

XUCRO: UM PROTÓTIPO DE CONSOLE DE VIDEOGAME¹

Gustavo Castro²

José Antônio O. de Figueiredo³

RESUMO

Os jogos sempre estiveram presentes desde os primeiros passos da computação e com o tempo acabaram se disseminando aonde fossem suportados. Atualmente, os sistemas embarcados, devido a sua flexibilidade e facilidade de manipulação, são utilizados nas mais diversas áreas da tecnologia com os mais diferentes objetivos. Neste trabalho, é apresentado o projeto e o desenvolvimento de um protótipo de console de videogame baseado em uma plataforma embarcada, o Arduino. Para o desenvolvimento do projeto, basicamente foi utilizando uma placa Arduino Mega 2560, um módulo de conexão Ethernet e de Armazenamento, um módulo de vídeo no padrão RCA, além de um *joystick*, baseado em dois botões digitais e uma alavanca analógica. Além do console, o projeto se propôs a desenvolver uma plataforma de compartilhamento de jogos, para que os usuários possam compartilhar seus jogos, e códigos-fonte. Ao final deste trabalho obteve-se sucesso em desenvolver um protótipo funcional, capaz de executar diversos jogos ao interagir com uma plataforma de compartilhamento, que funciona como um repositório público jogos.

Palavras-chave: Jogos. Consoles de videogame. Arduino. Sistemas Embarcados.

1 INTRODUÇÃO

Na área de sistemas embarcados, a possibilidade de aplicações destes abrange as mais diversas áreas da tecnologia, devido a sua flexibilidade e facilidade de manipulação. Quando relacionados ao conceito de videogames, é possível aplicá-los em várias funções, como por exemplos, na renderização de imagens, estabelecer conexões entre máquinas através de uma rede e capturar dados de entrada do meio externo através de botões.

Entretanto, trabalhos científicos focando o desenvolvimento de sistemas embarcados, aplicados ao conceito de videogame, não apresentam um grande

¹ Trabalho de Conclusão de Curso (TCC) apresentado ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-rio-grandense, Câmpus Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet, na cidade de Passo Fundo, em 2016.

² Graduando em Tecnologia de Sistemas para Internet pelo IFSUL campus Passo Fundo. E-mail: gustavocastro.pf@gmail.com

³ Orientador, professor do IFSUL. E-mail: jose.figueiredo@passofundo.ifsul.edu.br.

destaque, devido ao fato de que, geralmente os sistemas embarcados não apresentam um *hardware* de alta performance, quando associados a jogos eletrônicos. Porém, pode-se ter em vista de que o assunto pode ser relacionado em outras áreas além da de programação e eletrônica, como por exemplo, ser aplicada na área de educação, onde pode ser útil como ferramenta que auxilie no desenvolvimento de alunos.

Dessa forma, este trabalho se propõe a contribuir com o tema, projetando e prototipando um console de videogame, em uma plataforma embarcada, onde seja possível observar todo processo de desenvolvimento, além de desenvolver uma plataforma para o compartilhamento dos jogos para este console.

Entre os trabalhos relacionados, o projeto de graduação de Oliveira e Carvalho, *TuKolk: Um jogo de console utilizando a plataforma Arduino* (2014), pela Universidade Federal do Estado do Rio de Janeiro, destaca-se pelo fato de apresentar o desenvolvimento de um protótipo de console de jogo utilizando o sistema embarcado *Arduino*, uma proposta semelhante ao deste. Enquanto Oliveira e Carvalho (2014) apresentam uma proposta de console com apenas um jogo, nosso trabalho possui o intuito de apresentar uma *dashboard* com uma variedade de jogos, estes disponibilizados através de uma plataforma online.

As demais partes do artigo estão divididas da seguinte forma: a segunda seção apresenta o desenvolvimento do trabalho, realizando uma apresentação ao leitor do tema abordado, ou seja, jogos e suas plataformas, além de abordar a plataforma *Arduino* e suas respectivas características. Na seção que a sucede, é demonstrado a metodologia abordada para o desenvolvimento do trabalho, abordando diagramas para a realização do mesmo, o processo de criação do protótipo e os resultados obtidos. Na quarta seção estão dispostas as considerações finais e a premissas de trabalhos futuros.

2 REFERENCIAL TEÓRICO

Nesta seção são apresentados as tecnologias e materiais utilizados para o desenvolvimento do projeto, além de realizar a conceitualização do tema abordado neste trabalho.

2.1 VIDEOGAMES

Nesta sessão é apresentada uma visão geral sobre videogames, tal como jogos e as plataformas que os comportam, com o intuito de introduzir o leitor ao tema abordado no trabalho. Neste trabalho, a palavra jogo deve ser interpretada como jogo eletrônico, para melhor compreensão.

Oliveira (2010) define jogos como “qualquer atividade interativa, num ambiente computacional (computadores, videogames, portáteis etc) que envolva certas regras, um contexto e que não seja responsável apenas por calcular resultados diretos a partir de uma entrada (como calculadoras, processadores de texto, etc), e sim gerar um ambiente de interação contínua com o usuário, tendo suas ações reações que abrem a possibilidade de outras ações (iguais ou diferentes), até que um fim seja alcançado (o término do programa por alguma regra definida, ou pelo usuário).”

2.1.1 Plataformas

Segundo Rabin (2012), os jogos sempre estiveram presentes em todos os lugares onde pudessem ser suportados, e isso não mudou com o decorrer do tempo, hoje em dia existe uma vasta coleção de plataformas diferentes, cada qual com suas particularidades e recursos extremamente variados.

Computadores Pessoais

Atualmente os computadores pessoais são a plataforma mais utilizada para jogos, e o principal fator disto é o *hardware* do mesmo, que devido sua alta gama de opções, os torna mais atrativos para aqueles usuários que buscam os melhores gráficos, áudio e performance (Rabin, 2010).

Ironicamente, o maior defeito dos computadores pessoais também é seu *hardware*, devido ao fato que nem todo mundo possui o *hardware* mais atual para tirar o maior proveito possível do jogo, e isso acaba refletindo no desenvolvimento do jogo, onde a empresa desenvolvedora tenta atingir o maior número possível usuários, porém sem deixar a desejar na performance e na sofisticação do jogo (Rabin, 2010).

Consoles de Videogames

Rabin (2012) define consoles de videogames como um conjunto fixo de *hardware* com uma única finalidade: jogar jogos. A utilização de um *hardware* fixo é algo que acaba implicando nos dois extremos do desenvolvimento de jogos. Primeiramente, a ideia de utilizar um *hardware* constante na programação, é sinônimo de estabilidade, devido ao fato de que o desenvolvedor não vai ter que se preocupar com uma vasta lista de placas de vídeo incompatíveis ou com a incompatibilidade com dispositivos de entrada diferentes. Porém, um *hardware* restrito, acaba por limitar o desenvolvimento do jogos em diversos aspectos. Portanto, Rabin (2012, p.180) ressalta que “jogos destinados a consoles devem levar em conta as limitações do console específico e lidar com elas desde o início”.

2.2 ARDUINO

Segundo Arduino (2016), o Arduino é definido como uma plataforma de prototipação eletrônica baseada em um conjunto de *hardware* e *software* que possui o intuito de interagir com elementos do meio externo de forma fácil e flexível. O *hardware* é composto basicamente por uma placa com um microcontrolador Atmel AVR, que possui diversas portas para a entrada e saída de dados (I/O). Quanto a parte do *software*, o Arduino atua em conjunto com uma interface de desenvolvimento integrado (IDE), baseada em Processing, que é responsável pela programação do *hardware*.

McRoberts (2011), define o Arduino como um computador embarcado, devido ao fato de conseguir interagir com o ambiente em que ele se encontra. Essa interação pode ser realizada tanto de forma independente, quanto através de um *software* de um computador ou rede, onde o Arduino possa enviar ou recuperar dados, e/ou agir sobre eles.

A plataforma se destaca por ser distribuída sob licença open-source, tanto o *hardware*, quanto o *software*, dessa forma, pode-se ter total acesso sobre os diagramas esquemáticos da placa e também aos códigos fonte da *firmware*, sendo assim, podendo obter, modificar e redistribuir livremente os mesmos.

De acordo com Arduino (2016), a programação do microcontrolador da plataforma é feita através da linguagem Arduino, esta por sua vez tendo como base um framework em C/C++ para programação de microcontroladores, o *Wiring*.

Devido a sua licença open-source, o Arduino possui diversas versões, é até mesmo possível para os desenvolvedores criarem suas versões da placa, que atendam os requisitos de seu projeto. Dentre as versões mais conhecidas do Arduino, destacam-se: Diecimila, Mega, Yun, Nano e Uno.

2.2.1 Arduino Mega

Segundo Arduino (2016), a versão Mega, da família Arduino, pode ser baseada em dois microcontroladores, o Atmega 1280 ou Atmega 2560, este trabalho utilizou a versão que contém o microcontrolador 2560. Possuindo 54 pinos digitais de entrada e saída, 16 pinos analógicos de entrada, um cristal oscilador de 16 Mhz, uma porta de comunicação USB, um conector de alimentação, um conector ICSP e por fim, um botão reset. Na Figura 1 é apresentado um exemplar deste modelo.

Figura 1 - Arduino Mega



Fonte: (ARDUINO, 2016)

Quanto ao microcontrolador ao qual a versão Mega é baseada, se caracteriza pelo fato de possuir 256KB de memória flash, sendo que 8KB utilizado para o *bootloader* do Arduino, 8KB de memória SRAM, 4KB de memória EEPROM e modo de processamento 8-bit.

3 PROJETO

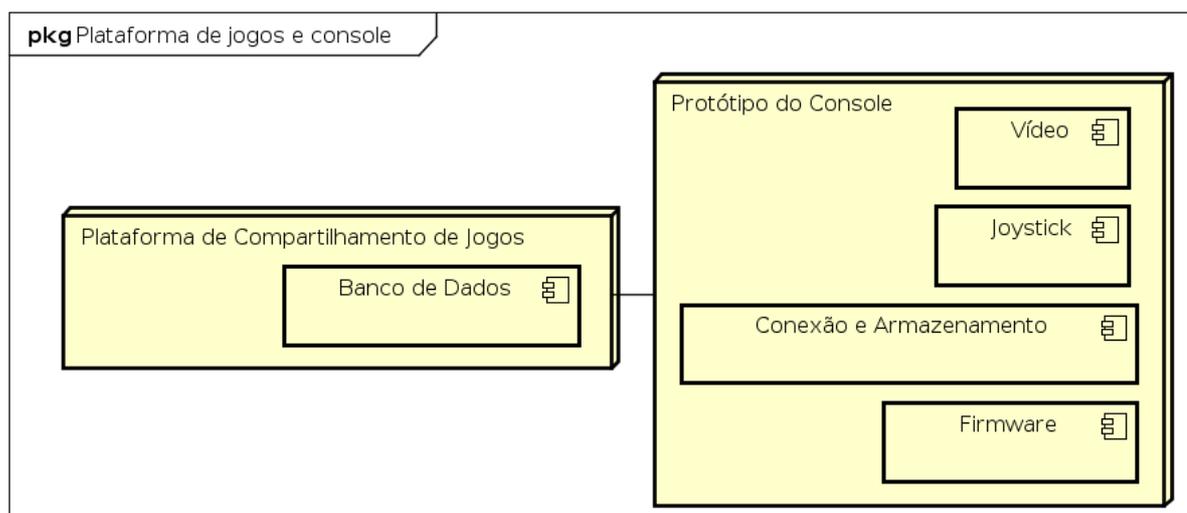
Nesta seção é descrita a metodologia proposta para o desenvolvimento deste projeto, incluindo processo necessário para realizá-lo, apresentando os materiais utilizados, o esquema montagem do *hardware* do console, além de demonstrar a

modelagem utilizada para desenvolver a *firmware* presente no console. O projeto recebeu o nome de Xucro, palavra utilizada em dialetos da região sul do Brasil, que significa algo indomável, livre, dessa forma fazendo alusão a comunidade Open-Source.

3.1 METODOLOGIA

Como descrito anteriormente, o projeto almeja desenvolver o protótipo de um console de videogame integrado a uma plataforma de jogos. Dessa forma o desenvolvimento está em duas partes: o desenvolvimento do console e o desenvolvimento da plataforma onde os jogos serão compartilhados, conforme pode ser visto na Figura 2.

Figura 2 - Plataforma de compartilhamento de jogos e protótipo do console



Fonte: (DO AUTOR, 2016)

3.1.1 Protótipo

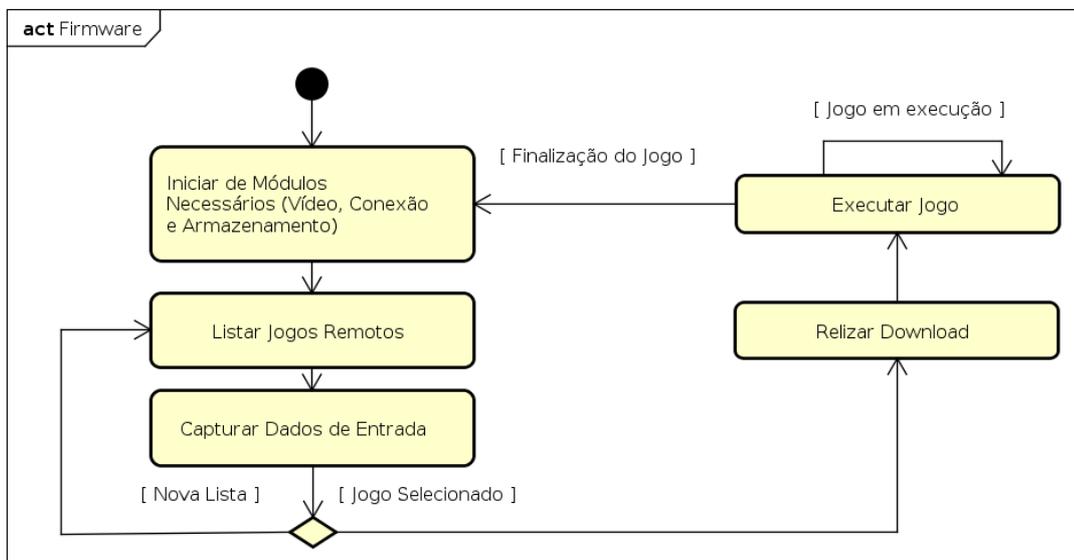
O protótipo do console, este é dividido em duas partes, o desenvolvimento da parte física, ou seja, o *hardware* do videogame, e a parte lógica, na qual se encontra a *firmware*, responsável por controlar o *hardware* e as interações externas.

O desenvolvimento do *hardware* do projeto será realizado com um microcontrolador, conectado a módulos que realizam operações específicas (*shields*). Tais módulos serão responsáveis por realizar tarefas como renderização de vídeo, conexão à rede, armazenamento de dados e captura de dados de entrada. Porém, todos os dados gerados ou consumidos por estes módulos serão tratados pelo microcontrolador.

3.1.2 Firmware

O funcionamento da *firmware* do console em um primeiro momento irá iniciar todos os módulos necessários, em seguida, a mesma realiza uma requisição a plataforma de compartilhamento de jogos, através de uma conexão web, onde é requisitado uma lista de jogos, quando a mesma é recebida, o console aguarda uma entrada de dados. O dado recebido pela entrada consiste na solicitação de um jogo ou de uma nova lista. Caso a solicitação seja de uma nova lista, o console simplesmente faz a requisição de uma nova lista e realiza o processo anterior, caso ao contrario, o console realiza a solicitação do jogo escolhido, e ao receber o arquivo da plataforma, o armazena no dispositivo de armazenamento, e em seguida, executa o jogo, até que ocorra a solicitação de encerramento jogo e o console é reiniciado ou desligado. O processo pode ser observado no diagrama de atividades na Figura 3.

Figura 3 - Funcionamento da firmware do console



Fonte: (DO AUTOR, 2016)

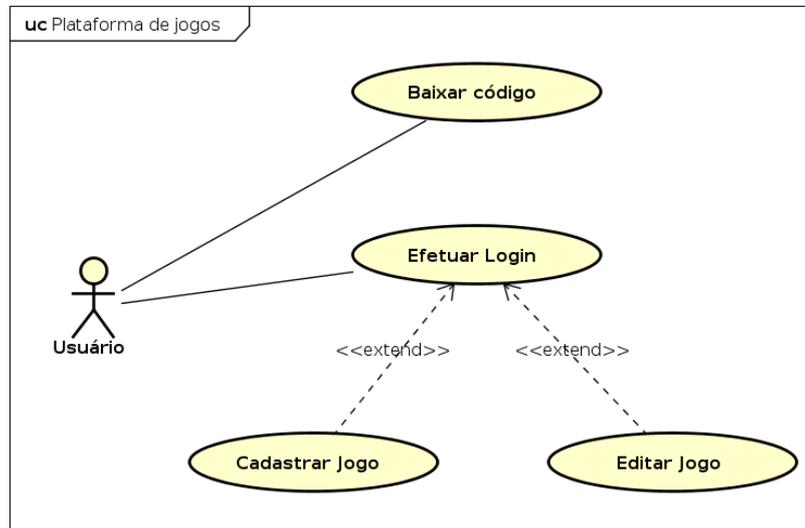
3.1.3 Plataforma de jogos

A segunda parte do projeto, consiste na plataforma de compartilhamento, o qual tem o objetivo de fornecer aos usuários finais uma forma de distribuir os jogos desenvolvidos para o console. O *software* foi desenvolvido na linguagem de programação PHP.

A plataforma opera como uma biblioteca de jogos, onde um usuário tem acesso ao código-fonte e ao código compilado dos jogos. Um usuário cadastrado

pode disponibilizar seus próprios jogos na plataforma, assim como realizar atualizações e edições nos mesmos, conforme demonstra o diagrama de caso de uso da Figura 4.

Figura 4 - Diagrama de caso de uso da plataforma de jogos



Fonte: (DO AUTOR, 2016)

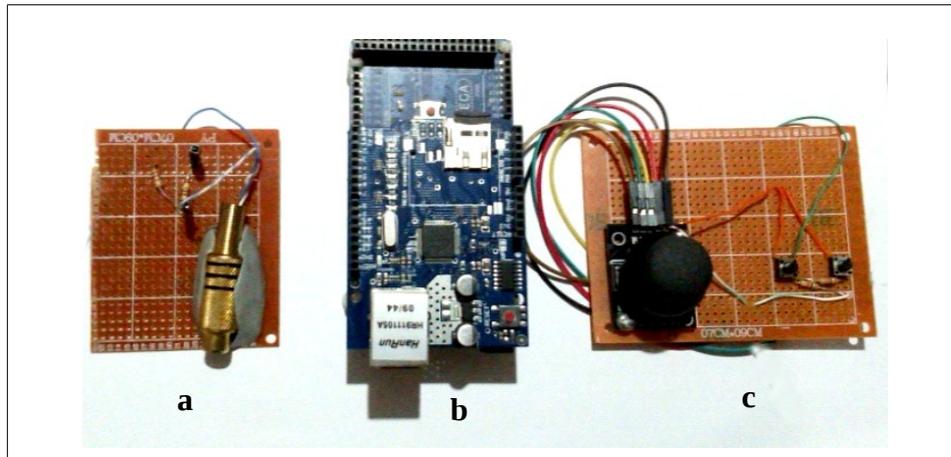
4 RESULTADOS OBTIDOS

Nesta sessão é apresentado o desenvolvimento do projeto e os resultados obtidos após sua finalização.

4.1 PROTÓTIPO DO CONSOLE

Na Figura 5, pode ser observado as partes do protótipo do console de videogame desenvolvido. Para o desenvolvimento do protótipo foi utilizado como base uma placa Arduino Mega 2560 (b), responsável pela gerência dos módulos e pelo processamento dos dados que por eles são utilizados. Para estabelecer conexão com a plataforma de jogos, que encontra-se disposta num servidor web, foi utilizado um módulo de Ethernet e Armazenamento (b). Quanto para a renderização de imagens, foi desenvolvido um módulo capaz de gerar imagens utilizando o padrão RCA (a), para exibir as imagens em uma televisão. Por fim, para realizar a entrada de dados, foi desenvolvido um joystick (c), este com três botões de pressão e uma alavanca analógica de dois eixos.

Figura 5: Protótipo do console de videogame



Fonte: (DO AUTOR, 2016)

O funcionamento do console tem início quando o mesmo é conectado a uma fonte de energia, nesse caso, utilizando uma fonte de 12 Volts de 1 A, bivolt, conectada rede elétrica. A primeira ação a ser realizada pelo sistema é exibir na tela uma imagem de apresentação do sistema, ou seja, o logo do console, conforme pode ser visto na Figura 6. Enquanto a imagem é mostrada, são iniciados os módulos de conexão e armazenamento. Caso ocorra algum erro na configuração de algum dos módulos, o sistema exibe uma mensagem na tela e o sistema encerra suas atividades. Vale ressaltar que processo de renderização de imagens é realizado através da biblioteca TVout⁴, e devido a quantidade limitada de memória SRAM do Arduino, de 8 KB, a resolução fica restrita a 128x96px, e com a ausência de um conversor de RGB para NTSC / PAL, o vídeo gerado é em preto e branco.

Figura 6: Tela de apresentação do sistema.

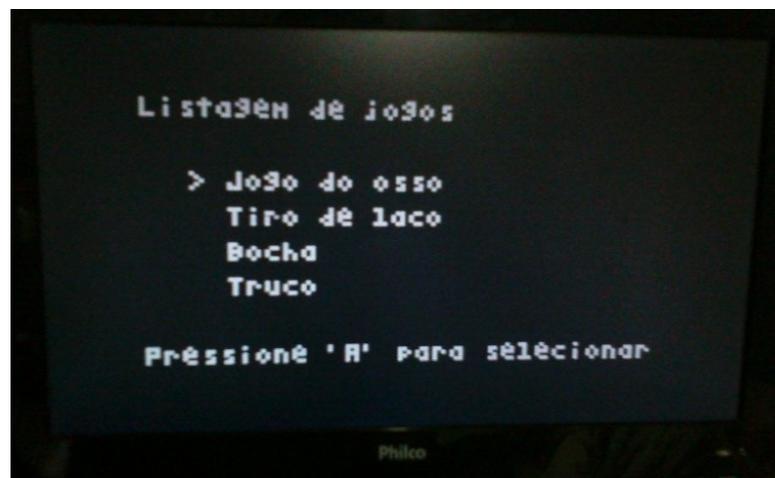


Fonte: (DO AUTOR, 2016)

⁴Tvout: <https://code.google.com/archive/p/arduino-tvout/wikis/FunctionalDescription.wiki>

Após iniciar os módulos necessários, o sistema exibe na tela uma lista de jogos. Nesta tela é possível selecionar um jogo para download, pressionando o botão 'A' do *joystick*, ou requisitar uma nova lista, utilizando a alavanca analógica, movendo-a para a direita para obter uma lista nova, ou para a esquerda para retornar a lista anterior. Nesta fase do trabalho, não foi possível criar um sistema de buscas ou paginação devido a limitação causada pela resolução da tela. Na Figura 7 pode-se observar como são apresentadas as listas, e através dela é possível verificar que não haveria espaço para uma sessão de paginação ou de pesquisa.

Figura 7: Listagem dos jogos



Fonte: (DO AUTOR, 2016)

Ao selecionar um jogo no console, o sistema irá realizar o download deste, localizado na plataforma de jogos. Para fazer o download do arquivo do jogo, o *software* faz uma requisição do arquivo binário do mesmo, e a cada *byte* recebido, o sistema o grava em um novo arquivo, no dispositivo de armazenamento. Este arquivo é armazenado em um cartão SD.

O processo de transferência de dados entre o console e a plataforma de jogos, tanto para realizar o download dos jogos, quanto para listar os jogos, ocorre utilizando o protocolo *HTTP*, onde o console realiza uma requisição ao IP do servidor, na porta 80, passando consigo juntos os parâmetros necessários para realizar tal ação. Caso a ação seja para realizar o download do jogo, o parâmetro passado é o identificador do jogo, que é um número único, caso o contrário, seja para baixar uma lista, ele envia o número da lista que deseja. Na Figura 8, pode-se observar a função responsável por estabelecer conexão entre o console e a plataforma.

Figura 8. Função responsável por estabelecer conexão entre o console e a plataforma

```

/*
 * @Nome: createConnection
 * @Funcao: Buscar elementos atraves da rede
 * @Parametros: resource: url a ser acessada
 *               id: paginacao ou id do item a ser buscado
 */
boolean createConnection(const char* resource, int id){
    if (_CLIENT.connect("10.0.0.101", 80)) {
        Serial.println("connected");
        _CLIENT.print("GET ");
        _CLIENT.print(resource);
        if(id!=NULL)
            _CLIENT.print(id);
        _CLIENT.println(" HTTP/1.1");
        _CLIENT.println("Host: 10.0.0.101");
        _CLIENT.println("Connection: close");
        _CLIENT.println();
        return true;
    }
    else {
        return false;
    }
}

```

Fonte (DO AUTOR, 2016)

Após a realização do download, o sistema inicia o processo de execução do jogo. O responsável pela execução do jogo é o *bootloader avr_boot*⁵. O *avr_boot* é o encarregado de iniciar os arquivos executáveis, ou seja, os jogos e a *dashboard*. Para carregar os executáveis, o *avr_boot*, ao iniciar, verifica a existência do arquivo '*FIRMWARE.BIN*', no dispositivo de armazenamento. Caso o arquivo exista, o *bootloader* irá carregar o executável desejado, caso ao contrário, este ficará aguardando pelo pressionamento do botão *reset* da placa, para novamente tentar carregar o executável.

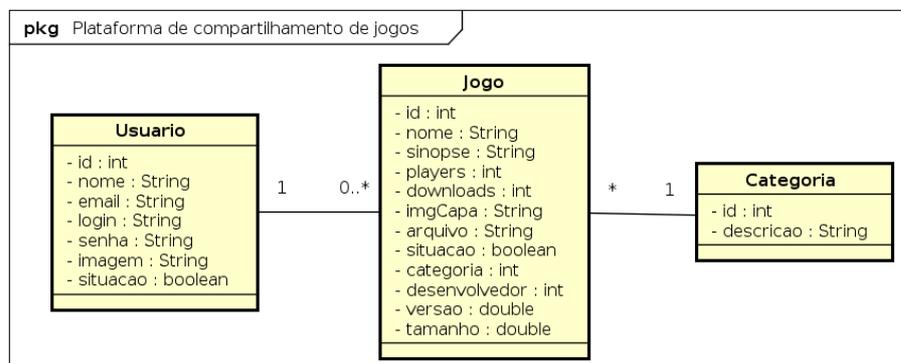
Compreendendo que o *bootloader* necessita de um arquivo chamado '*FIRMWARE.BIN*', a função responsável por executar o jogo após o download, deve renomear o arquivo baixado para que o mesmo possa ser interpretado pelo *avr_boot*, e o arquivo da *dashboard*, que por padrão deve se chamar '*FIRMWARE.BIN*', para que possa iniciar o console, é renomeado para '*DASH.BIN*'. Dessa forma, torna-se essencial uma função no jogo que retorne a *dashboard*, que realize o caminho inverso e renomeando o arquivo de inicialização do console para que possa ser interpretado pelo *avr_boot*.

⁵Avr_boot: https://github.com/zevero/avr_boot

4.2 PLATAFORMA DE COMPARTILHAMENTO DE JOGOS

Para o desenvolvimento da plataforma de jogos, foi utilizada a linguagem de programação para web PHP, além da linguagem de marcação HTML, utilizando o *framework Bootstrap*, para que o sistema possa ser acesso através de qualquer dispositivo. O sistema foi desenvolvido utilizando a estrutura disposta no diagrama de classes na Figura 9. O sistemas conta com três classes: usuário, jogo e categoria.

Figura 9. Diagrama de classes da plataforma de compartilhamento de jogos

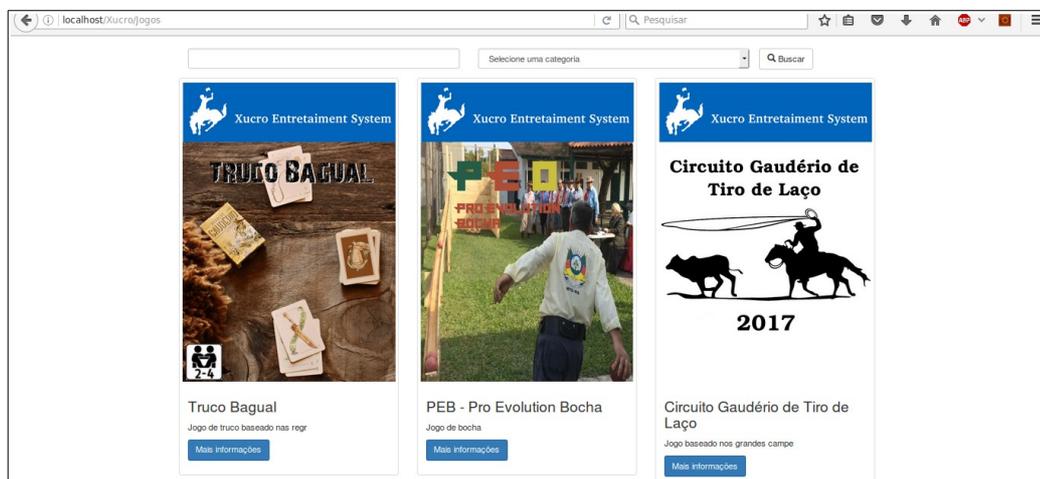


Fonte (DO AUTOR, 2016)

A plataforma conta basicamente com três áreas: Jogos, Cadastro de Usuário e Cadastro de Jogos.

A área de jogos apresenta uma lista de todos os jogos cadastrados pelos usuários que possuem cadastro, onde qualquer usuário tem acesso as informações básicas do jogo e além do código fonte do jogo. A lista de jogos e suas informações básicas de um jogo pode ser observado na Figura 10.

Figura 10: Lista de jogos e informações básicas



Fonte: (DO AUTOR, 2016)

Quanto à área de cadastrado de usuários, no momento conta apenas com um formulário para que possa incluir um usuário na base de dados e disponibilizar a ele a possibilidade de se tornar um contribuidor do plataforma, distribuir seus jogos e arquivos fonte.

A área de cadastro de jogos é baseada em um formulário onde o usuário cadastrado remete informações básicas sobre o jogo, além de o arquivo executável, para que o console possa baixar e executar, e o arquivo fonte, para que outros usuários possam estudar e modificar seu código.

5 DISCUSSÃO DOS RESULTADOS

O principal objetivo do trabalho era projetar e desenvolver um protótipo de um console de videogame que fosse capaz de interagir com uma plataforma de jogos, dessa forma, sendo possível acompanhar todo o processo de desenvolvimento do mesmo. Ao que se propôs o projeto, obteve-se resultados positivos, onde foi possível desenvolver o protótipo, e o mesmo foi capaz de interagir com a plataforma de jogos.

Mesmo a área de videogames não sendo o alvo principal dos sistemas embarcados, este trabalho deixa uma série de legados que podem ser utilizados em outras áreas, como por exemplo o sistema de atualização de *firmware* realizada através do *avr_boot*, essa pode ser uma excelente alternativa para sistemas que exigem que uma grande quantidade de ações sejam realizadas, por exemplo, a obtenção e tratamento de diferentes dados utilizando diferentes sensores ou módulos. Outro ponto a ser ressaltado, é a interação do embarcado com um sistema web, pois com disseminação da Internet das Coisas, a demanda por técnicas que abordem a área de interconexões é crescente. Dessa forma, o trabalho faz sua parte colaborando através da apresentação de um técnica para *download* de arquivos.

Em trabalhos futuros, em um primeiro momento, pretende-se melhorar o *hardware* do console, visando desenvolver um sistema melhor de interação com o usuário, aumentando a resolução das imagens renderizadas e gerando imagens coloridas. Pretende-se também melhorar o *software* do console, desenvolvendo um sistema para armazenar e acessar diversos jogos no dispositivo de armazenamento, ou uma aplicar técnica onde um usuário seja capaz de acessar todos seus jogos

independente do console em que esteja acessando, ou seja, a execução dos jogos de forma remota.

Em um segundo momento, com uma versão mais robusta do console, pretende-se realizar um estudo utilizando o videogame como ferramenta de apoio no ensino nas áreas de eletrônica, algoritmos e desenvolvimento de jogos; pois há a hipótese de que, uma abordagem lúdica na forma de ensino, possa ser capaz de facilitar a aprendizagem. Caso os resultados sejam positivos, cogita-se também a possibilidade de aplicá-lo em outras áreas, como a utilização do mesmo em conjunto à aulas de matemática e física.

ABSTRACT

Games have always been presented from the very first steps of computing, and over time they have spreaded to wherever they are borne. Currently, embedded systems, due to their flexibility and ease of manipulation, are used in the most diverse areas of technology with the most different objectives. In this work, the design and development of a video game console prototype based on an embedded platform, the Arduino, is presented. For the development of the project, basically was using an Arduino Mega 2560 board, an Ethernet and Storage connection module, a video module in the RCA standard, plus a joystick, based on two digital buttons and an analog lever. In addition to the console, the project set out to develop a platform for sharing games, so users can share their games, and source code. At the end of this work we succeeded in developing a functional prototype, capable of running several games while interacting with a sharing platform, which functions as a public repository games.

Keywords: Games. Videogames console. Arduino. Embeeded Systems

REFERÊNCIAS

ARDUINO. **Arduino**. Disponível em: <<http://www.arduino.cc/Portugues/HomePage>>. Acesso em: 27 mar. 2016.

BANZI, Massimo. **Getting Started with Arduino. 2. ed. Sebastopol**: O'reilly, 2011.

EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. **Arduino em Ação**. São Paulo: Novatec, 2013.

MCROBERTS, Michael. **Arduino Básico**. São Paulo: Novatec Editora, 2011.

OLIVEIRA, André Alves; CARVALHO, Fabrício Kolk. **TuKolk**: Um jogo de console utilizando a plataforma Arduino. 2014. 70 f. TCC (Graduação) - Curso de Sistemas de Informação, Universidade Federal do Estado do Rio de Janeiro Centro de

Ciências Exatas e Tecnologia Escola de Informática Aplicada, Rio de Janeiro, 2014. Disponível em: <<http://bsi.uniriotec.br/tcc/201408OliveiraKolk.pdf>>. Acesso em: 29 mar. 2016.

RABIN, Steve. **Introdução ao desenvolvimento de games**: Volume 2. 2. ed. São Paulo: Cengage Learning, 2012. 637 p.