



Universidade Estadual de Maringá
Departamento de Informática



Aluno: Anderson de Souza Zanichelli
Orientadora: Profa. Dra. Luciana Andréia Fondazzi Martimiano

Redes de Computadores

**Uso de informações de contexto para controle de acesso em provedores
de conteúdo**

Primeiro relatório técnico do Trabalho de Conclusão de Curso

Maringá
2015

1. Atividades Desenvolvidas

Na entrega da proposta do Trabalho de Conclusão de Curso havia sido definido o cronograma de atividades a serem realizadas como no quadro abaixo:

	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov
Elaboração e entrega da Proposta de TCC								
Estudo sobre AngularJs e Ionic								
Elaboração e entrega do 1º relatório técnico								
Estudo sobre NodeJs e MongoDB								
Modelagem da arquitetura do sistema								
Implementação dos sistemas								
Elaboração e entrega do 2º relatório técnico								
Testes								
Conclusão e entrega da monografia do TCC								
Defesa pública do TCC								

Quadro 1 – Cronograma de Atividades.

Apesar da universidade ter entrado em greve, consegui dar continuidade a grande parte das tarefas no desenvolvimento do trabalho de conclusão de curso.

Os meses iniciais foram dedicados ao estudo das tecnologias que seriam utilizadas no desenvolvimento da aplicação para o dispositivo móvel e da aplicação servidora. O estudo de AngularJs se deu através de livros, video aulas, sites e a documentação do framework. Já o estudo do framework Ionic foi somente através da documentação existente no site do próprio Ionic. Para a aplicação servidora foi estudado NodeJS através de livros e a documentação no site. NodeJS envolve o uso de Javascript no lado do servidor.

Um dos principais motivos para a escolha do NodeJs foi a vantagem de utilizar somente a linguagem Javascript, tanto no lado do cliente como no do servidor, o que facilita muito a troca de informações entre a aplicação cliente e a aplicação servidora. A comunicação entre as aplicações se dão através do uso de JSON (JavaScript Object Notation) que são objetos com uma notação simplificada que pode até ser lido por uma pessoa, por exemplo, uma pessoa poderia ser representada da seguinte forma:

```
pessoa:{“id”:1
        , “nome”:'Anderson'
        ,“sobrenome”:'Zanichelli'
        ,”e-mail”:'anderzanic@gmail.com'
        }
```

Além da comunicação entre as aplicações utilizarem JSON, as informações são armazenadas em um banco de dados baseado em documentos, o MongoDB, que também utiliza JSON, assim

temos Javascript sendo utilizados em todas as pontas.

Os meses seguintes, Junho até o momento, foram utilizados para a implementação das aplicações, de acordo com o cronograma eu deveria estar encerrando as atividades de implementação mas infelizmente ela seguirá um pouco mais.

A Modelagem da arquitetura do sistema foi pensada da seguinte forma:

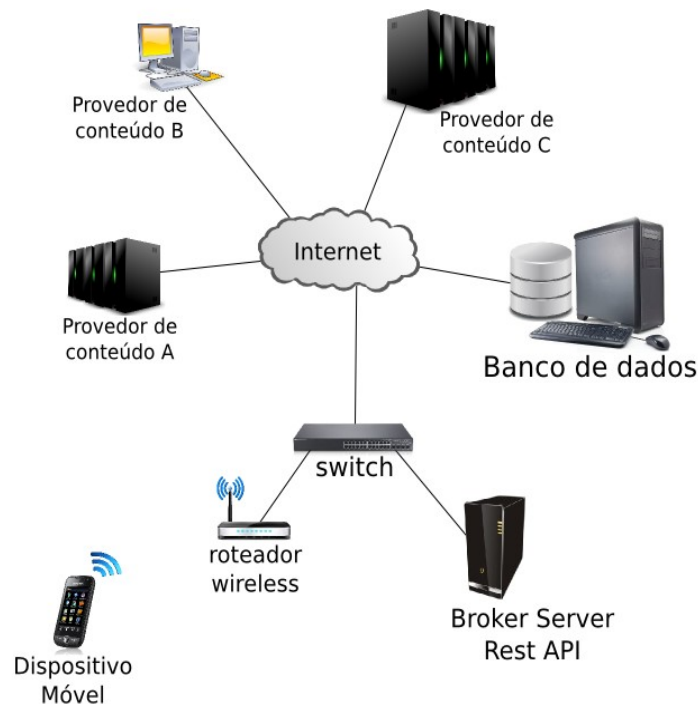


Figura 1 – Modelagem da arquitetura do sistema

- Os provedores de conteúdo registrarão no Broker os serviços que têm a oferecer;
- O Broker disponibilizará aos usuários/dispositivos informações dos serviços oferecidos pelos provedores e oferecerá aos usuários/dispositivos o serviço que atenderá aos requisitos registrados pelos usuários;
- O usuário poderá escolher quais serviços quer utilizar;
- O usuário poderá configurar quais atributos de contexto são mais relevantes para ele e assim registrá-los no Broker para que o Broker faça a escolha do melhor serviço.

2. Revisão da Literatura

O AngularJs é um framework estrutural para aplicações web dinâmicas, ele permite o uso de HTML e CSS para a estruturação da aplicação e também estende a sintaxe do HTML permitindo a criação de componentes que poderão ser utilizados de forma clara e sucinta.

O framework conta também com um mecanismo de injeção de dependências que tira do desenvolvedor a responsabilidade de implementar uma grande quantidade de código.

O Ionic é um framework que faz uso da API do Apache Cordova. O Apache Cordova permite que aplicações façam o uso das funções dos dispositivos através da linguagem Javascript, o Ionic além de outras coisas, como por exemplo oferecer componentes visuais para serem utilizados na aplicação desenvolvida, tem o importante papel de fazer a ponte entre o AngularJs e o Apache Cordova.

Os aplicativos desenvolvidos através desses frameworks não terão o mesmo desempenho que um aplicativo construído com a linguagem nativa. Mas como muitos aplicativos não necessitam de um alto poder de processamento, conseguimos criar muitas aplicações úteis utilizando essas tecnologias e graças ao desempenho do hardware dos dispositivos os usuários não se incomodarão com a diferença de desempenho em relação a uma aplicação nativa.

3. Resultados obtidos e dificuldades encontradas

Como o desenvolvimento de aplicativos móveis está em alta no mercado, está sendo relativamente fácil encontrar informações sobre todas essas tecnologias citadas, e não somente textos que explicam o básico, mas documentações completas nos sites dos fornecedores e também comunidades de desenvolvedores empenhados em divulgar e ensinar a desenvolver. É possível encontrar também muitas vídeo-aulas de qualidade que apresentam conteúdos interessantes e relevantes ao desenvolvimento de aplicativos móveis.

Uma das dificuldades encontradas foi na implementação de uma template que deve montar uma tela com diferentes componentes dinamicamente. A dificuldade encontrada foi referente a forma como o AngularJs renderiza as telas. Como o framework estende a linguagem HTML, ele permite a criação de novas tags para a linguagem, mas para que isso funcione o código passa por um processo de compilação antes da renderização.

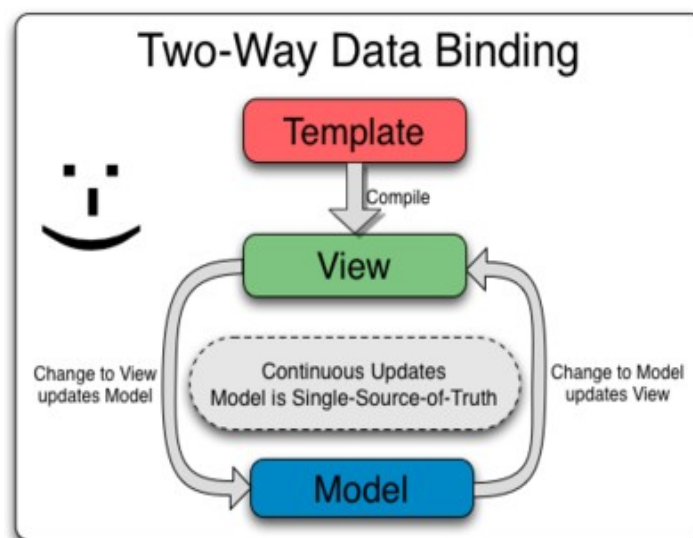


Figura 2 – Compilação e ciclo de vida de uma página Angular

O problema enfrentado foi em um trecho de código que é injetado na página após essa etapa de compilação, assim foi necessário buscar formas alternativas de como inserir esse trecho de código, as opções encontradas até o momento são o uso do framework jQuery, que é uma grande coleção de plugins para Javascript, onde o código injetado terá uma porção de Javascript que irão executar as ações necessárias e a outra opção será uma função na própria diretiva Angular que recompilaria o código que é injetado. Ainda estou trabalhando nessas opções.

4. Avaliação do cronograma de execução

Com uma pequena análise podemos perceber que o cronograma está sendo respeitado.

5. Referências Bibliográficas

ANGULARJS, Guia da Documentação de AngularJs. Disponível em: <https://docs.angularjs.org/guide/> Acesso em Setembro de 2015

FREEMAN A.; Pro AngularJs. Apress. 2014.

IONIC Framework, Documentação do Ionic. Disponível em: <http://ionicframework.com/docs> Acesso em Setembro de 2015

MONGODB, Documentação do MongoDB. Disponível em <http://docs.mongodb.org/manual/> Acesso em Setembro de 2015

NODEJS, Documentação do NodeJs. Disponível em: <https://nodejs.org/documentation/> Acesso em Setembro de 2015

OSMANI A.; Learning Javascript Design Patterns. O' Reilly – 2012.

PANDA S.; AngularJs Novice to Ninja. Site Point – 2014.