Sistema web para testar *live streams* de compartilhamento de conhecimento

Especificação Arquitetural

Versão 2.0

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 05/mai/20 | 1.0 | Versão inicial. | Leonardo Gravina |
| 29/jun/20 | 2.0 | Versão final, apresentada para a orientadora. | Leonardo Gravina |

Índice Analítico

1. Detalhamento Técnico dos Requisitos não-funcionais 4

1.1 Usabilidade 4  
1.2 Confiabilidade 4  
1.3 Desempenho 4

1.4 Suportabilidade 5  
1.5 Segurança 5  
1.6 Portabilidade 5  
1.7 Outros Requisitos Tecnológicos 5

2. Definição de *Framework* do Projeto 5

2.1 *Framework* de *back-end* 5  
2.2 *Framework* de *front-end* 6

3. Definição de Padrões Arquiteturais de Projeto 6  
 3.1 Camadas da aplicação 6

4. Visão de Implantação da Solução 7

Especificação Arquitetural

# Detalhamento Técnico dos Requisitos não-funcionais

## Usabilidade

- RN1: Tempo de treinamento moderador - Usuários normais devem se tornar produtivos nas operações destinadas ao perfil de moderador (Manter agenda de palestras, Manter palestras realizadas, Manter *newsletter*) com 4 horas de treinamento. E usuários avançados devem se tornar produtivos nestas operações com 2 horas de treinamento.

Vídeos curtos e manual devem ser produzidas explicando cada uma das operações citadas acima.

- RN2: Criação e envio de *newsletter* - O sistema web deve prover *interface* visual para criação e envio de *newsletter*. Dessa forma, o usuário moderador não precisará entender de HTML ou qualquer linguagem de programação para essas tarefas.

Utilizar o módulo ‘Simplenews’, que fornece uma base para criação e envio de *newsletter* com *interface* visual. Para permitir que as newsletter sejam enviadas o formato HTML (e não *Plain*) utilizar os módulos ‘Swift Mailer’ e ‘Mail System’.

- RN3: Tempo de criação e envio de *newsletter* - Com a *interface* visual para a criação e o envio de *newsletter* citada no RN2, essas tarefas em conjunto devem ser executadas em menos de 30 minutos pelo usuário normal com perfil de moderador.

Criar menu personalizado para o perfil moderador com as opções relativas a newsletter e deixar o tipo de conteúdo padrão com o layout da *newsletter* (cabeçalho, largura e tipo e tamanho de fonte). Isso agilizará as tarefas do usuário, que ficarão dentro do tempo previsto.

## Confiabilidade

- RN4: Disponibilidade - O sistema web deve estar disponível 95% do tempo.

O sistema web ficará hospedado nos servidores da empresa DialHost, que possui Data Center próprio com tecnologia de ponta, atendimento rápido via chat e telefone 24h por dia/7 dias por semana. Com essa estrutura, o sistema web ficará disponível conforme indicado.

- RN5: Tempo Médio entre Falhas (MTBF) - O Tempo Médio entre Falhas (MTBF) deve ser de no mínimo 12 horas.

Os sistemas web desenvolvidos a partir do framework Drupal são estáveis e combinado com um ótimo serviço de hospedagem (conforme citado no RN5) irão proporcionar atingir no mínimo o tempo médio entre falhas citado.

- RN6: Tempo Médio para Reparo (MTTR) - O Tempo Médio para Reparo (MTTR) deve ser de no máximo 1 hora.

Devido ao conceito de módulos, os sistemas web desenvolvidos a partir do framework Drupal possuem ótima manutenibilidade. Além disso, os reparos poderão ser testados e implementados em ambiente *off-line* para depois irem para o ambiente *online*. Dessa forma será possível atingir o tempo médio para reparo citado.

## Desempenho

- RN7: Capacidade - A expectativa é que cada *live stream* terá, em média, 50 participantes (Caso de Uso 3.1.6 Assistir *live stream*). Assim, o sistema deve acomodar o acesso simultâneo desse número de usuários.

Os sistemas web desenvolvidos a partir do framework Drupal suportam tranquilamente o acesso simultâneo do número de participantes citado.

## Suportabilidade

- RN8: Sobreposições de CSS ou *Javascript* - Qualquer sobreposição de CSS ou *Javascript* ao padrão *Bootstrap* 3 deve ser realizada (e comentada) em um único arquivo ou conjunto de arquivos classificados por partes do sistema (por exemplo topo, rodapé, busca, etc.). Dessa forma a manutenibilidade é facilitada pois qualquer membro da equipe de desenvolvimento saberá onde fazer manutenções no código.

Utilizar o módulo ‘Asset Injector’.

## Segurança

- RN9: *Backup* - O sistema web deve fazer *backups* automáticos, todo dia, mantendo os últimos 7 (sete). Os *backups* devem ser completos, ou seja, incluírem a base de dados e todos os arquivos referentes ao sistema.

Utilizar o módulo ‘Backup and Migrate’.

- RN10: *Log* de alterações - O sistema web deve registrar em um *log* todas as alterações (inclusão, edição e exclusão) feitas nos conteúdos do sistema, identificando o usuário, o tipo de conteúdo alterado, a data e hora da alteração e o tipo de operação.

Utilizar o módulo *View* para criar uma *view* de *logs* e disponibilizar, via interface, um *link* para acesso aos registros.

## Portabilidade

- RN11: Navegadores - O sistema web deve ser acessível pelo menos nos seguintes navegadores, a partir das respectivas versões: Google Chrome (60.0.3112.90), Mozilla Firefox (52.0.2), Internet Explorer (11), Opera (65.0.3467.72), Microsoft Edge (44.18362) e Safari (12.1.2).

O *framework* CSS *Bootstrap* 3 suporta todas as versões de navegadores citadas. Desta forma o sistema web será visualizado de forma correta em todos eles.

## Outros Requisitos Tecnológicos

- RN12: Responsividade - O sistema web deve ser responsivo, ou seja, ser acessível por *desktops*, *smartphones* e *tablets*, com visualização personalizada para as diversas resoluções dos dispositivos.

O *framework* CSS *Bootstrap* 3 tem como premissa o desenvolvimento de sistemas web responsivos e *mobile-first*. Dessa forma, cada dispositivo citado terá uma visualização personalizada, adequada a sua resolução.

# Definição de *Framework* do Projeto

## 2.1 *Framework* de *back-end*

O sistema web deve ser desenvolvido no *framework* Drupal versão 8.8.4 (versão mais atualizada em 25/03/2020). Por política de desenvolvimento, a empresa utiliza esse framework de *back-end* na criação de todas as suas aplicações web. Dessa forma a equipe já é treinada e o desenvolvimento é otimizado, além de manutenções e atualizações. Além disso, o Drupal é mantido por uma comunidade *open source* ativa, proporcionando uma ferramenta segura, estável e robusta.

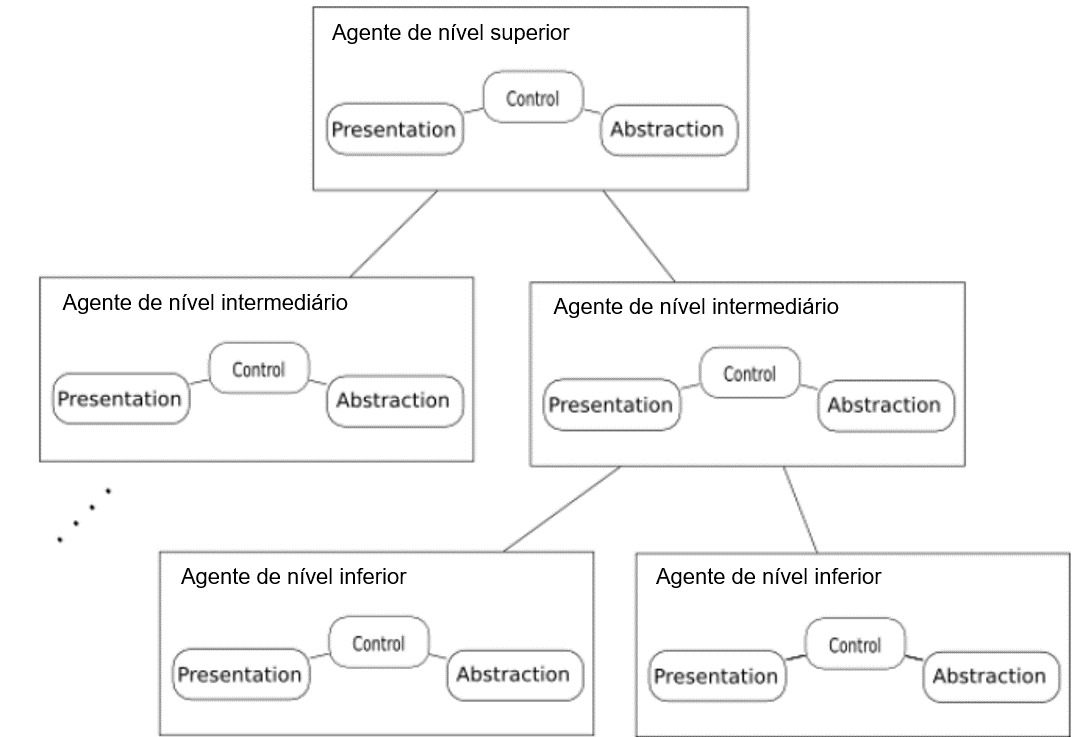
## 2.2 *Framework* de *front-end*

Para o desenvolvimento *front-end* do sistema web, o *framework* CSS Bootstrap 3 deve ser utilizado. Por política de desenvolvimento, a empresa utiliza esse framework na criação de todas as suas aplicações web. Dessa forma a equipe já é treinada e o desenvolvimento é otimizado, além de manutenções e atualizações. Além disso, o *Bootstrap* é um dos projetos mais bem avaliado no site GitHub, com mais de 111.600 estrelas e 51.500 forks, possui ampla documentação na página oficial e é um dos *frameworks* *front-end* mais utilizados no mundo.

# Definição de Padrões Arquiteturais de Projeto

## 3.1 Camadas da aplicação

Será seguido o padrão arquitetural PAC (Presentation-Abstraction-Control). O PAC tem uma estrutura que se repete várias vezes dentro da aplicação, agrupada em *Agent* ou *Triad*. Cada um destes agentes tem uma estrutura composta por um *Presenter* (P), por uma *Abstraction* (A), por um *Controller* (C) e podem ser tão simples quanto um objeto ou, tão complexos quanto todo um software. A estrutura é ilustrada a seguir.



# Visão de Implantação da Solução

