



SEDE GUAYAQUIL

FACULTAD DE INGENIERIA EN SISTEMAS

**CARRERA: INGENIERIA EN SISTEMAS CON
MENCIÓN EN GESTION ADMINISTRATIVA**

**PROYECTO DE GRADO PREVIO A LA OBTENCIÓN
DEL TÍTULO DE: INGENIEROS EN SISTEMAS**

TEMA:

**“SISTEMA INTEGRADO PARA LA
AUTOMATIZACIÓN DE UN LABORATORIO CLÍNICO
ORIENTADO A LA WEB”**

AUTORES:

Marvin Cristóbal Alvarez Cedeño

Richard Vinicio Astudillo Sarmiento

Alex Daniel Zambrano Herrera

DIRECTOR: Ing. Raúl Álvarez

Guayaquil, 30 de Junio de 2010

DECLARATORIA DE RESPONSABILIDAD

Este documento contiene información veraz en el cual los conceptos desarrollados, análisis realizados y conclusiones efectuadas en el presente trabajo de tesis, son de exclusiva responsabilidad de los autores.

Guayaquil, Noviembre 10 del 2010

Richard Vinicio Astudillo Sarmiento

Marvin Cristóbal Alvarez Cedeño

Alex Daniel Zambrano Herrera

DEDICATORIA

Con gran aprecio dedicamos este proyecto a Dios nuestro padre Celestial a quien le debemos todo lo que hemos logrado a lo largo de nuestra vida.

Y a nuestros padres quienes han sido nuestro gran apoyo y nuestra fuente inestimable de inspiración.

ÍNDICE

1.	INTRODUCCIÓN	1
2.	JUSTIFICACIÓN.....	2
3.	PLANTEAMIENTO DEL PROBLEMA	3
4.	RESUMEN DE LA PROPUESTA DE INTERVENCIÓN.....	4
4.1.	Descripción general del sistema	5
4.2.	El sistema.....	5
5.	OBJETIVOS	6
5.1.	General	6
5.2.	Específicos	6
6.	BENEFICIARIOS DE LA PROPUESTA	7
6.1.	Beneficios en el campo de la salud	7
6.2.	Beneficios en el mercado	8
6.3.	Beneficios en la población	8
7.	DELIMITACIÓN.....	9
7.1.	Espacial.....	9
7.2.	Temporal	9
7.3.	Académica	10
8.	MARCO REFERENCIAL.....	11
8.1.	Marco teórico	11
8.1.1	Análisis y diagramación de procesos empresariales.....	11
8.1.2	Análisis y diseño de sistemas orientado a objetos.....	18
8.1.3	Administración de base de datos Oracle 10g	31
8.1.4	Programación en PL/SQL Oracle	31
8.1.5	Oracle forms developer 10g	50
8.1.6	Estándares para desarrollo.....	56
8.2.	Marco conceptual.....	63
9.	Desarrollo de proyecto	67
9.1	Análisis y diseño de sistema	67
9.1.1.	Mapa de procesos del laboratorio clínico QUICKLAB	675

9.1.2	Cuadro PAT del laboratorio clínico QUICKLAB.....	68
9.1.3	Mapa causa efecto de un análisis clínico	69
9.1.4	Diagrama de casos de uso	70
9.1.5	Diagrama de clases laboratorio clínico.....	75
9.1.6	Diagrama de estado	75
9.1.7	Diagrama de secuencia.....	76
9.1.8	Gestión administrativa.....	77
9.1.9	Diagrama entidad relación del laboratorio clínico.....	78
9.2	Desarrollo del sistema.....	79
9.2.1	DISEÑO LOGICO DEL SISTEMA.....	79
9.2.1.1	Tablas del sistema quicklab.....	79
9.2.2	Documentación para el Usuario.....	135
9.2.2.1	Introducción.....	135
9.2.2.2	Requerimientos del sistema	135
9.2.2.3	Opciones del sistema	136
9.2.2.3.1	INGRESO AL SISTEMA	139
9.2.2.3.2	MANTENIMIENTO	139
9.2.2.3.3	PROCESOS	164
9.2.2.3.4	REPORTES	175
9.2.2.3.5	AYUDA	176
9.2.2.3.6	SALIR.....	176
9.2.3	ADMINISTRACIÓN DE USUARIOS.....	177
9.2.3.1	USUARIOS DE BASE DE DATOS.....	177
9.2.3.2	USUARIOS DEL SISTEMA.....	177
9.2.3.3	DEFINICIÓN DE ROLES	177
9.2.3.4	CREACIÓN DE USUARIOS.....	177
10.	CONCLUSIONES Y RECOMENDACIONES	189
10.1.	CONCLUSIONES	189
10.2.	RECOMENDACIONES	191
10.3.	BIBLIOGRAFIA	191
ANEXOS		195
ANEXO 1:	ESTRUCTURA DE FUNCIONAMIENTO DEL SISTEMA	196

INDICES DE IMÁGENES

FIG. N° 1: BOSQUEJO DE PROCESO DE QUICKLAB	5
FIG N° 2: MODELO DE CADENA DE VALOR	11
FIG. N°3: DIAGRAMA DE FLUJO SENCILLO CON LOS PASOS A SEGUIR SI UNA LÁMPARA NO FUNCIONA	17
FIG N° 4: EJEMPLO DE CLASES	21
FIG. N° 5: EJEMPLOS DE HERENCIA	22
FIG N° 6: EJEMPLOS DE POLIMORFISMO	23
FIG. N° 7: SIMBOLOGÍA PARA REALIZAR LOS DIAGRAMAS DE CLASES	25
FIG N° 8: EJEMPLOS DE AGREGACIÓN	25
FIG. N° 9: SÍMBOLOS PARA REPRESENTACIÓN GRÁFICA DE CASOS DE USO	26
FIG.N° 10: EJEMPLOS DE DIAGRAMAS DE CASOS DE USO	27
FIG. N°11: SIMBOLOGÍA PARA REALIZAR DIAGRAMAS DE ESTADOS	28
FIG. N°12: EJEMPLO DE DIAGRAMAS DE ESTADO	29
FIG. N°13: EJEMPLO DE DIAGRAMA DE SECUENCIA	30
FIG. N°14: ARQUITECTURA DE ORACLE	32
FIG N°. 15: ARQUITECTURA DE LA INSTANCIA DE ORACLE	33
FIG N°. 16: ESTRUCTURA DE PROCESO	38
FIG N°17: FASES PARA PROCESAR UNA INSTRUCCIÓN SQL	42
FIG N° 18: ARQUITECTURA DE ORACLE APLICACIÓN SERVER	51
FIG. N°. 19: COMPONENTES DE ORACLE DEVELOPER SUITE	51
FIG. N° 20: ELEMENTO DE UN REPORTE	55
Fig. N° 21: FORMULARIO DE FACTURA	58
Fig. N° 22: MAPA DE PROCESO QUICKLAB	67
Fig. N° 23: DIAGRAMA PAT QUICKLAB	68
Fig. N° 24: DIAGRAMA CAUSA EFECTO	69
Fig. N° 25: DIAGRAMA CASOS DE USO	70

LABORATORIO CLINICO

Fig. N° 26: DIAGRAMA CASOS DE USO FACTURACION		71
Fig. N° 27: DIAGRAMA CASOS DE USO CUENTAS POR COBRAR		72
Fig. N° 28: DIAGRAMA CASOS DE USO CAJA		73
Fig. N° 29: DIAGRAMA CLASES		74
Fig. N° 30: DIAGRAMA ESTADOS		75
Fig. N° 31: DIAGRAMA SECUENCIA LABORATORIO CLINICO		76
Fig. N° 32: DIAGRAMA SECUENCIA GESTION ADMINISTRATIVA		77
Fig. N° 33: DIAGRAMA ENTIDAD RELACION		78

INDICE DE TABLAS

Tabla N°. 1: CATALOGOTABLA	79
Tabla N°. 2: PACIENTE	80
Tabla N°. 3: TEST	81
Tabla N°. 4: ORDENES	83
Tabla N°. 5: RESULTADOEQUIPO	84
Tabla N°. 6: RANGODETALLE	85
Tabla N°.7: RANGO	86
Tabla N°. 8: MENU	86
Tabla N°. 9: ORDENDETALLE	87
Tabla N°. 10: MEDICO	88
Tabla N°. 11: INTERFAZEQUIPO	89
Tabla N°. 12: ETAPAHUMANA	90
Tabla N°. 13: EQUIPO	91
Tabla N°. 14: EMPRESA	92
Tabla N°. 15: CATALOGO	93
Tabla N°. 16: ACCESO	94
Tabla N°. 17: USUARIO	95
Tabla N°. 18: CLIENTE	96
Tabla N°. 19: DOCUMENTO	97
Tabla N°. 20: DOCUMENTODETALLE	98
Tabla N°. 21: FACTURA	99
Tabla N°. 22: FACTURADETALLE	100
Tabla N°. 23: PRODUCTO	101
Tabla N°. 24: VENDEDOR	102
Tabla N°. 25: CUENTAXCOBAR	103
Tabla N°. 26: FORMA_PAGO	104
Tabla N°. 27: TIPOMOVIMIENTO	105
Tabla N°. 28: TRXCAJA	106
Tabla N°. 29: TRXCAJADETALLE	107

INDICES DE PANTALLAS

PANTALLA N°1 :PRESENTACION	136
PANTALLA N°2 : LOGIN	137
PANTALLA N°3 : MENU	138
PANTALLA N°4 : MANTENIMIENTO VENDEDOR	139
PANTALLA N°5 : MANTENIMIENTO OPCIONES DE MENU	140
PANTALLA N°6 : MANTENIMIENTO ACCESOS	143
PANTALLA N°7 : MANTENIMIENTO EMPRESA	145
PANTALLA N°8 : MANTENIMIENTO CLIENTE	147
PANTALLA N°9 : MANTENIMIENTO PACIENTE	149
PANTALLA N°10 : MANTENIMIENTO MEDICO	152
PANTALLA N°11 : MANTENIMIENTO ETAPA HUMANA	156
PANTALLA N°12 : MANTENIMIENTO VENDEDOR	158
PANTALLA N°13 : MANTENIMIENTO RANGOS	160
PANTALLA N°14: MANTENIMIENTO TEST	161
PANTALLA N°15 : MANTENIMIENTO USUARIO	162
PANTALLA N°16 : SOLICITUD DE EXAMEN	164
PANTALLA N°17 : FACTURACION	169
PANTALLA N°18 : CAJA	173
PANTALLA N°19 : INGRESO DE RESULTADO	174
PANTALLA N°20 : REPORTE DE ORDENES DE CLIENTE	175
PANTALLA N°21 : REPORTE DE CUADRE DE CAJA	176

1. INTRODUCCIÓN

La implementación de los primeros sistemas informáticos en la mayor parte de centros hospitalarios fue en los laboratorios de análisis clínicos a finales de la década de los 70. La cantidad de datos analíticos que trata un laboratorio general era, ya entonces, lo suficientemente grande para exigir procesos mecanizados de tratamiento de los datos, en paralelo a las exigencias de automatización de los propios procesos analíticos.

En un principio, los objetivos de la informatización eran muy simples: organización del trabajo interno y emisión de dictámenes mecanografiados. Las etapas intermedias eran la entrada de peticiones al ordenador, edición de listas de trabajo para las distintas tareas manuales, conexiones on line con los principales auto analizadores, y los procesos de validación técnica y clínica.

Debido a la necesidad que encontramos en el stakeHolder nos orientamos a desarrollar un **“sistema integrado para la automatización de un laboratorio clínico orientado a la web”**, en el mundo globalizado en que vivimos hay que ofrecer soluciones integrales, eficientes y eficaces que permitan estar conectados sin importar el lugar, tiempo y distancia, una de las características del sistema es que los médicos y usuarios que estén registrados en la base de datos tendrán la ventaja de recibir vía correo electrónico un mensaje de texto, para indicarle al doctor y al paciente que están listos los resultados de los análisis y los puede revisar utilizando el internet, ingresando al sitio web, el paciente debe tener cancelado su orden clínica, para que tenga acceso a dichos resultados.

El sistema integrado vía web es una solución integral que brindará todos los controles necesarios para administrar de una forma eficaz y eficiente, el laboratorio clínico y así alcanzar los estándares de calidad y competitividad que exige la gestión clínica; el sistema nos brindara:

- Mayor automatización.
- Inclusión de técnicas nuevas con mayor complejidad de tratamiento de la información.
- Nuevas exigencias de calidad y protocolización.
- Consideración de la gestión administrativa como tarea básica del laboratorio.
- Integración de datos clínicos.

2. JUSTIFICACIÓN

En la actualidad las PYMES (laboratorios Clínicos) carecen de un sistema informático de apoyo para el control de la información y obtener de manera precisa y a tiempo los resultados requeridos, es por esta razón que planteamos la siguiente solución, un **“Sistema integrado para la automatización de un laboratorio clínico orientado a la web”**, este sistema permitirá el apoyo operacional y el control de la gestión administrativa integral de un laboratorio de análisis clínicos.

El sistema cuenta con un acceso vía web, el mismo que tiene un modulo de seguridad, en el cual se administrara las cuentas y las claves necesarias para el acceso al sistema, con esto podemos transparentar los procesos de cada usuario.

En la actualidad se requiere de sistemas que no se limiten a un lugar físico, tenemos una nueva generación de sistemas de información de laboratorios donde se integra la experiencia de los profesionales de la salud así como expertos en tecnología de información para obtener un sistema que sea flexible, fácil de utilizar y fácilmente expandible.

La información de los exámenes clínicos debe ser manejada de forma ética y profesional ya que de esto depende prevenir de manera oportuna la detección de las enfermedades posibles que tenga algún paciente. Por el gran volumen de información se requiere automatizar todos los procesos de un laboratorio clínico esto optimizaría tiempo y recursos.

3. PLANTEAMIENTO DEL PROBLEMA

Durante el desarrollo de este proyecto de tesis, se plantea la necesidad de contar con un sistema de laboratorio clínico, que permita ofrecer servicios en el campo de la salud.

Existe una necesidad de generar información de una forma ágil y eficiente, para que los profesionales en la salud, tomen decisiones a tiempo, disponer de acceso vía internet, para permitir que esta herramienta sea una vía de desarrollo en el campo de la salud.

Con el advenimiento de la era de la informática, hoy los cambios en teorías, la presentación de nuevos descubrimientos, la generación de corrientes ideológicas, hechos noticiosos ocurridos en cualquier parte del mundo, avances tecnológicos, son presentados rápidamente y publicados para todas las personas a través de Internet. No hay lugar a duda, de la gran importancia que juega este medio, para que llegue la información a tiempo a los médicos y usuarios.

Este proyecto está orientado a captar un grupo de PYMES en el campo de laboratorios clínicos en la ciudad de Guayaquil, lo que les permitirá gestionar la parte operativa y administrativa de una manera eficiente y eficaz, acorde a las necesidades de la sociedad actual, lo aplicaremos en el laboratorio CEPAC.

El paciente lleva la orden de examen al laboratorio, la recepta una secretaria y se ingresa la orden de examen existente, el sistema maneja comisiones por médico, se genera una factura, se recibe un abono, el sistema genera la orden de trabajo, la enfermera toma la muestra y se agrupan por tipo de examen pueden ser Bioquímica, Hematología y uro análisis; se hace el análisis clínico; el ingreso de resultados manuales los realiza el digitador y se generan los datos automáticos, se validan los resultados por el tecnólogo médico, revisar si el paciente no posee deudas; y se entregan los resultados de forma presencial o vía web. **Ver anexo1**

RESUMEN DE LA PROPUESTA DE INTERVENCIÓN

Nuestro proyecto está orientado a desarrollarse en un ambiente Cliente – Servidor, lo que permitirá al usuario acceder de forma rápida y segura.

El sistema nos permitirá **controlar la gestión operativa**, que se trata de capturar y procesar los datos que proporcionan los equipos médicos o los datos serán ingresados de forma manual y así presentar los reportes oportunos, para el médico y el usuario. Nos ayudara en la **gestión administrativa** en lo que corresponde a las, solicitudes de examen, facturación, caja y el ingreso de resultados de los exámenes realizados, de esta manera tener un buen control en el laboratorio, para brindar un servicio de excelencia, a los usuarios del laboratorio.

3.1. Descripción general del sistema

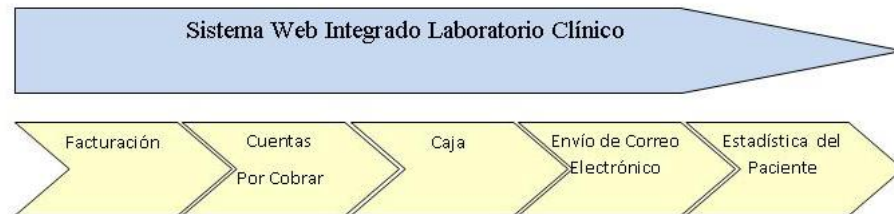


Fig. N° 1: Bosquejo de Proceso de Quiklab

Fuente: Autores

3.2. El sistema

El sistema estará ubicado en el **servidor WEB**, donde se encontrara instalado el sistema integrado. El sistema requiere de una base de datos diseñada en **Oracle 10g** esta nos permitirá almacenar, administrar información, llevar un control de todas las transacciones realizadas en la base de datos y además nos permite asignar el acceso con los permisos y roles necesarios para cada usuario. El sistema va a desarrollarse en **Oracle Forms** orientado a la web, es un lenguaje de programación de última generación que nos permitirá desarrollar los diversos módulos del sistema, **Oracle Report** este nos programa nos permite diseñar los reportes utilizando el lenguaje de consulta estructurado o **SQL** (por sus siglas en inglés structured query language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. **Flash** nos permite diseñar animaciones, **Help & Manual** nos permite crear un manual de usuario utilizando **HTML**, **PL/SQL** es un lenguaje que permite definir procedimientos utilizando sentencias SQL, el PL/SQL es el lenguaje de programación estructurado de las bases de datos Oracle, estas son las herramientas utilizadas en nuestro proyecto de tesis.

OBJETIVOS

3.3. General

“Crear un sistema integrado web para automatizar actividades dentro de un laboratorio clínico.”

3.4. Específicos

- Crear los formularios que nos permiten crear, modificar y desactivar, empresa, pacientes, clientes, medico, vendedor, etapa humana, test, rango, usuario, acceso y menú.
- Crear la solicitud del paciente
- Generar un informe con los datos de la solicitud del paciente.
- Generar un ticket-web que le permita al paciente consultar los resultados del examen.
- Revisar que el cliente no tenga deudas pendientes.
- Crear la factura del cliente, invocando las solicitudes que ya fueron creadas.
- Emitir un informe con los datos de la factura.
- Crear un formulario que permita ingresar los resultados del examen.
- Generar un informe con los resultados del examen
- Realizar el cobro de las facturas mediante el modulo de caja.
- Generar un informe para presentar el ticket de caja con el abono que realice a la factura.
- Llevar el control de los saldos de los clientes.
- Seguir un patrón para la presentación de resultados requeridos por el laboratorio clínico, paciente y medico.
- Facilitar la información a los médicos y pacientes a tiempo, mediante reportes en la web.
- Facilitar el historial de los exámenes clínicos al médico.

- Implementar un control de acceso al sistema, designando las opciones que pueda acceder el usuario.

4. BENEFICIARIOS DE LA PROPUESTA

QuickLab propone beneficios en conjunto con el campo de la salud, ya que permite procesar la información a tiempo para que los diagnósticos y tratamientos sean efectivos.

4.1. Beneficios en el campo de la salud

- La Organización mundial de la salud (OMS) define la salud como estado de completo bienestar físico, mental, emocional y social. La salud implica que todas las necesidades fundamentales de las personas estén cubiertas: afectivas, sanitarias, nutricionales, sociales y culturales, esto es una utopía ya que sólo entre el 10 y el 25% de la población mundial se encuentra completamente sana. Las enfermedades en nuestro país avanzan a pasos agigantados y se necesita que la tecnología ayude a automatizar los procesos de análisis clínicos para obtener información a tiempo e idónea para que los médicos tomen decisiones acertadas. QuickLab puede cubrir este servicio.
- QuickLab es un sistema web, que estará las 24 horas brindando servicio ya que la salud no espera hay que actuar inmediatamente de forma acertada, el médico puede estar y el paciente puede estar en cualquier lugar mediante acceso a internet tendrá los resultados esperados. Pero QuickLab podrá ofrecerle la mejor manera de ver los resultados de los análisis clínicos mediante un servicio web.
- El dueño del laboratorio clínico, tendrá la oportunidad de brindar un buen servicio y llevar un buen control de la administración de su negocio, ya que **QuickLab** maneja modulo de facturación, cuentas por cobrar y caja lo que le

ayudara a tener un buen control administrativo. Optimizando su tiempo y dinero. Mejorando la satisfacción del cliente.

4.2. Beneficios en el mercado

- El mercado de la salud contribuye fundamentalmente al desarrollo social de las regiones ecuatorianas, en especial de nuestro puerto principal Santiago de Guayaquil, permitiendo mejorar el estilo de vida de diferentes sectores sociales. QuickLab sería un gran aliado de los médicos y pacientes para tomar decisiones a tiempo.
- **QuickLab** está enfocado principalmente en el mercado de la salud; es uno de los mercados que cambia de una manera vertiginosa es por esto que debe estar muy atento y debe estar equipado con tecnología de punta que le permita ser competitivo y brindar un buen servicio.

4.3. Beneficios en la población

Ecuador tiene una población que está muy expuesta a las enfermedades por situación de insalubridad y por las amplias brechas sociales, culturales y económicas que existen en la sociedad, Guayaquil necesita estar preparado con laboratorios clínicos que brinden servicios eficientes, y eficaces para esto QuickLab brinda su servicio las 24 horas del día y los 365 días del año sin fronteras ya que los resultados de sus pruebas clínicas podrán ser revisadas a través de Internet.

5. DELIMITACIÓN

5.1. Espacial

Para poder cumplir con los objetivos propuestos es necesario definir el alcance que tendrá nuestro proyecto. De esta forma enfocaremos de mejor manera. El sistema a implementarse deberá tener:

- Nuestro proyecto propone la utilización de una [infraestructura virtual] que facilite la puesta en marcha de nuevos servicios con un mínimo costo y en el menor tiempo, aprovechando las infraestructuras existentes y explorando las posibilidades del nuevo planteamiento como el cloud computing (Nube de cómputo).
- Según Hugh Macleod: “el cloud computing es la verdadera batalla importante en este momento en la escena tecnológica: las compañías que dominen “la nube” serán los verdaderos actores del futuro, con esquemas de concentración muy importantes debido a la misma naturaleza de la actividad”.

5.2. Temporal

El nuevo enfoque de sistemas, de brindar un servicio vía web, le facilita al usuario, ya no invertir grandes cantidades de dinero en la compra software e infraestructura, nuestro sistema estará ubicado en el servidor de un proveedor de hosting y dominio, lo que facilitará a los usuarios, el acceso las 24 horas los 365 días del año, permitiendo mejorar el tiempo de respuesta en cuanto al acceso a los datos requeridos.

5.3. Académica

El sistema integrado para la automatización de un laboratorio clínico orientado a la web tendrá los siguientes módulos:

- Modulo de laboratorio clínico
- Modulo de facturación
 - Modulo de cuentas por cobrar
 - Modulo de caja bancos
 - Análisis clínico
 - Historial de pacientes según resultados
 - Envío de mensajes de texto a médicos y usuarios
 - Aspectos de seguridad

6. MARCO REFERENCIAL

6.1. Marco teórico

8.1.1 Análisis y diagramación de procesos empresariales

8.1.1.1 La cadena de valor

Según Michael E. Porter en su obra *Competitive Advantage: Creating and Sustaining Superior Performance*, dice que La cadena de valor empresarial, o cadena de valor, es un modelo teórico que permite describir el desarrollo de las actividades de una organización empresarial generando valor al cliente final.



Fig. N° 2.: Modelo de Cadena de valor

Fuente: http://es.wikipedia.org/wiki/Cadena_de_valor

Esquema del modelo de la Cadena de valor empresarial. En la parte superior, las actividades de apoyo o auxiliares. En la parte inferior, las actividades primarias.

¹ Información obtenida de:
Wiki pedía, Cadena de valor, 1 julio 2010
http://es.wikipedia.org/wiki/Cadena_de_valor

8.1.1.2. Actividades primarias

Las actividades primarias se refieren a la creación física del producto, su venta y el servicio postventa, y pueden también a su vez, diferenciarse en sub-actividades. El modelo de la cadena de valor distingue cinco actividades primarias:

- **Logística interna**: comprende operaciones de recepción, almacenamiento y distribución de las materias primas.
- **Operaciones (producción)**: procesamiento de las materias primas para transformarlas en el producto final.
- **Logística externa**: almacenamiento de los productos terminados y distribución del producto al consumidor.
- **Marketing y Ventas**: actividades con las cuales se da a conocer el producto.
- **Servicio**: de post-venta o mantenimiento, agrupa las actividades destinadas a mantener, realzar el valor del producto, mediante la aplicación de garantías.

8.1.1.3. Actividades de apoyo

Las actividades primarias están apoyadas o auxiliadas por las también denominadas 'actividades secundarias':

- **Infraestructura de la organización**: actividades que prestan apoyo a toda la empresa, como la planificación, contabilidad y las finanzas.
- **Dirección de recursos humanos**: búsqueda, contratación y motivación del personal.
- **Desarrollo de tecnología, investigación y desarrollo**: obtención, mejora y gestión de la tecnología.
- **Abastecimiento (compras)**: proceso de compra de los materiales.

Para cada actividad de valor añadido han de ser identificados los generadores de costes y valor.

8.1.1.4. Gestión de procesos de negocio

Se llama **Gestión de procesos de negocio** (*Business Process Management* o BPM en inglés) a la metodología empresarial cuyo objetivo es mejorar la eficiencia a través de la gestión sistemática de los procesos de negocio, que se deben modelar, automatizar, integrar, monitorizar y optimizar de forma continua. Como su nombre sugiere, *BPM* se enfoca en la administración de los procesos del negocio.

A través del modelado de las actividades y procesos puede lograrse un mejor entendimiento del negocio y muchas veces esto presenta la oportunidad de mejorarlos. La automatización de los procesos reduce errores, asegurando que los mismos se comporten siempre de la misma manera y dando elementos que permitan visualizar el estado de los mismos. La administración de los procesos permite asegurar que los mismos se ejecuten eficientemente, y la obtención de información que luego puede ser usada para mejorarlos. Es a través de la información que se obtiene de la ejecución diaria de los procesos, que se puede identificar posibles ineficiencias en los mismos, y actuar sobre las mismas para optimizarlos.

“Para soportar esta estrategia es necesario contar con un conjunto de herramientas que den el soporte necesario para cumplir con el ciclo de vida de BPM. Este conjunto de herramientas son llamadas Business Process Management System y con ellas se construyen aplicaciones BPM”.²

² Información obtenida de:

Wiki pedía, Gestion de procesos, 1 julio 2010

http://es.wikipedia.org/wiki/Gesti%C3%B3n_de_procesos_de_negocio

8.1.1.5. Aspectos básicos de procesos

8.1.1.5.1. Definición de proceso

“Un proceso se define como un conjunto de tareas, actividades o acciones lógicamente relacionadas entre sí que, a partir de una o varias entradas de información, materiales o de salidas de otros procesos, dan lugar a una o varias salidas también de materiales (productos) o información con un valor añadido”.³

Hay tres elementos importantes en un proceso:

- **Valor agregado:** Aquellas que transforman los datos e insumos para crear información y productos o servicios para el cliente.
- **Traspaso (flujo):** Aquellas en las que se entrega de manera interdepartamental o externa la información y productos.
- **Control:** Aquellas que permiten que las actividades de traspaso se lleven a cabo de acuerdo a especificaciones previas de calidad, tiempo y costo establecido.

Algunos ejemplos de procesos pueden ser los de producción de bienes, entrega de productos o servicios, el de gestión de las relaciones con los clientes (habitualmente gestionada por un sistema CRM), el de desarrollo de la estrategia general de la empresa, el de I+D+I de nuevos productos o servicios, etc.

Estos procesos deben estar correctamente gestionados empleando distintas herramientas de gestión de procesos (en definitiva gestión de la organización) como puede ser un sistema de planificación de recursos empresariales (ERP), un sistema de Workflow y otros más.

³ Información obtenida de:
Wiki pedía, Reingeniería de procesos, 1 julio 2010
http://es.wikipedia.org/wiki/Reingenier%C3%ADa_de_procesos#Definici.C3.B3n_de_proceso

8.1.1.5.2 Tipos de procesos

“Los procesos pueden ser clasificados en función de varios criterios. Pero quizá la clasificación de los procesos más habitual en la práctica es distinguir entre: estratégicos, claves o de apoyo”.⁴

1. Los procesos *clave* son también denominados operativos y son propios de la actividad de la empresa, por ejemplo, el proceso de aprovisionamiento, el proceso de producción, el proceso de prestación del servicio, el proceso de comercialización, etc.

2. Los procesos *estratégicos* son aquellos procesos mediante los cuales la empresa desarrolla sus estrategias y define los objetivos. Por ejemplo, el proceso de planificación presupuestaria, proceso de diseño de producto y/o servicio, etc.

3. Los procesos *de apoyo*, o de soporte son los que proporcionan los medios (recursos) y el apoyo necesario para que los procesos clave se puedan llevar a cabo, tales como proceso de formación, proceso informático, proceso de logística, etc.

También, podemos distinguir entre procesos clave y procesos críticos. En general, los procesos *clave* atienden a la definición expuesta anteriormente. Están principalmente orientados hacia la satisfacción del cliente y en ellos se emplean una gran cantidad de los recursos disponibles por la empresa. Por otro lado, un proceso es *crítico* cuando en gran medida la consecución de los objetivos y los niveles de calidad de la empresa dependen de su desarrollo.

⁴ información obtenida de:

Mailxmail, Gestión de calidad de Procesos, 1 julio 2010

<http://www.mailxmail.com/curso-gestion-calidad-procesos-tecnicas-herramientas-calidad/tipos-procesos>

8.1.1.5.3 Tipos de diagramas de flujo

- **Formato vertical:** En él el flujo o la secuencia de las operaciones, va de arriba hacia abajo. Es una lista ordenada de las operaciones de un proceso con toda la información que se considere necesaria, según su propósito.
- **Formato horizontal:** En él, el flujo o la secuencia de las operaciones, va de izquierda a derecha.
- **Formato panorámico:** El proceso entero está representado en una sola carta y puede apreciarse de una sola mirada mucho más rápido que leyendo el texto, lo que facilita su comprensión, aun para personas no familiarizadas. Registra no solo en línea vertical, sino también horizontal, distintas acciones simultáneas y la participación de más de un puesto o departamento que el formato vertical no registra.
- **Formato Arquitectónico:** Describe el itinerario de ruta de una forma o persona sobre el plano arquitectónico del área de trabajo. El primero de los flujogramas es eminentemente descriptivo, mientras que los utilizados son fundamentalmente representativos.

8.1.1.5.4 Simbología y significado

- **Óvalo:** Inicio y término. (Abre y/o cierra el diagrama).
- **Rectángulo:** Actividad. (Representa la ejecución de una o más actividades o procedimientos).
- **Rombo:** Decisión. (Formula una pregunta o cuestión).
- **Círculo:** Conector. (Representa el enlace de actividades con otra dentro de un procedimiento).
- **Triángulo boca abajo:** Archivo definitivo. (Guarda un documento en forma permanente).
- **Triángulo boca arriba:** Archivo temporal. (Proporciona un tiempo para el almacenamiento del documento).

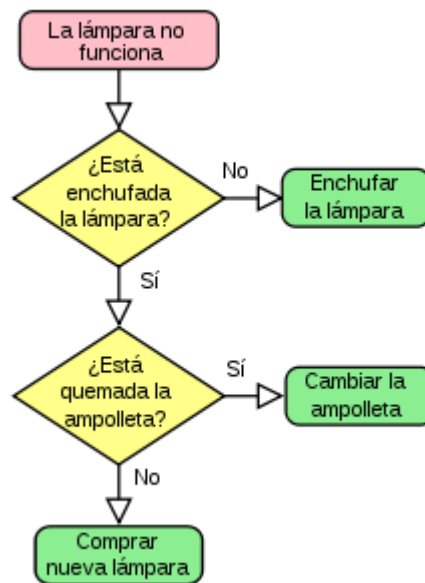


Fig. N°3.: Diagrama de flujo sencillo con los pasos
A seguir si una lámpara no funciona.

Fuente: http://es.wikipedia.org/wiki/Diagrama_de_flujo⁵

⁵ Información obtenida de:
Wiki pedía, diagrama de flujo, 1 julio 2010
http://es.wikipedia.org/wiki/Diagrama_de_flujo

8.1.2 Análisis y diseño de sistemas orientado a objetos

8.1.2.1 Modelo entidad relación

Un **diagrama o modelo entidad-relación** (a veces denominado por su siglas, E-R "Entity relationship", o, "DER" Diagrama de Entidad Relación) es una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información así como sus interrelaciones y propiedades.

8.1.2.1.1 Modelo

La metodología que se va a seguir para el diseño lógico en el modelo relacional consta de dos fases:

- Construir y validar los esquemas lógicos locales para cada vista de usuario.
- Convertir los esquemas conceptuales locales en esquemas lógicos locales.
- Derivar un conjunto de relaciones (tablas) para cada esquema lógico local.
- Validar cada esquema mediante la normalización.
- Validar cada esquema frente a las transacciones del usuario.
- Dibujar el diagrama entidad-relación.
- Definir las restricciones de integridad.
- Revisar cada esquema lógico local con el usuario correspondiente.

En esta fase, se construyen los esquemas lógicos locales para cada vista de usuario y se validan. En esta fase se refinan los esquemas conceptuales creados durante el diseño conceptual, eliminando las estructuras de datos que no se pueden implementar de manera directa sobre el modelo que soporta el SGBD, en el caso que nos ocupa, el modelo relacional. Una vez hecho esto, se obtiene un primer esquema lógico que se valida mediante la normalización y frente a las transacciones que el sistema debe llevar a cabo, tal y como se refleja en las especificaciones de requisitos de usuario. El esquema lógico ya validado se puede utilizar como base para el desarrollo de prototipos. Una vez finalizada esta fase, se dispone de un esquema lógico para cada vista de usuario que es correcto, comprensible y sin ambigüedad.

8.1.2.1.2 Convertir los esquemas conceptuales locales en esquemas lógicos locales.

En este paso, se eliminan de cada esquema conceptual las estructuras de datos que los sistemas relacionales no modelan directamente:

- a) Eliminar las relaciones de muchos a muchos, sustituyendo cada una de ellas por una nueva entidad intermedia y dos relaciones de uno a muchos de esta nueva entidad con las entidades originales. La nueva entidad será débil, ya que sus ocurrencias dependen de la existencia de ocurrencias en las entidades originales.
- b) Eliminar las relaciones entre tres o más entidades, sustituyendo cada una de ellas por una nueva entidad (débil) intermedia que se relaciona con cada una de las entidades originales. La cardinalidad de estas nuevas relaciones binarias dependerá de su significado.
- c) Eliminar las relaciones recursivas, sustituyendo cada una de ellas por una nueva entidad (débil) y dos relaciones binarias de esta nueva entidad con la entidad original. La cardinalidad de estas relaciones dependerá de su significado.
- d) Eliminar las relaciones con atributos, sustituyendo cada una de ellas por una nueva entidad (débil) y las relaciones binarias correspondientes de esta nueva entidad con las entidades originales. La cardinalidad de estas relaciones dependerá del tipo de la relación original y de su significado.
- e) Eliminar los atributos multievaluados, sustituyendo cada uno de ellos por una nueva entidad (débil) y una relación binaria de uno a muchos con la entidad original.
- f) Revisar las relaciones de uno a uno, ya que es posible que se hayan identificado dos entidades que representen el mismo objeto (sinónimos). Si así fuera, ambas entidades deben integrarse en una sola.
- g) Eliminar las relaciones redundantes. Una relación es redundante cuando se puede obtener la misma información que ella aporta mediante otras relaciones. El hecho de que haya dos caminos diferentes entre dos entidades no implica que uno de los caminos corresponda a una relación redundante, eso dependerá del significado de cada relación.

8.1.2.1.3 Orientación a objetos

Es fundamental comprender la orientación a objetos ya que se aplicarán en el análisis que se haga para la modelación de los diagramas.

- Clase
- Herencia
- Polimorfismo
- Encapsulamiento
- Sobrecarga
- Envío de mensajes
- Asociaciones
- Agregación

La orientación a objetos es un medio para la generación de programas, tiene varias ventajas:

- Fomenta una metodología basada en componentes para el desarrollo de software.
- Primero se genera un sistema mediante un conjunto de objetos
- Agregar funcionalidad a los componentes que ya había generado o agregándole nuevos componentes
- Se puede volver a utilizar los objetos, esto permite reducir tiempo de desarrollo de un sistema.

8.1.2.1.4 Clase

Es la abstracción de un objeto o también podemos decir que es un conjunto de datos y métodos. Ejemplo:

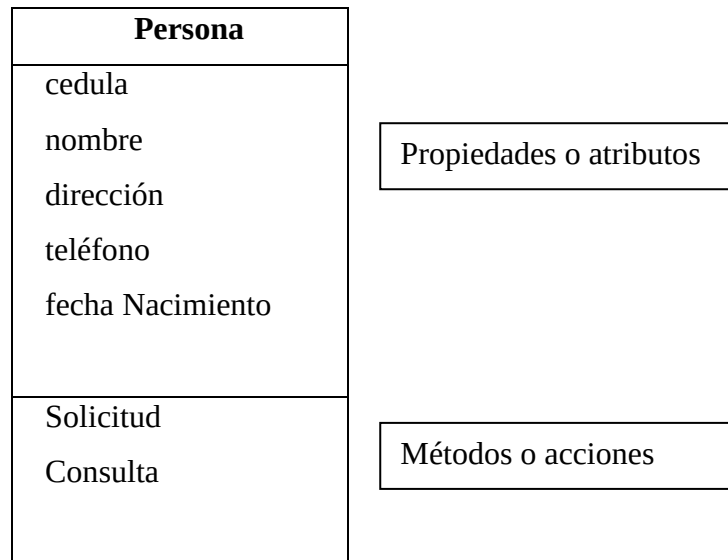


Fig. N° 4. Ejemplo de clases

Fuente: Autores

8.1.2.1.5 Objeto

Un objetos es la instancia de una clase; los objetos concretos y virtuales, están a nuestro alrededor, ellos conforman nuestro mundo, los mismos que se los puede representar en un software.

Un objeto es una instancia de una clase (o categoría). Ejemplo un alumno y un profesor son instancias de la clase persona. Un objeto cuenta con una estructura:

- Atributos (Propiedades)
- Acciones

Las acciones son todas las actividades que el objeto es capaz de realizar, los atributos y acciones, en conjunto se conocen como características o rasgos.

8.1.2.1.6 Herencia

La herencia significa pasar los bienes a un hijo el mismo que hereda clases – métodos, solo se puede instanciar a la clase hija.

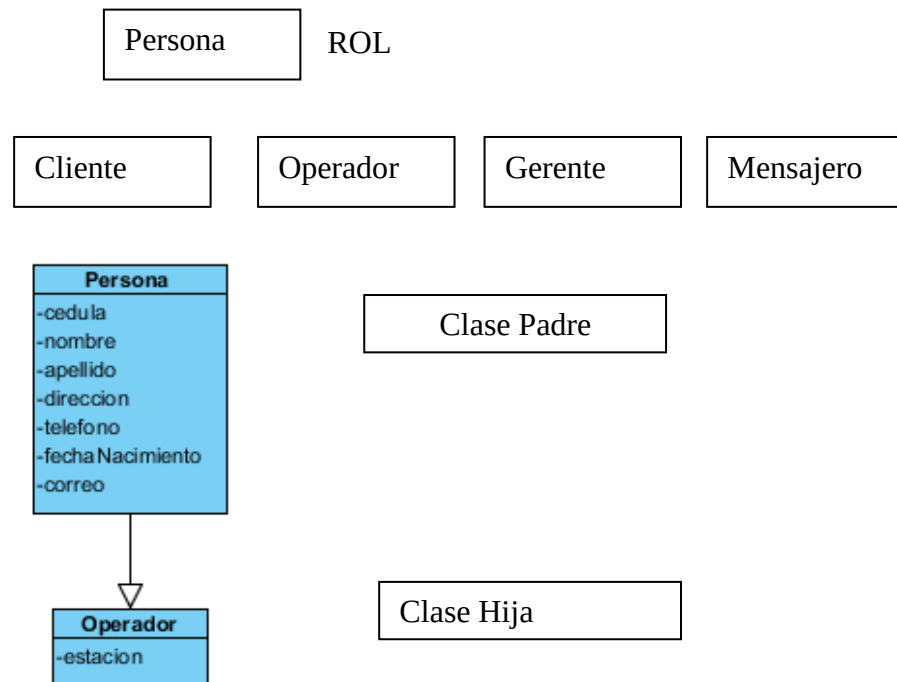


Fig. N° 5. Ejemplos de Herencia

Fuente: Autores

Un objeto es una instancia de una clase: como instancia de una clase, un objeto tiene todas las características de la clase de la que proviene. A esto se le conoce como herencia.

Un objeto no sólo hereda de una clase, sino que una clase también puede heredar de otra.

8.1.2.1.7 Polimorfismo

Es cuando una operación tiene el mismo nombre en diferentes clases. En el polimorfismo, una operación puede tener el mismo nombre en diversas clases, y funcionar distinto en cada una.

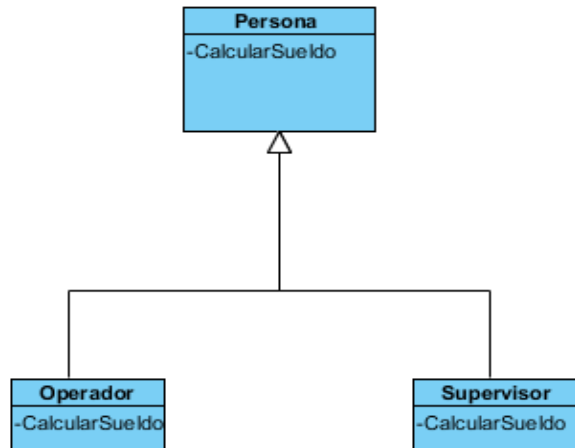


Fig. N° 6. Ejemplos de Polimorfismo
Fuente: Autores

8.1.2.1.8 Encapsulamiento

Solo la clase dueña la puede utilizar (privada o pública); El encapsulamiento permite reducir el potencial de errores que pueden ocurrir. En un sistema que consta de objetos, éstos dependen unos de otros estos dependen unos de otros en diversas formas.

8.1.2.1.9 Sobrecarga

Es llamado polimorfismo, son dos métodos que se llaman igual pero en la misma clase. Ejemplo

Calcular el sueldo de forma diferente.

Sueldo fijo + comisiones + bono Sueldo fijo + horas extras

8.1.2.1.10 Constructor

Es el método que nos permite instanciar una clase. Ejemplo:

```
Clase Ejemplo {  
    Ejemplo ( ) { //Constructor    }  
}  
Ejemplo = new Ejemplo ( );
```

8.1.2.1.11 Lenguaje unificado de modelado (UML)

Es una herramienta muy utilizada en el mundo actual para el desarrollo de sistemas informáticos. Permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas.

8.1.2.1.12 Diagramas UML

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. La finalidad de los diagramas es presentar varias perspectivas de un sistema es similar a un modelo a escala de un edificio junto con la incorporación con la interpretación del artista del edificio. Es importante destacar que un modelo UML describe lo que supuestamente hará el sistema, pero no dice cómo implementar dicho sistema.

8.1.2.1.13 Diagrama de clase

Los diagramas de clases utilizan los siguientes símbolos:

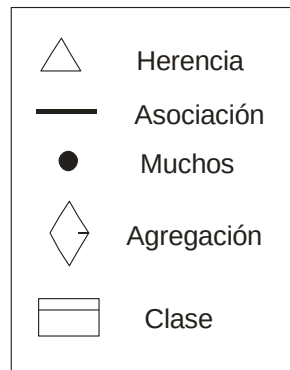


Fig. N° 7. Simbología para realizar los diagramas de clases

Fuente: Autores

Este diagrama nos permite representar las cosas que nos rodean, estas cosas pueden tener atributos (propiedades) y que realicen determinadas acciones. Cada acción puede representar un conjunto de tareas.

Las cosas se albergan en categorías (automóviles, mobiliario, lavadoras...) A tales categorías se la conoce como clases.

Una clase es una categoría o grupo de cosas que tienen atributos y acciones similares. Ejemplo:

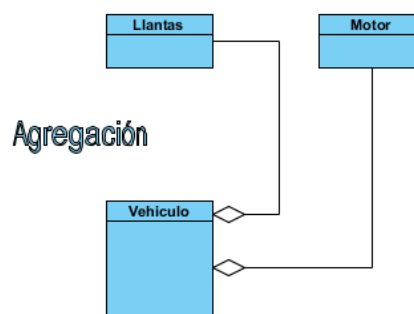


Fig. N° 8. Ejemplos de agregación

Fuente: Autores

8.1.2.1.14 Diagrama de casos de uso

En los diagramas de casos de uso se utilizan los siguientes símbolos:

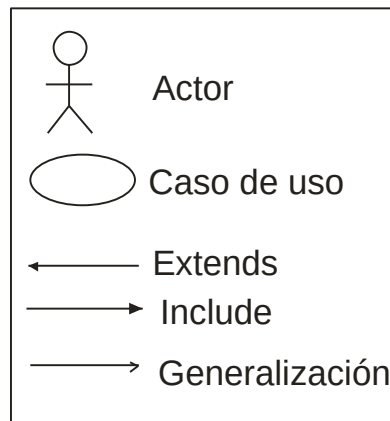


Fig. N° 9. Símbolos para representación gráfica de casos de uso

Fuente: Autores

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Para los desarrolladores de sistemas es una herramienta valiosa, ya que es una técnica de aciertos y errores para obtener los requerimientos del sistema desde el punto de vista del usuario. Un sistema debe ser utilizado por la gente en general (no solo por expertos en computación).

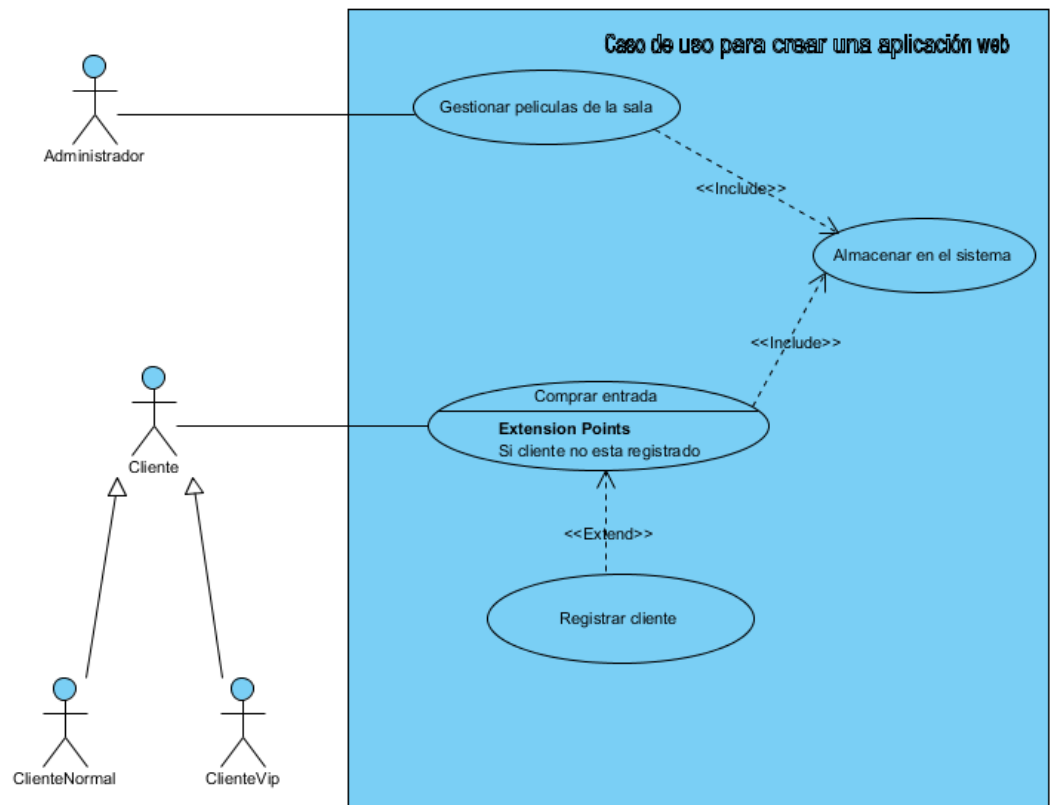


Fig. 10. Ejemplos de diagramas de casos de uso

Fuente: Autores

8.1.2.1.15 Diagramas de estado

Los diagramas de estado utilizan los siguientes símbolos:



Fig. N°11. Simbología para realizar diagramas de estados

Fuente: Autores

En cualquier momento, un objeto se encuentra en un estado en particular. Una persona puede ser recién nacida, infante, adolescente, joven o adulta. Un elevador se moverá hacia arriba, estará en estado de reposo o se moverá hacia abajo. Una lavadora puede estar en fase de remojo, lavado, enjuague, centrifugado o apagado.

El diagrama de estados UML, que se presenta, captura esta pequeña realidad.

Ejemplo:

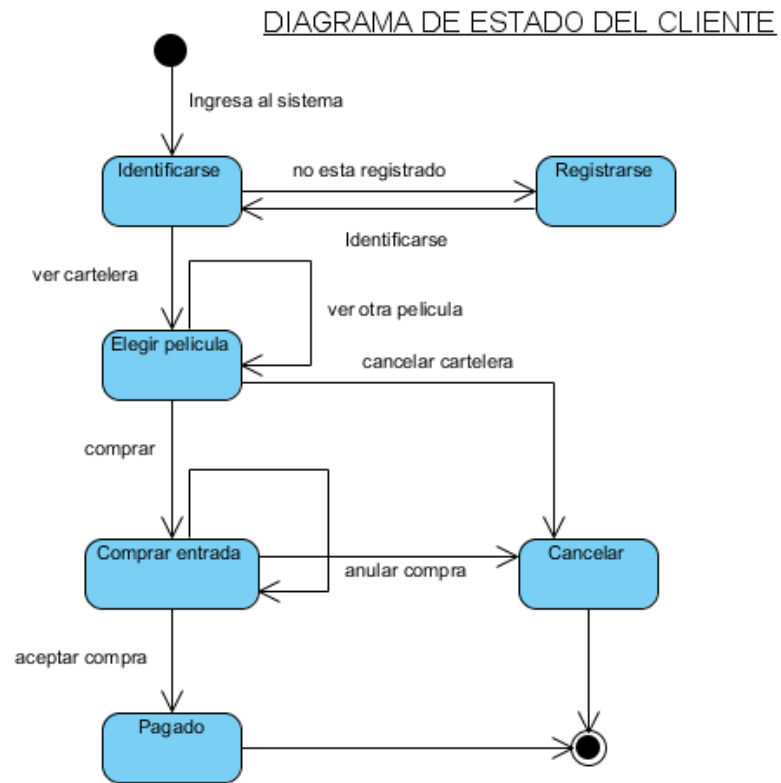


Fig. N°12. Ejemplo De Diagramas De Estado

Fuente: Autores

8.1.2.1.16 Diagrama de secuencias

Este diagrama de secuencia se lo realiza por cada caso de uso, nos permite tener claro que métodos se va a implementar.

Los diagramas de clases y los de objeto representan información estática. No obstante, en un sistema funcional los objetos interactúan entre sí, y tales interacciones suceden con el tiempo. El diagrama de secuencias UML muestra la mecánica de la interacción con base en tiempos. Ejemplo:

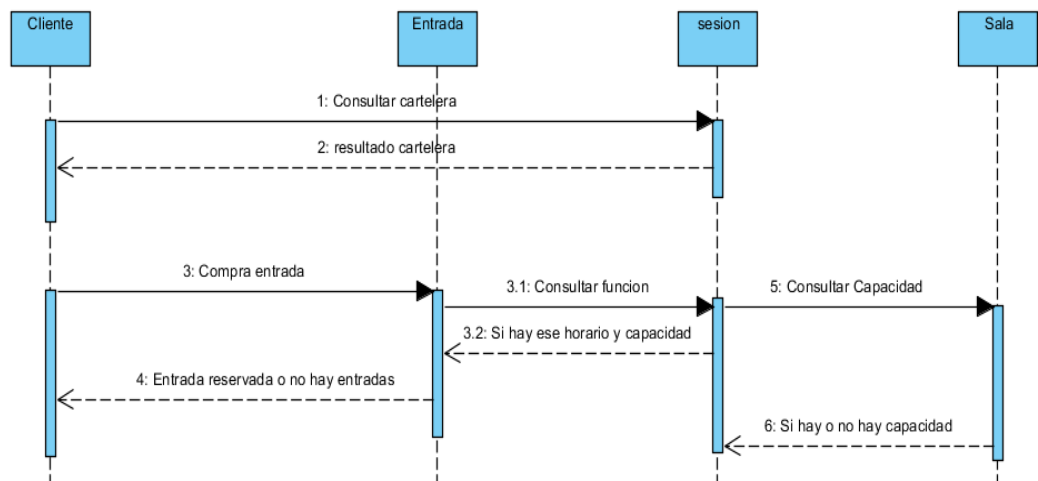


Fig. 13. Ejemplo de diagrama de secuencia

Fuente: Autores

8.1.3 Administración de base de datos Oracle 10g

8.1.3.1 Oracle 10g

Como lenguaje de programación y manejador de base de datos. Oracle Forms, un componente de Oracle Fusión Middleware, se establece la tecnología a largo Oracle para diseñar y construir aplicaciones empresariales de forma rápida y eficiente. Oracle mantiene su compromiso con el desarrollo de esta tecnología, y la liberación continua como un componente de la plataforma Oracle. Este continuo compromiso con la tecnología de formularios le permite aprovechar su inversión existente al permitir el mejoramiento y la integración de aplicaciones existentes de Oracle Forms para aprovechar las tecnologías web y arquitecturas orientadas a servicios (SOA).

8.1.3.2 Arquitectura de Oracle

La Arquitectura general de Oracle consiste de varios procesos corriendo en la máquina donde reside la instancia, más los espacios de memoria dedicados a ejecutar procesos específicos o al almacenaje de información de cada proceso y la base de datos física propiamente tal, con sus archivos de control, de datos y de transacciones.

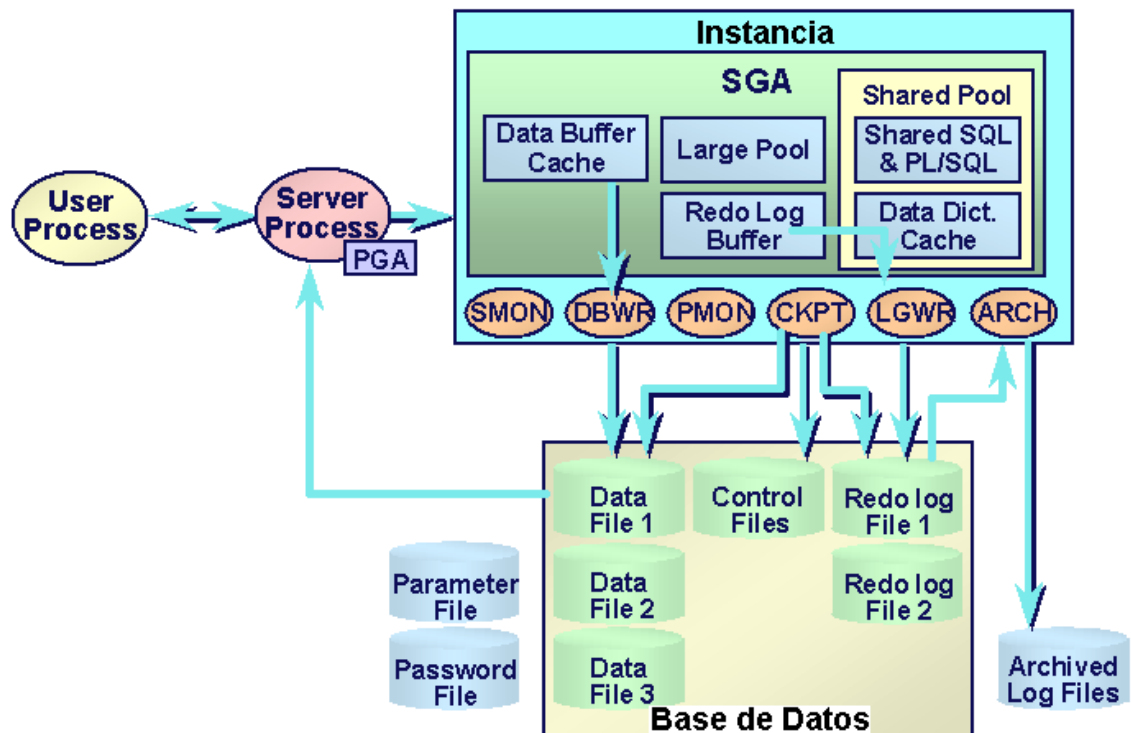


Fig. N°14: Arquitectura de Oracle

Fuente: www.napolifirewall.com/ORACLE_file/Image155.gif

8.1.3.3 Instancia de Oracle

Una instancia de Oracle está conformada por varios procesos y espacios de memoria compartida que son necesarios para acceder a la información contenida en la base de datos.

La instancia está conformada por procesos del usuario, procesos que se ejecutan en el background de Oracle y los espacios de memoria que comparten estos procesos.

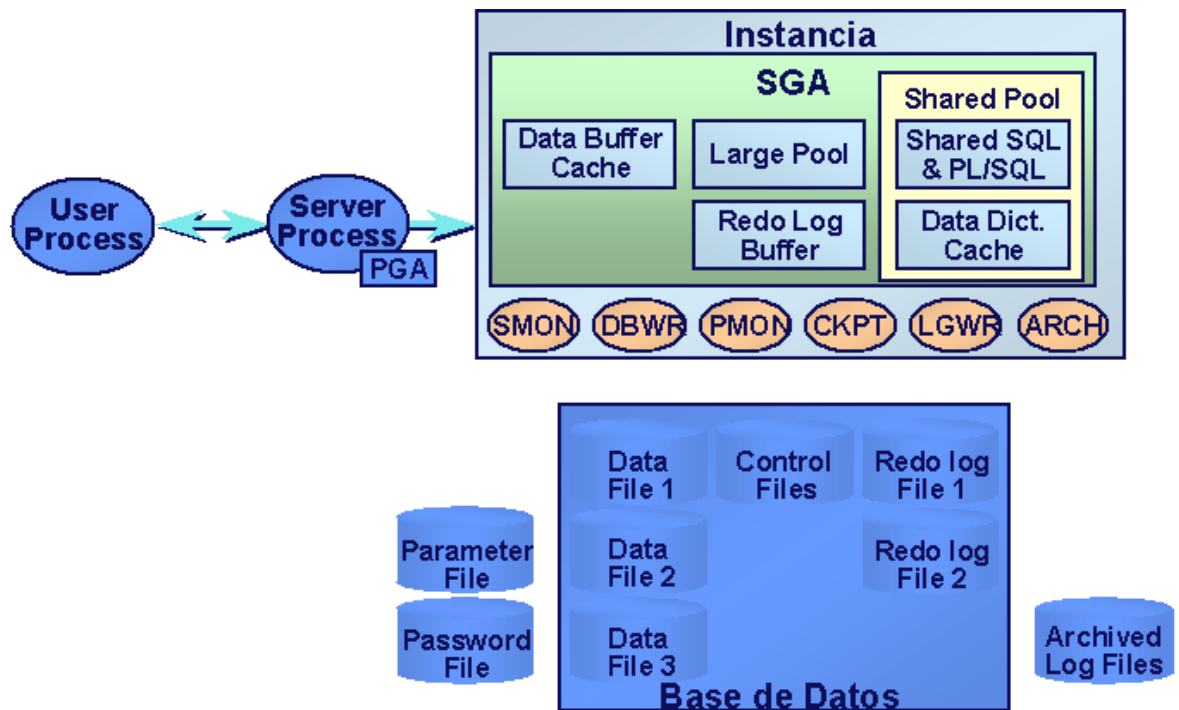


Fig. N°. 15: Arquitectura de la Instancia de Oracle

Fuente: <http://www.infor.uva.es/~jvegas/cursos/bd/orarq/orarq.html>

8.1.3.4 Área global del sistema (SGA)

El SGA es un área de memoria compartida que se utiliza para almacenar información de control y de datos de la instancia. Se crea cuando la instancia es levantada y se borra cuando ésta se deja de usar (cuando se hace shutdown). La información que se almacena en esta área consiste de los siguientes elementos, cada uno de ellos con un tamaño fijo:

El buffer de caché (data base buffer cache)

- Almacena los bloques de datos utilizados recientemente (se hayan o no confirmado sus cambios en el disco). Al utilizarse este buffer se reducen las operaciones de entrada y salida y por esto se mejora el rendimiento.
- **El buffer de redo log:** Guarda los cambios efectuados en la base de datos. Estos buffers escriben en el archivo físico de redo log tan rápido como se pueda sin perder eficiencia. Este último archivo se utiliza para recuperar la base de datos ante eventuales fallas del sistema.

- **El área shared pool:** Esta sola área almacena estructuras de memoria compartida, tales como las áreas de código SQL compartido e información interna del diccionario. Una cantidad insuficiente de espacio asignado a esta área podría redundar en problemas de rendimiento. En resumen, contiene las áreas del caché de biblioteca y del caché del diccionario de datos.
- El caché de biblioteca se utiliza para almacenar código SQL compartido. Aquí se manejan los árboles de parsing y el plan de ejecución de las queries. Si varias aplicaciones utilizan la misma sentencia SQL, esta área compartida garantiza el acceso por parte de cualquiera de ellas en cualquier instante.
- El caché del diccionario de datos está conformado por un grupo de tablas y vistas que se identifican la base de datos. La información que se almacena aquí guarda relación con la estructura lógica y física de la base de datos. El diccionario de datos contiene información tal como los privilegios de los usuarios, restricciones de integridad definidas para algunas tablas, nombres y tipos de datos de todas las columnas y otra información acerca del espacio asignado y utilizado por los objetos de un esquema.

La SGA se divide en varias partes:

8.1.3.5 Buffers de BD, database buffer cache

Es el caché que almacena los bloques de datos leídos de los segmentos de datos de la BD, tales como tablas, índices y clúster. Los bloques modificados se llaman bloques sucios. El tamaño de buffer caché se fija por el parámetro DB_BLOCK_BUFFERS del fichero init.ora.

Como el tamaño del buffer suele ser pequeño para almacenar todos los bloques de datos leídos, su gestión se hace mediante el algoritmo LRU.

8.1.3.5.2 Buffer redo log

Los registros Redo describen los cambios realizados en la BD y son escritos en los ficheros redo log para que puedan ser utilizados en las operaciones de recuperación hacia adelante, roll-forward, durante las recuperaciones de la BD. Pero antes de ser

escritos en los ficheros redo log son escritos en un caché de la SGA llamado redo log buffer. El servidor escribe periódicamente los registros redo log en los ficheros redo log.

El tamaño del buffer redo log se fija por el parámetro LOG_BUFFER.

8.1.3.5.2 Procesos de la instancia

Según lo que se advierte en la figura 12, los procesos que se implementan en una instancia de Oracle y su función principal son los siguientes:

- **DBWR (data base writer):** Es el responsable de la escritura en disco de toda la información almacenada en los buffers de bloques que no se han actualizado.
- **LGWR (log writer):** Es el responsable de escribir información desde el buffer de log hacia el archivo redo log.
- **CKPT (checkpoint):** Es el responsable de advertir al proceso DBWR de efectuar un proceso de actualización en el disco de los datos mantenidos en memoria, incluyendo la data files y control files (para registrar el checkpoint). Este proceso es opcional, si no está presente, es el proceso LGWR quien asume la responsabilidad de la tarea.
- **PMON (process monitor):** Su misión es monitorizar los procesos del servidor y tomar acciones correctivas cuando alguno de ellos se interrumpe en forma abrupta, limpiando la caché y liberando los posibles recursos que pudieran estar asignados en ese momento. También es responsable por el restablecimiento de aquel proceso que se ha interrumpido bruscamente.
- **SMON (system monitor):** Levanta una instancia cuando se le da la instrucción de partida (al comienzo del trabajo, encontrándose previamente en shutdown). Enseguida limpia los segmentos temporales y recupera las transacciones que pudieran haberse interrumpido debido a una falla del sistema. Además disminuye la fragmentación del sistema

agrupando aquellas extensiones libres que existen dentro de la base de datos.

- **ARCH (archiver):** La función de este proceso es la de respaldar la información almacenada en los archivos redo log cuando éstos se llenan. Este proceso está siempre activo cuando se ha establecido el modo ARCHIVELOG. Si el sistema no está operando en este modo se hace más difícil recuperar el sistema sin problemas luego de una falla general.

8.1.3.5.3 Estructuras de memoria internas

Oracle mantiene dos estructuras principales de memoria: el Área Global de Programa, Program Global Area, PGA; y el Área Global del Sistema, System Global Area o también Shared Global Area, SGA.

El PGA es la zona de memoria de cada proceso Oracle. No está compartida y contiene datos e información de control de un único proceso.

El SGA es la zona de memoria en la que la BD Oracle guarda información sobre su estado. Esta estructura de memoria está disponible para todos los procesos, por eso se dice que está compartida.

8.1.3.6 Estructuras externas

Por estructuras externas se entienden los ficheros que utiliza el servidor de BD, de los cuales ya se han ido contando algunos aspectos, y otros se han ido intuyendo. Estos ficheros guardan información tanto de los datos almacenados en la BD como la necesaria para gobernar la propia BD.

8.1.3.6.1 Ficheros de la BD

En estos ficheros reside la información de la BD. Solo son modificados por el DBWR. A ellos se vuelcan los bloques sucios de la SGA cuando se hace una validación o cuando sucede un checkpoint. Las validaciones de las transacciones no producen un volcado inmediato, sino lo que se conoce por un commit diferido. Toda actualización se guarda en los ficheros de redo log, y se lleva a la BD física cuando tenemos una buena cantidad de bloques que justifiquen una operación de E/S. Almacenan los segmentos (datos, índices, rollback) de la BD. Están divididos en bloques ($\text{Bloque Oracle} = c * \text{Bloque SO}$), cada uno de los cuales se corresponde con un buffer del buffer cache de la SGA. En el bloque de cabecera no se guardan datos de usuario, sino la marca de tiempo del último checkpoint realizado sobre el fichero.

8.1.3.6.2 Ficheros Redo Log

En ellos se graba toda operación que se efectúe en la BD y sirven de salvaguarda de la misma. Tiene que haber por lo menos 2, uno de ellos debe estar activo, online, y se escribe en ellos de forma cíclica. Existe la posibilidad de almacenar los distintos ficheros de redo log en el tiempo mediante el modo ARCHIVER. Así, se puede guardar toda la evolución de la BD desde un punto dado del tiempo.

Una opción es la utilización de archivos redo log multiplexados:

- Permite al LGWR escribir simultáneamente la misma información en múltiples archivos redo log.
- Se utiliza para protegerse contra fallos en el disco.
- Da una alta disponibilidad a los archivos redo log activos u online.

8.1.3.6.3 Estructuras de proceso

El servidor se vale de una serie de procesos que son el enlace entre las estructuras físicas y de memoria. A continuación se describen cada proceso y el papel que juega en la gestión de la BD. Todo esto se puede ver en la siguiente figura.

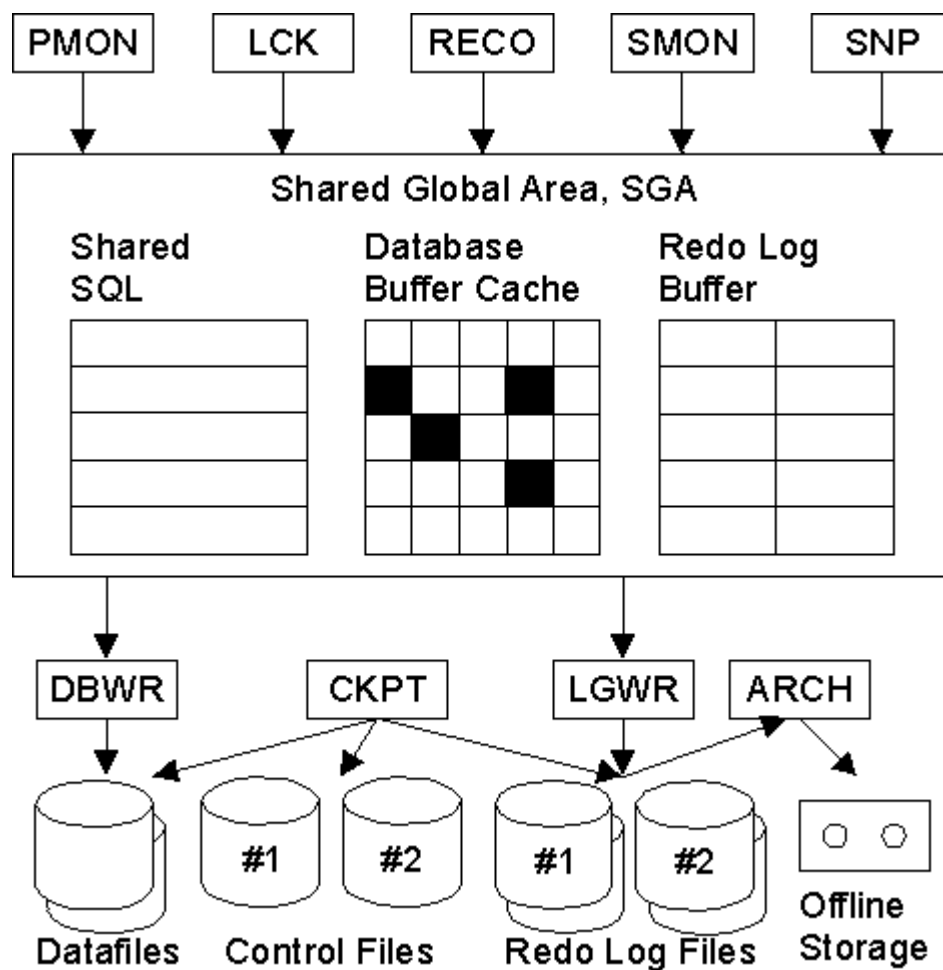


Fig. N°. 16: Estructura de proceso

Fuente: <http://www.infor.uva.es/~jvegas/cursos/bd/orarq/orarq.html>

8.1.4. Programación en PL/SQL ORACLE

8.1.4.1 ¿Qué es PL/SQL?

Es un lenguaje que permite definir procedimientos utilizando sentencias SQL.

“El lenguaje SQL no permite la definición de procedimientos, sólo permite la ejecución de sentencias SQL una a una. Así pues el lenguaje PL/SQL surgió para evitar esta limitación, el PL/SQL es el lenguaje de programación estructurado de las bases de datos Oracle y su estructura es muy similar a la de cualquier otro lenguaje con procedimientos como C o C++”.⁶

8.1.4.2 Bloques PL/SQL

8.1.4.2.1. Definición de un bloque PL/SQL

“Un bloque es la estructura básica de un programa PL/SQL y se define con la siguiente estructura”⁷:

8.1.4.2.2. Tipos de bloques PL/SQL

Un bloque PL/SQL se puede programar dentro de un Procedimiento, dentro de una función o de forma independiente. Este último caso se conoce como bloque anónimo.

Estructura de un bloque anónimo:

1. [DECLARE]
2. BEGIN
3. -- instrucciones
4. [EXCEPTION]
5. END;

⁶ Información obtenida de:

Plsql.biz, que es plsql, 1 julio 2010

http://es.wikipedia.org/wiki/Cadena_de_valor

Devjoker, Bloques-Plsql, 20 julio 2010

⁷ <http://www.devjoker.com/contenidos/Tutorial-PLSQL/27/Bloques-PLSQL.aspx>

Estructura de un bloque dentro de un procedimiento:

1. PROCEDURE nombreProcedimiento
2. IS
3. [DECLARE]
4. BEGIN
5. -- instrucciones
6. [EXCEPTION]
7. END;

Estructura de un bloque dentro de una función:

1. FUNCTION nombreFuncion
2. RETURN tipodedato
3. IS
4. [DECLARE]
5. BEGIN
6. -- instrucciones
7. RETURN valor;
8. [EXCEPTION]
9. END;

8.1.4.2.3. Concepto de cursor SQL

“PL/SQL utiliza cursores para gestionar las instrucciones **SELECT**. Un cursor es un conjunto de registros devuelto por una instrucción SQL. Técnicamente los cursores son fragmentos de memoria que reservados para procesar los resultados de una consulta **SELECT**”.⁸

Podemos distinguir dos tipos de cursores:

- **Cursores implícitos.** Este tipo de cursores se utiliza para operaciones **SELECT INTO**. Se usan cuando la consulta devuelve un único registro.
- **Cursores explícitos.** Son los cursores que son declarados y controlados por el programador. Se utilizan cuando la consulta devuelve un conjunto de registros. Ocasionalmente también se utilizan en consultas que devuelven un único registro por razones de eficiencia. Son más rápidos.

Un cursor se define como cualquier otra variable de PL/SQL y debe nombrarse de acuerdo a los mismos convenios que cualquier otra variable. Los cursores implícitos no necesitan declaración.

El siguiente ejemplo declara un cursor explícito:

```
declare  
  cursor c_paises is  
    SELECT CO_PAIS, DESCRIPCION  
    FROM PAISES;  
begin  
  /* Sentencias del bloque ... */  
end;
```

⁸ Información obtenida de:
Devjoker, Cursores en Plsql, abril 2010

<http://www.devjoker.com/contenidos/Tutorial-PLSQL/32/Cursores-en-PLSQL.aspx>

8.1.4.2.3.1 Cursores explícitos avanzados

Para procesar instrucciones SELECT que devuelvan más de una fila, son necesarios cursores explícitos combinados con una estructura de bloque.

Un cursor admite el uso de parámetros. Los parámetros deben declararse junto con el cursor.

El siguiente ejemplo muestra la declaración de un cursor con un parámetro, identificado por p_continente.

```
declare  
cursor c_paises (p_continente IN VARCHAR2) is  
SELECT CO_PAIS, DESCRIPCION  
FROM PAISES  
  
WHERE CONTINENTE = p_continente;  
begin  
/* Sentencias del bloque ... */  
end;
```

El siguiente diagrama representa como se procesa una instrucción SQL a través de un cursor.

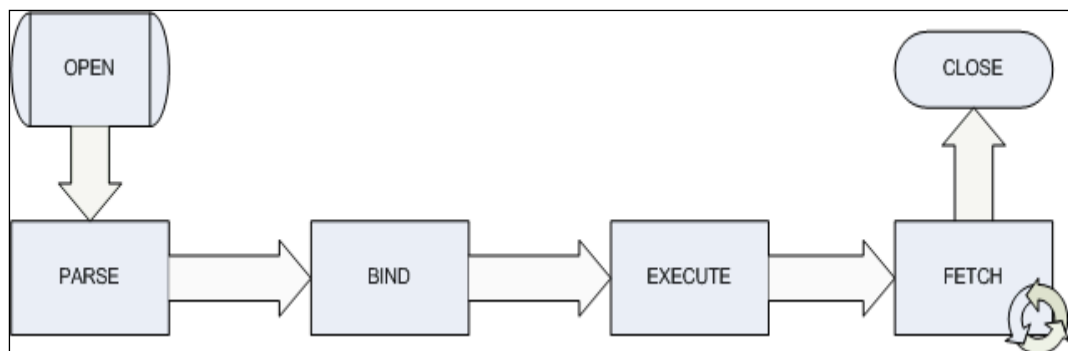


Fig. N°17: Fases para procesar una instrucción SQL

Fuente: <http://www.devjoker.com/contenidos/Tutorial-PLSQL/32/Cursores-en-PLSQL.aspx>

8.1.4.2.3.2 Manejo de excepciones

En PL/SQL una advertencia o condición de error es llamada una excepción.

Las excepciones se controlan dentro de su propio bloque. La estructura de bloque de una excepción se muestra a continuación.

```
DECLARE
--                               Declaraciones
BEGIN
--                               Ejecución
EXCEPTION
--                               Excepción
END;
```

“Cuando ocurre un error, se ejecuta la porción del programa marcada por el bloque **EXCEPTION**, transfiriéndose el control a ese bloque de sentencias”.⁹

El siguiente ejemplo muestra un bloque de excepciones que captura las excepciones **NO_DATA_FOUND** y **ZERO_DIVIDE**. Cualquier otra excepción será capturada en el bloque **WHEN OTHERS THEN**.

```
DECLARE
--                               Declaraciones
BEGIN
--                               Ejecución
EXCEPTION
WHEN      NO_DATA_FOUND      THEN
-- Se ejecuta cuando ocurre una excepción de tipo
NO_DATA_FOUND
WHEN      ZERO_DIVIDE      THEN
-- Se ejecuta cuando ocurre una excepción de tipo
ZERO_DIVIDE

WHEN      OTHERS      THEN
-- Se ejecuta cuando ocurre una excepción de un tipo
no
--      en      los      bloques      tratado
--                               anteriores

END;
```

⁹ Información obtenida de:
Devjoker, Excepciones , julio 2010

<http://www.devjoker.com/contenidos/catss/48/Excepciones-en-PLSQL.aspx>

8.1.4.2.3.3 Sentencia raise

La sentencia **RAISE** permite lanzar una excepción en forma explícita. Es posible utilizar esta sentencia en cualquier lugar que se encuentre dentro del alcance de la excepción.

```
DECLARE
-- Declaramos una excepción identificada por VALOR_NEGATIVO
VALOR_NEGATIVO EXCEPTION;
valor NUMBER;
BEGIN
-- Ejecución
valor := -1;
IF valor < 0 THEN
    RAISE VALOR_NEGATIVO;
END IF;
EXCEPTION
-- Excepción
WHEN VALOR_NEGATIVO THEN
    dbms_output.put_line('El valor no puede ser negativo');
END;
```

Con la sentencia **RAISE** podemos lanzar una excepción definida por el usuario o predefinida, siendo el comportamiento habitual lanzar excepciones definidas por el usuario.

8.1.4.2.3.4 SQLCODE y SQLERRM

Al manejar una excepción es posible usar las funciones predefinidas **SQLCode** y **SQLERRM** para aclarar al usuario la situación de error acontecida.

SQLcode devuelve el número del error de Oracle y un 0 (cero) en caso de éxito al ejecutarse una sentencia SQL.

Por otra parte, **SQLERRM** devuelve el correspondiente mensaje de error.

Estas funciones son muy útiles cuando se utilizan en el bloque de excepciones, para aclarar el significado de la excepción **OTHERS**.

Estas funciones no pueden ser utilizadas directamente en una sentencia SQL, pero sí se puede asignar su valor a alguna variable de programa y luego usar esta última en alguna sentencia.

```
DECLARE
err_num NUMBER;
err_msg VARCHAR2(255);
result NUMBER;
BEGIN
SELECT 1/0 INTO result
FROM DUAL;
EXCEPTION
WHEN OTHERS THEN

err_num := SQLCODE;
err_msg := SQLERRM;
DBMS_OUTPUT.put_line('Error:'||TO_CHAR(err_num));
DBMS_OUTPUT.put_line(err_msg);
END;
```

También es posible entregarle a la función **SQLERRM** un número negativo que represente un error de Oracle y ésta devolverá el mensaje asociado.

```
DECLARE
msg VARCHAR2(255);
BEGIN
msg := SQLERRM(-1403);
DBMS_OUTPUT.put_line(MSG);
END;
```

8.1.4.2.4 Procedimientos almacenados

“Un procedimiento es un subprograma que ejecuta una acción específica y que no devuelve ningún valor. Un procedimiento tiene un nombre, un conjunto de parámetros (opcional) y un bloque de código”.¹⁰

La sintaxis de un procedimiento almacenado es la siguiente:

¹⁰ Información obtenida de:

Devjoker, Procedimientos almacenados, 1 julio 2010

<http://www.devjoker.com/contenidos/catss/52/Procedimientos-almacenados-en-PLSQL.aspx>

```

CREATE [OR REPLACE]
PROCEDURE <procedure_name> [(<param1> [IN|OUT|IN OUT] <type>,
                             <param2> [IN|OUT|IN OUT] <type>, ...)]
IS
    -- Declaración de variables locales
BEGIN
    -- Sentencias
[EXCEPTION]
    -- Sentencias control de excepción
END [<procedure_name>];

```

El uso de OR REPLACE permite sobrescribir un procedimiento existente. Si se omite, y el procedimiento existe, se producirá, un error.

La sintaxis es muy parecida a la de un bloque anónimo, salvo porque se reemplaza la sección **DECLARE** por la secuencia **PROCEDURE... IS** en la especificación del procedimiento.

Debemos especificar el tipo de datos de cada parámetro. **Al especificar el tipo de dato del parámetro no debemos especificar la longitud del tipo.**

Los parámetros pueden ser de entrada (**IN**), de salida (**OUT**) o de entrada salida (**IN OUT**). El valor por defecto es **IN**, y se toma ese valor en caso de que no especifiquemos nada.

```

CREATE OR REPLACE
PROCEDURE Actualiza_Saldo(cuenta NUMBER,
                           new_saldo NUMBER)
IS
    -- Declaración de variables locales
BEGIN
    -- Sentencias
    UPDATE SALDOS_CUENTAS
        SET SALDO = new_saldo,
            FX_ACTUALIZACION = SYSDATE
        WHERE CO_CUENTA = cuenta;

END Actualiza_Saldo;

```

También podemos asignar un valor por defecto a los parámetros, utilizando la cláusula **DEFAULT** o el operador de asignación (**:=**).

```

CREATE OR REPLACE
PROCEDURE Actualiza_Saldo(cuenta NUMBER,
                           new_saldo NUMBER DEFAULT 10 )
IS
  -- Declaración de variables locales
BEGIN
  -- Sentencias
  UPDATE SALDOS_CUENTAS
    SET SALDO = new_saldo,
        FX_ACTUALIZACION = SYSDATE
  WHERE CO_CUENTA = cuenta;

END Actualiza_Saldo;

```

Existen dos formas de pasar argumentos a un procedimiento almacenado a la hora de ejecutarlo (en realidad es válido para cualquier subprograma). Estas son:

- **Notación posicional:** Se pasan los valores de los parámetros en el mismo orden en que el procedure los define.

```

BEGIN
  Actualiza_Saldo(200501,2500);
  COMMIT;
END;

```

- **Notación nominal:** Se pasan los valores en cualquier orden nombrando explícitamente el parámetro.

```

BEGIN
  Actualiza_Saldo(cuenta => 200501,new_saldo => 2500);
  COMMIT;
END;

```


8.1.4.2.5 Funciones almacenadas

Una función es un subprograma que devuelve un valor.

La sintaxis para construir funciones es la siguiente:

```
CREATE [OR REPLACE]  
FUNCTION <fn_name>[(<param1> IN <type>, <param2> IN <type>, ...)]  
RETURN <return_type>  
IS  
    result <return_type>;  
BEGIN  
  
    return(result);  
[EXCEPTION]  
    -- Sentencias control de excepción  
END [<fn_name>;];
```

El uso de OR REPLACE permite sobrescribir una función existente. Si se omite, y la función existe, se producirá, un error.

“La sintaxis de los parámetros es la misma que en los procedimientos almacenado, exceptuando que solo pueden ser de entrada”.¹¹

Ejemplo:

```
CREATE OR REPLACE  
FUNCTION fn_Obtener_Precio(p_producto VARCHAR2)  
RETURN NUMBER  
IS  
    result NUMBER;  
BEGIN  
    SELECT PRECIO INTO result  
    FROM PRECIOS_PRODUCTOS  
    WHERE CO_PRODUCTO = p_producto;  
    return(result);  
EXCEPTION  
WHEN NO_DATA_FOUND THEN  
    return 0;  
END ;
```

¹¹ Información Obtenida de:
Devjoker, funciones en plsql, agosto 2009

<http://www.devjoker.com/contenidos/catss/53/Funciones-en-PLSQL.aspx>

Si el sistema nos indica que el la función se ha creado con errores de compilación podemos ver estos errores de compilación con la orden **SHOW ERRORS** en SQL *Plus.

Una vez creada y compilada la función podemos ejecutarla de la siguiente forma:

```
DECLARE  
  Valor NUMBER;  
BEGIN  
  Valor := fn_Obtener_Precio('000100');  
END;
```

Las funciones pueden utilizarse en sentencias SQL de manipulación de datos (SELECT, UPDATE, INSERT y DELETE):

```
SELECT CO_PRODUCTO,  
        DESCRIPCION,  
        fn_Obtener_Precio(CO_PRODUCTO)  
FROM PRODUCTOS;
```

8.1.5 Oracle forms developer 10g

8.1.5.1. Introducción a oracle forms developer

Oracle Forms es una herramienta de *Oracle* que permite, de un modo sencillo y eficiente, diseñar pantallas para el ingreso, modificaciones, bajas y consultas de registros.

El usuario podrá, una vez definida la forma, trabajar con ella sin necesidad de generar códigos, dado que Oracle trae incorporado un conjunto de procedimientos y funciones.

8.1.5.2 Forms

Herramienta cliente-servidor que permite modificar, crear y borrar datos en la BBDD mediante la interacción del lenguaje PL/SQL y la programación visual orientada a eventos. **Ver figura # 14**

Forms se compone de tres programas principales

- Forms Designer crea las pantallas y da funcionalidad a las mismas.
- Forms Generate: convierte el fichero designer en un fichero seudoejecutable.
- Forms Runforms: ejecuta el seudoejecutable

Designer genera tres tipos de ficheros principales:

- .fmb: incluye todas las partes que intervienen en el desarrollo, pantallas, procedimientos, funcionalidad, etc.
- .fmt: contiene el fmb traducido al lenguaje C
- .fmx: seudoejecutable

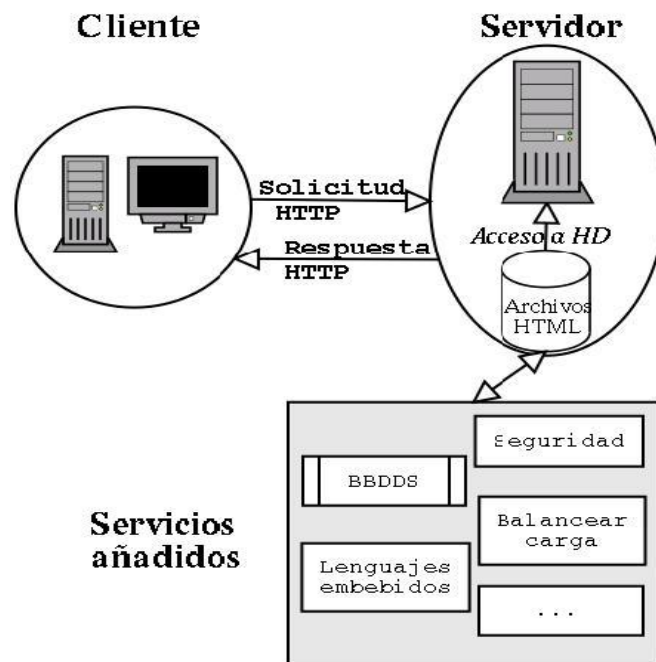


Fig. N° 18: Arquitectura de Oracle Aplicación Server

Fuente: <http://edgarramirez.files.wordpress.com/2008/11/dmr.jpg>

8.1.5.3 Componentes de oracle developer suite

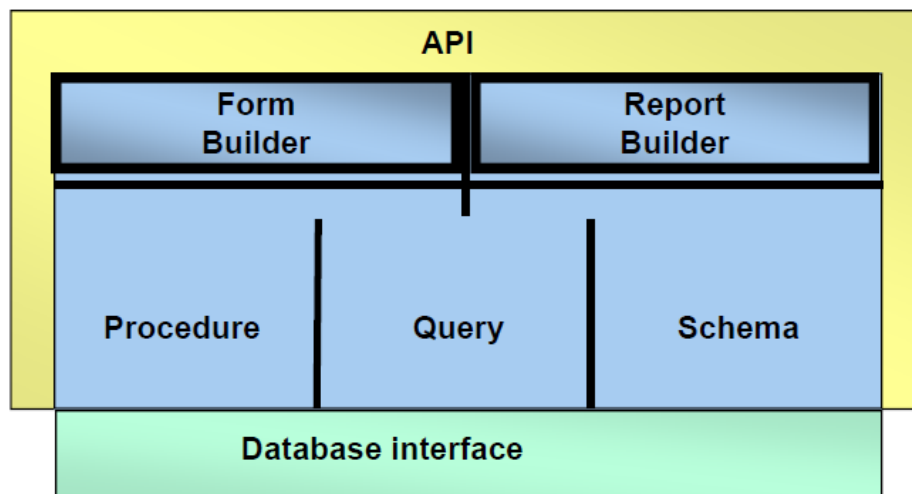


Fig. N°. 19: Componentes de Oracle Developer Suite

Fuente: Folleto del seminario Ing. Juan F. Vicuña P.

8.1.5.4. Variables de ambiente de Oracle Forms

- Forms_path
- Reports_path
- Ui_icon
- Oracle_home

8.1.5.5. El navegador de objetos

El navegador de objetos es una estructura similar a un árbol, desde la cual se tiene acceso a todos los objetos del formulario (bloques, alertas, lienzos, ventanas,...). Se puede seleccionar el botón "+" para agregar o "-" para eliminar el objeto.

Elementos: Objetos y propiedades

Objeto es todo aquello que se pueda insertar en una pantalla y tenga **Propiedades:** cuadros de texto, botones, etc.

Cuando se genera un objeto automáticamente se crean sus propiedades que tomaran un valor por defecto inicial dado por Forms, dichas propiedades se podrán modificar en tiempo de diseño o en tiempo de ejecución dependiendo del objeto y de la propiedad.

Los objetos se dividen en:

- **Ítems:** interfaz de comunicación entre el usuario y la aplicación, textos, Botones, listas, etc.
- **Bloques:** agrupaciones de ítems. Estos bloques pueden estar asociados a tablas o no, es mas todo ítem debe estar asociado a un bloque de tal modo que será necesario crear al menos un bloque por si solo existe un ítem en la aplicación.

8.1.5.6. Trigger

Permiten dar funcionalidad a los eventos (instrucciones asociadas a un ítem cuando se produce un evento). No deben confundirse con los eventos de forma aislada, un trigger es un conjunto formado por un evento y unas instrucciones, es un concepto algo distinto del simple concepto de evento que existe en otros entornos de desarrollo visuales, debe ser tratado de una forma más conceptual que practica, su dominio requiere un poco de experiencia. No obstante podemos encontrar referencias a un Trigger solamente como disparador o evento.

Los triggers principales en Forms son:

Key: apenas utilizado en Forms, son ejecutados al pulsar una tecla o Combinación de teclas, principalmente se usan en entornos de modo texto.

When: el más utilizado, son desencadenados por múltiples eventos

On: interviene en acciones por defecto del sistema, ON INSERT, ON DELETE,..

pre: se disparan antes de ejecutar el evento correspondiente. PRE INSERT, por ejemplo para comprobar condiciones específicas de inserción en función de unos valores.

post: misma funcionalidad que el anterior pero a la inversa temporalmente
POST INSERT, POST COMMIT.

8.1.5.7. Bloques basados en tablas

Los bloques son conjunto de elementos o ítems y que principalmente hacen referencia a los atributos de una tabla. También pueden existir elementos de un bloque que no se traducen en atributos de la tabla. Cada bloque tiene comportamientos diferentes dependiendo de los eventos que se disparen; por lo tanto cada bloque tiene sus propios disparadores o triggers.

8.1.5.8. Canvas o lienzos

Las canvas o lienzos son los contenedores de los bloques. Son los diseños de las pantallas y es en éstos donde se organiza la disposición de los botones, cuadros, y demás controles como serán vistos por el usuario. Dentro de un lienzo se pueden colocar varios bloques, no hay un límite.

8.1.5.9. Oracle reports

Oracle reports 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 jsp.

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.

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.

En la figura No. 20, se pueden ver los elementos que componen un reporte.

- **El modelo de datos** es un área donde se define principalmente el contenido del reporte; es decir, la definición de consultas, cálculos, funciones que retornarán los datos para el respectivo informe.
- **Disposición de papel:** es un área donde se define principalmente la parte visual del reporte; es decir, se definen los tipos de letra, tamaños, ubicación de las columnas, colores, espacios, encabezados, y en general, todo aquello que tiene que ver con la presentación del informe.
- **La pantalla de parámetros:** es un área donde se puede personalizar la entrada de los parámetros al reporte. Por ejemplo, si se tiene un reporte acerca de los elementos vendidos por una empresa, pudiesen ser parámetros la fecha inicial y la fecha final para los cuales se desea conocer esta información.

- **Disparadores de informes:** es un bloque donde se definen acciones que se disparan principalmente antes de que el reporte sea generado o después de la ejecución del mismo dependiendo de las políticas del negocio.

Unidades de programa y Bibliotecas conectadas al igual que Oracle forms es un área donde se pueden definir en la primera, procedimientos, funciones, paquetes que pueden ser utilizados desde el modelos de datos, y en el segundo, enlaces a librerías o bibliotecas pll, donde se encuentren las funciones, procedimientos o paquetes.

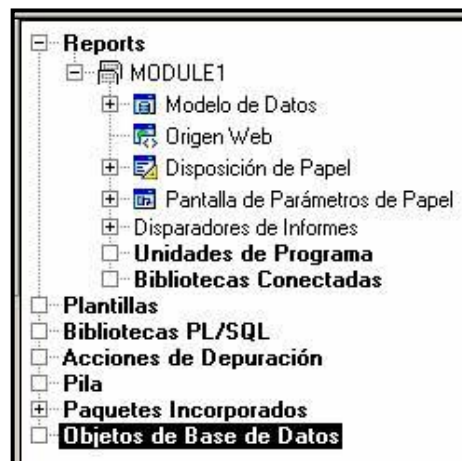


Fig. N° 20: Elemento de un reporte

Fuente: Oracle report

8.1.6. Estándares para desarrollo

8.1.6.1. Creación de objetos

Tipo de Objeto	SINONIMO	Nombre Archivo
BASE DE DATOS		SIGALAB
SHEMAS	(SCH)	SCH_LAB
TABLESPACE		LABORATORIO
Creación de Tablas		clientes
Creación de Claves Primarias	(PK)	PK_000
Creación de Foreign	(FK)	FK_000
Creación de Índices	(IDX)	IDX_000

8.1.6.2. Sinónimos

Para este objeto se recomienda asumir el nombre del objeto dueño, salvo el caso de alguna excepción, se ha considerado por ahora que sea público public.

Ejemplo:

```
create public synonym CLIENTE for SCH_LAB.CLIENTE
```

8.1.6.3. Declaración de variables en PL/SQL

Identificación:

a) Prefijo que indica el alcance de la variable y el tipo de dato que maneja.

Alcance	Tipo Dato
G : Global	V : Varchar2
L : Local	I : Integer
P : Parámetro	N : Number
	D : Date
	B : Boolean

b) Nombre de la variable, lo más descriptivos posible, no mayor a 30 caracteres

Ejemplos:

Nombre Variable	Características
Gv_Aplicacion	Global de tipo Varchar2
Li_Empleado	Local de tipo entero
Pn_ValorTotal	Parámetro de Tipo Numérico

El prefijo de alcance y la primera letra de la variable deben estar en mayúscula.

8.1.6.4.Cursores

Canalice todo tipo de recuperación de datos a través de este mecanismo, su definición se explica a continuación.

Identificación:

Como prefijo se utilizaran los caracteres C_ seguido del nombre del cursor siendo este los más descriptivos posibles.

Ejemplo: Cursor **C_DetalleValores**

De manejar parámetros, estos se identifican con la letra **C**, el tipo de dato y en nombre. Si existen más de uno, declarar estos en líneas consecutivas como lo muestra el ejemplo.

Ejemplo:

Cursor **C_DetalleValores** (**Pi_PAMETRO1** INTEGER,
Pv_PARAMETRO2 VARCHAR2) Is

Select

From

Where

Herramientas

Forms (Desarrollo)

Referido a los pasos iniciales en la elaboración de una forma

a) **Identificación o Nombre:**

Prefijo de 3 letras como referencia del modulo ó aplicación, más el nombre referencial del contenido de la forma.

Ejemplo: FRM_FACTURA

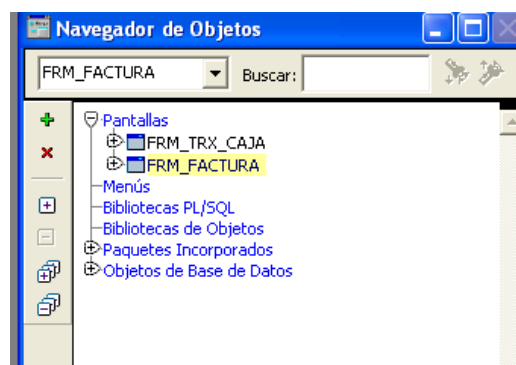


Fig. N° 21: Formulario de Factura

Fuente: Oracle Form

8.1.6.7. Elementos de forms

Triggers estos eventos son destinados para: validaciones, navegación, registro de transacción (consulta, ingreso, modificación), entre otros.

Lo recomendado en código pl/sql, dentro de estos elementos es que no exista un exceso de codificación en ellos, de ser necesario proceda a la construcción de Program Units (procedimientos, funciones, paquetes), de forma que el evento sea un ejecutor de estos.

Reglas de Identificación

Program Unit

Nombre	Tipo
PU_P_XYZ	Procedure
PU_F_XYZ	Function
PU_K_XYZ	Package

Ejemplo: PU_P_XYZ (Pi_CodigoEmpresa Integer, ..
Pv_CodigoAplicacion Varchar2) is

8.1.6.8. Elementos de la base

Usualmente el que más utilizaremos es el **Package**, la justificación de su construcción obedece a que este pueda ser usado en varias formas o procesos. Cumpliendo con esto unos de los principales requisitos: Ser lo más **general posible**.

Una forma sugerida, consiste en usar un número de parámetros de entrada (**in**), salida (**out**), que permita que el paquete cubra la generalidad de los casos.

Como parte del estándar implantado, se requiere generar Scripts que permitan su construcción como elemento de la base.

Se requieren de dos archivos por paquete a construir, uno para la **definición ó cabecera** (*.hed) y otro para el **cuerpo del paquete** (*.bod).

Nombre del paquete, formado por tres letras de la aplicación, más la letra **K**, que indica que es un package, seguido de un nombre que represente lo general del mismo (en funcionalidad), dado que lo particular lo indica los nombres de los **procedimientos** que lo conforman.

Ejemplo:

Nombre Paquete (Package)	Nombre de Archivo	Funcionalidad
LAB_K_MSG	LAB_K_MSG.hed	Control de Mensajes
	LAB_K_MSG.bod	

Definición del paquete

```
CREATE OR REPLACE PACKAGE LAB_K_MSG AS
```

Procedimientos que lo conforman

```
PROCEDURE LAB_P_OBTIENE_MENSAJE (Pi_CodigoIdioma Integer,
                                Pv_CodigoEntrada Varchar2,
                                Pv_Mensaje out Varchar2,
                                Pv_CodigoSalida out Varchar2,
                                Pv_Adicional out Varchar2,
                                Pv_MsgError out Varchar2);
```

Permisos de usos

grant execute on MGA_K_MSG to PUBLIC

Creación de Record Groups

Las primeras iniciales son RG_ y lo demás es el nombre referente al Lov asignado. Ejemplo:

RG_PERSONAL

Creación de Lovs

Las primeras iniciales son LV_ y lo demás es el nombre referente al Lov asignado en el Record Group que utiliza. Ejemplo:

Si el record group se llama RG_PERSONAL el lov es LOV_PERSONAL.

Nombre de los Bloques

Las primeras iniciales son BL_.

El nombre depende de:

- si es un bloque atachado de una tabla se llama:
BL_NOMBRETABLA
- si es un bloque de trabajo por lo general se llama:
BL_CONTROL

Nombre de los Ítems

Si el ítem es un campo de base se mantiene el mismo nombre.

Si el ítem es de trabajo se llama W_NOMBREITEM

Nombre de los Canvas

Las iniciales del canvas serán las siguientes CV_

Y tendrán un número de serie de acuerdo al orden de creación. Ejemplo:
CV_01 el siguiente será CV_02 etc.....

Nombre de Parámetros de Forma

Se llamarán con la inicial P_ y el nombre del parámetro. Ejemplo:
P_CODIGO_EMPRESA

Reportes

Se llamarán por lo general con el nombre del reporte para saber la forma y el reporte ligado a este.

8.1.6.9. Distribución de carpetas en desarrollo y producción

Carpeta Principal

Prefijo de 3 letras como referencia del modulo ó aplicación Ejemplo:
LAB

Subcarpetas

FRM Donde se graban las formas del módulo

REP Donde se graban los reportes del módulo

Y así se crea una carpeta por cada archivo si son script SCR etc.....

8.2 Marco conceptual

Advenimiento: Llegada de algo o alguien solemne o a algo que se aguarda por un período largo

Autenticación: Verificación de la identidad de una persona, usuario o proceso, para así acceder a determinados recursos o poder realizar determinadas tareas.

Bioquímica: La bioquímica es la ciencia que estudia los componentes químicos de los seres vivos, especialmente las proteínas, carbohidratos, lípidos y ácidos nucleicos, además de otras pequeñas moléculas presentes en las células.

Buffer: Es un área de datos compartida por dispositivos de hardware o procesos de programas que operan a distintas velocidades o con diferentes conjuntos de prioridades. El buffer permite que cada dispositivo o proceso opere sin verse interferido por otro.

Buffer redo log: Los registros Redo describen los cambios realizados en la BD y son escritos en los ficheros redo log para que puedan ser utilizados en las operaciones de recuperación hacia adelante, roll-forward, durante las recuperaciones de la BD. Pero antes de ser escritos en los ficheros redo log son escritos en un caché de la SGA llamado redo log buffer. El servidor escribe periódicamente los registros redo log en los ficheros redo log.

Cliente servidor: Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta.

Cloud computing: La computación en nube, del inglés cloud computing, es un paradigma que permite ofrecer servicios de computación a través de Internet.

Diseño estructurado: En programación y diseño de algoritmos, el diseño estructurado persigue elaborar algoritmos que cumplan la propiedad de modularidad, para ello, dado un problema que se pretende resolver mediante la elaboración de un programa de ordenador, se busca dividir dicho programa en módulos siguiendo los

principios de diseño de Descomposición por refinamientos sucesivos, creación de una Jerarquía modular y elaboración de módulos Independientes.

Estándares de calidad: Criterios respecto de los cuales se realiza el análisis comparativo y se emite un juicio de valor en relación con la calidad de los procesos y productos relativos a un área determinada.

Flexible: Dotados de fácil variación tanto en forma como en tamaño; Que sabe ceder en una negociación o conflicto.

Función: se presenta como un subalgoritmo que forma parte del algoritmo principal, el cual permite resolver una tarea específica.

Gestión administrativa: Conjunto de acciones mediante las cuales el directivo desarrolla sus actividades a través del cumplimiento de las fases del proceso administrativo: Planear, organizar, dirigir, coordinar y controlar.

Hematología: La hematología es la rama de la ciencia médica que se encarga del estudio de los elementos formes de la sangre y sus precursores, así como de los trastornos estructurales y bioquímicos de estos elementos, que puedan conducir a una enfermedad.

Java: Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90.

JPS: Java Server Pages (JSP) es la tecnología para generar páginas web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basado en scripts que utilizan una variante del lenguaje java.

Middleware: El middleware es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas.

Normalización: La hematología es la rama de la ciencia médica que se encarga del estudio de los elementos formes de la sangre y sus precursores, así como de los

trastornos estructurales y bioquímicos de estos elementos, que puedan conducir a una enfermedad.

Objeto: un objeto se define como la unidad que en tiempo de ejecución realiza las tareas de un programa. También a un nivel más básico se define como la instancia de una clase.

Package: es donde se guardan las definiciones de tipos y los objetos que puedan ser utilizados en los diferentes diseños VHDL, serían como los .hpp en C++.

Procedimientos almacenados: Un procedimiento es un subprograma que ejecuta una acción específica y que no devuelve ningún valor. Un procedimiento tiene un nombre, un conjunto de parámetros (opcional) y un bloque de código.

Protocolo: En informática, un protocolo es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red. Un protocolo es una convención o estándar que controla o permite la conexión, comunicación, y transferencia de datos entre dos puntos finales.

Query: Conjunto de condiciones o preguntas usadas para extraer información de la base de datos.

Reporte : REPRESENTACION DE RESULTADOS

Rollback: En tecnologías de base de datos, un rollback es una operación que devuelve a la base de datos a algún estado previo.

Stakeholders: Cualquier persona o entidad afectada por las actividades de una organización. Estas partes interesadas pueden ser internas –dentro de la propia compañía– o externas –clientes, proveedores, competencia, asociaciones.

Trigger: Un trigger (o disparador) en una Base de datos , es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE).

Utopia: El concepto utopía designa la proyección humana de un mundo idealizado.

Workflow: El **flujo de trabajo** (*workflow* en inglés) es el estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas.

XML: Lenguaje de descripción de páginas de Internet (eXtensible Markup Language), diseñado con la intención de reemplazar al estándar actual HTML.

9. Desarrollo de proyecto

9.1 Análisis y diseño de sistema

9.1.1 Mapa de Procesos del laboratorio clínico QUICKLAB

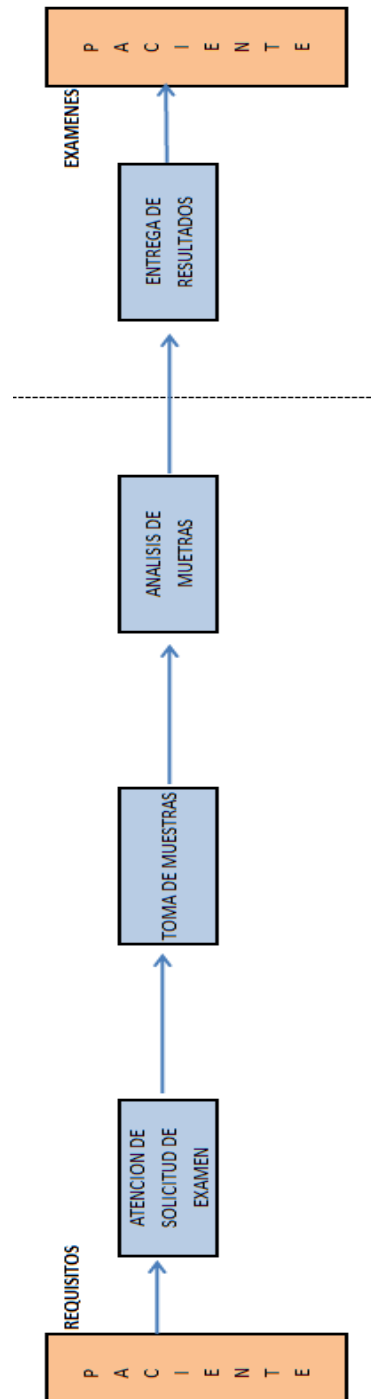


Fig. N° 22: MAPA DE PROCESO QUICKLAB

Fuente: Autores

9.1.2 Cuadro PAT del laboratorio clínico QUICKLAB

Macroproceso	Proceso	ACTIVIDADES	TAREAS
ANALISIS DE MUESTRAS	ANALISIS HEMATOLOGICO	RECEPTAR ORDEN	RECIBIR ORDEN DE EXAMEN
			INGRESAR ORDEN DE EXAMEN AL SISTEMA
		TOMAR MUESTRA	RECIBIR LA ORDEN GENERADA
			OBTENER LA MUESTRA
		ESTUDIAR MUESTRA	RECIBIR LA MUESTRA
			ANALISAR LA MUESTRA
		GENERAR INFORME	RECIBIR RESULTADOS
			INGRESAR RESULTADOS
			VALIDAR RESULTADOS
			GENERAR DOCUMENTOS

Fig. N° 23: DIAGRAMA PAT QUICKLAB

Fuente: Autores

9.1.3 Mapa causa efecto de un análisis clínico

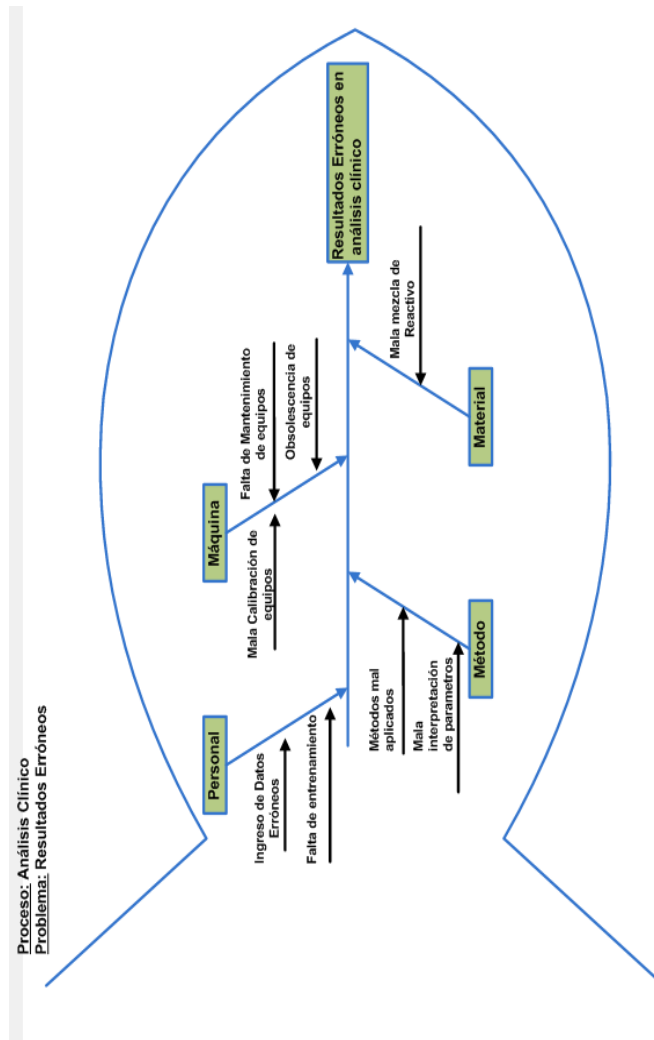
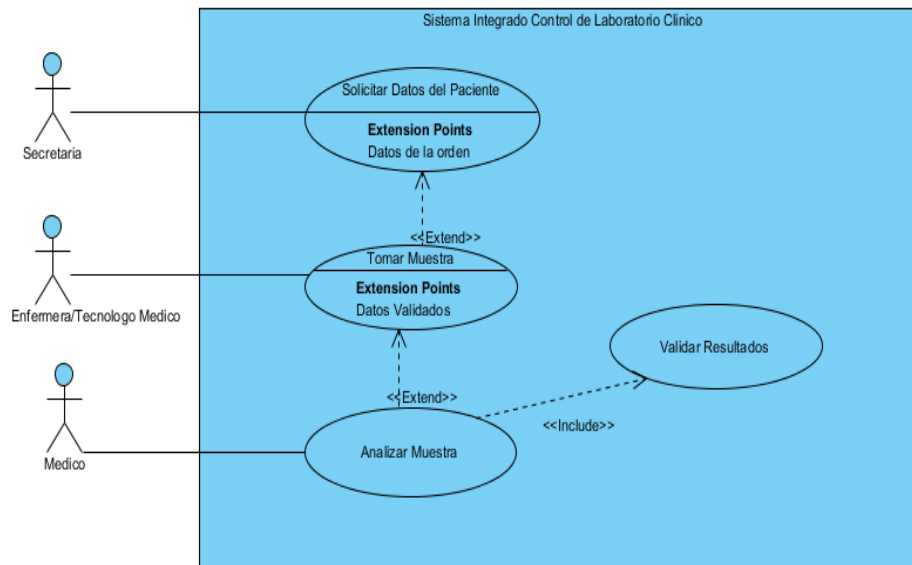


Fig. N° 24: DIAGRAMA CAUSA EFECTO

Fuente: Autores

9.1.4 Diagrama de casos de uso

9.1.4.1 Laboratorio clínico



**Fig. N° 25: DIAGRAMA CASOS DE USO
LABORATORIO CLINICO**

Fuente: Autores

NOMBRE:	LABORATORIO CLINICO
AUTOR:	INTEGRANTES DEL GRUPO
FECHA:	26 DE OCTUBRE 2010
DESCRIPCION:	Proceso de solicitud de examen, toma de muestra y validación de resultados
ACTORES:	Secretaria, Enfermera o Tecnólogo medico, Medico
PRECONDICIONES:	La secretaria debe haber ingresado al paciente en el sistema, y recibir la orden de examen
FLUJO NORMAL:	1.- Secretaria recepta datos del paciente 2.- La enfermera realiza la toma de la muestra al paciente 3.- El Medico analiza la muestra. 4.- se valida resultados
FLUJO ALTERNATIVO:	3.- El médico comprueba los resultados con los rangos establecidos, luego se emite reporte
POSCONDICIONES:	El mensaje ah sido almacenado en el sistema correctamente

9.1.4.2

Facturación

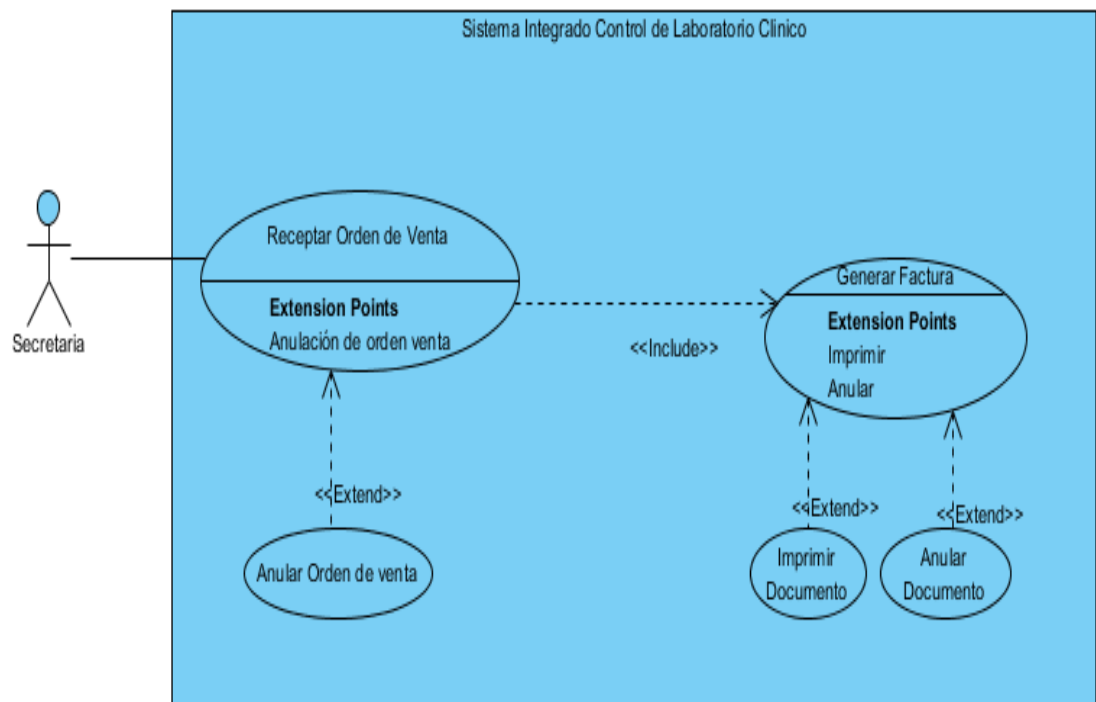


Fig. N° 26: DIAGRAMA CASOS DE USO
FACTURACION

Fuente: Autores

NOMBRE:	FACTURACION
AUTOR:	INTEGRANTES DEL GRUPO
FECHA:	26 DE OCTUBRE 2010
DESCRIPCION:	Proceso de facturación de servicio
ACTORES:	Secretaria
PRECONDICIONES:	Debe existir la orden de venta
FLUJO NORMAL:	1.- Secretaria recepta orden de venta 2.- Se genera la factura 3.- imprimir factura.
FLUJO ALTERNATIVO:	2.- se puede anular la factura
POSCONDICIONES:	Datos almacenados con éxito

9.1.4.3 Cuentas por cobrar

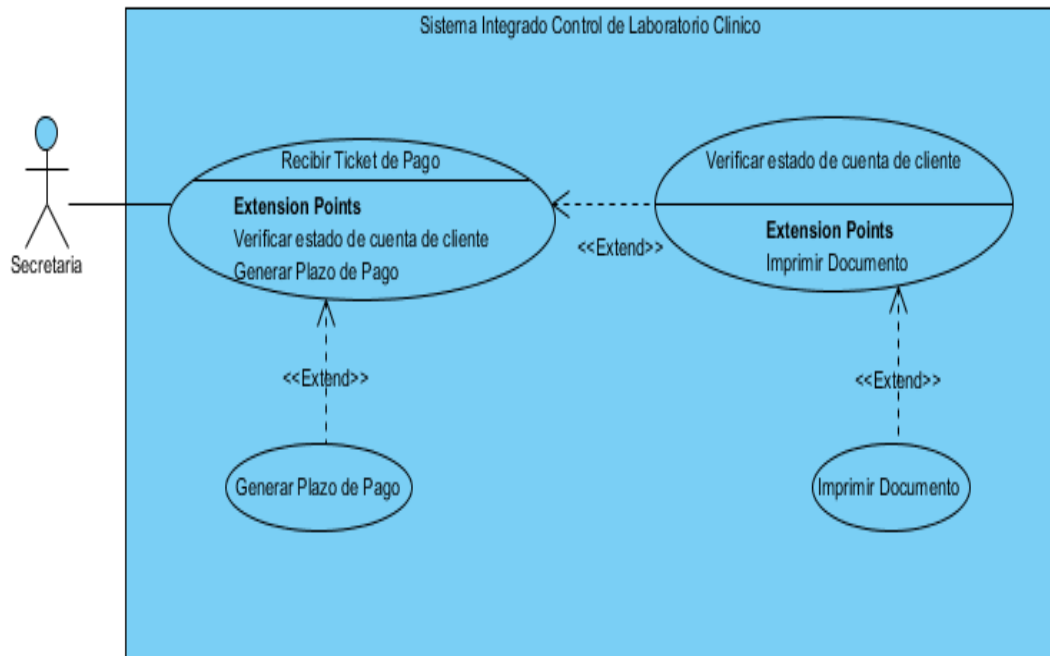


Fig. N° 27: DIAGRAMA CASOS DE USO
CUENTAS POR COBRAR

Fuente: Autores

NOMBRE:	CUENTAS POR COBRAR
AUTOR:	INTEGRANTES DEL GRUPO
FECHA:	26 DE OCTUBRE 2010
DESCRIPCION: Proceso de cuentas por cobrar, verificación de estado de deuda del cliente	
ACTORES: Secretaria	
PRECONDICIONES: Debe existir el cliente registrado en el sistema	
FLUJO NORMAL: 1.- Secretaria recepta ticket de pago 2.- verificación de estado de cuenta de cliente 3.- imprimir imprime documento.	
FLUJO ALTERNATIVO: 2.- verificación de estado de cuenta de cliente, se genera un plazo de pago	
POSCONDICIONES: Datos almacenados con éxito	

9.1.4.4 Caja

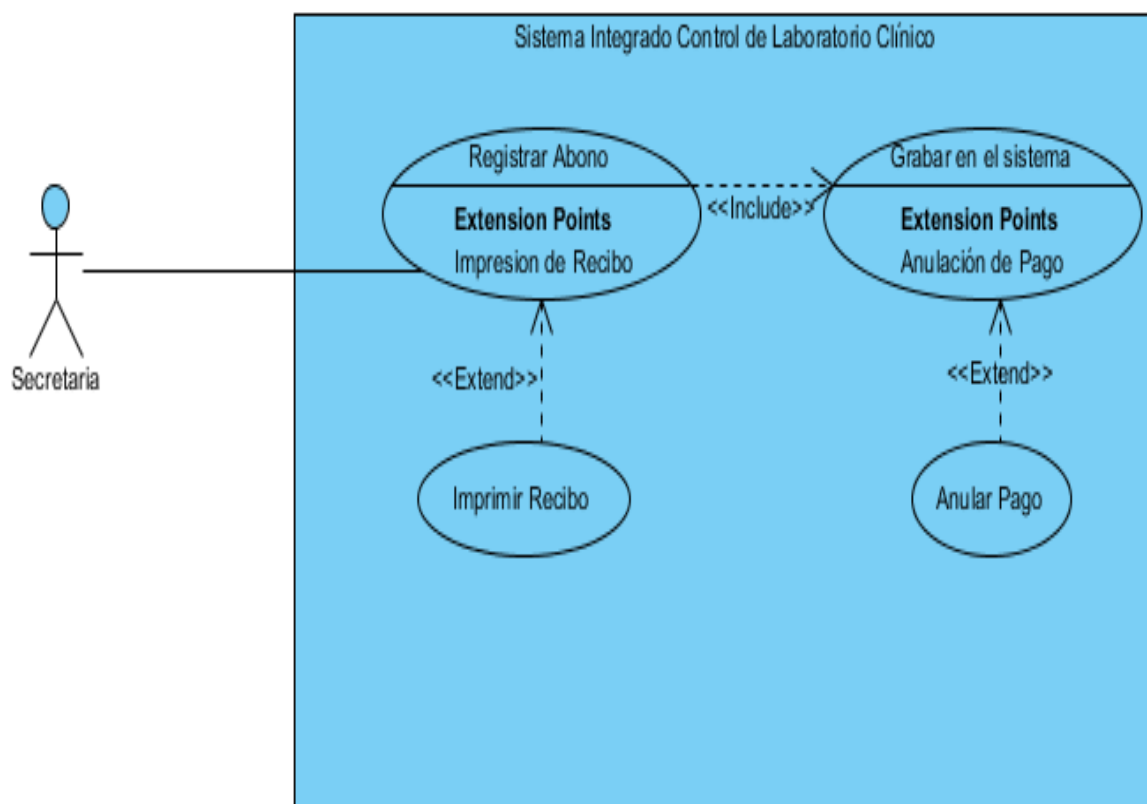


Fig. N° 28: DIAGRAMA CASOS DE USO
CAJA

Fuente: Autores

NOMBRE:	CAJA
AUTOR:	INTEGRANTES DEL GRUPO
FECHA:	26 DE OCTUBRE 2010
DESCRIPCION:	Proceso de caja, registra los abonos a medida que se hagan los pagos
ACTORES:	Secretaria
PRECONDICIONES:	Debe estar el cliente registrado en el sistema
FLUJO NORMAL:	1.- Secretaria registra abono 2.- almacena en el sistema 3.- imprimir recibo de pago.
FLUJO ALTERNATIVO:	2.- almacena el pago, anula el pago
POSCONDICIONES:	Datos almacenados con éxito

9.1.5 Diagrama de clases de Laboratorio clínico

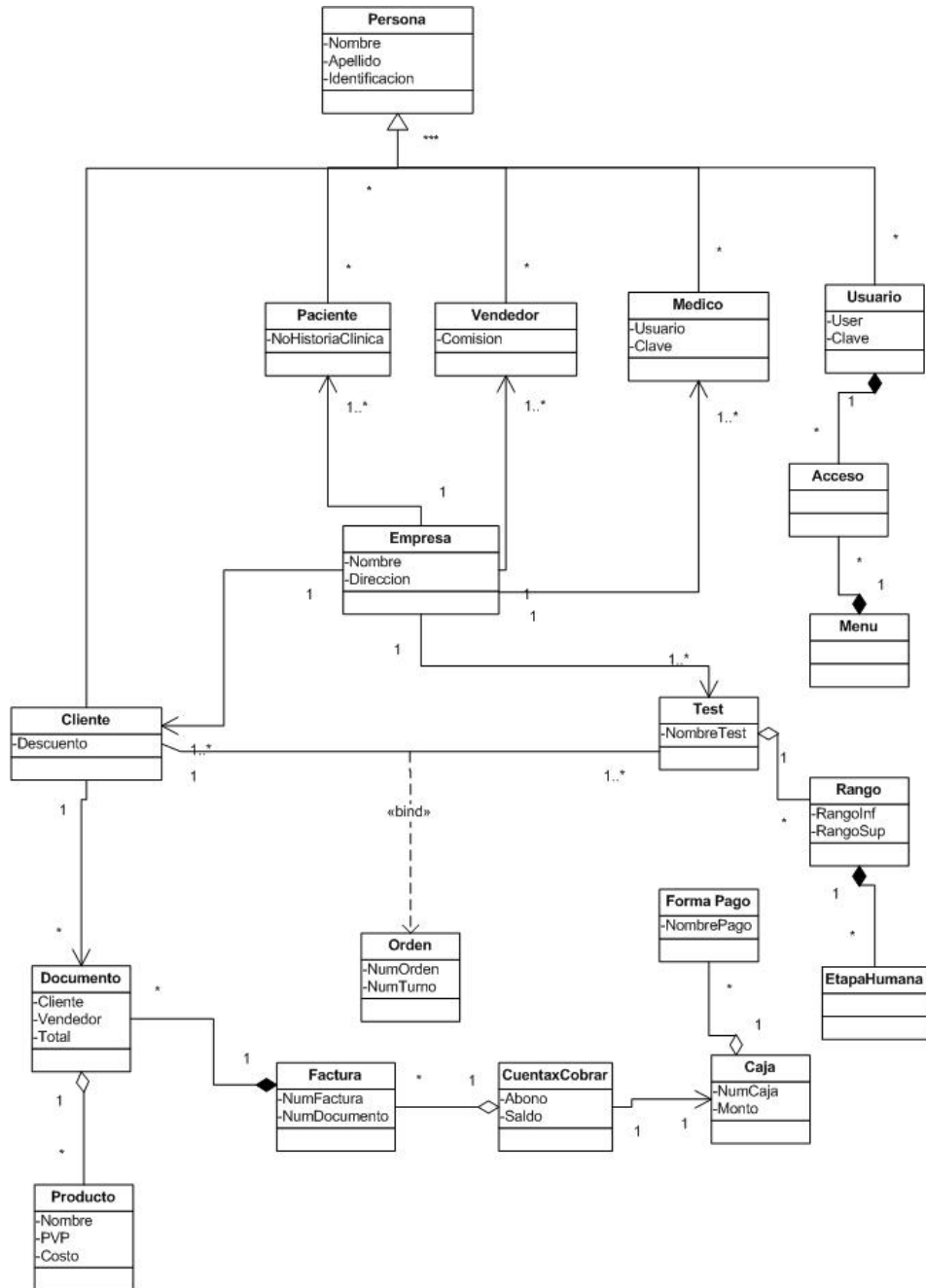


Fig. N° 29: DIAGRAMA CLASES
Fuente: Autores

9.1.6 Diagrama de estado

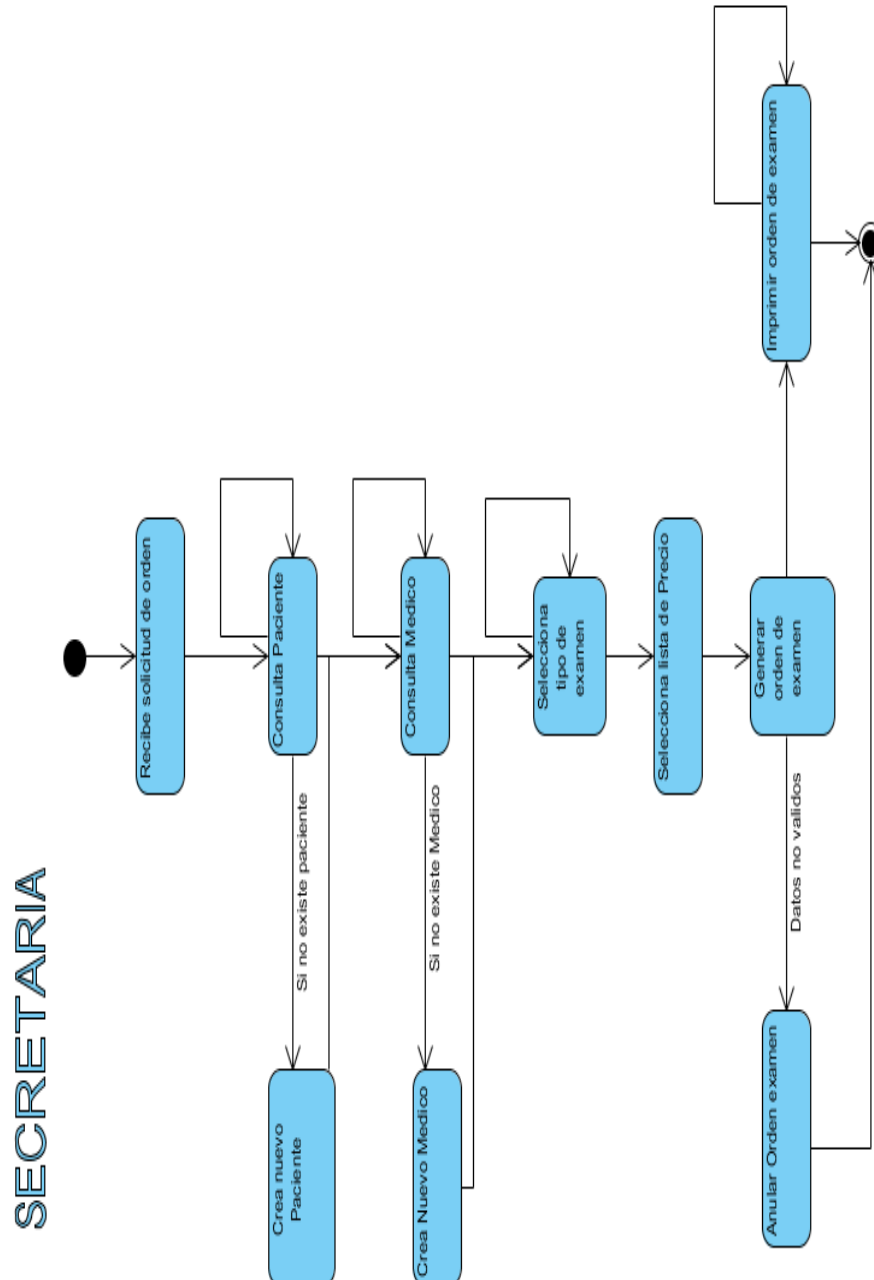


Fig. N° 30: DIAGRAMA ESTADOS
Fuente: Autores

9.1.7 Diagrama de secuencia

9.1.7.1 Laboratorio clínico

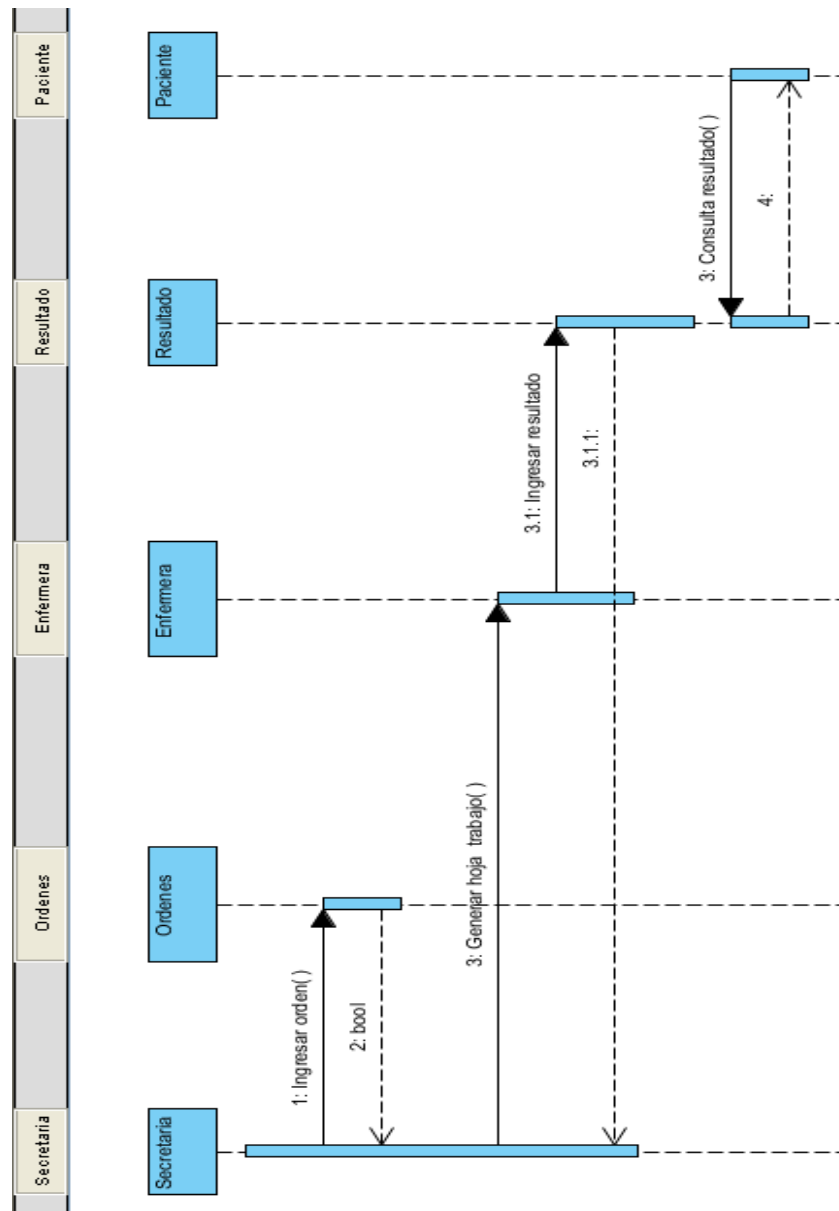


Fig. N° 31: DIAGRAMA SECUENCIA LABORATORIO CLINICO

Fuente: Autores

9.1.8 Gestión administrativa

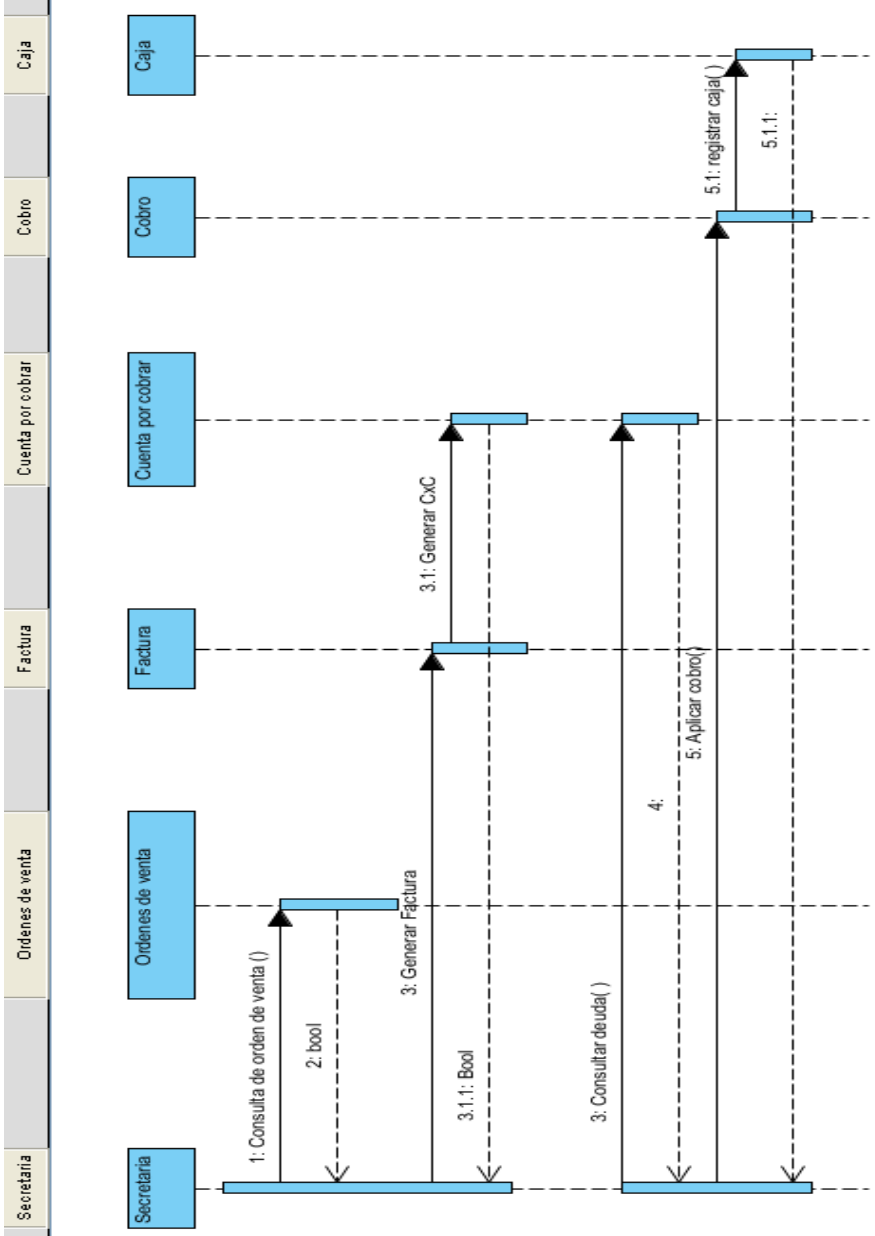


Fig. N° 32: DIAGRAMA SECUENCIA GESTION ADMINISTRATIVA
Fuente: Autores

9.1.9 Diagrama entidad relación del laboratorio clínico

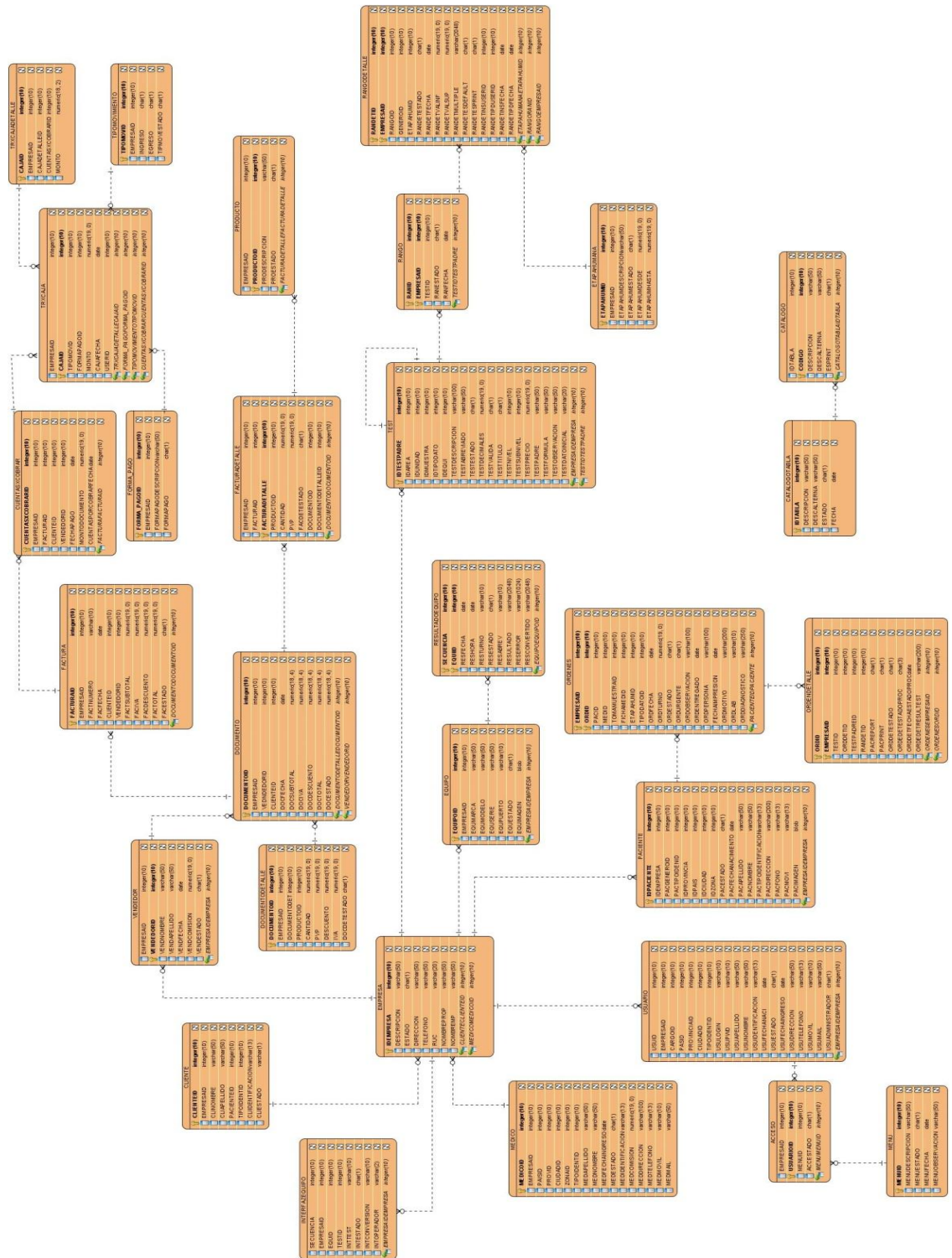


Fig. N° 33: DIAGRAMA ENTIDAD RELACION

Fuente: Autores

9.2 Desarrollo del sistema

9.2.1 DISEÑO LOGICO DEL SISTEMA

9.2.1.1 Tablas del sistema QuickLab

TABLESPACE: LABORATORIO

ESQUEMA: SCH_LAB

9.2.1.1.1. TABLA: CATALOGOTABLA

Se van a almacenar todos los datos de la Empresa.

COLUMNA	TIPO	Comentario
IDTABLA	NUMBER	Esta clave hace referencia a la secuencia de La empresa
DESCRIPCION	VARCHAR2(50)	Almacena alguna referencia de la empresa
DESCALTERNA	VARCHAR2(50)	Almacena el estado de la empresa que puede ser A= Activo, I=inactivo
ESTADO	VARCHAR2(50)	Almacena La dirección de la empresa
FECHA	DATE	Almacena el teléfono de la empresa

Tabla N°. 1: CATALOGOTABLA

Fuente: Autores

Llave Primaria:

PK_CATALOGOTABLA (IDTABLA)

9.2.1.1.2. TABLA: PACIENTE

Se creó esta tabla para grabar los datos del paciente que va a contener el laboratorio clínico

COLUMNA	TIPO	CONCEPTO
IDPACIENTE	Number	Es la clave primaria de la tabla Paciente
IDEMPRESA	Number	Esta clave relaciona con la tabla Empresa
PACGENEROID	Number	Esta clave relaciona con la tabla Genero
PACTIPOIDENID	Number	Esta clave relaciona a la tabla tipo de Identificación (Pasaporte o cedula)
IDPAIS	Number	Esta clave relaciona la tabla país
IDPROVINCIA	Number	Esta clave relaciona la provincia
IDCIUDAD	Number	Esta clave relaciona la ciudad
IDZONA	Number	Esta Clave relaciona la Zona (Norte - Dur)
PACESTADO	Char (1)	Guarda el estado del paciente Activo – Inactivo
PACFECHANACIMIENTO	Date	Guarda la fecha de nacimiento
PACPELLIDO	Varchar2(50)	Guarda el apellido del paciente
PACNOMBRE	Varchar2(50)	Guarda el nombre del paciente
PACTIPOIDENTIFICACION	Varchar2(13)	Guarda el número de Identificación del paciente (Pasaporte - Cedula)
PACDIRECCION	Varchar2(200)	Guarda la dirección del paciente
PACFONO	Varchar2(13)	Guarda el teléfono del paciente
PACMOVI	Varchar2(13)	Guarda el celular del paciente
PACIMAGEN	Blob	Guarda la imagen del paciente

Tabla N°. 2: PACIENTE

Fuente: Autores

Llave Primaria compuesta:

PK_PACIENTE (IDPACIENTE, IDEMPRESA)

Llaves foráneas:

FK_PACIENTE_EMPRESA (IDEMPRESA) referencia SCH_LAB.EMPRESA (IDEMPRESA)

9.2.1.1.3. TABLA: TEST

Se van a almacenar todos los exámenes que realiza el laboratorio clínico

COLUMNA	TIPO	CONCEPTO
IDTESTPADRE	Number	Es la clave primaria de la tabla Test
IDAREA	Number	Esta clave relaciona con la tabla Área
IDUNIDAD	Number	Esta clave relaciona con la tabla Unidad
IDMUESTRA	Number	Esta clave relaciona a la tabla Muestra
IDTIPODATO	Number	Esta clave relaciona la tabla Tipo Dato
IDEQUI	Number	Esta clave relaciona la tabla Equipo
TESTDESCRIPCION	Varchard2(100)	Guarda la descripción del test que va a tomar el paciente
TESTABREVIADO	Varchard2(50)	Guarda la abreviación del test tomado por el paciente
TESTESTADO	Char (1)	Guarda el estado del Test Activo – Inactivo
TESTDECIMALES	number	Guarda la cantidad de decimales con el que se va a presentar
TESTVALIDA	Char(1)	Guarda si es necesario validar el test
TESTTITULO	Char(1)	Titulo S= Examen, N= Prueba
TESTNIVEL	Number	Jerarquía, de los exámenes y pruebas
TESTSUBNIVEL	Number	Ordenamiento de la jerarquía por nivel
TESTPRECIO	Number	Guarda el precio del test
TESTPADRE	Varchar2(50)	Es un campo recursivo, sirve como referencia para saber quién es el padre
TESTFORMULA	Varchar2(50)	Guarda la fórmula del test
TESTOBSERVACION	Varchar2(50)	Guarda una observación del test
TESTDATOINICIAL	Varchar2(20)	Guarda los datos iniciales del Test

Tabla N°. 3: TEST

Fuente: Autores

Llave Primaria compuesta:

PK_TEST (EmpresaId, IdArea)

Llaves foráneas:

FK_TEST_EMPRESA (EMPRESAID) referencia SCH_LAB.EMPRESA (IDEMPRESA)

FK_INTERFAZEQUIPO_TEST

FK_ORDENESDETALLE_TEST

9.2.1.1.4. TABLA: ORDENES

Se van a almacenar todas las órdenes que llegan

COLUMNA	TIPO	Comentario
EMPRESAID	NUMBER	Esta clave hace referencia a la tabla empresa
ORDID	NUMBER	Es la clave primaria de la tabla Ordenes
PACID	NUMBER	Esta clave relaciona con la tabla paciente
MEDID	NUMBER	Esta clave relaciona con la tabla medico
TOMAMUESTRAID	NUMBER	Esta clave relaciona a la tabla Toma de muestra
FICHAMEDID	NUMBER	Esta clave relaciona la tabla ficha medica
ETAPAHUMID	NUMBER	Esta clave relaciona a la tabla de etapa humana
TIPODATOID	NUMBER	Esta clave relaciona la tabla tipo de dato
ORDFECHA	DATE	Guarda la fecha de de la orden
ORDTURNO	NUMBER	Guarda el turno de la orden
ORDESTADO	CHAR(1)	Guarda el estado de la Orden Activo – Inactivo
ORDURGENTE	CHAR(1)	Guarda el tipo de la orden
ORDOBSERVACION	VARCHAR2(100)	Guarda la descripción de la orden
ORDENTREGADO	DATE	Guarda la fecha en la que se entrega la

		orden
ORDPERSONA	VARCHAR2(100)	Persona que retira el examen
FECHAIMPRESION	DATE	Guarda la fecha de impresión de la orden
ORDMOTIVO	VARCHAR2(200)	Guarda el motivo de la orden
ORDLAB	VARCHAR2(10)	Guarda el nombre del laboratorio
ORDDIAGNOSTICO	VARCHAR2(250)	Guarda el diagnostico

Tabla N°. 4: ORDENES

Fuente: Autores

Llave Primaria compuesta:

PK_ORDEN (EMPRESAID, ORDID)

Llaves foráneas:

FK_ORDEN_ETAPAHUMANA (EMPRESAID, ETAPAHUMID) referencia

SCH_LAB.ETAPAHUMANA (EMPRESAID, ETAPAHUMID).

FK_ORDEN_MEDICO (EMPRESAID, MEDID) REFERENCIA

SCH_LAB.MEDICO (EMPRESAID, MEDICOID).

FK_ORDEN_PACIENTE (EMPRESAID, PACID)

REFERENCIASCH_LAB.PACIENTE (EMPRESAID, PACIENTEID).

9.2.1.1.5.TABLA: RESULTADOEQUIPO

Se van a almacenar todos los resultados del equipo que llegan

COLUMNA	TIPO DE DATO	Comentario
SECUENCIA	NUMBER	Esta clave hace referencia a la secuencia del trabajo del equipo
EQUIID	NUMBER	Es la clave primaria de la tabla resultado equipo
RESFECHA	DATE	Esta clave graba la fecha del resultado que vota el equipo
RESHORA	DATE	Graba la hora en la que el equipo vota el resultado
RESTURNO	VARCHAR2(10)	Graba el turno que le toca al paciente
RESESTADO	CHAR(1)	Guarda el estado del resultado A=Activo , I= inactivo
RESABREV	VARCHAR2(10)	Graba la abreviatura del resultado
RESULTADO	VARCHAR2(2048)	Guarda el resultado del examen
RESERROR	VARCHAR2(1024)	Cuando el equipo envía un aviso de error
RESCONVERTIDO	VARCHAR2(2048)	Cuando se aplica una formula

Tabla N°. 5: RESULTADOEQUIPO

Fuente: Autores

Llave Primaria compuesta:

PK_RESULTADOEQUIPO (SECUENCIA)

Llaves foráneas:

FK_RESULTADOEQUIPO_EQUIPO (EQUIPOID) REFERENCIA

SCH_LAB.EQUIPO (EQUIPOID).

9.2.1.1.6. TABLA: RANGODETALLE

Tener referencia de los rangos de los exámenes

COLUMNA	TIPO	CONCEPTO
RANDETID	NUMBER	Esta clave hace referencia a la tabla empresa
EMPRESAID	NUMBER	Es la clave primaria de la tabla Ordenes
RANGOID	NUMBER	Esta clave relaciona con la tabla paciente
GENEROID	NUMBER	Esta clave relaciona con la tabla medico
ETAPAHUMID	NUMBER	Esta clave relaciona a la tabla Etapa Humana
RANDETESTADO	CHAR(1)	Guarda el estado del rango A=Activo, I=Inactivo
RANDETFECHA	DATE	Guarda la fecha del rango
RANDETVALINF	NUMBER	Guarda los parámetros normales inferiores
RANDETVALSUP	NUMBER	Guarda los parámetros normales superiores
RANDETMULTIPLE	VARCHAR2(2048)	Guarda descripción del examen
RANDETESDEFAULT	CHAR(1)	Guarda el parámetro que se presenta por defecto
RANDETESPRINT	CHAR(1)	Guarda el estado si se desea imprimir o no
RANDETINSUSERID	NUMBER	Auditoria para verificar el ingreso del usuario al modificar un rango
RANDETIPDUSERID	NUMBER	Auditoria para verificar la actualización
RANDETINSFECHA	DATE	Verificación de la fecha en la que ingresa el usuario
RANDETIPDFECHA	DATE	Verificación de la fecha en la que actualiza el usuario

Tabla N°. 6: RANGODETALLE

Fuente: Autores

Llave Primaria compuesta:

PK_RANGODET (RANDETID)

Llaves foráneas:

FK_ORDENDETALLE_RANGODETALLE

9.2.1.1.7. TABLA: RANGO

Se van a almacenar todos los rangos que se manejan en los exámenes

COLUMNA	TIPO	CONCEPTO
RANID	NUMBER	Clave primaria de la tabla rango
EMPRESAID	NUMBER	Es la clave primaria de la tabla Empresa
TESTID	NUMBER	Esta clave relaciona con la tabla TEST
RANESTADO	CHAR(1)	Esta clave guarda el estado A=activo, I= Inactivo
RANFECHA	DATE	Graba la fecha del rango

Tabla N°.7: RANGO

Fuente: Autores

Llave Primaria compuesta:

PK_RANGO (EMPRESAID, RANID)

9.2.1.1.8. TABLA: MENU

Se van a almacenar todos los datos almacenados en el menú

COLUMNA	TIPO	CONCEPTO
MENUID	NUMBER	Guarda la secuencia de la tabla menú
MENUDESCRIPCION	VARCHAR2(50)	Guarda la descripción del menú
MENUESTADO	CHAR(1)	Guarda la secuencia del estado A= Activo, I=inactivo
MENUFECHA	DATE	Almacena la fecha en que se registro
MENUOBSERVACION	VARCHAR2(50)	Almacena alguna observación que se registre en el menú

Tabla N°. 8: MENU

Fuente: Autores

Llave Primaria compuesta:

PK_MENU (MENUID)

9.2.1.1.9. TABLA: ORDENDETALLE

Se van a almacenar todos los detalles de las órdenes ingresadas al laboratorio

COLUMNA	TIPO	CONCEPTO
ORDID	NUMBER	Secuencia del la Orden
EMPRESAID	NUMBER	Es la clave hace referencia a la tabla empresa
TESTID	NUMBER	Hace referencia a la tabla Test
ORDDETID	NUMBER	Hace referencia a la tabla orden detalle
TESTPADREID	NUMBER	Esta clave relaciona a la tabla Toma de muestra
RANDETID	NUMBER	Hace referencia a la tabla rango detalle
PACREPORT	CHAR(1)	Guarda si se desea reporte o no
PACPRINT	CHAR(1)	Guarda si se desea impresión o no
ORDDETESTADO	CHAR(1)	Guarda el estado si esta A=activo, I=inactivo
ORDDETESTADO PROC	CHAR(3)	Se define en qué estado se encuentra el proceso A=activo, E= eliminado, P= Procesado, V=validado
ORDDETFECHAE STADOPROC	DATE	Guarda la fecha de la orden
ORDDETRESULT EST	VARCHAR2(200)	Guarda el resultado del test

Tabla N°. 9: ORDENDETALLE

Fuente: Autores

Llave Primaria compuesta:

PK_ORDENDETALLE (EMPRESAID, ORDID, ORDDDETID)

Llaves foráneas:

FK_ORDENDETALLE_RANGODETALLE (RANDETID)

FK_ORDENDETALLE (EMPRESAID, ORDID)

FK_ORDENESDETALLE_TEST (EMPRESAID, TESTID)

9.2.1.1.10. TABLA: MEDICO

Se van a almacenar todos los datos del medico

COLUMNA	TIPO	CONCEPTO
MEDICOID	NUMBER	Almacena la secuencia del medico
EMPRESAID	NUMBER	Es la clave primaria de la tabla empresa
PAISID	NUMBER	Esta clave relaciona con la tabla país
PROVID	NUMBER	Esta clave relaciona con la tabla Provincia
CIUDADID	NUMBER	Esta clave relaciona a la tabla ciudad
ZONAID	NUMBER	Esta clave relaciona a la tabla Zona
TIPOIDENTID	NUMBER	Hace referencia al tipo de identificación
MEDAPELLIDO	VARCHAR2(50)	Almacena el Apellido del medico
MEDNOMBRE	VARCHAR2(50)	Almacena el nombre del medico
MEDFECHAINGRESO	DATE	Almacena la fecha en la que ingresa el medico
MEDESTADO	CHAR(1)	Almacena el estado A= Activo, I= inactivo
MEDIDENTIFICACION	VARCHAR2(13)	Almacena la identificación del medico
MEDCOMISION	NUMBER	Almacena el valor de la comisión que se le va a asignar al medico
MEDDIRECCION	VARCHAR2(100))	Almacena la dirección del medico
MEDTELEFONO	VARCHAR2(13)	Almacena el teléfono del medico
MEDMOVIL	VARCHAR2(10)	Almacena el móvil del medico
MEDMAIL	VARCHAR2(50)	Almacena el mail del medico

Tabla N°. 10: MEDICO

Fuente: Autores

Llave Primaria compuesta:

PK_MEDICO (EMPRESAID, MEDICOID)

Llaves foráneas:

FK_MEDICO_EMPRESA (EMPRESAID) REFERENCIA SCH_LAB.EMPRESA (EMPRESAID)

9.2.1.1.11. TABLA: INTERFAZEQUIPO

Se van a almacenar todos los datos de la interfaz del equipo tales como el puerto que usa el estado de equipo, etc.

COLUMNA	TIPO	CONCEPTO
SECUENCIA	NUMBER	Es el primary key de la tabla
EMPRESAID	NUMBER	Es la clave primaria de la tabla Empresa
EQUID	NUMBER	Guarda el id del equipo
TESTID	NUMBER	Esta clave relaciona con la tabla Test
INTTEST	VARCHAR2(10)	Almacena la relación del valor que da el equipo con el valor de la tabla test
INTESTADO	CHAR(1)	almacena el estado A= Activo , I=inactivo
INTCONVERSION	VARCHAR2(10)	Guarda la conversión en centenas, decenas, miles, millares, etc.
INTOPERADOR	VARCHAR2(2)	Almacena el operador que se va utiliza (+, -, *, /)

Tabla N°. 11: INTERFAZEQUIPO

Fuente: Autores

Llave Primaria compuesta:

PK_INTERFAZEQUIPO (EMPRESAID, SECUENCIA)

Llaves foráneas:

FK_INTERFAZEQUIPO (EMPRESAID, TESTID) REFERENCIA

SCH_LAB.TEST (EMPRESAID, TESTID)

9.2.1.1.12. TABLA: ETAPAHUMANA

Se van a almacenar todos los datos de la etapa humana tales como el Adulto, adolescente, Adulto mayor, etc.

COLUMNA	TIPO	CONCEPTO
ETAPAHUMID	NUMBER	Almacena la secuencia de la etapa humana
EMPRESAID	NUMBER	Es la clave primaria de la tabla EMPRESA
ETAPAHUMDESCRIPCION	VARCHAR2(50)	Este campo almacena una descripción de la etapa humana
ETAPAHUMESTADO	CHAR(1)	Almacena el estado de la etapa humana
ETAPAHUMDESDE	NUMBER	Se almacena datos numéricos en cuanto edad
ETAPAHUMHASTA	NUMBER	Se almacena datos numéricos en cuanto edad

Tabla N°. 12: ETAPAHUMANA

Fuente: Autores

Llave Primaria compuesta:

PK_ETAPAHUMANA (EMPRESAID, ETAPAHUMID)

9.2.1.1.13. TABLA: EQUIPO

Se van a almacenar todos los datos del equipo médico.

COLUMNA	TIPO	CONCEPTO
EQUIPOID	NUMBER	Esta clave hace referencia a la secuencia del equipo
EMPRESAID	NUMBER	Esta clave hace referencia a la tabla empresa
EQUIMARCA	VARCHAR2(50)	Almacena La Marca Del Equipo
EQUIMODELO	VARCHAR2(50)	Almacena el Modelo Del Equipo
EQUIUSERIE	VARCHAR2(50)	Almacena La serie Del Equipo
EQUIPUERTO	VARCHAR2(10)	Almacena el puerto que utiliza el Equipo
EQUIESTADO	CHAR(1)	Almacena el estado Del Equipo
EQUIMAGEN	BLOB	Almacena una imagen del equipo

Tabla N°. 13: EQUIPO

Fuente: Autores

Llave Primaria:

PK_EQUIPO (EQUIPOID)

Llave foránea:

PK_EQUIPO_EMPRESA (EMPRESAID) REFERENCIA SCH_LAB.EMPRESA (EMPRESAID).

9.2.1.1.14. TABLA: EMPRESA

Se van a almacenar todos los datos de la Empresa.

COLUMNA	TIPO DE DATO	Comentario
IDEMPRESA	NUMBER	Esta clave hace referencia a la secuencia de La empresa
DESCRIPCION	VARCHAR2(50)	Almacena alguna referencia de la empresa
ESTADO	CHAR(1)	Almacena el estado de la empresa que puede ser A= Activo, I=inactivo
DIRECCION	VARCHAR2(50)	Almacena La dirección de la empresa
TELEFONO	VARCHAR2(50)	Almacena el teléfono de la empresa
RUC	VARCHAR2(20)	Almacena el ruc de la empresa
NOMBREPROP	VARCHAR2(50)	Almacena el nombre del propietario de la empresa
NOMBREMP	VARCHAR2(50)	Almacena el nombre de la empresa

Tabla N°. 14: EMPRESA

Fuente: Autores

Llave Primaria:

PK_IDEMPRESA (IDEMPRESA)

Llave foránea:

FK_EQUIPO_EMPRESA - FK_MEDICO_EMPRESA. -

FK_PACIENTE_EMPRESA - FK_TEST_EMPRESA

FK_USUARIO_EMPRESA

9.2.1.1.15. TABLA: CATALOGO

Se van a almacenar todos los datos de la Empresa.

COLUMNA	TIPO	CONCEPTO
IDTABLA	NUMBER	Esta clave hace referencia a la secuencia de La empresa
CODIGO	NUMBER	Almacena alguna referencia de la empresa
DESCRIPCION	VARCHAR2(50)	Almacena el estado de la empresa que puede ser A= Activo, I=inactivo
DESCALTERNA	VARCHAR2(50)	Almacena La dirección de la empresa
ESPRINT	CHAR(1)	Almacena el teléfono de la empresa

Tabla N°. 15: CATALOGO

Fuente: Autores

Llave Primaria:

PK_CATALOGO (IDTABLA, CODIGO)

Llave Foránea:

FK_CATALOGO_CATALOGOTABLA (IDTABLA) REFERENCIA

SCH_LAB.CATALOGO (IDTABLA)

9.2.1.1.16. TABLA: ACCESO

Se van a almacenar todos los datos de la Empresa.

COLUMNA	TIPO	CONCEPTO
IDEMPRESA	NUMBER	Esta clave hace referencia a la secuencia de La empresa
USUARIOID	NUMBER	Almacena alguna referencia de la empresa
MENUID	NUMBER	
ACCESTADO	CHAR(1)	Almacena el estado de la empresa que puede ser A= Activo, I=inactivo

Tabla N°. 16: ACCESO

Fuente: Autores

Llave Primaria Compuesta:

PK_ACCESO (IDEMPRESA, USUARIOID, MENUID)

Llave Foránea:

FK_ACCESO_MENU (MENUID) REFERENCIA SCH_LAB.MENU (MENUID)

FK_ACCESO_USUARIO (USUARIOID) REFERENCIA SCH_LAB.USUARIO (USUID)

9.2.1.1.17. TABLA: USUARIO

Se van a almacenar todos los datos de los usuarios del sistema.

COLUMNA	TIPO	Comentario
USUID	NUMBER	Esta clave hace referencia a la secuencia del usuario
EMPRESAID	NUMBER	Almacena alguna referencia a la tabla empresa
CARGOID	NUMBER	Almacena el cargo
PASID	NUMBER	Almacena el país de donde es el usuario
PROVINCIAID	NUMBER	Almacena la provincia
CIUDADID	NUMBER	Almacena la ciudad
TIPOIDENTID	NUMBER	Almacena el tipo de identificación que posee el usuario
USULOGIN	VARCHAR2(10)	Almacena el usuario
USUPWD	VARCHAR2(10)	Almacena un password
USUAPELLIDO	VARCHAR2(50)	Almacena el apellido del usuario
USUNOMBRE	VARCHAR2(50)	Almacena el nombre del usuario
USUIDENTIFICACION	VARCHAR2(13)	Almacena el tipo de identificación
USUFECHANACI	DATE	Almacena la fecha de nacimiento del usuario
USUESTADO	CHAR(1)	Almacena el estado del usuario A= Activo , I= inactivo
USUFECHAINGRESO	DATE	Almacena la fecha en la que ingreso el usuario a laborar
USUDIRECCION	VARCHAR2(50)	Almacena la dirección del domicilio donde reside el usuario
USUTELEFONO	VARCHAR2(13)	Almacena el teléfono del usuario
USUMOVIL	VARCHAR2(10)	Almacena el numero del móvil del usuario
USUMAIL	VARCHAR2(50)	Almacena el mail del usuario
USUADMINISTRADOR	CHAR(1)	Almacena un estatus si es administrador del sistema o no

Tabla N°. 17: USUARIO

Fuente: Autores

Llave Primaria Compuesta:

PK_USUARIO (USUID)

Llave Foránea:

FK_USUARIO_EMPRESA (EMPRESAID) REFERENCIA SCH_LAB.EMPRESA (IDEMPRESA)

9.2.1.1.18. TABLA: CLIENTE

Se van a almacenar todos los datos de los clientes del laboratorio.

COLUMNA	TIPO	CONCEPTO
EMPRESAID	NUMBER	Esta clave hace referencia a al primary key de la tabla empresa
CLIENTEID	NUMBER	Almacena la secuencia del cliente
CLINOMBRE	VARCHAR2(50)	Guarda el nombre del cliente
CLIAPELLIDO	VARCHAR2(50)	Guarda el apellido del cliente
PACIENTEID	NUMBER	Hace referencia a la tabla paciente
TIPOIDENTID	NUMBER	Hace referencia al tipo de identificación
CLIIDENTIFICACION	VARCHAR2(13)	Almacena el tipo de identificación
CLUESTADO	CHAR(1)	Almacena el estado del cliente A=activo, I=inactivo

Tabla N°. 18: CLIENTE

Fuente: Autores

Llave Primaria:

CLIENTE_PK (EMPRESAID, CLIENTEID)

Llave Foránea:

FK_FACTURA_CLIENTE

9.2.1.1.19. TABLA: DOCUMENTO

Se van a almacenar todos los datos de los documentos que ingresan al laboratorio.

COLUMNA	TIPO	CONCEPTO
EMPRESAID	NUMBER	Hace referencia a la tabla empresa
DOCUMENTOID	NUMBER	Almacena secuencia de documento
VENDEDORID	NUMBER	Almacena el id del vendedor
CLIENTEID	NUMBER	Almacena el id de cliente
DOCFECHA	DATE	Fecha del documento
DOCSUBTOTAL	NUMBER	Subtotal del documento
DOCIVA	NUMBER	Almacena el IVA
DOCDESCUENTO	NUMBER	Almacena algún tipo de descuento
DOCTOTAL	NUMBER	Almacena el total
DOCESTADO	CHAR(1)	Almacena el estado del documento A=activo, I=inactivo

Tabla N°. 19: DOCUMENTO

Fuente: Autores

Llave Primaria:

DOCUMENTO_PK (EMPRESAID, DOCUMENTOID)

Llave Foránea:

FK_DOCUMENTODETALLE_DOCUMENTO

9.2.1.1.20. TABLA: DOCUMENTODETALLE

Se van a almacenar todos los datos del detalle de los documentos que ingresan al laboratorio.

COLUMNA	TIPO	Comentario
EMPRESAID	NUMBER	Hace referencia a la tabla empresa
DOCUMENTOID	NUMBER	Almacena secuencia de documento
DOCUMENTODET	NUMBER	Almacena el id del vendedor
PRODUCTOID	NUMBER	Almacena el id de cliente
CANTIDAD	NUMBER	Fecha del documento
PVP	NUMBER	Subtotal del documento
DESCUENTO	NUMBER	Almacena el IVA
IVA	NUMBER	Almacena algún tipo de descuento
DOCDETESTADO	CHAR(1)	Almacena el estado del documento A=activo, I=inactivo

Tabla N°. 20: DOCUMENTODETALLE

Fuente: Autores

Llave Primaria:

DOCUMENTODETALLE_PK (EMPRESAID, DOCUMENTOID, DOCUMENTODET)

Llave Foránea:

FK_DOCUMENTODETALLE_DOCUMENTO (DOCUMENTOID, EMPRESAID)

SCH_FAC.DOCUMENTO (DOCUMENTOID, EMPRESAID).

FK_DOCUMENTODETALLE_PRODUCTO (EMPRESAID, PRODUCTOID)

SCH_FAC.PRODUCTO (EMPRESAID, PRODUCTOID)

9.2.1.1.21. TABLA: FACTURA

Se van a almacenar todos los datos del detalle de los documentos que ingresan al laboratorio.

COLUMNA	TIPO	CONCEPTO
EMPRESAID	NUMBER(18)	Hace referencia a la tabla empresa
FACTURAID	NUMBER(18)	Almacena secuencia de la factura
FACNUMERO	VARCHAR2(10)	Almacena el numero de factura
FACFECHA	DATE	Almacena la fecha de la factura
CLIENTEID	NUMBER(18)	Almacena el id del cliente
VENDEDORID	NUMBER(18)	Almacena el id del vendedor
FACSUBTOTAL	NUMBER(18,4)	Almacena el subtotal de la factura
FACIVA	NUMBER(18,4)	Almacena el IVA de la factura
FACDESCUENTO	NUMBER(18,4)	Almacena el descuento del la factura
FACTOTAL	NUMBER(18,4)	Almacena el total de la factura
FACESTADO	CHAR(1)	Almacena el estado de la factura A=Activa,I=inactiva

Tabla N°. 21: FACTURA

Fuente: Autores

Llave Primaria:

FACTURA_PK (EMPRESAID, FACTURAID)

Llave Foránea:

FK_FACTURA_CLIENTE (EMPRESAID, CLIENTEID) SCH_FAC.CLIENTE
(EMPRESAID, CLIENTEID)

FK_FACTURA_VENDEDOR (EMPRESAID, VENDEDORID)

SCH_FAC.VENDEDOR (EMPRESAID, VENDEDORID)

9.2.1.1.22. TABLA: FACTURADETALLE

Se van a almacenar todos los datos del detalle de las facturas que ingresan al laboratorio.

COLUMNA	TIPO	Comentario
EMPRESAID	NUMBER(18)	Hace referencia a la tabla empresa
FACTURAID	NUMBER(18)	Almacena secuencia de la factura
FACTURADETALLE	NUMBER(18)	Almacena algún detalle de la factura
PRODUCTOID	NUMBER(18)	Almacena el id del producto
CANTIDAD	NUMBER(18,4)	Almacena la cantidad del producto
PVP	NUMBER(18,4)	Almacena el precio valor publico
FACDETESTADO	CHAR(1)	Almacena el Estado de la factura A= activa, I=inactiva
DOCUMENTOID	NUMBER(18)	Hace referencia a la tabla documento
DOCUMENTODETALLEID	NUMBER(18,4)	Hace referencia a la tabla documento detalle

Tabla N°. 22: FACTURADETALLE

Fuente: Autores

Llave Primaria:

FACTURADETALLE_PK (EMPRESAID, FACTURAID, FACTURADETALLE)

Llave Foránea:

FK_FACTURADETALLE_FACTURA (EMPRESAID, FACTURAID)

SCH_FAC.FACTURA (EMPRESAID, FACTURAID)

FK_FACTURADETALLE_PRODUCTO (EMPRESAID, PRODUCTOID)

SCH_FAC.PRODUCTO (EMPRESAID, PRODUCTOID)

9.2.1.1.23. TABLA: PRODUCTO

Se van a almacenar todos los datos del producto.

COLUMNA	TIPO	Comentario
EMPRESAID	NUMBER(18)	Hace referencia a la tabla empresa
PRODUCTOID	NUMBER(18)	Almacena secuencia de la Producto
PRODESCRIPCION	VARCHAR2(50)	Almacena algún detalle de la producto
PRODESTADO	CHAR(1)	Almacena el estado del producto A=activo, I=inactivo

Tabla N°. 23: PRODUCTO

Fuente: Autores

Llave Primaria:

PRODUCTO_PK (EMPRESAID, PRODUCTOID)

9.2.1.1.24. TABLA: VENDEDOR

Se van a almacenar todos los datos del vendedor del laboratorio.

COLUMNA	TIPO	CONCEPTO
EMPRESAID	NUMBER(18)	Hace referencia a la tabla empresa
VENDEDORID	NUMBER(18)	Almacena secuencia del vendedor
VENDNOMBRE	VARCHAR2(50)	Almacena el nombre del vendedor
VENDAPELLIDO	VARCHAR2(50)	Almacena el apellido del vendedor
VENDFECHA	DATE	Almacena la fecha en la que ingresa el vendedor
VENDCOMISION	NUMBER(18,4)	Almacena la comisión del vendedor
VENDESTADO	CHAR(1)	Almacena el Estado del Vendedor A= activa, I=inactiva

Tabla N°. 24: VENDEDOR

Fuente: Autores

Llave Primaria:

VENDEDOR_PK (EMPRESAID, VENDEDORID)

9.2.1.1.25. TABLA: CUENTAXCOBRAR

Se van a almacenar todos los datos de las cuentas por cobrar

COLUMNA	TIPO	Comentario
EMPRESAID	NUMBER(18)	Hace referencia a la tabla empresa
CUENTAXCOBRARID	NUMBER(18)	Almacena secuencia de la cuenta por cobrar
FACTURAID	NUMBER(18)	Hace referencia al id de factura
CLIENTEID	NUMBER(18)	Hace referencia a la tabla cliente
VENDEDORID	NUMBER(18)	Hace referencia a la tabla cliente
FECHAPAGO	DATE	Almacena La fecha del pago
MONTODOCUMENTO	NUMBER(18,2)	Almacena el monto del documento
SALDODOCUMENTO	NUMBER(18,2)	Almacena el saldo del documento
CUENTAXCOBRARFECHA	DATE	Almacena la fecha en que se genera la cuenta por cobrar

Tabla N°. 25: CUENTAXCOBAR

Fuente: Autores

Llave Primaria:

CUENTAXCOBRAR_PK (EMPRESAID, CUENTAXCOBRARID)

9.2.1.1.26. TABLA: FORMA_PAGO

Se van a almacenar todos los datos de las formas de pago

COLUMNA	TIPO	Comentario
EMPRESAID	NUMBER(18)	Hace referencia a la tabla empresa
FORMA_PAGOID	NUMBER(18)	Almacena secuencia de la forma de pago
FORMAPAGODESCRIPCION	VARCHAR2(50)	Almacena Una Descripción De La Forma De Pago
FORMAPAGOESTADO	CHAR(1)	Almacena El Estado De La Forma De Pago A=Activo I=Inactivo

Tabla N°. 26: FORMA_PAGO

Fuente: Autores

Llave Primaria:

FORMA_PAGO_PK (EMPRESAID, FORMA_PAGOID)

9.2.1.1.27. TABLA: TIPOMOVIMIENTO

Se van a almacenar todos los tipos de movimientos

COLUMNA	TIPO	Comentario
EMPRESAID	NUMBER(18)	Hace referencia a la tabla empresa
TIPOMOVID	NUMBER(18)	Almacena secuencia de la forma de pago
INGRESO	CHAR(1)	Almacena si el tipo de movimiento es ingreso o no S=si, N=no
EGRESO	CHAR(1)	Almacena si el tipo de movimiento es egreso o no S=si, N=no

Tabla N°. 27: TIPOMOVIMIENTO

Fuente: Autores

Llave Primaria:

TIPOMOVIMIENTO_PK (TIPOMOVID, EMPRESAID)

9.2.1.1.28. TABLA: TRXCAJA

Se van a almacenar todas las transacciones de caja

COLUMNA	TIPO	Comentario
EMPRESAID	NUMBER(18)	Hace referencia a la tabla empresa
CAJAID	NUMBER(18)	Almacena secuencia de CAJA
TIPOMOVID	NUMBER(18)	Hace referencia de la tabla tipomovimiento
FORMAPAGOID	NUMBER(18)	Hace referencia a la tabla forma de pago
MONTO	NUMBER(18)	Almacena el monto
CAJAFECHA	La DATE	Almacena la fecha de movimiento en caja
USUARIOID	NUMBER(18)	Hace referencia a la tabla usuario

Tabla N°. 28: TRXCAJA

Fuente: Autores

Llave Primaria:

TRXCAJA_PK (EMPRESAID, CAJAID)

Llave Foránea:

FK_TRXCAJA_TIPOMOV (EMPRESAID, TIPOMOVID)

SCH_CAJA.TIPOMOVIMIIENTO (EMPRESAID, TIPOMOVID).

FK_TRXCAJA_FORMAPAGO (EMPRESAID, FORMAPAGOID)

SCH_CAJA.FORMA_PAGO (EMPRESAID, FORMAPAGOID)

9.2.1.1.29. TABLA: TRXCAJADETALLE

Se van a almacenar todos Los detalles transacciones de caja

COLUMNA	TIPO	Comentario
EMPRESAID	NUMBER(18)	Hace referencia a la tabla empresa
CAJAID	NUMBER(18)	Hace referencia a la tabla trxcaja
CAJADETALLEID	NUMBER(18)	Hace referencia a la tabla caja detalle
CUENTAXCOBRARID	NUMBER(18)	Hace referencia a la tabla cuentas por cobrar
MONTO	NUMBER(18,2)	Almacena el monto

Tabla N°. 29: TRXCAJADETALLE

Fuente: Autores

Llave Primaria:

TRXCAJADETALLE_PK (EMPRESAID, CAJAID, CAJADETALLEID)

Llave Foránea:

FK_TRXCAJADETALLE_CXC (EMPRESAID, CUENTAXCOBRARID)

SCH_CAJA.CUENTAXCOBRAR (EMPRESAID, CUENTAXCOBRARID)

FK_TRXCAJADETALLE_TRXCAJA (EMPRESAID, CAJAID) SCH_CAJA.TRXCAJA
(EMPRESAID, CAJAID)

9.2.1.2 Paquetes del Sistema

PACKAGE	DESCRIPCION
LAB_K_CAJA	El paquete LAB_K_CAJA permite hacer todos los procesos de una caja sean estos registrar ingresos y egresos, consultar y eliminar transacción.
LAB_K_ORDENES	El paquete LAB_K_ORDENES permite hacer todos los procesos de consultar ordenes y actualizar los resultados de las ordenes.
Lab_K_Rango	El paquete LAB_K_RANGO permite hacer todos los procesos sean estos de ingresar, consultar, actualizar y eliminar rangos referenciales por prueba.
Lab_K_General	El paquete LAB_K_GENERAL es de uso general, en este paquete se crearon las funciones comunes del sistema.
	La función OBT_NOMBRE_PACIENTE obtiene el nombre del paciente, le enviamos como parámetro el código del paciente.
	La función OBT_NOMBRE_CLIENTE obtiene el nombre del cliente, le enviamos como parámetro el código del cliente.
	La función OBT_NOMBRE_VENDEDOR obtiene el nombre del vendedor, le enviamos como parámetro el código del vendedor.
	La función OBT_NOMBRE_MEDICO obtiene el nombre del médico, le enviamos como parámetro el código del médico.

	La función OBT_NOMBRE_USUARIO obtiene el nombre del usuario, le enviamos como parámetro el código del usuario registrado del sistema.
	La función OBT_NOMBRE_USULOGIN obtiene el nombre del usuario logeado, le enviamos como parámetro el código del usuario logeado del sistema.
	La función OBT_NOMBRE_FORMAPAGO obtiene la descripción de la forma de pago, le enviamos como parámetro el código de la empresa y el código de la forma de pago.
LAB_K_TEST	Este paquete permite hacer el proceso de mantenimiento de los exámenes tales consultar, insertar, modificar y eliminar el test.
LAB_K_CONSULTARCATALOGO	Paquete que nos sirve de consulta todo nuestro catalogo del sistema.

FUNCTION	DESCRIPCION
F_VALIDA_MAIL	Esta función valida si una cuenta de correo electrónico es válida.
LAB_F_ETAPA_HUMANA	Esta función devuelve la descripción de la etapa humana.
LAB_F_NOMBRE_USUARIO	Devuelve el nombre del usuario del sistema, recibe como parámetros el código de la empresa y el código del usuario.
N_EMPRESA	Devuelve la descripción de la empresa se inicia sesión.

VALIDARCEDULA	Esta función valida si una cedula esta correctamente ingresada y coincide el digito verificador
VAL_RUC	Esta función valida si una RUC esta correctamente ingresada

TRIGGERS	DESCRIPCION
TRGGRABARCLIENTE	Al ingresar un paciente, se genera automáticamente un cliente con los datos básicos del paciente.
TRGGRABARCUECTAXCOBRAR	Al generar una factura, se genera una cuenta por cobrar.
TRGGRABARVENDEDOR	Al ingresar un medico, se genera automáticamente un vendedor con los datos básicos del médico.

9.2.1.2.1 Package LAB_K_CAJA

El paquete LAB_K_CAJA permite hacer todos los procesos de una caja sean estos registrar ingresos y egresos, consultar y eliminar transacción.

```

type CUENTAXCOBRAR_CLIEBTE_RECORD is RECORD(

    EMPRESASID SCH_LAB.TRXCAJADETALLE.EMPRESASID%TYPE,

    CAJAID SCH_LAB.TRXCAJADETALLE.CAJAID%TYPE,

    CUENTAXCOBRARID
SCH_LAB.CUENTAXCOBRAR.CUENTAXCOBRARID%TYPE,

    FACTURASID SCH_LAB.CUENTAXCOBRAR.FACTURASID%TYPE,

    FACNUMERO SCH_LAB.FACTURA.FACNUMERO%TYPE,

```

```

FACFECHA SCH_LAB.FACTURA.FACFECHA%TYPE,

MONTODOCUMENTO
SCH_LAB.CUENTAXCOBRAR.MONTODOCUMENTO%TYPE,

CLIENTEID SCH_LAB.CUENTAXCOBRAR.CLIENTEID%TYPE,

CLIENTE VARCHAR2(200),

FECHAPAGO SCH_LAB.CUENTAXCOBRAR.FECHAPAGO%TYPE,

SALDODOCUMENTO
SCH_LAB.CUENTAXCOBRAR.SALDODOCUMENTO%TYPE,

ABONO NUMBER

);

type TBL_CUENTAXCOBRAR_CLIENTE is table of
SCH_LAB.LAB_K_CAJA.CUENTAXCOBRAR_CLIEBTE_RECORD

index by binary_integer;

type TBL_CAJA is table of sch_lab.TRXCAJA%rowtype index by binary_integer;

PROCEDURE P_CONSULTAR_CXC_CLIENTE(P_EMPRESAID NUMBER,

P_CLIENTEID NUMBER,

P_CAJAID NUMBER,

PO_TBL_CXC_CLIENTE IN OUT
TBL_CUENTAXCOBRAR_CLIENTE
);

```


Este procedimiento permite consultar todos los documentos con deuda (Saldo del documento mayor que cero) que tiene el cliente en cuentas por cobrar. Además permite consultar las transacciones por clientes con su respectivo código de caja.

```
PROCEDURE P_INSERTAR_CAJA(P_EMPRESAID NUMBER,  
                           P_CAJAID NUMBER,  
                           P_TIPOMOVID NUMBER,  
                           P_FORMAPAGOID NUMBER,  
                           P_MONTO NUMBER,  
                           P_CAJAFECHA DATE,  
                           P_USUARIOID NUMBER  
                           );
```

Este procedimiento inserta la transacción de caja con su tipo movimiento (ingreso o egreso), forma de pago (Efectivo, Cheque, etc.), el total del monto.

```
PROCEDURE P_INSERTAR_CAJA_DETALLE(P_EMPRESAID NUMBER,  
                                   P_CAJAID NUMBER,  
                                   P_CUENTAXCOBRARID NUMBER,  
                                   P_MONTO NUMBER,  
                                   PO_ERROR IN OUT VARCHAR2  
                                   );
```

Este procedimiento inserta el detalle de los documentos que van hacer afectados, dichos documentos están enlazados por cuentas por cobrar con su respectivo saldo. En la tabla Cuentas por cobrar se actualiza el saldo del documento siempre y cuando tenga saldo mayor que cero.

```
PROCEDURE P_ELIMINAR_CAJA(P_EMPRESAID NUMBER,  
  
    P_CAJAID NUMBER,  
  
    P_USUARIOID NUMBER  
  
    );  
  
end LAB_K_CAJA;
```

Este procedimiento elimina una transacción de caja y actualiza los saldos del cliente en cuentas por cobrar, toda transacción de caja se podrá eliminar haciendo este una reversión en cuenta por cobrar.

9.2.1.2.2 package LAB_K_ORDENES

create or replace package LAB_K_ORDENES is

El paquete LAB_K_ORDENES permite hacer todos los procesos de consultar ordenes y actualizar los resultados de las ordenes.

```
type ORDENES_record_type is RECORD(  
  
    EMPRESASID SCH_LAB.ORDENDETALLE.EMPRESASID%TYPE,  
  
    ORDID SCH_LAB.ORDENDETALLE.ORDID%TYPE,  
  
    ORDDDETID SCH_LAB.ORDENDETALLE.ORDID%TYPE,  
  
    TESTID SCH_LAB.ORDENDETALLE.ORDID%TYPE,  
  
    TESTDESCRIPCION SCH_LAB.TEST.TESTDESCRIPCION%TYPE,  
  
    TIPODATOID SCH_LAB.TEST.TIPODATOID%TYPE,  
  
    RANDETID SCH_LAB.ORDENDETALLE.RANDETID%TYPE,  
  
    RANDETVALINF SCH_LAB.RANGODETALLE.RANDETVALINF%TYPE,  
  
    RANDETVALSUP SCH_LAB.RANGODETALLE.RANDETVALINF%TYPE,  
  
    PACREPORT SCH_LAB.ORDENDETALLE.PACREPORT%TYPE,  
  
    PACPRINT SCH_LAB.ORDENDETALLE.PACPRINT%TYPE,  
  
    ORDDDETRESULTEST  
SCH_LAB.ORDENDETALLE.ORDDDETRESULTEST%TYPE);
```

type VL_ORDENESDETALLE is table of
SCH_LAB.LAB_K_ORDENES.ORDENES_record_type index by binary_integer;

```
PROCEDURE P_CONSULTAR_ORDENES_DETALLE  
  
(  
  
    P_EMPRESAID IN NUMBER,  
  
    P_ORDID  IN NUMBER,  
  
    P_ESTADO IN VARCHAR2,  
  
    PO_ORDENESDETALLE IN OUT VL_ORDENESDETALLE  
  
);
```

Este procedimiento permite consultar una orden de examen específica con sus respectivas pruebas y rangos referenciales (si el examen requiere rangos referenciales).

PROCEDURE P_ACTUALIZAR_ORDENES_DETALLE

```
(  
  
    P_EMPRESAID IN NUMBER,  
  
    P_ORDID IN NUMBER,  
  
    P_ORDDDETID IN NUMBER,  
  
    P_PACREPORT IN CHAR,  
  
    P_PACPRINT IN CHAR,  
  
    P_ORDDDETRESULTEST VARCHAR2,  
  
    P_ESTADO IN VARCHAR2  
  
);
```

Actualiza el estado si el examen se desea imprimir y actualiza el resultado de la prueba.

end LAB_K_ORDENES;

9.2.1.2.3 Package Lab_K_Rango

create or replace package LAB_K_RANGO is

El paquete LAB_K_RANGO permite hacer todos los procesos sean estos de ingresar, consultar, actualizar y eliminar rangos referenciales por prueba.

```
FUNCTION OBT_DESC_RANGO (PN_EMPRESAID  
RANGODETALLE.EMPRESAID%TYPE, PN_RANDETID  
RANGODETALLE.RANDETID%TYPE) RETURN VARCHAR2;
```

type RANGODETALLE_RECORD_TYPE is RECORD(

 randetid SCH_LAB.RANGODETALLE.randetid%TYPE,

 empresaid SCH_LAB.RANGODETALLE.empresaid%TYPE,

 rangoid SCH_LAB.RANGODETALLE.rangoid%TYPE,

 generoid SCH_LAB.RANGODETALLE.generoid%TYPE,

 etapahumid SCH_LAB.RANGODETALLE.etapahumid%TYPE,

 randetestado SCH_LAB.RANGODETALLE.randetestado%TYPE,

 randetfecha SCH_LAB.RANGODETALLE.randetfecha%TYPE,

 randetvalinf SCH_LAB.RANGODETALLE.randetvalinf%TYPE,

 randetvalsup SCH_LAB.RANGODETALLE.randetvalsup%TYPE,

 randetmultiple SCH_LAB.RANGODETALLE.randetmultiple%TYPE,

 randetesdefault SCH_LAB.RANGODETALLE.randetesdefault%TYPE,

 randetesprint SCH_LAB.RANGODETALLE.randetesprint%TYPE);

TYPE VL_RANGODETALLE is table of

SCH_LAB.LAB_K_RANGO.RANGODETALLE_record_type

index by binary_integer;

PROCEDURE P_CONSULTAR_RANGO_DETALLE(P_EMPRESAID IN
NUMBER,

 P_TESTID IN NUMBER,

P_ESTADO IN VARCHAR2,

PO_RANGODETALLE IN OUT VL_RANGODETALLE);

Permite consultar los rangos referenciales de una prueba específica, esta consulta devuelve e genero, la etapa humana, los rangos de valor inferior, superior y múltiple, si se desea imprimir los rangos en la hoja de resultado.

PROCEDURE P_GRABAR_RANGO_DETALLE(

P_TESTID NUMBER,

P_randetid SCH_LAB.RANGODETALLE.randetid%TYPE,

P_empresaid SCH_LAB.RANGODETALLE.empresaid%TYPE,

P_rangoid SCH_LAB.RANGODETALLE.rangoid%TYPE,

P_generoid SCH_LAB.RANGODETALLE.generoid%TYPE,

P_etapahumid SCH_LAB.RANGODETALLE.etapahumid%TYPE,

P_randetestado SCH_LAB.RANGODETALLE.randetestado%TYPE,

P_randetfecha SCH_LAB.RANGODETALLE.randetfecha%TYPE,

P_randetvalinf SCH_LAB.RANGODETALLE.randetvalinf%TYPE,

P_randetvalsup SCH_LAB.RANGODETALLE.randetvalsup%TYPE,

P_randetmultiple SCH_LAB.RANGODETALLE.randetmultiple%TYPE,

P_randetesdefault SCH_LAB.RANGODETALLE.randetesdefault%TYPE,

P_randetesprint SCH_LAB.RANGODETALLE.randetesprint%TYPE,

PO_MSGERROR IN OUT VARCHAR2);

Este procedimiento inserta, actualiza y elimina los rangos referenciales. Inserta cuando detecta el código del rango sea igual a cero, actualiza cuando encuentra un cambio en los valores del rango referencial de la prueba y hace una eliminación lógica cuando el estado del rango esta inactiva.

End LAB_K_RANGO;

9.2.1.2.4 Package Lab_K_General

CREATE OR REPLACE PACKAGE LAB_K_GENERAL IS

El paquete LAB_K_GENERAL es de uso general, en este paquete se crearon las funciones comunes del sistema.

FUNCTION OBT_NOMBRE_PACIENTE (PN_IDPACIENTE
PACIENTE.IDPACIENTE%TYPE) RETURN VARCHAR2;

La función OBT_NOMBRE_PACIENTE obtiene el nombre del paciente, le enviamos como parámetro el código del paciente.

FUNCTION OBT_NOMBRE_CLIENTE (PN_IDCLIENTE
CLIENTE.CLIENTEID%TYPE) RETURN VARCHAR2;

La función OBT_NOMBRE_CLIENTE obtiene el nombre del cliente, le enviamos como parámetro el código del cliente.

FUNCTION OBT_NOMBRE_VENDEDOR (PN_IDVENDEDOR
VENDEDOR.VENDEDORID%TYPE) RETURN VARCHAR2;

La función OBT_NOMBRE_VENDEDOR obtiene el nombre del vendedor, le enviamos como parámetro el código del vendedor.

```
FUNCTION OBT_NOMBRE_MEDICO (PN_EMPRESAID  
MEDICO.EMPRESAID%TYPE, PN_MEDICOID MEDICO.MEDICOID%TYPE)  
RETURN VARCHAR2;
```

La función OBT_NOMBRE_MEDICO obtiene el nombre del médico, le enviamos como parámetro el código del médico.

```
FUNCTION OBT_NOMBRE_USUARIO (PN_EMPRESAID  
USUARIO.EMPRESAID%TYPE, PN_USUARIOID USUARIO.USUID%TYPE)  
RETURN VARCHAR2;
```

La función OBT_NOMBRE_USUARIO obtiene el nombre del usuario, le enviamos como parámetro el código del usuario registrado del sistema.

```
FUNCTION OBT_NOMBRE_USULOGIN (PN_USUARIOID  
USUARIO.USUID%TYPE) RETURN VARCHAR2;
```

La función OBT_NOMBRE_USULOGIN obtiene el nombre del usuario logeado, le enviamos como parámetro el código del usuario logeado del sistema.

```
FUNCTION OBT_NOMBRE_FORMAPAGO (PN_EMPRESAID  
FORMA_PAGO.EMPRESAID%TYPE, PN_FORMAPAGOID  
FORMA_PAGO.FORMA_PAGOID%TYPE) RETURN VARCHAR2;
```

La función OBT_NOMBRE_FORMAPAGO obtiene la descripción de la forma de pago, le enviamos como parámetro el código de la empresa y el código de la forma de pago.

END;

create or replace package LAB_K_TEST is

```
FUNCTION OBT_DESC_TEST (PN_EMPRESAID TEST.EMPRESAID%TYPE,  
PN_TESTID TEST.TESTID%TYPE) RETURN VARCHAR2;
```

Esta función obtiene la descripción de la prueba, se envía 2 parámetros la empresa y el código de la prueba.

end LAB_K_TEST;

```
CREATE OR REPLACE FUNCTION f_valida_mail(l_user_name IN VARCHAR2)
```

```
RETURN NUMBER IS
```

```
bandera NUMBER;
```

```
l_dot_pos NUMBER;
```

```
l_at_pos NUMBER;
```

```
l_str_length NUMBER;
```

```
BEGIN
```

```
l_dot_pos := instr(l_user_name, '.');
```

```
l_at_pos := instr(l_user_name, '@');
```

```
l_str_length := length(l_user_name);
```

```
IF ((l_dot_pos = 0) OR (l_at_pos = 0) OR (l_dot_pos = l_at_pos + 1) OR
```

```
(l_at_pos = 1) OR (l_at_pos = l_str_length) OR
```

```
(l_dot_pos = l_str_length))
```

```

THEN

bandera :=2;

RETURN bandera;

END IF;

IF instr(substr(l_user_name

,l_at_pos)

,',') = 0

THEN

bandera:=2;

-- RETURN 'FAILURE';

END IF;

bandera:=1;

RETURN bandera;

--RETURN 'SUCCESS';

END f_valida_mail;

```

Esta función valida si una cuenta de correo electrónico es válida.

create or replace function LAB_F_ETAPA_HUMANA(P_EMPRESAID IN NUMBER,
P_ETAPAHUMANAIID IN NUMBER) return varchar2 is

Result varchar2(200);

begin

SELECT E.ETAPAHUMDESCRIPCION INTO RESULT

FROM ETAPAHUMANA E

WHERE E.ETAPAHUMID = P_ETAPAHUMANAIID AND E.EMPRESAID =
P_EMPRESAID;

return(Result);

end LAB_F_ETAPA_HUMANA;

Esta función devuelve la descripción de la etapa humana.

9.2.1.3 Funciones

9.2.1.3.1 Function Validarcedula

```
CREATE OR REPLACE FUNCTION VALIDARCEDULA (V_CEDULA IN  
VARCHAR2)
```

```
RETURN BOOLEAN
```

```
IS
```

```
RESULT BOOLEAN;
```

```
CONT NUMBER;
```

```
SUMA NUMBER := 0;
```

```
DECENA NUMBER := 10;
```

```
VALOR NUMBER;
```

```
BEGIN
```

*--PARA CUANDO EL CLIENTE NO CONOCE LA CEDULA O NO LA TIENE
DISPONIBLE SE DA ESTA OPCION*

IF V_CEDULA = '9999999999' THEN

RETURN TRUE;

END IF;

*--SI ES QUE ESTO DA ERROR ES PORQUE HAY LETRAS Y PUEDE SER UN
PASAPORTE*

*--POR ENDE SE PUEDE CAER BAJO ERROR ORA-01722 EL CUAL CONTROLA
LA EXCEPCION Y LA FUNCION RETORNA TRUE*

VALOR := TO_NUMBER(V_CEDULA);

IF LENGTH(V_CEDULA) <> 10 THEN

RETURN FALSE;

END IF;

FOR CONT IN 1..9 LOOP

VALOR := TO_NUMBER(Substr(V_CEDULA, CONT, 1));

--SI EL DIGITO ESTA EN LA POSICION IMPAR

IF Mod(CONT, 2) = 1 THEN

VALOR := VALOR * 2;

```

    IF VALOR > 9 THEN

        VALOR := VALOR - 9;

    END IF;

END IF;

SUMA := SUMA + VALOR;

END LOOP;

WHILE DECENA < SUMA LOOP

    DECENA := DECENA + 10;

END LOOP;

VALOR := TO_NUMBER(SUBSTR(V_CEDULA, 10, 1));

IF DECENA - SUMA = VALOR THEN

    RESULT := TRUE;

    dbms_output.put_line('LA CEDULA INGRESADA ES VALIDA...');

ELSE

    RESULT := FALSE;

    dbms_output.put_line('LA CEDULA INGRESADA ES INVALIDA...');

END IF;

RETURN RESULT;

```



```

EXCEPTION

WHEN OTHERS THEN

    IF SQLCODE = -6502 THEN

        RETURN TRUE;

    ELSE

        RETURN FALSE;

    END IF;

END VALIDARCEDULA;

```

Esta función valida si una cedula esta correctamente ingresada y coincide el digito verificador.

9.2.1.3.2 Function Val_Ruc

create or replace function val_ruc(aicruc IN VARCHAR2) RETURN VARCHAR2 IS

Type vacoeverruc is VARRAY(10) of number(3);

--coeficientes usados para verificar el decimo digito del RUC tipo juridico

vacoeverrucjur vacoeverruc:=vacoeverruc(4,3,2,7,6,5,4,3,2);

--coeficientes usados para verificar el noveno digito del RUC tipo publico

vacoeverrucpub vacoeverruc:=vacoeverruc(3,2,7,6,5,4,3,2);

--coeficientes usados para verificar el decimo digito del RUC tipo natural

```
vacoeverrucnat vacoeverruc:=vacoeverruc(2,1,2,1,2,1,2,1,2);
```

```
vnresultado number:=0;
```

```
vntotal number:=0;
```

```
vnconpos number:=1;
```

```
vndigver number:=0;
```

```
vnresiduo number:=0;
```

```
vcdigdos varchar2(2) := substr(aicruc,1,2);
```

```
vcdigtre varchar2(3):= substr(aicruc,11,3);
```

```
vcdigcua varchar2(4):= substr(aicruc,10,4);
```

```
vcterdig varchar2(1):=substr(aicruc,3,1);
```

```
vcdigverjur varchar2(1):= substr(aicruc,10,1);
```

```
vcdigverpub varchar2(1):= substr(aicruc,9,1);
```

```
BEGIN
```

```
--se verifica la longitud del ruc
```

```
if length(aicruc) <> 13 then
```

```
    return ('N');
```

```
--se verifica que los dos primeros digitos esten entre 1 y 22
```

```
elsif to_number(vcdigdos) < 1 or to_number(vcdigdos) > 22 then
```

```

return ('N');

--Verificacion para el ruc juridico.. cuando el tercer digito de la cedula es 9

elsif vcterdig = '9' then

    --Los tres ultimos digitos no pueden ser '000' porque por lo menos hay la sucursal
    principal

    if vcdigtre = '000' then

        return ('N');

    else

        -- Se hace el control del digito verificador con el modulo 11

        for i in 1..vacoeverrucjur.count loop

            vntotal := vntotal + to_number(substr(aicruc,i,1)) * vacoeverrucjur(i);

        end loop;

        vnresiduo := mod(vntotal,11);

        if vnresiduo = 0 then -- Si el residuo es 0 el digito verificador es el 0

            vndigver := 0;

            elsif vnresiduo = 1 then -- Si el residuo es 1 ruc es incorrecto

                return ('N');

            else

```

```

    vndigver := 11 - vnresiduo; --Caso contrario, el digito verificador se obtiene

end if;                --restando el residuo de 11

    if (vndigver = vcdigverjur) then --Si el digito de la posicion 10 del RUC es igual al
digito

        return ('S');                --verificador obtenido el RUC es correcto.

    else

        return ('N');

    end if;

end if;

--Verificacion para el RUC publico... cuando el tercer digito es 6

elsif vcterdig = '6' then

    --Los cuatro ultimos digitos no pueden ser '0000' porque por lo menos hay la sucursal
principal

    if vcdigcua = '0000' then

        return ('N');

    else

        -- Se hace el control del digito verificador con los coeficientes respectivos y el
modulo 11

```

```

for i in 1..vacoeverrucpub.count loop

    vntotal := vntotal + to_number(substr(aicruc,i,1)) * vacoeverrucpub(i);

end loop;

vnresiduo := mod(vntotal,11);

if vnresiduo = 0 then -- Si el residuo es 0 el digito verificador es el 0

    vndigver := 0;

elsif vnresiduo = 1 then -- Si el residuo es 1 ruc es incorrecto

    return ('N');

else

    vndigver := 11 - vnresiduo; --Caso contrario, el digito verificador se obtiene

                                --restando el residuo de 11

end if;

if (vndigver = vcdigverpub) then --Si el digito de la posicion 9 del RUC es igual al
digito

    return ('S');              --verificador obtenido el RUC es correcto.

else

    return ('N');

```

```

end if;

end if;

--Verificacion para el RUC de personas naturales... cuando el tercer digito es menor
que 6

elsif to_number(vcterdig) < 6 then

--Los tres ultimos digitos no pueden ser iguales a '000'

if vcdigtre = '000' then

return ('N');

else

-- Se hace el control del digito verificador con los coeficientes respectivos y el
modulo 10

for i in 1..vacoeverrucnat.count loop

vnresultado:= to_number(substr(aicruc,i,1)) * vacoeverrucnat(i);

--si el resultado es > que 9, se suman sus cifras

if vnresultado > 9 then

vnresultado := to_number(substr(to_char(vnresultado),1,1)) +

to_number(substr(to_char(vnresultado),2,1));

end if;

vntotal := vntotal + vnresultado;

```

```

end loop;

vnresiduo := mod(vntotal,10);

--Si el residuo es 0, el digito verificador es 0

if vnresiduo = 0 then

    vndigver := 0;

else

    vndigver := 10 - vnresiduo; --Caso contrario, el digito verificador se obtiene

                                --restando el residuo de 10

end if;

if (vndigver = vcdigverjur) then --Si el digito de la posicion 10 del RUC es igual al
digito

    return ('S');                --verificador obtenido el RUC es correcto.

else

    return ('N');

end if;

end if;

end if;

end;

```

Esta función valida si una Ruc esta correctamente ingresada y coincide el digito verificador.

9.2.2 Documentación para el Usuario

9.2.2.1 Introducción

Otorgar soporte a los usuarios del sistema, de forma oportuna para que ingrese información de manera consistente.

9.2.2.2 Requerimientos del sistema

EQUIPO	CARACTERÍSTICAS
Servidor de internet	El Sistema Operativo Windows XP SP2, 1 GB de Memoria RAM, H.D de 160GB, y un procesador mínimo de Dual Core 1.3GHz Hosting.
PC de usuario	El Sistema Operativo Windows XP SP2, Microsoft Office, 1 GB de Memoria RAM, H.D de 160GB, y un procesador mínimo de Celeron 3.3GHz

Se debe contratar un servicio de internet con un ancho de banda mínimo de 256kbps.

9.2.2.3 Opciones del sistema

El presente Manual está organizado de acuerdo a la secuencia de ingreso a las pantallas del sistema de la siguiente manera:

1. Ingreso al sistema
2. Mantenimiento
3. Procesos
4. Reportes
5. Ayuda
6. Salir

9.2.2.3.1 Ingreso al sistema

En esta pantalla el usuario da un clic sobre la imagen, la misma que tiene un vínculo hacia la pantalla de usuario y contraseña.

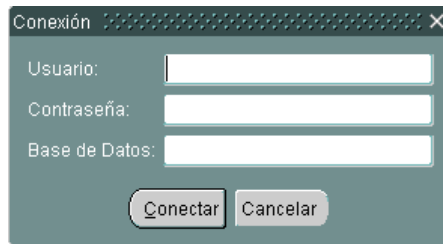


PANTALLA N°1 :PRESENTACION

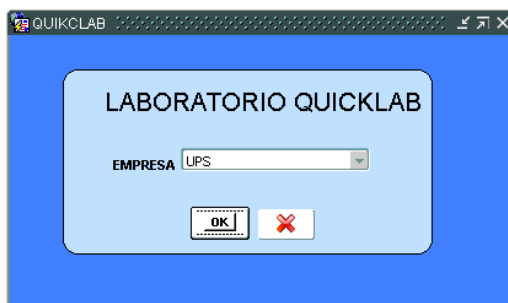
FUENTE: AUTORES

En esta pantalla el usuario debe digitar el Nombre de Usuario y Clave, presionar sobre el botón Conectar tal como se muestra en la figura siguiente, los datos que se ingresan se

los proporciona el administrador del sistema, si usted no está registrado debe de comunicar al administrador del sistema, con el fin de otórgale un usuario y clave para que pueda ingresar al sistema.

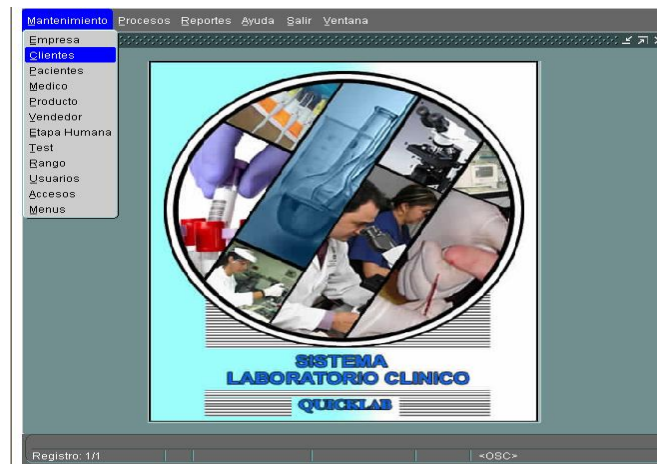


En esta pantalla debe seleccionar la empresa a la que desea ingresar, presionar sobre el botón OK como se muestra en la figura siguiente o presionar sobre el botón X para cancelar su ingreso al sistema.



PANTALLA N°2 : LOGIN
FUENTE: AUTORES

En esta pantalla se presentan todas las opciones del sistema.



PANTALLA N°3 : MENU
FUENTE: AUTORES

9.2.2.3.2 MANTENIMIENTO

9.2.2.3.2.1 VENDEDOR

Al dar clic sobre la opción vendedor se despliega la siguiente pantalla.

Para crear un vendedor nuevo

presionamos sobre el botón nuevo



Para grabar presionamos sobre el botón
grabar



Si deseamos cancelar presionamos sobre
el botón cancelar

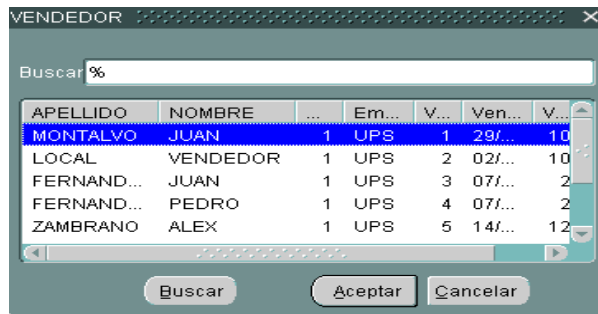


PANTALLA N°4 : MANTENIMIENTO VENDEDOR
FUENTE: AUTORES

Empresa.- Se presenta de forma automática dependiendo de la empresa seleccionada

Código.- El código se genera automáticamente

Botón buscar.-  Presenta una lista con los datos del vendedor, hay que seleccionar un registro y presionar el botón aceptar para presentar todos los datos del vendedor ya grabados en la base de datos, se puede buscar por apellido o se puede cancelar la búsqueda.



Nombre.- Nombre del vendedor

Apellido.- Apellido del vendedor

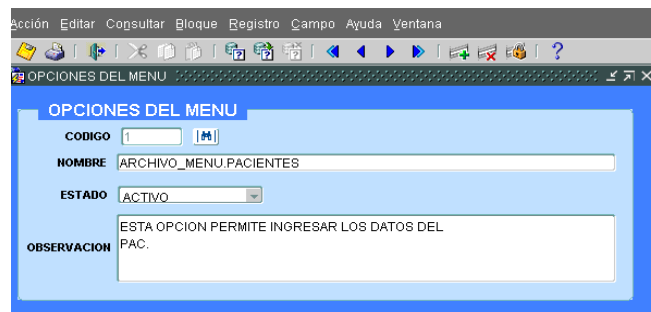
Fecha.- Fecha de ingreso

Comisión.- Es un valor en dólares

Estado.- Activo o Inactivo

9.2.2.3.2.2 Opciones del Menú

Se ingresa todas las opciones del menú para luego ser asignadas a los usuarios.



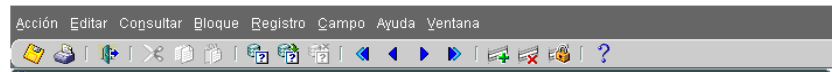
PANTALLA N°5 : MANTENIMIENTO OPCIONES DE MENU
FUENTE: AUTORES

Barra de menú

En la opción consultar, tenemos la opción introducir que nos permite digitar el código del usuario y luego presionamos la opción ejecutar y tenemos el resultado esperado.

Barra de herramientas

En esta barra de herramientas encontrara opciones y herramientas que le permiten manipular la información que se encuentra grabada en la base de datos.



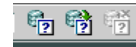
Grabar.- Graba información



Salir.- Salir de la forma



Consultar.- Búsqueda individual de las opciones de menú



Barra de navegación.- Puede visualizar de registro en registro



Nuevo.- Crea una opción de menú

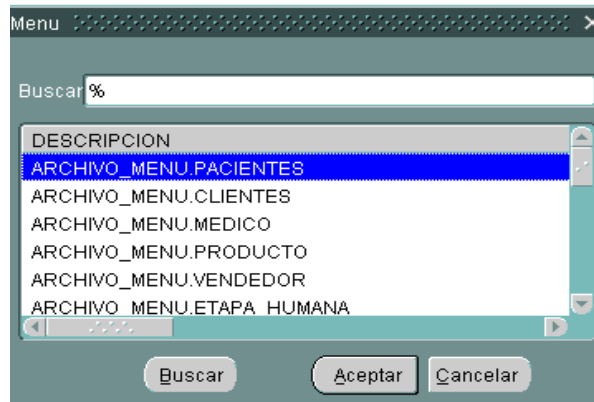


Eliminar.- Desactiva registro.



Código. El código se genera automáticamente

Botón Buscar.-  Presenta una lista con los datos de opciones del menú, hay que seleccionar un registro y presionar el botón aceptar para presentar todos los datos del vendedor ya grabados en la base de datos, se puede buscar por apellido o se puede cancelar la búsqueda.



Nombre.- Nombre de la opción del menú

Estado.- Activo o Inactivo

Observación.- Descripción de la opción de menú.

9.2.2.3.2.3 ACCESOS

En esta pantalla se consulta el usuario, para asignarle las opciones de menú que podrá utilizar.

The screenshot shows a software window titled 'PERMISOS'. It has a menu bar with 'Acción', 'Editar', 'Consultar', 'Bloque', 'Registro', 'Campo', 'Ayuda', and 'Ventana'. Below the menu bar is a toolbar with various icons. The main area is divided into two sections: 'USUARIOS' and 'OPCIONES MENU'.

USUARIOS

Código:
Nombre: Apellido:
Identificación:
Usuario Login:

OPCIONES MENU

Menu	Descripcion	Estado
22	MEDICOS_MENU.RESULTADO_DE_EXAMENES	ACTIVO
12	PROCESOS_MENU.SOLICITUD_DE_EXAMEN	ACTIVO
14	PROCESOS_MENU.INGRESO_DE_RESULTADOS	ACTIVO
5	ARCHIVO_MENU.VENDEDOR	ACTIVO
6	ARCHIVO_MENU.ETAPA_HUMANA	ACTIVO
7	ARCHIVO_MENU.TEST	ACTIVO
8	ARCHIVO_MENU.RANGO	ACTIVO
9	ARCHIVO_MENU.MENU	ACTIVO
10	ARCHIVO_MENU.ACCESOS	ACTIVO
11	ARCHIVO_MENU.MENUS	ACTIVO

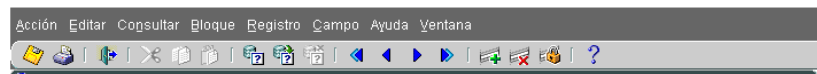
PANTALLA N°6 : MANTENIMIENTO ACCESOS
FUENTE: AUTORES

Barra de menú

En la opción consultar, tenemos la opción introducir que nos permite digitar el código del usuario y luego presionamos la opción ejecutar y tenemos el resultado esperado.

Barra de herramientas

En esta barra de herramientas encontrara opciones y herramientas que le permiten manipular la información que se encuentra grabada en la base de datos.



Grabar.- Graba información



Salir.- Salir de la forma



Consultar.



Introducir consulta.- Se introduce el código



Ejecutar consulta.- Se ejecuta la consulta



Cancelar consulta.- Se cancela la consulta

Barra de navegación.- Puede visualizar de registro en registro



Nuevo.- Crea una opción de menú



Eliminar.- Desactiva registro.



Código.- Código de usuario

Nombre.- Nombre de usuario

Apellido.- Apellido de usuario

Identificación.- Identificación de usuario

Usuario Login.- Usuario para ingresar al sistema

Menú.- Código de la opción de menú

Descripción.- Descripción de la opción del menú

Estado.- Activo o Inactivo

9.2.2.3.2.4 EMPRESA

Al dar clic sobre la opción **Empresa** se despliega la siguiente pantalla.

Para crear un vendedor nuevo

presionamos sobre el botón **nuevo**



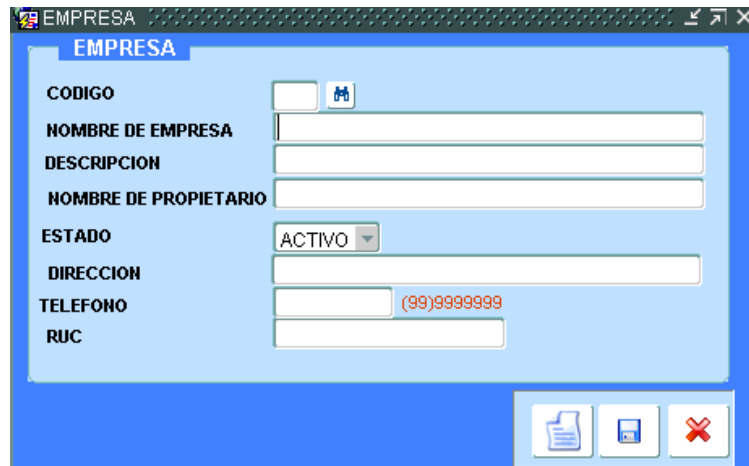
Para grabar presionamos sobre el botón

grabar



Si deseamos cancelar presionamos sobre

el botón **cancelar**

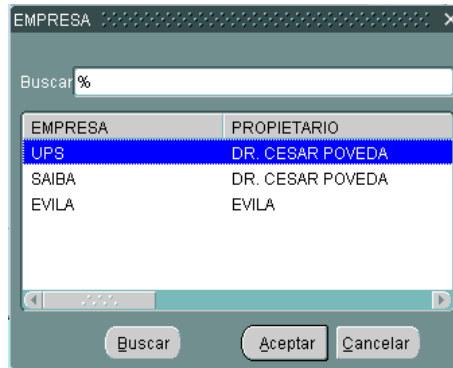


PANTALLA N°7 : MANTENIMIENTO EMPRESA

FUENTE: AUTORES

Código.- El código se genera automáticamente al momento de Grabar la transacción

Botón buscar.-  Presenta una lista con los datos de las empresas registradas en el sistema, hay que seleccionar un registro y presionar el botón aceptar o dar doble clic, para presentar todos los datos de la empresa ya grabados en la base de datos.



Nombre de Empresa.- Nombre de la Empresa

Descripción.- Razón social de la empresa

Nombre del Propietario.- Nombre del propietario de la empresa

Estado.- Estado activo o inactivo

Dirección.- Almacena la dirección de la empresa

Teléfono.- Teléfono de la empresa.

Ruc.- Ruc de la empresa.

9.2.2.3.2.5 CLIENTE

Al dar clic sobre la opción Cliente se despliega la siguiente pantalla.

Para crear un cliente nuevo presionamos sobre el botón **nuevo**



Para grabar presionamos sobre el botón **grabar**



Si deseamos cancelar presionamos sobre el botón **cancelar**

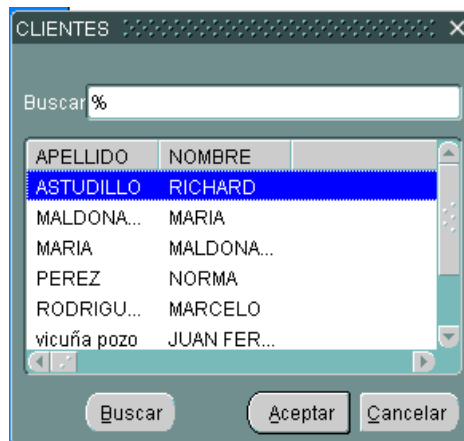


PANTALLA N°8 : MANTENIMIENTO CLIENTE
FUENTE: AUTORES

Empresa.- Se presenta de forma automática dependiendo de la empresa seleccionada

Código.- El código se genera automáticamente

Botón buscar.-  Presenta una lista con los datos del Cliente, hay que seleccionar un registro y presionar el botón aceptar para presentar todos los datos del cliente ya grabados en la base de datos, se puede buscar por apellido o se puede cancelar la búsqueda.



Nombre.- Nombre del Cliente

Apellido.- Apellido del Cliente

Teléfono.- Teléfono del cliente

Dirección.- Dirección del Cliente

Identificación.- se selecciona el tipo de identificación del cliente puede ser cedula, pasaporte, N/A.

Estado.- Activo o inactivo

9.2.2.3.2.6 PACIENTE

Al dar clic sobre la opción Paciente se despliega la siguiente pantalla.

Para crear un Paciente nuevo

presionamos sobre el botón **nuevo**



Para grabar presionamos sobre el botón

grabar



Si deseamos cancelar presionamos sobre

el botón **cancelar**

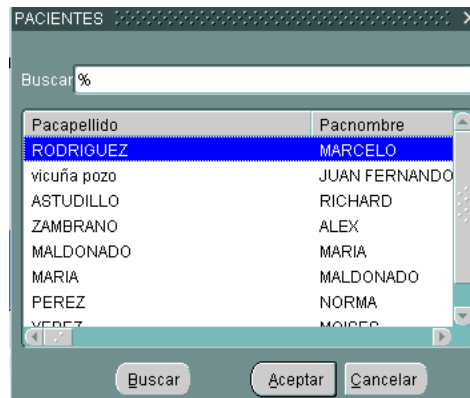


PANTALLA N°9 : MANTENIMIENTO PACIENTE

FUENTE: AUTORES

Código.- El código se genera automáticamente

Botón buscar.-  Presenta una lista con los datos de los Pacientes que están registrados en la base de datos, hay que seleccionar un registro y presionar el botón aceptar para presentar todos los datos del paciente ya grabados en la base de datos, se puede buscar por apellido o se puede cancelar la búsqueda.



Nombre.- Nombre del Paciente

Apellido.- Apellido del paciente

Teléfono.- Teléfono del cliente

Fecha de nacimiento: almacena la fecha de nacimiento del paciente

Dirección.- Dirección del Cliente

Identificación.- Se selecciona el tipo de identificación del cliente puede ser cedula, pasaporte, N/A.

Estado.- Activo o inactivo

Género.- Se escoge el género del paciente

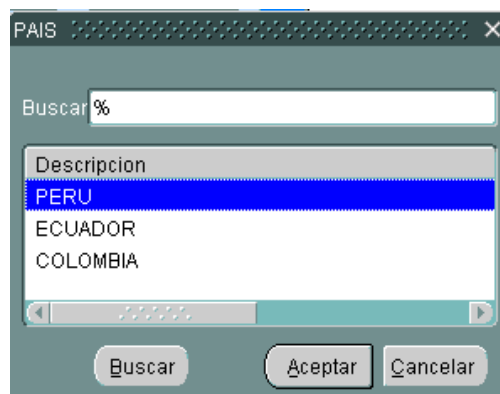
Fono1.- Almacena el número de teléfono del paciente

Móvil.- Almacena el celular del paciente

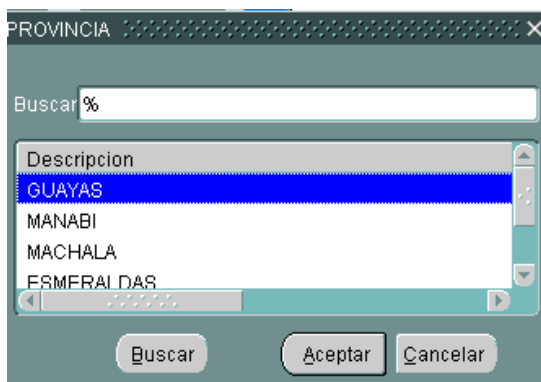
Mail.- Almacena el mail del paciente

País .- Tiene la opción de escoger el país de donde es nacido el Paciente,

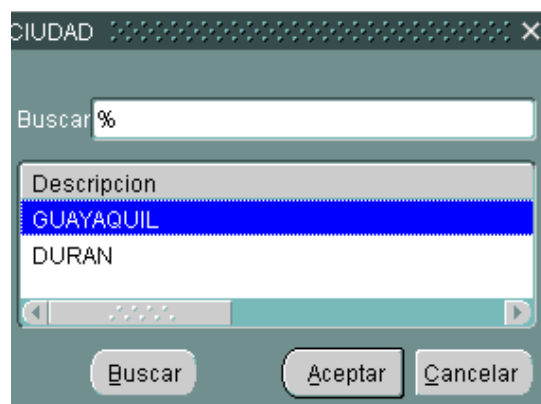
mediante el botón 



Provincia.- Una vez escogido el país automáticamente en la lista de valores aparecen las provincias de dicho país



Ciudad.- Una vez escogida la provincia automáticamente en la lista de valores aparecen las ciudades de dicha provincia, dando clic en el botón 



9.2.2.3.2.7 MEDICO

Al dar clic sobre la opción Medico se despliega la siguiente pantalla.

Para crear un Medico nuevo presionamos sobre el botón **nuevo**



Para grabar presionamos sobre el botón **grabar**



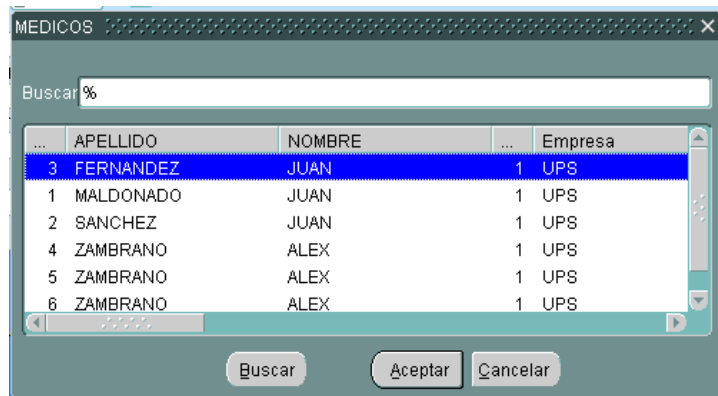
Si deseamos cancelar presionamos sobre el botón **cancelar**



PANTALLA N°10 : MANTENIMIENTO MEDICO
FUENTE: AUTORES

Código.- El código se genera automáticamente

Botón buscar.-  Presenta una lista con los datos de los Médicos que están registrados en la base de datos, hay que seleccionar un registro y presionar el botón aceptar para presentar todos los datos del paciente ya grabados en la base de datos, se puede buscar por apellido o se puede cancelar la búsqueda.



...	APELLIDO	NOMBRE	...	Empresa
3	FERNANDEZ	JUAN	1	UPS
1	MALDONADO	JUAN	1	UPS
2	SANCHEZ	JUAN	1	UPS
4	ZAMBRANO	ALEX	1	UPS
5	ZAMBRANO	ALEX	1	UPS
6	ZAMBRANO	ALEX	1	UPS

Nombre.- Nombre del Medico

Apellido.- Apellido del Medico

Teléfono.- Teléfono del Medico

Fecha de ingreso: Almacena la fecha de ingreso del médico, al sistema

Dirección.- Dirección del Medico

Identificación.- Se selecciona el tipo de identificación del cliente puede ser cedula, pasaporte, N/A.

Estado.- Activo o inactivo

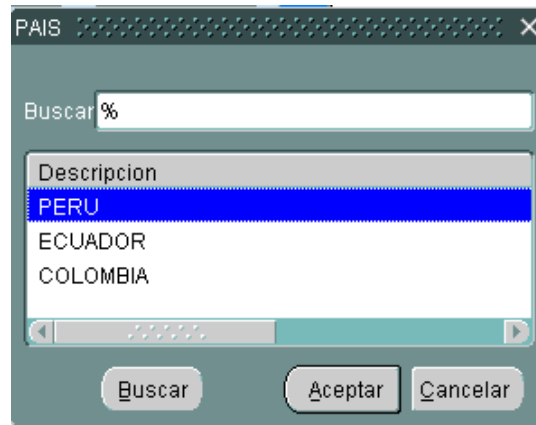
Comisión.- Se ingresa la comisión que va a ganar el medico

Fono1.- Almacena el número de teléfono del Medico

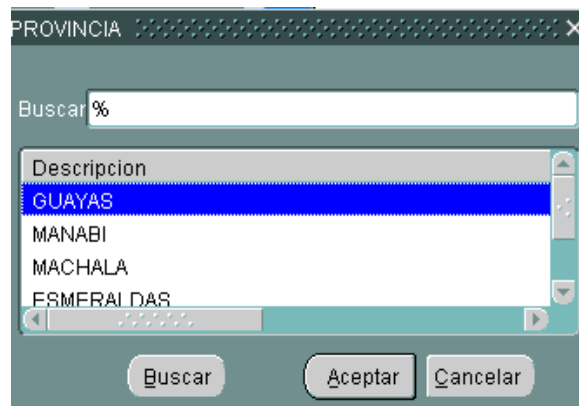
Móvil.- Almacena el celular del Medico

Mail.- Almacena el mail del Medico

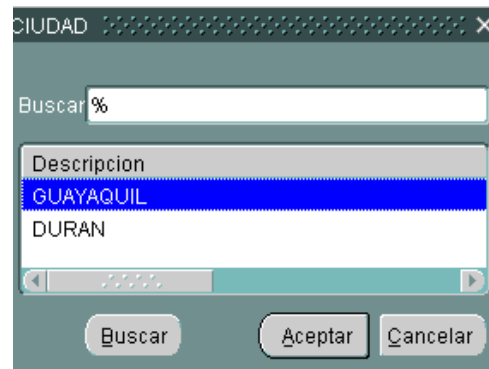
País.-  Tiene la opción de escoger el país de donde es nacido el Médico, mediante el botón



Provincia.- Una vez escogido el país automáticamente en la lista de valores aparecen las provincias de dicho país



Ciudad.-  Una vez escogida la provincia automáticamente en la lista de valores aparecen las ciudades de dicha provincia, dando clic en el botón



9.2.2.3.2.8 ETAPA HUMANA

Al dar clic sobre la opción Medico se despliega la siguiente pantalla.

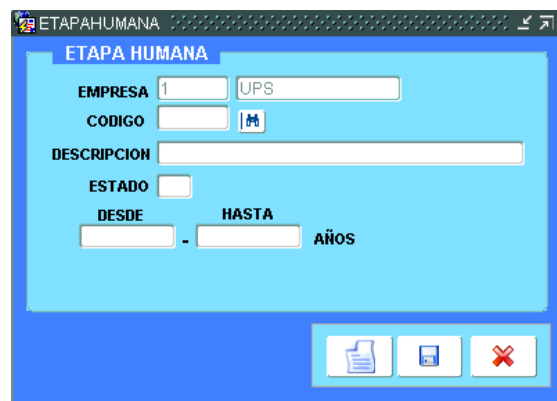
Para crear un Medico nuevo presionamos sobre el botón **nuevo**



Para grabar presionamos sobre el botón **grabar**



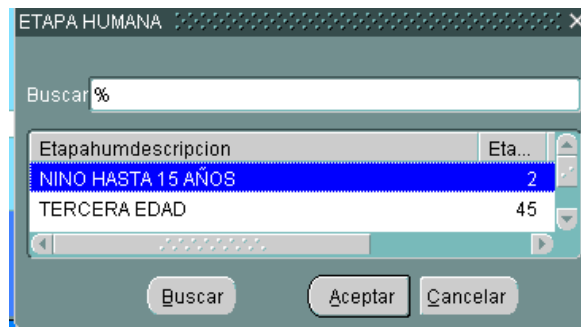
Si deseamos cancelar presionamos sobre el botón **cancelar**



PANTALLA N°11 : MANTENIMIENTO ETAPA HUMANA
FUENTE: AUTORES

Código.- El código se genera automáticamente

Botón buscar.-  Presenta una lista con los datos de las Etapas humanas que están registrados en la base de datos, hay que seleccionar un registro y presionar el botón aceptar para presentar todos los datos de la Etapa humana ya grabados en la base de datos.



Descripción .- descripción de la etapa humana

Estado: estado de la etapa humana

desde.- edad inicial

hasta: edad límite hasta donde es permitida la etapa humana

9.2.2.3.2.9 VENDEDOR

Al dar clic sobre la opción vendedor se despliega la siguiente pantalla.

Para crear un vendedor nuevo

presionamos sobre el botón nuevo



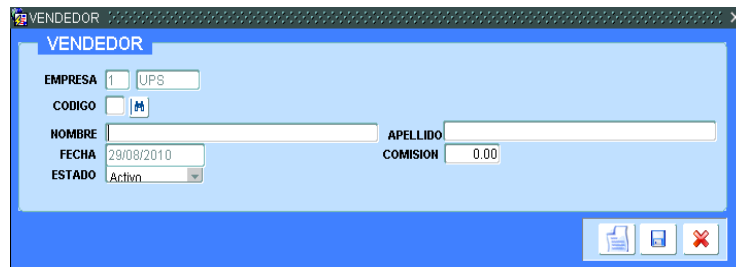
Para grabar presionamos sobre el botón

grabar



Si deseamos cancelar presionamos sobre

el botón cancelar



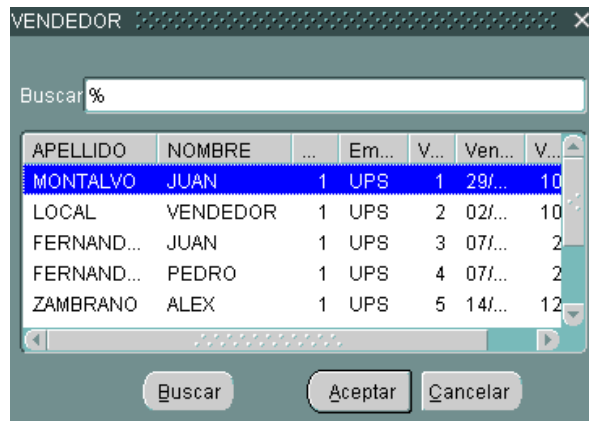
PANTALLA N°12 : MANTENIMIENTO VENDEDOR

FUENTE: AUTORES

Empresa.- Se presenta de forma automática dependiendo de la empresa seleccionada

Código.- El código se genera automáticamente

Botón buscar.-  Presenta una lista con los datos del vendedor, hay que seleccionar un registro y presionar el botón aceptar para presentar todos los datos del vendedor ya grabados en la base de datos, se puede buscar por apellido o se puede cancelar la búsqueda.



APELLIDO	NOMBRE	...	Em...	V...	Ven...	V...
MONTALVO	JUAN	1	UPS	1	29/...	10
LOCAL	VENDEDOR	1	UPS	2	02/...	10
FERNAND...	JUAN	1	UPS	3	07/...	2
FERNAND...	PEDRO	1	UPS	4	07/...	2
ZAMBRANO	ALEX	1	UPS	5	14/...	12

Nombre.- Nombre del vendedor

Apellido.- Apellido del vendedor

Fecha.- Fecha de ingreso

Comisión.- Es un valor en dólares

Estado.- Activo o Inactivo

9.2.2.3.2.10 RANGOS

En esta pantalla se crean los rangos de los test.

Generoid	Etapahumid	Randetvalinf	Randetvalsup	Randetmultiple	Randetesdefault	Randetesprint
☒ MASCULINO					☒	☒
☐ MASCULINO					☐	☐
☐ MASCULINO					☐	☐
☐ MASCULINO					☐	☐
☐ MASCULINO					☐	☐
☐ MASCULINO					☐	☐
☐ MASCULINO					☐	☐
☐ MASCULINO					☐	☐

PANTALLA N°13 : MANTENIMIENTO RANGOS
FUENTE: AUTORES

Ingresar / Modificar Nuevos Rangos

Test. – Consultas un test para realizar el ingreso de sus rangos respectivo

Genero.- Si es para masculino marcamos el casillero(le dejamos con el visto), si es de género femenino desmarcamos el casillero de selección (le quitamos el visto).

Etapas Humana.- consultamos la etapa humana

Rango Valor Inferior.- Es el valor mínimo de la prueba siempre y cuando la prueba acepte solo resultados numéricos (eso se configura en la pantalla de test.)

Rango valor Superior.- Es el valor máximo de la prueba siempre y cuando la prueba acepte solo resultados numéricos (eso se configura en la pantalla de test.)

Rango Múltiple.- Este rango es cuando el test está configurado para que los resultados solo sean de texto, el rango múltiple sirve más como una observación de alguna prueba que puede presentar varias referencias.

Rango se Imprime.- esta opción me permite parametrizar si se desea que salga ese rango al momento de imprimir.

Botón Grabar.- Clic en el botón grabar para realizar la transacción.

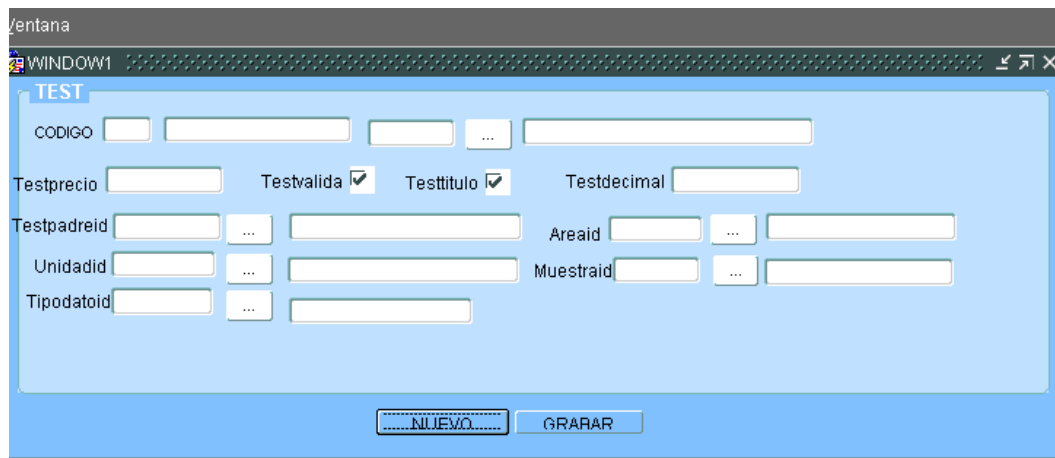
Eliminar Rango

Test.- Consultamos un test para realizar la eliminación de los rangos.

Botón Eliminar.- Clic en el botón eliminar y todos los rangos de esa prueba se eliminarán.

9.2.2.3.2.11 TEST

En la pantalla test se realizan las definiciones de los exámenes que se realizará el paciente.



PANTALLA N°14: MANTENIMIENTO TEST
FUENTE: AUTORES

Nuevo / modificar Examen

Test.- Ingresar el nombre de la nueva prueba.

Test Precio.- En este campo se ingresa el precio del examen siempre y cuando no sea Titulo y que el padre del mismo de tenga precio.

Test Valida.- Este casillero me indica si el test valida al ingreso de resultado.

Test Titulo.- Se indica si el examen es de ingreso de resultado o solo de etiqueta.

Test Padre.- se selecciona alguna prueba que sea titulo ósea solo de etiqueta... para poder jerarquizar los exámenes.

Unidad.- es la unidad de medida que va trabajar el examen.

Muestra.- es una descripción que tipo de muestra (sangre, suero, orina, heces, etc.)

Tipo Dato.- si el examen es numérico o solo texto.

Botón Grabar.- Graba l examen o la actualización del mismo.

9.2.2.3.2.12 USUARIO

Se ingresan los datos del usuario del sistema.

Acción Editar Consultar Bloque Registro Campo Ayuda Ventana

WINDOW1

MANTENIMIENTO DE USUARIO

EMPRESA CODIGO ADMINISTRADOR ☒

APELLIDO NOMBRE

TIPO IDENT. IDENTIFICACION

LOGIN CLAVE FECHA NAC.

PAIS PROVINCIA

CIUDAD CARGO

DIRECCION

TELEFONO MOVIL EMAIL

PANTALLA N°15 : MANTENIMIENTO USUARIO
FUENTE: AUTORES

Empresa.- Nombre de la empresa

Código.- Código del usuario

Administrador.- Si es Administrador

Apellido.- Apellido del Usuario

Nombre.- Nombre del Usuario

Tipo de Identificación.- Cedula o pasaporte

Identificación.- Número de cedula o pasaporte

Login.- Usuario

Clave.- Clave del usuario

Fecha Nac.- Fecha de Nacimiento

País.- País del usuario

Provincia.- Provincia del usuario

Ciudad.- Ciudad del usuario

Dirección.- Dirección del usuario

Teléfono.- Teléfono del usuario

Móvil.- Teléfono móvil del usuario

Email.- Correo electrónico del usuario

Botón Nuevo.- Crea un usuario nuevo

Botón Grabar.- Graba un usuario

Botón eliminar.- Elimina un usuario

Botón Cancelar.- Cancela los procesos.

9.2.2.3.3 PROCESOS

9.2.2.3.3.1 SOLICITUD DE EXAMEN

En esta pantalla se ingresa la solicitud de examen del paciente, en la cual se ingresa el médico, la fecha de entrega, el turno puede ser automático y se lo controla por día, en la observación algún detalle importante y se ingresa el vendedor, en el bloque de test se presenta automáticamente los test que realiza el laboratorio, se selecciona cada test haciendo doble clic y automáticamente se presenta el bloque de exámenes a realizar en la cual se ingresa la cantidad y el descuento en dólares el mismo que no puede ser mayor al precio del producto. Presentamos el total del descuento, el total del IVA y el total, una vez ingresado toda esta información se procede a grabar la solicitud para luego ser consultada desde la opción de facturación.

Acción Editar Consultar Bloque Registro Campo Ayuda Ventana

SOLICITUD

Nº EMPRESA 1 UPS FECHA ORDEN 30/08/2010

PACIENTE

PACIENTE 1 RICHARD ASTUDILLO EDAD 41

GENERO MASCULINO ETAPA HUMANA ADULTO FECHA NACIMIENTO 21/05/1969

FICHA MEDICA PARTICULAR MEDICO 1 JUAN MALDONADO

TOMA MUESTRA LOCAL FECHA ENTREGA 09/09/2010 dd/mm/yyyy

MOTIVO TURNO 1 URGENTE ☒

LABORATORIO OBSERVACION

DIAGNOSTICO VENDEDOR 1 JUAN MONTALVO

TEST	NOMBRE	PRECIO
HEMOG		5.00
GLUCO		3.00
COL		2.34
TRIG		1.78
URO		6.89
RWF		10.00

EXAMENES A REALIZAR				
PRODUCTO	CANT	PRECIO	DESC	SUBTOTAL
1 HEMOGRAMA	1	5.00	1	4.48
2 GLUCOSA	1	3.00	1	2.24
3 COLESTEROL	1	2.34	1	1.50

DESCUENTO: 3.00

IVA: 0.88

TOTAL: 8.22

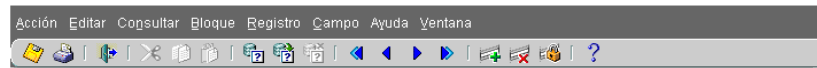
PANTALLA N°16 : SOLICITUD DE EXAMEN
FUENTE: AUTORES

Barra de menú

En la opción consultar, tenemos la opción introducir que nos permite digitar el código del usuario y luego presionamos la opción ejecutar y tenemos el resultado esperado.

Barra de herramientas

En esta barra de herramientas encontrara opciones y herramientas que le permiten manipular la información que se encuentra grabada en la base de datos.



Grabar.- Graba información



Salir.- Salir de la forma



Consultar.

Introducir consulta.- Se introduce el código



Ejecutar consulta.- Se ejecuta la consulta



Cancelar consulta.- Se cancela la consulta



Barra de navegación.- Puede visualizar de registro en registro



Nuevo.- Crea una opción de menú



Eliminar.- Desactiva registro.



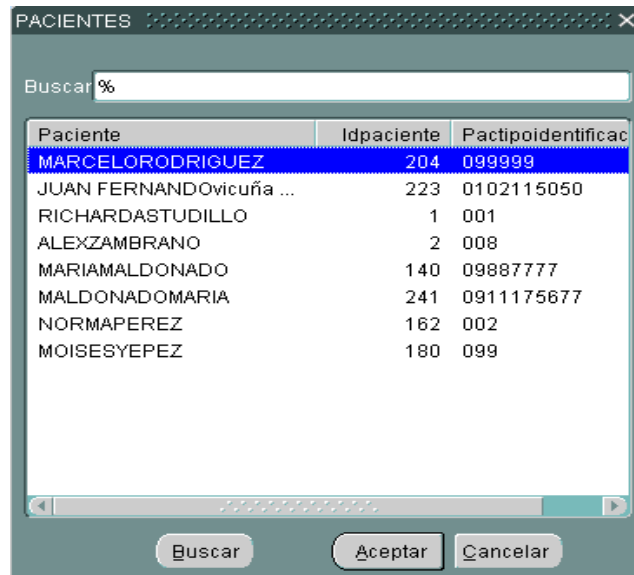
No.- Se genera automáticamente el número de la solicitud

Empresa.- Presenta el código y nombre de la empresa automáticamente

Fecha.- Fecha actual se presenta automáticamente

Bloque Paciente

Paciente.- Se ingresa el código del paciente o se consulta en el botón  el mismo que nos presenta una lista de pacientes.



Botón crear paciente.-  En el caso que el paciente no exista, este botón activa el formulario de paciente.

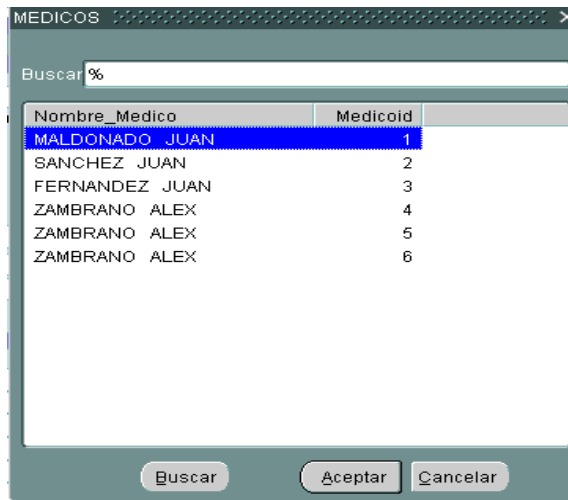
Género.- Masculino o Femenino

Etapas humana.- Clasificación de los pacientes de acuerdo a su edad

Edad.- Calcula automáticamente la edad dependiendo de la fecha de nacimiento.

Fecha de Nacimiento.- La fecha de nacimiento no puede ser al año actual.

Médico.- Se ingresa el código del médico o se consulta en una lista los médicos existentes.



Botón crear médico.-  En el caso que el médico no exista, este botón activa el formulario de medico.

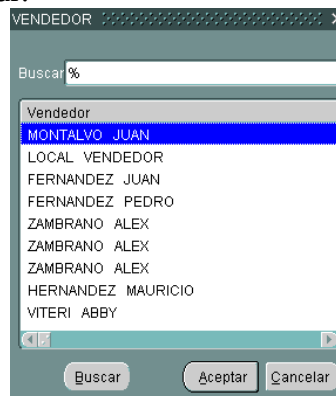
Fecha de entrega.- Se ingresa la fecha de entrega, no puede ser menor a la fecha actual y mayor a 30 días.

Turno.- Se le asigna automáticamente al momento de grabar o se puede ingresar

Urgente.- En caso de que sea urgente el examen

Observación.- Algún comentario técnico

Vendedor.- El código del vendedor o se busca en la lista desplegable, en el caso de no existir el vendedor se lo puede crear.



Botón crear vendedor.-  En el caso que el vendedor no exista, este botón activa el formulario de vendedor.

Ficha Médica.- Particular

Toma muestra.- Local o exterior

Motivo.- Comentario del médico

Laboratorio.- Comentario del laboratorio

Diagnóstico.- Comentario técnico

Test.- Automáticamente se presenta y se selecciona los test a realizarse, presionando doble clic.

Exámenes a realizar.- Se ingresa la cantidad y el descuento a realizar no puede ser mayor al precio del examen.

Cantidad.- Cantidad siempre será uno.

Descuento.- Se ingresa un valor en dólares, no puede ser mayor al precio del test.

9.2.2.3.3.2 FACTURACIÓN

En esta pantalla se genera la factura de la solicitud de examen realizada.

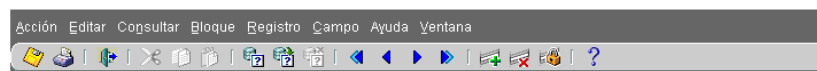
PANTALLA N°17 : FACTURACION
FUENTE: AUTORES

Barra de menú

En la opción consultar, tenemos la opción introducir que nos permite digitar el código del usuario y luego presionamos la opción ejecutar y tenemos el resultado esperado.

Barra de herramientas

En esta barra de herramientas encontrara opciones y herramientas que le permiten manipular la información que se encuentra grabada en la base de datos.



Grabar.- Graba información



Salir.- Salir de la forma



Consultar.

Introducir consulta.- Se introduce el código



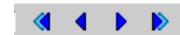
Ejecutar consulta.- Se ejecuta la consulta



Cancelar consulta.- Se cancela la consulta



Barra de navegación.- Puede visualizar de registro en registro



Nuevo.- Crea una opción de menú



Eliminar.- Desactiva registro.



No. Fact. .- Se genera un número secuencial.

Código Cliente.- Se ingresa el código del cliente.

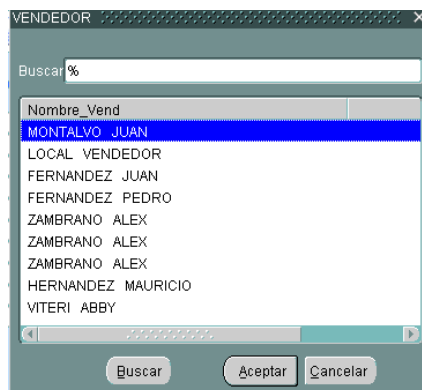
Botón Buscar Cliente.-  Presenta una lista con los clientes.



Botón crear cliente.-  Crea un cliente en caso que no esté registrado.

Vendedor.- Se ingresa el código del vendedor.

Botón buscar Vendedor.-  Presenta una lista con los vendedores.



Fecha.- Es la fecha actual en se genera la factura, se presenta automáticamente.

Botón facturar solicitud.- Presentar los documentos del cliente seleccionado, se puede seleccionar los documentos que se requieren facturar.

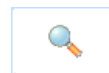


Botón seleccionar.- Selecciona el documento a facturar.

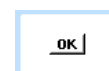


Botón buscar documento.-

Busca todos los documentos del cliente.



Botón aceptar.- Cuando ya se ha seleccionado los documento a facturar.



Botón cancelar.- Cancela la selección de los documentos.



Botón imprimir.- Presenta un reporte de la factura.



Bloque de Productos.- Presenta los productos seleccionados con los respectivos valores de los documentos.

9.2.2.3.3.3 CAJA

Esta pantalla nos permite llevar un control de los pagos que realiza el cliente sobre las facturas.

FECHA FAC.	FAC NUMERO	Cliente	MONTO	FECHA PAGO	ABONO	SALDO

PANTALLA N°18 : CAJA
FUENTE: AUTORES

Para realizar pagos a facturas de un cliente.

Cliente.- Consultamos al cliente.

Monto.- Ingresamos el monto a cancelar y el sistema automáticamente distribuye las cantidades ordenada de la deuda antigua a la más reciente.

Botón Grabar.- Clic en el botón grabar.

Para Eliminar una transacción de ingreso de caja.

Cliente. – Consultamos al cliente

Caja.- Consultamos la transacción de caja del cliente.

Botón Eliminar.- Clic en el botón Eliminar.

Para imprimir recibos de caja

Ciente. – Consultamos al cliente

Caja.- Consultamos la transacción de caja del cliente.

Botón Imprimir.- Clic en el botón imprimir.

9.2.2.3.3.4 INGRESO DE RESULTADOS

Se ingresan los resultados de los exámenes realizados al paciente.

INGRESO DE RESULTADO				
DATO ORDEN				
UPS	NO. ORDEN	TURNO	FECHA	
DATOS PACIENTE				
PACIENTE	MEDICO	GENERO	ETAPA HUMANA	FEC. NAC.
ORDENES DETALLE				
TEST	RESULTADO	IMPRIME	RANGO MIN	RANGO MAX
		☒		
		☐		
		☐		
		☐		
		☐		
		☐		
		☐		
		☐		
		☐		

PANTALLA N°19 : INGRESO DE RESULTADO
FUENTE: AUTORES

Ingreso / Modificacion/ Imprimir Resultado.

Paciente.- Consultamos al paciente al que se va ingresar los resultados.

No Orden.- Consultamos la Orden, un mismo paciente puede tener varias órdenes.

Medico.- Consultar al médico de la solicitud, también nos sirve como filtro al momento de consultar la orden.

Turno.- Es el turno asignado desde la solicitud a dicha orden.

Fecha.- Es la fecha de la solicitud

Genero.- Es el género del paciente.

Etapa Humana.- Es la descripción de etapa humana del paciente, esta información es ingresada desde la solicitud de orden.

Fecha Nacimiento.- fecha de nacimiento del paciente.

Ordenes detalle

Test.- Son la(s) prueba(s) que se realiza el paciente.

Resultado.- Donde se ingresa los resultados de cada prueba.

Imprime.- Si desea que se imprima esa prueba.

Rangos.- Son las referencias de las pruebas son campos solo de lectura.

Botón Grabar.- Dar clic al botón y se actualizar los resultados de dicha orden.

Botón Imprimir.- Dar clic en el botón imprimir y se visualiza el formato de impresión en pdf.

Botón Nuevo.- Limpia toda la pantalla de ingreso de resultado.

Botón Cancelar.- Sale de la pantalla.

9.2.2.3.4 REPORTES

9.2.2.3.4.1 ÓRDENES DEL CLIENTE

El reporte se puede presentar por fecha esta se puede ingresar o generar automáticamente desde el primer día mes hasta el último día del mes o se puede consultar por el código del cliente.

The image shows a screenshot of a software window titled 'ORDENES'. Inside the window, there is a form titled 'Ordenes del Cliente'. The form has two date fields: 'DESDE' with the value '01/08/2010' and 'HASTA' with the value '31/08/2010'. Below these, there is a 'CODIGO' field with a '%' symbol. To the right of the 'CODIGO' field is a magnifying glass icon. At the bottom right of the form is a printer icon. The window has a blue background and a title bar with standard window controls.

PANTALLA N°20 : REPORTE DE ORDENES DE CLIENTE
FUENTE: AUTORES

Desde.- Se ingresa la fecha en el formato día/mes/año

Hasta.- Se ingresa la fecha en el formato día/mes/año

Código.- Se ingresa el código del cliente.

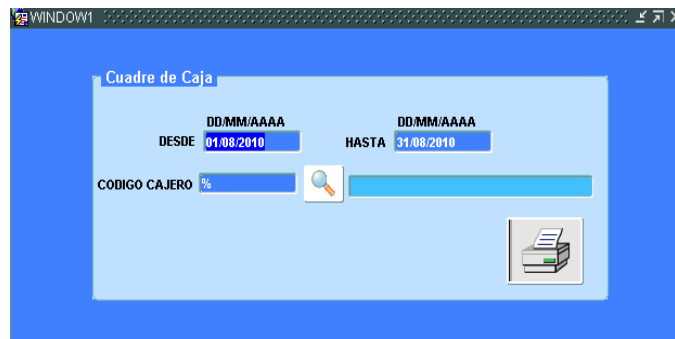
Botón buscar cliente.- Presenta una lista con los clientes existentes y sus

órdenes.

Botón imprimir.- Presenta el reporte en formato .pdf

9.2.2.3.4.2 CUADRE DE CAJA

El reporte se puede presentar por fecha esta se puede ingresar o generar automáticamente desde el primer día mes hasta el último día del mes o se puede consultar por el código del cajero.



PANTALLA N°21 : REPORTE DE CUADRE DE CAJA
FUENTE: AUTORES

Desde.- Se ingresa la fecha en el formato día/mes/año

Hasta.- Se ingresa la fecha en el formato día/mes/año

Código.- Se ingresa el código del cliente.

Botón buscar cajero.- Presenta una lista con los clientes existentes y sus órdenes.

Botón imprimir.- Presenta el reporte en formato .pdf

9.2.2.3.5 AYUDA

Presentamos un archivo de ayuda, que nos permite despegar cualquier duda que tengamos al ingresar información desde los formularios.

9.2.2.3.6 SALIR

Esta opción permite salir del sistema

9.2.3 ADMINISTRACIÓN DE USUARIOS

9.2.3.1. ADMINISTRADORES DE BASE DE DATOS

9.2.3.1.1. AZAMBRANO

NOMBRE	AZAMBRANO
PERFIL	ADMINISTRADOR
AUTENTICACIÓN	Password
TABLESPACE POR DEFECTO	LABORATORIO
TABLESPACE TEMPORAL	TEMP

ROLES:

- CONNECT
- RESOURCE

9.2.3.1.2. RASTUDILLO

NOMBRE	RASTUDILLO
PERFIL	ADMINISTRADOR
AUTENTICACIÓN	Password
TABLESPACE POR DEFECTO	LABORATORIO
TABLESPACE TEMPORAL	TEMP

ROLES:

- CONNECT
- RESOURCE

9.2.3.1.3. MALVAREZ

NOMBRE	MALVAREZ
PERFIL	ADMINISTRADOR
AUTENTICACIÓN	Password
TABLESPACE POR DEFECTO	LABORATORIO
TABLESPACE TEMPORAL	TEMP

ROLES:

- CONNECT
- RESOURCE

9.2.3.2. USUARIOS DEL SISTEMA

9.2.3.2.1. USRCAJERO

NOMBRE	USRCAJERO
PERFIL	USUARIO
AUTENTICACIÓN	Password
TABLESPACE POR DEFECTO	LABORATORIO
TABLESPACE TEMPORAL	TEMP

ROLES

- CAJERO
- CONNECT

9.2.3.2.2. USRENFERMERA

NOMBRE	USRENFERMERA
PERFIL	USUARIO
AUTENTICACIÓN	Password
TABLESPACE POR DEFECTO	LABORATORIO
TABLESPACE TEMPORAL	TEMP

ROLES:

- CONNECT
- ROLENFERMERA

9.2.3.2.3. USRMEDICO

NOMBRE	USRMEDICO
PERFIL	ADMINISTRADOR
AUTENTICACIÓN	Password
TABLESPACE POR DEFECTO	LABORATORIO
TABLESPACE TEMPORAL	TEMP

ROLES:

- CONNECT
- MEDICO

9.2.3.2.4. USRSECRETARIA

NOMBRE	USRSECRETARIA
PERFIL	USUARIO
AUTENTICACIÓN	Password
TABLESPACE POR DEFECTO	LABORATORIO
TABLESPACE TEMPORAL	TEMP

ROLES:

- CONNECT
- MEDICO

9.2.3.3. DEFINICION DE ROLES

9.2.3.3.1. CAJERO

PRIVILEGIO DE OBJETO	TABLA
INSERT SELECT UPDATE	CUENTAXCOBRAR
INSERT SELECT UPDATE	FORMA_PAGO
INSERT SELECT UPDATE	TIPOMOVIMIENTO
INSERT SELECT UPDATE	TRXCAJA
INSERT SELECT UPDATE	TRXCAJADETALLE
INSERT SELECT UPDATE	CLIENTE
INSERT SELECT UPDATE	DOCUMENTO
INSERT SELECT UPDATE	DOCUMENTODETALLE
INSERT SELECT UPDATE	FACTURA
INSERT SELECT UPDATE	FACTURADETALLE

INSERT SELECT UPDATE	PRODUCTO
INSERT SELECT UPDATE	VENDEDOR

9.2.3.3.2 . MEDICO

PRIVILEGIO DE OBJETO	TABLA
INSERT SELECT UPDATE	CUENTAXCOBRAR
INSERT SELECT UPDATE	FORMA_PAGO
INSERT SELECT UPDATE	TIPOMOVIMIENTO
INSERT SELECT UPDATE	TRXCAJA
INSERT SELECT UPDATE	TRXCAJADETALLE
INSERT SELECT UPDATE	CLIENTE
INSERT SELECT UPDATE	DOCUMENTO
INSERT SELECT UPDATE	DOCUMENTODETALLE
INSERT SELECT UPDATE	FACTURA
INSERT SELECT UPDATE	FACTURADETALLE

INSERT SELECT UPDATE	PRODUCTO
INSERT SELECT UPDATE	VENDEDOR

Privilegio de Objeto	TABLA
INSERT	ACCESO
SELECT	
UPDATE	
INSERT	CATALOGO
SELECT	
UPDATE	
INSERT	CATALOGOTABLA
SELECT	
UPDATE	
INSERT	EMPRESA
SELECT	
UPDATE	
INSERT	EQUIPO
SELECT	
UPDATE	
INSERT	ETAPAHUMANA
SELECT	
UPDATE	
INSERT	INTERFAZEQUIPO
SELECT	
UPDATE	
INSERT	MEDICO
SELECT	
UPDATE	
INSERT	MENU
SELECT	
UPDATE	
INSERT	ORDEN
SELECT	
UPDATE	

INSERT	ORDENDETALLE
SELECT	
UPDATE	
INSERT	PACIENTE
SELECT	
UPDATE	

Privilegio de Objeto	TABLA
INSERT	RANGO
SELECT	
UPDATE	
INSERT	RANGODETALLE
SELECT	
UPDATE	
INSERT	RESULTADOEQUIPO
SELECT	
UPDATE	
INSERT	TEST
SELECT	
UPDATE	
INSERT	USUARIO
SELECT	
UPDATE	

9.2.3.3.3 . ROLENFERMERA

Privilegio de Objeto	TABLA
INSERT SELECT UPDATE	ORDENDETALLE
INSERT SELECT UPDATE	RANGO

9.2.3.3.4 . ROL: ROLSECRETARIA

Privilegio de Objeto	TABLA
INSERT	ACCESO
SELECT	
UPDATE	
INSERT	CATALOGO
SELECT	
UPDATE	
INSERT	CATALOGOTABLA
SELECT	
UPDATE	
INSERT	EMPRESA
SELECT	
UPDATE	
INSERT	EQUIPO
SELECT	
UPDATE	
INSERT	ETAPAHUMANA
SELECT	
UPDATE	
INSERT	INTERFAZEQUIPO
SELECT	
UPDATE	
INSERT	MEDICO
SELECT	
UPDATE	
INSERT	MENU
SELECT	

Privilegio de Objeto	TABLA
INSERT SELECT UPDATE	MENU
INSERT	ORDEN
SELECT UPDATE	
INSERT SELECT UPDATE	ORDENDETAL LE
INSERT SELECT UPDATE	PACIENTE
INSERT SELECT UPDATE	RANGO
INSERT SELECT UPDATE	RANGODETA LLE
INSERT SELECT UPDATE	RESULTADOE QUIPO
INSERT SELECT UPDATE	TEST
INSERT SELECT UPDATE	USUARIO

9.2.3.3.5 . TECNOLOGOLAB

Privilegio de Objeto	TABLA
INSERT SELECT UPDATE	ETAPAHUMANA
INSERT SELECT UPDATE	ORDENDETALLE
INSERT SELECT UPDATE	PACIENTE
INSERT SELECT UPDATE	RANGO
INSERT SELECT UPDATE	RANGODETALLE
INSERT SELECT UPDATE	RESULTADOEQUIPO
INSERT SELECT UPDATE	TEST

9.2.3.4 . CREACIÓN DE USUARIOS

USUARIOS ADMINISTRADORES
Crear usuario en la base datos create user rastudillo identified by rastudillo default tablespace laboratorio temporary tablespace temp profile default;
PERMISOS grant connect to rastudillo; grant resource to rastudillo;
CREAR USUARIO EN LA BASE DATOS create user azambrano identified by azambrano default tablespace laboratorio temporary tablespace temp profile default;
PERMISOS grant connect to azambrano; grant resource to azambrano;
CREAR USUARIO EN LA BASE DATOS create user malvarez identified by malvarez default tablespace laboratorio temporary tablespace temp profile default;
PERMISOS grant connect to malvarez; grant resource to malvarez;
create user sch_caja identified by sch_caja default tablespace laboratorio temporary tablespace temp profile default; grant connect to sch_caja; -- Otorga el rol a un usuario específico grant resource to sch_caja; -- otorga permisos de select al rol grant dba to sch_caja

QUITAR PERMISOS
revoke dba from rastudillo;
Crear sinónimos
create public synonym paciente for sch_lab.paciente select * from paciente

10. CONCLUSIONES Y RECOMENDACIONES

10.1. CONCLUSIONES

En este trabajo de tesis hemos planteado la automatización de un laboratorio clínico, mediante el análisis, diseño, desarrollo e implementación e investigación de la herramienta Oracle. Esto nos ha permitido desarrollar un sistema para la automatización de los procesos de un laboratorio clínico.

La utilización de Oracle 10G nos permitió enfocarnos a las nuevas tecnologías como las apoyadas sobre interfaces web, que permiten modernizar sistemas orientados a cliente servidor, y expandir el acceso desde otros nodos de una red, sin requerir la instalación de software adicional, a excepción de un browser.

Esta tecnología nos ha permitido la integración por medio de una interfaz única el browser que puede funcionar en una extranet o intranet, se puede adaptar perfectamente para hacer posible la implementación del concepto de Automatización del Laboratorio Clínico, de una forma rápida, ágil y adecuada.

Uno de los puntos más importantes dentro del concepto de Automatización de un laboratorio clínico, es que todos los integrantes del laboratorio, tengan acceso a la información las 24 horas del día y los 7 días de la semana.

También podemos destacar es la baja curva de aprendizaje, necesaria para desarrollar e implementar aplicaciones utilizando Oracle Forms y agregando la cualidad de poder instalarse en cualquier sistema operativo.

El ciclo de desarrollo de un sistema basado en la metodología de Análisis y Diseño Estructurado nos permitió realizar un análisis de cada una de las necesidades que tenía nuestro proyecto y así poder diseñar y desarrollar un sistema robusto que se ajuste a las necesidades de un laboratorio clínico.

10.2. RECOMENDACIONES

Para este proyecto las recomendaciones que realizamos son las siguientes:

- Para el buen manejo del sistema es recomendable seguir el manual de usuario, el mismo que muestra un detalle de cada uno de los pasos a seguir de las diferentes pantallas y así ingresar información consistente.
- El requerimiento mínimo del hardware es indispensable para un buen rendimiento en flujo de la información.
- Para mantener la seguridad se recomienda cambiar las claves cada cierto tiempo (mínimo dos meses).
- Se recomienda un constante mantenimiento al sistema y al hardware para su óptima operatividad ya que esto ocasionaría rentabilidad en el futuro, mejoras en el conocimiento de las funciones y además agiliza los procesos del laboratorio clínico.

BIBLIOGRAFIA

RICCIO CHAVEZ, Francisco Martin, “Administracion De Oracle”, Alta Via Consulting- America Latina, EEUU 2008.

DJK, Bloques En Pl/Sql, 04 de marzo de 2006

<http://www.devjoker.com/contenidos/Tutorial-PLSQL/27/Bloques-PLSQL.aspx>

DJK, Cursores En Pl. /Sql, 04 de marzo de 2006

<http://www.devjoker.com/contenidos/Tutorial-PLSQL/32/Cursores-en-PLSQL.aspx>

Wiki pedía, Cadena de valor, 1 julio 2010

http://es.wikipedia.org/wiki/Cadena_de_valor

Zona Oracle, comunidad Oracle en español, 05/02/2007

<http://www.zonaoracle.com/?q=node/2193>

Oracle Tecnology, Definición de Oracle, 01/04/2010,

<http://www.oracle.com/technology/products/forms>

Wiki pedía, Definición De Procesos, 25 jul. 2010

[http://es.wikipedia.org/wiki/Reingenier%C3%ADa_de_procesos#Definici.C3.B3n_de_p
roceso](http://es.wikipedia.org/wiki/Reingenier%C3%ADa_de_procesos#Definici.C3.B3n_de_proceso)

Wiki pedía, Diagramas de flujo, 29 jul. 2010

http://es.wikipedia.org/wiki/Diagrama_de_flujo

Universidad de Córdoba, Estándares de Programación, Mayo 1993

<http://www.gestion.uco.es/gestion/aplicaciones/docs/NormasyEstandares.pdf>

DJK, Excepciones En Pl/Sql, 04 de marzo de 2006

<http://www.devjoker.com/contenidos/catss/48/Excepciones-en-PLSQL.aspx>

DJK, Funciones En Pl/Sql, 04 de marzo de 2006

<http://www.devjoker.com/contenidos/catss/53/Funciones-en-PLSQL.aspx>

Wiki pedía, Gestión de procesos de negocio, 13 julio 2010

http://es.wikipedia.org/wiki/Gesti%C3%B3n_de_procesos_de_negocio.

MORENO, Francisco, “Introducción A Programación Orientado A Objetos”, grupo EIDOS, Argentina 1999-2000.

Wiki pedía, modelo entidad relación, 26/03/2010

http://es.wikipedia.org/wiki/modelo_entidad-relaci%C3%B3n.

A.LITTLE, Thomas, “Modulación Y Análisis De Procesos Empresariales”, Alta Via Consulting- America Latina, EEUU 2003.

ESPINOZA, Francisco, “Oracle Forms 6i”, Publiguias, Argentina 2003.

Todo experto, Oracle, 02/08/2002,

<http://www.todoexpertos.com/categorias/tecnologia-e-internet/bases-de-datos/respuestas/163504/porque-utilizar-oracle>

Monografías, Oracle, 05/06/2010

<http://www.monografias.com/trabajos25/oracle/oracle.shtml>

DJK, Procedimientos En Pl/Sql, 04 de marzo de 2006

<http://www.devjoker.com/contenidos/catss/52/Procedimientos-almacenados-en-PLSQL.aspx>

Centro blogs, ¿Que Es Pl/Sql?, 12 de septiembre de 2006

<http://www.plsql.biz/2006/09/qu-es-el-plsql.html>

Mailxmail, Tipo De Procesos, 16/12/2009

<http://www.mailxmail.com/curso-gestion-calidad-procesos-tecnicas-herramientas-calidad/tipos-procesos>

FOWLET, Martin y SCOTT, Kendal, “Uml Gota A Gota”, Editorial Ferson, Mexico 1999.

SCHMULLER, Joseph, “Uml En 24 Horas”, Prentice Hall, Argentina 1997.

Anexos

ANEXO 1: ESTRUCTURA DE FUNCIONAMIENTO DEL SISTEMA

