



INSTITUTO FEDERAL DE CIÊNCIA E TECNOLOGIA DE
PERNAMBUCO - CAMPUS RECIFE

PROCESSOS E CONTROLES ELETRO-ELETRÔNICOS
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

YURI BEZERRA RODRIGUES
PEDRO IGOR BARROSO JATOBA

**EXPANDINDO O ALCANCE DA EAD: DESENVOLVIMENTO DE WEBSERVICE E
APLICAÇÃO MOBILE PARA A PLATAFORMA MOODLE-IFPE**

Recife
2019

**EXPANDINDO O ALCANCE DA EAD: DESENVOLVIMENTO DE WEBSERVICE E
APLICAÇÃO MOBILE PARA A PLATAFORMA MOODLE-IFPE**

Trabalho de conclusão de curso apresentado ao curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco, como requisito parcial para a obtenção do título de tecnólogo em Tecnologia em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Ramide Augusto Sales Dantas

Coorientador: Prof. Thiago Affonso de Melo Novais Viana
(Open University, Reino Unido)

Recife

2019

R696E

Rodrigues, Yuri Bezerra

Expandindo o alcance da ead: desenvolvimento de webservice e aplicação mobile para a plataforma moodle-ifpe/ Yuri Bezerra Rodrigues; Pedro Igor Barroso Jatobá.. – Recife: IFPE, 2021.

51 p.

Trabalho de conclusão de curso (Tecnólogo) – Instituto Federal de Pernambuco, Curso Tecnólogo em Análise e Desenvolvimento de Sistemas, 2021.

1. Plataforma Moodle – EaD. 2. Tecnologia REST. 3. Moodle Droid – aplicativo móvel. 4. Moodle Vox – acessibilidade. 5. Tecnologia para acessibilidade I. Título. II. Rodrigues, Yuri Bezerra III. Jatobá, Pedro Igor Barroso

Trabalho de Conclusão de Curso apresentado pelos estudantes **Yuri Bezerra Rodrigues** e **Pedro Igor Jatobá** à coordenação de Análise e Desenvolvimento de Sistemas, do Instituto Federal de Pernambuco, sob o título de *“EXPANDIDO O ALCANCE DA EAD: DESENVOLVIMENTO DE WEBSERVICE E APLICAÇÃO MOBILE PARA A PLATAFORMA MOODLE-IFPE”*, orientado pelo Prof. **Ramide Augusto Sales Dantas** e aprovado pela banca examinadora formada pelos professores:

Recife, 27 de dezembro de 2019

Prof(a). Dr. Ramide Augusto Sales Dantas
CTADS/DASE/IFPE

Prof(a). Dr. Eduardo de Melo Vasconcelos
CTADS/DASE/IFPE

Prof. Ms. Danilo Farias Soares da Silva
FACOL/UFRPE

Aluno: Pedro Igor Jatobá

Aluno: Yuri Bezerra Rodrigues

AGRADECIMENTOS

Dedico aos meus pais e a Deus, de onde vieram as melhores cobranças e incentivos para o término do curso. À minha irmã, Yane. Aos amigos que me aguentaram durante o tempo de curso, aos irmãos da vida: Barla, Pedro, Salvatore, Paulo, Alves, Larissa, Tainara e Carol. Especialmente Isabela, que soube nos momentos mais complicados me acalmar e dar forças. Aos professores que souberam aguentar meu jeito em sala de aula. Aos companheiros de trabalho, Julio, Halcyon, Carmen, Robson, Mariana, Rodrigo, Felipe, Aldo, Cintia, Juliano, Bruno e Marivaldo.

Yuri Bezerra Rodrigues

Dedico aos meus familiares que estavam comigo durante todo esse tempo e que se mantiveram dispostos a me incentivar e dedicar um pouco do tempo deles ao meu crescimento. Quero agradecer em especial ao meu pai que me inscreveu na área que viria a ser o meu futuro e que me traria grandes conquistas e desafios, ao meus colegas de curso que viriam a ser amigos pra vida, a minha namorada Mariana que sempre esteve me dando apoio e me fazendo perceber que eu sempre posso ser melhor do que eu fui ontem, e por fim, aos mestres que me apresentaram esse mundo que é a engenharia de software o qual não consigo me imaginar sem.

Pedro Igor Barroso Jatobá

RESUMO

O crescente uso de *smartphones* e acesso à Internet móvel no Brasil cria novas oportunidades para o uso de Educação a Distância (EaD). Com o objetivo de expandir o alcance da plataforma *Moodle* utilizada no Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco (IFPE), foi implementada uma coleção de serviços usando tecnologia REST (*Web Services*) para acesso a seus dados. Esses serviços permitiram o desenvolvimento de um aplicativo móvel de prova de conceito Moodle Droid, além de um aplicativo destinado a deficientes visuais, o Moodle Vox. Este trabalho apresenta a criação dos serviços e arquitetura REST do *Moodle* IFPE, o qual possui uma *Web API* própria com arquitetura em camadas e transporte de dados do tipo *JSON*.

Palavras-chave: Moodle, REST, JSON, WebAPI, EaD, m-learning, e-learning.

ABSTRACT

The growing use of smartphones and mobile Internet access in Brazil creates new opportunities for the use of Distance Education (DE). In order to expand the reach of the Moodle platform used at the Federal Institute of Education, Science and Technology of Pernambuco (IFPE), a collection of services using REST technology (Web Services) was implemented to access its data. These services enabled the development of a proof of concept mobile app (MoodleDroid) and another app for the visually impaired (MoodleVox). This work presents the development of the Moodle IFPE REST architecture and services, which has its own Web API with layered architecture and JSON data transport.

Keywords: Moodle, REST, JSON, WebAPI, EaD, m-learning, e-learning.

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Motivação	10
1.2	Objetivos	11
1.3	Estrutura do Trabalho	11
2	METODOLOGIA DE DESENVOLVIMENTO	13
3	REFERENCIAL TEÓRICO	15
3.1	E-Learning	15
3.2	M-Learning	16
3.3	SOA – Arquitetura Orientada a Serviços	16
3.4	Serviços Web	17
3.4.1	<i>Simple Object Access Protocol (SOAP)</i>	17
3.4.2	<i>Representational State Transfer (REST)</i>	18
3.5	Projeto de Arquitetura de Software	20
3.5.1	<i>Padrões de Arquitetura</i>	20
3.6	Json	21
3.7	Jersey	21
3.8	Android	21
4	ARQUITETURA DO SERVIÇO WEB MOODLE IFPE	23
4.1	Camadas do Webservice Moodle IFPE	23
4.1.1	<i>Configuration</i>	23
4.1.2	<i>Controller</i>	24
4.1.3	<i>DAO</i>	24
4.1.4	<i>Fachada</i>	25
4.1.5	<i>Factory</i>	26
4.1.6	<i>Model</i>	26
4.1.7	<i>Recursos</i>	27
4.1.8	<i>Util</i>	28
5	SERVIÇOS MOODLE IFPE	29
5.1	Login (POST, / LOGIN)	29
5.2	Retorna Notícias (POST, /RETORNANOTICIAS)	29
5.3	Retorna Fóruns Cursos (POST, /RETORNAFORUNSCURSOS)	29
5.4	Retorna Tópicos Fóruns (POST, /RETORNATOPICOSFORUNS)	29

5.5	Meus Contatos (POST, /MEUSCONTATOS)	29
5.6	Enviar Mensagens (POST, /ENVIARMENSAGENS)	30
5.7	Acrescentar Contatos (POST, /ACRESCENTACONTATOS)	30
5.8	Excluir Contatos (POST, /EXCLUIRCONTATOS)	30
5.9	Bloquear/Desbloquear Contatos (POST, /BLOQUEARDESBLOQUEARCONTATOS)	30
6	APLICATIVO MOODLE VOX	31
6.1	Moodle Droid	31
6.2	Características Básicas	32
6.3	Menu de Interação	33
<i>6.3.1</i>	<i>Módulo de Mensagens</i>	33
<i>6.3.2</i>	<i>Módulo de Disciplinas</i>	35
7	CONCLUSÕES	39
7.1	Trabalhos Futuros	40
	REFERÊNCIAS	
	ANEXOS	

1 INTRODUÇÃO

Com a introdução de novas tecnologias ao ensino no Brasil, a Educação a Distância (EaD) surgiu como uma nova forma de ensino-aprendizagem. Segundo a Presidência da República, Decreto-Lei Nº 9.057: “educação a distância a modalidade educacional na qual a mediação didático-pedagógica nos processos de ensino e aprendizagem ocorra com a utilização de meios e tecnologias de informação e comunicação, com pessoal qualificado, com políticas de acesso, com acompanhamento e avaliação compatíveis, entre outros, e desenvolva atividades educativas por estudantes e profissionais da educação que estejam em lugares e tempos diversos”.

Conforme Relatório Analítico da Aprendizagem a Distância no Brasil, CENSO EAD.BR 2016, os Ambientes Virtuais de Aprendizagem (AVA) são o recurso mais utilizado na distribuição de conteúdo em cursos regulamentados e cursos livres. Segundo João Vianney no CENSO EAD.BR 2016: “O perfil dos alunos que estudam a distância no Brasil é um indicador seguro do caráter inclusivo da modalidade. “

No Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco (IFPE), a educação a distância (EAD) utiliza como AVA o *Modular Object Oriented Distance Learning* (Moodle) (MOODLE, 2017), sistema idealizado por Martin Dougiamas de plataforma livre que pode ser parametrizada de acordo com as necessidades de cada instituição, o que permitiu ao IFPE fazer customizações que atendessem as suas necessidades. O Moodle possui uma concepção e desenvolvimento orientados a pedagogia social construtiva.

Das novas tecnologias ligadas à EAD, por exemplo, vídeo aulas e bibliotecas virtuais, destaca-se o *Mobile Learning (m-learning)*, que utiliza os dispositivos móveis em combinação com outras tecnologias da informação para disseminar conhecimento através das redes móveis e conexões sem fios. Apresenta-se assim para o IFPE, uma nova possibilidade de entrega de conteúdo aos alunos da modalidade de ensino a distância, diminuindo o deslocamento e tornando flexível os horários de estudo.

1.1 Motivação

A busca pela melhoria dos serviços prestados a comunidade acadêmica do ensino a distância relacionado aos serviços de educação, nos faz refletir o patamar que a EAD encontra-se no Instituto Federal de Pernambuco. Utilizando-se da aplicação *web* existente no Instituto (Moodle-IFPE), notou-se uma dificuldade em utilizar a plataforma em aparelhos móveis (celulares, tablets, etc.) com acesso à Internet, por exemplo, por não ser capaz de se adaptar a diferentes modelos e tamanhos de tela.

Com o crescente uso de celulares inteligentes e mais robustos, a inovação tecnológica por meio de ambientes virtuais de aprendizagem no Instituto Federal, fez com que criasse uma necessidade de serviços voltados para os dispositivos móveis. Viu-se então a oportunidade de criar um sistema computacional que facilitasse a extensão da plataforma de ensino para a EAD do IFPE para dispositivos móveis inteligentes.

Considerando que o ambiente de aprendizagem do IFPE foi customizado para as necessidades do Instituto, os serviços oferecidos pela API padrão do Moodle para as aplicações móveis tornou-se inadequados. Diante disso, foi identificada e necessidade de desenvolver uma Web API própria para o Moodle-IFPE, assim como um aplicativo móvel utilizando a plataforma Android para fins de teste e validação.

As APIs Web baseadas em HTTP são utilizadas frequentemente para a implementação dos sistemas distribuídos norteados pelo padrão arquitetural REST. Seu diferencial em relação ao conceito tradicional de sistemas distribuídos está em encapsular as informações por meio de recursos. O uso das APIs REST possibilita que diversas plataformas troquem informações com menor acoplamento se comparada ao compartilhamento entre bases de dados.

1.2 Objetivos

O objetivo geral é facilitar a adoção de tecnologias *m-learning* no IFPE por meio do desenvolvimento de uma *Web API* utilizando o padrão arquitetural REST que possa servir de modelo para o consumo de serviços do ambiente virtual de aprendizagem do instituto, auxiliando o ensino à distância da instituição. A ideia principal do sistema é servir como base de comunicação entre o *Moodle* e o aplicativo *mobile* a ser desenvolvido. Os objetivos específicos do trabalho foram:

- Criação de uma API Web que dê acesso aos dados do Moodle-IFPE;
- Criação de um Web Site descrevendo o uso da biblioteca e sua API;
- Criação de um aplicativo Android que atenda à comunidade do IFPE e valide o serviço criado.

1.3 Estrutura do Trabalho

No primeiro capítulo deste trabalho foi apresentado a motivação, justificativa, objetivos gerais e específicos. O segundo capítulo descreve a metodologia do presente trabalho. Nos capítulos 3 e 4 descreve-se as bases tecnológicas do trabalho, detalhando as tecnologias adotadas para a construção do sistema, arquitetura e a justificativa do emprego de cada uma delas no presente trabalho. O quinto capítulo apresenta-se os serviços desenvolvidos baseados

em tecnologias descritas no trabalho. No capítulo 6 é apresentado o modelo de aplicativo a ensino à distância para deficientes visuais e a metodologia adotada para a consolidação da aplicação e validação de sua proposta junto à comunidade acadêmica, que descreve os casos de uso do sistema, modelo Entidade Relacionamento (descreve a estrutura do banco de dados da aplicação) juntamente com a arquitetura do sistema. O capítulo 7 apresenta as conclusões do trabalho além de trabalhos futuros.

2 METODOLOGIA DE DESENVOLVIMENTO

Inicialmente foi realizado um levantamento bibliográfico sobre técnicas e ferramentas que utilizam APIs do tipo REST na modalidade de educação a distância. A partir desse estudo pode-se compreender as principais ferramentas já existentes e quais as principais técnicas utilizadas para atingir esse objetivo.

Em seguida, foram elaboradas e conduzidas aplicações de questionários e entrevistas com profissionais da área pedagógica com especialidade em EaD e com professores e tutores da EaD do IFPE. Tais ações possibilitaram o diagnóstico dos itens e requisitos das necessidades para o projeto no âmbito da EaD do IFPE e mapear as mesmas para a criação de uma ferramenta com os recursos desejados.

A partir dos dados obtidos nos questionários e entrevistas um planejamento de projeto e uma análise e validação de requisitos foram capazes de serem realizados visando as técnicas descritas na Engenharia de Software. Utilizando a união de métodos coerentes e coordenados buscando um processo de desenvolvimento de sistemas com foco na qualidade.

Visando a clareza dos objetivos, foram elaborados diagramas de caso de uso, diagramas de classe, diagramas de sequência e diagramas de atividades no intuito de documentar a ferramenta proposta.

Para o desenvolvimento da API REST foi utilizada a linguagem Java¹, devido à facilidade e seu alto grau de uso na atualidade, bem como ser a linguagem mais estudada nas instituições de ensino e em particular no IFPE. A implementação da arquitetura foi feita dentro da ferramenta Eclipse² utilizando um modelo de versionamento que garantisse o gerenciamento de diferentes versões de desenvolvimento do código fonte da arquitetura, dentre as especificações de software livre destaca-se o Git³, ferramenta projetada e desenvolvida por Linus Torvalds para o desenvolvimento do Kernel Linux, para tal foi criado um repositório dentro da plataforma GitHub⁴ visando um maior compartilhamento do desenvolvimento da API.

Foi seguida uma abordagem de desenvolvimento baseada em etapas baseada nas propostas de Richardson, Amundsen e Ruby (2013) e *Domain Application Protocol*

¹ Acessível em: https://www.java.com/pt_BR/about/

² Acessível em: <https://help.eclipse.org/2019-12/index.jsp>

³ Acessível em: <https://git-scm.com/>

⁴ Acessível em: <https://github.com/>

(ROBISON,2011). Essa abordagem auxilia na organização e descrição semântica do conteúdo proposto de Web APIs REST. As etapas de desenvolvimento são:

- Descrição Semântica de Dados: o objetivo nesta etapa é proporcionar uma interação autônoma com mecanismos corretos para que os consumidores interpretem de forma adequada as informações. Tal descrição visa o compartilhamento entre a Web API e seus agentes semânticos, possuindo um acoplamento conceitual.
- Utilização do Design de Representações Atômicas: define como objetivo principal um modelo de interação entre a API Web e seus clientes baseado em recursos. Entretanto, cada recurso recebe um endereço URI com seus conjuntos de operações.
- Identificação de Recursos: o objetivo principal dessa etapa é manter uma separação da camada de integração do restante das camadas de sistema, permitindo uma maior liberdade em modelagem de negócio, respeitando a responsabilidade e não atribuição de atividades fora do escopo do recurso

Com a definição das etapas e desenvolvimento da API, juntamente com as necessidades levantadas com os questionários, testes unitários dos serviços criados foram feitos utilizando a própria ferramenta Eclipse e o incremento da aplicação SOAPUI⁵ para validação da semântica JSON⁶ utilizada na arquitetura, tal ferramenta permite gerar testes de cargas em web services e a invocação dos serviços criados.

A partir da validação dos serviços, a criação do aplicativo do MoodleDroid IFPE foi iniciada feito para aparelhos com sistema operacional Android⁷ que visa adaptar a navegação do site de EaD do Instituto federal para o ambiente mobile se utilizando dos serviços da API. A partir da criação do MoodleDroid foi iniciado o desenvolvimento do aplicativo de interação por voz MoodleVox IFPE, que é mais tratado neste trabalho. O MoodleVox tem como sua principal função ser utilizado por portadores de deficiência visual trazendo todas as funcionalidades do aplicativo do MoodleDroid IFPE criando assim um ambiente que não restrinja nenhum aluno das funcionalidades de EaD no Instituto.

⁵ Acessível em: <https://www.soapui.org/open-source.html>

⁶ Acessível em: <https://www.json.org/json-en.html>

⁷ Acessível em: <https://developer.android.com/about>

3 REFERENCIAL TEÓRICO

Neste capítulo são apresentados conceitos, teorias e modelos que venham sustentar argumentações propostas no trabalho, além das tecnologias, frameworks e padrões de desenvolvimento utilizados durante o projeto.

3.1 E-Learning

Ao analisarmos a tradução literal do termo *e-learning*, um acrônimo de *eletronic learning*, temos o termo aprendizado eletrônico. Segundo (GOMES, 2005): “O conceito de *e-learning*, pode abarcar situações de apoio, tutorial ao ensino presencial, em que o professor-formador-tutor disponibiliza materiais, sugere recursos e interage *on-line* com os alunos (esclarecendo dúvidas, fomentando debates, estimulando a colaboração *on-line*), não constituindo este cenário um modelo de educação a distância.”

Para que o professor e/ou colaborador possam gerenciar os conteúdos dos cursos, é necessário ter uma plataforma conhecida como LMS ou Sistema de Gestão de Aprendizagem, tradução do termo em inglês *Learning Management System*. Com esse sistema, pode-se ter o controle de todo o processo de aprendizagem de seus participantes.

As plataformas de *e-learning* pode ser dividida em dois grandes grupos:

- Assíncrono: onde existe uma gravação prévia das informações que acabam sendo armazenadas em servidores, garantindo uma maior liberdade aos usuários com flexibilidade de horário.
- Síncrono: as informações são repassadas em tempo real, ou seja, uma sala virtual com horários predefinidos para apresentação do conteúdo pelo professor/tutor.

Atualmente, existem várias plataformas dentro dos sistemas LMS. Segundo a fundação Capterra (CAPTERRA, 2017), que apresenta análises de usuários validadas e pesquisas independentes em várias categorias de sistemas de negócios, os 3 primeiros colocados, levando em consideração clientes, usuários e mídias sociais são:

- Edmodo⁸: com maior número de clientes, é um sistema gratuito baseado na *web* que fornece compartilhamento de conteúdo, *feedback* em tempo real, ferramentas de comunicação e gerenciamento de sala de aula para o setor educacional.
- Moodle⁹: possuindo o maior número de usuários, é um sistema de gerenciamento gratuito de aprendizado *on-line* que fornece aos educadores uma solução de código

⁸ Acessível em: <https://go.edmodo.com/about/>

⁹ Acessível em: https://docs.moodle.org/38/en/About_Moodle

aberto para *e-learning* escalável, personalizável e segura, com a maior seleção de atividades disponíveis.

- Learning¹⁰ (SuccessFactors): sendo a terceira maior plataforma em usuários, combina aprendizado formal, informal, social e ampliado com recursos inovadores de gerenciamento de conteúdo, análise e mobilidade proporcionando um aprendizado que transforma os negócios e leva a resultados verificáveis, possuindo um foco maior na área profissional.

3.2 M-Learning

Desde o advento da *internet*, e a popularização de computadores pessoais vive-se uma era digital, onde as tecnologias surgem em ritmo quase frenético, e verdadeiramente estão transformando as relações que os indivíduos têm, seja com dispositivos eletrônicos, seja entre eles mesmos. (KENSKI, 2012; VEEN; VRAKING, 2009).

Com o a expansão da *internet* e a disseminação das tecnologias de informação e das comunicações de redes informatizadas, ampliou-se o horizonte para o desenvolvimento dos sistemas *e-learning*.

Nos últimos anos, os dispositivos móveis, ganharam espaço na sociedade por serem fáceis de transportar e, com auxílio das Tecnologias de Informação e Comunicação Móveis e Sem Fio (TIMS), poderem estar conectados à *internet* por meio de redes de celular ou dispositivos *wireless*. Dentro de suas casas os pais possuem *smartphones*, *tablets* e *notebooks* e não hesitam em incentivar seus filhos a utilizar tais aparelhos. (PRENSKY, 2001; VEEN; VRAKING, 2009).

Diante da popularização dos dispositivos móveis, um novo modelo de aprendizagem que une as TIMS e este tipo de equipamento vem sendo apresentado nos últimos anos: o *mobile learning* (*m-learning*), ou aprendizagem móvel.

O *m-learning* pode então ser definido como sendo a aprendizagem apoiada pelo uso de TIMS ou a aplicação de TIMS na educação (DESMOND, 2002; TRIFONOVA, 2003; FAGERBERG, REKKEDAL e RUSSELL, 2002).

3.3 SOA – Arquitetura Orientada a Serviços

O termo SOA foi usado pela primeira vez em 1996, por Roy Schulte e Yeffim V. Natiz da Gartner, que definiram como um estilo de computação multicamada que ajuda as

¹⁰ Acessível em: https://www.successfactors.com/en_gb/solutions/learning-development/learning.html

organizações a compartilhar a lógica e dados entre vários aplicativos e modos de uso. (ARNON, 2012)

SOA tem o propósito de atender os requisitos de baixo acoplamento, desenvolvimento baseado em padrões, e computação distribuída independente de protocolo. (PAPAZOGLU et al., 2007)

Arnon, em seu livro *SOA Patterns*, define a arquitetura orientada a serviços (SOA) como um estilo arquitetônico para a construção de sistemas baseados em interações de baixo acoplamento, granulação grossa e componentes autônomos chamados serviços.

Para Josuttis, SOA não é uma arquitetura concreta, mas sim um paradigma que nos leva a uma arquitetura concreta. SOA é uma abordagem, um modo de pensar, um sistema de valores que leva a certas decisões concretas ao projetar uma arquitetura de software. (JOSUTTIS, 2007)

3.4 Serviços Web

Segundo Josuttis (2007), a forma mais tradicional de implementação é através do padrão SOAP (*Simple Object Access Protocol*), entretanto é crescente a adoção de implementações que seguem os princípios arquiteturais REST (*REpresentational State Transfer*).

Um serviço *Web* é um componente de software armazenado em um computador que pode ser acessado por um aplicativo (ou outro componente de software) em outro computador por uma rede. Serviços *Web* se comunicam utilizando tecnologias como XML¹¹, JSON¹² e HTTP¹³. (DEITEL, 2011)

Nesta seção será apresentado conceitos *Web Services*, com destaque maior para a arquitetura REST.

3.4.1 *Simple Object Access Protocol (SOAP)*

O *Simple Object Access Protocol (SOAP)* é um protocolo independente de plataforma que utiliza a XML para fazer chamadas de procedimento remoto, geralmente sobre o HTTP. (DEITEL, 2011)

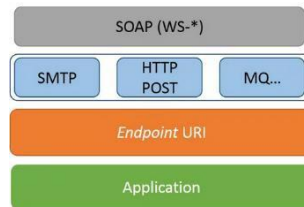
O padrão SOAP, utiliza envelopes de dados enviados do cliente para um Web Service. Os envelopes de dados são parametrizados dentro de um documento XML que possui internamente o conteúdo da mensagem, podendo ainda conter alguns elementos opcionais referentes à infraestrutura.

¹¹ Acessível em: <https://www.ibm.com/developerworks/xml/tutorials/xmlintro/xmlintro.html>

¹² Acessível em: <https://www.json.org/json-en.html>

¹³ Acessível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>

Figura 1- Camadas do Protocolo SOAP

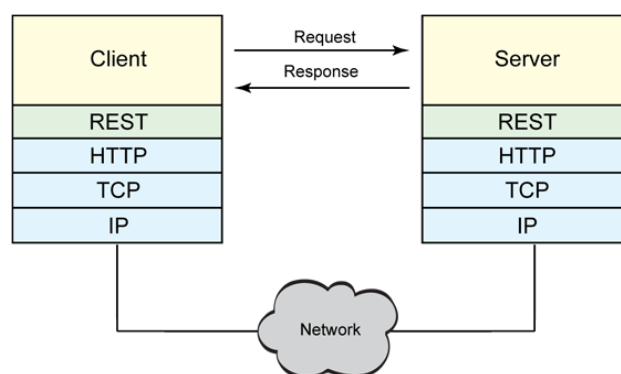


Fonte: Página da Web¹⁴

3.4.2 Representational State Transfer (REST)

O *Representational State Transfer* (REST) refere-se a um estilo arquitetônico de implementar serviços *Web*. Segundo FIELDING (2000), REST é uma coleção de princípios e restrições arquiteturais para o desenvolvimento de aplicações distribuídas na *Web*. Embora o próprio REST não seja um padrão, serviços *Web* ditos RESTful são implementados utilizando padrões *Web*. O REST também não está limitado a retornar dados no formato XML. Ele pode utilizar vários formatos, como XML, JSON, HTML, texto sem formatação e arquivos de mídia. (DEITEL, 2011)

Figura 2 - Camadas do Protocolo REST



Fonte: Página da IBM¹⁵

¹⁴ Disponível em: <https://www.thistechnologylife.com/soap-vs-rest/>

¹⁵ Disponível em: <https://www.ibm.com/developerworks/br/library/os-understand-rest-ruby/index.html>

O estilo arquitetural REST adota o princípio *Hypermedia as the Engine of Application State* (HATEOAS). Este princípio define que as mudanças de estado da aplicação devem ser guiadas através de controles hipermídia. (FIELDING, 2000). A arquitetura REST tem como um de seus pilares estabelecer um meio uniforme de comunicação entre cliente e o servidor, tornando-se mais simples e desacoplada. A forma mais usual de manter essa coerência de comunicação é utilizando as semânticas corretas dos métodos de acesso (ou verbos) do protocolo HTTP. Dentro as semânticas existentes, destacamos:

- PUT: alterar um recurso
- DELETE: remover um recurso
- POST: criar um recurso
- GET: obter um recurso

Ao utilizar os verbos HTTP, diferentes URIs e diferentes formatos de retorno, além de seguir rigorosamente as regras da arquitetura REST possuindo um certo nível de maturidade e coesão do Modelo de Maturidade de Richardson, pode-se denominar essa arquitetura como uma API RESTful. O Modelo de Maturidade de Richardson define quatro níveis:

- Nível 0: utilizar o HTTP como sistema de transporte para interações remotas, mas sem a utilização de qualquer um dos mecanismos da *Web*. Dessa forma, deve-se usar o HTTP como um mecanismo de encapsulamento.
- Nível 1 (Recursos): utilizar recursos, em vez de fazer todos os pedidos a um *endpoint*.
- Nível 2 (Verbos HTTP): utilizar os verbos HTTP na forma como foram concebidos e da forma como são utilizados no protocolo HTTP.
- Nível 3 (HATEOAS): o controle de hipermídia informa o que podemos fazer juntamente com a URI do recurso que irá ser utilizado.

Segundo Arnon Rotem-Gal-Oz no seu livro SOA Patterns, as três maiores vantagens de utilizar REST são (ARNON, 2012):

- Relativa facilidade de integração, uma boa API RESTful é encontrável a partir de sua URI inicial.
- Escalabilidade, ou seja, comunicação sem estados e repositórios replicados.
- Padrões difundidos: HTTP é o protocolo da web; emitir JSON ou ATOMPub significa que é muito mais fácil achar uma biblioteca que pode conectar em você em qualquer linguagem ou plataforma.

Entretanto, aplicações que utilizam esse modelo arquitetural do REST, sofrem com as limitações das linguagens de programação, tipicamente não orientadas a recursos, e portanto o

código que vai mapear as URIs tende a ser desorganizado. Por outro lado, é relativamente difícil fazer uma API REST orientada a hiper-texto. (ARNON, 2012)

Arnon ressalta: uma interface uniforme com 2 verbos é muito pequena para ser realmente útil. Para Arnon, implementações REST sobre HTTP estão limitadas ao uso de dois verbos GET e POST.

3.5 Projeto de Arquitetura de Software

O projeto da arquitetura represa a estrutura de dados e os componentes de programas necessários para construir um sistema computacional (PRESSMAN, 2011). Ainda segundo Pressman, a arquitetura de software serve como uma representação do sistema que permite análise e atendimento dos requisitos solicitados.

3.5.1 Padrões de Arquitetura

Pressman define as arquiteturas em um pequeno grupo de cinco estilos: arquiteturas centralizadas em dados, arquiteturas de fluxo de dados, arquiteturas de chamadas e retornos, arquiteturas orientadas a objetos e arquiteturas em camadas. (PRESSMAN, 2011)

- **Arquiteturas Centralizada em Dados:** um repositório de dados reside no centro dessa arquitetura e é em geral acessado por outros componentes que atualizam, acrescentam, eliminam ou de alguma outra maneira modificam dados contidos no repositório.
- **Arquiteturas de Fluxo de Dados:** essa arquitetura se aplica quando dados de entrada devem ser transformados por meio de uma série de componentes computacionais ou de manipulação em dados de saída.
- **Arquiteturas de Chamadas e Retornos:** permite-nos obter uma estrutura de programa relativamente fácil de modificar e aumentar.
- **Arquiteturas Orientadas a Objetos:** os componentes de um sistema encapsulam dados e as operações que devem ser aplicadas para manipular os dados. A comunicação e a coordenação entre componentes são realizadas através da passagem de mensagens.
- **Arquiteturas em Camadas:** são definidas várias camadas diferentes, cada uma realizando operações que progressivamente se tornam mais próximas do conjunto de instruções de máquina. Na camada mais externa, os componentes atendem operações de interface do usuário. Na camada mais interna, os componentes realizam a interface com o sistema operacional. As camadas intermediárias fornecem serviços utilitários e funções de software de aplicação.

3.6 Json

O JSON, como é conhecido o *JavaScript Object Notation*, foi concebido para ser um formato leve de troca de dados. Sua criação foi baseada em um subconjunto da linguagem de programação *JavaScript*. O JSON é desenvolvido em texto e completamente independente de linguagem. (JSON.ORG, 2017)

Dentre as vantagens do uso do JSON, destaca-se o *parsing* mais fácil, maior velocidade no transporte de dados, tamanho reduzido dos arquivos e maior simplicidade em sua leitura.

Ao analisarmos o JSON, destacamos duas estruturas básicas:

- Coleção de pares nome/valor, o que é caracterizado em várias linguagens como objetos.
- Um agrupamento de valores ordenados em uma lista, que na maioria das linguagens se caracteriza como um *array* ou lista.

Podemos representar o JSON das seguintes formas:

- Objeto: conjunto desordenado de pares nome/valor.
- *Array*: coleção de valores ordenados.
- Valor: assume um dos seguintes conjuntos, uma cadeia de caracteres (*string*), um número, booleano (verdadeiro ou falso), nulo, um objeto ou um *array*. Podendo ser estruturas aninhadas.
- *String*: uma coleção de nenhum ou mais.
- Número: estrutura similar a um número em C ou Java.

3.7 Jersey

O JERSEY, é um framework desenvolvido pela Sun de código aberto e utilizado para o desenvolvimento de serviços web RESTful (ORACLE CORPORATION). Tem em sua implementação o suporte para as anotações definidas no JSR-311 (JAX-RS) que tem um foco voltado para a utilização de URIs e utilização do protocolo HTTP.

As anotações JAVA tem o objetivo de adicionar declarações dos metadados nos objetos criados, possibilitando que as configurações de classe fiquem dentro do seu escopo, em vez de utilizar um XML à parte. O JERSEY foi utilizado para a implantação das características de uma arquitetura REST por sua compatibilidade com a linguagem Java.

3.8 Android

O sistema operacional Android foi desenvolvido pela Android, Inc., adquirida pelo Google em julho de 2005. Em novembro de 2007, foi formada a Open Handset Alliance, um

consórcio para desenvolver o Android, impulsionando a inovação na tecnologia móvel, melhorando a experiência do usuário e reduzindo os custos. (DEITEL, 2013)

A pesquisa mais recente da Gartner, disponibilizada em fevereiro de 2018, apresenta os resultados das vendas dos celulares do tipo *smartphone* ao longo de 2017, onde 85,9% dos smartphones vendidos usam o sistema operacional Android, 14% são baseados em iOS e demais sistemas com 0,1%.

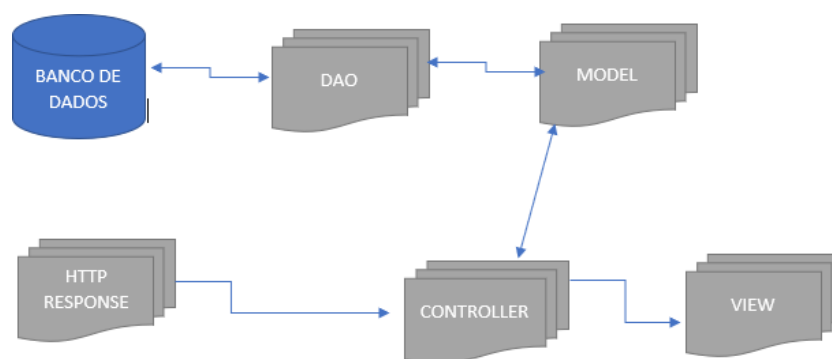
Os aplicativos Android são desenvolvidos com Java – a linguagem de programação mais usado do mundo. Essa linguagem foi uma escolha lógica para a plataforma Android, porque é poderosa, gratuita e de código-fonte aberto. (DEITEL, 2013). Atualmente, podemos desenvolver também através da linguagem de programação Kotlin é uma linguagem estaticamente tipada, desenvolvida pela JetBrains.

4 ARQUITETURA DO SERVIÇO WEB MOODLE IFPE

A arquitetura dos serviços seguiu o modelo em camadas para que ocorresse uma maior facilidade na compreensão do sistema, como também, um meio facilitador para manutenção do mesmo.

Dessa forma conseguimos manter padrões de projeto que são utilizados para o baixo acoplamento e a alta coesão. A coesão está ligada a ideia de que uma classe deve possuir uma única responsabilidade, ou seja, não possuir responsabilidades que não são as suas. Enquanto o acoplamento define a dependência entre uma classe e outra para o seu funcionamento.

Figura 3 - Arquitetura do Serviço WEB



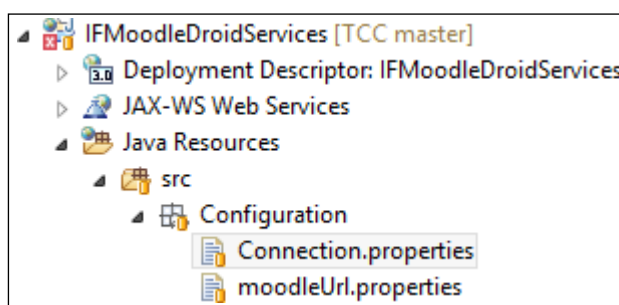
Fonte: os próprios autores

4.1 Camadas do Webservice Moodle IFPE

4.1.1 Configuration

Esse pacote possui as classes que ficam responsáveis pelas configurações das propriedades de comunicação com servidor do Moodle.

Figura 4 - Classes do pacote Configuration



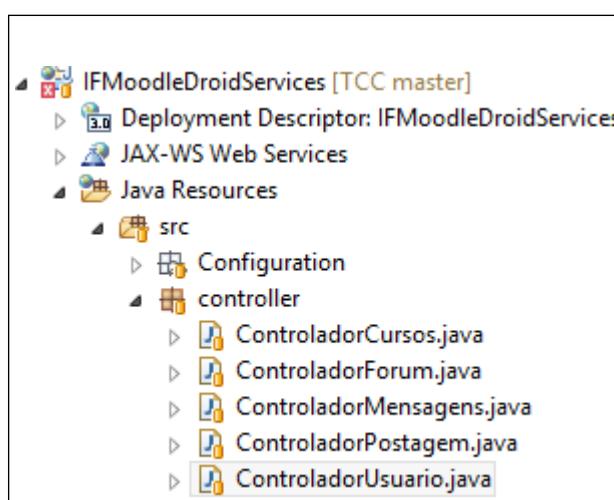
Fonte: os próprios autores

4.1.2 Controller

O objetivo deste pacote é fazer com que as requisições enviadas pelo usuário sejam direcionadas corretamente para as entidades que estão ligadas diretamente ao recurso solicitado.

Além de interagir com as ações do usuário, as classes relacionadas ao pacote, necessitam interagir com a visualização final do usuário, ou seja, uma interação com a tela correta para efetuar os ajustes solicitados através das interações com o sistema.

Figura 5 - Classes do pacote Controller



Fonte: os próprios autores

4.1.3 DAO

A utilização do padrão de projeto, *Data Access Object* ou DAO, é utilizado no projeto para que possamos separar a regra de negócio da persistência de dados. Dessa forma, uma vez alterada a forma de persistir os dados, a regra lógica desenvolvida não irá sofrer alterações.

Classes que estão dentro da arquitetura DAO, estão responsáveis em realizar as trocas de informações com os Sistemas de Gerenciamento de Banco de Dados, chamados de SGBD, fornecendo as operações de criação, leitura, alteração e deleção dos dados existentes nos sistemas. Podemos dividir o modelo de acesso aos dados em três classes:

- Interface: classes que definem operações padrões que o objeto deverá executar
- Classe Concreta: classes que implementam as interfaces, que possuem a responsabilidade de obter dados da fonte de dados
- Objeto: classes que possuem métodos simples de *get* e *set* com a responsabilidade de armazenar os dados recuperados usando classes do padrão DAO

Figura 6 - Trecho de código de classe concreta

```

public class UsuarioDAO implements UsuarioDAOInterface {

    @Override
    public Usuario verifyUsuario(Usuario usuario) throws Exception {
        Connection connection = null;
        PreparedStatement preparedStatement = null;
        ResultSet resultSet = null;
        Usuario usuarioTemp = null;
        boolean resultVazio = true;

        try {
            String query = "select * from mdl_user where username = ? and password = md5(?)";

            connection = ConnectionFactory.getInstance().getConnection();

            preparedStatement = connection.prepareStatement(query);
            preparedStatement.setString(1, usuario.getLogin());
            preparedStatement.setString(2, usuario.getSenha()
                + ConstantesSistema.getCODEPASSWORD());

            resultSet = preparedStatement.executeQuery();
        }
    }
}

```

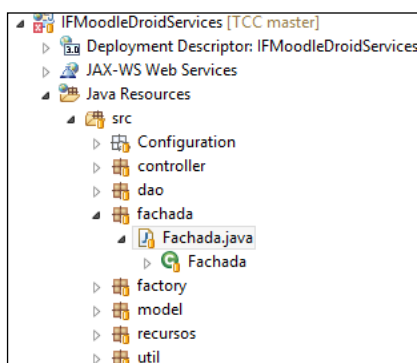
Fonte: os próprios autores

4.1.4 Fachada

As classes que aqui se encontram, utilizam-se do padrão de projeto *Facade*, um padrão estrutural, que define relações entre objetos e classes. Ao utilizar este tipo de padrão de projeto em nossa arquitetura, conseguimos simplificar a interface reduzindo a complexidade da API desenvolvida.

Ao utilizarmos este tipo de padrão, ficamos atrelados a uma única classe que irá fornecer métodos mais simples para as chamadas de métodos já existentes.

Figura 7 - Classe do pacote Fachada



Fonte: os próprios autores

4.1.5 Factory

Dentre os padrões de projeto mais utilizados nas arquiteturas de sistemas, os padrões *Factory*, ficam responsáveis por encapsular a criação dos objetos.

Para a arquitetura do projeto, criamos uma fábrica de conexão, que tem como responsabilidade a criação de instância de conexão, bem como estabelecer o tipo de conexão e drivers que serão utilizados.

Figura 8 - Trecho de código de classe

```
public class ConnectionFactory {
    private static ConnectionFactory connectionFactory;
    private Connection connection;
    private Properties properties = new Properties();
    private String urlProperties = ConstantesSistema.URL_PROPETIE;

    public static ConnectionFactory getInstance() {
        if (connectionFactory == null) {
            connectionFactory = new ConnectionFactory();
        }
        return connectionFactory;
    }

    public Connection getConnection() throws Exception {
        if (connection == null) {
            try {
                properties.load(this.getClass().getResourceAsStream(
                    urlProperties));
                String connection_url = properties.getProperty("url");
                String user = properties.getProperty("user");
                String password = properties.getProperty("password");
                DriverManager.registerDriver(new com.mysql.jdbc.Driver());
                connection = DriverManager.getConnection(connection_url, user,
                    password);
            } catch (SQLException ex) {
                throw new Exception("Problema ao se conectar com a base de dados.");
            } catch (IOException ex) {
                throw new Exception("Problema ao encontrar arquivos de configuracao.");
            }
        }
    }
}
```

Fonte: os próprios autores

4.1.6 Model

Neste ponto da arquitetura desenvolvida, temos as classes que vão definir nossos objetos. Ou seja, suas características e parâmetros a serem utilizados em instâncias que venham a ser criadas.

Figura 8 - Trecho de código de classe

```

@XmlRootElement
public class Usuario {

    private long id;
    private String login;
    private String senha;
    private String nome;
    private String sobrenome;

    public Usuario(long id, String login, String senha, String nome, String sobrenome) {
        this.id = id;
        this.login = login;
        this.senha = senha;
        this.nome = nome;
        this.sobrenome = sobrenome;
    }
}

```

Fonte: os próprios autores

4.1.7 Recursos

Para este nível da arquitetura, criamos a forma de comunicação com os serviços. Auxiliados pelo recurso nativo da linguagem Java, as *Java Annotations* ou anotações, permitem adições de informações de metadados no código fonte desenvolvido.

As anotações podem ser utilizadas de três formas básicas. Passando instruções ao compilador, instruções em tempo de execução e instruções em tempo de compilação. O uso de instruções ao compilador está restrito a três anotações nativas: *Deprecated*, *Override* e *SupressWarnings*.

Em nossa arquitetura, definimos os tipos de método, bem como os parâmetros que vão ser utilizados para cada serviço de interação com o usuário.

Figura 9 - Trecho de código de classe, serviço de login

```

@Path("/listener")
public class ListenerResourcePost {

    @Path("/login")
    @POST
    @Produces(MediaType.APPLICATION_JSON)
    @Consumes(MediaType.APPLICATION_FORM_URLENCODED)
    public Usuario fazLogin(@FormParam("login") String login,
        @FormParam("senha") String senha,
        @FormParam("flagDecryptacao") String flagEncriptacao,
        @Context HttpServletResponse servletResponse) throws IOException {

        return Fachada.getInstance().retornaUsuario(login, senha,
            flagEncriptacao);
    }
}

```

Fonte: os próprios autores

4.1.8 Util

Para esta parte da arquitetura do sistema, as classes aqui existentes, ficam responsáveis pelas constantes do sistema, bem como a parte inicial de encriptação dos dados, para que possamos entregar um sistema íntegro.

5 SERVIÇOS MOODLE IFPE

Neste capítulo são apresentados serviços desenvolvidos pela API a fim de criar um padrão de envio e recebimento de respostas entre o aplicativo desenvolvido e API.

5.1 Login (POST, /login)

Para este serviço passamos via método POST, o *login* e senha do usuário, bem como a flag de encriptação para que as informações não sejam descobertas durante o tráfego de rede. Tem-se como retorno a entrada do usuário após validação na aplicação.

5.2 Retorna Notícias (POST, /retornaNoticias)

Para este serviço passamos via método POST, o *login* e senha do usuário, bem como a flag de encriptação para que as informações não sejam descobertas durante o tráfego de rede. Dessa forma, temos como retorno de método as notícias gerais existentes na plataforma EAD do IFPE.

5.3 Retorna Fóruns Cursos (POST, /retornaForunsCursos)

Para este serviço passamos via método POST, o *login* e senha do usuário, bem como a flag de encriptação para que as informações não sejam descobertas durante o tráfego de rede. Dessa forma, temos como retorno de método todos os fóruns que o usuário que fez login possui.

5.4 Retorna Tópicos Fóruns (POST, /retornaTopicosForuns)

Para este serviço passamos via método POST, o *login* e senha do usuário, o curso no qual irá analisar os tópicos de um fórum, bem como a flag de encriptação para que as informações não sejam descobertas durante o tráfego de rede. Dessa forma, temos como retorno de método os tópicos do fórum de um determinado curso.

5.5 Meus Contatos (POST, /meusContatos)

Para este serviço passamos via método POST, o *login* e senha do usuário, além do código de usuário que está logado, bem como a flag de encriptação para que as informações não sejam descobertas durante o tráfego de rede. Dessa forma, temos como retorno de método todos os contatos de um determinado usuário.

5.6 Enviar Mensagens (POST, /enviarMensagens)

Para este serviço passamos via método POST, o *login* e senha do usuário, código de contato no qual irá enviar mensagem, corpo da mensagem, bem como a flag de encriptação para que as informações não sejam descobertas durante o tráfego de rede. Dessa forma, temos como retorno de método o sucesso da mensagem enviada ou o erro no envio.

5.7 Acrescentar Contatos (POST, /acrescentaContatos)

Para este serviço passamos via método POST, o *login* e senha do usuário, código do contato a ser adicionado, bem como a flag de encriptação para que as informações não sejam descobertas durante o tráfego de rede. Dessa forma, temos como retorno de método o sucesso da inclusão do contato ou o erro no acréscimo.

5.8 Excluir Contatos (POST, /excluirContatos)

Para este serviço passamos via método POST, o *login* e senha do usuário, código do contato a ser excluído, bem como a flag de encriptação para que as informações não sejam descobertas durante o tráfego de rede. Dessa forma, temos como retorno de método o sucesso da exclusão do contato ou o erro na exclusão.

5.9 Bloquear/Desbloquear Contatos (POST, /bloquearDesbloquearContatos)

Para este serviço passamos via método POST, o *login* e senha do usuário, código do contato a ser bloqueado/desbloqueado, bem como a flag de encriptação para que as informações não sejam descobertas durante o tráfego de rede. Dessa forma, temos como retorno de método o sucesso do bloqueio/desbloqueio do contato ou o erro na operação.

6 APLICATIVO MOODLE VOX

6.1 Moodle Droid

Dados da Anatel indicam que o Brasil terminou dezembro de 2018 com 229,2 milhões de celulares e densidade de 109,24 cel/100 hab, isto é, mais de um celular por habitante em média. Tendo em mente essa realidade a adaptação para tecnologias móveis Android, que é hoje o sistema operacional mais difundido entre os aparelhos móveis, facilitaria o acesso das lições diárias e das aulas, além de fóruns e salas para conversação entre alunos em qualquer lugar com o uso de tecnologias.

A partir dessas conclusões desenvolveu-se o aplicativo Moodle Droid sendo uma ferramenta feita para dispositivos móveis, com sistema operacional Android, para o acesso ao conteúdo de EaD do Instituto Federal de Pernambuco, que se utiliza da plataforma Moodle de ambientação virtual de salas de aula.

O Moodle Droid foi desenvolvido anteriormente a produção do trabalho descrito neste documento por outra equipe de desenvolvedores que atuava no mesmo projeto de pesquisa e serviu de base para o desenvolvimento do projeto Moodle Vox.

O IFPE Moodle Droid utiliza a API descrita anteriormente para acessar o banco de dados do Moodle e carregar as informações dos usuários, estrutura de cursos e de mensagens já apresentada, essa aplicação também tem como utilidade validar as funcionalidades da API.

Figura 10 - Tela Inicial e Login do aplicativo MoodleDroid



Fonte: os próprios autores

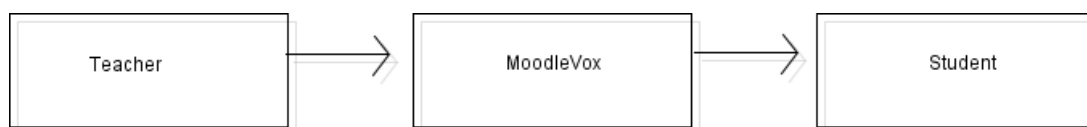
6.2 Características Básicas

O MoodleVox é uma aplicação desenvolvida especialmente para fins educacionais. Seu propósito é prover um ambiente virtual onde estudantes e professores participem em um ensino colaborativo a partir da troca de mensagens em chats e de discussão em fóruns que é idealizada para ser utilizada por usuários com algum tipo de deficiência visual, e toda sua interação é exclusivamente feita por comandos de voz. O sistema então apresenta, de forma oral, os comandos disponíveis que podem ser usados enquanto a aplicação estiver funcionando.

A conversação entre usuário-MoodleVox é garantida pela biblioteca de voz inteligente disponibilizada pelo Google e a lógica de negócio na interação do MoodleVox foi inteiramente baseada nos princípios organizacionais do Moodle.

O relacionamento entre professor e aluno na ferramenta MoodleVox foi simplificada na imagem abaixo:

Figura 11 - Componentes envolvidos no Moodle Vox



Fonte: os próprios autores

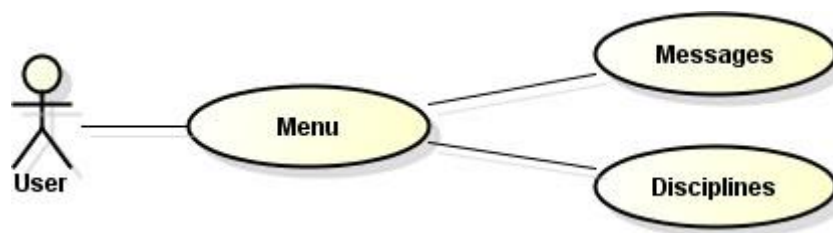
O professor torna acessível para os estudantes uma mensagem educacional. Essa mensagem pode ser síncrona, a partir de conversação pelo chat, ou pode ser assíncrona a partir dos tópicos nos fóruns das disciplinas. O retorno dos estudantes completa o loop de comunicação, fazendo com que o receptor se torne o professor e vice-versa.

A interação síncrona permite um maior grau de interação, porém a interação assíncrona cria a independência do tempo, tão almejada na Educação a Distância. A combinação desses dois modelos de interação fortifica a flexibilidade do conteúdo a ser transmitido para todos os usuários e minimiza as dificuldades encontradas pelos deficientes visuais para acessar o conteúdo educacional e a se comunicarem com seus colegas e professores.

6.3 Menu de Interação

A partir dos diagramas apresentados nos próximos tópicos, poderão ser percebidas as atividades interativas oferecidas pelo MoodleVox. A partir do Menu raiz de aplicação o usuário deve decidir para onde proceder, para o menu de Mensagens ou de Disciplinas, como mostrado abaixo:

Figura 12 - Fluxo Principal do MoodleVox

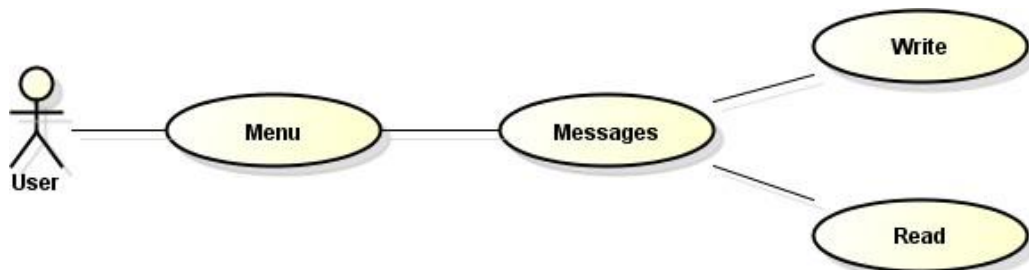


Fonte:Alana Luna, Pedro Jatoba, Thiago Viana,Felipe Pacheco (2016)

6.3.1 Módulo de Mensagens

A partir do menu de Mensagens o sistema disponibiliza opções de Escrita e Leitura para então o usuário escolher uma operação para proceder, como mostrado abaixo:

Figura 13 - Menu de Mensagens Fluxo Principal

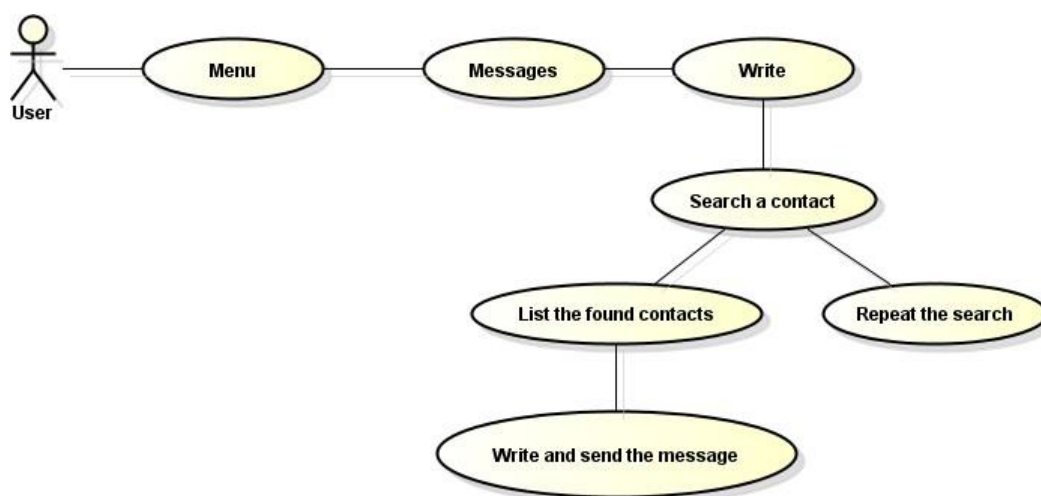


Fonte:Alana Luna, Pedro Jatoba, Thiago Viana,Felipe Pacheco (2016)

Se a opção de escrita é solicitada, o sistema pede para o usuário diga o nome do contato para o qual ele deseja mandar uma mensagem. A partir do nome suprido pelo usuário, o sistema

pesquisa em sua base de dados. Caso nenhum resultado condizer com o nome fornecido, uma mensagem de erro é falada ao usuário e o usuário retorna à pesquisa, e caso seja encontrado um ou mais contatos condizentes com o nome fornecido o sistema então apresenta uma lista com os resultados, e então o usuário deve escolher qual contato da lista de resultados ele deseja enviar a mensagem. Após escolhido o contato, o sistema pedirá ao usuário que fale a mensagem que deseja ser enviada, feito isso o sistema então manda a mensagem para o destinatário.

Figura 14 - Menu de Mensagens Módulo de Escrita



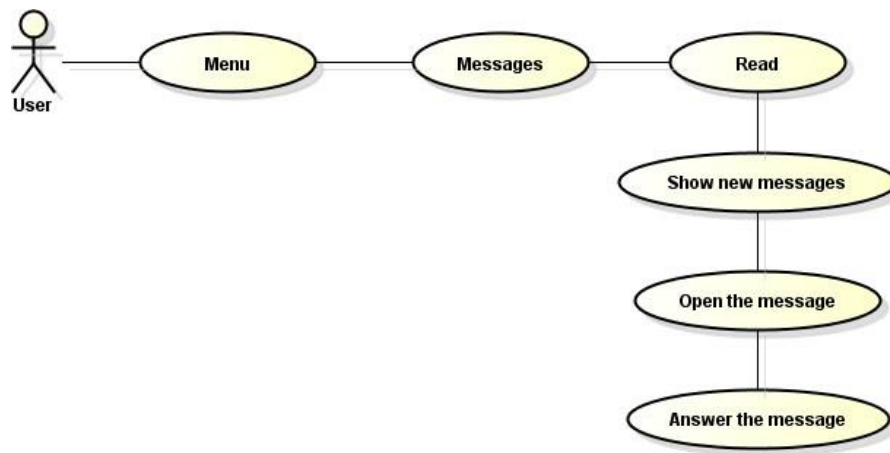
Fonte: Alana Luna, Pedro Jatoba, Thiago Viana, Felipe Pacheco (2016)

Se o módulo de leitura é selecionado, o sistema informa ao usuário se ele tem novas mensagens não lidas.

Caso não existam novas mensagens a aplicação informa ao usuário que ele tem uma caixa de entrada vazia, tendo novas mensagens, a aplicação informa o contato que enviou a mensagem e pergunta se o usuário pretende lê-la, caso seja pedida a leitura da mensagem o sistema então lê todo o seu conteúdo para o usuário e pergunta se ele pretende responder.

Caso deseje responder o usuário deve falar a mensagem logo após confirmar e o sistema a enviará em seguida, caso o usuário responda negativamente tanto quando for ler a mensagem ou respondê-la o sistema automaticamente voltara para o menu de leitura.

Figura 15 - Menu de Mensagens Módulo de Leitura



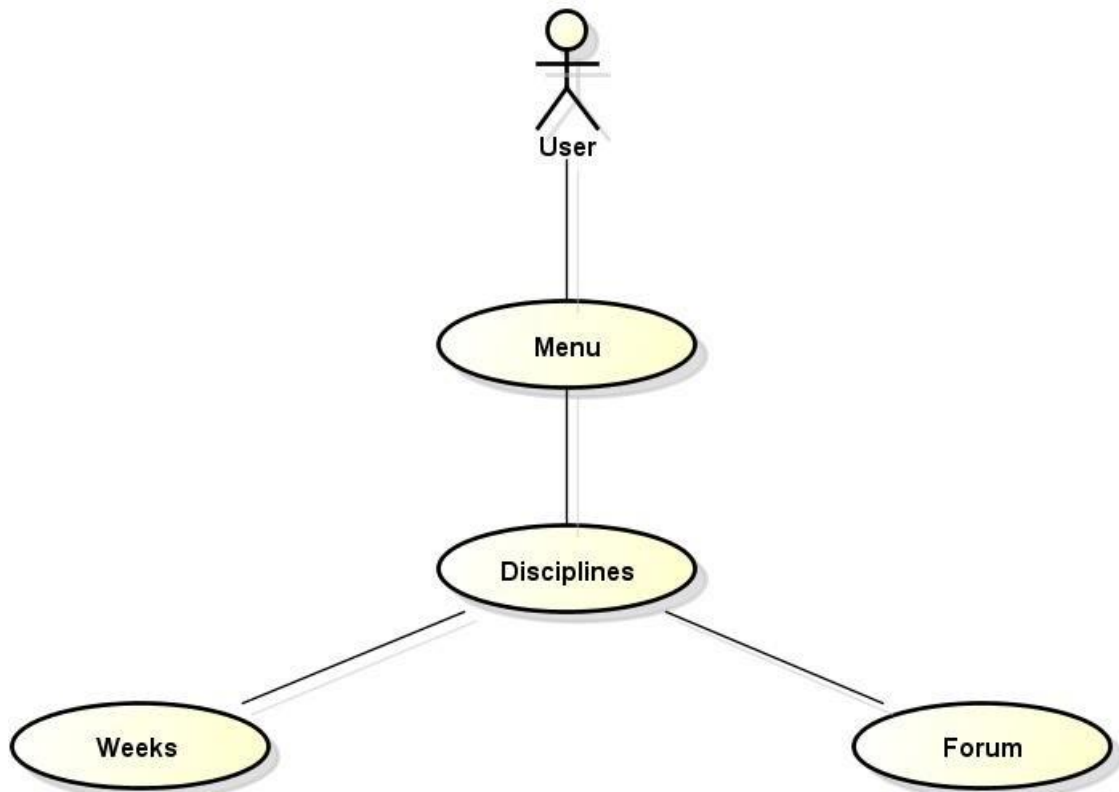
Fonte: Alana Luna, Pedro Jatoba, Thiago Viana, Felipe Pacheco (2016)

Para todas as opções de interação, o usuário pode retornar para os menus anteriores ou para o menu raiz do sistema, mesmo durante a execução da operação. O usuário pode também pedir para que o sistema repita as opções que foram apresentadas.

6.3.2 Módulo de Disciplinas

No módulo de disciplinas o usuário é apresentado às opções de disciplinas nas quais ele está devidamente matriculado e então ele escolhe qual disciplina acessar, e o módulo subsequente de Semanas e o módulo de Fóruns, o primeiro módulo garantindo acesso às aulas, arquivos e links da disciplina enquanto o segundo módulo apresenta os fóruns disponíveis da disciplina como também a opção de criar um novo fórum, como apresentado na figura a seguir:

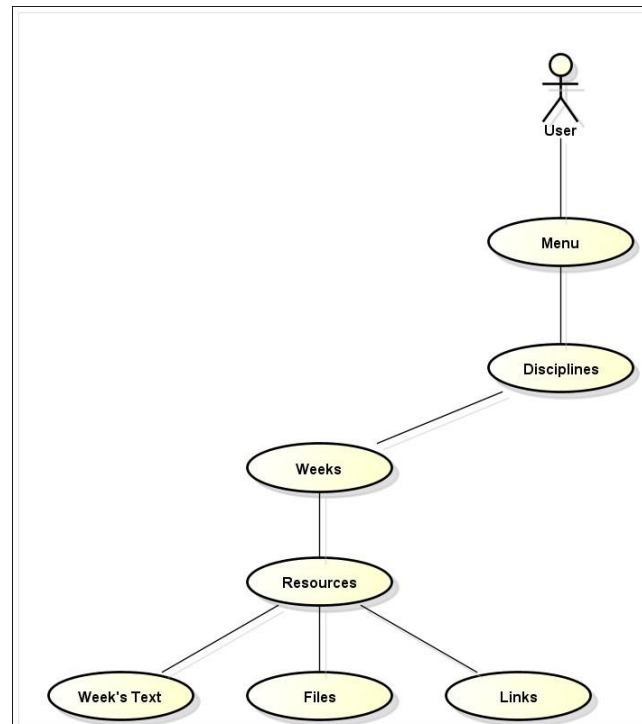
Figura 16 - Menu de Disciplinas Fluxo Principal



Fonte: Alana Luna, Pedro Jatoba, Thiago Viana, Felipe Pacheco (2016)

Ao entrar no módulo de Semanas o usuário têm as opções de escolher entre três alternativas de conteúdo: O Texto da Semana, que contém um resumo simplificado das aulas que foram apresentadas na semana atual, o módulo de Arquivos que contém todos os arquivos usados e disponibilizados pelo professor e o módulo de Links que contém um compilado de links disponibilizado e produzido pelos professores.

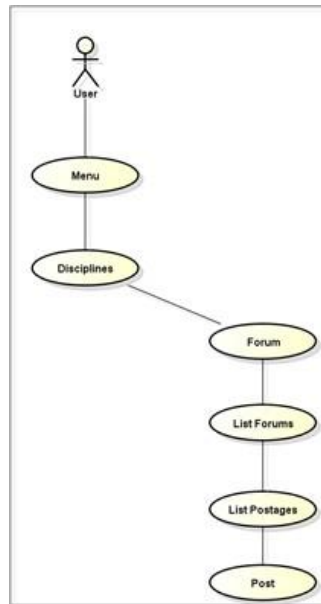
Figura 17 - Menu de Disciplinas Módulo de Semanas



Fonte: Alana Luna, Pedro Jatoba, Thiago Viana, Felipe Pacheco (2016)

Entrando no módulo de Fóruns o usuário terá a opção de participar de um fórum já criado ou criar um novo, quando estiver participando de um fórum o usuário pode postar no fórum ou ler as mensagens já postadas nele.

Figura 18 - Menu de Disciplinas Módulo de Fóruns



Fonte:Alana Luna, Pedro Jatoba, Thiago Viana,Felipe Pacheco (2016)

7 CONCLUSÕES

Este trabalho apresentou uma proposta de arquitetura de serviços para utilização dos recursos existentes na plataforma de ensino a distância Moodle do IFPE, bem como uma solução de um aplicativo móvel para a expansão do ensino e facilidade de acesso ao conteúdo ministrado pela instituição.

Os resultados deste trabalho, tanto a arquitetura de serviços quanto o aplicativo desenvolvido, buscam contribuir para que a comunidade acadêmica consiga evoluir em mais uma frente tecnológica de ensino.

Com tudo que foi desenvolvido, destacamos a simplicidade de acesso à informação na plataforma Moodle e os serviços utilizados para construir a informação desejada, com as suas devidas interações entre a arquitetura dos serviços e o aplicativo que consome.

A adaptação da plataforma Moodle Android que foi proposta desenvolveu-se até o presente momento funcionalidades de entrada no menu inicial feita por voz e demais interações entre menus subsequentes integradas com Feedback de erros por incoerência de reconhecimento de voz (sotaque, ruídos). As funcionalidades sobre o menu de Mensagens, e subsequentemente os fluxos de Leitura e Escrita foram desenvolvidas e explicadas em LUNA(2017), a outra metade do processo de desenvolvimento do Moodle Vox referentes ao núcleo de Disciplinas foram desenvolvidas neste trabalho.

O desenvolvimento do Moodle Vox gerou a publicação de um artigo intitulado: *Implementing an Android Tool for Visually Impaired Students of E-Learning* (ANEXO 1) na revista IJERA em 2016 (LUNA; PACHECO; JATOBÁ; VIANA, 2016), que tem como foco a demonstração dos fluxos do sistema. Luna e colegas realizaram uma extensão desse artigo investigando o estado da EaD no Brasil, sendo publicado o livro *The Use of Mobile Learning for Visually Impaired Students* (LUNA;DANTAS;VIANA, 2019)

7.1 Trabalhos Futuros

Com o intuito de melhorar a plataforma de ensino a distância, seria bastante desejável a utilização do aplicativo desenvolvido em meios reais, além da atualização do aplicativo para versões mais recentes do Android, comprovando assim a eficiência do *e-learning* junto ao *m-learning* ao auxílio da educação do Instituto Federal de Ciência e Tecnologia de Pernambuco.

Para o MoodleVox almeja-se: a total interação com a plataforma desenvolvida para adaptação do Moodle para Android; a disponibilização da ferramenta (protótipo), com a funcionalidade para acessibilidade de deficientes visuais, para a EaD do IFPE; e a publicação de artigos em revistas qualificadas e especializadas na área.

REFERÊNCIAS

Análises de LMS. Disponível em: <<https://www.capterra.com/learning-management-system-software/>> Acesso em: 20 de jan. 2017

ASSOCIAÇÃO BRASILEIRA DE EDUCAÇÃO A DISTÂNCIA. **Censo EAD Brasil 2016:** Relatório Analítico da Aprendizagem a Distância no Brasil. Disponível em: <http://abed.org.br/censoead2016/Censo_EAD_2016_portugues.pdf>

ARNON, R.G.O, **SOA Patterns.** NewYork: Manning Publications Co., 2012. ISBN 1933988266.

BRASIL. DECRETO Nº 9.057, DE 25 DE MAIO DE 2017. Brasília, 25 de maio de 2017. Disponível em: <https://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2017/Decreto/D9057.htm#art24>

DESMOND, K. **The Future of Learning:** From e-learning to m-learning. Ziff-Nov.2002, Papiere, 119, 2002. Disponível em: <<https://eric.ed.gov/?id=ED472435> >

EUZENAT, J. **Eight questions about semantic web annotations. Intelligent Systems, IEEE,** v. 17, n. 2, p.54-63, March 2002. ISSN 1541-1672.

Estatísticas de Celulares no Brasil. Disponível em: <<http://www.teleco.com.br/ncel.asp>>

FACEBERG, T.; REKKEDAL, T. Russell, J. **Designing and Trying Out a Learning Environment for Mobile Learners and Teachers,** 2002. Disponível em <http://learning.ericsson.net/mlearning2/project_one/NKI2001m-learning2.doc>

FIELDING, R. T. **REST: Architectural Styles and the Design of Network-based Software Architectures.** Tese (Doctoral dissertation) – University of California, Irvine, 2000. Disponível em: <<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm> >.

FOWLER, M. **Richardson Maturity Model**: Steps toward the glory of rest. 2010. Disponível em: <<https://martinfowler.com/articles/richardsonMaturityModel.html>>

JOSUTTIS, N. M. **SOA in Praticce: The Art of Distributed System Design**. Beijing: O'Reilly, 2007. ISBN 978-0-596-52955-0.

KENSKI, V. M. **Educação e Tecnologias**: o novo rimo da informação. Campinas, SP: Papiros, 2012.

LUNA, A.; DANTAS, R.; VIANA, T. **The Use of Mobile Learning for Visually Impaired Students**, 2019.

LUNA, A. et al. **Implementing an Android Tool for Visually Impaired Students of E-Learning**, 2016.

Linguagem Kotlin. Disponível em: <<https://kotlinlang.org/>>

Moodle. Disponível em <https://docs.moodle.org/38/en/Main_page> Acesso em: 20 de jan. 2017

PAPAZOGLU, MIKE P.; HEUVEL, WILLEM- Jan, 2007, **Service Oriented Architectures**: Approaches, Technologies and Research Issues, VLDB Journal, Springer-Verlag.

PRENSKY, M. **Digital Natives, Digital Immigrants Part 1. On The Horizon**, MCB UP Ltd, v.9, n. 5, 2001.

TRIFONOVA, A. **Mobile Learning – review of the literature**. Technical Report DIT-03-009, University of Trento, March 2003. Disponível em: <<http://eprints.biblio.unitn.it/359/>>

UNESCO. **Diretrizes de Políticas da UNESCO para Aprendizagem Móvel**, 2013.

Disponível em: <<http://unesdoc.unesco.org/images/0022/002277/227770por.pdf>>

VEEN, W; VRAKKING, B. **Homo Zappiens**: educando na era digital. Artmed Editora, 2009.

ANEXO 1

Implementing an Android Tool for Visually Impaired Students of E-Learning

Alana Da Silva Luna*, Pedro Igor Barroso Jatoba*, Thiago Viana*,
Felipe Pacheco*,
**Federal Institute of Pernambuco*

ABSTRACT

This article aims to describe the process of learning and development of an educational tool designed for mobile devices (smartphones) with Android technology. In summary, the application was developed based on the virtual learning environment Moodle and aims to develop a learning environment that supports the visually impaired students of e-learning, allowing them to ask questions, discuss and share ideas through forums and use chat rooms in real time. The fundamental purpose of this application is to cooperate with scientific knowledge in the sense that this is a representation of technological advance on the accessibility tools in distance education mode and provide comfort, flexibility and accessibility for the visually impaired students, realizing that education should always be inclusive.

Keywords: Accessibility, Assistive Technology, Educational Activities, E-learning, Interactive Tool, Mobile Devices.

I. INTRODUCTION

According to Article 1 of Decree No. 2494 which regulates Article 80 of the Law of Guidelines and Bases of National Education (Brazil, 1996): "E-learning is a way of teaching that enables self-learning, with the systematic mediation of organized didactic resources, presented in different media information, used alone or combined, and broadcast by various media".

E-learning can be set briefly as an educational modality in which students and teachers are separated physically or temporally and therefore it is necessary the use of information and communication technologies. Exercises and interaction between students and teachers are given through Virtual Learning Environments (VLEs). This interaction may be synchronous, real-time using chat and web conferencing, or asynchronous like forums and quizzes.

The first Brazil's EL experiences appeared in the early 1940s. The courses were prepared from printed materials and sent through correspondence, but these have failed to stimulate government and social responsiveness in the country for the new teaching method. In the 1960s, distance education acquired new media such as radio, television, video recorder and computer, in addition to the printed material.

With the advent of these new technologies and more recently the internet, distance education enabled the expansion of education in remote places and offered the opportunity of professional development for those students who could not attend regular classroom sessions.

Nowadays it is impossible to think about education without considering the strong presence of the Internet in the students' lives. According to the 2010 census of the Brazilian Institute of Geography and Statistics (IBGE), a third of the population has access to the internet, and according to data from the Ministry of Education (MEC), the courses of distance education in Brazil express a high growth rate of about 50% per year (ABED, 2011) while the regular education has a stabilizing trend, growing only 3.5% per year (Botelho, 2011).

Faced with the evident growth of distance education courses, several studies and surveys were developed to improve the methods of interaction between teacher and student in this modality. These researches have forwarded the distance education to a great evolution, especially because the teacher becomes a developer who contributes for the student's education and shows paths that can be followed to maximize learning (Saraiva, 1996).

Consequently, the prospect of the models of teaching was transformed and now the student acts as an autonomous subject during their cognitive development process. In other words, the teacher ceases to be centralized and the student begins to seek the knowledge through collaboration with the teacher and other students using technological tools. In this sense, distance education is characterized by the establishment of a communication in which teachers and students are physically separated, but are connected through a digital medium. Today, the most part of virtual learning environment offers tools focused exclusively on technologies for the Internet (Musa & Oliveira, 2000). The major focus of distance education is to bring the knowledge for areas of difficult access to education (Belloni, 1999) and the sad reality found in many places that have the greatest need for distance education is the difficulty of use and internet access.

In front of this obstacle, a new modality of education was developed, called Mobile Learning (M-Learning). The M-learning is a mobile learning that enables the execution of distance education environments through mobile devices such as palmtops, PDAs (Personal Digital Assistants), tablets and especially cell phones (Attewell, 2005). The student can access materials to study and interact with teachers and colleagues from anywhere using text messaging (SMS), photos, files, multimedia, e-mails and among others, even with the simplest model of a present-day cell phone (Mohamed, 2009).

According to the National Telecommunications Agency (ANATEL) and the Brazilian Institute of Geography and Statistics (IBGE), currently there are 192 million Brazilians (IBGE,2015) and 224 million mobile phones (ANATEL, 2015). In other words, there are more mobile phones in Brazil than Brazilians. And this is not necessarily reflected only in the metropolitan areas, given that, according to ANATEL, 5,564 municipalities in Brazil are served by mobile operators, representing 99.99% of the entire national territory (IBGE, 2015).

The future of distance education is very promising and it is also valid to point out that the today's world is very competitive to enter into labor market and requires a well-prepared professional who can achieve the requirements that the market demands. In this context of competition for a place in the market, it is important to note that according to the 2010 Census, of the 44 million Brazilians disabled who are of working age, 53.8% are unemployed or outside the labor market. This is due to the fact that there is a significant difference on the level of education between people with physical disabilities and the general population, because, according to the 2010 Census, 61.1% of the population aged 15 and over with disabilities have no education or has only incomplete primary. This percentage drops to 38.2% for people without physical disabilities.

It is apparent that we are living in a difficult scenario to provide education uniformly for everyone and that's why distance education, which already represents a breakthrough in educational accessibility offering education for the population that does not fit in the traditional education system, can also actively participate as an agent of the current conditions of education for those who are physically disabled.

Based on the above considerations, this project aims to conduct research in the area of technologies (specifically distance) for education mediated by mobile devices (especially mobile phones) in order to propose and model an interactive tool to aid the teaching-learning seeking to help the students in their interaction with distance learning courses. It is also a goal in this project to achieve the integration of this tool with the virtual learning environment currently used in the Directorate of Distance Education (dead) IFPE thus enabling students of e-Learning more ways of interaction and communication to optimize their learning process. Finally, the project also aims to assist the visually disabled students of distance education mode (to comprehend that education should always be inclusive and regardless of disabilities, which is paramount in the development of tools to aid these students and that can be applied in the realities of this important and growing mode of education within the IFPE.

II. RELATED WORK

In Brazil, the Law of Guidelines and Bases of National Education (Brazil, 1996), encourages, in Article 80, the development and broadcast of e-learning programs. The growing concern for an inclusive society, where the rights of minorities should be increasingly assured, is a reality in national and international scene.

People with visual impairments, but not only them, need access to technological tools that enable their educational and professional growth development. It is through the development of assistive technology, a concept used to identify resources and services that provide and expand functional abilities of people with disabilities, users of these technologies can increasingly be involved in activities that include the use of products, services and information.

Brazil has been following the development of Assistive Technology more slowly, compared to other countries, including other Latin American countries. However, there was an awakening to the issue in the last decade, but it is still far short of what some authors point out as desirable. Unfortunately, there are few offers of this category of applications in Portuguese, but some examples can be found.

III. MOODLE VOX SYSTEM

In the following subsections is shown the system as a whole and its main features.

3.1 General Explanation of Moodle Vox Environment

The MoodleVox is an application specially designed for educational activities. Its purpose is to provide a virtual environment where students and teachers participate in collaborative learning through the exchange of messages in chats and discussions in forums.

The tool has no visual interface, all interactions are by means of voice commands. It was developed for the system a speech recognition library containing all valid interaction commands that can be used while the application is running. The conversation between the MoodleVox system and the user is fulfilled through intelligent voice provided by Google and the logic of the MoodleVox interaction menu was based on organizational principles of Moodle. The relationship between teacher and student in MoodleVox tool environment.

The transmitter (the teacher) makes available to the receivers (students) an educational message. This message can be synchronous, through oral conversation in the chat, and you can send / receive the message instantly or can be asynchronous through topics in the forums of disciplines. Feedback from receivers complete the communication loop, so that, accordingly, the receiver turns into transmitter and vice versa. The synchronous interaction allows a greater degree of interaction. Already asynchronous interaction creates time independence as well as space. The combination of these two modes of interaction ensures flexibility of content to be transmitted to users of the system and minimizes the difficulties of the visually impaired to access the educational content and to communicate with colleagues and teachers.

3.2 Defining System Features

The requirements of survey activity fundamentally contributed to the software to be built correctly. In other words, it was necessary for those involved in the project know exactly what was expected of the application. The versioning, change control and monitoring of the progress of tasks were carried out through periodic in-person meetings and the use of GitHub's web hosting service.

3.3 Moodle Vox Interaction Menu

Through diagrams presented in the next topics, it can be viewed the interactive activities offered by MoodleVox. From the Menu application root, the user must decide whether to proceed with the Messages menu or the Disciplines menu, as shown in Figure 1.

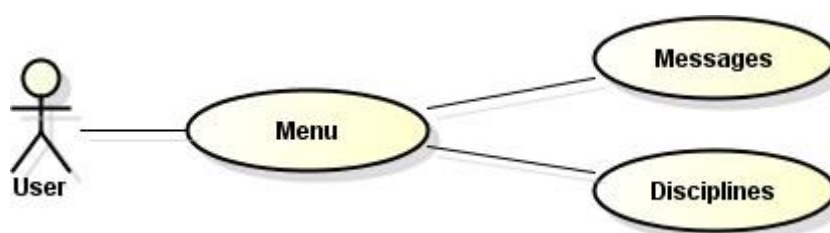


Figure 1. Diagram of Moodle Vox's main flow

3.4 Moodle Vox Interaction Menu – Message Module

From the Messages menu, the system provides the writing option and the reading option for the user to choose which operation to proceed, as shown graphically in the diagram in Figure 2.

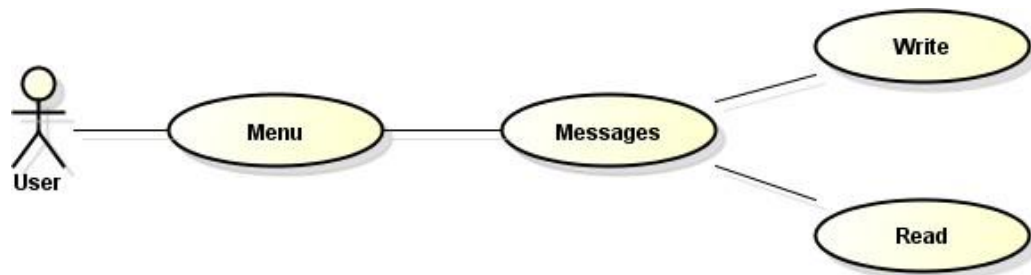


Figure 2. Message menu - Main Flow

If the 'Write' option was requested, the system prompts the user to enter the name of the contact to whom wishes to send a message. From the name supplied by the user, the system searches in their database. If no matching contact was found, an error message in the search is uttered by the system and the user can return to the search. Otherwise, with success in the search, the system returns a list of sought contacts based on the name that the user aforementioned. The user must choose one contact from all the contacts listed to send the message. Then, the system prompts the user to write the text of the message and send to the desired receiver.

If the 'Read' option is requested, the system informs if the user has new messages. If there are no messages, the application informs the user the status of the empty inbox. Having new messages, the application informs the emitter contact of the message and asks if the user wants to open it. If the answer is positive, the system reads the message's content to the user and asks if the user wishes to answer it. If so, the user can record the text of your message and then the system will send it. If the user responds negatively when it is prompted to open the message or whether to reply to the message, the system automatically returns to the 'Read' menu. As shown in Figure 3 and Figure 4.

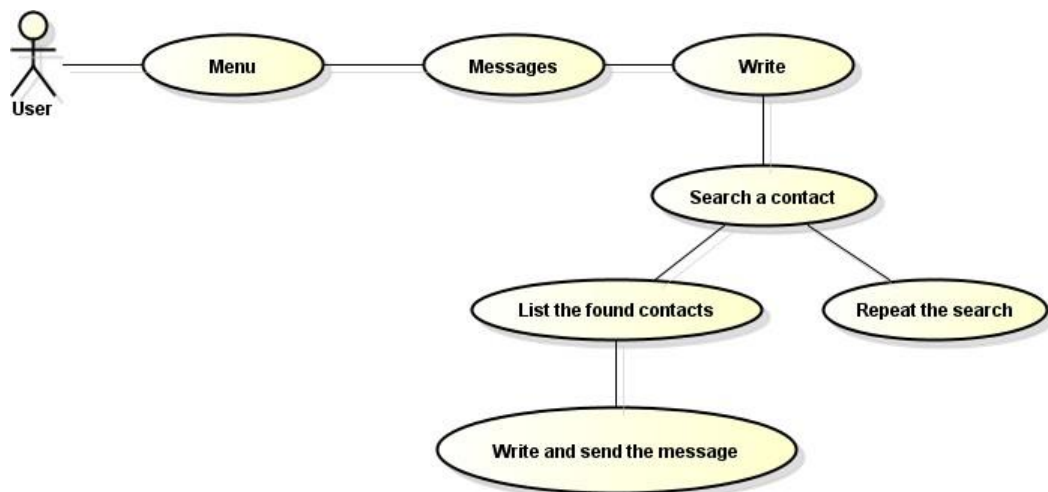


Figure 3. Message menu: Diagram of interaction - Write Module

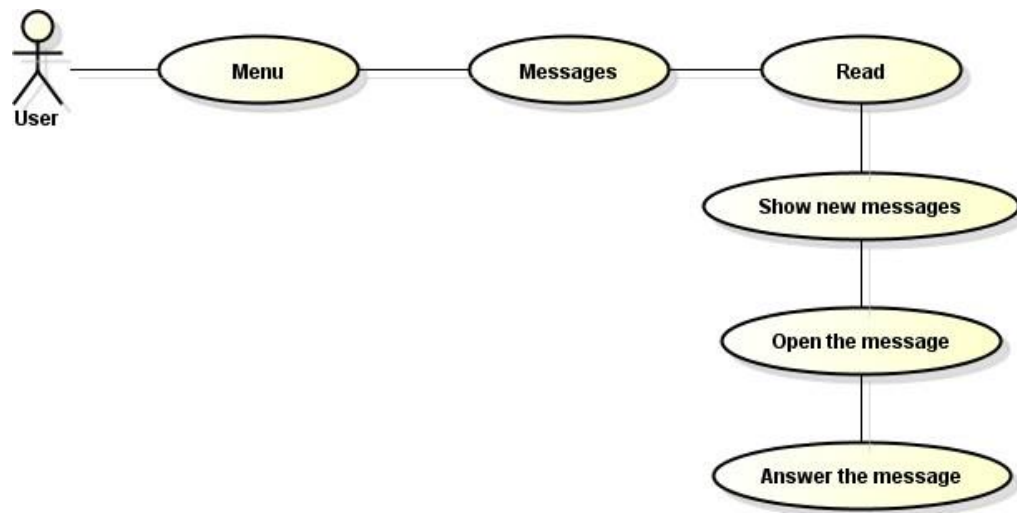


Figure 4. Message menu: Diagram of interaction - Reading Module

For both of interaction options, either in writing menu or reading menu, at any time during the execution of the operation, the user can return to the previous menu or return to the root menu. The user can also ask the system to repeat what was said in the current menu.

3.5 Moodle Vox Interaction Menu – Disciplines Module

In the Disciplines module the user has the option to choose which disciplines they want to access, and following this are the Weeks module and the Forum module of the chosen discipline, the first being the module to access the available classes, files and links and the other the module to access the existing forums about that discipline or to create one which are laid down in the Figure 5.

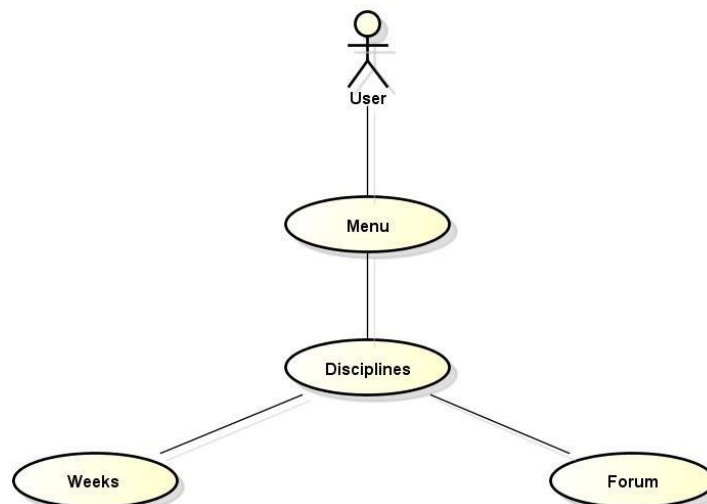


Figure 5. Disciplines menu - Main flow

Upon entering in the Weeks module, the user has the option to choose the Resources module and then three alternatives of content: The Week's Text that is a simple resume of this week classes, the Files module which contains all files provided by the teachers and the Links module that contains useful links provided by teachers. Lastly, entering the Forum Module will

give the user the option to participate on forums or create one, to participate the user can post on the already create forum and read the postages made by other users.

IV. CONCLUSION

E-learning is the decentralization of teacher's figure and the development of a more autonomous and independent learning, thereby building more active, creative and independent citizens. Based on this, the technology investment in education provides optimistic prospects for the future of e-learning. The number of teachers and researchers that are participating in the technological advancement for the benefit of education is developing rapidly, giving rise to new trends in the learning process, especially on mobile devices, because due to the wide spread of mobile technologies there is a greater tendency to computers be replaced by increasingly smaller devices that permit the mobility, which in this case are mobile devices.

The research developed in question cooperates with scientific knowledge in the sense that this is a representation of technological advances relating to the accessibility tools in distance education mode. In addition, the project also represents an evolution in social issues, as it seeks to disseminate education for persons who are visually impaired, using techniques and e-learning tools through mobile devices, cell phone in that case, contributing thus, the learning of students and interaction in e-learning mode and assumes the role of a viable alternative to eliminate the barriers of access of the visually impaired to Higher Education.

Scientific researches consist of investigations, observations and experiments. Scientific researches try to use these tools to help us to understand the world, satisfying the curiosity and it also help with inventions that bring us comfort. It is hoped through the tool presented in this article a contribution in the construction of a new educational paradigm, both in the quality of education, as the amount of who will receive it.

REFERENCES

- [1]. Agência Nacional De Telecomunicações – Anatel, 2015 Indicators. Available In: [Http://Www.Anatel.Gov.Br](http://www.anatel.gov.br). Last Acess: 19/04/2015.
- [2]. Associação Brasileira De Educação A Distância (Abed), Censoead.Br Analytical E-Learning Report In Brazil, Ed. Pearson, 2011.
- [3]. Attewell J, Mobile Technologies And Learning (Learning And Skills Development Agency, 2005).
- [4]. Belloni M. L., E-Learning (Campinas: Associated Authors, 1999).
- [5]. Botelho F. V. U., Breaking Barriers And Prejudices To E-Learning, 2nd Forum Of E-Learning, 2011.
- [6]. Botelho F. V. U., Breaking Barriers And Prejudices To E-Learning, 2nd Forum Of E-Learning, 2011.
- [7]. Instituto Brasileiro De Geografia E Estatística – Ibge, Censo 2010. Available In: [Http://Www.Censo2010.Ibge.Gov.Br/](http://www.censo2010.ibge.gov.br/). Last Acess: 19/04/15.
- [8]. Mohamed A., Mobile Learning: Transforming The Delivery Of Education And Training Ment (Au Press, Athabasca University, 2009).
- [9]. Musa D. L., Oliveira J. P. M., Intelligent Alerts In E-Learning (Technical Report, Ufrgs 2000).
- [10]. Saraiva, T. E-Learning In Brazil: History Lessons. Brasília, Year 16, N.70, Apr/Jun, 1996.