



**Universidade Federal de Goiás  
Instituto de Informática  
RELPREV**

**Versão do Documento 0.5**  
30/10/2013



## Histórico de Revisão

Data	Versão	Descrição	Autor
10/10/2013	0.1	Criação do documento e adição dos tópicos: 2.1, 5.2	Hálisson Bruno
10/10/2013	0.2	Formatação do documento e adição dos tópicos: 1, 5.2.	Ciro Anacleto
11/10/2013	0.3	Adequação do nome do artefato às políticas da Fábrica do INF-UFG	Danilo Guimarães
28/10/2013	0.4	Definição das tecnologias que serão utilizadas e adição da documentação da camada de segurança da aplicação	Ciro Anacleto
30/10/2013	0.5	Detalhamento da Arquitetura	Hálisson Bruno



## Sumário

### Sumário

1. Introdução	4
2. Metas e Restrições de Arquitetura	4
2.1. Metas	4
2.2. Restrições	4
3. Visão de Casos de Uso	4
4. Visão de Integração	4
4.1 Padrões de Integração	4
5. Visão Lógica	4
5.1. Visão Geral	4
5.2. Visão de Componentes	5
5.3. Visão de Banco de Dados	6
6. Visão de Processos de Sistema	6
7. Visão de Requisitos de Tamanho e Desempenho	6
8. Visão de Implementação	7
9. Visão de Implantação	7
10. Visão de Requisitos de Segurança	7
11. Outros Requisitos Não Funcionais e Especificação Suplementar	7
12. Referências	7
13. Aprovações	8



# 1. Introdução

Esse documento tem por objetivo apresentar a arquitetura do RelPrev, um sistema de software que auxiliará o Oficial de Segurança de Voo (OSV) responsável pela coleta e disponibilização dos dados, relatórios e outros elementos da prevenção de acidentes aeronáuticos nas unidades da FAB e em conformidade com o que estabelece o MCA 3-3 2012.

## 2. Metas e Restrições de Arquitetura

### 2.1. Metas

Prover uma arquitetura flexível, robusta e de fácil manutenção, que permita futuras integrações e desenvolvimento sem necessidade de muitas adaptações.

### 2.2. Restrições

## 3. Visão de Casos de Uso

*<Apresentar os principais casos de uso do sistema do ponto de vista da lógica e complexidade de negócio.>*

## 4. Visão de Integração

### 4.1. Padrões de Integração

O padrão de integração utilizado, principalmente pelo módulo webservice, é o padrão JSON, que deverá seguir o modelo pré-estabelecido. O JSON será utilizado tanto para requisições quanto para respostas, padronizando a comunicação com o serviço relprev.

Template JSON:

```
{"success": true, "data": [], "count": 0, "message": "Mensagem", "status": "status"}
```

## 5. Visão Lógica

### 5.1. Visão Geral

*<Apresentar a descrição da visão lógica da arquitetura do sistema, descrevendo os aspectos estáticos e dinâmicos da arquitetura que realizam os casos de uso mais importantes. Ou seja, descrever as classes mais importantes, sua organização em pacotes e subsistemas de serviço, e a organização desses subsistemas em camadas. Os diagramas de classe podem ser incluídos para ilustrar os relacionamentos entre as classes, os*



## 5.2. Visão de Componentes

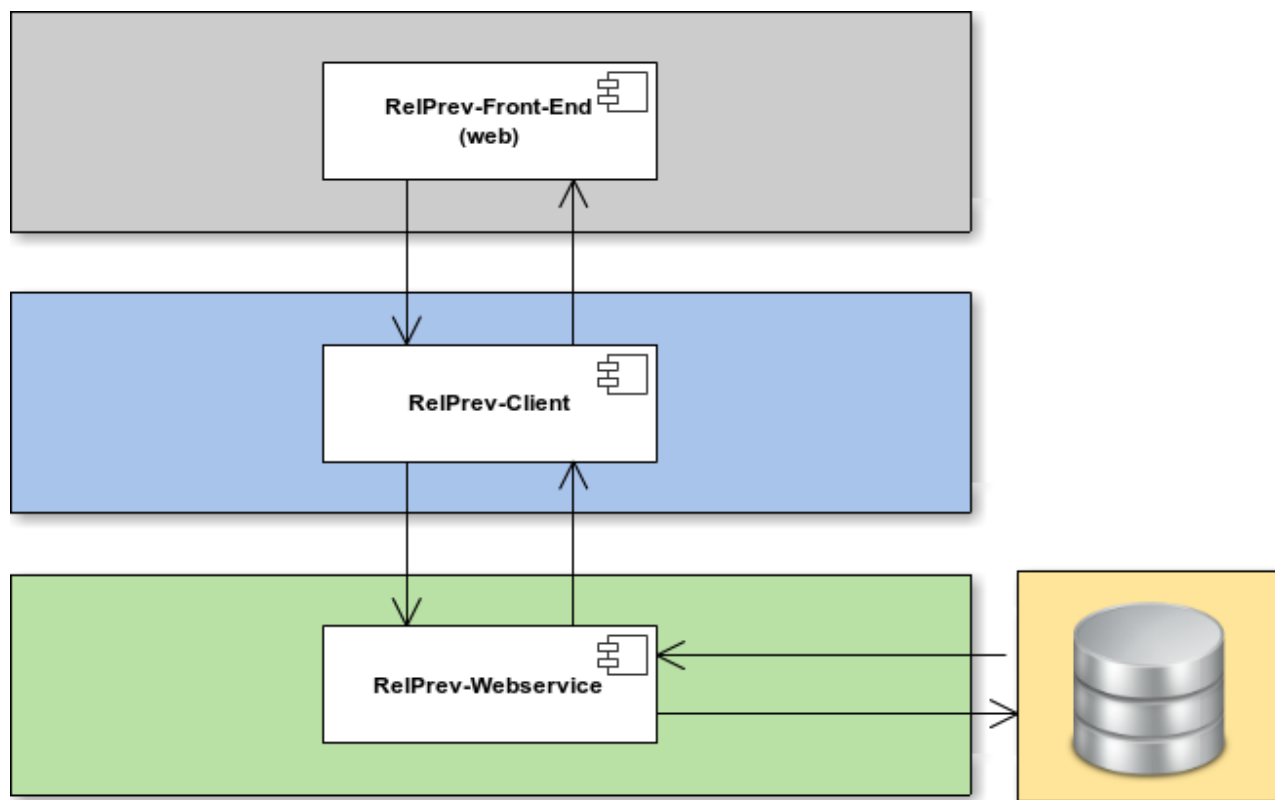


fig.1 - Componentes da arquitetura

### • RelPrev-Webservice

O módulo relprev-webservice é responsável pelo processamento de dados e persistência do sistema. O módulo utilizará princípios da arquitetura REST com o objetivo de prover uma arquitetura flexível e de fácil integração.

A comunicação com o relprev-webservice será feita exclusivamente via troca de JSON, tanto para requisições quanto para respostas, padronizando a comunicação entre os módulos.

#### > Tecnologias:

- Linguagem Java
- [Spring MVC](#)
- [Spring Security](#)
- [Spring Data](#)
- MySQL (Para a camada de Banco de dados)
- [Jackson](#)



## • RelPrev-Client

O módulo é responsável por abstrair a comunicação direta com o webservice. Ele será responsável por fornecer a interface e objetos que serão utilizados para qualquer integração com o webservice. Dessa forma não é necessário conhecer diretamente a API do webservice, para utilizar seus serviços, bastará apenas utilizar o relprev-client, que fará por exemplo a conversão do JSON vindo do serviço em objetos Java.

- Tecnologias:
  - Linguagem Java
  - Jackson

## • RelPrev-Front-End

O módulo relprev-front-end será responsável por prover uma interface gráfica para o usuário. Nela será possível preencher os relprevs, e também acessar todas as funcionalidades do sistema.

O módulo fará uso de tecnologia responsiva, para que seja acessível a um maior número de usuários, atingindo inclusive dispositivos mobile.

- Tecnologias:
  - HTML5
  - CSS3
  - [jQuery](#)
  - [Bootstrap \(twitter\)](#)
  - [chart.js](#)

## 5.3. Visão de Banco de Dados

*<Apresentar os modelos conceitual e lógico do banco de dados do sistema.>*

## 6. Visão de Processos de Sistema

*<Identificar e descrever os processos leves (threads de controle únicos) e pesados (agrupamentos de processos leves) do sistema, organizando em grupos de processos que se comunicam e descrevendo os modos principais de comunicação (tais como transmissão de mensagens e interrupções).>*

## 7. Visão de Requisitos de Tamanho e Desempenho

*<Apresentar a solução e os requisitos arquiteturais propostos para atender aos requisitos de desempenho do sistema, considerando critérios de análise como: capacidade de armazenamento, previsão de crescimento dos dados, quantidade de usuários simultâneos, quantidade de processos simultâneos para atender às demandas do sistema, localização geográfica dos usuários, volume de consulta de dados, volume de escrita de*



*dados, análise de sazonalidade na utilização do sistema e tempo máximo de resposta para as funcionalidades.>*

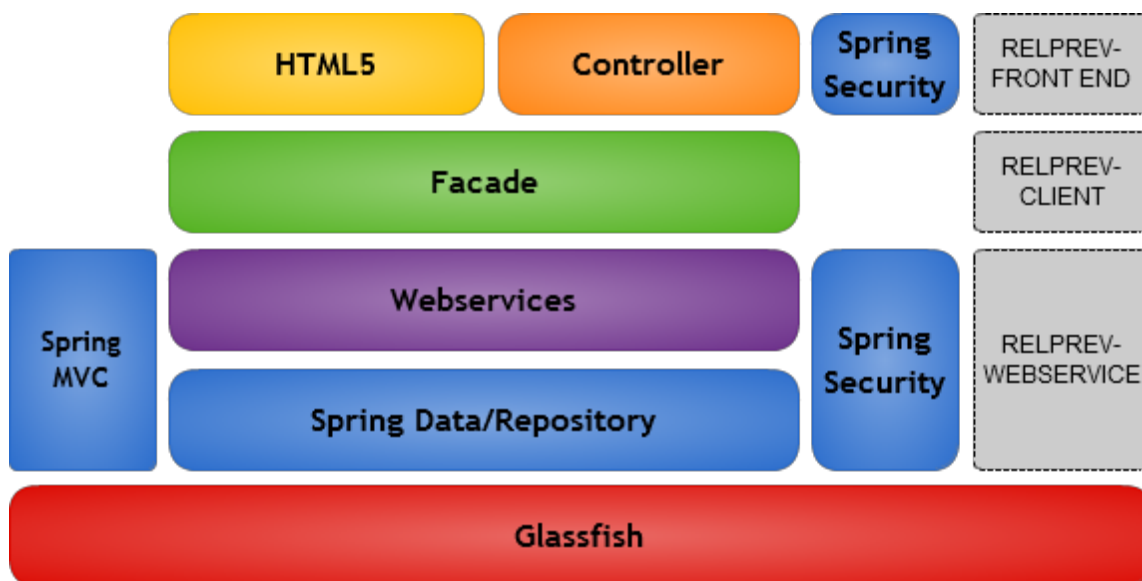
## 8. Visão de Implementação

O sistema RELPREV será desenvolvido utilizando a arquitetura orientada a serviços (SOA) onde cada funcionalidade está encapsulada por um web service. Este tipo de arquitetura provê baixa dependência entre as camadas do sistema e uma melhor manutenibilidade.

A camada de front-end (view) é responsável por prover a interface de usuário gráfica para os usuários finais do relprev. Uma das vantagens dessa arquitetura, é isolar o projeto front-end da complexidade de processamento do relprev, permitindo que o mesmo tenha apenas a responsabilidade de apresentar informações e obter entradas do usuário, sem se preocupar com demais processamentos.

A camada relprev-client fornecerá a comunicação entre o front-end e o webservice, ficando responsável exclusivamente por fornecer as ações e objetos existentes na api do webservice, abstraindo a necessidade de lidar com padrões de comunicação, integração, e etc.

Abaixo segue o diagrama de camadas a qual ilustra a estrutura a ser implementada:



img. 02 - RELPREV - Camadas da arquitetura - v0.3

## 9. Visão de Implantação

*<Esta seção é opcional e descreve uma ou mais configurações (hardware) de rede física nas quais o sistema será implantado e executado. Para cada configuração, deve-se indicar no mínimo os nós físicos (computadores, CPUs) que executam o software (processos) e as respectivas interconexões (barramento, LAN, ponto a ponto e outras). Esta visão é fundamental se o sistema for distribuído e isso acarretar implicações na arquitetura. Por exemplo, se houver um único servidor e vários clientes, a visão de implantação deve definir as responsabilidades do servidor e dos clientes como uma classe de nós; não haveria necessidade de mostrar cada nó de cliente se todos tivessem as mesmas capacidades.>*

## 10. Visão de Requisitos de Segurança



De acordo com o documento de requisitos RELPREV-022013-DRE-EOR-1.1.pdf, tópico 4.9. O sistema RELPREV possuirá controle de acesso para as funcionalidades de preenchimento de relatórios de prevenção, edição de relatórios de prevenção, adição de novos usuários, visualização dos logs da aplicação, visualização de relatórios do sistema e visualização de gráficos de estatísticas.

Os perfis de acesso (Roles) são:

- Relator
- OSV
- Supervisor
- Administrador

As senhas dos usuários serão armazenadas em banco de dados usando Hash MD5 para ocultar as informações.

Para a camada de segurança da aplicação foi escolhido a tecnologia [Spring Security](#) por ser um framework robusto, de fácil utilização, bem aceito no mercado e que provê um modelo seguro de controle de acesso baseado em perfis de acesso (Roles).

## 11. Outros Requisitos Não Funcionais e Especificação Suplementar

*<Esta seção deverá listar todos os padrões e requisitos com os quais o sistema deverá estar em conformidade e que possam ser atendidos no todo ou em parte pela arquitetura do sistema. Entre eles, destacam-se os padrões legais, padrões de comunicações, padrões de qualidade e de segurança, e-ping, e-mag, encoding, requisitos de usabilidade e acessibilidade, requisitos de confiabilidade, requisitos de hardware e software (requisitos de navegadores aplicáveis) e requisitos de documentação, dentre outros requisitos não funcionais que devam ser considerados na arquitetura do sistema.>*

## 12. Referências

- RELPREV-022013-DRE-EOR-1.1.pdf
- <http://spring.io/>
- <http://www.caelum.com.br/apostila-java-web/spring-mvc/#11-1-porque-precisamos-de-framework-ks-mvc>
- <http://www.slideshare.net/eduardo.bregaida/spring-mvc-qconsp>
- <http://www.loiane.com/2010/01/tutorial-comecando-com-spring-security/>
- <http://docs.spring.io/spring/docs/2.0.8/reference/mvc.html>
- <http://projects.spring.io/spring-security/>
- <http://jackson.codehaus.org/>
- <http://getbootstrap.com/>
- <http://jquery.com/>
- <http://www.chartjs.org/>





## 13. Aprovações

APROVAÇÃO	
Área Requisitante	Área de TI

\_\_\_\_\_  
<Nome>

<Cargo>

\_\_\_\_\_  
<Nome>

<Cargo>

\_\_\_\_\_, \_\_\_\_\_ de \_\_\_\_\_ de 20\_\_\_\_

CIÊNCIA
Contratada

\_\_\_\_\_  
<Nome>

<Cargo>

\_\_\_\_\_, \_\_\_\_\_ de \_\_\_\_\_ de 20\_\_\_\_