BookMinder

Projeto Arquitetural

There is guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute that allows you to toggle whether it is visible or hidden in this template. Use the Microsoft® Word® menu **Tools > Options > View > Hidden Text** check box to toggle this setting. There is also an option for printing: **Tools > Options > Print**.

# Objetivo

A arquitetura utilizada no sistema baseia-se na arquitetura MVC.

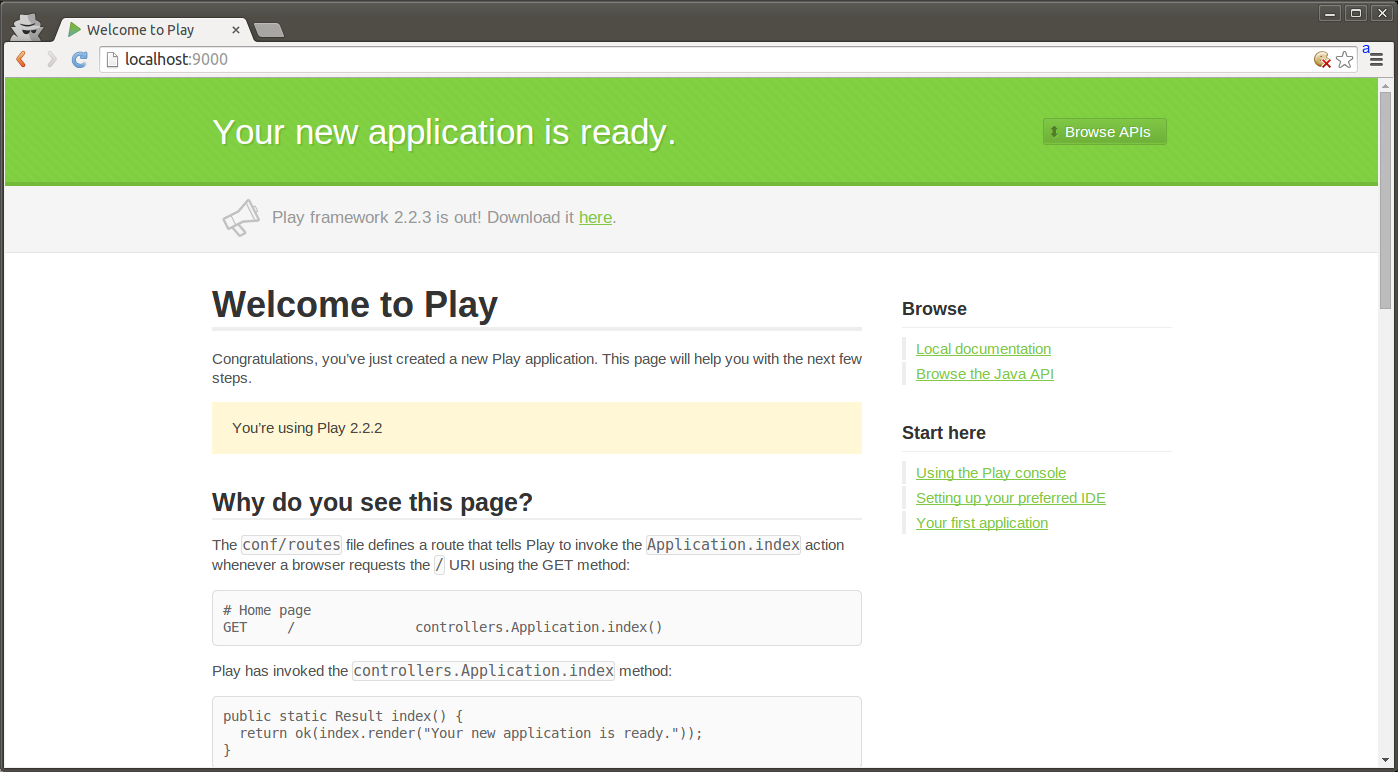
Objetivos:

* Separação de componentes por responsabilidades em comum.
* Utilizado para estruturar e definir melhor as funcionalidades de um sistema.
* Separar o código de acesso aos dados do código de negócio e interface com o usuário (Modularidade)

# Requisitos Críticos da Arquitetura

Para este projeto faremos uso do [Play! Framework](http://playframework.com/), framework que tem ficado cada vez mais famoso devido à sua facilidade de uso, rapidez no desenvolvimento, otimizações de performance e várias outras vantagens. Segue link da documentação do Play! Framework e seus demais módulos utilizados.

* <https://www.playframework.com/>
* <https://www.playframework.com/download>
* <https://www.playframework.com/documentation/2.3.x/Installing>



Notas:

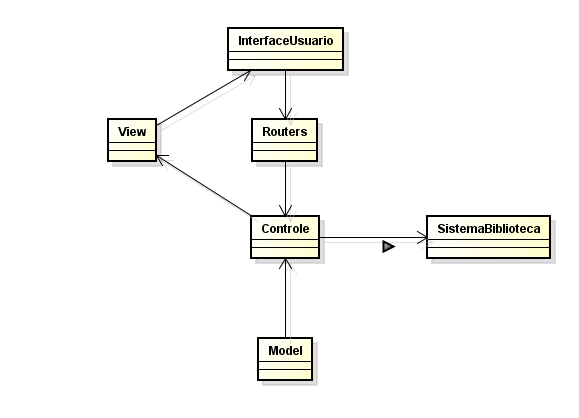
* Para executar o Play! Framework, é necessário o JDK 6 ou superior. Além disso, configurar o diretório bin do JDK na variável de ambiente PATH do seu Sistema Operacional (O Play! Framework irá utilizar o javac, além do java);
* Incluir o diretório que contém os programas play e play.bat também na variável PATH (No meu caso, Figura 1, o caminho será C:\dev\frameworks\play-2.0.4\play-2.0.4);
* No caso do Linux, não esqueça do chmod no arquivo play.

# Mecanismos Arquiteturais

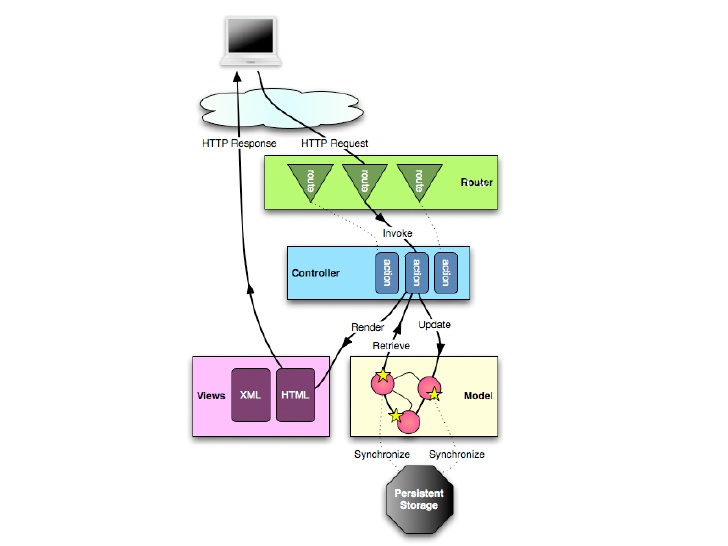
Os diretórios mais importantes de um projeto Play! são os seguintes:

* **app** - contêm os models, controllers e views do projeto;
* **conf**- contêm os arquivos de configuração como o arquivo de rotas (routes), configurações gerais da aplicação (application.conf), internacionalização, etc.;
* **project** – contêm os build scripts do projeto (os scripts são criados para a ferramenta sbt, que é uma ferramenta de build para Scala e Java). Raramente você terá que alterá-lo;
* **public** – contêm recursos como arquivos javascript, css e imagens.

# Principais Abstrações



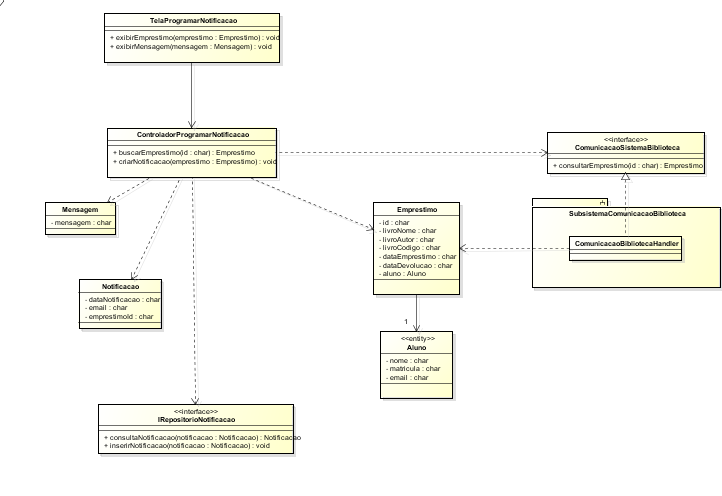
# Camadas do Framework da Arquitetura



# Visões Arquiteturais

• Lógica:

Caso de Uso: Programar Notificação

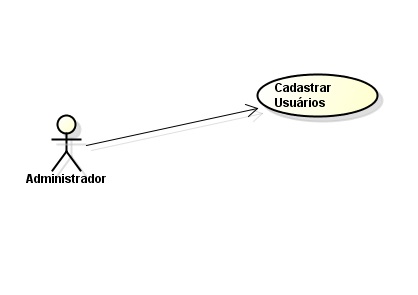


• Operacional:

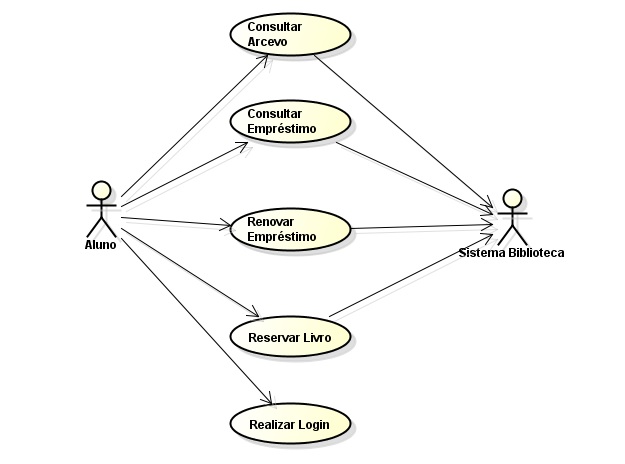
**O sistema possui dependência com Sistema Externo (Integração com Banco de Dados da instituição Faculdade Nova Roma).**

• Caso de uso:

1. Administrador



2. Aluno



3. Funcionário

