



**PODER JUDICIÁRIO  
TRIBUNAL DE JUSTIÇA DO MARANHÃO  
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO  
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

**Sumário**

1. Objetivo .....	2
2. Definições .....	2
2.1. Cliente.....	2
2.2. Arquiteto de Software .....	2
2.3. Analista de Requisitos .....	2
2.4. Gerente de Projetos .....	2
3. Processo: Cumprimento de Requisição.....	3
3.1. Papéis e Responsabilidades.....	3
3.2. Fluxo do Processo.....	7
3.3. Entradas e Saídas do Processo .....	8
3.4. Descrição do Fluxo do Processo .....	8
4. Matriz RECI (RACI) .....	12
5. Controles do Processo.....	13
5.1. Indicadores do Processo.....	13
6. Histórico de Revisão e Periodicidade .....	14



**PODER JUDICIÁRIO  
TRIBUNAL DE JUSTIÇA DO MARANHÃO  
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO  
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

**1. Objetivo**

Estabelece um padrão de trabalho para o processo de Processo de Gerenciamento de Arquitetura de Software.

**2. Definições**

**2.1. Cliente**

Representa os interesses do negócio. Deve ter conhecimento suficiente para decidir sobre os aspectos negociais, apoiar o Gerente de Projeto e a equipe do projeto.

**2.2. Arquiteto de Software**

Responsável por estabelecer a estrutura geral de cada visão de arquitetura: a decomposição da visão, o agrupamento dos elementos e as interfaces entre esses principais agrupamentos. Portanto, comparado aos outros papéis, a visão do arquiteto de software é ampla, e não detalhada.

**2.3. Analista de Requisitos**

Responsável por identificar os problemas e necessidades do cliente referente ao sistema em desenvolvimento ou em manutenção, propondo uma solução sistemática e detalhamento de todos os requisitos do sistema (Funcionais, não funcionais e Restrições do projeto). Conduz o levantamento das informações necessárias para análise e validação do sistema junto ao cliente e a equipe técnica de desenvolvimento do projeto.

**2.4. Gerente de Projetos**

Profissional experiente na gestão de projetos que zela pela sua correta execução e ajuda a equipe na resolução de possíveis impedimentos, dando apoio para que as atividades sejam executadas dentro do escopo, prazo e custos planejados.

Conhecimentos e habilidades:

- Conhecimento do processo e das práticas de Gestão de Projetos
- Capacidade de disseminação de conhecimento
- Compreensão do processo de mudanças
- Trabalho em equipe



**PODER JUDICIÁRIO  
TRIBUNAL DE JUSTIÇA DO MARANHÃO  
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO  
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

**3. Processo: Cumprimento de Requisição**

**3.1. Papéis e Responsabilidades**

<b>PAPEL</b>	<b>RESPONSÁVEL</b>	<b>ATIVIDADES</b>
<b>Arquiteto de Software</b>		<ol style="list-style-type: none"><li>1. Elencar possíveis soluções técnicas para atender os requisitos iniciais do projeto, considerando:<ul style="list-style-type: none"><li>- O ambiente tecnológico;</li><li>- Proposta arquitetural (se houver);</li><li>- Arquiteturas e soluções publicadas na base de conhecimento;</li><li>- Padrões de projeto;</li><li>- Recursos de mercado;</li><li>- Soluções de referência publicados na comunidade técnica.</li></ul></li><li>Se já existir uma arquitetura de projetos, deve-se avaliar se a ela atende as necessidades técnicas dos requisitos apresentados e caso contrário levantar possíveis pontos de melhoria a serem levados ao cliente para aprovação</li><li>2. Realizar a escolha da arquitetura ideal, levando em consideração as linguagens (ex. PHP, Java, .Net), ambiente de execução (servidor de aplicação, de banco de dados) e integrações com demais sistemas</li><li>3. Realizar Provas de Conceito para testar a eficácia de uma nova solução técnica, arquitetura proposta ou fatores que apresentem incerteza na sua viabilidade com o objetivo de mitigar riscos arquiteturais.</li><li>4. Atualizar Arquitetura com base na análise dos requisitos elaborados para a iteração e avaliando o impacto na arquitetura</li><li>5. Detalhar Arquitetura: descrever, especificar, diagramar e exemplificar como será o funcionamento do sistema. Utilizar, se necessário:<ul style="list-style-type: none"><li>- UML: diagramas de classe, de sequência, de implantação e padrões de projeto</li></ul></li></ol>



**PODER JUDICIÁRIO**  
**TRIBUNAL DE JUSTIÇA DO MARANHÃO**  
**DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO**  
**PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

PAPEL	RESPONSÁVEL	ATIVIDADES
		<p>- Mapa de Integração: que identifica e detalha as interfaces com os demais sistemas trazendo informações como, protocolo de comunicação e parâmetros de entrada e saída"</p> <p>6. Elaborar os diagramas abaixo, conforme necessidade:</p> <ul style="list-style-type: none"><li>- Diagrama de Classe</li><li>- Diagrama de Pacote;</li><li>- Diagrama de Colaboração;</li><li>- Diagrama de Implantação.</li><li>- Diagrama de Sequência (*)</li><li>- Diagrama de Componentes (*)</li></ul> <p>(*) Elaborar se:</p> <ul style="list-style-type: none"><li>- Casos de Uso significativo para o negócio, ou seja, representa ponto crítico para o sistema, pois o mesmo possibilita a interação de outros casos de uso;</li><li>- Casos de uso possui uma grande quantidade de consultas, integrações, processamento, validações e/ou cálculos</li></ul> <p>7. Planejar a integração do produto, definindo uma estratégia e identificando o ambiente e os procedimentos necessários para a integração</p> <p>8. Alocar servidores e banco de dados, estabelecer conectividades, atribuir licenças e permissões à equipe</p> <p>9. Criar o(s) ambiente(s) de desenvolvimento integrado</p> <p>10. Construir a infraestrutura arquitetural com todos os elementos definidos no documento de arquitetura e guias, tais como os frameworks, padrões de projeto, componentes e configurações de servidores e banco de dados</p> <p>11. Atualizar as estratégias de integração contidas no documento de arquitetura e/ou documento de especificação de integração de produto observando os seguintes itens:</p> <ul style="list-style-type: none"><li>- O sequenciamento de integração está consolidado;</li><li>- Os padrões para o desenvolvimento foram observados;</li><li>- A definição do ambiente de integração é viável;</li><li>- A estratégia e procedimentos de integração estão claros</li></ul>



**PODER JUDICIÁRIO**  
**TRIBUNAL DE JUSTIÇA DO MARANHÃO**  
**DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO**  
**PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

<b>PAPEL</b>	<b>RESPONSÁVEL</b>	<b>ATIVIDADES</b>
		<ol style="list-style-type: none"><li>12. Criar/atualizar documento Notas de Release com as informações e restrições da versão.</li><li>13. Realizar a integração conforme procedimentos e critérios estabelecidos anteriormente a fim de obter um produto executável a partir do código-fonte</li><li>14. Realizar a liberação da versão integrada</li></ol>
<b>Ferramenta de Integração</b>		<ol style="list-style-type: none"><li>1. O serviço analisa a conformidade das regras cadastradas com o código-fonte e armazena o resultado.</li><li>2. A ferramenta, após a análise estática do código, emite um feedback com o status da análise. Caso não haja erro será feita a verificação da integração. Caso haja a atividade é encerrada até que seja re-estartada.</li><li>3. Cada integração é verificada por um build automatizado (incluindo testes que são fundamentais) para detectar erros e devolver um feedback o mais breve possível. Essa abordagem reduz significativamente problemas com integração e permite que a equipe desenvolva software mais coeso de forma mais rápida.  No build de integração pode haver várias tarefas como por exemplo: Compilação, Testes Unitários, Cópia de bibliotecas externas, Criação de Banco de Dados, Empacotamento, etc. Não existe um limite estabelecido para o número de tarefas realizadas no build.  As tarefas configuradas são:<ul style="list-style-type: none"><li>• Fazer o checkout</li><li>• Resolver dependências</li><li>• Compilar</li><li>• Testar - Testes unitários automatizados. Teste de código (Testa comportamento/estado Ex. Chama o método com alguns parâmetros e verifica se o retorno é o esperado)</li></ul></li><li>4. Se a verificação não falhou:<ul style="list-style-type: none"><li>• Empacotar (gerar um EAR - binário implantável com a versão)</li><li>• Armazenar (no repositório de artefatos)</li></ul></li><li>5. Se a verificação falhou a ferramenta emite notificação (e-mail) automaticamente para a</li></ol>



**PODER JUDICIÁRIO  
TRIBUNAL DE JUSTIÇA DO MARANHÃO  
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO  
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

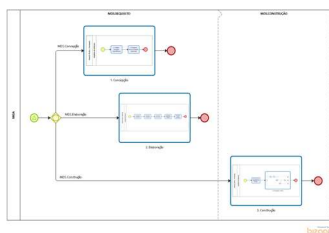
<b>PAPEL</b>	<b>RESPONSÁVEL</b>	<b>ATIVIDADES</b>
		equipe a respeito do problema encontrado na "quebra" do código.



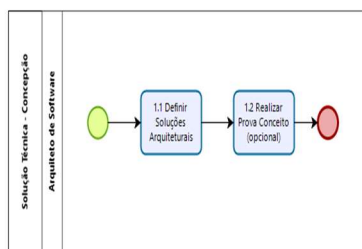
**PODER JUDICIÁRIO**  
**TRIBUNAL DE JUSTIÇA DO MARANHÃO**  
**DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO**  
**PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

**3.2. Fluxo do Processo**

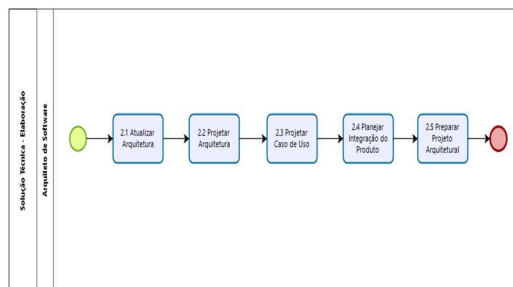


Concepção



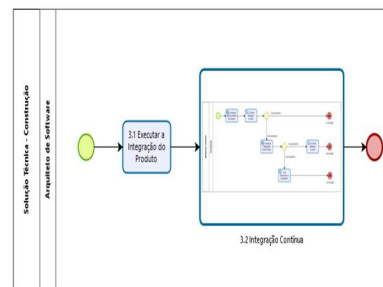
Powered by  
**bizagi**  
Modeler

Elaboração



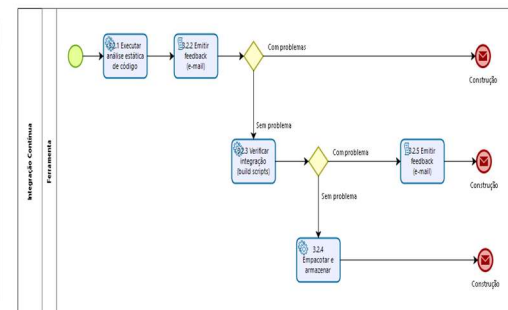
Powered by  
**bizagi**  
Modeler

Construção



Powered by  
**bizagi**  
Modeler

Integração Contínua



Powered by  
**bizagi**  
Modeler



**PODER JUDICIÁRIO  
TRIBUNAL DE JUSTIÇA DO MARANHÃO  
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO  
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

**3.3. Entradas e Saídas do Processo**

<b>ENTRADAS</b>	<b>SAÍDAS</b>
<ul style="list-style-type: none"><li>- Template Documento de Arquitetura</li><li>- Template Especificação de Integração de Produto</li><li>- Template Mapa de Integração</li><li>- Template Plano de Implantação</li><li>- Template Soluções Candidatas</li></ul>	<ul style="list-style-type: none"><li>- Ambiente(s) criado(s)</li><li>- Build</li><li>- Código-Fonte da POC</li><li>- Diagramas</li><li>- Documento de Arquitetura</li><li>- Especificação de Integração de Produto</li><li>- Infraestrutura criada</li><li>- Mapa de Integração</li><li>- Nota de Release</li><li>- Notificação (e-mail)</li><li>- Pacote de Sistema</li><li>- Plano de Implantação</li><li>- Soluções Candidatas</li></ul>

**3.4. Descrição do Fluxo do Processo**

<b>#IT</b>	<b>Atividades</b>	<b>Responsável</b>	<b>Procedimentos</b>
	1.1 DEFINIR Soluções Arquiteturais		<ul style="list-style-type: none"><li>- Elencar possíveis soluções técnicas para atender os requisitos iniciais do projeto, considerando:<ul style="list-style-type: none"><li>- O ambiente tecnológico;</li><li>- Proposta arquitetural (se houver);</li><li>- Arquiteturas e soluções publicadas na base de conhecimento;</li><li>- Padrões de projeto;</li></ul></li></ul>





**PODER JUDICIÁRIO**  
**TRIBUNAL DE JUSTIÇA DO MARANHÃO**  
**DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO**  
**PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

#IT	Atividades	Responsável	Procedimentos
			<ul style="list-style-type: none"><li>- Recursos de mercado;</li><li>- Soluções de referência publicados na comunidade técnica.</li></ul> <p>Se já existir uma arquitetura de projetos, deve-se avaliar se a ela atende as necessidades técnicas dos requisitos apresentados e caso contrário levantar possíveis pontos de melhoria a serem levados ao cliente para aprovação</p> <ul style="list-style-type: none"><li>– Realizar a escolha da arquitetura ideal, levando em consideração as linguagens (ex. PHP, Java, .Net), ambiente de execução (servidor de aplicação, de banco de dados) e integrações com demais sistemas</li></ul>
	1.2 REALIZAR Prova Conceito (opcional)		<ul style="list-style-type: none"><li>– Realizar Provas de Conceito para testar a eficácia de uma nova solução técnica, arquitetura proposta ou fatores que apresentem incerteza na sua viabilidade com o objetivo de mitigar riscos arquiteturais.</li></ul>
	2.1 ATUALIZAR Arquitetura		<ul style="list-style-type: none"><li>– Atualizar Arquitetura com base na análise dos requisitos elaborados para a iteração e avaliando o impacto na arquitetura</li></ul>
	2.2 PROJETAR Arquitetura		<ul style="list-style-type: none"><li>– Detalhar Arquitetura: descrever, especificar, diagramar e exemplificar como será o funcionamento do sistema.</li></ul> <p>Utilizar, se necessário:</p> <ul style="list-style-type: none"><li>- UML: diagramas de classe, de sequência, de implantação e padrões de projeto</li><li>- Mapa de Integração: que identifica e detalha as interfaces com os demais sistemas trazendo informações como, protocolo de comunicação e parâmetros de entrada e saída</li></ul>
	2.3 PROJETAR Caso de Uso		<ul style="list-style-type: none"><li>– Elaborar os diagramas abaixo, conforme necessidade:<ul style="list-style-type: none"><li>- Diagrama de Classe</li><li>- Diagrama de Pacote;</li><li>- Diagrama de Colaboração;</li><li>- Diagrama de Implantação.</li><li>- Diagrama de Sequência (*)</li></ul></li></ul>



**PODER JUDICIÁRIO**  
**TRIBUNAL DE JUSTIÇA DO MARANHÃO**  
**DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO**  
**PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

#IT	Atividades	Responsável	Procedimentos
			<ul style="list-style-type: none"><li>- Diagrama de Componentes (*)</li><li>(*) Elaborar se:<ul style="list-style-type: none"><li>- Casos de Uso significativo para o negócio, ou seja, representa ponto crítico para o sistema, pois ele possibilita a interação de outros casos de uso;</li><li>- Casos de uso possui uma grande quantidade de consultas, integrações, processamento, validações e/ou cálculos"</li></ul></li></ul>
	2.4 PLANEJAR Integração do Produto		<ul style="list-style-type: none"><li>- Planejar a integração do produto, definindo uma estratégia e identificando o ambiente e os procedimentos necessários para a integração</li></ul>
	2.5 PREPARAR Projeto Arquitetural		<ul style="list-style-type: none"><li>- Alocar servidores e banco de dados, estabelecer conectividades, atribuir licenças e permissões à equipe</li><li>- Criar o(s) ambiente(s) de desenvolvimento integrado</li><li>- Construir a infraestrutura arquitetural com todos os elementos definidos no documento de arquitetura e guias, tais como os frameworks, padrões de projeto, componentes e configurações de servidores e banco de dados</li></ul>
	3.1 EXECUTAR a Integração do Produto		<ul style="list-style-type: none"><li>- Atualizar as estratégias de integração contidas no documento de arquitetura e/ou documento de especificação de integração de produto observando os seguintes itens:<ul style="list-style-type: none"><li>- O sequenciamento de integração está consolidado;</li><li>- Os padrões para o desenvolvimento foram observados;</li><li>- A definição do ambiente de integração é viável;</li><li>- A estratégia e procedimentos de integração estão claros"</li></ul></li><li>- Criar/atualizar documento Notas de Release com as informações e restrições da versão.</li><li>- Realizar a integração conforme procedimentos e critérios estabelecidos anteriormente a fim de obter um produto executável a partir do código-fonte</li><li>- Realizar a liberação da versão integrada</li></ul>
	3.2 INTEGRAÇÃO Contínua		



**PODER JUDICIÁRIO**  
**TRIBUNAL DE JUSTIÇA DO MARANHÃO**  
**DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO**  
**PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

#IT	Atividades	Responsável	Procedimentos
	3.2.1 EXECUTAR análise estática de código		– O serviço analisa a conformidade das regras cadastradas com o código-fonte e armazena o resultado.
	3.2.2 EMITIR feedback		– A ferramenta, após a análise estática do código, emite um feedback com o status da análise. Caso não haja erro será feita a verificação da integração. Caso haja a atividade é encerrada até que seja re-estartada.
	3.2.3 VERIFICAR integração (build scripts)		– Cada integração é verificada por um build automatizado (incluindo testes que são fundamentais) para detectar erros e devolver um feedback o mais breve possível. Essa abordagem reduz significativamente problemas com integração e permite que a equipe desenvolva software mais coeso de forma mais rápida. No build de integração pode haver várias tarefas como por exemplo: Compilação, Testes Unitários, Cópia de bibliotecas externas, Criação de Banco de Dados, Empacotamento, etc. Não existe um limite estabelecido para o número de tarefas realizadas no build.  As tarefas configuradas são: <ul style="list-style-type: none"><li>• Fazer o checkout</li><li>• Resolver dependências</li><li>• Compilar</li><li>• Testar - Testes unitários automatizados. Teste de código (Testa comportamento/estado Ex. Chama o método com alguns parâmetros e verifica se o retorno é o esperado)</li></ul>
	3.2.4 EMPACOTAR e armazenar		– Se a verificação não falhou: <ul style="list-style-type: none"><li>• Empacotar (gerar um EAR - binário implantável com a versão)</li><li>• Armazenar (no repositório de artefatos)</li></ul>
	3.2.5 EMITIR feedback		– Se a verificação falhou a ferramenta emite notificação (e-mail) automaticamente para a equipe a respeito do problema encontrado na "quebra" do código



**PODER JUDICIÁRIO**  
**TRIBUNAL DE JUSTIÇA DO MARANHÃO**  
**DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO**  
**PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

**4. Matriz RECI (RACI)**

<b>Atividades</b>	<b>Arquiteto de Software</b>	<b>Gerente de Projetos</b>	<b>Equipe Técnica</b>	<b>Cliente</b>	<b>Ferramenta de Integração</b>	<b>Analista de Requisitos</b>	<b>-</b>
1.1 DEFINIR Soluções Arquiteturais	E	R/C	C	I	-	C	-
1.2 REALIZAR Prova Conceito (opcional)	E	R/C	C	I	-	C	-
2.1 ATUALIZAR Arquitetura	E	R/C	C	I	-	C	-
2.2 PROJETAR Arquitetura	E	R/C	C	I	-	C	-
2.3 PROJETAR Caso de Uso	E	R/C	C	I	-	C	-
2.4 PLANEJAR Integração do Produto	E	R/C	C	I	-	C	-
2.5 PREPARAR Projeto Arquitetural	E	R/C	C	I	-	C	-
3.1 EXECUTAR a Integração do Produto	E	R/C	C	I	-	C	-
3.2 INTEGRAÇÃO Contínua	-	R	-	-	-	-	-
3.2.1 EXECUTAR análise estática de código	E/I	R	C	-	E	-	-
3.2.2 EMITIR feedback	E/I	R	C	-	E	-	-
3.2.3 VERIFICAR integração (build scripts)	E/I	R	C	-	E	-	-
3.2.4 EMPACOTAR e armazenar	E/I	R	C	-	E	-	-
3.2.5 EMITIR feedback	E/I	R	C	-	E	-	-

**LEGENDA:** R – Responsável (*Accountable*) | E – Executor (*Responsible*) | C – Consultado (*Consulted*) | I – Informado (*Informed*)



**PODER JUDICIÁRIO  
TRIBUNAL DE JUSTIÇA DO MARANHÃO  
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO  
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

**5. Controles do Processo**

**5.1. Indicadores do Processo**

<b>Descrição</b>	<b>Definir com o cliente</b>
<b>Periodicidade</b>	
<b>Meta</b>	
<b>Cálculo</b>	



**PODER JUDICIÁRIO  
TRIBUNAL DE JUSTIÇA DO MARANHÃO  
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO  
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

**6. Histórico de Revisão e Periodicidade**

<b>Número</b>	<b>Data</b>	<b>Versão</b>	<b>Responsável</b>	<b>Descrição</b>
001	07/05/2020	1.0	CTIS/SONDA: Erick Saraiva	Elaboração do documento.