UnB - UNIVERSIDADE DE BRASÍLIA

FGA - FACULDADE UNB GAMA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA BIOMÉDICA

OPEN GAIT ANALYTICS - IMPLEMENTANDO UM SOFTWARE COMO SERVIÇO PARA ANÁLISE E SIMULAÇÃO DE MARCHA HUMANA

Roberto Aguiar Lima

ORIENTADOR(A): Dra. Lourdes Mattos Brasil
COORIENTADORA(A): Dra. VERA REGINA DA SILVA MARÃES

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA BIOMÉDICA

PUBLICAÇÃO: NUMERAÇÃO / 2015

BRASÍLIA/DF : Setembro– 2015

UnB - UNIVERSIDADE DE BRASÍLIA

FGA - FACULDADE UNB GAMA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA BIOMÉDICA

OPEN GAIT ANALYTICS - IMPLEMENTANDO UM SOFTWARE COMO SERVIÇO PARA ANÁLISE E SIMULAÇÃO DE MARCHA HUMANA

Roberto Aguiar Lima

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA BIOMÉDICA DA FACULDADE GAMA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA BIOMÉDICA.

APROVADO POR:

Prof. Dra. Lourdes Mattos Brasil
(Orientador(a))

Prof. Dra. VERA REGINA DA SILVA MARÃES
(Co-Orientador(a))

Prof. Dr(a).
(Examinador Externo)

BRASÍLIA/DF , 01 DE SetembroDE 2015

FICHA CATALOGRÁFICA

Roberto Aguiar Lima
OPEN GAIT ANALYTICS - IMPLEMENTANDO UM SOFTWARE COMO SERVIÇO PARA ANÁLISE E SIMULAÇÃO DE MARCHA HUMANA, [Distrito Federal] 2015.
NUMERAÇÃO . 28 p., 210 x 297 mm (FGA/UnB Gama, Mestre, Engenharia Biomédica, 2015). Dissertação de Mestrado - Universidade de Brasília. Faculdade Gama. Programa de Pós- Graduação em Engenharia Biomédica.
1. análise de marchar. 2. aprendizado de máquina
3. joelho. 4. simulação
I. FGA UnB Gama/ UnB. II.*OPEN GAIT ANALYTICS* - IMPLEMENTANDO UM SOFTWARE COMO SERVIÇO PARA ANÁLISE E SIMULAÇÃO DE MARCHA HUMANA
CDU: Nº da CDU (biblioteca)

REFERÊNCIA BIBLIOGRÁFICA

LIMA, R. A. (ANO). TÍTULO. Dissertação de Mestrado em Engenharia Biomédica, Publicação NO./ANO, Programa de Pós-Graduação em Engenharia Biomédica, Faculdade Gama, Universidade de Brasília, Brasília, DF, 28 p.

CESSÃO DE DIREITOS

AUTOR: Roberto Aguiar Lima
TÍTULO: *OPEN GAIT ANALYTICS* - IMPLEMENTANDO UM SOFTWARE COMO SERVIÇO PARA ANÁLISE E SIMULAÇÃO DE MARCHA HUMANA

GRAU: Mestre
ANO: 2015

2015.
ENDEREÇO.
CEP: ..., Brasília, DF – Brasil

DEDICATÓRIA

Para ..., com amor.

AGRADECIMENTOS

...

O Mestre na arte da vida faz pouca distinção entre o seu trabalho e o seu lazer, entre sua mente e seu corpo, entre sua educação e sua recreação. Ele simplesmente persegue sua visão de excelência em tudo o que faz, deixando para os outros a decisão de saber se está trabalhando ou se divertindo. Ele acha que está sempre fazendo as duas coisas simultaneamente.

Texto Budista

RESUMO

OPEN GAIT ANALYTICS - IMPLEMENTANDO UM SOFTWARE COMO SERVIÇO PARA ANÁLISE E SIMULAÇÃO DE MARCHA HUMANA

Autor: Roberto Aguiar Lima
Orientador(a): Prof(a). Dra. Lourdes Mattos Brasil
Coorientador(a): Dra. VERA REGINA DA SILVA MARÃES
Programa de Pós-Graduação em Engenharia Biomédica – Qualificação de Mestrado
BRASÍLIA/DF 2015

Texto corrido sem parágrafo. 1 página.

Palavras-chaves: análise de marchar, aprendizado de máquina, joelho, simulação.

ABSTRACT

OPEN GAIT ANALYTICS - IMPLEMENTING A SOFTWARE AS A SERVICE FOR HUMAN GAIT ANALYSIS AND SIMULATION

Author: Roberto Aguiar Lima
Supervisor: Prof(a). Dra. Lourdes Mattos Brasil
Co-supervisor: Dra. VERA REGINA DA SILVA MARÃES
Post-Graduation Program in Biomedical Engineering – Qualify of Master Degree Brasília, Month of Year.

Texto corrido sem parágrafo. 1 página.

Key-words: gait analysis, machine learning, knee, simulation.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Program Interface*
CSS	*Cascade Style Sheet*
FCE	Faculdade Ceilândia
FGA	Faculdade Gama
HTML5	*HiperText Markup Language*
GNU	GNU is Not Unix
GUGT	*Get Up and Go Test*
LIS	Laboratório de Informática em Saúde
LPH	Laboratório de Performance Humana
MVC	*Model View Controller*
UnB	Universidade de Brasília
VM	*Virtual Machine*

LISTA DE SÍMBOLOS

Símbolos Latinos

F_1, F_2	Força (N)

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO E FORMULAÇÃO DO PROBLEMA

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O presente trabalho visa iniciar um projeto de desenvolvimento de software como serviço, para análise e simulação de marchar humana.

1.2.2 Objetivos Específicos

Os objetivos específicos são:

- Definir um processo de desenvolvimento ágil adequado ao projeto;
- Explicitar uma visão arquitetural inicial do software;
- Escolher componentes de software a serem usados na solução;
- Criar um *backlog* inicial de histórias de usuários;
- Selecionar um conjunto mínimo de histórias de usuários, suficientes para uma *release* funcional do software;
- Implementar e testar estas histórias de usuários.

1.3 REVISÃO DA LITERATURA

Foram usados os seguinte serviços web para o levantamento bibliográfico deste trabalho:

- IEEE Xplore Digital Library (http://ieeexplore.ieee.org);
- PubMed (http://www.ncbi.nlm.nih.gov/pubmed);
- Portal de Periódicos CAPES/MEC (http://periodicos.capes.gov.br).

Foram utilizadas as seguintes chaves de pesquisa em cada um dos serviços acima: "*Gait Analysis Software*".

Os artigos considerados mais relevantes para o trabalho foram escolhidos, levando-se em consideração, entre outras coisas, a descrição de características interessantes a serem implementadas no software a ser desenvolvido.

Quando o assunto se trata de análise de marcha, a obra mais aclamada, inclusive citada em muitas das referências abaixo, é (PERRY; BURNFIELD, 2010). Como sugerido por (MALAS, 2010), esta é uma obra obrigatória a qualquer um que deseje estudar análise de marcha.

Em (VIEIRA et al., 2015) um sistema de análise e classificação de marcha é proposto, como alternativa a soluções de mercado mais caras. A proposta inicial é coletar dados a partir de marcadores posicionados no corpo do paciente, através de câmeras de vídeo, classificando padrões de marcha com aprendizado de máquina.

Em (DUHAMEL et al., 2004) é apresentada uma ferramenta para melhorar a confiabilidade de curvas para um paciente, classificar pacientes em determinadas populações e comparar populações. Trata-se de uma ferramenta estatística para análise de marchar.

Detecções de eventos do ciclo de marcha, são características interessantes para um software de análise de marcha. Em (GHOUSSAYNI et al., 2004) são documentados métodos para detecção de 4 eventos: contato do calcanhar, elevação do calcanhar, contato do dedão do pé e elevação do dedão do pé.

Uma comparação entre dois pacotes distintos para análise de marchar foi realizada em (MORAES; SILVA; BATTISTELA, 2003). Neste trabalho dados captados por câmeras e plataformas de força são coletados e passados aos pacotes de software Kin Trak e Ortho Trak.

Uma amostra de como um software pode ser utilizado para gerar bases de dados de análise de marcha, é visto em (MORENO et al., 2009). Neste artigo os autores capturam dados de crianças sadias, afim de obterem padrões para serem utilizados em sistemas de análise de movimentos.

Um sistema de aquisição e análise de marcha, foi desenvolvido e demostrado em (FERREIRA; CRISOSTOMO; COIMBRA, 2009). Neste trabalho, o hardware para captura de dados, e o software para análise dos dados, foram desenvolvidos num único projeto. Com os resultados gerados pelas análise feitas por este projeto, foi possível construir um robô bípede, que apresentou resultados satisfatórios caminhando num ciclo de marchar confortável.

A partir da análise de marcha, é possível criar métodos para se estabelecer o grau de desvio do ciclo de marcha, que um paciente pode apresentar. Em (BEYNON et al., 2010) é apresentado o método *Gait Profile Score*. O método em si é um bom candidato a funcionalidade em um software de análise de marcha, pois serviria de auxílio clínico ao profissional da área de saúde. Uma outra funcionalidade inspirada no campo clínico é mostrado em (CIPPITELLI et al., 2015). Neste trabalho os autores propõem a automatização do método *Get Up and Go Test*(GUGT), que é usualmente utilizado em análise de marcha no campo da reabilitação.

1.4 ORGANIZAÇÃO DO TRABALHO

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ANÁLISE DE MARCHA

2.2 MÉTODOS ÁGEIS

A muito tempo vários métodos para desenvolvimento de software são propostos. Em 2001, um grupo de pessoas muito experientes em densenvolvimento de software, juntaram-se em Salt Lake City,Utha, para resolverem problemas de desenvolvimento de software (GREENE; STELLMAN, 2014). Como resultado deste encontro, foi criado o manifesto ágil, que é reproduzido a seguir (BECK et al., 2001):

> Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:
>
> Indivíduos e interações mais que processos e ferramentas;
>
> Software em funcionamento mais que documentação abrangente;
>
> Colaboração com o cliente mais que negociação de contratos;
>
> Responder a mudanças mais que seguir um plano.
>
> Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.

Meses após a criação do manifesto, estas pessoas também criaram os princípios ágeis e a Aliança Ágil (LAYTON, 2012).

Os princípios ágeis são um conjunto de 12 itens com o objetivo de auxiliar na implantação de metodologias ágeis. Eles são reproduzidos a seguir a título de ilustração (BECK et al., 2001):

> 1. Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.
>
> 2. Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.
>
> 3. Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.
>
> 4. Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.
>
> 5. Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.
>
> 6. O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face.
>
> 7. Software funcionando é a medida primária de progresso.

> 8. Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
>
> 9. Contínua atenção à excelência técnica e bom design aumenta a agilidade.
>
> 10. Simplicidade–a arte de maximizar a quantidade de trabalho não realizado–é essencial.
>
> 11. As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis.
>
> 12. Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

Este manifesto, serviu de marco agregador de métodos e técnicas, que já existiam a época, mas não eram amplamente difundidas como *Scrum*, *Extreme Programming*, *kanban*, *lean*, entre outros. Estas técnicas, apesar de anteriores ao manifesto, possuem em seus cernes, muito em comum com os valores ágeis. A figura 1 tenta ilustrar esta idéia.

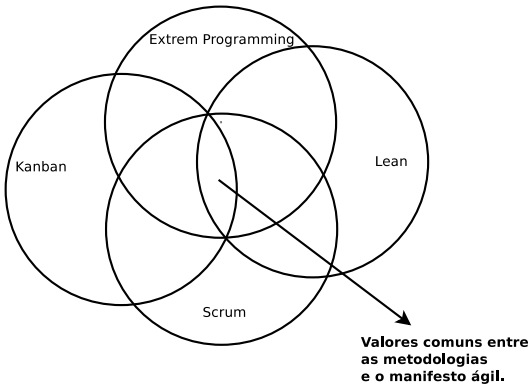

Figura 1 – Valores em comum. Adaptado de (GREENE; STELLMAN, 2014).

A adoção dos métodos ágeis hoje é praticamente hunanimidade. Isto se deve aos modelos de desenvolvimento adotados até a década de 1990. Esses modelos eram na sua grande maioria baseados no modelo *waterfall*, que imitava uma linha de produção onde o software era desenvolvido em fases. A saída de cada fase era a entrada da próxima. A figura 2, mostra um exemplo de processo baseado no modelo *waterfall*.

O maior problema do modelo *waterfall*, era que este era completamente averso a mudanças de requisitos. Hoje é notório que a grande maioria dos softwares necessitam ser bastante recepitivos a mudanças.

(Kristin Runyan; ASHMORE, 2014) compara o modelo *waterfall* como o modelo ágil.

Como evolução do modelo *waterfall*, surgiram os processos baseados em modelos iterativos. A diferença agora, é que o processo de desenvolvimento passa a ser baseado em ciclos. Cada ciclo passa por cada uma das fases do *waterfall*. Este tipo de processo,

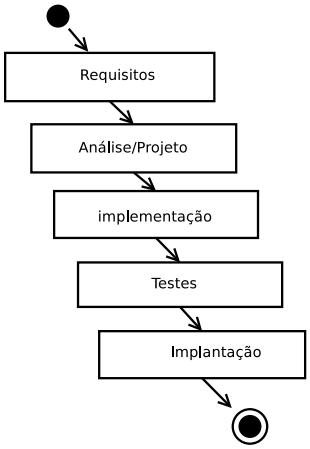

Figura 2 – Exemplo de processo baseado no modelo *waterfall*

Tabela 1 – Comparando Modelos de Desenvolvimento de Software

Watefall	*Ágil*
Prescritivo	Abstrato
Documentação extensiva	Mínimo de documentação
Sequencial	Contínuo
Formal	Informal
Focu no processo	Focu na comunicação
Mudança gradual	Mundança rápida

Fonte: (Kristin Runyan; ASHMORE, 2014)

apresentava melhoras, mas ainda era concebido sob a forma de um processo que visava resolver problemas determinísticos, como o das linhas de produção das fábricas. Mas, como descrito em (BECK, 2004), o processo de se construir software é mais parecido com o ato de dirigir. O motorista sabe o destino a que quer chegar, porém, durante o percurso pode haver um acidente e tem-se que mudar um pouco a rota, ou alguém está passando por uma faixa de pedestre e necessita-se parar, ou ainda um sinal de transito pode ficar vermelho. Esta é a principal motivação para que este trabalho, privilegie métodos ágeis de desenvolvimento.

2.2.1 Scrum

2.3 SOFTWARE COMO SERVIÇO

2.4 SOFTWARE LIVRE

2.5 APRENDIZADO DE MÁQUINA

2.6 CMAC

3 METODOLOGIA

3.1 AMBIENTE DE ESTUDO

Como este trata de um projeto de desenvolvimento de software na área de análise de marcha, dois ambientes de trabalho distintos foram amplamente utilizados. São estes:

1. Laboratório de Informática em Saúde (LIS) na Faculdade Gama (FGA) da Universidade de Brasília (UnB);
2. Laboratório de Performance Humana (LPH) na Faculdade Ceilândia (FCE) da UnB.

No LIS foram desempenhadas as tarefas relativas a engenharia de software e disponibilização do software. Foram utilizadas estações de trabalho do tipo *Power Mac* com sistema operacional *MAC OS X* 10.10.3, sendo que uma foi preparada para funcionar como servidor de aplicação e roteador de rede. A estação preparada, serve de hospedeira de duas máquinas virtuais (*Virtual Machines* - VMs) rodando através do software *Virtualbox* 4.3. Cada VM utiliza sistema operacional *Debian Wheezy GNU/Linux*. Uma destas VMs foi configurada como roteador e *firewall* e a outra como servidor de aplicações. Outra estação de trabalho *Power Mac*, idêntica, e um notebook rodando *Ubuntu 14.04 GNU/-Linux* foram utilizados como máquinas de desenvolvimento e simulação. Um diagrama da rede criada no LIS é mostrado na figura 3.

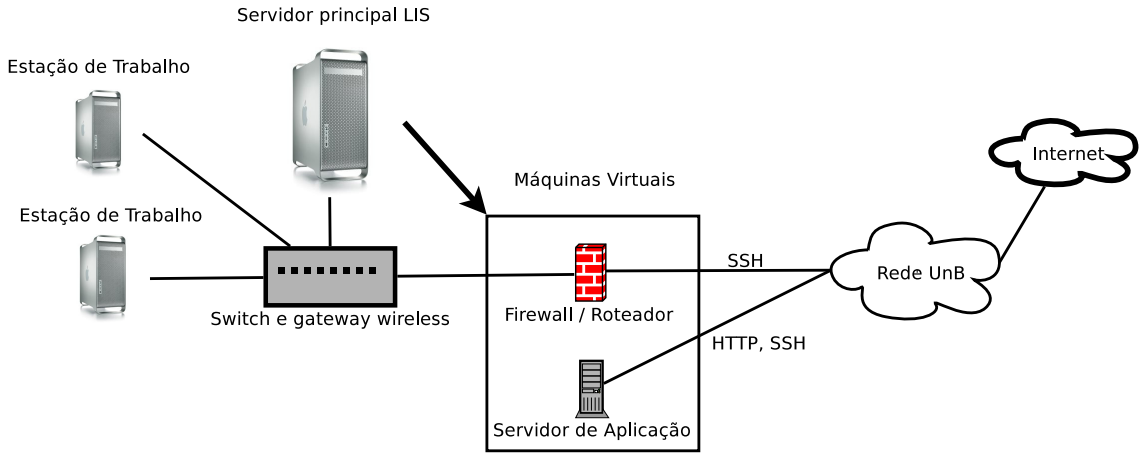

Figura 3 – Rede LIS

O LPH foi utilizado para captura de dados de marcha humana. Este laboratório está equipado para coletar dados de plataformas de força, eletromiógrafos e de marcadores posicionados no corpo do paciente através de câmeras de vídeo (*Motion Capture* - MOCAP). Para este trabalho foi utilizado o software *QTM 3.2* da *Qualisys*.

3.2 DELIMITAÇÃO DO ESTUDO

Este trabalho tem como foco estabelecer uma metodologia de desenvolvimento inicial a um sistema de análise e simulação de marcha. Ele não busca ser extensivo o suficiente para criar um produto pronto para o mercado, mas pretende, através da implementação de funcionalidades reais, estabelecer uma arquitetura mínima e funcional que sirva de base para a construção do sistema. Como consequência, o projeto também integra os principais componentes desta arquitetura, por exemplo, o serviço de banco de documentos, com a *Application Program Interface* (API) web. Um software como este, robusto o suficiente para ser viável no mercado, seria muito caro. Por exemplo um desenvolvedor sênior no mercado de Brasília, não custaria menos de R$ 100.000,00 por ano.

3.3 VISÃO

Apesar deste trabalho ter um objetivo específico e delimitado, dele nasce um projeto maior, cuja visão é o desenvolvimento de um software como serviço para análise e simulação de marcha, utilizando o estado da arte em técnicas para este fim. O software deverá será construído utilizando-se métodos ágeis, e terá uma arquitetura adaptável que permita evolução contínua.

A estratégia é lançar a versão inicial do software como projeto de código livre, conseguir parceiros e procurar um modelo de negócio sustentável para mantê-lo.

3.4 MODELO DE GESTÃO

O software terá um modelo de gestão baseado no método *SCRUM*, como definido por (SHWABER; BEEDLE, 2002). O método não é adotado na plenitude, sendo adaptado segundo as limitações de recursos do projeto.

Nesta fase inicial, o processo conta com um desenvolvedor, que também assume o papel de *scrum master*, e um *product owner*. Devido ao tamanho reduzido da equipe e da localização distinta dos membros, não há *scrum* diário, mas problemas de trabalho cotidianos são resolvidos por telefone, email ou mensagens instantâneas.

O princípio de *time boxing* é mantido. Ficou definido que o *sprint* consiste do prazo de 2 semanas. Ao final do sprint uma reunião em duas fases é realizada. A primeira fase consiste na revisão do *sprint* anterior. Já a segunda fase é o planejamento do próximo sprint.

Um *backlog* de produto é mantido. Na reunião de final de *sprint* é criado um *backlog* de *sprint*. Os itens de *backlog* são mantidos na forma de estórias de usuários, conforme (COHN, 2004). Na figura 4 é apresentada a visão geral do processo.

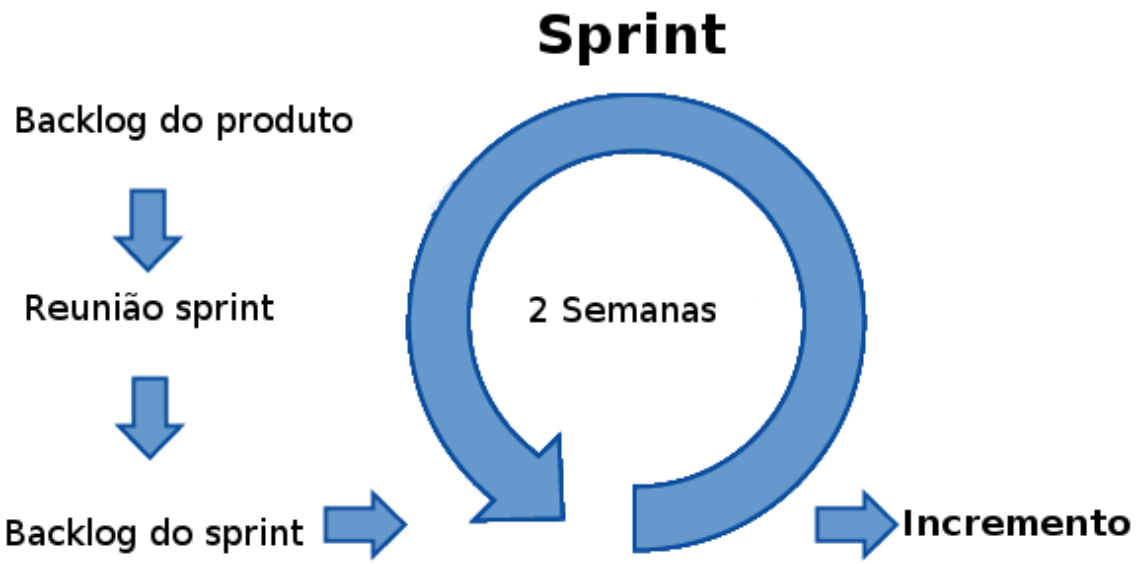

Figura 4 – Processo de desenvolvimento

3.5 MODELO DE ARQUITETURA

O modelo de arquitetura no seu nível mais elevado, pode ser visto como um modelo de três camadas, conforme a figura 5.

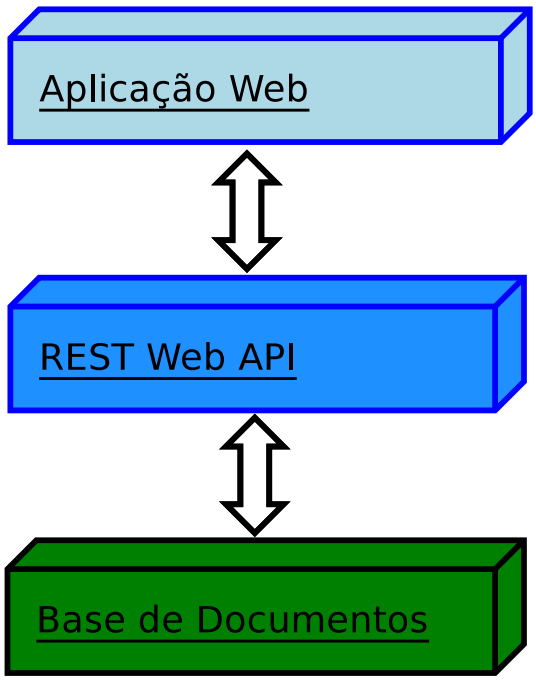

Figura 5 – Camadas arquiteturais

A camada web é responsável pela interação com o usuário. A camada *Web API* é responsável pela lógica de negócio. A camada de base de documentos é responsável pela

persistência dos dados da aplicação.

3.5.1 CAMADA DE APLICAÇÃO WEB

Esta camada foi projetada para rodar em *browsers* que suportam HTML 5. Ela é desenvolvida usando-se Javascript, CSS e HTML. Além disso, adotou-se o *framework* de desenvolvimento web *AngularJS*. (BRANAS, 2014) é um guia introdutório conciso no assunto. Segundo (FREEMAN, 2014), outro guia no assunto, o *AngularJS* se baseia no padrão de projeto *Model-View-Controller* (MVC), e sua enfase é em permitir a criação de aplicações: extensíveis, manuteníveis, testáveis e padronizadas.

3.5.1.1 SOLUÇÃO DE DESIGN WEB

Como, até o momento de conclusão deste trabalho, o projeto não conta com um *web designer*, uma solução técnica para minimizar as consequências deste problema, teve de ser adotada. Os usuários finais do software, profissionais da área de saúde, dificilmente se interessariam pelo mesmo sem uma interface atraente e amigável. A solução adotada foi utilizar a biblioteca *angular-material*. Esta biblioteca, como indicado pelo seu nome, é construída com o *AngularJS*. Ela disponibiliza serviços e diretivas que podem ser usados para construir a interface gráfica da aplicação. Diretivas são componentes que podem ser inseridos diretamente no código HTML da aplicação, dando a aparência de estender a própria HTML. Por exemplo a diretiva *md-button* da biblioteca, é um tipo de botão que não é próprio do HTML. Outros exemplos de diretivas são: caixas de diálogos, barras de ferramentas, barras de progresso, grades, *tooltip*, etc. Esta biblioteca é baseada na especificação *Material Design* criada pela empresa Google. A especificação discorre sobre padrões de designe gráfico e interação com usuário e é baseada no princípio da metáfora de materiais. Esta metáfora é uma teoria unificada de um espaço racionalizado e sistemas de movimento, isto segundo (GOOGLE, 2015). Outra vantagem da biblioteca é que ela é projetada para se adaptar a diferentes tipos de dispositivos com telas de tamanhos diferentes.

A figura 6, mostra um exemplo de uma tela criada com as diretivas do angular-material.

3.5.1.2 ORGANIZAÇÃO DO CÓDIGO FONTE

A criação de um ambiente de desenvolvimento web, para um software de média para grande complexidade, não é uma tarefa trivial de ser resolvida. É necessários criar padrões de organização de arquivos, configurar e instalar pacotes de software para desenvolvimento, testes, implantação, construção de builds, entre outros. Para facilitar esta tarefa, optou-se em utilizar o projeto *angular-seed*. A ideia deste projeto é servir de esqueleto de

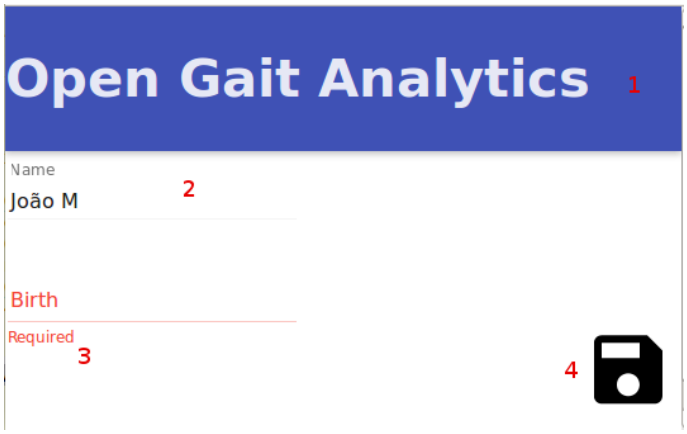

Figura 6 – Exemplo de uma tela criada com *angular-material*. 1) Diretiva *md-toolbar* ; 2) Diretiva *md-input-container*; 3) Diretiva *ng-messages* em conjunto com a *md-input-container*; 4) Diretiva *md-button*.

projetos web que utilizam o framework *AngularJS*. Para usar este projeto basta cloná-lo diretamente do seu repositório *git* no site github.com, conforme o comando abaixo.

```
git clone https://github.com/angular/angular-seed.git
```

Antes de começar a usar o *angular-seed* é necessário instalar o ambiente de execução *Javascript Node.JS*. Este ambiente é utilizado para execução de testes e construção de *builds*. Para uma introdução ao *Node.JS* veja (SYED, 2014).

4 RESULTADOS

5 DISCUSSÃO E CONCLUSÃO

6 TRABALHOS FUTUROS

REFERÊNCIAS BIBLIOGRÁFICAS

BECK, K. *Extreme Programming Explained: Embrace Change*. 2. ed. [S.l.]: Addison-Wesley Professional, 2004. ISBN 9780321278654. 18

BECK, K. et al. *Manifesto para Desenvolvimento Ágil de Software*. 2001. Disponível em: ⟨http://www.agilemanifesto.org/iso/ptbr/⟩. 16

BEYNON, S. et al. Correlations of the Gait Profile Score and the Movement Analysis Profile relative to clinical judgments. *Gait and Posture*, Elsevier B.V., v. 32, n. 1, p. 129–132, 2010. ISSN 09666362. Disponível em: ⟨http://dx.doi.org/10.1016/j.gaitpost.2010.01.010⟩. 14

BRANAS, R. *AgularJS Essentials*. Birmingham: Packt Publishing Ltd., 2014. ISBN 978-1-78398-008-6. 22

CIPPITELLI, E. et al. Kinect as a Tool for Gait Analysis: Validation of a Real-Time Joint Extraction AlgorithmWorking in Side View. *Sensors*, v. 15, n. 1, p. 1417–1434, 2015. ISSN 1424-8220. Disponível em: ⟨http://www.mdpi.com/1424-8220/15/1/1417/⟩. 14

COHN, M. *Users Stories Applied*. Crawfordsville: Addison Wesley, 2004. ISBN 978-0-321-20568-1. 20

DUHAMEL, a. et al. Statistical tools for clinical gait analysis. *Gait and Posture*, v. 20, n. 2, p. 204–212, 2004. ISSN 09666362. 14

FERREIRA, J. a. P.; CRISOSTOMO, M. M.; COIMBRA, a. P. Human gait acquisition and characterization. *IEEE Transactions on Instrumentation and Measurement*, v. 58, n. 9, p. 2979–2988, 2009. ISSN 00189456. 14

FREEMAN, A. *Pro AngularJS*. [S.l.]: Apress, 2014. ISBN 9781430264484. 22

GHOUSSAYNI, S. et al. Assessment and validation of a simple automated method for the detection of gait events and intervals. *Gait and Posture*, v. 20, n. 3, p. 266–272, 2004. ISSN 09666362. 14

GOOGLE. *Material Design*. 2015. Disponível em: ⟨www.google.com.br/design/spec/material-design/introduction.html⟩. 22

GREENE, J.; STELLMAN, A. *Learn Agile*. 1. ed. [S.l.]: O'Reilly Media, Inc., 2014. ISBN 9781449331924. 10, 16, 17

Kristin Runyan; ASHMORE, S. *Introduction to Agile Methods*. [S.l.]: Addison-Wesley Professional, 2014. 17, 18

LAYTON, M. C. *Agile Project Management For Dummies*. 1. ed. [S.l.]: For Dummies, 2012. ISBN 9781118235850. 16

MALAS, B. *Book Review - Gait Analysis: Normal and Pathological Function, 2nd Edition*. 2010. Disponível em: ⟨http://www.oandp.org/reading/gaitfunction.asp⟩. 14

MORAES, J.; SILVA, S.; BATTISTELA, L. Comparison of two software packages for data analysis at gait laboratories. *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439)*, v. 2, p. 1780–1783, 2003. ISSN 1094-687X. 14

MORENO, a. et al. Development of the spatio-temporal gait parameters of Mexican children between 6 and 13 years old data base to be included in motion analysis softwares. *2009 Pan American Health Care Exchanges - PAHCE 2009*, n. 2, p. 90–93, 2009. 14

PERRY, J.; BURNFIELD, J. M. *Gait Analysis Normal and Pathological Function*. 2nd. ed. [S.l.]: SLACK Inc., 2010. ISBN 978-1-55642-766-4. 14

SHWABER, K.; BEEDLE, M. *Agile Software Development with Scrum*. [S.l.]: Pearson, 2002. ISBN 0-13-067634-9. 20

SYED, B. A. *Beginning Node.js*. [S.l.]: Apress, 2014. ISBN 9781484201879. 23

VIEIRA, A. et al. Software for human gait analysis and classification. In: *4th Portuguese BioEngineering Meeting*. Porto: [s.n.], 2015. 14