

# Android App Lifestyle – Smartphone e Smartwatch Integrados em uma Nuvem Computacional via Web Services

## *Android App Lifestyle – Smartphone and Smartwatch Integrated Into a Cloud Computing by Web Services*

Matheus Klein Schaeffer

Curso de Ciência da Computação, Universidade de Passo Fundo (UPF)  
Passo Fundo/RS, Brasil  
matheusklsch95@upf.com

Jeangrei Emanoelli Veiga, Daiana Biduski, Marcelo Trindade Rebonatto, Ana Carolina B. De Marchi  
Programa de Pós-Graduação em Computação Aplicada, Universidade de Passo Fundo (UPF)  
Passo Fundo/RS, Brasil  
{jeangrei.upf, daianabiduski}@gmail.com, {rebonatto, carolina}@upf.br

**Resumo** — A hipertensão é uma doença crônica caracterizada pelos níveis elevados de pressão arterial. A falta de aderência ao tratamento da hipertensão é uma preocupação crescente entre os profissionais da saúde. Para tanto, soluções m-Health estão cada vez mais em evidência, já que podem automatizar processos e facilitar a inclusão das condições de saúde do paciente de forma prática e em qualquer lugar, permitindo o acompanhamento do profissional da saúde. Este trabalho tem como objetivo relatar o desenvolvimento de uma solução m-Health para automatizar o processo de acompanhamento para pacientes hipertensos. A solução será composta por um aplicativo Android para Smartphone o App Lifestyle, integrado a um dispositivo vestível (Smartwatch) e ao Server Lifestyle em Nuvem Computacional (Cloud Computing), todos componentes do Sistema e-Lifestyle. Para o desenvolvimento do App Lifestyle foram utilizados Android Studio, um relógio Motorola Moto 360 e um Smartphone LG K10. Os resultados apontaram que os frameworks e as APIs utilizados permitiram abstrair diversos conceitos de conexão e de comunicação entre dispositivos diferentes, o que facilitou o desenvolvimento.

**Palavras Chave** – *e-Health; Mobilidade; Web Services; Computação Vestível.*

**Abstract** — Hypertension is a chronic disease characterized by high blood pressure levels. Lack of adherence to the treatment of hypertension is a growing concern among health professionals. For this, m-Health solutions are increasingly evident, since they can automate processes and facilitate the inclusion of the patient's health conditions in a practical way and in any place, allowing the health professional to follow up. This work aims to report the development of an m-Health solution to automate the follow-up process for hypertensive patients. The solution will be comprised of an Android application for Smartphone Lifestyle App, integrated into a Smartwatch and Server Lifestyle, all components of the e-Lifestyle System. For the development of the Lifestyle App were used Android Studio, a Motorola Moto 360 watch and a

**LG K10 Smartphone. The results showed that the frameworks and the APIs used allowed to abstract several concepts of connection and communication between different devices, which facilitated the development.**

**Keywords** – *e-Health; Mobility; Web Services; Wearable computing.*

### I. INTRODUÇÃO

As condições crônicas de saúde consistem em problemas que persistem por longos períodos de tempo e demandam tratamento contínuo [1]. De acordo com a World Health Organization [2], cerca de 50% dos pacientes com doenças crônicas em países desenvolvidos não se submetem a um tratamento de forma correta. A principal causa relatada por diversos pacientes é o esquecimento da tomada dos medicamentos, o que acaba contribuindo para o aumento nos gastos com serviços de saúde e a diminuição da qualidade de vida [3][4].

Além do uso regular de medicamentos, o estímulo ao tratamento contínuo por meio da mudança do estilo de vida é uma das estratégias para prevenção da hipertensão arterial, doença crônica que atinge 32,5% de indivíduos adultos no Brasil [5]. De acordo com [6], a aderência ao tratamento pode ser direta ou indireta. A aderência direta se dá, por exemplo, pela medição de fluídos corporais e observação direta do paciente. A aderência indireta ocorre pelo monitoramento da medicação, relatórios pessoais, uso de aparelhos eletrônicos de monitoramento, entre outros [4][6].

Dentre as estratégias adotadas para estimular a aderência ao tratamento da hipertensão arterial estão políticas públicas combinadas com ações nos meios de comunicação e, mais recentemente, o uso das Tecnologias da Informação e Comunicação (TICs) aplicadas à área da saúde, também chamado de soluções e-Health. O e-Health “Figura 1” pode ser definido como o uso das TICs no apoio à saúde e áreas

relacionadas, com o objetivo de expandir a cobertura e melhorar a eficácia do atendimento. Por sua vez, quando dedicados aos dispositivos móveis são denominados m-Health, caracterizando-se como uma especialidade do e-Health [7]. Dispositivos móveis trazem a vantagem da mobilidade, ou seja, permitem o uso em praticamente qualquer lugar, possuem tamanho pequeno além de diversas modalidades de interação [8].

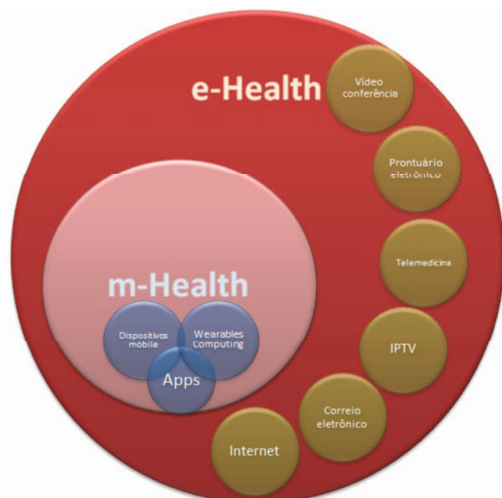


Figura 1. Esquema e-Health e m-Health. Adaptado de [9]

Muitas iniciativas de m-Health vêm sendo adotadas com este objetivo, como nos trabalhos de [4] e [10]. No entanto, a falta da disponibilização dos dados registrados sobre a condição de saúde do paciente para o profissional da saúde ainda é um problema nestes sistemas. O acesso a tais informações é de extrema importância já que possibilita ao profissional realizar o acompanhamento do tratamento e também a emissão de orientações mais precisas.

A implementação de sistemas que promovam esta mobilidade de informações é um desafio na computação, pois exige uma preocupação quanto à integração com outros sistemas e o uso de tecnologias inovadoras como dispositivos vestíveis (Computação vestível - Wearable Computing). Neste sentido, o uso de Web Services com Nuvem Computacional (Cloud Computing) se apresenta como uma solução para prover a integração com outros sistemas e plataformas. De acordo com a W3C [12], um Web Service é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes. O uso desta tecnologia possibilita que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. Uma Nuvem Computacional é um modelo de computação sob demanda composto por recursos de TI (hardware e/ou software) autônomos e interconectados [13]. Para que a integração de dispositivos vestíveis aconteça sem muito esforço, o ideal é a exploração de recursos nativos já existentes em dispositivos como o Smartwatch (relógio inteligente) para a visualização de mensagens e interação com o sistema.

Diante do contexto acima, este trabalho relata o desenvolvimento do aplicativo Android o App Lifestyle, para

Smartphone, e a extensão que integra o Smartwatch via comunicação Bluetooth, ambos partes integrantes do Sistema e-Lifestyle [14]. O Sistema e-Lifestyle é um sistema que tem como objetivo auxiliar na promoção de um estilo de vida saudável em pacientes hipertensos. A integração de diferentes tecnologias móveis e vestíveis ao sistema visa promover a facilidade no acesso e uso do sistema dentro do contexto do paciente, disponibilizando uma interface facilitada para a inserção de dados referentes à saúde do paciente e seu estilo de vida, como o exemplo do batimento cardíaco, pressão arterial, humor, peso, sono, etc; bem como o recebimento de notificações que podem ser exibidas tanto no Smartphone quanto no Smartwatch.

O trabalho se subdivide em cinco seções. Na seção 2 são apresentados alguns trabalhos relacionados e o diferencial deste trabalho. Na seção 3 serão descritos a arquitetura do sistema, as ferramentas utilizadas para o desenvolvimento, aspectos inerentes às interfaces e o funcionamento. Na seção 4 são apresentados os testes realizados, na seção 5 uma breve conclusão e na seção 6 agradecimentos.

## II. TRABALHOS RELACIONADOS

Há alguns trabalhos que apresentam soluções similares a que este propõe. O trabalho de [4] apresenta um aplicativo para Android e Smartwatch (no caso o Pebble, não possuindo compatibilidade com Android Wear) para manter alertas sobre o uso de medicamentos. Porém, o trabalho não possui a integração entre profissional e paciente.

O trabalho de [10] apresenta um sistema de alertas usando um sistema de mensagens de texto, possibilitando a conexão entre médico e paciente, porém não se trata de um aplicativo específico com comunicação própria, dificultando o processo.

Dentre os produtos comercialmente disponíveis, há o Tactio Health [15], aplicativo para gerenciamento de estilo de vida e o Glucosio [16], desenvolvido para o monitoramento de diabetes. O Tactio Health não fornece abertamente suporte à conexão entre paciente e médico e o Glucosio não oferece esse serviço.

## III. APP LIFESTYLE

O App Lifestyle é um aplicativo m-Health desenvolvido para uso de pacientes hipertensos, tendo como principal finalidade disponibilizar um acesso fácil ao Server Lifestyle via Web Services utilizando um Smartphone e Smartwatch. O desenvolvimento do App Lifestyle integrado ao Smartwatch via Bluetooth foi objeto de estudo deste trabalho.

### A. Arquitetura

O aplicativo App Lifestyle “Figura 2.c”, integrado ao Smartwatch “Figura 2.d” são parte integrante do Sistema e-Lifestyle apresentado na Figura 2. O Server Lifestyle “Figura 2.a” é um servidor disponibilizado em Nuvem Computacional, que possui um Sistema Gerenciador de Banco de Dados PostgreSQL, um servidor de aplicação Web Glassfish, um Web Services REST e um conjunto de formulários Web, definido como App Web “Figura 2.b”. A comunicação entre o App Lifestyle, instalado no Smartphone, e o Server Lifestyle ocorre a partir de uma conexão ativa com a internet “Figura 2.e” por Web Services REST “Figura 2.f” e a comunicação

entre o App Lifestyle e Smartwatch funciona por Bluetooth “Figura 2.g” .



Figura 2. Arquitetura do Sistema e-Lifestyle. Adaptado de [14].

O Server Lifestyle é responsável principalmente por manter e processar os dados gerados pelo sistema, além de prover uma interface de integração via Web Services e disponibilizar formulários Web para o profissional da saúde pode manter informações de seus pacientes, e o administrador do sistema gerenciar o mesmo.

O aplicativo App Lifestyle foi desenvolvido para ser executado em dispositivos móveis que utilizam o sistema operacional Android, como exemplo do Smartphone. O mesmo tem a responsabilidade de prover ao paciente uma forma de acesso ao Server Lifestyle para manter seus dados (ações sobre seu peso, pressão arterial, batimento cardíaco, sono, entre outras); enviar e receber mensagens de alerta sobre picos hipertensivos; e comunicar-se com o Smartwatch. A comunicação com Server Lifestyle “Figura 2.f” é realizada utilizando REST (*Representational State Transfer*) no formato JSON, um estilo de arquitetura para Web Services que usa métodos HTTP (GET, POST, PUT, DELETE) para conectar as máquinas [16]. Para implementar recursos REST no Android, foi utilizado o framework Retrofit, desenvolvido pela Square. Seu funcionamento ocorre da seguinte forma: a partir de uma URL usada como base (domínio do servidor e o Web service) e o “endpoint” (que contém os recursos a serem acessados) monta o método HTTP e faz a requisição ao Server Lifestyle, que processa e persiste a informação na base de dados.

Com o objetivo de promover o uso de dispositivos vestíveis [18], foi integrado um Smartwatch “Figura 2.f”, o qual possui como principal funcionalidade a interação com o sistema Lifestyle para o recebimento de mensagens, confirmar ou cancelar tarefas previamente programadas. Esse processo ocorre a partir das mensagens trocadas com o App Lifestyle, mantendo uma interface visual simples e funcional. A troca de mensagens é realizada a partir da comunicação Bluetooth, usando a biblioteca android.support.v4.

## B. Ferramentas de Desenvolvimento

Para o desenvolvimento do App Lifestyle foi utilizado a IDE (*Integrated Development Environment*) Android Studio, um relógio Motorola Moto 360, um smartphone LG K10. A versão do Android para o qual o aplicativo foi construído é a 4.3 – versão mais antiga compatível com o Android Wear, portanto requisito mínimo para o uso do aplicativo, sendo compatível com 76,9% dos dispositivos Android disponíveis no mercado [19].

Para a implementação das conexões cliente-servidor usando REST em Android foi utilizado o framework Retrofit (Square Inc.), devido a sua simplicidade de implementação. Os dados são enviados do Smartphone para o servidor Server Lifestyle e vice-versa no formato JSON, um formato utilizado para a transferência de dados entre computadores, fácil de ser lido e interpretado [20].

Para armazenar as informações dos usuários o aplicativo utiliza um recurso nativo do Android, chamado de Internal Storage. Esse recurso armazena um arquivo no diretório do aplicativo, podendo ser acessado apenas pelo próprio aplicativo. Para o projeto, os dados são armazenados como objeto no formato JSON.

A interface do App Lifestyle segue o padrão conhecido como “Google Material Design” elaborado pela Google [21], que inclui uma paleta de cores, guias de estilo para ícones, navegação, entre outros. O material foi criado para oferecer um estilo visual que pudesse ser usado por todos os serviços do Google, tendo como objetivo conveniência e facilidade de percepção.

## C. Funcionamento do App Lifestyle

Para utilizar o App Lifestyle, o paciente deve possuir um Smartphone ou Tablet com sistema operacional Android versão mínima 4.3. Para a instalação no dispositivo móvel é preciso ter acesso ao arquivo apk, e seguir o processo padrão de instalação de Apps Android. Após a instalação, a primeira interface apresentada é a tela de login, onde o paciente deve informar seu email e senha cadastrados. Ao acessar o aplicativo pela primeira vez, o paciente deverá realizar seu auto cadastro, clicando no link “criar conta” “Fig. 3”.

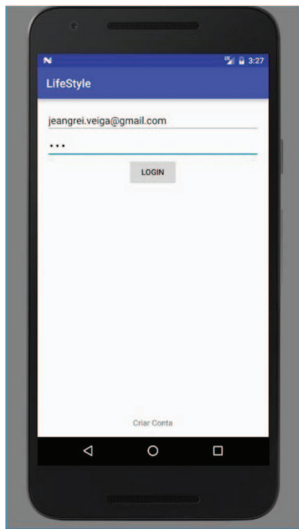


Figura 3. Interface de acesso

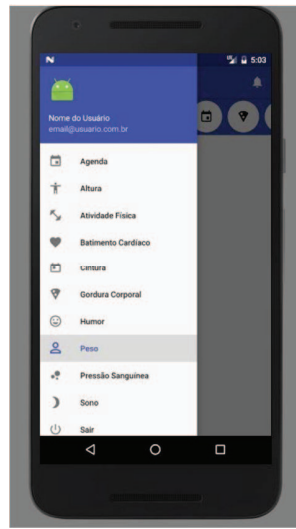


Figura 4. Interface Menu Principal

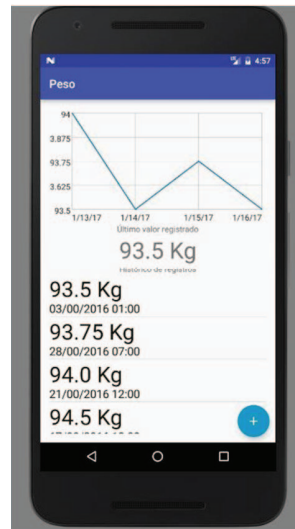


Figura 5. Interface com histórico de pesos registrados



Figura 6. Interface para inclusão da pressão arterial

Após realizar a autenticação com sucesso, o paciente tem acesso ao menu principal “Fig. 4”, onde consta sua identificação e uma lista de funcionalidades utilizadas para a manutenção do seu estilo de vida, sendo elas: altura, atividade física, batimento cardíaco, cintura, gordura corporal, humor, peso, pressão sanguínea e sono. Além disso, é possível interagir com uma agenda de lembretes, onde são programadas tarefas com alertas para tomar medicamentos, para realização de atividade física, entre outras, e visualizar alertas e mensagens trocadas com profissional da saúde.

Após a seleção de uma funcionalidade, o paciente visualiza o histórico de registros “Fig. 5”, podendo executar as operações de incluir, alterar ou excluir informações. Para o processo de alteração ou exclusão, o paciente precisa selecionar um registro do histórico por um segundo, assim, uma janela do tipo pop-up é exibida contendo essas opções. Antes da finalização do processo de exclusão, um novo pop-up é exibido solicitando a confirmação da operação.

O histórico dos registros também permite que o paciente inclua um novo registro ao clicar no botão contendo o sinal de adição “+”. A Fig. 6 apresenta a interface de cadastro da funcionalidade referente à pressão arterial. Para inserir um novo registro, o paciente precisa informar a data e hora da aferição, o valor e, se necessário, alguma observação. Ao final da operação, basta o paciente clicar no botão salvar.

Todas as operações de manutenção dos registros realizados no App Lifestyle são salvas no App e sincronizadas posteriormente com Server Lifestyle. Tal procedimento garante que os dados sejam persistidos na base de dados em Nuvem Computacional e estarão disponíveis para o profissional da saúde.

Após a sincronização com Server Lifestyle, as informações já podem ser visualizadas e/ou editadas pelo profissional. Com base nas informações adicionadas pelo paciente referente ao seu estilo de vida e sua saúde, profissional poderá avaliar e verificar situações como: em quais momentos a ação farmacológica está sendo mais eficiente, se o paciente está seguindo suas orientações para adoção de um estilo de vida mais saudável, entre outras.

A utilização do dispositivo Smartwatch não ocorre por meio de um aplicativo Android Wear tradicionalmente instalado no dispositivo, mas sim pela extensão das notificações Android geradas pelo App Lifestyle.

Este recurso é um Service [22] de notificação do Android que consta na versão 4.3 do Android, o qual está presente na biblioteca *NotificationCompat.WearableExtender* [23] e chamamos de *Service Wearable Notification* (SWN). O SWN ao perceber alguma notificação gerada pelo App Lifestyle, dispara um objeto chamado *Intent* e a notificação é exibida no dispositivo Wearable, no caso, o Smartwatch. O uso deste recurso facilitou a integração do Smartwatch sem a necessidade do esforço de implementação de outro aplicativo.

A funcionalidade de exibir notificações é apresentada tanto no dispositivo Smartphone quanto no dispositivo Smartwatch, desde que os mesmos estejam pareados via bluetooth. O Smartwatch apresenta a notificação utilizando uma interface pré-definida contendo um título e uma breve descrição “Fig. 7”, com uma opção positiva “Fig. 8” e uma opção negativa, exibidas quando o usuário desliza a interface da direita para a esquerda. Caso o paciente selecione a opção “Sim”, essa informação é atualizada no App Lifestyle e posteriormente sincronizada com Server Lifestyle.

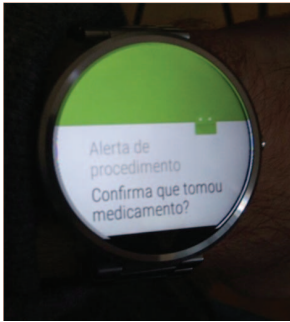


Figura 7. Tela principal de confirmação

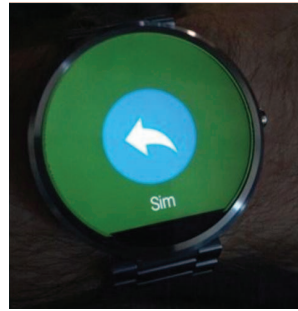


Figura 8. Tela com a opção positiva ("sim")

#### IV. TESTES DURANTE O DESENVOLVIMENTO

Foram realizados testes com o objetivo de verificar se a comunicação entre cliente e servidor foi realizada de forma eficaz, e para verificar se os frameworks utilizados cumpriram com o papel de minimizar o trabalho despendido neste processo.

Os testes utilizaram o navegador Mozilla Firefox e a ferramenta para teste de Web Services SoapUI. Para testar os métodos GET foram criados, através do SoapUI, diversos objetos com atributos aleatórios. Os demais métodos foram testados utilizando-se o aplicativo e as ferramentas SoapUI e Firefox, para verificar se os dados inseridos ou removidos eram respectivamente registrados ou apagados no banco de dados.

Para testes no aplicativo utilizou-se o recurso de programação Log, que exibe no terminal de texto do Android Studio informações em tempo de execução. No método GET, tal recurso foi utilizado para exibir se o vetor contendo as informações do usuário estava vazio ou não. A fim de verificar visualmente a origem de possíveis erros do sistema e também a forma em que os dados estavam sendo enviados, foram testados os métodos POST e PUT, a partir da exibição no App Lifestyle de uma caixa de diálogo com o conteúdo do JSON enviado ao servidor.

Foram realizados testes para verificar o tempo de resposta do servidor em redes WiFi, 2G e 3G. Nos testes realizados via WiFi, houve um atraso de 340 milissegundos nos métodos GET e DELETE e 400 milissegundos nos métodos POST e PUT. Nos testes realizados via 2G, o tempo para os métodos GET e DELETE foram de 1 segundo e 800 milissegundos e, para POST e PUT foram 3 segundos e 500 milissegundos. No 3G houve um atraso de 500 milissegundos (GET e DELETE) e 1 segundo (POST e PUT). Independente das condições de rede, se a resposta do servidor tiver atraso superior a 10 segundos, o Retrofit interrompe a conexão e exibe uma mensagem de erro.

Os testes realizados demonstraram que o framework utilizado é necessário para o desenvolvimento no âmbito do Android, devido à simplicidade da implementação. Como a conexão entre servidor e cliente é realizada com métodos

HTTP e toda implementação que manipula dados é no lado do servidor, a implementação do cliente é relativamente simples.

No Smartwatch foi possível verificar quais são os métodos para o envio e o recebimento de mensagens entre o paciente e o profissional da saúde. Foram testadas diversas demonstrações usando o sistema de notificações padrão do Android, com o uso de modelos fornecidos pela API. Conforme o Google Developers, as notificações podem ser transmitidas do telefone ao relógio tanto pela sincronização nativa de notificações como também pela API, denominada DataLayer.

#### V. CONCLUSÕES

Os frameworks e as APIs utilizadas para desenvolver o App Lifestyle permitiram abstrair diversos conceitos de conexão e comunicação entre dispositivos diferentes. Tais recursos facilitam o desenvolvimento de ferramentas que necessitam de comunicação com outros sistemas e dispositivos, como Server Lifestyle e o Smartwatch.

Durante o início do desenvolvimento do App Lifestyle, ocorrido dentro da estrutura computacional da Universidade, foram encontradas algumas dificuldades quanto ao uso da porta 8080 (usada pelo Server Lifestyle para receber requisições HTTP [24]). Isso ocorreu devido à política de segurança da informação da Universidade. A porta 8080 era bloqueada pelo firewall e o problema foi resolvido com a implantação do Server Lifestyle em Nuvem Computacional.

Com o uso do SWN durante o desenvolvimento, foi possível integrar com pouco esforço um Smartwatch ao projeto, porém, sem a utilização de outros recursos interessantes como o GPS, acelerômetro, entre outros.

O aplicativo Android App Lifestyle integrado ao dispositivo vestível Smartwatch busca proporcionar a integração entre o paciente e o profissional da saúde de forma contínua. Além disso, a integração de diferentes tecnologias permite a mobilidade dentro do contexto do paciente, motivando-o a seguir as recomendações de forma adequada e de acordo com o tratamento.

Este trabalho espera contribuir para as áreas da computação e da medicina do estilo de vida. Como trabalhos futuros, faz-se necessário a realização de novos testes a partir da integração dos aplicativos ora desenvolvidos juntamente com o Server Lifestyle. Deste modo, será possível validar a solução como um todo, verificando se ela realmente pode ocasionar em mudanças significativas na promoção de um estilo de vida saudável.

#### VI. AGRADECIMENTOS

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ) por seu apoio.

#### REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Moreira, M. C. N. et al. Doenças crônicas em crianças e adolescentes: uma revisão bibliográfica. *Ciência & Saúde Coletiva*, v. 19, n.7, 2014, ISSN 1413-8123.
- [2] SABATEL, E. "Adherence to long-term therapies evidence for action", Geneva: World Health Organization, 2003. Disponível em: <<http://apps.who.int/medicinedocs/en/d/Js4883e/>> Acesso em 24 de setembro de 2016.
- [3] VERVLOET, M. et al. The effectiveness of interventions using

- electronic reminders to improve adherence to chronic medication: a systematic review of the literature. *Journal of the American Medical Informatics Association*, v. 19, n. 5, p. 200–209, 2012.
- [4] MAGLOGIANNIS, I.; SPYROGLOU, G.; PANAGOPOULOS, C. Mobile Reminder System for Furthering Patient Adherence Utilizing Commodity Smartwatch and Android devices. p. 10–13, 2014.
- [5] MALACHIAS M. V. B. 7a Diretriz Brasileira De Hipertensão Arterial. *Arq Bras Cardiol*, v. 107, n. 3, 2016.
- [6] FARMER, K. C. Methods for measuring and monitoring medication regimen adherence in clinical trials and clinical practice. *Clinical therapeutics*, v. 21, n. 6, p. 1074–90; discussion 1073, 1999.
- [7] WORLD HEALTH ORGANIZATION. mHealth: New horizons for health through mobile technologies. *Observatory*, v. 3, n. June, p. 112, 2011.
- [8] DE SÁ, M. Designing and Evaluating Mobile Interaction: Challenges and Trends. *Foundations and Trends® in Human–Computer Interaction*, v. 4, n. 3, p. 175–243, 2010.
- [9] Palacios, E. R. eHealth, mHealth, Wearables, apps... Help! Instituto Aragonés de Ciencias de la Salud, 2015. Disponível em <<http://www.blog.iacsaragon.es/2015/07/ehealth-mhealth-wearables-apps-help.html>>. Acesso em 13 de outubro de 2015.
- [10] CHEN, Z.-W. et al. Comparison of an SMS text messaging and phone reminder to improve attendance at a health promotion center: a randomized controlled trial. *Journal of Zhejiang University*, v. 9, n. 1, p. 34–8, 2008.
- [11] CALLIGARO, Mike. “Why Persistent Storage Is A Good Thing”, Windows Mobile Team Blog, 2005. Disponível em: <<https://blogs.msdn.microsoft.com/windowsmobile/2005/07/14/why-persistent-storage-is-a-good-thing/>>. Acesso em 31 de outubro de 2016.
- [12] HAAS, Hugo; BROWN, Allen. Web Services Glossary. WTC 2004. Disponível em: <<https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice>>. Acesso em 31 de outubro de 2016.
- [13] HASSAN, Q. F. Demystifying Cloud Security. *CrossTalk*, p. 16–21, 2011.
- [14] VEIGA, J. E. Sistema e-Lifestyle: uma solução e-Health para Auxiliar na Promoção de Estilo de Vida Saudável em Hipertensos. Projeto de Dissertação (Mestrado em Computação Aplicada) – Instituto de Ciências Exatas e Geociências da Universidade de Passo Fundo, Passo Fundo, 2015.
- [15] Tactio Health Group. Tactio Health, 2016. Disponível em: <<http://www.tactiohealth.com>>. Acesso em 6 de dezembro de 2016.
- [16] Glucosio Open Source Project. Glucosio, 2016. Disponível em: <<https://github.com/glucosio>>. Acesso em 6 de dezembro de 2016.
- [17] Fielding, Roy Thomas. *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine, 2000.
- [18] SIEWIOREK, D.P.; SMILAGIC, A.; STARNER, T., “Application Design for Wearable Computing”. Series Editor: Mahadev Satyanarayanan, Carnegie Mellon University, 2008.
- [19] Google. Dashboards. Google Android Developers, 2016. Disponível em: <<https://developer.android.com/about/dashboards/index.html>>. Acesso em 11 de novembro de 2016.
- [20] Json. Disponível em: <<http://www.json.org/>>. Acesso em 11 de novembro de 2016.
- [21] Google. Google Material Design. Disponível em: <<https://material.google.com/>>. Acesso em 9 de novembro de 2016.
- [22] Google. Services, Android Developers. Disponível em: <<https://developer.android.com/guide/components/services.html>>. Acesso em 13 de novembro de 2016.
- [23] Google. Adding Wearable Features to Notifications, Android Developers. Disponível em: <<https://developer.android.com/training/wearables/notifications/index.html>>. Acesso em 12 de novembro de 2016.
- [24] Internet Assigned Numbers Authority. Service Name and Transport Protocol Port Number Registry, 2016. Disponível em: <<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt>> Acesso em 16 de novembro de 2016.