

## Estándares de diseño de Bases de datos

La Dirección General de Gobierno Digital participa activamente a lo largo del ciclo de vida del desarrollo de un software para asegurar la calidad del mismo. Uno de los instrumentos que facilitan esta tarea es la adopción de estándares de diseño de bases de datos.

El uso de estos estándares tiene innumerables ventajas, entre ellas:

- Asegurar la legibilidad del modelo de datos, inclusive para personas que no están relacionadas con el ambiente informático, en etapas de análisis y diseño;
- Facilitar la portabilidad entre motores de bases de datos, plataformas y aplicaciones;
- Facilitar la tarea de los programadores en el desarrollo de los sistemas.

Es por esto que la codificación de las tablas de las bases de datos a desarrollar debe cumplir ciertos requisitos, detallados en el presente documento. Estos requisitos pueden aplicarse a cualquier motor de bases de datos.

Brevemente se resume en los siguientes puntos:

### Reglas generales

- Los nombres de tablas y campos deben especificarse bajo el estándar **camelCase**. Este estándar especifica escribir las palabras compuestas eliminando los espacios y poniendo en mayúscula la primera letra de cada palabra. En este ámbito se utilizará la variante lowerCamelCase (la primer letra del nombre, en minúscula). Para más información, visite <http://en.wikipedia.org/wiki/CamelCase>.

- Únicamente se utilizarán caracteres alfabéticos, salvo que por la naturaleza del nombre se necesiten dígitos numéricos. Se prohíbe el uso de caracteres de puntuación o símbolos.

Ejemplo: localidadesCenso2003.

- Las letras acentuadas se reemplazarán con las equivalentes no acentuadas, y en lugar de la letra eñe (ñ) se utilizará (ni).

Ejemplos: anioExpediente, montoSenia.

Estándares de diseño de bases de datos: 18/10/06 11:46:46

- El nombre elegido debe ser lo más descriptivo posible, evitando términos ambiguos o que se presten a distintas interpretaciones.

Ejemplo: `tiposMunicipios => categoriasMunicipios.`

- El nombre no debe abreviarse, salvo que por necesidad específica deban especificarse más de una palabra en el mismo.

Ejemplo: `ido => idOrganismo, freg => fechaRegistro`

- Agregar comentarios a las bases de datos y los campos, sobre todo a los booleanos.

## Tablas

### Reglas generales

- Los nombres deben especificarse en plural, y de acuerdo a las reglas generales.

Ejemplos: `departamentos, facturas, monedas.`

- En el caso de tablas que se relacionan específicamente con otra tabla (ej. tablas tipo, nomencladores, entidades débiles), esta relación debe quedar expresada en el nombre.

Ejemplos: `domiciliosPersonas, categoriasMunicipios.`

- Las tablas de relación (objetos asociativos, representan relaciones de N a M) deben nombrarse utilizando los nombres de las tablas intervinientes, siguiendo un orden lógico de frase.

Ejemplos: `localidadesMunicipios, facturasNotas`

### Campos clave (Identificadores de tabla)

- Toda tabla debe poseer uno o más campos clave.
- Toda relación entre tablas debe implementarse mediante constraints (claves foráneas) con integridad referencial, de acuerdo al motor de base de datos utilizado.
- La integridad referencial deberá actualizar en cascada en todos los casos, y restringir el borrado salvo para las entidades débiles.

Ejemplos: no se podrá eliminar un registro de la tabla localidades que tenga ocurrencias en otras tablas; para este caso deberá implementarse el borrado lógico. Por el contrario, sí podrá habilitarse el borrado en cascada si la relación fuera entre las tablas facturas y

renglonesFactura.

- Los campos clave deben ubicarse al inicio de la definición de la tabla (deben ser los primeros).
- El nombre del campo clave debe estar compuesto por "id" + nombre de la tabla en singular (para claves no compuestas). Dependiendo de la naturaleza de la entidad, el nombre de la tabla a usar es el de la misma tabla, o el de la relacionada.

Ejemplos: `tabla localidades => idLocalidad.`

- Las claves compuestas sólo deben utilizarse en casos específicos, por ejemplo, tablas de relación o entidades débiles. Si una tabla X con clave compuesta necesita ser referenciada desde otra tabla Y, deberá generarse un campo clave en X al inicio de la misma como "idX", y generar un índice único en los campos que la identificaban.

## Otros campos

- Todo campo que represente un nombre o descripción, se colocará inmediatamente después de los campos clave, y se nombrará como a la tabla a la que pertenece, en singular.

Ejemplos: `tabla localidades => idLocalidad, localidad.`

`tabla sucursalesEmpresas => idEmpresa, idSucursal,`  
`sucursal`

- Algunos campos que representan datos, de acuerdo a su representación conceptual en el ámbito del negocio, deberán prefijarse de la siguiente manera:
  - Números: **num** (ejemplo: Número de factura => numFactura)
  - Fechas: **fecha** (ejemplo: Fecha de inscripción => fechaInscripcion)
  - Códigos: **codigo** (ejemplo: Código de producto: codigoProducto)
- Los campos booleanos deberán nombrarse de acuerdo al estado correspondiente al valor 1/Verdadero/True de los mismos.

Ejemplos: `autorizado, oculto, vigente.`

- Los campos de relación (foreign keys, claves foráneas) deben nombrarse de la misma manera que los campos clave (usando el nombre de la tabla a la que hacen referencia).

Ejemplos: `tabla personas => idTipoDocumento, idEstadoCivil`

Estándares de diseño de bases de datos: 18/10/06 11:46:46

- Si el framework de desarrollo utilizado genera o requiere nombres de campos con nomenclatura específica, deberá consultarse a la Dirección General de Gobierno Digital la conveniencia de su uso.