

Plataforma Web para o Armazenamento de Interações Realizadas em Objetos Conectados por IoT

**Trabalho de Conclusão do Curso de
Tecnologia em Sistemas Para Internet**

**Airton da Rocha Bernardoni
Orientador(a): Silvia de Castro Bertagnolli**

¹Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Porto Alegre
Av. Cel Vicente, 281, Porto Alegre – RS – Brasil
conaron@gmail.com, silvia.bertagnolli@poa.ifrs.edu.br

Resumo. *A IoT é utilizada nas mais diversas áreas do conhecimento com o propósito de interligar “coisas” ou objetos do mundo real. Na área de Educação, por outro lado, poucos são os estudos que realmente utilizam a IoT. Este artigo propõe a criação de uma plataforma para armazenar, analisar e processar dados recuperados de dispositivos IoT com uso destinado à educação. O desenvolvimento se dará através da modelagem do sistema utilizando um processo de desenvolvimento de software e uma linguagem para a sua modelagem. Além disso, ele utilizará a plataforma Java, com paradigma orientado a objetos, aliado ao sistema gerenciador de banco de dados PostgreSQL, bem como a integração por WebServices. Espera-se que essa plataforma possibilite ao educador realizar o acompanhamento da evolução cognitiva de estudantes, através de suas interações com os objetos.*

1. Introdução

A informática tem sido utilizada nas mais diversas áreas do conhecimento, visando otimizar os processos e possibilitando o controle e o gerenciamento das informações. Na área da educação muitos trabalhos têm sido desenvolvidos para aprimorar o processo de aprendizagem dos estudantes.

Há várias tecnologias que podem ser utilizadas na sala de aula: slides, blogs, edição colaborativa de documentos através de ferramentas específicas, objetos educacionais, robótica educativa, m-learning (*mobile-learning*), entre outros. No caso deste trabalho propõe-se a criação de uma plataforma para gerenciar dados que possibilitarão realizar o acompanhamento da evolução cognitiva de estudantes que utilizam dispositivos IoT (*Internet of Things*).

Este trabalho está vinculado ao projeto de pesquisa “Ludic-RIO: Rede Interconectada de Objetos Lúdicos e Acessíveis usando IoT” que está sendo desenvolvido no IFRS campus Porto Alegre. Basicamente, o projeto consiste em criar objetos lúdicos que utilizam a tecnologia Arduino ou Raspberry Pi, que se conectarão à rede e enviarão dados das interações para serem armazenados em um servidor. Esses dispositivos serão utilizados por estudantes para o desenvolvimento ou aquisição de alguma

habilidade/conhecimento. A partir da interação realizada pelos estudantes, os dados gerados serão enviados a um servidor central. Cada objeto irá se conectar através da IoT com o servidor, que armazenará as informações obtidas. Após, os professores ou profissionais qualificados poderão acessar as interações e visualizar o percurso cognitivo de cada aluno, de modo a identificar quais pontos devem ser aprimorados ou quais aspectos não foram bem compreendidos.

O potencial das ferramentas considerando seu bom uso e reconfiguração, conforme a situação, depende de um sistema que execute a administração especializada dos dados provenientes das interações. Neste sentido, o sistema aqui proposto será responsável por armazenar os dados enviados, e definir interfaces gráficas com o usuário que possibilitem a visualização destes de forma mais fácil e rápida, gerando também relatórios dos percursos cognitivos, além de gráficos que demonstrem a interação com os objetos lúdicos desenvolvidos.

Para desenvolver o sistema será utilizado um processo de desenvolvimento de software tradicional, vinculando a ele alguns artefatos e diagramas da UML. Além disso, o sistema será implementado utilizando a linguagem de programação Java, com banco de dados PostgreSQL e a tecnologia de Web Services. O uso de todos esses recursos justifica-se pela necessidade de analisar e estabelecer usos para informações enviadas pelos equipamentos que utilizam IoT e estejam inseridos no contexto do projeto de pesquisa acima referido.

As próximas seções do artigo apresentam a fundamentação teórica na seção 2, os trabalhos relacionados na seção 3; a modelagem do sistema na seção 4 e algumas conclusões parciais na seção 5.

2. Fundamentação Teórica

Para fundamentar o desenvolvimento deste trabalho foi necessário realizar um estudo bibliográfico dividido em duas ramificações: uma voltada para a educação, que consiste em definir alguns aspectos relacionados com desenvolvimento cognitivo; e outra relacionada aos aspectos técnicos que influenciam o desenvolvimento deste trabalho: com o IoT, *Web Services* e Reflexão Computacional.

2.1 Desenvolvimento Cognitivo

Segundo argumenta Coll (1992, p. 169-170) "O construtivismo é mudança de visão, pois ele não considera o conhecimento só pelo prisma do sujeito nem só pelo prisma do objeto, mas pela óptica da interação sujeito-objeto".

Para Celia (2003), partindo de uma perspectiva construtivista, é possível afirmar que o desenvolvimento está sustentado pelos princípios: (i) interacionista - a construção do conhecimento passa pela interação com os objetos e com os outros seres; (ii) dialético - o conhecimento parte do movimento e da dinâmica que a criança estabelece com o mundo que cerca; (iii) genético - o conhecimento é formado por uma gênese e constituído por processos; e (iv) estruturalista - o conhecimento se estrutura em estruturas mentais, que se integram a estruturas anteriores.

Com base na obra de Piaget (2011) e de Coll (1992) é possível afirmar que a evolução cognitiva "[...] é ativada pela ação e interação do organismo com o meio

ambiente - físico e social - que o rodeia [...]”, ou seja, há aquisição do conhecimento através das ações e interações que o sujeito realiza com objetos e com o meio em que está inserido, ou em que vive.

Assim, a partir da interação dos sujeitos com os objetos, que estão sendo desenvolvidos pelo projeto de pesquisa, ao qual este trabalho está vinculado, é possível registrá-las em uma plataforma, de modo que as interações realizadas com os objetos possam ser analisadas pelo professor e ele possa identificar, se está acontecendo uma evolução cognitiva do aluno, com base no histórico que o sistema irá disponibilizar.

2.2 IoT

A expressão *Internet of Things* foi utilizada pela primeira vez em 1999, por Kevin Ashton, em uma apresentação para a empresa Procter & Gamble. Apesar de, na época, estar se referindo ao uso da tecnologia RFID (*Radio-Frequency IDentification* - Identificação por Rádio Frequência), cerca de cinco anos após o termo foi relacionado ao uso de redes lógicas por equipamentos autônomos. Este segmento surgiu da evolução de setores tecnológicos como microeletrônica, utilização de sensores, comunicação e sistemas embarcados [Santos et. al 2016].

Equipamentos (como televisores, geladeiras, câmeras de segurança, lâmpadas, termostatos, entre outros) que possam realizar funções básicas de forma independente e conectada através da Internet são identificados como dispositivos físicos na Internet [Buyya e Dastjerdi 2016]. Segundo Cajide (2016), o mercado das IoT tem expandido constantemente. Estimativas apontam que em menos de cinco anos, mais de 50 bilhões de dispositivos estarão on-line.

O crescente aumento desse tipo de dispositivo está relacionado aos ambientes "smart", tais como *smarthouses*, *cities*, *campus*, etc. [Balandin, Andreev e Koucheryav 2014; Santos et. al 2016]. Dentro destes espaços, os dispositivos possuem funções próprias com dependência da internet para sua operação [Buyya e Dastjerdi 2016], como, por exemplo, um irrigador de jardim que ative automaticamente ao receber os dados de um sensor de umidade do outro lado da cidade.

A partir do ranking estabelecido pelo site Libelium é possível perceber que a IoT pode ser utilizada nas mais diversas áreas, tais como: (i) monitoramento de vagas para carros disponíveis para estacionar na cidade; (ii) monitoramento de vibrações e condições dos materiais em construções; (iii) monitoramento de fluxo de veículos e pedestres para detecção de congestionamentos; (iv) adaptação da intensidade da luz em ambientes fechados ou abertos de forma inteligente através de monitoramento climático; (v) placas inteligentes que notificam riscos em função do tempo ou de acidentes que tenham ocorrido na estrada; (vi) controle da emissão de gases tóxicos em fábricas e análise de poluentes emitidos pelos carros na cidade; (vii) monitoramento de volume de lixo em contêineres inteligentes para estabelecer rotas de coleta mais eficientes; (viii) controle dos níveis de poluição marinha; (ix) controle de clima isolado para maximizar a produção de vegetais; (x) estudo das condições meteorológicas para aprimorar a previsão do tempo.

As expectativas sobre o uso acadêmico e industrial da tecnologia IoT cresce de forma expressiva de 2012 até o presente momento. Ela é mencionada constantemente como a tecnologia do futuro e esta concepção se dá, em grande parte, à diversidade de

sua aplicação. Alguns obstáculos, que serão mencionados posteriormente, ainda impedem sua larga propagação em indústrias e instituições de ensino, mas há investimentos em pesquisas para viabilizar o seu uso por pessoas comuns [Buyya e Dastjerdi 2016; Santos et. al 2016].

A utilização em ambientes que não disponibilizam acesso facilitado à Internet, que necessitem de monitoramento constante, que possuam dados cuja coleta possa auxiliar no processo de ensino e aprendizagem caracterizam a IoT como uma tecnologia que mudará a forma de aprender e se apropriar do conhecimento.

A estrutura básica adotada pela tecnologia IoT é composta pelos seguintes itens [Santos et. al 2016]:

- Identificação - é a identidade do equipamento para estabelecer a conexão à Internet. Endereçamento IP, RFID ou NFC podem ser utilizados para este fim;
- Sensores/Atuadores: os sensores permitem que dados sobre o ambiente em que o equipamento se encontra possam ser coletados e armazenados ou transmitidos para servidores. Os atuadores são componentes que interagem com o ambiente de forma ativa, geralmente representados por braços mecânicos;
- Comunicação: o método de conexão e transmissão de dados através da Internet utilizando-se das tecnologias WiFi, Bluetooth, etc.;
- Serviços: a aplicação propriamente dita do equipamento. Destaca-se atualmente o uso de medição de temperatura, umidade, luminosidade, fluxo de fluidos, peso, dentre outros.
- Semântica: a utilização adequada dos dados obtidos pelo equipamento. A forma como esses dados são providos através de um padrão ou modelo, conhecido como metadados, geralmente através de formato XML.

Apesar do crescente número de possíveis aplicações da tecnologia atualmente, há fortes empecilhos a sua implementação industrial e acadêmica. Algumas das principais restrições tem relação com a segurança de dados, consumo de energia e endereçamento.

Para equipamentos alocados em ambientes inóspitos ou em lugares onde não possua infraestrutura com abastecimento de energia, o consumo da mesma é avaliado em cada estrutura montada considerando seus componentes e frequência de uso, já que a alimentação terá de ser por baterias. Além disso, ambientes que não disponibilizam acesso à rede WiFi podem apresentar problemas para a implantação de tecnologias IoT [Buyya e Dastjerdi 2016].

Outro problema que surge na utilização de IoT diz respeito à segurança dos dados. Embora os critérios de segurança aplicados sejam os mesmos de uma rede de computadores convencional, os recursos de hardware para tal são reduzidos. A capacidade de criptografar os dados é proporcional à capacidade do processador e memória em uso. Isso, aliado ao volume de dados que depende de tráfego e consumo de energia, acaba impedindo a proteção adequada dos dados gerados e transmitidos.

2.3 Web Services

Os Web Services são uma tecnologia para comunicação entre um servidor e um sistema cliente utilizando a Web. Essa comunicação é realizada através de uma interface padronizada seguindo a especificação XML (*eXtensible Markup Language*) e o protocolo HTTP (*HyperText Transfer Protocol*). Um serviço detalha a sua interface de comunicação usando o XML através do documento chamado WDSL (*Web Service Description Language*), cujo padrão é definido pela W3C (*World Wide Web Consortium*) [Lecheta 2015].

Além da compatibilidade multiplataforma o WebService garante a segurança do banco de dados armazenados no servidor através da conversão e do filtro das informações disponibilizadas. O acesso ao banco de dados é realizado somente pelo servidor, sem o envio de uma instrução direta do sistema cliente. Ao implementar o serviço, define-se quais informações serão recebidas pelo WebService na requisição, como o servidor irá tratá-las e o que irá realizar com estes dados. Por exemplo, para fazer um *select* em uma tabela do banco de dados, o WebService pode ser configurado para receber apenas um parâmetro de filtro específico e o nome da tabela. Eles serão enviados pelo sistema cliente através de uma estrutura XML e tratados de forma segura pelo servidor, antes de realizar efetivamente a operação de *select* no banco, essa solução pode prevenir o uso de *SQL Injection*.

Entre as tecnologias utilizadas para implementar e implantar *Web Services* têm-se [Lecheta 2015]: o SOAP (*Simple Object Access Protocol*) e o REST (*Representational State Transfer*). O primeiro é um protocolo de transferência que usa o formato XML. Como mencionado anteriormente, recomenda-se o uso de um documento WSDL para a descrição da estrutura que atende o WebService. Embora o XML demande de um interpretador que analise a estrutura de árvore, o que consome desempenho, ele possibilita a transferência de dados mais complexos, como a representação de uma lista de registros com camadas e subcamadas. Já o REST é uma alternativa ao SOAP desenvolvida para ser mais simples. Como não possui a obrigatoriedade de se comunicar através de XML, mas apenas pelo protocolo HTTP, dispensa o rigor da estrutura exigida e pode transmitir metadados em diversos formatos mais leves de interpretação, tais como: CSV (*Command Separated Value*), JSON (*Java Script Object Notation*) e RSS (*Really Simple Syndication*). A comunicação poderá se dar apenas com a URL (*Uniform Resource Locator*), tendo pré-definido seu retorno no servidor.

2.4 Reflexão Computacional

A reflexão computacional é a capacidade de um programa obter informações ou modificar o seu estado ou comportamento em tempo de execução, sendo que essa característica é chamada de introspecção. Utilizando-se essa capacidade é possível criar um "código que lida com uma classe cuja estrutura ele não conhece" [Guerra 2014].

A reflexão computacional usa o conceito de metadados para conseguir extrair informações dinâmicas sobre os objetos e suas classes. As maiores desvantagens de se utilizar essa técnica são o tempo de execução e a complexidade do código que aumentam [Guerra 2014].

No caso da linguagem de programação Java a reflexão é definida através da API (*Application Programming Interface*) `java.lang.reflect` que possui um conjunto de classes e métodos que viabiliza descobrir as informações durante a execução de um programa.

A próxima seção apresenta uma lista de trabalhos relacionados com IoT e vinculados à área de Educação, que é o foco deste trabalho.

3. Trabalhos Relacionados

A partir da análise da literatura foram encontrados diversos trabalhos que descrevem dispositivos físicos baseados em sensores, que são aplicados em outras áreas e não na área de educação que é o foco deste trabalho. Desse modo, nesta seção serão descritos alguns trabalhos que abordam a temática de IoT vinculada à área de educação.

3.1 Trabalho 1 - Internet das Coisas: tudo o que pode ser conectado, será conectado

Wadewitz (2016) em sua pesquisa “Sizing Up the Internet of Things”, vinculada à CompTIA, estabelece uma posição de mercado para o uso de equipamentos de IoT em diversas áreas. Embora não aborde o uso desta tecnologia na área de educação, o trabalho apresenta a estimativa atual de investimento no uso destes equipamentos.

Uma aplicação de mercado que é apontada é o desenvolvimento de aplicativos móveis, visto que esta área está em constante expansão. Ainda há muitos aspectos pendentes para normatização do uso de IoT, tais como governança e segurança, que indefinem o tempo em que o mercado a assimilará de fato. Com relação a isso, oportunidades estão sendo aprimoradas e o uso de publicidade já está em prática para adequar o público à realidade que se apresenta - o uso massivo de ferramentas IoT no cotidiano.

3.2 Trabalho 2 - A internet de todas as coisas e a educação: possibilidades e oportunidades para os processos de ensino e aprendizagem.

O uso de ferramentas IoT proporciona possibilidades educacionais vastas como personalização dos ambientes e recursos didáticos de alunos com necessidades especiais com base em suas peculiaridades cognitivas. Barros e Souza (2016) apontam que várias áreas podem ser aprimoradas com IoT, entre elas a Educação. Eles ainda argumentam que a IoT pode ser vista como uma rede de redes.

Esse mesmo trabalho aponta que a IoT permitirá modificar as formas como a interação e a comunicação ocorrem, ou seja, ambas podem utilizar o tempo real ou não, além de disponibilizar para a sociedade informações interligadas e interconectadas. Um dos grandes desafios identificados têm relação direta com as questões de segurança da comunicação e dos dados.

3.3 Trabalho 3 - *How IoT in education is changing the way we learn* (Como IoT na educação está mudando o modo como aprendemos)

A Internet se faz presente de forma cada vez mais impactante nas instituições de ensino, sendo que um exemplo disso é o sistema de e-learning (EaD através de plataformas online) que está ganhando espaço em cursos de diversos níveis de ensino [Meola 2016].

O aumento crescente de dispositivos que utilizam IoT, em diversas situações cotidianas, encaminha vastas possibilidades para o uso dessa tecnologia na educação. Percebe-se que, nos últimos anos, tem ocorrido uma grande mudança na forma de ver e compreender a aprendizagem, onde alunos conseguem executar tarefas auxiliados por ferramentas que dispensam a ação humana e professores que, caso existam, possam avaliá-los em tempo real sem qualquer contato presencial [Meola 2016].

3.4 Trabalho 4 - *IoT in education: the internet of school things* (IoT na educação: a internet das coisas escolares)

Conforme argumenta Augur [2016] a educação não se resume às aulas. Há um grande espaço de tempo destinado à avaliação, preparação de aulas e construção do perfil dos alunos. A IoT possibilita a automatização de grande parte dessas tarefas. Os alunos podem colher informações de espécimes de ambiente selvagem, cadernos podem ser digitalizados e avaliados sem o intermédio direto do educador. Embora pareça uma visão futurista da aplicação da IoT em ambiente educacional, esta ainda é simplória considerando que seu uso depende da criatividade dos educadores e da disponibilidade de infraestrutura adequada.

3.5 Trabalho 5 - *The Connected School: How IoT Could Impact Education* (A Escola Conectada: Como a IoT Impactará a Educação)

Segundo Cajide [2016] o uso de IoT na educação seguirá duas linhas: a aceleração no aprendizado dos alunos e o aumento na eficiência de como os professores poderão executar seu trabalho. Ferramentas com funções simples em conectividade incrementarão os métodos de estudos de estudantes e a comunicação entre professores e pais não dependerá somente de contato físico.

Com base na fundamentação teórica realizada e no levantamento de trabalhos relacionados foi proposta a plataforma descrita na próxima seção.

4. Proposta Metodológica

Para o desenvolvimento deste trabalho foi necessário selecionar um processo de desenvolvimento, de modo a estabelecer quais passos e artefatos de software seriam necessários para desenvolver a solução. Assim, optou-se por utilizar alguns passos que são utilizados para o desenvolvimento de diversos tipos de software: levantamento de requisitos; análise de requisitos; projeto e implementação com os testes.

4.1 Levantamento e Análise dos Requisitos

O foco dessa etapa consiste em compreender o problema, levantando e priorizando as necessidades e funcionalidades do sistema. Para realizar o levantamento dos requisitos foi necessário analisar o funcionamento dos objetos IoT.

Assim, pegou-se, por exemplo, o objeto "Urso", que compreende um urso de pelúcia que contém uma tela onde o estudante interage, seguindo uma sequência de passos, que compreendem um jogo. Esse jogo é composto por fases e níveis cujo propósito é auxiliar na aquisição do conhecimento matemático e linguístico. Por exemplo, a Figura 1 ilustra a fase inicial para a composição de palavras, onde o estudante se

identifica usando uma TAG RFID (cartão A) e deve completar a palavra com a letra que está ausente. Essas letras fazem parte de um alfabeto que foi criado usando cartões RFID. Ao completar a palavra e passar para a próxima fase o objeto envia para a plataforma o número da TAG do aluno, a fase que ele realizou, o número de tentativas para completar a palavra e o número de tentativas incorretas até completar a palavra. Todos esses dados são enviados para a plataforma que será desenvolvida usando a conexão com a rede.

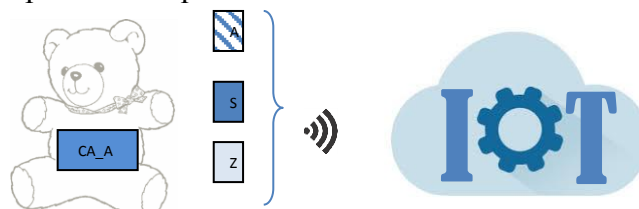


Figura 1. Representação de um Objeto IoT

A partir dessa análise chegou-se à conclusão de que a plataforma deve possibilitar mecanismos que permitam ao educador acompanhar a evolução do estudante a partir das suas interações com os objetos e que alguns cadastros serão necessários de modo a permitir esse acompanhamento.

Para a modelagem dos requisitos optou-se pela utilização do diagrama de casos de uso (Figura 2), de modo a tornar visíveis as funcionalidades que farão parte da plataforma. O cadastro dos educadores e dos objetos IoTs será realizado através de uma interface gráfica com o usuário. Desta forma, os educadores poderão cadastrar estudantes para, futuramente, analisar as interações dos mesmos e extrair relatórios e gráficos específicos. Caso desejem acompanhar o desenvolvimento de todos os estudantes de uma turma, poderão cadastrar a turma e vincular os estudantes à ela. Cabe observar que o usuário administrador pode utilizar o sistema como educador, tendo acesso a todas as funcionalidades disponibilizadas. A Figura 2 ilustra os usuários do sistema, bem como as funcionalidades as quais eles possuem acesso.

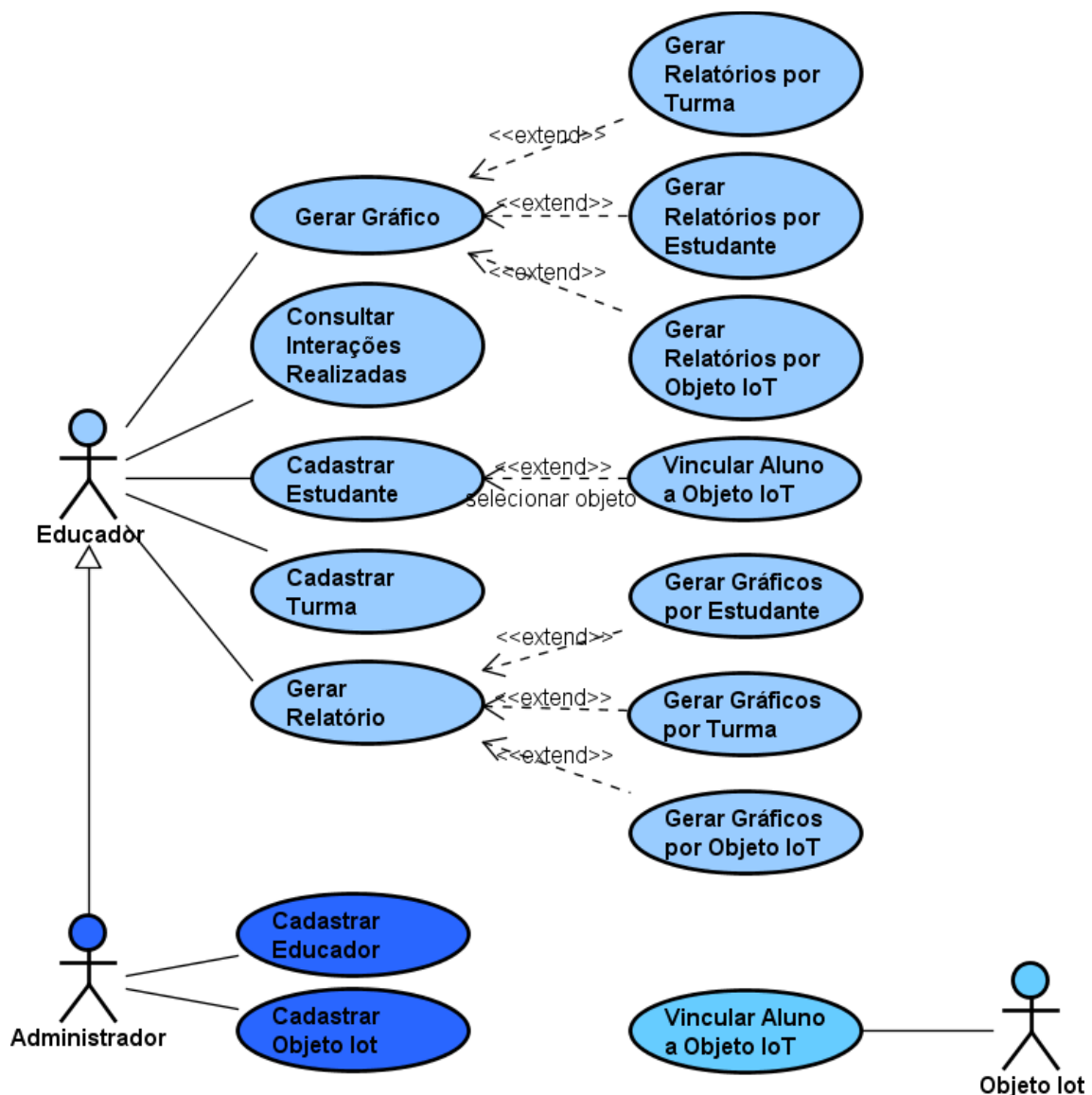


Figura 2. Diagrama de Casos de Uso da Plataforma Proposta

4.2 Análise e Projeto

Nessa fase devem ser levados em consideração projeto detalhado com diagramas, a linguagem de programação e o Sistema Gerenciador de Banco de Dados (SGBD) que serão utilizados, os padrões de *front-end* ou de interface gráfica com o usuário.

A partir da análise dos requisitos observou-se a necessidade de criar um Web Service. Através do Web Service os objetos IoT enviarão dados das interações com a identificação do estudante para tratamento no servidor. Para garantir a heterogeneidade das informações, está utilizada uma estrutura XML para o consumo de dados. Como os objetos IoT são diferentes e enviarão dados diversos, está utilizado o padrão SOAP sobre o protocolo HTTP, onde os dados enviados serão encapsulados com JSON.

Os dados obtidos a partir dos objetos IoT são processados de forma dinâmica. A solução escolhida para gerenciar as relações de forma isolada foi a definição de padrão

de relações para cada objeto IoT, composto previamente de identificação em tabela comum, tal como lista central; tabela com níveis e tabela com fases possíveis. A administração das relações é realizada através da aplicação pelo servidor também de forma dinâmica. A aplicação se encarrega de criar as relações e administrá-las conforme a estrutura de dados fornecida pelo objeto IoT em questão. Como a linguagem para o desenvolvimento do sistema é Java, o recurso para executar a persistência usando esse modelo consiste no uso de reflexão computacional. Através deste mecanismo, serão instanciadas classes, parâmetros e métodos determinados em tempo de execução, permitindo que o objeto (no código) seja manipulado conforme a identificação fornecida pelo objeto IoT ao transmitir dados para o Web Service.

O diagrama de classes (Figura 3) foi utilizado para mapear os recursos estruturais de código vinculados a esta fase do processo de desenvolvimento. Esse diagrama leva em consideração o projeto do Webservice e alguns aspectos da reflexão computacional.

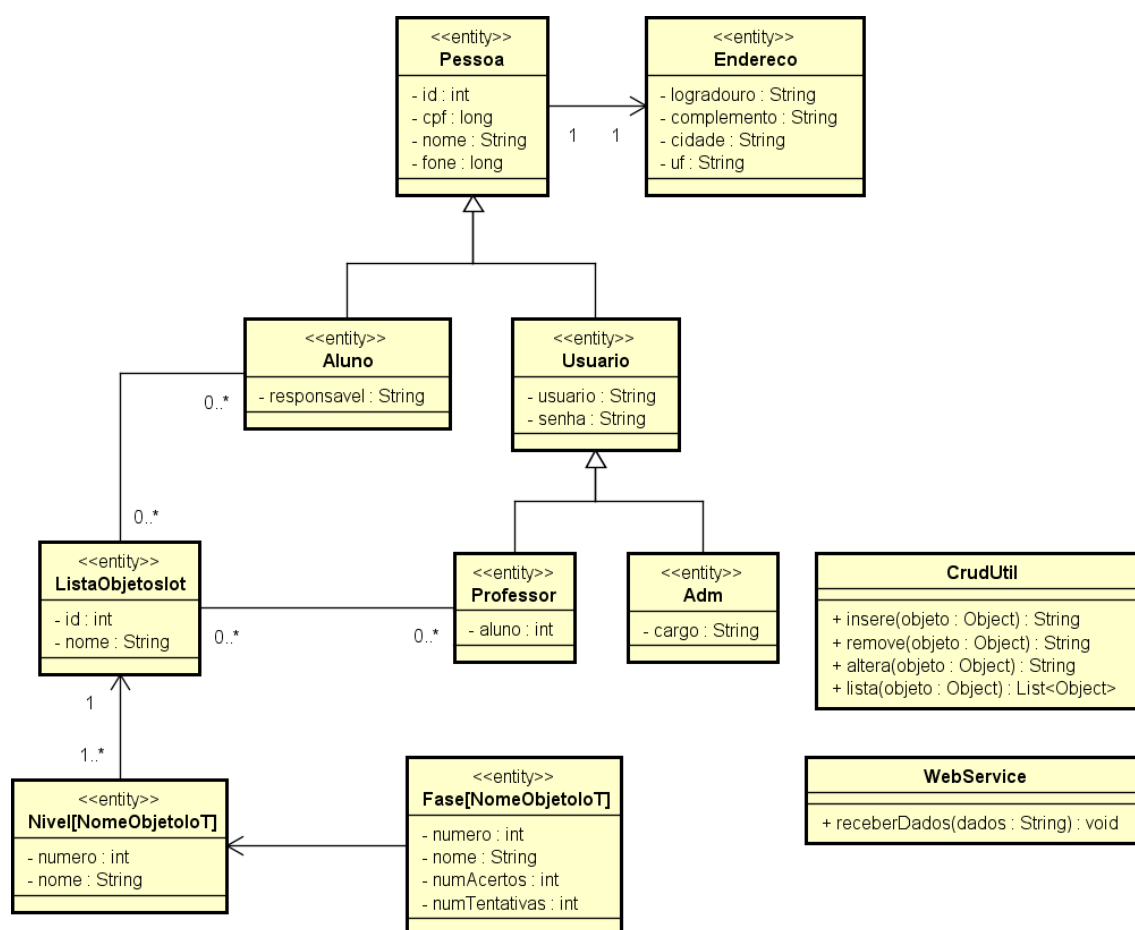


Figura 3. Diagrama de Classes da Plataforma Proposta

A partir do diagrama de classes é construído o código, bem como são realizados alguns testes do sistema. Alguns detalhes relacionados com a implementação e testes encontram-se descritos na próxima seção.

4.3 Implementação e Testes

Conforme já mencionado, a implementação consiste na codificação das soluções

definidas nas etapas anteriores utilizando-se as tecnologias selecionadas. Para o desenvolvimento da solução proposta por este trabalho foram selecionadas as seguintes tecnologias: linguagem de programação Java, os frameworks Hibernate e PrimeFaces com JSF. Já com relação ao banco de dados optou-se por utilizar o PostgreSQL. Apesar do MySQL obter excelência em agilidade e figurar a maior parte dos recursos que estão disponíveis no PostgreSQL, este último possui uma robustez superior em relação ao tratamento de dados com variedade maior de tipos. Como a diferença de velocidade é imperceptível em pequena escala e o processo de implantação é muito similar, o uso do PostgreSQL facilitará a evolução da aplicação quando o volume de dados atingir valores expressivos.

Inicialmente, foi necessário configurar as ferramentas para o desenvolvimento do sistema. Iniciou-se a configuração pela IDE NetBeans, após foram configurados o framework Hibernate e a ferramenta Maven. Abaixo, é possível encontrar alguns problemas encontrados na configuração inicial:

1. Framework Hibernate - construção automatizada se mostrou ineficiente, pois as versões das bibliotecas estavam desatualizadas. Poucos recursos funcionaram corretamente com as importações sugeridas pela documentação do próprio Hibernate;
2. Maven - ao utilizar a abordagem de bibliotecas modificada para o uso de dependências gerenciadas pelo repositório Maven, esta abordagem funcionou. A configuração dos frameworks passou a ser manual para não utilizar as versões sugeridas pelo NetBeans. Um teste preliminar de adequação de framework pela própria IDE, mesmo com Maven, registrava a versão depreciada das bibliotecas.
3. Abandono da unidade de persistência declarada (utilização do arquivo persistence.xml) para a utilização do arquivo de configuração específico do Hibernate com sua estrutura de TAGs específica. Somente assim, abordagem via construção de sessão e não via gerenciamento de entidade passou a funcionar.

Ao concluir essa configuração inicial, procedeu-se com a implementação e alguns testes, que foram realizados em paralelo com a etapa de codificação.

Para a realização de todos os testes foi definido dentro do projeto um pacote cujo nome é “teste”, o qual é composto por classes que realizam a execução de métodos verificando a passagem de parâmetros e seu retorno. Procurou-se cobrir todos os casos hipotéticos concebidos pelo desenvolvimento.

Devido às restrições de infraestrutura onde o projeto é desenvolvido, foi necessário dividi-los em sub-projetos, cada um com funções específicas (por exemplo, construção de sub-projeto para testar o polimorfismo) a serem testadas, e que depois serão incorporadas ao sistema. De modo a controlar e registrar o andamento do projeto optou-se por utilizar o GitHub, que possui o versionamento do projeto principal.

Ao definir a estrutura do projeto, passou-se para os testes, os quais encontram-se descritos abaixo:

1. Construção de uma classe de conexão e uma classe de operações básicas de manipulação de dados (CRUD - *Create, Retrieve, Update e Delete* / inclusão, consulta, atualização e exclusão) realizada para início de testes funcionais assim como uma classe modelo inicial (neste caso, optou-se pela classe Professor devido a sua simplicidade). Os métodos funcionaram corretamente com o uso da reflexão computacional, exceto o de consulta nomeada, pois o texto referia-se especificamente à classe utilizada;
2. Modificação da abordagem da consulta “nomeada” através de padrão desenvolvido na construção da classe modelo com a inclusão de método com retorno de String que simulasse a sintaxe do Hibernate para consulta. O método foi chamado de “toQuery()” em referência aos métodos padrões sobrescritos como, por exemplo, “toString()”;
3. Criação de segunda classe modelo verificar se o CRUD reflexivo iria ou não funcionar. Após o sucesso desse teste foi realizado o acoplamento da classe de conexão com a de manipulação para agilização do processo de construção e finalização de sessão;
4. O tratamento de exceções do Hibernate com as anotações referentes a restrições dos campos nas tabelas se mostrou insuficiente, pois ainda trata de forma abrangente muitas exceções de origem diferentes;
5. Mapeamento ORM (*Object Relational Mapping*) - Na tentativa de utilização de polimorfismo com as anotações utilizadas pelo Hibernate a abordagem *Joined* foi abandonada, com a intenção de reduzir o consumo de recursos no acesso ao banco de dados. Então, passou-se a utilizar a abordagem *MappedSuperclass* e *table_per_class*, as quais também não funcionaram. Após análise do próprio Hibernate, percebeu-se que as classes importadas do mesmo não atendiam esta funcionalidade corretamente, então passou-se a utilizar as classes do JPA através do Hibernate o que viabilizou o funcionamento do polimorfismo.
6. Como a utilização de construção de classes de entidade dinâmicas será o componente fundamental deste trabalho, foi removido o mapeamento de uma das classes para analisar o comportamento no acesso ao banco pelo Hibernate, o que não se mostrou promissor. A abordagem de mapeamento estático de classes foi abandonada completamente e é realizada em tempo de execução através da classe de manipulação no momento da construção da sessão também com o uso de reflexão computacional;
7. Construção de duas páginas para inserção de dados de testes através de JSF em Primefaces. Foram incluídas anotações necessárias para gerenciamento de Bean a fim de que o funcionamento do framework fosse possível. Os testes foram melhores do que com o Hibernate; Carregamento, construção e compilação de classe realizado e testado de forma simples (sem construção por ferramenta interativa apenas informação de String) em projeto paralelo, sendo que seu uso só foi possível através de reflexão.
8. Iniciado recurso de construção de classe “construtora de classes” para estruturar as classes pertinentes a ObjetosIoT consumidores do projeto, que ainda está em desenvolvimento

Observa-se que para o desenvolvimento deste trabalho, optou-se por utilizar retorno do tipo *String* em quase todos os métodos da classe de manipulação, pois isso facilita o tratamento do retorno.

Além desses testes, um serviço inicial já foi desenvolvido para conhecer a tecnologia e analisar o seu funcionamento. Cabe observar que está sendo utilizado o SOAP para a implementação do serviço, de modo a possibilitar a transmissão de dados de diversas estruturas por objetos em plataformas diferentes, assim como bibliotecas e APIs que facilitem o desenvolvimento e garantam qualidade ao mesmo.

Quando os testes de unidade e integração forem concluídos serão realizados os testes do sistema testando as funcionalidades de cada módulo, visando produzir um software com mais qualidade.

A técnica de reflexão computacional no Java consiste em acessar, manipular e utilizar parâmetros, métodos e características de objetos de modo dinâmico em tempo de execução. Para melhor compreender como ela foi utilizada foi elaborada a próxima seção.

4.4 Reflexão Computacional

Para criar em tempo de execução as tabelas e classes que possuem relação com os objetos IoT (ObjetosIoT) foi criada a classe CrudUtil, que é responsável pela geração dinâmica em tempo de execução.

Por exemplo, no Quadro 1, linha 22, é informado como parâmetro um objeto do tipo Class (classe) para início da sessão com o mapeamento direcionado a esta classe. Desta forma, é possível mapear dinamicamente a classe passada como parâmetro em tempo de execução. Na linha 24, utiliza-se a classe Configuration do Hibernate para iniciar a configuração paralela (apenas durante o período de vida do objeto sessão) do arquivo “hibernate.cfg.xml”. Na linha 25, é mapeada a classe através do método “addAnnotatedClass(Class object)” e é formalizada a configuração com o método configure(). A partir deste ponto, a classe passada como parâmetro já se encontra mapeada e configurada para operações como o banco de dados.

Quadro 1. Usando Reflexão Computacional (definição de elementos)

```
14. public class CrudUtil {
15.     //...outrasdefinições
22.     private void inicia_sessao(Class classe) {
23.         try {
24.             Configuration config = new Configuration();
25.             config.addAnnotatedClass(classe);
26.             config.configure();
27.             factory = config.buildSessionFactory();
28.             sessao = factory.openSession();
29.             sessao.beginTransaction();
30.         } catch (HibernateException e) {
31.             System.out.println("Criação da sessão falhou. "+ e);
32.         }
33.     }
```

Considerando, por exemplo, a listagem de registros (Quadro 2), pode-se observar o uso da reflexão na linha 127, onde é obtida a classe do objeto parâmetro. Utiliza-se esta classe para instanciar um método definido no padrão do projeto chamado “toQuery” (linha 128) que retorna a *namedQuery* específica da classe. Após a consulta realizada, é

retornada uma lista genérica (cada registro é instancia de um *Object*) para tratamento exterior.

Quadro 2. Usando Reflexão Computacional (listagem de registros do BD)

```
123. public List<Object> lista(Object origem) {
124.     List<Object> lista = null;
125.     inicia_sessao(origem.getClass());
126.     try {
127.         Class classe = origem.getClass();
128.         Method toQuery = classe.getDeclaredMethod("toQuery");
129.         String namedQuery = (String) toQuery.invoke(origem);
130.         lista = sessao.createQuery(namedQuery).list();
131.         sessao.flush();
132.         if (lista == null) {
133.             lista = new ArrayList();
134.             lista.add("Registro não encontrado");
135.         }
136.         sessao.flush();
137.     } catch (HibernateException e) {
138.         lista = new ArrayList();
139.         lista.add("Registro não encontrado");
140.     } catch (NoSuchMethodException | SecurityException |
141.             IllegalAccessException | IllegalArgumentException |
142.             InvocationTargetException ex) {
143.         Logger.getLogger(CrudUtil.class.getName()).log(
144.             Level.SEVERE, null, ex);
145.     } finally {
146.         encerra_sessao();
147.     }
148.     return lista;
149. }
150. }
```

O uso da reflexão computacional viabilizou a geração dinâmica de vários objetos e elementos que serão essenciais para dar continuidade à implementação. A partir desse momento todos os testes parciais servirão de ponto de partida para o desenvolvimento da plataforma em sua versão final.

5. Considerações Finais

A maior dificuldade encontrada, na primeira etapa do desenvolvimento deste trabalho, foi encontrar trabalhos semelhantes ao aqui proposto. Outro ponto que dificultou o desenvolvimento inicial desse trabalho foi compreender a origem dos dados, pois como eles são provenientes de objetos IoT que estão sendo desenvolvidos por bolsistas vinculados ao projeto, ficou difícil entender como seriam enviados os dados e como eles se integrariam à plataforma.

Com o andamento deste trabalho algumas dificuldades de configuração das ferramentas e da infraestrutura foram encontradas. O uso da reflexão computacional mostrou-se promissor e adequado para a produção de objetos dinâmicos em tempo de execução.

O trabalho terá continuidade seguindo os passos estabelecidos pelo cronograma esquematizado pelo Quadro 3.

Quadro 3. Cronograma Próximas Etapas

	ago	set	out	nov	dez
Construção do Web Service e testes iniciais	x				
Construção das classes e testes	x	x	x		
Mapeamento do ER		x	x		
Definição da solução com Reflexão computacional		x	x	x	
Desenvolvimento do Front-End	x	x	x	x	
Testes da plataforma	x	x	x	x	
Testes de integração	x	x	x	x	
Seminário de andamento			x		
Entrega da versão Final do TCC					x

Referências

- Augur, H. (2016), "IoT in education: the internet of school things", <https://www.ibm.com/blogs/internet-of-things/iot-education/>. Acesso em: 23 de junho de 2017.
- Barros, A. G. e Souza, C. H. M. (2016), "A internet de todas as coisas e a educação: Possibilidades e oportunidades para os processos de ensino e aprendizagem.", <http://revista.srvroot.com/linkscienceplace/index.php/linkscienceplace/article/view/249>. Acesso em: 17 de junho de 2017.
- Buyya, R.; Dastjerdi, A. V. (2016), Internet of Things: principles and paradigms. Cambridge: Elsevier.
- Cajide, J (2016), "The Connected School: How IoT Could Impact Education", http://www.huffingtonpost.com/jeanette-cajide/the-connected-school-how-b_8521612.html. Acesso em: 23 de junho de 2017.
- Balandin, S.; Andreev, S.; Koucheryavy, Y. (2014), "Internet of Things, Smart Spaces, and Next Generation Networks and Systems". In: 14th International Conference, NEW2AN 2014 and 7th Conference, SMART 2014, St. Petersburg, Russia, August 27-29, Proceedings.
- Celia, L. S. (2003), "Aquisição e desenvolvimento infantil (0-12 anos): um olhar multidisciplinar". Porto Alegre: EDIPUCRS.
- Coll, C. (1992), "As contribuições da psicologia para a educação: teoria genética e aprendizagem escolar". In: Leite, Luci Banks (Org). Piaget e a escola de Genebra. São Paulo: Cortez.
- Guerra, E. (2014), "Componentes reutilizáveis em Java com reflexão e anotações". Rio de Janeiro: Casa do Código.
- Lecheta, R. R. (2015), "Web Services RESTful: Aprenda a criar web services RESTful em Java na nuvem do Google". Rio de Janeiro: Novatec.

- Libelium. "Top 50 IoT Sensor Applications Ranking". Disponível em: http://www.libelium.com/resources/top_50_iot_sensor_applications_ranking/. Acesso em: 23 de junho de 2017.
- Meola, A. (2016), "How IoT in education is changing the way we learn", <http://www.businessinsider.com/internet-of-things-education-2016-9>. Acesso em: 17 de junho de 2017.
- Piaget, J. (2011), "Seis Estudos de Psicologia". São Paulo: Gen.
- Santos, B. P.; Silva, L.; Celes, C.; Borges, J.; Peres, B.; Vieira, M.; Vieira, L. F.; Loureiro, A. A. F. (2016), "Internet das Coisas: da teoria à prática". In: Simpósio Brasileiro de Redes de Computadores, Minicurso.
- Wadewitz, L. (2016), "Internet das Coisas: tudo o que pode ser conectado, será conectado". Disponível em: <http://www.administradores.com.br/artigos/tecnologia/internet-das-coisas-tudo-o-que-pode-ser-conectado-sera-conectado/92354/>. Acesso em: 17 de junho de 2017.