UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

# Decisão das tecnologias

## 1. O Software a ser desenvolvido

Em poucos detalhes, o que será desenvolvido neste projeto serão dois componentes, o primeiro denominado SDD-Scheduler, que consiste em um sistema que gerencia toda distribuição das disciplinas, a nível de back-end e que possui uma interface de comunicação utilizando o padrão RESTFUL. O segundo componente é o SDD-WEB, uma interface web que interage com o SDD-Scheduler e realiza as operações no sistema de forma amigável.

## 2. Estudo realizado

Há inúmeras soluções e framework disponíveis no mercado e nas mais diversas linguagens de programação que possuem como objetivo auxiliar e atuando nas características dos softwares que temos que produzir, algumas opções encontradas serão listadas a seguir.

### 2.1 SDD-Scheduler

Para o SDD-Scheduler é necessário desenvolver um software que interaja com o banco de dados da aplicação e realize operações como CRUD (Criação, leitura, modificação e remoção) nos modelos de dados propostos.

Também se espera que o SDD-Scheduler possua uma interface de comunicação Webservice utilizando o protocolo e padrão RESTFUL, deste modo focamos o estudo em linguagens e até mesmo frameworks que nos ajudem a ter uma boa produtividade para alcançarmos estes objetivos, dentre eles:

|  |  |  |  |
| --- | --- | --- | --- |
| **Linguagem** | **Framework?** | **Prós** | **Contras** |
| PHP | CakePHP | ➜ Comunidade grande.  ➜ Documentação completa e disponível em vários idiomas.  ➜ Alta produtividade.  ➜ Fácil extensão para se desenvolver API RESTFUL.  ➜ Geração automática de parte do código. | ➜ Não suporta thread (não se sabe se será necessário)  ➜ Da equipe, cerca de 40% apenas possui boa experiência com a linguagem.  ➜ Apenas 1 dos integrantes da equipe possui experiência com o framework cakePHP |
| Ruby | Rails | ➜ Comunidade grande.  ➜ Boa documentação da linguagem.  ➜ Alta produtividade.  ➜ Geração automática de parte do código.  ➜ Fácil extensão para se desenvolver em API RESTFUL | ➜ Menos de 20% da equipe possui experiência com a linguagem.  ➜ Menos de 10% da equipe possui experiência com o framework. |
| Python | Django | ➜ Comunidade grande.  ➜ Boa documentação da linguagem.  ➜ Alta produtividade.  ➜ Geração automática de parte do código.  ➜ Fácil extensão para se desenvolver em API RESTFUL | ➜ Cerca de 50% da equipe possui experiência com a linguagem.  ➜ Menos de 10% da equipe possui experiência com o framework. |
| Grails | Não | ➜ Comunidade parcialmente grande.  ➜ Mais de 90% dos integrantes da equipe possuem conhecimento em Java e em Grails pode se programar utilizando Java  ➜ Utilização do JUnit que já é natural para os integrantes do grupo para criação de testes unitários. | ➜ Baixa produtividade comparado aos outros frameorks e linguagens citados acima.  ➜ Para desenvolvimento de um webservice RESTFUL, muita coisa terá de ser feita na mão utilizando o padrão Jetty.  ➜ Apenas 1 integrante da equipe possui experiência com Grails. |
| Java | Dropwizzard | ➜ Mais de 90% dos integrantes da equipe possuem conhecimento em Java.  ➜ Utilização do JUnit que já é natural para os integrantes do grupo para criação de testes unitários.  ➜ Framework já especializado e integrado com o padrão Jetty do java para criação de APIs webservice, inclusive utilizando padrão RESTFUL | ➜ Menos de 20% da equipe possui experiência com o framework.  ➜ Desenvolvimento será mais lento pois em Java é necessário fazer muita coisa manual e de forma mais complexa que em outras linguagens fracamente tipadas como PHP, Python e Ruby. |

### 2.2 SDD-WEB

Para o SDD-WEB é necessário produzir uma interface web que consiga se comunicar com o SDD-Scheduler consumindo seus dados e realizando operações de acordo com o contrato que ele fornece.

Espera-se que a ferramenta e a linguagem escolhida possua facilidade para enviar requisições webservices baseada no protocolo RESTFUL para que esta integração possa ocorrer de forma mais tranquila.

Naturalmente, por se tratar de uma interface WEB, se espera uma ferramenta e/ou linguagem que forneça uma boa base para se trabalhar com aplicações WEB e até mesmo uma fácil integração com o bootstrap framework para que a aplicação seja responsiva.

Abaixo as ferramentas e tecnologias levantadas para serem utilizadas no desenvolvimento do SDD-WEB:

|  |  |  |  |
| --- | --- | --- | --- |
| **Linguagem** | **Framework?** | **Prós** | **Contras** |
| JavaScript | AngularJS | ➜ Tecnologia em ascenção no mercado WEB.  ➜ Comunidade grande.  ➜ Possui com Jquery uma gama poderosa de efeitos para aplicações WEB.  ➜ Pode manipular chamadas síncronas e assíncronas. | ➜ Cerca de 30% da equipe possui experiência com a linguagem e o framework. |
| PHP | Laravel ou CakePHP | ➜ Comunidade grande.  ➜ Documentação completa e disponível em vários idiomas.  ➜ Alta produtividade.  ➜ Modelo MVC implementado nativamente.  ➜ Fácil de se integrar com bootstrap, Jquery, etc. | ➜ Não existem ORMs nativos programados para chamadas em webservices, de modo que tudo deveria ser feito manualmente.  ➜ Não há a possibilidade de se utilizar o conceito de Threads que seria interessante para executar tasks em background para envio de e-mails, por exemplo. |

## 3. Tomada de decisão

Para levantar as tecnologias e escolher qual delas que deveríamos seguir no projeto, foi feita uma única reunião no dia 16/10/2015, cuja ata está disponível no repositório do projeto, onde membros das equipes de arquitetura e desenvolvimento se reuniram e começaram a levantar as ferramentas. Após o levantamento das possíveis tecnologias, iniciou-se uma discussão para se chegar aos prós e contras de cada uma.

Detalhes da reunião podem ser encontrados na respectiva ata. Abaixo segue um resumo do que foi decidido para cada componente:

### 3.1 SDD-Scheduler

Todos os membros do projeto que participaram da reunião tinham suas opiniões sobre qual tecnologia utilizar neste componente. Apesar de tecnologias que utilizam linguagens de mais alto nível e fracamente tipadas como PHP, Python e Ruby serem mais recomendadas e os líderes terem chegado a conclusão de que elas deviam ser usadas pois agilizariam o desenvolvimento, foi escolhido utilizar o Dropwizzard, um framework em Java, pois todos os membros da equipe e até mesmo do projeto possuem alguma experiência com a linguagem. Outro ponto forte para esta escolha é que facilitaria para a equipe de Testes criarem os testes, pois eles também possuem maior experiência com ferramentas focadas na tecnologia Java.

### 3.2 SDD-WEB

A maior parte dos membros da equipe de desenvolvimento front-end se sentiram confiantes em produzir uma aplicação WEB utilizando o AngularJS. Dado que ninguém se opôs a ideia e que a equipe se comprometeu em realizar suas atividades no tempo previsto, foi definido que o SDD-WEB seria desenvolvido utilizando AngularJS.

### 3.3 Outras ferramentas e tecnologias

Outras ferramentas e tecnologias deverão ser utilizadas no contexto do projeto para armazenamento de informações dos grupos, criação de protótipos e diagramas. Neste contexto, assumiu-se que as equipes teriam total autonomia para escolherem as ferramentas para criação de tais artefatos, entretanto, para armazenamento deve ser seguido o que a Gerência de Configuração mandar.