

Implementação de Módulos Web e Web Service Aplicados no Processo de Ensino-Aprendizagem

Fernando Moreira Borges Júnior¹, Daniela Eloise Flôr¹, Frank Willian Cardoso de Oliveira¹

¹Instituto Federal do Paraná
Paranavaí – PR – Brasil

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

fernando_latrel@hotmail.com, {daniela.flor, frank.willian}@ifpr.edu.br

Abstract: *The popularity and the advancement of mobile devices have brought a number of benefits to its users, with applications that allow you to develop various activities and tasks in any place or environment. Among these applications are those devoted to teaching and/or teaching-learning, in which, aimed at students in the study stimuli of various disciplines, from the principles of gamification and mobile learning. In this context, this paper presents the implementation of a web module to administer courses and their contents, and a web service for offers them to other applications and also provide other services.*

Resumo: *A popularidade e o avanço dos dispositivos móveis trouxeram uma série de benefícios aos seus usuários, com aplicações que permitem desenvolver diversas atividades e tarefas em qualquer lugar ou ambiente. Dentre essas aplicações estão aquelas voltadas ao ensino e/ou ensino-aprendizagem, no qual, visam o estímulo a estudantes no estudo de diversas disciplinas, partindo-se dos princípios da gamificação e do mobile learning. Nesse contexto, este trabalho apresenta a implementação de um módulo web para administrar cursos e seus conteúdos, e um web service para disponibiliza-los a outras aplicações e também disponibilizar outros serviços.*

1. Introdução

O Ensino a Distância (conhecido também como EAD) é uma alternativa de ensino, que possibilita aos discentes horários de estudo mais flexíveis. Tal forma de ensino tem recebido grande atenção no meio didático nos últimos anos, principalmente sendo proporcionada pelos novos artifícios e ferramentas da evolução tecnológica.

Dentre as diversas ferramentas disponíveis no mercado atualmente, o *smartphone* é um dos que mais se destaca, principalmente pelo fato de que pode ser levado para qualquer local e disponibiliza vários aplicativos para as mais diversas atividades.

Aplicativos de cunho didático também vem ganhando grandes proporções, e para que eles funcionem de forma efetiva e versátil, muitas vezes, é necessária a integração de diversas tecnologias, tal como um ambiente web para gerenciamento de informações, sendo elas, cursos ou outras, e um web service para disponibilizar estas informações e outros serviços para os aplicativos.

Este trabalho tem o intuito de desenvolver um módulo web e um web service, no qual, espera-se que, após o seu desenvolvimento e integração com aplicativos e/ou outras plataformas de auxílio didático, discentes se beneficiem, de forma que os mesmos supram suas dificuldades quanto a conteúdos disponibilizados em seus cursos e consequentemente melhorem o desempenho.

1.1. Justificativa

No cenário acadêmico, é comum que alunos possuam dificuldades no entendimento de algumas disciplinas, principalmente os alunos que estão ingressando em uma nova área, e isso, muitas vezes, levando-os a desistir do curso. Essa realidade se agrava nos cursos relacionados à computação, onde muitos nem sequer chegam a completar o primeiro ano letivo. Dessa forma, é necessário buscar alternativas para tentar contornar esse problema.

Ultimamente, os dispositivos móveis vêm sendo muito utilizados pelas pessoas, sendo para consumo de jogos, redes sociais, aplicativos bancários, entre outros, e com isso, também surgiu a Aprendizagem com Mobilidade, conhecida também como Mobile Learning (M-Learning), no qual, sua proposta é diversificar as formas de aprendizagem [Silva, Oliveira e Bolfe 2013], e em conjunto com esse conceito, também ocorreu uma maior aplicabilidade da prática da gamificação, que é uma forma de utilizar conceitos de jogos em atividades rotineiras [Borges et.al, 2013].

A proposta deste trabalho é desenvolver aplicações para o gerenciamento e fornecimento de dados e serviços, sendo eles de cunho disciplinar e administrativo, para outras plataformas (e.g. smartphones, tablets, entre outras).

Com o desenvolvimento destas aplicações, concomitantemente com as plataformas que irão fazer uso dela, espera-se que forneça mais uma forma de auxílio didático aos discentes e que ocorra uma melhor compreensão dos conteúdos abordados em aula aos mesmos, visando também, a diminuição da desistência de seus respectivos cursos.

1.2. Objetivos

O trabalho tem por objetivo o desenvolvimento de uma solução computacional para o gerenciamento de cursos e conteúdos e a sincronização de dados com dispositivos móveis.

Os seus objetivos específicos consistem em:

- Levantamento dos requisitos, modelagem, prototipação e implementação de uma aplicação web para gerenciamento dos conteúdos que serão disponibilizados no aplicativo móvel;
- Estudo e implementação de webservice para o compartilhamento de dados com o app mobile;
- Autenticação do usuário (app mobile) no web service; e
- Sincronização entre app mobile e dados do web service.

1.3. Metodologia

A presente proposta está vinculada a um projeto maior, que como pré abordado nas subseções precedentes, visou-se desenvolver um aplicativo voltado para o ensino, a princípio, para dispositivos móveis com sistema operacional Android. Neste trabalho

foram desenvolvidas partes do projeto, sendo elas, o módulo web e o web service, seguindo o padrão arquitetural MVC¹. O projeto, como um todo, também conta com colaboradores que auxiliam na produção dos conteúdos.

Como a aplicação desenvolvida não parte de uma solicitação de um cliente, a implementação foi realizada com base em aplicativos similares e já de grande sucesso no mercado, dentre eles o Duolingo², que é empregado na aprendizagem de novas línguas.

Para desenvolvimento do trabalho, frequentemente foram realizados encontros e reuniões com todos os envolvidos no projeto para a discussão de soluções que foram necessárias para suprir as necessidades do projeto.

1.4. Apresentação

Na seção 2 é apresentada uma introdução aos conceitos de Gamificação e Mobile Learning e a integração entre eles, e os conceitos de Web Service e da arquitetura REST. Na seção 3 são demonstradas as tecnologias utilizadas para a implementação da proposta. Na seção 4 é descrita a aplicação desenvolvida e suas características. Na seção 5 é exibida a conclusão do trabalho e as propostas para os trabalhos futuros a serem desenvolvidos.

2. Conceitos Básicos

Para a elaboração deste trabalho foram estudados conceitos de gamificação, mobile learning e web services, que serão abordados nas subseções subsequentes.

2.1. Gamificação e *Mobile Learning*

A gamificação consiste em fornecer aos mais diversos ambientes uma forma lúdica, dinâmica e intuitiva de atingir certos objetivos e/ou facilitar processos que são considerados maçantes por um público alvo. Tais objetivos são atingidos através da transformação de rotinas em conceitos utilizados em jogos, que possuem como principais características a competitividade, busca em aprender mais, recompensas, entre outros [Borges et.al, 2013]. Com isso, os indivíduos alvo da gamificação, passarão a ter um maior estímulo ao cumprir tais objetivos e automaticamente terão maior engajamento ao realizar tarefas que anteriormente já realizavam, e isso, de uma forma descontraída e mais eficiente. O conceito de gamificação pode e deve ser utilizado na educação como um instrumento de auxílio nos processos de ensino aprendizagem [Roque et.al, 2013].

Juntamente a gamificação, outro conceito que vem em uma crescente na atualidade é o de Aprendizagem com Mobilidade, mais popularmente conhecido como Mobile Learning (M-Learning), que consiste em uma expansão nas formas de ensino aprendizagem, aplicada através de dispositivos móveis, como smartphones, tablets, entre outros. Ele não visa alterar os atuais processos de ensino aprendizagem, e sim servir como um método auxiliador neste processo [Silva, Oliveira e Bolfe 2013], disponibilizando aos discentes uma forma prática de acesso ao conhecimento, mesmo em locais geograficamente distantes.

Ao integrar os conceitos de M-Learning e Gamificação, pode-se obter o benefício de agilizar e dar praticidade ao ensino, onde poderão ser criados jogos e aplicativos para

¹ Model-view-controller ou Modelo-visão-controle

² Duolingo: <https://pt.duolingo.com/>

dispositivos móveis, com o intuito de passar o conhecimento de uma forma lúdica, dinâmica e intuitiva aos discentes.

Normalmente, as aplicações Mobile Learning são compostas por aplicativos instalados no celular e uma aplicação web que é utilizada para gerenciar os cursos, questões e armazenar um histórico do desempenho dos usuários. Assim, na próxima subseção serão apresentados alguns conceitos de webservice e integração entre plataformas.

2.2. Web Service e Integração de Plataformas

Atualmente, com a expansão na quantidade de plataformas e tecnologias disponíveis para atendimento a diversas áreas do mercado e a necessidade de integração das mesmas, surgiu a necessidade da criação de uma API³ que realizasse essa tarefa, facilitando a comunicação entre elas, mesmo elas sendo desenvolvidas em linguagens diferentes [Moro, Dorneles e Rebonatto, 2011], e assim remover a necessidade de reprogramar uma aplicação ao integrar uma nova plataforma.

Visando esse déficit, surgiram os *Web Services*, no qual, resolvem o problema apresentado anteriormente. Sua função é disponibilizar serviços que são padrão da aplicação e também uma linguagem única de comunicação para todas as plataformas que venham fazer uso de tais serviços. Como artifício/meio de comunicação dessas plataformas com os serviços do web service é utilizado o protocolo HTTP⁴ para transporte das requisições. Uma forma visual de representar a comunicação de plataformas com o web service, pode ser vista na Figura 1.

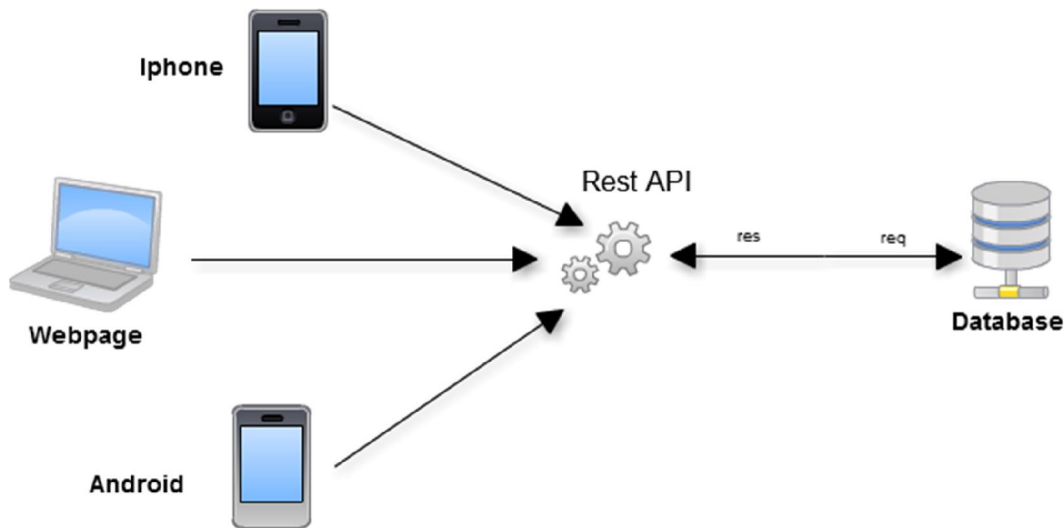


Figura 1 - Cenário de comunicação de plataformas com web service. [Fonte: <http://www.matera.com/br/2012/10/22/como-funciona-um-webservice-rest/>]

Conforme apresentado na Figura 1, os dispositivos com sistema operacional IOS “iPhone”, Android, “Webpage (página web)” são as plataformas cliente que compartilham informações por meio da API Rest. A “Rest API” é o web service da

³ Application Programming Interface ou Interface de Programação de Aplicativos.

⁴ Hypertext Transfer Protocol ou Protocolo de Transferência de Hipertexto.

aplicação (tal nomenclatura “Rest” será abordada mais à frente) e “Database” é o banco de dados único da aplicação.

Dentre as diversas arquiteturas aplicadas aos *Web Services* atuais, existe a REST⁵. A arquitetura REST procura elaborar web services leves, mais flexíveis e eficientes [Campos, 2013], e dentre suas características está a comunicação (provida dos recursos do protocolo HTTP) com os dispositivos em formatos de dados como o JSON, XML, RSS/Atom, entre outros, sendo assim, consequentemente mais versátil.

Esta arquitetura trabalha com requisições de busca, criação, atualização e exclusão de recursos, sendo elas representadas por GET, POST, PUT e DELETE, respectivamente, conforme Figura 2. Tais nomenclaturas são verbos do protocolo HTTP.

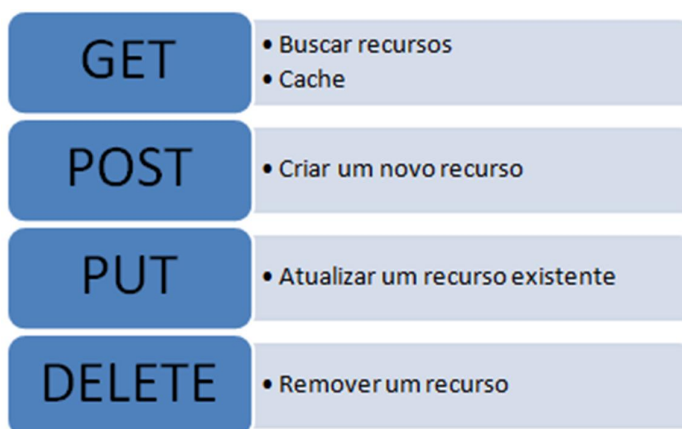


Figura 2 - Verbos para requisições HTTP. [Fonte: <http://zbra.com.br/2011/12/06/android-meets-restful-services/http-verbs/>]

Tais características fazem com que o REST seja uma arquitetura mais simples de ser programada e mais fluida em seu funcionamento [Moro, Dorneles e Rebonatto, 2011, apud Silva, 2008].

Para implementar um webservice REST em Java, linguagem utilizada no desenvolvimento da aplicação deste trabalho, uma das api's disponibilizadas é a JAX-RS⁶, que fornece uma série de recursos ao desenvolvedor, facilitando sua implementação.

3. Tecnologias e Ferramentas Utilizadas no Desenvolvimento

Para a elaboração do sistema foram utilizadas tecnologias e ferramentas, conforme demonstrado na Tabela 1.

Tabela 1 – Ferramentas e tecnologias utilizadas no desenvolvimento do sistema.

Tecnologia - Ferramenta	Descrição
Netbeans	Interface de desenvolvimento integrado de diversas linguagens e plataformas [Netbeans, 2016].
Android Studio	Interface de desenvolvimento integrado para desenvolvimento para plataforma Android [Android Studio, 2016].
MySQL	Banco de Dados [MySQL - Oracle, 2016].

⁵ Representational State Transfer.

⁶ JAX-RS: <https://jax-rs-spec.java.net/>

Apache Tomcat	Container de servlets [Project, A., 2016].
Servlets	Sistema de gerenciamento e processamento de requisições HTTP.
Java Server Faces	Framework de componentes visuais para desenvolvimento web [Caelum, S.D.].
Facelets	Linguagem de descrição de páginas que estabelece sistema de template para páginas web [Oracle, 2013].
Prime Faces	Framework de construção de interfaces web [Caelum, S.D.].
Bootstrap	Framework front-end para criar projetos web responsivos e mobile-first [Mark Otto, A., S.D.].
Hibernate	Framework de Mapeamento de Objeto Relacional no qual abstrai código SQL da aplicação java [Caelum, S.D.].
Spring Security	Framework de estruturamento de autenticação e controle de acesso a aplicações [Spring Security, 2016].
JAX-RS	API para suporte de desenvolvimento de Web Service RESTful [Rubbo, F., S.D.].
Maven	Ferramenta de gerenciamento de projeto e compilação [Porter, B., Zyl, J. e Lamy, O., 2016].
Subversion - Assembla	Sistema de versionamento de projeto.
Bcrypt	Método de criptografia de dados adaptativo ao hardware.
Base64	Método de codificação de dados para transferências via internet.

4. Módulo Web e Webservice Para Gestão de Cursos

A aplicação desenvolvida neste projeto consiste em um módulo web para gestão de cursos e conteúdos, e um *Web Service* para sincronização dos dados para com dispositivos móveis ou outras plataformas que forem necessárias. Mais detalhes desses elementos são apresentados nas subseções seguintes. A arquitetura geral da aplicação é apresentada na Figura 3.

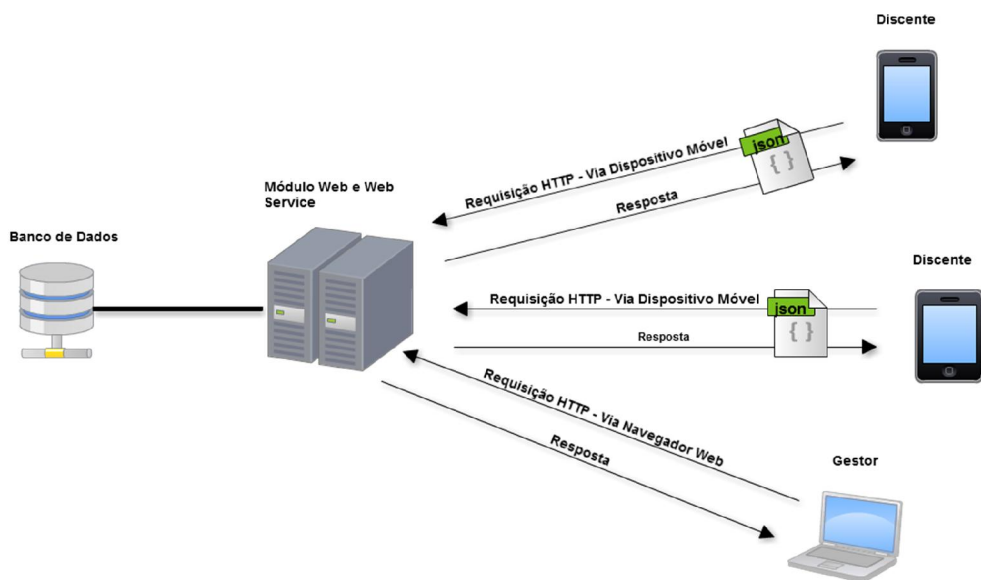


Figura 3 – Arquitetura geral da aplicação.

Conforme representado na Figura 3, os aplicativos instalados nos dispositivos dos discentes realizam requisições ao web service e o gestor do curso realiza acesso (requisições) através de navegador web. Para a definição de como a aplicação deveria se comportar para atender os discentes e os gestores, foram elaborados requisitos não funcionais e funcionais, conjuntamente com os diagramas de caso de uso e de classe.

4.1. Requisitos Não Funcionais

Para o desenvolvimento da aplicação foram levantados os seguintes requisitos não funcionais:

- Módulo Web

- Apenas usuário com privilégio de gestor terão acesso ao módulo de gestão ou painel administrativo.
- O acesso ao sistema será somente via internet e através de navegadores web.
- O módulo será desenvolvido em linguagem Java.
- O módulo apresentará interfaces responsivas.

- Módulo Web Service

- O módulo será desenvolvido em linguagem Java.

4.2. Requisitos Funcionais

Para o desenvolvimento da aplicação foram levantados os seguintes requisitos funcionais:

- Módulo Web

- Autenticar usuários.
- Cadastrar e gerenciar cursos, seções, categorias, lições e questões.
- Manter usuários.

- Módulo Web Service

- Sincronizar dispositivos mobile.
- Autenticar usuários mobile.
- Disponibilizar serviços.

4.3. Diagramas

Utilizando-se do conceito da UML⁷, foram elaborados os diagramas de caso de uso e de classe para o melhor entendimento da aplicação.

4.3.1. Diagrama de Caso de Uso

No diagrama de caso de uso é possível visualizar de forma gráfica os requisitos elaborados para a aplicação, conforme Figura 4.

⁷ Unified Modeling Language ou Linguagem de Modelagem Unificada.

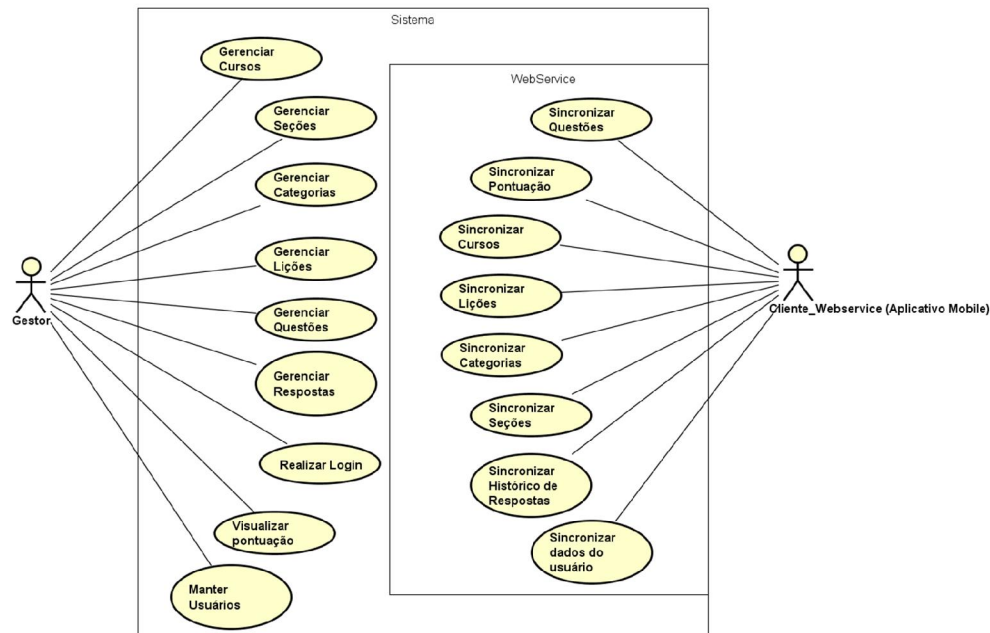


Figura 4 – Diagrama de caso de uso.

4.3.2. Diagrama de Classes

No diagrama de classes é possível visualizar a modelagem de dados, ou seja, a estrutura de classes da aplicação, conforme Figura 5.

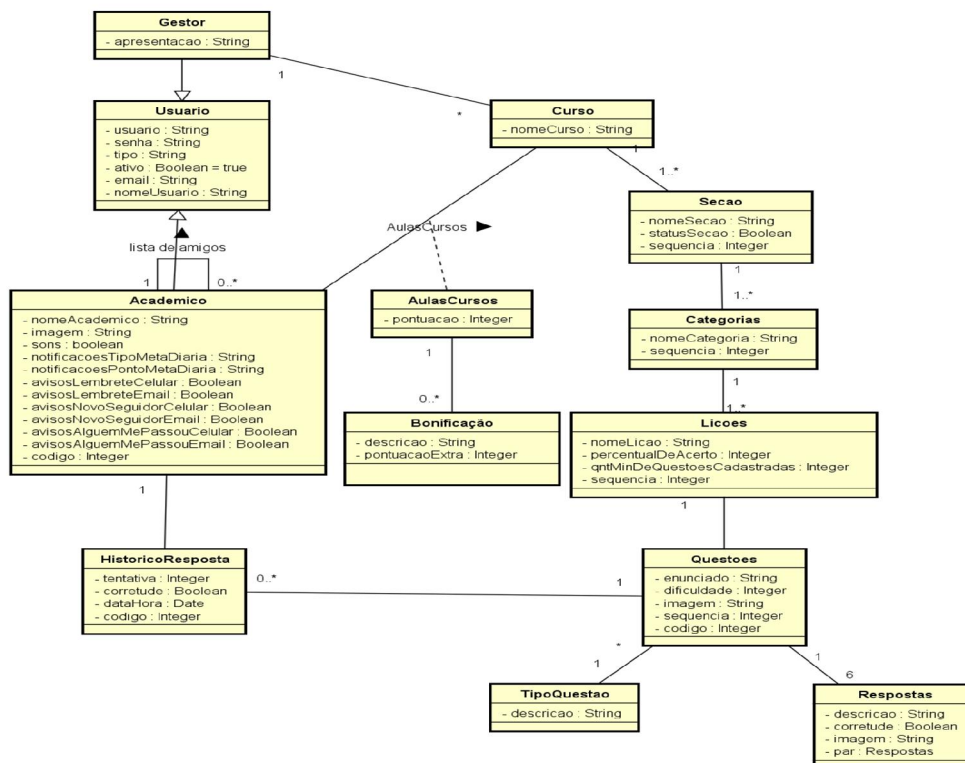


Figura 5 - Diagrama de classes.

4.4. Módulo Web

O módulo web foi criado com o intuito de alimentar e gerenciar os dados e informações que serão disponibilizadas aos dispositivos mobile e/ou demais plataformas que venham a consumir o web service.

A sua interface foi criada com os recursos do Java Server Faces, Prime Faces e Bootstrap. Com o uso de tais recursos deu-se foco a uma interface limpa e agradável, onde o usuário possa trabalhar por um longo tempo sem gerar possíveis cansaços a visão, como pode ser visualizado na Figura 6. Também devido aos mesmos recursos, foi criada uma interface responsiva, no qual, a aplicação pode ser acessada em dispositivos com telas de diversas resoluções sem que ocorra problemas na visualização da mesma, como pode ser visualizado nas Figuras 6, 7 e 8.

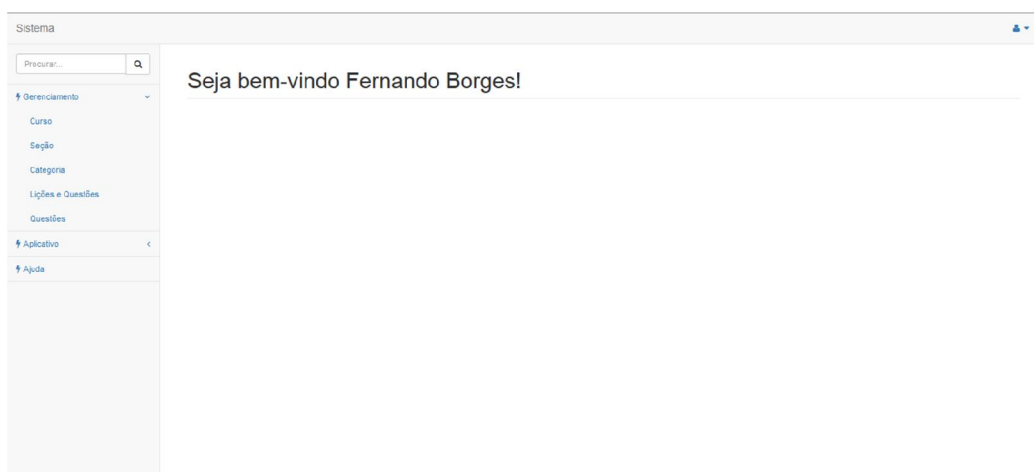


Figura 6 – Tela de boas-vindas do módulo web. Reprodução em uma resolução de 1366 x 768 pixels.

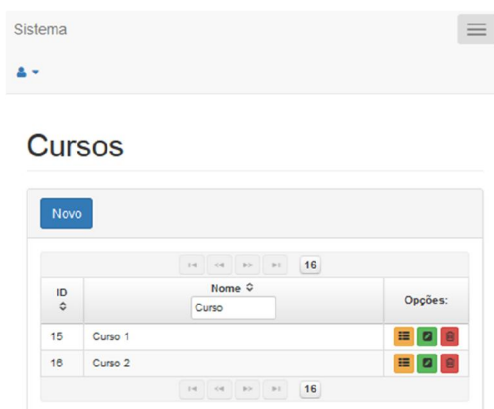


Figura 7 - Tela de cadastro de cursos. Reprodução em uma resolução de 476x640 pixels sem expansão do menu.

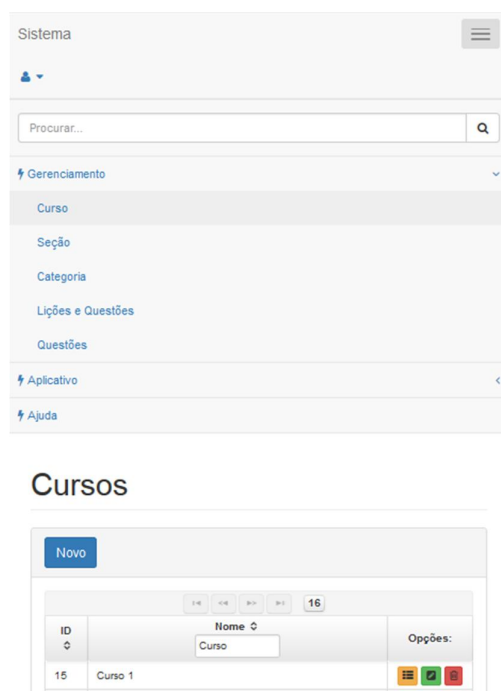


Figura 8 – Tela de cadastro de cursos. Reprodução em uma resolução de 476×640 pixels com expansão do menu.

A segurança do módulo web acaba sendo um dos pontos mais importantes do sistema, no qual, fica a cargo do Spring Security. O Spring Security é um framework fornecido a desenvolvedores para a estruturação da segurança a uma aplicação, gerenciando o acesso e autenticação a mesma [Spring Security, 2016]. Como uma medida a mais de segurança, no módulo web as senhas de acesso a aplicação foram criptografadas conforme o método Bcrypt [Tabela 1], no qual, o Spring Security gerencia a leitura de tais senhas do banco de dados. Também, foi definido um tempo limite de sessão de 20 minutos, expirando a mesma após esse tempo.

A interface web somente será acessada pelos gestores, onde poderão ser gerenciados os cursos, seções, categorias, lições, questões e usuários, tais funções serão definidas a seguir:

- Cursos: os cursos disponibilizados para os usuários;
- Seções: grupamento de categorias. Cada seção poderá apresentar níveis básicos a avançados para os usuários, conforme o seu desempenho no decorrer do curso;
- Categorias: conjunto de lições;
- Lições: conjunto de questões. Para a conclusão de uma lição, é necessário que o usuário tenha acertado a quantidade mínima de questões para a mesma, no qual é definida na hora que a mesma é cadastrada no sistema;
- Questões: questões disponibilizadas aos usuários. Tais questões serão constituídas pelos tipos de verdadeiro ou falso, múltipla escolha, selecionar os pares (para as questões do tipo “pares”, o gestor terá de cadastrar 3 respostas para cada questão);
- Usuários: tela onde o administrador pode visualizar os usuários que estão utilizando o *web service*.

4.5. Web Service

O Web Service (WS) foi construído com base na arquitetura REST e para obter os recursos fornecidos pela mesma, foi utilizada a API JAX-RS, que auxilia no desenvolvimento de tal arquitetura.

O WS tem como objetivo fornecer as funcionalidades/dados para o aplicativo mobile e/ou demais plataformas que venham a consumi-lo, sendo elas, fornecimento de cursos, questões e demais operações que poderão ser realizadas pelo usuário *mobile*.

O fornecimento das funcionalidades está por conta da sincronização dos bancos de dados do cliente e do servidor, onde, automaticamente a aplicação mobile realiza a sincronização das informações, dentre elas os progressos do usuário nos cursos. Essa sincronização ocorre de forma bidirecional, portanto as informações do usuário também ficam armazenadas no servidor, gerando assim uma espécie de backup para as mesmas, e devido a isso, proporcionando ao usuário o acesso a sua conta em dispositivos mobile distintos.

Para a comunicação entre os dispositivos móveis e o WS, todos os dados serão convertidos para o formato JSON⁸, que é uma sintaxe de estruturação de dados em formato textual, leve e simples, permitindo que grandes quantidades de dados sejam enviados, do e para o dispositivo mobile, sem gerar um grande consumo de banda (*internet*), permitindo maior fluidez do aplicativo e do web service.

4.5.1. Autenticação de Requisições

Como uma medida de segurança do WS, para cada requisição do cliente é enviado um código único para realizar a autenticação no WS, desta forma, evitando requisições não autorizadas.

Para requisições de dados públicos, i. e. cursos, questões, entre outros, somente o código único é suficiente para obter tais dados, levando-se em consideração que através da requisição de dados não é possível realizar modificações no WS.

Para requisições em que é necessário o armazenamento e/ou modificações de dados no WS, ou seja, dados que serão enviados do cliente para o WS, o sistema de autenticação funciona de forma diferente, sendo que, na hora em que for necessária a requisição, o código único original é modificado de forma a adicionar informações do usuário, como o nome de usuário e seu identificador/senha. Tais dados, antes de serem mesclados ao código único, são codificados pelo método Base64 [Tabela 1], ficando a estrutura do código conforme a Figura 9.

⁸ JavaScript Object Notation [http://www.w3schools.com/js/js_json_intro.asp]

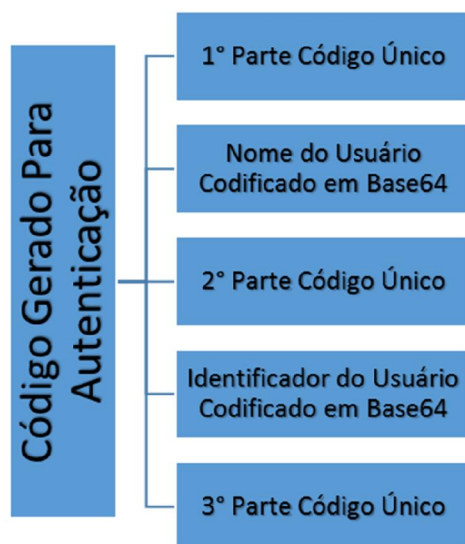


Figura 9 - Estrutura do código de autenticação no web service.

Depois de gerado o novo código, o mesmo é codificado em Base64 e enviado ao WS. No service, é realizado o processo inverso de codificação e mescla, de forma que seja possível juntar as partes do código único original para a primeira etapa de autenticação e a busca do usuário no banco de dados do WS para a segunda etapa. Realizada a autenticação, o WS aceita os dados enviados pelo cliente.

4.5.2. Processo de Cadastro de Usuários

Para que o usuário possa utilizar o aplicativo que irá consumir o WS, será necessário que o mesmo realize um cadastro em seu primeiro acesso, onde serão solicitados dados mínimos, como o nome, e-mail, nome de usuário e senha (ambos para acesso e autenticação).

Ao informar tais dados, eles são convertidos para o formato json, e se o usuário da aplicação possuir acesso à internet, eles são enviados para o WS. O processo de cadastro no WS consiste nas seguintes etapas:

- É verificado se a requisição está partindo de um dispositivo que possua o código único, sendo a mesma aprovada, é realizada a autenticação;
- O e-mail e o nome do usuário são consultados no banco de dados do WS, a fim de verificar se já não há usuário cadastrado com tais informações;
- Estando disponível o cadastro para o usuário que o solicitou, é realizado um processo de criptografia na senha que o mesmo informou, pelo método Bcrypt;
- O usuário é cadastrado no WS; e
- Após o cadastro ser realizado, os mesmos dados são retornados para o dispositivo que o solicitou, no entanto, já com a senha criptografada, desta forma, gravando-os no banco de dados do próprio dispositivo.

4.5.3. Processo de Acesso (Login) Pelo Usuário

Em casos em que o usuário já foi cadastrado no WS, somente será necessário que ele solicite o seu acesso, informando o nome de usuário e senha. Com o recebimento de tais informações no WS, são realizadas as seguintes etapas:

- É verificado se a requisição está partindo de um dispositivo que possua o código único, no qual, conste as informações do usuário (conforme figura 9), sendo a mesma aprovada, é realizada a autenticação;
- O nome do usuário é consultado no banco de dados do WS, afim de verificar se o mesmo já foi cadastrado;
- A senha informada pelo usuário é confrontada com a senha cadastrada no WS, isso, sendo ministrado pelos recursos do Spring Security; e
- Após o nome do usuário e a senha serem validados, os dados do usuário, constantes no WS, são enviados para o dispositivo que o solicitou e são gravados no banco de dados do próprio dispositivo.

4.5.4. Sincronização de Dados

O processo de sincronização de dados consiste no envio de dados do dispositivo do usuário para o WS e o processo inverso para o dispositivo do usuário.

Funcionamento no Web Service

Para que o WS possa identificar quais os objetos e seus respectivos dados que devem ser enviados para os dispositivos móveis, é utilizado um atributo denominado “habilitaSincronização”, que recebe como informação “verdadeiro ou falso”, sendo que, quando verdadeiro, o objeto estará visível para os dispositivos móveis, desta forma, sendo sincronizado. Por padrão, no momento em que o gestor cadastra novos cursos, lições, questões, entre outros, o atributo “habilitaSincronização” recebe “verdadeiro”, disponibilizando o objeto cadastrado automaticamente para todos dispositivos, sendo possível posterior alteração para “falso” no módulo web.

Funcionamento no Aplicativo (Dispositivo Móvel)

Para que ocorra a sincronização no aplicativo, é utilizado um atributo de status nos objetos constantes no dispositivo móvel, sendo eles:

- Inserir: quando algum dado deverá ser enviado ao WS;
Alterar: quando algum dado deverá ser alterado/atualizado no WS; e
- Sincronizado: quando o dado já foi enviado para o WS.

Tais status, permitem que o aplicativo trabalhe sem acesso à internet, a partir do momento em que o usuário já esteja autenticado na aplicação (logado), e no momento em que ele possuir acesso à internet, ocorrerá sequências de requisições partindo do aplicativo para o WS, ocorrendo a sincronização dos dados.

As sequências de requisições são gerenciadas (de forma não transparente para o usuário) pelo próprio aplicativo. Para o envio, ele lista do banco de dados do usuário, todos os objetos com status de inserir e alterar e os armazenam em listas. Tais listas, são lidas e de objeto em objeto, no qual, os dados são convertidos para o formato json, e são enviados para o WS, se a operação ocorrer sem nenhum problema, o objeto recebe o status de sincronizado. Tal processo pode ser visualizado no pseudocódigo⁹ da Figura 10.

⁹ Pseudocódigo: forma simplificada de demonstrar um algoritmo.

```

1  Método de Envio de dados para o service() {
2      listaInserir = lista com objetos com status de inserir;
3      listaAlterar = lista com objetos com status de alterar;
4      se (tamanho listaInserir > 0) {
5          envia listaInserir para o web service;
6          aguarda resposta do web service;
7          se (resposta do web service == ok) {
8              altera status dos objetos para "sincronizado";
9              atualiza objetos no banco de dados do dispositivo;
10         }
11     }
12     se (tamanho listaAlterar > 0) {
13         envia listaAlterar para o web service;
14         aguarda resposta do web service;
15         se (resposta do web service == ok) {
16             altera status dos objetos para "sincronizado";
17             atualiza objetos no banco de dados do dispositivo;
18         }
19     }
20 }

```

Figura 10 - Método de envio de dados ao web service.

Para o recebimento, são solicitados os dados ao WS, e. g. cursos, que após os mesmos serem recebidos, são armazenados em listas, que serão lidas, e de objeto em objeto ocorre o cadastro no banco de dados dispositivo do usuário. Tal processo pode ser visualizado no pseudocódigo da Figura 11.

```

1  Método de busca de dados(){
2      solicita ao web service lista com os objetos;
3      percorrer(lista){
4          busca = verifica se os objetos da lista estão no banco de dados do dispositivo;
5          se(busca = nulo){
6              insere objetos no banco de dados;
7          }
8          se não{
9              atualiza objetos do banco de dados;
10         }
11     }
12 }

```

Figura 11 - Método de busca de dados no web service.

5. Conclusão

Com a evolução tecnológica, surgiram vários meios de realizar diversas atividades nos mais distintos ambientes, e isso, na maioria das vezes sendo proporcionado por dispositivos móveis dos mais variados tipos. Com essa evolução também se fez necessário o desenvolvimento e/ou criação de diversos aplicativos para múltiplos ramos, dentre eles, o educacional. Neste trabalho foram desenvolvidas ferramentas necessárias para o funcionamento de um aplicativo desse ramo, sendo elas um web service e um módulo web para gerenciar cursos, conteúdos e os mais diversos serviços para o aplicativo. Também, foi possível analisar o proveito de tecnologias como web services, a web e dispositivos móveis no contexto do ensino-aprendizagem, principalmente quando se aplica os conceitos de mobile learning e gamificação, que podem ser muito úteis no cenário acadêmico, levando-se em consideração que tais técnicas podem facilitar e auxiliar a vida acadêmica dos alunos.

Espera-se que a aplicação, considerando o projeto como um todo, auxilie os alunos, de forma a impactar diretamente no melhor rendimento e entendimento dos

conteúdos fornecidos pelos seus professores e consequentemente a diminuição do índice de desistência em seus respectivos cursos.

5.1. Trabalhos Futuros

Estuda-se como trabalhos futuros a possibilidade de acesso a aplicação aos alunos via web, podendo os mesmos estudar diretamente no navegador, disponibilizando, assim, mais uma forma de gerenciamento dos cursos fornecidos; e utilizar os dados de redes sociais (Facebook e Google +) do dispositivo do usuário para o seu cadastramento e autenticação no web service.

Referências:

- Android Studio Features | Android Studio (2016).
<https://developer.android.com/studio/features.html>, [Acessado em maio/2016].
- Borges, S.S.; et al. (2013) Gamificação Aplicada à Educação: Um Mapeamento Sistemático. In: II Congresso Brasileiro de Informática na Educação (CBIE 2013) e XXIV Simpósio Brasileiro de Informática na Educação (SBIE 2013).
- Campos, J.; (2013) RESTMB: API RESTful para Android.
- Introdução ao JSF e Primefaces - Lab. Java com Testes, JSF e Design Patterns ([S.d.]).
<https://www.caelum.com.br/apostila-java-testes-jsf-web-services-design-patterns/introducao-ao-jsf-e-primefaces/#7-12-exercicio-opcional-adicione-paginacao-e-ordenacao-a-tabela>, [Acessado em maio/2016].
- Introduction to Facelets - The Java EE 6 Tutorial (2013).
<http://docs.oracle.com/javaee/6/tutorial/doc/giepx.html>, [Acessado em maio/2016].
- Mark Otto, a. ([S.d.]). Bootstrap · The world's most popular mobile-first and responsive front-end framework. <http://getbootstrap.com/>, [Acessado em maio/2016].
- Moro, T.D.; Dorneles, C.F.; Rebonatto, M.T.; (2011) Web services WS-* versus Web Services REST. In: REIC – Revista de Iniciação Científica, volume 11.
- MySQL | O Banco de Dados de Código Aberto Mais Popular | Oracle BR (2016).
<http://www.oracle.com/br/products/mysql/overview/index.html>, [Acessado em maio/2016].
- Porter, B., Zyl, J. e Lamy, O. (2016). Maven – Welcome to Apache Maven.
<https://maven.apache.org/>, [Acessado em maio/2016].
- Project, A. (2016). Apache Tomcat® - Welcome!. <http://tomcat.apache.org/>, [Acessado em maio/2016].
- Roque, A.S.; et al. (2013) Técnicas de Gameficação em AVAs: Um Estudo de Caso no Ambiente Virtual de Aprendizagem Moodle. In: IV Encontro Anual de Tecnologia da Informação. Anais.
- Rubbo, F. ([S.d.]). Construindo RESTful Web Services com JAX-RS 2.0.
<http://www.devmedia.com.br/construindo-restful-web-services-com-jax-rs-2-0/29468>, [Acessado em maio/2016].
- Silva, L.F., Oliveira, E.D., Bolfe, M. (2013) Mobilidade Learning: Aprendizagem com Mobilidade. In: ENEPE: Encontro de Ensino, Pesquisa e Extensão da Unoeste. Anais.

Servlets - Java para Desenvolvimento Web ([S.d.]). <https://www.caelum.com.br/apostila-java-web/servlets/#5-18-discussao-criando-paginas-dentro-de-uma-servlet>, [Acessado em maio/2016].

Spring Security (2016). <http://projects.spring.io/spring-security/>, [Acessado em maio/2016].

Uma introdução prática ao JPA com Hibernate - Java para Desenvolvimento Web ([S.d.]). <https://www.caelum.com.br/apostila-java-web/uma-introducao-pratica-ao-jpa-com-hibernate/#14-13-exercicios-buscando-com-jpql>, [Acessado em maio/2016].

Welcome to NetBeans (2016). <https://netbeans.org/>, [Acessado em maio/2016].