
SISTEMA INTEGRADO PARA ACOMPANHAMENTO DE PESO EM ACADEMIAS

Murilo Augusto Amorim

Graduando em Sistemas de Informação

Ely Fernando do Prado¹

Mestre em Ciência da Computação

RESUMO

Este trabalho foi desenvolvido com objetivo de realizar a interação entre varias plataformas promovendo a automação do processo de acompanhamento de peso em academias. A principal preocupação do trabalho foi tornar este processo mais intuitivo, fazendo com que os usuários das academias ao se pesar na balança não tenham preocupação em anotar o peso para então analisar o desenvolvimento e sim que tenham em seus dispositivos móveis um aplicativo que faça tal tarefa. Para isso usou-se uma balança residencial conectada a um microcontrolador Arduino que por meio de conexão serial envia dados para um computador, este que gera um QR Code através de uma aplicação desenvolvida em Java. A partir daí, uma aplicação móvel faz a leitura do QR Code,armazena os dados no SQLite construindo então um gráfico em HTML5 contendo os dados de cada pesagem do aluno.

Palavras-chave: Arduino. Dispositivo Móvel. QR Code. Balança, Célula de Carga.

1. INTRODUÇÃO

O aquecimento do mercado *fitness* posiciona o Brasil na segunda colocação no ranking mundial, esse fato somado a grande necessidade de profissionalismo do setor, obteve-se a idéia do desenvolvimento de um produto inovador que, fazendo uso do microcontrolador Arduino juntamente com a linguagem de desenvolvimento Java e o framework para desenvolvimento móvel Phonegap. Dessa forma pode se proporcionar ao usuário da academia um melhor acompanhamento de sua evolução mediante os treinamentos oferecidos pela academia, acompanhamento este que tem foco no peso, pois segundo Tessmer, Pinho e Fassa

¹ elyprado@libertas.edu.br



(2006) grande parte dos frequentadores de academias são insatisfeitos com seu corpo e tem o peso um fator importante para encontrar equilíbrio.

Nos tópicos seguintes, serão demonstradas explicações mais aprofundadas sobre cada uma das tecnologias utilizadas para o acompanhamento do peso do aluno: Arduino, QRCode, bibliotecas para gerar QRCode e Phonegap respectivamente.

2. REFERENCIAL TEÓRICO

Nesta seção serão abordadas algumas referências que tem o objetivo de embasar o processo de automação da pesagem em academias.

2.1. Automação

Automação é um conjunto de técnicas que auxiliam na construção de sistemas ativos que atuam com alta eficiência utilizando-se das informações recebidas do meio em que atuam (ROSARIO, 2009).

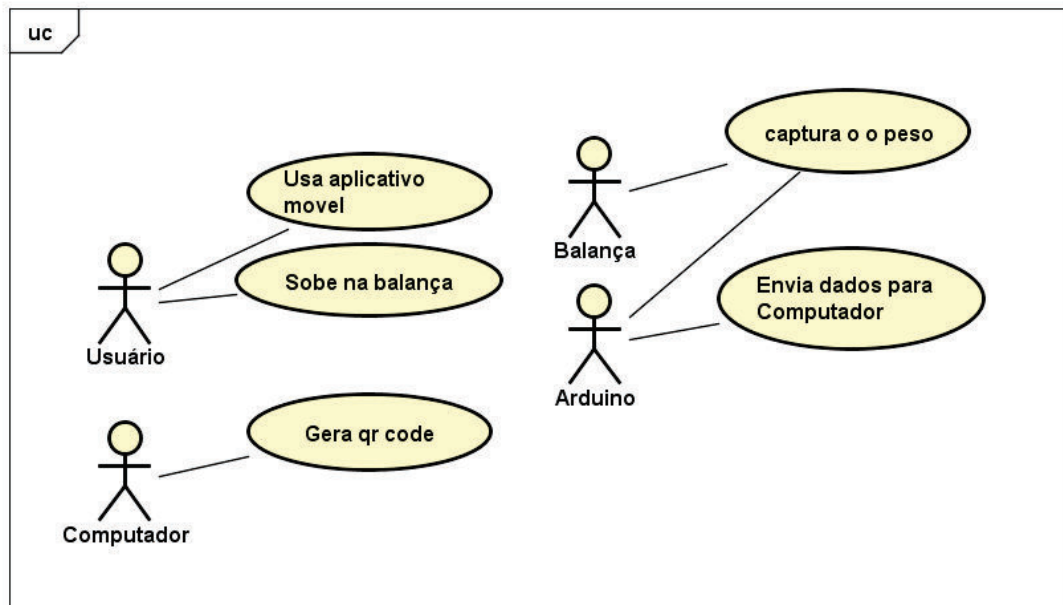
A integração da automação nasceu, durante os anos 20 quando Henry Ford criou a linha de montagem do modelo T, para aumentar a produção, reduzir os custos e garantir a segurança dos operadores que executam as tarefas mais perigosas(ROSARIO, 2009).

Com o sistema de automação residencial é possível controlar a distancia as luminárias, as tomadas, em fim, tudo o que possui controle. E, além disso, o controlador de automação residencial apresenta uma tecnologia que colabora para ligar e desligar dispositivos, colaborando assim para a redução da conta de energia elétrica (MONK, 2013).

No ambiente industrial a automação é a utilização de qualquer dispositivo para controlar maquinas e processos, entre eletrônicos pode-se ser usado computadores ou controladores lógicos que podem executar tarefas possíveis ou impossíveis para seres humanos (ROSARIO, 2009).

De acordo com Lima, Rosa (2013) a automação residencial tem a seguinte característica, ela elimina a dificuldade da localização do operador. Se não fosse assim, as tarefas que se faz hoje com praticidade de velocidade, não seriam tão praticas e rápidas.

3. MATERIAIS E MÉTODOS



powered by Astah

Para desenvolvimento do projeto, foram realizadas pesquisas exploratórias sobre a integração do dispositivo de medição de peso ao Arduino, bem como bibliotecas para a geração do código QRCode, e também sobre Phonegap, o que trará mais informações para o aperfeiçoamento do projeto.

Na seção a seguir será abordado sobre os materiais utilizados na construção do projeto tais como balança, modulo conversor, microcontrolador, QR Code, Phonegap, SQLite e Bootstrap, nesta sequência.

3.1. Balança

Para o projeto em questão foi utilizado uma balança com capacidade minima de 0,3Kg e máxima de 150Kg com dimensões de 24.0 cm x 14.8 cm x 2.5 cm, alimentada por 2 pilhas AAA. A captura do peso pela balança é feita através de 4 células de carga devidamente posicionadas em cada uma das extremidades do dispositivo, conectadas através de um circuito interno.

3.2. Módulo conversor

Para tratar os sinais analógicos enviados pelas células de carga, foi necessário primeiramente fazer a conversão destes para digitais e amplificar sua frequência, para isso usou-se o módulo Conversor 24bit Hx711.

3.3. Microcontrolador

Após conectar as células de carga no conversor, partiu então para a parte em que o microcontrolador vai pegar as informações do conversor, para isso usou-se o microcontrolador Arduino UNO R3 ATmega328P.

A plataforma Arduino teve seu início no Interaction Design Institute na cidade de Ivrea em 2005, onde o professor Massimo Banzi juntamente com um pesquisador visitante criaram a plataforma de desenvolvimento que recebeu o nome Arduino em referência ao bar frequentado pelo corpo docente e estudantes (MCROBERTS, 2011).

Segundo Mcroberts (2011), existem várias versões do Arduino baseadas em um microprocessador de 8 Bits, as primeiras versões usavam o microprocessador ATmega8 com Clock de 16MHZ, mais tarde usariam o ATmega168 com memória flash de 16KB, já as versões mais recentes como: Uno e Duemilanove usam o ATmega328 pronto para projetos mais exigentes existem também o Arduino Mega1280 e o Arduino Mega2560, com memórias de 128KB e 256KB respectivamente, tendo por padrão o C++ como linguagem para desenvolvimento. Cada Arduino possui 14 pinos que podem ser definidos como entrada ou saída e outras 6 para servirem de entradas analógicas.

Tendo em vista que o Arduino possui memória bem limitada, pode-se trabalhar com bibliotecas que são uma coleção de funções que quando adicionadas ao código acrescentam funcionalidades específicas ao projeto, algumas destas bibliotecas podem ser usadas por si próprias, porém existem outras que devem ser usadas juntamente com componentes eletrônicos, que são chamados Shields (MONK, 2015).

As bibliotecas são divididas em três grupos, são eles:

- Essenciais: Bibliotecas que já fazem parte do IDE do Arduino, ou seja, elas são responsáveis por disfarçar a complexidade de se trabalhar com microcontroladores, tornando a programação muito mais fácil e intuitiva.

- Padrão: São bibliotecas já inclusas juntamente com a instalação do IDE, que serão importantes ou ao menos úteis para qualquer um que for desenvolver algum projeto. Para incluí-la em um determinado projeto é necessário adicioná-la com a instrução include no topo do código.
- Terceiros: São bibliotecas desenvolvidas por outros usuários ou desenvolvedores, que ainda de acordo com o Mcroberts (2011), cada biblioteca é composta por um conjunto de funções úteis para determinado projeto, todas as bibliotecas são de código aberto pois ser de código aberto faz parte dos valores da comunidade Arduino. Essas bibliotecas de terceiros podem ser adicionadas ao projeto através do menu Sketch/Importar biblioteca.

Todas essas bibliotecas sendo elas de terceiros ou já embutidas na IDE do Arduino colaboram para que a programação no microcontrolador seja fácil e de rápida codificação, fazendo que qualquer um seja capaz de desenvolver projetos. (MCROBERTS, 2013)

3.4. QR Code

QR Code é um código bidimensional capaz de armazenar dados que podem ser lidos posteriormente através de um leitor instalado em um smartphone.

“O QR Code é um código bidimensional, o que significa que a informação muda horizontal e verticalmente. O leitor de QR Code verifica ambas as posições. Isso permite que um QR Code contenha muito mais informações que um código de barras. Isso também requer um leitor mais sofisticado, porém não se deve preocupar com isso, pois engenheiros de hardware e software já cuidaram disso e você pode fazer download de inúmeros leitores de QR Code para seu smartphone.”(WINTER, 2011, p.18)

Para gerar o QR Code utilizou-se a linguagem de desenvolvimento Java. De acordo com Schildt(2015) a linguagem de programação Java 1.0 foi originalmente lançada em 1995 pela Sun Microsystems e revolucionou a programação. Essa revolução transformou a web de maneira surpreendente, construindo uma web muito mais interativa.

Juntamente com a linguagem Java, usou-se uma biblioteca ou seja, um pacote com funcionalidades ou funções prontas, neste caso a biblioteca se chama Zxing, biblioteca esta que possibilita gerar o QRCode. A biblioteca Zxing é de licença livre para uso, podendo ser encontrada no endereço <https://github.com/zxing/zxing>,

3.5. Phonegap

Para o desenvolvimento da aplicação mobile que fará a leitura do QR Code foi utilizado o framework de desenvolvimento chamado Phonegap que é um projeto open source.

De acordo com Rohit e Yogesh (2012), Phonegap é um framework para desenvolvimento móvel que gera aplicações para diversos sistemas operacionais como IOS, Android, BlackBerry e WebOS, desenvolvimento este que usa linguagens HTML, CSS e Javascript. Isso significa que poderão ser desenvolvidas aplicações para diversos dispositivos cada qual com suas características nativas usando simplesmente as linguagens de desenvolvimento web.

O Phonegap resolve alguns problemas para desenvolvedores como, por exemplo:

Desenvolver aplicações móveis usando as familiares linguagens para web como HTML, CSS e Javascript.

O código pode ser migrado para outra plataforma de maneira rápida e fácil.

O primeiro código no Phonegap foi elaborado em agosto de 2008 no evento IphoneDevCamp na cidade de São Francisco, Califórnia, e o que contribuiu para a sua criação foi o simples fato de que todo o desenvolvimento do código para Iphone é feito em Objective-C o que seria um ambiente desconhecido para desenvolvedores Web. Então poderia alguém desenvolver um framework que proporcionasse a todos os desenvolvedores Web, usar o seu conhecimento em HTML, CSS e Javascript e que permita a interação com importantes partes nativas do Iphone? Surgiu então como um vencedor do concurso, o Phonegap começando também a fornecer suporte para a plataforma android.

3.6. SQLite

O próximo passo, fez-se uso do SQLite para gravação dos dados, este que segundo Kreibich(2010) é um software de armazenamento de licença gratuita de código aberto que fornece um sistema relacional de armazenamento de dados e o Phonegap dá suporte para utilização do SQLite como sistema de persistência de dados.

3.7. Bootstrap

Bootstrap é um framework *open source*, ou seja, de código aberto livre para uso, que de acordo com o site do Bootstrap “é o framework mais popular para desenvolvimento

responsivo móvel em projetos para Web”(http://getbootstrap.com/), site este que ainda disponibiliza o download do framework gratuitamente.

Segundo Abraham(2015) Bootstrap é um conjunto de ferramentas gratuitas para rápida criação de paginas web, criado em 2011 por *Mak Otto* e *Jacob Thornton* para uso interno no Twitter e que logo depois foi liberado para uso popular, antes do Bootstrap o desenvolvimento não possuía um padrão ou seja possuía leves mudanças de tempo em tempo o que dificulta a manutenção do código, atualmente Bootstrap tem se tornado uma das ferramentas mais utilizadas para desenvolvimento de páginas web.

4. MÉTODOS

4.1. Arduino conectado à balança

Primeiramente foi feita a conexão das células de carga, quatro no total, a uma protoboard esta que já estava com o conversor conectado. Em outra protoboard foi feita a conexão com o Arduino, sendo utilizadas as portas analógicas 0 e 1 e para alimentação foram utilizadas as portas GND e 5 v, como e demonstrado na Figura 1.

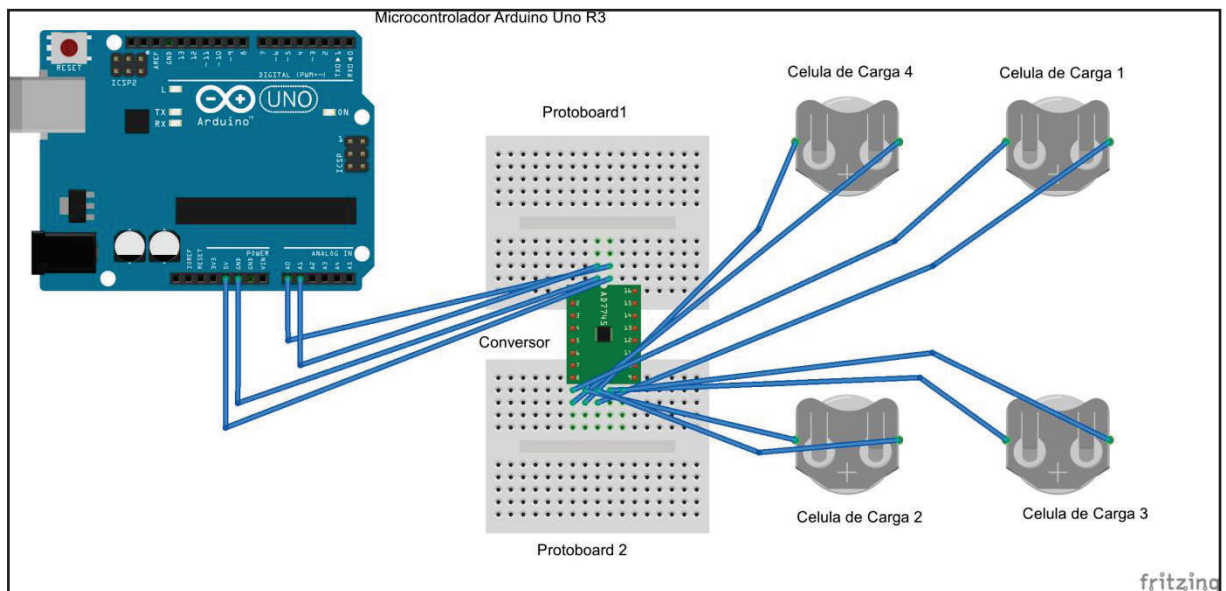


Figura 1: Ligação das células de carga ao arduino

4.2. Arduino conectado ao computador

Para a conexão Arduino/computador é utilizada a conexão serial pois ligar o Arduino ao computador com um cabo USB informa para o sistema operacional de seu computador que

você conectou um dispositivo serial. Quando você instala o IDE do Arduino, você também instala vários drivers que permitem que o Arduino seja reconhecido como um dispositivo que se comunica através de uma conexão serial. (MCROBERTS, 2013)

Para que o computador reconheça os dados enviados pelo Arduino, é necessário configurar essa conexão, configuração esta que neste projeto será feita através de uma aplicação desenvolvida em Java na IDE NetBeans.

No projeto deve ser inserida a biblioteca RXTX e sua dll que vão permitir que o software faça comunicação serial recebendo os comandos, acessando dados e fazendo chamadas de sistema. O arquivo com extensão jar desta biblioteca precisa ser inserido no diretório do sistema: C:/ProgramFiles/java/jre/lib/ext. Já arquivo dll deve ser copiado para o diretório C:/ProgramFiles/java/jdk/bin, para diretório C:/ProgramFiles/java/jre/bin e para C:/Windows/System32. Caso o sistema operacional for de 64bits outra cópia deve ser inserida no diretório C:/Windows/SysWOW64.

```
package supervisorio;
import Serial.Serialrxtx;
* @author Murilo
*/
public class Supervisorio {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Serialrxtx serial = new Serialrxtx();
        if (serial.iniciaSerial()) {
            while (true) {
            }
        } else {
        }
    }
}
```

Figura 2 - Comunicação Serial

Na Figura 2 podemos ver parte do código da aplicação nele encontramos o import da classe chamada Serialrxtx onde está o código de identificação da porta serial e logo abaixo é chamado o método iniciaSerial que inicia a comunicação com a porta serial este que também está na classe Serialrxtx. Após a comunicação do Arduino com o computador, iniciou-se o desenvolvimento da aplicação para gerar o QR Code uma vez que os dados que originam da

balança foram capturados pelo Arduino e enviados por este para o computador via comunicação serial.

Para o desenvolvimento desta aplicação uso-se a biblioteca Zxing que é uma biblioteca em Java para geração de códigos 1D/2D, na Figura 3 podemos observar o código para gerar o QR Code.

```
private void geraQrCode() {
    String peso = serial.getProtocolo().getLeituraComando();

    //pega apenas os numeros da variável
    String texto = "";
    for (int i = 0; i < peso.length(); i++) {
        char letra = peso.charAt(i);
        if (Character.isDigit(letra) || letra == '.') {
            texto = texto + letra;
        }
    }

    lblValor.setText(texto);

    String nomeArquivo = "qrcode.png";
    try {BufferedImage image = null;
        //Gera o QrCode e grava no arquivo
        FileOutputStream file = new
        FileOutputStream(nomeArquivo);
        ByteArrayOutputStream out =
        QRCode.from(texto).to(ImageType.PNG).stream();
        file.write(out.toByteArray());
        file.flush();
        file.close();

        //mostra qrcode gerado
        image = ImageIO.read(new File(nomeArquivo));
        ImageIcon imagelcon = new ImageIcon(image);
        jLabel1.setIcon(imagelcon);
    } catch (Exception ex) {

    }
}
```

Figura 3 - Geração do QRCode

4.3. Leitura do QR Code pelo aplicativo móvel

O próximo passo foi de necessidade do projeto que houvesse um leitor capaz de capturar e interpretar o QR Code gerado, necessidade essa que seria suprida por um aplicativo

móvel. Para o desenvolvimento da aplicação móvel levando em consideração o tempo a ser levado para aprendizagem do desenvolvimento móvel e todas as diferenças entre as plataformas, optou-se por usar o framework Phonegap. Também se considerou a facilidade no desenvolvimento multiplataforma e principalmente pelo fato de todo o desenvolvimento ser feito a partir de linguagens web, as quais em sua grande maioria já fazem parte do conhecimento adquirido na graduação até o período atual.

Para o desenvolvimento com Phonegap é preciso fazer a instalação do kit de desenvolvimento do android que pode ser baixado no endereço <http://developer.android.com/sdk/index.html>, após a instalação deste, deve-se executar o SDK manager, que está no diretório de instalação para atualizar o SDK. O próximo passo é a instalação do Apache Ant que nada mais é do que uma biblioteca java para automatizar a construção dos projetos, os chamados “builds”, o Apache Ant pode ser baixado no site <http://ant.apache.org/bindownload.cgi>.

Seguindo com a criação do ambiente para desenvolvimento com Phonegap, é necessário configurar as variáveis de ambiente, indo até as propriedades do sistema, aba avançado, variáveis de ambiente e variáveis de sistema, criando uma variável de nome ANDROID_HOME e em seu valor coloque o diretório de instalação do SDK, deve ser procurado pela variável com nome Path, editado seus valores para o seguinte valor: %ANT_HOME%\bin;%ANDROID_HOME%\platform-tools;%ANDROID_HOME%\tools;

Concluído a criação de variáveis, é preciso realizar a instalação do Node.js que é um engine Javascript voltado para projetos do Phonegap e pode ser baixado no link <http://nodejs.org/download/>. No Figura 4 podemos ver o código da função scan usada para ler o QR Code. No desenvolvimento da aplicação, o conhecimento adquirido na pesquisa somado ao exemplo encontrado no github.com/wildabeast/BarcodeDemo, lograram para êxito. A principal dificuldade encontrada foi que o device LG L7, com processador um core de 1GHZ, memória de 2,7 GB e sistema Operacional Android 4.0.3 disponível para teste pois estava com sua memória cheia memória estar cheia, não executou o aplicativo de maneira satisfatória, porém em qualquer um dos outros devices testados a aplicação funcionou corretamente, como por exemplo no Smartphone MotoG 2 Geração com processador Quad core de 1,2 GHZ, 16GB de memória e sistema operacional Android 4.4.4.

```
scan: function() {
    console.log('scanning');
    var scanner = cordova.require("cordova/plugin/BarcodeScanner");
    scanner.scan( function (result) {
if (!result.cancelled) {
    alert( "teste: " + result.text);
    inserePeso(result.text);
    alert('inseriu');
}
        }, function (error) {
            console.log("Scanning failed: ", error);
        });
    },
    encode: function() {
        var scanner = cordova.require("cordova/plugin/BarcodeScanner");
        scanner.encode(scanner.Encode.TEXT_TYPE, "http://www.nhl.com", function(success) {
            alert("encode success: " + success);
        }, function(fail) {
            alert("encoding failed: " + fail);
        }
        );
    }
}
```

Figura 4 - Leitura do QRCode

Para que a aplicação seja responsiva, ou seja, funcione em qualquer tamanho de tela em dispositivos Android, Windows Phone e posteriormente também para IOS, fez-se uso do framework Bootstrap. Foram importadas para o projeto as pastas css, img e js, contendo o estilo, diretório para imagens do Bootstrap e Javascript respectivamente, ambos configurados de forma responsiva.

4.4. Armazenamento dos dados e geração do gráfico

A aplicação agora precisaria pegar os dados enviados pelo QRCode e armazenar no dispositivo, funcionalidade esta que foi implementada usando SQLite. No código mostrado na Figura 5 pode-se ver as variáveis peso e db, seguida pelo método que verifica se o banco pode ou não ser inicializado, após esse, a variável db recebe a conexão com o banco e por último a função de populateDB com o parâmetro tx, este que recebe a instrução SQL para criar a tabela no banco caso não exista e inserir um registro contendo duas variáveis.

```
var peso;  
var db;  
if (!window.openDatabase) {  
    alert('Navegador não suporte SQLite.');}  
db = window.openDatabase("Teste", "1.0", "MuriloDB", 20000);  
function inserePeso(p) {  
    peso = p;  
    db.transaction(populateDB, errorCallback, successCB);  
}  
function populateDB(tx) {  
    tx.executeSql('CREATE TABLE IF NOT EXISTS pesos (data, peso)');  
    tx.executeSql('INSERT INTO pesos (data, peso) VALUES (datetime("now"), "' + peso + '");');  
    // alert('INSERT INTO pesos (data, peso) VALUES (datetime("now"), "' + peso + '");');  
}  
function errorCallback(tx, err) {  
    alert("Error processing SQL: " + err);  
}  
function successCB() {  
    alert("success!");  
}
```

Figura 5 - Conexão com o Banco de Dados SQLite

Desenvolvida parte de gravação dos dados obtidos pelo Arduino conectado na balança, como ultima funcionalidade era necessário mostrar esses dados de forma intuitiva para que o projeto alcance seu objetivo que é fazer com que o aluno realmente acompanhe seu peso pelo aplicativo. Para isso fez-se uso da biblioteca CanvasJS, que é uma biblioteca HTML 5 e Javascript para gerar gráficos.

Na Figura 5 pode-se ver o código html5 para gerar o gráfico. Primeiramente é feita a importação da biblioteca canvas.min.js usando a tag <script> como também a construção animada do gráfico grafico.js, é também determinado o tamanho da área do gráfico com a tag div com id chartContainer e é instanciada uma variável de nome peso que recebe os rótulos de cada ponto do gráfico e sua posição no eixo y.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>teste grafico</title>
<script src="js/canvasjs.min.js"></script>
<script src="js/grafico.js"></script>
<link rel="stylesheet" href="">
</head>
<body>
<div>
<button id="randomizeData">Teste</button>
<div id="chartContainer" style="height: 300px; width: 100%">
</div>
</div>
<script type="text/javascript">
var dados =
[
{ label: "primeiro", y: 450 },
{ label: "segundo", y: 414},
{ label: "terceiro", y: 460 }
];
dados.push( { label: "quarto", y: 510 } )
grafico(dados);
</script>
</body>
</html>
```

Figura 6- Geração do gráfico de peso

5. RESULTADOS

O projeto fará a automação do acompanhamento de pesagem de usuários de academias realizando interação entre plataformas onde o Arduino pegará a informação sobre o peso originadas da balança, passando para o computador, este que através de um sistema em java vai gerar um QR Code contendo essas informações. Uma aplicação móvel então fará a leitura do QR Code armazenando a informação do peso no banco de dados SQLite e finalmente usando HTML 5 será gerado um gráfico mostrando os dados passados pela balança. Na Figura 7 pode-se observar melhor como se dá a integração entre as diversas partes que compõe o sistema.

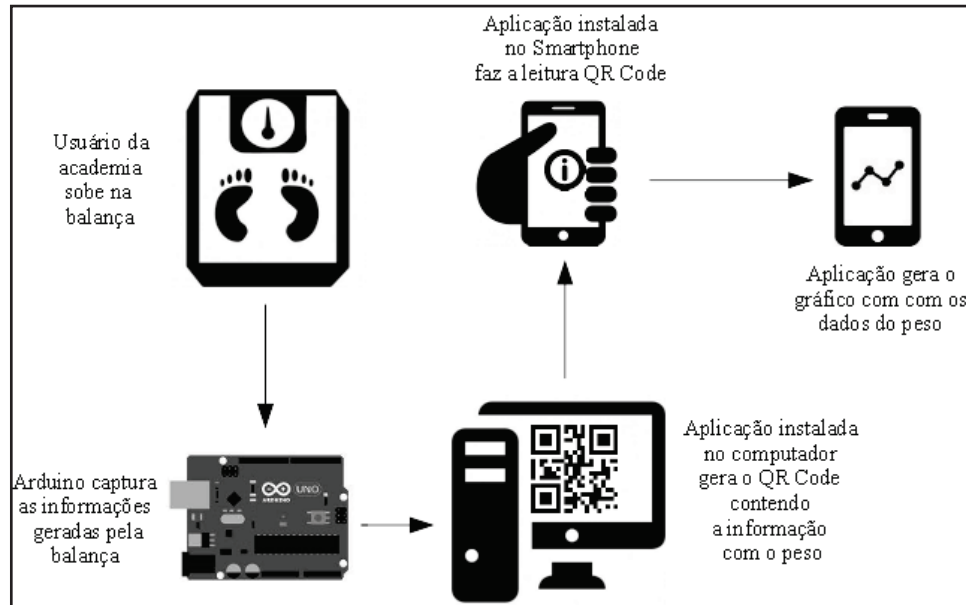
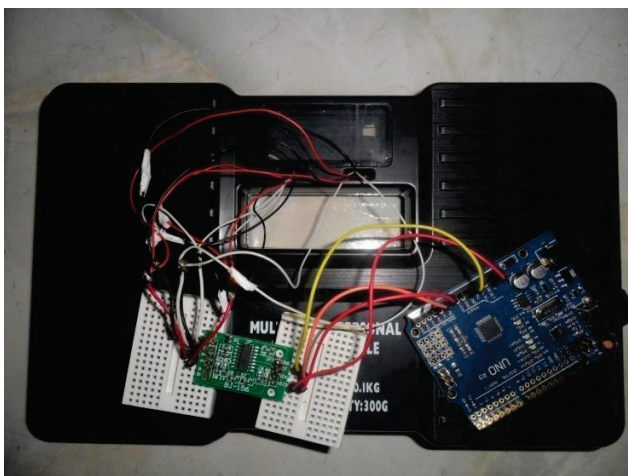
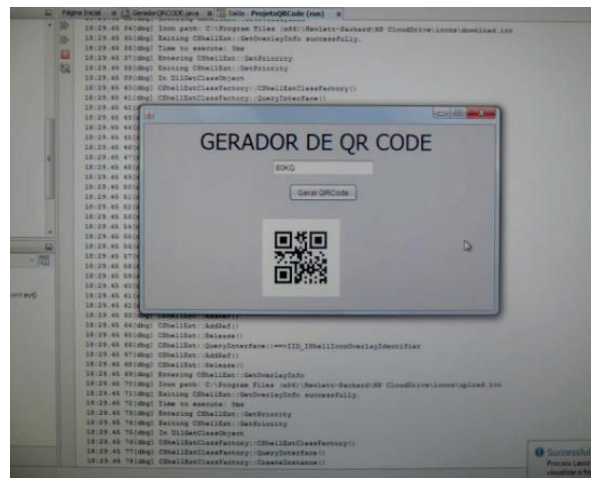


Figura 7 - Esquema de funcionamento do sistema

A figura 8(a) mostra a conexão entre Arduino, balança e conversor 24Bits, e a Figura 8(b) mostra o QR Code gerado pela aplicação no computador.



(a)



(b)

Figura 8- Resultados .(a) Conexão entre Arduino, conversor e balança.(b) QRCode Gerado pela aplicação em Java.

Após o QRCode ser gerado a aplicação móvel precisa ser instalada e ler essa informação com o leitor de QRCode conforme é mostrado nas Figuras 9a e 9b

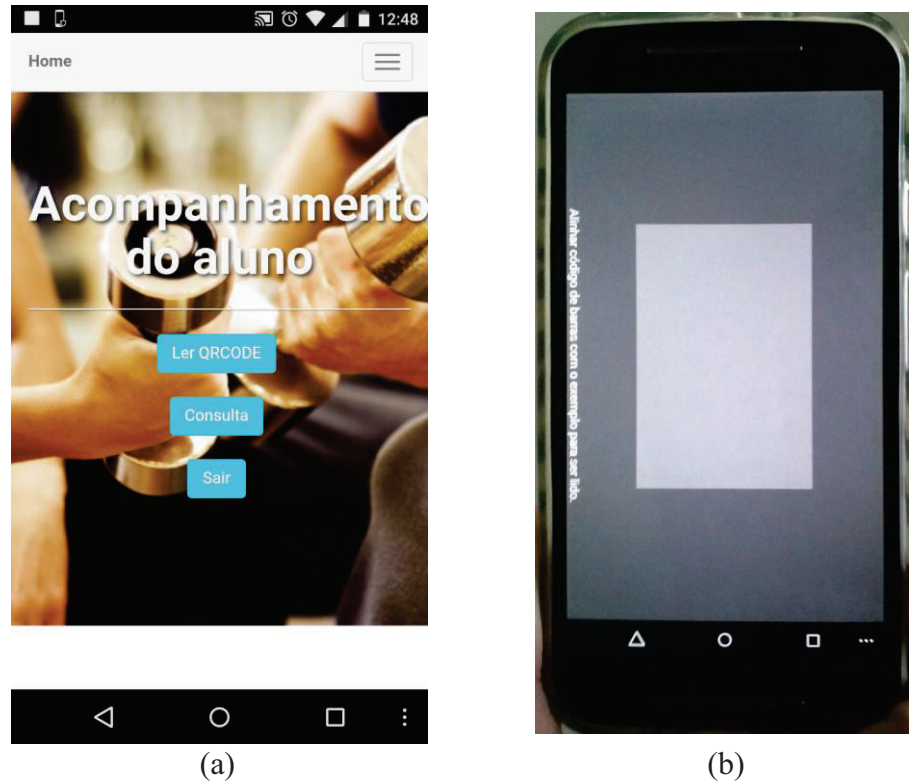


Figura 9- Aplicação móvel iniciada. (a) Tela principal. (b) Leitor de QR Code em execução. Leitor pronto, então é gerado o gráfico para acompanhamento, como é mostrado na Figura 10.



Figura 10- Aplicação móvel mostrando a tela com o gráfico em linha.

6. CONSIDERAÇÕES FINAIS

Conforme o objetivo do projeto, que foi desenvolver um aplicativo capaz de monitorar e acompanhar a informação do peso de uma pessoa através de um sistema integrado, foi possível observar que tal objetivo pode ser alcançado com êxito. Tanto a balança, como o microcontrolador, computador e smartphone comunicaram entre si cooperando para o funcionamento correto do sistema. Dentre as dificuldades encontradas, a principal delas foi de realizar a leitura da informação do peso pela célula de carga e o arduino, já que não foi encontrado na literatura nenhum tipo de material que descrevesse como se dá esse procedimento. Também houve dificuldades em pesquisar as bibliotecas que foram utilizadas, já que os requisitos do projeto exigiram que fossem utilizadas diversas tecnologias bastante diferentes uma das outras de forma integrada.

Como considerações finais do trabalho do trabalho, é esperado que o sistema integrado para acompanhamento de peso em academias, forneça um acompanhamento mais intuitivo da pesagem dos usuários em academias e com isso cada usuário tenha um melhor desempenho. Sendo o peso um dos principais fatores para a insatisfação corporal atualmente, espera-se também que os usuários vejam sua evolução facilmente e fiquem satisfeitos com seu desempenho.

Com o aplicativo móvel no smartphone de cada usuário das academias, o aplicativo pode aumentar a motivação de todos e a realização pessoal com seu corpo.

AGRADECIMENTOS

Os autores gostariam de agradecer ao professor André Márcio de Lima Curvello, Engenheiro da Computação, por ter contribuído neste projeto com seu vasto conhecimento em sistemas embarcados e eletrônica em geral, auxiliando a resolver os problemas na leitura dos valores de peso das células de carga com o arduino.

REFERÊNCIAS

ABRAHAM, Nikhil. **Coding For Dummies.1. ed.**New Jersey: **John Wiley & Sons, 2015. 288p.**

GHATOL Rohit , PATEL Yogesh. **Beginning Phonegap: Mobile Web Framework For Javascript and Html5.** New York: Apress, 2012. 344p



- KREIBICH, Jay A. **Using SQLite**. 1. ed. United States of America: O'Reilly, 2010. 530p.
- MICROBERTS, Michael. **Arduino Básico**. 1. ed. São Paulo: Novatec Editora, 2013. 423p.
- MONK, Simon. **Projetos com Arduino e Android: Use seu smartphone ou tablet para controlar o Arduino**. 1 ed. Porto Alegre: Grupo A Editora, 2013. 212p
- MONK, Simon. **Programação com Arduino II: Passos Avançados e Sketches**. 1 ed. Porto Alegre: Grupo A Editora, 2014. 260p.
- ROSA, Frankys Deylon, MESQUITA, Maycon J. Costa, ALMEIDA, Wil Ribamar Mendes, ARAUJO, Patricio Moreira de Filho. Controle e Supervisão Residencial Utilizando a Plataforma Arduino. **Acta Brazilian Science** p. 68-73, Abr. 2013 disponível em <<http://www.actabrazilianscience.com.br/arquivos/edicao1/vol1.pdf#page=68>>. Acesso em: 15 jun. 2015.
- ROSARIO, João Mauricio. **Automação Industrial**. 1 ed. São Paulo: Barauna Editora, 2009. 515p.
- SCHILDT, Hebert. **Java para Iniciantes**. 6 ed. Porto Alegre: Grupo A Editora, 2015. 704p
- TESSMER, Chiara Scaglioni. SILVA Marcelo Cozzensa. PINHO, Michele Nunes. GAZALLE, Fernando Kratz. FASSA, Anaclaudia Gastal. **Insatisfação corporal em frequentadores de academia**, Editora Universa, Pelotas, p 7-12, 2006. Disponível em <<http://portalrevistas.ucb.br/index.php/RBCM/article/viewFile/672/677..>> acesso em 13 jun 2015.
- SAMPAIO, Cleuton; RODRIGUES Francisco. **Mobile Game Jam: Criação de Jogos Móveis Multiplataforma**. 1 ed. Rio de Janeiro: Brasport, 2012. 284p.
- WINTER, Mick. **Scan-me- Everybody's guide to the magical world of QR Codes**. 1. ed. United States Of America: Westsong, 2011. 144p.

