



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

Orientador: Sandro Andrade

INF023 – Trabalho Final de Curso

manoelnetom@ifba.edu.br

Apresentação



- **Problemática**
- **Solução Proposta**
- **Avaliação Experimental**
- **Resultados e Discussão**
- **Conclusão**

Problemática

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

Problemática

- **O desenvolvimento de aplicativos possui grande variedade de frameworks, linguagens e ferramentas. Porém:**
 - **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 Referência Baseada em Plugins para o Desenvolvimento de Aplicativos no Contexto de Sistemas de Informação**
- **Possui quatro requisitos funcionais implementados e dispõe componentes de alto nível para uso de cada requisito**

Solução proposta



- **Um framework com uma arquitetura de plugins**
- **Possui um núcleo genérico sem implementações de UI (tarefa dos plugins)**
- **Foco em aplicativos voltados para sistemas de informação**
 - **suporte a CRUD facilitado**
 - **acesso a Internet, login e notificações do usuário**

Solução proposta



- **Utiliza o Qt (C++) no Backend e QML no desenvolvimento de plugins e na construção de componentes de UI**
- **Dispõe 10 Componentes Reutilizáveis para agilizar o desenvolvimento de interfaces, widgets e uso de recursos**

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 (via em 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**

- **Facilidade de extensão através dos plugins**
- **Baixo acoplamento entre os componentes**
- **Escrita de código reduzida através dos plugins**
- **Comunicação não bloqueante entre objetos**
- **Suporte a dois tipos de layout do aplicativo**
 - **SwipeView e StackView (é possível intercalar os dois)**

Solução proposta



- **Destques do Framework**
 - **Suporte para aplicações em plataforma desktop**
 - **Ideal para testes e depuração da aplicação**
 - **Fácil customização de estilo dos widgets**
 - **É possível utilizar Material Desig, Univernal ou Default, além de definir cores e tamanho de fonte via arquivo de configuração**

Solução proposta



- **Destaques do Framework**

- **Suporte a Basic Authentication na API de rede**
- **Suporte a download e upload de arquivos facilitado**
- **Arquivos dos plugins não precisam estar mapeados no qrc para serem empacotados**
- **Suporte a operações de baixo nível via procedure call**
 - **criar, carregar ou remover arquivos, exibir uma notificação ao usuário, persistir dados etc.**

Avaliação Experimental



- **Baseado na extração de métricas a partir do código fonte de dois aplicativos**
 - **Objetivo: validar a eficácia e os benefícios da arquitetura proposta**
- **Foi utilizado o Emile como objeto da avaliação**
 - **Foi desenvolvido duas versões do Emile**
 - **Uma versão foi baseado no framework**
 - **Foi utilizado o Qt/QML (versão sem o framework)**

Avaliação Experimental



- **Foi selecionado 3 features do Emile**
 - **Login do usuário**
 - **Inclui o logout**
 - **Gerenciamento de mensagens**
 - **Incluiu a listagem e o envio de mensagens**
 - **Gerenciamento de perfil do usuário**
 - **Incluiu 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)**
 - **Densidade de bugs encontrados em cada versão**
 - **Número de alterações realizadas na versão com o framework, necessárias para poder utilizá-lo**

Avaliação Experimental



- **Execução da avaliação**
 - **Versão com o framework**
 - **Três plugins foram implementados (18 arquivos)**
 - **17 alterações foram necessárias**
 - **(+/-) 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 (somente os plugins): 1116**
- **Versão sem o framework**
 - **Build e deploy mais rápido**
 - **Arquivo final (APK) menor (12 MB)**
 - **Número total de linhas (Todos os arquivos): 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**
 - **Acesso a galeria de imagens não foi possível**

Conclusão



- **A arquitetura destaca-se rica em recursos, proporciona baixo acoplamento e proporciona facilidade de extensão e manutenção através dos plugins**
- **Destaca-se pela facilidade de implementar comunicação com Web Services RESTful**
- **Permite ao desenvolvedor ganhar tempo e escrever 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



- **Construção de uma API para depuração facilitada**
- **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 como o OAUTH e BEARER Token**



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

Enoque Joseneas

enoquejoseneas@ifba.edu.br