

SGSP - Documento de Arquitetura

Versão <1.0>

SGSP - Sistema de Gerenciamento Sócio Pedagógico

Histórico de Revisões

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 07/04/2017 | 1.0 | Inserção dos Diagramas de Arquitetura e Diagrama de Pacotes. | Giovane, Igor, Leonardo, Matheus. |
|  |  |  |  |
|  |  |  |  |

SUMÁRIO

[1. INTRODUÇÃO 2](#_Toc428177093)

[1.1. Definições, Acrônimos e Abreviações 2](#_Toc428177094)

[1.2. Referências 2](#_Toc428177095)

[2. Representação da Arquitetura 2](#_Toc428177096)

[3. Metas e Restrições de Arquitetura 2](#_Toc428177097)

[4. Visão de Casos de Uso 2](#_Toc428177098)

[5. Visão Lógica 2](#_Toc428177099)

[5.1. Visão Geral 2](#_Toc428177100)

[5.2. Pacotes de Projeto 2](#_Toc428177101)

[5.3. Estratégia de Reuso 2](#_Toc428177102)

[6. Visão de Processos (OPCIONAL)| 2](#_Toc428177103)

[7. Visão de Implantação (opcional) 2](#_Toc428177104)

[8. Visão de Implementação 2](#_Toc428177105)

[9. Visão de Dados (opcional) 2](#_Toc428177106)

[10. Tamanho e Desempenho 2](#_Toc428177107)

[11. Qualidade 2](#_Toc428177108)

# INTRODUÇÃO

Este documento contém a definição da arquitetura do sistema para o Sócio Pedagógico do Intistuto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus Salto, em que os desenvolvedores deverão ter conhecimento e utilizar como guia de desenvolvimento.

O documento tem como objetivo de padroniza a organização dos componentes do projeto, define padrões de nomenclatura e para guia todo o desenvolvimento na ajuda na manutenção do sistema, além de permitir que todos os artefatos que realizam um caso de uso sejam rastreados visualmente.

## Definições, Acrônimos e Abreviações

Android Studio 2.3.1 – IDE utilizada para o desenvolvimento do sistema

Android SDK 25.0.5 - Ferramentas que fazem interface com a plataforma Android, estas ferramentas são necessárias para o desenvolvimento de aplicativos Android.

Astah 7.1.0 – Ferramenta utilizada para a criação do Diagrama de Pacotes e Classes.

## Referências

SGSP – Diagramas de Classes.astah

# Representação da Arquitetura

[Esta seção descreve qual é a arquitetura de software do sistema atual e como ela é representada. Nas **Visões de Casos de Uso**, **Lógica**, **do Processo**, **de Implantação** e **de Implementação**, este documento enumera as visões necessárias e, para cada uma delas, explica os tipos de elementos do modelo que são utilizados.]

# Metas e Restrições de Arquitetura

[Esta seção descreve os requisitos de software e os objetivos que têm um impacto significativo na arquitetura, como proteção, segurança, privacidade, uso de um produto desenvolvido internamente ou adquirido pronto para ser usado, portabilidade, distribuição e reutilização. Ela também captura as restrições especiais que podem ser aplicáveis: estratégia de projeto e implementação, ferramentas de desenvolvimento, estrutura das equipes, cronograma, código-fonte legado e assim por diante.]

# Visão de Casos de Uso

[Esta seção lista os casos de uso ou cenários do modelo de casos de uso. Deve ser observado se estes representam uma funcionalidade central e significativa do sistema final ou se têm uma ampla cobertura de arquitetura, ou seja, se experimentam muitos elementos arquiteturais ou se enfatizam ou ilustram um determinado ponto frágil da arquitetura.]

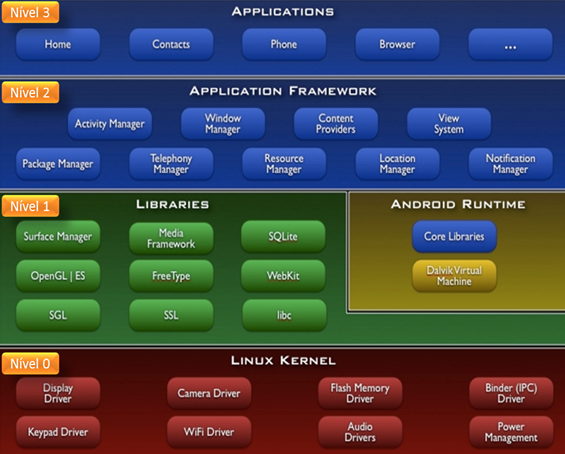
# Visão Lógica

**Introducão ao Android**

O Android é um sistema operacional para dispositivos móveis, ela se basea numa plataforma de código aberta sob a licença apache, os fabricantes tem a liberdade em modificar seu codigo fonte, com isso permite adicionar novos recursos e incorporá-los ao sistema, desta forma o software evolui constantemente.

**Arquiterura Android**

A arquitetura do sistema operacional Android é uma pilha de componentes de software. A arquitetura esta dividido em cinco secções e quatro camadas principais, e estás se dividem em níveis zero, um, dois e três, conforme figura abaixo.



**Nivel Zero – Kernel do Linux**

Na parte inferior das camadas é o Linux - Linux 3.6. Isso proporciona um nível de abstração entre o hardware do dispositivo e contém todos os drivers de hardware essenciais, como câmera, teclado, display etc. Além disso, o kernel lida com todas as coisas que o Linux é realmente bom, como redes e uma vasta gama de drivers de dispositivo , que tomam a dor de interface de hardware periférico.

**Nivel Um – Bibliotecas e Android Runtime**

**Bibliotecas:** Esta categoria abrange as bibliotecas baseadas em Java que são específicos para o desenvolvimento Android. Exemplos de bibliotecas nesta categoria incluem as bibliotecas do framework de aplicação para além dos que facilitam a construção de interface do usuário, desenho gráfico e acesso ao banco.

**Android Runtime:** Esta seção fornece um componente chave que faz uso de recursos do núcleo Linux como gerenciamento de memória e multi-threading, o que é intrínseco na linguagem Java. Dalvik Virtual Machine , que é um tipo de máquina virtual Java especialmente projetado e otimizado para Android.

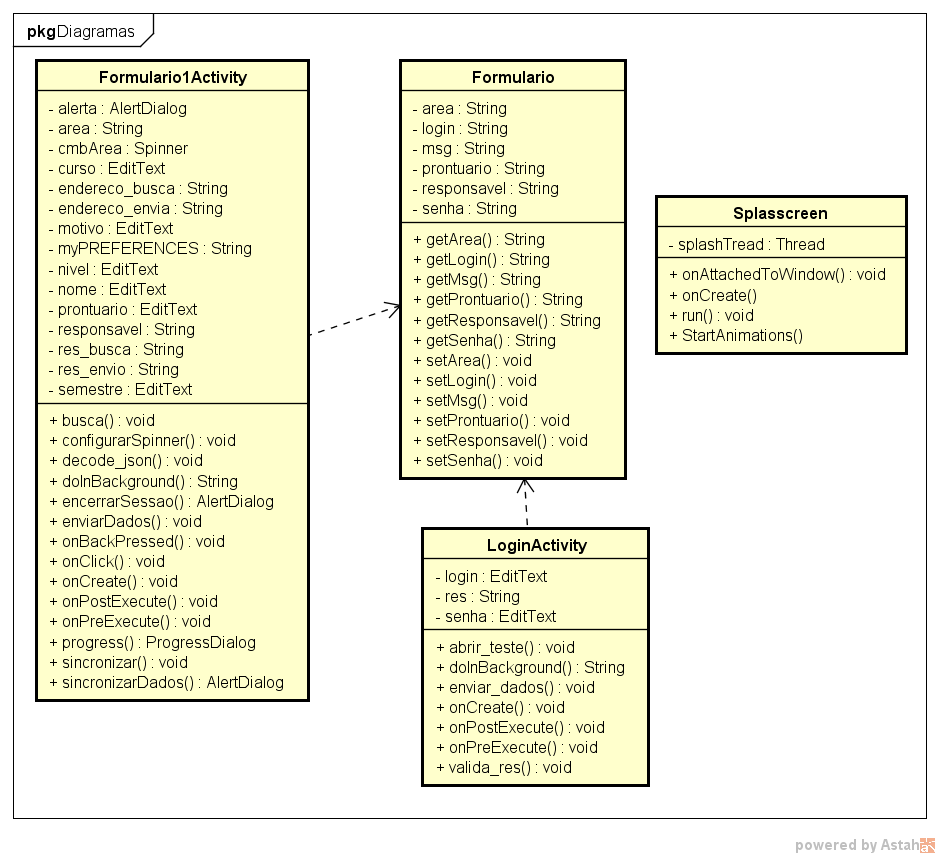
**Nivel Dois - Application Framework**

A camada Application Framework fornece vários serviços de nível superior para aplicações na forma de classes Java. Os desenvolvedores de aplicativos têm permissão para fazer uso desses serviços em suas aplicações.

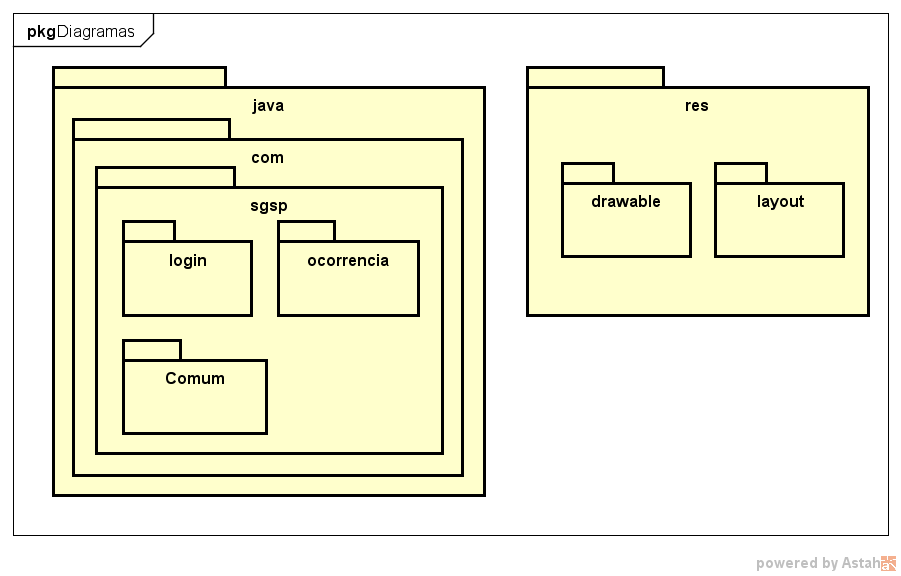
**Nivel Três - Aplicações**

Esta é a camada de interação entre o usuário e o dispositivo móvel, nela encontramos todos os aplicativos cliente como exemplo: e-mail, programa de SMS, calendário, mapas, navegador, contatos entre outros. Você vai escrever seu aplicativo para ser instalado em apenas esta camada.

## Visão Geral



## Pacotes de Projeto



Pacote “drawable”: Onde sera armazenado todos as imagens do aplicativo Android.

Pacote “layout”: Onde tera as tela do aplicativo.

## Estratégia de Reuso

[Descrever a estratégia de reuso e aquisição de componentes. Relacionar os componentes selecionados.]

# Visão de Processos (OPCIONAL)|

[Esta seção descreve a decomposição do sistema em processos leves (threads simples de controle) e processos pesados (agrupamentos de processos leves). Organize a seção em grupos de processos que se comunicam ou interagem. Descreva os modos principais de comunicação entre processos, como transmissão de mensagens e interrupções.]

# Visão de Implantação (opcional)

[Esta seção descreve uma ou mais configurações da rede física (hardware) na qual o software é implantado e executado. Ela é uma visão do Modelo de Implantação. Para cada configuração, ela deve indicar no mínimo os nós físicos (computadores, CPUs) que executam o software e as respectivas interconexões (barramento, LAN, ponto a ponto e assim por diante.) Inclua também um mapeamento dos processos da **Visão de Processos** nos nós físicos.]

# Visão de Implementação

[Esta seção deve descrever a estrutura geral do modelo de implementação. Inclua uma imagem da estrutura física no repositório para mostrar a correspondência com o empacotamento lógico. Inclua também, informações de ferramentas (IDE), como deve ser estruturado o código gerado. Deve mostrar fisicamente como os componentes são organizados para que a estruturação lógica da arquitetura seja implementada.]

# Visão de Dados (opcional)

[Uma descrição da perspectiva de armazenamento de dados persistentes do sistema. Esta seção será opcional se os dados persistentes forem poucos ou inexistentes ou se a conversão entre o Modelo de Projeto e o Modelo de Dados for simples.]

# Tamanho e Desempenho

[Uma descrição das principais características de dimensionamento do software que têm um impacto na arquitetura, bem como as restrições do desempenho desejado.]

# Qualidade

[Uma descrição de como a arquitetura do software contribui para todos os recursos (exceto a funcionalidade) do sistema: extensibilidade, confiabilidade, portabilidade e assim por diante. Se essas características tiverem significado especial, como, por exemplo, implicações de proteção, segurança ou privacidade, elas devem ser claramente delineadas.]