|  |
| --- |
| ***Estándares de Codificación y Base de Datos*** |

Contenido

[Estándares de Base de Datos 2](#_Toc422732988)

[Base de datos 2](#_Toc422732989)

[Tablas 3](#_Toc422732990)

[Campos 4](#_Toc422732991)

[PK 4](#_Toc422732992)

[FK 4](#_Toc422732993)

[Tabla de Abreviaturas de Campos 4](#_Toc422732994)

[Funciones 4](#_Toc422732995)

[Procedimientos 5](#_Toc422732996)

[Índices 5](#_Toc422732997)

[Vistas 5](#_Toc422732998)

[Triggers 5](#_Toc422732999)

[Aplicaciones 5](#_Toc422733000)

[Estándares de codificación 7](#_Toc422733001)

[Variables 7](#_Toc422733002)

[Funciones 8](#_Toc422733003)

[Parámetros 8](#_Toc422733004)

[Objetos 9](#_Toc422733005)

[Estructura de código 9](#_Toc422733006)

[Controles o Componentes de Interfaz 10](#_Toc422733007)

# Estándares de Base de Datos

## Base de datos

Definir nombre de base de datos.

El nombre de la base de datos será de acuerdo al nombre del proyecto.

<<Utilizar el estándar camelcase.

Ejemplo:

nombreBaseDatos.

Para los backup serán con el nombre de la base de datos

Para los respaldos de base de datos se realizara un formato para registrar las versiones el formato tendrá los siguientes datos.

La tabla tendrá la siguiente estructura

|  |  |  |  |
| --- | --- | --- | --- |
| intVersión | dtmFecha | chrDescripción | chrAutor |
|  |  |  |  |

Nombre: *tblVersiones*

Campos:

*intVersion decimal*

*dtmFecha date*

*chrDescripcion char(200)*

*chrAutor char(70)*

Los cambios en campos se incrementa en decimal y al agregar tablas o se borren se incrementa en enteros.>>

## Tablas

<<Utilizar el estándar camelcase.

Nota: A consideración del diseñador de base de datos se puede agregar una abreviatura de acuerdo al proceso.

Ejemplo: tblCONTablaUno. Donde “*CON*” indica que pertenece al área de contabilidad, se tienen que colocar solo 3 letras.

Los nombres de las tablas inician

Tablas – tbl

Ejemplo:

tblTablaUno.

Utilizar el estándar camelcase.

Todos los nombres serán en singular.

Para controlar los cambios de las tablas.

Se registrara el usuario y fecha

Pero tomando en cuenta que para las tablas que tengan PK se utilizara solo en la principal, pero en los casos en los que no exista una relación entre las tablas se utilizara en todas las tablas.

Cuando se sean tablas de relación separar por un \_

Ejemplo:

tblTabla1\_tblTabla2

>>

Campos

<<Utilizar el estándar camelcase.

Para las claves primarias se utilizara>>

### PK

<< Ejemplo:

pkNombreTabla. Solo para llaves simples

pkNombreCampo1, pkNombreCampo2. Para llaves compuestas>>

### FK

<< Ejemplo

fkNombreCampo. >>

### Tabla de Abreviaturas de Campos

<<

|  |  |
| --- | --- |
| Tipo Dato |  |
| Enteros | int |
| Decimales | dcm |
| Cadenas de Caracteres | chr |
| Fechas/Tiempo | dtm |
| Booleano | bln |
| BLOB | blb |

>>

## Funciones

<<Funciones – fnc

Ejemplo:

fncFuncionUno.

Para los parámetros utilizar p como inicial para nombre del parámetro y m para las variables locales.

Ejemplos:

fncFuncionUno(pNombreParametro)

{

mNombreVariable

}

>>

Procedimientos

<<Procedimientos – prc

Ejemplo:

prcProcedimientoUno.

Para los parámetros utilizar p como inicial para nombre del parámetro y m para las variables locales.

Ejemplos:

prcProcedimientoUno(pNombreParametro)

{

mNombreVariable

}

>>

Índices

<<Index – ind

Ejemplo

indIndexUno.

**>>**

Vistas

<<Vistas - vst

Ejemplo:

vstVistaUno.

>>

Triggers

<<Triggers – trg

Ejemplo:

trgTriggerUno.

>>

# Aplicaciones

<<

**Java**

Manejo de objetos y manejo de n capas mvc (Modelo vista controlador)

Patrones de diseño DAO (Objeto de acceso a datos)

FACADE (Fachada)

Singleton(Instancia única)

Utilización de java Enterprise Edition (JEE)

**Capa de vista**

Utilizar xhtml.

Separar el código de html.

Utilización del Framework Java Server Faces (JSF) la última versión estable.

Nombres de las vistas

Utilizar el estándar camelcase.

NombreCapaNombreArchivo y en singular

**Capa de modelo**

Enterprise Java Bean (EJB)

y para persistencia

Java Persistencia Api (JPA)

Para la persistencia el nombre de la clase es igual al nombre de la tabla.

En modelo pueden ser 2 archivos

-FacadeDAO – Clase con la plantilla de procedimientos generales.

NombreTablaDAO – Clase con los procedimientos de la tabla.

Para el controlador (CTR)

NombreTablaCTR – Nombre de la clase

Para la vista

NombreTabla.xhtml – Nombre de la vista

# Estándares de codificación

## Variables

<<

Se debe agregar documentación en la parte derecha de la declaración de la variable que describa la acción de la misma.

En caso de hacer referencia a campos de base de datos, nombrarla igual que el campo.

Para la estructura de variables que no hacen referencia a base de datos se utilizara el CamelCase.

El nombre debe ser descriptivo de acuerdo a su contenido de la manera más simple posible evitando abreviaturas.

Para las variables globales se utilizara un guion bajo “\_” antes del nombre.

Ejemplo variable global: \_contadorPromedioAprobado = 5.

Ejemplo variable local: contadorPromedioAprobado = 5.

Para el caso de las constantes se nombraran con mayúsculas.

Ejemplo: PI = 3.1416.

>>

## Funciones

<<

Utilizar Camelcase para los nombres

Iniciar el nombre con verbo en infinitivo que describa la acción además puede agregarse el complemento del verbo.

Se debe agregar documentación que describa la acción de la función, variables que recibe y el valor regresado.

El código incrustado dentro de las funciones debe estar identado.

Ejemplo:

*/\*\**

*\* Metodo      : “nombre de la función”*

*\* Descripcion: “descripción de la función”*

*\* parametros  : “Parámetros y su descripción”*

*\* return      : “valor regresado y descripción”*

*\*/*

calcularPromedio()

{

If(condicion)

{

….

}

}

>>

## Parámetros

<<

Utilizar Camelcase para los nombres.

El nombre debe ser descriptivo de acuerdo a su contenido de la manera más simple posible evitando abreviaturas.

Iniciar con una “p” antes del nombre del parámetro.

Ejemplo:

*/\*\**

*\* Método      : “nombre de la función”*

*\* Descripción: “descripción de la función”*

*\* parámetros  : “Parámetros y su descripción”*

*\* return      : “valor regresado y descripción”*

*\*/*

calcularPromedio( pValor1, pValor2)

{

…..

}

>>

## Objetos

<<

Utilizar Camelcase para los nombres.

Se debe agregar documentación que describa la acción de la función, variables que recibe y el valor regresado.

El nombre debe ser descriptivo de acuerdo a su contenido de la manera más simple posible evitando abreviaturas.

Iniciar con una “obj” antes del nombre del objeto.

Ejemplo:

Java Persona objPersona = new Persona();

Php $objPersona = new Persona();

>>

## Estructura de código

<<

Agregar comentarios generales en la parte superior de los archivo.

*/\**

*\* Titulo                  :*

*\* Descripción      :*

*\* Compañía          : Universidad Tecnológica de León*

*\* Fecha de creación :*

*\* Desarrollador               :*

*\* Versión                          :*

*\* ID Requerimiento      : RF12,RN05*

*\*/*

Para controles de cambio agregar

*/\**

*\* Descripción      : “descripción del cambio”*

*\* Fecha de modificación:*

*\* Desarrollador               :*

*\* Versión                          :*

*\*/*

Agrupar variables por tipo de funcionalidad

Ejemplo:

*//VARIABLES DE BASE DE DATOS*

…

*//VARIABLES DE MODULO*

…

Etc.

>>

## Controles o Componentes de Interfaz

<<

|  |  |
| --- | --- |
| **Control/Componente** | **Abreviatura** |
| Caja de Texto | txt |
| Combo | cmb |
| Boton | btn |
| Check | chk |
| Radio | rdo |
| Select | slt |
| Text Area | txa |
| Label | lbl |
| Grig | grd |
| Lista | lst |
| Form | frm |
| Calendario | cln |

Ejemplo:

txtNombre

Caja de Texto .- txtNombreControl

>>