

## **B. ANEXO B: GUÍA DE PROGRAMACIÓN**

### **B.1. Estándares de Programación**

Un aspecto muy importante al emprender un proyecto informático es la estandarización de normas y políticas que permitan la mejor comprensión de los documentos, código de programación, implementación de la base de datos y demás recursos inmersos, entre las personas relacionadas en el desarrollo.

Se ha comprobado que las personas encargadas del mantenimiento de la aplicación pasan la mitad del tiempo tratando de interpretar bloques de código implementado por terceras erróneamente escritas, es por esta razón que este documento pretende dar los lineamientos necesarios que permitan un mejor entendimiento de la codificación implementada en el diseño y desarrollo del Sistema de Gestión de Requerimientos, Ingreso y Procesamiento de Solicitudes de Compras.

#### **B.1.1. Propósito**

El presente documento tiene como finalidad dar a conocer a los interesados los estándares de programación que regirán el desarrollo y mantenimiento de la aplicación que se desea implementar, el mismo que servirá de base para las aplicaciones futuras.

#### **B.1.2. Descripción**

El presente documento presenta al interesado las reglas y políticas que permita estandarizar normar el desarrollo del Proyecto “Sistema de Gestión de Requerimientos, Ingreso y Procesamiento de Solicitudes de Compra” utilizando la plataforma Oracle<sup>®</sup> 10g como servidor de base de datos, OAS(Oracle<sup>®</sup> Application Server 10.1.2) como servidor de aplicaciones y como IDE de programación Oracle<sup>®</sup> Developer Suite 10.1.2 con lenguaje de programación PL/SQL.

Para una mejor comprensión del presente documento, la persona interesada deberá tener los conocimientos necesarios sobre las siguientes tecnologías:

- Bases de Datos
- Conocimientos sobre la tecnología Oracle<sup>®</sup>
- Lenguaje de programación PL/SQL(4gl de Oracle<sup>®</sup>)

## B.2. Estandarización del Diseño de la Base de Datos

En las siguientes líneas se desea implementar las pautas que normalicen el diseño e implementación del repositorio de información o base de datos.

### B.2.1. Objetos de una base de datos en Oracle<sup>®</sup> 10g

Los siguientes son tipos de objetos que se pueden definir en una base de datos Oracle<sup>®</sup> 10g

Objeto	Prefijo
Tabla	TAB
Trigger	TRG
Store procedure(Procedimientos almacenados)	PRO
Views(Vistas)	VIEW
Sequence(Secuencias)	SEQ
Synonym(Sinónimos)	SYN
Function(Funciones)	FUN
Index	IDX
Foreign key(claves foráneas)	FK
Primary key(Claves primarias)	PK
Unique(Claves Unicas)	UK
Constraints Not Null	NN
Paquete	PKG
Constraints Check	CK

**Tabla B.1** Tabla de Tipos de objetos de la BDD

### B.2.2. Áreas de Desarrollo

Las áreas de desarrollo de la Universidad Técnica del Norte están clasificadas de la siguiente forma:

Área	Prefijo
Académico	ACA
Estadístico	EST
Recaudación	REC
Tesorería	TES
Contabilidad	CON
Presupuesto	PRE
Recursos Humanos	RHU
Nómina	NOM

Activos Fijos	ACF
Inventarios	INV
Adquisiciones	ADQ
Puntos de Venta	PVE
Bienestar Socioeconómico	BSE
Bienestar Salud	BIS
Bienestar Odontológico	BIO
Bienestar Laboratorio Clínico	BIC
Seguridad	SEG
Auditoria	AUD
Uso Común(Globales)	UCO

**Tabla B.2** Tabla de Áreas de Desarrollo

### B.2.3. Nombres de los Objetos de la base de datos

Una vez definidos los tipos de objetos que maneja Oracle<sup>®</sup> y las áreas de desarrollo de módulos en la Institución, se sugiere crear los objetos de la base de datos utilizando la siguiente nomenclatura:

El nombre del objeto empezará por el prefijo del área de desarrollo, seguido por el prefijo del tipo de objeto, seguido del símbolo (\_) más el nombre del objeto que represente a la entidad.

A continuación se presenta algunos ejemplos de nombres de objetos:

Área de Desarrollo	Tipo de Objeto	Nombre del Objeto
Presupuesto	Tabla	PRE_TAB_PARTIDAS_PRESUP
Uso Común	Tabla	INS_TAB_PERSONA
Presupuesto	Procedure	PRE_PRO_PRIVI_PRE
Uso Común	Primary Key	UCO_PK_CODIGO

**Tabla B.3** Tabla de Nombres de Objetos

Si los nombres de los objetos son compuestos, como por ejemplo en la definición de nombres de claves foráneas, que relacionan 2 tablas, se intercalará el símbolo \_ entre cada nombre de la tabla, por ejemplo:

Área de Desarrollo	Tipo de Objeto	Nombre del Objeto
Académico	Foreign Key	ACA_FK_ESTUDIANTE_MATRICULA
Uso Común	Foreign Key	UCO_FK_PERSONA
Presupuesto	Foreign Key	EJPRE_EGRE_ITEM_CC_FK

**Tabla B.4** Tabla de Ejemplos de definición de objetos

### B.3. Estandarización del Lenguaje de Programación

Las políticas y normas que rigen la codificación de los procesos en el lenguaje de programación elegido es la siguiente:

#### B.3.1. Nombres de objetos

Para la declaración de variables de objetos como formularios, cuadros de texto, checkbox y otros se seguirá la siguiente nomenclatura:

Los nombres de los objetos empezarán con el prefijo de 3 caracteres que corresponde al prefijo relacionado con el tipo de objeto, seguido del nombre que se quiere asignar a la variable. Por ejemplo: CMD\_ACEPTAR

En el caso de que el nombre de la variable conste de 2 o más palabras el nombre del objeto empezará con un prefijo de 3 caracteres correspondiente al prefijo del objeto, seguido por el carácter \_ mas el nombre del que se desea asignar al objeto, en cada palabra se intercalara el carácter \_ para una mejor comprensión. Por ejemplo: CMD\_CALCULOS\_PORCENTAJES.

A continuación se listan los prefijos de objetos más comunes que se utilizan en algunos de los lenguajes de programación más utilizados.

Objeto	Prefijo	Ejemplo
Alertas	ALE	ALE_AVISO
Botones(CommandButton)	CMD	CMD_ACEPTAR
Checkbox	CHK	CHK_GENERO
Cuadros combinados(ComboBox)	LOV	LOV_ESTUDIANTES
Formulario(Form)	FRM	FRM_PRINCIPAL
Lista de Valores	LOV	LOV_ESTUDIANTES
Cuadros de Texto(TextBox)	TXT	TXT_NOMBRE
Etiquetas(Label)	LBL	LBL_NOMBRE
Botones de Radio(RadioButton)	RDB	RDB_TIPO_FACTURA
Windows	WIN	WIN_CLIENTES
Canvas o lienzos	CAN	CAN_CLIENTES
Grupos de Registros	GRE	GRE_CLIENTES

Tabla B.5 Tabla de Prefijos de Objetos

#### B.3.2. Nombres de variables de tipos de datos

Para el uso de tipos de datos primitivos se sugiere utilizar la siguiente nomenclatura:

El nombre de la variable de tipo primitivo empezara señalando el ámbito de acción: l(local), g(global), p(parámetro), luego un prefijo de 3 dígitos que identifique el tipo de dato que va a almacenar la variable seguido con el nombre de la variable cuya primera letra empieza con mayúscula, si el nombre de la variable está compuesta por 2 o más palabras se intercalara la letra mayúscula de cada palabra, no existirá espacios en blanco, guiones ni subrayados en la unión de las palabras: Ejemplos: intValor, intValorCoordenada, douValorIva

En el siguiente cuadro se muestra el prefijo asociado al tipo de datos que se desea utilizar:

Tipo de dato	Descripción	Prefijo	Ejemplo
Binary_Integer	Antigua versión del PLS_INTEGER. Almacena valores en un rango de -2147483647 .. 2147483647	bint	lbintSumaTotal, gbintSuma
Bfile	Almacena datos binarios no estructurados en archivos del sistema operativo, fuera de la base de datos. Una columna BFILE almacena un localizador del archivo a uno externo que contiene los datos. Admiten hasta 4Gigabytes.  El administrador de la base de datos debe asegurarse de que exista el archivo en disco y de que los procesos de Oracle <sup>®</sup> tengan permisos de lectura para el archivo.	bfil	gbfilVariable, gfilVariPrueba
Blob	Permite almacenar datos binarios no estructurados. Admiten hasta 4 Gigabytes.	blo	pbloImagen, gbloImagen
Boolean	Permite almacenar 2 valores lógicos, TRUE O FALSE.	boo	gbooRespuesta, pbooRespuesta
Char	Cadena de caracteres (alfanuméricos) de longitud fija. Mínimo 1 y Máximo 32767.	ch	lchSexo, pchSexo
Clob	Almacena datos de tipo carácter. Admiten hasta 4Gigabytes.	clo	gcloArchivo, lcloArchivo
Date	Almacena un punto en el tiempo (fecha y hora). El tipo de datos DATE almacena el año (incluyendo el siglo), el mes, el día, las horas, los minutos y los segundos(después de medianoche). Oracle <sup>®</sup> utiliza su propio formato interno para almacenar fechas. Los tipos de datos DATE se almacenan en campos de longitud fija de siete octetos cada uno, correspondiendo al siglo, año, mes, día, hora, minuto, y al segundo. Para entrada/salida de fechas, Oracle <sup>®</sup> utiliza por defecto el formato DD-MMM-AA. Para	dat	pdatFechaActual, gdatFechaActual

	cambiar este formato de fecha por defecto se utiliza el parámetro NLS_DATE_FORMAT. Para insertar fechas que no estén en el mismo formato de fecha estándar de Oracle ®, se puede utilizar la función TO_DATE con una máscara del formato: TO_DATE(el "13 de noviembre de 1992", "DD del MES, YYYY").		
Dec	Permite declarar números de punto fijo. Máximo precisión de 38 dígitos decimales.	dec	ldecTotal, gdecTotal
Decimal	Permite declarar números de punto fijo. Máximo precisión de 38 dígitos decimales.	deci	gdeciVariable, gdeciVarPrueba
Double precision	Permite declarar números de punto flotante. Máxima precisión de 126 dígitos binarios, de los cuales 38 son dígitos decimales	doup	ldoupCantidad gdoupCantTemporal
Int	Permite declarar enteros. Precisión máxima de 38 dígitos decimales	int	lintSuma, gintSumaTotal.
Integer	Permite declarar enteros. Precisión máxima de 38 dígitos decimales.	intg	lintgVariable, pintgSumaValores
Interval year to month	Permite manipular intervalos de tiempo de años y meses.	intv	pintvFechaCompra, gintvFechaRegistro
Lob	Permiten almacenar y manipular bloques grandes de datos no estructurados (tales como texto, imágenes, videos, sonidos, etc) en formato binario o del carácter. Admiten hasta 4 Gigabytes. Una tabla puede contener varias columnas de tipo LOB. Soportan acceso aleatorio. Las tablas con columnas de tipo LOB no pueden ser replicadas.	lob	plobVideoAcademico globTextoUtn
Long	Cadena de caracteres de longitud variable. Como máximo admite hasta 2 GB(2000 MB). Los datos LONG deberán ser convertidos apropiadamente al moverse entre diversos sistemas. Este tipo de datos está obsoleto (en desuso), en su lugar se utilizan los datos de tipo LOB(CLOB, NCLOB). Oracle ® recomienda que se convierta el tipo de datos LONG a alguno LOB si aún se está utilizando. No se puede utilizar en cláusulas WHERE, GROUP BY, ORDER BY, CONNECT BY ni DISTINCT. Una tabla sólo puede contener una columna de tipo LONG. Sólo soporta acceso secuencial.	lon	glonValor, plonTemporal
Long raw	Almacenan cadenas binarias de ancho variable. Hasta 2	lonr	llonrVariableDePru,

	GB. En desuso, se sustituye por los tipos LOB		plonrVariableTemp
Float	Almacena tipos de datos numéricos en punto flotante. Es un tipo NUMBER que sólo almacena números en punto flotante.	flo	gfloValorIce, pfloIva
Natural	Permite restringir una variable entera a valores solo positivos	nat	gnatValor, pnatValorTotal
Nchar	Cadena de caracteres de longitud fija que sólo almacena caracteres Unicode.	nch	gnchCadena, InchCadenaTemporal
Nclob	Almacena datos de tipo carácter. Admiten hasta 4Gigabytes. Guarda los datos según el juego de caracteres Unicode nacional.	nclo	gncloCadena, pncloCadenaTotal
Naturaln	Permite restringir una variable entera a valores solo positivos	natn	pnatnValores, InatnSumaTotal
Numeric	Permite declarar variables de punto fijo. Máximo precisión de 38 dígitos decimales	num	pnumValor, InumSumaTotal
Number	Almacena números fijos y en punto flotantes. Se admiten hasta 38 dígitos de precisión y son portables a cualquier entre los diversos sistemas en que funcione Oracle <sup>®</sup> . Para declarar un tipo de datos NUMBER en un CREATE o UPDATE es suficiente con: nombre_columna NUMBER opcionalmente se le puede indicar la precisión (número total de dígitos) y la escala(número de dígitos a la derecha de la coma, decimales, los cogerá de la precisión indicada): nombre_columna NUMBER(precision, escala). Si no se indica la precisión se tomará en función del número a guardar, si no se indica la escala se tomará escala cero. Para no indicar la precisión y sí la escala podemos utilizar: nombre_columna NUMBER (*, escala) Para introducir números que no estén en el formato estándar de Oracle <sup>®</sup> se puede utilizar la función TO_NUMBER.	numb	pnumbSumaTotales, InumbSuma
Nvarchar2	Cadena de caracteres de longitud variable que sólo almacena caracteres Unicode. Mínimo 1 y máximo 32767	nvar	pnvarCadena, gnvarCadenaDatos
Pls_integer	Tiene la misma funcionalidad que el number, pero ocupa menos espacio y mejor desempeño. El rango de magnitud está entre -2147483647 y 2147483647	pls	pplsValorPrueba, gpplsValorTotal
Positive	Permite restringir una variable entera a valores solo	pos	pposTemporal,

	positivos		gposSumaTemporal
Positiven	Permite restringir una variable entera a valores solo positivos, que no admiten valores nulos.	posn	lposnCoordenada, gposnSumaTotal
Raw	Almacenan cadenas binarias de ancho variable. Hasta 32767 bytes. En desuso, se sustituye por los tipos LOB.	raw	prawValorTemporal, grawPrueba
Real	Almacena valores con punto flotante. Almacena en un rango de 63 dígitos binarios, 18 para dígitos decimales	rea	preaSumalva, greaSumaEstudiantes
Row		row	provarPrueba, growTemporal
Rowid	<p>Almacenar la dirección única de cada fila de la tabla de la base de datos. ROWID físico almacena la dirección de fila en las tablas, las tablas en clúster, los índices, excepto en los índices-organizados(IOT). ROWID lógico almacena la dirección de fila en tablas de índice-organizado (IOT). Un ejemplo del valor de un campo ROWID podría ser: "AAAlugAAJAAC4AhAAI". El formato es el siguiente: Para "OOOOOFFFFBBBBBRRR", donde:</p> <p>OOOOO: segmento de la base de datos (AAAlug en el ejemplo). Todos los objetos que estén en el mismo esquema y en el mismo segmento tendrán el mismo valor.</p> <p>FFF: el número de fichero del tablespace relativo que contiene la fila(fichero AAJ en el ejemplo).</p> <p>BBBBB: el bloque de datos que contiene a la fila (bloque AAC4Ah en el ejemplo). El número de bloque es relativo a su fichero de datos, no al tablespace. Por lo tanto, dos filas con números de bloque iguales podrían residir en diferentes datafiles del mismo tablespace.</p> <p>RRR: el número de fila en el bloque (fila AAJ en el ejemplo). Este tipo de campo no aparece en los SELECT ni se puede modificar en los UPDATE, ni en los INSERT. Tampoco se puede utilizar en los CREATE. Es un tipo de datos utilizado exclusivamente por Oracle<sup>®</sup>. Sólo se puede ver su valor utilizando la palabra reservada ROWID, por ejemplo: select rowid, nombre, apellidos from clientes</p> <p>Ejemplo 2: SELECT ROWID, SUBSTR(ROWID,15,4) "Fichero", SUBSTR(ROWID,1,8) "Bloque",</p>	rowid	prowidVariTemporal, growidPrueba



	SUBSTR(ROWID,10,4) "Fila" FROM proveedores Ejemplo 3: una forma de saber en cuántos ficheros de datos está alojada una tabla: SELECT COUNT(DISTINCT(SUBSTR(ROWID,7,3))) "Numero ficheros " FROM facturacion		
Signtype	Permite restringir los valores de una variable a uno de los estados -1,0 y 1. Restringe una variable a uno de los 3 estados -1,0 y 1	sgt	psgtValCoordenada, lsgtEstado
Smallint	Máxima precisión de 38 dígitos decimales	smal	gsmalValorGlobal, psmalTemporal
Timestamp	Almacena toda una fecha completa, año, mes, día, hora, minuto, segundo. Los valores aceptados en fracción de segundos van desde 0 a 9, por defecto es 6	tims	ltimsFecNacimiento, gtimsFechaActual
Timestamp with Timezone	Almacena datos de tipo hora incluyendo la zona horaria (explícita), fraccionando los segundos. Los valores aceptados en fracción de segundos van desde 0 a 9, por defecto es 6.	timswt	ptimswtVariable, ltimswtTemporal
Timestamp with Local timezone	Almacena datos de tipo hora incluyendo la zona horaria local (relativa), fraccionando los segundos. Cuando se usa un SELECT para mostrar los datos de este tipo, el valor de la hora será ajustado a la zona horaria de la sesión actual	timswlt	ptimswltFecha, gtimswltFecTemporal
Urowid	ROWID universal. Admite ROWID a tablas que no sean de Oracle®, tablas externas. Admite tanto ROWID lógicos como físicos, tiene un tamaño máximo de 4000 bytes	urow	purowVariable, gurowVariTemporal
Varchar	Cadena de caracteres de longitud variable. Mínimo 1 y máximo 32767	varc	pvarcCadena, lvarcCadenaTemp
Varchar2	Cadena de caracteres de longitud variable. Mínimo 1 y máximo 32767	varc2	gvarc2Nombre, pvarc2ApeEstudiante
XMLType	Tipo de datos abstracto. En realidad se trata de un CLOB. Se asocia a un esquema XML para la definición de su estructura.	xml	pxmlEstructura, lxmlEstuctDocument

Tabla B.6 Tabla de Variables y tipos de datos

### B.3.3. Nombres de constantes

Todo nombre de constante empezará por el prefijo siguiente: C\_, seguido del tipo de dato que representa más el nombre de la constante. Por ejemplo: C\_floPorcentajelva, C\_douPi

### B.3.4. Nombres de Funciones y procedimientos

Todos los nombres de funciones y procedimientos estarán escritos en MAYUSCULAS, si el nombre de la función o procedimiento está compuesto de 2 o más palabras se lo hará intercalando el signo \_ entre cada palabra, por ejemplo:

- FUN\_CALCULOS\_EDAD\_ESTUDIANTES
- PRO\_PROCEDIMIENTO
- PKG\_PAQUETE.FUN\_CALCULO

### B.3.5. Documentación y comentarios en el código.

Todo bloque de código tendrá como encabezado las siguientes líneas:

```
/* Creado por: Egrd. Andrea Guevara  
Fecha de creación: 03/05/2010  
Última modificación: 17/07/2011  
Descripción del bloque: Una breve descripción sobre el bloque de código Siguiente.  
Descripción de Variables: Una breve descripción de las variables utilizadas y su utilización en el  
bloque de código */
```

### B.3.6. Palabras reservadas del lenguaje de programación

Todas las palabras reservadas que forman parte del lenguaje serán escritas en MAYÚSCULAS. Ejemplo:

```
/* Creado por: Egrd. Andrea Guevara  
Fecha de creación: 08/11/2010  
Última modificación: 21/12/2010  
Descripción del bloque: Este bloque permite sumar los subtotales del campo total.  
Descripción de Variables:  
InumPrueba Variable que almacena la suma de los totales */  
PACKAGE BODY CALCULOS IS  
PROCEDURE SUBTOTAL IS  
    InumPrueba NUMERIC;  
BEGIN InumPrueba:=454545;  
END;  
END;
```