

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE EDUCAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIAS
EDUCACIONAIS EM REDE – MESTRADO PROFISSIONAL

Garibaldi da Silveira Júnior

**FJSU: UM FRAMEWORK PARA O DESENVOLVIMENTO DE JOGOS
SÉRIOS UBÍQUOS**

Santa Maria, RS
2019

Garibaldi da Silveira Júnior

**FJSU: UM FRAMEWORK PARA O DESENVOLVIMENTO DE JOGOS SÉRIOS
UBÍQUOS**

Dissertação apresentada ao Programa de Pós-Graduação em Tecnologias Educacionais em Rede da Universidade Federal de Santa Maria (UFSM), como requisito parcial para a obtenção do título de **Mestre em Tecnologias Educacionais em Rede**.

Orientadora: Prof. Dra. Roseclea Duarte Medina

Santa Maria, RS
2019

da Silveira Júnior, Garibaldi
FJSU: UM FRAMEWORK PARA O DESENVOLVIMENTO DE JOGOS
SÉRIOS UBÍQUOS / Garibaldi da Silveira Júnior.- 2019.
124 p.; 30 cm

Orientadora: Roseclea Duarte Medina
Dissertação (mestrado) - Universidade Federal de Santa
Maria, Centro de Educação, Programa de Pós-Graduação em
Tecnologias Educacionais em Rede, RS, 2019

1. Jogos Sérios Ubíquos 2. Aprendizagem Baseada em
Jogos 3. Jogos sérios I. Duarte Medina, Roseclea II.
Título.

Sistema de geração automática de ficha catalográfica da UFSM. Dados fornecidos pelo autor(a). Sob supervisão da Direção da Divisão de Processos Técnicos da Biblioteca Central. Bibliotecária responsável Paula Schoenfeldt Patta CRB 10/1728.

©2019

Todos os direitos autorais reservados a Garibaldi da Silveira Júnior. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.
E-mail: garibaldi.dsj@gmail.com.

Garibaldi da Silveira Júnior

**FJSU: UM FRAMEWORK PARA O DESENVOLVIMENTO DE JOGOS SÉRIOS
UBÍQUOS**

Dissertação apresentada ao Programa de Pós-Graduação em Tecnologias Educacionais em Rede da Universidade Federal de Santa Maria (UFSM), como requisito parcial para a obtenção do título de **Mestre em Tecnologias Educacionais em Rede.**

Aprovado em 20 de março de 2019:

Roseclea Duarte Medina, Dra. (UFSM)
(Presidente/Orientadora)

Giliane Bernardi, Dra. (UFSM)

Felipe Becker Nunes, Dr. (AMF)

Santa Maria, RS
2019

Dedico esta dissertação a meus pais, demais familiares e amigos.

AGRADECIMENTOS

Primeiramente gostaria de agradecer a UFSM e ao PPGTER pela oportunidade de trabalhar em um projeto que representa minha maior área de interesse pessoal, profissional e acadêmico. Concluo esta etapa de minha vida acreditando que trabalhar com aquilo que se gosta dignifica e transforma o suor e lágrimas de momentos de pressão e estresse em sorrisos de satisfação e alegria no fim das contas.

Agradeço principalmente aos meus pais, Fátima e Garibaldi, e a toda minha família, visto o apoio que recebi em cada decisão que tomei desde o início de minha trajetória profissional e acadêmica. Também agradeço aos amigos de fé, que fizeram que todo o processo acadêmico de minha vida fosse mais leve, Jader, Yasmin e Alessandro, vocês merecem um lugar aqui.

Agradeço a todos os professores que passaram pela minha vida, desde minha trajetória inicial pela escola Dr. Lauro Dornelles em Alegrete-RS, da qual obtive a base necessária para iniciar minha jornada acadêmica.

Agradeço também aqueles professores do IFFar Campus Alegrete que fizeram parte de minha introdução na grande área da informática, tanto no curso técnico como no superior. Em especial às minhas orientadoras, do curso técnico em informática, Josiane Fontoura, onde descobri a modelagem 3D, e do curso tecnológico em Análise e Desenvolvimento de Sistemas, Andressa Falcade, onde comecei a trabalhar com mundos virtuais. Também agradeço aos professores Fábio Rossi e Cristhiano Bossardi, pelas tantas horas aula de programação, que contribuíram para que este projeto pudesse ser concluído.

Agradeço aos dedicados professores do PPGTER, que tornaram a experiência de realizar um mestrado desafiadora, prazerosa e gratificante. Em especial, agradeço à minha orientadora Roseclea Duarte Medina, pelos tantos puxões de orelhas, que me trouxeram amadurecimento, tanto pessoal como acadêmico, e por atuar como uma “roteadora” neste processo, indicando sempre o melhor caminho a seguir em minha pesquisa.

“Estar satisfeito consigo mesmo é considerar-se terminado e constrangido ao possível da condição do momento”

(Mario Sergio Cortella)

RESUMO

FJSU: UM FRAMEWORK PARA O DESENVOLVIMENTO DE JOGOS SÉRIOS UBÍQUOS

AUTOR: Garibaldi da Silveira Júnior
ORIENTADOR: Roseclea Duarte Medina

Atualmente, os jogos digitais acabam explorando o potencial de mobilidade dos dispositivos móveis, e adicionam recursos ubíquos, como conectividade com diversos tipos de sensores e dispositivos diferentes, além da adaptação de conteúdo em tempo real sem a percepção do jogador dos processos computacionais, sendo estes jogos entendidos como jogos ubíquos. Dentro deste panorama, surgiram os jogos sérios ubíquos, que são aqueles usados em contextos sérios, como a educação, saúde, treinamento, turismo etc. A proposta desses jogos vai ao encontro a necessidade dos professores de integrar as tecnologias ubíquas em ambiente de ensino, no entanto, é percebido que eles possuem dificuldades em adaptar-se às mesmas, enquanto elas já fazem parte do cotidiano dos alunos. Partindo deste princípio, faz-se necessário a existência de uma ferramenta, simples o suficiente para que professores e alunos de design de jogos digitais iniciantes consigam utilizá-la na criação de jogos sérios ubíquos. Esta dissertação apresenta a criação de um *framework* de caixa cinza para o desenvolvimento de jogos sérios ubíquos, denominado FJSU, que entrega ao desenvolvedor uma estrutura que pode ser flexibilizada e adaptada ao conteúdo conforme a proposta do jogo, que é dividida em dois módulos, sendo o primeiro relacionado ao desenvolvimento do jogo, que usa como base a engine Unity, denominado módulo de aplicação, e o segundo responsável pelo armazenamento dos dados do perfil do usuário, denominado módulo web. A presente pesquisa foi desenvolvida no Programa de Pós-Graduação em Tecnologias Educacionais em Rede da Universidade Federal de Santa Maria, na linha de pesquisa de Desenvolvimento de Tecnologias Educacionais. Como resultados da avaliação final, a partir da análise dos questionários e vídeos gravados durante essa etapa, foi constatado que mesmo com algumas limitações, o *framework* manteve resultados satisfatórios quanto seu objetivo de oferecer suporte aos alunos de design de jogos digitais como um facilitador na criação de jogos sérios ubíquos, no entanto, foi percebido a necessidade de avaliações específicas para o público docente.

Palavras-chave: Jogos Sérios Ubíquos. Aprendizagem Baseada em Jogos. Jogos Sérios.

ABSTRACT

FJSU: A FRAMEWORK FOR THE DEVELOPMENT OF UBIQUITOUS SERIOUS GAMES

AUTHOR: Garibaldi da Silveira Júnior
ADVISOR: Roseclea Duarte Medina

Nowadays, digital games end up exploiting the mobility potential of mobile devices, adding ubiquitous features such as connectivity to different types of sensors and different devices, and adapting content in real time without the player's perception of computational processes. games understood as ubiquitous games. Within this panorama, the ubiquitous serious games, which are those used in serious contexts such as education, health, training, tourism, etc., have emerged. The proposal of these games meets the need of the teachers to integrate the ubiquitous technologies in the teaching environment, however, it is perceived that they have difficulties in adapting to them, while they are already part of the daily life of the students. Based on this principle, it is necessary to have a tool, simple enough for education professionals, and digital game design students to be able to use it in the creation of serious games that are ubiquitous. This dissertation presents the creation of a gray box framework for the development of serious, ubiquitous games, called FJSU, which provides the developer with a structure that can be flexible and adapted to the content according to the game proposal, which is divided into two modules. the first one related to the development of the game, based on the Unity engine, called the application module, and the second responsible for storing user profile data, called the web module. The present research was developed in the Graduate Program in Educational Technologies in Network of the Federal University of Santa Maria, in the line of research of Development of Educational Technologies. As a result of the final evaluation, from the analysis of the questionnaires and videos recorded during this stage, it was observed that even with some limitations, the framework maintained satisfactory results in terms of its support of the game design student as a facilitator in the creation of serious games ubiquitous, however, it was perceived the need for specific assessments for the teachers public.

Keywords: Ubiquitous Serious Games. Game Based Learning. Serious Games.

LISTA DE FIGURAS

Figura 2.1 - Elementos de um Jogo Sériu Ubíquo	24
Figura 2.2 - <i>Walk Rally</i> - Jogo sensível a localização	29
Figura 2.3 - Tower Defense Exergame	30
Figura 2.4 - <i>Apple Harvesting Game</i> - Jogo para reabilitação	32
Figura 2.5 - Jogo adaptativo para o aprendizado de SQL	34
Figura 2.6 - Arquitetura da <i>engine Uimpala</i>	36
Figura 2.7 - Tipos de <i>Frameworks</i>	41
Figura 2.8 - Cartas do jogo <i>Hyper Try</i>	42
Figura 2.9 - Estrutura do <i>uOS</i>	43
Figura 2.10 - <i>Framework</i> para jogos adaptativos	44
Figura 2.11 – Criação de Exergames para ensino e aprendizagem de dança	45
Figura 3.1 - Etapas do processo metodológico	49
Figura 3.2 - Elementos reusáveis do FJSU	51
Figura 3.3 - Propriedades dos objetos	53
Figura 4.1 – Estrutura Simplificada do Framework	59
Figura 4.2 - Escrita do método <i>setMovement()</i>	60
Figura 4.3 - Estrutura de classes do <i>Framework</i>	61
Figura 4.4 - Menu do FJSU	63
Figura 4.5 - Página inicial do módulo web	65
Figura 4.6 - Página de administrador do módulo Web.	66
Figura 4.7 - Página de documentação	67
Figura 4.8 - Exemplo de caminho seguido pelo desenvolvedor	65
Figura 5.1 – Tipos de dispositivos	70
Figura 5.2 - Sensores	71
Figura 5.3 - Jogo Sériu Ubíquo desenvolvido	78
Figura 5.4 - Fotos da avaliação final do FJSU	90
Figura 5.5 - Telas dos jogos criados	98

LISTA DE QUADROS

Quadro 2.1. Comparação entre os <i>frameworks</i>	47
Quadro 5.1 - Respostas da avaliação enquanto caixa branca	79
Quadro 5.2 - Elementos alcançados nos jogos da pré-avaliação	82
Quadro 5.3 - Respostas do formulário da pré-avaliação	90
Quadro 5.4 - Respostas referentes ao grupo que utilizou o FJSU	91
Quadro 5.5 - Respostas referentes ao grupo que utilizou o <i>Unity</i> puro	95
Quadro 5.6 - Avaliação dos Vídeos	97

LISTA DE ABREVIATURAS E SIGLAS

UBIGAMES

GPS

QR

FJA

FJSU

NFC

Jogos Ubíquos

Sistema de Posicionamento Global

Quick Response

Framework para Jogos Adaptativos

Framework de Jogos Sérios Ubíquos

Near-field Communication

LISTA DE APÊNDICES

Apêndice A - Processos do levantamento de requisitos	114
Apêndice B - Caso de uso do FJSU	120
Apêndice C - Formulário da avaliação caixa branca	121
Apêndice D - Formulário da avaliação caixa cinza	122
Apêndice E - Roteiro de avaliação para quem utilizou o <i>Unity</i> puro	123

I

SUMÁRIO

1 INTRODUÇÃO	16
1.1 MOTIVAÇÃO	19
1.2 PROBLEMA	21
1.3 OBJETIVOS	22
1.3.1 Objetivo geral	22
1.3.2 Objetivos Específicos	22
2 JOGOS UBÍQUOS	24
2.1 CONECTIVIDADE	26
2.2 ADAPTAÇÃO DE CONTEÚDO.....	27
2.3 <i>LOCATION BASED GAMES</i>	28
2.4 <i>EXERGAMES</i>	30
2.5 JOGOS PARA REABILITAÇÃO	33
2.6 JOGOS ADAPTATIVOS	34
2.7 DESENVOLVIMENTO DE JOGOS UBÍQUOS	36
2.7.1 Unity.....	37
2.8.1 <i>Framework de Domínio</i>	40
2.8.2 <i>Framework de Caixa branca, preta e cinza</i>	41
2.9 TRABALHOS CORRELATOS	42
2.9.1 <i>Tardigrade</i>	42
2.9.2 <i>uOS</i>	43
2.9.3 <i>Framework</i> para jogos adaptativos	45
2.9.5 Comparação entre os <i>frameworks</i>	47
3. METODOLOGIA	50
3.1 FUNDAMENTAÇÃO TEÓRICA.....	50
3.2 DESENVOLVIMENTO DO PRODUTO	52
3.2.1 Módulo de aplicação.....	52
3.2.3 Módulo <i>Web</i>	54
3.3 AVALIAÇÃO	55
4. DESENVOLVIMENTO DO PRODUTO	58
4.1 DESENVOLVIMENTO DO <i>FRAMEWORK</i> FJSU	58
4.2 MÓDULO APLICAÇÃO	60
4.2.1 Caixa Branca.....	61
4.2.2 Caixa Cinza	63
4.3 MÓDULO <i>WEB</i>	65

4.4	MODELAGEM DE JOGOS SÉRIOS UBÍQUOS	68
5.	AVALIAÇÃO	71
5.1	RESULTADOS DO MAPEAMENTO SISTEMÁTICO	71
5.1.1	Q1: Como está sendo feita a entrada de dados no sistema?	71
5.1.2	Q2: Como os dados estão sendo tratados pelo sistema?.....	73
5.1.3	Q3: Como estão sendo avaliados os jogos?	75
5.1.4	Q4: Quais tecnologias estão sendo utilizadas no desenvolvimento dos jogos?	76
5.1.5	Discussão	77
5.2	AVALIAÇÃO COMO CAIXA BRANCA	78
5.3	AVALIAÇÃO COMO CAIXA CINZA.....	81
5.3.1	Pré-Avaliação	82
5.3.2	Avaliação Final	89
5.4	CONSIDERAÇÕES DA AVALIAÇÃO.....	101
6.	CONCLUSÃO.....	103
6.1	RESUMO DO TRABALHO	103
6.2	CONTRIBUIÇÕES	104
6.3	DIFICULDADES E LIMITAÇÕES	105
6.4	TRABALHOS FUTUROS	106
6.5	CONSIDERAÇÕES FINAIS	106
	REFERÊNCIAS	108

1 INTRODUÇÃO

A popularização dos dispositivos e tecnologias móveis trouxe demandas e possibilidades até então desconhecidas em algumas áreas de atuação. Krumm (2016) explica que a computação ubíqua representa a terceira era da computação, sendo a primeira era definida pelos *mainframes* – um computador de grande dimensão, onde várias pessoas o utilizam ao mesmo tempo para fins organizacionais – e a segunda pelos PCs (computadores pessoais).

Segundo Krumm (2016), a computação ubíqua retrata a expansão da popularidade dos dispositivos de pequena escala ligados por redes sem fio, sendo eles *smartphones*, *tablets*, *notebooks*, *smartwatches* e demais periféricos, que acabam fazendo parte das atividades diárias das pessoas. Estes dispositivos têm o papel de facilitar, tornar mais prática ou prazerosa a realização de uma tarefa diária comum, como pedir um táxi através de uma aplicação que reconheça a localização do dispositivo, incentivar a realização de uma atividade física, usando o *smartphone* para reconhecer a quantidade de passos ou quilômetros que a pessoa andou no dia, oferecendo algum *feedback* como motivação, ou também adaptar algum conteúdo através de informações obtidas quanto ao perfil do usuário.

Como descrito por Licoppe (2016), a computação ubíqua enriquece os ambientes urbanos de tecnologia computacional, aumentando as possibilidades de conexão e interação com os dispositivos presentes no ambiente, e quando somado às tecnologias móveis, não limitam seu espaço de interação a um ambiente estático geograficamente. Rabari & Storper (2014) chamam essa gama de sensores e dispositivos presentes no ambiente urbano de pele digital das cidades, sendo responsável por transmitir, receber e automatizar o uso das informações e dos dados gerados, tanto pelos usuários através de seus dispositivos móveis quanto pelo próprio ambiente físico.

De acordo com Salim & Haque (2015), esse conceito de pele digital permite a existência de intervenções urbanas, que envolvem a participação e interação das pessoas com o ambiente através dos dispositivos ubíquos, colocando um objetivo a ser alcançado através da interação com o ambiente físico. Os recursos usados podem ser a localização geográfica de pontos turísticos, objetos presentes no ambiente e as

próprias pessoas, onde a interação com os mesmos é feita no ambiente virtual, através dos dispositivos móveis.

Através do conceito citado anteriormente, surgiram os jogos ubíquos, que de acordo com McGonigal (2006), são aqueles que utilizam tanto recursos do ambiente virtual e o real na mecânica do jogo. Valente (2015) traz o contexto de pervasividade em jogos, que evoluiu para diferentes formas, como jogos de realidade virtual e os chamados *crossmedia games*, que são aqueles onde o jogo poderá ter participações de diferentes ambientes, mídias e dispositivos heterogêneos, não significando que todos devem atender o mesmo objetivo, mas sim que sejam parte de um todo. Esses conceitos também fazem parte dos jogos ubíquos, de forma que eles devem adaptar-se aos diferentes tipos de tecnologias e dispositivos existentes na atualidade.

Dentre as possibilidades trazidas pela era da computação ubíqua, os jogos para dispositivos móveis possuem destaque. De acordo com informações da empresa Newzoo (2018), que está à frente das principais pesquisas do mercado de jogos mobile, 51% do faturamento da indústria de jogos está relacionado a dispositivos móveis, superando os consoles de videogame e computadores, que anteriormente, mantinham uma maior fatia do mercado. Além disso, o relatório de Gartner (2018) apresenta tanto ambientes ubíquos como experiências imersivas como tecnologias emergentes para o ano de 2018. Tal informação quando interpretada com enfoque no mercado de games, reflete em uma maior mobilidade do jogador, e também na inibição da interação dele com os processos computacionais presentes no jogo, tornando-os automáticos e adaptados ao seu perfil, aumentando a imersão da experiência.

A mobilidade representa o principal quesito para a vantagem dos dispositivos móveis quando comparado às outras plataformas de jogos. Além disso, ela não atingiu apenas jogos casuais e de entretenimento, mas também aqueles denominados sérios, que possuem um propósito de treinamento ou aprendizado, possibilitando, por exemplo, o aprendizado de uma nova língua em uma viagem de ônibus.

Os Jogos Sérios, de acordo com a visão de Girard (2013), são aqueles que combinam o aprendizado com o entretenimento de um jogo, trazendo uma proposta voltada para áreas que envolvem o treinamento, educação, aquisição de conhecimento, desenvolvimento de habilidades etc. De acordo com Boyle et. al (2016), os jogos sérios podem trazer uma mudança de comportamento positiva, visto que o engajamento com o jogo pode trazer um tipo de aprendizado não intencional,

onde o jogador mantém o foco no desafio proposto, sem perceber que está adquirindo conhecimento com isso.

Unindo o propósito dos jogos sérios e da computação ubíqua, surgiram os jogos sérios ubíquos, que de acordo com Hwang (2009), podem ser entendidos como aqueles que trazem características digitais e físicas unidas por um propósito educacional, tornando o aprendizado sensível a contexto. O autor explica que desta forma, os jogos desenvolvidos podem detectar a situação em que o aluno se encontra, buscar informações relativas ao seu perfil, adaptar as mecânicas do jogo e prover conteúdo específico para suas necessidades e dificuldades.

Mortara (2014) complementa ao dizer que através da utilização de *smartphones* e demais dispositivos móveis, os jogadores podem conectar-se e compartilhar objetivos em uma perspectiva entre o mundo digital e o real. Já segundo Pimenta (2014), os jogos ubíquos trazem tanto os desafios de jogos eletrônicos como os da computação ubíqua, onde os mesmos procuram explorar diferentes formas de interação com o ambiente e realidade do jogador para promover imersão e engajamento ao jogo.

Essa perspectiva de ubiquidade adicionada aos jogos sérios, onde o desafio proposto permeia entre o real e o virtual, segundo Hwang (2016), possibilita alavancar a imersão do jogador na atividade proposta, onde desde que a atividade seja desafiadora, ela é capaz de não somente estimulá-lo em realizá-la, mas também na performance do aprendizado.

Dentro desta proposta, atividades educacionais podem ser desenvolvidas usando esses recursos que já fazem parte da rotina diária dos alunos. O estímulo da criatividade que os jogos digitais trazem, como citado por De Sena (2016), pode permitir que conteúdos de difícil assimilação possam ser entendidos rapidamente influenciando a curiosidade dos aprendizes, visto que a utilização de recursos como a câmera do *smartphone*, sensores de aproximação e de movimento, em uma atividade lúdica, podem instigá-los ao aprendizado sem que os mesmos percebam.

Para esta pesquisa, serão entendidos por ubíquos aqueles jogos sérios que como descritos por Klopfer (2012), permitem a mobilidade do jogador e buscam através de alguma tecnologia ou sensor, dados relacionados ao contexto do usuário ou ambiente, tratando-os de forma a facilitar a interação e adaptar o próprio conteúdo do jogo às necessidades do usuário.

Se faz necessário que exista uma organização de conceitos que permita praticidade no desenvolvimento de jogos sérios. De acordo com Battistella (2016), para que sejam alcançados os benefícios relacionados a motivação e engajamento dos alunos nas atividades educacionais existentes nos jogos sérios, eles devem ser desenvolvidos considerando aspectos fundamentais de design de jogos e também aspectos instrucionais.

No processo de desenvolvimento, considera-se aspectos fundamentais do design de jogos, como gênero, plataforma, modo de interação, regras, narrativa, mecânica e elementos do jogo (como personagens, cenários, objetos) (BATTISTELLA, 2016, p. 41).

Existem alternativas facilitadoras de desenvolvimento de *software*, que muitas vezes possibilitam o reuso de partes de código e permitem a integração de elementos prontos sem que seja necessário programá-los diretamente. São chamados de *frameworks* aquelas estruturas de *software* desenvolvidas para um domínio específico, que abordam os conceitos e aspectos predominantes deste, atendendo seus requisitos e necessidades, que possibilitam através do seu reuso a criação de outros *softwares* semelhantes (BRAGA, 2002).

Desta forma, analisando a influência das tecnologias ubíquas na sociedade atual, e visando uma simplificação da produção de jogos sérios que utilizam das mesmas, facilitando o processo de produzir um jogo sério ubíquo, de forma a atingir não só profissionais da área de tecnologia e *design de games*, mas também professores, possibilitando que mesmo aqueles que possuem pouco conhecimento das tecnologias que envolvem os jogos digitais consigam criar um jogo sério ubíquo para ser utilizado em suas aulas, é apresentado neste trabalho a criação do FJSU, um framework para o desenvolvimento de jogos sérios ubíquos.

1.1 MOTIVAÇÃO

A atual demanda das tecnologias ubíquas torna-se parte da indústria de jogos eletrônicos, que está cada vez mais buscando novas experiências com o intuito de incentivar e empolgar o jogador com o conteúdo abordado. Os jogos sérios ubíquos tendem a retirar a atenção do usuário dos processos computacionais existentes na sua aplicação, e voltá-la ao conteúdo abordado pelo jogo, camuflando a tecnologia existente (RODRIGUES, 2016).

Partindo do princípio que o desenvolvimento de jogos usando tecnologias ubíquas e adaptativas quanto ao seu conteúdo e conexão não mantém uma organização simplificada de entendimento teórico e prático, onde, de acordo com Buzeto (2015), fazem parte do desenvolvimento de jogos ubíquos, desafios relacionados a questões de heterogeneidade, mobilidade, integração espontânea e sensibilidade a contexto. A motivação para este trabalho foi sustentada pelo entendimento de uma forma que venha a apoiar os profissionais desenvolvedores de jogos e professores que possuem interesse nesse tipo de jogo.

Como dito por Muller & Cruz (2016), existe uma urgência quanto a adaptação do docente às mídias digitais atuais, referindo-se especialmente aos jogos. As autoras citam que nesse contexto, existem poucas ofertas para a formação de professores na utilização de jogos em ambiente escolar. Da mesma forma, De Sena (2016) cita que é necessária uma adaptação do sistema educacional atual, de forma a abranger a utilização de jogos digitais no processo de ensino, visto que eles têm potencial de estimular o pensamento criativo e inovador.

De acordo com Boyle et. al (2016), houve um afastamento da utilização de jogos sérios em quesitos educacionais devido as dificuldades encontradas na integração dos mesmos nos currículos, e também na adaptação do docente quanto a utilização deles. Além disso, o desenvolvimento dos jogos costuma ser complexo, e envolver custos financeiros e de tempo elevados.

Nesse contexto, foi buscada uma alternativa que venha a suprir esta necessidade, uma ferramenta simples o suficiente para que um professor com pouco conhecimento em tecnologia consiga desenvolver um jogo sério ubíquo. O entendimento do tipo de ferramenta escolhida para alcançar a demanda vista foi baseado no estudo de padrões de *software*, e analisando o trabalho de Braga (2002), foi percebido que um *framework* de domínio específico seria uma alternativa viável.

O desenvolvimento de jogos sérios ubíquos a partir de um *framework* pode simplificar o trabalho dos desenvolvedores, além de minimizar os esforços e recursos, como o tempo e a contratação de novos profissionais com diferentes habilidades, necessários para a implementação do jogo. De acordo com Johnson (1988), *frameworks* são baseados nos princípios da programação orientada a objetos, tendo o seu reuso por herança, e a ligação de componentes e objetos de maneira dinâmica. Além disso, como dito por Klock (2017) em relação a *frameworks*, pouco tem sido

explorado a questão de adaptação de conteúdo educacional para jogos conforme o perfil do usuário em ambientes computacionais.

A utilização do mesmo oferece ao desenvolvedor um suporte aos processos de desenvolvimento e criação dos módulos e elementos que compõem um jogo sério ubíquo, além de apresentar uma estrutura modelo flexível, que pode ser adaptada conforme a necessidade do desenvolvedor.

1.2 PROBLEMA

Devido à quantidade de conceitos e tecnologias envolvidas no processo de criação de um jogo sério ubíquo, a implementação do mesmo pode ser demorada e trabalhosa, visto que além do desafio de desenvolver um jogo, o profissional também deve conhecer as características dos dispositivos que o jogador irá utilizar, tal como o conteúdo do jogo.

Essa tarefa pode ser bastante complicada para um profissional da educação que não possua conhecimento das tecnologias envolvidas no processo de criação dos jogos sérios ubíquos, visto o tempo demandado para conhecer e estudar cada uma delas, além dos conceitos específicos ao *design de games*.

Da mesma forma, é complexo para os designers de jogos digitais assimilarem o conteúdo educacional disponibilizado no jogo, onde se faz necessário que o desenvolvedor aprenda todo o contexto do conteúdo utilizado no jogo para somente depois começar o desenvolvimento. De acordo com Hwang (2016), caso o desafio educacional não esteja alinhado a proposta do jogo, os jogadores tendem a procrastinar a realização dos mesmos, suprimindo o conteúdo educacional presente no jogo. Para que isso não venha a acontecer, é necessário que exista a compreensão da melhor forma de aprender determinado conteúdo educacional junto a uma condição de desafio de jogo que traga prazer ao jogador.

O tempo consumido no processo de aprendizado de desenvolvimento de jogos pelo docente e por game designers iniciantes pode ser minimizado com a aplicação de uma ferramenta de simples utilização, e com a possibilidade de que o mesmo possa criar os jogos sérios ubíquos, abstraindo os conceitos técnicos de programação e banco de dados, mantendo o foco no conteúdo educacional do jogo. Partindo desse pressuposto, se faz necessária a reflexão de como pode ser otimizado o

desenvolvimento de jogos sérios ubíquos por alunos de design de jogos digitais iniciantes e professores?

A definição de um *framework*, dada por Costa Segundo (2011) é de uma ferramenta que visa aumentar a produtividade de uma equipe de desenvolvimento facilitando a aplicação prática dos conceitos que englobam um jogo sério ubíquo, proporcionando uma visão ampla sobre eles, e definindo uma estrutura flexível a ser seguida. Tal definição representa uma alternativa viável para o problema enfrentado nesta pesquisa.

Esta pesquisa parte da hipótese que um *framework* para a criação de jogos sérios ubíquos pode proporcionar aos alunos de design de jogos digitais e professores, suporte suficiente para o entendimento dos processos que fazem parte da criação de um jogo sério ubíquo, servindo como ferramenta facilitadora, que pode ser adaptada com o conteúdo das diversas áreas de conhecimento.

1.3 OBJETIVOS

Nas subseções seguintes serão descritos o objetivo geral e os objetivos específicos deste trabalho.

1.3.1 Objetivo geral

O principal objetivo dessa pesquisa está no desenvolvimento de um *framework* com o intuito de facilitar a criação de jogos sérios ubíquos através da reutilização de elementos estruturais de *software*, possibilitando que profissionais das diversas áreas relacionadas a educação e alunos de design de jogos iniciais iniciantes produzam os mesmos em seu âmbito de atuação.

1.3.2 Objetivos Específicos

Foram estipulados objetivos específicos para que o objetivo geral fosse atingido, estando eles citados abaixo:

- a) Analisar a bibliografia existente a fim de verificar pesquisas que relatam o desenvolvimento de jogos sérios ubíquos, pervasivos e adaptativos;
- b) Identificar os frameworks existentes e realizar uma comparação entre eles e o FJSU;

- c) Investigar os requisitos necessários para estruturar o *framework*, de forma que ele possa ser utilizado tanto por alunos de design de jogos digitais iniciantes como por professores;
- e) Criar a estrutura que compõe o *framework*, observando os aspectos ubíquos e adaptação do conteúdo educacional;
- f) Avaliar o *framework* através de uma prova de conceito e aplicação com desenvolvedores externos.

2 JOGOS SÉRIOS UBÍQUOS

De acordo com Klopfer (2012) os jogos ubíquos adquiriram grande notoriedade devido à popularidade dos dispositivos móveis, como *smartphones*, *tablets* e *notebooks*, junto com a possibilidade da utilização e conexão em qualquer lugar. Com isso, novas possibilidades surgiram para diferentes contextos, como educação e saúde, permitindo o envolvimento de dados geográficos, reconhecimento gestual em jogos que envolvem tanto o entretenimento como o objetivo principal que o jogo sério busca atingir. Os mesmos trazem o aspecto de mobilidade somado a adaptação de conteúdo automática para o usuário, que torna a interação humano-computador invisível, que segundo Krumm (2016), é aspecto fundamental da computação ubíqua.

Os jogos ubíquos, também chamados de *ubigames* por Buzeto (2015), são aqueles que utilizam de elementos do plano virtual e do mundo real com o objetivo de tornar a experiência mais imersiva, aumentando o engajamento do jogador. O autor diferencia as aplicações ubíquas dos *ubigames*, onde as primeiras estão relacionadas a transparência dos processos computacionais para o usuário, mantendo o foco dele no conteúdo, já os segundos mantêm sua preocupação com as metáforas do jogo e o universo lúdico proposto. De outra forma, Rodrigues (2016) utiliza os termos ubíquos e pervasivos como ambíguos quando relacionados a jogos, onde representam a interação entre o meio real e virtual de forma relevante e herdando as características dos sistemas ubíquos.

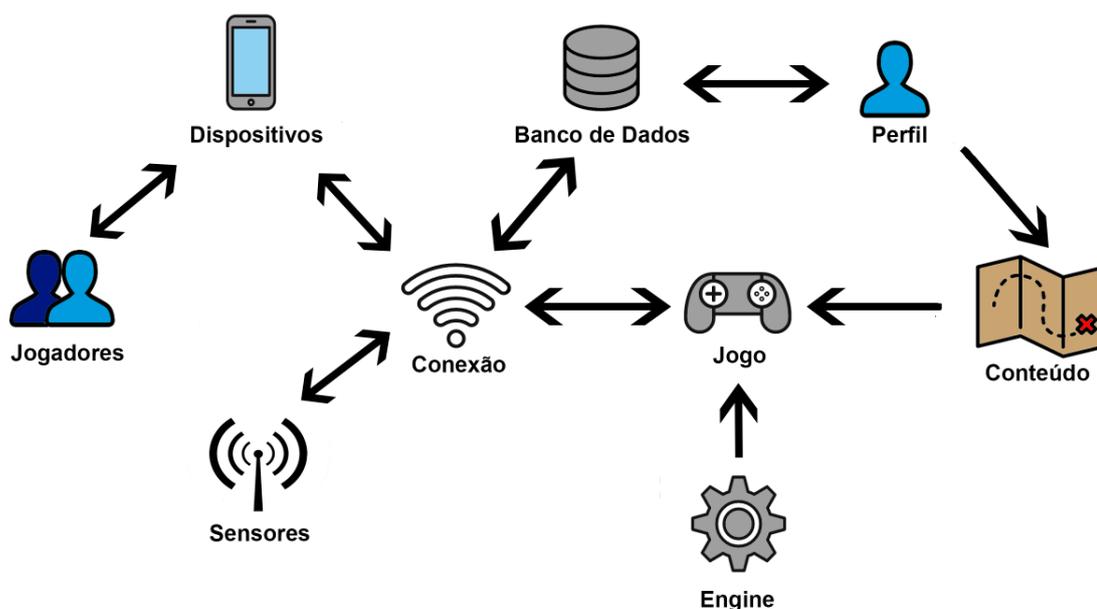
Segundo Rodrigues (2016), existem algumas variações de jogos sérios que podem ser confundidos com os jogos sérios ubíquos, estando entre eles os pervasivos. De acordo com Kittl et al. (2009), os jogos pervasivos são aqueles onde os processos computacionais que envolvem a aplicação prática dele estejam invisíveis aos olhos do utilizador, permitindo que ele não os perceba, possibilitando a imersão no ambiente pretendido.

Em outro contexto, McGonical (2006) cita os jogos pervasivos como aqueles que utilizam de intervenções no espaço social com uma proposta de jogo, unindo a perspectiva do ambiente físico as tecnologias de comunicação. Já aqueles considerados ubíquos utilizam de dispositivos de *hardware* heterogêneos, muitas vezes criados ou modificados apenas para o funcionamento do jogo, trazendo a computação para os mesmos ambientes físicos e transformando-os em uma mescla entre o real e o virtual.

Na pesquisa realizada neste trabalho, não foi encontrada uma definição clara da diferença entre jogos ubíquos e pervasivos, visto que existiu uma correspondência muito grande entre os conceitos que compõem um e outro. Portanto, para esta pesquisa, a interpretação dos mesmos seguirá como a de Rodrigues (2016), onde os termos são tratados como ambíguos, desde que possuam tanto os aspectos de conectividade heterogênea e adaptação de conteúdo.

O levantamento dos dados durante a etapa de fundamentação teórica, e a análise dos mesmos, resultaram no esquema que pode ser visualizado na figura 2.1, que forma o ciclo de vida dos jogos sérios ubíquos, sendo ele uma representação teórica de jogos deste tipo, não estando vinculado a um tipo de ferramenta ou *framework*.

Figura 2.1 - Elementos de um Jogo Sériu Ubíquo



Fonte: (O AUTOR).

Como pode ser visto, a *engine* do jogo oferecerá suporte para o seu desenvolvimento, tanto para a programação de elementos audiovisuais como para o código fonte, sendo o ponto inicial a ser verificado pelo desenvolvedor no momento de criar o jogo. Toda a interface que o desenvolvedor irá usar para a criação do jogo ficará a cargo da *engine*.

O elemento conexão representa todo o meio que é realizado a troca de

informações entre o jogo, sensores, dispositivos e o banco onde são armazenados os dados, como visto nos trabalhos de Buzeto (2015) e Carvalho (2015), envolvendo em sua essência tecnologias de rede sem fio, como WI-FI, 3G, *Bluetooth*, que procedem com a sincronização dos dados conforme a ocasião.

Uma das características dos jogos ubíquos é a utilização de sensores, como visto nos trabalhos de Al Osman (2016), Gicquel (2013), Brandt (2016), Duarte (2014) e Postolache (2015) que muitas vezes não estão ligados diretamente ao dispositivo, como sensores de proximidade, cartões de realidade aumentada e *chips* GSM, sensores de localização, que tanto podem enviar ou receber informações dos jogadores, através de seus dispositivos, enquanto interagem com o jogo.

As informações relativas ao contexto, que são obtidas tanto dos sensores, quanto dos dispositivos, jogadores e do próprio jogo, são salvas no banco de dados, onde a partir do tratamento dos dados, é criado o perfil do usuário, que armazena as principais preferências dele sem que o mesmo tenha consciência. O perfil do usuário representa os dados obtidos por determinadas ações do jogador com os elementos do jogo, que formalizam as adaptações em tempo real do conteúdo do jogo, embasadas no próprio perfil do jogador. Tais conceitos são vistos nos trabalhos de Vayanou (2014), Yahya & Noor (2015), Yang (2015) e Arnold (2013).

Após isso, é feita a adaptação do conteúdo com base no perfil do usuário, como pode ser visto nos trabalhos de Sajjadi (2014) e Silva (2015), sendo relativo a narrativa, dificuldade, jogabilidade, objetivos e tempo, essa adaptação dá seguimento ao processo do ciclo do jogo, que envia as informações de volta ao dispositivo do jogador e espera a próxima interação.

2.1 CONECTIVIDADE

A definição de Klopfer (2012) aborda que os jogos ubíquos devem funcionar nos dispositivos móveis comumente usados, como *smartphones* e *tablets*, não limitando-se a modelos específicos, de forma a abranger a maioria dos dispositivos e sistemas operacionais atualmente utilizados. Da mesma forma, eles devem utilizar os dados obtidos pela interação do jogador, analisando suas atitudes e padrões usados durante o jogo, a fim de adaptar a experiência conforme o contexto observado.

Segundo Buzeto (2015), pressupondo o desenvolvimento de jogos ubíquos pelo ponto de vista técnico, existe uma série de desafios a serem alcançados, sendo

eles heterogeneidade, mobilidade, integração espontânea e sensibilidade a contexto, além dos próprios componentes de jogos.

Entende-se por sensibilidade a contexto a capacidade de observar e compreender os recursos do ambiente e as interfaces de acesso aos mesmos. Heterogeneidade diz respeito a permitir a integração de diversos dispositivos que utilizem plataformas distintas de *hardware* e *software* além de redes de comunicação diferentes. Permitir que novos tipos de dispositivos e recursos sejam acessíveis, em tempo real e sem a necessidade de intervenção, faz parte do que se entende por integração espontânea. Dispositivos podem entrar e sair do ambiente de forma inesperada, como parte do que se espera do desafio de mobilidade. Por fim, fornecer componentes de jogo que controlam o ciclo de vida do jogo e fornece componentes reusáveis, simplificando seu desenvolvimento (BUZETO, 2015, pg. 6).

Para que os desafios citados sejam alcançados, é preciso que os elementos ligados ao desenvolvimento estejam em concordância com o objetivo proposto pelo jogo, portanto, a arquitetura utilizada deve envolver os dispositivos e sensores usados na captura dos dados e também definir como deve ser feito o tratamento dos mesmos.

No contexto deste trabalho, são entendidos como sensores, as tecnologias internas e externas aos dispositivos móveis, que trabalham com o monitoramento de atividade, tanto do jogador como do ambiente. Esse conceito foi definido seguindo o descrito por Banos (2014), onde os sensores obtêm os dados físicos e os transformam em um tipo de informação entendível pela máquina, no caso, os dispositivos móveis. O autor complementa que a mobilidade é um aspecto que torna um dispositivo perfeito para a coleta de dados de sensores, visto que permite a inclusão de processamento em nuvem, onde aparelhos menores e de baixo consumo de energia podem coletar os dados, e deixar o processamento para servidores físicos, além de garantir o monitoramento em ambientes onde é necessário uma maior mobilidade, como um espaço ao ar livre.

2.2 ADAPTAÇÃO DE CONTEÚDO

Além da adaptação de conectividade, no viés deste trabalho, os jogos sérios ubíquos consideram também a adaptação de conteúdo, sendo relativa ao contexto do jogador, possibilitando que o jogo seja moldado às suas necessidades e particularidades, determinando o seu perfil de jogador. Dentro deste processo, o jogo entenderia as necessidades do jogador, com relação a dificuldade e apresentação do

conteúdo educacional, tornando o acessível para jogadores pouco experientes ou atrativo para aqueles com maior conhecimento, otimizando o aprendizado.

Lopes e Bidarra (2011) citam, que mesmo em jogos comerciais, existe a preocupação de adaptar mecânicas e conteúdo do jogo de forma a oferecer a melhor experiência para o jogador.

O perfil do jogador irá armazenar as principais informações sobre o seu comportamento, que são usadas para a adaptação do conteúdo. Segundo Lopes (2017), são armazenados conceitos abstratos, e através dessas informações, podem ser definidas regras de adaptação através da linguagem de programação usada no desenvolvimento do jogo.

Como também citado por Carvalho (2015), no contexto do aprendizado, a adaptação de conteúdo é relativa à modificação do conteúdo educacional conforme a interação e a performance do jogador com os elementos que compõem o jogo, proporcionando o nível adequado de desafio a ele. Seguindo o mesmo princípio, Silva (2015) cita a importância do equilíbrio do fator dificuldade em um jogo, para evitar que jogadores com maior habilidade não se sintam entediados jogando, e também que aqueles que sentem dificuldades não venham a se frustrar. A autora também cita que enquanto alguns jogadores terminam o jogo com facilidade, outros não conseguem concluir a primeira fase.

Os trabalhos de Lopes (2017) e Silva (2015) são exemplos de formas de adaptação de conteúdo. O primeiro sendo relacionado à criação de mapas baseada no perfil do jogador, redefinindo plataformas, inimigos, e elementos que compõem o jogo, tudo sendo realizado através de algoritmos que capturam a habilidade do jogador, e adaptam a dificuldade do mapa. Já o segundo define um método de adaptação de dificuldade em tempo real, modificando a Inteligência Artificial (IA) dos inimigos do jogo *Defense of The Ancient* (DOTA).

Englobando os conceitos de conectividade e adaptação de conteúdo, podem ser citados como exemplos de modelos de arquitetura de jogos ubíquos, usadas por outros autores, os jogos sensíveis a localização, os *exergames*, jogos para reabilitação e os adaptativos.

2.3 LOCATION BASED GAMES

Os jogos sensíveis a localização ou *location based games* são exemplos de

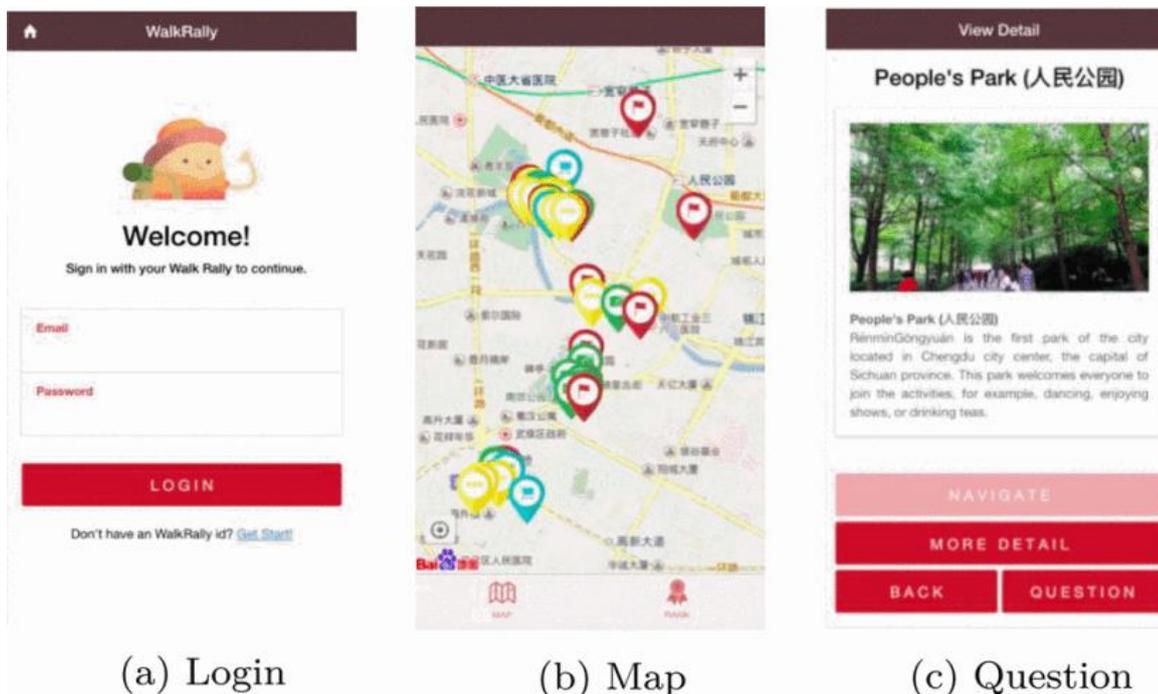
jogos que utilizam recursos relacionados ao contexto físico de onde o jogador se encontra. A conexão de rede sem fios permite que atividades ao ar livre possam ser realizadas nos jogos, e partindo desse princípio, os dispositivos móveis representam a principal plataforma utilizada para esse tipo de jogo. É importante que os dispositivos possuam sensores que permitem que o contexto de localização seja adquirido, como o Sistema de Posicionamento Global (GPS), além disso, outros artifícios podem ser utilizados, como a câmera do dispositivo, que pode efetuar a leitura de QR (*Quick Response*) Codes e outros tipos de códigos (HOLM, 2014).

O trabalho de Lumpoon (2016) traz um exemplo de jogo sério sensível a localização para o incentivo de turistas a conhecer a cultura da cidade de Chengdu, chamado *Walk Rally*. O jogo utiliza o princípio de caça ao tesouro, onde o jogador deve realizar o percurso pré-determinado pelo criador do jogo, passando pelos pontos de interesse representados no mapa e respondendo perguntas relativas àquele lugar. O sensor de GPS presente nos *smartphones* é responsável por determinar a localização do jogador neste caso.

O autor cita que realizou testes com turistas que estavam visitando o local, mantendo o percurso já programado e definindo objetivos. Os mesmos foram divididos em grupo, e realizaram um processo semelhante a uma maratona, onde partiram do ponto inicial ao mesmo tempo, e deveriam cumprir os objetivos propostos, promovendo a competição de qual grupo chegaria primeiro no ponto final.

Foram presenciadas reações positivas quanto aos jogadores na realização deste tipo de tarefa, aumentando a imersão da experiência no ambiente turístico e no conteúdo histórico abordado. Na figura 2.2 podem ser visualizadas as telas que representam o jogo *Walk Rally*.

Figura 2.2 - *Walk Rally* - Jogo sensível a localização



(a) Login

(b) Map

(c) Question

Fonte: (LUMPOON, 2016).

O FJSU traz em sua estrutura elementos que podem ser utilizados na criação de um *Location Based Game*, como a captura de dados de latitude e longitude da posição real do jogador, obtidos através do GPS, e também o suporte a imagem do mapa do posicionamento, através de uma integração com o Google Maps¹. Dessa forma, podem ser criados jogos sérios com intuito educacional, utilizando elementos de um *quiz game*, como um sistema de perguntas e respostas referentes ao conteúdo escolhido para ser abordado, enquanto o mapa real é utilizado como subsídio para a execução da atividade, podendo o conteúdo estar relacionado com aquela localidade em questão ou não.

Essa proposta pode ser interessante para professores que querem levar o conteúdo para fora do ambiente de sala de aula, os recursos ubíquos permitem que não só localidades históricas possam ser exploradas, mas também o conhecimento geográfico da região a partir dos dados de latitude e longitude.

2.4 EXERGAMES

Os *exergames*, de acordo com Whitehead (2010), são aqueles jogos que tem o objetivo de estimular alguma atividade física, de forma a motivar o jogador a chegar a um objetivo específico, como por exemplo, realizar uma corrida de 1 km ou efetuar

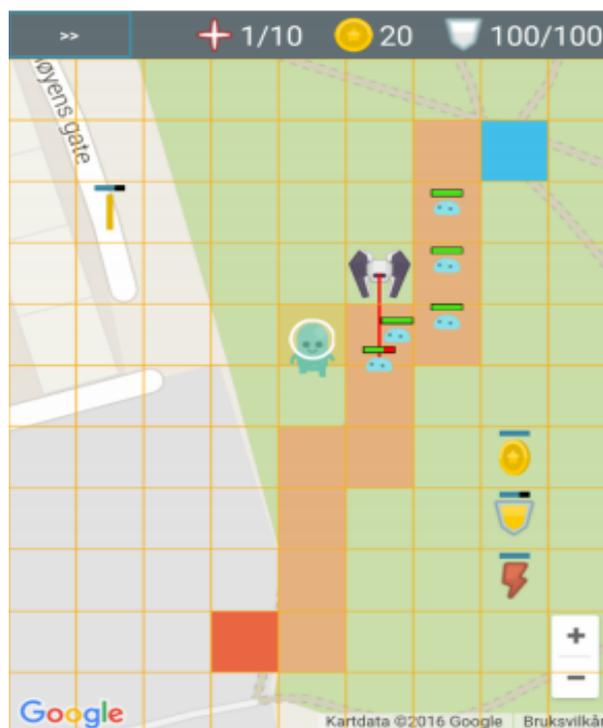
¹ <https://developers.google.com/maps/documentation/?hl=pt-br>

um movimento corporal corretamente. Os jogos também podem estar relacionados a mudança de comportamento, no caso o estímulo de uma vida mais saudável através de exercícios, controle da alimentação e do sono. Para a aplicação dos mesmos, são usados desde o GPS para o controle da localização, o pedômetro para medir a quantidade de passos realizados, e dispositivos como o Wii² e o *Kinect*³ para verificar se um movimento está sendo realizado da forma correta.

O trabalho de Brandt (2016) relata o desenvolvimento de um *exergame* no formato *tower defense*, em que o jogador deve deter uma onda de inimigos através da construção de torres em locais estratégicos do mapa. No caso do jogo desenvolvido, o jogador é representado por um avatar, movimentado usando a posição adquirida pelo GPS, e, deve coletar moedas espalhadas pelo mapa, além de chegar até a posição onde deseja construir uma torre.

Na figura 2.3 pode ser visualizado o jogo de *tower defense* desenvolvido, onde pode ser visto o mapa, o personagem que mostra a localização real do jogador e os inimigos.

Figura 2.3 - Tower Defense Exergame



Fonte: (BRANDT, 2016).

² <https://www.nintendo.co.uk/Support/Parents/Hardware/Wii/What-is-Wii-/What-is-Wii--920239.html>

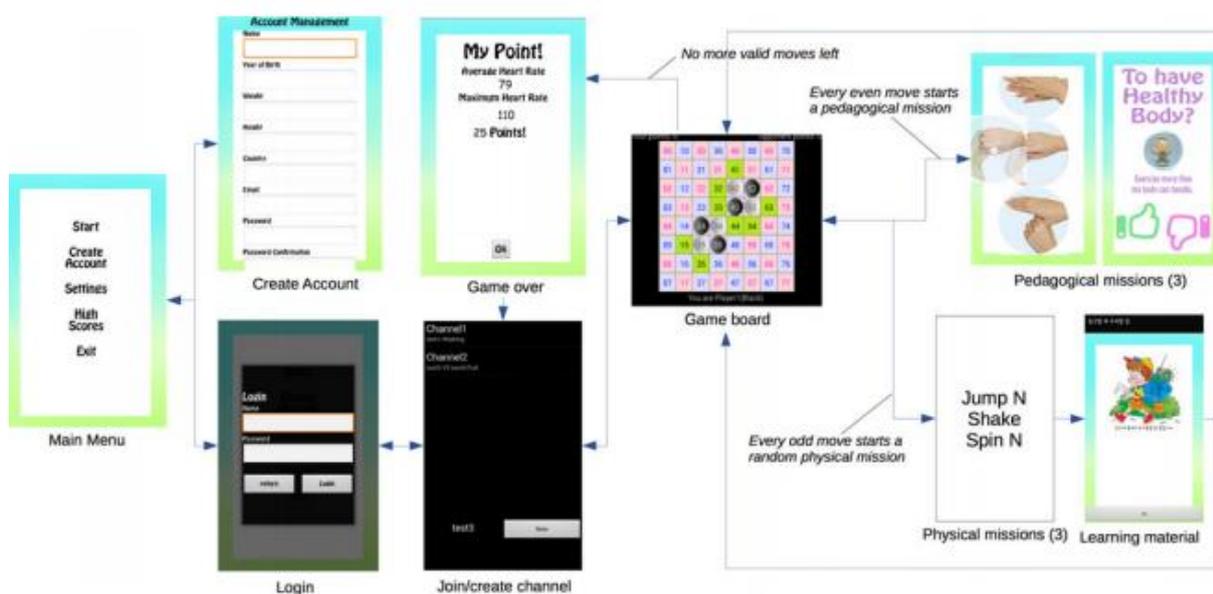
³ <https://www.xbox.com/en-US/xbox-one/accessories/kinect>

De forma a unir a prática de movimentação física, com educação, a proposta de Lindberg (2016) traz desafios pedagógicos em um exergame denominado Othello 2 (RO2), que explora o uso de smartphones e sensores vestíveis pelos alunos nas atividades realizadas por professores de educação física.

A proposta do autor está em transformar o jogo de tabuleiro *Othello*, em um exergame, através das tecnologias ubíquas. O jogador deve movimentar-se em um tabuleiro de sensores NFC, utilizando o *smartphone* como leitor dos mesmos, gerando uma atividade física a ser realizada pelo jogador, como pular ou girar um determinado número de vezes, ou sacudir o *smartphone*.

Após a realização da atividade física, é mostrado uma questão pedagógica relacionada ao currículo dos estudantes, no caso, educação física, sendo relacionadas aos conteúdos de saúde, prática esportiva e primeiros socorros. Com a resolução de ambas as atividades, o jogador acumula o ponto daquela ação. Esse processo pode ser visto na figura 2.4.

Figura 2.4 – Processo de jogo do Othello 2



Fonte: (LINDBERG, 2016).

A avaliação do jogo foi realizada por estudantes da terceira série da Coreia do Sul, e obteve resultados positivos, sendo o efeito do aprendizado mais notável se

comparado ao método tradicional de ensino, no entanto, segundo o autor, o efeito do aprendizado a longo prazo ainda deve ser estudado.

Os *Exergames* utilizam os sensores dos *Location Based Games* e também sensores de movimento, podendo os elementos do FJSU citados anteriormente, serem utilizados para o desenvolvimento de um jogo sério ubíquo neste estilo, englobando os elementos de gamificação, como moedas, pontuação e desafio, junto ao exercício físico.

2.5 JOGOS PARA REABILITAÇÃO

Os jogos para reabilitação, de acordo com Postolache (2015), são desenvolvidos com o propósito de auxiliar profissionais da fisioterapia e áreas afins, motivando os pacientes a realizarem os procedimentos do tratamento físico de membros, seja por movimentos ou gestos, controlando e registrando a execução dos mesmos e apresentando ao profissional uma linha do tempo da evolução do paciente e os resultados obtidos. O autor desenvolveu alguns exemplos em seu trabalho, utilizando o sensor *Kinect* para o controle de movimentos. O jogo *Apple Harvesting* foi criado com o intuito de ajudar no tratamento dos membros superiores, onde o jogador deve simular a coleta de maçãs, acumulando pontos, onde o tempo para a coleta e a dificuldade do jogo podem ser definidas pelo fisioterapeuta através de um aplicativo para dispositivos móveis.

Pode ser visto na figura 2.4 uma captura de tela do jogo sendo executado, onde o personagem do jogo é movimentado conforme o jogador, situado no canto inferior direito da imagem.

Figura 2.4 - *Apple Harvesting Game* - Jogo para reabilitação



Fonte: (POSTOLACHE, 2015)

Dentre os possíveis elementos do FJSU para a criação de um jogo para a reabilitação, está no reconhecimento de movimento utilizando o sensor de giroscópio do dispositivo, e a adaptação de conteúdo conforme o perfil do jogador. Dessa forma, pode ser criado um jogo que interprete se o movimento realizado pelo jogador, podendo ser um passo de dança ou gesto, está correto, e, a partir de uma constante análise de repetições, verificar o nível de aprendizado do mesmo.

2.6 JOGOS ADAPTATIVOS

Os jogos adaptativos, segundo Oliveira (2016), ajustam seu conteúdo em tempo real conforme a interação do jogador, modificando elementos do jogo conforme o fluxo do jogador, como as fases, inimigos e desafios, além de modificar a dificuldade, podendo facilitar ou dificultar, fazendo com que a experiência seja personalizada conforme a necessidade do jogador.

Segundo Carvalho (2015), a adaptação de conteúdo para o aprendizado em um jogo é diferente quando comparada a um software educacional tradicional, visto que os jogos buscam uma interação contínua do jogador com o jogo, de forma que dados e evidências produzidas pelo jogador durante a experiência, como derrotas, interações com NPC's (personagens não jogáveis) e tempo para realizar uma ação, possam ser capturadas a fim de melhorar a sua experiência. No entanto, essa adaptação deve ser realizada de forma que não interfira no fluxo natural do jogo,

fazendo com que o jogador não desprenda sua atenção, e perceba que seus dados estão sendo capturados.

Sawyer (2017) cita que o afeto é um quesito importante a ser levado em conta quanto ao desenvolvimento de ambientes de aprendizagem adaptativos, influenciando no impacto motivacional que a atividade do jogo vai causar no jogador. No seu trabalho, o autor analisou os aspectos emocionais do jogador, gravando suas expressões faciais enquanto jogava e criando modelos de adaptação de conteúdo do jogo *Crystal Island*. Segundo o autor, o modelo que utilização essa predição de conteúdo conforme o estado emocional do jogador influenciou na motivação do jogador.

O autor Soflano (2015) criou um jogo sério adaptativo com o objetivo de ensinar SQL com base no estilo de aprendizagem dos alunos. O mesmo foi desenvolvido em três módulos, um deles sem nenhum tipo de adaptação, o outro com adaptação de conteúdo relativo ao estilo de aprendizagem do jogador (por texto ou imagem) baseado em questionário previamente respondido, e o terceiro com adaptação de conteúdo em tempo real conforme a atividade do jogador enquanto interage com os elementos do jogo.

Na figura 2.5 é visto uma captura de tela de um momento da narrativa do jogo, onde as questões respondidas pelo jogador modificam a forma que o conteúdo do jogo é apresentado.

Os recursos que o FJSU possui para o desenvolvimento de jogos adaptativos estão em relação ao conteúdo, podendo classificá-lo conforme o nível de expertise do jogador ao desenvolver uma atividade do jogo. Conforme o número de falhas, o jogo tende a se tornar mais fácil, da mesma forma que um número sequencial de acertos aumenta o desafio.

Figura 2.5 - Jogo adaptativo para o aprendizado de SQL



Fonte: (SOFLANO, 2015).

2.7 DESENVOLVIMENTO DE JOGOS UBÍQUOS

O desafio de desenvolver um jogo ubíquo envolve a escolha de uma ferramenta que englobe os aspectos que foram considerados necessários para a criação do jogo. Para este fim, pode ser usado uma *engine*, que de acordo com Gregory (2009) é como uma seleção de componentes e mecânicas reusáveis que fazem parte de um gênero específico de jogo, e permitem a criação de novos jogos. Essa definição foi expandida de forma que as *engines* atuais acabam por permitir que jogos com uma ampla variedade de gêneros possam ser criados.

Tanto uma *engine* como um *framework* são considerados facilitadores de desenvolvimento de jogos, no entanto, eles se diferem por um *framework* ser formado por uma coleção de códigos de programação ou conceitos, e uma *engine* ser uma ferramenta mais complexa, que comumente possui interface gráfica e que permite uma interação mais intuitiva com o desenvolvimento. Muitas vezes não é necessário escrever em uma linguagem de programação específica usando uma *engine*, a própria interface gráfica e funções pré-programadas quando estruturadas permitem o desenvolvimento de jogos (GREGORY, 2009).

O trabalho de Buzeto (2015) traz a construção da *engine Uimpala*, que pode ser vista na Figura 2.6, e que entrega ao desenvolvedor uma arquitetura pronta para jogos ubíquos, que permite a integração de outras plataformas de desenvolvimento e disponibiliza uma série de componentes prontos. Segundo o autor, a tarefa de

controlar processos de entrada e saída em uma *engine* de jogos ubíquos se torna mais complexa, visto que os dispositivos que fazem parte do processo podem surgir em qualquer momento da execução do jogo, e não estão definidos de forma premeditada como em um jogo comum.

Figura 2.6 - Arquitetura da *engine Uimpala*



Fonte: (BUZETO, 2015).

2.7.1 Unity

Além das *engines* específicas para jogos ubíquos, também podem ser utilizadas aquelas que permitem a extensão de seus recursos através de *plugins* e bibliotecas. Dentre essas *engines*, a *Unity*⁴ pode ser estendida de forma a permitir que jogos ubíquos possam ser produzidos através dela. Além disso, através do *Asset Store*⁵, que é um repositório de *plugins* e recursos áudio visuais, a produtora da *Unity* permite que a comunidade possa colaborar e produzir conteúdo que venha a contribuir com o projeto de outros desenvolvedores.

A *Unity* é uma *engine* para desenvolvimento de jogos digitais, criado pela Unity Technologies, que oferece além do suporte a criação de jogos 3D, também possibilita desenvolver jogos 2D. Os principais pontos positivos da *engine* estão na

⁴ <https://unity3d.com/pt>

⁵ <https://www.assetstore.unity3d.com/en/>

disponibilização multiplataforma dos jogos criados, vasta documentação existente, mesmo para as versões mais novas da engine, e também a possibilidade de extensão da mesma, podendo ser adaptada a necessidade do desenvolvedor. Bartneck (2015) lista alguns recursos da Unity, que foram utilizados em seu trabalho, mas que competem a este também, sendo eles citados a seguir:

- a) Uma interface gráfica para fácil manipulação de objetos e controle de interações
- b) A possibilidade de comunicar-se com *hardware* externo;
- c) A possibilidade de processar dados multimídia;
- d) Ser disponibilizado gratuitamente em sua totalidade para uso de pesquisa e educação;
- e) Oferecer suporte para diferentes sistemas operacionais.

Além dos recursos citados, o autor cita que a Unity foi escolhida para seu trabalho devido a comunidade de desenvolvedores ser bastante ativa, tanto nos fóruns oficiais da plataforma como nas redes sociais. Além disso, oferecendo uma extensibilidade através da criação de plugins, que atualizam não só a sua interface, mas também os recursos e a comunicação com o hardware externo, facilitando o trabalho do desenvolvedor.

Pensando em aliar as características da engine Unity com a proposta deste trabalho, a engine foi escolhida para servir como base do *framework* FJSU, onde foi verificado que os recursos de extensibilidade e funções permitem simplificar a criação de jogos sérios ubíquos.

2.8 FRAMEWORKS PARA JOGOS UBÍQUOS

A utilização de um *framework* no domínio da engenharia de *software* fornece ao desenvolvedor uma estrutura modular específica para um contexto, que permite o reuso de componentes e o acoplamento de novas tecnologias, gerando uma nova aplicação (BRAGA, 2002). No mesmo sentido, Johnson & Foote (1988) tratam um *framework* como um conjunto de classes que possuem soluções relacionadas a problemas de um mesmo contexto, possibilitando a reutilização de componentes em um nível alto de granularidade. Já Fayad (1999) cita o *framework* como um programa

semipronto, criado por uma organização ou para um domínio específico, e a partir dele podem ser desenvolvidos outros programas customizados conforme a necessidade, facilitando o reuso.

Os componentes reusáveis de um *framework* para jogos ubíquos podem ser desde uma estrutura de classes, a implementação de uma classe contendo métodos reusáveis por herança, ou até mesmo a instanciação de um objeto completo do jogo. Esse objeto instanciado é tratado pelas *engines* como *prefab*, que segundo Xie (2012), são elementos pré-fabricados, formados por elementos gráficos e audiovisuais, além da implementação de scripts que trazem a interação com os demais elementos. Um exemplo de *prefab* pode ser um inimigo, que possui características visuais próprias, e códigos, que permitem suas interações e movimentos com o jogador e demais objetos do jogo.

De acordo com Kubo (2006) “um *framework* para jogos não é um conjunto de funções, ou uma API, relativas a um determinado objetivo. Ele também não é um produto pronto.” Portanto, a ideia de utilizar um *framework* está na possibilidade de um desenvolvedor não se deter aos aspectos que seguem um padrão, e simplesmente desenvolver o *design* do jogo.

Para que o reuso do *software* seja viável, é preciso que o *framework* tenha a característica de flexibilidade, sendo ela trazida pelos pontos variáveis da estrutura, também chamados de *hot spots*. Estas alternativas que trazem a flexibilidade ao *framework* desenvolvido representam padrões que podem ser substituídos por outros, ou que são opcionais, fazendo com que o desenvolvedor possa definir aquela estrutura que mais cabe ao seu jogo. Os *hot spots* são genéricos, podendo ser adaptados conforme a necessidade da aplicação, seu uso não é necessário em todas as aplicações, já os pontos fixos, ou *frozen spots*, definem os componentes básicos da arquitetura geral de um sistema e suas ligações, que são imutáveis em qualquer aplicação desenvolvida a partir do *framework* (BRAGA, 2002).

Seguindo os conceitos de Taligent (1994), os *frameworks* podem ser classificados em três tipos quanto a dimensão da sua utilização, sendo eles de aplicação, suporte e domínio.

- *Frameworks* de Aplicação: São aqueles que englobam as competências necessárias para serem aplicadas a uma grande variedade de tipos de programa, não limitando-se a um modelo específico, no entanto, resolve

apenas uma fatia desse problema. Os *frameworks* para desenvolvimento de interface gráfica são considerados de aplicação.

- *Frameworks* de Suporte: Irão prover os serviços de sistema operacional, como a gerência de drivers, acesso de arquivos e suporte à computação distribuída.
- *Frameworks* de Domínio: Aqueles que são relacionados a um domínio específico, englobam boa parte das soluções necessárias para criar um programa completo relacionado a uma área específica, como um framework para aplicações multimídia.

Partindo dos princípios apresentados, o produto desenvolvido neste estudo está classificado como um *framework* de domínio, visto que está relacionado a um domínio específico, no caso o desenvolvimento de jogos sérios ubíquos, e busca abranger boa parte das soluções necessárias para o desenvolvimento de uma aplicação desse tipo.

2.8.1 Framework de Domínio

Segundo Braga (2002), para o desenvolvimento do *framework* de domínio, é necessário que seja buscado algum modelo que represente o domínio alvo e consiga capturar as principais funcionalidades dele, podendo utilizar da análise de domínio, engenharia reversa ou análise a partir da experiência prática com aquele domínio específico.

No caso da proposta desta dissertação, a análise do domínio foi realizada com base na descrição de Arango (1994), onde é preciso analisar os aspectos que fazem parte de uma família de sistemas, e não apenas a um individual, dessa forma, possibilitando o reuso. O autor cita que para uma análise de domínio efetiva, é preciso a análise dos sistemas existentes dentro do domínio observado, além da entrevista com profissionais deste domínio, capturando o conhecimento suficiente relacionado aos aspectos do mesmo.

Nesse processo, são analisadas documentação, códigos-fonte, manuais e toda informação relativa ao processo de desenvolvimento de *softwares* do domínio alvo da pesquisa. Esse procedimento acaba se repetindo enquanto o domínio é analisado, refinando o resultado a cada ciclo.

Seguindo uma adaptação do que foi citado por Arango (1994), para a análise de domínio, foi realizado um processo de mapeamento sistemático, buscando os principais conceitos presentes nos jogos sérios ubíquos, e como está sendo feito o desenvolvimento deles, além da análise dos jogos e estruturas propostas pelos autores, tal como os processos que eles utilizaram para alcançar os seus resultados obtidos. Além da classificação específica de domínio, um *framework* também pode ser classificado como de caixa branca, preta ou cinza.

2.8.2 Framework de Caixa branca, preta e cinza

Segundo Markiewicz (2001), os *frameworks* podem ser classificados de três formas, sendo de caixa branca, preta ou cinza. O princípio de um *framework* de caixa branca está em oferecer ao desenvolvedor uma série de classes e componentes reusáveis relacionados a um domínio específico de maior abrangência, que podem ser utilizados através de herança.

Um *framework* de caixa preta oferece ao desenvolvedor uma plataforma mais robusta e intuitiva, chamada de *wizard*, que são *softwares* usados para a instanciação de um protótipo de *software* a partir da seleção dos componentes que farão parte do mesmo, isso torna um *framework* de caixa preta mais fácil de utilizar na visão do usuário.

Comparado com os de caixa preta, os *frameworks* de caixa branca são mais fáceis de desenvolver, já que contam apenas com a criação e estruturação das classes usadas para o domínio escolhido, sem envolver a criação de *wizards*, no entanto, a utilização de um *framework* de caixa branca é mais complicada para o usuário, já que todo o trabalho é feito através de codificação e manipulação de scripts, fazendo com que o conteúdo do *framework* mantenha-se apenas nos métodos e classes. Esse conceito pode ser observado na figura 2.7, que ilustra as diferenças entre os *frameworks* de caixa branca e preta, e coloca o *framework* de caixa cinza como um intermediário entre os dois.

A tendência de um *framework* de caixa branca é evoluir para um de caixa preta conforme o seu desenvolvimento, enquanto o mesmo possui componentes tanto dos *frameworks* de caixa branca e de caixa preta, pode ser considerado de caixa cinza. O *framework* de caixa cinza abrange aspectos presentes nos *frameworks* de caixa branca e preta, sendo um intermediário entre os dois.

Para este trabalho, inicialmente foi criado um *framework* de caixa branca, mas foi definido que o mesmo deveria seguir o formato de caixa cinza, abrangendo a utilização de uma interface de simples utilização pelos desenvolvedores.

Figura 2.7 - Tipos de *Frameworks*



Fonte: (O AUTOR)

2.9 TRABALHOS CORRELATOS

Com a definição de um *framework* para o domínio de jogos ubíquos, foram encontrados estudos relacionados ao tema, que abordam alguns aspectos em comum com o proposto neste trabalho. Devido à falta de bibliografia que abordasse a criação de *frameworks* específicos para jogos sérios ubíquos, foram verificados todos os que se assemelhavam ao FJSU, possuindo ou não o termo “sérios” na sua definição.

2.9.1 *Tardigrade*

A proposta de Rodrigues (2016) está na criação do *Tardigrade*, que é um *framework Android* para o desenvolvimento de jogos de cartas ubíquos, que busca diminuir a dificuldade de desenvolvimento de um jogo desse estilo. Este trabalho busca abranger tanto os jogos que utilizam cartas virtuais como aqueles de realidade mista, usando cartas tangíveis que possuem interação com os dispositivos Android. Foi criado o jogo *Hyper Try* como exemplo de instanciação do *framework*, que ilustra batalhas fantásticas, com as cartas representando titãs da natureza que tem seus poderes baseados em fogo, água e terra, usando das tecnologias ubíquas para

reconhecer as cartas. Na figura 2.8, podem ser visualizadas exemplos das cartas do jogo.

Figura 2.8 - Cartas do jogo *Hyper Try*



Fonte: (RODRIGUES, 2016).

A avaliação do *Tardigrade* foi realizada por um grupo de seis alunos voluntários que possuíam pouco conhecimento em programação para dispositivos Android, onde foi realizado um questionário inicial, referente a experiência dos mesmos, a capacitação deles para usar o *Tardigrade*, o desenvolvimento de atividades práticas e um questionário final quanto a experiência realizada, foi contabilizado o tempo de realização de cada atividade. Em média, a aplicação prática da atividade demorou 24 minutos, e em sua totalidade, a maioria das avaliações foram positivas. O próprio autor toma como limitação a avaliação, visto que o número de participantes pode ser insuficiente para um resultado consistente. Além disso, é citado que o *Framework* ser voltado apenas para *Android* torna-se uma limitação, visto que muitos dispositivos usam outros sistemas operacionais.

2.9.2 uOS

Seguindo a proposta de *frameworks* para jogos ubíquos, Buzeto (2015) toma ele pelo conceito de reconfiguráveis, que diz respeito a adaptação do jogo a configuração do ambiente em tempo real. O autor promove um estudo desde a

concepção da ideia até a construção dos mesmos, onde cria a plataforma *uOS*, que conta com os aspectos práticos necessários para alcançar os desafios, que segundo o autor são: heterogeneidade das tecnologias, mobilidade, sensibilidade ao contexto e integração espontânea dos dispositivos.

A plataforma desenvolvida conta com a utilização da *engine Uimpala*, o *plugin* de comunicação para a *engine Unity 3D*, que possibilita a comunicação entre as duas, além do *execution driver*, que permite que os jogos desenvolvidos possam ter seu formato aberto. Na figura 2.9 pode ser visualizada a estrutura do *uOS*.

Figura 2.9 - Estrutura do *uOS*



Fonte: (BUZETO, 2015).

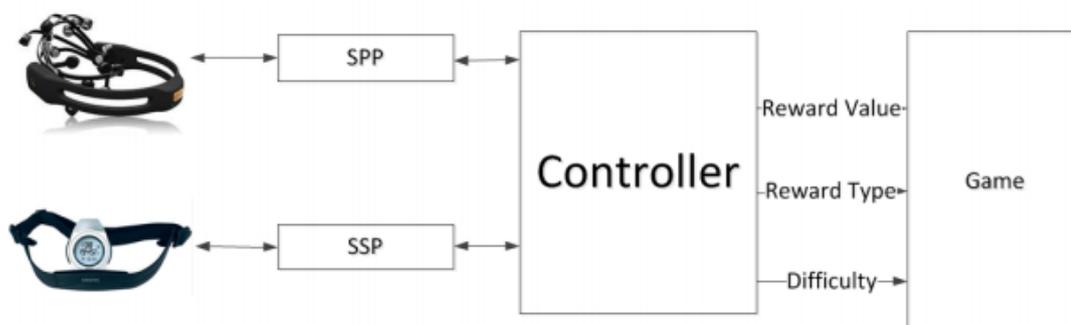
A avaliação do *uOS* contou com o desenvolvimento de dois jogos pelo grupo, o *Run Fast!*, que simula uma corrida de carros em uma tela pública, onde a interação é feita através do *smartphone* dos jogadores, e o *uMoleHunt* que também utiliza da tela pública, e cria um jogo de turnos, onde os jogadores são divididos em dois grupos, e devem descobrir dez nomes, selecionando uma letra por rodada, e revelando se a palavra possui ela. Além disso, foram selecionados 12 alunos para uma capacitação quanto a jogos reconfiguráveis, e que no período de um mês, criaram o projeto de 9 jogos distintos, desse total, cinco foram implementados. O autor cita como possíveis trabalhos futuros a integração entre outros ambientes, permitindo que distâncias

físicas fossem quebradas, a evolução dos elementos que compõem o *uOS*, a integração com outras plataformas de desenvolvimento de forma mais eficiente e a realização de testes mais avançados.

2.9.3 *Framework* para jogos adaptativos

O trabalho de Szegletes e Forstner (2013) traz o desenvolvimento de um *framework* para jogos mobile que usam sensores faciais, que tem como objetivo a adaptação de conteúdo através de sinais produzidos por comportamentos, gestos faciais e emoções do jogador (FJA). O estudo mantém seu foco no sistema de recompensas do jogo, possibilitando que a mesma seja diferente conforme a expectativa do jogador, segundo o autor, o estado cognitivo e emocional dele varia conforme essa expectativa, e pode produzir um maior nível de engajamento na tarefa a ser realizada. Na figura 2.10 pode ser visto a estrutura do *framework* citado.

Figura 2.10 - *Framework* para jogos adaptativos



Fonte: (SZEGLLETES; FORSTNER, 2013).

Como caso de estudo, foram recriados, usando o *framework*, modelos de jogos para crianças com dificuldades educacionais que já mostraram resultados positivos. Segundo o autor, a utilização de jogos que já foram avaliados pode mostrar de forma mais clara se o resultado da adaptação da recompensa otimiza o efeito positivo proporcionado.

2.9.4 *Framework* para exergames de ensino e aprendizagem de dança

Como pode ser visto no trabalho de Grammatikopoulou (2018), os exergames podem ser utilizados em cunho educacional. O autor desenvolveu um framework para a criação de jogos para o ensino e aprendizado de danças, teatro e atividades culturais cujo princípio está no movimento corporal (FEAD).

O modelo desenvolvido é dividido em duas interfaces, a primeira para o design do jogo, onde será inserido o conteúdo e capturado o movimento, e a segunda referente a parte jogável, onde o jogador irá treinar os movimentos propostos pelo jogo, sendo a captura deles feita por um sensor como o *Microsoft Kinect*. Na figura 2.4 pode ser visualizada a representação da segunda interface.

Figura 2.11 – Criação de exergames para ensino e aprendizagem de dança



Fonte: (Grammatikopoulou, 2018)

A avaliação do framework foi realizada em duas etapas, sendo a primeira relativa a testes e validação com profissionais de dança e educação, e pessoas sem o conhecimento abordado, onde foram avaliados se os objetivos educacionais do trabalho estavam sendo atingidos, e questões relacionadas a performance dos sensores. A segunda parte da avaliação estava relacionada a criação dos cenários dos jogos, onde os profissionais da área educacional criaram seus movimentos e conteúdo do jogo.

O processo de avaliação contou com o preenchimento de um questionário relativo à experiência de ambas as etapas. O *framework* foi avaliado positivamente, e

contribui como uma ferramenta que pode vir a facilitar o aprendizado de danças e movimentos artísticos, não substituindo o professor, mas servindo como um suporte, e sendo atrativo para crianças e adolescentes.

2.9.5 Comparação entre os *frameworks*

O estudo de *frameworks* relacionados a jogos sérios ubíquos permitiu que fossem analisados os aspectos presentes neles e também verificar quais possibilidades e necessidades podem ser exploradas dentro desse contexto.

Foi percebido que três dos quatro estudos analisados, o *Tardigrade*, o framework para criação de Exergames e o FJA, detém-se ao desenvolvimento de uma estrutura relativa a um caso específico dentro de um domínio complexo, não permitindo muitas alternativas ao desenvolvedor. O *Framework* para Jogos Sérios Ubíquos (FJSU), desenvolvido no presente estudo, procura abordar os conceitos de jogos sérios ubíquos em uma estrutura flexível e variável, possibilitando ao desenvolvedor escolher os aspectos que fazem parte do seu jogo, e apresentando a ele o suporte conceitual necessário para que o mesmo possa ser implantado.

O *uOS* apresenta uma estrutura de *framework* baseado em ferramentas e bibliotecas criadas pelo autor. O *framework* desenvolvido nesta presente dissertação foi criado para a *engine Unity*, usando os recursos da *unity* para a criação dos objetos reusáveis e a interface de utilização do *framework*, além de oferecer suporte a classes e bibliotecas reusáveis da linguagem C#.

No entanto, enquanto o *uOS*, mantém sua estrutura específica para profissionais ligados ao desenvolvimento de jogos, o FJSU procura atingir profissionais ligados à educação, portanto, busca simplificar a produção do jogos, e abstrair os conceitos ubíquos para o desenvolvedor, sendo que o mesmo deve preocupar-se apenas com as mecânicas do jogo a serem utilizadas, e o conteúdo.

No quadro 2.1 pode ser observada uma comparação com os *frameworks* observados neste capítulo e o FJSU. Pode ser visualizado que quanto ao suporte a dispositivos heterogêneos, apenas o *Tardigrade* não alcança esse aspecto, onde sua estrutura é voltada apenas para dispositivos que utilizam o sistema operacional Android.

Quadro 2.1. Comparação entre os *frameworks*

Conceito	<i>Tardigrade</i>	<i>uOS</i>	FJA	Exergames	FJSU
Suporte para dispositivos heterogêneos		X	X		X
Mobilidade	X	X		X	X
Suporte para a criação de conteúdo adaptativo		X	X	X	X
Adaptação por perfil do usuário			X		X
Adaptação em tempo real		X			X
Estrutura flexível		X			X
Estrutura de classes e bibliotecas reusáveis	X	X	X	X	X
Interface gráfica	X	X	X	X	X
Documentação	X	X	X	X	X

Fonte: (O AUTOR).

Quanto a mobilidade, o FJA é o único que possui fatores limitantes nesse sentido, onde apesar de utilizar um jogo para dispositivos mobile, os sensores faciais acabam por determinar até onde a movimentação do jogador pode ser realizada. Os outros três *frameworks* não possuem limitações nesse sentido.

O Tardigrade é o único que não possui suporte a conteúdo adaptativo, visto que seu foco está no desenvolvimento de jogos de um nicho específico. Já em relação a adaptação do conteúdo ao perfil do jogador, apenas o FJA e o FJSU alcançam esse aspecto, verificando a melhor forma que aquele conteúdo pode ser adaptado ao nível de expertise e conhecimento do jogador.

Em relação a adaptação em tempo real, enquanto o *UOs* adapta sua configuração para atender as necessidades do jogo, o FJSU mantém sua adaptação também quanto ao perfil do usuário, sendo a de ambos frameworks realizados em tempo real, sem a interação do jogador.

Quanto a sua estrutura flexível, o FJA possui o fator limitante dos sensores faciais, o Tardigrade está limitado a jogos de carta, e o framework para exergames produz apenas jogos baseado em movimentos físicos. Os outros *frameworks* analisados podem gerar diferentes tipos de jogos com estruturas diversas. Já em relação a estrutura de classes e bibliotecas reusáveis, interface gráfica, e documentação, todos os *frameworks* atingiram esses aspectos.

3. METODOLOGIA

Para alcançar o objetivo proposto neste trabalho, foi adotada uma metodologia de pesquisa qualitativa e quantitativa de natureza exploratória, que segundo Wazlawick (2009), permite a utilização de estudos de caso como meio de investigação, de forma a buscar evidências quanto ao cumprimento dos aspectos propostos pelo objetivo do estudo.

As etapas que compõem o processo metodológico adotado são descritas a seguir de maneira sequencial, buscando explicá-las de forma detalhada, sendo elas a fundamentação teórica, desenvolvimento do produto e a avaliação do *framework*. As mesmas podem ser vistas na figura 3.1.

Figura 3.1 - Etapas do processo metodológico



Fonte: O autor

3.1 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica do trabalho foi realizada utilizando recursos bibliográficos em um processo de mapeamento sistemático, definido por Kitchenham (2004), que proporcionou uma visão geral sobre o panorama de pesquisa atual de jogos sérios ubíquos além do suporte teórico para o estudo. O resultado desta etapa do estudo foi publicado como da Silveira Júnior e Medina (2017) na revista RENOTE,

em sua segunda edição do ano de 2017.

O processo de mapeamento sistemático foi realizado para obter dados referentes ao atual panorama de pesquisa envolvendo jogos sérios ubíquos dos últimos cinco anos. A partir desse objetivo, foram extraídas informações alusivas a captura e tratamento dos dados relativos ao contexto do usuário, a fim de buscar um estado da arte da literatura a ser utilizada como suporte a criação de um framework para desenvolvimento de jogos sérios ubíquos.

O trabalho teve como objetivo principal “identificar trabalhos relacionados a jogos sérios ubíquos”, e a partir desse objetivo, foram formuladas quatro questões de pesquisa, sendo elas:

Q1: Como está sendo feita a entrada de dados no sistema?

Q2: Como os dados estão sendo tratados pelo sistema?

Q3: Como estão sendo avaliados os jogos?

Q4: Quais ferramentas e *frameworks* estão sendo utilizados no desenvolvimento dos jogos?

Para a pesquisa bibliográfica, foram selecionadas as bibliotecas digitais IEEE *Xplore*, ACM Digital e o Banco de Teses & Dissertações da Capes, onde foi aplicado a *string* de busca, onde ela manteve-se da seguinte forma.

“game” AND (“based learning” OR “educational” OR “learning” OR “teaching” OR “training” OR “serious” OR “instructional”) AND (“ubiquitous” OR “adaptive” OR “pervasive” OR “context aware” OR “context sensitive”)

A *string* foi adaptada, sem modificar sua estrutura, conforme o motor de busca, para que sejam buscados apenas estudos com os termos incluídos no título, resumo e palavras-chave dos artigos. Também foi utilizada uma *string* com a tradução dos termos, a fim de também abranger artigos em português. Além disso, foram incluídos artigos por busca manual dos eventos SBIE, *Ubicomp* e do periódico RENOTE. Foram delimitados apenas estudos com ano de publicação equivalente ou superior ao de 2012.

Dentre os resultados da busca inicial, foram selecionados o total de 598 estudos. Em uma primeira etapa, foi realizada a leitura do título, palavras-chave e resumo de cada um dos artigos selecionados até o momento. Os artigos foram analisados seguindo os critérios de inclusão e exclusão, dessa forma reduzindo o

montante de estudos para 183, excluindo da lista os que se encaixam em pelo menos um critério de exclusão.

Foi realizada a leitura da introdução e conclusão dos artigos selecionados após a análise dos critérios de inclusão e exclusão, permitindo um refinamento mais apurado quanto aos estudos até então selecionados para este trabalho, dada essa etapa, o montante de estudos foi reduzido para 90, sendo este número o total de artigos final da seleção.

Os 90 artigos foram lidos na íntegra, e analisados a partir das questões norteadoras deste trabalho, promovendo uma análise do panorama atual de pesquisas relacionadas a jogos sérios ubíquos. Os resultados compreendidos durante o processo são apresentados na seção 4.

3.2 DESENVOLVIMENTO DO PRODUTO

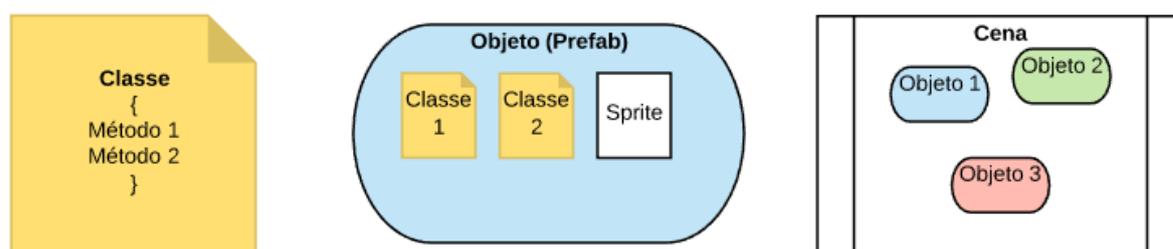
Na etapa de desenvolvimento foi realizada a modelagem e estruturação do *framework* conforme os objetivos do trabalho, e usando como suporte a fundamentação teórica observada. Nesse contexto, tomando como exemplo a arquitetura definida por Buzeto (2015), onde a plataforma desenvolvida engloba a execução de seus módulos sob uma *engine* desenvolvida pelo autor, e utilizando *plugins* para a *engine Unity*

3.2.1 Módulo de aplicação

O *framework* FJSU foi desenvolvido com sua execução sob a *engine Unity*, implementando sua estrutura de classes criada usando a linguagem C#, e usando a para a criação dos objetos que compõem o *framework*. A criação deles foi feita dentro da *engine Unity*, usando suas ferramentas próprias para a criação dos objetos, interface e exemplos disponíveis para o desenvolvedor.

A estrutura do FJSU é formada por classes abstratas, objetos reusáveis (*prefabs*) e cenas completas. Os objetos podem ser reutilizados inserindo-os na cena criada, ou também pode ser importada uma cena completa para posterior edição. A figura 4.7 ilustra a composição dos elementos citados.

Figura 3.2 Elementos reusáveis do FJSU



Fonte: (O AUTOR).

Como visto na imagem, uma classe é um *script* escrito na linguagem de programação C#, que é uma das utilizadas pela *Unity*. Uma classe possui métodos, que podem ser chamados através da invocação da classe em um script do jogo desenvolvido.

Os objetos, também chamados de *prefabs*, são elementos do jogo, podendo ser o jogador, um inimigo, uma parte do cenário, um NPC etc. Um objeto é formado por elementos próprios da *engine Unity*, como física de corpos e detectores de colisão, mas também dá a possibilidade de inserir outros *scripts* acoplados a ele, permitindo estender e reutilizar objetos conforme a necessidade.

As cenas representam a fase de um jogo, onde são incluídos vários *prefabs* com sua programação determinada, que compõem a mecânica do jogo, podendo ser um jogo de plataforma, um *puzzle*, um *quiz game* etc. O FJSU é composto por esses três tipos de elementos.

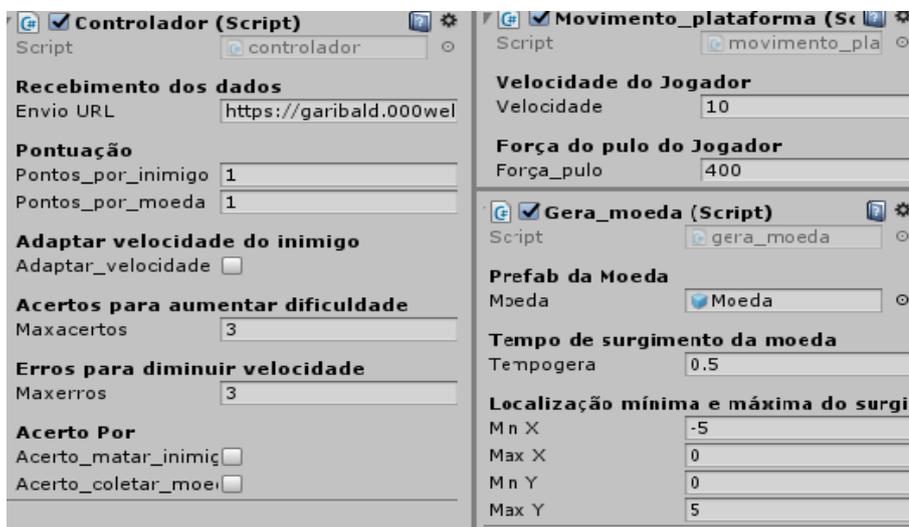
A interface do FJSU foi criada como uma sobreposição da própria interface da *Unity*, sendo esta modificação possível através da utilização da biblioteca *Unity Editor*⁶, que faz parte da estrutura principal da *engine Unity*. Essa modificação pode ser entendida como uma extensão das funcionalidades da *engine*, facilitando o reuso de objetos e funções.

Como etapa da definição da interface do *framework*, foram adicionados os recursos do mesmo a barra de menus do *Unity*, criando o próprio menu do FJSU, que possibilita a adição de novos objetos ao jogo desenvolvido. Também como forma de extensão dos objetos disponíveis, facilitando a interação do desenvolvedor com eles, foi modificado a tela de propriedades dos mesmos, fazendo com que variáveis como velocidade do jogador, força do pulo, quantidade de pontos por cada moeda coletada, número de inimigos inseridos na tela, pudessem ser modificadas sem contato com

⁶ <https://docs.unity3d.com/ScriptReference/Editor.html>

scripts, como pode ser visto na figura 3.2.

Figura 3.3 Propriedades dos objetos



Fonte: (O AUTOR).

3.2.3 Módulo Web

Para o gerenciamento das informações do perfil do usuário, foi criado um módulo web, que realiza a transferência dos dados entre o jogo desenvolvido e um banco de dados armazenado em um servidor externo, sendo um *web service* o responsável por esse intermédio. Para isso, foi definido um sistema de cadastro, onde o jogador cria uma conta básico com suas informações de login e senha, deixando a cargo do desenvolvedor apenas implementar o sistema de login ao jogo.

A transferência das informações do jogo para o servidor de banco de dados ocorre a cada 20 segundos de interação do jogador com o jogo, e permite que aquele estado de adaptação se mantenha em uma nova sessão do jogo.

A escolha de adaptação conforme perfil do jogador pode ser selecionada através das opções de propriedade do objeto controlador, onde é necessário apenas selecionar um *checkbox*, e adicionar os objetos necessários para realizar o login do jogador na cena do jogo. Este procedimento está disponível na documentação do FJSU ⁷.

⁷ <https://garibald.000webhostapp.com/documenta.php>

3.3 AVALIAÇÃO

A avaliação do FJSU foi realizada em três momentos, o primeiro momento enquanto o *framework* mantinha-se em formato de caixa branca, o segundo momento ocorreu no formato de caixa cinza, como uma pré-avaliação, e o terceiro momento foi a avaliação final do FJSU.

O primeiro momento da avaliação foi realizado enquanto o FJSU mantinha sua utilização baseada na instanciação das classes, para que os métodos pudessem ser usados. Durante todo esse primeiro momento, foram utilizados computadores com o sistema operacional Windows 7 e a versão 5.6 da engine Unity, que detinham as seguintes configurações de hardware: Intel Core i5-3570 CPU 3.40 GHz, memória RAM de 8 GB e HD de 1 TB.

Foram selecionados o total de oito alunos do curso de Sistemas de Informação do Campus São Borja do Instituto Federal Farroupilha para atuarem como avaliadores. Para esse momento inicial, visto o desconhecimento do público com a ferramenta Unity, foi definido que os avaliadores precisavam de um conhecimento mínimo da ferramenta, para poder utilizar o FJSU, portanto, foi ofertado um curso abordando o conteúdo necessário para os mesmos.

Após a realização do curso, os alunos seguiram um roteiro de utilização do FJSU, onde deveriam produzir um protótipo de jogo, seguindo os passos descritos. Em sequência, foi pedido o preenchimento de um formulário, abordando a experiência de utilização do *framework*, com questões baseadas na escala Likert (1932) e de resposta livre. Esse formulário pode ser visualizado Apêndice C.

O segundo momento foi realizado quando o *framework* se consolidou no formato de caixa cinza. Esse momento foi pensado como uma pré-avaliação, de forma a verificar em que aspectos o FJSU e a proposta de avaliação poderiam ser melhorados. Para esse momento, foi seguido o mesmo roteiro do primeiro momento de avaliação, no entanto, com a divisão dos avaliadores em dois grupos, aqueles que usaram o FJSU e aqueles que usaram a Unity em seu estado puro.

Da mesma forma que no primeiro momento, nessa etapa também foi aplicado um questionário, no entanto, foi adicionada uma questão para definir de qual grupo o avaliador fazia parte. Após essa aplicação, foi utilizada uma técnica de grupo focal, como descrito Trad (2009), consiste de um método de avaliação qualitativa, baseada na reunião de informações, obtidas através da análise da interação de um grupo

aberto de pessoas, dialogando acerca de um tema específico. A partir da realização do grupo focal, foi obtido dados importantes quanto ao FJSU e a proposta de avaliação.

No terceiro momento de avaliação, foi planejado a execução dos testes em um grupo maior de desenvolvedores, procurando abordar além dos erros obtidos no processo, mas também o desempenho do desenvolvedor enquanto programador de jogos. Para tanto, foi selecionada uma turma do curso de Jogos Digitais da Universidade Franciscana de Santa Maria para a realização da avaliação. Os alunos foram divididos em dois grupos, onde o primeiro iria utilizar o FJSU durante a avaliação, e o segundo usaria apenas as ferramentas disponibilizadas pela *Unity*.

O caso de uso proposto para o teste pode ser visualizado no Apêndice B. O objetivo de utilizar este método de aplicação está em verificar se o *framework* se mantém congruente quanto a reduzir o tempo de desenvolvimento quando comparado a *engine Unity* inalterada.

Os alunos que realizaram a atividade utilizando o FJSU responderam um questionário no final da aplicação, de forma a produzir dados referentes a usabilidade e utilidade do *framework* quanto a sua proposta. Esse questionário foi desenvolvido segundo o estudo de Venkatesh (2008), que propõem o modelo TAM 3, para mensurar a aceitação de tecnologia através de três indicadores, sendo eles, utilidade percebida, facilidade de uso percebida e intenção de uso. O modelo utiliza de afirmativas com escolhas baseadas na escala de Likert (1932), onde o aluno deverá escolher entre cinco alternativas que se delimitam entre concordo plenamente e não concordo quanto às afirmativas. O questionário de avaliação pode ser visualizado no Apêndice D.

Foram obtidos dados qualitativos através do método de análise de videogravação, que como descrito por Garcez et al. (2011), permite uma análise posterior do processo de avaliação, com possibilidade de rever o material gravado a quantidade de vezes necessária para apurar as questões avaliadas.

O software usado nessa etapa foi o Camstudio⁸, que possui licença gratuita, e permitiu que toda a interação dos usuários com o computador no momento da avaliação pudesse ser gravada em um arquivo de vídeo no formato mp4, sendo este um dos formatos mais aceitos por softwares e equipamentos de reprodução de mídia audiovisual.

⁸ <https://camstudio.org/>

Os vídeos gerados com a técnica citada foram vistos de forma a contribuir com a análise do resultado do questionário. Além disso, foi quantificado o tempo que cada avaliador levou para chegar a determinado objetivo do roteiro previsto, a quantidade de mensagens de erro que o software emitiu durante o processo e o número de consultas a documentação.

4. DESENVOLVIMENTO DO PRODUTO

Nesta seção são apresentados os processos usados no desenvolvimento do produto desenvolvido no presente estudo, de forma a explicar de forma detalhada como foi realizada a construção dos módulos que compõem o *framework* desenvolvido neste trabalho. Os processos apresentados a seguir são relativos a criação da estrutura principal do *framework* e organização dos modelos variáveis quanto a utilização do FJSU para o desenvolvimento de jogos ubíquos.

4.1 DESENVOLVIMENTO DO *FRAMEWORK*FJSU

O desenvolvimento do *framework* FJSU foi realizado tomando como base os estudos de Costa Segundo (2011) e Buzeto (2015), onde foi definido um processo flexível, que possibilita que as etapas de *design*, desenvolvimento e testes possam ser realizadas novamente em qualquer momento durante o desenvolvimento, encontrando novos pontos variáveis, que representam modificações nas funcionalidades do FJSU.

Inicialmente, foi realizada a análise de domínio, que segundo Braga (2002), que representa a etapa de *design* do *framework*, e está na observação das funcionalidades presentes na maioria das aplicações desenvolvidas para um domínio específico, que no caso deste trabalho, foi o de jogos sérios ubíquos. Este processo foi realizado junto ao levantamento bibliográfico, onde foi feita a leitura de cada trabalho selecionado através do mapeamento sistemático, e verificado as funcionalidades existentes em cada um deles.

Com o processo anteriormente descrito, foi possível identificar os *frozen spots*, esses que foram reconhecidos através dos jogos analisados no mapeamento sistemático, a partir da identificação de elementos que fazem parte dos jogos ubíquos sendo essenciais para qualquer instanciação do *framework*, dessa forma, foi criado o conceito da estrutura básica do FJSU, que aborda de forma encapsulada o principal processo de funcionamento e elementos que compõem os jogos sérios ubíquos.

Durante o processo de criação do *framework*, foram verificados aqueles elementos que são relacionados a instâncias específicas de jogos ubíquos, e não se fazem essenciais na estrutura de todos eles, sendo esses elementos os *hot spots*, que trazem flexibilidade a estrutura do *framework*, não se limitando a produção de um tipo

específico de jogo.

Esse processo foi produzido levando em conta os jogos ubíquos e jogos sérios ubíquos analisados no processo de levantamento bibliográfico. Elementos como a utilização de sensor de GPS, o uso do *Kinect*, ou a adaptação de narrativa do jogo, que não estão presentes em todos os jogos analisados, foram classificados como opcionais, ficando a cargo do desenvolvedor utilizar ou não em seu jogo. Já elementos como conexão à internet, utilização de banco de dados e adaptação de conteúdo fazem são essenciais para os jogos sérios ubíquos, portanto, foram considerados *frozen spots*.

Na figura 4.1 pode ser visualizada a estrutura simplificada do *framework*. Essa estrutura caracteriza a abstração dos elementos presentes nos dois módulos que compõem o FJSU. O módulo de Aplicação trata sobre aquelas bibliotecas e componentes reusáveis que são direcionadas ao jogo em si, que será disponibilizado ao aluno para jogar em seu dispositivo. O módulo Web está voltado para o armazenamento dos dados e controle da aplicação, pelo administrador do sistema e professores.

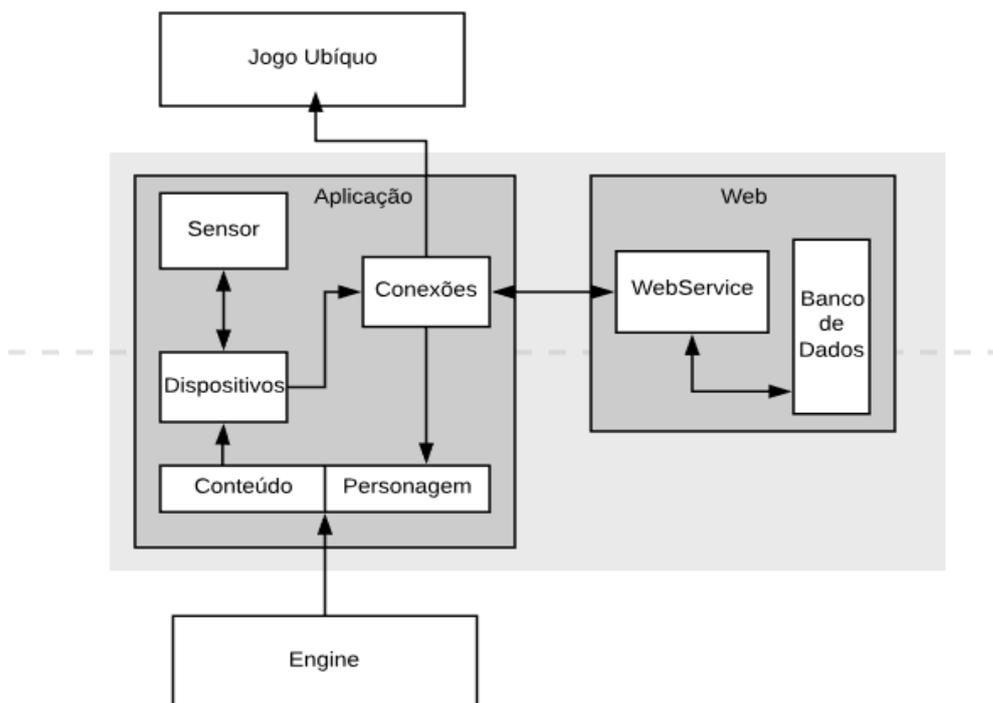
Como visto na figura 4.1, quanto às ligações da estrutura, o módulo de aplicação terá um ciclo baseado na criação de conteúdo jogável pela *Engine*, sendo eles os personagens e elementos do jogo, criados pelo desenvolvedor através do reuso dos objetos disponibilizados pelo *framework*.

Partindo desse ponto, devem ser implementados os métodos que o jogo irá utilizar conforme os dispositivos usados, para a obtenção e tratamento dos dados, podendo ser usados os métodos relacionados aos sensores dos dispositivos e localização espacial para isso.

Logo após os métodos relacionados a conexão ficarão responsáveis por administrar tanto o recebimento como envio de dados do servidor web para a adaptação do conteúdo, além disso, serão os responsáveis por mostrar esse conteúdo ao jogador no formato de um jogo ubíquo.

Já no módulo *web*, o sistema de *webservice* manterá o controle dos relatórios gerados para a administração do conteúdo usado pelo jogo, além dos formulários para a criação do mesmo por parte do administrador ou professores conteudistas. Além disso. Já o controle de banco de dados irá manter armazenado os conteúdos criados além do perfil do jogador, que pode ser buscado para o jogo através dos métodos do módulo aplicação.

Figura 4.1 - Estrutura Simplificada do *Framework*



Fonte: (O AUTOR).

4.2 MÓDULO APLICAÇÃO

O processo de desenvolvimento do módulo de aplicação foi realizado em dois momentos. O primeiro momento foi relacionado a criação do *framework* em um modelo de caixa branca, onde os componentes reusáveis mantinham o formato de classes abstratas e eram utilizados através da instanciação das mesmas e da invocação dos métodos através de linhas de código em um *script* controlador.

O segundo momento do desenvolvimento do módulo de aplicação foi referente ao formato de caixa cinza do *framework*, onde através da primeira avaliação, foi percebido dificuldades quanto a utilização da ferramenta pelos desenvolvedores, portanto, foi definido que o FJSU deveria seguir o processo para tornar-se um *framework* de caixa cinza seguindo o conceito de Markiewicz (2001). Os dois processos são descritos a seguir.

4.2.1 Caixa Branca

O módulo de aplicação mantém sua relação com o jogador, desde o layout que será apresentado a ele, a jogabilidade, interações com os dispositivos e sensores, conexões e adaptação de conteúdo. A premissa das bibliotecas e *prefabs* presentes nesse módulo está na criação da aplicação jogável, que estará em funcionamento no dispositivo utilizado pelo jogador.

Esse módulo foi desenvolvido sob a *engine Unity*, usando como linguagem padrão o C# e os componentes oficiais documentados pela desenvolvedora da engine. Na figura 4.2 pode ser visualizado o exemplo da escrita de um método presente no FJSU, que adiciona movimento ao personagem em um jogo de plataforma, com as funções de andar e pular.

Figura 4.2 - Escrita do método setMovement()

```
public void setMovement(GameObject personagem, Rigidbody2D corpo, float velocidade, float forçaPulo) {

    if (Input.GetAxisRaw ("Horizontal") > 0) {
        personagem.transform.Translate (Vector2.right * velocidade * Time.deltaTime);
        personagem.transform.eulerAngles = new Vector2 (0, 0);
    }

    if (Input.GetAxisRaw ("Horizontal") < 0) {
        personagem.transform.Translate (Vector2.right * velocidade * Time.deltaTime);
        personagem.transform.eulerAngles = new Vector2 (0, 180);
    }

    if (Input.GetAxisRaw ("Horizontal") == 0) {

    }

    //PULO
    if (Input.GetButtonDown ("Jump")) {
        corpo.AddForce (Vector2.up * forçaPulo);
    }

}
```

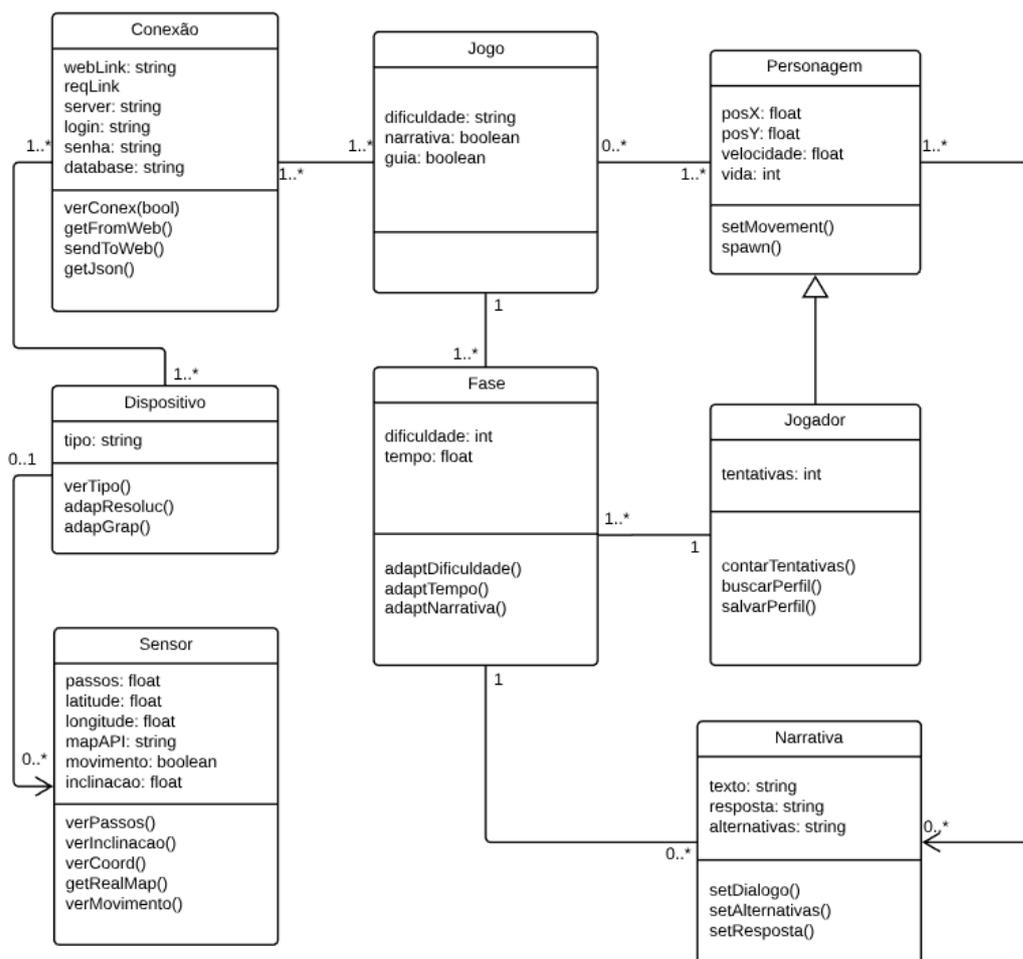
Fonte: (O AUTOR).

Os processos presentes neste módulo são relacionados ao controle de conexões de rede com a internet, banco de dados e sensores utilizados, interação com os sensores presentes no dispositivo e adaptação automática de conteúdo conforme a interação do jogador. Na figura 4.3 pode ser visualizada a estrutura de

classes, que norteou o desenvolvimento das classes e métodos desse módulo.

Por tratar-se de um *framework* de caixa branca composto por classes abstratas, para a sua utilização, deve ser criada uma classe que receberá como herança os métodos das classes abstratas.

Figura 4.3 - Estrutura de classes do *Framework*



Fonte: (O AUTOR).

A classe *Conexão* possui os métodos responsáveis por manter e tratar a conectividade do jogo com os demais dispositivos, banco de dados e *web service*. Além disso oferece recursos para o tratamento automático e adaptação de conteúdo conforme o tipo de conexão utilizada.

A classe *Dispositivo* é a que mantém relação com o principal dispositivo usado pelo jogador enquanto joga, dentro desta classe existem métodos para o reconhecimento do tipo de dispositivo, adaptação da qualidade gráfica conforme o dispositivo e de elementos da interface.

A classe *Sensor* possui os métodos responsáveis por administrar o uso

daqueles sensores internos e externos ao dispositivo, tanto para a captura de informações como o tratamento delas, para que possam ser adicionadas aos elementos do jogo.

A classe Jogo é a principal responsável por armazenar as variáveis globais, ela centraliza os processos das outras classes, e mantém as definições do usuário salvas enquanto ele está em jogo.

A classe Fase representa uma cena individual do jogo, que possui um nível de dificuldade específico e uma quantidade de tempo para ser completado, seus métodos são relacionados a adaptação da fase conforme o jogador interage com ela.

A classe Personagem possui os métodos necessários para que o próprio consiga movimentar-se e interagir com os elementos do jogo, representa qualquer personagem do jogo, desde o jogador, inimigos e outros NPC's. A classe Jogador é uma extensão da classe personagem, que é responsável por armazenar os dados do jogador, para que seja montado o seu perfil.

A classe Narrativa possui os métodos para a definição do conteúdo do jogo, sendo relativos a perguntas e respostas, podendo ser referente a um *quiz game* ou apenas na conversação dos personagens durante uma etapa de diálogo do jogo.

4.2.2 Caixa Cinza

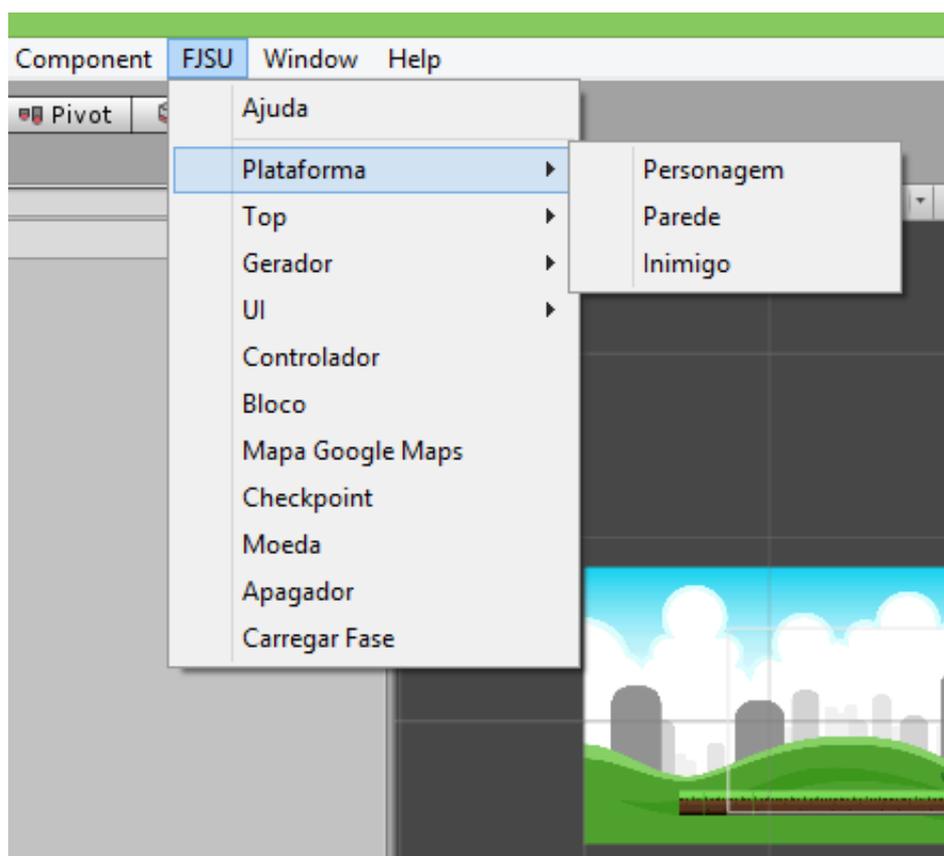
No segundo momento do desenvolvimento do módulo de aplicação do *framework*, o mesmo foi adaptado pensando em melhorar a usabilidade do usuário, de modo a possibilitar que o desenvolvedor possa criar *prefabs* e cenas funcionais com interação através do clique do mouse e com interface visual, sem depender de codificação. Essa adequação veio do pressuposto que o público alvo do *framework* são profissionais que não necessariamente possuem conhecimento em programação, dessa maneira, o *framework* deveria ter sua utilização simplificada, abstraindo o possível a utilização de códigos escritos pelo desenvolvedor, e tornando o processo de criar um jogo sério ubíquo feito pelo acoplamento de objetos reusáveis.

Nesse momento foi utilizado do recurso disponibilizado pela própria *Unity*, o *Unity Editor*, que permite a modificação da interface da *engine* conforme a necessidade, possibilitando a adaptação dela para a utilização do *framework* FJSU. O processo utilizado nessa etapa constou na criação de um *script*, que seria o

responsável por administrar a interface gráfica, que é incluído no pacote final do FJSU, fazendo com que a interface se adapte automaticamente ao *framework* quando importado em um novo projeto.

De forma a automatizar a ação do desenvolvedor, e também minimizar a dificuldade de utilização do *framework*, foram criados objetos e cenas baseadas nas classes criadas enquanto o mesmo estava no formato de caixa branca. Usando os próprios métodos das classes, os novos objetos foram inseridos em links da própria interface da *Unity*, criando um novo menu denominado FJSU. Cada objeto desse menu está documentado na página do *framework*, disponível no link do módulo web. Na figura 4.4 pode ser visualizado o menu do FJSU já instalado na interface do *Unity*, tal como seus componentes.

Figura 4.4 - Menu do FJSU



Fonte: (O AUTOR).

Ao selecionar um item do menu, a instância do objeto selecionado será adicionada a cena aberta no momento. Caso o desenvolvedor selecione, por exemplo, o item Personagem do menu de Plataforma, como pode ser visto na figura 4.5, será

criado um novo personagem 2D, com seu movimento horizontal controlável através das teclas do teclado e o pulo usando a barra de espaço. No entanto, alguns objetos necessitam que o *prefab* Controlador esteja instanciado na cena, onde o mesmo pode ser encontrado através do menu do FJSU.

Com a importação do FJSU, é criada uma pasta denominada Cenas, nela ficam disponíveis os exemplos criados para a posterior edição por parte do desenvolvedor, sendo mecânicas de jogos completos usando os próprios objetos do *framework*.

4.3 MÓDULO WEB

Sendo o segundo módulo do *framework*, o módulo Web é o responsável pela criação da plataforma administrativa de um jogo ubíquo, ficando a cargo de um administrador ou professor a utilização do ambiente desenvolvido.

As classes e métodos deste módulo foram criadas através da linguagem de *scripts* PHP⁹, pela funcionalidade de integrar um servidor web com um banco de dados MySQL¹⁰ e possibilidade de comunicação com as aplicações criadas pela engine *Unity*, sendo a usada para a criação do módulo de aplicação.

O objetivo principal em criar um ambiente administrativo para um jogo sério ubíquo está em oferecer ao docente ou administrador uma plataforma de cadastro do conteúdo que será utilizado no jogo, podendo gerenciar as escolhas a serem feitas em cada adaptação, seja por nível de dificuldade e estilo de aprendizado. Além disso, o ambiente pode gerar relatórios quanto a atividade do aluno, como número de erros e acertos e tempo para realizar uma atividade, permitindo ao administrador e ao docente gerir essas informações conforme necessidade.

O módulo *web* disponibilizado como objeto reutilizável deste trabalho conta com a gerência de conteúdo utilizado pelo jogo. Através do acesso a ele, o desenvolvedor pode criar questões e diálogos, e gerar relatórios deles em um formato JSON¹¹, escolhendo quais conteúdos quer adicionar aquele relatório, e se pretende que os mesmos sejam gerados de forma aleatória ou não. JSON é um acrônimo de Javascript Object Notation, e representa um formato de fácil criação e interpretação

⁹ <https://secure.php.net>

¹⁰ <https://www.mysql.com/>

¹¹ <https://www.json.org/>

de objetos, tanto para humanos como máquinas, comumente usados na troca de dados entre sistemas.

Caso o sistema seja usado por mais de uma pessoa, como diferentes professores, podem ser criadas contas separadas, onde cada um terá controle apenas do seu conteúdo. Na figura 4.5 pode ser visualizada a página inicial do site onde está hospedado o módulo *web*. Através do formulário de login, localizado no canto superior direito da página, pode ser acessado o painel administrativo. É disponibilizado um usuário padrão, sendo o usuário e a senha “admin”.

Figura 4.5 - Página inicial do módulo web



Fonte: (O AUTOR).

A página de acesso administrativo possui links para a criação do conteúdo educacional do jogo, sendo essas questões de três tipos, múltipla escolha, verdadeiro ou falso e dissertativa. A partir disso, os professores podem cadastrar uma delas, escolhendo a área de conhecimento e o nível de dificuldade. Pode ser gerado um relatório, abrangendo por exemplo, as questões criadas para a disciplina de matemática, com nível de dificuldade 3, podendo adicionar somente elas a determinada parte do jogo ou ele inteiro.

A utilização das questões dentro do jogo pode ser feita com a criação dos objetos categorizados como *quiz* no módulo de aplicação. Os objetos possuem a propriedade *url*, onde pode ser adicionado o link da página do relatório das questões

selecionadas. Com isso, será gerada uma nova questão aleatória das selecionadas, a cada interação do jogador com o objeto. Na figura 4.6 pode ser visto a página de administração do módulo web, onde pode ser realizado o processo citado anteriormente.

Figura 4.6 - Página de administrador do módulo Web.

The screenshot displays the 'Cadastro de questões' (Question Registration) interface. It features three filter buttons: 'Múltipla escolha' (Multiple choice), 'Verdadeiro ou falso' (True or false), and 'Descritiva' (Descriptive). Below this is the 'Relatório de Questões em JSON' (JSON Questions Report) section, which includes three dropdown menus for 'Tipo' (Type) set to 'Múltipla Escolha', 'Dificuldade' (Difficulty) set to '5', and 'Área de conhecimento' (Knowledge Area) set to 'Matemática'. An 'Enviar' (Send) button is positioned below the filters. The 'Relatório de Jogadores' (Player Report) section contains a table with the following data:

ID	Login	Dificuldade
1	mario	8
2	luigi	3
5	yoshi	1

Fonte: (O AUTOR).

Junto ao site em que está disponibilizado o módulo *web*, encontram-se informações sobre o *framework* e também a documentação do mesmo, onde pode ser encontrada a descrição de cada objeto, e como ele pode ser utilizado. Na figura 4.7 pode ser visualizada a página inicial da documentação.

Figura 4.7 - Página de documentação

Inspector

Para visualizar as propriedades dos objetos, é necessário clicar nele e na janela inspector, normalmente localizada no canto direito da tela.

UI

Todos os elementos da seção UI do FJSU devem ser inseridos dentro de um objeto Canvas, da própria Unity. O canvas pode ser adicionado clicando no menu GameObject, UI e Canvas.

Controlador

É o elemento principal do FJSU, irá armazenar os dados globais do jogador e também controlar os outros objetos do jogo. As adaptações automáticas do jogo podem ser ativadas e desativadas nele.

Personagem de Plataforma

O personagem do jogo de plataforma, que já possui movimentação 2D definida pelas setas do teclado, e o pulo pela barra de espaço. A velocidade de movimento e a força do pulo podem ser configurados através do inspector do objeto.

É necessário selecionar o controlador no seu inspector, tanto do script Movimento_plataforma como Adaptar_personagem.



Fonte: (O AUTOR).

Para acessar a documentação antiga, clique no link abaixo

[Documentação antiga](#)

Objetos

[Personagem de Plataforma](#)

[Parede de Plataforma](#)

[Inimigo de Plataforma](#)

[Personagem de Top-Down](#)

[Inimigo de Top-Down](#)

[Gerador de Inimigos](#)

[Gerador de Moedas](#)

[Quizz](#)

[Login](#)

[Pontuação](#)

[Pause](#)

[Camera do Dispositivo](#)

[Bloco](#)

[Mapa do Google Maps](#)

[Checkpoint](#)

[Moeda](#)

[Apagador](#)

[Carregar Fase](#)

[Camera Seguir o Jogador](#)

[Movimento usando o acelerômetro](#)

4.4 MODELAGEM DE JOGOS SÉRIOS UBÍQUOS

A etapa de modelagem de jogos sérios ubíquos representa o momento onde o desenvolvedor define quais aspectos o jogo que ele pretende desenvolver deve possuir, seguindo um caminho de *hot spots* pré especificados, que servem como um guia para o momento de criação do jogo.

Após a definição da estrutura principal, junto a análise de domínio, foram estabelecidos os *hot spots*, com base nos estudos analisados, sendo verificados os elementos variáveis nos jogos visualizados nos estudos, que possibilitam a flexibilidade no desenvolvimento de jogos sérios ubíquos a partir do *framework*. Este processo entrega ao desenvolvedor não somente uma opção quanto a um aspecto a ser definido no seu jogo, como por exemplo, quanto ao sistema de pontuação do mesmo, mas sim que possam ser definidas uma variante de estruturas complexas a partir da principal.

Os tipos *hot spots* que foram encontrados são relacionados aos aspectos de jogabilidade, estilo de jogo, plataforma atingida etc, que refletem em uma escolha

particular do desenvolvedor quanto ao formato definido para o seu jogo.

Figura 4.8 Exemplo de caminho seguido pelo desenvolvedor

Gráficos	2D		3D	
Estilo de Jogo	Plataforma	Shooter	Puzzle	RPG
Número de Jogadores	Singleplayer		Multiplayer	
Plataforma do jogo	multiplataforma	PC	Consoles	Mobile
Tipo de conexão	Online		Offline	
Tipo de armazenamento	Armazenamento no dispositivo		Armazenamento Externo	
Tipo de adaptação	Adaptação de narrativa		Adaptação de dificuldade	
Tipo de pontuação	Pontuação por moedas	Pontuação por inimigos	Pontuação por tempo	

Legenda

Escolha realizada pelo desenvolvedor

Opção disponível

Fonte: (O AUTOR).

A figura 4.8 ilustra a escolha feita no exemplo citado, no qual o caminho é representado pelos retângulos cor de laranja, sendo os retângulos brancos a representação de algumas das possibilidades de variações dos aspectos do jogo, que poderiam ter sido definidas pelo desenvolvedor.

Como pode ser visto no exemplo, o *designer* de jogos digitais definiu que o jogo utilizará gráficos em 2D, tornando-o mais acessível para dispositivos com pouco processamento. O estilo do jogo será um *puzzle*, onde o objetivo está na resolução de desafios de raciocínio, sendo muito utilizado em jogos casuais.

Quanto ao número de jogadores, ele foi definido como *singleplayer*, onde seu modo de jogo é baseado em apenas uma pessoa. Sua distribuição foi escolhida para funcionar na maioria das plataformas existentes. Deve ser necessário conexão para jogar, visto que o armazenamento externo indica que os dados do jogador não são

armazenados no dispositivo. O jogo deve adaptar a dificuldade conforme o nível do jogador, aumentando-a ou diminuindo-a de acordo com as ações do mesmo enquanto joga. Por fim, a pontuação do jogo é dada conforme o tempo que o jogador levar para concluir o desafio.

Os *hot spots* foram documentados no formato de levantamento de requisitos, de forma que a definição da estrutura do jogo seja feita de forma prática. O levantamento de requisitos do *framework* FJSU é realizado na forma de perguntas relativas ao processo de desenvolvimento de um jogo sério ubíquo, a partir dos processos que fazem parte da sua estrutura. Esse processo pode ser visualizado no Apêndice B deste trabalho.

5. AVALIAÇÃO

Nesta seção são descritos de forma detalhada os processos que fizeram parte da avaliação do *framework* FJSU, de forma a explicar como os resultados foram obtidos e trazendo a análise sobre os dados no final. Inicialmente serão expostos e discutidos os resultados do mapeamento sistemático realizado como levantamento bibliográfico deste trabalho e logo após, será apresentado todo o processo de avaliação do FJSU.

5.1 RESULTADOS DO MAPEAMENTO SISTEMÁTICO

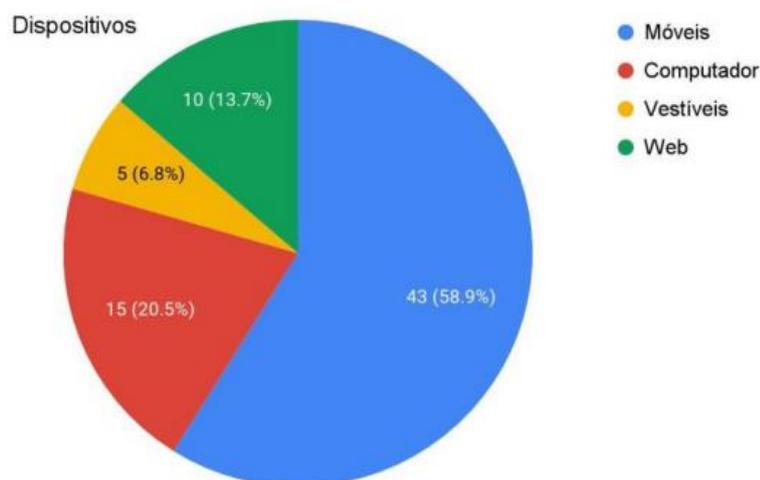
Os resultados dessa etapa foram coletados com base nas quatro questões de pesquisa que foram citadas na seção 3 deste trabalho, que nortearam esta pesquisa. A seguir, são mostrados os dados obtidos durante esse processo, tal como a discussão de cada uma das questões.

5.1.1 Q1: Como está sendo feita a entrada de dados no sistema?

A computação ubíqua tem por sua principal característica a mobilidade e o intuito de esconder os processos educacionais da percepção do usuário. Para tanto, são utilizados tecnologias que possibilitem a captura dos dados referentes ao contexto em que o usuário se encontra, estes dados podem ter relação a sua localização física, aspectos físicos do ambiente onde o mesmo se encontra, percepções referentes às características de aprendizado do usuário etc., realizando uma troca de dados constante entre os dispositivos e o ambiente, com o mínimo de interação possível do usuário (PIMENTA, 2014).

Toda essa demanda pode ser obtida através de sensores, que monitoram as atividades realizadas pelo usuário durante o jogo, no entanto, devido a numerosa quantidade de tipos diferentes de dados, a variedade de sensores utilizados também é alta. Com isso, se fez necessário analisar como está sendo feita a captura destes dados nos jogos sérios ubíquos desenvolvidos, permitindo o conhecimento sobre os principais dispositivos e sensores.

Figura 5.1 Tipos de dispositivos

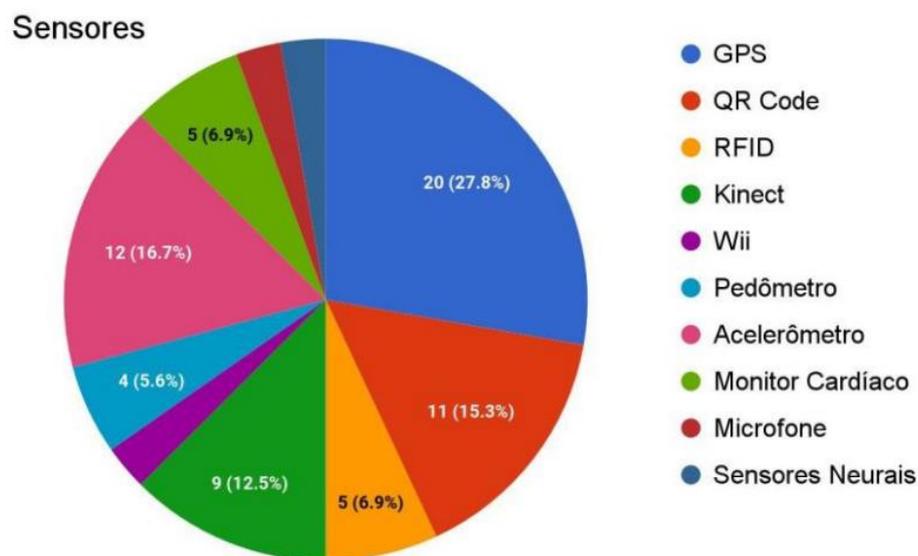


Fonte: O autor

Dentre os artigos selecionados, 73 deles possuíam relação direta com algum tipo de dispositivo, os outros 17 representavam um estudo bibliográfico, metodologia ou framework que abrangia o conceito sem focar em alguma tecnologia específica. Na figura 1, pode ser percebido que os dispositivos móveis representam 43 dos trabalhos que relatam o desenvolvimento ou aplicação de um jogo sério ubíquo, nessa categoria, foram considerados dispositivos como smartphones, tablets e notebooks.

Os computadores do estilo desktop representam 15 dos estudos, plataformas web são encontradas em 10 estudos e os dispositivos vestíveis fazem parte de cinco dos estudos analisados. Além do dispositivo principal, que suporta o jogo e permite a sua utilização, tem-se os sensores, que captam as informações e trazem a modularidade de um jogo sério ubíquo, não se restringindo a uma única possibilidade.

Figura 5.2 Sensores



Fonte: O autor

Dentre os sensores usados nos estudos selecionados, podem ser visualizados na Figura 2, aqueles que buscam dados referentes a localização física do usuário: O GPS representou 20 dos artigos selecionados, também foram encontrados jogos que utilizam RFID em cinco estudos e QRCode em 11. Dos sensores que estão relacionados ao monitoramento gestual e de atividades: O Kinect representa 9 estudos, partindo da mesma premissa que o Wii, que representa dois, também foram analisados 12 artigos que usam o Acelerômetro, 4 o Pedômetro, 5 o Monitor Cardíaco e 2 Sensores Neurais. Dois dos estudos usaram o microfone para reconhecimento de voz e tonalidade vocal.

5.1.2 Q2: Como os dados estão sendo tratados pelo sistema?

O tratamento dos dados permite que o aspecto de inteligência e ubiquidade de um jogo sério possa ser alcançado. Durante a análise dos artigos, foi percebido que alguns dos modelos de tratamento de dados seguiam um mesmo padrão, sendo eles:

- A. Feedback de performance: Os trabalhos de (Al Osman/2016, Postolache/2016, Vandewaetere/2013, Sajjadi/ 2014) utilizam o sistema de feedback, que transmite ao jogador a sensação de recompensa, e permite o engajamento e a motivação para continuar jogando. Dentro dos jogos sérios ubíquos, os trabalhos de esse sistema vêm com o acompanhamento da performance, medindo a evolução do jogador através dos dados obtidos pelos sensores,

podendo ser a quantidade de passos dados durante uma atividade, realização de um gesto corretamente, conclusão de uma atividade proposta, tempo gasto em uma atividade, expressões faciais e batimentos cardíacos. A recompensa normalmente é estabelecida em pontuação, algum item ou moeda virtual, que pode ser utilizado dentro do jogo, estimulando a competitividade entre os jogadores e trazendo a sensação de objetivo a atividade realizada.

- B. Análise de atividade: Os trabalhos de (Szegletes/2013, Yang/2015, Sajjadi/2014, Yahya/2015, Hamdaoui/2015) são exemplos de jogos que utilizam a análise de atividade. Nessa categoria, se encaixam os jogos ubíquos que procuraram uma adaptação automática do seu conteúdo conforme a utilização do usuário, buscando as fragilidades e necessidades do mesmo quanto a utilização, seja diminuindo a dificuldade para que o jogador não se sinta frustrado ao jogar, ou aumentando a mesma, permitindo um desafio maior. Essa adaptação pode ser realizada em três etapas: A primeira envolve escolher quais os recursos serão adaptados, como por exemplo a quantidade de inimigos ou o tempo para completar uma atividade; A segunda é definir e categorizar os comportamentos do jogador conforme os recursos escolhidos, como por exemplo, quando ele falhar no cumprimento de um objetivo por 10 vezes, é definido que o jogo deve diminuir a dificuldade, ou quando perceber através dos sinais vitais que o jogador encontra-se estressado, deve mudar para outra atividade ou oferecer *feedback* a ele; Por fim, a terceira etapa envolve a definição do algoritmo que irá adaptar o conteúdo.
- C. Questionário: Os trabalhos de (Arnold/2013, Soflano/2015, Vayanou/2014) fazem o uso de questionários, que abordam uma adaptação de conteúdo pré-estabelecida, seja no formato de perguntas que demonstram o propósito adaptativo ou através de diálogos da narrativa do jogo, que permitem que o processo seja camuflado. A partir de determinado ponto, pode ser descoberto o perfil do usuário, é definido como deve ser feito o andamento do jogo, para que o mesmo consiga interagir da melhor forma.

A partir das informações levantadas referente ao tratamento dos dados, foi possível entender e classificar como o jogo se comporta para adaptar-se a

necessidade do jogador. Dentre os métodos vistos, aqueles que se aplicam em formato de questionário mostraram-se em maior número e de forma mais simples, no entanto, são limitados, visto que abrangem apenas as possibilidades de adaptação pré-estabelecidas. Já aqueles que possuem a adaptação automática pela atividade do usuário, através da análise e apuração em tempo real das informações obtidas, tendem a absorver melhor o perfil do jogador, no entanto, sua aplicação e o processo de desenvolvimento é mais complexo, envolvendo muitas vezes sensores que descaracterizam a portabilidade e mobilidade do jogo.

5.1.3 Q3: Como estão sendo avaliados os jogos?

Quanto aos métodos de avaliação, foi percebido que os que utilizam questionários podem produzir boas respostas qualitativas quanto ao comportamento do jogo e possibilitar que melhorias possam ser efetuadas. A realização deles pode ser feita antes e após uma aplicação prática do jogo, como no estudo de Kohlmann (2012), Lumpoon (2016), Gicquel (2013) permitindo verificar o conhecimento prévio dos jogadores sobre o assunto abordado pelo jogo e também averiguar se o aluno conseguiu aprender. O estudo de AL OSMAN (2016) verifica o nível de stress do jogador antes e depois de jogar, sendo os resultados obtidos através de um formulário aplicado após a experiência.

Estudos com uma avaliação quantitativa como o de Soflano (2015) e Yang (2015), podem ser realizados através de questionários ou de forma automática, captando as ações do jogador durante o jogo e classificando-as conforme a necessidade da pesquisa. Os dados são baseados em números, apurando o resultado conforme a quantidade de testes realizados.

Propostas que envolvem uma avaliação complexa conduzida em uma fatia de tempo pré-determinada podem ser vistas nos estudos de Hwang (2012) e Köhlmann (2012), que definem entrevistas iniciais, verificando o perfil dos jogadores, e os conduzem a utilizar o jogo durante momentos do seu dia, como filas de banco, antes de dormir ou durante uma corrida, usando recursos de áudio e geoposicionamento. No final da pesquisa é realizado a análise dos dados obtidos no processo de avaliação, verificando as mudanças de comportamento ou conhecimento adquirido. Este tipo de avaliação costuma produzir bons resultados quanto à avaliação dos benefícios adquiridos com o uso regular do jogo.

Seguindo uma proposta de avaliação a partir de observações do comportamento dos jogadores enquanto envolvidos com a atividade definida, os estudos de Brandt (2016), seguem este modelo, onde não existe qualquer intromissão do avaliador enquanto o teste é realizado, de modo a captar a espontaneidade das ações do jogador e produzir suas conclusões a partir delas. Também podem ser vistos, como nos estudos de Szegletes (2013), Duarte (2014) métodos onde um avaliador acompanha a realização das atividades através de um dispositivo móvel ou computador, verificando os dados obtidos em tempo real. Este tipo de avaliação apresenta resultados satisfatórios quando usada para verificar reações emocionais e o tempo que o jogador leva para aprender a jogar.

Como método de avaliação específico para *frameworks*, os estudos de KLOPFER (2012), Rodrigues (2016), Buzeto (2015) utilizam estudos de caso, onde uma ou mais aplicações são criadas usando a estrutura do *framework* desenvolvido, como forma de validar se os aspectos necessários para aquele domínio específico são atendidos no jogo desenvolvido, e se o *framework* permanece eficiente como ferramenta para oferecer suporte durante todo o processo de desenvolvimento. Além disso, Buzeto (2015) aplicou uma capacitação para alunos de cursos relacionados a computação, onde foi ensinada a teoria básica que envolve jogos sérios ubíquos e a utilização do *framework*. Após a capacitação, os alunos desenvolveram e projetaram jogos, de forma a validar a ferramenta pelo público externo.

5.1.4 Q4: Quais tecnologias estão sendo utilizadas no desenvolvimento dos jogos?

São utilizadas plataformas de desenvolvimento para dispositivos móveis na maioria dos artigos selecionados, 23 deles usavam ferramentas como Android Studio, Ionic e Phonegap, foi percebido um recorrente aumento do uso daquelas consideradas híbridas¹² nos últimos anos quando comparado às nativas. Além disso as engines de jogos são usadas em 15 dos estudos, entre elas A Unity 3D, Unreal Engine, GameMaker Studio, RPG Maker, e também os mundos virtuais OpenSimulator e Second Life.

Quanto aos dispositivos e sensores utilizados para a entrada de dados, foi

¹² Ferramentas para desenvolvimento híbrido mesclam as tecnologias presentes nas aplicações para web e com o código nativo, permitindo que a aplicação seja compatível com vários dispositivos e utilize todos os recursos e sensores disponíveis.

percebido que aqueles considerados móveis estão em mais de 50% dos artigos, com isso, pode considerando a importância da mobilidade nos jogos sérios ubíquos, e também o principal foco dos mesmos. Além disso, os jogos para dispositivos vestíveis representaram a menor parcela dos estudados, no entanto, os mesmos mostraram-se recentes, indicando um fator de crescimento e possibilidade de futuras pesquisas usando-os.

5.1.5 Discussão

Quanto aos dispositivos e sensores utilizados para a entrada de dados, foi percebido que aqueles considerados móveis estão em mais de 50% dos artigos, com isso, pode ser considerado a importância da mobilidade nos jogos sérios ubíquos, e também o principal foco dos mesmos. Além disso, os jogos para dispositivos vestíveis representaram a menor parcela dos estudados, no entanto, os mesmos mostraram-se recentes, indicando um fator de crescimento e possibilidade de futuras pesquisas usando-os.

Os sensores mostraram-se diversificados e abrangendo aplicações para alguns propósitos, como a educação, condicionamento físico e mental e reabilitação motora, onde aqueles que buscavam o contexto de localização se mostraram em maior número, podendo ser constatado que esta informação está diretamente ligada a popularidade dos dispositivos móveis e os sensores embutidos na estrutura deles. Além disso, a presença de pesquisas que relatem a utilização de sensores biológicos mostra uma premissa do potencial e das possibilidades que um jogo sério ubíquo pode alcançar.

A partir das informações levantadas referente ao tratamento dos dados, foi possível entender e classificar como o jogo se comporta para adaptar-se a necessidade do jogador. Dentre os métodos vistos, aqueles que se aplicam em formato de questionário mostraram-se em maior número e de forma mais simples, no entanto, são limitados, visto que abrangem apenas as possibilidades de adaptação preestabelecidas. Já aqueles que possuem a adaptação automática pela atividade do usuário, através da análise e apuração das informações obtidas por um banco de dados, tendem a absorver melhor o perfil do jogador, no entanto, sua aplicação e o processo de desenvolvimento é mais complexo, envolvendo muitas vezes sensores que descaracterizam a portabilidade e mobilidade do jogo.

Quanto aos métodos de avaliação, foi percebido que os que utilizam questionários podem produzir boas respostas qualitativas quanto ao comportamento do jogo e possibilitar que melhorias possam ser efetuadas, no entanto, aqueles que são baseados em uma pesquisa quantitativa podem não ser tão efetivos para a precisão dos resultados, para uma análise desse tipo, a utilização de métodos automáticos de avaliação ou a presença de uma avaliação por observação de comportamento pode demonstrar melhores resultados.

O conhecimento quanto às tecnologias que estão sendo utilizadas no desenvolvimento de jogos ubíquos possibilitou visualizar as suas características necessárias, segundo Klopfer (2012), eles devem exigir o mínimo de recursos computacionais possíveis para poder funcionar na maioria dos dispositivos utilizados hoje em dia, como smartphones, tablets e computadores. Com isso, em sua maioria, os mesmos não utilizam motores gráficos 3D, que irão exigir maior processamento de vídeo e armazenamento, também gerando um consumo maior de bateria dos dispositivos móveis. Nesses casos, é optado pela utilização de ferramentas que possibilitam o desenvolvimento de jogos 2D, diminuindo o consumo dos recursos. Além disso, as ferramentas devem possuir integração aos sensores, ou possibilidade de extensão através de código aberto.

5.2 AVALIAÇÃO COMO CAIXA BRANCA

A avaliação foi aplicada em duas etapas do desenvolvimento do trabalho. A primeira etapa foi referente ao FJSU enquanto mantinha sua estrutura de caixa branca, ou seja, possuía apenas classes abstratas, que para a sua utilização era necessário criar uma instância da mesma através da codificação de um *script* controlador. A segunda etapa foi realizada quando o mesmo já estava em uma estrutura de caixa cinza, com os elementos instanciados a partir da seleção e clique do mouse, sem o envolvimento direto do desenvolvedor na criação de *scripts* e linhas de código.

A primeira etapa da avaliação foi realizada no formato de um roteiro a ser seguido usando o *framework*, onde os desenvolvedores fizeram a instanciação da classe e a utilização dos métodos para a criação de um protótipo de jogo.

A parte inicial da etapa de avaliação foi a oferta de um curso de quatro horas sobre o desenvolvimento de jogos usando a *engine Unity* para os alunos que iriam atuar como desenvolvedores voluntários. Esse curso abrangeu os aspectos básicos

de criação de jogos 2D, manipulação de objetos, *sprites* e criação de *scripts*, e teve como objetivo nivelar o conhecimento dos avaliadores, visto que os mesmos não possuíam experiência com desenvolvimento de jogos.

Logo após, foi realizada a avaliação, que ocorreu durante o período de 3 horas, abrangendo uma conceituação inicial da utilização do *framework*, a proposta do jogo que deveria ser desenvolvido, a apresentação do roteiro que eles deveriam seguir para criar o jogo e o preenchimento do formulário relativo à experiência enquanto utilizava o *framework*, sendo o mesmo encontrado no Apêndice C desta dissertação.

O roteiro utilizado pelos desenvolvedores constava na criação de um jogo de plataforma 2D com conteúdo buscado do servidor do módulo web, onde questões com alternativas eram mostradas ao jogador, que deveria movimentar o personagem até um dos blocos, e pular atingindo o que continha a alternativa correta. O resultado final desenvolvido por um dos avaliadores pode ser visualizado na figura 5.1

Figura 5.3 - Jogo Sériu Ubíquo desenvolvido



Fonte: (O AUTOR).

Para o levantamento dos dados, foi utilizado um formulário baseado na escala de Likert (1932), com o objetivo de avaliar os quesitos de usabilidade do *framework*, que foi aplicado aos voluntários após realizarem o roteiro previsto, o formulário pode ser visualizado no Apêndice D. Foram definidas 9 questões objetivas e 2 dissertativas

opcionais, que perguntavam, como por exemplo, se a documentação do FJSU auxiliou no momento no desenvolvimento, se foi necessário muito estudo para poder utilizar o FJSU, entre outros questionamentos.

Nesse momento da avaliação, não foi utilizado nenhum método para quantificar a pontuação de cada resposta, apenas foi verificado o número de avaliadores que preencheu determinada alternativa, e o percentual respondido de cada uma. O resultado do questionário pode ser visualizado no quadro 5.1.

Quadro 5.1 - Respostas da avaliação enquanto caixa branca

Questão	Nada	Pouco	Suficiente	Médio	Muito
Q1	0	0	4 (50%)	1 (12,5%)	3 (37,5%)
Q2	0	0	2 (25%)	2 (25%)	4 (50%)
Q3	0	1 (12,5%)	2 (25%)	2 (25%)	3 (37,5%)
Q4	2 (25%)	1 (12,5%)	2 (25%)	1 (12,5%)	2 (25%)
Q5	1 (12,5%)	3 (37,5%)	0	4 (50%)	0
Q6	0	0	1 (12,5%)	2 (25%)	5 (62,5%)
Q7	0	3 (37,5%)	4 (50%)	1 (12,5%)	0
Q8	0	2 (25%)	2 (25%)	3 (37,5%)	1 (12,5%)
Q9	0	0	1 (12,5%)	3 (37,5%)	4 (50%)

Fonte: (O AUTOR).

Das questões objetivas, foi verificado que aquelas alternativas que foram preenchidas por metade ou mais dos avaliadores receberam destaque, mostrando que todas concentram-se entre as colunas de “suficiente” e “muito”. As questões 2, 6 e 9, que abordam a utilidade do *framework*, importância da documentação e o aprendizado de desenvolvimento de jogos foram as que obtiveram maior número de alternativas como “muito” preenchidas, mostrando um resultado positivo, demonstrando que o *framework* útil seguindo sua proposta de desenvolvimento de jogos sérios ubíquos, que a documentação do mesmo está condizente ao servir como apoio ao desenvolvedor, e também, que o mesmo pode ser usado para o aprendizado na criação de jogos.

Da mesma forma, questão 5, que questiona o auxílio necessário para utilizar o *framework*, mostra que os avaliadores precisam de um recurso para que consigam utilizá-lo, dificultando a autonomia do desenvolvedor, mostrando-se um resultado não

satisfatório. As questões 1 e 7, que abordam a facilidade na utilização do *framework*, e as limitações que ele possui, obtiveram um resultado neutro, refletindo que apesar de alguns dos avaliadores consideraram o framework muito fácil de utilizar, a sua maioria se mostrou neutra, e quanto às limitações, foi percebido a mesma característica.

Das questões dissertativas, quanto aos recursos que o framework poderia ter, e os pontos a serem melhorados, foram obtidas as seguintes respostas:

1. “Poderia ter mais recursos, como suporte a realidade virtual”;
2. “Seria interessante exemplos prontos de uso prático inseridos na documentação do framework”;
3. “Poderia ser mais simples a forma de utilizar os métodos, sem precisar instanciar as classes”.

Estas requisições foram analisadas como atualização do *framework*. A questão de realidade virtual não foi implementada pelo motivo de já existirem bibliotecas gratuitas para esse fim. Quanto aos exemplos prontos, foi pensado para que no formato de caixa cinza, o FJSU já contasse com cenas inéditas e completamente editáveis pelos desenvolvedores.

Enquanto no formato de caixa branca, foi percebido que os desenvolvedores estavam com algumas dificuldades na utilização do *framework*, visto que não estavam acostumados a codificação de *scripts* para a *Unity*, portanto, foi entendido que o FJSU deveria seguir o caminho descrito por Braga (2002) para tornar-se um *framework* de caixa cinza, englobando uma interface visual para a utilização dos elementos, de forma a facilitar o entendimento para o desenvolvedor, tornando o processo intuitivo e prático no momento do desenvolvimento.

5.3 AVALIAÇÃO COMO CAIXA CINZA

Está segunda etapa da avaliação foi realizada de forma a avaliar o *framework* não somente em quesito de usabilidade, mas também buscando uma ampla visão sobre a proposta de facilitar e tornar mais rápido o desenvolvimento de jogos sérios ubíquos usando o FJSU. Para tanto, foi definido um modelo de caso de testes baseado no trabalho de Nunes (2014), tendo como parâmetro o desenvolvimento de um jogo

baseado no caso de uso apresentado no Apêndice D.

5.3.1 Pré-Avaliação

Para essa proposta, inicialmente foi realizada uma pré-avaliação com alunos de Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal de Santa Maria (UFSM). Oito alunos se disponibilizaram nesse momento para atuarem como desenvolvedores voluntários, que foram divididos em quatro duplas, de forma a equilibrar o grupo por experiência com a ferramenta.

Dos alunos participantes, quatro deles já haviam utilizado a *Unity*, apenas um desses quatro possuía um nível de conhecimento mais aprofundado na ferramenta. Os outros quatro alunos não possuíam experiência com a ferramenta, portanto, as duplas foram formadas por um aluno sem experiência e outro que possuía algum contato com a *Unity*.

Além dessa divisão, foi pedido que dois dos grupos utilizassem o FJSU para desenvolver o caso de teste proposto, sendo que os outros dois deveriam utilizar o *Unity* sem qualquer modificação, criando os elementos e *scripts* que fazem parte do desenvolvimento do jogo, apenas foram disponibilizadas as mesmas *sprites* para ambos os grupos.

Para a aplicação do teste, foi utilizado o caso de uso que pode ser visualizado no Apêndice C deste trabalho. Os alunos deveriam basear sua proposta de jogo de forma que todos os elementos essenciais sejam preenchidos, sendo eles, a necessidade de um personagem controlável pelo jogador, inimigos, um objetivo, um sistema de pontuação e conteúdo. Os grupos que utilizaram o FJSU foram instruídos a utilizar a documentação como suporte durante o processo de desenvolvimento e aprendizagem, da mesma forma, foi instruído aos outros grupos que não utilizaram o *framework* que buscassem no próprio site da desenvolvedora da *Unity* a documentação e suporte.

Foi estipulado o tempo de oitenta minutos para que os grupos se organizassem e desenvolvessem o jogo. Durante esse processo, os alunos foram acompanhados e auxiliados a como consultar a documentação para resolver alguma dúvida ou resoluções simples da interface da *Unity*, onde não foi realizada nenhum tipo de interação que solucione algum problema ou interrompa o processo de criação e aprendizado do aluno. Após esse tempo estipulado, foi verificado quais elementos que

os desenvolvedores conseguiram atingir no jogo criado, como pode ser visualizado no quadro 5.2.

Quadro 5.2 - Elementos alcançados nos jogos da pré-avaliação

Utilizou o FJSU?	Sim		Não	
	Grupo 1	Grupo 2	Grupo 3	Grupo 4
Personagem controlável	X	X	X	X
Inimigos	X	X		
Objetivo	X	X	X	X
Sistema de Pontuação	X	X	X	
Conteúdo	X	X		

Fonte: (O AUTOR).

Como pode ser visto no Quadro 5.1, tanto o Grupo 1 como o Grupo 2, sendo aqueles que utilizaram o FJSU durante o processo de avaliação, conseguiram criar a maioria dos elementos presentes no caso de uso no seu jogo, onde o Grupo 1 chegou a usar o módulo web para a gerência de conteúdo do jogo e adaptação de dificuldade dos inimigos conforme o perfil de jogador.

Quanto aos grupos que não usaram o *framework*, no tempo estipulado, o Grupo 3 conseguiu criar três dos elementos pedidos, sendo o personagem controlável, o objetivo e o sistema de pontuação, e o Grupo 4 conseguiu programar um personagem controlável e o objetivo do jogo.

Com esse resultado, pode ser observado que os grupos que utilizaram o *framework* conseguiram criar todos os elementos propostos no tempo estipulado, contando com o tempo de aprendizado da ferramenta e criação do jogo. Além disso, pode ser percebido as dificuldades que os grupos que não utilizaram o *framework* tiveram durante o processo, sendo relativos à criação de inimigos para o jogo e a definição do conteúdo, onde apenas foi definido um objetivo simples de levar o personagem até determinado ponto do mapa passando por obstáculos.

Como forma de levantamento de dados dos desenvolvedores voluntários, foi

aplicado o questionário encontrado no Apêndice D deste trabalho, sendo este baseado no modelo TAM 3 de Venkatesh (2008). O questionário contém 15 afirmativas divididas em três aspectos, utilidade, facilidade e intenção, e uma questionando qual plataforma foi utilizada pelo desenvolvedor no momento da avaliação, sendo o FJSU ou a *Unity* em seu formato original. As 15 afirmativas são citadas a seguir:

1. Consigo desenvolver um jogo usando o FJSU.
2. Consigo criar diferentes jogos usando o FJSU.
3. O software possui as ferramentas necessárias para criar um jogo.
4. Encontrei muitos erros durante o desenvolvimento.
5. A documentação do FJSU me ajudou no desenvolvimento.
6. Acho que o FJSU me ajuda a aprender sobre desenvolvimento de jogos.
7. Acho que o FJSU facilita o desenvolvimento de jogos.
8. Precisei estudar muito para utilizar o FJSU.
9. Precisei consultar a documentação muitas vezes.
10. As vídeo aulas me ajudaram a aprender.
11. Consigo encontrar facilmente as ferramentas que preciso.
12. Recomendo o FJSU para iniciantes no desenvolvimento de jogos.
13. Gostei de usar o FJSU para desenvolver um jogo.
14. Me sinto motivado para desenvolver um jogo usando o FJSU.
15. O tempo passou rápido enquanto usava o FJSU, me mantive concentrado(a).

As questões referentes a utilidade buscam entender o quão útil para o desenvolvimento de jogos sérios ubíquos foi a plataforma utilizada, sendo relativos a verificar o quão completo está o FJSU, se o mesmo não está limitado ou se possui muitos erros, além de verificar se as ferramentas disponíveis suprem as necessidades dos desenvolvedores para a criação do seu jogo.

As questões que abordam a facilidade estão relativas ao processo de aprendizado e intuição do desenvolvedor enquanto utiliza a ferramenta, as questões buscaram captar o quanto foi necessário consultar a documentação, e se a mesma foi suficiente e útil durante o processo.

As questões relativas à intenção buscam verificar o aspecto de imersão do desenvolvedor com a plataforma de desenvolvimento. Foram levantados dados

referentes a motivação para criar um jogo usando a plataforma, além de verificar o quanto ele ficou satisfeito com o resultado obtido. No quadro 5.3 pode ser observado o quantitativo de respostas para cada opção do questionário. Conforme a legenda, as células verdes representam o quantitativo daqueles avaliadores que utilizaram o FJSU (grupo 1), já as células azuis representam aqueles que usaram o Unity puro (grupo 2) sem qualquer modificação.

Os resultados foram classificados como positivo (P), caso a maioria das respostas marcadas estivessem entre as alternativas “concordo muito” e “concordo plenamente”, neutro (n), caso elas estivessem como “concordo”, ou mantivessem uma distribuição igual entre as classificações, e negativo (N), caso as mesmas estivessem entre “não concordo” e “concordo parcialmente”. No entanto, nas questões 4, 8 e 9, devido a estrutura da afirmação, os resultados positivos e negativos invertem-se, fazendo com que caso a maioria das respostas esteja entre “concordo muito” e “concordo plenamente”, o mesmo seja classificado como negativo, e caso esteja entre “não concordo” e “concordo parcialmente” como positivo.

Quadro 5.3 - Respostas do formulário da pré-avaliação

Questão	Não concordo		Concordo parcialmente		Concordo		Concordo muito		Concordo plenamente		Resultado		
											P = positivo n = neutro N = negativo		
Q1		1		1	1		1		1	1	P	N	
Q2		2			3					1	n	N	
Q3		1			1	2	2				P	n	
Q4	2		1	1		2					P	n	
Q5		1	1	1	1				1	1	n	N	
Q6		1			3	1				1	n	n	
Q7					1	2	2			1	P	n	
Q8	3	1				2					P	n	
Q9	1		2			2		1			P	n	
Q10			1		1	2			1	1	n	n	
Q11					2	2			1	1	n	n	
Q12					1	2	1		1	1	P	n	
Q13		1			2	1			1	1	n	n	
Q14		1			1	1	2			1	P	n	
Q15		1			1	1	2	1			P	n	
Total	6	10	5	3	18	20	10	2	6	10	P = 9 n = 6 N = 0	P = 0 n = 12 N = 3	
	Utilizaram o FJSU (Grupo 1)					Utilizaram o Unity puro (Grupo 2)							

Fonte: (O AUTOR).

Dentre as 15 afirmativas, para o grupo 1, 9 delas foram consideradas positivas, 6 neutras e 0 negativas. Já para o grupo 2, nenhuma foi considerada positiva, 12 foram consideradas neutras e 3 negativas. Esse resultado demonstra de forma global uma validação que tende ao lado positivo para o FJSU quando comparado ao Unity puro, sendo as questões individuais discutidas a seguir.

As primeiras 6 questões são relativas ao aspecto de utilidade. Na questão 1, que questiona se o desenvolvedor se acha capaz de criar um jogo usando a plataforma, foi percebido uma diferença entre as respostas, onde na primeira, o grupo 1 (FJSU) manteve duas respostas de alternativas positivas, já o grupo 2 (*Unity* puro), manteve as mesmas em alternativas negativas. Este resultado pode ser definido como positivo, visto que nessa posição os desenvolvedores, mesmo os iniciantes, consideram-se capazes de desenvolver um jogo usando a plataforma.

Já a questão 2, manteve um resultado neutro para o grupo 1, e um resultado negativo para o grupo 2. A questão verifica se o avaliador consegue criar diferentes jogos usando o *framework*. A partir da resposta, pode ser percebido que mesmo com uma avaliação neutra, os avaliadores sentiram maior confiança do que quando comparado ao grupo 2.

A questão 3 pergunta se o software utilizado possui as ferramentas necessárias para criar um jogo sério ubíquo. Os avaliadores do grupo 1 selecionaram em sua maioria alternativas positivas, enquanto a maioria daqueles do grupo 2 selecionaram “concordo”, classificando a resposta como neutra. A partir da resposta, pode ser entendido que os avaliadores entendem o FJSU como uma extensão das funcionalidades do Unity, com um foco na criação dos jogos sérios ubíquos, proporcionando ao desenvolvedor uma experiência otimizada para este fim.

Outro ponto a ser analisado é referente a questão 4, que questiona se foram percebidas muitas mensagens de erro durante a avaliação. A maioria das respostas do grupo 1 foram “não concordo” manteve a avaliação como positiva, em detrimento ao grupo que utilizou a *Unity*, que manteve a como neutra. É verificado que o resultado desta questão reflete a menor incidência de erros durante o processo de avaliação, quando comparado ao *Unity* puro.

A questão 5 refere-se à documentação do *framework*, as respostas do grupo 1 mantiveram uma divisão igual entre as possibilidades de classificação, podendo ela ser entendida como neutra, já a avaliação do grupo 2 manteve-se negativa. Pode ser entendido que a documentação do FJSU contribuiu para o aprendizado dos alunos

durante o processo de avaliação. Provavelmente pela necessidade de um maior tempo de estudo da documentação da Unity, o grupo 2 avaliou a afirmativa negativamente.

O grupo de questões do número 7 até o 12 são referentes ao aspecto de facilidade. Nesse contexto, pode ser verificado que existiu um contraste grande entre as respostas das questões 8 e 9, onde no caso da questão 8, que aborda o quanto de estudo foi necessário para utilizar a plataforma, todos os avaliadores do grupo 1 responderam não concordo para a afirmativa, sendo o resultado positivo, e os avaliadores do grupo 2 representaram uma avaliação neutra.

Já a questão 9, que questiona se foi preciso consultar a documentação muitas vezes durante o processo de avaliação, teve o resultado positivo para o grupo 1 e neutro para o grupo 2. Essas afirmativas envolvem o tempo de aprendizado da ferramenta, e o resultado pode ser considerado como positivo, visto que representa que o tempo para aprender a utilizar o FJSU é menor do que quando comparada a Unity em seu estado puro.

As questões 7 e 12 tem um resultado positivo para o grupo 1, e neutro para o grupo 2. Com isso, pode ser observado que em relação a facilidade na criação de jogos, o FJSU tende a superar o Unity. Consequentemente, os avaliadores entendem que o FJSU pode ser utilizado por iniciantes no desenvolvimento de jogos.

As questões de número 13 a 15 representam o aspecto de intenção, questionando o quanto o desenvolvedor gostou de utilizar a plataforma, se sentiu motivado e se ele se manteve concentrado durante o processo de avaliação. As questões 14 e 15, mantiveram avaliação positiva para o grupo 1 e neutra para o grupo 2, configurando que o desenvolvedor se manteve motivado e confortável utilizando o *framework*, preservando sua concentração na atividade exercida no momento em que estava avaliando o FJSU.

As questões 6, 10, 11 e 13 mantiveram a avaliação neutra por ambos os grupos. Pode ser entendido que nos aspectos avaliados, referente ao aprendizado de desenvolvimento de jogos, vídeo aulas disponíveis, interface do usuário e utilização por desenvolvedores iniciantes, o FJSU mantém-se semelhante ao Unity. Essa informação reflete que é necessária uma verificação de como melhorar essa interação com o usuário, e de que forma a utilização do mesmo pode ser facilitada por iniciantes, visto que é um dos propósitos do FJSU.

Após o processo de aplicação da avaliação, os alunos voluntários participaram

de um grupo focal, onde comentaram sobre a experiência com a ferramenta. O áudio desse processo foi gravado usando o aplicativo gravador de voz de um *smartphone* Lenovo modelo Vibe k5. Quanto ao ambiente, os alunos foram colocados em um círculo em volta do *smartphone*, e questionados sobre os pontos positivos e negativos do *framework* FJSU, quais falhas foram encontradas e o que poderia ser melhorado no processo de avaliação.

O áudio da gravação foi analisado posteriormente para obter informações individuais importantes para o amadurecimento, tanto do processo de avaliação como do próprio FJSU. Das informações analisadas, foram observados os seguintes apontamentos dos alunos:

1. “Utilizei o *framework* sem consultar a documentação, apenas me guiando pela experiência, e consegui desenvolver a maioria dos elementos elencados no caso de uso.”
2. “O fato de algumas tarefas serem realizadas no menu do FJSU e outras no do próprio *Unity* gerou uma inconsistência, dificultando o desenvolvedor a se guiar.”
3. “Os tutoriais segmentados acabam deixando limitado o processo de aprendizado, poderiam ser divididos em assuntos específicos.”
4. “O *framework* foi um facilitador, consegui desenvolver bastante coisa em pouco tempo.”
5. “Poderia ser feito um roteiro para o processo de avaliação dos desenvolvedores que irão usar o *Unity*.”

Segundo as considerações, pode ser percebido algumas potencialidades e fragilidades, tanto do FJSU como do método de avaliação. O primeiro e o quarto comentário evidenciam que o *framework* é intuitivo e útil quanto sua proposta de facilitar o desenvolvimento de jogos, visto que o desenvolvedor não precisou de instruções para desenvolver um jogo com personagem, inimigos, objetivos, pontuação e conteúdo.

O segundo comentário é referente a interface do *framework*, específico ao elemento “Canvas”, presente na *engine Unity*, e necessário para elementos da aba “UI” do FJSU. Essa questão foi levada em conta e a estrutura do *framework* foi modificada, de forma a englobar todas as funções necessárias na interface do FJSU.

Os comentários 3 e 4 são referentes ao método de avaliação. Para a avaliação

final do *framework*, foram reestruturadas as videoaulas da documentação, de forma a fugir do formato linear, permitindo que os vídeos possam ser assistidos sem uma ordem específica, abordando temas singulares. Além disso, foi criado o roteiro que o avaliador que não utilizou o *framework* usou, indicando material de aprendizado de cada elemento necessário para desenvolver o jogo no momento da avaliação.

5.3.2 Avaliação Final

Após o processo de análise dos resultados da pré-avaliação, foi planejada a execução da avaliação final, levando em conta as modificações necessárias, apontadas pelos avaliadores, tanto para a estrutura do framework quanto para a metodologia de avaliação aplicada.

Com isso, a avaliação foi aplicada em uma turma do terceiro semestre do curso de jogos digitais da Universidade Franciscana (UFN). Ao todo, foram 12 avaliadores, com uma faixa etária entre 17 e 30 anos de idade, sendo 11 homens e uma mulher, e todos possuindo algum conhecimento sobre a ferramenta *unity*, visto que estavam em processo de realização da disciplina do curso que aborda esta ferramenta.

No início da avaliação, os alunos foram separados fisicamente em dois grupos, onde cada um utilizava um computador. O grupo 1 realizou a avaliação utilizando o *framework* FJSU, e o grupo 2 utilizou o *Unity* puro. Essa separação foi feita de forma aleatória, visto que os perfis dos alunos eram semelhantes, e todos eles possuíam um nível de conhecimento correlato referente a *Unity*.

Com isso, os alunos foram informados sobre o processo a ser seguido, foi apresentado o diagrama de caso de uso do Apêndice D para ambos os grupos, sendo referente aos objetivos a serem atingidos pelo jogo desenvolvido durante a avaliação, e, foi enviado o documento presente no Apêndice E, para o grupo 2, sendo referente ao roteiro que deveria ser seguido pelo grupo, contendo *links* para explicações em vídeo e documentação referente às mecânicas que deveriam ser desenvolvidas para chegar no resultado proposto. Para o grupo 1, foi enviado o *link* para a documentação do FJSU, onde os mesmos poderiam procurar por conta própria o que fosse necessário entre os textos e vídeos disponibilizados.

Foi realizado a gravação da tela dos computadores durante todo o processo de

avaliação. O *software* utilizado foi o *CamStudio*¹³, sendo ele *software* gratuito e de código aberto, que oferece suporte à gravação de vídeo e áudio do computador. Essa gravação teve por finalidade a posterior análise da atividade dos usuários com o *framework*, e verificar as dificuldades no aprendizado da plataforma, e as potencialidades da mesma em comparação com o *Unity* puro. A gravação de quatro dos avaliadores apresentou problemas técnicos com o arquivo gerado pelo *CamStudio*, restando oito arquivos de vídeo da gravação de todo o processo, sendo quatro relativos ao grupo 1, e quatro relativos ao grupo 2.

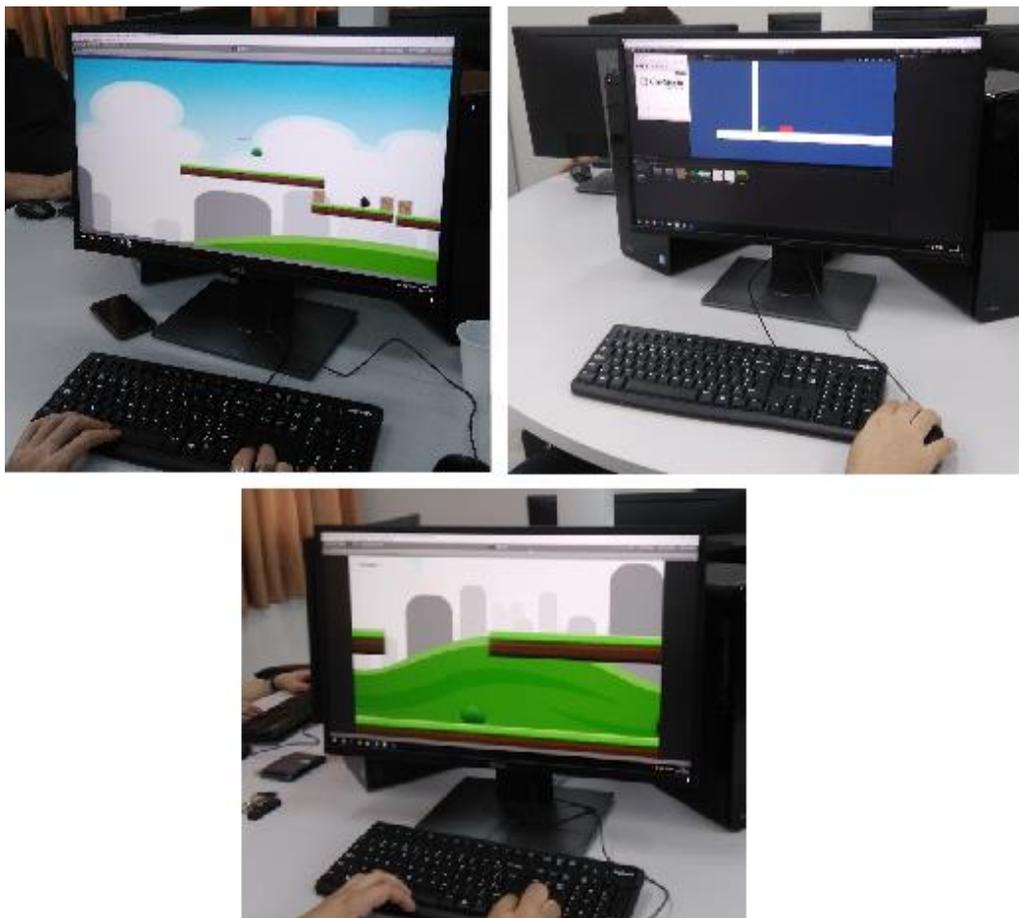
O processo durou uma hora, abrangendo tanto o aprendizado e habituação com a ferramenta quanto o próprio desenvolvimento do jogo. Foi separado o tempo de dois minutos no final da avaliação para os desenvolvedores realizarem um teste sobre o jogo desenvolvido, apenas verificando as funcionalidades criadas até o momento, sendo esse processo também gravado, para posterior análise. No final da avaliação, foi requisitado que os avaliadores preenchessem o formulário presente no Apêndice D. Na figura 5.2 podem ser visualizadas as fotos referentes ao processo de avaliação, sendo relativas a ambos os grupos.

Para a medição dos resultados obtidos com a avaliação, foi aplicado um questionário, baseando se em questões de múltipla escolha com cinco alternativas, que vão de “não concordo” até “concordo plenamente”, e que de forma sequencial possuem uma pontuação de 0 a 4. Foram utilizadas as 15 questões presentes no Apêndice D deste trabalho, as quais foram desenvolvidas com base no modelo TAM 3 de Venkatesh (2008).

Figura 5.4 - Fotos da avaliação final do FJSU

13

<https://camstudio.org/>



Fonte: (O AUTOR).

Para o cálculo da pontuação, foi definido que a soma da mesma deveria ser multiplicada por 1,7, atribuindo uma pontuação máxima de 102 pontos, mantendo uma margem de erro devido ao valor total de 100 não ser atingido devido ao resultado gerar uma dízima periódica. Dessa forma, foi catalogado a pontuação de cada resposta preenchida pelo usuário avaliador. Para as questões 4, 8 e 9, foi atribuída a pontuação inversa de forma regressiva, onde partiu-se da opção “não concordo”, relativa à pontuação 4 até a “concordo plenamente”, relativa à pontuação 0.

Após a aplicação do formulário, foram classificados os dados, sendo dividido em duas tabelas, onde a primeira é referente ao grupo 1, que utilizou o FJSU, e a segunda aqueles do grupo 2, que utilizaram o *Unity* puro. Foram adicionados aos campos da tabela a pontuação referente a resposta dos avaliadores, conforme a questão correspondente, como por ser visto no Quadro 5.3, como exemplo, o avaliador número 4 respondeu como “concordo” a questão 6 (Q6), acumulando dois pontos. No quadro, também podem ser visualizados a somatória e a média da

pontuação, sendo dividido por questões e por usuários, além da pontuação final, que avalia de forma geral o *framework*, sendo mostrada individualmente por cada usuário ou a média de todo o grupo.

Quadro 5.4 - Respostas referentes ao grupo que utilizou o FJSU

Usuário / Questão	1	2	3	4	5	6	Soma total por resposta	Médi a
Q1	4	2	1	4	4	4	19	3,17
Q2	2	1	2	2	3	1	11	1,83
Q3	2	2	2	3	3	4	16	2,67
Q4	3	5	3	3	2	4	20	3,33
Q5	3	2	2	2	2	3	14	2,33
Q6	2	2	2	2	4	3	15	2,5
Q7	2	2	2	1	2	4	13	2,17
Q8	4	3	5	3	4	3	22	3,67
Q9	3	5	5	2	3	3	21	3,5
Q10	3	2	3	1	4	4	17	2,83
Q11	2	2	2	3	4	3	16	2,67
Q12	5	2	2	3	4	4	20	3,33
Q13	3	2	3	3	3	3	17	2,83
Q14	2	2	2	2	3	2	13	2,17
Q15	2	2	2	4	4	4	18	3
SOMA	42	36	38	38	49	49		42
	71,4	61,2	64,6	64,6	83,3	83,3		71,4

Fonte: (O AUTOR).

Como pode ser visualizado no Quadro 5.4, o resultado final teve uma média de 71.4, podendo ser adjetivado como “Bom”, seguindo a escala da Figura 5.2. A pontuação mínima foi 61.2, e a máxima 83.3. As questões 1,4,8,9,12 e 15 podem ser avaliadas positivamente, pois tiveram uma pontuação média acima de 3, a qual se refere às opções de “concordo muito” e “concordo plenamente”. As questões 3,5,7,10,11,13 e 14 tiveram um resultado neutro, apresentando pontuação média maior que 2, a qual se refere a opção “concordo”. Já a questão 2 teve um resultado com a média abaixo de 2, podendo ser considerada como uma avaliação negativa.

Para a análise pontual das questões, foram utilizadas além das informações

dos questionários, a análise dos vídeos gerados pelo *software* de captura de tela como suplementação do resultado, visando analisar se a atividade dos mesmos reflete a avaliação descrita.

As questões 1 e 3 referem-se a como o avaliador interpreta o FJSU como uma ferramenta que possibilita o desenvolvimento de um jogo, se o mesmo possui as características necessárias para tal. Pode ser visto um resultado satisfatório na questão 1, que teve a pontuação média de 3,17, já na 3, é entendido como neutro, onde a média foi 2,67. O resultado das questões, indica que o desenvolvedor acredita que é possível desenvolver um jogo usando o FJSU, o resultado neutro na questão 3 indica que o *framework* possui as ferramentas necessárias para tal feito, e que as mesmas são suficientes, mas que podem ser melhoradas.

A questão 2 refere-se à possibilidade de criar diferentes tipos de jogos usando o FJSU. A mesma foi a única avaliada negativamente, com a pontuação de 1,83. Tal resultado pode ser considerado reflexo da baixa variedade de recursos disponíveis, limitando-se a jogos 2D, com o foco no gênero de plataforma. No entanto, foi analisado que alguns avaliadores conseguiram criar diferentes mecânicas usando as ferramentas disponíveis, como usar o inimigo como trampolim e fazer com que o gerador de moedas use o inimigo como ponto de referência, mostrando que a criatividade do desenvolvedor pode influenciar no resultado. Também pode ser levado em conta que o FJSU é voltado para programadores iniciantes no desenvolvimento de jogos, portanto, é interessante que o seu foco esteja na exploração da facilidade de uso pelo desenvolvedor, desta forma, simplificando a ação do usuário e diminuindo a quantidade de recursos do mesmo.

A questão 4 questiona a quantidade de erros que os avaliadores encontraram no processo. O resultado de 3,33 pontos foi considerado satisfatório, sendo reflexo da utilização do FJSU não envolver a codificação de scripts, apenas a instanciação de objetos e posicionamento dos mesmos. Através da análise dos vídeos, foi percebido que o único erro enfrentado por alguns dos desenvolvedores era relativo a não utilização do objeto “controlador”, que gerencia os demais objetos e centraliza as ações, no entanto, o erro foi resolvido quando os mesmos adicionaram o objeto ao jogo.

As questões 5 e 10 são relacionadas ao conteúdo de aprendizagem e suporte do FJSU, sendo a primeira referente a documentação, e a segunda as vídeo aulas. O resultado das mesmas foi avaliado como neutro, onde a primeira teve a pontuação

2,33 e a segunda 2,83. Tal resultado pode estar relacionado a pouca complexidade das informações presentes na documentação, referentes a cada objeto. No entanto, através dos vídeos, foi analisado que na maioria das vezes em que um avaliador sentia dificuldades, e consultava a documentação, a mesma era sanada.

A questão 6 está relacionada ao suporte que o FJSU apresenta para o aprendizado no desenvolvimento de jogos. O resultado da avaliação foi considerado neutro, com a pontuação de 2,5, tal fato foi devido ao *framework* não abordar aspectos mais avançados da programação, principalmente a criação de *scripts*. No entanto, como o *framework* é voltado para iniciantes na programação, foi realizada a escolha de trazer uma interface de desenvolvimento simplificada, sem a utilização de *scripts*.

Já a questão 7 está relacionada a facilidade que a ferramenta traz para o desenvolvimento de jogos. A alternativa foi avaliada como neutra, com a pontuação de 2,17, representando que os avaliadores tiveram um pouco de dificuldade no processo de desenvolvimento. Tal resultado pode ser entendido pelo pouco tempo de avaliação, onde os desenvolvedores tiveram tanto o processo de aprendizagem como do desenvolvimento de um jogo em 1 hora. Foi percebido pela análise dos vídeos que aqueles avaliadores que decidiram explorar o *framework* sem a ajuda da documentação, acabaram demorando alguns minutos para entender o processo de utilização do FJSU e para encontrar o objeto pretendido nos menus dele, provavelmente devido a eles estarem acostumados a utilizar a engine Unity sem modificação, e tentarem reproduzir o mesmo comportamento enquanto utilizavam o FJSU.

As questões 8 e 9 são relacionadas ao tempo de aprendizado necessário para utilizar o FJSU. As duas questões foram avaliadas positivamente, sendo a primeira com a pontuação de 3,67 e a segunda como 3,5, esse resultado reflete o que foi percebido na análise dos vídeos, onde quando um avaliador consultava a documentação ou uma parte de algum dos vídeos disponibilizados, conseguia aplicar aquele objeto estudado no jogo que estava desenvolvendo.

A questão 11 faz referência à interface do FJSU, o quanto os avaliadores conseguiram encontrar rapidamente as informações que precisavam. O resultado dessa questão foi avaliado como neutro, onde a pontuação foi de 2,67. Após análise dos vídeos, foi percebido que a pontuação é reflexo do sistema de nomes dos objetos nos menus, visto que não foi encontrado nenhuma maneira de incluir a descrição do item junto a ele na caixa de seleção, fazendo com que o avaliador não compreenda o

objetivo do mesmo de imediato, precisando do auxílio da documentação para tal, tal motivo também influenciou o resultado obtido na questão 7

A questão 12 questiona se o *framework* pode ser utilizado por pessoas iniciantes no desenvolvimento de jogos. O resultado foi entendido como satisfatório, sendo a pontuação de 3,33. Este resultado corrobora com o que foi abordado na pré-avaliação, onde alguns dos avaliadores não tinham experiência na área, e conseguiram utilizar o FJSU para desenvolver um jogo. Esse resultado vai de encontro a análise das questões 2 e 6, refletindo que o principal público alvo do *framework* são iniciantes no desenvolvimento de jogos.

As questões 13 e 14 abordam o quesito de motivação do avaliador enquanto utilizava o FJSU, as duas tiveram o resultado neutro, sendo a primeira com a pontuação média de 2,83 e a segunda 2,17, no entanto, a questão 15, que aborda a concentração do desenvolvedor enquanto utilizava o *framework*, manteve um resultado satisfatório, com a pontuação média de 3. Os resultados provavelmente estejam ligados às limitações do *framework* em produzir variados tipos de jogos, visto os avaliadores são desenvolvedores com alguma experiência, e futuros profissionais da área, que provavelmente procuram uma ferramenta com maior quantidade de recursos, mas, com maior complexidade. Já o resultado da questão 15 reflete a possibilidade de criar jogos simples em pouco tempo, visto que os desenvolvedores não precisam preocupar-se com a codificação, e podem manter sua concentração em estruturar o jogo conforme sua ideia.

O Quadro 5.5 apresenta o resultado do questionário aplicado com o grupo 2, sendo aqueles que não utilizaram o FJSU, mas sim o Unity puro, as respostas seguem o mesmo modelo do quadro anterior. Como pode ser visto, a pontuação final foi 63,2, mantendo um intermediário de “OK” e “Bom”. A menor pontuação foi 30,6 e a maior 86,7.

Dentre a média de pontos de cada questão, as questões 3,4 obtiveram um resultado satisfatório, as questões 1,2,5,6,7,8,10,11,13,14,15 mantiveram um resultado neutro, e as questões 9 e 12 obtiveram um resultado negativo. Abaixo serão analisadas as questões que apresentaram uma diferença de adjetivo em comparação com o quadro 5.3. Também foram visualizados os vídeos do grupo 2 para analisar apontamentos específicos das questões.

A questão 1 teve um resultado satisfatória apontado pelo grupo 1 (FJSU), já pelo grupo 2 (*Unity* puro), o mesmo foi neutro. Tal fato ocorreu devido ao *Unity* ser

uma ferramenta mais complexa, exigindo um pouco mais de experiência do desenvolvedor para criar um jogo.

A questão 2 teve um resultado negativo segundo o grupo 1, já segundo o grupo 2, a mesma foi considerada neutra. Essa pontuação reflete as limitações do FJSU, que engloba apenas jogos 2D, que se mostra pequeno quando comparado ao da Unity, visto que ela possui suporte tanto para jogos 3D como 2D. Da mesma forma, a questão 3 transmite a mesma informação, visto que teve resultado neutro avaliado pelo grupo 1, e satisfatório pelo grupo 2, podendo ser motivado pela gama de recursos que o Unity oferece ao desenvolvedor, como suporte a realidade virtual e realidade aumentada.

Quadro 5.5 - Respostas referentes ao grupo que utilizou o *Unity* puro

Usuário / Questão	1	2	3	4	5	6	Soma total por resposta	Médi a
Q1	1	2	2	4	4	4	17	2,833
Q2	2	2	2	4	4	3	17	2,833
Q3	4	2	2	4	4	4	20	3,333
Q4	0	4	2	4	4	4	18	3
Q5	1	2	1	4	4	3	15	2,5
Q6	2	2	1	4	4	4	17	2,833
Q7	2	1	2	4	4	3	16	2,667
Q8	0	3	2	3	2	2	12	2
Q9	0	2	1	1	0	2	6	1
Q10	1	2	3	4	4	3	17	2,833
Q11	1	1	3	2	2	3	12	2
Q12	0	2	1	3	4	1	11	1,833
Q13	1	2	2	3	4	3	15	2,5
Q14	1	1	3	4	3	2	14	2,333
Q15	2	2	1	3	4	4	16	2,667
SOMA	18	30	28	51	51	45		37,17
	30,6	51	47,6	86,7	86,7	76,5		63,2

Fonte: (O AUTOR).

As questões 8 e 9 apresentaram um grande contraste em comparação com as duas avaliações. O grupo 1 avaliou ambas como satisfatórias, já, as mesmas foram

avaliadas pelo grupo 2 como resultado neutro e negativo. Ambas as questões abordam o aprendizado da ferramenta, sendo a segunda específica com a documentação. Segundo o que foi analisado, os avaliadores não precisaram de muito tempo de estudo para aprender a usar o FJSU, diferente do que foi apontado pelo grupo que utilizou a *Unity*, visto que mesmo que eles possuíssem algum contato anterior com a ferramenta, ainda apresentavam dificuldades, e tiveram que recorrer a documentação e materiais de auxílio, como vídeos e fóruns de dúvidas sobre a *Unity*.

A questão 12 foi avaliada pelo grupo 1 como resultado satisfatório, já pelo grupo 2, a mesma teve o resultado negativo, sendo o maior contraste apresentado entre as avaliações. A questão aborda a utilização da ferramenta por iniciantes no desenvolvimento de jogos, e, como foi apontado pelo grupo 1, os mesmos consideram que o FJSU pode ser indicado para quem está começando a aprender, ou não possui conhecimento na área. Por se tratar de uma ferramenta mais abrangente, no entanto, mais complexa, a avaliação da *Unity* se manteve negativa, devido ao maior tempo para aprender a utilizar os recursos da plataforma.

A última questão que apresentou uma diferença foi a 15, na qual o grupo 1 a avaliou como satisfatória, já pelo grupo 2, a mesma foi avaliada como neutra. A questão aborda a concentração do avaliador enquanto estava desenvolvendo o jogo. Segundo o que foi analisado pelos vídeos, os avaliadores do grupo 2 perdiam muito tempo enquanto pesquisavam uma forma de programar a ação que o personagem deveria realizar, e também por realizar a troca entre a tela de criação de *scripts* e de estruturação do jogo, perdendo o foco da ação realizada naquele momento.

Além disso, os desenvolvedores acabavam por se frustrar devido a erros ou problemas enfrentados no processo de programação, isso acabava por diminuir a concentração deles na atividade realizada, sendo tanto a estruturação do jogo como a programação de *scripts*. Diferente dos avaliadores do grupo 1, que puderam colocar suas ideias em prática sem sair da tela principal do jogo, mantendo o seu foco na estruturação do mesmo. Isso incentivou uma maior exploração da plataforma, tornando a fluída, visto que, caso ocorra algum problema ou equívoco com o objeto criado, este pode ser removido rapidamente, e outro criado em seu lugar, sem a necessidade de alterações em demais objetos e códigos.

Através da observação dos vídeos de cada avaliador, foi possível comparar o tempo que cada um levou para que determinado elemento fosse inserido no jogo, como personagem, sistema de pontuação etc. Para a verificação do tempo, foi

registrado os minutos e segundos do momento em que o desenvolvedor concluiu a implementação do elemento, sendo referente ao processo todo que durou uma hora. Além disso, informações referentes ao número de mensagens de erro obtidas durante o processo e a quantidade de consultas na documentação que foi necessário fazer para chegar no resultado, nesse quesito, foi considerado a quantidade de vezes que o avaliador saiu da tela de desenvolvimento do Unity e mudou para a tela de documentação ou vídeo aula.

Quadro 5.6 - Avaliação dos Vídeos

	Avaliadores do Grupo 1				Avaliadores do Grupo 2			
	1	2	3	4	5	6	7	8
Personagem	14m 10s	09m 37s	11m 36s	11m 09s	35m 54 s	43m 5s	49m 35s	38m 43s
Objetivo	26m 5s	27m 5s	14m 0s	23m 57s	X	X	X	X
Pontuação	38m 07s	35m 10s	19m 51s	39m 21s	X	X	X	X
Adaptação	X	X	53m 2s	X	X	X	X	X
Erros	3	1	0	0	0	2	1	1
Consultas	34	28	15	3	57	51	47	53
Média	Erros	1	Consultas	20	Erros	1	Consultas	52

Fonte: (O AUTOR).

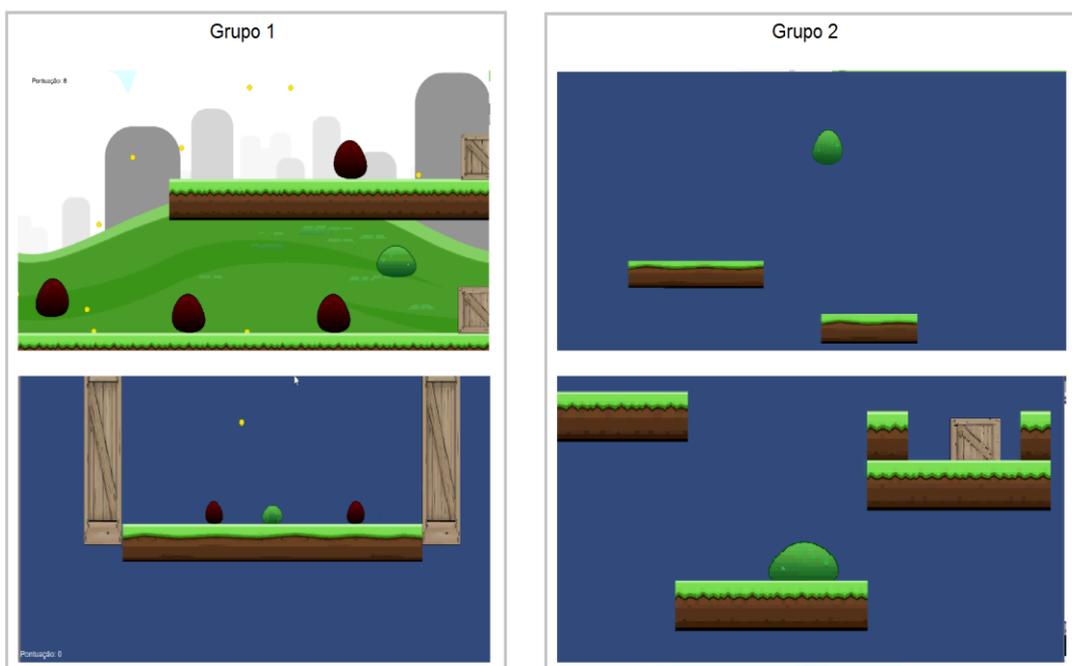
Pode ser visualizado no Quadro 5.5 o resultado da avaliação da observação dos vídeos. Dentre os elementos avaliados, o “Personagem” pode ser considerado um objeto controlável pelo jogador, que interage com os outros elementos do jogo, “Objetivo” é considerado aquilo que o jogador deve fazer para obter êxito, seja matar inimigos, sobreviver por determinado tempo, chegar até o fim da fase etc. “Pontuação” é referente ao sistema que o jogo irá pontuar o progresso do jogador, e dar o *feedback* para ele quanto a sua performance no jogo, e “Adaptação” é relacionado ao material de conteúdo adaptado ao perfil do jogador.

A partir dos resultados do Quadro 5.6, pode ser observado alguns pontos relevantes para a pesquisa. Todos os avaliadores de ambos os grupos conseguiram alcançar o desenvolvimento de um personagem controlável pelo jogador, no entanto, o maior tempo registrado pelo grupo 1, pelo avaliador 1, ainda é bem inferior aquele

menor tempo registrado pelo grupo 2, pelo avaliador 5. Foi registrado que durante esse tempo existiu o processo de aprendizagem da ferramenta, onde alguns alunos do grupo 1 tentaram usar recursos próprios do *Unity*, ao arrastar os *assets* para a tela de desenvolvimento do jogo, no entanto, quando acabaram por perceber a interface do FJSU, passaram a utilizá-la durante o restante do processo de avaliação. Tal fato pode ter ocorrido devido a eles estarem acostumados na utilização do Unity em seu estado puro, visto que é a *engine* usada nas aulas do curso dos mesmos

Quanto a criação de um objetivo para o jogo, foi verificado que todos os avaliadores do grupo 1 conseguiram definir. Sendo o primeiro a eliminação de todos os inimigos da tela, o segundo a coleta de moedas, o terceiro um labirinto cujo objetivo é a sobrevivência entre uma horda de inimigos, e o último foi uma fase de plataforma que deveria ser percorrida até o final, contendo inimigos e moedas. Dentre o espaço de tempo definido para a avaliação, nenhum dos alunos do grupo 2 conseguiu desenvolver um objetivo para o jogo, limitando-se apenas a um personagem controlável pelo jogador. Na figura 5.4, pode ser visualizado uma imagem da tela de quatro dos jogos desenvolvidos, sendo no grupo 1, as imagens relativas aos avaliadores 3 e 4, e do grupo 2 relativas aos avaliadores 5 e 6.

Figura 5.4 - Telas dos jogos criados



Fonte: (O AUTOR).

Quanto ao sistema de pontuação, foi verificado novamente que todos os avaliadores do grupo 1 conseguiram implementar, sendo do avaliador 1 por eliminar inimigos, dos avaliadores 2 e 3 pela coleta de moedas, e do avaliador 4 por ambos os sistemas. Na figura 5.4, podemos visualizar os inimigos, sendo os objetos vermelhos, e as moedas, sendo as esferas amarelas.

Já quanto a adaptação, apenas o avaliador 3 conseguiu implementar, onde conforme o número de inimigos eliminados durante o trajeto, a velocidade deles era aumentada, forçando o jogador a escapar do labirinto evitando pular em cima dos mesmos, para que a dificuldade não venha a subir de nível. O motivo para a não implementação do sistema de adaptação pelos outros avaliadores do grupo 1 foi a falta de tempo, visto que o avaliador 3 atingiu o desenvolvimento do personagem, objetivo e sistema de pontuação do jogo em um espaço de tempo muito menor se comparado aos outros avaliadores.

Ainda assim, o sistema de adaptação foi aplicado pelo avaliador 3 após os 50 minutos do decorrer do processo, representando que existiu uma dificuldade para chegar a um conceito de adaptação durante o desenvolvimento, mas que o mesmo foi alcançado.

Quanto a quantidade de erros observada durante o processo de avaliação, ambos os grupos obtiveram a mesma média de 1 erro. Foi percebido que no grupo 1, todos os erros aconteceram devido a não utilização do objeto controlador, que gerencia a troca de informações entre os objetos do jogo. No momento em que o avaliador percebeu a falta do mesmo, e o inseriu na cena do jogo, a mensagem de erro desfez-se. Esse fato ocorreu devido ao avaliador não verificar na documentação que o objeto é essencial na criação dos jogos usando o FJSU, sendo tal informação presente no primeiro parágrafo da documentação.

Já no grupo 2, as mensagens de erro foram variadas, entre iniciar o jogo sem selecionar o objeto do personagem no *script* ou digitá-lo em uma sintaxe diferente da aceita. Por tratar-se de um problema que demanda um maior conhecimento da linguagem de programação C#, utilizada pelo *Unity*, o tratamento deste tipo de erro é mais trabalhoso e requer uma análise maior do ocorrido, e estudo da solução, quando comparado com o erro obtido pelo grupo 1.

Quanto às consultas, foi percebido a melhor forma que os desenvolvedores encontraram para aprender a utilizar a ferramenta. A média de consultas do grupo 1 foi de 20, e a do grupo 2 foi de 52, mostrando que o grupo que utilizou o FJSU precisou

consultar menos da documentação para realizar as atividades propostas. Dentre esse resultado, foi analisado que os avaliadores 1,2,5,6,7 e 8 acabaram por utilizar frequentemente as videoaulas e a documentação para a criação do seu jogo, já os avaliadores 3 e 4 buscaram explorar por conta própria os recursos da ferramenta, com pouco contato com a documentação, apenas para sanar algumas dúvidas no início do desenvolvimento. Isso refletiu no baixo número de consultas, e também de mensagens de erro, visto que os avaliadores buscaram informações sobre os objetos antes de utilizá-los de fato.

Pode ser interpretado que a documentação possui clareza suficiente para que os desenvolvedores consigam entender o funcionamento do objeto e o implementem no jogo sem muito contato com a mesma, e, como foi visto, aqueles avaliadores que acabaram explorando o *framework* após uma leitura inicial da documentação, e, através dos testes, acabaram por descobrir o funcionamento de cada objeto, conseguiram criar os elementos do jogo em menor tempo se comparado aos outros avaliadores, mantendo um maior foco no que estava sendo desenvolvido.

5.4 CONSIDERAÇÕES DA AVALIAÇÃO

O processo de avaliação do FJSU foi realizado em dois momentos, no primeiro, enquanto em formato de caixa branca. Essa avaliação foi realizada por oito alunos que atuaram como desenvolvedores voluntários, e seguiram um roteiro usando a ferramenta, e, após o preenchimento de um formulário com questões relativas à experiência. Neste momento, foi possível visualizar os aspectos essenciais dos *frameworks* que deveriam ser melhorados, onde nesse ponto foi visto que a interação do desenvolvedor com a plataforma estava se mostrando dificultosa por iniciantes, fazendo-se necessário que uma interface de fácil interação fosse implementada, tornando o um *framework* de caixa cinza.

No segundo momento, enquanto o FJSU já estava no formato de caixa cinza, foi realizado um processo de pré-avaliação, para mensurar se os parâmetros usados na avaliação estavam de acordo com os objetivos da mesma. Para tal, foi realizado o mesmo processo utilizado anteriormente, onde os desenvolvedores deveriam seguir um roteiro pré-determinado, e, no final dele, preencher um formulário para a obtenção dos dados relativo à experiência. No entanto, os alunos foram divididos em dois grupos neste momento, onde metade deles utilizaram o FJSU, e a outra usou o *Unity*

puro, a fim de comparar as duas plataformas. Este momento foi importante para perceber que a etapa de avaliação deveria amadurecer em alguns aspectos antes de ser aplicada, tal como, algumas modificações no *framework* e na documentação precisaram ser feitas.

A avaliação final foi realizada com uma turma de alunos do curso superior de Jogos Digitais. Nesta etapa, também foi utilizado um roteiro, as turmas foram divididas da mesma forma e foi realizado o preenchimento de um formulário na parte final da avaliação, no entanto, durante todo o processo, foi realizada a gravação da tela dos computadores dos avaliadores através de um *software*. Este registro foi de extrema importância para a análise dos resultados da avaliação final, pois permitiu quantificar elementos da interação dos desenvolvedores enquanto utilizavam a ferramenta, como consultas a documentação e quantidade de erros. Além disso, os mesmos foram necessários para verificar a congruência das informações obtidas pelo questionário.

Os resultados da avaliação final mostraram-se satisfatórios, foi percebido que um desenvolvedor iniciante consegue utilizar o FJSU para criar um jogo simples em pouco tempo. A comparação com o *Unity* em seu estado puro permitiu resultados concisos quanto ao tempo economizado para a criação dos elementos do jogo, e também quanto ao tempo de aprendizado da plataforma, mostrando que o mesmo além de ser otimizado para o uso por iniciantes, também permite que elementos fundamentais de um jogo sério ubíquo possam ser desenvolvidos rapidamente.

6. CONCLUSÃO

Neste capítulo é apresentado o resumo do trabalho, evidenciando as principais contribuições da pesquisa com a área de desenvolvimento de jogos e tecnologias educacionais, além dos resultados alcançados no processo, apontando as dificuldades enfrentadas durante a pesquisa e as principais limitações que foram encontradas na trajetória da mesma. Também são apresentados como trabalhos futuros algumas possibilidades de extensão da pesquisa. Além disso, no final do capítulo, são apresentadas as considerações finais deste trabalho.

6.1 RESUMO DO TRABALHO

Com o avanço das tecnologias, passando pela computação móvel até a atual computação ubíqua, o modo como os jogadores interagem com o jogo mudou, atingindo um patamar onde muitas vezes é difícil dividir o real do virtual, mesclando elementos presentes nos jogos, como desafios, pontuação, recompensas, com atividades reais, como uma corrida, dança ou o aprendizado. Tais jogos que atingem esse patamar de interação com o jogador, onde ele não percebe os sistemas computacionais que fazem parte do processo do jogo, e trazem uma premissa de aprendizado ou treinamento, são chamados de jogos sérios ubíquos.

Esse aspecto transformou não só o jogador, mas também os profissionais das áreas da educação, que precisam adaptar seu modo de atuar conforme a demanda tecnológica da geração atual, visto a influência desta nos alunos da atual geração. Com isso, se faz necessária uma ferramenta que auxilie tanto alunos de design de jogos digitais, a como os professores, sendo simples o suficiente para que um professor com pouco conhecimento das tecnologias ubíquas e em programação consiga desenvolver um jogo, e também que sirva como um facilitador para a tarefa do designer de jogos digitais, permitindo o reuso de componentes durante o desenvolvimento do jogo sério ubíquo.

Os resultados iniciais do presente estudo permitiram analisar como está sendo realizado o desenvolvimento, aplicação e avaliação dos jogos sérios ubíquos através de um processo de mapeamento sistemático, proporcionando o embasamento necessário para a análise de domínio. O estudo resultante do processo de mapeamento sistemático foi publicado na Revista Novas Tecnologias na Educação

(RENOTE), em sua segunda edição do ano de 2017.

Tendo em vista o desafio no desenvolvimento de jogos sérios ubíquos, foi criado um *framework* de caixa cinza denominado FJSU, tendo por objetivo oferecer suporte ao desenvolvedor quanto a criação de jogos desse tipo, através da reutilização de objetos e funções, evitando o contato do desenvolvedor com os scripts no momento da estruturação do jogo, agilizando o processo de design do jogo e tornando o acessível para iniciantes na programação.

O *framework* utilizou como base a *engine* de jogos *Unity*, e usou como plataforma de desenvolvimento os recursos oferecidos pela própria, onde sua interface foi modificada através de scripts na linguagem C#, e na criação de objetos, de forma a atender as necessidades do FJSU em facilitar o desenvolvimento de jogos ubíquos.

O FJSU foi avaliado por uma turma de alunos do terceiro semestre do curso superior de Jogos Digitais da UFN, na cidade de Santa Maria no Rio Grande do Sul. Em sua avaliação, o mesmo foi comparado com a *Unity* em seu formato original, e mostrou resultados positivos quanto a facilidade de utilização por iniciantes, menor tempo para realizar o desenvolvimento e também menor tempo para o aprendizado, no entanto, os resultados negativos foram relacionados às limitações da plataforma e a possibilidade de criar diferentes tipos de jogos usando o *framework*.

6.2 CONTRIBUIÇÕES

O desenvolvimento de jogos sérios ubíquos pode ser um desafio para professores e *designers* de *games* iniciantes. A utilização de um *framework* de caixa cinza como ferramenta simplifica e agiliza o processo de desenvolvimento dos mesmos, visto que suprime a etapa de codificação e permite ao desenvolvedor manter o foco na estrutura do jogo.

Como principais contribuições desta dissertação, podem ser destacados:

- Criação do *framework* de caixa cinza FJSU, que utiliza a *engine* Unity como base, e oferece aos professores e designer de jogos digitais iniciante uma ferramenta para a criação de jogos sérios ubíquos de forma simplificada, abstraindo a programação por scripts do processo, e mantendo sua estrutura por reuso de objetos.
- Realização do mapeamento sistemático referente a jogos sérios ubíquos, observando o panorama atual da pesquisa e os principais conceitos e aplicações em ambientes

sérios de aprendizagem e treinamento;

- Desenvolvimento da estrutura de classes e objetos que possibilitaram a criação do FJSU enquanto mantinha-se como *framework* de caixa branca;
- Utilização e avaliação do FJSU por uma turma de 12 alunos do curso de Jogos Digitais da Universidade Franciscana de Santa Maria, validando de forma prática a proposta da ferramenta;
- Publicações
 - 1 artigo completo publicado em periódico nacional (DA SILVEIRA JÚNIOR e MEDINA., 2017).

6.3 DIFICULDADES E LIMITAÇÕES

Dentre as etapas que permearam o desenvolvimento do *framework* FJSU, foram encontradas algumas dificuldades e desafios no processo, além de problemas limitadores que precisaram ser contornados para que o resultado esperado fosse atingido. A existência de tais dificuldades e limitações foi percebida devido a necessidade de explorar a grande gama de recursos que a *engine Unity* oferece, e perceber qual a melhor maneira de criar os módulos do *framework*, de forma que facilite o trabalho do desenvolvedor. Dentro desse contexto, foram encontrados os problemas citados abaixo:

- Devido a *Unity* utilizar uma versão própria da linguagem C#, não foi possível que o *framework* se adaptasse a outras *engines* enquanto mantinha seu formato de caixa branca, portanto, este aspecto precisou ser ignorado como objetivo da versão final do FJSU;
- A importação da biblioteca de scripts do FJSU, e a utilização dos seus métodos enquanto mantinha-se no formato de caixa branca demonstrou-se trabalhosa em demasia nas primeiras avaliações. Partindo deste problema, foi visto que a proposta se encaixaria melhor em um modelo de caixa cinza, onde para isso, foi utilizado a biblioteca *Unity Editor*, que traz ferramentas para modular a interface da *Unity*;
- A criação dos objetos presentes na versão final do FJSU foi pensado de uma forma que exija o mínimo de interação do desenvolvedor, no entanto, foi necessário uma adaptação da forma como os *scripts* são criados, onde o nome de determinados objetos é essencial, e, para efetivo funcionamento deles, não

pode ser mudada, como é o caso do Personagem e do Controlador;

- As funcionalidades referentes a utilização de realidade virtual e realidade aumentada não puderam ser implementadas na versão atual do FJSU, visto que tornaria o processo de desenvolvimento mais complexo, e já existem bibliotecas prontas e gratuitas, como a Vuforia¹⁴ e o Google VR¹⁵ para tal;

6.4 TRABALHOS FUTUROS

Durante o desenvolvimento do FJSU, foram encontradas algumas necessidades e possibilidades de extensão do *framework*, que não puderam ser implementadas tanto devido à complexidade da atual etapa do trabalho como do tempo necessário para que as mesmas fossem implantadas no *framework*. Portanto, ficam as mesmas como oportunidades de pesquisa listadas a seguir:

- Aumento dos tipos de jogos suportados pelo FJSU, como *puzzles*, estratégia, RPG e *shooters*;
- Explorar as possibilidades da adaptação conforme o perfil do usuário em práticas educacionais, e investigar a implementação de conteúdo conforme estilos de aprendizagem;
- Considerar novas avaliações utilizando docentes como desenvolvedores, a fim de verificar sua aplicação na área educacional;
- Explorar novas funcionalidades para os sensores dos dispositivos móveis, como giroscópio, gps, câmera e estender as bibliotecas existentes;

6.5 CONSIDERAÇÕES FINAIS

A atual era da computação ubíqua proporciona oportunidades e desafios no desenvolvimento de jogos, desde a grande variedade de dispositivos disponíveis atualmente até a gama de sensores e tecnologias diferentes que envolvem a comunicação dos mesmos que devem ser entendidas. Em virtude disso, os processos de desenvolvimento estão se tornando cada vez mais complexos, atribuindo ao desenvolvedor a necessidade de conhecer as tecnologias e procedimentos que envolvem a criação de um jogo ubíquo. Para tanto, os frameworks surgem como uma

¹⁴ <https://www.vuforia.com/>

¹⁵ <https://developers.google.com/vr/develop/unity/get-started-android>

alternativa viável para a criação de jogos nessa geração, visto que facilitam e agilizam o trabalho do desenvolvedor, que pode manter seu foco no próprio design do jogo.

Seguindo esta perspectiva, esta dissertação apresentou o processo de desenvolvimento e avaliação de um *framework* de caixa cinza para o desenvolvimento de jogos sérios ubíquos, denominado FJSU, onde seu módulo de aplicação foi criado com base na *engine Unity*, e o módulo web como um *web service* na linguagem de programação PHP, utilizando o banco de dados *mySQL*.

Foi demonstrado que o *framework* desenvolvido neste trabalho surge como uma alternativa viável para facilitar o desenvolvimento de jogos ubíquos para profissionais ligados à área de tecnologia.

Foi verificado a necessidade de avaliações com professores atuando como desenvolvedores, a fim de evidenciar se a ferramenta atinge um nível de usabilidade compatível com a proposta de um *framework* onde os docentes podem criar seus próprios jogos sérios ubíquos. A implementação do FJSU e às avaliações realizadas acabaram ocupando muito do tempo disponível para a realização do projeto, tornando inviável uma nova avaliação, portanto, a mesma foi categorizada como trabalho futuro.

As avaliações iniciais e casos de teste realizados foram importantes para a formatação do *framework*, para que ele atendesse as necessidades dos desenvolvedores de maneira simplificada. A avaliação final demonstrou que foi possível chegar a um resultado satisfatório quanto à pretensão do produto desta dissertação, verificando que o FJSU contribuiu para os desenvolvedores de jogos em seu objetivo de simplificar a criação deles, e tornar o processo de desenvolvimento dinâmico.

REFERÊNCIAS

- AL OSMAN, H.; DONG, H.; & EL SADDIK, A. Ubiquitous biofeedback serious game for stress management. **IEEE Access**, v. 4, p. 1274-1286, 2016.
- ALEXANDER, C.; ISHIKAWA, S.; SILVERSTEIN, M. **Uma linguagem de padrões**. Porto Alegre: Bookman, 2013, 1215 p.
- ARANGO, G. **A brief introduction to domain analysis**. In: Proceedings of the 1994 ACM symposium on Applied computing. ACM, 1994. p. 42-46.
- ARNOLD, S., FUJIMA, J., KARSTEN, A., & SIMEIT, H. **Adaptive behavior with user modeling and storyboarding in serious games**. In *Signal-Image Technology & Internet-Based Systems (SITIS), 2013 International Conference on* (pp. 345-350). IEEE, 2013.
- BANOS, Oresti et al. Physiodroid: Combining wearable health sensors and mobile devices for a ubiquitous, continuous, and personal monitoring. **The Scientific World Journal**, v. 2014, p. 1-11, set. 2014.
- BARTNECK, C. et al. The robot engine—**Making the unity 3D game engine work for HRI**. In: 2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN). IEEE, 2015. p. 431-437. 2015
- BATTISTELLA, P. E. . **ENgAGED: Um processo de desenvolvimento de jogos para ensino em computação**. Tese (Doutorado em Ciência da Computação). Universidade Federal de Santa Catarina, Santa Catarina. 2016.
- BRANDT, A J. . **Location-Based Tower Defense-Designing an Outdoor Exergame**. 2016. Dissertação (Mestrado em Ciência da Computação). Norwegian University of Science and Technology, Trondheim, 2016.
- BRAGA, R. T. V.. **Um processo para construção e instanciação de frameworks baseados em uma linguagem de padrões para um domínio específico**. Tese (Doutorado em Ciência da Computação e Matemática Computacional). Universidade de São Paulo, São Paulo, 2002.
- BOYLE, E. A. et al. **An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games**. *Computers & Education*, v. 94, p. 178-192, 2016.
- BUZETO, F. N.. **Jogos ubíquos reconfiguráveis: da concepção à construção**. Tese (Doutorado em Ciência da Computação). Universidade de Brasília, Brasília, 2015.
- CARVALHO, Maira B. et al. A case study on service-oriented architecture for serious games. **Entertainment Computing**, v. 6, p. 1-10, 2015.
- COSTA SEGUNDO, R. M.. **Athus: um framework para o desenvolvimento de jogos para TV Digital utilizando Ginga**. Dissertação (Mestrado em Informática). Universidade Federal da Paraíba, João Pessoa, 2011.

DA SILVEIRA JÚNIOR, G; MEDINA, R. D. Jogos Sérios Ubíquos: Um mapeamento Sistemático. **RENOTE**, v. 15, n. 2., 2017.

DELAMARO, M.; JINO, M.; MALDONADO, J. **Introdução ao teste de software**. Elsevier Brasil, 2017.

DE SENA, Samara et al. **Aprendizagem baseada em jogos digitais: a contribuição dos jogos epistêmicos na geração de novos conhecimentos**. **RENOTE**, v. 14, n. 1, 2016.

DUARTE, N., POSTOLACHE, O., & SCHARCANSKI, J. **KSGphysio-Kinect serious game for physiotherapy**. In *Electrical and Power Engineering (EPE), 2014 International Conference and Exposition on* (pp. 606-611). IEEE, 2014.

FAYAD, M. E.; SCHMIDT, D.; AND JOHNSON, R. **Building Application Frameworks: Object-Oriented Foundations of Framework Design**, John Wiley & Sons, New York, September 1999.

GARTNER. **5 Trends emerge in gartner hype cycle for emerging technologies 2018**. Disponível em <<https://www.gartner.com>> Acesso em: 21/11/2018. 2018

GARCEZ, A.; DUARTE, R.; EISENBERG, Z. **Produção e análise de vídeograções em pesquisas qualitativas**. *Educação e Pesquisa*, v. 37, n. 2, p. 249-261, 2011.

GREGORY J. **Game engine architecture**. crc Press; 2009 Jun 15.

GICQUEL, P. Y., LENNE, D., & MOULIN, C. **Design and use of CALM: An ubiquitous environment for mobile learning during museum visit**. In *Digital Heritage International Congress (DigitalHeritage)* (Vol. 2, pp. 645-652). IEEE. 2013

GIRARD, C.; ECALLE, J.; & MAGNAN, A. **Serious games as new educational tools: how effective are they?** A meta-analysis of recent studies. *Journal of Computer Assisted Learning*, 29(3), 207-219. 2013

GRAMMATIKOPOULOU, A. et al. **An adaptive framework for the creation of exergames for intangible cultural heritage (ICH) education**. *Journal of Computers in Education*, p. 1-34, 2018.

HAMDAOUI, N.; IDRISSE, M. K.; & BENNANI, S. **AMEG: Adaptive mechanism for educational games based on IMSLD and artificial intelligence**. In *Intelligent Systems: Theories and Applications (SITA), 2015 10th International Conference on* (pp. 1-6). IEEE. 2015

HOLM, J.; KARI, L. **Designing ActionTrack: A state-of-the-art authoring tool for location-based games and other activities**. *Information Visualisation (IV), 2014 18th International Conference on*. IEEE, 2014.

HWANG, GJ. et al. **A context-aware ubiquitous learning environment for conducting complex science experiments**. *Computers & Education*, v. 53, n. 2, p. 402-413, 2009.

HWANG, G.J, et al. **Development of a personalized educational computer game based on students' learning styles.** Educational Technology Research and Development. v. 60, n. 4, p. 623-638, ago. 2012.

HWANG, G.J et al. **Effects of an augmented reality-based educational game on students' learning achievements and attitudes in real-world observations.** Interactive Learning Environments, v. 24, n. 8, p. 1895-1906, 2016.

JOHNSON, R. E., & FOOTE, B. Designing reusable classes. **Journal of object-oriented programming**, v. 1, n. 2, p. 22-35, 1988.

KAPP, K. M. **The Gamification of Learning and Instruction: Game-based methods and strategies for training and education.** San Francisco: John Wiley & Sons, 2012.

KITCHENHAM, B. **Procedures for performing systematic reviews.** Keele, UK, Keele University, 33 (2004), 1-26. 2004.

KITTL, C.; EDEGGER, F.; PETROVIC, O. **Learning by pervasive gaming: An empirical study.** Innovative Mobile Learning: Techniques and Technologies: Techniques and Technologies.:60-82. 2009.

KLOCK, A.. **Análise da influência da gamificação na interação, na comunicação e no desempenho dos estudantes em um sistema de hipermídia adaptativo educacional.** Dissertação (Mestrado em Computação Aplicada). Universidade do Estado de Santa Catarina, Joinville, 2017.

KLOPFER, E.; SHELDON, J.; PERRY, J.; CHEN, V. H. Ubiquitous games for learning (UbiqGames). **Journal of Computer Assisted Learning**, v. 28, p. 465-476, 2012.

KRUMM J, EDITOR. **Ubiquitous computing fundamentals.** CRC Press; 2016 Apr 19.

KÖHLMANN, W., ZENDER, R., & LUCKE, U. **FreshUP—implementation and evaluation of a pervasive game for freshmen.** In *Pervasive computing and communications workshops (PERCOM workshops), 2012 IEEE international conference on* (pp. 691-696). IEEE, 2012.

KUBO, M. **FMMG: um framework para jogos multiplayer móveis.** Tese (Doutorado em Engenharia). Universidade de São Paulo, São Paulo, 2006.

LIKERT, R. A technique for the measurement of attitudes. **Archives in Psychology**, v. 140, p. 1-55, 1932.

LINDBERG, R.; SEO, J.; LAINE, T. **Enhancing physical education with exergames and wearable technology.** IEEE Transactions on Learning Technologies, v. 9, n. 4, p. 328-341, 2016.

LOPES R. R.; BIDARRA, R. **Adaptivity challenges in games and simulations:a survey,** IEEE Transactions on Computational Intelligence and AI in Games, vol. 3, no. 2, pp. 85 –99, june 2011

LOPES, R.; EISEMANN, E.; BIDARRA, R. **Authoring adaptive game world generation**. IEEE Transactions on Computational Intelligence and AI in Games, 2017.

LUMPOON P. N.; THIENGBURANATHUM, P. **Effects of integrating a mobile game-based learning framework in a cultural tourism setting**. In Software, Knowledge, Information Management & Applications (SKIMA), 10th International Conference on 2016 Dec 15 (pp. 281-285). IEEE. 2016.

MARKIEWICZ, M. E., & DE LUCENA, C. J. Object oriented framework development. **Crossroads**, v. 7, n. 4, p. 3-9, 2001.

MCGONIGAL, J. E. (2006). **This might be a game: ubiquitous play and performance at the turn of the twenty-first century**. Tese (Doutorado em Filosofia). University of California, Berkeley, 2006.

MORTARA, M., CATALANO, C. E., BELLOTTI, F., FIUCCI, G., HOURY-PANCHETTI, M., & PETRIDIS, P. Learning cultural heritage by serious games. **Journal of Cultural Heritage**, v. 15, n. 3, p. 318-325, 2014.

MÜLLER, A. CRUZ, D. **Formação docente para inclusão de games na educação básica: relato de uma experiência= Training teachers for inclusion of games in basic education: an experience report**. *Obra digital: revista de comunicação*, n. 10, 2016.

NUNES, F. B. (2014). **UVLEQoC: A Ubiquitous Virtual Learning Environment with Quality of Context**. Dissertação (Mestrado em Informática). Universidade Federal de Santa Maria, Santa Maria, 2014.

OLIVEIRA, W., SILVA, T. R., & ARANHA, E. **Aplicação de jogos adaptativos na educação: uma revisão sistemática da literatura**. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 27, No. 1, p. 587). 2016

PIMENTA, M, et al. **A game engine for building ubigames**. Proceedings of the 13th Annual Workshop on Network and Systems Support for Games. IEEE Press, 2014.

POSTOLACHE, O., CARY, F., GIRÃO, P. S., & DUARTE, N.. **Physiotherapy assessment based on Kinect and mobile APPs**. In *Information, Intelligence, Systems and Applications (IISA), 2015 6th International Conference on* (pp. 1-6). IEEE. 2015

RABARI, Chirag; STORPER, Michael. The digital skin of cities: urban theory and research in the age of the sensed and metered city, ubiquitous computing and big data. **Cambridge Journal of Regions, Economy and Society**, v. 8, n. 1, p. 27-42, 2014.

RODRIGUES, W. A. (2016). **Tardigrade: um framework android para desenvolvimento de jogos de cartas ubíquos**. Dissertação (Mestrado em Ciência da Computação). Universidade Federal do Ceará, Fortaleza, 2016.

SAJJADI, P.; VAN BROECKHOVEN, F.; DE TROYER, O. **Dynamically adaptive educational games: A new perspective**. In *International Conference on Serious Games* (pp. 71-76). Springer, Cham. 2014

SALIM, F.; HAQUE, U. Urban computing in the wild: A survey on large scale participation and citizen engagement with ubiquitous computing, cyber physical systems, and Internet of Things. **International Journal of Human-Computer Studies**, v. 81, p. 31-48, 2015.

SAWYER, R. et al. **Enhancing student models in game-based learning with facial expression recognition**. In: Proceedings of the 25th conference on user modeling, adaptation and personalization. ACM, 2017. p. 192-201.

SILVA, M. P.. **Inteligência artificial adaptativa para ajuste dinâmico de dificuldade em jogos digitais**. Dissertação (Mestrado em Ciência da Computação). Universidade Federal de Minas Gerais, Belo Horizonte, 2015.

SOFLANO, M.; CONNOLLY, T. M.; HAINEY, T. **An application of adaptive games-based learning based on learning style to teach SQL**. *Computers & Education*, 86, 192-211. 2015

SZEGLETES, L.; FORSTNER, B. **Reusable framework for the development of adaptive games**. In Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference on (pp. 601-606). IEEE. 2013.

TALIGENT WHITE PAPERS. **Building Object Oriented Framework**, inc, 1994. Disponível em <<https://lhcb-comp.web.cern.ch/lhcb-comp/Components/postscript/buildingoo.pdf>> Acesso em: 10 de out. 2017.

LICOPPE, C.; LEVALLOIS, C. B. **Configuring Commuters' Accessibility to Multimedia Mobile Services: The Case of Bluetooth 'Augmented' Advertising in the Paris Metro**. In: Mobilities: new perspectives on transport and society, pg 245 - 270, Routledge, 2016

WAZLAWICK, R. S. **Metodologia da Pesquisa para Ciência da Computação**. Rio de Janeiro: Elsevier, 2009. 6a reimpressão.

WHITEHEAD, A et al. **Exergame effectiveness: what the numbers can tell us**. In: Proceedings of the 5th ACM SIGGRAPH Symposium on Video Games. ACM, 2010. p. 55-62.

VALENTE, L; FEIJÓ, B; DO PRADO LEITE, J. C. S. Mapping quality requirements for pervasive mobile games. **Requirements Engineering**, v. 22, n. 1, p. 137-165, 2017.

VANDEWAETERE, M.; CORNILLIE, F.; CLAREBOUT, G.; DESMET, P. Adaptivity in educational games: Including player and gameplay characteristics. **International Journal of Higher Education**, v. 2, n. 2, 2013.

VAYANOU, M.; KARVOUNIS, M.; KATIFORI, A.; KYRIAKIDI, M.; ROUSSOU, M.; IOANNIDIS, Y. E. **The CHESS Project: Adaptive Personalized Storytelling Experiences in Museums**. In *UMAP Workshops*. 2014.

VENKATESH, V.; BALA, H. Technology acceptance model 3 and a research agenda on interventions. **Decision sciences**, v. 39, n. 2, p. 273-315, 2008.

XIE, J. **Research on key technologies base Unity3D game engine**. In Computer Science & Education (ICCSE), 2012 7th International Conference on (pp. 695-699). IEEE. 2012

YAHYA, W.; NOOR, N. **Decision Support System for Learning Disabilities Children in Detecting Visual-Auditory-Kinesthetic Learning Style**. In *The 7th International Conference on Information Technology*. 2015

YANG, Y.; LEUNG, H.; LIU, Z.; ZHAN, Y.; ZENG, L. **Sequential-Global Learning Style Detection Based on Users' Navigation Patterns in the Prerequisite Structure**. In *International Conference on Hybrid Learning and Continuing Education* (pp. 283-293). Springer International Publishing. 2015

APÊNDICE A - PROCESSOS DO LEVANTAMENTO DE REQUISITOS

1. Introdução

O seguinte processo serve como suporte ao levantamento de requisitos de um jogo sério ubíquo, sendo preenchido com os dados sugeridos pelo desenvolvedor no momento inicial da criação do jogo.

Foram definidas quatro categorias de processos principais, com seus devidos subprocessos, que englobam o levantamento de requisitos, sendo relativas ao Dispositivo, Conexão, Estilo do jogador e Experiência do jogador. Essa definição foi dada para que os aspectos de um jogo ubíquo possam ser atingidos, essa estrutura pode ser visualizada no quadro 1.

Quadro 1 Estrutura dos processos

Dispositivo	1. Interoperabilidade 2. Gráfico
Conexão	3. Uso de Internet 4. Sensores 5. Banco de Dados
Estilo do Jogador	6. Conteúdo 7. Forma de Adaptação
Experiência do Jogador	8. Feedback 9. Motivação

Fonte: (O AUTOR).

2. Dispositivo

Quando tratamos de jogos ubíquos, inicialmente, deve ser feito o levantamento de requisitos quanto a definição das tecnologias que serão utilizadas, sejam os tipos de dispositivos alvo e sistemas.

Quadro 2.1 Processo Dispositivo

Subprocesso	Pergunta	Síntese
Interoperabilidade	<i>P1. Quais dispositivos o jogo busca atender?</i>	É preciso que seja definido se o jogo será voltado para dispositivos móveis, computadores de mesa, web, dispositivos vestíveis ou realidade virtual.
	<i>P2. Qual ferramenta será usada para o desenvolvimento?</i>	Com a definição das questões anteriores, pode ser definido qual <i>engine</i> será usada para o desenvolvimento do jogo proposto, verificando se a mesma abrange suporte suficiente para os sensores e tecnologias que fazem parte dos dispositivos alvo.
Gráfico	<i>P3. O jogo será 2D ou 3D?</i>	A escolha da dimensão da ambientação do jogo influi nas características de processamento gráfico, portanto, visando a portabilidade e atingir um maior número de dispositivos mais antigos, pode ser desenvolvido um jogo 2D, já quanto a proposta de imersão, e tecnologias de realidade virtual, um ambiente 3D pode ser mais interessante.

Fonte: (O AUTOR).

3 Conexão

A definição de conexão está ligada a presença do aspecto de mobilidade do jogo, podendo ela ser contínua ou intermitente, definindo como será feito o processo de salvamento dos dados e troca de informações.

Quadro 3.1 Processo Conexão

Subprocesso	Pergunta	Síntese
Uso de Internet	<i>P4. É necessário conexão de alta velocidade?</i>	Jogos multiplayer costumam requisitar conexão de alta velocidade, para que não ocorram perdas de performance e atraso na troca de informações.
	<i>P5. A conexão deverá ser contínua ou poderá ser intermitente?</i>	Essa questão é relativa caso o jogador perca a conexão por alguns momentos. Quando é utilizado um banco de dados externo, os dados podem ser salvos no dispositivo e assim que ele reconectar, os mesmos serem enviados, viabilizando a conexão intermitente.
Sensores	<i>P6. Quais tipos de sensores serão utilizados?</i>	Referente às tecnologias que capturam as informações de contexto, como o GPS, pedômetro, sensor cardíaco. Deve ser estudado o objetivo do jogo para verificar qual o melhor sensor a ser utilizado na situação.
Banco de Dados	<i>P7. O banco de dados será interno ou externo?</i>	Um banco de dados externo possibilita que os dados possam ser compartilhados e alimentados por diferentes e diversos dispositivos, além de proporcionar que não ocorra a perda dos mesmos com algum problema no dispositivo. A utilização de um banco de dados interno é propícia caso a inexigibilidade do uso de internet.

Fonte: (O AUTOR).

4. Estilo do Jogador

O estilo do jogador define a forma que ele está mais adaptado a aprender ou realizar uma atividade, partindo disso, é considerada a adaptação do conteúdo para esse formato. Este processo está ligado aos padrões “criar / buscar perfil de usuário”, “buscar contexto”, “tratar contexto” e “salvar perfil”.

Quadro 4.1 Processo Estilo do Jogador

Subprocesso	Pergunta	Síntese
Conteúdo	<i>P8. Qual o objetivo do jogo?</i>	O conteúdo do jogo deve estar alinhado com o objetivo do mesmo, visto que, no caso de um jogo sério, o mesmo é voltado para a educação ou treinamento, portanto, se faz necessário que não existam dúvidas quanto aquele conhecimento que será absorvido pelo jogador.
	<i>P9. Qual o gênero do jogo?</i>	A definição do gênero do jogo possibilita que seja definido como será distribuído o conteúdo. Como exemplo, um RPG tende a possuir uma narrativa mais complexa, diferente de um Puzzle, que está mais voltado para a jogabilidade e raciocínio.
Adaptação	<i>P10. Como será feito o processo de adaptação?</i>	O processo de adaptação de conteúdo costuma ser realizado de forma manual, com a aplicação de um questionário antes da interação, que leva o jogador a diferentes caminhos conforme as respostas, ou automático, que permite que as ações do jogador sejam capturadas e tratadas, levando a um consenso lógico e pré-programado de adaptação de conteúdo.

Fonte: (O AUTOR).

5. Experiência do Jogador

O processo de experiência do jogador traz os conceitos que abrangem a forma de como as informações são enviadas a ele, verificando técnicas para trazer emoções positivas ao jogador enquanto interage com o conteúdo abordado pelo jogo, e influenciar a motivação. Este processo está ligado aos padrões de “tratar contexto”, “salvar perfil” e “apresentar conteúdo adaptado”.

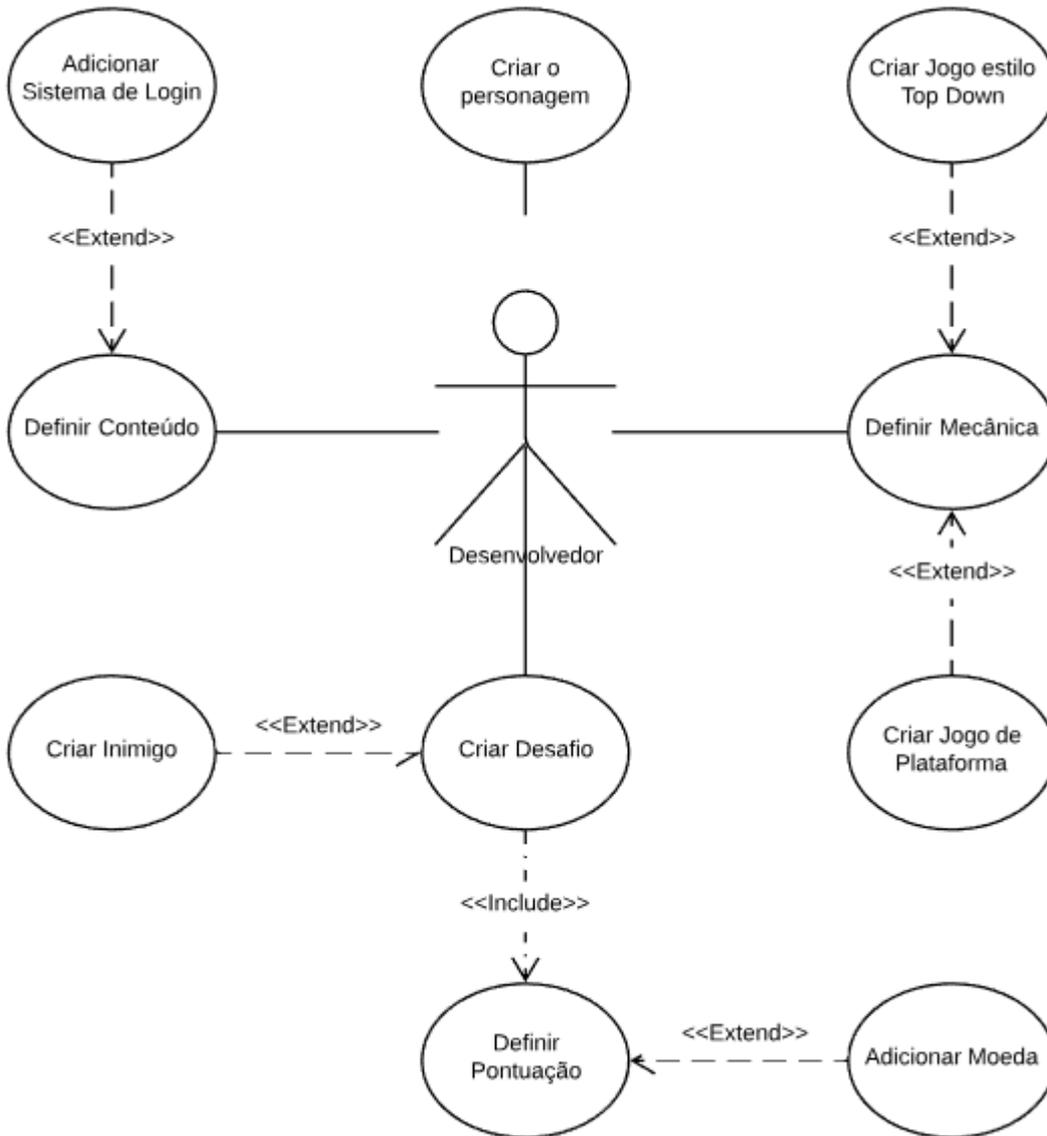
Quadro 5.1 Processo Experiência do Jogador

Subprocesso	Pergunta	Síntese
Feedback	<i>P11. Como o usuário recebe o retorno das suas atividades?</i>	O acompanhamento das atividades do jogador é importante quando é preciso de uma medição quanto a um aspecto específico dele, como a quantidade de quilômetros registrados pelo GPS, ou um movimento gestual realizado corretamente. O retorno pode ajudar o jogador a manter-se jogando para cumprir o objetivo e também a entender o que fez de errado e pode melhorar.
	<i>P12. O jogo possui pontuação?</i>	Um sistema de pontuação pode representar ao jogador o acompanhamento das atividades realizadas. Pode ser feito em tempo real, somando a pontuação a cada atividade realizada, como responder uma questão ou cumprir parcialmente um objetivo estipulado, ou também pode ser dada com o término do jogo, como resultado final.
Motivação	<i>P13. O jogo possui recompensas?</i>	O sistema de recompensas pode trazer ao jogador a motivação esperada em um jogo sério. O objetivo de alcançar determinada recompensa, como itens e moedas do jogo, pode estimular o jogador a realizar uma tarefa que pode não ser considerada tão prazerosa,

		como uma corrida ou o aprendizado sobre algum conteúdo.
	<i>P14. Como é o aspecto competitivo do jogo?</i>	A competição pode estimular ou desmotivar um jogador conforme o estilo dele e do jogo. Jogos multiplayer costumam explorar este aspecto, no entanto, no caso de jogos sérios, o conteúdo não pode ser esquecido em virtude da jogabilidade.

Fonte: (O AUTOR).

APÊNDICE B - CASO DE USO DO FJSU



APÊNDICE C - FORMULÁRIO DA AVALIAÇÃO CAIXA BRANCA

Objetivas

1. Quanto eu achei fácil de utilizar o framework?
2. Quão útil eu achei o framework?
3. Acho que posso utilizar o framework para desenvolver um jogo ubíquo mais rapidamente?
4. Precisei aprender muitas coisas novas para utilizar o framework?
5. O quanto você acha que precisa de auxílio para utilizar o framework?
6. O quanto a documentação do framework auxilia na utilização?
7. Quão limitado você achou o framework?
8. Me sinto seguro para desenvolver um jogo usando o framework?
9. Acho que posso adquirir mais habilidades com desenvolvimento de jogos usando o framework?

Dissertativas

1. Que tipo de recurso você acha que o framework poderia ter?
2. Quais pontos positivos e que podem ser melhorados que você percebeu no framework?

APÊNDICE D - FORMULÁRIO DA AVALIAÇÃO CAIXA CINZA

Utilidade

1. Consigo desenvolver um jogo usando o FJSU.
2. Consigo criar diferentes jogos usando o FJSU.
3. O software possui as ferramentas necessárias para criar um jogo.
4. Encontrei muitos erros durante o desenvolvimento.
5. A documentação do FJSU me ajudou no desenvolvimento.
6. Acho que o FJSU me ajuda a aprender sobre desenvolvimento de jogos.

Facilidade

7. Acho que o FJSU facilita o desenvolvimento de jogos.
8. Precisei estudar muito para utilizar o FJSU.
9. Precisei consultar a documentação muitas vezes.
10. As vídeo aulas me ajudaram a aprender.
11. Consigo encontrar facilmente as ferramentas que preciso.
12. Recomendo o FJSU para iniciantes no desenvolvimento de jogos.

Intenção

13. Gostei de usar o FJSU para desenvolver um jogo.
14. Me sinto motivado para desenvolver um jogo usando o FJSU.
15. O tempo passou rápido enquanto usava o FJSU, me mantive concentrado(a).

APÊNDICE E - ROTEIRO DE AVALIAÇÃO PARA QUEM UTILIZOU O *UNITY* PURO

1 - Criar personagem

Vídeo:

<https://www.youtube.com/watch?v=2pGD25hxs0A>

Documentação:

Sprites: <https://docs.unity3d.com/Manual/Sprites.html>

Rigid Body 2D: <https://docs.unity3d.com/Manual/class-Rigidbody2D.html>

Collider 2D: <https://docs.unity3d.com/Manual/Collider2D.html>

Movimento:

<https://www.fabricadejogos.net/posts/tutorial-jogo-de-plataforma-no-unity-5-parte-3-movimentando-o-personagem/>

Pulo:

<https://www.fabricadejogos.net/posts/tutorial-jogo-de-plataforma-no-unity-5-parte-6-pulo/>

1.1 Camera Seguir Jogador

Vídeo:

https://www.youtube.com/watch?v=9VYrs1_GoBU

Documento:

<https://www.fabricadejogos.net/posts/tutorial-de-um-jogo-de-plataforma-unity-parte-4/>

2 - Criar Inimigo

Vídeo:

<https://www.youtube.com/watch?v=D96XY5u-0fo>

Matar inimigo no pulo:

<https://www.youtube.com/watch?v=sdGeGQPPW7E>

Morte do jogador:

<https://forum.unity.com/threads/how-to-kill-a-player-in-a-2d-game.411044/>

3 - Moedas e pontuação

Vídeo:

https://www.youtube.com/watch?v=300_t88GnY8

Documentação:

<https://unity3d.com/pt/learn/tutorials/projects/space-shooter-tutorial/counting-points-and->

[displaying-score](#)

3.1 Checkpoint

Video:

<https://www.youtube.com/watch?v=JR7UaE-lsrE>

ou

<https://www.youtube.com/watch?v=WUx3svZ3WB4>

Documento:

<https://answers.unity.com/questions/1004843/2d-c-player-dying-at-respawn-at-checkpoint.html>

3.2 Login

Video:

https://www.youtube.com/watch?v=vFs0_skd0E4

ou

https://www.youtube.com/watch?v=h9Fv9b39_tw