

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UMA ARQUITETURA DE SOFTWARE BASEADA
EM SERVIÇOS PARA SINALIZAÇÃO DIGITAL
INTERATIVA**

DOUGLAS OLIVEIRA DE FREITAS

ORIENTADOR: PROF. DR. HÉLIO CRESTANA GUARDIA

São Carlos - SP
Maio/2013

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UMA ARQUITETURA DE SOFTWARE BASEADA
EM SERVIÇOS PARA SINALIZAÇÃO DIGITAL
INTERATIVA**

DOUGLAS OLIVEIRA DE FREITAS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Sistemas Distribuídos e Redes.
Orientador: Dr. Hélio Crestana Guardia

São Carlos - SP
Maio/2013

**Ficha catalográfica elaborada pelo DePT da
Biblioteca Comunitária da UFSCar**

F866as Freitas, Douglas Oliveira de.
Uma arquitetura de software baseada em serviços para
sinalização digital interativa / Douglas Oliveira de Freitas. --
São Carlos : UFSCar, 2013.
69 f.

Dissertação (Mestrado) -- Universidade Federal de São
Carlos, 2013.

1. Arquitetura de redes de computador. 2. Computação
ubíqua. 3. Computação móvel. I. Título.

CDD: 004.65 (20ª)

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

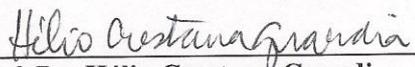
Programa de Pós-Graduação em Ciência da Computação

“Uma Arquitetura de Software Baseada em Serviços para Sinalização Digital Interativa”

Douglas Oliveira de Freitas

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação

Membros da Banca:



Prof. Dr. Hélio Crestana Guardia
(Orientador - DC/UFSCar)



Profa. Dra. Junia Coutinho Anacleto
(DC/UFSCar)



Prof. Dr. Ivan Luiz Marques Ricarte
(DCA/UNICAMP)

São Carlos
Junho/2013

AGRADECIMENTO

A Deus, por me dar força para superar todas as dificuldades, juntamente com os irmãos de São Carlos, que me acompanhavam em suas orações.

Agradeço à minha família e à minha esposa pelo apoio durante todo o período do mestrado, tendo paciência nos momentos que precisei ficar ausente.

Agradeço ao professor Hélio Crestana Guardia, que me orientou durante todo o projeto, dando visões amplas de como desenvolver projeto e torná-lo útil para a sociedade.

RESUMO

O uso de monitores de vídeo como forma de sinalização e apresentação de conteúdos digitais em ambientes de uso coletivo é crescente, com inúmeras finalidades. Além disso, para a maior parte dos cenários de sinalização, o ajuste dinâmico dos conteúdos a apresentar é algo desejável. Informações contextuais, relacionadas ao local de sinalização e ao perfil dos observadores, podem ser levadas em consideração para a seleção de conteúdos e para a interação com usuários. Interações podem ocorrer de forma implícita e transparente, apenas identificando usuários e sistemas mutuamente para a seleção do conteúdo exibido, ou permitindo a troca de dados explícita entre os usuários e o sistema de sinalização. Dispositivos computacionais móveis e tecnologias de comunicação sem fio ampliam as possibilidades de interação com usuários, criando um cenário complexo para exibição de dados públicos e privados. A análise dinâmica de parâmetros de contexto, a obtenção e a geração de conteúdos e a interação com usuários, contudo, requerem processamentos que evidenciam a complexidade da codificação de sistemas de sinalização digital. Este trabalho apresenta uma arquitetura de sistema distribuído e de software baseado em serviços, desenvolvida para a criação de sistemas de sinalização digital interativa. A arquitetura permite a criação e o gerenciamento centralizado, mas escalável, de sistemas de sinalização digital contendo diferentes pontos de apresentação e conjuntos de usuários. Uma implementação do modelo arquitetural e de software proposto foi realizada e os resultados mostram sua capacidade de atender à criação de sistemas de sinalização digital dinâmicos, flexíveis e interativos.

Palavras-chave: sinalização digital, computação móvel, computação ubíqua, interação móvel, computação baseada em serviços, arquitetura baseada em serviço, interação com displays públicos.

ABSTRACT

The use of large displays for digital signage in collective spaces is increasing for numerous purposes. For most signalling scenarios, the dynamic adjustment of the display contents is desirable. Contextual information related to the site and to the signalling profile of observers may be taken into consideration for the selection of content and for user interaction. Interactions can occur implicitly and transparently, by just mutually identifying users and systems for the selection of the content displayed, or by allowing the explicit data exchange between users and the signalling system. Mobile computing devices and wireless technologies extend the possibilities of interaction with users, creating a complex scenario for the exhibition of public and private data. The dynamic analysis of contextual information, the retrieval and generation of presentation data, and the interaction with users all require significant processing which demonstrate the complexity of encoding a digital signage system. This thesis presents a distributed system and service-based software architecture developed for creating interactive digital signage systems. The architecture allows the centralized, but scalable, creation and management of digital signage systems containing different displays and users. An implementation of the proposed architectural model and software was carried out and the results show its ability to serve the creation of dynamic, flexible and interactive digital signage systems.

Keywords: digital signage, mobile computing, ubiquitous computing, mobile interaction, computer-based services, service-based architecture, interactions with public displays.

LISTA DE FIGURAS

Figura 1- Tipos de usuários num espaço público (KAVIANI et al. 2009).....	14
Figura 2- Arquitetura de software do E-Channel System (CLINCH et al., 2011).....	17
Figura 3- Imagens da interface de controle do cliente e do canal (CLINCH et al., 2011).....	18
Figura 4- Arquitetura do Magic Broker (BLACKSTOCK et al. 2010).....	19
Figura 5- Arquitetura do sistema AwareMedia (BARDRAM; HANSEN; SOEGAARD, 2006).....	21
Figura 6- Utilização de serviços na arquitetura BlueYou (ALMEIDA; MOSCHETTO; GUARDIA, 2009).....	22
Figura 7- Arquitetura do Instant Places (JOSÉ et al, 2012).....	24
Figura 8- Suporte a interfaces distribuídas do OUSN. (HOSIO et al., 2010).....	25
Figura 9- Arquitetura do sistema do DHUB.....	30
Figura 10- Arquitetura do software do ponto de apresentação.....	33
Figura 11- Diagrama de sequência da comunicação do ponto de apresentação com o servidor de informação.....	34
Figura 12- Objetos de apresentação do Display Manager.....	35
Figura 13- Arquitetura do software do dispositivo móvel.....	38
Figura 14- Diagrama de sequência geral da comunicação do dispositivo móvel com o ponto de apresentação.....	38
Figura 15: Arquitetura do sistema adaptado para a implementação do protótipo.....	42
Figura 16- Detalhes da comunicação realizada no ponto de apresentação.....	45
Figura 17- Diagrama de classe do sistema no ponto de apresentação.....	46
Figura 18- Componentes para exibição de conteúdo.....	48
Figura 19- Exemplos de componentes simples e complexos para exibição de conteúdo.....	49
Figura 20- Formulário completo de cadastro de serviços no sistema.....	51
Figura 21- Arquitetura do software do dispositivo móvel (implementação).....	52
Figura 22- Interface de interação do dispositivo móvel.....	53

LISTA DE TABELAS

Tabela 1- Tipos de conteúdos exibido em monitores públicos (CLINCH 2011)	11
Tabela 2- Tabela de comparação entre soluções para sistemas de sinalização digital	26
Tabela 3- Tempo de resposta para diferentes interações	54

LISTA DE ABREVIATURAS E SIGLAS

API - *Application Programming Interface*

HTML - *HyperText Markup Language*

HTTPS - *HyperText Transfer Protocol Secure*

JSON - *JavaScript Object Notation*

JVM - *Java Virtual Machine*

IHC - *Interação Humana-Computador*

NUI - *Natural User Interface*

SOA - *Service Oriented Architecture*

TLS - *Transport Layer Security*

WIMP - *Windows, Icons, Menus, Pointing device*

WSDL - *Web Service Definition Language*

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO.....	1
1.1 Considerações Iniciais.....	1
1.2 Cenários do uso de <i>displays digitais</i>	3
1.3 Motivação e Objetivos	5
1.4 Metodologia de Desenvolvimento do Trabalho	6
1.5 Organização do Trabalho	8
1.6 Considerações Finais	8
CAPÍTULO 2 - SINALIZAÇÃO DIGITAL	9
2.1 Considerações Iniciais.....	9
2.2 Sinalização digital.....	9
2.3 Gerenciamento de conteúdo digital.....	11
2.4 Tratamento de contexto.....	12
2.5 Interações com usuário	13
2.6 Trabalhos relacionados	16
2.6.1 E-Channel System	17
2.6.2 Magic Broker	18
2.6.3 AwareMedia	20
2.6.4 BlueYou.....	22
2.6.5 Instant Places.....	23
2.6.6 The Open Urban Service Network (OUSN)	24
2.6.7 Tabela de comparação.....	26
2.7 Considerações Finais	27
CAPÍTULO 3 - ARQUITETURA DE SISTEMAS PARA SINALIZAÇÃO DIGITAL...29	29
3.1 Considerações Iniciais.....	29
3.2 Arquitetura de um sistema de visualização interativa.....	29
3.2.1 Servidor de informação	31
3.2.2 Ponto de apresentação	31
3.2.3 Interação com o <i>display</i>	36
3.3 Avaliação da arquitetura.....	39
3.4 Considerações Finais	40

CAPÍTULO 4 - DESENVOLVIMENTO E RESULTADOS	41
4.1 Considerações Iniciais.....	41
4.2 Desenvolvimento.....	41
4.2.1 Software do ponto de apresentação.....	43
4.2.2 Serviços no ponto de apresentação.....	46
4.2.3 Display Manager.....	48
4.2.4 Criação de sistemas de sinalização digital.....	48
4.2.5 Software para interação via dispositivos móveis.....	52
4.2.6 Avaliação do sistema de interação.....	54
4.3 Avaliação do sistema de sinalização digital interativa.....	55
4.4 Considerações Finais.....	57
CAPÍTULO 5 - CONCLUSÕES	59
5.1 Contribuição.....	59
5.2 Trabalhos Futuros.....	60
REFERÊNCIAS	62
APÊNDICE A	66

Capítulo 1

INTRODUÇÃO

1.1 Considerações Iniciais

Dispositivos computacionais portáteis, como *notebooks*, *tablets* e *smartphones* e a onipresença de redes de acesso à Internet permitem que usuários tenham acesso a dados informatizados em qualquer lugar. Além do acesso à Internet, exemplos de aplicações para esses dispositivos incluem agendas eletrônicas, serviços de mensagens, anotações e uma infinidade de aplicações específicas, facilitando a comunicação entre pessoas.

Além da comunicação individualizada provida pelos dispositivos computacionais pessoais, outra forma de interação digital com usuários tem ganhado atenção recentemente em ambientes coletivos: monitores de vídeo (*displays*), usados para a apresentação de diferentes tipos de mídia.

Comumente chamada de *digital signage* (SCHAEFFLER, 2008), essa forma de sinalização usando mídia digital apresenta enorme potencial para interação estática e dinâmica com os visualizadores das informações exibidas.

A utilização de *displays* digitais em áreas públicas, por exemplo, tem sido adotada para exibir informações úteis para os usuários desses locais, apresentando conteúdos gerais ou mesmo específicos para o contexto em que estão. Em aeroportos e em estações rodoviárias, por exemplo, é possível ver *displays* exibindo tanto conteúdo de entretenimento quanto informações de interesse aos usuários, como horários de partidas, promoções temporárias e propagandas. O dinamismo dessa forma de apresentação a torna bastante interessante, chamando a atenção

dos usuários e favorecendo a divulgação de conteúdos associados a produtos e serviços relevantes em diversos contextos.

Em ambientes empresariais, *displays* digitais são usados para exibir conteúdos gerais, como notícias, comunicados, ou informações específicas da própria empresa em seus diferentes setores. Assim, mais do que servir para a mera exibição repetitiva de diferentes tipos de mídia, esses *displays* têm potencial para atender especificamente, e de forma dinâmica, diversos domínios de aplicação.

De maneira geral, *displays* digitais são usados para informação e entretenimento, possibilitando a exibição de notícias, propagandas e informações de interesse local. Nos casos mais simples, há apenas a apresentação cíclica de imagens e vídeos e cabe ao programa gerente do *display* apenas percorrer uma lista de mídias e exibi-las. A atualização eventual do repositório de mídias, nesses casos, é comumente feita de forma manual e desconectada (*off-line*).

Alternativamente, é possível ter a obtenção dinâmica de conteúdos, recuperados estática ou dinamicamente a partir de repositório(s) externo(s). Nesses casos, requer-se capacidade de comunicação em rede no *display*, incluindo mecanismos de segurança para as transmissões.

Evoluções da forma de apresentação repetitiva incluem a seleção dinâmica dos conteúdos exibidos. Para tanto, pode-se combinar informações sobre os usuários diante do *display*, além de diversas outras informações sobre o contexto local (DEY, 2001), com informações sobre a semântica das mídias disponíveis para apresentação (SHIRKY, 2005). Características de seus dispositivos pessoais de comunicação digital, por exemplo, permitem que pessoas próximas a um sistema computacional sejam eventualmente identificadas.

A descoberta das informações de contexto, contudo, apresenta vários desafios para a infraestrutura computacional no *display*. A classificação de mídias e a combinação de contextos e marcações (*tags*) dessas mídias também tem sido objeto de estudo (KALENDER; DANG; USKUDARLI, 2010). Para a extração de informações contextuais, cabe à infraestrutura de suporte computacional no *display* acomodar a execução de diferentes tipos de serviços de processamento, cuidando de detalhes tanto de suas execuções efetivas, quanto de suas interações em rede e da apresentação dos conteúdos gerados.

Diferentes mecanismos podem ser utilizados para a criação de serviços nos sistemas computacionais associados aos *displays*, incluindo a composição de

funcionalidades usando interfaces padronizadas e bem definidas. *Web Services* (BOOTH et al., 2004) e REST (RICHARDSON; RUBY, 2007) são exemplos de modelos para a criação de serviços e para a interação entre eles.

A possibilidade de interação bidirecional, em que usuários realizam requisições implícita ou explicitamente ao sistema de apresentação de conteúdos, estende as funcionalidades associadas ao uso dos *displays* (VOGEL; BALAKRISHNAN, 2006). Muitas são as aplicações potenciais para essa forma de interação, que pode usar interfaces naturais, com telas sensíveis ou reconhecimento de gestos, e pode também combinar o uso dos *displays* e dispositivos computacionais móveis para apresentação de informações de interesse coletivo e individual.

1.2 Cenários do uso de *displays* digitais

Inúmeras são as aplicações que podem beneficiar-se do modelo de comunicação provido pelos *displays* digitais. O uso de *displays* em locais com vários usuários tem-se tornado comum em ambientes de trabalho, sendo útil para entretenimento e para apresentação de informações específicas, e em ambientes públicos, nos quais o foco é geralmente direcionado à publicidade e à exibição de notícias de interesse coletivo (SCHAEFFLER, 2008). Diferentes trabalhos tratam de aspectos específicos de ajuste de conteúdos, como no caso dos murais digitais para exibição de dados de interesse coletivo (GREENBERG; ROUNDING, 2001; BLY; HARRISON; IRWIN, 1993; HUANG et al., 2002; BARDRAM; HANSEN; SOEGAARD 2006; CLINCH et al., 2011). Há também estudos direcionados à forma de apresentação dos dados, tratando questões de simplicidade e facilidade de visualização (VALKANNOVA et al. 2013). Outros trabalhos tratam especificamente da seleção de conteúdos para exibição (MÜLLER et al., 2008), enquanto outros ainda procuram prover suporte para direcionar informações para exibição ora em *displays*, ora em dispositivos móveis (HOSIO et al., 2010).

Um cenário complexo do uso dessa plataforma pode ser caracterizado num ambiente hospitalar, onde sistemas computacionais para visualização têm

despertado interesse crescente (KUZIEMSKY et al., 2011). Nesse caso, além de exibir informações na forma de diferentes tipos de mídia, como imagens e textos com avisos e notícias de interesse geral, pode-se usar um *display* para apresentação de conteúdo específico aos pacientes ou aos profissionais de saúde localizados próximos à área de visualização.

Ao verificar que um paciente requer atenção, por exemplo, é possível usar um *display* para alertar os profissionais relacionados que estejam mais próximos, sem necessidade de verificar entre os funcionários quem poderia ir até o local onde está o paciente.

Já um profissional de saúde pode observar informações relevantes à sua área ou função de trabalho, dependendo também de dados contextuais de onde o *display* está localizado. Conseguindo identificar um profissional de saúde próximo ao *display*, serviços (programas) ali executados podem recuperar informações da agenda de trabalho desse profissional e emitir alertas sobre eventos relevantes.

Uma forma possível de identificação de usuários próximos ao *display* consiste em usar mecanismos providos pelas tecnologias de transmissão sem fio, como procedimentos de *Inquiry* com a tecnologia Bluetooth (BLUETOOTH SIG, 2009), e varreduras de rede para dispositivos com a tecnologia WiFi (SHIH; BAHL; SINCLAIR, 2002), ou mesmo com o uso de câmeras para identificação facial dos usuários.

Além da interação passiva das pessoas com a plataforma de software associada ao *display*, aplicações específicas no ambiente hospitalar, e em cenários do uso de *displays* de maneira geral, podem envolver ações dos usuários relacionadas à operação da interface de visualização. Interações podem ocorrer, por exemplo, em um tratamento em fisioterapia que exige determinados tipos de movimentos. Neste caso, é possível que um *display* mostre os movimentos que devem ser realizados e ao mesmo tempo monitore os movimentos do paciente, motivando-o a continuar com o tratamento através de mensagens ou de um sistema de pontuação.

Também é possível a um usuário fazer a solicitação explícita de informações que deseja visualizar na tela, repassando ao sistema computacional do *display* indicações da mídia que deve ser obtida para visualização. Em outro caso, pode-se desejar exibir algum arquivo de mídia que se encontre armazenado no dispositivo

móvel do profissional de saúde, que deve ser transferido ao sistema do *display* para observação coletiva.

Também pode ser relevante solicitar a transferência de informações do *display* para um dispositivo móvel pessoal. Pode haver ainda situações em que o profissional deseja usar o sistema computacional associado ao *display* como um ponto de acesso à rede, que atue como intermediário no encaminhamento de informações desse usuário próximo para outro usuário ou sistema remoto. A existência de programas específicos sendo executados nos dispositivos móveis dos usuários é necessária nesse caso, contudo, para permitir a interação com o sistema executado no *display*.

1.3 Motivação e Objetivos

Considerando o suporte computacional para a forma de comunicação aberta, possivelmente pública, e dinâmica, provida pelos *displays*, há muitos aspectos a serem tratados. Requisitos do software incluem desde a implementação de políticas para seleção de conteúdos, baseadas em diferentes informações de contexto, até a criação de mecanismos para interações com usuários e para apresentações efetivas de diferentes tipos de mídias digitais nos *displays* (CLINCH ET AL., 2011).

Também pode ser relevante que sistemas associados aos *displays* ofereçam aos usuários interações ativas, para requisição de conteúdos ou serviços de interesse, que são exibidos na tela de visualização ou enviados aos dispositivos de comunicação móvel desses usuários. Para tanto, mecanismos de comunicação que suportem o uso de diferentes tecnologias de transmissão sem fio são requeridos do sistema computacional associado ao *display*.

Usando aplicações específicas nos dispositivos móveis, pode ser relevante permitir que usuários enviem informações aos sistemas de software executados nos *displays* que, ligados a uma infraestrutura de rede, podem atuar como intermediários no encaminhamento destes dados à Internet ou a outros usuários.

O suporte do *display* para receber dados de forma segura e promover o seus encaminhamentos é complexo. Questões de tecnologia, identificações e autenticações são críticas nesses casos. Além disso, a existência de um mecanismo

genérico de comunicação, possível de ser adequado a diferentes aplicações que realizam comunicação, é desejável.

Aspectos associados à detecção de vários tipos de informações de contexto são relevantes, assim como mecanismos para possibilitar diferentes formas de interação com usuários. O gerenciamento e a autoconfiguração da apresentação de conteúdos, considerando diferentes mídias, no *display* é outro problema relevante a ser tratado.

Assim, muitos são os desafios para a viabilização de sistemas de *display* com toda funcionalidade que essa plataforma pode realizar, principalmente ao considerar-se a criação de sistemas para usos gerais, adaptáveis a diferentes cenários de apresentação.

Nenhum sistema ou plataforma de sinalização existente atualmente oferece suporte a todos os recursos mencionados acima, sendo em geral dedicados a cenários específicos. Assim, a criação de novos sistemas apoiados pela tecnologia existente requer ajustes e modificações significativas, principalmente para novos cenários ou tendo como foto estudos específicos (CHIH; PARKER, 2008, SAS; DIX, 2008).

Nesse sentido, o objetivo deste trabalho é apoiar a criação de sistemas de sinalização digital interativa, que sejam dinâmicos e capazes de fornecer serviços a usuários com a utilização de diferentes recursos e formas de interação. Como resultado, espera-se criar uma arquitetura de sistema e de software que facilite a criação de novas aplicações para este ambiente.

1.4 Metodologia de Desenvolvimento do Trabalho

A partir dos cenários de visualização considerados, diferentes trabalhos relacionados foram avaliados, procurando identificar aspectos relevantes às potenciais aplicações de sistemas de apresentação de conteúdo via *displays*. Como resultado, buscou-se a criação de uma infraestrutura genérica para o desenvolvimento de sistemas de sinalização digital. Diferentes metodologias e tecnologias foram consideradas para o desenvolvimento almejado, com ênfase em funcionalidades, flexibilidade e desempenho. Com a arquitetura de software criada,

foi possível simplificar a criação de aplicações para exibição de diferentes tipos de mídia e a realização de atividades complexas, com geração dinâmica de conteúdo para *displays* em áreas de acesso coletivo.

O desenvolvimento foi dividido em duas partes, sendo uma para a criação de uma infraestrutura para o desenvolvimento de aplicações para uso em *displays* e outra parte voltada à criação de software de suporte para interação de *displays* com o usuário, que pode ser realizada com diferentes tecnologias.

A arquitetura de software para sinalização digital consiste basicamente em um software baseado em serviços, que podem ser ativados e desativados dinamicamente. Do ponto de vista do hardware da plataforma, o dispositivo de visualização é um sistema computacional interligado a um monitor, como um grande *display*, e que contém interfaces de comunicação em rede. De maneira simplificada, os termos **ponto de apresentação**, ou simplesmente ***display***, serão usados nesse texto para referir-se a essa plataforma de hardware e software.

Avaliações foram realizadas para demonstrar as diferentes funcionalidades providas pela arquitetura, tanto para a criação de serviços quanto para a interação com usuários. A integração de sistemas desenvolvidos com a arquitetura proposta e sistemas de informação também pode servir para interações com usuários em sistemas administrativos e em aplicações específicas.

Sistemas desenvolvidos com a arquitetura e infraestrutura criadas também servem para o estudo de mecanismos e procedimentos de interação em espaços públicos, tanto de forma visível aos demais usuários, quanto de forma privada, transmitindo dados e realizando sua visualização em dispositivos pessoais de comunicação digital dos usuários. Avaliações com interação dos usuários poderão ser analisadas com a plataforma, como as realizadas em (SHNEIDERMAN; PLAISANT, 2009). Do estudo de formas de interação previsto foram identificadas demandas adicionais a serem providas pelo *software* de controle dos pontos de apresentação e apoiadas pela plataforma desenvolvida neste trabalho.

1.5 Organização do Trabalho

O Capítulo 2 aborda o uso de *displays* para sinalização digital e trabalhos relacionados para esta plataforma. Tipos de conteúdos que são exibidos e tipos de interação de usuários com *displays* também são abordados neste capítulo, juntamente com tecnologias para desenvolvimento de uma infraestrutura para sinalização digital interativa. O Capítulo 3 apresenta a arquitetura do sistema distribuído proposto, com a descrição de cada elemento com suas respectivas funções. O Capítulo 4 descreve o desenvolvimento de um protótipo funcional a partir da infraestrutura criada e aborda questões de desempenho e usabilidade em estudo exploratório realizado. Por fim, o Capítulo 5 apresenta a conclusão do trabalho, apontando as contribuições e trabalhos futuros.

1.6 Considerações Finais

Neste capítulo foram abordados cenários que podem beneficiar-se do uso de uma plataforma de sinalização digital. Considerando as diferentes necessidades de uso, uma infraestrutura de desenvolvimento de sistemas de sinalização digital necessita ser genérica para que possam ser desenvolvidos serviços específicos para cada cenário, manipulando, gerando ou transmitindo informações entre usuários e aplicações. Este trabalho apresenta uma solução para esta demanda de software identificada.

Nos capítulos seguintes são abordadas algumas soluções de sinalização digital existentes e os recursos que podem ser agrupados para facilitar e enriquecer o uso de sinalização digital.

Capítulo 2

SINALIZAÇÃO DIGITAL

2.1 Considerações Iniciais

Este capítulo aborda diferentes perfis de usuários de acordo com o nível de interação de cada um com sistemas de sinalização digital, os tipos de mídias mais usados para exibição e diferentes soluções existentes para sistemas de sinalização digital relacionadas a este projeto, concluindo com uma tabela criada para comparação entre funcionalidades importantes em ambientes de sinalização.

2.2 Sinalização digital

Propagandas e divulgação de conteúdos tradicionalmente têm sido feitas com pôsteres, faixas, folders e outros meios estáticos. Ao usar sistemas computacionais, passando a exibir conteúdos digitais, essa forma de divulgação recebeu o nome de *digital signage*. Contudo, não há ainda uma definição clara do que exatamente esse termo significa ou de quais são todas as capacidades dessa forma de apresentação de informações digitais.

A descrição mais simples para um *digital signage* seria “um *display* digital gerenciado remotamente, normalmente vinculado com vendas e publicidade” (SCHAEFFLER, 2008).

Harrison define *digital signage* como uma forma de publicidade OOH (*Out-of-Home*) em que o conteúdo, ou as mensagens exibidas numa tela eletrônica, ou o sinal digital, podem ser alterados sem modificações físicas, geralmente com o objetivo de apresentar mensagens em lugares e momentos específicos (HARRISON; ANDRUSIEWICZ, 2003).

Diferentes formas de exibir conteúdos digitais têm surgido. Algumas utilizam monitores de grandes dimensões com baixa ou alta definição, ou uma composição de monitores, formando grades de 8x8 ou 16x16 monitores, podendo assim exibir imagens com grande qualidade. Outras formas de exibir conteúdos incluem até a projeção em vapor de água ou fumaça, substituindo monitores tradicionais, podendo assim utilizar grandes áreas como lagos para exibir filmes, por exemplo.

Os usos de sistemas de sinalização digital são diversos, podendo abranger desde exibição de informações públicas ou privadas de anúncios publicitários a conteúdos interativos, de forma que cada usuário selecione os conteúdos que deseja visualizar.

Com o uso da Internet, o acesso a conteúdos para exibição pode ser feito remotamente, obtendo-os de forma dinâmica a partir de provedores de serviços, de acordo com preferências. Neste caso, não é necessário que administradores de sistemas de sinalização digital atualizem os conteúdos, uma vez que estes podem ser gerenciados pelos provedores de serviço.

Com recursos de monitores com telas sensíveis ao toque (*touch screen*), foi possível que usuários passassem a interagir com os sistemas de sinalização, que fornecem na tela listas de ações que podem ser realizadas.

Em geral, contudo, sistemas de sinalização digital não detectam a presença de pessoas na área de visualização dos *displays*, para exibir conteúdos de maneira automática ou mesmo para que estes possam interagir.

Diante das tecnologias de comunicação sem fio, foi possível que através de dispositivos móveis comuns, como *smartphones*, os usuários passassem a interagir com os sistemas de sinalização, podendo realizar diferentes atividades.

2.3 Gerenciamento de conteúdo digital

O gerenciamento dos conteúdos digitais exibidos no sistema de sinalização é importante para priorizar ou destacar conteúdos que usuários desejam visualizar, de acordo com suas solicitações, ou a partir de informações detectadas.

Os tipos de mídias mais comuns em sistemas de sinalização são imagens e vídeos. Clinch realizou um estudo de caso em sistemas de sinalização digital, identificando os principais conteúdos que os administradores definiram para apresentação em seus sistemas de visualização (CLINCH, 2011). A Tabela 1 mostra os tipos principais de conteúdos exibidos.

Tabela 1- Tipos de conteúdos exibido em monitores públicos (CLINCH 2011)

Tipo de conteúdo	Porcentagem total do conteúdo
Imagens	1494/1796 (83,18 %)
Vídeos	128/1796 (7,30 %)
Páginas web	73/1796 (4,06 %)
<i>Streams</i> de vídeo	6/1796 (0,33 %)

Além destas formas de mídias, a apresentação de mensagens de texto direcionadas aos usuários é comum. Em alguns casos, o software associado ao sistema de sinalização pode ser capaz de gerar imagens a partir das mensagens a apresentar, ampliando a variedade de origens geradoras de conteúdo para exibição.

Uma vez que os conteúdos são exibidos, é necessário definir o tempo em que estes deverão permanecer no *display*, antes da alternância para outro conteúdo. Isso pode ser pré-estabelecido pelo sistema ou definido de acordo com fatores dinâmicos.

Também é possível que *displays* exibam mais de um tipo de conteúdo simultaneamente, dependendo da resolução do monitor utilizado. Para isso, é importante que o sistema possa gerenciar várias fontes de conteúdos e os posicionamentos desses conteúdos na tela.

Com o uso de informações de contexto, por exemplo, é possível que o sistema de sinalização exiba informações determinadas como de interesse aos visualizadores, podendo fazer uso de uma base de dados com preferências locais e

de usuários, conhecidas a priori ou não, ou analisando dados locais coletados de diferentes tipos de sensores.

2.4 Tratamento de contexto

O uso de informação de contexto para definir conteúdos de interesse a usuários tem-se intensificado nos últimos anos (JOSÉ et al, 2012; ALMEIDA; MOSCHETTO; GUARDIA, 2009). Informações de contexto variam desde dados sobre o local onde se encontra o sistema de sinalização até características de identificação de grupos e usuários específicos que visualizam os conteúdos exibidos.

Com relação à localização em que um sistema está instalado, é possível usar informações como idioma padrão, cidade, notícias locais, dados de clima e descrição do prédio, por exemplo, gerando dados específicos de interesse dos usuários locais. Isso pode ser útil em diversos cenários de aplicação de sinalização digital, como em hospitais, fábricas e mercados.

Com um mecanismo de obtenção automática de conteúdos, um sistema de sinalização digital pode buscar informações identificadas como relevantes para serem exibidas de acordo com dados locais, sem necessariamente conhecer informações sobre os visualizadores.

Para que a seleção de conteúdo torne-se mais eficiente, é possível que o sistema colete informações dos usuários locais, através de sensores de registros de interações ou de cadastros que os usuários tenham realizado. Fazer uso de cadastros dos usuários é interessante, pois permite identificar preferências individuais e coletivas. Esse conhecimento possibilita selecionar e exibir informações de interesse mais preciso aos visualizadores.

Um sistema pode detectar a presença de usuários buscando identificação com câmeras ou identificando cada indivíduo através de seus dispositivos móveis, por exemplo, com a interface *Bluetooth* ativada.

Salvando dados sobre as interações de cada usuário com os conteúdos exibidos, é possível ainda analisar preferências individuais, que complementam os registros dos usuários no sistema. Isso permite exibir novos conteúdos de maneira

dinâmica de acordo com horários em que as pessoas visualizam ou interagem com um sistema.

É possível identificar dados de contexto de diferentes formas, auxiliando na seleção automática de informações a serem exibidas ou indicando aos administradores os perfis de usuários para que possam ser inseridos conteúdos específicos.

Na definição de Dey, contexto é qualquer informação que pode ser usada para caracterizar a situação de uma entidade. Uma entidade pode ser uma pessoa, um lugar, ou um objeto que é considerado relevante para a interação entre o usuário e a aplicação, incluindo os próprios usuários ou aplicações (DEY, 2001).

Dey também define de forma genérica o conceito de *context-awareness*, sendo que um sistema é considerado *context-aware* se ele usa dados de contexto para prover informação relevante e/ou serviços para o usuário, onde a relevância depende das atividades desses usuários.

Com isso, todas as informações que são relevantes para a escolha do melhor conteúdo ou serviço para um usuário são consideradas dados de contexto, pois suas características estão relacionadas.

2.5 Interações com usuário

Com o objetivo de prover interação entre os usuários e o sistema de sinalização, é útil identificar os tipos de usuários que observam e interagem com *displays*.

Kaviani realizou um estudo sobre estes tipos de usuários, dividindo-os em três categorias, como pode ser visto na Figura 1 (KAVIANI et al., 2009).

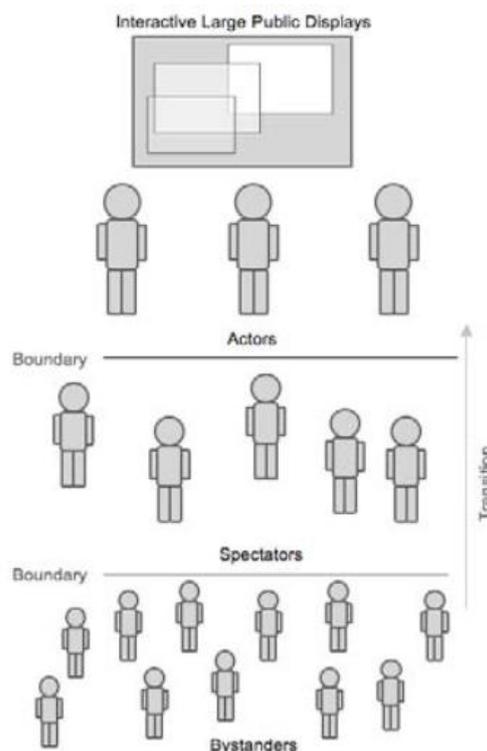


Figura 1- Tipos de usuários num espaço público (KAVIANI et al. 2009)

De acordo com a definição dos autores, *Actors* são usuários que possuem interesse no conteúdo e que interagem com o sistema de apresentação para, por exemplo, solicitação de outros conteúdos. *Spectators* são usuários que possuem interesse no conteúdo, mas não interagem com o sistema; *Bystanders* são usuários que aparentemente, têm contato com o conteúdo exibido mas, aparentemente, têm pouco interesse por ele.

Usuários classificados como *Actors* são propensos a realizarem interações. Já *Spectators*, que possuem interesse em visualizar os conteúdos do display, podem ser incentivados a utilizar os recursos de interação, uma vez que a forma de interação seja simples e intuitiva (JOSÉ et al. 2012). Com a análise de dados de contexto local dos usuários e do ambiente e a adequação dinâmica de conteúdos, é possível que mais *Bystanders* passem a ter interesse pelos conteúdos e, eventualmente, venham a realizar interações com o sistema de visualização.

Há diferentes formas de um usuário interagir com um sistema de sinalização digital. Uma interação pode ocorrer através de gestos, de fala, ou mesmo de forma passiva, com detecção de usuários próximos através de imagens de câmeras, ou usando sensores de dispositivos de comunicação ativos portados. Dispositivos de

comunicação pessoal, como os *smartphones*, também oferecem suporte para diferentes tipos de interação, incluindo formas em que o usuário não se expõe e nem expõe suas ações em público.

Interações normalmente ocorrem quando usuários são incentivados. Müller, por exemplo, pesquisou diferentes formas de atrair usuários para interagirem com *displays*, usando quatro modelos mentais: pôsteres, janelas, espelhos e sobreposição (MÜLLER et al., 2010). Um pôster exhibe conteúdos da forma tradicional, como imagens e textos. Já a janela permite que seja visualizado um ambiente remoto sob um determinado ponto de vista, causando aos usuários a impressão de observarem uma janela. O conceito de espelho é usado para exibir o público visualizador e inserir efeitos visuais no display, como objetos nas mãos dos usuários, por exemplo. A sobreposição, diferentemente das demais formas de interação, usa o ambiente para exibir conteúdos através de projetores, criando aplicações em que o usuário pode “tocar” em objetos ao seu redor.

As diferentes formas de interação apresentam requisitos diferentes do sistema de sinalização digital, possivelmente incorporando o uso de diferentes tecnologias de interação e de comunicação.

Formas de interação que os usuários podem realizar em sistemas de sinalização digital incluem a detecção de visualizadores, o uso de gestos, e até dispositivos específicos que estejam com o usuário.

Com o sistema Kinect, por exemplo, é possível definir a distância dos usuários e os movimentos que são realizados por eles, podendo assim realizar ações específicas para interação com o sistema de sinalização digital. Isso pode ser útil para, por exemplo, escolha de conteúdos ou para a avaliação de informações na tela.

Com o uso de dispositivos móveis, é possível que sejam enviadas informações como texto e imagens diretamente aos usuários próximos ao *display*. Também é possível definir diferentes formas de interação, através do uso de sensores de fala, de gestos e através de teclados ou telas sensíveis a toque. Estes dispositivos podem ser de uso comum aos usuários, como *smartphones* ou *tablets*.

Para que seja possível o uso de diferentes dispositivos móveis em interações, é necessário que haja um padrão de comunicação sem fio entre eles. Algumas tecnologias comuns possibilitam trocas de pacotes para redes sem fio, como WiFi e Bluetooth.

Usando o suporte de comunicação provido pelas tecnologias de transmissão sem fio, o envio de mensagens entre os dispositivos móveis e os sistemas de sinalização digital pode ser padronizado, por exemplo com o uso de XML (*Extensible Markup Language*) (W3C, 2000) ou JSON (JavaScript Object Notation) (CROCHFORD, 2006). Já a comunicação entre o sistema e provedores de conteúdos pode ser feita de diferentes formas, como através da solicitação de serviços Web (Web Services), ou requisições REST, ou através da troca de mensagens usando algum *middleware* com suporte a primitivas do tipo *publish/subscribe*.

2.6 Trabalhos relacionados

De acordo com os cenários de interesse para sinalização digital, diversas soluções de projetos e sistemas puderam ser identificados. Alguns destes trabalhos são caracterizados e apresentados nesta seção.

Tendo examinado diferentes cenários e sistemas de sinalização digital, algumas funcionalidades comumente providas também puderam ser identificadas:

- Visualização de mídias digitais
- Visualização de textos
- Suporte à criação de serviços específicos
- Mecanismos para detecção da proximidade de usuários
- Suporte para interação com usuários próximos
- Envio automático de conteúdo para usuários
- Gerenciamento dinâmico dos conteúdos na tela

Essas funcionalidades podem ser observadas nos trabalhos relacionados apresentados a seguir e, ao final da seção, é exibida uma tabela comparativa entre eles.

2.6.1 E-Channel System

Desenvolvido por Clinch et al., (2011), este trabalho tem como objetivo criar canais de conteúdos para exibição em *displays* de forma simples aos administradores dos sistemas. Usando o conceito de canais, é possível criar fluxos de conteúdo aos quais os *displays* podem associar-se, semelhantemente à sintonização de uma televisão a um canal de transmissão de vídeo. As configurações dos canais e dos displays são feitas através de uma interface web, sendo possível configurá-la remotamente.

A Figura 2 apresenta a arquitetura do sistema que gerencia os canais com seus respectivos conteúdos. Cada *display*, por sua vez, se inscreve aos canais que deseja exibir, usando uma API para obter a agenda de exibição, que indica a ordem e o tempo que cada conteúdo deve ser apresentado.

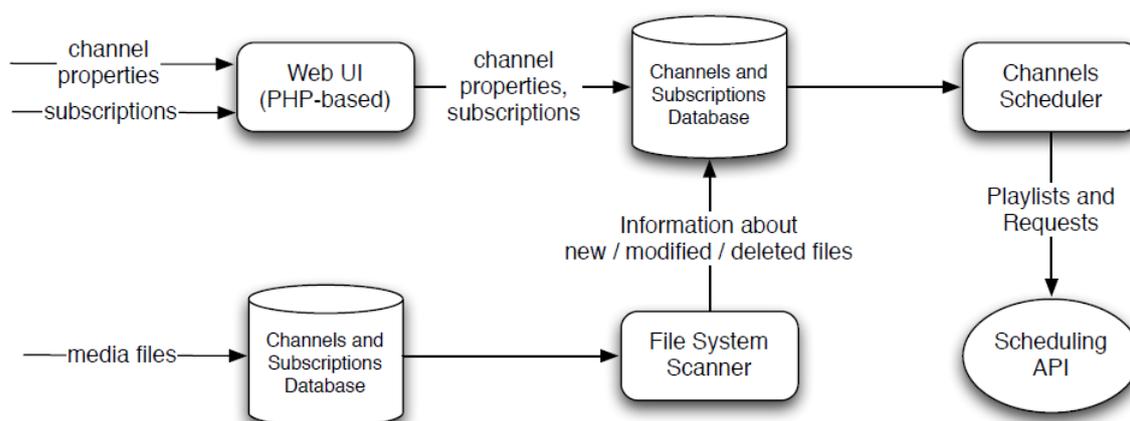


Figura 2- Arquitetura de software do E-Channel System (CLINCH et al., 2011)

Os conteúdos que podem ser inseridos no sistema são imagens, vídeos, páginas web ou fluxos (*streams*) de vídeo. A aplicação que faz a exibição dos conteúdos no *display* é simples, exibindo cada arquivo de mídia durante um intervalo de tempo. O sistema não permite a exibição de mensagens de texto.

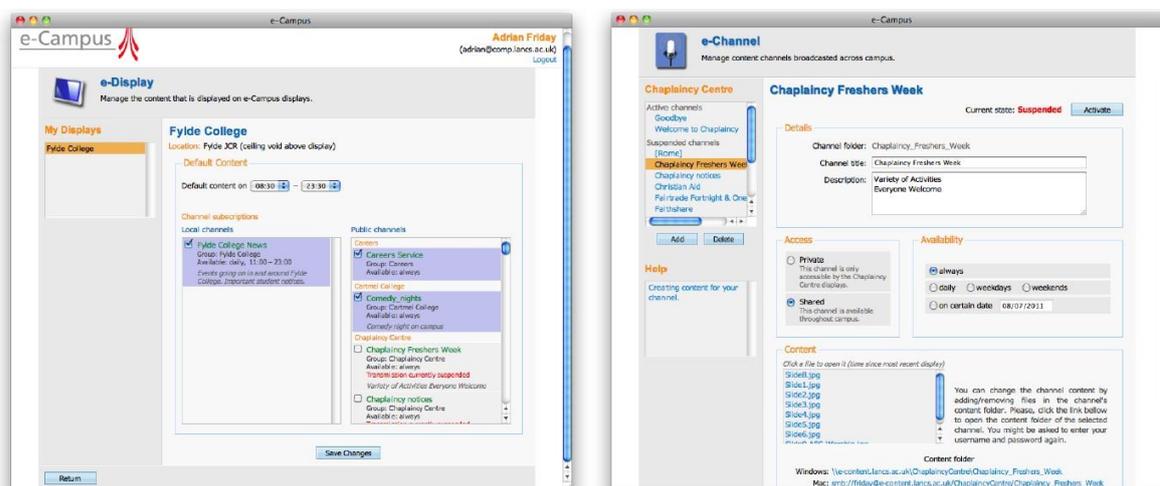
Clinch et al., (2011), os autores também apresentam o resultado de um estudo de caso com a plataforma durante três anos, avaliando o uso da interface e os tipos de conteúdos mais usados. Imagens e vídeos predominaram.

No E-Channel System, o *display* é usado para somente exibir conteúdos, não havendo qualquer tipo de interação com os visualizadores.

O sistema web de configuração é dividido em duas partes, sendo uma para o gerenciamento dos canais e outra para a configuração dos *displays*. Na Figura 3, pode ser observada cada uma destas telas de configuração. Ao criar um canal, são enviados para o servidor os arquivos correspondentes, indicando a ordem e o tempo de exibição (b). Já para configurar o *display* é necessário apenas indicar o canal que este deverá ser exibido (a).

Para exibição de imagens, há um tempo padrão de 10 segundos para cada uma. A sequência de exibição dos arquivos pode ser definida por uma lista ordenada. Esta lista pode ser executada uma ou repetidas vezes.

Uma funcionalidade importante que este trabalho não possui é a criação de serviços e a interação com o usuário, limitando seu uso para apresentação apenas.



(a) Controlling subscriptions on a display.

(b) Controlling channel availability.

Figura 3- Imagens da interface de controle do cliente e do canal (CLINCH et al., 2011)

2.6.2 Magic Broker

Magic Broker (BLACKSTOCK et al., 2010) é um *middleware* para encaminhamento de mensagens no modelo *publish/subscribe*. Usando essa plataforma, os autores implementaram um sistema de sinalização digital interativa, combinando visualização e interação através de mensagens. O sistema oferece suporte para diferentes protocolos de comunicação, sendo o principal baseado em

HTTP/XML, podendo também ter suporte a TCP/IP, SOAP, SMS e protocolos de alto nível, como o Twitter. A Figura 4 apresenta a arquitetura do projeto.

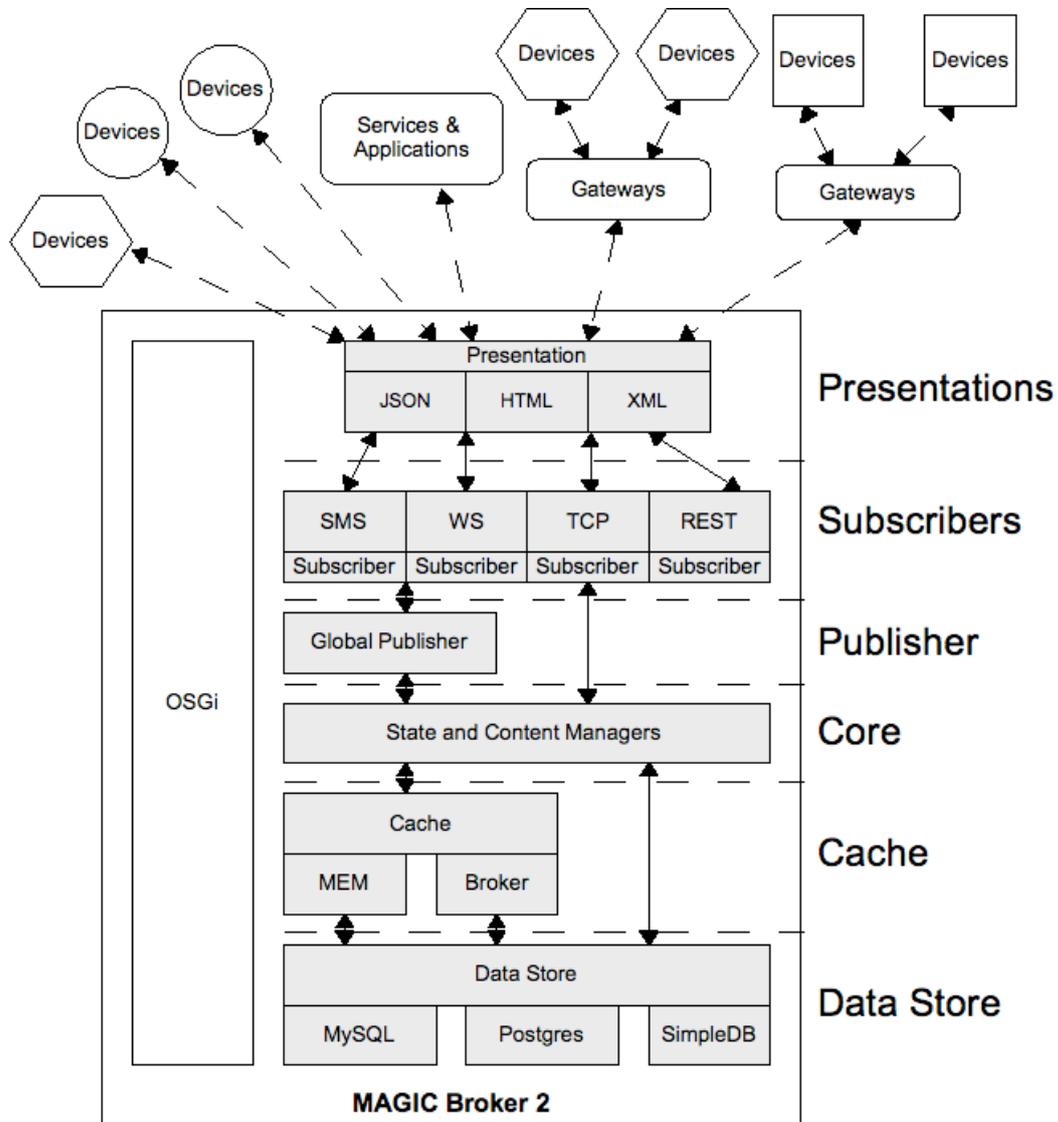


Figura 4- Arquitetura do Magic Broker (BLACKSTOCK et al. 2010)

No sistema de avaliação desenvolvido sobre a plataforma, foram considerados dois modos de interação com os conteúdos do *display*: (1) uso de sensores para detecção de dispositivos *Bluetooth* ou de presença de usuários, permitindo alguma ação no *display*, e (2) interações diretas dos usuários, que enviam requisições para o Magic Broker. O envio de mensagens SMS a partir dos dispositivos pessoais dos usuários foi usado nos testes relatados pelos autores.

Neste sistema, uso do modelo *publish/subscribe* é importante para indicar os serviços aos quais cada *display* tem acesso e para o melhor encaminhamento de mensagens entre o servidor e os *displays* clientes.

No sistema apresentado, não havia mecanismos para criar novos serviços de forma dinâmica. Somente os *displays* podiam subscrever-se aos serviços disponíveis, e os usuários apenas interagem com os serviços disponíveis na tela. Na documentação disponível, não foram fornecidos detalhes das interfaces de interação pelo dispositivo móvel ou feitas considerações de acessibilidade ou adaptação de conteúdos no *display*.

2.6.3 AwareMedia

AwareMedia (BARDRAM; HANSEN; SOEGAARD, 2006) é um sistema distribuído desenvolvido com a plataforma .NET, que consiste em um servidor para sinalização digital e clientes com dispositivos com *displays touch screen*, além de sensores bluetooth. Seu objetivo principal é o monitoramento de cirurgias, transmitindo vídeos e dados de saúde do paciente para coordenadores localizados em outro local do hospital. A Figura 5 apresenta a arquitetura do sistema AwareMedia.

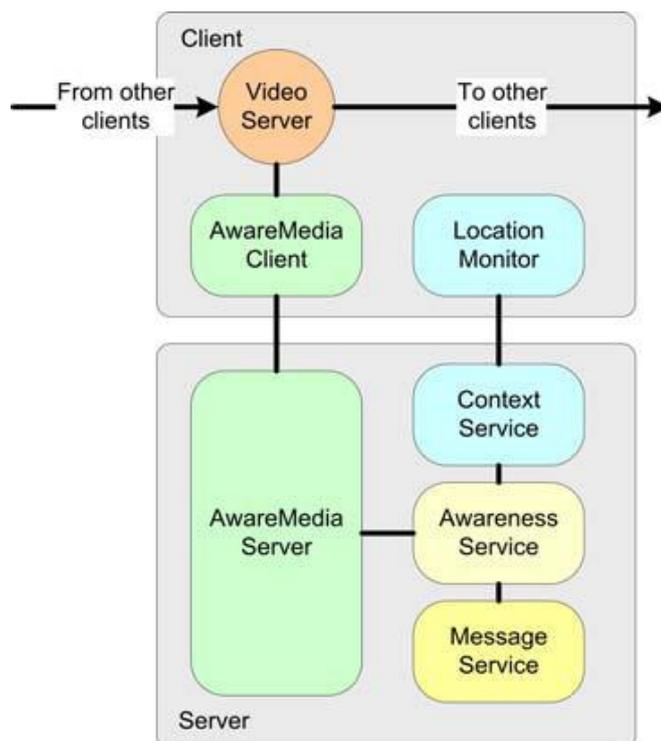


Figura 5- Arquitetura do sistema AwareMedia (BARDRAM; HANSEN; SOEGAARD, 2006)

Neste sistema, a transmissão de vídeo de uma cirurgia é feita utilizando a estratégia *P2P multicasting*.

A entidade Client (Sistema de visualização) realiza busca por dispositivos móveis que possuem *Bluetooth* ou mesmo dispositivos *Bluetooth* avulsos fornecidos pelo hospital para os profissionais de saúde e é possível enviar para a equipe coordenadora os dados temporais do pacientes, como batimentos cardíacos ou pressão.

Com um monitor *touchscreen*, é possível trocar mensagens com a equipe de coordenação.

Além de indicar a localização dos profissionais de saúde, há também uma agenda de cirurgias com suas respectivas equipes técnicas.

Neste projeto, destaca-se a possibilidade de visualizar mais de uma fontes de conteúdo, como *stream* de vídeo de cirurgia, juntamente com os dados temporais do paciente. A possibilidade dos usuários enviarem mensagens de texto para exibição no *display* é importante para a interação.

Uma funcionalidade relevante que não é atendida neste projeto é a interação com dispositivos móveis de usuários, por exemplo, para enviar e receber conteúdos.

2.6.4 BlueYou

O projeto BlueYou é uma arquitetura ciente de contexto para comunicação sem fio usando dispositivos móveis e um *display* (ALMEIDA; MOSCHETTO; GUARDIA, 2009). Com esta arquitetura, os usuários podem receber informações em seus dispositivos móveis de forma automática, indicando previamente a um servidor os serviços de interesse. A Figura 6 apresenta a arquitetura do sistema do projeto e os protocolos de comunicação entre os elementos da arquitetura.

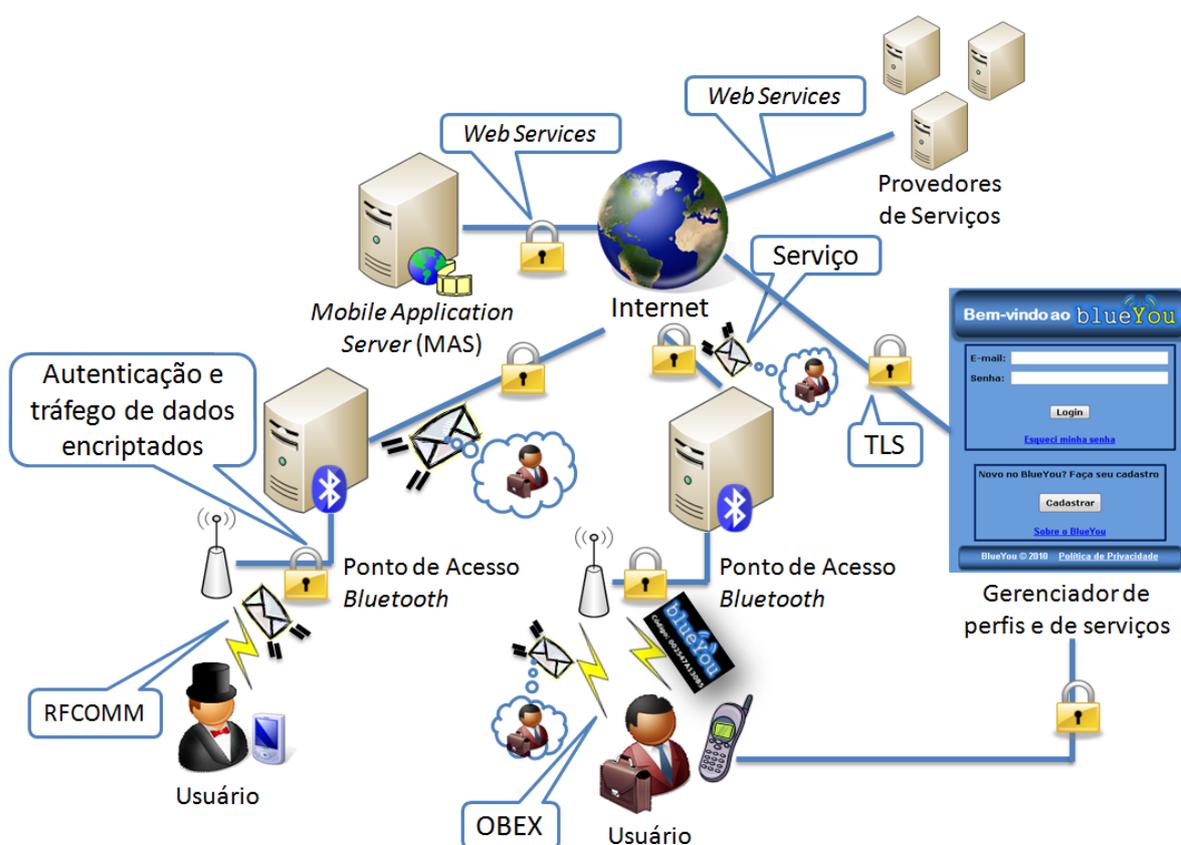


Figura 6- Utilização de serviços na arquitetura BlueYou (ALMEIDA; MOSCHETTO; GUARDIA, 2009)

A arquitetura possui um elemento denominado MAS (*Mobile Application Server*), que gerencia os serviços disponíveis e selecionados por cada usuário. Para tanto, o MAS faz adaptações de conteúdo de acordo com informações de contexto e envia os dados para Pontos de Acesso, que os encaminham aos dispositivos móveis identificados numa região próxima. Toda comunicação feita pelo MAS para execução de serviços é através de *web services*.

Um elemento Ponto de Acesso, por sua vez, possui um monitor para a exibição de conteúdos associados a serviços em execução. Com *layout* estático, não há uma configuração dinâmica dos conteúdos do *display*.

Um usuário interage com o sistema de sinalização digital usando seu dispositivo móvel, enviando e recebendo requisições via *bluetooth*.

Neste sistema, usuários podem receber conteúdos de interesse sem ter que realizar solicitações explícitas ao sistema de sinalização digital, bastando apenas manter ligada a interface de comunicação *Bluetooth*. Além de serviços padronizados para os usuários, uma API desenvolvida permite a criação de serviços especializados, que podem utilizar o *display* para exibir conteúdos e permitir a interação do usuário com os elementos da tela.

A interação entre o sistema e os usuários está associada à apresentação de conteúdo específico no *display* e ao envio personalizado de mídias digitais aos usuários próximos usando a tecnologia *bluetooth*.

2.6.5 Instant Places

Uma dificuldade encontrada na seleção de conteúdos para um *display* é a definição de preferências dos usuários locais. Mesmo com um estudo dos grupos de usuários, observa-se que interesses podem ser dinâmicos ao longo do dia e os conteúdos selecionados podem não ser desejados pelos visualizadores. Isso gera uma manutenção frequente dos conteúdos para manter o público atento ao *display*. Para tratar deste problema, o sistema ***Instant Places*** (JOSÉ et al, 2012) oferece uma solução que permite que os usuários enviem os conteúdos de interesse através de seus próprios dispositivos móveis para o *display*.

Uma constatação dos autores é que mesmo que a plataforma permita que os usuários enviem ou interajam com os conteúdos do *display*, é necessário que a interface de interação seja intuitiva. Müller verificou que os usuários não realizam interações com os *displays* quando não conhecem o procedimento ou o resultado final de sua interação (MÜLLER et al., 2010). Permitindo que os visualizadores interajam com o *display*, é possível que estes insiram conteúdos de interesse ou mesmo avaliem conteúdos que estão sendo exibidos.

Pensando em prover uma interface de interação intuitiva e comum aos usuários, José desenvolveu uma plataforma para envio e avaliação de conteúdos

enviados pelos próprios usuários (JOSÉ et al, 2012). Na Figura 7, pode-se observar a arquitetura do sistema *Instant Places*, que utiliza sistemas de redes sociais, como Facebook, Twitter e Flickr, para que os usuários interajam com os conteúdos exibidos no *display*.

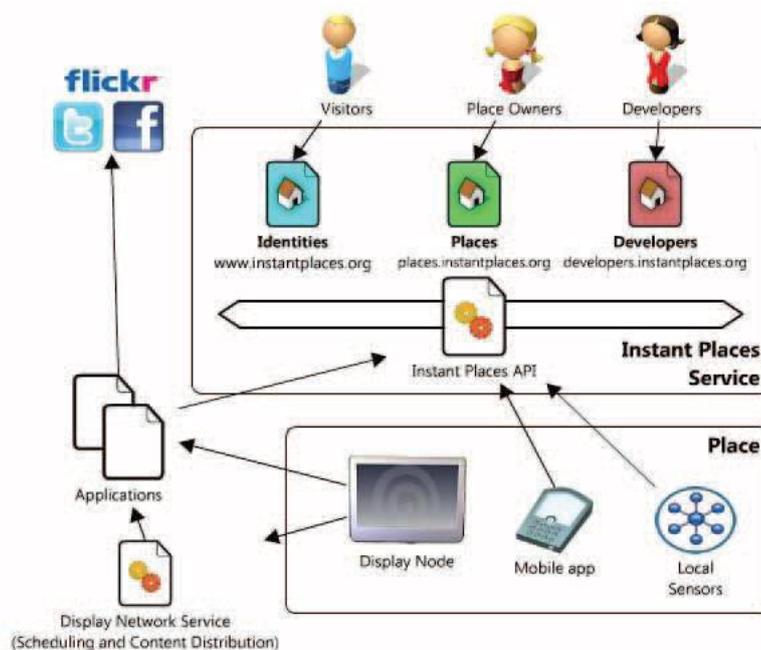


Figura 7- Arquitetura do Instant Places (JOSÉ et al, 2012)

Como resultado do uso das estratégias e dos mecanismos desenvolvidos, observou-se nos testes relatados pelos autores que os usuários não tiveram dificuldades em interagir com o *display*, uma vez que as interfaces utilizadas são comuns a todos os usuários, na faixa etária de 18 a 20 anos.

O trabalho apresentado pelos autores, contudo, não tem suporte para o uso de aplicações genéricas, fora do escopo do uso de redes sociais e não há adaptação de conteúdos exibidos no *display*.

2.6.6 The Open Urban Service Network (OUSN)

Em Hosio et al., (2010), os autores apresentam uma plataforma com suporte a aplicações distribuídas com interfaces de usuário em *displays* públicos interativos e em dispositivos móveis pessoais, tratando do uso de telas públicas e privadas.

Nesse trabalho, os autores argumentam que as soluções de interfaces de sistemas mais comuns utilizam o paradigma “uma tela, uma entrada” (OLSEN, 2007), que não é viável para os tipos de cenários apresentados na sinalização digital. Utilizando o conceito de *Distributed User Interface* (DUI), Hosio et al., (2010), os autores apresentam mecanismos para a criação de aplicações em que interfaces de interação no dispositivo móvel podem enviar requisições a elementos presentes em um *display* público. De maneira complementar, o uso de dispositivos móveis permite que sejam criadas aplicações com conteúdos privados, que são exibidos apenas no dispositivo de um usuário.

Nesse projeto fez-se o uso de um *browser* para exibir conteúdos no sistema de sinalização digital, utilizando recursos HTML 5 (BRANDON, 2011) e Javascript (CROCKFORD, 2008). No dispositivo móvel, foi criada uma aplicação em Java Micro Edition (Java ME) para exibir conteúdo e permitir interação com o *display*. Na Figura 8 é possível observar o modelo de aplicações de suporte com DUIs. Os dispositivos móveis requisitam acesso às aplicações do *display* em sua tela privada, e o servidor de aplicação provê as interfaces de interação para o dispositivo móvel.

Este trabalho não possui uma forma simples de criação de aplicações e não há a detecção de interfaces bluetooth para transferência de dados de forma automática para o dispositivo móvel do usuário.

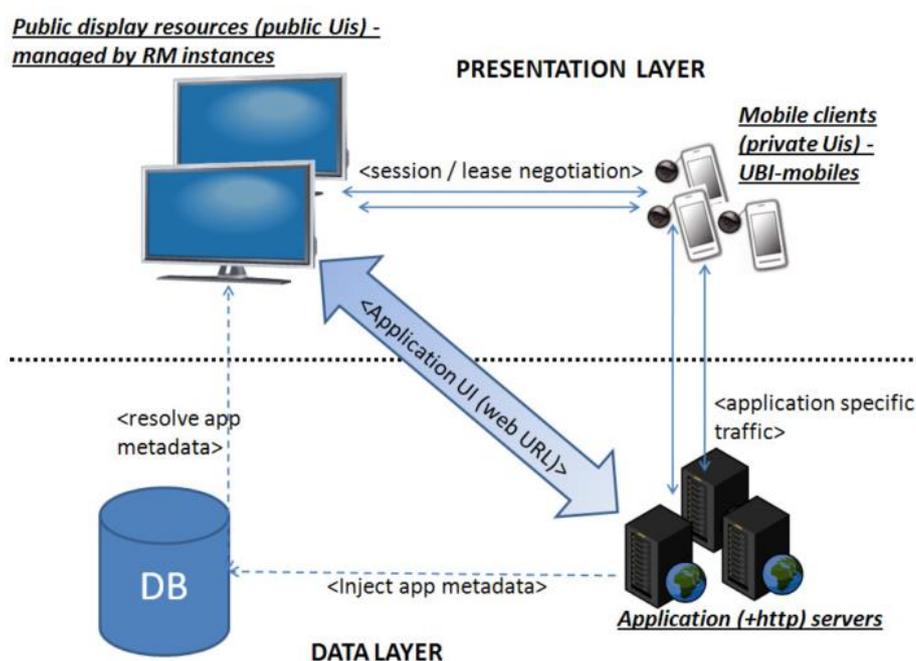


Figura 8- Suporte a interfaces distribuídas do OUSN. (HOSIO et al., 2010)

2.6.7 Tabela de comparação

Os trabalhos relacionados à sinalização digital apresentados foram avaliados e classificados de acordo com os recursos mais relevantes oferecidos em cada sistema. A Tabela 2, apresentada a seguir, mostra uma comparação entre os sistemas e suas funcionalidades, de acordo com os critérios identificados no início desta seção:

1. Visualização de mídias digitais
2. Visualização de textos
3. Suporte à criação de serviços específicos
4. Mecanismos para detecção da proximidade de usuários
5. Suporte para interação com usuários próximos ao *display*
6. Envio automático de conteúdo para usuários
7. Gerenciamento dinâmico dos conteúdos na tela

Tabela 2- Tabela de comparação entre soluções para sistemas de sinalização digital

	Instant Places	AwareMedia	Magic Broker	Blue You	OUSN
1	Sim	Sim	Sim	Sim	Sim
2	Sim	Sim	Sim	Sim	Sim
3	Não	Não	Não	Não	Não
4	Não	Sim	Sim	Sim	Não
5	Sim	Sim	Sim	Sim	Sim
6	Não	Não	Não	Sim	Não
7	Não	Não	Não	Não	Sim

A funcionalidade de criação de serviços indicada se refere à possibilidade de criar novos serviços através de uma interface simples para usuários comuns, ou mesmo para a criação de serviços complexos.

A partir destas comparações, é possível observar características variadas nos sistemas de sinalização digital, que devem ser providas por um software de apoio. Nenhum dos sistemas apresentados possui suporte para todas as características identificadas. Além disso, poucos possuem interfaces (APIs) que possibilitem a criação de recursos adicionais sobre a infraestrutura desenvolvida.

2.7 Considerações Finais

Há diferentes pesquisas em desenvolvimento utilizando sistemas de sinalização digital, com os quais os usuários podem interagir de forma passiva ou ativa.

Nesses sistemas, é possível identificar características e demandas da infraestrutura de software, variando desde a apresentação direta de conteúdo único até a orquestração de serviços variados que geram conteúdos dinamicamente e competem pelo uso do espaço de apresentação e pelos mecanismos para interação com usuários. Uma relação dessas características e de serviços associados pode ser resumida como segue:

- Apresentação de conteúdo, associando o tipo da mídia a programas de exibição específicos;
- Gerenciamento das áreas de apresentação no display;
- Criação de serviços variados para execução no sistema associado ao display e geração dinâmica de conteúdos;
- Leitura de dados através de diferentes dispositivos de entrada, WIMP ou baseados em interfaces naturais (NUIs);
- Comunicação em rede;
- Interação entre processos locais;
- Envio de informações usando dispositivos de entrada e saída;
- Detecção de dispositivos móveis próximos aos displays;
- Coleta de informações de contexto relacionadas ao ambiente, a dispositivos, aos usuários, e outros fatores relevantes;
- Intermediação de comunicação com redes externas;

- Suporte para migração de conteúdos e transferência de domínio de aplicação;
- Suporte à privacidade para exibição de conteúdos em diferentes classes de dispositivos, variando de dispositivos pessoais móveis a grandes telas de visualização coletiva;
- Suporte à segurança nas comunicações, com autenticação e confidencialidade de dados;
- Interface de programação para utilização do suporte disponível para a criação de diferentes aplicações.

Tendo em vista as demandas identificadas, este trabalho apresenta uma arquitetura de software para a criação de sistemas de sinalização digital genéricos, flexíveis e ricos. Os recursos criados atendem às demandas especificadas acima e oferecem uma interface simples para administradores criarem e manterem sistemas de visualização, com serviços dinâmicos e interativos.

A auto organização do uso do espaço de visualização também é considerada.

Capítulo 3

ARQUITETURA DE SISTEMAS PARA SINALIZAÇÃO DIGITAL

3.1 Considerações Iniciais

Os cenários de sinalização digital podem variar de simples exibidores repetitivos de mídias digitais, até ambientes complexos, com apresentação simultânea de vários conteúdos, selecionados dinamicamente em função de informações contextuais.

Também é possível haver interações diretas e indiretas entre os visualizadores e os conteúdos apresentados ou com serviços (programas) executados junto ao *display*.

Para possibilitar a criação desses cenários variados, este trabalho apresenta uma arquitetura de sistema distribuído para a criação de sistemas de sinalização digital e uma arquitetura de software para o software a ser executado em cada elemento deste sistema.

3.2 Arquitetura de um sistema de visualização interativa

A arquitetura de um sistema de sinalização apresentada neste trabalho, denominada DHUB, é baseada num modelo cliente/servidor, conforme ilustrado na Figura 9, e contempla as funcionalidades identificadas na sessão anterior. Assim,

prevê-se a criação de sistemas de sinalização digital interativa compostos de um servidor de informação, de pontos de apresentação (computador contendo um monitor de grande dimensão) e de usuários, que podem interagir com o ponto de apresentação usando diferentes interfaces.

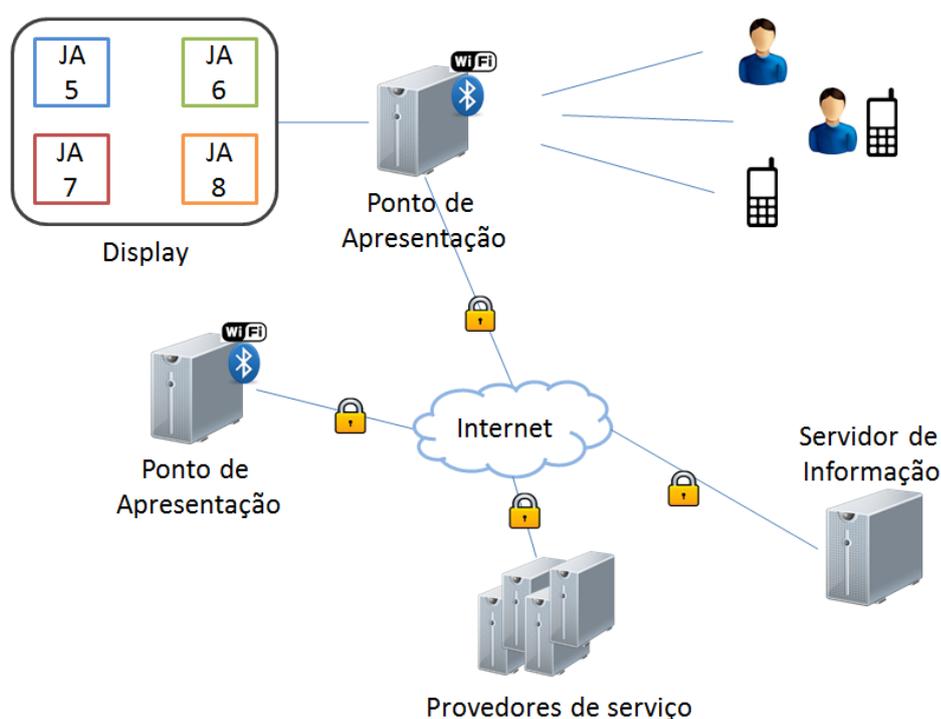


Figura 9- Arquitetura do sistema do DHUB

Neste esquema, cabe ao servidor de informação atuar como um repositório, tanto para a manutenção de informações de configuração dos pontos de apresentação, quanto para intermediar comunicações entre os elementos do sistema. Sua implementação é prevista num modelo baseado em serviços, que também inclui suporte para comunicações no modelo *publish/subscribe*, típico de um *middleware* orientado a mensagens.

O ponto de apresentação representa cada local onde há um *display*, ou seja, uma tela grande para apresentação de mídias digitais. Cada *display* pode ter configurações distintas dos demais, definidas através de serviços estáticos e dinâmicos que executam. Associado a um *display*, há um sistema computacional capaz de processar códigos implementados como serviços, que podem realizar acessos em rede, tanto para comunicação local quanto remota. Serviços do *display* podem também gerar ou obter conteúdos para apresentação. A orquestração dos

serviços ativos e o gerenciamento do uso do espaço de visualização da tela, representado pelo elemento *display* na Figura 9, é responsabilidade do software do ponto de apresentação.

Dispositivos de comunicação dos usuários próximos aos *displays* também podem fazer parte da arquitetura do sistema de sinalização. A simples ativação de interfaces de comunicação sem fio nos dispositivos já tem efeito na identificação do usuário ao qual pertencem. Mais do que isso, esses dispositivos podem hospedar software específico para interação com serviços executados no ponto de apresentação.

Nas sessões a seguir são descritas a arquitetura definida para o desenvolvimento de um sistema de sinalização digital e a arquitetura de software dos elementos previstos neste sistema (servidor, ponto de apresentação e dispositivos de interação com usuário).

3.2.1 Servidor de informação

O servidor de informação é utilizado como repositório de informações gerais do sistema distribuído estabelecido. Entre as informações que armazena estão dados sobre a configuração de cada ponto de apresentação, sobre os serviços que cada ponto de apresentação executa, sobre os usuários e informações relacionadas, e sobre informações coletadas durante a operação do sistema.

3.2.2 Ponto de apresentação

O ponto de apresentação é o componente mais complexo da arquitetura desenvolvida, pois é responsável por gerenciar números variados de serviços associados às funcionalidades locais. Cabe a ele também gerenciar o uso do espaço de apresentação no *display*.

Um sistema de sinalização digital pode exibir diferentes tipos de conteúdo, obtidos estática ou dinamicamente, ou gerados dinamicamente pelos serviços sendo executados no computador associado ao *display*. A orquestração dos serviços de um ponto de apresentação é realizada por seu software de controle. Para que possam ser exibidos conteúdos de diferentes serviços simultaneamente é

necessário que haja também um gerenciamento do uso dos espaços no *display*. Essa atividade também cabe ao software definido nesta arquitetura para o ponto de apresentação.

De maneira geral, o software do ponto de apresentação pode ser visto como um **gerente de serviços** e de **interfaces de apresentação e comunicação**. Para implementar as funcionalidades previstas, dois conceitos (objetos) são empregados: **serviço** e **objeto de apresentação**.

Cada **serviço** contém um código associado, um ou mais canais de entrada e saída e, normalmente, algum objeto de apresentação. Objetos de apresentação estão relacionados a **janelas de apresentação**, que contêm indicação da área de visualização prevista, atributos relacionados a uma espécie de prioridade desta área, uma definição do tipo dos dados gerados para exibição, além de funções (*put* e *get*) de acesso a esses dados.

Para a orquestração dos serviços no ponto de apresentação, alguns serviços essenciais (*core*) são previstos. Estes serviços obrigatórios possibilitam que os demais serviços configurados para o ponto de apresentação exibam conteúdo na tela, interajam com usuários, e enviem informações via rede para outros serviços locais ou remotos. São eles: **Display Manager**, **Communication Manager** e **Interaction Manager**.

Os demais serviços executados no ponto de apresentação estão relacionados às suas funcionalidades referentes a processamentos e à apresentação de mídias digitais, geradas estática ou dinamicamente. Esses serviços são criados pelos administradores do ponto de apresentação (sinalizador digital). Um serviço pode definir conteúdos a serem exibidos, produzidos localmente ou obtidos a partir da interação com outros serviços externos, ou ainda como resultado da interação com usuários próximos ao ponto de apresentação. Provedores de serviço externos podem ser acessados via rede, com comunicação por Web Services / REST ou pelo envio e recebimento de mensagens (*publish/subscribe*). Os mecanismos de comunicação previstos no ponto de apresentação permitem que serviços subscrevam-se a eventos registrados no servidor de informações ou em outros servidores acessíveis via rede.

Os serviços que possuem conteúdos para serem exibidos devem estar associados a objetos de apresentação. Estes, por sua vez, devem estar associados a uma espécie de janela no *display*, denominada **Janela de Apresentação** (JA).

Uma vez definidos os serviços locais a um ponto de apresentação, cada um com seus atributos e possivelmente relações de comunicação e interação, essas informações são armazenadas no servidor de informações. Para a ativação de um ponto de apresentação e execução dos serviços, é necessário consultar o servidor de informação, que fornece a lista de serviços configurados para cada ponto de apresentação. O ponto de apresentação, por sua vez, obtém e executa estes serviços, gerenciando suas execuções e possibilitando suas comunicações com serviços externos e interações com usuários locais.

A Figura 10 apresenta a arquitetura do software previsto em um ponto de apresentação.

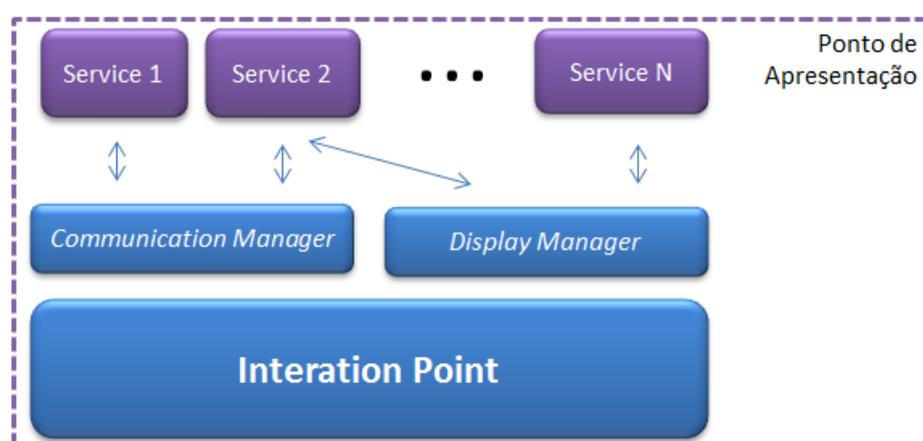


Figura 10- Arquitetura do software do ponto de apresentação

Na Figura 10, é possível observar que o ponto de apresentação possui um serviço de destaque, chamado *Interaction Point*, que é responsável pela interação com as interfaces e dispositivos de comunicação presentes. O ponto de apresentação possui ainda dois serviços, chamados *Display Manager* e *Communication Manager*, responsáveis, respectivamente, pelo gerenciamento do uso do display e por primitivas de comunicação

Serviços locais podem comunicar-se em rede com serviços externos. Para essas comunicações é previsto o suporte do *Communication Manager*. Uma API de comunicação deve ser especificada, contendo funcionalidades para identificação de objetos armazenados no servidor de informação e subscrição e recebimento de notificações de eventos, juntamente com a opção para enviar mensagem para outros

serviços, locais ou externos. Funcionalidades desta API podem ser usadas pelos serviços configurados para o ponto de acesso.

Inicialmente, um ponto de apresentação deve obter a lista de serviços que irá executar. Cada um desses serviços tipicamente comunica-se com o servidor de informação para armazenamento ou para solicitar informações de contexto. No modelo de comunicação previsto, serviços podem ainda registrar e subscrever-se a eventos registrados no servidor, publicando e recebendo notificações pertinentes.

A Figura 11 apresenta de forma genérica um diagrama de seqüência da comunicação do ponto de apresentação com o servidor de informação.

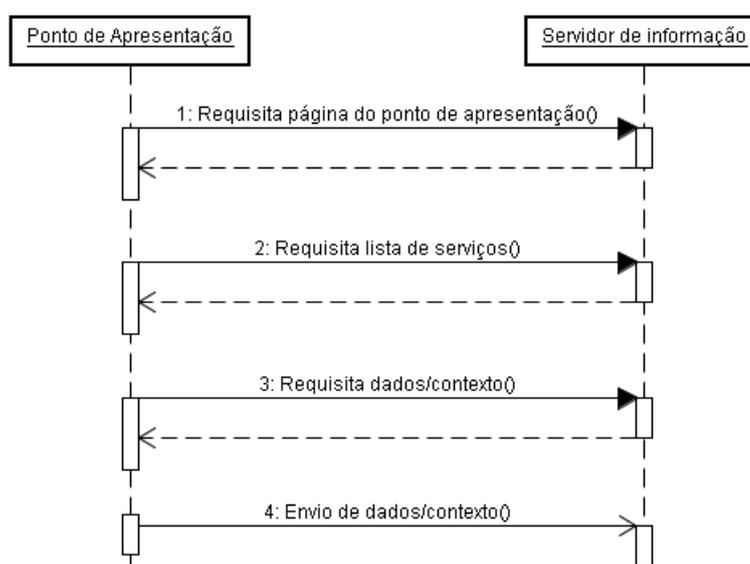


Figura 11- Diagrama de seqüência da comunicação do ponto de apresentação com o servidor de informação

A interação com usuários locais, próximos ao *display*, também requer o uso do *Communication Manager*, mas cabe ao *Interaction Point* tratar das interações efetivas com os dispositivos. Assim, *Interaction Point* é um conjunto de aplicações que permite a comunicação do ponto de apresentação com os dispositivos de interação dos usuários, tanto para encaminhar requisições de interação como para enviar conteúdos ou detectar dispositivos ativos próximos ao display. Isso pode ser feito, por exemplo, através de um serviço que monitore dispositivos com a tecnologia *Bluetooth*.

O uso de interface *Bluetooth* é útil para a detecção de dispositivos através de um procedimento de *Inquiry*, indicando a presença de usuários e, eventualmente,

possibilitando suas identificações, quando há uma associação entre o dispositivo móvel e um visualizador conhecido.

A tecnologia *Bluetooth* também pode ser usada para o envio de conteúdos aos usuários de forma automática, quando estes se aproximam de um ponto de apresentação. Um usuário, por exemplo, pode indicar conteúdos de interesse para que os receba automaticamente ao se aproximar de um ponto de apresentação. Outras formas de apresentação de informação direcionada aos usuários identificados podem ser usadas, por exemplo, publicando um evento que indica a detecção de proximidade de um usuário. Qualquer serviço, local ou remoto, subscrito a este evento pode, então, receber uma notificação apropriada.

Outra função importante para o *Interaction Point* é permitir a interação entre software sendo executado em um dispositivo de usuário com serviços executados no ponto de apresentação. Para isso, o *Interaction Point* atua também como um proxy para tratar questões da tecnologia de implementação.

O *Display Manager*, por sua vez, gerencia os objetos de apresentação associados aos serviços que exibem conteúdo no *display*. Na Figura 12, observa-se a organização do **objeto de apresentação**, que possui serviços associados a ele, para que conteúdos sejam exibidos no *display* através da janela de apresentação.

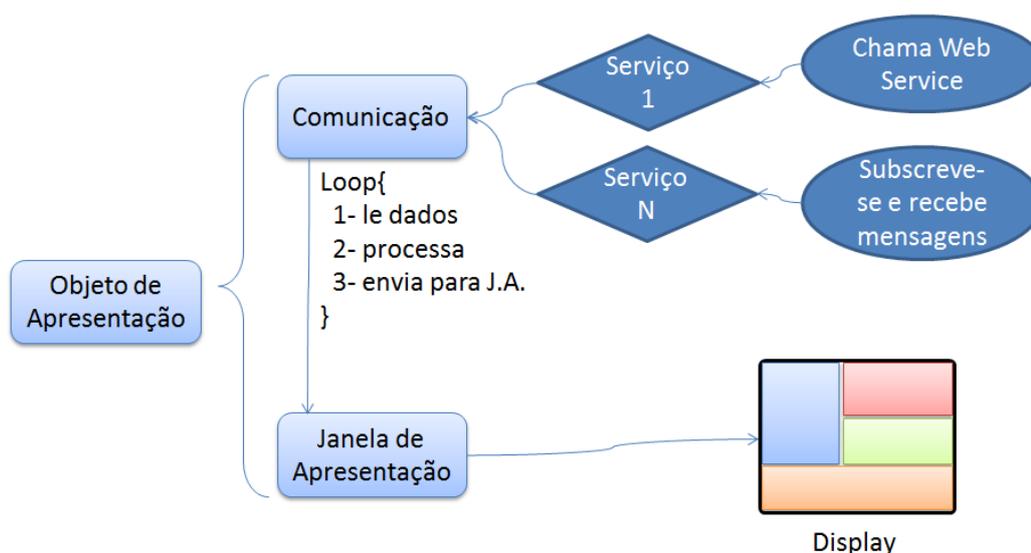


Figura 12- Objetos de apresentação do Display Manager

A janela de apresentação contém parâmetros que são usados pelo *Display Manager* para controlar, por exemplo, o tamanho de uma figura geométrica a ser

exibida e sua posição no *display*. Alguns parâmetros das janelas de apresentação são:

- Posição inicial e tamanho mínimo e máximo da janela no *display*;
- Peso, que pode ser usado para dar mais destaque à janela quando o serviço associado está em evidência, ou apenas para auxiliar em efeitos visuais diversos;
- Tipo dos dados que estão sendo apresentados, permitindo que um componente de visualização compatível seja usado;
- Tipo de transição dos dados, caso haja uma lista de conteúdo a ser exibida, de forma que a janela faça a transição entre eles.

3.2.3 Interação com o *display*

Interações do sistema de sinalização com usuários próximos ao *display* podem ser passivas ou ativas. Interações passivas podem ocorrer através da seleção de conteúdos para exibição considerando as presenças de grupos ou usuários específicos. Também é possível que ações realizadas pelos usuários, voluntária ou involuntariamente, direcionem ações do mecanismo de apresentação. Interações ativas envolvem movimentações dos usuários, interações físicas com dispositivos de entrada de dados, ou mesmo o uso de programas específicos executados em dispositivos móveis, como *smartphones* e *tablets*.

Algumas formas de os sistemas de sinalização digital identificarem usuários próximos incluem o uso de câmeras, detectando grupos de usuários e, se desejado, indivíduos. Grupos de usuários e indivíduos também podem ser identificados por características de comunicação dos seus dispositivos, particularmente por características das interfaces de rede Bluetooth ou WiFi. Programas específicos executados nesses dispositivos também podem prover identificações.

Usuários, identificados ou não, com interesse nas informações apresentadas no *display* podem desejar interagir com o sistema de sinalização. Para isso, a plataforma deve prover diferentes meios de interação, de acordo com as necessidades e preferências dos usuários. As possibilidades de interação envolvem a coleta de dados providos pelos usuários, que são processados e podem influenciar

a seleção e a geração dos dados exibidos no *display* ou gerar dados de retorno individualizado para cada usuário.

De maneira geral, qualquer interação está relacionada a serviços executando no ponto de apresentação, que devem prover interfaces para entrada e saída de dados independente de tecnologia.

Devido a diferentes possibilidades de dispositivos de interação, a arquitetura para sinalização digital criada inclui uma API com funções básicas para que qualquer dispositivo possa ser usado para interação usando as funções pré-estabelecidas.

Um exemplo do uso da API para interação é um cenário onde um usuário seleciona conteúdos do *display*. A ideia é que esta tarefa possa ser realizada de diferentes formas, seja tocando no *display*, movendo a mão em direção ao conteúdo, tendo o movimento detectado por sensores, seja selecionando o conteúdo numa aplicação executada num dispositivo móvel, ou combinando estes tipos de interação. Para todas elas, serviços do ponto de apresentação devem usar dados de entrada recebidos através de funções da API especificada.

Serviços executados no ponto de apresentação podem ou não permitir a interação de usuários usando dispositivos móveis. Alguns serviços podem desejar enviar conteúdos privados para exibição na tela de um dispositivo de usuário. Nos casos mais simples, um mecanismo de encaminhamento direto de dados sobre a tecnologia Bluetooth pode transferir dados para um dispositivo, usando a técnica *objectpush*, ficando a cargo do Sistema Operacional do dispositivo ativar a execução de programa associado ao tipo de arquivo para sua apresentação local. Interações mais específicas podem requerer a execução de programas próprios nos dispositivos dos usuários. Diferentes tecnologias de implementação podem ser usadas para tanto.

Visando a tornar os mecanismos de interação padronizados nos dispositivos de usuário, esta arquitetura prevê um modelo de programa orientado a serviços também nos dispositivos. Uma lógica semelhante à do software do ponto de apresentação é prevista neste caso, como ilustra a Figura 13.

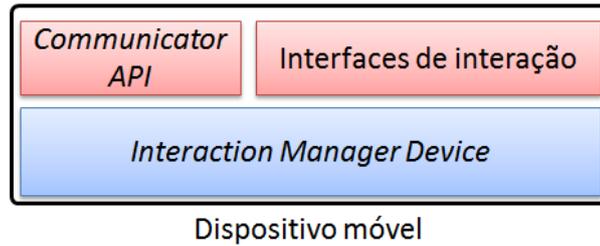


Figura 13- Arquitetura do software do dispositivo móvel

É possível desenvolver interfaces de interação com a criação de aplicações utilizando diferentes tecnologias que apoiem-se numa API comum para comunicação com o ponto de apresentação.

A API de comunicação do dispositivo com o ponto de apresentação deve ser usada como um padrão na comunicação, possuindo versões para diferentes tecnologias de interesse para o desenvolvimentos de aplicações.

A Figura 14 apresenta de forma genérica o diagrama de sequência de uma interação de dispositivo móvel com um ponto de apresentação para comunicação com um serviço selecionado. Neste exemplo, pode-se pensar que o dispositivo móvel, após conectar-se com o ponto de apresentação através do proxy (Ponto de Interação), solicita os serviços disponíveis e carrega as interfaces de interação selecionadas pelo usuário, podendo assim enviar requisições a um serviço específico.

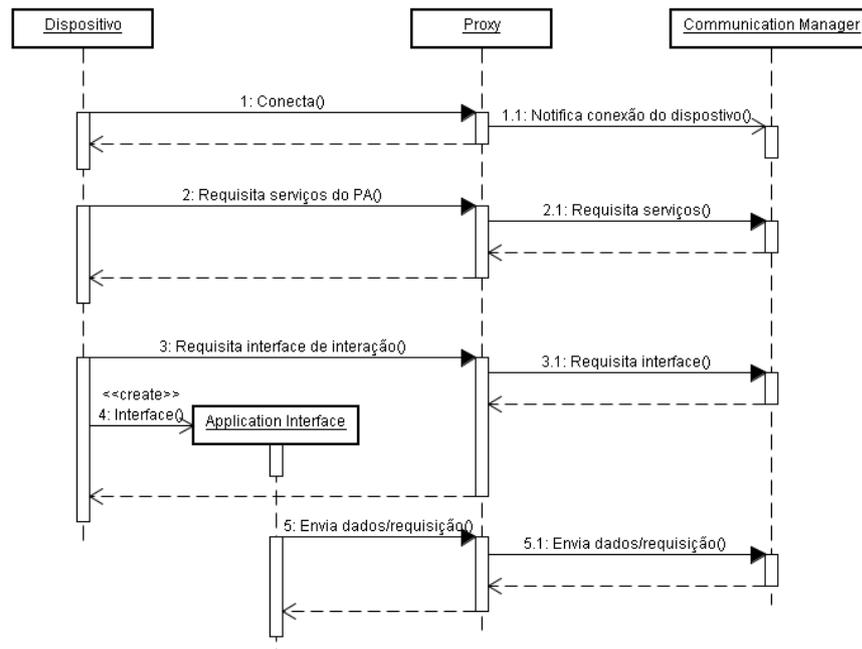


Figura 14- Diagrama de sequência geral da comunicação do dispositivo móvel com o ponto de apresentação

3.3 Avaliação da arquitetura

Após as definições da arquitetura de sistema distribuído, baseada no modelo cliente/servidor, e da arquitetura do software em cada elemento deste sistema, baseada em serviços, buscou-se avaliar se as funcionalidades desejadas eram atendidas em um possível sistema de sinalização digital. Esse procedimento ocorreu de forma cíclica com o refinamento da proposta, culminando nos modelos apresentados neste capítulo.

Por exemplo, considerando os cenários de sinalização digital e suas funcionalidades, descritos nos capítulos anteriores, foram feitos ajustes para possibilitar flexibilidade na configuração de aspectos de usabilidade e acessibilidade. Outro fator importante para a definição da arquitetura foi permitir que a exibição de conteúdos no *display* fosse feita de forma semelhante em dispositivos móveis dos usuários, facilitando o desenvolvimento de serviços a serem incorporados ao ponto de apresentação, principalmente aqueles com interação.

Outro aspecto considerado relevante durante as avaliações foi a possibilidade de criar componentes de exibição específicos, a partir de componentes básicos, ou usando uma API de comunicação associada à arquitetura.

Alguns requisitos não funcionais importantes associados ao projeto de sistemas de sinalização digital incluem o baixo tempo de resposta nas interações dos usuários. A solução encontrada para uma interação rápida foi a atribuição ao ponto de apresentação das funções de controle das interações dos dispositivos locais, não sendo necessário o encaminhamento de dados via rede até o servidor de informações. Com isso, previu-se melhoras no tempo de resposta e na escalabilidade da arquitetura do sistema distribuído resultante.

3.4 Considerações Finais

A análise de cenários variados previstos para o uso de sistemas de sinalização digital proporcionou que as arquiteturas de sistema e de software especificadas sejam ricas em funcionalidades e genéricas.

O uso de padrões de comunicação e a criação de APIs favorece o desenvolvimento de serviços e interfaces de interação por desenvolvedores sem necessidade de conhecer o funcionamento interno da arquitetura.

Capítulo 4

DESENVOLVIMENTO E RESULTADOS

4.1 Considerações Iniciais

A partir da arquitetura apresentada, foi desenvolvido um protótipo para avaliar as funcionalidades providas e o desempenho de um sistema de sinalização digital que utilize a organização de sistema e de software proposta.

Na implementação realizada, constatou-se a possibilidade de uso de diferentes tecnologias, respeitando a arquitetura de sistema e de software e o modelo de comunicação previsto.

4.2 Desenvolvimento

A Figura 15 apresenta a arquitetura do sistema, adequada para implementação do protótipo.

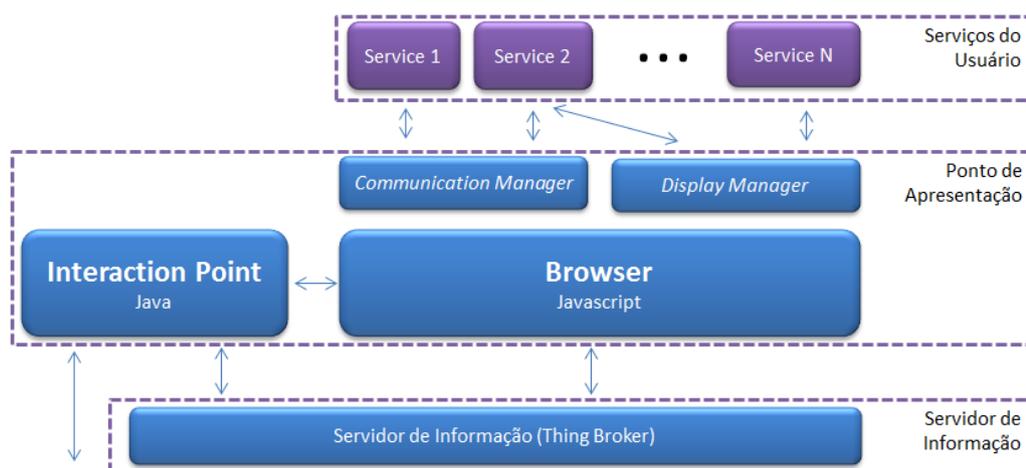


Figura 15: Arquitetura do sistema adaptado para a implementação do protótipo

O protótipo implementado utiliza o sistema *Thing Broker* (CALDERON et al., 2013) para o armazenamento previsto no **servidor de informações**. Comunicações no modelo de serviços, chamadas via protocolo REST, são usadas para armazenar e recuperar diferentes tipos de informações de forma agnóstica aos seus conteúdos. Assim, informações de objetos, como serviços e outras configurações de um ponto de apresentação, e eventos, como a detecção de usuários próximos a um *display*, podem ser armazenadas neste servidor e recuperadas de forma explícita ou através da subscrição a eventos.

Orientado a um modelo de suporte ao conceito de *Internet of Things*, *Thing Broker* é utilizado através de serviços REST, que proveem acesso a informações armazenadas. O servidor foi codificado usando a linguagem de programação Java EE e banco de dados Mongo DB. Mongo DB utiliza o conceito de tabelas hash distribuídas (DHT), que favorecem a escalabilidade no acesso e na recuperação de dados.

Além de oferecer acesso a chamadas de serviços REST, o servidor *Thing Broker* permite o estabelecimento de comunicações no modelo *publish/subscribe*. Desta forma, publicadores registram no servidor os serviços que oferecem, aos quais podem subscrever-se os clientes interessados. À medida que eventos são publicados, através de serviços solicitados ao servidor, clientes subscritos a eles recebem notificações correspondentes. Nesse encaminhamento de informações são oferecidos os modelos PULL, em que um cliente solicita dados de eventos ocorridos, e PUSH, em que o servidor envia notificações automaticamente aos clientes.

Usando o modelo de *Internet of Things*, todos os objetos identificados no servidor *Thing Broker* para representar um sistema de sinalização digital, como pontos de acesso, configurações, usuários, dispositivos e serviços, são identificados de forma única (*things*). Relacionamentos entre *things* são possíveis e *things* podem inscrever-se a eventos associados a outras *things*. Cabe ao servidor *Thing Broker* encaminhar as notificações dos eventos publicados e responder às solicitações de serviço recebidas.

Para casos em que conteúdos são enviados para serviços, o servidor *Thing Broker* verifica as entidades relacionadas (subscritas) ao serviço (evento) que recebeu o conteúdo, notificando-as do evento ocorrido. Publicação de conteúdo e requisição de usuário são exemplos de eventos sobre os quais um serviço ou outra entidade pode receber notificação. Somente as entidades relacionadas (subscritas) receberão notificação da ocorrência de um evento, como a chegada de dados, sem necessidade de verificações periódicas na maior parte dos casos.

No modelo adotado, a configuração de pontos de apresentação, com serviços variados, é simplificada e flexível. Como resultado, tem-se uma arquitetura escalável, podendo haver diversos pontos de apresentação, provedores de serviço e dispositivos móveis num cenário de sinalização digital. A existência de um servidor de informações (*Thing Broker*) centralizado não é um gargalo ao sistema, contudo, dados o modelo de comunicação adotado, orientado a eventos, e o armazenamento distribuído das informações que este gerencia. Para um conjunto grande de pontos de apresentação, é possível utilizar vários servidores de informação. O maior processamento é feito nos pontos de apresentação, na execução de seus serviços e no controle das interações desses serviços com usuários ou com outros serviços remotos.

4.2.1 Software do ponto de apresentação

O **ponto de apresentação** tem como função principal executar serviços dos usuários, permitindo o uso do *display* e das interfaces de comunicação. Outra função importante é o gerenciamento das interações dos usuários com os serviços em execução. As funções do ponto de apresentação foram divididas em duas partes, o *Interaction Point* e *Web Browser*.

Coube ao *browser* executar os serviços obrigatórios: gerenciamento das áreas de exibição do display e comunicação com outros serviços, além de quaisquer serviços específicos definidos para o ponto de apresentação.

Ao *Interaction Point* coube a responsabilidade pelo gerenciamento da comunicação dos serviços locais com usuários, através das interações com seus dispositivos móveis, incluindo a detecção de dispositivos *bluetooth* próximos. O *Interaction Point* foi implementado como um processo Java, externo ao *browser*.

A implementação de parte do software do ponto de apresentação baseando-se em um *web browser* foi pautada por diversos fatores. Primeiro, esta tecnologia permite que o sistema de sinalização digital seja adaptável às tecnologias de visualização de conteúdo, mantendo-o atualizado nos quesitos de apresentação de dados no *display* e no uso de protocolos de comunicação, com suporte provido pelas principais empresas de software para Internet, como Google, Mozilla e Microsoft. Como os componentes gráficos dos conteúdos são tratados em código HTML, a atualização de novos recursos é feita de forma simples, pois é preciso apenas a troca dos arquivos atualizados no servidor, sem necessidade de uma reinicialização do servidor ou ajustes específicos nos pontos de apresentação.

A usabilidade do sistema também é favorecida pela adoção do *browser*, uma vez que, por ser de uso comum dos usuários no dia-a-dia, torna-se fácil o acesso do usuário às interfaces de interação. Com elementos de interface comuns como formulários e botões, é possível desenvolver interfaces intuitivas, visando um maior número de interações dos usuários que se sintam confortáveis para realizar a interação.

Na arquitetura desenvolvida, serviços locais ao ponto de apresentação podem comunicar-se em rede com serviços externos. Considerando uma implementação que tem um *Web Browser* como gerenciador dos serviços, isso pode ocorrer através de solicitações REST via código Ajax. Para comunicações locais, entre os serviços, é preciso o suporte do *Communication Manager*. No caso de implementações baseadas em Browser, isso pode ocorrer diretamente através do mecanismo existente para transmissão de dados entre códigos *javascript*. Por outro lado, há situações que requerem a intermediação explícita do *Communication Manager*. Para tanto, uma API de comunicação foi especificada, contendo funcionalidades

para identificação de objetos armazenados no servidor de informação (*Thing Broker*) e subscrição e recebimento de notificações de eventos.

No Apêndice A – API de comunicação, é possível visualizar mais detalhes da API que foi desenvolvida para este protótipo.

Uma restrição à implementação dos serviços em um *browser* é que, para a comunicação dos usuários com os serviços do ponto de apresentação, é necessário o uso de um *proxy*. Assim, para interações mais específicas, envolvendo a troca de dados entre usuários e os serviços do ponto de apresentação, serviços executados nos dispositivos de usuário estabelecem conexão com o *Interaction Point*, que repassa requisições aos serviços. O *proxy websocket*, implementado como parte das funções do *Interaction Point*, permite que o Browser do ponto de apresentação receba requisições dos dispositivos de interação dos usuários, mantendo uma conexão estabelecida. Por ser uma conexão em rede local, dada a proximidade dos usuários que interagem com o sistema, o tempo de resposta das interações deve ser reduzido.

A Figura 16 apresenta os protocolos de comunicação usados no ponto de apresentação. A comunicação do ponto de apresentação se dá com todos os elementos da arquitetura, como o servidor de informação, provedores de serviço e dispositivos de interação dos usuários.

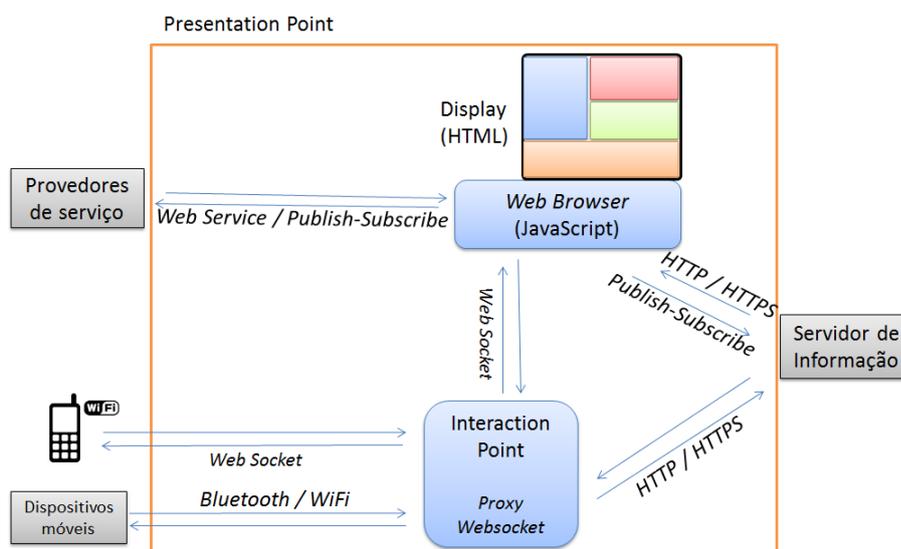


Figura 16- Detalhes da comunicação realizada no ponto de apresentação

Na Figura 16, vê-se o *browser* usado como orquestrador dos serviços no ponto de apresentação e para interações a partir de dispositivos móveis de usuários.

Para tanto, as funcionalidades essenciais previstas na arquitetura do ponto de apresentação e alguns serviços de teste foram implementados usando as tecnologias HTML 5 e JavaScript. Assim, cada serviço definido foi implementado na forma de código JavaScript, organizado na forma de uma página HTML.

A Figura 17 apresenta um diagrama de classe simplificado do sistema de sinalização digital interativo proposto, ilustrando as relações entre os elementos da arquitetura, como serviços, conteúdos, usuários, dispositivos móveis e pontos de apresentação. Segundo o modelo, cada ponto de apresentação contém os serviços associados às mídias digitais que deve apresentar e é possível relacionar os dispositivos móveis aos seus respectivos usuários. Serviços permitem gerações dinâmicas de conteúdos, análises de informações contextuais, e diferentes tipos de interação, como executar ações ou selecionar, trocar, inserir ou remover conteúdo.

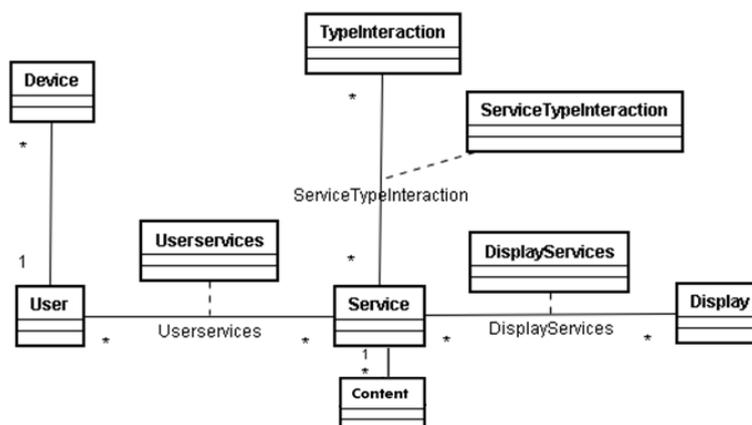


Figura 17- Diagrama de classe do sistema no ponto de apresentação

4.2.2 Serviços no ponto de apresentação

Os elementos *Communication Manager* e *Display Manager* foram implementados na forma de serviços, definidos como códigos javascript numa página HTML acessada pelo programa *browser* executado no ponto de apresentação.

Para poder interagir com funcionalidades de código e realizar acessos a interfaces de rede e dispositivos, o papel do *Interaction Point* foi implementado por uma aplicação Java, fora do *browser*.

As comunicações foram implementadas de acordo com o modelo previsto. Para tanto, mensagens utilizam uma estrutura padrão para identificação do

destinatário e do remetente, do tipo de mensagem e do conteúdo, estruturados no formato JSON (CROCHFORD, 2006). Na implementação realizada, os acessos a diferentes tipos de interfaces de comunicação, como WiFi, Bluetooth e Ethernet são vistos de forma abstrata, independente de tecnologia.

Para exemplificar, podemos usar o ponto de apresentação, que possui diferentes tecnologias para comunicação em rede. O ponto de apresentação se comunica com o servidor de informação usando uma rede diferente da usada para receber interações dos dispositivos móveis. Os dispositivos móveis, por sua vez, podem utilizar a interface Bluetooth para serem detectados pelo ponto de apresentação, podendo enviar conteúdos específicos usando a mesma interface de comunicação. As funcionalidades de um *proxy websocket* foram embutidas no *Interaction Point* e possibilitam a comunicação entre um dispositivo móvel e o ponto de interação. Cabe ao *Communication Manager* gerenciar cada comunicação realizada, independente do tipo de interface de comunicação em uso.

Outro componente relevante implementado parcialmente na validação da arquitetura foi um serviço denominado *Processador de Contexto*, presente no ponto de apresentação. Diferentes informações contextuais podem ser tratadas em um sistema de sinalização digital. Informações do ambiente em que o ponto de apresentação está instalado, por exemplo, podem ser obtidas por um serviço local, possivelmente interagindo com o *Interaction Point* para acesso aos dados externos. Diferentes lógicas de interpretação e manipulação dos dados coletados podem ser empregadas. Para viabilizar o uso desses dados, a arquitetura proposta prevê, dentro do modelo de serviços e interações previsto, que o serviço coletor dos dados os publique no servidor de informações. Desta forma, qualquer serviço que tiver se subscrito ao evento associado a essa publicação poderá ter acesso a esses dados. Cabe, então, à lógica de implementação do ponto de apresentação e seus serviços estabelecer os relacionamentos entre serviços através do modelo de subscrição e publicação de eventos. Depois de processar os dados obtidos, um novo evento poderia ser gerado, por exemplo, para indicar tendências e interesses ao serviço de seleção de conteúdos, subscrito a um evento.

Do mesmo modo, um serviço que detecta a presença de usuários, identificados pela detecção dos seus dispositivos, pode publicar essas informações, que são encaminhadas para serviços que adequam conteúdos e que promovem a

interação com usuários específicos. Para tanto, bastaria que esses serviços fossem subscritos ao evento de detecção de usuários.

4.2.3 Display Manager

A exibição de conteúdo é tratada pelo serviço *Display Manager*, que associa conteúdos a exibir com *Janelas de Apresentação*. A forma como os conteúdos são apresentados depende da especificação dada pelo serviço ou pelo uso padrão do sistema. Para tornar-se genérico, tratando diferentes formas de apresentação e de conteúdo, foi usado o conceito de componente de exibição. Os componentes são responsáveis por exibir conteúdos de diferentes tipos, permitido também ao desenvolvedor criar novos componentes para uso nos serviços que necessitam. A Figura 18 apresenta os componentes para exibição de conteúdo que podem estar associados à Janela de Apresentação.

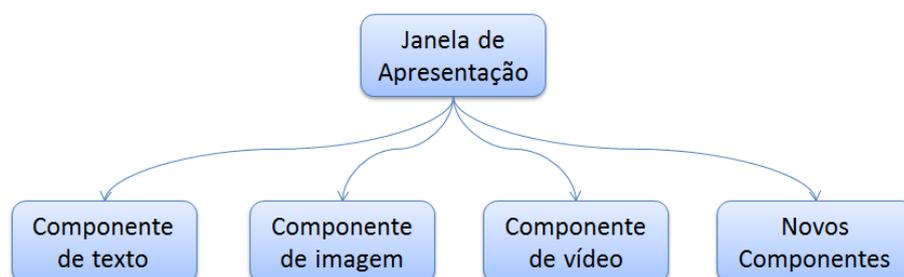


Figura 18- Componentes para exibição de conteúdo

4.2.4 Criação de sistemas de sinalização digital

Para facilitar a criação de sistemas de sinalização digital, criou-se um sistema web, que permite ao administrador definir os pontos de apresentação e os serviços executados em cada um deles. Com esse sistema, o administrador pode cadastrar novos serviços, tornando-os disponíveis para a configuração dos pontos de apresentação.

Inicialmente, há uma definição de componentes de exibição de conteúdo na tela. Um conjunto de componentes pré-definidos está disponível para os serviços. Administradores podem definir novos componentes, de acordo com o tipo de dados

que se deseja exibir numa Janela de Apresentação. Um componente é especificado em linguagem HTML, podendo ser composto por outros componentes, e incluindo a especificação de efeitos em CSS. Na Figura 19, é possível observar um exemplo de componentes sendo exibidos no *display*, cada um associado a uma forma distinta de conteúdo, desde um simples texto ou imagem, até estruturas complexas. No exemplo, vê-se componentes que exibem informações de clima, permitem a criação de desenhos feitos de pontos, e apresentam grupos de imagens posicionadas dentro da região de exibição.



Figura 19- Exemplos de componentes simples e complexos para exibição de conteúdo

O gerenciamento de serviços também é feito através do sistema web definido para configurações. Esse sistema permite a criação de novos serviços e a associação dos serviços aos pontos de apresentação. Um serviço pode atender a diferentes necessidades, permitindo desde a simples exibição de informações textuais numa janela ao qual está associado, até processamentos complexos, que envolvem a obtenção de dados, seus processamentos e a geração dinâmica de conteúdo para apresentação. Um serviço pode ser subscrito a eventos, cujos dados processa e publica na forma de outros eventos. Por exemplo, é possível criar-se um serviço que usa a notificação da presença de usuários para direcionar conteúdos à sua janela de apresentação.

De maneira geral, um serviço deve ser estruturado da seguinte maneira:

- Definição de uma janela de apresentação, especificando o componente a ser utilizado para exibição dos dados no display
- Definição de código em JavaScript. Como também podem estar associados a eventos ocorridos no sistema de gerenciamento de janelas do ambiente gráfico do computador, serviços também podem definir funções para ações associadas a componentes gráficos definidos em sua janela de apresentação. Isso inclui, por exemplo, a possibilidade de ter um “botão” na janela deste serviço e uma função (call-back function) que deve ser executada automaticamente quando esse botão for pressionado, caso o display tenha uma interface touchscreen, ou caso outra forma de interação com a tela esteja disponível.
- Caso tenha interação com usuários, um serviço também pode definir código HTML a ser executado por um browser em dispositivos desses usuários. Caso o serviço de interação tenha componentes gráficos ativáveis, também possível definir serviços JavaScript associados à página de interação para o tratamento de eventos. Esse código HTML também pode incluir códigos de estilo (CSS).

O formulário de cadastro de um serviço é mostrado na Figura 20. Desejando-se criar um serviço simples, basta preencher apenas os campos da coluna esquerda, usando componentes de visualização de conteúdo padrão. Também é possível criar serviços com definições mais complexas, indicando os códigos associados. Para tanto, basta preencher também o formulário da direita, enviando os arquivos de descrição do serviço.

Cadastrar serviço

Nome

Descrição

Palavras chaves

Selecione o tipo de componente para exibir os dados

URL do provedor de conteúdo

Salvar

Possui descrição estrutural?

Arquivo de layout no display No file chosen

Arquivo de eventos no display No file chosen

Possui descrição estrutural no dispositivo?

Arquivo de layout no dispositivo No file chosen

Arquivo de eventos no dispositivo No file chosen

Número de usuários simultâneos

Figura 20- Formulário completo de cadastro de serviços no sistema

Para determinar os serviços que serão executados nos pontos de apresentação, é usado um formulário com a lista de serviços disponíveis, para que sejam marcados os que serão usados. Para cada serviço, é possível ainda inserir informações adicionais, como posição da janela de apresentação na tela ou o tempo de atualização dos conteúdos.

O ponto de apresentação, ao ser iniciado, realiza os seguintes passos:

- Requisita ao servidor de informação os serviços que este deve executar.
- O servidor envia a lista de serviços que devem ser executados, com seus respectivos metadados.
- Com a descrição de cada serviço, o ponto de apresentação requisita os arquivos para execução do serviço no servidor.
- Durante a execução dos serviços, o Display Manager gerencia o espaço de cada serviço no display, modificando-os de acordo com as interações dos usuários.
- Se o serviço permitir a interação pelo usuário, os códigos apropriados ficam disponíveis aos dispositivos na forma de uma estrutura HTML. Caso solicitado, o servidor envia para o dispositivo as especificações para exibição na interface de interação para o serviço selecionado.

4.2.5 Software para interação via dispositivos móveis

O software do dispositivo móvel para interação com o ponto de apresentação foi modificado como pode ser observado na Figura 21.

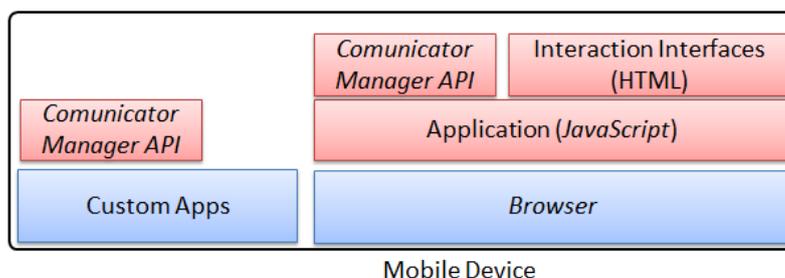


Figura 21- Arquitetura do software do dispositivo móvel (implementação)

O código para interações de usuário, através de seus dispositivos móveis, e serviços no ponto de apresentação foi denominado *Interaction Manager Device*. Esse código foi separado no *Browser* em *Custom Apps*. *Custom Apps* são as aplicações desenvolvidas para a interação do usuário com os serviços de interação, que utilizam recursos específicos das plataformas dos dispositivos móveis. O *Browser*, por sua vez, por ser encontrado em diferentes tipos de plataformas (Android, iPhone, iPad, BlackBerry, Windows Phone), sendo assim possível desenvolver versões únicas de uma aplicação para as diferentes plataformas.

Para dispositivos móveis como *smartphones* e *tablets*, é recomendado o uso de *browser* para que um maior número de dispositivos interaja com os serviços do ponto de apresentação, tornando a aplicação independente de plataforma, uma vez que já existe browsers desenvolvidos para diferentes sistemas operacionais.

Assim, para um dispositivo móvel interagir com o ponto de apresentação utilizando um *browser*, é necessário que este tenha acesso a uma URL. Para isso fez-se o uso de QR Code no *display*, permitindo que os dispositivos móveis leiam e acessem uma página inicial para carregar as interfaces de interação.

Na figura 22 (a), vemos a primeira tela de acesso aos serviços do ponto de apresentação, onde são exibidos todos os serviços disponíveis para interação. Essa tela é montada automaticamente pelo código executado no ponto de apresentação, que percorre a lista dos serviços disponíveis que têm código de interação. Na figura 22 (b), vê-se o resultado ao selecionar-se o serviço de desenho. Neste caso, é

carregada uma área de desenho com algumas funções para limpar a tela e escolher uma cor para desenhar.

Cada serviço pode possuir uma interface independente, sendo possível desenvolver tipos variados de interação.

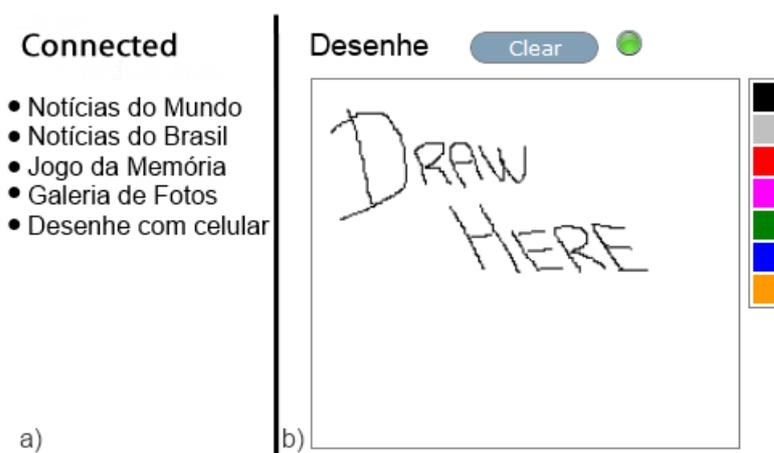


Figura 22- Interface de interação do dispositivo móvel

Neste exemplo, a interface de interação foi criada para *Browser*, implementada com HTML 5 e JavaScript. Com o modelo de interação baseado em um *Browser*, é possível que diferentes dispositivos interajam com o ponto de apresentação, independente do sistema operacional ou do *browser* utilizado, sendo necessário apenas o suporte a conexão *websocket*.

Com o uso de HTML 5, é possível desenvolver interfaces ricas para interação, por esta ser uma linguagem de marcação com muitas bibliotecas de elementos gráficos. Com o uso de uma API para fazer interação com o ponto de apresentação, o desenvolvedor pode construir qualquer tipo de interface com diferentes tecnologias, de acordo com as necessidades dos usuários, provendo acessibilidade e configurações diferenciadas.

Para casos em que é desejado o uso de recursos específicos, como câmera, sensores de movimento do dispositivo, ou o uso do sistema de notificação do dispositivo para alerta de eventos, é possível desenvolver aplicações específicas para cada tipo de dispositivo, com a utilização da API desenvolvida neste projeto para comunicação com o ponto de apresentação (*Communication Manager API*).

4.2.6 Avaliação do sistema de interação

Para que o usuário tenha uma boa experiência na interação com o ponto de apresentação, é necessário calcular o tempo de resposta dos dois tipos de envio de requisições: simples e compostas. As requisições simples são requisições que contêm apenas uma mensagem a ser transmitida, que é processada rapidamente pelo ponto de apresentação. Requisições compostas são formadas de duas ou mais mensagens, que são enviadas para o ponto de apresentação de forma contínua. Um exemplo de requisição composta é o desenho de uma curva no *display*, em que é necessário o envio contínuo da posição do próximo ponto da curva feita pelo usuário.

Em testes realizados, o cálculo do tempo de resposta foi obtido iniciando um contador no dispositivo móvel e enviando uma requisição ao ponto de apresentação. Após o processamento da requisição, o *display* enviava uma mensagem de confirmação ao dispositivo, que parava o contador e informava o tempo total do envio da requisição até o recebimento da mensagem de confirmação do ponto de apresentação.

Como era esperado, o tempo de resposta para as interações dos usuários muda de acordo com o tipo de interação. Em requisições simples, a resposta da interação do *display* foi de aproximadamente 0.1 segundo. Já para requisições compostas, o tempo obtido foi, em média, de 0.3 segundo, dependendo do uso dos recursos do ponto de apresentação por outros serviços. A Tabela 3 apresenta mais informações sobre os tempos de resposta obtidos.

Tabela 3- Tempo de resposta para diferentes interações

	Número de requisições	Tempo de resposta
Requisição simples	1 mensagem	0.1 segundos
Requisição simples	3 mensagens	0.1 segundos
Requisição simples	10 mensagens	0.2 segundos
Requisição composta	10 mensagens	0.2 segundos
Requisição composta	40 mensagens	0.3 segundos
Requisição composta	100 mensagens	0.4 segundos

4.3 Avaliação do sistema de sinalização digital interativa

A implementação realizada do modelo arquitetural desenvolvido para sistemas de sinalização digital foi avaliada através de um conjunto de aplicações teste. Nos testes realizados com o sistema desenvolvido, avaliaram-se as funcionalidades disponíveis e o desempenho no tempo de resposta das interações dos usuários. Para isso, foi necessário criar alguns serviços para demonstração. Foram eles:

- Clima tempo: serviço que exibe informações de clima, como temperatura. Não há interação com o dispositivo móvel por tratar-se de um serviço apenas informativo.
- Notícias do Brasil e do Mundo: serviço que obtém notícias através de provedores de serviço, sem necessidade de ter os conteúdos atualizados pelo administrador. É possível aos usuários interagir com esse serviço alternando entre os conteúdos disponíveis para exibição por este serviço.
- Jogo da memória: serviço de entretenimento, em que mais de um usuário podem interagir simultaneamente com o jogo.
- Lousa mágica: serviço para desenho colaborativo, em que um ou mais usuários podem interagir desenhando numa região da tela.

Observa-se que um sistema de sinalização digital interativa pode ser usado para estudos de diferentes comportamentos dos usuários com grandes *displays* exibindo conteúdos (MÜLLER et al., 2010; KAVIANI et al., 2009). É possível estudar perfis de usuários observando os tipos de conteúdo que estes mais visualizam; a facilidade de uso de diversas interfaces de interações nos dispositivos móveis, analisando padrões utilizados hoje e novas formas de interação; a seleção de conteúdos para exibição com análise de informações de contextos locais; tratamentos que motivam a interação do usuário e observam as ações realizadas, como no caso de um programa de fisioterapia utilizando sistemas de sinalização e sensores de movimento para avaliações médicas (KIRMIZIBAYRAK, 2011).

Os testes realizados com o sistema DHUB desenvolvido destinaram-se a demonstrar as funcionalidades propostas, bem como avaliar o desempenho das interações com os usuários. Análises mais detalhadas, como comportamentos e uso

de recursos específicos de interação podem ser realizados por equipes de pesquisa em IHC.

Com o ponto de apresentação em execução, usuários que passavam próximos ao *display* exibindo as aplicações de teste observavam os conteúdos e podiam visualizar uma imagem com um QR Code na tela. Sentindo-se motivados, esses usuários podiam usar seus dispositivos móveis para carregar a página de interação com o sistema. Instalado num corredor de acesso público, diversos usuários realizaram acesso ao ponto de acesso.

Durante o período de teste, de uma semana, cerca de 10 pessoas interagiram com o sistema e informaram suas opiniões sobre o sistema. De maneira geral, o maior motivo das pessoas terem interagido foi pela facilidade ter o QR Code na tela informando que era possível interagir, juntamente com o serviço de desenho colaborativo que continha a mensagem “Desenhe com seu celular”. Com o serviço de desenho chamando atenção, os usuários descobriram que podiam interagir com outros serviços também.

Com foco no desenvolvimento da arquitetura e numa infraestrutura de validação para as funcionalidades previstas, as análises de uso realizadas foram simples e não trataram questões relacionadas a interações com usuário (IHC).

Do ponto de vista funcional, as aplicações de teste desenvolvidas mostraram-se viáveis e fáceis de serem moldadas à estrutura de código prevista na arquitetura desenvolvida para sistemas de sinalização digital interativa.

Além disso, informalmente, algumas sugestões foram coletadas das impressões dos usuários com o uso do sistema teste:

- Para que o ponto de apresentação chame a atenção das pessoas, é necessário que os *layouts* sejam coloridos, e com algum tipo de animação para que o movimento também atraia os usuários;
- Só foi possível descobrir que os serviços de notícias poderiam receber interação pelo fato de aparecer na lista de serviços no dispositivo móvel;
- As interfaces de interação no dispositivo estavam pequenas, e era difícil de interagir, pois era necessário ampliar a imagem na tela;
- Poderiam ser desenvolvidos serviços em que os usuários deixem mensagens e fotos na tela. Acredita-se que um maior número de pessoas interagiria com

o sistema desta forma, podendo poderiam deixar suas opiniões para o público.

As sugestões dos usuários foram importantes para o desenvolvimento de serviços mais atrativos e intuitivos com os usuários, facilitando o uso pelos mesmos. Os pontos informados são semelhantes aos encontrados em alguns trabalhos relacionados (CLINCH et al., 2011, MÜLLER et al., 2010, JOSÉ et al., 2012, HOSIO et al., 2010).

Muitos dispositivos móveis não puderam realizar interações por seus *browsers* não terem suporte a *websocket*. Para muitos casos, foi indicado o uso do *browser Opera Mobile*, que possui suporte a alguns recursos do protocolo *websocket*, mas que deve ser ativado manualmente.

O protocolo RFC 6455, que especifica o protocolo *websocket*, foi finalizado em dezembro de 2011. Entretanto, ainda não foram desenvolvidas as APIs definitivas para *browsers* dos dispositivos móveis, havendo atualmente apenas versões iniciais para *Opera Mobile* ou *Chrome for Mobile*. *Browsers* de computadores já possuem suporte a grande parte dos recursos de *websocket*.

É possível criar serviços com diferentes funcionalidades, para que diferentes perfis de desenvolvedores possam usar a arquitetura desenvolvida. Grupos de pesquisa de IHC podem desenvolver aplicações específicas para analisar interações com usuários com grandes *displays*, de forma pública ou privada, quando o conteúdo é exibido apenas no dispositivo móvel.

4.4 Considerações Finais

O protótipo da arquitetura foi desenvolvido e testado, realizando cadastros de diferentes tipos de serviços com interações distintas, com o envio de requisições simples e compostas.

O desenvolvimento do sistema de sinalização digital proporcionou a criação de diversos módulos capazes de permitir a generalização do sistema, atendendo às funcionalidades propostas para o sistema.

Os testes realizados com usuários que utilizaram o sistema deram produziram diferentes sugestões para criação de serviços e configurações de *layout* dos

conteúdos, para que o sistema de sinalização chame mais atenção dos visualizadores.

Contudo, ainda é preciso que diferentes tipos de avaliações sejam realizados com o sistema desenvolvido por grupos de pesquisa em IHC e, eventualmente, avaliações mais formais sobre as funcionalidades providas.

Outros sistemas computacionais podem utilizar o sistema de sinalização para exibir conteúdos no *display*, como resultados de mineração de dados em banco de dados, estudos de grafos em redes de conhecimento entre outras fontes de pesquisa.

Todos os recursos identificados para o sistema podem ser desenvolvidos na arquitetura, que é flexível, genérica e amplamente funcional.

Capítulo 5

CONCLUSÕES

5.1 Contribuição

O estudo de diferentes tipos de cenários de sinalização digital interativa permitiram identificar demandas e desenvolver uma arquitetura para sistemas de sinalização genéricos, flexíveis e amplamente funcionais. O protótipo desenvolvido mostrou-se eficiente no ambiente de teste, permitindo a diferentes tipos de dispositivos móveis interagirem com serviços implementados em um ponto de apresentação.

O estudo exploratório realizado com os usuários mostrou que a arquitetura de sistema distribuído e de software nos elementos previstos atende às funcionalidades desejáveis de sistemas de sinalização digital interativa. A implementação do modelo proposto também mostrou que sistemas criados com a plataforma podem ser flexíveis, sendo acrescentadas novas funcionalidades de forma simples, uma vez que o sistema é constituído de módulos. O uso de uma API de comunicação favorece uma forma única de uso dos recursos do sistema e, mesmo com a criação de novas funcionalidades, é possível que os serviços desenvolvidos continuem funcionando.

Por utilizar *browsers* tanto para o ponto de apresentação como para dispositivos móveis, foi possível interagir com o sistema desenvolvido usando a maioria dos *smartphones* atuais. Apesar de muitos *browsers* nativos não terem suporte a *websocket*, essa funcionalidade será desenvolvida em breve e atualizada nesses navegadores, viabilizando a interação por mais dispositivos.

As interações com os dispositivos móveis podem ser feitas usando fluxo contínuo de dados, que possibilita o desenvolvimento de serviços que exijam mais interações com o usuário, como por exemplo, serviço de desenho colaborativo, ou tratamentos médicos aplicados em fisioterapia (FERGUS et al., 2009).

É possível que sejam desenvolvidas aplicações específicas para usos de outros recursos de dispositivos, bastando fazer uso da API de comunicação com o ponto de apresentação.

Diversos grupos de pesquisa podem utilizar o sistema, como grupos de IHC, para o desenvolvimento de aplicações em *displays* públicos de forma simples, bastando conhecer as linguagens HTML e Javascript. Por fazer uso de APIs, não é necessário ter conhecimento profundo de como a arquitetura foi desenvolvida.

Outros tipos de sistemas podem fazer uso da arquitetura para exibir dados, como sistemas de mineração de dados ou coletas de contexto de ambientes utilizando os recursos de interação disponíveis.

A arquitetura desenvolvida mostrou-se adequada para possibilitar a criação de sistemas de sinalização digital interativos com funcionalidades variadas e dinâmicas. O modelo de serviços definido oferece ampla flexibilidade aos sistemas. Do mesmo modo, o mecanismo de interação com usuários é genérico e independente de serviço.

De maneira geral, acredita-se que a arquitetura proposta para o sistema de sinalização e para o software em seus componentes é viável e apropriada.

A flexibilidade e o modelo de comunicação estabelecidos permitem que sistemas de sinalização sejam utilizados em diferentes cenários e sirvam de objeto de estudo para questões pertinentes a várias outras áreas de pesquisa.

5.2 Trabalhos Futuros

O uso de sensores, como detectores de movimento, câmeras e microfones, pode ser mais explorado na plataforma desenvolvida, possivelmente criando uma API genérica para que qualquer novo tipo de dispositivo de interação possa ser utilizado.

Para os conteúdos exibidos na tela, é interessante o uso de efeitos visuais, como, por exemplo, de gravidade e colisões, para aumentar a formas de

visualização de dados, como em caso de mineração de dados, em que há necessidade de visualizar grandes quantidades de dados de forma interativa e agrupada (CHIH; PARKER, 2008).

É importante considerar o design e avaliação de interfaces de interação para dispositivos móveis, para que usuários tenham melhores experiências com o sistema de sinalização digital (SAS; DIX, 2008).

O uso de um proxy *websocket* no servidor de informação é interessante para que os administradores possam obter informações do estado dos pontos de apresentação em tempo real, como quais pontos estão ativos, número de usuários que estão interagindo ou iniciar novos serviços remotamente.

Interações dos dispositivos poderiam utilizar o proxy do servidor de informação para universalização de APIs, porém alguns ambientes podem não permitir acesso à Internet ou podem ser limitados para determinados computadores. Testes de desempenho podem ser realizados com o uso de proxy local e remoto para interação.

É possível avaliar também o uso de serviços (PAPAZOGLU; HEUVEL, 2007), para permitir integrações com outros sistemas, aumentando assim suas funcionalidades.

REFERÊNCIAS

ALMEIDA, R. , MOSCHETTO, D., & GUARDIA, H. Modelo de Disseminação e Compartilhamento de Conteúdo com suporte à Comunicação Oportunística. 2009 *WebMedia* ACM 1-58113-000-0/00/0004

BARDRAM, J. E., HANSEN, T. R., & SOEGAARD, M. AwareMedia - A Shared Interactive Display Supporting Social, Temporal, and Spatial Awareness in Surgery. In: *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. 2006, CSCW '06 (p. 109). New York, New York, USA: ACM Press. doi:10.1145/1180875.1180892

BLACKSTOCK, M., KAVIANI, N., LEA, R., & FRIDAY, A. MAGIC Broker 2: An open and extensible platform for the Internet of Things. In: *2010 Internet of Things (IOT)* (pp. 1–8). IEEE. doi:10.1109/IOT.2010.5678443

BLUETOOTH SIG. IrDA Interoperability. Bluetooth SIG, 2001a. Disponível em: <<http://www.Bluetooth.com/NR/rdonlyres/29E461C6-1A4E-4F09-B971-50717014E8D2/913/OBEX2.pdf>>. Acesso em: 08 fev. 2012.

BLY, S. A., HARRISON, S. R., & IRWIN, S. Media spaces: bringing people together in a video, audio, and computing environment. In: *Communications of the ACM*, 1993, 36(1), 28-46. doi:10.1145/151233.151235

BOOTH, D. et al. Web Services Architecture. Disponível em: <<http://www.w3.org/TR/ws-arch/wsa.pdf>>. Acesso em: 4 fev. 2012.

BRANDON, D. HTML 5 programming: conference tutorial. *J. Comput. Sci. Coll.* 26, 5 (May 2011), 61-61.

CALDERON, R.; BLACKSTOCK; M, LEA, R.; ALMEIDA, R. Developing Cross-Display Applications Using the Really Easy Displays (RED) Framework. In: *The International Symposium on Pervasive Displays*, 2013, California

CHIH, C. H., & PARKER, D. S.: The persuasive phase of visualization. In: *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2008, KDD 08 (pp. 884–892). ACM. doi:10.1145/1401890.1401996

Clements, P., Kazman, R. and Klein, M. Evaluating Software Architectures: Methods and Case Studies. *Addison-Wesley, Boston* [Mass.]; London, 2001

CLINCH, S., DAVIES, N., FRIDAY, A., & EFSTRATIOU, C. Reflections on the long-term use of an experimental digital signage system. In: *Proceedings of the 13th international conference on Ubiquitous computing*. 2011, UbiComp '11 (p. 133). New York, New York, USA: ACM Press. doi:10.1145/2030112.2030132

CROCHFORD, D. The application/json Media Type for JavaScript Object Notation (JSON). Disponível em: <<http://www.ietf.org/rfc/rfc4627.txt?number=4627>>. Acesso em: 10 fev. 2012

- CROCKFORD, D. JavaScript: The Good Parts. *O'Reilly Media*, Inc., 2008.
- DEREK T. SANDERS, J. A. HAMILTON, JR., AND RICHARD A. MACDONALD. Supporting a service-oriented architecture. In: *Proceedings of the 2008 Spring simulation multiconference*. 2008, SpringSim '08. Society for Computer Simulation International, San Diego, CA, USA, 325-334.
- DEY, A. K. Understanding and Using Context. *Personal Ubiquitous Comput.* 5, 1 (January 2001), 4-7. DOI=10.1007/s007790170019 <http://dx.doi.org/10.1007/s007790170019>.
- FERGUS, P., KAFIYAT, K., MERABTI, M., TALEB-BENDIAB, A., & EL RHALIBI, A. Remote physiotherapy treatments using wireless body sensor networks. In: *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing Connecting the World Wirelessly*. 2009, IWCMC '09 (p. 1191). New York, New York, USA: ACM Press. doi:10.1145/1582379.1582640
- GREENBERG, S., & ROUNDING, M. The notification collage. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2001, CHI '01 (pp. 514-521). New York, New York, USA: ACM Press. doi:10.1145/365024.365339
- HARRISON, J. V., & ANDRUSIEWICZ, A. The digital signage exchange. In: *Proceedings of the 4th ACM conference on Electronic commerce*. 2003, EC '03 (p. 274). New York, New York, USA: ACM Press. doi:10.1145/779928.779995
- HOSIO, S., JURMU, M., KUKKA, H., RIEKKI, J., & OJALA, T. Supporting distributed private and public user interfaces in urban environments. In: *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications – 2010, HotMobile '10* (p. 25). New York, New York, USA: ACM Press. doi:10.1145/1734583.1734590
- Huang, E. M., & Mynatt, E. D. Shared Displays for Small Communities : Optimizing for Privacy and Relevance. In: *Position paper and Invited Presentation for the Workshop on Public, Community, and Situated Displays at the ACM Conference on Computer Supported Cooperative Work*. 2002 CSCW
- HUANG, E. M., TULLIO, J., COSTA, T. J., & McCarthy, J. F. Promoting awareness of work activities through peripheral displays. In: *CHI '02 extended abstracts on Human factors in computing systems*. 2002, CHI '02 (p. 648). New York, New York, USA: ACM Press. doi:10.1145/506443.506527
- JOSÉ, R., PINTO, H., SILVA, B., MELRO, A., & RODRIGUES, H. Beyond interaction. In: *Proceedings of the 2012 International Symposium on Pervasive Displays*. 2012, PerDis '12 (pp. 1–6). New York, New York, USA: ACM Press. doi:10.1145/2307798.2307806
- KALENDER, M., DANG, J., & USKUDARLI, S. Semantic TagPrint - Tagging and Indexing Content for Semantic Search and Content Management. In: *IEEE Fourth International Conference on Semantic Computing*. 2010 (pp. 260-267). IEEE. doi:10.1109/ICSC.2010.53
- KAVIANI, N., FINKE, M., FELS, S., LEA, R., & WANG, H. What goes where? Designing interactive large public display applications for mobile device interaction. In: *Proceedings of the First International Conference on Internet Multimedia Computing and Service*. 2009, ICIMCS '09 (p. 129). New York, New York, USA: ACM Press. doi:10.1145/1734605.1734637

KAZMAN R. & KLEIN, M. Designing and analyzing software architectures using ABASs. In: *Proceedings of the 22nd international conference on Software engineering*. 2000, ICSE '00. ACM, New York, NY, USA, 820-. DOI=10.1145/337180.337836 <http://doi.acm.org/10.1145/337180.337836>

KAZMAN, R., BASS, L., ABOWD, G. AND WEBB, M. SAAM: a method for analyzing the properties of software architectures. In: *Proceedings of the 16th international conference on Software engineering*. 1994, ICSE '94. IEEE Computer Society Press, Los Alamitos, CA, USA, 81-90.

KIRMIZIBAYRAK, C., RADEVA, N., WAKID, M., PHILBECK, J., SIBERT, J., & HAHN, J. Evaluation of gesture based interfaces for medical volume visualization tasks. In: *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry – 2011, VRCAI '11* (p. 69). New York, New York, USA: ACM Press. doi:10.1145/2087756.2087764

KÖNIG, W. A., RÄDLE, R., & REITERER, H. Squidy - A Zoomable Design Environment for Natural User Interfaces. In: *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems – 2009, CHI EA '09* (p. 4561). New York, New York, USA: ACM Press. doi:10.1145/1520340.1520700

KUZIEMSKY, C., PEYTON, L., WEBER, J., TOPALOGOU, T., & KESHAVJEE, K. 3rd Annual Workshop on Interoperability and Smart Interactions in Healthcare (ISIH). In *Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research*. 2011 (CASCON '11). IBM Corp., Riverton, NJ, USA, 351-352.

LIEGL, P. The Strategic Impact of Service Oriented Architectures, 2007, 475-484. doi:10.1109/ECBS.2007.71

MICROSYSTEMS, SUN. Sun Academic Introduction to Development with Java FX. 2009, *Pearson Custom Publishing*.

MÜLLER, J., ALT, F., MICHELIS, D., & SCHMIDT, A. Requirements and design space for interactive public displays. In: *Proceedings of the international conference on Multimedia*. 2010, MM '10 (p. 1285). New York, New York, USA: ACM Press. doi:10.1145/1873951.1874203

MÜLLER, J., JENTSCH, M., KRAY, C., KRÜGER, A. Exploring factors that influence the combined use of mobile devices and public displays for pedestrian navigation. In *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges (NordiCHI '08)*, 2008. ACM, New York, NY, USA, 308-317. DOI=10.1145/1463160.1463194 <http://doi.acm.org/10.1145/1463160.1463194>

OLSEN, D. R. Evaluating user interface systems research. In: *Proceedings of the 20th annual ACM symposium on User interface software and technology*. 2007, UIST '07 (p. 251). New York, New York, USA: ACM Press. doi:10.1145/1294211.1294256

PAPAZOGLU, M. P., & HEUVEL, W.-J. Service oriented architectures: approaches, technologies and research issues. In: *The VLDB Journal*, 2007, 16(3), 389-415. doi:10.1007/s00778-007-0044-3

- PERLIN, K., & FOX, D. Pad - An Alternative Approach to the Computer Interface. In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. 1993, SIGGRAPH '93 (pp. 57-64). New York, New York, USA: ACM Press. doi:10.1145/166117.166125
- RICHARDSON, L. & RUBY, S. In: *Restful Web Services (First ed.)*. 2007, O'Reilly
- ROSENBLUM, D. S., & WOLF, A. L. A design framework for Internet-scale event observation and notification. In: *Software Engineering Notes 1997*, ACM SIGSOFT, 22(6), 344–344–360–360. doi:10.1145/267895.267920SAS, C. & DIX, A. (2007). *Designing and Evaluation Mobile Phone-Based Interaction with Public Displays*. CHI'07. San Jose, USA
- SCHAEFFLER, J. *Digital Signage: Software, Networks, Advertising, and Displays A Primer for Understanding the Business*. In: Focal Press. 2008 Book
- SHIH, E., BAHL, P., & SINCLAIR, M. J. Wake on wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices. In: *Proceedings of the 8th annual international conference on Mobile computing and networking – 2002*, MobiCom '02 (p. 160). New York, New York, USA: ACM Press. doi:10.1145/570645.570666
- SHIRKY, C., *Ontology is Overrated: Categories, Links and Tags*. Available at: www.shirky.com/writings/ontology_overrated.html Acesso em 10 jan, 2013
- SHNEIDERMAN, B. & PLAISANT, C. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, MA: Addison-Wesley Publishing Co. 2009, 5th Edition, pp. 672, ISBN 0-321-53735-1
- Synapse - Synapse for Kinect: <http://synapsekinect.tumblr.com/post/6610177302/synapse>
- VALKANOVA, N., JORDA, S., TOMITSCH, M., VANDE MOERE, A. Reveal-it!: the impact of a social visualization projection on public awareness and discourse. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*, 2013. ACM, New York, NY, USA, 3461-3470. DOI=10.1145/2470654.2466476 <http://doi.acm.org/10.1145/2470654.2466476>
- VOGEL, D., & BALAKRISHNAN, R. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In S. Feiner & J. A. Landay (Eds.). 2004, *Work* (Vol. 6, pp. 137-146). ACM. doi:10.1145/1029632.1029656
- W. W. W. Consortium. Extensible Markup Language Language 1.0 specification. W3C Recommendation, retrieved on October 6, 2000 from <http://www.w3.org/TR/2000/REC-xml-20001006>, October 2000.
- _____. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). 2007b. Disponível em: <<http://www.w3.org/TR/soap12-part1/>>. Acesso em: 26 mar. 2013.
- _____. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. 2007c. Disponível em: <<http://www.w3.org/TR/wsdl20/>>. Acesso em: 26 mar. 2013.

Apêndice A

API DE COMUNICAÇÃO

A API de comunicação da arquitetura é dividida em duas partes, sendo uma a API usada pelo ponto de apresentação e outra usada para os dispositivos móveis que interagem com o ponto de apresentação.

Na lista a seguir são apresentados as funções desenvolvidas para a API do ponto de apresentação para comunicação com o servidor de informação e proxy websocket (para envio de mensagens à dispositivos móveis). Todas as mensagens estão estruturadas no formato JSON, que é compacto e amplamente usado para transmissão em redes sem fio.

- `carrega_monitor(monitor_id)`
 - Descrição: Obtém a lista de serviços a serem executados
 - Destino: servidor de informação
 - Entrada: Identificador do ponto de apresentação
 - Saída: Lista de serviços que devem ser executados
- `carrega_servico(servico_id)`
 - Descrição: Executa cada serviço solicitado
 - Destino: servidor de informação
 - Entrada: Identificador do serviço
 - Saída: Informa os dados necessários sobre o serviço, para que o ponto de apresentação possa executá-lo
- `envia_dados(servico_id, dados)`
 - Descrição: O serviço envia dados ao servidor de informação ou para o dispositivo móvel
 - Destino: servidor de informação

- Entrada: identificador do serviço e conjunto de dados (Objeto JSON)
- Saída: Status do processamento função, verdadeiro se processou corretamente ou falso caso tenha ocorrido alguma falha
- recebe_dados()
 - Descrição: Recebe informação do servidor de informação, podendo ter diferentes serviços como destino.
 - Destino: ponto de apresentação (origem: servidor de informação ou dispositivo móvel)
 - Entrada: Nenhum
 - Saída: Conjunto de informações, como identificação do serviço de destino e conjunto de dados.
- get_info_device(serviço_id, dispositivo_id)
 - Descrição: Recebe informações do dispositivo móvel
 - Destino: servidor de informação
 - Entrada: Identificador do serviço e do dispositivo móvel
 - Saída: Conjunto de informações do dispositivo móvel e dados do usuário
- get_usuario (usuario_id)
 - Descrição: Obtém informações do usuário como nome e informações de contexto e serviços de interesse
 - Destino: servidor de informação
 - Entrada: Identificador do usuário
 - Saída: Conjunto de dados contendo
- conecta(url)
 - Descrição: Faz conexão com o Interaction Point via websocket
 - Destino: proxy websocket
 - Entrada: URL do proxy para conexão
 - Saída: Status da conexão sendo verdadeiro se ocorreu com sucesso ou falso se houve uma falha
- Eventos: entrada_dispositivo (dispositivo_id), saida_dispositivo (dispositivo_id)

- Descrição: Controle de eventos de chegada e saída de dispositivos móveis
- Destino: proxy websocket
- Entrada: Identificação do dispositivo móvel
- Saída: Status do processo realizado pelos serviços associados ao dispositivo, sendo verdadeiro se ocorreu com sucesso ou falso se houve uma falha.

A lista abaixo se refere à API do dispositivo móvel, que pode comunicar-se apenas com o ponto de apresentação para a solicitação de informação.

- conecta (URL)
 - Descrição: Faz conexão com o proxy para interação com o ponto de apresentação
 - Destino: proxy websocket
 - Entrada: URL do proxy para conexão
 - Saída: Status da conexão sendo verdadeiro se ocorreu com sucesso ou falso se houve uma falha
- carrega_servicos (dispositivo_id)
 - Descrição: Obtém a lista de serviços disponíveis ao ponto de apresentação
 - Destino: proxy websocket
 - Entrada: Identificador do dispositivo móvel
 - Saída: Lista de serviços a serem disponibilizados ao usuário
- carrega_interface (servico_id)
 - Descrição: Obtém a interface de interação para exibir na tela
 - Destino: proxy websocket
 - Entrada: Identificação do serviço
 - Saída: Dados necessários para o carregamento da interface de interação
- envia_dados(servico_id, dados)
 - Descrição: Envia dados de interação do usuário para o serviço no ponto de apresentação
 - Destino: proxy websocket

-
- Entrada: Identificação do serviço e conjunto de dados
 - Saída: Status do processamento função, verdadeiro se processou corretamente ou falso caso tenha ocorrido alguma falha
 - recebe_dados()
 - Descrição: Recebe dados de um serviço do ponto de apresentação para a interface de interação do usuário
 - Destino: dispositivo móvel (origem: ponto de apresentação)
 - Entrada: Nenhum
 - Saída: Conjunto de informações, como identificação do serviço de destino e conjunto de dados.