Sistema Pré-Crime

Documento de Arquitetura de Software

Versão 1.0

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 10/07/2018 | 1.0 | Documentação do sistema | Paulo Vitor e Bruno Gois |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Índice Analítico

Conteúdo

[1. Introdução 3](#_Toc361651315)

[1.1 Finalidade 3](#_Toc361651316)

[1.2 Escopo 3](#_Toc361651317)

[1.3 Definições, Acrônimos e Abreviações 3](#_Toc361651318)

[1.4 Referências 3](#_Toc361651319)

[1.5 Visão Geral 3](#_Toc361651320)

[2. Representação Arquitetural 4](#_Toc361651321)

[3. Metas e Restrições da Arquitetura 4](#_Toc361651322)

[4. Visão de Casos de Uso 4](#_Toc361651323)

[4.1 Realizações de Casos de Uso 5](#_Toc361651324)

[5. Visão Lógica 5](#_Toc361651325)

[5.1 Visão Geral 5](#_Toc361651326)

[5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura 6](#_Toc361651327)

[5.3 Diagrama de componentes 7](#_Toc361651328)

[6. Visão de Processos 8](#_Toc361651329)

[7. Visão de Implantação 8](#_Toc361651330)

[8. Visão da Implementação 8](#_Toc361651331)

[9. Qualidade 8](#_Toc361651334)

[10. Exceções 9](#_Toc361651335)

Documento de Arquitetura de Software

# Introdução

Esse documento provê uma visão de alto nível dos objetivos da arquitetura, dos estilos arquiteturais e componentes que foram selecionados para poder estruturar as funcionalidades propostas pelos casos de usos levantados do Sistema Pre-crime.

## Finalidade

* Este documento oferece uma visão geral arquitetural abrangente do Sistema pre-crime, usando diversas visões arquiteturais para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao utilizando a linguagem de modelagem unificada (UML – *Unified Modeling Language*).

A finalidade do documento de arquitetura de software é definir o modelo arquitetural de nosso sitema para ser aplicado ao sistema pre-crime, bem como renuir as informações necessárias da arquitetura, e mostra uma visao dos requsitos arquitetural, funcional e não funcional para dar suporte ao desenvolvemento do site Pre-crime.

## Escopo

* Este Documento de Arquitetura de Software é aplicado ao Sistema Pre-Crime, que será desenvolvido pelos alunos Paulo Vitor e Bruno Gois do curso de Sistemas de Informação da Universidade Federal de Sergipe – Campus Itabaiana, como projeto da disciplina Engenharia de Software II.

Este Documento define as partes significativas dopoto de vista da arquitetural, como sua divisão em camadas e pacotes ,estão contempaldos por este, os padroes de software, componetes de software, paltaformas de desenvolvimento, plataformas de hardware, softwares de desenvolvimento, servidores de aplicação, servidores de banco de dados, sistemas operacionais, frameworks.

## Referências

* Os seguintes documentos foram utilizados como referência para a elaboração do documento arquitetural:
* Documento de Visão;
* Documento de Requisitos;
* Especificação de Use Case;

## Visão Geral

* Com o objetivo de contemplar todos os aspectos da arquitetura, estratificamos o documento nas seguintes subseções:
* Subseção 2: Descreve o uso de cada visão;
* Subseção 3: Descreve as restrições arquiteturais do sistema;
* Subseção 4: Descreve os requisitos funcionais que causam significante impacto na arquitetura;
* Subseção 5: Descreve a visão lógica da arquitetura;
* Subseção 6: Descreve a visão de processos;
* Subseção 7: Descreve a visão de implantação;
* Subseção 8: Descreve a visão de implementação;
* Subseção 9: Descreve a visão de dados;
* Subseção 10: Descreve as principais características de dimensionamento do software que têm um impacto na arquitetura;
* Subseção 11: Descreve como a arquitetura do software contribui para todos os recursos.
* Subseção 12: Mostra a hierarquia de exceções.

# Representação Arquitetural

* Este documento apresenta a arquitetura como um arranjo das visualizações, mencionadas acima. Essas visões são apresentadas como Modelos do StarUML e utiliza a Linguagem de Modelagem Unificada (UML – *Unified Modeling Language).*
* Para representar a arquitetura do software, foram utilizados como base os seguintes estilos arquiteturais:
* Camadas;
* Cliente-Servidor;
* Sub-rotinas;
* Baseado em Eventos;
* Repositório de banco de dados.
* MVC

# Metas e Restrições da Arquitetura

* Para que o software tenha um comportamento esperado por nossos *stakeholders*, definimos as seguintes restrições:
* O sistema web fornece a maioria do conteúdo para exibição. Uma interfacce para esse sistema deve ser capaz de manipular grandes volumes de trafego.
* O Sistema deverão funcionar plenamente nos mais diversos browsers disponíveis no mercado, a exemplo, do *Mozilla Firefox*, *Google Chrome, Opera, Safari, Netscape* e *Internet Explorer*, em suas versões mais recentes;
* O sistema será implementado utilizando os recursos inicialmente disponibilizados pelo *.NET Framework* e ASP.NET MVC CORE, tendo como linguagem de programação principal o C#;
* O sistema terá os seus dados persistidos no banco de dados PostgreSQL;
* Apenas usuários autenticados podem ser acessar o sistema;
* O Sistema será multiplataforma, funcionado principalmente nos sistemas operacionais das plataformas Windows, Linux e MAC OS, em suas versões mais recentes, principalmente;
* O Sistema terá como *Design Pattern* o modelo MVC.

# Visão de Casos de Uso

* Uma descrição de visualização de casos de uso da arquitetura de software. São os cenários e/ou casos de uso que tem o foco em interação.Que representam a funcionalidade central se significativa.Também descreve o conjunto de cenários e/ou casos de uso que possuem cobertura arquitetural substancial ou que enfatizam ou ilustram um determinado ponto complicado da arquitetura.

Os use cases abaixo mostra a prioridade da ordem que sera implementado o sistema.

* CSU001 – Autenticação;
* CSU002 – Apresentar Localização;
* CSU003 – Apresentar Ocorrência;
* CSU004 – Cadastrar Crimes;
* CSU005 – Estatisticas;
* CSU006 – Dicas de Segurança;
* CSU007 – Alterar Configurações;
* 

**Figura 1 – Diagrama de Casos de Uso**

## Realizações de Casos de Uso

* A seguir temos uma breve descrição de cada caso de uso, utilizado na descrição das principais funcionalidades do sistema. São eles:
* CSU001 – Autenticação

Autenticar na ferramenta, criando um login e senha ou autenticar com uma conta já existente.

* CSU002 – Apresentar Localização

Definir sua localização de forma automática ou manual.

* CSU003 – Apresentar Ocorrência

Definir o local da ocorrência de forma automática ou manual.

* CSU004– Cadastrar Crimes

Aqui o usuario vai cadastrar o crime que foi sofrido pelo mesmo ou um crime que ele presenciol.

* CSU005 – Estatisticas

Apresentar os dados estatísticas de cada local solicitado.

* CSU006 – Dicas de Segurança

Apresentar várias dicas de segurança sobre cada tipo crime cometido.

* CSU07 – Alterar Configurações

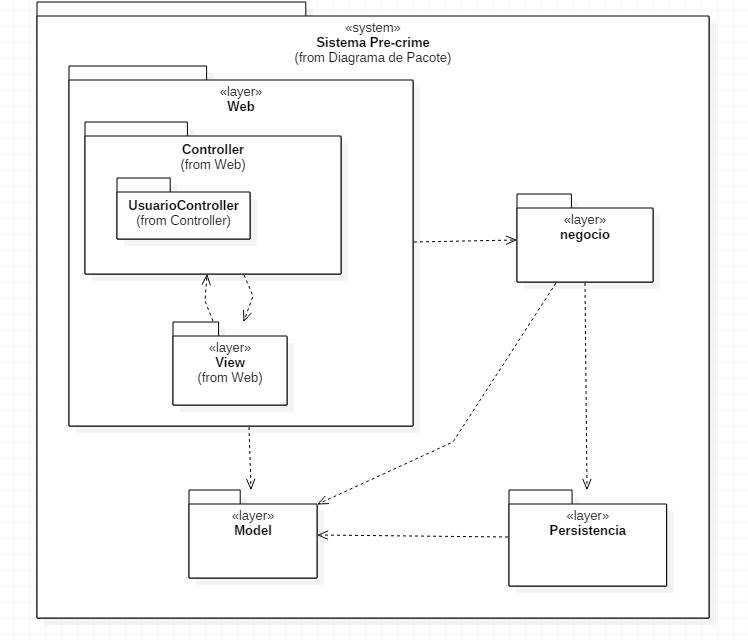
Alterar a localização afim de ver as ocorreências em outros locais.

# Visão Lógica

## Visão Geral

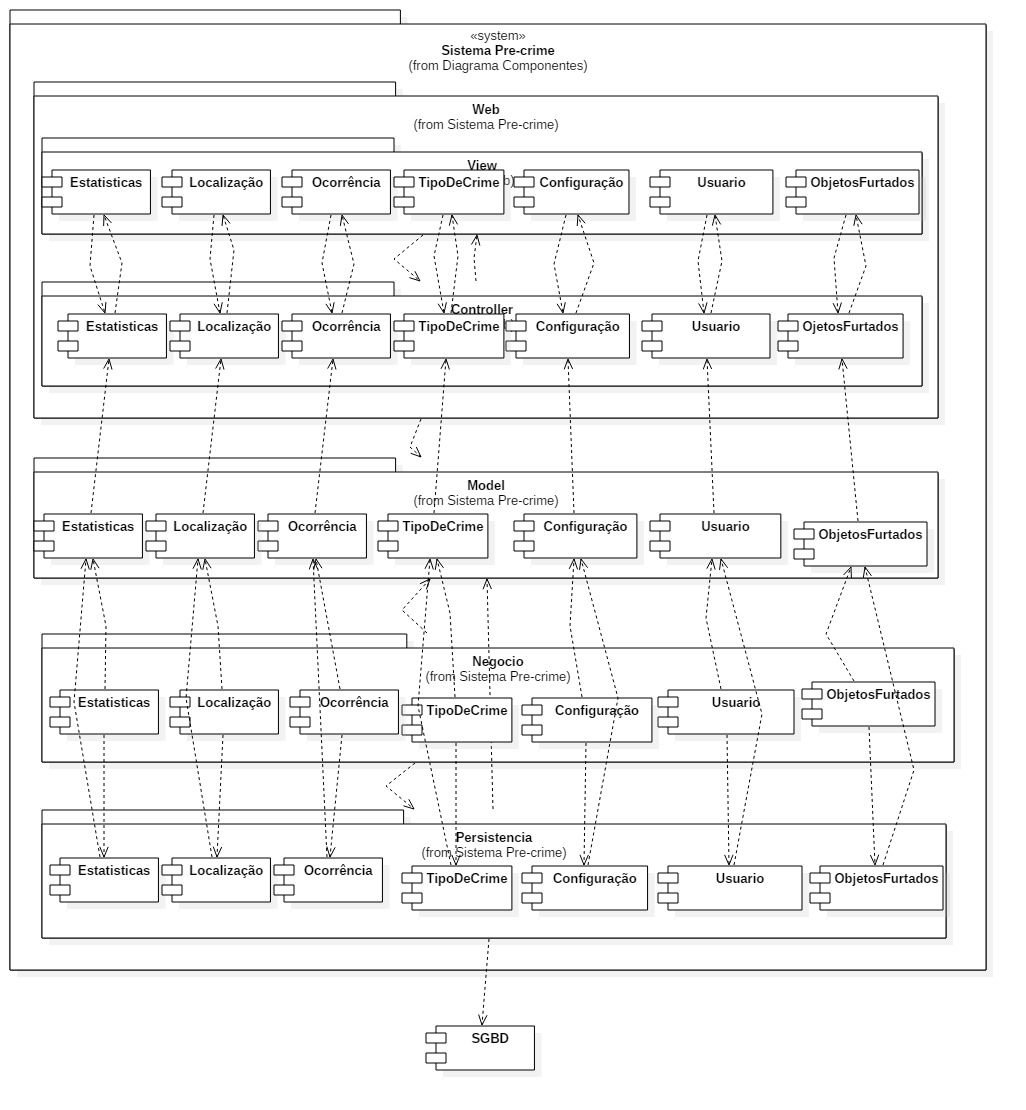
* A visão lógica do Sistema Pre-Crime é composta principalmente por 5 pacotes:
* Views: A camada view é responsável pela interação do usuário com o sistema, nessa camada são realizados etapas como construção de interface de usuário e interação para as configuração das funcionalidades.
* **Model**: Conjunto de classes que esncapsula dados e regras de negòcio que khe sãoaplicadas. Em outras palavaras, agrupa os modelos de dados que serão utilizados pela aplivação.
* **Controller:** Gerencia a interação ente o model e o View. Conjunto de classe sresposaàveis por definir os comportamentos da aplicação à nìvel de servidor. Também conhecido por ser o coração da aplicação.
* **Negócio:** Fornece Regras de Negócio.
* **Persistência:** Armazenar os objetos persistentes do sistema emalgum meio de armazenamento fisico (banco de dados, arquivo, etc.).

## Pacotes de Design Significativos do Ponto de Vista da Arquitetura



**Figura 2 – Diagrama de Pacotes**

## Diagrama de componentes

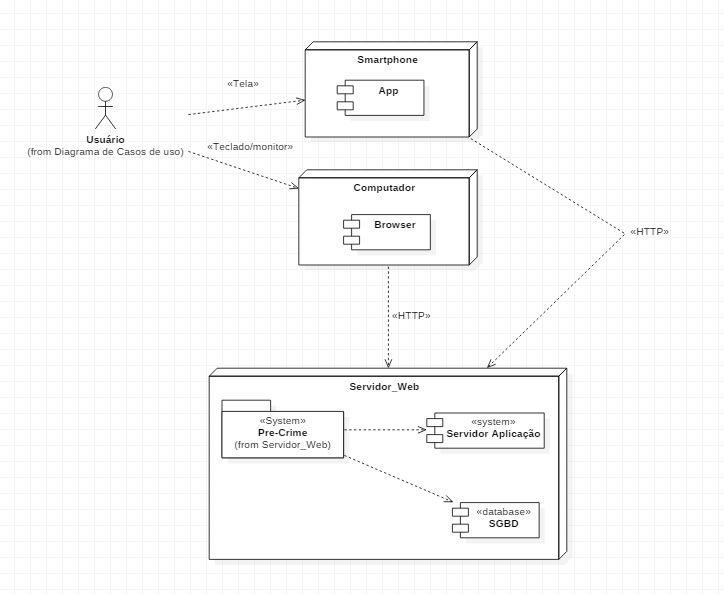


**Figura 3 – Diagrama de Componentes**

# Visão de Processos

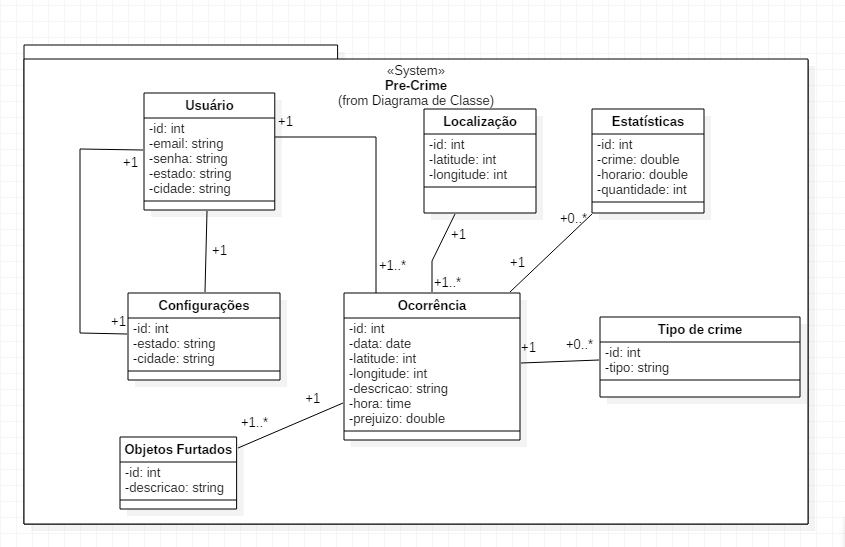
* Não se aplica

# Visão de Implantação



**Figura 4 – Diagrama de Implantação**

# Visão da Implementação



**Figura 5 – Diagrama de Classes**

# Qualidade

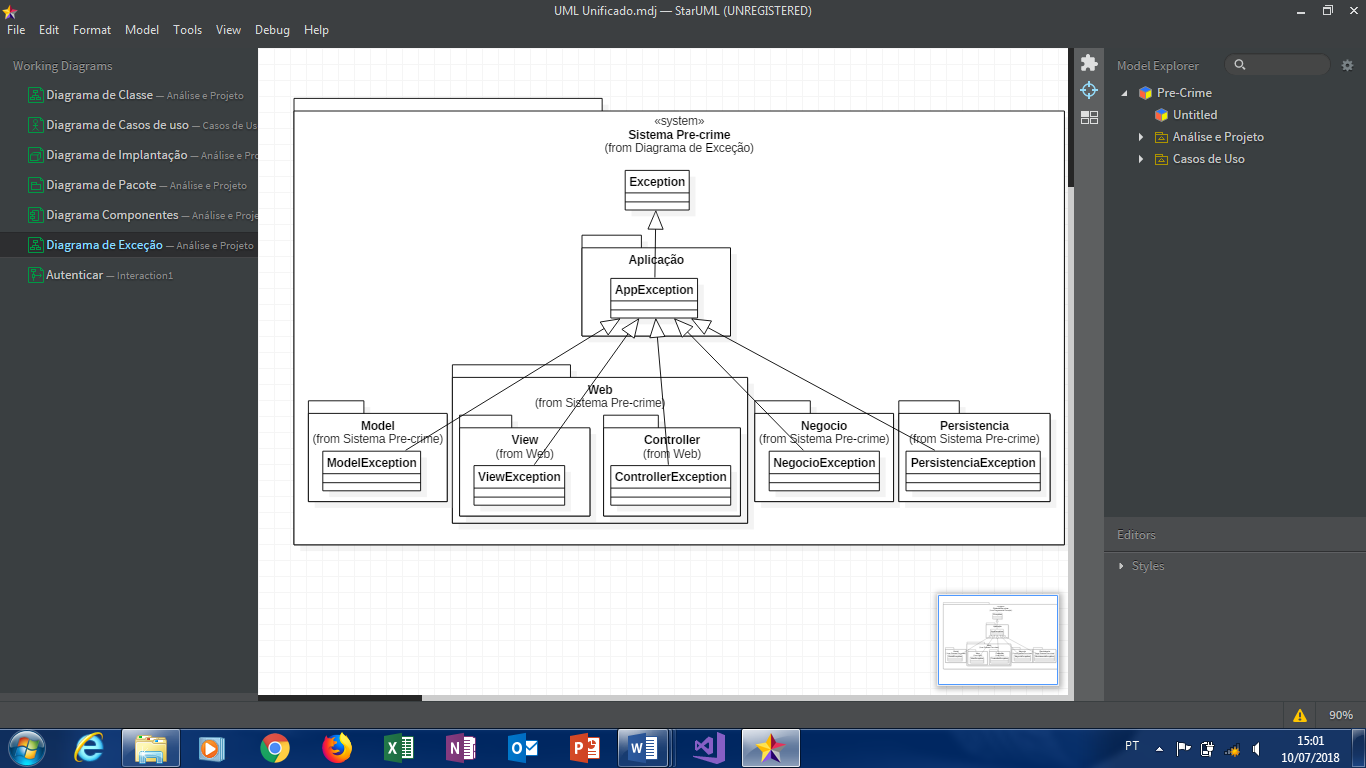
A qualidade e seus atributos é a base para as estratégias e decisões da arquitetura. O padrão utilizando para a arquitetura é o MVC Core, solução que atende a qualidade esperada para o sistema, já os requisitos não funcionais não se aplica em nosso sistema.

Especificar os requisitos não funcionais.

|  |  |
| --- | --- |
| Código | Descrição |
| RN001U | O sistema deve facilitar a realização dos cadastro de crimes |
| RN002U | Permitir que usuários solicitem Dicas de segurancas de forma rapida e sem estar logado |
| RN003U | O sistema deve assegurar que os usuarios fiquem anônimos |
| RN004U | O sistema deve funcionar em todas os sistemas operacionais dos smartphone |

# Exceções

O tratamento de exceções se dará preferencialmente em camadas. O sistema possuirá tratamento de exceções específicas definidas pela *Application.Exception* que utilizará os tratamentos de exceções genéricos definidos em *System.Exception.*



**Figura 6 – Diagrama da Arquitetura de Exceções**