#### Universidade Federal de Goiás

Sistemas de informação

Manutenção de software

**Documento de arquitetura do sistema Gestor Bases**

#### 

#### 

#### 

#### Célia

Diana

Jones

Josué

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### **Goiânia, 2019**

#### **Introdução**

#### Escopo

Definições, Acrônimos e Abreviações

#### **Stakeholders e concerns**

Stakeholders

Concerns

#### **Atributos de qualidade priorizados**

#### **Representação arquitetural**

**1.** **Introdução**

Este documento tem como objetivo apresentar a arquitetura do sistema Gestor Base.

**1.1.** **Escopo**

Este documento tem como escopo documentar as partes significativas do ponto de vista da arquitetura apresentando as principais siglas e abreviações, visão geral do sistema, stakeholders e visões detalhadas.

## **1.2.** **Definições, Acrônimos e Abreviações**

* **Framework:** Conjunto flexível e extensível de classes, interfaces, padrões e métodos que soluciona problemas recorrentes e já conhecidos no ambiente de programação do mundo real;
* **MTV:** Model - template - views
* **URL:** é a abreviatura de Uniform Resource Locator, ou Localizador Padrão de Recursos em português, designando o endereço de um recurso disponível em uma rede.

# **1.Stakeholders e concerns**

Este capítulo contém as informações sobre os interessados e os seus interesses relevantes para arquitetura do sistema Gestor Base.

## **1.1.** **Stakeholders**

Segue abaixo os stakeholders identificados para o Gestor Base:

* Analista de dados
* Gestor de TI

## **1.2.Concerns**

Segue abaixo os concerns identificados para o Gestor Base:

**2.** **Atributos de qualidade priorizados**

Segue abaixo os atributos de qualidade priorizados nesta arquitetura:

# **3.** **Representação arquitetural**

O sistema Gestor Bases faz uso do framework web Django, que é escrito em Python e usa estruturação em MTV - Model (contém as classes e descrição dos dados que serão gerenciados), template (contém as interfaces de interação com o usuário) e view (contém a lógica de negócios da aplicação), tal framework facilita a geração automática de formulários conforme o modelo de dados, URLs de maneira simples, disposição de um sistema de templates extensível e amigável dentre outras vantagens e para que haja persistência de dados é usado o banco de dados PostgreSQL que é um banco de dados relacional.

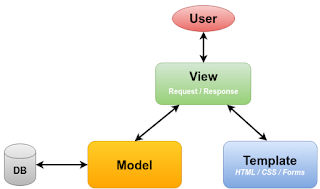


Figura 1: Arquitetura geral

**4. Referências**

**Django:** http://www.djangobrasil.org/

**Django:** https://pythonacademy.com.br/assets/ebooks/desenvolvimento-web-com-python-e-django/desenvolvimento-web-com-python-e-django.pdf

**Figura1:** https://thegeekswatch.blogspot.com/2016/08/framework-wars-django-vs-ruby-on-rails.html