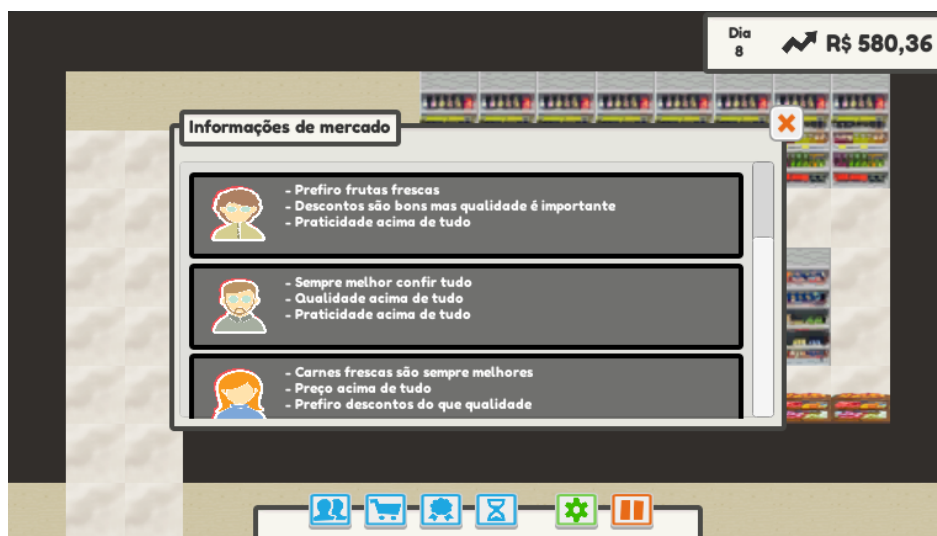
Fonte: O autor.

7.9.3. Implementação

O módulo é chamado pela base através de um método de inicialização, carregando em memória todos os clientes. A cada turno são executados os clientes relacionados navegando pela coleção de módulos em busca dos que sejam compatíveis, utilizando a chave “`NpcBuyStation`” e métodos compatíveis com a chamada “`buyItem(string itemTag, float totalValue)`”. Na tela, Figura 43, é

disponibilizado um link de menu para uma janela de descrição dos clientes para o próximo turno.

Figura 43: Tela: informações de clientes.



Fonte: O autor.

8. DESENVOLVIMENTO DO JOGO NO AMBIENTE

O jogo configurado inicia com uma tela da funcionalidade de mensagem de boas vindas da base informando a problemática e o objetivo do jogador. Fechando a tela de boas-vindas, Figura 44, é iniciado o jogo, a mensagem dessa tela fica disponível a todo momento caso o jogador necessita relembrar.

Figura 44: Tela: mensagem inicial.



Fonte: O autor.

Foram definidos 10 turnos e \$100 e como definidos na mensagem inicial, diferentes objetivos:

- Carteira positiva ao final dos 10 turnos.
- Estocar 20 itens diferentes.
- Acumular \$1000.

Atingir qualquer um dos objetivos resulta em vitória encerrando o jogo, sendo a primeira a condição básica e a demais extras permitindo o fim do jogo antes do período estimado.

Foram definidos diferentes tipos de itens e clientes por rodada, relacionando a lista de compras com os diferentes itens para produzir níveis de dificuldade, a Figura 45 ilustra o exemplo:

No primeiro turno foram disponibilizados apenas 5 diferentes itens no fornecedor com faixa de preço similares, entre 1 e 5, e 3 clientes, com listas não maiores que 2 itens diferentes e 3 similares, que verificam todos expositores e com informações simples e diretas.

Figura 45: Tela: informações de clientes simples.



Fonte: O autor.

No oitavo turno foram disponibilizados 15 diferentes itens no fornecedor com faixa de preço variando de 1 a 55 e 10 clientes com lista de até 6 itens diferentes e 10 similares, que podem vir a verificar apenas os 2 primeiros expositores e com informações que requer interpretação.

Seguindo a proposta dos jogos epistêmicos e ilustrado na Figura 46, a cada turno o jogo requiere do jogador as razões para as decisões tomadas no respectivo turno.

Figura 46: Tela: requisição de informações ao jogador.



Fonte: O autor.

9. CONSIDERAÇÕES FINAIS

Nas últimas décadas, a popularização e evolução da cultura dos jogos eletrônicos instigou avanços nas tecnologias aplicadas em termos de potencial computacional e mobilidade do hardware, nas práticas do uso e desenvolvimento de jogos e nas formas e métodos de comunicação, cooperação e colaboração para resolução de problemas. Ou seja, em toda ecologia cognitiva que permeiam os jogos eletrônicos. Tais evoluções provocaram ondas de amadorismo e profissionalismo, conforme alcançava novos grupos da sociedade que antes não tratavam com tanto entusiasmo a cultura dos jogos eletrônicos, criando, no que se tange às comunidades de desenvolvimento de jogos, grupos com alto nível profissional e criativo existindo conjuntamente a grupos extremamente amadores e sem experiência ou conhecimento do que são jogos e como criá-los. Nesse âmbito, a adoção de padrões de projeto, paradigmas de desenvolvimento ou algo mais simples como boas práticas, auxiliam no nivelamento ou elevação da qualidade dos jogos criados. O mesmo se faz valer - podendo ser até mais importante - para o desenvolvimento de jogos educacionais onde há um menor interesse da fatia dos desenvolvedores profissionalizados, devido ao menor retorno financeiro, e maior interesse da fatia de amadores e entusiastas que veem nos jogos educacionais o potencial como instrumentos no suporte à aprendizagem. A análise de modelos para desenvolvimento de jogos educacionais, das ferramentas utilizadas e de alguns jogos, junto à adaptação da teoria dos jogos epistêmicos, extraíndo um modelo para o desenvolvimento de jogos eletrônicos que fosse adequado às práticas atuais, possibilitou observar uma tendência maior a aproximação a um modelo de desenvolvimento de software do que a de um jogo eletrônico.

Resultado justificável pelo fato do ambiente de autoria não ser em si um jogo, nesse contexto, tanto a teoria dos jogos epistêmicos quanto o design de jogos tiveram o papel de modelador do escopo ao ditar como esta deve se comportar para produzir um jogo.

Portanto, não pensar no jogo educacional como um jogo que deve ser, mas apenas como uma ferramenta no suporte à aprendizagem, acaba por descartar aquilo que faz dos jogos um bom método nas práticas de aprendizagem. Perdendo características descritas por Paul Gee [Gee, 2007] relacionadas também a aprendizagem e que podem ser vitais na diferenciação de um jogo qualquer à um bom jogo. Nesse sentido, o que os ambientes de autoria para jogos educacionais podem viabilizar, em troca da liberdade criativa, são as diretrizes para a produção de um jogo que proporcione uma boa aprendizagem ao aluno. Ao desenvolver o ambiente descrito, foi possível verificar um potencial a ser explorado em seu uso como um guia para o autor, que em geral não se trata de um especialista, no desenvolvimento de um jogo educacional ao definir diretrizes fundamentadas por teorias já estabelecidas como a dos jogos epistêmicos.

Por outro lado, o potencial é limitado pelo custo de implementação de cada componente do ambiente, dependente da mecânica central. Como da proposta, ao utilizar como base o estilo de jogo por turno em um cenário formado por uma matriz de quadrados, foi adotado como solução mais prática a adequação dos componentes ao modelo para que se tornem compatíveis, o que pode inviabilizar seu uso em uma base com estilo de jogo diferente. Portanto quanto mais estilos de jogos forem suportados pelo ambiente maior será a complexidade e custo de implementação de novos componentes para que estes possam ser aplicados a diferentes estilos de jogos.

9.1. Trabalhos futuros

Mesmo que constatadas as vantagens da aplicação de modelos para o desenvolvimento de jogos educacionais em projetos de ambientes de autoria, ainda há muito a ser feito para que se possa afirmar de forma categórica as vantagens de sua aplicação. Nesse sentido, é essencial a criação de um ambiente de autoria com um maior número e variação de módulos, incluindo diferentes propostas de modelos para produção de jogos, no sentido de garantir a qualidade e uniformidade dos jogos produzidos por diferentes autores.

A adição de um maior número de módulos permitiria ainda compreender a relação entre a variedade de elementos disponíveis para a montagem dos jogos e a uniformidade dos jogos produzidos conforme os modelos aplicados. Além da adição de módulos, também é importante a adoção de um modelo para a coleta de informações — similar ao realizado na produção do jogo — que propicie a obtenção de informações relevantes, viabilizando a análise inteligente dos dados obtidos, como por exemplo: a avaliação da evolução aprendizagem, definição de perfis de aprendizagem e análise comportamental do aluno.

No ambiente proposto não foi previsto a criação de forma cooperativa ou colaborativa dos jogos, assim como o suporte a jogadores simultâneos. A adição do suporte a múltiplos usuários não só facilitaria a produção de jogos, mas também no seu uso no processo de aprendizagem, se ao seguir o modelo de produção de um jogo epistêmico até a produção do jogo no ambiente ocorre a aprendizagem por parte do aluno.

10. REFERÊNCIAS BIBLIOGRÁFICAS

Magnavox, (1972). Odyssey. Console de videogame. Califórnia, Estados Unidos.

Nintendo, (1983). Sistema de Entretenimiento Nintendo. Console de videogame. Quioto, Japão.

Crawford, C. (1997). The Art of Computer Game Design. Washington State University, REV.5.27.97.

Koster, Raph. (2004). A Theory of Fun for Game Design.

Gee, J. P. (2003). What video games have to teach us about learning and literacy. New York: Palgrave/ Macmillan.

Castorina, J. A., (1990). Piaget Vigostsky Novas Contribuições para o Debate, São Paulo, Ed Atica.

Shaffer, D.W. (2006) “Epistemic frames for epistemic games”, In: Computers and Education, 46, 223–234.

Brown, J. S., Collins , A., & Duguid, P. (1989). Situated Cognition and the Culture of Learning. Educational Researcher; v18 n1, pp. 32-42.–42.

Lave, J., & Wenger, E. (1991). Situated learning: Legitimate peripheral participation. Cambridge, UK: Cambridge University Press.

Crowley, K., & Jacobs, M. (2002). Islands of expertise and the development of family scientific literacy. In G. Leinhardt, K. Crowley, & K. Knutson (Eds.), Learning conversations in museums. Mahwah, NJ: Lawrence Erlbaum.

Shaffer, D. W. (2004a). Pedagogical praxis: the professions as models for post-industrial education. Teachers College Record, 106(7).

Koster, R., (2012). A Theory of Fun 10 Years Later. Playdom, San Diego.
Disponível em:<https://www.raphkoster.com/gaming/gdco12/Koster_Raph_Theory_Fun_10.pdf>.Acesso em: out. 2018.

Ubisoft, (2014). Child of The Light. Videogame. Microsoft Windows. Montreal, França.

Capcom, (2001). Phoenix Wright: Ace Attorney. Videogame. Osaka, Japão.

.GEARS Studio, (2013). Flappy Bird. Videogame. Android. Hanói, Vietnã.

Schell, J., (2008) The Art of Game Design : a Book of Lenses. Amsterdam; Boston :Elsevier/Morgan Kaufmann. Impresso.

Adams, E. W., 2010. Fundamentals of Game Design (2nd ed.). New Riders Publishing, Thousand Oaks, CA, USA.

3909 LLC, (2013). "Papers, Please". Videogame. Microsoft Windows.

ITycom. ITystudio. Disponível em:<<https://itystudio.com>>.Acesso em: out. 2017.

Serious Factory. Virtual Training Suite. Disponível em:<<https://www.seriousfactory.com/virtual-training-suite/>>.Acesso em: out. 2017.

Mujika, A., Oyarzun, D., Iparragirre, I., Dominguez J., & Arambarri, J. (2011). "MACHS: An authoring tool to create serious games for machine-tool operator training," *2011 9th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, Stara Lesna, pp. 147-152.

Perez-Colado, I. J., Perez-Colado, V. M., Martínez-Ortiz, I., Freire-Moran, M., & Fernández-Manjón, B. (2017) "uAdventure: The eAdventure reboot: Combining the experience of commercial gaming tools and tailored educational tools," *IEEE Global Engineering Education Conference (EDUCON)*, Athens, 2017, pp. 1755-1762.

Torrente, J., del Blanco, Á., Marchiori, E. J., Moreno-Ger P., & Fernández-Manjón, B. (2010). "<e-Adventure>: Introducing educational games in the learning process," *IEEE EDUCON 2010 Conference*, Madrid, pp. 1121-1126.

IEEE/LOM Standard (2002): IEEE Draft Standard for Learning Object Metadata, Approved specifications of the IEEE, 1484.12.1-2002, 44.

Aguar, E. V. B., & Flores, A. L. P. (2014) *Objetos de Aprendizagem: Conceitos Básicos*. In: Tarouco, L. M. R., Costa, V. M., Ávila, B. G., Bez, M. R., & Santos, E. F. (Org.). *Objetos de Aprendizagem: Conceitos Básicos*. 1ed. Porto Alegre: Evangraf, p. 12-28.

Shaffer, D. W. (2005). Epistemic games. *Innovate*, 1(6), in press.

Collins, A., & Ferguson, W. (1993). Epistemic forms and games. *Educational Psychologist*, 28(1), 25.

Shaffer, D. W. (2004b). Epistemic frames and islands of expertise: Learning from infusion experiences. Paper presented at the International Conference of the Learning Sciences (ICLS), Santa Monica, CA.

Beckett, K. & Shaffer, D.W. (2005). Augmented by reality: The pedagogical praxis of urban planning as a pathway to ecological thinking. *Journal of Educational Computing Research* 33(1): 31-52.

Svarovsky, G. N., Shaffer, D. W. (2007). SodaConstructing Knowledge through Exploratoids. *Journal of research in science teaching*, vol. 44, no. 1, pages 133–153.

Shaffer, D. W. (2006). Epistemic frames for epistemic games. *Computers and Education*, pages 223–234.

Arastoopour, G., Shaffer, D.W., Swiecki, Z., Ruis, A.R., & Chesler, N.C. (2015). Teaching and Assessing Engineering Design Thinking with Virtual Internships and Epistemic Network Analysis. Paper presented at the Harvey Mudd Design Workshop. Claremont, CA.

D'Angelo, C.M., Shaffer, D.W., & Chesler, N. (2011). Undergraduate engineers engaging and reflecting in a professional practice simulation. Paper presented at the annual meeting of the American Society for Engineering Education. Vancouver, BC.

Brasil. (2018) Ministério da Educação. Base Nacional Comum Curricular – BNCC. Brasília, DF.

Gee, J. P., (2007). Good Video Games + Good Learning : Collected Essays on Video Games, Learning, and Literacy. Nova York :P. Lang, 2007. Impresso.