

DroidMetronome

Arquitetura de Software - Projeto Integrador

Gabriela Aimée Guimarães
Gleibson Wemerson Silva Borges
Gustavo Moraes dos Santos
Hiago Augusto Koziel Rahmig
Julianny Alves da Silva
Paulo de Oliveira Neto
Pedro Henrique Silva Farias
Renan Ofugi Mikami



DroidMetronome - Projeto Integrador

DroidMetronome é um projeto de aplicativo para Android que simula as funções de um metrônomo digital, tanto para auxiliar no estudo musical quanto para acompanhar as interpretações musicais.

O mesmo se refere ao projeto de Software apresentado como trabalho de conclusão para obtenção de aprovação na disciplina de **Arquitetura de Software**, Gerência de Configuração de Software, Gerência de Projeto de Software, Manutenção de Software, Qualidade de Software, e Verificação e Validação.



DroidMetronome - Objetivo

Uma vez terminada a elicitação de requisitos de um sistema, obtém-se a necessidade de se ter uma solução de como deve ser estruturado o software de modo que venha atender as restrições e funcionalidades que o software necessita.

Criando assim no contexto do projeto o **Documento de Arquitetura de Software (DAS)**, para tal solução.



DroidMetronome - O Documento

O documento foi estruturado da seguinte forma, seguindo o modelo do RUP:

1. Introdução
 - 1.1 Finalidade
 - 1.2 Escopo
 - 1.3 Definições, Acrônimos e Abreviações
 - 1.4 Referências
 - 1.5 Visão Geral
2. Representação da Arquitetura
3. Metas e Restrições de Arquitetura
4. Visão de Casos de Uso
 - 4.1 Realizações de Casos de Uso
5. Visão Lógica
 - 5.1 Visão Geral
 - 5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura
6. Visão de Processos
7. Visão de Implantação
8. Visão de Implementação
 - 8.1 Visão Geral
 - 8.2 Camadas
 - 8.2.1 View
 - 8.2.2 Model
 - 8.2.3 Controller
9. Tamanho e Desempenho
10. Qualidade



Finalidade do Documento

Apresentar uma visão geral da arquitetura do sistema e utilizar uma série de visões arquiteturais diferentes para ilustrar os diversos aspectos do sistema.

E tem como intenção capturar e transmitir as decisões significativas do ponto de vista da arquitetura que foram tomadas em relação ao projeto.



Representação da Arquitetura

O documento apresenta a arquitetura como uma série de visões, no qual tais visões são apresentadas como modelos e utilizam a linguagem unificada de modelagem (UML).

Visões
Casos de Uso
Processos
Implantação
Implementação



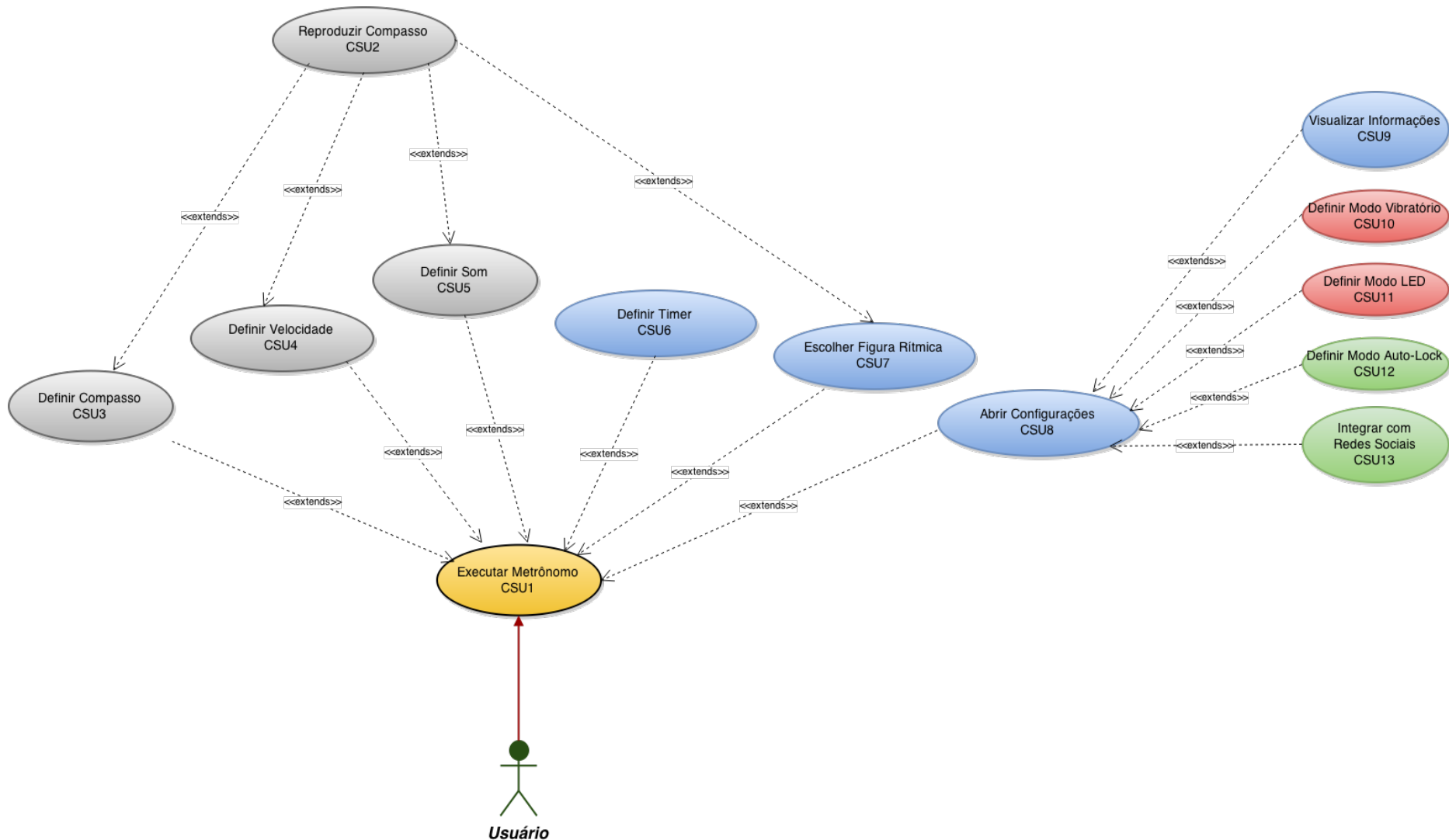
Metas e Restrições da Arquitetura

As restrições de requisito e de sistema possuem relação significativa com a arquitetura, sendo assim, referente ao nosso projeto podemos citar alguns destas restrições :

REFERENCIAS	DESCRIÇÃO
RDES1	O sistema não pode travar/encerrar o processo com frequencia.
RIHC1	As entradas que precisam ser fornecidas serão inseridas através de parâmetros pré-definidos.
RUS3	Os sons das batidas, vibrações e luzes LED podem ocorrer individuais ou simultaneamente.



DroidMetronome - Visão de Casos de Uso



DroidMetronome - Visão de Casos de Uso

CASO DE USO	DESCRIÇÃO
CSU1	O usuário inicializa o sistema DroidMetronome.
CSU2	O usuário inicia as batidas do metrônomo com as configurações selecionadas pelo usuário.
CSU3	O usuário define a divisão quantitativa de pulsos e repousos em uma composição musical.
CSU4	O usuário muda a velocidade dos batimentos do metrônomo sendo estes em BPM.
CSU5	O usuário seleciona um dos sons já pré-definidos.
CSU6	O usuário defini o tempo de duração das batidas.
CSU7	O usuário defini a figura rítmica do compasso.
CSU8	O usuário abre as configurações do metrônomo.
CSU9	O usuário visualiza as informações e descrições sobre o metrônomo.
CSU10	O usuário ativa o modo vibratório, de modo que os batimentos sejam acompanhados por vibrações do aparelho.
CSU11	O usuário ativa o modo LED, de modo que a luz LED do aparelho acenda e desliga acompanhando o metrônomo.
CSU12	O usuário ativa o Auto-lock que impede a tela do aparelho ser bloqueada durante a reprodução o metrônomo.
CSU13	O usuário compartilha o DroidMetronome a outros usuários através de redes sociais.



DroidMetronome - Visão Lógica

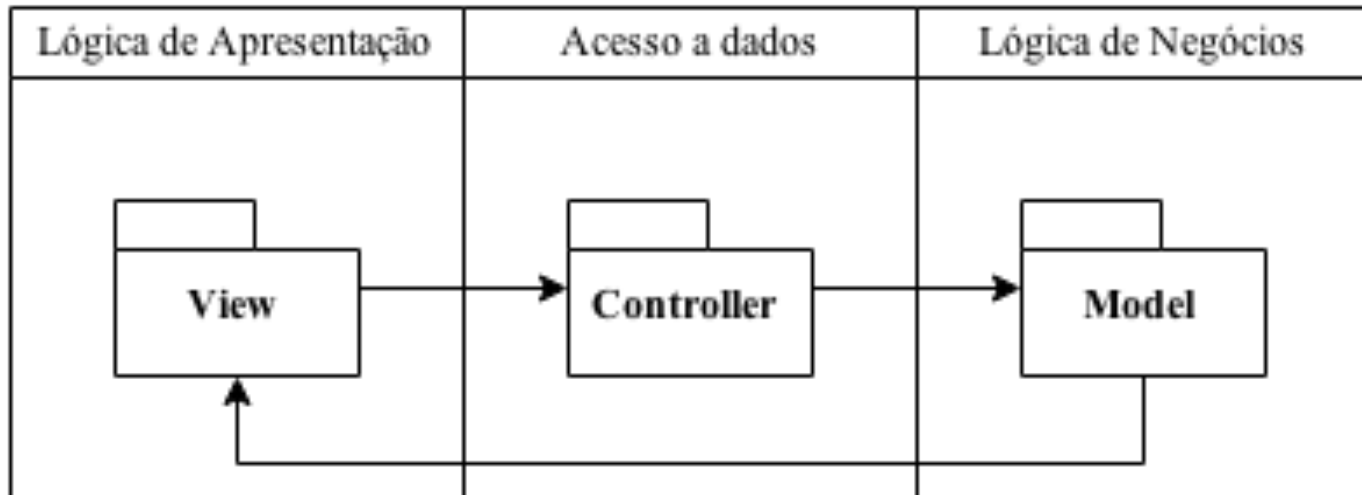
Descreve as classes mais importantes, sua organização em pacotes e subsistemas de serviço, e a organização desses subsistemas em camadas.

A visão geral do DroidMetronome é composta por 3 pacotes:

- View: Este não está preocupado em como a informação foi obtida ou onde ela foi obtida, apenas exibe a informação ao usuário.
- Controller: Este determina o fluxo de apresentação servindo como uma camada intermediária, ou seja, o intercessor entre o View e o Model.
- Model: Este é responsável por tudo o que a aplicação irá fazer. Modelando os dados e o comportamento do sistema, bem como preocupa com o armazenamento, manipulação e geração de dados, sendo um encapsulador de dados e de comportamento independente da apresentação (view).



DroidMetronome - Visão Lógica



DroidMetronome - Visão Lógica

**Pacotes de Design Significativos do Ponto de Vista da Arquitetura -
DIAGRAMA DE CLASSES**

[https://drive.google.com/file/d/0B3Y7QqQ4H6oVOHUyUG1LS1B1LUU/view?
usp=sharing](https://drive.google.com/file/d/0B3Y7QqQ4H6oVOHUyUG1LS1B1LUU/view?usp=sharing)



DroidMetronome - Visão de Implementação

Seguindo o padrão arquitetural MVC ,o sistema pode ser dividido em 3 camadas. Nessas camadas são separados as classes do código com base em sua influência no sistema.

- As classes da camada view são responsáveis apenas pelo front-end do sistema.
- As classes da camada model são as classes atômicas e as classes compostas essenciais para construção do sistema.
- As classes da camada controller são responsáveis pela construção do back-end do sistema.



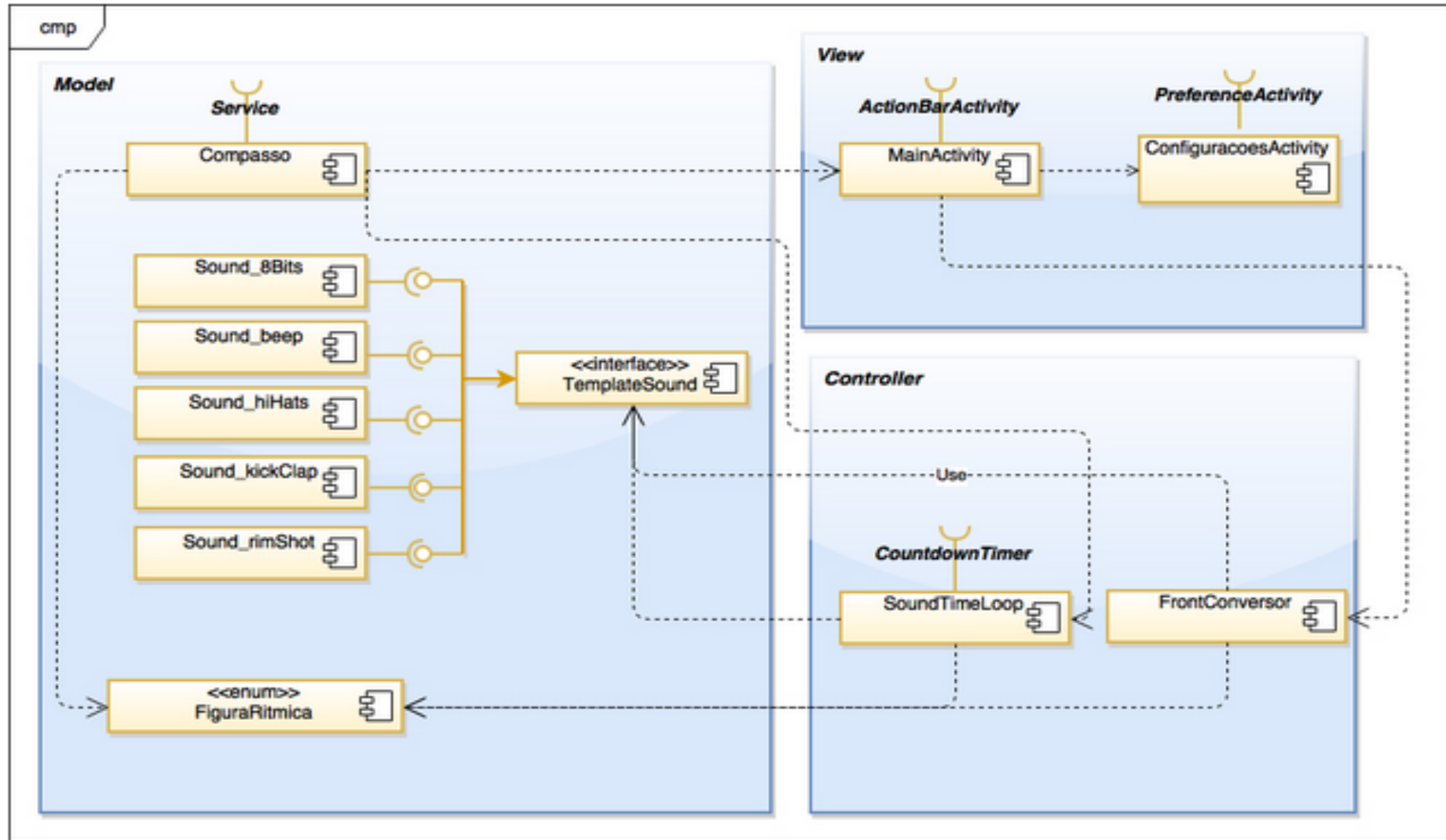
DroidMetronome - Visão de Implementação

CAMADAS:

- View: A camada view é responsável pela interação do usuário com o sistema ,nessa camada são realizados etapas como construção de interface de usuário e interação para as configuração das funcionalidades. A classe que compõe essa camada é a MainActivity.
- Model: A camada model é responsável pelo armazenamento das classes mais atômicas do projeto ,nessa camada são realizados ações como a execução básica dos sons e a execução complexa dos sons. As classes que compõem essa camada são, AudioPlayer e Compasso.
- Controller: A camada controller é responsável pela execução de algoritmos complexos como interpretação dos dados da interface para o sistema e o controle do ciclo de vida da aplicação. As classes que compõem essa camada são, FrontConversor e Executer.

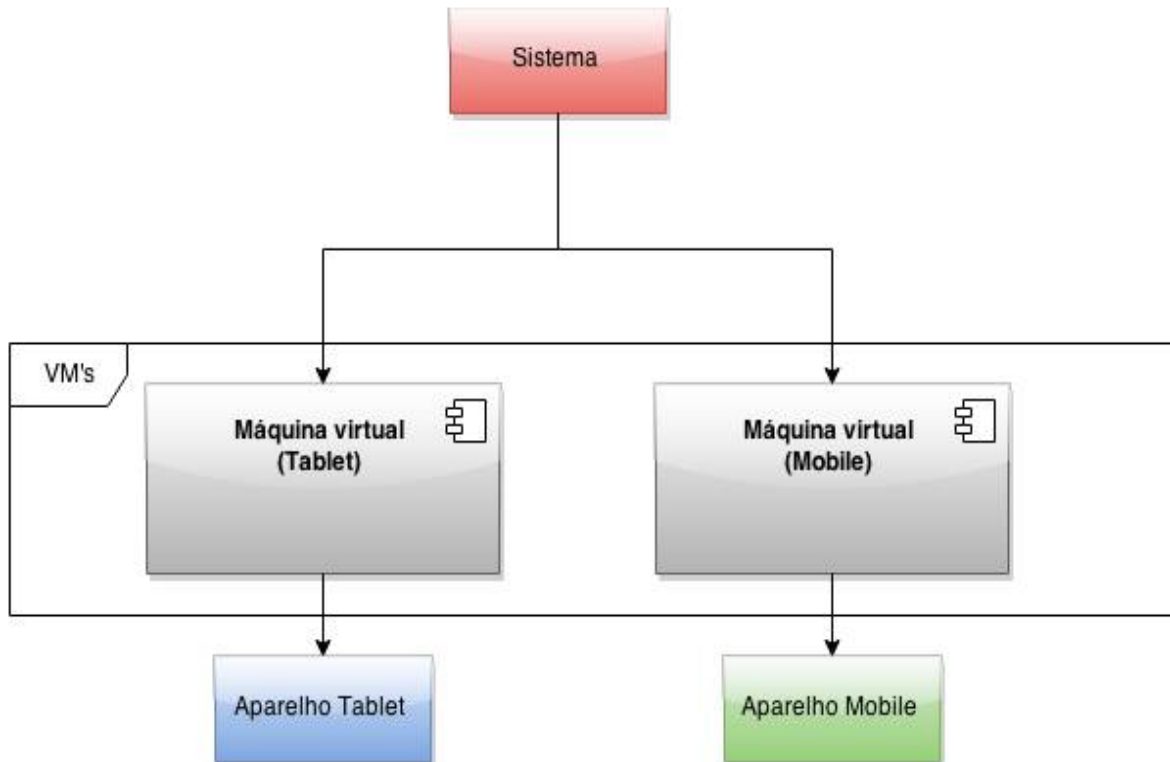


Diagrama de Componentes



DroidMetronome - Visão de Implantação

O sistema é construído na linguagem java ,com o foco voltado para dispositivos android. A linguagem java utiliza de máquinas virtuais para executar o sistema nas mais diversas plataformas (mobile, tablets, entre outros). As máquinas virtuais interpretam o código e executam as instruções nele contidas diretamente nos sistemas, como no diagrama abaixo:



DroidMetronome - Tamanho e desempenho

O sistema DroidMetronome possui tamanho em disco de 216 Kb e é desenvolvido para a plataforma móvel android, entretanto o sistema deve ser construído visando o desempenho do aparelho em que será instalado. A criação de um sistema com muitas classes e arquivos avulsos devem ser ponderados de forma a não afetar (ou afetar o mínimo possível) o desempenho do sistema e do firmware durante a execução.



DroidMetronome - Qualidade

Sabendo que qualidade e seus atributos é a base para as estratégias e decisões de design da arquitetura. O padrão de arquitetura MVC foi a solução mais satisfatória para atender a qualidade esperada do sistema. Sistema este que devera ser desenvolvido na linguagem android, visando uma interface que seja interativa e fácil de se usar.



DroidMetronome

FIM!!!

