**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, em parte de forma intuitiva, 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, utilizando o sistema de gerenciamento de banco de dados PostgreSQL e a integração por WebServices. Espera-se que essa plataforma possibilite ao educadorrealizar 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 ofThings*).

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 RaspberryPi, 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. Fundamentaçãoteórica

Para fundamentar o desenvolvimento deste trabalho foi necessário realizar um estudo bibliográfico de três aspectos teóricos essenciais ao desenvolvimento do trabalho: IoT, *Web Services* e Reflexão Computacional.

## 2.1 IoT

A expressão *Internet ofThings* 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-FrequencyIDentification* - Identificação por Radio 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 Koucheryavy 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 ativa 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 IoTdiz 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.

## 2.2 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 *SQLInjection*.

Entre as tecnologias utilizadas para implementar e implantar Web Services têm-se [Lecheta 2015]: o SOAP (*SimpleObject 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.

## 2.3 ReflexãoComputacional

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. Internet das Coisas: tudo o que pode ser conectado, será conectado

Wadewitz (2016) em sua pesquisa “Sizing U pthe 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.

Umaaplicação de mercado que é apontadaé 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 indefine 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 quotidiano.

## 3.2. 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. *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 on-line) que está ganhando espaço em cursos de diversos níveis de ensino [Meola 2016].

O aumento crescente de dispositivos que utilizam IoT, em diversas situações quotidianas, 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. *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. *The Connected School: How IoT Could Impact Education*(A Escola Conectada: Como a IoT Impactará na 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.

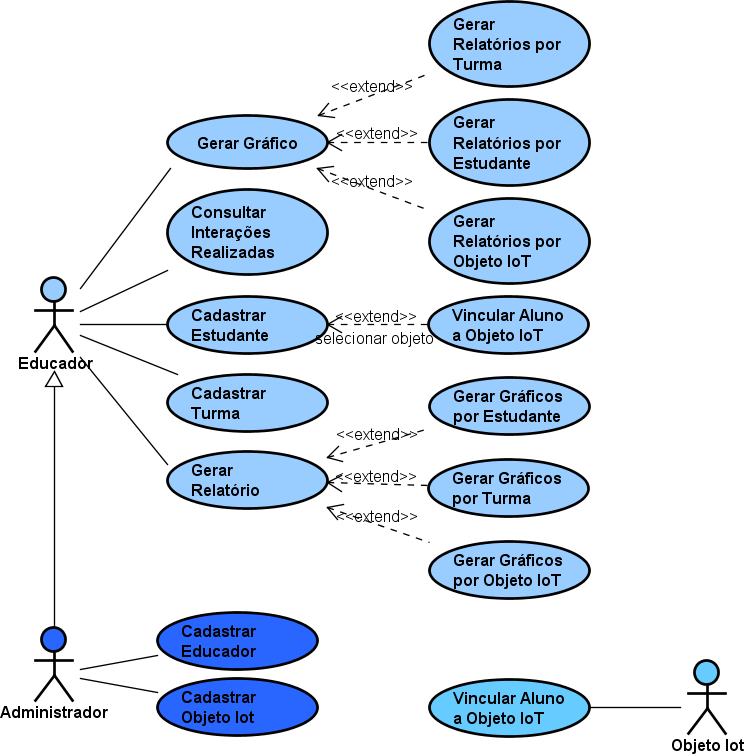
Assim, pegou-se, por exemplo, o objeto "Urso", que compreende um urso de pelúcia que contém uma tela onde o estudante interage, seguindo uma sequência de passos, que compreendem um jogo. Esse jogo é composto por fases e níveis cujo propósito é auxiliar na aquisição do conhecimento matemático e linguístico.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

|  |  |  |
| --- | --- | --- |
| ***urso.gif***  CA\_A  S  Z  A1  I  T | ***wifi.png*** | nuvem.jpg |

**Figura1. Representação de um ObjetoIoT**

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**

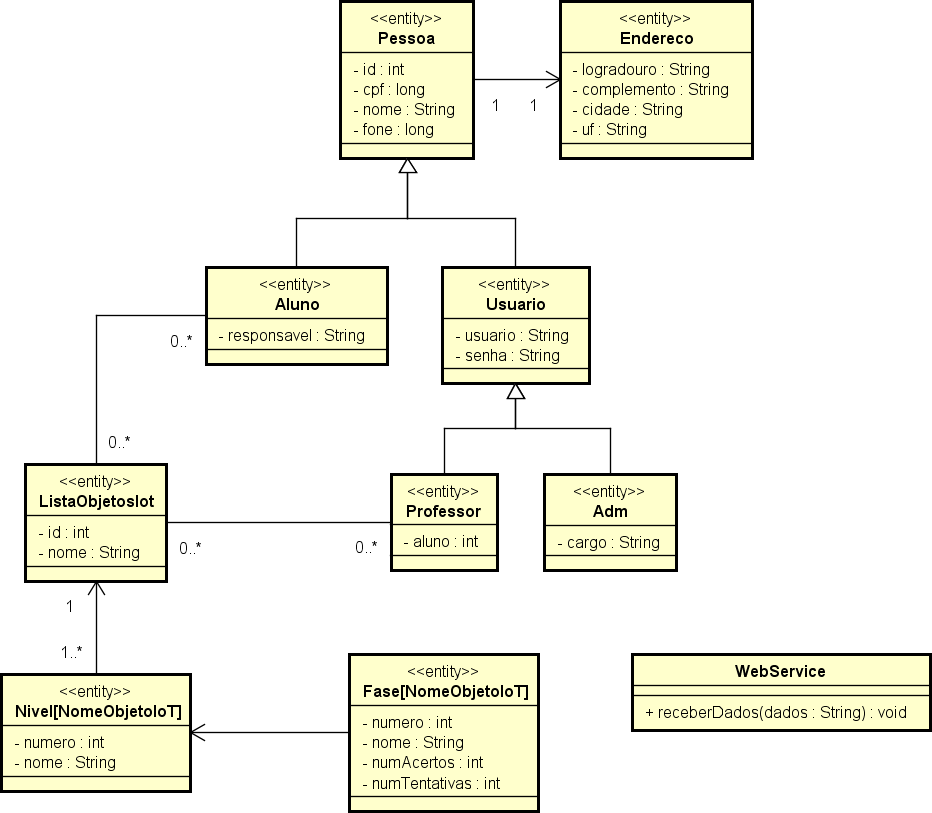
## 4.3 Análise e Projeto

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 2. Diagrama de Casos de Uso da Plataforma Proposta**

Observa-se que para o desenvolvimento dessa solução foram selecionadas as seguintes tecnologias: linguagem de programação Java, os frameworks Hibernate e PrimeFaces com JSF. Com relação ao banco de dados será utilizado o PostgreSQL. Alguns detalhes relacionados com a implementação e testes encontram-se descritos na próxima seção.

## 4.4 Implementação

A implementação consiste na codificação das soluções definidas nas etapas anteriores utilizando-se as tecnologias selecionadas. Além disso, será utilizado 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.Ainda serão realizados os testes do sistema testando as funcionalidades de cada módulo, visando produzir um software com mais qualidade.

FALAR DOS TESTES AQUI!!!!

# 5. ConsideraçõesFinais

A maior dificuldade encontrada até o momento 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.

O trabalho terá continuidade seguindo os passos estabelecidos pelo cronograma esquematizado pelaTabela 1.

**Tabela 1. Cronograma**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | mar | abr | mai | jun | jul | 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/>.Acessoem: 23 dejunho de 2017.

Barros, A. G. e Souza, C. H. M. (2016), “A internet de todas as coisas e a educação: Possibilidades e oportunidades para osprocessos de ensino e aprendizagem.”,<http://revista.srvroot.com/linkscienceplace/index.php/linkscienceplace/article/view/249>. Acessoem: 17 dejunho de 2017.

Buyya, R.; Dastjerdi,A. V. (2016), Internet of Things: principles and paradigms. Cambridge: Elsevier.

Cajide, J (2016), "The ConnectedSchool: HowIoTCouldImpactEducation", <http://www.huffingtonpost.com/jeanette-cajide/the-connected-school-how-_b_8521612.html>.Acessoem: 23 dejunho 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, 2014, Proceedings.

Guerra, E. (2014), "Componentesreutilizáveisem 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 nanuvem do Google". Rio de Janeiro: Novatec.

Libelium. "Top 50 IoT Sensor Applications Ranking". Disponívelem: <http://www.libelium.com/resources/top_50_iot_sensor_applications_ranking/>. Acessoem: 23 dejunho 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>. Acessoem: 17 dejunho de 2017.

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: SimpósioBrasileiro de Redes de Computadores, Minicurso.

Wadewitz, L. (2016) "Internet das Coisas: tudo o que podeserconectado, seráconectado".Disponívelem:<http://www.administradores.com.br/artigos/tecnologia/internet-das-coisas-tudo-o-que-pode-ser-conectado-sera-conectado/92354/>.Acessoem: 17 dejunho de 2017.

Lista sequencial de testes:

O método utilizado para a realização de todos os testes de desenvolvimento até agora foi a construção e a utilização de uma classe java principal de nome “Teste” em um pacote de nome “teste” para a execução de métodos de classes e testar seu retorno na aplicação. Quando não atendidas as expectativas do cenário de teste realizado, as exceções ou erros retornados na execução foram tratadas conforme seu retorno ou sub-cenário de teste especifico para o caso. Outra situação que define bem o método de desenvolvimento deste projeto é o fato de haver parcos recursos de desenvolvimento disponíveis nos locais onde a possibilidade de construção do mesmo era possível. A forma encontrada para lidar com este contratempo foi a realização de sub-projetos com as funções específicas (por exemplo, construção de um projeto com o propósito especifico de testar o polimorfismo) a serem testadas e depois incorporação do recurso no projeto principal. Eu prefiro o versionamento direto do projeto principal, mas simplesmente não era possível abrir o projeto acadêmico em qualquer lugar que me encontrava com computadores para o mesmo (trabalho ou ambientes sem internet ou banco de dados).

1. Construção do projeto através de métodos padrões da IDE NetBeans com as bibliotecas disponíveis pela plataforma de desenvolvimento. O primeiro framework utilizado no teste foi o Hibernate e a construção automatizada se mostrou depreciada com as versões das bibliotecas desatualizadas em relação a algumas formas de recursos utilizados como construção de unidade genérica e link de dependência. (Na verdade não funcionou quase nada com as importações sugeridas pela documentação do próprio Hibernate).
2. Abordagem de bibliotecas modificada para o uso de dependências gerenciadas pelo repositório Maven. 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. A forma de abordagem via construção de seção e não via gerenciamento de entidade passou a funcionar.
4. Construção de uma classe de conexão e uma classe de operações de manipulação de banco básicas (inclusão, remoção, consulta e alteração) realizada para início de testes funcionais assim como uma classe modelo inicial (professor por escolha do desenvolvedor). Os métodos específicos à classe se mostraram satisfatórios quando testados em versão para reflexão à exceção da consulta nomeada, pois o texto referia-se especificamente à classe utilizada.
5. 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 carinhosamente chamado de “toQuery” como referência aos métodos padrões sobre-escritos como “toString”.
6. Criação de segunda classe modelo para uso da mesma classe de manipulação simultaneamente (Aluno). Como o uso foi atendido satisfatoriamente, foi realizado o acoplamento da classe de conexão com a de manipulação para agilização do processo de construção e finalização de sessão.
7. O tratamento de exceções do Hibernate com as anotações referentes a restrições dos campos nas tabelas se mostrou frustrante pois ainda trata de forma abrangente muitas exceções de origem diferentes.
8. Optou-se por retorno de string a todos os métodos da classe de manipulação para facilitar o tratamento do mesmo, embora ainda não tenham sido realizados todos os cenários idealizados.
9. 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 o banco de dados, então passou-se a utilizar a abordagem mappesuperclass e table\_per\_class. Não funcionaram. Após analise exaustiva do próprio Hibernate, percebeu-se que as classes importadas do mesmo não atendiam esta funcionalidade corretamente, então passou-se a utilizar as “super” classes do JPA através do Hibernate (não tenho como descrever isto de forma educada, mas é como ter que andar de costas para ir pra frente...). Agora o polimorfismo funciona.
10. Como a utilização de construção de classes de entidade dinâmicas será componente fundamental do projeto, foi removido o mapeamento de uma das classes para analisar o comportamento no acesso ao banco pelo Hibernate. Não funcionou. Após tentativas frustradas de alterar o arquivo de configuração do Hibernate em tempo de execução, descobriu-se o recurso de configuração do mesmo de forma dinâmica. 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 através de reflexão. Agora funciona.
11. Ainda existem exceções incongruentes com o Hibernate em situações de entrada de dados ao banco. Não foi possível até o momento realizar um mapeamento satisfatório das exceções do framework. Exemplo é a inserção de registro que é realizada corretamente e ainda assim retorna uma exceção.
12. Construção de duas para inserção de dados em caráter de teste 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. Exceções podem ser tratadas de forma silenciosa sem apresentar estouro na tela, salvo as de estrutura de bean.
13. Carregamento, construção e compilação de classe realizado e testado de forma simples (sem construção por ferramenta interativa apenas informação de string) em projeto paralelo. Seu uso só foi possível através de reflexão.
14. Iniciado recurso de construção de classe “construtora de classes” para estruturar as classes pertinentes a ObjetosIot consumidores do projeto. Ainda não foram realizados testes.

Trecho de código para ilustração:

|  |  |
| --- | --- |
| 1. | package util; |
| 2. | import java.lang.reflect.InvocationTargetException; |
| 3. | import java.lang.reflect.Method; |
| 4. | import java.util.ArrayList; |
| 5. | import java.util.List; |
| 6. | import java.util.logging.Level; |
| 7. | import java.util.logging.Logger; |
| 8. | import org.hibernate.HibernateException; |
| 9. | import org.hibernate.Session; |
| 10. | import org.hibernate.SessionFactory; |
| 11. | import org.hibernate.cfg.Configuration; |
| 12. | import org.hibernate.metadata.ClassMetadata; |
| 13. |  |
| 14. | public class CrudUtil { |
| 15. |  |
| 16. | SessionFactory factory; |
| 17. | Session sessao; |
| 18. |  |
| 19. | public CrudUtil() { |
| 20. | } |
| 21. |  |
| 22. | private void inicia\_sessao(Class classe) { |
| 23. | try { |
| 24. | Configuration config = new Configuration(); |
| 25. |  |
| 26. | config.addAnnotatedClass(classe); |
| 27. | config.configure(); |
| 28. | factory = config.buildSessionFactory(); |
| 29. | sessao = factory.openSession(); |
| 30. | sessao.beginTransaction(); |
| 31. | } catch (HibernateException e) { |
| 32. | System.out.println("Criação da sessão falhou. " + e); |
| 33. | } |
| 34. | } |
| 35. |  |
| 36. | private void encerra\_sessao() { |
| 37. | sessao.close(); |
| 38. | factory.close(); |
| 39. | } |
| 40. |  |
| 41. | public String exclui(Object registro) { |
| 42. | String saida; |
| 43. |  |
| 44. | inicia\_sessao(registro.getClass()); |
| 45. | ClassMetadata metadata = sessao.getSessionFactory()  .getClassMetadata(registro.getClass()); |
| 46. | Object teste = sessao.get(registro.getClass(), metadata.getIdentifier(registro)); |
| 47. |  |
| 48. | if (teste == null) { |
| 49. | encerra\_sessao(); |
| 50. | return "Registro não encontrado"; |
| 51. | } |
| 52. |  |
| 53. | try { |
| 54. | sessao.delete(teste); |
| 55. | sessao.getTransaction().commit(); |
| 56. | sessao.flush(); |
| 57. | saida = "Regitro excluido com sucesso"; |
| 58. | } catch (RuntimeException e) { |
| 59. | if (sessao.getTransaction() != null) { |
| 60. | sessao.getTransaction().rollback(); |
| 61. | } |
| 62. | saida = e.toString(); |
| 63. | } finally { |
| 64. | encerra\_sessao(); |
| 65. | } |
| 66. | return saida; |
| 67. | } |
| 68. |  |
| 69. | public String altera(Object registro) { |
| 70. | String saida; |
| 71. |  |
| 72. | inicia\_sessao(registro.getClass()); |
| 73. | ClassMetadata metadata = sessao.getSessionFactory()  .getClassMetadata(registro.getClass()); |
| 74. | Object teste = sessao.get(registro.getClass(), metadata.getIdentifier(registro)); |
| 75. |  |
| 76. | if (teste == null) { |
| 77. | encerra\_sessao(); |
| 78. | return "Registro não encontrado"; |
| 79. | } |
| 80. |  |
| 81. | try { |
| 82. | sessao.merge(registro); |
| 83. | sessao.getTransaction().commit(); |
| 84. | sessao.flush(); |
| 85. | saida = "Regitro alterado com sucesso"; |
| 86. | } catch (RuntimeException e) { |
| 87. | if (sessao.getTransaction() != null) { |
| 88. | sessao.getTransaction().rollback(); |
| 89. | } |
| 90. | saida = e.toString(); |
| 91. | } finally { |
| 92. | encerra\_sessao(); |
| 93. | } |
| 94. | return saida; |
| 95. | } |
| 96. |  |
| 97. | public String insere(Object registro) { |
| 98. | String saida; |
| 99. | inicia\_sessao(registro.getClass()); |
| 100. |  |
| 101. | try { |
| 102. | sessao.save(registro); |
| 103. | sessao.getTransaction().commit(); |
| 104. | sessao.flush(); |
| 105. | saida = "Regitro gravado com sucesso"; |
| 106. | } catch (RuntimeException e) { |
| 107. | String razao = e.toString(); |
| 108. | if (sessao.getTransaction() != null) { |
| 109. | sessao.getTransaction().rollback(); |
| 110. | } |
| 111. | if (razao.contains("null")) { |
| 112. | razao = razao.replaceAll(".\*\\.", "")  + " não pode estar vazio"; |
| 113. | } else { |
| 114. | razao = "Campos únicos duplicados."; |
| 115. | } |
| 116. | saida = razao; |
| 117. | } finally { |
| 118. | encerra\_sessao(); |
| 119. | } |
| 120. | return saida; |
| 121. | } |
| 122. |  |
| 123. | public List<Object> lista(Object origem) { |
| 124. | List<Object> lista = null; |
| 125. |  |
| 126. | inicia\_sessao(origem.getClass()); |
| 127. |  |
| 128. | try { |
| 129. | Class classe = origem.getClass(); |
| 130. | Method toQuery = classe.getDeclaredMethod("toQuery"); |
| 131. | String namedQuery = (String) toQuery.invoke(origem); |
| 132. | lista = sessao.createQuery(namedQuery).list(); |
| 133. | sessao.flush(); |
| 134. | if (lista == null) { |
| 135. | lista = new ArrayList(); |
| 136. | lista.add("Registro não encontrado"); |
| 137. | } |
| 138. | sessao.flush(); |
| 139. | } catch (HibernateException e) { |
| 140. | lista = new ArrayList(); |
| 141. | lista.add("Registro não encontrado"); |
| 142. | } catch (NoSuchMethodException | SecurityException | IllegalAccessException | IllegalArgumentException | InvocationTargetException ex) { |
| 143. | Logger.getLogger(CrudUtil.class.getName()).log(Level.SEVERE, null, ex); |
| 144. | } finally { |
| 145. | encerra\_sessao(); |
| 146. | } |
| 147. | return lista; |
| 148. | } |
| 149. |  |
| 150. | } |

Reflexão

A técnica de reflexão computacional no Java consiste em acessar, manipular e utilizar parâmetros, métodos e características de objetos indiferente à sua classe de instância. Abaixo alguns exemplos ilustrativos do uso no desenvolvimento do projeto:

Iniciar sessão na Tabela correspondente:

Na linha 22, informamos como parâmetro um objeto do tipo Class (classe) para início da sessão com o mapeamento direcionado a esta classe. Desta forma, conseguimos mapear dinamicamente a classe passada como parâmetro em tempo de execução. Na linha 24, utilizamos 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 26, mapeamos a classe através do método “addAnnotatedClass(Class object)” e formalizamos a configuração com o método configure(). A partir deste ponto, a classe passada como parâmetro já se encontra mapeada e disposta para operações como o banco.

Exclusão de registros

Na linha 41, informamos como parâmetro um objeto cuja classe será utilizada para identificação para mapeamento (linha 44). Na linha 46, utilizamos um objeto temporário para testar a existência do objeto através da classe e dos parâmetros instanciados do mesmo. Neste caso a chave primaria da tabela a que a classe instanciada do objeto.

Exemplo hipotético de uso:

CrudUtil dao = new CrudUtil();

Professor professor = new Professor(24);

Aluno aluno = new Aluno(24);

System.out.println("Exclusão professor resultado: "

+ dao.exclui(professor));

System.out.println("Exclusão aluno resultado: "

+ dao.exclui(aluno));

Alteração de registros

Assim como no processo de exclusão, o registro tem sua existência no banco de dados testada através de dados obtidos via reflexão.

Exemplo hipotético de uso:

CrudUtil dao = new CrudUtil();

Professor professor = new Professor(24);

professor.setNome(“Antonio”);

// Trocando de Adamastor para antonio

Aluno aluno = new Aluno(24);

aluno.setCidade(“Porto Alegre”);

// Trocando de Canoas para Porto Alegre

System.out.println("Alteração professor resultado: "

+ dao.altera(professor));

System.out.println("Alteração aluno resultado: "

+ dao.altera(aluno));

Gravação de registros novos

Uma vez que tem sua classe utilizada para início da sessão (linha 102) o mapeamento da classe utiliza os parâmetros do objeto por sua instância original diretamente no comando de gravação no banco.

Exemplo hipotético de uso:

CrudUtil dao = new CrudUtil();

Professor professor = new Professor(24);

professor.setNome(“Antonio”);

Aluno aluno = new Aluno(24);

aluno.setNome(“José”);

System.out.println(dao.insere(professor));

System.out.println(dao.insere(aluno));

Listagem de registros

De forma mais explícita, na linha 129, obtemos a classe do objeto parâmetro. Utilizamos esta classe para instanciar um método definido no padrão do projeto chamado “toQuery” (linha 130) 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.

Exemplo hipotético de uso:

CrudUtil dao = new CrudUtil();

Professor professor = new Professor();

Aluno aluno = new Aluno();

List<professor> listaProfessor = (List<professor>) dao.lista(professor);

// ou registro específico

professor.setId(24);

List<professor> listaProfessor = (List<professor>) dao.lista(professor);

professor = listaProfessor(0);

Alteraçõespropostas

Carlos:

* Recomendoincluir no TCC2, um capituloexplicandoEvoluçãocognitiva de estudantes
* Aprovo o presentetrabalho de conclusão de curso, pelo o tema, estrutura e organização do texto, recomendoacrescentar no referencialteórico o que significauma “Evoluçãocognitiva de estudantes” citandoautores e justificando a importância do trabalho para acompanharoumapearestaevolução. Na Metodologia, sintofalta de identificar que público-alvopretendeatingir, Ensino Fundamental, MédioouSuperior?\_Eporfimuma boa revisão no texto, poisapresentaalgunsparágrafosconfusos.

Tanise:

* texto no geralestábemescrito mas a seção 4 deveserrevisadapoisapresentaalgunserros, em especial o parágrafo que inicia com: Para modelarosrequisitosoptotu-se..
* trabalhopossui boa apresentação e organização, mas a figura do diagrama de casos de usoficoumuitopequena.
* textoestábemestruturado e organizado. A minhasugestãoseriaretirar a seção 3.1, alterar o texto de introdução da seção 3 e iniciar com IoTnaeducação.
* trabalhoapresenta um ótimoconteúdo mas sentifalta do diagrama de classes.
* trabalhoestábemescrito e organizado. Algunserros de gramáticaforamencontradosnaseção 4. Acho que osignificado de IoTpoderiaestar no resumo. Acho que a frase “Este artigoapresenta .. “ que está no resumonão define com clareza o objetivo do trabalho. O diagrama de classes poderiatersidoapresentado no artigo.