

Metodologia de Administração de Dados

MAD





2024 Ministério da Saúde

Esta obra é disponibilizada nos termos da Licença Creative Commons – Atribuição – Não Comercial – Compartilhamento pela mesma licença 4.0 Internacional. É permitida a reprodução parcial ou total desta obra, desde que citada a fonte.

MINISTRA DE ESTADO

Nísia Verônica Trindade Lima

SECRETÁRIA DE INFORMAÇÃO E SAÚDE DIGITAL

Ana Estela Haddad

DIRETORA DO DEPARTAMENTO DE INFORMAÇÃO E INFORMÁTICA DO SUS

Paula Xavie dos Santos

COORDENADOR-GERAL DE INFRAESTRUTURA E SEGURANÇA DA INFORMAÇÃO

Ramon Moreno de Matos Vieira

COORDENADORA DA COORDENAÇÃO DE GESTÃO DE BANCO DE DADOS

Núbia Moreira dos Santos

EQUIPE TÉCNICA

Núbia Moreira dos Santos, Pedro Alexandre Dobbin, Isabela Fonseca Lima, Italo Araujo Trovão, Gislaine Mabel de Nobrega Trindade, Dawid Lopes de Souza, Erci dos Santos Junior, Fernando Benayon Santos, Murilo Cesar Lima, Yuri Henrique da Costa Ferreira, Mateus Ciríaco Amaral, Guilherme da Costa Silva, Mateus Brunoro de Souza, Gabriel Gonçalves Nogueira, Luiz Gustavus Capellini, Sebastião Rodrigues da Costa Neto, Carlos Henrique Jacinto, Alan Brito Muniz, Romário Gomes Barbalho e Renato de Sant'Anna Lopes.

ORGANIZAÇÃO E ELABORAÇÃO

Núbia Moreira dos Santos, Pedro Alexandre Dobbin, Isabela Fonseca Lima, Italo Araujo Trovão, Gislaine Mabel de Nobrega Trindade, Erci dos Santos Junior, Dawid Lopes de Souza, Luana Larissa Andrade, Vivian Furlan de Camargo Ramos Mendonça, Blanda Helena de Mello, Igor Oliveira Vieira, Walter da Silva Domingos.

PROJETO GRÁFICO E CAPA

Juliana Pires de Abreu Batista

HISTÓRICO DE REVISÃO

DATA	AUTOR	DESCRIÇÃO	VERSÃO
15/01/2021	Eliete Colucci Sousa	Criada a 1 versão da Metodologia de Administração de Dados (MAD).	1.0
20/05/2023	Gislaine Mabel da N. Trindade	Revisão Técnica do documento e atualização, fluxogramas, análise no que concerne a LGPD para contemplar o cenário atual do DATASUS.	1.1
23/09/2024	Isabela Fonseca Lima, com apoio de Mateus Brunoro de Souza, Luana Larissa Andrade e Gislaine Mabel da N. Trindade.	Revisão geral do documento. Atualização e inclusão de novos conteúdos. Mapamento de processos.	2.0

APRESENTAÇÃO

No mundo contemporâneo, onde a informação é um ativo extremamente valioso, a correta administração dos dados e dos bancos de dados é essencial para garantir eficiência nos processos e subsidiar a tomada de decisões estratégicas. Este desafio é cada vez maior, sobretudo porque a tecnologia evolui rapidamente, a quantidade de informações que circulam pela Internet em segundos é bastante expressiva e a falta de critérios claros para avaliar fontes e garantir a qualidade da informação pode resultar na disseminação de conteúdo de baixa qualidade e confiabilidade.

A capacidade de coletar, analisar e interpretar grandes volumes de dados permite que órgãos públicos se adaptem rapidamente às necessidades da população, identifiquem oportunidades de melhoria e mitiguem riscos. Dessa forma, uma gestão adequada dos dados não apenas optimiza processos internos, mas também contribui para a transparência e a eficácia na prestação de serviços internos e externos. Além disso, contribui de maneira expressiva para a tomada de decisões rápidas e assertivas, necessidade que a cada dia é mais latente.

Sendo assim, melhorar a qualidade das informações e garantir a organização, padronização, qualidade e gestão da informação é fundamental que seja realizado uma gestão de informações eficiente e isso é alcançado por meio da definição de padrões, procedimentos, métodos, técnicas e outros.

Este documento tem como finalidade propor uma **Metodologia de Administração de Dados (MAD)**, contendo a regulamentação e a descrição dos padrões e procedimentos a serem adotados em relação à Administração de Dados no âmbito do Ministério da Saúde.

METODOLOGIA DE ADMINISTRAÇÃO DE DADOS

Este documento contém informações a respeito dos papéis e responsabilidades da equipe de Gestão de Banco de Dados do Ministério da Saúde, além de apresentar o padrão de nomenclatura para objetos de banco de dados, o processo de modelagem de dados (importante para a gestão do modelo corporativo e seus indicadores), a privacidade de dados, o tratamento de dados, a anonimização de dados, os esquemas, a criação de usuários, a hospedagem e internalização de sistemas sob a ótica de banco de dados entre outros assuntos.

Essas etapas são fundamentais para garantir a correta administração dos dados e por este motivo, é importante que este documento seja atualizado de forma contínua, de acordo com as necessidades apresentadas ao longo dos tempos.

SUMÁRIO

1. Introdução	8
1.1 Contexto de fluxo de dados na RNDS	8
1.2 Modelo Informacional	9
1.3 Modelo Computacional	9
1.4 Fluxo para propor novos modelos de Interoperabilidade para a RNDS	10
1.5 Dados da RNDS	10
2. Padrão de interoperabilidade HL7 FHIR - Adotado pela RNDS	12
2.1 Identidade do Recurso e Metadados	13
2.2 Dados padrão	13
3. A especificação FHIR	14
4. Objetivo da FHIR na MAD	15
4.1 Semântica e FHIR	16
5. Papéis e responsabilidades da equipe de banco de dados	17
5.1 Administração de Dados (AD)	17
5.2 Administração de Banco de Dados (DBA)	18
5.3 Engenheiro de dados	19
6. Padrão de nomenclatura	21
7. Modelagem de dados	22
7.1 O Processo de modelagem de dados (PMD)	22
7.2 Detalhamento do processo de modelagem de dados	23
7.3 Análise do modelo de dados	23
7.4 Analisar script	24
8. Banco de dados	27
8.1 Esquema	27
8.2 Criação de Usuários dos esquemas	27
8.3 Dono de esquema (owner)	28
8.4 Replicação de Esquemas	28
8.5 Anonimização de dados	28
8.6 Acesso a Base de Dados	29
9. Procedimentos para acesso aos banco de dados	30
9.1 Regras de preenchimento	31
9.2 Renovação da permissão e alteração de IP	32

SUMÁRIO

9.3 Permissão emergencial	32
9.4 Permissão DDL	33
9.5 Cessão de dados (extração)	34
10. Hospedagem e internalização de sistemas	36
10.1 Internalização	36
10.2 Hospedagem	36
11. Conclusão	38
Anexo 1 - Orientações para construção de modelo de dados com qualidade	39
Anexo 2 - Padrão de nomenclatura	74
Anexo 3 - Guia rápido de modelagem de dados	100
Anexo 4 - Artefatos	107

1. INTRODUÇÃO

A Rede Nacional de Dados em Saúde (RNDS) é a plataforma nacional de interoperabilidade (troca de dados), instituída pela Portaria GM/MS nº 1.434, de 28 de maio de 2020 como medida para enfrentamento da pandemia de COVID-19. A RNDS, além de ser um projeto estruturante, é um programa do Governo Federal voltado para a transformação digital da saúde no Brasil e tem o objetivo de promover a troca de informações entre os pontos da Rede de Atenção à Saúde, e assim promover a transição e continuidade do cuidado nos setores público e privado.

Na perspectiva de um estabelecimento de saúde, a RNDS desempenha um papel central, possibilitando que um estabelecimento de saúde compartilhe informações acessíveis e qualificadas para o repositório centralizado, de onde outros estabelecimentos podem acessá-las uma vez que estejam integrados, promovendo a continuidade do cuidado do cidadão.

Quando um estabelecimento de saúde se integra à RNDS, ele pode contribuir com as informações em um contexto centrado no cidadão assistido, bem como, pode consumir informações geradas pelos demais estabelecimentos, em um contexto de atendimento ao cidadão – as informações disponíveis na RNDS podem ser acessadas apenas em um contexto de atendimento, não estão disponíveis para uso indiscriminado. É importante ressaltar que a segurança da informação é garantida por meio de métodos criptográficos, a fim de salvaguardar que os dados individuais dos pacientes permaneçam protegidos contra acessos não autorizado entre sistemas.

1.1 Contexto de fluxo de dados na RNDS

A RNDS como plataforma de interoperabilidade necessita de um padrão para permitir a troca de informações, motivo pelo qual foi estabelecido, a nível nacional, o uso obrigatório do padrão de interoperabilidade em saúde para trocas com a RNDS, o HL7 FHIR¹². O padrão FHIR (*Fast Healthcare Interoperability Resources*) é um padrão desenvolvido pela organização *Health Level Seven International* (HL7) para a troca de informações eletrônicas em saúde, credenciada pela ANSI-standards (*American National Standards*) para a definição de frameworks e padrões para a troca, integração, compartilhamento e recuperação de informações eletrônicas.

Nesse cenário, como em outros padrões modernos, o padrão FHIR adota uma estratégia de camadas, que visa a separação de uma camada para a tomada de decisões de especialistas em saúde, e outra para a representação destas decisões abstraídas da área da saúde para uma representação computacional. Os profissionais da saúde que definem as informações que irão compor um conjunto de dados, são especialistas em seus contextos clínicos, por exemplo obstetrícia, transplante, imagens diagnósticas, entre outros, e não tem contato com a representação computacional do padrão FHIR. Essa equipe especialista é responsável por selecionar os componentes conceituais para contemplar os requisitos de informações de saúde em um contexto clínico e, posteriormente, a equipe de tecnologia é responsável por representar estas informações computacionalmente, nos diferentes formatos permitidos no padrão HL7 FHIR³ (JSON, XML e RDF). Essa separação é denominada, no contexto do Ministério da Saúde e RNDS, de Modelo Informacional e Modelo Computacional.

1.2 Modelo Informacional

Um Modelo Informacional (MI) é uma abstração conceitual que descreve um conjunto de informações dentro do contexto da saúde. Esse modelo é elaborado por especialistas em saúde, e resulta no desenvolvimento de um Modelo de Informações para um contexto clínico, que é instituído independente de padrões, essencial para atingir objetivos de interoperabilidade. Uma vez instituído o MI, conforme as definições estabelecidas na norma ISO/TS 13972:2015(E), que orienta, com especificações detalhadas de modelos clínicos, os processos e características, o MI representa, conceitualmente, a estrutura de informações que se deseja compartilhar em um contexto clínico.

Uma vez que a RNDS troca informações seguindo um padrão, definido como o padrão HL7 FHIR, é necessário definir os contextos clínicos que a RNDS espera receber. Atualmente, já existem desenvolvidos os modelos: Registro de Imunobiológico Administrado (RIA), MI instituído conforme portaria Nº 25 de 25 de novembro de 2023, Resultados de Exames Laboratoriais (REL), MI instituído conforme portaria atualizada Nº 883 de 29 de novembro de 2022, Registro de Atendimento Clínico (RAC), MI instituído conforme portaria Nº 234 de 18 de julho de 2022, Sumário de Alta Hospitalar (SA), MI instituído conforme portaria Nº 701 de 29 de setembro de 2022. A RNDS está em desenvolvimento e constante processo de evolução, a publicação de novos modelos faz parte de seu plano para garantir a interoperabilidade em todos os níveis de atenção em escala nacional.

1.3 Modelo Computacional

Um Modelo Computacional (MC) é a representação em formato computacional das informações que foram definidas no Modelo Informacional (MI), seguindo as normas estabelecidas pela ISO/TS 13972:2015(E), e as definições quanto a restrições e terminologias nas portarias.

¹ <https://www.hl7.org/fhir/>

² <https://www.hl7.org/>

³ <https://www.hl7.org/fhir/resource-formats.html>

É apenas no momento de definir o Modelo Computacional que o padrão de interoperabilidade adotado será evidenciado, uma vez um Modelo Informacional é independente de padrões, podendo ser traduzido e representado em qualquer padrão. Neste cenário, é no MC que as restrições nacionais necessárias são definidas, tais como definições de terminologias, dicionários, vocabulários, classificações e mapeamento para atender o escopo do Modelo Informacional, não só do ponto de vista técnico e estrutural, mas semântico.

Retomando o conceito da estratégia de um modelo de camadas, uma vez definido o modelo informacional (MI), o próximo passo é representar estas informações computacionalmente, através do Modelo Computacional (MC) em um formato compartilhável, possível de ser lido por máquinas e humanos, bem como garantindo a qualidade da informação trocada. A modelagem do computacional não segue necessariamente a mesma ordem lógica das informações da modelagem informacional. A modelagem computacional vem para formalizar a modelagem da informação de forma digital e computacionalmente compartilhável, e pode nortear a criação de um único recurso FHIR ou da composição de vários, relacionados entre si, de acordo com o escopo do que foi definido no modelo informacional.

1.4 Fluxo para propor novos modelos de Interoperabilidade para a RNDS

De forma geral, a proposição de novos modelos informacionais e computacionais é feita através da proposição da necessidade e justificativa para o Comitê Gestor de Saúde Digital (CGSD) instituído pela resolução CIT nº 05, de 25 de agosto de 2016, o qual avalia e toma providências de forma tripartite. Tratando-se de uma plataforma de dados nacional, a RNDS apenas recebe dados que estejam no padrão HL7 FHIR, desta forma, o processo de recebimento de dados deve, obrigatoriamente, passar pela qualificação e padronização dos dados para atender ao padrão adotado.

1.5 Dados da RNDS

Em um país de dimensões continentais como o Brasil, esse desafio se torna ainda mais complexo, devido a grande quantidade de ministérios, órgãos, empresas públicas e privadas, secretarias, dentre outros presentes nos mais de 5570 municípios e Distrito Federal. Sistemas e artefatos tecnológicos já se tornaram itens indispensáveis na rotina diária. De acordo com [e-Estonia 2020], 67% da população da Estônia utiliza a carteira digital de identidade regularmente, 99% dos serviços públicos estão online, 2773 diferentes serviços podem se comunicar por uma camada de troca de dados chamada X-Road. A incrível digitalização dos serviços faz da Estônia um exemplo tecnológico de qualidade, rapidez e sistematização não burocrática a ser seguida. O estudo de técnicas que visam a troca de dados entre sistemas de maneira segura, rápida e íntegra pode ser avaliado como um fator que alavanca o processo de digitalização de serviços públicos, reduzindo assim tempo de trabalho, tempo de resposta em atendimentos e solicitações da sociedade e público em geral e além do mais, pode fornecer mais transparência e padronização.

Quando se trata de alcançar a interoperabilidade em ambientes complexos de saúde digital, o conceito de barramento de interoperabilidade emerge como uma estrutura fundamental que engloba três momentos críticos: ingestão, armazenamento e exposição de dados (disponibilização). Cada um desses momentos desempenha um papel essencial na garantia de que sistemas de informação em saúde possam compartilhar informações de maneira eficaz e ordenada, beneficiando pacientes e profissionais de saúde.

A exposição de dados por meio de um aplicativo com interface front-end ou API (*Application Programming Interface*), representam duas abordagens distintas para disponibilizar informações. O primeiro, refere-se a um aplicativo com interface com o usuário de forma visual, onde os dados são apresentados e permite a interação pelos usuários finais. Por outro lado, uma API traz um catálogo de serviços de dados, que permite a sistemas e aplicativos que se comunicarem e troquem dados de forma programática, uma vez que as regras definidas para integração/autenticação são atendidas. As APIs não possuem uma interface de usuário visível, projetadas para que sistemas interajam de forma automatizada. Essa abordagem é valiosa quando se deseja permitir a integração de sistemas, facilitando a troca de informações entre diferentes aplicativos, sistemas e plataformas de maneira eficiente e padronizada. Neste cenário, os dados podem ser acessados através da plataforma fornecida, por meio de sua interface, seja para o cidadão – através do aplicativo de celular, quanto para o profissional – no contexto de um atendimento em prontuários integrados, bem como no consumo automatizado e sistematizado dos sistemas através das APIs.

Os canais oficiais de disseminação de informações sobre a RNDS são os meios autorizados e estabelecidos pelo Datasus e Ministério da Saúde para compartilhar os manuais, portarias e comunicados sobre os modelos de interoperabilidade em produção, em desenvolvimento e previstos. Através desses canais, o Datasus garante que as informações importantes cheguem aos implementadores, gestores de saúde e partes interessadas de forma apropriada, promovendo a transparência e a coesão nas comunicações.

2. PADRÃO DE INTEROPERABILIDADE HL7 FHIR – ADOTADO PELA RNDS

A RNDS apenas recepciona e troca de dados que estejam no padrão de interoperabilidade definido pelo Ministério da Saúde, o HL7 FHIR. Neste cenário, um desafio central para a adoção de padrões de saúde é como lidar com a grande variabilidade causada pelos diversos processos de saúde. Com o tempo, mais informações são adicionadas à especificação, acentuando gradualmente custo e complexidade às implementações resultantes. A alternativa é contar com personalizações, mas elas também criam muitos problemas de implementação. Neste contexto, as soluções FHIR são construídas a partir de um conjunto de componentes modulares denominados “Recursos”. Esses recursos podem ser facilmente reunidos em sistemas funcionais que resolvem problemas clínicos e administrativos do mundo real por uma fração do preço das alternativas existentes. O FHIR é adequado para uso em uma ampla variedade de contextos – aplicativos de telefonia móvel, comunicações em nuvem, compartilhamento de dados baseado em EHR, comunicação de servidor em grandes provedores institucionais de saúde e muito mais. O padrão é publicado de forma mista, com partes normativas, que são mantidas estáveis para os implementadores, e partes ainda em uso experimental.

O HL7 monitora ativamente as implementações para continuar melhorando as especificações e responder às necessidades em sua plataforma ⁴disponibilizada para a comunidade internacional. Apesar de não haver um consenso mundial sobre padrões de saúde, devido às muitas vantagens que o FHIR oferece, ele tem sido amplamente utilizado, e sua comunidade está crescendo rapidamente. Orienta-se acompanhar o canal oficial de discussões para novas implementações e atualizações do padrão HL7 FHIR. O FHIR possui recursos para conceitos administrativos como paciente, provedor, organização e dispositivo, bem como uma ampla variedade de conceitos clínicos que abrangem problemas, medicamentos, diagnósticos, planos de cuidados, questões financeiras e muito mais.

⁴ <https://confluence.hl7.org/display/FHIR>

2.1 Identidade do Recurso e Metadados

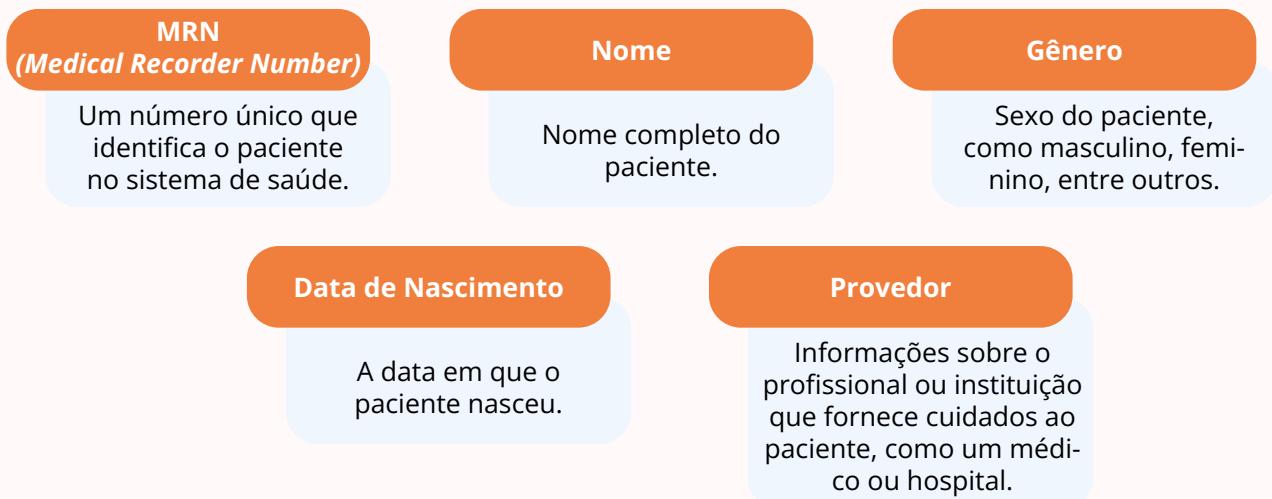
O recurso “*Patient*” por exemplo, representa as informações relacionadas a uma pessoa no sistema de saúde. Este recurso pertence à camada básica da estrutura do FHIR. Cada instância desse recurso é identificada de maneira única, geralmente expressa por meio de uma URL. Além disso, o recurso contém metadados, que oferecem detalhes sobre o recurso, como sua versão, data de criação ou última atualização, e a entidade responsável por essas ações. Esses metadados desempenham um papel crucial no rastreamento e gerenciamento das informações, garantindo que os sistemas operem com os dados mais recentes e precisos.

Definição de extensões através de uma URL

FHIR é projetado para ser extensível, permitindo que as organizações adicionem informações que não estão no seu padrão original, conhecido como recurso “canônico”. Uma “extensão” permite que novos dados sejam incluídos em um recurso padrão original, como o recurso *Patient*. Cada extensão tem uma URL única que aponta para sua definição, garantindo que qualquer sistema que encontre essa extensão possa entender seu propósito e significado. Assim, enquanto o FHIR fornece um conjunto de campos padrão, as organizações podem adaptá-lo às suas necessidades específicas usando extensões, e criando perfis específicos.

2.2 Dados padrão

No contexto do recurso canônico “*Patient*”, os dados padrão podem incluir campos como:



O padrão FHIR, desenvolvido pela HL7, tem experimentado várias versões desde o seu lançamento, cada uma introduzindo melhorias, correções e novos recursos para responder às necessidades em constante evolução da comunidade de saúde. Na gestão de versionamento, é crucial para a metodologia de administração de dados reconhecer e gerenciar essas diferentes versões, tal como orienta a documentação oficial para os níveis de maturidade⁵.

⁵ <https://hl7.org/fhir/R4/foundation-module.html>

3. A ESPECIFICAÇÃO FHIR

O padrão FHIR é estruturado em uma série de módulos de especificação, cada um abordando diferentes aspectos e componentes da interoperabilidade em saúde. Em termos gerais, a especificação FHIR é dividida nos conjuntos de módulos, abaixo traz-se alguns principais, conforme a versão que a própria RNDS implementa, a R4:

- **Fundação⁵:** A infraestrutura básica de definição sobre a qual o restante da especificação é construído;
- **Suporte ao Implementador⁶:** Serviços para ajudar os implementadores a fazer uso da especificação;
- **Segurança e Privacidade⁷:** Documentação e serviços para criar e manter segurança, integridade e privacidade;
- **Conformidade⁸:** como testar a conformidade com a especificação e definir guias de implementação;
- **Terminologia⁹:** Uso e suporte de terminologias e artefatos relacionados;
- **Dados vinculados¹⁰:** métodos definidos de troca de recursos;
- **Administração¹¹:** Recursos básicos para rastrear pacientes, profissionais, organizações, dispositivos, substâncias etc.;
- **Clínico¹²:** conteúdo clínico central, como problemas, alergias e processo de atendimento (planos de atendimento, encaminhamentos);
- **Medicamentos¹³:** gerenciamento de medicamentos e rastreamento de imunização;
- **Diagnóstico¹⁴:** Observações, relatórios de diagnóstico e solicitações de conteúdo relacionado.

Estes módulos são projetados para fornecer uma abordagem modular e flexível para representar, trocar e gerenciar informações de saúde. Eles cobrem uma ampla gama de tópicos, desde recursos clínicos básicos, como “Paciente” ou “Observação”, até componentes mais avançados, como operações, extensões e terminologias. Os recursos têm uma ampla gama de usos, desde conteúdo clínico puro, como planos de cuidados e relatórios de diagnóstico, até infraestrutura pura, como cabeçalho de mensagem e declarações de capacidade. Todos eles compartilham características técnicas comuns, mas podem ser usados de formas totalmente diferentes.

6 <https://hl7.org/fhir/R4/implsupport-module.html>

10 <https://hl7.org/fhir/R4/exchange-module.html>

14 <https://hl7.org/fhir/R4/diagnostics-module.html>

7 <https://hl7.org/fhir/R4/seccpriv-module.html>

11 <https://hl7.org/fhir/R4/administration-module.html>

8 <https://hl7.org/fhir/R4/conformance-module.html>

12 <https://hl7.org/fhir/R4/clinicalsummary-module.html>

9 <https://hl7.org/fhir/R4/terminology-module.html>

13 <https://hl7.org/fhir/R4/medications-module.html>

4. OBJETIVO DA FHIR NO MAD

O principal propósito da integração do FHIR em nossa metodologia de administração de dados é alcançar a interoperabilidade aprimorada. Isso significa que buscamos garantir que todos os sistemas de informação em saúde dentro da organização estejam em harmonia com o padrão FHIR, possibilitando uma troca de informações de saúde eficaz e segura. Além disso, temos a intenção de padronizar todos os nossos registros de saúde eletrônicos e outros dados pertinentes usando os recursos e estruturas fornecidos pelo FHIR. Este padrão não apenas facilitará a interoperabilidade, mas também melhorará a consistência e a qualidade dos dados.

No entanto, é importante destacar que reconhecemos a necessidade de flexibilidade. Portanto, outro objetivo crucial é implementar a extensibilidade do FHIR, permitindo que a organização personalize e expanda os dados conforme as necessidades específicas. Isso será feito mantendo os mais altos padrões de segurança e garantindo conformidade com os regulamentos pertinentes.

A modelagem FHIR segue uma abordagem de composição, na qual definimos as informações básicas, as definições de negócios e as estratégias para atingir um objetivo de interoperabilidade específico. A modelagem computacional implica em encontrar recursos, perfis ou composições FHIR adequadas para transmissão computacional de uma mensagem, podendo utilizar recursos canônicos ou especializados, conforme necessário. Com o FHIR, os casos de uso específicos geralmente são implementados combinando recursos por meio do uso de referências de recursos. Embora um único recurso possa ser útil por si só para um determinado caso de uso, é mais comum que os recursos sejam combinados e adaptados para atender aos requisitos específicos do caso de uso. Dessa forma, a integração do FHIR não só estabelece um padrão sólido de interoperabilidade, mas também permite a personalização e adaptação para atender às necessidades específicas da organização em relação aos dados de saúde.

Dois tipos especiais de recursos são usados para descrever como os demais recursos são definidos e usados:

- **Declaração de capacidade (CapabilityStatement):** descreve as interfaces que uma implementação expõe para troca de dados.

- **Definição da Estrutura (StructureDefinition):** fornece regras adicionais que servem para restringir se opcional ou obrigatória, cardinalidade, ligações de terminologia, tipos de dados e extensões definidas nos recursos usados pela implementação.

Sendo assim, o alicerce básico do FHIR é um recurso¹⁵. Todo conteúdo intercambiável é definido como um recurso. Todos os recursos compartilham o seguinte conjunto de características:

- Uma maneira comum de defini-los e representá-los, construídos a partir de tipos de dados reutilizáveis comuns;
- Um conjunto comum de metadados;
- Uma parte legível por humanos;

É impraticável modelar a totalidade dos dados de saúde num único modelo de informação. Cada iniciativa de modelagem em saúde, desde especificações de mensagens HL7 versão 2 até recursos FHIR, decompõe o domínio de saúde em subdomínios ou fragmentos de modelo de informações menores e mais gerenciáveis. Com o FHIR, cada recurso é essencialmente um trecho do domínio mais amplo de informações sobre saúde. Ao dividir o modelo de informação sobre cuidados de saúde em partes menores (recursos, composições ou perfis para FHIR), é importante ter uma estrutura e um conjunto de diretrizes para promover a consistência e a integridade dentro das estruturas de recursos e na forma como os recursos se referenciam entre si.

4.1 Semântica e FHIR

A semântica refere-se ao significado, desempenhando um papel crucial na administração de dados para garantir que as informações sejam compreendidas de maneira uniforme em diferentes sistemas e contextos. No setor de saúde, a necessidade de uma interpretação consistente é tão fundamental que faz parte da interoperabilidade entre sistemas de informação. Ao incorporar princípios semânticos, o FHIR garante que os dados de saúde sejam não apenas transferidos, mas também interpretados de forma consistente e, independentemente do sistema ou plataforma em uso, o profissional de saúde que acessa os dados pode tomar uma decisão com base em dados contextualizados.

O FHIR atende a essa demanda, estabelecendo padrões para a troca de informações de saúde com precisão semântica. Ele incorpora elementos como “*CodeableConcept*”, que traduz conceitos codificados em textos comprehensíveis; “*ValueSet*”, que define conjuntos específicos de códigos para contextos particulares; e “*CodeSystem*”, que detalha sistemas de codificação específicos. Central a isso, a “terminologia” no FHIR abrange o conjunto de códigos e conceitos usados em registros de saúde eletrônicos, refletindo o estudo de termos e suas relações em um domínio específico.

15 <https://hl7.org/fhir/R4/resource.html>

5.PAPÉIS E RESPONSABILIDADES DA EQUIPE DE BANCO DE DADOS

5.1 Administração de Dados (AD)

O Administrador de Dados é o responsável por manter a qualidade e a atualização dos modelos de dados corporativos, bem como de sua documentação, disponibilizando-a para o entendimento de todos, garantindo a integridade entre os modelos de dados¹ e mantendo-os aderentes às regras de negócio definidas pelo Ministério da Saúde (MS), independentemente da tecnologia e da plataforma a ser utilizada.

Compete a este perfil gerenciar o repositório dos modelos de dados e metadados corporativo do MS, documentando as etapas, eliminando redundâncias e garantindo a segurança e qualidade de todos os dados. Para isso, é necessário que o Gestor da Informação realize a manutenção dos dados, de forma que os bancos possam ser atualizados, possibilitando confiança e precisão nas informações estratégicas.

Este perfil também ajuda a promover a conceituação, a padronização, a segurança, a integridade, o compartilhamento e a qualidade dos dados, visando o aperfeiçoamento dos processos de gestão por meio do uso da informação.

Principais atividades de um AD

- Desenvolver, manter e atualizar o Dicionário de Dados Corporativo;
- Manter os dados corporativos atualizados;
- Apoiar a equipe de desenvolvimento a trabalharem na modelagem de dados de suas aplicações, viabilizando a consistência e normalização dos dados, compartilhamento e integração de dados, eliminação (ou minimização) de dados redundantes e garantir que o modelo lógico e físico de dados seja implementado corretamente;
- Analisar pedidos de termos de responsabilidades para garantir que as informações estejam

disponíveis apenas para quem precisar delas;

- Definir os níveis de integridade e segurança dos dados nos diversos níveis em que a informação solicitada;
- Definir os níveis de acesso aos dados para garantir que as informações estejam disponíveis somente a quem precisar delas;
- Analisar scripts de criação, alteração de objetos, e atualização de dados;
- Extrações (ETL) de dados das bases dos sistemas da saúde para atendimento de auditorias por partes de outros órgãos do governo;
- Anonimizar dados;
- Administrar o power center (criação de usuários, permissionamento, criação de projetos); e
- Monitorar cargas no power center,

5.2 Administração de Banco de Dados (DBA)

O DBA é o responsável por administrar os bancos de dados, portanto, organizam e controlam uma variedade de Sistemas de Gerenciamento de Banco de Dados (SGBDs), que incluem Oracle, PostgreSQL, MySQL, SQL Server, Aurora, Redshift, Greenplum e MongoDB.

Além disso, também preparam a infraestrutura necessária para disponibilizar os bancos de dados, providenciam a otimização de uma base de dados existente, garantem o correto funcionamento, a segurança, a integridade e a escalabilidade por meio de atualizações, monitoramento e aprimoramento do sistema.

Principais atividades de um DBA

- Instalar e configurar os SGBD's;
- Gerenciar a segurança dos dados, incluindo a configuração de permissões de acesso e autenticação;
- Planejar e executar rotinas de backup regulares para proteger contra perda de dados;
- Desenvolver estratégias de recuperação para restaurar dados em caso de falhas;
- Monitorar o desempenho do banco de dados e identificar áreas que precisam de otimização;

1 Um modelo de dados é uma representação abstrata que ilustra a estrutura, as interações e as limitações dos dados em um sistema de informação. Ele funciona como uma ferramenta essencial para organizar e compreender como os dados são coletados, armazenados e manipulados, facilitando o planejamento e a implementação eficaz de sistemas de informação. O tópico "Modelagem de Dados" neste documento explica os detalhes desta etapa.

- Ajustar consultas, índices e configurações para otimizar o desempenho;
 - Aplicar patches e atualizações de software para manter o SGBD atualizado e seguro;
 - Realizar migrações de banco de dados conforme a necessidade do ambiente;
 - Realizar tarefas de manutenção, como compactação de tabelas e reorganização de dados;
- Avaliar as necessidades futuras de armazenamento e desempenho e planejar a capacidade do banco de dados;
- Configurar sistemas de monitoramento para alertar sobre problemas de desempenho ou segurança;
- Responder a alertas e resolver incidentes em tempo hábil;
 - Diagnosticar e resolver problemas de banco de dados, como bloqueios, lentidão e falhas;
 - Documentar o ambiente de banco de dados; e
 - Apoiar áreas técnicas em esclarecimentos pontuais a respeito de projetos de banco de dados.

5.3 Engenheiro de dados

O Engenheiro de Dados é o responsável por criar mecanismos para buscar dados brutos e, por meio de ETL, transformá-los em insights estratégicos para melhorar os processos internos e as tomadas de decisões.

Principais atividades do Engenheiro de Dados

- Desenvolver, construir, testar arquiteturas e manter pipelines disponíveis para uso;
- Criar arquiteturas conforme solicitado pela área de negócios;
- Coletar, processar, transformar dados para aplicações que utilizarão Big Data;
- Identificar maneiras de melhorar os níveis de qualidade e confiabilidade dos dados;
- Preparar dados para modelagem preditiva e prescritiva;
- Direcionar os dados para o Data Lake;
- Implementar processos ETL via Power center, NIFI, Python e Airflow;

- Instalar, sustentar e administrar as Ferramentas ETL;
- Monitorar e realizar manutenções de Ferramentas e processos ETL;
- Modelagem de processos ETL e desenvolvimento de pipelines adequados ao negócio;
- Disseminação de dados no Opendatasus;
- Atualização de Code Systems;
- Recepção de novos profiles;
- Níveis de acesso aos projetos e ferramentas ETL;
- Elaborar documentações; e
- Apoio às áreas técnicas das secretarias na utilização do datalake e outros.

6. PADRÃO DE NOMENCLATURA

O padrão de nomenclatura para administração de dados é um conjunto de diretrizes que estabelece como os elementos de dados devem ser nomeados e organizados em um banco de dados ou sistema de informação. Esses padrões são importantes para garantir a consistência, clareza e compreensão entre diferentes usuários e sistemas. Entre outros benefícios, o padrão de nomenclatura ajuda a melhorar a qualidade das estruturas dos dados promovendo organização, padronização e facilitando o entendimento. Esse processo facilita a gerência dos dados e promove segurança, qualidade, monitoramento, privacidade, aderência e conformidade com as normas.

As definições contidas nesses documentos devem ser **obrigatoriamente seguidas nos modelos de dados da instituição** conforme portarias números 664 e 665 publicadas em 10/08/2016, que torna obrigatório que **todo desenvolvimento de sistema deverá seguir as metodologias MDS (Metodologia de Desenvolvimento de Sistemas) e MAD (Metodologia de Administração de Dados).**



Qualquer alteração de estrutura (Data Definition Language - DDL²), deve ser executada primeiro no ambiente de DESENVOLVIMENTO e, somente após o retorno com sucesso deverá ser solicitada para os outros ambientes. A aplicação no DESENVOLVIMENTO é **obrigatória**.

² A estrutura DDL (Data Definition Language) refere-se a um subconjunto de comandos SQL (Structured Query Language) usados para definir e gerenciar a estrutura de bancos de dados e seus objetos. Os comandos DDL permitem criar, modificar e excluir estruturas de dados, como tabelas, índices e esquemas.

7. MODELAGEM DE DADOS

Os modelos de dados são construídos com o objetivo de entender e registrar formalmente os dados de uma realidade de negócio da instituição, especificando-os em um artefato denominado Modelo de Dados. Essa prática consiste em levantar, analisar, categorizar, interligar e explorar todos os dados e tipos de informações que irão sustentar uma aplicação. Para isso, devem-se traduzir os requisitos de informações em entidades negociais com seus atributos e relacionamentos entre estas, incluindo as respectivas regras de gestão, associadas aos processos de decisão e operações institucionais. Com isso, obtemos modelos de dados:

- Em conformidade com os padrões definidos;
- Com qualidade e documentados, de acordo com boas práticas de mercado;
- Com reutilização de objetos de banco de dados; e
- Com estruturas mais organizadas, processo realizado por meio da Normalização das variáveis de uma tabela, quando viável.

7.1 O Processo de Modelagem de Dados (PMD)

O processo de modelagem dos dados aumenta a consistência da nomenclatura, regras, semânticas, padronização, segurança e ao mesmo tempo, otimiza a análise de dados.

Consiste na criação de estruturas no sistema de armazenamento, quer seja físico ou em nuvem, para possibilitar a associação e o resgate de informações em um determinado padrão e momento.

7.2 Detalhamento do Processo de Modelagem de Dados

O processo de modelagem de dados podem ser analisados sob duas óticas diferentes, do ponto de vista negocial e do ponto de vista técnico. A Coordenação de Gestão de Banco de Dados realiza apenas a análise técnica do modelo.

A análise técnica consiste em verificar se os modelos de dados (lógico e físico) estão aderentes às regras da MAD, bem como validar se as tabelas, colunas e seus relacionamentos estão de acordo com a documentação vigente, além de garantir a conformidade do script enviado com a linguagem utilizada no SGBD solicitado, ajudando a mitigar possíveis erros e falhas de execução destes scripts. Essa análise é importante para garantir a integridade dos dados.

Para realizar essa análise, é necessário que a área solicitante envie os documentos abaixo:

- Modelo de dados ou script de DDL; e
- Documentação de metadados negociais (e relacionamento com metadados técnicos).

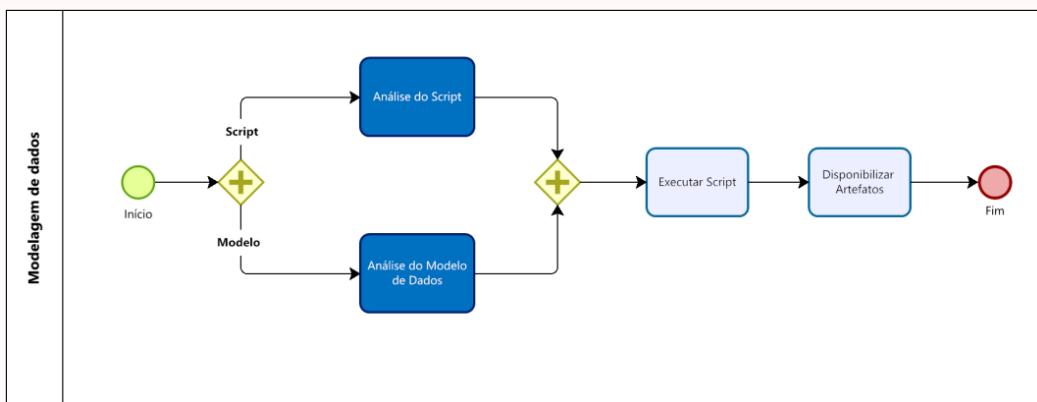


Figura 1. Modelagem de dados

7.3 Análise do Modelo de Dados

A equipe de Banco de Dados realiza a análise de modelo de dados elaborado pela equipe responsável pelo desenvolvimento do projeto. Esse processo é realizado desta forma:

ANÁLISE DE MODELO DE DADOS

Análise do Modelo de Dados

Objetivo:

Avaliar se o script enviado pela equipe de desenvolvimento responsável pelo projeto está em conformidade com a MAD.

Entradas:

- Projeto autorizado pelo Escritório de Projetos;
- Solicitação formalizada via Sistema Eletrônico de Informações - SEI;
- Modelo de dados na ferramenta padrão do Datasus ou script de criação dos objetos; e
- Documentação do projeto: Guia de Internalização e/ou Hospedagem.

Atividades:

- Análise da documentação do projeto;
- Reuniões para esclarecimentos; e
- Homologação do modelo de dados.

Saídas:

- Termo de Aceite do Modelo de Dados pela equipe responsável pelo projeto (necessário quando houver alteração do modelo proposta pela equipe de Banco de dados e aceita);
- Modelo de Dados disponibilizado na ferramenta padrão do Datasus; e
- Metadados no Repositório Central de Metadados para Governança de Dados.

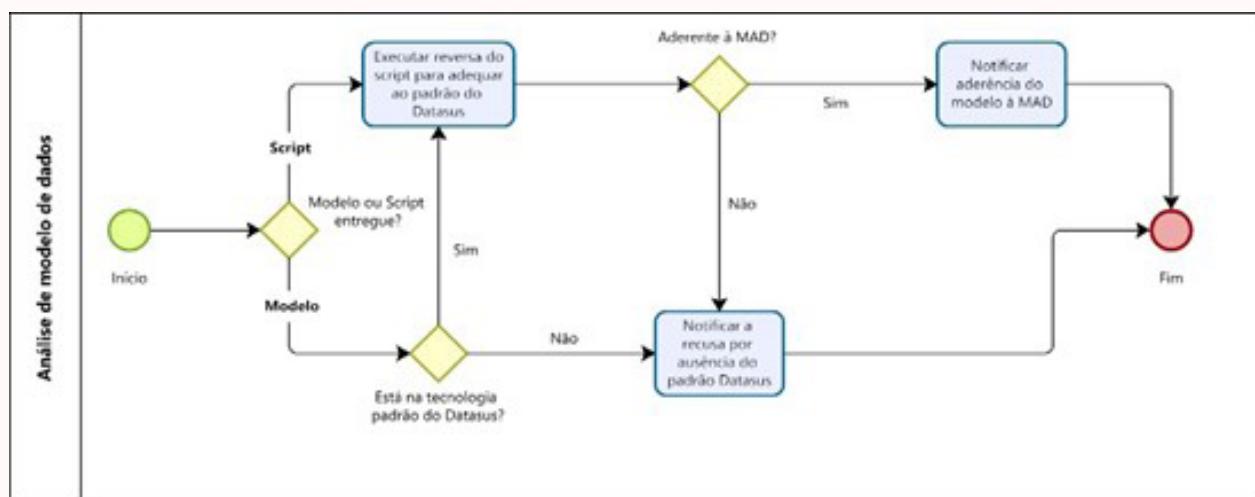


Figura 2. Análise de modelo de dados

7.4 Analisar script

A COBD realiza a avaliação técnica do script enviado pela equipe de desenvolvimento, e o mesmo deve estar em conformidade com a MAD em termos de padrões de nomenclatura, qualidade do modelo e documentação.

ANALISAR SCRIPT

Análise do Script

Objetivo:

Avaliar se o script enviado pela equipe de desenvolvimento responsável pelo projeto está em conformidade com a MAD.

Entradas:

- Solicitação via Central de Serviços; e
- Script a ser avaliado.

Atividades:

- Análise técnica do script verificando se está em conformidade com a MAD em termos de padrões de nomenclatura, qualidade de modelagem e documentação; e
- Pedido de alteração do script, ser for o caso, via Central de Serviços.

Saídas:

- Script avaliado;
- Objeto criado e/ou alterado no banco de dados; e
- Objeto disponibilizado no banco de dados.

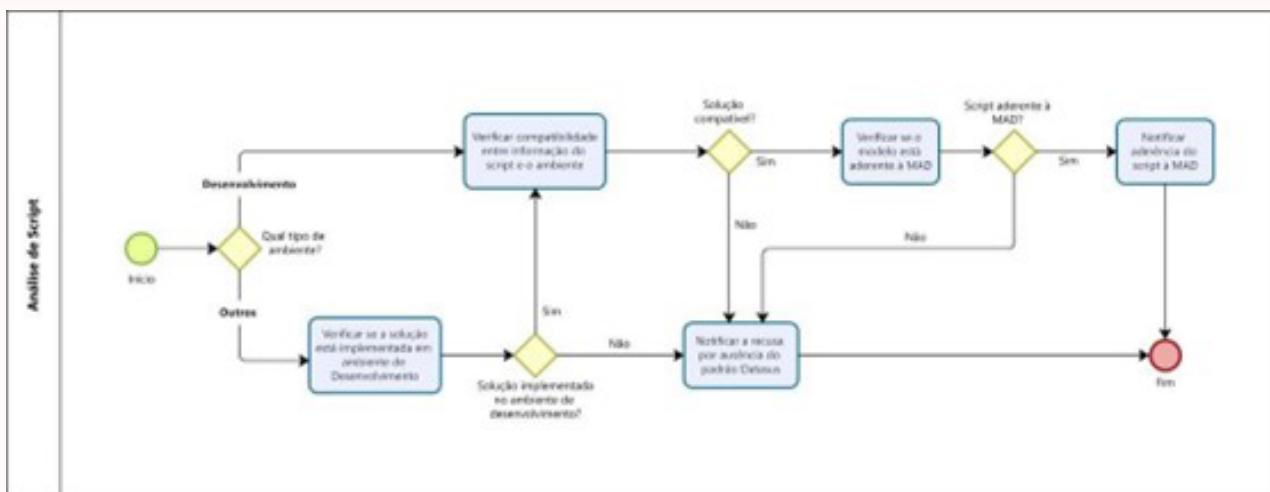
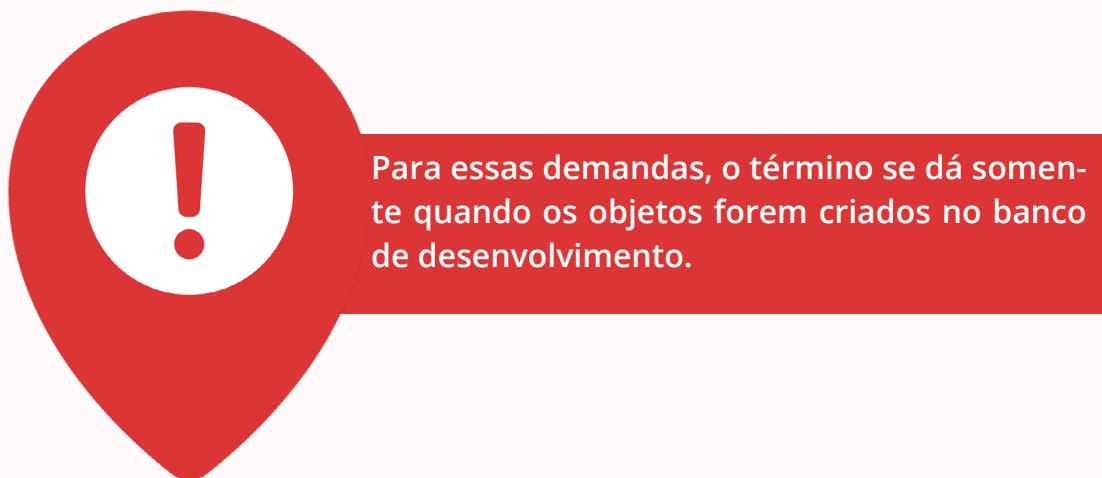


Figura 3. Fluxo de Análise de Script

Os processos de modelagem de dados são categorizados da seguinte forma:

ETAPA	DESCRIÇÃO	FAZ ANÁLISE TÉCNICA?	FAZ ANÁLISE NEGOCIAL?
1. Homologação de modelo	Modelos de dados que passam pelo processo de internalização, isto é, os modelos foram construídos por outra equipe, interna ou externa, e a COBD avalia se o modelo está ou não em conformidade com a MAD. Nessa categoria, é recebido um modelo de dados que deve estar na ferramenta de modelagem de dados adotada oficialmente pelo DATASUS ou um script contendo todos os objetos do projeto e com este será feita uma engenharia reversa para gerar o modelo.	Sim	Não
2. Avaliação de Script	A COBD recebe scripts, desenvolvidos internamente ou externamente, sendo avaliado se os objetos de banco de dados, que estão sendo criados ou mantidos, estão ou não em conformidade com a MAD em termos de nomenclatura, qualidade e documentação do modelo (análise técnica).	Sim	Não

Tabela 1 – Categorias de Modelagem de Dados



8. BANCO DE DADOS

Segundo Korth, um banco de dados “é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico”. O banco de dados é caracterizado pelo agrupamento de informação sobre um determinado assunto, armazenam os dados de forma organizada, gerando conjuntos de arquivos.

8.1 Esquema

Segundo Korth, um banco de dados “é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico”. O banco de dados é caracterizado pelo agrupamento de informação sobre um determinado assunto, armazenam os dados de forma organizada, gerando conjuntos de arquivos.

8.2 Criação de Usuários dos esquemas

Para criar e utilizar um esquema, por padrão, são criados dois usuários, sendo um de manutenção e outro de aplicação. O usuário de manutenção tem a finalidade de proporcionar atualizações e correções dos dados, enquanto o usuário de sistema tem como função principal a utilização do banco por parte do sistema (da aplicação), de acordo com a política de segurança e da arquitetura do banco de dados.

Usuário de Manutenção

Esse usuário tem a função de acessar a base de dados através de ferramentas via SQL, possuindo as permissões de seleção, inclusão, alteração e exclusão. [A nomenclatura seguirá o Padrão de Nomenclatura.](#)

Usuário de Aplicação

Esse usuário tem a função de acessar a base de dados através da aplicação, possuindo privilégios de seleção, inclusão, alteração e exclusão. [A nomenclatura seguirá o Padrão de Nomenclatura.](#)

[menclatura](#). As nomenclatura dos usuários, em ambos os casos, devem seguir o padrão para conforme documentação oficial.

8.3 Dono de esquema (owner)

Os sistemas de gerenciamento de banco de dados (SGBDs) implementam mecanismos de segurança baseados na propriedade dos objetos, permitindo a restrição de acesso aos dados. Cada objeto possui um usuário “owner”, que é o responsável por esse objeto. Por meio desse usuário, é possível organizar fisicamente os objetos, facilitando sua identificação.

Em um banco de dados, podem existir vários proprietários de diferentes objetos, incluindo tabelas, esquemas, visões, procedimentos, funções, entre outros.

8.4 Replicação de Esquemas

A replicação de esquema é permitida somente entre os ambientes de Desenvolvimento, Testes e Homologação. As informações disponíveis neste ambiente não podem ser iguais às informações do ambiente de produção.

OBS: Não é permitido replicar dados de produção para os ambientes de Desenvolvimento, Testes e Homologação.

8.5 Anonimização de dados

Dados pessoais são qualquer informação que possa identificar ou conduzir à identificação de uma pessoa, direta ou indiretamente. Para facilitar o entendimento, são exemplos de dados pessoais: CPF, Endereço, E-mail, CEP, Nome do Pai, Nome da mãe, Número de telefone, IP de máquina, Geolocalização e demais documentos, especialmente os que possuem números.

A anonimização é um procedimento que consiste em remover ou substituir as informações pessoais ou sensíveis dos registros de dados. Conforme Art. 5º da lei 13.709 de 2018 inciso III: “Dado anonimizado é o dado relativo a titular que não possa ser identificado, considerando a utilização de meios técnicos razoáveis e disponíveis na ocasião de seu tratamento”.

Este processo é fundamental para evitar a identificação direta de uma pessoa, tornando a informação não mais pessoal. A anonimização ajuda a garantir que não seja possível reverter os dados transformados para revelar a identidade do titular das informações pessoais. Isso significa que os dados pessoais são separados de forma segura de suas origens durante esse procedimento.

O Datasus dispõe de 2(duas) técnicas de anonimização que são utilizadas sob demanda pela área negocial, são elas:

- **Supressão de dados:** Este procedimento consiste em omitir colunas de uma tabela que contêm informações sensíveis, extraiendo apenas os dados relevantes solicitados. Assim, os dados sensíveis não são expostos, garantindo maior proteção à privacidade.
- **Encobrimento/mascaramento de dados:** Trata-se de um procedimento que protege dados sensíveis ao substituí-los por caracteres ou valores fictícios. Esse processo é realizado automaticamente pelo banco de dados, assegurando que as informações confidenciais permaneçam ocultas enquanto os dados continuam disponíveis para análise ou uso em ambientes não críticos.

8.6 Acesso a Base de Dados

Banco de dados é um ambiente onde estão armazenadas as informações mais importantes do negócio. É um ambiente sensível, que possibilita o controle de todos os processos, portanto deve ser acessado exclusivamente pelos colaboradores em suas rotinas de trabalho. É muito importante que hajam mecanismos de proteção em relação ao ambiente de banco de dados e aos dados, principalmente em relação à acessos não autorizados. Além disso, é importante também garantir o que de acordo com Ramakrishnan e Gehrke (2008 cap. 21), são os principais objetivos relacionados à segurança em banco de dados: a confidencialidade, a integridade e a disponibilidade da informação.

Integridade

Garante que as informações em um sistema fiquem livres de acessos indevidos. Ou seja, está diretamente ligada a manter um dado protegido de ameaças internas e externas do negócio e toda a preservação do ciclo de vida dos dados.

Disponibilidade

Garante que os dados estejam disponíveis e acessíveis para usuários autorizados. Está relacionado a manter o acesso a eles de modo constante, evitando interrupções.

Confidencialidade

Garante que dados sejam mantidos em segredo ou privados e não devem ser acessadas por indivíduos não autorizados. O usuário terá acesso apenas à informação de que necessita para o cumprimento das suas funções.

9. PROCEDIMENTOS PARA ACESSO AOS BANCO DE DADOS

O acesso ao banco de dados é restrito e disponibilizado para agentes internos ou externos, para isso são criados usuários, onde o CPF do requerente é vinculado ao acesso.

A autorização para acesso aos bancos de dados em ambiente de produção do Ministério da Saúde ocorre, exclusivamente, por meio do preenchimento do **Termo de Responsabilidade (TR)**. O TR é um documento que consiste na solicitação de um usuário para acesso à base de dados de produção, a fim de obter privilégios de consulta e/ou edição desses dados. O objetivo do TR é garantir o cumprimento do princípio da Integridade, protegendo os dados de acesso indevidos e não autorizados pela área negocial. Além disso, é um dos instrumentos que formalizar e documenta quem está acessando determinada informação dentro do Ministério da Saúde.

Atualmente, os Termos de Responsabilidade são divididos em duas categorias no que diz respeito aos usuários, são elas:

- **Cadastro de usuário via CPF (TR CPF):**

Deve ser utilizado quando uma pessoa física, pertencente ou não ao quadro de colaboradores do Ministério da Saúde, solicita o acesso a um determinado esquema ou banco de dados em ambiente de produção. Este acesso é concedido único e exclusivamente por meio do número de CPF, que é vinculado a um IP de máquina. Em caso de alterações do número de IP, abaixo explicamos melhor de como proceder nestes casos.

- **Cadastro de usuário via Sistema (TR SISTEMA):**

Deve ser utilizado quando um esquema solicita acesso a um outro esquema em ambiente de produção. O acesso é liberado em nome de um determinado usuário de sistema, por exemplo: ETL_DC.

O preenchimento de um Termo de Responsabilidade (TR) para **acesso a banco de dados** deve ser realizado pelo link: <https://integrangs.saude.gov.br/internal/#/term>, nele é possível encontrar as orientações necessárias para o seu devido preenchimento. Em caso de TR CPF, é importante que seja preenchido corretamente e assinado via GOV.BR, disponível em <http://assina->

dor.itb.br/. Outra opção para assinar o Termo de Responsabilidade, é a assinatura padrão do Sistema Eletrônico de Informações – SEI.

Após o preenchimento do TR, a equipe de banco de dados (SEIDIGI/DATASUS/CGIE/COBD) analisa as informações descritas e, **somente após essa análise e a sua devida aprovação por parte da COBD**, o termo poderá ser despachado via SEI para assinatura do gestor(a) responsável pela informação.

O Gestor(a) da Informação ou o(a) seu(sua) substituto(a) é o(a) responsável por autorizar o Termo de Responsabilidade. Na ausência de um perfis citados, o coordenador(a) geral ou diretor(a) responsável pela área gestora do sistema podem autorizar o acesso.

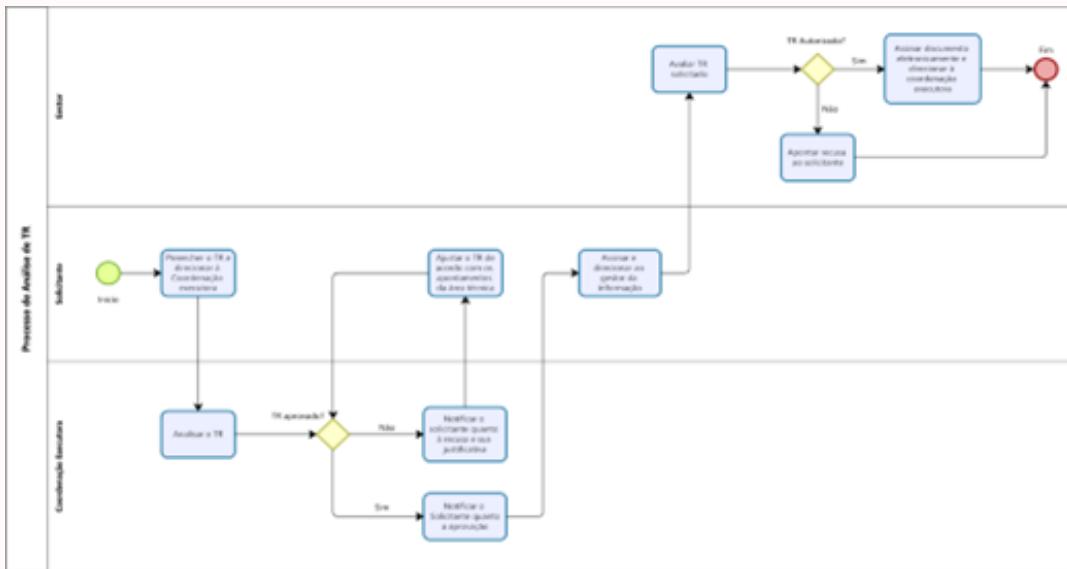
O perfil de Gestor(a) da Informação é designado pela área responsável pelo dado, que deve formalizar a atualização das designações deste perfil via SEI e encaminhar para o DATASUS, indicando quem são os(as) servidores gestores(as) daquela informação. Uma vez que o pedido é recebido pelo DATASUS, a Coordenação Geral de Relacionamento, Governança e Projeto (CGRGP) emite e publica uma portaria explicitando a designação da área gestora e com isso, são atualizados os sistemas: Coordenação de Estratégia de Dados - CED, Redmine e Planilha de Inventário de Sistemas (documento interno).

9.1 Regras de preenchimento

- Não é permitido acesso por usuário CPF para utilização de software(s) de tecnologias para a análise de dados, estatística, big data, ETL e de linguagem de programação em base de dados quentes transacionais. Essas atividades são realizadas apenas por meio de usuário de aplicação, solicitando através do TR SISTEMA.
- Na aba esquemas, é obrigatório o preenchimento do nome do esquema e seu respectivo banco ou string de conexão.
- Somente o Gestor da Informação ou o Gestor da Informação substituto nomeados em portaria podem atuarizar o TR.

Após as devidas autorizações do(a) gestor(a) do sistema, o processo deverá ser tramitado para a COBD, para que sejam tomadas as devidas providências quanto a concessão dos privilégios ao usuário conforme descrito no Termo de Responsabilidade (TR). O acesso será permitido ao usuário de acordo com o prazo autorizado pelo(a) gestor(a) da informação via despacho no processo SEI.

Em ausência à determinação do prazo pela área gestora da informação, por padrão(default) a permissão será concedida por no máximo 1 ano. Após este período, é necessário realizar a renovação do acesso, realizando todo o processo novamente, ou seja, novo termo preenchido, analisado, autorizado e aprovado pelo novo prazo.



! Todos os Termos de Responsabilidade devem constar o(a) gestor(a) da informação, de acordo com a informação cadastrada na plataforma Coordenação de Estratégia de Dados (CED). Compete exclusivamente à área gestora da informação manter atualizada as informações de gestores(as) responsável, e notificar a Coordenação-Geral de Relacionamento, Governança e Projetos (CGRGP) por meio de ofício via SEI. As designações anteriores são revogadas por meio de publicação de uma nova portaria.

9.2 Renovação da permissão e alteração de IP

- Para renovação de permissão de usuário, é necessário o preenchimento de um novo TR CPF, todo o processo de análise e autorização será realizado novamente.
- Caso o usuário tenha trocado de máquina e teve seu endereço IP alterado, é necessário que seja formalizado via SEI, preferencialmente em um novo processo, por meio de um despatcho solicitando a atualização do IP na concessão do seu acesso, para este caso não precisa preencher um novo TR.

9.3 Permissão emergencial

Para acesso emergencial é necessária autorização do(a) gestor(a) da informação, podendo exclusivamente neste caso ser enviado via e-mail, explicitando os privilégios necessários. O pedi-

do será analisado e caso seja aprovado, será concedido o acesso à informação por até 5 dias úteis.

Caso o gestor da informação não esteja disponível, o(a) coordenador(a)-geral, diretor(a) ou secretário(a) geral, poderá autorizar. O pedido de cessão de dados deve ser formalizado via processo no SEI, com a justificativa (motivo) do pedido de acesso aos dados, a finalidade (objetivo) da sua utilização, a indicação da pessoa que irá manusear o dado, assim como o anexo do e-mail corporativo enviado.

9.4 Permissão DDL

Somente a equipe DBA da coordenação de banco de dados do Ministério da Saúde tem permissão de DDL. Qualquer alteração na estrutura do banco de dados (DDL - (Data Definition Language) deve ser previamente submetida via chamado para à avaliação da equipe DBA para assegurar a conformidade com os padrões da MAD e boas práticas de estrutura de dados. Esse processo tem como objetivo mitigar riscos e impactos negativos no ambiente de banco de dados, além de reforçar o controle e a segurança dos bancos de dados administrados pela COBD. Exemplos de riscos associados:

1. ALTERAÇÃO ACIDENTAL OU MALICIOSA	
PROBLEMA	IMPACTO
Usuários com permissões de DDL podem, intencionalmente ou por engano, excluir ou alterar estruturas críticas do banco de dados (como tabelas ou índices).	Perda de dados, corrupção do banco ou interrupção de sistemas dependentes.
2. PERDA DE DADOS	
PROBLEMA	IMPACTO
Comandos como DROP TABLE ou ALTER TABLE podem resultar na exclusão ou modificação de dados.	Dificuldade ou impossibilidade de recuperação, mesmo com backups, se não forem frequentes.
3. INSTABILIDADE NO SISTEMA	
PROBLEMA	IMPACTO
Alterações não planejadas na estrutura do banco de dados podem levar a erros de compatibilidade em aplicações que dependem dessas estruturas.	Interrupção de serviços, falhas em aplicações e degradação de desempenho.
4. IMPACTO NO DESENHO	
PROBLEMA	IMPACTO
Alterações como a criação de índices ou execução de grandes operações de DDL podem impactar significativamente o desempenho do banco de dados, especialmente em produção.	Lentidão e aumento no tempo de resposta do sistema.

5. ABERTURA PARA ATAQUES EXTERNOS

PROBLEMA	IMPACTO
Usuários com más intenções podem explorar permissões de DDL para comprometer o sistema (ex.: criar tabelas falsas para desviar dados ou criar backdoors).	Violação de segurança e roubo de informações.

6. DESVIO DE CONTROLES DE AUDITORIA

PROBLEMA	IMPACTO
Permissões DDL podem ser usadas para modificar esquemas ou tabelas de forma que ocultem atividades inadequadas ou fraudulentas.	Dificuldade em rastrear atividades suspeitas e garantir conformidade regulatória.

9.5 Cessão de dados (extração)

A cessão de base de dados é o processo de disponibilização de uma cópia, seja ela total ou parcial, de uma base de dados. Os dados são disponibilizados em um arquivo externo, onde o usuário acessa e faz o download da informação.

A Lei Geral de Proteção de Dados Pessoas (13.709 de 2018), em seu artigo 13(treze) dispõe:

Art. 13. Na realização de estudos em saúde pública, os órgãos de pesquisa poderão ter acesso a bases de dados pessoais, que serão tratados exclusivamente dentro do órgão e estritamente para a finalidade de realização de estudos e pesquisas e mantidos em ambiente controlado e seguro, conforme práticas de segurança previstas em regulamento específico e que incluam, sempre que possível, a anonimização ou pseudonimização dos dados, bem como considerem os devidos padrões éticos relacionados a estudos e pesquisas.

Portanto, para este caso, o pedido de cessão de dados deve ser formalizado via processo no SEI, com a justificativa (motivo) do pedido de acesso aos dados, a finalidade (objetivo) da sua utilização, a indicação da pessoa que irá manusear o dado e o prazo requerido para obter o acesso à informação.

Neste caso, a área solicitante deve preencher este Termo de Responsabilidade, que é diferente do Termo disponibilizado no IntegraMS, conforme mencionado anteriormente nos casos de acesso à banco de dados.

Compete à Área Gestora:

1. Analisar o pedido e verificar a viabilidade dele;
2. Caso o pedido inclua dados sensíveis não totalmente anonimizados, enviar o Termo de Responsabilidade (Cessão de Dados) para ser preenchido e assinado pelo Solicitante;
3. Tramitar via processo no SEI o Termo de Responsabilidade para a COBD;

4. Determinar de maneira clara qual o tipo de acesso à informação está sendo autorizado (total ou parcial);
5. Elaborar e enviar para a COBD os scripts para a extração dos dados com as regras de negócio;
6. Um despacho solicitando e autorizando a cessão e informando o e-mail do solicitante para disponibilizar os dados;

Compete à COBD:

1. Verificar se o pedido foi devidamente autorizado pelo gestor da informação;
2. Executar o script e disponibilizar o arquivo com os dados solicitados à área solicitante;
3. Os dados serão compartilhados no e-mail corporativo da pessoa autorizada a manusear do dado, por meio da plataforma sharepoint.
4. Formalizar via processo no SEI que os dados foram cedidos conforme solicitado e autorizado, tanto à área gestora da informação quanto à área solicitante.

A Coordenação de Gestão de Banco de Dados atua com bastante cautela no que diz respeito à disponibilização de dados sensíveis. Os processos que foram apresentados neste documento foram criados com o objetivo de promover mais segurança e garantir a privacidade das informações, em respeito e em cumprimento às leis e princípios que regem o tratamento de dados pessoais no Brasil.

Por este motivo, os e-mails que solicitam, aprovam ou disponibilizam dados relacionados ao ambiente de produção devem ser criptografados, com confirmação de entrega e de leitura.



Preferencialmente, os e-mails enviados para a COBD com pedido de acesso à dados ou bancos de dados de ambientes de produção, desenvolvimento ou homologação devem ser criptografados, com confirmação de entrega e leitura.

10. HOSPEDAGEM E INTERNALIZAÇÃO DE SISTEMAS

10.1 Internalização

Internalização refere-se ao processo cujas soluções tecnológicas são mantidas e potencialmente evoluídas na infraestrutura e pelas equipes do DATASUS.

Para o processo de internalização de sistema, é obrigatório que todos os objetos estejam dentro do padrão de nomenclatura da MAD, bem como nome do esquema, nomes dos usuários de manutenção e aplicação.

10.2 Hospedagem

Os sistemas que foram desenvolvidos por equipes externas ao Datasus, sendo tão somente implantados em sua infraestrutura a pedido das áreas negociais, não precisa seguir os padrões de nomenclatura da MAD, com exceção do nome do esquema, nomes dos usuários de manutenção e aplicação.

Entre os exemplos que se enquadram nesse caso, destacam-se os softwares de prateleira, que utilizam um conjunto de termos padronizados (terminologias) e que não devem ser alterados, porque a consistência e a padronização desses termos são fundamentais para o correto uso e entendimento dos dados.

Outro exemplo é o SNOMED (Systematized Nomenclature of Medicine) - uma terminologia médica padronizada, que organiza de forma sistemática uma coleção de termos médicos, códigos, sinônimos e definições. Esses termos são usados para documentar informações clínicas, como diagnósticos, tratamentos e procedimentos, e são essenciais para garantir que as informações sejam compreendidas de maneira uniforme por profissionais de saúde em diferentes contextos.

maneira como os termos são nomeados e classificados tem impacto direto na forma como eles

são usados em sistemas de saúde, como prontuários eletrônicos, diagnósticos e tratamentos. Alterações nos nomes ou nas definições podem afetar a precisão e a utilidade dos dados e por esse motivo, são preservados no processo de hospedagem.

OBS: Futuramente, caso haja o interesse pela internalização, é necessário que o sistema seja adequado aos padrões desta metodologia (MAD).

Acesse a página abaixo para maiores informações: <https://datasus.saude.gov.br/internalacoes/>

II. CONCLUSÃO

A definição dos processos e padrões descritos neste documento tem como objetivo promover a melhoria contínua dos serviços e do suporte aos sistemas do Ministério da Saúde. Além disso, busca garantir a segurança de todos os envolvidos, tanto da equipe executora quanto da equipe solicitante. Com a implementação da padronização, será possível promover uma evolução constante nas estruturas de dados, de forma simples, eficiente e que assegure a qualidade dos bancos de dados do Datasus.

Ao adotar uma metodologia de administração de dados bem estruturada e adequadamente implementada, será possível aprimorar a qualidade dos dados, aumentar a eficiência operacional e facilitar a tomada de decisões. Além disso, essa metodologia contribui para o crescimento sustentável dos dados e dos projetos vinculados ao Ministério da Saúde.

A organização e padronização das estruturas de dados minimizam as chances de erros e inconsistências, resultando em menos retrabalho e menores custos associados à correção de problemas. Assim, a Metodologia de Administração de Dados também contribui para a eficiência no uso de recursos públicos, por meio da redução de custos.

Por fim, é fundamental que todas as orientações aqui descritas sejam de fato seguidas. Reforçamos que esta Coordenação de Gestão de Banco de Dados – COBD está à disposição para apoiar e esclarecer quaisquer dúvidas durante o processo de entendimento e implementação das informações contidas neste documento.

ANEXO

1

**ORIENTAÇÕES PARA
CONSTRUÇÃO DE
MODELO DE DADOS
COM QUALIDADE**

APRESENTAÇÃO

A Administração de Dados em uma instituição é uma área que deve ter uma ampla atuação e andar em conjunto com a área de Desenvolvimento de Sistemas.

A Administração de Dados tem como objetivo principal fazer a gestão do modelo de dados corporativo da instituição, promovendo sua conceituação, segurança, integridade, compartilhamento e qualidade.

Para isso a Administração de Dados deve atuar de forma a:

- Conhecer o negócio da empresa de forma, a saber, as áreas que utilizam os dados, considerando o nível de acesso que cada um terá do dado;
- Modelar os dados corporativamente;
- Respeitar as ,últiplas visões e utilizações de um mesmo dado;
- Disponibilizar os dados de forma organizada;
- Compartilhar os dados evitando redundância;
- Servir como base para a construção de sistemas de informações flexíveis, integrados e eficientes.

O objetivo deste documento é apresentar diretrizes para a construção de um modelo de dados com qualidade, e que devem obrigatoriamente ser seguidas, visando assim a melhoria da qualidade da informação do Ministério da Saúde.

HISTÓRICO DE REVISÃO

DATA	AUTOR	DESCRIÇÃO	VERSÃO
15/01/2021	Eliete Colucci Sousa	Criada a 1 versão da Metodologia de Administração de Dados (MAD).	1.0
27/01/2015	Eliete Colucci Sousa	Revisão de itens de acordo com observações sobre o uso da MAD.	2.1
06/01/2016	Eliete Colucci Sousa	Revisão de itens de acordo com observações sobre o uso da MAD.	2.2
20/07/2016	Eliete Colucci Sousa	Alterações nos seguitimes itens da Cartilha do formato da MAD de forma a deixar de acordo com o padrão de publicação do Portal do DATASUS.	2.2.1
21/11/2016	Eliete Colucci Sousa	Alterações na formatação do texto diagramado, de forma a facilitar a sua revisão. Além disso, os seguintes itens foram incluídos/revisados. 1) Inclusão de classes de dados; 2) Alteração de procedimento de criação de AUDITORIA; 3) Alteração de procedimento de criação de concessão de permissões de objetos em banco de dados; 4) Alteração de procedimento de criação de texto do uso de palavras em outro idioma; 5) Inclusão de regra para datatype padrão para campos do tipo caracter com tamanho máximo delimitado e numérico com precisão; 6) Incluir resposta padrão para campos BLOB.	2.2.2
31/07/2018	Eliete Colucci Sousa	Pequenas alterações para tornar mais claro o texto de alguns itens. Inclusão do item "Privilégios Padrão", que define quais são os privilégios concedidos, por padrão, na criação de objetos.	2.3
20/01/2023	Gislaine Mabel da N. Trindade	Revisão técnica do documento e atualização do documento para contemplar o cenário atual do DATASUS.	2.4

1. INTRODUÇÃO

A Administração de Dados possui vários processos, mas o foco deste documento é a modelagem de dados.

Sendo assim, nele são tratados os padrões e introduzidas as diretrizes para uma modelagem de dados de qualidade, servindo de orientação para os técnicos e os usuários.

Como a qualidade de um modelo é imprescritível para que o Ministério da Saúde possa prestar seus serviços ao cidadão, estas diretrizes devem obrigatoriamente ser seguidas.



2. DEFINIÇÃO DE PADRÕES

2.1 Padrões de nomes de objetos de banco de dados

A nomenclatura para objetos de banco de dados esta definida na norma constante no link:
<https://ced.saude.gov.br/index.php?r=site%2Fmad> - Nomenclatura.

o Padrão de nomes deve seguir a norma Padrão ISO/IEC 1179-5, que é um padrão internacional, assim como o Catálogo de Padrões do Governo (e-ping - Padrões de Interoperabilidade do Governo Brasileiro).



FIQUE LIGADO

É de fundamental importância, que o modelo de dados corporativo da instituição esteja de acordo com o padrão de nomenclatura definido para objetivos de banco de dados, pois isso facilita o entendimento de todos que manipulam e necessitam de modelo de dados para o desenvolvimento de seus trabalhos. Isso é um dos itens de garantia de qualidade do modelo.

3. DIRETRIZES

O assunto qualidade de modelo de dados está diretamente ligado com garantia da qualidade da informação, pois um bom modelo de dados é peça fundamental para a garantia de dados confiáveis e seguros.

Um modelo de dados representa a definição, caracterização e relacionamento dos dados em um determinado ambiente, e por isso, para que ele possua qualidade de produzir dados corretos, precisos, consistentes, completos, integrados, aderentes às regras de negócio e aos domínios estabelecidos.

Além disso, um modelo de dados deve sempre que possível levar em consideração a possibilidade de futuras implementações, considerando aspectos de aquisição, arquivamento e padronização dos dados, bem como no armazenamento, recuperação e reutilização das informações.

Este item tem por objetivo descrever procedimentos que devem ser adotados na elaboração/manutenção de modelos de dados, visando nortear a Administração de Dados na construção de modelos de dados com qualidade.

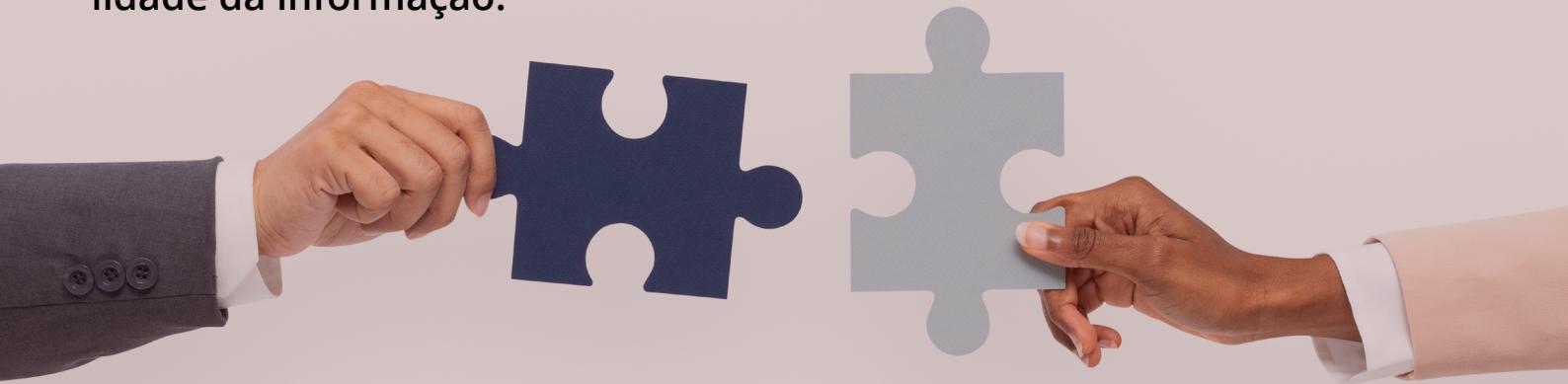


FIQUE LIGADO

Devemos analisar o modelo de dados como se estivéssemos com uma lupa, para observarmos com cuidado o que está contido no modelo de dados.

Portanto, na elaboração e manutenção de um modelo de dados temos que trabalhar de forma a garantirmos a qualidade do modelo. Para isso, as regras técnicas, normas para modelagem de dados e documentação descritas neste documento devem ser seguidas, pois são boas práticas de modelagem de dados. Veja o Anexo I os indicadores de qualidade de um modelo de dados.

Devemos todos trabalharmos juntos para a melhoria da qualidade da informação.



Assim estaremos construindo para a melhoria dos serviços prestados pelo Ministério da Saúde ao cidadão.

3.2 Regras Técnicas

Com relação às regras técnicas, são descritas a seguir orientações quanto ao bom uso de objetos de banco de dados.



PRESTE ATENÇÃO!
As diretrizes a seguir dão boas dicas.

3.2.1 Primary Key

Uma primary key permite a identificação única do registro, podendo ser um sequencial (controlada por uma sequência do banco de dados), que não tem significado negocial, ou uma chave negocial, simples ou composta. No caso da PK ser um sequencial, é interessante que se defina uma UK como chave negocial da tabela, por exemplo, em uma tabela de Pessoas Físicas a chave negocial seria o nº do CPF.

Conforme o documento de padrão de nomenclatura (item 10.2 – Tipos de Tabelas) está definido quais tipos de tabelas são classificados como negociais e quais não são. Toda tabela

definida como negocial é obrigatória a definição de uma PK, para as que não são classificadas como negocial, a PK é opcional.

3.2.2 Foreign Key (FK)

Uma ou mais colunas que são geradas por um relacionamento com a tabela detentora destas colunas e mais outra tabela, que vai determinar os valores possíveis para estas colunas. Em uma FK podemos ter uma coluna ou mais de uma (chave composta). No caso de chave composta é interessante que as colunas tenham preenchimento obrigatório, pois o SGBD somente realiza a verificação de integridade quando todas as colunas estiverem preenchidas.

3.2.3 Constraint para Coluna NOT NULL

Para essas constraints não há necessidade de se definir um nome, pois estas são criadas automaticamente pelo banco de dados.

Conforme Anexo I, o não atendimento deste item contrariará o indicador de qualidade documentação.

3.2.4 Constraint Default de Coluna

Para colunas com preenchimento obrigatório, o uso do valor default é obrigatório somente nos casos de criação de coluna com preenchimento obrigatório em tabela já existente.

Para essas constraints não há necessidade de se definir um nome, pois estas são criadas automaticamente pelo banco de dados.

Conforme os [Indicadores de qualidade de modelo de dados](#), o não atendimento deste item contrariará os indicadores de qualidade acessibilidade e documentação.

3.2.5 Lista de Valores

Esta situação se refere a colunas que possuem um domínio definido, por exemplo, estado civil, tipo sanguíneo, tipo de pessoa, situação de contrato, etc.

É importante ficar claro que nestas situações é imperativo que:

- Haja a implementação de controles com os valores válidos constantes na lista, que pode ser por check constraint ou tabela de domínio, conforme regras definidas neste item;
- Todos os valores devem ter o significado de cada um definido e documentado, pois isso diminui os problemas com informações sem credibilidade por falta de entendimento dos dados armazenados na base de dados.

Então para a implementação de lista de valores devem ser seguidas obrigatoriamente as 5 regras descritas a seguir.

I. Devem ter preenchimento obrigatório.

O preenchimento obrigatório é importante, pois NULL não é um valor pré-definido e sim ausência de valor.

Acrescentando-se que registros com NULL prejudicam a performance de consulta, pois estes não são considerados em index.

Além disso, na obtenção de indicadores que envolvam essas colunas de domínio com valores NULL, esses registros também não são considerados.

Nos casos onde é necessário NULL, deve-se definir como valor do domínio, a coluna preenchida totalmente com o dígito 9, seja para datatype numérico ou de caractere. Por exemplo:

- Coluna numérica de tamanho 1, o valor correspondente deve ser **9**;
- Coluna numérica de tamanho 2, o valor correspondente deve ser **99**;
- Coluna numérica de tamanho 3, o valor correspondente deve ser **999**;
- Coluna caractere de tamanho 1, o valor correspondente deve ser **'9'**;
- Coluna caractere de tamanho 2, o valor correspondente deve ser **'99'**;
- Coluna caractere de tamanho 3, o valor correspondente deve ser **'999'**.

A descrição do valor deve ser definida da forma mais adequada ao negócio que está atendendo.

II. Utilizar CK com a definição dos valores do domínio:

- Domínio possui no máximo dois valores e mais o valor para NULL conforme descrito acima;
- Os valores contidos na lista devem indicar que são estáveis, isto é, que a lista não vai aumentar;
- Para o caso da CK depender do conteúdo de uma outra coluna da tabela, deve ser criada uma CK da tabela;
- No comentário do campo, onde é descrita a sua finalidade, é obrigatório que contenha cada valor e a descrição do significado de cada um.

Na situação aqui indicada é obrigatório o uso de uma CK.

III. Domínio Sim ou Não

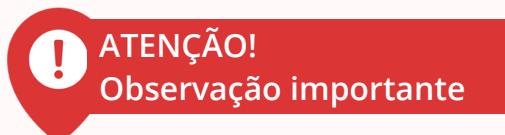
Para colunas cujo conteúdo deve expressar Sim ou Não, os valores do domínio devem ser S ou N, respectivamente.

IV. Utilizar uma tabela de domínio, quando o domínio não se encaixa nas situações relacionadas com indicação de uso de CK.

V. Como incluir dados em tabela de domínio?

- Para tabelas do DBGERAL, a responsabilidade é da coordenação de banco de dados;
- Para tabelas de negócio temos as seguintes situações:
 - Manutenção de dados em tabela que não tem impacto no negócio e pode ter muitas alterações: o indicado é que o sistema tenha telas para manutenção das tabelas;
 - Manutenção de dados em tabela que não tem impacto no negócio e cujo previsto é que praticamente não haja alterações em seu conteúdo: o indicado é que o sistema não tenha telas para manutenção das tabelas (atualização por script);
 - Manutenção de dados em tabela que tem impacto no negócio: o indicado é que o sistema não tenha telas para manutenção das tabelas, até porque em caso de alterações no conteúdo, a equipe responsável precisa analisar o impacto no sistema.

Conforme os [Indicadores de qualidade de modelo de dados](#), o não atendimento deste item contrariará os indicadores de qualidade acessibilidade, credibilidade e flexibilidade.



Regras para colunas com DOMÍNIO DEFINIDO:

- Preenchimento Obrigatório
- Para o controle dos valores do domínio, devemos utilizar CK ou tabela?
 - Check Constraint (CK) se possuir no máximo 2 valores e mais um para quando o NULL representar o NULL, quando necessário (normalmente colunas com prefixo ST ou TP);
 - Quando possuir mais valores, deve-se ter uma tabela para armazenar os valores de domínio (normalmente coluna com prefixo CO recebendo FK da tabela de domínio).
 - Se uma coluna for criada como CK (3 valores no máximo), e devido essa lista não ser estável e os valores do domínio começarem a crescer, o controle deverá ser alterado para tabela com tais valores, exportando FK → por isso, somente utilize CK quando tiver certeza que a lista de valores é estável.

3.2.6 Date x Timestamp

A utilização desses datatypes às vezes causa um pouco de confusão. Quando devo usar cada um?

O que é melhor

DATE ou TIMESTAMP

Utilize o **DATE** quando é necessário apenas a DATA ou DATA e HORA com precisão de segundos.

Utilize o **TIMESTAMP** quando é necessário DATA e HORA com precisão de milissegundos ou para utilização de timezone. O timezone é apropriado quando é necessário exibir informações de data e horários usando o fuso horário do sistema cliente.

3.2.7 Tabelas de Auditoria

Tabelas que permitem efetuar um rastreamento de qualquer operação que ocorre nos dados de uma tabela. Para que uma tabela de auditoria seja eficaz, deve responder quem fez a operação, quando foi feita e o que foi feito.

Para a auditoria funcionar, devem ser inseridos na aplicação comandos para se obter o usuário final e o IP da máquina onde o usuário está logado:

```
-- Login/Identificador do usuário como parâmetro  
EXEC DBGERAL.PKG_INFO_LOG.PC_INFORMA_USUARIO('USERNAME');  
  
-- IP do usuário como parâmetro  
EXEC DBGERAL.PKG_INFO_LOG.PC_INFORMA_IP('10.0.0.1');
```



FIQUE LIGADO

A execução desses procedimentos deve ser incluída na aplicação é fundamental, para que possamos ter a identificação dos dados do usuário (pessoa física) que está executando a operação. Assim teremos uma auditoria EFICAZ!

Então sempre que criar a auditoria, o solicitante deve ser comunicado disso.

Quanto aos privilégios de acesso de tabelas de AUDITORIA:

- A alimentação de tabela de auditoria pode ser feita por trigger ou pela aplicação. É sugerido a utilização de trigger, pois julga ser um método mais efetivo.
- Para garantir a confiabilidade e a rastreabilidade das informações, tabelas de auditoria não podem ter seu dado alterado. Por isso os privilégios de acesso a essas tabelas seguem as seguintes regras:
 - Alimentada por trigger: grant de SELECT;
 - Alimentada pelo aplicativo: grant de SELECT e INSERT.

Preste muita atenção!

```
    non - text - :200px;">persisted progress bar  
<html> <errorMessage = ko , observable> <br>  
<p style="color:orange;">HTML font code is here  
function todoitem(data) : <html> <errorGroup>  
    var self = this <html> <errorGroup>  
    data = data || <html> <errorGroup>
```

Investigue antes. Nunca deixe mais de uma trigger de auditoria habilitada.

Mas então como proceder para sistemas legados em que a equipe decide por utilizar a nova estrutura:

- Se não há auditoria, auditoria será criada no padrão da metodologia MAD.
 - Se há auditoria em outro formato, é sugerido a utilização do padrão da equipe, mas a decisão é da equipe responsável pelo aplicativo. Nunca se esquecer, que na criação de uma nova auditoria para uma tabela, a trigger da antiga deve ser desabilitada.
 - Para projetos novos, deve ser adotado o padrão da MAD.

A estrutura da tabela de auditoria no padrão da MAD possui as mesmas colunas da tabela origem e mais as de controle, conforme abaixo:

COLUNA DE CONTROLES DA AUDITORIA		
COLUNA	TIPO/TAMANHO	Descrição
AU_DT_OPERACAO	TIMESTAMP(6)	Data da operação, armazenada no formato TIMESTAMP "DD/MM/YYYY HH24:MI:SS.FF".
AU_TP_OPERACAO	VARCHAR2(1)	Indica a operação efetuada pelo usuário na origem. Podendo ser: (I: para inclusão de registro (comando insert)); (A: para alteração de registro (comando update)); (E: para exclusão de registro (comando delete)).
AU_DS_USUARIO_SESSAO	VARCHAR2(30)	Usuário de banco da sessão ORACLE.
AU_NU_IP_SESSAO	VARCHAR2(15)	Endereço IP da máquina que originou a sessão ORACLE.
AU_DS_USUARIO_REDE	VARCHAR2(40)	Usuário de rede logado na máquina.
AU_DS_HOST_REDE	VARCHAR2(40)	Nome da máquina na rede.
AU_DS_USUARIO_APPLICACAO	VARCHAR2(100)	Usuário final, informado pela aplicação. Informado e recuperado através do package DBGERAL.PKG_INFO_LOG.

COLUNA DE CONTROLES DA AUDITORIA		
COLUNA	TIPO/TAMANHO	DESCRIÇÃO
AU_NU_IP_APPLICACAO	VARCHAR2(15)	Endereço IP do usuário final, informado pela aplicação. Informado e recuperado através da package DBGERAL.PKG_INFO_LOG.
AU_CO_TRANSACAO	VARCHAR2(20)	Código identificador da transação no banco de dados.
AU_DS_USUARIO_COMPLEMENTO	VARCHAR2(500)	Informação complementar do usuário final, informada pela aplicação. Informado e recuperado através da package DBGERAL.PKG_INFO_LOG.

Quando gerar AUDITORIA em uma tabela



A decisão de gerar ou não AUDITORIA para uma tabela é do Gestor de Negócio e/ou Coordenador da Equipe de Sustentação responsável pelo aplicativo.

É importante ressaltar que uma **AUDITORIA cresce de forma bem mais rápida que a tabela original**, pois a cada operação de INSERT / UPDATE / DELETE é gerado um novo registro na AUDITORIA.

Sendo assim, ao se analisar a necessidade da AUDITORIA, os seguintes itens deverão ser considerados:

- Os dados da tabela são críticos para o negócio.
- Qual é o crescimento da tabela? Para isso, deve-se verificar o nº inicial de linhas, o % de crescimento e a periodicidade para essa %.
- Qual é a justificativa para a criação da AUDITORIA.

O questionamento sobre a necessidade de AUDITORIA ou não, deve ser feito no momento de sua criação no DESENVOLVIMENTO, seja a pedido do Gestor ou por concessão de privilégios.



ATENÇÃO! TODOS OS AD'S

A geração e atualização de auditoria é de uso exclusivo do DBA.

Mas como os DBA's não têm proximidade com a área de desenvolvimento, o AD deve enviar as linhas de comando para o DBA possa executar. Preste atenção nas informações abaixo.

Para gerar e atualizar a estrutura de tabelas de auditoria devem ser observados os itens descritos a seguir.

1. Gerar Auditoria

O AD deve enviar a linha de comando a seguir para que o DBA a execute e gere a auditoria para os casos em que há necessidade de que uma tabela tenha auditoria.

```
EXEC DBDBA.PKG_AUDITORIA.PC_GERA_AUDITORIA('OWNER','TABLE_NAME');
```

Ao responder o atendimento da demanda, o solicitante deve ser comunicado que:

2. Atualizar Auditoria

Sempre que a estrutura de uma tabela for alterada, o AD deve verificar se essa tabela possui auditoria no padrão MAD e enviar a linha de comando a seguir para que o DBA a execute e atualize a auditoria.

Mas atenção nos casos de RENAME de coluna, se a package for executada diretamente, será incluída uma nova coluna com o novo nome. Então, nessas situações, o correto é executar o comando para renomear a coluna na tabela de auditoria: é bom que o AD passe esse comando para o DBA.

```
EXEC DBDBA.PKG_AUDITORIA.PC_ATUALIZA_AUDITORIA('OWNER','TABLE_NAME');
```

3. Gerar Script com Comandos de Atualização de Auditoria de Todas as Tabelas de um Esquema

```
DECLARE CURSOR TBS IS
SELECT OWNER, TABLE_NAME
FROM DBA_TABLES
WHERE OWNER = 'DBGERAL'
AND (TABLE_NAME LIKE 'TB\_%' ESCAPE '\' OR TABLE_NAME LIKE 'RL\_%' ESCAPE '\' OR TABLE_NAME LIKE 'RT\_%' ESCAPE '\');
BEGIN
FOR R IN TBS
LOOP
```

```
DBGERAL.PKG_INFO_LOG.FC_ATUALIZA_ESTRUTURA(R.OWNER, R.TABLE_NAME);
END LOOP;
END;
```

4. Verificar de Auditoria Desatualizadas em um Esquema

```
SELECT TB.OWNER, TB.TABLE_NAME
FROM DBA_TABLES AU, DBA_TABLES TB
WHERE AU.OWNER = TB.OWNER AND AU.OWNER = 'OWNER'
AND SUBSTR(AU.TABLE_NAME,3,30) = SUBSTR(TB.TABLE_NAME,3,30) AND AU.
TABLE_NAME LIKE 'AU\_%' ESCAPE '\'
AND (TB.TABLE_NAME LIKE 'TB\_%' ESCAPE '\' OR TB.TABLE_NAME LIKE 'RL\_%'
ESCAPE '\' OR TB.TABLE_NAME LIKE 'RT\_%' ESCAPE '\' );
AND DBGERAL.PKG_INFO_LOG.FC_VERIFICA_ESTRUTURA(TB.OWNER,TB.TAB-
LE_NAME) > 0
```

5. Observações Importantes

- Tabela de auditoria somente deve possuir privilégio de acesso de SELECT. Para os casos em que a tabela é atualizada pelo aplicativo também deve ter o INSERT.
- Campos BLOB não são auditados, para o caso de tabelas que ainda possuam campos desse tipo.
- Na exclusão de colunas da tabela origem, estas não são excluídas da tabela de auditoria, pois esta visa manter o controle das informações existentes e as que já existiram na tabela de origem.
- Tabela de auditoria não deve possuir constraint, pois as restrições de integridade devem ser garantidas pela tabela origem.
- Quando uma tabela de auditoria nos padrões da MAD é criada, é importante comunicar a necessidade dos comandos que devem ser inseridos na aplicação.

O não atendimento deste item contrariará os *Indicadores de Qualidade de Modelo de Dados, em especial o aspecto de credibilidade*.

3.2.8 Tabelas de Histórico

Tabela utilizada para armazenar os dados históricos para atender uma determinada funcionalidade necessária para o negócio.

Diferentemente de tabelas de auditoria, não há necessidade de que as tabelas com dados históricos possuam todas as colunas da tabela de origem.

A definição de quais colunas devem fazer parte da tabela de histórico deve constar nos requisitos no aplicativo, assim como a forma de manutenção de seus dados, que pode ser por trigger ou pelo próprio aplicativo.

O não atendimento deste item contrariará os Indicadores de Qualidade de Modelo de Dados, em especial o aspecto de credibilidade.

3.2.9 Tabelas Associativas

Uma tabela associativa representa uma entidade que não existe por si só e sua existência está condicionada à existência de duas ou mais entidades com relacionamento do tipo N:N. Além disso, o identificador negociável da tabela é formado exclusivamente pelas colunas que são geradas pela FK dessas tabelas relacionadas.

Vale ressaltar que mesmo que estas tabelas possuam atributos próprios devem ser tratadas como associativas (tabelas com prefixo RL_ ou RT_).

O não atendimento deste item contrariará os Indicadores de Qualidade de Modelo de Dados, em especial o aspecto de credibilidade.

3.2.10 Exclusão Lógica

Para o caso de tabelas com exclusão lógica, é importante verificar qual será o mecanismo de controle quanto ao uso de registros que estejam marcados como “Excluídos”, isto porque estes registros não podem ser utilizados. Sendo assim, ou a aplicação deve tratar isso ou então devem ser implementadas triggers nas tabelas que recebem FK dessas tabelas com exclusão lógica, de forma que os registros marcados como excluídos não sejam mais utilizados.

Mas o que é importante de observar é que a exclusão lógica de uma tabela deve ser tratada por uma coluna, onde o nome obrigatoriamente é **ST_REGISTRO_ATIVO** com datatype **VARCHAR2(1)** e preenchimento obrigatório. Além disso, deve possuir uma **check constraint** com os valores ‘S’ ou ‘N’ e a descrição pode ser “Indica se o registro está ativo ou não (excluído logicamente). O seu domínio é: S – Sim (está ativo) ou N – Não (não está ativo). O controle no uso de registros excluídos deve ser feito pela aplicação.”

Uma observação importante é que quando a tabela é criada já com essa coluna, não é obrigatório a definição de um valor **DEFAULT**, mas quando a coluna é adicionada em uma tabela já existente, deve-se definir o valor **DEFAULT**, pois assim a coluna pode ser criada como **NOT NULL** e o valor **DEFAULT** será preenchido para todos os registros existentes na tabela.

O não atendimento deste item contrariará os Indicadores de Qualidade de Modelo de Dados, em especial o aspecto de credibilidade.

3.2.11 Index

A criação de **index** deve ser feita visando uma melhoria de performance em consultas ao banco de dados, sendo que isso é possível na criação de index adequados. Outra forma de termos performance é por entender o comportamento dos objetos de forma a produzirmos os comandos SQL mais eficientes que implicarão em respostas mais rápidas para os usuários dos sistemas e uma melhoria no funcionamento dos bancos de dados, mas este será um assunto para ser tratado posteriormente.

A criação de index deve ser bem planejada, pois além das consultas, nas inclusões, atualizações e deleções de registros, eles também são atualizados e com isso há uma perda de performance nessas operações.

Então a criação de index deve ser feita de forma que beneficie mais a consulta e traga bom ganho de performance, que valha a pena o tempo perdido nas atualizações.

Sendo assim a criação de index é importante quando:

- Uma coluna é referenciada em um comando SQL de consulta na cláusula de restrição (filtro), através de:
 - Uma igualdade: sg_uf = 'DF';
 - Comparação com intervalo limitado: nu_vaga > 100;
 - Comparação com intervalo limitado: nu_vaga between 100 AND 200;
 - Para ORDER BY ou GROUP BY é bom que tenhamos um index considerando todas as colunas envolvidas e na ordem em que são especificadas nessas cláusulas (é bom que estas colunas tenham valores preenchidos, pois colunas com NULL não são consideradas em index). No caso do ORDER BY é melhor que as colunas sejam NOT NULL;
 - Utilizando-se MAX ou MIN, desde que a função faça parte da seleção e esteja sozinha, pois em caso contrário o index não é utilizado;
 - Para colunas que são FK, desde que estas não tenham pouca variação de valores, por exemplo, colunas de domínio;
 - Na utilização de OR na cláusula de junção/restrição, deve-se ter index em todas as colunas envolvidas no OR;
 - Quando se fizer uso de função na cláusula de junção/restrição, definir um index baseado em função;
 - O uso de consultas envolvendo na cláusula de restrições campos textuais devem ser evitados. Caso seja inevitável e o SGBD for o Oracle, utilizar index Intermedia Text (para

campos VARCHAR2, CHAR e CLOB).

O não atendimento deste item contrariará os *Indicadores de Qualidade de Modelo de Dados*, em especial o aspecto de credibilidade.

3.2.12 Triggers

O uso de triggers para implementação de regras de negócio é aconselhável evitar, sendo interessante a utilização para auditoria, histórico e garantia de integridade quando não é possível a implementação por constraints.

Nos casos de uso de Trigger em implementação de regra de negócio do aplicativo, a responsabilidade é da equipe de sustentação.

3.2.13 Views

O uso de view tem como objetivo facilitar a visualização dos dados, sendo que com elas é possível simplificar comandos com colunas resultantes de agrupamentos de dados de mais de uma tabela, cálculos, quando há necessidade de inibir colunas ou linhas de tabelas, por questões de segurança, etc.

Sendo assim, o uso de views é interessante quando se tem uma consulta de uso frequente, consultas complexas, para restringir o acesso à visualização de dados.

O não atendimento deste item contrariará os *Indicadores de Qualidade de Modelo de Dados*, em especial o aspecto de credibilidade.

3.2.14 View Materializada

Tipo especial de view com característica com armazenamento de dados próprio, originados de uma ou mais tabelas. Bastante utilizadas para facilitar consultas sem impactar na performance das tabelas transacionais. Também permite tratamento de regras negociais e criação de index, diferentemente das views convencionais.

O não atendimento deste item contrariará os *Indicadores de Qualidade de Modelo de Dados*, em especial o aspecto de credibilidade.

3.2.15 Campo BLOB

Quanto ao uso de campos com tipo BLOB, o armazenamento não deve ocorrer em base de dados, mas sim utilizando a tecnologia NFS (Network File System).

Havendo a necessidade de armazenar o nome do arquivo e a descrição do local de

armazenamento na base de dados, devem ser utilizados campos com prefixo **NO** (nome) e **DS** (descrição), respectivamente, do tipo texto.

Para o armazenamento em filesystem, a equipe responsável pelo aplicativo deverá procurar a equipe de Infraestrutura para obter as devidas orientações.

Para maiores informações, consultar o **memorando nº 32/2016/CGAM/DATASUS/SE/MS, SIPAR 2500.083574/2016-10** e **Nota Técnica nº 063/2016/Infraestrutura DATASUS**.

*O não atendimento deste item contrariará os **Indicadores de Qualidade de Modelo de Dados**, em especial o aspecto de credibilidade.*

3.2 Regras Técnicas

Para a concessão de privilégios, temos as seguintes observações:

- Na criação de tabelas os privilégios que são concedidos, por padrão, são **SELECT / INSERT / UPDATE**. Havendo necessidade de **DELETE**, deve ser informado na criação do objeto ou sob demanda após sua criação, informando a necessidade ou não de criação de auditoria no padrão DATASUS (tabelas com prefixo **AU_**).
- As exceções para concessão de privilégios para tabelas são descritas a seguir:
 - **Tabela de Auditoria Padrão DATASUS (AU_)**: somente privilégio de **SELECT**.
 - **Tabela de LOG**, cuja manutenção é de responsabilidade da equipe responsável pelo aplicativo (**TL_**): somente privilégio de **SELECT / INSERT**.
 - **Tabela de Histórico (TH_)**: somente privilégio de **SELECT / INSERT**. Havendo necessidade de qualquer outro privilégio este deve ser solicitado.
- Para o caso de **DROP** e recriação de objetos, os privilégios concedidos aos que estão especificados como padrão neste item. Caso o objeto tenha privilégios que não sigam o padrão e que devam ser mantidos, estes devem ser especificados na demanda.
- Os privilégios são concedidos para as **roles padrão do esquema** e estas para o usuário de manutenção (menos ambiente de Produção) e aplicação (todos os ambientes).
- Para o caso de alteração em **View Materializada**, esta é recriada, portanto deve seguir as orientações para o caso de **DROP** e recriação.

3.4 Normas para Modelagem Relacional

Resumidamente, as normas para modelagem de dados devem estar de acordo com o definido nos subitens de 3.3.1 à 3.3.4.

Este item apresenta boas práticas a serem observadas no processo de modelagem de dados.



PRESTE ATENÇÃO!

Se as normas técnicas são boas para serem seguidas, as negociais completam para termos modelos de qualidade.

3.4.1 Tabelas

a) Sempre que for identificada a existência de tabelas distintas com muitas propriedades, atributos e relacionamentos em comum, recomenda-se avaliar a possibilidade de aplicar o conceito de generalização/especialização.

b) Em princípio, as tabelas deverão ser modeladas respeitando-se as regras de normalização. Desnormalizações poderão ser feitas desde que haja justificativa técnica da sua necessidade, como por exemplo, para facilitar o acesso às informações e/ou para melhoria de performance. No entanto, para qualquer desnormalização deverá haver um procedimento implementado no banco de dados que garanta a integridade da informação. As justificativas para tais procedimentos deverão ser documentadas na ferramenta case.

c) Em princípio todas as tabelas deverão possuir **PK**. Para os casos de tabelas em que a **Primary Key** é uma coluna sequencial, não temos a garantia de unicidade de registros de forma negocial e para essas situações deverá ser verificada a existência de pelo menos uma **Unique Key (UK)** composta ou simples. Nessa verificação não são levadas em consideração tabelas de apoio, log e temporárias.

d) O relacionamento entre as tabelas deverá ser feito através da criação de chave estrangeira (**FK**). Excepcionalmente, quando isso não for possível, deverá ficar documentado na tabela o motivo e a forma que será utilizada para garantir a integridade dos dados, devendo ser observado o item que trata de **Primary Key** no documento.

e) As tabelas mais volumosas deverão ser estudadas de forma a se decidir a criação de índices, bem como outras questões físicas como, por exemplo, particionamento.

f) Tabelas com muitas colunas sem preenchimento obrigatório deve-se avaliar a real necessidade de tais colunas ou até da tabela. Havendo a necessidade de se manter essa ocorrência, deve haver uma justificativa técnica e documentá-la.

3.4.2 Colunas

- a) Toda coluna cujo conteúdo é uma lista de valores deve ter seu domínio definido.
 - b) Colunas cujo conteúdo é uma lista de valores implementada por FK com tabela de domínio ou CK devem possuir preenchimento obrigatório, pois as colunas desse tipo não devem admitir ocorrência nula, visto que o nulo não é um valor pré-definido e sim ausência de valor.
 - c) Colunas cujo conteúdo representam a mesma informação em modelos diferentes devem possuir o mesmo tipo/tamanho e lista de valores quando for o caso.
-

3.4.3 Outros Objetos

- a) As tabelas e colunas devem estar documentadas conforme definido neste documento.
 - b) No caso de ausência de constraint onde deveria existir, deve ser verificado o motivo desse tipo de ocorrência e caso seja necessário a não alteração, isso deve possuir uma justificativa técnica e ser documentada.
 - c) O desenho gráfico do modelo de dados deve possuir estética agradável com utilização de cores para diferenciar módulos funcionais e tabelas compartilhadas de outros modelos, além do que deve ser evitado o cruzamento de linhas.
-

3.4.4 Qualidade Negocial

- a) Nome de qualquer objeto deve estar de acordo com a sua finalidade, sendo que para tabelas e colunas isso deve ser sempre verificado e, para outros tipos de objetos, somente aqueles que devem ter o nome representativo de acordo com o negócio, conforme especificado no documento de normas de nomenclatura de objetos de banco de dados.
- b) O modelo de dados deve ser aderente aos documentos de especificação do projeto. No caso de situações onde esta regra não seja seguida, deve ser documentada a justificativa para tal.

Com as orientações deste documento seguidas, poderemos ter um modelo com qualidade.



4. DICIONÁRIO DE DADOS

4.1 Diretrizes para Documentação do Modelo de Dados

Neste item vamos descrever algumas regras básicas que devem ser seguidas para que possamos ter um bom dicionário de dados que é o “Registro detalhado dos conceitos que compõem um universo pré-definido”.

Portanto é importante entender que “A representação gráfica e denominação dos elementos que compõem um modelo de dados não são suficientes para traduzir todos os conceitos do negócio”.

Para que tenhamos uma boa descrição dos elementos de dados devemos considerar os seguintes fatores:

- clareza;
- objetividade;
- respeito à legislação e normativos;
- respeito à terminologia da área negocial que está sendo tratada;
- respeito às normas da língua portuguesa; e
- citação de exemplos.

E para gerar um dicionário de dados que seja bem compreendido e que gere um metadados de qualidade, o que deve ser feito?

Em primeiro lugar, na descrição de uma tabela não deve constar que “armazena dados ...”, mas que representa uma entidade negocial, então na descrição da tabela deve ser bem-conceituada essa entidade.

No caso das colunas, se a entidade que a tabela representa estiver bem-conceituada, a descrição destas fica simplificada, mas deve ficar claro o que a coluna representa dentro do contexto da entidade.

Para facilitar, a descrição da tabela é interessante responder às seguintes perguntas sobre a entidade que representa:

- O que é a entidade?
- O que faz a entidade?
- Para que serve?
- Quando alguém ou algo passa a ser, ou deixa de ser, um elemento dessa entidade?
- Em caso de tabela de domínio, devem constar exemplos.
- A exclusão é lógica?

Assim, a descrição da tabela é formada pelo conjunto dessas respostas. Outro ponto a ser observado é que nem sempre é conseguido responder a todas as perguntas, mas é preciso tentar.

Para que um modelo de dados seja considerado de qualidade, ele tem que ser bem documentado, ou seja, o dicionário de dados deve conter informação relevante e é obrigatório!



FIQUE LIGADO

Um bom dicionário de dados contribui para a formação de um metadados com informações que agregam valor ao conhecimento do negócio.

4.2 Descrições Específicas

A seguir temos sugestões de algumas descrições específicas de determinadas categorias de colunas que são muito utilizadas. É importante ficar claro que as descrições utilizadas no modelo não precisam ser exatamente iguais às aqui sugeridas, mas o significado deve ser o mesmo.

- Colunas que são PK e alimentadas por sequence:

◦ ORACLE: “Representa a chave primária sequencial da tabela, que é controlada pela sequence [NomeSequence] do banco de dados específica para a tabela.”

- **POSTGRE ou MySQL:** “Representa a chave primária sequencial da tabela, que é controlada pelo banco de dados através do datatype desta coluna SERIAL.” Este caso é para quando não é definida uma sequence específica.

- **Colunas que são FK em uma tabela**

- [Descrição da finalidade da coluna dentro de contexto da entidade negocial representada pela tabela.] + [A coluna é FK da tabela [NomeTabela].]

- **Coluna ST_REGISTRO_ATIVO:**

- “Indica se o registro está ativo ou não (excluído logicamente). O seu domínio é: S – Sim (está ativo) ou N – Não (não está ativo). O controle no uso de registros excluídos deve ser feito pela aplicação.”

- **Coluna que possui CK:**

- [Descrição da finalidade da coluna] + [Domínio]

Observação: Entender como domínio, a indicação de todos os valores e a descrição do significado de cada um (semelhante ao ST_REGISTRO_ATIVO).

- **Tabela de Domínio:**

- O comentário da tabela de domínio, além da descrição de sua finalidade, deve conter a indicação de todos os valores e a descrição do significado de cada um. Caso o domínio seja composto por mais de 10 valores, indicar exemplos.

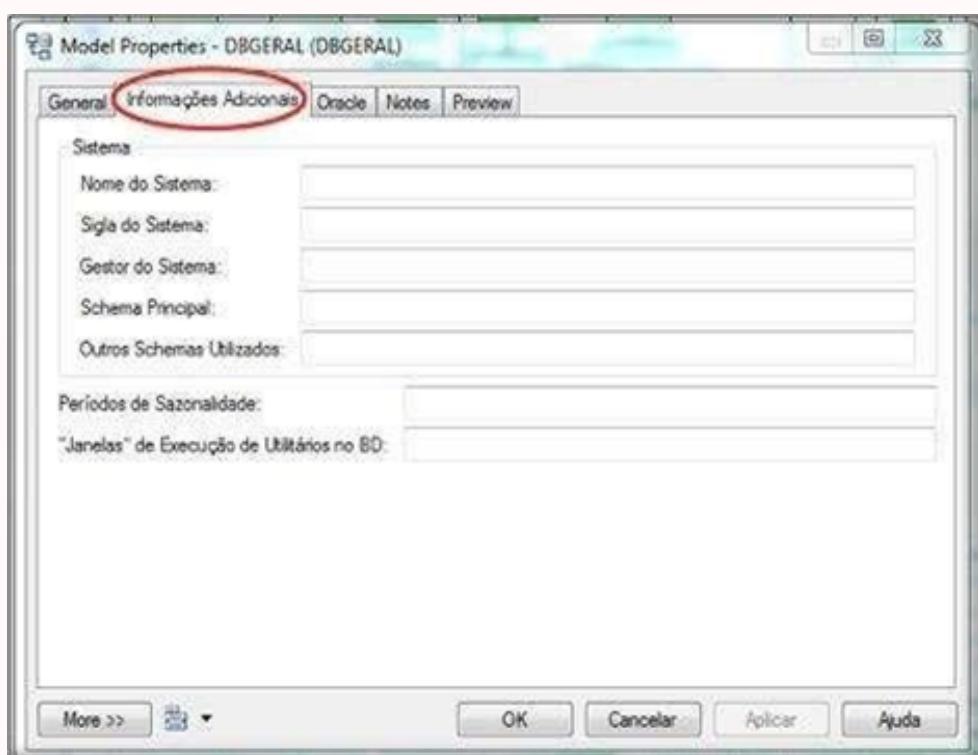
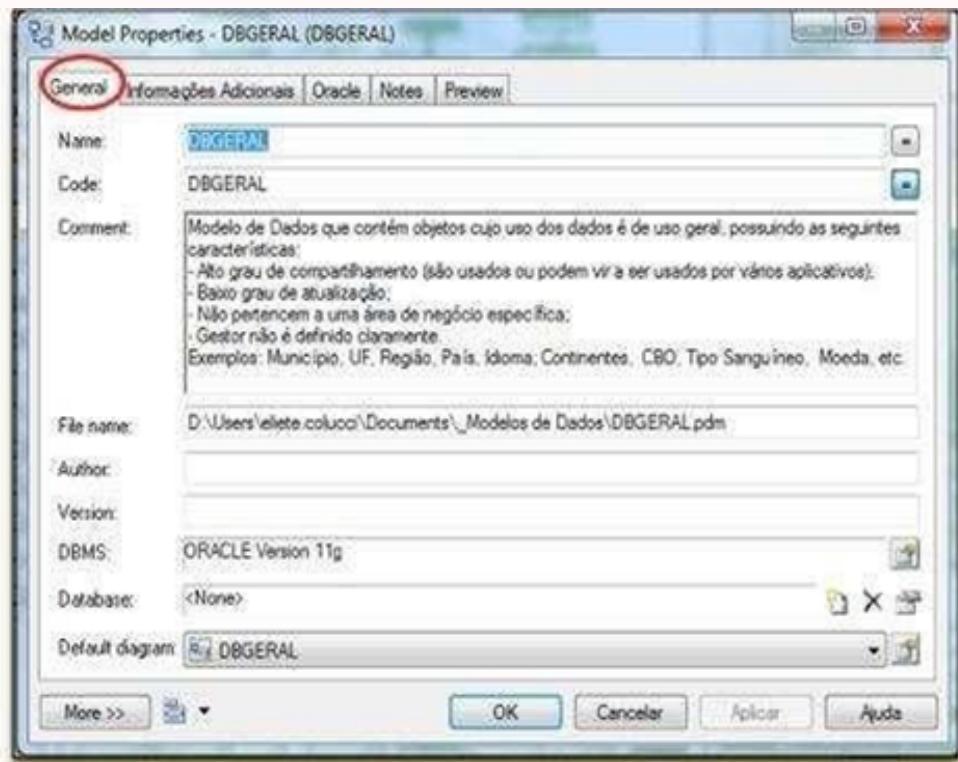
4.3 Armazenamento de Dicionário de Dados

As informações expostas neste anexo devem ficar armazenadas na ferramenta CASE adotada pelo Ministério da Saúde, sendo que o que estiver no **Comment** deve ser gerado via script para armazenamento no Banco de Dados. Os formulários com tais informações do dicionário de dados ficam descritos a partir da Figura 18.

É importante salientar que os formulários são definidos no driver do banco de dados que devem ser fornecidos para todos que utilizam a ferramenta CASE e, para cada SGBD, devemos ter essas definições.

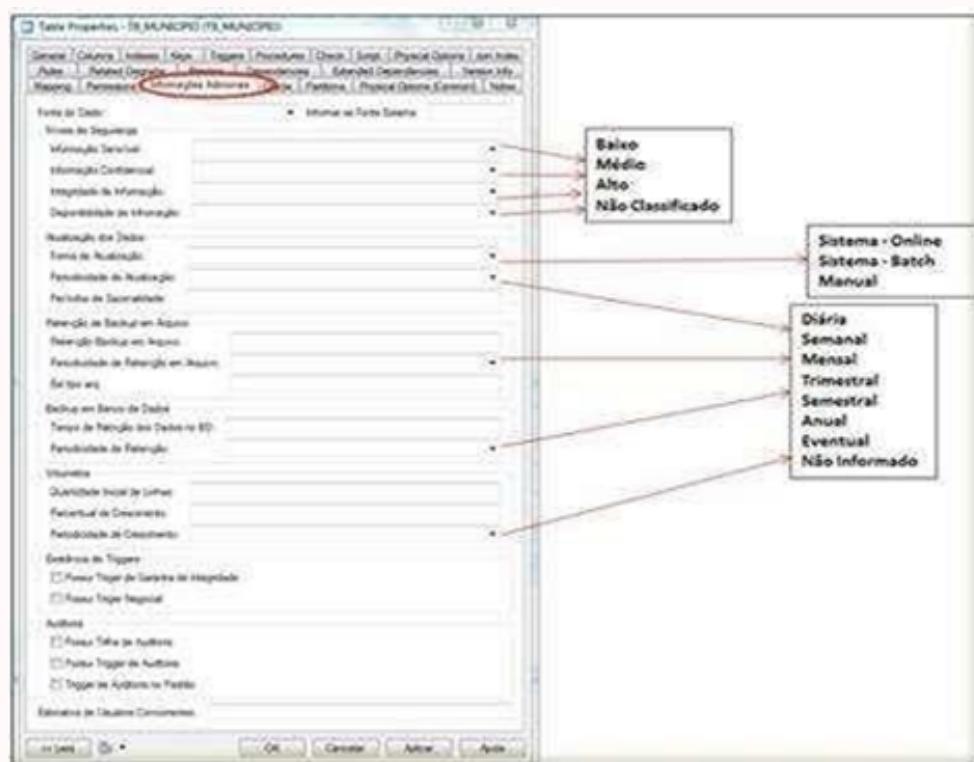
4.3.1 Modelo de Dados

A informação para dicionário de dados para o “Modelo de Dados” na ferramenta case pode ser visualizada nas proximas imagens.



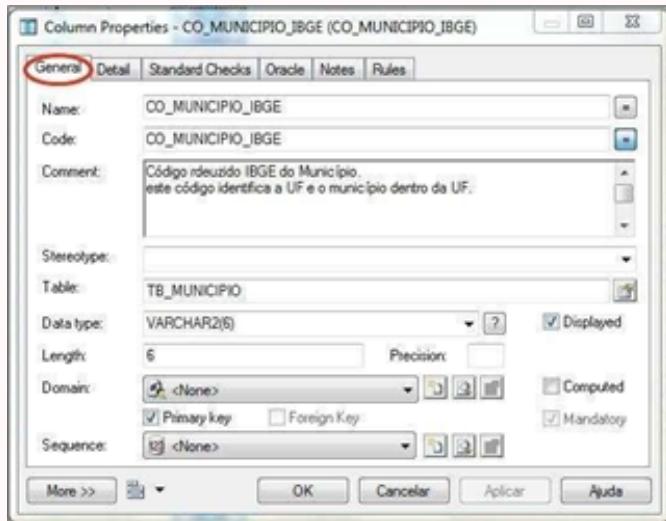
4.3.2 Tabela

A informação para dicionário de dados para a "Tabela" na ferramenta case pode ser visualizada nas proximas imagens.



4.3.3 Coluna

A informação para dicionário de dados para a “Coluna” na ferramenta case pode ser visualizada nas proxima imagem.

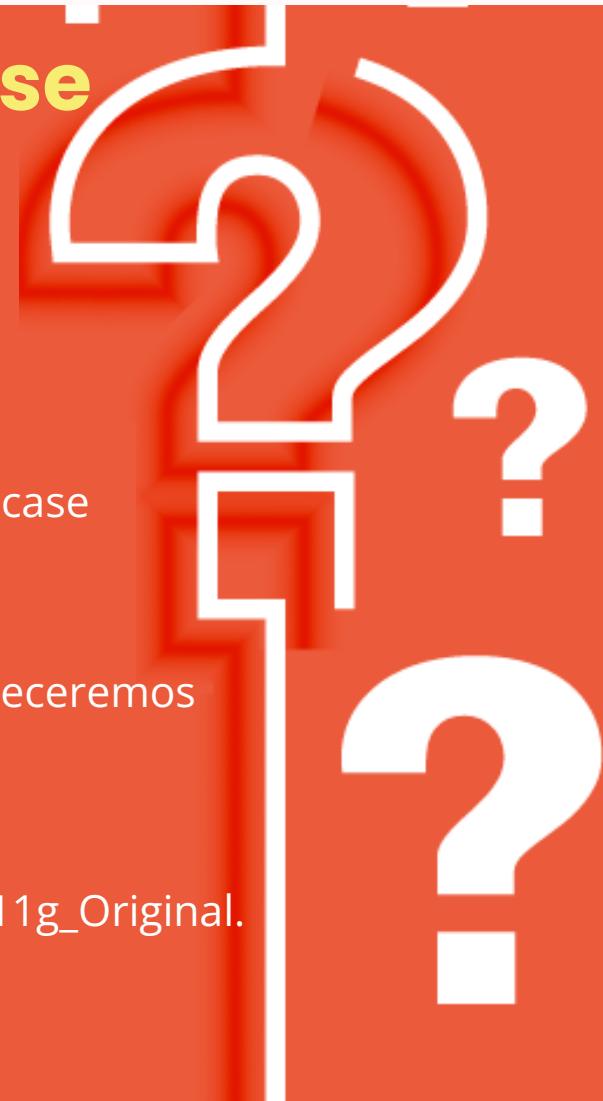


E como faço para esse novo formulários na ferramenta case PowerDesigner

Já temos um drive pronto na ferramenta case PowerDesigner para o SGBD Oracle.

É só procurar a equipe de banco que forneceremos o novo drive ora11g.xdb.

Depois, na pasta *C:\Program Files\Sybase\PowerDesigner15\Resource Files\DBMS*, renomear o arquivo ora11g.xdb para ora11g_Original.xdb e copiar o novo drive.



5. OUTROS TIPOS DE DEMANDAS

5.1 Concessão de Permissões e Auditoria

A segurança de dados é um requisito de gerenciamento de sistemas de banco de dados para proteger os dados contra usuários não autorizados, devendo ser implementada por meio de mecanismos de concessão de permissões de usuários e regras para tais concessões, de forma que haja garantia de proteção dos dados.

É importante salientar que, pelos procedimentos que temos, a concessão de permissões e a geração de AUDITORIA estão ligadas conforme descrito nos itens 5.1.1 e 5.1.2.

Observe que, nos itens 5.1.1 e 5.1.2, quando é dito “Questionar Auditoria”, deve-se considerar os quesitos citados no item 3.2.7.

5.1.1 Concessão para tabelas de próprio esquema

As regras definidas na tabela 3 devem ser consideradas para a concessão de privilégio de **DELETE** em tabelas no próprio esquema.

AMBIENTE	TIPO DE USUÁRIO	AUDITORIA	PROCEDIMENTO	AUTORIZAÇÃO
PRODUÇÃO	Aplicação	Possui auditoria	Conceder acesso	Desnecessária
		Não possui auditoria	Conceder acesso + Auditoria (questionar somente se não existe no desenvolvimento)	Caso seja decidido pela "Não Criação da Auditoria", formalizar por e-mail junto à área gestora / coordenação da área de sustentação responsável
	Manutenção	Não se aplica	Não se aplica	Não se Aplica
HOMOLOGAÇÃO	Aplicação	Possui auditoria	Conceder acesso	Desnecessária
		Não Possui Auditoria	Conceder acesso + Auditoria (questionar somente se não existe no desenvolvimento)	Caso seja decidido pela "Não Criação da Auditoria", há necessidade de formalização por e-mail junto à área gestora / coordenação da área de sustentação responsável da não necessidade de criação de auditoria
	Manutenção	Possui auditoria	Conceder acesso	Desnecessária
		Não possui auditoria	Conceder acesso	Desnecessária
TREINAMENTO	Aplicação	Possui auditoria	Conceder acesso	Desnecessária
		Não Possui Auditoria	Conceder acesso + Auditoria (questionar somente se não existe no desenvolvimento)	Caso seja decidido pela "Não Criação da Auditoria", formalizar por e-mail junto à área gestora / coordenação da área de sustentação responsável
	Manutenção	Possui auditoria - Não precisa	Conceder acesso	Desnecessária
		Não possui auditoria - Não precisa	Conceder acesso	Desnecessária
DESENVOLVIMENTO	Aplicação	Possui auditoria	Conceder acesso	Desnecessária - Informar ao solicitante a necessidade de criação de auditoria em todos os ambientes
		Não possui auditoria	Conceder Acesso + Questionar Auditoria	1) Caso seja decidido pela "Não Criação da Auditoria", formalizar por e-mail junto à área gestora / coordenação da área de sustentação responsável
	Manutenção	Possui auditoria	Conceder acesso	
		Não possui auditoria	Conceder acesso	Desnecessária - Informar ao solicitante necessidade de criação de auditoria em todos os ambientes

5.1.2 Concessão para tabelas de esquema diferente

As regras definidas na tabela devem ser consideradas para a concessão de privilégio de **SELECT/INSERT/UPDATE/DELETE** em esquema diferente, sendo que a auditoria não é necessária para SELECT.

AMBIENTE	TIPO DE USUÁRIO	AUDITORIA	PROCEDIMENTO	AUTORIZAÇÃO
PRODUÇÃO	Aplicação	Possui auditoria	Termo de responsabilidade	SIPAR
		Não possui auditoria	Conceder acesso + Auditoria (questionar somente se não existe no desenvolvimento)	1) SIPAR 2) Caso seja decidido pela "Não criação da auditoria", formalizar por e-mail junto à área gestora / coordenação da área de sustentação responsável
	Manutenção	Não se aplica	Não se aplica	Não se Aplica
HOMOLOGAÇÃO	Aplicação	Possui auditoria	Conceder acesso	
		Não Possui Auditoria	Conceder acesso + Auditoria (questionar somente se não existe no desenvolvimento)	Formalização por e-mail da área gestora responsável pelo objeto envolvido autorizando a concessão de acesso
	Manutenção	Possui auditoria	Conceder acesso	
		Não possui auditoria	Conceder acesso	
TREINAMENTO	Aplicação	Possui auditoria	Conceder acesso	
		Não Possui Auditoria	Conceder acesso + Auditoria (questionar somente do desenvolvimento)	Formalização por e-mail da área gestora responsável pelo objeto envolvido autorizando a concessão de acesso
	Manutenção	Possui auditoria - Não precisa	Conceder acesso	
		Não possui auditoria - Não precisa	Conceder acesso	
	DESENVOLVIMENTO	Possui auditoria	Conceder acesso	
		Não possui auditoria	Conceder Acesso + Questionar Auditoria	Formalizaçao por e-mail de área gestora responsável pelo objeto envolvido autorizando a concessão de acesso. Caso seja a mesma área gestora não há necessidades de autorização de acesso, mas deve ser informado as futuras necessidades para os demais ambientes
		Possui auditoria	Conceder acesso	
		Não possui auditoria	Conceder acesso	

5.2 Solicitação de DROP / TRUNCATE / DELETE

O DROP de um objeto em banco de dados elimina-o, não sendo mais possível recuperá-lo caso não haja uma cópia de segurança.

No caso do comando TRUNCATE em uma tabela, todas as linhas são removidas e não pode ser revertida, como no caso de DELETE é preciso executar comando COMMIT para confirmar exclusão ou o comando ROLLBACK para reverter a transação antes de efetuar a mudança.

Devido a essas características dos comandos DROP e TRUNCATE, são necessários alguns cuidados adicionais.

Pontos Importantes

- O comando TRUNCATE remove todas as linhas de uma tabela.
- Para a solicitação de DROP ou TRUNCATE de tabelas, as regras são as seguintes:
 - **Ambiente de desenvolvimento, treinamento e homologação:**
A operação será realizada sem necessidade de formalização da área gestora responsável pelo objeto.

Nesta situação, deverá ser informada a necessidade de formalização para produção (conforme item abaixo), bem como uma análise de impacto (baseada em FKs e roles), caso a tabela seja utilizada por outros esquemas.
 - **Ambiente de produção:**
Somente será necessária a formalização da área gestora havendo linhas na tabela.

6. INDICADORES DE QUALIDADE DE MODELOS DE DADOS

Para que um modelo de dados seja considerado de boa qualidade, ele deve atender aos requisitos funcionais do projeto e seguir boas práticas de modelagem. Assim, para avaliar se um modelo está aderente às melhores práticas, devem ser observados os seguintes pontos:

a) Acessibilidade

O modelo deve disponibilizar informações de forma clara, simples e rápida, facilitando seu acesso e manipulação.

b) Reutilização

Os objetos do modelo devem ser reutilizáveis em outros contextos da instituição, contribuindo para a criação de um modelo corporativo consistente.

c) Amplitudes

O modelo deve ser completo, profundo e objetivo, abrangendo todas as necessidades de informação do negócio. Além disso, deve ser de fácil compreensão e possuir condições adequadas para atualização.

d) Representação Concisa

O modelo deve conter apenas as informações realmente necessárias ao negócio. Informações irrelevantes ou que não serão mantidas atualizadas tendem a prejudicar a análise e aumentar o esforço de manutenção (por exemplo, tabelas com muitas colunas NULL).

e) Valor Agregado

As informações presentes no modelo devem trazer benefícios práticos ao negócio ou prover vantagens a partir do seu uso.

f) Credibilidade

As informações devem refletir a realidade de forma correta e confiável, conforme legislação vigente e com consistência entre os modelos. Formatos distintos para o mesmo dado devem ser evitados
(ex.: sexo representado como “F/M” em um modelo e “1/2” em outro).

g) Flexibilidade

O modelo deve permitir alterações com impacto minimizado, proporcionando suporte adequado às mudanças necessárias.

h) Documentação

Todos os elementos do modelo devem ser documentados conforme os padrões definidos pelo órgão, com descrições completas, claras e inequívocas.

i) Legibilidade

O diagrama deve ser visualmente agradável e fácil de interpretar, seguindo critérios de estética, tais como:

- evitar curvas e cruzamentos desnecessários,
- manter a hierarquia visual (colocando as entidades “pais” acima das “filhas”),
- organizar relacionamentos de forma limpa e ordenada.

7. CONCLUSÃO

Nesta cartilha temos os principais aspectos que devemos nos preocupar para que tenhamos modelos de dados com qualidade. Essas orientações merecem uma reflexão e uma mudança na construção de modelos de dados, de forma que possamos garantir a qualidade técnica destes, além de que sejam aderentes ao negócio que devem atender.

Estou com uma dúvida. A quem devo recorrer?

E quando há uma dúvida, procure se informar nesta cartilha, ou então consulte quem realmente entende do assunto!

Entre em contato com a equipe de banco. Podemos ser contactados por:
e-mail: datasus.ad@saudade.gov.br

ANEXO

2

PADRÃO DE NOMENCLATURA

HISTÓRICO DE REVISÃO

DATA	AUTOR	DESCRIÇÃO	VERSÃO
02/02/2010	CGAM/DAAED	Esclarecimentos de regras de acesso e concessão de permissões.	2.2
25/05/2012	Eliete Colucci Sousa	Complementação do conteúdo sobre concessão de insert na tabela de log de operações.	2.1
08/09/2014	Eliete Colucci Sousa	Revisão geral do documento, visando a sua publicação como norma do DATA-SUS.	2.2
13/02/2015	Eliete Colucci Sousa	Inclusão de normas para criação de usuários.	2.2
02/09/2015	Eliete Colucci Sousa	Incluir mais esclarecimentos sobre a padronização de nomenclatura.	2.3
01/07/2016	Eliete Colucci Sousa	Alterar a descrição para o uso dos prefixos de colunas VL, QT; prefixos de tabelas Dimensão e Fato.	2.3.1
21/11/2016	Eliete Colucci Sousa	Inclusão da classe de dados.	2.4
30/07/2018	Eliete Colucci Sousa	Especificação de tamanho para campo Sigla (prefixo SG)	2.3.3
10/05/2023	Gislaine Mabel Da Nobrega Trindade	Revisão Técnica do documento e atualização do documento para contemplar o cenário atual do DATASUS	2.3.4
27/12/2024	Isabela Fonseca Lima	Atualizações na última versão do documento	2.3.5

1. OBJETIVO

Padronizar os nomes de objetos no banco de dados criados utilizado neste órgão. Este documento define regras, padrões, nomenclaturas e orientações de uso quanto as bases de dados do DATASUS.

O padrão aqui estipulado segue o ISO/IEC – 11179-5, que foi preparado pelo subcomitê ISO/IEC JCT1/SC32, sendo constituído de várias partes relacionadas, sendo que a parte 5 se refere a padronização e especificação de elementos de dados.

2. ALGUMAS DEFINIÇÕES

- **Constraint** - Regras de validação que podem ser de integridade, de negócio ou estar sujeitas a determinadas condições como valores ou intervalos.
- **Modelo de Dados** - São representações visuais que ilustram as entidades de dados, seus atributos e os relacionamentos entre as entidades de uma organização.
- **Objeto** -Toda parte de uma modelo de dados físico que é passível de implementação em um SGBD, por exemplo, tabelas, constraints, índices, views, etc;
- **SGBD** – Sistema Gerenciador de Banco de Dados.

3. SISTEMAS GERENCIADORES DE BANCO DE DADOS

Os Sistemas Gerenciadores de Banco de Dados (SGBD) utilizados no DATASUS para gerenciamento (armazenamento e recuperação) dos dados no Banco de Dados são:

- Oracle;
- Postgres;
- SQL Server;
- MySQL.

4. NOMENCLATURA DE OBJETOS

O nome do objeto deve apresentar clareza, de forma a indicar a sua finalidade no negócio a que está vinculada. Não deve ferir a norma culta da língua portuguesa.

O nome de um objeto de banco de dados deverá ser formado por uma ou mais palavras com todas as letras em maiúsculas e no singular, cada uma separada pelo caractere separador underscore (_). A sequência das palavras deve ser lógica de tal forma que dê um significado preciso em Português.

No caso de objetos do SGBD MySQL, que é Case Sensitive, todos deverão ser criados / mantidos com letras minúsculas.

Para a formação de cada palavra de um nome, deve-se sempre considerar a portabilidade entre SGBD's, sendo assim utilize apenas os caracteres alfabéticos não acentuados e numéricos.

Para o caso de abreviações, as seguintes regras devem ser observadas:

- Utilize abreviações ou siglas somente quando o nome completo ultrapassar o tamanho máximo estipulado e nestas situações, somente se for de aceitação e conhecimento geral do órgão;
- A palavra principal que compõe o nome de um elemento de dado não deve ser abreviada;
- Na abreviação de palavras, siga as regras definidas no capítulo 5 deste documento.

Palavras em outro idioma não serão permitidas. As exceções deverão ser tratadas caso a caso.

Nome de qualquer tipo de objeto e colunas de tabelas deve possuir a formação Prefixo_Nome Objeto. O prefixo para o caso de tabelas e colunas define uma categorização para cada um, sendo que no caso de Colunas não é indicado tipo e tamanho que deve estar associado, pois isso depende do negócio, mas é importante observar o bom senso nesse tipo de definição, por exemplo, para colunas cujo tamanho é de informações conhecidas como CPF ou CNPJ deve-se utilizar o tipo e tamanho de acordo com a definição existente, sendo CPF com varchar2(11) e

CNPJ com varchar2(14). Para os casos, onde o domínio está definido em tabela do DBGERAL deve ser utilizada está com FK.

O prefixo de cada objeto deve indicar da finalidade do mesmo, conforme definido no Anexo I. Para informações comumente utilizadas e que o conjunto dos atributos contidos em cada classe compõe uma informação foi definido no Anexo II “Classe de Dados”. Exemplos: CPF, telefone, carteira de trabalho, etc. As informações contidas nesse anexo, devem seguir as regras lá definidas.

Para o caso de exclusão lógica de registro em uma tabela o tratamento deve ser o seguinte:

- a. Deve ter um campo com nome ST_REGISTRO_ATIVO datatype VARCHAR2(1) e domínio S ou N;
- b. A sua descrição pode ser “Indica-se o registro está ativo ou não (excluído logicamente). O seu domínio é: S – Sim (está ativo) ou N – Não (não está ativo). O controle no uso de registros excluídos deve ser feito pela aplicação.”;
- c. Quando a tabela é criada já com essa coluna, não é obrigatório a definição de um valor DEFAULT, mas quando a coluna é adicionada em uma tabela já existente, deve-se definir o valor DEFAULT, pois assim a coluna pode ser criada como NOT NULL e o valor DEFAULT será preenchido para todos os registros existentes na tabela.

Para o caso de necessidade de exclusão física de registros deve ser informada a necessidade de auditoria ou não. No caso de não necessidade é preciso autorização do gestor.

Não serão criados sinônimos para acessar os objetos. Será necessário fazer referência ao esquema (owner) antes do nome do objeto.

O tamanho máximo de um nome de objeto não pode exceder 30 caracteres. Para os casos de Intermedia Index o tamanho máximo é de 25 caracteres, devido a uma limitação do SGBD.

No caso de o nome do objeto ultrapassar o tamanho máximo estipulado, a seguinte regra deve ser utilizada:

- Verificar a possibilidade de substituir palavras por acrônimos ou siglas;
- Abreviar as palavras utilizando-se da norma culta de nossa língua.

Na tabela [Padrões para Tipo de Objetos](#), serão tratadas as seguintes regras, que devem ser consideradas para as referências indicadas a seguir.

- **[NOME DA TABELA]:** nesta referência o nome da tabela não deve conter o prefixo, a menos que seja uma tabela com prefixo RL, TL_ ou AU_ ; não deve conter o caractere separador _.
- **[NOME DA COLUNA]:** nesta referência o nome da coluna não deve conter o caractere separador _, sendo que este caractere deverá conter quando for utilizada mais de uma

coluna (para separar cada coluna).

- [NOME FK]: nesta referência o nome da FK deve conter o prefixo; não deve conter o caractere separador _.

4.1 Datatype de Objetos

SGBD	Caracter com Tamanho Máximo Delimitado	Numérico
Oracle	VARCHAR2	Deve ser informada a precisão.
Postgres	VARCHAR	Deve ser informada a precisão.
MySQL	varchar / character varying	Deve ser informada a precisão.
SQLServer	varchar / nvarchar	Deve ser informada a precisão.

5. ABREVIAÇÃO

Para abreviação de palavras contidas nos nomes de objetos deve preferencialmente utilizar a regra definida no documento “Governo Brasileiro - Comitê Executivo de Governo Eletrônico - Catálogo de Padrões de Dados” (item 4 da bibliografia citada no documento principal da MAD). As regras contidas nesse documento são as seguintes: Uso de Siglas / Acrônimos e Abreviaturas na Nomenclatura de Dados.

- Sempre que possível, evitar o uso de abreviaturas/acrônimos, pois prejudicam o entendimento;
- Não devem ser utilizadas preposições, e na necessidade de se utilizar verbos, utilize no presente;
- Se a palavra, termo ou nome não tiver uma sigla /acrônimos conhecidos, use as diretrizes abaixo para construir a abreviatura:
 - Apenas aquelas que tenha no total mais de 8 caracteres podem ser abreviados;
 - A abreviatura deve ter no máximo dois terços do tamanho da palavra original;
 - Acrônimo e abreviaturas devem ter pelo menos 2 caracteres;

Dar preferência a:

- a. Abreviaturas de negócio à termos de tecnologia da informação;
- b. Termos da tecnologia da informação à criação de novos;
- c. Criar abreviaturas evitando ambiguidade;
- d. Regra geral para criar novas abreviaturas para palavra:
 - O escrever a primeira sílaba e a primeira letra da segunda sílaba, ex.: gramática=gram;

portugues=port; numeral=num;

- Se a segunda sílaba iniciar por duas consoantes, escrever as duas, ex.: construção=constr; secretário=secr;

Atenção: Se a abreviatura resultante coincidir com uma existente ou sugerir ambiguidade, escrever a segunda sílaba completa e incluir a primeira letra da terceira sílaba, ex.: profissional=profiss

6. PADRONIZAÇÃO DE NOMES DE OBJETOS

Os dados são armazenados no banco de dados, sob um formato representado por tipo de dados e diferentes formatos.



ATENÇÃO!

- Não usar espaço em branco;
- Não usar hífen, acentos e caracteres especiais;
- Não usar mais que 30(trinta) caracteres;
- Não usar verbos;
- Escrever em maiúsculo ou minúsculo;
- Não utilizar palavras no plural;
- Não usar preposições;
- Não usar números;
- Não usar nomes próprios;
- Separe os nomes com underline;
- Crie nomes sucintos e objetivos ao negócio.

PADRÕES PARA TIPO DE OBJETOS

TIPO DE OBJETO	DESCRIÇÃO	PADRÃO ESTABELECIDO	EXEMPLOS
Bitmap Index	Tipo de índice que pode ser utilizado para otimizar consultas que utilizam como filtro de dados, colunas que possuem baixa cardinalidade (colunas que possuem pouca variação de valores nas linhas de uma tabela).	IB_+[NOME DA TABELA]+_-+[NOME DA COLUNA]	Tabela: TB_USUARIO Coluna: NU_CPF Bitmap Index: IB_USUARIO_NUCPF
Check Constraint (CK)	Definição de restrições de para uma coluna/tabela.	CK para uma coluna de uma tabela: CK_+[NOME DA TABELA] TABELA]+_-+[NOME DA COLUNA]	Tabela: TB_CADASTRO Coluna: ST_FUNCAO Constraint: CK_CADASTRO_STFUNCAO Tabela: TB_UF Coluna: CO_IBGE Constraint: CK_UF_COIBGE
Cluster	Grupo de tabelas que compartilham os mesmos blocos de dados, desde que compartilhem colunas em comum e são usadas frequentemente em conjunto.	Formação com 2 tabelas: TC_+[NOME DA TABELA 1] +_- [NOME DA TABELA 2] Formação com mais de 2 tabelas: TC_+[NOME DA TABELA 1] +_- ... +-[NOME DA TABELA n] Quando o nome com a regra acima não ficar legível utilizar a seguinte regra: TC_+[NOME DO CLUSTER], neste caso o nome do cluster deve ser significativo para o negócio ao qual pertence	Tabela 1: TB_GRUPO Tabela 2: TB_SUBGRUPO Cluster: TC_GRUPOSUBGRUPO IC_CIDADEESTADOCOUF
Cluster Index	Índice para cluster.	IC_+[NOME DO CLUSTER]	
Column	Coluna de uma entidade ne-gocial que é representada em uma tabela, sendo que cada coluna corresponde a uma coluna na tabela.	Ver tabela Padrão de Colunas	Ver tabela Padrão de Colunas
Database Link (Oracle) ou Linked Server (SQL Server)	Objeto criado em um schema que possibilita o acesso para este schema a objetos de outro servidor de banco de dados.	Brasília: [NOME DO BANCO] Rio: LK_[INSTÂNCIA SISTEMA DESTINO]_[SIGLA SISTEMA DESTINO/ENTIDADE EXTERNA ORIGEM]_[FLAG DE LOCALIZAÇÃO]	DFPO1 LK_RJPO1

TIPO DE OBJETO	DESCRIÇÃO	PADRÃO ESTABELECIDO	EXEMPLOS
Esquema	Nomes de schemas não devem utilizar caractere separador e devem ser compostos de apenas uma palavra com no máximo 20 caracteres.	DB[NomeSchema], quando esquema não for de uma aplicação. DB[SiglaAplicacao], quando o esquema for de uma aplicação DBDM[NomeSchema], quando esquema for de DataWarehouse.	DBGERAL, esquema da aplicação cuja sigla é SIARH DBCADSUS, esquema da aplicação cuja sigla é CADSUS DBDMSISAGUA, esquema de DW da aplicação cuja sigla é SISAGUA
Foreign Key	Restrição de integridade que determina que uma coluna ou um conjunto de colunas que possuem valores em outras tabelas. Relativa a uma referência ou a um relacionamento.	Existe um relacionamento entre as duas tabelas: FK_+[NOME DA TABELA PAI]+_[NOME DA TABELA FILHO] Existe mais de um relacionamento entre as duas tabelas: FK_+[NOME DA TABELA PAI]+_[NOME DA TABELA FILHO]+[NOME FK], neste caso o nome da FK deve ser significativo para o negócio ao qual pertence	Tabela Pai: TB_UF Tabela Filho: TB_MUNICIPIO Constraint: FK_UF_MUNICIPIO Tabela Pai: TB_UF Tabela Filho: TB_MUNICIPIO Constraint 1: FK_UF_SERVIDORUFNASCIMENTO Constraint 2: FK_UF_SERVIDORUFENDERECO 3-Formação quando existir relacionamento com uma chave candidata da tabela pai (UK) FK_+[NOME DA TABELA PAI]+[NOME SIGNIFICATIVO DO CAMPO NO PAI]+_[NOME DA TABELA FILHO]+[NOME SIGNIFICATIVO DO CAMPO NO FILHO]
Function	São rotinas de processamento que retornam valores.	FC_+[NOME DA FUNÇÃO]	FC_CALCULA_DV

TIPO DE OBJETO	DESCRIÇÃO	PADRÃO ESTABELECIDO	EXEMPLOS
Index	Estruturas opcionais associadas a tabelas e 'clusters' que permitem que as consultas SQL sejam executadas com melhor performance.	<p>1- Formação simples IN_+[NOME DA TABELA]+_+[NOME DA COLUNA]</p> <p>2- Formação com mais de uma coluna na formação do índice IN_+[NOME DA TABELA]+_+[NOME INDEX], neste caso o nome do index deve ser significativo para o negócio ao qual pertence</p> <p>3- Formação quando o índice for de uma coluna que é FK na tabela com uma coluna IN_+[FK]+[NOME DA TABE- LA]+_+[NOME DA COLUNA]</p> <p>4- Formação quando o índice for de uma coluna que é FK na tabela de mais de uma coluna IN_+[FK]+[NOME DA TABE- LA]+_+[NOME FK]</p>	Tabela: TB_USUARIO Coluna: NU_CPF Index: IN_USUARIO_NUCPF Tabela: TB_LOCALIDADE Colunas: CO_MUNICIPIO_IBGE, CO_UF_IBGE Index: IN_LOCALID_COMUNICIBGECOUFIBGE ou IN_LOCAL_UFMUNICIPIO Tabela: TB_MUNICIPIO Coluna: CO_UF_IBGE Index: IN_FKMUNICIPIO_COUFIBGE Tabela: RL_ESTAB_EQUIPE_PROF FK: FK_EQUIPE_PROF_EQUIPE Coluna 1: CO_MUNICIPIO Coluna 2: CO_AREA Coluna 3: SEQ_EQUIPE Index: IN_FKRLESTEQUPRF_FKEQUPRFEQU

TIPO DE OBJETO	DESCRIÇÃO	PADRÃO ESTABELECIDO	EXEMPLOS
Instância de Banco	Nome de instância de banco de dados.	<p>UF = Unidade da Federação com dois caracteres (DF);</p> <p>Finalidade da Instância:</p> <ul style="list-style-type: none"> P – Produção T – Treinamento D – Desenvolvimento H – Homologação <p>SGBD:</p> <ul style="list-style-type: none"> O – Oracle P – PostgreSql M – MySql S – SqlServer <p>Sequencial: número de 1 a 9, indicando o sequencial daquela instância.</p> <p>Observações:</p> <ul style="list-style-type: none"> - Caso a instância não siga o padrão de nomenclatura, acrescentar F antes do SGBD. - Para instâncias de Data Warehouse, os nomes devem ser acrescidos da letra W antes do SGBD. <p>Rio: [Finalidade da Instância] + UF + [Sequencial], onde: Finalidade da Instância: BD – Produção DS – Desenvolvimento HO – Homologação</p>	<p>DFF01 (fora do padrão)</p> <p>BDRJ01 DSRJ01</p>
Intermedia Index	Índice para pesquisa textual.	<p>ITM_+[NOME DA TABELA]+_-+[NOME DA COLUNA]</p> <p>Obs: O nome da tabela deve perder o prefixo TB e os separadores “_”. O nome da constraint deve conter apenas 25 caracteres, pois é uma limitação do sgbd Oracle.</p>	<p>Tabela: TB_TEXTO Coluna: DS_CONTEUDO Intermédia Index: ITM_TEXTO_DSCONTEUDO</p>

TIPO DE OBJETO	DESCRIÇÃO	PADRÃO ESTABELECIDO	EXEMPLOS
Materialized View	É uma representação de uma ou de várias tabelas armazenadas em banco, mas com armazenamento próprio. A MV também poderá ser utilizada como forma de replicação de tabelas em instâncias distintas.	Views de uma ou mais tabelas MV_+[NOME DA VIEW]	MV_SERVIDOR_ATIVO
Package	Grupo de procedures, funções, comandos sql e variáveis, que executa diretamente no banco de dados.	Views para replicação de tabelas deve seguir a regra de nome de tabela	TB_UNIDADE_UORG
Package Body	Grupamento de comandos PL/SQL e especificações disponíveis para todos os objetos públicos listados na Package.	PKG_+[NOME DO PACKAGE]]+-[BOD Y]	PKG_CADASTRO_BASE
Partition Index	Índice de partição.	PI_+[NOME DA TABELA] +_+[NOME DA COLUNA] +_+[RANGE]	PI_AUTORIZA_DTANO-CMPT_1992
Primary Key	Identifica de forma única uma linha de uma tabela.	PK_+[NOME DA TABELA]	Tabela: TB_MUNICIPIO Primary Key: PK_MUNICIPIO
Schema	Conjunto de objetos de propriedade de um usuário (OWNER).	DB+[SIGLA DO SISTEMA]	Tabela: TB_TIPO_ENTIDADE Primary Key: PK_TIPOENTIDADE
Sequence	Objeto do banco utilizado para gerar números inteiros únicos.	1) Vinculada a uma tabela específica: SQ_+[NOME DA TABELA]+-[NOME DA COLUNA] 2) Não Vinculada a uma tabela específica: SQ_+[NOME DA SEQUENCE], neste caso o nome da sequence deve ser significativo para o negócio ao qual pertence	Tabela: TB_CADASTRO_SERVIDOR Sequence: SQ_CADASTRO-SERVIDO_COSEQPOS ICA Sequence não vinculada à tabela: SQ_GERA_CODIGO
Stored Procedure	Conjunto de procedimentos armazenados no banco.	SP_+[NOME DA PROCEDURE]	SP_ESTRUTURA
Table	Armazenamento dos dados da entidade negocial que representa	Ver Tabela Padrões Para Tipo de Tabelas	Ver Tabela Padrões Para Tipo de Tabelas

TIPO DE OBJETO	DESCRIÇÃO	PADRÃO ESTABELECIDO	EXEMPLOS
Table Partition	Particionamento de tabelas com grande volume de dados, que divide fisicamente tabelas, sem a necessidade de reescrever o código fonte de sua aplicação, pois logicamente a tabela particionada é como uma tabela normal.	PD_+[NOME DA TABELA]_+[NOME DA COLUNA]+_[RANGE]	PD_AUTORIZA_DTA-NOCMPT_1992
Tablespace	Unidade de armazenamento lógico de um banco de dados que consiste em um ou mais arquivos denominados arquivos de dados (datafiles), que são estruturas físicas compatíveis com o sistema operacional onde está o banco de dados. Os dados do banco de dados são armazenados coletivamente nos datafiles que constituem cada tablespace.	<p>Padrão de Brasília: - TD_+[NOME DO SCHEMA] (tablespace de dados)</p> <p>- TI_+[NOME DO SCHEMA] (tablespace de índice)</p> <p>- TT_+[NOME DO SCHEMA] (tablespace temporária)</p> <p>Padrão Rio: - TD_+[NOME DO SCHEMA] _+[Nº DA TABLESPACE] (tablespace de dados)</p> <p>- TI_+[NOME DO SCHEMA] _+[Nº DA TABLESPACE] (tablespace de índice)</p> <p>- TT_+[NOME DO SCHEMA] _+[Nº DA TABLESPACE] (tablespace temporária)</p> <p>- TR_+[NOME DO SCHEMA] _+[Nº DA TABLESPACE] (Tablespace Segmento de Rollback)</p>	TD_DBGERAL TI_DBGERAL TT_TMPSEG TR_RBSSEG1 TD_CNES_01 TI_CNES_01 TT_CNES_01 TR_CNES_01
Trigger After de Auditoria	<p>Procedimento a ser disparado após uma exclusão / atualização / inclusão de registro na tabela associada para inclusão de registro na tabela de auditoria.</p> <p>Essa nomenclatura também deve ser utilizada quando há mais de um tipo de operação disparando a trigger (para cada linha do comando).</p>	<p>Trigger de Auditoria</p> <p>- TRA_[NOME DA TABELA], onde o nome da tabela é com o prefixo da tabela origem, mas sem o caractere separador _ entre as palavras. Ultrapassando o tamanho máximo de caracteres para nome (30), a última palavra será truncada no 30º caractere.</p>	<p>Tabela: TB_PESSOA Trigger: TRA_PESSOA</p> <p>Tabela: RL_DEMANDA_TAREFA Trigger: TRA_RLDEMANDATAREFA</p>

TIPO DE OBJETO	DESCRIÇÃO	PADRÃO ESTABELECIDO	EXEMPLOS
Trigger Before / After Delete (Row / Statement)	Procedimento a ser disparado antes ou após a exclusão de um registro da tabela associada (para cada linha do comando).	Trigger Before Delete TBD_+[NOME DA TABELA] Trigger After Delete TAD_+[NOME DA TABELA]	Tabela: TB_FORNECEDOR_TELEFONE Trigger: TBD_FORNECEDORTELEFONE Tabela: RL_FORNECEDOR_ENDERECO Trigger: TAD_RLFORNECEDOR ENDERECO
Trigger Before / After Insert (Row / Statement)	Procedimento a ser disparado antes ou após a inserção de um registro na tabela associada (para cada linha do comando).	Trigger Before Insert TBI_+[NOME DA TABELA] Trigger After Insert TAI_+[NOME DA TABELA]	Tabela: TB_FORNECEDOR_TELEFONE Trigger: TBI_FORNECEDORTELEFONE Tabela: RL_FORNECEDOR_ENDERECO Trigger: TAI_RLFORNECEDOR ENDERECO
Trigger Before / After Insert or Update or Delete (Row / Statement)	Procedimento a ser disparado antes ou após qualquer uma das três operações de registro (insert, delete, update) na tabela associada. Essa nomenclatura também deve ser utilizada quando há mais de um tipo de operação disparando a trigger (para cada linha do comando).	Trigger Before All TBA_+[NOME DA TABELA] Trigger After All TAA_+[NOME DA TABELA]	Tabela: TB_FORNECEDOR_TELEFONE Trigger: TBA_FORNECEDORTELEFONE Tabela: RL_FORNECEDOR_ENDERECO Triggers: TAA_RLFORNECEDOR ENDERECO
Trigger Before / After Update (Row / Statement)	Procedimento a ser disparado antes ou após a atualização de um registro na tabela associada (para cada linha do comando).	Trigger Before Update TBU_+[NOME DA TABELA] Trigger After Update TAU_+[NOME DA TABELA]	Tabela: TB_FORNECEDOR_TELEFONE Trigger: TBU_FORNECEDORTELEFONE Tabela: RL_FORNECEDOR_ENDERECO Triggers: TAU_RLFORNECEDOR ENDERECO
Trigger Instead Of	Procedimento a ser disparado após a execução de um comando (para todas as linhas afetadas pelo comando).	TIO_+[NOME DA TABELA ou VIEW]	Tabela: TB_FORNECEDOR_TELEFONE Trigger: TIO_FORNECEDORTELEFONE Tabela: RL_FORNECEDOR_ENDERECO Triggers: TIO_RLFORNECEDOR ENDERECO

TIPO DE OBJETO	DESCRIÇÃO	PADRÃO ESTABELECIDO	EXEMPLOS
Unique Key	<p>Identifica de forma única uma linha de uma tabela, mas não é a primary key.</p> <p>Normalmente é utilizada para identificar a chave negocial da tabela nos casos em que a PK é uma coluna controlada por sequence</p>	<p>Formação simples UK_+[NOME DA TABELA]+_+[- NOME DA COLUNA]</p> <p>Formação com mais de um coluna na UK UK_+[NOME DA TABELA]+_+[- NOME UK], neste caso o nome da UK deve ser significativo para o negócio ao qual pertence</p>	<p>Tabela: TB_SERVIDOR Campo: NU_CPF Unique Key: UK_SERVIDOR_CPF</p> <p>Tabela: TB_TELEFONE Campos: TP_TELEFONE, NU_DDD e NU_TELEFONE Unique Key: UK_TELEFONE_IDENTTELEFONE</p>
Usuário de Database Link (padrão geral)	Nomenclatura para links entre esquemas/bancos para os sistemas do DATASUS	<p>LK_[NOME DO SCHEMA]_[UF DE LOCALIZAÇÃO]</p> <p>Obs1: O nome do schema deve perder o prefixo DB e o caractere ("_").</p>	<p>O sistema PORTAL que utiliza o schema dbportal no DF, deseja acessar objetos que estão no banco de dados RJPO1.</p> <p>Nome do link: RJPO1 Nome do usuário: LK_PORTAL_DF</p> <p>Onde: RJPO1 – Banco de Dados que sofre o acesso (destino). PORTAL – Nome do SCHEMA sem o prefixo DB_(origem). DF – Sigla da UF de localização do schema. (origem).</p>
View	Representação lógica de uma ou de várias tabelas armazenadas em banco de dados, mas sem armazenamento próprio.	VW_+[NOME DA VIEW]	VW_SERVIDOR_ATIVO

Observação:

1. Nos casos onde é inserido no padrão [NOME DA TABELA] esta deve ser referenciada sem seu prefixo e sem o caractere _ (separador entre palavras);
2. Nos casos onde é inserido no padrão [NOME DA COLUNA] esta deve ser referenciada sem o caractere _ (separador entre palavras).

6.1 Tipos de Tabelas

A tabela deverá ter seu nome no singular, pois uma tabela é uma entidade, desta forma, apesar de possuir vários dados, ela representa uma informação daquele dado.

PADRÕES PARA TIPOS DE TABELA

TIPO DE TABELA	DESCRIÇÃO	TIPO DE TABELA DE NEGÓCIO (SIM/NÃO)	PADRÃO ESTABELECIDO
Tabela de Sistema	Tabela utilizada para armazenar dados de aplicação.	SIM	TB_+[NOME DA TABELA]
Tabela de Relacionamento (Associativa)	Tabela que resolve relacionamentos "N para N" entre duas tabelas.	SIM	RL_+[NOME DA TABELA1]+_+[NOME DA TABELA2]
Tabela Log de Operações	Tabela utilizada para armazenar os dados das operações de usuários realizadas no esquema. Exemplo de operações: insert, update, delete. Este tipo de tabela somente pode ser utilizado para log's alimentados cuja responsabilidade é da equipe de Desenvolvimento de Sistemas.	NÃO	TL_+[NOME DA TABELA]
Tabela Auditoria	Tabela utilizada para armazenar os dados das operações de usuários realizadas no esquema do padrão da MAD. Exemplo de operações: insert, update, delete. Para estas tabelas a alimentação dos dados será feito por uma trigger e os usuários possuirão somente privilégio de SELECT (uso exclusivo da equipe de Banco de dados)	NÃO	AU_+[NOME DA TABELA ORIGEM,] onde o nome da tabela é com o prefixo da tabela origem, mas sem o caractere separador _ entre as palavras. Ultrapassando o tamanho máximo de caracteres para nome (30), a última palavra será truncada no 30º caráter.
Tabela Temporária	Tabela utilizada em rotinas dos sistemas para armazenamento temporário de dados.	SIM	TM_+[NOME DA TABELA]
Tabela de Histórico	Tabela utilizada para armazenar os dados históricos de uma determinada funcionalidade, podendo ter vida útil para os dados.	SIM	TH_+[NOME DA TABELA]
Tabela Auxiliar	Qualquer tabela utilizada para auxiliar em um procedimento esporádico. Deve ser excluída após a execução do procedimento.	NÃO	TA_+[NOME DA TABELA]
Tabela de Backup	Tabela criada com a finalidade de armazenar um backup de uma tabela.	NÃO	BK_+[YYYYMMDD]+_+[NOME DA TABELA]

TIPO DE TABELA	DESCRIÇÃO	TIPO DE TABELA DE NEGÓCIO (SIM/NÃO)	PADRÃO ESTABELECIDO
Tabela de Relacionamento Ternário (Associativa)	Tabela que resolve relacionamentos "N para N", envolvendo três tabelas ou mais.	SIM	RT_+[NOME DA TABELA DE RELACIONAMENTO] neste caso o nome da TABELA DE RELACIONAMENTO deve ser significativo para o negócio ao qual pertence
Tabela Dimensão (Data Mart / Data Warehouse)	Armazena as descrições textuais das dimensões de negócio (região, UF, etc.). Deve ser utilizada somente para Data Mart ou Data Warehouse.	SIM	TD_+[NOME DA TABELA]
Tabela Fato Dimensão	Coleção de itens de dados relacionados, consistindo de métricas e contexto de dados (são armazenados os indicadores de desempenho do negócio e tem como característica principal a presença de dados altamente redundantes para se obter um melhor desempenho). Deve ser utilizada somente para Data Mart ou Data Warehouse.	SIM	TF_+[NOME DO SISTEMA]

Observação:

1. Nos casos em que é inserido no padrão [NOME DA TABELA] está deve ser referenciada sem seu prefixo e sem o caractere _ (separador entre palavras);
2. Nos casos em que é inserido no padrão [NOME DA COLUNA] está deve ser referenciada sem o caractere _ (separador entre palavras).

6.2 Padrões de nomenclaturas para colunas

PADRÃO DE COLUNAS		
COLUNA	UTILIZAÇÃO	COMPOSIÇÃO
Código	Coluna cujo conteúdo expresse um código, cujo conteúdo não é obtido de uma sequence. Também é utilizada essa regra quando a coluna é uma FK, isto é, coluna herdada de outra tabela, e cuja PK é uma coluna CO_+SEQ_+[NOME DO ATRIBUTO].	CO_+[NOME DA COLUNA]

COLUNA	UTILIZAÇÃO	COMPOSIÇÃO
Código com Sequence	<p>Coluna cujo conteúdo expresse um código obtido de uma sequence.</p> <p>O datatype deve ser de natureza numérica ou SERIAL (para o Postgres).</p> <p>A descrição de colunas desse tipo deve ser: <i>Para sequence específica para a tabela: "Representa a chave primária sequencial da tabela, que é controlada pela sequence [NomeSequence] do banco de dados específica para a tabela."</i></p> <p>Para sequence NÃO específica para a tabela: <i>Representa a chave primária sequencial da tabela, que é controlada sequence [NomeSequence] do banco de dados."</i></p> <p>Para coluna com datatype SERIAL no Postgres: <i>Representa a chave primária sequencial da tabela, que é controlada pelo banco de dados através do datatype desta coluna SERIAL."</i></p>	<p>SQ_+[NOME DA TABELA]+_[NOME DA COLUNA] (vinculada a uma tabela)</p>
Coluna de Controle de Tabela de Auditoria	Coluna que é incluída na geração da tabela de auditoria de colunas utilizadas para armazenamento de informações de controle que permitem o rastreamento da operação na tabela origem.	AU_+[NOME DA COLUNA]
Coordenada Geográfica	Coluna cujo conteúdo expressa um conjunto de coordenadas geográficas. Utilizado para plotar mapas e fazer referenciamento geográfico.	CG_[NOME DA COLUNA]
Data	Coluna cujo conteúdo expresse uma data do calendário civil.	DT_+[NOME DA COLUNA]
Descrição	Coluna cujo conteúdo é livre e em forma cursiva independentemente do tipo e tamanho utilizado (texto).	DS_+[NOME DA COLUNA]
Hora	Coluna cujo conteúdo expresse uma hora ou horário.	HR_+[NOME DA COLUNA]
Imagen	<p>Coluna cujo conteúdo expresse uma binária como imagens, vídeo, audio, bem como qualquer outro tipo de multimídia ou dados em geral.</p> <p>Este tipo de coluna não pode ser mais utilizado.</p>	IM_+[NOME DA COLUNA]
Nome	Coluna cujo conteúdo é de natureza alfanumérica e expressa um nome por extenso sendo composta de palavras, abreviaturas ou ambas.	NO_+[NOME DA COLUNA]
Número	Coluna cujo conteúdo é representado por algarismos, não significando, necessariamente, que o tipo do campo tenha que ser possuir datatype de natureza numérica.	NU_+[NOME DA COLUNA]
Quantidade	Coluna cujo conteúdo expressa um quantitativo. O datatype deve ser de natureza numérica.	QT_+[NOME DA COLUNA]

COLUNA	UTILIZAÇÃO	COMPOSIÇÃO
Sigla	Coluna cujo conteúdo expressa uma Sigla representativa de algo. O datatype deve ser de natureza alfanumérica. tamanho de uma sigla não deve ultrapassar 10 caracteres. Em caso de necessidade de tamanho maior, somente com justificativa.	SG_+[NOME DA COLUNA]
Situação ou Status	Coluna cujo conteúdo expressa a situação ou o status do registro ou de algum atributo. Deve ter uma lista de valores atrelada, que pode ser uma tabela de domínio ou uma check constraint. Obs.: Esta categoria de registro deve expressar um código, seja numérico ou alfanumérico, nunca um conteúdo por extenso/ discursivo.	ST_+[NOME DA COLUNA]
Taxa	Coluna cujo conteúdo expressa indica uma taxa. O datatype deve ser de natureza numérica.	TX_+[NOME DA COLUNA]
Tipo	Coluna cujo conteúdo expressa o tipo do registro ou de algum outro atributo. Deve ter uma lista de valores atrelada, que pode ser uma tabela de domínio ou uma check constraint. Obs.: Esta categoria de registro deve expressar um código, seja numérico ou alfanumérico, nunca um conteúdo por extenso / discursivo.	TP_+[NOME DA COLUNA]
Valor	Coluna cujo conteúdo expressa um valor numérico. O datatype deve ser de natureza numérica.	VL_+[NOME DA COLUNA]

Observação:

1. Nos casos onde é inserido no padrão [NOME DA COLUNA] esta deve seguir as normas especificadas no item 4.1.

7. CLASSE DE DADOS

CLASSE DE DADOS			
CLASSE	NOME DESCRIPTIVO DO ATRIBUTO	DATATYPE/TAMANHO	OBSERVAÇÕES
Caixa Postal	Número da Caixa Postal	VARCHAR2(12)	Deve ser preenchida somente com dígitos de 0 à 9.
	Número do CEP	VARCHAR2(8)	Deve estar cadastrado na tabela DBGERAL.TB_CEP.
CEP	Número do CEP	VARCHAR2(8)	Deve estar cadastrado na tabela DBGERAL.TB_CEP.
Ano	Número do Ano	VARCHAR2(4)	Deve possuir somente dígitos de 0 à 9.
Mês	Número do Mês	VARCHAR2(2)	Deve possuir somente os valores 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11 e 12.
Número de Documento Identificador de Pessoas	Número do CNPJ	VARCHAR2 (14)	Deve estar cadastrado nas tabelas do schema DBPESSOA TB_PESSOA e TB_PESSOA_JURIDICA.
	Número do CPF	VARCHAR2 (11)	Deve estar cadastrado nas tabelas do schema DBPESSOA TB_PESSOA e TB_PESSOA_FISICA.
	Número do Cartão Nacional de Saúde (CNS)	VARCHAR2(15)	Deve possuir somente dígitos de 0 à 9 e estar de acordo com o CADSUS.
	Número do Passaporte	N/A	N/A
Gênero (Sexo)	Número do NIS, PIS, PASEP	VARCHAR2(11)	Deve possuir somente dígitos de 0 à 9.
	Código do gênero (sexo)	VARCHAR2(1)	Deve possuir somente os códigos e referenciado pelo DBGERAL.TB_SEXO: M=Masculino; F=Feminino; I=Ignorado; X=Inválido (Para uso pelo CAD-SUS).

CLASSE	NOME DESCRIPTIVO DO ATRIBUTO	DATATYPE/TAMANHO	OBSERVAÇÕES
Documento de Identidade de Pessoa Física	Documento de Identidade	N/A	A formatação e colunas devem ser livres, mas sempre deverá conter as informações nº do documento, órgão emissor, UF emissora, data de emissão do documento.
	Código Tipo do Documento	VARCHAR2(3)	Deve estar cadastrado em DBGERAL.TB_TIPO_DOCUMENTO .CO_TIPO_DOCUMENTO
Título de Eleitor	Número do Título de Eleitor	VARCHAR2(12)	Deve possuir somente dígitos de 0 à 9.
	Número da Zona Eleitoral do Título de Eleitor	VARCHAR2(4)	Deve possuir somente dígitos de 0 à 9.
	Número da Seção Eleitoral do Título de Eleitor	VARCHAR2(4)	Deve possuir somente dígitos de 0 à 9.
Carteira de Trabalho e Previdência Social	Número da Carteira de Trabalho e Previdência Social (CTPS)	VARCHAR2(7)	Deve possuir somente dígitos de 0 à 9.
	Número de Série da CTPS	VARCHAR2(4)	Deve possuir somente dígitos de 0 à 9.
	Sigla da UF de emissão da CTPS	VARCHAR2(2)	Deve estar cadastrada da tabela DBGERAL.TB_UF.
	Data da emissão da CTPS	DATE	N/A
Telefone	Número do Telefone	VARCHAR2(15)	Deve possuir somente dígitos de 0 à 9.
	Número do DDD	VARCHAR2(2)	Deve possuir somente dígitos de 0 à 9.
	Número do DDI	VARCHAR2(3)	Deve possuir somente dígitos de 0 à 9.
	Número do Ramal	VARCHAR2(4)	Deve possuir somente dígitos de 0 à 9.
	Tipo do Telefone	VARCHAR2(2)	Deve estar cadastrado na tabela DBGERAL.TB_TIPO_TELEFONE. CO_TIPO_TELEFONE.
Situação Ativo / Inativo	Campo de situação, com prefixo ST, cujo domínio é Ativo ou Inativo.	VARCHAR2(1)	Deve ter preenchimento obrigatório e possuir somente os códigos: A - Ativo I - Inativo
Situação Sim / Não	Campo de situação, com prefixo ST, cujo domínio é Sim ou Não.	VARCHAR2(1)	Deve ter preenchimento obrigatório e possuir somente os códigos: S - Sim N - Não
InterNet	Descrição do endereço de E-mail.	VARCHAR2(60)	N/A
	Descrição do endereço da URL (Localizador Padrão de Recursos - Uniform Resource Locator).	VARCHAR2(100)	N/A

CLASSE	NOME DESCRIPTIVO DO ATRIBUTO	DATATYPE/TAMANHO	OBSERVAÇÕES
	Código do Banco	VARCHAR2(3)	Deve possuir somente dígitos de 0 à 9 e estar cadastrado na tabela DBGERAL.TB_BANCO.
Dados Bancários de Conta Corrente	Código da Agência com o dígito verificador	VARCHAR2(6)	Deve estar cadastrado na tabela DBGERAL.TB_AGENCIA_BANCARIA.
	Número da Conta corrente sem o dígito verificador	VARCHAR2(20)	O formato é o seguinte: 0NNN-ND, onde NNNN é o número da agência e D é o dígito verificador da agência. No caso de não se conhecer o DV, preencher com o caractere _ (underscore).
			O último é o dígito digitador do nº da conta bancária.

ANEXO

3

**GUIA RÁPIDO DE
MODELAGEM DE DADOS**

ITEM	DESCRIÇÃO
SGBD	São utilizados: Oracle; Postgres; SQL Server e My SQL.
Primary Key (PK)	Obrigatório para toda tabela de negócio, se for uma sequence, é interessante a definição de uma UK como chave negocial.
Foreign Key (FK)	Relacionamento entre tabelas. Quando for composta é interessante que Do preenchimento das colunas seja obrigatório.
Default de Coluna	Define um valor para o preenchimento da coluna, quando este não é definido na inclusão de uma linha na tabela. Para que o uso desse valor seja correto, é obrigatório que a coluna tenha preenchimento obrigatório .
Coluna com Lista de Valores	<p>Coluna com lista de valores deve ter preenchimento obrigatório (CK ou FK de tabela de domínio) → na necessidade de NULL, criar um código preenchido com totalmente com dígito 9.</p> <p>Uso de CK Obrigatório:</p> <ul style="list-style-type: none"> • Domínio possui dois valores e mais o valor para NULL conforme descrito acima; • Os valores contidos na lista devem indicar que são estáveis, isto é, que a lista não vai aumentar. • <p>Domínios de Sim / Não: valores devem ser obrigatoriamente S ou N.</p> <p>Uso de Tabela de Domínio Obrigatório: Quando a lista de valores não se encaixa nas características definidas para CK. Nesse caso as tabelas que necessitarem do uso do domínio, deverão obrigatoriamente receber uma FK da tabela origem.</p>
Date x Timestamp	Date – data / data e hora com precisão de segundos Timestamp – data e hora com precisão de milissegundos e timezone (fuso horário).
Exclusão Lógica	Para o caso de exclusão lógica de registro em uma tabela esta deve ter obrigatoriamente um campo com nome ST_REGISTRO_ATIVO datatype VARCHAR(1) e domínio S/N e preenchimento obrigatório . A sua descrição deve ser <i>"Indica se o registro está ativo ou não (excluído logicamente). O seu domínio é: S – Sim (está ativo) ou N – Não (não está ativo). O controle no uso de registros excluídos deve ser feito pela aplicação."</i>
BLOB	Não deve ser armazenado em base de dados, mas sim utilizando a tecnologiaFS (Network Fife System).
Triggers	O uso de triggers para implementação de regras de negócio é aconselhável evitar. É interessante o uso para auditoria, histórico e garantia de integridade quando não é possível por constraints.
Generalização / Especialização de Tabelas	Sempre que for identificada a existência de tabelas distintas com muitas propriedades, atributos e relacionamentos em comum recomenda-se o uso do conceito de generalização / especialização.
Normalização	As tabelas devem seguir obrigatoriamente as regras de normalização. As exceções serão tratadas as parte e com justificativa técnica.
Estrutura de uma Tabela	Os dados contidos nas colunas das tabelas devem agregar valor ao negócio. Tabelas com muitas colunas com preenchimento opcional devem ser bem discutidas, pois isso pode ser um indicativo de que existam informações não necessárias ao negócio. Além disso, devem estar aderentes aos documentos de especificação do projeto.

ITEM	DESCRÍÇÃO
Documentação	<p>As tabelas e colunas devem obrigatoriamente estar documentadas, seguindo as orientações constantes em https://datasus.saude.gov.br/metodologia-de-administracao-de-dados-mad/.</p> <p>Resumidamente as orientações são:</p> <p>Tabela: a descrição deve conceituar a entidade negocial representada pela tabela, procurando esclarecer para “o que é”, “o que faz” e “para que serve” a entidade.</p> <p>Coluna: a descrição deve conceituar o significado da coluna dentro da entidade (a descrição da coluna fica simplificada, se a tabela estiver bem conceituada). Para as que são FK, deve ser informada a tabela origem da FK.</p> <p>Lista de Valores: se CK descrição da coluna deve conter os valores e significados. Se tabela de domínio descrição da tabela deve conter os valores e significados, sendo todos (até 10) ou exemplos (acima de 10).</p>
Nome de Tabela e Coluna	Devem estar de acordo com a sua finalidade e o padrão de nomenclatura.

NORMAS GERAIS	
ITEM	DESCRÍÇÃO
Nome de Objeto	<p>Deverá ser formado por uma ou mais palavras com todas as letras em maiúsculas e no singular, cada uma separada pelo caractere separador do <i>underscore</i> (_). A sequencia das palavras deve ser lógica de tal forma que dê um significado preciso em Português.</p> <p>Não devem ser utilizadas preposições, e na necessidade de se utilizar verbos, utilize no presente.</p> <p>Palavras em outro idioma, que não seja o Português somente devem ser utilizadas se forem de aceitação e conhecimento geral no país (exemplos: e-mail, marketing, feedback, etc.).</p> <p>Deve possuir a formação Prefixo_NomeObjeto.</p>
Abreviações	Utilize abreviações somente quando o nome completo ultrapassar o tamanho máximo estipulado e nestas situações, somente se for de aceitação e conhecimento geral do órgão.
Tamanho dos Nomes	<p>A palavra principal que compõe um elemento de dado não deve ser abreviada.</p> <p>Schema: máximo de 20 caracteres. Objetos: Máximo de 30 caracteres. Index Intermedia: máximo de 25 caracteres.</p>

OBJETO	PADRÃO ESTABELECIDO
Bitmap Index	IB_+[NOME DA TABELA]+_[NOME DA COLUNA]
CK	CK_+[NOME DA TABELA] +_[NOME DA COLUNA]
Cluster	TC_+[NOME DA TABELA 1] +_+ [NOME DA TABELA 2] (2 tabelas) TC_+[NOME DA TABELA 1] +_...+_+ [NOME DA TABELA n] (mais de 2 tabelas) ou se ficar ilegível TC_+[NOME DO CLUSTER]
Cluster Index	IC_+[NOME DO CLUSTER]
Database Link (Oracle) ou Linked Server (SQL Server)	[NOME DO BANCO] (Brasília) LK_[INSTÂNCIA SISTEMA DESTINO]_[SIGLA] SISTEMA DESTINO/ENTIDADE EXTERNA ORIGEM]_[FLAG DE LOCALIZAÇÃO] (Rio)
Esquema	DB+[SIGLA DO SISTEMA] (no máximo 20 caracteres)
FK	FK_+[NOME DA TABELA PAI]+_+ [NOME DA TABELA FILHO] (2 tabelas) FK_+[NOME DA TABELA PAI]+_+ [NOME DA TABELA FILHO]+[NOME FK] (2 tabelas com mais de um relacionamento) FK_+[NOME DA TABELA PAI]+[NOME SIGNIFICATIVO DO CAMPO NO PAI]+_+[NOME DA TABELA FILHO]+[NOME SIGNIFICATIVO DO CAMPO NO FILHO] (relacionamento com uma UK)
Function	FC_+[NOME DA FUNÇÃO]
Index	IN_+[NOME DA TABELA]+_+[NOME DA COLUNA] (uma coluna) IN_+[NOME DA TABELA]+_+[NOME DA COLUNA 1] ...]+_+[NOME DA COLUNA n] ou IN_+[NOME DA TABELA]+_+[NOME INDEX] (mais de 2 colunas) IN_+[FK]+[NOME DA TABELA]+_+[NOME DA COLUNA] (index em coluna FK – 1 coluna) N_+[FK]+[NOME DA TABELA]+_+[NOME FK] (index em coluna FK – mais de 1 coluna)
Instância BD	Brasília: UF + [Finalidade da Instância] + SGBD + [Sequencial], onde: Finalidade da Instância: P / D / T / H SGBD: O / P / M / S Rio: [Finalidade da Instância] + UF + [Sequencial], onde: Finalidade da Instância: BD / DS / HO
Intermedia Index	ITM_+[NOME DA TABELA]+_+[NOME DA COLUNA]
Materialized View	MV_+[NOME DA VIEW] TB_+[NOME DA TABELA] (repl. de tab.)
Package	PKG_+[NOME DO PACKAGE]view

NORMAS GERAIS

ITEM	DESCRIÇÃO
Package Body	PKG_+[NOME DO PACKAGE]+_+[BODY]
Partition Index	PI_+[NOME DA TABELA] +_[NOME DA COLUNA] +_[RANGE]
PK	PK_+[NOME DA TABELA]
Sequence	SQ_+[NOME DA TABELA]+_[NOME DA COLUNA] (vinculada a uma tabela) SQ_+[NOME DA SEQUENCE] (não vinculada a uma tabela)
Stored Procedure	SP_+[NOME DA PROCEDURE]
Table Partition	PD_+[NOME DA TABELA] +_[NOME DA COLUNA]+_[RANGE]
Tablespace	Brasília: TD_+[NOME DO SCHEMA] TI_+[NOME DO SCHEMA] TT_+[NOME DO SCHEMA] TR_+[NOME DO SCHEMA] Rio: TD_+[NOME DO SCHEMA]]_+[Nº TABLESPACE] TI_+[NOME DO SCHEMA]]_+[Nº TABLESPACE] TT_+[NOME DO SCHEMA]]_+[Nº TABLESPACE] TR_+[NOME DO SCHEMA]]_+[Nº TABLESPACE]
Trigger	TRA_[NOME DA TABELA] (auditoria – uso exclusivo da GAAD) TBD_[NOME DA TABELA] (before delete) TAD_[NOME DA TABELA] (after DELETE) TBI_[NOME DA TABELA] (Before INSERT) TAI_[NOME DA TABELA] (After INSERT) TBU_[NOME DA TABELA] (Before UPDATE) TAU_[NOME DA TABELA] (Afte Update) TBA_[NOME DA TABELA] (Before + 1 oper.) TAA_[NOME DA TABELA] (After + 1 oper.) TIO [NOME DA TABELA] (Instead Of)
Unique Key	UK_+[NOME DA TABELA]+_[NOME DA COLUNA] (1 coluna) UK_+[NOME DA TABELA]+_[NOME DA COLUNA 1]]+_[NOME DA COLUNA n] (mais de 1 coluna) ou UK_+[NOME DA TABELA]+_[NOME UK]
Usuário de Database Link	LK_[NOME DO SCHEMA]_[UF DE LOCALIZAÇÃO]
View	VW_+[NOME DA VIEW]

CATEGORIA TABELA	PADRÃO ESTABELECIDO
Sistema (negócio)	TB_[NOME DA TABELA]
Relacionamento (negócio)	RL_[NOME DA TABELA PAI]_[NOME DA TABELA FILHA]
LOG (sustentação)	TL_[NOME DA TABELA]
Auditoria (GAAD)	AU_[NOME DA TABELA] (
Temporária (negócio)	TM_[NOME DA TABELA]
Histórico (negócio)	TH_[NOME DA TABELA]
Auxiliar	TA_[NOME DA TABELA]
Backup	BK_[NOME DA TABELA]
Relac. Ternário (negócio)	RT_[NOME DA TABELA]
Data Mart - Dimensão (negócio)	TD_[NOME DA TABELA]
Data Mart - Fato (negócio)	TF_[NOME DA TABELA]

CATEGORIA COLUNA	PADRÃO ESTABELECIDO
Código	CO_+[NOME DA COLUNA]
Código com Sequence	CO_+SEQ_+[NOME DA COLUNA]
Coluna de Contr. de Auditoria	AU_+[NOME DA COLUNA] (uso exclusivo da GAAD)
Coord.Geográfica	CG_[NOME DA COLUNA]
Data	DT_+[NOME DA COLUNA]
Descrição	DS_+[NOME DA COLUNA]
Hora	HR_+[NOME DA COLUNA]
Imagen	IM_+[NOME DA COLUNA]
Nome	NO_+[NOME DA COLUNA]
Número	NU_+[NOME DA COLUNA]
Quantidade	QT_+[NOME DA COLUNA]
Sigla	SG_+[NOME DA COLUNA]
Situação / Status	ST_+[NOME DA COLUNA]
Taxa	TX_+[NOME DA COLUNA]
Tipo	TP_+[NOME DA COLUNA]

PRIVILÉGIOS PADRÃO	
OBJETO	PRIVILÉGIO
Tabelas	SELECT / INSERT/UPDATE DELETE somente sob demanda
Tabelas de Auditoria (AU_)	SELECT
Tabelas de LOG (TL_)	SELECT / INSERT
Tabelas de Histórico (TH_)	SELECT / INSERT UPDATE sob demanda



ATENÇÃO!

DROP e recriação de objetos: os privilégios concedidos são os que estão especificados como padrão neste item. Caso o objeto tenha privilégios que não sigam o padrão e que devem ser mantidos, estes devem ser especificados na demanda.

Os privilégios são concedidos para as roles padrão do esquema e estas para o usuário de manutenção e aplicação.

ANEXO

4

ARTEFATOS

Os artefatos apresentados fazem parte da Metodologia de Administração de Dados (MAD) e têm como objetivo padronizar processos, orientar as equipes e fortalecer a governança de dados no Ministério da Saúde. Cada documento apoia etapas essenciais da gestão da informação, contribuindo para maior organização, qualidade e confiabilidade dos dados.

[Informações do projeto e Cadastro de ferramentas técnicas](#)

[Mapa de migração de dados](#)

[Checklists AD - Repasse de requisitos - Documentação do projeto](#)

[Termo de aceite](#)

[Ata de reunião](#)

[Lista de participantes - reunião](#)



Agora tem
ESPECIALISTAS

Da consulta ao tratamento



MINISTÉRIO DA
SAÚDE

GOVERNO FEDERAL
BRASIL
UNIÃO E RECONSTRUÇÃO