



Uma Arquitetura de Referência Baseada em Plugins para Sistemas de Informação Mobile **Enoque Joseneas**

Orientador: Sandro Andrade

INF023 – Trabalho Final de Curso

manoelnetom@ifba.edu.br

Apresentação



- **Problema**
- **Solução Proposta**
- **Justificativa**
- **Avaliação Experimental**
- **Resultados**
- **Conclusão**

Problema

- **O desenvolvimento de aplicativos conta com uma grande variedade de frameworks, linguagens e ferramentas. Porém carece de:**
 - **Soluções arquiteturais de alto nível (baixo acoplamento e facilidade de extensão)**
 - **Criação de componentes de UI em linguagem de alto nível (No android, utiliza-se arquivos xml + classes Java)**

Problema



- **Suporte facilitado para comunicação com serviços RESTful**
- **Não foi encontrado na literatura um framework baseado em plugins para o desenvolvimento de aplicativos**

Solução proposta



- **Uma Arquitetura de plugins através de um framework reutilizável**
 - **Baixo acoplamento e fácil extensibilidade**
 - **Plugins independentes entre si e do núcleo do framework**
 - **Suporte facilitado a reuso de componentes**
 - **Dispõe de 10 componentes reutilizáveis**
- **Dispõe quatro requisitos funcionais via componentes de alto nível para uso de cada requisito**

Solução proposta



- **Requisitos Funcionais Implementados**
 - **Acesso a rede (HTTP) para comunicação com serviços RESTful facilitado, através de APIs de alto nível**
 - **Suporte a persistência de dados local via SQLITE e QSettings (persistência simples – chave/valor)**
 - **Suporte a notificações via push através do Firebase e local via *procedure call***
 - **Suporte a comunicação entre objetos e plugins através de eventos via *Observer***

Solução proposta



- **Destaques do Framework**

- **Suporte a download e upload de arquivos**
 - **Acesso a galeria de imagens/arquivos do sistema**
- **Suporte para execução em modo desktop**
 - **Ótimo para realizar testes durante o desenvolvimento**
- **Suporte a dois tipos de layout do aplicativo**
 - **SwipeView e StackView (é possível usar os dois)**
- **Fácil customização de estilo dos widgets**
 - **Configuração de propriedades em arquivo json**

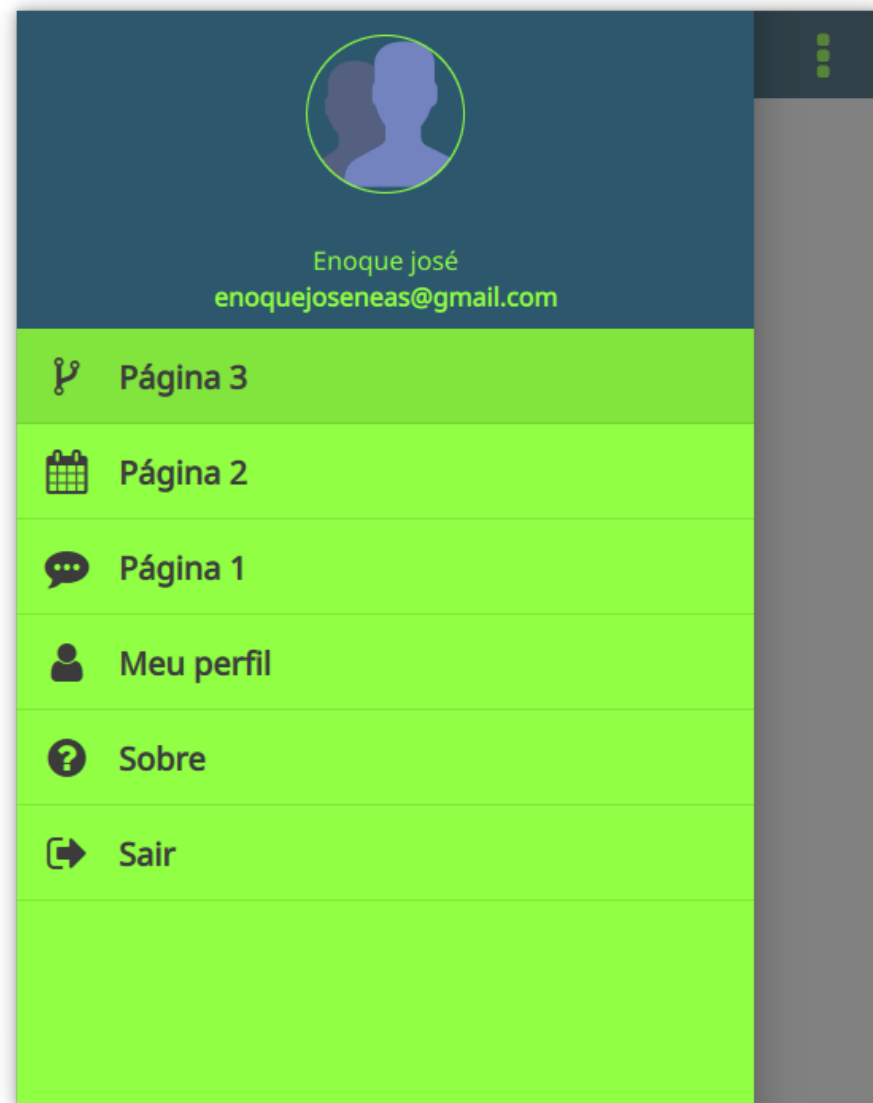
Solução proposta



- **Infraestrutura de Plugins**
 - **Plugins devem criar um arquivo json de configuração**
 - **Plugins podem adicionar uma lista de observers**
 - **Plugins podem adicionar arquivos e imagens em seu diretório que serão empacotados no aplicativo**

Solução proposta

```
{  
  pages:[  
    {  
      qml: "Page1.qml",  
      title: "Pagina_1",  
      awesomeIcon: "commenting",  
      order:3,  
      roles: [ "student" ],  
      showInDrawer: true ,  
      showInTabBar: true  
    },  
    {  
      qml: "Page2.qml",  
      title: "Pagina_2",  
      awesomeIcon: "calendar",  
      order:4,  
      roles: [ "student", "teacher" ],  
      showInDrawer: true ,  
      showInTabBar: true  
    },  
    {  
      qml: "Page3.qml",  
      title: "Pagina_3",  
      awesomeIcon: "code_fork",  
      order:5,  
      roles: [ "student", "teacher" ],  
      showInDrawer: true ,  
      showInTabBar: true  
    }  
  ],  
  ...  
}
```



Solução proposta

- **Trabalhos correlacionados**
 - **4 trabalhos foram destacados**

Recurso	Este trabalho	CReAMA	MoCA	Trabalho de Jauri com SenchaTouch e PhoneGap	Trabalho de Lisandro utilizando Xamarin.Forms
Provê suporte multiplataforma Android e iOS	Sim	Não	Não	Sim	Sim
Provê suporte a plataforma desktop	Sim	Não	Não	Não	Não
Provê suporte a recursos extensíveis através de plugins	Sim	Não	Não	Não	Não
Provê APIs de alto nível para recursos de rede (HTTP), banco de dados e UI	Sim	Sim	Sim	Sim	Sim
Provê suporte a reuso de componentes	Sim	Sim	Sim	Sim	Sim
Provê suporte a comunicação entre objetos através de eventos	Sim	Não	Sim	Sim	Sim

Justificativa



- **Criação rápida de aplicativos móveis**
 - **Escrita de código reduzida através dos plugins**
 - **O desenvolvedor escreverá somente as features do aplicativo**
- **Bom desempenho**
 - **Backend em Qt/C++**
 - **Comunicação não bloqueante entre objetos**
- **Construção de componentes de UI em linguagem de alto nível através do QML**

Avaliação Experimental



- **Baseado na extração de métricas a partir do código fonte de dois aplicativos**
 - **Objetivo: validar os benefícios da arquitetura proposta**
- **Foi utilizado o Emile como objeto da avaliação**
 - **Foram desenvolvidas duas versões do Emile**
 - **Uma versão foi baseada no framework e a outra não**
 - **Foram escolhidas três funcionalidades do Emile**

Avaliação Experimental



- **Funcionalidades selecionadas**
 - **Login do usuário**
 - **Inclui o logout**
 - **Gerenciamento de mensagens**
 - **Inclui a listagem e o envio de mensagens**
 - **Gerenciamento de perfil do usuário**
 - **Inclui a visualização e a edição dos dados do usuário**

Avaliação Experimental



- **Métricas escolhidas na avaliação**
 - **Número de linhas de código implementado em cada versão (com e sem o framework)**
 - **Apenas dos plugins na versão com o framework**
 - **Todos os arquivos na versão sem o framework**
 - **Densidade de bugs encontrados em cada versão**
 - **Coletados a partir da execução do aplicativo**

Avaliação Experimental



- **Execução da avaliação**
 - **Versão com o framework**
 - **Três plugins foram implementados**
 - **18 arquivos foram criados**
 - **(+/-) 120 horas de desenvolvimento**
 - **Versão sem o framework**
 - **23 arquivos foram criados**
 - **Forte acoplamento dos recursos no núcleo da aplicação**
 - **(+/-) 240 horas de desenvolvimento**

Resultados e Discussão



- **Versão com o framework**
 - **Maior tempo de build e deploy**
 - **Arquivo final (APK) maior (16 MB)**
 - **Número total de linhas: 1116**
- **Versão sem o framework**
 - **Build e deploy mais rápido**
 - **Arquivo final (APK) menor (12 MB)**
 - **Número total de linhas: 5206**

Resultados e Discussão



- **Versão com o framework**
 - **5 bugs foram identificados, porém mais complexos**
 - **Maior dificuldade de depurar erros internos**
 - **Bugs críticos em telas que faziam acesso a rede**
- **Versão sem o framework**
 - **4 bugs foram identificados**
 - **Bugs somente de UI**

Conclusão



- **A arquitetura é rica em recursos, proporciona baixo acoplamento e facilidade de extensão e manutenção por conta dos plugins**
- **Destaca-se pela facilidade de implementar comunicação com Web Services RESTful**
- **Permite ao desenvolvedor ganhar tempo escrevendo menos código através das APIs fornecidas, dedicando-se apenas na criação de plugins**
- **O desenvolvedor não precisará escrever código Java, C++ ou Objective C, trabalhando apenas com QML (alto nível)**

Trabalhos Futuros



- **Documentação para uso das APIs com exemplos de código**
- **Suporte a escrita de plugins em C++**
- **Melhorar o suporte ao iOS**
- **Correção dos bugs encontrados na API de rede**
- **Adicionar suporte a outros tipos de autenticação na API de rede tais como o OAUTH 1.0 e 2.0**



Uma Arquitetura de Referência Baseada em Plugins para Sistemas de Informação Mobile

Enoque Joseneas

enoquejoseneas@ifba.edu.br