

TECNOLOGIA PARA INTEGRAÇÃO DE GAME ROBÓTICO DE FUTEBOL COM A INTERNET E AS REDES SOCIAIS

Antonio Valerio Netto ¹

RESUMO

O artigo descreve o desenvolvimento de módulos computacionais para o aprimoramento do RoboGol. Trata-se de em um sistema criado para a realização de partidas de futebol de robôs, onde cada robô é controlado individualmente por um humano por meio de joystick. A nova versão do RoboGol busca uma integração com smartphones via bluetooth. E posteriormente, estabelecer uma conexão via Internet com uma plataforma cloud computing para envio de informações referente às partidas de futebol de robô (identificação de cada jogador, número de gols de cada time, tempo decorrente da partida e o número de identificação da mesa). Estas informações são utilizadas para montar um ranking dos melhores jogadores e enviar mensagens para as suas redes sociais (Facebook e Twitter). Estas integrações buscam gerar promover uma comunidade nas redes sociais e gerar um maior engajamento dos usuários, despertando um maior interesse por outras modalidades relacionadas ao futebol de robôs.

Palavras-chave: Robótica móvel, Robogol, Redes sociais, Internet das coisas, Futebol de robôs.

ABSTRACT

The article describes the development of computational modules for the improvement of RoboGol. It is in a system created for the accomplishment of football matches of robots, where each robot is controlled individually by a human by means of joystick. The new version of RoboGol seeks integration with smartphones via bluetooth. And later, establish an Internet connection with a cloud computing platform to send information regarding the robot football matches (identification of each player, number of goals of each team, time of departure and the identification number of the table). This information is used to assemble a ranking of the best players and send messages to your social networks (Facebook and Twitter). These integrations seek to generate a community in social networks and generate a greater engagement of users, arousing a greater interest in other modalities related to robot soccer.

Keywords: Mobile Robotics, Robogol, Social networks, Internet of things, Robot soccer.

1 INTRODUÇÃO

O RoboGol utiliza o conceito de game cibernético. São robôs móveis que até um leigo pode controlar, desenvolvidos para o entretenimento. Consiste em um sistema criado para a realização de partidas de futebol de robôs, onde cada robô é controlado individualmente por um humano. As

¹ Doutor em computação pela USP e bolsista DT CNPq, antonio.valerio@pq.cnpq.br

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

partidas podem ser jogadas por até quatro jogadores (1×1 ou 2×2), os quais comandam os robôs em ações de movimentação para frente e para trás, chute e rotação no próprio eixo, por meio de joysticks com tecnologia wireless. Os robôs são ágeis e robustos, possibilitando movimentos rápidos e respostas precisas aos comandos dos jogadores (Robogol, 2017). Na Figura 1 é exibida uma imagem do RoboGol com informações adicionais. A nova versão é chamada de RoboGol Ultimate. A nova versão do RoboGol busca uma integração com smartphones via bluetooth. E posteriormente, estabelecer uma conexão via Internet com uma plataforma cloud computing para envio de informações referente às partidas de futebol de robô (identificação de cada jogador, número de gols de cada time, tempo decorrente da partida e o número de identificação da mesa). Estas informações são utilizadas para montar um ranking dos melhores jogadores e enviar mensagens para as suas redes sociais.

Figura 1 - Ilustração da mesa do RoboGol com a descrição dos elementos disponíveis.



Fonte: Elaborado pelo autor.

As novas funcionalidades relacionadas ao *hardware* envolvem a criação de um número para identificação da mesa, assim como a aquisição dos dados referentes aos jogadores e à partida realizada (número de gols de ambos os times, tempo de jogo e número da mesa). O RoboGol Ultimate foi idealizado para possuir uma maior iteratividade com os usuários. As melhorias propostas na plataforma de *software* incluem três subprojetos:

RoboGol Server: Aplicação responsável pela comunicação entre o dispositivo móvel e a mesa do RoboGol. Essa aplicação é executada na placa de alta performance localizada na própria mesa.

RoboGol Administrador: Sistema Web responsável por todo o gerenciamento das mesas, participação em eventos, locais físicos, usuários, envio de partidas e configurações. Esta aplicação

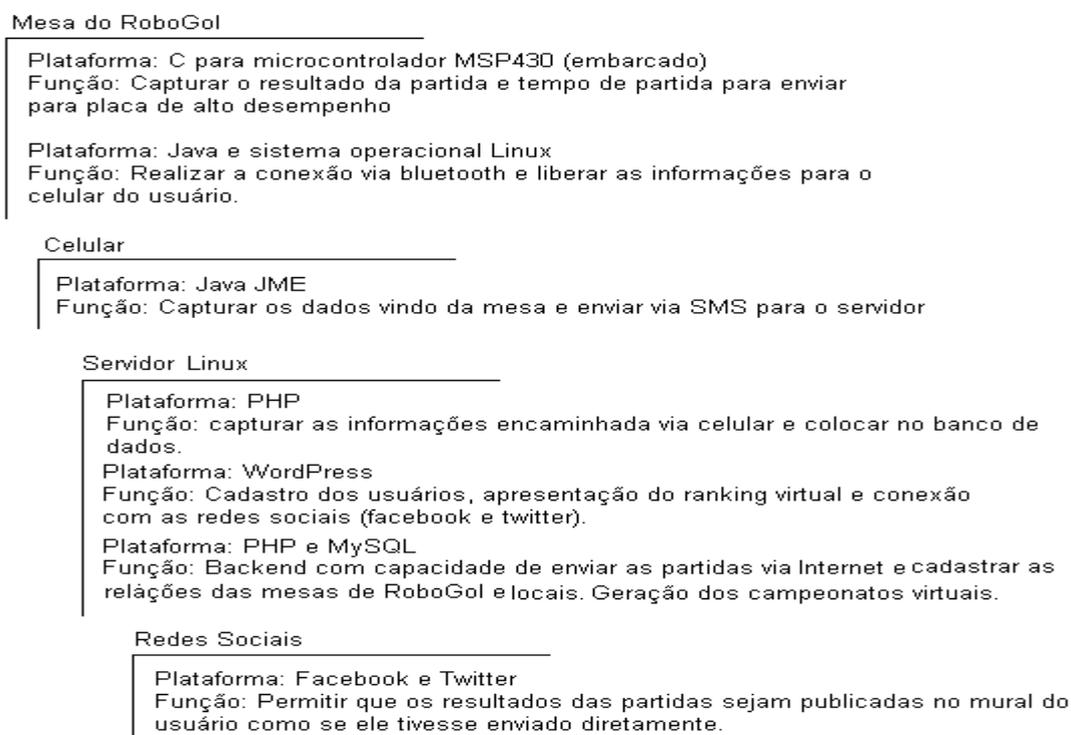
NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

também é responsável por algumas interações com o usuário final por meio do *website*, como o *ranking* geral, o *ranking* do usuário, as “minhas partidas” e as configurações de envio das informações para as redes sociais (facebook e twitter).

RoboGol Mobile: Aplicação para dispositivo móvel (celular ou *smartphone*) responsável pelo envio dos dados da partida recém jogada por meio de um serviço de mensagens curtas. O serviço SMS, conhecidas popularmente como mensagens de texto ou torpedos.

Para melhor entendimento da integração das plataformas tecnológicas envolvidas foi montado um diagrama que pode ser observado na Figura 2.

Figura 2 - Diagrama das plataformas tecnológicas aplicadas ao sistema RoboGol Ultimate.



Fonte: Elaborado pelo autor

2 METODOLOGIA

2.1 DESENVOLVIMENTO DO MÓDULO DE *HARDWARE*

Para a codificação das modificações do *software* embarcado já existente foi utilizada a Linguagem C. Para realizar a codificação, depuração e gravação do mesmo foram utilizadas os programas: ImageCraft IDE for ICC430; e NoICE Debbuger for MSP430. Foi inicialmente realizado um estudo no código original, para fazer uma análise do que deveria ser alterado e em

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

quais arquivos necessitariam ocorrer estas modificações. Logo após os estudos foi iniciada a codificação com a criação de um número de identificação (ID) para cada mesa. Também foi realizada a aquisição dos dados referentes a cada partida executada (número de gols de cada time, assim como o tempo decorrente da partida e o número da mesa). Depois de realizada esta aquisição por meio do *software* embarcado presente no microcontrolador MSP430 (controle da mesa), os dados da mesa são unificados em um vetor de dados que é enviado para um segundo MSP430 que é o responsável, além do controle do som, por também conectar o sistema, por meio de uma serial, a uma placa de alta performance Nano-ITX. Esta placa foi introduzida na nova versão do RoboGol para permitir a conexão com a Internet por meio da comunicação 3G (*chip* de celular).

2.1.1 IMPLEMENTAÇÃO

Na placa-mãe do RoboGol, há dois MSP's sendo que um é do controle da mesa e o outro apenas para o controle do som. Neste segundo MSP430 existe implementado uma comunicação serial RS232C. Realizada a análise inicial no circuito elétrico da placa-mãe, pode-se constatar que os MSP430 são interligados por seis terminais como pode ser visto na Tabela 1, e que a comunicação serial foi desenvolvida apenas para o MSP430 de som.

Tabela 1 - Descrição das ações e reações.

MSP430 de controle	Nome da Ligação	Msp430 de som
P4.0	SOM_DATA0	P1.0
P4.1	SOM_DATA1	P1.1
P4.2	SOM_DATA2	P1.2
P4.3	SOM_DATA3	P1.3
P4.4	SOM_DATA4	P1.4
P4.5	CLK	P1.7

Fonte: Elaborado pelo autor.

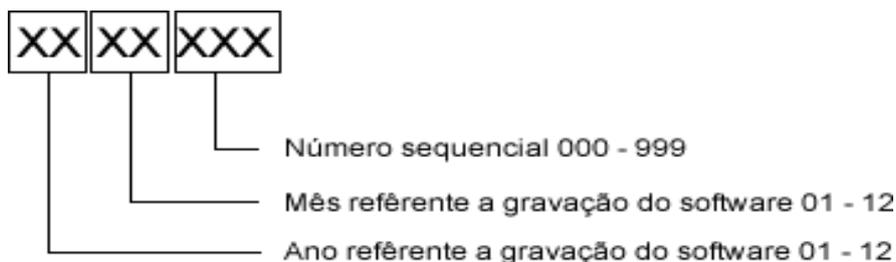
A partir dessa informação foi realizado um estudo na documentação da Texas Instrument (Texas, 2015) em conjunto com o esquema elétrico. Foi verificado que os MSP430 possuem terminais com protocolos para comunicação UART, SPI e I²C, para realizar a comunicação necessária, porém os terminais que possuem estes protocolos já estavam sendo utilizados para outras finalidades dentro do projeto original do RoboGol.

Durante a realização do estudo foi verificado também que a codificação do *software* embarcado da mesa estava com trechos comentados, facilitando o entendimento do mesmo. Foi

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

desenvolvido um protocolo próprio para comunicação dos MSP's, onde o MSP430 de controle da mesa envia comandos para o MSP430 de som solicitando qual som deveria ser tocado. Mediante este fato realizou-se uma análise neste protocolo para verificar quais seriam as possibilidades de mudanças e o que estas mudanças acarretariam no restante do *software* e no funcionamento do produto. Verificou-se que haveria a necessidade de modificação para receber os dados, pois o protocolo teria que distinguir quando seria comando para som e quando seria comando de dados da mesa. Identificou-se também que já havia uma estrutura de comunicação serial, e que necessitaria apenas de alguns ajustes para que fosse realizada a comunicação. Para a realização de um *ranking* dos melhores jogadores ou melhores duplas, viu-se a necessidade da mesa ter uma identificação, sendo assim foi desenvolvido um padrão para um número de identificação da mesa como é mostrado na Figura 3.

Figura 3 - Estrutura padrão do número da mesa.



Fonte: Elaborado pelo autor.

Este número da mesa é gerado no momento em que é gravado, inicialmente, o microcontrolador, não podendo ser alterado após a gravação. Deve ser inserido na variável:

```
const unsigned char _numero_mesa[] = "XXXXXXXX";
```

Como foi dito acima, não é possível alterar este número, mas pode ser acessado para visualização. Foi inserida uma nova opção no menu “Informações do Sistema”, chamada “Número da Mesa” onde consta o ID. Para realização deste desenvolvimento houve a necessidade de alteração nos arquivos: “Menu.c” e “Gui.c”.

Depois de criado o número da mesa, iniciou-se a aquisição dos dados da partida, onde foi necessário fazer a aquisição do número de gols do Time A, número de gols do Time B, tempo da partida, e número da mesa (ID). O número de gols de cada time e o tempo da partida já existiam, pois são os mesmo apresentados no placar digital da mesa. Foi realizada a aquisição destes dados a partir das variáveis: `gASCIIfolA[0]` `gASCIIfolA[1]` `gASCIIfolB[0]` `gASCIIfolB[1]`; `gASCIIminutos[0]` `gASCIIminutos[1]` `gASCIIfegundos[0]` `gASCIIfegundos[1]`

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

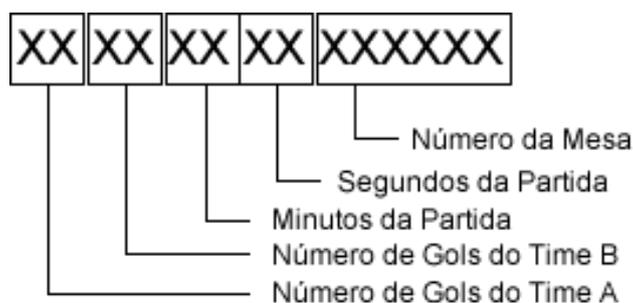
Depois de realizada a aquisição dos dados foi unificada juntamente com o número da mesa em um vetor de dado. Esta composição no vetor foi montada e pode ser vista na Tabela 2. Para que se possa ter uma melhor visualização deste vetor deve-se observar a Figura 4 que apresenta a sua estrutura.

Tabela 2 - Representação do código-fonte para a criação do vetor de informações da partida.

Código	Descrição
<code>dados_partida[0]=insere_zero(gASCIIfgolA[0],16);</code>	1º Char dos gols do Time A
<code>dados_partida[1]=insere_zero(gASCIIfgolA[1],17);</code>	2º Char dos gols do Time A
<code>dados_partida[2]=insere_zero(gASCIIfgolB[0],18);</code>	1º Char dos gols do Time B
<code>dados_partida[3]=insere_zero(gASCIIfgolB[1],19);</code>	2º Char dos gols do Time B
<code>dados_partida[4]=insere_zero(gASCIIfminutos[0],20);</code>	1º Char dos min. da Partida
<code>dados_partida[5]=insere_zero(gASCIIfminutos[1],21);</code>	2º Char dos min. da Partida
<code>dados_partida[6]=insere_zero(gASCIIfsegundos[0],22);</code>	1º Char dos seg. da Partida
<code>dados_partida[7]=insere_zero(gASCIIfsegundos[1],23);</code>	2º Char dos seg. da Partida
<code>dados_partida[8]=_numero_mesa[0];</code>	1º Char Número da Mesa
<code>dados_partida[9]=_numero_mesa[1];</code>	2º Char Número da Mesa
<code>dados_partida[10]=_numero_mesa[2];</code>	3º Char Número da Mesa
<code>dados_partida[11]=_numero_mesa[3];</code>	4º Char Número da Mesa
<code>dados_partida[12]=_numero_mesa[4];</code>	5º Char Número da Mesa
<code>dados_partida[13]=_numero_mesa[5];</code>	6º Char Número da Mesa
<code>dados_partida[14]=_numero_mesa[6];</code>	7º Char Número da Mesa

Fonte: Elaborado pelo autor.

Figura 4 - Estrutura padrão do número da mesa.



Fonte: Elaborado pelo autor.

Para realizar o envio dos dados da mesa foi necessário que se fizesse uma modificação no protocolo de envio dos comandos de som, para que o mesmo enviasse ao final da partida os dados referentes à partida. Para o envio dos comandos de som o MSP430 de controle da mesa utiliza um protocolo de comunicação. Este protocolo nada mais é do que a função *playSample()*, onde esta função envia um valor pré-definido no arquivo “main.h”. Este envio é feito juntamente com um comando para iniciar o toque do som. O envio destes dados é realizado *nibble a nibble*, ou seja, são transmitidos quatro bits por vez. Como um dado possui oito bits, a cada dois envios, tem-se um

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais.** Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

dado completo sendo enviado. Ao transmitir o primeiro *nibble* e setado o terminal de *clock* do protocolo, em seguida é enviado o segundo *nibble*. Este envio do comando era solicitado por meio do comando *playSample* (“*Sample*”), onde o “*Sample*” é um dos *define* referentes ao som.

A seguir pode-se observar um exemplo desta chamada:

```
switch(gsom_random & 0x03){
case 0b00:      playSample(_FIM_JOGO1);      break;
case 0b01:      playSample(_FIM_JOGO2);      break;
case 0b10:      playSample(_FIM_JOGO3);      break;
case 0b11:      playSample(_FIM_JOGO4);      break;
}
```

Neste caso, o comando está fazendo um *switch* para que a cada fim de jogo seja tocado um som diferente. Para que fosse possível realizar o envio dos dados da partida, houve a necessidade de uma modificação neste protocolo, ou seja, na função *playSample*. Para esta modificação foi inserida no arquivo “main.h” o seguinte define:

```
#define _dados_partida 0x0F
```

Este *define* funciona como um *sample*, mas ao invés de determinar o som que será tocado, ele é utilizado para diferenciar os sons dos dados que irão ser enviados. Para início da modificação foi realizada a inserção do argumento *unsigned char dados* na função *playSample* deixando-a da seguinte maneira:

```
void playSample (unsigned char sample, unsigned char dados)
```

Com esta modificação a sua chamada teve a necessidade de ser alterada para *playSample* (“*Sample*”, “*dados*”), onde “*dados*” são as informações da partida que necessitam ser enviados para o MSP430 de som. Porém, este argumento apenas será inserido quando for realizado o envio dos dados. Nas demais chamadas é atribuído o valor 0x00. Assim que terminar a partida é atribuído a cada posição do vetor “*dados_partida*”, o caractere referente aos dados da partida. Logo após é realizada a chamada da função a cada 200ms enviando um caractere por vez. Para o envio destes caracteres foi utilizado o padrão ASCII, como os dados a serem enviados são apenas números, foi utilizado à parte da tabela ASCII apenas dos números como pode ser visto na Tabela 3.

Tabela 3 - Descrição das ações e reações.

Glifo	Binario	Decimal	Hexadecima I
0	00110000	48	30

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais.** Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

1	00110001	49	31
2	00110010	50	32
3	00110011	51	33
4	00110100	52	34
5	00110101	53	35
6	00110110	54	36
7	00110111	55	37
8	00111000	56	38
9	00111001	57	39

Fonte: Elaborado pelo autor.

Com isto, a cada final de partida é chamada a função *playSample()*, da seguinte maneira:

```
playSample(_dados_partida,dados_partida[0]);
```

É enviado o *sample* de dados e o dado correspondente à partida. Assim que for chamada a função, a mesma verifica se está sendo enviado um *sample* de som ou de dados. Caso seja de dados é enviado o caractere correspondente ao dado sendo que foi mantida a forma de envio de um *nibble* por vez. Após ter sido realizado o envio dos dados da partida, é feita a leitura e envio dos dados para a placa de alta performance Nano-ITX por meio da comunicação serial.

No MSP430 de som existe uma rotina de interrupção que assim que percebe o pulso de *clock* ativado pelo protocolo de envio, e inicia a leitura dos dados. Ao receber o *sample* de som, a função executa a leitura de cada *nibble*, e depois de realizada esta leitura o sistema unifica ambos os *nibbles* por meio da linha de comando:

```
sample_idx = ((tmpA & 0x01) << 4) | (tmpB & 0x0F);
```

Junto a este envio estão sendo recebidos no final da partida, os dados referentes à mesma. Também foi incluída logo após estes dados, outra linha de comando com a mesma função modificando somente a variável que esta recebendo a informação, como pode ser visto a seguir:

```
dados_idx = ((tmpA & 0x0F) << 4) | (tmpB & 0x0F);
```

Foi utilizada essa forma para separar os dados e os comandos de som, pois a variável “*sample_idx*” é utilizada apenas para as linhas de comando referente ao som. A variável “*dados_idx*” é utilizada para as linhas de comandos referentes aos dados da partida. Para este tratamento dos dados da partida foi inserido no código já existente um “*switch – case*” onde verifica se o dado que foi recebido corresponde à tabela ASCII na parte dos números. Tendo esta

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

correspondência positiva, é enviada para a porta serial, pelo comando serialOUT("X"), o número decimal corresponde ao código ASCII recebido. Este “*switch – case*” pode ser visto a seguir:

```
switch(dados_idx){
    case 0x30:
        serialOUT("0");
        break;
    case 0x31:
        serialOUT("1");
        break;
    case 0x32:
        serialOUT("2");
        break;
    case 0x33:
        serialOUT("3");
        break;
    case 0x34:
        serialOUT("4");
        break;
    case 0x35:
        serialOUT("5");
        break;
    case 0x36:
        serialOUT("6");
        break;
    case 0x37:
        serialOUT("7");
        break;
    case 0x38:
        serialOUT("8");
        break;
    case 0x39:
        serialOUT("9");
        break;}

```

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

Quando terminada a partida, o MSP430 de controle da mesa envia para o MSP430 de som, os dados da partida que são imediatamente enviados para porta serial que encaminha para placa de alta performance Nano-ITX.

2.1.2 ROBOGOL SERVER

Desenvolvido na tecnologia *Java Second Edition* (JSE), esta aplicação intermedia a comunicação da mesa com o dispositivo do usuário. A mesma é executada na placa de alta performance da mesa do RoboGol. Trata-se de um dispositivo similar a um microcomputador convencional que roda o sistema operacional Linux em um *hardware* chamado Compact Flash Disk. As classes do pacote “*br.com.robogolultimate.server*”, formam todo o sistema do RoboGolServer:

- BluetoothServer: classe para envio e recebimento de dados do dispositivo pareado ou conectado, a classe é responsável pela conexão com o aparelho *bluetooth*, e cuida para que somente um aparelho possa se conectar a mesa. Esta classe é instanciada junto com a inicialização do aplicativo e só irá parar quando o aplicativo também parar de ser executado. Conforme o aparelho se conectar, irá ser gerado um código de conexão, que é a identificação do aparelho que está conectado. Esse código contém somente números.
- Mensagem: classe que contém somente atributos estáticos para identificar e organizar melhor os tipos das mensagens que irão chegar e ser enviados para o dispositivo do usuário que estiver conectado ou tentando se conectar.
- SerialComm: classe importante para o recebimento dos dados da partida. Fica como um ouvinte “escutando” a serial conectada com a placa mãe do RoboGol. Quando a partida terminar, a placa mãe do RoboGol irá enviar a partida para a serial, então neste momento a classe irá escutar esses dados tratá-los e enviar para a instância da classe BluetoothServer a partir do método com a seguinte assinatura *public void enviarMsg (String msg)*. A classe SerialComm possui um padrão *Singleton* que em engenharia de *software* quer dizer que só poderá haver uma instância da classe na memória. A partir do método *getInstance()* é possível adquirir a instancia da mesma. Importante ressaltar que o dado ou mensagem que a mesa envia via serial tem o tamanho de 15 caracteres no formato, 101104001105001, onde:

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

- 10: total de gols time A;
- 11: total de gols time b;
- 0400: (04) minutos e (00) segundos - duração da partida;
- 1105001: número da mesa.

Antes do envio da mensagem pelo *bluetooth* para o dispositivo celular conectado, são inseridos caracteres adicionais na *string* entre os caracteres de duração da partida e a mesa, tais como: hora, minuto, dia, mês e ano, sem nenhum espaço ou qualquer outro tipo de caractere. A classe também apresenta um mecanismo de *timeout* para limpar a mensagem caso venha algum dado não condizente com os dados da partida. A classe *SerialComm* implementa a interface *SerialPortEvent Listener* para a manipulação da serial por meio do método com a assinatura *public void serialEvent (SerialPortEvent event)*.

- *Config*: classe responsável pelo arquivo de configuração do sistema que está presente no mesmo diretório da aplicação.
- *Log*: classe responsável pelo arquivo de log do sistema, ou seja, arquivo onde é registrada toda e qualquer atividade da aplicação. Essa classe se torna indispensável caso algum usuário queira verificar algum erro ou alguma informação importante da aplicação.

Para o aplicativo *RoboGolServer* ser executado a tecnologia Java deverá estar instalada juntamente com as APIs *RXTXComm* e *BlueCove*.

2.1.3 ROBOGOL ADMINISTRADOR

Este subprojeto do *RoboGol* foi desenvolvido com o *framework* *Zend 2*, utilizando a linguagem de programação *PHP*. O objetivo do *RoboGol* Administrador é a criação de um espaço *Web*, integrado ao site *www.robogol.com.br* onde é possível gerenciar informações sobre as mesas de *RoboGol*, como locais dos eventos ou torneios. O ambiente de administração também permite o gerenciamento de pessoas com acesso administrativo, as quais podem gerenciar todas as informações e as pessoas categorizadas como monitores, os quais, só podem realizar envios de partidas para uma determinada mesa de *RoboGol*.

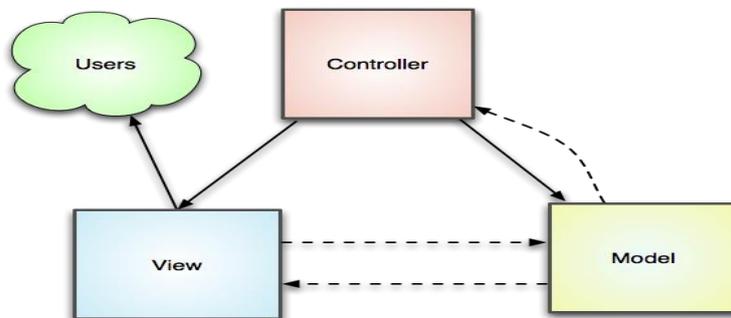
Fica a cargo do ambiente administrativo o disparo das mensagens de e-mail, tweets e mensagens para o facebook, assim como a configuração das mensagens a serem enviadas a cada

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031
rede social. O disparo só acontece após o envio dos dados de uma partida de RoboGol pelo ambiente administrativo.

Com a escolha do *framework* Zend, a arquitetura utilizada para o desenvolvimento do subprojeto foi o MVC4. A maioria do código de aplicações Web está sob uma das três categorias: apresentação, lógica de negócios e acesso aos dados. Nos modelos do padrão MVC o resultado final é um código-fonte de apresentação que pode ser consolidado em uma parte de sua aplicação com sua lógica de negócio em outra parte e seu código de acesso aos dados também em outra.

Os três elementos do MVC são apresentados na Figura 5, tem-se:

Figura 5 - Representação das passagens das informações entre as camadas MVC.



Fonte: Elaborado pelo autor.

- *Model* - É a parte da aplicação que define sua funcionalidade básica por trás de um conjunto de abstrações. Rotinas de acesso a dados e lógica de negócio podem ser definidos no modelo.
- *View* - Definem exatamente o que será apresentado ao usuário. Geralmente os controladores passam os dados para cada apresentação interpretá-los em algum formato. Apresentações também coletam frequentemente os dados do usuário. Em uma aplicação Web, a apresentação é basicamente formada por HTML5 ou outros recursos visuais.
- *Controller* - Controladores vinculam todo o padrão em um conjunto. Manipulam os modelos, decidem qual apresentação será exibida baseada em solicitações do usuário e em outros fatores, repassam os dados que cada apresentação necessita, ou transfere completamente o controle para outro controlador.

2.1.4 AMBIENTE ADMINISTRATIVO – *BACKEND*

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

Para a criação do *ranking* mencionado anteriormente, é necessário armazenar as informações das partidas realizadas em uma determinada mesa de RoboGol, junto com as pessoas que realizaram a partida e o local. Para gerenciar estas informações, foi desenvolvido o ambiente administrativo.

A primeira tela do RoboGol Administrador, é onde se faz a identificação com *login* e senha. Esta identificação irá diferenciar se o usuário é de nível administrador ou de nível monitor. Após a identificação do usuário, é apresentada ao mesmo, a barra superior com as informações que podem ser manipuladas. A barra sofre alterações, dependendo do perfil do usuário, um usuário moderador, só poderá enviar partidas para a mesa no qual o mesmo foi cadastrado por um usuário administrador. Com apenas o nível administrativo, é possível gerenciar os usuários que terão acesso ao *backend*. Nesta tela, é possível adicionar novos usuários, definindo seu nome, senha, seu papel (administrador ou monitor) e as mesas que este usuário poderá enviar as partidas. Também é possível editar as informações de um usuário existente ou removê-lo do sistema.

Na parte de gerenciamento das mesas (Figura 6) são listadas as que já são cadastradas. Cada mesa tem um código, um nome de batismo, que é utilizado para distinguir as mesas com o dispositivo móvel, o seu tipo (2 ou 4 jogadores), além da possibilidade de habilitar a opção do robô maluco, o qual apresenta movimentos aleatórios, permitindo jogar o RoboGol sozinho.

Figura 6 - Tela de gerenciamento das mesas.



Código da Mesa	Nome de Batismo	Tipo da Mesa	RoboMaluco	Ativo	
1105005	Vila	1 x 1	Desativado	<input type="checkbox"/>	 
1105002	Mesa SMS	1 x 1	Desativado	<input type="checkbox"/>	 
1105001	Teste 2	1 x 1	Desativado	<input type="checkbox"/>	 
1105000	Teste 4	2 x 2	Desativado	<input type="checkbox"/>	 

Fonte: Elaborado pelo autor.

A Figura 7 apresenta a lista de todas as partidas enviadas pelo sistema e é possível realizar o filtro de observar as partidas enviadas por um determinado usuário. A Figura 8 apresenta a tela de envio de partida via o RoboGol Administrador, esta tela é acessada via o botão "Enviar nova partida" disponível na Figura 7. A primeira escolha que deve ser realizada antes de enviar uma partida é a escolha da mesa, esta escolha é limitada dependendo do perfil do usuário, a hora do jogo

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais.** Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031 e a duração da partida (valor padrão de três minutos). Após a escolha da mesa é verificada as informações da quantidade dos jogadores (2 ou 4 jogadores) e apresentado os próximos campos a serem preenchidos pelo usuário que enviará a partida. As informações são o *login* do usuário de cada time e o número de gols de cada time.

Figura 7 - Lista das partidas enviadas via RoboGol Administrador.

Data	Mesa	Hora	Duração	Jogador 1 A	Jogador 2 A	Jogador 1 B	Jogador 2 B	Gols A	Gols B	
2012-05-31	Teste 2	12:01	3	natan		eduardo		5	3	
2012-05-31	Teste 4	12:00	4	natan		eduardo		3	3	
2012-05-31	Mesa SMS	11:58	9	natan		eduardo		1	8	
2012-05-31	Mesa SMS	11:56	3	natan				5	6	
2012-05-31	Mesa SMS	11:55	4	natan		eduardo		2	3	
2012-05-31	Mesa SMS	11:53	4	natan				2	3	
2012-05-31	Mesa SMS	11:49	4	natan				2	3	
2012-05-31	Mesa SMS	11:47	5	natan		eduardo		4	2	
2012-05-31	Mesa SMS	11:44	4	natan				2	3	
2012-05-31	Mesa SMS	08:43	0					0	2	

Fonte: Elaborado pelo autor.

Figura 8 - Tela de envio de partida do RoboGol.

Enviar nova partida

Código das Mesas:

Hora do jogo:

Duração da partida:

Time A

Jogador 1:

Número de Gols A:

Time B

Jogador 1:

Número de Gols B:

Fonte: Elaborado pelo autor.

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

O texto das mensagens disparadas pelo RoboGol Administrador são configuradas na opção de "configurações" (Figura 9). Ao configurar as mensagens a serem disparadas, é possível incluir algumas informações que variam dependendo dos dados de cada partida, como por exemplo, o nome dos jogadores, o local da partida, data e número de gols. Além disso, é possível configurar o envio dos resultados das partidas para a página do facebook ou para o twitter de cada jogador (Figura 10). Na Figura 11 é apresentando exemplos de mensagens enviados pelas redes sociais.

Figura 9 - Tela de configuração das mensagens de e-mail e redes sociais do RoboGol.

The screenshot shows the 'Configurações' (Settings) page of the RoboGol Administrador. The page has a dark header with navigation links: 'RoboGol Administrador', 'Usuários', 'Torneios', 'Locais', 'Eventos', 'Mesas', 'Vínculo Mesas', 'Enviar Partida', and 'Configurações' (highlighted). The main content area is titled 'Configurações' and includes a 'Redes do Cliente' button. There are three main sections for configuration:

- Enviar e-mail ao jogador?** with 'SIM' and 'NÃO' buttons.
- Título e-mail ao jogador** with a text input field containing 'RoboGol - Partida'.
- E-mail ao jogador após o término da partida:** A large text area containing a sample email message:

```
Caro ##NOME##,  
  
Parabéns, você jogou uma partida de RoboGol!  
Veja seu ranking no endereço: www.robogol.com.br  
  
Um forte abraço,  
Equipe RoboGol - rumo a copa de 2014!
```

Below the email configuration, there are two more sections for social media:

- Tweet enviado para o jogador após o término da partida:** A text input field with the content: 'Acabei de jogar uma partida de #RoboGol com o placar de ##PLACAR## - Local: ##LOCAL##'.
- Texto enviado ao facebook do jogador após o término da partida:** A text input field with the content: 'Acabei de jogar uma partida de RoboGol com o placar de ##PLACAR## - Local: ##LOCAL##'.

At the bottom, there is a 'Link junto com a mensagem do facebook:' section with a text input field containing 'www.robogol.com.br' and a 'Salvar' button.

Fonte: Elaborado pelo autor.

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais.** Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

Figura 10 - Tela de configuração das mensagens enviadas para as redes sociais do cliente que adquiriu uma mesa do RoboGol.



Fonte: Elaborado pelo autor.

Figura 11 - Exemplo de mensagens enviadas para as redes sociais.



Fonte: Elaborado pelo autor.

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

2.1.5 INTEGRAÇÃO COM AS REDES SOCIAIS

O RoboGol Administrador foi desenvolvido, inicialmente, para o gerenciamento das mesas e das partidas. Ao enviar uma partida, o gerenciamento do *ranking* virtual é alimentado e atualizado. Notou-se que as informações de uma determinada partida poderiam ganhar espaço e gerar uma discussão saudável caso estivesse disponibilizada nas redes sociais.

2.1.5.1 FACEBOOK

Para iniciar o desenvolvimento para o facebook, é necessário possuir uma conta na rede social e entrar na página de desenvolvedor. Após acessar a área de aplicativos, disponibilizado na barra superior. As informações *App ID / API Key* e *App Secret* foram alteradas.

Um aplicativo facebook, pode pertencer a seis categorias de configurações específicas:

1. Website com *login* facebook: É necessário criar um aplicativo facebook, caso um determinado *site* queira que o *login*, como por exemplo, para deixar um comentário, seja realizado pelo *login* do facebook;
2. Aplicativo no facebook: Disponibiliza uma página Web dentro de um quadro no próprio facebook. No caso do RoboGol Ultimate é o acesso ao *ranking* (Figura 12) dentro do próprio facebook;
3. Web móvel: Desenvolver um aplicativo Web dentro do ambiente do facebook móvel;
4. Aplicativo nativo para iOS: Publicar via aplicativos do sistema móvel iOS para o facebook;
5. Aplicativo nativo para Android: Publicar via aplicativos do sistema móvel Android para o facebook;
6. Aba de página: Desenvolver uma aba customizada para a página do facebook.

Para o aplicativo do RoboGol Ultimate foi utilizado o *login* no *website*, para ter acesso as informações do facebook via o site do RoboGol. Posteriormente, foi criado um aplicativo no facebook, para uma melhor visualização das informações dos usuários. O aplicativo do RoboGol Ultimate foi desenvolvido utilizando o PHP SDK, pois é a mesma linguagem de programação utilizada para o desenvolvimento do *site* do RoboGol e também para o RoboGol Administrador. Ao acessar as informações do PHP SDK (Facebook, 2015) é possível ter acesso aos tutoriais de

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

instalação e inicialização, ao código-fonte necessário para a comunicação da aplicação desenvolvida com o próprio facebook e as chamadas disponibilizadas pelo SDK. Para o desenvolvimento do aplicativo, foram utilizadas quatro chamadas:

1. `getUser`: Retorna o ID do usuário, caso esteja com sua sessão ligada, ou “0” caso contrário. Foi utilizado para verificar se já tinha usuário ou não;
2. `getLoginUrl`: Para fazer o usuário utilizar o facebook, é necessário acessar a URL gerada por esta chamada, criando a ponte entre o RoboGol e o facebook;
3. `getAccessToken`: Após o usuário estar logado, é necessário armazenar no banco do RoboGol Administrador o *token* do usuário, para poder escrever no seu mural. Este *token* é uma sequência de letras e números não contendo nenhuma informação do usuário e o *token* perde validade caso o usuário troque sua senha, ou desautorize a utilização do aplicativo. Nesses casos, é necessário adquirir um novo *token*;
4. `API`: É utilizado para enviar comandos, como escrever em uma página do facebook e receber informações, como por exemplo, as fotos dos usuários, utilizado no *ranking*.

Figura 12 - Tela do aplicativo RoboGol Ultimate no facebook.

Classificação	P	V	E	D	Gols	J
1 Rafinha	28	9	1	4	39	14
2 meg	27	9	0	0	64	9
3 musstaffar	21	7	0	6	40	13
4 Natanael	16	5	1	2	20	8
5 Igor	12	4	0	1	12	5
6 Poderoso Trown	10	3	1	2	21	6
7 Valerio	10	3	1	2	15	6
8 Léo	10	3	1	1	12	5
9 yuri5*frap	9	3	0	1	14	4
10 regis	9	3	0	1	14	4
11 bruno	9	3	0	1	11	4
12 duds	8	2	2	3	14	7
13 vasco	8	2	2	1	13	5
14 Azeit	7	2	1	1	10	4
15 Papa Tango	7	2	1	2	11	5

Fonte: Elaborado pelo autor.

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

2.1.5.2 TWITTER

Para iniciar o desenvolvimento para o twitter, é necessário possuir uma conta na rede social e entrar na página de desenvolvedor. O *site* inicial oferece acesso ao status atual da API, o *blog* de desenvolvimento com informações recentes, lista de discussões entre os desenvolvedores e a documentação da API. Para acessar as aplicações criadas e mantidas por um determinado usuário, é necessário acessar o submenu disponibilizado no nome do usuário. Ao acessar o *link* das aplicações, é apresentada a lista de aplicações mantidas pelo usuário. Ao acessar um dos aplicativos, é apresentado detalhes do aplicativo. Estes detalhes apresentam as informações técnicas necessárias para uma aplicação acessar os dados mantidos no twitter. Por questão de sigilo e segurança, as informações *Consumer Key* e *Consumer secret* foram ocultadas.

2.1.6 ROBOGOL MOBILE

Trata-se do aplicativo que é instalado dentro do aparelho celular do usuário que deseja enviar os dados da partida da mesa. A primeira versão foi desenvolvida na plataforma Java J2ME, visando abranger o maior número de dispositivos móveis. Esta versão foi descontinuada e substituída pelo desenvolvimento em ambiente Android.

Os fabricantes de celular oferecem para cada dispositivo uma especificação de JSR (*Java Specification Requests*), que é um grupo de APIs disponíveis no *Java Virtual Machine*. A comunicação *bluetooth* foi escolhida pelo fato de estar presente nos dispositivos celulares vendidos. A conexão entre a mesa e o celular é realizada a partir da escolha da mesa, onde primeiramente faz-se uma busca no alcance do *bluetooth* do aparelho do cliente, mostrando todas as mesas disponíveis e é exibido ao usuário para que escolha a mesa em que irá jogar.

Quando se inicia a conexão entre a mesa e o celular, abre-se um canal de comunicação entre ambos na forma de *InputStream in* de entrada e *OutputStream out* de saída dos dados, canal de passagem dos *bytes*. Para saber qual dispositivo está conectado é enviado um código de conexão para mesa, caso haja alguma desconexão, é enviado um código da mesa, assim podendo a mesa deixar com que haja uma nova conexão, visto que a mesa só pode aceitar um aparelho conectado. A partir da conexão é iniciada a partida na mesa, então o RoboGol Mobile irá ficar como um ouvinte escutando a mesa, e ao termino a mesa irá enviar os dados restantes da partida para serem atribuídos na aplicação. Após o termino do processamento do texto enviado para o celular, é gravado em um

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

local de armazenamento do aparelho controlado pela API RMS, que primeiro é serializado, e posteriormente, gravado.

Ao final da gravação dos dados da partida é realizada uma pergunta ao usuário na forma de formulário com as opções de escolha sim e não, caso o usuário queira enviar os dados para o *site*. Caso a escolha fosse sim, era gerado um SMS com os dados da partida com 160 caracteres, por meio do método smsText, posteriormente, enviado pelo método “enviar” da classe SMSService, passando como parâmetro o texto gerado pelo método smsText. Enviado o SMS, é feita uma pergunta ao usuário se deseja jogar novamente, com opções de escolha sim ou não. Caso a escolha seja sim, o fluxo de comunicação permanece aberto, e inicia-se novamente uma partida, caso contrário, é gerado um pedido de desconexão para a mesa e limpando o fluxo de comunicação assim como o código da conexão, e o usuário é direcionado ao menu principal da aplicação do RoboGol Mobile no seu aparelho celular.

Em linhas gerais, o aplicativo deve ser instalado no celular do usuário para que ele possa interagir com todas as funcionalidades que são oferecidas pelo RoboGol. É importante comentar que a escolha da tecnologia JAVA (JME) foi devido a grande maioria da faixa etária de usuários do RoboGol ter esta plataforma disponibilizada nos seus celulares. A seguir são apresentados os passos que o usuário deve proceder para poder enviar as partidas do RoboGol pelo celular.

Instalar o aplicativo (RoboGolMobile) no aparelho celular. Assim que iniciar o aplicativo aparecerá uma tela com duas opções: “options” e “sair”. Para terminar a aplicação, clique em sair, se deseja jogar, clique em “options”. Aparecerá uma tela onde o usuário poderá escolher entre ver as partidas jogadas, no item “Partidas Jogadas”, ou jogar uma nova partida, clicando em “Iniciar”. Se o seu aparelho não é sensível ao toque, você deve clicar na opção “Select” para escolher a opção desejada.

Após escolher “Jogar”, será perguntado ao usuário se deseja jogar com as mesmas configurações do jogo anterior, se escolher “sim” para esta opção, o sistema iniciará a procura por mesas disponíveis para se conectar via *bluetooth*, se escolher “não”, o aparelho iniciará a procura por mesas disponíveis. Ao encontrar alguma mesa, o sistema irá mostrar na tela para que o usuário escolha qual mesa irá se conectar.

O usuário terá que optar por qual time irá jogar. Existem duas opções, time A e time B. Após selecionar um dos times lhe será perguntado se seu adversário será o Robô Maluco. Caso seja, o usuário deverá entrar com seu *login* e senha. Caso esteja jogando em dupla, entrar também com o *login* e senha do colega. Caso não seja o adversário o Robô Maluco, deverá entrar com o *login* e senha do time adversário que pode ser composto de até dois jogadores. Assim que o último jogador

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

for inserido, o aplicativo mostrará uma mensagem: "Partida em andamento" e está pronto a conexão entre a mesa do RoboGol e o seu celular.

Após o termino da partida, a mesa enviará os dados dos jogadores e da partida para o aplicativo do celular, e lhe será perguntado se deseja enviar os dados para o *site* www.robogol.com.br via SMS. Se desejar enviar os dados, os mesmos serão publicados e participarão de um *ranking* e o usuário receberá um e-mail comentado que participou de uma partida de RoboGol. Além disso, caso o usuário tenha cadastrado sua rede social (facebook), o resultado da partida será postado em seu mural. No *site* do RoboGol, o usuário poderá conferir os dados de suas partidas e seu *ranking*. É importante comentar que o usuário precisa se cadastrar no *site* para poder ter seu ranking computado.

Os dados presentes na mensagem SMS enviada de um celular de um determinado jogador de RoboGol, é enviado para um número pré-definido que está conectado a um computador. Para fazer esta passagem das informações via SMS, o computador precisa estar equipado com um modem GSM/GPRS. Com os dados do SMS dentro de uma variável na linguagem PHP, é possível enviar e povoar o banco de dados utilizado pelo RoboGol Administrador, também desenvolvido em PHP. Com os dados da partida enviados via SMS inseridos no banco de dados, os mesmos são computados com os outros dados gerados via o próprio sistema para a geração atualizada do *ranking* e dos disparos das informações para as redes sociais.

3 CONSIDERAÇÕES FINAIS

O projeto obteve êxito no seu desenvolvimento que tinha como foco a integração do game RoboGol com a Internet e as redes sociais por meio dos celulares. Esta integração exigiu uma equipe multidisciplinar que teve que aprender e trabalhar com diversas plataformas computacionais, desde o *software* embarcado dos microcontroladores MSP430 presente nas mesas do RoboGol, passando por Java para Linux da placa de alto desempenho responsável pela comunicação *bluetooth* com o celular; caminhando pela plataforma Java JME do celular e finalizando com a plataforma PHP e banco MySQL no servidor e a programação das redes sociais (facebook e twitter).

Durante o desenvolvimento do projeto RoboGol Ultimate foram tomadas decisões tecnológicas que acabaram não sendo adequadas. Com a escolha inicial do *site* gerenciado por WordPress e o RoboGol Administrador por JSP foi necessária a criação de várias pontes de comunicação entre estas duas linguagens e não foi possível resolver o problema da velocidade da troca de dados. Diante disso, a versão final aplicou-se a linguagem PHP e a base de dados MySQL.

NETTO, Antonio Valerio. **Tecnologia para integração de game robótico de futebol com a internet e as redes sociais**. Revista Interdisciplinar Científica Aplicada, Blumenau, v.12, n.2, p. 84-105, TRI II 2018. ISSN 1980-7031

Outro problema foi à utilização de duas bases de dados distinta. O PostgreSQL com o JSP e o MySQL com o PHP. Após algumas baterias de teste, foi verificado o problema de inconsistência entre os dois bancos. A versão atual utiliza apenas um banco MySQL disponibilizado na Web. Todas as informações necessárias, tanto para o site quanto para o Administrador do RoboGol estão em um mesmo lugar, facilitando o acesso.

Observou-se que a proposta de criar um pagamento on-line pelo celular para jogar o RoboGol, não se tornou viável como produto comercial devido aos custos que seriam atrelados ao sistema. Diante disso, não se viabilizou financeiramente. Com relação à plataforma inicialmente escolhida para *mobile* que foi o Java JME, muitos dos seus serviços foram descontinuados. E o projeto acabou tendo que migrar para o Android 5.1.

Um desafio que foi inesperado esteve relacionado com a utilização de totens para o cadastramento dos usuários para jogar o RoboGol. Observou-se que não se poderia utilizar a mesma plataforma presente no site do RoboGol (www.robogol.com.br) por questões de usabilidade. Nos testes de campo ela se mostrou falha e por isto, a equipe teve que desenvolver outra interface para o cadastro de usuários. Um dos problemas foi à questão do usuário se cadastrar em seu facebook e não fechar a sessão do mesmo. Quando outra pessoa ia se cadastrar usando o totem encontrava o facebook do usuário anterior aberto. O que gerava quebra de segurança. Isto foi solucionado forçando o fechamento da sessão.

4 AGRADECIMENTO

O autor agradece o apoio do CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) por meio do seu programa de Desenvolvimento Tecnológico e Extensão Inovadora (DT) e do programa de Formação de Recursos Humanos em Áreas Estratégicas (RHAE).

5 REFERÊNCIAS

ROBOGOL **Sobre o produto**. Disponível em: <http://www.robogol.com.br/sobre> Acesso em Maio, 2015.

TEXAS. **MSP Mixed Signal Microcontroller**. Disponível em: <http://focus.ti.com/lit/ds/symlink/msp430f149.pdf> Acesso em Julho, 2015.

FACEBOOK. **Developers**. Disponível em: <https://developers.facebook.com/docs/reference/php> Acesso em Julho, 2015.