

Projeto Maritaca: Arquitetura e Infraestrutura para Coleta Móvel de Dados Usando *Smartphones* *

Bruno G. dos Santos, Alvaro H. Mamani-Aliaga, Jimmy V. Sánchez,
Matheus F. Mendonça, Tiago Barabasz e Arlindo F. da Conceição¹

¹ Instituto de Ciência e Tecnologia (ICT)
Universidade Federal de São Paulo (UNIFESP)
Rua Talim, 330. São José dos Campos - SP

arlindo.conceicao@unifesp.br

Abstract. *This paper presents the architecture of Maritaca, an infrastructure for data gathering using Android mobile devices. The system creates applications for mobile devices without need of knowledge about programming techniques. The system is based on free software and can be accessed in the following address: `maritaca.unifesp.br`.*

Resumo. *Este trabalho apresenta a arquitetura do projeto Maritaca, uma infraestrutura para coleta de dados a partir de dispositivos móveis Android. O sistema permite a criação de aplicações para a coleta móvel de dados sem a necessidade de conhecimentos de técnicas de programação. O sistema é baseado em tecnologias abertas e pode ser acessado a partir do seguinte endereço: `maritaca.unifesp.br`.*

1. Introdução

Nos últimos anos, o mercado de comunicação pessoal móvel evoluiu rapidamente, devido a três fatores: a queda dos preços, o lançamento de dispositivos móveis com alta capacidade de processamento e o surgimento de novas tecnologias para o desenvolvimento de Aplicações Móveis (Apps). Pode-se citar como exemplos destes avanços o lançamento de processadores *multicore* para dispositivos móveis e o amadurecimento das plataformas de programação para sistemas móveis Android [Pereira and Da Silva 2009]. Estes fatores, combinados, criaram condições para o surgimento de uma nova categoria de aplicações: a **Coleta Móvel de Dados** (CMD) [Rezende et al. 2010].

Entretanto, a criação de aplicações para coleta móvel de dados continua exigindo o trabalho de programadores, pois ainda é preciso programar os questionários eletrônicos. Atualmente, essa é a principal limitação para a ampla utilização de CMDs, pois nem toda empresa possui recursos humanos ou financeiros, para elaborar aplicações móveis.

Para enfrentar ou reduzir estas limitações e, desse modo, contribuir para a ampla utilização de ferramentas móveis para Coleta de Dados, desenvolvemos o **Projeto Maritaca**, que visa prover soluções tecnológicas abertas e infraestrutura para criação de aplicações para Coleta Móvel de Dados (CMD). O nome Maritaca¹ vem do acrônimo *MARitaca Is a Tool to creAte Cellular phone Applications*.

*Este trabalho recebeu o apoio da FINEP, edital Telessaúde e Telemedicina, processo 04.11.0077.00, referência 1488/10. Cabe mencionar os apoios recebidos por FAPESP, CNPq, IBOPE e FAP-UNIFESP, que indiretamente colaboraram para a execução deste projeto.

¹*Maritaca* é o nome popular utilizado para designar diversas espécies de aves *psitaciformes*, da família

2. Trabalhos Relacionados

Existem algumas ferramentas com propósitos similares aos do Maritaca. A ferramenta *App Inventor*², permite construir visualmente aplicativos para a plataforma Android. Concentra-se no desenho passo-a-passo de elementos de interface, conectando-os aos respectivos eventos. O diferencial positivo do projeto Maritaca em relação ao *App Inventor* consiste em permitir o desenho mais simples e intuitivo das interfaces, isso é possível porque concentra-se em aplicações de CMD.

O *Nokia Data Gathering*³ é um sistema para criação de questionários móveis que, colocados em um servidor na Internet, podem ser acessados pelos dispositivos móveis com acesso a rede, onde os dados são coletados e armazenados nos celulares e podem ser transmitidos para um servidor. Recentemente, a solução passou a ser distribuída como software livre.

Um projeto similar que devemos citar é o DoForms⁴, um sistema para criação de questionários móveis multiplataforma, cuja proposta é semelhante ao Projeto Maritaca, porém, é de código fechado e possui limite de uso grátis.

3. Funcionalidades do Sistema

O sistema permite usuários construam aplicações para coleta de dados, que podem ser instaladas em quaisquer dispositivos móveis compatíveis com Android 2.2, ou superior. No dispositivo móvel, a App permite a coleta de dados utilizando interfaces amigáveis, onde os dados são armazenados no dispositivo móvel até serem transferidos para o servidor. Para realizar a coleta não é necessário que o usuário esteja conectado à Internet.

A seguir são apresentados os principais passos para a utilização da plataforma.

3.1. Passo-a-passo para utilização da solução

Os passos a seguir exemplificam a utilização da plataforma:

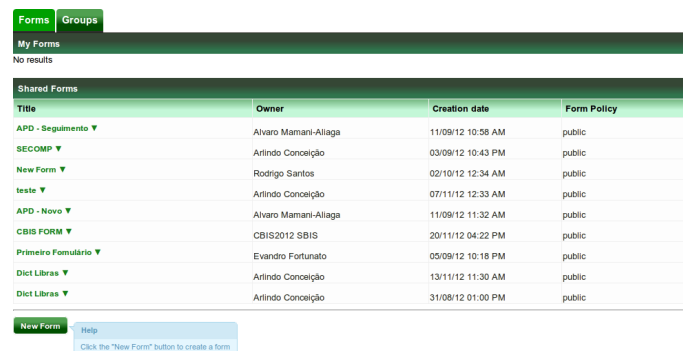
1. O sistema está disponível em <http://maritaca.unifesp.br>. Para utilizá-lo, o usuário necessita autenticar-se na tela inicial, o que pode ser realizado de duas maneiras: (i) utilizando usuário e senha registrados no sistema (ii) ou com uso de OpenID [Recordon and Reed 2006].
2. Após a autenticação, é apresentada ao usuário a tela principal do sistema, destinada ao gerenciamento de questionários (Figura 1). A lista de questionários está organizada entre formulários criados pelo próprio usuário (parte superior) e formulários com ele compartilhados (parte inferior). Há ainda um menu com abas na parte superior para o gerenciamento de grupos de usuários.
3. Para criar um novo formulário, basta pressionar o botão *New Form*, no canto inferior esquerdo da tela. A Figura 2 ilustra a interface para criação e edição de formulários contendo três perguntas como exemplo. A interface utiliza controles HTML5 arrastáveis (*Drag and Drop*), que permitem ao usuário “arrastar” as

dos *psitacídeos*, gênero *pionus*. Esta simpática ave foi escolhida como símbolo do projeto por sua capacidade de adaptação a diversos biomas, podendo ser encontrada em quase todo o território nacional.

²<http://appinventor.googlelabs.com/about>

³<http://projects.developer.nokia.com/ndg/wiki>

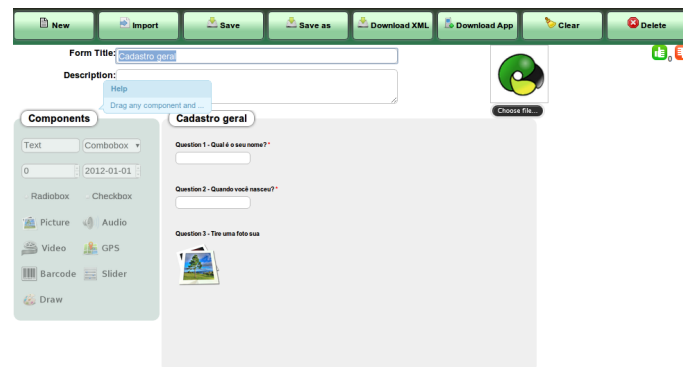
⁴<http://www.doforms.com/>



Title	Owner	Creation date	Form Policy
APD - Seguimento ▼	Alvaro Mamani-Allaga	11/09/12 10:58 AM	public
SECOMP ▼	Arlindo Conceição	03/09/12 10:43 PM	public
New Form ▼	Rodrigo Santos	02/10/12 12:34 AM	public
teste ▼	Arlindo Conceição	07/11/12 12:33 AM	public
APD - Novo ▼	Alvaro Mamani-Allaga	11/09/12 11:32 AM	public
CBIS FORM ▼	CBIS2012 SBIS	20/11/12 04:22 PM	public
Primeiro Formulário ▼	Evandro Fortunato	05/09/12 10:18 PM	public
Dict Libras ▼	Arlindo Conceição	13/11/12 11:30 AM	public
Dict Libras ▼	Arlindo Conceição	31/08/12 01:00 PM	public

Figura 1. Tela inicial de gerenciamento de formulários.

componentes a serem utilizadas no questionário. O usuário pode escolher um tipo de pergunta que queira coletar dentre o conjunto de componentes disponíveis na caixa de ferramentas localizada no lado esquerdo da interface.



Form Title:

Description:

Components:

- Text
- Combobox
- 0
- 2012-01-01
- Radiobox
- Checkbox
- Picture
- Audio
- Video
- GPS
- Barcode
- Slider
- Draw

Questions:

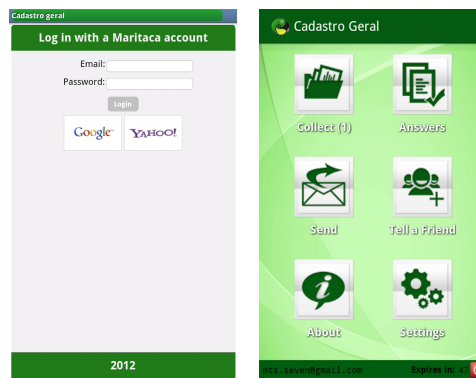
Question 1 - Qual é o seu nome? *

Question 2 - Quando você nasceu? *

Question 3 - Tenha uma foto sua

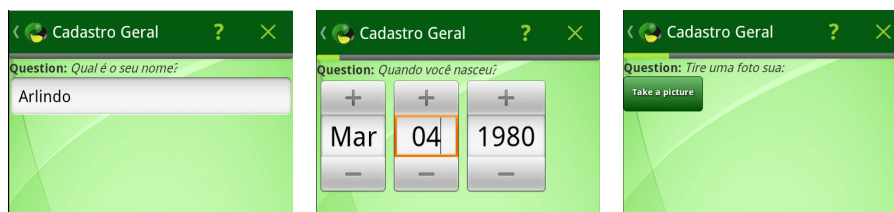
Figura 2. Editor de Formulários, preenchido.

- Uma vez criado o formulário, o usuário pode descarregá-lo para um dispositivo móvel com sistema operacional Android. A instalação do aplicativo no dispositivo móvel é trivial. A Figura 3(a) mostra a tela de autenticação na aplicação móvel. A Figura 3(b) mostra o menu principal, a partir da qual inicia-se a coleta clicando-se no botão *Collect*.
- A Figura 4 mostra a interface que foi especificada como exemplo na Figura 2 renderizada automaticamente no dispositivo móvel. Cada pergunta é mostrada em uma tela a qual é organizada em duas partes: acima, com o título do formulário, botões (*ajuda* e *cancelar*), e barra de progresso; e abaixo a questão em si. Os controles de navegação (*retornar* e *avançar*) podem ser executados através de comandos *touch*.
- A coleta de dados pode ser realizada inúmeras vezes. Para coletar os dados não é necessário que o dispositivo móvel esteja conectado à Internet, pois uma vez coletados os dados, o usuário pode conectar-se à Internet e enviar as coletas realizadas para o servidor. Quando armazenados no servidor, os dados podem ser visualizados acessando-se a plataforma. Por exemplo, a Figura 5 ilustra a visualização de dados coletados com o formulário criado para este exemplo.



(a) Tela de autenticação de (b) Tela inicial

Figura 3. Aplicação móvel



(a) Primeira questão, do tipo texto. (b) Segunda questão, do tipo data. (c) Terceira questão, do tipo foto.

Figura 4. Perguntas renderizadas no dispositivo móvel.

3.2. Os tipos de pergunta

Um questionário é composto por uma ou mais perguntas e cada pergunta pode ser de diferentes tipos: texto, numérico, data, múltipla escolha (*radio button*), seleção múltipla (*combo box* e *check bok*), foto, áudio, vídeo, localização, código de barra, porcentagem (controle deslizante ou *slider*) e desenho. Implementaremos ainda outros tipos, pois uma das características mais positivas da plataforma, graças ao seu desenho modular, é a facilidade para se criar e integrar ao sistema novas componentes.

O usuário pode personalizar valores padrão (*default*) para alguns tipos de campos a serem coletados, configurar validações para os valores inseridos e definir uma ordem de navegação entre as perguntas dependente dos valores inseridos.

3.3. Compartilhamento de dados

Atualmente, o sistema permite criar 3 classes de formulários: privados, públicos e compartilhados. O formulário **privado** pode ser utilizado apenas pelo seu criador e o formulário **público** pode ser visto por qualquer usuário da plataforma. Já os formulários compartilhados podem ser de dois subtipos: hierárquico e social. No formulário **compartilhado hierárquico**, o seu criador pode convidar, por exemplo, os usuários A e B, mas os dados coletados por A não são visíveis por B e vice-versa. Por sua vez, no formulário **compartilhado social** os usuários A e B podem ver os dados coletados um pelo outro.

Author	Date	Qual é o seu nome?	Quando você nasceu?	Tire uma foto sua!
sbrc2012@maritaca.com	1354120884096	Artindo	Tue Mar 04 00:00:00 BRT 1980	

Figura 5. Editor de Formulários, preenchido.

4. Arquitetura da Solução

O projeto Maritaca substitui com vantagens os modelos de coleta tradicionais baseados em lápis e papel. O desenvolvimento de um aplicativo de coleta móvel de dados permite o armazenamento de informações em meio eletrônico, o que facilita a manipulação das informações, assim como o seu compartilhamento.

O projeto Maritaca foi desenvolvido como uma aplicação de nuvem, prevendo sua utilização como “software como serviço”. Os dados coletados a partir de dispositivos móveis são armazenados na nuvem e são visualizados através de um navegador web.

A versão atual do projeto encontra-se disponível para utilização em <http://maritaca.unifesp.br>. O código fonte da solução e sua documentação associada está disponível, sob a licença GPL3, em <http://sourceforge.net/p/maritaca>.

Nesta seção será apresentada a arquitetura do projeto, suas componentes e o modelo de integração entre as componentes.

4.1. Componentes da Plataforma Maritaca

A Figura 6 apresenta as principais componentes da arquitetura da solução em nuvem e ilustra a relação entre essas componentes. A arquitetura é composta por:

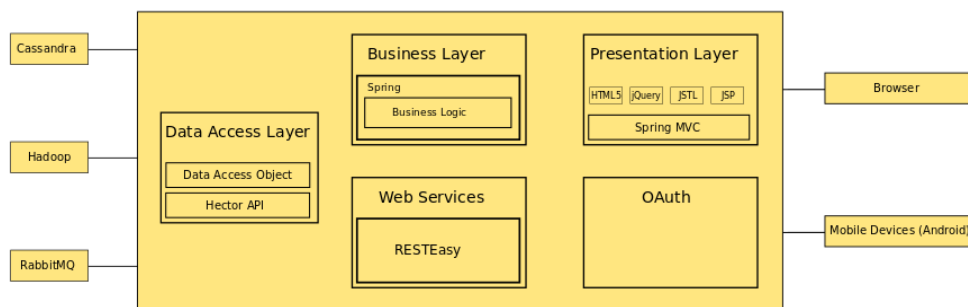


Figura 6. Componentes da solução em nuvem.

- **Servidor de Aplicações:** a parte servidora do projeto utiliza o servidor de aplicações *JBoss* onde hospedam-se os serviços e componentes Web. Toda a parte servidora, exceto alguns *scripts* de manutenção, foram implementados em Java utilizando o arcabouço *Spring* [Tate and Gehtland 2005]. Todos os *webservices* implementados utilizam a abordagem de serviços *RESTful*.
- **Editor de Formulários:** apesar de estar embutido na plataforma Web, o Editor de Formulários é uma aplicação Web independente, escrita em HTML5 e Ajax. Esta componente gera como resultado um **descriptor de questionário**, que é gravado em formato XML.

- **Componente móvel:** é uma aplicação Android que interpreta o arquivo XML e gera as interfaces automaticamente. Trata-se de uma *Engine* de interpretação, baseada no padrão de projeto *Interpreter* [Gamma and Helm 1994].
- **Servidor de Dados Cassandra:** componente para armazenamento escalável de dados estruturados, baseado no paradigma *NO-SQL*.
- **Sistema de arquivos Hadoop:** um sistema de arquivos distribuído e escalável utilizado para armazenar dados não estruturados (imagens, áudio, vídeo etc.).

4.2. Servidor de Aplicações

No módulo servidor encontram-se os serviços e funcionalidades Web do projeto, dentre os quais pode-se citar: armazenamento de dados estruturados e não estruturados (conteúdo multimídia e Apps), os serviços RESTful utilizados para a comunicação entre o servidor e o dispositivo móvel, o editor de formulários etc. O diagrama da Figura 6 mostra os diferentes componentes e arcabouços utilizados no projeto e ilustra a relação entre eles.

A interação entre o usuário e a plataforma Web, via *browser*, é feita através da camada Web do sistema, que utiliza principalmente Spring MVC, JQuery e HTML5. A interação entre dispositivos móveis e a plataforma utiliza-se sempre de serviços RESTful.

O desenho da arquitetura prevê a criação de instâncias dos componentes do sistema (JBoss, Cassandra e Hadoop) em um *cluster* computacional. O balanceamento de carga de requisições web será implementado utilizando-se o módulo mod_JK do servidor de páginas Apache.

4.3. Módulo Móvel

O usuário pode editar os questionários a partir de um navegador padrão e, uma vez editado, o questionário pode ser preenchido a partir do dispositivo móvel, onde os dados ficam temporariamente armazenados até serem transferidos para o servidor.

O Módulo Móvel consiste em uma *engine* que traduz o descritor do questionário (representado em formato XML) em uma hierarquia de objetos instanciados responsáveis pela renderização das interfaces e validação dos dados. O modelo computacional utilizado para representar os questionários é a parte mais sofisticada do componente móvel e foi o desenvolvimento tecnológico que viabilizou a solução.

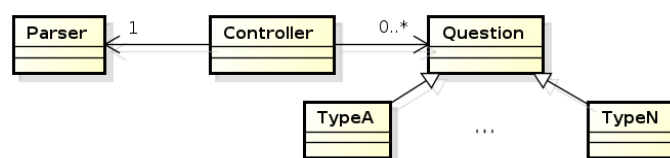


Figura 7. Arquitetura da análise do XML usando o padrão de projeto *Interpreter*

A técnica de mapeamento do XML em uma lista de objetos está inspirada no padrão de projeto *Interpreter* [Gamma and Helm 1994] (vide Figura 7). A aplicação móvel é uma controladora de contexto, que sempre aponta para um objeto em uso, no caso do Maritaca o objeto seria do tipo *Question*.

Para ser utilizada a partir do dispositivo móvel, a aplicação móvel requer que o usuário autentique-se no servidor, deste modo, certifica-se que o usuário tem autorização

para coletar dados para aquele formulário. Esse processo é feito utilizando o arcabouço OAuth [Tassanaviboon and Gong 2011] sobre o protocolo HTTPS.

A Figura 8 ilustra os passos para a autenticação segundo o padrão OAuth: (i) o usuário faz uma requisição de autorização para o servidor, onde é redirecionado para uma tela de autenticação; (ii) se o usuário é autenticado com sucesso, gera-se um código, esse passo é chamado de “confirmação da autorização”; (iii) com este código, o usuário gera um *token*, que necessita ser renovado após um período; (iv) por fim, de posse de um *token* válido para o formulário, o usuário pode coletar, salvar e visualizar os dados, onde a transferência de dados é feita utilizando-se de serviços RESTful.



Figura 8. Modelo de autenticação entre o componente Móvel e o Servidor utilizando OAuth.

A interpretação do XML foi implementada baseando-se no uso do arcabouço Simple⁵ para a serialização e deserialização de arquivos XML. Isto é, este arcabouço converte diretamente arquivos XML em objetos e vice-versa. A vantagem do uso deste arcabouço é devido a simplificação do mapeamento entre o formato XML e o formato dos objetos, simplificando a manutenção do código e criação de novas componentes.

4.4. Formato do arquivo XML de integração

O principal método de integração entre as componentes é o arquivo XML gerado pelo Editor de Formulários, a qual contém as questões do formulário representadas como *tags*. Cada *tag* possui os seguintes atributos básicos: *id*, *next*, *previous*, *required*, *label*, *help* e *type*.

Alguns tipos de *tag* podem possuir estruturas *conditions*, utilizadas para definir a navegação condicional entre as perguntas. A navegação entre as perguntas é definida no rótulo *condition*, onde a resposta da pergunta atual é utilizada para determinar a próxima pergunta a ser exibida. Por exemplo, considere a seguinte questão: “Qual a sua idade?” Se a resposta for um valor inferior a 18 anos, a próxima questão poderia ser: “Qual o nome do seu responsável?” Caso contrário, essa pergunta poderia ser omitida.

Alguns tipos de questões possuem validadores para os dados coletados. Por exemplo, em uma questão numérica, pode-se definir limites mínimos e máximos para as entradas. Desse modo, em uma questão sobre a idade do entrevistado, os valores mínimos e máximos para as respostas podem ser definidos, respectivamente, como 0 e 100. Essa validação evita a coleta de dados errados.

4.5. Captura de dados não usuais: multimídia, localização geográfica etc.

Além de prover a coleta de dados usuais, tais como textos e números, a solução permite também a coleta de dados não usuais, tais como multimídia (áudio, vídeo e imagens) [da Conceição et al. 2008], informações de localização geográfica, desenhos e

⁵Serialização para XML, mais informações em <http://simple.sourceforge.net/>.

códigos de barra etc. Em suma, o questionário poderá conter perguntas tais como: *Qual a sua localização atual? Tire uma foto! Registre o áudio!*

4.6. Construção automática de aplicações

Sempre que um formulário é salvo, o sistema gera um novo aplicativo Android (arquivo em formato APK) e o armazena no sistema de arquivos distribuído Hadoop. Esse processo de compilação e montagem leva alguns segundos, mas como é realizado em *background* não afeta a percepção de usabilidade do usuário. Essa não foi a primeira abordagem adotada, inicialmente planejava-se criar uma única aplicação Android na qual os descritores XML seriam carregados. Entretanto, notou-se que esse método dificultava o processo de divulgação de aplicações e a manutenção das versões das Apps.

5. Descrição da demonstração planejada

Para apresentação do Projeto *Maritaca* é necessário um computador com acesso à Internet para confecção de um formulário. Após criado o formulário, é necessário transferir a aplicação gerada para um dispositivo móvel com suporte a plataforma Android.

6. Conclusões

Este trabalho apresentou o Projeto Maritaca, uma solução aberta para a Coleta Móvel de Dados. Iniciado a mais de quatro anos, o projeto evoluiu de um gerador automático de aplicações para uma solução completa que permite armazenamento e compartilhamento de dados. O Maritaca busca resolver um problema prático: o elevado custo de produção de sistemas móveis. Esperamos que ele seja amplamente utilizado na resolução de problemas cotidianos, contribuindo, assim, para a utilização de sistemas móveis por não especialistas. Atualmente, trabalhamos para que ele permita não apenas a configuração dos questionários, mas também das respostas. Isto é, o usuário poderá configurar os dados a serem coletados e também a forma como estes dados serão exibidos.

Referências

- da Conceição, A., Pereira, R., Rezende, J., Silva, B., Correia, R., Domingues, H., Kon, R., and Kon, F. (2008). Projeto Borboleta: Ferramentas Móveis e Multimídia para Atenção Básica Domiciliar. In *Congresso Brasileiro de Informática em Saúde. Artigo curto*.
- Gamma, E. and Helm, R. (1994). *Design Patterns*. Addison-Wesley Professional.
- Pereira, L. and Da Silva, M. (2009). *Android para desenvolvedores*. Brasport.
- Recordon, D. and Reed, D. (2006). OpenID 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital identity management*, pages 11–16, New York, NY, USA. ACM.
- Rezende, J. V. P., Silva, B. N. M., and da Conceição, A. F. (2010). Plataforma para desenvolvimento simples e flexível de questionários para Coleta Móvel de Dados (CMD). In *I Workshop de Pesquisa e Desenvolvimento em Software Livre (WPeDSL)*, Natal-RN.
- Tassanaviboon, A. and Gong, G. (2011). OAuth and ABE based authorization in semi-trusted cloud computing: aauth. In *Proceedings of the second international workshop on Data intensive computing in the clouds*, DataCloud-SC '11, pages 41–50, New York, NY, USA. ACM.
- Tate, B. and Gehrtland, J. (2005). *Spring: a developer's notebook*. O'Reilly Media, Incorporated.