RecTourist

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**.

1. Objetivo

O documento detalhará partes do projeto de arquitetura proposta para o desenvolvimento do sistema

Neste documento detalharemos as principais partes do Projeto de arquitetura proposta para o desenvolvimento do sistema.

A arquitetura em que o sistema baseia-se será em camadas MVC, Padrões Orientados a Objetos com destaque no mercado, onde destacaremos em cada parte da arquitetura o motivo da sua criação.

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)

1. Metas Arquiteturais e Filosofia
2. Premissas e Dependências

* Dependência de Banco de Dados externo.
* Dependência com a API do Google Maps.
* Garantia no serviço dos servidores de aplicação para que o aplicativo funcione de forma esperada.
* A interface gráfica será de fácil entendimento para o usuário final.

1. Requisitos Críticos da Arquitetura

Utilizaremos o Spring Framework como principal arquitetura.

Para mais informações clique nos links a seguir:

* <http://projects.spring.io/spring-framework/>
* <http://projects.spring.io/spring-data/>
* [http://projects.spring.io/spring-security/](http://projects.spring.io/spring-framework/)

1. Decisões, Restrições e Justificativas.
2. Mecanismos Arquiteturais

A arquitetura do projeto baseada nos fundamentos do Spring Framework e IoC e injeção de dependências, logo para cada camada possui uma maneira diferente de implementá-la, descreveremos cada uma delas para utilização do Framework:

**Descrição de Pacotes:**

**View**: Contêm as classes que definem a UI com as funcionalidades e o usuário.

**Comunicação:** Contêm a classe e interfaces que fornecerá acesso unificado as camadas de negócio e interface gráfica e também com subsistemas.

**Controller:** Encontram-se as classes que irão tratar as requisições do usuário na interface e também a parte lógica de negócios.

**DAO:** Encontram-se as classes que fazem conexão com o banco de dados e que trata as operações de inserir, consulta, remoção e alteração desses dados.

**Entidades:** Encontram – se as classes que fazem parte do domínio do projeto.

**Exception:** Agrupar – se as classes responsáveis por tratarem erros excepcionais na execução e compilação

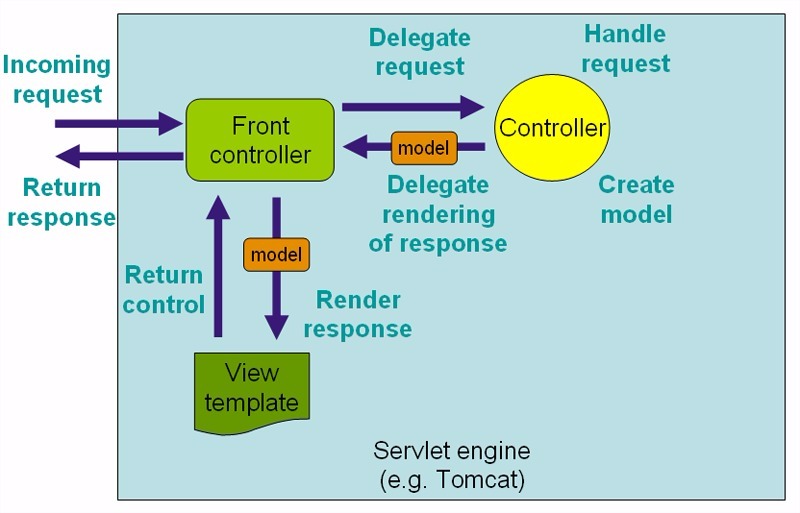
**Message:** Encontra-se as classes que serão utilizadas para geração de mensagem da aplicação

Para pasta WEB-INF terá o arquivo de necessário para a aplicação de visualização e de configuração do Spring

Pasta:

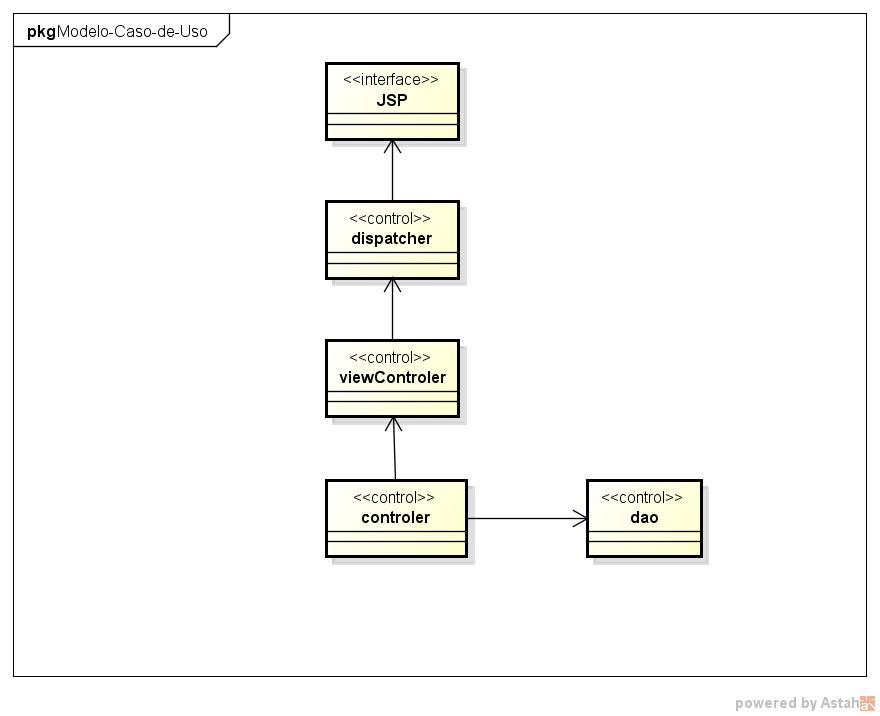
* Resources: pasta contendo todos os arquivos separados em pasta CSS, JS, Imagem do sistema.
* Pages: pasta contendo todos os arquivos para view do sistema separada por função.
* Arquivo do Spring: Responsável por configurar o contêiner do Spring
  + Spring-mvc.xml : Tem a função de configurar o a parte de view do sistema como a pasta que vai procurar as views que serão utilizadas .
  + Spring-data.xml: Tem a função de configurar a conexão com o banco de dados como seja por JDBC ou JPA.
  + Spring-security.xml: Tem a função de configurar a permissão de que tem acesso ao sistema e realizar a autenticação do usuário com seu papel (ROLE) no sistema.
  + Spring-context.xml: Responsável por iniciar as configurações supracitadas além de ser esse aquivo que é referenciado pelo web.xml.
* Web.xml: Responsável por carregar e configurar a aplicação do contêiner do Spring

Sobre a parte de view nesse projeto será utilizado JavaServer Pages (JSP) é uma tecnologia que ajuda os desenvolvedores de software a criarem páginas web geradas dinamicamente baseadas em HTML, XML ou outros tipos de documentos.

****

No nosso caso o Spring framework especificamente o modulo de SpringMvc que trata a parte servlet/filtro é o dispatcher/filtroSecurity configurado pelo spring-mvc e o spring-security inicializado pelo no web.xml.

1. Principais Abstrações



1. Camadas do Framework da Arquitetura

O Spring Framework é composto por recursos organizados em cerca de 20 módulos. Estes módulos são agrupados em Núcleo Container, Data Access / Integração, Web, AOP (Aspect Oriented Programming), Instrumentação, Mensagens e teste, como mostrado no diagrama a seguir.



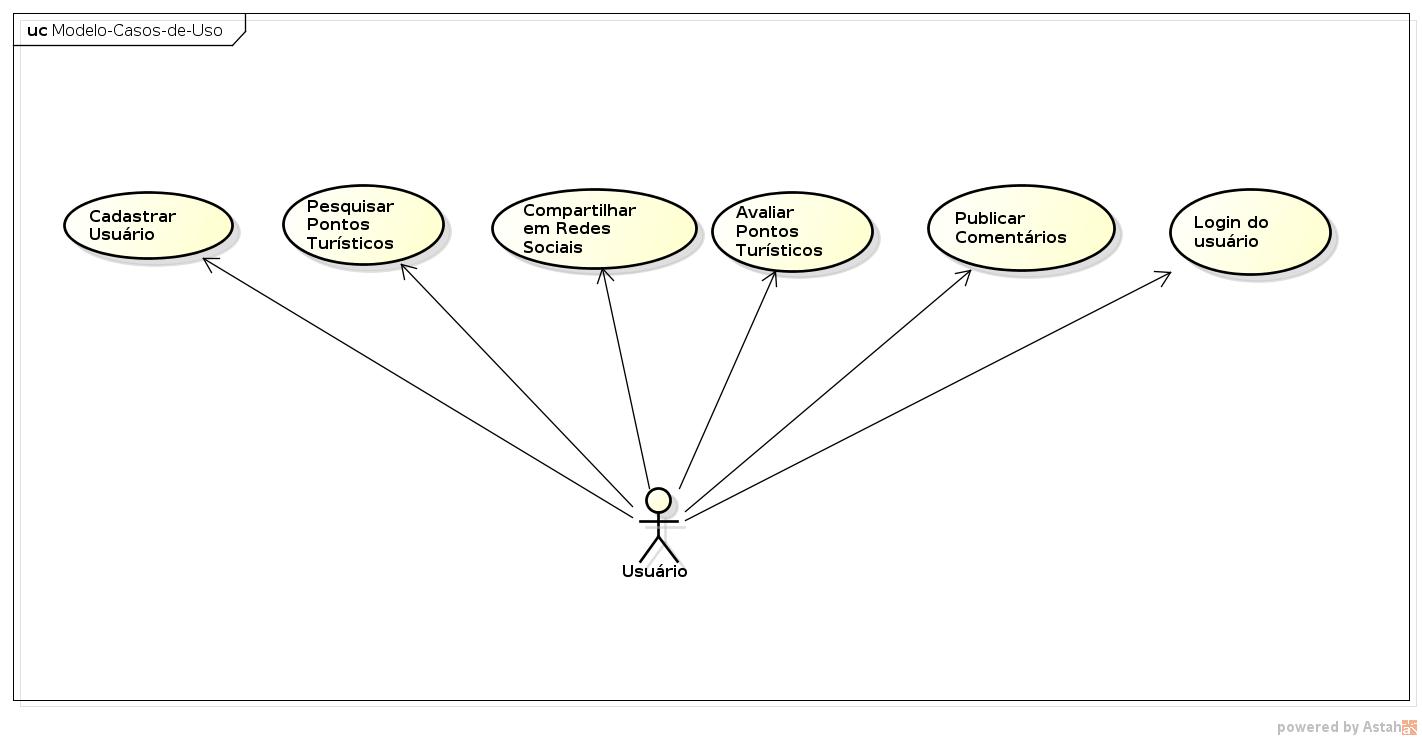
Os blocos de construção descritos anteriormente tornar Primavera uma escolha lógica em muitos cenários, a partir de applets para aplicações empresariais de pleno direito que usam a funcionalidade de gerenciamento de transações e framework web integração do Spring.

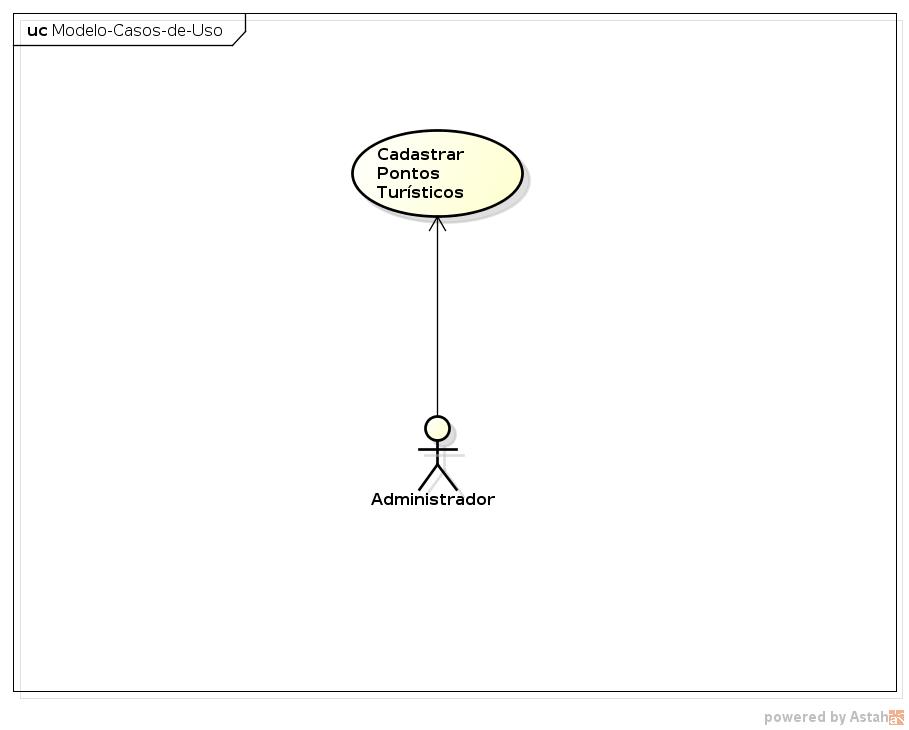
Nesse projeto serão usados:

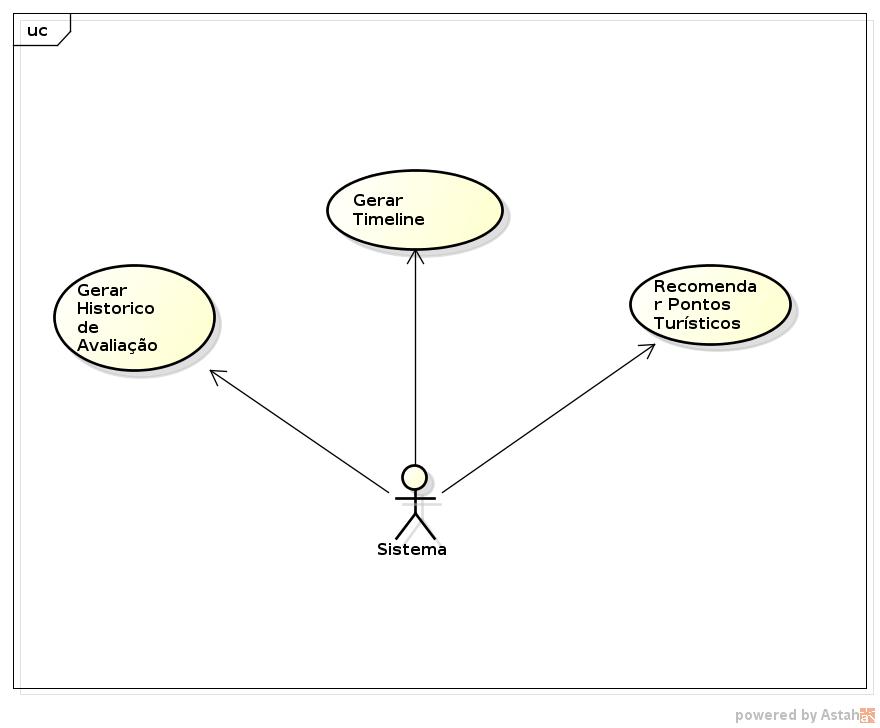
* Os **Spring-core** e **spring-beans** módulos fornecer as peças fundamentais dos blocos a cima, incluindo as características de IoC e injeção dependência. O BeanFactory é uma aplicação sofisticada do padrão de factory. Ele elimina a necessidade de singleton programáticas e permite dissociar a configuração e especificação de dependências de sua lógica do programa real.
* O **Spring-jdbc** módulo fornece um JDBC camada abstrata que elimina a necessidade de fazer a codificação JDBC tedioso e análise de códigos de erro específico de banco de dados de fornecedores.
* O **Spring-ORM** módulo fornece camadas de integração para os populares de mapeamento objeto-relacional APIs, incluindo JPA, JDO e Hibernate e o próprio Spring Data JPA . Usando a primavera-ORM módulo que você pode usar todos esses enquadramentos O / R de mapeamento em combinação com todas as outras características ofertas Spring, como o recurso de gerenciamento de transação declarativa simples mencionado anteriormente.
* A **Spring-web** módulo oferece recursos de integração básicas orientadas para a web como a funcionalidade de upload de arquivo multipart e a inicialização do contêiner IoC usando ouvintes Servlet e um contexto de aplicação orientada a web. Ele também contém as peças relacionadas com a web de apoio a comunicação remota do Spring.
* A **Spring-webMVC** módulo (também conhecido como o Web-Servlet módulo) contém model-view-controller (do Spring MVC) de implementação para aplicações web. Framework MVC do Spring fornece uma separação clara entre o código e modelos de formulários web domínio e integra-se com todos os outros recursos do Spring Framework.

1. Visões Arquiteturais

Visão do usuário:



Visão do administrador:

 Visão do Sistema: