

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE TECNOLOGÍA

SISTEMA UNIFICADO DE PROVEEDORES SUP

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN
ANÁLISIS DE SISTEMAS INFORMÁTICOS

FELIPE FERNANDO BARRERA ARELLANO
mailto:felfer_ba@hotmail.com

DIRECTOR: ING. DANIEL MANANGÓN
<mailto:daniel.manangon@epn.edu.ec>

Quito, Junio 2008

DECLARACIÓN

Yo, Felipe Fernando Barrera Arellano, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Felipe Fernando Barrera Arellano

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Felipe Fernando Barrera Arellano, bajo mi supervisión.

Ing. Daniel Manangón

DIRECTOR DE PROYECTO

INTRODUCCIÓN	I
RESUMEN.....	II
CAPÍTULO I.....	1
ASPECTOS GENERALES.....	1
1.1 ÁMBITO	1
1.2 PLANTEAMIENTO DEL PROBLEMA	2
1.3 FORMULACIÓN Y SISTEMATIZACIÓN DEL PROBLEMA.....	3
1.4 OBJETIVOS DE LA INVESTIGACIÓN	4
<i>OBJETIVO GENERAL.....</i>	<i>4</i>
<i>OBJETIVOS ESPECÍFICOS</i>	<i>4</i>
1.5 ALCANCE	4
1.6 JUSTIFICACIÓN DEL PROYECTO	5
1.7 ASPECTOS METODOLÓGICOS.....	6
CAPÍTULO II	8
MARCO TEÓRICO	8
2.1 INGENIERÍA DE SOFTWARE	8
2.1.1 <i>CICLO DE VIDA DE DESARROLLO DE SOFTWARE.....</i>	<i>8</i>
2.1.2 <i>MODELOS DE CICLOS DE VIDA DE DESARROLLO DE SOFTWARE..</i>	<i>9</i>
2.1.2.1 Modelo Espiral.....	10
2.1.2.1.1 Etapas del Modelo Espiral	11
2.2 METODOLOGÍA DE DESARROLLO DE SOFTWARE	12
2.2.1 <i>METODOLOGÍA OMT.....</i>	<i>13</i>
2.2.1.1 Fases de la Metodología OMT.....	13
2.2.1.1.1 Análisis	13
2.2.1.1.2 Diseño	15
2.2.1.1.3 Implementación.....	16
2.2.1.1.1 Modelo de Objetos.....	17
2.2.1.1.2 Modelo Dinámico	18
2.2.1.1.3 Modelo Funcional	18
2.3 LENGUAJE UNIFICADO DE MODELADO UML.....	19
2.3.1 <i>OBJETIVOS DEL UML</i>	<i>19</i>
2.3.2 <i>DIAGRAMAS EN UML.....</i>	<i>19</i>
2.3.2.1 Diagrama de Casos de Uso	20
2.3.2.2 Diagrama de Actividades	23
2.3.2.3 Diagrama de Clases.....	27
2.3.2.4 Diagrama de Colaboración	31
2.3.2.5 Diagrama de Componentes	32
2.3.2.6 Diagrama de Estados.....	35
2.3.2.7 Diagrama de Secuencia.....	36
2.4 HERRAMIENTAS CASE (COMPUTER AIDED SOFTWARE ENGINEERING, INGENIERÍA ASISTIDA POR COMPUTADORA)	37
2.4.1 <i>Clasificación de herramientas CASE.....</i>	<i>38</i>
2.4.2 <i>Clasificación de las herramientas CASE Cliente/Servidor</i>	<i>46</i>
2.4.3 <i>RATIONAL ROSE 2000</i>	<i>49</i>
2.4.4 <i>SQL SERVER 2000</i>	<i>50</i>
2.4.5 <i>VISUAL BASIC</i>	<i>51</i>
2.4.6 <i>CLIENTE SERVIDOR.....</i>	<i>54</i>

CAPÍTULO III.....	57
SISTEMA UNIFICADO DE PROVEEDORES SUP.....	57
3.1 INVESTIGACIÓN PRELIMINAR	57
3.1.1 <i>ESPECIFICACIÓN DE REQUERIMIENTOS</i>	57
3.1.1.1 Descripción de Actores	57
3.1.1.1.1 Administrador	57
3.1.1.1.2 Usuario	57
3.1.1.2 Diagrama de Casos de Uso	57
3.1.1.2.1 Actividad Actor Administrador	57
3.1.1.2.2 Actividad Actor Usuario	59
3.1.1.3 Diccionario de Casos de Uso	61
3.1.1.3 .1 Actor Administrador	61
3.1.1.3 .2 Actor Usuario.....	61
3.2 ANALISIS Y DISEÑO	62
3.2.1 <i>MODELO ESTÁTICO</i>	62
3.2.1.1 Diagrama de Clases.....	62
3.2.1.2 Diccionario de Clases	63
3.2.1.3 Diagrama de Objetos.....	74
3.2.2 <i>MODELO DINÁMICO</i>	75
3.2.2.1 Diagramas de Interacción	75
3.2.3 <i>MODELO FUNCIONAL</i>	78
3.2.3.1 Diagrama de Actividades	78
3.2.3.2 Diagrama de Estados.....	79
3.3 CONSTRUCCIÓN.....	79
3.3.1 <i>PRINCIPALES ALGORITMOS</i>	79
3.3.1.1 Guardar	79
3.3.1.2 Modificar.....	80
3.3.1.3 Buscar	80
3.3.1.4 Mostrar	80
3.3.1.5 Generar Archivo.....	80
3.3.1.6 Exportar Datos	81
3.4 PRUEBAS.....	81
3.4.1 <i>PRUEBA FUNCIONAL</i>	81
CONCLUSIONES.....	90
RECOMENDACIONES.....	91
GLOSARIO DE TÉRMINOS.....	92
BIBLIOGRAFÍA.....	94
ANEXO	95

ÍNDICE DE FIGURAS

Figura 2.1	Modelo Espiral.....	10
Figura 2.2	Grafo Caso de Uso.....	20
Figura 2.3	Grafo Actor.....	21
Figura 2.4	Relación de Asociación de Caso de Uso Actor.....	21
Figura 2.5	Relación de Generalización entre casos de uso.....	22
Figura 2.6	Relación Incluye entre casos de uso.....	22
Figura 2.7	Relación Extend entre casos de uso.....	23
Figura 2.8	Grafo Actividad.....	23
Figura 2.9	Transición entre actividades.....	24
Figura 2.10	Condición entre flujo de actividades.....	24
Figura 2.11	Merge entre flujo de actividades.....	24
Figura 2.12	Barra de sincronización entre flujo de actividades.....	25
Figura 2.13	Grafo Inicio y Fin de flujo de actividades.....	25
Figura 2.14	Grafo Carril de separación de actividades.....	26
Figura 2.15	Grafo Representación Objeto.....	26
Figura 2.16	Representación Actividad y Objeto.....	27
Figura 2.17	Grafo Representación de una Clase.....	27
Figura 2.18	Relación de dependencia entre clases.....	30
Figura 2.19	Relación de generalización entre clases.....	30
Figura 2.20	Ejemplo de Diagrama de Colaboración.....	31
Figura 2.21	Grafo Representación de componente.....	32
Figura 2.22	Representación de una interfaz.....	33
Figura 2.23	Relación de realización.....	34
Figura 2.24	Relación de dependencia entre componentes.....	34
Figura 2.25	Representación de interfaz en forma extendida.....	34
Figura 2.26	Representación de interfaz de exportación.....	35
Figura 2.27	Ejemplo de Diagrama de Objetos.....	35
Figura 2.28	Ejemplo de Diagrama de Secuencia.....	36
Figura 2.29	Ejemplo de creación y final de líneas de vida.....	37
Figura 2.30	Modelo Cliente Servidor.....	55
Figura 3.1	Ingreso de Nuevo Funcionario.....	82
Figura 3.2	Mensaje de ingreso exitoso.....	83
Figura 3.3	Formulario de ingreso información bancaria.....	83
Figura 3.4	Asignación de Operación 1.....	84
Figura 3.5	Asignación Operación 2.....	84
Figura 3.6	Asignación Operación 3.....	84
Figura 3.7	Acreditación monto por quincena.....	85
Figura 3.8	Consulta de pago de quincena.....	86
Figura 3.9	Reporte de pago de quincena.....	86
Figura 3.10	Mensaje de confirmación de registros y monto para transferencia.....	87
Figura 3.11	Reporte Edad de Funcionarios Activos.....	87
Figura 3.12	Reporte de Tiempo de Servicio.....	88
Figura 3.13	Consulta de Proyectos por Especialista Sectorial.....	88
Figura 3.14	Consulta de Proyectos por Especialista Alterno.....	89

INTRODUCCIÓN

Los avances de la tecnología y la reducción de pasos automatizando un proceso hacen que los sistemas tengan gran acogida en las empresas que necesitan reducir tiempo y recursos en sus actividades para cumplir un objetivo determinado.

Con la modernización de las empresas en hardware es necesaria la implementación de software que satisfaga uno o varios requerimientos de los usuarios que ven en los sistemas una forma de incrementar sus actividades siendo más eficientes en sus puestos de trabajo.

La diversidad de herramientas de construcción de software facilita una propuesta factible para cubrir los requerimientos de un usuario automatizando un proceso, en el mercado actualmente existen herramientas de construcción que se ajustan a presupuestos dependiendo del alcance que se pretende cubrir con la construcción de un sistema.

Las metodologías aplicadas en el desarrollo de software son una guía que se puede moldear para ajustar un proyecto a las necesidades que se necesitan cubrir, las herramientas de modelamiento de sistemas permiten tener una visión amigable de las fases que se siguen para obtener un resultado final, los lenguajes de programación de cuarta generación son poderosas fuentes de construcción de algoritmos que facilitan la interacción entre objetos. Todas las facilidades que se tienen en la actualidad para satisfacer las necesidades de una empresa automatizando procesos van de la mano de personas capaces de interpretar las reglas de negocio de una empresa y en base a sus conocimientos proponer soluciones informáticas que mejoren los procesos internos de una empresa reduciendo recursos económicos y facilitando el trabajo de los recursos humanos.

RESUMEN

Todos los procesos realizados por el área administrativa de una corporación crecen a medida que la corporación va aumentando sus recursos humanos y clientes.

La función de los recursos humanos es responsable de atraer, desarrollar y mantener la fuerza de trabajo de la empresa. Los sistemas de información de los recursos humanos apoyan actividades como identificar empleados potenciales, llevar registros completos de empleados existentes y crear programas para desarrollar los talentos y habilidades de los empleados.

Este es el caso de la Representación de Ecuador del Banco Interamericano de Desarrollo, mismo que requiere un sistema de información que sea capaz de automatizar los procesos que hace el área administrativa para gestionar empleados, proveedores y proyectos que es la actividad que fundamenta la presencia del BID en la República del Ecuador.

De esta manera nace SUP Sistema Unificado de Proveedores, este sistema cubre los requerimientos del BID para manejo de empleados información personal, educativa, contractual y familiar, crea procesos simples de creación de archivos planos para transferencias interbancarias en convenio con el Banco Internacional. De la misma manera crea archivos para transferencias a proveedores. Y administra proyectos desde la perspectiva de asignaciones de operaciones en ejecución.

SUP esta formado por un grupo de interfaces desarrolladas en Visual Basic 6.0 que administran la información contenida en una base de datos relacional construida en SQL 2000.

CAPÍTULO I

ASPECTOS GENERALES

1.1 ÁMBITO

El Banco Interamericano de Desarrollo BID, Representación Ecuador se encuentra ubicado en la ciudad de Quito, provincia de Pichincha en la Av. 12 de Octubre N24-528 y Luis Cordero, Edificio World Trade Center Torre B, Piso 9.

El BID es una iniciativa largamente esperada por los países latinoamericanos, se creó en 1959 como una innovadora institución financiera multilateral para el desarrollo económico y social de América Latina y el Caribe. Sus programas de préstamos y de cooperación técnica van más allá del mero financiamiento, por cuanto apoyan estrategias y políticas para reducir la pobreza, impulsar el crecimiento sostenible, expandir el comercio, la inversión y la integración regional, promover el desarrollo del sector privado y modernizar el Estado.

Los programas e instrumentos del BID fueron modelos para la creación de otras instituciones multilaterales de desarrollo a nivel regional y subregional. El BID constituye actualmente la principal fuente de financiamiento multilateral para proyectos de desarrollo económico, social e institucional en los países de América Latina y el Caribe.

El Grupo BID está integrado por el Banco Interamericano de Desarrollo, la Corporación Interamericana de Inversiones (CII) y el Fondo Multilateral de Inversiones (Fomin). La CII enfoca su acción en el apoyo a la pequeña y mediana empresa, mientras que el Fomin promueve el desarrollo del sector privado a través de donaciones e inversiones.

Los países miembros originales del BID son Argentina, Bolivia, Brasil, Chile, Colombia, Costa Rica, Ecuador, El Salvador, Guatemala, Haití, Honduras, México, Nicaragua, Panamá, Paraguay, Perú, República Dominicana, Uruguay, Venezuela y Estados Unidos.

El número de países miembros aumentó con el ingreso de Trinidad y Tobago en 1967, país al que pronto se unieron Barbados y Jamaica en 1969, Canadá en 1972, Guyana en 1976, Bahamas en 1977 y Suriname en 1980. Entre 1976 y

1986, ingresaron al Banco los 19 países miembros no regionales, que incluyen a 16 países europeos, Israel y Japón. Belice ingresó en 1992 y, tras la desintegración de Yugoslavia, en 1993 se unieron Croacia y Eslovenia. La República de Corea ingresó como país miembro en 2005.

El Banco actualmente tiene 47 países miembros, de los cuales 26 son miembros prestatarios en la región. El poder de votación de cada país está basado en las suscripciones del mismo al capital ordinario (CO) de la institución.

1.2 PLANTEAMIENTO DEL PROBLEMA

El BID al ser un institución multilateral, cuenta con sistemas especializados desarrollados exclusivamente para su misión de desarrollo y sostenibilidad de sus países miembro, los sistemas de desembolsos y seguimiento para ejecución de préstamos son centralizados en su SEDE, con lugar en Washington D.C. Estados Unidos de América, estos sistemas son globales y obligatorios para todos sus países miembros.

La Representación del BID en Ecuador cuenta con los sistemas globales del Banco, pero en la actualidad la Representación requiere un sistema interno que le permita tener información local de su personal, proveedores, operaciones en ejecución y toda la información relacionada a estas directrices, tales como especialista a cargo de las múltiples operaciones del BID en Ecuador, información de trayectoria profesional, contratos, estado civil de los funcionarios del BID, cargas familiares, empresas que prestan servicios al Banco, identificadas como proveedores del mismo.

Esta información está disponible en aplicaciones ofimáticas archivos word, excel access, en los ordenadores de los funcionarios administrativos que mensualmente tienen que utilizarla para realizar los pagos al personal local del BID y de sus proveedores.

Este proceso es lento y poco confiable, ya que para realizar los pagos se debe realizar una carta al Gerente del Banco Internacional, solicitándole autorizar la acreditación de los montos correspondientes a los salarios a los diferentes Bancos beneficiarios del Staff del BID.

Tomando en cuenta la delicada situación, se plantea desarrollar un sistema informático que apoye en todas las actividades de manejo de información

confidencial del personal del BID, operaciones en ejecución, especialista y ejecutor a cargo y proveedores, proporcionando seguridad en los datos centralizados en un servidor local con accesos limitados exclusivos para personal del área administrativa y la gerencia del BID.

1.3 FORMULACIÓN Y SISTEMATIZACIÓN DEL PROBLEMA

¿Cómo obtener información del personal, proyectos y proveedores del BID?

¿Cómo debería el BID realizar los pagos al personal y proveedores?

Interrogantes que el sector Administrativo del BID tiene, que serán resueltas con el desarrollo de una Base de Datos que almacene información de recursos humanos, proveedores y proyectos en ejecución del BID en Ecuador, misma que debe estar relacionada para poder generar reportes que reflejen información de los funcionarios del BID tal como Educación, Estado Civil, Cargas Familiares, Proyectos a Cargo en Ejecución, Dependencia a la cual pertenece dentro del BID; también información de las empresas que prestan servicios al Banco, denominados Proveedores, como Dirección, Teléfono, Contacto, Servicios que prestan entre otros.

Para poder ingresar modificar y actualizar la información en dicha base de datos se debe desarrollar una interface que permita al usuario tener una forma amigable de mantener la información requerida.

La manera más eficiente y segura que el BID tendría para realizar sus pagos, es realizando transferencias interbancarias en línea, los recursos del BID destinados para pago de Nomina de su Personal Local y de sus Proveedores, los tiene en el Banco Internacional, mismo que posee servicios de transferencias interbancarias en línea a través de su Portal Web; la propuesta es obtener un archivo plano en formato establecido por el Banco Internacional, que contenga información tal como Banco Beneficiario, Beneficiario, Tipo de Cuenta, Monto a Acreditar, Fecha de la Transferencia, etc., información que estará disponible en la Base de Datos propuesta y que será alimentada por la persona encargada del Pago de Nómina quincenalmente a través de una Interface.

1.4 OBJETIVOS DE LA INVESTIGACIÓN

OBJETIVO GENERAL

- ✓ Apoyar en todas las actividades de manejo de información confidencial del personal del BID, operaciones en ejecución, especialista y ejecutor a cargo y proveedores, proporcionando seguridad en los datos centralizados en un servidor local con accesos limitados exclusivos para personal del área administrativa y la gerencia del BID.

OBJETIVOS ESPECÍFICOS

- ✓ Desarrollar una Base de Datos relacionada y una interface para mantenerla.
- ✓ Generar informes a traves de datos que se desplegarán en el front end de la Aplicación.
- ✓ Controlar y verificar errores antes de enviar la información
- ✓ Validar el acceso de personas a las aplicaciones que generarán la información privada de ejecutores y presupuestos

1.5 ALCANCE

SUP será desarrollado con el lenguaje de programación de cuarta generación Microsoft Visual Studio 6.0, Paquete Microsoft Visual Basic 6.0 en lo que se refiere a interface para acceder a la información de la Base de Datos Relacionada, misma que estará desarrollada en Microsoft SQL 2000. La Base de Datos SUP residirá en un servidor del BID, SUP será un sistema orientado a objetos multiusuario. Las potencialidades de SUP serán:

- **Módulo de Administración**
 - Conexión Base de Datos
 - Crear Usuarios
 - Crear Perfiles (accesos a módulos)
 - Desconectar Base de Datos
- **Módulo Funcionarios**
 - Ingresar Nuevos Funcionarios

- Editar Funcionarios
- Ingresar Educación de Funcionarios
- Ingresar Familia de Funcionarios
- **Módulo de Ingreso Proveedor**
 - Crear Proveedores
 - Modificar Proveedores
- **Módulo de Ingreso Proyectos en Ejecución**
 - Crear Proyectos
 - Modificar Proyectos
 - Asignar Proyectos
- **Módulo Ingreso Pago Nómina**
 - Ingresar monto a acreditar quincenalmente a los Beneficiarios.
 - Cargar Beneficiarios para transferencias interbancarias de la base.
 - Generar un archivo plano .prn de acuerdo a formato establecido por el Banco Internacional para transferencias interbancarias.
- **Módulo Ingreso Pago Proveedores**
 - Ingresar monto a acreditar a los Proveedores.
 - Generar un archivo plano .prn de acuerdo a formato establecido por el Banco Internacional para transferencias interbancarias.
- **Módulo Consultas**
 - Realizar consultas de los registros de pagos nómina y proveedores.
 - Hacer consultas de Funcionarios ingresados.
 - Calcular edad de Funcionarios e hijos.
 - Calcular Tiempo de Servicio
- **Módulo Mantenimiento**
 - Actualizar Tablas relacionadas a Funcionarios

1.6 JUSTIFICACIÓN DEL PROYECTO

La función de los recursos humanos es responsable de atraer, desarrollar y mantener la fuerza de trabajo de la empresa. Los sistemas de información de los recursos humanos apoyan actividades como identificar empleados potenciales,

llevar registros completos de empleados existentes y crear programas para desarrollar los talentos y habilidades de los empleados.

Los sistemas de recursos humanos, al nivel estratégico, identifican los requerimientos de mano de obra (habilidades, nivel educativo, tipos de puesto, número de puestos y costo)

El Sistema Unificado de Proveedores SUP será desarrollado con fines administrativos, ya que el Área Administrativa del Banco Interamericano de Desarrollo, Representación Ecuador, no dispone de un Sistema que le proporcione información de su Recurso Humano, de los Proyectos a cargo de los Especialistas, información personal, estudios, cargas familiares, pagos realizados, proveedores.

El BID posee Sistemas de Administración de Proyectos pero no dispone de un Sistema con fines administrativos local, con SUP se espera tener un ahorro en tiempo para la generación de informes que ayuden en la toma de decisiones referente a la contratación de recursos externos con la información de proveedores. También se pretende poder generar reportes de Pagos quincenales y poder generar un archivo plano con formato requerido por el Banco Internacional para realizar transferencias interbancarias a través de su Portal Web.

1.7 ASPECTOS METODOLÓGICOS

Un proyecto informático sigue fases que se ajustan a requerimientos de un usuario o empresa, a estas fases se las resume en una metodología que pretende organizar el trabajo proporcionando una guía para construir software de calidad.

Junto con la metodología de trabajo se van moldeando al proyecto modelos a seguir que facilitan la determinación de fases que se ajustan a las necesidades que se van a cubrir con el proyecto en construcción. Estas fases van creando documentación similar a un plano que se usa en las construcciones civiles, que en informática se conocen como diagramas, estos siguen un estandar del lenguaje unificado de modelamiento, con toda la documentación se van armando los algoritmos que automatizan un proceso dando como resultado un sistema. En la Tabla 1.1 se resume la metodología, modelo y diagramas separados en fases de construcción de SUP.

PARADIGMA ESPIRAL	METODOLOGIA OMT	DIAGRAMAS UML	
ESPECIFICACION DE REQUERIMIENTOS		Descripción de Actores	
		Diagrama de Casos de Uso	
		Diccionario de Casos de Uso	
ANÁLISIS Y DISEÑO	MODELO ESTÁTICO	Diagrama de Clases	
		Diccionario de Clases	
		Diagrama de Objetos	
	MODELO DINÁMICO	Diagramas de Interacción	Diagrama de Secuencia
			Diagrama de Colaboración
	MODELO FUNCIONAL	Diagrama de Actividades	
Diagrama de Estados			
CONSTRUCCIÓN	Rational Rose 2000 SQL Server 2000 Visual Basic 6.0 Cliente Servidor		
PRUEBAS	Prueba Funcional		
MANTENIMIENTO	A posterior, basados en la documentación técnica se podrá dar un correcto mantenimiento.		

Tabla 1.1

Fuente: Apuntes de Clase, Materia Desarrollo de Sistemas de Información, Profesora Ing. Miriam Peñafiel.

CAPÍTULO II

MARCO TEÓRICO

2.1 INGENIERÍA DE SOFTWARE

Según PRESSMAN¹, la Ingeniería de Software es el establecimiento y uso de los principios idóneos de la ingeniería con el fin de obtener, económicamente, software que sea confiable y que trabaje eficientemente en máquinas reales (computadores).

Según THAYER², es la ciencia de la ingeniería que aplica los conceptos de análisis, diseño, codificación, pruebas, documentación y administración para obtener con éxito grandes programas de computador contruidos para los usuarios.

Según FAIRLEY³, la Ingeniería de Software se define como la disciplina tecnológica preocupada de la producción sistemática y del mantenimiento de los productos de software que son desarrollados y modificados a tiempo y dentro de un presupuesto definido

2.1.1 CICLO DE VIDA DE DESARROLLO DE SOFTWARE

El Ciclo de Vida de Desarrollo de Software es el conjunto de etapas que se sigue para construir, entregar y hacer evolucionar el software, desde la concepción de una idea hasta la entrega y retiro del sistema.

Senn (1992)⁴, establece el CVDS en seis fases

1. Investigación preliminar.
2. Determinación de los requerimientos del sistema.
3. Diseño del sistema.
4. Desarrollo del software.
5. Prueba de los sistemas.
6. Implantación y evaluación.

¹ Presuman, Ingeniería de Software. Un enfoque práctico. Editorial McGraw-Hill

² Richard Thayer, Requerimientos de Ingeniería de Software, IEEE, 1992

³ R. Fairley, Ingeniería de Software, Editorial MacGraw-Hill, 1987

⁴ Senn, Análisis y Diseño de Sistemas de Información, 2da Edición

Kendall (1997)⁵, divide el CVDS en siete fases que son las siguientes:

1. Identificación de problemas, oportunidades y objetivos.
2. Determinación de los requerimientos de información.
3. Análisis de las necesidades del sistema.
4. Diseño del sistema recomendado.
5. Desarrollo y documentación del software.
6. Prueba y mantenimiento del sistema.
7. Implementación y evaluación del hardware.

Según BOOCH⁶, lo que da inicio al CVDS son macroprocesos dentro de los cuales existe un miniproceso. Las fases del macroproceso son:

1. Conceptualización
2. Análisis
3. Diseño
4. Evolución
5. Mantenimiento

2.1.2 MODELOS DE CICLOS DE VIDA DE DESARROLLO DE SOFTWARE

Un modelo de CDVS se define como la visión de todas las actividades que ocurren durante el desarrollo de software, intenta determinar el orden de las etapas involucradas y los criterios de transición asociadas entre estas etapas.

El Modelo de Ciclo de Vida de Desarrollo de Software tiene como objetivos.

- ✓ Describir las fases principales de desarrollo de software.
- ✓ Definir las actividades primarias a ejecutarse durante las fases.
- ✓ Proveer un espacio de trabajo para la definición de un detallado proceso de desarrollo de software.

Entre los principales Modelos de Ciclos de Vida de Desarrollo de Software están:

- ✓ Cascada
- ✓ Incremental
- ✓ Desarrollo Evolutivo

⁵ Kendall&Kendall, Kenneth y Julie, Análisis y Diseño de Sistemas, 3era. Edición, Editorial Prentice Hall, 1997, Mexico.

⁶ Booch, B. "Object Oriented Development Trans. Of Soft. Eng." Vol SE-12, Num 2 Febrero 1986

✓ Espiral

Para desarrollar el Sistema Unificado de Proveedores SUP el Modelo de Ciclo de Vida de Desarrollo de Software que se ajusta a las necesidades y características del mismo, es el Modelo Espiral.

2.1.2.1 Modelo Espiral

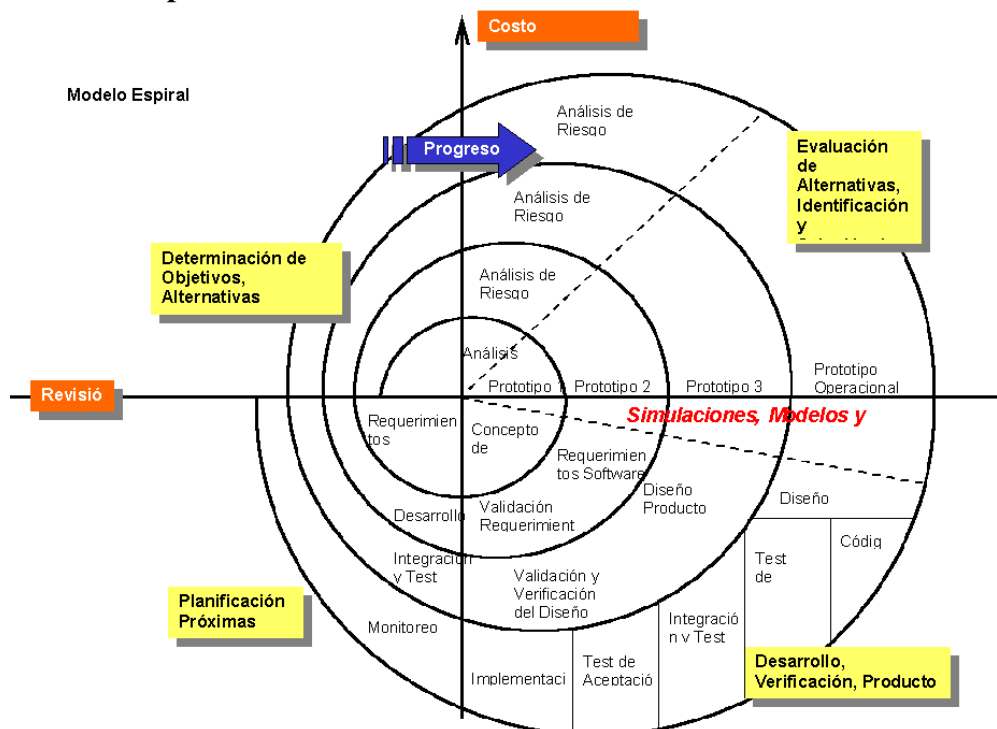


Figura 2.1 Modelo Espiral

Fuente: Modelo Espiral para Desarrollo y Mejoramiento de Software, vol 21, 1988, Boehm, B.

El Modelo en Espiral, propuesto por BOEHM⁷, es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial, para asegurar la desaparición de incertidumbres y/o ignorancias. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. En el modelo espiral, el software se desarrolla en una serie de versiones incrementales.

El Modelo en Espiral está orientado a riesgos, que divide un proyecto de software en miniproyectos. Cada miniproyecto se centra en uno o más riesgos importantes

⁷ Boehm, B. "A Spiral model for Software Development and Enhancement" vol 21, Mayo 1988

hasta que todos estos estén controlados. En cada ciclo o espiral las especificaciones del producto se van resolviendo.

2.1.2.1.1 Etapas del Modelo Espiral

- **Planificación**

Es la determinación de objetivos, límites y condiciones de contorno (alcance del sistema), condiciones que limitan de alguna manera el desarrollo, economía, tiempo, etc. del sistema.

Las actividades de planificación son:

Predecir la duración de las actividades y tareas de nivel individual. Recursos requeridos, concurrencia y solapamiento de tareas para el desarrollo en paralelo y camino crítico a través de la red de actividades.

Estimar recursos.

Predicción de personal, esfuerzo y costo que se requerirán para terminar las actividades y productos conocidos asociados con el proyecto.

Planificar tareas y solapamiento de tareas y actividades.

Definir y desarrollar los requerimientos de software.

Definir los requisitos de la interfase.

Priorizar e integrar los requisitos de software.

- **Análisis de Riesgos**

El análisis de riesgo es el desarrollo de un plan para descubrir los riesgos más importantes y resolver los mismos (mitigar riesgos).

Eliminación de los aspectos no compatibles con las condiciones de contorno o límites.

Las actividades de Análisis de Riesgo son:

Identificar ideas o necesidades.

Formular soluciones potenciales.

Conducir estudio de viabilidad (factibilidad técnica, económica, operativa y legal).

Planificar la transición del sistema.

Refinar y finalizar la idea o necesidad.

Analizar las funciones del sistema.

Desarrollar la arquitectura del sistema.

Descomponer los requisitos del sistema.

Planificación de contingencias (situación que puede hacer que algo fracase).

- **Ingeniería**

La ingeniería es el desarrollo del producto o prototipo según las condiciones de la etapa anterior.

Las actividades de la ingeniería son:

Realizar el desarrollo arquitectónico.

Analizar el flujo de la información.

Diseñar la base de datos.

Seleccionar o desarrollar algoritmos.

Realizar el diseño detallado de la etapa.

Crear el código fuente.

- **Evaluación**

La evaluación es la etapa en la cual se evalúa los resultados del prototipo obtenido se verifica y se valida.

Las actividades de la evaluación son:

Crear los datos de prueba de código fuente.

Ejecutar las tareas de de verificación y validación.

Recoger y analizar los datos de la métrica.

Planificar las pruebas.

Desarrollar las especificaciones de las pruebas.

Ejecutar las pruebas.

Generar los aspectos de mejora, errores, defectos y ampliaciones.

2.2 METODOLOGÍA DE DESARROLLO DE SOFTWARE

La Metodología de Desarrollo de Software se define como el conjunto de métodos que se aplican a lo largo del ciclo de vida de desarrollo de software, unificados o acoplados bajo un enfoque filosófico.

Es una versión amplia y detallada de un ciclo de vida completo de desarrollo de software que incluye,

- ✓ Reglas, procedimientos, métodos, herramientas.
- ✓ Funciones individuales y en grupo de cada tarea.
- ✓ Productos resultantes
- ✓ Normas de Calidad

2.2.1 METODOLOGÍA OMT

La Metodología OMT (Object Modeling Technique) fue desarrollada por James Rumbaugh y Michael Blaha en 1991, mientras James Rumbaugh dirigía un equipo de investigación de los laboratorios General Electric⁸.

La Metodología OMT es no propietaria, lo que le permite tener versatilidad para acoplarse a todas las necesidades de desarrollo de software orientado a objetos, basada en el análisis y diseño.

2.2.1.1 Fases de la Metodología OMT

Las fases que conforman la Metodología OMT son:

- ✓ Análisis
- ✓ Diseño
- ✓ Implementación

2.2.1.1.1 Análisis

En la etapa de análisis se determina **QUÉ** hará el sistema, se realiza el estudio de la situación actual y se define globalmente un nuevo sistema que satisfaga los requerimientos planteados. Esta etapa está dividida en dos subetapas:

- **Investigación**

La investigación implica el levantamiento de toda la información necesaria para evaluar el sistema actual y conocer los requerimientos.

⁸ Referencia OMT

- **Diseño Conceptual**

La finalidad del diseño conceptual es plantear propuestas alternativas que, teniendo en cuenta las restricciones que debían admitirse, satisfagan los requerimientos.

En cualquier paradigma se puede identificar 5 áreas de esfuerzo para la etapa de análisis.

1. Reconocimiento del problema

Comprende el estudio de la especificación del sistema (si existe) y el plan de proyecto, reconocer los elementos básicos del problema y cómo los percibe el cliente.

2. Evaluación y síntesis

- Estudio de la situación actual.
- Identificación de requisitos de datos, funciones y comportamiento.

La evaluación es definir lo observado, evaluar los problemas actuales y la información deseada mientras que la síntesis es presentar uno o más enfoques o soluciones globales.

3. Modelado

- De la situación actual (es necesario cuando el analista no conoce las reglas de negocio)
- De la situación propuesta

Durante la evaluación y la síntesis el analista crea modelos en un esfuerzo para entender mejor el flujo de datos y de control, el tratamiento funcional, el comportamiento operativo y el contenido de la información. Los modelos sirven como fundamento para el diseño de software y son la base para la generación de una especificación de software.

4. Especificación

- Refinación y análisis de los requisitos (funcionales y no funcionales) para verificar que estén completos, claros y consisos.

- Elaborar la especificación (SRS = Software Requirements Specification)

El resultado de la ingeniería de sistemas es una representación del sistema. Puede ser un prototipo, una especificación, un modelo simbólico, etc. En cualquier caso debe comunicar la funcionalidad y características de comportamiento que se desea en el sistema que se va a construir.

5. Revisión y aprobación

El producto debe ser revisado para determinar que este completo, claro y consiso para su aprobación.

2.2.1.1.2 Diseño

El diseño del sistema tiene como finalidad traducir los resultados de la etapa de análisis a un esquema técnico que indica **CÓMO** el sistema cumplirá con los requerimientos del usuario.

A partir de los modelos del análisis decidir cómo construir el sistema (trazar los planos de la aplicación).

- Crear y mantener los diseños de la base de datos en base a los modelos de datos.
- Crear y mantener los diseños de las aplicaciones con base al diseño de la base de datos y a los modelos funcionales.
- Crear y mantener los diseños arquitectónicos (sitios de instalación, caminos de comunicaciones, distribución de la aplicación)
- Realizar la distribución tecnológica

El objetivo principal del diseño es entregar las funciones requeridas por el usuario tomando en cuenta consideraciones sobre:

- **Rendimiento**

Es la forma de trabajo del sistema. El rendimiento se expresa en términos de:

- Volumen de procesamiento: transacciones por unidad de tiempo.
- Tiempo de corrida: Tiempo para procesar una cantidad de trabajo.

- Tiempo de respuesta: Tiempo transcurrido desde el enter hasta la aparición de la respuesta en el Terminal.

- **Control**

Los sistemas deben tomar en cuenta procesos especiales de control así como procesos de recuperación de la información.

- Uso de dígitos verificadores
- Creación de libros diarios y líneas de auditoría.
- Uso de totales de control en proceso en lotes.
- Limitaciones de accesos a los programas y archivos.

- **Cambiabilidad**

Es la facilidad que tienen los sistemas de ser modificados ya sea para suprimir un error o instalar una mejora. Este tiene un impacto muy grande en los costos de mantenimiento del software. Los sistemas más fáciles de modificar son aquellos que están formados de módulos pequeños e independientes hasta que sea posible y que tiene su función o tarea bien definida, por que podrán ser sacados y modificados fácilmente sin que altere todo el sistema.

Estos factores son inversamente proporcionales.

Un sistema con controles de mucha seguridad tiende a degradar el rendimiento.

Un sistema diseñado para un alto rendimiento presentará dificultades para resultados de requerimientos complejos.

2.2.1.1.3 Implementación

Es la forma de llegar a contar con el sistema que atienda los requerimientos del usuario de acuerdo al diseño obtenido. Un sistema puede ser completado por construcción por compra o por migración.

- **Construcción**

Creación de los archivos de la base de datos definida en la etapa de diseño, ingreso de datos en las tablas de códigos.

Creación de los programas de la aplicación (lógica de proceso de datos y controles o interfaces), basándose en la documentación preparada en la etapa de diseño y ajustándose a las normas y estándares existentes para el efecto.

Los productos de esta etapa serán:

- Las bases de datos creadas
- Las interfaces de usuario
- Reportes
- Consultas
- Menús
- Librerías de código
- Librerías de instalación
- Manuales

- **Compra**

Investigación acerca de paquetes que cubran los requerimientos determinados en el análisis.

Evaluación de las alternativas encontradas mediante pruebas y pilotos

Proceso de adquisición

- **Migración**

Construcción del sistema con herramientas nuevas para reemplazar sistemas obsoletos

La Metodología OMT emplea tres clases de modelos para describir el sistema.

- ✓ Modelo de Objetos
- ✓ Modelo Dinámico
- ✓ Modelo Funcional

2.2.1.1.1 Modelo de Objetos

El modelo de objetos representa los objetos del sistema. En él se define la estructura de los objetos y clases así como las relaciones que les unen.

Comprende tanto un Diagrama de Clases como un Diccionario de Datos que las explique. Este modelo debe ser refinado por medio de la iteración.

- Identificar las clases
- Preparar un diccionario de clases (datos)
- Identificar las asociaciones entre objetos.
- Identificar atributos
- Organizar y simplificar clases usando herencia.
- Verificar que existan trayectorias en el modelo de objetos para preguntas probables
- Iterar y refinar el modelo
- Agrupar las clases en módulos

2.2.1.1.2 Modelo Dinámico

El modelo dinámico representa la interacción entre esos objetos representados como eventos, estados y transiciones.

Los pasos que se siguen en el modelo dinámico son:

- Preparar escenarios de una interacción típica.
- Identificar eventos entre objetos preparando un trazado de eventos para cada escenario.
- Construir los diagramas de estados.
- Revisar los eventos entre los objetos para verificar su consistencia.

2.2.1.1.3 Modelo Funcional

El modelo funcional representa los métodos del sistema desde la perspectiva de flujo de datos.

- Especifica el significado de las operaciones o métodos en el modelo de objetos y de las acciones en el modelo dinámico.
- Muestra cómo se calculan los valores sin importar la secuencia, las decisiones ni la estructura de los objetos.
- Se utilizan diagramas de flujo de datos para mostrar las dependencias funcionales.

2.3 LENGUAJE UNIFICADO DE MODELADO UML

Lenguaje Unificado de Modelado o UML por sus siglas en inglés Unified Modeling Language; es una combinación de las mejores prácticas en la industria de tecnología de software orientado a objetos, especialmente de la simbología de los tres principales modelos de análisis y diseño orientado a objetos de Booch (Booch), OMT Object Modeling Technique (Rumbaugh) y OOSE Object Oriented Software Engineering (Jacobson)⁹.

Es un lenguaje gráfico para la especificación, visualización, construcción y documentación de piezas de información usadas y producidas durante el proceso de desarrollo de software.

Es una respuesta de la OMG Object Management Group para definir una notación estándar para el modelado de aplicaciones construidas mediante objetos.

2.3.1 OBJETIVOS DEL UML

Definir formalmente un metamodelo para expresar modelos de análisis y diseño.

Intercambiar mecanismos entre las distintas herramientas de software que implementan ADOO (Análisis y Diseño Orientado a Objetos).

Mostrar una notación comprensible por el ser humano para representar modelos ADOO.

Proveer a los desarrolladores un lenguaje de modelamiento visual listo para utilizar.

Proporcionar mecanismos de extensión y de especialización para ampliar los conceptos básicos.

Ser independiente de los lenguajes de programación y de los métodos y procesos de desarrollo de software

2.3.2 DIAGRAMAS EN UML

- ✓ Diagrama de Casos de Uso
- ✓ Diagrama de Actividades

⁹ César Liza Avila, Modelando con UML Principios y Aplicaciones, 1era Edición, Agosto 2001

- ✓ Diagrama de Clases
- ✓ Diagrama de Colaboración
- ✓ Diagrama de Componentes
- ✓ Diagrama de Estados
- ✓ Diagrama de Secuencia

2.3.2.1 Diagrama de Casos de Uso

Representa un conjunto de casos de uso, actores y sus relaciones. Organiza los comportamientos del sistema.

Representa lo que hace el sistema y cómo se relaciona con su entorno. Es un diagrama utilizado para especificar el comportamiento externo de un sistema, es decir el comportamiento real con lo que nos rodea.

Representa una secuencia de acciones realizadas por el sistema que produce un resultado observable y de valor para algo o para alguien.

Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja actualmente, o de cómo se desea que trabaje. No es realmente un enfoque orientado a objetos, más bien es un enfoque de construcción de escenarios en los cuales se modelan los procesos del sistema. Sin embargo, constituye un buen modo de llevar a cabo la fase de captura de requisitos del sistema al comienzo del análisis orientado a objetos.

Típicamente, se modela un Caso de Uso para cada escenario en el sistema o negocio. Cada Caso de Uso puede estar definido simplemente por una sentencia de texto que describe el escenario. También se puede describir mediante una secuencia de pasos ejecutados dentro del escenario o condiciones pre-post para que el escenario comience o termine, respectivamente.

Un Caso de Uso es representado por una elipse y describe una situación de uso del sistema interactuando con actores.



Figura 2.2 Grafo Caso de Uso

Un actor es un agente externo al sistema, alguien o algo que solicita un servicio al sistema o actúa como catalizador para que ocurra algo, es representando mediante el siguiente grafo.

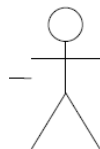


Figura 2.3 Grafo Actor

UML especifica que un actor es una clase de objetos, no una instancia particular de una clase. Durante el análisis del negocio se puede construir un diagrama de Casos de Uso que represente al sistema y dibujar paquetes que representen los diferentes dominios (subsistemas) del sistema. Para cada paquete se puede crear un diagrama de Casos de Uso hijo donde se describen los casos de uso del dominio. Esto se puede repetir refinando un Caso de Uso en un nuevo diagrama hijo, y así sucesivamente creando una jerarquía de diagramas de Casos de Uso.

Un diagrama de casos de uso muestra las relaciones entre los actores y los casos de uso dentro del sistema. Estas relaciones pueden ser:

- Asociación entre actores y casos de uso.
- Generalización entre actores.
- Generalización entre casos de uso.
- Incluye (include) entre casos de uso.
- Extiende (extend) entre casos de uso.

La **relación de asociación** representa la participación de un actor en un caso de uso, siempre parte de los actores y viajan en una sola dirección, también se la conoce como relación de comunicación y suele estereotiparse como `<<communicates>>` es la única relación posible entre un actor y un caso de uso, se la representa mediante una línea sólida.

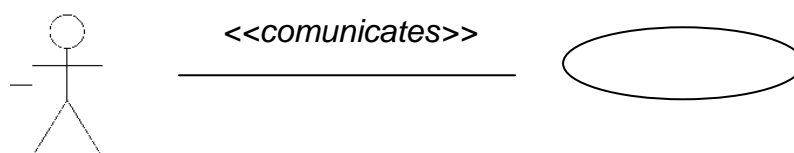


Figura 2.4 Relación de Asociación de Caso de Uso Actor

La **relación de generalización** trata de representar la relación entre dos objetos del mismo tipo en el cual uno de ellos se comporta igual que el otro pero tiene características adicionales que lo diferencian. Este tipo de relación es una relación de herencia. Se representa con una línea sólida con cabeza de flecha hueca apuntando desde el caso de uso hijo al caso de uso padre o desde el actor hijo al actor padre.

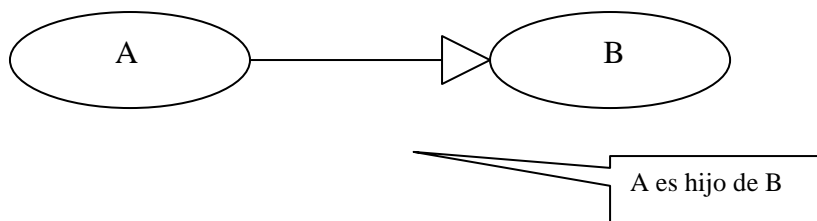


Figura 2.5 Relación de Generalización entre casos de uso

Los casos de uso a menudo son incluidos como parte de otro u otros casos de uso. Una relación incluye entre casos de uso significa que el caso de uso base incorpora explícitamente el comportamiento de otro caso de uso. El caso de uso base siempre utiliza el caso de uso incluido. El objetivo de esta relación es permitir invocar el mismo comportamiento muchas veces, colocando el comportamiento común en un caso de uso que puede ser invocado por otro u otros casos de uso. Esta relación es de dependencia puesto que su ejecución depende del caso de uso base, pues es este quien lo invoca. El caso de uso incluido no puede ejecutarse sin el caso de uso que lo incluye. Se representa mediante una línea discontinua con una cabeza de flecha abierta, desde el caso de uso base hacia el caso de uso incluido.

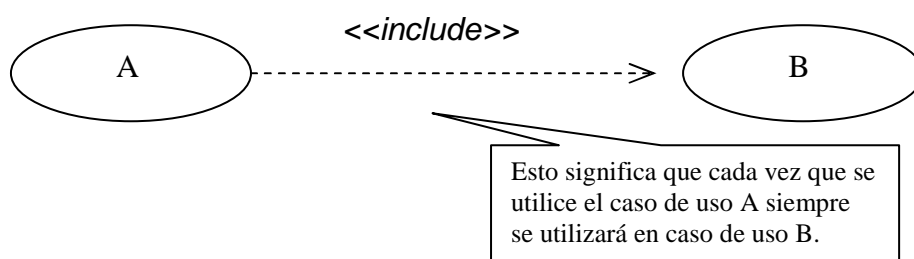


Figura 2.6 Relación Incluye entre casos de uso

Una **relación extend** entre casos de uso significa que se ejecuta el caso de uso base pero, bajo ciertas condiciones, este caso de uso llama a otro caso de uso

que extiende el comportamiento del primero. Se representa mediante una línea discontinua con una cabeza de flecha abierta, desde el caso de uso extendido hacia el caso de uso base.

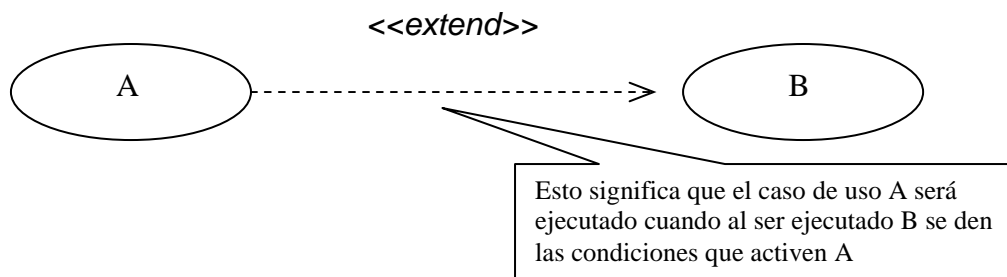


Figura 2.7 Relación Extend entre casos de uso

2.3.2.2 Diagrama de Actividades

Muestra el flujo de actividades de un sistema. Permite modelar la función de un sistema y resaltar el flujo de control entre objetos.

Una actividad representa la realización de una o varias tareas y causa un cambio en el sistema. Una actividad puede estar compuesta por otras actividades y por acciones, y puede ser descompuesta en otros diagramas de Actividades, una actividad puede ser interrumpida y toma algún tiempo en completarse.

Una acción es una actividad que no puede ser descompuesta.

Las **actividades** y las **acciones** se representan con un rectángulo con las puntas redondeadas.

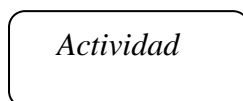


Figura 2.8 Grafo Actividad

La **transición** es el paso de la terminación de una actividad para el comienzo de otra. Este paso se representa mediante una línea que va desde la actividad que finaliza hacia la actividad a ejecutarse.

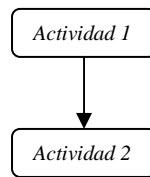


Figura 2.9 Transición entre actividades

En los diagramas de actividades se pueden incluir caminos alternativos dependiendo de una condición. Estas **condiciones** se representan por un rombo al cual llega la transición de la actividad origen y del cual salen las múltiples transiciones a cada actividad destino, cada una con una condición diferente verdadera o falsa.

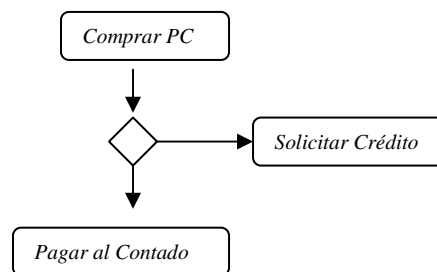


Figura 2.10 Condición entre flujo de actividades

Las decisiones implican dividir un flujo de actividades en diferentes caminos. UML proporciona una forma de volver a unirlos con el mismo símbolo, a esta acción de unir dos flujos se la denomina merge. Un **merge** tiene dos o más flujos entrantes pero solo uno saliente.

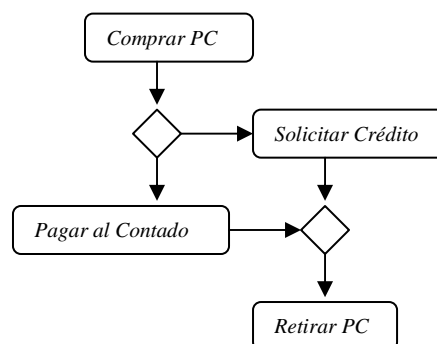


Figura 2.11 Merge entre flujo de actividades

En algunos flujos de procesos especialmente flujos de procesos de negocios, algunas actividades pueden realizarse simultáneamente, lo que lo diferencia de un flujo de procesos tradicional. Estas actividades se las puede modelar con las barras de sincronización. Una **barra de sincronización** se representa con una línea vertical u horizontal gruesa.

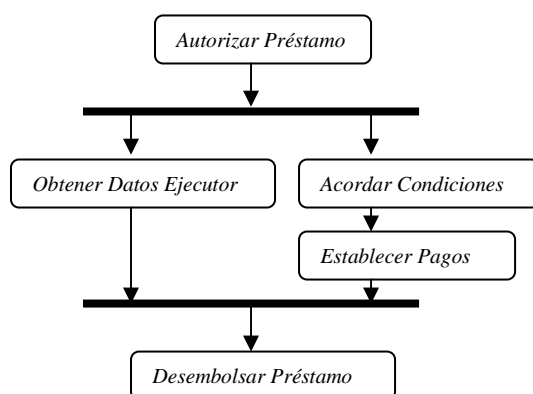


Figura 2.12 Barra de sincronización entre flujo de actividades

Un flujo de actividades tiene un inicio y un fin. Para representar el inicio se utiliza un círculo completamente relleno y para el fin se usa un círculo relleno dentro de una circunferencia.



Figura 2.13 Grafo Inicio y Fin de flujo de actividades

Un **carril** es una franja dentro del diagrama de actividades que muestra las actividades que están bajo la responsabilidad de algún objeto, persona, área organizacional, máquina o de algún rol en particular. Los carriles se representan mediante líneas verticales continuas a ambos lados. El orden de los carriles no tiene significado alguno.

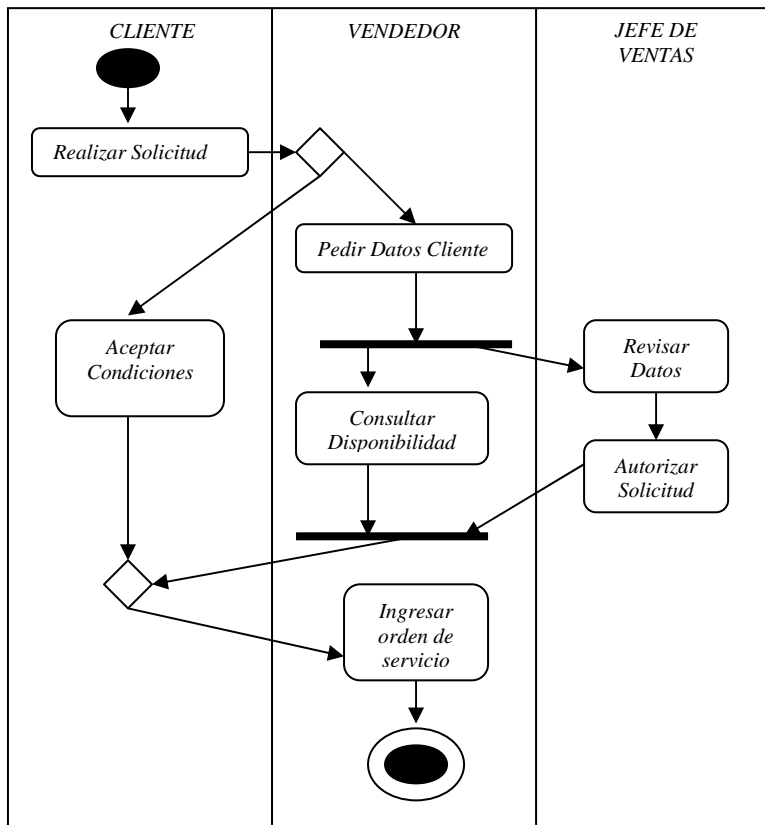


Figura 2.14 Grafo Carril de sepación de actividades

Cuando un objeto tiene o es responsable de algunas actividades se le asigna un carril del diagrama de actividades. En muchas ocasiones un objeto cambia de estado cuando transcende de una actividad a otra. Esto se representa con un rectángulo que denota a un objeto y entre corchetes indicar el estado del mismo.

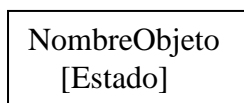


Figura 2.15 Grafo Representación Objeto

Los objetos que salen de una actividad son representados con el rectángulo característico y con una flecha discontinua desde la actividad hacia el objeto. Inversamente sucede cuando un objeto entra a una actividad. Un objeto regularmente es la salida de una actividad y la entrada de una o más actividades.

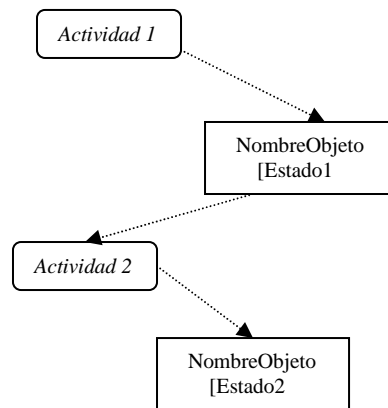


Figura 2.16 Representación Actividad y Objeto

2.3.2.3 Diagrama de Clases

Muestra un conjunto de clases (grupos de objetos que tienen las mismas características y comportamientos), interfaces y colaboraciones y las relaciones entre ellas. Estos diagramas son los más comunes en el modelado de sistemas orientados a objetos y cubren la vista estática de un sistema.

Las **clases** contienen atributos y operaciones. Las **relaciones** entre clases pueden ser de dependencia, generalización y asociación.

Las clases son representadas mediante un rectángulo con tres divisiones internas llamadas compartimientos, en los cuales se indica el nombre de la clase, sus atributos (lo que caracteriza a la clase) y operaciones (son la forma de cómo interactúa un objeto de la clase con su entorno).

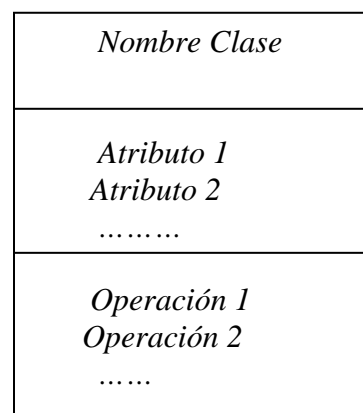


Figura 2.17 Grafo Representación de una Clase

Los atributos de una clase pueden ser accedidos por todas las clases o por las clases en las que son definidos o por los descendientes de las clases donde son definidos. A esta característica se la llama **visibilidad** (*visibility*), existen tres tipos de visibilidad: public (+), private (-), protected (#).

Public indica que el atributo será visible tanto dentro como fuera de la clase, se lo representa con el signo. +

Private indica que el atributo sólo será accesible desde dentro de la clase, sólo sus métodos los podrán acceder, se lo representa con el signo. –

Protected indica que el atributo será accesible por métodos de la clase en la cual se lo define además de las subclases que se dividen de ella, se lo representa con el signo. #

Cuando una visibilidad no esta especificada se asume que es public.

La **multiplicidad** de los atributos muestra la cantidad de veces que un atributo se repite, se lo coloca después del nombre del atributo entre corchetes.

El **tipo** de atributo es el tipo de dato que tiene el atributo depende del lenguaje de programación tales como int, char, flota, date, string, etc.

Las operaciones listadas en el tercer compartimiento de una clase, describen el comportamiento de los objetos de una clase, es decir, como la clase interactúa con su entorno.

Una **operación** representa el servicio que puede ser requerido por una instancia de la clase y afecta su comportamiento. Un **método** es la implementación de una operación, es la forma específica de cómo se lleva a cabo una operación. La visibilidad de las operaciones es la misma especificada para los atributos.

Para representar casos particulares de clases tenemos diferentes tipos de conceptos tales como: abstracta, parametrizada, asociación, activa, utilidad, interfaz, hoja, raíz, metaclass.

Las **clases abstractas** son aquellas que no tienen instancias y son aquellas que sirven para definir subclases las cuales si podrán ser instanciadas. Las clases abstractas se denotan con el nombre de la clase en letra itálica y opcionalmente colocando la etiqueta *{abstract}* cualquier clase que no sea abstracta será concreta o simplemente clase.

El **polimorfismo** son operaciones con el mismo nombre pero con diferente implementación.

La **clase parametrizada** permite escribir una plantilla de clase, que se pueden aplicar a varios casos que se diferencian solo mediante de los tipos de parámetros enviados.

Una clase de asociación modela las propiedades de una relación de asociación entre dos clases.

Una **clase activa** es una clase cuyos objetos son activos (tienen su propio flujo de control), a diferencia de los objetos pasivos que solo reaccionan al enviarles un mensaje. Una clase activa es concurrente es decir tiene dos o más procesos en ejecución.

La **clase utilidad** es una agrupación de variables globales y procedimientos públicos declarados en forma de una clase.

La **clase interfaz** consta solo de operaciones, son útiles para otras clases que solicitan sus servicios logrando menor acoplamiento entre clases, facilitando el mantenimiento ante posibles cambios.

La **clase hoja** es la que no tiene descendientes y se encuentra en el último nivel de descendencia de jerarquía de clases.

La **clase raíz** no tiene padres y se encuentra en el nivel más alto de jerarquía de clases.

La **metaclase** es un tipo de clase en la cual sus instancias son otras clases.

Existen tres tipos de relaciones entre clases:

- Relación de dependencia.
- Relación de generalización.
- Relación de Asociación.

La **relación de dependencia** es una relación semántica entre dos elementos en el cual el cambio en un elemento (el elemento independiente), puede afectar la semántica del otro elemento (elemento dependiente). El elemento dependiente es aquel que necesita del independiente para poder cumplir su responsabilidad. Se representa con una línea discontinua, dirigida según el sentido de la dependencia que a veces incluye una etiqueta y un estereotipo que dan una explicación del tipo de dependencia presentada.

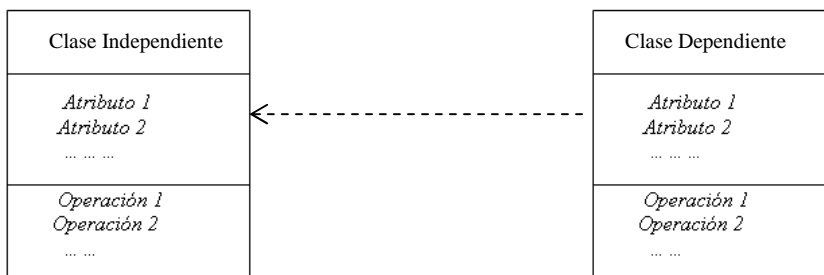


Figura 2.18 Relación de dependencia entre clases

La **relación de generalización** es una relación entre dos clases donde una de ellas, llamada subclase o clase hija hereda los atributos y los comportamientos de la otra llamada superclase o clase padre. Esta relación involucra el mecanismo de herencia por el cual el hijo comparte la estructura y el comportamiento visible (public, protected) del padre, pero además puede tener visibilidad propia (private). Se representa mediante una línea continua con punta de flecha hueca dirigida desde la clase hija hacia la clase padre.

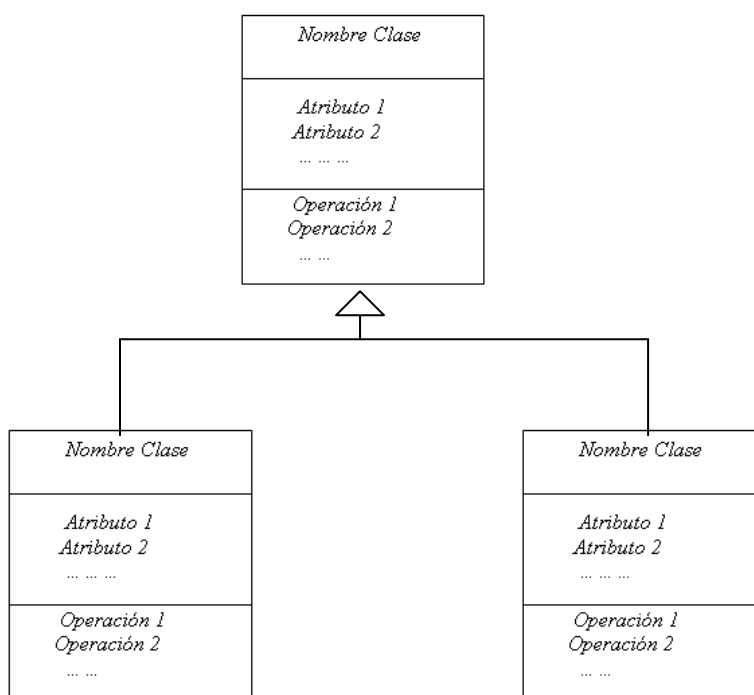


Figura 2.19 Relación de generalización entre clases

Las relaciones de asociación son relaciones bidireccionales potenciales entre instancias de clases. En una asociación, una instancia de una clase puede estar relacionada (potencialmente) con un número mínimo y máximo instancias de otra

clase (o de la misma clase). Estos números definidos desde la perspectiva de cada una de las clases que están asociadas se denominan cardinalidad de la asociación.

Uno y sólo uno: 1

Cero o uno: 0..1

Uno o más: 1..*

Cero o más: 0..*

Entre N y M : N..M , donde N y M son números naturales y $M \geq N$.

2.3.2.4 Diagrama de Colaboración

Es un diagrama de interacción que resalta la organización estructural de los objetos que envían y reciben mensajes. Los diagramas de colaboración muestran la colaboración entre objetos para realizar una tarea mediante el uso de mensajes enviados entre ellos, muestran el contexto de la operación y no reservan una dimensión para el tiempo si no que enumeran los mensajes para indicar la secuencia.

Tienen el siguiente aspecto.

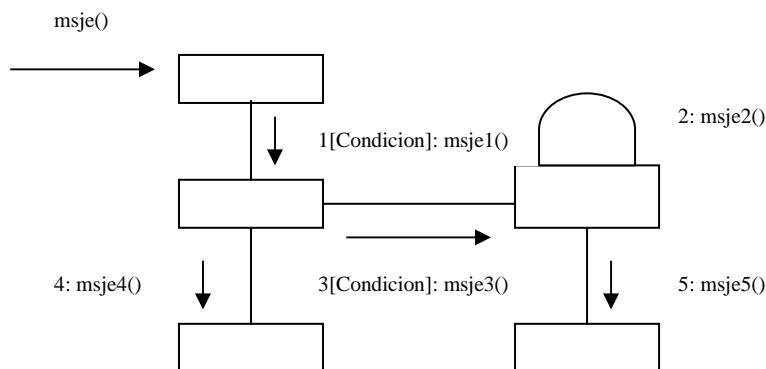


Figura 2.20 Ejemplo de Diagrama de Colaboración

Los elementos que intervienen en el diagrama de colaboración son los enlaces, numeraciones, condiciones, iteraciones entre otros.

El objeto se representa con un rectángulo que contiene el nombre y la clase del objeto en el siguiente formato. **nombreObjeto: nombreClase.**

El flujo de mensaje representa el envío de un mensaje se representa mediante una flecha dirigida colocada cerca de un enlace, esta flecha va acompañada del nombre del mensaje con sus respectivos parámetros entre paréntesis.

Existen mensajes al mismo objeto se representa como un enlace consigo mismo.

2.3.2.5 Diagrama de Componentes

Muestra un conjunto de componentes y sus relaciones. Describe la vista de implementación estática del sistema. Permiten visualizar las partes de un sistema, mostrando las diferentes formas en que pueden ensamblarse para construir ejecutables, modela la dimensión física del software es decir, como los componentes se almacenan en archivos de disco. Los diagramas de componentes están formados por componentes, interfaces y relaciones entre ellos.

Los **componentes** son cada una de las partes físicas y reemplazables de un sistema formados por un conjunto de elementos lógicos, representan la implementación física de un paquete que contiene elementos lógicos como clases, interfaces y colaboraciones. Existen diferentes tipos de componentes, entre ellos los más conocidos son los ejecutables, librerías, tablas, los archivos de código fuente, documentos, archivos de ayuda, archivos de datos, scripts, archivos de login, archivos de inicialización, archivos de instalación y desinstalación, archivos de texto, además de los componentes comerciales provistos por muchos sistemas operativos. Se representan con un rectángulo que tiene en uno de sus lados dos pequeños rectángulos sobresalientes.

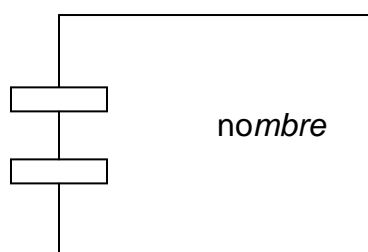


Figura 2.21 Grafo Representación de componente

Existe una estrecha relación entre las clases y los componentes, los componentes implementan a una o más clases, las clases son abstracciones mientras que los

componentes son reales, los componentes se corresponden con los paquetes, sin embargo los paquetes son contextuales pues, existen en tiempo de desarrollo, mientras que los componentes son físicos ya que existen en tiempo de ejecución. Los componentes pueden participar en relaciones de dependencia, generalización y asociación. Los componentes pueden alcanzar las operaciones de las clases solo mediante interfaces. UML define cinco estereotipos estándar para los componentes.

Executables son aquellos componentes que pueden ejecutarse en un nodo.

Library son las librerías de nuestros programas ya sean estáticas o dinámicas.

Table es una tabla de una base de datos.

File es un archivo de código fuente, datos o simplemente texto.

Document es un documento como las ayudas.

La importancia de los componentes se resume en 4 aristas.

- Permiten observar la estructura final del sistema.
- Facilitan la documentación técnica.
- Permiten probar y modificar el sistema.
- Tienen potencial de ser reutilizados en la construcción de otro sistema.

Una interfaz es el rostro que presenta al mundo un componente o una clase con la cual interaccionan otros componentes o clases. Se representa mediante un círculo con una línea que lo une al componente que lo implementa.

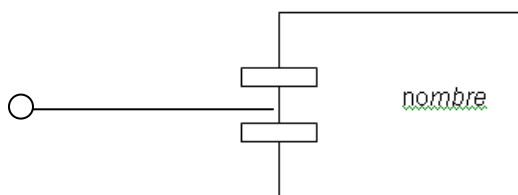


Figura 2.22 Representación de una interfaz

En su forma expandida una interface se representa como una clase estereotipada como <<interface>> mostrando sus atributos y operaciones.

El componente que implementa la interface es conectado mediante una relación de realización con una línea discontinua terminada en una cabeza hueca que parte del componente a la interface.

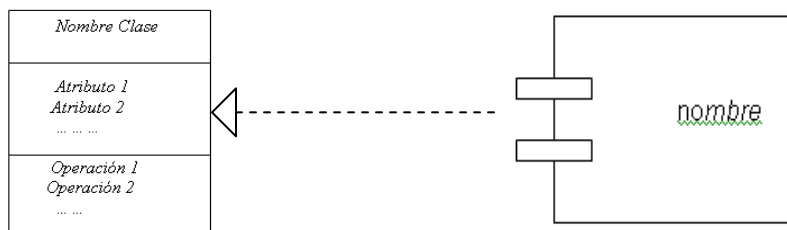


Figura 2.23 Relación de realización

Los componentes que utilizan la interfaz implementada por otro componente se representan mediante una **relación de dependencia**, se representa con una línea discontinua con cabeza de flecha abierta apuntando hacia la interfaz.

Si la interfaz es representada en forma compacta el uso de la interfaz por un componente que no lo implementa será de la siguiente manera.

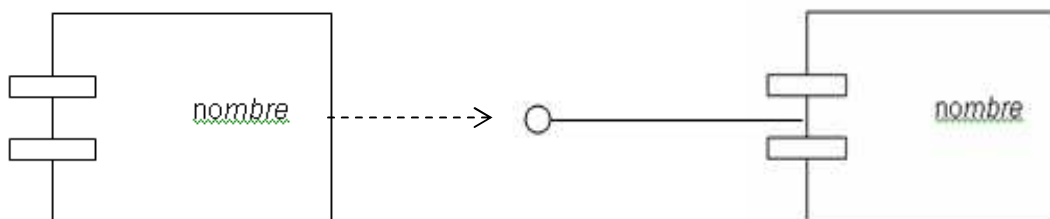


Figura 2.24 Relación de dependencia entre componentes

Mientras que si la interfaz se representa de forma extendida, el uso de la interfaz por un componente que no lo implementa será:

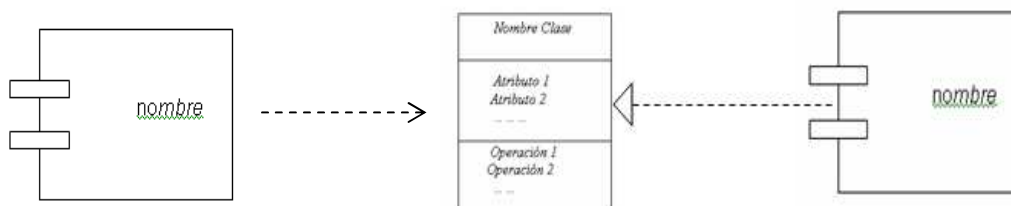


Figura 2.25 Representación de interfaz en forma extendida.

Las interfaces que un componente realiza son conocidas como interfaz de **exportación** (export interface) esto es un componente exporta una interfaz a otros. Las interfaces que usan los componentes y que son provistas por otros componentes se denominan interfaz de **importación** (import interface). Una

variables de estado (valores característicos de los atributos) y el último para las acciones

El estado inicial representado por un círculo relleno y el estado final representado por un círculo relleno rodeado de una circunferencia.

Un evento es una concurrencia que puede causar la transición del objeto de un estado a otro.

La transición es una acción que conceptualmente es instantánea no interrumpible.

2.3.2.7 Diagrama de Secuencia

Es un diagrama de interacción que resalta la ordenación temporal de los mensajes. Presenta un conjunto de objetos y los mensajes enviados y recibidos por esos objetos.

La creación de diagramas de secuencia toma parte de la investigación para conocer el sistema, por lo que es parte del análisis del mismo.

Los objetos que interactúan se colocan a lo largo de una línea horizontal imaginaria, mientras que los mensajes enviados por estos objetos se van colocando a lo largo del eje Y, y según su orden de aparición cronológica, donde ocurran más tarde, más abajo se ubicarán. El objeto que inicia la iteración se ubica en la parte superior izquierda del diagrama, mientras que los otros objetos se van colocando sucesivamente hacia la derecha del mismo.

Para su representación se hace uso de diferentes elementos objetos, actores, líneas de vida, focos de control y mensajes.

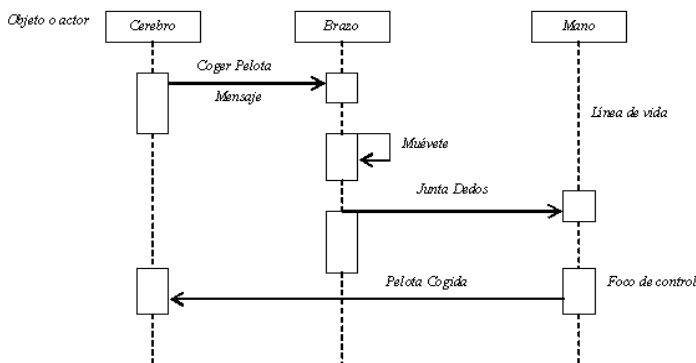


Figura 2.28 Ejemplo de Diagrama de Secuencia

Los objetos son representados con los rectángulos de la parte superior, los mensajes son las flechas que salen de los focos de control que se representan con rectángulos verticales y las líneas de vida son las líneas discontinuas que nacen en el objeto atravesando los focos de control.

Las líneas de vida de un objeto, indica la vida del objeto durante la interacción. Debe tenerse en cuenta que algunos objetos pueden ser creados y destruidos por la interacción. La creación se denota por una línea estereotipada con create, mientras que la destrucción se muestra con una X al final de su línea de vida.

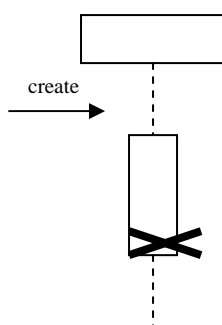


Figura 2.29 Ejemplo de creación y final de líneas de vida

Los focos de control muestran el período de tiempo en el cual el objeto se encuentra desarrollando alguna operación, bien sea por si mismo o por medio de la delegación a través de sus procedimientos subordinados.

Los mensajes entre objetos se representan mediante la línea sólida dirigida desde el objeto emisor del mensaje hacia el objeto receptor. Este ultimo ejecutara la acción indicada por el emisor.

2.4 HERRAMIENTAS CASE (Computer Aided Software Engineering, Ingeniería Asistida por Computadora)

Las herramientas Case son un conjunto de métodos utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo del sistema de información, completamente o en algunas fases, son un conjunto de:

- Utilidad.
- Métodos.
- Técnicas.

También pueden mejorar la productividad en el desarrollo de una aplicación de bases de datos. Y por productividad se entiende tanto la eficiencia en el desarrollo, como la efectividad del sistema desarrollado.¹⁰

La eficiencia se refiere al costo, tanto en tiempo como en dinero, de desarrollar la aplicación.

La efectividad se refiere al grado en que el sistema satisface las necesidades de los usuarios. Para obtener una buena productividad, subir el nivel de eficacia puede ser más importante que aumentar la eficiencia.

2.4.1 Clasificación de herramientas CASE

Las herramientas CASE pueden clasificarse por su función, su papel como instrumentos para administradores o personal técnico, por su utilización en los distintos pasos del proceso de ingeniería del software, la arquitectura de entorno (hardware y software) que les presta su apoyo, o incluso por su origen o su coste. En muchos casos, las únicas herramientas disponibles para el ingeniero del software eran compiladores y editores de texto. Estas herramientas abarcan solo la codificación, actividad que no debería de ocupar mas del 20% del proceso global del software.

Herramientas de la ingeniería de la información. Al modelar los requisitos de información estratégica de una organización, las herramientas de la ingeniería de la información proporcionan un metamodelo del cual se derivan sistemas de información específicos. En lugar de centrarse en los requisitos de una aplicación específica, estas herramientas CASE modelan la información de negocios cuando esta se transfiere entre distintas entidades organizativas en el seno de una compañía. El objetivo primordial de las herramientas de esta categoría consiste en representar objetos de datos de negocios, sus relaciones, y la forma en que fluyen estos objetos de datos entre distintas zonas de negocio en el seno de la compañía.

¹⁰María Mercedes Marqués Andrés, <http://www3.uji.es/~mmarquez/f47/apun/node75.html>, Febrero 2001

Modelado de procesos y herramientas de administración. Si una organización intenta mejorar un proceso de negocios (o de software) lo primero que debe de hacer es entenderlos. Las herramientas de modelado de procesos (también denominadas herramientas de tecnología de procesos) se utilizan para representar los elementos clave del proceso para entenderlo lo mejor posible. Estas herramientas también pueden proporcionar vínculos con descripciones de procesos que ayuden a quienes estén implicados en el proceso de comprender las tareas que se requieren para llevar a cabo ese proceso. Además, también pueden proporcionar vínculos con otras herramientas que proporcionen un apoyo para actividades de proceso ya definidas.

Herramientas de planificación de proyectos. Las herramientas de esta categoría se concentran en dos áreas primordiales: estimación de esfuerzos de proyecto y de costes de software, y planificación de proyectos. Las primeras calculan su esfuerzo estimado, la duración del proyecto y su número de personas empleando una o más de las técnicas presentadas. Por su parte, las herramientas de planificación de proyectos capacitan al administrador para definir todas las tareas del proyecto (la estructura de desglose de tareas), para crear una red de tareas (normalmente empleando una entrada gráfica), para representar la interdependencia entre tareas y para modelar la cantidad de paralelismo que sea posible para ese proyecto.

Herramientas de análisis de riesgos. La identificación de riesgos potenciales y el desarrollo de un plan para mitigar, monitorizar y administrar esos riesgos tiene una importancia fundamental en los grandes proyectos. Estas herramientas en si, capacitan al administrador del proyecto para construir una tabla de riesgos proporcionando una guía detallada en la identificación y análisis de riesgos.

Herramientas de administración de proyectos. La planificación del proyecto y el plan del proyecto deben de seguirse y de monitorizarse de forma continua. Además, el gestor deberá de utilizar las herramientas que recojan métricas que en ultima instancia proporcionen una indicación de la calidad del producto del

software. Las herramientas de esta categoría suelen ser extensiones de herramientas de planificación de proyectos.

Herramientas de seguimiento de requisitos. Cuando se desarrollan grandes sistemas, el sistema proporcionado suele no satisfacer los requisitos especificados por el cliente. El objetivo de estas herramientas es proporcionar un enfoque sistemático para el aislamiento de requisitos, comenzando por la solicitud del cliente de una propuesta (RFP) o especificación. Las herramientas de trazado de requisitos típicas combinan una evaluación de textos por interacción humana. Con un sistema de gestión de bases de datos que almacena y categoriza todos y cada uno de los requisitos del sistema que se analizan partir de la RFP o especificación original.

Herramientas de métricas y gestión. Las métricas de software mejoran la capacidad del administrador para controlar y coordinar el proceso del software y la capacidad del ingeniero para mejorar la calidad del software que se produce. Las métricas y herramientas de medida actuales se centran en procesos, proyectos y características del producto. Las herramientas orientadas a la administración capturan métricas específicas del proyecto (p. ej.: LDC/persona-mes, defectos por punto de función) que proporcionan una indicación global de productividad o de calidad. Las herramientas orientadas técnicamente determinan métricas técnicas que proporcionan una mejor visión de la calidad del diseño o del código.

Muchas de las herramientas métricas avanzadas mantienen una base de datos de medidas de medias de la industria. Basándose en características de proyectos y de productos proporcionados por el usuario, estas herramientas «califican» los números locales frente a los valores medios de la industria (y frente al rendimiento local anterior) y sugieren estrategias para llegar a mejoras.

Herramientas de documentación. Las herramientas de producción de documentos y de autoedición prestan su apoyo a casi todos los aspectos de la ingeniería del software, y representan una importante oportunidad de «aprovechamiento» para todos los desarrolladores de software. La mayor parte de

las organizaciones dedicadas al desarrollo de software invierte una cantidad de tiempo considerable en el desarrollo de documentos, y en muchos casos el proceso de documentación en 51' resulta bastante deficiente. No es infrecuente que una organización de desarrollo de software invierta hasta Un 20 0 un 30 por ciento de su esfuerzo global de desarrollo de software en la documentación. Por esta razón, las herramientas de documentación suponen una oportunidad importante para mejorar la productividad.

Herramientas de software de sistema. CASE es una tecnología de estaciones de trabajo. Por tanto, el entorno CASE debe adaptarse a un software de sistema en red de alta calidad, al correo electrónico, a los boletines electrónicos y a otras capacidades de comunicaciones.

Herramientas de control de calidad. La mayor parte de las herramientas CASE que afirman que tienen como principal interés el control de calidad son en realidad herramientas métricas que hace una auditoría del código fuente para determinar si se ajusta o no a ciertos estándares del lenguaje. Otras herramientas extraen métricas en un esfuerzo por extrapolar la calidad del software que se está construyendo.

Herramientas de gestión de bases de datos. El software de gestión de bases de datos sirve como fundamento para establecer una base de datos CASE (depósito), que también se denominará base de datos del proyecto. Dado el énfasis acerca de los objetos de configuración, las herramientas de gestión de bases de datos para CASE pueden evolucionar a partir de los sistemas de gestión de bases de datos relacionales (SGBDR) para transformarse en sistemas de gestión de bases de datos orientadas a objetos (SGBDOO).

Herramientas de gestión de configuración de software. La gestión de configuración de software (GCS) se encuentra en el núcleo de todos los entornos CASE. Las herramientas pueden ofrecer su asistencia en las cinco tareas principales de CICS: identificación, control de versiones, control de cambios, auditoría y contabilidad de estados. La base de datos CASE proporciona Un

mecanismo para identificar todos los elementos de configuración y relacionarlo con otros elementos; el proceso de control que se describa se puede implementar con ayuda de herramientas especializadas; un acceso sencillo a los elementos de configuración individuales facilita el proceso de auditoría; y las herramientas de comunicación CASE pueden mejorar enormemente la contabilidad de estados (ofreciendo información acerca de los cambios a todos aquellos que necesiten conocerlos).

Herramientas de análisis y diseño. Estas herramientas capacitan al ingeniero del software para crear modelos del sistema que haya que construir. Los modelos contienen una representación de los datos, de la función y del comportamiento (en el nivel de análisis), así como caracterizaciones del diseño de datos, arquitectura, procedimientos e interfaz. Al efectuar una comprobación de la consistencia y validez del modelo, las herramientas de análisis y diseño proporciona al ingeniero del software un cierto grado de visión en lo tocante a la representación del análisis, y ayudan a eliminar errores antes de que se propaguen al diseño, o lo que es peor, a la propia implementación.

Herramientas PRO/SIM. Las herramientas PRO/SIM (de prototipos y simulación) proporcionan al ingeniero del software la capacidad de predecir el comportamiento de un sistema en tiempo real antes de llegar a construirlo. Además, capacitan al ingeniero del software para desarrollar simulaciones del sistema de tiempo real que permitirán al cliente obtener ideas acerca de su funcionamiento, comportamiento, y respuesta antes de la verdadera implementación.

Herramientas de desarrollo y diseño de interfaz. Estas herramientas son en realidad un conjunto de primitivas de componente de programas tales como menús, botones, estructuras de ventanas, iconos, mecanismos de desplazamiento, controladores de dispositivos etc. Sin embargo, estos conjuntos de herramientas se están viendo sustituidos por herramientas de generación de prototipos de interfaz que permiten una rápida creación en pantalla de sofisticadas

interfaces de usuario, que se ajustan al estándar de interfaz que se haya adoptado para el software.

Herramientas de generación de prototipos. Se puede utilizar toda una gama de este tipo de herramientas, los generadores de pantallas permiten al ingeniero del software definir rápidamente la disposición de la pantalla para aplicaciones interactivas. Otras herramientas de prototipos CASE más sofisticadas permiten la creación de Un diseño de datos, acoplado con las disposiciones de la pantalla y de los informes simultáneamente. Muchas herramientas de análisis y diseño proporcionan extensiones que ofrecen alguna opción de generación de prototipos. Las herramientas PRO/SIM generan un esqueleto de código fuente en Ada y C para las aplicaciones de ingeniería (en tiempo real). Por último, una gama de herramientas de cuarta generación poseen también características de generación de prototipos.

Herramientas de programación. La categoría de estas herramientas abarca los compiladores, editores, y depuradores que están disponibles para prestar su apoyo en la mayoría de los lenguajes de programación convencionales. Además, los entornos de programación orientados a objetos (OO), los lenguajes de cuarta generación, los entornos de programación gráfica, los generadores de aplicaciones, y los lenguajes de consulta de bases de datos residen también en esta categoría.

Herramientas de integración y comprobación. En su directorio de herramientas de comprobación de software, Software Quality Engineering define las siguientes categorías de herramientas de comprobación:

- Adquisición de datos: herramientas que adquieren datos que son utilizados durante la comprobación.
- Medida estática: herramientas que analizan el código fuente sin ejecutar casos de prueba.
- Medida dinámica: herramientas que analizan el código fuente durante la ejecución.

- Simulación: herramientas que simulan las funciones del hardware o de otros elementos externos.
- Administración de comprobaciones: herramientas que prestan su asistencia en la planificación, desarrollo y control de las comprobaciones.
- Herramientas de funcionalidad cruzada: se trata de herramientas que cruzan los límites de las categorías anteriores.

Debería tenerse en cuenta que muchas de las herramientas de comprobación poseen características que abarcan dos o más de las categorías anteriores.

Herramientas de análisis estático. Estas herramientas prestan su asistencia al ingeniero del software a efectos de derivar casos prácticos. Se utilizan tres tipos distintos de herramientas estáticas de comprobación en la industria: Herramientas de comprobación basadas en código, lenguajes de comprobación especializados, y herramientas de comprobación basadas en requisitos. Las herramientas de comprobación basadas en código admiten Un código fuente (o PDL) como entrada, y efectúan un cierto número de análisis que dan lugar a la generación de casos de prueba. Los lenguajes de comprobación especializados (p. ej.: ATLAS) capacitan al ingeniero del software para escribir detalladas especificaciones de comprobación que describirán todos los casos de prueba y la logística de su ejecución. Las herramientas de comprobación basadas en requisitos aíslan los requisitos específicos del usuario y sugieren casos de prueba (0 clases de comprobaciones) que ejerciten estos requisitos.

Herramientas de análisis dinámico. Son herramientas que interactúan con un programa que se esté ejecutando comprobando la cobertura de rutas, las afirmaciones acerca del valor de variables específicas y en general instrumentan el flujo de ejecución del programa. Las herramientas dinámicas pueden ser intrusivas o no intrusivas. Las herramientas intrusivas modifican el software que hay que comprobar mediante sondas que se insertan (instrucciones adicionales) y que efectúan las actividades mencionadas anteriormente. Las herramientas no intrusivas utilizan un procesador hardware por separado que funciona en paralelo con el procesador que contenga el programa que se está comprobando.

Herramientas de gestión de comprobación. Son herramientas que se utilizan para comprobar y coordinar la comprobación de software para cada uno de los pasos principales de comprobación. Las herramientas de esta categoría administran y coordinan la comprobación de regresiones, efectúan comparaciones que determinan las diferencias entre la salida real y la esperada, y efectúan comprobaciones por lotes de programas con interfaces interactivas entre hombre y máquina. Además de las funciones indicadas anteriormente, muchas herramientas de gestión de comprobaciones sirven también como controladores de comprobación genéricos. Un controlador de comprobación lee uno o más casos de prueba de algún archivo de pruebas, da formato a los datos de prueba para que se ajusten a las necesidades del software que se está probando, e invoca entonces al software que sea preciso comprobar.

Herramientas de comprobación cliente/servidor. El entorno C/S exige unas herramientas de comprobación especializadas que ejerciten la interfaz gráfica de usuario y los requisitos de comunicaciones en red para el cliente y el servidor.

Herramientas de reingeniería. La categoría de herramientas de reingeniería se puede subdividir en las funciones siguientes:

- Herramientas de ingeniería inversa para producir especificaciones: se toma el código fuente como entrada y se generan modelos gráficos de análisis y diseño estructurados, listas de utilización y otras informaciones de diseño.
- Herramientas de estructuración y validación de código: se analiza la sintaxis del programa, se genera una gráfica de control de flujo y se genera automáticamente un programa estructurado; y
- Herramientas de reingeniería para sistemas en línea: se utilizan para modificar sistemas de bases de datos en línea (p. ej.: para convertir archivos IDMS o DB2 traduciéndolos a un formato de entidades y relaciones).

Muchas de las herramientas anteriores están limitadas a lenguajes de programación específicos (aun cuando se abarcan la mayoría de los lenguajes

principales) y requieren un cierto grado de interacción con el ingeniero del software.

Las herramientas de ingeniería inversa y progresiva de la próxima generación hará un uso mucho mayor de técnicas de inteligencia artificial, aplicando una base de conocimientos que sea específica del dominio de la aplicación (esto es, un conjunto de reglas de descomposición que se aplicarían a todos los programas de una cierta zona de aplicación tal como sí control de fabricación o la aviónica). El componente de inteligencia artificial asistirá en la descomposición y reconstrucción del sistema, pero seguirá requiriendo una interacción con un ingeniero de software a lo largo del ciclo de la reingeniería.

2.4.2 Clasificación de las herramientas CASE Cliente/Servidor

Las herramientas CASE Cliente/Servidor se pueden clasificar en dos grupos: las más modestas y baratas (como Visual Basic, Power Builder, Delphi, Erwin, etc.), y las llamadas herramientas integradas (IEF, Oracle CASE, etc.). Su costo está en proporción directa con su rango de aplicabilidad para desarrollar sistemas de información. Se ha demostrado que las herramientas del primer grupo no sirven para desarrollar sistemas de complejidad muy grande (sistemas distribuidos, multiplataformas, o cualquier otro que consuma gran cantidad de recursos durante su desarrollo). Esto influye claramente en las políticas de desarrollo de una empresa que posea alguna herramienta, de forma tal que se han desarrollado metodologías para elegir la herramienta CASE más acorde a las características del proyecto a llevar a cabo. Si bien la diversidad de herramientas CASE es bastante marcada, las empresas fabricantes están mostrando varias tendencias fundamentales de integración, a saber.

En el mercado actualmente existe una gama amplia de herramientas para construcción de software entre bases de datos, lenguajes de programación y herramientas de modelamiento, la elección de cuál es la que se aplica en el Sistema Unificado de Proveedores SUP depende de los siguientes factores:

Licencia que posee el BID para uso de motor de base de datos.

Presupuesto para adquirir licencias de uso de paquetes de programación, tomando en cuenta que el BID no es una empresa dedicada a desarrollo de software se responsabiliza a quien construya el Sistema, la adquisición de las mismas.

Entre los diferentes tipos de base de datos, podemos encontrar los siguientes:

MySql: es una base de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez. No es recomendable usar para grandes volúmenes de datos.

PostgreSql y Oracle: Son sistemas de base de datos poderosos. Administra muy bien grandes cantidades de datos, y suelen ser utilizadas en intranets y sistemas de gran calibre.

Access: Es una base de datos desarrollada por Microsoft. Esta base de datos, debe ser creada bajo el programa access, el cual crea un archivo .mdb con la estructura ya explicada.

Microsoft SQL Server: es una base de datos más potente que access desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de informaciones

El BID posee licencia para el uso de Microsoft SQL Server 2000 por ende este motor de base de datos se utilizará para el desarrollo de la Base de Datos de SUP.

Al igual que existe una gama amplia de bases de datos también en lenguajes de programación se dispone de un listado extenso como el siguiente:

- ✓ ActionScript
- ✓ Ada
- ✓ C++
- ✓ C#
- ✓ VB.NET

- ✓ Visual FoxPro
- ✓ Clarion
- ✓ Delphi
- ✓ Harbour
- ✓ Eiffel
- ✓ Java
- ✓ JavaScript
- ✓ Lexico (en castellano)
- ✓ Objective-C
- ✓ Ocaml
- ✓ Oz
- ✓ Perl (soporta herencia múltiple)
- ✓ PHP (en su versión 5)
- ✓ Python
- ✓ Ruby
- ✓ Smalltalk
- ✓ Magik (SmallWorld)

Para la construcción de SUP se requiere un lenguaje de programación de cuarta generación orientado a objetos, por experiencia y dominio en programación el lenguaje en el cual será construido SUP es Microsoft Visual Studio Paquete Visual Basic 6.0.

El software comercial de UML también tiene variedad para modelar el Sistema SUP, siendo los principales la lista siguiente.

- ✓ Enterprise Architect de Sparx Systems
- ✓ Borland Together
- ✓ Corel iGrafx
- ✓ Microsoft Visio
- ✓ PowerDesigner de Sybase
- ✓ Rational Rose de IBM
- ✓ Poseidon for UML de GentleWare
- ✓ MagicDraw UML

Las herramientas de modelamiento que se ajustan a los requerimientos que tiene SUP será Rational Rose y Power Designer.

SUP será creado siguiendo el paradigma en Espiral y para cumplir sus ciclos se utilizará Lenguaje de Programación Visual Basic 6.0 que se conectará a una base de datos en Microsoft SQL Server 2000 y todos sus diagramas de modelamiento UML se construirán en Rational Rose 2000 y Power Designer v9. Basicamente por motivo de manejo de licencias y experiencia en la utilización de estas herramientas.

Debido a que el Sistema Unificado de Proveedores SUP no es un sistema de complejidad muy grande y por un limitado presupuesto para su desarrollo, las herramientas CASE a utilizarse en este sistema se describen a continuación.

2.4.3 RATIONAL ROSE 2000

RATIONAL ROSE es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto:

- Concepción y formalización del modelo,
- Construcción de los componentes,
- Transición a los usuarios y
- Certificación de las distintas fases

La interfaz de Rational Rose está formada por los siguientes elementos principales:

1. Browser ó Navegador, que permite navegar rápidamente a través de las distintas vistas del modelo
2. Ventana de documentación, para manejar los documentos del ítem seleccionado en cualquiera de los diagramas.

3. Barra de herramientas Standard, para acceder rápidamente a las acciones comunes a ejecutar para cada uno de los diagramas del modelo.
4. Barra de herramientas Diagrama, muestra el conjunto de herramientas disponibles para el diagrama activo.
5. Ventana de Diagrama, que permite desplegar y editar cualquiera de los diagramas UML
6. Ventana Registro o Log, que registra todas las órdenes ejecutadas y los errores que se producen durante su ejecución.
7. Barra de Estado, que muestra el programa de la carga del modelo, el estado de lectura/escritura del elemento seleccionado, y otros datos de utilidad.

2.4.4 SQL SERVER 2000

SQL Server 2000 es un potente motor de bases de datos de alto rendimiento capaz de soportar millones de registros por tabla con un interface intuitivo y con herramientas de desarrollo integradas como Visual Studio 6.0 o .NET, además incorpora un modelo de objetos totalmente programable (SQL-DMO) con el que podemos desarrollar cualquier aplicación que manipule componentes de SQL Server, es decir, hacer aplicación para crear bases de datos, tablas, DTS, backups, etc., todo lo que se puede hacer desde el administrador del SQL Server y podemos hacerlo no solo en Visual C++ sino también en Visual Basic, ASP y .NET.

Es una base de datos moderna con una arquitectura implantada completada por Microsoft, y diseñada para las direcciones más exigentes de aplicaciones de base de datos requeridas por decisiones de apoyo operacionales para sistemas implantados hoy y en el futuro. **SQL Server 7.0** y MSDE 1.0 soporta todos los 32-bit de las plataformas de Microsoft Windows (excepto Win32), códigos base compatibles con adaptaciones dinámicas del hardware capaces de soportar un sistema en el cual ya esté instalado. Esto significa que los desarrolladores pueden escribir un conjunto sencillo de aplicaciones con códigos fuente y mandarlo a las bases de datos de un Windows 98, de un Windows NT **Server** y de un Windows

NT **Server** Enterprise Edition. Las filas de Base de Datos también son compatibles con todas las ediciones de **SQL Server 7.0**.

Una versión especial y variante del **SQL Server 7.0**, edición Estándar, es la llamada **SQL Server Developer's Edition**, (edición Promotora), compatible con Microsoft Visual Studio y autorizado aplicaciones desarrolladas que utilizan solamente Visual Studio.

SQL SERVER facilita la creación y la invocación de procedimientos o búsquedas, ya que trabaja con la arquitectura cliente-servidor.

La asignación dinámica de recursos permite la escalabilidad del uso del disco y memoria para acomodarse a las necesidades de la base de datos en cada momento. Esta flexibilidad permite un mejor rendimiento y simplifica la administración del software. La eliminación de dispositivos también es una ventaja complementaria.

2.4.5 VISUAL BASIC

Visual Basic es un lenguaje de programación de los llamados "visuales", puesto que parte de la programación que se realiza con él se basa en la utilización de elementos visuales. La palabra "Visual" hace referencia al método que se utiliza para crear la interfaz gráfica de usuario. En lugar de escribir numerosas líneas de código para describir la apariencia y la ubicación de los elementos de la interfaz, simplemente podemos agregar objetos prefabricados en su lugar dentro de la pantalla, que ahorran mucho tiempo de programación y que sobrepasan el concepto de la programación convencional en cuanto a la estructura secuencial del programa.

Tradicionalmente el lenguaje Basic (Beginners All-Purpose Symbolic Instruction Code) se ha considerado como el lenguaje adecuado para principiantes en el mundo de la programación, si bien se le achaca que no tiene la potencia suficiente para cubrir todas las expectativas de los programadores avanzados. Ciertamente que aunque nos permite la creación de componentes y trabajar con objetos, otros lenguajes tienen más potencia en la programación orientada a

objetos, pero a medida que han avanzado en las versiones de Visual Basic, ha aumentado su versatilidad.

Visual Basic es un lenguaje orientado a eventos. En las aplicaciones tradicionales o "por procedimientos", la aplicación es la que controla qué partes de código y en qué secuencia se ejecutan. La ejecución comienza con la primera línea de código y continúa con una ruta predefinida a través de la aplicación, llamando a los procedimientos según se necesiten.

En una aplicación controlada por eventos, el código no sigue una ruta predeterminada; ejecuta distintas secciones de código como respuesta a los eventos. Los eventos pueden desencadenarse por acciones del usuario, por mensajes del sistema o de otras aplicaciones, o incluso por la propia aplicación. La secuencia de estos eventos determina la secuencia en la que se ejecuta el código, por lo que la ruta a través del código de la aplicación es diferente cada vez que se ejecuta el programa.

Para manejar los objetos y controles del lenguaje hay que conocer tres conceptos asociados a ellos:

Propiedades: Para todos los objetos y controles de Visual Basic, hay definidas una serie de propiedades que nos permiten controlarlos: nombre, tamaño, color, posición, etc..., la mayoría de las propiedades serán comunes, pero evidentemente, habrá objetos que tengan unas determinadas propiedades específicas.

Eventos: Cada objeto lleva asociados unos determinados eventos que le pueden ocurrir, por ejemplo a un botón, le puede ocurrir que el usuario pulse con el botón principal del ratón sobre él, eso es el evento `NombreDelBotón_Click()`, bien cuando esto ocurra, se dispara este evento, al cual nosotros le añadiremos el código de lo que queremos que haga la aplicación cuando el usuario pulse el botón.

Métodos: Son procedimientos asociados a un objeto, es decir, nos permiten realizar acciones como por ejemplo escribir un texto en un objeto, necesitaríamos

usar el método Print, con la siguiente sintaxis: NombreDelObjeto.Print "Texto deseado".

Encontraremos tanto propiedades, eventos como métodos comunes, si bien, dependiendo de los objetos que utilicemos encontraremos elementos específicos y exclusivos del objeto.

El esquema de programación es muy sencillo, para crear una aplicación, primero crearemos un formulario vacío sobre el que iremos poniendo los controles que deseemos, en tiempo de diseño podremos especificar las propiedades iniciales de esos controles. Posteriormente escribiremos el "código respuesta" a los eventos que queramos controlar de nuestros objetos. En ese "código respuesta", que es lo que llamaremos tiempo de ejecución, podremos modificar algunas propiedades de los objetos, lanzar métodos, ejecutar rutinas, llamar a otros procedimientos etc..

Visual Basic se encuentra disponible en tres versiones, cada una de las cuales está orientada a unos requisitos de programación específicos.

La Edición de Aprendizaje de Visual Basic permite a los programadores crear robustas aplicaciones para Microsoft Windows y Windows NT®. Incluye todos los controles intrínsecos, además de los controles de cuadrícula, de fichas y los controles enlazados a datos.

La Edición Profesional proporciona a los profesionales un conjunto completo de herramientas para desarrollar soluciones para terceros. Incluye todas las características de la Edición de Aprendizaje, así como controles ActiveX adicionales, el diseñador de aplicaciones para Internet Information Server y Visual Database Tools and Data.

La Edición Empresarial permite a los profesionales crear sólidas aplicaciones distribuidas en un entorno de equipo. Incluye todas las características de la Edición Profesional, así como herramientas de Back Office como SQL Server, Microsoft Transaction Server, Internet Information Server, Visual SourceSafe, SNA Server, etc.

2.4.6 CLIENTE SERVIDOR

Entre las principales definiciones se tiene¹¹:

Desde un punto de vista conceptual:

«Es un modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información»

En términos de arquitectura:

Los distintos aspectos que caracterizan a una aplicación (proceso, almacenamiento, control y operaciones de entrada y salida de datos) en el sentido más amplio, están situados en más de un computador, los cuales se encuentran interconectados mediante una red de comunicaciones»

IBM define al modelo Cliente/Servidor

«Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores".

Definición del Modelo Cliente/Servidor¹²

La tecnología cliente/servidor es el procesamiento cooperativo de la información por medio de un conjunto de procesadores, en el cual múltiples clientes, distribuidos geográficamente, solicitan requerimientos a uno o más servidores centrales. Desde el punto de vista funcional, se puede definir la computación cliente/servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aun en entornos multiplataforma.

¹¹ Fuente: <http://www.inei.gob.pe/biblioineipub/bancopub/inf/lib5038/defi.htm>

¹² Fuente: http://ar.geocities.com/r_niella/Document/t_cap1.htm

En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor, y este envía uno o varios mensajes con la respuesta. En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras. Además como veremos en el modelo de implementación, el concepto es utilizado en forma constante para varias funciones e implementado de distintas formas.

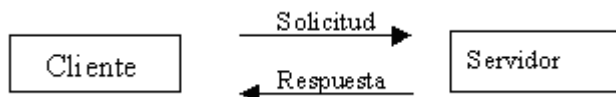


Figura 2.30 Modelo Cliente Servidor

La idea es tratar a una computadora como un instrumento, que por sí sola pueda realizar muchas tareas, pero con la consideración de que realice aquellas que son más adecuadas a sus características. Si esto se aplica tanto a clientes como servidores se entiende que la forma más estándar de aplicación y uso de sistemas clientes/servidores es mediante la explotación de las PC a través de interfaces gráficas de usuario; mientras que la administración de datos y su seguridad e integridad se deja a cargo de computadoras centrales tipo mainframe. Como se desprende de las definiciones anteriores, tanto clientes como servidores son entidades independientes que operan conjuntamente a través de una red para realizar una tarea. Pero para hacer la distinción respecto de otras formas de arquitecturas o software distribuidos, se presenta una lista de características que debieran cumplir los sistemas cliente/servidor:

- ✓ Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- ✓ Existe una clara distinción de funciones basada en el concepto de "servicio", que se establece entre clientes y servidores.
- ✓ La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.

- ✓ Los clientes corresponden a procesos activos en cuanto a que son éstos lo que hacen peticiones de servicios a los servidores. Estos últimos tienen un carácter pasivo ya que esperan las peticiones de los clientes.
- ✓ No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio.
- ✓ Las plataformas de software y hardware entre clientes y servidores son independientes. Precisamente una de las principales ventajas de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.
- ✓ El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema cliente/servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

CAPÍTULO III

SISTEMA UNIFICADO DE PROVEEDORES SUP

3.1 INVESTIGACIÓN PRELIMINAR

3.1.1 ESPECIFICACIÓN DE REQUERIMIENTOS

3.1.1.1 Descripción de Actores

El Sistema Unificado de Proveedores SUP requiere dos agentes externos para su manejo que son:

3.1.1.1.1 Administrador

El Administrador es un actor que basará su interacción con SUP para hacer la conexión a la base de datos y tendrá la responsabilidad de administrar los perfiles (accesos a los módulos) y la creación de usuarios (personas que tendrán accesos a SUP con los perfiles antes creados según políticas y necesidades del BID Representación Ecuador).

3.1.1.1.2 Usuario

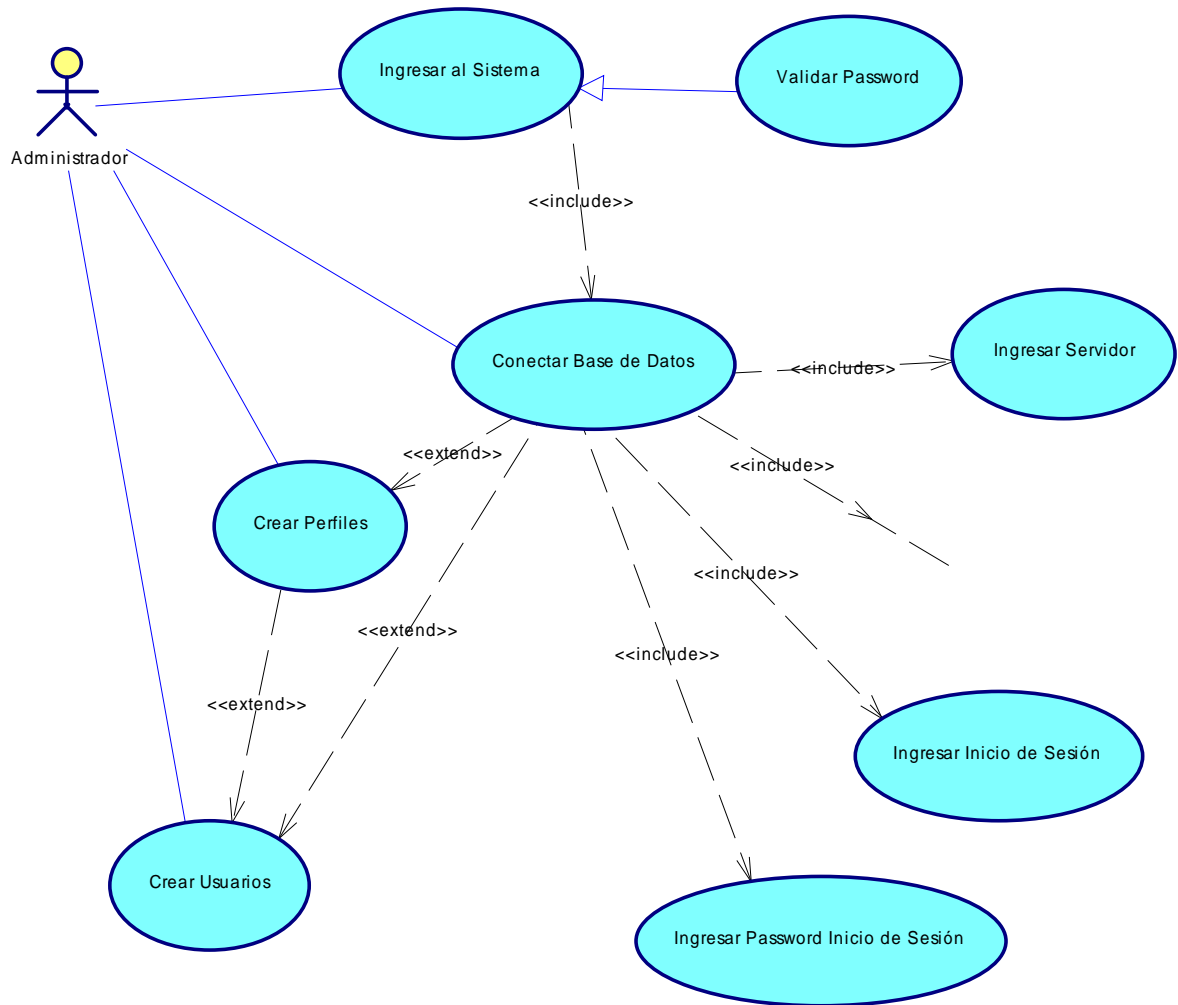
El Usuario o usuarios serán actores que tendrán accesos determinados por el administrador para poder interactuar con SUP, basados en la actividad que realizaran dentro del Sistema.

3.1.1.2 Diagrama de Casos de Uso

3.1.1.2.1 Actividad Actor Administrador

SUP será un sistema que necesita verificar la identidad de una persona para que pueda acceder al sistema y hacer la conexión a la base de datos creando un cadena de conexión, ingresando el servidor, nombre de la base de datos, inicio de sesión y password de ese inicio de sesión, cuando ya se haya hecho dicha conexión, esta persona debe crear perfiles (accesos a los módulos que posee el sistema) y además debe crear usuarios (personas a las cuales se asigna los perfiles creados).

Object-Oriented Model	
Model: Diagrama Casos de Uso Administrador	
Package:	
Diagram: UseCaseDiagram_1	
Author: Felipe Barrera	Date : 10/06/2008
Version : 1.0	



3.1.1.2.2 Actividad Actor Usuario

Cuando ya estén creados el usuario o los usuarios de SUP, éste debe ingresar al Sistema, mismo que debe validar el nombre de usuario y su password para dejarle ingresar, este usuario estará en la capacidad de ingresar información a la base de datos, ya sea de Funcionarios, Proveedores o Proyectos en ejecución en la Representación de Ecuador del BID, toda esta información ingresada puede ser modificada por el usuario, además el usuario podrá ingresar la información de la familia, estudios y bancaria de los Funcionarios del BID para hacer transferencias interbancarias generando un archivo .prn, pero antes de generar este archivo el usuario debe tener la facilidad de verificar la información mediante una consulta de transferencia y poderla imprimir. Toda la información referente a los funcionarios del BID Representación Ecuador también debe ser consultada y los resultados de la consulta debe ser exportada a archivos .xls para su posterior tratamiento, al igual que las transferencias interbancarias de funcionarios, el usuario debe poder generar otro archivo .prn para transferencias interbancarias de proveedores, los préstamos en ejecución deben ser asignados a tres funcionarios con cargo de Especialistas del BID Representación Ecuador que son Sectoriales Principal y Alterno y un Especialista Financiero, el usuario también debe generar un reporte de los préstamos a cargo de un Especialista ya sea principal, alterno o financiero e imprimirlo. Se requiere poder generar reportes de la Edad de los funcionarios, tiempo de servicio en el BID y Representación Ecuador y un reporte de la edad de los hijos de los funcionarios por intervalos de edades que deben ser variables.

3.1.1.3 Diccionario de Casos de Uso

3.1.1.3 .1 Actor Administrador

Name	Code
Ingresar al Sistema	Ingresar_al_Sistema
Validar Password	Validar_Password
Conectar Base de Datos	Conectar_Base_de_Datos
Crear Perfiles	Crear_Perfiles
Crear Usuarios	Crear_Usuarios
Ingresar Servidor	Ingresar_Servidor
Ingresar Nombre Base de Datos	Ingresar_Nombre_Base_de_Datos
Ingresar Inicio de Sesión	Ingresar_Inicio_de_Sesion
Ingresar Password Inicio de Sesión	Ingresar_Password_Inicio_de_Sesion

3.1.1.3 .2 Actor Usuario

Name	Code
Ingresar al Sistema	Ingresar_al_Sistema
Validar Usuario	Validar_Usuario
Comprobar Nombre de Usuario y Password	Comprobar_Nombre_de_Usuario_y_Password
Ingresar Información	Ingresar_Informacion
Modificar Información	Modificar_Informacion
Generar Consulta	Generar_Consulta
Exportar Datos Consulta xls	Exportar_Datos_Consulta_xls
Ingresar Funcionario	Ingresar_Funcionario
Ingresar Familia	Ingresar_Familia
Ingresar Estudios	Ingresar_Estudios
Realizar Consulta de Transferencia	Realizar_Consulta_de_Transferencia
Filtrar Especialistas por Area	Filtrar_Especialistas_por_Area
Asignar Especialista a Operación	Asignar_Especialista_a_Operacion
Ingresar Datos para transferencia interbancaria	Ingresar_Datos_para_transferencia_interbancaria
Ingresar Información Bancaria	Ingresar_Informacion_Bancaria
Ingresar Proveedor	Ingresar_Proveedor
Ingresar Préstamo	Ingresar_Prestamo
Generar Archivo prn	Generar_Archivo_prn
Asignar Esp Sectorial	Asignar_Esp_Sectorial
Generar Reporte de transferencia	Generar_Reporte_de_transferencia
Imprimir Reporte	Imprimir_Reporte
Asignar Esp Alterno	Asignar_Esp_Alterno
Determinar Cargo	Determinar_Cargo
Asignar Esp Financiero	Asignar_Esp_Financiero
Consultar Operaciones	Consultar_Operaciones
Generar Reporte	Generar_Reporte
Calcular Años de Servicio	Calcular_Anos_de_Servicio
Calcular Edad Hijos	Calcular_Edad_Hijos
Calcular Edad Funcionario	Calcular_Edad_Funcionario

3.2 ANALISIS Y DISEÑO

3.2.1 MODELO ESTÁTICO

3.2.1.1 Diagrama de Clases

3.2.1.2 Diccionario de Clases

Entity list

Name	Code	Generate
Area	AREA	TRUE
AreaOcupacion	AREAOCUPACION	TRUE
Asignaciones	ASIGNACIONES	TRUE
BancoBeneficiario	BANCOBENEFICIARIO	TRUE
Beneficiario	BENEFICIARIO	TRUE
BID	BID	TRUE
Busqueda	BUSQUEDA	TRUE
Cargo	CARGO	TRUE
Clase	CLASE	TRUE
Cluster	CLUSTER	TRUE
Consulta	CONSULTA	TRUE
ConsultaProv	CONSULTAPROV	TRUE
ConsultaProy	CONSULTAPROY	TRUE
Contrato	CONTRATO	TRUE
CuentasBeneficiariosBancoInternacio nal	CUENTASBENEFICIARIOSBANCOINTE RNACIONAL	TRUE
CuentasBeneficiariosSIP	CUENTASBENEFICIARIOSSIP	TRUE
CuentasProveedoresBancoInternacion al	CUENTASPROVEEDORESBANCOINTER NACIONAL	TRUE
CuentasProveedoresSIP	CUENTASPROVEEDORESSIP	TRUE
Educacion	EDUCACION	TRUE
Especialista	ESPECIALISTA	TRUE
EstadoCivil	ESTADOCIVIL	TRUE
EstadoOperacion	ESTADOOOPERACION	TRUE
Familia	FAMILIA	TRUE
Funcionario	FUNCIONARIO	TRUE
Funcionario_SIP	FUNCIONARIO_SIP	TRUE
FuncionarioEdad	FUNCIONARIOEDAD	TRUE
FuncionarioIdioma	FUNCIONARIOIDIOMA	TRUE
HijoEdad	HIJOEDAD	TRUE
Idioma	IDIOMA	TRUE
NivelEducacion	NIVELEDUCACION	TRUE
NominaBancoInternacional	NOMINABANCOINTERNACIONAL	TRUE
NominaSIP	NOMINASIP	TRUE
NominaUnificada	NOMINAUNIFICADA	TRUE
Pais	PAIS	TRUE
Perfiles	PERFILES	TRUE
Prestamo	PRESTAMO	TRUE
Proveedor	PROVEEDOR	TRUE
Proyecto	PROYECTO	TRUE
Sexo	SEXO	TRUE
TiempoServicioBIDCOF	TIEMPOSERVICOBIDCOF	TRUE
TipoActividad	TIPOACTIVIDAD	TRUE
TipoCuenta	TIPOCUENTA	TRUE
TipoEspecialista	TIPOESPECIALISTA	TRUE
TipoOperacion	TIPOOPERACION	TRUE
TransferenciaBancoInternacional	TRANSFERENCIABANCOINTERNACIO	TRUE

	NAL	
TransferenciaSIP	TRANSFERENCIASIP	TRUE
TransferenciaUnificada	TRANSFERENCIAUNIFICADA	TRUE
Unidad	UNIDAD	TRUE
Usuario	USUARIO	TRUE

Identifier list of entity

Name	Code	Parent
PK_Area	PK_AREA	Entity 'Area'
PK_AreaOcupacion	PK_AREAOCUPACION	Entity 'AreaOcupacion'
PK_BancoBeneficiario	PK_BANCOBENEFICIARIO	Entity 'BancoBeneficiario'
PK_Beneficiario	PK_BENEFICIARIO	Entity 'Beneficiario'
PK_BID	PK_BID	Entity 'BID'
PK_Cargo	PK_CARGO	Entity 'Cargo'
PK_Clase	PK_CLASE	Entity 'Clase'
PK_Cluster	PK_CLUSTER	Entity 'Cluster'
PK_Contrato	PK_CONTRATO	Entity 'Contrato'
PK_CuentasBID	PK_CUENTASBID	Entity 'CuentasBeneficiariosBancoInternacional'
PK_CuentasBeneficiarios	PK_CUENTASBENEFICIARIOS	Entity 'CuentasBeneficiariosSIP'
PK_CuentasProveedoresBancoInternacional	PK_CUENTASPROVEEDORES SBANCOINTERNACIONAL	Entity 'CuentasProveedoresBancoInternacional'
PK_CuentasProveedoresSIP	PK_CUENTASPROVEEDORES SSIP	Entity 'CuentasProveedoresSIP'
PK_Educacion	PK_EDUCACION	Entity 'Educacion'
PK_Especialista	PK_ESPECIALISTA	Entity 'Especialista'
PK_EstadoCivil	PK_ESTADOCIVIL	Entity 'EstadoCivil'
PK_EstadoOperacion	PK_ESTADOOPERACION	Entity 'EstadoOperacion'
PK_Familia	PK_FAMILIA	Entity 'Familia'
PK_Funcionario	PK_FUNCIONARIO	Entity 'Funcionario'
PK_Funcionario_SIP	PK_FUNCIONARIO_SIP	Entity 'Funcionario_SIP'
PK_Idioma	PK_IDIOMA	Entity 'Idioma'
PK_NivelEducacion	PK_NIVELEDUCACION	Entity 'NivelEducacion'
PK_NominaBancoInternacional	PK_NOMINABANCOINTERNACIONAL	Entity 'NominaBancoInternacional'
PK_NominaSIP	PK_NOMINASIP	Entity 'NominaSIP'
PK_Nomina	PK_NOMINA	Entity 'NominaUnificada'
PK_Pais	PK_PAIS	Entity 'Pais'
PK_Prestamo	PK_PRESTAMO	Entity 'Prestamo'
PK_Proveedor	PK_PROVEEDOR	Entity 'Proveedor'
PK_Proyecto	PK_PROYECTO	Entity 'Proyecto'
PK_Sexo	PK_SEXO	Entity 'Sexo'
PK_TipoActividad	PK_TIPOACTIVIDAD	Entity 'TipoActividad'
PK_TipoCuenta	PK_TIPOCUENTA	Entity 'TipoCuenta'
PK_TipoEspecialista	PK_TIPOESPECIALISTA	Entity 'TipoEspecialista'

PK_TipoOperacion	PK_TIPOOPERACION	Entity 'TipoOperacion'
PK_TransferenciaBancoInternacional	PK_TRANSFERENCIABANCOINTERNACIONAL	Entity 'TransferenciaBancoInternacional'
PK_TransferenciaSIP	PK_TRANSFERENCIASIP	Entity 'TransferenciaSIP'
PK_Unidad	PK_UNIDAD	Entity 'Unidad'

Data item list

Name	Code	Domain	Data Type	Length
Area_Id	AREA_ID	<None>	SI	
Area_Nombre	AREA_NOMBRE	<None>	A15	15
AreaOcupacion_ID	AREAOCUPACION_ID	<None>	SI	
AreaOcupacion_Nombre	AREAOCUPACION_NOMBRE	<None>	A15	15
BancoBeneficiario_ID	BANCOBENEFICIARIO_ID	<None>	VMBT8	8
BancoBeneficiario_Descripcion	BANCOBENEFICIARIO_DESCRIPCION	<None>	VMBT30	30
Beneficiario_ID	BENEFICIARIO_ID	<None>	VMBT6	6
Beneficiario_Nombre	BENEFICIARIO_NOMBRE	<None>	A30	30
Beneficiario_Identificacion	BENEFICIARIO_IDENTIFICACION	<None>	VMBT13	13
Beneficiario_Pago	BENEFICIARIO_PAGO	<None>	SI	
BID_CodigoEmpresa	BID_CODIGOEMPRESA	<None>	VMBT4	4
BID_CuentaDebito	BID_CUENTADEBITO	<None>	VMBT9	9
Busqueda_FuncionarioId	BUSQUEDA_FUNCIONARIOID	<None>	VMBT6	6
Busqueda_FuncionarioNombre	BUSQUEDA_FUNCIONARIO_NOMBRE	<None>	A30	30
Cargo_Id	CARGO_ID	<None>	SI	
Cargo_Nombre	CARGO_NOMBRE	<None>	A22	22
Clase_ID	CLASE_ID	<None>	SI	
Clase_Descripcion	CLASE_DESCRIPCION	<None>	A26	26
Cluster_ID	CLUSTER_ID	<None>	SI	
Cluster_Descripcion	CLUSTER_DESCRIPCION	<None>	A40	40
Beneficiario	BENEFICIARIO	<None>	A30	30
Empleado	EMPLEADO	<None>	VMBT6	6
Fecha	FECHA	<None>	VMBT8	8
Monto	MONTO	<None>	N18	18
Empresa	EMPRESA	<None>	A25	25
Representante	REPRESENTANTE	<None>	A35	35
NumeroFactura	NUMEROFACTURA	<None>	VMBT15	15
Cantidad	CANTIDAD	<None>	N18	18
FechaPago	FECHAPAGO	<None>	A8	8
Especialista	ESPECIALISTA	<None>	A40	40
Proyecto	PROYECTO	<None>	A10	10
Descripción	DESCRIPCION	<None>	A50	50

Préstamo	PRESTAMO	<None>	A15	15
Ejecutor	EJECUTOR	<None>	A20	20
Contrato_Id	CONTRATO_ID	<None>	SI	
Contrato_Nombre	CONTRATO_NOMBRE	<None>	A15	15
CuentasBeneficiariosBancoInternacional_ID	CUENTASBENEFICIARIOSBANCOINTERNACIONAL_ID	<None>	VMBT10	10
CuentasBeneficiarios_ID	CUENTASBENEFICIARIOS_ID	<None>	VMBT20	20
CuentasProveedorBancoInternacional_ID	CUENTASPROVEEDORBANCOINTERNACIONAL_ID	<None>	VMBT9	9
CuentasProveedoresSIP_ID	CUENTASPROVEEDORESSIP_ID	<None>	VMBT20	20
Educacion_Id	EDUCACION_ID	<None>	I	
Educacion_Universidad	EDUCACION_UNIVERSIDAD	<None>	A50	50
Educacion_Titulo	EDUCACION_TITULO	<None>	A50	50
Educacion_Anio	EDUCACION_ANIO	<None>	A4	4
Especialista_Id	ESPECIALISTA_ID	<None>	VMBT6	6
Especialista_Nombre	ESPECIALISTA_NOMBRE	<None>	A40	40
EstadoCivil_Id	ESTADOCIVIL_ID	<None>	SI	
EstadoCivil_Nombre	ESTADOCIVIL_NOMBRE	<None>	A10	10
EstadoOperacion_Id	ESTADOOPERACION_ID	<None>	VMBT2	2
EstadoOperacion_Descripcion	ESTADOOPERACION_DESCRIPCION	<None>	VMBT30	30
Familia_Id	FAMILIA_ID	<None>	I	
Familia_Parentesco	FAMILIA_PARENTESCO	<None>	A10	10
Familia_Nombre	FAMILIA_NOMBRE	<None>	A25	25
Familia_Apellido	FAMILIA_APELLIDO	<None>	A25	25
Familia_FechaNacimiento	FAMILIA_FECHANACIMIENTO	<None>	DT	
Funcionario_Id	FUNCIONARIO_ID	<None>	VMBT6	6
Funcionario_CI	FUNCIONARIO_CI	<None>	VMBT10	10
Funcionario_Nombre1	FUNCIONARIO_NOMBRE1	<None>	A15	15
Funcionario_Nombre2	FUNCIONARIO_NOMBRE2	<None>	A15	15
Funcionario_Apellido1	FUNCIONARIO_APELLIDO1	<None>	A15	15
Funcionario_Apellido2	FUNCIONARIO_APELLIDO2	<None>	A15	15
Funcionario_Foto	FUNCIONARIO_FOTO	<None>	PIC	
Funcionario_Email	FUNCIONARIO_EMAIL	<None>	A20	20
Funcionario_CallePrincipal	FUNCIONARIO_CALLEPRINCIPAL	<None>	A30	30
Funcionario_CalleTransversal	FUNCIONARIO_CALLETRANSVERSAL	<None>	A30	30

Funcionario_CasaNumber	FUNCIONARIO_CASA NUMBER	<None>	A10	10
Funcionario_Barrío	FUNCIONARIO_BARRIO	<None>	A30	30
Funcionario_Telefono Convencional	FUNCIONARIO_TELEFONO CONVENCIONAL	<None>	A10	10
Funcionario_Telefono Celular	FUNCIONARIO_TELEFONO CELULAR	<None>	A10	10
Funcionario_Telefono Conyuge	FUNCIONARIO_TELEFONO CONYUGE	<None>	A10	10
Funcionario_Oficina	FUNCIONARIO_OFICINA	<None>	A10	10
Funcionario_FechaIngreso BID	FUNCIONARIO_FECHA INGRESOBID	<None>	DT	
Funcionario_FechaIngreso COF	FUNCIONARIO_FECHA INGRESOCOF	<None>	DT	
Funcionario_FechaFin Contrato	FUNCIONARIO_FECHA FINCONTRATO	<None>	DT	
Funcionario_Transferencia Servicio	FUNCIONARIO_TRANSFERENCIA SERVICIO	<None>	DT	
Funcionario_FechaNacimiento	FUNCIONARIO_FECHA NACIMIENTO	<None>	DT	
Funcionario_Extensión	FUNCIONARIO_EXTENSION	<None>	A4	4
Funcionario_Parqueadero	FUNCIONARIO_PARQUEADERO	<None>	A10	10
Funcionario_CelularBID	FUNCIONARIO_CELULARBID	<None>	A10	10
Funcionario_TipoTelefono	FUNCIONARIO_TIPOTELEFONO	<None>	A10	10
Funcionario_ID	FUNCIONARIO_ID	<None>	VMBT6	6
Funcionario_Nombre	FUNCIONARIO_NOMBRE	<None>	A30	30
Funcionario_Id	FUNCIONARIO_ID	<None>	VMBT6	6
Funcionario_Nombre	FUNCIONARIO_NOMBRE	<None>	A30	30
Funcionario_Edad	FUNCIONARIO_EDAD	<None>	I	
Apellido	APELLIDO	<None>	A25	25
Nombre	NOMBRE	<None>	A25	25
FechaNacimiento	FECHANACIMIENTO	<None>	DT	
Edad	EDAD	<None>	I	
Pais	PAIS	<None>	A25	25
Idioma_Id	IDIOMA_ID	<None>	SI	
Idioma_Nombre	IDIOMA_NOMBRE	<None>	A10	10
NivelEducacion_Id	NIVEL EDUCACION_ID	<None>	SI	
Nivel_Educacion_Descripcion	NIVEL_EDUCACION_DESCRIPCION	<None>	A15	15
NominaBancoInternacional_ID	NOMINABANCOINTERNACIONAL_ID	<None>	NO	
NominaBancoInternacional_Filler	NOMINABANCOINTERNACIONAL_FILLER	<None>	VMBT1	1
NominaUnificada_Val	NOMINAUNIFICADA_	<None>	VMBT13	13

orCredito	VALORCREDITO			
NominaBancoInternacional_FechaPago	NOMINABANCOINTERNACIONAL_FECHAPAGO	<None>	VMBT8	8
NominaSIP_ID	NOMINASIP_ID	<None>	NO	
NominaSIP_FechaPago	NOMINASIP_FECHAPAGO	<None>	VMBT8	8
NominaSIP_ConceptoPago	NOMINASIP_CONCEPTOPAGO	<None>	VMBT2	2
NominaUnificada_ValorCredito	NOMINAUNIFICADA_VALORCREDITO	<None>	VMBT13	13
Nomina_ID	NOMINA_ID	<None>	NO	
TipoCuenta_ID	TIPOCUENTA_ID	<None>	VMBT2	2
NumeroCuenta	NUMEROCUENTA	<None>	VMBT10	10
NominaUnificada_ValorCredito	NOMINAUNIFICADA_VALORCREDITO	<None>	VMBT13	13
Beneficiario_ID	BENEFICIARIO_ID	<None>	VMBT6	6
NominaUnificada_Filler	NOMINAUNIFICADA_FILLER	<None>	VMBT1	1
NominaUnificada_FechaPago	NOMINAUNIFICADA_FECHAPAGO	<None>	VMBT8	8
CuentaBeneficiaria	CUENTABENEFICIARIA	<None>	VMBT20	20
TipoCuentaBeneficiaria	TIPOCUENTABENEFICIARIA	<None>	VMBT2	2
NombreBeneficiario	NOMBREBENEFICIARIO	<None>	A30	30
IdentificacionBeneficiario	IDENTIFICACIONBENEFICIARIO	<None>	VMBT13	13
Nomina_ConceptoPago	NOMINA_CONCEPTOPAGO	<None>	VMBT2	2
FechaIngreso	FECHAINGRESO	<None>	DT	
NominaUnificada_Quincena	NOMINAUNIFICADA_QUINCENA	<None>	A7	7
NominaUnificada_Meses	NOMINAUNIFICADA_MES	<None>	A12	12
NominaUnificada_Año	NOMINAUNIFICADA_AÑO	<None>	A4	4
Pais_Id	PAIS_ID	<None>	SI	
Pais_Nombre	PAIS_NOMBRE	<None>	A10	10
Nombre	NOMBRE	<None>	A25	25
accesos	ACCESOS	<None>	I	
Prestamo_Id	PRESTAMO_ID	<None>	A15	15
Prestamo_Ejecutor	PRESTAMO_EJECUTOR	<None>	A20	20
Proveedor_RUC	PROVEEDOR_RUC	<None>	VMBT15	15
Proveedor_Nombre	PROVEEDOR_NOMBRE	<None>	A35	35
Proveedor_Empresa	PROVEEDOR_EMPRESA	<None>	A25	25
Proveedor_Direccion	PROVEEDOR_DIRECCION	<None>	A50	50
Proveedor_Telefono	PROVEEDOR_TELEFONO	<None>	A10	10

	NO			
Proveedor_NumeroReferencia	PROVEEDOR_NUMEROREFERENCIA	<None>	VMBT9	9
Proveedor_Email	PROVEEDOR_EMAIL	<None>	A35	35
Proveedor_Celular	PROVEEDOR_CELULAR	<None>	A10	10
Proveedor_Servicios	PROVEEDOR_SERVICIOS	<None>	A50	50
Proveedor_Web	PROVEEDOR_WEB	<None>	A25	25
Proveedor_Fax	PROVEEDOR_FAX	<None>	A10	10
Proyecto_Id	PROYECTO_ID	<None>	A10	10
Proyecto_Descripcion	PROYECTO_DESCRIPCION	<None>	A50	50
Sexo_Id	SEXO_ID	<None>	SI	
Sexo_Nombre	SEXO_NOMBRE	<None>	A10	10
Funcionario_Id	FUNCIONARIO_ID	<None>	VMBT6	6
Funcionario_Nombre	FUNCIONARIO_NOMBRE	<None>	A30	30
TiempoBID	TIEMPOBID	<None>	I	
TiempoCOF	TIEMPOCOF	<None>	I	
TipoActividad_ID	TIPOACTIVIDAD_ID	<None>	SI	
TipoActividad_Nombre	TIPOACTIVIDAD_NOMBRE	<None>	A15	15
TipoActividad_Iniciales	TIPOACTIVIDAD_INICIALES	<None>	A15	15
TipoCuenta_ID	TIPOCUENTA_ID	<None>	VMBT2	2
TipoCuenta_Descripcion	TIPOCUENTA_DESCRIPCION	<None>	A20	20
TipoEspecialista_Id	TIPOESPECIALISTA_ID	<None>	SI	
TipoEspecialista_Descripcion	TIPOESPECIALISTA_DESCRIPCION	<None>	A20	20
TipoOperacion_Id	TIPOOPERACION_ID	<None>	VMBT10	10
Tipo_Operacion_Descripcion	TIPO_OPERACION_DESCRIPCION	<None>	A50	50
TransferenciaBancoInternacional_ID	TRANSFERENCIABANCOINTERNACIONAL_ID	<None>	NO	
TransferenciaBancoInternacional_ValorPagar	TRANSFERENCIABANCOINTERNACIONAL_VALORPAGAR	<None>	VMBT13	13
TransferenciaBancoInternacional_FechaPago	TRANSFERENCIABANCOINTERNACIONAL_FECHAPAGO	<None>	VMBT8	8
TransferenciaBancoInternacional_Factura	TRANSFERENCIABANCOINTERNACIONAL_FACTURA	<None>	VMBT15	15
TransferenciaBancoInternacional_DescripcionPago	TRANSFERENCIABANCOINTERNACIONAL_DESCRIPCIONPAGO	<None>	A30	30
TransferenciaSIP_ID	TRANSFERENCIASIP_ID	<None>	NO	
TransferenciaSIP_Val	TRANSFERENCIASIP_	<None>	VMBT13	13

orPagar	VALORPAGAR			
TransferenciaSIP_FechaPago	TRANSFERENCIASIP_FECHAPAGO	<None>	VMBT8	8
TransferenciaSIP_Factura	TRANSFERENCIASIP_FACTURA	<None>	VMBT15	15
TransferenciaSIP_DescripcionPago	TRANSFERENCIASIP_DESCRIPCIONPAGO	<None>	A30	30
TU_TipoRegistro	TU_TIPOREGISTRO	<None>	A1	1
TU_NumeroTransferencia	TU_NUMEROTRANSFERENCIA	<None>	VMBT4	4
BID_CodigoEmpresa	BID_CODIGOEMPRESA	<None>	VMBT4	4
Proveedor_Nombre	PROVEEDOR_NOMBRE	<None>	A35	35
BID_CuentaDebito	BID_CUENTADEBITO	<None>	VMBT9	9
TU_ValorPagar	TU_VALORPAGAR	<None>	VMBT13	13
TU_ReferenciaPago	TU_REFERENCIAPAGO	<None>	A30	30
CuentasProveedorBancoInternacional_ID	CUENTASPROVEEDORBANCOINTERNACIONAL_ID	<None>	VMBT9	9
Proveedor_Empresa	PROVEEDOR_EMPRESA	<None>	VMBT15	15
Proveedor_Direccion	PROVEEDOR_DIRECCION	<None>	A35	35
Proveedor_Telefono	PROVEEDOR_TELEFONO	<None>	A35	35
Proveedor_NumeroReferencia	PROVEEDOR_NUMEROREFERENCIA	<None>	A9	9
TU_FechaInstruccionAnio	TU_FECHAINSTRUCCIONANIO	<None>	VMBT2	2
TU_FechaInstruccionMes	TU_FECHAINSTRUCCIONMES	<None>	VMBT2	2
TU_FechaInstruccionDia	TU_FECHAINSTRUCCIONDIA	<None>	VMBT2	2
TU_Factura	TU_FACTURA	<None>	VMBT15	15
TU_MontoPagar	TU_MONTOPAGAR	<None>	VMBT13	13
TU_Descripcion	TU_DESCRIPCION	<None>	A30	30
CuentasProveedoresSIP_ID	CUENTASPROVEEDORRESSIP_ID	<None>	VMBT20	20
TipoCuenta_ID	TIPOCUENTA_ID	<None>	VMBT2	2
TU_ConceptoPago	TU_CONCEPTOPAGO	<None>	VMBT2	2
TU_Email	TU_EMAIL	<None>	A100	100
Proveedor_Celular	PROVEEDOR_CELULAR	<None>	A11	11
TU_FechaPago	TU_FECHAPAGO	<None>	VMBT8	8
Unidad_ID	UNIDAD_ID	<None>	SI	
Unidad_Siglas	UNIDAD_SIGLAS	<None>	A10	10
Unidad_Descripcion	UNIDAD_DESCRIPCION	<None>	A50	50
Nombre	NOMBRE	<None>	A30	30
Id	ID	<None>	A15	15
Password	PASSWORD	<None>	A60	60

Perfil_Nombre	PERFIL_NOMBRE	<None>	A25	25
---------------	---------------	--------	-----	----

Relationship list

Name	Code	Entity 2	Entity 1
FK_Asignaciones_Especialista	FK_ASIGNACIONES_ESPECIALISTA	Asignaciones	Especialista
FK_Asignaciones_Especialista1	FK_ASIGNACIONES_ESPECIALISTA1	Asignaciones	Especialista
FK_Asignaciones_Especialista2	FK_ASIGNACIONES_ESPECIALISTA2	Asignaciones	Especialista
FK_Asignaciones_Prestamo	FK_ASIGNACIONES_PRESTAMO	Asignaciones	Prestamo
FK_CuentasBeneficiariosBancoInternacional_BancoBeneficiario	FK_CUENTASBENEFICIARIOSBANCOINTERNACIONAL_BANCOBENEFICIARIO	CuentasBeneficiariosBancoInternacional	BancoBeneficiario
FK_CuentasBeneficiariosBancoInternacional_Beneficiario	FK_CUENTASBENEFICIARIOSBANCOINTERNACIONAL_BENEFICIARIO	CuentasBeneficiariosBancoInternacional	Beneficiario
FK_CuentasBeneficiariosBancoInternacional_TipoCuenta	FK_CUENTASBENEFICIARIOSBANCOINTERNACIONAL_TIPOCUENTA	CuentasBeneficiariosBancoInternacional	TipoCuenta
FK_CuentasBeneficiariosSIP_BancoBeneficiario	FK_CUENTASBENEFICIARIOS_SIP_BANCOBENEFICIARIO	CuentasBeneficiariosSIP	BancoBeneficiario
FK_CuentasBeneficiarios_Beneficiario	FK_CUENTASBENEFICIARIOS_BENEFICIARIO	CuentasBeneficiariosSIP	Beneficiario
FK_CuentasBeneficiarios_TipoCuenta	FK_CUENTASBENEFICIARIOS_TIPOCUENTA	CuentasBeneficiariosSIP	TipoCuenta
FK_CuentasProveedoresBancoInternacional_BancoBeneficiario	FK_CUENTASPROVEEDORESBANCOINTERNACIONAL_BANCOBENEFICIARIO	CuentasProveedoresBancoInternacional	BancoBeneficiario
FK_CuentasProveedoresBancoInternacional_Proveedor	FK_CUENTASPROVEEDORESBANCOINTERNACIONAL_PROVEEDOR	CuentasProveedoresBancoInternacional	Proveedor
FK_CuentasProveedoresBancoInternacional_TipoCuenta	FK_CUENTASPROVEEDORESBANCOINTERNACIONAL_TIPOCUENTA	CuentasProveedoresBancoInternacional	TipoCuenta
FK_CuentasProveedoresSIP_BancoBeneficiario	FK_CUENTASPROVEEDORES_SIP_BANCOBENEFICIARIO	CuentasProveedoresSIP	BancoBeneficiario
FK_CuentasProveedoresSIP_Proveedor	FK_CUENTASPROVEEDORES_SIP_PROVEEDOR	CuentasProveedoresSIP	Proveedor

FK_CuentasProveedoresSIP_TipoCuenta	FK_CUENTASPROVEEDORESIP_TIPOCUENTA	CuentasProveedoresSIP	TipoCuenta
FK_Educacion_Funcionario	FK_EDUCACION_FUNCIONARIO	Educacion	Funcionario
FK_Educacion_NivelEducacion	FK_EDUCACION_NIVEL EDUCACION	Educacion	NivelEducacion
FK_Educacion_Pais	FK_EDUCACION_PAIS	Educacion	Pais
FK_Especialista_TipoEspecialista	FK_ESPECIALISTA_TIPOESPECIALISTA	Especialista	TipoEspecialista
FK_Familia_Funcionario	FK_FAMILIA_FUNCIONARIO	Familia	Funcionario
FK_Familia_Pais	FK_FAMILIA_PAIS	Familia	Pais
FK_Familia_Sexo	FK_FAMILIA_SEXO	Familia	Sexo
FK_Funcionario_Area	FK_FUNCIONARIO_AREA	Funcionario	Area
FK_Funcionario_AreaOcupacion	FK_FUNCIONARIO_AREAOcupacion	Funcionario	AreaOcupacion
FK_Funcionario_Cargo	FK_FUNCIONARIO_CARGO	Funcionario	Cargo
FK_Funcionario_Clase	FK_FUNCIONARIO_CLASE	Funcionario	Clase
FK_Funcionario_Cluster	FK_FUNCIONARIO_CLUSTER	Funcionario	Cluster
FK_Funcionario_Contrato	FK_FUNCIONARIO_CONTRATO	Funcionario	Contrato
FK_Funcionario_EstadoCivil	FK_FUNCIONARIO_ESTADOCIVIL	Funcionario	EstadoCivil
FK_Funcionario_Pais	FK_FUNCIONARIO_PAIS	Funcionario	Pais
FK_Funcionario_Sexo	FK_FUNCIONARIO_SEXO	Funcionario	Sexo
FK_Funcionario_TipoActividad	FK_FUNCIONARIO_TIPOACTIVIDAD	Funcionario	TipoActividad
FK_Funcionario_Unidad	FK_FUNCIONARIO_UNIDAD	Funcionario	Unidad
FK_FuncionarioIdioma_Funcionario	FK_FUNCIONARIOIDIOMA_FUNCIONARIO	FuncionarioIdioma	Funcionario
FK_FuncionarioIdioma_Idioma	FK_FUNCIONARIOIDIOMA_IDIOMA	FuncionarioIdioma	Idioma
FK_NominaBancoInternacional_Beneficiario	FK_NOMINABANCOINTERNACIONAL_BENEFICIARIO	NominaBancoInternacional	Beneficiario
FK_NominaBancoInternacional_CuentasBeneficiariosBancoInternacional	FK_NOMINABANCOINTERNACIONAL_CUENTASBENEFICIARIOSBANCOINTERNACIONAL	NominaBancoInternacional	CuentasBeneficiariosBancoInternacional
FK_NominaBancoInternacional_TipoCuenta	FK_NOMINABANCOINTERNACIONAL_TIPOCUENTA	NominaBancoInternacional	TipoCuenta
FK_NominaSIP_BancoBeneficiario	FK_NOMINASIP_BANCOBENEFICIARIO	NominaSIP	BancoBeneficiario
FK_NominaSIP_Ben	FK_NOMINASIP_BEN	NominaSIP	Beneficiario

eficiario	EFICIARIO		
FK_NominaSIP_CuentasBeneficiariosSIP	FK_NOMINASIP_CUENTASBENEFICIARIOS SIP	NominaSIP	CuentasBeneficiarios SIP
FK_NominaSIP_TipoCuenta	FK_NOMINASIP_TIPO CUENTA	NominaSIP	TipoCuenta
FK_NominaUnificada_BancoBeneficiario	FK_NOMINAUNIFICADA_BANCOBENEFICIARIO	NominaUnificada	BancoBeneficiario
FK_Prestamo_EstadoOperacion	FK_PRESTAMO_ESTADOOPERACION	Prestamo	EstadoOperacion
FK_Prestamo_Proyecto	FK_PRESTAMO_PROYECTO	Prestamo	Proyecto
FK_Prestamo_TipoOperacion	FK_PRESTAMO_TIPOOPERACION	Prestamo	TipoOperacion
FK_TransferenciaBancoInternacional_CuentasProveedoresBancoInternacional	FK_TRANSFERENCEBANCOINTERNACIONAL_CUENTASPROVEEDORES BANCOINTERNACIONAL	TransferenciaBancoInternacional	CuentasProveedores BancoInternacional
FK_TransferenciaSIP_CuentasProveedoresSIP	FK_TRANSFERENCEBANCOINTERNACIONAL_CUENTASPROVEEDORES SIP	TransferenciaSIP	CuentasProveedores SIP
FK_TransferenciaUnificada_BancoBeneficiario	FK_TRANSFERENCEBANCOINTERNACIONAL_CUENTASPROVEEDORES SIP	TransferenciaUnificada	BancoBeneficiario
FK_TransferenciaUnificada_Proveedor	FK_TRANSFERENCEBANCOINTERNACIONAL_CUENTASPROVEEDORES SIP	TransferenciaUnificada	Proveedor

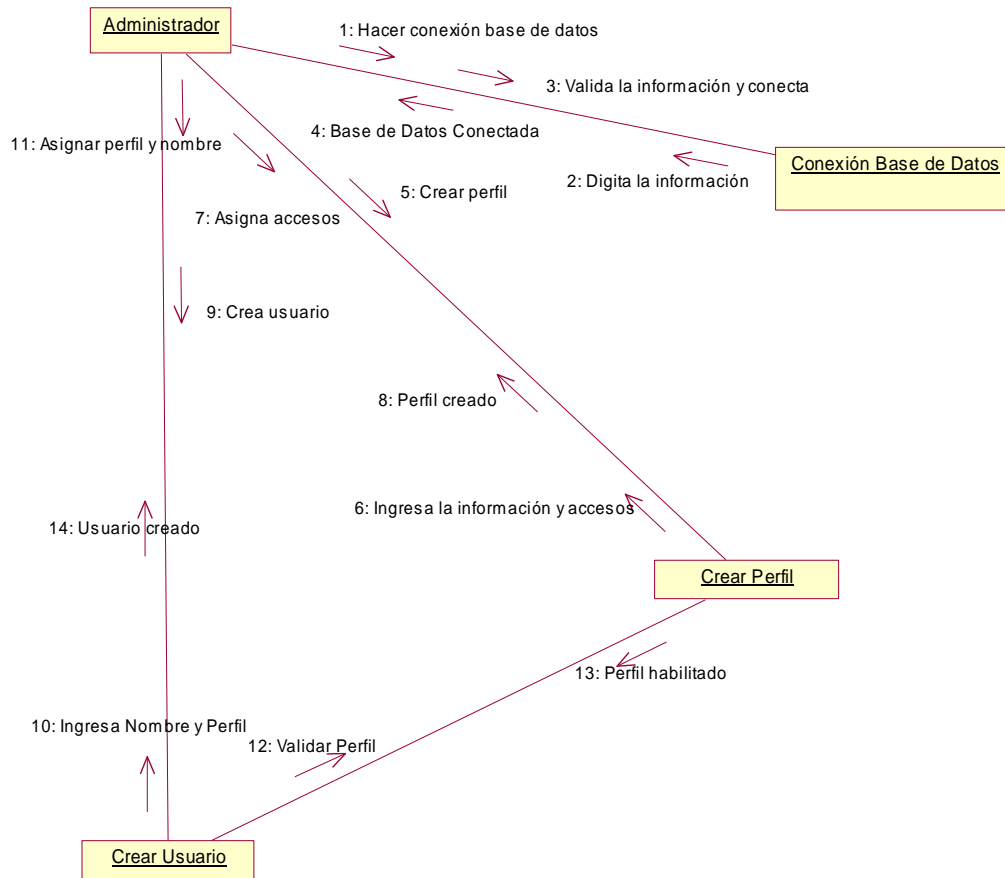
3.2.1.3 Diagrama de Objetos

3.2.2 MODELO DINÁMICO

3.2.2.1 Diagramas de Interacción

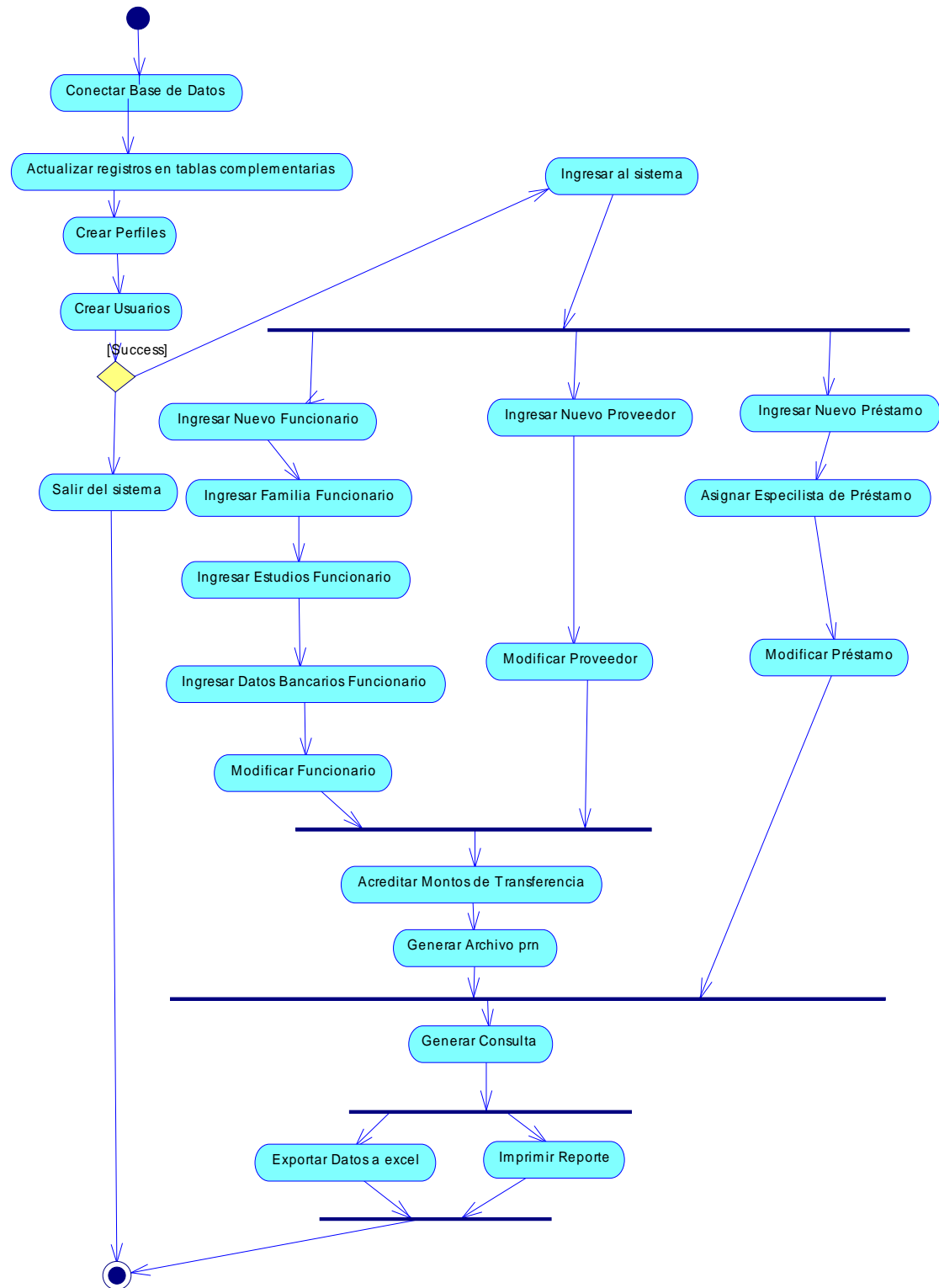
3.2.2.1.1 Diagrama de Secuencia

3.2.2.1.2 Diagrama de Colaboración

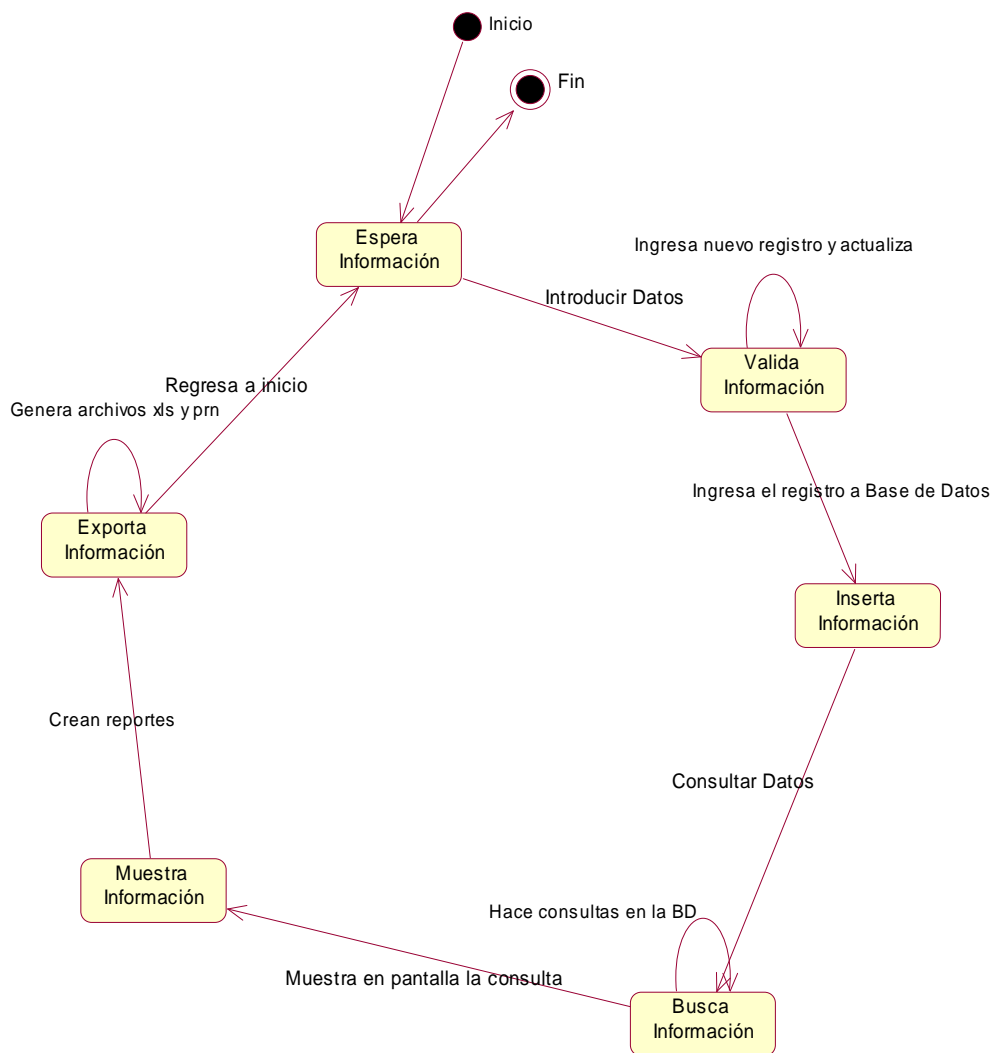


3.2.3 MODELO FUNCIONAL

3.2.3.1 Diagrama de Actividades



3.2.3.2 Diagrama de Estados



3.3 CONSTRUCCIÓN

3.3.1 PRINCIPALES ALGORITMOS

3.3.1.1 Guardar

Asigna los valores encontrados en los textbox a variables.

Hace conexión a la tabla de la base de datos correspondiente

Ingresa un nuevo registro en la tabla

Actualiza la información en la tabla

Valida información ingresada

Si la información es incorrecta o incompleta muestra mensaje de error

Termina evento Guardar

Si la información es correcta muestra mensaje de confirmación de ingreso

3.3.1.2 Modificar

Asigna los valores encontrados en los textbox a variables.

Hace conexión a la tabla de la base de datos correspondiente

Forma una instrucción sql Insert con la información de las variables con condición Where con la Calve primaria de la Tabla Funcionario que es el número de empleado.

Ejecuta la instrucción SQL

Actualiza la información en la tabla

Valida información ingresada

Si la información es incorrecta o incompleta muestra mensaje de error

Termina evento Modificar

Si la información es correcta muestra mensaje de confirmación de actualización

3.3.1.3 Buscar

Asigna a una variable el parametro de busqueda siempre clave primaria de la tabla.

Se forma una instrucción SQL con la información suministrada.

Se ejecuta la instrucción SQL

3.3.1.4 Mostrar

Se asignan a los textbox la información consultada en el algoritmo Buscar..

3.3.1.5 Generar Archivo

Se ejecuta el algoritmo Buscar.

Se crea un archivo plano con el nombre de la fecha del sistema en formato largo.

Se hace un count del resultado de la Instrucción SQL del algoritmo Buscar.

Se crea un ciclo For con los datos resultante del Count

Se crea nuevas lineas en este archivo con la informción de la consulta.

Se muestra en pantalla la confirmación de terminación exitosa de la creación del archivo

3.3.1.6 Exportar Datos

Se utiliza el algoritmo Buscar.

Se abre la aplicación Excel

Se cuenta el número de Registros resultantes con el count y el número de columnas en la instrucción sql de Buscar.

Se crea una hoja de Excel con el número de columnas y filas resultantes del paso anterior.

Se llena las celdas una a una con la información

Estos son los principales algoritmos del Sistema Unificado de Proveedores SUP, se comenta que SUB tiene en su codificación 2 Módulos en el uno se guardan todas las variables globales del Sistema y en el segundo todas las funciones que facilitan la construcción y entendimiento de la codificación del Sistema

3.4 PRUEBAS

3.4.1 PRUEBA FUNCIONAL

Objetivo General: Describir los datos de entrada al programa y describir de forma precisa la salida esperada para estos datos de prueba.

Objetivo Específico: Crear un nuevo funcionario Mario Mena y asignarle dos proyectos en ejecución como especialista sectorial principal y 1 proyecto como especialista alterno, además generar un archivo .prn para transferencia interbancaria por la primera quincena de trabajo del mes de Junio de 2008 por el monto de 1200.45 dólares a la Cta. Corriente 145552525 del Banco del Pichincha, se debe generar el reporte del archivo de transferencia, además se debe calcular la edad, siendo la fecha de nacimiento el 24 de

mayo de 1972, el tiempo de servicio en BID y COF; siendo las fechas de ingreso 15 mayo 1995 y 20 febrero 2000 según el orden descrito.

Primero, ingresamos al Sistema SUP en el menú escogemos Funcionarios y elegimos la opción Nuevo, en el formulario que nos aparece ingresamos toda la información respectiva de acuerdo a los datos proporcionados, como lo muestra la Figura 3.1. Luego damos click en el botón Guardar, nos aparecerá un mensaje que confirma que ha sido ingresado el registro con éxito, como lo muestra la Figura 3.2.

Con esto hemos registrado al funcionario, como le debemos hacer un Transferencia interbancaria por la primera quincena trabajada, en el menú Pago Nómina damos click en la opción Cuentas Beneficiarias, se nos desplegará el formulario de la Figura 3.3, en este debemos escoger el Funcionario Mena Mario Eduardo del combo de selección e ingresar el Número de Cuenta, el tipo de cuenta y el Banco Beneficiario de acuerdo a los datos proporcionados.

SISTEMA UNIFICADO DE PROVEEDORES SUP - [NUEVO FUNCIONARIO]

Sesión Administración Funcionarios Pago Nomina Proyectos Proveedores Pago Proveedores Consultas Mantenimiento Ayuda

Ecuador

Datos Personales

Número de Empleado 241398 CI ó Pasaporte 1715097895

Nombre Mario Eduardo Mena Carrasco
Nombre 1: Nombre 2: Apellido 1: Apellido 2:

Dirección v. 12 de Octubre Luis Cordero N5674 La Floresta
Calle Principal Calle Transversal Número Barrio

Teléfono 2453452 098765432 098123456 Sexo Masculino
Domicilio Celular Personal Celular Conyuge

Fecha de Nacimiento 24/05/1972 País Ecuador Estado Civil Casado

Información General

Oficina 1008 Extensión Oficina 6945 Parqueadero BID-65
Celular BID 099678543 Tipo Blackberry E-mail mariom@iadb.org

Relación Contractual

Cargo Especialista
Tipo Funcionario Local Unidad MIF/CEC
Clúster Social
Clase Profesional Contrato Activo
Actividad Operativo Área FOMIN

Pago Transferencia Bancaria

Poner un check si al funcionario se le hará transferencia interbancaria

NUEVO GUARDAR SALIR

ID Usuario felipeb

10/06/2008 18:51 Estado CONECTADO <<Base de Datos >>SUP_Desarrollo <<Nombre Usuario>> Felipe Barrera <<Perfil Usuario>> Administrador

Figura 3.1 Ingreso de Nuevo Funcionario



Figura 3.2 Mensaje de ingreso exitoso

 A screenshot of a web-based form for entering banking information. The form is divided into two main sections: 'Datos Beneficiario' and 'Datos Bancarios'.

 Under 'Datos Beneficiario':

- 'Funcionario' is a dropdown menu showing 'MENA MARIO EDUARDO'.
- 'Número Empleado' is a text input field containing '241398'.
- 'Identificación' is a text input field containing '0001715097895'.

 Under 'Datos Bancarios':

- 'Banco Beneficiario' is a dropdown menu showing 'Banco del Pichincha'.
- 'Tipo Cuenta' is a dropdown menu showing 'Cuenta Corriente'.
- 'Número de Cuenta' is a text input field containing '0000000000145552525'.

 At the bottom of the form, there are three buttons: 'GUARDAR', 'CANCELAR', and 'SALIR'.

Figura 3.3 Formulario de ingreso información bancaria

Continuando con el ejercicio de prueba, a este funcionario debemos asignarle 2 operaciones como Especialista Sectorial, para lo cual en el menú de SUP escojemos la opción Asignación, se nos desplegará el formulario de la Figura 3.4, escojemos del combo contrato el préstamo para asignarle a Mario Mena, y en la casilla de Especialista Sectorial escojemos el Nombre Mena Mario Eduardo, en alterno cualquier otro diferente y en financiero lo mismo; por último damos click en Guardar.

El mismo procedimiento lo repetimos como lo muestran las Figuras 3.5 y 3.6 en el orden del Sectorial y Alterno para cumplir con los requerimientos de la prueba.

Operaciones	
Contrato	1002/SF-EC
Ejecutor	EC-MIDUVI
Descripción	Apoyo al Sector Habitacional Fase I
Especialista	
Sectorial	Mena Mario
Alternativo	Guerrero Rosa
Financiero	Schneider Santiago
<input type="button" value="GUARDAR"/> <input type="button" value="CANCELAR"/> <input type="button" value="SALIR"/>	

Figura 3.4 Asignación de Operación 1

Operaciones	
Contrato	1026/OC-EC
Ejecutor	EC-CONAM
Descripción	Atención Integrada Niños y Menores S.R.
Especialista	
Sectorial	Mena Mario
Alternativo	Auerbach Paula
Financiero	Sotomayor Elsa
<input type="button" value="GUARDAR"/> <input type="button" value="CANCELAR"/> <input type="button" value="SALIR"/>	

Figura 3.5 Asignación Operación 2

Operaciones	
Contrato	ANT/SF-1031-EC
Ejecutor	MEF
Descripción	Banco Procredit S.A.
Especialista	
Sectorial	Guerrero Rosa
Alternativo	Mena Mario
Financiero	Schneider Santiago
<input type="button" value="GUARDAR"/> <input type="button" value="CANCELAR"/> <input type="button" value="SALIR"/>	

Figura 3.6 Asignación Operación 3

Continuamos con la Transferencia Interbancaria que debemos hacer a este funcionario, para lo cual, en el menú escojemos la opción Pago Nómina, Nueva Transferencia, se nos desplegará el formulario de la Figura 3.7, en este formulario del combo de selección escojemos el Funcionario Mena Mario Eduardo e ingresamos el monto a ser acreditado más la fecha en la que se debe hacer la transferencia a su cuenta del Banco del Pichincha.

The image shows a web form titled "BANCO INTERAMERICANO DE DESARROLLO". The form is divided into several sections:

- Datos Beneficiario:** Includes a dropdown for "Funcionario" (MENA MARIO EDUARDO), a text input for "Número Empleado" (241398), and a text input for "Identificación" (0001715097).
- Valor Acreditado:** Includes a text input for "Cantidad" (1200.45) and a dropdown for "Fecha de Pago" (20080615).
- Tipo Transacción:** Includes two radio buttons: "Transferencia Interbancaria SIP" (selected) and "Banco Internacional".
- Banco Beneficiario:** Includes a dropdown for "Banco" (Banco del Pichincha), a text input for "Cuenta" (0000000000145552525), and a dropdown for "Tipo Cuenta" (Cta. Corriente).
- Fecha Pago Nomina:** Includes three dropdowns: "Quincena" (Primera), "Mes" (Junio), and "Año" (2008).

At the bottom of the form, there are five buttons: "GUARDAR", "NUEVO", "CONSULTAR", "SALIR", and "GENERAR ARCHIVO".

Figura 3.7 Acreditación monto por quincena

En este mismo formulario, Figura 3.7 damos click en el botón Consultar, después de haber ya guardado el actual registro y se nos desplegará en pantalla el formulario de la Figura 3.8 con la quincena, mes y año puestos por default para la consulta de los registros para transferencia interbancaria.

SISTEMA UNIFICADO DE PROVEEDORES SUP - [CONSULTA NOMINA]

Sesión Administración Funcionarios Pago Nomina Proyectos Proveedores Pago Proveedores Consultas Mantenimiento Ayuda

Ecuador

Parámetros de la Consulta
 Quincena: Primera Mes: Junio Año: 2008 [Consultar]

BENEFICIARIO	EMPLEADO	FECHA_PAGO	CANTIDAD	BANCO
▶ MENA MARIO EDUAR	241398	20080615	1200,45	Banco del Pichincha

Ordenar por: []
 Cerrar Imprimir Eliminar

Transacciones por Banco

Num. Registros	Banco	Monto
▶ 1	Banco del Pichincha	1200,45

RESUMEN
 Suma total de registros: 1
 Suma total de dinero: 1200,45 USD

ID Usuario: felipeb

20/07/2008 16:10 Estado CONECTADO <<Base de Datos>>SUP_Desarrollo <<Nombre Usuario>> Felipe Barrera <<Perfil Usuario>> Administrador

Figura 3.8 Consulta de pago de quincena

Damos clic en el botón Imprimir para generar el reporte de transferencias de la Figura 3.9.

BANCO INTERAMERICANO DE DESARROLLO
 REPRESENTACION EN ECUADOR

Beneficiario	Empleado	Fecha	Monto
MENA MARIO EDUARDO	241398	20080615	1200,45
TOTAL			1200,45

Figura 3.9 Reporte de pago de quincena

En el formulario de la Figura 3.7 damos clic en el botón generar archivo y nos aparecerá el mensaje de la Figura 3.10 con la confirmación del número de registros y el monto de los mismos para ese período.

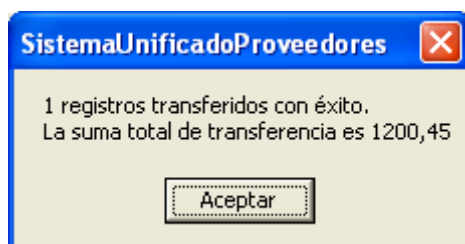


Figura 3.10 Mensaje de confirmación de registros y monto para transferencia

Para ir completando nuestra prueba, nos hace falta generar los reportes de edad y tiempo de servicio del Funcionario Mena Mario Eduardo, para generar estos reportes vamos al menú Reportes y escojemos el que necesitamos, Edad Funcionario y Tiempo de Servicios, el resultado será el mostrado en las Figura 3.11 y 3.12 respectivamente.



BANCO INTERAMERICANO DE DESARROLLO

REPRESENTACION EN ECUADOR

REPORTE DE EDADES FUNCIONARIOS ACTIVOS

Número Empleado	Nombre	Edad
109117	Alemán Marco	40
11760	Auerbach Paula Florencia	33
241398	Mena Mario Eduardo	36
258965	Sotomayor Elsa Elia	0
587845	Villalba Crcitina Alexandra	0
597843	Frias Gabriel Eduardo	22
852951	Reyes Juan Francisco	32
897423	Castellanos Juan Hernando	22
951357	Poveda Rafael Antonio	0

Figura 3.11 Reporte Edad de Funcionarios Activos



BANCO INTERAMERICANO DE DESARROLLO

REPRESENTACION EN ECUADOR

REPORTE DE TIEMPO SERVICIO FUNCIONARIOS ACTIVOS

Número Empleado	Nombre	Tiempo Servicio	
		BID	COF
109117	Alemán Marco	0	1
11760	Auerbach Paula Florencia	28	2
241398	Mena Mario Eduardo	13	8
258965	Sotomayor Elsa Elia	0	0
587845	Villalba Crsitina Alexandra	0	0
597843	Frias Gabriel Eduardo	2	1
852951	Reyes Juan Francisco	13	8
897423	Castellanos Juan Hernando	2	1

Figura 3.12 Reporte de Tiempo de Servicio

Verificamos los Préstamos asignados al Funcionario Mena Mario Eduardo como lo muestran las Figuras 3.13 y 3.14, de Principal y de alterno respectivamente.

Parámetros de Búsqueda

Especialista: Cargo Alterno Financiero

Especialista	Proyecto	Descripción	Préstamo	Ejecutor
▶ Mena Mario	EC0138	Apoyo al Sector Habitac	1002/SF-EC	EC-MIDUMI
Mena Mario	EC0173	Atención Integrada Niñ	1026/OC-EC	EC-CONAM

Resumen
2 Operaciones a Cargo

Figura 3.13 Consulta de Proyectos por Especialista Sectorial

Parámetros de Búsqueda

Especialista: Cargo Alterno Financiero

Especialista	Proyecto	Descripción	Préstamo	Ejecutor
▶ Guerrero Rosa	336/EC	Banco Procredit S.A.	ANT/SF-1031-EC	MEF

Resumen
1 Operaciones como
alternas

Figura 3.14 Consulta de Proyectos por Especialista Alternos

CONCLUSIONES

El Sistema Unificado de Proveedores SUP se desarrolló adoptando la Metodología OMT, reúne todos los requisitos para catalogarlo como un Sistema de Calidad que cumple con los requerimientos que tiene el Banco Interamericano de Desarrollo BID, Representación Ecuador, para el manejo de Proveedores.

El BID define el Universo de Proveedores fusionando los Funcionarios del BID y las empresas que otorgan un servicio al BID.

SUP es un Sistema que facilita el manejo de información relacionada con el Funcionario, su familia, la educación adquirida y los proyectos que tienen a cargo un subgrupo de Funcionarios definidos como Especialistas.

SUP minimiza y facilita la generación de información para transferencias interbancarias de todos sus proveedores (Funcionarios y Empresas Externas).

SUP es una herramienta poderosa que automatiza las acciones realizadas en el Sector Administrativo del Banco Interamericano de Desarrollo Representación Ecuador, minimizando el tiempo destinado a las actividades que SUP, es capaz de administrar como manejo de Funcionarios y pago de proveedores por transferencias interbancarias, mitiga errores antes de crear procesos de flujo para aprobación de transferencias y se ha convertido en una herramienta confiable para cumplir con procesos delicados, como es el manejo de información confidencial de sueldos e información personal de los funcionarios que trabajan para el BID en la oficina de país.

RECOMENDACIONES

Se recomienda capacitación al personal que va interactuar con el Sistema SUP para evitar inconsistencias dentro de la información que administra el Sistema.

Se deben acordar formatos de informes de errores que presente el Sistema SUP para la respectiva corrección y mitigar lo más posible fallas que se puedan dar en el proceso de producción de SUP.

Se debe mantener una administración responsable del Sistema SUP por parte del administrador del mismo, debido a la información tan delicada que se maneja dentro de este Sistema.

Se debe establecer mediante un informe que contemple el costo beneficio el incremento de nuevos requerimientos por parte del BID para la reforma y adición de nuevos modulos dentro del Sistema.

GLOSARIO DE TÉRMINOS

Análisis de Sistemas.- análisis de un problema que la organización tratara de resolver con un sistema de información.

Automatización.- uso de la computadora para acelerar la realización de tareas existentes.

Cambio de paradigma.- reconceptualización radical de la naturaleza del negocio y de la naturaleza de la organización.

Ciclo de vida de los sistemas.- metodología tradicional para desarrollar un sistema de información que divide el proceso de desarrollo de sistemas en fases formales que deben completarse secuencialmente con una división muy formal de las actividades de los usuarios finales y los especialistas de sistemas de información.

Desarrollo de sistemas.- actividades involucradas en la producción de una solución de sistemas de información para un problema organizacional o una oportunidad.

Diseño de Sistemas.- Detalla cómo cumplirá un sistema los requerimientos de información determinado por el análisis de sistemas.

Estudio de factibilidad.- como parte del proceso de análisis de sistemas, la manera de determinar si la solución es factible, dados los recursos y limitaciones de la organización.

Interfaz de usuario final.- parte de un sistema de información a través de la cual el usuario final interactúa con el sistema, como las pantallas y los comandos en línea.

Iterativos.- proceso consistente en repetir los pasos una y otra vez para construir un sistema.

Mantenimiento.- cambios en el hardware, software, documentación o procedimientos para que un sistema en producción corrija errores, cumpla con los nuevos requerimientos o mejore la eficiencia del proceso.

Metamodelo.- modelo que sirve para expresar otros modelos.

Prueba.- proceso exhaustivo y completo que determina si el sistema produce los resultados deseados en condiciones conocidas.

Prueba unitaria.- proceso de probar por separado cada programa del sistema. También se conoce como prueba de programas

Racionalización de procedimientos.- la agilización de procedimientos operativos estandarizados, eliminando los cuellos de botella obvios para que la automatización haga mas eficientes los procedimientos operativos.

Requerimientos de información.- una declaración detallada de las necesidades de información que un nuevo sistema debe satisfacer; identifica quién necesita qué información y cuándo, dónde y cómo.

BIBLIOGRAFÍA

César Liza Ávila, ***Modelando con UML. Principios Aplicaciones***, Primera Edición, Agosto 2001

Edward Yourdon, ***Administrando el ciclo de vida del Sistema***, Editora Campus- Brasil 1989

Edward Yourdon, ***Análisis Estructurado Moderno***, Prentice- Hall Hispanoamericana 1993

Edward Yourdon - Larry Constantine, ***Structured Design***, Prentice-Hall
Roger Presuman, ***Ingeniería del Software - Un enfoque práctico***, McGraw-Hill 1992

Kendall y Kendall, ***Análisis y diseño de Sistemas***, Prentice hall Mexico 1991
Senn, James A., ***Análisis y diseño de Sistemas de Información***, Mc-Graw Hill Mexico 1991

Software Engineering, Methods, management and Case tool, Mc-Graw Hill USA 1991

Victor Weimberg, ***Structured Analysis***, Yourdon Press 1980

ANEXO

MANUAL DE USUARIO