

<b>G4S</b>	
Architecture Notebook	Date: 19/04/2011

# Gasoline for Savers (G4S)

## Architecture Notebook

### 1. Purpose

Este documento descreve a filosofia, decisões, restrições, justificações, elementos significantes, e qualquer outro aspecto global desse sistema obedecendo o design e a implementação descrita.

### 2. Architectural goals and philosophy

A nossa arquitetura possui uma filosofia Cliente-Servidor orientado a Serviços. Como estamos lidando com uma plataforma mobile, nosso sistema tem como prioridade, a otimização dos recursos de hardware, como processamento e memória, a fim de uma maior eficiência no uso da bateria. Outra fator primordial é o Design de Interação, que deve ser o mais simples e intuitivo possível, devido as limitações inerentes da plataforma.

Problemas e situações críticas que guiarão a nossa filosofia: O sistema precisa utilizar eficientemente o hardware? É preciso ter uma boa usabilidade? É preciso um bom gerenciamento da comunicação com o servidor?

### 3. Assumptions and dependencies

Podemos listar como exemplos de pressupostos: conexão com a internet, utilização de GPS, API do GOOGLE MAPS, API do SQLite, entre outros.

Como dependências temos as áreas críticas de portabilidade e desempenho para arquitetura mobile. Outra dependência significativa é a falta de experiência da equipe no desenvolvimento de aplicações portáteis e o tempo a ser gerenciado devido aos compromissos com as demais disciplinas.

### 4. Architecturally significant requirements

Podemos listar como links e referências:

<http://developer.android.com/index.html> - Para consulta e apoio ao desenvolvimento baseado em android.  
<http://pt.wikipedia.org/wiki/Cliente-servidor> e [http://pt.wikipedia.org/wiki/Service-oriented\\_architecture](http://pt.wikipedia.org/wiki/Service-oriented_architecture) - Para referência em relação à arquitetura Cliente-Servidor Orientado a Serviço.

### 5. Decisions, constraints, and justifications

- Utilizar a plataforma Android  
-Justificativa: Devido a popularidade, já que é um sistema operacional para plataforma mobile mais difundido, pela perspectiva de futuro, curva de aprendizado e pela possibilidade de desenvolvimento de software livre.
- Restringir a área de atuação do aplicativo  
-Justificativa: Restringimos ao Brasil, visto que este aplicativo não teria área de atuação válida para o exterior e notamos que na região escolhida existe uma falta de concorrência e suporte à ideia geral do aplicativo.
- Utilizar a API do GOOGLE MAPS  
-Justificativa: Estamos utilizando a API do GOOGLE MAPS pela fácil integração à plataforma escolhida e pela sua completude, dessa forma não precisaremos desenvolver uma ferramenta para tratar a localização geográfica e sua localização.
- Utilizar a arquitetura baseada em protótipos  
-Justificativa: Utilizamos para um desenvolvimento ágil, com o objetivo de apresentar um modelo executável, até mesmo algo que poderia ser apresentado à um possível cliente.

<b>G4S</b>	
Architecture Notebook	Date: 19/04/2011

## 6. Architectural Mechanisms

### GPS

Definir a posição geográfica do dispositivo utilizado

Atributos: latitude e longitude

Possui como função definir a posição geográfica do dispositivo em determinado momento através de um hardware específico

Estado: modelado, porém não implementado

### Google Maps

Exibir para o usuário a sua posição geográfica através de uma interface

Atributos: mapas, marcadores e escala

Possui como função exibir para o usuário informações geográficas como posição ou rota desejada através de uma interface gráfica

Estado: modelado e 70% implementado

### Wireless ou 3G

Se conectar ao servidor utilizado e requisitar informações e/ou funções

Atributos: taxa de transferência e preço

Possui como função a conexão ao servidor para requisição de informações e/ou funções através de um protocolo de comunicação confiável.

Estado: modelado e 95% implementado

## 7. Key abstractions

GeoLocalização, é um requerimento básico para o sistema, pois é com ele que se colocará no mapa a localização do usuário e dos postos.

Rota, mostra a rota entre o usuário e os postos.

GoogleMapa, abstração geral do mapa onde será localizado o usuário, os postos com os preços e a região.

Tela, será a região onde o aplicativo será mostrado, podendo ser celular, tablet, etc.

## 8. Layers or architectural framework

Nossa arquitetura usa o padrão dos desenvolvedores de android designado pelo site android developers, feito para todos os desenvolvedores se guiarem. A consistência do aplicativo será garantida pelos vários testes que teremos no seu desenvolvimento (TDD), a segurança será mantidas pelos testes dos casos de usos.

## 9. Architectural views

Nossa arquitetura será descrita de três maneiras a fim de detalhá-la:

- **Lógica:** A arquitetura do projeto foi criada utilizando dois tipos bem conhecidos, a cliente-servidor e a orientada a serviços. Ela foi projetada assim devido ao funcionamento do sistema e também devido aos tipos de serviços em que ele oferece. A parte cliente-servidor foi destinada a parte do sistema de requisição de mapas do Google Maps e de da parte de requisição dos postos de combustíveis ao servidor da ANP que estão no pacote de serviços do programa. A parte de orientação a serviços vem dos serviços que o aplicativo irá prover que são mostrar as rotas até o posto escolhido e de calcular qual é a melhor rota para o usuário escolher, que será mostrado diretamente ao usuário na interface. Por hora não há dados persistentes

<b>G4S</b>	
Architecture Notebook	Date: 19/04/2011

no sistema, todos serão adquiridos da internet e usaremos para utilizar o Google Maps a própria API do android que já trata toda essa comunicação (protocolos, pedidos, respostas, etc.) e para o banco da ANP usaremos SQL por meio de pedidos em HTTP.

- **Operational:** O sistema pode ser dividido em Android OS , que irá rodar o nosso aplicativo, O GPS do aparelho, que dará a localização atual do usuário e será gerenciado pelo GPS manager , as informações que colheremos do Banco de dados da ANP será controlada pelo NetworkManager e o Google Maps será obtido pela própria API de Android. O sistema também terá o RouteManager que fará o cálculo da melhor rota para o usuário e mostrará a ele. Em resumo o aplicativo rodará e terá 2 threads de tempo integral rodando que será o NetworkManager e o GPS manager e uma thread que rodará de acordo com a necessidade do usuário que é a RouteManager e terá o aplicativo em si rodando, isso compões o sistema.
- **Use case:** Os casos de uso são: consultar preços, consultar localização, traçar melhor rota, usar mapa. Eles estão representados no diagrama UML.