Alocação de recursos computacionais a turmas de disciplinas

Software Architecture Document

# Introduction

# Architecture Goals and Constraints

Com o objetivo do projeto já definido, foram especificados casos de arquitetura, afins de manter o foco em um projeto escalável, de fácil manutenção, confiável e com necessidade de pouca banda de dados (client-side). As ferramentas utilizadas devem abranger o conhecimento do arquiteto ou não divergir muito de seu horizonte de conhecimento, para não haver um trabalho muito árduo em pesquisas. Com o objetivo de possuir um banco de dados o SQL server e ASPNET MVC 5 e .Net 4.5.

# Architectural Style/ Patterns

Algumas definições sugeridas e abordadas pelo arquiteto seria a utilização de MVC 5 do AspNet e AngularJS para saciar o desejo de pouco banda de dados entre servidor e cliente.

Para concretização do framework de front-end (camada de visualização) seria o bootstrap, Html5 e CSS3.

A aplicação deverá ter a usabilidade de ser acessada por qualquer dispositivo com acesso a web, facilitando o acesso do usuário com estilos de design totalmente responsivos para cada dispositivo, tanto quanto o uso de mobilidade.

# Guiding Principles

Somente ferramentas “Open-Source” poderão ser utilizadas, afim do projeto não ficar dependente de licenças ou manutenções depreciadas. As conexões com o banco serão feitas através de procedures para minimizar a resposta do banco de dados.

# Analysis Results

A utilidade do software necessita alguns observações em alguns casos críticos, como por exemplo: A segurança da autenticação e da permissão dos usuários, que foi saciada pelo sistema de login de AspNET; A flexibilidade para a utilidade de dispositivos móveis para a utilização do software através de qualquer dispositivo, que foi resolvida pelo bootstrap e o Html5; A escalabilidade para a inserção de novos módulos ou funções que foi provinda do MVC AspNET e do AngularJS; A latência do projeto que pode ser minimizada por procedures no SQL Server e através do AngularJS que minimiza as requisições para o servidor, reduzindo o tempo de resposta e o número de requisições para o servidor.

# Mapping Architecture to Requirements

## Significant Driving Requirements

| **Requirements** | **Architecture Solution / Notes** |
| --- | --- |
| Baixa Latência | Com a utilização do AngularJS houve uma minimização na latência do projeto, character |
| Segurança e permissão | A questão de segurança e permissão de usuários foi resolvida pelo Sistema de autenticação do AspNET. |
| Usabilidade multiPlataformas | Para satisfazer esse caso foi utilizado HTML5 e o framework de design do bootstrap, tornando o aplicativo responsive. |
| Aplicação Escalável | O MVC 5 do AspNET possuí características nativas para saciar esse desejo, facilitando a manutenção e seu crescimento. Além disso o SGBD do SQL server mantêm o banco totalmente escalável. |

## Use-Case Realizations

A Aplicação auxilia profissionais da área docente a administrar seus pedidos e alocações de recursos computacionais, podendo realizar pedidos de recursos a qualquer momento, facilitando a interação com a instituição a qual trabalha. O prazo de solicitação de recursos é estimado por um prazo pré-estabelecido pela instituição responsável, onde profissionais poderão solicitar recursos computacionais desde que atendam seus critérios como horário de aula por exemplo. Ao fim do prazo será executado um algoritmo de balanceamento de recursos, sendo o mais justo possível com os professores de acordo com sua carga horária. Após a finalização os usuários poderão solicitar mudanças, a qual solicitarão diretamente ao dono do recurso alocado.

# Relations among views

| **Element in View X** | **Element in View Y** |
| --- | --- |
|  |  |

# Analysis Results

As respostas do Sistema não retornadas em um tempo muito curto, utilizando uma banda de 3G, devido a alguns indicadores, como a quantidade de registros é muito baixa e a validação e a montagem de telas são criadas no lado do cliente, reduzindo a manipulação do servidor.