



UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN

ESCUELA DE INGENIERÍA DE SISTEMAS

**“Sistema de Archivo Virtual para el Departamento de Recursos
Humanos de la Universidad del Azuay”**

Tesis previa a la obtención del título de
Ingeniera de Sistemas

AUTORA: MARÍA GABRIELA AGUIRRE VICUÑA

DIRECTOR: ING. JAIME VELEZ

CUENCA, ECUADOR

2007

AUTORÍA

El autor es el único responsable de los conceptos, conclusiones y observaciones emitidas en la presente TESIS.

María Gabriela Aguirre Vicuña

AGRADECIMIENTO

Esencialmente a mi madre y esposo, por haberse esforzado durante todos estos años, para hacer posible la culminación de mis estudios.

A todos mis profesores que fueron siempre la guía y el apoyo a lo largo de la carrera, como de la realización de esta tesis.

DEDICATORIA

Este trabajo está dedicado a mi madre y abuela, quienes fueron las que siempre me apoyaron para llegar a alcanzar la culminación de mi carrera.

A mi esposo, que con su ayuda y conocimientos me supo siempre apoyar durante mis estudios universitarios, y fue un pilar fundamental para llegar a esta etapa.

A mis compañeros, que trabajamos juntos y siempre estuvieron a mi lado y me supieron brindar su apoyo en el momento necesario.

INDICE DE CONTENIDOS

AUTORÍA.....	II
AGRADECIMIENTO	III
DEDICATORIA.....	IV
INDICE DE CONTENIDOS.....	1
RESUMEN.....	2
ABSTRACT.....	3
INTRODUCCIÓN.....	4
CAPITULO I. UNIVERSIDAD DEL AZUAY.....	5
1.1 HISTORIA.....	5
1.2 MISIÓN, VISIÓN, PRINCIPIOS, VALORES.....	6
1.3 ESTRUCTURA ORGÁNICA FUNCIONAL.....	7
1.4 ORGANIGRAMA GENERAL.....	8
CAPITULO II. ARCHIVO VIRTUAL.....	9
2.1 ACTUAL MANEJO DE ARCHIVO.....	9
2.2 DESCRIPCIÓN DE ARCHIVO DE ALMACENAMIENTO MAGNÉTICO.....	9
2.3 TIPOS DE ARCHIVOS DE DATOS (.DOC, .TXT, .XLS, .DAT, .RTF).....	9
2.4 PROGRAMAS ASOCIADOS PARA LA INTERPRETACIÓN DE ARCHIVOS DE DATOS (WORD, EXCEL, WORDPAD).....	10
2.5 TIPOS DE ARCHIVOS GRÁFICOS (.JPG, .JPEG, .GIF, .BMP).....	11
2.6 PROGRAMAS ASOCIADOS PARA LA INTERPRETACIÓN DE ARCHIVOS GRÁFICOS (INTERNET EXPLORER).....	13
CAPITULO III. DESARROLLO DE LA APLICACIÓN.....	14
3.1 HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO.....	14
3.1.1 Oracle 8.0.....	14
3.1.2 Forms 6.0.....	19
3.1.2 Report 6.0.....	26
3.1.4 PL/SQL Developer 4.0.....	27
3.2 ANÁLISIS.....	36
3.2.1 Análisis de Necesidades.....	36
3.2.2 Análisis de la Viabilidad.....	36
3.2.3 Análisis de Requisitos.....	36
3.2.4 Análisis Estructurado.....	46
3.3 DISEÑO.....	69
3.4 PROGRAMACIÓN.....	71
3.5 PRUEBAS.....	71
CONCLUSIONES.....	72
RECOMENDACIONES.....	73
BIBLIOGRAFÍA.....	74
ANEXOS.....	75
MANUAL DEL USUARIO.....	75

RESUMEN

El objetivo de esta tesis es modelar, diseñar y desarrollar un Sistema de Archivo Virtual para el Departamento de Recursos Humanos de la Universidad del Azuay, de manera que toda información que se maneje quede totalmente automatizada, y que agilite la consulta de datos de los empleados, así como, controlar la variación de estados de los empleados, creando una ficha con todos los datos de cada uno de los empleados y su relación laboral con la Universidad del Azuay, datos como: fecha de Ingreso; fecha de Salida; asensos; multas; felicitaciones; títulos; etcétera.

El Sistema de Archivo Virtual está desarrollado en la base de datos *Oracle* y cuenta con una interfaz gráfica y diferentes reportes que son solicitados por parte del departamento o al departamento, que sirven como una ayuda general.

ABSTRACT

The objective of this thesis is to model, design, and develop a Virtual File System for the University of Azuay's Human Resources Department in order to completely automate the handling of information. The idea is to expedite consultation of employees' data and to control the variation of their status, creating a file with all the information about each employee and his or her work relationship with the University of Azuay, including data such as arrival and departure dates, promotions, fines, congratulations, degrees, etc.

The Virtual File System has been developed on the Oracle database and counts on a graphic interface and different reports that are requested by the department or for the department and give general help.

INTRODUCCIÓN

Esta tesis proporciona una fuente de información del personal que labora para la Universidad del Azuay, así como, sus entornos laborales, históricos, reportes y datos en general de uso exclusivo para el Departamento de Recursos Humanos.

Está diseñada bajo tres capítulos:

Capítulo I “Universidad del Azuay”, proporciona una información sobre la Universidad del Azuay en general, como su historia, misión, visión, y un esquema de jerarquización.

Capítulo II “Archivo Virtual”, muestra los diferentes tipos de archivos de datos que se manejan en el sistema y los programas asociados para la interpretación de los mismos.

Capítulo III “Desarrollo de la Aplicación”, contiene la parte esencial del Sistema de Archivo Virtual, desde la descripción y forma de uso de las herramientas utilizadas para el desarrollo, así como el análisis, diseño, y la implementación del mismo.

Capítulo I. Universidad del Azuay

1.1 Historia.

La Universidad del Azuay nació en 1968 y tiene su sede en la ciudad de Cuenca, capital de la provincia del Azuay. Es una Universidad, particular y católica, creada por el Gobierno Ecuatoriano al amparo del Modus Vivendi y de acuerdo con la Ley de Universidades y Escuela Politécnicas. Inició sus actividades como una sede de la Pontificia Universidad Católica del Ecuador; y desde el año 1990 funciona como una Universidad autónoma. ¹

En el año 2006 se constituyó en la primera universidad ecuatoriana en lograr la Acreditación por parte del Consejo Nacional de Evaluación y Acreditación, CONEA. El nombre de la Universidad es un nombre que evoca una vinculación telúrica, que hace referencia a su propia geografía que tiene una fecunda historia. ¹

Lo utilizó Fray Vicente Solano para nombrar el primer periódico cuencano editado en 1828. La Universidad del Azuay trata de ser como lo traduce del cañari los investigadores: licor o fermento, levadura o estímulo para despertar el espíritu, para dinamizar la acción, para avivar el fuego de la creación. ¹

Como Universidad pretendemos ofrecer carreras que respondan a las necesidades de la región y del país dentro de una concepción integral del ser humano, de tal manera que a la sólida preparación profesional se una el compromiso de servir a la sociedad, especialmente a los sectores más necesitados, buscando su superación. ¹

Desarrollar acciones que proyecten a la comunidad su acervo cultural, científico, tecnológico y ético. ¹

¹ www.uazuay.edu.ec

Reseña Histórica.htm

Actualizar permanentemente los conocimientos provenientes del desarrollo científico cultural y revertirlos a la comunidad universitaria, a los ex alumnos y a la sociedad.¹
Poner en práctica sus principios respetando y defendiendo la multiplicidad cultural del Ecuador y su patrimonio histórico, natural y ecológico. ¹

Contribuir al desarrollo de la ciencia, la cultura, la tecnología, las artes y las letras, a través de la docencia, la investigación y las acciones más idóneas dentro y fuera de la comunidad universitaria. ¹

Propender a la integración y cooperación interinstitucionales cuando los fines sean coincidentes o complementarios. Todo esto equilibrado en los campos técnico y humanístico. ¹

1.2 Misión, Visión, Principios, Valores.

Misión

Formar personas con sólidos valores y conocimientos; y responder a las necesidades de la sociedad, mediante la variada, oportuna y permanente renovación de su oferta académica. ²

Visión

La Universidad del Azuay se propone ser una institución con calidad académica y humanística, que aporte al conocimiento y promueva el desarrollo integral de la persona y su entorno. ²

Principios

- ◆ Excelencia Académica
- ◆ Trabajo por una sociedad justa guiada por los principio cristianos.
- ◆ Pluralismo ideológico y ejercicio de la razón para su desenvolvimiento institucional.

www.uazuay.edu.ec

¹ Reseña Histórica.htm

² Misión, Visión, Principios.htm

- ◆ Búsqueda de la verdad con absoluta libertad y sin prejuicios tanto para la docencia como en la investigación.
- ◆ Apertura a todas las corrientes del pensamiento, que serán expuestas y estudiadas de manera rigurosamente científica.
- ◆ No se privilegiará ni perjudicará a nadie por su ideología. ²

Valores

- ◆ Honestidad
- ◆ Responsabilidad
- ◆ Trabajo en equipo
- ◆ Actitud de servicio
- ◆ Innovación
- ◆ Desarrollo continuo
- ◆ Liderazgo
- ◆ Respeto a la naturaleza
- ◆ Vinculación con la Comunidad
- ◆ Uso adecuado de los bienes universitarios

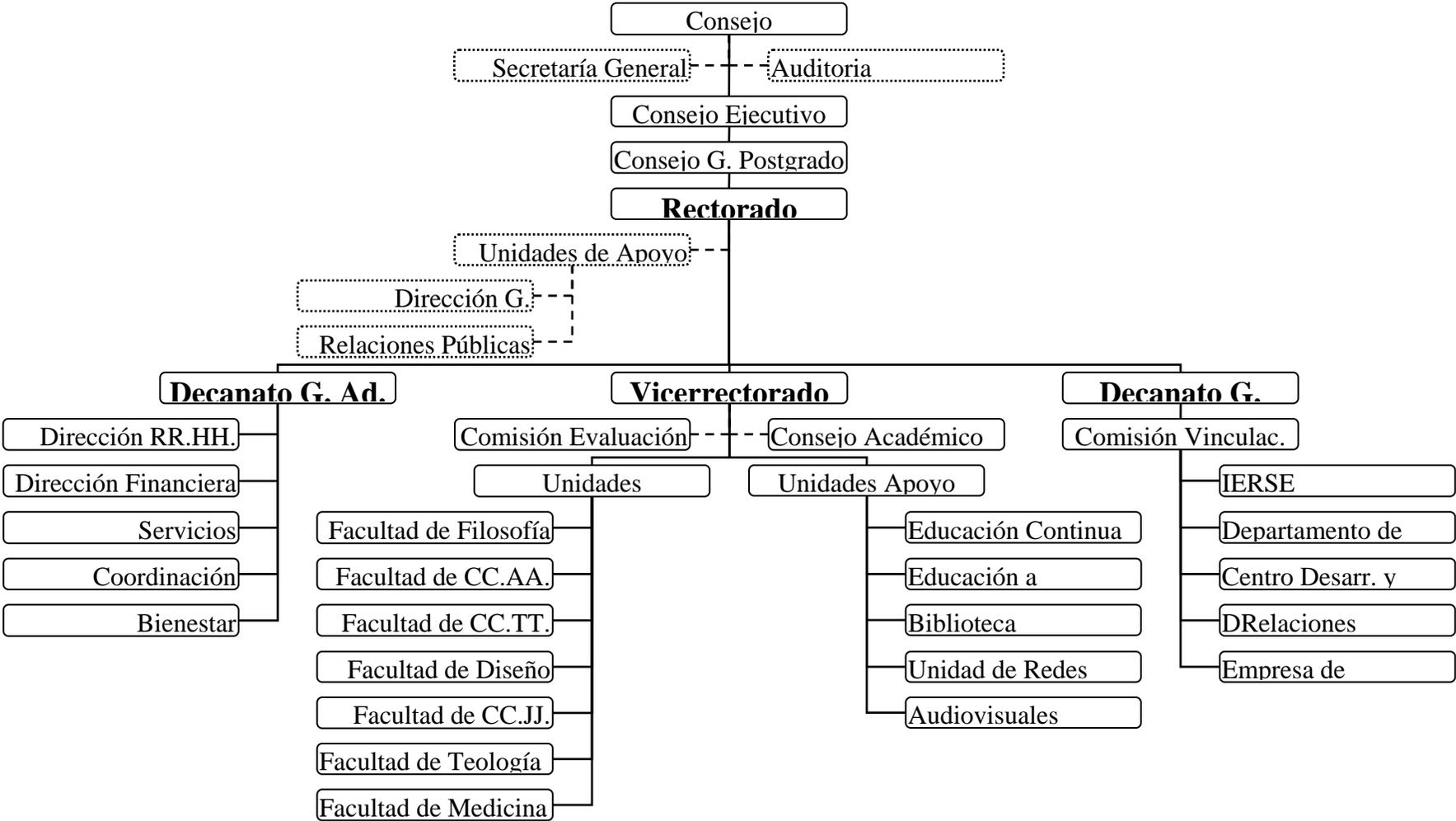
1.3 Estructura orgánica funcional.

Toda empresa cuenta en forma implícita o explícita con cierto juego de jerarquías y atribuciones asignadas a los miembros o componentes de la misma. En consecuencia se puede establecer que la estructura organizativa de una empresa es el esquema de jerarquización y división de las funciones componentes de ella. Jerarquizar es establecer líneas de autoridad (de arriba hacia abajo) a través de los diversos niveles y delimitar la responsabilidad de cada empleado ante solo un superviso inmediato. Esto permite ubicar a las unidades administrativas en relación con las que le son subordinadas en el proceso de la autoridad.

www.uazuay.edu.ec

² Misión, Visión, Principios.htm

1.4 Organigrama General.



Capitulo II. Archivo Virtual.

2.1 Actual manejo de archivo.

A la fecha el Departamento de Recursos Humanos de la Universidad del Azuay controla y lleva manualmente el archivo y registro de sus empleados, lo cual representa pérdida de tiempo, necesidad de extenso espacio físico y por ende más personal calificado.

2.2 Descripción de Archivo de almacenamiento magnético.

Los archivos son las unidades básicas de almacenamiento que permiten a la computadora distinguir entre los diversos conjuntos de información. Aunque no siempre es el caso, un archivo se suele encontrar en un formato legible por los usuarios. Aun así, en un archivo se agrupan instrucciones, números, palabras o imágenes en unidades coherentes que el usuario puede recuperar, modificar, eliminar, guardar o enviar a un dispositivo de salida (disquete, CD,) o almacenamiento magnético.

Normalmente los archivos están formados por un nombre, un punto y una extensión (PROGRAMA.EXE). El nombre nos sirve para diferenciar unos archivos de otros y la extensión para atribuirle unas propiedades concretas. Estas propiedades asociadas o "tipo de archivo" vienen dadas por las letras que conforman la extensión.

Normalmente su máximo son tres letras aunque existen algunas excepciones (.jpeg, .html, .java, etc.). Cada uno de estos pequeños grupos de caracteres está asociado a un tipo de archivo.

2.3 Tipos de archivos de Datos (.doc, .txt, .xls, .dat, .rtf).

Dentro de los documentos de texto hemos de diferenciar entre el texto plano y el enriquecido. Es decir, entre los formatos que sencillamente guardan las letras (txt, dat, log...) y los que podemos asignarles un tamaño, fuente, color, etc. (doc)

DOC → *Microsoft Word*

.DOC: archivo de texto con estilo (posibilidad de asignarle formato a las letras). Se genera y se abre con los principales procesadores de texto para *Windows*: *MicrosoftWord*, *WordPerfect*, *Display Write*, *WordStar*.³

TXT → *Block de notas / WordPad*

.TXT: archivo de texto plano, sin gráficos, habitual para registros. Se abre con cualquier editor de texto, incluido el Bloc de Notas, de *Windows*.⁴

Pongamos por ejemplo un archivo llamado "DOCUMENTO.TXT", su nombre será DOCUMENTO y su extensión TXT. Esta extensión está asociada con el tipo de archivos que contienen texto, por lo tanto podemos suponer que habrá algo escrito dentro. Nuestro sistema operativo (*Windows* en este caso) tendrá una lista de los programas con los que puede ser utilizado este archivo y si deseamos visualizarlo éste será abierto con el NotePad o Bloc de Notas.

RTF → *Microsoft Word*

.RTF: Formato de archivo de texto enriquecido que permite intercambiar texto entre distintos procesadores de texto y en distintos sistemas operativos. Accesible con cualquier procesador de texto.⁴

XLS → Hoja de cálculo

.XLS: Herramienta de cálculo muy potente que facilita la creación de documentos, conocidos como hojas de cálculo o libros de trabajo. Introduciendo los datos en las celdas y mediante el uso de fórmulas, se pueden realizar cálculos simples y complejos de una manera rápida y automatizada.⁴

2.4 Programas asociados para la interpretación de archivos de datos (Word, Excel, WordPad).

Word: Es un procesador de textos que facilita la creación de documentos de texto en general, de una manera rápida y eficiente. Además ofrece la capacidad de editar,

www.hisp@zone.com

³ Tutoriales Informática – Los archivos tipos, extensiones y programas para su uso.htm

www.monografias.com

⁴ Tipos de extensiones_com.htm

revisar, corregir y formatear un documento para darle una presentación profesional.

Excel: Es una herramienta de cálculo potente y versátil, que facilita la creación de documentos denominados hojas de cálculo y más propiamente conocidos como libros de trabajo, que pueden incluir varias hojas en su interior. Automatiza los cálculos mediante el manejo de fórmulas, de igual manera se pueden graficar los datos para producir documentos claros y profesionales.

WordPad: Es un sencillo programa de procesamiento de texto que viene incluido en la lista de accesorios gratuitos en *Windows*. Un procesador de texto sirve para el tratamiento de textos. Es un sistema basado en el ordenador para escribir y Formatear textos; aporta ventajas de almacenamiento y localización, de copiar bloques de texto búsqueda de palabras.

Estas herramientas forman parte del paquete de *Microsoft Office XP* (versión). Son los programas más usados y conocidos debido a sus excelentes prestaciones y sencillez en el manejo, cada versión presentan características nuevas y potentes.

2.5 Tipos de archivos gráficos (.jpg, .jpeg, .gif, .bmp).

Podemos utilizar alguno que otro formato para visualizar, almacenar o escanear imágenes, como pueden ser BMP, GIF, JPG o JPEG, entre otros. La diferencia radica en que unos comprimen más que otros, o que algunos almacenan de forma más eficiente según el tipo de imagen. Por esta razón, se explicará para qué sirven cada uno de los formatos, y en qué situación utilizarlos. Además se verá cuáles son los más utilizados por los usuarios del PC, y cuáles son cuando se hace referencia a las páginas *Web de Internet*.⁵

JPG → XnView / ACDSSee

JPEG → XnView / ACDSSee

.JPEG: También se le ve como JPG es uno de los más extendidos, por su compresión y calidad, en páginas webs para logotipos y cabeceras.

.JPEG: Es el mejor formato para fotografía o imágenes de tonos continuos. Se trata de un formato abierto, cuyos derechos son libres, y que puede ser usado o implementado en un programa (tanto para reconocerlo como para editarlo o guardar archivos en él) libremente, sin tener que pagar derechos por ello a nadie. ⁵

El formato JPEG nació como una respuesta a las limitaciones de otros formatos, entre ellos el GIF, en cuanto a calidad y tamaño de archivos. JPEG es un formato de compresión con pérdida, esto quiere decir que, al guardar una imagen en este formato, algo de la información que contiene esa imagen se reduce, es decir, está pierde un poco de calidad, aunque, generalmente, esta pérdida de calidad es imperceptible al ojo humano. Con ello se consigue reducir el tamaño del archivo y, por tanto, mejorar la velocidad de bajada de tus páginas web. Dependiendo de la profundidad de color que tenga la imagen cada pixel puede ocupar 1 o varios bytes. ⁵

GIF → XnView / ACDSSee

.GIF: uno de los dos formatos de archivo de gráficos preferido en la *Web* (el otro es .JPG). Comprimido al igual que los .JPG, pero por otro sistema llamado LZW, patentado por Unisys. Se abre con cualquier visor de gráficos. Este formato cuenta con características que lo hacen ideal para el uso en páginas web, como es la posibilidad de darle un fondo transparente o insertarle movimiento. ⁵

El formato GIF, propietario de *CompuServe*, corresponde a las siglas de *Graphics Interchange Format*. Es el formato más utilizado para mostrar gráficos de colores indexados e imágenes en documentos HTML (*hypertext markup language*) sobre *World Wide Web* y otros servicios online. Gif es un formato de imágenes comprimidas diseñado para minimizar el tiempo de transferencia de archivos sobre las líneas telefónicas. ⁵

Se trata de un formato de "Mapa de Bits" (como, por ejemplo, lo son el BMP o el PCX) que significa que, en cada imagen, hay una tabla que indica los colores que se representarán en la imagen, a los que luego se referencia desde la imagen en sí. GIF tiene una profundidad de color máxima de 8 bits, lo que da un tope de 256 colores.

A pesar de esto, GIF posee la capacidad de aparentar más tonos de color usando una técnica de promediación de colores que consiste, básicamente, en obtener un color intermedio que no esté en su tabla a colocando juntos varios pixels de colores que si están en esa tabla. Este formato es preferible para las imágenes de tonos no continuos o cuando hay grandes áreas de un mismo color. ⁵

BMP → *XnView / ACDSee*

.BMP: Extensión que nace del nombre de este formato *BitMaP* o Mapa de Bits, gran calidad pero tamaño excesivo no suele ser muy utilizado en *Internet* por su carga lenta. El formato BMP (*Bit Map*) es el formato de las imágenes en *bitmap* de *Windows*. Aunque muy extendido, tiene la dificultad de la escasa compresión que realiza en los archivos por lo que ocupan rápidamente casi 1Mb. Pero el formato de Mapa de Bits tiene una importante característica a su favor, es que casi todos los usuarios tienen una PC que puede soportarlo.

Se ve con el accesorio de *Microsoft Paint* o con cualquier visor de gráficos. Los archivos de mapas de bits se componen de direcciones asociadas a códigos de color, uno para cada cuadro en una matriz de pixeles. Normalmente, se caracterizan por ser muy poco eficientes en su uso de espacio en disco, pero pueden mostrar un buen nivel de calidad. A diferencia de los gráficos vectoriales, al ser reescalados a un tamaño mayor, pierden calidad. Otra desventaja de los archivos BMP es que no son utilizables en páginas *web* debido a su gran tamaño con relación a su resolución. ⁵

2.6 Programas asociados para la interpretación de archivos gráficos (Internet Explorer).

Internet Explorer es una colección de redes de equipos que conectan a millones de equipos de todo el mundo. *Microsoft Internet Explorer* le permite conectar a *Internet* para tener acceso a vastos almacenes de información en dichos equipos.

Capítulo III. Desarrollo de la Aplicación.

3.1 Herramientas Utilizadas para el desarrollo.

3.1.1 Oracle 8.0.

INTRODUCCION

Oracle es básicamente un herramienta cliente/servidor para la gestión de base de datos, pues bien, para su utilización primero es necesario la instalación de la herramienta servidor (*Oracle9i*) y posteriormente podríamos atacar a la base de datos desde otros equipos con las herramientas de desarrollo como *Oracle Designer* y *Oracle Developer*, que son las herramientas de programación sobre *Oracle*; a partir de esta premisa se desarrolla las principales acepciones de *Oracle* y sus aplicaciones en las distintas áreas de trabajo.

Una base de datos *Oracle* consta de archivos físicos, áreas de memoria y procesos, la distribución de estos componentes varía en función de la arquitectura de la base de datos que se elija.

Es el conjunto de datos que proporciona la capacidad de almacenar y acudir a éstos de forma recurrente con un modelo definido como relacional. Además es una suite de productos que ofrece una gran variedad de herramientas, reportes y utilitarios.

Oracle corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo. Soporta unos 17 idiomas, corre automáticamente en más de 80 arquitecturas de *hardware* y *software* distinto sin tener la necesidad de cambiar una sola línea de código. Esto es porque más el 80% de los códigos internos de *Oracle* son iguales a los establecidos en todas las plataformas de sistemas operativos. ⁶

⁶ www.monografias.com/trabajos25/oracle/oracle.shtml

Reseña Histórica

El manejador de Base de datos *ORACLE*, surgió a final de los años 70 y principio de los años 80. George Koch y su equipo de tropas de asalto de técnicos fue el primero en desembarcar en el terreno de *Oracle* en 1982, durante un proceso de evaluación de sistema de gestión de base de datos para una importante aplicación comercial que George estaba diseñando y construyendo.

Cuando terminó, la evaluación fue descrita en *Computer World* como el estudio más severo de SGBD que se había hecho nunca. *Oracle* conocida entonces como *Relational Software*, tenía poco más de 25 empleados en aquel tiempo y solo unos pocos clientes importantes. Sin embargo, cuando se completó el estudio, *Oracle* fue declarada vencedora. George afirmó que el SGBD *Oracle* era técnicamente el mejor producto del mercado.

La compañía de *Oracle Corporation* estaba trabajando entonces para perfeccionar su joven producto, para comprender los tipos de características y funcionalidad que podría hacerlo útil y productivo en el mundo de los negocios. El esfuerzo contribuyó a su refinamiento. Algunas de las características de *Oracle*, tales como las salidas de *SQL*FORMS* fueron el resultado de aquel esfuerzo. ⁶

Evolución

El poderoso modelo relacional ha evolucionado desde herramientas y los modelos de datos de redes. La mayor aceptación y uso de un modelo de datos es el modelo relacional que fue conocido en 1969 con la revisión hecha por IBM.

Un modelo relacional posee tres aspectos: ⁶

- Estructuras: Definición de objetos que contengan datos y que son accesibles a los usuarios.
- Operaciones: Definir acciones que manipulen datos u objetos.
- Reglas: Leyes para gobernar la información, cómo y qué manipular.

Una tabla está compuesta por una matriz bidimensional de filas y columnas.

⁶ www.monografias.com/trabajos25/oracle/oracle.shtml

En cualquier ocasión la información es cambiada en una base de datos relacional, cualquier información es el resultado de una consulta presentada por el usuario en el formato de filas/columnas. ⁶

Estructura

Oracle soporta dos tipos de almacenamiento, por carácter(*RAW*) o por bloques (*Files System*), generalmente es recomendable que sean colocados en *Raw Device*.

Raw Device: es un dispositivo de caracteres disponibles en algunos sistemas operativos el cual es asignado directamente a *Oracle*. ⁶

Oracle corre más rápidamente con *Raw Device* que con *Files System*, por varias razones:

1. I/O (*Input/Output*) es realizado directamente en el disco por *Oracle*, independientemente del sistema operativo.
2. El *buffer* cache del sistema del sistema operativo es dejado a un lado.
3. Los *buffers* del sistema operativo y de *Oracle* son independiente entre sí.

Con la intención de evitar la contención de los discos, se debe considerar la instalación de *Oracle* en dispositivos separados, especialmente si se tienen varios discos, y más esencialmente, si se poseen más de una controladora de disco. La planeación debe realizarse teniendo en cuenta los siguientes criterios:

- Los *Files System* y sus dispositivos asignados.
- El *swapping* y paginamiento en *Oracle*, deberán estar en los dispositivos más rápidos.
- Los *tablespace* para tablas e índices en dispositivos separados.
- Los *Log Files* en un dispositivo separado al del *tablespace* de RDBMS *Oracle*. ⁶

Estructura Física y Lógica

Las estructuras físicas tales como los archivos del sistema operativo, son almacenados en dispositivos tangibles como son cintas magnéticas, discos y otros. A cada archivo le corresponde un espacio en el sistema operativo.

⁶ www.monografias.com/trabajos25/oracle/oracle.shtml

Oracle requiere de varios archivos para su funcionamiento, los cuales conforman su estructura física. A la estructura lógica le corresponde un espacio por unidad, pero sus limitaciones son independientes de las localizaciones de espacio físico. ⁶

PROGRAMAS Y ARCHIVOS QUE COMPONE ORACLE

El PGA (Programa Global Área)

Es también llamado Área Global del Programa, consta de datos e información de control de los procesos, asegurando el uso correcto de éstos. El PGA contiene información acerca de las conexiones y los procesos que se realizan en *Oracle*, su tamaño es variable en longitud, pero no es dinámico. El PGA se activa al conectarse un usuario; es un área de memoria utilizada por un único proceso de usuario de *Oracle*. La memoria de la PGA no se comparte. ⁶

El SGA (Sistema Área Global)

Se puede llamar *Shared* global área. Se podría definir como una serie de *buffers* en memoria residente, a través de la cual todas las transacciones y el almacenamiento de datos fluyen (facilita la transferencia de información entre usuarios). El SGA es localizado en memoria al iniciarse una instancia y desaparece al bajarla. Además, almacena la mayor parte de la información sobre la propia estructura de la base de datos que es consultada con más frecuencia.

Su tamaño no puede ser cambiado, pero si puede ser visto con el comando "*SHOW SGA*" en el *SQL*DBA*. Su longitud está definida por los parámetros del archivo de iniciación *INIT.ORA*. ⁶

Esta Compuesto por:

- *Diccionario Cache*
- *Los Redo Log Buffers*
- *Los Database Buffers*

⁶ www.monografias.com/trabajos25/oracle/oracle.shtml

Diccionario Cache

Es un conjunto de tablas que son usadas para proveer información asociada con la base de datos, donde se definen los nombres de usuarios, privilegios, objetos, retenciones y espacios que conforman un RDBMS *Oracle*. La base de datos gestiona internamente el tamaño de la caché de diccionario. ⁶

Redo Log Buffers

Es un espacio reservado en memoria que contiene los cambios realizados a los bloques de datos, permitiendo la reconstrucción de la base de datos en caso de ocurrir un accidente. ⁶

Database Buffers

Es un espacio reservado en memoria para las operaciones normales de la base de datos, el cual depende del tamaño especificado en el archivo de inicialización (INIT.ORA). ⁶

Shared Pool

El área compartida almacena la caché de diccionario de datos y la caché de biblioteca (información sobre las instrucciones ejecutadas contra la base de datos).

Mientras la caché de *buffer* de bloques de datos y la caché de diccionario permiten que los usuarios de la base de datos compartan la información sobre los datos y su estructura, la caché de biblioteca permite que se compartan las instrucciones SQL que se emplean con más frecuencia. El área compartida incluye la caché de biblioteca, la caché de diccionario de datos y el área SQL compartida. ⁷

Procesos de Segundo Plano

Las relaciones entre las estructuras físicas y de memoria de la base se mantienen y aplican mediante procesos en segundo plano. Estos procesos son propios de la base de datos y su número varía en función de la configuración de la base de datos. La base de datos gestiona estos procesos y necesita muy poco trabajo administrativo. ⁷

⁶ www.monografias.com/trabajos25/oracle/oracle.shtml

⁷ OSBORNE/McMRAW-WILL/Interamericana de España – Oracle 7.0 Manual de referencia.

DBWR (*DataBase Writer*). Escribe en los ficheros de la Base de Datos los *buffers* de datos modificados en memoria. ⁷

LGWR (*LoG Writer*). Escribe en los ficheros históricos de la Base de Datos (redo log files) cuando los *redo log buffers* se han llenado o cuando se produce un *commit*. Esto permite el *recovery*. ⁷

SMON (*System MONitor*). Trata las recuperaciones en caso de fallo. Limpia los segmentos temporales cuando no están en uso. Realiza el *Coalesce de tablespaces*, uniendo en una sola extensión extensiones contiguas. ⁷

PMON (*Process MONitor*). Recuperaciones en caso de fallo de un usuario. Libera recursos inactivos. En configuraciones *Multi-Threaded Server* restablece los procesos dispatcher y servidores caídos. ⁷

SMON y PMON se activan en forma periódica para comprobar si es necesaria su intervención. ⁷

CKPT. Proceso de Apoyo al LGWR. Actualiza ficheros de control y las cabeceras de ficheros de datos agilizando puntos de ruptura. Este proceso se activa si indicamos en INIT.ORA la línea. ⁷

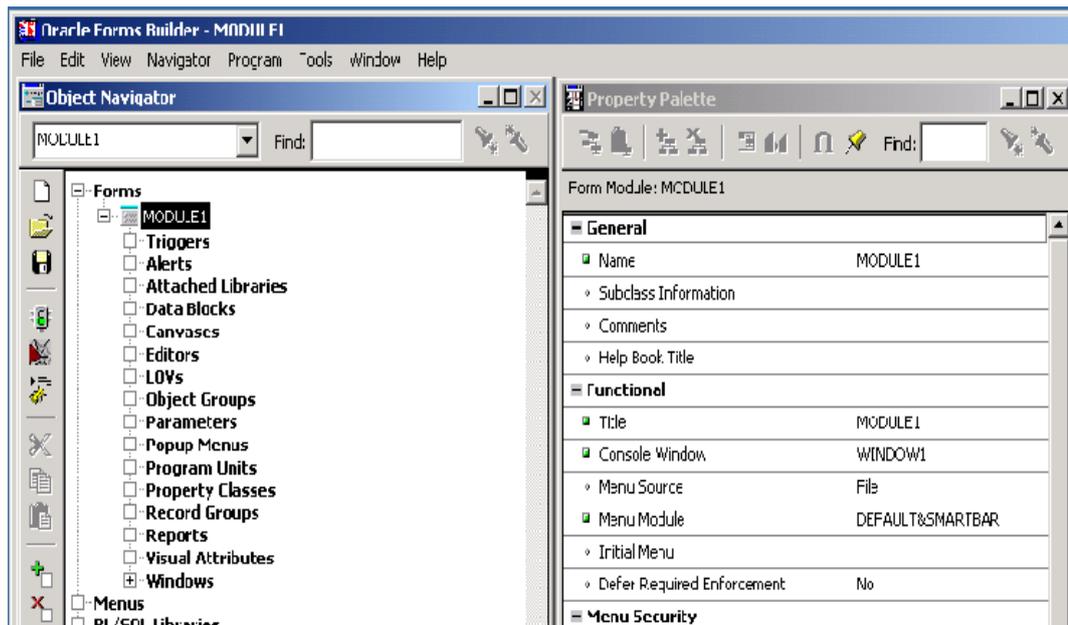
ARCH. Proceso encargado de copiar los históricos (*redo log files*) en ficheros aparte (*archiver files*) antes de que sean sobrescritos. ⁷

3.1.2 Forms 6.0.

El entorno de FORMS consiste de una pantalla con una barra de herramientas, un navegador de objetos y una paleta de propiedades, tal como se aprecia en la figura: ⁸

⁷ OSBORNE/McMRAW-WILL/Interamericana de España – Oracle 7.0 Manual de referencia.

⁸ ERAZO Juan Carlos: “Curso básico de developer forms”

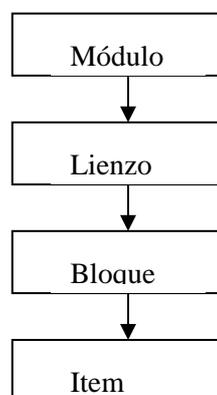


Para agregar un nuevo objeto se debe ubicar el cursor sobre éste y presionar el botón con el símbolo más (+) de la barra de herramientas izquierda. Para eliminarlo se debe presionar el botón con el símbolo equis (X) en la misma barra de herramientas. 8

Cada objeto tiene un conjunto de propiedades visibles en la paleta de propiedades. Para modificar una propiedad solamente se necesita ubicar el cursor en la celda junto al nombre de la misma y cambiar su contenido. FORMS soporta la creación de formularios (formas) y menús. Sin embargo dentro de su entorno todos se conocen como módulos. 8

Crear una forma básica

Para crear una forma (o módulo) es necesario conocer la jerarquía de objetos de FORMS:



- El Lienzo es el área de la ventana en la cual se ubicarán los ítems.
- El bloque es una agrupación lógica de ítems. Los bloques pueden estar asociados o no con la base de datos en cuyo caso se llamarán bloques de datos. También pueden ser usados para controlar la ejecución del módulo, en cuyo caso se llamarán bloques de control. Estos no están asociados directamente con objetos de la base de datos.
- El ítem es el objeto de interfaz que despliega información. Cada pequeño objeto susceptible de ser incluido en un lienzo se considera como un ítem. Es así como podemos encontrar: campos de texto, listas de valores, imágenes, botones de chequeo, botones comunes, sonidos, controles de *Visual Basic* y contenedores OLE, entre otros. ⁸

Crear una forma manualmente

Para crear una forma se usan estos pasos:

1. Crear un nuevo módulo. Propiedades a tener en cuenta:

- a. Nombre
- b. Título
- c. Ventana
- d. Módulo menú: indica el módulo usado para crear el menú. También indica que la forma será padre para todos los demás módulos. El nombre del módulo debe incluirse completo (nombre + extensión)
- e. Primer bloque de navegación: le indica a la forma que se ubique en ese módulo al momento de ejecutarse
- f. Unidad de validación: indica el orden en que se procesarán los triggers.
- g. Modo de interacción: indica cómo el usuario interactuará con la forma cuando se ejecuta una consulta. Por ejemplo, en modo *BLOCKING* el usuario no podrá cambiar el tamaño de la ventana ⁸

2. Crear un nuevo lienzo (*CANVAS*). Propiedades a tener en cuenta:

- a. Nombre
- b. Tipo: indica la forma como se desplegará en la pantalla. Por ejemplo *Content* significa que ocupará toda el área de contenido

⁸ ERAZO Juan Carlos: “Curso básico de developer forms”

- c. Visible
- d. Ventana: indica en qué ventana se desplegará
- e. Ancho y alto
- f. Apariencia del borde (*BEVEL*)
- g. Atributos visuales ⁸

3. Crear bloques: Propiedades a tener en cuenta:

- a. Nombre
- b. Estilo de navegación: indica como se procederá en caso de estar en el primero o en el último registro del bloque
- c. Bloques anterior y siguiente
- d. Atributos visuales para el registro actual
- e. Longitud del arreglo de consulta. Indica el número de registros que se traerán en una sola operación de lectura.
- f. Número de registros cargados: especifica el número de registros que se cargarán a memoria durante una consulta sobre el bloque (por defecto son tres)
- g. Número de registros desplegados: especifica el número de registros que un bloque puede desplegar cada vez. Esta propiedad afecta directamente la forma como se verá el bloque
- h. Consultar todos los registro: indica si al momento de consultar el bloque se traerán todos los registros o solo la cantidad especificada en la propiedad Longitud del arreglo de consulta . Se usa comúnmente cuando hay ítems de resumen (por ejemplo sumas)
- i. Bloque de base de datos: especifica el tipo de bloque
- j. Consulta, inserción, actualización y eliminación permitida: indican las operaciones que se pueden ejecutar sobre los registros del bloque
- k. Forzar llave primaria: comúnmente *FORMS* usa el *ROWID* para identificar los registros en el bloque. Con esta propiedad se le indica que use las llaves primarias de la tabla base.
- l. Origen de datos ⁸

⁸ ERAZO Juan Carlos: “Curso básico de developer forms”

m. Nombre origen de datos

n. Destino de datos ⁸

4. Crear ítems: Las propiedades de los ítems varían de acuerdo al tipo de los mismos.

Estas son algunas de las propiedades comunes

a. Nombre

b. Tipo

c. Habilitado

d. Justificación: indica la ubicación en el bloque

e. Restricción del texto: indica la forma como se desplegará el texto (mayúsculas, minúsculas, etc.)

f. Navegable con el teclado

g. Ítems anterior y siguiente

h. Tipo de dato

i. Requerido

j. Máscara de formato

k. Elemento de base de datos

l. Nombre de columna

m. Consulta, inserción, actualización permitida

n. Visible

o. Lienzo contenedor

p. Visible ⁸

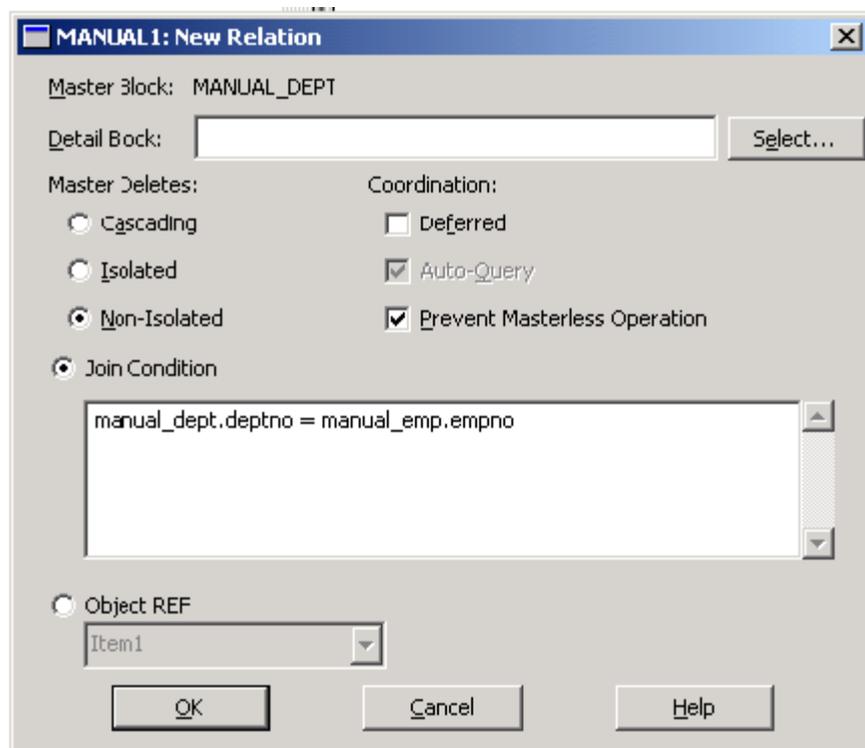
Crear relaciones entre bloques

No todas las formas contienen un único bloque sobre el cual interactuar. Muchas de ellas contienen más de uno (algunos maestro detalle) los cuales están relacionados entre si. Para crear una relación entre dos bloques (ya deben estar creados ambos bloques) se deben seguir los siguientes pasos:

1. Ubicarse sobre el bloque maestro y crear una nueva relación
2. Seleccionar el bloque detalle en la ventana de creación de relaciones
3. Seleccionar las operaciones de eliminación de registros: en cascada para eliminar los registros hijos junto con el padre.

⁸ ERAZO Juan Carlos: “Curso básico de developer forms”

4. Seleccionar la coordinación entre los bloques: diferida para que los registros hijos no aparezcan cuando se hace la consulta sobre el bloque maestro; para verlos se necesita hacer la consulta sobre el bloque hijo.
5. Escribir la operación de *join*
6. O seleccione un objeto existente en ambos bloques sobre el cual se basará la relación. Una vez creada la relación *forms* añadirá nuevas unidades de programa y *triggers* para controlar las operaciones de coordinación entre ambos bloques. Es posible crear las relaciones de forma automática. Para ello se necesita crear primero el bloque maestro y luego el detalle maestro. ⁸



Uso Asistente Layout

Para crear al layout manualmente se debe invocar el editor y posteriormente añadir uno a uno los objetos requeridos. Antes de ubicar los objetos se debe tener cuidado de seleccionar el lienzo y el bloque con el cual se trabajará, de lo contrario los objetos aparecerán ubicados fuera de lugar. La siguiente tabla muestra la descripción de algunos de los botones de la barra de herramientas del editor que se usan para añadir objetos al lienzo. ⁸

⁸ ERAZO Juan Carlos: “Curso básico de developer forms”

	Añadir ítem texto
	Añadir campo de texto
	Añadir marco (frame)
	Añadir botón de radio
	Añadir botón de chequeo
	Añadir imagen
	Añadir botón
	Añadir ítem tipo lista desplegable
	Añadir ítem para desplegar texto
	Añadir lienzo en forma de separador

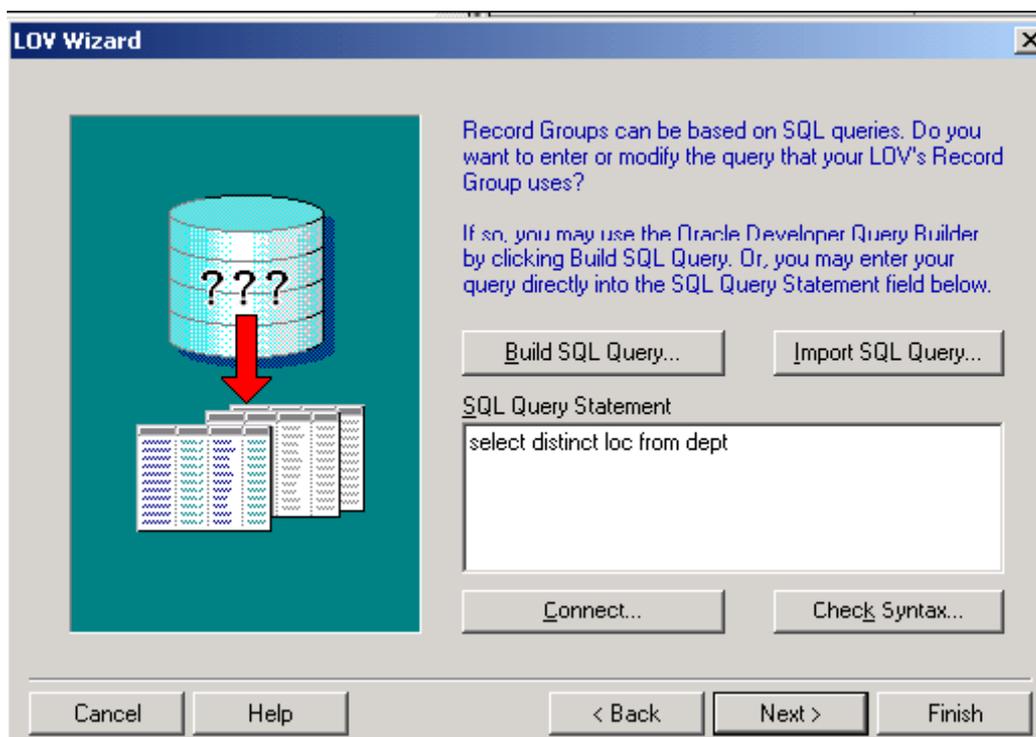
Existen adicionalmente algunos botones en la barra de herramientas horizontal que nos permiten manipular la alineación de los objetos.

	Alinear los ítems a la izquierda.
	Alinear los ítems al centro vertical
	Alinear los ítems a la derecha
	Alinear los ítems a la parte superior
	Alinear los ítems al centro horizontal
	Alinear los ítems a la parte inferior

Lista de Valores (LOV)

Para crear la lista de valores se deben seguir estos pasos que coinciden con las pantallas del asistente.

1. Escoger un grupo de registros o crear uno basado en una consulta
2. Ingresar la consulta para obtener los registros



3. Escoger las columnas del grupo de registros que irán en la lista de valores
4. Colocar un nombre para las columnas de la lista y asignar el ítem al cuál retornará la selección
5. Escoger el título de la lista, el ancho, el alto y la ubicación
6. Seleccionar el número de registros a recuperar
7. Asignar la lista a un ítem del bloque ⁸

3.1.2 Report 6.0.

Oracle reports es una poderosa herramienta que tiene por objetivo el diseño y la generación de informes. Permite la creación de reportes en archivos jsp (*Java Server pages*), rdf, xml, rtf entre otros, pero siendo los más usados los anteriormente citados. ⁹

⁸ ERAZO Juan Carlos: "Curso básico de developer forms"

⁹ www.zonaoracle.com

De igual manera permite enviar el resultado de los informes a archivos de texto, pdf, html, xml, rtf, de texto delimitados, entre otros, lo cual permite su lectura y publicación en diversos formatos. Al igual que oracle forms, esta herramienta contiene un navegador de objetos desde el cual se puede acceder a cada uno de los elementos que conforman la definición de un reporte.⁹

3.1.4 PL/SQL Developer 4.0.

PL/SQL es un lenguaje de programación estructurado y procedimental que amplía la funcionalidad de SQL, añadiendo estructuras habituales en otros lenguajes de programación, entre las que se encuentran:¹⁰

- ◆ Variables y Tipos
- ◆ Estructuras de control
- ◆ Procedimientos y Funciones
- ◆ Tipos de Objetos y Métodos.

La unidad básica en PL/SQL es el bloque. Todos los programas PL/SQL están compuestos por bloques que pueden estar anidados. Un bloque PL/SQL está compuesto de tres partes principales:

- ◆ Sección Declarativa (opcional). Contiene las variables, constantes ...
- ◆ Sección Ejecutable (obligatoria). Contiene órdenes SQL para manipular datos de la base de datos y órdenes PL/SQL para manipular los datos del bloque
- ◆ sección de excepciones (opcional). Especifica las acciones a realizar en caso de error o cuando se producen excepciones en la ejecución.¹⁰

La estructura general es:

[DECLARE

variables, constantes, excepciones de usuario...]

BEGIN

órdenes SQL

órdenes PL/SQL

⁹ www.zonaoracle.com

¹⁰ ORACLE: Introduction to Oracle 9i: PL/SQL. Student guide. 2001 "Practica8.pdf"

[*EXCEPTION*

acciones a realizar al ocurrir un error]

END;

Podemos crear diferentes tipos de bloques:

- ◆ Bloques anónimos: Se construyen de forma dinámica y se suelen ejecutar una sola vez.
- ◆ Bloques nominados: Igual que los anónimos pero con una etiqueta que les da nombre.
- ◆ Subprogramas: Procedimientos, paquetes y funciones, almacenados en la base de datos y que se ejecutan en múltiples ocasiones. Los subprogramas se ejecutarán mediante una llamada.
- ◆ Disparadores (“*Triggers*”): Bloques nominados que se almacenan en la base de datos y que se ejecutan ante algún suceso.¹⁰

Variables

Las variables se definen en la sección declarativa de los bloques PL/SQL dónde también pueden inicializarse. La asignación de nuevos valores a las variables puede hacerse en la parte ejecutable del bloque.

Pueden utilizarse para pasar valores como argumentos a subprogramas. Estas podrán ser de tipo *IN* (variable de entrada, *OUT*, variable de salida o *INOUT*, variable de entrada/salida). También podrán utilizarse para almacenar valores devueltos o requeridos por una orden SQL.¹⁰

Todas las variables tienen un tipo. Los posibles tipos de una variable son:

- ◆ Escalar. Almacenan un valor único. Son los mismos que los de las columnas de las tablas (*VARCHAR2*, *NUMBER*, *DATE*, *CHAR*, *LONG*, *LONG_RAW*, *BINARY_INTEGER*, *LAW_INTEGER*) más el *BOOLEAN*
- ◆ Compuesto. Grupos de datos: tablas PL/SQL, registros...
- ◆ Puntero. Designan elementos de otros programas.

¹⁰ ORACLE: Introduction to Oracle 9i: PL/SQL. Student guide. 2001 “Practica8.pdf”

- ◆ LOB (*Large Objects*). Almacenan gran cantidad de información. Las variables de tipo LOB permiten almacenar datos no estructurados (imágenes, texto...) de hasta 4GB de tamaño¹⁰

Declaración de Variables

Sintaxis:

<identificador> [*CONSTANT*] <tipo_de_dato> [*NOT NULL*] [{:= | *DEFAULT* <expresión>}];

Ejemplo:

DECLARE

fecha *DATE*;

dep_num *NUMBER(2) NOT NULL := 10*;

ciudad *VARCHAR2(10) := 'Ciudad Real'*;

Km_a_milla *CONSTANT NUMBER := 1.4*;

Las variables declaradas como *NOT NULL* siempre deben ser inicializadas.

La inicialización puede hacerse utilizando := o la palabra reservada *DEFAULT*.

Si una variable no se inicializa contendrá el valor *NULL*.

Las constantes deben ser inicializadas.¹⁰

Asignación de valores a variables

Sintaxis:

<identificador> := <valor>;

ATRIBUTO %TYPE

El atributo *%TYPE* se utiliza para declarar una variable con el mismo tipo que una columna de una tabla o que otra variable definida anteriormente.¹⁰

Sintaxis:

<identificador> {<tabla>.<columna> | <nombre_variable>}%*TYPE*;

¹⁰ ORACLE: Introduction to Oracle 9i: PL/SQL. Student guide. 2001 "Practica8.pdf"

Ejemplo:

```
var_nombre Empleados.nombre%TYPE;  
balance NUMBER;  
balance_minimo balance%TYPE := 10;
```

Variables booleanas

Las variables BOOLEANAS pueden tomar el valor *TRUE*, *FALSE* o *NULL*. Las variables pueden combinarse mediante operadores lógicos (*NOT*, *AND*, *OR*).¹⁰

Las expresiones pueden devolver valores BOOLEANOS utilizando operadores relacionales (<, <=...).¹⁰

Órdenes ejecutables

Comentarios en PL/SQL

Pueden añadirse comentarios al código. Estos comentarios pueden ser especificados con:¹⁰

```
/*comentario  
más comentario */  
-- comentario de línea
```

Funciones PL/SQL

Las funciones utilizables en SQL (*LOWER*, *UPPER*, *INITCAP*, *CANCAT*, *SUBSTR*, *LENGTH*, *ROUND*, *TRUNC*, *MOD*, *MONTHS_BETWEEN*, *ADD_MONTHS*, *NEXT_DAY*, *LAST_DAY*) excepto las de agrupamiento (ya que éstas se aplican sobre una columna de una tabla).¹⁰

Conversión de tipos

Existen funciones de conversión de tipos: *TO_CHAR*, *TO_DATE*, *TO_NUMBER*

Sintaxis:

```
TO_CHAR (<valor>, <formato>)  
TO_DATE(<valor>, <formato>)  
TO_NUMBER(<valor>, <formato>)
```

¹⁰ ORACLE: Introduction to Oracle 9i: PL/SQL. Student guide. 2001 "Practica8.pdf"

Operadores

Los operadores en PL/SQL son los mismos que para SQL: Aritméticos, Lógicos, Concatenación, Paréntesis. Y además, existe el operador exponencial (**).¹⁰

Entrada/Salida

Los programas PL/SQL suelen realizar operaciones específicas sin interactuar con el operador. Sin embargo, existen algunas funciones que nos pueden ayudar a depurar programas y a interactuar con el usuario mostrando datos por pantalla y pidiendo datos al usuario.¹⁰

Salida de datos

Para mostrar una cadena por pantalla podemos utilizar:

```
DBMS_OUTPUT.PUT_LINE(<cadena de caracteres>);
```

Si los datos a mostrar no son cadenas puede utilizarse la función *TO_CHAR()* para transformarlo, y el operador `||` para concatenar.

El paquete *DBMS_OUTPUT* implementa una cola, en la cual se van almacenando los mensajes de salida. Si queremos que los mensajes aparezcan por pantallas tenemos que activar la opción *SERVEROUTPUT*:

```
SET ServerOutput ON;10
```

Entrada de datos

Cuando trabajamos pidiendo datos al usuario es habitual especificar la opción *SET VERIFY OFF* para evitar que el sistema nos muestre el valor que tenía la variable antes y que nos confirme el nuevo valor que toma.

Para pedir datos al usuario se utiliza una variable de sustitución, dentro del código fuente del bloque PL/SQL, si esta variable no está inicializada, se le pedirá el valor al usuario:¹⁰

¹⁰ ORACLE: Introduction to Oracle 9i: PL/SQL. Student guide. 2001 "Practica8.pdf"

```

SET ServerOutput ON;
SET VERIFY OFF;
DECLARE
    vv NUMBER :=&v;
BEGIN
    DBMS_OUTPUT.PUT_LINE('Valor de v: '||vv);
END;

```

Órdenes SQL en PL/SQL

Podemos utilizar instrucciones SQL dentro de los bloques PL/SQL para recuperar datos de la base de datos o para actualizar los datos que contiene.¹⁰

Sintaxis:

```

SELECT <lista>
    INTO {<variable>[, <variable>, ...] | <registro>}
    FROM <tabla>
WHERE <condición>;

```

El *SELECT* almacenará los valores que obtenga en las variables indicadas tras *INTO* y en el mismo orden.

Es obligatorio incluir la cláusula *INTO*.

El *SELECT* debe prepararse para que sólo devuelva una fila.

Ejemplo:

```

DECLARE
    v_apellidos VARCHAR2(50);
    v_nombre VARCHAR2(30);
BEGIN
    SELECT apellidos, nombre
        INTO v_apellidos, v_nombre
        FROM Empleados
        WHERE id='5';
END;
INSERT, UPDATE, DELETE

```

¹⁰ ORACLE: Introduction to Oracle 9i: PL/SQL. Student guide. 2001 "Practica8.pdf"

La sintaxis no varía:

- ◆ *INSERT INTO* tabla *VALUES*(...);
- ◆ *UPDATE* tabla
 SET valor = expresión
 WHERE condición;
- ◆ *DELETE FROM* tabla *WHERE* condición;

Las expresiones y en las condiciones podemos utilizar variables PL/SQL.

Estructuras de control

Orden IF

El funcionamiento de la estructura *IF* es el habitual. Su Sintaxis es la siguiente: ¹⁰

```
IF <expresión1> THEN  
    <Secuencia_ordenes1>;  
    [ELSIF <expresión2> THEN  
        <Secuencia_ordenes2>;]  
    [ELSE  
        <Secuencia_ordenesN>;]  
END IF;
```

Ejemplo:

```
SET ServerOutput ON;  
SET VERIFY OFF;  
DECLARE  
    v_num NUMBER := &v;  
BEGIN  
    IF v_num < 50 THEN  
        DBMS_OUTPUT.PUT_LINE('Valor pequeño'  
    ELSIF v_num < 100 THEN  
        DBMS_OUTPUT.PUT_LINE('Valor mediano');  
    ELSE  
        DBMS_OUTPUT.PUT_LINE('Valor grande');  
    END IF;  
END;
```

10 ORACLE: Introduction to Oracle 9i: PL/SQL. Student guide. 2001 "Practica8.pdf"

Orden LOOP

Los bucles *LOOP* son bucles que se ejecutan siempre, para salir de ellos tendremos que poner una instrucción de salida dentro del bucle. ¹⁰

Su Sintaxis es:

```
LOOP
    [EXIT WHEN <condición>]
END LOOP;
```

Ejemplo:

```
SET ServerOutput ON;
SET VERIFY OFF;
DECLARE
    num NUMBER :=1;
BEGIN
    LOOP
        DBMS_OUTPUT.PUT_LINE('Valor: || num);
        num := num +1;
        EXIT WHEN num > 10;
    END LOOP;
END;
/
```

Orden FOR

Los bucles *FOR* se repiten un número determinado de veces. ¹⁰

Sintaxis:

```
FOR <contador> IN <min>..<max> [REVERSE] LOOP
    .....
END LOOP;
```

¹⁰ ORACLE: Introduction to Oracle 9i: PL/SQL. Student guide. 2001 "Practica8.pdf"

Ejemplo:

```
SET ServerOutput ON;
SET VERIFY OFF;
DECLARE
    num NUMBER;
BEGIN
FOR num IN 1..10 LOOP
    DBMS_OUTPUT.PUT_LINE('Valor: || num);
    END LOOP;
END;
```

Orden WHILE

Los bucles *WHILE* son iguales que en otro lenguajes de programación.¹⁰

Sintaxis:

```
WHILE <condición> LOOP
    .....
END LOOP;
```

Ejemplo:

```
SET ServerOutput ON;
SET VERIFY OFF;
DECLARE
    num NUMBER:=1;
BEGIN
    WHILE num <=10 LOOP
        DBMS_OUTPUT.PUT_LINE('Valor: || num);
        num := num + 1;
    END LOOP;
END;
```

¹⁰ ORACLE: Introduction to Oracle 9i: PL/SQL. Student guide. 2001 "Practica8.pdf"

3.2 Análisis.

3.2.1 Análisis de Necesidades.

Dentro del análisis de las necesidades parte principalmente desde la observación del desempeño actual del departamento de recursos humanos de la Universidad del Azuay, y el método de entrevista con el personal que labora dentro de este departamento; los cuales muestran que en su funcionamiento no poseen ningún sistema de apoyo para sus labores, a pesar de que dicho departamento cuenta con computadora, los archivos del personal tanto docente como administrativo se manejan aun en archivadores y los procesos son totalmente manuales.

Por lo que es esencial la implementación de un sistema automatizado para el manejo del archivo del personal de acuerdo con la información que poseen y de la manera como están acostumbrados a trabajar; dicho de otra manera pasar todo a una forma automática, rápida y segura.

3.2.2 Análisis de la Viabilidad

Este análisis refleja la viabilidad para implementar el sistema dentro del departamento.

Viabilidad Económica: El costo económico que el sistema implica; es viable ya que el departamento cuenta con computadora en la que se instalaría el sistema. El sistema es desarrollado en *Oracle 8.0* ya que la Universidad cuenta con las licencias respectivas. Solo estaría el tiempo en capacitación al personal para el manejo del sistema.

3.2.3 Análisis de Requisitos

Es el proceso de descubrimiento, refinamiento, modelado y especificación se centra especialmente en el *software*, para comprender la naturaleza de los programas a realizarse, es necesario comprender el dominio de la información del *software*, así como la función requerida, comportamiento, rendimiento del *software*, descripción de interfaz del *software* con otros elementos del sistema, establecer restricciones que

debe cumplir; y proporcionar los medios para valorar la calidad una vez construido el *software*.

Si bien es cierto que el departamento de Recursos Humanos de la Universidad del Azuay no cuenta con un sistema propio para su dependencia, los demás departamentos están trabajando en sistemas desarrollados en base de datos *Oracle* y la Universidad cuenta con el departamento de Centro de Cómputo donde manejan toda la información de los sistemas que se encuentran funcionando. Este departamento me proporcionó su base de datos pudiendo ocupar todas las tablas con la información almacenada en ella que sea necesario para desarrollar el sistema.

Tabla Empleado

Tn_empleado
Cod_empleado
Cod_empresa
Cod_tipo_empleado
Cod_tipo_identificacio n
Identificacion
Apellido_paterno
Apellido_materno
Primer_nombre
Segundo_nombre
Sexo
Estado_civil
Cod_nacionalidad
Cod_pais
Nivel
Codigo
Direccion
Direccion_numero
Fecha_nacimiento
Afiliacion_iess
Libreta_militar
Activo
Discapacitado

→ Nombre de la tabla

Atributos de la tabla

Tabla Empleado – Titulo

Tn_empleado_titulo
Cod_empleado
Secuencia
Cod_tipo_posgrado
Titulo
Universidad_adq

Fecha
Cod_empresa
Nro_afiliacion
Cod_carrera

Tabla Empleado – Carga

Tn_empleado_carga
Cod_empleado
Cod_empresa
Secuencia
Tipo_carga
Apellidos
Nombres
Sexo
Fecha_nacimiento
Fecha_defuncion
Carga_tributaria
Carga_institucion
Carga_familiar
Seguro

Tabla Empleado – Banco

Tn_empleado_banco
Cod_empleado
Cod_empresa
Secuencia
Cod_banco
Nro_cuenta
Fecha_inicio
Fecha_final
Tipo
Estado_activo

Tabla Empleado – Teléfono

Tn_empleado_telefono
Cod_empleado
Cod_empresa
Secuencia
Telefono

Tabla Empleado – Email

Tn_empleado_mail
Cod_empleado
Cod_empresa

Mail
Si_no

Tabla Cargo

Tn_cargo
Cod_cargo
Cod_empresa
Cod_nivel
Nombre
Cod_cargo_padre
Salario_nomina

Tabla Parámetro

Tn_parametro
Cod_parametro
Cod_empresa
Cod_parametro_alterno
Nombre
Tipo_valor
Valor
Observaciones

Tabla Asistencia – Docente

Pe_asistencia_docente
Cod_empleado
Secuencia_marca
Fecha_marca
Cod_periodo
Tipo_marca
Hora_marca
Minuto_marca
Estado
Cod_clase
Docente_alumno
Adicionado_por
Fecha_adicion
Modificado_por
Fecha_modificacion
Observaciones

Tabla Asistencia Docente – Histórico

Pe_asistencia_docente_h
Cod_empleado
Secuencia_marca

Fecha_marca
Cod_perodo
Tipo_marca
Hora_marca
Minuto_marca
Estado
Cod_clase
Docente_alumno
Adicionado_por
Fecha_adicion
Modificado_por
Fecha_modificacion
Observaciones

Tabla Horario – Detalle

Tna_horario_detalle
Cod_horario
Dia
Secuncia
Hora_e
Hora_s

Tabla Registro

Tna_registro
Cod_emplado
Cod_horario
Fecha
Dia
Secuencia
Hora_e
Hora_s
D_hora_e
D_hora_s

Tabla Registro – Histórico

Tna_registro_h
Cod_empleado
Cod_horario
Fecha
Dia
Secuencia
Hora_e
Hora_s
D_hora_e
D_hora_s

De otra manera para las necesidades que requiere el departamento de Recursos Humanos de la Universidad del Azuay fue necesario crear nuevas tablas e incorporarlas a la base de datos ya existente. Las tablas creadas para el uso exclusivo de dicho departamento y que se manejan en el “Sistema de Archivo Virtual” son las que describo a continuación.

Tabla Registro – Asistencia

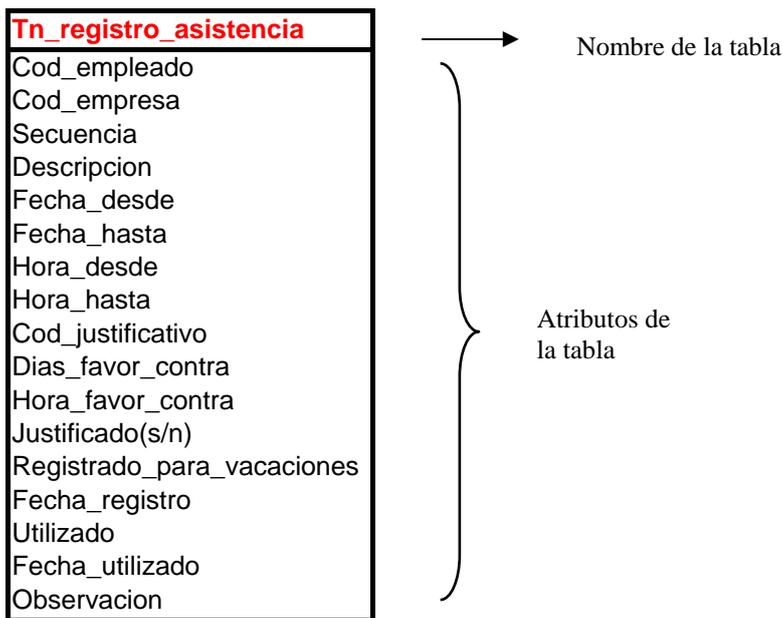


Tabla Documento

Tn_documento
Cod_documento
Descripcion
Utilitario
Path

Tabla Empleado – Documento

Tn_empleado_documento
Cod_empleado
Cod_empresa
Cod_documento
Path
Descripcion

Tabla Empleado – Dependencia

Tn_empleado_dependencia
Cod_empleado
Cod_empresa
Cod_secuencia
Cod_division_dependencia
Fecha_inicial
Fecha_final
Activo
Predeterminado
Cod_cargo
Cod_cargo_padre

Tabla Justificativo

Tn_justificativo
Cod_justificativo
Descripcion

Tabla Vacación Anual

Tn_vacacion_anual
Cod_empleado
Cod_empresa
Periodo_desde
Periodo_hasta
Observaciones
Total_dias
Total_dias_favor
Total_dias_contra
Total_horas_favor
Total_horas_contra
Activo

De igual manera el sistema cuenta con vistas que son un reflejo de las tablas con otros nombres.

Las vistas que se crearon para el Sistema de Archivo Virtual se detallan a continuación en lenguaje PL/SQL.

Vista Rol Empleado (vi_rol_empleado)

```
create      or      replace      view      vi_rol_empleado      as
select
a.cod_rol,a.fecha_inicial,a.fecha_final,a.descripcion,b.cod_empleado
,b.cod_empresa,
b.cod_division_dependencia,c.descripcion      dependencia,b.egreso
neto_recibido
from      tn_rol      a,tn_rol_empleado      b,tn_division_dependencia      c
where
      b.cod_rubro      =      201
and
      a.cod_rol      =      b.cod_rol
and
      a.cod_empresa      =      b.cod_empresa
and
      a.cod_empresa      =      c.cod_empresa
and
      b.cod_division_dependencia      =      c.cod_division_dependencia
```

Vista Docente Período (vi_docente_periodo)

```
create      or      replace      view      vi_docente_periodo      as
select
a.cod_docente,a.cod_carrera,a.cod_materia,c.nombre,b.cod_periodo,b.n
ombre      periodo,d.nombre      carrera
from      te_clase      a,te_periodo      b,te_materia      c,te_carrera      d
where
      a.cod_periodo      =      b.cod_periodo
and
      a.cod_materia      =      c.cod_materia
and
      a.cod_carrera      =      d.cod_carrera
```

Vista Horario (vi_horario)

```
create      or      replace      view      vi_horario      as
(select      a.cod_horario      cod_horario, trunc(sum((b.hora_s      -
1900)/100),0)      horas      from      tna_horario      a, tna_horario_detalle      b
where      a.cod_horario      =      b.cod_horario
and      a.recargo      =      1
and      b.hora_s      >      1900
group by a.cod_horario)
```

Vista Horario Docente (vi_horario_docente)

```
create      or      replace      view      vi_horario_docente      as
select      a.cod_docente, a.cod_carrera, e.nombre
carrera, a.cod_materia, c.nombre, b.cod_periodo, b.nombre      periodo,
dia, decode(d.dia, 1, 'LUNES', 2, 'MARTES', 3, 'MIERCOLES', 4, 'JUEVES', 5, 'VI
ERNES', 6, 'SABADO', 7, 'DOMINGO')      dia_n
, d.hora_desde, d.minuto_desde, d.hora_hasta, d.minuto_hasta
from      te_clase      a, te_periodo      b, te_materia      c, te_clase_horario
d, te_carrera      e
where      a.cod_periodo      =      b.cod_periodo
and      a.cod_materia      =      c.cod_materia
and      a.cod_clase      =      d.cod_clase
and      a.cod_periodo      =      d.cod_periodo
and      a.cod_docente      =      d.cod_docente
and      a.cod_carrera      =      e.cod_carrera
```

Vista Rol Empleado (vi_rol_empleado)

```
create      or      replace      view      vi_rol_empleado      as
select
a.cod_rol,a.fecha_inicial,a.fecha_final,a.descripcion,b.cod_empleado
,b.cod_empresa,
b.cod_division_dependencia,c.descripcion      dependencia,b.egreso
neto_recibido
from      tn_rol      a,tn_rol_empleado      b,tn_division_dependencia      c
where      b.cod_rubro      =      201
and      a.cod_rol      =      b.cod_rol
and      a.cod_empresa      =      b.cod_empresa
and      a.cod_empresa      =      c.cod_empresa
and      b.cod_division_dependencia      =      c.cod_division_dependencia
```

De igual manera la Universidad cuenta con una vista creada a partir de la tabla de empleados en la que se encuentran concatenados los nombres y apellidos de los empleados dicha vista responde al nombre de **vi_tn_empleado** y se detalla a continuación en lenguaje PL/SQL.

Vista Empleado (vi_tn_empleado)

```
create      or      replace      view      vi_tn_empleado      as
(select      cod_empleado,
substr(rtrim(      apellido_paterno)||'      '||
rtrim(      apellido_materno)||'      '||
rtrim(      primer_nombre)||'      '||
rtrim(      segundo_nombre),1,43)      NOMBRES,
cod_empresa
from      tn_empleado )
```

3.2.4 Análisis Estructurado

El análisis estructurado es la actividad de construcción de modelos que reflejan el flujo de datos y el contenido de la información.

Modelo Funcional

Describe las transformaciones de datos del sistema, y las operaciones existentes dentro del sistema, con resultados de cálculo sin detallar ni como ni cuando se calculan.

D.F.D.(Diagrama de Flujo de Datos) → Modela las funciones del sistema.

D.E.R. (Diagrama Entidad Relación) → Modela los datos del sistema.

Diagrama de Flujo de Datos (D.F.D)

Este modelo sirve para ilustrar las funciones que el sistema debe realizar y describir las transformaciones de las entradas en salidas. Un diagrama de flujo está formado por:

- ◆ Procesos: Están representados por círculos o burbujas. Representan las variadas funciones individuales que el sistema lleva a cabo. Las funciones transforman entradas en salidas.
- ◆ Flujos: Están representados por flechas. Son conexiones entre procesos y representan la información que el proceso requiere como entrada o la información que genera como salida.
- ◆ Almacenamiento de datos: Están representados por dos líneas paralelas o por una elipse. Muestran colecciones de datos que el sistema debe recordar por un periodo de tiempo. Luego de construido el sistema funciona como base de datos.

Procesos: que son los componentes funcionales del sistema

- Están representados por círculos o burbujas.
- Representan las variadas funciones individuales que el sistema lleva a cabo. Las funciones transforman entradas en salidas.

Almacenes: que representan datos almacenados o en reposo

- Están representados por dos líneas paralelas
- Muestran colecciones de datos que el sistema debe recordar por un periodo de tiempo.
- Luego de construido el sistema funciona como base de datos.

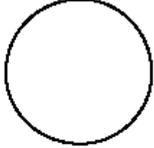
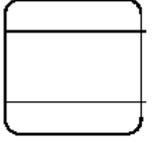
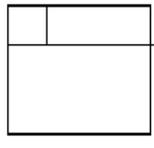
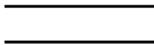
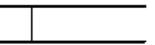
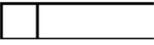
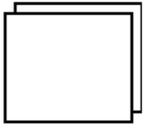
Entidades externas: que representan la fuente y/o el destino de la información del sistema

- Muestran las entidades externas con el que cada grupo debe comunicarse.

Flujos de datos: que representan los datos que fluyen entre las funciones

- Están representados por flechas.
- Son conexiones entre procesos
- Representan información requerida por proceso como entrada.

Simbología

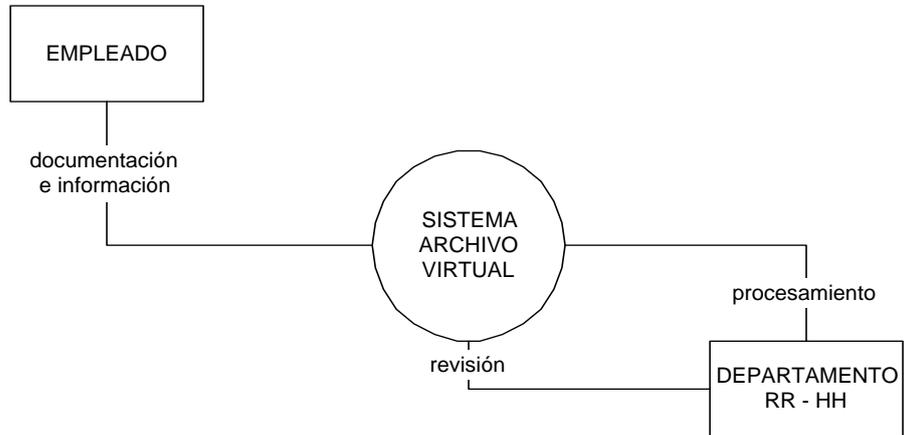
	Yourdon, DeMarco	Gane y Sarson	SSADM MÉTRICA
Flujos de datos			
Procesos			
Almacenes de datos			
Entidades externas			

- ◆ El modelo utiliza la simbología de Yourdon, DeMarco.
- ◆ El dfd resalta solo un aspecto funcional.

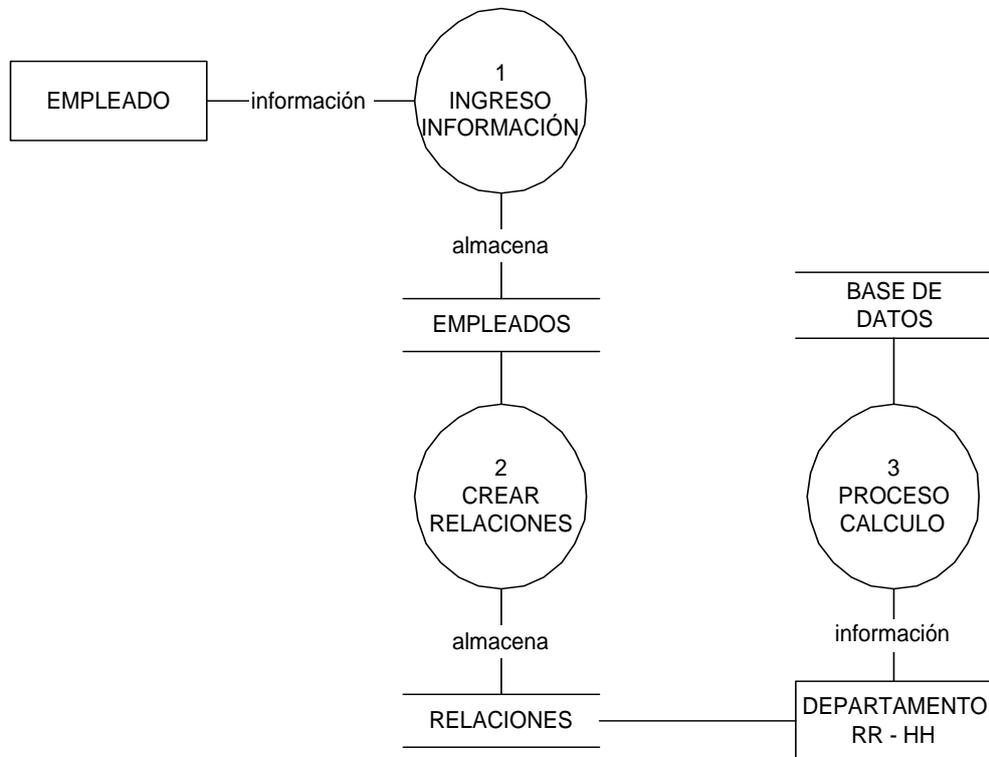
DIGRAMA DE FLUJO DE DATOS(D.F.D)

SISTEMA DE ARCHIVO VIRTUAL

NIVEL 1

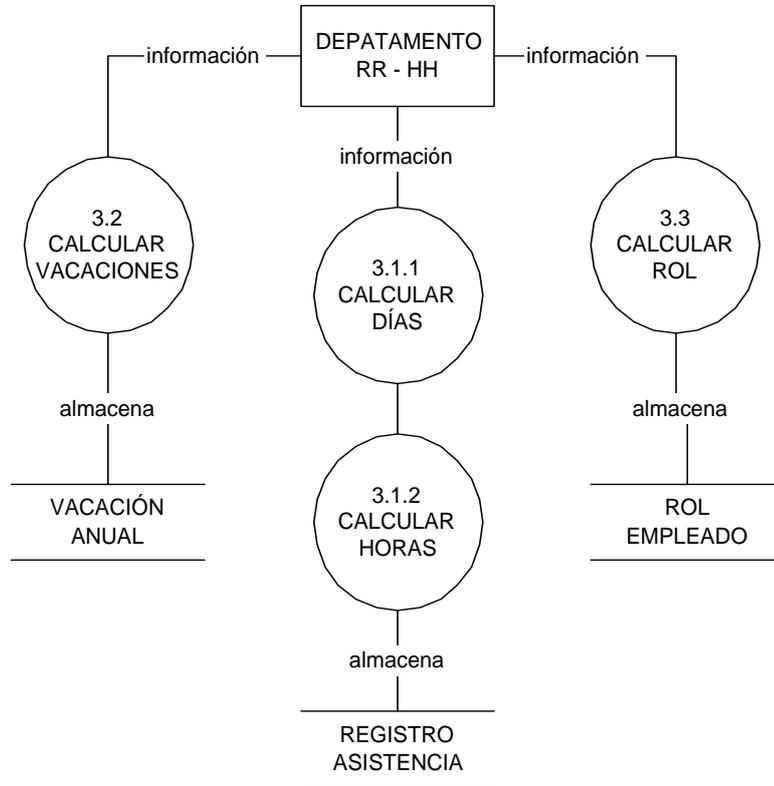


NIVEL 2



NIVEL 3

PROCESO 3



NIVEL 3

PROCESO 2

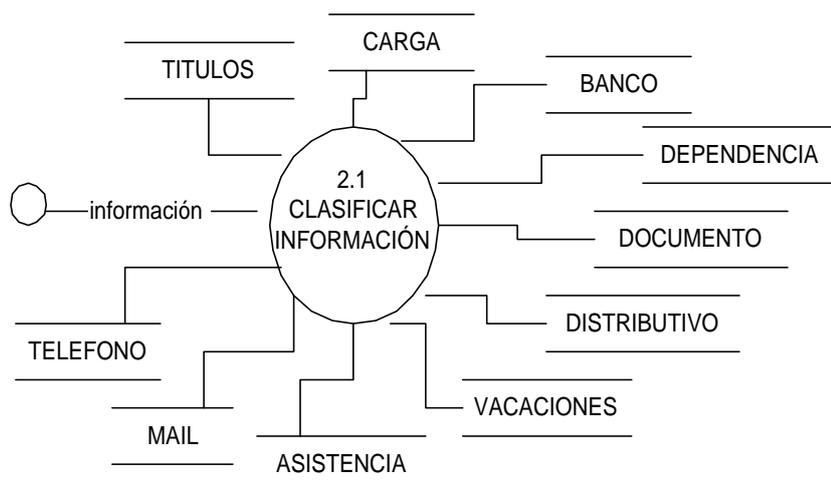


Diagrama Entidad-Relación (D.E.R)

El D.E.R. es la representación (gráfica) de los datos del sistema y sus relaciones, independientemente de las funciones y procesos que operan sobre ellos, describe las características de los datos, en forma abstracta.

Componentes:

- Entidades
 - Relaciones
-
- Entidades: Se representan mediante el rectángulo. Son cosas u objetos del mundo real, abstractas o concretas, tangibles o intangibles.

Una entidad se describe por sus atributos, un atributo es una propiedad o una cualidad, es lo que se puede decir de la entidad.

- Relación: Asociación o vinculación entre dos o más entidades.

Diagrama E-R, vista alto nivel de los datos (no muestra detalles), da énfasis a los datos, a los usuarios de alto nivel le importa más la información que los procesos, todo almacén de datos del diagrama flujo datos debe corresponderse con entidad del diagrama E-R, balanceo DFD, D.E-R.

Cardinalidad

En una relación que asocia a dos o más entidades, la conexión entre las instancias de las mismas puede tener distintos tipos de cardinalidad. Por ejemplo una relación entre dos entidades, A y B, puede asociar una instancia de A con 1 y/o más instancias de B y viceversa.

1-1: Una instancia de A se relaciona a lo sumo con una instancia de B y viceversa.

1-n: Una instancia de A se relaciona con 0, 1 o varias instancias de B, pero cada instancia de B esta asociada a lo sumo con una de A.

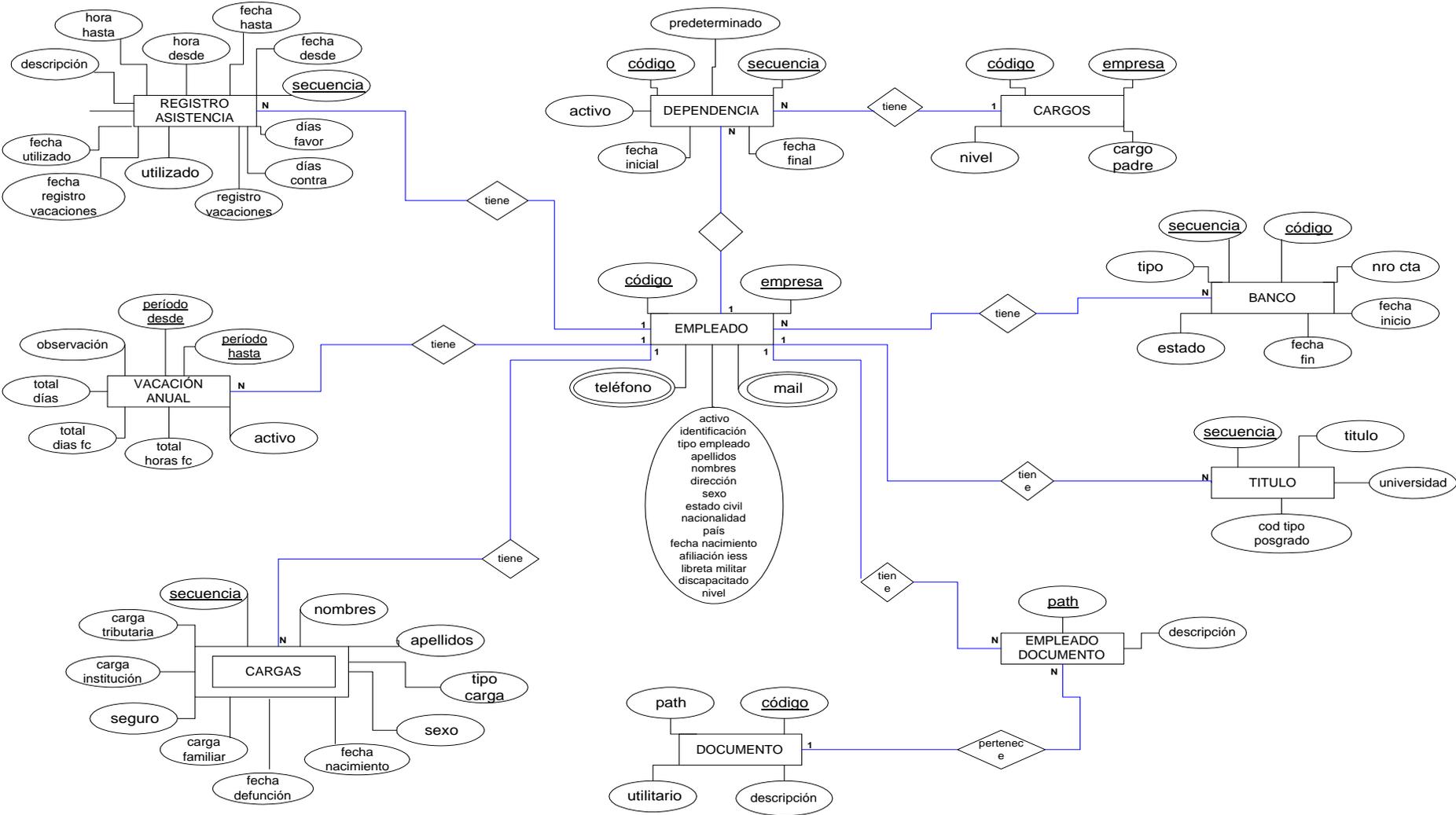
n-1: Las instancias de A pueden asociarse a lo sumo con una de B, y cada instancia de B puede asociarse con 0, 1 o mas instancias de A.

Modalidad:

La modalidad de una relación es 0 si no hay necesidad explícita de que ocurra una relación o de que sea opcional la modalidad es 1 si la ocurrencia de la relación es obligatoria

Entidades dependientes o débiles: En ocasiones ocurre que existen entidades que solo importan si existen otras. Es decir por si solas no son importantes para el sistema.

DIAGRAMA ENTIDAD RELACIÓN



Modelo de Objetos

Describe objetos del sistema y sus relaciones, es un modelo estático; estructura que se organiza como colección de objetos discretos que contienen estructura de datos y su comportamiento.

- D.C. (Diagrama de Clases) → Describe las clases de objetos
- D.D (Diccionario de Datos) → Lista organizada de datos del sistema

Diagrama de Clases

El modelo de objetos captura la estructura estática de sistema, y tiene como propósito presentar a los objetos, relaciones entre ellos y atributos que caracterizan cada clase; se construye en torno a objetos y no a funcionalidad.

Objetos:

- ◆ Es algo que tiene sentido en el contexto de la aplicación
- ◆ Posee su propia identidad y se distinguen entre si
- ◆ Tienen 2 propósitos
 - ◆ Promover comprensión del mundo real
 - ◆ Proporcionar base práctica para implementación por computadora

Clases:

- ◆ Grupo de objetos con propiedades similares, relaciones comunes con otros y semántica común.
- ◆ Describe conjunto de objetos potenciales

Se identifican aquellas entidades o clases que conforman una situación en el análisis y representarlas con objetos genéricos que tengan el nivel de abstracción adecuado que permitan reconocer las características del objeto que se está modelando. Estas entidades poseerán entonces características o atributos que determinarán en forma detallada y que ayudará a relacionarlas con otras entidades del mismo modelo.

Clave Primaria

En todas las entidades es preciso identificar aquel atributo que identifica unívocamente a cada ocurrencia dentro de ella. Es la que llamamos clave principal o primaria. Que se indicará con un asterisco (*) delante del atributo (s) que sean clave (s) principal (es).

Atributos:

- ◆ Valor de dato almacenado en objetos de una clase
- ◆ Cada atributo tiene un valor para cada instancia
- ◆ El nombre del atributo es único en cada clase
- ◆ Son valores puros de datos y no de objetos

Diccionario de datos

El diccionario de datos es una lista organizada de todos los datos pertenecientes al sistema, con una serie de definiciones precisas y rigurosas para que tanto el analista como el usuario comprendan.

Los analistas utilizan los diccionarios de datos por cuatro razones:

1. Para manejar los detalles en sistemas grandes ya que es imposible de recordar todo lo referente a un sistema.
2. Para comunicar un significado común para todos los elementos del sistema.
3. Para documentar las características del sistema.
4. Localizar errores en el sistema.

Contenido de un Diccionario Sistema

1. Definiciones lógicas de datos:

- Elemento de Dato (Atributos de la Entidad).
- Estructura de Dato.

Los elementos de dato se agrupan para formar una estructura de dato.

- Nombre: Son los bloques básicos para todos los demás datos del sistema, por sí solo no lleva ningún significado al usuario. Son los atributos de las entidades.

Ejemplo:

Codigo,Id,Nombre,Apellido,Direccion.....

- Descripción: Es una descripción abreviada de la información incluida en el atributo.
- Tipo: Indica la característica del atributo como el tipo de dato sea carácter, numérico, booleano, etc.
- Tamaño: Indica la cantidad de espacio necesario para cada dato sin considerar la forma en que serán almacenados.
- Primary key(PK): Identifica el atributo unívocamente a cada ocurrencia dentro de la entidad.
- Foreign key(FK): Son las llaves primarias (primary key) de otras entidades que intervienen en esta entidad.
- Tabla Relación: Indica cuales son las tablas donde el almacenamiento de datos de donde ejerce la fuente de datos.

DICCIONARIO DE DATOS

TN_REGISTRO ASISTENCIA

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_empleado	Código del empleado	NUMBER	5	S	S	TN_EMPLEADO
cod_empresa	Código de la empresa	NUMBER	2	S	S	TN_EMPLEADO
Secuencia	Secuencia de la asistencia	NUMBER	5	S		ninguna
Descripcion	Descripción	VARCHAR2	100			ninguna
fecha_desde	Fecha Desde	DATE				ninguna
fecha_hasta	Fecha Hasta	DATE				ninguna
hora_desde	Hora Desde	VARCHAR2	5			ninguna
hora_hasta	Hora Hasta	VARCHAR2	5			ninguna
cod_justificativo	Código del justificativo	NUMBER	3		S	TN_JUSTIFICATIVO
dias_favor_contra	Número de días a favor o contra(-)	NUMBER	2			ninguna
hora_favor_contra	Número de horas a favor o contra(-)	NUMBER	2			ninguna
Justificado(s/n)	Si está o no justificado	VARCHAR2	1			ninguna
Registrado_para_vacaciones	Si está registrado o sin registrar	VARCHAR2	1			ninguna
fecha_registro	Fecha del registro	DATE				ninguna
Utilizado	Si está utilizado o no	VARCHAR2	1			ninguna
fecha_utilizado	Fecha Utilizada	DATE				ninguna
Observacion	Observaciones	VARCHAR2	200			ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
justificado	S	justificado
	N	sin justificar
registrado_para_vacaciones	S	registrado
	N	sin registrar
Utilizado	S	si se ha utilizado
	N	no se ha utilizado

TN_DOCUMENTO

NOMBRE	DESCRIPCION	TIPO	TAMAÑO	PK	FK	TABLA RELACION
cod_documento	Código del documento	NUMBER	3	S	S	TN_DOCUMENTO
Descripcion	Descripción del documento	VARCHAR2	100			ninguna
Utilitario	Utilitario usado	VARCHAR2	50			ninguna
Path	Ruta de acceso al programa ejecutable	VARCHAR2	100			ninguna

TN_EMPLEADO_DOCUMENTO

NOMBRE	DESCRIPCION	TIPO	TAMAÑO	PK	FK	TABLA RELACION
cod_empleado	Código del empleado	NUMBER	5	S	S	TN_EMPLEADO
cod_empresa	Código de la empresa	NUMBER	2	S	S	TN_EMPLEADO
cod_documento	Código del documento	NUMBER	3	S	S	TN_DOCUMENTO
Path	Nombre + extensión del documento	VARCHAR2	100	S		ninguna
Descripcion	Descripción	VARCHAR2	100			ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
Cod_documento	1	MICROSOFT WORD
	2	MICROSOFT EXCEL
	3	INTERNET EXPLORER

TN_EMPLEADO_DEPENDENCIA

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_empleado	Código del empleado	NUMBER	5	S	S	TN_EMPLEADO
cod_empresa	Código de la empresa	NUMBER	2	S	S	TN_EMPLEADO
cod_secuencia	Código de la secuencia	NUMBER	4	S		ninguna
cod_division_dependencia	Código de división de la dependencia	VARCHAR2	10	S		ninguna
fecha_inicial	Fecha Inicial	DATE				ninguna
fecha_final	Fecha Final	DATE				ninguna
Activo	Si está o no activo	NUMBER	1			ninguna
Predeterminado	Si está o no predeterminado	NUMBER	1			ninguna
cod_cargo	Código del cargo	NUMBER	4	S		ninguna
cod_cargo_padre	Código del cargo padre	NUMBER	4			ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
Activo	1	estado activo
	0	estado inactivo
Predeterminado	1	está predeterminado
	0	no esta predeterminado
Cod_division_dependencia	50	Dir. Escuela Economia
	52	Fac.Filosofia
	156	Fac. Medicina
cod_cargo	20	Director Ejecutivo
	21	Asistente Administrativo
	1	Rector
	66	Laboratorista

En el caso de cod_division_dependencia y cod_cargo ponemos solo alguno a manera de ejemplo pero son muchos mas

TN JUSTIFICATIVO

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_justificativo	Código del justificativo	NUMBER	3	S		ninguna
Descripcion	descripción del justificativo	VARCHAR2	50			ninguna

TN VACACION ANUAL

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_empleado	Código del empleado	NUMBER	5	S	S	TN_EMPLEADO
cod_empresa	Código de la empresa	NUMBER	2	S	S	TN_EMPLEADO
periodo_desde	Empiezan las vacaciones	DATE		S		ninguna
periodo_hasta	Terminan las vacaciones	DATE		S		ninguna
observaciones	Observaciones varias	VARCHAR2	100			ninguna
total_dias	Nro total de días que dispone para vacaciones	NUMBER	2			ninguna
total_dias_fc	Nro total de días a favo o en contra	NUMBER	3			ninguna
total_horas_fc	Nro total de horas a favo o en contra	NUMBER	3			ninguna
activo	estado del período de vacaciones	VARCHAR2	1			ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
Activo	S	estado activo
	N	estado inactivo

PE_ASISTENCIA_DOCENTE

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_empleado	Código del empleado	NUMBER		11	S	ninguna
secuencia_marca	Secuencia de marcado	NUMBER		9	S	ninguna
fecha_marca	Fecha de marcado	DATE				ninguna
cod_periodo	Período al que corresponde	NUMBER		4		ninguna
tipo_marca	Entrada o Salida	VARCHAR2		1		ninguna
hora_marca	Hora de marcado	NUMBER		2		ninguna
minuto_marca	Minuto de marcado	NUMBER		2		ninguna
estado	Normal o Anulado	VARCHAR2		1		ninguna
cod_clase	Código de la clase	NUMBER		8		ninguna
docente_alumno	Docente o Alumno	VARCHAR2		1		ninguna
adicionado_por	Asistencia o Nomina	VARCHAR2		10		ninguna
fecha_adicion	Fecha de adición	DATE				ninguna
modificado_por	Modificado	VARCHAR2		10		ninguna
fecha_modificacion	Fecha de modificación	DATE				ninguna
observaciones	Observaciones	LONG				ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
Tipo_marca	S	salida
	E	entrada
Estado	N	normal
	A	anulado
Docente_alumno	D	docente
	A	alumno

PE_ASISTENCIA_DOCENTE_H

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_empleado	Código del empleado	NUMBER	11	S		ninguna
secuencia_marca	Secuencia de marcado	NUMBER	9	S		ninguna
fecha_marca	Fecha de marcado	DATE				ninguna
cod_periodo	Período al que corresponde	NUMBER	4			ninguna
tipo_marca	Entrada o Salida	VARCHAR2	1			ninguna
hora_marca	Hora de marcado	NUMBER	2			ninguna
minuto_marca	Minuto de marcado	NUMBER	2			ninguna
estado	Normal o Anulado	VARCHAR2	1			ninguna
cod_clase	Código de la clase	NUMBER	8	S		ninguna
docente_alumno	Docente o Alumno	VARCHAR2	1			ninguna
adicionado_por	Asistencia o Nomina	VARCHAR2	10			ninguna
fecha_adicion	Fecha de adición	DATE				ninguna
modificado_por	Modificado	VARCHAR2	10			ninguna
fecha_modificacion	Fecha de modificación	DATE				ninguna
observaciones	Observaciones	LONG				ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
Tipo_marca	S	salida
	E	entrada
Estado	N	normal
	A	anulado
Docente_alumno	D	docente
	A	alumno

TN_EMPLEADO

NOMBRE	DESCRIPCIÓN	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_empleado	Código del empleado	NUMBER		5	S	ninguna
cod_empresa	Código de la empresa	NUMBER		2	S	TN_TIPO_EMPLEADO
cod_tipo_empleado	Docentes, Administrativos.....	NUMBER		2	S	TN_TIPO_EMPLEADO
cod_tipo_identificacion	Cédula o Ruc	NUMBER		1		ninguna
identificacion	Número de identificación	VARCHAR2		20		ninguna
apellido_paterno	Apellido Paterno	VARCHAR2		15		ninguna
apellido_materno	Apellido Materno	VARCHAR2		15		ninguna
primer_nombre	Primer Nombre	VARCHAR2		15		ninguna
segundo_nombre	Segundo Nombre	VARCHAR2		15		ninguna
sexo	Femenino, Masculino	VARCHAR2		1		ninguna
estado_civil	Casado, Soltero,	VARCHAR2		2		ninguna
cod_nacionalidad	Código de nacionalidad	VARCHAR2		3		ninguna
cod_pais	Código del país	VARCHAR2		3		ninguna
nivel	Nivel del país	NUMBER		1		ninguna
codigo	Código de la ciudad	VARCHAR2		6		ninguna
direccion	Dirección Calle	VARCHAR2		80		ninguna
direccion_numero	Dirección Número	VARCHAR2		10		ninguna
fecha_nacimiento	Fecha de nacimiento	DATE				ninguna
afiliacion_ies	Número de afiliación IESS	VARCHAR2		12		ninguna
libreta_militar	Número de libreta militar	VARCHAR2		12		ninguna
activo	Activo o no	VARCHAR2		1		ninguna
discapacitado	Discapacitado o no	VARCHAR2		1		ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
Cod_tipo_empleado	1	docentes
	2	administrativos
	3	autoridades
	4	por contrato
Cod_tipo_identificacion	1	cedula
	2	ruc
Sexo	F	femenino
	M	masculino
Estado_civil	SO	soltero
	CA	casadp
	DI	divorciado
	VI	viudo
	UL	union libre
Activo	S	estado activo
	N	estado inactivo
Discapacitado	S	discapacitado
	N	no discapacitado

TN_EMPLEADO_BANCO

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_empleado	Código Empleado	NUMBER		5 S	S	TN_EMPLEADO
cod_empresa	Código Empresa	NUMBER		2 S	S	TN_EMPLEADO
secuencia	Secuencia Banco	NUMBER		2 S		ninguna
cod_banco	Código Banco	NUMBER		3 S		ninguna
nro_cuenta	Número de la cuenta	VARCHAR2		20		ninguna
fecha_inicio	Fecha inicio cuenta	DATE				ninguna
fecha_final	Fecha final cuenta	DATE				ninguna
tipo	Ahorros o Corriente	VARCHAR2		1		ninguna
estado_activo	Activo o Inactivo	NUMBER		1		ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
Cod_banco	1	BANCO DEL AUSTRO
	2	BANCO DEL PICHINCHA
	7	PRODUBANCO
	19	BANCO DE MACHALA
	5	BANCO DEL PACIFICO
	6	BANCO DEL PICHINCHA
	22	UNIBANCO
	24	VISA AUSTRO
	10	BANCO INTERNACIONAL
	Tipo	A
C		CORRIENTE
Estado_activo	1	ACTIVO
	0	INACTIVO

En el caso de Cod_banco mostramos algunos de los codigo existentes

TN EMPLEADO CARGA

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_empleado	Código Empleado	NUMBER		5 S	S	TN_EMPLEADO
cod_empresa	Código Empresa	NUMBER		2 S	S	TN_EMPLEADO
secuencia	Secuencia empleado carga	NUMBER		2 S		ninguna
tipo_carga	Tipo de carga	VARCHAR2		2		ninguna
apellidos	Apellidos	VARCHAR2		60		ninguna
nombres	Nombres	VARCHAR2		60		ninguna
sexo	Femenino, Masculino	VARCHAR2		1		ninguna
fecha_nacimiento	Fecha Nacimiento	DATE				ninguna
fecha_defuncion	Fecha Defunción	DATE				ninguna
carga_tributaria	Si o no	VARCHAR2		1		ninguna
carga_institucion	Si o no	VARCHAR2		1		ninguna
carga_familiar	Si o no	VARCHAR2		1		ninguna
seguro	Si o no	VARCHAR2		1		ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
Tipo_carga	ES	Esposa
	HI	Hijo
	PC	Padres
Sexo	F	Femenino
	M	Masculino
Carga_tributaria	S	Si
	N	No
Carga_institucion	S	Si
	N	No
Carga_familiar	S	Si
	N	No
Seguro	S	Si
	N	No

TN_EMPLEADO_TITULO

NOMBRE	DESCRIPCION	TIPO	TAMAÑO	PK	FK	TABLA RELACION
cod_empleado	Código Empleado	NUMBER		5 S		ninguna
secuencia	Secuencia empleado titulo	NUMBER		2 S		ninguna
cod_tipo_posgrado	Código tipo posgrado	NUMBER		2	S	COD_TIPO_TITULO
titulo	Nombre del titulo	VARCHAR2		2		ninguna
universidad_adq	Universidad adquirida	VARCHAR2		60		ninguna
fecha	Fecha en la que adquirió	VARCHAR2		60		ninguna
cod_empresa	Código Empresa	VARCHAR2		1 S		ninguna
nro_afiliacion	Número de Afiliación	DATE				ninguna
cod_carrera	Código de la carrera	DATE			S	COD_AREA_CARRERA

TN_EMPLEADO_TELEFONO

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_empleado	Código Empleado	NUMBER		5 S	S	TN_EMPLEADO
cod_empresa	Código Empresa	NUMBER		2 S	S	TN_EMPLEADO
secuencia	Secuencia empleado teléfono	NUMBER		2 S		ninguna
telefono	Número telefonico	VARCHAR2		2		ninguna

TN_EMPLEADO_MAIL

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_empleado	Código Empleado	NUMBER		5 S		ninguna
cod_empresa	Código Empresa	NUMBER		2 S		ninguna
mail	Dirección de correo electrónico	VARCHAR2		40		ninguna
si_no	Si tiene o no mail	NUMBER		1		ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
Si_no	1	Con dirección
	0	Sin dirección

TNA_REGISTRO

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_empleado	Código Empleado	NUMBER		5 S		ninguna
cod_horario	Código Horario	NUMBER		4 S		ninguna
fecha	Fecha	DATE		S		ninguna
dia	Día	NUMBER		1 S		ninguna
secuencia	Secuencia Registro	NUMBER		1 S		ninguna
hora_e	Hora Entrada	NUMBER		4		ninguna
hora_s	Hora Salida	NUMBER		4		ninguna
d_hora_e	Día Hora Entrada	NUMBER		4		ninguna
d_hora_s	Día Hora Salida	NUMBER		4		ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
Cod_horario	1	Horario 1
	2	Horario 2
	3	Horario 3
Dia	2	Lunes
	3	Martes.....
	6	Viernes
	7	Sábado

TNA_REGISTRO_H

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_empleado	Código Empleado	NUMBER		5	S	ninguna
cod_horario	Código Horario	NUMBER		4	S	ninguna
fecha	Fecha	DATE			S	ninguna
dia	Día	NUMBER		1	S	ninguna
secuencia	Secuencia Registro Histórico	NUMBER		1	S	ninguna
hora_e	Hora Entrada	NUMBER		4		ninguna
hora_s	Hora Salida	NUMBER		4		ninguna
d_hora_e	Día Hora Entrada	NUMBER		4		ninguna
d_hora_s	Día Hora Salida	NUMBER		4		ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
Cod_horario	1	Horario 1
	2	Horario 2
	3	Horario 3
Dia	2	Lunes
	3	Martes.....
	6	Viernes
	7	Sábado

TN_CARGO

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_cargo	Código Cargo	NUMBER		4	S	ninguna
cod_empresa	Código Empresa	NUMBER		2	S	TN_CARGO
cod_nivel	Código Nivel	NUMBER		2		ninguna
nombre	Nombre del Cargo	VARCHAR2		50		ninguna
cod_cargo_padre	Código cargo padre	NUMBER		4	S	TN_CARGO
salario_nomina	Salario nomina	NUMBER		14.2		ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
Cod_cargo	1	RECTOR
	25	SECRETARIA EJECUTIVA
	116	AUDITOR
	31	DIRECTOR RR-HH
	32	ANALISTA DE PERSONAL
	34	DIRECTOR FINANCIERO
	58	RECEPCIONISTA
	66	LABORATORISTA

TN_PARAMETRO

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_parametro	Código Parámetro	NUMBER		4	S	ninguna
cod_empresa	Código Empresa	NUMBER		2	S	ninguna
cod_parametro_alterno	Código Parámetro Alterno	VARCHAR2		30		ninguna
nombre	Nombre del Parámetro	VARCHAR2		50		ninguna
tipo_valor	Ingresados, Variar.....	VARCHAR2		1		ninguna
valor	Valor del Parámetro	VARCHAR2		255		ninguna
observaciones	Observaciones	VARCHAR2		100		ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
Tipo_valor	I	INGRESADOS
	V	VARIAR
	N	INAMOVIBLES

TN_HORARIO_DETALLE

NOMBRE	DESCRIPCION	TIPO	TAMANO	PK	FK	TABLA RELACION
cod_horario	Código Horario	NUMBER		4 S	S	TNA_HORARIO
dia	Día	NUMBER		1 S		ninguna
secuencia	Secuencia horario detalle	NUMBER		2 S		ninguna
hora_e	Hora Entrada	NUMBER		5		ninguna
hora_s	Hora Salida	NUMBER		5		ninguna

VALOR DISCRETO

ATRIBUTO	VALOR	SIGNIFICADO
Cod_horario	1	Horario 1
	2	Horario 2
	3	Horario 3
Dia	2	Lunes
	3	Martes
	4	Miercoles
	5	Jueves
	6	Viernes
	7	Sábado

3.3 Diseño.

Proceso de diseño.- el objetivo del diseño es proporcionar representaciones del *software* en las que se pueda valorar la calidad. Es un proceso en el que se aplican técnicas para definir un producto *software*.

En el diseño encontraremos:

- a) Diseño de datos: transformar información del dominio aplicación en estructuras de datos. *DD, *DER.

- b) Diseño arquitectónico: Muestra como está estructurado el *software*, y la relación entre elementos estructurales del programa. *DFD, *DFC.

- c) Diseño de interfaz: como se va a comunicar el software con el usuario y también como se comunica consigo mismo. *DFD.

- d) Diseño procedimental: transformar en programas (descripción procedimental) los elementos estructurales.

Técnicas para definir un diseño:

- ◆ Diseño de presentar organización jerárquica,
- ◆ Diseño debe contener abstracción de datos y procedimental,
- ◆ Diseño debería producir módulos que representen características funcionales independientes.
- ◆ Se debe producir diseño utilizando método que sea repetido usando un estandar.

Principios para un diseño:

- ◆ Considerar enfoques alternativos (recursos disponibles y requisitos),
- ◆ Diseño debe basarse en análisis,
- ◆ Debe minimizar distancia entre software y problema,
- ◆ Debe presentar uniformidad o integración,
- ◆ Programar no es diseñar y diseñar no es programar,
- ◆ Se debe valorar calidad mientras se diseña no al terminar,

Diseño modular efectivo.

- ◆ Independencia funcional.- tratar de crear módulos que desarrollen única función con interfaz sencilla.
- ◆ Cohesión.- módulo realice una sola tarea dentro de un procedimiento con poca interacción con otros.
 - ✓ Coh. Lógica.- realiza tarea relacionadas lógicamente.
 - ✓ Coincidentalmente cohesivo.- tareas pocas relacionadas unas con otras.
 - ✓ Coh. Temporal.- tareas relacionadas porque deben hacerse en mismo espacio de tiempo.
 - ✓ Coh. Procedimental.- elementos de procesamiento están relacionados y deben ejecutarse en un orden específico.
 - ✓ Coh. De comunicación.- elementos de procesamiento se concentran en una área de estructura de datos.
- ◆ Acoplamiento.- medida de interconexión entre módulos de la estructura de programas.
 - ✓ Acop. De datos.- se pasan datos simples.
 - ✓ Acop. De marca.- se pasa porción, estructura más compleja.
 - ✓ Acop. De control.- se pasa indicador de control entre módulos.
 - ✓ Acop. Común.- módulo hace referencia a área global de datos.
 - ✓ Acop. De contenido.- hace uso de datos o información de control mantenidos dentro de otro módulo.
- ◆ Abstracción.- permite concentrarse en un problema a un nivel de generalización independientemente de los detalles de nivel inferior.
 - ✓ Abs. Procedimental: es una secuencia dada de instrucciones que tienen una función específica y limitada.
 - ✓ Abs. de datos: es una colección determinada de datos que describen un objeto de datos.
 - ✓ Abs. de control: implica un mecanismo de control del programa sin especificar detalles internos.

- ◆ Modularidad.- divide el software en componentes identificables y tratables por separado denominados módulos.
 - ✓ Capacidad de descomposición modular,
 - ✓ Capacidad empleo componentes modulares – reutilización,
 - ✓ Capacidad de comprensión modular – facilita construcción y cambios,
 - ✓ Continuidad modular – minimiza impacto de cambios.

3.4 Programación.

La aplicación realizada en el lenguaje de programación *PL/SQL* y procedimientos en *FORMS*, el cual es un lenguaje de cuarta generación, está orientada a eventos, o sea, el código escrito en las aplicaciones responde siempre que se ejecute una acción y todos los objetos se encuentran ligados al lenguaje de programación utilizado, por lo tanto contiene código con propiedades y métodos de objetos, que actúen según lo esperado.

3.5 Pruebas

Las pruebas del Sistema deben ser realizadas una *PC* conectada en red por el puerto *TCP/IP* de *Windows* con acceso al servidor de datos de la Universidad, y con las licencias instaladas para la utilización de la base de datos *Oracle*, de tal manera de acceder a la información y realizar las pruebas correspondientes al funcionamiento y funcionalidad de la aplicación.

CONCLUSIONES

Teóricas

La implementación del Sistema de Archivo Virtual, desarrollado para el Departamento de Recursos Humanos de la Universidad del Azuay, brinda una completa información llevada de una manera totalmente automatizada fácil y rápida, contiene fichas con todos los datos de los empleados y su relación laboral; la aplicación permite consultar, obtener reportes y controla la variación de los estados de los empleados.

Metodológicas

El análisis realizado se basó básicamente en el dominio de la información del software para poder comprender la naturaleza de los programas realizados, estableciendo restricciones, creando nuevas tablas necesarias para el contenido de la información y el flujo de datos.

El desarrollo de la aplicación está basado en el análisis y modelado del sistema, para seguir al paso de la programación realizada en lenguaje *PL/SQL*, el cuál permite el manejo de la base de datos *Oracle* y procedimientos en el *Forms 6.0*, el cuál interactúa de manera directa con la base de datos.

El método de programación orientado a eventos, debido a que es el tipo de programación que soporta *Forms 6.0*.

Para la conexión con el servidor de datos, se utilizó el sistema de comunicación estándar que viene con *Windows(TCP/IP)*

Pragmáticas

El Sistema de Archivo Virtual es un avance al que todas las empresas deben llegar, por la agilidad en la que se maneja la información, y la capacidad de almacenamiento con la que cuenta la base de datos, permitiendo obtener consultas históricas, tener los datos organizados y en orden.

Como ventajas, podemos destacar el ahorro de espacio físico y de papel ya que los archivos son almacenados en forma magnética, a mas de la agilidad, y una completa funcionalidad para las necesidades del departamento.

RECOMENDACIONES

El personal es el principal recurso de toda empresa, por lo que el Departamento de Recursos Humanos debe llevar la información de forma organizada rápida y segura; y esto se logró por la implementación del sistema manejando la información en forma automatizada, por lo que todas las empresas deberían crear un archivo virtual de acuerdo a sus propias necesidades.

Una vez implementado el sistema y ya en uso se recomienda dar un mantenimiento, cada que se realice una actualización de datos, sacando respaldos del disco duro en donde se encuentra la unidad virtual de los archivos del sistema, ya que la información se maneja en archivos magnéticos.

Acostumbrarse a almacenar todo documento digital en el sistema, a más de que podemos acceder con facilidad a ellos, también representa un ahorro de papel.

BIBLIOGRAFÍA

Capítulo I:

www.uazuay.edu.ec

- 1 Reseña Histórica.htm
- 2 Misión, visión, principios.htm

Capítulo II:

- 3 www.hisp@zone.com

Tutoriales Informática – Los archivos tipos, extensiones y programas para su uso.htm

www.monografias.com

- 4 Tipos de extensiones_com.htm
- 5 Formatos de imagen.htm
- 5 Formatos gráficos.htm

www.wikipedia.com

Windows bitmap

Capítulo III:

- 6 www.monografias.com/trabajos25/oracle/oracle.shtml

OSBORNE/McGRAW-HILL/Interamericana de España, S.A.

- 7 Oracle 7.0. Manual de referencia, 1ra Edición año 2000
- 7 Forms 6.0, Manual de referencia, 1ra Edición año 2000
- 7 PL/SQL Developer 4.0, Manual de referencia, 2da. Edición año 2002

- 8 ERAZO Juan Carlos: “Curso básico de developer forms”
Cali, Junio 2002.

- 9 www.zonaoracle.com

- 10 ORACLE: Introducción a ORACLE 9i: PL/SQL. Student Guide. 2001
“practica8.pdf”

ANEXOS

MANUAL DEL USUARIO

(Véase archivo adjunto **ManualDelUsuario.pdf**)

DISEÑO DE TESIS

(Véase archivo adjunto **DiseñoTesis.pdf**)