

**DOCUMENTO DE ARQUITETURA DE SOFTWARE - DAS**

Solução: [nome da solução]

[ Digite o resumo do documento aqui. O resumo normalmente é uma breve apresentação do conteúdo do documento. ]

[Nota: O gabarito a seguir é fornecido para utilização elaboração do documento. 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.]

**CONTROLE DE REVISÃO**

|  |  |  |  |
| --- | --- | --- | --- |
| Versão | Data  | Autor | Descrição |
| *[0.1 Informe a versão do documento.* *Obs: A versão final para entrega será 1.0]* | *[dd/mm/aaaa]* | *[Informe o(s) autor(es) das alterações no documento]* | *[Descreva as alterações feitas no documento de forma resumida]* |
| *[0.2] Representando a revisão de número 2 do documento* | *[dd/mm/aaaa]* | *[Informe o(s) autor(es) das alterações no documento]* | *[Descreva as alterações feitas no documento de forma resumida]* |
| *[1.0] Representando a Primeira versão oficial do documento* | *[dd/mm/aaaa]* | *[Informe o(s) autor(es) das alterações no documento]* | *[Descreva as alterações feitas no documento de forma resumida]* |

**SUMÁRIO**

[1. INTRODUÇÃO 4](#_Toc25681013)

[2. REPRESENTAÇÃO ARQUITETURAL 7](#_Toc25681014)

[3. RESTRIÇÕES E METAS ARQUITETURAIS 8](#_Toc25681015)

[4. VISÃO DE CASOS DE USO 9](#_Toc25681016)

[5. VISÃO LÓGICA 11](#_Toc25681017)

[6. VISÃO GERAL 12](#_Toc25681018)

[7. VISÃO DE PROCESSOS 14](#_Toc25681019)

[8. VISÃO DE IMPLEMENTAÇÃO 15](#_Toc25681020)

[9. VISÃO DE IMPLANTAÇÃO 19](#_Toc25681021)

[10. VISÃO DE DADOS 21](#_Toc25681022)

[11. DIMENSIONAMENTO E PERFORMANCE 22](#_Toc25681023)

[12. QUALIDADE – iso 25000 23](#_Toc25681024)

[13. APROVAÇÕES 24](#_Toc25681025)

**DOCUMENTO DE ARQUITETURA DE SOFTWARE – DAS**

1. **INTRODUÇÃO**
	1. **OBJETIVO**

*[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.]*

Este documento provê uma visão de alto nível e explana a visão arquitetural do sistema {NOME-SISTEMA}.

Este documento define metas e objetivos arquiteturais, os cenários de casos de uso suportados pelo sistema, estilos arquiteturais e componentes utilizados. Este documento fornece um alinhamento para as decisões de arquitetura e projeto realizados desde a fase conceitual até a sua implantação.

A fim de descrever o software da forma mais apurada possível, a estrutura deste documento de arquitetura de *software* – DAS é baseada no modelo de visão de arquitetura 4+1 proposto inicialmente por Philippe Kruchten’s.



Figura 1-Modelo 4+1 da Engenharia de *Software*

* 1. **ESCOPO**

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

O escopo deste DAS é descrever a arquitetura do {NOME-SISTEMA}.

Este documento descreve os aspectos de projeto do {NOME-SISTEMA} que são considerados significativamente relevantes na visão de arquitetura, isto é, os elementos e comportamentos que são fundamentais para guiar e estabelecer diretrizes para a construção do sistema. Stakeholders que exigirem um entendimento técnico do {NOME-SISTEMA} são convidados para tomarem ciência deste documento e posteriormente, participar do processo de revisão dos artefatos complementares à este projeto como Documentos de Casos de Uso, Diagramas de Processos de Negócio e demais artefatos que contribuem para o enriquecimento da documentação do produto final a ser entregue. Por fim, o código-fonte poderá ser revisado após a entrega.

* 1. **DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES**

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

|  |  |
| --- | --- |
| Abreviação | Descrição |
| MVC | Model View Controller |
| ... | ... |

Tabela 1-Tabela de abreviações

|  |  |
| --- | --- |
| Termo | Definição |
| Model View Controller | Padrão de arquitetura de *software* (não confundir com um *design pattern*) que separa a representação da informação da interação do usuário com ele. O modelo (*model*) consiste nos dados da aplicação, regras de negócios, lógica e funções. Uma visão (*view*) pode ser qualquer saída de representação dos dados, como uma tabela ou um diagrama. É possível ter várias visões do mesmo dado, como um gráfico de barras para gerenciamento e uma visão tabular para contadores. O controlador (*controller*) faz a mediação da entrada, convertendo-a em comandos para o modelo ou visão. |
| Linux | Linux é um termo popularmente empregado para se referir a sistemas operativos (português europeu) ou sistemas operacionais (português brasileiro) que utilizam o núcleo Linux.... |
| ... | ... |

Tabela 2-Tabela de definições

* 1. **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. Obrigatoriamente devem ser referenciados os documentos do projeto tais como: TAP, Documento de Visão, Documentos de Requisitos, etc. Seguir padrão ABNT]

|  |
| --- |
| Fonte |
| [1] FOWLER, Martin. GUI Architectures. Disponível em: <https://martinfowler.com/eaaDev/uiArchs.html>. Acesso em 20 Jul. 2017. |
| [2] RED HAT. Linux Platforms. Disponível em: <https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux/first-time>. Acesso em 20 Ago. 2017 |
| ... |

Tabela 3-Tabela de referências

* 1. **VISÃO DO PROJETO**

[Esta subseção descreve de forma resumida uma breve descrição sobre o projeto/sistemas que o Documento de Arquitetura de Software está representando.]

* 1. **VISÃO GERAL**

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

Este documento consiste em {?} seções, descritas a seguir:

**Seção 1**: {descrever} *[Adicionar hyperlinks para cada seção no documento]*;

**Seção 2**: {descrever} *[Adicionar hyperlinks para cada seção no documento]*;

**Seção N:** {descrever} *[Adicionar hyperlinks para cada seção no documento]***;**

1. **REPRESENTAÇÃO ARQUITETURAL**

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

A modelagem, implementação e documentação de um sistema exige que a visão do sistema seja dividida em diferentes perspectivas. Por este motivo, a arquitetura do projeto {NOME-SISTEMA} será representada em uma abordagem de 6 visões. A seguir temos uma breve descrição de cada uma destas visões:

* **Visão de casos de uso**: Definir os principais direcionadores do sistema, que são compostos pelos requisitos funcionais do sistema;
* **Visão lógica**: Contém as definições do sistema em visão de diagramas de classes e objetos que descrevem os serviços que o sistema irá prover ao usuário final;
* **Visão de processos**: Detalha os processos que irão dar subsídio ao entendimento dos mecanismos do sistema, representados em forma de diagramas de comunicação, sequência e atividades.
* **Visão de implementação**: Define as especificações do sistema como interfaces e componentes utilizados, detalhando como estes componentes são dispostos em camadas e subsistemas;
* **Visão de implantação**: Descreve como os nós de hardware serão dispostos para trazer vida ao sistema, bem como configurações físicas necessárias para instalação, empacotamento e distribuição;
* **Visão de dados**: Descreve em uma visão de dados, a descrição das principais tabelas utilizadas e diagramas de entidade-relacionamento.
1. **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.]

1. **VISÃO DE CASOS DE USO**

**Audiência**: Analistas de Requisitos, *Stakeholders* e usuários finais.

**Área**: Esta seção lista os casos de uso ou cenários do modelo de casos de uso que representam alguma funcionalidade central significativa do sistema final ou se possuem ampla cobertura arquitetural e se estes experimentam muitos elementos arquiteturais ou se ilustram um ponto frágil especifico da arquitetura.

|  |  |  |
| --- | --- | --- |
| Image result for attachment flat icon | Diagrama de caso de uso relevante para a arquitetura do projeto. | **SUBSTITUIR** |

* + 1. **ATORES**

|  |  |
| --- | --- |
| Ator | Descrição (responsabilidade e atuação) |
| Agente Administrativo | Responsável por realizar, manter e solucionar aspectos administrativos do negócio, atuando como agente em casos de obstrução sistêmica (decisão manual, análise de dados, aprovação e/ou rejeição de requisições, etc). |
| Usuário Externo | Usuário final da aplicação que irá fazer adesão ao produto para conduzir seus interesses de financiamento. |
| ... | ... |

Tabela 4-Tabela de atores do sistema

* 1. **DESCRIÇÃO SIGNIFICATIVA DOS PRINCIPAIS CASOS DE USO OU HISTÓRIAS DE USUÁRIO**

|  |  |
| --- | --- |
| http://img.clubic.com/07423357-photo-tuleap.jpg | https://gestaoaplicacoes.mec.gov.br/projects/{meu-projeto} |

|  |  |
| --- | --- |
| Caso de Uso/História de Usuário | Descrição (O que se propõe a fazer) |
| Manter Dados Financeiros | Este Caso de uso será utilizado para registrar dados dos financiamentos das instituições que irão realizar a adesão ... |
| Integrar Dados RFB | Este Caso de uso será utilizado para apoiar o processo de validação de alteração de dados cadastrais do representante ... |
| ... | ... |

Tabela 5-Tabela com os principais casos de uso ou histórias de usuário do sistema

* + 1. **REALIZAÇÕES DE CASO DE USO**

[Esta seção descreve como a aplicação realmente trabalha através de uma seleção de realizações de casos de uso que explanam a utilização dos elementos de projeto que contribuem para a funcionalidades do sistema. Cada realização de caso de uso irá definir o projeto físico em termos colaboração entre classes e objetos que suportam o caso de uso. Portanto, cada realização de caso de uso é representada tipicamente por diagramas de sequência, evidenciando a colaboração entre a interação de objetos físicos.]

|  |  |  |
| --- | --- | --- |
| Image result for attachment flat icon | Diagrama de sequência de realização de casos de uso da estrutura relevante do projeto (quantos forem necessários para documentar). | **SUBSTITUIR** |

1. **VISÃO LÓGICA**

**Audiência**: Projetistas, Engenheiros e Desenvolvedores

**Área**: 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. Define a estrutura lógica da arquitetura e descreve as camadas que compõem a solução definindo as responsabilidades de cada uma.]

|  |  |  |
| --- | --- | --- |
| Image result for attachment flat icon | Diagrama de classe da visão lógica relevante para a arquitetura do projeto. | **SUBSTITUIR** |

|  |  |  |
| --- | --- | --- |
| Image result for attachment flat icon | Diagrama de componente da visão lógica de toda a arquitetura do projeto. | **SUBSTITUIR** |

1. **VISÃO GERAL**

Os principais componentes lógicos do projeto são:

|  |  |
| --- | --- |
| Elemento | Responsabilidade |
| Cliente web | * Prover interface para acesso ao usuário final;
* Realizar e centralizar a comunicação entre os demais serviços do sistema;
* Disponibilizar informações analíticas referente aos...
 |
| API Projeto | * Prover integração entre a camada de *front-end*;
* Acessar os dados do sistema;
* Disponibilizar serviços para acesso aos verbos REST que a aplicação de *front-end* irá consumir.
 |
| ... | ... |

Tabela 6-Visão geral dos componentes lógico do projeto

* 1. **ESTRUTURA DE CAMADAS**

[Para cada camada, descrever seu conceito e suas responsabilidades. Adicionar também uma lista dos subsistemas localizados na camada. Uma camada é um subsistema que agrega valor a subsistemas com menor nível de abstração. Subsistemas são coesos (Filosofia Unix).

Boas práticas:

* *Definir interfaces entre subsistemas para possibilitar o trabalho em paralelo;*
* *Observar o uso de padrões de projeto que melhorem ao invés de degradar a arquitetura;*
* *Responsabilidades similares devem ser agrupadas;*
* *Componentes com complexidade alta devem ser decompostos.]*

|  |  |  |
| --- | --- | --- |
| Image result for attachment flat icon | Diagrama de pacote de toda a arquitetura do projeto (quantos forem necessários para documentar). | **SUBSTITUIR** |

|  |  |  |
| --- | --- | --- |
| Camada/Subsistema | Descrição(caminho.do.pacote) | Responsabilidades |
| APRESENTAÇÃO | br.mec.gov.app.components | Responsável por gerenciar a construção de componentes de interface de usuário utilizando a biblioteca material-angular e todos os procedimentos de pré e pós renderização, como validação de campos, máscara e repasse de dados para a camada de serviços. |
| NEGÓCIO | ... | ... |
| SERVIÇO |  |  |
| AUTENTICAÇÃO |  |  |
| AUTORIZAÇÃO |  |  |
| PERSISTÊNCIA |  |  |
| INTEGRAÇÃO |  |  |
| EXCEÇÕES |  |  |
| AUDITORIA |  |  |
| ... |  |  |

Tabela 7-Definição das camadas e subsistemas de *Front-End*

|  |  |  |
| --- | --- | --- |
| Camada/Subsistema | Descrição (caminho.do.pacote) | Responsabilidades |
| NEGÓCIO | ... | ... |
| SERVIÇO |  |  |
| AUTENTICAÇÃO |  |  |
| AUTORIZAÇÃO |  |  |
| PERSISTÊNCIA |  |  |
| INTEGRAÇÃO |  |  |
| EXCEÇÕES |  |  |
| AUDITORIA |  |  |
| ... |  |  |

Tabela 8-Definição das camadas e subsistemas de *Back-End*

1. **VISÃO DE PROCESSOS**

**Audiência**: Analistas de Negócio, *Stakeholders*, Usuário Final e Integradores.

**Área**: 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 processo que se comunicam ou interagem. Descreva os modos principais de comunicação entre processos, como transmissão de mensagens e interrupções.

|  |  |  |
| --- | --- | --- |
| Image result for attachment flat icon | Diagrama de comunicação em uma visão de processos relevante para arquitetura do projeto. | **SUBSTITUIR** |

|  |  |  |
| --- | --- | --- |
| Image result for attachment flat icon | Diagrama de sequência em uma visão de processos relevante para arquitetura do projeto (necessário caso o diagrama de atividade não seja fornecido). | **SUBSTITUIR** |

|  |  |  |
| --- | --- | --- |
| Image result for attachment flat icon | Diagrama de atividades em uma visão de processos relevante para arquitetura do projeto (necessário caso o diagrama de sequência não seja fornecido). | **SUBSTITUIR** |

1. **VISÃO DE IMPLEMENTAÇÃO**

**Audiência**: Arquiteto de *software*, Desenvolvedor, Testador.

**Área**: 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.

* 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 o relacionamentos entre as camadas.]

* 1. **PADRÕES DE PROJETO (DESIGN PATTERNS) UTILIZADOS**

|  |  |
| --- | --- |
| https://www.riversideca.gov/utilities/images/eucalyp-coglight.png | Ver documento de diretrizes arquiteturais. |

* + 1. **PADRÕES DE PROJETO *FRONT-END***

[Descrever aqui os padrões de projeto utilizados na aplicação front-end e suas responsabilidades]

|  |  |
| --- | --- |
| Padrão de Projeto | Responsabilidades |
| SERVICE | ... |
| REPOSITORY | ... |
| OBSERVABLE | ... |

Tabela 9-Padrões de Projeto da aplicação *front-end*

* + 1. **PADRÕES DE PROJETO *BACK-END***

[Descrever aqui os padrões de projeto utilizados na aplicação *back-end* e suas responsabilidades]

|  |  |
| --- | --- |
| Padrão de Projeto | Responsabilidades |
| REST CONTROLLER | ... |
| FILTER | ... |
| MODEL | ... |
| REPOSITORY | ... |
| ... | ... |

Tabela 10-Padrões de Projeto da aplicação *back-end*

* 1. **COMPONENTES E FRAMEWORKS RELEVANTES PARA A ARQUITETURA DO PROJETO**

|  |  |
| --- | --- |
| Tecnologia/Framework/API | # |
| Angular | 4.3.3 |
| ChartJS | 2.6.0 |

Tabela 11-Tecnologias *Front-End*

|  |  |
| --- | --- |
| Tecnologia/Framework/API | # |
| Infinispan | 6.1.1 |
| Lucene/Solr | 3.6.2 |

Tabela 12-Tecnologias de API *Back-End*

* 1. **CONTROLE DE TRANSAÇÕES E CONCORRÊNCIA**

[Esta subseção descreve os mecanismos utilizados para o controle de transações da aplicação. O controle de transação ficará a cargo do servidor de aplicação ou da arquitetura do projeto? Existe a necessidade de realizar transações distribuídas? Qual padrão de projeto será utilizado? O projeto irá se basear em l*ock* otimista/pessimista? Descrever também a API que será utilizada ou conjunto destas.

Links de referência:

<https://docs.spring.io/spring/docs/current/spring-framework-reference/html/transaction.html>
<https://docs.jboss.org/hibernate/orm/3.5/reference/pt-BR/html/transactions.html>
<http://docs.oracle.com/javaee/7/api/javax/persistence/LockModeType.html>
<http://php.net/manual/en/pdo.transactions.php>
<https://laravel.com/docs/5.4/database>]

* 1. **ESTRATÉGIA DE REUSO**

[Esta subseção descreve os princípios utilizados para promover o reuso de forma à prevenir o forte acoplamento entre camadas]

* 1. **ESTRATÉGIA DE CACHE**

[Esta subseção descreve os princípios utilizados para soluções de cache e acesso à dados comumente utilizados]

* 1. **ESTRATÉGIA DE LOG**

[Esta subseção descreve os padrões utilizados para *logging* da aplicação e tratamento de exceções]

* 1. **SEGURANÇA E CONTROLE DE ACESSO**

[Esta subseção descreve os princípios básicos de segurança a serem utilizados, com foco no processo de autenticação e autorização de usuário. Quais serão os dados de entrada para verificação de acesso ao usuário. Descrever também procedimentos adotados em relação à auditoria e criptografia de dados. Descrever quais recursos serão sensíveis ou parcialmente protegidos como documentos e imagens e como a aplicação irá tratar.]

* 1. ***BUILD* E VERSIONAMENTO**

[Esta subseção descreve os padrões utilizados para as estratégias de *build* e versionamento da aplicação. Detalhar possíveis configurações adicionais e ajustes necessários para o mapeamento de versão da aplicação]

* 1. **GERENCIAMENTO DE DEPENDÊNCIAS**

[Esta subseção descreve as ferramentas utilizadas para gestão de dependências da aplicação, os profiles e configurações extras utilizadas]

* + 1. **GERENCIAMENTO DE DEPENDÊNCIAS FRONT-END**

[Descrever como as dependências da aplicação de *front-end* são mantidas e principais arquivos e diretórios que mantém a estrutura de dependências. Descrever restrições de atualização de bibliotecas e componentes e suas compatibilidades. Descrever os procedimentos para configuração/instalação das dependências caso seja necessário ajuste manual por parte do desenvolvedor]

* + 1. **GERENCIAMENTO DE DEPENDÊNCIAS *BACK-END***

[Descrever como as dependências da aplicação de *back-end* são mantidas e principais arquivos e diretórios que mantém a estrutura de dependências. Descrever restrições de atualização de bibliotecas e componentes e suas compatibilidades. Descrever os procedimentos para configuração/instalação das dependências caso seja necessário ajuste manual por parte do desenvolvedor]

* 1. **PADRÃO DE NOMENCLATURA/CODIFICAÇÃO**

[Esta subseção descreve os padrões de nomenclaturas de classes, métodos e variáveis e possíveis esquemas de configuração utilizado pelo projeto. Para cada situação, descrever um exemplo real utilizado e um exemplo não recomendado.]

* 1. **CONFIGURAÇÃO ADICIONAL DE AMBIENTE DE DESENVOLVIMENTO (OPCIONAL)**

[Esta subseção descreve configurações manuais realizadas para viabilizar o projeto em ambiente local de desenvolvimento. Descrever aqui também os requisitos necessários como softwares utilizados ou arquivos de configuração que deverão ser modificados.]

* 1. **INTERNACIONALIZAÇÃO - i18n (OPCIONAL)**

[Esta subseção descreve os mecanismos utilizados para adoção de internacionalização e gestão de arquivos de mensagens. Informar se o sistema deverá tratar diferentes idiomas.]

* 1. **ESTRATÉGIAS DE TESTE (OPCIONAL)**

[Esta seção descreve as estratégias de teste do sistema, quais ferramentas ou mecanismos serão utilizados, configurações ou detalhes [Esta seção descreve as estratégias de teste do sistema, quais ferramentas ou mecanismos serão utilizados, configurações ou detalhes para viabilizar a cobertura e cenário para notificação em caso de erros críticos ou eventos inesperados. Descrever também quais artefatos serão produzidos e entregues apontando a URL do repositório]

* + 1. **TESTE DE UNIDADE**

[Esta seção descreve a estratégia de teste de unidade da aplicação]

* + 1. **TESTE DE INTEGRAÇÃO (*END-TO-END*)**

[Esta seção descreve a estratégia de teste de integração]

* + 1. **TESTE DE *DEPLOYMENT***

[Esta seção descreve a estratégia de teste de *deployment* da aplicação e possível roteiro de *deployment* a ser utilizado. Descrever o nome do *job jenkins* utilizado, a ordem de procedimentos necessários para realizar o *deployment* de forma correta da aplicação e seus componentes auxiliares como execução de *scripts*, habilitação de portas de *firewall*, configuração prévia de serviços, etc. Detalhar apenas itens relevantes.]

1. **VISÃO DE IMPLANTAÇÃO**

**Audiência**: Gerentes de implantação, Analistas de Suporte, Analistas de Segurança.

**Área**: 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 (barramentos, LAN, ponto a ponto, etc.). Inclui também um mapeamento dos processo da Visualização do Processo sobre os nós físicos.

|  |  |  |
| --- | --- | --- |
| Image result for attachment flat icon | Diagrama de distribuição (utilizar de preferência o modelo de estrutura físico ao invés do modelo de estrutura lógica). | **SUBSTITUIR** |

* 1. **CONFIGURAÇÃO DO DOMÍNIO**

[Esta seção descreve o nome do domínio da aplicação e de demais módulos necessários. Caso a aplicação possua mais de uma API de *backend*, favor detalhar indicando a sugestão de nomenclatura conforme alinhamento prévio com a equipe de gestão de configuração do Ministério da Educação]

|  |  |
| --- | --- |
| DNS | CONTEXTO DA APLICAÇÃO : PORTA |
| dsv-projeto.mec.gov.br | {app-server}/projeto-web:4200 |
| dsv-projeto-api.mec.gov.br | {app-server}/projeto-api:8080 |

Tabela 13-Informar todos os domínios previstos em todos os ambientes necessários

* 1. **CONFIGURAÇÃO DO SERVIDOR DE APLICAÇÃO**

[Esta seção descreve as configurações necessárias no que tange à configuração do servidor de aplicação, parâmetros de JVM, cache e demais configurações físicas relacionadas ao projeto. Detalhar ao máximo possível todas as necessidades para que o projeto seja disponibilizado corretamente.]

|  |  |
| --- | --- |
| https://www.riversideca.gov/utilities/images/eucalyp-coglight.png | Ver documento de apoio:**02004**-MOVING PARA AMBIENTE ZEND SERVER – JENKINS (PHP) |

|  |  |  |
| --- | --- | --- |
| Agrupador | Sistema | Ambiente |
| ... | ... | .... |

Tabela 14-Estrutura exigida pela equipe CGI para configuração dos repositórios de configuração (CECAP)

* 1. **CONFIGURAÇÃO DE CONEXÃO COM BANCO DE DADOS**

[Esta seção descreve as configurações necessárias no que tange à configuração do servidor de aplicação referente às fontes de dados e demais configurações necessárias relacionadas ao projeto. Detalhar ao máximo possível todas as necessidades para que o projeto seja disponibilizado corretamente. Especificar aqui os itens relevantes do app-config-file.php (Ex: *Doctrine*) para projetos PHP ou persistence.xml (Ex: *Hibernate*) ou correspondente para projetos JAVA. Informar para todos os ambientes.]

* 1. **CONFIGURAÇÃO DE INTEGRAÇÃO COM SERVIÇOS EXTERNOS E/OU OBJETOS REMOTOS**

[Esta seção descreve as configurações necessárias no que tange à configuração de serviços externos, objetos remotos, liberação de portas e demais configurações necessárias relacionadas ao projeto. Detalhar ao máximo possível todas as necessidades para que o projeto seja disponibilizado corretamente.]

* 1. **ARTEFATOS DE IMPLANTAÇÃO**
		1. ARTEFATO (S) DE FRONT-END

[Descrever em detalhes o pacote a ser disponibilizado como: conteúdo previsto, tamanho aproximado em MB e se irá utilizar bibliotecas ou módulos externos.]

|  |  |  |
| --- | --- | --- |
| Nome do artefato + Extensão | Tamanho Mínimo Aproximado (MB) | Observações (Nome Datasource, Conteúdo Previsto, Demais configurações) |
| meu-projeto.zip | 7 | Um arquivo index.html, arquivos *javascript*, *cascading style sheets* e imagens. |
| … | … | … |

Tabela 15-Artefatos de front-end do sistema

* + 1. **ARTEFATO(S) DE BACK-END (API’S)**

[Descrever em detalhes o pacote a ser disponibilizado como: conteúdo previsto, tamanho aproximado em MB e se irá utilizar bibliotecas ou módulos externos.]

|  |  |  |
| --- | --- | --- |
| Nome do artefato + Extensão | Tamanho Mínimo Aproximado (MB) | Observações (Nome Datasource, Conteúdo Previsto, Demais configurações) |
| meu-projeto.ear | 23 | *Datasource*: meu-projeto-dsConteúdo: arquivos binários em *bytecode*, *properties* e xml. |
| … | … | … |

Tabela 16-Artefatos de *back-end* do sistema

1. **VISÃO DE DADOS**

**Audiência**: Projetistas, Especialistas de Dados, Administradores de Dados.

**Área**: Uma descrição da perspectiva de armazenamento de dados persistentes do sistema. Esta seção será opcional se os dados persistentes forem poucos ou in

existentes ou se os dados forem poucos ou inexistentes ou se a conversão entre o Modelo de Design e o Modelo de Dados for trivial.

|  |  |  |
| --- | --- | --- |
| http://www.powerdesigner.biz/images/logo-powerdesigner.png | Diagrama de entidade e relacionamento das tabelas relevantes para o projeto. | **SUBSTITUIR** |

1. **DIMENSIONAMENTO E PERFORMANCE**

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

* 1. **VOLUMETRIA**

[Uma descrição das principais características de volumetria desejadas.

Exemplos:

* Previsão de 50000 usuários ativos nos períodos de pico compreendendo os intervalos de 12:00 – 14:00 e das 18:00 as 22:00;
* Acesso diário de 1000 usuários (25%);
* Previsão de alocação de espaço em disco de até 1GB por usuário ativo;
* Os serviços de integração deverão retornar as consultas realizadas em até 10 segundos;
* Estimativa de membros ativos no primeiro mês: 20000;
* Tráfego de envio de arquivos de documentos (.pdf e .docx) de até 1GB/dia;]
	1. **PERFORMANCE**

[Uma descrição das principais características de performance desejadas.

Exemplos:

* Pesquisas deverão retornar em 90% dos casos em até 2 segundos;
* Transações financeiras deverão ser concluídas em até 10 segundos;
* O usuário não deverá aguardar mais que 15 segundos para a geração de relatórios analíticos;
* Processos em lote deverão possuir máxima prioridade;
* O sistema deverá ser projetado para suportar alta disponibilidade em períodos de inscrição e adesão.
* O sistema deverá suportar ao menos 50 mil usuários simultâneos.]
1. **QUALIDADE – ISO 25000**

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

|  |  |
| --- | --- |
| https://www.riversideca.gov/utilities/images/eucalyp-coglight.png | Ver documento de diretrizes arquiteturais(APÊNDICE A – TABELA DE CLASSIFICAÇÃO DO SISTEMA). |

* 1. **MATRIZ DE REQUISITOS NÃO FUNCIONAIS DE QUALIDADE**

|  |  |
| --- | --- |
| VISIBILIDADE | Externo |
| INTERAÇÃO | Humana |
| VOLUME | Grande |
| COMPLEXIDADE | Complexo |
| CRITICIDADE | Crítico |
| ID DO REQUISITO DE QUALIDADE (RNF-QA) | **SEÇÕES ABORDADAS PELO DAS** |
| 2.2.2.3 | 2,3 |
| 2.2.2.4 | 3,4.2,5 |
| 2.2.2.5 | 5 |
| … | ... |
|  |  |

Tabela 17-Matriz de requisitos não funcionais de qualidade

1. **APROVAÇÕES**

|  |
| --- |
| APROVAÇÃO |
| Gerente de Projetos Contratante | **Núcleo de Arquitetura** |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_<Nome> | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_<Nome> |

|  |
| --- |
| CIÊNCIA |
| Gerente de Projetos Contratada |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_<Nome> |

|  |
| --- |
| CIÊNCIA |
| Arquiteto Líder Contratada |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_<Nome> |

|  |
| --- |
| CIÊNCIA |
| Arquiteto Responsável Contratada |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_<Nome> |

##

 **ORIENTAÇÕES**

Os textos de orientação na cor ◼, contidos entre colchetes, deverão ser retirados na entrega da versão final

Ao preencher as informações de um item, justificar caso não seja explorado no documento

*[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.]*

99.9 ITEM

Informações em destaque na cor ◼, deverão ser susbstituídas pelas informações do projeto a ser documentado

Informações descritas na cor ◼, não deverão ser substituídas. Este conteúdo faz parte do documento final

Este documento provê uma visão de alto nível e explana a visão arquitetural do sistema {NOME-SISTEMA}.

Este documento define metas e objetivos arquiteturais, os cenários de casos de uso suportados pelo sistema.

Substituir por completo os itens sinalizados com a palavra SUBSTITUIR (não colocar o conteúdo dentro)

|  |  |  |
| --- | --- | --- |
| Image result for attachment flat icon | Diagrama de caso de uso relevante para a arquitetura do projeto. | **SUBSTITUIR** |

Este item não deverá ser removido do documento final

|  |  |
| --- | --- |
| http://img.clubic.com/07423357-photo-tuleap.jpg | https://gestaoaplicacoes.mec.gov.br/projects/{meu-projeto} |

Itens de ajuda deverão ser removidos do documento final

|  |  |
| --- | --- |
| https://www.riversideca.gov/utilities/images/eucalyp-coglight.png | Ver documento de diretrizes arquiteturais. |