



# Princípios Arquiteturais do Projeto

Projeto [Nome do Projeto]  
Cliente [Nome do Cliente]

<b>Projeto:</b>	Projeto XXX		
<b>Responsável:</b>	[Arquiteto Responsável pelo Projeto]	<b>Versão:</b>	0.1

# Sumário

1 Histórico do documento.....	3
2 Objetivo deste Documento.....	3
3 Princípios Gerais.....	3
3.1 MVP.....	3
3.2 Segurança na camada de transporte – SSL/TLS.....	3
3.3 Fontes e Famílias Tipográfica.....	4
3.4 Codificação de Caracteres (Encoding).....	4
3.5 Internacionalização e Regionalização.....	4
3.6 Padrão de Interoperabilidade do Estado de Goiás.....	4
4 Princípios de Negócio.....	4
4.1 Acesso pelo Portal.....	4
4.2 Autenticação pelo Portal.....	5
5 Princípio de Dados.....	5
5.1 Dados da Aplicação.....	5
5.2 Banco de Dados.....	5
5.3 Dados Corporativos.....	5
6 Princípios da Aplicação.....	5
6.1 Single Sign On.....	5
6.2 Módulos da Aplicação.....	5
7 Princípios Tecnológicos.....	6
7.1 Plataformas.....	6
7.2 Ferramentas.....	6
7.3 Frameworks e Componentes.....	6
7.3.1 Frameworks Front-end:.....	6
7.3.2 Frameworks Back-end.....	7
7.3.3 Componentes.....	7
7.4 CAS.....	7
7.5 RedMine.....	7
7.6 GitLab.....	7
7.7 Jenkins.....	7

<b>Projeto:</b>	Projeto XXX		
<b>Responsável:</b>	[Arquiteto Responsável pelo Projeto]	<b>Versão:</b>	0.1

## 1 Histórico do documento

Versão	Data	Descrição	Responsável
[1.0]	[18/08/2016]	[Descrição da alteração]	[Usuário que realizou a alteração]

## 2 Objetivo deste Documento

Este documento tem como objetivo definir os princípios arquiteturais do projeto, sendo utilizado como base para o início das atividades de Arquitetura/Projeto e Implementação da aplicação.

Os princípios listados neste documento consistem nas premissas arquiteturais do projeto, baseadas nas informações conhecidas até o momento das necessidades e características do sistema a ser desenvolvido.

## 3 Princípios Gerais

*[Descreve as características gerais do projeto, como etapas do processo, escopo inicial, tipo da aplicação (web, desktop, móvel, entre outros.)]*

### 3.1 MVP

Em desenvolvimento de produtos, o MVP é o produto com o maior retorno sobre o investimento versus o risco. É o ponto entre produtos que falham no lançamento por não possuir os requisitos necessários, e produtos com muitos requisitos que diminuem o retorno sobre o investimento e aumentam o risco. No contexto do Portal essa definição é obtida através da visão dos usuários. Qual o mínimo necessário para que usuários possam acessar aplicações de forma segura em ambiente de intranet e internet, gerenciar seu perfil alterando senhas e auto cadastro. Já para os usuários gestores de sistema e analistas de sistemas, qual o mínimo necessário para que esses possam controlar como suas aplicações são apresentadas aos usuários. Acreditamos que esse escopo mínimo é apresentado nos requisitos acima na visão de arquitetura de negócio. Defendemos a utilização de uma plataforma de autenticação e autorização de renome no mercado para garantir o acesso seguro de usuários, associada a pequenas alterações em telas existentes e modularizações para favorecer alguns requisitos levantados.

### 3.2 Segurança na camada de transporte – SSL/TLS

Será utilizado os protocolos TLS/SSL para transporte de dados de forma segura entre aplicação e usuário, e entre aplicações. É recomendado sempre a utilização da versão mais atual do protocolo TLS. A infraestrutura de chaves publicas utilizada será a ICP-Brasil V3. Essa infraestrutura de chaves é utilizada somente no ambiente de produção. Logo para desenvolvimento de aplicações e homologação das mesmas

<b>Projeto:</b>	Projeto XXX		
<b>Responsável:</b>	[Arquiteto Responsável pelo Projeto]	<b>Versão:</b>	0.1

sugere-se a configuração do protocolo SSL/TLS através da API JSSE da plataforma Java. Em caso de dúvida procure a equipe de arquitetura para configuração de componentes que utilizam SSL.

### 3.3 Fontes e Famílias Tipográfica

As aplicações na SEGPLAN deverão ser construídas utilizando as fontes disponíveis na biblioteca X FreeType interface que é utilizada pelo sistema operacional Red Hat Enterprise Linux Server release 6.7 (Santiago). A lista de fontes disponíveis pode ser adquirida com a equipe de SO Unix através do comando fc-list. A instalação de novas fontes no SO para atender a aplicação web ou para utilização em relatórios gerados pela aplicação deve obedecer o licenciamento de cada fonte respectivamente.

### 3.4 Codificação de Caracteres (Encoding)

Para uma maior compatibilidade nos textos servidos via web a codificação de caracteres utilizada será a UTF-8. Em caso de códigos legados que utilizam ISO-8859-1, identificar com a equipe de Arquitetura e Tecnologia a forma de conversão do projeto ou componente para nova codificação.

### 3.5 Internacionalização e Regionalização

O País deverá ser US e língua EN (en\_US). Essa configuração é adotada devido a maior facilidade de manutenção de aplicações e identificações de problemas sem o erro de semântica ocorrido nas traduções de erros. Aplicações que façam uso da localização padrão para conversão de moedas e números devem se atentar a escolher a regionalização correta (pt\_BR).

### 3.6 Padrão de Interoperabilidade do Estado de Goiás

Os serviços REST disponibilizados pela aplicação devem seguir o padrão definido pelo Padrão de Interoperabilidade do Estado de Goiás (Pin-Goiás).

## 4 Princípios de Negócio

*[Lista os princípios envolvidos ao negócio da aplicação, como restrições, disponibilidade, etc.*

*Integração com outras aplicações corporativas (origem dos dados). Requisitos de segurança, na visão de negócio (acesso pelo portal, através de outro sistema ou base própria de usuário).]*

### 4.1 Acesso pelo Portal

A aplicação deve ser acessível através do Portal de Acesso da SEGPLAN.

<b>Projeto:</b>	Projeto XXX		
<b>Responsável:</b>	[Arquiteto Responsável pelo Projeto]	<b>Versão:</b>	0.1

## 4.2 Autenticação pelo Portal

A aplicação estará disponível para acesso pelo usuário autenticado pelo portal, sem a necessidade de uma nova validação de usuário/senha.

## 5 Princípio de Dados

*[Lista os princípios relacionados aos dados relacionados à aplicação, como a organização das informações, o tipo de banco de dados, comportamento esperado, volatilidade dos dados, tempo de armazenamento, etc.]*

### 5.1 Dados da Aplicação

Os dados da aplicação serão representados nos formatos XML e JSON.

### 5.2 Banco de Dados

O banco de dados padrão do projeto será o PostgreSQL.

### 5.3 Dados Corporativos

Os dados corporativos serão obtidos através de serviços disponibilizados pela aplicação corporativa SCORP.

## 6 Princípios da Aplicação

*[Define a organização dos pacotes e componentes da aplicação a partir do tipo dela, por exemplo, aplicação web ou desktop, dividido em camadas, (camada de interface, negócio, persistência), integração com outras aplicações corporativas (origem dos dados), requisitos de segurança, utilização de componentes ou frameworks já existentes, etc.]*

### 6.1 Single Sign On

No sistema, o protocolo utilizado para autenticação de usuários será o CAS, outros protocolos podem ser utilizado com o mesmo CAS Server caso seja necessário futuramente integrar diferentes aplicações ao CAS Server.

### 6.2 Módulos da Aplicação

A aplicação utiliza o Maven e está dividido nos seguintes módulos:

- site-web

Contém as páginas e recursos necessários para a implementação da interface com o usuário do sistema.

<b>Projeto:</b>	Projeto XXX		
<b>Responsável:</b>	[Arquiteto Responsável pelo Projeto]	<b>Versão:</b>	0.1

- api-web

Contém os serviços disponibilizados como APIs para acesso pelo módulo site-web e clientes externos.

- dominio

Contém a implementação da regra de negócio da aplicação, bem como o acesso à base de dados e outros serviços/APIs externos à aplicação.

## 7 Princípios Tecnológicos

*[Lista as tecnologias envolvidas, assim como as ferramentas utilizadas, repositório, frameworks, etc.]*

### 7.1 Plataformas

A plataforma de execução do sistema será:

- Sistema Operacional Red Hat Enterprise Linux Server release 6.7 (Santiago)
- Java HotSpot(TM) 64-Bit Server VM 1.8.0\_40
- JBoss EAP 6.4 em modo Domain

### 7.2 Ferramentas

As seguintes ferramentas serão utilizadas para o desenvolvimento do projeto:

- IntelliJ IDEA
- Maven
- Bower
- Gulp

### 7.3 Frameworks e Componentes

Os seguintes frameworks serão utilizados para o desenvolvimento da aplicação:

#### 7.3.1 Frameworks Front-end:

- Angular 2
- Bootstrap

#### 7.3.2 Frameworks Back-end

- JavaEE

<b>Projeto:</b>	Projeto XXX		
<b>Responsável:</b>	[Arquiteto Responsável pelo Projeto]	<b>Versão:</b>	0.1

- Spring Data
- Jersey 2
- Swagger 2
- JAX-RS

### 7.3.3 Componentes

- Goias CAS

## 7.4 CAS

O CAS pode ser considerado uma plataforma para Single Sign-On corporativo. A plataforma possui suporte a uma gama de protocolos de autenticação para diferentes tecnologias, podendo ser dividido em dois componentes. CAS Server; responsável por autenticar usuários por diferentes formas de autenticação e validar o acesso de usuários junto as aplicações. CAS Client, responsável por receber as credenciais dos usuário e enviar ao Cas Server para validação.

Referências:

- <http://jasig.github.io/cas/development/protocol/CAS-Protocol-Specification.html>
- <http://jasig.github.io/cas/development/protocol/CAS-Protocol.html>

## 7.5 RedMine

Ferramenta para o controle de atividades dos integrantes do projeto.

## 7.6 GitLab

Solução de repositório distribuído.

## 7.7 Jenkins

Ferramenta de integração contínua.