Implementación MAAGTIC

Guía de Estándares Java

Versión 1.0

25/05/2011

Histórico de Versiones

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 25/05/2011 | 1.0 | Generación de documento | Patricia Vázquez Montellano |
|  |  |  |  |

Tabla de Contenido

[1. Introducción 5](#_Toc294256908)

[1.1 Objetivo 5](#_Toc294256909)

[1.2 Definiciones, acrónimos y abreviaturas 5](#_Toc294256910)

[1.3 Referencias 5](#_Toc294256911)

[2. General 5](#_Toc294256912)

[2.1 Estructura de directorios 5](#_Toc294256913)

[3. Estándares de programación Java 6](#_Toc294256914)

[3.1 Referencias 6](#_Toc294256915)

[3.2 Comentarios Iniciales 6](#_Toc294256916)

[3.3 Sangría 6](#_Toc294256917)

[3.4 Longitud de las líneas 6](#_Toc294256918)

[3.5 Saltos de línea 6](#_Toc294256919)

[3.6 Inicialización de variables 7](#_Toc294256920)

[3.7 Paquetes y sentencias 7](#_Toc294256921)

[3.8 Comentarios de bloque 7](#_Toc294256922)

[3.9 Comentario de una sola línea 7](#_Toc294256923)

[3.10 Comentarios específicos 8](#_Toc294256924)

[3.11 Declaración de variables 8](#_Toc294256925)

[3.12 Declaración de clases e interfaces 8](#_Toc294256926)

[3.13 Declaración if-else 9](#_Toc294256927)

[3.14 Declaración for 9](#_Toc294256928)

[3.15 Declaración while 9](#_Toc294256929)

[3.16 Declaración do-while 9](#_Toc294256930)

[3.17 Declaración switch 9](#_Toc294256931)

[3.18 Declaración try-catch 10](#_Toc294256932)

[3.19 Clases e Interfaces 10](#_Toc294256933)

[3.20 Métodos 10](#_Toc294256934)

[3.21 Variables 10](#_Toc294256935)

[3.22 Constantes 11](#_Toc294256936)

[4. Aplicaciones Web 11](#_Toc294256937)

[4.1 Elementos visuales 11](#_Toc294256938)

[4.2 Imágenes 11](#_Toc294256939)

[4.3 Diccionario de imágenes 12](#_Toc294256940)

[4.4 Botones de uso más común 12](#_Toc294256941)

[4.5 Diagrama de navegación 12](#_Toc294256942)

[5. Diseño páginas Web 13](#_Toc294256943)

[5.1 Estándar de interfaz 13](#_Toc294256944)

[5.2 Áreas de trabajo 14](#_Toc294256945)

[5.3 Páneles 14](#_Toc294256946)

[5.4 Catálogo 14](#_Toc294256947)

**Guía de Estándares Java**

# Introducción

Este documento presenta los estándares para desarrollar una solución tecnológica luego de seguir el proceso de Desarrollo de Soluciones Tecnológicas (DST) según se indica en el MAAGTIC. Estos estándares se refieren básicamente al uso del lenguaje Java como tal y no al uso de herramientas de desarrollo en Java.

## Objetivo

Este documento tiene como objetivo brindar una guía de los estándares a emplear en los desarrollos de aplicaciones tecnológicas con Java.

## Definiciones, acrónimos y abreviaturas

N/A.

## Referencias

* Anexo A. Nomenclatura para identificadores
* Anexo B. Identificadores para tipos de datos en Java
* Anexo C. Identificadores para tipo de datos SQL.
* Anexo D. Base de datos SQL.

# General

## Estructura de directorios

Se generará un directorio que contenga toda la información del proyecto, dentro del cual se albergarán los directorios de cada uno de los módulos que lo conforman.

La estructura a utilizar es la siguiente:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \Proyecto |  |  |  |  |  |
|  | \Sistema01 |  |  |  |  |
|  |  | \java |  |  |  |
|  |  |  | \src |  |  |
|  |  |  | \webcontent |  |  |
|  |  |  | \fnc |  |  |
|  |  |  | \wwwrooting |  |  |
|  |  |  |  | \activex |  |
|  |  |  |  |  | \ayuda |
|  |  |  |  |  | \estilos |
|  |  |  |  |  | \flash |
|  |  |  |  |  | \img |
|  |  |  |  |  | \plantillas |
|  | \Sistema02 |  |  |  |  |

| **Directorio** | **Descripción** |
| --- | --- |
| **\Proyecto** | Es el directorio principal que alberga los archivos de los diferentes módulos que lo conforman. |
| **\Sistema01** | Se generará un directorio por cada uno de los sistemas que conforman el proyecto.  Adicional a estos sistemas se contará con el general, en el cual se almacena la información que se comparte entre los sistemas.  El nombre de este directorio se compondrá de las siglas que se les asigne a cada sistema. |
| **\java** | Contiene los programas que conforman los diferentes módulos de un sistema. Contendrá la estructura que genera el IDE al desarrollar la aplicación (src y Webconten son estándares de Java), además de los directorios de las partes estáticas, como ayudas, manual de usuario, imágenes, entre otras (a partir del directorio wwwrooting). |
| **\fnc** | Contiene los archivos de las funciones de JavaScript (\*.js) |
| **\activex** | Contiene los activex de la aplicación. |
| **\estilos** | Contiene la página de estilos utilizada por el sistema. |
| **\flash** | Contiene las animaciones realizadas en flash de la aplicación. |
| **\img** | Contiene las imágenes utilizadas para la interfaz gráfica del módulo. |
| **\plantillas** | Contiene las plantillas de Word o Excel utilizadas para generar documentos bajo este formato. |

# Estándares de programación Java

## Referencias

Los estándares aquí mencionados se obtuvieron del *Code Conventions for the Java Programming Language,* publicado por Oracle Sun Developer Network. Estas convenciones pueden consultarse en la siguiente dirección electrónica para mayor referencia:

<http://java.sun.com/docs/codeconv/index.html>

## Comentarios Iniciales

Todos los códigos fuente deben empezar con un encabezado que contenga: El nombre del archivo, la línea base, fecha e último cambio.

/\*

\* nombre del archivo

\*

\* línea base

\*

\* fecha

\*

\* Ultimo cambio

\*/

## Sangría

Cuatro (4) espacios deben ser usados como sangría o bien el tabulador.

## Longitud de las líneas

Las líneas no deben ser mayores a 80 caracteres.

## Saltos de línea

Cuando una expresión no cabe en una sola línea el salto de línea debe hacerse:

* Después de una coma
* Antes de un operador

Es necesario alinear la nueva línea con el inicio de la expresión de la línea anterior.

Si el alineado hace confuso el código puede utilizarse una sangría.

someMethod(longExpression1, longExpression2, longExpression3,

longExpression4, longExpression5);

var = someMethod1(longExpression1,

someMethod2(longExpression2,

longExpression3));

longName1 = longName2 \* (longName3 + longName4 - longName5)

+ 4 \* longname6; // PREFERIBLE

longName1 = longName2 \* (longName3 + longName4

* longName5) + 4 \* longname6; // EVITARLO

## Inicialización de variables

Tratar de inicializar las variables locales al momento en que son declaradas.

La única razón de no inicializarlas cuando se declaran es si el valor inicial depende de algún cálculo.

## Paquetes y sentencias

La primera línea después de los comentarios iniciales son las sentencias de paquetes y luego los imports utilizados. Ejemplo

package java.awt;

import java.awt.peer.CanvasPeer;

## Comentarios de bloque

Los comentarios de bloque son usados para describir archivos, métodos, estructuras de datos o algoritmos. Se usan al principio de cada archivo o antes de cada método.

/\*-

\* Este es un comentario de bloque con una característica

\* muy especial que es importante considera.

\*

\* uno

\* dos

\* tres

\*/

## Comentario de una sola línea

Los comentarios de una sola línea deben seguir la sangría del código.

Si el comentario no puede ser escrito en una sola línea debe usarse un comentario de bloque.

El comentario debe ser precedido por una línea en blanco.

if (condicion) {

/\* comentario de la condición. \*/

...

}

## Comentarios específicos

Los comentarios específicos pueden aparecer en la misma línea del código que describen pero deben ser suficientemente cortos para separarlos de la condición.

Si más de un comentario específico es necesario para la misma línea de código es mejor usar el comentario de una sola línea.

if (a == 2) {

return TRUE; /\* caso especial \*/

} else {

return isPrime(a); /\* trabaja solo para.. \*/

}

## Declaración de variables

Una declaración de variable por cada línea.

int level; /\* nivel \*/

int size; /\* tamaño de tabla\*/

argv++; // Correcto

argc--; // Correcto

argv++; argc--; // INCORRECTO

## Declaración de clases e interfaces

Para la declaración de clases e interfaces es necesario seguir las siguientes reglas:

* No usar espacios entre el nombre de un método y el paréntesis de inicio “(“.
* Las llaves que abren “{“ aparecen al final de la misma línea de la declaración.
* Las llaves que cierran “}” empiezan en una línea con la misma sangría del estatuto correspondiente, excepto cuando es nulo deben aparecer inmediatamente después de la llave que abre.
* Los métodos están separados por una línea en blanco.

class Sample extends Object {

int ivar1;

int ivar2;

Sample(int i, int j) {

ivar1 = i;

ivar2 = j;

}

int emptyMethod() {}

...

}

## Declaración if-else

if (*condition*) {

*statements*;

}

if (*condition*) {

*statements*;

} else {

*statements*;

}

if (*condition*) {

*statements*;

} else if (*condition*) {

*statements*;

} else {

*statements*;

## Declaración for

for (*initialization*; *condition*; *update*) {

*statements*;

}

## Declaración while

while (*condition*) {

*statements*;

}

## Declaración do-while

do {

statements;

} while (condition);

## Declaración switch

switch (condition) {

case ABC:

statements;

/\* falls through \*/

case DEF:

statements;

break;

case XYZ:

statements;

break;

default:

statements;

break;

}

## Declaración try-catch

try {

statements;

} catch (ExceptionClass e) {

statements;

}

try {

*statements*;

} catch (ExceptionClass e) {

*statements*;

} finally {

*statements;*

}

## Clases e Interfaces

* Las clases deben ser sustantivos.
* El nombre de las clases debe ser simple y descriptivo.
* Si se trata de un nombre compuesto utilizar mayúscula en la primera letra de cada palabra.
* Evitar acrónimos y abreviaturas.

Class Raster;

class ImageSprite;

interface RasterDelegate;

interface Storing;

## Métodos

* Deben ser verbos.
* En el caso de verbos compuestos a partir del segundo verbo la primera letra de cada verbo debe ser mayúscula.

run();  
runFast();  
getBackground();

## Variables

* Deben ser cortos y mnemónicos.
* Evitar variables de un solo carácter, excepto que sean variables temporales usados como contadores. En tal caso debe usarse i, j, k, m y n para variables enteras y c, d y e para caracteres.
* No deben iniciar con guión bajo \_ o signo de pesos $, aunque es permitido por el lenguaje.

int i;

char c;

float myWidth;

## Constantes

Deben ser todas mayúsculas separadas por guión bajo (“\_”).

static final int MIN\_WIDTH = 4;

static final int MAX\_WIDTH = 999;

static final int GET\_THE\_CPU = 1;

# Aplicaciones Web

## Elementos visuales

El nombre de las páginas jsps, servlets o componentes que desplieguen una vista sobre el navegador debe encontrarse conformado por cuatro partes, la primera es el identificador de la página escrito en minúsculas, el número del módulo de dos dígitos, el número del submódulo u opción de menú de dos dígitos y, por último, un consecutivo de las páginas de tres dígitos. Por lo que la sintaxis sería la siguiente:

[Identificador de la página][Número del módulo][Número del submódulo][Consecutivo]

Ejemplos:

pg0101001.jsp/class

pg0205201.jsp/class

## Imágenes

El estándar de las imágenes incluye la forma de clasificar las imágenes, nombrarlas y el formato del diccionario de imágenes. El nombre de las imágenes está compuesto máximo por 12 caracteres (8 para el nombre y tres para la extensión).

Las imágenes están compuestas por tres partes, la primera es el identificador (consta de 1 caracter), la segunda es un nombre representativo (9 caracteres) y el último es un consecutivo (2 caracteres).

Los botones están compuestos de igual manera en la primera y segunda parte pero la última o tercera indica el estado del botón (2 caracteres).

Los estados del botón son:

01 - Activo o Habilitado.

02 - Presionado.

03 - Inactivo o Inhabilitado.

Sintaxis:

[Identificador][Nombre de la imagen o botón][Estado]

La clasificación de las imágenes se muestra en la siguiente tabla.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Imagen** | **Identificador** | **Estado** | **Ejemplo** | |
| Fondos | f |  | fRojo01.jpg | fBann02.jpg |
| Gráficos | g |  | gPrinc01.jpg | gMater02.jpg |
| Fotografías | i |  | iMP01.jpg | iPerson02.jpg |
| Botones | b | 1,2,3 | bAcep01.jpg | bAcep02.jpg |
|  |  | 1,2,3 | bGuard02.jpg | bGuard03.jpg |
| Botones menú | m | 1,2,3 | mCalli01.jpg | mModul02.jpg |
|  |  | 1,2,3 | mAten02.jpg | mSopor03.jpg |

## Diccionario de imágenes

El formato del diccionario de imágenes está compuesto por el nombre de la imagen, la imagen, su tamaño, colores, ubicación física, el frame donde se encuentra y una breve descripción.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nombre** | **Imagen** | **Tamaño** | **Colores** | **Ubicación de la Imagen** | **Contenedor** | **Descripción** |
| gDaed01.jpg |  | 93 x 119 px  (5 KB) | 16M | ./img/daedalus | fr01 | Logotipo representativo del producto. |
| gflash02.gif | Image2 | 357 x 42 px  (3 KB) | 256 | ./img/daedalus | fr02 | Banner que promociona todos los productos |

## Botones de uso más común

A continuación se muestra una lista con el estándar de los botones de uso más común.

| **Acción** | **Activo** | **Presionado** | **Inactivo** |
| --- | --- | --- | --- |
| Nuevo | bNuevo01.jpg | bNuevo02.jpg | bNuevo03.jpg |
| Actualizar | bAct01.jpg | bAct02.jpg | bAct03.jpg |
| Guardar | bGuard01.jpg | bGuard02.jpg | bGuard03.jpg |
| Borrar | bBorra01.jpg | bBorra02.jpg | bBorra03.jpg |
| Cancelar | bCanc01.jpg | bCanc02.jpg | bCanc03.jpg |
| Aceptar | bAcep01.jpg | bAcep02.jpg | bAcep03.jpg |
| Salir | bSalir01.jpg | bSalir02.jpg | bSalir03.jpg |
| Contáctanos | mContc01.jpg | mContct02.jpg | mContc03.jpg |
| Registrar | mRegis01.jpg | mRegis02.jpg | mRegis03.jpg |
| Soporte | mSoprte01.jpg | mSoprte02.jpg | mSoprte03.jpg |
| Download | mDownl01.jpg | mDownl02.jpg | mDownl03.jpg |

## Diagrama de navegación

Este diagrama tendrá en la parte superior derecha el nombre del Site al que pertenece el Diagrama, la fuente a emplear deberá ser Folio Lt BT, con un tamaño de 20 puntos.

Los recuadros que contendrán la descripción y/o nombre de la página tendrán un tamaño de 4.1 cm. de ancho, el alto se define por default de 1 cm., es posible aumentarlo dependiendo del número de flechas que salgan o entren al recuadro, y la letra un tamaño máximo de 10 puntos, pudiéndose adecuar de la manera que mejor convenga para que el texto pueda ser contenido sin problema en dichos recuadros y deberá estar centrado.

La manera en que se comunica una página con otra es a través de líneas con puntas de flecha, las cuáles apuntan a la página destino, cuando sea inevitable que se crucen dichas líneas, se deberán emplear diferentes colores.

La primera página del Módulo se denominará con el nombre del Producto o Servicio. El recuadro que representa esta página se ubicará en la parte superior izquierda del diagrama. A partir de este recuadro se desprenderán las opciones del menú hacia abajo, es decir, la manera en que el sistema crecerá será verticalmente.

A continuación se muestra un ejemplo del diagrama:

Sistema

Módulo

Opción 1

Opción 2

Opción 3

Opción 4

Opción 5

Opción 6

Opción 7

# Diseño páginas Web

La construcción de páginas JSP’s deberán seguir los siguientes estándares para los diferentes tipos de páginas que se requieran:

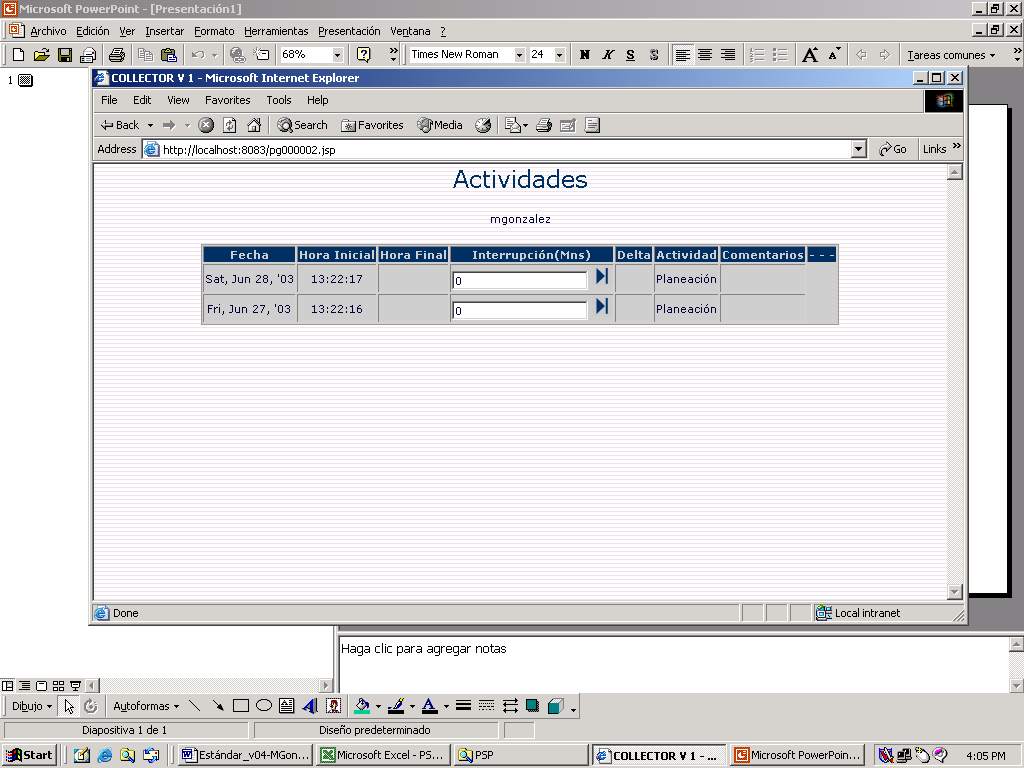
## Estándar de interfaz

Todas las páginas deberán estar divididas en 3 partes: Título, Datos y Páneles.

La parte superior contará con los siguientes elementos:

| **Elemento** | **Descripción** |
| --- | --- |
| 1. Logotipo | Imagen representativa de la SCT |
| 1. Título | Título de la página |
| 1. Botón de Menú | Botones Principales |

Los estilos y colores del sistema, deberán encontrarse guiados por el estándar que se utiliza en el portal de la SCT.



REPORTES



## Áreas de trabajo

En el área de trabajo contendrá toda la información necesaria para el proceso. Podrá tener el scroll vertical pero nunca el horizontal, se deberá ajustar la información a que solo utilice el ancho de la página. El estándar de desarrollo es con el monitor configurado a 800 X 600.

## Páneles

En está área se encuentra los páneles de actualización, navegación y botones propios de la página. En caso de no utilizar alguno de ellos, se tendrán ocultos los botones.

## Catálogo

Los catálogos se distribuirán en la página conforme a la siguiente plantilla:

| **Elemento** | **Contiene ...** |
| --- | --- |
| 1. Frame Principal | Logotipos, Título Principal, Botones Principales |
| 1. Frame Secundario | Subtítulo de la página, Botones secundarios |
| 1. Área de Listado | Listado de Registros |
| 1. Frame Navegación | Paneles de Navegación y Actualización |

**Anexo A. Nomenclatura para identificadores**

Para definir un identificador es necesario seguir las siguientes recomendaciones:

* Los identificadores deben ser una o más palabras que representen el contenido de la información que almacena
* No deben contener acentos, eñes (ñ, Ñ) ni caracteres especiales, incluyendo – y \_
* En caso de que el nombre del identificador se encuentre formado por más de una palabra, debe colocarse la primera letra de cada palabra en mayúscula
* En caso de que la descripción contenga conjunciones, éstas serán omitidas
* Deben utilizarse las abreviaciones naturales de cada palabra, o en su caso, generar la abreviación partiendo de la primera letra y la primera consonante de la palabra
* Debido a la frecuencia con que aparecen algunas palabras, se aceptan como mnemónicos estándar los siguientes:

|  |  |
| --- | --- |
| **Palabra** | **Mnemónico** |
| Catálogo | Ctlgo |
| Centro de Costos | CC |
| Clave | Cve |
| Código Postal | CP |
| Cuenta | Cta |
| Cuentas | Ctas |
| CURP | CURP |
| Descripción | Dsc |
| Detalle | Dtl |
| Dirección | Dir |
| Estado | Edo |
| Encabezado | Enc |
| Efectivo | Evo |
| Flujo | Fjo |
| Número | Num |
| Póliza | Pol |
| Presupuesto | Presto |
| RFC | RFC |
| Saldo | Sdo |
| Teléfono | Tel |

**NOTA.** La lista se irá formando conforme se detecten más palabras comunes, no se permite usar mnemónicos que no se encuentren en este manual, para usarlos es necesario que se incluyan aquí.

**Anexo B. Identificadores para tipos de datos en Java.**

* El identificador de la variable tiene como propósito poder reconocer de una forma rápida el tipo de dato que almacena. El identificador deberá ser siempre escrito con letras minúsculas.
* Se listan los identificadores para los tipos de datos que serán utilizados para el desarrollo

|  |  |
| --- | --- |
| **Tipo** | **identificador** |
| byte | bt |
| Short | Sh |
| int | i |
| Long | lg |
| Float | f |
| Double | d |
| Char | c |
| Boolean | l |

**Tipos de datos nativos de Java**

|  |  |
| --- | --- |
| **Tipo** | **identificador** |
| Integer | i |
| String | c |
| Character | c |
| Double | I |
| Float | d |
| Date | dt |
| Boolean | l |
| TimeStamp | ts |

**Anexo C. Identificadores para tipos de datos sql**

El identificador de la variable tiene como propósito poder reconocer de una forma rápida el tipo de dato que almacena. El identificador deberá ser siempre escrito con letras minúsculas y es independiente a la base de datos .

|  |  |
| --- | --- |
| **Tipo** | **identificador** |
| Number | i |
| Varchar2 | c |
| Date | dt |
| Logical | l |
| Decimal | d |
| Blob | b |
| TimeStamp | ts |

Para definir un tipo de dato lógico se utilizará un dato numérico de una posición, y se le pondrá la validación de que el valor sea entre 0 y 1.

**Anexo D. Base de datos SQL**

**Nombre de la Base de Datos**

El nombre físico de la base de datos debe ser un sustantivo, que haga referencia al contenido de su información, este nombre constará de 8 caracteres y deberá seguir la *Nomenclatura para Identificadores*[[1]](#footnote-1).

Sintaxis:

[Sustantivo]

**Tablas**

El nombre de las tablas constará de 15 caracteres como máximo. Los primeros 3 serán la abreviación en mayúsculas del grupo de tablas afines al que pertenezcan. A continuación se presentan algunos ejemplos:

|  |  |
| --- | --- |
| **Grupos** | **Abreviación en Mayúsculas** |
| Clientes | CLI |
| Usuario | USR |
| Proveedores | PRO |

Los restantes 12 caracteres se utilizarán para especificar el nombre representativo al contenido de la tabla. En este nombre se escribe la primera letra con mayúsculas y las demás en minúsculas, siguiendo la *Nomenclatura para Identificadores[[2]](#footnote-2).*

Sintaxis:

[ID\_Grupo][Nombre\_Representativo]

Para mayor comprensión se presentan los siguientes ejemplos:

|  |  |
| --- | --- |
| CLIDireccion | CLI |
| USRDireccion | USR |
| PRVDireccion | PRO |

**Campos**

El nombre de los campos constará de 15 caracteres como máximo, éste debe ser representativo de la información. El primer caracter indicará el *Tipo de Dato[[3]](#footnote-3)* que almacena el campo y se escribirá con minúsculas, después la primera letra de cada palabra o abreviación que se utilice se escribirá con mayúsculas y minúsculas. El nombre representativo se debe definir siguiendo la *Nomenclatura para Identificadores[[4]](#footnote-4)*.

Sintaxis:

[Tipo\_de\_Dato][Nombre\_Representativo]

A continuación se presenta algunos ejemplos:

|  |  |
| --- | --- |
| iCveProyecto | cRazonSocial |
| cDscProyecto | iNumContrato |
| cRFC | iCveTipoProyecto |

**Indices**

Dentro del diseño y desarrollo del Sistema se utilizan cuatro tipos de índices:

* Indice Primario
* Indice Alterno único
* Indice Alterno No único
* Indice Foráneo

Todos los tipos de índices pueden contener uno o más campos.

El nombre de los índices se formarán de la siguiente manera:

| **Tipo de Indice** | **Sintaxis** |
| --- | --- |
| Indice Primario | [PK][Nombre\_de\_la\_Tabla] |
| Indice Alterno | [AK][Nombre\_Representativo] |
| Indice Alterno No Único | [EK][Nombre\_Representativo] |
| Indice Foráneo | [FK][Nombre\_de\_la\_Tabla\_Foránea] |

Para definir el nombre representativo se deberá seguir la *Nomenclatura para Identificadores[[5]](#footnote-5)*.

1. Ver Anexo A. Nomenclatura para Identificadores [↑](#footnote-ref-1)
2. Ver Anexo A. Nomenclatura para Identificadores [↑](#footnote-ref-2)
3. Ver Anexo E. Tipos de Datos SQL [↑](#footnote-ref-3)
4. Ver Anexo A. Nomenclatura para Identificadores [↑](#footnote-ref-4)
5. Ver Anexo A. Nomenclatura para Identificadores [↑](#footnote-ref-5)