

figurethplop figureFigure

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO RIO GRANDE DO SUL
CÂMPUS BENTO GONÇALVES-RS

Jean Carlo Machado

Construção de um jogo, ubíquo com HTML5

Bento Gonçalves, Julho 2013

Jean Carlo Machado

Construção de um jogo, ubíquo com HTML5

Trabalho de conclusão de curso

Projeto de Pesquisa apresentado junto
ao Curso de Tecnologia em Análise e
Desenvolvimento de Sistemas do Ins-
tituto Federal de Educação, Ciência e
Tecnologia do Rio Grande do Sul, como
requisito parcial ao desenvolvimento do
Trabalho de Conclusão de Curso.

Orientador: Dr. Rogério Tessari

Bento Gonçalves, Julho 2013

Este trabalho trata-se de como criar um jogo utilizando das tecnologias emergentes, HTML 5, Javascript e CSS o qual seja transparente à plataforma de utilização, mesmo utilizando recursos específicos de hardware ou software, Fêz-se um levantamento das tecnologias existentes foi proposta uma metodologia para a criação de um jogo de ação que funcione em todas as plataformas populares, tendo em foco principal a utilização dos recursos oferecidos pelas plataformas *mobile* emergentes.

Palavras chave: multiplataforma, sistema operacional mobile, smartphone, compração, jogos

Lista de Figuras

1	Soluções implementadas pelo projeto Apache Cordova.	14
2	Ciclo de vida do software no OpenUP.	18

Lista de Tabelas

1	Cronograma do projeto.	19
---	--------------------------------	----

Conteúdo

1	CONTEXTUALIZAÇÃO	4
2	PROBLEMA	9
3	OBJETIVOS	10
4	JUSTIFICATIVA	11
5	REVISÃO BIBLIOGRÁFICA	12
5.1	HTML5	12
5.2	Paradigmas de páginas para dispositivos móveis	12
5.3	Frameworks polyfill	13
5.4	Bibliotecas canvas	13
5.5	Processos de software	15
6	METODOLOGIA	17
6.1	Plataforma de desenvolvimento	17
6.2	Processo de desenvolvimento	17
6.3	Trabalhos similares	18
7	CRONOGRAMA	19
8	REFERNCIAS	20

1 CONTEXTUALIZAÇÃO

Os seres humanos sempre demonstraram interesse por jogos, há referências arqueológicas que indicam a prática de jogos há mais de 5000 anos. Estes são usados para o entretenimento, todavia, outros fatores colaterais, em grande maioria benéficos são observáveis na prática de jogar, dentre estes constam: melhorias na capacidade lógica, motora, bem como na resolução de problemas de caráter genérico.

Os jogos na computação surgiram em meados dos anos 60, estes eram extremamente simples e totalmente dependentes da plataforma onde eram desenvolvidos, sem mencionar o fato de serem protótipos de laboratório os quais nunca saíram destes locais.

Com a criação da Atari em 1972, os jogos se popularizaram criando um mercado competitivo onde grandes corporações internacionais disputavam entre si para criar as melhores plataformas (consoles) e jogos. Essas corporações dentre as maiores ressaltam-se: Nintendo, Sony e Microsoft; devido ao alto teor de competitividade, desenvolveram novas tecnologias, aumentando a interatividade e imersão do usuário.

Grandes exemplos destas tecnologias são: a criação de renderizadores de elementos de 3 dimensões (3D), bibliotecas de física newtoniana, grandes produções musicais de tema para jogos, bem o acréscimo no nível de detalhamento gráfico devido às GPU's *Graphics Processing Unit's* (ou unidades de processamento gráfico) mais poderosas utilizadas.

Porém juntamente com a evolução das tecnologias aumentou também a dificuldade de se construir jogos de caráter comercial. Quando à priori era necessário uma equipe de 5 a 6 pessoas durante uma média de 6 meses para construir um jogo diga-se popular, hoje faz-se necessário período de 3 a 4 anos com equipes de 500 a 1000 pessoas, como é o caso do GTA 4 que levou 4 anos para ser desenvolvido com uma equipe de aproximadamente 1000 pessoas.

À partir dos anos 80, com a popularização da plataforma PC, decorrente do barateamento do custo de fabricação de circuitos integrados e da criação de grades softwares como sistemas operacionais e aplicativos office, os jogos de console perderam parcela de mercado, todavia não deixaram de existir. Este declínio de mercado ocorreu porque com um PC de aproximadamente 200U\$ obtinha-se uma plataforma que além de jogar possibilitava a realização de inúmeras outras tarefas como estudar e trabalhar, possibilitando assim, um grande benefício em relação aos 150U\$ em média gastos com consoles que serviam exclusivamente para jogos.

Recentemente, devido barateamento e miniaturização dos componentes, tendência já prevista por Moore em 1965, bem como a massificação da internet, surge um novo mercado, o de aplicativos mobile. Torna-se assim possível construir celulares, palmtops e tablets com a arquitetura muito semelhante às dos PC's, inclusive rodando os mesmos sistemas operacionais, que quando não idênticos sofrem poucas customizações, o que permite migrar se não todos, a grande maioria dos aplicativos desenvolvidos para os computadores de mesa para os dispositivos mobile, agregando além de seus benefícios primevos, os trunfos da mobilidade e onipresença que este novo paradigma permite.

Contudo, com a grande popularização deste tipo de arquitetura, cada vez mais novos sistemas operacionais para smartphones e aparelhos semelhantes fazem-se disponíveis, cada qual com particularidades tais como linguagens suportadas, recursos mínimos e aparelhos compatíveis, são exemplos de sistemas operacionais mobile: Android, FirefoxOS, Ubuntu, Windows Phone e o ainda não lançado Taizen. Sob esta perspectiva os desenvolvedores sentem cada vez mais o desafio que é construir aplicativos com abrangência suficiente de plataforma, de modo a alcançar todos ou a grande maioria de seus usuários.

Para tratar deste problema algumas soluções já foram propostas dentre as quais cito:

- utilização de linguagens de programação rodando máquinas virtuais (teoricamente independente de plataformas) como o Java;
- utilização da linguagem amplamente utilizada como padrão para o desenvolvimento como no caso o HTML em especial sua nova especificação o HTML5;

Sobre a utilização de linguagens interpretadas ou parcialmente compiladas rodando em máquinas virtuais, nota-se problemas, pois estas tecnologias nem sempre estão disponíveis em todas as plataformas e os aplicativos são submetidos à temas e outros detalhes de arquitetura específicos de plataforma para plataforma.

A alternativa da utilização da linguagem HTML5, é bem mais promissora, pois em sua nova especificação o HTML5 não trata mais somente de hipertextos trafegando pela rede, serve também para desenhar *on the fly* (ou dinamicamente), permite também rodar em modo offline, implementa semântica e acessibilidade, entre outros recursos, na criação de aplicativos Web de estilo Desktop.

Assim como as maiores tecnologias concorrentes o Adobe Flash e o Microsoft Silverlight o HTML5 também não está totalmente difundido nas plataformas mobile, todavia suas perspectivas são interessantes, estima-se que para o ano de 2016 85% dos navegadores estarão utilizando esta tecnologia. HTML5 não é somente um objetivo de mercado, outra pesquisa aponta que 2/3 dos desenvolvedores de software manifestam interesse em criar aplicações com a tecnologia, o que aponta a linguagem como uma ferramenta bastante promissora nos anos vindouros. Contudo, o HTML trata apenas da estruturação e disposição do conteúdo, para a criação de aplicativos completos é necessária a utilização de scripts os quais implementam a lógica da funcionalidade e de alguma tecnologia para a customização da aparência do aplicativo. As soluções mais populares para estas demandas são o Javascript e o CSS (*Cascading Style Sheets*), respectivamente, que serão tratados mais adiante.

Os jogos, não são a exceção em representatividade nas plataformas mobile, e devido à sua intrínseca complexidade de arquitetura tendem a ser um desafio ainda maior no quesito portabilidade. Este problema se agrava quando os recursos de hardware ou software específicos de plataforma, como o acelerômetro, câmera, GPS, lista de contatos, etc, cuja implementação de soluções de software é extremamente dependente de arquitetura forem utilizados na construção dos jogos.

Quando tratando destas especificidades de plataforma, muitas vezes não é possível utilizar somente o HTML5, pois sua especificação ainda não está completamente fechada outrossim muitos browsers ainda não implementam estas especificações, demanda-se então a utilização de algum *polyfill* (ou *polyfilter*), que nada mais é que uma API [Application Programming Interface (ou Interface de Programação de Aplicativos)] para a utilização de recursos que normalmente não estão disponíveis para ferramentas Web.

Sob este prisma o trabalho a ser desenvolvido propõe a construção de um jogo, desenvolvido em HTML5, Javascript, CSS3 e alguma ferramenta polyfil (caso haja necessidade), que possibilite a criação bem como a utilização linear em todas as plataformas populares, fazendo uso do acelerômetro como recurso de hardware específico de plataforma.

A criação do jogo não é o foco principal do trabalho, todavia afim de ilustrar os conceitos abordados, propõe-se um jogo de plataforma 2D, no estilo de Mario, onde o jogador terá de passar por uma série de desafios para conseguir sair de cada fase. Sendo composto na sua totalidade de 3 fases com dificuldade incremental.

Caberá ao jogador, conduzir seu personagem através do cenário, de modo que o mesmo sobreviva aos desafios propostos, caso o personagem venha a morrer o usuário deverá recomeçar o jogo. O acelerômetro será utilizado como acessório

para alterar os eixos do jogo, sendo caso o jogador vire o dispositivo, o jogo deve "virar" também, derrubando os personagens e alterando cenário.

Cabe ressaltar que a funcionalidade relativa ao acelerômetro serve para elucidar um componente específico de plataforma o qual deverá ser suportado.

2 PROBLEMA

A dificuldade de criar jogos, fornecendo uma experiência de alta similaridade de utilização por arquitetura, que suporte as plataformas de maior popularidade, utilizando um único set de tecnologias, fazendo uso de recursos específicos de plataforma.

3 OBJETIVOS

Objetivo primário

Em suma este trabalho tem por objetivo principal a criação de um jogo, de plataforma, de presença ubíqua, ou seja, que rode em todas plataformas mobile de maior popularidade, vulgo Android, IOS, Windows 8 e FirefoxOS, suportando também as funcionalidades independentes de hardware para os sistemas operacionais de computadores de mesa diga-se, Linux, Windows e Mac; e que forneça uma experiência de usuário o mais linear possível. O foco do trabalho reside na prova da factibilidade do conceito e não na construção do jogo em si.

Objetivos secundários

Dentre os objetivos de caráter secundários constam:

- Desenvolver o aplicativo utilizando somente de tecnologias open source;
- Deseja-se também identificar os pontos relevantes (fraquezas e acertos) que a implementação atual do HTML5, nos diferentes tipos de plataforma, dessa forma sendo possível obter uma visão panorâmica de o quanto falta para o HTML5 estar plenamente estabelecido;
- Objetiva-se também demonstrar os conhecimentos adquiridos no curso de Análise e Desenvolvimento de Sistemas, abordando às áreas de programação, desenvolvimento de sistemas, interação humano computador, engenharia de software entre outros;

4 JUSTIFICATIVA

O trabalho se justifica, por fornecer um material de *proof-of-concept* (ou prova de conceito) sobre a factibilidade da construção de jogos de experiência linear em múltiplas plataformas utilizando as tecnologias emergentes, que devido a isso estão pouco difundidas através de trabalhos acadêmicos.

Servirá também para demonstrar o estado da arte do HTML5 e os contornos necessários, para se desenvolver um jogo completo, multiplataforma fazendo-lhe uso.

Considerando à popularidade destas tecnologias e suas perspectivas o trabalho também poderá auxiliar desenvolvedores, a selecionar um conjunto de tecnologias os quais poderão trabalhar com maior facilidade em meio ao ecossistema de soluções existentes e não cair nos erros que este tipo de tecnologia pioneira ainda pode pregar.

Vários trabalhos tratam da utilização de HTML5 na criação de jogos, todavia, este trabalho se diferencia pela abordagem da criação de jogos, na perspectiva das diferenças de cada plataforma e na procura de uma solução que permita tanto o desenvolvimento quanto uma utilização horizontalizada.

5 REVISÃO BIBLIOGRÁFICA

5.1 HTML5

O padrão HTTP é conhecido por ser o principal fomentador da WEB e a especificação de texto deste padrão é o conhecido HTML, e sua concepção inicial, Tim Berners-Lee acreditava que seria possível interligar hipertextos em computadores diferentes com uso de links globais também chamados de hiperlinks (SILVA 2011), todavia a iteratividade necessária para a construção de jogos animados em HTML é algo recente, anteriormente só se obtinha com a utilização de ferramentas de terceiros como o Adobe Flash. Esta iteratividade é obtida através da utilização do recurso canvas, que é a tag HTML que permite-se "desenhar" dentro da página. Caso o navegador não suporte a tag nada será mostrado, caso contrário a tag pode ser manipulada por Javascript. Atualmente o canvas suporta somente o desenvolvimento 2D, sua implementação 3D está em desenvolvimento o chama-se WebGL. Por consequência do ainda baixo nível de especificação do WebGL, não optamos por o desenvolvimento de um aplicativo 3D.

5.2 Paradigmas de páginas para dispositivos móveis

Atualmente existe duas formas de desenvolver aplicativos mobile com HTML5, são elas:

- Páginas desenvolvidas na maneira convencional, com otimizações para esta categoria de dispositivos, como poucas ou nenhuma imagem, com resolução ajustável, etc.
- Páginas com aparência nativa: São aquelas páginas que são construídas aparentando serem interfaces gráficas desktop da própria plataforma móvel. Ou seja, as páginas aparentam ser aplicações nativas, utilizando todos os

componentes específicos e a forma de usabilidade totalmente voltada para os monitores otimizados (FRANZINI, 2013).

Páginas ando serem interfaces gráficas desktop

5.3 Frameworks polyfill

Como dito anteriormente, o HTML5 por não ser um padrão completamente especificado, deixa lacunas de suporte em plataformas, para recursos específicos tanto de hardware quanto de software, daí surge a necessidade dos polyfills para completar estas camadas.

Várias soluções estão

5.4 Bibliotecas canvas

Um fator complicante é que a utilização de canvas puro para a criação de jogos é um tanto complexa, e demanda muito tempo devido ao grande número variáveis que devem ser contabilizadas, nestes casos complexos é recomendável a utilização de bibliotecas que implementem recursos adicionais, àqueles fornecidos pela especificação como um motor de física como detecção de colisões e encontro de forças, etc.

São diversas as soluções que propõe soluções para este tipo de problema, dentre elas cito:

- o ambiente *HTML5 Development Environment* (ou ambiente de desenvolvimento HTML5) da Intel, que fornece uma solução completa para o desenvolvimento em plataforma cruzada, com serviços de empacotamento, aplicativos para a criação e testes de aplicativos com montagem de interfaces drag and drop (Intel XDK), baseados na nuvem, e bibliotecas para a construção de jogos utilizando aceleração de hardware, o que garante até

Feature	iPhone /iPhone 3G	iPhone 3GS and newer	 Android 1.0 - 4.2	Windows Phone	 BlackBerry BlackBerry 10 and PlayBook OS	 BlackBerry 4.6-4.7	 BlackBerry 5.x-6.0+	 bada Bada	Symbian	webOS	 TIZEN Tizen
Accelerometer	Yes	Yes	Yes	Yes	Yes	N/A	Yes	Yes	Yes	Yes	Yes
Camera	Yes	Yes	Yes	Yes	Yes	N/A	Yes	Yes	Yes	Yes	Yes
Compass	N/A	Yes	Yes	Yes	Yes	N/A	N/A	Yes	N/A	Yes	Yes
Contacts	Yes	Yes	Yes	Yes	Yes	N/A	Yes	Yes	Yes	N/A	Yes
File	Yes	Yes	Yes	Yes	Yes	N/A	Yes	N/A	N/A	N/A	Yes
Geolocation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Media	Yes	Yes	Yes	Yes	Yes	N/A	N/A	N/A	N/A	N/A	Yes
Network	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Notification (alert)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Notification (sound)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Notification (vibration)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Storage	Yes	Yes	Yes	Yes	Yes	N/A	Yes	N/A	Yes	Yes	Yes

Figura 1: Soluções implementadas pelo projeto Apache Cordova.

duas vezes mais performance que aplicativos mobile, baseados em Web tradicionais. Estas soluções da Intel são free, open source e funcionam através de um plugin para o Google Chrome, ou seja, o desenvolvimento também é multi-plataforma. Perante as características acima descritas esta tecnologia é a mais promissoras dentre as pesquisadas.

- o GWT do Google que é um framework para o desenvolvimento de HTML5 em Java, todavia, esta solução não é adequada pois foge do escopo da linguagem HTML5 que é o que deseja-se demonstrar aqui.

Quando especificidades de plataforma são encontradas, como detalhes de implementação de API's de hardware ou software, vê-se necessário a utilização de alguma ferramenta que abstraia estes detalhes em uma implementação genérica, dentre as bibliotecas para isto existe o apache cordova, também conhecido como PhoneGap. É uma biblioteca polyfill de licença livre que implementa recursos comuns na grande maioria das plataformas mobile, facilitando assim a construção de sistemas que utilizem de recursos que ainda não bem definidos pelo HTML5.

Javascript O Javascript é uma linguagem de script, criada pela Netscape, tendo sua primeira aparição em 1995, e atualmente uma das linguagens de

propósito genérico mais populares do mundo, interpretada em praticamente todos os navegadores atuais, este há de ser utilizado para a criação da lógica iterativa do jogo.

CSS Já o CSS, principalmente em sua versão 3 permite a estilização de páginas de marcação como o HTML e XML em todos os aspectos de desing e layout, seu principal benefício é permitir a separação entre o formato e o conteúdo do documento. Este recurso há de ser utilizado para implementar a estilização do jogo.

As tecnologias acima mencionadas (HTML5, Javascript e CSS) são especificadas por uma organização a W3C *World Wide Web Consortium*, a qual consiste em um consórcio internacional de 400 membros dos mais diversos tipos de entidades, que tem por objetivo estabelecer os padrões de desenvolvimento e tecnologias utilizadas para a construção de conteúdo para a Web.

Um ponto bastante mencionado na bibliografia é que para a criação de jogos fluentes usualmente recomenda-se a utilização da técnica de *double buffering*, segundo (SILVA, 2010) ela consiste em criar um canvas secundário, desenhar nele e depois copiar as variáveis para o canvas principal. Evitando assim que o jogador perceba a tela sendo desenhada em cada frame.

5.5 Processos de software

Como o jogo é um software complexo demanda-se a utilização de metodologias de engenharia de software, dentre os processos de software mais conhecidos, academicamente destacamos:

- OpenUP: este é bem detalhado e de característica iterativa e incremental. Gerando assim, um levantamento mais apurado dos riscos, requisitos e outros detalhes do sistema e a criação incremental do sistema, com requisitos maleáveis.

- Cascata: processo antigo, caracteriza-se por ser pouco maleável aos requisitos mapeados posteriormente ao processo de análise;
- Processo ágil - SCRUM: sua utilização é flexível e sendo um método àgil especifica pouca documentação, ou como dizem, somente a documentação necessária, este processo é bem conhecido e aceito na comunidade de desenvolvimento de software.

6 METODOLOGIA

Primeiramente há de ser criado uma versão demo utilizando de imagens e audio de características livre na internet, após o demo criado pretende-se então criar os modelos reais, bem como selecionar os as faixas musicais finais.

6.1 Plataforma de desenvolvimento

Como ambiente de desenvolvimento escolhido utilizar-se-á o XDK da intel, que é um, este ambiente proporciona o desenvolvimento na nuvem e internamente contém um compilador para cada plataforma necessária o que facilita na geração e distribuição de binários. O SDK é atraente também por conter otimizações de software para aumentar o desempenho em jogos. Como dito anteriormente o SDK é um plugin do Google Chrome, então com a liberdade de plataforma, foi escolhido o sistema operacional Linux para realizar o desenvolvimento do software.

6.2 Processo de desenvolvimento

Como processo de desenvolvimento foi escolhido o OpenUp, justamente por este ser incremental e possuir uma alta gama de artefatos, características aprováveis para o desenvolvimento de algo complexo como um jogo.

O Open Up define 4 estágios para o desenvolvimento de software, são elas: **concepção, elaboração, construção e transição**, abaixo estas estão destritas de acordo com o escopo do projeto.

- Concepção: Nesta fase será feita uma análise nos riscos e requisitos do trabalho proposto bem como dar-se-á início ao GDD;
- Elaboração: Esta fase detalhará o funcionamento arquitetural do aplicativo, e de ser concluída a construção do GDD.

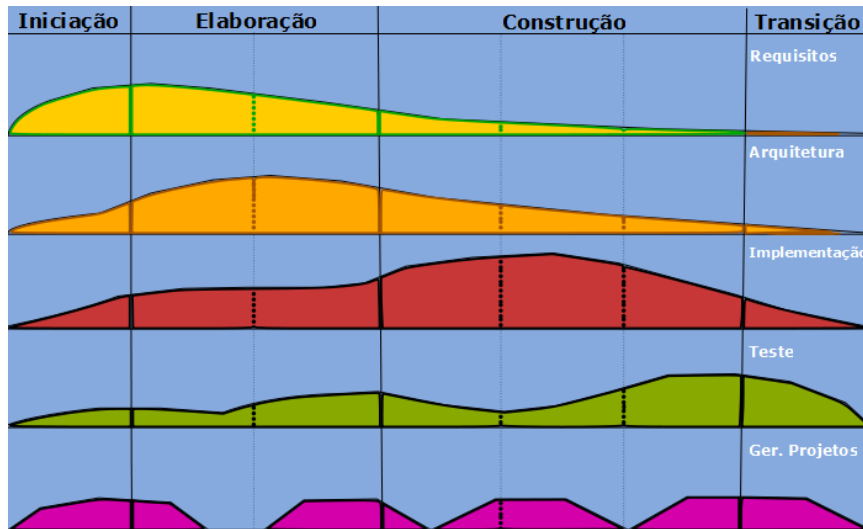


Figura 2: Ciclo de vida do software no OpenUP.

- Construção: É neste estágio que a criação do aplicativo acontecerá, em um primeiro momento da forma de um demo, e posteriormente como uma aplicação completa.
- Transição: Baterias de testes e a documentação dos resultados será desenvolvido neste estágio.

6.3 Trabalhos similares

(SILVA,2010), demonstra a utilização de HTML5 para a criação de jogos simples, todavia seu trabalho não se foca nas diferenças entre uma plataforma e outra, simplesmente comentando o fato, meu trabalho além de demonstrar as discrepâncias, há de propor meios para mitigá-las.

Identificador	Tarefa	Duração	Início	Término
1	Concepção	5 dias	1 agosto	7 agosto
2	Elaboração	15 dias	8 agosto	29 agosto
3	Construção	15 dias	30 agosto	19 setembro
4	Construção	10 dias	31 agosto	3 outubro
	Total	45 dias	1 agosto	3 outubro

Tabela 1: Cronograma do projeto.

7 CRONOGRAMA

O cronograma foi especificado de acordo com o detalhado na metodologia, suas datas estão especificadas de acordo com dias úteis.

8 REFERNCIAS

CUNHA, Cláudio Rodrigues da; SILVA, Júlia Marques Carvalho da; BERTCH, Magda. Proposta de um modelo de atributos para o aprimoramento da comunicação afetiva para professores que atuam na educação à distância. In: Simpósio Brasileiro de Informática na Educação, 2008, Fortaleza, Ceará. Anais... Fortaleza: SBIE, 2008.

CUNHA, Cláudio Rodrigues da. A Gestão de Psicologias a Distância. In: Gestão do Conhecimento: gestão de pessoas, administração pública e educação, 2007, Vol. 1, Rio de Janeiro: Publit Soluções Editoriais.

JAQUES, Patrícia Augustin; VICARI, Rosa Maria. Estado da arte em Ambientes Inteligentes de Aprendizagem que consideram a afetividade do aluno. Informática na educação, UFRGS: Porto Alegre, v. 8, n. 1, p. 15-38, 2005.

LIAO, W.; ZHANG, W.; ZHU, Z.; JI, Q.; GRAY, W. D. Toward a decision theoretic framework for affect recognition and user assistance. In: International Journal of Human-Computer Studies, 64 (9) pp. 847-873. 2006.

LONGHI, Magalí Teresinha; BERTCH, Magda; BEHAR, Patricia Alejandra. Reconhecimento dos estados afetivos de um aluno em Ambientes Virtuais de Aprendizagem. Porto Alegre, 2007. Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2007.

MOODLE. Moodle. Disponível em: <http://www.moodle.org.br/>. Acesso em: abr, 2013. NUNES, Maria Augusta S. N.; BEZERRA, Jonas S.; REINERT, Danilo; MORAES, Daniel; SILVA, Éden P.; PEREIRA, Avner J. S. Computação Afetiva e sua influência na personalização de Ambientes Educacionais: gerando equipes compatíveis para uso em AVAs na EaD. Departamento de Computação, Universidade Federal de Sergipe, Sergipe, 2010.

PICARD, Rosalind. Affective computing. Cambridge: MIT Press, 1997.

SANTO, Edméa Oliveira. Ambientes virtuais de aprendizagem: por autorias livres, plurais e gratuitas. In: Revista FAEBA, v.12, no. 18.2003. Disponível em: <http://www.comunidadesvirtuais.pro.br/hipertexto/home/ava.pdf>. Acesso em: set, 2012.

SILVA, Júlia Marques Carvalho da. Análise da Influência de parâmetros afetivos em um sistema computacional de apoio a aprendizagem de algoritmos. Itajaí, 2005. No f. 140. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação)-Centro de Ciências Tecnológica da Terra e do Mar, Universidade do Vale do Itajaí, Itajaí, 2005.

Simpósio Brasileiro de Informática na Educação, 2011. Aracaju.

referências jean

SILVA, Jucimar Maria Júnior; FIRMINO, Emiliano Carlos M. Desenvolvimento de jogos em HTML5. Coordenação da engenharia da Computação, Univerisdade Federal do Amazonas, Amazonas, 2010.

SILVA, Maurício Samy. HTML5 A LINGUAGEM DE MARCAÇÃO QUE REVOLUCIONOU A WEB. Editora novatec, p. 15, 2011.

FRANZINI, Fernando .Nova tendência de aplicativos móveis web . Disponível em: <http://www.infobase.com.br/nova-tendencia-de-aplicativos-moveis-web/>. Acesso em: jun, 2013.

Acessível em: <http://www.failwars.blog.br/nerd-feelings/incrvel-evoluio-dos-video-games-de-console-de-1967-2012/> Acessível em: <http://pt.wikipedia.org/wiki/Hist>