



Ministério da Educação – Brasil
Universidade Federal dos Vales do Jequitinhonha e Mucuri – UFVJM
Minas Gerais – Brasil
Revista Vozes dos Vales: Publicações Acadêmicas
Reg.: 120.2.095 – 2011 – UFVJM
ISSN: 2238-6424
QUALIS/CAPES – LATINDEX
Nº. 11 – Ano VI – 05/2017
<http://www.ufvjm.edu.br/vozes>

Utilização da Plataforma Android para Desenvolvimento do Aplicativo Mate Código de Barras para o Ensino de Física e Matemática

Prof. Dr. Mauro Lúcio Franco
Doutor em Química pela Universidade Federal de Minas Gerais – UFMG
Docente da Faculdade de Ciências Sociais Aplicadas e Exatas – FACSAB da
Universidade Federal dos Vales do Jequitinhonha e Mucuri UFVJM - MG/Brasil
<http://lattes.cnpq.br/5529582752535382>
E-mail: ml.franco@ufvjm.edu.br

Prof. Dr. Wederson Marcos Alves
Doutor em Engenharia Agrícola pela Universidade Federal de Viçosa – UFV
Docente da Faculdade de Ciências Sociais Aplicadas e Exatas – FACSAB, da
Universidade Federal dos Vales do Jequitinhonha e Mucuri – UFVJM - MG/Brasil
<http://lattes.cnpq.br/8599448364867450>
E-mail: wederson.alves@ufvjm.edu.br

Cássio Gonçalves Sena
Mestrando em Tecnologia Ambiente e Sociedade -TAS - UFVJM/MG
Docente da Faculdade Presidente Antônio Carlos de Teófilo Otoni/MG - Brasil
<http://lattes.cnpq.br/3314411776560208>
E-mail: cassiosena@gmail.com

Resumo: O presente trabalho foi elaborado tendo em vista o desenvolvimento de um aplicativo para dispositivos móveis com Android para calcular o dígito verificador dos códigos de barras buscando favorecer o ensino de física e matemática. Para um produto final que atendesse os requisitos foi seguido as metodologias previstas na engenharia de software. Foi utilizado para criação a interface de desenvolvimento (IDE) Android Studio junto com Software Development Kit (SDK) e a linguagem java,

pois esta plataforma favorece a criação de aplicativos de forma produtiva, padronizada e dinâmica. Com a finalização do aplicativo pode-se observar uma boa disponibilização do conteúdo didático abordado como também uma interação com ambiente externo.

Palavras-chave: Física, Matemática, Android, Java, Aplicativo Educacional, Códigos de Barras

Introdução

O ensino de física e matemática tem-se tornado motivo de preocupação para os educadores nas escolas públicas brasileiras. Conforme Franco et al. (2016), a busca por novas metodologias e recursos pedagógicos que possam ajudar a compreender os fenômenos estudados e o dia a dia do aluno é de fundamental importância para o processo de ensino-aprendizagem. É preciso repensar a forma tradicional de ensino utilizados nas aulas de física e matemática, mudança já sugerida no Parâmetro Curricular Nacional (Brasil 1997, p15) onde diz que “há urgência em reformular objetivos, rever conteúdos e buscar metodologias compatíveis com a formação que hoje a sociedade reclama”.

A popularização dos smartphones abriu novos caminhos para o ensino na sala de aula, por serem extremamente flexíveis, convenientes e projetados para comunicação, devido sua portabilidade com tamanhos reduzidos e a diversidade de aplicativos com múltiplas funções integradas que os tornam celulares inteligentes, um acessório indispensável para a nova geração (M. Opera, Cristina Miron, 2014).

A aprendizagem com dispositivos móveis (m-learning) introduz a ideia de aprender praticamente de qualquer lugar (Z.J. Lu, 2014). Conforme Hui et al. (2005) o m-learning têm o potencial de reduzir a distância transacional entre o aluno e instrutor e permitir experiências de aprendizagem mais colaborativas, ricamente contextualizadas e continuamente acessível.

O presente trabalho é parte do projeto de pesquisa apresentado ao Programa de Pós-Graduação *Stricto Sensu* Mestrado em Tecnologia, Ambiente e Sociedade (PPGTAS) da Universidade Federal dos Vales do Jequitinhonha e Mucuri. O aplicativo aqui apresentado é denominado de Mate Código Barras para Android e tem o objetivo de contribuir com ensino-aprendizado nas disciplinas de

física e matemática. Para tal finalidade foi utilizado a interface de desenvolvimento (IDE) Android Studio que incorpora em suas bibliotecas chamadas de Software Development Kit (SDK) que compõem uma plataforma de desenvolvimento Android em conjunto com a linguagem Java Development Kit (JDK), que é uma plataforma gratuita, moderna e ágil para desenvolvimento de aplicativos, que foi criado e é mantida pela empresa Google Inc.

A linguagem Java permite desenvolver aplicativos que são executados em uma variedade de dispositivos, sem nenhum código específico para a plataforma. Os programadores Java experientes podem se aprofundar rapidamente no desenvolvimento com Android, usando as APIs (interfaces de programação de aplicativo) Android e outras disponíveis de terceiros. (Deitel, 2013, p. 5).

A interface sensível com usuário foi desenvolvida utilizando as orientações definidas nas especificações do Material Design, que é uma linguagem de design visual desenvolvida pela [Google](#) (Google Developers, 2016), que são os recursos gráficos e layouts como grids, animações e transições, preenchimentos, e efeitos de profundidade como luzes e sombras estes componentes e funcionalidades estão disponíveis no Android 5.0 e em posteriores.

Processo de desenvolvimento

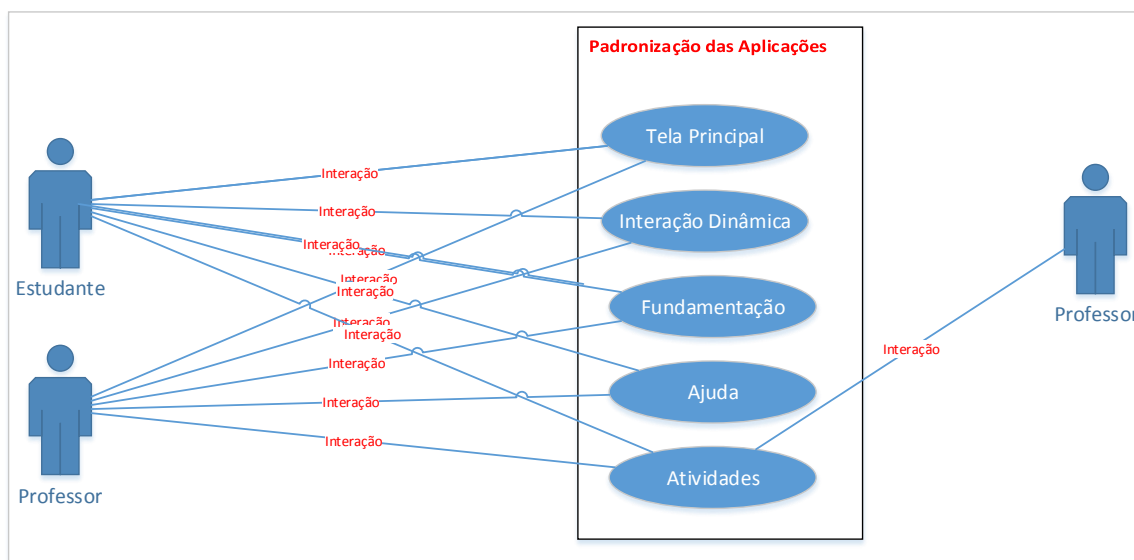
O processo de desenvolvimento de aplicativos voltados para educação fundamenta-se em conceitos educacionais e computacionais com o intuito da construção do conhecimento de um produto baseado em uma proposta pedagógica predefinida.

O aplicativo Mate Código de Barras foi desenvolvido baseado na proposta de Benitti et al. (2005) que diz que o processo de desenvolvimento de aplicativos educacionais baseia-se em quatro etapas de forma iterativa e incremental: concepção, elaboração, concepção, elaboração, finalização e viabilização. A concepção é fase que define as etapas do aplicativo educacional onde o ponto de partida são os objetivos de aprendizagem que foram estruturados em requisitos computacionais, momento que os profissionais da área de computação e da educação são determinantes. O conteúdo didático do aplicativo foram definidos por professores de baseado na concepção de software levando em conta as etapas de implementação, avaliação e validação. Foi criado o protótipo funcional do aplicativo direcionado pela etapa de concepção.

A finalização não se resume na etapa final, mas é fase o qual o aplicativo foi testado pelos utilizadores finais, que neste caso, validados por alunos do ensino médio. Nesta etapa também foi realizado pelos autores ajustes os quais observados no processo de validação. É importante ressaltar, que o usuário terá todo o conteúdo organizado (manual) disponibilizado no próprio aplicativo podendo ser acessado a qualquer momento pelos professores do ensino médio como recurso didático para aulas de física e matemática. A visualização é o processo que os profissionais da educação, neste caso os professores, recebem o treinamento de interação, como visto no Quadro 1, e como as atividades didáticas disponibilizadas pelo aplicativo poderão ser utilizadas nas aulas de física e matemática.

Ainda na etapa de visualização, a mesma contempla o acompanhamento da utilização do aplicativo, tendo em vista que alguns erros ainda poderão correr, falhas essas que não foram encontrados na fases anteriores. Outro aspecto importante neste estágio e presente em todos aplicativos desenvolvidos é a manutenção e suporte, tanto na área didática quanto na computacional.

Quadro 1 – Etapas da finalização

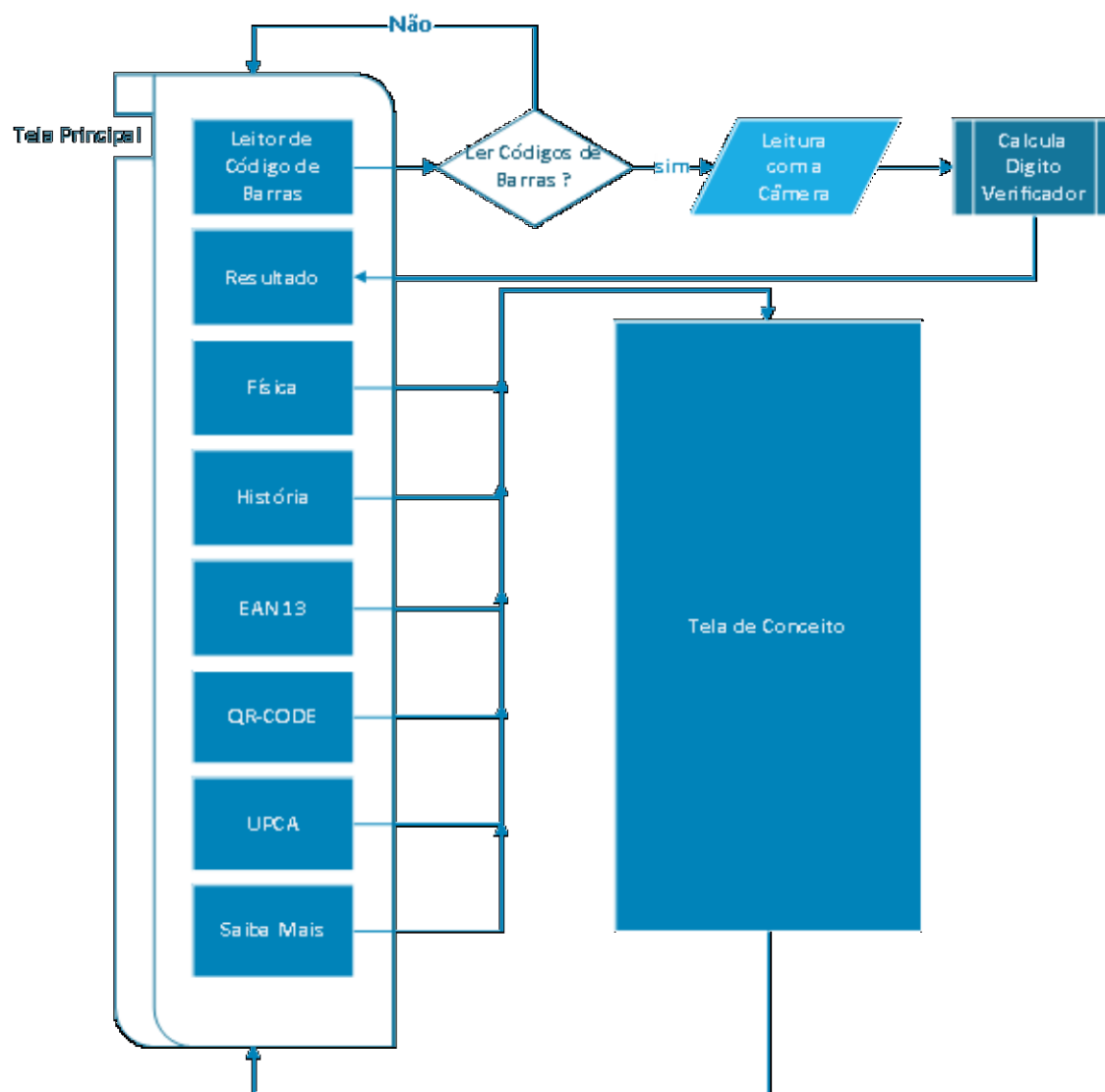


Visão Geral

O cenário de funcionamento do aplicativo será demonstrado no fluxograma no Quadro 2 que contém o fluxo de ações esperada pela aplicação.

No diagrama foi exposto as representações dos recursos suportadas pela aplicação e seus acessos e interação. Após a inicialização a aplicação exibe a tela principal com as opções de entrada.

Quadro 2 – Aplicativo representado através do fluxograma



O fluxograma acima não exibe todas as telas, pois em algumas interações telas auxiliares são exibidas através de recurso da linguagem Java para Android denominadas de intents, que o Guia de Programação (2016) conceitua como “um

objeto de mensagem que pode ser utilizado para solicitar uma ação de outro componente de aplicativo”.

Metodologia

O aplicativo Mate Código de Barras foi desenvolvido utilizando fases descritas anteriores que nos termos da engenharia de software segue os requisitos do método incremental, que segundo Sommerville (2003) define que todo o escopo do aplicativo é definido na primeira etapa juntamente com os professores de física e matemática e os desenvolvedores do aplicativo educacional, e o método iterativo que baseado no conceito de Sommerville (2003) diz que as etapas do projeto são repetidas o qual no presente projeto foram utilizadas as três últimas fases, ou seja várias partes do sistema foram desenvolvidas de forma separadas e integradas quando foram concluídas.

Processo de Desenvolvimento do Aplicativo

As telas de interface com usuário foram construídas baseadas em Material Designer Android ViewPager Screen Slides, onde as trocas das telas são feitas deslizando o dedo sobre a tela do dispositivo, baseado nos objetivos da Interação Humano-Computador (IHC), que define a experiência do usuário (EU), do inglês user experience (UX) onde determina como o usuário deverá interagir com o aplicativo até mesmo o design emocional, como por exemplo as cores de botões, texto e telas.

Baseado na definição do Guia de Programação (2016) os “ViewPager Screen Slides a são transições entre uma tela e outra são comuns com a interface do usuário (UI) com assistentes de configuração ou apresentações de slides”.

Para codificar dos layouts do aplicativo que define a estrutura visual para a interface do usuário, foi utilizada a linguagem XML (eXtensible Markup Language) em português Linguagem Extensível de Marcação Genérica. O XML é o vocabulário do que corresponde às classes e subclasses de View, como as de widgets e layouts.

No Quadro 3 mostra um exemplo do arquivo `layout_fisica.xml` do aplicativo que utiliza um [Layout](#) Linear Vertical para conter um [Exibidor](#) de Texto, Visualizador de Imagem e Visualizador Web que exibe o texto como páginas Web em HTML — com uma série de elementos aninhados.

Cada arquivo de layout deve conter exatamente um elemento raiz, que deve ser um objeto View ou ViewGroup. Com o elemento raiz definido, é possível adicionar objetos ou widgets de layout extras como elementos filho para construir gradualmente uma hierarquia de View que define o layout.

Quadro 3 - Exemplo layout declarado em xml

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="1">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/imageView"
        android:layout_gravity="center_horizontal" />
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="10dp">
        <WebView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/webView_fisica"
            android:text="@string/desc_fisica"
            android:layout_weight="0.78"
            android:textAlignment="gravity" />
    </ScrollView>
</LinearLayout>
```

Para escrever o código previsto em cada ação com usuário foi utilizada a linguagem de programação Java Development Kit (JDK), que é totalmente personalizada para plataforma Android que é disponibilizada no SDK são as iniciais para Software Development Kit, onde contém diversas ferramentas que são utilizadas pelo IDE (Interface de Desenvolvimento) Android Studio quanto pelo desenvolvedores. No Quadro 4 existe parte dos códigos em java do arquivo `MainActivity.java` responsável pela execução principal do aplicativo.

Quadro 4 – Codigos do MainActivity escrito em Java.

```
public class MainActivity extends ActionBarActivity {
    @InjectView(R.id.toolbar)
    Toolbar toolbar;
    @InjectView(R.id.tabs)
    PagerSlidingTabStrip tabs;
    @InjectView(R.id.pager)
    ViewPager pager;
    private MyPagerAdapter adapter;
    private Drawable oldBackground = null;
    private int currentColor;
    private SystemBarTintManager mTintManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ButterKnife.inject(this);
        setSupportActionBar(toolbar);
        mTintManager = new SystemBarTintManager(this);
        mTintManager.setStatusBarTintEnabled(true);
        adapter = new MyPagerAdapter(getSupportFragmentManager());
        pager.setAdapter(adapter);
        tabs.setViewPager(pager);
    }
}
```

Telas e Layouts do Aplicativo Mate Códigos de Barras

Como já mencionado anteriormente os desenho e cores das telas (layouts) foram padronizados buscando uma melhor eficiência e usabilidade do aplicativo. Abaixo serão mostrados todos os passos para utilização do aplicativo.

Passo 1

Na tela principal o aluno ou professor quando iniciar o aplicativo poderá visualizar um menu superior na tela. Na primeira opção é a de “INÍCIO”, onde possui os botões para orientações e acesso direto as opções do Menu como mostrado no Quadro 5.

Quadro 5 – Tela Principal do Aplicativo



Passo 2

O primeiro passo da utilização como recurso didático para aulas de física e matemática é realizar a leitura do código de barras barra tocando no ícone de leitura localizado no Menu Início como exibido no Quadro 6.

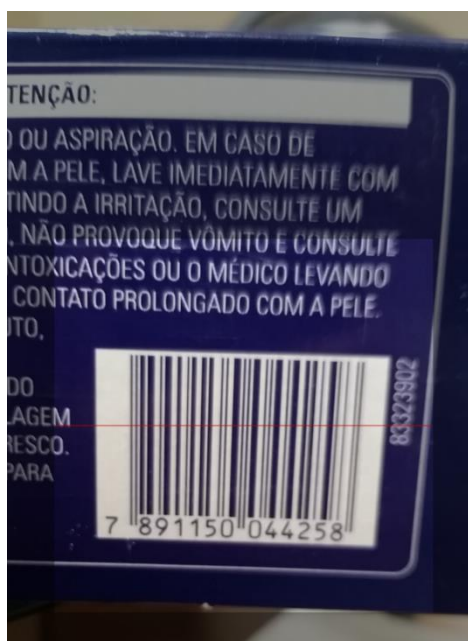
Quadro 6 – Inicialização da leitura do Código de Barras



Passo 3

Neste passo o usuário apontará a câmera do smartphone sobre o código de barra do tipo EAN-13 que deseja fazer a leitura, deverá utilizar a linha guia vermelha para melhor precisão de leitura, como mostrado no Quadro 7

Quadro 7 – Realizando a leitura do código de barras através da câmera do dispositivo



Passo 4

Após a leitura o aplicativo será automaticamente direcionado para a tela “RESULTADO” contendo todos os cálculos matemáticos que foram utilizados para encontrar o dígito verificador do código de barras, como mostrado no Quadro 8.

Quadro 8 – Tela de resultados exibindo como se calcula o dígito verificador do código de barras.

Mate Código de Barras

INÍCIO RESULTADO FÍSICA HISTÓRIA EAN-

Calculando o Dígito Verificador

7	8	9	1	1	5	0	0	4	4	2	5
x	x	x	x	x	x	x	x	x	x	x	x
1	3	1	3	1	3	1	3	1	3	1	3

O sistema EAN-13, se utiliza de um vetor fixo, que chamaremos, vetor de peso que é:
 $w = (1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1)$.

Multiplique cada dígito do código pelo dígito colocado imediatamente abaixo dele, como na tabela acima:

$$(1 \times 7) + (3 \times 8) + (1 \times 9) + (3 \times 1) + (1 \times 1) + (3 \times 5) + (1 \times 0) + (3 \times 0) + (1 \times 4) + (3 \times 4) + (1 \times 2) + (3 \times 5) = 92 + X$$

Agora, o dígito de verificação X é escolhido de forma tal que a soma acima seja múltiplo de 10. Consequentemente, deve-se tomar $X = 8$. Verifique no código de barra da embalagem

Os resultados também poderão ser acessados novamente através do ícone presente no menu “INÍCIO”.

Passo 5

Buscando alinhar a teoria à prática foi disponibilizado no aplicativo cinco telas como instrumento de ensino-aprendizagem contextualizado, com intuito da continuidade da aprendizagem além da sala de aula aos alunos. As telas foram dispostas da seguinte forma: uma de “FÍSICA” que apresenta a fundamentação teórica aplicada nos leitores óticos utilizados em estabelecimento de vendas, a de “HISTÓRIA” dos códigos de barras, apresentando o contexto histórico e seus inventores, outra “EAN-13” apresentado os conceitos deste tipo de código de barra, que é o código utilizado nos produtos brasileiros, seguida pela tela “QR-CODE” que é um código de barras mais moderno capaz de armazenar pequenos textos ao invés de números, na próxima tela exibe todos detalhes do código de barras “UPC-A”, que é uma variação do código EAN-13 e por último a tela “SAIBA MAIS” onde são apresentados links para vídeos e sites que possibilita o aluno quanto o professor de expandir o conhecimento na área. As telas são mostradas no Quadro 9 abaixo.

Quadro 9 – Telas com as fundamentações teóricas auxiliares



EAN-13 (sigla de **European Article Number**) é um código de barras no padrão EAN definido pela GS1, adaptado em mais de cem organizações, para a identificação dos itens, principalmente nos pontos de venda a retalho (português europeu) ou varejo (português brasileiro) de todo o mundo, com exceção América do Norte (EUA e Canadá) onde se usa o código de barras UPC. No EAN-13 o símbolo codifica treze números que estão divididos em quatro partes; dos treze dígitos, doze são dos dados referentes ao produto e um é o dígito verificador (Codificação EAN-13, 2007).

Sobre os Códigos de Barras:

[Como adquirir um código para seu produto.](#)

[Crie o seu código de barras EAN13.](#)

[Crie o seu código QR gratuitamente.](#)

Vídeos

[Reflexão Total da Luz](#)

[Isto é Matemática - O Código de Barras](#)

[Leitor de Código de Barras](#)

Conclusão

Através dos estudos realizados é possível verificar um crescimento expressivo de dispositivos móveis que possuem o sistema operacional Android, principalmente em escolas, com excelentes recursos que podem ser explorados como recursos didáticos para favorecer o ensino-aprendizagem. Tal popularização se torna um motivador para desenvolver aplicativos, principalmente que contribuam na educação em vários conteúdos.

Com ajuda da plataforma de desenvolvimento Android e a linguagem java foi possível construir o aplicativo com todas as funcionalidades de uma forma lúdica e dinâmica. Seguindo o mesmo padrão adotado neste artigo novos aplicativos serão desenvolvidos uma vez que a padronização de layouts nos permite incluir novos conteúdos com temas diversificado e atividades que favoreçam a interatividade. Uma importante funcionalidade aproveita no smartphone para a criação do aplicativo a sua câmera que favorece a leitura do código de barras a partir da imagem capturada.

O próximo passo do aplicativo é aproveitado outros recursos disponíveis no dispositivo quanto também o acesso a internet para que os conteúdos possam

ser disponibilizados de forma dinâmicos e favorecer a coleta de dados dos alunos para um feedback para os professores onde os mesmo poderão fazer uma avaliação de forma instantânea.

Agradecimentos:

1 - Programa de Popularização e Difusão da Ciência no Vale do Mucuri - Apoio: PROEXT – MEC/SESu.

2 - Programa Institucional de Bolsa de Iniciação à Docência - Apoio: PIBID/CAPES

3 - Programa de Educação Tutorial - PET/Novas Tecnologias Voltadas para o Ensino - Apoio: MEC/SESu

Referências

FRANCO, Mauro Lúcio; et al. *Utilização da Linguagem Action Script 3.0 no Desenvolvimento de uma Plataforma Virtual de Ensino de Física*. Vozes dos Vales, 2016.

HUI, D.; Bo, H.; Yiyi, Z.; Zhi, Y.; Meng, M.; Shangkai, G. *Construction of a knowledge center for medical image processing*. In Proceedings of the IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, China, 2005.

INTENTS E FILTROS DE INTENTS, GOOGLE DEVELOPERS. Disponível em: <<https://developer.android.com/guide/components/intents-filters.html?hl=pt-br>> Acesso em: 11 março. 2017.

MATERIAL DESIGN PARA ANDROID, GOOGLE DEVELOPERS. Disponível em: <<https://developer.android.com/design/material/index.html?hl=pt-br>>. Acessado em: 15 março. 2017

OLIVEIRA, S. R. B.; VASCONCELOS, A. M. L. de; ROUILLER, A. C. *Uma proposta de um ambiente de implementação de processo de software*. Disponível em: <www.dcc.ufla.br/info_comp/artigos/v4.1/art09.pdf>. Acesso em: 10 março. 2017.

OPREA, M.; MIRON, C. *Mobile Phones in the Modern Teaching of Physics*. Romanian Reports in Physics, 2014.

PAPER PRESENTED AT THE ANNUAL SCIENTIFIC SESSION OF FACULTY OF PHYSICS. University of Bucharest, Bucharest-Magurele, Romania, 2013.

BENITTI, Fabiane Barreto Vavassori *et al.* *Processo de Desenvolvimento de Software Educacional: proposta e experimentação*. CINTED-UFRGS. 2005

SOMMERVILLE, I. *Engenharia de software*. São Paulo: Ed Addison-Wesley, 2003

LU, Zhongyu Joan. *Learning with Mobile Technologies, Handheld Devices, and Smart Phones: Innovative Methods*. IGI Global, Hershey, PA, USA, 2012.

Processo de Avaliação por Pares: (*Blind Review* - Análise do Texto Anônimo)

Publicado na Revista Vozes dos Vales - www.ufvjm.edu.br/vozes em: 05/2017

Revista Científica Vozes dos Vales - UFVJM - Minas Gerais - Brasil

www.ufvjm.edu.br/vozes

www.facebook.com/revistavozesdosvales

UFVJM: 120.2.095-2011 - QUALIS/CAPES - LATINDEX: 22524 - ISSN: 2238-6424
Periódico Científico Eletrônico divulgado nos programas brasileiros *Stricto Sensu*
(Mestrados e Doutorados) e em universidades de 38 países,
em diversas áreas do conhecimento.