**Calendário Gerenciador de Eventos**

**Aline N. Favretto**​**1**​**, Gustavo L. Schroeder**​**2**​**, Leonardo S. de Paula**​**3**

1​Ciência da Computação – Universidade do Vale do Rio do Sinos (Unisinos)

Av. Unisinos, 950 – 93.022-750 – São Leopoldo – RS – Brasil

anfavretto@edu.unisinos.br1, gschroeder@edu.unisinos.br2

***Resumo.*** *Este meta-artigo descreve os objetivos, funcionalidades, limitações e recursos utilizados para o desenvolvimento da aplicação Calendário Gerenciador de Eventos, desenvolvido utilizando em seu core o framework JavaScript AngularJS. Todas as tecnologias utilizadas foram escolhidas devido ao interesse dos participantes em aprofundar conhecimentos nas mesmas.*

# Introdução

No presente documento detalhamos o trabalho prático desenvolvido para a disciplina de Desenvolvimento para a Web no semestre 2017/01 na Universidade do Vale do Rio dos Sinos (UNISINOS). O trabalho tem como objetivo criar um gerenciador de calendário no qual seja possível adicionar, editar e excluir eventos e lembretes, bem como visualizá-los por dia, semana e mês e receber avisos quando for a hora marcada para os mesmos.

As tecnologias utilizadas para o mesmo foram AngularJs, Bootstrap, Modernizr,

LocalStorage, Moment.js, LESS CSS e HTML5.

# Objetivo

Os objetivos do desenvolvimento são: criar uma interface simples e intuitiva, manter dados inseridos (permanentemente, se recurso disponível em navegador utilizado, caso contrário, mantê-los durante a sessão). Para alcançar os mesmos utilizamos das bibliotecas JavaScript já descritas na Introdução e de boas práticas de programação.

# Funcionalidades

As funcionalidades previstas são: adicionar, editar e excluir eventos e lembretes, visualizá-los por dia, semana e mês e receber avisos quando for a hora marcada para os mesmos.

# Limitações

Algumas limitações foram encontradas durante o desenvolvimento da aplicação, como a aprendizagem de novas tecnologias, como AngularJS e bibliotecas JavaScript que os membros do grupo não conheciam e não tinham experiência.

Devido a isso implementação referente a edição de eventos e lembretes não foi concluída, além disso o alerta dos eventos e lembretes também não foi desenvolvida.

# Recursos utilizados

Os recursos utilizados serão descritos a seguir.

**5.1. AngularJS**

Utilizado para o desenvolvimento da camada lógica da aplicação (Controller). Foi escolhido por ser um framework, com uma curva de aprendizado moderada. Devido a ser uma biblioteca mais complexa não foi utilizado 100% de seu potencial para a aplicação.

Entre os facilitadores do framework que levaram a sua escolha estão:

* O framework tem a filosofia que os testes da aplicação são tão importantes quando seu desenvolvimento.
* Injeção de dependência
* Two-way data binding
* Reutilização de componentes

**5.2. Bootstrap**

É um dos mais famosos frameworks para design de WebSites e aplicações Web. Foi escolhida pela gigante gama de Browsers compatíveis com a tecnologia e facilidade de desenvolvimento utilizando Less CSS, além da possibilidade de reutilização de componentes.

**5.3. Moment.js**

Utilizado para manipulação, transformação e formatação de datas. Foi escolhida por se uma biblioteca leve e por suportar datas em todos os formatos, locais, horas relativas e fusos horários.

**5.4. Modernizr**

Modernizr é uma biblioteca JavaScript utilizada para detecção de funcionalidades do Web Browser, como suporte a funcionalidades refentes ao HTML5 e CSS3. É utilizado em nossa aplicação web para verificação se o Browser possui suporte a Local Storage.

**5.5. Local Storage**

A opção por Local Storage foi feita devido ao usuário poder sair do navegador e quando abrir o calendário ver os eventos que cadastrou, pois isso foi decidido por persistir os dados de forma local.

**5.6. Less CSS**

Less é um extensor da linguagem CSS, adicionando várias funcionalidades para deixar o código CSS de fácil manutenção, e pode ser estendida facilmente.

Foi escolhida para ser trabalhada juntamente com o framework Bootstrap.

# Conclusão

All images and illustrations should be in black-and-white, or gray tones, excepting for the papers that will be electronically available (on CD-ROMs, internet, etc.). The image resolution on paper should be about 600 dpi for black-and-white images, and 150-300 dpi for grayscale images. Do not include images with excessive resolution, as they may take hours to print, without any visible difference in the result.

# References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in brackets, e.g. [Knuth 1984], [Boulic and Renault 1991]; or dates in parentheses, e.g. Knuth (1984), Smith and Jones (1999).

The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the subsequent should be indented by 0.5 cm.

# References

Boulic, R. and Renault, O. (1991) “3D Hierarchies for Animation”, In: New Trends in Animation and Visualization, Edited by Nadia Magnenat-Thalmann and Daniel Thalmann, John Wiley & Sons ltd., England.

Dyer, S., Martin, J. and Zulauf, J. (1995) “Motion Capture White Paper”, http://reality.sgi.com/employees/jam\_sb/mocap/MoCapWP\_v2.0.html​, December.

Holton, M. and Alexander, S. (1995) “Soft Cellular Modeling: A Technique for the Simulation of Non-rigid Materials”, Computer Graphics: Developments in Virtual Environments, R. A. Earnshaw and J. A. Vince, England, Academic Press Ltd., p. 449-460.

Knuth, D. E. (1984), The TeXbook, Addison Wesley, 15​th​ edition.

Smith, A. and Jones, B. (1999). On the complexity of computing. In ​*Advances in Computer Science*​, pages 555–566. Publishing Press.