

Um modelo de arquitetura orientada a serviços aplicada à sistemas de monitoramento remoto

Aline M. S. Costa¹, Francicleia R. Cardoso¹,
Luis C. de O. Silva¹, Sérgio S. Costa¹

¹ Curso de Engenharia da Computação – Universidade Federal do Maranhão (UFMA)
São Luís – MA – Brazil

acostta2, leiacardoso21, lcoliveira.silva, sergio.costa@ufma.br

Abstract. *Remote monitoring systems allows to capture data from regions of difficult access. Recent technologies have made possible the development of these systems with low cost. This work presents a service-oriented architecture model that allows remote access to the data collected by these systems. This architecture will allow access to this data, regardless of the type of monitoring system. For the validation of the proposed architecture, a monitoring system, a web service and a client application for visualization of the data were developed.*

Resumo. *Sistemas de monitoramento permitem capturar dados de regiões extensas e de difícil acesso. Tecnologias recentes, tem possibilitado o desenvolvimento destes sistemas com baixo custo. Este trabalho apresenta um modelo de arquitetura orientada a serviços que possibilita o acesso remoto aos dados coletados por estes sistemas. Esta arquitetura possibilitará o acesso a estes dados, independentemente do tipo de sistema de monitoramento. Para a validação da arquitetura proposta, foram desenvolvidos um sistema de monitoramento, um serviço web e uma aplicação cliente para visualização dos dados.*

1. Introdução

Diversas tecnologias foram criadas com o objetivo de estender as capacidades humanas. Por exemplo, a capacidade de capturar dados de ambientes de difícil acesso ou de áreas muito extensas. Essa necessidade deu origem a uma área denominada de sensoriamento remoto. De acordo com [Florenzano 2007], sensoriamento remoto é o termo dado a obtenção de dados por meio de sensores instalados em plataformas terrestres, aéreas e orbitais. Esse termo pode ser utilizado para designar qualquer forma de captura de dados remotamente, incluindo grandes plataformas como satélites até minúsculas como os *smartdust* (poeiras inteligentes). Sensores podem ser acoplados a robôs ou veículos móveis, o que possibilita a aquisição de dados em locais de difícil acesso e de larga escala. Segundo [Dudek and Jenkin 2010], a robótica móvel é uma área de pesquisa que lida com veículos autônomos e semi-autônomos em um espaço de larga-escala. Este último requisito é o que difere esta de outras áreas da robótica. Como observado em [Dudek and Jenkin 2010], essa é uma área intrinsecamente interdisciplinar e envolve áreas como a mecânica, a computação, elétrica, psicologia e mecatrônica.

Independentemente do seu modo de operação e ambiente de utilização, todos estes veículos são embarcados com diversos sensores que coletam dados do ambiente. Deste modo, este trabalho tem como objetivo desenvolver e testar uma arquitetura orientada a

serviços que possa ser aplicada independentemente do sistema de monitoramento. Para tanto, foram desenvolvidos um sistema embarcado composto por sensores, um serviço *web* e uma aplicação cliente para validar a arquitetura. Essa aplicação permite visualizar os dados capturados pelo veículo e mostra-los independentemente do veículo e dos sensores existentes. Este artigo está organizado da seguinte maneira: a Seção 2 apresenta brevemente alguns conceitos fundamentais e trabalhos relacionados; a Seção 3 apresenta a arquitetura desenvolvida; a Seção 4 os experimentos executados e a Seção 5 apresenta as considerações finais.

2. Fundamentação

Como apresentado anteriormente, este trabalho tem como objetivo prover um modelo de arquitetura orientada a serviços aplicada a monitoramento de dados por robôs ou veículos móveis. A arquitetura denomina-se orientada a serviços quando esta emprega o conceito de um produtor e um consumidor de serviços [Ferreira 2015]. Usualmente estes serviços são desenvolvidos e disponibilizados através da *web*. Um serviço *web* é um tipo de arquitetura que permite a troca de informações em ambientes eminentemente heterogêneos [Magri 2013]. Quanto a robôs ou veículos móveis, estes são denominados hardwares móveis compostos por sensores e atuadores. Eles podem ser classificados de duas maneiras, quanto ao modo de operação ou quanto ao ambiente de atuação [Bräunl 2008, Secchi 2008].

Quanto ao modo de operação, eles podem ser classificados como remotos ou autônomos. Os veículos remotos são controlados por operadores através de controle remoto e estão em contínua comunicação com uma base de operação, por meio de sinais de rádio, acústicos ou por cordões umbilicais, exigindo supervisão e intervenção humana, o que limita seu raio de ação. Não possuem uma trajetória predefinida e por isso são empregados quando um ou mais parâmetros da missão são desconhecidos. Por outro lado, os veículos autônomos são providos de sensores e atuadores gerenciados por sistema central e são capazes de executar instruções definidas pelo usuário. Geralmente dependem da eletrônica embarcada para a execução de tarefas, estas, por sua vez, consistem em uma série de instruções pré-programadas em tempo real por meio de dados ou informações provenientes dos sensores contidos no veículo.

Quanto ao ambiente de atuação eles podem ser classificados como terrestre, aéreo, submarino e subaquático. Os veículos de ambiente terrestre são conhecidos também como UGV (*Unmanned Ground Vehicles*) ou veículos terrestres não tripulados. Fazem parte desta categoria tanto os robôs remotos quanto os autônomos. São empregados tanto no ramo civil como no militar, destacando-se a exploração espacial. Os veículos de ambiente aéreo são conhecidos também como UAVs (*Unmanned Aerial Vehicles*) ou veículos aéreos não tripulados. Eles podem ser empregados no ramo civil, em aplicações como, controle de pragas em lavouras, fiscalização do avanço do desmatamento florestal e telecomunicação. Os veículos de ambiente submarino são conhecidos também como UUVs (*Unmanned Underwater Vehicle*) ou veículos submarinos não tripulados. Estes veículos podem ser usados pela indústria petrolífera e em monitoramento ambiental. Os veículos de ambiente subaquático são conhecidos também como ASVs (*Autonomous Surface Vessel*) ou embarcações autônomas de superfície. Embora sejam relativamente mais simples que os veículos das outras categorias eles se destacam por características próprias como versatilidade, segurança do equipamento, acessibilidade e baixo custo. Atuam no

ramo da comunicação entre embarcações de base e outros sistemas de monitoramento, além de monitoramento ambiental, levantamento batimétrico e resguardo de fronteiras e portos [Bräunl 2008, Secchi 2008].

Na próxima seção será descrita a arquitetura que pode ser utilizada independentemente do tipo de robô ou veículo móvel.

3. Arquitetura

A Figura 1 apresenta a arquitetura composta por quatro componentes: a) sistemas de monitoramento ambientais (SMAs), b) serviço *Web*, c) banco de dados e d) clientes. Os SMAs são componentes de hardware compostos por sensores ambientais que podem enviar uma coleção de dados a um serviço *web*. O serviço *web* armazena os dados coletados em um banco de dados, que poderão então ser consumidos, por aplicações clientes, desenvolvidas para plataformas *Desktop*, *web* ou *mobile*.

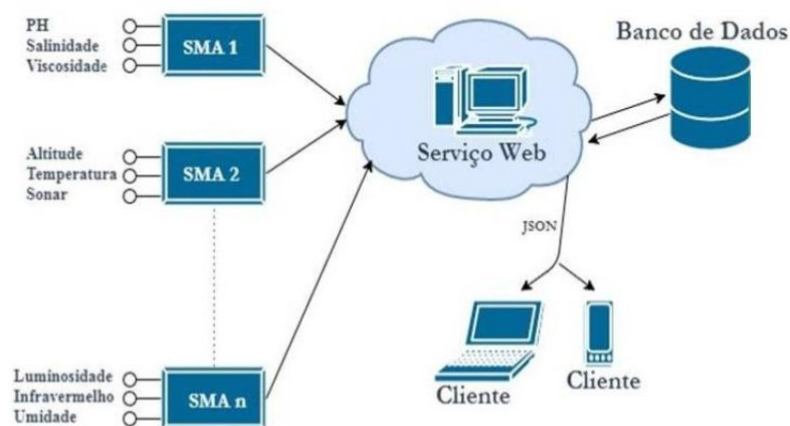


Figura 1. Visão geral da arquitetura.

O SMA pode ser desenvolvido em uma plataforma de baixo custo, incluindo Arduino, placas e sensores. Contudo, o tipo de *hardware* dependerá principalmente do ambiente que irá atuar. O serviço *Web* segue o modelo arquitetural REST (*Representational State Transfer*) ou RESTful. Esse modelo refere-se a uma coleção de princípios de arquiteturas de rede que foca em acesso a recursos de formas simples e sem manutenção de estado [Josuttis 2007]. A implementação deste serviço é independente de linguagem de programação e sistema operacional. Os dados podem ser armazenados em um banco de dados relacional, como MySQL¹ ou PostgreSQL². A comunicação entre o serviço *web* e os clientes pode ser feita através do formato de dados JSON (*JavaScript Object Notation*). De acordo com [Crockford 2006] o JSON pode ser uma alternativa ao XML (*eXtensible Markup Language*), onde por vezes se sobressai em questão de velocidade de leitura dos dados. Na seção que segue será explanado com mais detalhes o desenvolvimento destes componentes.

¹<https://www.mysql.com/cloud/>

²<https://www.postgresql.org/>

4. Experimentos e resultados

Nesta seção será detalhada o desenvolvimento, experimentos e resultados alcançados com cada um dos quatro componentes.

4.1. Sistema de monitoramento ambiental

Para validar a arquitetura, foi feita a simulação do sistema, utilizando a plataforma Arduino, que simula um monitoramento subaquático, denominado aqui como sistema teste. A conexão do sensor à placa foi feita com o auxílio de uma *Protoboard*. A plataforma é composta por microcontroladores programáveis montados em placas de circuito impresso com reguladores de tensão, porta serial *UART* e que podem interagir com seu ambiente por *hardware* e *software* [McRoberts 2011]. Além de um módulo GPS GPSNEO-6M, sensor DHT11 de umidade e temperatura, e de um módulo para armazenamento externo composto por cartão microSD. A montagem do sistema teste é mostrado na Figura 2.

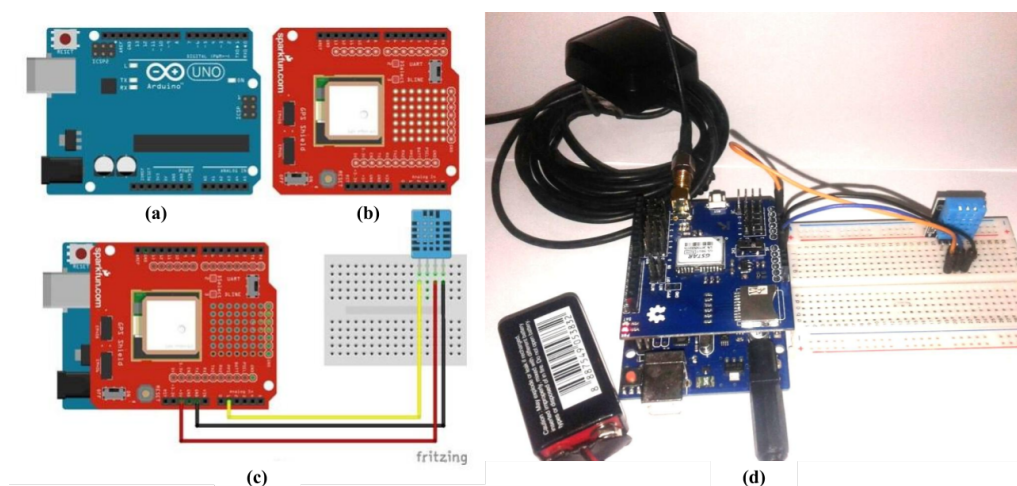


Figura 2. Simulação do sistema de monitoramento, onde são apresentadas a placa Arduino UNO (a), *shield* GPS (b), e o sistema acoplado incluindo o sensor DHT11 (c), e o protótipo do sistema de monitoramento Teste (d).

Na execução dos testes foram armazenados os dados de data, hora, latitude, longitude, o identificador do sensor e do veículo, e os valores sensoriais de temperatura e umidade. Para estes experimentos, os SMAs não enviaram os dados em tempo real ao servidor. Ao invés disso, os dados foram extraídos do dispositivo de armazenamento e enviados ao servidor através de um computador externo. Mesmo não sendo ideal, o uso de um computador externo ainda será necessário em locais sem acesso a internet.

4.2. Servidor Web

O servidor foi desenvolvido usando plataforma Java Enterprise Edition (Java EE), com um servidor de container GlassFish³. Os serviços disponibilizados pelo servidor são: a) sensores, b) veículos e c) monitora. O serviço "veículo", através de um método GET, retorna o identificador, a descrição e a URL, para uma figura do veículo. O serviço "sensor", dado um veículo, retorna as informações dos sensores disponíveis neste. O serviço "monitora", através de método GET, disponibiliza as informações de todos os dados de um

³<https://javaee.github.io/glassfish/>

dado sensor, como data, hora, valor dos sensores e identificador dos veículos e dos sensores. Todas essas informações são apresentadas em formato JSON. A Figura 3 apresenta a requisição GET ao serviço monitora e a apresentação dos dados através do navegador respectivamente.

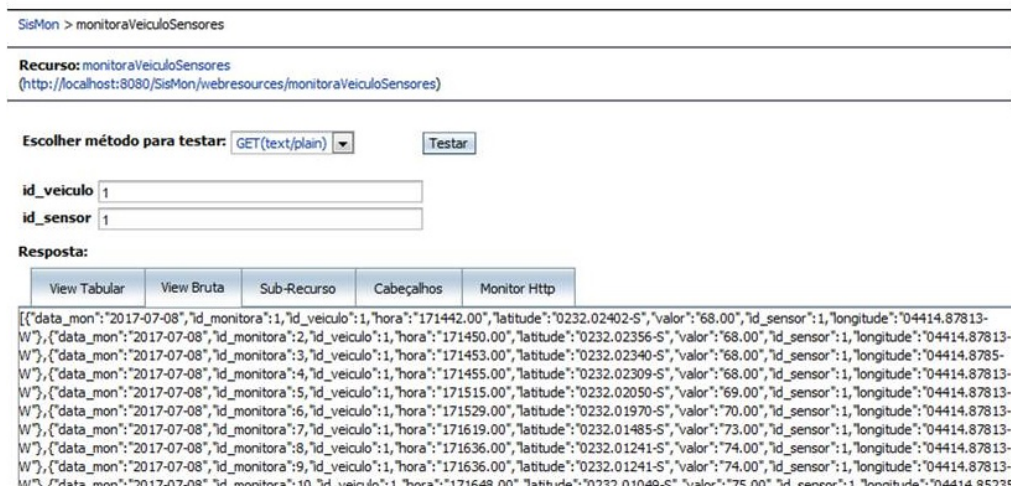


Figura 3. Requisição GET

4.3. Banco de dados

Para o armazenamento e manutenção dos registros de cada SMA se fez necessária a implementação de um banco de dados, que é basicamente uma estrutura computacional que armazena informações e permite que os usuários as busquem e as modifiquem. A Figura 4 ilustra as tabelas implementadas.

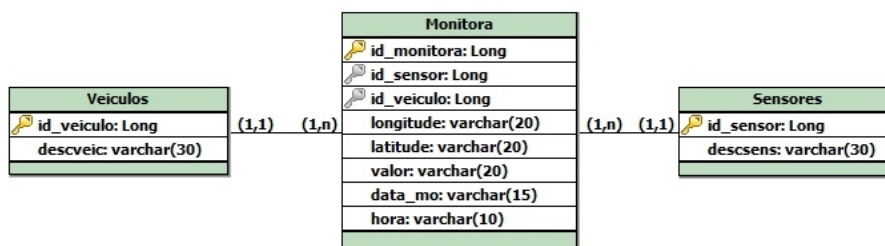


Figura 4. Modelo lógico do banco de dados da aplicação

As tabelas veículos e sensores possuem as informações de descrição e seus identificadores, e a tabela monitora possui as informações que serão coletadas por cada sensor.

4.4. Cliente

Os dados disponibilizados podem ser consumidos por uma aplicação cliente, esta aplicação permite a visualização dos dados capturados pelos sensores instalados no sistema de monitoramento. Foi desenvolvido uma aplicação *mobile* para visualização dos dados através do aplicativo. Esta aplicação foi desenvolvida para a plataforma iOS usando a linguagem Swift. Para estes experimentos o design e as funcionalidades foram simplificadas, como apresentadas na Figura 5.

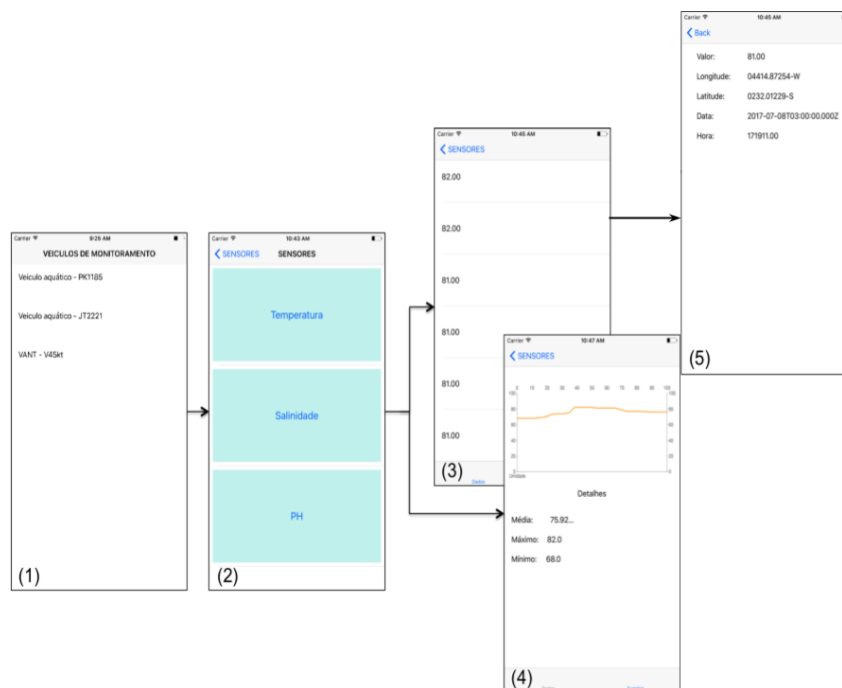


Figura 5. Interface do aplicativo - Telas de 1 a 5

A Tela 1, recebe os dados do servidor, referentes aos veículos de monitoramento, e os apresenta na tela. A atual versão, apresenta apenas o nome do veículo e seu identificador, futuramente, em sua versão final, ela deverá mostrar também a imagem dos veículos cadastrados. Selecionado um veículo, a Tela 2 apresenta os sensores presentes no SMA selecionado pelo usuário, e cada sensor apresenta seus dados, que podem ser visualizados nas telas que seguem. Atualmente testamos duas formas de visualização, o formato tabela, apresentado na Tela 3, e o formato gráfico, apresentado na Tela 4. Para a geração do gráfico, que apresenta as informações recebidas pelos SMAs, que são os dados aferidos pelos sensores acoplados, foi utilizado o serviço Google Charts. Como exemplo para esta aplicação mostra-se em formato de gráfico, na tela 4, os dados de um sensor de salinidade. Também é possível observar algumas informações de sumarização, obtidas através dos dados iniciais deste sensor, como o máximo, mínimo, média e desvio padrão. Ademais, cada sensor associa informações espaço-temporais, ou seja a localização espacial, a data e o horário da ocorrência. Para visualizar essas informações adicionais, o usuário pode selecionar um valor específico e a aplicação irá mostrar estes detalhes na Tela 5, como mostra a Figura 5. Na versão final da aplicação, a tela 5 deverá mostrar a localização deste ponto em um mapa. Os SMAs e sensores apresentados na Figura 5 são ilustrativos.

5. Conclusão

Neste trabalho, foi apresentado um modelo de arquitetura orientada a serviços que pode ser aplicada em sistemas de monitoramento remoto.

Para tanto, foi implementado um sistema embarcado que pudesse alimentar um banco de dados com informações de geolocalização e dos dados coletados por alguns sensores embarcados. Além disso, foi desenvolvida uma aplicação, na plataforma iOS,

para visualização dos dados coletados pelo sistema através dos serviços *web*. Foi incluída a visualização de dados estatísticos como valores máximos e mínimos, e desvio padrão dos últimos dados recebidos em determinado período, além da visualização no formato de gráfico. Porém, outras visualizações poderão ser desenvolvidas, como por exemplo, a geração de mapas geográficos.

Referências

- Bräunl, T. (2008). *Embedded robotics: mobile robot design and applications with embedded systems*. Springer Science & Business Media.
- Crockford, D. (2006). The application/json media type for javascript object notation (json).
- Dudek, G. and Jenkin, M. (2010). *Computational principles of mobile robotics*. Cambridge university press.
- Ferreira, P. B. V. (2015). *Arquitetura REST em smartphones Android*. PhD thesis.
- Florenzano, T. G. (2007). *Iniciação em sensoriamento remoto*. Oficina de Textos.
- Josuttis, N. M. (2007). *SOA in practice: the art of distributed system design*. "O'Reilly Media, Inc."
- Magri, J. A. (2013). Criando e usando web service. *Augusto Guzzo Revista Acadêmica*, (11):166–183.
- McRoberts, M. (2011). *Arduino básico*. São Paulo: Novatec.
- Secchi, H. A. (2008). Uma introdução aos robôs móveis. *Instituto de Automática–INAUT–Universidad Nacional de San Juan–UNSJ–Argentina*.