Um Software coma Serviço para Análise de Marcha

Roberto Aguiar Lima, Vera Regina Da Silva Marães, and Lourdes Mattos Brasil

***Abstract*—Este trabalho descreve o desenvolvimento da primeira versão de um software como serviço para análise e simulação de marcha humana. A principal vantagem deste enfoque é a disponibilização via web do software. Depois de implantado num servidor central, os usuários poderão acessar o software através de um browser recente que suporte HTML 5. O objetivo do software é minimizar a necessidade de pesquisadores da área de escrever código em suas pesquisas. O software permite importar dados gerados de marcadores de superfície a partir de um sistema de captura de movimentos, plotar a progressão espacial destes marcadores, os ângulos, velocidades angulares e acelerações angulares dos marcadores. Além disso é possivel visualizar os marcadores numa animação em 3D. O código fonte está disponível como software livre e frequentemente recebe novas funcionalidades, sendo que uma nova comunidade está sendo criada em torno do software.**

***Index Terms*—Análise de marcha, software como serviço.**

I. Introduction

COM o advento da internet foi possível criar serviços que podem ser implementados e implantados num servidor central e disponibilados para qualquer parte do mundo. Além disso, os browsers webs modernos se tornaram uma verdadeira plataforma, permitindo a criação de interfaces de usuário ricas, inclusive com apresentação de gráficos e animações em 3D.

Estas duas tecnologias, servidores e browsers web, podem ser usadas como base para o que se conhece como software como serviço. Em [1] são citadas vária vantagens desta abordagem, destacando-se entre elas a fácil disponibilização do software, por exemplo, o usuário do software não precisa instalar uma versão do software em seu computador, basta acessá-lo pelo browser web que o sistema estará disponível.

Segundo [2], apesar dos avanços ocorridos na análise de marcha até meados do meio do século vinte, foi só após o advento dos computadores modernos que a análise de marcha clínica tornou-se amplamente disponível. Pacotes do software para análise de marcha foram implementados [3]. Mas até agora no século 21, nenhum fornecedor de software comprometeu-se a entregar um sistema de análise de máquina disponibilizado como serviço na web, em outras palavras, profissionais de saúde ou pesquisadores de análisde de marcha que querem usar um software para isto, têm que utilizar software instalados numa máquina com arquitetura específica e sistema operacional específico. Eles ficam responsáveis por backups dos dados, ao surgirem novas funcionalidades no software, os mesmos têm que instalá-lo novamente, se eles quiserem compartilhar dados terão que copiá-los e enviá-los a seu destinatário. Todos esses problemas são eliminados quando se utiliza Software como Serviço. Além disto técnicas de análise de marcha são bem documentadas hoje em dia [4] [5] [6], o que facilita a escolha de características a serem adotados para um software. Com isto em mente, o presente trabalho descreve a implementação de um software como serviço para análise de marcha, que foi inicialmente concebido por [7]. O software implementado é capaz de recuperar dados gerados por um sistema de captura de imagens, no caso dados de marcadores de superfície, nomear marcadores, gerar gráficos de progreção espacial dos marcadores, definir ângulos usando os marcadores, plotar gráficos dos ângulos, velocidades angulares e acelerações angulares, além de exibir uma animação interativa em 3D dos marcadores.

R. A. Lima e L. M. Brasil são da FGA/UnB
V. R. S. Marães é da FCE/UnB

II. Metodologia

Dois ambientes de pesquisa foram utilizados, um foi o Laboratório de Performance Human (LPH) na Faculdade Ceilândia (FCE) da Universidade de Brasília (UnB), e o Laboratório de Informática em Saúde (LIS) na Faculdade Gama (FGA) da UnB. No LPH foram coletados os dados e no LIS foi desenvolvido o software.

O projeto no qual ocorreu a coleta foi aprovado pelo Comitê de Ética da Faculdade de Saúde da UnB, processo N11911/12.

A. Coleta de Dados

Os dados coletados foram captados através de câmeras Oqus MRI da Qualisys usando-se o software QTM do mesmo fabricante. Os dados são referentes a marcadores de superfície dispostos ao longo do corpo do paciente. O processo de coleta é descrito na Fig. 1. Para este trabalho um paciente saudável do sexo masculino com idade entre 20 e 30 anos foi selecionado. Também é necessario definir quais os pontos no corpo do paciente devem receber marcadores de superficie, neste caso os pontos de interesse foram o trocanter esquerdo, joelho esquerdo e tíbia esquerda. O proximo passo se refere a coleta dos dados em si. O paciente deve repetir um ciclo de marcha confortavel de aproximadamente 5 segundos, por 5 vezes na frente das câmeras. Os dados devem ser convertidos para formato adequado à linguagem MATLAB. Esta opção está disponível no QTM.

B. Processo de desenvolvimento

O processo de desenvolvimento foi inspirado no scrum [8]. O scrum é um porcesso de desenvolvimento ágil [9], que tem como principais características ser iterativo e incremental. Estas características são essenciais para lidar com a natureza volátil dos requisitos de software.

O arcabouço do processo pode ser visto na Fig. 2. O processo consiste de iterações, chamadas de sprint, que duram em média duas semanas. Um backlog de produto é administrado por um product owner. Este backlog fica sempre em aberto,

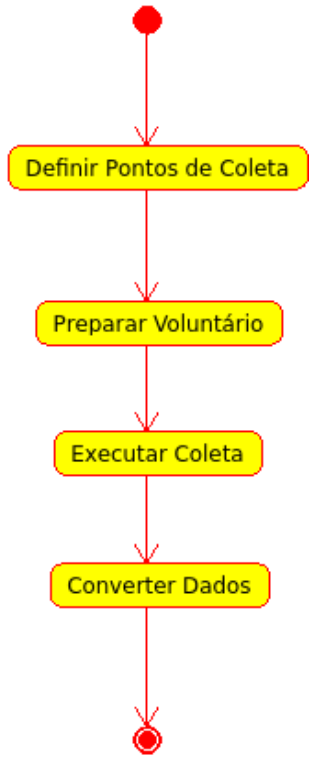


Fig. 1. Processo de coleta de dados.

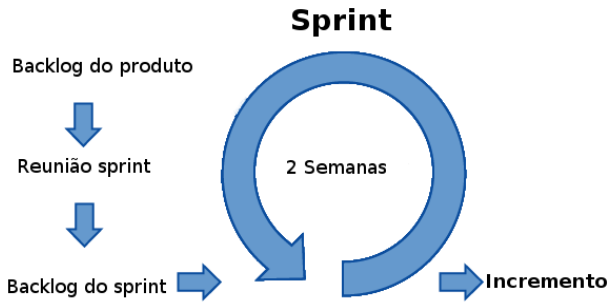


Fig. 2. Processo de desenvolvimento.

recebendo adições na forma de histórias do usuário [10]. Antes de cada iteração uma reunião com a equipe é feita. Esta reunião é dividida em duas fases. Na primeira fase são apresentados os resultados da última iteração. Na segunda fase histórias do usuário contidas no backlog do produto são selecionadas. Estas histórias compõe o backlog do sprint, que são as funcionalidades a serem implementadas durante o sprint. Ao final do sprint um incremento é produzido. O incremento consiste num pedaço de software funcional pronto para ser implantado.

C. Arquitetura da Aplicação

A Fig. 3 mostra uma visão de alto nível da arquitetura do software. 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.

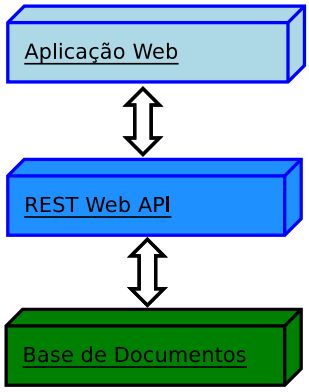


Fig. 3. Visão de alto nível da arquitetura.

1) Camada 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. Esta tecnologia foi descrita em [11]. Uma das vantagens de se usar o AngularJS é a possibilidade de se criar diretivas. Diretivas são componentes que podem ser embutidas num template HTML. Neste projeto também foi utilizada a biblioteca de diretivas Angular-Material. 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. Segundo [12], esta metáfora é uma teoria unificada de um espaço racionalizado e sistemas de movimento. A intenção da escolha desta biblioteca foi a de criar uma experiência de usuário aceitável para profissionais da área de sáude.

Para executar animações gráficas em 3D, foi utilizada a biblioteca ThreeJS, descrita em [13]. Esta é uma biblioteca de alto nível que se aproveita do padrão WebGL, implementado nos broswers webs recentes. O padrão WebGL, ver [14] usa nativamente a GPU do computardor, permitindo assim criar aplicações que necessitam boa performance para gerar gráficos.

A Fig. 4 mostra a interação entre os principais componentes da camada web. O usuário ao interagir com a aplicação, através de seu browser, dispara algum evento, por exemplo, um clique num botão. O framework AngularJS detecta o evento e o repassa para um controlador desenvolvido especiamente para a aplicação. Caso seja necessária a execução de alguma lógica de negócio e dados da aplicação, o controlador faz uma requisição para uma Facade, também desenvolvida para a aplicação. Esta Facade é responsável por encapsular a requisição transformado-a numa requisição HTTP para o backend, ou seja, o servidor da aplicação na internet. Despois da resposta, que vem no formato JSON, ser enviada pelo backend e recebida pela Facade, esta a encaminha para o Controlador que fica responsável por executar a lógica necessária para os dados serem apresentados ao usuário.

2) Camada REST Web API: Esta camada é responsável pela execução da lógica de negócio, como por exemplo, fazer cálculos de ângulos e extrair dados do arquivo em formato MATLAB proveniente do QTM. Esta camada foi

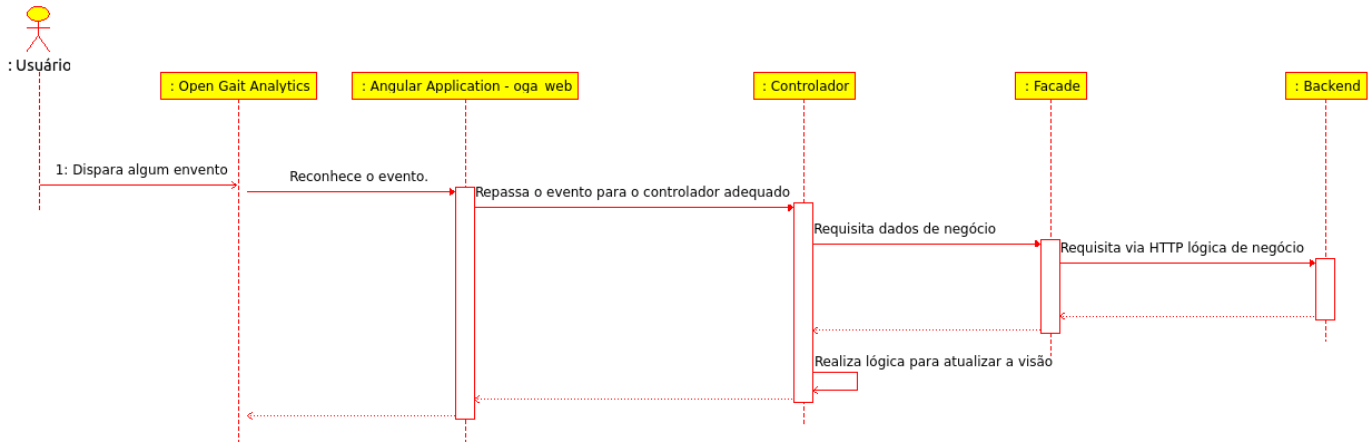


Fig. 4. Interação dos componentes da camada Web. O usuário, usando o seu browser web, interage com a aplicação, clicando num botão por exemplo, o framework AngularJS reconhece o evento e dispara o procedimento corrente para tratá-lo, que está dentro de uma classe controladora. Caso seja necessário recuperar algum dado no backend, o controlador delega esta tarefa para uma classe Facade.

implementada usando-se o estilo arquitetural Representational Estate Transfer (REST), descrito por [15]. A maior vantagem desta tecnologia é o alto desacoplamento com a camada web. Para implementar este estilo foi escolhido o framework Flask [16]. Esta é uma biblioteca escrita em Python, que segundo [16] é conhecida por sua filosofia minimalista, permitindo-se criar um projeto simples no início, mas que permite evolução para modelos mais complexos conforme a necessidade do momento.

A Fig. 5 mostra a interação entre alguns componentes da camada REST Web API. Depois do browser do usuário encaminhar a requisição HTTP para o servidore web, este o encaminha para a aplicação escrita em Python com o framework Flask, o framework roteia a requisição para uma função escrita em Python, esta pode acessar bibliotecas disponíveis para a aplicação ou fazer uma consulta ao banco de documentos através da biblioteca PyMongo. A lógica de negócio pode ser implementada dentro desta função, ou caso seja muito complexa, numa biblioteca escrita especiamente para a funcinalidade. Depois de processado o resultado é convertido para o formato JSON e retornado para o browser do usuário.

3) Camada de Base de Documentos: A camada de base de documentos é responsável pela persistência dos dados da aplicação. Esta camada basicamente é formada por uma base implementada usando-se a tecnologia MongoDB. Uma introdução ao MongoDB é feita em [17]. A escolha desta tecnologia, em detrimento a um banco de dados relacional que é mais comum, foi devido a natureza multidimensional dos dados, que é mais facilmente modelada num banco de documento, e também por causa das facilidades relativas a escalabilidade do MongoDB. A Fig. 6 mostra a base de dados oga, que possui duas coleções de dados patients e positionals_data.

III. Resultados

O software desenvolvido permite o cadastro de pacientes e de amostras de dados espaciais coletadas pelo software QTM. Depois de se cadastrar o paciente, o usuário deve incluir uma nova amostra de dados. Para isso ele deve importar o arquivo

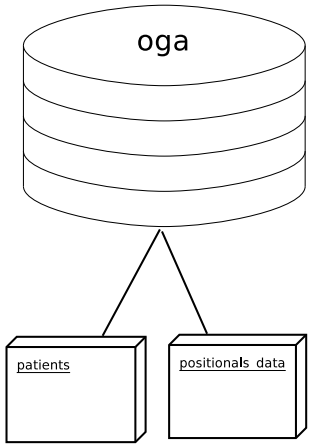


Fig. 6. Base de documentos da aplicação.

com a amostra para o sistema. Após o processo de importação a tela descrita na Fig. 7 é mostrada.

A partir desta tela é possível executar uma animação dos dados coletados (Fig. 8). Nesta tela é possífel acelerar ou desacelerar, aplicar zoom in e zoom out, fazer pan e visualizar um quadro específico da animação. O recurso de visualizar o quadro é importante pois é com ele que se deve descobrir o quadro referente ao contato inicial e ao balanço terminal. Estes dados devem ser cadastrados na tela mostrada na Fig. 7. Sem este procedimento não é possível visualisar os gráficos de progreção espacial e de ângulos.

O software permite a nomeação dos marcadores vistos na animação. Depois de se nomear um marcador específico, por exemplo, o calcâneo esquerdo, é possível ver sua progressão espacial como na Fig. 10.

Outro recurso importante da aplicação é o cadastramento de ângulos. Este recurso pode ser visto na Fig. 11. A partir desta tela é possível ver os ângulos (Fig. 11), velocidades angulares (Fig. 12) e acelerações angulares (Fig. 13).

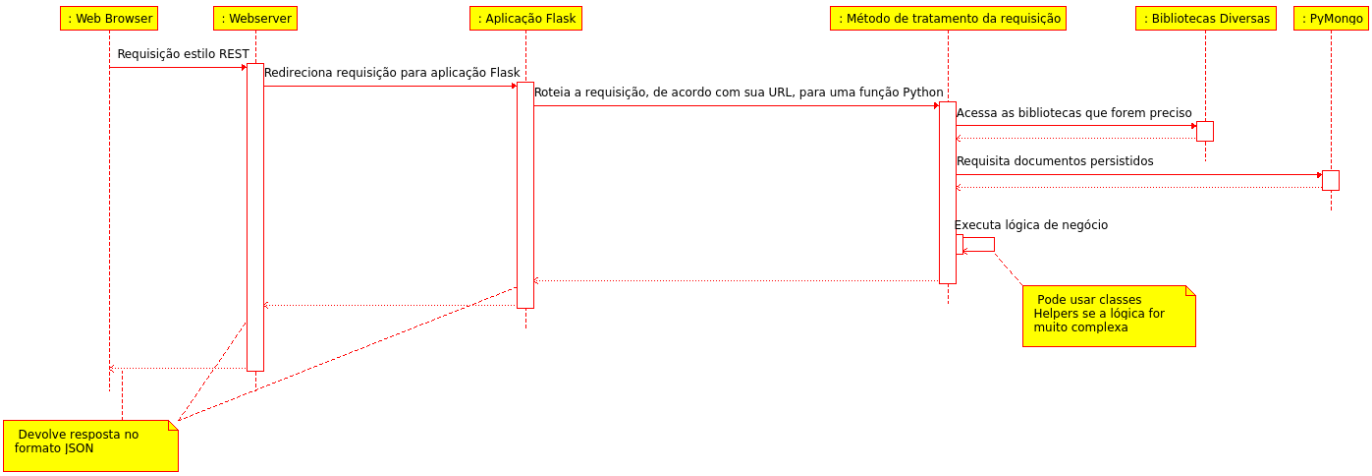


Fig. 5. Interação dos componentes da camada REST Web API. Quando a aplicação web necessita de dados, ela faz uma requisição HTTP a camada REST Web API. Esta requisição é recebida pelo servidor web que a redireciona para a aplicação Flask. A aplicação Flask identifica a requisição e a encaminha para uma função Python apropriada para ser processada. Esta função tem acesso a diversas bibliotecas para ajudar no processamento, como NumPy, Matplotlib, bibliotecas específicas da aplicação entre outras. Se algum dado persistido for necessário, também pode-se utilizar a biblioteca PyMongo para buscá-los na base de documentos. Finalmente uma resposta HTTP é criada. O conteúdo desta resposta é um objeto no formato JSON que será consumido pela camada web.

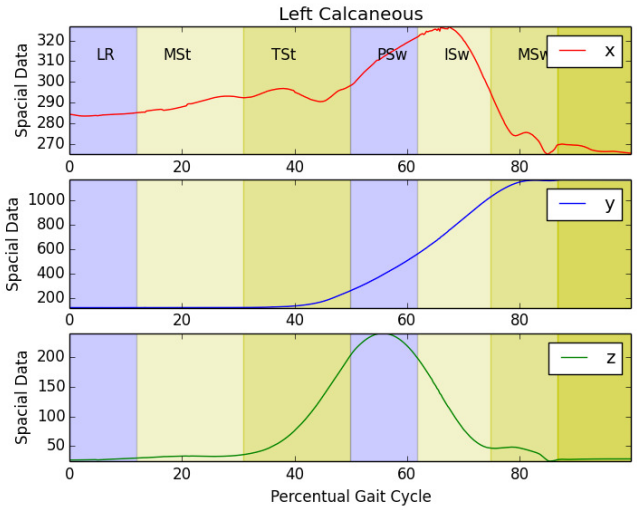


Fig. 9. Progressão no espaço do calcâneo esquerdo.

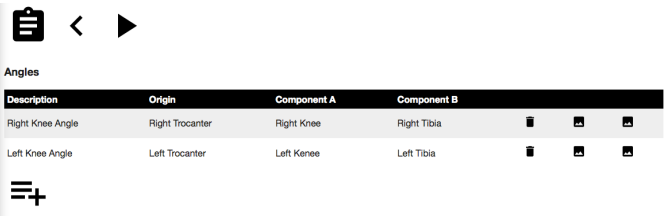


Fig. 10. Recurso para criação de ângulos.

IV. Discussão

A criação de um software específico para análise de marcha totalmente disponível em ambiente web é uma inovação na área. A maior vantagem deste enfoque é a fácil disponibilização do software, que depois de ter uma versão implantada na internet, pode ser acessado de qualquer parte

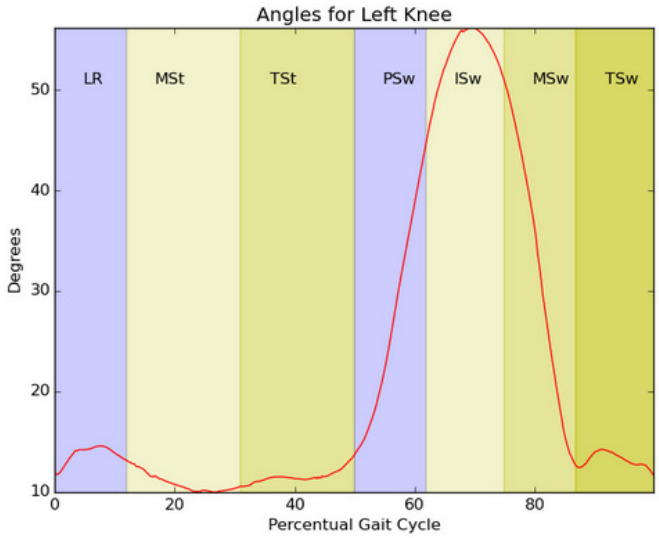


Fig. 11. Ângulo de um joelho.

do mundo, desde que o usuário tenha acesso a um browser HTM5 recente. Outro tema importante a ser levantado sobre este projeto, é a possibilidade de se formar uma base com dados de marcha humana coletados ao redor do globo, isso por si só seria de valor inestimável a pesquisadores da área.

O software está disponibilizado como software livre. A intenção da equipe é atrair outros desenvolvedores e profissionais de saúde que contribuam e usufruam do mesmo. Também não se descarta a idéia de se criar um projeto para crowdfounding e assim permitir que mais pessoas contribuam com o mesmo, além é claro de se conseguir recurso para mantê-lo.

Como trabalhos futuros, espera-se que novos métodos de coletas de dados sejam suportados, como por exemplo, plataformas de forças, EMG e IMU. Além disso, um módulo

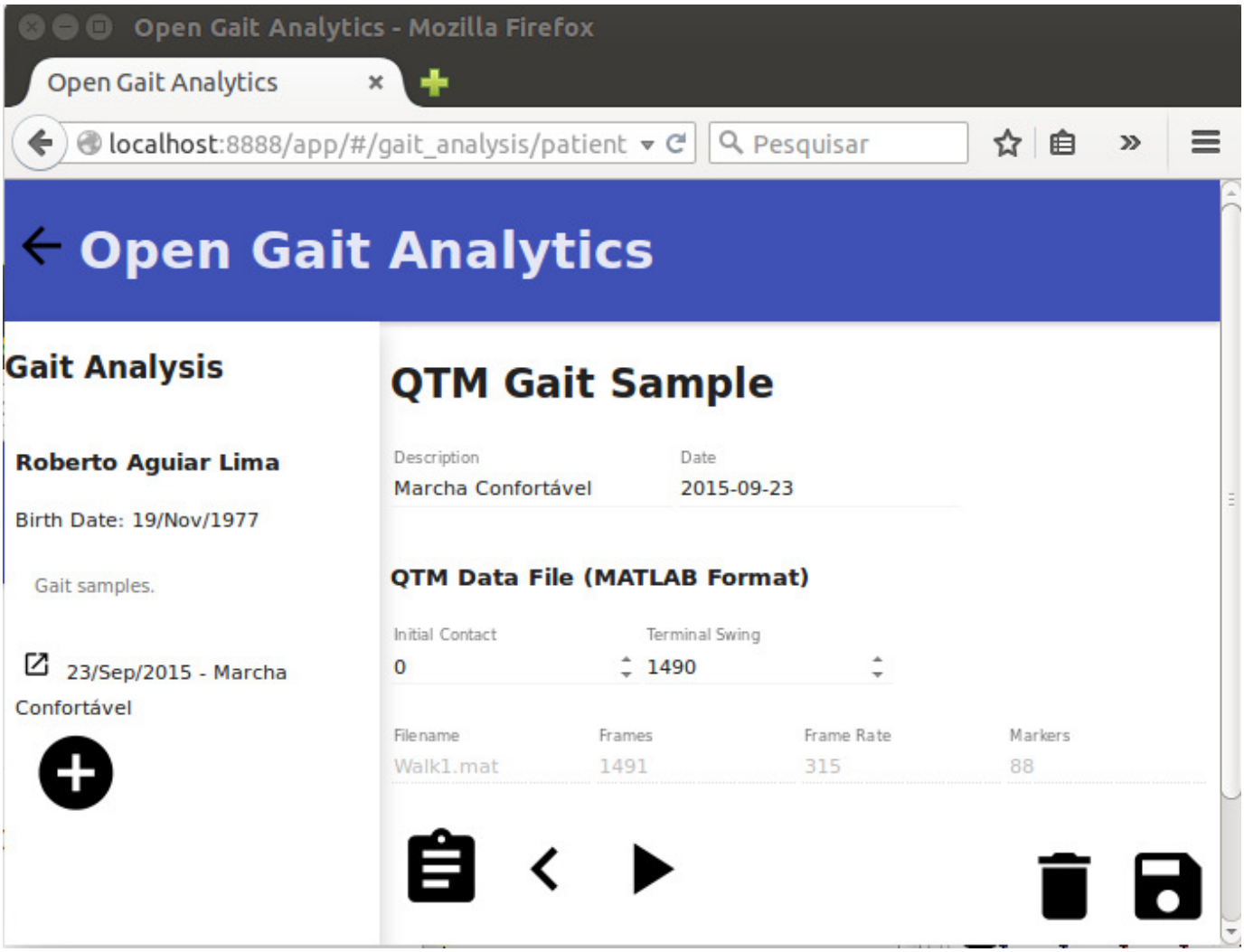


Fig. 7. Dados importados do QTM.

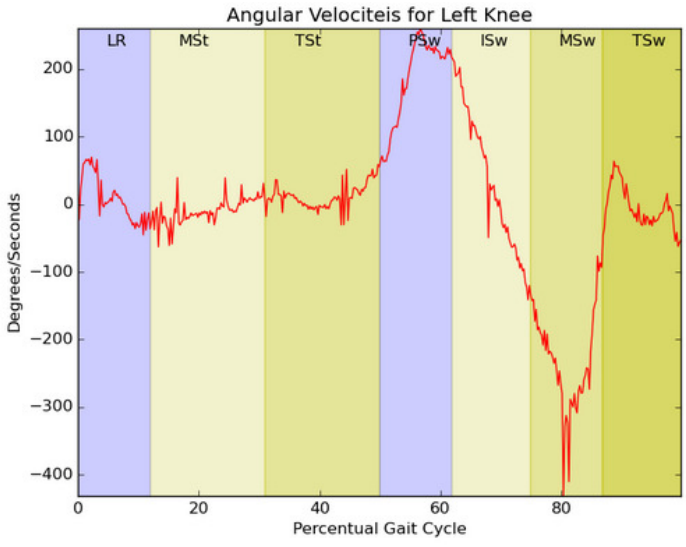


Fig. 12. Velocidades angulares de um joelho.

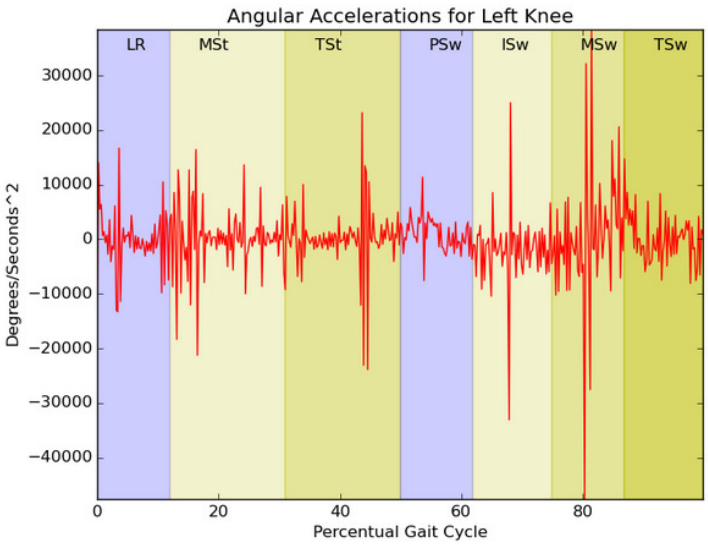


Fig. 13. Acelerações angulares de um joelho.

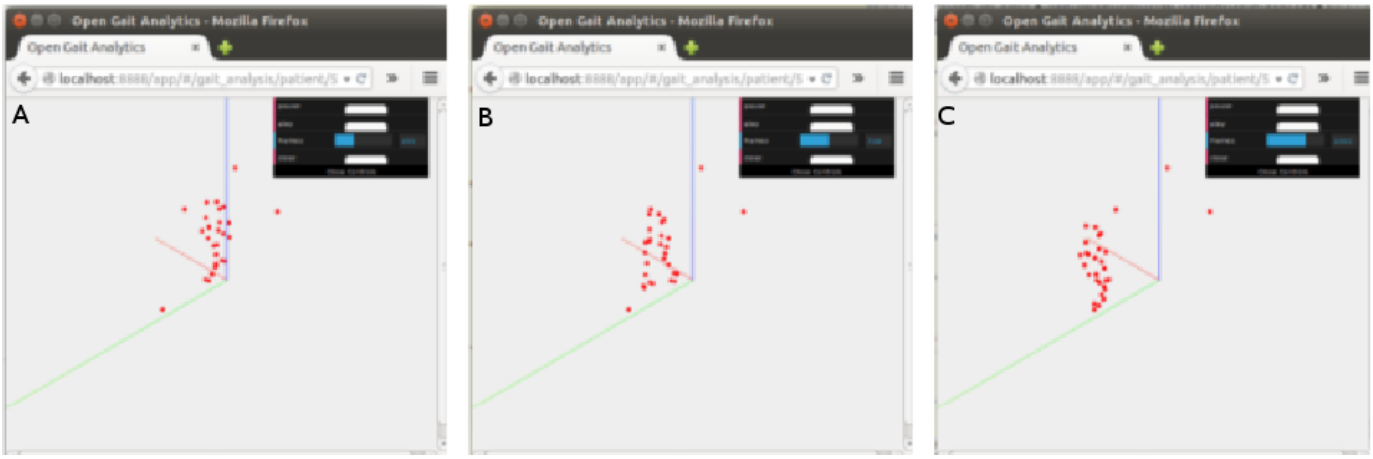


Fig. 8. Animação em 3D dos dados coletados. A) Contato inicial do membro inferior esquerdo. B) Membro inferior esquerdo no período de balanço. C) Membro inferior esquerdo no balanço terminal.

de simulações está sendo desenvolvido e se espera anexar métodos de aprendizado de máquina para classificação e regressão.

V. Conclusion

O presente trabalho apresentou o software desenvolvido pelo LIS FGA/UnB em cooperação com o o LPH FCE/UnB. O código fonte do projeto foi disponibilizado sob a licença MIT [18] no endereço web http://github.com/rob-nn/open_gait_analytics.

O software conta com recursos para importação de dados oriundos do software QTM da Qualisys. Permite fazer a nomeação de marcadores fixados no corpo do paciente no momento da coleta, além de habilitar a visualisação destes dados em gráficos. Também é possível cadastrar ângulos e visualizá-los em gráficos assim como suas velocidades angulares. Uma animação em 3D com os dados é visualizável. Todos os gráficos são visualizados de acordo com as fase do ciclo de marcha.

O sistema é implantável na internet e é acessível através de um browser HTML5 recente. Ele também terá sua continuidade, implantação e adição de novos recursos, sob a tutela do LIS FGA/UnB e do LPH FCE/UnB.

Acknowledgment

O primeiro autor agradece ao apoio financeiro concedido pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) para realização do projeto. Os autores agradecem também ao LIS FGA/UnB e ao LPH FCE/UnB, pela disponibilização de equipamentos e instalações para a execução do projeto.

References

[1] A. Fox and D. Patterson, *Engineering Long-Lasting Software: An Agile Approach Using SaaS and Cloud Computing*. Strawberry Canyon LLC, 2012.

[2] R. Baker, "The history of gait analysis before the advent of modern computers," *Gait and Posture*, vol. 26, no. 3, pp. 331–342, 2007.

[3] J. Moraes, S. Silva, and L. Battistela, "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)*, vol. 2, pp. 1780–1783, 2003.

[4] J. Perry and J. M. BurnField, *Gait Analysis Normal and Pathological Function*, 2nd ed. SLACK Inc., 2010.

[5] H. Skinner, *Gait Analysis an Introduction fourth*, 4th ed. Elsevier Ltd, 2007.

[6] J. P. Ferreira, M. M. Crisostomo, and a. P. Coimbra, "Human gait acquisition and characterization," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 9, pp. 2979–2988, 2009.

[7] R. A. Lima, "Implementando um Software como Serviço para Análise e Simulação de Marcha Humana," 2015.

[8] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*, 1, Ed. Pearson, 2001.

[9] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, "Manifesto para Desenvolvimento Ágil de Software," 2001. [Online]. Available: http://www.agilemanifesto.org/iso/ptbr/

[10] M. Cohn, *Users Stories Applied*. Crawfordsville: Addison Wesley, 2004.

[11] R. Branas, *AgularJS Essentials*. Birmingham: Packt Publishing Ltd., 2014.

[12] Google, "Material Design," 2015. [Online]. Available: www.google.com.br/design/spec/material-design/introduction.html

[13] J. Dirksen, *Learning Three.js - the JavaScript 3D Library for WebGL*, 2nd ed. Packt Publishing, 2015.

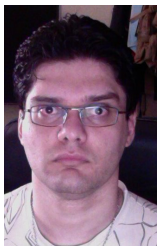
[14] K. Matsuda and R. Lea, *WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL (OpenGL)*. Addison-Wesley Professional, 2013.

[15] M. Grinberg, *Flask Web Development*, 1st ed. O'Reilly Media, Inc., 2014.

[16] I. Maia, *Building Web Applications with Flask*, 1st ed. Packt Publishing, 2015.

[17] E. Plugge, P. Membrey, and D. Hows, *MongoDB Basics MongoDB Basics*, 1st ed. Apress, 2014.

[18] MIT, "The MIT License (MIT)," 2015. [Online]. Available: https://opensource.org/licenses/MIT



Roberto Aguiar Lima Possui graduação na área em Ciência da Computação pela Universidade Católica de Brasília (2000). Atualmente cursa o programa de Mestrado em Engenharia Biomédica da Universidade de Brasília. Possui mais de 20 anos como desenvolvedor de software.

Vera Regina Da Silva Marães Possui graduação em Fsioterapia pela Universidade Federal de São Carlos (1995), mestrado em Biologia Celular e Estrutural pela Universidade Estadual de Campinas (1999) e doutorado em Fisioterapia pela Universidade Federal de São Carlos (2004). Atualmente é docente adjunto III da Faculdade de Ceilândia; do Programa de Pós-Graduação em Engenharia Biomédica e do Pro-Ensino na Saúde da Universidade de Brasília. Tem experiência na área de Fisioterapia, atuando principalmente nos seguintes temas: ensino na saúde, adaptação de próteses em amputados, funcionalidade humana, fisiologia do exercício, reabilitação cardíaca e variabilidade da frequência cardíaca.

Lourdes Mattos Brasil Lourdes Mattos Brasil possui graduação em Engenharia Elétrica pela Universidade Federal de Santa Catarina (1989), Mestrado em Engenharia Elétrica/Engenharia Biomédica pela Universidade Federal de Santa Catarina (1994) e concluiu o doutorado em Engenharia Elétrica/Sistemas de Informação - Engenharia Biomédica pela Universidade Federal de Santa Catarina em 1999, sendo parte realizado na Facultés Universitaires Notre-Dame de La Paix (FUNDP), Namur Belgium (1998). Realizou estágio Pós-Doutoral no Programa de Pós-Graduação em Engenharia Biomédica da Universidade Federal da Paraíba (2001-2002). Atualmente é Professor Adjunto da Universidade de Brasília (UnB), Faculdade UnB-Gama (FGA), onde atua na Engenharia Eletrônica, bem como é coordenadora do Lato Sensu em Engenharia Clínica e dos Laboratórios de Informática em Saúde (LIS) e de Nanotecnologia (NTEC-FGA). Também é pesquisadora/docente do Stricto Sensu, Mestrado em Engenharia Biomédica. Tem experiência na área de Engenharia Biomédica, com ênfase em Informática em Saúde, atuando principalmente nos seguintes temas: informática em saúde, inteligência artificial em saúde, engenharia biomédica, redes neurais artificiais, sistemas baseados no conhecimento, descoberta de conhecimento, realidade virtual, sistemas tutores inteligentes, sistemas especialistas híbridos, sistemas inteligentes, processamento de imagens, nanotecnologia e gestão do conhecimento. Suas produções técnicas-científicas permeiam essas áreas, bem como suas publicações de livros, prêmios, palestras/seminários, pareceres, consultorias. Também participa como membro das sociedades brasileira e internacional: SBC, SBEB e IEEE.