

---

# Instituto Federal de Pernambuco

---

## Sisplan Documento de Arquitetura de Software

**Versão <1.0>**

*[Nota: O gabarito a seguir é fornecido para utilização com o Rational Unified Process. O texto em azul exibido entre colchetes e em itálico (style=InfoBlue) foi incluído para orientar o autor e deve ser excluído antes da publicação do documento. Um parágrafo digitado após esse estilo será automaticamente definido como normal (style=Body Text).]*

*[Para personalizar campos automáticos no Microsoft Word (que exibem um segundo plano cinza quando selecionados), selecione File>Properties e substitua os campos Title, Subject e Company pelas informações apropriadas para este documento. Depois de fechar o diálogo, os campos automáticos podem ser atualizados no documento inteiro, selecionando Edit>Select All (ou Ctrl-A) e pressionando F9 ou simplesmente clique no campo e pressione F9. Esse procedimento deverá ser executado separadamente para os Cabeçalhos e Rodapés. Alt-F9 alterna entre a exibição de nomes de campos e do conteúdo dos campos. Consulte a Ajuda do Word para obter informações adicionais sobre como trabalhar com campos.]*

Sisplan	Versão: <1.0>
Documento de Arquitetura de Software	Data: <18/10/2000>
<identificador do documento>	

## Histórico da Revisão

Data	Versão	Descrição	Autor	
<dd/mmm/aa>	<x.x>	<detalhes>	<nome>	

Sisplan	Versão: <1.0>
Documento de Arquitetura de Software	Data: <18/10/2000>
<identificador do documento>	

# Índice

1. Introdução	4
1.1 Objetivo	4
1.2 Escopo	4
1.3 Definições, Acrônimos e Abreviações	4
1.4 Referências	4
1.5 Visão Geral	4
2. Representação Arquitetural	4
3. Restrições e Metas Arquiteturais	4
4. Visão de Casos de Uso	5
5. Visão Lógica	5
5.1 Visão Geral	5
5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura	5
5.3 Realizações de Casos de Uso	5
6. Visão de Processos	5
7. Visualização da Implementação	5
8. Visão da Implementação	5
8.1 Visão Geral	5
8.2 Camadas	6
9. Visão de Dados (opcional)	6
10. Tamanho e Desempenho	6
11. Qualidade	6

Sisplan	Versão: <1.0>
Documento de Arquitetura de Software	Data: <18/10/2000>
<identificador do documento>	

# Documento de Arquitetura de Software

## 1. Introdução

No atual cenário de gestão de processos da Instituição é necessário se utilizar dois Sistemas com diferentes funcionalidades para que seja possível atender as demandas organizacionais.

### 1.1 Objetivo

O seguinte projeto visa unificar os sistemas de gestão de processos Agatha e Geplanes em um único sistema. Permitindo assim a utilização dos recursos de ambos sistemas em uma única interface.

[Esta seção define a função ou finalidade do Documento de Arquitetura de Software, na documentação completa do projeto e descreve resumidamente a estrutura do documento. O público-alvo do documento é identificado com uma indicação de como deve utilizá-lo.]

### 1.2 Escopo

O Sistema desenvolvido é composto pelas seguintes áreas:

- Nível
  - Camada
  - Iniciativas
  - Planos de Ação
    - 5W2H
  - Análises SWOT
  - Indicadores

A autenticação e dados do usuário são controlados pelo Diretório de Usuários padrão do IFPE

[Faça uma breve descrição da aplicação do Documento de Arquitetura de Software; o quê é afetado ou influenciado por este documento.]

### 1.3 Definições, Acrônimos e Abreviações

Container: Ambiente de desenvolvimento iniciado sob demanda visando a portabilidade da aplicação, diminuição de erros e unificação do ambiente utilizado entre os desenvolvedores e coim o cliente.

Docker: Gerenciador de containers.

LDAP: Protocolo Leve de Autenticação por Diretórios, permite o gerenciamento de autenticação com Diretórios de Usuários.

Django: Framework Python MVC utilizado durante o desenvolvimento do Sisplan.

MVC: Model – View – Controller, modelo arquitetural baseado em três camada (Modelo, Visualizador e Controlador).

[Esta subseção fornece as definições de todos os termos, os acrônimos e as abreviações necessários para a interpretação apropriada do Documento de Arquitetura de Software. Essas informações podem ser fornecidas em relação ao Glossário do projeto.]

### 1.4 Referências

[Esta subseção fornece uma lista completa de todos os documentos mencionados em outra parte do Documento de Arquitetura de Software. Identifique cada documento pelo seguinte: título, número do relatório (se for o caso), data e organização responsável pela publicação. Especifique as origens a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento.]

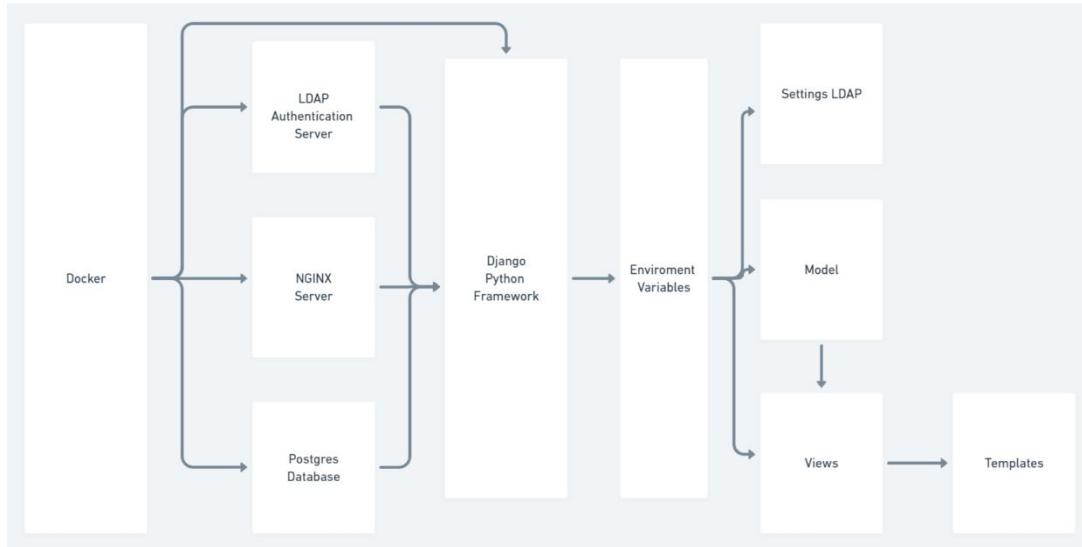
Sisplan	Versão: <1.0>
Documento de Arquitetura de Software	Data: <18/10/2000>
<identificador do documento>	

## 1.5 Visão Geral

Dentro desse documento estará listado o modo como foi implementada a solução, os components do sistema e como eles se relacionam.

*[Esta subseção descreve o que o restante do Documento de Arquitetura de Software contém e explica como o Documento de Arquitetura de Software é organizado.]*

## 2. Representação Arquitetural



Para inicialização do ambiente é utilizado o Docker, como gerenciador de containers ele permite a configuração e portabilidade do ambiente de desenvolvimento de forma simplificada. Esse ambiente é composto pelos seguintes containers:

- um servidor de autenticação LDAP, se for necessário montar ele – no caso do uso em produção é utilizado o servidor padrão do IFPE;
- um servidor nginx para aplicação;
- o sistema em Django; e
- o banco de dados em Postgres;

*[Esta seção descreve qual é a arquitetura de software do sistema atual e como ela é representada. Dos Casos de Uso, Implementação, Processo , Lógicae Visualizações de Implementação, ela enumera as visualizações necessárias e, para cada uma, explica que tipos de elementos de modelos a mesma contém.]*

## 3. Restrições e Metas Arquiteturais

*[Esta seção descreve os requisitos e objetivos do software que têm algum impacto sobre a arquitetura; por exemplo, segurança, garantia, privacidade, uso de um produto desenvolvido internamente e pronto para ser usado, portabilidade, distribuição e reutilização. Ela também captura as restrições especiais que podem ser aplicáveis, como design e estratégia de implementação, ferramentas de desenvolvimento, estrutura de equipe, planejamento, códigos de legado e assim por diante.]*

## 4. Visão de Casos de Uso

*[Esta seção lista os casos de uso ou cenários do modelo de casos de uso se eles representam alguma funcionalidade central significativa do sistema final ou se têm uma ampla cobertura arquitetural—se eles experimentam muitos elementos arquiteturais ou se enfatizam ou ilustram um ponto frágil específico da*

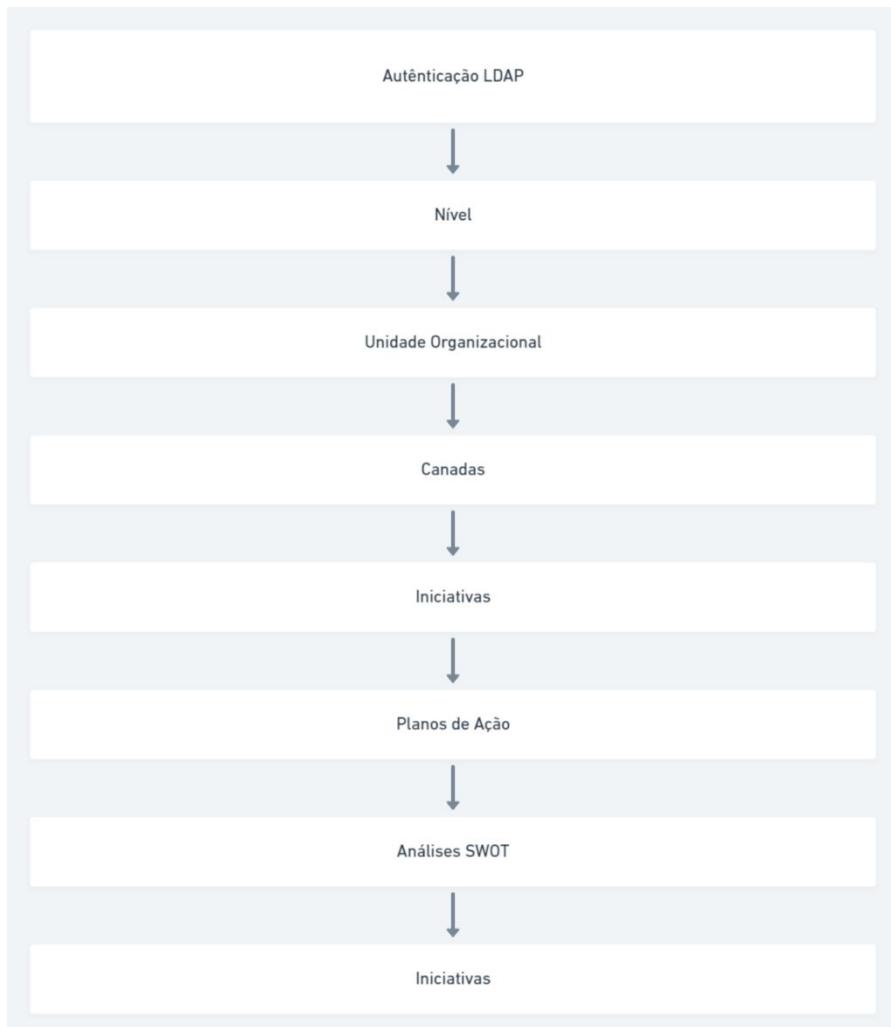
Sisplan	Versão: <1.0>
Documento de Arquitetura de Software	Data: <18/10/2000>
<identificador do documento>	

*[arquitetura.]*

## 5. Visão Lógica

*[Esta seção descreve as partes significativas do ponto de vista da arquitetura do modelo de design, como sua divisão em subsistemas e pacotes. Além disso, para cada pacote significativo, ela mostra sua divisão em classes e utilitários de classe. Você deve apresentar as classes significativas do ponto de vista da arquitetura e descrever suas responsabilidades, bem como alguns relacionamentos, operações e atributos de grande importância.]*

### 5.1 Visão Geral



*[Esta subseção descreve toda a decomposição do modelo de design em termos de camadas e de hierarquia de pacotes.]*

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

*[Para cada pacote significativo, inclua uma subseção com o respectivo nome, uma breve descrição e um diagrama com todos os pacotes e classes significativos nele contidos.]*

*[Para cada classe significativa no pacote, inclua o respectivo nome, uma breve descrição e, opcionalmente, uma descrição de algumas das suas principais responsabilidades, operações e atributos.]*

Sisplan	Versão: <1.0>
Documento de Arquitetura de Software	Data: <18/10/2000>
<identificador do documento>	

### 5.3 Realizações de Casos de Uso

[Esta seção ilustra o funcionamento do software, apresentando algumas realizações (ou cenários) de casos de uso selecionadas e explica como os diversos elementos do modelo de design contribuem para a respectiva funcionalidade.]

## 6. Visão de Processos

### 6.1 Autenticação

A autenticação é baseada em Diretórios de Usuário, permitindo assim sua utilização em conjunto com qualquer servidor de autenticação baseado nesse modelo. Utilizando o protocolo LDAP e é automaticamente configurado pelo Django utilizando o modulo django-ldap3.

### 6.2 Regras de negócio aplicadas

A utilização do sistema se baseia na criação de unidades organizacionais, que serão utilizadas como base para um Nível que pode ser planejado.

No planejamento do Nível é possível elaborar uma Camada que por sua vez pode ser aplicada em cima de Iniciativas, as Iniciativas geram Planos de Ação.

Os Planos de Ação aplicam os princípios do 5W2H e respondem no planejamento as seguintes perguntas:

- O quê?
- Por que?
- Quem?
- Onde?
- Quando?
- Como?
- Quanto?

Com o Plano de Ação definido é possível aplicar a Análise SWOT nele, nessaa etapa é definido no planejamento os seguintes pontos:

- Pontos Fortes
- Pontos Fracos
- Oportunidades
- Ameaças

Por fim, é possível com a Análise SWOT definida aplicar Indicadores a partir de analyses estatísticas.

[Esta seção descreve a decomposição do sistema em processos leves (encadeamentos 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.]

## 7. Visualização da Implementação

A implementação do Sistema é totalmente baseada na utilização de containers. Isso significa que o mesmo sistema e condições de ambientes será replicado em desenvolvimento, testes e produção.

[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. No mínimo, para cada configuração, ela deve indicar os nós físicos (computadores, CPUs) que executam o software e suas interconexões (barramento, LAN, ponto a ponto, etc.) Inclui também um mapeamento dos processos da Visualização do Processo sobre os nós físicos.]

Sisplan	Versão: <1.0>
Documento de Arquitetura de Software	Data: <18/10/2000>
<identificador do documento>	

## 8. Visão da Implementação

[Esta seção descreve a estrutura geral do modelo de implementação, a divisão do software em camadas e subsistemas no modelo de implementação e todos os componentes significativos do ponto de vista da arquitetura.]

### 8.1 Visão Geral

[Esta subseção nomeia e define as diversas camadas e o seu conteúdo, as regras que determinam a inclusão em uma camada específica e as fronteiras entre as camadas. Inclua um diagrama de componentes que mostre os relacionamentos entre as camadas.]

### 8.2 Camadas

#### 8.2.1 Controlador

Os controladores no Django são denominados ‘views’. Porém a funcionalidade é a mesma, de interceptor os dados dos modelos e submeter para as visualizações. Cada recurso utilizado no sistema possui quatro controladores geralmente: Criação, Atualização, Exclusão e Listagem.

Os recursos utilizados no sistema são:

- Unidade Organizacional
- Nível
- Camadas
- Iniciativa
- Plano de Ação
- SWOT
- Indicador

#### 8.2.2 Visualizador

O Visualizador do Django se encontra na pasta ‘template’, cada um dos recursos possui um visualizador utilizado em criação e edição do recurso. Além desses visualizadores existem os de listagem, em cada um dos recursos com exceção da Unidade Organizacional.

#### 8.2.3 Modelo

Os modelos se encontram no arquivo ‘models’ e cada um dos recursos do sistema possui um Model especificado, definindo a sua estrutura esperada em processos de criação e edição. Além disso eles são utilizados em processos de leitura para obter a formatação de leitura.

[Para cada camada, inclua uma subseção com o respectivo nome, uma lista dos subsistemas localizados na camada e um diagrama de componentes.]

## 9. 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 Design e o Modelo de Dados for trivial.]

## 10. Tamanho e Desempenho

O Django oferece bom desempenho para aplicações de diversos tamanhos. Os limites da aplicação tem de ser analisados através de testes e variam principalmente de acordo com o hardware utilizado.

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

Sisplan	Versão: <1.0>
Documento de Arquitetura de Software	Data: <18/10/2000>

## 11. Qualidade

*[Uma descrição de como a arquitetura do software contribui para todos os recursos (exceto a funcionalidade) do sistema: capacidade de extensão, credibilidade, portabilidade e assim por diante. Se essas características possuírem significado especial, como implicações de segurança, garantia ou privacidade, elas deverão ser delineadas claramente.]*