

UNIVERSIDAD DE GUAYAQUIL

Facultad de Ciencias Matemáticas y Físicas

**Carrera de Ingeniería en Sistemas
Computacionales**

**“Sistema de Gestión de Proyectos – Módulo de Integración,
Parametrización y Configuraciones”**

TESIS DE GRADO

Previo a la Obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

Autores:

ANDRADE ORTIZ GREGORIO ARTURO

CASTILLO LOJA MARGARITA LUISA

TOMALÁ MAZZINI CRISTIAN ARGELIS

GUAYAQUIL - ECUADOR

Año: 2008

AGRADECIMIENTO

A Dios por haberme colmado de buena salud y guiado por el camino correcto.

A mis padres y hermanas por el amor y apoyo en todo momento a lo largo de toda mi carrera.

A mis verdaderos amigos con quien compartí momentos agradables y que de una u otra forma hicieron más sencillo el camino hacia el éxito ahora alcanzado.

Gregorio Andrade

Agradezco a todas las personas que de una u otra manera apoyaron que mi carrera universitaria culmine exitosamente, un agradecimiento especial a aquellos amigos con los que compartí muchas divertidas horas de estudio.

Claro y a ti Débora, gracias por tu afecto y compañía.

Cristian Tomalá.

AGRADECIMIENTO

A mi Padre Celestial quién siempre me lo dio todo y me ha bendecido con una vida muy feliz.

Quiero agradecer a mi segunda madre Nieves por siempre hacerme reír, a mis tíos Guadalupe y Amable porque siempre encontré en su hogar felicidad.

A mis primos Vanesa y Julio por la paciencia que me entregaron cada vez que requerí de sus enseñanzas. Un agradecimiento desde el fondo de mi corazón a mi adorado papi Loja por haberme dado sus cuidados.

A Ronald quién me enseñó el poder de ser mejor, gracias por su cálido amor, su amistad y compañía, sobretodo gracias por pensar que el mundo puede ser diferente.

A mis compañeros de trabajo, mi querido Don Galo y Sandra por haberme dado ejemplo de familia y mostrarme con su ejemplo de vida que el amor existe. A mi estimado P. Antonio quién siempre me animó con sus palabras y sus oraciones.

Margarita Castillo.

DEDICATORIA

A mi padre y a mí querida hermana que siempre estuvieron conmigo brindando su ayuda y consejos en todo momento de mi vida.

Gregorio Andrade.

A la mejor madre del mundo, mi madre Mercedes por haberme enseñado el valor del esfuerzo, por madrugar y acompañarme a estudiar desde que era una niña y regalarme un mundo lleno de fantasías y amor.

A mí querido padre Walter quien me ha colmado de cariño, dulzura, abrazos y un apoyo incondicional en toda mi carrera educativa y enseñarme desde pequeña a hablar en público. Los amo.

Margarita Castillo

A mi querida madre, quién ha sido mi inspiración, mi soporte y mi ejemplo. A ella mi más profunda veneración y respeto.

A mi padre, de quién aprendí la actividad más satisfactoria del mundo: el trabajo.

Cristian Tomalá.

TRIBUNAL DE GRADUACIÓN

PRESIDENTE - TRIBUNAL

PRIMER VOCAL

SEGUNDO VOCAL

SECRETARIO

AB.

DECLARACIÓN EXPRESA

“La autoría de la tesis de grado corresponde exclusivamente a las suscritas, perteneciendo a la Universidad de Guayaquil los derechos que generen la aplicación de la misma.”

RESUMEN

Módulo de Integración, Parametrizaciones y Configuraciones

SGPI (Sistema Gestor de Proyectos Informáticos) es un sistema compuesto por doce módulos que permite gestionar la administración de proyectos informáticos en los departamentos de Tecnología de la Información de empresas, instituciones, etc.

El módulo de Integración, Parametrizaciones y Configuraciones, permite a los otros módulos del sistema integrarse en un solo ente administrador, a través de éste el usuario podrá acceder a todas las funcionalidades de SGPI como la aprobación de proyectos, planificación de recursos, controles de hitos en el tiempo, alarmas de advertencias, en fin el sinnúmero de actividades que están presentes cuando se desarrolla un proyecto informático.

Este módulo es el encargado de revisar los doce modelos de datos e integrarlos en un solo modelo funcional para crear la Base de Datos en nuestro motor Oracle 9i, es también el responsable de el control de versiones y cambios sobre la estructura de la base. Además establece la metodología de Integración de los doce proyectos de aplicación creadas en lenguaje Java la configuración del servidor de aplicaciones.

ABSTRACT

Integration, Parameters and Settings Module

IPMS(Informatics Projects Manager System) is a system of twelve modules which lets manage the process of informatics projects which takes place in the different information technology department of enterprises or institutions, etc.

The Integration, Parameters and Settings Module , let to the other modules of the system integrates with each other in one great manager interface, thought it the user be able to access every IPMS' functionalities such as projects approbations , resource planning, checking times in the chronograms, warning alarms, it means a variety of activities that appear when people develop informatics projects.

This module is the responsible to check the twelve data models and integrate them and to create just one model for the Data Base Oracle 9i. Besides the module control the versions and changes in the official database structure. It establishes the integration methodology of the twelve applications projects created with Java and the configuration of the application server.

INDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	IV
TRIBUNAL DE GRADUACIÓN	V
DECLARACIÓN EXPRESA	VI
RESUMEN	VII
INDICE GENERAL	VIII

CAPITULO 1.

SISTEMA GESTOR DE PROYECTOS INFORMATICOS

Introducción.	Pág.
1.1. Antecedentes y Problemática	
1.2. Resolución a la Problemática.....	1
1.3. Actores de SGPI.....	1-1
1.4. Algoritmo del Flujo de Entrada del Sistema.....	1-3
1.5. Módulos de SGPI.....	1-5
1.6. Misión.....	1-12
1.7. Visión.....	1-12
1.8. Objetivos	
1.8.1. Objetivos Generales.....	1-12
1.8.2. Objetivos Específicos.....	1-13
1.9. Alcances.....	1-15
1.10. Arquitectura.....	1-17
1.11. Metodología.....	1-19
1.12. Planificación de Actividades y Cronograma.....	1-21
1.13. Recursos del Proyecto.....	1-25
1.13.1. Recurso Humano.....	1-25
1.13.2. Recursos Hardware.....	1-27

1.13.3.	Recursos Software.....	1-28
1.13.4.	Recursos de Infraestructura.....	1-29
1.14.	Estudio Estadístico.....	1-29
1.15.	Administración del Riesgo.....	1-31
1.15.1.	Identificación del Riesgo.....	1-32
1.15.1.1.	Definiciones de los Riesgos.....	1-32
1.15.2.	Análisis del Riesgo.....	1-36
1.15.2.1.	Impacto y Probabilidad del Riesgo.....	1-37
1.15.3.	Planificación del Riesgo.....	1-43
1.15.4.	Monitoreo del Riesgo.....	1-50

CAPITULO 2

2. Análisis

2.1	Análisis de Requisitos SRD (Documentación de Especificaciones de Requisitos)	
2.1.1.	Recopilación de Información.....	2-1
2.2.	Modelo Entidad–Relación.....	2-2
2.3.	Diccionario de Datos.....	2-2
2.4.	Diagrama de Flujo de Datos.....	2-2
2.4.1.	Nivel 0 (Diagrama de Contexto).....	2-3
2.4.2.	Nivel 1.....	2-3
2.5.	Diagrama de Casos de Uso.....	2-5

CAPITULO 3

3. Diseño

3.1.	Diseño de Componentes.....	3-1
------	----------------------------	-----

Nodo de Requerimiento

Nodo de Pre – Proyecto

Nodo de Control de Hitos

Nodo de de Alarmas

Nodo de Líderes y Colaboradores

Nodo de Infraestructura

Nodo de Proveedores

Nodo de Presupuesto

Nodo de Documentación

Nodo de Reportes

CAPITULO 4

4. Codificación.

4.1. Estándares de Programación de Pl/Sql.....	4-1
4.1.1. Descripción del Paquete GENK_VALIDA.....	4-3
4.1.2. Descripción del paquete GENK_INSERTA.....	4-5
4.1.3. Descripción del paquete GENK_BITACORIZACION.....	4-9
4.2. Estándares De Programación De Java.....	4-15
4.2.1. Controlador Web.....	4-15

CAPITULO 5

5. Pruebas

5.1. Pruebas de Desarrollo.....	5-2
5.1.1. Pruebas de Calidad Interna.....	5-3

5.1.2. Pruebas con Usuarios.....	5-3
5.2. Métricas Consideradas.....	5-4

CONCLUSIONES Y RECOMENDACIONES.

ANEXOS.

BIBLIOGRAFIA Y REFERENCIAS WEB.

INDICE DE FIGURAS

CAPITULO 1.

SISTEMA GESTOR DE PROYECTOS INFORMÁTICOS

	Pág.
Figura 1-1. Arquitectura de Sistema Gestor de Proyectos Informáticos.	1-18
Figura 1-2. Modelo de ciclo de vida Incremental.....	1-20
Figura 1-3. Algoritmo de integración incremental.....	1-21
Figura 1-4. Gráfico de Etapas de desarrollo de SGPI.....	1-22
Figura 1-5. Distribución del tiempo en el proceso de desarrollo del software.....	1-29
Figura 1-6. Administración del Riesgo.....	1-31
Figura 2-1. Recopilación de Información.....	2-1
Figura 2-2. Nivel 0. Diagrama de Contenidos.....	2-3
Figura 2-3. DFD Nivel 1.....	2-4
Figura 2-4. Diagrama de Casos de Uso.....	2-5
Figura 3-1. Diseño de Componentes Parte I	3-3
Figura 3-2. Diseño de Componentes Parte II	3-5
Figura 3-3. Diseño de Componentes Parte III	3-6
Figura. 4-1. Procedimiento P_Cargo de GENK_VALIDA	4-3
Figura. 4-2. Cursor C_CARGOS de GENK_VALIDA.....	4-4
Figura. 4-3. Procedimiento P-CARGOS de GENK_INSERTA.....	4-5
Figura. 4-4. (Continúa) Procedimiento P_CARGOS de GENK_INSERTA.....	4-6
Figura. 4-5. Función F_HOST de GENK_PARAMETROS.....	4-8

Figura. 4-6. Bloque de excepciones del Procedimiento P_CARGOS de GENK_INSERTA.....	4-8
Figura. 4-7. Procedimiento P_Bitacoriza_Info de GENK_BITACORIZACION...	4-10
Figura. 4-8. Encabezado de GENK_GLOBALES.....	4-13
Figura. 4-9. Cuerpo de GENK_GLOBALES.....	4-14
Figura. 4-10. Import del Controlador Web.....	4-15
Figura. 4-11. Clase Controlador Web_Parte1.....	4-17
Figura. 4-12. Clase Controlador Web_Parte2.....	4-18
Figura. 4-13. Clase Controlador Web_Parte3.....	4-19
Figura. 4-14. Clase Controlador Web_Parte4.....	4-20
Figura. 4-15. Clase Controlador Web_Parte5.....	4-21
Figura. 4-16. Clase Controlador Web_Parte6.....	4-23
Figura. 4-17. Clase Controlador Web_Parte7.....	4-26
Figura. 4-18. Clase Controlador Web_Parte8.....	4-27

INDICE DE TABLAS

CAPITULO 1.

SISTEMA GESTOR DE PROYECTOS INFORMÁTICOS

	Pág.
Tabla 1-1. Planificación Etapas de Análisis y Diseño.....	1-23
Tabla 1-2. Planificación Etapas de Codificación, Integración y Pruebas Finales..	1-24
Tabla 1-3. Recursos Humanos de SGPI.....	1-26
Tabla 1-4. Recursos Hardware.....	1-27
Tabla 1-5. Recursos Software.....	1-28
Tabla 1-6. Impacto y Probabilidades del Riesgo.....	1-37
Tabla. 4-1. Tabla de Identificador de Tipos de Datos.....	4-1
Tabla 5-1. Pruebas sobre hardware.....	5-5

CAPITULO 1

SISTEMA GESTOR DE PROYECTOS INFORMÁTICOS.

1. INTRODUCCION

1.1 Antecedentes y Problemática

El área de software / Ingeniería Informática / Tecnología de la Información cualesquiera de los nombres que se use para identificar al área de la empresa cuyo trabajo es desarrollar soluciones informáticas será nuestro lugar de acción que de ahora en adelante llamaremos área de sistemas informáticos.

La Ingeniería de Software no es distinta de las demás ingenierías, es por eso que el área de sistemas informáticos debe producir software como su tópico central siguiendo metodologías, normas de diseño, de mantenimiento, etc., Pero además tiene que encargarse

**Sistema Gestor de Proyectos Informáticos
Módulo de Integración, Parametrización y
Configuraciones.**

de la gestión del proyecto así como lo haría un arquitecto, o un maestro de obra cuando se levanta una construcción; teniendo en cuenta que no solo es importante terminar y entregar la obra, sino que se lo debe hacer en el tiempo exacto para que sea de utilidad optimizando recursos.

1.2. Solución a la Problemática.

Ahora bien, desarrollaremos un software que permita gestionar la producción de sistemas informáticos, **Sistema Gestor de Proyectos Informáticos**; proporcionará al recurso humano del área de sistemas, preocuparse tan solo por la producción del producto, ya que este sistema gestionará los diferentes ámbitos del desarrollo y puesta en marcha de los proyectos, el sistema será quién tendrá el conocimiento de la organización y planificación.

Sistema de Gestión de Proyectos Informáticos (SGPI) es un proyecto emprendido para gestionar la producción del software de las empresas que cuentan con un área de sistemas. SGPI está dividido en 12 módulos a continuación usted podrá observar la planeación, análisis, desarrollo e implementación del **Módulo de Integración, Parametrización y Configuración.**

Una de las fases del ciclo de vida de un sistema es la Integración, hacer que todos los módulos se interrelacionen y comuniquen para lograr el objetivo

deseado y que además funcione correctamente. Cada uno de los 12 módulos en que se dividió a SGPI puestos a prueba por separado puede obtener resultados exitosos pero el cliente necesita un **todo** para satisfacer sus necesidades y claro un todo que funcione correctamente: que cumpla con las necesidades que originaron su requerimiento y que mejore indudablemente la funcionalidad de las actividades del área.

Entonces ¿qué sucederá cuando se las acople a todas en una sola entidad? ¿Funcionaran? Eso dependerá del buen trabajo del equipo de integración, aunque es muy relevante anotar que se va a integrar una gran cadena de eslabones que han sido labrados por separado pero que pasaría si los eslabones no tienen un buen acabado, ni las medidas exactas, el trabajo de integración se convertiría en castigo fácilmente.

Integrar necesita planificación y administración desde el momento mismo de su concepción, estableciendo parámetros de desarrollo, procesos formales, control de los procesos de cambios, entre otros.

Doce grupos con doce módulos bajo su cargo, doce puntos de vistas, 31 personas a las que administrar y una progresión aritmética de consensos a los que hay que llegar. El punto de mayor importancia dentro del proyecto es la Administración del Recurso Humano, facilitar la comunicación y coordinar

las actividades, todos estos puntos deben ser realizados a prior sino queremos fracasar en la integración.

Si no existiera la Gestión de Software, tendríamos doce diferentes formatos, doce diferentes objetivos y doce diferentes resultados disparados cada uno en diversas direcciones, gestionamos el desarrollo de SGPI para administrar recursos humanos, cronogramas, control de cambios, autorizaciones, etc.

1.3. Actores de SGPI

- Usuarios de las diferentes áreas de la Empresa
- Gerentes de cartera de Proyectos
- Project Manager(Gerentes de Proyectos)
- Colaboradores (Ingenieros de Desarrollo)
- Jefe de área de Infraestructura
- Proveedores de Infraestructura
- Proveedores de Soluciones de Software
- Proveedores de Personal Outsourcing

1.4. Algoritmo del flujo de entrada del Sistema

- 1) El requerimiento llega por parte del área Cliente (Marketing, Ventas, etc.) a SISTEMAS y el Gerente de Cartera de Proyectos lo prioriza y lo asigna a un Project Manager, éste realiza un estudio de Factibilidad y el Cronograma del Proyecto y lo devuelve a la Gerencia de Cartera de

Proyectos para su aprobación y asignación de Recursos; una vez asignados los Recursos y aprobado el Cronograma se Inicia El Proyecto.

- 2) Durante el Desarrollo del Proyecto nos encontramos con las Etapas del mismo (análisis, diseño, desarrollo, pruebas, capacitación, puesta en producción) y los documentos (entregables) que se generan en cada etapa.
- 3) Los Project Manager son los encargados de gestionar dentro de las mismas áreas de sistemas el desarrollo del proyecto por ejemplo en caso de que se necesiten servidores, con el área de infraestructura. Asimismo es el que realiza la gestión hacia los Clientes Internos (que enviaron el requerimiento). En Proyectos en conjunto con algún proveedor de Soluciones Informáticas realiza las gestiones de administración del proyecto con el Gerente de Proyecto (Project Manager) de la empresa que provee la solución
- 4) El Project manager es el responsable durante el desarrollo del proyecto, lidera los recursos humanos, entrega el proyecto a los clientes según lo requerido.

1.5. Módulos de SGPI

1) Módulo de Personalización, Administración de Usuarios y Seguridades

- Este módulo me permitirá gestionar a los usuarios de la aplicación.
Cada usuario deberá ser creado a nivel de base de datos y de aplicación.
- Entre las principales opciones de éste módulo tenemos:
- Creación: Usuarios, Roles, Módulos, Sub-módulos, Opciones, Aplicaciones
- Asignaciones de: Usuarios y Roles.
- Roles y Módulos.
- Módulos y sub-módulos.
- Sub-módulos y opciones
- Módulos y Aplicaciones
- Habilitar y deshabilitar Usuarios
- Cambios de Contraseña
- Establecer políticas de Seguridad para Claves
- Personalización de las pantallas y viñetas del usuario dependiendo de su tipo, género, gustos y preferencias.

2) Módulo de Gestión de Requerimientos

- Este módulo es para que las áreas clientes formalicen la solicitud de requerimientos de proyectos hacia el área de sistemas. Entre sus principales opciones tenemos:
- Definición de las áreas que pueden generar un Requerimiento
- Creación de Tipos de Requerimientos
- Creación de Requerimientos
- Flujo de Aprobación de un Requerimiento
- Definición de Niveles de Aprobación
- Revisiones y versiones de Requerimientos
- Generación de un Requerimiento
- Priorización de los requerimientos de un departamento
- Priorización de los requerimientos de la empresa
- Adjuntar imágenes/Flujos/pantallas

3) Módulo de Gestión del Pre-Proyecto

- Ayudará a la gestión del pre –proyecto entre el gerente y el líder del proyecto. Iniciando con la solicitud de recursos para el proyecto. Entre sus principales opciones tenemos
- Elaboración de Plan Técnico
- Elaboración del Cronograma (adjuntar)
- Solicitud de Recursos

- Asignación de un Código de Proyecto
- Asignación de los Recursos
- Definición de Principales Hitos del proyecto

4) Módulo de Control de Etapas e Hitos del Proyecto

- Ayudará a las revisiones del líder del proyecto y del área generadora del proyecto. Se definirán estos 2 tipos de hitos y se podrán generar Actas de Reunión de Seguimientos, Informes automáticos indicando si el proyecto está en VERDE, AMARILLO o ROJO.
- De acuerdo al Estatus del Proyecto se generarán alarmas del mismo a los responsables. Entre las principales opciones tenemos:
- Revisión de Hitos
- Elaboración y Generación de Actas
- Afectaciones al Módulo de Requerimientos
- Afectaciones al Módulo de presupuestos y Costos
- Afectaciones al Módulo de Gestión del proyecto

5) Módulo de administración de Documentación Digital del Proyecto

- En éste módulo se crearán categorías de Documentación mediante las cuales los diferentes departamentos de la empresa, incluido el departamento de sistemas colocarán los documentos del proyecto. Habrá 2 opciones de realizarlo: a través de formularios que solicitan la

información y adjuntar el documento. Entre sus principales opciones tenemos:

- Creación de categorías de documentos
- Elaboración de Plan Técnico
- Elaboración de Doc. Análisis
- Elaboración de Doc. Diseño
- Elaboración de Doc. Plan de Pruebas
- Elaboración de Doc. Manuales de Capacitación
- Elaboración de Doc. Pase a Producción
- Elaboración de Cierre de Proyecto

6) Módulo de Integración, Parametrizaciones y Configuraciones

- Este módulo esta detallado a lo largo de la documentación.

7) Módulo de Control de Líderes y Colaboradores

- En éste módulo se podrán ingresar las empresas que prestan servicio de outsourcing así como registrar a los colaboradores internos y externos. Se incluirá una base de datos completa con sus fotos digitalizadas y huella dactilar. Se crearán Pruebas de Conocimientos para validar su conocimiento inicial. Asimismo se crearán las evaluaciones que serán realizadas al final de cada proyecto, su

calificación, sus competencias y habilidades. Entre sus principales opciones tenemos:

- Creación de Líderes y colaboradores
- Adjuntar hoja de vida
- Prueba de Conocimientos
- Administración de Evaluaciones

8) Módulo de Consultas y Reportes Gerenciales

- En éste módulo tendremos todas las consultas y reportes para las Gerencias de los diferentes departamentos acerca de los proyectos de sus áreas. Consultas y Reportes para las Gerencias de Proyectos. Entre las principales opciones tenemos:
 - Consulta y Reporte de Mejores Colaboradores
 - Consulta y Reporte de Mejores Líderes
 - Consulta y Reporte de Proyectos más Extensos
 - Consultas y Reportes de Presupuestos generales

9) Módulo de Control Presupuesto y Costos del Proyecto

- En éste módulo se definirá el Presupuesto de todas las áreas de Sistemas para Proyectos. Cada Proyecto contará con su propio presupuesto relacionado directamente con los costos del proyecto. Para el cálculo de estos Costos se tomarán en cuenta los Recursos,

Infraestructura, Suministros entre los rubros importantes. Entre las principales opciones tenemos:

- Definición de los Costos del Proyecto
- Asignación de Presupuesto
- Flujo de Transferencia de presupuesto

10) Módulo de Comunicación y Alarmas del Proyecto

- En éste módulo se definirá a los responsables de las áreas y del departamento de Sistemas por cada Etapa del Proyecto a quienes les llegarán las alarmas del mismo. Las alarmas deben ser configuradas vía SMS y vía email. Entre sus principales opciones tenemos
- Definición de responsables por Etapa
- Definición del Mensaje/Alarma por Etapa
- Definición de Criterios / disparadores de alarmas

11) Módulo de Gestión de Proveedores y Administración de Contratos

- En éste módulo se administrarán a todos los proveedores de servicios para el área de Sistemas, en el que se incluyen proveedores de hardware, de software, de infraestructura, de personal y sus respectivos contratos – anexos, se podrán realizar evaluación de Proveedores de Soluciones Informáticas, se definirán los diferentes

costos de sus servicios, se publicarán para los proveedores actuales Solicitudes de Servicio para que los proveedores concursen entre sí y así poder escoger la mejora alternativa. Entre las principales opciones tenemos:

- Definición de Proveedores y Contratos
- Evaluación de Soluciones Informáticas
- Definición de Costos de Servicios
- Concurso de Solicitudes de Servicio

12) Módulo de Gestión de Infraestructura del Proyecto

- En éste módulo se administrará y gestionará la infraestructura que se requiere para un proyecto, desde la compra de la infraestructura, su llegada e implementación. Si se afecta el Cronograma por la llegada a des-tiempo de la infraestructura se deberá enviar las alarmas correspondientes y se deberán actualizar los Hitos y el Cronograma respectivo. Entre las principales opciones tenemos:
 - Definición de la Infraestructura Total para Proyectos
 - Definición de Infraestructuras por cada Proyecto
 - Definición de Responsables de Infraestructura
 - Definición de Servicios que ofrece la infraestructura
 - Definición de Fechas de entrega, desembarco e implementación

1.6. Misión

Ofrecer a las personas involucradas en la gestión y desarrollo de proyectos informáticos una herramienta altamente funcional capaz de automatizar la mayoría de los procesos inherentes a la administración de los mismos teniendo como premisa fundamental la disponibilidad de información totalmente confiable que sirva como base para la toma de decisiones de manera oportuna y acertada.

1.7. Visión

Que la solicitud de proyectos o requerimientos futuros se convierta en una secuencia sencilla y transparente de procesos interrelacionados que arrojen como resultado soluciones rápidas , altamente funcionales que satisfagan en su totalidad las necesidades plasmadas en los requerimientos; permitiendo así a la empresa una mayor proyección de crecimiento en el medio en el cual se desempeña.

1.8. Objetivos

1.8.1. Objetivos Generales

SGPI mejorará la productividad del área de Sistemas, facilitando la comunicación y entendimiento entre los distintos participantes en la producción de software a lo largo del ciclo de vida del proyecto, teniendo en

cuenta su papel y responsabilidad, así como las necesidades de todos y cada uno de ellos.

Módulo de Integración, Configuraciones y Parametrización integrará todos los MER de los diferentes módulos, construirán las APIS generales de integración con otros sistemas. Tendrán las parametrizaciones y configuraciones generales del Sistema. Entre sus principales opciones tenemos:

- Parametrización de localidades, empresas, oficinas
- Definición de Api's de Integración con otros Sistemas
- Configuraciones generales

Ser gestores de software, planificar y administrar los recursos, controlar los cambios de versiones.

1.8.2. Objetivos Específicos

Con respecto a la Integración y Funcionalidad

- Integrar el modelado de datos (MER), procesos, ambientes, API's de los módulos del sistema SGPI.
- Integrar los diferentes módulos del sistema, asegurando la interoperabilidad y funcionalidad.

- Crear parametrización de reglas y políticas del sistema gestor.

Con respecto a la Gestión de Personal

- Liderar los grupos de desarrollo, evitando conflictos y propender a la negociación.
- Facilitar la comunicación y entendimiento entre los distintos participantes de los módulos de SGPI en la producción de software a lo largo del ciclo de vida del proyecto, teniendo en cuenta su papel y responsabilidad, así como las necesidades de todos y cada uno de ellos, motivar al equipo que conforma el proyecto.
- Mantener comunicado al grupo de proyecto de las actividades a realizarse y del estado del proyecto.

Con respecto a la Planificación y Administración

- Identificar y gestionar los riesgos inherentes al proyecto con respecto al Recurso humano, y tecnologías utilizadas.
- Conocer las problemáticas de integración más características así como las estrategias, estándares y tecnologías de integración.

Con respecto a la Gestión de Software

- Organizar al personal para un desarrollo de software eficiente y eficaz.
- Control de versiones a lo largo del desarrollo del proyecto.

1.9. Alcances

- Definir la parametrización a usar en los modelos entidad – relación, controlar su ejecución (nombres de entidades, tipos de datos, etc.).
- Revisar el diseño del MER de cada uno de los módulos, verificando si no existe redundancia de tablas o relaciones, verificar los problemas de análisis de manera general y esto incluye revisar los MER de cada uno de los módulos, más no realizar el trabajo implícito de cada módulo en el análisis y diseño de sus diagramas como lo es el proceso de normalización.
- Para el proceso de integración se usará el modelo de ciclo de vida Incremental (**Ver Figura 1-3**), para garantizar productos usables desde el primer momento de la integración, controlando fallos de interoperabilidad de manera temprana y preparar procesos contingentes para el siguiente recorrido del ciclo de vida del sistema.

- Definir el estándar de programación de la aplicación y de la programación dentro de la base de datos (paquetería, procedimientos almacenados).
- Parametrización de formatos a usar para definir localidades, empresas, oficinas, que son datos almacenados en común para todos los módulos.
- Permitir a los módulos definir reglas y políticas por ejemplo: Un líder de proyecto (Módulo # 7) no podrá trabajar menos de 50 horas semanales.
- Realizar las tablas referenciales del Modelo Entidad - Relación que son usadas en común por los módulos del Sistema, por ejemplo la tabla ciudades (UIO, GYE,...), estados (Activo, Inactivo, Aprobado, etc.),
- Determinar y organizar un cronograma para los periodos de prueba de la integración que los demás grupos deberán cumplir puntualmente (mínimo 1 persona por grupo) para verificar la usabilidad de cada uno de sus módulos dentro de la integración al API General.
- Establecer políticas dentro del grupo de trabajo determinando: fechas máximas de entregas, formas de envíos, medios de comunicación a utilizar, responsabilidades a su cargo, consecuencias de retrasos, etc.

- Organizar reuniones periódicas y documentación de las actividades realizadas a lo largo del proyecto.
- Monitorear los riesgos que afectan al proyecto, ser proactivos y sugerir planes de contingencia.
- Revisar el cumplimiento de las actividades encomendadas por parte del grupo integrador a los líderes de los demás módulos.
- Aplicar correctas metodologías para realizar el control de versiones de los cambios a realizarse en Oracle y en la aplicación.

1.10. Arquitectura

SGPI utilizará arquitectura **Tres Capas**, donde se diseñaran los objetos del negocio, reglas de negocio y la interfaz del usuario.

Pudiendo cada una de estas áreas estar separadas en diferentes máquinas para optimizar su rendimiento y aprovechar su individualidad.

La **Figura 1-1** muestra la arquitectura de SGPI, donde las páginas jsp contendrán las diferentes clases y el desarrollo de la programación de la aplicación (reglas del negocio), que serán levantadas en un servidor de

aplicaciones y este mantendrán la comunicación con la Base de Datos Oracle. (Objetos del negocio). El servidor de aplicaciones puede ser cualquiera de los detallados como JBoss, IAS, pero en nuestro caso utilizaremos el *webserver* Tomcat como nuestro servidor de aplicaciones ¹

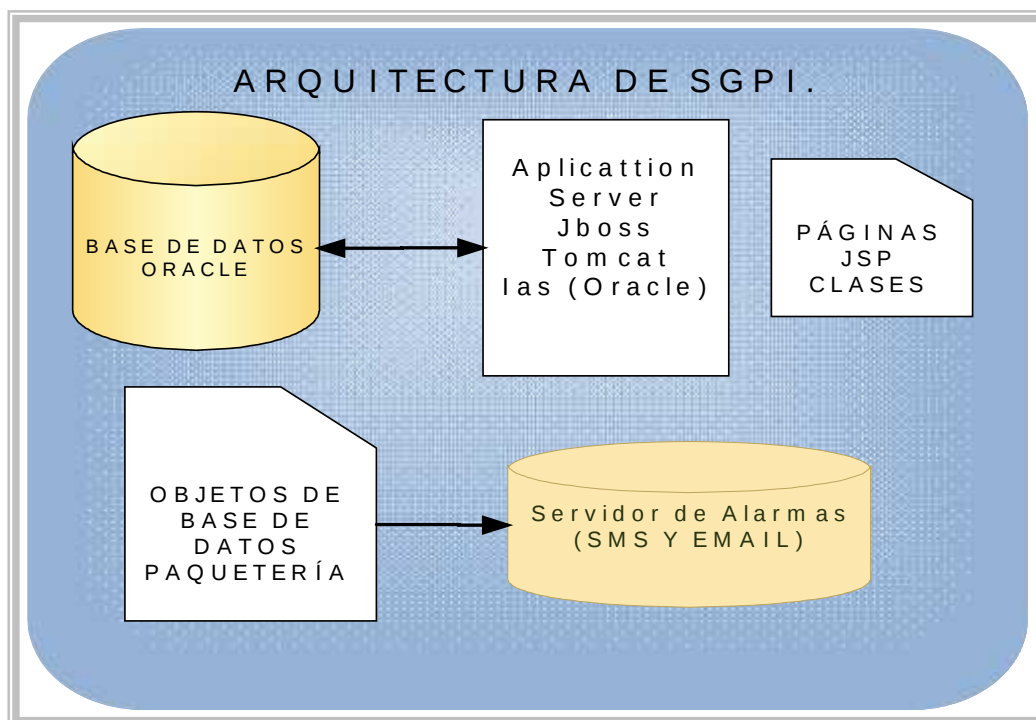


Figura 1-1.
Arquitectura de Sistema Gestor de Proyectos Informáticos.

SGPI utilizará arquitectura **Tres Capas**, donde se diseñaran los objetos del negocio, reglas de negocio y la interfaz del usuario. Pudiendo cada una de estas áreas estar separadas en diferentes máquinas para optimizar su rendimiento y aprovechar su individualidad.

1. Tomcat es un servidor web con soporte de servlets y JSPs, procesando servlets y jsp, (Tomado de <http://es.wikipedia.org/wiki/Tomcat>)

Los objetos de base de datos y paquetería Oracle hace referencia a los datos almacenados en la base y programación generadas en paquetes que tienen comportamiento y uno de ellos es el envío de datos al Servidor de Alarmas para que este gestione la entrega de alarmas por los diferentes medios de comunicación como emails y sms.

1.11. Metodología

El módulo de Integración, Parametrización y Configuraciones utilizará el modelo de ciclo evolutivo incremental debido al origen y finalidad de nuestro módulo, la integración. **(Ver Figura 1-2)**. En este ciclo de vida cada secuencia que se haga del proceso se tendrá un incremento y además tendremos un producto que gradualmente va aumentando su operatibilidad y compenetración.

Este ciclo nos permitirá tener en rangos de tiempo no muy dispersos productos funcionales asegurando así la detección de problemas de manera temprana, facilitando la integración final, ya que esta solo será un incremento n de nuestro ciclo de vida. Permittiéndonos no esperar tener todos los procesos terminados de los doce módulos para comenzar a realizar las operaciones de integración y la corrección de errores que se susciten en los incrementos graduales.

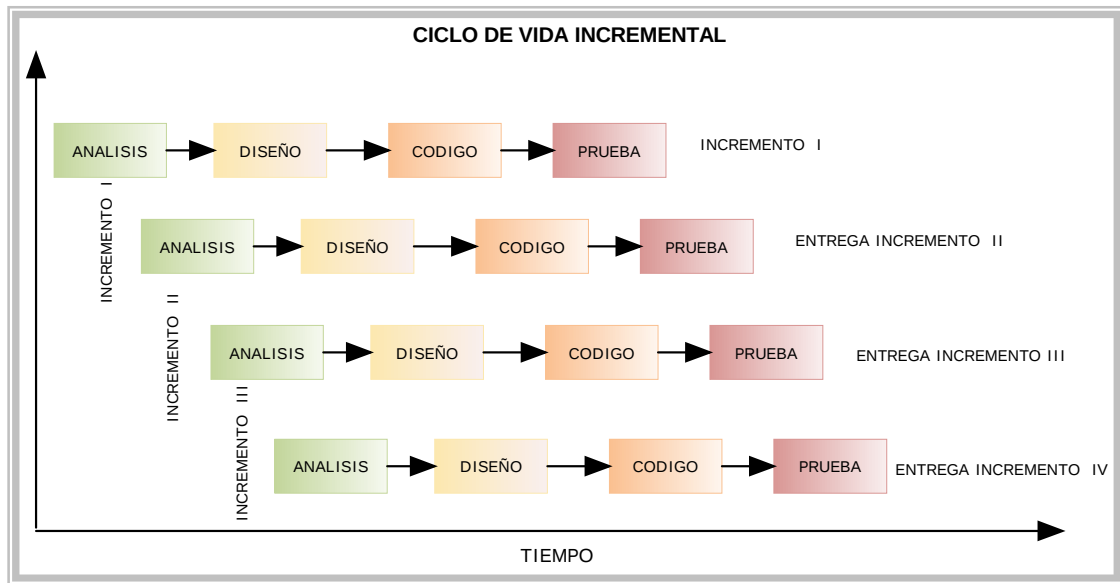


Figura 1-2.
Modelo de ciclo de vida Incremental

A continuación en la **Figura 1-3** se muestra el modelo incremental adaptado a nuestro fin; la integración de los diferentes módulos que conforman SGPI.

- a) Procesos, subprocesos operables funcionales de cada uno de los módulos los cuales se nos serán entregados.
- b) Se realizará la integración de los subprocesos al API general, para evaluar el cumplimiento de los estándares definidos. Si se encuentra alguna irregularidad se les informará a los dueños del módulo para corregir el problema y si no, se le dará una observación a favor para que la producción continúe siguiendo el mismo lineamiento.

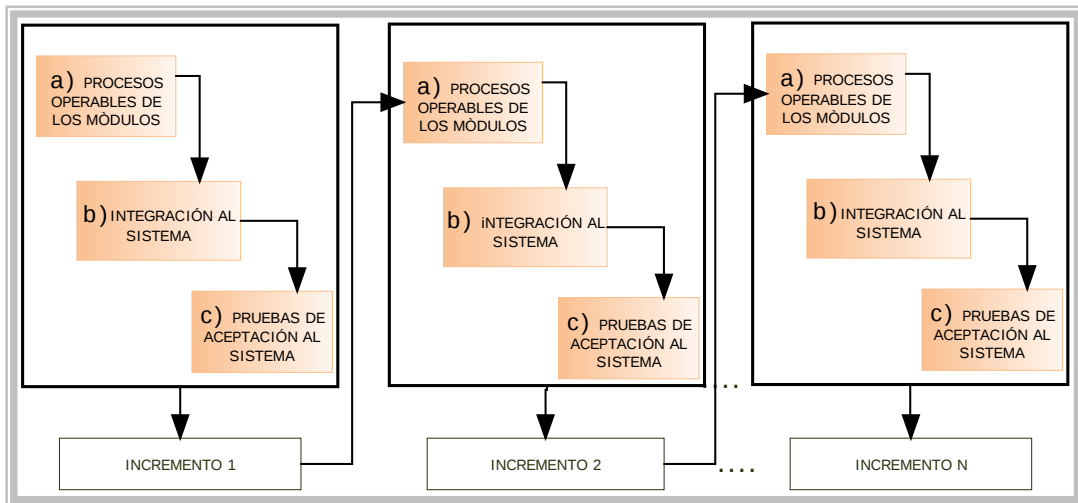


Figura 1-3.
Algoritmo de integración incremental.

c) Finalizado el paso anterior se comprobará mediante las pruebas de aceptación si:

- El sistema puede soportar los volúmenes de información.
- Los tiempos de respuestas cumplen las expectativas

Esta prueba tendrá el objetivo de que cada uno de los grupos valide si su módulo funciona como lo tenían previsto (una vez ya integrado).

1.12. Planificación de Actividades y Cronograma

Realizar la planeación debe ser una de las prioridades al iniciar un proyecto, después de que la idea ha sido concebida y asimilada, realizar una planificación con su respectivo cronograma garantizando un trabajo formal y la aplicación de estrategias para alcanzar nuestras metas.

La planeación se realizó en base al cronograma establecido al comienzo del curso, ajustándolo al horario del equipo de desarrollo de SGPI. En la siguiente tabla observamos el resumen de la planificación, sin tomar en cuenta tiempos de holguras ni posibles prorrogas.

En la primera columna encontramos el número de la actividad , en la segunda columna la descripción, aquellas que se encuentran dentro de un marco azul representa una etapa específica del desarrollo del proyecto y las actividades que le corresponden a esta etapa se listan debajo de ella. Las descripciones que tienen este símbolo  representan a los hitos del proyecto y ➤ representa una actividad, en la última columna el tiempo esta en días aunque en los subtotales se expresa en meses.

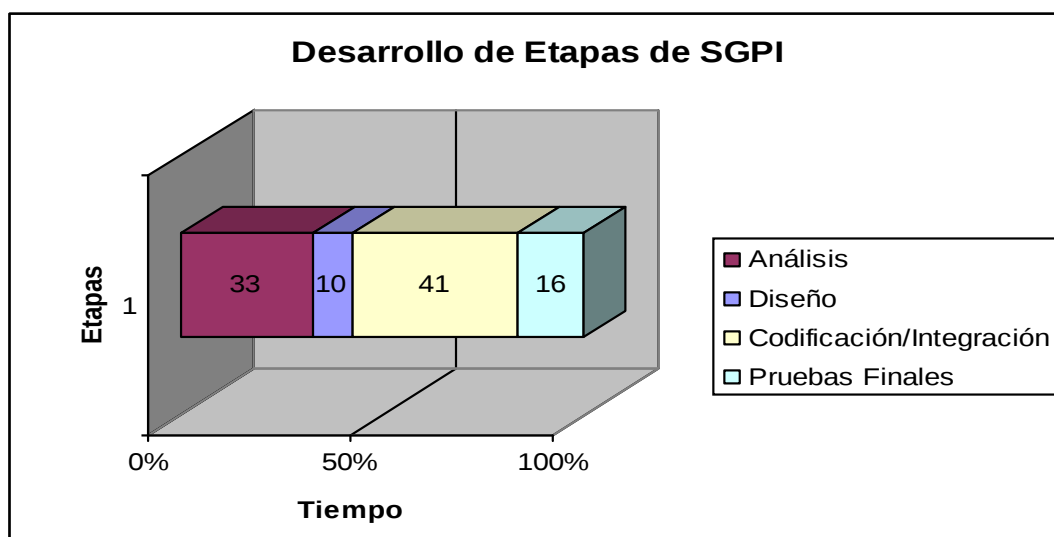


Figura 1-4
Gráfico de Etapas de desarrollo de SGPI

° ACTIVIDAD	DESCRIPCIÓN DE ACTIVIDAD	TIEMPO (DÍAS) DE CADA ACTIVIDAD
0	ANÁLISIS	80
1	DETERMINACIÓN DE OBJETIVOS.	3
1.1	➤ Reuniones con Sistei Solutions	1
1.2	➤ Objetivos Generales y Específicos	3
2	ALCANCE DEL PROYECTO.	4
3	RESTRICCIONES Y LIMITACIONES DEL PROYECTO.	1
4	RIESGOS DEL PROYECTO.	5
4.1	➤ Identificación del riesgo.	1
4.2	➤ Interpretación del Riesgo.	1
4.3	➤ Probabilidad e impacto.	1
4.4	➤ Tabla de riesgo.	1
4.5	➤ Plan de contingencia del riesgo.	5
5	REVISIÓN DE ALCANCES	11
5.1	➤ Entrevista con Sistei Solutions	1
5.2	➤ Revisión y Modificación de Alcances	10
6	ORGANIZACIÓN DEL PERSONAL.	2
6.1	➤ Definir Políticas Grupales	2
6.2	➤ Definir Responsabilidades de cada módulo	2
7	PLANIFICACIÓN TEMPORAL DEL PROYECTO.	4
7.1	➤ Plan temporal de tareas.	3
7.2	➤ Diagrama de gantt del avance del proyecto.	1
8	PRIMERA INTEGRACIÓN DEL MER.	7
8.1	➤ Elaboración del MER del módulo 6 (Parametrizaciones)	3
8.2	➤ Reuniones entre los 12 módulos	4
8.3	➤ Desarrollo del Diccionario de Datos	3
9	RECURSOS DEL PROYECTO.	2
9.1	➤ Recurso humano.	1
9.2	➤ Recurso de hardware/ infraestructura	1
9.3	➤ Recurso de software.	2
10	SEGUNDA INTEGRACIÓN DEL MER.	5
10.1	➤ Reuniones entre los 12 módulos	5
10.2	➤ Desarrollo del Diccionario de Datos	5
11	ANÁLISIS DE INFORMACIÓN.	12
11.1	➤ Diagrama de flujo de información del proyecto.	3
11.2	➤ Identificación de los requerimientos del sistema.	8
11.3	➤ Desarrollo del Diagrama de Casos de uso	4
12	TERCERA INTEGRACIÓN DEL MER.	24
12.1	➤ Reuniones entre los 12 módulos	18
12.2	➤ Desarrollo del Diccionario de Datos	12
12.3	➤ Entrega del MER 1.0	1
14.1	➤ Realización de los Scripts del modelado de datos	2
14.2	➤ Realización de los Scripts del Manejo de Usuarios y Privilegios	6
14	DISEÑO.	25
14.3	➤ Diagrama de componentes principales.	6
14.4	➤ Diseño de Interfaces.	10
15	PRIMER ENTREGA DEL PROYECTO	
	Total en meses	3.5

Tabla 1-1

Planificación Etapas de Análisis y Diseño.

LISTA DE ACTIVIDAD	DESCRIPCIÓN DE ACTIVIDAD	TIEMPO (DÍAS) DE CADA ACTIVIDAD
16	❏ CÓDIGO/ INTEGRACION	100
16.1	➤ Diseño y Construcción del Primer Modelo	25
16.2	➤ Pruebas y Retroalimentación	25
16.3	➤ Construcción del Modelo Final.	30
16.4	➤ Análisis de Riesgo.	2
16.5	➤ Integración de los 11 módulos	10
16.6	❏ MODIFICACIONES EN OBJETOS DE NEGOCIOS	10
17	❏ SEGUNDA ENTREGA DEL PROYECTO	
	Total en meses	3.3
18	❏ PRUEBAS/INTEGRACION	40
18.1	➤ Pruebas del sistema con datos reales.	5
18.2	➤ Formulación de Recomendaciones.	5
18.3	➤ Evaluación de la primera construcción.	2
18.4	➤ Revisión, Pruebas y adaptaciones de la integración	30
21	❏ TERCERA ENTREGA DEL PROYECTO.	
	Total en meses	1.3
22	➤ Pruebas de Flujo del Sistema	15
22.1	➤ Ajustes del Flujo de Sistema	6
22.2	➤ Integración de módulo de alarmas	5
22.3	➤ Pruebas Alfa Finales del sistema	5
23	❏ CUARTA ENTREGA DEL PROYECTO	
	Total en meses	0.5
Tiempo de desarrollo del proyecto:		8.1 meses

Tabla 1-2

Planificación Etapas de Codificación, Integración y Pruebas Finales

Dentro de la planeación se ha contemplado las pruebas de integración incluidas en el 41% de la etapa de Codificación e Integración, y un 16% a pruebas finales y seguimientos de flujo de información como podemos observar en la **Figura 1-4**, a la etapa de diseño fue a la que menos tiempo se le dedico. Al final se debe comparar la planificación con la ejecución.

1.13. Recursos Del Proyecto

La planificación de los recursos de un proyecto previstos con anticipación es de gran ayuda porque tendremos documentación que indique y detalle cuáles son los recursos necesarios para las actividades que se presentarán a lo largo del proyecto sin mencionar que es el elemento principal para estimar los costos del proyecto final.

Para el desarrollo de SGPI se necesitarán recursos humanos, hardware, software, infraestructura, entre otros. En los siguientes enunciados se detallarán los recursos que se necesitarán para cumplir con los objetivos y alcances mencionados.

1.13.1. Recurso Humano

El grupo de Integración, Configuración y Parametrizaciones está conformado por 3 personas que desempeñarán las siguientes actividades principales.

- Administradores de la Base de Datos
- Gestor de Proyectos de Software
- Control de Cambios y Versiones de la Base de Datos y la aplicación.

Se necesita que el grupo encargado de este módulo tenga conocimientos acerca de la Administración de Bases de Datos Oracle, PL/SQL, lenguaje

java, administración de recursos humanos, gestión de proyectos entre otros. Debe tener un alto desempeño en sus actividades y propiciar la comunicación entre los módulos del sistema SGPI.

El grupo total de personas involucradas en el desarrollo de este proyecto se detallan en la tabla 1-3. Aunque en páginas anteriores ya hemos descrito la funcionalidad de cada módulo, mostramos en forma de resumen el nombre del grupo responsable, la descripción de su ámbito de trabajo y el número de integrantes que conforman el grupo.

Nombre	Descripción	# de Integrantes
Grupo 1	Seguridades	3
Grupo 2	Requerimientos	3
Grupo 3	Pre - Proyecto	3
Grupo 4	Control de Hitos	3
Grupo 5	Documentación	3
Grupo 6	Integración	3
Grupo 7	Líderes y Colaboradores	3
Grupo 8	Reportes Gerenciales	3
Grupo 9	Costos y Presupuestos	2
Grupo 10	Alarmas	2
Grupo 11	Contratos	2
Grupo 12	Infraestructura	1
Total		31

Tabla 1-3
Recursos Humanos de SGPI

1.13.2. Recursos Hardware

Para el desarrollo del módulo de Integración, Parametrización Y Configuraciones se necesitará equipos para el servidor de base de datos y el servidor de aplicaciones, aunque podríamos tener los dos esquemas en un mismo equipo, pero de manera preferente ponemos cada capa en hardware separado por motivos de seguridad y confiabilidad.

EQUIPO	DESCRIPCION
	Requerimientos mínimos [2]
Servidor de Oracle 1 unidad	Protocolo TCP/IP
	Pentium 300 o compatible
	256 Mb de memoria RAM
	Al menos 200 Mb de memoria virtual mínima y 400Mb de máxima
	Monitor VGA de 256 colores
	140 MB en el disco duro del sistema (normalmente C) y otras 300 MB más en ese mismo disco para el software de Administrador y otras herramientas de Oracle.
	2,85 GB en el disco que albergará la carpeta raíz de Oracle, el llamado Oracle Home, 1 GB más si se usa el Oracle Management Server y otros 2,35 GB más si se usa el Oracle Internet Directory. Las herramientas de administrador requieren 750 MB y las de ejecución (Runtime) otras 300 MB.
	Además hay que tener en cuenta que los tamaños de disco indicados sólo son válidos si se usan discos duros con el sistema NTFS, si es FAT32 hay que doblar el tamaño (llegando a las 13 GB con instalación absoluta).
	Requerimientos utilizados
	Pentium IV de 3.2 Ghz. 2 Gb de memoria RAM 160 Gb de Disco Duro Los demás requerimientos son proporcionales a los detallados en los mínimos.
Servidor de Aplicaciones 1 unidad	Requerimientos utilizados Protocolo TCP/IP Pentium IV de 3.2 Ghz. 2 Gb de memoria RAM 80 Gb de Disco Duro
	Requerimientos mínimos Procesador Pentium 200 Mhz Memoria RAM de 128 Mb
Cliente 12 unidades	Requerimientos utilizados Procesador Pentium IV de 3.2 Ghz. 1 Gb de memoria RAM 60 Gb de disco duro

Tabla 1-4.
Recursos Hardware.

1.13.3. Recursos Software

En el desarrollo de SGPI, necesitaremos los siguientes software instalados en nuestro hardware, usaremos preferentemente Windows 2003 Server Professional básicamente el Sistema utilizará tecnología Oracle y Java es así que se ha desglosado los principales componentes, obviando software de apoyo como, procesadores de texto, editor de gráficos, diseñadores de software, porque no influye directamente con el desempeño de SGPI, así que pueden ser de cualquier fabricante y cualquier versión que convenga. Cualquier otro elemento o componente adicional importante se lo detallará en el manual técnico.

EQUIPO	DESCRIPCION
SERVIDORES	Sistema Operativo Windows 2003 Server de Microsoft Base de Datos Oracle 9i release 2 Enterprise edition. Es una base de datos robusta, segura confiable, multiplataforma, además incluye un conjunto de opciones que permiten escalar a nivel de software sus funcionalidades. PLSQL Developer 7.0 PLSQL Developer es un ambiente integrado de desarrollo que esta enfocado al desarrollo de programas de unidades de almacenamiento para Base de Datos Oracle.
SERVIDORES	IDE Eclipse 3.5, My Eclipse 4, Máquina Virtual Java JDK 1.5 Tomcat 5.0 como Servidor de Aplicaciones
EQUIPOS DE DESARROLLO	Base de Datos Oracle 9i release 2 Enterprise edition. Es una base de datos Sistema Operativo Windows XP SP 2 de Microsoft PLSQL Developer 7.0 IDE Eclipse 3.5, My Eclipse 4, Máquina Virtual Java JDK 1.5

Tabla 1-5.
Recursos Software.

1.13.4. RECURSOS INFRAESTRUCTURA

La infraestructura necesaria para el desarrollo de SGPI está listada de la siguiente manera:

- Comunicaciones: Un laboratorio de computación que cuente con red cableada de 100 Mbps o inalámbrica IEEE 802.11b 11 Mbps.
- Sistema Central: Servidor de Oracle
- Periféricos : Computadores Clientes

1.14. Estudio Estadístico

Es importante recopilar datos estadísticos y las tendencias en cuanto tenga que ver con el desarrollo del software, en este caso específico la integración del software. Según estudios realizados en la Ciudad de Guayaquil y Quito se ha determinado 3 clases de proyectos: pequeños (1 – 6 meses), medianos (7 – 15 meses) y los grandes que tienen un tiempo de culminación mayor los 15 meses, obteniendo las siguiente estadísticas en cuanto a la producción del Software. [1]

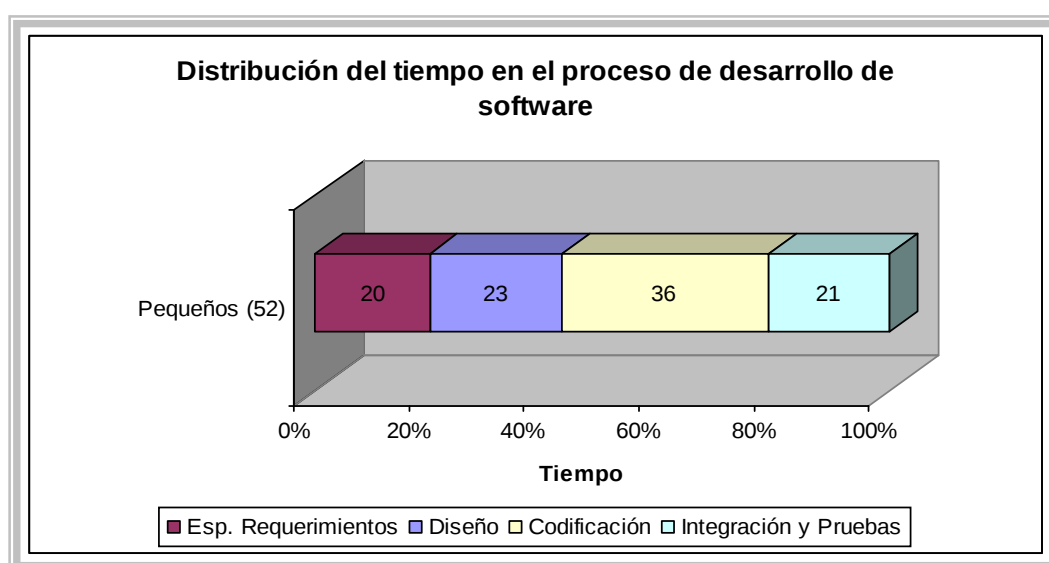


Figura 1-4.

Distribución del tiempo en el proceso de desarrollo del software

Según cronograma hemos determinado que nuestro proyecto es pequeño por el determinante tiempo, asumiendo su complejidad implícitamente.

Como la **Figura 1-4** indica en proyectos pequeños el tiempo esta determinado de la siguiente forma, el 20 % del tiempo del desarrollo del proyecto se lo ocupa en la especificación de requerimientos, el 23% al diseño, el 36% a codificación y el 21% a la Integración y Pruebas, todo esto con una muestra de 52 empresas desarrolladoras de software de Quito y Guayaquil [1].

Hemos considerado este factor estadístico como referencia para plantearnos el tiempo estimado a cada uno de las fases del desarrollo de SGPI, así como un punto referencial para el control de tareas en cada una de las fases.

Es de esta forma que podemos notar que la integración no nos debería tomar más allá de un 30% como tope máximo, siempre y cuando los rangos de las otras fases (Análisis, Diseño y Codificación) no sobrepasen las tendencias mostradas.

Para controlar estos tiempos referenciales es necesaria la Gestión de Proyectos del cual el módulo de Integración, Parametrización y Configuraciones es también responsable, y que se lo conseguirá con los puntos detallados en los objetivos y alcances del proyecto.

1.15. Administración Del Riesgo

La administración de riesgos se preocupa de identificar los riesgos y figurar planes para minimizar su efecto sobre el proyecto. Teniendo en cuenta que un riesgo es la probabilidad de que una circunstancia adversa se presente. Vamos a analizar los riesgos de proyecto que afectan la calendarización o recursos y los riesgos de producto afectan la calidad o desempeño del software que se está desarrollando.

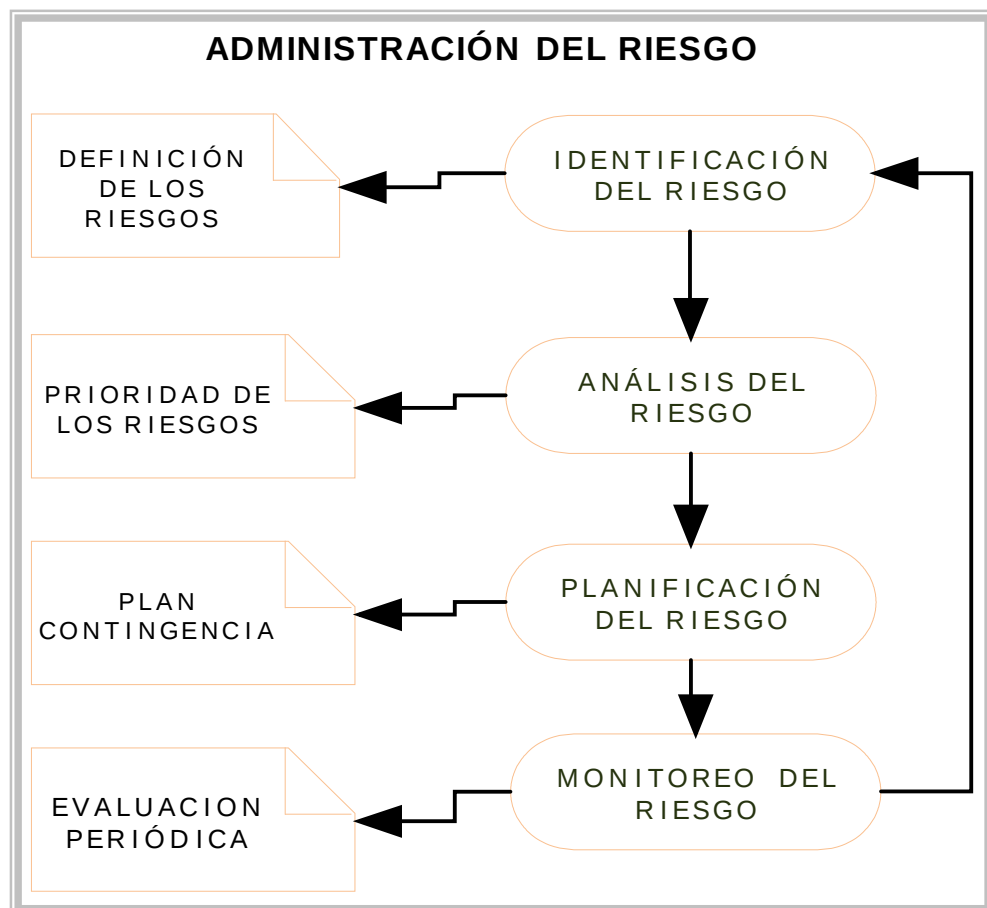


Figura 1-5.
Administración del Riesgo.

1.15.1. Identificación Del Riesgo

Se ha realizado un análisis sobre los principales riesgos de SGPI que afectan directamente al módulo de Integración Parametrización y Configuraciones.

1.15.1.1. Definiciones De Los Riesgos

- **Equipos de Computación no estén disponibles.**

Contamos con un laboratorio destinado para la preparación y capacitación de los cursantes del Seminario de Graduación, pero no está disponible en el horario libre de los participantes de SGPI, y las demás horas que están libres el 94% de los participantes está trabajando. Además no existe otro lugar que se haya proporcionado equipos computacionales para el desarrollo del software.

- **Daños Imprevistos de Hardware**

El laboratorio de computación asignado para el desarrollo de SGPI, es además usado por otros dos cursos de Seminario de Graduación, al no tener una responsabilidad exclusiva sobre el mantenimiento de los equipos se corre el riesgo de no darle un adecuado cuidado dependiendo del fin con el que están siendo utilizados. Podría darse un mal uso de software sobre hardware ocasionando pérdida de información.

- **Falta de backups de los módulos desarrollados y de la base de datos.**

Al hablar de la falta de backup nos referimos a los respaldos de los módulos de programación y de la base de datos que se encuentran avanzados y que podrían almacenarse en medios de almacenamiento portátiles para poder llevarlos a reuniones con el resto del equipo, para revisiones de calidad o de los avances.

- **Reclutamiento de Personal Aleatorio**

Proceso de reclutamiento inexistente, es un grupo de 31 personas reunidas sin selección alguna. Donde el patrón aleatorio es haberse matriculado en el Paralelo “A”.

- **Asignación de roles aleatorio.**

Evaluaciones de destrezas, habilidades o experiencia inexistentes. Los módulos fueron asignados por sorteo.

- **Capacitación Inadecuada**

El contenido de los módulos del seminario podrán ser inadecuados para el fin definido, o el contenido a desarrollar dentro de cada curso podría no contener información relevante para el proyecto.

- **Rendimiento no óptimo en el desarrollo de los cursos de capacitación.**

Los participantes podrán no explotar el contenido de los cursos u obtener un bajo rendimiento de aprendizaje.

- **Indisponibilidad de reunirse el grupo de desarrolladores**

El personal de desarrollo de SGPI podría no tener horarios uniformes o regulares para reuniones, desmejorando la comunicación, que es el componente más importante cuando se tiene un grupo de desarrollo numeroso.

- **Indisponibilidad de reunirse con el grupo integrador**

Las reuniones para integrar los diferentes módulos al API principal son un punto crítico, cada grupo debe estar presente porque su módulo puede verse afectado en cada iteración de cambio.

- **Modificaciones de último momento por parte de la empresa que facilitó los requerimientos del software.**

Falta de reuniones previas con la empresa que facilitó la documentación acerca de SGPI, aumentan el riesgo de modificaciones en el momento de codificación, ya que los coordinadores al revisar prototipos podrían no estar satisfechos con el producto.

- **Mala interpretación de los requerimientos por parte de los desarrolladores.**

Los desarrolladores podrían tener problemas al interpretar los requerimientos de diversas formas y cuando se deba relacionarlos para que hagan un proceso en común cada parte tendrá una interpretación diferente de lo que necesitaban como requerimiento.

- **Tiempo muy limitado para la planificación y recopilación de requerimientos.**

El tiempo de recopilación de requerimientos, especificaciones y análisis de SGPI podría ser muy corto y apresurado, omitiendo diversos puntos por la premura del tiempo de presentación

- **Ausencia de investigación, entrevistas a empresas que tiene áreas de sistemas.**

Ausencia de autogestión en la búsqueda de información referente a SGPI, material como entrevistas y exploración del campo de trabajo en este caso el área de sistemas de la empresa y la metodología con la cual trabajan.

- **Crear la idea la integración es la Panacea**

Los responsables de los módulos podrían llegar a pensar que si algo no funciona correctamente no es necesario corregirlo en el instante que lo detectan ya que al momento de la integración lo harán funcionar correctamente.

- **Posponer la integración para cuando todo este terminado**

Esperar que lo módulos estén terminados para *ahora si* empezar la integración, corriendo el riesgo que el proyecto se retrase.

1.15.2. Análisis De Riesgos

Se calcula la exposición a pérdidas de cada riesgo. Básicamente se pretende multiplicar la probabilidad de que se presente un riesgo por la pérdida que ocasionaría si se presenta. Esta puede medirse en tiempo de retraso o en costos adicionales ocasionados

El impacto y al probabilidad están definidos numéricamente por la siguiente tabla, se usará indistintamente su valor numérico o el nombre.

En Anexos usted puede encontrar la tabla completa de la evaluación de los riesgos.

	Cualificación (adjetivo)	Cuantificación (valor)
IMPACTO	Catastrófico	3
	Serio	2
	Tolerable	1
	Insignificante	0
	Cualificación (adjetivo)	Cuantificación (valor)
PROBABILIDAD	Alta	3
	Media	2
	Baja	
	Improbable	0

Tabla 1- 6.
Impacto y Probabilidades del Riesgo

1.15.2.1. Impacto y Probabilidad Del Riesgo

- **Equipos de Computación no estén disponibles.**

Impacto.- Serio, si no tenemos equipos disponibles nuestro tiempo de desarrollo y avance se ve afectado produciendo retraso en las actividades planificadas en forma grupal.

Probabilidad.- Una probabilidad alta porque no se posee infraestructura para mejorar la situación descrita.

- **Daños Improvistos de Hardware**

Impacto.- El daño del hardware o equipo de desarrollo tiene un impacto mínimo, ya que se cuentan con equipos suficientes para tenerlos como salvaguardas.

Probabilidad.- Se le ha asignado una probabilidad baja ya que existen particiones de discos y sesiones por cada usuario, además los otros usuarios con que se comparten los equipos son conocedores de informática.

- **Falta de realizaciones de *backups***

Impacto.- Un impacto crítico sería debido a que una falta de respaldo en un medio de almacenamiento portable no permitiría que fuéramos poco a poco uniendo los módulos de desarrollo para realizar proezas, ajustes y modificaciones necesarias retrasando lo que sería la unión de módulos y pruebas.

Probabilidad.- Una probabilidad baja debido a que normalmente los discos duros no implican un daño del equipo, además disponemos de medios de almacenamiento portables.

- **Reclutamiento de Personal Aleatorio**

Impacto.- Impacto tolerable ya que aunque no exista previa selección, todos los integrantes son egresados de la Carrera de Ingeniería en Sistemas Computacionales.

Probabilidad.- Media por la forma de reclutar a los estudiantes para los cursos de graduación, sin orientación o pruebas previas.

- **Asignación de roles aleatorio.**

Impacto.- Serio debido a la metodología de asignar las diferentes actividades a desarrollar dentro del proyecto.

El sorteo implica una asignación aleatoria de roles sin tomar en cuenta si hay personal más entrenado o capacitado que otro para desenvolver un determinado papel dentro del proyecto.

Probabilidad.- Alta debido a que es el método que se utiliza para la asignación de tareas.

- **Capacitación Inadecuada**

Impacto.- Impacto de magnitud seria, si el contenido de los cursos no están de acordes al desarrollo del proyecto, se invertiría tiempo en la asistencia, mermando el tiempo para la realización del proyecto.

Probabilidad.- Baja, ya que la Universidad se ha encargado de seleccionar a personal serio y con experiencia en el área.

- **Rendimiento no óptimo en el desarrollo de los cursos de capacitación.**

Impacto.- Un impacto alto ya que los participantes podrían no tener una buena formación académica previa para desenvolverse con satisfacción y excelencia en el desarrollo de los cursos, provocando una disminución en la productividad en el desarrollo del proyecto.

Probabilidad.- Se ha determinado una probabilidad baja, porque se ha notado progresos en la capacitación en curso.

- **Indisponibilidad de reunirse el grupo de desarrolladores**

Impacto.- Alto ya que se está usando la división del sistema en módulos que tienen interoperabilidad entre ellos y si los desarrolladores no pueden reunirse para pulir sus fusiones y relaciones conllevaría retraso en el cronograma y deficiencias en nivel de acoplación.

Probabilidad.- Alta debido a los horarios de reunión de los diferentes líderes de proyecto, la coordinación de día y hora se vuelve ardua.

- **Indisponibilidad de reunirse con el grupo integrador**

Impacto.- Alto porque es el grupo integrador el responsable de la funcionalidad de la aplicación final entregable, si los demás módulos no se reúnen con el grupo integrador no habrá funcionalidad en el producto final.

Probabilidad.- Se define una probabilidad media ya que el grupo integrador no ha tenido problemas por sus horarios de trabajo.

- **Modificaciones de último momento por parte de la empresa que facilitó los requerimientos del software.**

Impacto.- Alto debido a la importancia de los requerimientos y que la recolección de los mismos son los fundamentos para realizar el análisis y diseño y si esto llegará a estar mal especificado conllevaría al efecto bola de nieve, que indica que mientras los errores sean desde los puntos orígenes tendrá más repercusión en el esfuerzo – coste proyecto.

Probabilidad.- Media, solo ha habido pocas reuniones con la empresa y poco tiempo para hacer aclaraciones.

- **Mala interpretación de los requerimientos por parte de los desarrolladores.**

- Impacto.- Alto el cronograma de producción se retrasará hasta que los desarrolladores vuelvan a plantearse nuevamente el problema en un punto donde el proyecto ya este en su etapa de codificación.

Probabilidad.- Baja, los desarrolladores han tenido múltiples reuniones para definir los requerimientos llegando a acuerdos claros.

- **Tiempo muy limitado para la planificación y recopilación de requerimientos.**

Impacto.- Alto, realizar un análisis de un proyecto que va a durar 9 meses implica dar por lo menos un 20% de este tiempo a la especificación de requerimientos, además el factor inexperiencia es un hoyo negro donde podemos caer al obviar puntos importantes dentro de nuestra planificación.

Probabilidad.- Alta, el tiempo propuesto para la entrega de recopilación se ve afectado por la falta de tiempo de los participantes.

- **Ausencia de investigación, entrevistas a empresas que tiene áreas de sistemas.**

Impacto.- Alto, tiene que ver con la recopilación de requerimientos también falta de autogestión, que recae directamente sobre el éxito del desarrollo del producto.

Probabilidad.- Media, el grupo ha conversado con algunos gerentes de software, más no ha documentado las entrevistas.

- **Crear la idea que Integración es la Panacea**

Impacto.- Alto porque no se definen los procesos individualmente sino que se va a esperar un punto crítico como la integración para comenzar a hacerlo.

Probabilidad.- Media, en el transcurso del proyecto el 50% de los grupos nos transmitió esa idea.

- **Posponer la integración para cuando todo este terminado**

Impacto.- Alto, retrasaría la entrega del proyecto y se deberá aumentar recursos de tiempo, de esfuerzo y coordinación

Probabilidad.- Baja, por el modelo incremental que se utilizará para integrar los módulos.

1.15.3. Planificación del riesgo.

Una vez que los riesgos han sido identificados claramente proponemos una planificación para aplacar el impacto o probabilidad que el riesgo pueda generar.

Dentro de la planificación de riesgo tenemos estrategias evasivas, que son aquellas que reducen la probabilidad de que el riesgo ocurra, estrategias de minimización que reducen el impacto y un plan de contingencia para no vernos afectado cuando el riesgo ocurra.

- **Equipos de Computación no estén disponibles.**

Evasiva.- Adecuar pequeños espacios dentro de la universidad con equipos necesarios.

Minimización.- Organizar el cronograma de tal forma que mermemos el tiempo de reunión conjunta, para realizar la mayor parte del trabajo en lugares individuales.

Plan de Contingencia.- Organizar pequeños lugares de trabajo dentro o fuera de las instalaciones Universitarias con equipos propios de los desarrolladores.

- **Daños Improvistos de Hardware**

Evasiva.- Dando mantenimiento preventivo a los equipos.

Minimización.- Tener equipos de reservas para suplir cualquier eventualidad.

Plan de Contingencia.- Compartir recursos físicos entre los desarrolladores y tener secuencias de trabajo, por ejemplo una persona podría necesita la computadora sólo de 8 a 9 de la mañana y otro a partir de las 9.

- **Falta de realizaciones de *backups***

Evasiva.- Establecer políticas de respaldos como el tiempo en que un respaldo debe ser hecho o las circunstancias que demanden hacerlo.

Minimización.- Realizar respaldo con mayor frecuencia para que no existan brechas demasiadas amplias de cambios sin *backups*.

Plan de Contingencia.- Tener siempre disponible de manera inmediata los últimos backups para realizar una recuperación desde la últimas modificaciones almacenadas en el respaldo.

▪ **Reclutamiento de Personal Aleatorio**

Evasiva.- Realizar reuniones previas a la inscripción en los cursos de Seminario para que los participantes reciban orientación e inducción acerca del contenido del curso y del proyecto.

Minimización.- Cursos de capacitación relacionados con el proyecto a realizar (si se lo hace).

Plan de Contingencia.- Demandar a los participantes esfuerzo y dedicación dentro de la capacitación.

▪ **Asignación de roles aleatorio.**

Evasiva.- Presentar los ambientes de los módulos en que se va a dividir el proyecto, sus alcances, etc. Y proponer a los participantes que realicen una pre – asignación según ellos estimen conveniente a sus habilidades.

Minimización.- Propender a la auto educación y a la investigación para desarrollar con precisión la tarea asignada.

Plan de Contingencia.- Inducir al grupo de participantes a ser colaboradores y brindar ayuda con sus destrezas a aquellos que lo necesiten.

- **Rendimiento no óptimo en el desarrollo de los cursos de capacitación.**

Minimización.- Establecer diversas metodologías de estudios para ir al ritmo adecuado en el curso, cambiar hábitos de estudios.

Plan de Contingencia.- Realizar cursos que contengan ejercicios prácticos que profundicen en el conocimiento humano

- **Capacitación Inadecuada**

Evasión.- El personal encargado de la capacitación debería además de tener experiencia sobre el tema a tratar ser conocedor de pedagogía.

Minimización.- Entregar el material semanas antes que los cursos comiencen.

Plan de Contingencia.- Buscar otras fuentes de conocimientos alternas, como libros relacionados, revistas, Internet, etc.

- **Indisponibilidad de reunirse el grupo de desarrolladores**

- **Indisponibilidad de reunirse con el grupo integrador**

Evasión.- Definir en lo posible horarios de reunión en los días de fines de semana.

Minimización.- Cada grupo deberá designar un representante para que sea portavoz en las reuniones, y que es más fácil reunirse 12 personas que 31.

Plan de Contingencia.- Realizar una planeación de reuniones (día, hora) con anticipación que deberá ser aprobado o ajustado por cada uno de los grupos, mantener comunicación vía e-mail, Chat, etc., cualquier medio remoto.

- **Modificaciones de último momento por parte de la empresa que facilitó los requerimientos del software.**
- **Mala interpretación de los requerimientos por parte de los desarrolladores.**

Evasión.- Establecer desde el comienzo del curso citas, reuniones, entrevistas con la empresa y/o desarrolladores.

Minimización.- Usar metodologías de software que permitan la corrección de los requerimientos sin tener mucha repercusión en el desarrollo del proyecto, aunque ninguna de ellas evitan el retraso en el proyecto, hay algunas que facilitan el trabajo de la recopilación de información.

Plan de Contingencia.- Contactarse con la empresa que facilitó los requerimientos, establecer medios de comunicación y horarios para entrevistas o charlas con claras especificaciones de lo que se desea, presentar diagramas de flujo de datos, esquemas, prototipos para establecer los requerimientos de manera clara.

- **Tiempo muy limitado para la planificación y recopilación de requerimientos.**

Evasión.- Reorganizar el cronograma, asignando más tiempo a la especificación de requerimientos mínimo el 20% de la duración total del proyecto.

Plan de Contingencia.- Participantes investigativos, delegar funciones entre el grupo para realizar investigaciones de las especificaciones de requerimiento, cada grupo deberá investigar procesos específicos, realizar entrevistas, y hacer investigación de campo.

- **Ausencia de investigación, entrevistas a empresas que tiene áreas de sistemas.**

Evasión.- Tomar un mayor número de muestras de empresas de software para realizar entrevistas.

Minimización.- Proporcionar documentación detallada acerca del Dercas

Plan de Contingencia.- Realizar entrevistas al personal del departamento de sistemas de las empresas donde los participantes trabajan

- **Crear la idea que Integración es la Panacea**

Evasión.- Mostrar a los integrantes del grupo estadísticas acerca del desarrollo de software y explicar la importancia de refinar un producto antes de integrarlo.

Minimización.- En la fase de integración según el algoritmo incremental usado por el Grupo de integración, se evaluará los procesos y se le dará el visto bueno siempre y cuando sus procesos sean eficientes.

Plan de Contingencia.- Llamar a reunión a los grupos de SGPI, y mostrarles los lineamientos de codificación y depuración que están usando vs. los que deben usar según haya quedado definido en documentación de estándares previamente aprobada.

- **Posponer la integración para cuando todo este terminado**

Evasión.- Presentar al comienzo del desarrollo del proyecto el algoritmo incremental a ser usado para la integración de los módulos de SGPI.

Minimización.- Establecer un cronograma con fechas y horas límites de entrega de los proceso, paquetes de los módulos para que la integración no se atrase y sea una actividad trabajado en paralelo junto con los demás módulos.

Plan de Contingencia.- Determinar y estudiar un nuevo cronograma de trabajo con mayor exigencia y repercusiones sobre los responsables del módulo según sea el caso.

1.15.4. Monitoreo del riesgo.

Después que los riesgos hayan sido identificados, se deberá mencionar en reuniones periódicas el análisis del riesgo determinando si la probabilidades han aumentado o no. Es importante esta última etapa de la administración para cambiar las prioridades de los riesgos según transcurra el tiempo, ya que lo que al principio del proyecto pudo haber sido un riesgo de baja prioridad pudo convertirse en nuestra principal amenaza.

CAPITULO 2

2 ANÁLISIS

2.1 Análisis de Requisitos SRD (Documentación de Especificaciones de Requisitos)

2.1.1 Recopilación de Información

El requerimiento llega por parte del área Cliente (Mkt, Vtas, Sac, etc) a SISTEMAS y el Gerente de Cartera de Proyectos lo prioriza y lo asigna a un Project Manager, éste realiza un estudio de Factibilidad y el Cronograma del Proyecto y lo devuelve a la Gerencia de Cartera de Proyectos para su aprobación y asignación de Recursos; Una vez asignados los Recursos y aprobado el Cronograma se Inicia El Proyecto.

Durante el Desarrollo del Proyecto nos encontramos con las Etapas del Mismo (análisis, diseño, desarrollo, pruebas, capacitación, puesta en producción) y los

documentos (entregables) que se generan en cada etapa.

Los Project Manager son los encargados de gestionar dentro de las mismas áreas de sistemas el desarrollo del proyecto (Ej.: en caso de que se necesiten servidores, con el área de infraestructura). Asimismo es el realiza la gestión hacia los Clientes Internos (que enviaron el requerimiento). En Proyectos en conjunto con algún proveedor de Soluciones Informáticas realiza las gestiones de administración del proyecto con el Gerente de Proyecto (Project Manager) de la empresa que provee la solución

El Project manager es el responsable durante el desarrollo del proyecto, lidera los recursos humanos y entrega el proyecto a los clientes según lo requerido.

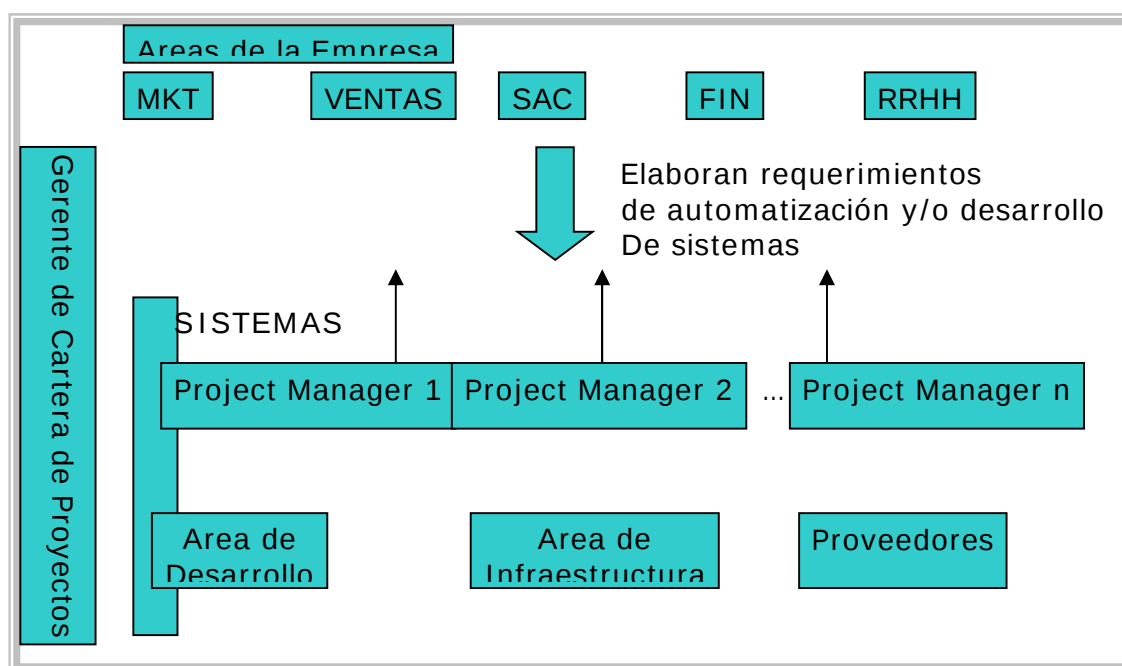


Figura 2-1.

Recopilación de Información

2.2. MODELO ENTIDAD RELACION

El modelo entidad relación (MER) se forma a partir de los 11 modelos formulados por los responsables de cada módulo, el módulo de Reportes Gerenciales ha sido obviado por la naturaleza de su estructura.

Grupo de Integración, aplicó técnicas de normalización cada vez que se realizaba una integración entre los MER individuales

2.3. DICCIONARIO DE DATOS

En el diccionario de datos se detalla cada una de las entidades con el prefijo **GEN** que son responsabilidad del Grupo de Integración.

El diccionario de datos lo encontramos detallado en los Anexos

2.4. DIAGRAMA DE FLUJO DE DATOS

El Diagrama de Flujo de Datos posee una visión general de las funciones y transformaciones de datos en una organización. Se lo puede llevar a una representación de un modelado lógico y gráfico del sistema.

El Diagrama de Flujo de Datos identifica entradas, salidas, procesos y relaciones con el exterior tanto a nivel grafico, por refinamiento como a nivel detallado.

2.4.1 DIAGRAMA DE FLUJO DE DATOS NIVEL 0

El Diagrama de Flujo de Datos en su nivel 0 también conocido como Diagrama de Contexto, consiste de los siguientes pasos:

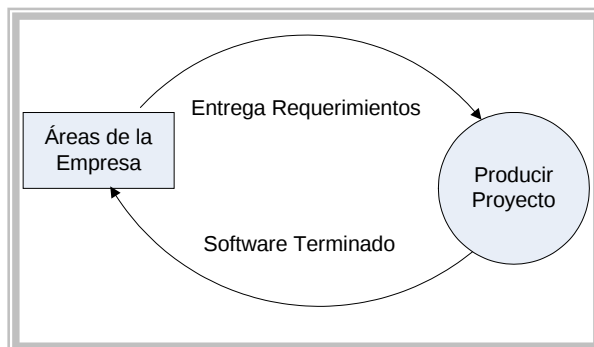


Figura 2-2.
Nivel 0. Diagrama de Contenidos

- 1 Un área de la empresa solicita un requerimiento al SGPI.
- 2 El SGPI gestiona la producción del sistema proveyendo un producto terminado (software) al área de la empresa que solicitó el requerimiento.

2.4.2. DIAGRAMA DE FLUJO DE DATOS NIVEL 1

El Diagrama de Flujo de Datos del SGPI en su nivel 1, consiste de los siguientes pasos:

- 1 Un área de la empresa solicita un requerimiento de un proyecto al SGPI.
- 2 El SGPI prioriza el requerimiento enviado por parte de un área de la empresa.
- 3 El SGPI asigna al Project Manager el requerimiento.
- 4 Se realiza el Plan Técnico del Proyecto apoyándose en los documentos digitales.

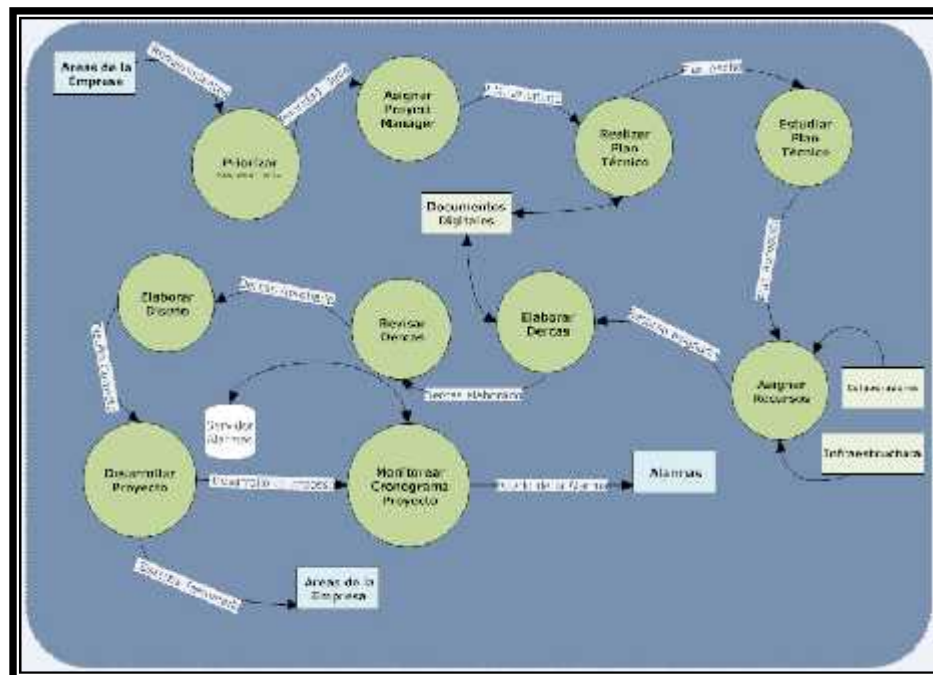


Figura 2-3.

DFD Nivel 1

- 5 Se estudia el Plan Técnico para realizar su aprobación.
- 6 Se asigna recursos de infraestructura hardware, software y colaboradores.
- 7 Se elabora el DERCAS a través de los documentos digitales.
- 8 Se revisa el DERCAS para su posterior aprobación.
- 9 Se elabora el documento de diseño con el fin de verificar que sus requerimientos estén correctos.
- 10 Comienza el desarrollo del proyecto para la puesta en marcha a producción del sistema.

- 11 Se monitorea el cronograma guía del proyecto evaluando el cumplimiento de los hitos y enviando señales de alarmas a los responsables de estos hitos.
- 12 Cuando el sistema es terminado es entregado al área que generó el requerimiento.

2.5 Diagrama de Casos de Uso

2.5.1 Descripción.

El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el Sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

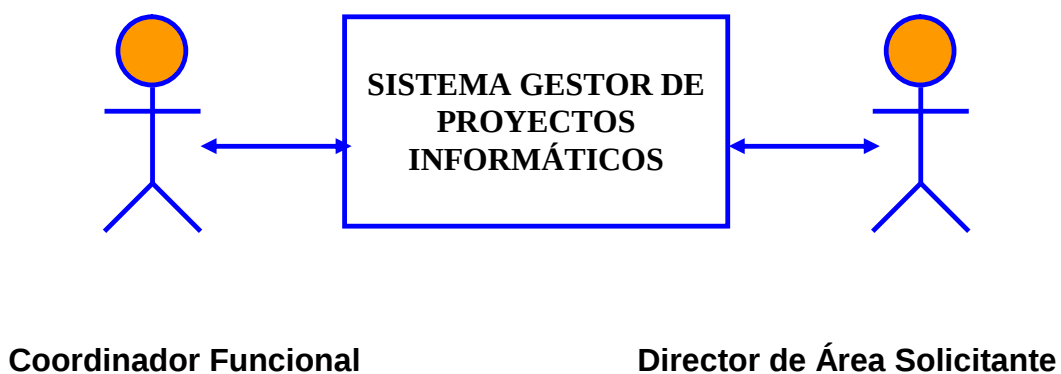


Figura 2-4.

Diagrama de Casos de Uso

Nombre:	Gestión de Requerimientos
Autor:	Grupo 6
Fecha:	14/07/2007
Descripción: Permite gestionar la creación de un requerimiento (proyecto para un área de la empresa).	
Actores: Coordinador Funcional ingresa un documento para su revisión. Coordinador Funcional establece el flujo de revisión. Líder del Proyecto revisa que el documento se encuentre de acuerdo a los requerimientos. Líder del Proyecto aprueba el documento para su publicación. Jefes / Gerentes de Áreas Involucradas aprueba el documento para su publicación. Jefe de Organización y Métodos publica el DDF para su envío a SIS.	
Precondiciones: El Coordinador Funcional debe poseer un documento de requerimiento para ingresarlo al sistema.	
Flujo Normal: 1. El Coordinador Funcional pulsa un botón para elaborar un DDF. 2. El SGPI muestra un API para el ingreso y grabación del documento en la base de datos. 3. El Coordinador Funcional pulsa un botón para establecer el flujo de revisión. 4. El SGPI muestra un API para el ingreso y grabación del flujo de revisión en la base de datos. 5. El Líder del Proyecto pulsa un botón para revisar que el documento se encuentre de acuerdo a los requerimientos. 6. El Jefe / Gerente del Área Involucrada pulsa un botón para revisar	

que el documento se encuentre de acuerdo a los requerimientos.

7. El SGPI muestra un API con el documento para su revisión y grabación del documento con los requerimientos completos en la base de datos.
8. El Líder del Proyecto pulsa un botón para aprobar el documento y que este sea publicado.
9. El Jefe / Gerente del Área Involucrada pulsa un botón para aprobar el documento y que este sea publicado.
10. El SGPI muestra un API para la aprobación y grabación del documento en la base de datos.
11. El SGPI muestra un API al Jefe de Organización y Métodos con el documento aprobado si no hay más personas en el flujo de aprobación.
12. El Jefe de Organización y Métodos pulsa un botón para publicar el DDF para su envío a SIS.
13. El SGPI muestra un API para la publicación y grabación del DDF en la base de datos.

Flujo Alternativo:

7. El SGPI muestra un API con el documento para su revisión y grabación del documento, si los requerimientos están incompletos se graba en la base de datos el documento con estado incompleto; para que el Coordinador Funcional reelabore el documento con los requerimientos necesarios.
11. El SGPI muestra un API al Jefe de Organización y Métodos con el documento aprobado si no hay más personas en el flujo de aprobación, si hay más personas en el flujo de aprobación no muestra el documento aprobado hasta que todas las personas involucradas en el flujo de aprobación aprueben el documento.

Poscondiciones:

El requerimiento ha sido creado, aprobado y guardado en el sistema.

Los demás casos de uso se encuentran detallados en la sección de Anexos.

CAPITULO 3

3. DISEÑO

El diseño de sistema es representar la arquitectura del software y definición de subsistemas y el diseño de objetos describe objetos, hasta un nivel en el cual pueden ser implementados, en un lenguaje de programación.

El Diseño de pantallas se lo obvia en este capítulo ya que el trabajo de diseño se desarrolló de forma modular.

3.1 DISEÑO DE COMPONENTES

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, etc. Un diagrama de componentes representa las dependencias entre componentes software.

Primero tenemos que identificar los Nodos que intervienen en el **SGPI** que serán los distintos módulos que tendremos; luego identificar los componentes por cada nodo estableciendo sus relaciones de dependencia y si el caso lo amerita displayar las interfaces que se relacionarán entre distintos nodos.

Tenemos el **NodoRequerimiento** en el cual constarán todos los componentes que intervengan en el módulo Requerimiento del sistema. Primero tendremos el componente SolicitudRequerimiento el cual receptará todas las solicitudes que lleguen de las áreas de la empresa y este vendría a ser otro componente llamado AgenteCliente los cuales estarán directamente relacionados.

Luego tenemos el componente PriorizaRequerimiento donde se dará un orden de prioridad a las diferentes requerimientos que lleguen por parte del AgenteCliente, esta priorización lo realizará el Gerente de Proyectos que es otro componente de este nodo y así mismo estarán directamente relacionados; después ya una vez que se priorizaron los requerimientos se procederá al ingreso del mismo encerrado en el componente IngresaRequerimiento para después llamar una Interfaz del módulo Requerimiento la cual se relacionará con el siguiente Nodo que veremos a continuación.

En el **NodoPreProyecto** lo primero que realiza es un estudio de factibilidad del requerimiento displayado en el anterior nodo, esto lo realiza en el componente FactibilidadProyecto; este estudio lo realizará un Líder de Proyecto que previamente se había asignado. Luego el Gerente de Proyectos asignará los recursos en el componente AsignaRecursos para ya ahora sí, arrancar con el proyecto visto en componente InicioProyecto, este a su vez llamará a una Interfaz del módulo de PreProyecto en el cual nos permitirá realizar los procesos del siguiente nodo.

El **NodoControlHitos** se realizará el seguimiento de las distintas tareas asignadas a los proyectos. Tenemos primero el componente RevisarStatus realizado por un Líder de Proyecto que es el encargado de lanzar los informes de los proyectos de acuerdo su estado; se generará informes estáticos y/o informes automáticos estos últimos estarán directamente relacionados con el NodoAlarmas por medio de la Interfaz Módulo e Hitos.

El **NodoAlarmas** encargado de generar alertas en el sistema al momento de definir responsables y relacionado con un Servidor de Alarmas. **(Ver Figura 3-1)**

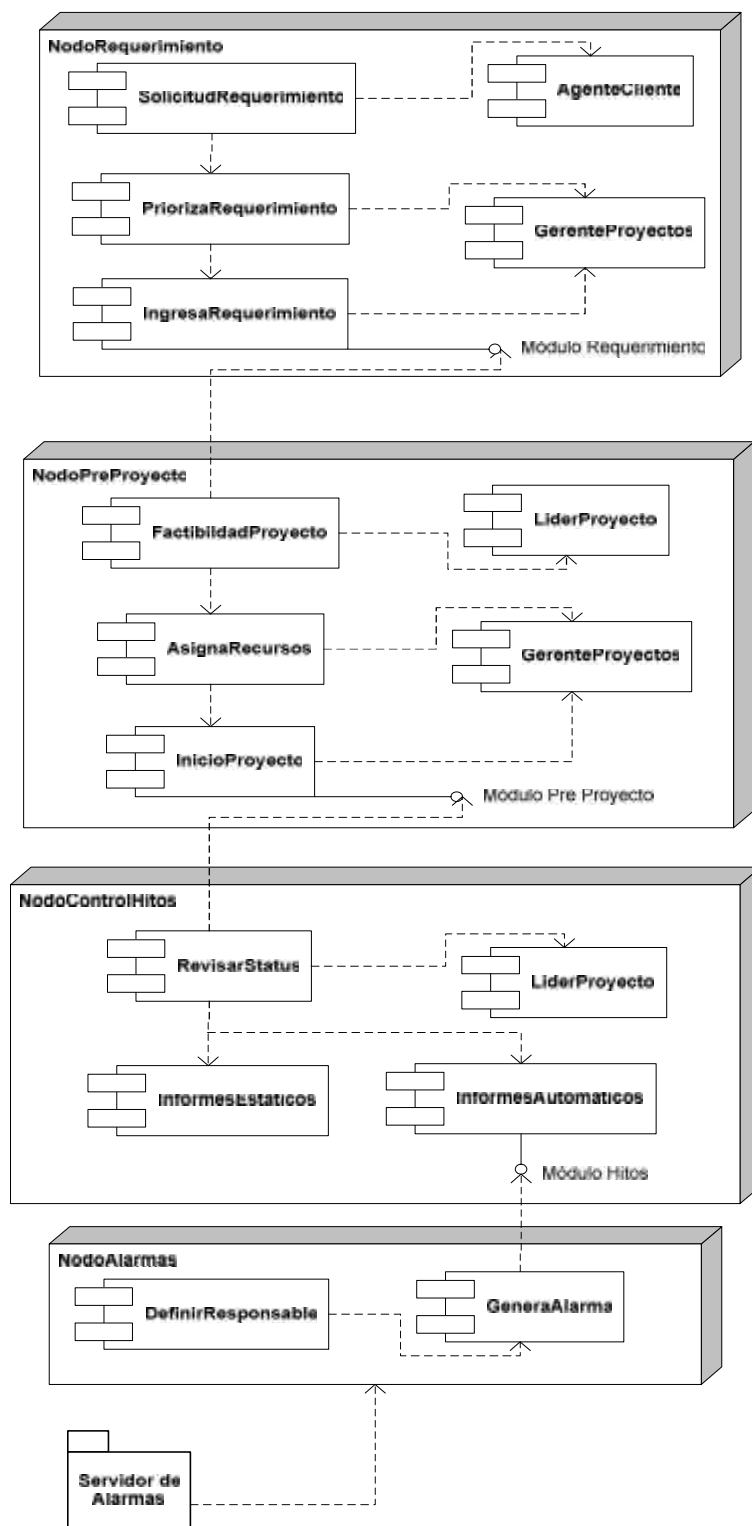


Figura 3-1

Diseño de Componentes Parte I

El **NodoLideresColaboradores** se encargará de llevar un registro del personal que tenga las características idóneas para la consecución de un proyecto. El primer componente que encontramos es RegistroColaboradores ya sean estos colaboradores internos o externos (outsourcing), agrupados en el componente Internos y Outsourcing. Luego se someterán los colaboradores registrados a evaluaciones, componente PruebaConocimientos, para después generar la creación de Líderes; displayadas posteriormente en la interfaz Módulo Líderes.

El **NodoInfraestructura** es el encargado de definir toda la infraestructura de Hardware y Software que hay disponible para los proyectos, luego se asignarán infraestructuras por proyectos en el componente InfraestructuraXProyecto, capturadas luego en la Interfaz Módulo Infraestructura.

El **NodoProveedores** encargado de someter a concursos a los aspirantes a suministrar Hardware, Software, Infraestructura y Personal; para después displayar en una Interfaz El Contrato con el Proveedor.

El **NodoPresupuesto** es el que va a calcular el costo y asignará un presupuesto a los distintos proyectos que estén en la bandeja. El componente CostoProyecto estará relacionado con la Interfaz de Líderes,

Infraestructura y Proveedores de los distintos nodos ya vistos anteriormente.

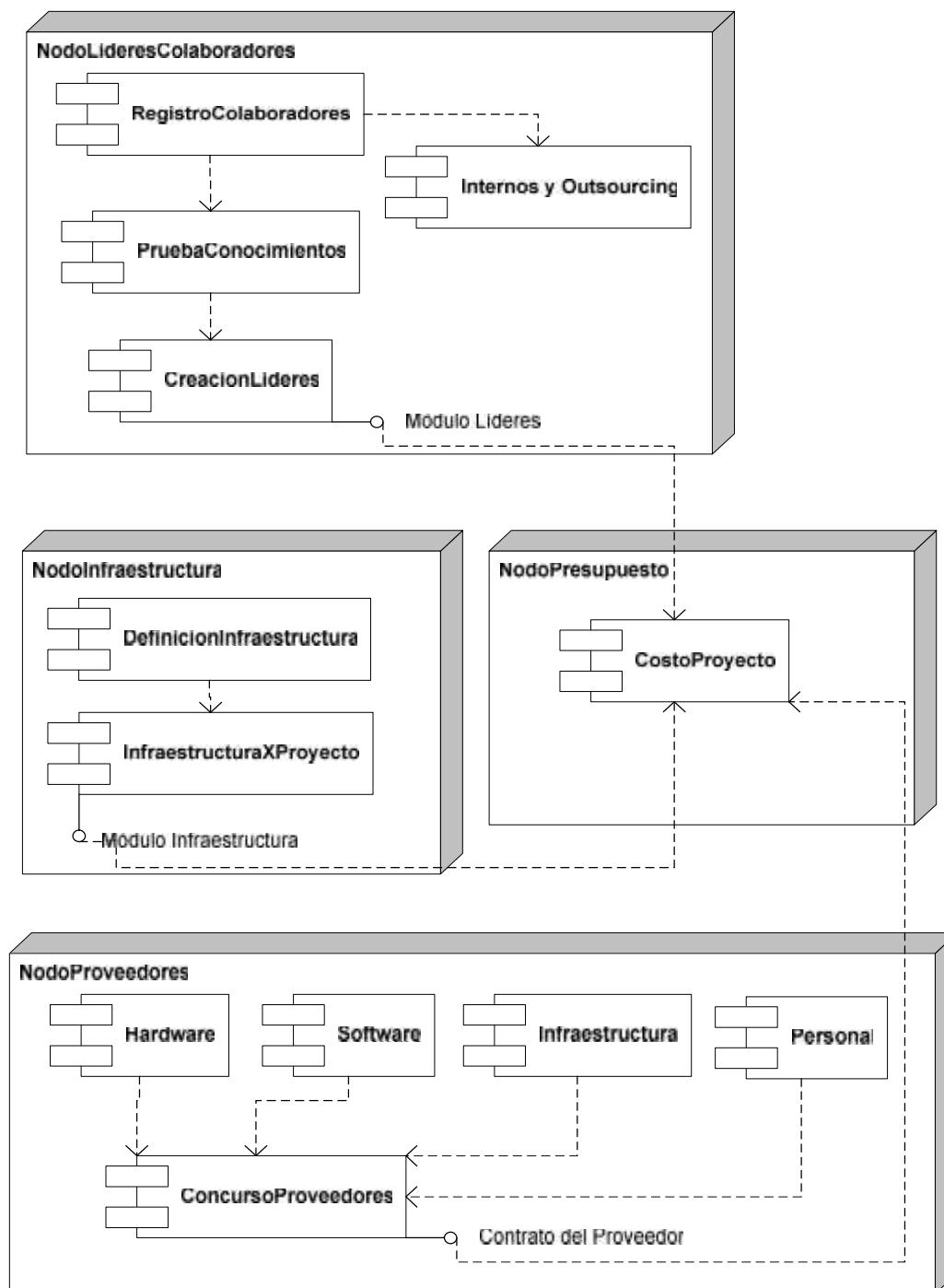


Figura 3-2

Diseño de Componentes Parte II

El **NodoDocumentación** relacionado con todos los módulos que generan algún tipo de documento y además las distintas áreas de la empresa también adjuntarán archivos o documentos que serán también digitalizados por este nodo.

El **NodoReportes** se encargará de coger toda la data de los distintos módulos y luego generarán los diferentes reportes que serán enviados a la Gerencia de Proyectos.

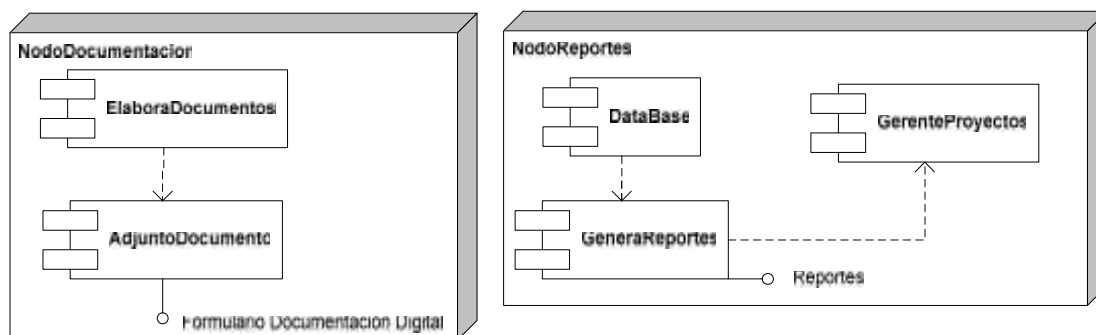


Figura 3-3

Diseño de Componentes Parte III

CAPITULO 4

4. CODIFICACIÓN

En este capítulo se detalla más específicamente la codificación y los comentarios de los paquetes desarrollados en PL/SQL y los principales procesos de la programación en lenguaje Java, cada uno de los bloques de programación están comentados de manera que se describe su comportamiento y/o objetivo.

La codificación de este módulo es referente al manejo/actualización de las tablas generales del proyecto, tales como tabla Personas, Cargos, Países, etc.

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

4.1 Estándares De Programación De Pl/Sql

Los nombres de los paquetes llevarán el siguiente formato: las tres primeras letras del módulo del proyecto seguido de la letra **K** haciendo referencia que es un paquete, luego un caracter guión bajo (_), y a continuación un nombre que describa la funcionalidad del paquete. Ej: GENK_VALIDA, este es un paquete del módulo General que realiza validaciones, todo esto en mayúscula.

Los nombres de los procedimientos y funciones llevarán el siguiente formato: comenzarán con la letra **P** ó **F** haciendo referencia si un procedimiento ó función, luego un caracter guión bajo (_), y a continuación un nombre que describa la funcionalidad del procedimiento o función. Ej: P_CARGOS. Este es un procedimiento que define los cargos del personal enrolado.

Tipo de Dato	Caracter Identificativo
NUMBER	n
VARCHAR2	v
BOOLEAN	b
ROWTYPE	r

Tabla. 4-1

Tabla de Identificador de Tipos de Datos

Los nombres de los parámetros de los procedimientos y funciones llevarán el siguiente formato: comenzarán con la letra **P** haciendo referencia que es un

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

parámetro seguido de una letra en minúscula correspondiente a su tipo de dato, en la *tabla 4-1* se detalla algunos de los principales tipos de datos utilizados. Luego el caracter guión bajo (_) y por último el nombre descriptivo del parámetro empezando con mayúscula; y si es un nombre compuesto por más de una palabra debe empezar cada palabra con mayúscula. Ej: Pn_CodigoCargo. Donde el parámetro es tipo number y representa el código de un determinado cargo.

Los nombres de los cursores llevarán el siguiente formato: comenzarán con la letra **C** haciendo referencia que es un cursor, luego un caracter guión bajo (_), y a continuación un nombre que describa la funcionalidad del cursor. Ej: C_CARGOS. Donde C representa que es un cursor y se esta realizando un proceso referente a cargos, todo esto en mayúscula.

Los nombres de los parámetros de los cursores llevarán un formato similar al de los parámetros de los procedimientos, pero con la diferencia que estos empezarán con la letra **C**. Ej: Cn_CodigoCargo. Donde C denota que es un cursor haciendo referencia a procesos de cargos, todo esto en mayúscula.

Cada módulo manejará sus excepciones y se bitacorizarán en una tabla general de errores para todos los módulos.

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

4.1.1. Descripción del Paquete GENK_VALIDA

Ahora comencemos a explicar la funcionalidad de un procedimiento del paquete de validación:

```

CREATE OR REPLACE PACKAGE BODY GENK_VALIDA IS
-- EL OBJETIVO DE ESTE PACQUETE ES ALMACENAR LOS PROCEDIMIENTOS DE VALIDACIÓN PARA EL MODELO GENERAL

PROCEDURE P_CARGO2 (Pn_CodigoCargo          IN      NUMBER,
                    Pr_RegCargo              IN OUT CEN_CARCCS%ROWTYPE,
                    Pb_Existencia            IN OUT BOOLEAN,
                    Pv_MsgError              IN OUT VARCHAR2) IS

/* OBJETIVO:
   ESTE PROCEDIMIENTO SIRVE PARA VERIFICAR LA EXISTENCIA DE LOS CARGOS.
   AUTOR CREACION:  CRUEG6
   FECHA CREACION:  01-SEPTIEMBRE-2007
   PARAMETROS:
       Pn_CodigoCargo  INGRESA:  CODIGO DEL CARGO
       Pr_RegCargo     RETORNA:  RETORNA TODO EL REGISTRO EN CASO DE QUE EXISTA
       Pb_Existencia   RETORNA:  TRUE SI EXISTE, FALSE SI NO EXISTE
       Pv_MsgError     RETORNA:  EL MENSAJE DE ERROR SI ES QUE OCURRIERE ALGUNO
   MODIFICACIONES:  (POR FAVOR INCLUIR LOS COMENTARIOS EN LAS MODIFICACIONES QUE REALICE)
*/

```

Figura. 4-1

Procedimiento P_Cargo de GENK_VALIDA

El parámetro Pn_CodigoCargo envía al procedimiento el código de cargo que va a validar si existe o no.

El parámetro Pr_RegCargo retorna todo el registro de la tabla **GEN_CARGOS** en caso de que exista.

El parámetro Pb_Existencia retorna TRUE si existe ó FALSE si no existe.

El parámetro Pv_MsgError retorna un mensaje de error si es que ocurriese alguno.

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

Declaramos un cursor para que retorne registros de la tabla GEN_CARGOS enviándole como parámetro el código del cargo.

```

CURSOR C_CARGOS(Cn_CodigoCargo NUMBER) IS
    SELECT * FROM GEN_CARGOS
    WHERE CODIGO_CARGO = Cn_CodigoCargo;
BEGIN
    Pb_Existe := FALSE;
    Pv_MsgError := NULL;
    IF Pn_CodigoCargo IS NOT NULL THEN
        IF C_CARGOS%ISOPEN THEN
            CLOSE C_CARGOS;
        END IF;
        -- APERTURA DE CURSORES
        OPEN C_CARGOS(Pn_CodigoCargo);
        FETCH C_CARGOS INTO Pr_RegCargo;
        IF C_CARGOS%FOUND THEN
            Pb_Existe := TRUE;
        ELSE
            Pb_Existe := FALSE;
        END IF;
        CLOSE C_CARGOS;
    ELSE
        Pv_MsgError := 'Parámetros Incorrectos -> GENK_VALIDA.GEN_P_CARGOS';
        RETURN;
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        Pv_MsgError:='GENK_VALIDA.GEN_P_CARGOS->' || SQLERRM;
END P_CARGOS;

```

Figura. 4-2

Cursor C_CARGOS de GENK_VALIDA

Dentro del cuerpo del procedimiento inicializamos los parámetros de salida Pb_Existe y Pv_MsgError a **FALSE** y **NULL** respectivamente.

Ahora validamos que el parámetro Pn_CodigoCargo no esté **NULL** y si lo estuviese asigno al parámetro Pv_MsgError un mensaje de error concatenándole la palabra reservada **SQLERRM** que retorna el mensaje de

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

error Oracle. Si el parámetro Pn_CodigoCargo no es **NULL**, valido si el cursor está abierto, si lo está lo cierro, con esto se asegura que el cursor está cerrado ya que no puedo abrir un cursor que esté abierto, abro el cursor enviándole como parámetro Pn_CodigoCargo; luego asigno lo que retorna el cursor a la variable tipo registro Pr_RegCargo, valido si el cursor retorna algún registro se le asigna al parámetro Pb_existe **TRUE** y si no encontró ningún registro se asigna **FALSE**, y como último paso cierro el cursor.

4.1.2. Descripción del paquete GENK_INSERTA

Detallamos el proceso del paquete de Inserción GENK_INSERTA, que realiza inserciones en una determinada tabla de Base de Datos.

Para cada tabla general habrá un procedimiento dentro del paquete de inserción.

```
CREATE OR REPLACE PACKAGE BODY GENK_INSERTA IS
--EL OBJETIVO DE ESTE PAQUETE ES HACER LAS INSERCIONES EN LAS TABLAS DEL MODULO GENERAL

PROCEDURE P_CARGOS (Pr_CodigoCargo          IN      NUMBER,
                    Pv_Monorelargo          IN      VARCHAR2,
                    Pv_MsgError             IN OUT VARCHAR2) IS

/*OBJETIVO:
    ESTE PROCEDIMIENTO SIRVE PARA REALIZAR INSERCIONES EN LA TABLA GEN_CARGOS
AUTOR CREACION: GRUPO6
FECHA CREACION: 01-SEPTIEMBRE-2007
PARAMETROS: En_CodigoCargo          INGRESA: CODIGO DEL CARGO
              Ev_Monorelargo          INGRESA: NOMBRE CARGO
              Ev_MsgError             RETORNA: EL MENSAJE DE ERROR SI ES QUE OCURRIERE ALGUN
MODIFICACIONES: (POR FAVOR INCLUYA LOS COMENTARIOS EN LAS MODIFICACIONES QUE REALICE)
*/
```

Figura. 4-3

Procedimiento P-CARGOS de GENK_INSERTA

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

El parámetro Pn_CodigoCargo envia al procedimiento el código de cargo que va a se insertado en la tabla **GEN_CARGOS**.

El parámetro Pv_NombreCargo envia al procedimiento el nombre de cargo que va a se insertado en la tabla **GEN_CARGOS**.

El parámetro Pv_MsgError sirve para retornar un mensaje de error si es que ocurriese alguno.

```

CANNOT_INSERT_NULL EXCEPTION;
PRAGMA EXCEPTION_INIT (CANNOT_INSERT_NULL, -1400);
BEGIN
    IF Pn_CodigoCargo IS NULL THEN
        Gv_Comentario := 'EL CODIGO DEL CARGO NO PUEDE SER NULO';
        Gv_Campo1:= 'Pn_CodigoCargo';
    ELSIF Pv_NombreCargo IS NULL THEN
        Gv_Comentario := 'EL NOMBRE DEL CARGO NO PUEDE SER NULO';
        Gv_Campo1:= 'Pv_NombreCargo';
    END IF;
    Gv_Sentencia := 'INSERT INTO GEN_CARGOS VALUES('
                    ||':Cn_CodigoCargo,:Cv_NombreCargo,'
                    ||'SYSDATE,USER, ''
                    ||GENK_PARAMETROS.F_HOST
                    ||'',NULL,NULL,NULL)';
    EXECUTE IMMEDIATE Gv_Sentencia USING Pn_CodigoCargo,
                                         Pv_NombreCargo;

    COMMIT;
    Pv_MsgError := NULL;

```

Figura. 4-4

(Continúa) Procedimiento P_CARGOS de GENK_INSERTA

Declaramos una variable tipo **EXCEPTION** para invocarla dentro del controlador de excepciones cuando un campo de base no permita valores nulos. Con la palabra reservada **PRAGMA EXCEPTION_INIT** definimos en que momento se debería ir por la excepción la variable declarada y mandamos

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

como parámetros el nombre de la variable y el número de error que Oracle reconoce cuando no acepta valores nulos (**-1400**).

Dentro del cuerpo del procedimiento validamos si el parámetro Pn_CodigoCargo es **NULL** y si lo es, asigno a la variable global **Gv_Comentario** declarada en la cabecera del paquete un mensaje de error indicando el motivo y a la variable global **Gv_Campo1** le asigno el nombre del parámetro. También validamos el parámetro Pv_NombreCargo haciéndole similar asignación.

La sentencia SQL la colocamos en una cadena, variable global Gv_Sentencia, la instrucción SQL se creará dinámicamente. El SQL nativo gestiona el cursor automáticamente utilizando la instrucción **EXECUTE IMMEDIATE**. También usamos la palabra reservada **USING** cuando enviamos parámetro de ejecución en la cadena, identificadas porque llevan 2 puntos antes del nombre.

En la tabla GEN_CARGOS existe un campo estacion_ingreso, al momento de insertar el valor al dicho campo se llamará a una función de otro paquete en el cual devolverá el nombre de la máquina en la que se realiza la inserción; este paquete se llama **GENK_PARAMETROS** y la función **F_HOST**. (**Ver Figura 4-5.**)

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

Escribimos la palabra reservada **COMMIT** que es para confirmar la transacción que se acaba de realizar e inicializamos el parámetro Pv_MsgError a NULL.

```
CREATE OR REPLACE PACKAGE BODY GENK_PARAMETROS IS

FUNCTION F_HOST RETURN VARCHAR2 IS
Lv_HostName VARCHAR2(30);
BEGIN
    Lv_HostName := UTL_INADDR.GET_HOST_NAME();
    DBMS_OUTPUT.PUT_LINE(Lv_HostName);
    RETURN Lv_HostName;
END F_HOST;
```

Figura. 4-5

Función F_HOST de GENK_PARAMETROS

```
GENK_BITACORIZACION.P_BITACORIZA_INTRO(Pv_Id_Proceso => 'GENK_INSERTA_P_CARGOS',
    Pv_fecha => NULL,
    Pv_usuario => NULL,
    Pv_Comentario => 'INSERTION EN GEN_CARGOS FINALIZÓ CORRECTAMENTE',
    Pv_Campo1 => NULL,
    Pv_Campo2 => NULL,
    Pv_Campo3 => NULL,
    Pv_Msg => Pv_MsgError);
DBMS_OUTPUT.PUT_LINE('Insertión finalizó correctamente');
EXCEPTION
    WHEN CANNOT_INSERT_NULL THEN
        GENK_BITACORIZACION.F_BITACORIZA_ERROR(Pv_Id_Proceso => 'GENK_INSERTA_P_CARGOS',
            Pv_fecha => NULL,
            Pv_usuario => NULL,
            Pv_Comentario => Pv_Comentario,
            Pv_Campo1 => Pv_Campo1,
            Pv_Campo2 => SQLCODE,
            Pv_Campo3 => SQLERRM,
            Pv_Msg => Pv_MsgError);
    WHEN DUP_VAL_IN_INDEX THEN
        GENK_BITACORIZACION.F_BITACORIZA_ERROR(Pv_Id_Proceso => 'GENK_INSERTA_P_CARGOS',
            Pv_fecha => NULL,
            Pv_usuario => NULL,
            Pv_Comentario => 'EL CÓDIGO DEL CARGO YA EXISTE',
            Pv_Campo1 => Pv_Campo1,
            Pv_Campo2 => SQLCODE,
            Pv_Campo3 => SQLERRM,
            Pv_Msg => Pv_MsgError);
    WHEN OTHERS THEN
        Pv_MsgError:='GENK_INSERTA_P_CARGOS-'|| SQLERRM;
END P_CARGOS;
```

Figura. 4-6

Bloque de excepciones del Procedimiento P_CARGOS de GENK_INSERTA

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

Llamamos al paquete de bitacORIZACIÓN para guardar un mensaje indicando que la inserción finalizó correctamente y si no envía a bitacorar los errores que se haya producidos en el bloque **EXCEPTION**. (*Ver figura 4.6*).

4.1.3. Descripción del paquete GENK_BITACORIZACION

GENK_BITACORIZACION maneja la información acerca de los errores que se han producido, existen niveles de error: como el error, info, debug y warning. Para bitacorizar es necesario tener creadas las tablas GEN_PARAMETROS, GEN_TIPOS_PARAMETROS y GEN_BITACORA donde se guardan toda la parametrización del sistema.

Comenzemos por detallar el procedimiento **P_BITACORIZA_INFO**, donde se muestra como se almacena información de error tipo INFO(información), de la misma forma existen procedimientos para error tipo error, debug y warning.

En las primera líneas de la **Figura 4-7** se declaran los parámetros de entrada y de salida que se usarán. El parámetro Pv_Id_Proceso sirve para enviar al procedimiento el nombre identificador del proceso, paquete, función, etc. que hace el llamado a GENK_BITACORIZA, este dato será insertado en la tabla GEN_BITACORA, en el campo que almacena la fuente donde se suscitó el error.

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

```

CREATE OR REPLACE PACKAGE BODY GENK_BITACORIZACION IS

PROCEDURE P_BITACORIZA_INFO(Pv_Id_Proceso IN VARCHAR2,
                             Pv_Fecha      IN DATE DEFAULT SYSDATE,
                             Pv_Usuario    IN VARCHAR2,
                             Pv_Comentario IN VARCHAR2,
                             Pv_Campo1     IN VARCHAR2 DEFAULT NULL,
                             Pv_Campo2     IN VARCHAR2 DEFAULT NULL,
                             Pv_Campo3     IN VARCHAR2 DEFAULT NULL,
                             Pv_Msg        OUT VARCHAR2) IS

PRAGMA AUTONOMOUS_TRANSACTION;
Lv_Msg VARCHAR2(500);
Le_MiError EXCEPTION;

CURSOR C_NIVEL_BITA IS
    SELECT NVL(TO_NUMBER(VALOR),2)
    FROM GEN_PARAMETROS G
    WHERE G.ID_TIPO_PARAMETRO = 1
    AND G.ID_PARAMETRO = 'NIVEL_BITA';

Ln_Nivel GEN_BITACORA.NIVEL%TYPE;
Lb_Found BOOLEAN;
Le_NoBitacoriza EXCEPTION;

BEGIN
    OPEN C_NIVEL_BITA;
    FETCH C_NIVEL_BITA INTO Ln_Nivel;
    Lb_Found := C_NIVEL_BITA%FOUND;
    CLOSE C_NIVEL_BITA;
    IF NOT Lb_Found THEN
        Ln_Nivel := 4;
    END IF;
    IF Ln_Nivel < 1 THEN
        RAISE Le_NoBitacoriza;
    END IF;

```

Figura. 4-7

Procedimiento P_Bitacoriza_Info de GENK_BITACORIZACION

El parámetro Pv_Fecha sirve para enviar al procedimiento la fecha y hora que se realizó la bitacorización y también se inserta en la tabla GEN_BITACORA.

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

El parámetro Pv_Usuario sirve para enviar al procedimiento el usuario que realizó el proceso insertándose también en la tabla **GEN_BITACORA**. El parámetro Pv_Comentario sirve para enviar al procedimiento una descripción del motivo de la bitacorización insertándose también en la tabla **GEN_BITACORA**.

Los parámetros Pv_Campo1, Pv_Campo2 y Pv_Campo3 sirven para enviar al procedimiento mas detalles de la bitacorización insertándose también en la tabla **GEN_BITACORA**. El parámetro Pv_Msg sirve para retornar un mensaje de error si es que ocurriese alguno.

Implementamos **PRAGMA_AUTONOMUS_TRANSACTION**. Para las funciones empaquetadas, se debe utilizar el pragma (directorio del compilador). El pragma hace que el compilador de PL/SQL niegue a la función empaquetada el acceso de lectura/grabación a las tablas de la base de datos, a las variables empaquetadas, o a ambas.

Si se intenta compilar una función que viole el pragma, usted consigue un error de la compilación. El pragma **PRAGMA_AUTONOMUS_TRANSACTION** permite la implementación de SUB-Transacciones, que pueden ejecutar commit o rollback de manera independiente a la transacción externa.

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

Declaramos 2 variables locales Lv_Msg para capturar los posibles errores y Le_MiError de tipo **EXCEPTION** para saltar al controlador de excepciones cuando se lo invoque con la palabra reservada **RAISE** en el cuerpo del procedimiento.

Declaramos 3 variables locales más; Ln_Nivel para almacenar el nivel de bitacORIZACIÓN que está parametrizado en la base de datos, Lb_Found para guardar TRUE si encuentra el nivel y FALSE si no lo encuentra, Le_NoBitacoriza de tipo **EXCEPTION** para saltar al controlador de excepciones cuando el nivel encontrado cumpla una validación.

Declaramos un cursor para que me retorne el campo valor de la tabla GEN_PARAMETROS filtrándole que el tipo de parámetro sea igual a **1**(para nuestro ejemplo) y el id del parámetro sea igual a **NIVEL_BITA**. Abrimos el cursor, asignamos el valor que me trae el cursor a la variable Ln_Nivel, asigno a Lb_Found TRUE o FALSE si el cursor trae o no registro, y cierro el cursor.

Si no encontré registro le asigno a Ln_Nivel, esto quiere decir que no se especificó ningún nivel específico, entonces asignamos el valor de 4 para que bitacorize en los 4 niveles existentes(error, info, debug y warning).

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

Valido si Ln_Nivel es menor al nivel de bitacORIZACIÓN que estoy en ese momento en este caso **INFO** nivel **1** si se cumple se salta a la excepción Le_NoBitacoriza. Luego valido que el parámetro Pv_Id_Proceso no sea **NULL** y si es nulo se salta a la excepción Le_MiError.

Realizamos el **Insert** a la tabla GEN_BITACORA enviándole como valores los parámetros del procedimiento, pero para el campo **NIVEL** invocamos a la variable global **Gn_Info** definida en el paquete **GENK_GLOBALES** y se realiza el COMMIT.

En la cabecera del paquete declaramos las variables globales que vamos a utilizar.

```
CREATE OR REPLACE PACKAGE GENK_GLOBALES AS
  Gr_Info          NUMBER;
  Gr_Error         NUMBER;
  Gr_Debug         NUMBER;
  Gr_Warning       NUMBER;
END GENK_GLOBALES;
```

Figura. 4-8

Encabezado de GENK_GLOBALES

En el cuerpo del paquete realizamos los respectivos querys para que me retorne el valor del parámetro ya sea INFO o ERROR y lo almaceno en las respectivas variables que fueron declaradas en la cabecera. **(Ver Figura 4-9)**

Además en el bloque de EXCEPTION, se setea el SQLCODE(número de error Oracle) en la variable error l paquete GENK_MANEJA_ERROR

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

```

CREATE OR REPLACE PACKAGE BODY GENK_GLOBALES AS
BEGIN
  BEGIN
    SELECT TO_NUMBER(VAOR)
    INTO Gn_Info
    FROM GEN_PARAMETROS
    WHERE ID_PARAMETRO = 'INFO';
  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      GENK_MANEJA_ERROR.ERROR(Pn_CodigoError => SQLCODE);

    WHEN TOO_MANY_ROWS THEN
      GENK_MANEJA_ERROR.ERROR(Pn_CodigoError => SQLCODE);

    WHEN OTHERS THEN
      GENK_MANEJA_ERROR.ERROR(Pn_CodigoError => SQLCODE);

  END;

  BEGIN
    SELECT TO_NUMBER(VAOR)
    INTO Gn_Error
    FROM GEN_PARAMETROS
    WHERE ID_PARAMETRO = 'ERROR';
  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      GENK_MANEJA_ERROR.ERROR(Pn_CodigoError => SQLCODE);

    WHEN TOO_MANY_ROWS THEN
      GENK_MANEJA_ERROR.ERROR(Pn_CodigoError => SQLCODE);

    WHEN OTHERS THEN
      GENK_MANEJA_ERROR.ERROR(Pn_CodigoError => SQLCODE);

  END;

```

Figura. 4-9

Cuerpo de GENK_GLOBALES

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

4.2 ESTÁNDARES DE PROGRAMACIÓN DE JAVA

4.2.1 Controlador Web

Usamos el *framework* Hibernate que es un motor de persistencia, que independiza la utilización de diferentes base de datos, y permite una programación organizada, concentrando su dificultad en el *mapping*.

El controlador web es el que se encarga de recibir y enviar los requerimientos al servidor a través de los *servlets*.

```
package sgpi.web;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.swing.JOptionPane;

import sgpi.pojo.general.*;
import sgpi.web.util.*;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.*;
import java.sql.Date;

import org.hibernate.criterion.DetachedCriteria;
import org.hibernate.criterion.Order;
import org.hibernate.criterion.Property;
import org.springframework.beans.factory InitializingBean;
import org.springframework.web.bind.ServletRequestBindingException;
import org.springframework.web.bind.ServletRequestUtils;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.multiaction.MultiActionController;

import sgpi.AccesoBaseDatos;
import sgpi.AppException;
```

Figura. 4-10

Import del Controlador Web

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

Los *import* del paquete **java** son para poder llamar a las funciones que se crearon en las clases `ParseException`, `SimpleDateFormat`, todas las clases del util y la clase `Date`. Los *import* del paquete **org.hibernate** son para poder llamar a las funciones que se crearon en las clases `DetachedCriteria`, `Order` y `Property`.

Los *import* del paquete **org.springframework** son para poder llamar a las funciones que se crearon en las clases `InitializingBean`, `ServletRequestBindingException`, `ServletRequestUtils`, `ModelAndView` y `MultiActionController`.

Creamos la clase `ControladorWeb` (Ver **Figura 4-11**) con un **extends** haciendo referencia a una Superclase de Spring **MultiActionController** y un **implements** indicando que se usará la Interface **InitializingBean**.

Declaro un atributo de alcance **private** con tipo de dato **int** para el tamaño de la paginación de un listado que se presentará al momento de realizar una búsqueda.

También declaro un atributo de tipo de dato **AccesoBaseDatos** para poder acceder a los distintos métodos de la clase `AccesoBaseDatos`.

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

Creo el método `formatoDate` que me devuelve un tipo de dato **Date** le mando como parámetros un servlet y una cadena de string; este método nos servirá para poder setear una fecha al momento de mandarlo a grabar en *Persona*.

```
public class ControladorWeb extends MultiActionController implements InitialisingBean {

    private int tamanioPagina;
    AccesoBaseDatos accesoDB;

    /**
     * Da formato a un texto dd MM/yyyy
     *
     */
    private Date formatoDate(HttpServletRequest request, String parameter) {
        Date fecha = null;
        SimpleDateFormat d = new SimpleDateFormat("dd/MM/yyyy");
        try {
            String texto = ServletRequestUtils.getStringParameter(request, parameter);
            fecha = new Date(d.parse(texto).getTime());
        } catch (ParseException e) {
            e.printStackTrace();
        } catch (ServletRequestBindingException e) {
            e.printStackTrace();
            ActionMessage msg = new ActionMessage("mensaje.error", e.getMessage());
        }
        return fecha;
    }

    private Cargo formatoCargo(HttpServletRequest request, String parameter) {
        Cargo cargo = new Cargo();
        try {
            String codigo = ServletRequestUtils.getStringParameter(request, parameter);
            cargo = new Cargo(Integer.valueOf(codigo));
        } catch (ServletRequestBindingException e) {
            System.out.println(e.getMessage());
        }
        return cargo;
    }
}
```

Figura. 4-11

Clase Controlador Web_Parte1

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

Creo el método `formatoCargo` que me devuelve un tipo de dato **Cargo** le mando como parámetros un servlet y una cadena de string; este método nos servirá para poder setear el código del cargo en la clase `Persona`.

```
private Departamento formatoDepartamento(HttpServletRequest request, String parametro) {
    Departamento departamento = new Departamento();
    try {
        String codigo = ServletRequestUtils.getStringParameter(request, parametro);
        departamento = new Departamento(Integer.valueOf(codigo));
    } catch (ServletRequestBindingException e) {
        System.out.println(e.getMessage());
    }
    return departamento;
}

private Estado formatoEstado(HttpServletRequest request, String parametro) {
    Estado estado = new Estado();
    try {
        String codigo = ServletRequestUtils.getStringParameter(request, parametro);
        estado = new Estado(Integer.valueOf(codigo));
    } catch (ServletRequestBindingException e) {
        System.out.println(e.getMessage());
    }
    return estado;
}

private EstadoCivil formatoEstadoCivil(HttpServletRequest request, String parametro) {
    EstadoCivil estadoCivil = new EstadoCivil();
    try {
        String codigo = ServletRequestUtils.getStringParameter(request, parametro);
        estadoCivil = new EstadoCivil(Integer.valueOf(codigo));
    } catch (ServletRequestBindingException e) {
        System.out.println(e.getMessage());
    }
    return estadoCivil;
}
```

Figura. 4-12

Clase Controlador Web_Parte2

(Ver **Figura 4-12**) . Creo el método `formatoDepartamento` que me devuelve un tipo de dato **Departamento** le mando como parámetros un servlet y una cadena de string; este método nos servirá para poder setear el código del departamento en la clase `Persona`.

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

Creo el método `formatoEstado` que me devuelve un tipo de dato **Estado** le mando como parámetros un servlet y una cadena de string; este método nos servirá para poder setear el código del estado en la clase `Persona`.

Creo el método `formatoEstadoCivil` que me devuelve un tipo de dato **EstadoCivil** le mando como parámetros un servlet y una cadena de string; este método nos servirá para poder setear el código del estado civil en la clase `Persona`.

```
private Profesion formatoProfesion(HttpServletRequest request, String parametro) {
    Profesion profesion = new Profesion();
    try {
        String codigo = ServletRequestUtils.getStringParameter(request, parametro);
        profesion = new Profesion(Integer.valueOf(codigo));
    } catch (ServletRequestBindingException e) {
    }
    return profesion;
}

private TipoIdentificacion formatoTipoIdentificacion(HttpServletRequest request, String parametro) {
    TipoIdentificacion tipoIdentificacion = new TipoIdentificacion();
    try {
        String codigo = ServletRequestUtils.getStringParameter(request, parametro);
        tipoIdentificacion = new TipoIdentificacion(Integer.valueOf(codigo));
    } catch (ServletRequestBindingException e) {
    }
    return tipoIdentificacion;
}

private TipoRelacion formatoTipoRelacion(HttpServletRequest request, String parametro) {
    TipoRelacion tipoRelacion = new TipoRelacion();
    try {
        String codigo = ServletRequestUtils.getStringParameter(request, parametro);
        tipoRelacion = new TipoRelacion(Integer.valueOf(codigo));
    } catch (ServletRequestBindingException e) {
    }
    return tipoRelacion;
}
```

Figura. 4-13

Clase Controlador Web_Parte3

Módulo de Integración, Parametrización y Configuraciones.

(Ver Figura 4-13). Creo el método `formatoProfesion` que me devuelve un tipo de dato **Profesion** le mando como parámetros un servlet y una cadena de string; este método nos servirá para poder setear el código de la profesión en la clase `Persona`.

Creo el método `formatoTipoIdentificacion` que me devuelve un tipo de dato **TipoIdentificacion** le mando como parámetros un servlet y una cadena de string; este método nos servirá para poder setear el código del tipo de identificación en la clase `Persona`.

Creo el método `formatoTipoRelacion` que me devuelve un tipo de dato **TipoRelacion** le mando como parámetros un servlet y una cadena de string; este método nos servirá para poder setear el código del tipo de relación en la clase `Persona`.

```
private Area formatoArea(HttpServletRequest request, String parameter) {
    Area area = new Area();
    try {
        String codigo = ServletRequestUtils.getStringParameter(request, parameter);
        area = new Area(Integer.valueOf(codigo));
    } catch (ServletRequestBindingException e) {
        System.out.println(e.getMessage());
    }
    return area;
}
```

Figura. 4-14

Clase Controlador Web_Parte4

Módulo de Integración, Parametrización y Configuraciones.

Creo el método `formatoArea` que me devuelve un tipo de dato **Area** le mando como parámetros un servlet y una cadena de string; este método nos servirá para poder setear el código del área en la clase `Departamento`.

```

/**
 * Editor de personas
 */
public ModelAndView personaEditor(HttpServletRequest request, HttpServletResponse response)
throws ServletException {

    List cargos = getAccesoDB().consultaCargos();
    List tiporelacion = getAccesoDB().consultaTipoRelacion();
    List tipoidentificacion = getAccesoDB().consultaTipoIdentificacion();
    List estadoscivil = getAccesoDB().consultaEstadosCiviles();
    List parroquias = getAccesoDB().consultaParroquias();
    List ciudades = getAccesoDB().consultaCiudades();
    List provincias = getAccesoDB().consultaProvincias();
    List paises = getAccesoDB().consultaPaises();
    List profesiones = getAccesoDB().consultaProfesiones();
    List departamentos = getAccesoDB().consultaDepartamentos();
    List estados = getAccesoDB().consultaEstados();

    System.out.println("Cargos" + cargos.size());
    System.out.println("Relaciones" + tiporelacion.size());
    System.out.println("Identificaciones" + tipoidentificacion.size());
    System.out.println("Estados Civiles" + estadoscivil.size());
    System.out.println("Parroquias" + parroquias.size());
    System.out.println("Ciudades" + ciudades.size());
    System.out.println("Provincias" + provincias.size());
    System.out.println("Paises" + paises.size());
    System.out.println("Profesiones" + profesiones.size());
    System.out.println("Departamentos" + departamentos.size());
    System.out.println("Estados" + estados.size());
    Persona persona = new Persona();
}

```

Figura. 4-15

Clase Controlador Web_Parte5

Creo el método `personaEditor` que me devuelve un tipo de dato **ModelAndView** le mando como parámetros un servlet para request y un servlet para response; este método nos servirá para poder realizarle

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

Mantenimiento (Ingreso, Consulta, Modificación y Eliminación) a la clase Persona.

Declaro variables de tipo **List** para traer la descripción de los campos que son hijos de otras tablas, para que se llenen los combos en la página JSP.

Las variables son cargos, tipoRelacion, tipoIdentificacion, estadosCivil, parroquias, ciudades, provincias, paises, profesiones, departamentos y estados. Estas variables invocan al método **getAccesoDB()** que es el método encapsulado del atributo accesoDB de la interface AccesoBaseDatos declarado al inicio de la clase ControladorWeb, y este método a su vez llama a los diferentes métodos de consulta para cada variable declarada.

Los métodos de consulta son consultaCargos(), consultaTipoRelacion(), consultaTipoIdentificacion(), consultaEstadosCivil(), consultaParroquias(), consultaCiudades(), consultaProvincias(), consultaPaises(), consultaProfesiones(), consultaDepartamentos() y consultaEstados(). Todos estos métodos realizan la consulta a la base de datos usando **Hibernate**.

Presento por la Consola de Eclipse **Mensajes** con el nombre de la lista y la cantidad de registros que me devuelven cada uno de las listas.

Instancio a la clase Persona creando un objeto persona.

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

```

String accion= ServletRequestUtils.getStringParameter(request, "accion");
Integer codigo = Integer.valueOf(ServletRequestUtils.getIntParameter(request, "codigo", 0));
if ("guardar".equals(accion)) {
    if (codigo != Integer.valueOf(0)) {
        persona = getDatosDB().getPersona(codigo);
    }
    persona.setApellidos(ServletRequestUtils.getStringParameter(request, "apellidos");
    persona.setDireccion(ServletRequestUtils.getStringParameter(request, "direccion");
    persona.setEmail(ServletRequestUtils.getStringParameter(request, "email");
    persona.setTelefono(ServletRequestUtils.getStringParameter(request, "telefono");
    persona.setNombreCompleto(ServletRequestUtils.getStringParameter(request, "nombreCompleto");
    persona.setNombre(ServletRequestUtils.getStringParameter(request, "nombre");
    persona.setNumeroIdentificacion(ServletRequestUtils.getStringParameter(request, "numeroIdentificacion");
    persona.setSexo(ServletRequestUtils.getStringParameter(request, "sexo");
    persona.setLugarTrabajo(ServletRequestUtils.getStringParameter(request, "lugarTrabajo");
    persona.setCargo(formatoCargo(request, "cargo"));
    persona.setDepartamento(formatoDepartamento(request, "departamento");
    persona.setEstado(formatoEstado(request, "estado");
    persona.setEstadoCivil(formatoEstadoCivil(request, "estadoCivil");
    persona.setFechaNacimiento(formatoFecha(request, "fechaNacimiento");
    persona.setParroquia(formatoParroquia(request, "parroquia");
    persona.setProfesion(formatoProfesion(request, "profesion");
    persona.setTipoIdentificacion(formatoTipoIdentificacion(request, "tipoIdentificacion");
    persona.setTipoRelacion(formatoTipoRelacion(request, "tipoRelacion");
    getDatosDB().guardaPersona(persona);
}

```

Figura. 4-16

Clase Controlador Web_Parte6

Declaro una variable tipo String para guardar la acción que me trae el ServletRequest desde una página JSP por medio del action del formulario de la página que lo está invocando. Declaro una variable tipo Integer que va a guardar el código que me trae el ServletRequest desde la página JSP que lo invocó. Valido la acción que recibió el String que me envió el servlet, para las opciones del menú **menubar** creadas en el JSP.

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

Si la acción es **guardar**, primero valido que el registro que voy a insertar me trae o no un código, porque si el código es nulo tengo que mapear la secuencia del objeto para que genere el siguiente código, caso contrario invoco al método `getAccesoDB()` y este al método `getPersona` para que me traiga la persona actual enviándole como parámetro el código.

Invoco a los distintos métodos de seteos de la clase `Persona` por medio del objeto `persona` y recibe como parámetros los datos que me trae el `Servlet` desde la página `JSP` que lo invocó. Para terminar el proceso de guardar invoco al método **`getAccesoDB()`** que es el método encapsulado del atributo `accesoDB` declarado al principio de la clase `ControladorWeb` y este método llama al método `guardaPersona` realizando la inserción o actualización a la base de datos usando **Hibernate**.

(Ver figura 4-17). La acción **consultar** no se la invoca desde el menú **menubar** sino que se la llama dentro de la página `JSP buscar` específica para realizar búsquedas dinámicas.

Si la acción es **siguiente**, invoco al método **`getAccesoDB()`** que es el método encapsulado del atributo `accesoDB` declarado al principio de la clase `ControladorWeb` y este método llama a otro método `getPersonaSgteAnterior`

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

enviándole como parámetros el código y el valor **1** para que vaya al siguiente registro. Y si la acción es **anterior** realizo lo mismo pero ahora quemando el valor en el parámetro a **-1**.

Si la acción es **borrar**, invoco al método **getAccesoDB()** que es el método encapsulado del atributo `accesoDB` declarado al principio de la clase `ControladorWeb` y este método llama al método `getPersona` para que me traiga la persona actual enviándole como parámetro el código.

Se llama de nuevo al método `getAccesoDB()` y este invoca al método `removeObjeto` realizando la inserción o actualización a la base de datos usando **Hibernate**.

Si la acción es **nuevo** instancio una nueva persona del objeto creado. Instancio a la clase `HashMap` creando un objeto con el cual mapeo el objeto persona y las listas que son usadas para llenar los combos y las paso a sus respectivas variables de los JSP desde las cuales son seteadas.

Y por último retorno el mapping con la clase `personaForm` que es creado en el archivo de configuración de Hibernate `views.properties`

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

```

else if ("consultar".equals(accion)) {
    persona = getAccesoDB().getPersona(codigo);
}
else if ("siguiente".equals(accion)) {
    persona = getAccesoDB().getPersonaSigteAnterior(codigo, 1);
}
else if ("anterior".equals(accion)) {
    persona = getAccesoDB().getPersonaSigteAnterior(codigo, -1);
}
else if ("borrar".equals(accion)) {
    persona = getAccesoDB().getPersona(codigo);
    getAccesoDB().removeObjeto(persona);
}
else { //nuevo
    persona = new Persona();
}
HashMap map = new HashMap();
map.put("persona", persona);
map.put("cargos", cargos);
map.put("tipoRelacion", tiporelacion);
map.put("tipoIdentificacion", tipoidentificacion);
map.put("estadosCivil", estadoscivil);
map.put("parroquias", parroquias);
map.put("ciudades", ciudades);
map.put("provincias", provincias);
map.put("países", países);
map.put("profesiones", profesiones);
map.put("departamentos", departamentos);
map.put("estados", estados);
return new ModelAndView("personaForm", map);
}

```

Figura. 4-17

Clase Controlador Web_Parte7

(Ver figura. 4-18).Creo el método buscarPersona que me devuelve un tipo de dato ModelAndView le mando como parámetros un servlet para request y un servlet para response; este método nos servirá para poder realizarle una búsqueda dinámica a la clase Persona dependiendo de un criterio de búsqueda.

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

Declaro una variable de tipo int que guarda el número de página que me trae el servlet desde la página JSP que lo llamó. Declaro un string que guarde lo que trae el servlet que es el criterio de la búsqueda llamado desde una página JSP.

```
public ModelAndView buscarPersona(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    int numeroPagina = ServletRequestUtils.getIntParameter(request, "pageNum", 1);
    String porNombre = ServletRequestUtils.getStringParameter(request, "nombreCompleto", "");

    DetachedCriteria query = DetachedCriteria.forClass(Persona.class);
    query.add(Property.forName("nombreCompleto").like("%" + porNombre + "%"));
    query.addOrder(Order.asc("nombreCompleto"));
    Pagina pagina = getAccesoDB().getQueryPorPagina(query, numeroPagina, tamañoPagina);

    Map refData = new HashMap();
    refData.put("pagina", pagina);

    return new ModelAndView("buscarPersona", refData);
}
```

Figura. 4-18

Clase Controlador Web_Parte8

Declaro una variable de tipo **DetachedCriteria** que es una clase de Hibernate que va a guardar a que clase le voy a realizar la búsqueda, enviándole como parámetro la clase Persona.

Le añado el criterio de búsqueda a la variable antes declarada y le asigno con la función **like** el string que declaré y me guardó el criterio que me trajo el servlet. Luego ordeno mi búsqueda con la clase de Hibernate **addOrder** y como parámetro el criterio del ordenamiento.

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

Instancio a la clase Pagina realizada en el paquete sgpi.web.util; que nos sirve para la paginación entre los registros que me traerá la búsqueda, invoco al método `getAccesoDB` que es el método encapsulado del atributo `accesoDB` declarado al principio de la clase `ControladorWeb` y este método llama al método **`getQueryPorPagina`** enviándole como parámetros la variable tipo `DetachedCriteria`, la variable `int` que me guardó el número de la página y el atributo declarada al inicio de la clase `ControladorWeb` para el tamaño de la paginación.

Instancio a la clase `Map` creando un objeto con el cual mapeo a la pagina y las paso la variable que el JSP usa para invocarla. Y por último retorno el `map` con la clase `buscarPersona` que es creado en el archivo de configuración de Hibernate `views.properties`.

Repito el procedimiento de `personaEditor` y `buscarPersona` para cualquier proceso de mantenimiento de una clase creada y mapeada a la base. Se lo realiza con las clases `Cargo`, `Area`, `Departamento`, `Provincia` y `Estado`.

Esta es la funcionalidad de la clase `Matriz` o controladora de nuestra aplicación Web llamada `ControladorWeb` creada en el paquete `sgpi.web`.

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

CAPITULO 5

5. PRUEBAS

Se ha realizado tres tipos de prueba: pruebas de desarrollo, pruebas de calidad internas y pruebas con usuarios.

Se realizó pruebas a nuestro módulo de forma local e integrado en el servidor web central de nuestro proyecto. Así mismo se realizaron pruebas generales de todo el proyecto como un solo ente de software, ya no por módulos.

Las pruebas a manera local de nuestro módulo se las ha hecho desde el inicio siguiendo el consejo de que no se debe esperar que este terminado para

probar, y el del entorno general se lo realizó cada vez que un nuevo módulo era integrado al sistema, que se desarrollo en el último mes.

En el siguiente gráfico podemos ver la distribución de las etapas del software durante el tiempo calendario según un estudio exploratorio de las empresas de software ecuatorianas, realizado por la *Escuela Politécnica del Litoral*.- *“Aspectos de la calidad y dificultades de la gestión de proyectos de Software”*.

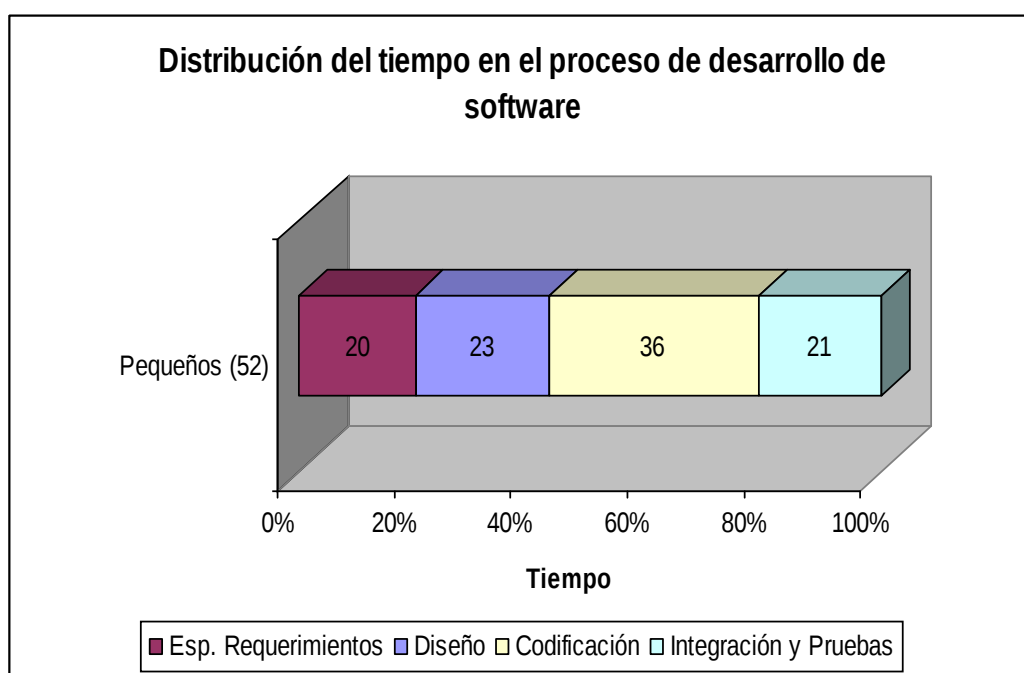


Figura 5-1

Estadísticas del desarrollo de software

Como vemos lo óptimo para un sistema pequeño que toma de 1 a 6 meses, (nos ubicamos en este rango debido a nuestro horarios irregulares y demás riesgos), debió ser mayor al 20% de nuestro cronograma.

Nuestro tiempo de prueba ha sido de pocas semanas, se planificó un 16% del cual se ha llevado a cabo un poco más del 10% de todo nuestro calendario de actividades, tratamos de enfocarnos a la interactividad entre los diferentes módulos.

5.1. Pruebas de Desarrollo.

Las pruebas de desarrollo se enfocan en el nivel del código, integración de módulos, además de maximizar la cobertura de las pruebas en todo el código, estas fueron realizadas a medida que un nuevo módulo se integraba al sistema general, observando su interactividad y si el manejo de librerías de cierto módulo no afectaba a otro que estaba funcionando ya de manera correcta en el servidor Web.

El problema más frecuente fue que ciertos módulos que funcionaban localmente al integrarlos no se levantaban en nuestro servidor, teniendo que realizar múltiples algoritmos cada vez que integrábamos el nuevo módulo, es decir se revisaban todos los procesos posibles para que el error no ocurra, en algunos casos el error era a causa de librerías que no habían sido situadas en el servidor, otro caso que se presentó fue que los módulos antes de ser integrados trabajaban con una base de datos local, y al integrarlo en el

servidor Web y trabajar con la base de datos principal, la data no estaba actualizada o la estructura de la base de datos no era exactamente igual.

5.1.1 Pruebas de Calidad Interna.

Aquí se evaluó la interfaz de usuario, teniendo como puntos evaluados la compatibilidad entre módulos y la funcionalidad del sistema como un todo, y se realizó además el algoritmo de instalación general. Esta prueba se la denomina Alfa y los usuarios evaluadores fueron los propios desarrolladores, aunque intercambiando módulos, es decir nuestro módulo fue evaluado por el grupo de Costos y nosotros evaluamos el de ellos, esto a nivel de interfaz, la retroalimentación fue importante y como resultados obtuvimos que el factor común fue deficiencias como obviar títulos de las páginas o usar palabras poco descriptivas cuando se listaba información, la interoperabilidad entre módulos tuvieron muchos tropiezos, ya que no se sabía el momento certero para invocar pantallas de otros módulos.

5.1.2 Pruebas con Usuarios.

Estas pruebas son conocidas como Beta y se las realizan con usuarios actuales o potenciales de la aplicación. Estas pruebas son las únicas que realmente dan información acerca del éxito o fracaso. Estas pruebas no fueron realizadas en su totalidad con el proyecto integrado, pero la interfaz de mantenimiento que es parte de nuestro módulo de integración,

parametrización y configuraciones fue probado por un grupo de personas que son parte de una empresa de software de la ciudad de guayaquil, de esta manera retroalimentamos nuestras falencias y obteniendo el visto bueno después de 1 semana de prueba y mejoras.

5.2 Métricas Consideradas.

El módulo de Integración, Parametrización y Configuraciones tiene una labor casi intangible en cuanto a codificación o desarrollo de interfaces, así que las métricas aplicadas en los siguientes párrafos se ha considerado la labor de integración solamente, evaluando principalmente tiempos de rendimiento del servidor, control de cambios y versiones y la administración de los mismos.

- **Facilidad de la Instalación.**

Se diseñó un manual de instalación, detallando los requerimientos hardware y versiones software necesarias para una correcta implantación del sistema, además detallando los posibles errores y soluciones.

- **Facilidad de Cambio.**

La forma de desarrollo utilizado en SGPI, facilita los cambios ya que cada parte del proyecto esta dividido según su funcionalidad y objetivo en diferentes archivos: jsp, xhtml, js, etc.

- **Control de Versiones**

Se tiene un estricto control de cambios en la estructura de bases de datos, así como el control de las versiones de la misma, probando todas las funcionalidades cada vez que se utilizaba una nueva versión y distribuyéndola a todo el grupo de desarrollo.

- **Evaluación de Rendimiento.**

Se realizaron pruebas, usando tres servidores con distintas características, evaluando tiempos de conexión, carga y respuesta (en segundos).

Características	Procesador	Memoria	Disco Duro	Tiempo Conexión	Tiempo Carga	Tiempo Respuesta
Servidor 1	P IV 2.4 GHz.	1 GB.	160 GB.	8	4	2
Servidor 2	P IV 2.4 GHz.	1 GB.	80 GB.	6	2	2
Servidor 3	Core 2 Dúo 2 GHz.	2 GB.	80 GB.	2	2	0.5

Tabla 5-1

Pruebas sobre hardware

Obteniendo mejores resultados en el servidor 3, aunque en el proyecto utilizamos el servidor 1.

CONCLUSIONES

1. Organización del Personal.- Uno de los alcances del módulo de integración, era la organización del personal que desarrollaba el proyecto, al término de la fase final de integración la mayor dificultad fue que el grupo de trabajo era muy heterogéneo en cuanto al nivel de conocimientos y al nivel investigativo, lo cual es comprensible porque las personas tienen intereses diferentes.

Pero también es válido destacar la puntualidad de la mayoría de los grupos en las reuniones y el empeño por sacar adelante el proyecto, las reuniones del grupo de trabajo fueron solo los fines de semana, dio buenos resultados fijar después de las 8 semanas de trabajo un día en la semana donde los líderes conversaban los avances y se proponían límites para las nuevas actividades.

2. Pruebas De Calidad Del Software.- La demora en la entrega de las actividades, el retraso de los hitos son características moda en el desarrollo del producto de software, y este no fue la excepción, el período de de análisis tomó mucho más tiempo del esperado y el período de pruebas fue casi nulo, sabiendo de antemano que un sistema que no es probado lo suficiente carece de excelencia.

3. Comunicación de todo el grupo de trabajo.- La tecnología de la comunicación fue un pilar importante en el desarrollo de nuestro proyecto, los correos electrónicos y el *chat* nos mantuvieron comunicados a lo largo de estos meses de desarrollo y además ayudó a que todas las peticiones, sugerencias y comunicados queden asentados de manera formal, ya que los acuerdos de manera verbal muchas veces quedaban en el olvido y eran motivo de discusión. Enviar una copia de cada correo electrónico enviado al coordinador del curso de graduación también sirvió para que esté enterado de los avances del proyectos y sus posibles estancamientos.

4. Comunicación interna de los grupos de trabajo.- Teniendo doce grupos conformados reuniendo a 31 personas, se designó a un representante por grupo para que la toma de decisiones sólo se hiciera ente 12 personas lo cual era más manejable, aunque hubieron dificultades porque muchas veces uno de los integrantes del grupo desconocía las decisiones que había tomado su líder y viceversa, existiendo confusión y malos entendidos, aunque se fueron superando con el tiempo.

5. Infraestructura y equipos de computación.- Aunque esto fue contemplado en el análisis de riesgo, lo mencionamos porque fue uno

de nuestros principales problemas, nuestro horario de actividades era demasiado complejo y no contar con el número adecuado de equipos afectó el aprendizaje, los últimos meses las reuniones integración se realizaron en otra locación que carecía de un buen sistema de red.

6. Tema del Proyecto.- Aunque es conocido que los ingenieros somos sinónimos de ingenio y podemos realizar cualquier proyecto propuesto obviando las limitantes y los visibles riesgos, el tema del proyecto que se realizará durante casi un año de trabajo debería ser escogido por todo el recurso humano, adecuarse más a las necesidades actuales, hacer algo atípico. Una de los principales problemas de nuestro proyecto fue la ausencia de un usuario final real, como ingenieros conocemos y/o trabajamos en el desarrollo del software pero un *pseudocliente* externo hubiera mejora la calidad del software, porque innegablemente nuestro producto lo usará un cliente verdadero.

RECOMENDACIONES.

- 1. Organización del personal.-** Entonces cuando se tiene un proyecto con un tiempo límite de 9 meses, conseguir que el grupo sea homogéneo, retrasa el tiempo de entrega y se ve reflejado en la calidad de software, posiblemente se podría evitar esto si se toma más tiempo para la inducción previo a la matriculación de los estudiantes en un determinado tópico, indicando los alcances de la materia, el grado de dificultad, etc.
- 2. Pruebas De Calidad Del Software.-** Puede ser que el sistema esté integrado pero el período de pruebas debe ser prioritariamente importante antes de la entrega final, el producto puede estar terminado pero no probado de manera adecuada ni acorde a ninguna métrica ni metodología aplicada eficazmente.
- 3. Infraestructura y Equipos de Computación.-** El número de equipos debe ser igual al número de desarrolladores, y el sistema de red con el servidor no debe de sufrir falencias por largo tiempo, porque trabajar un sistema integrado de manera local hace más fácil la omisión de fallos.

4. Tema del Proyecto.- Sería de más incentivo desarrollar algo nuevo para la comunidad con resultados para usuarios palpablemente reales y poner a discusión el tema a elegir.

5. Modularidad del Proyecto.- Debería tomarse en cuenta ciertos aspectos para la división de un sistema en módulos, tomando en cuenta la dificultad de integración, algunos módulos compartían muchas similitudes donde la división trajo más dificultades que beneficios, aquellas partes del proceso que tengan la misma finalidad deberían ser mantenidas en un solo módulo.

ANEXOS

- **RESUMEN DE LA ADMINISTRACIÓN DEL RIESGO**
- **MODELO ENTIDAD RELACIÓN**
- **DICCIONARIO DE DATOS DE TABLAS GENERALES DE SGPI**
- **CASOS DE USO**

RESUMEN DE LA ADMINISTRACIÓN DEL RIESGO

ADMINISTRACION DEL RIESGO							
TIPO	DEFINICIÓN	AFECTA	P	I	N.R.	ESTRATEGIA	FACTORES
TECNOLÓGICO	⌘ Equipos de Computación no estén disponibles..	Proyecto	3	2	6	Organizar lugares de trabajo dentro o fuera de la Universidad.	Falta de presupuesto y administración.
	⌘ Daños imprevistos del hardware.	Producto	1	0	0	Dar mantenimiento frecuente a los equipos Establecer políticas para la realización de backups	El laboratorio asignado es además usados por otras personas.
	⌘ Falta de backup de los módulos desarrollados.	Producto	1	3	3		Mala Administración de la Información.
PERSONAL	⌘ Reclutamiento de Personal Aleatorio.	Proyecto	2	1	2	Cursos de inducción acerca del contenido y alcances del curso	Falta de orientación pre inscripción
	⌘ Asignación de roles aleatorio.	Producto	3	2	6	Colaboración mutua por parte de los grupos.	Sorteo para designar desarrollo de módulos
	⌘ Capacitación Inadecuada.	Producto	1	2	2	Buscar otras fuentes de información.	Contenido del curriculum deficiente
	⌘ Rendimiento bajo durante el desarrollo de la capacitación.	Producto	1	3	3	Establecer metodologías prácticas	Metodología de enseñanza
	⌘ Crear la imagen que la integración es la panacea.	Proyecto	2	3	6	Mostrar y analizar estadísticas	Falta de experiencia
	⌘ Esperar que todo este terminado para comenzar a integrar.	Proyecto	0	3	0	Explicar el modelo incremental de integración	Falta de experiencia
ORGANIZACIONAL	⌘ Indisponibilidad de reunirse el grupo de desarrolladores.	Proyecto	3	3	9	Mantener una alta comunicación vía e-mail, chat.	Horarios diferentes de trabajo en diferentes lugares de la ciudad.
	⌘ Indisponibilidad de reunirse el grupo de desarrolladores con el grupo integrador	Proyecto	2	3	6	Mantener una alta comunicación vía e-mail, chat, además solo deberan tener un representante por grupo(responsable)	Horarios diferentes de trabajo en diferentes lugares de la ciudad.
REQUERIMIENTOS	⌘ Mala interpretación de los requisitos por parte de los desarrolladores.	Producto	1	3	3	Mantener comunicación continua con los desarrolladores.	Comunicación Deficiente
	⌘ Modificación de Requerimientos a ultimo momento por parte de la empresa reguladora	Producto	2	3	6	Reuniones, entrevistas, diseñar esquemas que mejoren la comprensión.	Levantamiento de información deficiente
	⌘ Ausencia de investigaciones y entrevistas al negocio.	Producto	1	3	3	Realizar entrevistas al área de sistemas en nuestro entorno	Falta de Autogestión.
ESTIMACIÓN	⌘ Tiempo limitado para el levantamiento de información	Proyecto	3	3	9	Delegar funciones dentro del grupo para investigar mayor volumen de información.	Planificación y Cronogramación

Donde :

P = Prioridad

I = Impacto

N.R. = Nivel de Riesgo

2.1 Diccionario de Datos

2.1.1 DESCRIPCIÓN DE LAS TABLAS DE USO GENERAL

Las tablas de Uso General son aquellas que contienen información de interés para más de un módulo, aunque también existen tablas propias del módulo de integración que nos servirán para llevar controles.

Los campos Fecha_Ingreso Estacion_Ingreso Usuario_Ingreso Fecha_Modificacion Estacion_Modificacion Usuario_Modificacion son con fines de control y auditoria, se encuentran en la mayoría de las tablas.

NOMBRE DE LA TABLA: GEN_CARGOS

PROPÓSITO: Almacena los cargos o puestos de trabajo que tiene una empresa (Presidente, Supervisor, Secretaria, Vendedor,..)

GEN_CARGOS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_CARGO	S	PK	NUMBER(3)	Identificador del Cargo
CODIGO_ACTIVIDAD_SEGURO	S		VARCHAR2(15)	Identificador de la actividad(Descripción)
NOMBRE_CARGO	N		VARCHAR2(60)	Descripción del cargo
DOMINIO	S		VARCHAR2(60)	Nombre del dominio
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
ESTACION_INGRESO	S		VARCHAR2(20)	Estación donde se ingreso la información
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó la transacción
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación
ESTACION_MODIFICACION	S		VARCHAR2(20)	Estación donde se produjo la modificación

NOMBRE DE LA TABLA: GEN_TIPOS_IDENTIFICACION

PROPÓSITO: Almacena los distintos tipos de identificación que puede tener una clase de persona. (Cédula, RUC, Pasaporte,..)

GEN_TIPOS_IDENTIFICACION

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_TIPO_IDENTIFICACION	S	PK	NUMBER(3)	Identificador del Tipo de Identificación
DESCRIPCION	S		VARCHAR2(60)	Descripción del tipo de identificación
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
ESTACION_INGRESO	S		VARCHAR2(20)	Estación donde se ingreso la información
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó la transacción
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación
ESTACION_MODIFICACION	S		VARCHAR2(20)	Estación donde se produjo la modificación

NOMBRE DE LA TABLA: GEN_PROFESIONES

PROPÓSITO: Almacena todas las posibles profesiones que puede tener una persona (Ing., Ab., Ec...)

GEN_PROFESIONES

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_PROFESION	S	PK	NUMBER(3)	Identificador de la profesión
NOMBRE_PROFESION	S		VARCHAR2(100)	Nombre de la profesión
ABREVIATURA	S		VARCHAR2(6)	Abreviatura global
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
ESTACION_INGRESO	S		VARCHAR2(20)	Estación donde se ingreso la información
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó la transacción
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación
ESTACION_MODIFICACION	S		VARCHAR2(20)	Estación donde se produjo la modificación

NOMBRE DE LA TABLA: GEN_ESTADOS_CIVIL

PROPÓSITO: Almacena todos los tipos de estados civiles que puede tener una persona. (Soltero, casado,..)

GEN_ESTADOS_CIVIL

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_ESTADO_CIVIL	S	PK	NUMBER(3)	Identificador Estado Civil
DESCRIPCION	S		VARCHAR2(30)	Nombre
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
ESTACION_INGRESO	S		VARCHAR2(20)	Estación donde se ingreso la información
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación
ESTACION_MODIFICACION	S		VARCHAR2(20)	Estación donde se produjo la modificación

NOMBRE DE LA TABLA: GEN_PAISES

PROPÓSITO: Contendrá el detalle del país en el cual reside una persona.

GEN_PAISES

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_PAIS	S	PK	NUMBER(3)	Identificador del País
NOMBRE_PAIS	S		VARCHAR2(100)	Nombre del País
NACIONALIDAD	N		VARCHAR2(100)	Nacionalidad
CONTINENTE	S		VARCHAR2(30)	Continente al que pertenece
INDICATIVO_PAIS	S		NUMBER(6)	Código indicativo del país.
SIGLAS	S		VARCHAR2(15)	Siglas globales con la que se identifica un país
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
ESTACION_INGRESO	S		VARCHAR2(20)	Estación donde se ingreso la información
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación
ESTACION_MODIFICACION	S		VARCHAR2(20)	Estación donde se produjo la modificación

NOMBRE DE LA TABLA: GEN_PROVINCIAS

PROPÓSITO: Contendrá el detalle de la provincia en el cual reside una persona.

GEN_PROVINCIAS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_PROVINCIA	S	PK	NUMBER(3)	Identificador de la Provincia
CODIGO_PAIS	S	FK/PK	NUMBER(3)	Tabla GEN_PAISES
NOMBRE_PROVINCIA	S		VARCHAR2(100)	Nombre de la Provincia
SUFIJO_PROVINCIA	N		VARCHAR2(6)	Abreviación(UIO)
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
ESTACION_INGRESO	S		VARCHAR2(20)	Estación donde se ingreso la información
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación
ESTACION_MODIFICACION	S		VARCHAR2(20)	Estación donde se produjo la modificación

NOMBRE DE LA TABLA: GEN _ CIUDADES

PROPÓSITO: Contendrá el detalle de las ciudad en el cual reside una persona.

GEN_CIUDADES

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_CIUADAD	S	PK	NUMBER(3)	Identificador de la Ciudad
CODIGO_PROVINCIA	S	FK/PK	NUMBER(3)	Tabla GEN_PROVINCIAS
CODIGO_PAIS	S	FK/PK	NUMBER(3)	Tabla GEN_PAISES
NOMBRE_CIUADAD	S		VARCHAR2(100)	Nombre de la Ciudad
SUFIJO_CIUADAD	N		VARCHAR2(6)	Abreviación(UIO)
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
ESTACION_INGRESO	S		VARCHAR2(20)	Estación donde se ingreso la información
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación
ESTACION_MODIFICACION	S		VARCHAR2(20)	Estación donde se produjo la modificación

NOMBRE DE LA TABLA: GEN_PARROQUIAS

PROPÓSITO: Contendrá el detalle de la parroquia en el cual reside una persona.

GEN_PARROQUIAS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_PARROQUIA	S	PK	NUMBER(3)	Identificador de la Parroquia
CODIGO_CIUDAD	S	FK/PK	NUMBER(3)	Tabla GEN_CIUDADES
CODIGO_PROVINCIA	S	FK/PK	NUMBER(3)	Tabla GEN_PROVINCIAS
CODIGO_PAIS	S	FK/PK	NUMBER(3)	Tabla GEN_PAISES
NOMBRE_PARROQUIA	S		VARCHAR2(100)	Nombre de la Parroquia
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
ESTACION_INGRESO	S		VARCHAR2(20)	Estación donde se ingreso la información
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación
ESTACION_MODIFICACION	S		VARCHAR2(20)	Estación donde se produjo la modificación

NOMBRE DE LA TABLA: GEN_ESTADOS

PROPÓSITO: Almacena los estados de los diferentes procesos que hay en todos los módulos (activo, inactivo, en espera, pagado, pendiente, etc.)

GEN_ESTADOS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_ESTADO	S	PK	VARCHAR2(2)	Identificador de Estado
DESCRIPCION	S	FK/PK	VARCHAR2(20)	Descripción de Estado
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_TIPO_PROYECTO

PROPÓSITO: Almacena los tipos de proyecto que puede tener un proyecto.

GEN_TIPO_PROYECTO

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_TIPO_PROYECTO	S	PK	NUMBER(3)	Identificador de Tipo de Proyecto
DESCRIPCION	S	FK/PK	VARCHAR2(30)	Descripción del tipo de proyecto
CODIGO_ESTADO	S	FK/PK	VARCHAR2(2)	Tabla GEN_ESTADOS
COSTO_MAXIMO	S		NUMBER(10,2)	Costo máximo de un tipo de proyecto
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_TIPO_RELACION

PROPÓSITO: Almacena el tipo de personal que puede tener una empresa (Por Horas, Estables, Practicantes,...)

GEN_TIPO_RELACION

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_TIPO_RELACION	S	PK	NUMBER(2)	Identificador de la Relación
NOMBRE	S		VARCHAR2(30)	Descripción del tipo de relación
DESCRIPCION	N		VARCHAR2(10)	Descripción del tipo de relación
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_MODULOS

PROPÓSITO: Determina los diferentes módulos y submódulos de la aplicación.

GEN_MODULOS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_MODULO	S	PK	VARCHAR2(3)	Identificador del módulo
NOMBRE	S		VARCHAR2(60)	Nombre del módulo
TIENE_MODULO	N	FK	VARCHAR2(3)	Si pertenece a un módulo (submódulo)
DESCRIPCION	N		VARCHAR2(10)	Breve descripción de lo que hace
CODIGO_USUARIO_RESP	S		VARCHAR2(15)	Usuario responsable del módulo
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_ROLES_MODULOS

PROPÓSITO: Determina los diferentes roles que van a tener aquellos usuarios que manejan los diferentes módulos de la aplicación.

GEN_ROLES_MODULOS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_ROL	S	PK/FK	VARCHAR2(8)	Tabla GEN_ROLES
CODIGO_MODULO	S	PK/FK	VARCHAR2(3)	Tabla GEN_MODULOS

NOMBRE DE LA TABLA: GEN_USUARIOS_MODULO

PROPÓSITO: Determina los diferentes módulos que van a tener los usuarios de la aplicación.

GEN_USUARIOS_MODULO

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_USUARIO	S	PK	VARCHAR2(15)	Identificador del usuario
CODIGO_MODULO	S	PK/FK	VARCHAR2(3)	Tabla GEN_MODULOS
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_IDIOMAS

PROPÓSITO: Almacena los idiomas que se usarán en la emisión de los mensajes (español o inglés)

GEN_IDIOMAS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_IDIOMA	S	PK	NUMBER(3)	Identificador del idioma
DESCRIPCION	S	PK/FK	VARCHAR2(50)	Descripción del idioma
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_MENSAJES

PROPÓSITO: Guarda los distintos mensajes de errores que puedan darse al momento de ingreso de datos a la base (Campo debe ser ingresado, Parámetro no existe,...)

GEN_MENSAJES

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_MENSAJE	S	PK	NUMBER(3)	Identificador del mensaje
CODIGO_IDIOMA	S	PK/FK	NUMBER(3)	Tabla GEN_IDIOMAS
DESCRIPCION	S		VARCHAR2(500)	Descripción del mensaje
CODIGO_ORACLE	N		VARCHAR2(20)	Código de error de oracle
ADICIONAL	N		VARCHAR2(1)	Si es un código adicional a la base de datos oracle
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(30)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(30)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_RESPONSABLES

PROPÓSITO: Detalla los datos generales y específicos de las personas responsables de actividades dentro del proyecto o personas encargadas de hacer pedidos, etc. En fin se detalla datos de gente responsable de algo dentro de Gestor de Proyectos

GEN_RESPONSABLES

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_RESPONSABLE	S	PK	NUMBER(3)	Identificador de responsable
CODIGO_PERSONA	S	PK/FK	NUMBER(6)	Tabla GEN_PERSONAS
CODIGO_ESTADO	S	PK/FK	VARCHAR2(2)	Tabla GEN_ESTADOS
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(30)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(30)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_USUARIOS_ROLES

PROPÓSITO: Determina los diferentes roles que van a tener los usuarios de la aplicación.

GEN_USUARIOS_ROLES

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_USUARIO	S	PK	VARCHAR2(15)	Identificador del usuario
CODIGO_PERSONA	S	PK/FK	NUMBER(6)	Tabla GEN_PERSONAS
CODIGO_ESTILO	S	PK/FK	VARCHAR2(8)	Tabla PAS_ESTILOS
FECHA_EXP_CLAVE	S		DATE	Fecha en que se expira la clave
USUARIO_PRIMER_INGRESO	S		VARCHAR2(1)	Si el usuario es de primer ingreso
CUENTA_MAIL	S		VARCHAR2(30)	Correo electrónico
NOMBRE	S		VARCHAR2(10)	Nombre del usuario
USUARIO_CLAVE	S		VARCHAR2(8)	Clave del usuario
CODIGO_ESTADO	S	PK/FK	VARCHAR2(2)	Tabla GEN_ESTADOS
FECHA_INICIO	S		DATE	Fecha inicial del usuario
FECHA_FIN	S		DATE	Fecha final del usuario
OBSERVACIONES	S		VARCHAR2(200)	Observaciones

FECHA_INGRESO	S	DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S	VARCHAR2(15)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION	S	DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S	VARCHAR2(15)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_ESTADOS_TABLAS

PROPÓSITO: Almacena todas las tablas con un estado específico (activo, inactivo, etc.)

GEN_ESTADOS_TABLAS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
NOMBRE_TABLA	S	PK	VARCHAR2(100)	Nombre de la Tablas del Modelo
ESTADO _ REGISTRO	S	PK	VARCHAR2(1)	Estado Identificador
DESCRIPCION_ESTADO	S		VARCHAR2(60)	Descripción del Estado
DESCRIPCION_CLIENTE	N		VARCHAR2(60)	Descripción del Cliente
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_SERVIDORES

PROPÓSITO: Almacenan los datos de todos los servidores o equipos que se usarán en el Proyecto

GEN_SERVIDORES

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CÓDIGO _ SERVIDOR	S	PK	NUMBER(2)	Identificador Servidor de equipo.
NOMBRE_SERVIDOR	S		VARCHAR2(30)	Nombre de la máquina, Net Bios
DIRECCION_IP	S		VARCHAR2(30)	0000.0000.0000.0000
DOMINIO	S		VARCHAR2(60)	Nombre del dominio
FECHA_INGRESO	S		DATE	Fecha cuando se ingresa la transacción
USUARIO_INGRESO	S		NUMBER(4)	Código de usuario que ingresa la transacción

NOMBRE DE LA TABLA: GEN_PERSONAS

PROPÓSITO: Almacena todas las personas, colaboradores que estarán involucradas en los distintos proyectos.

GEN_PERSONAS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_PERSONA	S	PK	NUMBER(6)	Identificador de la persona.(secuencial)
CODIGO_TIPO_RELACION	S	FK	NUMBER(3)	Tabla GEN_TIPO_RELACION
CODIGO_TIPO_IDENTIFICACION	S	FK	NUMBER(3)	Tabla GEN_TIPO_IDENTIFICACION
NUMERO_IDENTIFICACION	S		VARCHAR2(15)	Numero de la cedula, pasaporte, dependiendo del campo Tipo de Identificación
PRIMER_NOMBRE	S		VARCHAR2(20)	Primer nombre
SEGUNDO_NOMBRE	N		VARCHAR2(20)	Segundo Nombre
PRIMER_APELLIDO	S		VARCHAR2(20)	Primer Apellido
SEGUNDO_APELLIDO	S		VARCHAR2(20)	Segundo Apellido
FECHA_DE_NACIMIENTO	N		DATE	Fecha de Nacimiento
CODIGO_ESTADO_CIVIL	S	FK	NUMBER(3)	Tabla GEN_ESTADOS_CIVIL
IDENTIFICACION_MILITAR	N		VARCHAR2(10)	Numero de identificación militar
CERTIFICADO_VOTACION	N		VARCHAR2(60)	Número de Certificado de Votación
NUMERO_CARGAS_FAMILIAR	N		NUMBER(2)	Número de Cargas Familiares
CODIGO_PAIS	S	FK	NUMBER(3)	Tabla GEN_PAISES
CODIGO_PROVINCIA	S	FK	NUMBER(3)	Tabla GEN_PROVINCIAS
CODIGO_CIUDAD	S	FK	NUMBER(3)	Tabla GEN_CIUDADES
CODIGO_PARROQUIA	S	FK	NUMBER(3)	Tabla GEN_PARROQUIAS
FECHA_INGRESO_LABORAL	S		DATE	Fecha en que ingreso al Dpto. Sistemas
ESTA_EN_ROL	N		VARCHAR2(1)	S/N
FECHA_INGRESO_ROL	S		DATE	Fecha de ingreso al rol
DIAS_OTRAS_ANTIGUEDADE	N		NUMBER(4)	Antigüedad
FECHA_SALIDA	S		DATE	Fecha en que salio del Dpto.
TIPO_AFILIACION_SINDIC	N		VARCHAR2(10)	Descripción Sindicato
LUGAR_DE_TRABAJO	N		VARCHAR2(10)	Lugar específico en el Dpto.
CODIGO_CARGO	S	FK	NUMBER(3)	Tabla GEN_CARGOS
MARCA_RELOJ	N		VARCHAR2(1)	Campo Pendiente
HORAS_DIARIAS_TRABAJO	N		NUMBER(5,2)	Numero de horas trabajadas
CODIGO_ESTADO	S	FK	VARCHAR2(1)	Tabla GEN_ESTADOS

DIAS_TRABAJADOS_MES	N		NUMBER(3)	Numero de Días trabajados durante el mes
DIRECCION	S		VARCHAR2(30)	Dirección Domiciliaria
TELEFONO	N		VARCHAR2(10)	Teléfono
MAIL	N		VARCHAR2(30)	Correo electrónico
CODIGO_PROFESION	S	FK	NUMBER(6)	Tabla GEN_PROFESIONES
CODIGO_PROYECTO_ACTUAL	N		VARCHAR2(100)	Proyecto en que trabaja
CODIGO_PROYECTO_ANTERIOR	N		VARCHAR2(100)	Ultimo proyecto
PESO_PROYECTO	N		NUMBER(3)	Impacto del proyecto
PESO_PRIVILEGIO	N		NUMBER(2)	Privilegios
TIPO_PERSONA	S		VARCHAR2(3)	Tipo de Persona
SEXO	S		VARCHAR2(1)	F o M
FIRMA	N		BLOB	Imagen de la firma
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
ESTACIÓN_INGRESO	S		VARCHAR2(10)	Estación donde se ingreso la información
FECHA_MODIFICACIÓN	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación
ESTACIÓN_MODIFICACIÓN	S		VARCHAR2(10)	Estación donde se produjo la modificación

NOMBRE DE LA TABLA: GEN _ PROYECTOS

PROPÓSITO: Es una de las principales entidades donde se almacenan todos los datos concernientes a un proyecto, como es el responsable del proyecto, las fechas inicio – fin, los recursos que ocuparán, su dimensión, etc.

GEN _ PROYECTOS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_PROYECTO	S	PK	NUMBER(6)	Identificador del proyecto(secuencial)
CODIGO_AREA	S	PK/FK	NUMBER(3)	Tabla GEN_AREAS
CODIGO_TAM_REQUERIMIENTO	S	FK	NUMBER(10,2)	Tabla GRE_TAM_REQUERIMIENTO
CODIGO_TIPO_REC_INFORMAT	S	FK	NUMBER(3)	Tabla GEN_TIPOS_RECURSOS
CODIGO_TIPO_PROYECTO	S	FK	NUMBER(3)	Tabla GEN_TIPO_PROYECTO
CODIGO_ESTADO	S	FK	VARCHAR2(2)	Tabla GEN_ESTADOS
NOMBRE_PROYECTO	S		VARCHAR2(100)	Nombre que se le ha dado al proyecto

TIEMPO_PROYECTO	N	NUMBER(4)	Tiempo de duración del proyecto
TOTAL_RECURSO_HUMANO	N	NUMBER(3)	Recurso humano implicado en el proyecto
TOTAL_RECURSO_INFORMATICO	N	NUMBER(3)	Recurso informático implicado en el proyecto
PESO_PROYECTO	N	NUMBER(4)	Peso que tiene un proyecto
FECHA_INICIO		DATE	Fecha de inicio del proyecto
FECHA_FIN		DATE	Fecha final del proyecto
NOMBRE_USUARIO		VARCHAR2(10)	Nombre del usuario
NOMBRE_TERMINAL		VARCHAR2(10)	Nombre de la Terminal
CODIGO_TIPO		NUMBER(10,2)	Código del tipo de proyecto
OBSERVACION		VARCHAR2(10)	Observación
CODIGO_RESPONSABLE		NUMBER(3)	Código del responsable del proyecto
SECUENCIAL		NUMBER(4)	Secuencial del proyecto
FECHA_INICIO_REAL		DATE	Fecha de inicio real del proyecto
FECHA_FIN_REAL		DATE	Fecha de fin real del proyecto
PORCENTAJE_AVANCES		NUMBER(4,3)	Porcentaje de avance del proyecto
PESO_PORCENTUAL		NUMBER(4,3)	Peso porcentual del proyecto
CODIGO_CATEGORIA	FK	NUMBER(3)	Tabla CEH_CATEGORIA
CODIGO_COSTO	FK	NUMBER(3)	Tabla CYP_COSTO_PROYECTO
CODIGO_REQUERIMIENTO	FK	NUMBER	Tabla GRE_REQUERIMIENTO
FECHA_INGRESO		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO		VARCHAR2(15)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION		VARCHAR2(15)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_PROCESOS

PROPÓSITO: Almacena los distintos procesos transaccionales que se puedan presentar en una empresa; como por ejemplo: Pago a Proveedores, Hoja de Cotización, Orden de Compra, etc.

GEN_PROCESOS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_PROCESO	S	PK	NUMBER(6)	Identificador del proceso
DESCRIPCION	S		VARCHAR2(60)	Descripción del proceso
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_APLICACIONES

PROPÓSITO: Guarda las distintas aplicaciones del sistema.

GEN_APLICACIONES

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_APLICAION	S	PK	VARCHAR2(3)	Identificador de la aplicación
CLASE	S	PK/FK	VARCHAR2(1)	Clase de la aplicación
NOMBRE	S		VARCHAR2(100)	Nombre de la aplicación
INTEGRA_APLICAION	N		VARCHAR2(1)	Si es una aplicación que integra el sistema
DESCRIPCION	S		VARCHAR2(100)	Descripción de la aplicación
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN _ ACTIVIDADES

PROPÓSITO: Almacena las distintas actividades que se pueden presentar en cada uno de los procesos transaccionales como por ejemplo: El proceso Pago a Proveedores puede tener muchas actividades como pueden ser Emisión de órdenes , Anulación de Ordenes, etc.

GEN _ ACTIVIDADES

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CÓDIGO_ACTIVIDAD	S	PK	NUMBER(3)	Identificador de la actividad
CODIGO_PROCESO	S	FK/PK	NUMBER(6)	Tabla GEN_PROCESOS
DESCRIPCION	S		VARCHAR2(60)	Descripción de la Actividad
ACTIVO	S		VARCHAR2(1)	Si la actividad está activa
EJECUCION	S		VARCHAR2(1)	Si la actividad se puede ejecutar
CODIGO_AREA	S	FK	NUMBER(3)	Tabla GEN_AREAS
CODIGO_APLICACION	S	FK	VARCHAR2(3)	Código de Aplicación
CODIGO_PROGRAMA	S		VARCHAR2(15)	Código de programa
DURACION_HORAS	S		NUMBER(5,2)	Duración de la actividad
FRECUENCIA	S		VARCHAR2(60)	Frecuencia de la actividad
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(30)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(30)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_AREAS

PROPÓSITO: Almacena todos las áreas de departamentos de una empresa (Financiero, RR.HH,..)

GEN_AREAS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_AREA	S	PK	NUMBER(3)	Identificador del área
NOMBRE_AREA	S		VARCHAR2(10)	Nombre del área
CODIGO_RESPONSABLE	S	FK	NUMBER(3)	Tabla GEN_RESPONSABLE (Persona Responsable del Área)
ACTIVO	S		VARCHAR2(1)	Si está activa el área
UBICACIÓN	S		VARCHAR2(30)	Descripción de la Ubicación
REPRESENTANTE	S		VARCHAR2(30)	Representante del Área
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(30)	Código de usuario que ingresa la transacción

FECHA_MODIFICACION	S	DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S	VARCHAR2(30)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_REPORTES

PROPÓSITO: Almacena los reportes que generará el sistema haciendo referencia a la empresa y aplicación al que pertenece.

GEN_REPORTES

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_REPORTES	S	PK	VARCHAR2(6)	Identificador del reporte
CODIGO_APLICACION	S	PK/FK	VARCHAR2(3)	Tabla GEN_APLICACIONES
NOMBRE_REPORTES	S	PK/FK	VARCHAR2(60)	Nombre del reporte
FORMATO_REPORTES	S		VARCHAR2(30)	Descripción de formato
TIPO_EJECUCION	S		VARCHAR2(1)	Forma de ejecución
TIPO_DESTINO	S		VARCHAR2(1)	Tipo de destino
ORIENTACION	S		VARCHAR2(1)	Orientación
TIPO_LETRA	S		VARCHAR2(30)	Tipo de letra
ANCHO_LETRA	S	PK/FK	NUMBER(2)	Ancho de letra
MODO_EJECUCION	S		VARCHAR2(1)	Modo de ejecución
NUMERO_COPIAS	S		NUMBER(2)	Número de copias
CODIGO_TEMPLATE	S		NUMBER(3)	Template Pre-Definidos
FECHA_INGRESO	S		DATE	Fecha en que se ingreso la transacción
USUARIO_INGRESO	S		VARCHAR2(15)	Código de usuario que ingresa la transacción
FECHA_MODIFICACION	S		DATE	Fecha en que se modificó un registro
USUARIO_MODIFICACION	S		VARCHAR2(15)	Código de usuario que realizó la modificación

NOMBRE DE LA TABLA: GEN_TIPO_RECURSOS

PROPÓSITO: Almacena los distintos tipos de recursos que puede requerir el sistema.

GEN_TIPO_RECURSOS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_TIPO_RECURSO	S	PK	NUMBER(3)	Identificador del tipo de recurso
NOMBRE	S		VARCHAR2(50)	Nombre del tipo de recurso
CODIGO_ESTADO	S	FK	VARCHAR2(2)	Tabla GEN_ESTADOS
FECHA_INICIO	S		DATE	Fecha de inicio del tipo de recurso
FECHA_FIN	S		DATE	Fecha final del tipo de recurso
COSTO	S		NUMBER(12,2)	Costo del tipo de recurso

NOMBRE DE LA TABLA: GEN_DET_RECURSO_INFORMATICO

PROPÓSITO: Almacena el detalle de los recursos informáticos.

GEN_DET_RECURSO_INFORMATICO

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_DET_REC_INFORMAT	S	PK	NUMBER(3)	Identificador del detalle del recurso informático
CODIGO_TIPO_REC_INFORMAT	S	FK	NUMBER(3)	Tabla GEN_TIPO_RECURSOS
DESCRIPCION	S		VARCHAR2(100)	Descripción del detalle de recurso informático
CANTIDAD_DISPONIBLE	S		NUMBER(3)	Cantidad disponible del recurso informático

NOMBRE DE LA TABLA: GEN_RECURSOS

PROPÓSITO: Almacena los distintos recursos que puede requerir el sistema.

GEN_RECURSOS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_RECURSO	S	PK	NUMBER(3)	Identificador del recurso
CODIGO_TIPO_RECURSO	S	FK	NUMBER(3)	Tabla GEN_TIPO_RECURSOS
CODIGO_PROVEEDOR	S	FK	NUMBER(10)	Tabla GPC_PROVEEDOR
CODIGO_ESTADO	S	FK	VARCHAR2(2)	Tabla GEN_ESTADOS
FECHA_INICIO	S		DATE	Fecha inicial del recurso
FECHA_FIN	S		DATE	Fecha final del recurso
DESCRIPCION	S		VARCHAR2(100)	Descripción del recurso

NOMBRE DE LA TABLA: GEN_RECURSOS_PROYECTO

PROPÓSITO: Almacena los distintos recursos del proyecto que puede requerir el sistema.

GEN_RECURSOS

CAMPO	OBLIGATORIO	PK/FK	TIPO/DATO	DESCRIPCION
CODIGO_RECURSO_PROYECTO	S	PK	NUMBER(3)	Identificador del recurso por proyecto
CODIGO_RECURSO	S	FK	NUMBER(3)	Tabla GEN_RECURSOS
CODIGO_PROYECTO	S	FK	NUMBER(10)	Tabla GEN_PROYECTO
CODIGO_AREA	S	FK	VARCHAR2(2)	Tabla GEN_PROYECTO

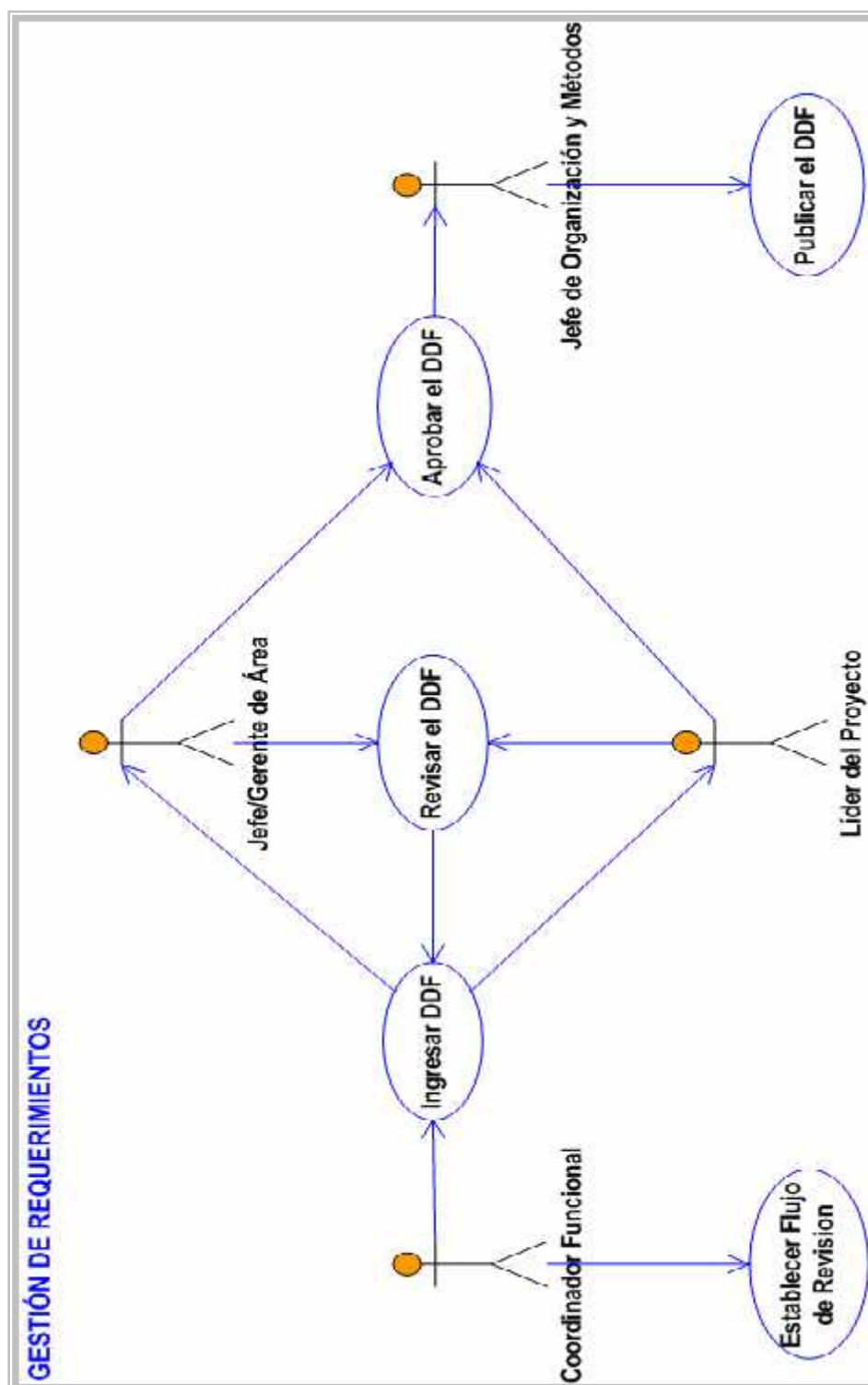


Figura 2-4.
Caso de Uso de la Gestión de Requerimientos

Nombre:	Gestión del Pre-Proyecto
Autor:	Grupo 6
Fecha:	14/07/2007
Descripción: Permite gestionar la elaboración de un pre-proyecto.	
Actores: Líder del Proyecto envía el DDF hacia el Gerente de Proyectos. Director de Sistemas asigna la prioridad del Proyecto. Gerente de Proyectos asigna la prioridad del Proyecto. Director de Área Solicitante asigna la prioridad del Proyecto. Gerente de Proyectos ingresa el requerimiento en la matriz de Proyectos. Gerente de Proyectos asigna el Código a un Proyecto. Gerente de Proyectos asigna el Líder de SIS a un Proyecto. Líder de Proyectos SIS revisa si el DDF se encuentra completo en cuanto a los requerimientos. Líder de Proyectos SIS elabora el Plan Técnico del Proyecto. Líder de Proyectos SIS elabora el Cronograma del Proyecto. Líder de Proyectos SIS envía el Plan Técnico al Gerente de Proyectos. Líder de Proyectos SIS envía el Cronograma al Gerente de Proyectos. Gerente de Proyectos revisa el Plan Técnico. Gerente de Proyectos aprueba el Plan Técnico. Líder de Proyectos SIS envía el Plan Técnico al Líder de Proyecto. Líder del Proyecto revisa el Plan Técnico con los líderes funcionales si fuese necesario. Líder del Proyecto aprueba el Plan Técnico.	
Precondiciones: El Líder del Proyecto debe poseer un documento (DDF) de gestión de pre-proyecto para ingresarlo al sistema.	
Flujo Normal:	

1. El Líder del Proyecto pulsa un botón para enviar el DDF hacia el Gerente de Proyectos.
2. El SGPI muestra un API para el envío y grabación del DDF en la base de datos.
3. El Director de Sistemas pulsa un botón para asignar la prioridad del proyecto.
4. El Gerente de Proyectos pulsa un botón para asignar la prioridad del proyecto.
5. El Director de Área Solicitante pulsa un botón para asignar la prioridad del proyecto.
6. El SGPI muestra un API para el ingreso y grabación de la prioridad del proyecto en la base de datos.
7. El Gerente de Proyectos pulsa un botón para ingresar el requerimiento en la matriz de Proyectos.
8. El SGPI muestra un API para el ingreso y grabación del requerimiento en la matriz de proyectos de la base de datos.
9. El Gerente de Proyectos pulsa un botón para asignar el código a un proyecto.
10. El SGPI muestra un API para la asignación y grabación del código a un proyecto en la base de datos.
11. El Gerente de Proyectos pulsa un botón para asignar el Líder de SIS a un Proyecto.
12. El SGPI muestra un API para la asignación y grabación de un Líder de SIS a un proyecto en la base de datos.
13. El Líder de Proyectos SIS pulsa un botón para revisar si el DDF se encuentra completo en cuanto a los requerimientos.
14. El SGPI muestra un API con el documento al Líder de Proyectos SIS para su revisión y grabación si el DDF se encuentra completo en cuanto a los requerimientos en la base de datos.

15. El Líder de Proyectos SIS pulsa un botón para elaborar el Plan Técnico del Proyecto.
16. El SGPI muestra un API para el ingreso y grabación del Plan Técnico del Proyecto en la base de datos.
17. El Líder de Proyectos SIS pulsa un botón para elaborar el Cronograma del Proyecto.
18. El SGPI muestra un API para el ingreso y grabación del Cronograma del Proyecto en la base de datos.
19. El Líder de Proyectos SIS pulsa un botón para enviar el Plan Técnico al Gerente de Proyectos.
20. El SGPI muestra un API para el envío y grabación del Plan Técnico en la base de datos.
21. El Gerente de Proyectos pulsa un botón para revisar el Plan Técnico.
22. El SGPI muestra un API con el Plan Técnico al Gerente de Proyectos para su revisión y grabación en la base de datos.
23. El Gerente de Proyectos pulsa un botón para aprobar el Plan Técnico.
24. El SGPI muestra un API para la aprobación y grabación del Plan Técnico en la base de datos.
25. El Líder de Proyectos SIS pulsa un botón para enviar el Plan Técnico al Líder de Proyecto.
26. El SGPI muestra un API para el envío y grabación del Plan Técnico en la base de datos.
27. El Líder del Proyecto pulsa un botón para revisar el Plan Técnico con los líderes funcionales si fuese necesario.
28. El SGPI muestra un API con el Plan Técnico al Líder del Proyecto para su revisión y grabación en la base de datos.
29. El Líder del Proyecto pulsa un botón para aprobar el Plan Técnico.
30. El SGPI muestra un API para la aprobación y grabación del Plan Técnico en la base de datos.

Flujo Alternativo:

14. El SGPI muestra un API con el documento al Líder de Proyectos SIS para su revisión y grabación si el DDF se encuentra completo en cuanto a los requerimientos, en el caso de que el DDF se encuentra incompleto en cuanto a los requerimientos se graba en la base de datos el DDF con estado incompleto; para que este sea enviado al Líder del Proyecto para que realice su flujo correspondiente y cumpla con los requerimientos necesarios.
22. El SGPI muestra un API con el Plan Técnico al Gerente de Proyectos para su revisión y grabación, si el Plan Técnico está incorrecto se graba en la base de datos el Plan Técnico con estado incorrecto; para el Líder de Proyecto SIS reelabore el Plan Técnico Correcto.
28. El SGPI muestra un API con el Plan Técnico al Líder del Proyecto para su revisión y grabación, si el Plan Técnico está incorrecto se graba en la base de datos el Plan Técnico con estado incorrecto; para el Líder de Proyecto SIS reelabore el Plan Técnico Correcto.

Poscondiciones:

El pre-proyecto ha sido elaborado, aprobado y guardado en el sistema.

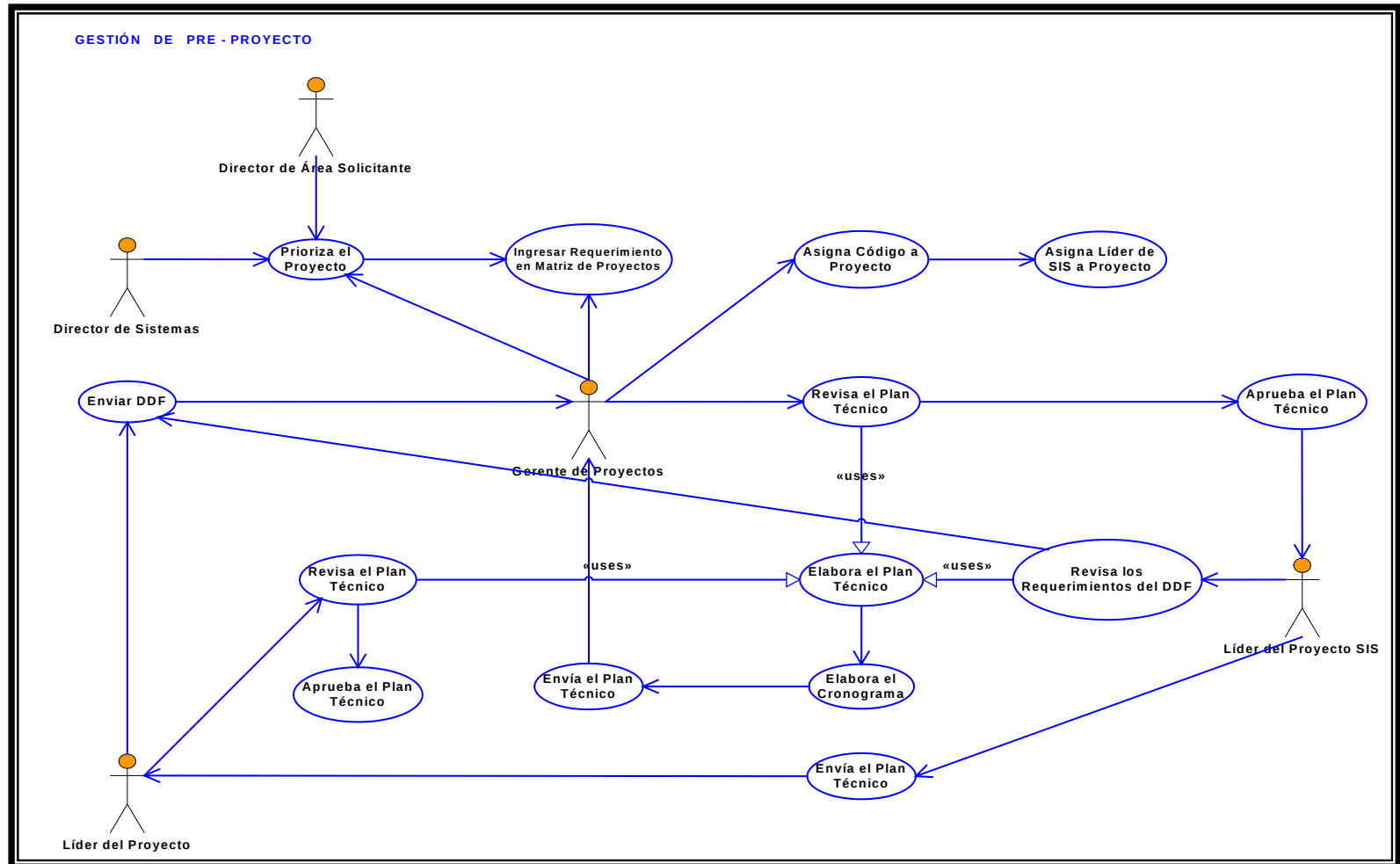


Figura 2-5
Diagrama de Caso de Uso: Gestión de Pre-Proyecto

Sistema Gestor de Proyectos Informáticos
Módulo de Integración, Parametrización y Configuraciones.

Nombre:	Gestión del Proyecto
Autor:	Grupo 6
Fecha:	14/07/2007
Descripción: Permite gestionar el desarrollo de un proyecto.	
Actores: Líder de Proyecto SIS asigna actividades del cronograma al Equipo de Desarrollo. Líder de Proyecto SIS distribuye el cronograma de trabajo al Equipo de Desarrollo. Equipo de Desarrollo elabora el DERCAS. Equipo de Desarrollo envía el DERCAS al Líder de Proyecto SIS. Líder de Proyecto SIS revisa el DERCAS. Líder de Proyecto SIS envía el DERCAS al Líder de Proyecto. Líder de Proyecto revisa el DERCAS pudiendo reunirse con Líderes y/o Colaboradores. Líder de Proyecto informa al Líder de Proyecto SIS de la aprobación. Equipo de Desarrollo elabora el documento de diseño. Líder de Proyecto SIS revisa el Diseño. Líder de Proyecto SIS informa al Equipo de Desarrollo sobre la autorización e inicio de la implementación. Equipo de Desarrollo realiza la implementación del proyecto. Equipo de Desarrollo realiza las pruebas internas de funcionalidad y stress. Equipo de Desarrollo elabora la documentación que consiste de pruebas internas, manuales de usuario y operador. Líder de Proyecto SIS revisa y corrige el documento de plan de pruebas previo a la ejecución de las mismas. Líder de Proyecto revisa y corrige el documento de plan de pruebas previo a	

la ejecución de las mismas.

Líder de Proyecto SIS envía el documento de plan de pruebas al Líder del Proyecto.

Líder de Proyecto coordina las pruebas con el Usuario Final.

Líder de Proyecto coordina la capacitación con el Usuario Final.

Líder de Proyecto SIS designa el Instructor para capacitación de los Integrantes del Equipo de Desarrollo.

Instructor procede a capacitar al Usuario Final.

Instructor recoge firmas de capacitación por parte de los asistentes a la capacitación.

Líder de Proyecto ejecuta las pruebas con el Usuario Final.

Equipo de Desarrollo ejecuta las pruebas con el Usuario Final.

Líder de Proyecto SIS recoge firmas de plan de pruebas.

Líder de Proyecto SIS envía un mail de paso a producción a la Jefatura de Proyectos y Jefatura de Producción, incluyendo los documentos de Plan de Puesta en Producción, Plan de Pruebas y Control de Objetos.

Jefe de Proyectos registra el control de versiones de objetos.

Jefe de Producción designa un responsable de realizar el pase a producción.

Jefe de Producción verifica que el pase a producción no produzca conflictos con los sistemas a su cargo.

Jefe de Producción coordina con el Líder de Proyecto SIS personal de soporte durante el pase.

Líder de Proyecto SIS designa personal para soporte de pase a producción.

Jefe de Producción realiza el pase a producción.

Líder de Proyecto SIS designa Instructor Operativo.

Jefe de Producción designa operadores que recibirán la capacitación operativa.

Jefe de Producción coordina sala de capacitación y horarios.

Instructor Operativo capacita al personal de producción.

Instructor Operativo recoge firmas de soporte de capacitación por personal que recibió la capacitación y del Jefe de Producción.

Instructor Operativo informa al Líder de Proyecto SIS la culminación de la capacitación.

Líder de Proyecto SIS determina X días para verificaciones Post-Producción.

Líder de Proyectos SIS elabora informe de verificaciones Post-Producción.

Líder de Proyectos SIS envía un mail al Gerente de Proyectos para la liberación de recursos.

Líder de Proyectos SIS recoge firma de cierre del proyecto por parte del Líder de Proyecto.

Líder de Proyecto firma el cierre del proyecto.

Líder de Proyecto SIS elabora un documento de entrega de proyecto a producción.

Líder de Proyecto SIS recoge la firma en un documento de entrega a producción por parte del DBA NET. GSI.

Precondiciones:

El Líder de Proyecto SIS debe poseer un Plan Técnico aprobado por el Líder de Proyecto para gestionar un proyecto e ingresarlo al sistema.

Flujo Normal:

1. El Líder de Proyecto SIS pulsa un botón para asignar actividades del cronograma al Equipo de Desarrollo.
2. El SGPI muestra un API con las actividades del cronograma al Líder de Proyecto SIS para su asignación y grabación en la base de datos.
3. El Líder de Proyecto SIS pulsa un botón para distribuir el cronograma de trabajo al Equipo de Desarrollo.
4. El SGPI muestra un API para la distribución y grabación del cronograma de trabajo en la base de datos.
5. El Equipo de Desarrollo pulsa un botón para elaborar el DERCAS.
6. El SGPI muestra un API para el ingreso y grabación del DERCAS en

- la base de datos.
7. El Equipo de Desarrollo pulsa un botón para enviar el DERCAS al Líder de Proyecto SIS.
 8. El SGPI muestra un API para el envío y grabación del DERCAS en la base de datos.
 9. El Líder de Proyecto SIS pulsa un botón para revisar el DERCAS.
 10. El SGPI muestra un API con el DERCAS al Líder de Proyectos SIS para su revisión y grabación en la base de datos.
 11. El Líder de Proyecto SIS pulsa un botón para enviar el DERCAS al Líder de Proyecto.
 12. El SGPI muestra un API para el envío y grabación del DERCAS en la base de datos.
 13. El Líder de Proyecto pulsa un botón para revisar el DERCAS pudiendo reunirse con Líderes y/o Colaboradores.
 14. El SGPI muestra un API con el DERCAS al Líder de Proyecto para su revisión y grabación en la base de datos.
 15. El Líder de Proyecto pulsa un botón para informar al Líder de Proyecto SIS de la aprobación del DERCAS.
 16. El SGPI muestra un API para la realización del informe y grabación de la aprobación del DERCAS en la base de datos.
 17. El Equipo de Desarrollo pulsa un botón para elaborar el documento de diseño.
 18. El SGPI muestra un API para el ingreso y grabación del documento de diseño en la base de datos.
 19. El Líder de Proyecto SIS pulsa un botón para revisar el diseño.
 20. El SGPI muestra un API con el diseño al Líder de Proyecto SIS para su revisión y grabación en la base de datos.
 21. El Líder de Proyecto SIS pulsa un botón para informar al Equipo de Desarrollo sobre la autorización e inicio de la implementación.

22. El SGPI muestra un API para la realización del informe y grabación sobre la autorización e inicio de la implementación en la base de datos.
23. El Equipo de Desarrollo pulsa un botón para realizar la implementación del proyecto.
24. El SGPI muestra un API para el ingreso y grabación de la implementación del proyecto en la base de datos.
25. El Equipo de Desarrollo pulsa un botón para realizar las pruebas internas de funcionalidad y stress.
26. El SGPI muestra un API para el ingreso y grabación de las pruebas internas de funcionalidad y stress en la base de datos.
27. El Equipo de Desarrollo pulsa un botón para elaborar la documentación que consiste de pruebas internas, manuales de usuario y operador.
28. El SGPI muestra un API para el ingreso y grabación de la documentación en la base de datos.
29. El Líder de Proyecto SIS pulsa un botón para revisar el documento de plan de pruebas previo a la ejecución de las mismas.
30. El SGPI muestra un API con el documento de plan de pruebas al Líder de Proyecto SIS para la revisión, ingreso y grabación en la base de datos.
31. El Líder de Proyecto SIS pulsa un botón para corregir el documento de plan de pruebas previo a la ejecución de las mismas.
32. El SGPI muestra un API con el documento de plan de pruebas al Líder de Proyecto SIS para la corrección, ingreso y grabación en la base de datos.
33. El Líder de Proyecto pulsa un botón para revisar el documento de plan de pruebas previo a la ejecución de las mismas.
34. El SGPI muestra un API con el documento de plan de pruebas al

Líder de Proyecto para la revisión, ingreso y grabación en la base de datos.

35. El Líder de Proyecto pulsa un botón para corregir el documento de plan de pruebas previo a la ejecución de las mismas.
36. El SGPI muestra un API con el documento de plan de pruebas al Líder de Proyecto para la corrección, ingreso y grabación en la base de datos.
37. El Líder de Proyecto SIS pulsa un botón para enviar el documento de plan de pruebas al Líder del Proyecto.
38. El SGPI muestra un API para el envío y grabación del documento de plan de pruebas en la base de datos.
39. El Líder de Proyecto pulsa un botón para coordinar las pruebas con el Usuario Final.
40. El SGPI muestra un API para la coordinación y grabación de las pruebas en la base de datos.
41. El Líder de Proyecto pulsa un botón para coordinar la capacitación con el Usuario Final.
42. El SGPI muestra un API para la coordinación y grabación de la capacitación en la base de datos.
43. El Líder de Proyecto SIS pulsa un botón para designar el Instructor para la capacitación de los Integrantes del Equipo de Desarrollo.
44. El SGPI muestra un API para la asignación y grabación del Instructor para la capacitación en la base de datos.
45. El Instructor pulsa un botón para proceder a capacitar al Usuario Final.
46. El SGPI muestra un API para el ingreso y grabación de la capacitación en la base de datos.
47. El Instructor pulsa un botón para recoger firmas de capacitación por parte de los asistentes a la capacitación.

48. El SGPI muestra un API para el ingreso y grabación de las firmas de la capacitación en la base de datos.
49. EL Líder de Proyecto pulsa un botón para ejecutar las pruebas con el Usuario Final.
50. El Equipo de Desarrollo pulsa un botón para ejecutar las pruebas con el Usuario Final.
51. El SGPI muestra un API para la ejecución y grabación de las pruebas en la base de datos.
52. El Líder de Proyecto SIS pulsa un botón para recoger firmas del plan de pruebas.
53. El SGPI muestra un API para el ingreso y grabación de las firmas del plan de pruebas en la base de datos.
54. El Líder de Proyecto SIS pulsa un botón para enviar un mail de paso a producción a la Jefatura de Proyectos y Jefatura de Producción, incluyendo los documentos de Plan de Puesta en Producción, Plan de Pruebas y Control de Objetos.
55. El SGPI muestra un API para el ingreso y grabación del mail de paso a producción en la base de datos.
56. El Jefe de Proyectos pulsa un botón para registrar el control de versiones de objetos.
57. El SGPI muestra un API para el registro y grabación del control de versiones de objetos en la base de datos.
58. El Jefe de Producción pulsa un botón para designar un responsable de realizar el pase a producción.
59. El SGPI muestra un API para el ingreso y grabación de un responsable de realizar el pase a producción en la base de datos.
60. El Jefe de Producción pulsa un botón para verificar que el pase a producción no produzca conflictos con los sistemas a su cargo.
61. El SGPI muestra un API para la verificación y grabación del pase a

- producción en la base de datos.
62. El Jefe de Producción pulsa un botón para coordinar con el Líder de Proyecto SIS el personal de soporte durante el pase.
 63. El SGPI muestra un API para la coordinación y grabación del personal de soporte en la base de datos.
 64. El Líder de Proyecto SIS pulsa un botón para designar personal para soporte de pase a producción.
 65. El SGPI muestra un API con el personal de soporte al Líder de Proyecto SIS para el ingreso y grabación del personal de pase a producción en la base de datos.
 66. EL Jefe de Producción pulsa un botón para realizar el pase a producción.
 67. El SGPI muestra un API para el ingreso y grabación del pase a producción en la base de datos.
 68. El Líder de Proyecto SIS pulsa un botón para designar el Instructor Operativo.
 69. El SGPI muestra un API para el ingreso y grabación del Instructor Operativo en la base de datos.
 70. El Jefe de Producción pulsa un botón para designar operadores que recibirán la capacitación operativa.
 71. El SGPI muestra un API para el ingreso y grabación de los operadores que recibirán la capacitación operativa en la base de datos.
 72. EL Jefe de Producción pulsa un botón para coordinar la sala de capacitación y horarios.
 73. El SGPI muestra un API para la coordinación y grabación de la sala de capacitación y horarios en la base de datos.
 74. El Instructor Operativo pulsa un botón para capacitar al personal de producción.

75. El SGPI muestra un API para el ingreso y grabación del personal de producción capacitado en la base de datos.
76. El Instructor Operativo pulsa un botón para recoger las firmas de soporte de capacitación por parte del personal que recibió la capacitación y del Jefe de Producción.
77. El SGPI muestra un API para el ingreso y grabación de las firmas de soporte de capacitación en la base de datos.
78. El Instructor Operativo pulsa un botón para informar al Líder de Proyecto SIS la culminación de la capacitación.
79. El SGPI muestra un API para el ingreso y grabación de la culminación de la capacitación en la base de datos.
80. El Líder de Proyecto SIS pulsa un botón para determinar X días para verificaciones Post-Producción.
81. El SGPI muestra un API para el ingreso y grabación de los X días para las verificaciones post-producción en la base de datos.
82. EL Líder de Proyectos SIS pulsa un botón para elaborar el informe de verificaciones Post-Producción.
83. El SGPI muestra un API para el ingreso y grabación del informe de verificaciones post-producción en la base de datos.
84. El Líder de Proyectos SIS pulsa un botón para enviar un mail al Gerente de Proyectos para la liberación de recursos.
85. El SGPI muestra un API para el envío y grabación de un mail para la liberación de recursos en la base de datos.
86. El Líder de Proyectos SIS pulsa un botón para recoger la firma de cierre del proyecto por parte del Líder de Proyecto.
87. El SGPI muestra un API para la recolección y grabación de la firma de cierre del proyecto en la base de datos.
88. El Líder de Proyecto pulsa un botón para firmar el cierre del proyecto.
89. El SGPI muestra un API para el ingreso y grabación de la firma de

cierre del proyecto en la base de datos.

90. EL Líder de Proyecto SIS pulsa un botón para elaborar un documento de entrega de proyecto a producción.
91. El SGPI muestra un API para el ingreso y grabación del documento de entrega a producción en la base de datos.
92. El Líder de Proyecto SIS pulsa un botón para recoger la firma en un documento de entrega a producción por parte del DBA NET. GSI.
93. El SGPI muestra un API para el ingreso y grabación de la firma del documento de entrega a producción en la base de datos.

Flujo Alternativo:

10. El SGPI muestra un API con el DERCAS al Líder de Proyectos SIS para su revisión y grabación en la base de datos, en el caso de que el DERCAS se encuentre incorrecto se graba en la base de datos con estado incorrecto; para que este sea enviado al Equipo de Desarrollo para que reelabore el DERCAS y cumpla con los requerimientos correctos.
14. El SGPI muestra un API con el DERCAS al Líder de Proyecto para su revisión y grabación en la base de datos, en caso de que el DERCAS se encuentre incorrecto se graba en la base de datos con estado incorrecto; para que este sea enviado al Líder de Proyecto SIS para que revise nuevamente DERCAS y cumpla con los requerimientos correctos.
20. El SGPI muestra un API con el diseño al Líder de Proyecto SIS para su revisión y grabación en la base de datos, en caso de que el Diseño se encuentre incorrecto se graba en la base de datos con estado incorrecto; para que este sea enviado al Equipo de Desarrollo para que reelabore el documento de diseño y cumpla con los requerimientos correctos.
30. El SGPI muestra un API con el documento de plan de pruebas al

Líder de Proyecto SIS para la revisión, ingreso y grabación en la base de datos, en el caso de que el documento de plan de pruebas se encuentre incorrecto se graba en la base de datos con estado incorrecto; para que este sea enviado al Líder de Proyecto SIS y Líder de Proyecto para que revisen y corrijan nuevamente el documento de plan de pruebas y cumpla con los requerimientos correctos.

51. El SGPI muestra un API para la ejecución y grabación de las pruebas en la base de datos, si las pruebas no fueron satisfactorias se graban en la base de datos con estado fallido para que se corrijan los escenarios fallidos.
61. El SGPI muestra un API para la verificación y grabación del pase a producción en la base de datos, en el caso de no necesitar soporte del equipo de desarrollo se realiza el Pase a Producción.
67. El SGPI muestra un API para el ingreso y grabación del pase a producción en la base de datos, en el caso de que no exista la necesidad de capacitación operativa se graba en la base de datos e informa al Líder de Proyecto que el desarrollo se encuentra en ambiente productivo; para determinar los X días para verificaciones de post-producción.
81. El SGPI muestra un API para el ingreso y grabación de los X días para las verificaciones post-producción en la base de datos, en el caso de que las verificaciones estén incorrectas se graba en la base de datos con estado de verificación incorrecta; para que se determine nuevamente los X días para realizar las verificaciones post-producción.

Poscondiciones:

El proyecto ha sido elaborado, aprobado, cerrado y guardado en el sistema.

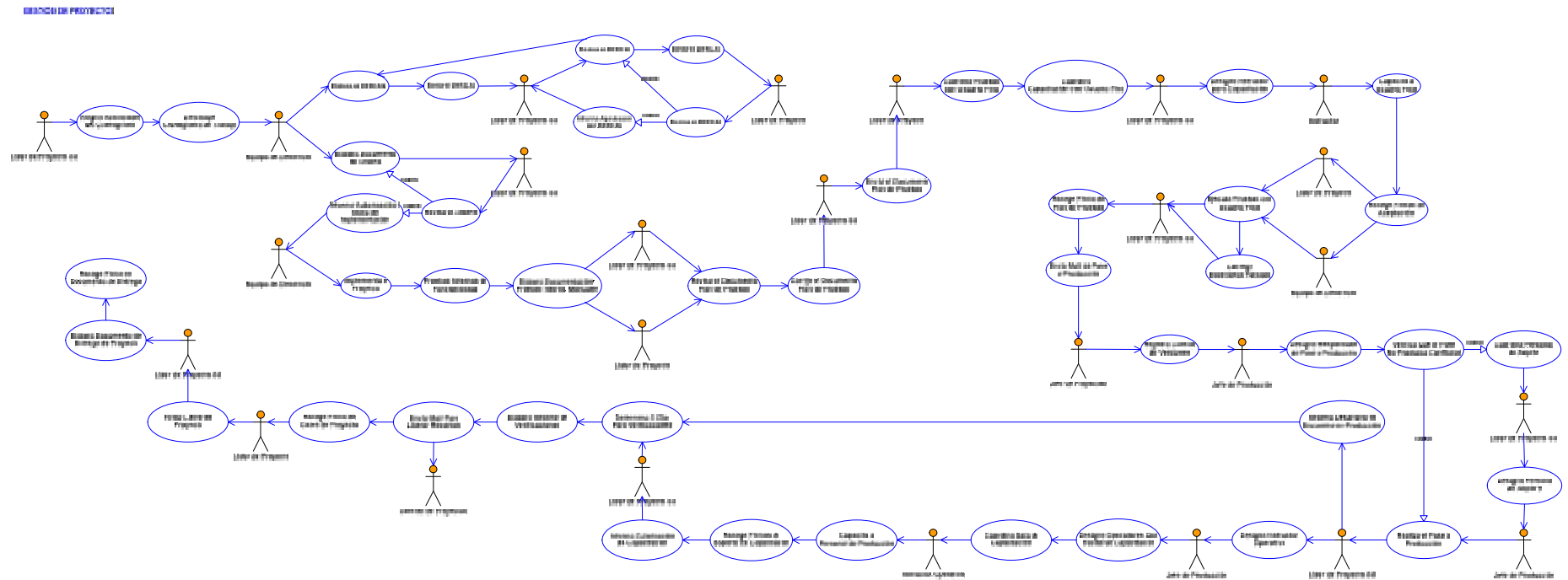


Figura 2-6
Diagrama de Caso de Uso: Gestión de Proyecto

Sistema Gestor de Proyectos Informáticos

Módulo de Integración, Parametrización y Configuraciones.

BIBLIOGRAFÍA Y REFERENCIAS WEB

- 1) **Pressman Roger** *"Ingeniería de Software" (6ta edición)* Mcgraw-Hill / Interamericana De Mexico, 2005
- 2) **Bauer Christian, Gavin King** *"Hibernate in Action"* Manning Publications; 1 edition (August 1, 2004)
- 3) www.hibernate.org .- Soporte acerca de Hibernate, ejemplos y recomendaciones.
- 4) struts.apache.org .- Soporte acerca de struts, ejemplos, configuraciones y requisitos.
- 5) eclipsetutorial.sourceforge.net .- Tutorial de Eclipse Básico
- 6) www.myeclipseide.com .- Tutorial de My Eclipse y ejemplos con Hibernate
- 7) www.gennio.com/tags/myeclipse .- Ejemplos prácticos en myeclipse
- 8) www.webestilo.com/javascript/ .- Ejemplos de javascript
- 9) www.zonaoracle.com .- Ejemplos de PLSQL, paquetes y procedimientos.

INDICE

AGRADECIMIENTO	II
DEDICATORIA	IV
TRIBUNAL DE GRADUACIÓN	V
DECLARACIÓN EXPRESA	VI
INDICE GENERAL	VII

MANUAL TÉCNICO

SISTEMA GESTOR DE PROYECTOS INFORMATICOS

	Pág.
1. Introducción.....	1
2. Instalación de Base de Datos Oracle 9i.....	2
3. Creación De Base de Datos Oracle 9i.....	11
4. Creación De Conexión a la Base de Datos	20
5. Instalación del J2SE 5.0 JDK (JAVA 2 STANDARD EDITION 5.0 JAVA DEVELOPMENT KIT).....	21
6. Instalación Del Entorno de Desarrollo Integrado Eclipse.....	25
7. Instalación Del Entorno de Desarrollo Integrado MyEclipse.....	26
8. Instalación Del Servidor de Aplicaciones Tomcat 5.5.....	33
9. Configuración Del Entorno de Desarrollo Integrado Eclipse.....	39
10. Configuración Del Entorno de Desarrollo Integrado Eclipse Con Jasper.....	42

11. Configuración Del Entorno de Desarrollo Integrado Eclipse Con Display (Para Mostrar Tablas En Una Página).....	43
12. Configuración Del Entorno de Desarrollo Integrado Eclipse Con Librerías	43
13. Importar, Deployar y Levantar Proyectos.....	45

MANUAL DE USUARIO

SISTEMA GESTOR DE PROYECTOS INFORMATICOS

Introducción.	Pág.
14. Mantenimiento de Personas.....	1
15. Mantenimiento de Cargos.....	9
16. Mantenimiento de Áreas.....	12
17. Mantenimiento de Provincias.....	14
18. Mantenimiento de Estados.....	16

MANUAL TÉCNICO

SISTEMA GESTOR DE PROYECTOS INFORMÁTICOS

ÍNDICE

MANUAL DE INSTALACIÓN	3
INTRODUCCION	3
INSTALACIÓN DEL SERVIDOR DE APLICACIONES	3
COPIAR DRIVERS DE CONEXIÓN	9
COPIAR POLÍTICAS DEL SERVIDOR.....	9
INICIAR Y DETENER EL SERVIDOR DE APLICACIONES	10
<i>INICIAR EL SERVIDOR DE APLICACIONES</i>	<i>10</i>
<i>DETENER EL SERVIDOR DE APLICACIONES</i>	<i>11</i>
ADMINISTRACIÓN DEL SERVIDOR DE APLICACIONES.....	12
<i>INGRESO</i>	<i>12</i>
<i>CREAR EL PROVEEDOR DE DATOS.....</i>	<i>13</i>
<i>PROBAR EL PROVEEDOR DE DATOS.....</i>	<i>17</i>
CREACIÓN DE LA FUENTE DE DATOS	19
IMPLEMENTAR LA APLICACIÓN ACollect	21
COPIAR REPORTES Y LOGO	24

MANUAL DE INSTALACIÓN

INTRODUCCION

Este manual esta dirigido al administrador del sistema y contiene información para instalar el servidor de aplicaciones, crear la conexión a la base de datos e implementar el sistema.

Para la instalación de AClient, el administrador del sistema deberá poseer el CD de instalación entregado.

INSTALACIÓN DEL SERVIDOR DE APLICACIONES

A continuación se detallan los pasos a seguir para la instalación del servidor de aplicaciones, **Sun Java™ System Application Server Platform Edition 8.1 2005Q1**:

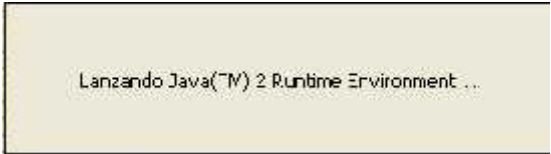
Paso 1

Ejecución del archivo j2eesdk-1_4_02_2005Q2-windows.exe que se encuentra en el CD de instalación en el directorio DRIVE_CD:\Servidor de Aplicaciones, una vez ejecutado se comienza a extraer los archivos de instalación y se presenta la esta pantalla.

Extraendo archivos de instalación...

Paso 2

Lanzando el Java RunTime Enviroment.

Lanzando Java(TM) 2 Runtime Environment ...

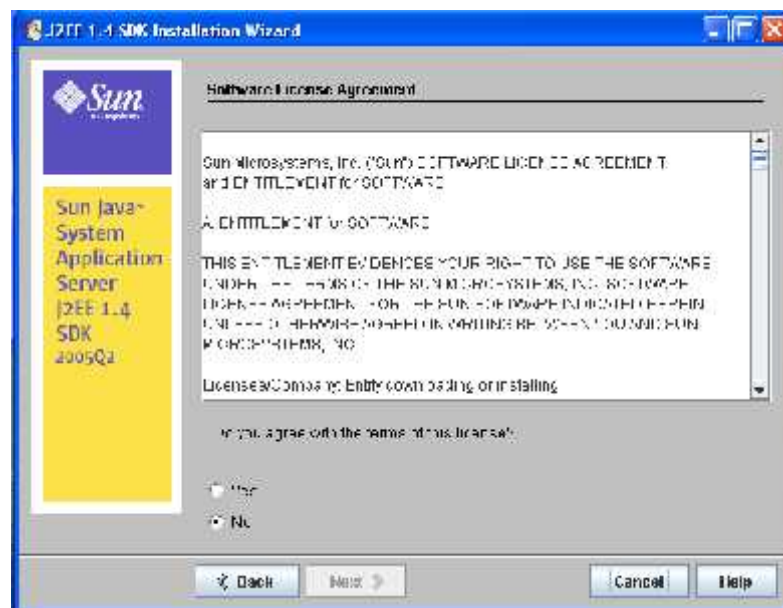
Paso 3

Bienvenida al servidor de aplicaciones, presione el botón NEXT.



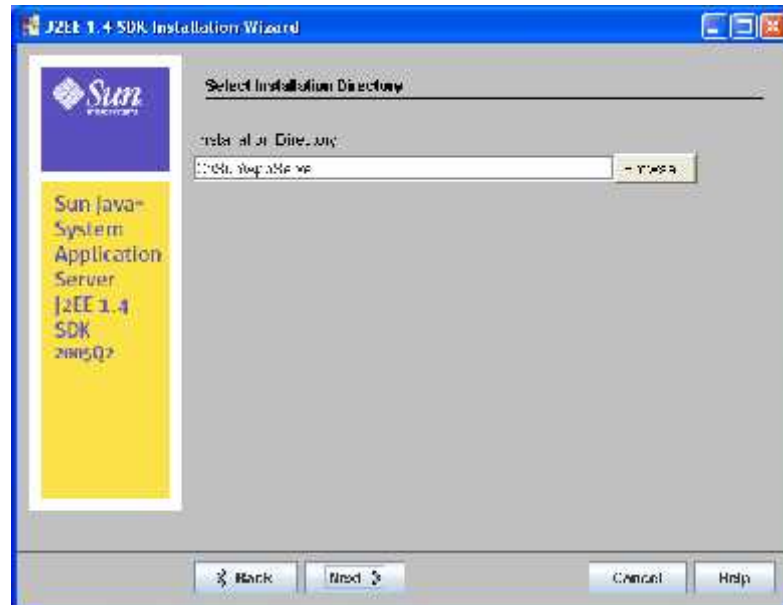
Paso 4

Aceptación de la licencia, seleccione la opción YES. Una vez seleccionada la opción YES se habilita el botón NEXT y presiónelo.

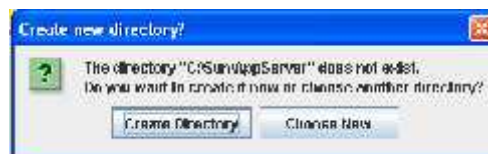


Paso 5

Selección del directorio de instalación. Se recomienda dejar el directorio por defecto, presione el botón NEXT.

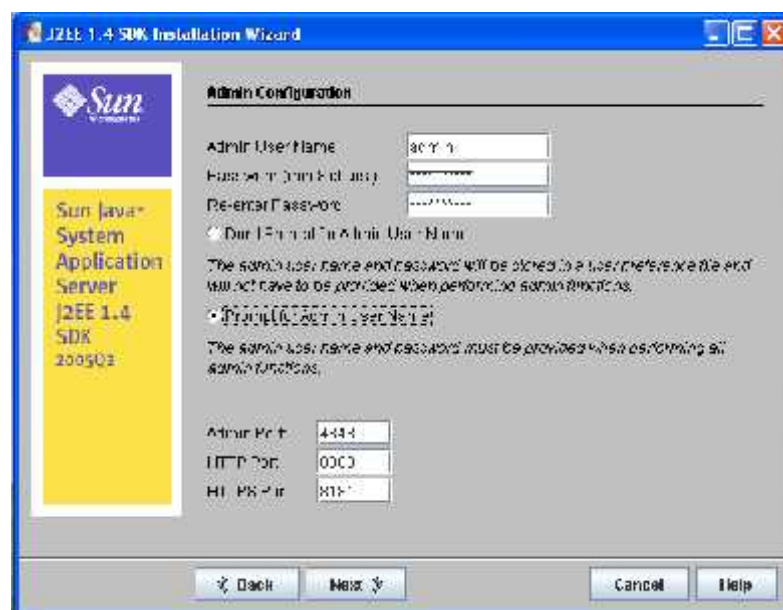


Si el directorio no existe de click en el botón Create Directory.



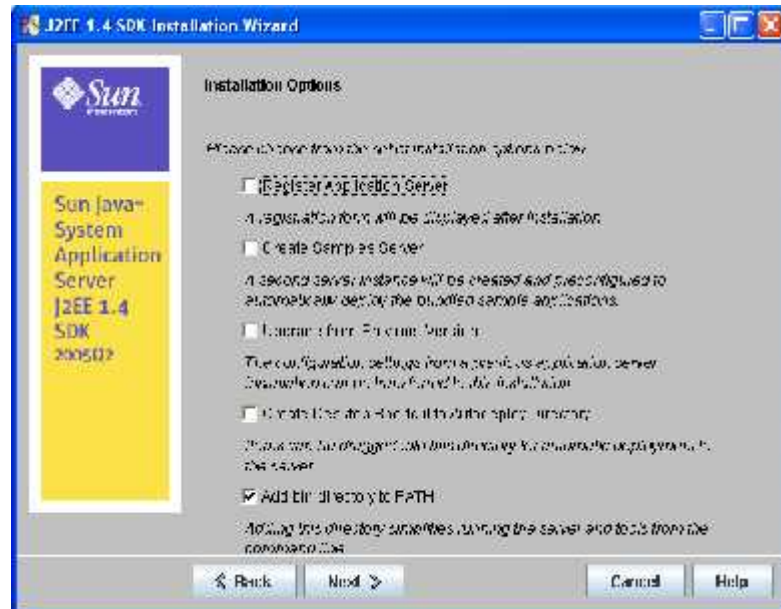
Paso 7

- Crear un password para el servidor de aplicaciones
- Elegir la opción Prompt for Admin User Name para ingresar siempre el password
- Asignación de puertos (se recomienda los puertos por defecto).
- Presione el botón NEXT.



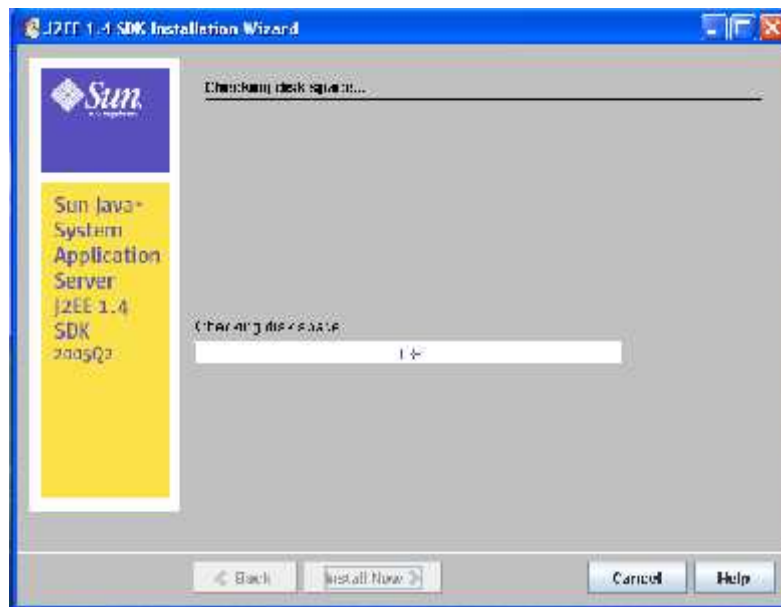
Paso 8

Solo marcar la opción Add bin directory to PATH, presionar el botón NEXT



Paso 9

Chequeo del espacio en disco.



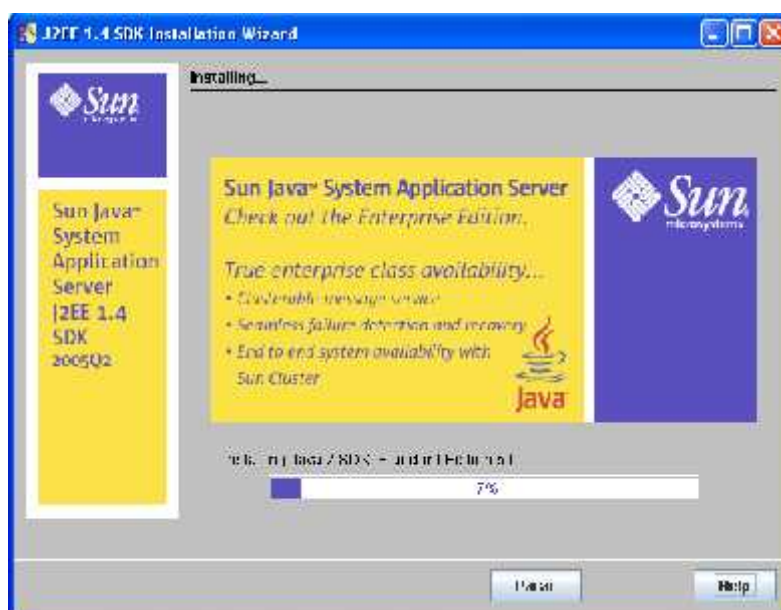
Paso 10

Comienzo de la instalación, darle click al botón Install Now:



Paso 11

Proceso de instalación.



Paso 12

Finalización de la instalación del servidor de aplicaciones, presione el botón FINISH



Paso 13

Pantalla de agradecimiento.



Aquí finaliza la instalación del servidor de Aplicaciones.

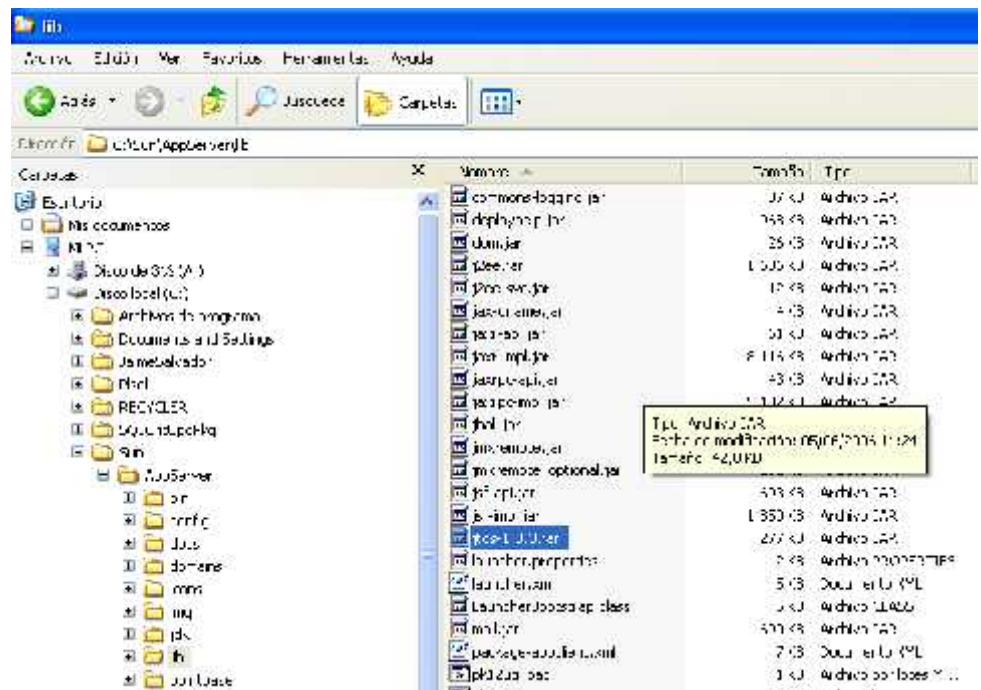
COPIAR DRIVERS DE CONEXIÓN

Una vez instalado el Servidor de Aplicaciones procedemos a copiar los drivers de conexión a la base de datos, para que la aplicación pueda conectarse a la fuente de datos.

Utilizaremos la variable `INSTALL_DIR_APPSERVER`, que representa el directorio de instalación del servidor de aplicaciones. Por ejemplo:

```
INSTALL_DIR_APPSERVER = C:\Sun\AppServer\
```

Copiar los archivos de conexión de base de datos `jtds-1.0.3.jar` y `driveradapter.jar` que se encuentra en el CD de instalación en el directorio `DRIVE_CD:\JAR a INSTALL_DIR_APPSERVER\lib`.



COPIAR POLÍTICAS DEL SERVIDOR

Debemos reemplazar el archivo de políticas del servidor de aplicación, `server.policy`, para ello copiamos el archivo ubicado en el CD de instalación:

DRIVE CD:\Servidor de Aplicaciones\server.policy

Lo copiamos dentro del directorio del Servidor de Aplicaciones:

```
INSTALL DIR APPSERVER\domains\domain1\config\
```

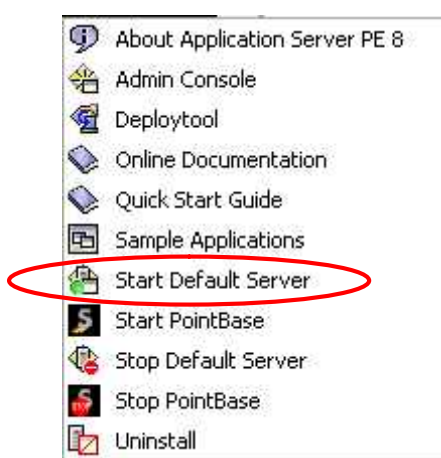
INICIAR Y DETENER EL SERVIDOR DE APLICACIONES

En esta sección aprenderemos a iniciar y detener el servidor de aplicaciones.

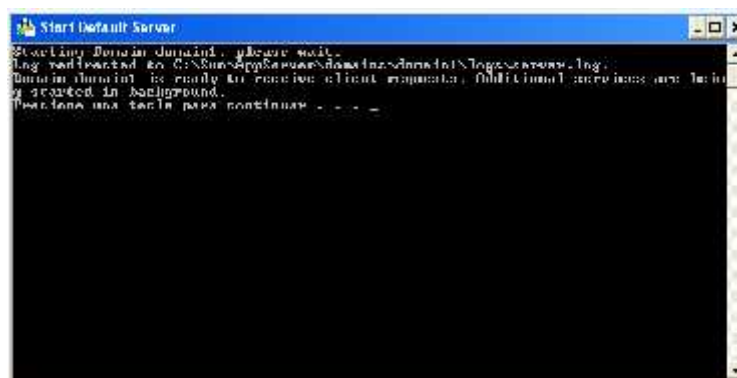
INICIAR EL SERVIDOR DE APLICACIONES

Iniciar el Servidor de Aplicaciones nos permite habilitar el acceso al mismo, tanto a la administración como el acceso a los usuarios del sistema.

Para Iniciar el servicio se debe ingresar al menú Sun Microsystems y elegir la opción Start Default Server.



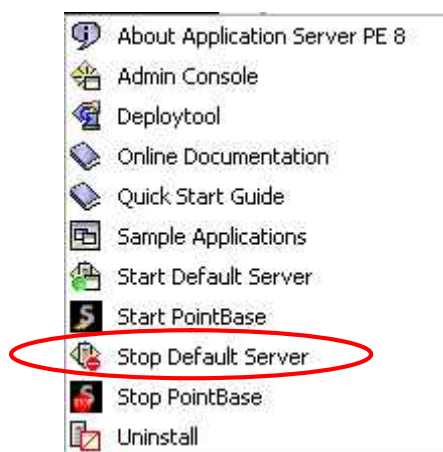
Después de elegir la opción se presenta una consola de sistema operativo que indica que el servidor se levanta correctamente.



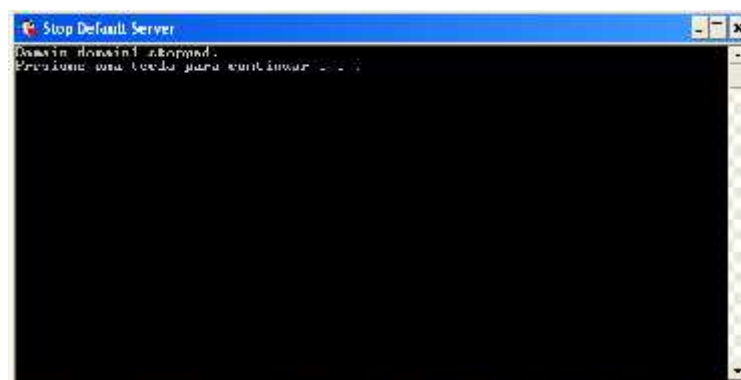
DETENER EL SERVIDOR DE APLICACIONES

Detener el Servidor de Aplicaciones nos permite deshabilitar el acceso al mismo, tanto a la administración como el acceso a los usuarios del sistema.

Para detener el servicio se debe ingresar al menú Sun Microsystems y elegir la opción Stop Default Server.



Después de elegir la opción aparece una consola de sistema operativo que indica que el servidor se ha detenido.



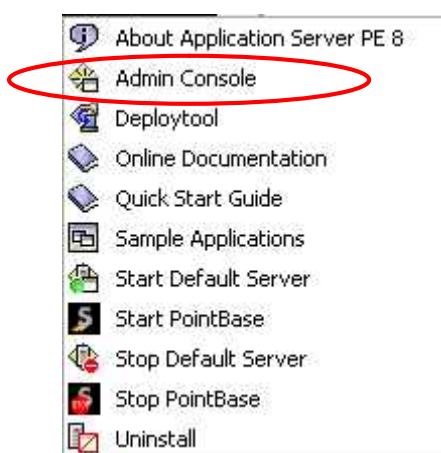
ADMINISTRACIÓN DEL SERVIDOR DE APLICACIONES

En la consola de administración del servidor de aplicaciones se configuran los diferentes servicios que tiene el SUN APPLICATION SERVER.

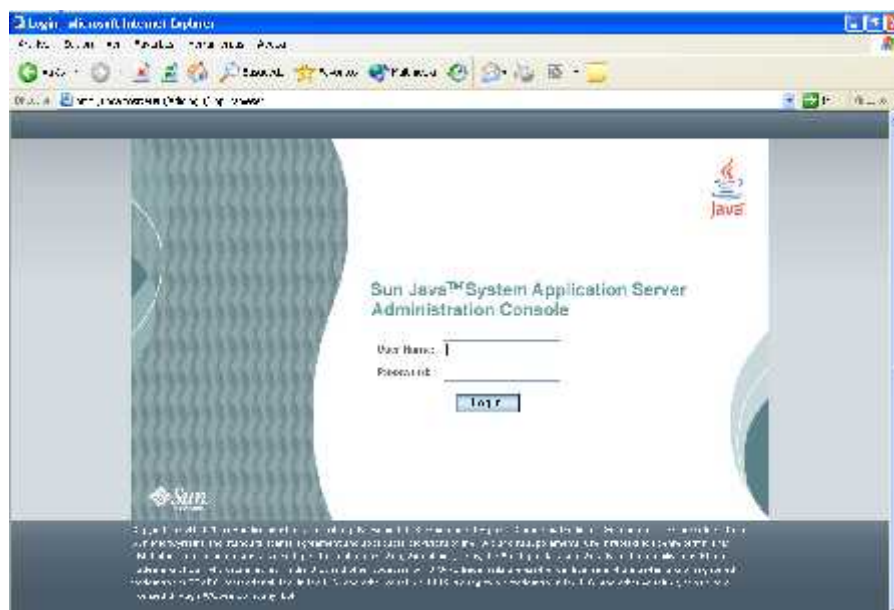
INGRESO

Iniciar el servidor de aplicaciones, ver sección Iniciar y Detener el Servidor de Aplicaciones

Luego de iniciado el Servidor de Aplicaciones podemos acceder a la consola de administración, para ello se elige la opción Admin Console del menú Sun Microsystems.



Inmediatamente se iniciará el Internet Explorer o el navegador de internet configurado como predeterminado y podremos ingresar el usuario y la clave ingresada durante la instalación.



Si ingresamos correctamente el usuario y clave nos dirigimos a la pantalla principal de administración.



CREAR EL PROVEEDOR DE DATOS

El proveedor de datos o mejor conocido como Connection Pool, es simplemente la utilización del driver de conexión para la Base de Datos a la que nos estamos conectando.

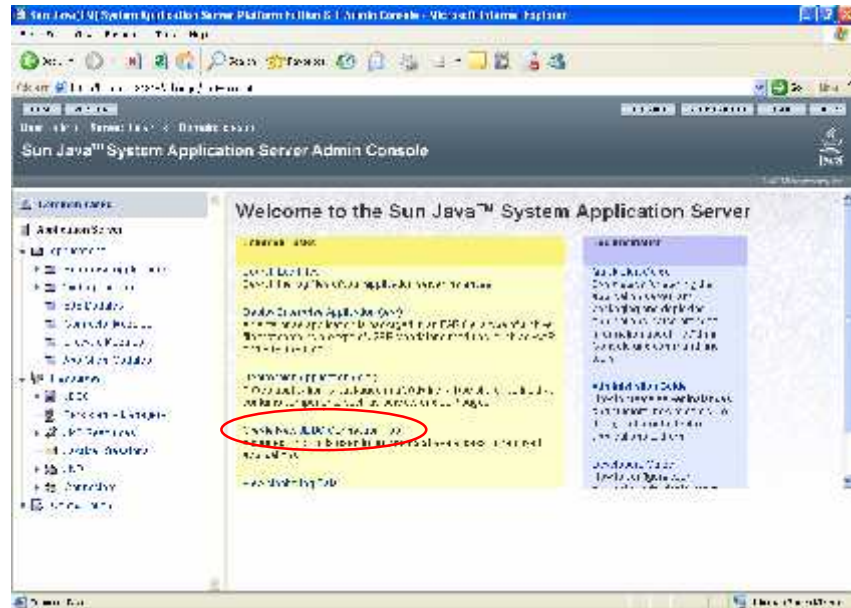
El Connection Pool almacena, organiza y obtiene los datos de una base de datos relacional mediante la utilización de JDBC API.

Seguiremos los siguientes pasos para su creación:

Paso 1

Crear el Connection Pool, el usuario tiene dos opciones a seguir:

- **Create New JDBC Connection Pool**, desde la pantalla de bienvenida al servidor de aplicaciones



- Elegir desde el árbol de tareas y dando click en la opción JDBC -> **Connection Pool**. Se presenta la pantalla de todas los **Connection Pool** creados. En esta pantalla de click en el botón **New...**.



Paso 2

Ingreso de la Configuración General.

- Name: Se escoge un nombre adecuado para el **Connection Pool**, ingresamos **ACollectPool**.
- Resource Type: Se escoge un tipo de recurso, elegimos **javax.sql.DataSource**.
- Database Vendor: No se escoge.
-

Create Connection Pool

Identify the general settings for the connection pool.

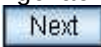
Next Cancel

* Indicates required field

General Settings

* Name: AClientTool
Resource Type: javax.sql.DataSource
Database Vendor:

Next Cancel

- Para finalizar la configuración general damos click en el botón "Next" 

Paso 3

Ingresamos el Datasource Classname, com.sun.sql.datasource.DriverAdapter y continuamos dando click en el botón "Next" 

Create Connection Pool

Review the connection pool general settings before proceeding.

Previous Next Cancel

* Indicates required field

General Settings

Name: AClientTool
Resource Type: javax.sql.DataSource
Database Vendor:
* Datasource Classname: com.sun.sql.datasource.DriverAdapter
Vendor-specific classname that implements the DataSource and/or XADataSource APIs

Previous Next Cancel

Paso 4

La siguiente pantalla nos presentará 4 secciones de configuración, a saber:

- Sección Pool Settings, dejamos los valores por defecto.

Pool Settings

Initial and Minimum Pool Size: connections
Minimum and initial number of connections instantiated in the pool (default is 8)

Maximum Pool Size: connections
Maximum number of connections that can be created to satisfy client requests (default is 25)

Pool Resize Quantity: connections
Number of connections to be removed when pool throughput drops

Min Lifetime: seconds
Minimum time that a connection can remain in the pool (default is 300)

Max Wait Time: milliseconds
Amount of time caller waits before connection timeout is sent

2. Sección Connection Validation, hacemos los siguientes cambios:

Connection Validation

Connection Validation: ☒ Required
Verify connections; otherwise, always attempt to reconnect in case of failure

Validation Method:

Table Name:
If table validation selected, specify table name

On Any Failure: ☒ Close All Connections
Close all connections and reconnect on failure; otherwise, reconnect only when used

3. Sección Transaction Isolation, dejamos los valores por defecto

Transaction Isolation

Transaction Isolation:

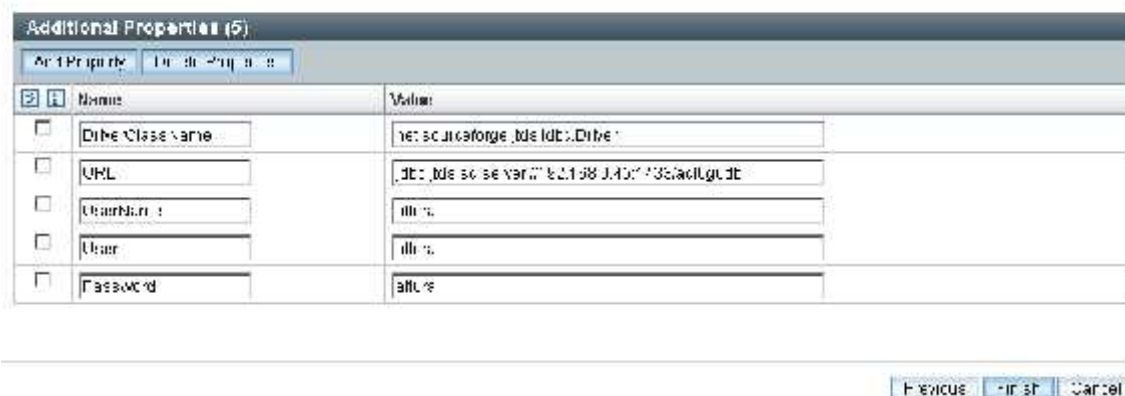
If unspecified, use default level for JDBC Driver

Isolation Level: ☐ Constant
All connections use same isolation level; requires Transaction Isolation

4. Sección Properties, agregamos y eliminamos opciones con los botones Add Property y Delete Properties, hasta dejar configuradas las siguientes opciones:

DriverClassName:
net.sourceforge.jtds.jdbc.Driver
URL:
jdbc:jtds:sqlserver://HOSTNAME:1433/DATABASE
UserName:
Nombre del usuario de la base de datos
User:
Nombre del usuario de la base de datos
Password:
Clave de acceso del usuario de la base de datos

Properties



Name	Value
Driver Class Name	net.sourceforge.jtds.Driver
URL	jdbc:sqlserver://201.98.1.101:1433;database=...
CharacterSet	utf-8
User	sa
Password	sa!@#

Nota:

La variable HOSTNAME significa el nombre del equipo o la dirección IP del servidor de la base de datos.

La variable DATABASE es el nombre de la base de datos o Esquema de datos a la que se quiere acceder.

Paso 5

Finalmente damos click sobre el botón "Finish" para finalizar con la configuración del Connection Pool.

PROBAR EL PROVEEDOR DE DATOS

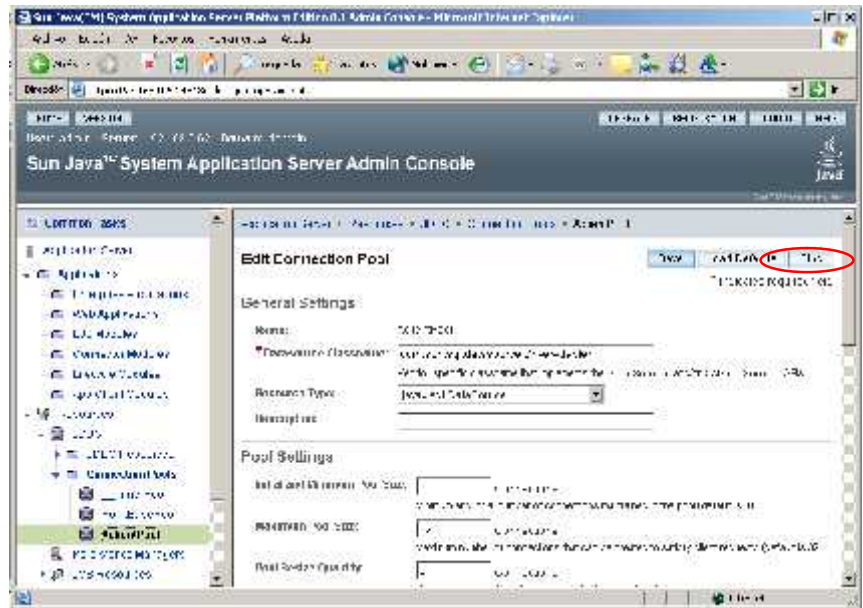
El siguiente paso en la configuración del Servidor de Aplicaciones es probar la conexión, configurada durante la creación del proveedor de datos, para lo cual seguimos los siguientes pasos:

Paso 1

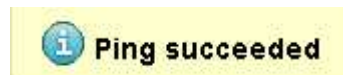
Como en el primer paso de la sección anterior, nos dirigimos a la pantalla Current Pool, donde damos click en el Connection Pool que acabamos de crear ACollect Pool.

Paso 2

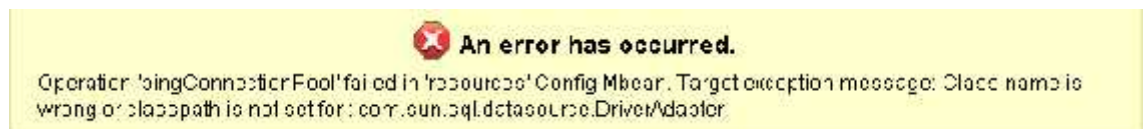
A continuación se abrirá la página con la configuración de nuestro proveedor de datos, hacemos click en el botón "Ping"  para la prueba de conexión.



Si el **Connection Pool** esta configurado correctamente se presentara el siguiente mensaje en la pantalla de edición del **Connection Pool**.



Caso contrario se presentara un mensaje de error indicando la descripción del mismo que se ha encontrado en la configuración del **Connection Pool**, lo cual debemos verificar si hemos ingresado correctamente la configuración y si tenemos conectividad con la Base de Datos.



CREACIÓN DE LA FUENTE DE DATOS

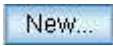
Una vez creado el proveedor de datos, tenemos que enlazar la fuente de datos al sistema, para ello creamos un JDBC Resource, seguimos los siguientes pasos:

Paso 1

En el árbol de tareas de la consola de administración del servidor de aplicaciones damos click a la opción JDBC Resources, se nos presenta la pantalla de Current Resources.



Paso 2

Damos click en el botón “New”  para crear un nuevo JDBC Resource.

Paso 3

Aparece la pantalla donde ingresaremos los siguientes parámetros:

- JNDI Name, el valor que se le da a este campo no se puede cambiar y debe ser jdbc/AClientDB
- Pool Name, se escoge el Connection Pool creado anteriormente “ACollectPool”
- Description, no es necesario ingresar un valor
- Status, dejamos el valor por defecto

Create JDBC Resource

OK Cancel

When you create a JDBC resource, you specify a unique JNDI name that identifies the resource.

* Indicates required field

* JNDI Name:

* Pool Name: Use the Connection Pools page to create new pools.

Description:

Status: ☒ Enabled

Paso 3

Damos click en el botón "OK"  para guardar la configuración y verificamos en la pantalla de Current Resources si se creo nuestro Fuente de Datos.

JDBC Resources

JDBC resources provide applications with a means of connecting to a database.

Current Resources (3)			
New... Delete			
☒	JNDI Name	Enabled	Description
☐	jdbc_TimorPool	true	
☒	jdbcAClientDB	true	
☐	jdbcPostBase	true	

IMPLEMENTAR LA APLICACIÓN ACOLLECT

Implementar la Aplicación ACollect, significa desempaquetar los archivos que forman el sistema Web y alojarlos en el servidor de aplicaciones, para lo cual seguimos los siguientes pasos:

Paso 1

En la consola de administración del servidor de aplicaciones ubicar en el árbol de tareas la opción Web Applications, donde se presentará la pantalla para implementar ACollect y mostrará las aplicaciones que ya han sido instaladas.



Paso 2

Damos click en el botón “Deploy”  y se muestra la pantalla Deploy Web Module.

Deploy Web Module

Specify the location of an application to deploy. Applications can be in packaged files, such as .war or in the standard Web Module directory format.

Location: ☒ Specify a package file to upload to the Application Server

File To Upload: 

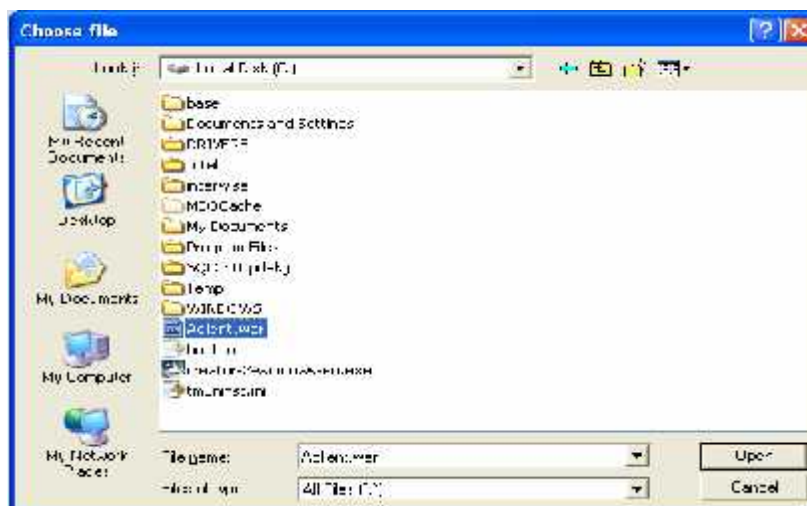
☐ Specify a package file in a directory path that must be accessible from the server

File Or Directory:

Paso 3

Dando click en el botón “Brose...” aparecerá la pantalla donde ubicaremos el archivo a implementar:

DRIVE_CD:\WAR\ACollect.war



Una vez que localizamos el archivo damos click en el botón “Open”.

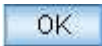
Paso 4

Damos click en el botón “Next” , esperamos unos segundos hasta que el archivo sea subido al servidor de aplicaciones. A continuación se presenta la pantalla de Configuración General para poder implantar ACollect.

Paso 5

Dentro de la pantalla General realizamos los siguientes cambios en los campos:

- Application Name: ACollect
- Context Root: /ACollect

Una vez hechos los dos cambios hacemos click en el botón “OK”  para comenzar a procesar el archivo ACollect.war.



Mientras procesa el archivo aparecerá el botón **Processing...** deshabilitado, al momento de completarse el proceso de implementación aparecerá nuevamente la pantalla de Web Application.

Web Applications

A Web application module consists of a collection of Web resources such as JavaServer Pages (JSPs), servlets, and HTML pages that are packaged in a WAR (Web Application Archive) file or directory.

Deployed Web Applications (1)			
Deploy Undeploy Enable Disable			
☐ Application Name	Enabled	Context Root	
☒ AClient	TRUE	/AClient	Launch

Paso 8

Salimos del Servidor de Aplicaciones haciendo click en el botón **"LOGOUT"** que se encuentra en la parte superior derecha del navegador.



Volveremos a la página de inicio del Servidor de Aplicaciones y cerramos el navegador.



Paso 9

Debemos reemplazar el archivo de configuración de la aplicación web sun-web.xml, para ello copiamos el archivo ubicado en el CD de instalación:

`DRIVE_CD:\WAR\sun-web.xml`

Lo copiamos dentro del directorio del Servidor de Aplicaciones:

`INSTALL_DIR_APPSERVER\domains\domain1\applications\j2ee-modules\ACollect\WEB-INF`

COPIAR REPORTES Y LOGO

Como último paso en la configuración del Servidor de Aplicaciones es copiar los reportes y el logo del Municipio de Portoviejo, para ello debemos copiar la carpeta reports que se encuentra en el CD de instalación:

`DRIVE_CD:\reports\*.*`

Lo copiamos dentro del directorio del Servidor de Aplicaciones:

`C:\reports\`



S O L U C I O N E S D E N E G O C I O S

Dirección: Kennedy Norte, Av. Orellana, Edf. Torre Atlas, Piso 7, Oficina 4

Telef.: +593 2 240 1322
+593 4 269 0416

Celular: +593 9 992 0072
+593 9 402 7781

E-mail: info@altura.com.ec

Quito - Ecuador
Guayaquil - Ecuador

MANUAL TÉCNICO

1 INTRODUCCIÓN

Este manual está dirigido al administrador del sistema “Sistema Gestor de Proyectos Informáticos (SGPI)”.

En este manual se usa la terminología técnica de manera libre ya que el manual esta dirigido a administradores conocedores de informática.

Contiene la información necesaria para instalar un Servidor de Base de Datos Oracle 9i en Sistema Operativo Windows XP, instalar y configurar un entorno de desarrollo integrado Eclipse, MyEclipse, servidor de aplicaciones Tomcat, crear la conexión a la base de datos Oracle e implementar el sistema SGPI.

Aunque hemos trabajado en el Servidor de Oracle y Servidor de Aplicaciones con Windows 2003 Server la instalación no varía.

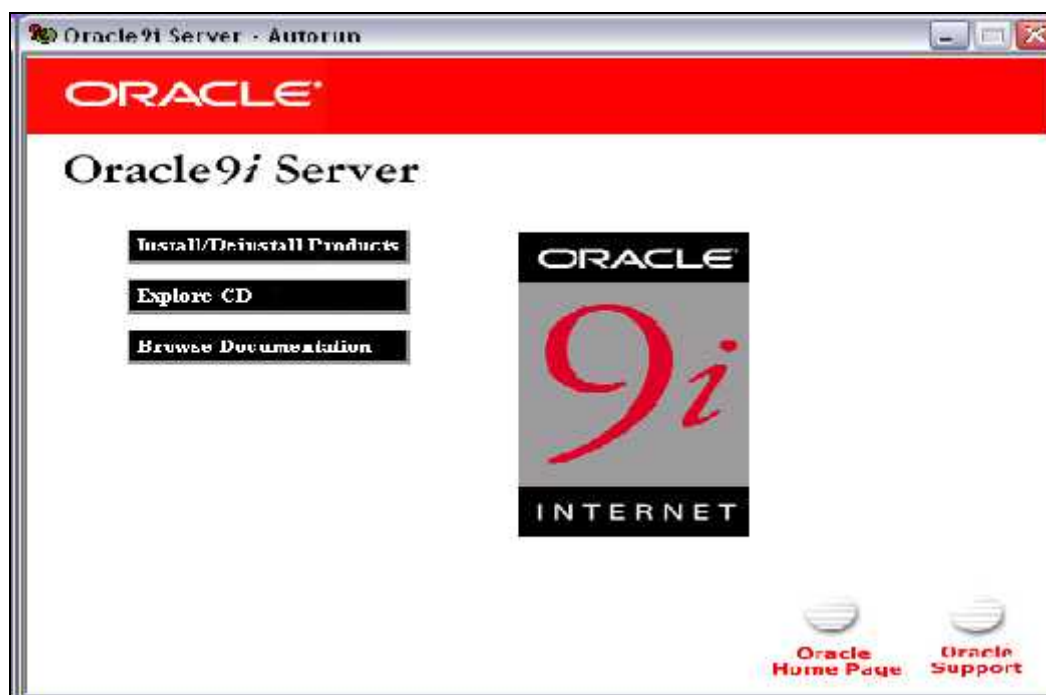
Para la instalación del SGPI el administrador del sistema deberá poseer el CD del manual técnico entregado junto al instalador de SGPI.

2. Instalación De Base De Datos Oracle 9i

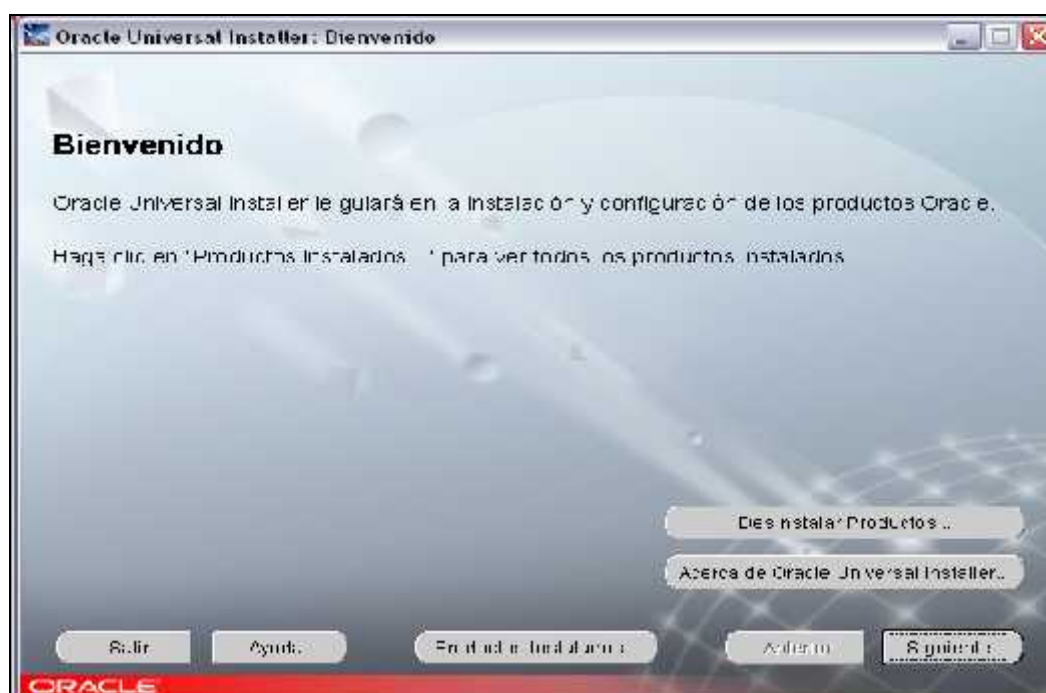
Para proceder a la instalación del servidor de base de datos se requieren de tres CDs que componen el programa de instalación de oracle, los cuales se los puede descargar gratuitamente (siempre que no sea con fines lucrativos) desde la propia página Web de oracle (www.oracle.com).

A continuación se detallan los pasos a seguir para la instalación del servidor de base de datos oracle 9i en sistemas operativo Windows XP.

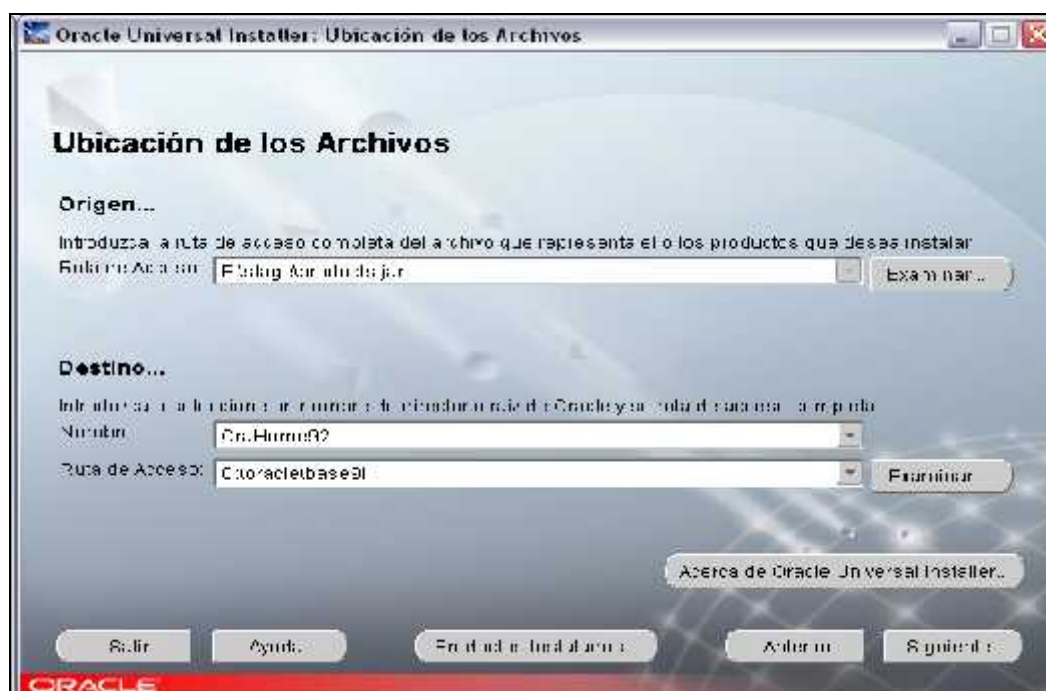
1. Introducimos el CD 1 de la instalación. Se realiza la ejecución del archivo autorun.exe que se encuentra en el CD de instalación en el directorio DRIVE_CD:\autorun\autorun.exe, una vez ejecutado se comienza a extraer los archivos de instalación y se presenta la siguiente pantalla.

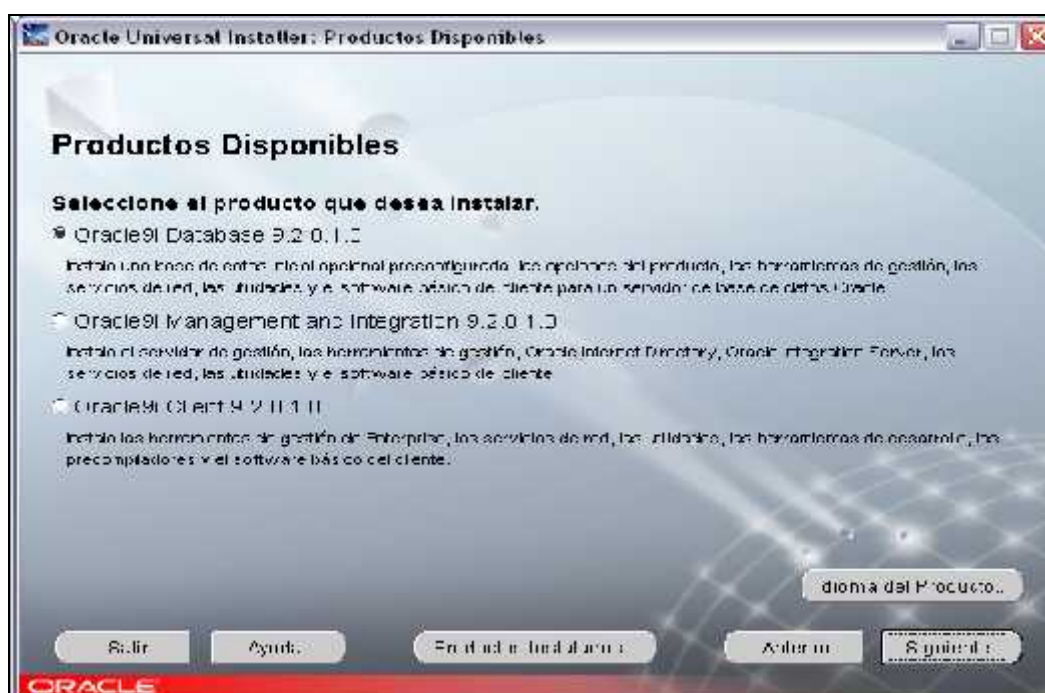


2. Presionamos el botón que indica Install/Deinstall Products, el cual con dicha acción presenta la siguiente pantalla.



3. Bienvenido a la Instalación Universal de Oracle, en la cual si desea ver los productos de oracle instalados en el servidor presione el botón que indica Productos Instalados.... Luego presione el botón Siguiente, el cual nos presenta la siguiente pantalla.





7. Seleccionaremos el tipo de instalación que deseamos, en este caso seleccionaremos el tipo por defecto Enterprise Edition. Si queremos realizar una instalación avanzada (especificando manualmente las opciones a instalar) seleccionamos Personalizado. Luego presione el botón Siguiente.



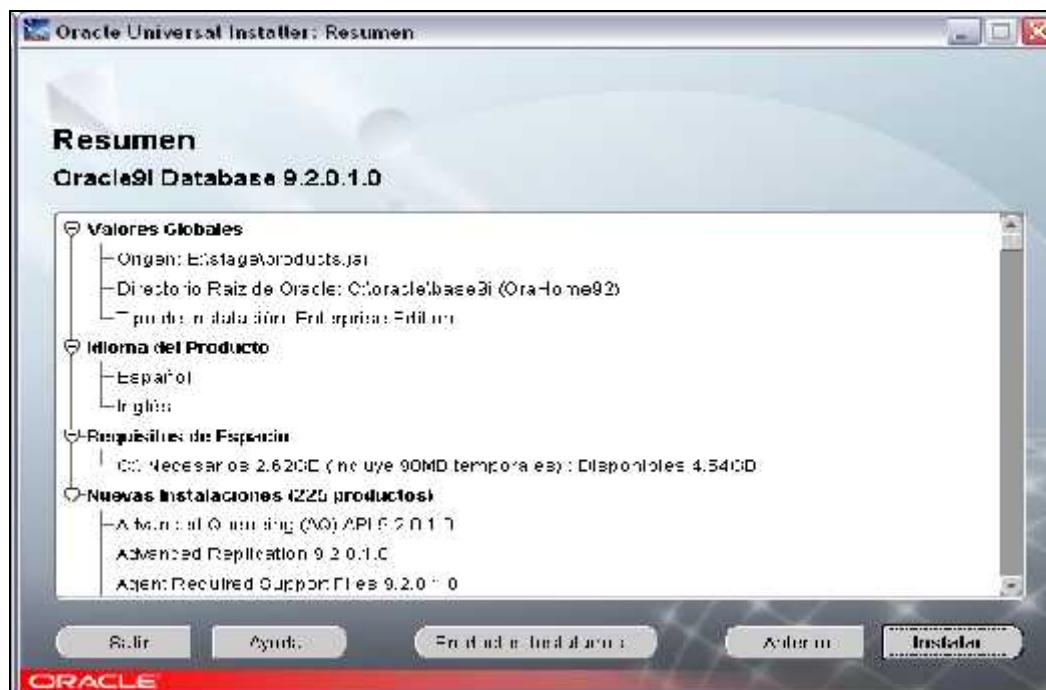
8. Seleccionaremos la configuración de la base de datos a instalar dependiendo del uso que le queramos dar a nuestra base de datos, normalmente es para Uso General. Si no queremos crear una base de datos en el proceso de instalación (se puede crear en otro momento) seleccionaremos Solo Software. Luego presione el botón Siguiente.



9. Seleccionaremos el puerto para el Servicio de Oracle MTS (Microsoft Transaction Server) normalmente se suele seleccionar el puerto por defecto 2030. Este parámetro es muy importante pues, si deseamos cambiar el puerto por defecto, cuando queramos que un cliente se conecte al servidor deberemos especificar el puerto que ya hemos especificado en este punto de la instalación. Luego presione el botón Siguiente.



10. Como último paso de la preinstalación nos aparecerá una ventana con el software que se va a instalar, luego de comprobar que es correcto presionaremos el botón Instalar.



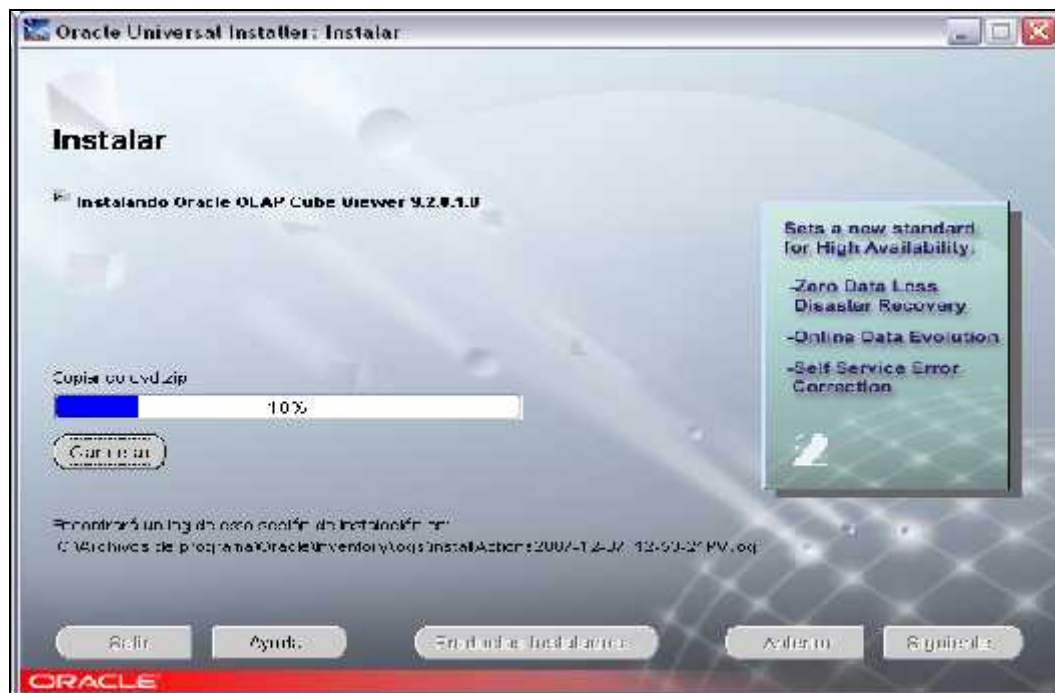
11. Comienza la instalación de oracle.



12. Introducimos el CD 2 de instalación. Luego presione el botón Aceptar.



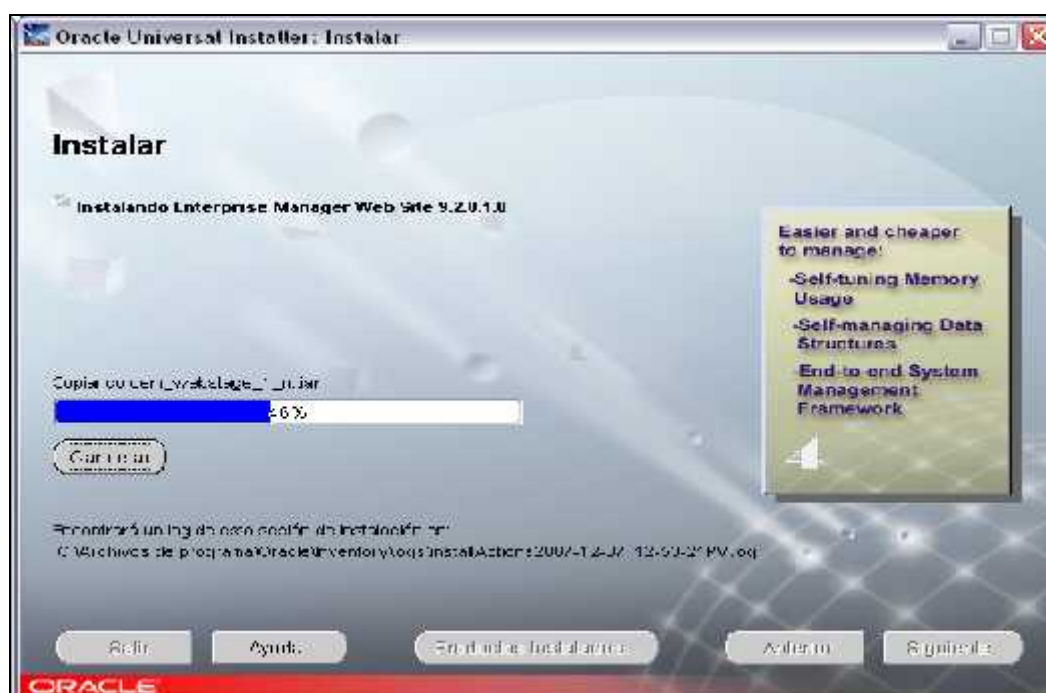
13. Continúa la instalación de oracle.



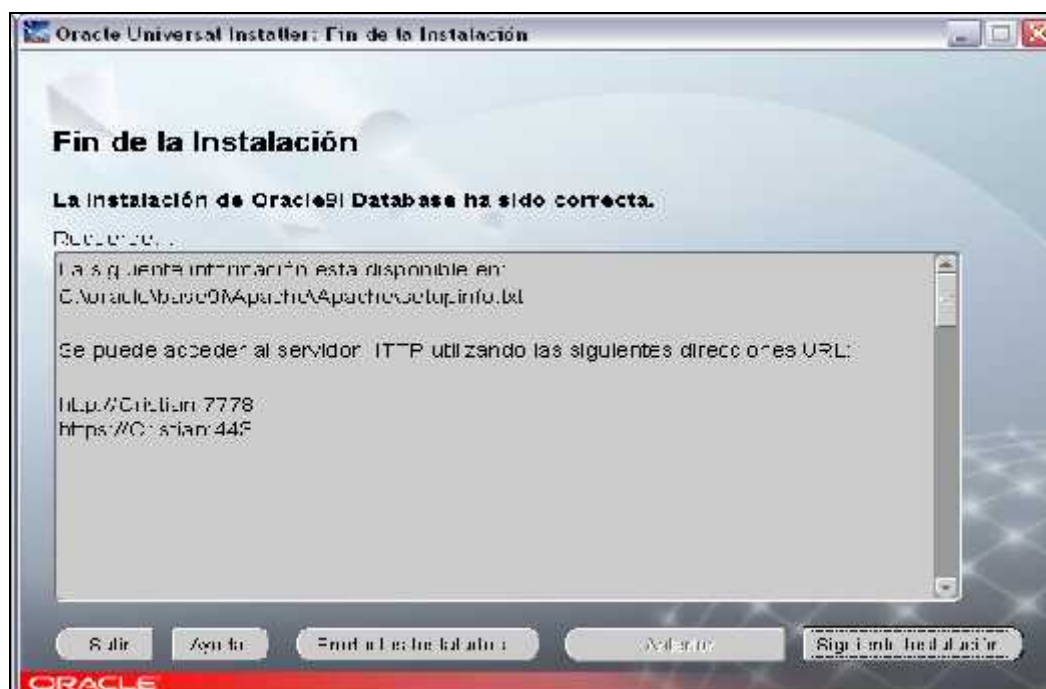
14. Introducimos el CD 3 de instalación. Luego presione el botón Aceptar.



15. Continúa la instalación de oracle.



16. Luego de la instalación de Oracle aparecerá una ventana indicando que el proceso de instalación ha finalizado. Luego presione el botón Salir.



17. Confirmamos que deseamos salir de la aplicación y presionamos el botón Sí. Con esto finaliza la instalación del servidor de base de datos oracle.

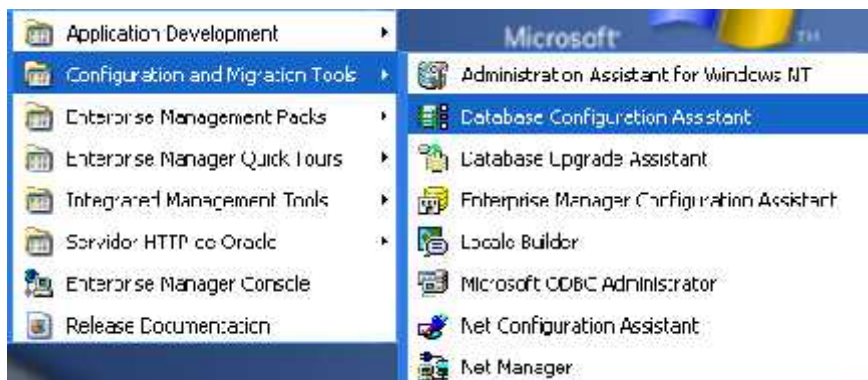


3. Creación De Base De Datos Oracle 9i

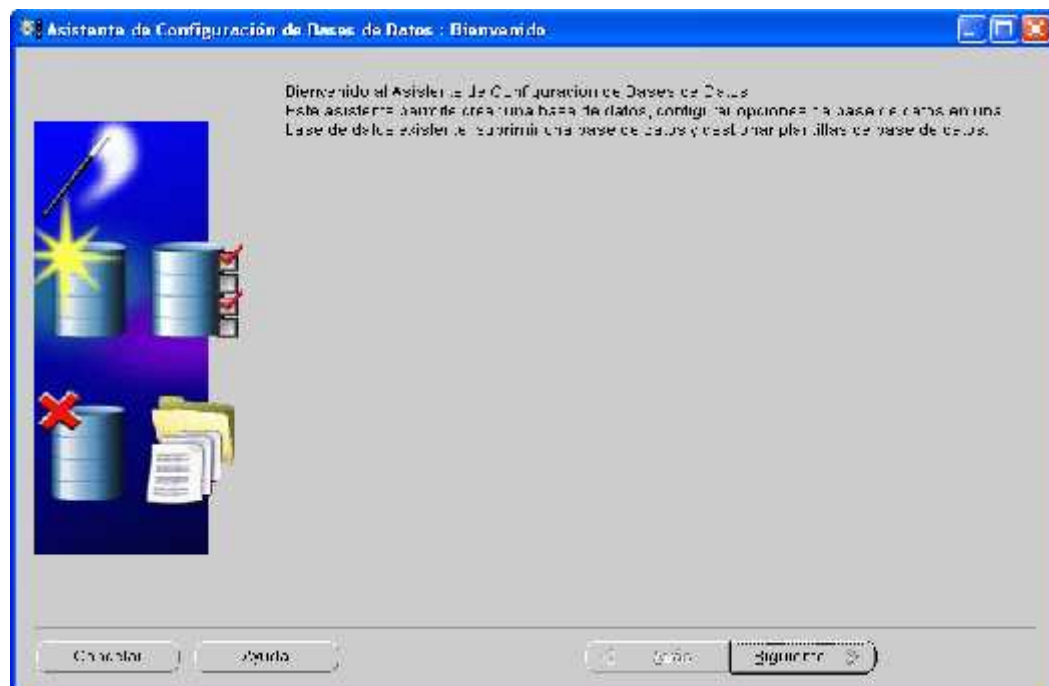
Para proceder a la creación de la base de datos se requieren tener instalado el software de base de datos.

A continuación se detallan los pasos a seguir para la creación de la base de datos oracle 9i en sistemas operativo Windows XP.

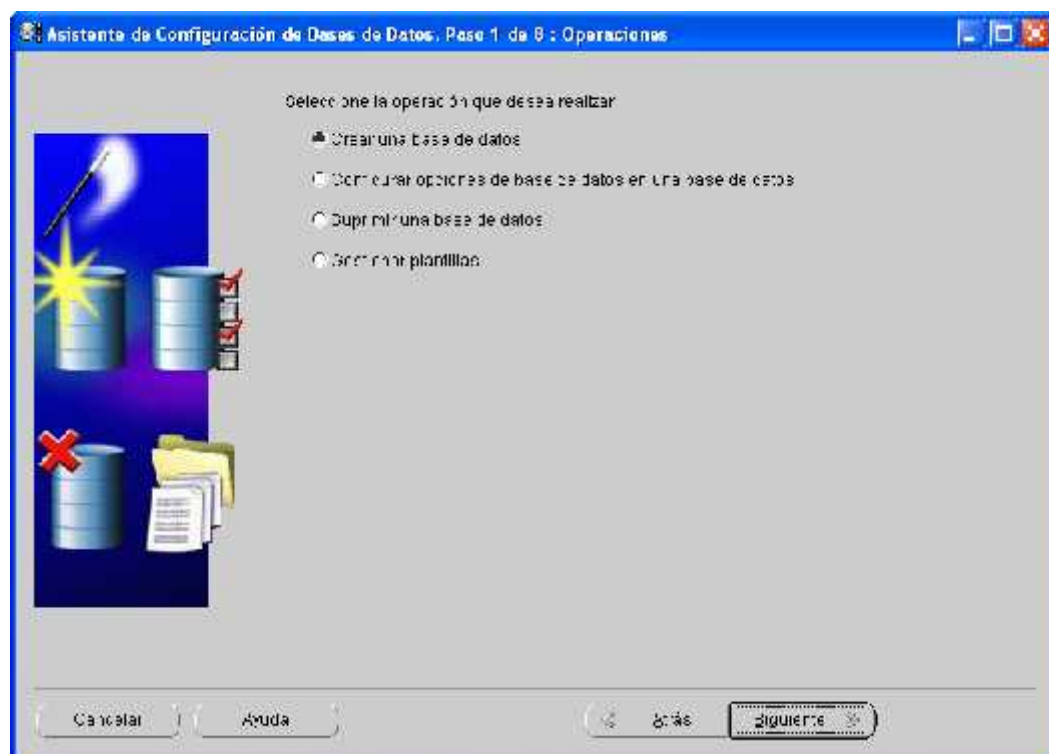
1. Para comenzar con la creación de la base de datos se debe ingresar al menú Oracle – OraHome92 (en el que se instaló la base de datos) elegir el submenú Configuration and Migration Tools y dar clic en la opción del Asistente de Configuración de Base de Datos (Database Configuration Assistant), el cual con dicha acción presenta la siguiente pantalla.



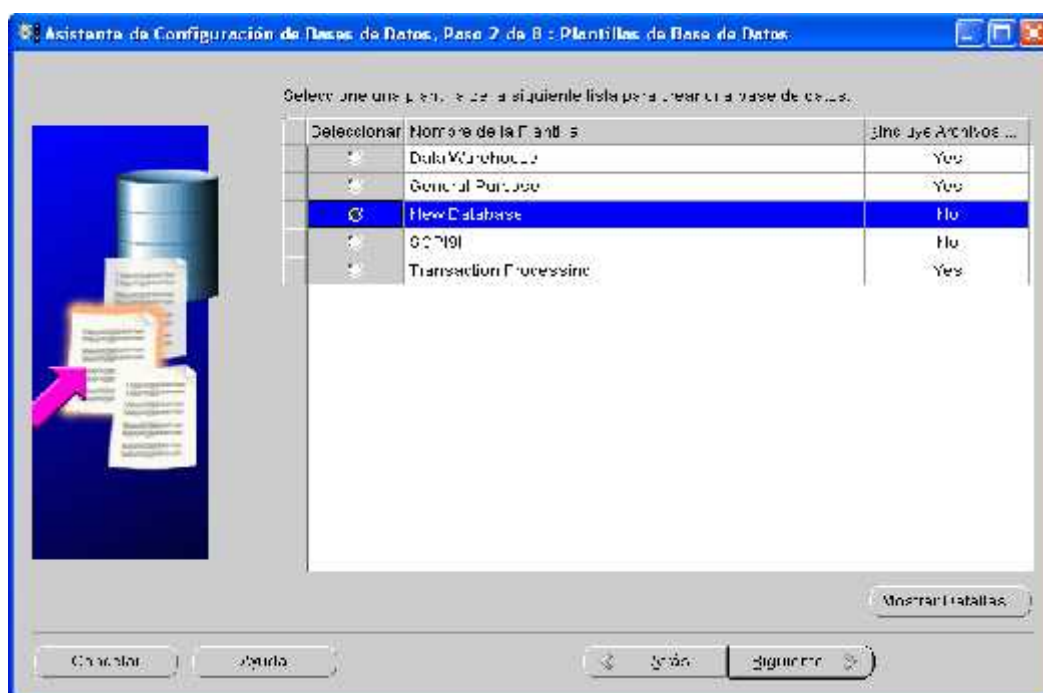
2. Bienvenido al Asistente de Configuración de Base de Datos, el cual nos va a permitir crear la base de datos. Luego presione el botón Siguiente.



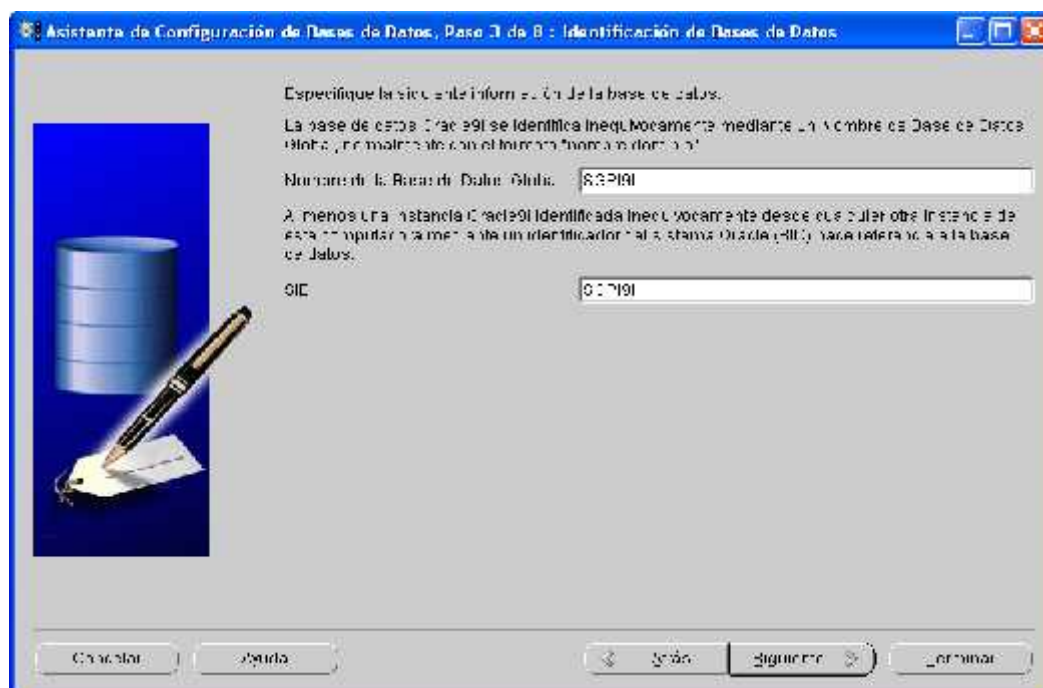
3. Seleccionamos la opción de Crear una base de datos, una vez creada la base de datos se pueden cambiar la configuración inicial de la base de datos. Luego presione el botón Siguiente.



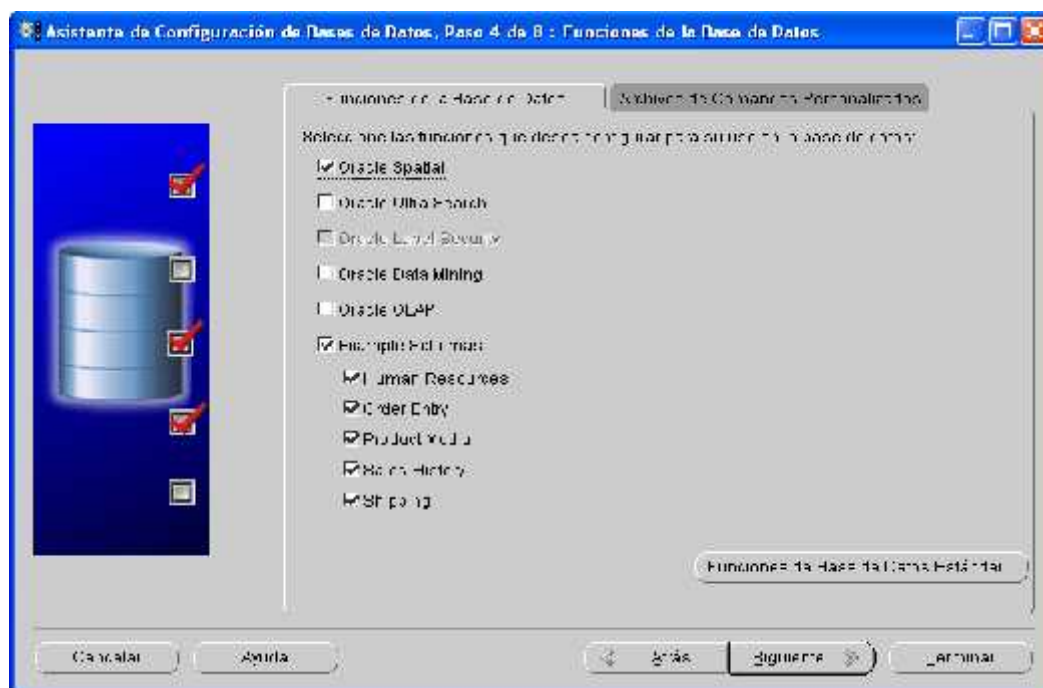
4. Seleccionamos la plantilla de la base de datos que vamos a utilizar, en este caso New Database para gestionar la creación de una nueva base de datos. Si desea ver los detalles (ubicación, nombre de archivos, parámetros de inicialización, etc.) de creación de la base de datos presione el botón Mostrar Detalles. Luego presione el botón Siguiente.



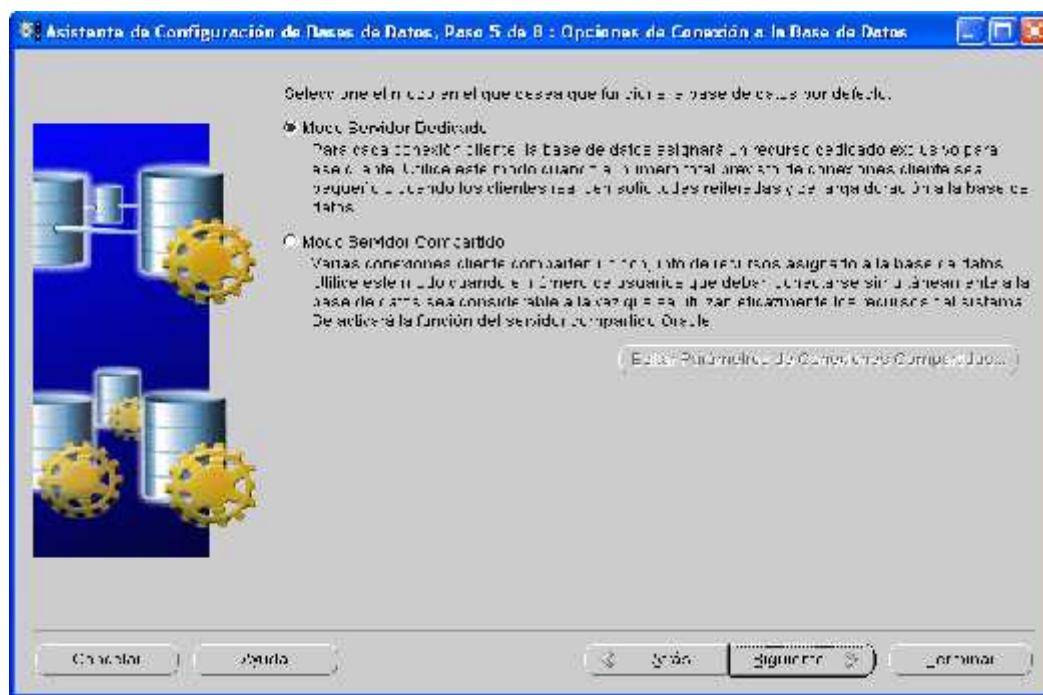
5. Procedemos a la identificación de la base de datos, en la cual especificamos el nombre de la base con un máximo de 8 caracteres. El SID es un identificador interno que utilizará Oracle para referenciar a nuestra base de datos, se puede elegir uno diferente al del nombre de la base de datos, aunque se suele utilizar el mismo como en este caso. Luego presione el botón Siguiente.



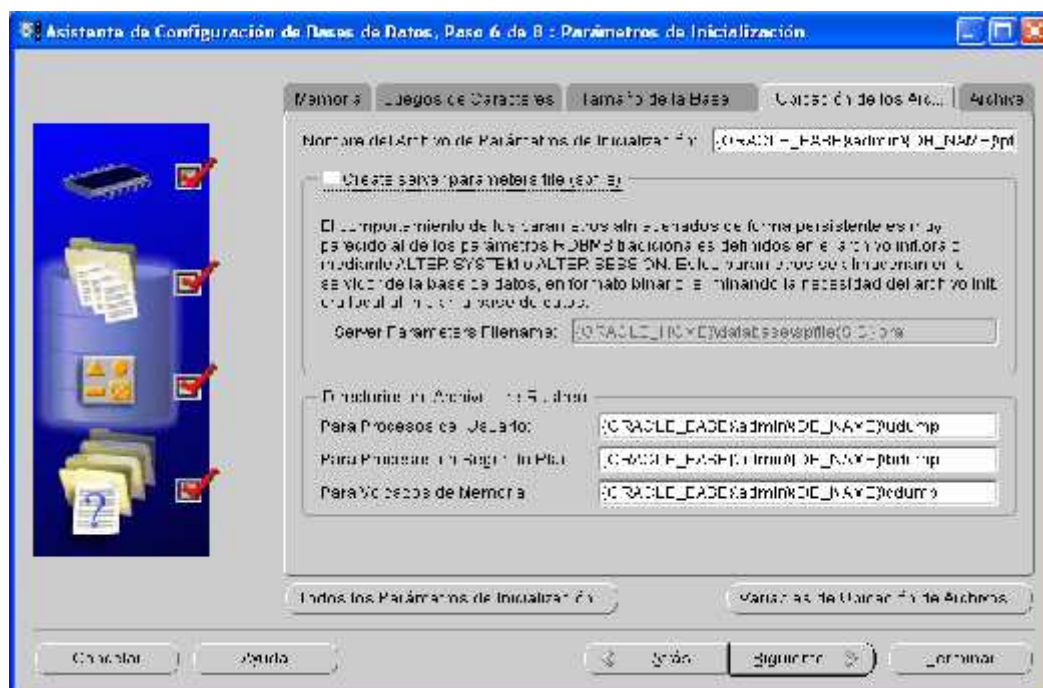
6. Seleccionamos las Funciones de Base de Datos que vamos a utilizar, en este caso seleccionaremos Oracle Spatial y Example Schemas (Esquemas de ejemplo). Luego presione el botón Siguiente.



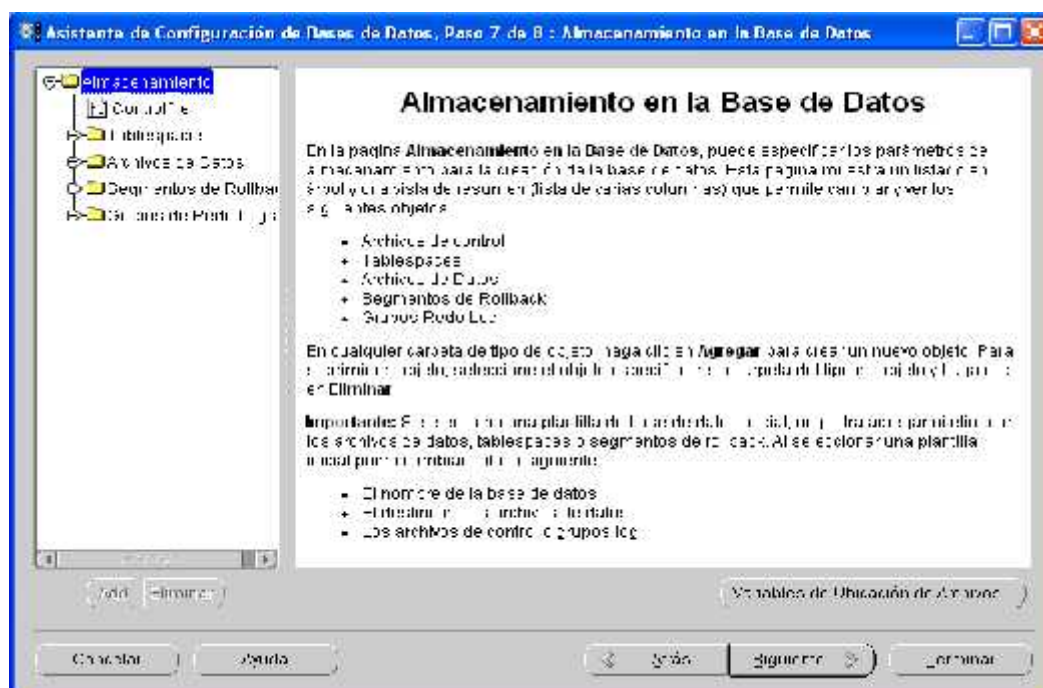
7. Seleccionamos el modo de conexión con el que va a funcionar la base de datos, en este caso seleccionaremos el Modo de Servidor Dedicado, el cual es para un número pequeño de clientes conectados a la base de datos. Luego presione el botón Siguiente.



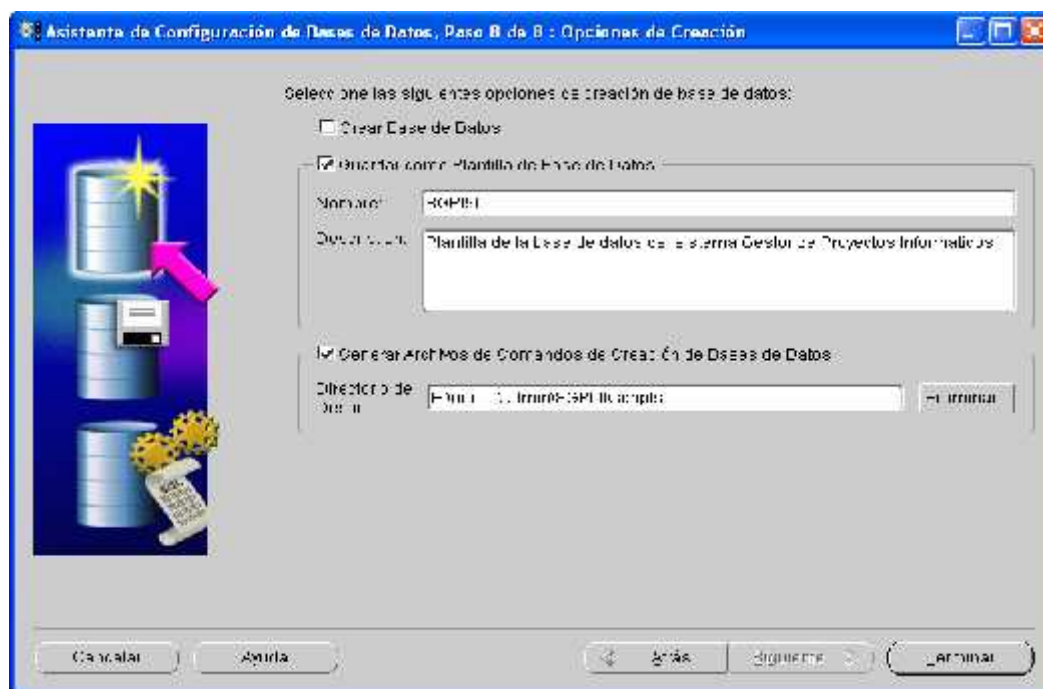
8. Se presentan los parámetros de inicialización de la base de datos. Con respecto a las pestañas de la memoria, juego de caracteres, tamaño de la base de datos y modo de archive se elegirán los valores por defecto de los parámetros de inicialización de la base de datos. En la pestaña de Ubicación de los Archivos, no se va a seleccionar crear el parámetro spfile como lo muestra por defecto al inicializar los parámetros ya que este se lo creará por medio de comandos. Luego presione el botón Siguiente.



9. En este punto de la creación de la base de datos nos muestra en una ventana en la que se especifica los parámetros de almacenamiento para la creación de la base de datos. Luego presione el botón Siguiente.



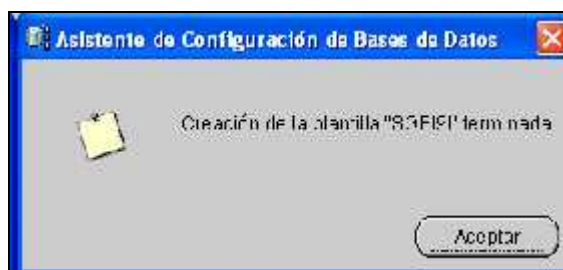
10. Seleccionaremos las opciones de creación de la base de datos. En este caso vamos a seleccionar Guardar como plantilla de Base de Datos, en la cual colocaremos el nombre de la plantilla y una pequeña descripción de la misma; seleccionaremos también la opción de Generar Archivos de Comandos de Creación de Base de Datos en el que elegiremos la ruta de creación del scripts de la base de datos. Luego presione el botón Terminar.



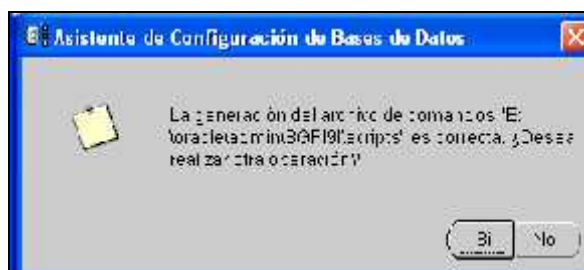
11. Se presenta una pantalla en la que nos muestra un resumen de las operaciones que realizará. Luego presione el botón Aceptar.



12. Aparecerá una ventana en la que muestra que la plantilla ha sido terminada. Luego presione el botón Aceptar.



13. Aparecerá una ventana en la que muestra el directorio en el que se creó el archivo de comandos (scripts) de creación de la base de datos, y si esta generación fue correcta. Luego presione el botón Aceptar.




```

C:\WINDOWS\system32\cmd.exe

E:\oracle\admin\SGPI9I\scripts>mkdir E:\oracle\oradata\SGPI9I

E:\oracle\admin\SGPI9I\scripts>set ORACLE_SID=SGPI9I

E:\oracle\admin\SGPI9I\scripts>E:\oracle\ora92\bin\oradim.exe -new -sid SGPI9I
-startnode n -pfile E:\oracle\ora92\database\initSGPI9I.ora

E:\oracle\admin\SGPI9I\scripts>E:\oracle\ora92\bin\oradim.exe -edit -sid SGPI9I
-startnode a

E:\oracle\admin\SGPI9I\scripts>E:\oracle\ora92\bin\orapwd.exe file=E:\oracle\ora
92\database\PWD$SGPI9I.ora password=change_on_install

E:\oracle\admin\SGPI9I\scripts>E:\oracle\ora92\bin\sqlplus /nolog EE:\oracle\adm
in\SGPI9I\scripts>CreateDB.sql

SQL*Plus: Release 9.2.0.1.0 - Production on 08 Dic 8 17:47:56 2007

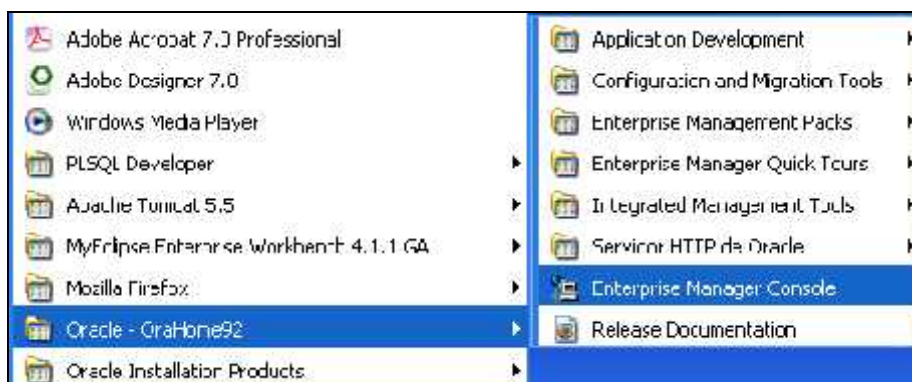
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Conectado a una instancia inactiva.
SQL> spool E:\oracle\ora92\assistants\dba\logs\CreateDB.log
SQL> startup nomount pfile="E:\oracle\admin\SGPI9I\scripts\init.ora";

```

4. Creación De Conexión A La Base De Datos

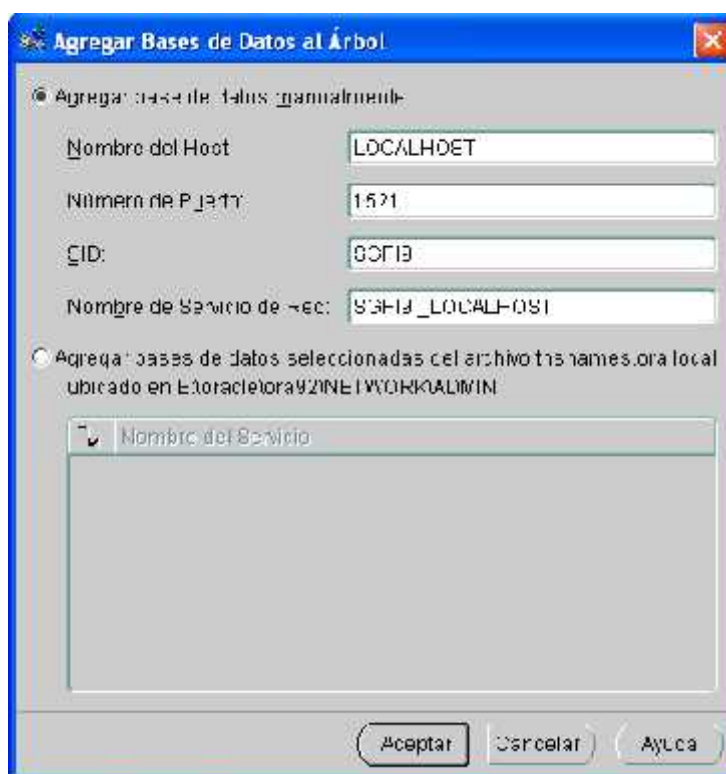
1. Para comenzar con la creación de una conexión a la base de datos por medio de la consola del Enterprise Manager, se debe ingresar al menú Oracle – OraHome92 (en el que se instaló la base de datos) elegir la opción del Enterprise Manager Console, el cual con dicha acción presenta la siguiente pantalla.



2. Bienvenido a la Consola de Oracle Enterprise Manager, el cual nos va a permitir crear la conexión a la base de datos. Elegimos la opción de Iniciar en Modo Autónomo. Luego presione el botón Aceptar.



3. Seleccionamos la opción de Crear una base de datos, una vez creada la base de datos se pueden cambiar la configuración inicial de la base de datos. Luego presione el botón Aceptar.



5. Instalación del J2SE 5.0 JDK (JAVA 2 STANDARD EDITION 5.0 JAVA DEVELOPMENT KIT).

El J2SE 5.0 JDK, sirve para crear aplicaciones JAVA.

Para proceder a la instalación del J2SE 5.0 JDK también conocido como SDK (Software Development Kit), del entorno de ejecución de Java 2, Edición Estándar (JRE), se requiere poseer el instalador del J2SE 5.0 JDK (jdk-1_5_0-windows-i586.exe) para S.O. Windows el cual se lo puede descargar gratuitamente desde la propia página web de Sun Microsystems (http://java.sun.com/javase/downloads/index_jdk5.jsp).

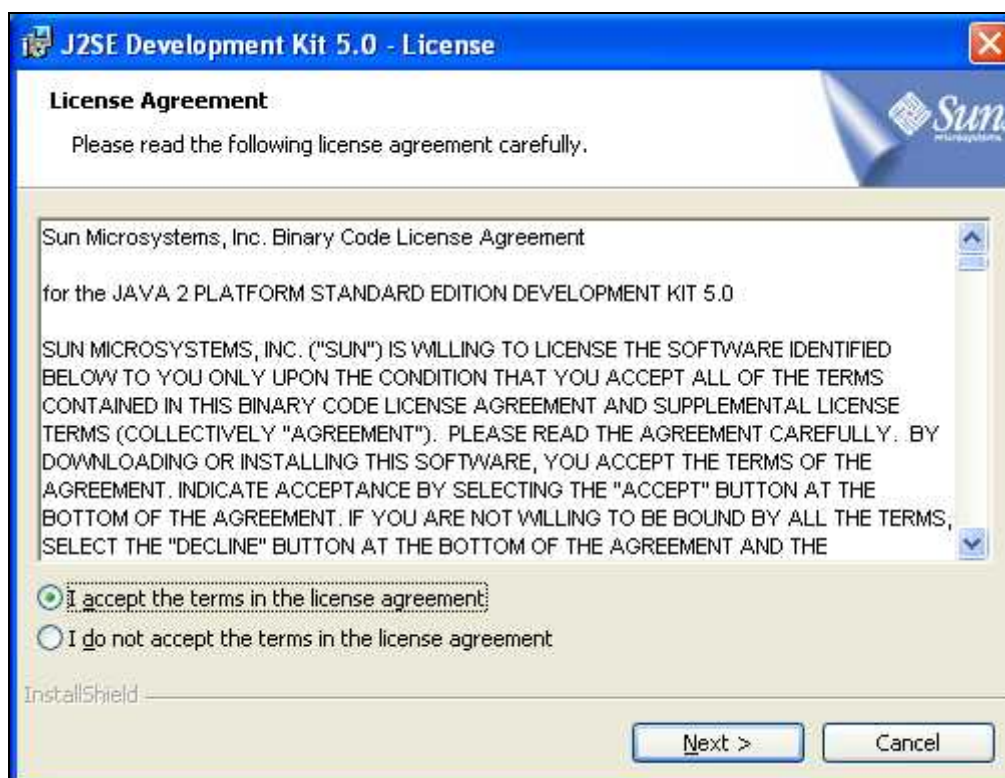
A continuación se detallan los pasos a seguir para la instalación del J2SE 5.0 JDK en sistemas operativo Windows XP.

1. Ejecutamos el archivo que descargamos de la página web mencionada anteriormente. Se realiza la ejecución del archivo jdk-1_5_0-windows-i586.exe, una vez ejecutado se comienza a extraer los archivos de instalación y se presentan las siguientes pantallas.

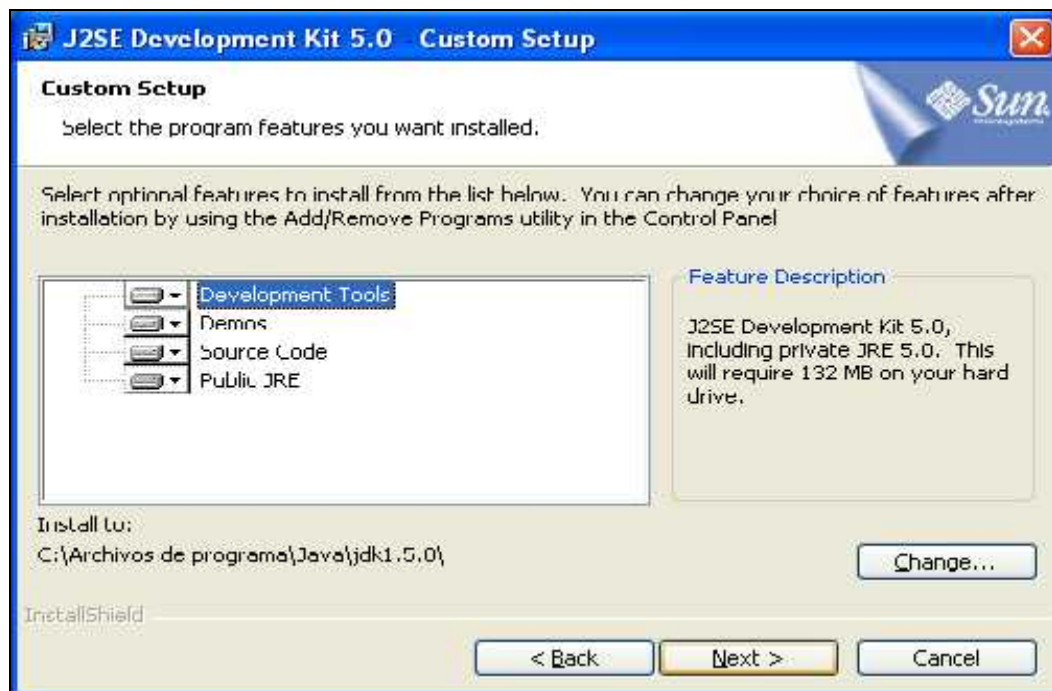




2. Seleccionamos la opción que indica que acepta los términos de la licencia (I accept the terms in the license agreement). Presionamos el botón Next para obtener la siguiente pantalla.



3. Escogemos la opción de Herramientas de Desarrollo (Development Tools). Presionamos el botón Next, en la que dicha acción instalará el J2SE Development Kit 5.0 para obtener la siguiente pantalla.



4. Seleccionamos el navegador de internet en el que deseamos registrar la instalación realizada. Presionamos el botón Next para obtener la siguiente pantalla.



5. Luego de la instalación de J2SE 5.0 JDK aparecerá una ventana indicando que el proceso de instalación ha finalizado. Luego presione el botón Finish.



6. Con esto finaliza la instalación del servidor J2SE 5.0 JDK.

6. Instalación Del Entorno De Desarrollo Integrado Eclipse

Un entorno de desarrollo integrado (IDE), puede hacernos el trabajo más sencillo, sobre todo si nuestro desarrollo maneja un buen número de clases. Eclipse es un IDE de código abierto.

Para proceder a la instalación del IDE Eclipse en su versión 3.1.2 se requiere poseer el archivo ZIP (eclipse-SDK-3.1.2-win32.zip) el cual se lo puede descargar gratuitamente desde la propia página web de eclipse (www.eclipse.org).

Su instalación es sencilla, solo tenemos que descomprimir el archivo ZIP en el directorio en donde deseamos tenerlo instalado. Crearemos una carpeta con nombre bin en la unidad C, obteniendo el directorio C:\bin, en el cual vamos descomprimir el archivo ZIP, como se muestra en la pantalla.



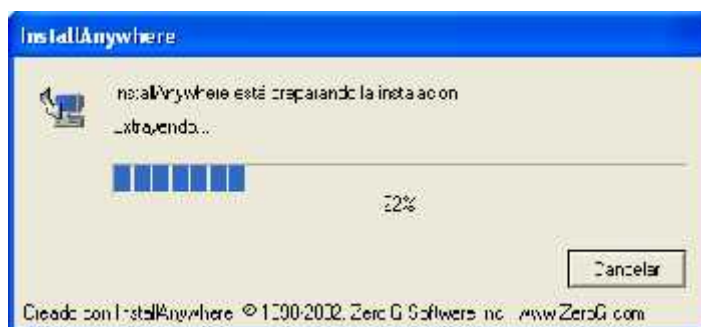
7. Instalación Del Entorno De Desarrollo Integrado Myeclipse

MyEclipse es un IDE que se basa en la plataforma de Eclipse.

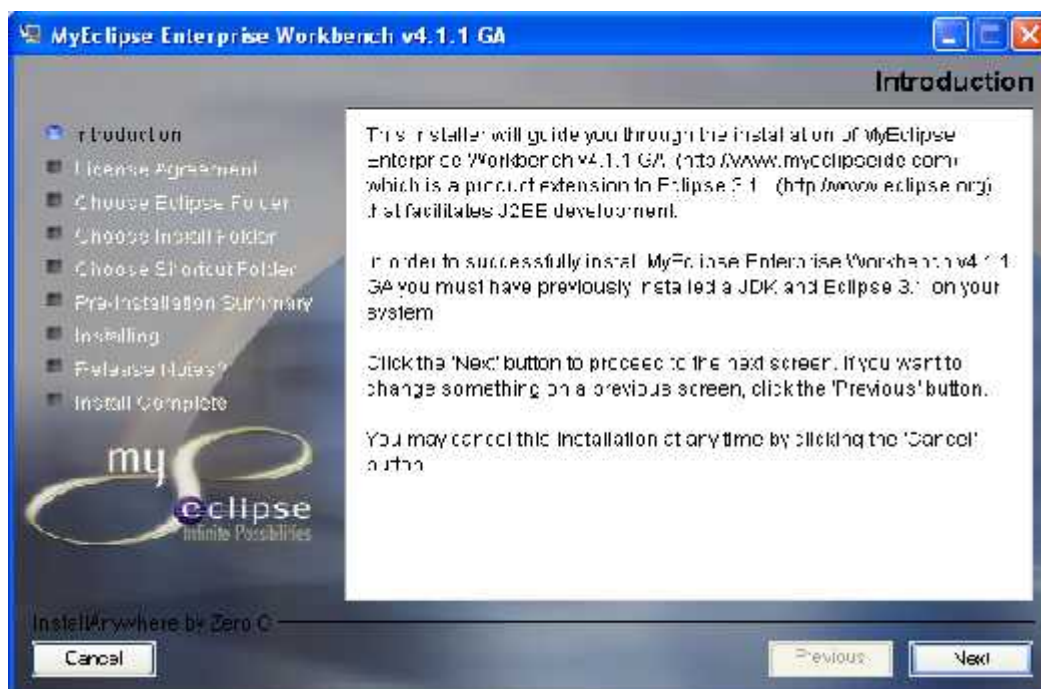
Para proceder a la instalación del IDE MyEclipse en su versión 4.1.1, se requiere poseer el archivo ejecutable de MyEclipse o instalador (EnterpriseWorkbenchInstaller_4.1.1GA_E3.1.exe) el cual se lo puede descargar gratuitamente desde la propia página web de MyEclipse (<http://www.myeclipseide.com/>).

A continuación se detallan los pasos a seguir para la instalación del IDE MyEclipse 4.1.1 en sistema operativo Windows XP.

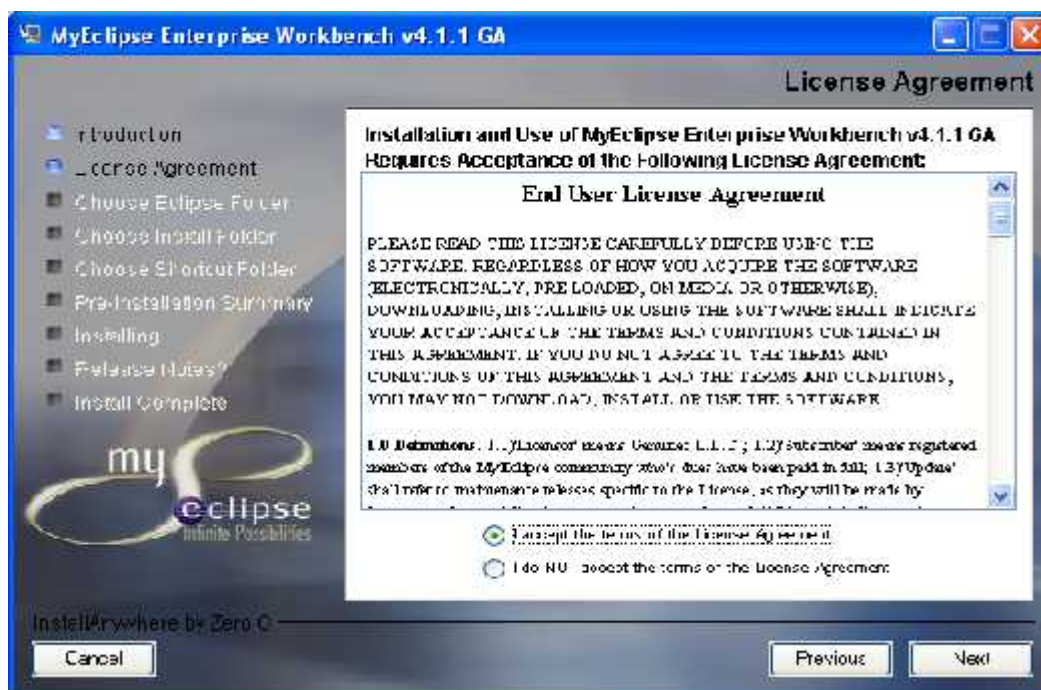
1. Ejecutamos el archivo que descargamos de la página web mencionada anteriormente. Se realiza la ejecución del archivo EnterpriseWorkbenchInstaller_4.1.1GA_E3.1.exe, una vez ejecutado se comienza a extraer los archivos de instalación y se presenta la siguiente pantalla.



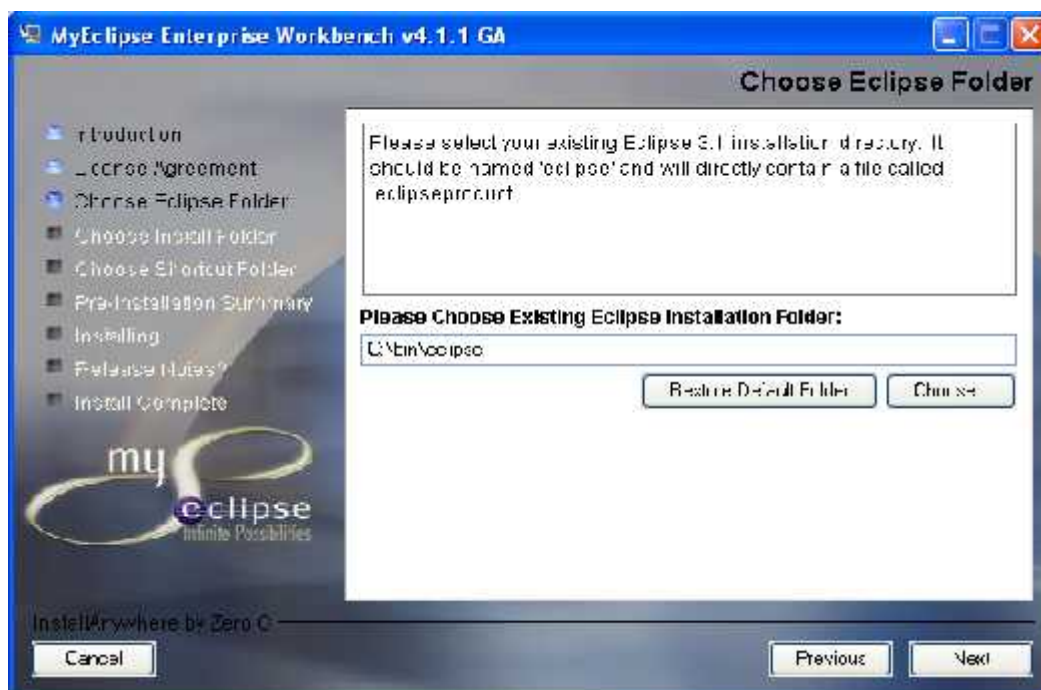
2. Bienvenidos a la instalación de MyEclipse. Presionamos el botón Next para obtener la siguiente pantalla.



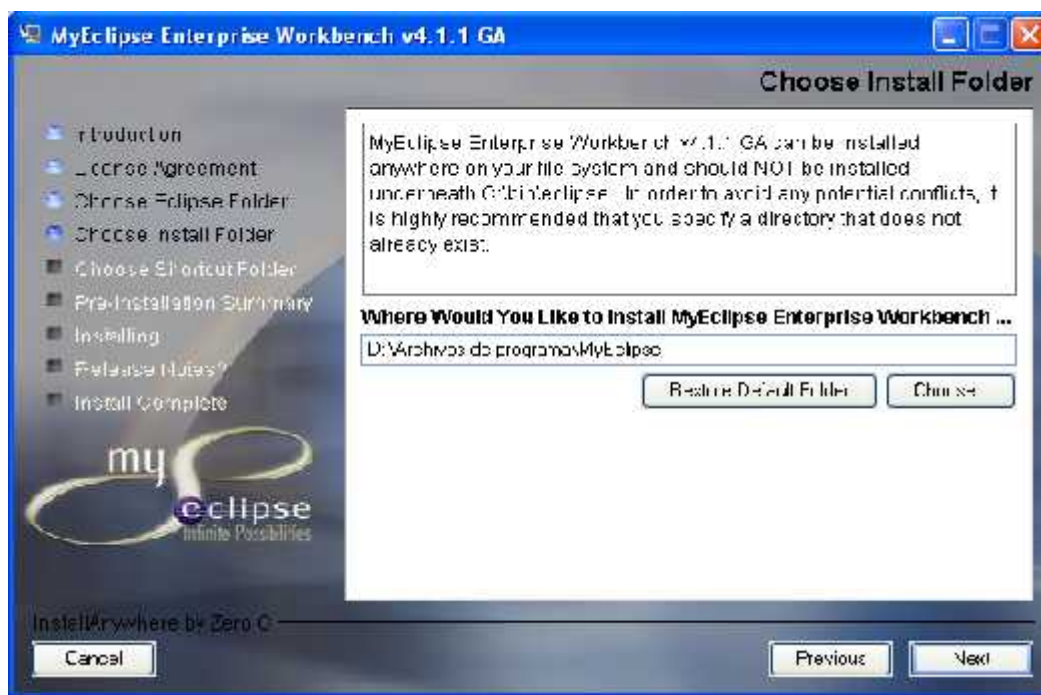
3. Seleccionamos la opción que indica que acepta los términos de la licencia (I accept the terms of the License Agreement). Presionamos el botón Next para obtener la siguiente pantalla.



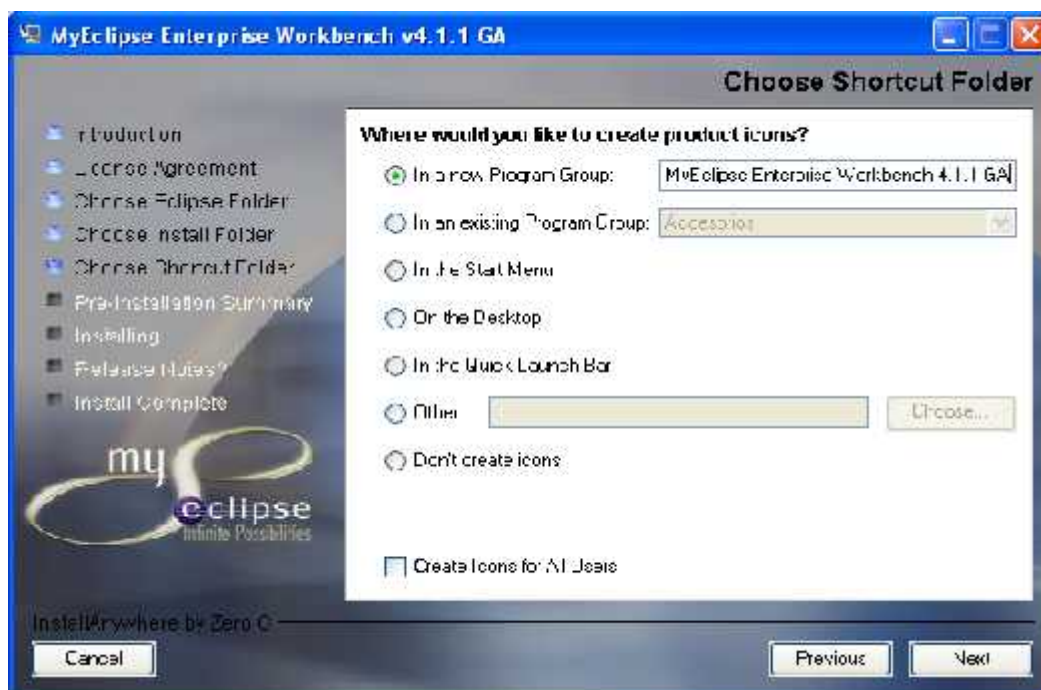
4. Escogemos la ruta en la que se encuentra instalado el IDE Eclipse, es este caso C:\bin\eclipse. Presionamos el botón Next, para obtener la siguiente pantalla.



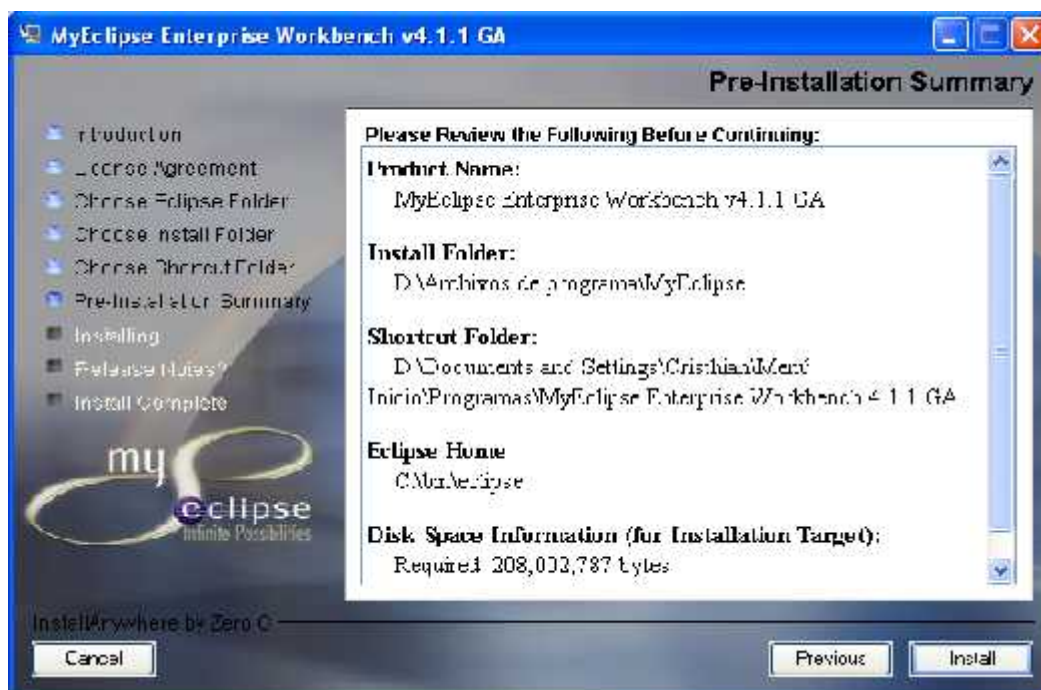
5. Escogemos la ruta en la que desea instalar el IDE MyEclipse, es este caso dejamos seleccionada la opción por defecto D:\Archivos de programa\MyEclipse. Presionamos el botón Next, para obtener la siguiente pantalla.



6. Escogemos la opción de en donde deseamos crear iconos del producto MyEclipse, en este caso dejamos seleccionada la opción por defecto In a new Program Group. Presionamos el botón Next, para obtener la siguiente pantalla.



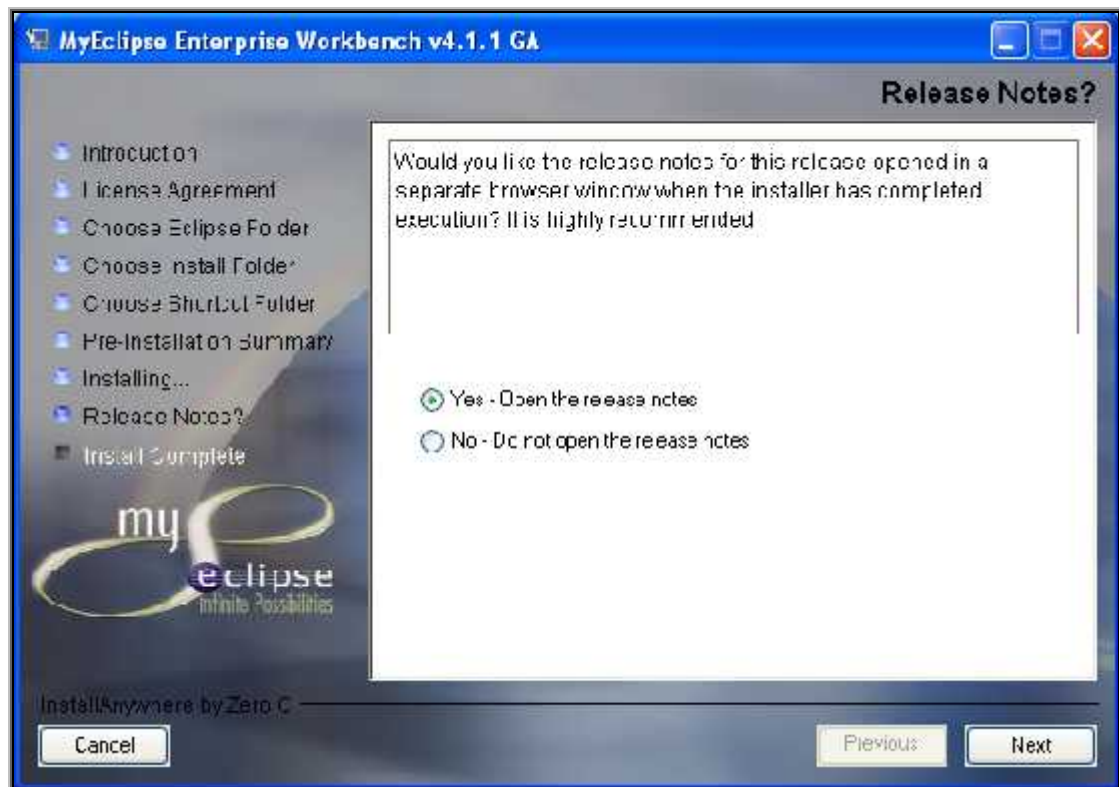
7. Se presenta una pantalla en la que debemos verificar la ruta y espacio de las opciones seleccionadas anteriormente, para proceder a la instalación. Presionamos el botón Install, para obtener la siguiente pantalla.



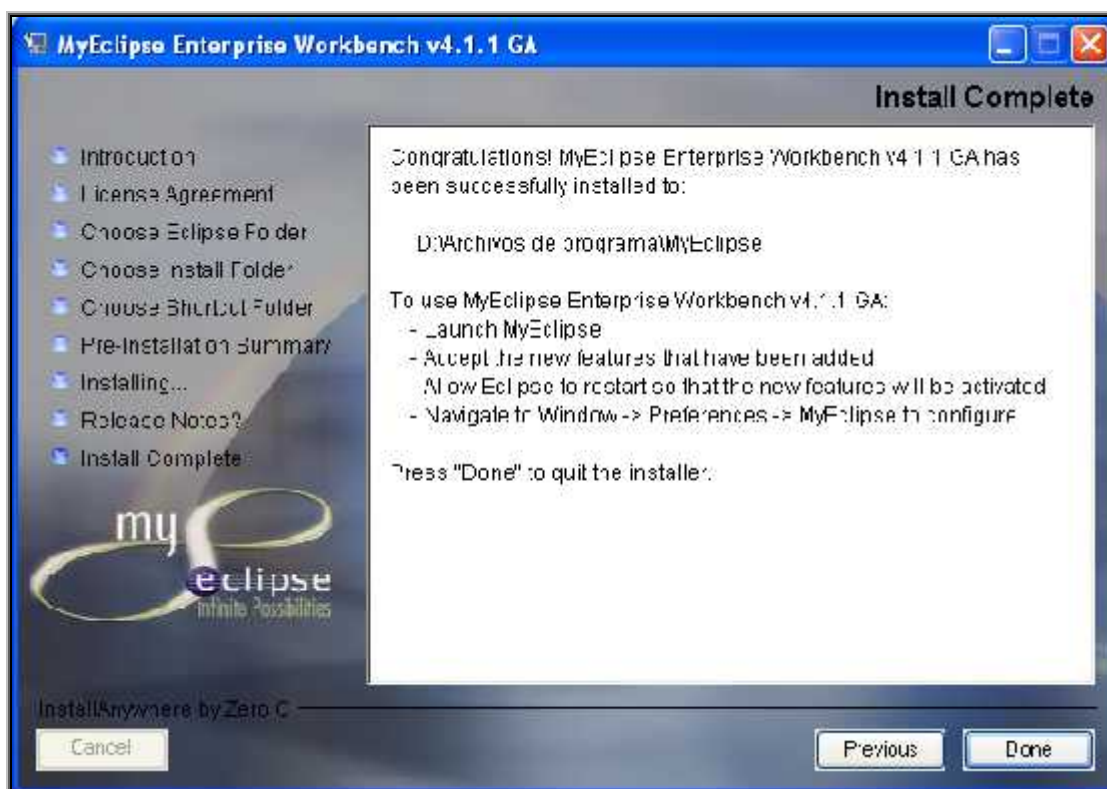
8. Se instalará el IDE MyEclipse.



9. Seleccionamos la opción de abrir para leer el archivo que posee las características del IDE MyEclipse. Presionamos el botón Next para obtener la siguiente pantalla.



10. Luego de la instalación del IDE MyEclipse aparecerá una ventana indicando que el proceso de instalación ha finalizado. Luego presione el botón Done.



11. Con esto finaliza la instalación del servidor IDE MyEclipse.

8. Instalación Del Servidor De Aplicaciones Tomcat 5.5

Tomcat también conocido como Jakarta Tomcat o Apache Tomcat, funciona como un servidor de servlets. Tomcat implementa las especificaciones de los servlets y de los JSP de Sun Microsystems.

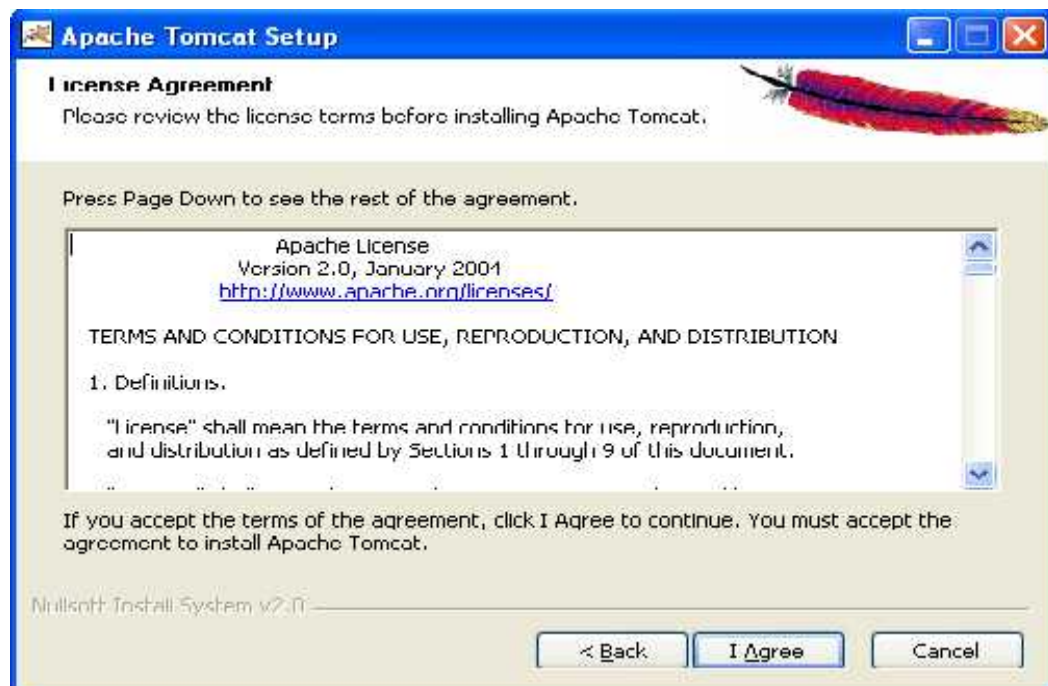
Para proceder a la instalación del Servidor de Aplicaciones Tomcat en su versión 5.5, se requiere poseer el archivo ejecutable del Tomcat o instalador (jakarta-tomcat-5.5.9.exe) el cual se lo puede descargar gratuitamente desde la propia página Web del servidor de aplicaciones Apache Tomcat (<http://tomcat.apache.org/download-55.cgi>).

A continuación se detallan los pasos a seguir para la instalación del servidor de aplicaciones Tomcat en sistema operativo Windows XP.

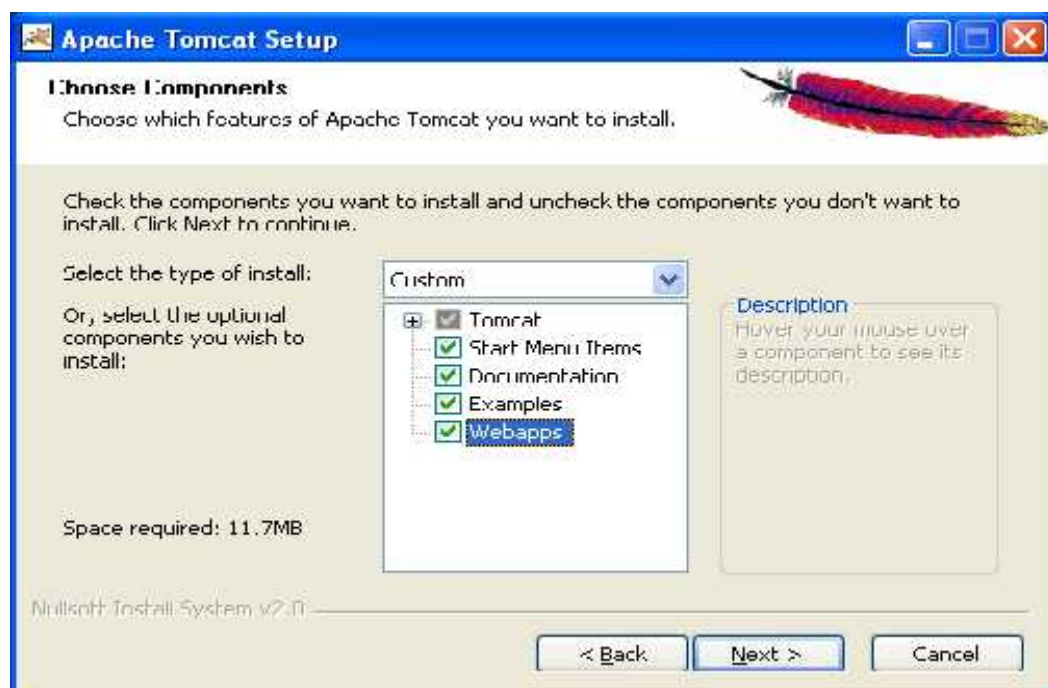
1. Ejecutamos el archivo que descargamos de la página Web mencionada anteriormente. Se realiza la ejecución del archivo jakarta-tomcat-5.5.9.exe, una vez ejecutado se comienza a extraer los archivos de instalación. Presionamos el botón Next para obtener la siguiente pantalla.



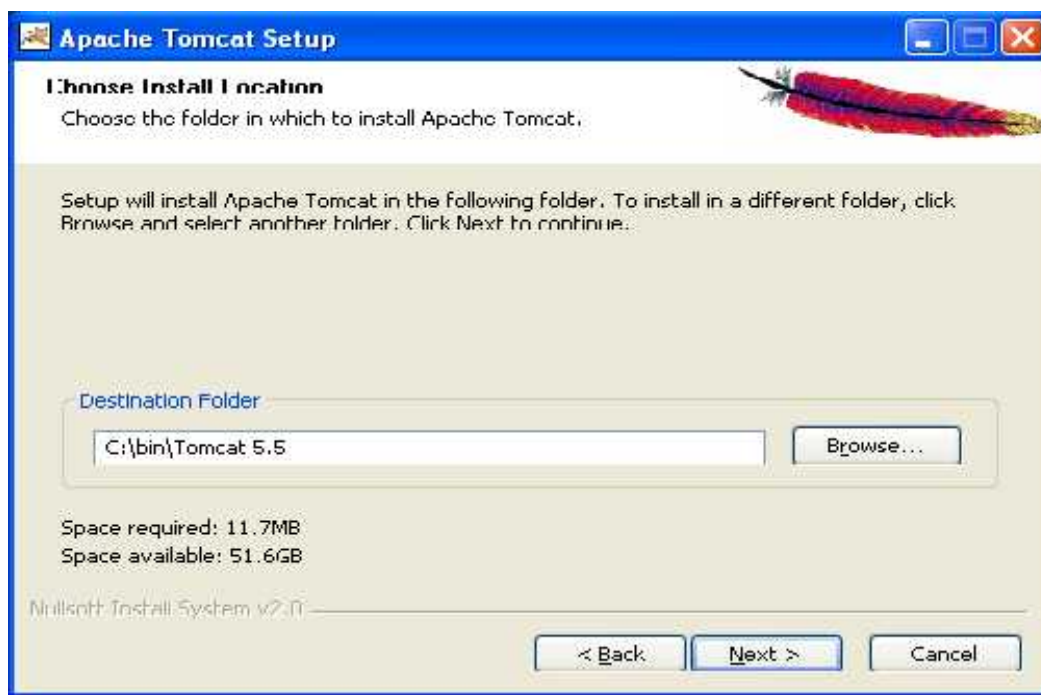
2. Bienvenidos a la instalación del servidor de aplicaciones Tomcat. Presionamos el botón I Agree para aceptar los términos de la licencia y obtener la siguiente pantalla.



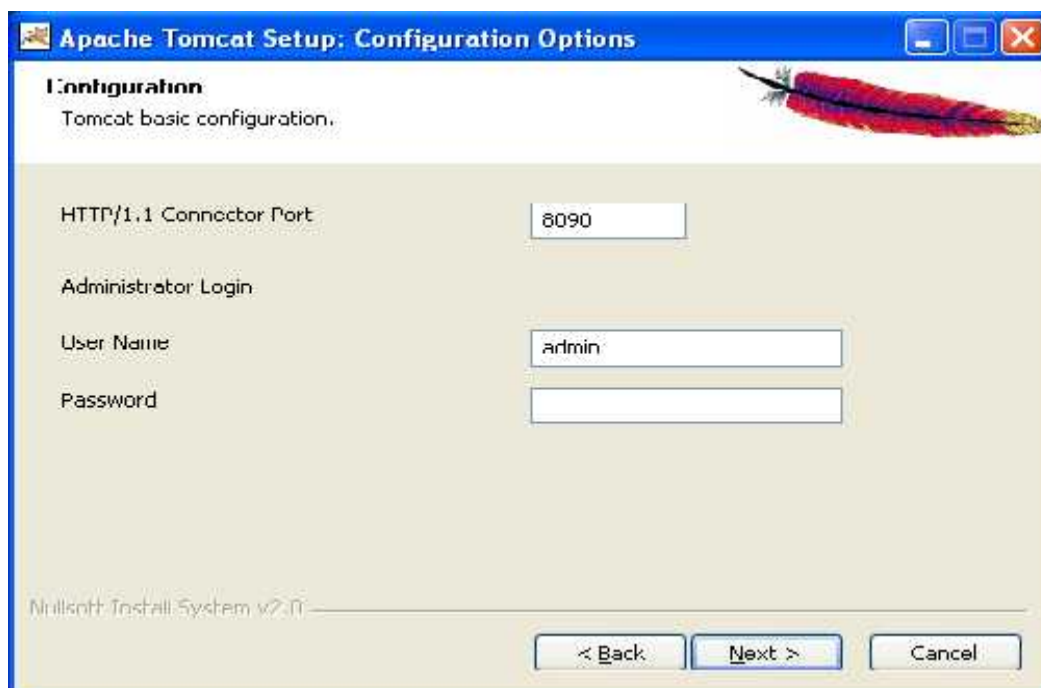
3. Escogemos los componentes que deseamos instalar con el servidor de aplicaciones. Presionamos el botón Next, para obtener la siguiente pantalla.



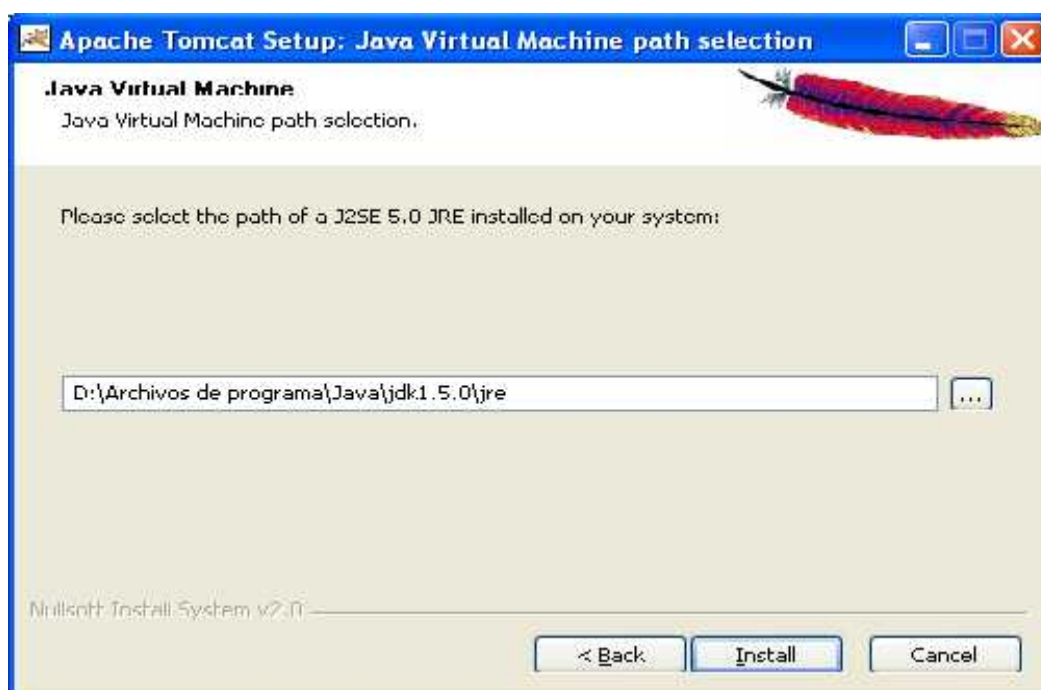
4. Escogemos la ruta en la que desea instalar el servidor de aplicaciones Apache Tomcat, en este caso seleccionamos la ruta C:\bin\Tomcat 5.5. Presionamos el botón Next, para obtener la siguiente pantalla.



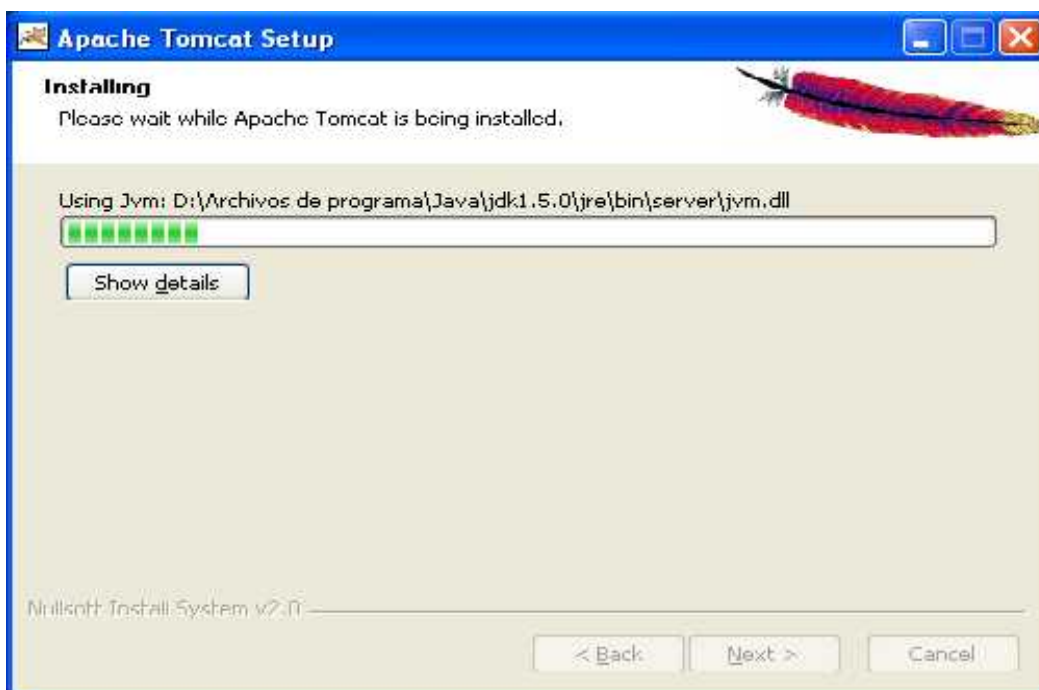
5. Colocamos el puerto por el cual se va a conectar el servidor de aplicación, en este caso 8090 porque el puerto por defecto (8080) por lo general siempre está ocupado con otra aplicación. Colocamos el usuario y clave, en este caso dejamos el usuario por defecto admin y no colocaremos ninguna clave. Presionamos el botón Next, para obtener la siguiente pantalla.



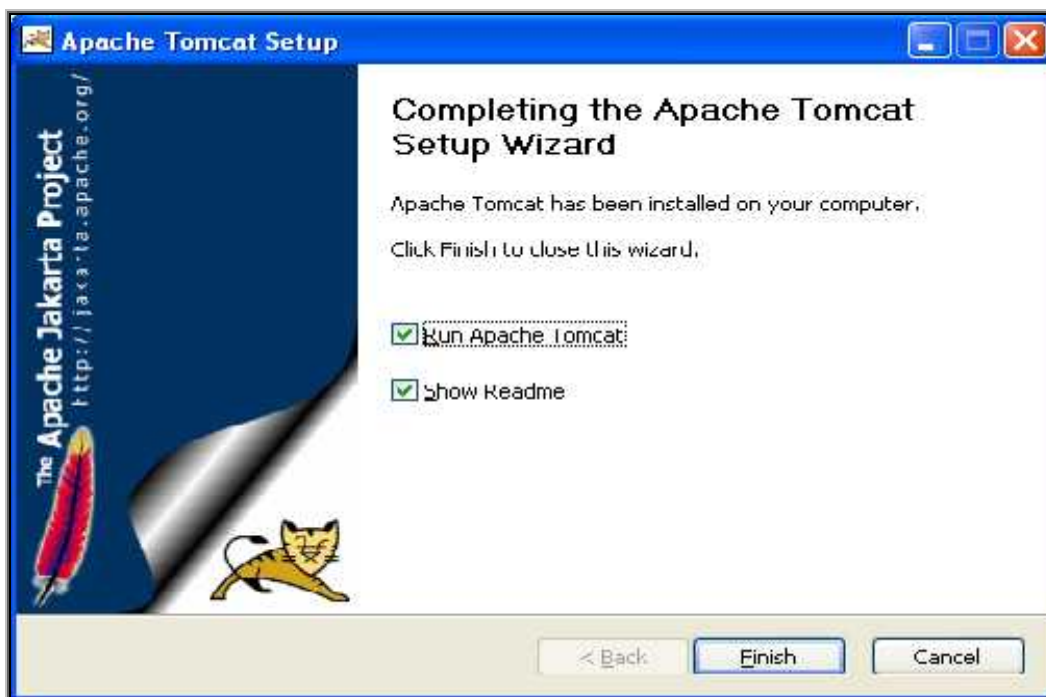
6. Escogemos la ruta en la que se encuentra instalada la máquina virtual de java C:\Archivos de programa\Java\jdk1.5.0\jre, para proceder a la instalación. Presionamos el botón Install, para obtener la siguiente pantalla.



7. Se instalará el Servidor de Aplicaciones Apache Tomcat.



8. Luego de la instalación del Servidor de Aplicaciones Tomcat aparecerá una ventana indicando que el proceso de instalación ha finalizado. Luego presione el botón Finish.

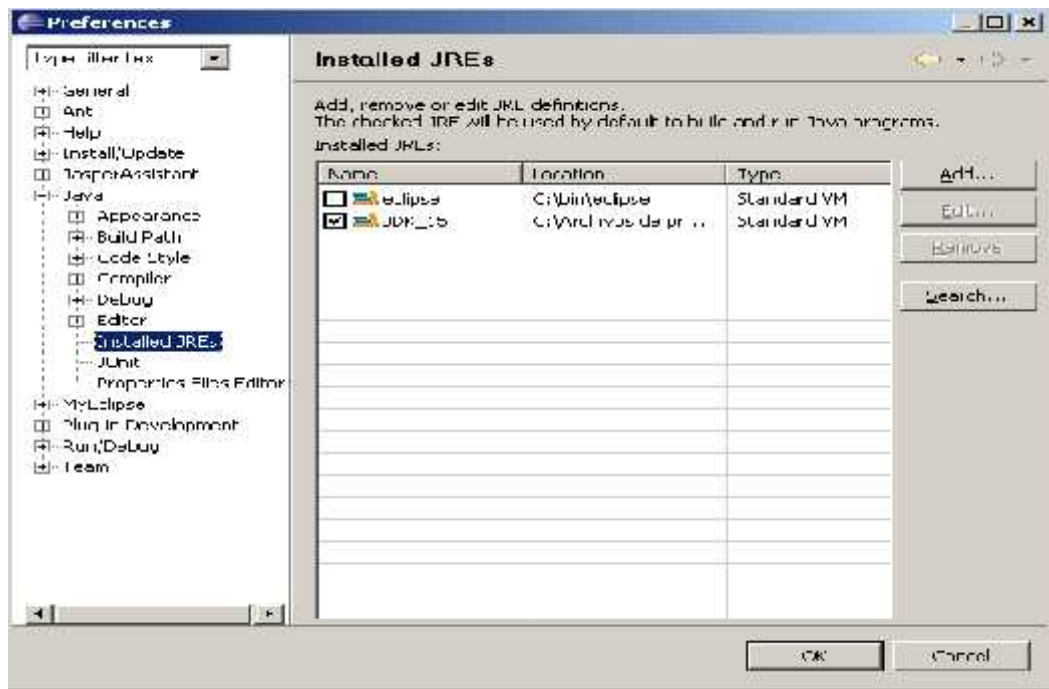


9. Con esto finaliza la instalación del Servidor de Aplicaciones Apache Tomcat.

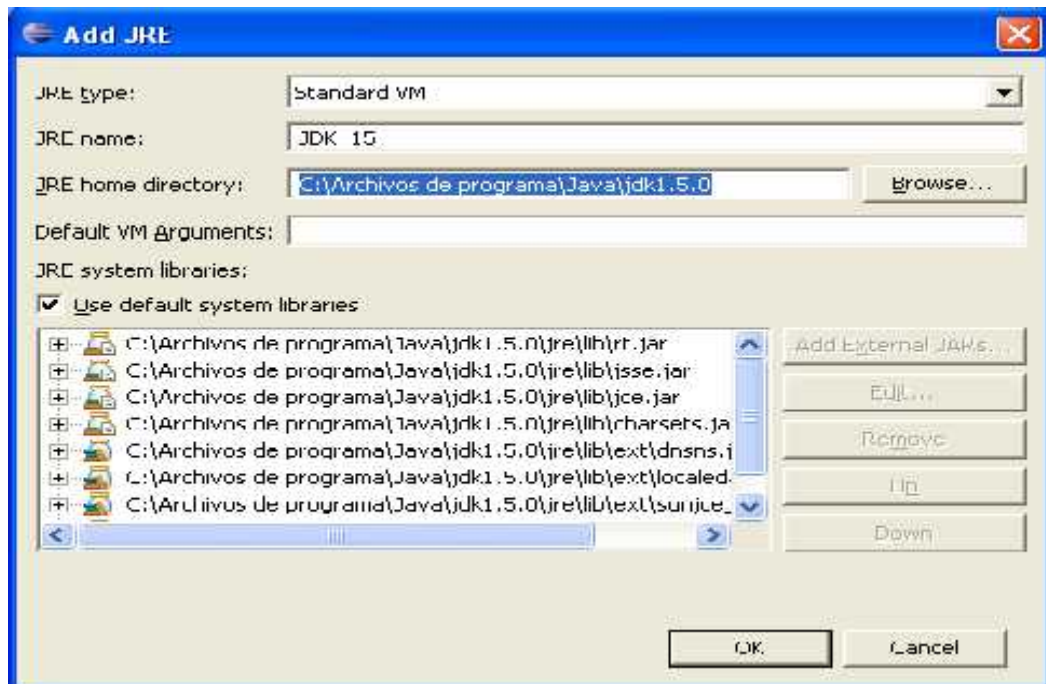
9. Configuración Del Entorno De Desarrollo Integrado Eclipse

1. Añadir el kit de java el cual va a utilizar el Servidor de Aplicaciones Tomcat, que se usará en el proyecto

En la barra de menú de Eclipse selecciono Windows → Preferences → Java → Installed JREs. Se presentará una ventana donde estarán todos los JRE que podrían ser usados en un proyecto. Se deberá añadir el JRE que se usará para nuestro proyecto.



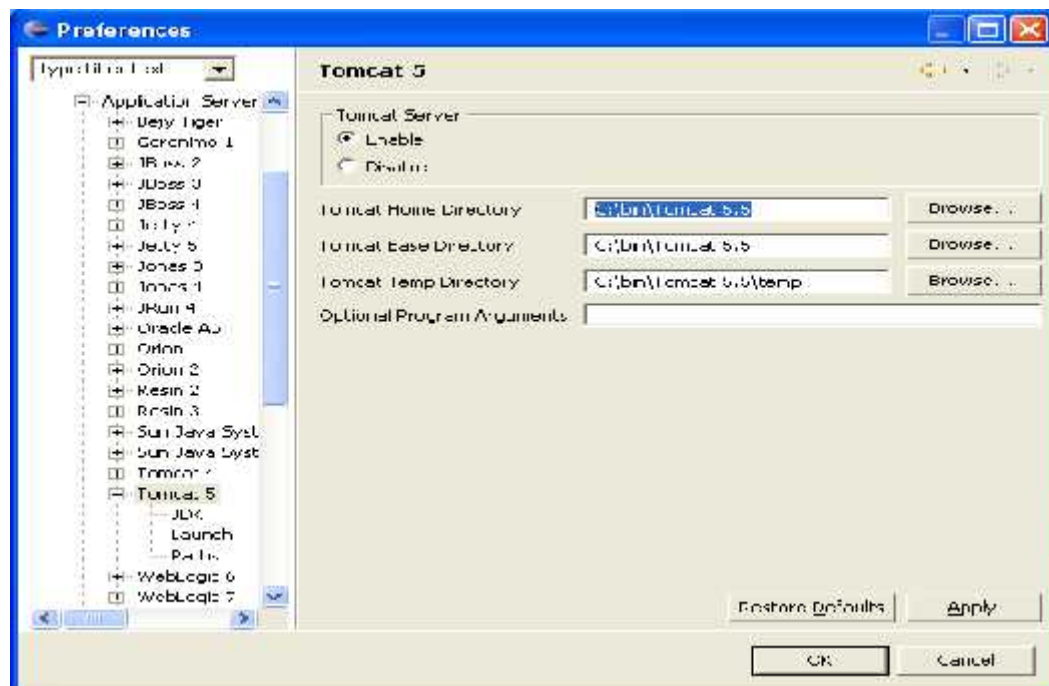
Seleccionamos Add → En el Browse buscamos la ruta en el cual tenemos instalado el JRE que usaremos para utilizarlo en el Servidor de Aplicaciones C:\Archivos de programa\Java\jdk1.5.0; y le asignamos un nombre(JDK_15) al JRE que hemos añadido.



Y por último presionamos OK.

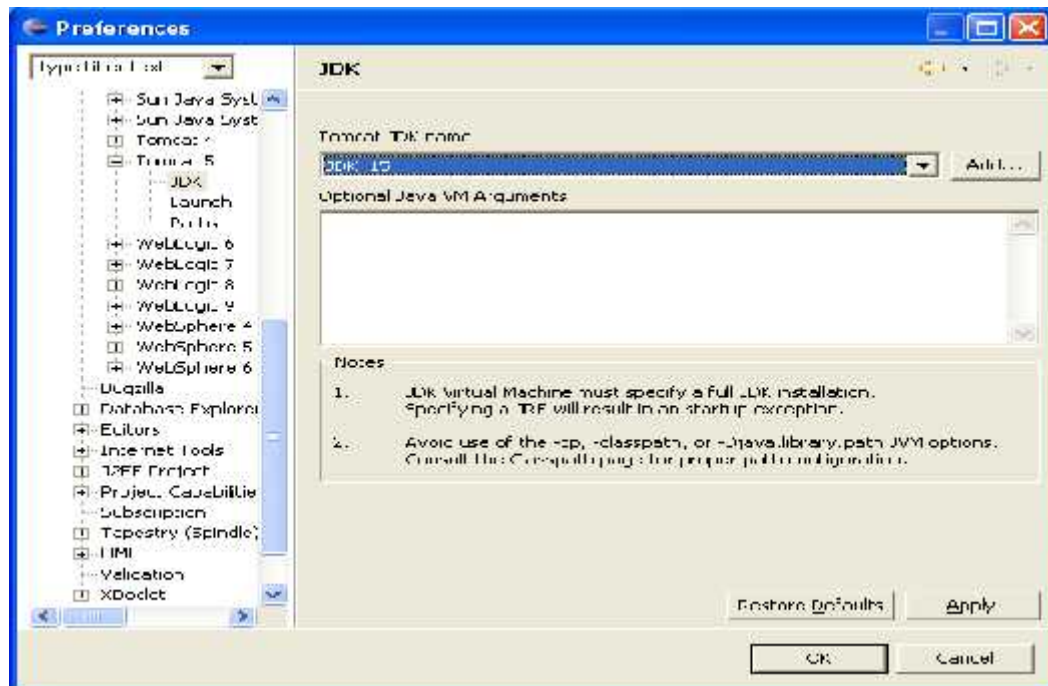
2.- Añadir el Servidor de Aplicaciones Tomcat 5.5

En la barra de menú de Eclipse selecciono Windows → Preferences → MyEclipse → Application Server → Tomcat 5. Se presentará una ventana en el cual se escogerá la ruta en donde se encuentra instalado el Tomcat. En el Browse buscamos la ruta C:\bin\Tomcat 5.5 y lo habilitamos escogiendo la opción Enable. Y presionamos Apply para que aplique los cambios realizados.



3.- Escoger el kit de java con el que trabajará el Servidor de Aplicaciones Tomcat 5.5

En la barra de menú de Eclipse selecciono Windows → Preferences → MyEclipse → Application Server → Tomcat 5 → JDK → Escoger JDK_15 (creado en el paso 1)



Y presionamos OK para aplicar los cambios.

10. Configuración Del Entorno De Desarrollo Integrado Eclipse Con Jasper

Jasper es una Herramienta que nos servirá para realizar reportes estáticos en las páginas JSP que se creen en los proyectos.

Se deben seguir los siguientes pasos:

- 1.- Descomprimir los archivos de feature.rar y copiarlos a la carpeta de features que se encuentra en la ruta de instalación de Eclipse que hayamos escogido: C:\bin\Eclipse\features
- 2.- Descomprimir los archivos de plugins.rar y copiarlos a la carpeta de plugins que se encuentra en la ruta de instalación de Eclipse que hayamos escogido: C:\bin\Eclipse\plugins

11. Configuración Del Entorno De Desarrollo Integrado Eclipse Con Display (Para Mostrar Tablas En Una Página)

Es un archivo tld que es un taglib que se usa cuando una página requiera mostrar datos en una tabla.

1.- Descomprimir las librerías del shared-lib.rar y copiarlas a la carpeta de features que se encuentra en la ruta de instalación de Eclipse: C:\bin\Tomcat 5.5\shared\lib

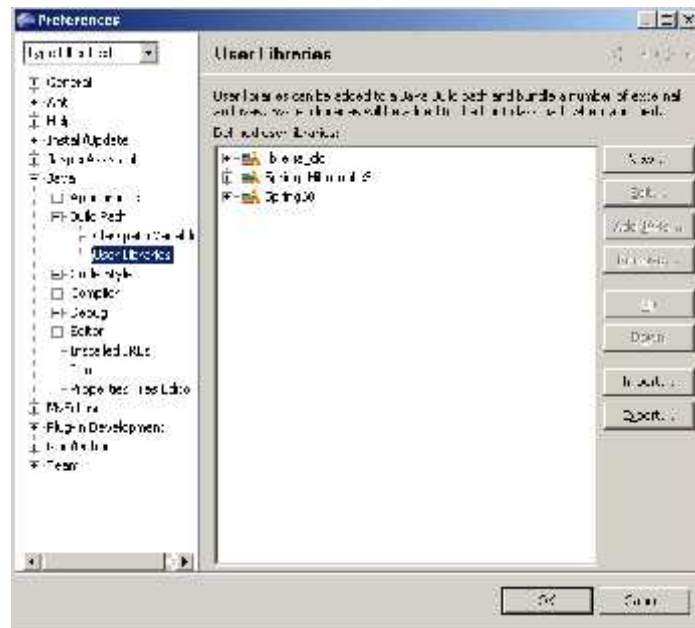
2.- Copiar el archivo displaytag-12.tld en el proyecto que necesite utilizarlo, en la ruta \webroot\WEB-INF\tld\

3.- Debemos de llamar a este archivo dentro del JSP que estemos utilizando para invocar a todos los taglib que usaremos en las distintas páginas. Para nuestro proyecto lo invocamos en el archivo JSP include con la siguiente línea:

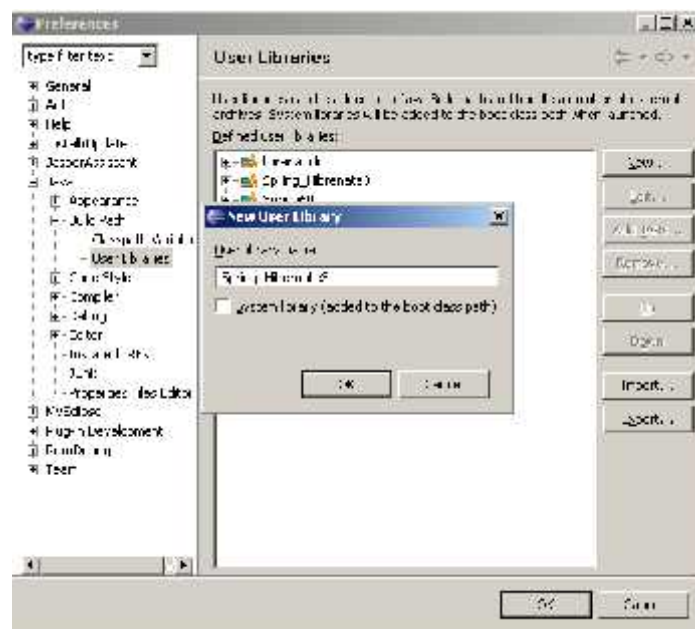
```
<%@ taglib prefix="display" uri="/WEB-INF/tld/displaytag-12.tld" %>
```

12. Configuración Del Entorno De Desarrollo Integrado Eclipse Con Librerías

a. CREACIÓN DE LAS LIBRERIAS DE USUARIOS (Spring_Hibernate3)



En la barra de menú de Eclipse selecciono Windows → Preferences → Java → Build Path → User Libraries. Presiono el botón New para crear una nueva librería, me aparecerá una ventana en la cual le daremos un nombre a la librería que se creará.



Una vez creado el nombre de la nueva librería; luego presionando el botón Add JARs comenzamos a añadir los jar que vamos a necesitar para poder hacer uso de dicha librería; para nuestro proyecto tenemos que añadir los

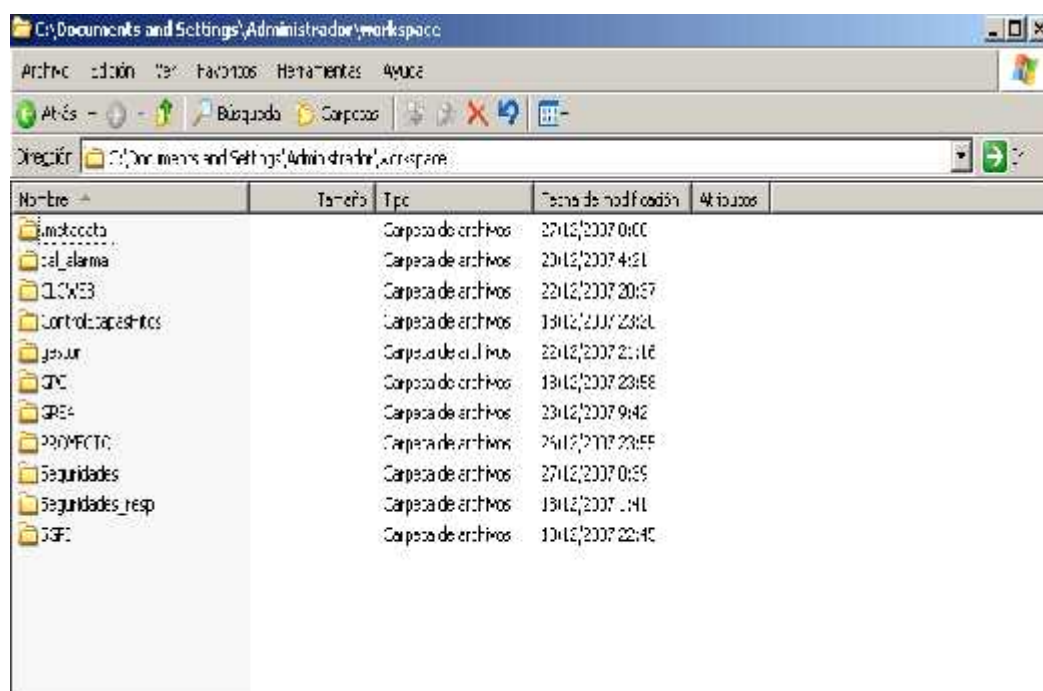
jar del Servidor de Aplicaciones Tomcat 5.5 que se encuentran en las rutas C:\bin\Tomcat 5.5\common\lib y C:\bin\Tomcat 5.5\shared\lib.

Y Presionamos OK para aplicar los cambios.

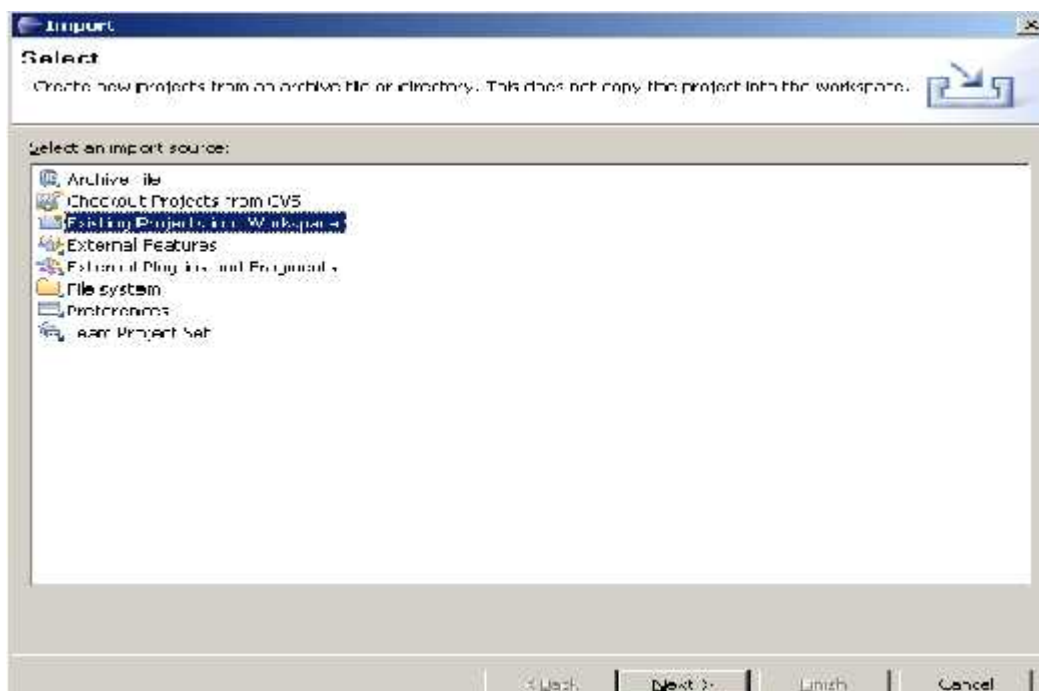


13. Importar, Deployar Y Levantar Proyectos

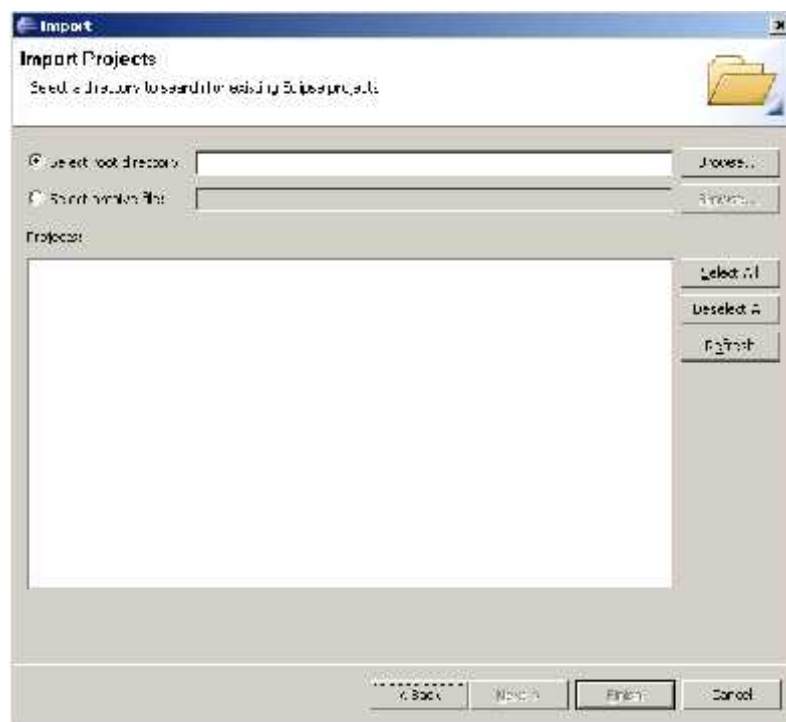
Para poder importar un proyecto es necesario que el proyecto se encuentre en el área de trabajo que Eclipse por default crea que es en el WORKSPACE ubicado en la ruta C:\Documents and Settings\Administrador\workspace.



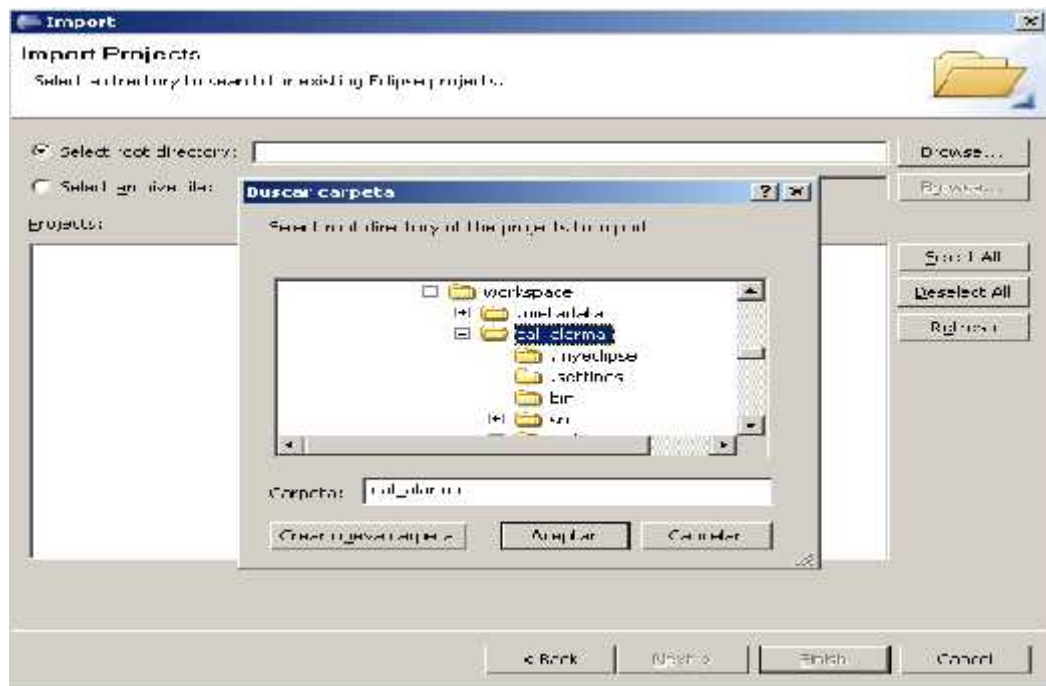
Una vez que el Proyecto se encuentre en el workspace, en la barra de Menú de Eclipse seleccionamos File → Import. Se nos abrirá la siguiente ventana, y escogemos la opción Existing Projects into Workspace, y presionamos Next.



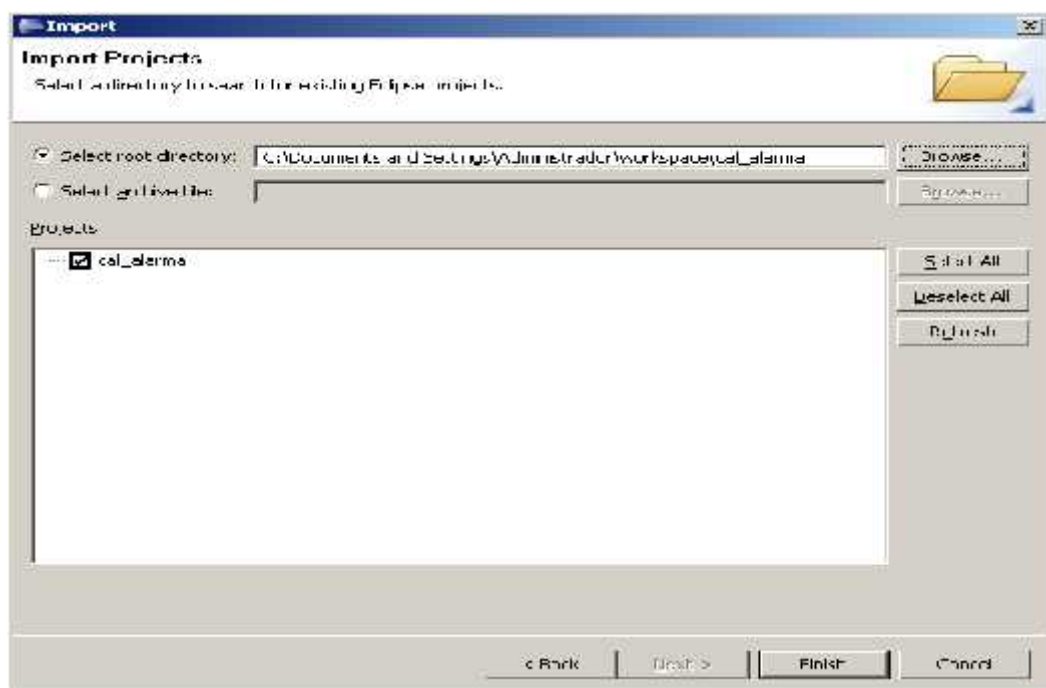
Se nos abrirá la siguiente pantalla, presionando el botón Browse tenemos que seleccionar un directorio en donde esté el Proyecto que vamos a importar; en este caso buscaremos en el workspace.



Una vez marcado y escogido el Proyecto presionamos el botón Aceptar.

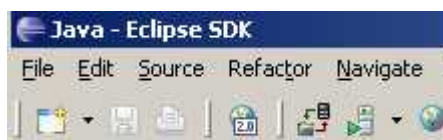


Y por último presionamos el botón Finish para finalizar con la importación.

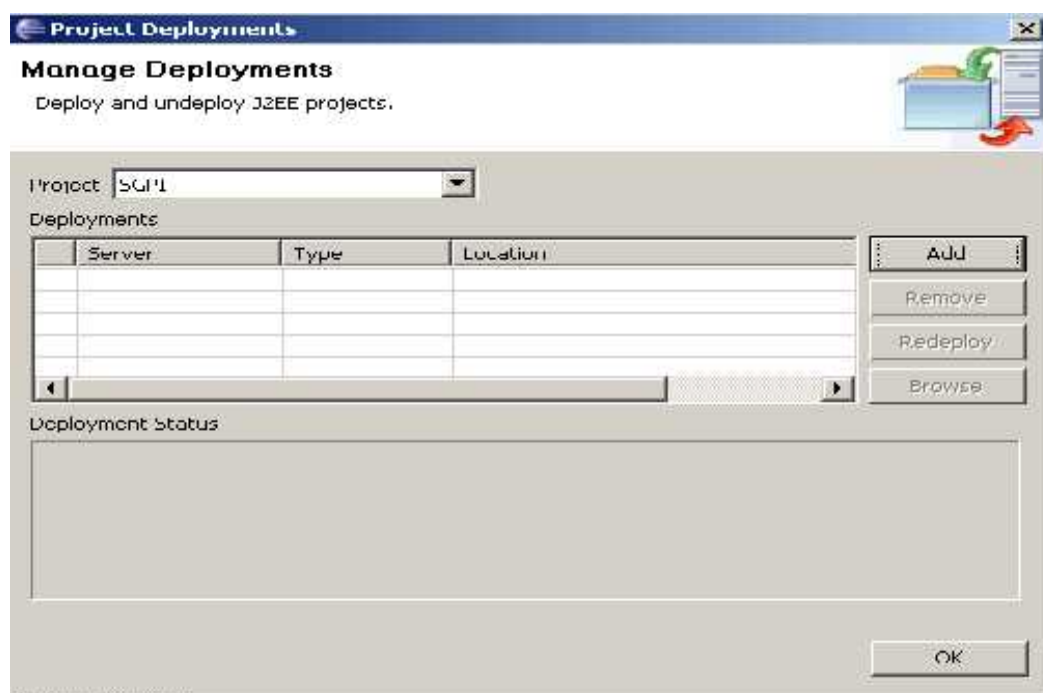


Una vez que ya hemos importado el Proyecto verificamos que estén seteadas todas las librerías que cada proyecto está usando.

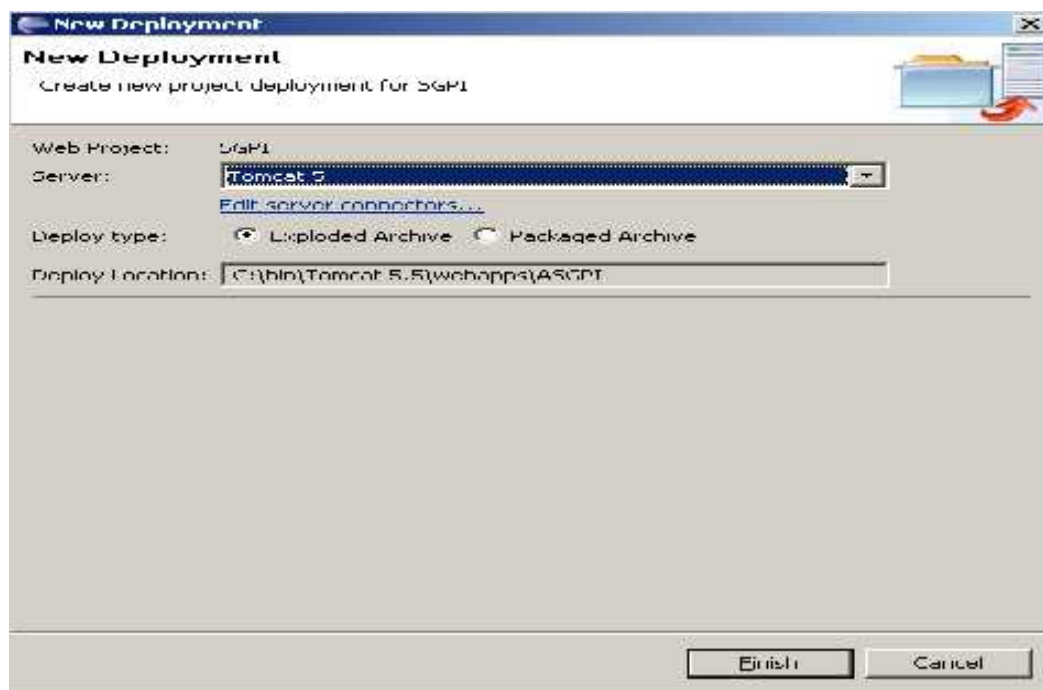
Deployar es levantar el Proyecto en el Servidor de Aplicaciones.



Para deployarlo presionamos el siguiente ícono  ubicado en la barra de herramientas de Eclipse; a continuación se abrirá la siguiente pantalla, en la cual tenemos que añadir el servidor de aplicaciones presionando el botón Add.

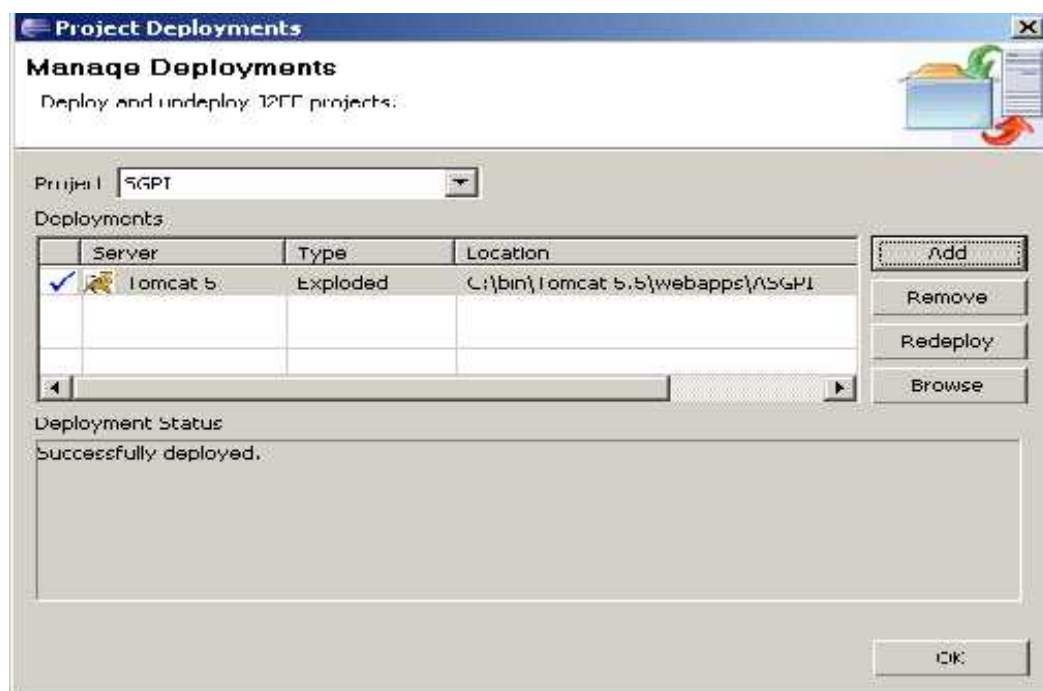


Seleccionamos el Servidor de Aplicaciones Tomcat 5 configurado en Eclipse y presionamos Finish para que deploye el Proyecto.

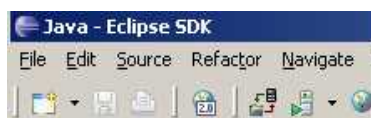


En la ventana principal de Deployar Proyectos ya estará incluido el Proyecto que acabamos de deployar con su respectivo Servidor de Aplicaciones. Presionamos OK para aplicar los cambios.

Nota: Nos damos cuenta que al Deployar el Proyecto se nos guarda en la ruta C:\bin\Tomcat 5.5\webapps.



Una vez ya importado y deployado un proyecto; procedemos a levantar el Servidor de Aplicaciones para verificar si el Proyecto es levantado satisfactoriamente.

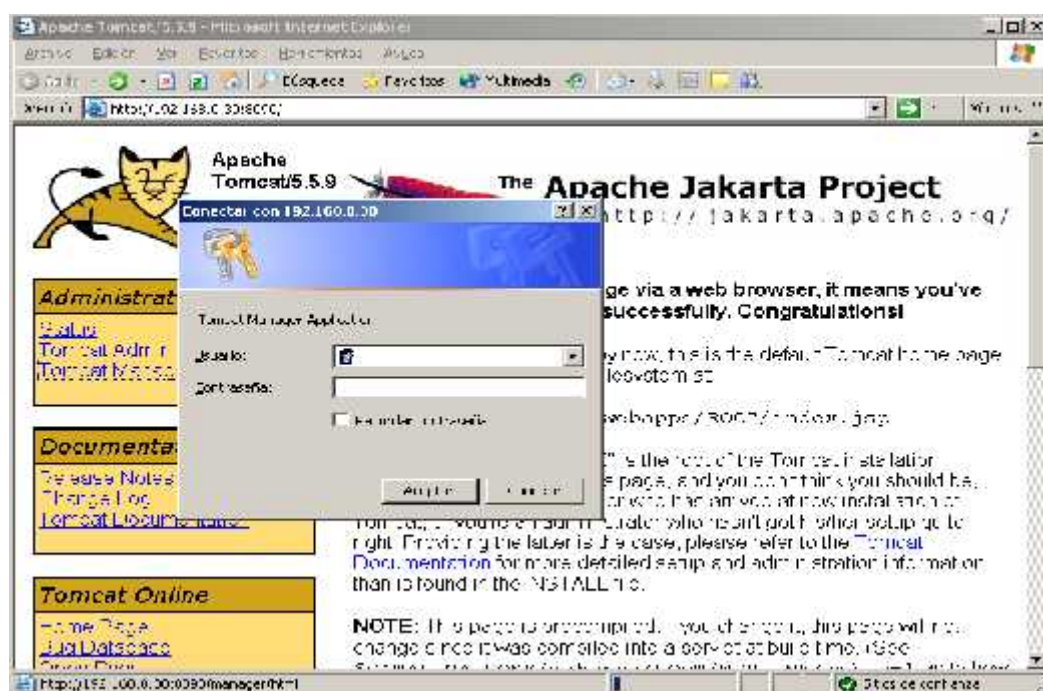


Para Levantar el Servidor de Aplicaciones presionamos el siguiente ícono  ubicado en la barra de herramientas de Eclipse; y en la Consola de Eclipse se displayará todos los procesos que está levantando el Servidor de Aplicaciones y si ocurriese algún error lo mostrará; al final tendrá que aparecer un mensaje de que el servidor está levantado (Server startup).

Para verificar si nuestro Proyecto fue levantado con éxito; debemos abrir el Browser (Explorador) y direccionarlo hacia la página principal del Servidor de Aplicaciones <http://192.168.0.30:8090> que es la IP local y el puerto que escogimos al momento de la instalación para que se levante nuestro Servidor de Aplicaciones.



Damos click en el link [Tomcat Manager](#) y se nos abrirá la siguiente ventana, en la cual nos pedirá usuario y contraseña para poder conectarme al Servidor definidos al momento de la instalación del Servidor: (usuario: admin).



Se levantará la siguiente página en el cual estarán todos los Proyectos que han sido deployados y si se ha levantado satisfactoriamente estará su estado en TRUE caso contrario su estado será FALSE.

Y ahora dándole click en cualquiera de los link de los Proyectos que están en estado true; se levantará la página de inicio de dicho Proyecto.



MANUAL DE USUARIO

INTRODUCCIÓN

Este manual esta dirigido a los capacitadores y/o usuarios finales del Sistema Gestor de Proyectos Informáticos en el área de mantenimiento de los entes principales como Personas, Áreas, Departamentos, Cargos y Provincia.

Se detalla cada uno de los pasos a seguir para navegar entre la información, añadir un registro nuevo, buscar y eliminar, al pie de cada figura se realiza una completa descripción.

Para ir a esta sección de pantallas de mantenimiento desde el menú general se elegirá la opción “Mantenimiento” como detalla la figura 1.

Las opciones de Nuevo, Actualizar, Búsqueda, Navegación y Borrado de un registro son iguales en todos los entes con pequeñas variaciones que se puntualizarán, razón por la cual no haremos mayor énfasis en cada una de ellas ya que en la primera parte del Mantenimiento de Personas explicaremos paso a paso como funciona cada una de las opciones y este pasará a ser la guía de ejemplo y referencia para el mantenimiento de área, cargos, departamento, etc.

1. MANTENIMIENTO DE PERSONAS

Personas almacena toda la información del recurso humano que esta involucrado en un proyecto ya sea un programador, un analista, un administrador de base de datos, un líder de sistema, etc. Cada recurso nuevo contratado para integrarse al staff del área de sistemas informáticos deberá registrarse como un nuevo registro. Generalmente la información necesaria para llenar un nuevo registro se encuentra en la hoja de vida y este puede venir impreso o de manera digital desde el Dpto. de RRHH.

Se detallan cada uno de los campos que forman el registro de una persona.

Persona #.- El código que identifica a la persona

Tipo de Identificación.- Lista los diferentes tipos como cedula, pasaporte, etc.

Nombres.- Se ingresa los nombres de la persona

Apellidos.- Se ingresan los apellidos

Nombres Completos.- Se realiza automáticamente

Fecha de nacimiento.- (dd/mm/yyyy) es el formato en que se presenta la fecha, para llenarlo deberá elegir la fecha con ayuda del icono calendario que se encuentra junto a la caja de texto.

Dirección.- Dirección Domiciliaria

Teléfono.- Teléfono para localizar

Mail.- Mail de contacto

Lugar de trabajo.- Ciudad donde trabaja

Sexo.- Masculino, Femenino

Estado Civil.- Soltero, Casado, Divorciado, etc.

País.- Su país de origen

Provincia.- Provincia donde nació

Ciudad.- El cantón donde nació

Parroquia.- Parroquia donde nació

Cargo.- El cargo que desempeña en la empresa

Profesión.- Título profesional

Departamento.- Departamento asignado dentro de la empresa

Estado.- Si está activo, inactivo(renuncia, despido) y todos los posibles estados de una persona dentro de una empresa.



Figura 1
Menú Principal

La primera opción es el Mantenimiento de la entidad Personas, al dar clic en la opción Personas la primera pantalla a mostrar es la plantilla general de los datos de una persona, y en la parte superior en el menú encontraremos las opciones Nuevo, Guardar, Borrar, Siguiente, Anterior, Buscar y Cerrar, a lo largo del manual iremos revisando cada una de las diversas opciones.

En la figura 2 y 3 (la misma pantalla ha sido dividida en dos partes debido que el desplazamiento del scroll bar no permite ver la imagen de una sola vista) podemos ver la plantilla con todos los datos del ente Persona, donde se muestran los campos en blanco y los combos seteados en la opción por DEFAULT, el primer campo etiquetado como Persona # se encuentra en blanco debido a que no son un registro válido sino que solo muestra el formato de pantalla.

MODULO DE INTEGRACIÓN PARAMETRIZACIÓN Y CONFIGURACION

Nuevo Guardar Borrar Siguierte Anterior Buscar Cerrar

Mantenimiento de Personas

Persona de:

Tipo Relación:

Nombres:

Apellidos:

Nombre Completo:

Fecha de Nacimiento:

Tipo Identificación:

N° Identificación:

Dirección:

Teléfono:

Mail:

Figura 2
Menú Personas – Parte 1

Dirección:

Teléfono:

Mail:

Lugar de Trabajo:

Sexo:

Estado Civil:

Etnia:

Provincia:

Ciudad:

Parroquia:

Cargos:

Profesión:

Departamento:

Especialidad:

Figura 3
Menú Personas – Parte 2

1.1. Navegación Siguiente - Anterior

El botón siguiente y anterior del menú nos permite navegar entre los registros existentes en la base de datos, estando en la pantalla principal (figura 2 y 3), presionamos el botón siguiente y podemos visualizar el primer registro (Ver Figura 4), ahora si podemos ver en el campo Persona # el número 1, representando el número de

The screenshot shows a web application interface with a dark blue header bar containing the text 'MODULO DE INTEGRACIÓN PARAMETRIZACIÓN Y CONFIGURACION'. Below the header is a navigation menu with buttons: 'Nuevo', 'Guardar', 'Borrar', 'Siguiente', 'Anterior', 'Buscar', and 'Cancelar'. The main content area has a light green background and is titled 'Mantenimiento de Personas'. It contains a form for editing a person's record, labeled 'Persona # 1'. The form fields are as follows:

Tipo Relación:	EMPLEADO
Tipo Identificación:	CEDULA
A. Identificación:	0001213245
Nombres:	MARTIN LUIS
Apellidos:	CASTELLANO
Nombres Completos:	MARTIN LUIS CASTELLANO
Fecha de Nacimiento:	1985-05-22
Dirección:	CALLE 100 N. 42A # 100-100
Teléfono:	041628015

Figura 4
Personas – Navegación Siguiente

registro, cada vez que seleccionemos la opción "Siguiente" se visualizará el campo siguiente ordenados según el campo Persona #, la navegación terminará cuando pulsando la opción "Siguiente" se llegue hasta el último registro del banco de datos Personas. Si el banco de datos contiene a 12 personas registradas podrá pulsar la opción "Siguiente" hasta que el campo Persona # tenga como valor el 12. Si en la pantalla inicial (Figura 2 y 3) al presionar "Siguiente" no aparece el registro Persona # 1, indica que no existen registros en el ente Persona.

La opción "Anterior" funciona de manera similar, cada vez que pulse esta opción se visualizará el registro anterior, (Ver figura 5), si estamos en el registro Persona # 3 y pulsamos la opción "Anterior", regresaremos al registro previo a este y enseguida visualizaremos el registro Persona # 2. La opción "Anterior" llegará a su fin de navegación al encontrar al registro Persona #1. Deberán existir al menos 2 registros para que la opción "Anterior" tenga funcionalidad y se pueda observar la navegabilidad.

MODULO DE INTEGRACIÓN PARAMETRIZACIÓN Y CONFIGURACION

Nuevo Guardar Borrar Siguiente Anterior Buscar Cerrar

Mantenimiento de Personas

Persona # 3

Tipo Relación:

Tipo Identificación:

* Identificación:

MODULO DE INTEGRACIÓN PARAMETRIZACIÓN Y CONFIGURACION

Nuevo Guardar Borrar Siguiente Anterior Buscar Cerrar

Mantenimiento de Personas

Persona # 2

Tipo Relación:

Tipo Identificación:

* Identificación:

Nombre:

Apellidos:

Nombre Completo:

Fecha de Nacimiento:

dd/mm/yyyy

Figura 5
Persona- Navegación anterior

1.2 Buscar Personas

Al elegir la opción “Buscar” la primera pantalla que podrá visualizar una tabla que lista la personas almacenadas, identificadas por su código y sus nombres completos ordenadas alfabéticamente.

En la parte inferior de la lista encontramos dos opciones de anterior y siguiente que ofrece la navegabilidad entre los registros, cada tabla contiene un máximo de 10 registros si existiesen mas, el botón siguiente estará en negrilla indicando que existe una siguiente página, si damos clic podremos observar una segunda tabla y se activará y se pondrá en negrilla la opción “Anterior” (Ver figura 6).

Se puede buscar de 2 formas la primera es navegando con los botones “siguiente” y “anterior” hasta encontrar a la persona que buscamos, si es así damos clic sobre su código y observamos en una pantalla siguiente toda la información de esa persona. Para nuestro ejemplo damos clic sobre el código 13 y se mostrará por pantalla toda la información de la persona Verónica Moreno. (Ver figura 8)



Figura 6
Persona- Listado de nombres

La segunda forma de búsqueda se maneja un solo criterio (forma) de búsqueda, en este caso se buscará mediante el nombre o apellido, en la caja de texto se debe ingresar el nombre o el apellido o parte de ellos y presionamos en el botón “Buscar”, para nuestro ejemplo escribimos en la caja de texto “VER” y listará todas aquellas coincidencias. (Ver Figura 7).

Figura 7
Personas – Listado Búsqueda según Criterio

Figura 8
Personas – Resultado de la Búsqueda

1.3 Agregar Nuevo Registro Persona

Para agregar un nuevo registro en el menú principal elegimos la opción Nuevo, entonces se visualizará la plantilla de datos Persona con los campos en blanco listos para ser llenados, cuando se hayan terminado de ingresar los datos en el menú principal se elige la opción guardar, aparecerá un cuadro de dialogo confirmando la transacción. Si desea guardar los datos elija la opción “Aceptar”, caso contrario elija “Cancelar”. (Ver Figura 9)

MODULO DE INTEGRACIÓN PARAMETRIZACIÓN Y CONFIGURACION

Nuevo Guardar Borrar Siguiente Anterior Buscar Borrar

Mantenimiento de Personas

Persona #:

Tipo Relación:

Tipo Identificación:

Identificación:

Nombres:

Apellidos:

Nombre Completo:

Fecha de Nacimiento:

Dirección:

Teléfono:

Microsoft Internet Explorer

?

Aceptar Cancelar

Figura 9
Persona - Guardar nuevo registro

1.4 Modificar Registro Persona

Para modificar un registro primero se lo debe buscar por cualquiera de las formas antes explicadas y una vez que se ha situado sobre el registro se modifican los campos necesarios y se selecciona la opción Guardar del menú, a continuación aparecerá un cuadro de dialogo confirmando la transacción. Si desea guardar los datos elija la opción "Aceptar", caso contrario elija "Cancelar".

1.5. Borrar Registro Persona

Para borrar un registro primero se lo debe buscar por cualquiera de las formas antes explicadas y una vez que se ha situado sobre el registro se debe elegir la opción Borrar del menú, a continuación aparecerá un cuadro de dialogo confirmando su transacción, si esta seguro de borrar el registro presione "Aceptar" sino elija "Cancelar", tener en cuenta que una vez borrado el registro no hay forma de recuperarlo. (Ver Figura 10).

MODULO DE INTEGRACIÓN, PARAMETRIZACIÓN Y CONFIGURACIÓN

Nuevo Guardar Borrar Siguiente Anterior Buscar Borrar

Mantenimiento de Personas

Persona #: 20

Tipo Relación: [SELECTED]

Tipo Identificación: [SELECTED]

Identificación: [?]

Nombre: [?]

Apellidos: [ALFARO SALVATIERRA]

Nombre Completo: [ALFARO SALVATIERRA]

Fecha de Nacimiento: [9/3/12/00]

Dirección: [Calle Los Angeles No. 45 y 12]

Teléfono: [011-12345678]

¿Proceder a borrar?

Sí No

Figura 10
Persona Borrar registro

2. MANTENIMIENTO DE CARGOS

En Cargos encontramos los diversos cargos establecidos dentro del área de sistemas informáticos tales como: analista de sistemas, diseñador Web, líder de sistemas, etc.

Se detalla cada uno de los campos que forman la entidad Cargos:

Cargo #.- El código del cargo

Nombre.- El nombre del cargo

En la pantalla del menú principal (Ver Figura 1), elegimos la opción Cargos y la primera pantalla que se presenta tiene los registros en blanco (Ver Figura 11) ya que muestra la plantilla de los campos, para comenzar a navegar entre los registros usamos las flechas anterior y siguiente tal como se hizo en el Mantenimiento de Personas.



Figura 11
Menú Cargos

2.1 Buscar Cargos

Para buscar cargos, seleccionamos la opción Buscar del menú Cargos y se visualizará la pantalla con una tabla que lista todos los cargos establecidos dentro de la empresa. (Ver figura 12)



Figura 12
Cargos - Listado de cargos

En la parte inferior se encuentran botones de anterior y siguiente con los cuales se pueden navegar a través de las tablas, cada una de las tablas tiene un máximo de 10 registros, para acceder a la información completa se

debe seleccionar el código del cargo deseado por ejemplo si se desea visualizar la información del Analista de sistemas se da clic sobre el código 7.

Para una búsqueda con criterio en la caja de texto se ingresa el nombre del cargo o palabras descriptivas de este. En nuestro ejemplo en la caja de texto ingresamos la palabra “DESA” y damos clic sobre el botón Buscar y se listará los cargos que tengan relación con la descripción. (Ver figura 13).

Para un mayor detalle del paso a paso revisar el párrafo de búsqueda de Mantenimiento de Personas.

Codigo	Nombre
9	DESARROLLADOR DE INTERFAZ WEB
11	DESARROLLADOR JUNIOR
10	DESARROLLADOR SENIOR

Figura 13
Cargos – Búsqueda según criterio

2.2 Agregar Nuevo Registro Cargo

Para agregar un nuevo registro en el menú principal elegimos la opción Nuevo, entonces se visualizará la plantilla de datos Cargos con el campo en blanco listo para ser llenado, cuando termine de ingresar los datos en el menú principal se elige la opción guardar, aparecerá un cuadro de dialogo confirmando la transacción. Si desea guardar los datos elija la opción “Aceptar”, caso contrario elija “Cancelar”.

Para un mayor detalle del paso a paso revisar el párrafo de Agregar Nuevo Registro de Mantenimiento de Personas.

2.3 Modificar Registro Cargos

Para modificar un registro primero se lo debe buscar por cualquiera de las formas antes explicadas y una vez que se ha situado sobre el registro se modifican los campos necesarios y se selecciona la opción Guardar del menú, a continuación aparecerá un cuadro de dialogo confirmando la transacción. Si desea guardar los datos elija la opción “Aceptar”, caso contrario elija “Cancelar”.

2.4. Borrar Registro Cargos

Para borrar un registro primero se lo debe buscar por cualquiera de las formas antes explicadas y una vez que se ha situado sobre el registro se debe elegir la opción Borrar del menú, a continuación aparecerá un cuadro de dialogo confirmando su transacción, si esta seguro de borrar el registro presione “Aceptar” sino elija “Cancelar”, tener en cuenta que una vez borrado el registro no hay forma de recuperarlo.

3. MANTENIMIENTO DE AREAS

Aquí se encuentra cada una de las áreas de la empresa como marketing, ventas, financiera, etc.

Se detallan cada uno de los campos;

Área #.- El código de área

Nombre.- El nombre que identifica al área

Activo.- El estado del área

Ubicación.- Ubicación dentro de las instalaciones como palta baja, mezanine, cuarto piso, etc.

Representante.- Nombre del jefe o representante del área

The screenshot shows a web application interface. At the top, a dark blue header bar contains the text 'MODULO DE INTEGRACIÓN PARAMETRIZACIÓN Y CONFIGURACION' in white. Below this, a lighter blue bar contains navigation buttons: 'Nuevo', 'Guardar', 'Borrar', 'Siguiente', 'Anterior', and 'Cerrar'. The main content area has a light green background and is titled 'Mantenimiento de Areas' in blue. Below the title, there are five labels with corresponding input fields: 'Area #', 'Nombre', 'Activo', 'Ubicación', and 'Representante'. Each label is followed by a white rectangular input field.

Figura 14
Menú Áreas

En la pantalla del menú principal (Ver Figura 1), elegimos la opción Áreas y la primera pantalla que se presenta tiene los registros en blanco (Ver Figura 14) ya que muestra la plantilla de los campos, para comenzar a navegar entre los registros usamos las flechas anterior y siguiente tal como se hizo en el Mantenimiento de Personas.

3.1 Agregar Nuevo Registro Área

Para agregar un nuevo registro en el menú principal elegimos la opción Nuevo, entonces se visualizará la plantilla de datos Área con el campo en blanco listo para ser llenado, cuando termine de ingresar los datos en el menú principal se elige la opción guardar, aparecerá un cuadro de dialogo confirmando la transacción. Si desea guardar los datos elija la opción "Aceptar", caso contrario elija "Cancelar". (Ver Figura 15)



Figura 15
Áreas – Guardar Registro

Para un mayor detalle del paso a paso revisar el párrafo de Agregar Nuevo Registro de Mantenimiento de Personas.

3.2. Modificar Registro Área

Para modificar un registro primero se lo debe buscar navegando con las opciones anterior y siguiente del menú principal de áreas, ya que este ente no tiene opción buscar, una vez situado en el registro se modifican los campos necesarios y se selecciona la opción Guardar del menú, a continuación aparecerá un cuadro de dialogo confirmando la transacción. Si desea guardar los datos elija la opción "Aceptar", caso contrario elija "Cancelar".

3.3. Borrar Registro Área

Para borrar un registro primero se lo debe buscar navegando con las opciones anterior y siguiente del menú principal de áreas, ya que este ente no tiene opción buscar, una vez situado sobre el registro se debe elegir la opción Borrar del menú, a continuación aparecerá un cuadro de dialogo confirmando su transacción, si esta seguro de borrar el registro presione "Aceptar" sino elija "Cancelar", tener en cuenta que una vez borrado el registro no hay forma de recuperarlo.

4. MANTENIMIENTO DE PROVINCIAS

En Provincia encontramos todas las provincias del país Ecuador.

Se detalla cada uno de los campos que forman la entidad Provincia:

Provincia #.- El código de la provincia.

País.-Se elije de la lista el país, por default se carga Ecuador.

Nombre.- El nombre de la provincia.

Sufijo.- Se almacena el sufijo telefónico

En la pantalla del menú principal (Ver Figura 1), elegimos la opción Provincias y la primera pantalla que se presenta tiene los registros en blanco (Ver Figura 16) ya que muestra la plantilla de los campos, para comenzar a navegar entre los registros usamos las flechas anterior y siguiente tal como se hizo en el Mantenimiento de Personas.

Figura 16
Menú Provincias

4.1 Agregar Nuevo Registro Provincia

Para agregar un nuevo registro en el menú principal elegimos la opción Nuevo, entonces se visualizará la plantilla de datos Provincia con los campos en blanco listo para ser llenado, cuando termine de ingresar los datos en el menú principal se elije la opción guardar, aparecerá un cuadro de dialogo confirmando la transacción. Si desea guardar los datos elija la opción "Aceptar", caso contrario elija "Cancelar". (Ver Figura 17)

Para nuestro ejemplo se ingresa la Provincia de Santa Elena se elije como país a Ecuador y le ponemos como sufijo telefónico 07.

The image displays two screenshots of a web application interface for 'Mantenimiento de Provincias' (Province Maintenance). The interface is titled 'MODULO DE INTEGRACIÓN PARAMETRIZACIÓN Y CONFIGURACIÓN' and includes navigation buttons: 'Nuevo', 'Guardar', 'Borrar', 'Siguiente', 'Anterior', and 'Cerrar'.

The top screenshot shows the form with the following fields:

- Provincia: [dropdown menu]
- País: [dropdown menu]
- Nombre: [text input field]
- Suñijo: [text input field]

The bottom screenshot shows the same form, but with a 'Microsoft Internet Explorer' dialog box open. The dialog box contains a question mark icon and the text '¿Está seguro de que desea guardar los datos?' (Are you sure you want to save the data?). It has two buttons: 'Aceptar' (Accept) and 'Cancelar' (Cancel).

Figura 17
Menú Provincias

Para un mayor detalle del paso a paso revisar el párrafo de Agregar Nuevo Registro de Mantenimiento de Personas.

4.2. Modificar Registro Provincia

Para modificar un registro primero se lo debe buscar navegando con las opciones anterior y siguiente del menú principal de provincias, ya que este ente no tiene opción buscar, una vez situado en el registro se modifican los campos necesarios y se selecciona la opción Guardar del menú, a continuación aparecerá un cuadro de dialogo confirmando la transacción. Si desea guardar los datos elija la opción "Aceptar", caso contrario elija "Cancelar".

4.3. Borrar Registro Provincia

Para borrar un registro primero se lo debe buscar navegando con las opciones anterior y siguiente del menú principal de provincias, ya que este ente no tiene opción buscar, una vez situado sobre el registro se debe elegir la opción Borrar del menú, a continuación aparecerá un cuadro de dialogo confirmando su transacción, si esta

seguro de borrar el registro presione “Aceptar” sino elija “Cancelar”, tener en cuenta que una vez borrado el registro no hay forma de recuperarlo.

5. MANTENIMIENTO DE ESTADOS

Es este almacén de datos se encuentran todos los estados de un proyecto como aprobado, activo, finalizado, en espera, etc. En el transcurrir del desarrollo de proyectos seguramente se crearán nuevos estados o se deseará consultar la descripción de alguno específicamente.

Se detalla cada uno de los campos que forman la entidad Estados:

Estado #.- El código del estado, en este caso particular el código no es automático sino que es una cadena de caracteres que identifique de manera clara el estado, por ejemplo Estado #. A Descripción: Aprobado

Descripción.- Descripción del estado

En la pantalla del menú principal (Ver Figura 1), elegimos la opción Estados y la primera pantalla que se presenta tiene los registros en blanco (Ver Figura 18) ya que muestra la plantilla de los campos, para comenzar a navegar entre los registros usamos las flechas anterior y siguiente tal como se hizo en el Mantenimiento de Personas.

Figura 18
Menú Estados

2.1 Buscar Estados

Para buscar estados, seleccionamos la opción Buscar del menú Estados y se visualizará la pantalla con una tabla que lista todos los estados. (Ver figura 19)



Figura 19
Estados Listado de estados existentes.

En la parte inferior se encuentran botones de anterior y siguiente con los cuales se pueden navegar a través de las tablas, cada una de las tablas tiene un máximo de 10 registros, para acceder a la información completa se debe seleccionar el código del estado. Para una búsqueda con criterio en la caja de texto se ingresa el nombre del cargo o palabras descriptivas de este. Para un mayor detalle del paso a paso revisar el párrafo de búsqueda de Mantenimiento de Personas.

5.2 Agregar Nuevo Registro Estado.

Para agregar un nuevo registro en el menú principal elegimos la opción Nuevo, para nuestro ejemplo ingresamos en Estado #, un código descriptivo como PRB y en Descripción A Prueba para describir un estado de un proyecto que esta a prueba.



Figura 20
Estados – Nuevo Registro

Entonces se visualizará la plantilla de datos de Estado con los campos en blanco listos para ser llenado, cuando termine de ingresar los datos en el menú principal se elige la opción guardar, aparecerá un cuadro de dialogo confirmando la transacción. Si desea guardar los datos elija la opción "Aceptar", caso contrario elija "Cancelar". (Ver Figura 20).

5.3 Modificar Registro Estado.

Para modificar un registro primero se lo debe buscar por cualquiera de las formas antes explicadas ya sea navegando en la tabla del listado o haciendo la búsqueda con criterio y una vez que se ha situado sobre el registro se modifican los campos necesarios y se selecciona la opción Guardar del menú, a continuación aparecerá un cuadro de dialogo confirmando la transacción. Si desea guardar los datos elija la opción "Aceptar", caso contrario elija "Cancelar".

5.4. Borrar Registro Estado.

Para borrar un registro primero se lo debe buscar por cualquiera de las formas antes explicadas ya sea navegando en la tabla del listado o haciendo la búsqueda con criterio y una vez que se ha situado sobre el registro se debe elegir la opción Borrar del menú, a continuación aparecerá un cuadro de dialogo confirmando su transacción, si esta seguro de borrar el registro presione "Aceptar" sino elija "Cancelar", tener en cuenta que una vez borrado el registro no hay forma de recuperarlo.