Academic Advisor

Architecture Notebook

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**.

# Purpose

Este documento visa demonstrar principais aspectos do projeto Academic Advisor. Sua arquitetura, forma de implementação e aspectos.

# Architectural goals and philosophy

O sistema Academic Advisor prover integração de um serviço web e mobile para atender necessidades de alunos e futuro-alunos em universidades. Provendo um contato inicial entre estre essas duas entidades, para que troquem informações sobre o curso que se tem em mente.

O sistema funciona somente com conexão à internet. Podendo usar uma rede privada no caso da universidade. Embora esses dados devam ser disponibilizados publicamente para acesso de usuários fora desta rede.

Primeira versão do sistema mobile vai ser desenvolvi Android, amplamente divulgada e acessível na maioria dos smartphones. Isso pode limitar acesso inicialmente para alguns interessados.

As demandas iniciais poderão ser atendidas sem grandes problemas. Primeiras versões usaram o mínimo de funcionalidades para atender o público. O feedback de usuários será muito importante para aperfeiçoamento do sistema.

Versão mobile pretende fornecer funcionalidades básica como consulta de agenda, cursos, por exemplo.

Versão web vai fornecer formulários para cadastro, agenda e relatórios em forma de lista.

Sistema web vai ser projetado para funcionar em navegadores web, principais funcionalidades de cadastro estarão nele.

Futuramente pode-se ocorrer problemas com integrações de serviços, como por exemplo consultar serviços públicos de universidade para gerar cadastros dinâmicos.

# Assumptions and dependencies

Assume-se que a universidade vai fornecer dados para carga no sistema. E que ela desponha de infraestrutura de redes que permita comunicação entre outras máquinas. Qualquer problema com múltiplos acessos é de responsabilidade do cliente.

Como o Sistema vai possuir uma versão web browser e outra web mobile. Será necessário manter coerência no design de interface.

Ponto crítico será a sincronia de dados. Pois a base será de responsabilidade da universidade. Assim como cadastros. Caso está não atualize ou disponibilize. Vai gerar conflitos. Como exemplo temos a mudança de currículo de um curso ou extinção do mesmo.

Atualmente o projeto possui apenas um desenvolvedor previsto. Arquiteto também participará no desenvolvimento de integrações e análise de domínio.

# Architecturally significant requirements

Algumas outras funcionalidade

Notificações via e-mail

Agenda atualizada

Reutilização de classes e dll’s

Navegação simples e objetiva

Fácil adaptação e customização para outros clientes

# Decisions, constraints, and justifications

Perfil de administrador não deve efetuar cadastro via Sistema mobile

Sistema mobile deve permitir apenas ações simples, como efetuar autenticação no sistema; Cancelar Agendamento; consultar listas de cursos;

O back-end do sistema deve ser reutilizável tanto na versão mobile e web.

Usar tecnológica fron-end que use design responsivo para utilização em qualquer dispositivo.

Não é necessário alta perfonamance de sistema e múltiplos servidores ou espelhamento de bases.

Usar framework opensource e que suas comunidades atualizarem constantemente, com amplo suporte destes

Usar o paradigma orientados a objetos para o desenvolvimento

Não é uma rede-social, não terá chat.

# Architectural Mechanisms

Mecanismo de notificação de modificação de eventos no sistema (e-mail)

Sistema mobile deve usar pouca memória.

Consultas a base de dados devem ser limitadas com no máximo 10 registros.

Pelo menos dois filtros devem ser aplicados a consultas ao banco.

# Key abstractions

* Multiplas camadas
* Sistema web
* Sistema mobile
* Language java
* Tecnologia android

**Frameworks e pacotes**

* Fresco - para carregar imagens
* Crashlytics - crash support
* Android Data Binding
* Retrofit - client-server framework
* Realm - para database
* ReactiveX - backend e frontend

# Layers or architectural framework

MVC – Model View Controller – com reutilização das camadas de Controller e Model

Desenvolvimento divido em front-end e back-end

Utilizar bibliotecas javas para prover serviços, como envio de email

# Architectural views

Na visão arquitetural teremos duas formas de acesso ao sistema.

Web – Onde o administrador terá como dar manutenção no sistema e administrar seus cadastros. Assim como alunos poderão preencher seus relatórios e agendas.

Mobile – visão mais simples do sistema onde um administrador não terá acesso. Apenas perfil de usuário do tipo aluno poderá ver conteúdo.

Alguns casos de uso relevantes para o sistema:

UC02 Verificar Agenda

UC09 Cadastrar Roteiro do Curso

UC11 Preencher Relatório

UC07 Procurar Veteranos disponíveis em certo curso