Acadsystem

Projeto Arquitetural

There is guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute that allows you to toggle whether it is visible or hidden in this template. Use the Microsoft® Word® menu **Tools > Options > View > Hidden Text** check box to toggle this setting. There is also an option for printing: **Tools > Options > Print**.

# Objetivo

Este documento tem o objetivo de descrever filosofia, decisões, restrições, justificativas, premissas e quaisquer outros aspectos que dão forma ao projeto e sua implementação.

# Metas Arquiteturais e Filosofia

O principal direcionador da escolha foi a necessidade por usabilidade e fornecimento de informações fieis os resultados obtidos pelos usuários em suas diversas atividades na academia. Possuindo fácil acesso ao sistema, em dispositivos comumente utilizado pelos usuários, como smartfones e desktops.

O banco de dados utilizado será o Mysql, trazendo facilidade de manutenção e de utilização, por parte dos desenvolvedores, acelerando o processo de produção.

A linguagem de programação também possui como qualidade a facilidade de uso, integridade e segurança.

# Premissas e Dependências

# Requisitos Críticos da Arquitetura

<http://php.net/>

http://www.scriptcase.com.br/

# Decisões, Restrições e Justificativas

* **Uma página web:** Para acesso ao sistema, onde será possível acessa-lo de locais diferentes bastando ter internet e configurações compatíveis de sistema operacional e browser.
* **Utilizar o Mysql para administração da persistência**: Este SGBD funciona em mais de 20 plataformas proporcionando consistência, alta performance confiabilidade e fácil utilização, além de flexibilidade e controle.
* **Linguagem de programação:** PHP é a plataforma escolhida, por apresentar o paradigma adequado a solução, além da facilidade de uso e de integração com outras tecnologias.

# Mecanismos Arquiteturais



* **Aplicação**:

Através da ferramenta Scriptcase é possível criar sistemas de informática projetados para utilização através de navegador, internet ou aplicativos desenvolvidos utilizando tecnologias web tais como: HTML, Javascritpt, CSS. Suas aplicações podem ser executadas a partir de um servidor HTTP ou localmente, no dispositivo do usuário, a linguagem back-end utilizada pela ferramenta é o PHP.

* **Aplicações do Framework**:

1. Menu - As aplicações de menu servem para criar uma estrutura hierárquica de navegação das aplicações do sistema. É um recurso extremamente útil, uma vez que organiza as aplicações de modo visual, facilitando o acesso do usuário às aplicações.

1. Formulário - São as aplicações para atualização de Dados. O Scriptcase cria uma aplicação com todos os recursos de inclusão, alteração, exclusão e navegação de registros.
2. Consulta - São as aplicações para visualização dos dados. Uma consulta no Scriptcase tem também o caráter de relatório, uma vez que gera a saída em PDF, XLS, XML, RTF e outros formatos.
3. Calendário - Permite geração de Aplicações Calendário. Ótimo para criação de aplicações de agendamento e compromisso.
4. Dash Board - Permite exibir vários tipos de aplicações aninhadas em Widgets semelhantes àquelas para o iGoogle.
5. Aba - O Scriptcase permite através desta opção reunir várias aplicações já desenvolvidas e dar à aplicação resultante um destes formatos ou ambos, onde cada aba ou cada opção da pasta corresponde a uma aplicação.
6. Filtro - Formulário de filtro para ligar alguma aplicação que necessite receber uma string contendo uma cláusula where
7. Controle - São as aplicações criadas para infraestrutura do sistema. Podem ser utilizadas para captação de dados (parâmetros) da interface para integra-los com a camada modelo.   
   Ex.: login de sistema, seleção da empresa, seleção de acesso e etc.
8. PDF -  Scriptcase oferece uma ferramenta para gerar relatórios de qualidade, usando formatos pré-definidos.  
   Use esta aplicação para gerar relatórios para as instituições de governo ou outra organização que formato exigido específicos.

* **Bibliotecas** **Externas**:

1. Bootstrap - é um framework front-end que facilita a vida dos desenvolvedores web a criar sites com tecnologia mobile (responsivo) sem ter que digitar uma linha de CSS. O termo “Bootstrap” em inglês significa “inicialização”, algo que possui um ponto de partida.
2. JQuery - é uma biblioteca de código aberto e possui licença dual, fazendo uso da Licença MIT ou da GNU General Public License versão 2. A sintaxe do jQuery foi desenvolvida para tornar mais simples a navegação do documento HTML, a seleção de elementos DOM, criar animações, manipular eventos e desenvolver aplicações AJAX.
3. Angular JS - Na execução do Angular.js, ocorre o carregamento, onde sua árvore DOM e Javascritpt é transformada em um aplicativo angular. Isso ocorre, pois, o código HTML com caracteres especiais (diretivas e filtrosangular) é compilado e angular realiza uma vinculação entre Controller, Model e View (padrão MVC).

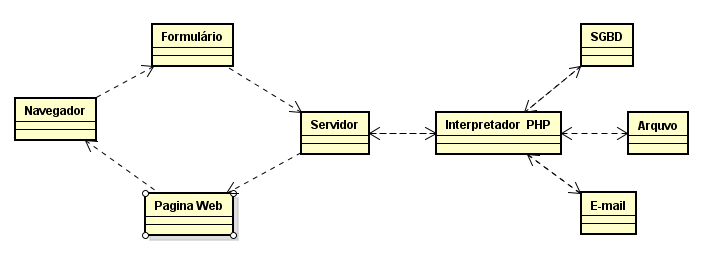
* **Plataforma PHP**:

O PHP um acrônimo recursivo para PHP: Hypertext Preprocessor é uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML. O PHP é considerado uma linguagem de back end pois funciona no servidor de aplicações. Ao baixar o Scriptcase o PHP está incluso.

* **Servidor** **Apache**:

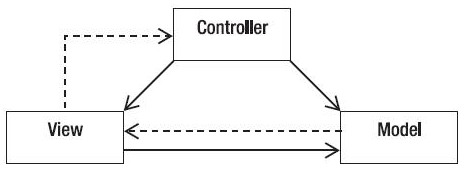
O servidorApache (ou Servidor HTTP Apache, em inglês: Apache HTTP Server, ou simplesmente: Apache) é o mais bem-sucedido servidor web livre. Foi criado em 1995 por Rob McCool, então funcionário do NCSA (National Center for Supercomputem Applications). O servidor é compatível com o protocolo HTTP versão 1.1. Suas funcionalidades são mantidas através de uma estrutura de módulos, permitindo inclusive que o usuário escreva seus próprios módulos, utilizando a API do software, ao baixar o Scriptcase o Apache está incluso.

# Principais Abstrações



# Camadas do Framework da Arquitetura

* **Arquitetura MVC:**

****

Uma arquitetura que permite dividir o desenvolvimento da aplicação em três camadas conceituais: Model, View e Controller (Modelo - Visão - Controlador).

1. O **model** seria a parte da modelagem de dados e regras de negócio. É nela que vão constar as classes, as consultas ao banco de dados e as regras de negócio do sistema.
2. **View** é camada responsável pela interação entre sistema e usuário. Tendo como responsabilidade: manipular os dados para e apenas exibi-los.
3. O **Controller** responsável por controlar todo o fluxo de informação que passa pelo sistema. Define quais informações devem ser geradas, quais regras devem ser acionadas e para onde as informações devem ir, é na controladora que essas operações devem ser executadas. Em resumo, é a controladora que executa uma regra de negócio (modelo) e repassa a informação para a visualização (visão).

# Visões Arquiteturais

• Lógica:

•operacional:

• Caso de uso:

