

UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS
CURSO BACHARELADO EM SISTEMAS DE INFORMAÇÃO

**APLICATIVO MÓVEL PARA AUXILIAR PROFESSORES NO USO DE OBJETOS
VIRTUAIS DE APRENDIZAGEM**

JAZAN NOGUEIRA DELFINO

PICOS – PIAUÍ

2016

JAZAN NOGUEIRA DELFINO

**APLICATIVO MÓVEL PARA AUXILIAR PROFESSORES NO USO DE OBJETOS
VIRTUAIS DE APRENDIZAGEM**

Trabalho de conclusão de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação do Campus Senador Helvídio Nunes de Barros da Universidade Federal do Piauí como parte dos requisitos para obtenção do Grau de Bacharel em Sistemas de Informação.

Orientador: Professor Esp. Leonardo Pereira de Sousa.

PICOS – PIAUI

2016

FICHA CATALOGRÁFICA

**Serviço de Processamento Técnico da Universidade Federal do Piauí
Biblioteca José Albano de Macêdo**

D349a Delfino, Jazan Nogueira.

Aplicativo móvel para auxiliar professores no uso de objetos virtuais de aprendizagem. – 2016.

CD-ROM : il.; 4 ¾ pol. (55 f.)

Monografia(Bacharelado em Sistemas de Informação) –
Universidade Federal do Piauí, Picos, 2016.

Orientador(A): Profº. Esp. Leonardo Pereira de Sousa.

1. Dispositivos Móveis. 2. *Android*. 3. Aplicativo Pedagógico. I. Título.

CDD 005.1

APLICATIVO MÓVEL PARA AUXILIAR PROFESSORES NO USO DE OBJETOS
VIRTUAIS DE APRENDIZAGEM

JAZAN NOGUEIRA DELFINO

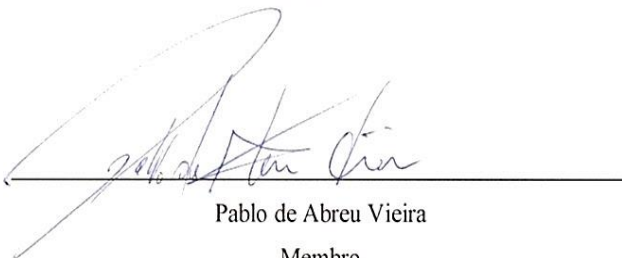
Monografia APROVADA como exigência parcial para obtenção do grau de
Bacharel em Sistemas de Informação.

Data de Aprovação

Picos – PI, 22 de FEVEREIRO de 2016



Prof. Esp. Leonardo Pereira de Sousa
Orientador



Pablo de Abreu Vieira
Membro



Márcio Alves de Macêdo
Membro

Dedico este trabalho a minha família, em especial meus pais e minhas irmãs, pelo apoio e incentivo e por sempre acreditarem nos meus sonhos.

AGRADECIMENTOS

Agradeço primeiro a Deus por ter me proporcionado saúde e coragem para que eu pudesse chegar a mais esta etapa. Aos meus professores que fizeram parte dessa longa jornada, cada um ensinando e cativando de maneiras diferentes, mas o intuito de não apenas passar o conteúdo, mas para nos preparar para a vida; Aos demais coordenadores e funcionários da UFPI, pois sem ela meu sonho não teria sido concretizado. A minha família, pelo apoio, dedicação e confiança desde o início. Aos amigos do CTI que sempre apoiaram que de certa forma fizeram parte dessa realização.

E agradeço especialmente a meu orientador Prof. Leonardo Pereira de Sousa que foi de suma importância para elaboração dessa monografia, onde se mostrou disponível e compreensível para a conclusão desse trabalho. Bem como os avaliadores que fazem parte dessa banca.

A todos vocês meu muito obrigado.

“Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se lhe faltasse uma gota”.

Madre Teresa de Calcutá

RESUMO

Um grande problema que os professores vêm enfrentando atualmente em sala de aula, é o uso de aparelho celular pelos alunos, que na maioria das vezes não são utilizados para fins educativos, mas para distração, acarretando dessa forma, sérios problemas, tais como: queda no rendimento escolar, trabalhos não realizados, falta de concentração, dentre outros aspectos negativos que prejudicam a educação. Analisando essa situação e compreendendo as dificuldades encontradas pelos professores em possuir e empregar a tecnologia em prol do conhecimento, foi desenvolvido um aplicativo, para minimizar o problema e tornar essa ferramenta como uma aliada na prática educativa, visto que nos dias atuais não se pode fugir dessa tecnologia tão utilizada e atrativa aos alunos, uma vez que as Tecnologias de Informação e Comunicação, as conhecidas TICs, estão ocupando um espaço na prática educativa moderna. O objetivo deste trabalho é o desenvolvimento de um aplicativo para dispositivos móveis com suporte ao sistema operacional *Android*, que auxilie o professor no uso de um objeto virtual de aprendizagem, que amplie a interação aluno-professor no processo de transmissão dos conteúdos, facilitando a metodologia de ensino-aprendizagem, tornando-a mais eficaz com o uso dessa ferramenta pedagógica.

Palavras-chave: Dispositivos móveis, *Android*, API, E-mail, Ensino, Apoio Pedagógico

ABSTRACT

A big problem that teachers are facing today in the classroom is the use of mobile device by students, which most of the times are not used for educational purposes, but for distraction, causing this way, serious problems, such as drop in school performance, works not carried out, lack of concentration, among other negative aspects that affect the education. Analyzing this situation and understanding the difficulties encountered by the teachers in the possession and employ technology in favor of knowledge, was developed an application, to minimize the problem and make this tool as an ally in the educational practice, since today we cannot evade this technology as used and attractive to the students, once that information and communication technologies, the known ICT, are tanking a space in the modern educational practice. The objective of this work is the creation of an application for mobile devices with Android operating system support, which helps the teacher in the use of a virtual learning object, which extend the interaction student-teacher in the process of transmission of content, facilitating the teaching-learning methodology, making it more effective with the use of this, pedagogical tool.

Keywords: Mobile Devices, Android, API, E-mail, ICT.

LISTA DE ILUSTRAÇÕES

Figura 1: Arquitetura <i>Android</i>	23
Figura 2: Versões <i>Android</i>	26
Figura 3: Distribuição do <i>Android</i> nos dispositivos.....	27
Figura 4a- Servidor SMTP.....	29
Figura 4b: <i>OAuth</i>	30
Figura 5: Funcionamento do GCM.....	32
Figura 6: Representação dos Diagramas UML.....	35
Figura 7: Diagrama de Caso de uso.....	39
Figura 8: Diagrama de Classe.....	41
Figura 9a: Tela de escolha de conta	43
Figura 9b: Tela de Boas-Vindas	43
Figura 10: Tela de Cadastro de Disciplina	43
Figura 11: Disciplinas Cadastradas	44
Figura 12: Disciplina Selecionada	44
Figura 13: Tela de Cadastro Aluno	45
Figura 14: Lista de Alunos.....	46
Figura 15: Tela envio de arquivos.....	46
Figura 16: Listagem de mensagens por assunto.....	47
Figura 17: Notificação de <i>E-mail</i>	48

LISTA DE QUADROS

Quadro 1: Requisitos Funcionais.....	37
Quadro 2: Requisitos Não-Funcionais.....	38

LISTA DE ABREVIATURAS E SIGLAS

ADT - *Android Development Tool*
API - *Application Programming Interface*
DEX - *Dalvik Executbale*
DVM - *Dalvik Virtual Machine*
GCM - *Google Cloud Messaging*
GPS – *Global Positionig System*
IDE - *Integrated Development Environment*
IMAP – *Internet Message Access Protocol*
JVM - *Java Virtual Machine*
OHA - *Open Handset Alliance*
SDK - *Software De Development Kit*
SQL - *Structured Query Language*
TIC's – *Tecnologias da informação e comunicação*
UML - *Unified Modeling Language*
URI- *Uniform Resource Identifiers*

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVO	15
1.2	ESTRUTURA DA MONOGRAFIA.....	15
2	REFERENCIAL TEÓRICO	17
2.1	SISTEMAS DE INFORMAÇÃO.....	17
2.1.1	<i>CLASSIFICAÇÃO DOS SISTEMAS</i>	<i>18</i>
2.2	TECNOLOGIAS DA INFORMAÇÃO MÓVEL	19
2.3	APRENDIZAGEM COM MOBILIDADE	20
3	AMBIENTE DE DESENVOLVIMENTO	22
3.1	SISTEMA OPERACIONAL <i>ANDROID</i>	22
3.1.1	<i>PLATAFORMA ANDROID.....</i>	<i>23</i>
3.1.2	<i>ARQUITETURA ANDROID</i>	<i>23</i>
3.1.3	<i>VERSÕES ANDROID</i>	<i>27</i>
3.2	<i>GMAIL API</i>	<i>28</i>
3.3	LINGUAGEM <i>JAVA</i>	31
3.4	<i>SQLITE</i>	<i>31</i>
3.5	<i>ANDROID STUDIO</i>	<i>32</i>
3.6	ENGENHARIA DE <i>SOFTWARE</i>	33
3.7	UML	35
4	MODELAGEM DO APLICATIVO	37
4.1	CONSTRUÇÃO DA APLICAÇÃO	37
4.2	REQUISITOS DE <i>SOFTWARE</i>	38
4.3	DIAGRAMA DE CASO DE USO.....	39
4.4	DIAGRAMA DE CLASSES.....	41
4.5	FUNCIONALIDADES DO SISTEMA	43
5	RESULTADOS E DISCUSSÕES	48
5.1	PROBLEMAS ENCONTRADOS	48
5.2	RESULTADOS ALCANÇADOS.....	49
6	CONCLUSÕES E TRABALHOS FUTUROS.....	51
7	REFERÊNCIAS BIBLIOGRÁFICAS	52

1 INTRODUÇÃO

Um grande problema enfrentado tanto pelas escolas públicas quanto privadas é a falta de interesse dos alunos pelo conteúdo ensinado em sala de aula. Isso se deve a alguns fatores, tais como: a metodologia de ensino aplicada, a falta de recursos tecnológicos dentre outros elementos que contribuem para a exclusão digital e até mesmo a evasão escolar. O vínculo entre aluno e professor se decompõe em razão disso, tornam-se mais horizontal e sem lugar determinado. Aceitar e tomar consciência dessas modificações extremamente profundas e das suas decorrências cria um desafio para o professor que não pode ser ignorado.

A necessidade de inserir a tecnologia dentro da sala de aula é desafiadora, devido alguns fatores complicadores para tal inserção, como a falta de equipamentos e de infraestrutura. Uma pesquisa realizada pelo Instituto Claro (2011) afirma que 41% dos professores têm dificuldade de trabalhar com essas tecnologias.

Desponta assim outro grande problema, a pouca familiaridade do professor com as novas tecnologias, pois este não tem acesso a nenhum curso de capacitação promovido pelas instituições às quais mantém vínculos empregatícios que objetive melhorar suas metodologias pedagógicas em sala de aula, dificultando assim a criação de novos ambientes de aprendizagem.

Os professores em exercício não foram preparados para usar tecnologias digitais em sala e os que ainda estão na graduação também não estão sendo preparados. A educação continuada não resolve uma falha de formação inicial. As faculdades de educação das universidades deveriam ser verdadeiros laboratórios de inovação pedagógica (BORGES, 2013).

Com os avanços tecnológicos, as pessoas sentem necessidades de obter novos hábitos, metodologias de ensino e aprendizagem diferentes. Portanto, uma das soluções encontradas para atender a esses anseios, foi a aplicação de novas tecnologias, como os ambientes virtuais de aprendizagem, de maneira que seja possível tornar a aula mais atrativas e interativas, promovendo uma inovação na forma de ensinar e, conseqüentemente, novas experiências para aquisição do conhecimento (HOFSTAETTER, 2009).

Observando o ensino médio, a última etapa da Educação Básica, onde a incidência de distração é maior por parte dos alunos, uma vez que estes tendem a não prestar atenção nas aulas devido ao uso excessivo de dispositivos móveis, seja trocando mensagens ou realizando ligações que de um modo geral atrapalha o rendimento do mesmo e dos demais colegas. Esse problema tende a se agravar com avanço da tecnologia, e o uso constante desses dispositivos

móveis entre os jovens nessa nova era digital, pois a tecnologia quando mal usada, pode causar danos irreparáveis principalmente no que concerne à educação.

Devido a esse problema e com a perspectiva de oferecer alternativas que visem melhorar o processo ensino-aprendizagem, foi implementado um aplicativo móvel com para o sistema operacional *Android*. O objetivo dessa aplicação é auxiliar o professor no uso de uma ferramenta tecnológica, permitindo assim, levar para sua sala de aula um ambiente virtual novo e interativo, possibilitando cadastrar suas disciplinas e enviar conteúdo aos alunos através de um sistema de *e-mail*. Permitindo ao professor melhorar sua metodologia de ensino e atender dessa forma, à necessidade do aluno.

1.1 OBJETIVO

O presente trabalho tem por objetivo desenvolver um aplicativo para dispositivos móveis com suporte ao sistema operacional *Android*, que auxilie o professor no uso de um objeto virtual de aprendizagem, ampliando a interação aluno-professor no processo de transmissão dos conteúdos. A aplicação que será desenvolvida fará o cadastro de disciplinas, permitindo ao professor cadastrar alunos e fazer o envio de arquivos por meio dessa ferramenta. Dessa forma será possível ao professor estabelecer uma maior interação com aluno, de modo que possa melhorar sua metodologia aplicada em sala de aula.

1.2 ESTRUTURA DA MONOGRAFIA

Após a introdução que relatou sobre as motivações e objetivos para o desenvolvimento do presente trabalho, serão apresentados os próximos capítulos que estão organizados da seguinte forma:

- Capítulo 2 – Apresenta de uma maneira geral o que é um sistema de informação, seus benefícios dentro das organizações, bem como sua classificação de acordo com o papel que desempenha no cotidiano de uma empresa. Nesse mesmo capítulo é feita uma abordagem sobre o avanço das tecnologias móveis no âmbito social e educacional.
- Capítulo 3 – Neste capítulo são apresentadas as principais tecnologias utilizadas para desenvolvimento de aplicações *Android*.
- Capítulo 4 – Neste capítulo é feita uma descrição das técnicas utilizadas para desenvolvimento da aplicação *Android*. Bem como as funcionalidades do sistema e alguns resultados obtidos durante a etapa de desenvolvimento.

- Capítulo 5 – Apresenta os resultados obtidos na etapa de construção do aplicativo.
- Capítulo 6 - Apresenta a conclusão do trabalho e indicações para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Neste capítulo é feita uma abordagem em relação ao uso das tecnologias móveis dentro do ambiente escolar e suas contribuições no processo educacional brasileiro.

2.1 SISTEMAS DE INFORMAÇÃO

O mundo vivencia a era da informação, demandando das organizações em geral uma gestão estratégica competente, que pode ser favorecida pelo emprego de mecanismos inteligentes proporcionados pela tecnologia e sistemas informação. A tecnologia proporciona mecanismos tecnológicos e computacionais para a produção de informações, e, cada vez mais os sistemas são aprimorados, gerando modificações nos processos, estrutura e estratégia de negócios.

Um sistema de informação pode ser descrito, de maneira técnica, como um grupamento de elementos interconectados que colhe, restaura, processa, registra e distribui informações para fundamentar a tomada de decisões, a coordenação e a gerência de uma organização. As modificações resultantes do desenvolvimento tecnológico nas áreas de informação e comunicação influenciaram significativamente a sociedade. Para seguir essas modificações, tanto os indivíduos quanto as organizações têm buscado maneiras mais aceleradas para se inserir nesse modelo atual de mercado.

As informações confiáveis e proporcionadas em tempo hábil à tomada de decisão são de fundamental importância para as empresas contemporâneas. Nesse sentido, fazer a utilização adequada de tecnologias da informação é fundamental para assegurar a confiabilidade e exatidão das informações. GUIMARÃES E ÉVORA (2004) afirma que, a Tecnologia da Informação é um aglomerado de técnicas e instrumentos, mecanizadas ou não, que se compromete a assegurar a qualidade e rigor das informações dentro da rede empresarial.

BORGES (2008) caracteriza a Tecnologia de Informação como todo e qualquer mecanismo que contenha a habilidade para tratar dados e informações, seja de maneira sistêmica ou esporádica, independente da forma como é aplicada. A procura pela resolução dos problemas direciona os gestores a juntar as partes que estruturam a organização para constituir um sistema que oferecerá condições para administrar o todo. Segundo os trabalhos de DA SILVA (2010), sistema é uma junção de partes interagentes e interdependentes que, em conjunto, constituem um todo com objetivo específico e executam determinada função. O

desenvolvimento de um sistema se produz pela união de diferentes componentes interdependentes que unidos tendem a alcançar um objetivo comum.

Os autores GREEF E FREITAS (2012) sustentam que o sistema é a acomodação dos componentes de um todo que, de modo coordenado, constituem a estrutura organizada, com a intenção de concretizar uma ou mais atividades ou, também, um conjunto de acontecimentos que recomeçam em um ciclo na efetivação de tarefas predefinidas. A cobrança de um mercado cada vez mais competitivo, dinâmico e especialmente globalizado impulsionam as empresas a atuarem com um sistema de informação competente, assegurando níveis mais altos de produtividade e eficácia. Para o autor O'BRIEN (2004), a finalidade de utilizar os sistemas de informação é a concepção de um ambiente empresarial em que as informações sejam seguras e possam transcorrer na estrutura organizacional.

2.1.1 CLASSIFICAÇÃO DOS SISTEMAS

Os sistemas de informação são classificados de acordo com as funções que desempenham, como, níveis de gestão, tipos de decisões e grau de funcionalidade, objetivando atender os interesses organizacionais. Segundo LAUDON E LAUDON (2004, p.40) os sistemas são divididos em quatro níveis organizacionais, que dão auxílio na tomada de decisões dentro da empresa.

A classificação dos sistemas de informação acontece da seguinte forma:

- Sistema do nível operacional: Tem por objetivo acompanhar as principais transações da empresa, como vendas, folhas de pagamento e contas a receber.
- Sistema do nível de conhecimento: Ajuda na organização da empresa e auxilia na integração de novas tecnologias.
- Sistema do nível gerencial: Utilizados para dar suporte às funções de planejamento e controle no processo organizacional da empresa, fornecendo informações em tempo hábil para tomada de decisão.
- Sistema do nível estratégico: Esse sistema é utilizado pelo gerente sênior nas questões estratégicas da empresa.

Os sistemas de informação possuem alguns tipos específicos que correspondem a cada nível organizacional, dentre eles podem ser citados os seis mais importantes:

- Sistema de Processamento de transações: É um sistema utilizado no nível operacional com a função de registrar as transações diárias para uma melhor condução dos negócios da empresa.

- Sistemas de trabalho com conhecimento: Sistema utilizado no nível de conhecimento da organização, auxiliando os trabalhadores do conhecimento na criação de novas informações para serem integradas a empresa.
- Sistemas de escritório: Sistema utilizado com objetivo de aumentar a produtividade na manipulação dos dados de um escritório.
- Sistemas de informação gerencial: Sistema utilizado no nível gerencial da empresa, que tem por objetivo fornecer relatórios aos gerentes e controlar as decisões no nível gerencial.
- Sistemas de apoio à decisão: Ajuda os gerentes a tomar decisões não usuais, que se alteram com rapidez e que não são facilmente especificadas com antecedência.
- Sistemas de apoio ao executivo: Utilizados no nível estratégico da empresa ajudando a alta gerência na tomada de decisões estratégicas, como o planejamento a longo prazo.

2.2 TECNOLOGIAS DA INFORMAÇÃO MÓVEL

As tecnologias móveis passaram por um grande desenvolvimento desde que os primeiros telefones celulares foram colocados no mercado, no fim da década de 70. Desde então, as estatísticas de adoção dos dispositivos de comunicação sem fio em todos os cantos do planeta se intensificaram, na evolução tecnológica que originou a convergência digital do mesmo modo em que desdobrou e potencializou as funções dos aparelhos. Para SABILA, VARGAS e VIVA (2013) os dispositivos móveis, atualmente, estão sendo empregados nas mais diferentes áreas. Esta utilização vem se expandido cada vez mais, porque existe uma natural evolução social em que as gerações passadas tem se apossado cada vez mais destas tecnologias, e as gerações mais novas, agora consideradas “nativos digitais”, já tornaram tais dispositivos uma extensão do lar ou, até mesmo, de seu próprio corpo.

Alguns estudos afirmam que os usuários de dispositivos móveis não só os utilizam para desempenhar tarefas, mas também como forma de expressão pessoal de suas identidades. De acordo com os estudos de JARVENPAA E LANG (2005) os paradoxos da tecnologia móvel emergem no processo de ação e experiência desta tecnologia a partir da perspectiva do usuário, levando em consideração que tais ações e experiências são dependentes de fatores situacionais e contextuais. Isto é, o contexto social, organizacional, tecnológico e cultural influenciam as motivações e objetivos do indivíduo para usar uma tecnologia móvel.

Assim sendo, as decisões para usar a tecnologia móvel em diversas situações para diferentes propósitos, como comunicação, coordenação, eficiência, mobilidade e socialização, acabam por afetar, reforçar ou modificar aqueles contextos. Portanto, independentemente da finalidade específica, eventualmente o uso da tecnologia pode gerar situações de conflito para o usuário, ou seja, de situações paradoxais. Beneficiando uma utilização mais frequente, há a possibilidade de personalização destes dispositivos que acompanham o estilo de vida de cada pessoa, permitindo a customização e o acompanhamento de tendências de modo bem como de aplicativos que podem, desde facilitar o dia-a-dia, até aplicativos de entretenimento. Segundo SQUIRRA (2010) no quadro social atual, a evolução tecnológica, com a ampliação das possibilidades de comunicação *on-line* – agora, substancialmente móvel – no princípio do *anytime, anywhere, anyhow*, se viabiliza através dos dinâmicos, plurais e interativos recursos da comunicação digital que acenam para a necessidade de uma formatação diferenciada dos modelos e práticas para a educação.

2.3 APRENDIZAGEM COM MOBILIDADE

A educação sempre foi uma das principais preocupações e prioridades para uma sociedade que busca enriquecimento em todos os aspectos. Os métodos ultrapassados, pouco intuitivos, e a crescente necessidade de capacitação, formação e atualização profissional, aliada à exigência de mobilidade, contribuíram para o surgimento deste tipo de tecnologia *mobile learning*. Até a rede mundial de computadores mudou sua forma e alguns portais adaptaram-se a telas pequenas de celulares, para que a informação pudesse chegar a qualquer pessoa e em qualquer lugar (DA SILVA, DE OLIVEIRA, BOLFE, 2013).

Para uma geração que já nasceu submersa na tecnologia, fazer o uso dos recursos disponíveis pode ser mais produtivo e eficiente, uma vez que, produzindo mídias, essas poderão ser facilmente compartilhadas no meio virtual. Embora o uso inadequado possa prejudicar o rendimento dos alunos, esses equipamentos, quando utilizados com objetivos específicos e bem definidos, são capazes de promover a interação e auxiliar no processo de ensino-aprendizagem, é o que aponta MACHADO (2010) “*que esses dispositivos podem ser incluídos em projetos educacionais*”.

ALMEIDA(2001) ainda afirma que “*diante do atual contexto escolar brasileiro, os educadores necessitam de alternativas pedagógicas que auxiliem o processo de ensino aprendizagem de forma mais eficiente*”. Assim, fica claro, que a sociedade necessita de ferramentas que agreguem valor ao processo de ensino, de modo que os conteúdos sejam facilmente absorvidos pelos alunos. A frequente utilização dos recursos tecnológicos nas

escolas vai além da inovação da forma de se ensinar, e vem para reforçar a ideia de recurso eficiente do ponto de vista pedagógico.

Os dispositivos tecnológicos estão invadindo as salas de aulas, contudo os seus recursos são pouco explorados. Às vezes, um kit multimídia com computador e *data show* não está disponível, mas quase todo aluno possui um celular e traz a tecnologia para a sala de aula, mas quase nada temos que permita seu uso para educação.

A ideia é incorporar as tecnologias digitais, principalmente as móveis, para promover a mobilidade na educação, por meio de aplicativos específicos e recursos disponíveis. É tirar proveito dos milhares de celulares dos alunos e inseri-los no plano de aula, a fim de compartilhar experiências, transformar o conhecimento em valor e estimular o interesse no conteúdo abordado, fazendo com que o processo de ensino-aprendizagem seja algo agradável para o aluno, bem como para o educador.

Essas mudanças de concepções podem ser introduzidas com a presença dos dispositivos móveis portando o sistema operacional *Android*. Software esse, capaz de oferecer inúmeras aplicações com propósitos educativos, que poderão propiciar as condições para os alunos desenvolverem a capacidade de procurar e selecionar informações, resolvendo os problemas e auxiliando-os a aprender.

3 AMBIENTE DE DESENVOLVIMENTO

Neste capítulo, são apresentadas as tecnologias e as ferramentas utilizadas ao longo do trabalho, bem como os detalhes importantes de autenticação e o protocolo de comunicação entre o aplicativo e a *Gmail* API.

3.1 SISTEMA OPERACIONAL *ANDROID*

O surgimento do sistema operacional *Android* foi um grande avanço na área da tecnologia de telefonia móvel. Até então todas as empresas desse setor disponibilizava de um *software* próprio, e não permitia a utilização de seu *software* em celulares de fabricantes diferentes. Sendo a primeira plataforma de aplicativos para telefones celulares, de código fonte aberto, que se fez notar pelos maiores mercados de telefonia celular do mundo (ABLESON et al., 2013).

Segundo GLAUBER (2015), o sistema operacional *Android* está presente em cerca de um bilhão de dispositivos, principalmente nos *smartphones* e *tablets*, podendo ser encontrados em outros dispositivos, como TVs, automóveis, relógios etc. Com a expansão das tecnologias móveis, fez surgir uma gama de aplicações para esses dispositivos, possibilitando o rápido crescimento da plataforma.

Sob essa perspectiva a empresa *Google* de serviços e *software online*, comprou a *Android inc* uma pequena empresa de desenvolvimento de *software* para celulares, com objetivo de criar de uma plataforma aberta de fácil adaptação e que fosse ao mesmo tempo gratuita. LECHETA (2011) afirma que o grande avanço desse *software* proporcionou, a criação de uma aliança comercial que culminou, em novembro de 2007, na fundação da OHA (*Open Handset Alliance*), tendo como principal objetivo produzir padrões abertos para dispositivos móveis que pudessem ser personalizados por qualquer fabricante.

À medida que o sistema operacional *Android* foi crescendo a OHA decidiu desenvolver dispositivos de *hardware* compatíveis com o novo sistema, possibilitando a grande adesão de fabricantes de dispositivos móveis. No início essa associação contava com 34 empresas e atualmente conta com 81, dentre elas podem ser citadas a: *Samsung*, *LG*, *Motorola* e muitas outras. (DEITEL et. al., 2013).

LECHETA (2012, p.21) pontua que:

“Para os fabricantes de celulares, o fato de existir uma plataforma única e consolidada é uma grande vantagem para criar novos aparelhos. A grande vantagem para eles é que a plataforma também é livre e de código aberto. A licença do

Android é flexível e permite que cada fabricante possa realizar alterações no código-fonte para customizar seus produtos, e o melhor de tudo, sem necessidade de compartilhar essas alterações com ninguém.”

3.1.1 PLATAFORMA ANDROID

Atualmente tem-se presenciado uma revolução no setor de telefonia móvel. Uma das principais mudanças que contribuíram para essa revolução foram os dispositivos móveis, que cada vez mais detém um grande poder de processamento. A consequência desses avanços tecnológicos é grande, em virtude do volume de informações em um curto intervalo de tempo. Essa presença universal da informação tem por objetivo encurtar as distancias entre as pessoas e oferecer mudanças no contexto social e econômico.

O mercado de smartphones tem apresentado uma crescente alta nas vendas, o que tem motivado os fabricantes desses dispositivos a oferecerem, aparelhos mais modernos com suporte a diversas tecnologias, como a integração do GPS (*Global Positioning System*) que é um sistema de navegação por satélite que recebe informações sobre a posição de algo em qualquer horário. Esses dispositivos funcionam sob um *software* cuja função é criar uma camada de controle entre o *software* e o *hardware*.

No momento presente o mercado telefonia móvel tem oferecido uma gama de opções ao usuário quando se trata de sistemas operacionais para *smartphones* e *tablets*. Segundo JORDÃO (2013) o sistema operacional *Android* é o mais utilizada no mundo, uma vez que este foi pioneiro no lançamento de alguns recursos, em virtude de ser baseado no *kernel 2.6* do *Linux*, sendo capaz de gerenciar todos os processos, que vão desde a memória, *threads*, segurança dos arquivos e pastas, até redes e *drivers*. LECHETA (2013) afirma que os aplicativos *Android* são desenvolvidos utilizando a linguagem de programação *Java* e podem ser portados com bastante facilidade, já que o *kernel* mantém um nível de abstração entre o *hardware* do dispositivo e o restante do sistema.

3.1.2 ARQUITETURA ANDROID

O sistema operacional *Android* tem uma arquitetura dividida em camadas que trabalha em conformidade com o *kernel* do *Linux*. Dessa forma essas ferramentas interagem entre si funcionando como uma pilha, na qual cada camada apresenta recursos dependentes da camada posterior, necessitando de uma comunicação entre elas, visto que todas as camadas oferecem suporte à outra (UZEJKA, 2011).

De acordo com LEMOS (2011, p.10) a arquitetura do *Android* dá-se da seguinte forma:

“A arquitetura da plataforma *Android* é constituída por cinco módulos: *Kernel Linux*, Biblioteca, *Runtime* (ambiente de execução), *Framework* de Aplicações e Aplicações. A arquitetura da plataforma é baseada no *kernel* do *Linux*, ele funciona como uma camada de abstração entre o *hardware* e o restante dos *softwares* da plataforma. Com isso, ele tem várias formas de execução das aplicações. Por exemplo: gerenciamento de memória, gerenciamento de processos, pilhas de protocolos, módulo de segurança, entre outros.”

Pode ser observada na (Figura 1) a arquitetura do sistema operacional *Android* apresentando uma visão mais ampla e fornecendo um melhor entendimento sobre a plataforma.

Na primeira camada da pilha, que é o nível zero, encontra-se o *Kernel* do *Linux* baseado no sistema operacional *Linux* versão 2.6. Essa camada vai funcionar como um nível de abstração entre o *hardware* e *software*, na qual são encontrados os programas que fazem o gerenciamento de memória, configuração de segurança e os *drivers* de *hardware* que juntos fazem o controle dos dispositivos de rede, do teclado e demais componentes que constituem o dispositivo. Algumas funções do *Kernel* são utilizadas pelo *Android*, objetivando otimizar o processamento das aplicações e o gerenciamento da memória (GOMES; FERNANDES; FERREIRA, 2012, p.9).

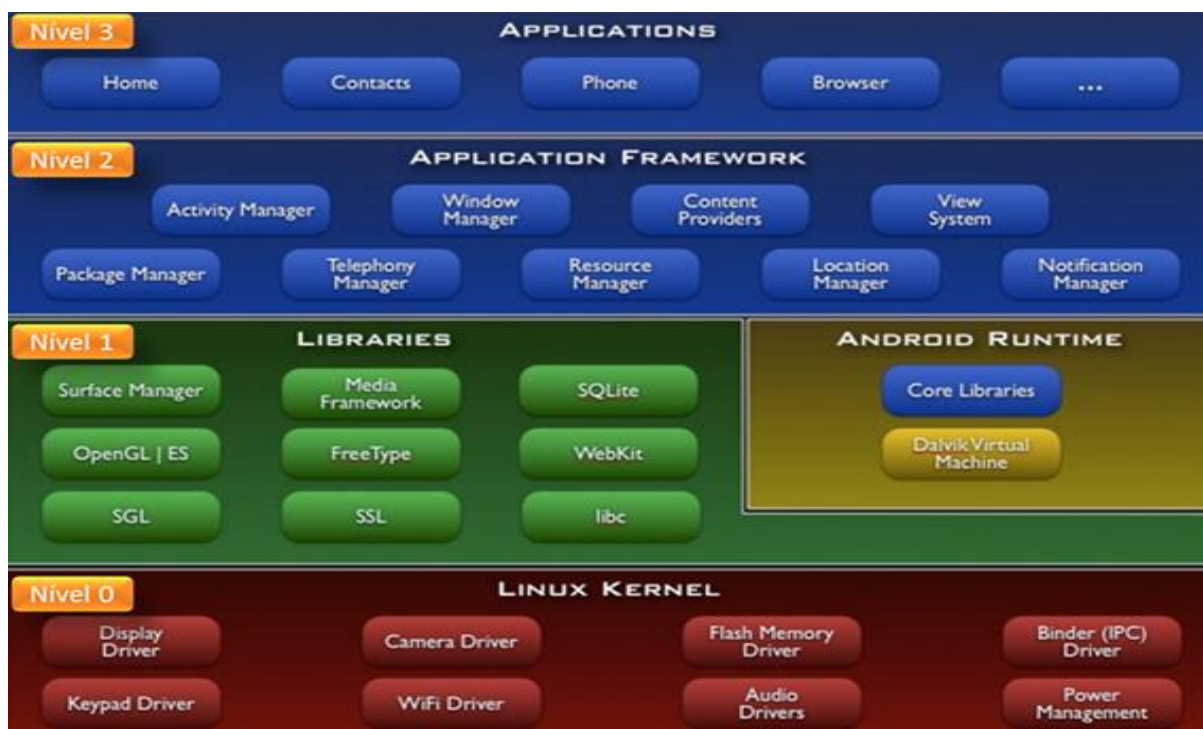


Figura 1- Arquitetura *Android*

Fonte: <http://www.tiselvagem.com.br/geral/desenvolvendo-para-android-arquitetura-android/>

No nível um, são encontradas as camadas de bibliotecas (*Libraries*) que possuem um conjunto de instruções, que informam o dispositivo de como lidar com vários tipos de dados, como as bibliotecas do *C/C++* utilizadas pelos componentes do sistema. Nessa camada encontra-se outro grupo de bibliotecas dentre elas a *Dalvik Virtual Machine* (DVM) que substitui a *JVM (Java Virtual Machine) software* responsável por carregar e executar os aplicativos *Java*. (GADELHA, 2011).

A Máquina Virtual *Dalvik* executa apenas arquivos do tipo “.*dex*” (*Dalvik Executable*) permitindo que cada aplicação execute seu próprio processo, em sua própria instância de máquina, de modo que nenhuma aplicação possa depender de outra. Essa técnica utilizada pela DVM tem como objetivo melhorar o gerenciamento da memória e o desempenho de *hardware* em dispositivos que apresentam pouco poder de processamento.

PEREIRA (2009, p.08) pontua que:

“A pequena camada do ambiente de execução (*Android Runtime*) é uma instância da máquina virtual *Dalvik*, criada para cada aplicação executada no *Android*. A *Dalvik* é uma máquina virtual com melhor desempenho, maior integração com a nova geração de *hardware* e projetada para executar várias máquinas virtuais paralelamente. Foi projetado para funcionar em sistemas com baixa frequência de CPU, pouca memória RAM”.

O *Android* possui um conjunto de bibliotecas que são utilizadas pelos componentes do sistema, dentre elas estão:

- *System C library* – É uma implementação proveniente da biblioteca C, responsável por fazer alocação e deslocação de memória, sendo utilizada por aplicações que apresentam a mesma arquitetura do sistema operacional *Linux*.
- *Media Library* – Biblioteca que suporta os mais diversos formatos de imagem áudio e vídeo, dentre eles: *MP3, MP4, JPG, PNG* etc.
- *Surface Manager* – Responsável por gerenciar o acesso ao sistema e as camadas de aplicação.
- *LibWebCore* – é uma biblioteca que acelera o poder de processamento do navegador da plataforma *Android*
- *SGL*– é uma biblioteca utilizada para modelagem de imagens *2D* e *3D*.
- *3D libraries* – Biblioteca responsável por renderizar modelos tridimensionais.
- *FreeType* – biblioteca para renderização de fontes vetoriais.

- *SQLite* – Biblioteca responsável por criar um banco de dados interno para o *Android*, que não necessita ser configurada. Disponível para armazenamento de dados de qualquer aplicação

Na camada de nível dois encontra-se o *framework*¹ de aplicação (*application framework*) que tem por objetivo proporcionar uma série de recursos aos desenvolvedores como a permissões de acesso e os serviços básicos de *interface*², em virtude de ser escrito na linguagem *Java* esse *framework* faz a utilização de varias APIs (*Application Programming Interface*). CIRIACO (2009) define essas APIs como: “*um conjunto de padrões que permite a construção de aplicativos e a sua utilização de maneira evidente para os usuários*”.

“Na camada do *framework* (*Application Framework*), encontramos todas as APIs e os recursos utilizados pelos aplicativos, como classes visuais, que incluem listas, grades, caixas de texto, botões e até um navegador web embutido, *View system* (componentes utilizados na construção de aplicativos), provedor de conteúdo (*Content Provider*), que possibilita que uma aplicação possa acessar informações de outra aplicação, ou até mesmo compartilharem as suas informações possibilitando a troca de informações entre aplicativos e gerenciadores de recursos (permite definir e carregar recursos *Run time*), gerenciador de localização (GPS e *CELL ID*), gerenciador de notificações (fornece informações sobre eventos que ocorrem no dispositivo), de pacote e de atividade que controla todo o ciclo de vida da aplicação e o acesso e navegação entre aplicativos (PEREIRA; SILVA, 2009, p.6)”.

Alguns componentes que constituem essa camada serão descritos a seguir:

- *Intent*- É uma classe do *Android* que realiza requisições ao sistema operacional, e o mesmo verifica a intenção da mensagem para tomar uma decisão, que pode ser desde iniciar uma nova *activity* ou abrir uma página *web*.
- *Activiy Manager* - Representa uma classe da aplicação sendo responsável por controlar as passagens de parâmetros e o gerenciamento de processos. Tendo como objetivo estabelecer uma interação na troca de informações por meio de *Intents* cuja função é encaminhar mensagens ao sistema operacional.
- *Service* - Executa as atividades em segundo plano, e na maioria das vezes são utilizadas para criar serviços de longo prazo para que não sofra com problemas de encerramento ou com componentes que utilizam *interface*.
- *BroadCastReceiver* - É considerado um gerenciador de notificações, cujo papel e receber é tratar os eventos provenientes de outras aplicações. Funcionando como um mecanismo de alerta, avisando ao usuário que algo aconteceu. Essa comunicação é

realizada por meio de uma *intent* que é enviada ao sistema, onde o mesmo irá fazer o gerenciamento e decidir em qual módulo do sistema será recebida essa mensagem.

- *ContentProvider* - É um componente do *Android* que permiti compartilhar dados entre várias aplicações. Funcionam por meio de uma *URI (Uniform Resource Identifiers)* na qual os aplicativos atendem alguns padrões de URI possibilitando a comunicação entre diferentes serviços que não se conhecem.
- *View* – Representa os componentes de *interface* gráfica utilizados para exibir os objetos visuais da aplicação, na tela do dispositivo.

No nível três encontra-se a camada de aplicação (*Applications*) contento os aplicativos básicos do *Android*, como o programa de *sms*, mapas, *e-mail*, navegador entre outros, podendo ser substituídos pelo usuário, sem que altere o funcionamento do sistema.

3.1.3 VERSÕES ANDROID

Desde o surgimento do *Android*, esse sistema vem recebendo um grande número atualizações, que incrementaram novas funcionalidades e correção de erros das versões anteriores. O qual sempre busca inovar seu ambiente gráfico e fazer melhorias internas. Fica a cargo dos fabricantes de aparelhos fazerem a distribuição das atualizações, uma vez que esse *software* é desenvolvido pela *Google* e a mesma se encarrega de fazer as correções de erros e adição de novas funcionalidades.



Figura 2 : Versões *Android* (2014)

Fonte: <http://www.ferramentaspc.com.br/2015/02/como-descobrir-versao-do-android.html>

Cada versão do *Android* recebe nome de sobremesas, cujas iniciais estão em ordem alfabética, de acordo com a (Figura 2), iniciando com a versão *Cupcake*, *Donut*, *Eclair*, *Froyo*, *Gingerbread*, *Honeycomb*, *Ice Cream Sandwich*, *Jelly Bean*, e a mais atual, *Lollipop*.

Segundo MEIRA (2014) essas versões do sistema operacional *Android* podem ser customizadas por fabricantes de aparelhos, operadoras de telefonia móvel e desenvolvedores. Visando diversificar as funcionalidades e os recursos que o sistema oferece. Mas isso pode trazer sérios problemas de fragmentação, pois existe ainda um grande número de dispositivos com *hardware* antigo que apresenta incompatibilidade com as novas versões do sistema.

A Figura 3 apresenta uma distribuição das versões do sistema *Android* no ano de 2013, os dados apresentados no gráfico são baseados nos dispositivos que acessaram a loja de aplicativos da *Google*. Percebe-se no gráfico que quem ocupa a maior fatia é o *Gingerbird*, lançado em 2013 que trouxe vários recursos de melhoria na interface e a integração dos sensores de movimento (CANALTECH, 2013).

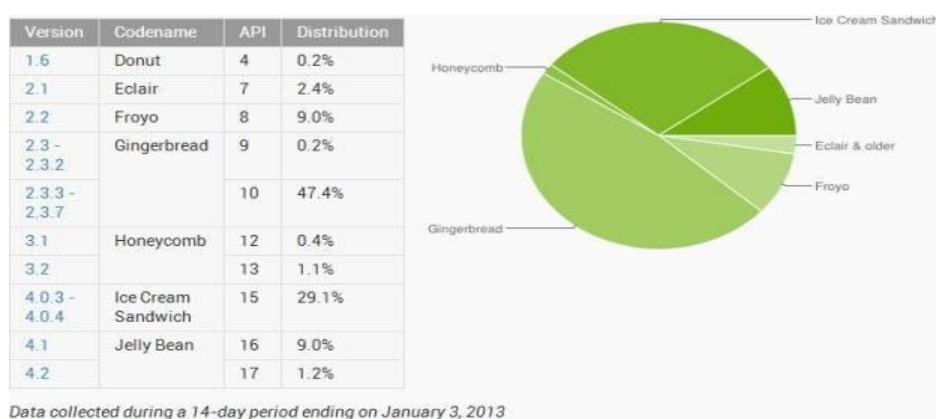


Figura 3: Distribuição do *Android* nos dispositivos

Fonte: <http://canaltech.com.br/noticia/android/Android-Jelly-Bean-ja-detem-10-do-mercado-e-Gingerbread-fica-abaixo-dos-50/>

No entanto, nem todos os usuários do *Android* tem acesso as recentes versões do sistema, isso acontece em virtude da incompatibilidade que o *software* apresenta em relação ao *hardware* de alguns dispositivos.

3.2 GMAIL API

Atualmente nos encontramos em uma sociedade dinâmica na qual precisamos nos adequar constantemente, quando se refere ao uso de recursos tecnológicos para estabelecermos a comunicação com outras pessoas. Segundo MONTEIRO (2010) assim que o homem começou a escrever cartas como forma de transmissão de informação, ele sentiu a necessidade de evoluir para um tipo de serviço que fosse mais rápido e eficiente. Essa necessidade deu origem a diferentes meios de comunicação.

Segundo DUARTE (2014) com a evolução da *internet*, as cartas que antes eram entregues a uma pessoa física, passaram a ser eletrônicas sendo chamadas de *e-mail* ou correio eletrônico, que substituiria os endereços físicos das pessoas por endereços eletrônicos tais como (nomeusuario@provedor.com).

A facilidade em poder utilizar um serviço de *e-mail*, fez com que muitas empresas deixassem de ver esse serviço apenas como trocas de mensagem entre os usuários, passando a utiliza-lo não só como fator de produtividade, mas também como um canal de comunicação entre a empresa e os seus clientes.

Esse serviço de *e-mail* ou correio eletrônico se tornou popular em razão de sua facilidade, na troca de informações e dados entre as pessoas, através de um sistema de *e-mail*.

Esse sistema utiliza o protocolo SMTP (*Simple Mail Transfer Protocol*) que tem como função transferir *e-mails* de um cliente para um servidor de *e-mail*, bem com a transferência de e-mails entre servidores (CARISSIMI, ROCHOL, GRANVILLE, 2009).

BROOKSHEAR (2013) pontua que:

“Existem dois protocolos populares que podem ser usados para acessar *e-mails* que chegaram e estão acumulados no servidor de *e-mail* de um usuário. São eles: POP3 (*Post Office Protocol version 3*) e IMAP (*Internet Message Access Protocol*). O POP3 é o mais simples dos dois. Usando POP3 um usuário transfere (baixa) as mensagens para seu computador local, no qual elas podem ser lidas, armazenadas em diversos diretórios, editadas e manipuladas de qualquer forma que o usuário desejar. Isso é feito na máquina local do usuário usando o armazenamento em massa. O IMAP permite a um usuário armazenar mensagens e matérias relacionadas na mesma máquina do servidor de correio.”

Esse processo é demonstrado através da Figura 4-a que apresenta o modelo de transferência de *e-mails* utilizando os protocolos SMTP, POP/IMAP.

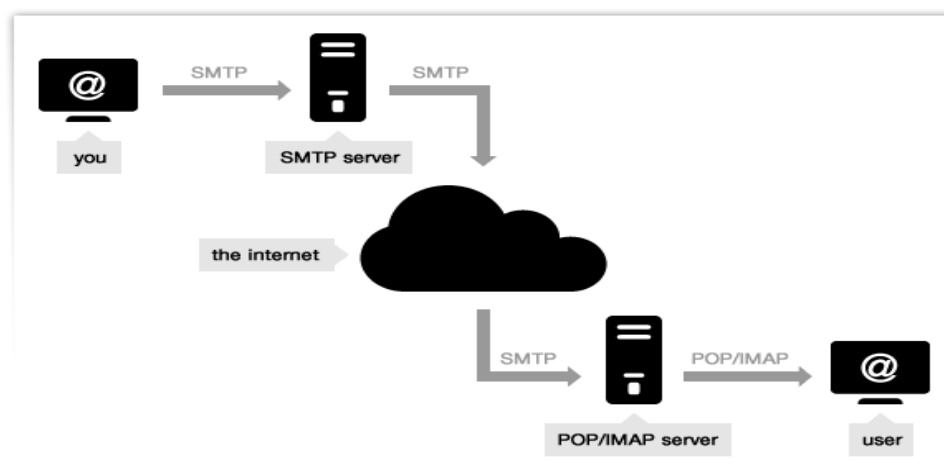


Figura 4a- Servidor SMTP

Fonte: <http://www.serversmtp.com/pt-pt/o-que-e-servidor-smtp>

O elevado crescimento de aplicações para dispositivos móveis que oferecem serviços de *e-mail* motivou a *Google* lançar a *Gmail API* objetivando substituir o IMAP, um protocolo no qual o *software* de *e-mails* estabelece a conexão com o servidor e realiza o sincronismo das mensagens. Após realizar esse processo, o protocolo modifica o status da mensagem tanto no *software* quanto no servidor, permitindo que as alterações feitas nas mensagens aconteçam quase que em tempo real. Essa configuração é fundamental para quem acessa a conta em diferentes locais, uma vez que essa mensagem apresentará o mesmo status independentemente do local que for acessada.

Para PEPATO (2014) o objetivo da *Gmail API* é facilitar o desenvolvimento de uma aplicação que atribua as características do *Gmail* como: ler mensagens, enviar mensagens, procurar por mensagens específicas etc. Uma vez que essa nova API oferece acesso *RESTful* ao *inbox* ou caixa de entrada com suporte as operações de CRUD (*create, read, update, delete*) em diferentes tipos de dados, como *messages, labels, drafts e threads*. Todos esses recursos disponíveis para aplicação necessitam de uma autorização do padrão *OAuth* que garante a segurança dos dados do usuário e da aplicação.

OAuth é um protocolo aberto de autenticação que possibilita ao usuário ter acesso a uma parte de seus dados por meio de APIs, que são disponibilizadas pela *Google, Facebook ou Twitter*, por exemplo. Segundo MEIRA (2014) no *OAuth*, o usuário autoriza junto ao serviço, que uma determinada aplicação tenha acesso a parte de seus dados. No momento que o usuário concede a autorização, a aplicação recebe um *token* com o qual consegue obter as informações desejadas. Dessa forma, as credenciais do usuário nunca são fornecidas a aplicações de terceiros.

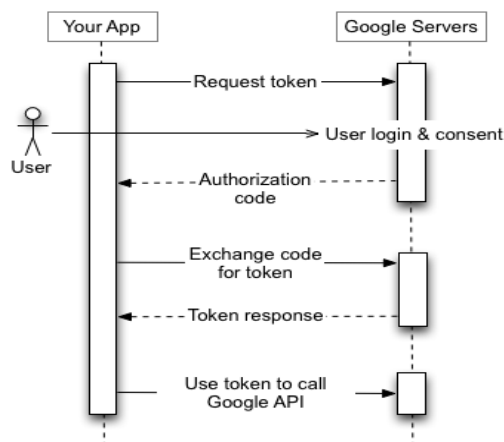


Figura 4b: *OAuth 2.0*

Fonte: <https://developers.google.com/identity/protocols/OAuth2>

3.3 LINGUAGEM JAVA

Para o desenvolvimento de aplicativos *Android*, *Java* tem sido a linguagem de programação mais utilizada, por ser de alto nível e de código fonte aberto, tornando-se a ferramenta perfeita para criação de aplicações *Android*, onde a mesma tem inúmeras comunidades na web, de fóruns e tutoriais que facilitam o desenvolvimento da aplicação (CURVELO et.al., 2015). Os aplicativos desenvolvidos em *Java* podem ser executados em uma grande quantidade de dispositivos, que não necessitam de código específico para a plataforma. Por ser orientada a objetos, possui acesso a bibliotecas de classes que ajudam a desenvolver aplicativos de maneira eficiente.

Essa linguagem foi uma escolha lógica para a plataforma *Android*, porque é poderosa, gratuita e de código-fonte aberto. O *Java* é usado para desenvolver aplicativos empresariais de larga escala, melhorar a funcionalidade de servidores Web, fornece aplicativos para aparelhos de consumidor (por exemplo, telefones celulares, *paggers* e assistentes digitais pessoais) e para muitos outros propósitos (DEITEL et. al, 2013, p.5).

O programa escrito em *Java* é transformado em *bytecodes* pelo compilador e são interpretados pela JVM. Em consequência disso programas escritos em *Java* podem ser executados em diferentes plataformas, desde que tenham uma versão da JVM instalada. Essa técnica de conversão garante uma melhor portabilidade aos programas *Java*, visto que esses programas podem ser executados em diferentes sistemas operacionais. (MASSAGO; SCHÜTZER, 2010).

3.4 *SQLITE*

O *SQLite* é o banco de dados interno do *Android*, que possibilita ao desenvolvedor armazenar os dados da sua aplicação em tabelas utilizando comandos SQL, por ser um banco de dados simples e eficiente, tornou-se popular entre os desenvolvedores. Como o *SQLite* é uma ferramenta *open-source* e integrado no *Android*, ele não requer configuração ou instalação para ser utilizado, tornando-o a escolha natural para um ambiente em que devemos prezar por desempenho, disponibilidade de memória e praticidade (LUZZI, 2013).

Na prática o *SQLite* suporta os recursos da sintaxe SQL (*Structured Query Language*), e uma de suas características é a quantidade de memória a ser utilizada no tempo de execução. Como não necessita de nenhuma configuração inicial para gerar o banco de dados, é preciso especificar a instrução SQL para que ele possa ser criado automaticamente,

visto que o *SQLite* não valida os campos armazenados se os mesmos forem iguais aos campos das colunas selecionadas.

Com o *SQLite* é possível fazer o uso de vários recursos presente em um banco de dados relacional, criar tabelas com as chaves primarias e estrangeiras, *views*, gatilhos e índices. Para criar uma tabela utiliza-se o comando *CREATE TABLE* da linguagem SQL e para realizar uma manipulação nas tabelas, são utilizados os comandos (*INSERT*, *UPDATE* E *DELET*) assim é possível fazer modificações no banco à medida que for necessário. Enfim, o desenvolver irá dispor de uma série de recursos que o ajudará no armazenamento das informações de sua aplicação.

3.5 ANDROID STUDIO

Quando se pensa em desenvolvimento de aplicativos para a plataforma *Android*, se faz necessário o uso de um ambiente de desenvolvimento integrado ou IDE (*Integrated Development Environment*) que melhor se encaixa as necessidades do desenvolvedor, de maneira a proporcionar maior produtividade no desenvolvimento de um projeto. Dentre as IDEs pode ser citado o *Android Studio* como uma das ferramentas de desenvolvimento mais utilizada, por oferecer ao programador uma capacidade de expansão e vários pacotes de tradução.

O *Android Studio* é um ambiente de desenvolvimento integrado, que foi desenvolvido pela *Google* para criar aplicações *Android*, que tem por objetivo oferecer ao desenvolvedor um ambiente robusto e de fácil usabilidade, com inúmeras ferramentas que permitem agilizar a criação do aplicativo. O *Android Studio* oferece um emulador de testes ao desenvolver, facilitando na execução das aplicações em diferentes dispositivos. Desse modo o desenvolver poderá analisar as mudanças visuais e o desempenho da aplicação em tempo real.

De acordo com ROCHA (2014) *Android Studio* integra:

“As funções do *software* incluem a edição inteligente de códigos, recursos para design de *interface* de usuário e análise de performance, entre outras coisas. A *Google* recomendava que os desenvolvedores utilizassem o IDE *Eclipse* para fazer aplicativos para o *Android*, mas agora a Gigante das Buscas oferece instruções para que os usuários migrem para o *Android Studio*.”

O *Google* oferece várias APIs para serem utilizadas nas aplicações dentre elas podem ser citada, o *Google Cloud Messaging* (GCM) que tem a função de enviar dados do servidor para a aplicação *Android* por meio de uma nuvem de dados, uma forma diferente de enviar as notificações para as aplicações. Os servidores GCM criam um enfileiramento de mensagens

que serão entregues através de um canal de comunicação entre o servidor e as aplicações *mobile*. Essa característica que o GCM oferece, elimina a necessidade da execução da aplicação, pois o sistema operacional executa um *broadcast* informando que a mensagem chegou (GLAUBER, 2015, p 404).

A Figura 5 ilustra a transferência de informações entre o servidor e aplicação *Android* utilizando o *Google Cloud Messaging*.

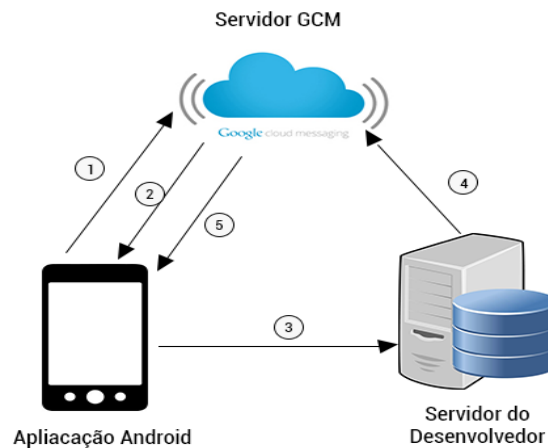


Figura 5: Funcionamento do GCM

Fonte em: <https://dariomungoi.wordpress.com/category/android-dev-iniciantes/page/2/>

Uma grande vantagem em poder utilizar o *Android Studio* para desenvolvimento *mobile*, é o fato dele importar projetos do *Eclipse*, nesse caso é necessário fazer a atualização do *Android Development Tool (ADT)* um *plugin* que tem por objetivo auxiliar no desenvolvimento de aplicações *Android*. O *Android Studio* oferece recursos de customização ao desenvolvedor, permitindo-o configurar o teclado de modo a diminuir o impacto na usabilidade da ferramenta de desenvolvimento. Dentre esses recursos está a filtragem de erros no *log do grade*, que ao executar um aplicativo o desenvolvedor poderá filtrar as mensagens específicas ou mensagens de erro da aplicação.

3.6 ENGENHARIA DE SOFTWARE

O *software* no cenário atual tem ocupado um lugar de destaque na economia mundial, isso se deve ao uso de computadores em diversas áreas do conhecimento, o que tem provocado uma crescente demanda por soluções tecnológicas. Segundo FALBO (2005, p.2) a engenharia de *software* surgiu para melhorar a qualidade do *software* e a produtividade na etapa de desenvolvimento.

A importância do *software* para as organizações tem motivado cada vez mais os desenvolvedores a utilizarem os processos da engenharia de *software*, na construção de soluções tecnológicas eficientes, que ofereça qualidade e segurança da informação. Desta forma, é necessário que haja um planejamento adequado, para que a qualidade do *software* seja alcançada, conforme foi estabelecido na etapa de requisitos. Cujo foco principal dessa qualidade é garantir que o produto final atenda as expectativas do cliente, dentro daquilo que foi acordado nos requisitos.

Para (VASCONCELOS et al., 2006, p.18) a engenharia de *software* não trata apenas da construção do sistema em si, mas também toda a documentação que será utilizada no desenvolvimento, bem como sua instalação e manutenção. Essas etapas de desenvolvimento passam por um ciclo de vida que contém os processos, as atividades e as tarefas que serão desenvolvidos nas fases de construção do sistema, abrangendo desde definição dos requisitos até a instalação do produto.

O ciclo de vida de um *software* é uma representação do esqueleto ou da estrutura pré-definida na fase de criação. Podendo ser dividido em diferentes modelos: Cascata, Prototipagem, Métodos formais, Espiral, Incremental etc. Esses modelos de ciclo de vida servem para guiar o engenheiro de *software* na construção do sistema. Pois não existe um modelo ideal que atenda todas as necessidades de um cliente, visto que alguns fatores como, tempo, custo de produção, complexidade do projeto, irão contribuir diretamente para a escolha desse ciclo.

Um modelo de ciclo de vida pode ser entendido como passos ou atividades que devem ser executados durante um projeto. Para a definição completa do processo, a cada atividade, devem ser associados técnicas, ferramentas e critérios de qualidade, entre outros, formando uma base sólida para o desenvolvimento. Adicionalmente, outras atividades tipicamente de cunho gerencial, devem ser definidas, entre elas atividade de gerência e de controle e garantia da qualidade (FALBO, 2005, p.7).

Segundo PRESSMAN (2011) uma metodologia de processo serve como alicerce para identificação das atividades que serão aplicados no projeto de engenharia de *software*. Por meio dessas informações o processo de *software* formará uma base para gerenciar os projetos e determinar quais conteúdos e métodos serão aplicados. Essa base compreende cinco etapas: comunicação, planejamento, modelagem, construção e emprego, podendo ser aplicada em qualquer projeto de *software*.

3.7 UML

A UML (*Unified Modeling Language*) é uma linguagem visual utilizada para modelar sistemas orientados a objetos, constituída por elementos gráficos que representam os paradigmas de orientação a objetos. Quando os elementos gráficos são definidos nessa linguagem, tem-se a possibilidade de construir os diagramas que irá representar as perspectivas do sistema (BEZERRA, 2013, p.14).

Esses elementos gráficos possuem uma sintaxe e uma semântica que definem o significado do elemento utilizado e sua funcionalidade. Cada elemento da UML pode ser extensível, permitindo que sejam adaptadas as características do projeto em desenvolvimento.

Quando se faz o uso da UML para modelar as características de um *software*, alguns processos devem ser levados em consideração, visto que cada processo tem sua particularidade em relação ao modo de encadear as tarefas a serem executadas. A UML utiliza cinco processos de desenvolvimento: análise de requisitos, análise, design, programação e testes. Essas fases não ocorrem de maneira sequencial, pois à medida que os problemas são identificados numa fase, está pode sofrer modificações e melhorias, contribuindo assim para o resultado final do sistema e gerando um *software* com alto padrão de qualidade e desempenho.

A UML é composta por 13 digramas, cujo objetivo é fornecer diferentes visões do sistema a ser desenvolvido, analisando e modelando sob vários aspectos, permitindo assim que cada diagrama complemente o outro. Cada diagrama fornece uma perspectiva sobre o sistema que está sendo modelado, desse modo todas as partes irrelevantes são dispensadas evitando a poluição do diagrama, cabe ao desenvolvedor corrigir erros de modelagem para evitar problemas de interpretação no futuro (PRESSMAN, 2011).

Os diagramas UML podem ser agrupados em duas grandes categorias: Estruturais e Comportamentais. O que os diferencia em relação a sua usabilidade, é que os diagramas estruturais são utilizados para documentar os aspectos estáticos do sistema, já os comportamentais têm como objetivo documentar os aspectos dinâmicos.

Durante o desenvolvimento de um *software* é importante fazer a utilização de vários digramas, isso facilita a descoberta de erros em digramas anteriores, evitando falhas futuras no decorrer do desenvolvimento da aplicação.

Cada diagrama da UML analisa o sistema, ou parte dele, sob uma determinada ótica. É como se o sistema fosse modelado em camadas, sendo que alguns diagramas enfocam o sistema de forma mais geral apresentando uma visão mais externa do sistema, como é o objetivo do Digrama de Casos de Uso, enquanto uns oferecem

uma visão mais técnica ou ainda visualizando apenas uma característica específica do sistema ou um determinado processo (GUEDES, 2011, p.30).

A Figura 6 seguir apresenta os digramas estruturais e comportamentais da UML 2.0.

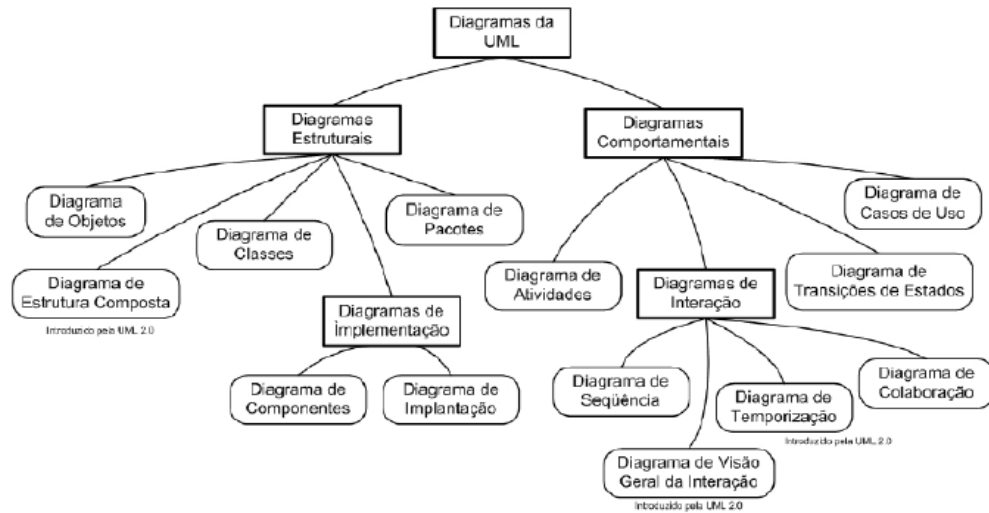


Figura 6: Representação dos Digramas da UML

Fonte: <http://evertongomede.blogspot.com.br/2010/09/representacao-dos-diagramas-da-uml-20.html>

Os diagramas UML exibidos Figura 6 se apresentam como a estrutura de uma classe. A utilização de vários diagramas na modelagem de um software é muito importante em virtude de alguns diagramas deixarem falhas, dessa forma é possível sanar alguns erros durante o processo de desenvolvimento da aplicação.

4 MODELAGEM DO APLICATIVO

Neste capítulo é descrito o modelo proposto para esse trabalho. Sendo detalhada as tecnologias utilizadas, as funcionalidades do sistema e os métodos para conclusão da aplicação.

4.1 CONSTRUÇÃO DA APLICAÇÃO

O sistema operacional utilizado para fornecer a gerência e a interação das atividades nas etapas de desenvolvimento da aplicação foi o *Windows 7 Ultimate 64-bit* instalado em um computador com 4 GB de memória RAM, com um HD de 500 GB, e um processador *core i3*. Cujos requisitos possibilitaram a instalação de algumas bibliotecas e *plug-ins* utilizados pelo *Android Studio*.

Para desenvolvimento de aplicações *Android* a *Google* disponibiliza duas opções aos desenvolvedores: o ADT e o *Android Studio*. Para construção desse aplicativo foi escolhido o *Android Studio* com versão 3.4 utilizando a API 19 do *Android Kitkat*. A escolha dessa IDE, seu deu em razão dos recursos que ela oferece:

- *Interface* amigável,
- Customização do teclado.
- Instalação das bibliotecas do SDK (*Software Development Kit*) sem complicações.
- Editor de *Layout* compatível com o tema escolhido.
- Renderização Inteligente
- Filtragem de erros de mensagens
- Criação de hierarquia entre as *Activities*
- Monitoramento da alocação de memória, permitindo ao desenvolvedor analisar o recurso que apresenta o maior consumo de memória.

Para modelagem do sistema foi utilizado o programa *Astah Ruting*, que possibilitou a criação de todos os diagramas UML, objetivando apresentar as características e funcionalidades do aplicativo.

Após a instalação do *Android Studio* é necessário fazer a configuração do emulador que será utilizado para realização dos testes. O emulador utilizado foi o *Genymotion*, uma vez que este apresenta um desempenho superior em relação ao emulador nativo do *Android*, o qual possibilita ao desenvolver utilizar todos os componentes de *hardware* presente em um dispositivo real (GLAUBER, 2015).

4.2 REQUISITOS DE SOFTWARE

Projetar um *software* nem sempre é uma tarefa fácil para o engenheiro de *software*, portanto é nessa fase de análise e requisitos que ele procura entender as necessidades do cliente, objetivando produzir um *software* de acordo com o que foi documentado. Os requisitos são divididos em duas categorias:

- Requisitos funcionais (RF): São as funções que o sistema deverá fornecer, e como ele irá reagir a entradas específicas em determinadas situações (SOMERVILE, 2007).
- Requisitos não funcionais (RNF): São aqueles que não estão diretamente relacionados às funções do sistema, podendo estar relacionados às propriedades do sistema como o tempo de execução, armazenamento em disco dentre outros fatores correlacionados a qualidade de um *software* (SOMERVILE, 2007).

Após a análise foi possível definir os requisitos funcionais e os requisitos não funcionais do aplicativo. O Quadro 1 apresenta os requisitos funcionais com seus identificadores e suas dependências.

Quadro 1: Requisitos funcionais

Identificador	Descrição	Depende de
RF01	O sistema solicitará do usuário um <i>login</i> e senha que está cadastrado no dispositivo móvel.	
RF02	O sistema possuirá apenas um usuário que nesse caso será o professor .	RF01
RF03	O sistema permitirá o cadastro de disciplinas e alunos, dando permissões ao usuário de fazer edição, alteração e exclusão dos cadastros citados.	
RF04	O usuário logado poderá fazer buscas por disciplinas e alunos.	RF01
RF05	O sistema permitirá que o usuário, possa ligar, para o aluno bem como poderá enviar <i>sms</i> , ver no mapa e enviar <i>e-mail</i> para um determinado aluno.	
RF06	O usuário poderá selecionar a disciplina e os arquivos as serem enviados para alunos cadastros.	
RF07	O sistema permitirá ao usuário fazer a leitura de <i>e-mails</i> que chegarão à aplicação.	

O Quadro 2 apresenta os requisitos não funcionais, contendo os identificadores de cada requisito do sistema, bem como sua categoria e sua dependência caso exista e dependências.

Quadro 2: Requisitos Não Funcionais

Identificador	Descrição	Categoria	Depende de
RNF01	Para obter os dados da conta cadastrada no dispositivo móvel, a aplicação precisará ser assinada para obter um certificado digital que é único para cada aplicação	Segurança	
RNF02	Para fazer autenticação na aplicação e necessário que o usuário tenha uma conta do <i>Google</i> para ter acesso aos <i>e-mails</i> da <i>Gmail</i> API.	Manutenibilidade	RNF01
RNF03	O sistema só será executado em dispositivos móveis com o sistema operacional <i>Android</i> a partir da versão 3.0	Portabilidade	
RNF04	O sistema não possibilita a integração de arquivos vindos de outras aplicações. Uma vez que o explorador de arquivos só possui permissões de acesso da memória interna e externa do aparelho	Confiabilidade	
RNF05	A <i>interface</i> do sistema deve ser amigável e objetiva, ou seja, suas funções devem estar bem visíveis e possuir uma padronização de cores.	Usabilidade	

4.3 DIAGRAMA DE CASO DE USO

Para GUEDES (2011) o diagrama de casos de uso serve para facilitar a compreensão do sistema em relação as suas funcionalidades, dentre os demais diagramas UML, é o que melhor atende as necessidades do engenheiro de *software*, pois apresenta uma visão geral dos componentes do sistema, facilitando na identificação dos requisitos do *software*.

Com intuito de obter uma melhor compreensão das funcionalidades do sistema, foi criado o diagrama de caso de uso, no qual são apresentados os detalhes técnicos das funcionalidades do sistema. A (Figura 7) apresenta o diagrama de caso de uso, utilizando as técnicas de modelagem definidas pela UML. A seguir serão demonstrados os detalhes da documentação explicitando o ator que está relacionado o caso de uso.

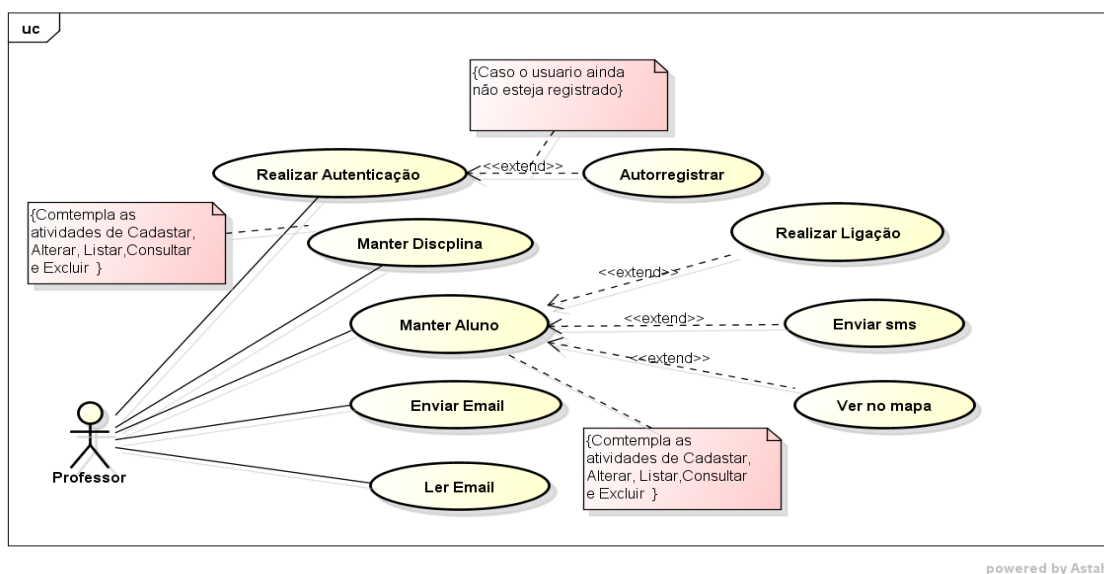


Figura 7 - Diagrama de Caso de Uso

Pode-se perceber que no diagrama de caso de uso, o ator (professor), realiza uma autenticação no sistema com a conta cadastrada no dispositivo. A relação <<extend>> entre o caso de uso (Realizar Autenticação) e (Autorregistrar) significa dizer que o caso de uso que foi estendido, funcionará como o caso de uso base, uma vez que novos passos serão inseridos. Feita a autenticação, o professor deverá cadastrar uma disciplina com seus respectivos alunos. As relações <<extend>> estabelecidas com o caso de uso (manter aluno), significa dizer que novos passos deverão ser seguidos, objetivando executar a funcionalidade do caso de uso estendido. A associação estabelecida do ator (professor) com os casos de uso (Enviar *E-mail*) e (Ler *E-mail*) significa dizer que os dois irão colaborar para uma determinada ação realiza pelo ator (professor), que é de enviar os *e-mails* aos alunos e ler os *e-mails* de reposta dos integrantes da disciplina.

A seguir são apresentados os detalhes do caso de uso da (Figura 7), explicitando os atores que estão associados ao caso de uso, uma descrição:

- **Realizar Autenticação:**

Ator: Professor.

Descrição: O professor terá que fazer autenticação com a sua conta cadastrada no dispositivo, caso ele não tenha, pode ser criada uma nova conta que dará acesso ao sistema.

- **Cadastrar Disciplina**

Ator: Professor

Descrição. Responsável por realizar o cadastro das disciplinas, com permissões para editar, listar, consultar e excluir uma disciplina.

- **Cadastrar Alunos**

Ator: Professor

Descrição: Responsável por cadastrar os alunos nas disciplinas. Com as permissões para editar, listar e excluir um aluno.

- **Enviar *E-mails***

Ator: Professor

Descrição: Responsável por fazer o envio de *e-mails* aos alunos cadastrados nas respectivas disciplinas.

- **Ler *E-mails***

Ator: Professor

Descrição: Responsável por fazer a leitura dos *e-mails* que irão chegar na aplicação.

- **Realizar Ligação**

Ator: Professor

Descrição: Responsável por realizar a ligação para um determinado aluno.

- **Enviar SMS**

Ator: Professor

Descrição: Responsável por enviar *sms* um determinado aluno.

- **Visualizar endereço do aluno no mapa**

Ator: Professor

Descrição: Responsável por visualizar no mapa o endereço de um determinado aluno.

4.4 DIAGRAMA DE CLASSES

GUEDES (2011, p.101) afirma que:

“O diagrama de classes é um dos mais importantes e mais utilizados da UML. Seu principal enfoque está em permitir a visualização das classes que compõem o sistema com seus respectivos atributos e métodos, bem como em demonstrar como as classes do diagrama se relacionam, complementam e transmitem informações entre si. Esse diagrama apresenta uma visão estática de como as classes estão organizadas, preocupando-se em como definir a estrutura lógica das mesmas.”

O diagrama de classe (Figura 8) apresenta de maneira detalhada as informações contidas em cada classe com seus respectivos atributos e métodos.

Esse diagrama é composto por quatro classes principais (Disciplina, Aluno, Mensagem, Resposta). Como o aplicativo está utilizando a API do *Gmail* para atribuir as características de um sistema *e-mail*, ela deixa o programador livre de criar as classes responsáveis por se comunicar com o servidor. Sendo a API responsável por estabelecer essa comunicação.

Conforme a (Figura 8) as classes apresentam três divisões que são descritas pela UML. A primeira divisão define o nome da classe utilizada no sistema, à segunda apresenta os atributos, uma representação das características de uma classe e a terceira divisão apresenta os métodos ou operações contidas nela.

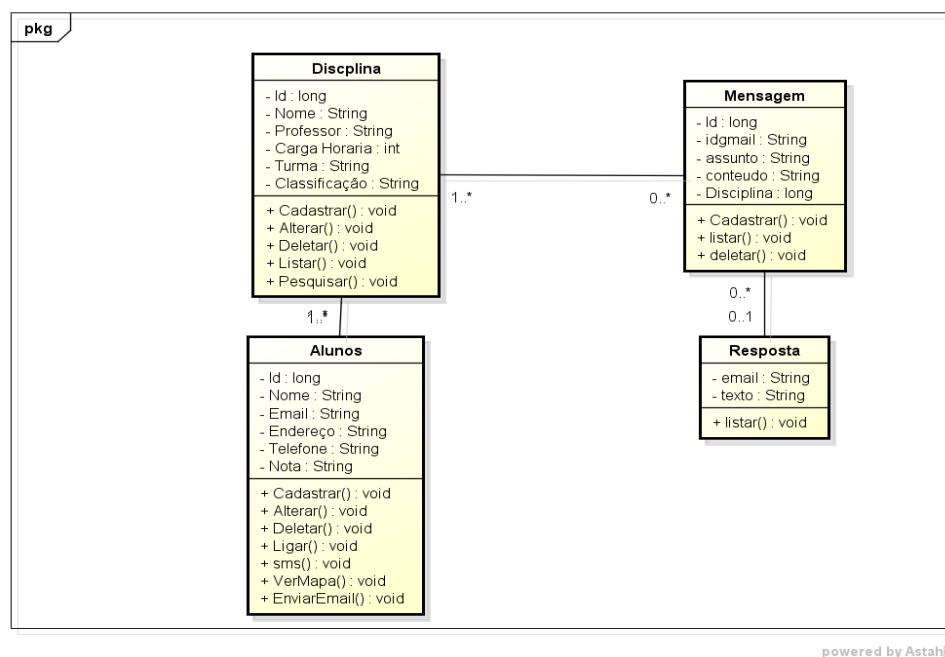


Figura 8- Diagrama de Classes

A classe *Disciplina* é responsável por realizar um CRUD na aplicação cujo objetivo é manter os dados atualizados. A classe *Aluno* tem a função de alterar as informações de um determinado aluno, como criar ou alterar as informações do mesmo. Quando uma disciplina é excluída, todas as informações que estão vinculadas a ela no banco de dados são excluídas, uma vez que, é implementado o conceito de chave estrangeira cuja função é criar um link de dados entre duas tabelas, objetivando garantir a integridade dos dados referenciados.

A classe Mensagem tem a função de salvar no banco de dados, o assunto de cada mensagem enviada, e desse modo é apresentado um formato de lista na aplicação, conforme a ordem de envio.

A classe Resposta tem com objetivo listar as mensagens respondidas por cada aluno. Cada resposta é encaminhada ao assunto correspondente, uma vez que, cada mensagem tem um único “id” de identificação. Dessa forma quando o usuário selecionar um assunto para verificar novas mensagens, a aplicação realizará uma requisição a *GMAIL* API para obter novas mensagens.

4.5 FUNCIONALIDADES DO SISTEMA

O sistema foi desenvolvido para a plataforma *Android*, objetivando atender os professores com ferramenta de auxílio no envio de atividades aos alunos. O propósito do aplicativo consiste em fins educativos visando estabelecer uma interação entre o professor e o aluno.

Após a instalação do aplicativo, será necessário que o usuário escolha uma conta já cadastrada no dispositivo (Figura 9 a), caso contrário ele poderá adicionar uma nova conta, que dará permissões de acesso ao sistema.

Essa autenticação dará permissões de acesso a *Gmail* API possibilitando ao usuário ler e enviar as mensagens através da aplicação. Ao selecionar a conta, o usuário é apresentado à tela de boas-vindas do sistema (Figura 9 b), na qual possui um botão que deve ser clicado para dar início ao cadastro da disciplina.

Na tela Cadastrar Disciplina (Figura 10) o usuário terá que selecionar o ícone (+) para inserir uma nova disciplina. Dessa forma cada disciplina inserida no banco de dados terá uma única chave de identificação, o que dará total segurança para ao usuário no momento de alterar ou excluir os dados de uma disciplina.

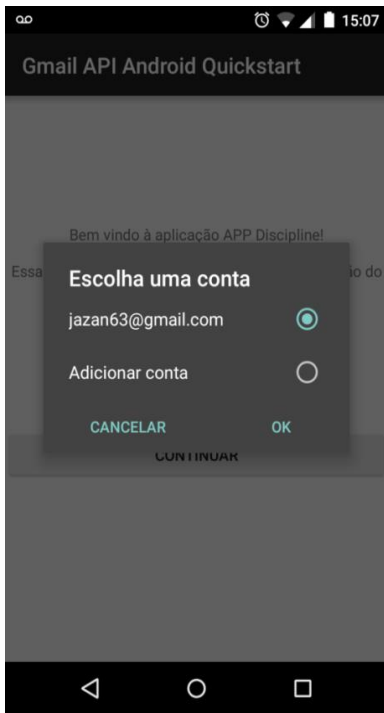


Figura 9 a – Tela de escolha de conta

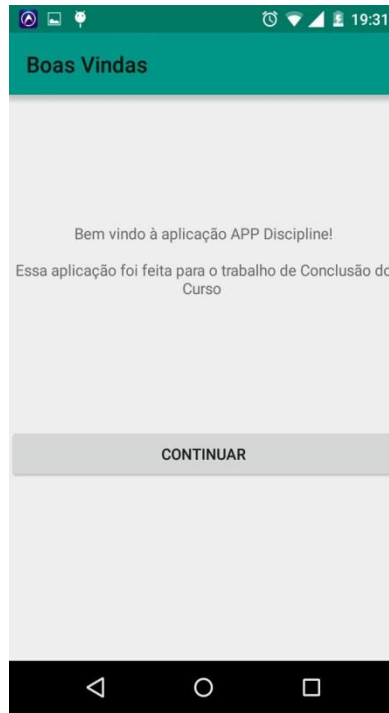


Figura 9 b – Tela de Boas-Vindas

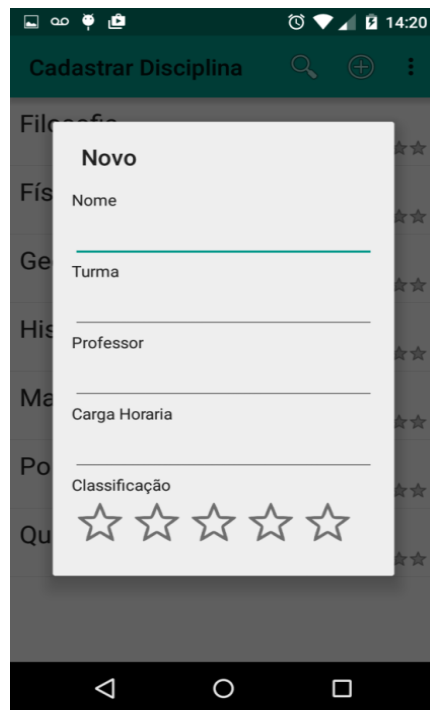


Figura 10 – Tela Cadastro de Disciplina

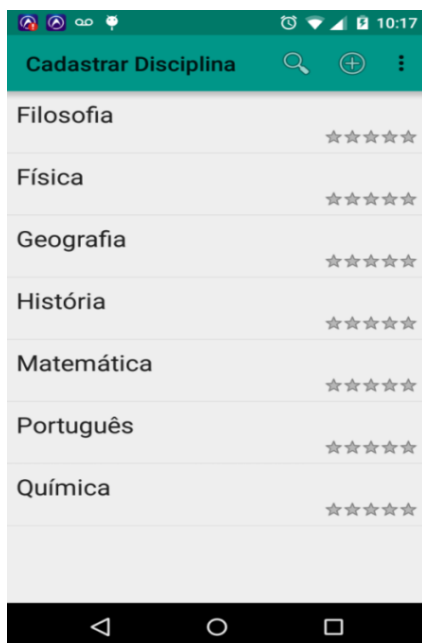


Figura 11 – Disciplinas Cadastradas

A (Figura 11) apresenta uma listagem das disciplinas cadastradas na aplicação, ao pressionar a tela por um tempo de aproximadamente 3 segundos (que equivale a um *click* longo) é exibido uma mensagem “item selecionado” para excluir a disciplina, e dando um *click* curto, é chamada a tela seguinte (Figura 12) que apresenta três botões que podem ser clicados.

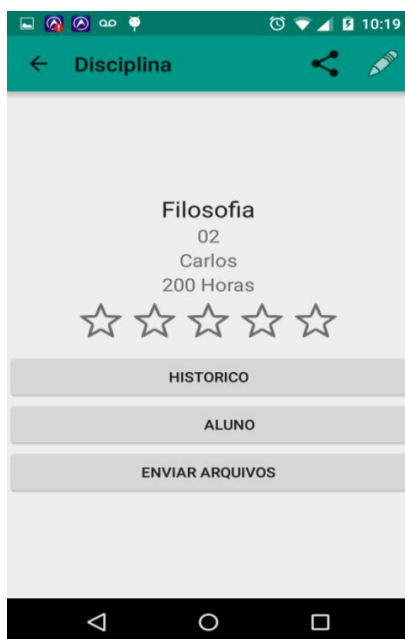


Figura 12 – Tela da disciplina selecionada

Ao pressionar o botão Aluno, como pode ser visto na (Figura 12), é chamada a tela de cadastro do aluno, contendo um ícone que deve ser clicado para cadastrar um novo aluno. Para inseri-lo é preciso preencher os campos apresentados e clicar em gravar. Após a inserção do aluno é apresentado uma listagem com todos os alunos já cadastrados na disciplina.

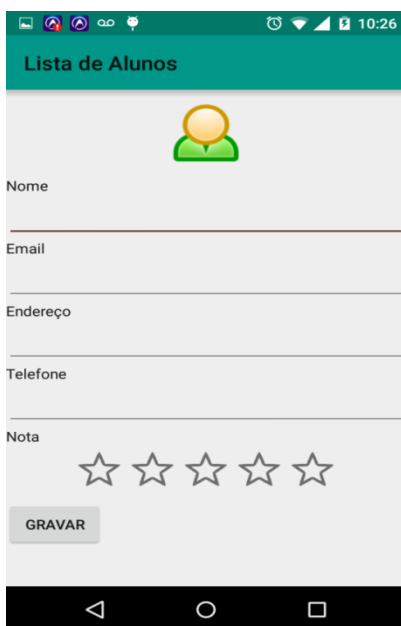


Figura 13 – Tela de Cadastro do Aluno

Na tela que apresenta a listagem alunos cadastrados, o usuário tem a opção de dar um *click* longo no nome do aluno, desse modo é exibido um menu (Figura 14) com as seguintes opções: ligar para o aluno, enviar *sms*, visualizar endereço, enviar *e-mail* e deletar.

Para fazer o envio de arquivos aos alunos cadastrados na respectiva disciplina é necessário que o usuário abra a tela Disciplina (Figura 12) que contém um botão de enviar arquivos. Para concluir o processo usuário terá que dar um *click* curto no botão (Figura 15), selecionar arquivos, cuja função é pesquisar os arquivos na memória externa ou interna do dispositivo. Quando o usuário faz a escolha de um determinado arquivo para ser anexado, é carregado o caminho desse arquivo para um campo de texto, possibilitando ao usuário apagá-lo, caso queira removê-lo antes de ser enviado.

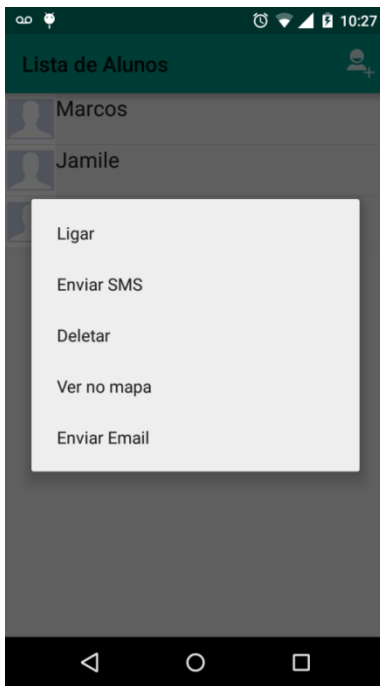


Figura 14 – Lista de alunos



Figura 15 – Tela envio de arquivos

5 RESULTADOS E DISCUSSÕES

Esse capítulo apresenta os problemas encontrados na etapa de desenvolvimento da aplicação, bem como os resultados obtidos na etapa de testes.

5.1 PROBLEMAS ENCONTRADOS

Os resultados apresentados durante os testes da aplicação demonstram algumas limitações em relação à usabilidade da *GMAILAPI*, uma vez que essa API só possibilita a transferência de arquivos de no máximo 35 Mb. Para fazer o envio de um determinado arquivo aos alunos cadastrados na disciplina, é necessário que todos estejam com os *e-mails* cadastrados corretamente na aplicação, caso contrário ocorrerá falha na entrega da mensagem. Cada mensagem enviada deverá conter um assunto, de forma que o destinatário saiba de que se trata essa informação. Cada assunto é salvo no banco de dados da aplicação, em forma de lista como apresenta a (Figura 16). Desse modo o usuário terá que clicar sobre o assunto para ler a mensagem. Esse processo de leitura só acontece se houver uma conexão de *internet* ligada no dispositivo, uma vez que, o aplicativo realiza uma requisição ao *GMAIL* API para receber as respostas do usuário e exibi-las na aplicação.

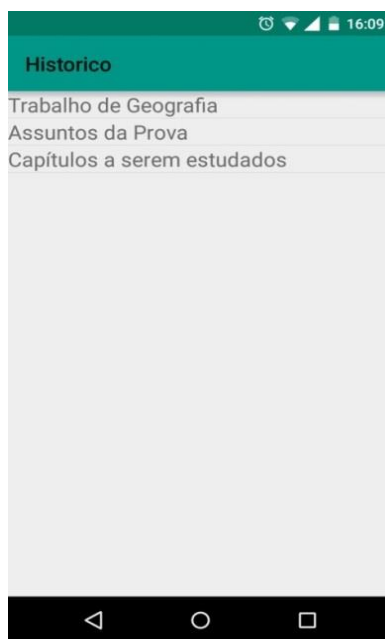


Figura 16: Listagem de mensagens por assunto

Outra característica apresentada por essa API é que ao receber um *e-mail* de resposta de um determinado aluno, é exibida uma notificação de um novo *e-mail*, como mostra a (Figura 17). Uma vez que, para ter acesso ao banco de mensagens do *Gmail* é necessário se autenticar com a conta cadastrada no dispositivo. Isso acarretará numa grande quantidade de

notificações na barra de *status* do aparelho, visto que os aplicativos *e-mail*, nativos do *Android* utilizam essa mesma conta para receber *e-mails*.

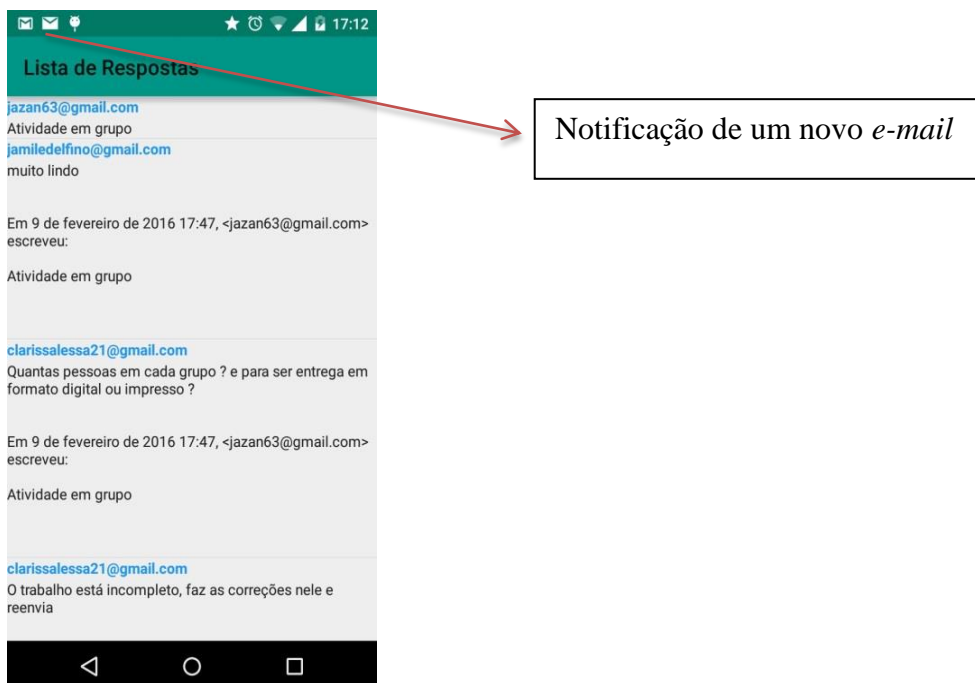


Figura 17: Notificação de *e-mail*

5.2 RESULTADOS ALCANÇADOS

Para obtenção dos resultados fornecidos pela aplicação, foram utilizadas ferramentas capazes de emular e manipular as informações do aplicativo implementado. Durante o processo de desenvolvimento do aplicativo percebeu-se que a cada novo assunto enviado, era criado uma nova mensagem para esse assunto, impossibilitando ao usuário dessa aplicação escrever uma nova mensagem ou anexar um conteúdo na mesma. Em virtude de a *GMAIL* API apresentar algumas limitações para ser utilizada na aplicação, seu uso se mostrou satisfatório para objetivo proposto da aplicação, cuja finalidade era de enviar mensagem com ou sem anexo, aos alunos cadastrados na aplicação.

Levando se em consideração os propósitos educacionais dessa aplicação, pode se afirmar que ela é útil a um professor que demonstra ter poucos conhecimentos das TICs utilizadas atualmente nas instituições de ensino, uma vez que essa aplicação apresenta telas intuitivas, proporcionando ao professor uma interação agradável. Dessa forma o professor irá dispor de um canal de comunicação com os alunos, através de um sistema de *e-mail*, o qual

lhe possibilitará agrupar *e-mails* em um só assunto, recurso esse, oferecido por poucas aplicações *Android*.

Outro ponto a se destacar em relação a essa aplicação é o fato dela possibilitar ao professor enviar diferentes formatos de arquivos para diferentes contas de *e-mails*. Dessa forma ele terá uma limitação no tamanho do arquivo e não na quantidade de arquivos a serem enviados.

6 CONCLUSÕES E TRABALHOS FUTUROS

É possível destacar que ao mesmo tempo em que a tecnologia é vista como objeto de ensino, sua prática em sala de aula não tem se mostrado fácil, devido às dificuldades encontradas, no que se refere ao ambiente escolar. Atualmente existe uma grande expectativa em torno dessas novas tecnologias, acreditando que elas possam contribuir de maneira significativa para um novo conceito de ensino. Entretanto, existem grandes obstáculos a serem superados, uma vez que, a qualidade e quantidade de recursos tecnológicos para alcançar bons resultados ainda são insuficientes, em razão da falta de investimento e infraestrutura.

Pode se perceber que as novas tecnologias podem ter um impacto expressivo sobre o papel dos professores, embora alguns ainda se sintam inseguros e despreparados. O desafio é transformar os estudantes como participantes ativos do processo de aprendizagem e não como receptores passivos de informações, estimulando os professores a dar início à reformulação de suas aulas e a encorajar seus discentes a compartilharem de novas experiências.

Durante a etapa de pesquisa para desenvolver essa aplicação foram identificados inúmeros fatores que impossibilitam a inserção da tecnologia no ambiente escolar, seja por falta de infraestrutura, falta de equipamentos etc. Essa problemática motivou o desenvolvimento de um aplicativo móvel para a plataforma *Android*, em razão da grande quantidade de dispositivos móveis que utilizam este sistema operacional.

O aplicativo desenvolvido consiste em cadastrar disciplinas e alunos vinculados na mesma, permitindo ao professor fazer o envio de mensagens aos alunos através de um sistema de *email*, o possibilitando-o anexar os arquivos aplicados em sala de aula, bem como a visualização das mensagens dos alunos sobre um conteúdo postado pelo professor.

Em relação aos trabalhos futuros, percebe-se a necessidade de melhorias, em relação à usabilidade da aplicação, de modo que seja utilizada em um ambiente real, cujo propósito é oferecer um objeto virtual de aprendizagem ao professor que facilite sua interação com os alunos. Doravante têm-se como propósito, tornar o sistema responsivo para que essa aplicação se adapte a diferentes tipos de tela.

7 REFERÊNCIAS BIBLIOGRÁFICAS

ABLESON, W. F.; SEN, R.; KING, C.; ORTIZ, C. H. **Android em Ação** 3ª Edição. Rio de Janeiro: Elsevier, 2013.

ALMEIDA, P. N. **Educação lúdica: Técnicas e jogos pedagógicos**. 11. ed. São Paulo: Loyola, 2001.

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 2ª Edição. Rio de Janeiro: Elsevier, 2013.

BOOCH, Grady; RUMBAUCH, James; JACOBSON, Ivar. **UML- Guia do usuário**. Rio de Janeiro: Elsevier, 2005.

BORGES, M. A. G. A informação e o conhecimento como insumo ao processo de desenvolvimento. **Revista Ibero-americana de Ciência da Informação (RICI)**, v.1 n.1, p.175-196, jul./dez. 2008. BORGES, M. A. G. A informação e o conhecimento como insumo ao processo de desenvolvimento. *Revista Ibero-americana de Ciência da Informação (RICI)*, v.1 n.1, p.175-196, jul./dez. 2008.

BORGES, P. **Professor é chave para o sucesso no uso de tecnologia na sala de aula**. 2013. Disponível em < <http://ultimosegundo.ig.com.br/educacao/2013-08-22/professor-e-chave-para-o-sucesso-no-uso-de-tecnologia-na-sala-de-aula.html>>. Acesso em 27 de abril de 2015.

CANALTECH. **Android Jelly Bean já detém 10% do Mercado e Gingerbread fica abaixo dos 50%**. Disponível em: < <http://canaltech.com.br/noticia/android/Android-Jelly-Bean-ja-detem-10-do-mercado-e-Gingerbread-fica-abaixo-dos-50/>>. Acesso em 25 de fevereiro de 2016.

CIRIACO, Douglas. **O que é API?**. Disponível em:< <http://www.tecmundo.com.br/programacao/1807-o-que-e-api-.htm>>. Acesso em 08 de janeiro de 2016.

CURVELO, André. et.al. **Linguagens para sistemas embarcados**. Disponível em:< <http://www.embarcados.com.br/editorial-linguagens-para-sistemas-embarcados/>>. Acesso em 25 de fevereiro de 2016.

DA SILVA, M. B. **A Aplicação da Folksonomia em Sistemas de Informação**. In: ENCONTRO BRASILEIRO DE ARQUITETURA DA INFORMAÇÃO – EBAI, 4, 2010, São Paulo. Anais. São Paulo: EBAI, 2010.

DA SILVA, L. F. ; DE OLIVEIRA E. D. ; BOLFE Marcelo. **Mobile learning: Aprendizagem com mobilidade**. Encontro de Ensino, Pesquisa e Extensão da Unioeste, Presidente Prudente, 2013, vol. 5, n. Especial, Jul–Dez, 2013, p. 59-65. Disponível em:< <http://www.unoeste.br/site/enepe/2013/suplementos/area/Exactarum/Computa%C3%A7%C3%A3o/MOBILE%20LEARNING%20APRENDENDIZAGEM%20COM%20MOBILIDADE.pdf>> Acesso em 17 maio de 2015.

DEITEL, P. J.; DEITEL, H. M.; DEITEL, A.; MORGANO, M.. **Android para programadores: Uma abordagem baseada em aplicativos**. 1ª Edição revisada e ampliada. Porto Alegre –RS: Bookman Editora, 2012.

FALBO, Ricardo de Almeida. **Engenharia de Software** Notas de Aula. Disponível em: <<http://www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/NotasDeAula.pdf>>. Acesso em 04 de janeiro de 2016.

GADELHA, Reginaldo. **Desenvolvendo para Android: Arquitetura Android**. Disponível em: <<http://www.tiselvagem.com.br/geral/desenvolvendo-para-android-arquitetura-android/>>. Acesso em: 3 de novembro de 2015.

GALARDO, D. **Getting started whit the Eclipse Platform** Disponível em: <<http://www.ibm.com/developerworks/library/os-ecov/>>. Acesso em 19 de maio de

GLAUBER, Nelson. **Dominando o Android Do Básico ao Avançado**.1ª Edição. São Paulo. Novatec. 2015.

GOMES, Rafael Caveari; FERNANDES, Jean Alves R; FERREIRA, Vinicius Corrêa. **Sistema Operacional Android**. Disponível em:< <http://www.midiacom.uff.br/~natalia/2012-1-sisop/tgrupo1.pdf>>. Acesso em 5 de outubro de 2015.

GREEF, A. C.; FREITAS, M. C. D. **Fluxo Enxuto de Informação: um novo conceito**. Disponível em:<<http://www.portaldeconhecimentos.org.br/index.php/por/content/view/full/15811>>. Acesso em 20 novembro de 2015.

GREEF, A. C. FREITAS, M. do C. D. **Perspectivas em Ciência da Informação**, v.17, n.1, p.37-55, jan./mar. 2012.

GUEDES, Gilleanes T. A. **UML 2: Uma abordagem prática**. 2ª Edição. São Paulo: Novatec, 2011.

GUIMARÃES, E. M.P.; ÉVORA, Y.D. M. **Sistema de informação: instrumento para tomada de decisão no exercício da gerência**. Vi. Inf. , Brasília, v.33, n.1, p. 72-80, jan/abril 2004. Disponível em: <http://www.scielo.br/scielo.php?pid=S0100-19652004000100009&script=sci_arttext> . Acesso em 8 de Janeiro de 2015.

HOFSTAETTER, Andrea. **Objetos virtuais de aprendizagem possibilidades para a educação em artes visuais**. Disponível em :<http://www.ufrgs.br/gearte/pesquisas/pesquisa_andrea01.pdf>. Acesso em 29 de fevereiro de 2016.

INSTITUTO CLARO. **Uso das TICs em sala de aula: desafio para os professores**. Disponível em: <<https://www.institutoclaro.org.br/em-pauta/levar-as-tics-para-a-sala-de-aula-desafios-para-os-professores/>>. Acesso em 20 de maio de 2015.

JARVENPAA, S.; LANG,K. Managing the Paradoxes of Mobile Technology. **Information Systems Management**, 22(4), 7-23, 2005. Disponível em:

<http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1070161> . Acesso em 14 de janeiro de 2016.

JORDÃO, Fabio. **A história dos sistemas operacionais [infográfico]**. Disponível em:<<http://www.tecmundo.com.br/sistema-operacional/2031-a-historia-dos-sistemas-operacionais-ilustracao-.htm>>. Acesso em 4 de fevereiro de 2016.

JUNIOR, P. G. G.; SCHLEMMER, E.. **Aprendizagem com mobilidade (*m-learning*): Novas possibilidades para as práticas pedagógicas e a formação docente?** Congresso Internacional ABED de Educação a Distância, Santos, 2008, p.10, 2008. Disponível em:<<http://www.abed.org.br/congresso2008/tc/5112008112157PM.pdf>>. Acesso em 15 de maio de 2015.

DUARTE, Vânia Maria do Nascimento. O e-mail – Um gênero textual dinâmico. Disponível em: <<http://mundoeducacao.bol.uol.com.br/redacao/o-emailum-genero-textual-dinamico.htm>>. Acesso em 25 de fevereiro de 2016.

LAUDON, Kenneth C; LAUDON, Jane P. **Sistemas de Informação Gerenciais** Administrando a empresa digital. 5 ed. São Paulo: Pearson Prentice Hall, 2004.

LECHETA, R. R. **Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 3ª Edição. São Paulo: Novatec , 2013.

LEMOS, Antônio Carlos. **Plataforma Android**. Minas Gerais, 2011.

LUZZI, L. Android – **Persistência de Dados usando SQLite**.2013. Disponível em:<<http://www.mobiltec.com.br/blog/index.php/android-persistencia-de-dados-usando-sqlite/>>. Acesso em 19 de maio de 2015.

MACHADO, J. L. de A. **Celular na sala de aula: O que fazer?** 2010. Disponível em: <<http://www.planetaeducacao.com.br/portal/artigo.asp?artigo=1621>>. Acesso em 25 de maio de 2015.

MASSAGO, S.; SCHÜTZER, W. **Tutorial de Java**. Disponível em: <<http://www.dm.ufscar.br/profs/waldeck/curso/java/>>. Acesso em 20 de maio de 2015.

MEIRA, Guilherme Tebaldi. **Projeto e desenvolvimento de um aplicativo móvel para acesso ao Portal do Aluno da UFES na plataforma Android**. Disponível em:<<http://www.lprm.inf.ufes.br/sites/default/files/monografia-versaofinal.pdf>> Acesso em 05 de dezembro de 2015.

MONTEIRO, Claudia Guerra. **O papel educativo dos meios de comunicação**. Disponível em:<http://www.ipv.pt/forumedia/3/3_fi3.htm> . Acesso em 25 de fevereiro de 2016.

O'BRIEN, J. A. **Sistemas de Informação e as decisões gerenciais na era da Internet**. 2ª Edição. São Paulo: Saraiva, 2004.

OTSUKA, G. S.; ZANELATO A. P. A.. **O sistema android no universo dos dispositivos móveis**. Disponível

em:<<http://intertemas.unitoledo.br/revista/index.php/ETIC/article/viewFile/3759/3520>>. Acesso em 15 de maio de 2015.

PASSOS, Thiago de Sousa. **Android, Arquitetura e Desenvolvimento**. Disponível em:<<https://android-development.googlecode.com/files/Thiago%20Souza%20-%20Android,%20Arquitetura%20e%20Desenvolvimento.pdf>> . Acesso em 9 de dezembro de 2015.

PEPATO, Ricardo. **Google Lança API Beta para o Gmail**. Disponível em: <<http://www.infoq.com/br/news/2014/07/google-gmail-api>>. Acesso em 12 de janeiro de 2016.

PEREIRA, Lúcio Camilo Oliva; SILVA, Michel Lourenço da. **Android para desenvolvedores**. Rio de Janeiro: Brasport, 2009.

PEREIRA, L. R.; SCHÜTZER, V. R. N.; SCHÜTZER, E. ; DALFOVO, O. Seminário Nacional de Educação Profissional e tecnológica : **O uso da tecnologia na educação, priorizando a tecnologia móvel**, 4.,2012. Belo Horizonte: Ministério da Educação, 2012. 17p. Disponível em: <http://www.senept.cefetmg.br/galerias/Anais_2012/GT-02/GT02-014.pdf>. Acesso em 10 de maio de 2015.

PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 7ª Edição. Porto Alegre: AMGH, 2011.

ROCHA, Leonardo. **Android Studio: ferramenta de criação de apps da Google ganha versão 1.0**. Disponível em <<http://www.tecmundo.com.br/android/69111-android-studio-ferramenta-criacao-apps-google-ganha-versao-1-0.htm>> Acesso em 12 de dezembro de 2015.

SABOIA, Juliana; VARGAS, Patrícia Leal de; VIVA, Marco Aurélio de Andrade. O uso dos dispositivos móveis no processo de ensino e aprendizagem no meio virtual. **Revista Cesuca Virtual: Conhecimento sem fronteira**. Cachoeirinha- RS, v.1, n. 1, jul/2013. Disponível em: <<http://ojs.cesuca.edu.br/index.php/cesucavirtual/article/viewFile/424/209>>. Acesso em 26 de fevereiro de 2016.

SIMONE, Sergio de. **Google Lança API Beta para o Gmail**. Disponível em: <<http://www.infoq.com/br/news/2014/07/google-gmail-api>>. Acesso em 04 de janeiro de 2016.

SOMMERVILLE, Ian. **Engenharia de Software**. 8ª Edição. São Paulo: Pearson Addison - Wesley, 2007.

SQUIRRA, S. Os desafios do ensino no futuro. Disponível em:<http://www.ubaweb.com/revista/g_mascara.php?grc=31184>. Acesso em 2 de fevereiro de 2016.

UZEJKA, Guilherme de Moraes. **Modelando um cliente voip inteligente para a plataforma Android**. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/31033/000782093.pdf?sequence=1>>. Acesso em 8 dezembro de 2015.

VASCONCELOS, Alexandre Marcos Lins. et al. **Introdução à Engenharia de Software e à qualidade de software**. Disponível em: <
http://www.cin.ufpe.br/~if720/downloads/Mod.01.MPS_Engenharia%26QualidadeSoftware_V.28.09.06.pdf> . Acesso em 04 de janeiro de 2016.



TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA
"JOSÉ ALBANO DE MACEDO"

Identificação do Tipo de Documento

- () Tese
() Dissertação
 Monografia
() Artigo

Eu, Joson Nogueira Delfino,
autorizo com base na Lei Federal nº 9.610 de 19 de Fevereiro de 1998 e na Lei nº 10.973 de
02 de dezembro de 2004, a biblioteca da Universidade Federal do Piauí a divulgar,
gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação
Aplicativos Móvel para Auxílio Professor no Uso
De Objetos Virtuais De Aprendizagem
de minha autoria, em formato PDF, para fins de leitura e/ou impressão, pela internet a título
de divulgação da produção científica gerada pela Universidade.

Picos-PI 11 de Mais de 2016.

Joson Nogueira Delfino
Assinatura