

**Universidad Católica Boliviana San Pablo**  
Unidad Académica Santa Cruz  
Carrera de Ingeniería de Sistemas



## **PROYECTO DE GRADO**

### **“GIMNASOFT”**

“Sistema de Información para el Seguimiento de las Actividades  
Físicas de los Clientes y Control Económico  
del Gimnasio Natural Body Club”

Tesis para optar al Título de Licenciado en Ingeniería de Sistemas

**Autor:**  
Bernardo Campos Saavedra

Santa Cruz de la Sierra – Bolivia  
2007

## ***Dedicatoria***

*Para mis padres quienes fueron las personas  
que me motivaron para que salga profesional.  
A mis hermanos que me apoyaron incondicionalmente.  
A mi esposa Rocío y mi hijita Dayana  
Quienes son mi inspiración para salir adelante.*

## *Agradecimientos*

A Dios por darme la vida, la salud, y la fuerza de voluntad para culminar este trabajo.

A mi Madre Matilde Saavedra, por darme el apoyo y la paciencia incondicional. A mi padre Cristino Campos por la constante preocupación demostrada para que salga adelante en mis estudios.

A mi esposa Rocío Cuellar por haber pasado los momentos difíciles, quien comprendió y sacrificó su tiempo a mi lado. A mi hija Dayana que fue la fuente de mi inspiración.

A todos mis hermanos, quienes me apoyaron en el transcurso de mi carrera, a Wilmer, quien incondicionalmente me apoyó en todos los aspectos, económica, técnica y moralmente.

Al Lic. Edwin Vargas, por la colaboración, paciencia y motivación recibida para llevar a cabo este trabajo.

A la Universidad y a todos los docentes que me formaron durante los años de estudio.

Al Sr. Wilson Machicado, Administrador del Gimnasio “Natural Body Club”, quien abrió sus puertas y confianza depositada en mi persona.

A todas las personas que de una u otra forma han colaborado para la culminación del trabajo.

# INDICE DE CONTENIDO

INTRODUCCION .....	1
<b>PARTE I. ANTECEDENTES Y OBJETIVOS.....</b>	<b>2</b>
<b>CAPITULO 1. GIMNASIO “NATURAL BODY CLUB” Y DESCRIPCIÓN DEL PROYECTO “GIMNASOFT” .....</b>	<b>3</b>
1.1 El Gimnasio “Natural Body Club” .....	3
1.2 Visión del Gimnasio .....	3
1.3 Misión del Gimnasio.....	4
1.4 Organigrama del Gimnasio.....	4
1.5 Estadísticas.....	5
1.6 Flujograma de Atención en el Gimnasio “Natural Body Club” .....	8
<b>CAPITULO 2. OBJETIVOS DEL PROYECTO “GIMNASOFT” .....</b>	<b>11</b>
2.1 Antecedentes y Justificaciones .....	11
2.2 Objetivos del Proyecto.....	13
2.2.1 Objetivo General.....	14
2.2.2 Objetivos Específicos .....	14
2.3 Alcance del Proyecto .....	14
<b>PARTE II. MARCO TEÓRICO .....</b>	<b>20</b>
<b>CAPITULO 3. CONCEPTOS Y TERMINOLOGIAS UTILIZADAS EN EL GIMNASIO “NATURAL BODY CLUB” .....</b>	<b>21</b>
3.1 El Ejercicio .....	21
3.2 Signos Vitales .....	23
3.3 Instructor .....	23
3.4 Casillero .....	23
3.5 Estado del Cliente .....	23
3.6 Ficha Técnica.....	24
3.7 Medidas Antropométricas.....	24
3.8 Paquetes .....	25
3.9 Los Antecedentes.....	25
3.10 El Calentamiento.....	25
3.11 Sobreentrenamiento .....	26
3.12 Normas para realizar ejercicios.....	27
3.13 Las Vitaminas .....	28
3.14 Peso Ideal.....	29
3.15 Como medir la aptitud física.....	30
3.16 Bondades que aporta el ejercicio físico .....	31
3.17 Ejercicios Respiratorios .....	31
3.18 La Concentración.....	32

<b>CAPITULO 4. CONCEPTOS DEL FLUJO DE CAJA.....</b>	<b>33</b>
4.1 Flujo de Caja.....	33
4.2 Cuentas que Aumentan el Flujo de Caja.....	34
4.3 Cuentas que Disminuyen el Flujo de Caja.....	35
4.4 Control Interno.....	37
 <b>CAPITULO 5. ADMINISTRACION DE GIMNASIOS .....</b>	<b>38</b>
5.1 Administración de Gimnasios (Instalaciones Deportivas).....	38
5.1.1 Otros servicios complementarios.....	40
5.1.2 Plan general de funcionamiento.....	40
5.1.3 Redacción de las normas y los reglamentos de uso .....	40
5.1.4 Plan de Mantenimiento .....	40
5.2 Disciplinas .....	41
5.2.1 Aeróbic / Anaeróbicos .....	41
5.2.2 Natación .....	42
5.2.3 Sauna.....	42
5.2.4 El Spinning .....	43
5.2.5 El Hapkido .....	44
5.2.6 Taekwondo.....	45
5.2.7 Entrenamiento con pesas .....	46
 <b>CAPÍTULO 6. LENGUAJE UNIFICADO DE MODELADO (UML) .....</b>	<b>47</b>
6.1 Introducción a UML .....	47
6.2 Bloques de Construcción de UML.....	48
6.2.1 Elementos en UML.....	48
6.2.2 Relaciones de UML .....	53
6.2.3 Diagramas de UML .....	55
 <b>CAPITULO 7. EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE.....</b>	<b>59</b>
7.1 Introducción .....	59
7.2 Un Proceso dirigido por Casos de Uso .....	60
7.3 Un Proceso centrado en la Arquitectura .....	61
7.4 Un Proceso iterativo e incremental .....	61
7.5 La Fase de Inicio establece la viabilidad .....	62
7.6 La Fase de Elaboración se centra en la Factibilidad.....	63
7.7 La Fase de Construcción construye el Sistema.....	63
7.8 La Fase de Transición se mete dentro del entorno del usuario .....	64
7.9 Captura de Requisitos .....	64
7.10 Análisis .....	68
7.11 Diseño .....	72
7.12 Implementación .....	77
7.13 Prueba .....	81

<b>PARTE III. PROCESO DE DESARROLLO DEL SOFTWARE.....</b>	<b>83</b>
<b>CAPITULO 8. MODELO DE NEGOCIO Y DE DOMINIO.....</b>	<b>84</b>
8.1 Modelo de Negocios.....	84
8.1.1 Procesos de Negocio.....	84
8.1.2 Diagrama de Actividades.....	86
8.2 Compresión del Contexto Mediante un Modelo de Dominio .....	98
<b>CAPITULO 9. MODELO DE REQUISITOS .....</b>	<b>100</b>
9.1 Captura de requisitos como casos de usos.....	100
9.1.1 Identificación de actores .....	100
9.1.2 Identificación y descripción de los casos de uso .....	101
<b>CAPITULO 10. MODELO DE ANÁLISIS .....</b>	<b>115</b>
10.1 Análisis de la Arquitectura .....	115
10.1.1 Identificación de Paquetes del Análisis .....	115
10.1.1 Detallando los Paquetes del Análisis.....	116
10.2 Análisis de los Casos de Uso.....	119
10.3 Análisis de Clases .....	131
10.3.1 Clases de Interfaz.....	131
10.3.2 Clases de Control.....	135
10.3.3 Clases de Identidad.....	138
<b>CAPITULO 11. MODELO DE DISEÑO .....</b>	<b>141</b>
11.1 Diseño de la Arquitectura.....	141
11.1.1 Diagramas de Despliegue .....	141
11.1.2 Subsistemas de Diseño .....	142
11.2 Diseño de los Casos de Uso.....	144
11.3 Diseño de las Clases .....	160
<b>CAPITULO 12. MODELO DE IMPLEMENTACION.....</b>	<b>161</b>
12.1 Implementación de la Arquitectura .....	161
12.1.1 Identificación de componentes. ....	162
12.1.2 Despliegue de Componentes sobre los Nodos.....	162
12.2 Implementación de Clases.....	163
12.2.1 Implementación de la Clase de Acceso a Datos .....	165
12.2.2 Implementación de la Clase Control de Interfaz. ....	171
12.2.3 Implementación de la Clase Entidad “Cliente”. ....	174
12.2.4 Implementaciones de Operaciones .....	180
12.2.5 Otras Implementaciones .....	181

<b>CAPITULO 13. MODELO DE PRUEBAS.....</b>	<b>182</b>
13.1 Pruebas de Unidad .....	182
13.2 Pruebas de Integración.....	184
13.3 Pruebas de Sistema. ....	185
Conclusiones.....	188
Recomendaciones .....	189
Bibliografía.....	190
Otras Fuentes de Información.....	191
Entrevistas .....	192
<b>ANEXOS .....</b>	<b>193</b>
Anexo A	
Anexo B	

## INDICE DE FIGURAS

Figura 1.1	Organigrama del Gimnasio “Natural Body Club” .....	1
Figura 1.2	Clientes inscritos a disciplinas en el mes de Mayo .....	5
Figura 1.3	Grafico de Clientes inscritos a disciplinas en el mes de Mayo .....	6
Figura 1.4	Clientes Inscritos a Disciplinas en el mes de Junio.....	6
Figura 1.5	Grafico de Clientes Inscritos a Disciplinas en el mes de Junio.....	7
Figura 1.6	Actividad realizada vs Tiempo en Minutos .....	7
Figura 1.7	Grafico de Actividad realizada vs Tiempo en Minutos .....	8
Figura 1.8	Flujograma de Atención al Cliente en el Gimnasio .....	9
Figura 3.1	Vitaminas en Alimentos .....	29
Figura 3.2	Peso Ideal según Altura.....	30
Figura 3.3	Respiración con los labios fruncidos.....	32
Figura 5.1	Ejercicio Aeróbic .....	42
Figura 5.2	Máquina para realizar Spinning .....	43
Figura 5.3	El Hapkido .....	45
Figura 6.1	Clases .....	48
Figura 6.2	Interfaces .....	49
Figura 6.3	Colaboraciones .....	49
Figura 6.4	Casos de Uso .....	50
Figura 6.5	Clases Activas .....	50
Figura 6.6	Componentes.....	50
Figura 6.7	Nodos .....	51
Figura 6.8	Mensajes.....	51
Figura 6.9	Estados .....	52
Figura 6.10	Paquetes.....	52
Figura 6.11	Notas .....	53
Figura 6.12	Dependencias .....	54
Figura 6.13	Asociaciones .....	54
Figura 6.14	Generalizaciones .....	54
Figura 6.15	Realizaciones.....	54
Figura 6.16	Diagrama de Clases.....	55
Figura 6.17	Diagrama de Casos de Uso .....	56
Figura 6.18	Diagrama de Secuencia .....	57
Figura 6.19	Diagrama de Estados.....	57
Figura 6.20	Diagrama de Componentes .....	58
Figura 6.21	Diagrama de Despliegue .....	58
Figura 7.1	Un Ciclo con sus fases e iteraciones .....	60
Figura 7.2	Los Casos de Uso enlazan los flujos de trabajo .....	60
Figura 7.3	Proceso Unificado de Desarrollo de Software .....	62
Figura 7.4	Flujo de trabajo para la Captura de Requisitos en forma de Casos de Uso .....	64
Figura 7.5	Las entradas y los resultados de identificar actores y Casos de Uso	65
Figura 7.6	Entradas y resultados de Priorizar los Casos de Uso .....	66



Figura 7.7	Entradas y resultados de Detallar un Caso de Uso.....	66
Figura 7.8	Entrada y resultados de Prototipar la interfaz de Usuario.....	67
Figura 7.9	Las entradas y resultados de encontrar generalizaciones en el modelo de Casos de Uso.....	68
Figura 7.10	El flujo de trabajo en el análisis, incluyendo a los trabajadores participantes y sus actividades .....	68
Figura 7.11	Las entradas y los resultados del análisis arquitectónico .....	69
Figura 7.12	Las entradas y los resultados del análisis de un Caso de Uso.....	70
Figura 7.13	Las entradas y los resultados del análisis de una clase .....	71
Figura 7.14	Las entradas y los resultados del análisis de un paquete.....	72
Figura 7.15	El flujo de trabajo en el diseño, incluyendo los trabajadores participantes y sus actividades .....	73
Figura 7.16	Las entradas y los resultados del diseño de la arquitectura.....	74
Figura 7.17	Las entradas y los resultados del diseño de un caso de uso .....	75
Figura 7.18	Las entradas y los resultados del diseño de una clase.....	76
Figura 7.19	Las entradas y los resultados del diseño de un subsistema .....	76
Figura 7.20	Flujo de trabajo en la etapa de implementación, incluyendo los trabajadores participantes y sus actividades.....	77
Figura 7.21	La entrada y el resultado de la implementación de la arquitectura...	78
Figura 7.22	Entrada y resultado de la integración de sistemas.....	79
Figura 7.23	La entrada y el resultado de la implementación de subsistemas .....	79
Figura 7.24	La entrada y el resultado de la implementación de una clase .....	80
Figura 7.25	La entrada y el resultado de la prueba de una unidad .....	81
Figura 8.1	Diagrama de Procesos de Negocio del Dominio.....	86
Figura 8.2	Diagrama de Actividad del Proceso “Inscribir Cliente” .....	87
Figura 8.3	Diagrama de Actividad del Proceso “Registrar Venta” .....	88
Figura 8.4	Diagrama de Actividad del Proceso “Registrar Compra” .....	88
Figura 8.5	Diagrama de Actividad del Proceso “Congelar Cliente” .....	89
Figura 8.6	Diagrama de Actividad del Proceso “Controlar Casilleros” .....	90
Figura 8.7	Diagrama de Actividad del Proceso “Pagar a Empleado” .....	91
Figura 8.8	Diagrama de Actividad del Proceso “Registrar Ficha Clínica” .....	92
Figura 8.9	Diagrama de Actividad del Proceso “Verificar Ingresos/Egresos”...	92
Figura 8.10	Diagrama de Actividad del Proceso “Otorgar Dieta” .....	93
Figura 8.11	Diagrama de Actividad del Proceso “Definir Paquete” .....	94
Figura 8.12	Diagrama de Actividad del Proceso “Pagar Deuda de Cliente” .....	95
Figura 8.13	Diagrama de Actividad del Proceso “Crear Disciplina” .....	96
Figura 8.14	Diagrama de Actividad del Proceso “Controlar Asistencia .....	97
Figura 8.15	Diagrama de Clases de Dominio.....	99
Figura 9.1	Caso de Uso: Abrir Caja .....	102
Figura 9.2	Caso de Uso: Cerrar Caja.....	103
Figura 9.3	Caso de Uso: Pagar a Empleado .....	103
Figura 9.4	Caso de Uso: Inscribir Cliente .....	104
Figura 9.5	Caso de Uso: Congelar Inscripción.....	105
Figura 9.6	Caso de Uso: Registrar Venta .....	106

Figura 9.7	Caso de Uso: Definir Disciplina .....	107
Figura 9.8	Caso de Uso: Pagar Deuda de Cliente.....	107
Figura 9.9	Caso de Uso: Registrar Cliente .....	108
Figura 9.10	Caso de Uso: Registrar Producto .....	110
Figura 9.11	Caso de Uso: Registrar Compra.....	110
Figura 9.12	Caso de Uso: Buscar Cliente.....	111
Figura 9.13	Caso de Uso: Registrar Empleado.....	112
Figura 9.14	Caso de Uso: Buscar Empleado .....	113
Figura 9.15	Caso de Uso: Controlar Asistencia .....	114
Figura 10.1	Paquetes del Análisis.....	116
Figura 10.2	Paquete “Gestión de Empleado” .....	116
Figura 10.3	Paquete de “Gestión de Producto” .....	117
Figura 10.4	Paquete “Gestión de Clientes” .....	117
Figura 10.5	Paquete “Gestión de Venta”.....	118
Figura 10.6	Paquete “Gestión de Compra” .....	118
Figura 10.7	Paquete “Gestión de Paquete .....	118
Figura 10.8	Paquete “Gestión de Caja” .....	119
Figura 10.9	Diagrama de Colaboración del Caso de Uso “Abrir Caja” .....	119
Figura 10.10	Diagrama de Colaboración del Caso de Uso “Cerrar Caja” .....	120
Figura 10.11	Diagrama de Colaboración del Caso de Uso “Pagar a Empleado” .....	121
Figura 10.12	Diagrama de Colaboración del Caso de Uso “Inscribir Cliente” .....	122
Figura 10.13	Diagrama de Colaboración del Caso de Uso “Congelar Inscripción” .....	123
Figura 10.14	Diagrama de Colaboración del Caso de Uso “Registrar Venta” .....	124
Figura 10.15	Diagrama de Colaboración del Caso de Uso “Definir Disciplina” .....	125
Figura 10.16	Diagrama de Colaboración del Caso de Uso “Pagar Deuda de Cliente” .....	126
Figura 10.17	Diagrama de Colaboración del Caso de Uso “Registrar Cliente” .....	127
Figura 10.18	Diagrama de Colaboración del Caso de Uso “Registrar Producto” .....	128
Figura 10.19	Diagrama de Colaboración del Caso de Uso “Registrar Compra” .....	129
Figura 10.20	Diagrama de Colaboración del Caso de Uso “Registrar Empleado” .....	129
Figura 10.21	Diagrama de Colaboración del Caso de Uso “Controlar Asistencia” .....	130
Figura 11.1	Diagrama de Despliegue para el Sistema.....	142
Figura 11.2	Subsistemas para el sistema GimnaSoft.....	143

Figura 11.3	Diagrama de secuencia del Caso de Uso “Abrir Caja” .....	144
Figura 11.4	Interfaz de Usuario para el Caso de Uso “Abrir Caja” .....	145
Figura 11.5	Diagrama de secuencia del Caso de Uso “Cerrar Caja” .....	146
Figura 11.6	Interfaz de Usuario para el Caso de Uso “Cerrar Caja” .....	147
Figura 11.7	Diagrama de secuencia del Caso de Uso “Pagar a Empleado” .....	148
Figura 11.8	Interfaz de Usuario para el Caso de Uso “Pagar Empleado” .....	149
Figura 11.9	Diagrama de Secuencia del Caso de Uso “Inscribir Cliente” .....	150
Figura 11.10	Interfaz de Usuario para el Caso de Uso “Inscribir Cliente” .....	151
Figura 11.11	Diagrama de secuencia del caso de Uso “Congelar Cliente” .....	152
Figura 11.12	Interfaz de Usuario para el Caso de Uso “Congelar Cliente” .....	153
Figura 11.13	Diagrama de Secuencia del Caso de Uso “Pagar Deuda De Cliente” .....	154
Figura 11.14	Interfaz de Usuario para el Caso de Uso “Pagar Deuda De Cliente” .....	155
Figura 11.15	Diagrama de Secuencia del Caso de Uso “Definir Disciplina” .....	156
Figura 11.16	Interfaz de Usuario para el Caso de Uso “Definir Disciplina” .....	157
Figura 11.17	Diagrama de Secuencia del Caso de Uso “Controlar Asistencia” .....	158
Figura 11.18	Interfaz de Usuario para el Caso de Uso “Controlar Ingreso de Clientes” .....	159
Figura 12.1	Componentes identificados dentro de cada subsistema .....	162
Figura 12.2	Asignación de Componentes a Nodos.....	163
Figura 13.1	Validación de usuarios .....	184
Figura 13.2	Tabla Caja, cargada con mas de 5000 registros .....	186
Figura 13.3	Uso de la CPU al cargar datos de caja .....	186

# INTRODUCCION

Actualmente los Gimnasios cuentan con máquinas que son capaces de controlar: el nivel de grasa, las distancias recorridas, controlar el peso, medir la fuerza, medir la altura, pero no cuentan con un sistema totalmente integrado que lleve el control de esta información por cada cliente además del control del movimiento económico que se origina y de esta forma hacerla accesible tanto para los clientes y el personal administrativo cuando se la requiera.

El presente documento describe el desarrollo de un Sistema de Información para el Gimnasio “Natural Body Club” el cual contemplará: El registro de los alumnos a las diferentes disciplinas que ofrece el Gimnasio, los créditos otorgados a los mismos para el gimnasio y para el consumo de productos. El control muscular de los alumnos definido por fechas, el control de sus dietas y antecedentes será registrado por la secretaria previamente controlados por un Instructor. Los pagos y créditos al mismo personal serán controlados por el sistema. Todos los movimientos económicos originados por el Gimnasio como los mencionados arriba se podrán registrar en un Flujo de Caja el cual mantendrá un saldo de acuerdo al movimiento originado sea Ingreso o Egreso.

En el Capítulo I, se conoce el Gimnasio a través de la misión, visión, datos estadísticos y un organigrama que muestra la estructura del mismo. En el Capítulo 2, se da a conocer los antecedentes y justificaciones, los objetivos a cumplir y el alcance a contemplar en el proyecto. En el Capítulo 3, se definen terminologías propiamente del Gimnasio. En el Capítulo 4, se define la terminología utilizada para el manejo de la parte económica específicamente para el Flujo de Caja. En el Capítulo 5, se explican algunos conceptos claves para la administración de un Gimnasio, además de algunas disciplinas que se imparten. En los Capítulos 6 y 7 se describen la parte teórica de los modelos que se aplicaran para el desarrollo del sistema. A partir del tema 8 al 13 se describen los diferentes modelos que han guiado para el desarrollo del software.

## **PARTE I**

## **ANTECEDENTES Y OBJETIVOS**

## **CAPITULO 1.**

### **GIMNASIO “NATURAL BODY CLUB”**

El presente capítulo describe al Gimnasio “Natural Body Club”, se describe su misión, su visión, la estructura organizacional, algunos datos estadísticos, además de un flujograma general que describe el proceso de atención a clientes y sus diferentes secciones.

#### **1.1 EL GIMNASIO “NATURAL BODY CLUB”**

“Natural Body Club” nace por la iniciativa de un Joven Profesional, en el año 2002, con el objetivo de preparar profesionales de alta competición solo en las áreas de Full Contact y Kick Boxing. En el año 2003, después de haber obtenido logros con estas dos disciplinas, se decide invertir para establecer gran parte de las secciones con que cuenta actualmente el Gimnasio.

#### **1.2 VISION DEL GIMNASIO**

La Visión es la descripción de como se vería la Institución si se llevan a cabo con éxito sus estrategias de desarrollo y alcanza su mayor potencial. Al interior de esa descripción se deben encontrar la misión, las estrategias básicas, los criterios de desempeño, las más importantes normas para la toma de decisiones y los estándares éticos y técnicos que se esperan de todos los funcionarios. Por tanto el Gimnasio “Natural Body” declara su siguiente Visión:

*“Natural Body Club tiene la visión de desarrollar una excelencia en calidad de servicio dentro de la actividad deportiva, comprometidas con la salud y el bienestar integral de nuestros socios y la preparación de atletas competitivos y representativos.”*

### **1.3 MISION DEL GIMNASIO**

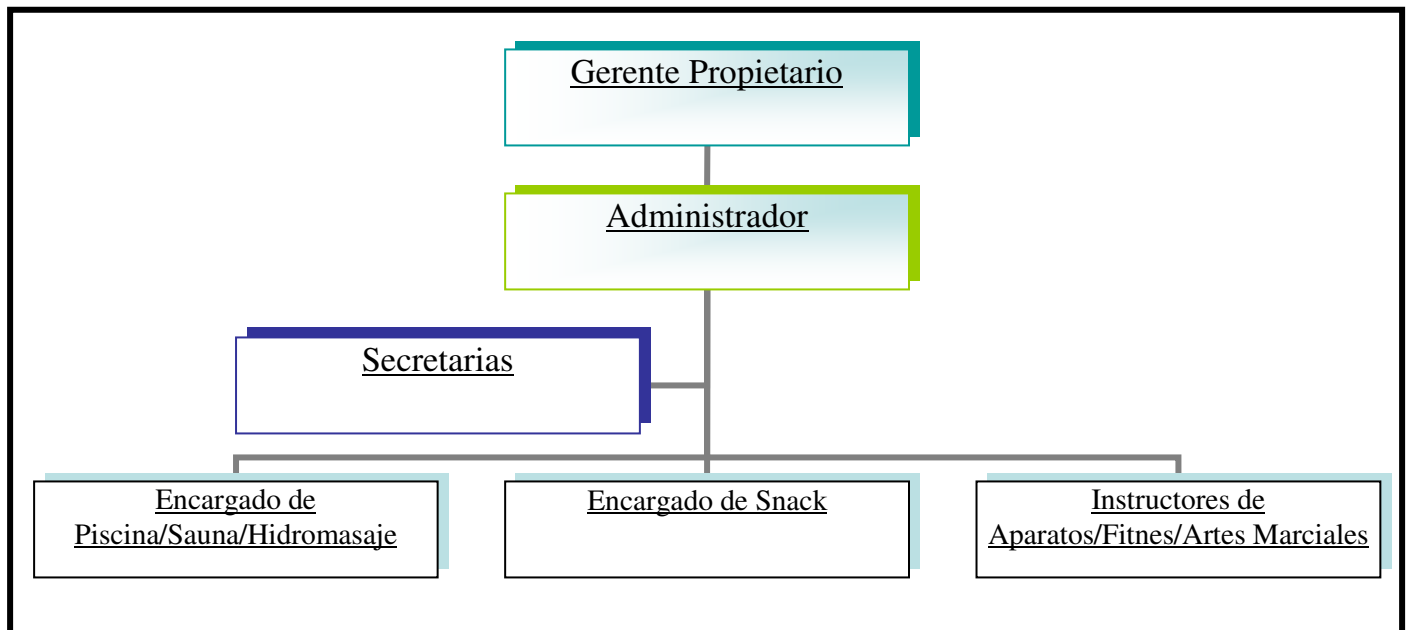
La Misión de una organización es donde se da a conocer la razón de ser de la organización. Específica el rol funcional que la organización va a desempeñar en su entorno e indica con claridad el alcance y dirección de sus actividades. El Gimnasio “Natural Body” declara su siguiente Misión:

*“Es nuestra misión brindar un ambiente idóneo para la preparación física, potencializando el rendimiento humano en sus distintas actividades bajo la supervisión de su personal idóneo y capacitado para dicho fin.”*

### **1.4 ORGANIGRAMA**

En la figura 1.1 se muestra el organigrama del Gimnasio, donde se puede observar todas las áreas cómo están organizadas actualmente.

Como se puede observar en la figura en la jerarquía más alta se encuentra el Gerente Propietario, quien tiene como subordinado al Administrador del Gimnasio y éste a su vez está encargado de organizar, planificar y orientar las diferentes secciones del Gimnasio, y del personal que trabaja en el mismo.

*Figura 1.1 Organigrama del Gimnasio “Natural Body Club”*

## 1.4 ESTADÍSTICAS

Los siguientes cuadros son la muestra del movimiento originado por el Gimnasio durante los meses de mayo, junio del año 2006. En la figura 1.2 se muestra un cuadro de Clientes inscritos a las diferentes disciplinas ofrecidas por el Gimnasio durante el mes de Mayo.

Inscritos por Disciplinas	Total mes de Mayo
Spinning	21
Aparatos	37
Aerobic	23
Sauna	20
Natacion	5
Hap ki do	3

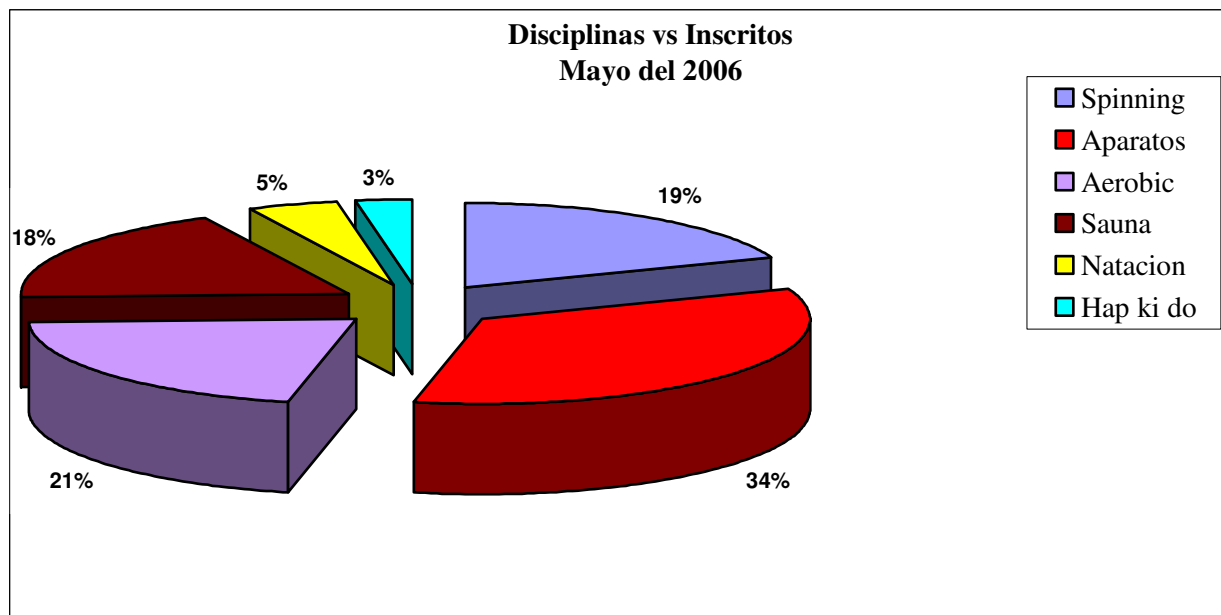
[\*Fuente: Registros Diarios del Gimnasio]

*Figura 1.2 Clientes inscritos a disciplinas en el mes de Mayo*

Como se puede observar, la disciplina que tuvo mayoría de inscritos fue la de **Aparatos**, seguida de la disciplina de **Aerobic** y en tercer lugar la disciplina de **Spinning**.

En términos porcentuales se observa que el 24% de clientes se inscribieron a la disciplina de **Aparatos**, seguida de la disciplina de **Spinning** con el 21%. Véase la figura 1.3





**Figura 1.3 Grafico de Clientes inscritos a Disciplinas en el mes de Mayo**

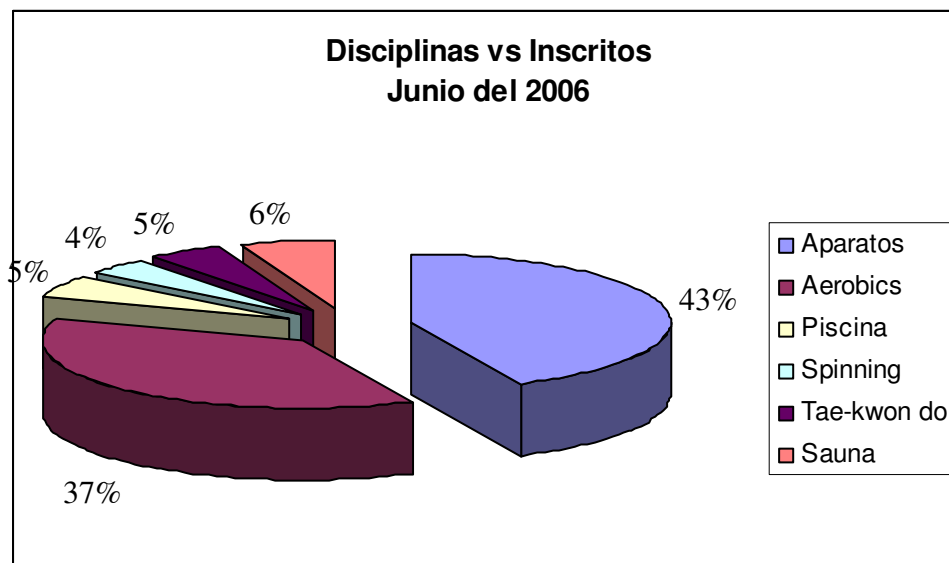
En la figura 1.4 se muestra un cuadro de Clientes inscritos a las diferentes disciplinas ofrecidas por el Gimnasio durante el mes de Junio.

Inscritos por Disciplinas	Total mes de Junio
Aparatos	74
Aerobic	64
Natacion	9
Spinning	7
Sauna	10
Tae-kwon do	8

[\*Fuente: Registros Diarios del Gimnasio]

**Figura 1.4 Clientes inscritos a disciplinas en el mes de Junio**

Durante el mes de Junio se observa que la disciplina de **Aparatos** mantiene el primer lugar, con la diferencia de un incremento del 50% de los inscritos con respecto al pasado mes. En segundo lugar la disciplina de **Aerobic** con 64 inscritos, número que también supera al mes anterior. Véase el gráfico de la figura 1.5 para observar el resto de las disciplinas.



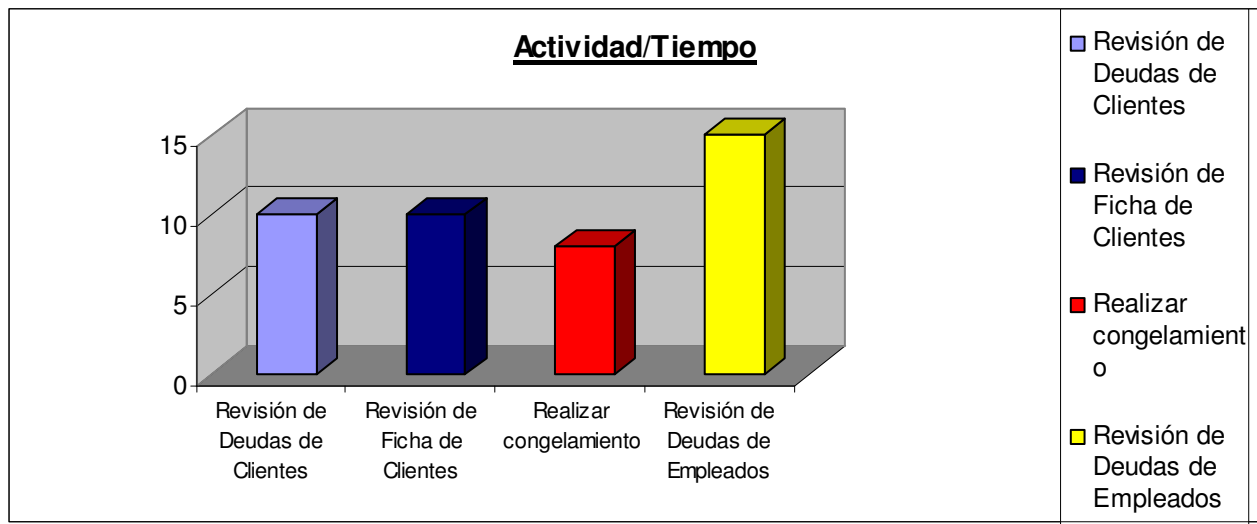
***Figura 1.5 Grafico de Clientes inscritos a disciplinas en el mes de Junio***

Se realizó un cálculo aproximado del tiempo que lleva realizar algunas tareas comunes en el gimnasio. En la figura 1.6, se muestra un cuadro de la relación tiempo-actividad.

Actividad	Tiempo en Minutos
Revisión de Deudas de Clientes	10
Revisión de Ficha de Clientes	5
Realizar congelamiento	8
Revisión de Deudas de Empleados	15
<b>Total Tiempo Consumido</b>	<b>38</b>

***Figura 1.6 Actividad realizada vs Tiempo en Minutos***

Como se observa en el cuadro, sumando el tiempo de todas las actividades da un total de 38 minutos que el personal invierte para poder obtener datos de algunas de las tareas que se realizan en el gimnasio.



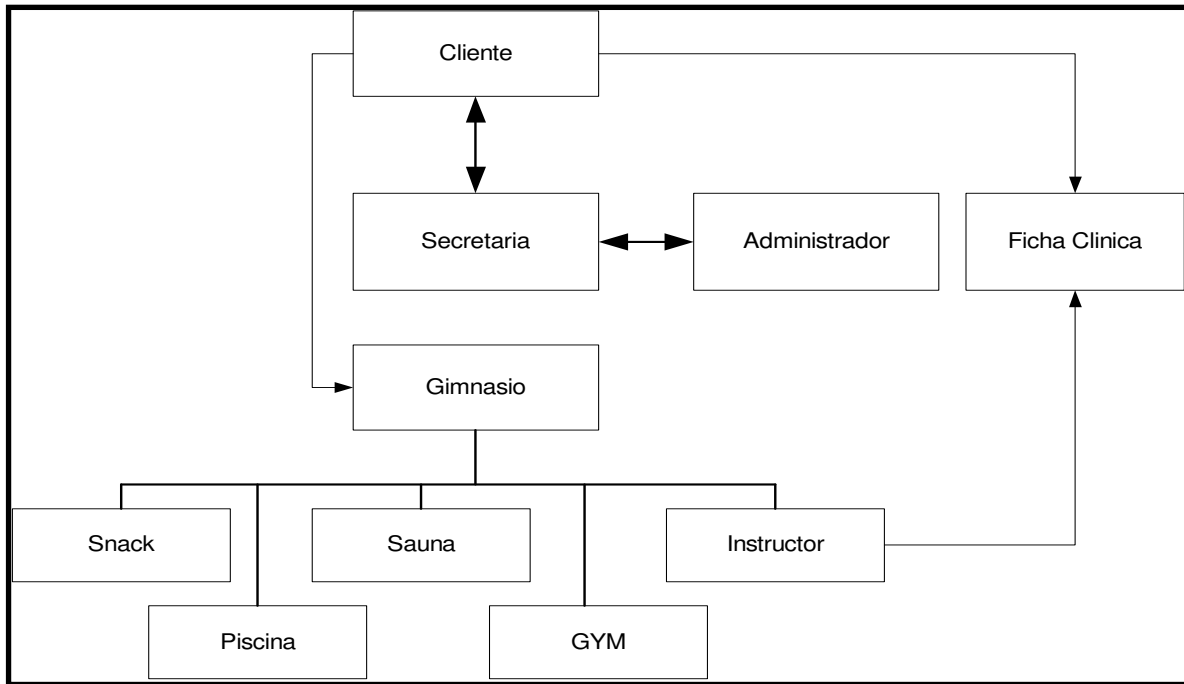
**Figura 1.7 Grafico de Actividad realizada vs Tiempo en Minutos**

En la figura 1.7 se puede observar en un grafico de barras, que la actividad “Revisión de Deudas de Empleados” es la que mayor tiempo consume al personal.

## **1.6 FLUJOGRAMA DE ATENCIÓN EN EL GIMNASIO “NATURAL BODY CLUB”**

En la figura 1.8 se muestra el flujo que sigue un cliente para ser atendido en el Gimnasio.

Un cliente al llegar al Gimnasio se dirige primeramente a la Recepción la cual es atendida por una secretaria que otorga la información sobre las disciplinas y paquetes que ofrece el Gimnasio. El cliente se registra dejando sus datos personales y haciendo efectiva la cancelación de su inscripción.



**Figura 1.8** *Flujograma de Atención al Cliente en el Gimnasio.*

Una vez inscrito el Cliente, éste puede acceder a las diferentes secciones del Gimnasio, según el paquete inscrito:

**GYM:** Lugar donde se encuentran todos los aparatos para el trabajo de musculación. En esta sección el Cliente está dirigido por un instructor. El cual le da las pautas necesarias para cumplir el objetivo fijado por el Cliente.

**Saunas:** Cuartos con temperaturas arriba de los 40 grados, a vapor o seco, con olores de plantas medicinales para realizar la limpieza del organismo y de la piel.

**Snack:** El cliente puede dirigirse al snack cuando él desee servirse una ensalada de frutas, jugos, o algún plato que brinda el snack.

**Piscina.** El Cliente puede dirigirse a la piscina que tiene el Gimnasio para realizar ejercicios de Natación. El Cliente puede acceder a esta sección siempre y cuando haya incluido su paquete de inscripción.

**Instructor.** El Cliente una vez tiene su asistencia al Gimnasio, el instructor tiene el trabajo de orientar y guiar en todas las normas para la realización de los ejercicios como ser el número de sesiones, repeticiones y series. Por otro lado el seguimiento de sus medidas antropométricas y signos vitales cada cierto tiempo del Cliente. Estos datos son registrados en una Ficha Clínica (ver Anexo B: Registros del Gimnasio) la cual es almacenada y consultada para realizar el seguimiento y cumplimiento de sus metas.

## **CAPITULO 2.**

### **OBJETIVOS DEL PROYECTO “GIMNASOFT”**

En el presente capítulo se describen los antecedentes que tiene la institución, seguida de las justificaciones que se dan a los mismos, se establecen el Objetivo General a alcanzar y los Objetivos Específicos que se utilizarán para alcanzar el Objetivo General. Y por último se detalla el alcance del proyecto “GIMNASOFT”.

#### **2.1 ANTECEDENTES Y JUSTIFICACIONES**

A continuación se da a conocer los antecedentes de la institución, es decir, la forma de cómo está operando en la actualidad, posteriormente se justifica por cada antecedente cómo el sistema cubrirá tales funciones.

Existe redundancia al anotar los datos del cliente en el momento de atenderlo; por ejemplo, los datos personales se deben escribir en la ficha técnica, en la agenda, en el libro de casilleros, en las tarjetas de acceso y en los registros de Excel.

Dificultad para el control sobre el movimiento de la parte económica, que incluye los gastos diarios que se generan como el manejo de pagos, ingresos generados por el gimnasio, créditos a los clientes, control de productos en el snack y en el gimnasio.

No existe control de asistencia tanto de los Instructores y Clientes al Gimnasio. Por ejemplo conocer Clientes inscritos en cierto horario, citando el nombre del instructor, puede llevar aproximadamente 3 horas de trabajo.

No existe un apropiado manejo de información del Cliente para realizar un seguimiento adecuado para el cumplimiento de sus metas, como ser control de su peso, de sus medidas antropométricas, su dieta, etc.

Para almacenar todos los documentos que se generan de cierto cliente, existe la necesidad de tener un tablero para las fichas técnicas, en promedio mensual se generan 100 fichas clínicas, además tomando en cuenta que para cada cliente puede llegar a tener más de 2 fichas clínicas dependiendo del tiempo que el cliente lleve en el gimnasio. Por otro lado están los libros en Excel, los cuales pueden ocupar hasta 3 hojas con aproximadamente 500 filas solo por un mes, una agenda para registrar actividades y recordatorios. En otras palabras la información se tiene diversificada lo que la hace difícil para acceder en cualquier momento.

Existen situaciones en que los empleados se retiran del Gimnasio o cumplen su horario de trabajo, y que posterior a ello, se hace complicado luego revisar o continuar su tarea vigente, por el posterior empleado que continuará su trabajo.

“GIMNASOFT” ayudará en el registro de los datos evitando reescribir los mismos, ya que toda la información se manejará de forma integrada, además que se utilizará controles de autocompletar el cual ayudará al escribir el texto completo.

La parte económica se llevará de forma exacta, consistente y sin posibilidad de ser alterada, permitiendo de esta forma un control total de los ingresos, egresos y créditos otorgados por el Gimnasio.

Se permitirá ingresar al Gimnasio previa autenticación en el Sistema, el Sistema realizará una revisión del estado del cliente, avisando si éste está activo o no, tiene cuentas pendientes o está al día en sus cuotas.

La información personal de cada cliente se agrupará en una sola interfaz de usuario y se podrá ver la evolución de sus medidas, dietas y todo a lo que su ficha técnica se refiere.

Con la implementación de “GIMNASOFT” se reducirá el espacio físico de almacenamiento de las fichas técnicas, y otros documentos donde se registra información ya que toda la información se encontrará en una base de datos central de un computador.

Con el sistema no existirá el problema de retiro del personal o cumplimiento del horario de trabajo, ya que toda la información estará disponible en la base de datos y cualquier otro usuario podrá recuperar la información del anterior empleado, y de esta forma conocer alguna situación pendiente.

Además la seguridad que brindará el sistema con respecto a las fichas técnicas, datos económicos, será a través de diferentes niveles de acceso de usuarios. Es decir, cada usuario tendrá su autenticación antes de entrar al sistema. De esta manera el usuario restringido no podrá: Adulterar la ficha clínica, los datos económicos, los horarios y otros. Por otro lado el usuario con los permisos administrativos podrá ingresar a todas las opciones del sistema.

## **2.2 OBJETIVOS DEL PROYECTO**

A continuación se describen el Objetivo General y los Objetivos Específicos a alcanzar en el presente proyecto.



### **2.2.1 Objetivo General**

Desarrollar un Sistema de Información que permita llevar el seguimiento de las actividades físicas de los clientes y control económico del Gimnasio “Natural Body Club”.

### **2.2.2 Objetivos Específicos**

- ✓ Analizar el proceso de atención y seguimiento del cliente y los diferentes documentos que intervienen en el mismo.
- ✓ Aplicar el Proceso Unificado de Desarrollo (PUD), utilizando el Lenguaje Unificado de Modelado (UML) para las diversas etapas de desarrollo del sistema.
- ✓ Diseñar un modelo de base de datos para almacenar los datos referentes al cliente, datos económicos y del personal del gimnasio de manera consistente y además con la facilidad para obtener la información requerida.
- ✓ Implementar el software utilizando la Orientación a Objetos y aplicando el Patrón de Capas. Además de utilizar una interfaz gráfica provista de iconos que identifiquen la tarea a realizar, ventanas que permitan utilizar en un solo ambiente las tareas más importantes, y otros elementos que permitan al personal del Gimnasio utilizarlo sin mayores complicaciones.

## **2.3 ALCANCE DEL PROYECTO**

La realización del proyecto está orientada al Gimnasio “*Natural Body Club*”, ubicado en la Calle Suárez Arana, primer Anillo.

Para la realización del proyecto se utilizará un conjunto de herramientas software y técnicas que permitirán llevar a cabo la realización del Sistema de Información, por ejemplo se utilizará un

manejador de Base de Datos Relacional como el Access 2003, un Lenguaje de Programación como el Visual Basic .Net, Visio 2003 y Enterprise Architect para la realización de Diagramas, Word 2003 para la redacción de manuales y otros.

En el aspecto operacional se incluirán los siguientes módulos principales:

### **MÓDULO DE FICHAS CLÍNICAS**

El sistema tendrá un modulo denominado Ficha del Cliente en el cual se tendrá disponible las siguientes opciones:

*Datos Personales.* Se podrá registrar los datos personales del cliente como: Nombre, apellidos, edad, sexo, número de hijos, correo electrónico, fecha de nacimiento, estado civil, lugar de trabajo, etc.

*Medidas Antropométricas.* Cada determinado tiempo se podrá registrar las medidas antropométricas como ser: Peso, Cuello, Pectoral, Brazo Izquierdo, Cadera, etc.

*Signos Vitales.* Cada determinado tiempo se podrá registrar los signos vitales que incluye: Estatura, Peso Corporal, Ritmo Cardíaco, Presión Arterial, Grasa Corporal, Masa Muscular Magra.

*Graficación.* Permitirá la graficación tanto de las medidas antropométricas como los signos vitales y de esta forma poder comparar el avance o retroceso que se haya tenido. La graficación se podrá realizar de acuerdo a un rango de fechas previamente seleccionado.

*Antecedentes.* Se podrá registrar los antecedentes médicos como ser Patologías, Tratamientos actuales, Medicación, Cirugías, etc.

*Dieta.* Se podrá detallar el tipo de dieta a seguir por el Cliente.

*Buscar Cliente:* Integramente contará con una opción donde se podrá buscar al cliente por el código asignado o por apellido. Y como resultado el sistema mostrará la ficha clínica completa del mismo.

## **MÓDULO DE CAJA.**

Todos los ingresos y egresos generados por el gimnasio como por el snack, son registrados actualmente de forma manual. Ver Anexo B: Registros del Gimnasio

El sistema contará con las siguientes partes en este proceso:

*Apertura de la Caja.* La Apertura de Caja se realizará para empezar a operar el manejo de ingresos y egresos. En esta parte se colocará como saldo inicial el monto con el cual la caja empezará a operar en el día, un detalle y la fecha de inicio.

*Registro de Ingresos-Egresos.* Se podrá registrar los montos ya sean de ingreso o egreso y el sistema aumentará o disminuirá el saldo ingresado inicialmente.

*Cierre de Caja.* El proceso de Cierre de Caja coloca todas las transacciones realizadas en una fecha a un estado de “cerradas”.

## **COMPRAS**

Actualmente el gimnasio cuenta con saunas, piscina, snack y el área del gimnasio, en las cuales se generan compras de productos por cada área. Y no existe un control sobre los productos que tienen en el snack.

El sistema llevará el control y registro de los productos, de todas las áreas del gimnasio, controlando así las cantidades existentes de los productos, y con opción a que sea controlado o no el producto en stock. Por otro lado las compras originadas, serán reflejadas en el movimiento de caja diaria, disminuyendo el saldo de forma automática.

## **VENTAS**

Las ventas al igual que las compras serán registradas tanto para Clientes como para Empleados del Gimnasio. Las ventas podrán ser al crédito o al contado. Cada venta al contado que se registre será colocada como ingreso y sumará al saldo actual en caja. Por otro lado, reducirá la cantidad en stock que se tenga del producto y revisará si no se pasa la cantidad en stock.

El sistema emitirá un mensaje de advertencia si la cantidad de venta excede a la cantidad en stock, no permitirá realizar la venta, hasta que el stock sea reabastecido.

## **CONTROL DE INGRESO DEL CLIENTE**

Actualmente no existe un control de acceso de los clientes al gimnasio. Lo que da motivo que algunos clientes ingresen en horarios no programados. El sistema tendrá una interfaz separada pero integrada a la base de datos para que el cliente al momento de ingresar al Gimnasio pueda ingresar su código de registro. El sistema emitirá una alarma si el cliente tiene cuentas pendientes, o está ingresando en horarios en los cuales no se ha programado, caso contrario registrará la hora y fecha de ingreso.

## **PAGOS A EMPLEADOS**

El modulo de pagos a empleados incluye los pagos a los Instructores, Secretaria y Administrador del Gimnasio. Se podrá realizar los pagos a los empleados cancelando el sueldo completo o dando un adelanto. Por otro lado, el sistema podrá descontar de su sueldo los créditos que el empleado tiene por consumo del snack o por otros consumos generales. Los pagos o adelantos que se den serán reflejados en el movimiento de la caja, realizando el descuento respectivo del saldo actual de caja.

## **DISCIPLINAS**

Permitirá gestionar todas las disciplinas que el gimnasio desee ofrecer a sus clientes, incluyendo los precios de cada una de ellas. De esta forma pueda armar paquetes uniendo varias disciplinas destinados a la promoción de sus clientes.

**EMPLEADOS.**

Permitirá la gestión de los empleados. Permitiendo llevar el control de sus datos personales como ser: Nombre, apellido, sexo, teléfono, dirección, sueldo por mes, y horarios de asistencia al gimnasio.

**PRODUCTOS.**

Permitirá la gestión de los productos, que se vendan tanto en Snack como del propio Gimnasio. Controlar la cantidad disponible en stock por producto. Opcionalmente también se podrá configurar por producto si se desea que el sistema lleve el control de cantidades en stock.

**COBROS A CLIENTES.**

Cuando un cliente desee cancelar las cuentas pendientes, sean deudas por consumo del snack o por inscripción, el sistema tendrá una opción para buscar al cliente que desea pagar, mostrando su total de deudas, montos pagados y su saldo deudor. También el sistema permitirá aumentar el saldo en el flujo de caja, dado que se origina un ingreso en efectivo.

**INGRESOS Y EGRESOS GENERALES.**

El sistema permitirá el registro de ingresos y egresos varios para llevar el control de entradas y salidas de dinero generados por motivos generales como son las llamadas por teléfono, comidas extras, compra de productos de limpieza, ambientadores, etc. Estos ingresos y egresos también serán tomados en cuenta en el flujo de caja.

**ADMINISTRACION DE USUARIOS.**

El sistema permitirá la administración de usuarios para poder controlar el acceso del personal al sistema. El usuario Administrador tendrá todos los permisos necesarios para crear otros usuarios y asignarle los permisos necesarios a los demás módulos del sistema.

### **AUTENTICACIÓN DE USUARIOS.**

Al ingresar al sistema, éste pedirá nombre de Usuario y Contraseña para poder ingresar. El sistema validará el ingreso solo por tres intentos incorrectos que se realicen, después el sistema será abortado.

### **CONFIGURACIÓN DEL SISTEMA.**

La configuración del sistema manejará parámetros generales que son utilizados en todo el entorno y funciones del sistema. Tales parámetros son: Nombre del Gimnasio, Dirección, Teléfono, Administrador, Tipo de Cambio actual, Copias de Seguridad de la base de datos.

### **GENERACIÓN DE REPORTES**

El sistema podrá desplegar reportes tales como:

Ingresos y Egresos generados en un rango de fechas, o por productos o por Cliente.

Listado de las frecuencias de ingreso de clientes al gimnasio.

Lista de las disciplinas que ofrece el Gimnasio.

Listado de Clientes de acuerdo al estado (congelado, activo, inactivo).

Listado de la evolución de las medidas antropométricas por cliente.

Listado de Dietas por Cliente.

Listado de Clientes con cuentas pendientes de pago.

Listado de disponibilidad en almacén de los productos.

Listado de Instructores y sus horarios asignados.

Todos los reportes tendrán opción de poder ser convertidos a los siguientes formatos: Excel (xls), Word (.doc) o formato Pdf. Esto con el fin de facilitar la entrega de informes al Propietario del Gimnasio y además de permanecer en contacto con sus clientes enviándoles lista de precios, recordatorios de deudas u otros a través de su propio correo electrónico.

## **PARTE II**

## **MARCO TEÓRICO**

## CAPITULO 3.

# CONCEPTOS Y TERMINOLOGÍAS UTILIZADAS EN EL GIMNASIO “NATURAL BODY CLUB”

En el presente capítulo, se definen todos los términos utilizados en el Gimnasio “Natural Body Club”, además de definir también Conceptos implícitos que están incluidos en la realización de la actividad física.

### 3.1 EL EJERCICIO.

Los objetivos del ejercicio son mejorar el consumo de oxígeno y los procesos metabólicos, conseguir fuerza y entrenamiento, disminuir la grasa corporal, y mejorar el movimiento de las articulaciones y los músculos. Todos estos beneficios son esenciales para la buena salud, y se debería intentar incorporar una rutina de ejercicio a nuestra vida diaria. La Asociación Americana de Cardiología recomienda que las personas hagan ejercicio al menos 30 minutos tres o cuatro veces por semana, y 10-24 minutos más de 5 veces por semana, y se consigue estar más cerca del nivel óptimo con cada sesión.<sup>1</sup>

Los expertos dividen el ejercicio en tres **categorías generales**: aeróbico, fuerza y flexibilidad, y recomiendan un programa equilibrado que incluya las tres.

---

<sup>1</sup> [www.well-connected.com](http://www.well-connected.com)



### **3.1.1 Entrenamiento aeróbico (de resistencia)**

Los ejercicios aeróbicos crean resistencia y mantienen el bombeo cardiaco a un ritmo constante pero elevado durante un periodo prolongado. Practicarlos regularmente puede potenciar la función cardiaca, aumentar las HDL (el colesterol "bueno"), fortalecer los huesos de la columna vertebral, y disminuir el riesgo de infarto, hipertensión arterial, accidente vascular cerebral, diabetes, e incluso algunas formas de cáncer. El ejercicio aeróbico también reduce la grasa corporal, y puede mejorar el bienestar de la persona. Correr, nadar, montar en bicicleta, subir escaleras, y el aeróbic son todos ellos ejemplos de ejercicio aeróbico.

### **3.1.2 Entrenamiento isométrico (de fuerza o resistencia)**

Mientras que el ejercicio aeróbico se centra en la resistencia, el ejercicio isométrico se basa en la fuerza. La adición de 10-20 minutos de un ligero entrenamiento de fuerza dos o tres veces por semana es importante para un programa de ejercicio equilibrado. Las personas que sólo hacen ejercicio aeróbico suelen perder fuerza en la parte superior del cuerpo. El ejercicio isométrico aumenta la fuerza muscular y quema la grasa, ayuda a mantener la densidad ósea, y mejora la digestión. Parece disminuir las LDL (el llamado "colesterol malo"). El ejercicio isométrico es beneficioso para todo el mundo, incluso para personas de 90 años. De hecho, el entrenamiento muscular es más importante a medida que se envejece, ya que tras los 30 años todos sufrimos un lento proceso de erosión muscular, que puede reducirse o incluso revertir añadiendo un entrenamiento de resistencia al programa de ejercicio.

### **3.1.3 Entrenamiento de flexibilidad (estiramientos)**

El entrenamiento de flexibilidad utiliza los ejercicios de estiramiento para prevenir los calambres, la rigidez y las lesiones. Asegura asimismo un amplio rango de movilidad. El yoga y el tai-chi, que se centran en la flexibilidad, el equilibrio y el control de la respiración, pueden incluso disminuir el estrés y ayudar a bajar la presión arterial y pueden incluso tener efectos beneficiosos sobre los niveles de colesterol. Se recomienda realizar ejercicios de estiramiento durante 10-12 minutos al menos tres veces por semana. Mientras se hacen, hay que extender los músculos hasta el punto de máxima tensión sin dolor, y mantenerlos ahí durante 20-60 segundos (los principiantes pueden necesitar comenzar con 5-10 segundos). Se debe recordar respirar

constantemente mientras se mantenga el estiramiento, e inspirar cuando se vuelva a la posición de relajación.

### **3.2 SIGNOS VITALES**

Se denominan Signos Vitales, a las señales o reacciones que presenta un ser humano con vida que revelan las funciones básicas del organismo. Los Signos Vitales son: Respiración, Pulso, Reflejo Pupilar, Temperatura y Presión Arterial. Al prestar atención a un cliente es importante valorar el funcionamiento del organismo y detectar las alteraciones que son frecuentes en caso de accidentes; para ello es necesario controlar la respiración y el pulso las cuales orientan al Instructor para iniciar los debidos ejercicios.

Cifras normales de respiración son:

Niños de meses: 30 a 40 respiraciones por minuto

Niños hasta seis años: 26 a 30 respiraciones por minuto

Adultos: 16 a 20 respiraciones por minuto

Ancianos: menos de 16 respiraciones<sup>2</sup>

### **3.3 INSTRUCTOR**

El Instructor es la persona encargada de asignar, controlar y realizar el seguimiento de la dosificación de ejercicios y dietas al Cliente.

### **3.4 CASILLERO**

Son casillas utilizadas para guardar la indumentaria de los clientes cuando ingresan al Gimnasio. Cada casillero tiene su llave de seguridad.

### **3.5 ESTADOS DEL CLIENTE**

Se identifican tres estados por las cuales un cliente puede estar.

El Cliente Activo, es aquel que está inscrito en una de las disciplinas y asiste en su horario definido.

---

<sup>2</sup> [JAC01], Jacobs, Fernando “Soporte Básico de Vida”

El Cliente Congelado es aquel Cliente que está inscrito en el Gimnasio, pero que por algún motivo puede suspender su asistencia al Gimnasio. Esta suspensión puede ser hasta cinco días como máximo

El Cliente Pasivo, es aquel que está registrado en el Gimnasio, pero que ya terminó sus actividades en el Gimnasio, y por tanto ya no asiste.

### **3.6 FICHA TÉCNICA.**

La Ficha Técnica es una hoja donde registran los Instructores los Signos Vitales y Medidas Antropométricas de cada cliente.

### **3.7 MEDIDA ANTROPOMÉTRICAS**

Se debe recordar antes de comenzar la descripción de las medidas antropométricas, la posición que debe adoptar el cliente para la mayoría de las mediciones. Debe permanecer el sujeto de pie, con la cabeza y los ojos dirigidos hacia el infinito, las extremidades superiores relajadas a lo largo del cuerpo con los dedos extendidos, apoyando el peso del cuerpo por igual en ambas piernas, los pies con los talones juntos formando un ángulo de 45°. Esta posición, es llamada en Cineantropometría, la "posición de atención antropométrica" o "posición estándar erecta". Esta posición es más cómoda para el estudiado, diferenciándose con respecto a la posición anatómica en la orientación de las manos.<sup>3</sup>

Las técnicas de medición de las medidas referidas en la ficha cineantropométrica son las siguientes:

**3.7.1 Peso:** Mide la masa corporal total de un individuo. El mismo debe tomarse con la menor cantidad de ropa posible. Cuando el peso no se toma con la cantidad mínima de ropa se puede estar sobre-estimando (pesando más de lo real) a un individuo; dando lugar a una evaluación inadecuada del cliente.

---

<sup>3</sup> <http://www.ugr.es>

**3.7.2 Talla:** Mide el tamaño del individuo desde la coronilla de la cabeza hasta los pies (talones). Es la distancia entre el vértex y las plantas de los pies del estudiado en cm.

**3.7.3 Talla Sentado:** Distancia entre el vértex y el plano de sustentación del estudiado medida en cm.

### **3.8 PAQUETES**

La terminología paquete se refiere a una o más disciplinas que están disponibles para que los Clientes puedan inscribir. Cada paquete lleva un precio definido por el Administrador y la cantidad de personas a las que está dirigido.

### **3.9 LOS ANTECEDENTES**

Son los problemas de salud anteriores que tuvo el paciente, los mismos serán registrados por el Instructor para considerar la dosificación de ejercicios y dieta ideal

### **3.10 EL CALENTAMIENTO**

Hay diferentes opciones de calentamiento. El Cliente debe elegir una alternativa cada día.

Opciones:

Bicicleta fija; 5 minutos

Caminar; 8-10 minutos

Mover cada articulación en todas direcciones; comenzando por la cabeza y terminando con los pies:

**Cabeza:** Realizar círculos, suavemente. Mover hacia atrás lentamente; mantener boca cerrada.

Doblar cuello de un lado para otro, manteniendo la vista fija delante y la nariz hacia delante.

**Hombros:** Hacer círculos hacia atrás; de a uno o los dos juntos.

**Codos:** Dedos en los hombros. Hacer círculos hacia atrás.

**Brazos:** Abiertos y estirados hacer círculos hacia atrás, círculos pequeños y luego cada vez más grandes.

**Muñecas:** Mantener brazos extendidos hacer círculos con las manos.

**Tronco:** Mantener brazos extendidos a nivel de los hombros y doblar el tronco de un lado para otro. No mover los pies.

**Pelvis:** Elevar brazos para balance o dejarlas en las caderas. Hacer movimiento de caderas como la danza del vientre. Realizar movimientos circulares. Levantar una cadera y luego la otra, hacia los hombros.

**Caderas:** Balancear 1 pierna, luego la otra, de lado a lado. Después, de adelante hacia atrás. Doblar rodillas y hacer número ocho.

**Piernas:** Con una pierna estirada balancearla delante y detrás de la otra.

### **3.11 SOBREENTRENAMIENTO**

Dolores musculares, cansancio, dificultad para dormir y alteraciones del apetito, son sólo algunos de los efectos visibles del exceso de ejercicio.

Las causas más frecuentes de este trastorno pueden resumirse en las siguientes: prácticas intensas y continuadas durante largos períodos de tiempo. Aumento rápido en la intensidad del entrenamiento. Tiempos de recuperación insuficientes. Presencia de ansiedad, depresión, estrés o dietas estrictas.

Y si bien muchos de sus efectos son visibles a simple vista, también existe una serie de complicaciones que repercuten a nivel interno en el organismo:

*A nivel muscular:* presencia de dolor muscular y articular, que no desaparece después del descanso habitual. También hay un aumento de las lesiones.

*A nivel cardiovascular:* aumento de la frecuencia cardiaca y de la tensión arterial en reposo.

*A nivel sanguíneo:* disminución del número de linfocitos y gammaglobulinas, lo que produce una inmunodepresión del organismo, con el posible aumento del contagio de enfermedades infecciosas. También disminuyen las vitaminas C y B, aumentan los niveles de ácido láctico (lo

que supone la aparición de fatiga precoz durante el ejercicio), bajan los niveles de testosterona, y se reducen también las proteínas totales.

*A nivel metabólico:* disminución de la eficiencia energética y pérdida de peso (que en ocasiones se manifiesta como anorexia).

*A nivel neurológico:* sensación de agotamiento y cansancio, alteraciones del ritmo del sueño (insomnio) y disminución de la destreza muscular.

*A nivel psicológico:* alteraciones del carácter (irritabilidad y agresividad) y disminución del rendimiento intelectual (falta de atención y concentración).

### **3.12 NORMAS PARA REALIZAR EJERCICIOS.**

Para la realización de los ejercicios se tienen que cumplir ciertas normas, las cuales son la realización de **Series, Sesiones, Repeticiones y Descansos**.<sup>4</sup>

Se entiende por **repeticiones** el número de veces seguidas que repite un mismo movimiento. Las **series** (o tandas) agrupan cierto número de repeticiones, separadas por periodos de descanso o por la realización de ejercicios distintos. Una **sesión** corresponde al conjunto de series -habitualmente de distintos movimientos- que se realizan de forma seguida. Por ejemplo, una sesión puede agrupar 6 ejercicios distintos, de cada uno de los cuales haga 4 series con 15 repeticiones en cada una de ellas.

El número de repeticiones y series, así como el período de descanso entre ellas pueden variar en función de los objetivos específicos que persiga el programa que el Instructor ha diseñado para un caso específico. Sin embargo, y salvo que el Instructor haya observado que tiene una base muscular previamente buena y que es necesario aumentar la potencia de su musculatura antes que su resistencia, en general le recomendará:

---

<sup>4</sup> <http://www.espalda.org>

**Hacer 3 series de cada tipo de ejercicio** con el número de repeticiones que pueda en cada una de ellas. Hacer menos de 2 series es prácticamente inútil. Algunos programas de ejercicios requieren hacer 4 series, en vez de 3. A medida que le vaya siendo más fácil completar las series, aumente el número de repeticiones en cada una de ellas, pero siga haciendo el mismo número de series.

**Descansar entre una serie y otra, lo mismo que tarde en hacer una.** Así por ejemplo, si tarda 30 segundos en hacer una primera serie, no debe descansar más de 30 segundos antes de hacer la segunda. Es posible que en la última serie la musculatura esté cansada y no pueda hacer el mismo número de repeticiones que en las demás. Es mejor hacer menos repeticiones en la última serie que prolongar el tiempo de descanso entre ellas. Aumente el tiempo de descanso entre series sólo si así se lo indica el Instructor por el tipo de programa que específicamente le haya prescrito a usted. Por ejemplo, en programas destinados a aumentar la potencia de la musculatura, en la que se hacen series de ejercicios muy intensos con pocas repeticiones, el tiempo de descanso puede aumentar hasta el doble del que tarde en hacer la serie.

### 3.13 LAS VITAMINAS

Las vitaminas y minerales son los obreros del cuerpo que participan en el buen funcionamiento de los músculos y las hormonas y en la transformación de los alimentos en la energía que se requiere a diarios. En la figura 3.1 aparecen los alimentos que proporcionan las vitaminas naturales.<sup>5</sup>

<b>LO QUE SE DEBE COMER PARA QUE NO FALTEN VITAMINAS.</b>	
Vitamina A	Se halla en la yema del huevo, mantequilla, nata de la leche, maíz o choclo, espinacas, semillas de cereales, albaricoque, brócoli, zanahorias, perejil, aceite de hígado de bacalao.
Vitamina B1	Leche, huevos, carne de cerdo, carne de aves, legumbres, soya, semilla de cereales, pan integral, perejil, nueces, ajos, levadura.
Vitamina B2	Leche, huevos, carne de bovino y de aves, hígado, semillas de cereales, frutos

<sup>5</sup> [FRA01], Fraga, Anelise “La Belleza al Alcance de Todos”

	secos, soya, perejil.
Vitamina B3	Huevos, hígado, hortalizas verdes, legumbres, productos integrales, levadura.
Vitamina B6	Leche, yema de huevo, pescado, verduras, productos integrales, ajo.
Vitamina B10	Carne, leche, queso, hortalizas verdes, zanahoria, levadura.
Vitamina C	Frutas cítricas, papaya, rábano, perejil, ensaladas, berros, uvas.
Vitamina D	Hígado, leche, mantequilla, pescado, aceite de hígado de bacalao, aceites vegetales, verduras y frutas amarillas.
Vitamina E	Yema de huevo, leche, pescado, soya, hortalizas verdes, aceites vegetales.

**Figura 3.1: Vitaminas en Alimentos**

### 3.14 PESO IDEAL

Ciertos especialistas definen el Peso Ideal como *"aquel con el cual un individuo se encuentra a gusto, permitiendo que se desarrollen normalmente todas las funciones biológicas"*.

Cabe señalar que con la misma altura, ciertas personas se mantienen espontáneamente bien a 60 kg, mientras que otras lo hacen a 50 ó 55 Kg., por ejemplo. Este concepto se fundamenta en ciertas bases fisiológicas: todos los individuos son diferentes; hay diferencias en la relación masa muscular/masa grasa, secreciones hormonales, inervación muscular, etc. La altura, entonces, no es más que uno de los tantos parámetros para estimar el peso teórico óptimo.

También se debe pensar que el hombre tiene mayor desarrollo de masa ósea y muscular, y el músculo es mucho más pesado que el tejido graso. Es por ello que se han diseñado tablas de pesos teóricos para hombres y tablas para mujeres.

Altura (mts)	Sexo masculino	Sexo femenino	Altura (mts)	Sexo masculino	Sexo femenino
1,50	-	44,5-50kg	1,72	63,1-68,3	57,5-64,3
1,51	-	45,1-50,5	1,73	63,8-69,1	58,3-65,1
1,52	-	45,6-51	1,74	64,5-69,8	59-65,8
1,53	-	46,1-54,4	1,75	65,2-70,6	59,7-66,5
1,54	-	46,7-52,1	1,76	65,6-71,3	60,4-67,2
1,55	-	47,2-52,6	1,77	66,7-72	61,1-67,8



1,56	-	47,7-53,2	1,78	67,4-72,8	61,7-68,6
1,57	54,3-58,1kg	48,2-53,7	1,79	68,1-73,6	62,5-69,3
1,58	54,8-58,9	48,6-54,3	1,80	68,8-74,4	63,3-70,1
1,59	53,3-59,6	49,3-54,8	1,81	69,5-75,4	64-70,7
1,60	55,8-60,3	49,9-55,3	1,82	70,2-76,3	64,7-71,5
1,61	56,4-60,9	50,4-55,4	1,83	70,9-77,2	65,4-72,2
1,62	56,9-61,4	51-56,6	1,84	71,7-78,1	66,1-72,9
1,63	57,5-61,8	51,5-57,5	1,85	72,4-79	66,6-73,6
1,64	58-62,5	52-58,2	1,86	73,1-79,7	-
1,65	58,6-63	52,6-58,9	1,87	73,8-80,8	-
1,66	59,1-63,7	53,3-59,6	1,88	74,5-81,7	-
1,67	59,6-64,4	54-60,7	1,89	75,4-82,6	-
1,68	60,2-65,1	54,7-61,5	1,90	76,3-83,5	-
1,69	60,9-65,6	55,4-62,2	1,91	77,2-84,4	-
1,70	61,7-66,6	56,1-62,8	1,92	78,1-85,3	-
1,71	62,4-67,4	56,8-63,6	1,93	79-86	-

**Figura 3.2 Peso Ideal según Altura.**

### 3.15 COMO MEDIR NUESTRA APTITUD FISICA

Para saber si se puede hacer gimnasia se debe medir el pulso en la muñeca en la arteria radial durante 10 segundos y multiplicarlo por 6, así obtendremos los latidos del corazón durante un minuto.

Se lo puede medir en reposo al despertar, sabiendo que a menor frecuencia mayor aptitud física, en los hombres la frecuencia apta sería de 60 a 90 latidos por minuto, y en la mujeres de 70 a 100, no apto sería a partir de 90 en hombres y de 100 en mujeres.

También se puede medir después de hacer ejercicio subiendo y bajando un escalón durante 3 minutos, descansar 30 segundos y tomar el pulso, **seremos aptos para el ejercicio si la frecuencia en hombres es de 77 a 100 y en mujeres de 86 a 110**, si está más alto no es aconsejable y quizá se deba comenzar preparándose poco a poco.

### 3.16 BONDADES QUE APORTA EL EJERCICIO FÍSICO

- Devuelve la línea si hay exceso de peso
- Disminuye el colesterol, el riesgo de infarto, baja la tensión si está alta
- Es tan eficaz como la psicoterapia en el tratamiento de la depresión
- Estimula la liberación de endorfinas, hormonas internas que producen sensaciones de placer y bienestar
- Distrae y evade de las preocupaciones.

*“Hipócrates decía: El ejercicio físico es imprescindible porque disipa toda clase de venenos derivados de una mala dieta.”*

### 3.17 EJERCICIOS RESPIRATORIOS

Los ejercicios respiratorios tienen como finalidad:

- ☐ Enseñar al cliente a respirar de forma que aproveche al máximo la capacidad de sus pulmones.
- ☐ Disminuir el atrapamiento aéreo y para descender los niveles de CO<sub>2</sub> en sangre
- ☐ Enseñar al cliente a disminuir la frecuencia respiratoria y a respirar despacio y rítmicamente.

#### 3.17.2 Preparación del cliente.

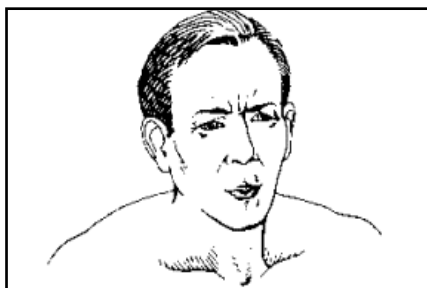
- ☐ Informar al cliente sobre los ejercicios que se van a realizar.
- ☐ Estimular al cliente para que colabore en la ejecución de los ejercicios.
- ☐ La posición del cliente dependerá del tipo de ejercicio a realizar.

#### 3.17.3 Tipo de ejercicios.

**i. Aumentar la relación inspiración espiración.** De forma que la espiración sea el doble de la inspiración. *Ejemplo:* Contar hasta cinco en la inspiración y hasta diez en la espiración.

**ii. La respiración con los labios fruncidos.** Este tipo de respiración prolonga la exhalación, esto puede ayudar a prevenir el colapso bronquiolar y el atrapamiento de aire.

**Ejemplo:** El cliente, realiza una inspiración lenta por la nariz y espira luego, mas lentamente y prolongando la espiración a través de los labios fruncidos.



**Figura 3.3. Respiración con los labios fruncidos**

### **3.18 LA CONCENTRACION**

Para el logro de las aspiraciones, para el buen éxito en todo lo que se persigue, el hombre necesita de algo vital en todo momento y lugar: **la concentración de propósito, de pensamiento, de sentimiento y acción.**

Mediante la concentración en la actividad física, la mente y el cuerpo aprenden a actuar juntos, sin malgastar energía física o mental. Este es el estado ideal que permite al deportista lograr con éxito sus metas propuestas.

Con una práctica perseverante se aprende a dominar la mente de modo de poder mantenerla fija algún tiempo, en una misma línea de pensamiento. Fijar la mente en un solo punto, sin vagar y sin entregarse a distracciones que causan los objetos externos, la actividad de los sentidos o la mente misma; esto es concentración.

## **CAPITULO 4.**

### **CONCEPTOS DEL FLUJO DE CAJA**

En el presente capítulo se explica el concepto del Flujo de Caja, también se definen los movimientos económicos que se son generados en el Gimnasio como ser las Ventas, Créditos, Compras, etc.

#### **4.1 FLUJO DE CAJA**

El concepto de flujo de caja se refiere al análisis de las entradas y salidas de dinero que se producen (en una empresa, en un producto financiero, etc.), y tiene en cuenta el importe de esos movimientos, y también el momento en el que se producen. Estas van a ser las dos variables principales que van a determinar si una inversión es interesante o no.

Por ejemplo, un depósito bancario. En el momento inicial, se produce (desde nuestra perspectiva) un flujo de caja negativo (el dinero que sale de nuestro bolsillo para ir al banco). Cada año de los que dure el depósito, se cobrará unas cantidades en concepto de intereses. Esos son flujos de caja positivos, que se producen en los distintos momentos de tiempo. Al finalizar el periodo del depósito, el banco devuelve el dinero que se había invertido, por lo que se produce un nuevo flujo positivo de caja.

## **4.2 CUENTAS QUE AUMENTAN EL FLUJO DE CAJA**

En esta sección se explica, todas las cuentas que hacen que se originen los Ingresos, que pueden ser como consecuencia de la venta de mercaderías, prestación de servicios y por los réditos de las inversiones efectuadas de otras empresas o instituciones financieras (Bancos).

### **4.2.1 Ventas**

El término Venta según la Real Academia Española lo define como: “Contrato en virtud del cual se transfiere a dominio ajeno una cosa propia por el precio pactado.” <sup>1</sup>

Desde el punto de vista contable las ventas representan los ingresos por la venta de mercaderías, servicios o productos manufacturados por las empresas (industria).

### **4.2.2 Condiciones de Ventas al Crédito**

En la mayoría de los negocios, las condiciones de venta a crédito estipulan el pago dentro de los 30, 60 y 90 días plazo antes de otorgar créditos los negocios investigan la reputación de crédito de cada cliente potencial, pero independientemente de lo cuidadoso que sea un negocio en la extensión de créditos, siempre existe la susceptibilidad de cuentas de dudosa cobrabilidad.

Un modo de determinar la eficiencia con la cual se cobran las cuentas es por su movimiento. El movimiento de las cuentas a cobrar se calcula dividiendo las ventas netas a crédito del año por el promedio de Cuentas a Cobrar, si se dispone de la Posición Financiera mensual se debe utilizar para el computo del promedio, caso contrario se suele utilizar el Balance Inicial y Final del año.

---

<sup>1</sup> <http://www.rae.es>

### **4.2.3 Ingresos No Ordinarios**

Son ingresos provenientes de operaciones ocasionales y que no son consecuencia del giro habitual de la empresa, tales como:

- a) Venta de desperdicios.
- b) Comisiones percibidas
- c) Ganancia en venta de bienes de uso.
- d) Resultados por Exposición a la Inflación (Saldo Acreedor).
- e) Descuento y bonificación sobre compras.
- f) Intereses percibidos.
- g) Diferencia de cambio (saldo acreedor)

### **4.2.4 Cuentas por Cobrar**

Atendiendo a su origen, las cuentas por cobrar pueden ser clasificadas en: **Provenientes de ventas de bienes o servicios y No provenientes de venta de bienes o servicios.**

El primer grupo de Cuentas por Cobrar está formado por aquellas cuyo origen es la venta a crédito de bienes o servicios y que, generalmente están respaldadas por la aceptación de una "factura" por parte del cliente.

El segundo grupo de Cuentas por Cobrar (No Provenientes de Ventas a Crédito) se refiere a derechos por cobrar que la empresa posee originados por transacciones diferentes a ventas de bienes y servicios a crédito como ser: Alquileres por cobrar, Cuentas por cobrar a los accionistas, Intereses por cobrar

## **4.3 CUENTAS QUE DISMINUYEN EL FLUJO DE CAJA**

Los egresos, representan el costo que origina el uso de las cosas o la compra de servicios con la finalidad de generar los ingresos y mantener en marcha la empresa.

### **4.3.1 Sueldos y Salarios**

Su saldo es acreedor y representa los sueldos y salarios pendientes de pago. Su movimiento es el siguiente: Se acredita por el liquido pagable de la planilla de haberes y se debita por todo parcial o total.

### **4.3.2 Aseo y Limpieza**

Corresponde todos los gastos erogados o cargables por este concepto como ser materiales de limpieza y/o remuneración al personal independiente que presta este servicio.

### **4.3.3 Descuento sobre Ventas**

Existen dos tipos de descuentos: Descuento comercial y descuento por pronto pago.

**Descuento Comercial**, son las deducciones que el vendedor realiza al precio de lista por motivos principalmente comerciales (Cantidad, volumen de Ventas y Pagos al Contado) con el fin de beneficiar al cliente. Usualmente este tipo de descuentos no es motivo de registro ya que el precio de factura tanto en compras y ventas de bienes son registrado al precio neto pagado o recibido por la transacción.

**Descuento por pronto pago**, es aquel descuento que se concede o se obtiene (según se trate de ventas o compras) por el pago realizado con anterioridad del vencimiento convenido. Este tipo de descuentos se deben contabilizar utilizando la cuenta “Descuento y bonificación sobre Ventas o Compras” o “Descuento por Pronto Pago”, ya que se trata de descuento no incluido en la factura, y en la operación original no se conoce si aprovechará la oferta del descuento por pronto pago.

### **4.3.4 Gastos de Comunicación**

Son los gastos por concepto de: Uso de teléfono, conferencias, telex, fax, franqueo de correspondencia, compra de periódicos, etc.

### **4.3.5 Gastos Generales**

Se consideran en este rubro aquellos gastos de tipo general, que no son factibles de identificar con los específicamente descritos anteriormente.

### **4.3.6 Compras**

Es la adquisición de bienes y/o servicios para su posterior comercialización, en tanto correspondan a las actividades comerciales habituales de la empresa.

## **4.4 CONTROL INTERNO**

Un gran número de la totalidad de las transacciones de una empresa originan entradas y salidas de dinero, de modo que, todo negocio debe establecer medidas de seguridad para controlar el efectivo, pues éste es el valor de activo más susceptible al robo que cualquier otro. Por otro lado se sugiere: Cuatro principios de control interno.<sup>2</sup>

**a) Crear documentación fuente.**

Es muy importante crear documentación fuente tales como: Recibos de Caja ingreso, egreso, tiras de papel de cajas registradoras, volantes de remesas recibidas, etc. A objeto de dar fe de la recepción y entrega de fondos.

**b) Preparación de cortes de caja.**

Es necesario efectuar cortes de Caja a objeto de verificar con alguna frecuencia la existencia real del efectivo, como el manejo correcto por el encargado (Tesorero)

**c) Dividir la responsabilidad.**

Es muy importante separar la función del manejo material de los fondos de la función del registro de ingresos y egresos de Caja.

**d) Depositar íntegros los ingresos de caja.**

A objeto de lograr un mayor control interno de Caja deben depositarse diariamente en cuenta corriente bancaria respaldados con papeletas de depósitos y éstas deben igualar con los ingresos del día en Caja.

---

<sup>2</sup> [ORE98], Funes Orellana, Juan “Contabilidad Intermedia”



## **CAPITULO 5.**

### **ADMINISTRACION DE GIMNASIOS**

En el presente capítulo se describe la teoría básica sobre la Administración de Gimnasios como ser: puntos a tomar en cuenta para establecer Normas, Plan de Mantenimiento de los Equipos y descripción de algunas Secciones que comprende un Gimnasio.

#### **5.1 ADMINISTRACIÓN DE GIMNASIOS (INSTALACIONES DEPORTIVAS)**

El deporte ha pasado de ser una actividad eminentemente social, a una actividad económica. Por lo tanto, la búsqueda de formas de gestión más eficaces en las instalaciones deportivas debe ser una prioridad. Cada vez se hace más necesario el desarrollo de planes de gestión para cada una de las instalaciones deportivas.

El responsable de la instalación ha de elaborar o revisar anualmente el plan de gestión, documento básico en la gestión de las instalaciones deportivas, que incluye aspectos de organización interna, de programación de actividades y de presupuesto. Este plan debe contener dos documentos importantes, como son el plan de utilización que recoge la distribución horaria de los espacios deportivos y vestuarios con la previsión de usuarios, y el plan de mantenimiento

que recoge todos los trabajos de mantenimiento preventivo, así como una previsión de posibles tareas correctivas a efectuar durante el año y la previsión de consumos, en función del estado de las instalaciones, horas de funcionamiento, uso previsto, etc. Así pues, se deberán cuantificar los consumos eléctricos, combustibles, agua, teléfono. También se tendrán que contemplar otros productos necesarios, como pueden ser los productos de uso diario de limpieza, cloro, papel higiénico, etc.

La **disponibilidad horaria** es uno de los factores indicativos de la rentabilidad social de una instalación deportiva. En algunos lugares, las instalaciones deportivas llegan a estar cerradas en horario matinal. En otros casos, el nivel de ocupación durante estas horas no alcanza el 20% de sus posibilidades. Si se tiene en cuenta que el coste principal de las instalaciones es fijo, aumentar la disponibilidad horaria no debería causar mayores problemas.

El argumento más extendido para justificar el cierre es la baja utilización en estas horas; por tanto, cabría ofrecer nuevas actividades y facilitar el acceso a las instalaciones a otros colectivos que tengan mayor disponibilidad en estas franjas horarias. Estos colectivos son básicamente, los escolares (clases de Educación Física), así como las personas adultas y tercera edad, que podrían utilizar la instalación en programas de actividad físico-recreativa adecuados a sus necesidades.

Los gestores de las instalaciones deben conocer la dinámica de las distintas prácticas deportivas, estar al corriente de las tendencias actuales de manera que el servicio ofrecido se dirija a la mayor diversidad posible de colectivos, ofreciendo aquellas actividades que en cada momento la sociedad demande, sin dejar de ofrecer las actividades más tradicionales que tienen la aceptación de los usuarios.<sup>1</sup>

---

<sup>1</sup> [FAN97], “La evaluación de la gestión de un centro deportivo”

### **5.1.1 Otros servicios complementarios**

Las Instalaciones Deportivas deben contar con una serie de servicios complementarios básicos que mejoren el servicio prestado a los usuarios en los espacios deportivos: vestuarios, duchas, aseos, botiquín de primeros auxilios...

### **5.1.2 Plan general de funcionamiento**

Los responsables de la gestión deportiva deben realizar un seguimiento continuado y un control de uso exhaustivo acerca de las actividades que se desarrollan en las Instalaciones Deportivas. En función de los datos obtenidos se podrán tomar unas decisiones referentes a la planificación de actividades y los recursos humanos que contarán con mayores posibilidades de éxito que si se tomaran de una forma arbitraria. Así pues, conocer el número usuarios y de servicios deportivos producidos en cada espacio, actividad y franja horaria (curva de producción de servicios deportivos), facilitará la planificación del personal necesario para cubrir la demanda, así como las posibles modificaciones en la estructuración de las actividades.<sup>2</sup>

### **5.1.3 Redacción de las normas y los reglamentos de uso**

El reglamento de uso de las Instalaciones Deportivas debe contener generalmente, los siguientes apartados: ámbito de aplicación, datos técnicos y características de la instalación, sistemas y modalidades de acceso, calendario de apertura, horarios de apertura y cierre, importe de las tasas y precios públicos, formas de pago, normas de uso, reglamento sancionador, disponibilidad de hoja de reclamaciones, etc.

### **5.1.4 Plan de Mantenimiento**

La gestión del mantenimiento tiene como principal objetivo conservar las instalaciones y el material deportivo en condiciones óptimas de utilización el máximo tiempo posible a lo largo de su período de vida útil.

#### **a) Mantenimiento preventivo.**

Se conoce como mantenimiento preventivo la programación de inspecciones, tanto de funcionamiento como de seguridad, ajustes, reparaciones (soldaduras, sustitución de piezas),

---

<sup>2</sup> [FAN97], “La funcionalidad y el coste de los equipamientos deportivos.

análisis higiénico-sanitarios, limpieza, engrasado de partes articuladas (puertas y ventanas, ruedas, postes telescópicos), calibración, que deben llevarse a cabo en forma periódica en base a un plan establecido y no a una demanda del operario o usuario; también es conocido como *mantenimiento preventivo planificado*. Su propósito es prever las fallas manteniendo los sistemas de infraestructura, equipos e instalaciones productivas en completa operación a los niveles y eficiencia óptimos. La característica principal de este tipo de mantenimiento es la de inspeccionar los equipos y detectar las fallas en su fase inicial, y corregirlas en el momento oportuno.

**b) Mantenimiento correctivo.**

Otra forma de mantenimiento es el llamado mantenimiento correctivo, dentro del cual podríamos distinguir entre:

*Mantenimiento correctivo no planificado:* Corrección de las averías o fallas, cuando éstas se presentan, y no planificadamente, al contrario del caso de mantenimiento preventivo. Esta forma de mantenimiento impide el diagnóstico fiable de las causas que provocan la falla, pues se ignora si falló por mal trato, por abandono, por desconocimiento del manejo, por desgaste natural, etc. El ejemplo de este tipo de mantenimiento correctivo no planificado es la habitual reparación urgente tras una avería en el equipamiento deportivo de la instalación.

*Mantenimiento correctivo planificado:* El mantenimiento correctivo planificado consiste la reparación de un equipo o máquina cuando se dispone del personal, repuestos, y documentos técnicos necesarios para efectuarlo.

**5.2 DISCIPLINAS.**

Son las diferentes actividades que se realizan dentro del Gimnasio, estas son:

**5.2.1 Aerobic / Anaeróbicos.**

Si bien los ejercicios aeróbicos aparecen como una excelente alternativa para bajar de peso o mantenerse en forma, entre sus beneficios también se incluye el hecho de que son de gran ayuda para controlar la presión arterial y darle fuerza al corazón.



***Figura 5.1 Ejercicio Aeróbic***

La actividad cardiovascular (o aeróbica) mejora la circulación coronaria, favoreciendo la distribución de los capilares en el músculo cardíaco y la habilidad del corazón para desarrollar nuevos ramales de arterias sanas, que permitan llevar la sangre a lugares donde antes llegaba en forma deficiente. También se produce un aumento de volumen de la cavidad ventricular, lo que supone una disminución de la frecuencia cardíaca en reposo y el consiguiente ahorro de gasto cardíaco.

De igual forma, numerosos estudios demuestran que la actividad aeróbica produce una reducción de los niveles de grasas en la sangre, y un descenso del colesterol y los triglicéridos que se depositan en las paredes de las arterias

### **5.2.2 Natación**

La natación es el arte de sostenerse y avanzar, usando los brazos y las piernas, sobre o bajo el agua. Puede realizarse como actividad lúdica o como deporte de competición. Debido a que los seres humanos no nadan instintivamente, la natación es una habilidad que debe ser aprendida. A diferencia de otros animales terrestres que se dan impulso en el agua, en lo que constituye en esencia una forma de caminar, el ser humano ha tenido que desarrollar una serie de brazadas y movimientos corporales que le impulsan en el agua con potencia y velocidad. En estos movimientos y estilos se basa la evolución de la natación competitiva como deporte.

### **5.2.3 Sauna**

Una **sauna** es una sala o habitación con un ambiente muy cálido, pudiendo tener un baño de vapor (*sauna húmeda* o *baño turco*) o bien ser una *sauna seca* (que utiliza piedras u otro material, que se calienta, sin liberar vapor). Es frecuente que a la permanencia en la sauna

le siga un baño o una ducha de agua fría, para liberar el cuerpo de toxinas expelidas con el sudor, y un masaje.<sup>3</sup>

La sauna húmeda raramente supera los 60°C, mientras que en la sauna seca el cuerpo humano tolera fácilmente temperaturas superiores a 80°C durante cortos periodos de

tiempo. Entre los beneficios de la sauna se encuentran el alivio de los dolores de columna, el aumento de la circulación sanguínea, la hidratación de la piel y desobstrucción de los poros, combate el estrés y la hipertensión... Además de eso, relaja la musculatura, limpia y desobstruye las vías respiratorias, desintoxica y expulsa las impurezas del organismo y combate enfermedades del aparato respiratorio. Sin embargo, la frecuencia habitual o prolongada de saunas debe ser autorizada por un médico, pues ciertas patologías respiratorias y circulatorias no se benefician con la permanencia en un ambiente caliente, así que la frecuencia de uso para las personas que las padezcan no debe superar más que una vez por semana.

#### **5.2.4 El Spinning**

El Spinning consiste en la realización de ejercicios sobre la bicicleta, compaginándolos con la música y efectuando coreografías, fundamentalmente buscando la diversión del público asistente, al mismo tiempo que trabaja a nivel global la tonificación del tronco y las extremidades superiores junto con un importante trabajo cardiovascular.



***Figura 5.2 Maquina para realizar el Spinning***

Los ritmos son diversos, pudiendo ir desde una gran lentitud, a una velocidad máxima, donde también se combinan las intensidades de trabajo.

---

<sup>3</sup> [www.wikipedia.org](http://www.wikipedia.org)

Tonificación de brazos, abdomen o cintura escapular, es relativamente fácil de efectuar. Si además incluimos ayudas de material como pesas de aeróbic o tensores, la tonificación puede completarse a cualquier grupo muscular.

Algo que no podemos olvidar a la hora de realizar spinning y que un buen profesor o instructor del mismo siempre ha de insistir en ello, es la corrección postural. La enorme combinación de ejercicios mencionados, el efectivo trabajo realizado, los beneficios que se buscan, pueden no conseguirse sin una correcta posición sobre la bici, donde, en todo momento tengamos en tensión el abdomen (a modo de corsé) y una buena elevación de la línea de hombros, sin llegar a arquear la espalda.

### **5.2.5 El Hapkido**

El Hapkido se basa en tres principios fundamentales <sup>4</sup> estos son:

**YU: Principio del agua, no resistencia, penetrar en la defensa del adversario y neutralizarlo utilizando técnicas blandas ó duras según su técnica**

No detiene la fuerza de un agresor directamente con fuerza, la redirecciona. Uno no detiene el golpe de un agresor aplicando fuerza en oposición directa al ataque, sino aplicando fuerza al lado, tangencialmente, de esta forma el ataque puede desviarse con menor utilización de energía.

**WON: Principio del círculo, tendencia al círculo para desviar la energía del adversario y aprovecharla para reducirlo ó vencerle.**

En el entrenamiento de Hapkido se hace énfasis en la teoría del círculo. Cada hombre tiene su propio círculo. Dentro de este círculo su territorio es privado. Si alguien trata de entrar en este dominio sin aprobación, entonces cada hombre tiene derecho de defenderse contra esta invasión.

---

<sup>4</sup> <http://hispagimnasios.com>



*Figura 5.3 El Hapkido*

En la práctica, cuando se produce un ataque y este no entra ilegalmente en el círculo de uno, y uno decide bloquear, es considerado una pérdida de tiempo y energía

**WHA: Principio de la armonía, armonía entre las técnicas duras-blandas, armonía entre mente y cuerpo.**

Cuando uno encuentra la armonía con uno mismo, el próximo paso es armonizar con el agresor. y por último aprender a armonizar con el entorno.

### **5.2.6 Taekwondo**

La principal característica de este arte marcial se centra en el uso de pies y manos sin protecciones para repeler a los oponentes.

Una de las cuestiones más importantes del Taekwondo es que no sólo es un arte marcial de autodefensa, sino que acentúa el buen sentido de sus practicantes. La autoconfianza hace a la gente más generosa en su actitud hacia el resto de la gente. Pueden pelear de igual a igual con sus oponentes, pero su código prohíbe asaltos injustos o el uso innecesario de la fuerza.



### **5.2.7 Entrenamiento con Pesas**

El entrenamiento con pesas es una de las técnicas que permite aumentar la fortaleza y resistencia muscular, trabajando varios grupos de músculos al mismo tiempo. Este trabajo puede realizarse por un objetivo estético -fisicoculturismo-, y también para otros propósitos como adquirir mayor fuerza para otras actividades deportivas o rehabilitarse de una lesión después de un período de reposo forzado.<sup>5</sup>

---

<sup>5</sup> <http://www.alemana.cl>

## **CAPITULO 6.**

# **LENGUAJE UNIFICADO DE MODELADO (UML)**

UML será utilizado en todas las fases de desarrollo del presente Sistema de Información, ya que presenta herramientas que ayudarán a visualizar, especificar, construir y documentar los artefactos del sistema, desde una perspectiva Orientada a Objetos.

### **6.1 INTRODUCCION A UML**

Es un lenguaje estándar que puede utilizarse para visualizar, especificar, construir y documentar los de un sistema que involucra una gran cantidad de software.

UML es un lenguaje ya que proporciona un vocabulario y unas reglas que se centran en la representación conceptual y física de un sistema, y que indican cómo crear y leer modelos bien formados.

UML es un lenguaje que mezcla gráficos y texto, pero en la notación UML hay una semántica bien definida, que se puede interpretar los modelos sin ambigüedad.

En este contexto, especificar significa construir modelos precisos, no ambiguos y completos. UML cubre la especificación de todas las decisiones de análisis, diseño e implementación que deben realizarse al desarrollar y desplegar un sistema.

Con UML sus modelos pueden conectarse de forma directa con una gran variedad de lenguajes de programación. Es posible establecer correspondencias desde un modelo *UML* a un lenguaje de programación, o incluso a tablas en una base de datos relacional o al almacenamiento persistente en una base de datos orientada a objetos.

## 6.2 BLOQUES DE CONSTRUCCIÓN DE UML

El vocabulario de UML incluye tres clases de bloques de construcción: Elementos, Relaciones, Diagramas.

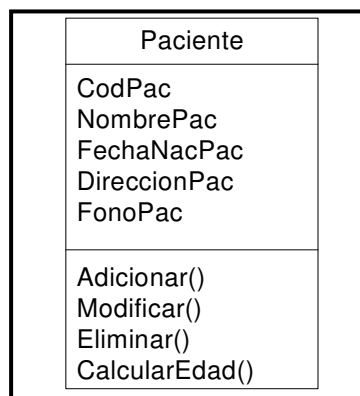
### 6.2.1 Elementos en UML

Hay cuatro tipos de elementos en UML, los cuales son: *elementos estructurales*, *de comportamiento*, *de agrupación* y *de anotación*.

#### a) Elementos Estructurales

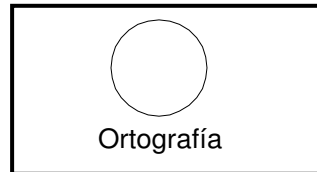
Son partes estáticas de un modelo, y representan cosas que son conceptuales o materiales.

**1.- UNA CLASE** es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. Una clase implementa una o más interfaces. *Ver la figura 6.1.*



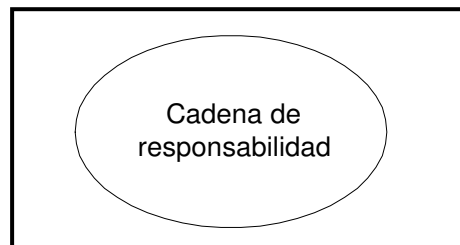
**Figura 6.1 Clases**

**2.- UNA INTERFACE** es una colección de operaciones que especifican un servicio de una clase o componente. Por lo tanto, una interfaz describe el comportamiento visible externamente de ese elemento. Una interfaz define un conjunto de especificaciones de operaciones, pero nunca un conjunto de implementaciones de operaciones. Gráficamente, un interfaz se representa como un círculo junto con su nombre. (Ver figura 6.2)



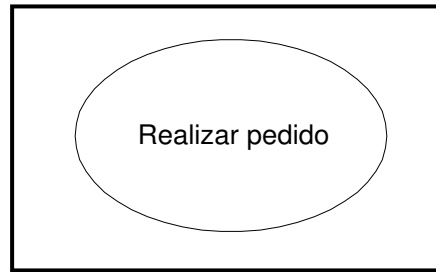
**Figura 6.2 Interfaces**

**3.- UNA COLABORACIÓN** define una interacción y es una sociedad de roles y otros elementos que colaboran para proporcionar un comportamiento cooperativo que la suma de los comportamientos de sus elementos. Una clase dada puede participar en varias colaboraciones. Estas colaboraciones, representan, pues, la implementación de patrones que forman un sistema. Gráficamente, una colaboración se representa como una elipse de borde discontinuo, incluyendo sólo su nombre. (Ver figura 6.3).

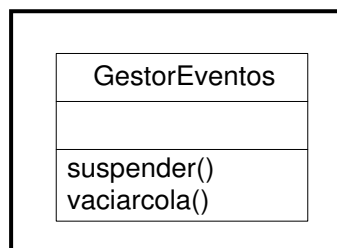


**Figura 6.3 Colaboraciones**

**4.- UN CASO DE USO** es una descripción de un conjunto de secuencias de acciones que un sistema ejecuta y que produce un resultado observable de interés para un actor particular. Un caso de uso se utiliza para estructurar los aspectos de comportamiento en un modelo. Un caso de uso es realizado por una colaboración. Gráficamente, un caso de uso se representa como una elipse de borde continuo, incluyendo normalmente solo su nombre. (Ver figura 6.4).

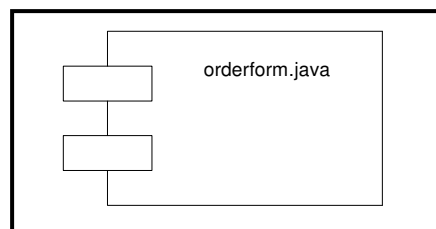
**Figura 6.4 Casos de uso**

**5.- UNA CLASE ACTIVA** es una clase cuyos objetos tienen uno o más procesos o hilos de ejecución y por lo tanto pueden dar origen a actividades de control. Una clase activa es igual que una clase, excepto en que sus objetos representan elementos cuyo comportamiento es concurrente\* con otros elementos. Gráficamente, una clase activa se representa como una clase, pero con líneas mas gruesas, incluyendo su nombre, atributos y operaciones. (Ver figura 6.5).

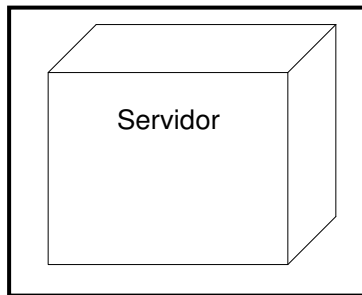
**Figura 6.5 Clases Activas**

**5.- UN COMPONENTE** es una parte física y reemplazable de un sistema que conforma con un conjunto de interfaces y proporciona la implementación de dicho conjunto.

Un componente representa típicamente el empaquetamiento físico de diferentes elementos lógicos, como clases, interfaces y colaboraciones. Gráficamente, un componente se representa como un rectángulo con pestañas, incluyendo sólo su nombre. (Ver figura 6.6).

**Figura 6.6 Componentes**

**7.- UN NODO** es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, que por lo general dispone de algo de memoria y, con frecuencia, capacidad de procesamiento. Un conjunto de componentes puede residir en un nodo y puede también migrar de un nodo a otro. Gráficamente, un nodo se representa como un cubo, incluyendo sólo su nombre (ver figura 6.7).

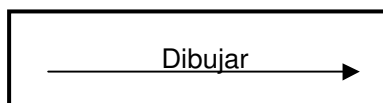


**Figura 6.7 Nodos**

## **b) Elementos de Comportamiento**

Son las partes dinámicas de los modelos UML. En total hay dos tipos principales de elementos de comportamiento:

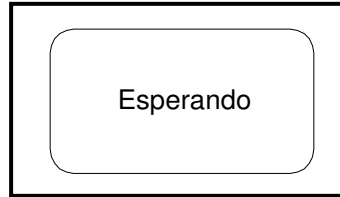
**1.- UNA INTERACCIÓN** es un comportamiento que comprende un conjunto de mensajes intercambiados entre un conjunto de objetos, dentro de un contexto particular, para alcanzar un propósito específico. El comportamiento de una sociedad de objetos o una operación individual puede especificarse con una interacción. Una interacción involucra muchos otros elementos, incluyendo mensajes, secuencias de acción (el comportamiento invocado por un mensaje) y enlaces (conexiones entre objetos). Gráficamente, un mensaje se muestra como una línea dirigida, incluyendo casi siempre el nombre de su operación (ver figura 6.8)



**Figura 6.8 Mensajes**

**2.- UNA MAQUINA DE ESTADOS** es un comportamiento que especifica las secuencias de estados por las que pasa un objeto o una interacción durante su vida en respuesta a

eventos, junto con sus reacciones a estos eventos. El comportamiento de una clase individual o una colaboración de clases puede especificarse con un maquina de estados.



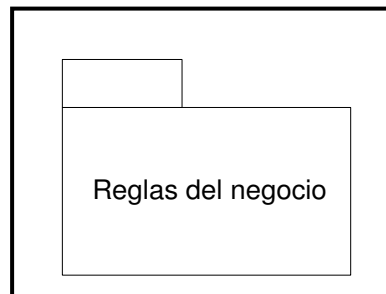
**Figura 6.9 Estados**

Una maquina de estados involucra a otros elementos, incluyendo estados, transiciones (el flujo de un estado a otro), eventos (que disparan una transición) y actividades (la respuesta a una transición). Gráficamente, un estado se representa como un rectángulo de esquinas redondeadas, incluyendo su nombre y sus subestados, si los tiene (ver figura 6.9).

### **c) Elementos de Agrupación**

Son las partes de organizativas de los modelos UML. Estos son las cajas en las que puede descomponerse un modelo. Hay un elemento de agrupación principal, las cuales son los paquetes.

**1.- UN PAQUETE** es un mecanismo de propósito general para organizar elementos en grupos. Los elementos estructurales, los elementos de comportamiento, e incluso otros elementos de agrupación pueden incluirse en un paquete. Ver figura 6.10



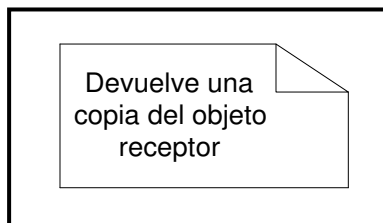
**Figura 6.10 Paquetes**

También hay variaciones, tales como los *frameworks*, los modelos y los subsistemas (tipos de paquetes)

#### **d) Elementos de Anotación**

Son las partes explicativas de los modelos UML. Son comentarios que se pueden aplicar para describir, clasificar y hacer observaciones sobre cualquier elemento de un modelo. Hay un tipo principal de elemento de anotación llamado nota.

**1.- UNA NOTA** es un símbolo para mostrar restricciones y comentarios junto a un elemento o una colección de elementos. Gráficamente, una nota se representa como un rectángulo con una esquina doblada, junto con un comentario textual o gráfico (ver figura 6.11).



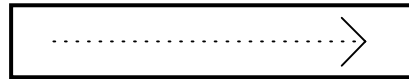
**Figura 6.11 Notas**

### **6.2.2 Relaciones de UML**

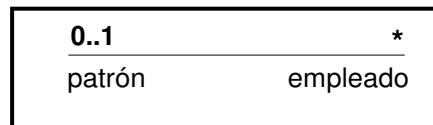
Hay cuatro tipos de relaciones como ser: Dependencia, Asociación, Generalización, Realización. Estas relaciones son los bloques básicos de construcción para relaciones de UML. Se utilizan para escribir modelos bien formados:

**1.- UNA DEPENDENCIA** es una relación semántica entre dos elementos, en la cual un cambio a un elemento (elemento independiente) puede afectar a la semántica del otro elemento (el elemento dependiente). Gráficamente, una dependencia se representa como una línea discontinua, posiblemente dirigida, que incluye a veces una etiqueta (ver figura 6.12).

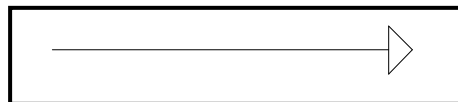


**Figura 6.12 Dependencias**

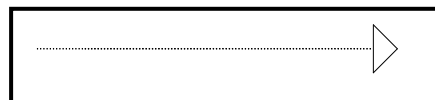
**2.- UNA ASOCIACIÓN** es una relación estructural que describe un conjunto de enlaces, los cuales son conexiones entre objetos. La agregación es un tipo especial de asociación, que representa una relación estructural entre un todo y sus partes. Gráficamente una asociación se representa como una línea continua, posiblemente dirigida, que a veces incluye una etiqueta, y a menudo incluye otros adornos, como la multiplicidad y los nombres de rol (ver figura 5.13).

**Figura 6.13 Asociaciones**

**3.- UNA GENERALIZACIÓN** es una relación de especialización / generalización en la cual los objetos del elemento especializado (el hijo) pueden sustituir a los objetos del elemento general (el padre). De esta forma, el hijo comparte la estructura y el comportamiento del padre. Gráficamente, una relación de generalización se representa como una línea continua con una punta de flecha vacía apuntando al padre. (Ver figura 6.14)

**Figura 6.14 Generalizaciones**

**4.- UNA REALIZACIÓN** es una relación semántica entre clasificadores, en donde un clasificador especifica un contrato que otro clasificador garantiza que cumplirá. Se pueden encontrar relaciones de realización en dos sitios: entre interfaces y las clases y componentes que las realizan, y entre los casos de uso y las colaboraciones que los realizan. Ver figura 6.15

**Figura 6.15 Realización**

### 6.2.3 Diagramas de UML

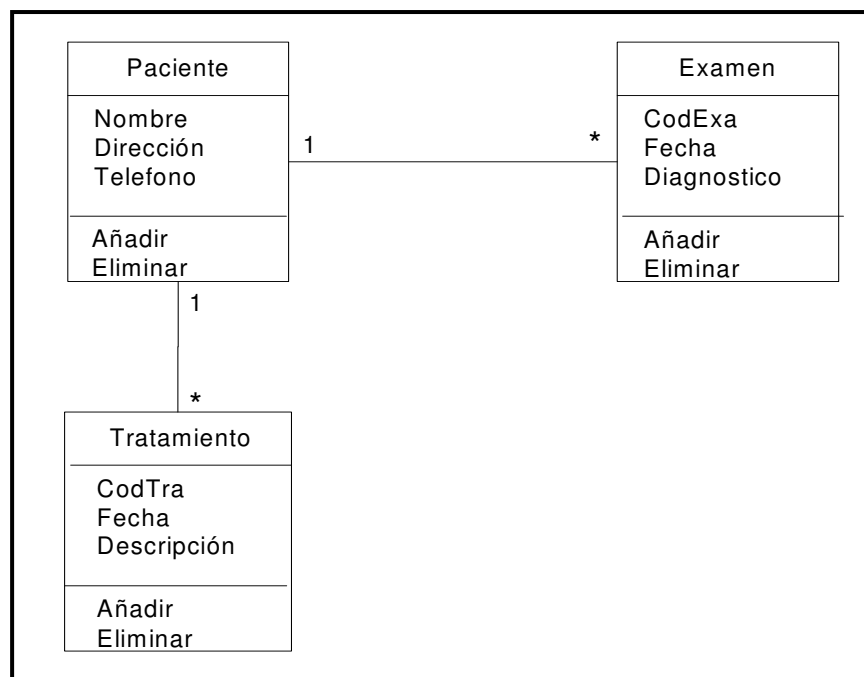
Un diagrama es la representación grafica de un conjunto de elementos, visualizado la mayoría de las veces como un grafo conexo de nodos (elementos) y arcos (relaciones). Los diagramas se dibujan para visualizar un sistema desde diferentes perspectivas, de forma que un diagrama es una proyección de un sistema.

#### a) Diagrama de Clases

*“Un diagrama de clases es un diagrama que muestra un conjunto de interfaces, colaboraciones y sus relaciones. Gráficamente, un diagrama de clases es una colección de nodos y arcos”*

Los diagramas de clases se utilizan para modelar la vista de diseño estático de un sistema. Principalmente, esto incluye modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas. Los diagramas de clases son importantes también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

En la figura 6.16 se puede observar un ejemplo de diagrama de clases.



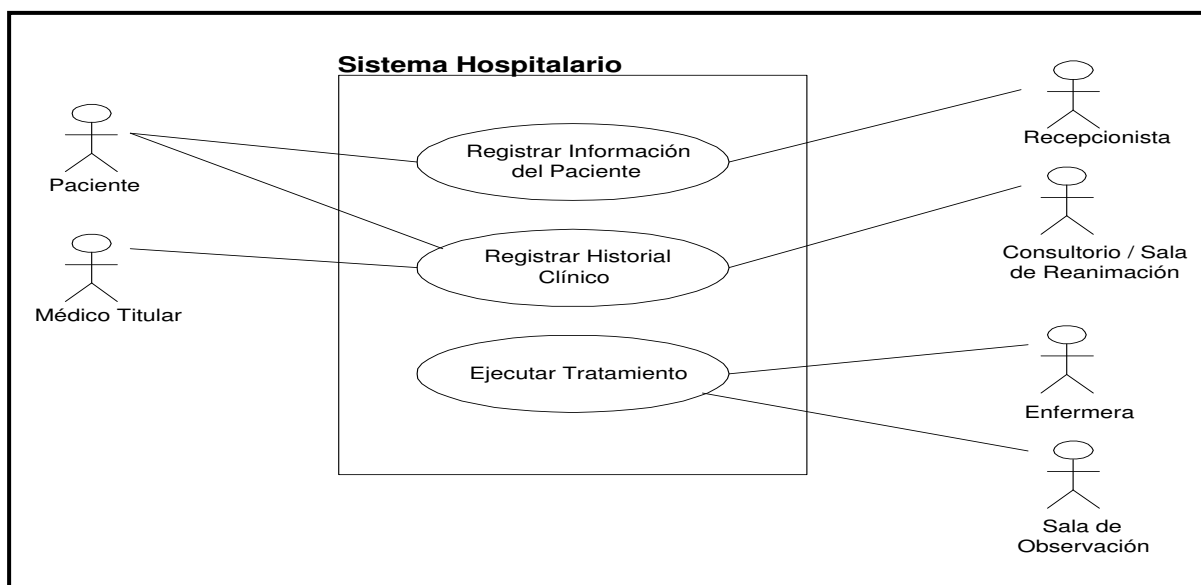
**Figura 6.16 Diagrama de Clases**

## b) Diagramas de Casos de Uso

*“Es una descripción de un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable de valor para un actor”*

La mayoría de las veces, implica modelar el contexto del sistema, subsistema o clase, o el modelado de los requisitos de comportamiento de esos elementos.

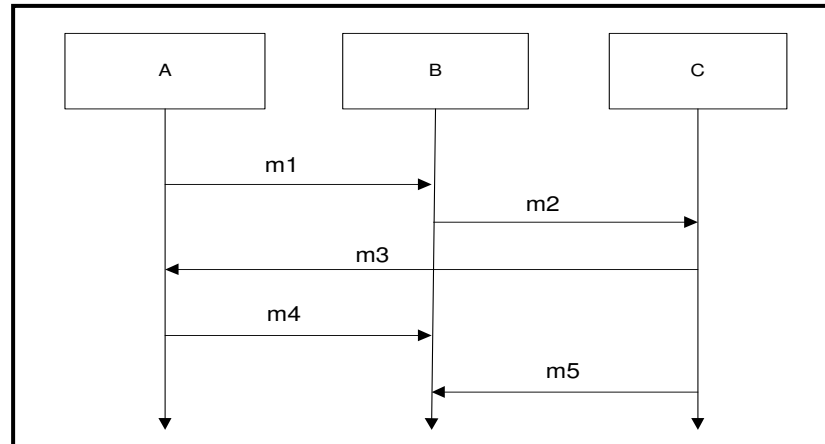
Cada Caso de Uso puede estar definido por: texto que lo describe, secuencia de pasos ejecutados dentro del escenario, condiciones pre-post para que el escenario comience o termine, mezclando las anteriores.



**Figura 6.17 Diagrama de Casos de Usos**

## c) Diagrama de Secuencia

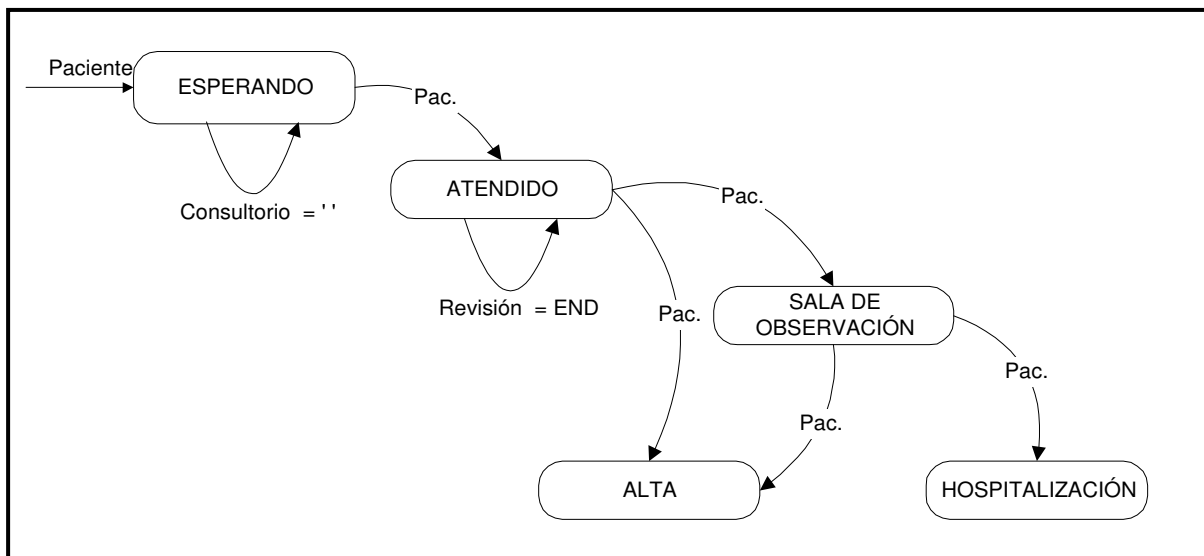
*“Es un diagrama de interacción que resalta la ordenación temporal de los mensajes; un diagrama de colaboración es un diagrama de interacción que resalta la organización estructural de los objetos que envían y reciben mensajes. Los diagramas de secuencia y los diagramas de colaboración son isomorfos, es decir que se puede tomar uno y trasformarlo en el otro” [BOO99]*



**Figura 6.18 Diagrama de secuencia**

#### **d) Diagrama de Estado**

“Un diagrama de estados muestra una máquina de estados, destacando el flujo de control entre estados. Una máquina de estados por las que pasa un objeto a lo largo de su vida en respuesta a eventos, junto con sus respuesta a esos eventos”. Véase en la figura 6.19 un ejemplo de un diagrama de estado.

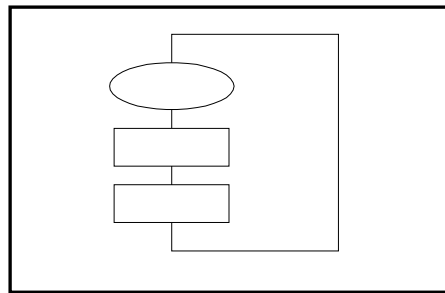


**Figura 6.19 Diagrama de Estados**

**e) Diagrama de Componentes**

*Un diagrama de componentes muestra la organización y las dependencias entre un conjunto de componentes. Los diagramas de componentes cubren la vista de implementación estática de un sistema. Se relacionan con los diagramas de clases en que un componente se corresponde, por lo común, con una o más clases, interfaces o colaboraciones.[BOO99]*

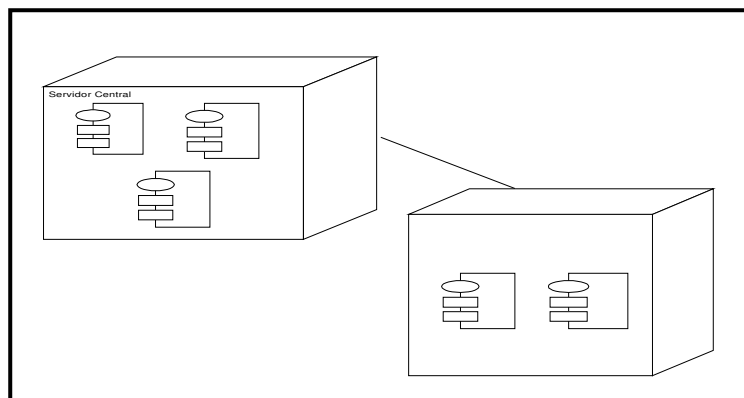
Un ejemplo de diagrama de componente véase en la figura 6.20.



**Figura 6.20 Diagrama de Componente**

**f) Diagramas de Despliegue**

Los diagramas de despliegue se utilizan para modelar la vista de despliegue estática de un sistema, esto implica modelar la topología del hardware sobre el que se ejecuta el sistema. Los diagramas de despliegue son fundamentalmente diagramas de clases que se ocupan de modelar los nodos de un sistema. Véase el diagrama en la figura 6.20.



**Figura 6.20 Diagrama de Despliegue**

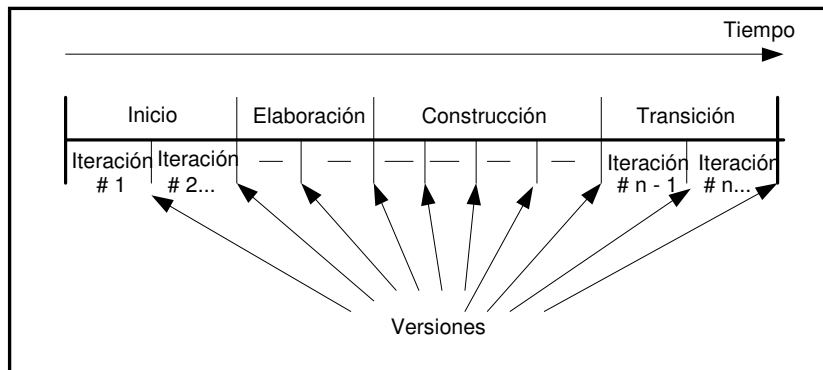
## **CAPITULO 7.**

# **EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE**

El presente capítulo describirá el Proceso Unificado de Desarrollo de Software, donde se explicará en qué consiste los verdaderos aspectos definitorios del Proceso que son: Dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental, también se definirán las actividades que se realizan en los flujos de trabajos.

### **7.1 INTRODUCCIÓN**

El Proceso Unificado es un enfoque de ciclo de vida de desarrollo de software que se adapta muy bien a UML. El objetivo del Proceso Unificado es permitir la producción de un software de la alta calidad que satisfaga las necesidades de los usuarios finales dentro de planificaciones y presupuestos predecibles.



**Figura 7.1** Un ciclo con sus fases e iteraciones

El ciclo de vida del Proceso Unificado consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide a su vez en iteraciones. Véase la figura 7.1.

## 7.2 UN PROCESO DIRIGIDO POR CASOS DE USO

El Proceso Unificado está dirigido por casos de uso, quiere decir que el proceso de desarrollo sigue un hilo, avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Véase la figura 7.2.



**Figura 7.2** Los casos de uso enlazan los flujos de trabajo

Los desarrolladores crean un modelo de análisis que utiliza el modelo de casos de uso como entrada. Cada caso de uso en el modelo de casos de uso se traducirá en una realización de caso de uso en el modelo de análisis. La dualidad caso de uso/realización de caso de uso es la base de una trazabilidad directa entre los requisitos y el análisis. Tomando los casos de uso uno a uno, se pueden identificar las clases que participan en la realización de los casos de uso, es decir las clases se recogen de las descripciones de los casos de uso.

A continuación, se implementan las clases diseñadas mediante un conjunto de ficheros (código fuente) en el modelo de implementación, a partir de los cuales se pueden producir (compilar y enlazar) ejecutables, como DLL's y componentes ActiveX.

Por último, durante el flujo de trabajo de prueba los ingenieros de prueba verifican que el sistema implementa de verdad la funcionalidad descrita en los casos de uso y que satisface los requisitos del sistema.

### **7.3 UN PROCESO CENTRADO EN LA ARQUITECTURA**

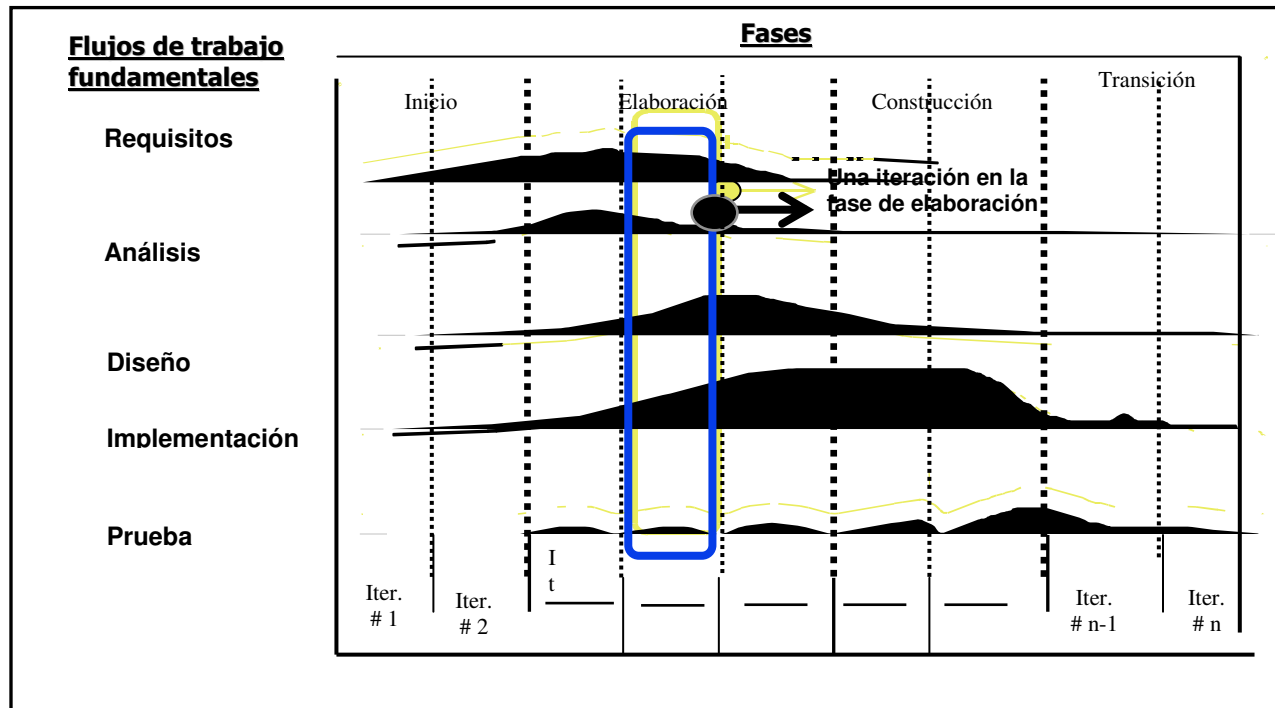
El Proceso Unificado está centrado en la arquitectura software lo que incluye los aspectos estáticos y dinámicos más significativos del sistema. ¿Cómo se puede determinar qué elementos son importantes?. Su importancia reside en el hecho de que guían en el trabajo con el sistema, tanto en este ciclo como a través del ciclo de vida completo. Estos elementos significativos, incluyen algunos de los subsistemas, dependencias, interfaces, colaboraciones, nodos y clases activas.

La arquitectura también se ve influida por la plataforma en la que tiene que funcionar el software, los bloques de construcción de que se dispone, consideraciones de implantación, sistemas heredados y requisitos no funcionales.

### **7.4 UN PROCESO ITERATIVO E INCREMENTAL**

El Proceso Unificado es iterativo e incremental porque propone una comprensión incremental del problema a través de refinamientos sucesivos y un crecimiento incremental de una solución efectiva a través de varios ciclos. Como parte del enfoque iterativo se encuentra la flexibilidad para acomodarse a nuevos requisitos o a cambios tácticos en los objetivos del negocio. Véase la figura 7.4





**Figura 7.3 Proceso Unificado de Desarrollo de Software**

Dentro de cada fase, se puede descomponer adicionalmente el trabajo *en iteraciones con sus incrementos resultantes*. Cada fase termina con un hito.

Como se observa en la figura 7.3 el ciclo de vida abarca fases básicas, que se describen a continuación.

## 7.5 LA FASE DE INICIO ESTABLECE LA VIABILIDAD

Para justificar la puesta en marcha del proyecto se siguen cuatro pasos:

1. Delimitar el ámbito del sistema propuesto, es decir, definir los límites del sistema y empezar a identificar las interfaces con sistemas relacionados que están fuera de los límites.
2. Describir o esbozar una propuesta de la arquitectura del sistema, y en especial de aquellas partes del sistema que son nueva, arriesgada o difícil.
3. Identificar riesgos críticos, es decir, los que afectan a la capacidad de construir el sistema, y determinar si se puede determinar una forma de mitigarlos.

4. Demostrar a usuarios o clientes potenciales que el sistema propuesto es capaz de solventar sus problemas o de mejorar sus objetivos de negocio.

## **7.6 LA FASE DE ELABORACIÓN SE CENTRA EN LA FACTIBILIDAD.**

Las tareas a realizar en esta fase son las siguientes:

1. Se crea una línea base para la arquitectura que cubre la funcionalidad del sistema significativa arquitectónicamente y las características importantes para las personas involucradas.
2. Identifica los riesgos significativos, es decir, los riesgos que podrían perturbar los planes, costes y planificaciones de fases posteriores.
3. Especifica los niveles a alcanzar por los atributos de calidad, como la fiabilidad y los tiempos de respuesta.
4. Recopila casos de uso para aproximadamente el 80 por ciento de los requisitos funcionales, suficientes para planificar la fase de construcción.

## **7.7 LA FASE DE CONSTRUCCIÓN CONSTRUYE EL SISTEMA.**

El propósito es dejar listo un producto software en su versión operativa inicial. Entre las actividades que se tienen que realizar en esta fase se tienen las siguientes:

1. La extensión de la identificación, descripción y realización de casos de uso a todos los casos de uso.
2. La finalización del análisis (posiblemente quedan todavía más de la mitad de los casos de uso por ser analizados cuando comienza la fase de construcción), del diseño, de la implementación y de la prueba (posiblemente queda un 90 por ciento).
3. El mantenimiento de la integridad de la arquitectura, modificándola cuando sea necesario.
4. La monitorización de riesgos críticos y significativos arrastrados desde las dos primeras fases.

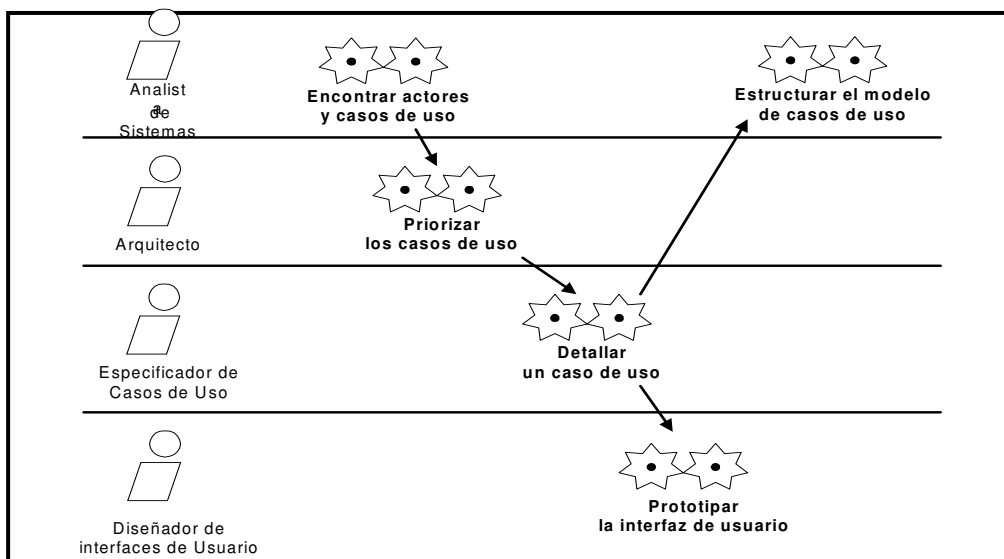
## 7.8 LA FASE DE TRANSICIÓN SE METE DENTRO DEL ENTORNO DEL USUARIO.

Se distribuye un producto software (versión beta) para determinar si el sistema hace lo que demandan sus usuarios y el negocio. Entre las actividades ha realizar para la fase de transición se tiene:

1. Preparar las actividades, como la preparación del lugar.
2. Aconsejar al cliente sobre la actualización del entorno en los que se supone que el software va a funcionar.
3. Preparar los manuales y otros documentos para la entrega del producto.
4. Corregir los defectos encontrados a lo largo de las pruebas realizadas a la versión beta.
5. Modificar el software al detectar problemas que no habían sido previstos.

## 7.9 CAPTURA DE REQUISITOS

El diagrama utiliza calles para mostrar qué trabajadores ejecutan qué actividades, se describen los flujos de trabajo como una secuencia de actividades que están ordenadas, así que una actividad produce una salida que sirve de entrada a la siguiente actividad. (*Véase Figura 7.4*).

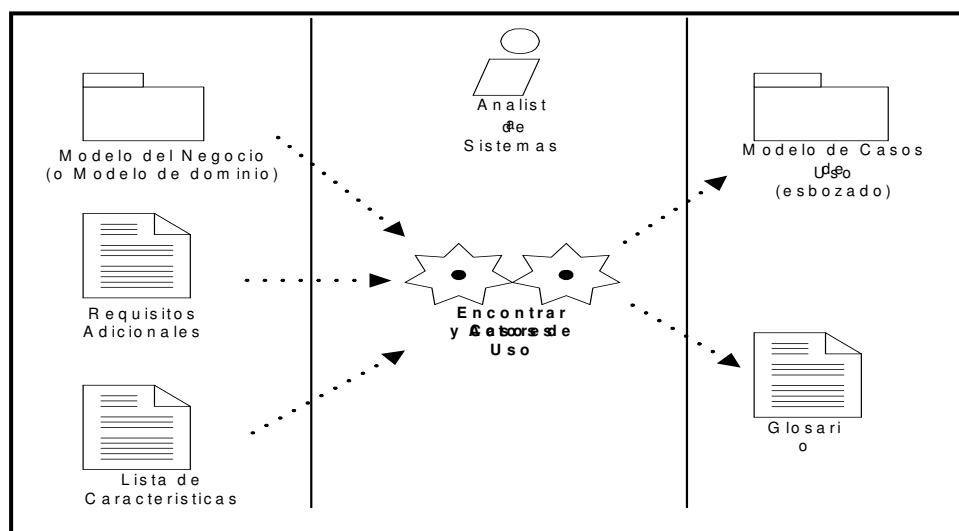


*Figura 7.4 Flujo de trabajo para la captura de requisitos en forma de casos de uso*

### 7.9.1 Actividad: Encontrar Actores y Casos de Uso

Esta actividad consta de cuatro pasos (*Véase Figura 7.5*):

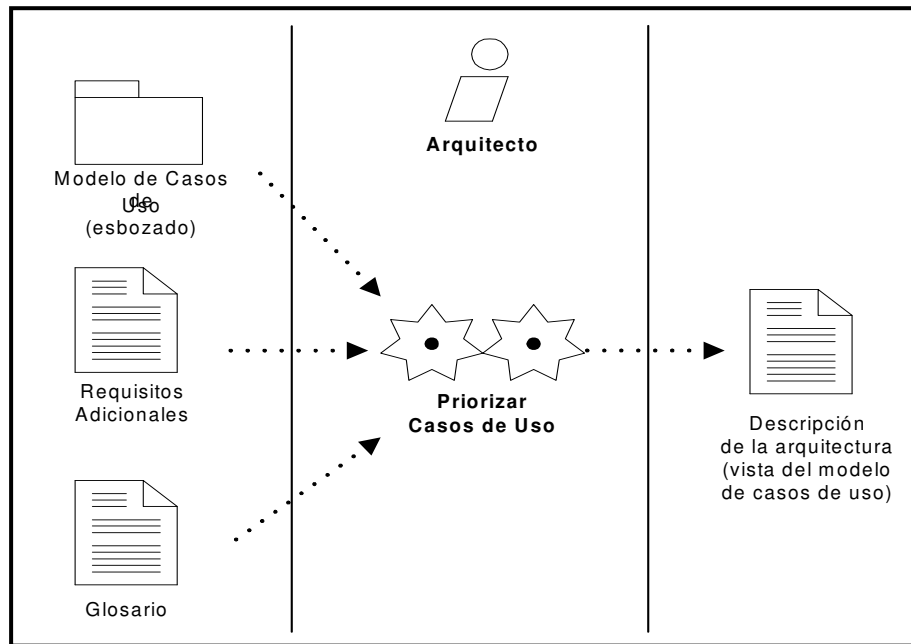
- Encontrar los actores.
- Encontrar los casos de uso.
- Describir brevemente cada caso de uso.
- Describir el modelo de caso de uso completo, preparación de un glosario de términos.



*Figura 7.5 Las entradas y los resultados de identificar actores y casos de uso.*

### 7.9.2 Actividad: Priorizar Casos de Uso

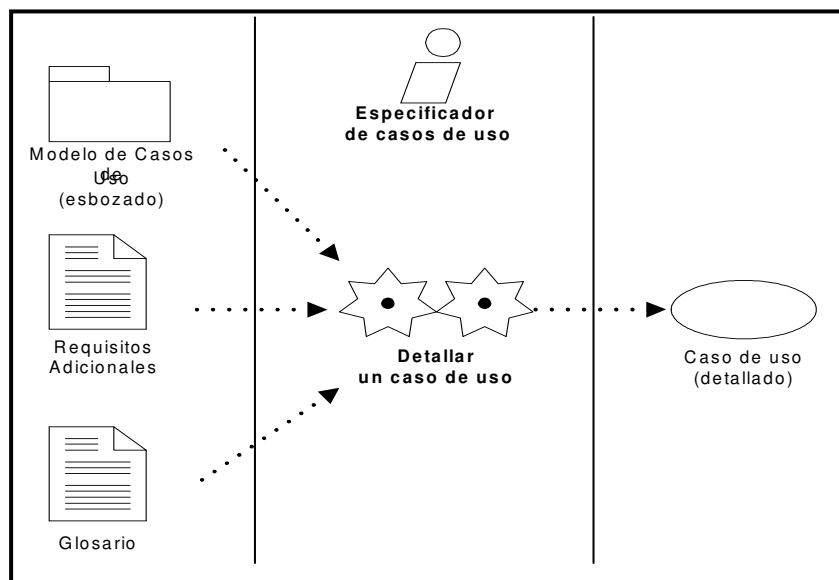
La vista de la arquitectura del modelo de casos de uso debería incluir los casos de uso que describen alguna funcionalidad importante y crítica, o que impliquen un requisito importante que deba desarrollarse pronto dentro del ciclo de vida del software. (*Véase Figura 7.6*)



**Figura 7.6 Entradas y resultados de priorizar los casos de uso**

### 7.9.3 Actividad: Descripción Detallada de un Caso de Uso

El objetivo de detallar cada caso de uso es describir su flujo de sucesos en detalle, incluyendo cómo comienza, termina e interactúan con los actores. (Véase **Figura 7.7**)

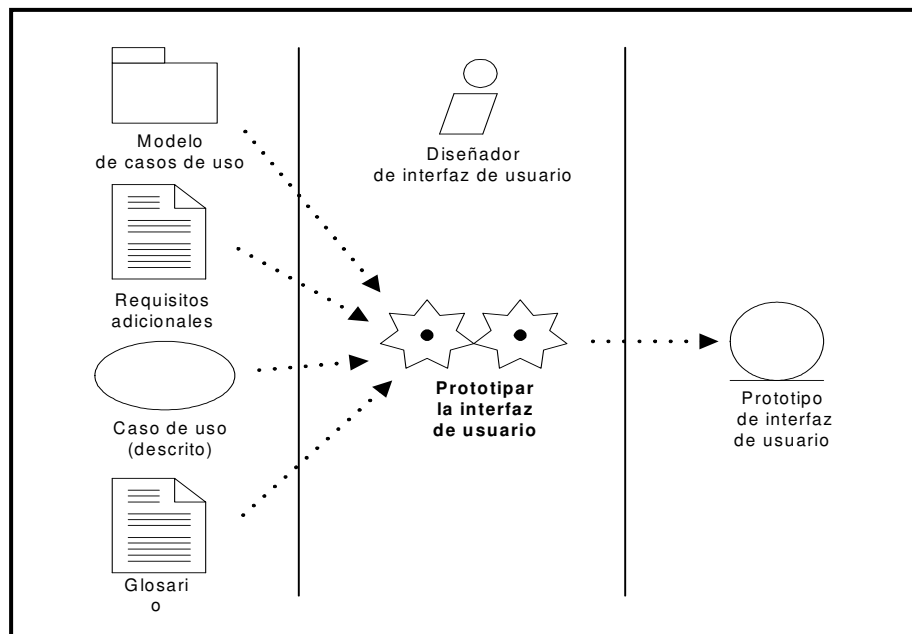


**Figura 7.7 Entradas y resultados de detallar un caso de uso**

### 7.9.4 Actividad: Prototipar la Interfaz de Usuario

El objetivo de esta actividad es construir un prototipo de interfaz de usuario. (Véase *Figura 7.8*)

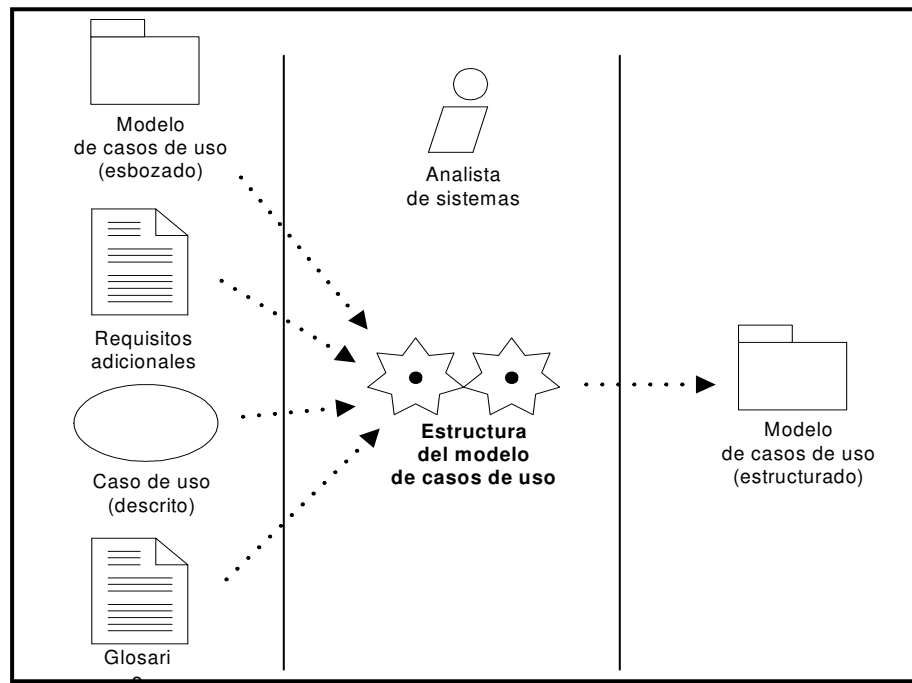
Comienza con los casos de uso y se intenta discernir qué se necesita de las interfaces de usuario para habilitar los casos de uso para cada actor. Esto es, un diseño lógico de la interfaz de usuario, luego el diseño físico de la interfaz de usuario y por último luego se desarrollarán prototipos que ilustren cómo pueden utilizar el sistema los usuarios para los casos de uso.



*Figura 7.8 Entrada y resultados de Prototipar la interfaz de usuario*

### 7.9.5 Actividad: Estructurar el Modelo de Casos de Uso

El objetivo de estructurar el modelo de casos de uso es: extraer descripciones de funcionalidad (de casos de uso) generales y compartidas que pueden ser utilizadas por descripciones (de casos de uso) más específicas. (Véase *Figura 7.9*)

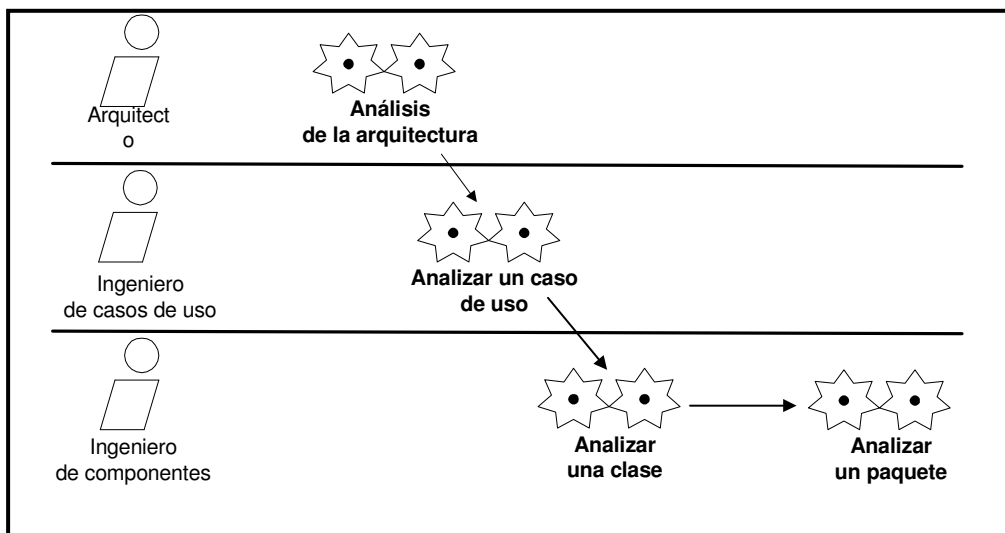


**Figura 7.9** Las entradas y resultados de encontrar generalizaciones en el modelo de Casos de Uso.

## 7.10 ANÁLISIS

Se utilizará un diagrama de actividad para razonar acerca de su comportamiento dinámico.

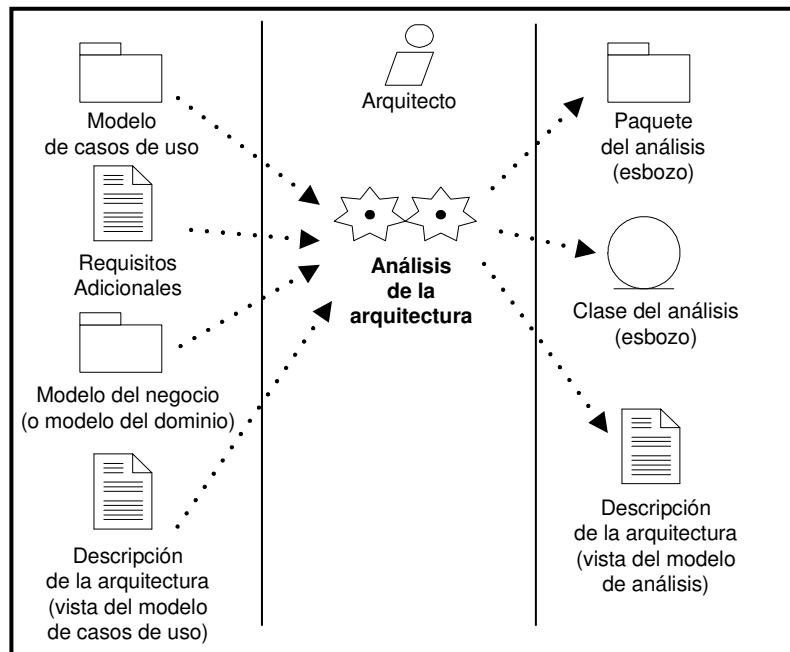
(Véase *Figura 7.10*)



**Figura 7.10** El flujo de trabajo en el análisis, incluyendo a los trabajadores participantes y sus actividades.

### 7.10.1 Actividad: Análisis de la Arquitectura

El propósito es esbozar el modelo de análisis y la arquitectura mediante la identificación de paquetes del análisis, clases del análisis evidentes, y requisitos especiales comunes. (Véase *Figura 7.11*)



**Figura 7.11** Las entradas y los resultados del análisis arquitectónico.

- Identificación de Paquetes del Análisis. Entre las “asignaciones” adecuadas de casos de uso a un paquete en concreto tenemos las siguientes:

- Los casos de uso requeridos para dar soporte a un determinado proceso de negocio.
- Los casos de uso requeridos para dar soporte a un determinado actor del sistema.
- Los casos de uso que están relacionados mediante relaciones de generalización, y de extensión.

- Identificación de Clases de Entidades Obvias. Se debe preparar una propuesta preliminar de las clases de entidad más importantes (10 o 20) basado en las clases del dominio o las entidades del negocio que se identificaron durante la captura de requisitos.

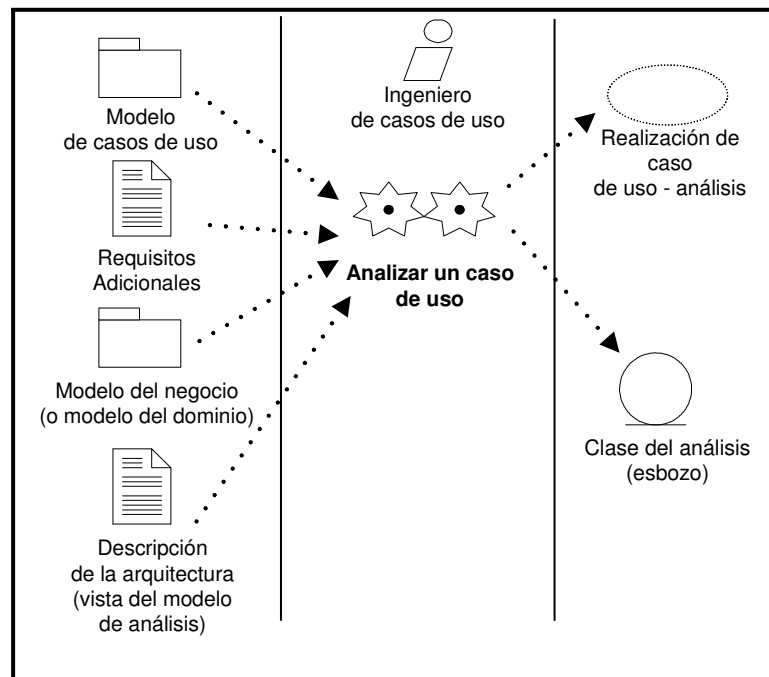
- Identificación de Requisitos Especiales Comunes. Es un requisito que aparece durante el análisis para luego ser tratado adecuadamente en las subsiguientes actividades de diseño e implementación.



### 7.10.2 Actividad: Analizar un Caso de Uso

Se analiza un caso de uso para (*Véase figura 7.12*):

- Identificar las clases del análisis cuyos objetos son necesarios para llevar a cabo el flujo de sucesos del caso de uso.
- Distribuir el comportamiento del caso de uso entre los objetos del análisis que interactúan.
- Capturar requisitos especiales sobre la realización de los casos de uso.



**Figura 7.12 Las entradas y los resultados del análisis de un caso de uso**

Normas general para identificar las clases del análisis:

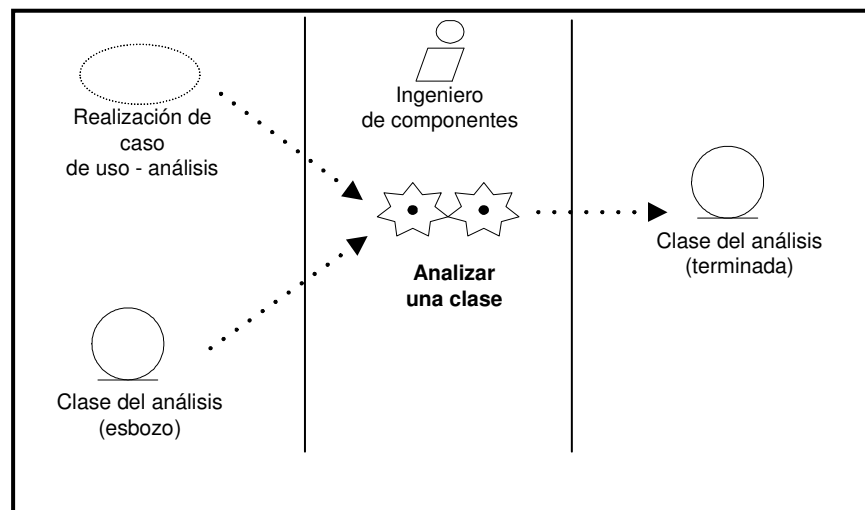
- Identificar clases de entidad mediante el estudio en detalle de la descripción del caso de uso y de cualquier modelo del dominio que se tenga.
- Identificar una clase de interfaz central para cada actor humano, y dejar que esta clase represente la ventana principal del interfaz de usuario con el cual interactúa el actor.
- Identificar una clase de interfaz primitiva para cada clase de entidad. Estas clases representan objetos lógicos con las cuales interactúa el actor (humano) en la interfaz de usuario durante el caso de uso.

- Identificar una clase de interfaz central para cada actor que sea un sistema externo, y dejar que esta clase represente la interfaz de comunicación.
- Identificar una clase de control responsable del tratamiento de control y de la coordinación de la realización de caso de uso, y después refinar esta clase de control de acuerdo a los requisitos del caso de uso.

### 7.10.3 Actividad: Analizar una Clase

Los objetivos de analizar una clase son:

- Identificar y mantener las responsabilidades de una clase del análisis, basadas en su papel en las realizaciones de caso de uso (*Véase figura 7.13*).
- Identificar y mantener los atributos y relaciones de la clase del análisis.
- Capturar requisitos especiales sobre la realización de la clase del análisis.

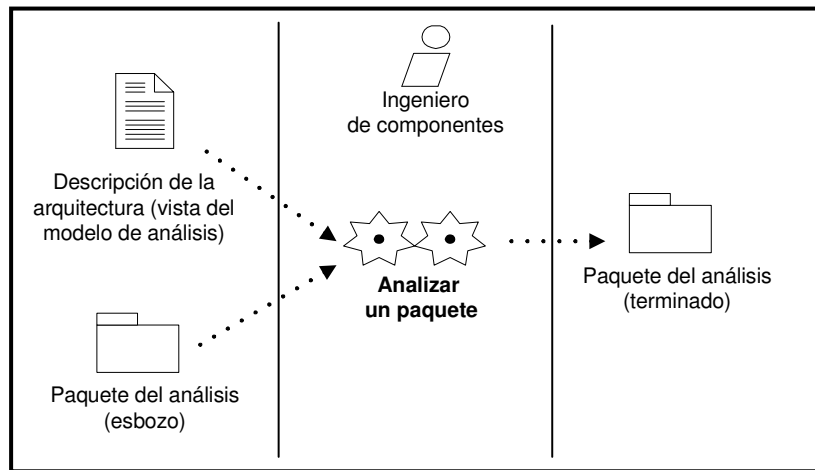


*Figura 7.13 Las entradas y los resultados del análisis de una clase.*

### 7.10.4 Actividad: Analizar un Paquete.

Los objetivos de analizar un paquete, como se muestra en la figura 7.14, son:

- Garantizar que el paquete del análisis es tan independiente de otros paquetes como sea posible.
- Garantizar que el paquete del análisis cumple su objetivo de realizar algunas clases del dominio o casos de uso.
- Describir las dependencias de forma que pueda estimarse el efecto de los cambios futuros.



**Figura 7.14** Las entradas y los resultados del análisis de un paquete

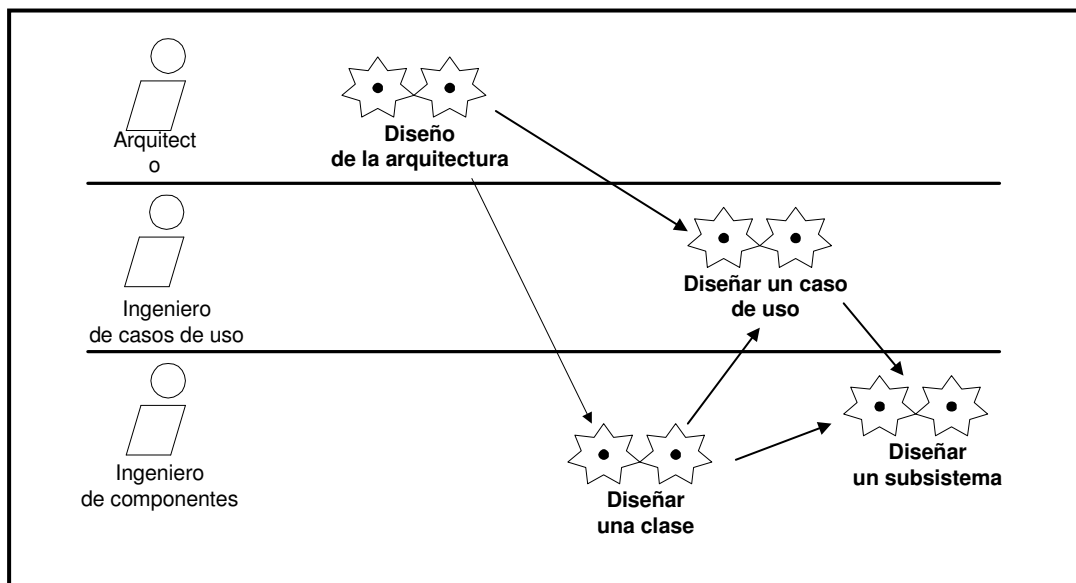
Las siguientes son algunas normas generales para esta actividad:

- Definir y mantener las dependencias del paquete con otros paquetes cuyas clases contenidas estén asociadas con él.
- Asegurar que el paquete contenga las clases correctas.
- Limitar las dependencias con otros paquetes.

## 7.11 DISEÑO

En concreto, los propósitos del diseño son:

- Adquirir una comprensión a profundidad de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia, tecnologías de interfaz de usuario, etc.
- Crear una entrada apropiada y un punto de partida para actividades de implementación subsiguientes capturando los requisitos o subsistemas individuales, interfaces y clases.
- Ser capaces de descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo, teniendo en cuenta la posible concurrencia. Véase **Figura 7.15**

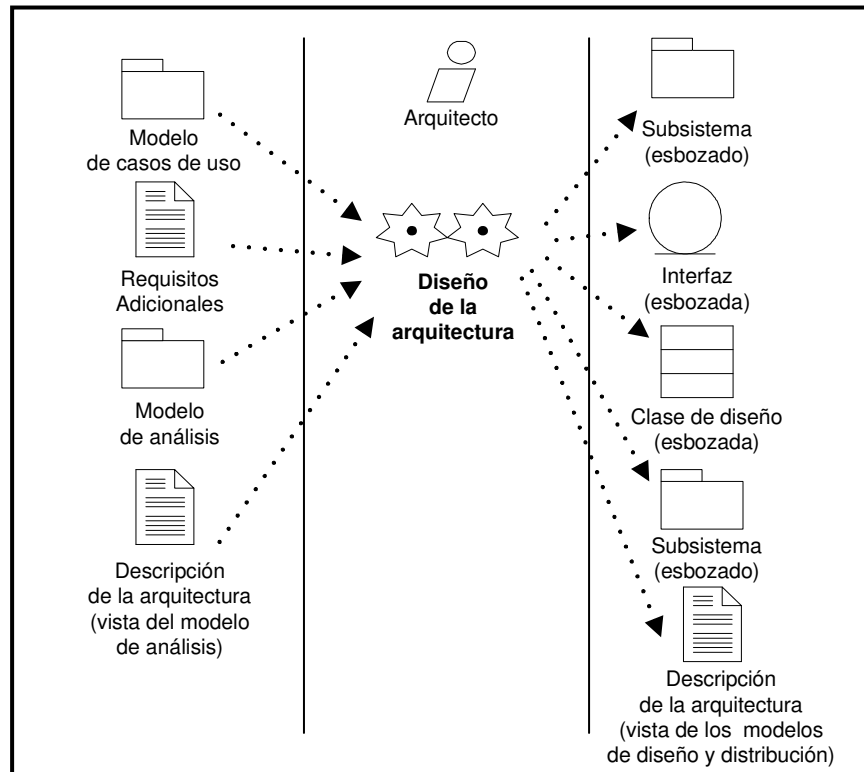


**Figura 7.15** El flujo de trabajo en el diseño, incluyendo los trabajadores participantes y sus actividades

### 7.11.1 Actividad: Diseño de la Arquitectura

El objetivo del diseño de la arquitectura es esbozar los modelos de diseño y despliegue y su arquitectura mediante la identificación de los siguientes elementos (*véase Figura 7.16*):

- Nodos y sus configuraciones de red
- Subsistemas y sus interfaces
- Clases del diseño significativas para la arquitectura, como las clases activas.
- Mecanismos de diseño genéricos que tratan requisitos comunes, como los requisitos especiales sobre persistencia, distribución, rendimiento, y demás, tal y como se capturaron durante el análisis sobre las clases y las realizaciones de caso de uso – análisis.

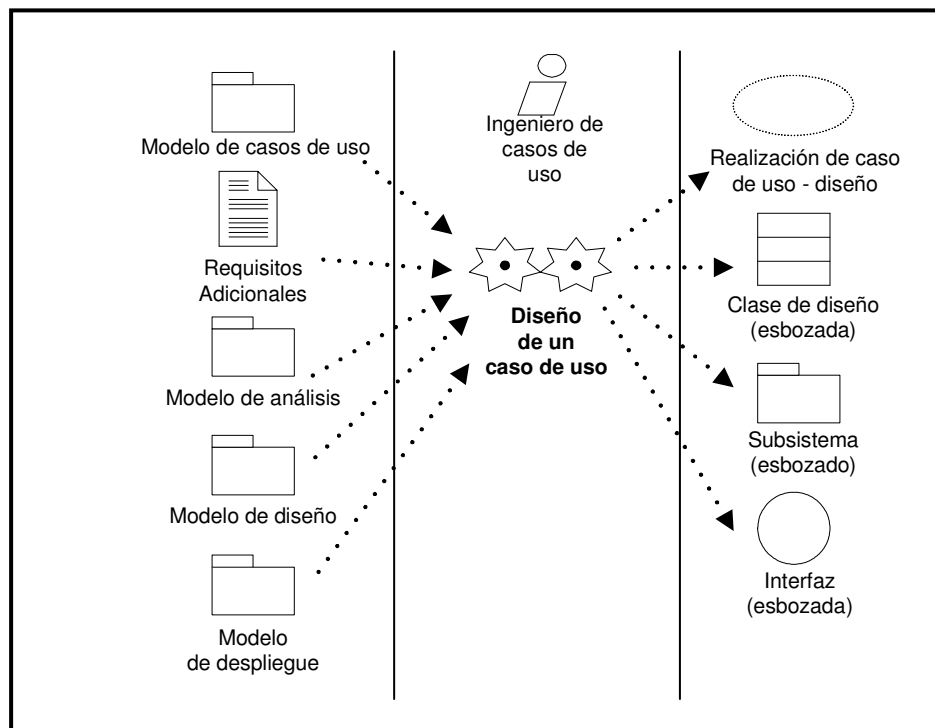


**Figura 7.16** Las entradas y los resultados del diseño de la arquitectura

### 7.11.2 Actividad: Diseño de un Caso de Uso

Los objetivos del diseño de un caso de uso son (*véase Figura 7.17*):

- Identificar las clases del diseño y/o los subsistemas cuyas instancias son necesarias para llevar a cabo el flujo de sucesos del caso de uso.
- Distribuir el comportamiento del caso de uso entre los objetos del diseño que interactúan y/o entre los subsistemas participantes.
- Definir los requisitos sobre las operaciones de las clases del diseño y/o sobre los subsistemas y sus interfaces.
- Capturar los requisitos de implementación

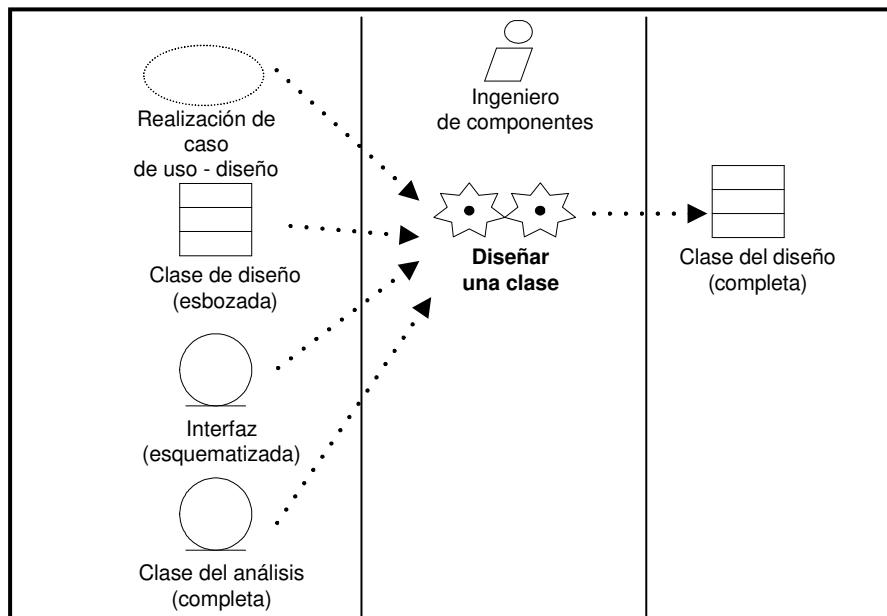


**Figura 7.17** Las entradas y los resultados del diseño de un caso de uso

### 7.11.3 Actividad: Diseño de una Clase

El propósito de diseñar una clase es crear una clase del diseño que cumpla su papel en las realizaciones de los casos de uso y los requisitos no funcionales que se aplican a estos (*Véase Figura 7.18*). Esto incluye el mantenimiento del diseño de clases en sí mismo y los siguientes aspectos de éste:

- Sus operaciones
- Sus atributos
- Las relaciones en las que participa
- Sus métodos (que realizan sus operaciones)
- Los estados impuestos
- Sus dependencias con cualquier mecanismo de diseño genérico
- Los requisitos relevantes a su implementación
- La correcta realización de cualquier interfaz requerida

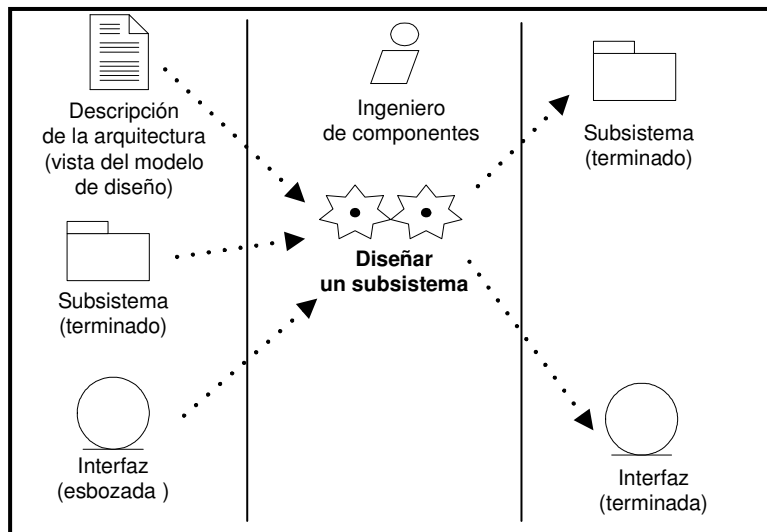


**Figura 7.18** La entrada y los resultados del diseño de una clase.

#### 7.11.4 Actividad: Diseño de un Subsistema

Los objetivos del diseño de un subsistema son (*Véase Figura 7.19*):

- Garantizar que el subsistema es tan independiente como sea posible de otros subsistemas y/o de sus interfaces.
- Garantizar que el subsistema proporciona las interfaces correctas.



**Figura 7.19** Las entradas y los resultados del diseño de un subsistema

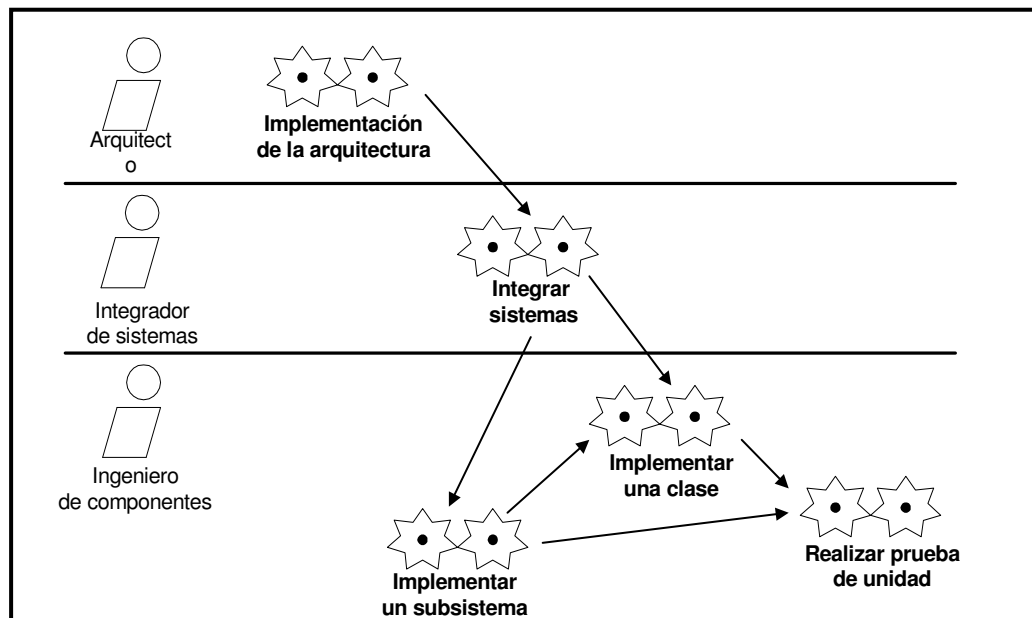
- Garantizar que el subsistema cumple su propósito de ofrecer una realización correcta de las operaciones tal y como se definen en las interfaces que proporciona.

## 7.12 IMPLEMENTACIÓN

Los propósitos de la implementación son:

- Planificar las integraciones de sistema necesarias en cada iteración. Para ello se sigue un enfoque incremental, es decir en una sucesión de pasos pequeños y manejables.
- Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue.
- Implementar las clases y subsistemas encontrados durante el diseño.
- Probar los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones de sistema.

En la figura 7.20 se utiliza un diagrama de actividades para razonar su comportamiento dinámico.



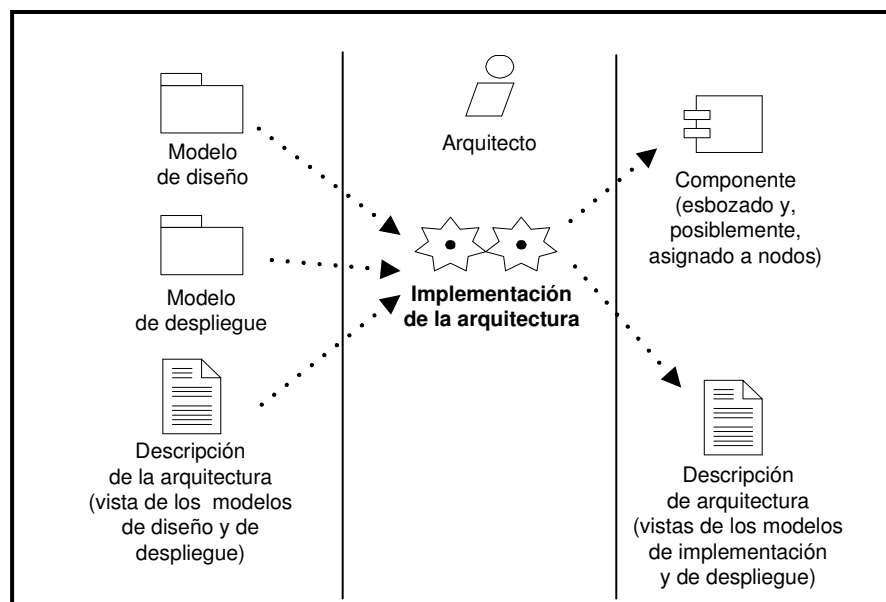
*Figura 7.20 Flujo de trabajo en la etapa de implementación, incluyendo los trabajadores participante y sus actividades*



### 7.12.1 Actividad: Implementación de la Arquitectura

El propósito de la implementación de la arquitectura es esbozar el modelo de implementación y su arquitectura mediante (*Véase Figura 7.21*):

- La identificación de componentes significativos arquitectónicamente, tales como componentes ejecutables.
- La asignación de componentes a los nodos en las configuraciones de redes relevantes.

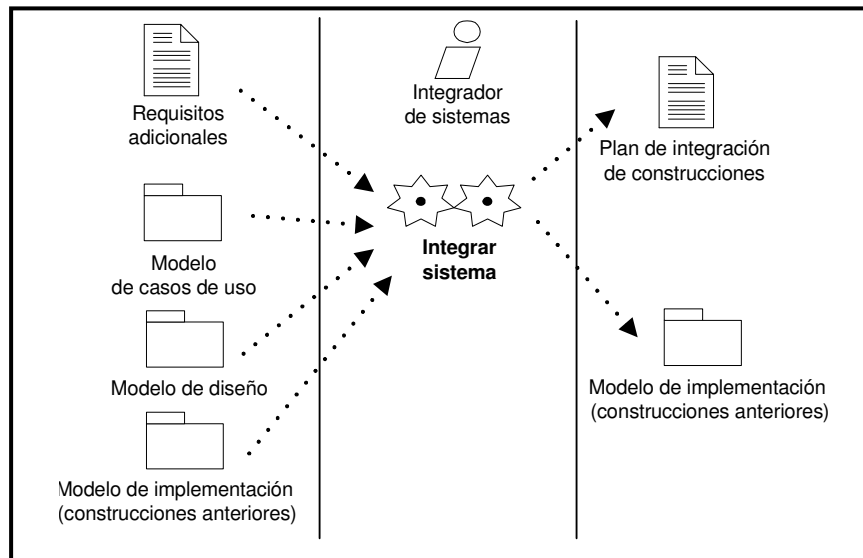


*Figura 7.21 La entrada y el resultado de la implementación de la arquitectura*

### 7.12.2 Actividad: Integrar el Sistema

Los objetivos de la integración del sistema son:

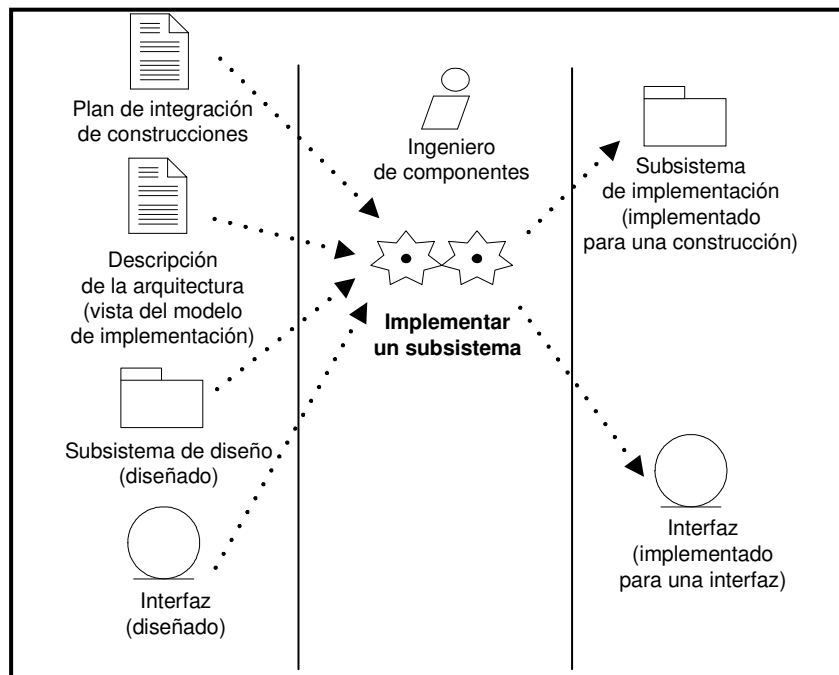
- Crear un plan de integración de construcciones que describa las construcciones necesarias en una iteración y los requisitos de cada construcción (*véase Figura 7.22*).
- Integrar cada construcción antes de que sea sometida a pruebas de integración.



**Figura 7.22** *Entrada y resultado de la integración de sistemas.*

### 7.12.3 Actividad: Implementar un Subsistema

El propósito de implementar un subsistema es el de asegurar que un subsistema cumple su papel en cada construcción, tal y como se especifica en el plan de integración de la construcción.



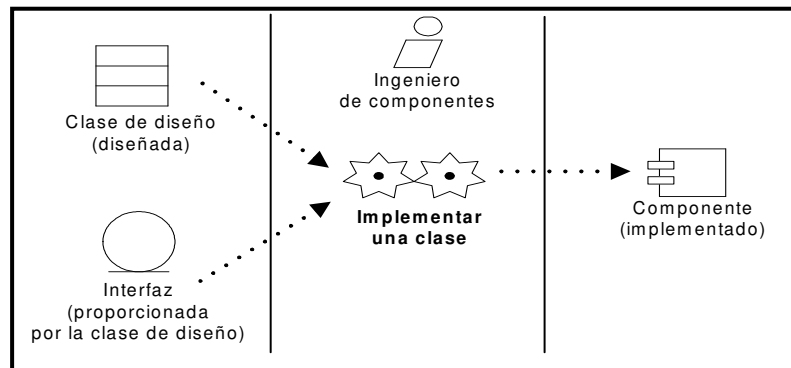
**Figura 7.23** *La entrada y el resultado de la implementación de subsistemas*

Esto quiere decir que se asegura que los requisitos implementados en la construcción y aquellos que afectan al subsistema son implementados correctamente por componentes o por otros subsistemas (recursivamente) dentro del subsistema (*véase Figura 7.23*).

#### 7.12.4 Actividad: Implementar una Clase

El propósito de la implementación de una clase es implementar una clase de diseño en un componente fichero. Esto incluye lo siguiente (*Véase Figura 7.24*):

- Esbozo de un componente fichero que contendrá el código fuente
- Generación de código fuente a partir de la clase de diseño y de las relaciones en que participa.
- Implementación de las operaciones de la clase de diseño en forma de métodos.
- Comprobación de que el componente proporciona las mismas interfaces que la clase de diseño.

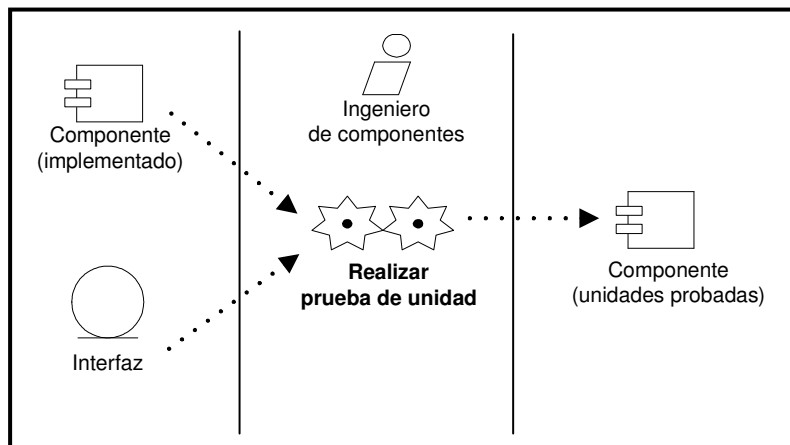


*Figura 7.24 La entrada y el resultado de la implementación de una clase.*

#### 7.12.5 Actividad: Realizar Prueba de Unidad

El propósito de realizar la prueba de unidad es probar los componentes implementados como unidades individuales (*véase Figura 7.25*). Se llevan a cabo los siguientes tipos de pruebas de unidad:

- La prueba de especificación, o “prueba de caja negra”, que verifica el comportamiento de la unidad observable externamente.



*Figura 7.25 La entrada y el resultado de la prueba de una unidad.*

- La prueba de estructura, o “prueba de caja blanca”, que verifica la implementación interna de la unidad.

## 7.13 PRUEBA

Anteriormente se habría descritos las pruebas estáticamente, ahora se verá su comportamiento dinámico.

### 7.13.1 Actividad: Planificar prueba

El propósito de planificar la prueba, es planificar los esfuerzos de prueba en una iteración llevando a cabo las siguientes tareas:

- Describiendo una estrategia de prueba.
- Estimando los requisitos para el esfuerzo de la prueba.
- Planificando el esfuerzo de la prueba.

### 7.13.2 Actividad: Diseñar prueba

Los propósitos de diseñar las pruebas son: Diseño de los casos de prueba de integración para verificar que los componentes interaccionan entre sí de la forma apropiada después de haber sido integrados en una construcción. Diseño de los casos de prueba de sistema instanciados bajo condiciones diferentes (configuraciones hardware, diferentes niveles de carga del sistema, diferentes

números de actores y diferentes tamaños de la base de datos). Diseño de los casos de prueba de regresión. Identificación y estructuración de los procedimientos de prueba. Reutilizar procedimientos de prueba existentes tanto como sea posible, lo que significa que se puede necesitar modificarlos de forma que puedan usarse para especificar cómo realizar un caso de prueba nuevo o cambiado.

### **7.13.3 Actividad: Implementar prueba**

El propósito de la implementación de las pruebas es automatizar los procedimientos de prueba creando componentes de prueba si esto es posible, pues no todos los procedimientos de prueba pueden ser automatizados.

### **7.13.4 Actividad: Realizar pruebas de integración.**

Se llevan a cabo los siguientes pasos:

- Realizar las pruebas de integración relevantes a la construcción realizando los procedimientos de prueba manualmente para cada caso de prueba o ejecutando cualquier componente de prueba que automatice los procedimientos de prueba.
- Comparar los resultados de las pruebas con los resultados esperados.

### **7.13.5 Actividad: Realizar prueba de sistema.**

La prueba de sistema puede empezar cuando las pruebas de integración indican que el sistema satisface los objetivos de calidad de integración fijados en el plan de prueba de la iteración actual; por ejemplo, el 95 por ciento de los casos de prueba de integración se ejecutan con el resultado esperado.

### **7.13.6 Actividad: Evaluar prueba.**

Los diseñadores de pruebas evalúan los resultados de la prueba comparando los resultados obtenidos con los objetivos esbozados en el plan de prueba. Éstos preparan métricas que les permiten determinar el nivel de calidad del software y qué cantidad de pruebas es necesario hacer.

## **PARTE III**

## **MODELOS DE DESARROLLO DEL SOFTWARE**

## **CAPITULO 8.**

### **MODELO DE NEGOCIO Y DE DOMINIO**

En el presente capítulo se comprende el dominio del problema, a través del Modelo de Negocio y de Dominio [LAR99]. Se empieza describiendo una lista de usuarios del negocio, posteriormente los procesos que cada usuario realiza en el negocio. Se muestra un diagrama de Casos de Uso para observar de forma visual los usuarios con los Procesos. Asimismo por cada Proceso se utiliza un diagrama de Actividades para ver las acciones que realiza un usuario al momento de llevar a cabo su Proceso.

#### **8.1 MODELO DE NEGOCIOS.**

Para entender el funcionamiento del Gimnasio “Natural Body Club”, se muestra el Modelo de Negocios que ofrece el Proceso Unificado, a través de Procesos y sus especificaciones respectivas, utilizando un diagrama de actividades por cada caso identificado.

##### **8.1.1 Procesos de Negocio.**

Listado de Procesos y Actores identificados en el Negocio.

##### **Procesos de Negocio**

Inscribir Cliente

Registrar Venta

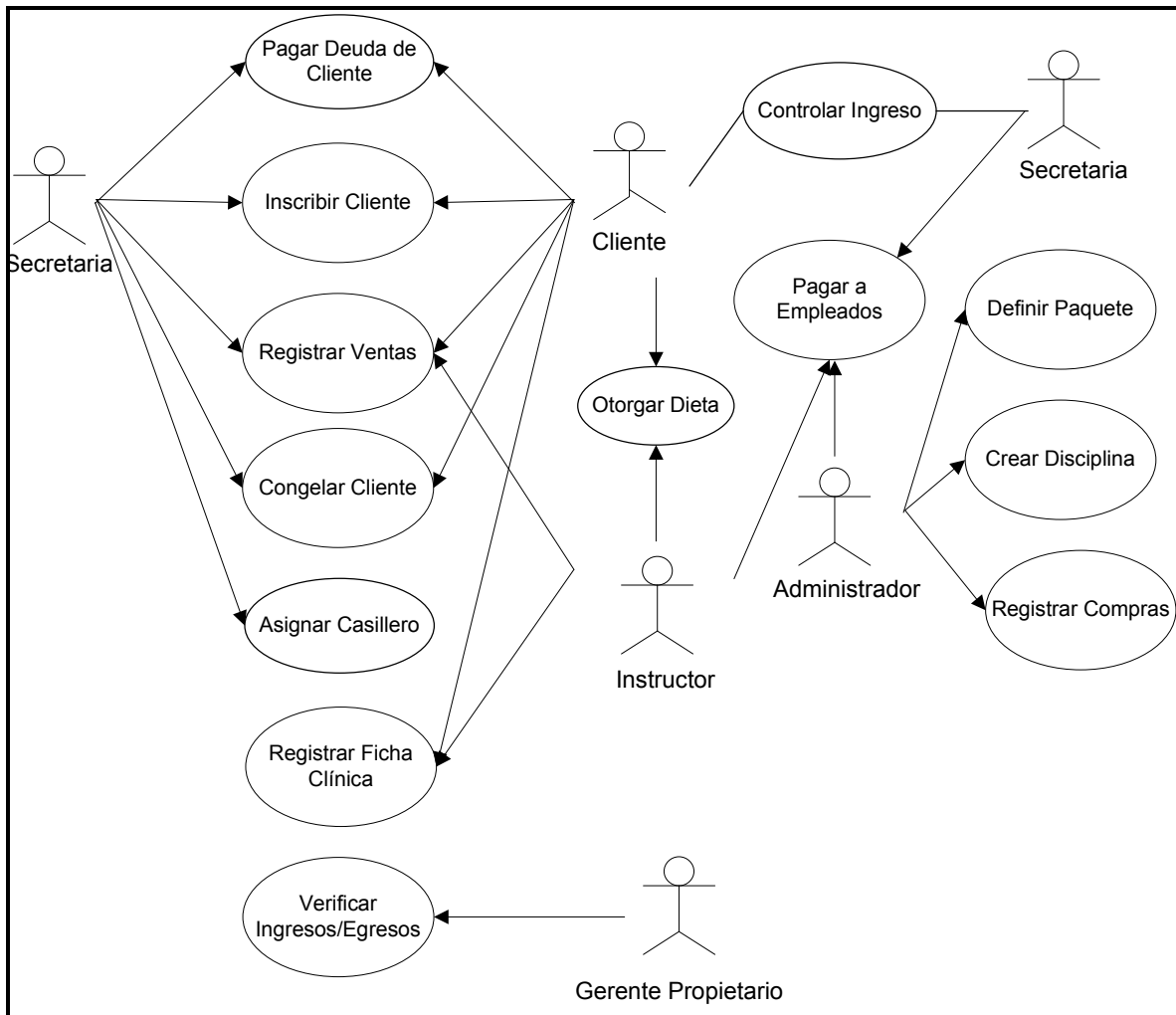
Registrar Compra  
Congelar Cliente  
Controlar Casillero  
Registrar Pago a Empleado  
Registrar Ficha Clínica  
Otorgar Dieta  
Registrar Gastos del Gimnasio  
Definir Paquetes  
Crear Disciplinas  
Verificar Ingresos/Egresos  
Verificar Ficha Clínica  
Controlar Ingreso de Cliente

**Actores.**

Administrador  
Secretaria  
Cliente  
Instructor  
Gerente Propietario

En la figura 8.1 se muestra un diagrama de Casos de Uso para relacionar los Procesos de Negocios, con los actores identificados.





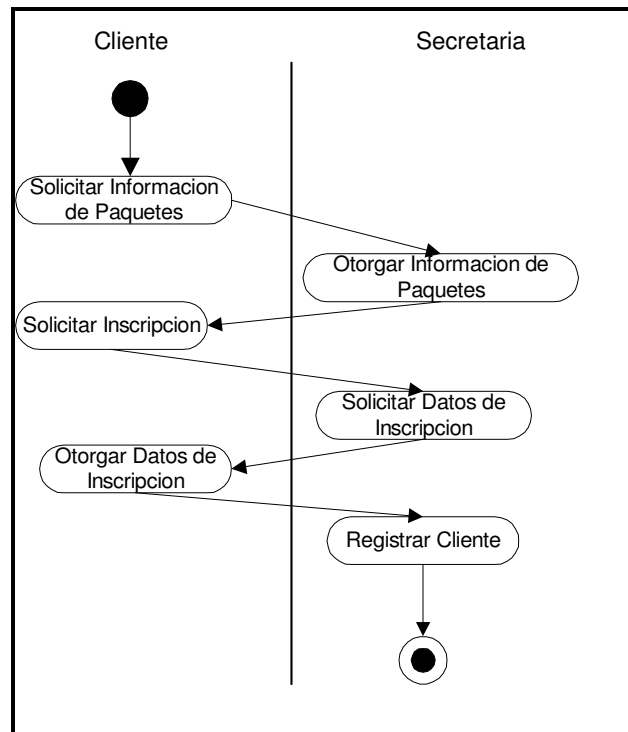
*Figura 8.1 Diagrama de Procesos de Negocio del Dominio*

### 8.1.2 Diagrama de Actividades.

Para la realización de cada caso de uso, se utilizó el diagrama de actividades que ayudará a entender los procesos que existen por cada caso de uso.

**Proceso: Inscribir Cliente**

El proceso, se inicia cuando el posible Cliente quiere inscribirse al Gimnasio para asistir como Cliente regular. En este Caso participan el Cliente y la Recepcionista.



**Figura 8.2 Diagrama de Actividad del Proceso "Inscribir Cliente"**

**Proceso: Registrar Venta**

En este proceso se inicia cuando el Cliente solicita la venta de un producto del snack o del gimnasio a la Secretaria.

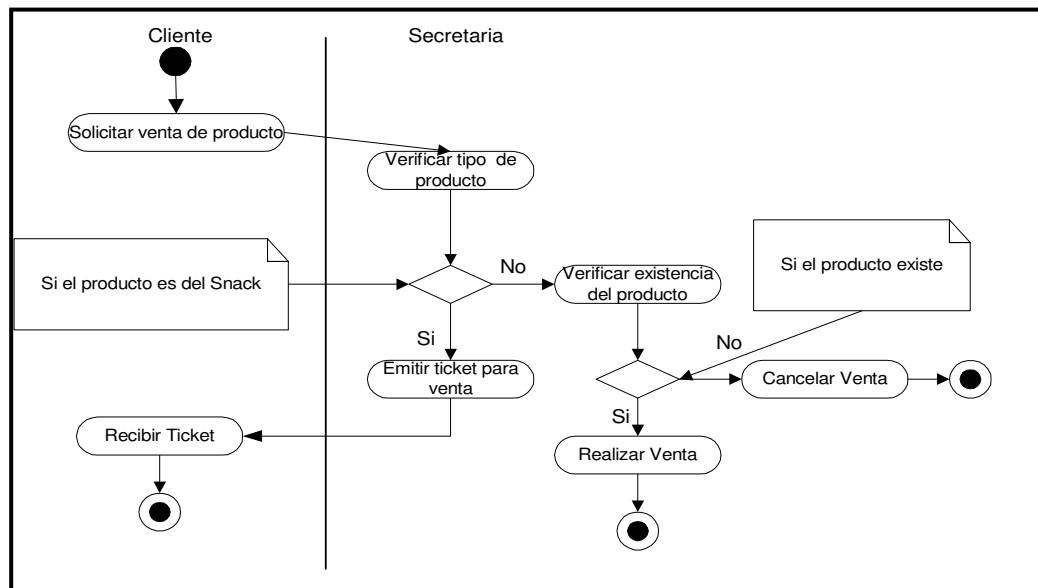


Figura 8.3 Diagrama de Actividad del Proceso "Registrar Venta"

**Proceso: Registrar Compras**

El proceso se inicia cuando el Administrador del gimnasio desea realizar una compra de un producto sea del snack o del gimnasio a uno de sus proveedores.

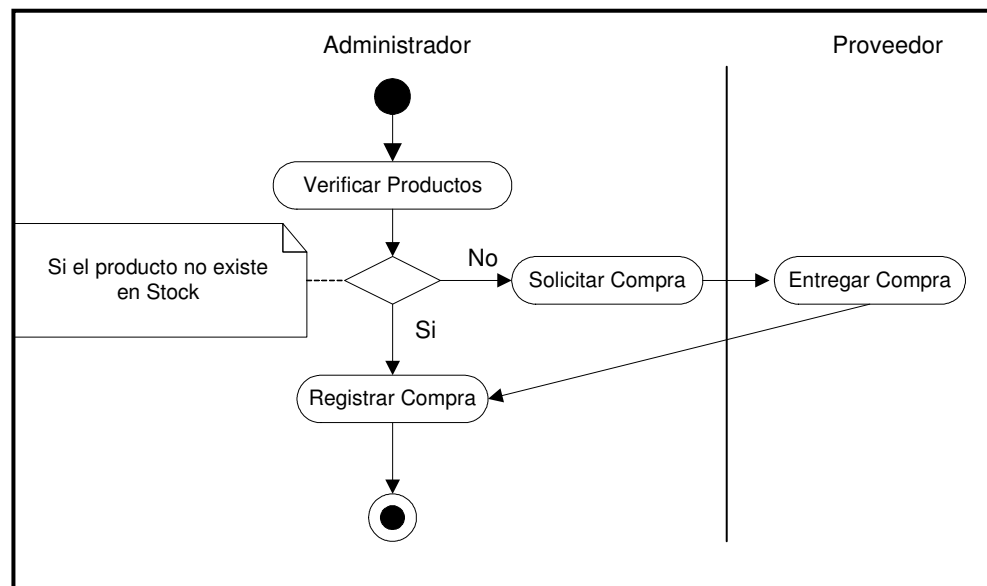
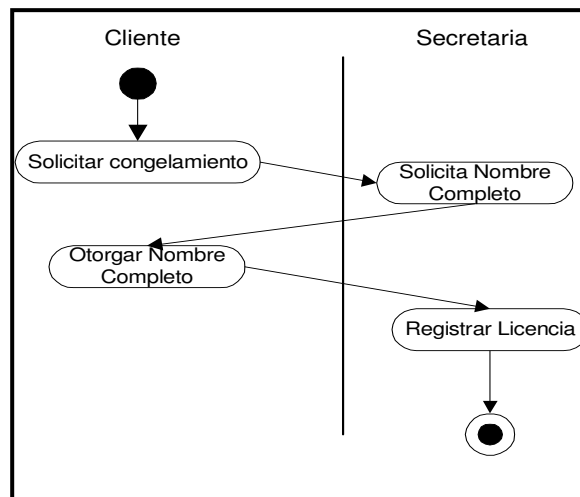


Figura 8.4 Diagrama de Actividad del Proceso "Registrar Compra"

**Proceso: Congelar Inscripción**

Este proceso se da cuando un cliente solicita a la secretaria el congelamiento de su inscripción. Esto para que esos días solicitados tenga licencia de asistencia al gimnasio. La solicitud se la puede realizar por teléfono o de forma personal.



**Figura 8.5 Diagrama de Actividad del Proceso "Congelar Cliente"**

**Proceso: Controlar Casilleros**

En el proceso participa la Secretaria y el Cliente, cuando el Cliente llega al gimnasio solicita un casillero para guardar su indumentaria, la secretaria le entrega una llave y registra el número de casillero entregado.

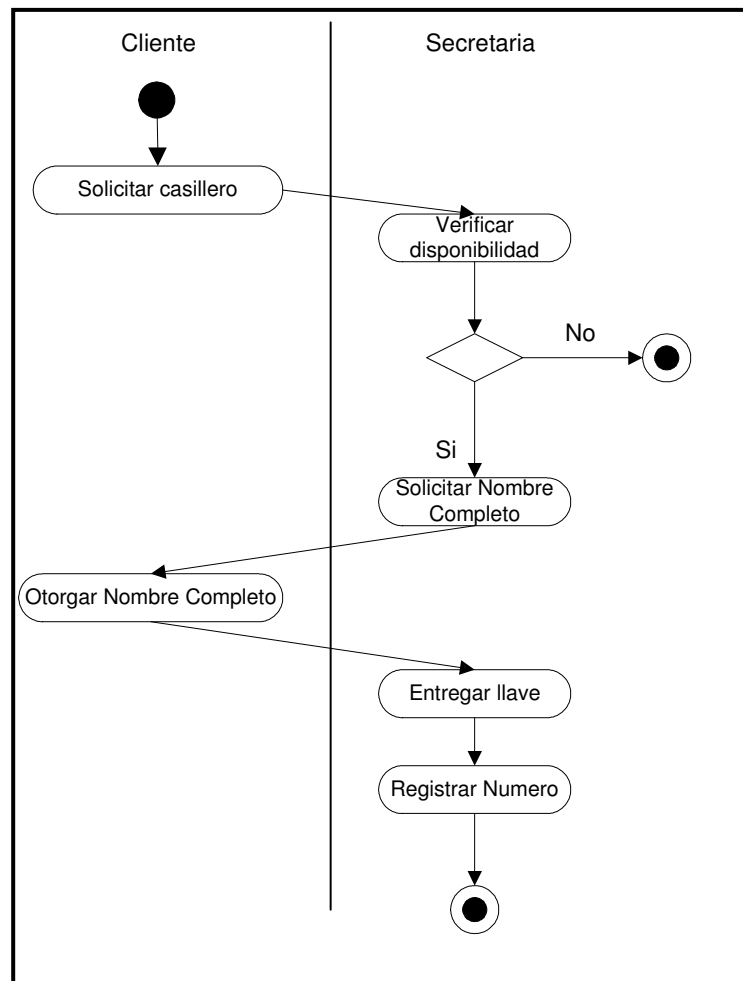
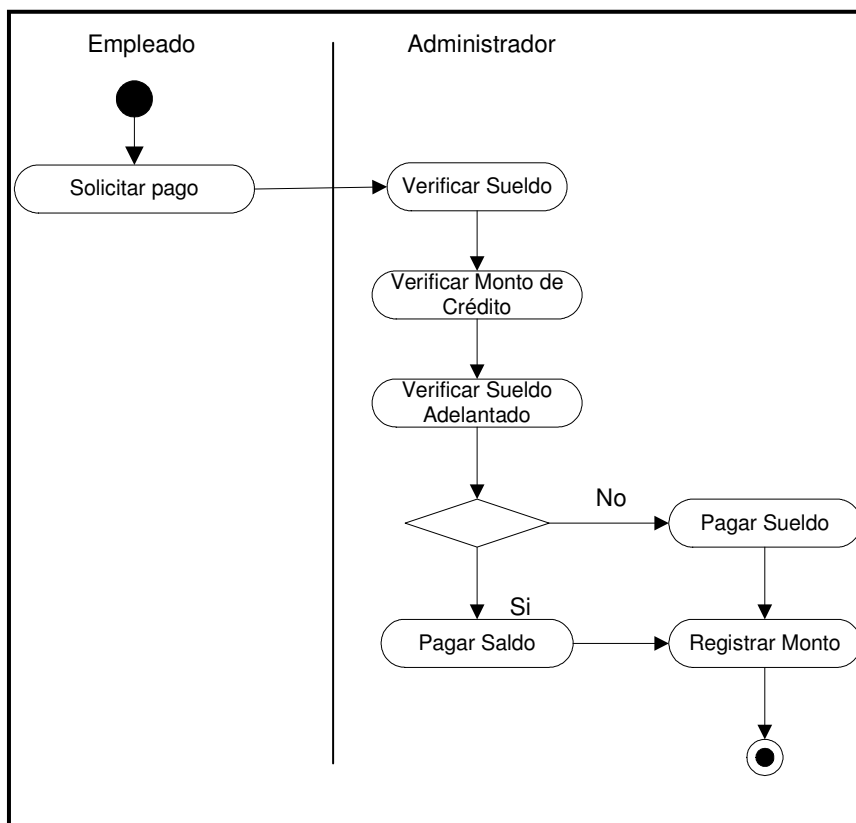


Figura 8.6 Diagrama de Actividad del Proceso “Controlar Casilleros”

**Proceso: Pagar Empleados**

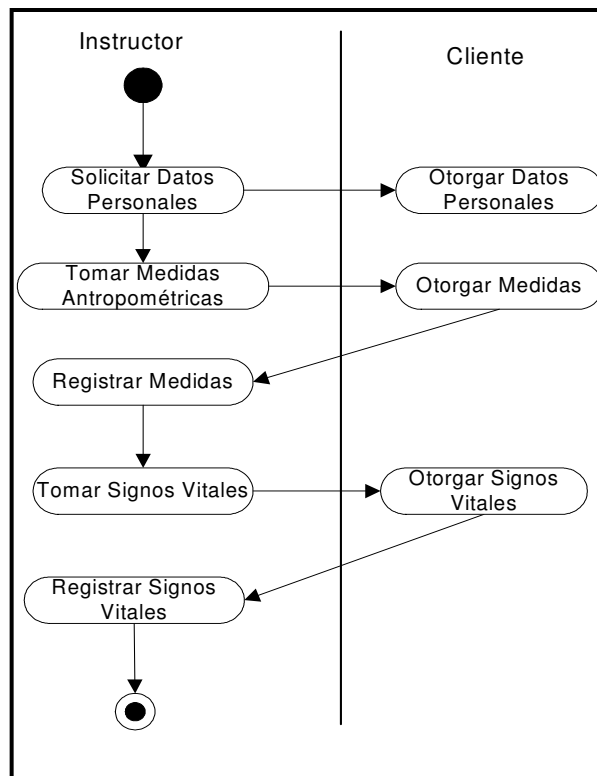
En el proceso participan el Administrador del Gimnasio y el Empleado. Se inicia cuando el Administrador va a realizar los pagos de los sueldos. Se tiene que tomar en cuenta si el pago se lo hace por adelanto o sueldo completo.



**Figura 8.7 Diagrama de Actividad del Proceso “Pagar a Empleado”**

**Proceso: Registrar Ficha Clínica**

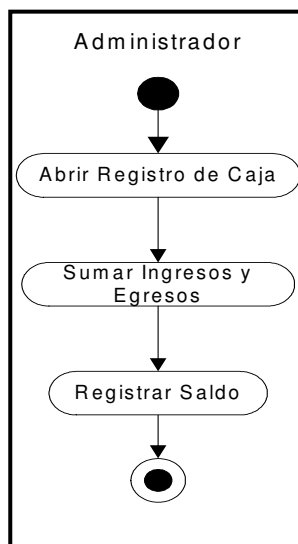
En este proceso participan el Instructor y el Cliente, el primero registra la Ficha Clínica cada cierto tiempo para medir la antropometría y Signos Vitales del Cliente. Esto para tomar acciones sobre las rutinas en sus ejercicios y la dieta asignada.



**Figura 8.8 Diagrama de Actividad del Proceso “Registrar Ficha Clínica”**

### Proceso: Verificar Ingresos/Egresos

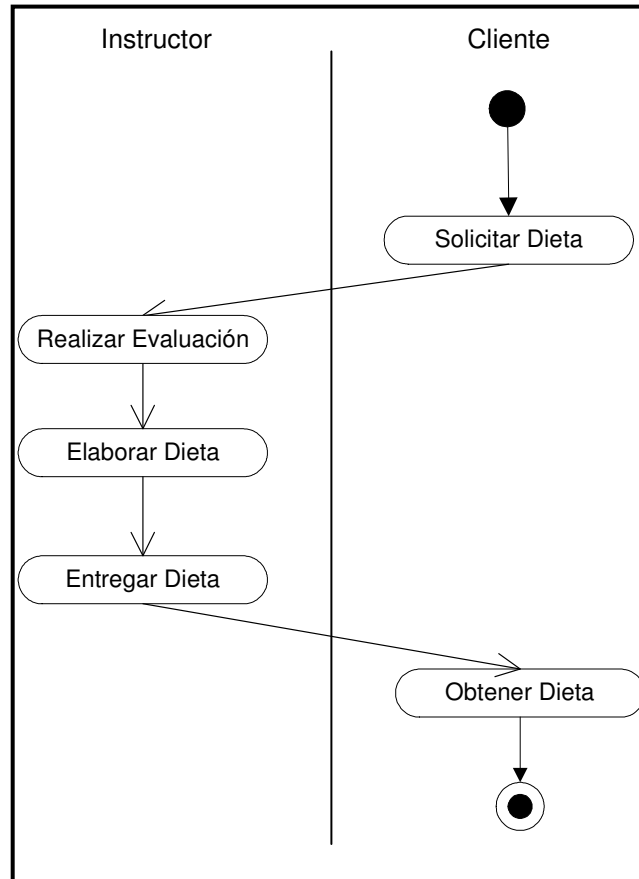
El proceso lo realiza el Administrador para conocer el estado del movimiento de la caja que se ha realizado durante un día, y de esta forma ver si existe saldo positivo o negativo,



**Figura 8.9 Diagrama de Actividad del Proceso “Verificar Ingresos/Egresos”**

**Proceso: Otorgar Dieta.**

Este proceso inicia cuando el Cliente solicita al Entrenador una Dieta para llevar a cabo, sea para aumentar o disminuir de peso.

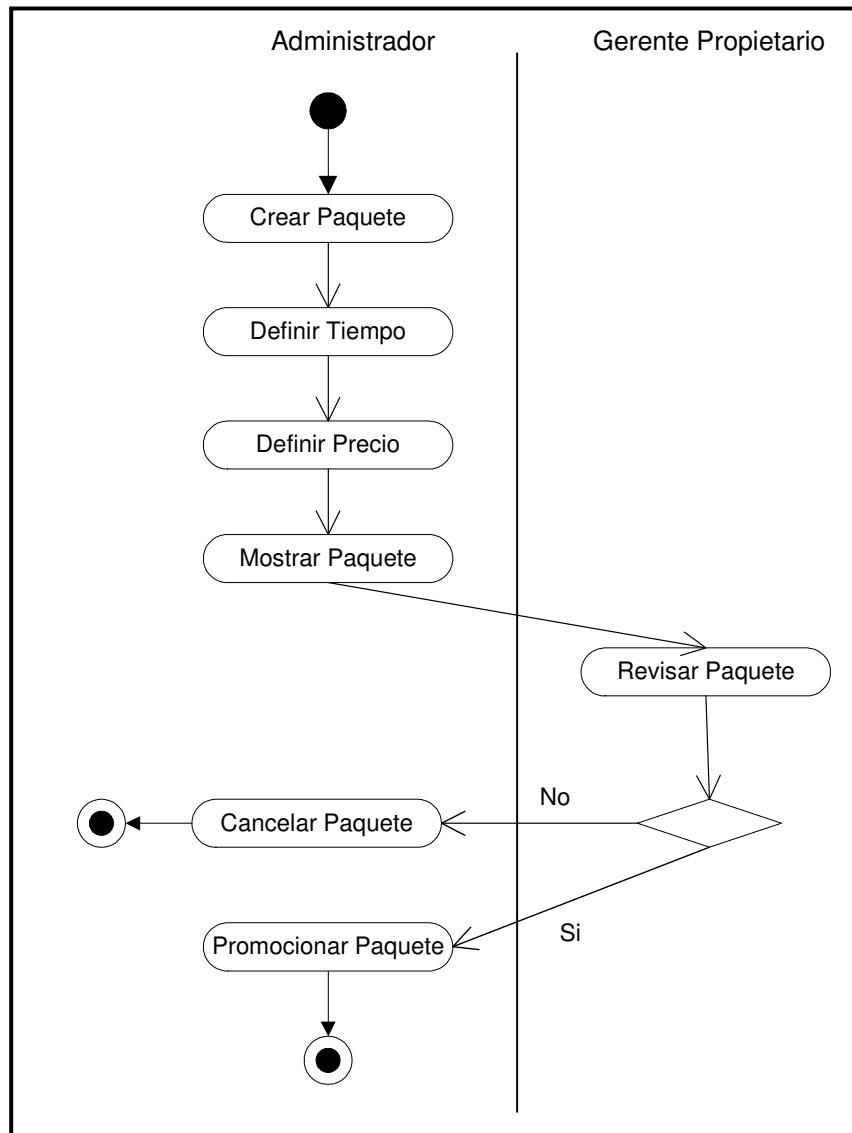


**Figura 8.10 Diagrama de Actividad del Proceso "Otorgar Dieta"**

**Proceso: Definir Paquete.**

Este proceso se inicia cuando el Administrador decide crear y promocionar un Paquete, el cual tiene que ser revisado y aprobado por el Gerente Propietario.





**Figura 8.11** Diagrama de Actividad del Proceso "Definir Paquete"

**Proceso: Pagar Deuda de Cliente.**

El proceso se inicia cuando el Cliente realiza el pago de un saldo de su Inscripción.

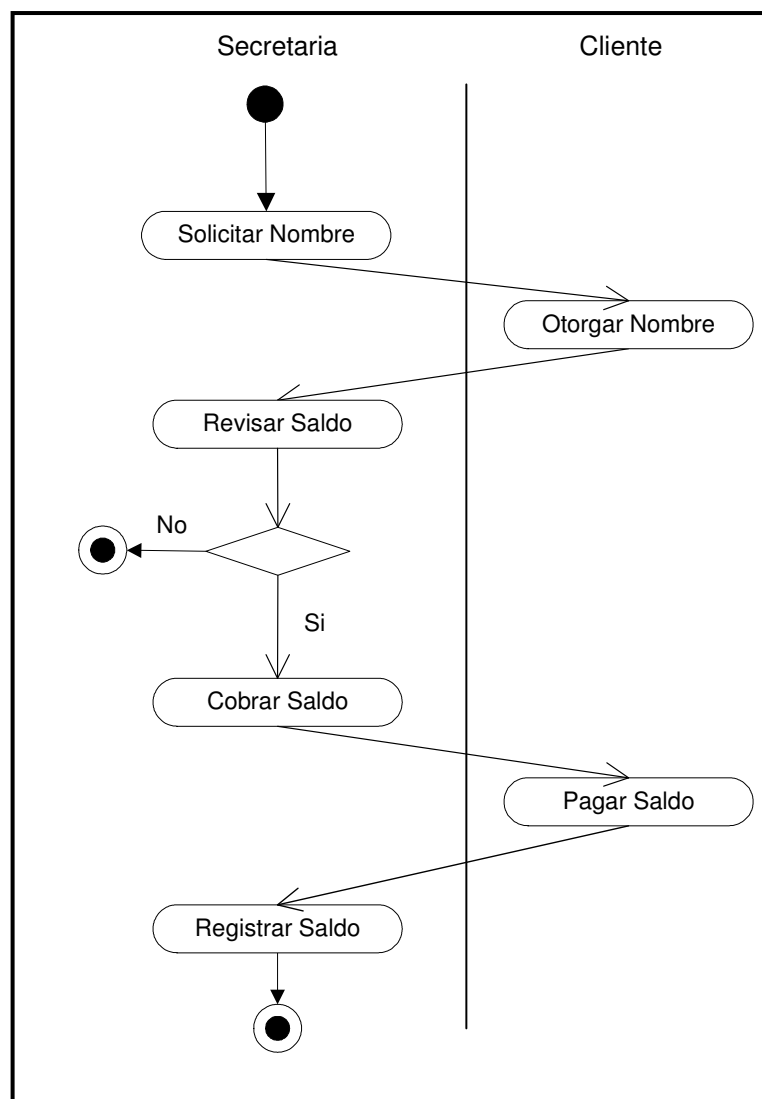
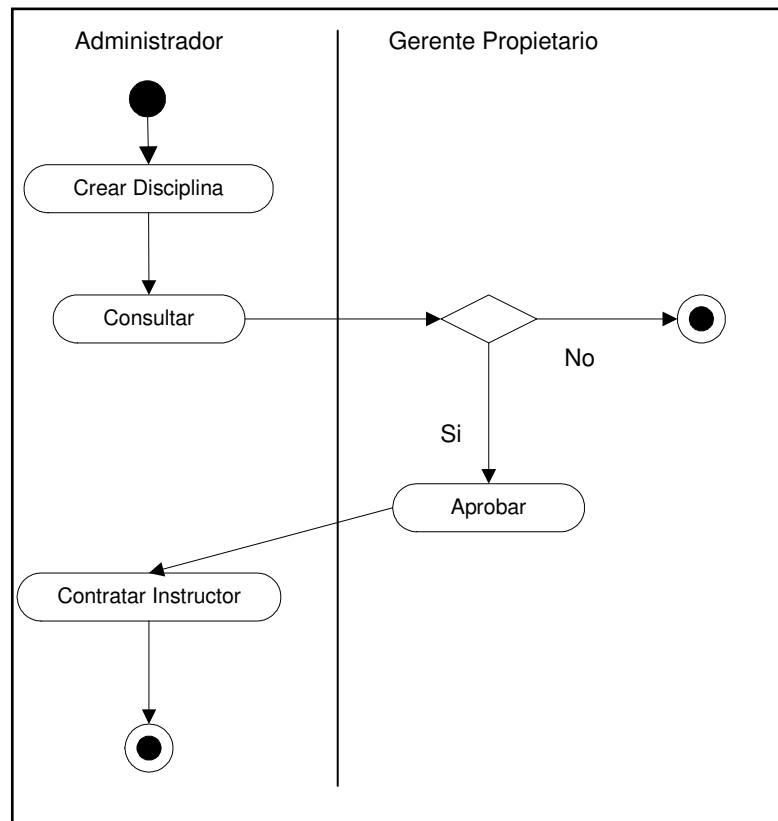


Figura 8.12 Diagrama de Actividad del Proceso "Pagar Deuda de Cliente"

**Proceso: Crear Disciplina**

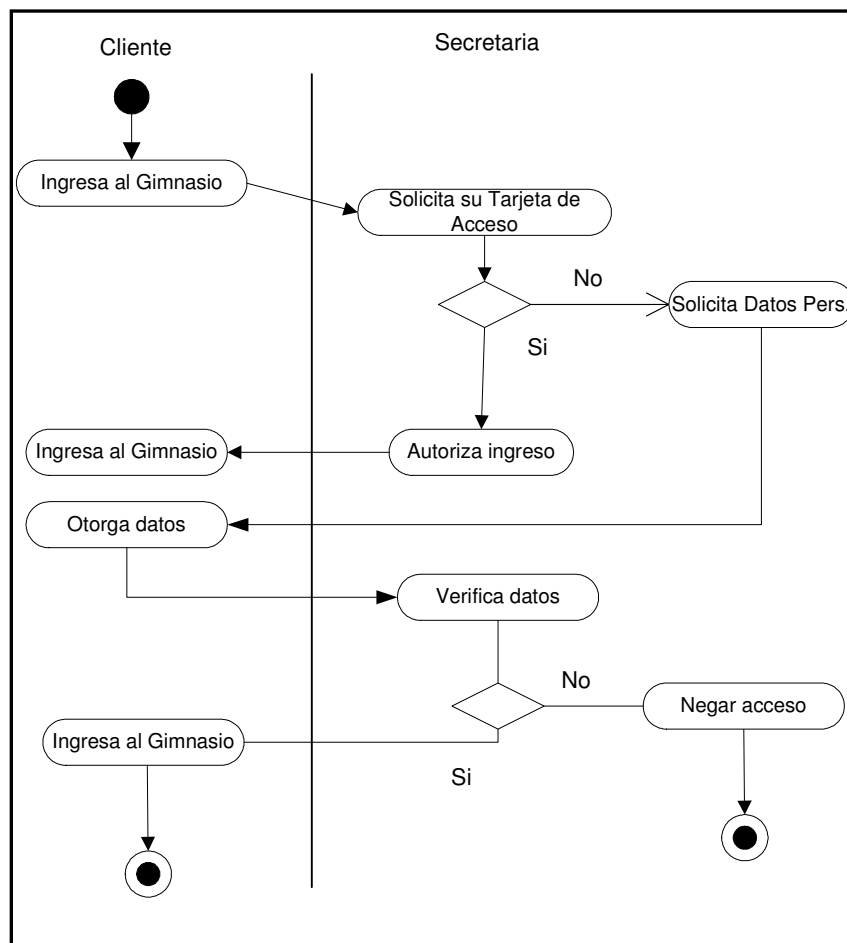
Este proceso se inicia cuando el Administrador ha detectado la necesidad de ofrecer una nueva disciplina a los clientes del Gimnasio. El administrador realiza la consulta respectiva con el Gerente Propietario, quien aprueba o no la creación de la nueva disciplina. Siendo positiva la respuesta, el Administrador inicia la tarea de contratar un Instructor.



**Figura 8.13 Diagrama de Actividad del Proceso "Crear Disciplina"**

**Proceso: Controlar Ingreso de Clientes**

Este proceso se inicia cuando el Cliente llega al Gimnasio. La secretaria le solicita su tarjeta de ingreso, el cliente se la muestra y la secretaria autoriza el ingreso al Gimnasio. Si el cliente se olvida de su tarjeta, la secretaria solicita sus datos al cliente como nombre y apellido, luego verifica si está registrado en los libros de Excel.



**Figura 8.14 Diagrama de Actividad del Proceso “Controlar Ingreso de Clientes”**

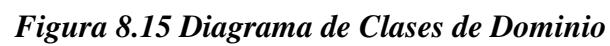
## 8.2 COMPRESIÓN DEL CONTEXTO MEDIANTE UN MODELO DE DOMINIO.

“Un buen modelo de dominio captura las abstracciones y la información esenciales necesarias para entender el dominio en el contexto de los requisitos actuales, y ayuda a la gente a entender el dominio sus conceptos, terminologías y relaciones”.<sup>1</sup>

El modelo de dominio observado en la *Figura 8.15* muestra, mediante un diagrama de clases, las clases más relevantes que se identificaron a partir de los Procesos del Negocio.

---

<sup>1</sup> UML y Patrones, C. Larman



## CAPITULO 9.

### MODELO DE REQUISITOS

#### 9.1 CAPTURA DE REQUISITOS COMO CASOS DE USOS.

En la presente sección se identifican los requisitos para el sistema a través de los Casos de Usos encontrados en el dominio del problema, como primer paso se describen los actores encontrados y su tarea que realiza. Por último se tiene la descripción de cada Caso de Uso, representado con su respectivo diagrama.

##### 9.1.1 Identificación de Actores.

Se identificaron los siguientes actores:

**Actor Cliente.** El cliente representa la persona que es objeto de atención por parte del Gimnasio.

**Actor Administrador.** El administrador es la persona que se encarga de administrar y de informar al Gerente Propietario sobre el estado del Gimnasio.

**Actor Secretaria.** Es la persona encargada de tomar los primeros datos del cliente, asignar un turno y la disciplina del cliente.

**Actor Gerente Propietario.** Es el propietario del Gimnasio y también se encarga de la supervisión de las actividades del Gimnasio.

**Actor Instructor.** Es el encargado de hacer el seguimiento al cliente sobre las actividades físicas que realiza en el gimnasio.

**Actor Proveedor.** Es la empresa que vende productos de Snack como también productos del Gimnasio.

### **9.1.2 Identificación y descripción de los Casos de Uso**

Ahora se muestran los Casos de Uso por cada actor del sistema:

**Secretaria.**

- Abrir Caja
- Cerrar Caja
- Registrar Cliente
- Registrar Ventas
- Registrar Compras
- Inscribir Cliente
- Congelar Cliente
- Pagar Deuda de Cliente
- Registrar Medidas Antropométricas
- Registrar Signos Vitales
- Registrar Antecedentes
- Registrar Dieta
- Buscar Empleado
- Buscar Cliente
- Controlar Asistencia

**Administrador.**

- Pagar a Empleados
- Registrar Productos
- Definir Disciplina



Gerente Propietario

Generar Ingresos y Egresos del Gimnasio

Después de identificar los casos de uso, ahora se mostrará gráficamente cómo queda representado cada uno y además de su descripción completa, indicando quién inicia el caso de uso, las condiciones, el proceso y cómo termina.

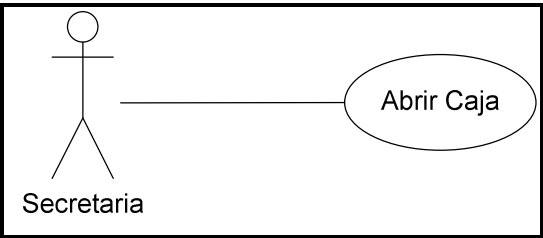


Figura 9.1 Caso de Uso: Abrir Caja

<b>Caso de Uso: Abrir Caja</b>
<b>Precondición:</b> Verificar si la caja ya está abierta.
<b>Curso Normal:</b> <div><div>1. El Caso de Uso se inicia cuando la secretaria empieza a realizar las operaciones del día.</div><div>2. La secretaria ingresa un Monto inicial para movimientos de caja.</div><div>3. La secretaria asigna fecha de inicio de Movimientos.</div><div>4. La secretaria escribe una descripción de apertura de Movimientos.</div><div>5. La secretaria realiza la apertura de la caja.</div></div>
<b>Curso Alterno:</b> <div><div>2. La secretaria asigna un valor negativo, el sistema emite un mensaje de valor no válido.</div><div>3. La secretaria asigna una fecha pasada, con movimientos cerrados, el sistema emite un mensaje de fecha no valida.</div></div>
<b>Postcondición:</b> Caja Abierta

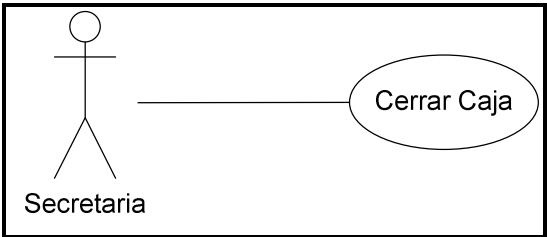


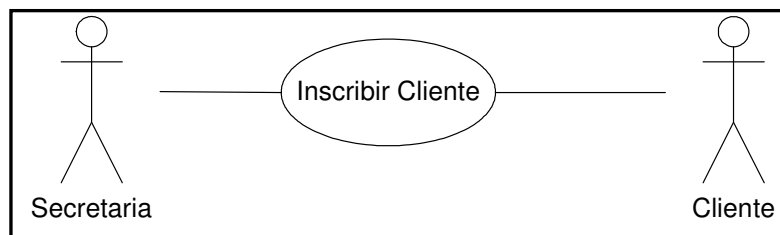
Figura 9.2 Caso de Uso: Cerrar Caja

<b>Caso de Uso: Cerrar Caja</b>
<b>Precondición:</b> Ningún Caso de Uso en realización.
<b>Curso Normal:</b> <div><div>1. El Caso de Uso se inicia cuando se necesita Cerrar los movimientos de caja.</div><div>2. La Secretaria presiona la opción Cerrar Caja.</div><div>3. El sistema muestra una ventana, mostrando el Saldo de las transacciones del día, la fecha de Cierre, la fecha de la Próxima apertura, y el Monto de Apertura.</div><div>4. La secretaria puede modificar la próxima fecha de apertura.</div><div>5. La secretaria presiona Cerrar.</div><div>6. El sistema cambia de estado a Cerrado las transacciones.</div><div>7. El sistema emite un mensaje para Abrir nueva Caja</div><div>8. La secretaria confirma la Apertura de nueva Caja, se inicia el Caso de Uso Abrir Caja.</div></div>
<b>Cursos Alternos.</b> <div><div>8. La Secretaria rechaza la Apertura de nueva Caja, el caso de uso termina.</div></div>
<b>Postcondición:</b> <div><div>El sistema cambia de estado a Cerrado todas las transacciones realizadas desde la fecha de apertura hasta la fecha actual.</div></div>



Figura 9.3 Caso de Uso: Pagar a Empleado

<b>Caso de Uso: Pagar a Empleados</b>
<b>Precondición:</b> Empleado registrado en el sistema.
<b>Curso Normal:</b> <ol style="list-style-type: none"> <li>1. El Administrador inicia el caso de uso cuando va a realizar el pago de sueldo al Empleado.</li> <li>2. El Administrador ingresa a una pantalla de Pagos a Empleados</li> <li>3. El Administrador busca el nombre del Empleado</li> <li>4. El Sistema recupera los datos del Empleado como Sueldo Mensual, Monto de Créditos otorgados.</li> <li>5. El Administrador ingresa el Pago de Sueldo completo.</li> <li>6. El sistema hace la diferencia entre Pago de Sueldo y Sueldo para verificar si el Pago de Sueldo fue completo o solo un Adelanto.</li> <li>7. El caso de uso termina cuando se registra el Pago y se imprime la Boleta de Pago.</li> </ol>
<b>Curso Alterno :</b> <ol style="list-style-type: none"> <li>3. El Administrador no encuentra al Empleado, se inicia el Caso de Uso Registrar Empleado.</li> <li>6. El Administrador selecciona una opción para descontar del Sueldo Mensual el Monto de Créditos.</li> </ol>
<b>Postcondición:</b> Pago registrado

**Figura 9.4 Caso de Uso: Inscribir Cliente**

<b>Caso de Uso: Inscribir Cliente</b>
<b>Precondición:</b> Disciplina definidas, Cliente Registrado
<b>Curso Normal:</b> <ol style="list-style-type: none"> <li>1. El Cliente inicia el caso de uso cuando solicita inscripción a la Secretaria.</li> </ol>

<ol style="list-style-type: none"><li>La Secretaria ofrece una lista de disciplina que incluye sus precios y horarios disponibles.</li><li>El Cliente elige cualquiera de las disciplinas ofrecidas.</li><li>La Secretaria busca al Cliente, se inicia el Caso de Uso Buscar Cliente.</li><li>El Cliente cancela el Pago Total de la Inscripción.</li><li>La Secretaria digita el Pago de Inscripción, según la Disciplina seleccionada.</li><li>El sistema determina el Cambio que se tiene que entregar al Cliente.</li><li>La secretaria registra la inscripción del Cliente.</li></ol>
<b>Curso Alterno</b> <ol style="list-style-type: none"><li>La Secretaria no encuentra al Cliente en el sistema, registra al nuevo Cliente, se inicia el Caso de Uso Registrar Cliente.</li><li>El Cliente puede cancelar el Pago de forma parcial su Inscripción.</li></ol>
<b>Postcondición:</b> Cliente inscrito.

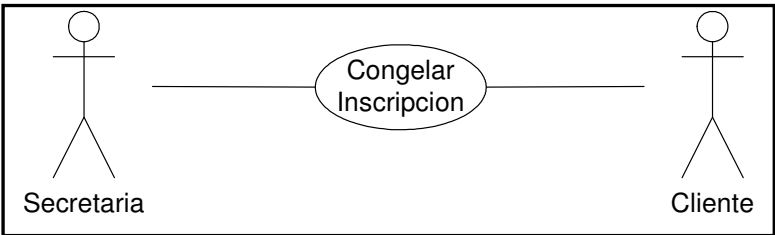
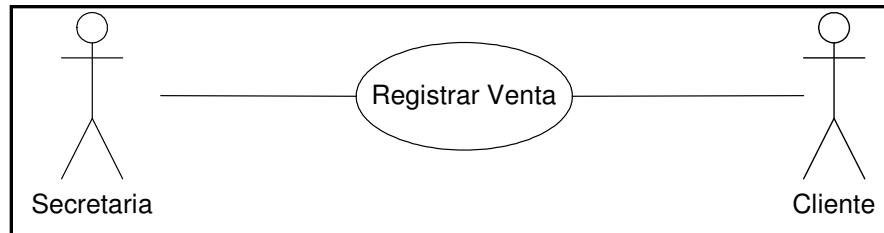


Figura 9.5 Caso de Uso: Congelar Inscripción

<b>Caso de Uso: Congelar Inscripción</b>
<b>Precondición:</b> Inscripción Registrada.
<b>Curso Normal:</b> <ol style="list-style-type: none"><li>Cliente inicia el caso de uso cuando solicita congelar la Inscripción.</li><li>La Secretaria busca al Cliente, se inicia el Caso de Uso Buscar Cliente.</li><li>El Sistema recupera los datos Personales, Monto de Pago, Fecha de Inscripción y Disciplina Inscrita.</li><li>La Secretaria presiona la opción Congelar.</li><li>El Sistema cambia de Estado al Cliente de Activo a Congelado y el Caso de Uso termina.</li></ol>
<b>Cursos Alternos:</b> <ol style="list-style-type: none"><li>El Sistema verifica si el Estado del Cliente no está Congelado o como Pasivo, emite un mensaje si esta en cualquiera de los dos estados.</li></ol>

**Postcondición:** Cliente Congelado.



**Figura 9.6 Caso de Uso: Registrar Venta**

**Caso de Uso: Registrar Venta**

**Precondición:** Producto registrado, Cliente registrado.

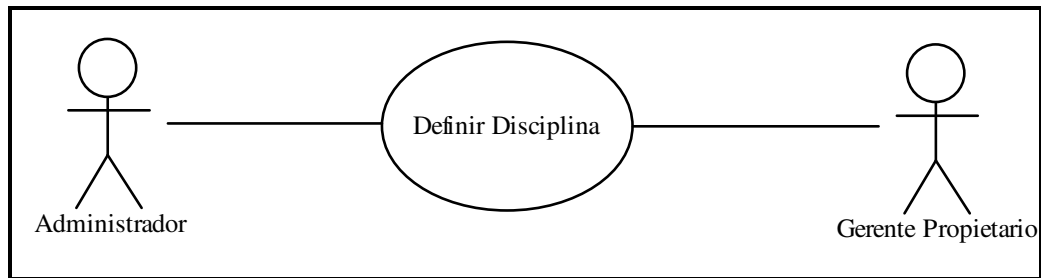
**Curso Normal:**

1. El Cliente inicia el caso de uso cuando solicita un Producto de Gimnasio o un Producto de snack
2. La Secretaria busca el nombre del Cliente.
3. El sistema recupera los datos del Cliente.
4. La Secretaria busca el producto para Vender
5. El sistema recupera el Nombre y Precio de Venta del producto.
6. La Secretaria introduce la cantidad de Productos a vender.
7. La Secretaria agrega el Producto al detalle de la Venta.
8. El sistema calcula el Total a Pagar.
9. La secretaria introduce el Monto de Pago y almacena la Venta.
10. El sistema registra la Venta.

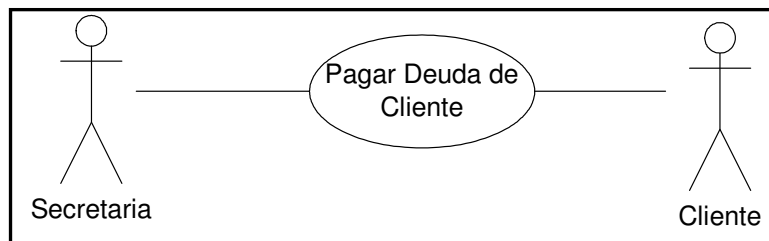
**Cursos Alternos:**

2. El nombre del cliente no se encuentra registrado, se inicia el Caso de Uso Registrar Cliente.
4. El Producto no se encuentra registrado, se inicia el Caso de Uso, Registrar Producto.
6. La cantidad de Ventas no está disponible para vender, el sistema emite un mensaje de “cantidad no disponible”
7. La Secretaria puede quitar del Detalle de Venta un producto.

**Postcondición:** Venta registrada.

**Figura 9.7 Caso de Uso: Definir Disciplina**

<b>Caso de Uso: Definir Disciplina</b>
<b>Precondición:</b> Ninguna
<b>Curso Normal:</b> <ol style="list-style-type: none"> <li>1. El Administrador inicia el Caso de Uso para definir Disciplinas para ofrecer a los Clientes que quieran inscribirse.</li> <li>2. El sistema muestra una ventana para introducir los datos necesarios.</li> <li>3. El Administrador digita el nombre de la disciplina, ingresa el horario en que se dictará la disciplina y define el Precio de la disciplina.</li> <li>4. La disciplina se registra y el caso de uso termina.</li> </ol>
<b>Postcondición:</b> Disciplina registrada.

**Figura 9.8 Caso de Uso: Pagar Deuda de Cliente**

<b>Caso de Uso: Pagar Deuda de Cliente</b>
<b>Curso Normal:</b> <ol style="list-style-type: none"> <li>1. El caso de uso se inicia cuando el cliente paga una deuda sea de inscripción o de ventas que se le realizaron.</li> </ol>

2. La secretaria busca el nombre del Cliente, se inicia el Caso de uso Buscar Cliente.
3. El sistema recupera el Monto de Deuda que tiene el cliente de los productos comprados o deuda de inscripción.
4. La secretaria introduce el monto que el cliente va a pagar.
5. El sistema calcula el Saldo de la Deuda.
6. La Secretaria registra el Pago y el caso de uso termina.

**Postcondición:** Pago registrado.

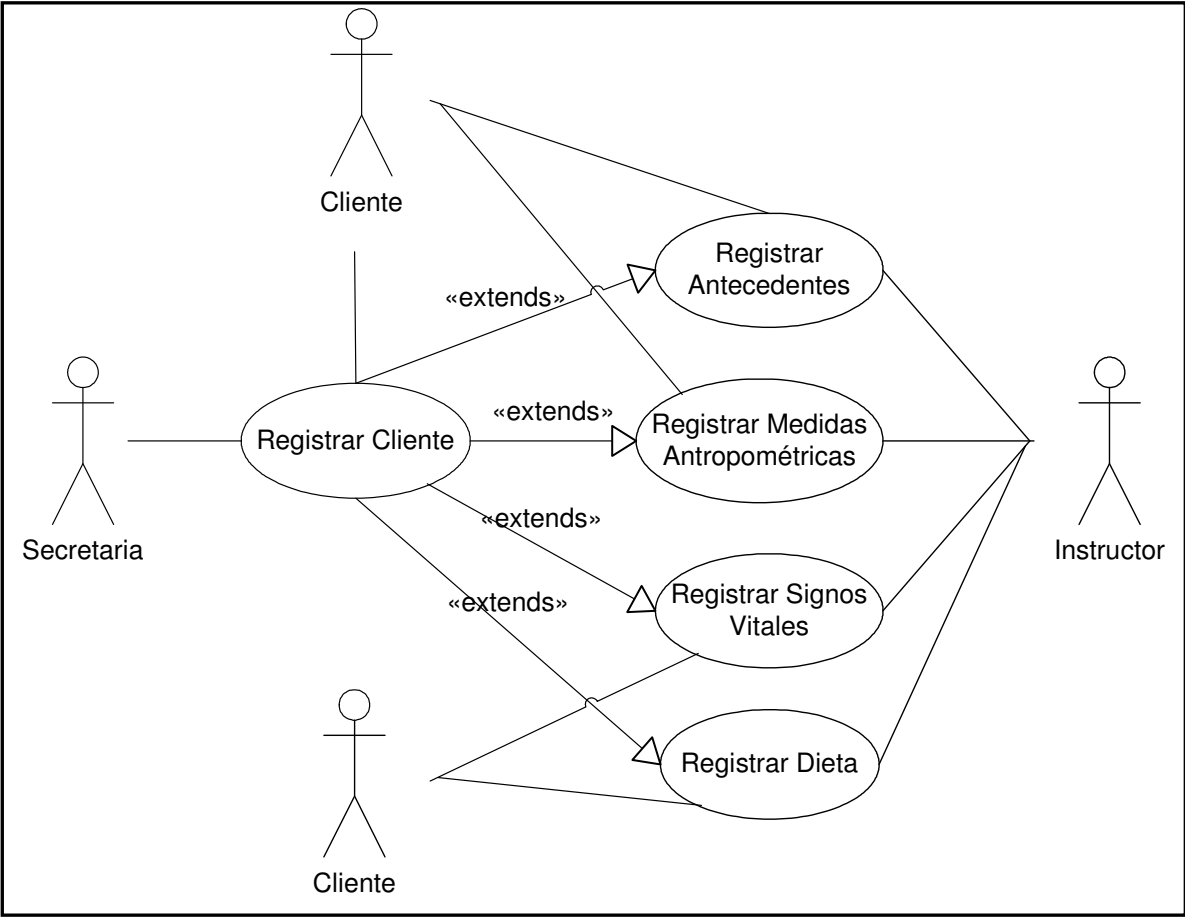


Figura 9.9 Caso de Uso: Registrar Cliente

**Caso de Uso: Registrar Cliente**

**Curso Normal:**

1. El caso de uso se inicia cuando la secretaria registra a una persona como Cliente del Gimnasio.

<ol style="list-style-type: none"> <li>La secretaria solicita los datos de la persona como: Nombre, Carnet de Identidad, Dirección, Teléfono.</li> <li>La secretaria registra en el sistema al Cliente.</li> <li>El Cliente puede solicitar que le den una Dieta, se inicia el caso de uso <b>Registrar Dieta</b>, o también el Instructor puede solicitar los antecedentes, se inicia el Caso de Uso <b>Registrar Antecedentes</b>, o realizar una evaluación de los signos vitales, se inicia el caso de uso <b>Registrar Signos Vitales</b> o tomar las <b>Medidas Antropométricas</b>.</li> </ol>
<b>Sección Secundaria</b>
<b>Caso de Uso: Registrar Dieta</b>
<b>Curso Normal:</b> <ol style="list-style-type: none"> <li>El caso de uso se inicia cuando el Cliente solicita una Dieta a su Instructor.</li> <li>La secretaria ingresa a una pantalla para registrar la Dieta.</li> <li>La secretaria introduce la Dieta asignada por el Instructor.</li> <li>La secretaria registra la Dieta.</li> </ol>
<b>Caso de Uso: Registrar Signos Vitales</b>
<b>Curso Normal:</b> <ol style="list-style-type: none"> <li>El caso de uso se inicia cuando el Instructor, toma los signos vitales del cliente.</li> <li>La secretaria ingresa los datos de los signos vitales.</li> <li>La secretaria registra los signos vitales.</li> </ol>
<b>Postcondición:</b> Signos Vitales registrados
<b>Caso de Uso: Registrar Medidas Antropométricas</b>
<b>Curso Normal:</b> <ol style="list-style-type: none"> <li>El caso de uso se inicia cuando el Instructor, toma las medidas antropométricas del cliente.</li> <li>La secretaria ingresa los datos de las medidas antropométricas.</li> <li>La secretaria registra las medidas antropométricas.</li> </ol>
<b>Postcondición:</b> Medidas Antropométricas registradas
<b>Caso de Uso: Registrar Antecedentes</b>
<b>Curso Normal:</b> <ol style="list-style-type: none"> <li>El caso de uso se inicia cuando el Instructor, evalúa si el cliente tiene algún antecedente</li> </ol>



Patológico, Cirugía, etc.
2. La secretaria ingresa los datos de los Antecedentes.
3. La secretaria registra los Antecedentes.
<b>Postcondición:</b> Antecedentes registrados.

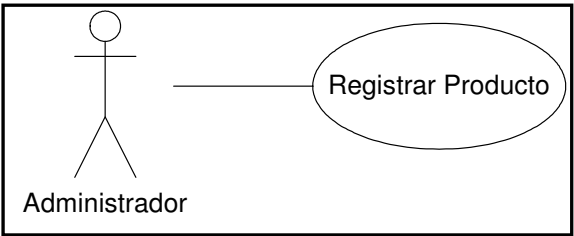


Figura 9.10 Caso de Uso: Registrar Producto

<b>Caso de Uso: Registrar Producto</b>
<b>Curso Normal:</b> <ul style="list-style-type: none"><li>1. El caso de uso se inicia cuando el administrador realiza el registro de un producto.</li><li>2. El sistema muestra una pantalla para registrar el nombre, el precio unitario.</li><li>3. El administrador ingresa los datos del producto y registra el Producto.</li><li>4. El sistema registra el Producto.</li></ul>
<b>Postcondición:</b> Producto Registrado.

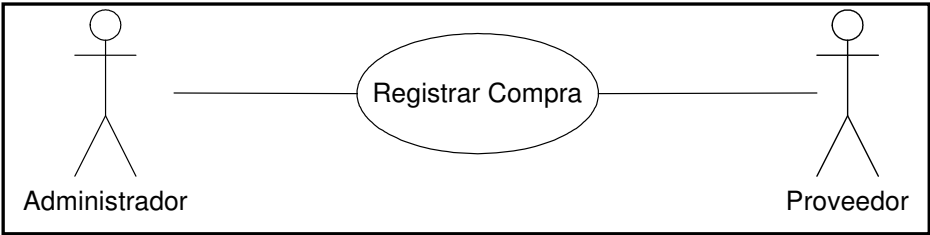


Figura 9.11 Caso de Uso: Registrar Compra

<b>Caso de Uso: Registrar Compra</b>
<b>Precondición:</b> Producto Registrado.
<b>Curso Normal:</b> <ul style="list-style-type: none"><li>1. El caso de uso se inicia cuando el administrador realiza la compra de un producto.</li></ul>

2. El sistema muestra una pantalla para registrar la compra.
3. El administrador busca el producto a comprar.
4. El sistema recupera el nombre del producto.
5. El administrador ingresa la cantidad de compra y el precio de compra
6. El administrador agrega el Producto al detalle de la compra.
7. El sistema calcula el total a pagar de Compras.
8. El administrador registra la Compra.

**Curso Alterno:**

3. El administrador no encuentra el producto, se inicia el caso de uso Registrar Producto.

6. El administrador puede Quitar del detalle a un producto.

**Postcondición:** Compra Registrada

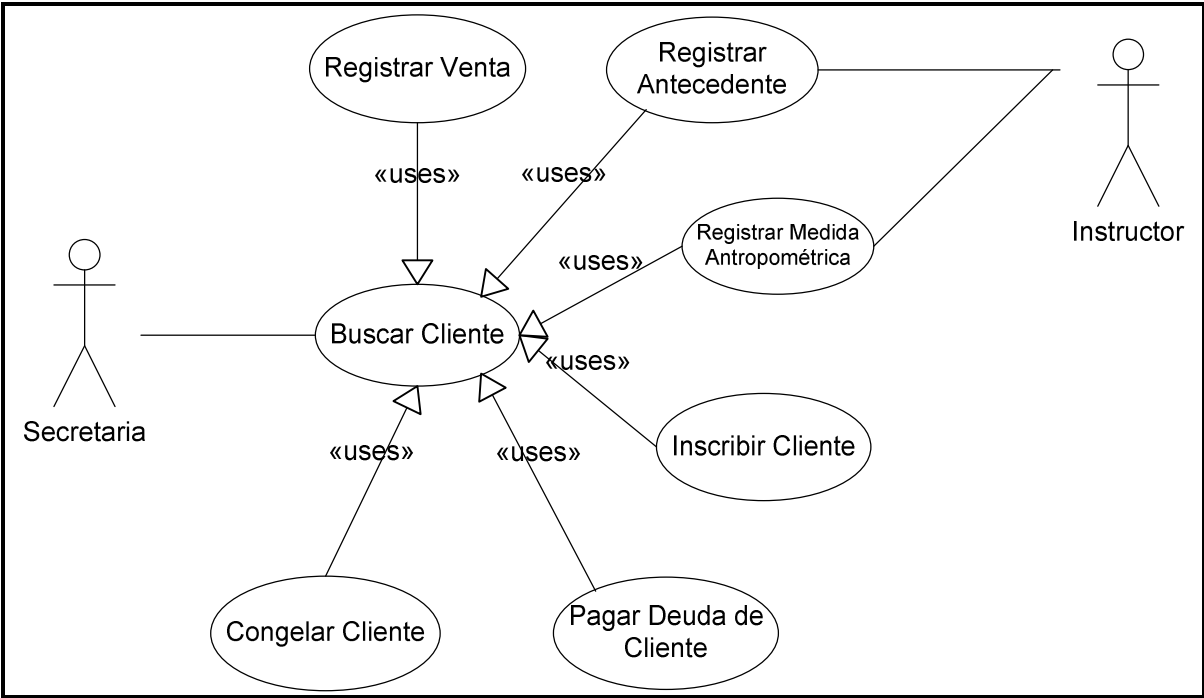
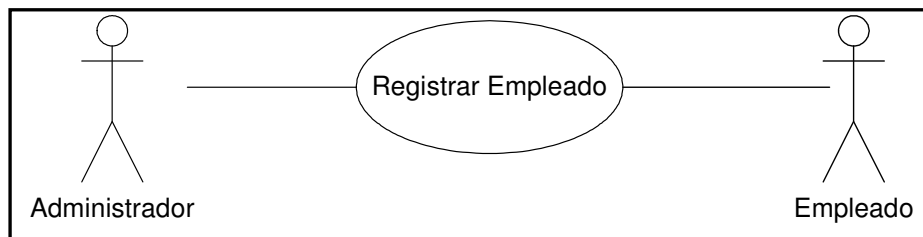


Figura 9.12 Caso de Uso: Buscar Cliente

<b>Caso de Uso: Buscar Cliente</b>
<b>Precondición:</b> Criterio de búsqueda introducido.
<b>Curso Normal:</b> <ol style="list-style-type: none"> <li>1. El caso de uso se inicia cuando la secretaria busca un Cliente, para realizar cualquiera de los casos de uso.</li> <li>2. El sistema muestra una pantalla para buscar por algunos criterios como: Carnet de Identidad, Apellidos.</li> <li>3. La secretaria ingresa los datos de búsqueda y presiona Buscar.</li> <li>4. El sistema busca el Cliente en el sistema, según los parámetros elegidos y despliega el resultado.</li> <li>5. La secretaria elige al Cliente y el Caso de Uso termina.</li> </ol>
<b>Curso Alterno:</b> <ol style="list-style-type: none"> <li>3. El sistema puede no encontrar al Cliente de acuerdo al criterio introducido, la secretaria introduce un criterio de búsqueda diferente y presiona Buscar.</li> </ol>
<b>Postcondición:</b> Cliente buscado.

**Figura 9.13 Caso de Uso: Registrar Empleado**

<b>Caso de Uso: Registrar Empleado</b>
<b>Precondición:</b> Criterio de búsqueda introducido.
<b>Curso Normal:</b> <ol style="list-style-type: none"> <li>1. El caso de uso se inicia cuando el Administrador incorpora nuevo personal al Gimnasio, donde éste tiene que ser registrado.</li> <li>2. El sistema muestra una pantalla para registrar los datos del Empleado como ser: nombre, apellido, dirección, teléfono, sueldo mensual y horario de asistencia.</li> <li>3. El Administrador ingresa los datos del Empleado.</li> </ol>

4. El sistema registra los datos y el Caso de Uso termina.
<b>Postcondición:</b> Empleado Registrado.

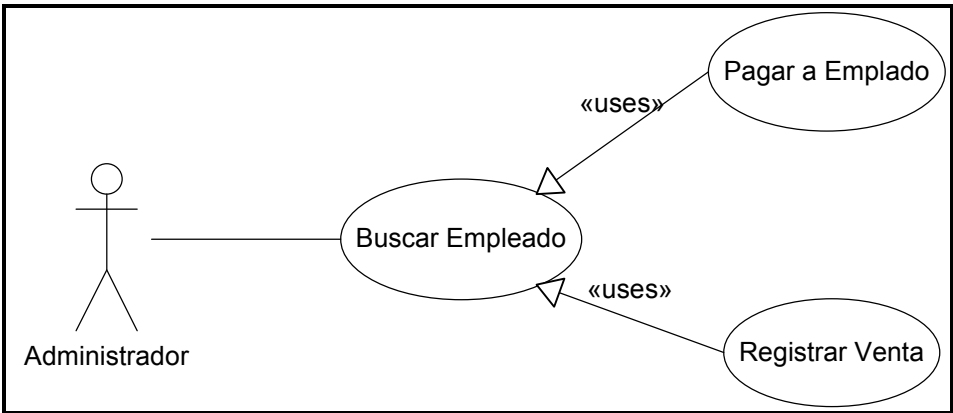


Figura 9.14 Caso de Uso: Buscar Empleado

<b>Caso de Uso: Buscar Empleado</b>
<b>Precondición:</b> Criterio de búsqueda introducido.
<b>Curso Normal:</b> <ul style="list-style-type: none"><li>1. El caso de uso se inicia cuando el Administrador busca un Empleado, para realizar cualquiera de los casos de uso.</li><li>2. El sistema muestra una pantalla para buscar por algunos criterios como: Carnet de Identidad, Apellidos.</li><li>3. El Administrador ingresa los datos de búsqueda y presiona Buscar.</li><li>4. El sistema busca el Empleado en el sistema, según los parámetros elegidos y despliega el resultado.</li><li>5. El Administrador elige al Empleado y el Caso de Uso termina.</li></ul>
<b>Curso Alterno:</b> <ul style="list-style-type: none"><li>4. El sistema puede no encontrar al Empleado de acuerdo al criterio introducido, el Administrador introduce un criterio de búsqueda diferente y presiona Buscar.</li></ul>
<b>Postcondición:</b> Empleado buscado.

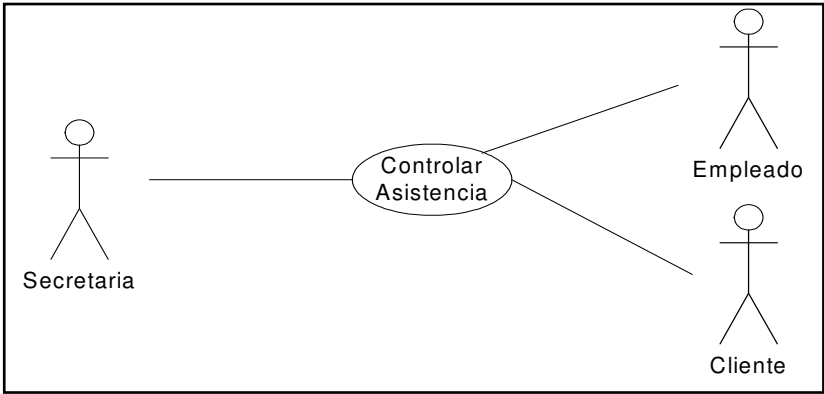


Figura 9.15 Caso de Uso: Controlar Asistencia

<b>Caso de Uso: Controlar Asistencia</b>
<b>Precondición:</b> Cliente Registrado, Cliente Inscrito o Empleado Registrado.
<b>Curso Normal:</b> <ul style="list-style-type: none"><li>1. El caso de uso se inicia cuando el cliente o el empleado llega al Gimnasio.</li><li>2. La secretaria solicita el código del cliente o del empleado.</li><li>3. El cliente o empleado entrega el código a la secretaria.</li><li>4. La secretaria introduce el código en una pantalla de control de ingreso y presiona el botón Ingresar.</li><li>5. El sistema busca un cliente o empleado que coincida con el código introducido, luego verifica si tiene una inscripción y si está en el horario que tiene que ingresar.</li><li>6. El sistema emite una autorización de acceso al cliente o empleado si tiene inscripción y si está en el horario establecido.</li></ul>
<b>Curso Alterno 1:</b> <ul style="list-style-type: none"><li>3. El cliente o empleado puede no recordar su código, y da su nombre y su apellido.</li><li>4. La secretaria busca por su nombre en la pantalla de control de ingreso y presiona el botón Ingresar.</li></ul>
<b>Curso Alterno 2:</b> <ul style="list-style-type: none"><li>5. El sistema puede no encontrar al cliente o empleado, puede que su inscripción haya expirado o su horario establecido no es el asignado.</li><li>6. El sistema emite un mensaje de advertencia.</li></ul>
<b>Postcondición:</b> cliente o empleado autorizado.

## **CAPITULO 10.**

### **MODELO DE ANÁLISIS**

El presente capítulo tiene por objetivo realizar el análisis del sistema, de acuerdo a los requisitos encontrados en el dominio del problema.

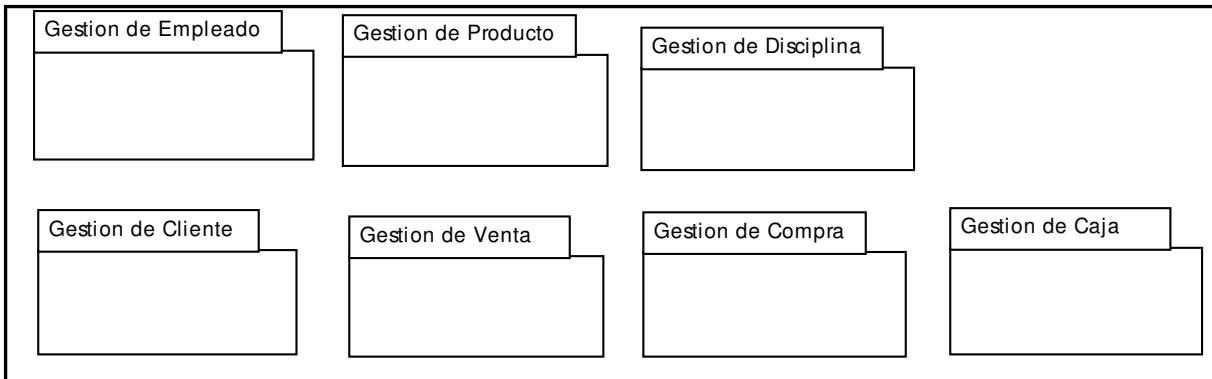
Primeramente se realiza un análisis de la arquitectura, encontrando paquetes para los casos de uso, luego se analizan los casos de uso identificando todos los objetos que interactúan en cada caso a través de un diagrama de colaboración.

#### **10.1 ANÁLISIS DE LA ARQUITECTURA**

En el análisis de la arquitectura se identifican los paquetes del análisis, tomando como criterio los casos de uso requeridos para dar soporte a un determinado proceso de negocio.

##### **10.1.1 Identificación de Paquetes del Análisis**

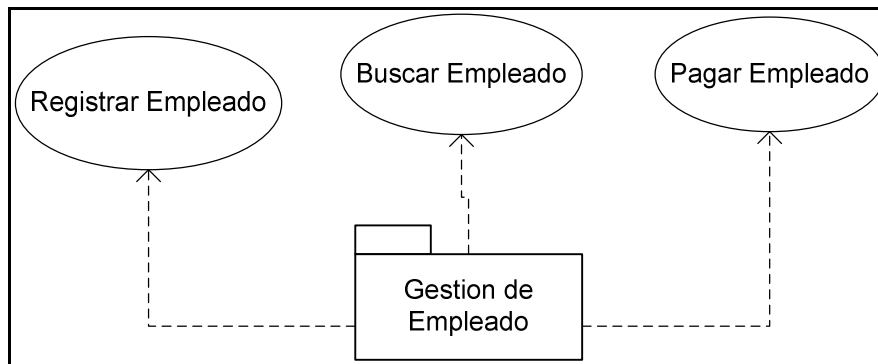
Se identificaron siete paquetes en el análisis: Gestión de Empleado, Gestión de Producto, Gestión de Disciplina, Gestión de Cliente, Gestión de Venta, Gestión de Compra y Gestión de Caja. Véase la figura 10.1

*Figura 10.1 Paquetes del Análisis*

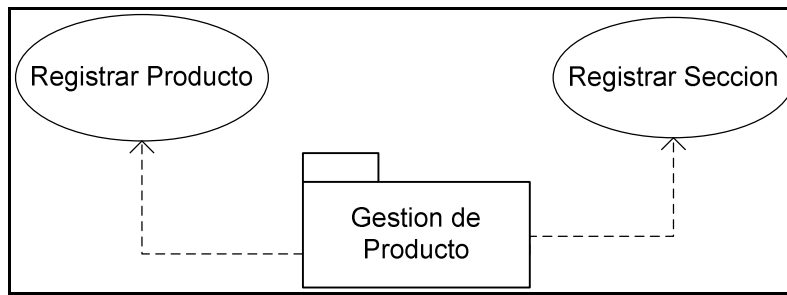
### 10.1.1 Detallando los Paquetes del Análisis

Para comprender el contenido de cada paquete encontrado, se muestra la relación que tienen con sus respectivos Casos de Uso:

**Paquete “Gestión de Empleado”:** Comprende los casos de usos que dan soporte al Empleado que son: Registrar Empleado, Buscar Empleado, Pagar Empleado.

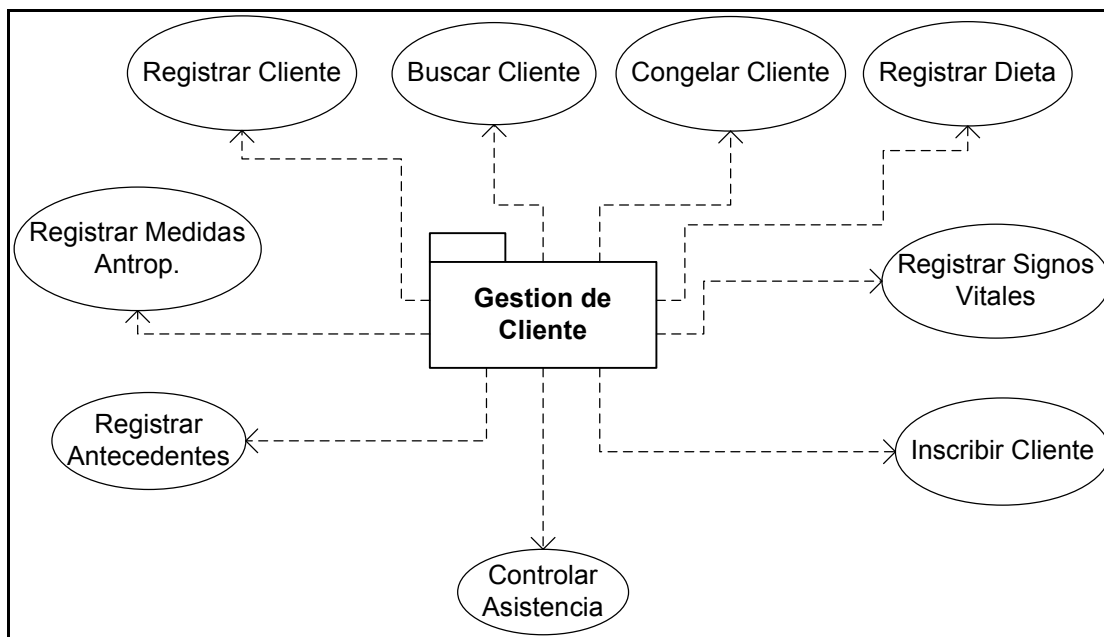
*Figura 10.2 Paquetes del Análisis “Gestión de Empleado”*

**Paquete “Gestión de Producto”:** Comprende los casos de uso: Registrar Producto, Registrar Sección.



**Figura 10.3 Paquetes del Análisis “Gestión de Producto”**

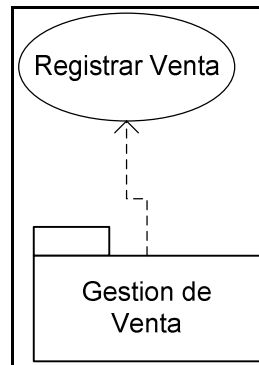
**Paquete “Gestión de Cliente”:** Comprende los casos de uso: Registrar Cliente, Buscar Cliente, Congelar Cliente, Registrar Dieta, Registrar Medidas Antropométricas, Registrar Antecedentes, Registrar Signos Vitales, Inscribir Cliente, Controlar Asistencia. Hay que considerar en este paquete que se están detallando todos los casos de uso para mayor claridad, sin tomar en cuenta el mecanismo de extensibilidad y de usabilidad que se encontró en el capítulo de Modelo de Requisitos.



**Figura 10.4 Paquetes del Análisis “Gestión de Cliente”**

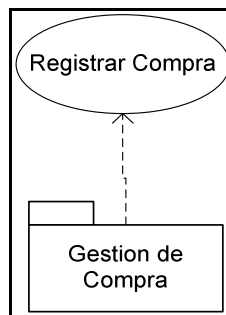


**Paquete “Gestión de Venta”:** El paquete de Gestión de Venta contiene el caso de uso Registrar Venta.



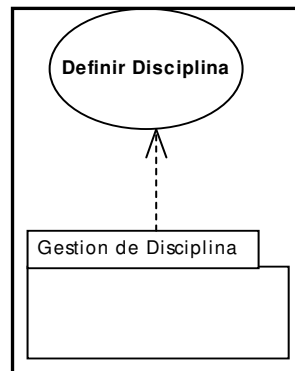
*Figura 10.5 Paquetes del Análisis “Gestión de Venta”*

**Paquete “Gestión de Compra”:** El paquete de Gestión de Compra contiene el caso de uso Registrar Compra.



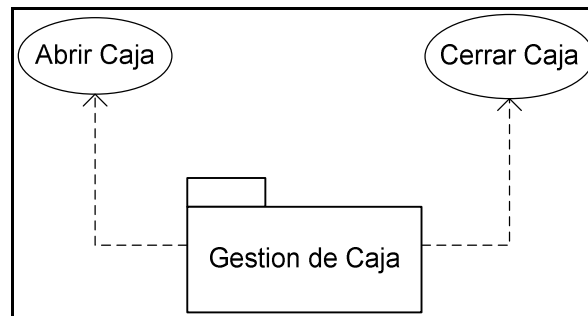
*Figura 10.6 Paquetes del Análisis “Gestión de Compra”*

**Paquete “Gestión de Disciplina”:** En el paquete Gestión de Disciplina contiene el caso de uso Definir Disciplina.



*Figura 10.7 Paquetes del Análisis “Gestión de Disciplina”*

**Paquete “Gestión de Caja”:** Comprende los siguientes casos de uso: Abrir Caja y Cerrar Caja.

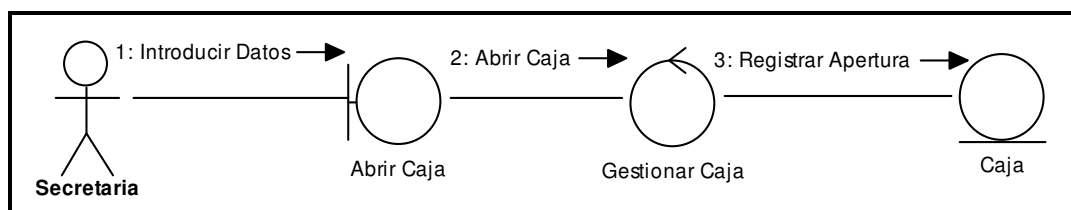


*Figura 10.8 Paquete del Análisis “Gestión de Caja”.*

## 10.2 ANÁLISIS DE LOS CASOS DE USO

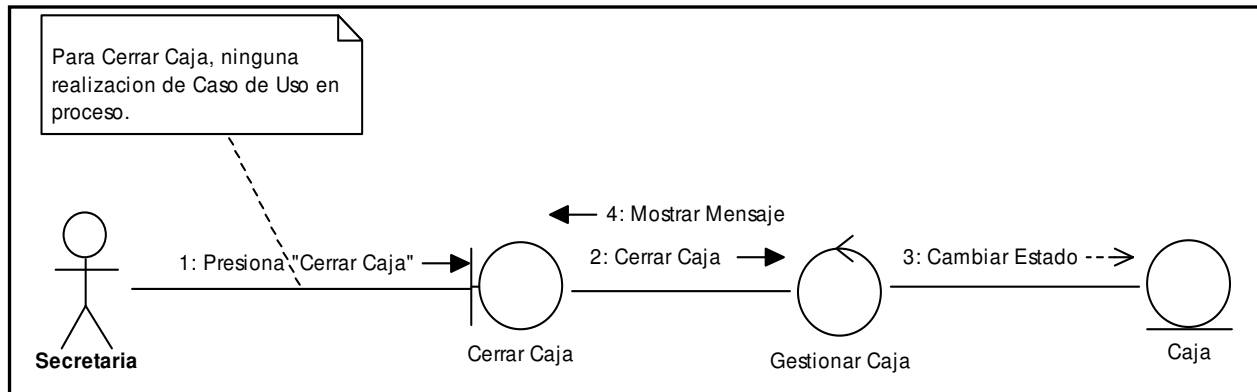
A través de esta actividad se realiza un estudio en detalle de los casos de uso para identificar las clases que participan y colaboran entre sí, para llevar a cabo los casos de usos. Se utilizan diagramas de colaboración donde se puede observar como los objetos interactúan unos con otros para el mismo fin. Esto se describe textualmente en los flujos de sucesos descritos para cada diagrama de colaboración.

- ✓ **Análisis del Caso de Uso “Abrir Caja”:** El diagrama de colaboración de la figura, muestra la relación que existe entre los objetos encontrados en el caso de Uso Abrir Caja. La secretaria ingresa a una interfaz para realizar la Apertura de Caja, introduciendo el monto inicial, fecha actual y una descripción, la clase de control Gestionar Caja, se encarga de abrir la primera transacción con los datos introducidos.



*Figura 10.9 Diagrama de Colaboración del Caso de Uso “Abrir Caja”*

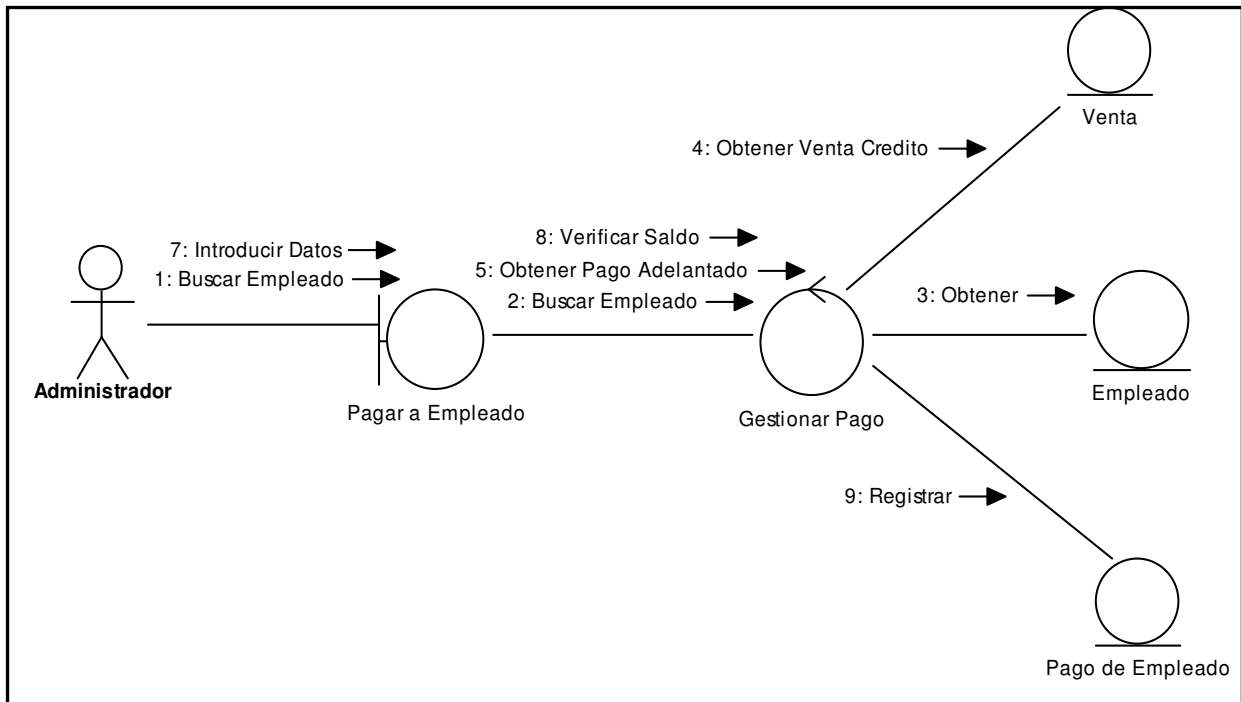
- ✓ **Análisis del Caso de Uso “Cerrar Caja”:** Como se muestra en la figura 10.10, la secretaria ingresa a una pantalla para realizar el cierre de los movimientos realizados en caja.



**Figura 10.10 Diagrama de Colaboración del Caso de Uso “Cerrar Caja”**

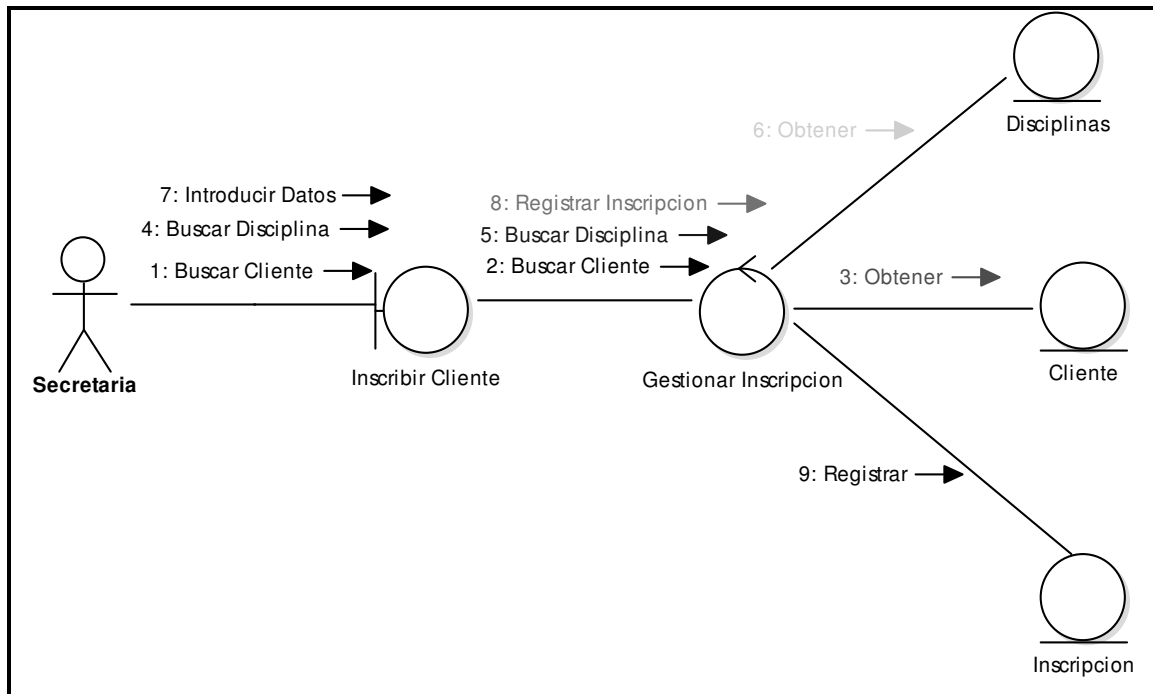
A través de la interfaz se envía un mensaje de Cerrar Caja a la clase de control Gestionar Caja quien es la encargada de obtener los movimientos originados por la clase entidad Caja de acuerdo a los parámetros introducidos por la secretaria. Nuevamente, la clase Gestionar Caja realiza el cierre de todos los movimientos según los parámetros y finalmente emite un mensaje de caja cerrada.

- ✓ **Análisis del Caso de Uso “Pagar a Empleado”:** El Administrador primeramente necesita buscar al empleado, la clase de control Gestionar Empleado obtiene los datos del Empleado como sueldo, nombre. El administrador introduce los datos del Sueldo, la clase Gestor de Pago de Empleado se encarga de registrar en la clase Pago de Empleado.



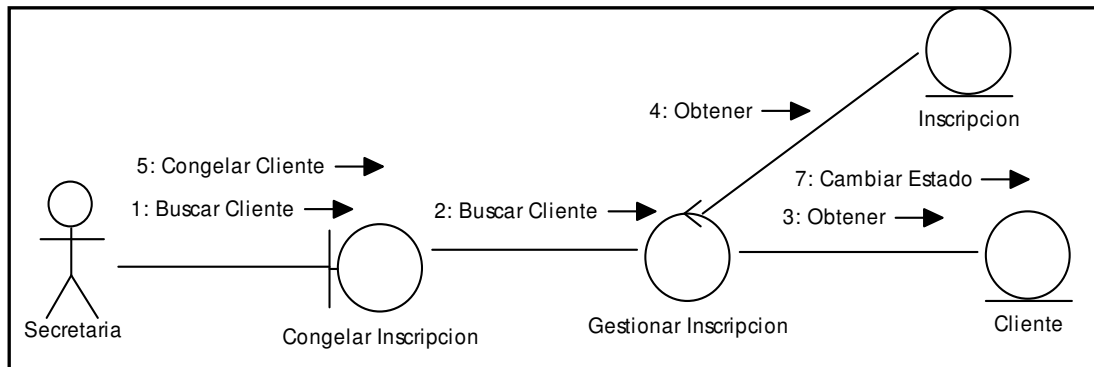
*Figura 10.11 Diagrama de Colaboración del Caso de Uso "Pagar a Empleado"*

- ✓ **Análisis del Caso de Uso "Inscribir Cliente":** El mensaje Buscar Cliente se envía a la clase de interfaz Inscribir Cliente. El mensaje Buscar Cliente se envía a la clase control Gestionar Inscripción. Obtener Cliente se envía a la clase entidad Cliente. La Secretaria, busca una Disciplina, la clase Gestionar Inscripción Obtiene las disciplinas de la clase entidad Disciplina. La secretaria introduce los datos de inscripción, que luego son registrados en la clase entidad Inscripción.



**Figura 10.12 Diagrama de Colaboración del Caso de Uso "Inscribir Cliente"**

- ✓ **Análisis del Caso de Uso "Congelar Cliente":** La secretaria introduce los criterios para buscar al Cliente, la clase Gestionar Incripción, obtiene los datos de la clase Cliente, la clase Gestión de Incripción, obtiene la última Incripción del Cliente. La secretaria presiona la opción Congelar, el sistema cambia de estado del Cliente a través de la clase Gestión de Incripción de Activo a Congelado.



**Figura 10.13 Diagrama de Colaboración del Caso de Uso "Congelar Inscripción"**

- ✓ **Análisis del Caso de Uso "Registrar Venta":** La secretaria busca al cliente para realizar la venta, la clase Gestionar Cliente busca al Cliente. La secretaria busca el producto a vender, la secretaria introduce el monto de pago, la cantidad de venta, el sistema calcula el total de la venta a través de la clase Gestionar Venta.

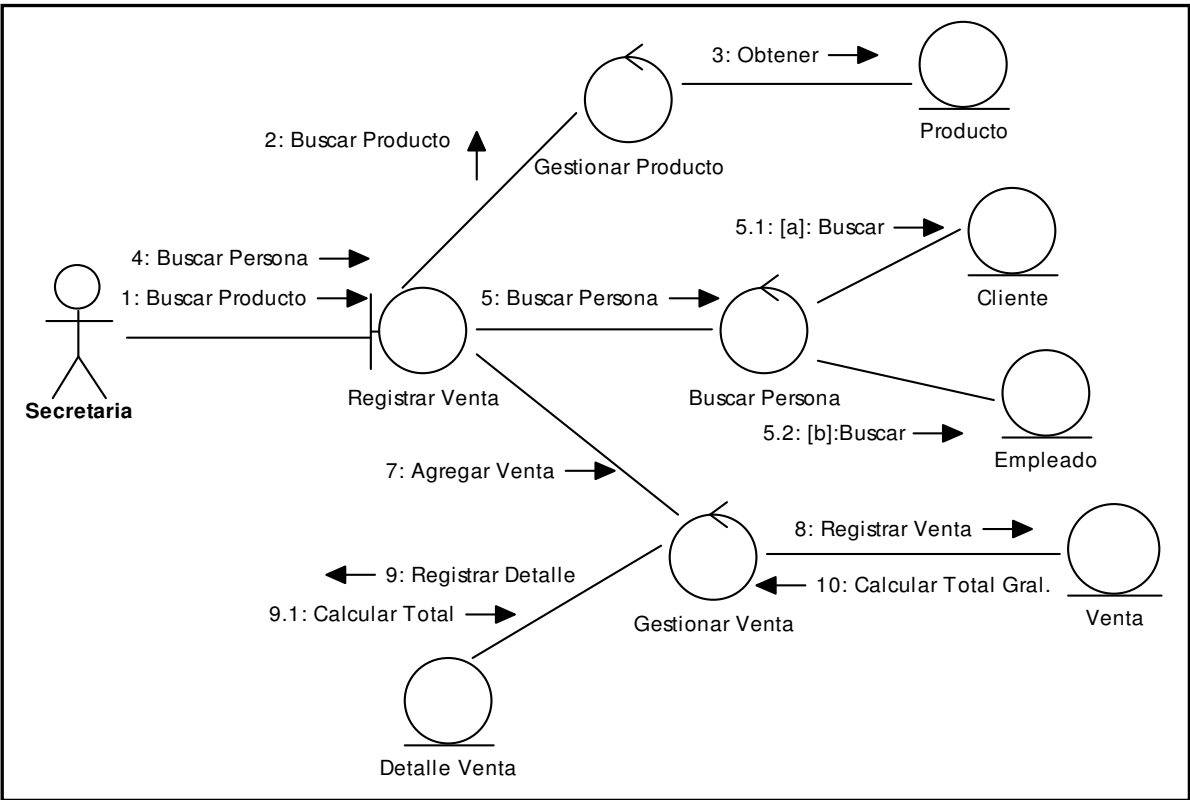
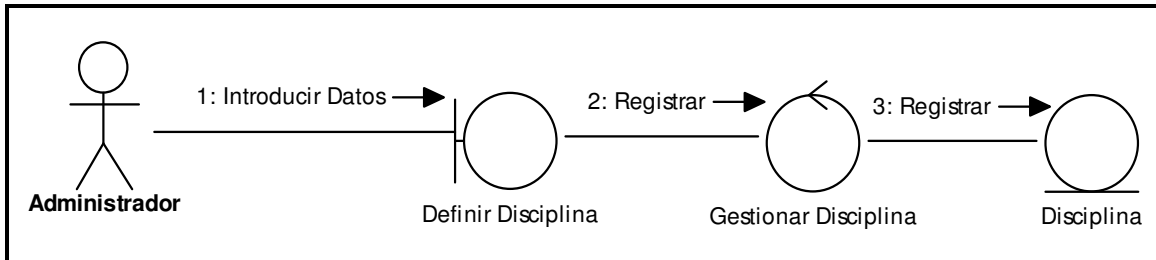


Figura 10.14 Diagrama de Colaboración del Caso de Uso “Registrar Venta”

- ✓ **Análisis del Caso de Uso “Definir Disciplina”:** El Administrador ingresa a una interfaz e introduce los datos como nombre y precio, para definir una disciplina. Se envía el mensaje Registrar a la clase de control Gestionar Disciplina. La clase Gestionar Disciplina envía el mensaje Registrar a la clase entidad Disciplina.

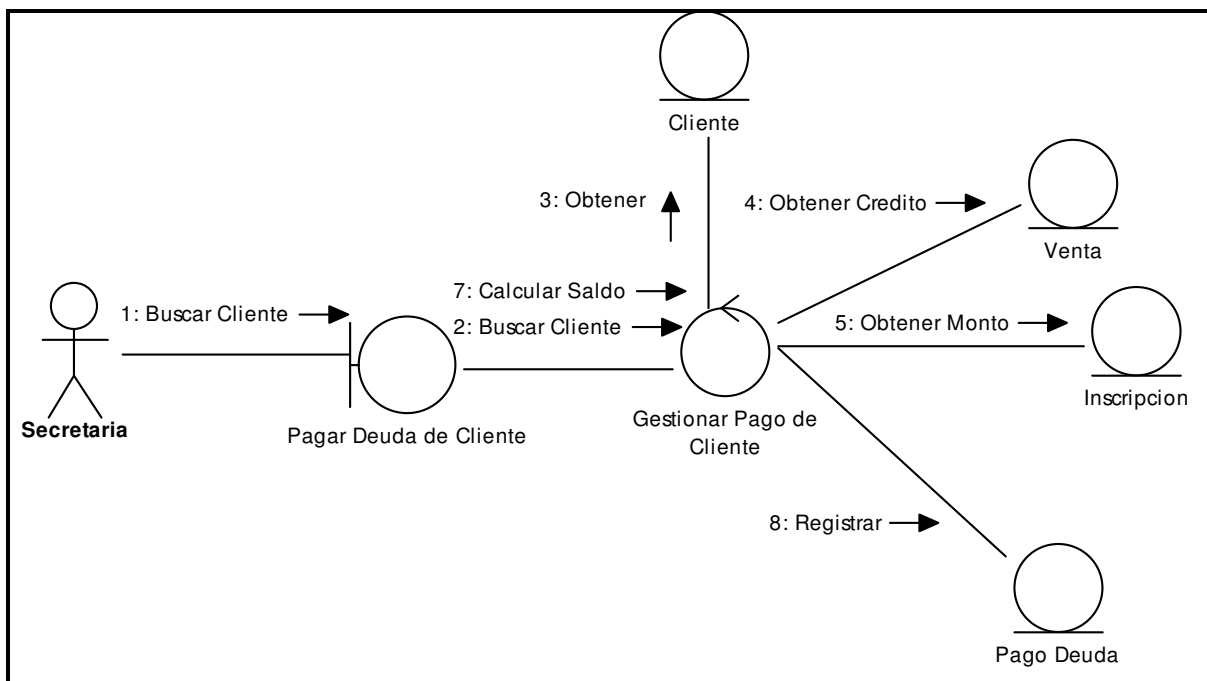


**Figura 10.15 Diagrama de Colaboración del Caso de Uso “Definir Disciplina”**



✓ **Análisis del Caso de Uso “Pagar Deuda de Cliente”:**

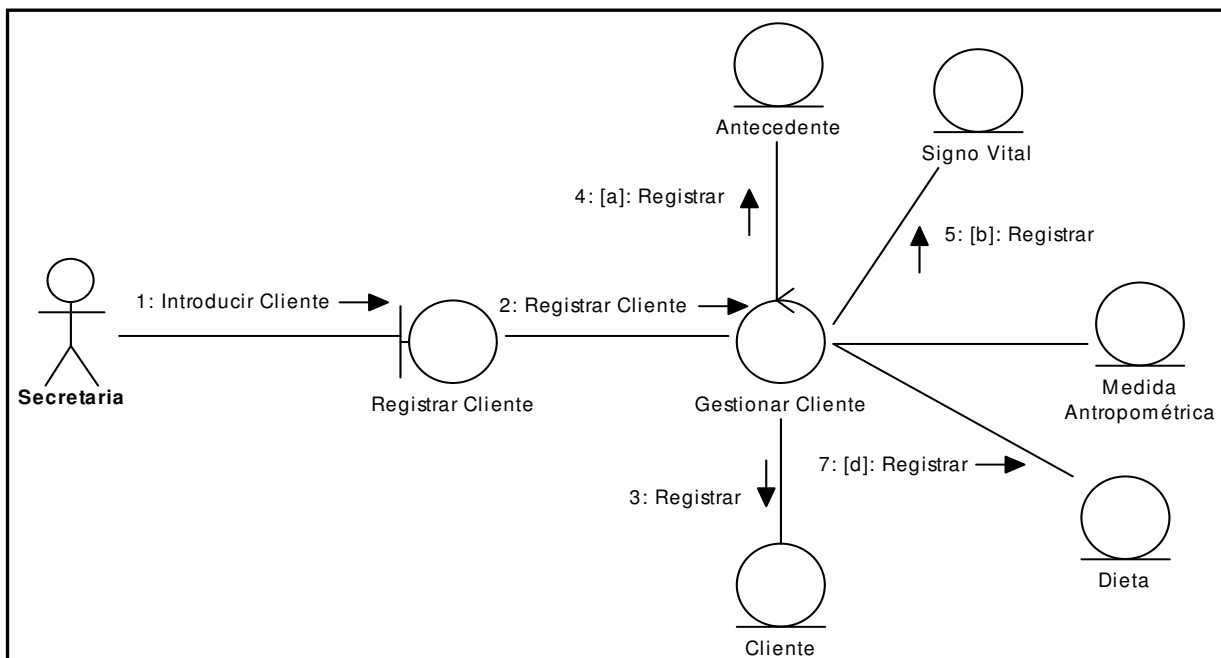
El actor secretaria, primeramente envía el mensaje de Buscar Cliente a través de la clase de interfaz Pagar Deuda de Cliente, esta a su vez envía otro mensaje de Buscar a la clase de control Gestionar Pago de Cliente, quien es la encargada de acceder a la clase entidad Cliente, y obtener los datos del cliente. La clase de control Gestionar Pago de Cliente, obtiene los montos de deuda de las clases de entidad Venta e Inscripción. La secretaria introduce el Monto de Pago que realiza el Cliente, la clase de Control Gestionar Pago de Cliente, realiza el cálculo del Saldo de la deuda.



**Figura 10.16 Diagrama de Colaboración del Caso de Uso “Pagar Deuda de Cliente”**

✓ **Caso de Uso “Registrar Cliente”:**

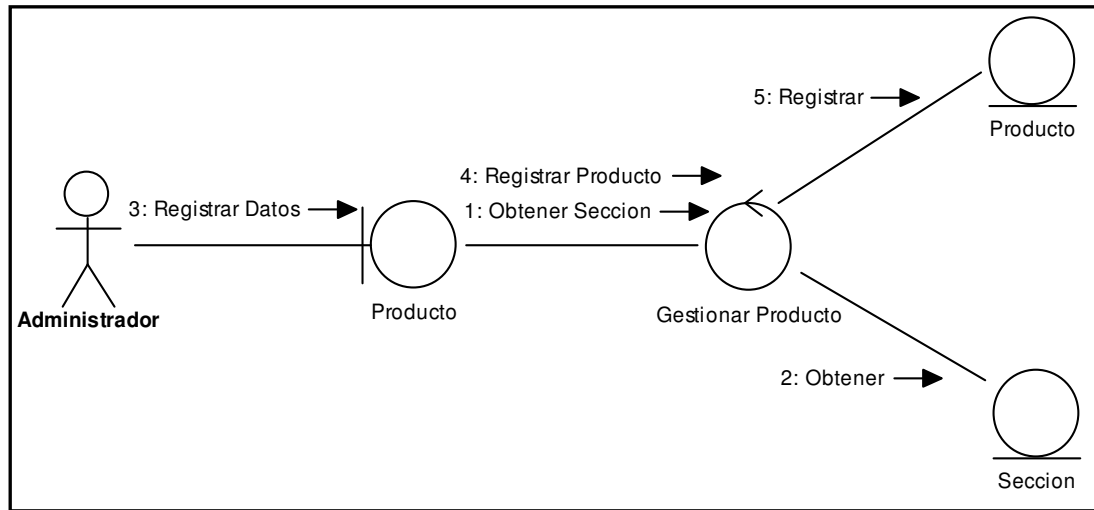
El actor introduce los datos del cliente, a través de la clase interfaz Registrar Cliente. La clase Registrar Cliente envía un mensaje a la clase de control Gestionar Cliente, quien a su vez, envía los datos del cliente a la clase entidad Cliente. Opcionalmente la clase de control puede enviar datos a la clase entidad Antecedentes, Signos Vitales, Medidas Antropométricas y a Dieta.



*Figura 10.17 Diagrama de Colaboración del Caso de Uso “Registrar Cliente”*

✓ **Análisis del Caso de Uso “Registrar Producto”:**

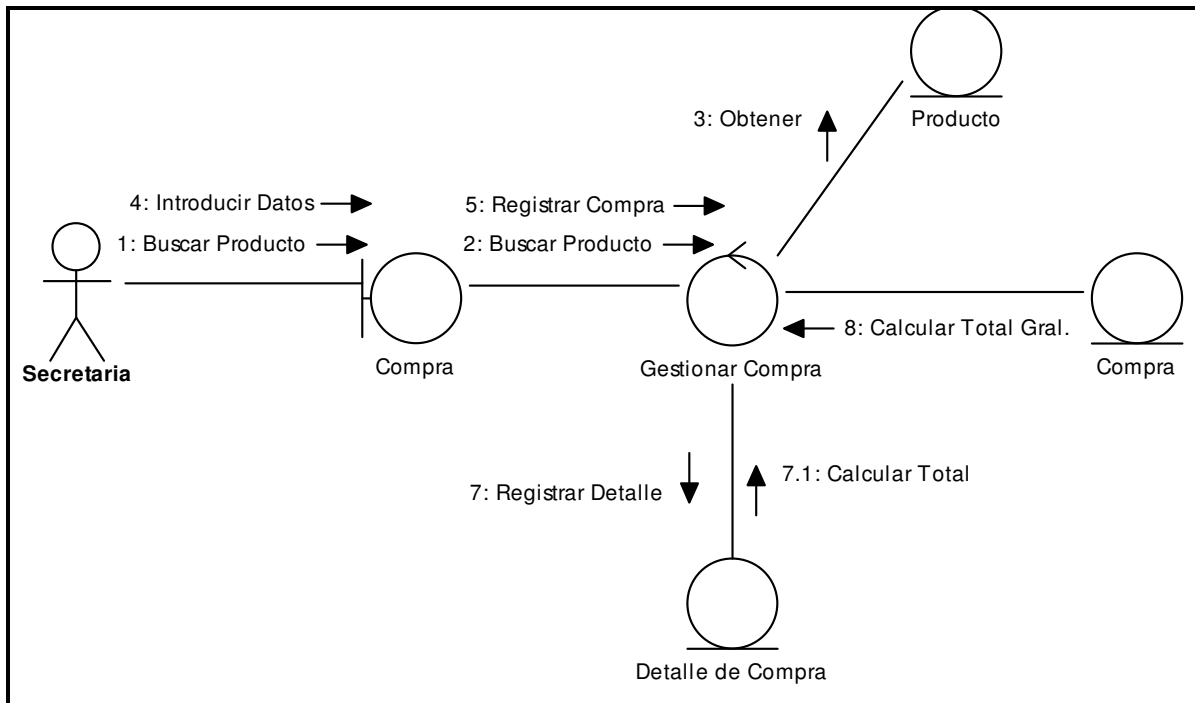
La clase interfaz, envía un mensaje de Obtener Sección a la clase de control Gestionar Producto, la cual accede a la clase entidad Sección. El administrador es el que define qué productos serán registrados. Introduce los datos del producto como nombre, precio unitario. La clase Gestionar Producto envía el mensaje de registro a la clase Producto.



*Figura 10.18 Diagrama de Colaboración del Caso de Uso "Registrar Producto"*

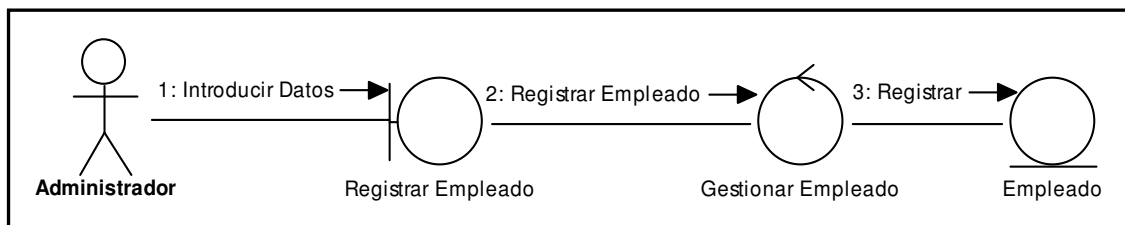
#### ✓ Análisis del Caso de Uso "Registrar Compra"

El actor Secretaria realiza la búsqueda a través de la clase de control Gestionar Compra, quien se encarga de obtener el producto de la clase entidad Producto. El actor introduce los datos de la compra, la clase Gestionar Compra, se encarga de Registrar en la clase Compra y de Agregar el Detalle de la compra, así también se encarga de realizar el calculo por detalle y el calculo general de la compra.



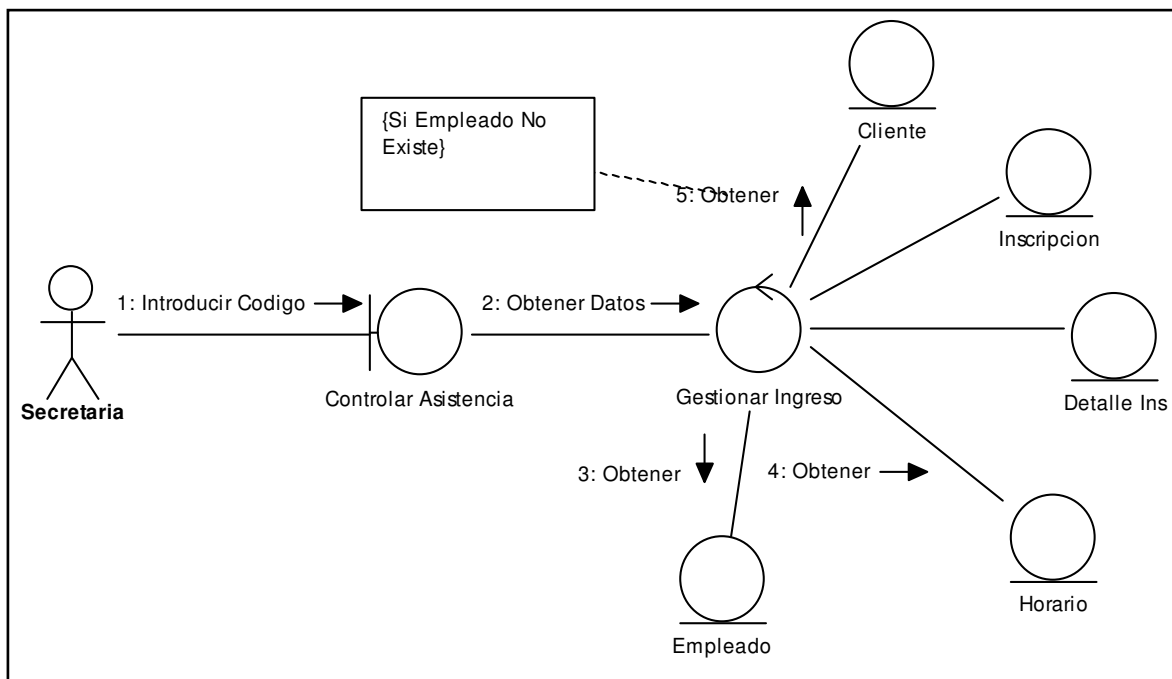
*Figura 10.19 Diagrama de Colaboración del Caso de Uso "Registrar Compra"*

- ✓ **Análisis del Caso de Uso "Registrar Empleado":** Introducir Datos se envía a la clase Registrar Empleado, el mensaje Registrar Empleado se envía a la clase de control Gestionar Empleado, quien envía el mensaje de Registrar a la clase Entidad Empleado.



*Figura 10.20 Diagrama de Colaboración del Caso de Uso "Registrar Empleado"*

- ✓ **Análisis del Caso de Uso “Controlar Asistencia”:** Introducir Código se envía a la clase Controlar Asistencia. Obtener Datos se envía a la clase de control Gestionar Ingreso. Se obtiene los datos de Empleado, luego se obtiene datos de Horario. Si no se encuentra los datos de Empleado. Se obtiene datos de Cliente, luego se verifica si en Inscripción está registrado y por último se envía Obtener a la clase Horario.



**Figura 10.21 Diagrama de Colaboración del Caso de Uso “Controlar Asistencia”**

### 10.3 ANÁLISIS DE CLASES.

En la presente sección se describen, las responsabilidades, los atributos, y las asociaciones de las clases de interfaz, de control y de identidad, de acuerdo a las clases identificadas en el análisis de los casos de uso.

#### 10.3.1 Clases de Interfaz.

Ahora se detalla las clases de Interfaz que fueron encontradas en la realización de los Casos de Uso:

➤ Clase de Interfaz *Abrir Caja*

<b>Nombre de la Clase</b>	Abrir Caja
<b>Responsabilidades</b>	Permite registrar información para abrir caja
<b>Atributos.</b>	Cuadro de texto Descripción, Cuadro de texto Monto de Apertura, cuadro de fecha Fecha de Apertura

➤ Clase de Interfaz *Cerrar Caja*

<b>Nombre de la Clase</b>	Cerrar Caja
<b>Responsabilidades</b>	Permite realizar el cierre de caja.
<b>Atributos.</b>	Cuadro de texto Saldo Actual, Cuadro de fecha Fecha de Cierre

➤ **Clase de Interfaz *Pagar a Empleado***

<b>Nombre de la Clase</b>	Pagar a Empleado
<b>Responsabilidades</b>	Permite realizar el pago a un empleado.
<b>Atributos.</b>	Cuadro de texto Nombre, Cuadro de texto Apellido Paterno, Cuadro de texto Apellido Materno, Cuadro de texto Sueldo Mensual, Lista desplegable Mes a Pagar, Cuadro de texto Monto de Pago, Cuadro de texto Saldo de Pago.

➤ **Clase de Interfaz *Inscribir Cliente***

<b>Nombre de la Clase</b>	Inscribir Cliente
<b>Responsabilidades</b>	Realizar la inscripción de un Cliente al Gimnasio
<b>Atributos.</b>	Lista desplegable de Disciplinas, Cuadro de texto Nombre, Cuadro de texto Apellido Paterno, Cuadro de texto Apellido Materno, Cuadro de texto Monto de Pago, Cuadro de texto Saldo, Cuadro de fecha Fecha de Inicio, Cuadro de fecha Fecha Fin, Cuadro de texto Nro de Dias.

➤ **Clase de Interfaz *Congelar Cliente***

<b>Nombre de la Clase</b>	Congelar Cliente
<b>Responsabilidades</b>	Congelar la inscripción de un Cliente inscrito.
<b>Atributos.</b>	Cuadro de texto Nombre, Cuadro de texto Apellido Paterno, Cuadro de texto Apellido Materno, Cuadro de fecha Fecha Inicio, Cuadro de fecha Fecha Fin, lista desplegable de estado del cliente.

➤ **Clase de Interfaz *Registrar Venta***

<b>Nombre de la Clase</b>	Venta
<b>Responsabilidades</b>	Realizar el registro de una Venta
<b>Atributos.</b>	Cuadro de texto Nombre Producto, Cuadro de texto Cantidad de Venta, Cuadro de fecha Fecha de Venta, Cuadro de texto Total, Cuadro de texto Total General, Lista de Detalle.

➤ **Clase de Interfaz *Definir Disciplina***

<b>Nombre de la Clase</b>	Definir Disciplina
<b>Responsabilidades</b>	Realizar el registro de Disciplinas
<b>Atributos.</b>	Cuadro de texto Nombre de la Disciplina, Cuadro de texto Precio.



➤ **Clase de Interfaz *Pagar Deuda de Cliente***

<b>Nombre de la Clase</b>	Pagar Deuda de Cliente
<b>Responsabilidades</b>	Realizar el pago de una deuda de un Cliente, sea por inscripción o por ventas al crédito.
<b>Atributos.</b>	Cuadro de texto Nombre, Cuadro de texto Apellido Paterno, Cuadro de texto Apellido Materno, Cuadro de texto Monto de Venta, Cuadro de texto Monto de Inscripción, Cuadro de texto Pago, Cuadro de texto Saldo.

➤ **Clase de Interfaz *Producto***

<b>Nombre de la Clase</b>	Producto
<b>Responsabilidades</b>	Realizar el registro de los Productos
<b>Atributos.</b>	Lista desplegable Sección, Cuadro de texto Nombre del Producto, Cuadro de texto Precio Unitario, Cuadro de texto Cantidad en Stock.

➤ **Clase de Interfaz *Compra***

<b>Nombre de la Clase</b>	Compra
<b>Responsabilidades</b>	Realizar el registro de Compras
<b>Atributos.</b>	Lista desplegable de Producto, Cuadro de texto Cantidad de Compra, Cuadro de fecha Fecha de Venta, Cuadro de texto Total, Cuadro de texto Total General, Lista de Detalle.

➤ **Clase de Interfaz *Control de Asistencia***

<b>Nombre de la Clase</b>	Control de Asistencia
<b>Responsabilidades</b>	Realizar el registro de Ingreso al Gimnasio.
<b>Atributos.</b>	Cuadro de texto Código, lista desplegable de Apellido Paterno, cuadro de etiqueta de Hora y Fecha actual, cuadro de etiqueta de Mensaje, botón Ingresar.

**10.3.2 Clases de Control.**➤ **Clase de Control *Gestionar Caja***

<b>Nombre de la Clase</b>	Gestionar Caja
<b>Responsabilidades</b>	Encargada de Abrir y Cerrar la Caja, también de realizar el registro a la entidad Caja.

➤ **Clase de Control *Gestionar Venta***

<b>Nombre de la Clase</b>	Gestionar Venta
<b>Responsabilidades</b>	Encargada de Mantener las Ventas de empleados o a clientes, calcular total y total general de la venta.

➤ **Clase de Control** *Gestionar Pago de Empleados*

<b>Nombre de la Clase</b>	Gestionar Pago de Empleados
<b>Responsabilidades</b>	Encargada de Mantener el Pago, Verificar los Créditos otorgados, Calcular los Saldos.

➤ **Clase de Control** *Gestionar de Inscripción*

<b>Nombre de la Clase</b>	Gestión de Inscripción
<b>Responsabilidades</b>	Realizar la inscripción de un cliente, calcular el monto de cambio.

➤ **Clase de Control** *Buscar Persona*

<b>Nombre de la Clase</b>	Buscar Persona
<b>Responsabilidades</b>	Realizar la búsqueda de Clientes o de Empleados

➤ **Clase de Control** *Gestionar Disciplina*

<b>Nombre de la Clase</b>	Gestionar Disciplina
<b>Responsabilidades</b>	Encargada de Mantener el registro de Disciplina.

➤ **Clase de Control** *Gestionar Pago de Cliente*

<b>Nombre de la Clase</b>	Gestionar Pago de Cliente
<b>Responsabilidades</b>	Encargada de Calcular el Monto de deuda que tiene un cliente sea por las ventas o por inscripción.  Encargada de Registrar el Pago de la Deuda.

➤ **Clase de Identidad *Pago deuda***

<b>Nombre de la Clase</b>	Pago Deuda.
<b>Responsabilidades</b>	Almacenar los datos referentes a la Deuda de un Cliente.
<b>Atributos</b>	Fecha de Pago, Monto Pago.
<b>Relaciones</b>	Gestionar Pago de Cliente

➤ **Clase de Control *Gestionar Producto***

<b>Nombre de la Clase</b>	Gestionar Producto
<b>Responsabilidades</b>	Encargada de Mantener los Productos.

➤ **Clase de Control *Gestionar Compra***

<b>Nombre de la Clase</b>	Gestionar Compra
<b>Responsabilidades</b>	Encargada de Mantener una compra.  Encargada de Calcular el Total y el Total General de la Compra.

➤ **Clase de Control *Gestionar Ingreso***

<b>Nombre de la Clase</b>	Gestionar Ingreso
<b>Responsabilidades</b>	Encargada de coordinar acciones entre las clases de entidad Cliente, Empleado, Horario, Inscripción, DetIns.

**10.3.3 Clases de Identidad.**➤ **Clase de Identidad *Caja***

<b>Nombre de la Clase</b>	Caja
<b>Responsabilidades</b>	Mantener los datos referentes al flujo de caja.
<b>Atributos</b>	Monto Saldo, Fecha Apertura, Fecha Cierre
<b>Relaciones</b>	Gestionar Caja

➤ **Clase de Identidad *Venta***

<b>Nombre de la Clase</b>	Venta
<b>Responsabilidades</b>	Mantener los datos referentes a las ventas realizadas.
<b>Atributos</b>	Monto Total, Fecha Actual, Descripción
<b>Relaciones</b>	Gestionar Pago, Gestionar Venta

➤ **Clase de Identidad *Empleado***

<b>Nombre de la Clase</b>	Empleado
<b>Responsabilidades</b>	Mantener los datos referentes a los empleados que trabajan en el Gimnasio.
<b>Atributos</b>	Nombre, Apellido Paterno, Apellido Materno, Dirección, Teléfono, Carnet de Identidad, Sueldo
<b>Relaciones</b>	Buscar Persona, Gestionar Pago, Gestionar Empleado

➤ **Clase de Identidad *Pago de Empleado***

<b>Nombre de la Clase</b>	Pago de Empleado
<b>Responsabilidades</b>	Mantener los datos referentes a los pagos que se realizan a los empleados del Gimnasio.
<b>Atributos</b>	Pago Sueldo, Saldo, Fecha Actual, Mes de Pago
<b>Relaciones</b>	Gestionar Pago

➤ **Clase de Identidad *Disciplina***

<b>Nombre de la Clase</b>	Disciplina
<b>Responsabilidades</b>	Mantener los datos referentes a las disciplinas que ofrece el Gimnasio a los clientes.
<b>Atributos</b>	Descripción, Precio
<b>Relaciones</b>	Gestionar Disciplina

➤ **Clase de Identidad *Cliente***

<b>Nombre de la Clase</b>	Cliente
<b>Responsabilidades</b>	Mantener los datos referentes al Cliente.
<b>Atributos</b>	Nombre, ApellidoP, ApellidoM, Dirección, TelDomicilio, TelfCelular, TelfTrabajo, Fecha Nacimiento, Edad.
<b>Relaciones</b>	Gestionar Inscripción, Buscar Persona, Gestionar Pago de Cliente, Gestionar Cliente, Gestionar Ingreso.

➤ **Clase de Identidad *Inscripción***

<b>Nombre de la Clase</b>	Inscripción.
<b>Responsabilidades</b>	Mantener los datos referentes a la Inscripción.
<b>Atributos</b>	Descripción, Fecha Inicio, Fecha Fin, Nro Días, Monto Pago.
<b>Relaciones</b>	Gestionar Inscripción, Gestionar Pago de Cliente, Gestionar Ingreso.

➤ **Clase de Identidad *Horario***

<b>Nombre de la Clase</b>	Horario.
<b>Responsabilidades</b>	Mantener los datos referentes a los horarios asignados a las disciplinas.
<b>Atributos</b>	Hora Ini, Hora Fin, strDescripcion.
<b>Relaciones</b>	Gestionar Ingreso.

## **CAPITULO 11.**

### **MODELO DE DISEÑO**

“En el diseño modelamos el sistema y encontramos su forma (incluida la arquitectura) para que soporte todos los requisitos –incluyendo los requisitos no funcionales y otras restricciones- que se le supone.”<sup>1</sup>

#### **11.1 DISEÑO DE LA ARQUITECTURA.**

En el diseño de la arquitectura se identifican los nodos y configuraciones de red, subsistemas, clases del diseño significativas y los mecanismos de diseño genéricos que tratan requisitos comunes.

##### **11.1.1 Diagramas de Despliegue.**

“El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El modelo de despliegue se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño”.<sup>2</sup>

En la figura 11.1, se muestra el diagrama de despliegue que comprende los elementos hardware y software necesarios para tomar en cuenta en el diseño del sistema.

---

<sup>1</sup> El Proceso Unificado de Desarrollo de Software, James Rumbaugh, Pág. 205

<sup>2</sup> El Proceso Unificado de Desarrollo de Software, James Rumbaugh, Pág. 217



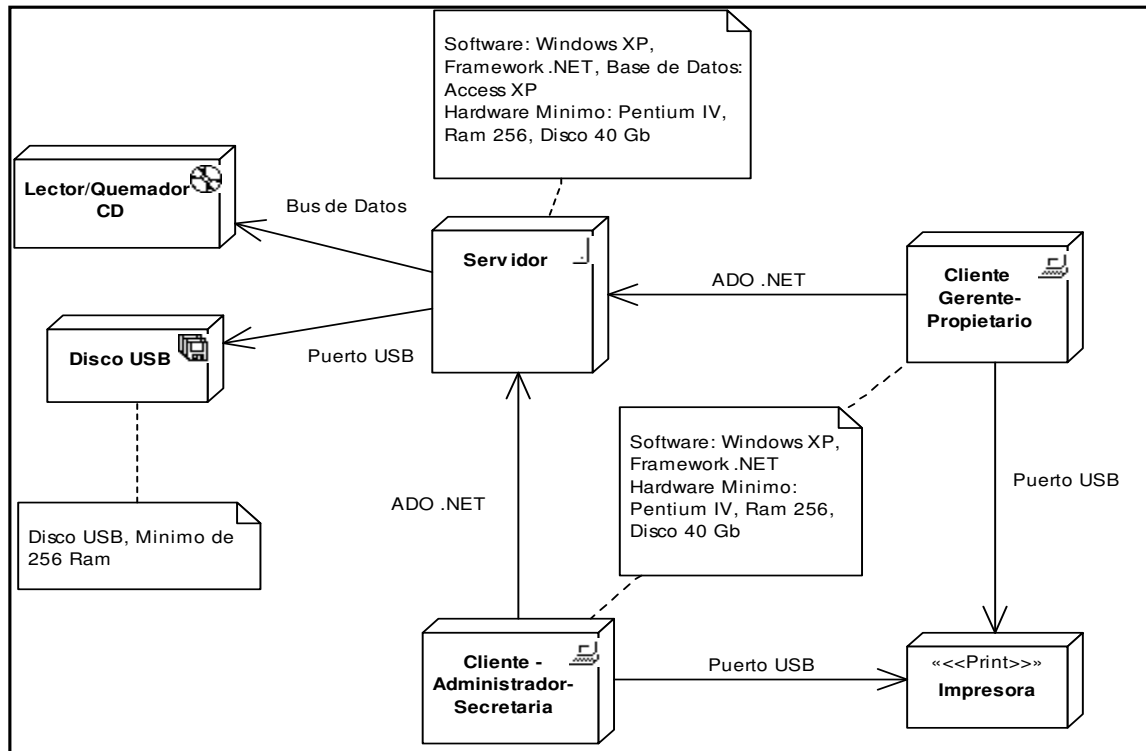
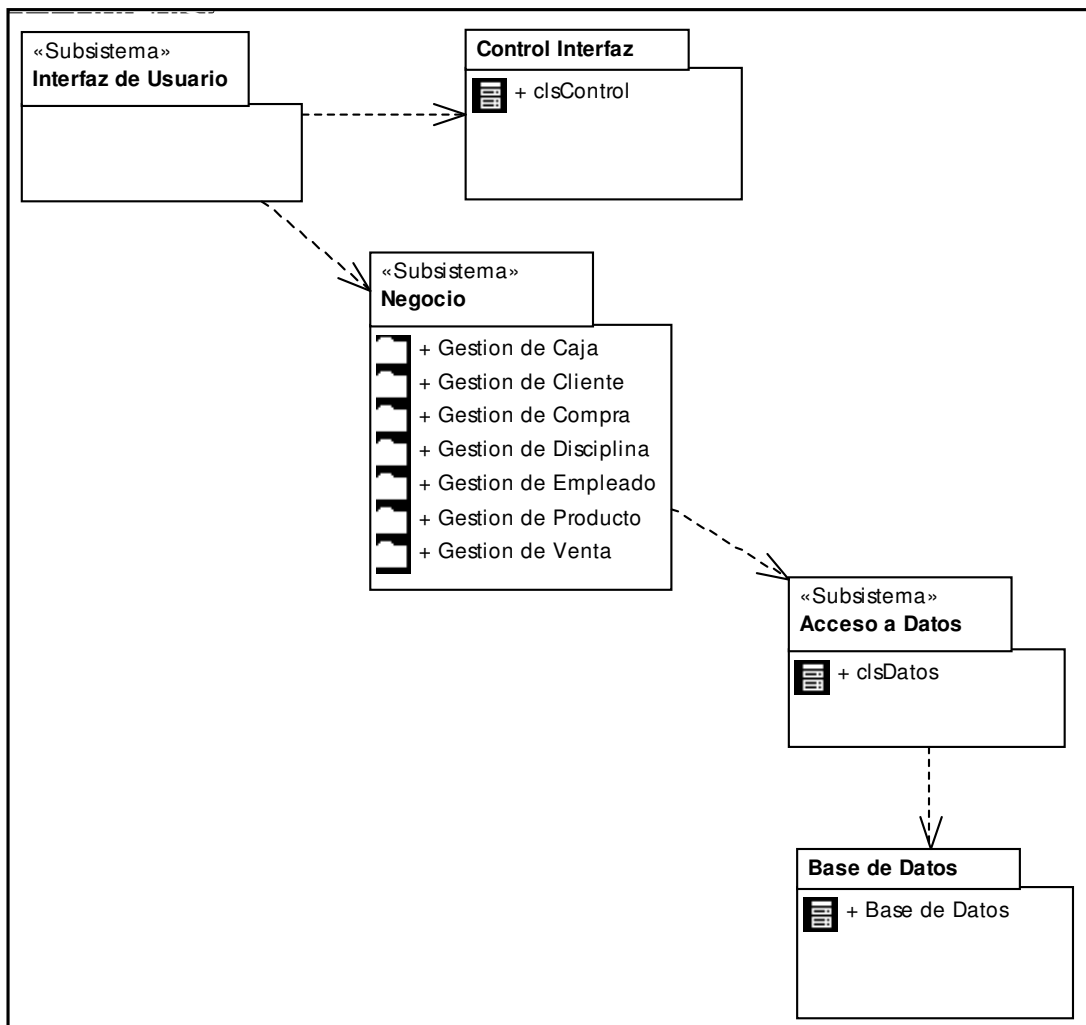


Figura 11.1 Diagrama de Despliegue para el Sistema.

### 11.1.2 Subsistemas de Diseño

En la figura 11.2, se muestran los subsistemas encontrados en el sistema **Gimnasoft**, esto con el fin de organizar el modelo de diseño en piezas manejables.

Como se puede observar se ha agrupado la Interfaz de Usuario en un subsistema, que ha su vez utilizará el subsistema Clases de Negocio. Por ultimo el subsistema Clases de Negocio utilizará el subsistema Clase de Datos para acceder a la base de datos.



*Figura 11.2 Subsistemas para el sistema GimnaSoft*

El subsistema **Negocio** agrupa los paquetes encontrados en la actividad *Identificación de Paquetes del Análisis*.

El subsistema **Control** agrupa las clases de control que se identificaron en la realización de casos de usos-análisis.

El subsistema **Acceso a Datos**, contiene una clase encargada de acceder al subsistema **Base de Datos**.

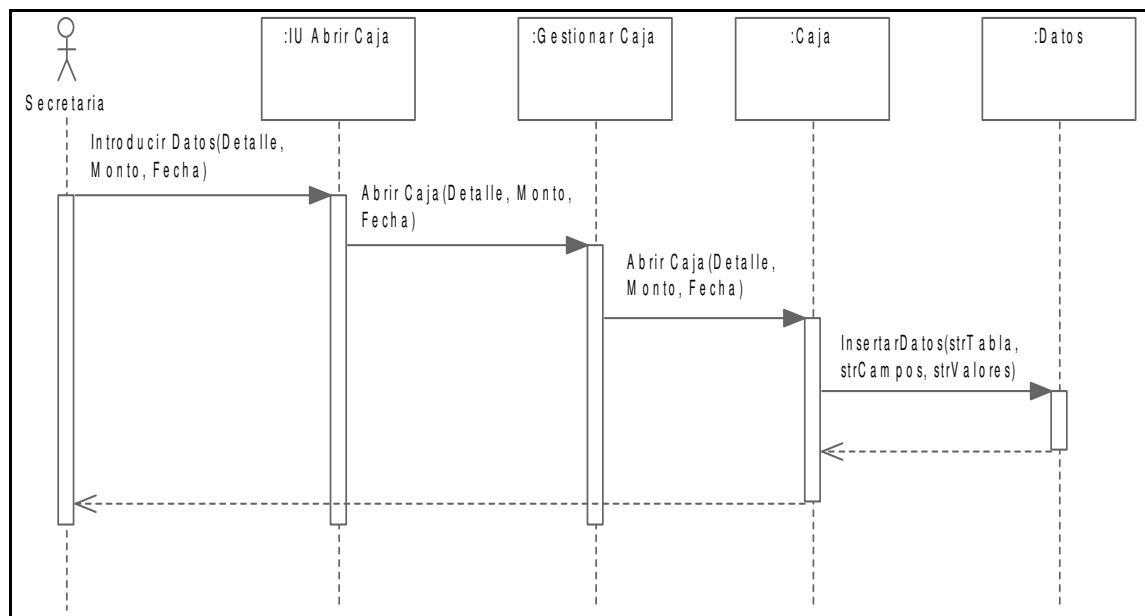
## 11.2 DISEÑO DE LOS CASOS DE USO.

“En el diseño, es preferible representar los objetos implicados que interactúan para llevar a cabo un caso de uso, con diagramas de secuencia, ya que nuestro centro de atención principal es el encontrar secuencias de interacciones detalladas y ordenadas en el tiempo”.<sup>3</sup>

“En diseño, podemos añadir una capa de interfaz de usuario al diagrama de secuencia que ayude a aclarar las cosas”<sup>4</sup>

En las secciones siguientes se utilizan diagramas de secuencia donde se puede observar como los objetos del diseño interactúan unos con otros para el mismo fin. Esto se describe textualmente en los flujos de sucesos descritos para cada diagrama de secuencia. Cada diseño de caso de uso presenta una interfaz de usuario con la cual el usuario interactúa.

### 11.2.1 Caso de Uso “Abrir Caja”

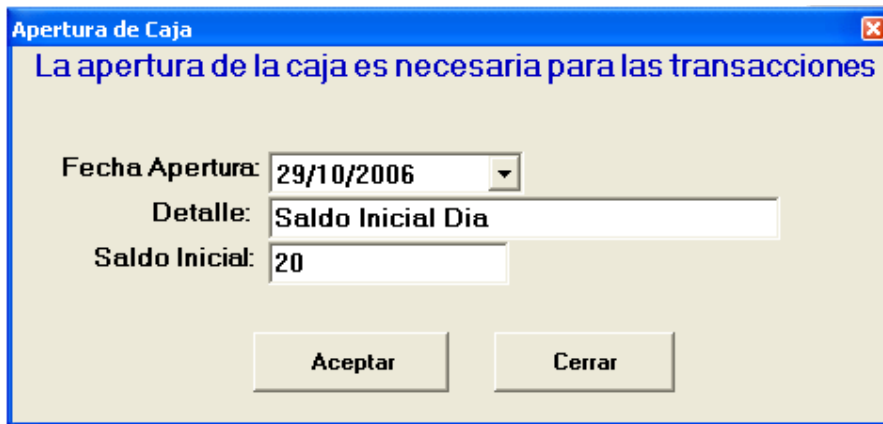


**Figura 11.3 Diagrama de secuencia del caso de uso “Abrir Caja”**

<sup>3</sup> El Proceso Unificado de Desarrollo, James Rumbaugh, pág. 212.

<sup>4</sup> Programacion UML 2.0, Jim Arlow, pag. 272.

La secretaria introduce los datos para abrir la caja como: Descripción, Monto y Fecha de apertura, a través de la interfaz de usuario Abrir Caja. Se envía el mensaje de Abrir Caja a la clase entidad Caja. La clase Caja, envía el mensaje Insertar Datos. La clase Caja envía un mensaje al actor Secretaria.



Apertura de Caja

La apertura de la caja es necesaria para las transacciones

Fecha Apertura: 29/10/2006

Detalle: Saldo Inicial Dia

Saldo Inicial: 20

Aceptar Cerrar

*Figura 11.4 Interfaz de Usuario para el caso de uso “Abrir Caja”*

### 11.2.2 Caso de Uso “Cerrar Caja”

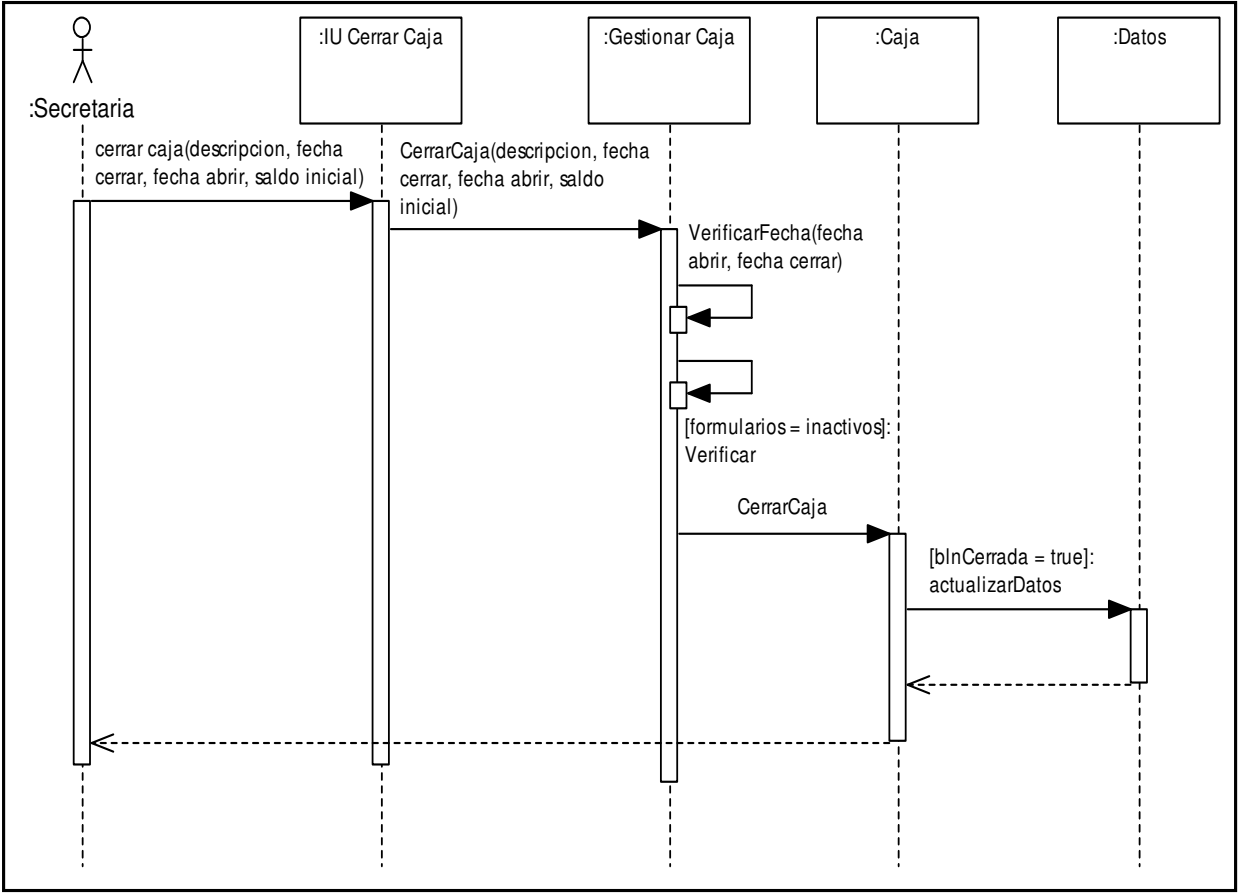


Figura 11.5 Diagrama de secuencia del caso de uso “Cerrar Caja”

La secretaria envía el mensaje Cerrar Caja a IU Cerrar Caja, enviando como parámetro una Descripción, Fecha de Cierre, Fecha Abrir y Saldo Inicial. El mensaje Cargar Ultima se envía a Gestionar Caja. Cargar Última se envía a la clase Caja. GetUltimoRegistro envía la clase Caja a la clase Datos. Gestionar Caja envía el mensaje Mostrar Saldo a la interfaz de usuario, que luego se envía a la Secretaria. El actor Secretaria presiona el botón Cerrar, se envía el mensaje Cerrar Caja a Gestionar Caja. La clase Gestionar Caja verifica si no existe alguna transacción en curso. Se envía el mensaje Cerrar Caja a la clase entidad Caja. El mensaje Actualizar Datos se envía a la clase Datos. La clase Caja envía un mensaje de “caja cerrada” al actor Secretaria.

**Figura 11.6** *Interfaz de Usuario para el Caso de Uso “Cerrar Caja”*

### 11.2.3 Caso de Uso “Pagar a Empleado”

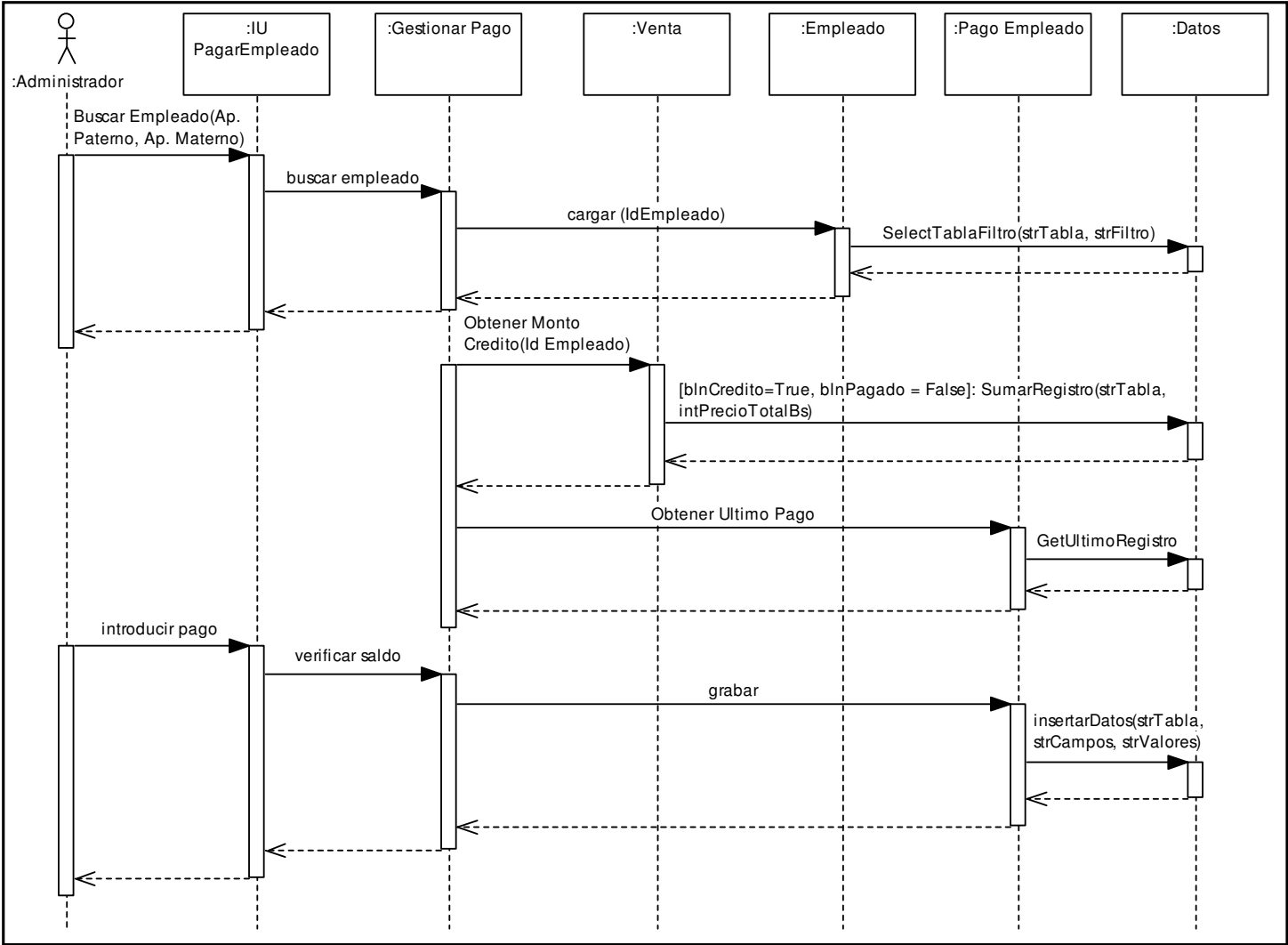


Figura 11.7 Diagrama de secuencia del caso de uso “Pagar a Empleado”

El actor Administrador, envía el mensaje Buscar Empleado a la clase de interfaz Pagar Empleado. Se envía un mensaje Buscar Empleado a la clase Gestionar Pago, que a su vez envía un mensaje Cargar a la clase Empleado. Empleado envía un mensaje SelectTablaFiltro a la clase de acceso a la base de datos. La clase Gestionar Pago envía el mensaje Obtener Monto Credito a la clase Venta. Se envía un mensaje SumarRegistro con parámetros strTabla, intPrecioTotalBs a la clase Datos, además que cumplan las condiciones de blnCredito=True y blnPagado = False. El mensaje Obtener Ultimo Pago se envía a Pago Empleado. Se envía un mensaje GetUltimoRegistro a la clase Datos. El actor introduce el monto de pago, Verificar Saldo se envía a la clase Gestionar Pago. El mensaje Grabar se envía a Pago Empleado. El mensaje InsertarDatos se envía a la clase Datos, que contiene los parámetros strTabla, strCampos y strValores.

Pagos a Empleados

Nro Boleta: Fecha: 07/10/2006

Apellido P.: Campos

Apellido M.: Saavedra

Nombre: Bernardo

Sueldo: 200

Descripción: Pago de Anticipo

Total Adelantado: 0

Ultimo Mes Pagado:

Monto Saldo: 200

Del año:

Créditos: 0

Mes a Pagar: Enero

Monto a Pagar: 100

☐ Pagar Créditos.

**Figura 11.8** Interfaz de Usuario para el Caso de Uso “Pagar Empleado”



11.2.4 Caso de Uso “Inscribir Cliente”

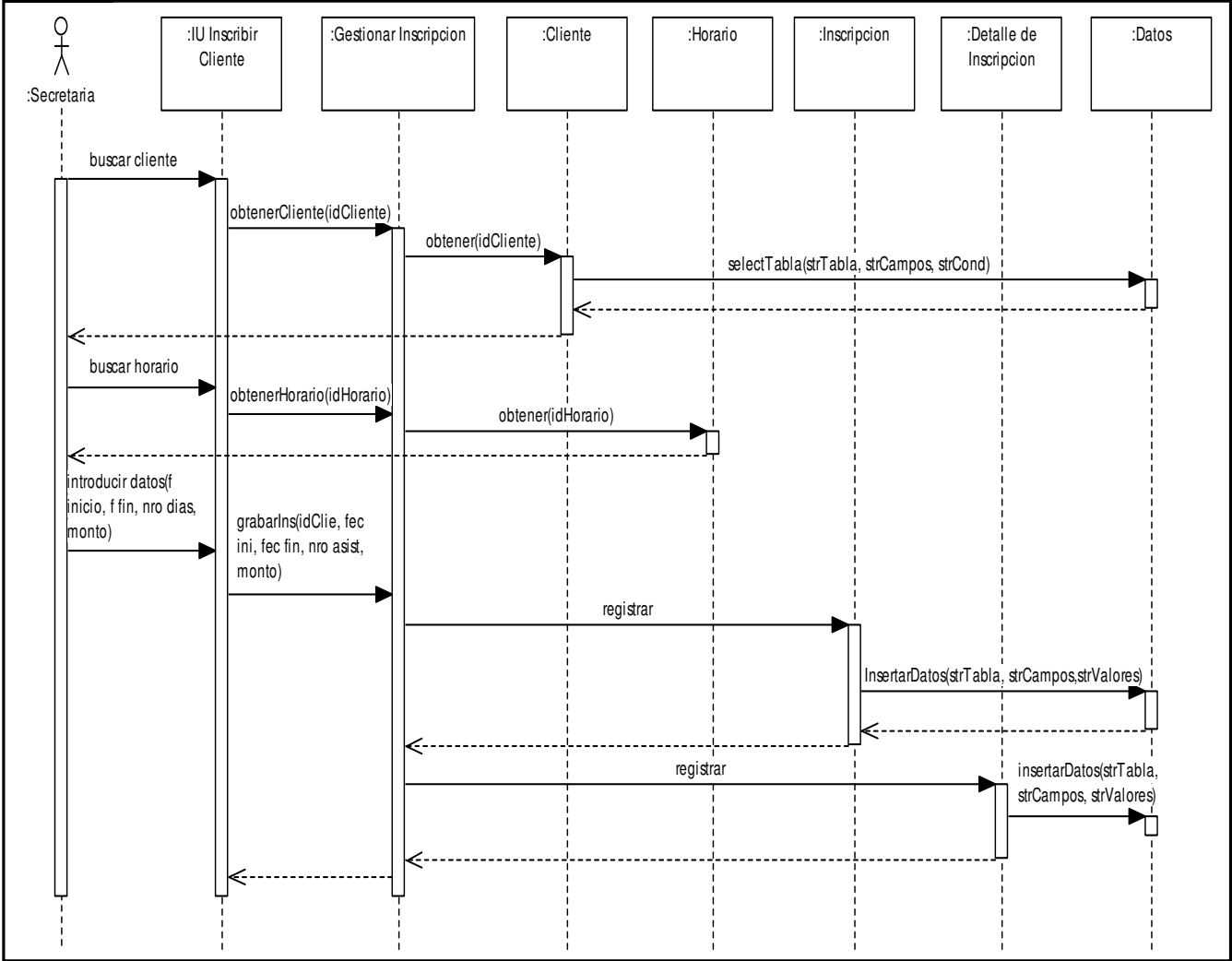


Figura 11.9 Diagrama de secuencia del caso de uso “Inscribir Cliente”

La secretaria busca un cliente para realizar la inscripción. La clase Gestionar Inscripción envía un mensaje Obtener a la clase Cliente. La clase Cliente envía el mensaje SelectTablaFiltro a la clase Datos. Esta retorna los datos del cliente.

La secretaria busca los diferentes horarios disponibles, se envía un mensaje Obtener de la clase Horarios. La secretaria introduce la fecha de inicio, la fecha fin, y el número de días. Se envía el mensaje GrabarIns con sus parámetros respectivos. La clase Gestionar Inscripción envía un mensaje de Registrar a la clase Inscripción. Se envía el mensaje de InsertarDatos a la clase Datos. La clase Gestionar Inscripción envía el mensaje Registrar a Detalle de Inscripción., ésta ultima envía el mensaje InsertarDatos a la clase Datos.

**Inscripcion de Clientes**

Cliente: Andia Perez David +

Disciplinas y Horarios: Aparatos / 121.05 Bs / 15 \$us / De: 1:56 A: 2:56

strDescripcion	int
Aparatos / 121.05 Bs / 15 \$us / De: 1:56 A: 2:56	0

Fecha Inicio: 18/12/2007 Fecha Fin: 18/12/2007

Nro de Asistencias: 0

Efectivo Bs.: A cuenta: Total Bs: 121,05

Cambio Bs.: Deuda:

<< < > >>

Adicionar Grabar Cancelar Modificar Eliminar Cerrar

Figura 11.10 Interfaz de Usuario para el Caso de Uso “Inscribir Cliente”

11.2.5 Caso de Uso “Congelar Cliente”

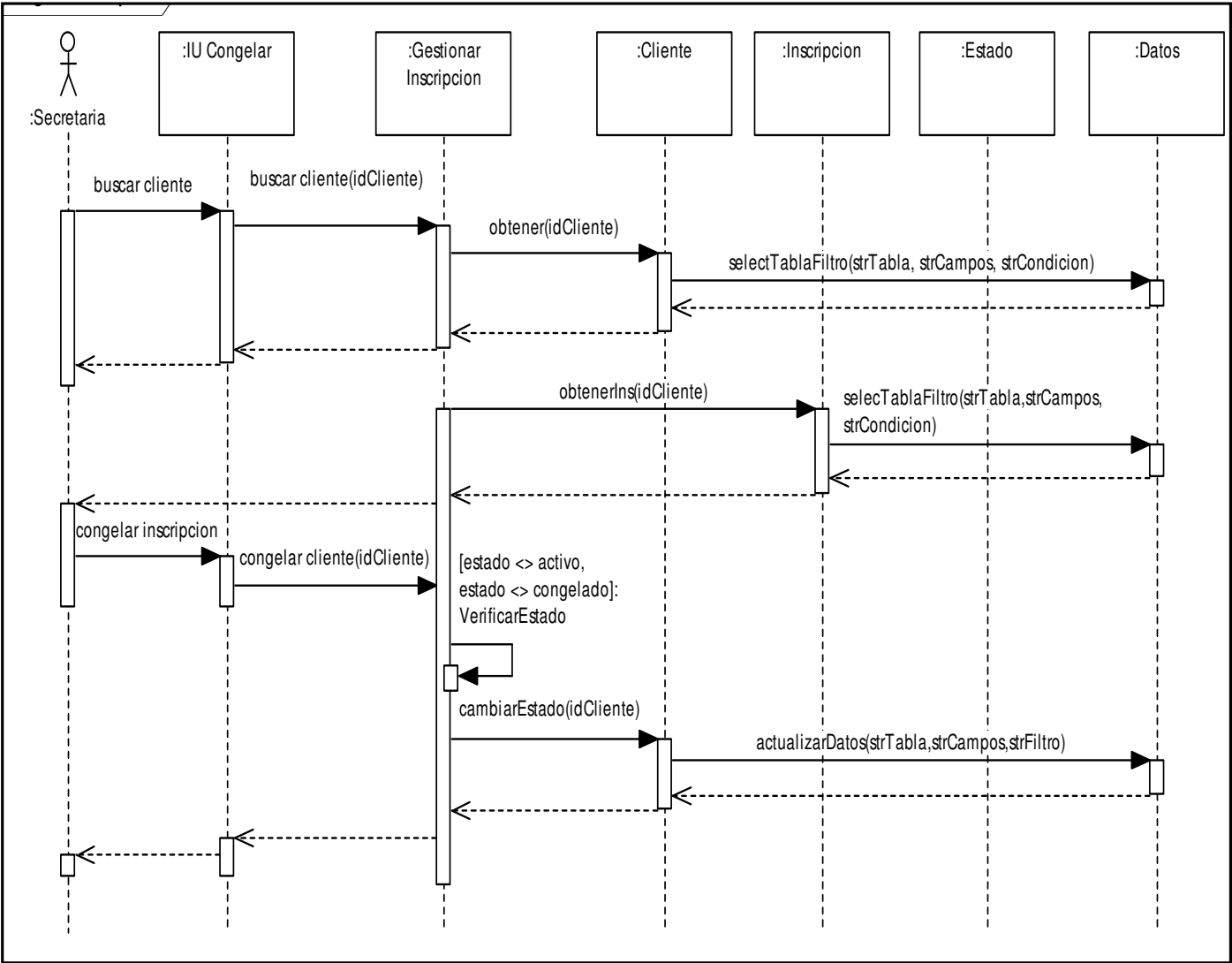


Figura 11.11 Diagrama de secuencia del caso de uso “Congelar Cliente”

El cliente busca al cliente. La clase Gestionar Inscripción envía el mensaje Obtener con el código del cliente. La clase Cliente envía el mensaje SelectTablaFiltro a la clase Datos. La clase Gestionar Inscripción envía el mensaje TieneIns para verificar si el cliente tiene una Inscripción. Se envía el mensaje SelectTablaFiltro a la clase Datos. La secretaria presiona Congelar Inscripción. La clase Gestionar Inscripción verifica el estado actual del cliente que sea diferente de Activo y de Congelado. Se envía el mensaje CambiarEstado a la clase Cliente. Se envía el mensaje ActualizarDatos para cambiar el estado del cliente a congelado.

Congelar Cliente

Congelar Cliente

Ciente: 125 campos saavedra bernardo

Fecha de Inicio: 23/12/2007

Estado del Cliente: Activo

Detalle de la Inscripción:

	Descripcion
▶	Aparatos / 121 bs. / 15\$us / De 1:56 A 2:56
*	

Congelar

Cerrar

Figura 11.12 Interfaz de Usuario para el Caso de Uso “Congelar Cliente”

### 11.2.6 Caso de Uso “Pagar deuda de Cliente”

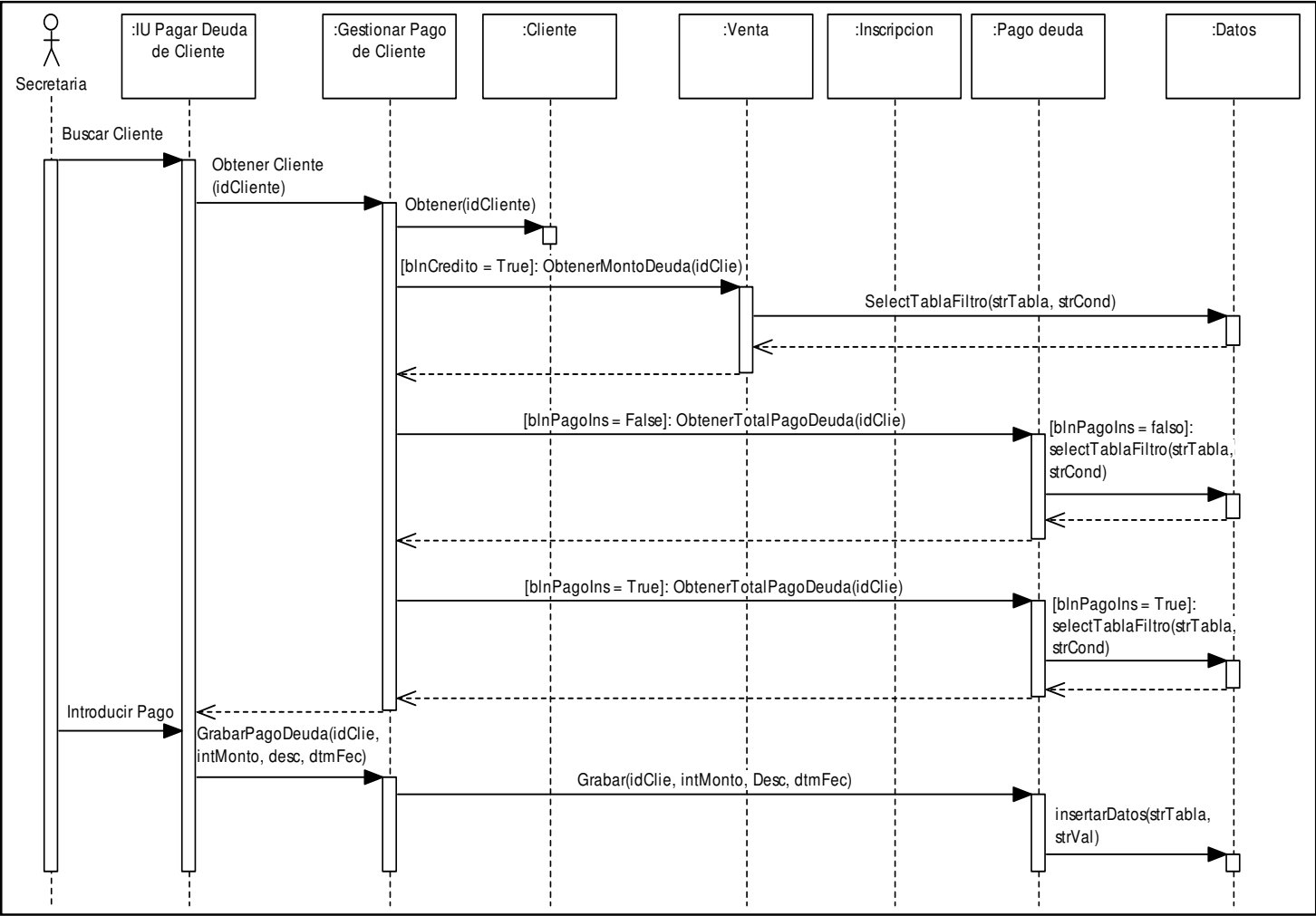


Figura 11.13 Diagrama de secuencia del caso de uso “Pagar Deuda de Cliente”

La secretaria busca un cliente para realizar el cobro sea de inscripción o de ventas realizadas. Se envía el mensaje Obtener Cliente a la clase Gestionar Pago de Cliente. Se envía el mensaje Obtener con parámetro idClie a la clase Cliente. Se envía el mensaje ObtenerMontoDeuda como parámetro el idClie y con una condición blnCredito = true, a la clase Venta. Se envía el mensaje SelectTablaFiltro con sus parámetros para acceder a la base de datos. Se envía el mensaje ObtenerTotalPago con parámetro idClie y con la condición de que blnPagoIns = false. Se envía el mensaje selectTablaFiltro a la clase Datos. Se envía el mensaje ObtenerTotalPagoDeuda para obtener el total de los pagos realizados por inscripción. Se envía el mensaje selectTablaFiltro para recuperar datos dada la condición y los parámetros. La secretaria introduce el monto de cobro y guarda la transacción.

**Figura 11.14** Interfaz de Usuario para el Caso de Uso “Pagar Deuda de Cliente”

### 11.2.7 Caso de Uso “Definir Disciplina”

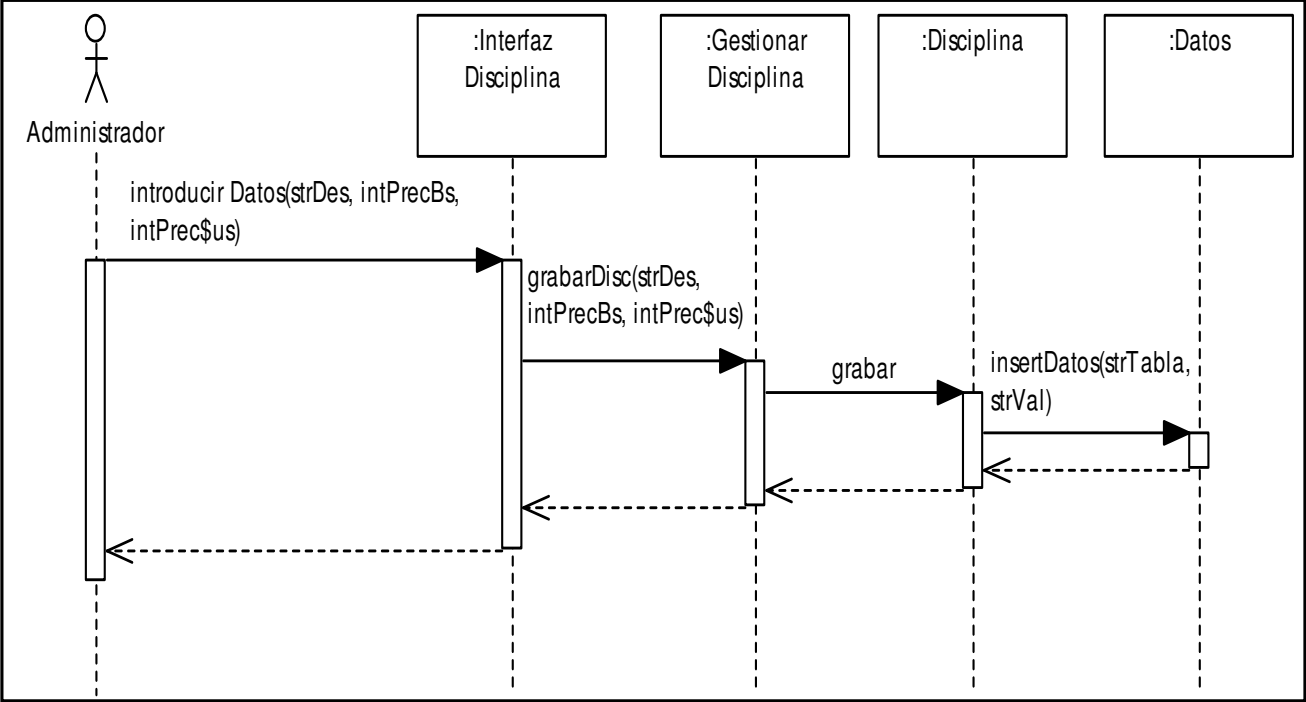


Figura 11.15 Diagrama de secuencia del caso de uso “Definir Disciplina”

El administrador del gimnasio ingresa a una interfaz para definir una disciplina, introduce los datos Descripción, PrecioBs, Precio\$us. La interfaz Disciplina envía el mensaje GrabarDisc con parámetros strDes, intPrecBs, intPrec\$us a la clase Gestionar Disciplina. Se envía el mensaje Grabar a la clase entidad Disciplina. Se envía el mensaje insertDatos con parámetros como strTabla, strVal a la clase Datos, que es la encargada de insertar los datos a la base de datos.

The screenshot shows a window titled "Gestionar Disciplinas" with a table of disciplines and input fields for adding a new one. The table has columns: intIdDi, strDescripcion, intPrecioBs, and intPrecioSus. The input fields are labeled: Código, Descripción, Precio Bs., and Precio \$us.

intIdDi	strDescripcion	intPrecioBs	intPrecioSus
22	Aerobics	161,4	20
23	Spinning	161,4	20
24	Aparatos	121,05	15
25	Sauna	80,7	10
27	Piscina	5	0,62
28	Hidromasaje	10	1,24
29	Hapkido	161,4	20

Código: 24  
Descripción: Aparatos  
Precio Bs.: 121,05  
Precio \$us.: 15

Buttons: Adicionar, Grabar, Cancelar, Modificar, Eliminar, Cerrar

**Figura 11.16** Interfaz de Usuario para el caso de uso “Definir Disciplina”



11.2.8 Caso de Uso “Controlar Asistencia”

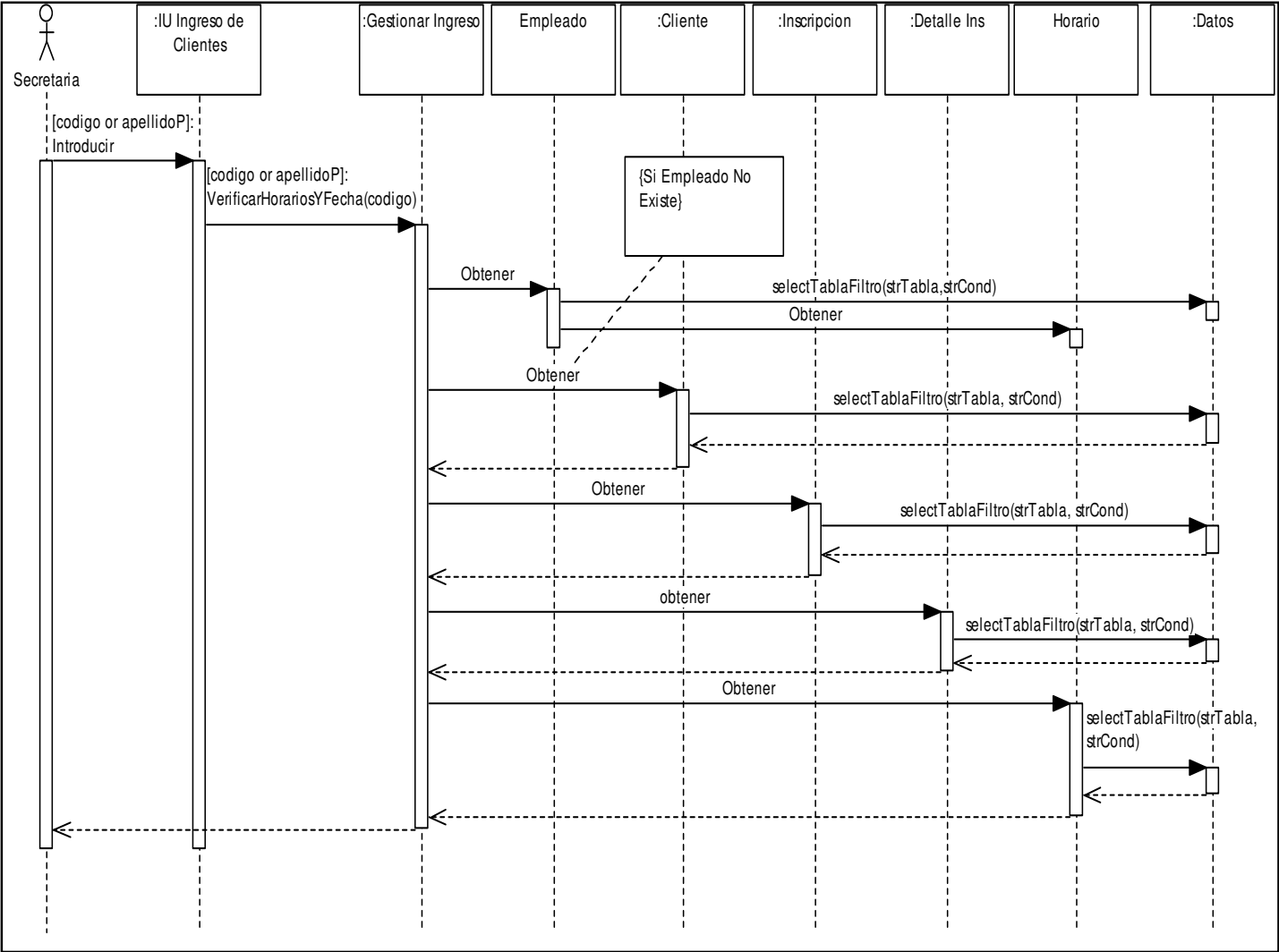


Figura 11.17 Diagrama de secuencia del caso de uso “Controlar Asistencia”

La secretaria introduce el código o busca por apellido paterno al cliente o empleado. Se envía el mensaje VerificarHorariosYFecha a Gestionar Ingreso. Se envía el mensaje Obtener a la clase Empleado primeramente, luego se obtiene el horario asignado al Empleado. Se envía el mensaje obtener a Cliente (si no se encontró a Empleado). Se envía el mensaje selectTablaFiltro con parámetros a la clase Datos que se encarga de acceder a la base de datos. Se envía el mensaje Obtener a la clase Inscripción. Se envía el mensaje selectTablaFiltro con parámetros a la clase Datos para acceder a la base de datos. Se envía el mensaje Obtener a la clase DetalleIns. Se envía el mensaje selectTablaFiltro para obtener todos los detalles. Se envía el mensaje Obtener a la clase Horarios, para recuperar los horarios asignados a la inscripción.

**Control de Asistencia**

**Control de Asistencia a Natural Body**

Código del Cliente:

Nombre:

Apellidos:

Fecha Actual: 09/10/2006

Hora Actual: 23:28:24

**CLIENTE NO REGISTRADO**

Ingresar

*Figura 11.18 Interfaz de Usuario para el caso de uso “Controlar Asistencia”*

### **11.3 DISEÑO DE LAS CLASES.**

El patrón **Representación de Objetos como Tablas** [BW96] propone la definición de una tabla en una BDR por cada clase de objeto persistente. Los atributos de los objetos que contienen tipos de datos primitivos (número, cadena de texto, booleano, etc.) se corresponden con las columnas.

## **CAPITULO 12.**

### **MODELO DE IMPLEMENTACION**

La implementación trata sobre transformar un modelo de diseño en código ejecutable. Con el resultado del modelo de diseño se implementará el sistema en componentes, ficheros de código fuente, scripts y similares.

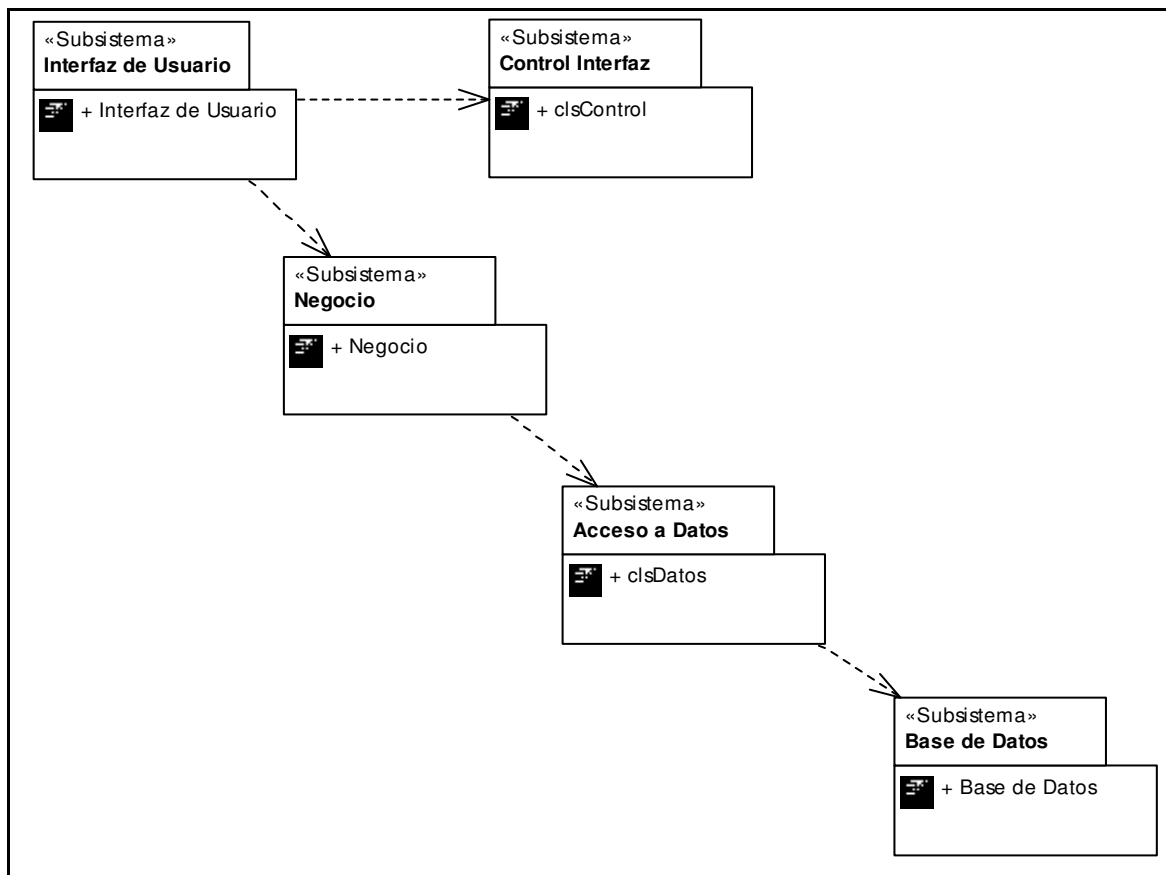
En este modelo también se verá cómo se organizan los componentes de acuerdo al entorno de implementación, el lenguaje de programación utilizado y cómo dependen los componentes unos de otros.

#### **12.1 IMPLEMENTACIÓN DE LA ARQUITECTURA**

En esta sección se identifican los componentes ejecutables y la asignación de los componentes a los nodos en las configuraciones de la red que son relevantes.

### 12.1.1 Identificación de Componentes.

Es importante en el trabajo de implementación, identificar los componentes claves. En este punto los componentes identificados siguen la traza directa del diagrama de subsistemas.



*Figura 12.1 Componentes identificados dentro de cada subsistema*

### 12.1.2 Despliegue de Componentes sobre los Nodos.

Siguiendo la configuración de Nodos identificadas en el modelo de diseño, sección 11.1.1, en esta sección se asigna cada uno de los componentes encontrados, a los diferentes nodos.

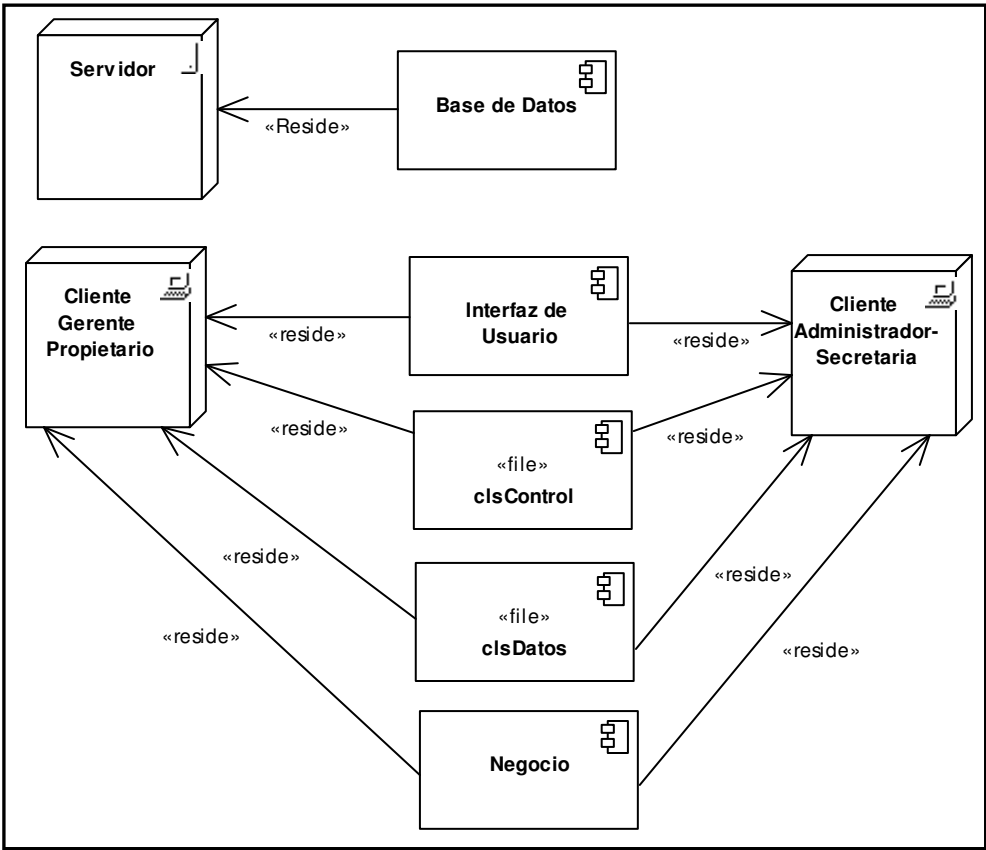


Figura 12.2 Asignación de Componentes a Nodos

12.2 IMPLEMENTACION DE CLASES.

El propósito de la implementación de una clase es implementar una clase de diseño en un componente fichero.

Previamente se elabora una definición de convenciones de programación utilizadas para el proyecto.

Prefijos para controles

Tipo de Control	Prefijo
Check Box	chk
Combo box, drop – down list box	cbo
Command button	cmd

**Capítulo 12.****Modelo de Implementación**

Data grid	grd
List box	lst
Data combo	dbc
Data list	dbl
Form	frm
Frame	fra
Graph	gra
Image	img
Label	lbl
Listview	lvw
Menu	mnu
Radio button	rdo
Picture Box	pic
Text Box	txt

***Prefijos para Objetos de Accesos a Datos (ADO .NET)***

Data Adapter	da
Data Reader	dr
Dataset	ds

***Prefijos para Menu y Submenu***

Menu y Submenu	mnu<<NombreOpcion>>
----------------	---------------------

***Prefijos para alcance de Variables***

Publica	p
Local to Procedure	Ninguno

***Prefijos para tipos de una variable***

Boolean	bln
Byte	byt
Date	dtm
Double	dbl
Integer	int
Long	lng
String	str

Const	C_ <Nombre Mayuscula
Enum	enm

**Prefijos para clase**

Class	cls
Module	mod

**Prefijo para Aplicaciones**

Aplicación de Windows	App<Nombre Aplicación>
-----------------------	------------------------

**Formularios y Reportes**

Formulario Standard	frm<Nombre>
Reportes (Designer)	rpt< Nombre>
Formulario con Reportes	frmRpt<Nombre>

**12.2.1 Implementación de la Clase de Acceso a Datos**

Como se muestra en el diseño se tiene un subsistema de Acceso a Datos, el cual es compuesto por una clase que se encarga de realizar: la conexión a la Base de Datos, cerrar la conexión, actualizar, eliminar, recuperar, consultar, contar registros, verificar existencia de datos, etc.

Se muestra un resumen de la clase:

```
Public Class ClsDatos

#Region "Declaracion de constantes para la conexion"
    Const C_USUARIO = "sa"
    Const C_CONTRASEÑA = "formatec"
#End Region

#Region "Declaración de atributos"

    Private strConexionString As String
    Private strNombreBD As String
    Private strPathBD As String
    Private strUsuario As String
    Private strContraseña As String

    Private Conex As SqlConnection
    Private ConexOle As OleDb.OleDbConnection

    Private Dr As SqlDataReader
    Private Da As SqlDataAdapter
```



```

Private C_DB As String = pstrPathApp &
"\BaseDatos\bdNaturalBody.mdb"
Private C_PC As String = pstrNombrePC

Private C_CONEXION_OLE = "Provider=Microsoft.Jet.OLEDB.4.0;Password='';User
ID=Admin;Data Source=" + C_DB

#End Region

#Region "Declaración de métodos"

Public Property ConexionString() As String
Get
    Return (strConexionString)
End Get
Set (ByVal Value As String)

End Set
End Property

Public Property NombreBD() As String
Get
    Dim pos, cant As Integer
    pos = C_DB.LastIndexOf("\", C_DB.Length)
    cant = C_DB.Length - pos
    strNombreBD = C_DB.Substring(pos, cant)
    Return strNombreBD.Remove(0, 1)
End Get
Set (ByVal strNombreBD As String)

End Set
End Property

Public Property PathBD() As String
Get
    Dim pos As Integer = C_DB.LastIndexOf("\", 1)
    Return strPathBD = C_DB.Substring(0, pos - 1)
End Get
Set (ByVal strNombreBD As String)

End Set
End Property

Public Property PathNombreBD() As String
Get
    Return C_DB
End Get
Set (ByVal strNombreBD As String)

End Set
End Property

'Conecta a la db
Public Sub ConectarDBOLE()
Try
    strConexionString = C_CONEXION_OLE
    ConexOle = New OleDb.OleDbConnection(strConexionString)
    If ConexOle.State = ConnectionState.Open Then
        ConexOle.Close()
    End If

```

```

        ConexOle.Open()

        Catch e As ExecutionEngineException
            MsgBox(e.Message)
        Catch e As Exception
            MsgBox(e.Message)

    End Try
End Sub

Public Sub CerrarDBOLE()
    Try
        ConexOle.Close()
    Catch e As Exception
        MsgBox(e.Message)
    End Try
End Sub

Public Function ExisteDatosOle(ByVal strTabla As String, Optional ByVal
strCondicion As String = "") As Boolean
    Try
        Dim sqlQuery As String
        If strCondicion = "" Then
            sqlQuery = "select * from " + strTabla
        Else
            sqlQuery = "select * from " + strTabla + " where " + strCondicion
        End If

        ConectarDBOLE()
        Dim cmdCommand As New OleDb.OleDbCommand(sqlQuery, ConexOle)

        Dim dr As OleDb.OleDbDataReader = cmdCommand.ExecuteReader()
        If dr.Read Then
            dr.Close()
            CerrarDBOLE()
            Return True
        Else
            dr.Close()
            CerrarDBOLE()
            Return False
        End If

    Catch e As Exception
        MsgBox("Capa:Datos, Función:ExisteDatosOle, Error:" + e.Message)
    Finally

    End Try
End Function

Public Function SelectTablaOLE(ByVal strTabla As String) As
OleDb.OleDbDataReader
    Dim sqlQuery As String
    Dim dr As OleDb.OleDbDataReader
    Try
        sqlQuery = "select * from " & strTabla
        ConectarDBOLE()

        Dim cmdCommand As New OleDb.OleDbCommand(sqlQuery, ConexOle)
        Dim dt As DataTable

```

```

        dr = cmdCommand.ExecuteReader

        Return (dr)

    Catch e As Exception
        MsgBox("Capa:Datos, Función:SelectTablaOLE, Error: " + e.Message)
    Finally
        'CerrarDBSQL()
    End Try

End Function

Public Function SelectTablaFiltroOLE(ByVal strTabla As String, ByVal
strcondicion As String) As OleDb.OleDbDataReader
    Dim sqlQuery As String
    Dim dr As OleDb.OleDbDataReader
    Try
        sqlQuery = "select * from " & strTabla & " where " & strcondicion
        ConectarDBOLE()

        Dim cmdCommand As New OleDb.OleDbCommand(sqlQuery, ConexOle)
        dr = cmdCommand.ExecuteReader()
        Return (dr)
        CerrarDBOLE()
    Catch e As Exception
        MsgBox("Capa:Datos, Función:SelectTablaFiltroOLE, Error: " +
e.Message)
    Finally
        'CerrarDBSQL()
    End Try

End Function

Public Function CantidadFilas(ByVal strTabla As String, ByVal strcondicion As
String) As Integer
    Dim sqlQuery As String
    Dim ds As New DataSet()
    Dim dt As New DataTable()
    Dim fila As DataRow
    Dim c As Integer
    Try
        sqlQuery = "select * from " & strTabla & " where " & strcondicion
        ConectarDBSQL()
        Dim x As New SqlClient.SqlDataAdapter(sqlQuery, Conex)

        x.Fill(dt)
        c = dt.Rows.Count

        CerrarDBSQL()
        CantidadFilas = c
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

End Function

Public Sub EliminarDatosOle(ByVal strTabla As String, ByVal strCondicion As
String)
    Dim sqlEliminar As String
    Try
        ConectarDBOLE()
        sqlEliminar = "delete from " & strTabla & " where " & strCondicion
        Dim cmdCommand As New OleDb.OleDbCommand(sqlEliminar, ConexOle)
        cmdCommand.ExecuteNonQuery()
        CerrarDBOLE()
    
```

```

        Catch e As Exception
            MsgBox("Capa:Datos, Función:EliminarDatosOle, Error: " + e.Message)
        End Try
    End Sub

    Public Function GetUltimoRegistroOLE(ByVal strTabla As String, ByVal strCampo
As String, Optional ByVal strCond As String = "") As Integer
        Dim sqlSelect As String
        Try
            ConectarDBOLE()

            If strCond = "" Then
                sqlSelect = "select max(" & strCampo & ") from " & strTabla
            Else
                sqlSelect = "select max(" & strCampo & ") from " & strTabla & "
where " & strCond
            End If

            Dim cmdCommand As New OleDb.OleDbCommand(sqlSelect, ConexOle)

            Dim dr As OleDb.OleDbDataReader =
cmdCommand.ExecuteReader(CommandBehavior.SingleRow)

            If dr.Read = False Then
                GetUltimoRegistroOLE = 0
            Else
                If dr.IsDBNull(0) Then
                    GetUltimoRegistroOLE = 0
                Else
                    GetUltimoRegistroOLE = dr.Item(0)
                End If
            End If
            CerrarDBOLE()
        Catch e As Exception
            MsgBox("Capa:Datos, Función:GetUltimoRegistroOLE, Error: " +
e.Message)
        End Try
    End Function

    Public Sub InsertarDatosOLE(ByVal strTabla As String, ByVal strCampos As
String, ByVal strValores As String)
        Dim valor = "", sqlInsert As String
        Dim obj As Object
        Dim i As Integer

        Try
            ConectarDBOLE()

            sqlInsert = "insert into " & strTabla & "(" & strCampos & ")" & "
values(" & strValores & ")"
            Dim cmdCommand As New OleDb.OleDbCommand(sqlInsert, ConexOle)

            cmdCommand.ExecuteNonQuery()
            CerrarDBOLE()
        Catch e As Exception
            MsgBox("Capa:Datos, Función:InsertarDatos, Error: " & e.Message)
        End Try
    End Sub

```

```

Public Sub InsertarDTToDbOLE(ByVal dtabla As DataTable, ByVal strCampos As
String, ByVal strTabla As String)
    Dim sql As String
    Try
        ConectarDBOLE()
        sql = "select " & strCampos & " from " & strTabla
        Dim da As New OleDb.OleDbDataAdapter(sql, ConexOle)
        Dim cb As OleDb.OleDbCommandBuilder
        cb = New OleDb.OleDbCommandBuilder(da)
        da.SelectCommand.CommandText = sql
        da.InsertCommand = cb.GetInsertCommand
        da.Update(dtabla)
        CerrarDBOLE()
    Catch e As Exception
        MsgBox("Capa:Datos, Función:InsertarDTToDbOLE, Error: " & e.Message)
    End Try

End Sub

Public Overloads Sub ActualizarDatosOLE(ByVal strTabla As String, ByVal
strCampos As String, ByVal strCondicion As String)
    Dim valor = "", sqlUpdate As String
    Dim obj As Object

    Try
        ConectarDBOLE()
        sqlUpdate = "Update " & strTabla & " set " & strCampos & " where " &
strCondicion
        Dim cmdCommand As New OleDb.OleDbCommand(sqlUpdate, ConexOle)
        cmdCommand.ExecuteNonQuery()
        CerrarDBOLE()
    Catch e As Exception
        MsgBox("Capa:Datos, Función:Actualizar Datos OLE, Error: " &
e.Message)
    End Try
End Sub

Public Overloads Sub ActualizarDatosOLE(ByVal strTabla As String, ByVal
strCampos As String)
    Dim valor = "", sqlUpdate As String
    Dim obj As Object
    Dim i As Integer

    Try
        ConectarDBOLE()
        sqlUpdate = "Update " & strTabla & " set " & strCampos
        Dim cmdCommand As New OleDb.OleDbCommand(sqlUpdate, ConexOle)
        cmdCommand.ExecuteNonQuery()
        CerrarDBOLE()
    Catch e As Exception
        MsgBox("Capa:Datos, Función:Actualizar Datos OLE, Error: " &
e.Message)
    End Try
End Sub
#End Region

End Class

```

## 12.2.2 Implementación de la clase Control de Interfaz.

Como se muestra en el diseño en la figura 12.1, se tiene un subsistema de Control de Interfaz, algunas de las tareas que realiza esta clase son las de limpiar los controles de texto, cargar listas, cargar combo, etc.

Se muestra un resumen de esta clase.

```
Public Class clsControl

    Public Shared Sub BuscarComboOle(ByVal combo As ComboBox, ByVal tabla As String,
    ByVal cond As String, ByVal campoReturn As String)
        Try
            Dim objDat As New ClsDatos()
            Dim dr As OleDb.OleDbDataReader
            dr = objDat.SelectTablaFiltroOLE(tabla, cond)
            If dr.Read Then
                combo.Text = dr.Item(campoReturn)
            End If
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub

    Public Shared Sub ComboAutoCompleteKeyUp(ByVal sender As Object, ByVal e As
    System.Windows.Forms.KeyEventArgs)
        Dim CB As ComboBox = DirectCast(sender, ComboBox)
        Dim inicio As Integer = CB.Text.Length
        Dim indice As Integer
        Dim t As String = e.KeyCode.ToString
        If t.Length = 1 AndAlso (IsNumeric(t) OrElse (t >= "A" AndAlso t <= "Z") _
        OrElse InStr("/*-+()!#$%&=?;¿~{}[] , . - _ < > @ | ° ~", t) > 0)
        Then
            Dim p As Integer = CB.SelectionStart
            indice = CB.FindString(CB.Text.Substring(0, p))
            If indice >= 0 Then
                CB.SuspendLayout()
                CB.SelectedIndex = indice
                CB.Select(p, CB.Text.Length)
                CB.ResumeLayout()
            End If
        End If
    End Sub

    Friend Shared Sub ComboAutoComplete(ByVal Combo As ComboBox)
        AddHandler Combo.KeyUp, AddressOf ComboAutoCompleteKeyUp
    End Sub

    Public Shared Sub LimpiarControles(ByVal ctrl As Control)
        Dim Controles As New Control()
        For Each Controles In ctrl.Controls
            If TypeOf Controles Is TextBox Then
                Controles.Text = ""
            End If
        End For
    End Sub
```

```

        If TypeOf Controles Is Date Then
            Controles.Text = ""
        End If
        If TypeOf Controles Is ComboBox Then
            Controles.Text = ""
        End If
    Next
End Sub

Public Shared Sub cargarLista(ByVal lista As ListBox, ByVal dt As DataTable)
    Try
        Dim intLimite As Integer
        intLimite = dt.Rows.Count
        lista.DataSource = dt.DataSet
        lista.Items.Clear()
        Dim i, aux As Integer
        aux = intLimite - 1
        For i = 0 To aux
            lista.Items.Add(dt.Rows.Item(i).Item("strDescripcion"))
        Next
    Catch ex As Exception
        MsgBox("Clase:Control," & "Funcion: Cargar Grid, " & "Error: " &
            ex.Message)
    End Try
End Sub

Public Shared Sub CargarForm(ByVal f As Form, Optional ByVal dialog As Boolean =
False, Optional ByVal TaskBar As Boolean = False, Optional ByVal Parent As Form =
Nothing, Optional ByVal Max As Boolean = False)
    Try
        f.MdiParent = Parent
        f.ShowInTaskbar = TaskBar
        If Max = True Then
            f.WindowState = FormWindowState.Maximized
        End If

        If Not dialog Then
            f.Show()
        Else
            f.ShowDialog(f.Owner)
        End If
    Catch ex As Exception
        MsgBox(ex.Message, MsgBoxStyle.OKOnly, "Error")
    End Try
End Sub

Public Shared Sub cargarGrid(ByVal grid As DataGrid, ByVal dt As DataTable)
    Try
        grid.DataSource = dt
        Dim da As OleDb.OleDbDataAdapter
        If dt Is Nothing Then
            grid.DataSource = dt
            Exit Sub
        End If
        If dt.Rows.Count > 0 Then
            grid.Select()
        End If
    Catch ex As Exception
        MsgBox("Clase:Control," & "Funcion: Cargar Grid, " & "Error: " &
            ex.Message)
    End Try
End Sub

```

```
Public Shared Sub cargarComboDataSource(ByVal combo As ComboBox, ByVal dat As
DataTable, ByVal desc As String, ByVal id As String)
    Dim strDescripcion As String
    Dim strCodigo As String
    Try
        combo.DataSource = Nothing
        combo.Items.Clear()
        combo.DataSource = dat
        combo.DisplayMember = desc
    Catch ex As Exception
        MsgBox("Clase:Control," & "Funcion: Cargar Combo, " & "Error: " &
ex.Message)
    End Try
End Sub

Public Shared Sub cargarCombo(ByVal combo As ComboBox, ByVal dat As Object, ByVal
desc As String, ByVal id As String)
    Dim strDescripcion As String
    Dim strCodigo As String
    Try
        combo.Items.Clear()
        While dat.read
            strDescripcion = dat.Item(desc)
            strCodigo = dat.Item(id)
            combo.Items.Add(New clsItemBox(strDescripcion, strCodigo))
        End While
    Catch ex As Exception
        MsgBox("Clase:Control," & "Funcion: Cargar Combo, " & "Error: " &
ex.Message)
    End Try

End Sub

Public Shared Sub BlogDesbControles(ByVal ctrl As Control, ByVal modo As
EstadoBotones)
    Dim Controles As New Control()
    Dim est As Boolean
    If modo = EstadoBotones.edit Or modo = EstadoBotones.nuevo Then
        est = True
    Else
        est = False
    End If

    For Each Controles In ctrl.Controls
        If TypeOf Controles Is TextBox Then
            Controles.Enabled = est
        End If
        If TypeOf Controles Is Date Then
            Controles.Enabled = est
        End If
        If TypeOf Controles Is ComboBox Then
            Controles.Enabled = est
        End If

        If TypeOf Controles Is GroupBox Then
            Controles.Enabled = est
        End If

        If TypeOf Controles Is CheckBox Then
            Controles.Enabled = est
        End If
    Next
End Sub
```



```

Public Shared Sub BloqDesbBotones(ByVal ctrl As Control)
    Dim c As New Control()

    For Each c In ctrl.Controls
        If TypeOf c Is Button Then
            c.Enabled = Not c.Enabled
        End If
    Next
End Sub

Public Shared Sub BloqDesbMenu(ByVal menu As Menu)
    Dim mnuItem As MenuItem

    For Each mnuItem In menu.MenuItems
        mnuItem.Enabled = Not mnuItem.Enabled
    Next
End Sub

End Class

```

### 12.2.3 Implementación de clase Entidad “Cliente”.

Como se muestra en el diseño figura 12.1, se tiene un subsistema compuesto por las entidades del negocio. Se muestra la implementación de la clase Cliente.

```

Public Class clsCliente
    Private intIdCliente As Integer
    Private strNombre As String
    Private strApellidoP As String
    Private strApellidoM As String
    Private dtmFechaNac As Date
    Private intEdad As Integer
    Private strCI As String
    Private strDireccion As String
    Private strTelDom As String
    Private strTelCel As String
    Private strProfesion As String
    Private strLugarTrabajo As String
    Private strTelOf As String
    Private strEstadoCivil As String
    Private intNroHijos As Integer
    Private dtmFechaIngreso As Date
    Private strCorreo As String

    Private enuModo As EstadoBotones

    Property IdCliente()
        Get
            Return intIdCliente
        End Get
        Set(ByVal Value)
            intIdCliente = Value
        End Set
    End Property

```

```
Property Nombre()  
    Get  
        Return strNombre  
    End Get  
    Set(ByVal Value)  
        strNombre = Value  
    End Set  
End Property  
  
Property ApellidoP()  
    Get  
        Return strApellidoP  
    End Get  
    Set(ByVal Value)  
        strApellidoP = Value  
    End Set  
End Property  
  
Property ApellidoM()  
    Get  
        Return strApellidoM  
    End Get  
    Set(ByVal Value)  
        strApellidoM = Value  
    End Set  
End Property  
  
Property FechaNac()  
    Get  
        Return dtmFechaNac  
    End Get  
    Set(ByVal Value)  
        dtmFechaNac = Value  
    End Set  
End Property  
  
Property Edad()  
    Get  
        Return intEdad  
    End Get  
    Set(ByVal Value)  
        intEdad = Value  
    End Set  
End Property  
  
Property CI()  
    Get  
        Return strCI  
    End Get  
    Set(ByVal Value)  
        strCI = Value  
    End Set  
End Property  
  
Property Direccion()  
    Get  
        Return strDireccion  
    End Get  
    Set(ByVal Value)  
        strDireccion = Value  
    End Set  
End Property
```

```
Property TelDom()  
    Get  
        Return strTelDom  
    End Get  
    Set(ByVal Value)  
        strTelDom = Value  
    End Set  
End Property  
  
Property TelCel()  
    Get  
        Return strTelCel  
    End Get  
    Set(ByVal Value)  
        strTelCel = Value  
    End Set  
End Property  
  
Property Profesion()  
    Get  
        Return strProfesion  
    End Get  
    Set(ByVal Value)  
        strProfesion = Value  
    End Set  
End Property  
  
Property LugarTrabajo()  
    Get  
        Return strLugarTrabajo  
    End Get  
    Set(ByVal Value)  
        strLugarTrabajo = Value  
    End Set  
End Property  
  
Property TelOf()  
    Get  
        Return strTelOf  
    End Get  
    Set(ByVal Value)  
        strTelOf = Value  
    End Set  
End Property  
  
Property EstadoCivil()  
    Get  
        Return strEstadoCivil  
    End Get  
    Set(ByVal Value)  
        strEstadoCivil = Value  
    End Set  
End Property  
  
Property NroHijos()  
    Get  
        Return intNroHijos  
    End Get  
    Set(ByVal Value)  
        intNroHijos = Value  
    End Set  
End Property
```

```
Property FechaIngreso()  
    Get  
        Return dtmFechaIngreso  
    End Get  
    Set (ByVal Value)  
        dtmFechaIngreso = Value  
    End Set  
End Property  
  
Property Correo()  
    Get  
        Return strCorreo  
    End Get  
    Set (ByVal Value)  
        strCorreo = Value  
    End Set  
End Property  
  
Public Overloads Sub Cargar()  
    Try  
        Dim objDat As New ClsDatos()  
        Dim dr As OleDb.OleDbDataReader  
  
        dr = objDat.SelectTablaOLE("tblCliente")  
        If dr.Read = True Then  
            intIdCliente = dr.Item("intIdCliente")  
            strNombre = dr.Item("strNombre")  
            strApellidoP = dr.Item("strApellidoP")  
            strApellidoM = dr.Item("strApellidoM")  
            dtmFechaNac = dr.Item("dtmFechaNac")  
            intEdad = dr.Item("intEdad")  
            strDireccion = dr.Item("strDireccion")  
            strTelDom = dr.Item("strTelDom")  
            strTelCel = dr.Item("strTelCel")  
            strTelOf = dr.Item("strTelOf")  
            strProfesion = dr.Item("strProfesion")  
            strLugarTrabajo = dr.Item("strLugarTrabajo")  
            strEstadoCivil = dr.Item("strEstadoCivil")  
            intNroHijos = dr.Item("intNroHijos")  
            dtmFechaIngreso = dr.Item("dtmFechaIngreso")  
            strCorreo = dr.Item("strCorreo")  
        Else  
            intIdCliente = 0  
            strNombre = ""  
            strApellidoP = ""  
            strApellidoM = ""  
            dtmFechaNac = Now.Date  
            intEdad = 0  
            strDireccion = ""  
            strTelDom = ""  
            strTelCel = ""  
            strTelOf = ""  
            strProfesion = ""  
            strLugarTrabajo = ""  
            strEstadoCivil = ""  
            intNroHijos = ""  
            dtmFechaIngreso = Now.Date  
            strCorreo = " @ "  
        End If  
    End Try  
End Sub
```

```

        Catch ex As Exception
            MsgBox("Capa:Datos, Función:Cargar, Error: " + ex.Message)
        End Try

    End Sub

    Public Sub CargarCombo(ByVal combo As ComboBox)
        Try
            Dim objDat As New ClsDatos()
            clsControl.cargarCombo(combo, objDat.SelectTablaOLE("tblCliente"),
                                   "strApellidoP", "intIdCliente")
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub

    Public Overloads Sub Cargar(ByVal id As Integer)
        Try
            Dim strCond As String
            strCond = "intIdCliente=" & id
            Dim objDat As New ClsDatos()
            Dim dr As OleDb.OleDbDataReader

            dr = objDat.SelectTablaFiltroOLE("tblCliente", strCond)
            If dr.Read = True Then
                intIdCliente = dr.Item("intIdCliente")
                strNombre = dr.Item("strNombre")
                strApellidoP = dr.Item("strApellidoP")
                strApellidoM = dr.Item("strApellidoM")
                dtmFechaNac = dr.Item("dtmFechaNac")
                intEdad = dr.Item("intEdad")
                strDireccion = dr.Item("strDireccion")
                strTelDom = dr.Item("strTelDom")
                strTelCel = dr.Item("strTelCel")
                strTelOf = dr.Item("strTelOf")
                strProfesion = dr.Item("strProfesion")
                strLugarTrabajo = dr.Item("strLugarTrabajo")
                strEstadoCivil = dr.Item("strEstadoCivil")
                intNroHijos = dr.Item("intNroHijos")
                dtmFechaIngreso = dr.Item("dtmFechaIngreso")
                strCorreo = dr.Item("strCorreo")
            Else
                intIdCliente = 0
                strNombre = ""
                strApellidoP = ""
                strApellidoM = ""
                dtmFechaNac = Now.Date
                intEdad = 0
                strDireccion = ""
                strTelDom = ""
                strTelCel = ""
                strTelOf = ""
                strProfesion = ""
                strLugarTrabajo = ""
                strEstadoCivil = ""
                intNroHijos = ""
                dtmFechaIngreso = Now.Date
                strCorreo = " @ "
            End If

        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub

```

```

Function Grabar() As Boolean
    Dim strCampos As String
    Dim strValor As String

    Try
        Dim objDat As New ClsDatos()
        If strNombre = "" Then
            MsgBox("El nombre no puede estar en blanco")
            Return False
        End If

        If strApellidoP = "" Then
            MsgBox("El Apellido Paterno no puede estar en blanco")
            Return False
        End If
        If strDireccion = "" Then
            MsgBox("La Direccion no puede estar en blanco")
            Return False
        End If

        strCampos =
"strNombre,strApellidoP,strApellidoM,dtmFechaNac,intEdad,strCI" & _
" ,strDireccion,strTelDom,strTelCel,strProfesion,strLugarTrabajo" & _
" ,strTelOf,strEstadoCivil,intNroHijos,dtmFechaIngreso,strCorreo"

        strValor = "'" & strNombre & "'," & _
            "'" & strApellidoP & "'," & _
            "'" & strApellidoM & "'," & _
            "'" & dtmFechaNac & "'," & _
                CStr(intEdad) & "," & _
            "'" & strCI & "'," & _
            "'" & strDireccion & "'," & _
            "'" & strTelDom & "'," & _
            "'" & strTelCel & "'," & _
            "'" & strProfesion & "'," & _
            "'" & strLugarTrabajo & "'," & _
            "'" & strTelOf & "'," & _
            "'" & strEstadoCivil & "'," & _
                CStr(intNroHijos) & "," & _
            "'" & dtmFechaIngreso & "'," & _
            "'" & strCorreo & "'"

        objDat.InsertarDatosOLE("tblCliente", strCampos, strValor)
        intIdCliente = objDat.GetUltimoRegistroOLE("tblCliente", "intidCliente")
        objDat = Nothing

        Return True

    Catch ex As Exception
        MsgBox("Clase:clsCliente, Método Grabar" & "Error:" & ex.Message)
        Return False
    Finally
        End Try
    End Function

    Public Sub Eliminar(ByVal intIdEnt As Integer)
        Try
            Dim objAntec As New clsAntecedente()
            Dim objDieta As New clsDieta()
            Dim objDat As New ClsDatos()
            Dim strCondicion As String

            If MsgBox("Esta seguro que desea eliminar toda la informacion del
Cliente?",MsgBoxStyle.YesNo, "Advertencia") = MsgBoxResult.Yes Then
                strCondicion = "intIdCliente = " & intIdEnt
                objAntec.Eliminar(intIdEnt)
            End If
        End Try
    End Sub

```

```

        objDieta.Eliminar(intIdEnt)
        objDat.EliminarDatosOle("tblCliente", strCondicion)
    End If

    objDat = Nothing
Catch ex As Exception
    MsgBox("Error al Eliminar. " & ex.Message)
End Try
End Sub

Function Actualizar() As Boolean
    Try
        Dim strCampos As String
        Dim objDat As New ClsDatos()
        If strNombre = "" Then
            MsgBox("El Nombre del Cliente no puede estar en blanco")
            Return False
        End If

        strCampos = "strNombre = '" & strNombre & "', strApellidoP='" &
strApellidoP & "', strApellidoM='" & _
strApellidoM & "', dtmFechaNac=#" & dtmFechaNac & "#, intEdad=" & intEdad
& ",strCI='" & strCI & "'," & _
        "strDireccion = '" & strDireccion & "', strTelDom = '" & strTelDom & "',
strTelCel = '" & strTelCel & "'," & _
        "strProfesion = '" & strProfesion & "', strLugarTrabajo = '" &
strLugarTrabajo & "'," & _
        "strTelOf = '" & strTelOf & "', strEstadoCivil = '" & strEstadoCivil &
        "', intNroHijos = " & intNroHijos & "," & _
        "dtmFechaIngreso = #" & dtmFechaIngreso & "#, strCorreo = '" & strCorreo
& "'"

        objDat.ActualizarDatosOLE("tblCliente", strCampos, "intIdCliente=" &
intIdCliente.ToString)
        Return True
    Catch ex As Exception
        MsgBox("Error al Actualizar: " & ex.Message)
        Return False
    End Try
End Function
End Class

```

### 12.2.4 Implementación de operaciones.

La implementación realizada en esta sección incluye de algunas clases que son relevantes en el sistema **GimnaSoft**.

```

Public Function CerrarCaja(ByVal dtmFechaCerrar As Date, ByVal dtmFechaAbrir As Date)
As Boolean
    If dtmFechaAbrir < dtmFechaCerrar Then
        MsgBox("La Fecha de Apertura tiene que ser mayor o igual que la Fecha de
Cierre")
        Return False
    End If

    Dim objDat As New ClsDatos()
    objDat.ActualizarDatosOLE("tblCaja", "blnCerrada=True", "dtmFecha= #" &
dtmFechaCerrar & "#")
    MsgBox("Caja Cerrada", MsgBoxStyle.Information, "Mensaje")
    Return True
End Function

```

```
Public Sub AbrirCaja(ByVal dtmFechaAbrir As String)
    Dim objDat As New ClsDatos()
    Dim strVal As String
    strVal = "#" & dtmFechaAbrir & "#," & "" & strDetalle & "," & intSaldoBs
    objDat.InsertarDatosOLE("tblCaja", "dtmFecha,strDetalle,intSaldoBs", strVal)
    MsgBox("Caja Abierta", MsgBoxStyle.Information, "Mensaje")
End Sub
```

### 12.2.5 Otras implementaciones.

Ver en **Anexo B: Implementación de Clases**, cómo la clase **clsBaseDatos** realiza la copia de seguridad, la clase **clsCifrarDescifrar** encripta y desencripta contraseña de usuarios. La implementación de un formulario genérico para mostrar los diferentes reportes.



## CAPITULO 13.

### MODELO DE PRUEBAS

Las pruebas realizadas al software *Gimnasoft* se dividieron en dos partes, las pruebas a nivel estático y las pruebas a nivel dinámico.

#### 13.1 PRUEBAS DE UNIDAD.

A nivel estático se realizaron las pruebas de unidad. El propósito de realizar la prueba de unidad es probar los componentes implementados como unidades individuales.

Se realizaron pruebas de Caja Blanca a medida que se desarrollaba el código y se aseguró que todas las sentencias se ejecutaran. Para las Pruebas de Caja Negra, se muestra el resultado de algunas de ellas. Se toma como referencia los casos de uso a nivel de diseño.

Caso de Uso: Registrar Venta				
Valor Introducidos	Acción	Clases	Esperado	Resultado
Campos en blanco	Grabar	Interfaz Venta , Gestionar Venta	Mensaje de Advertencia	“Seleccione un Cliente”
Cliente Seleccionado	Grabar	Interfaz Venta , Gestionar	Mensaje de	“Seleccione un producto”

**Capítulo 13.****Modelo de Pruebas**

		Venta	Advertencia	
Producto Seleccionado	Grabar	Interfaz Venta , Gestionar Venta	Mensaje de Advertencia	“Ingrese una cantidad”
Cantidad Ingresada	Grabar	Interfaz Venta , Gestionar Venta	Mensaje de Advertencia	“Agregue un detalle”
Detalle Agregado	Grabar	Interfaz Venta , Gestionar Venta	Mensaje de Advertencia	“Introduzca el efec
Efectivo Introducido	Grabar	Interfaz Venta , Gestionar Venta	Cambio Calculado (CC)	CC = Efectivo - PrecioT

<b>Caso de Uso: Abrir Caja</b>				
<b>Valor Introducidos</b>	<b>Acción</b>	<b>Clases</b>	<b>Esperado</b>	<b>Resultado</b>
Campos en blanco	Abrir	Interfaz Caja, Gestionar Caja	Mensaje de Advertencia	“Ingrese el Monto Inicial, la Descripción y una Fecha de Apertura”
Datos Introducidos	Abrir	Interfaz Caja, Gestionar Caja, Entidad Caja, clsDatos	Mensaje de Aviso, Registro agregado a la tabla Caja.	“Caja Abierta”, Registro insertado.

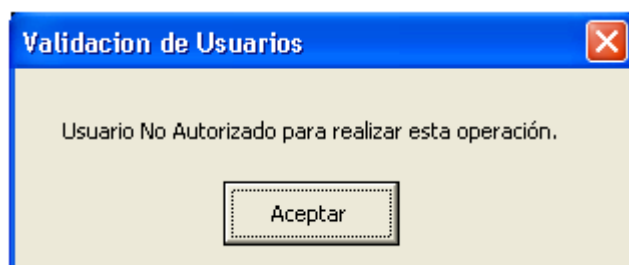
<b>Caso de Uso Cerrar Caja</b>				
<b>Valor Introducidos</b>	<b>Acción</b>	<b>Clases</b>	<b>Esperado</b>	<b>Resultado</b>
Ninguno	Cerrar	Interfaz, Gestionar Caja	Revisión de Procesos pendientes	“Cierre todas las ventanas para realizar el cierre”
Ninguno	Cerrar	Interfaz, Gestionar Caja, Entidad Caja, clsDatos	Mensaje de Aviso, Registro de Cajas cerrado	Se cambiaron de estado las transacciones en caja. A “Cerrada”. Mensaje de Aviso “Caja Cerrada”,

Caso de Uso: Pagar a Empleado				
Valor Introducidos	Acción	Clases	Esperado	Resultado
Campos en blanco	Grabar	Interfaz, Control	Mensaje de Advertencia	“Seleccione un Empleado”
Empleado Seleccionado	Grabar	Interfaz, Control	Mensaje de Advertencia	“Ingrese el Monto de Pago”
Monto ingresado mayor a Sueldo	Grabar	Interfaz, Control	Mensaje de Advertencia	“El Monto Ingresado es mayor que el Sueldo definido”

A nivel dinámico se realizaron las Pruebas de Integración y las Pruebas de Sistema.

## 13.2 PRUEBAS DE INTEGRACION.

- ✓ **Niveles de Usuario.** Dado que el software trabaja a nivel multiusuario, se comprobó que los diferentes niveles de usuarios se validaran correctamente, antes de ingresar a las opciones autorizadas. En las figura 13.1, se muestran el resultado obtenido.



**Figura 13.1 Validación de usuarios**

- ✓ **Reportes.** Se comprobó que la salida de los reportes reflejen exactamente lo que se haya definido en los filtros.
- ✓ **Módulos Integrados.** Como se ha mostrado el software, maneja los ingresos y egresos de caja, lo cual significa que los datos registrados como ser Pagos, Adelantos, Ventas, Compras, y otros que se realizan en módulos separados, sean estos reflejados en el modulo de Caja.

### 13.3 PRUEBAS DE SISTEMA.

En las pruebas a nivel dinámico, se realizaron las siguientes actividades:

- ✓ **Base de Datos Vacía.** Cuando se inicia las actividades con el software es normal que la base de datos se inicie sin datos. El software ha sido probado de esta forma, obteniendo su funcionalidad de forma integral.
- ✓ **Sistema Operativo Diferente.** Se realizó la instalación del software en una computadora con sistema operativo Windows 2000, sin tener el office instalado, y el sistema se instaló correctamente, se realizó las pruebas en los Reportes y éstos se desplegaron correctamente.
- ✓ **Niveles de Carga.** Se implementó un modulo de prueba para sobrecargar la base de datos con más de 5000 registros en las siguientes tablas: Empleados, Clientes y Caja. Posteriormente se ejecutó el software y éste trabajo sin problema alguno.

En la figura 13.2, se muestra la tabla Caja cargada con datos de prueba, con círculo rojo se muestra la cantidad de registros cargados.

tblCaja : Tabla

	intIdCaja	intIdUsuario	strDetalle	dtmFecha	intIngresoBs
	209	0	xx	01/01/1900	100
▶	210	0	xx	01/01/1900	100
	211	0	xx	01/01/1900	100
	212	0	xx	01/01/1900	100
	213	0	xx	01/01/1900	100
	214	0	xx	01/01/1900	100
	215	0	xx	01/01/1900	100
	216	0	xx	01/01/1900	100
	217	0	xx	01/01/1900	100
	218	0	xx	01/01/1900	100
	219	0	xx	01/01/1900	100
	220	0	xx	01/01/1900	100
	221	0	xx	01/01/1900	100
	222	0	xx	01/01/1900	100
	223	0	xx	01/01/1900	100
	224	0	xx	01/01/1900	100
	225	0	xx	01/01/1900	100
	226	0	xx	01/01/1900	100

Registro: 14 de 5011

Figura 13.2 Tabla Caja, cargada con más de 5000 registros.

En la figura 13.3, se muestra en el Administrador de Tareas el nivel de carga de la CPU y el uso de la memoria, tras acceder al modulo de caja y realizar la carga de la tabla.

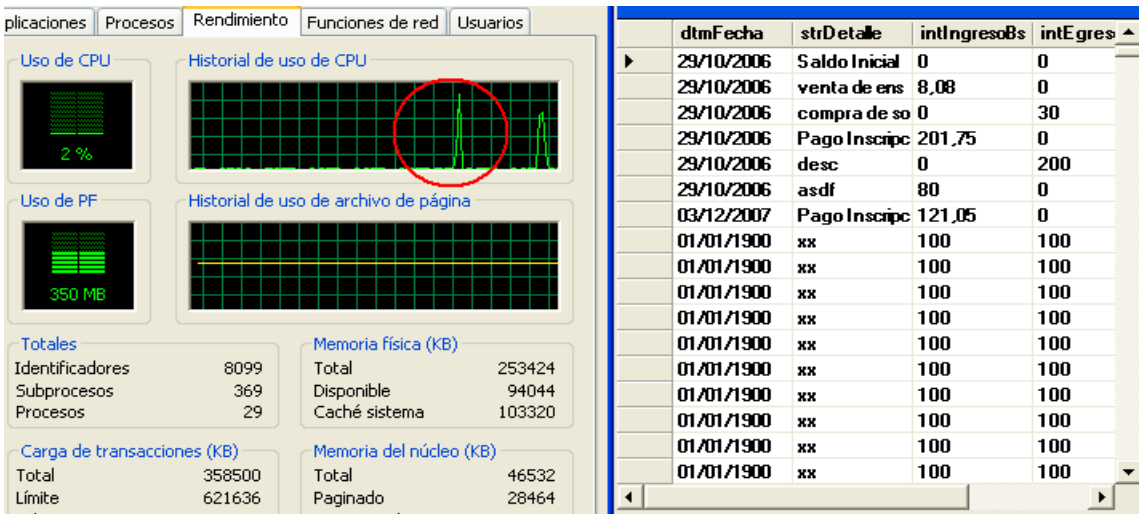


Figura 13.3 Uso de la CPU al cargar datos de Caja.

✓ **Configuración del Sistema Operativo.**

- **Fecha Atrasada.** Dado que el software en todas sus opciones maneja la fecha como parámetro de las transacciones y como parámetro para realizar el seguimiento de las actividades físicas, es importante entonces que la fecha que se obtenga del software de base esté correctamente definida.
- **Colores de Personalización diferentes.** Se cambio los colores del Windows y se comprobó que estos no afectan a los colores establecidos al sistema, ya que se colocó colores que respeten la propia configuración.
- **Resolución de Pantalla.** Se cambiaron las configuraciones de la pantalla de 800x600 y luego a 1024x768, para verificar que las pantallas logren ingresar en el tamaño adecuado y el texto en las mismas no sea alterado.
- **Uso de Memoria.** El lenguaje Visual Basic .Net posee dentro de su Framework, la capa Runtime del Lenguaje Común (CLR), que es la responsable de los servicios básicos de .Net, tales como la administración de memoria, la recolección de elementos no utilizados y otros, una gran ventaja del lenguaje que no se poseen otros. Por otro lado, en el código fuente se ha tenido muy en cuenta la liberación de recursos y destrucción de objetos cuando ya no son utilizados. De esta manera se asegura que el software realiza un tratamiento óptimo de la memoria de la computadora, dando así mayor rapidez para realizar las operaciones.

## CONCLUSIONES

Se logró recabar, estudiar la documentación y tener las entrevistas necesarias con el cliente para entender el dominio del problema.

En el desarrollo de un Sistema de Información es importante mantener una relación de cerca y directa con las personas involucradas en el dominio del problema, para lograr entender los verdaderos requerimientos y además armonizar la lógica y terminología del Cliente con las personas encargadas del Desarrollo del Software. La relación directa tiene que ser de forma periódica ya que los requerimientos del negocio pueden llegar a cambiar y por tanto, estos pueden quedar fuera del modelo diseñado que tiene el software. Todo lo anteriormente descrito, se ha cumplido, ya que se aplicó el Proceso Unificado de Desarrollo que permite realizar las diferentes fases del software a través de iteraciones.

Se logró diseñar una base de datos que permite el almacenamiento y recolección de información sin complicaciones y de forma consistente.

El software fue diseñado para lograr una facilidad de uso, unificando algunas de las tareas comunes en un solo formulario, como las búsquedas, como las compras, ventas, pago de sueldo y los ingresos / egresos en solo ambiente. Así también el software ha sido implementado utilizando el patrón de capas, dividiendo así las funcionalidades del software, y por tanto esto permitirá dar mayor mantabilidad al mismo.

## RECOMENDACIONES

Actualmente el software lleva el control de medidas de los clientes. Pero no el control de las actividades físicas que realiza el cliente. Por tanto es recomendable adicionar las series, repeticiones y sesiones que realiza el cliente en el gimnasio.

Se recomienda implementar el control de acceso tanto de los clientes, como del personal, a través de un lector de huellas dactilar. Esto para dar mayor agilidad y mayor seguridad en el control de acceso, evitando así el recordar un código, o manejar una tarjeta que puede ser extraviada.

Es importante en el futuro actualizar el software en función de los requerimientos nuevos que puedan surgir.

El sistema ofrece actualmente un modelo de seguridad que restringe al usuario para poder acceder a diferentes opciones. Se recomienda ampliar este modelo de seguridad para hacerlo mas completo, como por ejemplo los permisos de solo lectura sobre algunos formularios.

Para lograr un sistema mas integral de apoyo a los clientes, se recomienda implementar el envío de reportes a correos electrónicos desde la aplicación, tal como las medidas, dietas, recordatorios de inicio de actividades, recordatorios de deudas pendientes.



## BIBLIOGRAFIA

- [ORE98], Funes Orellana, Juan “Contabilidad Intermedia” , Cochabamba, Bolivia, 1998.
- [ORE98], Funes Orellana, Juan “Contabilidad Básica” , Cochabamba, Bolivia, 1998.
- [FRA01], Fraga, Anelise “La Belleza al Alcance de Todos”, Cochabamba, Bolivia, 2001
- [JAC01], Jacobs, Fernando “Soporte Básico de Vida”, Cochabamba, Bolivia, 2001
- [FAN97], “La evaluación de la gestión de un centro deportivo”, 1997
- [FAN97], “La funcionalidad y el coste de los equipamientos deportivos. Manuales de gestión de Centros Deportivos”, 1997.
- [CAS91], “Técnicas de dirección y marketing para entidades deportivas”, 1991
- [ATE03], “Dirección y gestión de centros deportivos”, 2003.
- [LAR99], Larman, Craig, “UML y patrones: Introducción al análisis y diseño orientado a objetos”, Mexico, 1999.
- [PRE02], Roger Pressman, “Ingeniería del software. Un enfoque practico”, 5ta Edición. Madrid – España, 2002.
- [KEN97], Kendall & Kendall “Análisis y Diseño de sistemas”, 4ta edición New Jersey – USA, 1997-
- [JAC99], Boch, Jacobson “UML Gota a Gota”, 2da Edición New Jersey – USA, 1999.
- [BOO99], Booch, Jacobson, Rumbaugh “El lenguaje unificado de modelado”, Editorial Addison Wesley Iberoamericana, Madrid, 1999.
- [JAC00], Jacobson, Ivar; Booch, Grady “Proceso Unificado de Desarrollo de Software, Madrid, España, 2000
- [JIM05], Jim Arlow, Ila Neustadt “Programacion UML 2”, Madrid España, 2005

## OTRAS FUENTES DE INFORMACIÓN

- [www.monografias.com](http://www.monografias.com).  
Concentración en el deportista, Definición de Sauna, Aerobic.
- [www.ccong.org.co/empleoenaccion/glosario.htm](http://www.ccong.org.co/empleoenaccion/glosario.htm)  
Definición de Misión.
- [www.policia.gov.co/inicio/portal/portal.nsf/paginas/GlosarioInstitucional](http://www.policia.gov.co/inicio/portal/portal.nsf/paginas/GlosarioInstitucional)  
Definición de Visión.
- [www.espalda.org/divulgativa/ejercicios/normas/series.asp](http://www.espalda.org/divulgativa/ejercicios/normas/series.asp)  
Normas para realizar ejercicios.
- [www.alemana.cl/bys/afi/afi006.html](http://www.alemana.cl/bys/afi/afi006.html)  
Sobreentrenamiento.
- [www.contabilidad.tk](http://www.contabilidad.tk)  
Valoración de los ingresos.
- [www.elblogsalmon.com](http://www.elblogsalmon.com)  
Definición y explicación del Flujo de Caja.
- <http://www.cs.ualberta.ca/>  
UML
- <http://www.gestiopolis.com>  
Clasificación de las Cuentas por Cobrar
- <http://www.well-connected.com>  
El Ejercicio y las Categorías de Ejercicios.
- <http://www.rae.es>  
Diccionario de la Real Academia Española.

## ENTREVISTAS

- Lic. Daniel Gentili, Gerente Propietario del Gimnasio “Natural Body Club”
- Sr. Wilson Machicado, Administrador del Gimnasio “Natural Body Club”
- Srta. Jessica Cuellar, Secretaria del Gimnasio “Natural Body Club”
- Sr. Cesar García, Entrenador del Gimnasio “Master Gym”
- Lic. Edwin Vargas, Catedrático de la Universidad Católica Boliviana
- Lic. René Ovando, Catedrático de la Universidad Católica Boliviana
- Lic. Jorge Ordoñez, Catedrático de la Universidad Católica Boliviana
- Lic. Wilmer Campos, Decano de la Facultad de Tecnologías, Universidad Domingo Savio.

## **ANEXOS**

## **Anexo A: Registros del Gimnasio**

### **Formulario de Ficha Técnica**

En este formulario el Instructor del Gimnasio registra los datos personales del Cliente, los Antecedentes Médicos, controla los Signos Vitales y las Medidas Antropométricas cada semana.

**Anexo A: Registros del Gimnasio**

<u>LISTAS DE PRECIOS SIMPLES</u>	
APARATOS + SAUNA sin corredora	15 \$
AEROBICS+ GIMNASIA LOCALIZADA	20 \$
SPINNING + APARATO + SAUNA + PISCINA	35 \$
NATACIÓN (3 veces por semana)	25 \$
HIDROGIMNASIA (3 veces por semana)	30 \$
HAPKIDO	20 \$
SPINNING	25 \$
<u>PAQUETES DE PRECIOS SIMPLES PARA GRUPOS</u>	
APARATOS + SAUNA sin corredora	2 pers. por 30 \$
AEROBICS+ GIMNASIA LOCALIZADA	3 pers. Por 45 \$
SPINNING	2 pers. Por 45 \$
TAE-BO	2 pers. Por 45 \$
NATACIÓN	2 pers. Por 45 \$
TAE-KWONDO	2 pers. Por 45 \$
<u>LISTA DE PRECIOS COMPLETOS</u>	
APARATOS + SAUNA sin corredora	20 \$
AEROBICS+SAUNA + PISCINA	25 \$
AEROBICS+APARATOS+ caminadora+sauna+piscina	25 \$
AEROBICS+APARATOS	20 \$
SPINNING + SAUNA + PISCINA	30 \$
SPINNING + aparatos+ caminadora	30 \$
SPINNING+APARATOS+ caminadora+sauna+piscina	35 \$
<u>PAQUETES DE PRECIOS COMPLETO PARA GRUPOS</u>	
APARATOS + SAUNA sin corredora	2 pers. por 35 \$
AEROBICS+SAUNA + PISCINA	2 pers. por 35 \$
AEROBICS+APARATOS+ caminadora+sauna+piscina	2 pers. por 45 \$
AEROBICS+APARATOS	2 pers. por 35 \$
SPINNING + SAUNA + PISCINA	2 pers. por 55 \$
SPINNING + aparatos+ caminadora	2 pers. por 55 \$
SPINNING+APARATOS+ caminadora+sauna+piscina	2 pers. por 60 \$
<u>SECCIONES POR DIA</u>	
APARATOS + SAUNA + piscina	2\$
APARATOS CON CAMINADORA	2\$
SPINNING	2\$
HIDROGIMNASIA	3\$

**Lista de Precios de Disciplinas y Paquetes**

En esta hoja se detallan los Precios por Disciplinas y por Paquetes que ofrece el Gimnasio a sus Clientes.

**Anexo A: Registros del Gimnasio**

	A	B	C	D	E
1	FECHA	DETALLE	INGRESO	EGRESO	CAJA
2	01/07/2005	RUSH			
3		PECHUGA		9	
4		FRUTILLA		4	
5		AGUA MINERAL	2,5		
6		JUGO DE PAPAYA	2,5		
7		2 FILETES	16		
8		PECHUGA		9	
9		LIMON		1	
10		AGUA MINERAL	2,5		
11		ENSALADA DE FRUTAS	4		
12	02/07/2005	MINERAGUA	3,5		
13		RUSH	10		
14		ENSALADA DE FRUTAS	4		
15		JUGO DE GUINEO	2,5		
16		LECHE			
17	04/07/2005	PECHUGA		8	
18		CARNE		3	
19		GUINEO		8	
20		LECHE		5	
21		MANZANA		2	
22		ARROZ		4	
23		JUGO DE DURAZNO	3		
24		LECHE		2,8	
25		PECHUGA (omar)		13,5	
26		JUGO DE GUINEO	2,5		
27	05/07/2005	PECHUGA		25,5	
28		LECHE		5	
29		GUINEO		3	
30		FRUTILLA		4	
31		LECHUGA		1	

**Hoja de Registros utilizando el Microsoft Excel**

Utilizando los libros de Excel se registran los movimientos económicos tanto del Gimnasio como del Snack.

## Anexo B: Implementación de Clases

La clase **clsBaseDatos**, permite establecer y retornar, el nombre de la Base de Datos, la ruta de donde está ubicada la Base de Datos y la copia de seguridad que se genera. Así también se tiene el procedimiento que realiza la copia de seguridad, colocando la fecha y la hora actual del sistema.

```
Public Class clsBaseDatos
    Private strNombre As String
    Private strPathOrigen As String
    Private strPathDestino As String
    Private strNombreBackup As String

    Property Nombre()
        Get
            Return strNombre
        End Get
        Set(ByVal Value)
            strNombre = Value
        End Set
    End Property

    Property PathOrigen()
        Get
            Return strPathOrigen
        End Get
        Set(ByVal Value)
            strPathOrigen = Value
        End Set
    End Property

    Property PathDestino()
        Get
            Return strPathDestino
        End Get
        Set(ByVal Value)
            strPathDestino = Value
        End Set
    End Property

    Property NombreBackup()
        Get
            Return strNombreBackup
        End Get
        Set(ByVal Value)
            strNombreBackup = Value
        End Set
    End Property

    Public Sub CopiaSeguridad()
        Dim file As System.IO.File
        Dim NombreOriginal, NombreCopia, fecha, fechayhora As String

        fecha = Today.Day & "-" & Today.Month & "-" & Today.Year
        fechayhora = fecha & "-" & Now.Hour & "-" & Now.Minute

        strNombreBackup = "BDNaturalBody" & fechayhora & ".mdb"
        Dim objDat As New ClsDatos()
        strNombre = objDat.PathNombreBD
        Dim dirCopia As IO.Directory
```



---

---

***Anexo B***

```
If Not dirCopia.Exists(pstrPathApp & "\CopiaBaseDatos") Then
    dirCopia.CreateDirectory(pstrPathApp & "\CopiaBaseDatos")
End If
strNombreBackup = pstrPathApp & "\CopiaBaseDatos\" & strNombreBackup
file.Copy(strNombre, strNombreBackup, True)
End Sub
```

## Anexo B: Implementación de Clases

La clase **clsCifrarDescifrar**, permite encriptar y desencriptar la contraseña que se define a cada usuario.

```
Public Class clsCifrarDescifrar

    Function TestEncoding(ByVal plainText As String, ByVal password As String) As String
        Dim wrapper As New clsSimple3Des(password)
        Dim cipherText As String
        cipherText = wrapper.EncryptData(plainText)
        Return cipherText
    End Function

    Function TestDecoding(ByVal cipherText As String, ByVal password As String) As String
        Dim wrapper As New clsSimple3Des(password)
        Try
            Dim plainText As String = wrapper.DecryptData(cipherText)
            Return plainText
        Catch ex As System.Security.Cryptography.CryptographicException
            Dim mensaje As String
            mensaje = "El dato no puede ser desencriptado."
            Return mensaje
        End Try
    End Function

End Class
```

La clase **clsSimple3Des** utiliza el espacio de nombre System.Security.Cryptography que proporciona el Visual Basic .Net. Esta clase es utilizada por **clsCifrarDescifrar**

```
Imports System.Security.Cryptography

Public NotInheritable Class clsSimple3Des
    Private TripleDes As New TripleDESCryptoServiceProvider()

    Private Function TruncateHash(ByVal key As String, ByVal length As Integer) As Byte()
        Dim sha1 As New SHA1CryptoServiceProvider()
        Dim KeyBytes() As Byte = System.Text.Encoding.Unicode.GetBytes(key)
        Dim hash() As Byte = sha1.ComputeHash(KeyBytes)
        ReDim Preserve hash(length - 1)
        Return hash
    End Function

    Sub New(ByVal key As String)
        TripleDes.Key = TruncateHash(key, TripleDes.KeySize \ 8)
        TripleDes.IV = TruncateHash(key, TripleDes.BlockSize \ 8)
    End Sub

    Public Function EncryptData(ByVal plaintext As String) As String
        Dim plaintextBytes() As Byte = System.Text.Encoding.Unicode.GetBytes(plaintext)
        Dim ms As New System.IO.MemoryStream()
        Dim encStream As New CryptoStream(ms, TripleDes.CreateEncryptor(),
            System.Security.Cryptography.CryptoStreamMode.Write)
        encStream.Write(plaintextBytes, 0, plaintextBytes.Length)
        encStream.FlushFinalBlock()
        Return Convert.ToBase64String(ms.ToArray)
    End Function
```

```
Public Function DecryptData(ByVal encryptedtext As String) As String
    Dim encryptedBytes() As Byte = Convert.FromBase64String(encryptedtext)
    Dim ms As New System.IO.MemoryStream()
    Dim decStream As New CryptoStream(ms, TripleDes.CreateDecryptor(),
        System.Security.Cryptography.CryptoStreamMode.Write)
    decStream.Write(encryptedBytes, 0, encryptedBytes.Length)
    decStream.FlushFinalBlock()
    Return System.Text.Encoding.Unicode.GetString(ms.ToArray)
End Function

End Class
```

## Anexo B: Implementación de Clases

El formulario que muestra los reportes, presenta el procedimiento **CargarReporte** que permite cargar un reporte según el reporte especificado.

```
Public Sub cargarreporte(ByVal enuMostrarReporte As Reporte)
    Me.WindowState = FormWindowState.Maximized
    Dim CurrentDir As String = Directory.GetCurrentDirectory
    crwReporte.Width = Me.Width
    crwReporte.Height = Me.Height
    crwReporte.DisplayGroupTree = False
    crwReporte.ShowGroupTreeButton = False

    enuMostrarRep = enuMostrarReporte

    Select Case enuMostrarRep
        Case enuMostrarRep.precios
            Dim paramFields As New CrystalDecisions.Shared.ParameterFields()
            Dim paramField As New CrystalDecisions.Shared.ParameterField()
            Dim discretVal As New CrystalDecisions.Shared.ParameterDiscreteValue()
            paramField.ParameterFieldName = "paramEmpresa"
            discretVal.Value = pstrEmpresa
            paramField.CurrentValues.Add(discretVal)
            paramFields.Add(paramField)

            Dim paramField1 As New CrystalDecisions.Shared.ParameterField()
            Dim discretVal1 As New CrystalDecisions.Shared.ParameterDiscreteValue()
            paramField1.ParameterFieldName = "paramDireccion"
            discretVal1.Value = pstrDireccion
            paramField1.CurrentValues.Add(discretVal1)
            paramFields.Add(paramField1)

            Dim paramField2 As New CrystalDecisions.Shared.ParameterField()
            Dim discretVal2 As New CrystalDecisions.Shared.ParameterDiscreteValue()
            paramField2.ParameterFieldName = "paramTelefono"
            discretVal2.Value = pstrTel
            paramField2.CurrentValues.Add(discretVal2)
            paramFields.Add(paramField2)

            crwReporte.ParameterFieldInfo = paramFields
            Dim rptLP As New rptListaPrecios()

            crwReporte.ReportSource = rptLP

        Case enuMostrarRep.empleados
            Try
                Dim paramFields As New CrystalDecisions.Shared.ParameterFields()
                Dim paramField As New CrystalDecisions.Shared.ParameterField()
                Dim discretVal As New CrystalDecisions.Shared.ParameterDiscreteValue()
                paramField.ParameterFieldName = "paramEmpresa"
                discretVal.Value = pstrEmpresa
                paramField.CurrentValues.Add(discretVal)
                paramFields.Add(paramField)

                Dim paramField1 As New CrystalDecisions.Shared.ParameterField()
                Dim discretVal1 As New CrystalDecisions.Shared.ParameterDiscreteValue()
                paramField1.ParameterFieldName = "paramDireccion"
                discretVal1.Value = pstrDireccion
                paramField1.CurrentValues.Add(discretVal1)
                paramFields.Add(paramField1)

                Dim paramField2 As New CrystalDecisions.Shared.ParameterField()
                Dim discretVal2 As New CrystalDecisions.Shared.ParameterDiscreteValue()
```

```

paramField2.ParameterFieldName = "paramTelefono"
discretVal2.Value = pstrTel
paramField2.CurrentValues.Add(discretVal2)
paramFields.Add(paramField2)

Dim paramField3 As New CrystalDecisions.Shared.ParameterField()
Dim discretVal3 As New CrystalDecisions.Shared.ParameterDiscreteValue()

paramField3.ParameterFieldName = "paramNombre"
discretVal3.Value = pstrNombre
paramField3.CurrentValues.Add(discretVal3)
paramFields.Add(paramField3)

crwReporte.ParameterFieldInfo = paramFields
Dim StrCommCate As String = "SELECT
tblEmpleado.strNombre,tblEmpleado.strApellidoP,
tblEmpleado.strApellidoM,tblEmpleado.strTelefonoDom,tblEmpleado.
strTelefonoCel,tblEmpleado.
intSueldoMes FROM tblEmpleado " & strCondicion
Dim oleConn As New ClsDatos()
oleConn.ConectarDBOLE()
Dim dspc As New dsEmpleado()

Dim sqlDaCate As New OleDb.OleDbDataAdapter(StrCommCate,
oleConn.ConexionString)
sqlDaCate.Fill(dspc, "tblEmpleado")
Dim info As New rptEmpleados()
info.SetDataSource(dspc)
crwReporte.ReportSource = info
Catch ex As Exception
MsgBox(ex.Message)
End Try

Case enuMostrarRep.ClientesIns

Dim paramFields As New CrystalDecisions.Shared.ParameterFields()
Dim paramField As New CrystalDecisions.Shared.ParameterField()
Dim discretVal As New CrystalDecisions.Shared.ParameterDiscreteValue()
paramField.ParameterFieldName = "paramFechaIni"
discretVal.Value = dtmParam1
paramField.CurrentValues.Add(discretVal)
paramFields.Add(paramField)

Dim paramField1 As New CrystalDecisions.Shared.ParameterField()
Dim discretVal1 As New CrystalDecisions.Shared.ParameterDiscreteValue()
paramField1.ParameterFieldName = "paramFechaFin"
discretVal1.Value = dtmParam2
paramField1.CurrentValues.Add(discretVal1)
paramFields.Add(paramField1)

crwReporte.ParameterFieldInfo = paramFields

Dim StrCommCate As String = "SELECT tblCliente.strNombre,
tblCliente.strApellidoP,tblCliente.strApellidoM, tblCliente.intEdad,
tblCliente.strTelCel FROM tblCliente WHERE " & strCondicion

Dim oleConn As New ClsDatos()
oleConn.ConectarDBOLE()
Dim dspc As New dsClientesIns()

Dim sqlDaCate As New OleDb.OleDbDataAdapter(StrCommCate,
oleConn.ConexionString)
sqlDaCate.Fill(dspc, "tblCliente")
Dim info As New rptClientesIns()
info.SetDataSource(dspc)
crwReporte.ReportSource = info

End Select

```

Forma de llamar al procedimiento **CargarReporte**:

```
enuMostrarRep = Reporte.precios  
Dim frmreporte As New frmReporte()  
clsControl.CargarForm(frmreporte, False, False, Me)  
frmreporte.cargarreporte(Reporte.precios)
```