**Documento de Arquitetura de Software**

**Zeitgeist**

**Autores:**

**Alex Lazzarotti - RA: 11420764**

**Carina Roriz Pimenta – RA: 11410938**

**João Carlos Sanglard Heringer – RA: 11320524**

**Marcelino Pimenta Bello Lisboa – RA: 11411311**

**Vivian Henrique Silva - RA: 11410827**

**Vitor Alessandro - RA: 11411938**

**Belo Horizonte**

**Setembro/2017**

## **Objetivos de negócio**

A arquitetura escolhida tem como objetivo os seguintes itens:

* Boa usabilidade para montagem do jornal
* Simplicidade tecnológica para que o time de desenvolvimento tenha o mínimo de impedimentos e máxima produtividade
* Controle de acesso para garantir a definição de papéis no processo de montagem do jornal
* Facilidade de leitura e entendimento do jornal em aplicativos móveis de diversos tamanhos e diferentes sistemas operacionais

## **Visualização de Negócio**

Servidor de Aplicação

Servidor de Banco de Dados

Aplicação Mobile

Aplicação Web - Single Page Application (SPA)

Log

Entries

Apache

MySQL

HTTP

HTTP

Web Service

Aplicativo Híbrido

Android/IOS

(Ionic)

PHP e

AngularJS

## 

## 

## **Condutores Arquiteturais**

Os condutores arquiteturais do projeto Zeitgeist são:

* Suportabilidade: o sistema deverá ser acessado pelos principais sistemas operacionais móveis (Android e IOS);
* Autenticação: para acessar o gerenciador de conteúdo do jornal é necessário realizar login com usuário e senha cadastrados anteriormente;

● Autorização: cada usuário do gerenciador de conteúdo do jornal possui um determinado perfil que define quais funcionalidades do sistema ele poderá acessar;

● Usabilidade: o gerenciador de conteúdo deve permitir a criação de uma nova edição do jornal de forma simples e intuitiva e a versão mobile do jornal deverá ser intuitiva e de fácil manuseio, permitindo que o usuário consiga ler a edição atual, acessar edições anteriores e enviar um e-mail de sugestão ou reclamação com o menor número de toques na tela possível;

● Aplicativo mobile por restrição do cliente;

## **Estilo Arquitetural**

O projeto é composto principalmente pelo padrão modelo-visão-controlador (MVC).

Foi criada com MVC uma API que gerencia e controla as requisições enviadas via URL (REST), essa API foi desenvolvida com a linguagem de programação PHP e é utilizada pelo gerenciador de conteúdo desenvolvido com HTML, Materialize e AngularJS e também pelo aplicativo mobile híbrido desenvolvido com o framework Ionic 1 que utiliza o AngularJS como meio de desenvolvimento.

## **Plataforma Arquitetural**

Banco de dados MySQL e linguagem de programação PHP.

## **Decisões Arquiteturais**

|  |  |  |
| --- | --- | --- |
| ***Condutor Arquitetural*** | ***Mecanismo Arquitetural*** | ***Justificativa*** |
| Suportabilidade | Ionic | O framework possibilita a criação de aplicativos híbridos que podem ser compilados para Android e IOS de maneira rápida. |
| Autenticação | Tela de login | A tela de login com usuário e senha salvos em banco de dados foi a solução mais simples e viável para facilitar e automatizar o processo de autenticação do usuário no sistema. |
|  | Senha criptografada em banco | Foi escolhida a criptografia SHA2 para salvar as senhas no banco de dados, para garantir de forma segura que elas não serão expostas. |
|  | Dados salvos na sessão | Os dados do usuário em sessão facilitam a verificação do usuário logado e permitindo que nenhum acesso não autenticado seja permitido. |
| Autorização | Definição de perfis | Foi escolhido criar uma estrutura de perfis para definir os níveis de autorização dos usuários no sistema gerenciador de conteúdo. |
| Usabilidade | Materialize | O framework Materialize será utilizado para o gerenciador de conteúdo e este possibilita a criação de templates intuitivos com boa usabilidade para o usuário. |
|  | Ionic | Serão desenvolvidos layouts para o aplicativo mobile com a ajuda dos componentes e temas do Ionic, pensando sempre na facilidade de utilização dos usuários. |

## **Modelagem Arquitetural**

***Diagrama de Pacotes:***

Usuário

Jornal

Página

Dados Página

PHP e AngularJS

Imagem Páginas

Template

Dados Template

***Diagrama de Componentes:***

Banco de Dados

(MySQL)

Servidor Apache

(PHP)

Aplicativo Mobile

(Android/IOS)

Navegador Web

(HTML e AngularJS)

Log Entries

***Diagrama de implantação:***

Servidor de Aplicação e Banco de Dados

Navegador Web

Dispositivo Móvel

(Android ou IOS)

Páginas HTML com Materialize

MySQL Server

API (PHP)

Aplicativo Instalado

Log Entries

## **Gestão de Riscos Arquiteturais**

|  |  |  |
| --- | --- | --- |
| **# do Risco** | **Impacto** | **Descrição do Risco** |
| 1 | Alto | Falta de conhecimento da equipe sobre o AngularJS |
| 2 | Alto | Falta de conhecimento sobre o Ionic |
| 3 | Alto | Realização de testes da aplicação |
| 4 | Médio | Equipamentos e ambiente para testes |
| 5 | Médio | Realização de testes de usabilidade |
| 6 | Alto | Definição do servidor de hospedagem |

|  |  |
| --- | --- |
|  | **Plano de Mitigação e Contingência do Risco 1** |
|  |  |
| **Ação #1** | **O quê:** *Fazer cursos online e ler sobre o assunto, além de tirar suas dúvidas antes de iniciar o desenvolvimento* |
| **Quem:***Toda a equipe* |
| **Quando:** *Nas duas primeiras semanas de Setembro de 2017.* |
| **Onde:** *Nas dependências próprias.* |
| **Por quê:** *Para facilitar o processo de desenvolvimento do projeto.* |
| **Como:** *Fazer cursos online;*  *Ler sobre AngularJS*  *Estudar exemplos de aplicações;*  *Fazer exemplos de código* |
| **Custo:** *Variável, aproximadamente 20 horas* |

|  |  |
| --- | --- |
|  | **Plano de Mitigação e Contingência do Risco 2** |
|  |  |
| **Ação #1** | **O quê:** *Fazer cursos online e ler sobre o assunto, além de tirar suas dúvidas antes de iniciar o desenvolvimento* |
| **Quem:***Toda a equipe* |
| **Quando:** *Nas 3 primeiras semanas de Setembro de 2017* |
| **Onde:** *Nas dependências próprias.* |
| **Por quê:** *Para facilitar o processo de desenvolvimento do projeto.* |
| **Como:** *Fazer cursos online;*  *Ler sobre Ionic;*  *Estudar exemplos de aplicações;*  *Fazer exemplos de código;* |
| **Custo:** *Variável, aproximadamente 20 horas.* |

|  |  |
| --- | --- |
|  | **Plano de Mitigação e Contingência do Risco 3** |
|  |  |
| **Ação #1** | **O quê:** *Estudar cronograma e casos de teste para a aplicação* |
| **Quem:***André, Ater e Napoliana.* |
| **Quando:** *Na terceira semana de Setembro de 2017.* |
| **Onde:** *Nas dependências próprias.* |
| **Por quê:** *Cobrir todos os possíveis bugs do sistema para garantir o menor número de problemas possíveis.* |
| **Como:** *Estudar as telas, funcionalidade e requisitos do sistemas para montar os casos de teste.* |
| **Custo:** *Aproximadamente 15 horas.* |

|  |  |
| --- | --- |
|  | **Plano de Mitigação e Contingência do Risco 4** |
|  |  |
| **Ação #1** | **O quê:** *Selecionar quais os equipamentos e como será montado o ambiente de teste.* |
| **Quem:***André, Ater e Napoliana.* |
| **Quando:** *Na terceira semana de Setembro de 2017.* |
| **Onde:** *Nas dependências próprias.* |
| **Por quê:** *Para garantir que os testes sejam realizados em ambientes idênticos ou semelhantes ao cenário real.* |
| **Como:** *Selecionar os celulares e computadores que serão utilizados para teste e configurar o servidor de teste com a aplicação e banco de dados de homologação* |
| **Custo:** *Aproximadamente 10 horas.* |

|  |  |
| --- | --- |
|  | **Plano de Mitigação e Contingência do Risco 5** |
|  |  |
| **Ação #1** | **O quê:** *Selecionar funcionários que avaliarão a usabilidade das telas* |
| **Quem:***André, Ater e Napoliana.* |
| **Quando:** *ao final de cada Sprint.* |
| **Onde:** *Nas dependências próprias.* |
| **Por quê:** *Para garantir que o sistema tenha boa usabilidade e verificar se ainda possui implementações pendentes.* |
| **Como:** *Verificar junto ao cliente em uma visita técnica quem são os funcionários que vão utilizar o sistema e possuem capacidade para testar as principais funções do sistema.* |
| **Custo:** *Aproximadamente 16 horas* |

|  |  |
| --- | --- |
|  | **Plano de Mitigação e Contingência do Risco 6** |
|  |  |
| **Ação #1** | **O quê:** *Verificar, contratar e configurar o servidor de hospedagem para o projeto* |
| **Quem:***Tarley* |
| **Quando:** *Na primeira semana de Setembro de 2017.* |
| **Onde:** *Nas dependências próprias.* |
| **Por quê:** *Para garantir que o sistema seja acessado remotamente para testes e para entrar em produção* |
| **Como:** *Verificar quais são as opções de servidores disponíveis e viáveis para o projeto em termos técnicos e financeiros* |
| **Custo:** *Aproximadamente 8 horas* |