Documento de Arquitetura

**I9 Educatio**

Cliente: Escola Santa Monica



Responsável: Rodrigo Sales

Histórico de Alterações (deste template)

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Descrição | Autor |
| 03/11/2011 | 01.00-D01 | Iniciação do Documento de Arquitetura | Rodrigo Sales |
| xx/xx/xxxx | 01.00-D02 |  |  |
| xx/xx/xxxx | 01.00-D03 |  |  |
| xx/xx/xxxx | 01.00-D04 |  |  |
| xx/xx/xxxx | 01.00 |  |  |

Lista de Aprovadores

|  |  |
| --- | --- |
| Nome | Cargo |
|  |  |
|  |  |
|  |  |

**Conteúdo**

[1. Introdução 2](#_Toc174941281)

[1.1 Propósito 2](#_Toc174941282)

[1.2 Público Alvo 2](#_Toc174941283)

[1.3 Convenções, termos e abreviações 2](#_Toc174941284)

[2. Representação Arquitetural 2](#_Toc174941285)

[3. Objetivos e Restrições Arquiteturais 2](#_Toc174941286)

[3.1 Requisitos de Relevância Arquitetural 2](#_Toc174941287)

[4. Visão Lógica 2](#_Toc174941288)

[4.1 Módulos 2](#_Toc174941289)

[4.1.1 <Módulo X> 2](#_Toc174941290)

[4.2 Componentes 2](#_Toc174941291)

[4.2.1 <Componente X> 2](#_Toc174941292)

[5. Visão de Execução 2](#_Toc174941293)

[5.1 Processos 2](#_Toc174941294)

[5.1.1 <Processo XYZ> 2](#_Toc174941295)

[6. Visão de Implementação 2](#_Toc174941296)

[6.1 Padrões Arquiteturais 2](#_Toc174941297)

[6.1.1 <Padrão XYZ> 2](#_Toc174941298)

[6.2 Padrão de Codificação 2](#_Toc174941299)

[6.3 Estrutura de Diretórios do Código Fonte 2](#_Toc174941300)

[6.4 Bibliotecas e *Frameworks* 2](#_Toc174941301)

[6.5 Interfaces e Integrações 2](#_Toc174941302)

[6.6 Boas Práticas 2](#_Toc174941303)

[6.7 Ambiente de Desenvolvimento 2](#_Toc174941304)

[7. Visão Física 2](#_Toc174941305)

[7.1 Ambiente de Produção 2](#_Toc174941306)

[7.1.1 Hardware 2](#_Toc174941307)

[7.1.2 Software 2](#_Toc174941308)

[7.2 Ambiente de Homologação 2](#_Toc174941309)

[7.3 Ambiente de Desenvolvimento 2](#_Toc174941310)

[8. Cenários 2](#_Toc174941311)

[9. Referências 2](#_Toc174941312)

# Introdução

Este documento especifica a arquitetura do sistema I9 Educatio, descrevendo seus principais padrões, módulo, componentes, *frameworks* e integrações.

## Propósito

O propósito deste documento é dar uma visão de alto nível da solução técnica a ser seguida, enfatizando os componentes e *frameworks* a serem reutilizados e desenvolvidos, além das interfaces e integrações entre os mesmos.

## Público Alvo

Esse documento se destina à equipe do projeto, como referência consolidada para o desenvolvimento e evolução do sistema, e ao cliente, como principal ponto de visibilidade da solução técnica a ser seguida.

## Convenções, termos e abreviações

Esta seção explica o conceito de alguns termos importantes que serão mencionados no decorrer deste documento. Estes termos são descritos na tabela a seguir, estando apresentados por ordem alfabética.

|  |  |
| --- | --- |
| Termo | Descrição |
| Componente | Elemento de software reusável, independente e com interface pública bem definida, que encapsula uma série de funcionalidades e que pode ser facilmente integrado a outros componentes. |
| COTS | Do inglês *Commercial off-the-shelf*, são artefatos de software ou hardware prontos para uso. Podem ser obtidos por compra, aluguel ou licença de uso geral. Componentes e bibliotecas FOSS (Free Open-Source Software) também são considerados COTS. |
| Módulo | Agrupamento lógico de funcionalidades para facilitar a divisão e entendimento de um software. |

# Representação Arquitetural

*<Descreva nesta seção o modelo adotado para a representação da arquitetura do sistema. O framework “4+1” é utilizado como base, mas algumas das visões podem ser omitidas de acordo com o contexto do projeto.*

*As seções específicas de cada visão indicarão quais delas podem ser omitidas e em que situações isso pode ocorrer. O texto abaixo deve ser ajustado para refletir as opções seguidas em cada projeto, indicando quais visões estão sendo representadas e quais foram omitidas.>*

*http://marcomendes.com/blog/2011/08/o-modelo-de-visualizacao-41-para-a-modelagem-arquitetural/*

A especificação da arquitetura do sistema I9 Educatio segue o *framework* “4+1” [1], que define um conjunto de visões, como ilustrado na Figura 1. Cada uma dessas visões aborda aspectos de relevância arquitetural sob diferentes perspectivas:

* A **visão lógica** apresenta os elementos de projeto significantes para a arquitetura adotada e os relacionamentos entre eles. Entre os principais elementos estão módulos, componentes, pacotes e classes mais importantes da aplicação;
* A **visão de execução** apresenta os aspectos de concorrência e sincronização do sistema, alocando os elementos da visão lógica para processos, *threads* e tarefas de execução;
* A **visão de implementação** aborda os aspectos relativos à organização do código fonte do sistema, padrões arquiteturais utilizados e orientações e normas para a desenvolvimento do sistema;
* A **visão física** apresenta o hardware envolvido e o mapeamento dos elementos de software para os elementos de hardware nos ambientes do sistema.
* Os **cenários** apresentam um subconjunto dos casos de uso significativos do ponto de vista da arquitetura.

Visão de Execução

Visão Lógica

Visão de Implementação

Visão Física

Cenários

Figura . Visões do *framework* “4+1”

# Objetivos e Restrições Arquiteturais

*<Nesta seção, devem ser descritos os objetivos que se pretende atingir com a arquitetura definida e as restrições do ambiente com impacto na definição dessa arquitetura.*

*Objetivos comuns: portabilidade, desempenho, escalabilidade, tolerância a falhas etc. Tais objetivos normalmente derivam dos requisitos não funcionais do sistema.*

*Restrições comuns: limitação de memória, limitação de banda, perfil da equipe de desenvolvimento, ambiente de produção, integração com sistemas legados etc. Tais restrições normalmente derivam das restrições e premissas descritas no Documento de Requisitos do Cliente e do contexto no qual o sistema será desenvolvido.*

*Iniciar descrevendo os objetivos a serem atingidos e as principais restrições de forma textual. Depois, listar os requisitos (mais comumente os não funcionais) com impacto na definição da arquitetura.*

*Alterar o texto abaixo.>*

A arquitetura definida tem como principal objetivo tornar o sistema...

## Requisitos de Relevância Arquitetural

*<Devem ser listados os requisitos que impactam na arquitetura e o impacto de cada um deles. Adicionalmente, o tratamento dado a cada um deles na arquitetura deve ser indicado.*

*Comumente, os requisitos não funcionais, as restrições e as premissas descritos no Documento de Requisitos do Cliente impactam mais diretamente na arquitetura do sistema.*

*Alterar o texto abaixo, se necessário.>*

Esta seção lista os requisitos com impacto na arquitetura do sistema e o tratamento dado a cada um deles.

|  |  |  |
| --- | --- | --- |
| **Requisito** | **Impacto** | **Tratamento** |
| <ID do requisito> | <Impacto na arquitetura> | <Tratamento dado na arquitetura> |
| <RNF 005> | <Número de acessos ao serviço XYZ deve ser minimizado> | <Será implementado um mecanismo de *caching* baseado nos parâmetros passados ao serviço XYZ> |
| <REST 001> | <Consumo de memória deve ser minimizado> | <O número de camadas do sistema será reduzido para diminuir a necessidade de objetos intermediários> |

# Visão Lógica

*<Descrever aqui os elementos de projeto significantes para a arquitetura adotada, incluindo módulos, componentes, pacotes e classes mais importantes da aplicação (em nível crescente de granularidade). Os relacionamentos entre os elementos devem ser descritos também.*

*Iniciar com um diagrama de alto nível da organização do sistema e depois fornecer mais detalhes sobre os principais elementos e as interfaces entre os mesmos.*

*Aspectos a serem lembrados na definição da arquitetura, de acordo com o contexto do projeto: segurança, apresentação, lógica de negócio, persistência, integração, distribuição etc..*

*Alterar o texto abaixo, se necessário.>*

Esta seção apresenta a organização do sistema do ponto de vista funcional. Os principais elementos, como módulos e principais componentes são especificados. As interfaces entre esses elementos também são especificadas. A Figura 2 ilustra a arquitetura lógica do sistema.

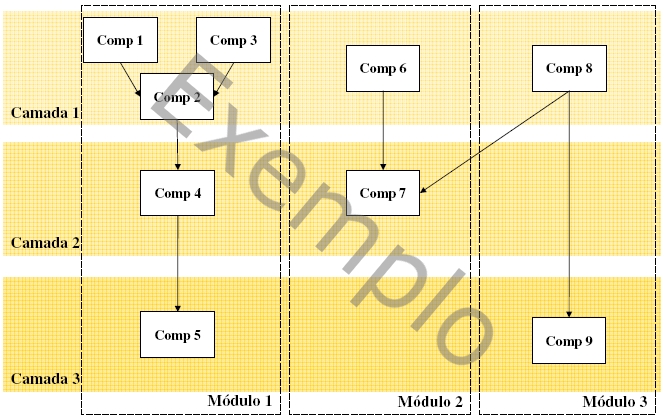


Figura . Visão geral da arquitetura

## Módulos

*<Opcionalmente, caso o sistema seja dividido em módulos, listar os módulos e os requisitos atendidos (em geral, requisitos funcionais descritos no Documento de Requisitos do Cliente) por cada um deles.*

*Esse nível de divisão é o de menor granularidade e ao mesmo tempo de maior abstração. Nesse caso, reuso é mais difícil de ser obtido devido ao nível de especialização dos módulos.*

*Alterar o texto abaixo, se necessário.>*

Esta seção descreve os principais módulos da aplicação, enfatizando suas responsabilidades e funcionalidades, assim como os requisitos atendidos.

### <Módulo X>

*<Descrever brevemente o módulo, indicando os requisitos atendidos.*

*Repetir esta subseção para cada um dos módulos do sistema.*

*Alterar o texto abaixo.>*

O módulo <Módulo X> agrupará as funcionalidades relativas ao...

|  |  |
| --- | --- |
| **Responsabilidades** | <Responsabilidades do módulo>. |
| **Requisitos Atendidos** | <Requisitos atendidos pelo módulo>. |

## Componentes

*<Os principais componentes do sistema devem ser especificados nesta seção. A especificação provida nesta seção é de alto nível e tem por objetivo servir de base para a especificação dos componentes na fase de projeto.*

*Para cada componente, deve ser justificado se o mesmo será reusado, adquirido como COTS (ambos tratados como reuso), ou desenvolvido pela equipe do projeto. Caso haja necessidade de aquisição, o Gerente de Projeto deverá ser envolvido.*

*Caso a decisão seja por usar COTS é necessário que um estudo de viabilidade seja realizado considerando os seguintes requisitos:*

1. *Funcionalidade, performance, qualidade e segurança*
2. *Termos e condições de garantia*
3. *Riscos*
4. *Responsabilidades do provedor de dar suporte a manutenção*

*Quando houver mais de uma alternativa e não houver decisão semelhante na Base de Conhecimento da empresa, o sub-processo de Tomada de Decisão deve ser utilizado. Os critérios para a decisão devem ser baseados nos objetivos e restrições arquiteturais descritos na seção 3. No caso de COTS, aspectos como tipo de licença, custo, nível de suporte e nível de maturidade devem ser observados.*

*É necessário ainda identificar se o componente será desenvolvido com o objetivo de ser reusado – o que possui um custo elevado e maiores preocupações com relação a sua modelagem. Caso não exista o objetivo de desenvolver um componente reusável, é necessário justificar o motivo desta decisão.*

*Caso deseje, pode-se optar por listar os componentes agrupando-os nos módulos descritos na subseção anterior (Módulos). Neste caso, esta subseção deve ser removida e replicada para cada módulo da aplicação.*

*Alterar o texto abaixo.>*

Esta seção descreve os principais componentes do sistema <Nome do Sistema>.

### <Componente X>

*<Descrever brevemente o componente, indicando sua origem e versão, caso esteja sendo reusado.*

*Repetir esta subseção para cada um dos componentes do sistema.*

*Alterar o texto abaixo.>*

O componente <Componente X> será responsável por...

|  |  |
| --- | --- |
| **Responsabilidades** | <Responsabilidades do componente>. |
| **Requisitos Atendidos** | <Requisitos atendidos pelo componente>. |
| **Justificativa** | *<É obrigatório justificar a escolha do componente sobre outras opções disponíveis. Este campo deve ser preenchido com uma das seguintes opções:*   * ***Única opção****:**O componente será utilizado por ser a única opção identificada para atender o requisito;* * ***Padrão da empresa****: O componente será utilizado por já ser padrão na empresa;* * ***Tomada de Decisão****: Escolha realizada através do subprocesso de Tomada de Decisão. Neste caso deve-se criar um link para a planilha com a decisão.>* |
| **Será reusado?** | <Sim/Não>.  <Justificativa para reusar ou não o componente>. |
| **Origem** | <Origem e versão do componente a ser reusado. Quando o componente não for reusado, deve-se colocar NA.>. |
| **Será reusável?** | <Sim/Não>.  <Justificativa para desenvolver o componente de forma que este possa ser reusado por outros projetos>. |

# Visão de Execução

*<Opcionalmente, caso o sistema tenha uma execução complexa, envolvendo diferentes processos e threads, e questões de concorrência e sincronização forem relevantes, especificar o mapeamento dos elementos lógicos nos elementos de execução. Caso o sistema não se enquadre nestas características, substituir o conteúdo da seção por NA.*

*Iniciar com um diagrama que mostre os principais elementos da arquitetura lógica sendo mapeados para os processos e threads do sistema. Caso seja necessário, detalhar os processos e threads do sistema textualmente.É sugerida a criação de diagramas de seqüência ou de estados adicionais para detalhar de forma mais clara as interações entre os processos e/ou threads.>*

Esta seção apresenta a o mapeamento dos elementos lógicos da arquitetura nos processos e *threads* de execução do sistema. A Figura 3 ilustra esse mapeamento.

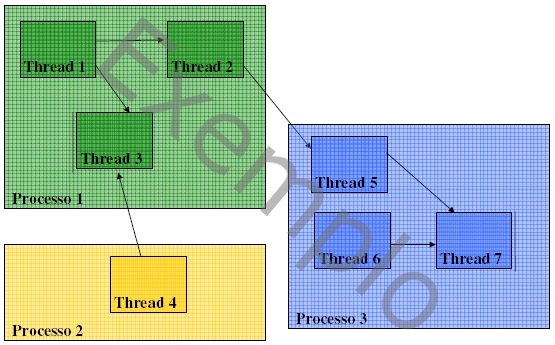


Figura . Diagrama de execução do sistema

## Processos

*<Opcionalmente, listar os processos do sistema, observando aspectos de redundância e sincronização.>*

Esta seção lista os processos do sistema, juntamente com suas necessidades de redundância e sincronização com demais processos.

|  |  |  |
| --- | --- | --- |
| **Processo** | **Instâncias** | **Sincronização** |
| <Nome do Processo> | <Quantidade de processos (1, se não tiver redundância)> | <Lista dos processos com os quais deve sincronizar> |

### <Processo XYZ>

*<Opcionalmente, listar as threads de execução do processo, observando aspectos de sincronização.*

*Repetir esta subseção para os demais processos, se necessário.>*

|  |  |  |
| --- | --- | --- |
| ***Thread*** | **Instâncias** | **Sincronização** |
| <Nome da *Thread*> | <Quantidade de *threads* (1, se não for pooled)> | <Lista das *threads* com os quais deve sincronizar> |

# Visão de Implementação

*<Nesta seção devem ser dadas as orientações aos desenvolvedores para o projeto e implementação do sistema. Padrões e normas de codificação, padrões arquiteturais, boas práticas, ambiente de desenvolvimento, bibliotecas e frameworks urilizados devem ser descritos.*

*Opcionalmente, pode ser definido um Guia de Desenvolvimento que conterá algumas dessas orientações, ficando aqui apenas uma referência para as mesmas.*

*É fortemente recomendada a referência aos guias existentes nos sub-processos de Análise.*

*Opcionalmente, iniciar com um diagrama da visão de implementação, que adicionará detalhes de tecnologias e padrões para tratamento de aspectos não funcionais à visão lógica da arquitetura.*

*Alterar o texto abaixo, se necessário.>*

Esta seção traz as orientações para o projeto e implementação do sistema de acordo com a arquitetura estabelecida. A Figura 4 ilustra a visão de implementação da arquitetura do sistema.

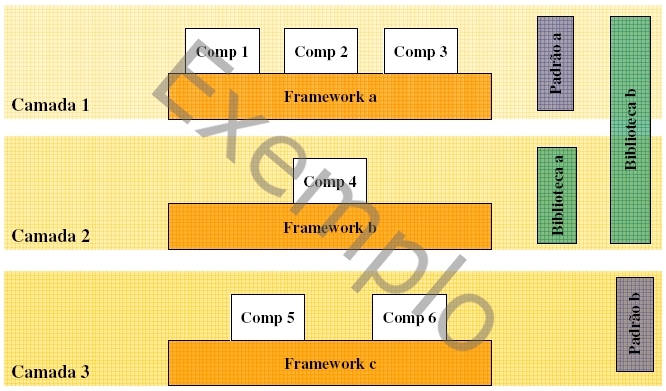


Figura . Visão de implementação da arquitetura do sistema.

## Padrões Arquiteturais

*<Descreva nesta seção todos os padrões arquiteturais a serem usados no sistema, justificando a aplicação de cada um deles. Opcionalmente, adaptações ao padrão e exemplos de uso no sistema podem ser descritos.*

*Caso não haja adaptações nem exemplos de uso, pode ser utilizada a tabela abaixo. Do contrário, seguir o estilo de uma subseção para cada padrão.*

*Todos os padrões usados devem estar catalogados no Catálogo de Padrões da empresa. Caso algum novo padrão necessite ser catalogado, o mesmo deve ser submetido ao SEPG.*

*Aspectos a serem lembrados na definição dos padrões arquiteturais, de acordo com o contexto do projeto: segurança, apresentação, lógica de negócio, persistência, tratamento de erros, logging, controle de sessões, controle de transações, controle de concorrência, integração, distribuição etc.>*

|  |  |
| --- | --- |
| **Padrão** | **Justificativa** |
| <Nome do Padrão> | <Justificativa para seu uso. *Se aplicável, justificar também o porque da não utilização de outro padrão que aborda o mesmo problema.*> |

### <Padrão XYZ>

*<Descrever brevemente como e onde o padrão será utilizado na arquitetura do projeto.*

*Repetir esta subseção para cada um dos padrões arquiteturais do sistema.>*

#### Adaptações

*<Descrever as adaptações necessárias ao padrão no contexto do sistema. Opcionalmente, esta seção pode conter diagramas ou trechos de código que exemplifiquem as adaptações.>*

#### Exemplos de Uso

*<Opcionalmente, esta seção pode trazer alguns cenários de uso do padrão no sistema.>*

## Padrão de Codificação

*<Descrever o padrão de codificação a ser seguido no projeto.*

## Estrutura de Diretórios do Código Fonte

*<Descrever o padrão de estrutura de diretórios do sistema, mapeando os diretórios dos principais elementos da arquitetura. Esta estrutura de diretórios diz respeito apenas à estrutura do código-fonte, e tipicamente começa a partir do diretório “src”. As demais estruturas de diretórios encontram-se descritas no Plano de Gerência de Configuração.*

*Para aplicações Java, também se deve descrever a forma como será realizado o empacotamento dos componentes. Tipicamente o empacotamento dos componentes segue o prefixo* ***org.<nome da empresa>.<nomeAplicacao>.<nomeComponente>****. É sugerido ainda que sejam criados mais dois pacotes após a estrutura anterior, de nomes* ***api*** *e* ***impl,*** *para agrupar respectivamente as classes “públicas” e a classes “privadas” do componente.*

## Bibliotecas e *Frameworks*

*< Bibliotecas e frameworks formam a infra-estrutura, provendo funcionalidades básicas, na qual o sistema será baseado. Os principais frameworks e bibliotecas do sistema devem ser especificados nesta seção. A especificação provida nesta seção é de alto nível e tem por objetivo servir de base para a especificação dos mesmos na fase de projeto.*

*Para cada framework/biblioteca, deve ser justificado se o mesmo será reusado ou adquirido como COTS (ambos tratados como reuso), ou desenvolvido pela equipe do projeto. Caso haja necessidade de aquisição, o Gerente de Projeto deverá ser envolvido.*

*Caso a decisão seja por usar COTS é necessário que um estudo de viabilidade seja realizado considerando os seguintes requisitos:*

1. *Funcionalidade, performance, qualidade e segurança*
2. *Termos e condições de garantia*
3. *Riscos*
4. *Responsabilidades do provedor de dar suporte a manutenção*

*Quando houver mais de uma alternativa e não houver decisão semelhante na base de conhecimento da empresa, o sub-processo de Tomada de Decisão deve ser utilizado. Os critérios para a decisão devem ser baseados nos objetivos e restrições arquiteturais descritos na seção 3. No caso de frameworks/bibliotecas adquiridos como COTS, aspectos como tipo de licença, custo, nível de suporte e nível de maturidade devem ser observados.>*

Esta seção descreve as bibliotecase os *frameworks* utilizados pelo sistema <Nome do Sistema>.

|  |  |  |  |
| --- | --- | --- | --- |
| **Biblioteca / *Framework*** | **Justificativa** | **Versão** | **Ambiente** |
| <Nome da biblioteca/*framework*> | *<Este campo deve ser preenchido com uma das seguintes opções:*   * ***Única opção****:**A biblioteca será utilizada por ser a única opção identificada para atender o requisito;* * ***Padrão da empresa****: A biblioteca será utilizada por já ser padrão na empresa;* * ***Tomada de Decisão****: Escolha realizada através do subprocesso de Tomada de Decisão. Neste caso deve-se criar um link para a planilha com a decisão.>* | <Versão a ser utilizada na aplicação> | <Ambiente onde a biblioteca será utilizada. Ex.: Todos, Produção, Homologação ou Desenvolvimento.> |

## Interfaces e Integrações

*< Detalhar obrigatoriamente todas as interfaces com sistemas legados, externos, componentes / módulos compartilhados entre equipes de um mesmo projeto ou entre projeto, e qualquer outra interface cuja especificação não se encontre sobre o controle da equipe do projeto. A adição de outras interfaces além destas, tipicamente interfaces existentes internamente no sistema, fica a cargo do autor do documento. Caso não existam interfaces com as características listadas, substituir o conteúdo da seção por NA. A coluna Tipo deve ser preenchida com os valores “I” para interfaces internas do sistema, ou “E” para interfaces externas ao sistema.*

*Exemplos de interfaces especiais: interfaces através de web services, rmi, rpc etc. Exemplos de integrações: protocolo de acesso a device drives, acesso a sistemas legados etc. >*

Esta seção descreve as interfaces especiais e integrações a serem realizadas pelo sistema.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Interface** | **Componentes** | **Tipo** | **Tecnologias** | **Documentação Adicional** |
| <# da interface> | <Componentes envolvidos na interface / integração> | <*I* ou *E* > | <Tecnologias envolvidas na interface / integração> | <Referências para a documentação adicional da interface / integração, se disponível> |
| <1> | <C1 e C2> | <I> | <Web services> | <Referência para wsdl> |
| <2> | <C5, C6 e C9> | <E> | <Protocolo proprietário XYZ> | <Referência para a especificação do protocolo XYZ> |

## Boas Práticas

*<Descrever as práticas recomendadas para o desenvolvimento do sistema.*

## Ambiente de Desenvolvimento

*<Descrever o as ferramentas necessárias para a fase de projeto e implementação do sistema. Tais ferramentas devem estar listadas no Plano de Projeto, sendo aqui detalhadas suas características e a forma como as mesmas serão utilizadas no projeto.*

*Alterar o texto abaixo.>*

O ambiente de desenvolvimento será composto por...

# Visão Física

*<Opcionalmente, caso haja distribuição física no sistema, o mapeamento dos elementos de software para o hardware é descrito nesta seção. Minimamente, o ambiente de produção deve ser atendido. Os ambientes de homologação e desenvolvimento podem ser opcionalmente descritos. Caso o sistema não se enquadre nestas características, substituir o conteúdo da seção por NA >*

Esta seção descreve os elementos de hardware do sistema e o mapeamento dos elementos de software para os mesmos.

## Ambiente de Produção

*<Descrever o ambiente de produção e o mapeamento dos elementos de software para o hardware do sistema, focando nos processos que estarão sendo executados em cada um dos hardwares do sistema. Caso exista um Plano de Implantação para a aplicação, não será necessário detalhar nesta seção o hardware e software utilizado em cada ambiente já existente no Plano de Implantação, sendo suficiente apenas descrever a distribuição dos elementos de software para o hardware existente. Neste caso, deve-se referenciar o Plano de Implantação na seção Referências.>*

Esta seção descreve o ambiente de produção do sistema e o mapeamento dos elementos de software para o hardware disponível. A Figura 5 ilustra a visão física do ambiente de produção.

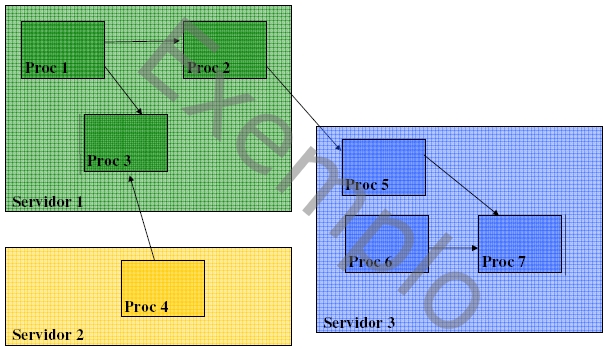


Figura . Ambiente de produção do sistema

### Hardware

*<Descrever o hardware do ambiente de produção.>*

### Software

*<Descrever o software do ambiente de produção.>*

## Ambiente de Homologação

*<Opcionalmente, descrever o ambiente de homologação e o mapeamento dos elementos de software para o hardware do sistema.>*

## Ambiente de Desenvolvimento

*<Opcionalmente, descrever o ambiente de desenvolvimento e o mapeamento dos elementos de software para o hardware do sistema.>*

# Cenários

*<Opcionalmente, caso sinta necessidade de detalhar os casos de descrever o conjunto de casos de uso com maior relevância arquitetura, pode-se detalhá-los nesta seção. Caso contrário deve-se apenas mencionar que estes casos de uso já se encontram descritos no Documento de Requisitos do Produto, e suas realizações serão detalhadas no Modelo de Projeto.*

*Podemos definir que os casos de uso com relevância arquitetural são aqueles que representam uma funcionalidade central e significativa da aplicação, ou possuem uma cobertura arquitetura substancial, ou ainda ilustram um determinado ponto complicado da arquitetura.*

*Caso opte por detalhar estes casos de uso, deve-se atentar a quantidade de informações replicadas. As informações que devem ser descritas nesta seção são o nome do caso de uso, seu diagrama, um texto descritivo e sua realização.>*

Os cenários da aplicação já se encontram descritos no Documento de Requisitos do Produto [?] e suas realizações serão detalhas no Modelo de Projeto [?].

# Referências

<Esta seção deve prover uma lista de todos os documentos relacionados a este documento.>

1. Architectural Blueprints—The “4+1” View Model of Software Architecture; IEEE Software 12; Novembro 1995, p. 42-50;
2. Padrão de Codificação Java;
3. Padrão de Codificação C/C++;
4. Padrão de Codificação C Embarcado;
5. Melhores Práticas de Codificação Java;
6. Melhores Práticas de Codificação C e C++;

<Para documentos internos:>

Título; Versão (se aplicável); Local de armazenamento da ferramenta de gerência de configuração CVS;

1. Plano do Projeto; Versão xx.xx; Local ...
2. Plano de Gerência de Configuração; Versão yy.yy; Local ...

*<Para documentos externos:>*

1. Título; Versão (se aplicável); Data; Instituição, divisão ou equipe responsável pelo documento; Link para o documento;

*<Para sites:>*

1. Título do Site; Endereço; (Data do último acesso);

…