**PS-MCTIC**

**Guia Operacional – Regras e Padrões de Modelagem de Dados**

**Versão 1.2**

|  |
| --- |
| **Histórico de Revisões** |
| **Versão** | **Data** | **Descrição** | **Autor** |
| 1.0 | 15/01/2015 | Elaboração do Artefato | GNPTI |
| 1.1 | 28/01/2015 | Inclusão da referência do PS-MCTI | RSI Informática |
| 1.2 | 28/03/2017 | Atualização da sigla MCTI para MCTIC e descrição pela fusão do Ministério das Comunicações. | CTIS |

Sumário

[1. INTRODUÇÃO 4](#_Toc478460248)

[2. OBJETIVO 4](#_Toc478460249)

[3. REGRAS E PADRÕES 5](#_Toc478460250)

[3.1 Normalização 5](#_Toc478460251)

[3.2 Modelo de Dados 6](#_Toc478460252)

[3.3 Entidade 6](#_Toc478460253)

[3.4 Atributo 7](#_Toc478460254)

[3.5 Relacionamento 8](#_Toc478460255)

[3.6 Chave Primária 8](#_Toc478460256)

[3.7 Regras de Integridade 9](#_Toc478460257)

[3.8 Dicionário de Dados 10](#_Toc478460258)

[3.9 Redundância 12](#_Toc478460259)

[3.10 Padrões de Nomenclatura 12](#_Toc478460260)

[3.11 Exemplo de modelo de dados 15](#_Toc478460261)

[4. REFERÊNCIAS 15](#_Toc478460262)

Índice de Figuras

[Figura 1 – Modelo de Dados SAA 14](#_Toc409102399)

# INTRODUÇÃO

O objetivo deste documento é padronizar a modelagem de dados dos sistemas do MCTIC, para que o modelo de dados possa ser criado de forma clara e objetiva, representando o negócio e facilitando a manutenção e entendimento do sistema. Esse guia irá apoiar a atividade Projetar Banco de Dados do Processo de Software (PS-MCTIC).

# OBJETIVO

Modelar significa criar um modelo que explique as características de funcionamento e comportamento de um software, a partir do qual ele será criado, facilitando seus entendimentos e seu projeto. Processo de modelagem representa o negócio da organização, definindo os objetos de dados de interesse e suas interações.

A abordagem que se dispensa ao assunto, normalmente atende a três perspectivas: Modelagem Conceitual, Modelagem Lógica e Modelagem Física.

* **Modelagem Conceitual** é usada como representação de alto nível e considera exclusivamente o ponto de vista do usuário criador do dado;
* **Modelagem Lógica** agrega alguns detalhes de implementação, de acordo com a tecnologia de banco de dados adotada;
* **Modelagem Física** demonstra como os dados são fisicamente armazenados. Limitado pelo SGBD (Sistema Gerenciador de Banco de Dados).

Abaixo apresentamos os tipos de sistemas de se enquadram na abordagem da modelagem de dados:

* Modelagem de Dados para Sistemas OLTP
	+ A modelagem de dados para sistemas OLTP deve seguir a técnica de modelagem de dados relacional. Baseados em uma teoria formal, chamada modelo relacional de dados, onde os dados são percebidos pelo usuário por meio de linhas em tabelas (tuplas em relações), estrutura lógica. O modelo relacional é semântico, isto é, relativo ao significado. Assim, trata dados com significado, ou seja, informações propriamente ditas. Na modelagem de dados se deve ter a preocupação com o que efetivamente os dados representam no modelo em que serão utilizados.
* Modelagem de Dados Gerenciais (DATAMART/DATA WAREHOUSE)
	+ A modelagem de dados gerenciais deve seguir a técnica de modelagem de dados multidimencionais.

# REGRAS E PADRÕES

O modelo de dados representa, de forma gráfica e textual, os conceitos corporativos que fazem parte dos negócios do MCTIC. Participam desse escopo informações que precisam ser persistidas dentro das necessidades do negócio. Destaca-se que a dicionarização dos objetos de dados é parte do modelo de dados.

Cada conceito deve ser representado uma única vez e, desta forma, evitar redundância, gerar consistência e promover o reuso.

As técnicas de modelagem de dados são regras definidas para determinado tipo de tecnologia, como relacional, hierárquico, etc.

Os padrões de modelagem de dados são padrões estabelecidos pela organização, baseados na experiência corporativa e boas práticas do mercado. Por exemplo, padrões de nomenclatura.

## Normalização

Esta técnica possibilita uma estrutura estável e flexível, onde são evitadas redundância de dados, inconsistências e anomalias de atualização. A redundância de dados é um fato gerador de inconsistências. Já as anomalias de atualização criam dificuldades operacionais para manutenção do banco de dados.

O processo de normalização deu origem a três principais formas normais: primeira (1FN), segunda (2FN) e terceira (3FN).

O modelo de dados deve estar na terceira forma normal.

* **Primeira Forma Normal (1FN)**
	+ Uma entidade está na primeira forma normal se todos os seus atributos pertencem a domínios simples, ou seja, contem valores atômicos. Cada atributo não pode representar informações concatenadas. Cada informação deve ser representada uma única vez em cada ocorrência. Assim, o objetivo dessa etapa é eliminar dados estruturados e dados repetitivos, isto é, respectivamente, atributos compostos e atributos multivalorados.
* **Segunda Forma Normal (2FN)**
	+ Uma entidade está na 2FN se, e somente se, ela estiver em 1FN e todos os seus atributos não chave forem dependentes completa e exclusivamente da chave primária. Ou seja, não contenha atributos que dependente funcionalmente de subconjuntos da chave primária composta. Assim, o objetivo dessa etapa é eliminar as dependências parciais. Para tanto, devem ser geradas novas entidades contendo colunas que se encontram em dependência parcial com a chave primária.
* **Terceira Forma Normal (3FN)**

Uma entidade está na 3FN se, e somente se, estiver na 2FN e todos os seus atributos dependerem completa e exclusivamente da chave primária. Ou seja, não pode haver relação de identificação entre os atributos que não façam parte da chave primária (dependência transitiva). Assim, o objetivo é eliminar dependências transitivas.

## Modelo de Dados

Para a modelagem de dados relacional é adotada a notação conhecida popularmente como “pé de galinha”.

O modelo de dados deverá ser construído na Ferramenta Case adotada pelo MCTIC.

O modelo de dados deve atender as regras, as características da técnica de modelagem adotada e os padrões estabelecidos pelo MCTIC.

O processo adotado define a construção do modelo de dados lógico. Este modelo deve seguir os padrões de nomenclatura e estar normalizado até a terceira forma normal.

Cada modelo de dados lógico deve refletir apenas um banco de dados. O modelo de dados que sofrerá validação é o modelo de dados lógico, este modelo deverá refletir exatamente os objetos de dados do ambiente hierarquicamente superior ao de desenvolvimento e as alterações propostas.

A modelagem dos dados deve ser realizada em conjunto pelo analista de sistemas responsável, e validada por um administrador de dados. Cabe ao analista de sistemas realizar as criações/alterações gráficas e textuais no modelo de dados lógico.

Entidades de outros bancos de dados que tiverem relacionamento, apenas lógico, devem ser representadas na cor verde, apenas com os atributos que compõe a chave primária. Na definição deve constar o banco de dados de origem.

##  Entidade

Verificar se existe entidade corporativa que represente o conceito identificado, o que evita redundância e gera a consistência dos dados.

Caso for de conhecimento do modelador o potencial de reuso do conceito, consultar a equipe de administração de dados sobre a possibilidade de modelagem de estrutura corporativa.

Não podem existir duas entidades iguais no mesmo modelo de dados, ou seja, que representem o mesmo conceito e características, mesmo que possuam nomes diferentes.

Cada entidade deve possuir um identificador (chave primária).

Cada entidade deve possuir uma definição, indicando o que representa, o que faz e para que serve.

Entidades que representem situações, estados ou conceitos que mudem ao longo do tempo, devem ter entidade específica para a identificação da marca de tempo da situação. Ex: TBSituacaoObjeto. Os atributos seriam CDObjeto, DTSituacaoObjeto, TPSituacaoObjeto, onde CDObjeto e DTSituacaoObjeto formariam a chave primária.

Estruturas que permitem parametrização devem ser evitadas. Tais estruturas genéricas, em vez de representar as entidades e seus relacionamentos, implementam objetos que mascaram as entidades, atributos e seus conteúdos respectivamente, excluindo, desta forma, o aspecto semântico da modelagem de dados e não possibilitando a representação do negócio.

## Atributo

Cada atributo deve estar diretamente relacionado com o conceito da entidade em que participa.

Não deve ocorrer replicação de atributo, este somente pode ser representado na entidade de origem e sua informação ser obtida através da herança dos atributos identificadores dos relacionamentos.

Não devem ser representados atributos repetitivos, conforme orientado no item referente à normalização.

Cada atributo possuirá uma definição, a qual explicitará seu significado e indicará exemplos.

Atributos do tipo código que possuem origem em outra entidade devem ser representados por meio de relacionamento com a entidade de origem da informação.

Atributos com domínio estável e restrito (até cinco valores) devem ter definida check constraint para restringir o domínio.

## Relacionamento

Os relacionamentos devem ser definidos de acordo com o negócio, bem como as cardinalidades, opcionalidades e se são identificadores ou não.

O relacionamento deve ser identificado, se a entidade filha for fraca, isto é, entidade que depende da existência do conceito da entidade mãe para existir.

Cada entidade do modelo deve participar de pelo menos um relacionamento. Caso isso não ocorra é possível que a entidade isolada não faça parte do contexto modelado.

Não é possível excluir uma ocorrência de um registro de entidade mãe, sem que os registros vinculados das entidades filhas sejam também excluídos. As regras de integridade referencial devem ser declaradas para as chaves estrangeiras, isto é, “restrict’, “set null” e “cascade”. Onde cascade deve propagar a eliminação até as ocorrências das tabelas filhas; e restrict, onde a operação de delete é restrita. Sempre deve ser utilizada a regra “restrict”.

Não deverão ser adotados relacionamentos baseados em índices únicos, mas sim em chaves primárias.

Os relacionamentos do tipo muitos para muitos deverão ser substituídos por entidade associativa.

No caso de mais de um relacionamento entre duas entidades deverão ser alterados os nomes dos atributos herdados pela entidade filha, definindo o papel representado por cada relacionamento.

No caso de especialização, somente deverão ser representados os relacionamentos com cardinalidade 1:1.

Duas entidades que possuam restrição de uma ocorrência referente a determinado conceito para uma ocorrência de outro conceito serão representadas como entidades distintas, com identificação diferenciada e objetivando representar a relação do relacionamento de 1:N.

## Chave Primária

Toda entidade terá uma chave primária estabelecida. A chave primária estará diretamente relacionada com o conceito representado pela entidade.

Na escolha dos atributos que compõe a chave primária devem ser privilegiados aqueles relacionados ao negócio. Não devem ser adotados atributos com origem externa à organização na chave primária.

A escolha da chave primária não deve ser influênciada para garantir a eficiência de acesso.

O processo de escolha das chaves primárias das entidades de um banco de dados relacional constitui fator crítico que afeta a estabilidade do banco, pois os relacionamentos são implementados por meio da redundância das chaves nas entidades relacionadas (chaves estrangeiras). Portanto, qualquer alteração na chave primária pode repercutir em todos os relacionamentos nos quais a mesma esteja envolvida.

Preferencialmente, os atributos que compõe a chave primária deverão ser numéricos ou temporais. Atributos alfabéticos podem gerar dúvidas quanto à grafia.

Não deverão participar da chave primária atributos cujos valores sejam alterados com frequencia.

A chave artificial será adotada, como chave primária, quando a entidade representar a origem da informação na organização. A criação de atributos sequenciais desnecessários deve ser evitada, pois aumenta a complexidade do modelo de dados e elimina a rastreabilidade.

Chaves primárias de entidades associativas devem ser formadas pela concatenação das chaves primárias das entidades mães.

As entidades fracas deverão possuir chave primária concatenada, formada com os atributos que compõe a chave primária da entidade forte (mãe) acrescida de identificação própria.

## Regras de Integridade

O objetivo de se estabelecer regras de integridade é assegurar que as informações no banco de dados sejam precisas, isto é, preservá-lo contra atualizações inválidas.

É papel do analista de sistemas, modelador, definir estas regras e declará-las no modelo de dados quando isto for possível, ou definir rotinas que possam garanti-las.

* **Regra de Integridade Referencial**
	+ A integridade referencial é a responsável por manter em sincronismo as referências. É uma restrição a qual define que tudo o que é referenciado necessita existir na origem e o sistema gerenciador de banco de dados deve garantir que esta condição permaneça sempre verdadeira. A chave estrangeira ou foreing key (FK) é um conjunto de atributos de uma entidade cujos valores devem obrigatoriamente corresponder a valores de chave primária ou candidata de alguma entidade. Ademais, possibilita a capacidade de rastrear atributos, por meio da herança de chaves candidatas.
* **Regra de Integridade de Domínio**
	+ A integridade de domínio é a determinação, quando sabida, de características que limitem o conjunto de valores possíveis para um determinado atributo. A integridade de domínio verifica se os valores a serem inseridos atendem ao domínio previamente estabelecido, que é definido pela regra de negócio. Sempre, que possível, deverão ser definidas as restrições de domínio.
		- Tipo de Dados: é um conjunto definido de valores. Cada atributo deve ter um tipo de dados atribuído. Exemplo: int. Observar o tipo mais adequado para o uso e o significado pretendido. Ao ser herdado o atributo deve manter o mesmo tipo de dados utilizado na entidade de origem.
		- Tamanho: realizar estimativa apropriada que evite o super-dimensionamento ou sub-dimensionamento do tamanho do tipo de dados do atributo, visando a estabilidade.
			* Para atributos de natureza nome utilizar no mínimo 100 caracteres;
			* Para atributos de natureza descrição utilizar no mínimo 200 caracteres.
		- Lista de valores válidos: quando possível, a definição de um conjunto de valores válidos será estabelecida. Estes valores deverão ser estáveis e restritos, sendo que muitas vezes é possível determinar uma faixa de valores. É necessário estabelecer uma check constraint, o que obrigatoriamente deve ser definido nos atributos do tipo Situação ou Tipo. Recomenda-se a utilização de número, evitando erros de familiaridade. Se o domínio for muito volátil, criar uma tabela para codificar e descrever o domínio.
		- Valor nulo: o domínio de um atributo pode ou não incluir um valor diferente de todos os demais valores válidos do domínio, o valor nulo, isto é, ausência de valor. Determinado pela restrição de nulidade, que contempla as definições Null e Not Null.
		- No caso de atributos do tipo identificador ou código que não possuam domínio conhecido, deverá ser criada uma entidade de domínio.
		- Default: valor padrão que um atributo irá adotar, caso não seja declarado um valor para este.
		- Sequence/Identity: especifica um valor para o início de uma sequência numérica de forma automática e um valor de incremento. Utilizar em chaves artificiais.

## Dicionário de Dados

Por intermédio do dicionário de dados que se conhece o significado, características e uso que se deve dar a um determinado elemento. É de fundamental importância para a compreensão do acervo dos dados de uma organização. Um dos produtos deste repositório de metadados é a visão patrimonial dos dados de uma empresa, o que facilita a reutilização de elementos no ambiente.

As definições devem ser completas e inequívocas, onde devem ser observados aspectos com clareza, objetividade e normas da língua portuguesa. Evitando, assim, ambiguidade, sintetização e possibilidade de inferência. Cada um dos elementos identificados e representados deverá ser definido claramente para que, associando-se seu nome, sua representação e sua definição, sejamos capazes de ter o completo entendimento do conceito que estes procuram transmitir.

O que se deseja saber prioritariamente é o significado e não a finalidade. Assim não deve ser utilizado o conceito de local de armazenamento, evitando expressões como “Esta tabela contém os dados...” ou “Armazena...”.

Uma das dificuldades para o controle da redundância é justamente a identificação da ocorrência da mesma. Sendo assim, para que se detecte uma redundância é necessário reconhecer, com segurança, que um dado é exatamente igual ao outro. A verificação entre dados de um mesmo sistema (ou de sistemas diferentes, mas cujos dados envolvidos são conhecidos) se faz por meio da descrição e da lógica associada ao dado, ou seja, do tratamento ou utilidade prevista para ele. Desta maneira, é possível identificar dados iguais com denominações diferentes.

A análise das descrições contidas no dicionário de dados também permite verificar a inadequação do dado com relação ao seu uso. Casos de erros de lógica nos quais o uso previsto para o dado não foi atendido pela descrição do mesmo, bem como problemas de integridade de domínio remetem a consultas no dicionário para seu esclarecimento.

* Entidade
	+ Descrever o que representa a entidade, que tipo de conteúdo ela armazenará, condições de atualização, processos associados, população máxima e mínima esperada, o que engloba a categoria de elementos desta entidade e o que está excluída desta categoria.
* Atributo
	+ Descrever objetivamente o significado do atributo, situações de utilização obrigatória, regras de integridade associadas, domínio de valores com a descrição de cada um deles. Caracterize o atributo informando seu tamanho, tipo e nulidade (se assume ou não valores nulos e em que situações). Se existir algum atributo com domínio relacionado, indicar a dependência entre eles.
* Relacionamento
	+ Descrever a função que cada entidade desempenha no relacionamento.

## Redundância

A identificação de duplicidades pressupõe a identificação do atributo e uma forma de verificar a igualdade de significado entre eles.

A redundância deve ser combatida, e se não eliminada, cuidadosamente controlada. As razões para investir na eliminação das redundâncias são as seguintes:

* + Perda de espaço
	+ Possibilidade de falta de integridade entre ocorrências do atributo redundante.

A primeira razão, a menos grave, é o dilema espaço versus performance, o qual é rotineiro e deve ser considerado caso a caso. A segunda razão é crítica, pois pode originar resultados errados. Não pretendemos sugerir que toda redundância deva necessariamente ser eliminada. Algumas vezes existem sólidas razões técnicas ou de negócios para que se mantenham múltiplas cópias do mesmo dado. Entretanto, o sistema deve ter conhecimento dessas redundâncias e assumir a responsabilidade de propagar as atualizações.

## Padrões de Nomenclatura

* O nome deve ser descritivo, claro, objetivo e designar exatamente o que representa, bem como redigido na língua portuguesa, no masculino e singular.
* Sempre que possível, devem ser utilizados substantivos, que definam o objeto de forma unívoca dentro do ambiente corporativo.
* Utilizar apenas siglas ou abreviações de aceitação universal. EX: FGTS, CEP e CNPJ.
* Eliminar artigos, preposições ou separadores.
* Para nomes compostos, iniciar cada palavra que compõe o nome com letra maiúscula.
* Não utilizar acentos, algarismos e caracteres especiais.
* Não utilizar espaços em branco.
* Os termos devem ficar compostos em uma sequencia lógica para a compreensão.
* Limite de 40 caracteres
* Não devem ser utilizados termos específicos de determinada área, e sim utilizados conforme o conceito corporativo.
* Os nomes de entidades associativas devem definir o conceito da associação, o que gerar melhor compreensão.
* Os nomes das tabelas com característica exclusivamente física devem ser iniciados com prefixos específicos.
* Entidades fracas devem ter em seus nomes o conceito da entidade forte. Ex: ItemPedido.

**Regra geral:**

**{Prefixo} + {Nome do Objeto}**

onde:

* O prefixo deve ser escrito em letras maiúsculas.
* Os termos que compõe o nome devem ser iniciados em letras maiúsculas.

Ex: **TBPerfil**

CDPerfil

|  |  |  |
| --- | --- | --- |
| **Tipo de Objeto**  | **Nomenclatura recomendada**  | **Exemplo**  |
| TABELA  | TB + Nome da Tabela  | TBPessoa TBOrgaoConveniado  |
| PRIMARY KEY  | PK+ Nome da Tabela (sem prefixo TB)  | PKPessoa PKPerfil  |
| FOREIGN KEY  | FK + Nome da Tabela Origem + Nome da Tabela (FK) (sem prefixos TB)  | FKUFOrgao FKUFRegiao  |
| CHECK CONSTRAINT  | CC + Nome da tabela sem TB + Nome da Coluna  | CCFuncionarioSexo  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo Atributo**  | **Nomenclatura**  | **Exemplo**  | **Uso**  | **Tipo SQL**  |
| Código  | CD+Descrição  | CDFuncionario CDCPF  | Identificadores numéricos ou alfanuméricos estruturados  | Numeric, int, char, smallint, tinyint  |
| Data/Hora  | DT+Descrição  | DTNascimento  | Determinação de dia, mês, ano e horário  | datetime  |
| Número  | NR+Descrição  | NRImovel  | Informação de grandeza numérica  | numeric, int, smallint, tinyint  |
| Nome  | NO+Descrição  | NOTrabalhador  | Denominação própria. Nomes de pessoas, cidades  | char, varchar (mínimo 70)  |
| Valor  | VL+Descrição  | VLSalario  | Valores monetários  | numeric (nn,2)  |
| Tipo  | TP+Descrição  | TPAssunto  | Caracterização de atributos (domínios) Domínios até 5 valores usar Check Constraint  | numeric, smallint, tinyint |
| Sigla  | SG+Descrição  | SGOrgao  | Atributos codificadores do tipo caracter  | char, varchar  |
| Identificador (auto-incremento)  | ID+Descrição  | IDUsuario  | Atributos identificadores, inclusive gerados automaticamente pelo BD ou sistema  | numeric, int, smallint, bigint  |
| Situação/status  | ST+Descrição  | STAndamento  | Status ( até 5 valores)  | smallint, char(1) , bit |
| Indicador  | IC+Descrição  | ICPago  | Duas condições contrapostas. Booleanos (set ou reset – apenas 1 bit)  | bit NOT NULL  |
| Endereço  | ED+Descrição  | EDFiscal  | Atributos alfanuméricos contendo informações de endereço do tipo rua, bairro, número, etc.  | varchar, char  |
| Matrícula  | MT+Descrição  | MTFuncionario  | Atributos codificadores alfanuméricos  | char, varchar , numeric  |
| Memo  | MM+Descrição  | MMParecer  | Texto sem limite de tamanho  | text (> 8k) varchar (< 8k)  |
| Descrição  | DS+Descrição  | DSClasse  | Descrição com limite de tamanho  | char, varchar (Mínimo 100)  |
| Texto  | TE+Descrição  | TEParecer  | Texto com limite de tamanho  | varchar  |
| Taxa  | TX+Descrição  | TXComissao  | Taxas  | numeric  |
| Quantidade  | QT+Descrição  | QTAcessos  | Número de unidades  | Numeric, int,smallint, tinyint  |
| Link  | LK+Descrição  | LKPagina  | Atributos alfanuméricos contendo endereços de páginas WEB  | varchar  |
| Binário  | BI+Descrição  | BIFoto  | Atributos BLOB (imagens, vídeo, som)  | image  |
| Endereço Eletrônico  | EE+Descrição  | EEUsuario  | Atributos alfanuméricos contendo endereços de e-mail  | varchar  |
| Sequencial  | SQ+Descrição  | SQCarteira  | Sequencial  | numeric  |
| Observação  | OB+Descrição  | OBCargo  | Observações genéricas textuais sem formatação definida  | varchar  |
| Percentual  | PC+Descrição  | PCSalario  | Percentual de valores  | numeric  |

## Exemplo de modelo de dados



Figura 1 – Modelo de Dados SAA

# REFERÊNCIAS

* COUGO, Paulo; **Modelagem conceitual e projeto de banco de dados**, Rio de Janeiro, Elsevier, 1997, 21ª reimpressão
* C. J. Date; **Introdução a sistemas de banco de dados**, Rio de Janeiro Elsevier, 2003, 9ª reimpressão.
* PS-MCTIC – Processo de Software do Ministério de Ciência, Tecnologia, Inovações e Comunicações.