

PRÁCTICA EMPRESARIAL EN VISION LTDA.

CODIFICACIÓN, REALIZACIÓN DE PRUEBAS Y DOCUMENTACIÓN DE PROGRAMAS PARA EL REGISTRO DE INFORMACIÓN Y GENERACIÓN DE REPORTES A TRAVÉS DE LA METODOLOGÍA EXTREME PROGRAMMING CON EL FIN DE CONSOLIDAR EL SISTEMA DE GESTIÓN DE CALIDAD DE VISIÓN

RICARDO ARMANDO PABÓN PINILLA
Código: 2000681

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2006

PRÁCTICA EMPRESARIAL EN VISION LTDA.

CODIFICACIÓN, REALIZACIÓN DE PRUEBAS Y DOCUMENTACIÓN DE PROGRAMAS PARA EL REGISTRO DE INFORMACIÓN Y GENERACIÓN DE REPORTES A TRAVÉS DE LA METODOLOGÍA EXTREME PROGRAMMING CON EL FIN DE CONSOLIDAR EL SISTEMA DE GESTIÓN DE CALIDAD DE VISIÓN

RICARDO ARMANDO PABÓN PINILLA
Código: 2000681

**Trabajo de grado para optar al título de
Ingeniero de Sistemas**

Tutor Trabajo de Grado
Ing. GERARDO ARMANDO GARAVITO FLÓREZ
Representante Legal Visión Ltda.

Director
Mg. JOSÉ CÁRCAMO SEPÚLVEDA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2006

DIRECTOR PROYECTO DE GRADO (TUTOR)

GERARDO ARMANDO GARAVITO FLÓREZ
Ingeniero de Sistemas
Representante Legal Visión Ltda.

AUTOR

RICARDO ARMANDO PABÓN PINILLA
Estudiante de Ingeniería de Sistemas e Informática
C.C. 91528186 de Bucaramanga
Código 2000681

*A Jehová Dios, mis padres,
hermanas y abuela
quienes han sido mi
soporte espiritual, anímico
y material durante toda mi
vida.*

AGRADECIMIENTOS

El autor del presente proyecto expresa sus agradecimientos a:

Ingeniero Gerardo Armando Garavito Flórez, Representante Legal de Visión Ltda. quien fue gestor y tutor del proyecto y brindó su apoyo constante y valiosas orientaciones en el desarrollo del mismo.

Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander porque sus políticas abrieron la posibilidad de desarrollar este proyecto y su formación fue fundamental para mi desempeño en la práctica empresarial.

El equipo de trabajo de Visión Ltda. por su colaboración y experiencia que fueron soporte para el trabajo dentro de la empresa.

TABLA DE CONTENIDO

	Pág.
LISTA DE FIGURAS	IV
LISTA DE ANEXOS	VI
GLOSARIO	VII
RESUMEN	X
INTRODUCCION	1
1 DESCRIPCIÓN DEL PROYECTO	3
1.1 OBJETIVOS	3
1.1.1 Objetivo General	3
1.1.2 Objetivos Específicos	3
1.2 JUSTIFICACIÓN	4
1.2.1 Definición del Problema	4
1.2.2 Descripción de los Objetivos	4
1.2.3 Impacto y Viabilidad	5
1.3 PLAN DE TRABAJO	6
1.3.1 Fases del Modelo de Iteración Controlada	6
2 MARCO TEÓRICO	9
2.1 SISTEMA DE GESTIÓN DE CALIDAD	9
2.1.1 Calidad	9
2.1.2 Normas de Calidad ISO 9000	10
2.1.3 Norma Técnica Colombiana NTC-ISO 9000:2000	15
2.2 EXTREME PROGRAMMING (XP)	18
2.2.1 Etapas del Proceso de Desarrollo	19
2.2.2 Prácticas XP	20
2.3 BASES DE DATOS	24
2.3.1 Oracle	25
2.4 ORACLE 9I DEVELOPER SUITE	25

2.4.1	Oracle Designer	26
2.4.2	Forms Builder.....	28
2.4.2.1	Elementos básicos de los formularios.....	29
2.4.2.2	Asistente para la construcción de bloques.....	30
2.4.2.3	Asistente para la definición de los elementos del formulario	35
2.4.2.4	Crear un bloque de detalle.....	38
2.4.2.5	Crear un bloque de control e ítems no asociados a una tabla	40
2.4.2.6	Crear listas dinámicas.....	43
2.4.2.7	Crear mensajes de alerta.....	50
2.4.2.8	Triggers.....	52
2.4.3	Reports Builder	54
2.4.3.1	Elaboración de reportes.....	55
2.5	FILOSOFÍA DE TRABAJO DE LA EMPRESA.....	63
2.5.1	Política de Calidad	63
2.5.2	Misión.....	63
2.5.3	Visión	64
3	PRÁCTICA EMPRESARIAL EN VISIÓN LTDA.	65
3.1	INICIACIÓN	65
3.1.1	Inducción a la Filosofía de Trabajo de la Empresa	65
3.1.2	Capacitación en Designer, Forms y Reports.....	65
3.1.2.1	Designer	65
3.1.2.2	Forms.....	66
3.1.2.3	Reports	66
3.1.3	Revisión del Modelo GCI	67
3.1.3.1	Definición del modelo de datos.....	68
3.1.4	Definición de las Opciones a Ajustar y Elaborar	70
3.1.5	Evaluación de las Herramientas Oracle Developer 6i y Oracle Developer 9i.....	72
3.1.5.1	Oracle Developer 6i	72

3.1.5.2	Oracle Developer 9i	72
3.1.5.3	Conclusiones de la evaluación.....	74
3.2	ELABORACIÓN.....	74
3.2.1	Diseño Detallado de las Opciones	74
3.2.2	Realización del Plan de Pruebas	82
3.2.2.1	Casos de prueba.....	83
3.2.3	Elaboración de Reportes.....	84
3.3	CONSTRUCCIÓN	85
3.3.1	Codificación de Opciones	85
3.3.2	Validación Unitaria de las Opciones.....	86
3.3.3	Ajuste de las Opciones	87
3.3.4	Pruebas Funcionales y de Operatividad	87
3.3.5	Ajustes Finales.....	88
3.4	TRANSICIÓN	88
3.4.1	Implantación de las Opciones	88
3.4.2	Ajustes	89
3.4.3	Documentación	89
3.4.4	Entrega Final de las Opciones y su Documentación.....	90
	CONCLUSIONES	91
	BIBLIOGRAFÍA.....	94
	ANEXOS.....	96

LISTA DE FIGURAS

	Pág.
FIGURA 1. OBJECT NAVIGATOR	27
FIGURA 2. OBJECT NAVIGATOR	29
FIGURA 3. VENTANA DE CREACIÓN DE UN BLOQUE DE DATOS	31
FIGURA 4. TIPO DE ORIGEN DE LOS DATOS.....	32
FIGURA 5. ELEMENTOS DEL BLOQUE	32
FIGURA 6. CONEXIÓN A LA BASE DE DATOS.....	33
FIGURA 7. SELECCIÓN DE LA TABLA O VISTA ASOCIADA AL BLOQUE	33
FIGURA 8. INDICANDO LOS ELEMENTOS DEL BLOQUE	34
FIGURA 9. FIN DEL ASISTENTE DE BLOQUES.....	35
FIGURA 10. DEFINICIÓN DEL CANVAS.....	36
FIGURA 11. SELECCIÓN DE CAMPOS DEL BLOQUE A SER MOSTRADOS....	36
FIGURA 12. DEFINICIÓN DE PROPIEDADES DE LOS ITEMS.....	37
FIGURA 13. FORMULARIO GENERADO	38
FIGURA 14. DEFINIENDO LA RELACIÓN CON EL BLOQUE MAESTRO.....	39
FIGURA 15. RELACIONES ENTRE BLOQUES.....	39
FIGURA 16. FORMULARIO CON BLOQUES MAESTRO Y DETALLE	40
FIGURA 17. MOSTRAR LA VENTANA DE PROPIEDADES.....	41
FIGURA 18. VENTANA DE PROPIEDADES.....	42
FIGURA 19. VENTANA DE EJEMPLO DE USO DE UN BLOQUE DE CONTROL	42
FIGURA 20. CONSULTA PARA LA LISTA DE VALORES.....	44
FIGURA 21. COLUMNAS DE LA LISTA DE VALORES	45
FIGURA 22. PROPIEDADES DE LAS COLUMNAS DE LA LISTA DE VALORES	45
FIGURA 23. ELEMENTO DEL FORMULARIO AL QUE SE RETORNARÁ EL VALOR.....	46
FIGURA 24. ELEMENTOS QUE DISPARARÁN LA LISTA DE VALORES	47

FIGURA 25. LISTA DE VALORES EN LA APLICACIÓN.....	47
FIGURA 26. LISTA DE TRIGGERS.....	48
FIGURA 27. PROGRAMACIÓN DEL EVENTO	49
FIGURA 28. VENTANA DE INICIO DE “ORACLE 9I REPORTS DEVELOPER”...55	
FIGURA 29. TIPO DE REPORTE.....	56
FIGURA 30. ESTILO DEL REPORTE	57
FIGURA 31. QUERY EN EL ASISTENTE	58
FIGURA 32. DEFINICIÓN DE NIVELES DE AGRUPAMIENTO.....	58
FIGURA 33. CAMPOS DE LA CONSULTA QUE SE MOSTRARÁN.....	59
FIGURA 34. SELECCIÓN DE CAMPOS PARA CÁLCULO EN EL REPORTE	60
FIGURA 35. PESTAÑA DE LABELS	61
FIGURA 36. PLANTILLA DEL REPORTE	61
FIGURA 37. REPORTE GENERADO CON UNA PLANTILLA PREDEFINIDA	62
FIGURA 38. FORMULARIO DE PASO DE PARÁMETROS POR PARTE DEL USUARIO	62
FIGURA 39. FORMULARIO DE DATOS DEL PRODUCTO.....	75
FIGURA 40. FORMULARIO DE DATOS DE LA FACILIDAD	76
FIGURA 41. FORMULARIO DE DATOS DE LA HERRAMIENTA.....	76
FIGURA 42. FORMULARIO DE DATOS DE LA OPCIÓN.....	77
FIGURA 43. FORMULARIO DE DATOS DEL PROYECTO	78
FIGURA 44. FORMULARIO DE DATOS DE LOS INVOLUCRADOS EN EL PROYECTO.....	78
FIGURA 45. FORMULARIO DE DATOS DEL SEGUIMIENTO DEL PROYECTO	79
FIGURA 46. FORMULARIO DE DATOS DE LA SOLICITUD.....	80
FIGURA 47. VENTANA DE SEGUIMIENTO DE ATENCIÓN DEL REQUERIMIENTO.....	81
FIGURA 48. VENTANA DE LECCIONES RELACIONADAS.....	81
FIGURA 49. FORMULARIO DE DATOS DE LAS LECCIONES.....	81

LISTA DE ANEXOS

- Anexo A. DTD for Customizing Graphs in Oracle Reports
- Anexo B. Carta de la Empresa Certificando el Cumplimiento la Práctica

GLOSARIO

CASE: acrónimo de “Computer Aided Software Engineering”, que traducida a español es Ingeniería del Software Asistida por Computador.

CASE METHOD: método de desarrollo de software que plantea una secuencia de etapas detallada abarcando descripción, definición de objetivos y metas, productos de la etapa, factores críticos de éxito, y la lista de tareas que conviene realizar. Se basa en un análisis y desarrollo del tipo descendiente ("topdown") en que el ciclo de vida de un sistema se compone de las siguientes etapas: estrategia, análisis, diseño (que incluye etapas de construcción y documentación), transición y producción.¹

DTD (DOCUMENT TYPE DEFINITION): define la estructura de un documento XML y puede estar dentro del mismo documento o en un archivo de texto separado. Describirá típicamente cada elemento admisible dentro del documento, los atributos posibles y (opcionalmente) los valores de atributo permitidos para cada elemento. Es más, describirá los anidamientos y ocurrencias de elementos. La mayoría de los DTD se componen generalmente de definiciones de ELEMENT y definiciones de ATTLIST.²

EXTENSIBLE MARKUP LANGUAGE (XML): es un lenguaje de marcado ampliable o extensible desarrollado por el World Wide Web Consortium (W3C). Se basa en documentos de texto plano en los que se utilizan etiquetas para delimitar los elementos de un documento. XML define estas etiquetas en función del tipo de datos que está describiendo, además de permitir definir nuevas etiquetas y ampliar

¹ SÁNCHEZ Víctor G. y RÍOS Homero V. “Metodología CASE para el desarrollo de Sistemas”. México. 1996. www.lania.mx/biblioteca/newsletters/1996-otono-invierno/articulo1.html.

² Tomado de es.wikipedia.org/wiki/DTD

las existentes. Entre sus aplicaciones se destaca el uso como estándar para el intercambio de datos entre diversos programas con lenguajes privados.³

J2EE (JAVA 2 ENTERPRISE EDITION): grupo de especificaciones diseñadas por Sun que permiten la creación de aplicaciones empresariales, esto sería: acceso a base de datos (JDBC), utilización de directorios distribuidos (JNDI), acceso a métodos remotos (RMI/CORBA), funciones de correo electrónico (JavaMail), aplicaciones Web(JSP y Servlets)...etc. Aquí es importante notar que J2EE es solo una especificación, esto permite que diversos productos sean diseñados alrededor de estas especificaciones. Algunos ejemplos son Tomcat y Weblogic.⁴

HERRAMIENTA CASE: herramienta software que da asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida del desarrollo.

ORACLE CONTAINERS FOR J2EE (OC4J): es el núcleo del componente de ejecución de J2EE del servidor de aplicaciones de Oracle.⁵

PL/SQL: lenguaje de programación procedimental de Oracle, que amplía las capacidades del estándar SQL. Los programas escritos en este lenguaje solo pueden compilarse utilizando una de las herramientas propias de Oracle.⁶

SINGLE SIGN-ON (SSO): se entiende por SSO toda tecnología que facilite las tareas de inicio de sesión y de acceso a recursos de red desde distintas plataformas. Se trata de un proceso que no tiene por qué constar necesariamente de una sola operación, ni conllevar siempre un inicio de sesión. Así, por ejemplo,

³ Tomado de es.wikipedia.org/wiki/XML

⁴ Tomado de www.osmosislatina.com/java/componentes.htm

⁵ Tomado de www.oracle.com/technology/tech/java/oc4j/index.html

⁶ Tomado de es.wikipedia.org/wiki/PL/SQL

las soluciones SSO piden a los usuarios, a veces, más datos de identificación antes de permitirles el acceso a otro sistema o red.⁷

UNIFIED MODELING LANGUAGE (UML): es un lenguaje de modelado que se usa para especificar, visualizar, construir y documentar los componentes de un sistema de software en forma gráfica. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.⁸

⁷ Tomado de www.windowstimag.com/atrasados/2000/40_mar00/articulos/suplemento/singlesign.htm

⁸ Tomado de www.creangel.com/uml/intro.php

RESUMEN

Título: ^{*} Codificación, realización de pruebas y documentación de programas para el registro de información y generación de reportes a través de la metodología Extreme Programming con el fin de consolidar el Sistema de Gestión de Calidad de Visión.

Autor: RICARDO ARMANDO PABÓN PINILLA ^{**}

Palabras claves: ISO 9000, Extreme Programming, Sistema de Gestión de Calidad, Oracle Developer 6i, Oracle 9i, ISO 9001:2000, GCI.

Descripción: La práctica empresarial desarrollada en la empresa Visión Ltda. aborda la necesidad de agilizar sus procesos internos relacionados con la gestión de calidad, definida por la familia de normas ISO 9000, a fin de incrementar su productividad mediante el desarrollo de una aplicación. Esta labor implicó actividades de codificación, realización pruebas y documentación de programas para el registro de información y generación de reportes orientadas por la metodología Extreme Programming y los procedimientos establecidos en la empresa bajo su Sistema de Gestión de Calidad, valiéndose de los recursos existentes tales como la herramienta software de desarrollo Oracle Developer 6i, la herramienta de bases de datos Oracle 9i y la infraestructura de redes y equipos disponible.

Haciendo referencia al Sistema de Gestión de Calidad, este recoge políticas y procesos para asegurar la calidad de un producto. La norma ISO 9001:2000, que sirve de base para certificar la calidad en la actualidad, proporciona los principios básicos de la gestión de la calidad y son reglas de carácter social encaminadas a mejorar la marcha y funcionamiento de una organización mediante la mejora de sus relaciones internas. En Colombia la norma que establece los requisitos sobre los cuales se estructura el Sistema de Gestión de Calidad es la Norma Técnica Colombiana NTC-ISO 9000:2000.

En el desarrollo del proyecto se realizó una evaluación de las herramientas de desarrollo disponibles, lo que logró establecer la más ajustada a las condiciones de la empresa; se establecieron claramente las opciones a realizar tomando como base el modelo GCI. Seguidamente, se elaboró un diseño detallado ajustado a los requerimientos definiendo las pruebas que validarían las opciones del sistema. Se continuó con la programación, validación y documentación de los formularios y reportes y finalmente se hizo entrega del producto.

* Proyecto de grado en la modalidad de Práctica Empresarial realizado en Visión Ltda..

** Facultad de Físico-Mecánicas, Escuela de Ingeniería de Sistemas, Tutor: Ing. Gerardo Armando Garavito Flórez, Director: Mg. José Cárcamo Sepúlveda.

ABSTRACT

Title: ^{*} Codification, tests and documentation of programs to register of information and reports generation supported in Extreme Programming method to consolidate the Vision Quality Management System.

Author: RICARDO ARMANDO PABÓN PINILLA ^{**}

Key words: ISO 9000, Extreme Programming, Quality Management System, Oracle Developer 6i, Oracle 9i, ISO 9001:2000, GCI.

Description: The developed enterprise practice in the company Visión Ltda. approaches to make agile the Quality Management internal processes. Quality Management System is regulated by ISO 9000 standard, and the Vision objective is to increase its productivity by means of software development. This work implied codification, tests and documentation activities of programs to register of information and reports generation oriented by the Extreme Programming method to consolidate the Vision Quality Management System and the procedures established in the enterprise, using the available means such as the software development tool Oracle Developer 6i, the database tool Oracle 9i and the networks infrastructure and computational instruments.

Quality Management System gathers policies and processes to assure the product quality. The ISO 9001:2000 standard, which serves as base to certify the quality at the present time, supplies the basic principles of Quality Management and are rules of social character directed to improve the march and operation of an organization by means of the improvement of its internal relations. In Colombia the Standard that groups the Quality Management System requirements is the Colombian Technical Standard NTC-ISO 9000:2000.

In the development of the project an evaluation of the available development tools were made, which managed to establish the most fit to the conditions of the company; the program options were determined by means of the GCI model; the detail design was made supported in the requirements defining the tests that would validate the options of the system; it was continued with the programming, validation and documentation of the forms and reports and finally the product was delivered.

* Project of degree in the modality of enterprise practice.

** Físico-Mecánicas Faculty, System Engineering School, Leading: Eng. Gerardo Armando Garavito Flórez, Director: Mg. José Cárcamo Sepúlveda.

INTRODUCCION

El objetivo de complementar la solución software para el manejo del Sistema de Gestión de Calidad de la empresa es apoyar y hacer más eficientes los procesos involucrados en el mismo, es decir, ayudar a que se produzca con la alta calidad que se logra por la aplicación de la norma ISO 9000:2000 con una notable reducción en los tiempos y costos de los procesos.

El desarrollo de este proyecto influirá solo en los aspectos relacionados con actividades asociadas a los proyectos, contratos, gestión del recurso humano, manejo de clientes, entre otros.

Lo correspondiente a los procedimientos mencionados anteriormente se hará mediante el desarrollo de un sistema para brindar soporte a los clientes contribuyendo también a lograr uno de los principios base de la norma: “Enfoque al Cliente”.

Para el proceso de Gestión Documental se deberá tener en cuenta parte de la documentación del Sistema de Gestión de Calidad, lo que abarcará también el conjunto de procedimientos documentados exigidos por la norma.

Después de la fase de iniciación y con un diseño previo, producto de proyectos anteriores relacionados con la gestión documental y entidades compatibles con las que se considerarán en el sistema a realizar, se continúa con la fase de elaboración, en donde se completa un diseño detallado del sistema para iniciar la codificación.

Para lograr la satisfacción del cliente con el producto final es indispensable refinar los requerimientos a través de la buena comunicación con los usuarios del sistema. Se realizan por lo tanto bajo la metodología de Extreme Programming

(XP) una serie de iteraciones en las que se hacen aproximaciones a una solución simple que pueda funcionar y ser implementada en un momento determinado del proyecto. Se considera un diseño adecuado para el software aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada y refleja claramente la intención de implementación de los programadores.

1 DESCRIPCIÓN DEL PROYECTO

1.1 OBJETIVOS

1.1.1 Objetivo General

Codificar, realizar pruebas y documentar programas para el registro de información y generación de reportes a través de la metodología Extreme Programming y los procedimientos establecidos en la empresa utilizando UML y las herramientas de desarrollo Oracle y XML con el fin de consolidar el Sistema de Gestión de Calidad de Visión.

1.1.2 Objetivos Específicos

- Definir las opciones a adaptar y desarrollar realizando una revisión del modelo GCI (Gestión Corporativa Integral) de la empresa para obtener un manejo del Sistema de Gestión de Calidad mejorado y ajustado a los requerimientos definidos por la empresa y complementando las opciones existentes.
- Programar opciones para el registro y generación de reportes que permitan reducir los tiempos de las actividades y el intercambio de información de los procesos del Sistema de Gestión de Calidad de Visión empleando Oracle y XML.
- Documentar las actividades y productos de los componentes para el registro y generación de reportes bajo los lineamientos de la metodología Extreme Programming, soportada con Case Method, y el lenguaje de modelado de software UML que se encuentren contemplados en el Sistema de Gestión de Calidad de Visión.

- Realizar pruebas orientadas por el Ingeniero tester de Visión en los escenarios establecidos por los procesos del Sistema de Gestión de Calidad que verifiquen la funcionalidad de las opciones elaboradas dentro de los requerimientos previamente establecidos por la empresa.

1.2 JUSTIFICACIÓN

1.2.1 Definición del Problema

Visión Ltda. es una empresa dedicada al “diseño, desarrollo, implantación, mantenimiento, soporte y comercialización de sistemas de información y suministro de servicios de consultoría informática” que posee un Sistema de Gestión de Calidad el cual se maneja de forma manual y semiautomatizada con la ayuda de recursos informáticos.⁹ La empresa requiere agilizar los procesos internos relacionados con la gestión de calidad con el fin de incrementar su productividad.

Además, Visión posee un sistema denominado GCI (Gestión Corporativa Integral) para el apoyo a la gestión administrativa y comercial de una empresa, dentro del cual se tiene implementado el manejo de un sistema de gestión de calidad, hecho que la empresa desea aprovechar para mejorar y adaptar la funcionalidad del modelo en este último aspecto para automatizar las actividades internas que generan demora.

1.2.2 Descripción de los Objetivos

Por la necesidad y oportunidad presente en la empresa, se requiere la elaboración de componentes software para el registro de información y la generación de

⁹ Tomado del alcance de la certificación ISO 9001:2000 de la empresa.

reportes que integren el sistema de gestión de calidad, permitan ejercer un mejor control de tiempos sobre las actividades de la empresa, y aumenten la productividad tanto del personal encargado de las funciones relacionadas con el manejo y control de la información entre los procesos como de los que se encuentran en otras actividades al permitírsele estar concentrados en las responsabilidades propias de estas. A estas opciones se le realizarán pruebas unitarias en escenarios establecidos para verificar la funcionalidad y operatividad de las mismas dentro de los requerimientos previamente establecidos por la empresa.

Además, como parte de la labor de ingeniería y el ajuste de los componentes a las condiciones de la misma, es necesario documentar las actividades y productos de los componentes que se elaboren conformándose a los requerimientos de la empresa, lo que implica actualizar los manuales relacionados con el control de los procesos de gestión de calidad en la empresa y los correspondientes al modelo GCI en el manejo del sistema de gestión de calidad.

1.2.3 Impacto y Viabilidad

Este proyecto contribuye al fin que la empresa desea lograr a través de la elaboración y documentación de componentes software para el registro de información y la generación de reportes que integrarán el sistema de gestión de calidad de visión, además de que mejorará el modelo GCI de la empresa lo que le representará un valor agregado.

Para esto la empresa posee las herramientas necesarias: Oracle 9i, XML, Developer 6i, Developer 9i, Designer 6i y Designer 9i, los lineamientos definidos en su sistema de gestión de calidad y un equipo de desarrollo, conformado por personal de la empresa y un estudiante en práctica empresarial encargado de elaborar e implementar las opciones necesarias. Este equipo deberá realizar tareas que comprenden el diseño del modelo de datos y los componentes

software, la selección de las herramientas adecuadas para el desarrollo, la organización en grupos de trabajo con variadas fortalezas técnicas y la elaboración e integración de módulos que compartan datos y funcionalidades, y que conlleven la realización de actividades como la asistencia técnica en el desarrollo, la revisión de operatividad de los programas, el desarrollo de pruebas y la generación de escenarios de operación de los componentes. Para esto se ha dispuesto que el estudiante que participa en el mismo realizando la presente práctica empresarial reciba la capacitación y orientación necesaria en cuanto a las herramientas que utilizará y la forma de trabajo dentro de la empresa.

1.3 PLAN DE TRABAJO

UML ofrece teoría sobre las fases que componen el desarrollo de un proyecto software, de la cual se ha adoptado para este proyecto específicamente, y por conveniencia de estar acorde con la metodología XP, el modelo de ciclo de vida de Iteración Controlada.

1.3.1 Fases del Modelo de Iteración Controlada

1. Fase I: Iniciación

Corresponde a una comprensión inicial de la necesidad o problema y la definición del producto que debe ser elaborado. La meta en esta etapa es establecer claramente cuáles son las opciones a realizar. Para esto se acordará en conjunto con la empresa cuales son los requerimientos de las mismas. Las actividades que se realizarán en esta etapa son:

- Revisión del modelo GCI
- Definición de las opciones a ajustar y elaborar

- Elaboración y entrega de informe de avance

2. Fase II: Elaboración

En esta fase la meta es completar un muy detallado diseño para iniciar la codificación del sistema. Los requerimientos aquí son refinados y surgen nuevos problemas en función de establecer los medios apropiados para la satisfacción de los requerimientos del usuario.

Para el desarrollo de esta fase se utilizará la herramienta Designer 9i, la cual esta basada en la metodología CASE que se fundamenta en un análisis y desarrollo del tipo descendiente ("topdown").

Designer permite a partir de los requerimientos obtenidos en la etapa de análisis realizar el diseño lógico y físico del modelo de datos y generarlo y además realizar la documentación requerida del sistema.

Las actividades de esta fase son:

- Diseño detallado de las opciones
- Elaboración y entrega de informe de avance
- Realización del plan de pruebas
- Diseño de la estructura de los documentos que generará el sistema

3. Fase III: Construcción

En este punto el objetivo es construir el sistema con una funcionalidad completa y verificar la operatividad del mismo. Adicionalmente son necesarias actividades de integración entre los módulos creados para lo cual quizás sean necesarios sistemas provisionales y subsistemas de integración. En esta etapa se realizarán las siguientes actividades:

- Codificación de opciones
- Ajuste de las opciones
- Elaboración y entrega de informes de avance
- Validación unitaria de las opciones
- Ajustes
- Pruebas funcionales y de operatividad
- Ajustes finales

4. Fase IV: Transición

El objetivo en la fase de transición es la adaptación del cliente al uso de la aplicación y documentar las aplicaciones realizadas y las nuevas necesidades que se pudieran presentar. Las actividades correspondientes son:

- Implantación de las opciones
- Ajustes
- Elaboración y entrega de informe de avance
- Documentación
- Entrega final de las opciones y su documentación
- Elaboración y entrega de informe final

2 MARCO TEÓRICO

2.1 SISTEMA DE GESTIÓN DE CALIDAD

En la actualidad el término Calidad ha afectado los sectores empresarial, comercial y de servicios como también a las personas a nivel individual, e incluso se han establecido modelos para dirigir y controlar todas las actividades asociadas a la calidad dentro de una empresa o institución denominados sistemas de gestión de calidad. Sin embargo, las personas asocian la calidad con el costo, la marca, la atracción o la durabilidad del producto. Aunque algunos de estos factores pueden formar parte del conjunto de cualidades para determinar la calidad de un producto, es mayor el número de atributos que se requieren. Por lo tanto, es necesario definir qué es calidad a fin de que exista un parámetro sobre el cuál juzgarla para un producto o servicio determinado y además considerar las normas actuales que la rigen.

2.1.1 Calidad

Se definirá la calidad de la siguiente manera: La calidad en un producto o servicio es el conjunto de atributos necesario para satisfacer las expectativas del cliente al menor costo.¹⁰ Esta definición involucra que para establecer la calidad de algo se debe considerar si se satisfacen las necesidades del cliente o usuario, siendo este alguien de la empresa o ajeno a ella, y que la forma como se realice resulte en un menor costo del mismo.

De acuerdo a lo anterior, es posible establecer que la calidad posee dos componentes: calidad de diseño y calidad de conformidad. La primera va ligada a

¹⁰ Curso Virtual sobre ISO 9001 del Sena regional Bucaramanga.

las funciones y características de un producto para lograr mayor satisfacción en el cliente mientras que la segunda al cumplimiento de las especificaciones propias del producto en cuanto a los límites en sus parámetros y su fiabilidad.

2.1.2 Normas de Calidad ISO 9000

Para asegurar la calidad de un producto se han creado normas y requisitos específicos. Un conjunto de estas es la familia de normas básicas de calidad ISO 9000, y es la base para certificar la calidad en la actualidad.

El primer conjunto de estas normas es la ISO 9000: Sistemas de Gestión de la Calidad – Principios y Vocabulario, la cual define los términos fundamentales utilizados en la familia de normas ISO 9000 que se necesitan para evitar malentendidos en su utilización.

Otro conjunto de normas es la ISO 9001: Sistemas de Gestión de la Calidad – Requisitos, que establece los requisitos para cumplir eficazmente con los requisitos del cliente y con los requisitos reglamentarios aplicables, para conseguir e incrementar la satisfacción del cliente, y está orientada a la eficacia del sistema de gestión de la calidad

Y finalmente el conjunto de normas ISO 9004: Sistemas de Gestión de la Calidad – Recomendaciones para la Mejora del Desempeño, proporciona ayuda para la mejora del sistema de gestión de la calidad a fin de beneficiar a todas las partes interesadas a través del mantenimiento de la satisfacción del cliente. Esta norma abarca tanto la eficacia como la eficiencia del sistema de gestión de la calidad.

Las normas conocidas, ISO 9001, ISO 9002 e ISO 9003 han sido integradas en la nueva ISO 9001:2000, que proporciona los principios básicos de la gestión de la calidad y son reglas de carácter social encaminadas a mejorar la marcha y funcionamiento de una organización mediante la mejora de sus relaciones

internas. Estas normas, han de combinarse con los principios técnicos para conseguir una mejora de la satisfacción del cliente, consiguiendo que este repita los hábitos de consumo y se haga fiel a los productos o servicios de la empresa.

El sistema de gestión de calidad de una empresa, de acuerdo a las normas ISO 9001 e ISO 9004, descansa sobre ocho principios fundamentales, estos son:

- 1.- Enfoque al cliente: las organizaciones dependen de sus clientes, por lo tanto deben comprender sus necesidades actuales y futuras, satisfacer sus requisitos y esforzarse en exceder sus expectativas.
- 2.- Liderazgo: los líderes establecen la unidad de propósito y la orientación de la organización. Deben crear y mantener un ambiente interno, en el cual el personal pueda llegar a involucrarse en el logro de los objetivos de la organización.
- 3.- Participación del personal: el personal, a todos los niveles, es la esencia de la organización, y su total compromiso posibilita que sus habilidades sean usadas para el beneficio de la organización.
- 4.- Enfoque basado en procesos: un resultado deseado se alcanza más eficientemente cuando las actividades y los recursos relacionados se gestionan como un proceso.
- 5.- Enfoque de sistema para la gestión: identificar, entender y gestionar los procesos interrelacionados como un sistema, contribuye a la eficacia y eficiencia de la organización en el logro de sus objetivos.
- 6.- Mejora continua: la mejora continua del desempeño global de la organización, debe de ser un objetivo permanente de esta.
- 7.- Enfoque basado en hechos para la toma de decisiones: las decisiones eficaces se basan en el análisis de los datos y en la información previa.

- 8.- Relaciones mutuamente beneficiosas con el proveedor: una organización y sus proveedores son interdependientes, y una relación mutuamente beneficiosa aumenta la capacidad de ambos para crear valor.

La norma ISO 9001:2000 dispone de los siguientes apartados donde se indican los requisitos a cumplir:

- 1.- Sistema de gestión de la calidad
- 2.- Responsabilidad de la dirección
- 3.- Gestión de los recursos
- 4.- Realización del producto
- 5.- Medición, análisis y mejora

Por razón de que el presente proyecto abordará los procedimientos internos relacionados con el sistema de gestión de calidad y además su realización implicará la elaboración de unos productos internos para el sistema ya establecido. A continuación solo se dará consideración a los capítulos 4 y 7 de la norma correspondientes a estas temáticas que incidirán notoriamente en el desarrollo de la práctica.

- Capítulo 4: Sistema de Gestión de la Calidad

Contempla los siguientes requisitos:

- a.- La organización debe establecer, documentar, implementar y mantener un sistema de gestión de la calidad.
- b.- La organización debe determinar la secuencia e interacción de los procesos y gestionarlos adecuadamente.

c.- En caso que la organización subcontrate un proceso externamente, debe controlar la calidad del mismo.

d.- La organización debe tener la siguiente documentación:

- 1.- Una política de la calidad.
- 2.- Objetivos de la calidad.
- 3.- Un manual de la calidad.
- 4.- Procedimientos documentados.
- 5.- Los documentos necesarios para asegurar la eficaz planificación, operación y control de los procesos.
- 6.- Los registros necesarios para demostrar el funcionamiento y eficacia del sistema de gestión de la calidad.

Como mínimo la norma obliga a tener los siguientes procedimientos para el sistema de gestión de calidad:

- 1.- Control de la documentación
- 2.- Control de los registros
- 3.- Auditorias internas
- 4.- Control de las no conformidades o del producto no conforme
- 5.- Acciones correctivas
- 6.- Acciones preventivas

- Capítulo 7: Realización del producto

Explica cómo se tiene que desarrollar la fabricación de un producto o la prestación de un servicio.

a.- Planificar y desarrollar los procesos necesarios para la realización del producto.

b.- Determinar los requisitos especificados por el cliente. Dentro de estos, deben figurar los requisitos para las actividades de entrega y las de postventa.

c.- Determinar los requisitos legales y reglamentarios relacionados con el producto.

d.- Planificar y controlar el diseño y desarrollo del producto. Esto implica registrar los elementos de entrada del diseño, verificar todos los resultados, validar el diseño y desarrollo de acuerdo con lo planificado y registrar todos los cambios del mismo.

e.- Asegurarse de que el producto comprado cumple los requisitos de compra especificados.

f.- Evaluar y seleccionar a los proveedores en función de su capacidad para suministrar productos de acuerdo a los requisitos.

g.- Establecer e implementar la inspección u otras actividades necesarias, para asegurarse de que el producto comprado cumple los requisitos.

h.- Planificar y llevar a cabo la producción y la prestación del servicio bajo condiciones controladas.

i.- Validar los procesos de producción y prestación del servicio donde los productos resultantes no puedan verificarse mediante actividades de seguimiento o medición posteriores.

j.- Identificar el producto por medios adecuados, a través de toda la realización del producto (trazabilidad).

k.- Cuidar los bienes que son propiedad del cliente, mientras estén bajo nuestro control.

l.- Preservar la conformidad del producto durante el proceso interno, y la entrega al destino previsto.

m.- Determinar el seguimiento y la medición a realizar, y los dispositivos de medición y seguimiento necesarios para proporcionar la evidencia de la conformidad del producto.

2.1.3 Norma Técnica Colombiana NTC-ISO 9000:2000

En Colombia la norma que establece los requisitos sobre los cuales se estructura el Sistema de Gestión de Calidad es la Norma Técnica Colombiana NTC-ISO 9000:2000. Esta norma establece seis (6) procedimientos obligatorios:

- 1.- Control de documentos
- 2.- Control de registros
- 3.- Control del producto no conforme
- 4.- Acciones correctivas
- 5.- Acciones preventivas
- 6.- Auditorias Internas

Además, determina dos (2) procesos que deben contemplarse:

- Gestión Documental

- Medición y Mejora

A continuación se considerarán en más detalle los procesos exigidos por la norma.

- **Gestión Documental**

En el Sistema de Gestión de la Calidad el requisito sobre la Documentación tiene como propósito establecer cómo está documentado el Sistema de Gestión de la Calidad en la organización. La norma identifica dos tipos de Documentos:

- ◆ Documentos
- ◆ Procedimientos documentados

Los documentos son la forma como usted provee información para el desempeño de las actividades en su organización.

El Procedimiento documentado se refiere a un procedimiento escrito obligatorio, que se exige para describir cómo su organización desempeña la actividad.

Requisitos de la Documentación: la Documentación del Sistema de Gestión de Calidad puede estar en cualquier tipo de medio y debe incluir:

- a) Declaraciones documentadas de una Política de la Calidad y de Objetivos de la calidad.
- b) Un manual de la calidad, el cual especifica el sistema de gestión de calidad de la empresa y contiene: alcance del sistema, reseña histórica de la empresa, misión, visión, mapa de procesos, política y objetivos de calidad y caracterizaciones de los procesos.
- c) Los procedimientos documentados requeridos en esta Norma Internacional (ISO 9001).

- d) Los documentos necesitados por la organización para asegurarse de la eficaz planificación, operación y control de sus procesos.
- e) Los registros requeridos por la Norma Internacional ISO 9001.

- **Medición y Mejora**

Comprende las siguientes actividades:

- Medir la satisfacción del cliente respecto a nuestros productos y organización.
- Realizar a intervalos planificados auditorias internas para determinar el estado del sistema de gestión de la calidad.
- Aplicar los métodos apropiados para medir la evolución de los procesos del sistema de gestión de la calidad.
- Medir y hacer un seguimiento de las características del producto para verificar que se cumplen los requisitos del mismo.
- Asegurar que el producto que no sea conforme con los requisitos, se identifica y se controla para prevenir su uso o entrega no intencionada.
- Determinar, recopilar y analizar los datos para demostrar la idoneidad y la eficacia del sistema de gestión de la calidad.
- Mejorar continuamente la eficacia del sistema de gestión de la calidad.
- Tomar acciones correctivas para eliminar las causas de las no conformidades, con objeto de prevenir que vuelvan a ocurrir.
- Tomar acciones preventivas para eliminar las causas de las no conformidades potenciales, con objeto de prevenir su ocurrencia.

2.2 EXTREME PROGRAMMING (XP)

Es una metodología que implementa un ágil estilo iterativo de desarrollo y esta centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

XP promueve en su práctica los siguientes principios fundamentales:

- Realimentación continua entre el cliente y el equipo de desarrollo. Esto se realiza a través de historias de usuario que consisten en fichas en las cuales el usuario establece los requerimientos a través de descripciones breves de las características que debe tener el sistema.
- Comunicación fluida entre todos los participantes
- Simplicidad en las soluciones implementadas
- Coraje para enfrentar los cambios.

Además define sus reglas y prácticas en cuatro etapas principales:

- Planeación
- Diseño
- Codificación
- Pruebas

A continuación se detallan cada una de estas.

2.2.1 Etapas del Proceso de Desarrollo

- Planeación

Esta etapa corresponde al proceso durante el cual se hace una clara identificación de los requerimientos y se proyectan las acciones de desarrollo necesarias a ser realizadas. Se descompone en dos fases: exploración y planificación de la entrega

- Fase I: Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de poco tiempo dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

- Fase II: Planificación de la Entrega

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Esta fase dura unos pocos días.

- Diseño

Esta etapa corresponde a la definición de la estructura del sistema e incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se

puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Al final de la última iteración el sistema estará listo para entrar en producción.

Los elementos que deben tomarse en cuenta durante la elaboración del plan de la iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior.

- **Codificación**

En esta etapa se construye el sistema como tal y se realizan pruebas preliminares al código por parte de los programadores para validar que posea las características establecidas por el cliente o usuario.

- **Pruebas**

El poner el sistema en funcionamiento requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación y reiniciar el proceso en busca de realizar una nueva versión.

2.2.2 Prácticas XP

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. XP apuesta por un crecimiento lento del costo del cambio y con un comportamiento asintótico. Esto se consigue gracias a las

tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las prácticas que se describirán a continuación.

- El juego de la planificación

Es un espacio frecuente de comunicación entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración. Esta práctica se puede ilustrar como un juego, donde existen dos tipos de jugadores: Cliente y Programador. El cliente establece la prioridad de cada historia de usuario, de acuerdo con el valor que aporta para el negocio. Los programadores estiman el esfuerzo asociado a cada historia de usuario. Se ordenan las historias de usuario según prioridad y esfuerzo, y se define el contenido de la entrega y/o iteración, apostando por enfrentar lo de más valor y riesgo cuanto antes. Este juego se realiza durante la planificación de la entrega, en la planificación de cada iteración y cuando sea necesario reconducir el proyecto.

- Entregas pequeñas

La idea es producir rápidamente versiones del sistema que sean operativas, aunque obviamente no cuenten con toda la funcionalidad pretendida para el sistema pero si que constituyan un resultado de valor para el negocio.

- Metáfora

En XP no se enfatiza la definición temprana de una arquitectura estable para el sistema. Dicha arquitectura se asume evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de

desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema.

- Diseño simple

Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente. Kent Beck dice que en cualquier momento el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos.

- Pruebas

La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial.

- Refactorización (*Refactoring*)

La refactorización es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. La refactorización mejora la estructura interna del código sin alterar su comportamiento externo. No se puede imponer todo en un inicio, pero en el transcurso del tiempo este diseño evoluciona conforme cambia la funcionalidad del sistema. Para mantener un

diseño apropiado, es necesario realizar actividades de cuidado continuo durante el ciclo de vida del proyecto.

- Programación en parejas

Toda la producción de código debe realizarse con trabajo en parejas de programadores. Las principales ventajas de introducir este estilo de programación son: muchos errores son detectados conforme son introducidos en el código (inspecciones de código continuas), por consiguiente la tasa de errores del producto final es más baja, los diseños son mejores y el tamaño del código menor (continua discusión de ideas de los programadores), los problemas de programación se resuelven más rápido, se posibilita la transferencia de conocimientos de programación entre los miembros del equipo, varias personas entienden las diferentes partes sistema, los programadores conversan mejorando así el flujo de información y la dinámica del equipo, y finalmente, los programadores disfrutan más su trabajo.

- Propiedad colectiva del código

Cualquier programador puede cambiar cualquier parte del código en cualquier momento. Esta práctica motiva a todos a contribuir con nuevas ideas en todos los segmentos del sistema, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código.

- Integración continua

Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente. La integración continua a menudo reduce

la fragmentación de los esfuerzos de los desarrolladores por falta de comunicación sobre lo que puede ser reutilizado o compartido.

- Cliente in-situ

El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Gran parte del éxito del proyecto XP se debe a que es el cliente quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita, ya que esta última toma mucho tiempo en generarse y puede tener más riesgo de ser mal interpretada.

- Estándares de programación

XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios.

2.3 BASES DE DATOS

En ingeniería del software una base de datos es una estructura física y funcional que agrupa y organiza un conjunto de datos relacionados que conforman la información requerida y generada por un sistema computacional. De acuerdo a su estructura puede ser de varios tipos: jerárquica, relacional y orientada a objetos. El modelo más usado en la actualidad es el relacional cuyo objetivo es obtener integridad y evitar la redundancia de la información dentro del sistema.

La base de datos puede ser representada de forma lógica y física. La representación lógica corresponde a la descripción de las entidades que

intervienen en el sistema y sus relaciones entre sí. La representación física define la estructura de las tablas de la base de datos y el tipo de datos que contienen.

Para la construcción y administración de bases de datos se utilizan los sistemas manejadores de bases de datos cuyo fundamento para la estructuración y recuperación de los datos, en el caso de los tipos relacionales, es el lenguaje SQL.

2.3.1 Oracle

Es una herramienta cliente/servidor integral y robusta para la gestión de Bases de Datos (SGDB) relacionales. Oracle es uno de los sistemas de bases de datos más completos, destacándose entre sus características el soporte de transacciones que ofrece, su estabilidad, la escalabilidad, ser multiplataforma y su alto rendimiento.

La versión de Oracle utilizada en el desarrollo del presente proyecto para la elaboración y manejo de la base de datos del sistema fue Oracle 9i.

2.4 ORACLE 9I DEVELOPER SUITE

Oracle 9i Developer Suite (Oracle9iDS) es un completo e integrado conjunto de herramientas que permite:

- Construir aplicaciones de internet y servicios web de transacciones usando Java y XML.
- Implementar inteligencia de negocios en las aplicaciones sin el costo de herramientas adicionales.

Ofrece soporte completo para el ciclo de vida del desarrollo, abarcando modelado, codificación, pruebas y validación del código. Brinda también facilidades para el

trabajo en equipo, control y gestión del desarrollo para administrar la continua evolución de las aplicaciones.

Oracle9iDS se integra firmemente con Oracle 9i Application Server (Oracle9iAS) y la base de datos de Oracle9i, proporcionando una plataforma integrada para la construcción y ejecución de aplicaciones de internet y servicios web escalables, rápidos y confiables que pueden accederse desde cualquier dispositivo, en cualquier lugar y en cualquier momento.

Oracle9i Application Server (Oracle 9iAS) es el servidor de Aplicaciones de Oracle que ofrece una solución consistente y abierta para las aplicaciones Internet, basada en estándares, en tecnologías orientadas a componentes del tipo Java y J2EE que facilitan el desarrollo rápido y la reutilización de código, así como la ejecución de aplicaciones web sobre una amplia gama de plataformas (Linux, Windows 2000/NT, Unix,...) con las capacidades de escalabilidad necesarias para servir a un elevado número de usuarios.

Las herramientas integradas a Oracle 9i Developer Suite que se analizarán en el presente documento por ser de interés en el desarrollo de la práctica son: Oracle Designer, Forms Builder y Reports Builder.

2.4.1 Oracle Designer

Es una herramienta CASE de Oracle que permite crear sistemas que reflejan las necesidades de negocio y requerimientos de una organización. Reúne un conjunto de herramientas que soportan una amplia gama de tareas en las áreas de estrategia, análisis, diseño y generación de código, cubriendo así todas las fases del ciclo de vida del desarrollo. Tiene un repositorio o sitio central de almacenamiento donde se recoge la información pertinente de los proyectos de sistemas de forma integrada. (Véase figura 1)



Figura 1. Object Navigator

Las áreas en las que se agrupan los componentes de Designer 9i son: *Model System Requirements* (Requerimientos del modelo del sistema), *Transform Preliminary Designs* (Transformar los diseños preliminares), *Design and Generate* (Diseñar y generar) y *Repository Tools* (Herramientas del repositorio).

- Model System Requirements: agrupa componentes que permiten: modelamiento de procesos del sistema, re-examinar los métodos usados para lograr las metas de la organización, documentar los procesos, elaborar los diagramas de flujo de datos del sistema, elaborar diagramas jerárquicos de funciones y crear el modelo entidad relación del sistema sobre la base de la metodología CASE.
- Transform Preliminary Designs: los componentes de esta área permiten transformar los modelos lógicos definidos del sistema en diseños físicos preliminares tanto de la base de datos como de la aplicación.

- Design and Generate: sus componentes permiten diseñar un sistema que ayude a solucionar las necesidades la Empresa, proporcione un entorno de desarrollo para los ingenieros y diseñadores de sistema, cree componentes del servidor y aplicaciones del cliente con las definiciones registradas en el repositorio.
- Repository Tools: reúne herramientas de esta que permiten entrar y editar la información en el repositorio, mostrar las relaciones entre los elementos del repositorio, generar vistas y reportes a la medida de los contenidos del repositorios, administrar la información y realizar procesos de ingeniería inversa.

2.4.2 Forms Builder

El diseño de los formularios en Oracle 9i Forms Developer se fundamenta en la estructura de la base de datos del sistema a implementar. La definición de estos requiere un concienzudo establecimiento de los requerimientos.

Al iniciar el programa se creará un módulo de formulario por defecto en la ventana "Object Navigator". (Véase figura 2)

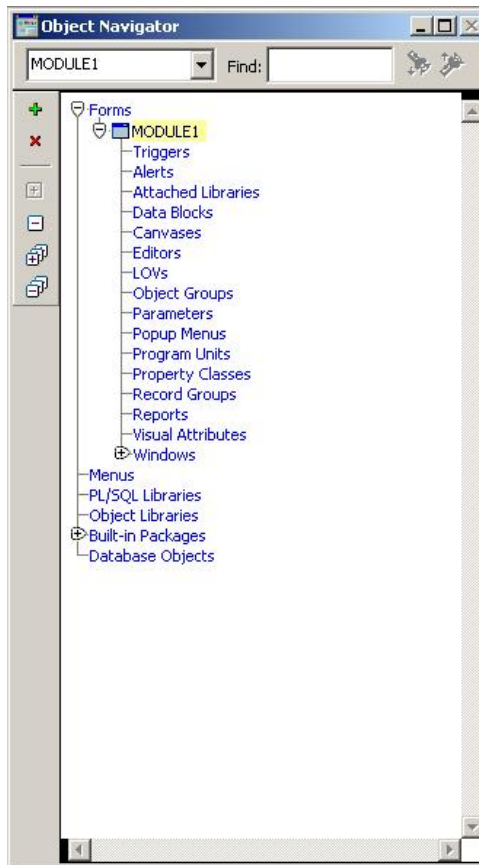


Figura 2. Object Navigator

2.4.2.1 Elementos básicos de los formularios

Los elementos básicos de los formularios son: Canvases, Data Blocks, Triggers, Alerts, LOVs, Windows, Program Units y Reports.

- **Canvases:** se refiere a la superficie en la cual son desplegados los elementos del formulario. Los hay de varios tipos:
 - **Content:** canvas básico de contenido.
 - **Stacked:** canvas que se ubica dentro de otro canvas.
 - **Vertical Toolbar:** canvas que define una barra de herramientas vertical.


- **Horizontal Toolbar:** canvas que define una barra de herramientas horizontal.
- **Tab:** canvas tipo tab o de pestañas.
- **Data Blocks:** estructura de datos que puede estar asociada a una tabla de la base de datos y que contiene los elementos que componen el formulario.
- **Triggers:** son rutinas PL/SQL que se ejecutan en respuesta a eventos efectuados por el usuario o acciones del sistema.
- **Alerts:** elementos que definen las ventanas de advertencia que se muestran ante acciones efectuadas por el usuario asociadas a confirmación de la acción o errores en el uso de la aplicación.
- **LOVs:** define una ventana en la que se despliega una lista de valores válidos que pueden ser ingresados por el usuario en un determinado campo del formulario.
- **Windows:** elemento que define las ventanas en las que se mostraran los formularios.
- **Program Units:** unidades de programa PL/SQL utilizadas en el formulario correspondientes a: procedimientos, funciones o packages.
- **Reports:** define los reportes asociados al formulario.

2.4.2.2 Asistente para la construcción de bloques

Iniciamos el asistente para la elaboración del formulario haciendo clic en el botón



de la barra de herramientas, o haciendo clic con el botón derecho del mouse en cualquier parte del navegador de objetos y seleccionando la opción Data Block

Wizard o ubicándonos en el navegador de objetos dentro del módulo seleccionando la sección “Data Blocks” y haciendo clic en el botón  de la barra lateral del navegador, acción que desplegará una ventana donde debera escoger entre crear un bloque manualmente o mediante el asistente (Véase figura 3).

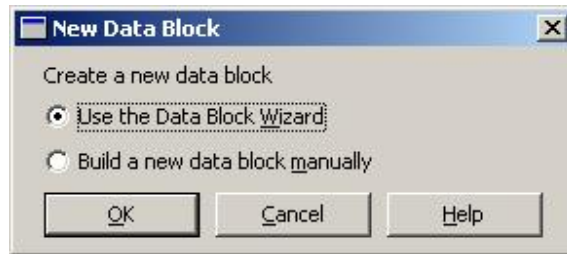


Figura 3. Ventana de creación de un bloque de datos

Al seleccionar el asistente se muestra una ventana en donde se indica el tipo de origen de los datos para el bloque que se va a crear, si provienen de una tabla o una vista o de un procedimiento almacenado. (Véase figura 4)

Oprimiendo el botón “Next” se continúa a la siguiente ventana en donde indicaremos a que tabla o vista estará asociado el bloque y los campos que incluirá. Al hacer clic en el botón “Browse” deberá estar conectado a la base de datos para poder seleccionar la tabla o vista de la base de datos. (Véase figuras 5 y 6)



Figura 4. Tipo de Origen de los Datos

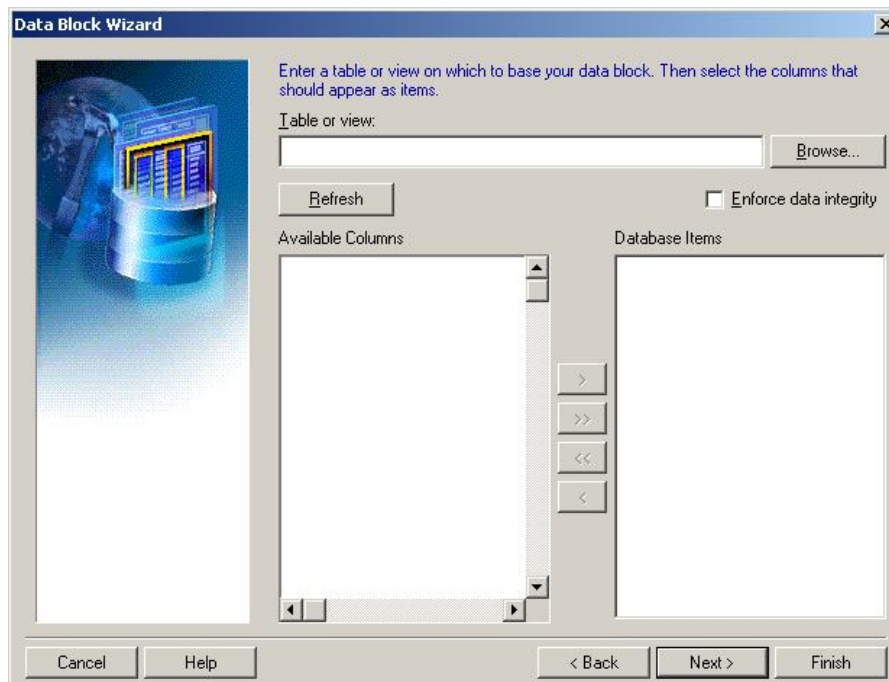


Figura 5. Elementos del Bloque

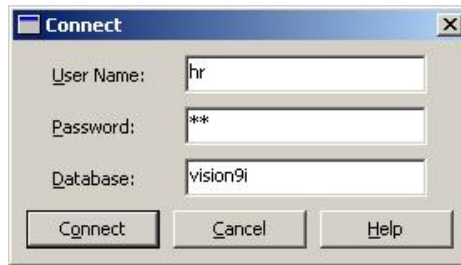


Figura 6. Conexión a la Base de Datos

Estando conectado, al hacer clic en el botón “Browse” se mostrará en donde podrá seleccionar de una lista de los elementos de la base de datos la tabla o vista que desea asociar al bloque de datos. Para esto la ventana cuenta en su parte superior unas cajas de chequeo que indican los tipos de elementos que se muestran en la lista. (Véase figura 7)

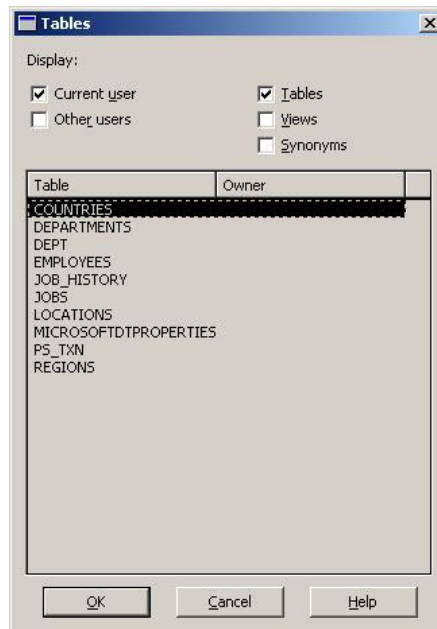


Figura 7. Selección de la Tabla o Vista Asociada al Bloque

Al seleccionar la tabla o la vista, entonces se cargaran todos los campos correspondientes en la caja de texto de la parte izquierda de la ventana de elementos del bloque. A continuación deberá indicar los elementos que corresponderán al bloque de datos. (Véase figura 8)

Se continúa entonces con una ventana en donde podrá asignarle un nombre al bloque. Por defecto es el mismo nombre de la tabla o vista seleccionada.

Finalmente, se muestra una ventana en donde debe indicar si solo desea crear el bloque de datos o continuar con el asistente de definición de los elementos del formulario. Para finalizar el asistente hacemos clic en "Finish". (Véase figura 9)

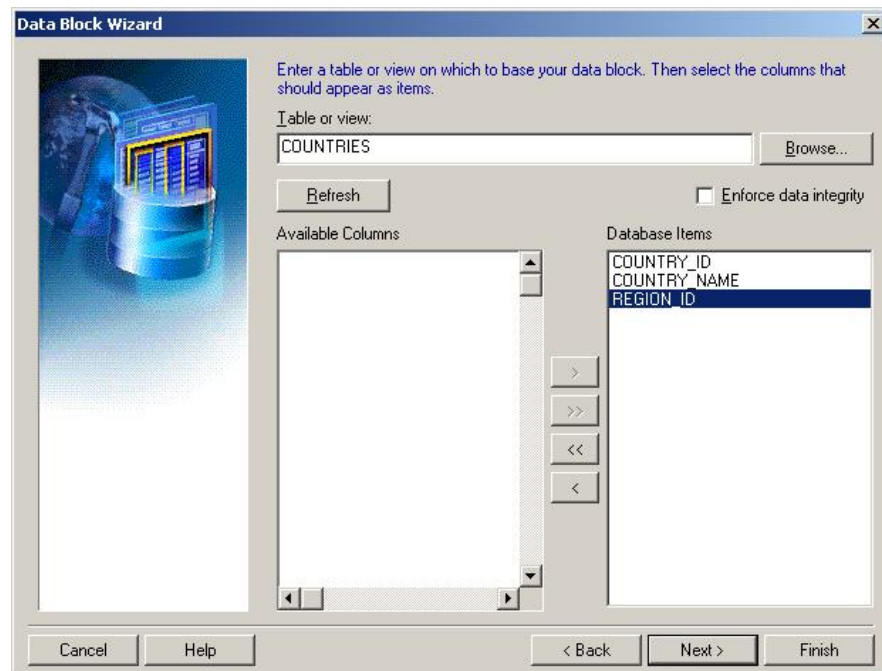


Figura 8. Indicando los Elementos del Bloque



Figura 9. Fin del Asistente de Bloques

2.4.2.3 Asistente para la definición de los elementos del formulario

Al iniciar el asistente denominado “Layout Wizard” se muestra una ventana en donde se define el Canvas en el que mostraran los elementos. (Véase figura 10)

A continuación se pasa a una ventana en donde definirán los elementos del bloque que se mostraran. Para cada elemento del bloque, en esta ventana se puede definir el tipo de elemento del formulario en que se mostrará, bien sea una caja de texto, una menú desplegable, una caja de chequeo, etc. (Véase figura 11)

Al hacer clic en “Next” se muestra una ventana en donde definimos algunas propiedades de los elementos establecidos como título que tendrán y su tamaño. (Véase figura 12)

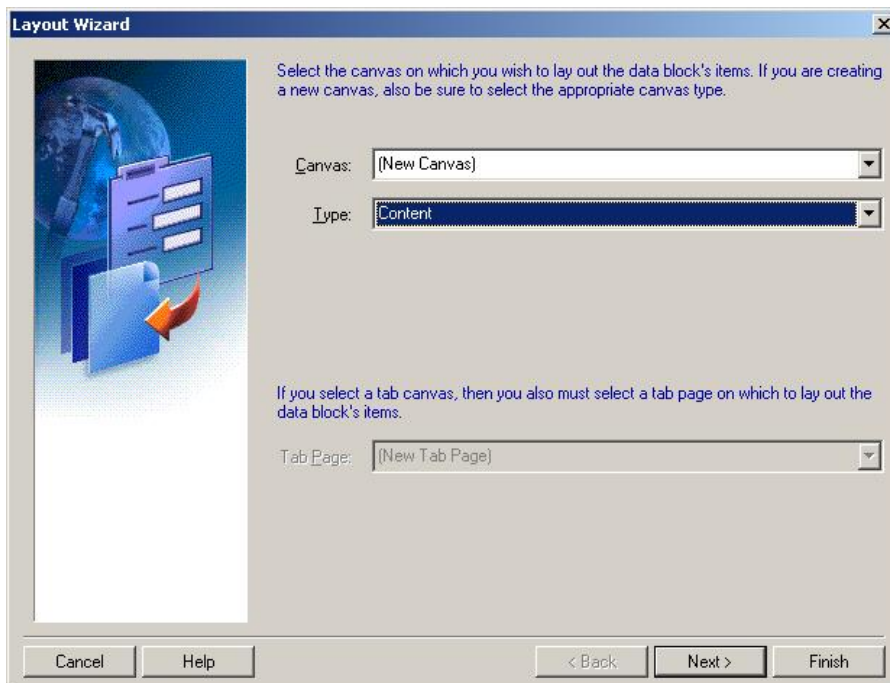


Figura 10. Definición del Canvas.

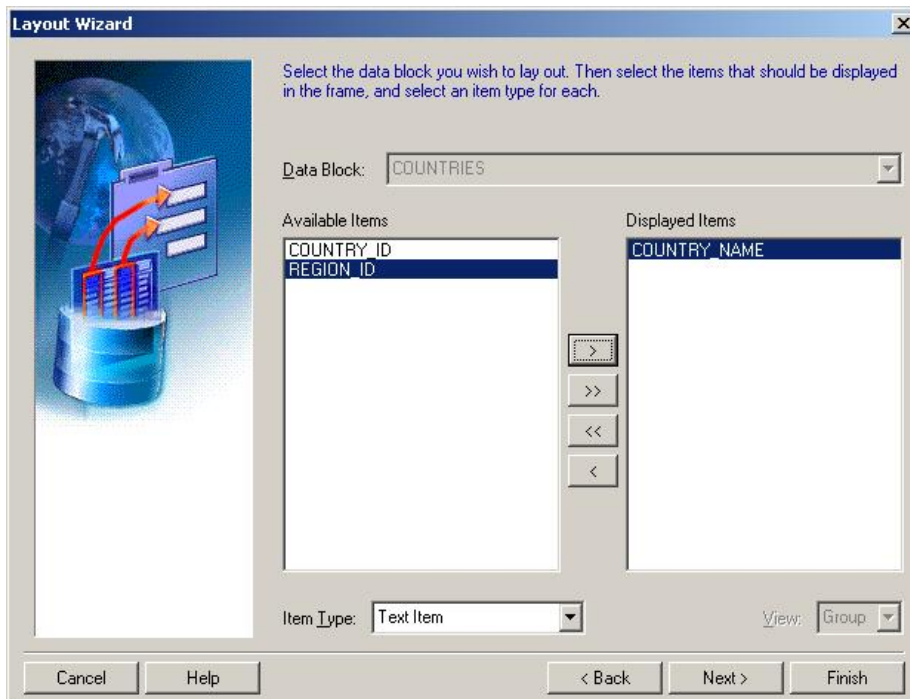


Figura 11. Selección de Campos del Bloque a ser Mostrados

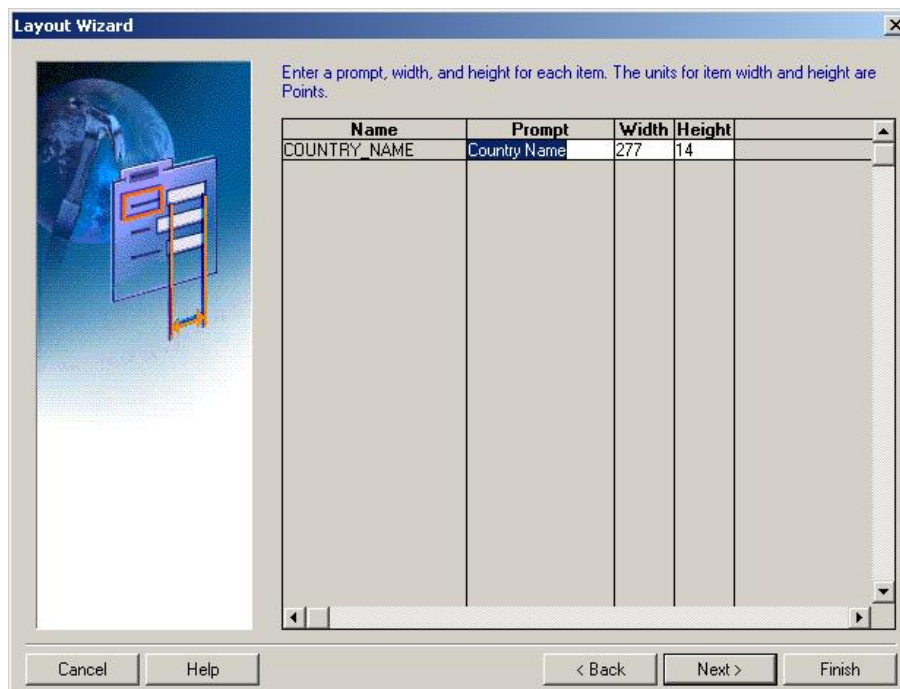


Figura 12. Definición de Propiedades de los Items

Se indica en la siguiente ventana la forma como se desea mostrar los elementos: como formulario o en forma tabular. La primera indica que cada elemento se distribuirá en el canvas en forma independiente, mientras que el segundo distribuye los elementos horizontalmente como columnas de una tabla.

Haciendo clic en “Next” se pasa a una ventana en donde se indica el título del frame donde se mostrarán los elementos, el número de registros por campo, la distancia entre los elementos y si se desea mostrar una barra de deslizamiento para recorrer los registros del bloque.

Al terminar se generará el formulario automáticamente. (Véase figura 13)

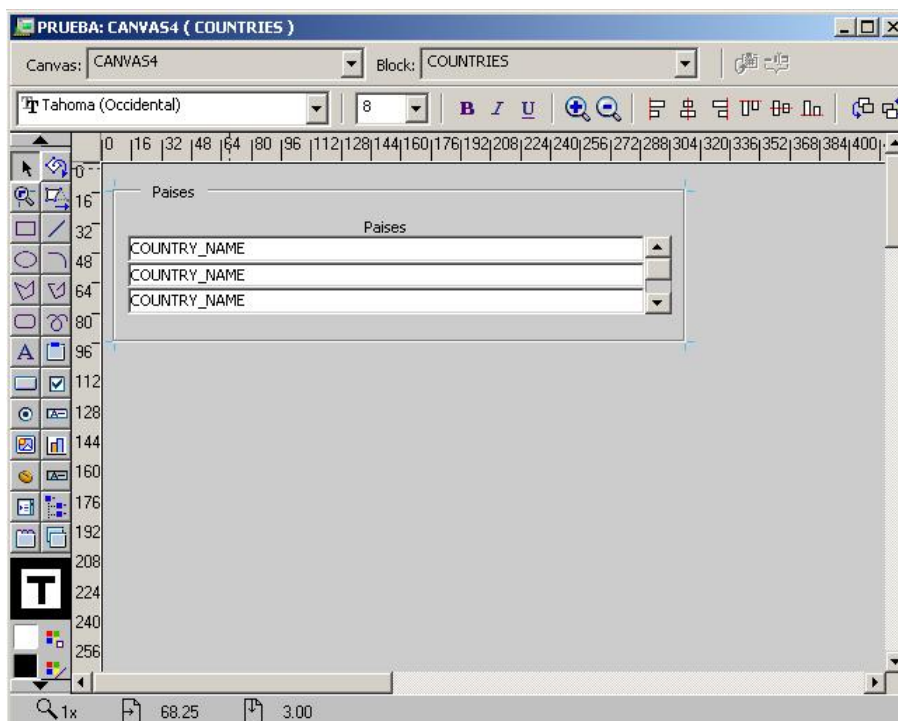


Figura 13. Formulario Generado

2.4.2.4 Crear un bloque de detalle

En la elaboración de formularios una cuestión fundamental es el poder desplegar la información contenida en varias tablas relacionadas entre sí. Esto se conoce como un formulario Maestro – Detalle.

Para la elaboración de un formulario de este tipo nos valdremos del asistente para la creación de bloques. Primero se creará el bloque maestro y a continuación se procederá a crear el bloque de detalle, para el cual se debe tener en cuenta que debe tener algún tipo de relación con el bloque maestro.

Al utilizar el asistente para crear el bloque de detalle, después de establecer la tabla o vista asociada al bloque, se mostrará una ventana que nos permitirá asociar el bloque a su bloque maestro a través de la relación que definamos entre ellos. (Véase figura 14)

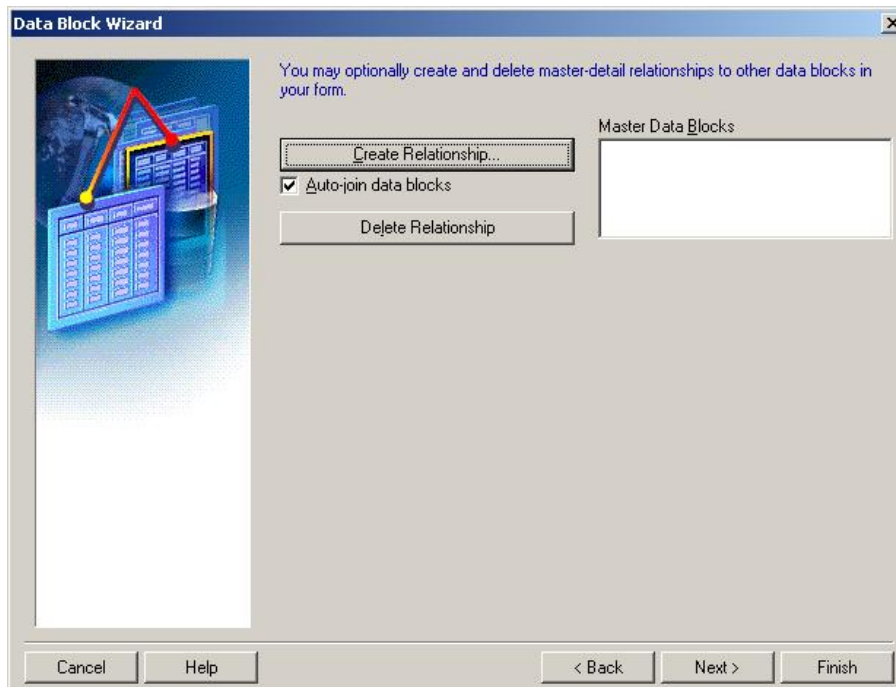


Figura 14. Definiendo la Relación con el Bloque Maestro

Al hacer clic en el botón “Create Relationship” y teniendo chequeada la caja Auto-join data blocks, se nos mostrará una ventana en donde se nos mostrarán los bloques con los que existe una relación definida. Se selecciona de la lista la relación que queremos establecer. (Véase figura 15)

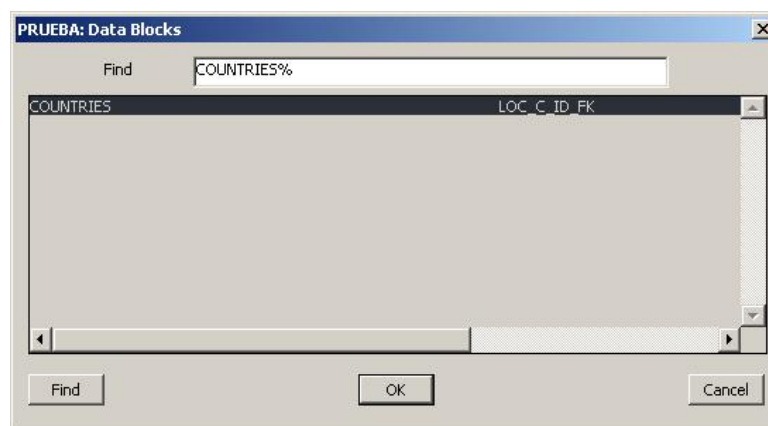


Figura 15. Relaciones entre Bloques

Como resultado se muestra la condición de join entre las tablas que define el bloque como un bloque de detalle. Se continúa con los pasos ya vistos del asistente para la definición del bloque. (Véase figura 16)

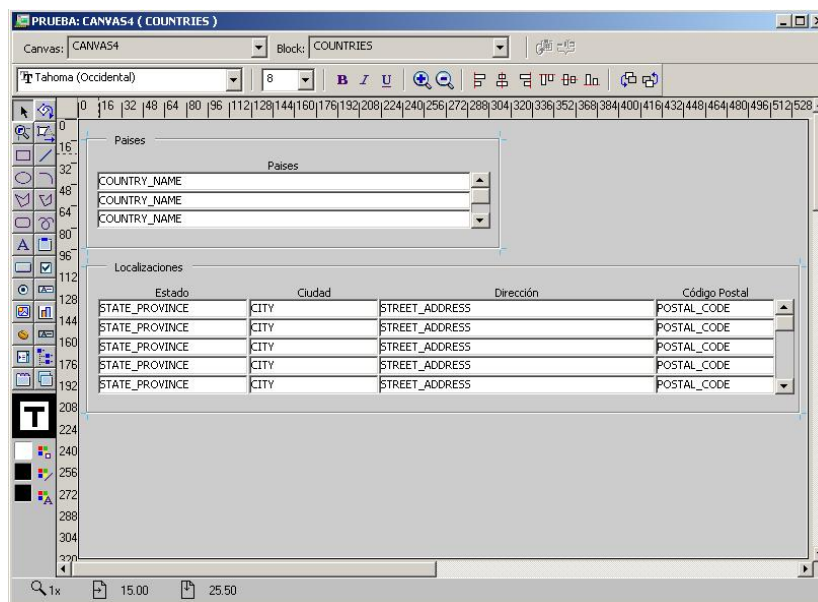



Figura 16. Formulario con Bloques Maestro y Detalle

Automáticamente Oracle 9i Forms Builder crea el manejo de eventos asociados con la modificación de los registros entre el maestro y el detalle para evitar errores asociados a la relación entre las tablas padre e hijo de la base de datos.

2.4.2.5 Crear un bloque de control e ítems no asociados a una tabla

En función de hacer una interfase amigable al usuario, a veces se requiere mostrar información adicional en un formulario que no esta asociada a las tablas en las cuales están basados los bloques que lo conforman. En estos casos resulta práctico crear ítems no asociados a una tabla.

Para crear ítems no asociados a una tabla nos ubicamos en la sección “ítems” del bloque donde deseamos crear el ítem en el navegador de objetos y damos clic en el botón  de la barra de herramientas lateral. A continuación modificamos las propiedades del ítem en la paleta de propiedades, la cual se activa haciendo clic con el botón derecho del mouse sobre el ítem y seleccionan “Property Palette”. (Véase Figura 17)

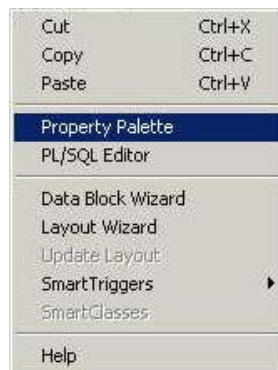


Figura 17. Mostrar la Ventana de Propiedades

Una vez en esta ventana, en la sección general podemos cambiar el nombre del ítem y el tipo. En la sección Database, en el atributo Database Item indicamos que no está asociado a la base de datos cambiando su valor a “NO”. (Véase figura 18)

Por otro lado, también puede definir bloques que no estén asociados a una tabla, los cuales se denominan bloques de control y cuyos ítem no están asociados a una base de datos. Estos son prácticos para definir parámetros de control que debe suministrar el usuario en tiempo de ejecución. Un ejemplo de esto son ventanas que adicionales que se muestran al efectuar una acción en un formulario y que solicitan parámetros necesarios para continuarla como la que se muestra en la figura 18 y que corresponde a una ventana utilizada en el desarrollo del presente proyecto para la generación de un archivo en Excel con el contenido de los datos generados en un formulario. (Véase figura 19)



Figura 18. Ventana de Propiedades

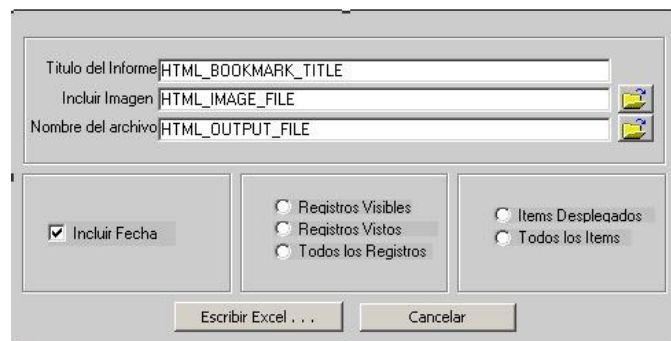



Figura 19. Ventana de Ejemplo de Uso de un Bloque de Control

2.4.2.6 Crear listas dinámicas

Las listas dinámicas son estructuras que permiten al usuario seleccionar los datos que desea que se asignen a los elementos de un formulario. Estas son denominadas Listas de Valores y se identifican con la sigla LOV. Se basa en una consulta SQL, lo que le otorga la propiedad de dinamicidad porque esta consulta es realizada cada vez que el usuario llama la lista de valores.

Para crear una lista de valores debe ubicarse en la sección “LOV” del formulario en el navegador de objetos y hacer clic en el botón  de la barra de herramientas lateral. A continuación se mostrará una ventana en la que deberá indicar si desea crearla manualmente o usando el asistente. Usando el asistente inicialmente se mostrará una ventana donde se indicará que se desea crear una lista de valores basada en una consulta. Al hacer clic en “Next” se mostrará la ventana en donde ingresaremos la consulta en la que se basará la lista de valores. (Véase figura 20)

El botón “Check Syntax” permite asegurarse de que la consulta es correcta. Se hace clic en “Next” para continuar.

Se muestra seguidamente una ventana en donde se indicara los campos de la consulta que se incluirán como columnas en la lista de valores. Es necesario tener en cuenta que se debe incluir todos los campos que serán asignados al seleccionar un elemento de la lista. (Véase figura 21)

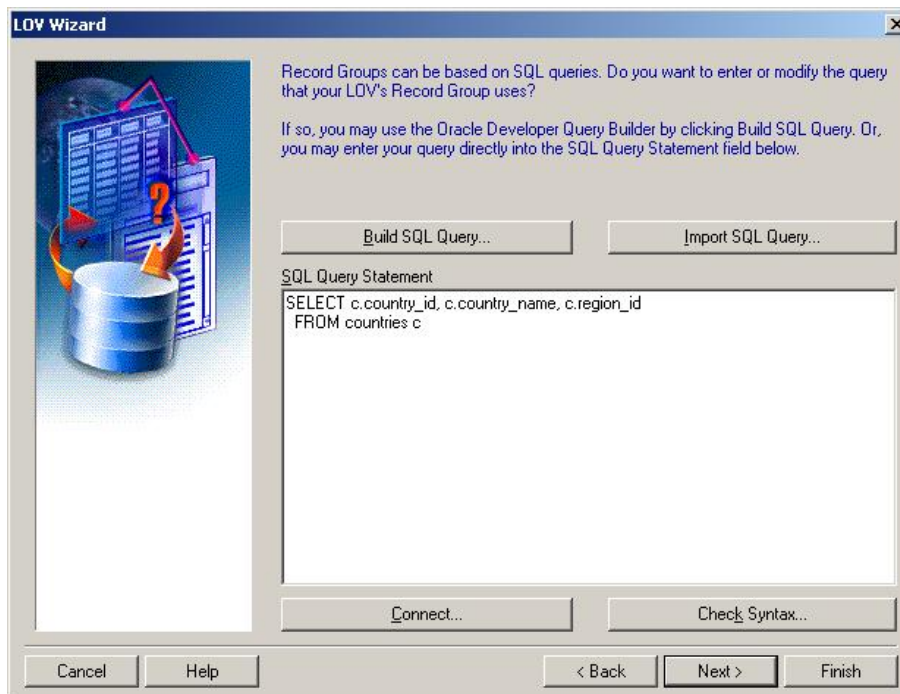


Figura 20. Consulta para la Lista de Valores

Continuando con el asistente se muestra una ventana en la que se definirán las propiedades de las columnas que se mostraran en la lista de valores. Además se debe indicar a que elementos del formulario se asignaran los valores retornados por la lista. Para esto se debe ubicar en la columna "Return Value" y hacer clic en el botón "Look up return item", el cual desplegará una lista de los elementos del formulario a los que podrá asignar el valor de la columna de la lista de valores. Es necesario tener en cuenta que la lista debe retornar al menos uno de los campos que se indicaron y aquellos elementos que no se desean ver al desplegarse la lista colocarles un ancho de 0. (Véase figuras 22 y 23)

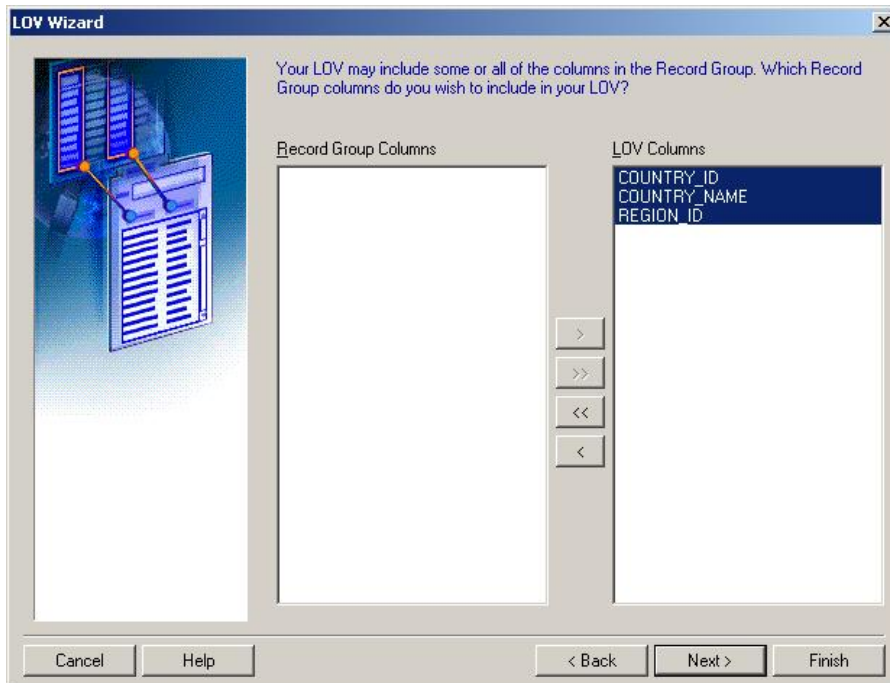


Figura 21. Columnas de la Lista de Valores

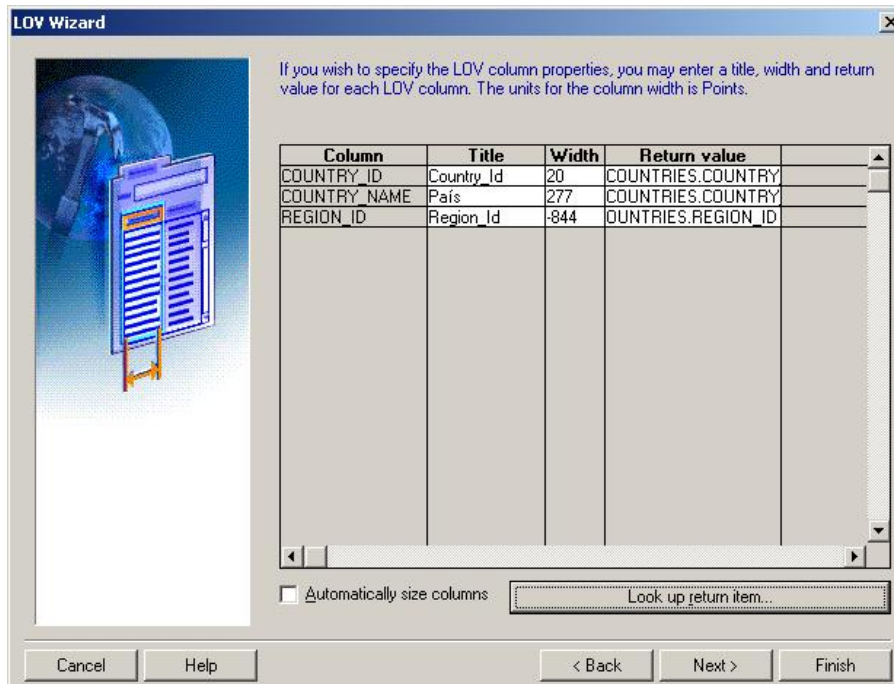


Figura 22. Propiedades de las Columnas de la Lista de Valores

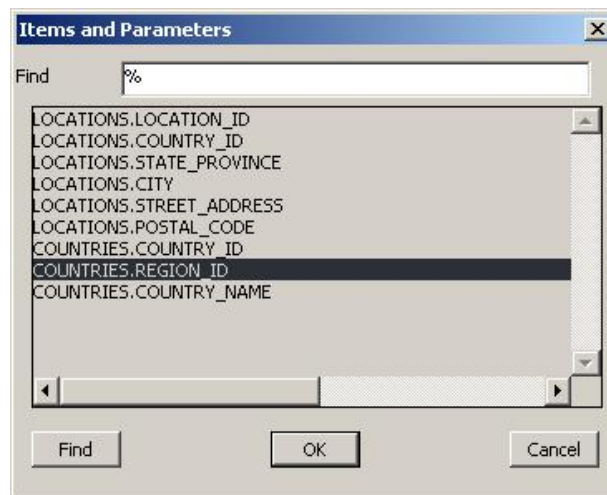


Figura 23. Elemento del Formulario al que se Retornará el Valor

Continúan dos ventanas en donde se definen las propiedades de la ventana de la lista de valores. En la primera se definen el título, el tamaño y la posición y en la segunda se indica el número filas que se cargaran en la ventana inmediatamente el usuario la dispere.

Seguidamente, se mostrará una ventana en donde se indicarán los elementos del formulario que en los que el usuario podrá disparar la lista de valores. (Véase figura 24)

Finalmente, para dar por terminada la definición de la lista de valores se dará clic en el botón “Finish”.

Al correr el formulario y cargarse la ventana con los menús y la barra de herramientas por defecto de Oracle 9i Forms Builder la lista de valores se podrá disparar ubicándose en alguno de los elementos que asociamos en el asistente para esta acción y pulsando las teclas “CTRL + L” o en el menú “Edit” en la opción “Display List”. (Véase figura 25)

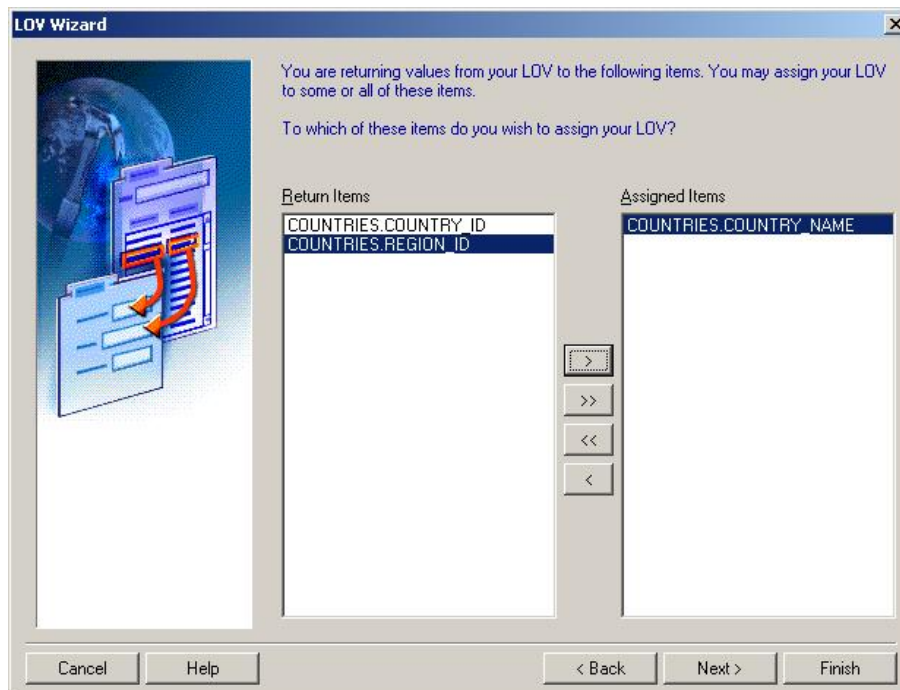


Figura 24. Elementos que Dispararán la Lista de Valores

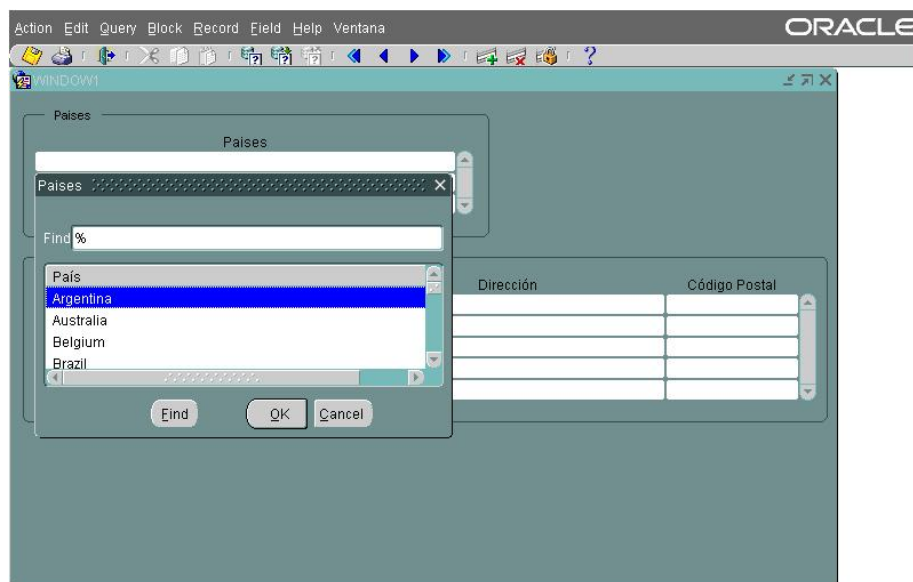



Figura 25. Lista de Valores en la Aplicación

- **Llamar la lista de valores usando el evento “When-Button-Pressed”**

La lista de valores asociada a un bloque puede ser llamada usando la función “show_lov(‘nombre de la lista’)”. Un ejemplo de esto puede ser usando un botón para efectuar esta función.

El botón puede crearse insertando un item en el mismo bloque asociado a la lista de valores o en un bloque de control y declarándolo en la ventana de propiedades del mismo, en el atributo “item type” de la sección general, como “Push Button”. Además, en la sección “Physical” verificamos que el atributo visible se encuentre en “Yes” y en el atributo “Canvas” le indicamos el canvas en que deseamos mostrarlo.

A continuación se ubica en la sección “Trigger” del botón que creamos en el navegador de objetos y hace clic en el botón  , lo que mostrará una ventana para escoger el evento que deseamos programar, en este caso “When-Button-Pressed”. (Véase figura 26)

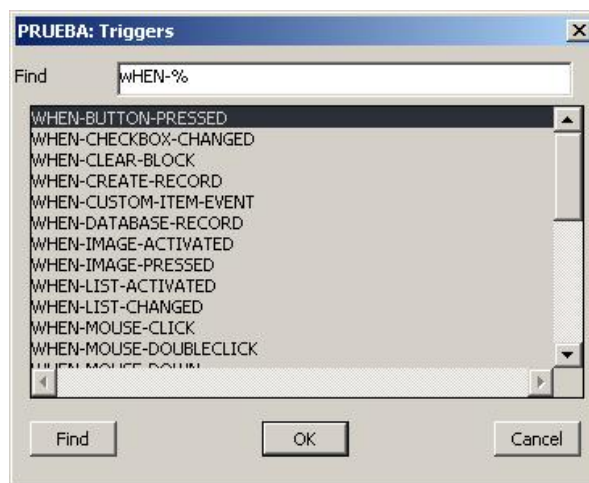


Figura 26. Lista de Triggers

Al hacer clic en el botón “OK” se nos mostrará la ventana para editar el código PL/SQL del evento (Véase figura 27). Allí escribimos:

Declare

```
lov_return boolean;
```

Begin

```
lov_return := show_lov('LOV9');
```

end;

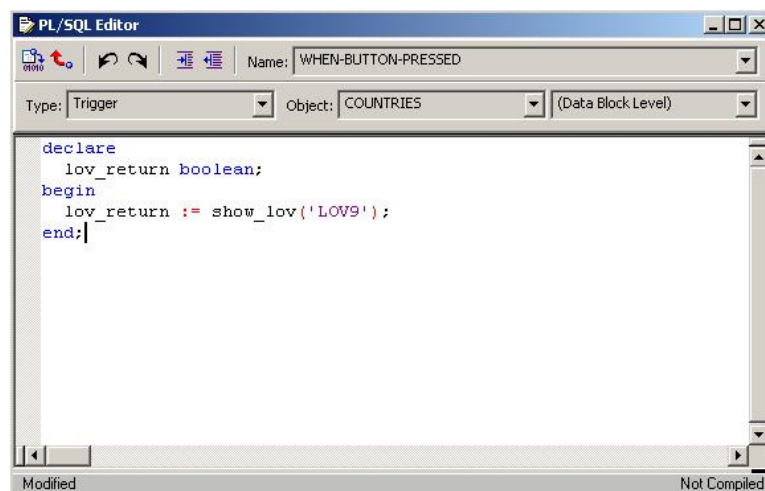



Figura 27. Programación del Evento

Con esto cuando presionemos con el mouse ese botón en la ventana, se desplegará la lista de valores que le indicamos en el evento.

2.4.2.7 Crear mensajes de alerta

Los mensajes de alerta son mecanismos para obligar al usuario a confirmar una determinada acción o informar sobre el efecto que la misma tendrá sobre el sistema. También son útiles para evitar errores por parte del usuario que pudieran perjudicar el comportamiento del sistema.

En Oracle 9i Forms Builder estas estructuras pueden ser de tres tipos: “Stop”, “Caution” y “Note”. “Stop” define un mensaje de alerta que detiene la acción del usuario para que este confirme que realmente desea efectuarla; “Caution” define un mensaje de advertencia sobre el efecto que tendrá la acción del usuario sobre el sistema o algún error asociado a la acción del usuario o durante la ejecución de la misma; y “Note” define un mensaje que informa al usuario sobre algo que debe tener en cuenta con relación a la acción que esta efectuando. Aunque se pueden definir estos tres tipos de mensajes de alerta, el comportamiento dentro de Oracle 9i Forms Builder es el mismo para todos y es el desarrollador el que asigna el significado de cada uno de acuerdo a lo anteriormente expuesto.

Para crear un mensaje de alerta, se ubica en la sección “Alerts” del módulo en que se encuentra trabajando en el navegador de objetos y hace clic en el botón . A continuación debe dirigirse a la ventana de propiedades del objeto. En la sección “General” puede editar el nombre de la alerta; en la sección “Functional” puede: asignarle un título en el atributo “Title”, escribir el mensaje que desea que se muestre en el atributo “Message”, definir el tipo de alerta en el atributo “Alert Style”, establecer los botones que se mostrarán y su valor y en el atributo “Default Alert Button” indicar el botón que estará seleccionado por defecto.

Una vez definido el mensaje de alerta debemos establecer que acción disparará el mensaje de alerta. Pongamos como ejemplo el evento “Exit”. Se desea que cuando el usuario efectuó la acción de salir del formulario se confirme si desea salir guardando los datos en la base de datos o salir sin guardarlos. Entonces

creamos en la sección “Triggers” del formulario en el navegador de objetos el evento “Key-Exit” que se dispara al salir del formulario. A continuación en la ventana del editor de PL/SQL escribimos:

```
DECLARE
    BOTON NUMBER;
    BLOQUE VARCHAR2(80) := :system.current_block;
BEGIN
    IF :SYSTEM.FORM_STATUS = 'CHANGED' THEN
/*La variable numérica denominada en el código “BOTON” recibe el valor del botón
que presione el usuario al desplegarse el mensaje de alerta*/
        BOTON := SHOW_ALERT ('ALERT1');
        IF BOTON = ALERT_BUTTON1 THEN
            COMMIT_FORM;
            DO_KEY ('exit');
        ELSIF BOTON = ALERT_BUTTON2 THEN
            clear_form(no_validate);
            exit_form(no_validate);
        END IF;
    ELSE
        exit_form(no_validate);
    END IF;
END;
```

Usando la función “SHOW_ALERT ('Nombre del mensaje de alerta)’” se llama al mensaje de alerta que se desea mostrar. Capturando el valor retornado por esta función al mostrarse el mensaje de alerta se podrán efectuar las acciones: salir guardando los datos y salir sin guardar.

2.4.2.8 Triggers

Los “Triggers” o disparadores son unidades de programa PL/SQL que se ejecutan cuando un determinado evento ocurre. Estos son clasificados dependiendo del nivel de operación en: Triggers del nivel del formulario, Triggers del nivel del bloque y Triggers del nivel del elemento del bloque. Esta clasificación es evidente en el navegador de objetos porque cada nivel posee una sección denominada “Triggers”.

Oracle clasifica los triggers en 8 categorías:

- Triggers de procesamiento del bloque: se accionan en respuesta a eventos que se presentan en el procesamiento a nivel del bloque.
- Triggers de eventos de interfase: se accionan en respuesta a eventos ocurridos en la interfase del formulario.
- Triggers de bloques maestro-detalle: obligan a la coordinación entre el bloque maestro y el bloque de detalles.
- Triggers de manejo de mensajes: se accionan en respuesta a mensajes de error o de información generados por el form builder.
- Triggers de navegación: se accionan cuando el usuario se desplaza entre formularios o elementos de un formulario
- Triggers de consulta: se accionan al ejecutarse una consulta.
- Triggers de transacciones: se accionan cuando el formulario interactúa con la base de datos.
- Triggers de validación: se accionan durante el proceso de validación de los datos de un elemento del formulario o un registro del bloque.

Algunos de los triggers más usados son los que se describen a continuación:

- **ON-CLEAR-DETAILS:** de los niveles del formulario o el bloque y de la categoría de bloques maestro-detalle, es usado para la coordinación entre bloques maestro y detalle cuando el form builder trata de borrar un registro de un bloque maestro en una relación maestro – detalle.
- **WHEN-NEW-FORM-INSTANCE:** del nivel del formulario y de la categoría de navegación, se acciona cuando se inicia un formulario.
- **ON-CHECK-DELETE-MASTER:** del nivel del formulario o el bloque y de la categoría de bloques maestro-detalle, se acciona cuando se da una orden de borrado de un registro en un bloque maestro. Detiene el borrado si se encuentran detalles para el registro.
- **PRE-INSERT:** del nivel del formulario o el bloque y de la categoría de transacciones, se acciona antes de insertar un registro en la base de datos.
- **PRE-QUERY:** del nivel del formulario o el bloque y de la categoría de consultas, se ejecuta antes de que una consulta sea enviada a la base de datos para llenar los registros de un bloque. Frecuentemente puede ser usado para alterar la cadena de consulta que sera enviada.
- **POST-QUERY:** del nivel del formulario o el bloque y de la categoría de consultas, se acciona después que una consulta se ejecuta para llenar los registros de un bloque.
- **WHEN-VALIDATE_ITEM:** del nivel del formulario, el bloque o el elemento del bloque y de la categoría de validación, ejecuta código PL/SQL para complementar el proceso de validación.

- **WHEN-BUTTON-PRESSED:** del nivel del formulario, el bloque o el elemento del bloque y de la categoría de eventos de interfase, se acciona cuando un botón es presionado.
- **PRE-TEXT-ITEM:** del nivel del formulario, el bloque o el elemento del bloque y de la categoría de navegación, ejecuta código PL/SQL tan pronto como el cursor se dirige al elemento del bloque.
- **POST-TEXT-ITEM:** del nivel del formulario, el bloque o el elemento del bloque y de la categoría de navegación, ejecuta código PL/SQL tan pronto como el cursor se mueve del elemento del bloque.

2.4.3 Reports Builder

Oracle 9i Reports Developer fundamenta la generación de reportes en consultas mediante las cuales se extraen los datos que se desean mostrar. Es una herramienta versátil que da la posibilidad de generar reportes dinámicamente por medio del paso de parámetros y la facilidad de construirlos de forma automática con base en una consulta.

Para facilitar la elaboración de consultas, scripts de creación de tablas e inserción de datos existen herramientas como SQL Navigator 4 y Toad VI. Estas herramientas poseen una ayuda visual mediante la cual se ofrece un esquema de las tablas a consultar, del cual se escogen los campos de los que se obtienen los datos. La facilidad que ofrecen estas herramientas es que minimizan los errores en que normalmente incurre el programador al construir consultas, además de que con base en las relaciones establecidas en la base de datos construyen automáticamente las condiciones de join necesarias.

2.4.3.1 Elaboración de reportes

Teniendo la consulta con la que se obtendrán los datos que se necesitarán para construir el reporte se podrá empezar a trabajar con Reports. Al iniciar el programa se mostrara una pequeña ventana en la que se ofrecen las opciones: “Usar el asistente de informes”, “Construir un reporte nuevo manualmente” y “Abrir un reporte existente”. (Véase figura 28)

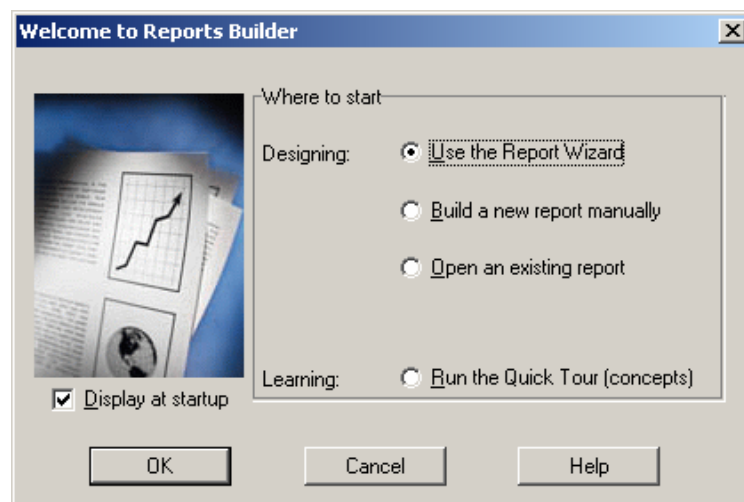


Figura 28. Ventana de Inicio de “Oracle 9i Reports Developer”

La opción “Usar el asistente de informes” permite la construcción automática de reportes a partir de una consulta y suministrando ciertos parámetros relacionados con la estructura del mismo; “Construir un reporte nuevo manualmente” crea el espacio de trabajo para la elaboración del reporte de forma manual, es decir, creando cada uno de los elementos que lo compondrán; y “Abrir un reporte existente” permite abrir un archivo que contiene un reporte ya elaborado. Por lo general se usará el asistente, así que a continuación se hará un análisis del uso de esta opción.

- Asistente para la Construcción de Reportes

Al seleccionar el asistente se muestra una ventana con nueve Tabs o pestañas en donde se suministran los parámetros que se requieren para la construcción del reporte. Aunque son pestañas, los parámetros se deben dar en orden pasando de una pestaña a otra oprimiendo el botón “Next >”. En caso de que se desee regresar a una pestaña anterior para modificar los parámetros que fueron dados en dicho formulario, se puede regresar con el botón “< Back” o haciendo clic en la parte superior de la ventana en la pestaña a la que se quiere retornar.

En el primer formulario que se presenta se debe indicar el tipo de reporte que se desea crear. Se puede escoger entre web, en papel o ambos. Lo normal será que se escoja ambos tipos de reportes para tener flexibilidad en el trabajo del mismo en relación con uno u otro tipo. (Véase figura 29)

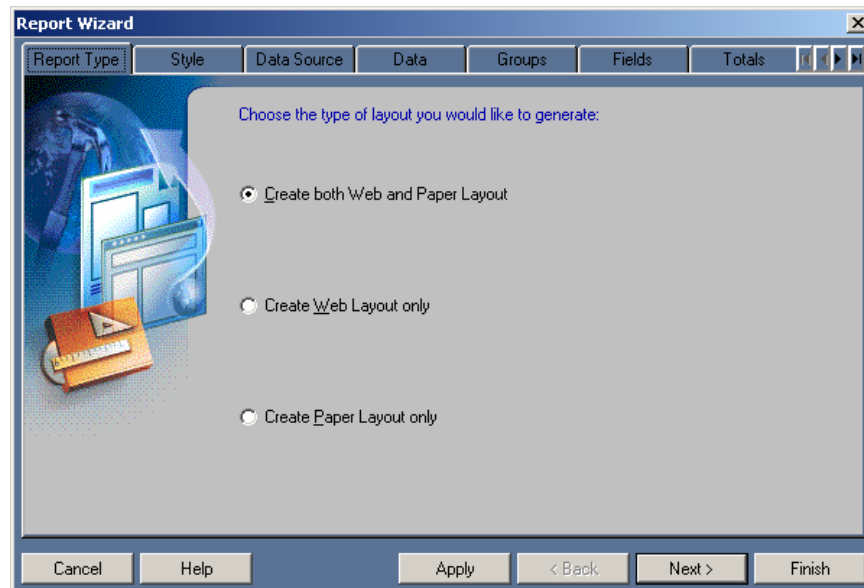


Figura 29. Tipo de Reporte

Oprimiendo el botón “Next” se continúa a la segunda pestaña con un formulario en el que se debe escoger el estilo del reporte y se le puede asignar un título. Los estilos entre los que se puede escoger son: Tabular simple, agrupado a la

izquierda y agrupado por encabezado, y Matricial simple y con grupo. La Diferencia entre uno tabular y uno matricial radica en que el tabular crece dinámicamente en un solo sentido, por columnas o por filas, mientras que el matricial crece dinámicamente en los dos sentidos, tanto en columnas como en filas. (Véase figura 30)

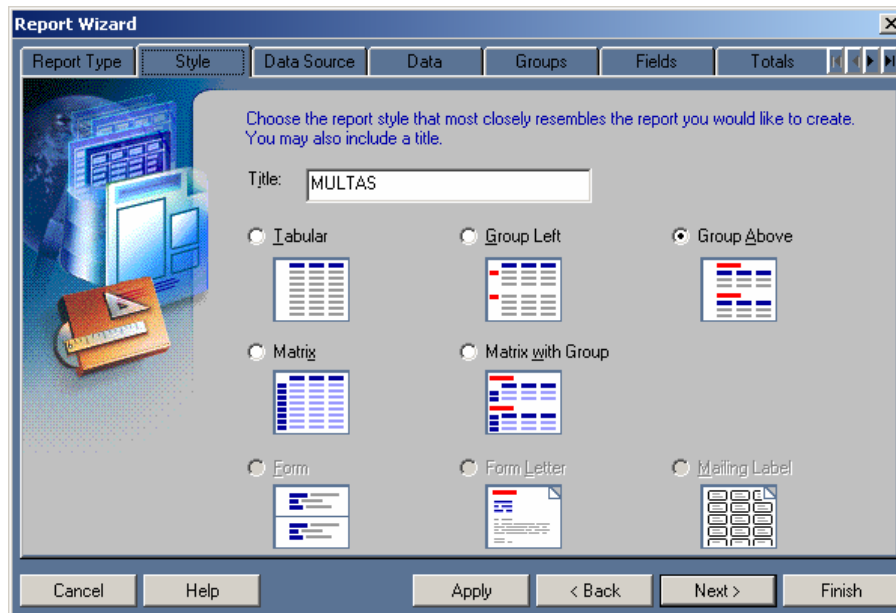


Figura 30. Estilo del Reporte

La siguiente pestaña es donde se ingresa la consulta a través de la cuál se extraerán los datos requeridos para la construcción del reporte. (Véase figura 31)

Se continúa entonces con la pestaña de definición de grupos en caso de que se halla escogido un estilo de reporte agrupado. En esta pestaña se definen solo los campos que serán encabezados para el reporte en diferentes niveles de agrupamiento. El nivel 1 corresponde al nivel más alto de agrupamiento. (Véase figura 32)

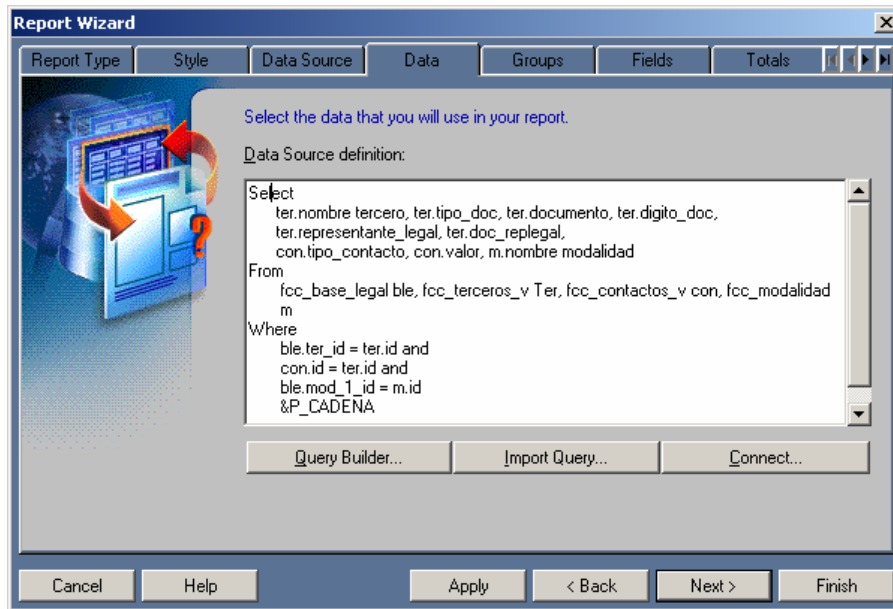


Figura 31. Query en el Asistente

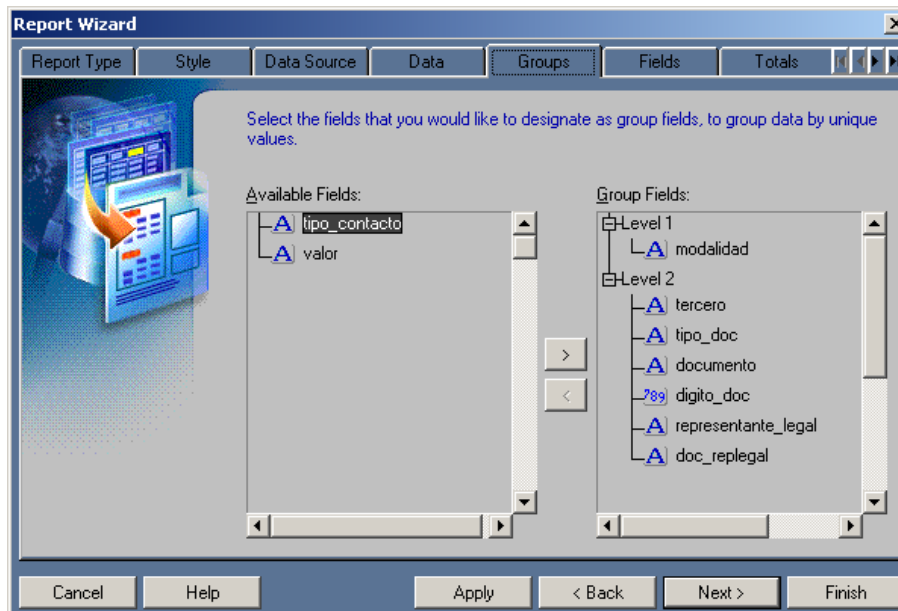


Figura 32. Definición de Niveles de Agrupamiento

El siguiente formulario es el de campos. En este se eligen los campos de la consulta que se desean mostrar adicionalmente a los que se han elegido como

encabezados en los niveles de agrupamiento. Los que no se encuentren en la columna de la derecha no se mostrarán en el reporte. (Véase figura 33)

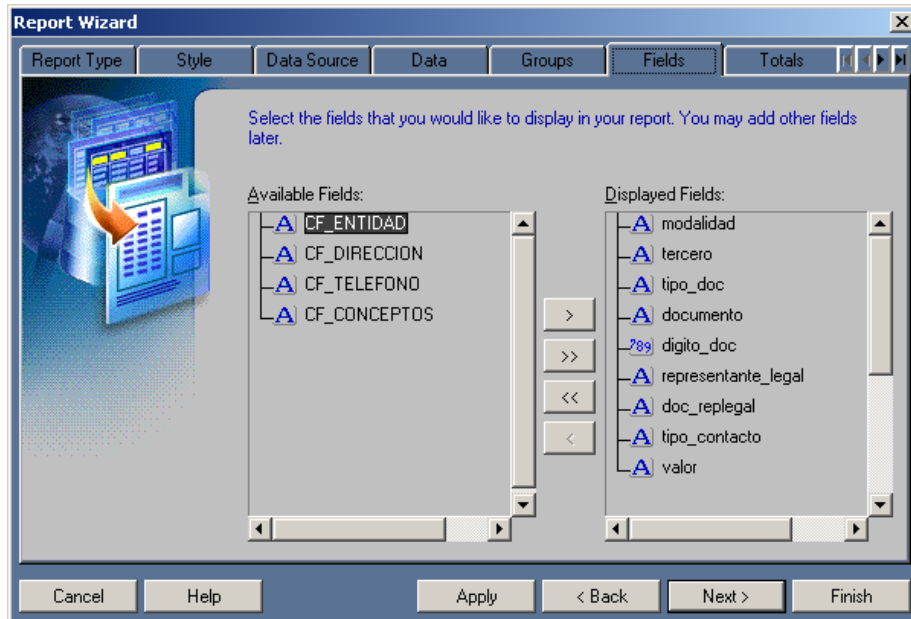


Figura 33. Campos de la Consulta que se Mostrarán

La siguiente pestaña es la de totales, la cual permite mostrar cálculos de suma, promedio, conteo, mínimo, máximo y porcentaje para los campos en los que se necesite mostrar este tipo de operaciones. (Véase figura 34)

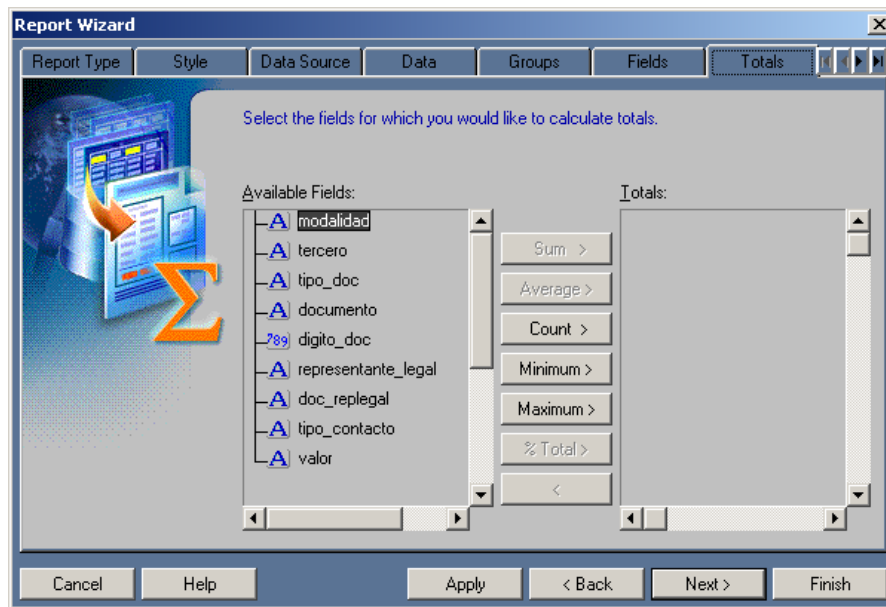


Figura 34. Selección de Campos para Cálculo en el Reporte

Sigue la pestaña de títulos para los campos que se van a mostrar. En esta de asignan títulos que serán mostrados en el reporte para los datos que se mostrarán. (Véase figura 35)

Finalmente se establece el tipo de plantilla que se escogerá para el reporte. Se puede escoger entre las opciones: Predefinida, Archivo o Ninguna plantilla. La primera genera un reporte un formato predefinido de colores y bien tabulado, mientras que la última arroja una plantilla organizada por defecto por el Reports. (Véase figuras 36 y 37)

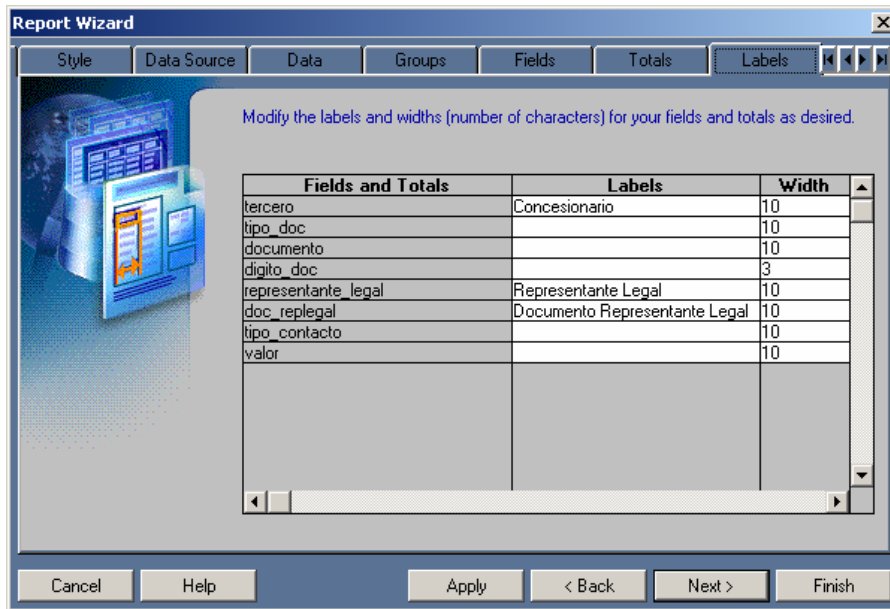


Figura 35. Pestaña de Labels

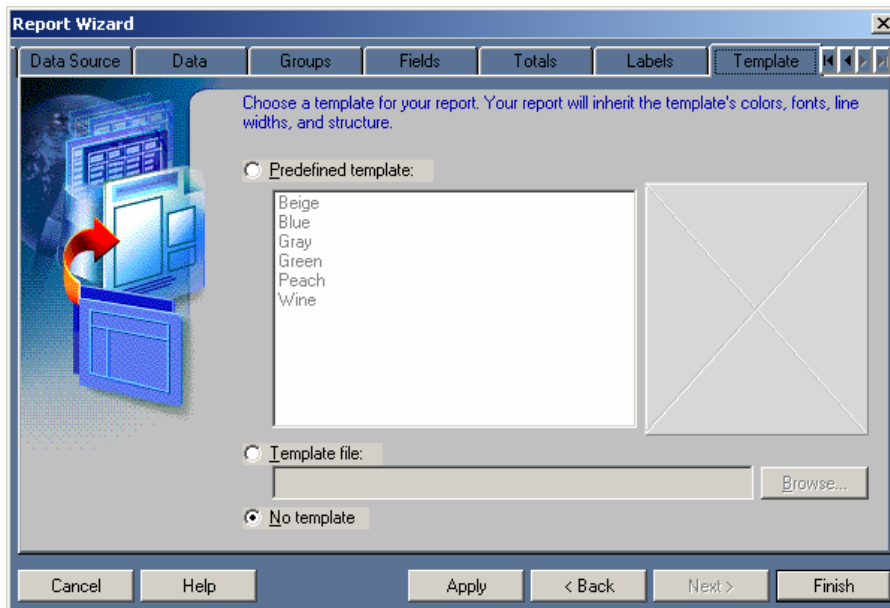


Figura 36. Plantilla del Reporte

First Name	Last Name	Phone Number	Job Title	Department Name	Salary	Hire Date
Michael	Hartstein	515.123.5555	Marketing Manager	Marketing	13000	17-FEB-96
Den	Raphaely	515.127.4561	Purchasing Manager	Purchasing	11000	07-DEC-94
John	Russell	011.44.1344.429268	Sales Manager	Sales	14000	01-OCT-96
Steven	King	515.123.4567	President	Executive	24000	17-JUN-87
Nancy	Greenberg	515.124.4569	Finance Manager	Finance	12000	17-AUG-94
Shelley	Higgins	515.123.8080	Accounting Manager	Accounting	12000	07-JUN-94
Total:					86000	
Average:					14333.333333333333	

Figura 37. Reporte Generado con una Plantilla Predefinida

Además, Reports ofrece la posibilidad de generar reportes dinámicos a través del paso de parámetros por parte del usuario. Estos se definen en el espacio de trabajo denominado "Paper Parameter Form". (Véase figura 38)

Figura 38. Formulario de Paso de Parámetros por Parte del Usuario

También Reports ofrece la facilidad de incluir gráficos en los reportes, que incluso pueden ser generados dinámicamente gracias a que basa la construcción del gráfico en atributos establecidos en un archivo XML.

2.5 FILOSOFÍA DE TRABAJO DE LA EMPRESA

La filosofía de trabajo al interior de Visión Ltda. está soportada en el Sistema de Gestión de Calidad que tiene implementado. A continuación se entra a considerar los lineamientos que definen la actividad de la empresa con base en la norma ISO 9000.

2.5.1 Política de Calidad

“Generar soluciones tecnológicas de software con calidad e innovación, siguiendo metodologías de clase mundial y utilizando tecnologías abiertas, cumpliendo con los tiempos pactados y acompañando al cliente en su operación y uso, buscando consolidar la imagen de solidez tecnológica de los productos y la empresa, ofreciendo valor agregado, ampliando la cobertura del mercado y la solidez financiera de la organización”

2.5.2 Misión

“Proveer soluciones informáticas y de gestión con parámetros de calidad, siguiendo estándares nacionales e internacionales, con un recurso humano capacitado, calificado y comprometido. Garantizando resultados excelente, efectivos y eficientes, que generan valor agregado a nuestros clientes, mediante la asimilación, adaptación e innovación tecnológica, que aseguren la sostenibilidad y desarrollo de la empresa.”

2.5.3 Visión

“Para el año 2008 Visión tendrá presencia y establecimiento de sus marcas a nivel nacional e internacional, fortaleciendo su posicionamiento en estos mercados.”

3 PRÁCTICA EMPRESARIAL EN VISIÓN LTDA.

3.1 INICIACIÓN

3.1.1 Inducción a la Filosofía de Trabajo de la Empresa

Actividad realizada por la empresa en función de que el desarrollo del proyecto se ajuste a las necesidades de esta y que los nuevos miembros del equipo de trabajo se integren al tener objetivos comunes. Está actividad fue realizada al estudiante en práctica durante los dos (2) primeros días de su labor dentro de la empresa, al final de la cual se efectuó una evaluación para determinar el grado de conocimiento adquirido por el estudiante.

3.1.2 Capacitación en Designer, Forms y Reports

3.1.2.1 Designer

La capacitación realizada en el manejo de la herramienta “Oracle9i Designer” un abarcó un período de un (1) día y comprendió las siguientes temáticas:

- Creación del modelo entidad relación
- Paso del modelo de datos lógico al diseño físico de la base de datos
- Realización de un proceso de Ingeniería Inversa

3.1.2.2 Forms

La actividad de capacitación realizada por parte de uno de los ingenieros de la empresa en el manejo de la herramienta “Forms Developer 9i” abarcó un período de un (1) día y comprendió las siguientes temáticas:

- Reconocimiento de los elementos que componen los módulos en el manejo de la herramienta
- Manejo del asistente para la construcción de bloques
- Manejo del asistente para la definición de los elementos del formulario
- Manejo de Triggers
- Creación de formularios con bloques maestro – detalle
- Creación de formularios con bloques de control
- Creación de formularios con listas dinámicas
- Creación de mensajes de alerta en los formularios
- Vinculación de reportes a los formularios

Posteriormente se paso a realizar unos formularios de práctica para determinar el grado de aprendizaje en la capacitación.

3.1.2.3 Reports

Se realizó una actividad de capacitación por parte de uno de los ingenieros de la empresa en el manejo de la herramienta “Reports Developer 9i” que abarcó un

período de un (1) día. La actividad tuvo como objetivo instruir en la generación de reportes dinámicos y con características tales como:

- Tabulares sencillos o por grupo
- Matriciales, en los que las columnas y las filas crecen dinámicamente y se realizan referencias cruzadas.
- Gráficos en forma estática y dinámica

Posteriormente se paso a realizar unos reportes de práctica para determinar el grado de aprendizaje en la capacitación.

3.1.3 Revisión del Modelo GCI

El modelo GCI (Gestión Corporativa Integral) forma parte de los productos de Visión y permite realizar la gestión de los procesos internos de una empresa junto con los documentos que estos implican. En vista de que una de las políticas de la empresa en cuanto al desarrollo de software es la compatibilidad entre las aplicaciones, para el diseño del modelo del sistema de soporte al cliente se tuvo en cuenta la estructura de algunas de las entidades definidas en él junto con el modelo establecido en CRM, otro de los productos ofrecidos por la empresa. Tanto el modelo GCI como el de CRM cuentan con derechos de autor y patrimoniales registrados a nombre de Visión Ltda.

Las entidades definidas como necesarias de estos modelos se escogieron en base a los siguientes requerimientos:

- Las solicitudes son realizadas por clientes que tienen contrato establecido con la empresa.

- Por cada solicitud se debe diligenciar un formulario en el que se establezca la persona que realizó la solicitud, sobre que producto se realiza, a que contrato corresponde y demás detalles específicos relacionados con la solicitud como tal.

Las entidades con sus atributos definidas en los modelos GCI y CRM y tomadas para el sistema fueron:

- Municipio
- Tipo de Relación Comercial
- Vigencia
- Persona
- Tercero
- Área
- Cargo
- Producto
- Oportunidad (proyecto)
- Contrato

Este análisis y la posterior definición del modelo de datos tuvieron en conjunto una duración de dos (2) semanas.

3.1.3.1 Definición del modelo de datos

Los requerimientos generales dados para definir el modelo de datos del sistema fueron:

- Una solicitud esta asociada a un producto y contrato y consta de: la solicitud como tal, la fecha y hora en que fue realizada, una prioridad, la persona que la realiza, la persona que la atiende, las personas que la resuelven y los módulos del producto implicados.
- El sistema debe llevar un seguimiento de la solicitud a través de: un estado, fecha de asignación a las personas que la resuelven, fecha de solución deseada, fecha de entrega de solución por parte de quien resuelve, fecha de solución final por parte y un resultado.
- Un contrato puede implicar la elaboración de varios productos y un producto puede estar implicado en varios contratos.
- Los productos (sistemas) poseen varias facilidades
- Las facilidades pueden tener asociadas varias opciones
- Una opción puede incluir varias opciones
- Una opción puede tener varios módulos.
- Los módulos pueden ser realizados con varias herramientas de desarrollo.
- Existen solicitudes que pueden ser atendidas mediante lecciones aprendidas, es decir, seguir un conjunto de pasos ya definidos dentro del sistema.
- Una lección aprendida puede estar asociada a varios módulos y un módulo puede estar implicado en varias lecciones.
- El sistema debe manejar los datos de las lecciones aprendidas elaboradas tales como: fecha de elaboración, fecha de aprobación, un nombre, el detalle de la lección, quién la elaboró y quién la aprobó.

- Al momento de realizarse una solicitud se debe poder establecer si ya existe una lección aprendida que le de solución.
- Deben establecerse las actividades y subactividades involucradas por proyecto y producto junto con los responsables de cada una de ellas.
- Debe realizarse un seguimiento de las actividades y subactividades de los proyectos a través de la fecha de inicio, fecha de finalización definida y fecha de finalización real.

En base a lo anterior se definieron las siguientes entidades junto con sus atributos, integrándolas a las ya establecidas por los modelos GCI y CRM:

- Facilidad
- Opción
- Módulo
- Herramienta
- Solicitud
- Lección
- Actividad

3.1.4 Definición de las Opciones a Ajustar y Elaborar

En la atención de las solicitudes realizadas por los clientes de Visión Ltda., es necesario establecer la relación existente entre las partes, qué persona vinculada al cliente la realiza y quién hace la recepción de la misma. Esto nos indica que debemos recopilar una cantidad considerable de datos del cliente y de la relación con el mismo para poder dar trámite a una solicitud.

Las solicitudes se realizan sobre los productos que ofrece la empresa, los cuales corresponden a las aplicaciones desarrolladas. Puesto que las solicitudes pueden ser realizadas sobre el detalle de la aplicación, se requiere considerar de manera global los partes que componen una aplicación tales como: facilidades, opciones, módulos, etc.

Finalmente las solicitudes pueden resultar en que se realicen las siguientes acciones a fin de atenderlas:

- Realizar codificación a fin de corregir un problema o ajustar la aplicación a un determinado requerimiento.
- Elaborar una lección, es decir, un instructivo de pasos para indicar al usuario la forma de resolver la necesidad expuesta.
- Suministrar al usuario una lección elaborada que le permite solucionar su inquietud.

Adicionalmente, y a fin de lograr una buena calidad de los productos y servicios que ofrece la empresa, es necesario realizar la gestión de las solicitudes, es decir, todo lo que involucra la planificación y control de la atención de las solicitudes. Esto implica, de acuerdo al conjunto de acciones requeridas para atender el requerimiento, asignar personal dentro de la empresa para solucionar la necesidad expuesta por el cliente y llevar un control de las responsabilidades otorgadas a las personas relacionadas con la atención de una determinada solicitud.

El tiempo abarcado en el análisis de las opciones que compondrán el sistema fue de dos (2) semanas.

3.1.5 Evaluación de las Herramientas Oracle Developer 6i y Oracle Developer 9i

En vista de que la empresa dispone para el desarrollo de software de las herramientas integradas Oracle Developer 6i y Oracle Developer 9i, se realizó una evaluación de las mismas con el fin de determinar la más ajustada a las condiciones de la empresa.

3.1.5.1 Oracle Developer 6i

Oracle Developer 6i integra un conjunto de herramientas para la construcción de aplicaciones que cubren todas las fases del ciclo de vida del desarrollo.

Reúne características similares a las de Oracle Developer 9i expuestas en este documento con relación a Designer, Forms y Reports, aunque por ser una versión anterior carece de algunas facilidades.

En cuanto a la implantación, el entorno de ejecución de las aplicaciones puede ser cliente/servidor o web. El primero requiere de un servidor que aloje los módulos de la aplicación y en el equipo cliente los runtime (archivos de ejecución) de la misma para ejecutarlos. El segundo exige valerse de un servidor web.

3.1.5.2 Oracle Developer 9i

Oracle 9i Developer Suite (Oracle9iDS) es un completo e integrado conjunto de herramientas que permite:

- Construir aplicaciones de internet y servicios web de transacciones usando Java y XML.
- Implementar inteligencia de negocios en las aplicaciones sin el costo de herramientas adicionales.

Ofrece soporte completo para el ciclo de vida del desarrollo, abarcando modelado, codificación, pruebas y validación del código. Brinda también facilidades para el trabajo en equipo, control y gestión del desarrollo para administrar la continua evolución de las aplicaciones.

Utiliza como servidor de aplicaciones Oracle 9i Application Server debido a que el entorno de las aplicaciones desarrolladas con esta herramienta es web. Oracle 9iAS incorpora una amplia gama de servicios y funcionalidades que se pueden utilizar para implementar aplicaciones en un entorno distribuido con la fiabilidad y escalabilidad necesarias, siendo además el único servidor de aplicaciones que permite acelerar las operaciones de las aplicaciones Web y que dispone de capacidades de Business Intelligence. El servidor de aplicaciones Oracle 9iAS tiene además una serie de servicios integrados, entre los que merece destacar:

- Servicio de Caché: Oracle 9iAS, proporciona memoria caché de aplicaciones dentro del contexto de una completa plataforma de Internet. Oracle 9i Caché mantiene la información reciente cercana a la aplicación, acelerando el acceso a datos desde un navegador.
- Servicios Web/HTTP: Como consecuencia de haber adoptado el servidor web Apache, Oracle iAS asegura que las aplicaciones Web pueden incrementar su escalabilidad y fiabilidad.
- Servicios Business Intelligence: Las funciones de Business Intelligence ha sido tradicionalmente consideradas como un software especializado, separado de la plataforma de Internet. Con Oracle 9iAS, Oracle incluye capacidad de Business Intelligence en un servidor de aplicaciones.
- Servicios de Formularios y Reportes: Oracle 9iAS también actúa como el servidor de implantación para las aplicaciones que utilizan formularios y reportes de datos realizados en Developer. Estos servicios optimizan las

aplicaciones para minimizar el tráfico en la red y aseguran la escalabilidad para grandes aplicaciones empresariales.

- Servicios de Integración: La plataforma de Internet de Oracle proporciona servicios de integración accesibles desde Oracle 9iAS para conectar sitios Web y servicios basados en las últimas tecnologías de software o en sistemas heredados, incluyendo soporte extensible a XML.

Valiéndose de un servidor en el que se alojen los módulos de la aplicación y teniendo configurado el Oracle 9iAS se puede acceder a la aplicación desde cualquier equipo de la red a través de cualquier navegador de internet inicializando el OC4J, teniendo en cuenta la limitación de las licencias de usuario.

3.1.5.3 Conclusiones de la evaluación

En vista de que la empresa desea un sistema que satisfaga las necesidades de la empresa con un menor costo se escogió como herramienta de desarrollo Oracle Developer 6i, porque Oracle Developer 9i exigía un servidor adicional con Oracle 9iAS y costos adicionales por concepto de licencias.

3.2 ELABORACIÓN

3.2.1 Diseño Detallado de las Opciones

Las opciones a elaborar se establecieron con la orientación del Tutor de la práctica durante un período de un (1) mes.

- **Opción de gestión de productos**

La opción de Gestión de Productos contiene cuatro (4) formularios en los cuales se ingresan los datos correspondientes al manejo de los productos software elaborados por la empresa.

El primer formulario permite definir los productos y servicios ofrecidos por la empresa, dentro del sistema de gestión se establece esta opción que permite registrar los datos correspondientes y el diseño de la cual fue ajustado a las necesidades de la empresa. (Véase figura 39)

The image shows a screenshot of a software application window titled 'SISTEMA GCI PARA EL SISTEMA DE GESTION DE CALIDAD DE VISION'. The main window is titled 'GESTION DE PRODUCTOS' and contains a tabbed interface with four tabs: 'SISTEMA', 'FACILIDAD', 'HERRAMIENTA', and 'OPCIÓN'. The 'OPCIÓN' tab is currently selected. Inside this tab, there is a form titled 'Datos del Producto'. The form contains the following fields and controls:

- Nombre: Text input field.
- Nombre Corto: Text input field.
- Versión: Text input field.
- Descripción: Large text area.
- Valor Unitario: Text input field.
- Porcentaje de Descuento: Text input field.
- Fecha Vencimiento del Descuento: Text input field.
- Código: Text input field.
- Tecnología: Dropdown menu.
- Tiene Derecho de Autor: Check box.

Figura 39. Formulario de Datos del Producto

Los tres formularios siguientes permiten establecer en el sistema una descripción detallada de cada producto (Véase figuras 40, 41 y 42)

SISTEMA GCI PARA EL SISTEMA DE GESTION DE CALIDAD DE VISION

GESTION DE PRODUCTOS

SISTEMA FACILIDAD HERRAMIENTA OPCION

Facilidades

Sistema

Nombre	Descripción

Figura 40. Formulario de Datos de la Facilidad

SISTEMA GCI PARA EL SISTEMA DE GESTION DE CALIDAD DE VISION

GESTION DE PRODUCTOS

SISTEMA FACILIDAD HERRAMIENTA OPCION

Datos de la Herramienta

Nombre	Fabricante	Versión	Sale del mercado	Soporte hasta	Descripción

Figura 41. Formulario de Datos de la Herramienta

The screenshot shows a software interface for 'SISTEMA GCI PARA EL SISTEMA DE GESTION DE CALIDAD DE VISION'. The window title bar includes standard OS controls. Below the title bar is a menu bar with four items: 'SISTEMA', 'FACILIDAD', 'HERRAMIENTA', and 'OPCIÓN'. The 'OPCIÓN' menu is currently selected. The main content area is organized into several sections:

- Sistema:** A dropdown menu.
- Facilidades:** A list box with a scroll bar.
- Opciones:** A table with three columns: 'Opción', 'Observaciones', and 'Incluye las Opciones'. Each column has a list box with a scroll bar.
- Detalle del Modulo:** A section containing several input fields:
 - 'Nombre del Módulo': text input.
 - 'Persona que elaboró': dropdown menu.
 - 'Persona que aprueba': dropdown menu.
 - 'Descripción': large text area.
 - 'Archivo': text input.
 - 'Fecha de Elaboración': text input.
 - 'Fecha de Aprobación': text input.
- Herramienta:** A list box with a scroll bar.
- Comentario:** A text input field.

Figura 42. Formulario de Datos de la Opción

- **Opción de asignaciones por producto y proyecto**

La opción de Asignaciones por Producto y Proyecto contiene tres (3) formularios en los cuales se ingresan los datos correspondientes a las actividades y personas involucradas por proyecto junto con las responsabilidades asignadas y su seguimiento (Véase figuras 43, 44 y 45).

SISTEMA GCI PARA EL SISTEMA DE GESTION DE CALIDAD DE VISION

ASIGNACIONES POR PRODUCTO Y PROYECTO

DETALLE DEL PROYECTO RESPONSABLES SEGUIMIENTO

Proyecto

Contrato Oportunidad

Detalle del Proyecto

Producto

Actividades

Actividad

Subactividades

Subactividad

Figura 43. Formulario de Datos del Proyecto

SISTEMA GCI PARA EL SISTEMA DE GESTION DE CALIDAD DE VISION

ASIGNACIONES POR PRODUCTO Y PROYECTO

DETALLE DEL PROYECTO RESPONSABLES SEGUIMIENTO

Proyecto

Contrato Oportunidad

Asignados al Proyecto

Responsable

Figura 44. Formulario de Datos de los Involucrados en el Proyecto

The screenshot shows a software interface for project management. At the top, the title bar reads 'SISTEMA GCI PARA EL SISTEMA DE GESTION DE CALIDAD DE VISION'. Below the title bar is a menu bar with three tabs: 'DETALLE DEL PROYECTO', 'RESPONSABLES', and 'SEGUIMIENTO'. The 'SEGUIMIENTO' tab is selected. The main area contains several sections:

- Sistema:** Includes three dropdown menus for 'Producto', 'Contrato', and 'Proyecto'.
- Actividades Globales:** Includes a text input field for 'Actividad'.
- Responsables:** Includes a list box for 'Responsable' with a scroll bar.
- Actividades Asignadas:** Includes a table with the following columns: 'Subactividad', 'Fecha Asignación', 'Fecha Terminación', and '% Dedicación'. The table has four rows and a vertical scroll bar on the right.

Figura 45. Formulario de Datos del Seguimiento del Proyecto

- **Opción de soporte al cliente**

La opción de Soporte al Cliente reúne los datos núcleo del sistema de soporte. Contiene dos (2) formularios en los cuales se ingresan los datos correspondientes a las solicitudes realizadas por los clientes y las lecciones elaboradas. (Véase figuras 46 y 49).

SISTEMA GCI PARA EL SISTEMA DE GESTION DE CALIDAD DE VISION

SOPORTE AL CLIENTE

SOLICITUD | **LECCION**

Datos de la Solicitud

Fecha y Hora: Automática | Atiende: [dropdown]

Requerimiento: [text area]

Sistema: [dropdown] | Contrato: [dropdown] | Proyecto: [text area]

Solicita

Nombre: [dropdown] | Área: [text area]

Detalle

Opción	Necesidad	Modulo	
[text area]	[text area]	[text area]	Seguimiento
[text area]	[text area]	[text area]	Seguimiento
[text area]	[text area]	[text area]	Seguimiento

[button: Lecciones Relacionadas]

Control y Seguimiento

Fecha Deseada: [text area] | Fecha Asignada: [text area]

Fecha Entrega: [text area] | Fecha Solución: [text area]

Prioridad: [dropdown] | Estado: [dropdown]

Observaciones: [text area]

Figura 46. Formulario de Datos de la Solicitud

El formulario de la solicitud incluye un detalle que asocia el requerimiento a un módulo específico del sistema. El detalle de la solicitud incluye dos (2) botones: seguimiento y lecciones relacionadas. El primero conduce a una ventana en la que se registran los datos de seguimiento de la solicitud por módulo (Véase figura 47). El segundo botón muestra en una ventana las lecciones que pueden atender la solicitud en base al producto involucrado en general o los módulos relacionados (Véase figura 48).

Figura 47. Ventana de Seguimiento de Atención del Requerimiento

Figura 48. Ventana de Lecciones Relacionadas

Figura 49. Formulario de Datos de las Lecciones

3.2.2 Realización del Plan de Pruebas

El plan de pruebas, cuya elaboración se extendió por un período de dos (2) semanas, define las actividades a desarrollarse durante el proceso de pruebas del sistema. Cada una de estas actividades se encuentra enmarcada en un caso o escenario de prueba teniendo en cuenta los módulos que serán incluidos, los recursos que serán empleados, así como los participantes de la misma.

Las pruebas deben realizarse con datos reales, por lo que la preparación de estos datos es responsabilidad de los usuarios. Debe tomarse en cuenta que los datos suministrados deben ser representativos en el sistema.

Como parte de la ejecución de las pruebas, se deben incluir aspectos de concurrencia. Este proceso implica que la misma aplicación puede ser utilizada por dos o más usuarios al mismo tiempo, por lo tanto requiere de al menos dos usuarios ubicados en computadores diferentes con el mismo detalle de la prueba a ejecutar, de tal forma que realicen la prueba al mismo tiempo, verificando al final de la misma los resultados esperados y la correcta realización de los procesos por parte del sistema.

Las pruebas estipuladas deben referirse a los distintos procesos. Durante la ejecución de las mismas se podrán probar las consultas y reportes como complemento a la verificación de los resultados de los procesos.

El plan de pruebas debe ser consistente y completo de conformidad a los requerimientos planteados por los usuarios y definidos por los documentos resultantes de las etapas de análisis y diseño del sistema. Para la elaboración del plan de pruebas se debe tomar en cuenta tanto los requerimientos planteados por los usuarios directos, como también los planteados por los usuarios indirectos.

Dentro de las pruebas, se debe validar que los campos definidos en las diferentes ventanas acepten valores propios del campo, como por ejemplo que los campos numéricos solo acepten números ya sean positivos o negativos, según corresponda. Además, se debe evaluar el rendimiento o tiempo de respuesta de cada una de las opciones del sistema así como también que cumpla con las condiciones apropiadas para que la interfaz sea amigable al usuario y evite al máximo errores por parte del usuario.

3.2.2.1 Casos de prueba

Se definieron los siguientes casos de prueba globales:

- Definición de tipos de relación comercial
- Definición de vigencias
- Definición de terceros en el sistema
- Asignar relaciones comerciales al tercero
- Registro de los datos de personas involucradas en el sistema
- Registro de productos
- Registro de oportunidades, incluyendo datos de la propuesta, los productos y los costos asociados a la oportunidad
- Registro de contratos
- Registro de datos de la facilidad
- Registro de datos de la herramienta
- Registro de datos de la opción

- Registro de datos del proyecto
- Registro de involucrados en el proyecto
- Registro de datos del seguimiento del proyecto
- Registro de datos de la solicitud
- Registro del seguimiento de atención del requerimiento
- Consulta de las lecciones relacionadas con el requerimiento
- Registro de datos de las lecciones

3.2.3 Elaboración de Reportes

En el desarrollo de sistemas de gestión y en general de sistemas de información son fundamentales los reportes para la toma de decisiones, especialmente en el nivel de la administración de una organización.

Establecer adecuadamente los reportes que se requerirán es parte fundamental a fin de lograr satisfacer las necesidades del cliente. En este proceso ayuda el considerar las siguientes preguntas:

- ¿Qué datos desea el usuario que se muestren y en que prioridad?
- ¿Qué tan detallado se desea el reporte?
- ¿En qué forma se desea presentar los datos visualmente? ¿Tabularmente o se incluirán gráficos?
- Si el usuario modifica los datos usando un formulario e inmediatamente después desea generar un reporte, ¿el reporte deberá abarcar dichos cambios?

- ¿Estará el reporte incluido dentro de un formulario?
- ¿Cómo desean generar el reporte? ¿Impreso, en html, pdf o cualquiera de estos?
- ¿Requerirá que el usuario suministre algunos parámetros específicos para la generación del mismo?
- ¿Existe un estándar para los reportes que se elaboran en la organización?
- Para los reportes en web, ¿serán estas páginas estáticas o busca generar sitios web dinámicos?

La empresa requirió la elaboración de múltiples reportes, los cuales fueron de tipo tabular y matricial y algunos incluyendo gráficos, y su codificación abarcó un período de un (1) mes. Previo a su elaboración fue necesario considerar documentos existentes que sirvieron de guía para la elaboración de los mismos y tener estrecha comunicación con el ingeniero encargado del proyecto, con el fin de definir claramente los requerimientos de cada reporte y realizar los ajustes necesarios posteriormente a las entregas parciales y funcionales de los mismos.

Adicionalmente, se realizó un estudio de la estructura del archivo XML que se crea para los gráficos en el reporte, con el fin de que el usuario pudiera definir dinámicamente el tipo de gráfico y algunos de sus atributos. (Véase anexo A)

3.3 CONSTRUCCIÓN

3.3.1 Codificación de Opciones

Se elaboraron múltiples formularios, los cuales requirieron bloques maestro y detalle, bloques de control, canvas dentro de otro canvas, canvas del tipo tab, ventanas emergentes, listas de valores y en general el manejo de todos los

elementos básicos de los formularios. Esta labor comprendió un período de dos (2) meses, durante los cuales se incluyen las actividades de ajustes a los módulos elaborados que requieren de labores de programación.

Previo a la elaboración de los formularios fue necesario establecer bien los requerimientos del sistema, además de entender la filosofía de desarrollo establecida dentro de la empresa, para que la aplicación resultante fuera satisfactoria para el cliente y acorde a las políticas de desarrollo establecidas por la empresa.

La codificación se soportó en los diseños realizados de cada una de las interfases definidas con el ingeniero encargado del proyecto y los usuarios del sistema. Además, para lograr que estas fueran amigables al cliente se establecieron las facilidades que deberían tener como, por ejemplo, el cargue automático de datos con base en unos parámetros dados.

Las herramientas utilizadas en la elaboración de los formularios y reportes del sistema fueron Forms Builder y Reports Builder de Oracle9i Developer Suite, y por ende el lenguaje de programación manejado fue PL/SQL. Además, se utilizó la herramienta “SQL Navigator 4” que facilitó la elaboración de consultas, scripts de creación de tablas e inserción de datos.

3.3.2 Validación Unitaria de las Opciones

A medida que se codificaban las opciones, se validaba la correspondencia de cada una de ellas con los requerimientos establecidos. En vista de que las herramientas utilizadas para el desarrollo definen con base en la base de datos establecida los tipos de datos de cada una de las estructuras de entrada y salida en los formularios asociadas a la misma, la preocupación principal se centraba en tener un modelo de datos apropiado.

Además, durante la programación de cada uno de los componentes de la aplicación, se verificaba que los resultados fueran los esperados y que el tiempo de respuesta fuera el apropiado. El análisis del rendimiento de las opciones debía considerar las consultas incluidas en el código y las rutinas de código implementadas.

3.3.3 Ajuste de las Opciones

Posterior a la elaboración de las opciones, el programador realiza las verificaciones necesarias a fin de que el producto a entregar sea satisfactorio para el usuario. Al realizar estas nota asuntos que paso por alto durante el desarrollo y debe hacer ajustes.

Los ajustes realizados a las opciones estuvieron relacionados con la conformidad de las interfases al estándar definido para las mismas, el rendimiento de las consultas elaboradas y facilidades adicionales para hacer la interfaz amigable al usuario o para hacerla acorde con la lógica de uso de la misma.

3.3.4 Pruebas Funcionales y de Operatividad

Con base en los escenarios de prueba definidos en el plan de pruebas se realizaron las pruebas funcionales y de operatividad de las opciones del sistema con los mismos usuarios.

Estas pruebas fueron orientadas por el ingeniero de pruebas designado para el proyecto y en estas se verificó que se realizara un registro y consulta apropiada de los datos del sistema y que brindara las facilidades necesarias a los usuarios para un uso apropiado del mismo.

De estas pruebas se tomaron las observaciones realizadas por los usuarios junto con los errores reportados durante las mismas.

3.3.5 Ajustes Finales

Las observaciones y los errores presentados durante la ejecución del plan de pruebas conducen a realizar ajustes con el fin de satisfacer completamente los requerimientos establecidos durante la fase de iniciación.

En vista que el diseño realizado del modelo de datos y las opciones fue ajustado al análisis realizado en el proyecto los ajustes finales resultan ser pocos y no afectan la estructura en general del sistema. Además, la política de mantener estrecha comunicación con el cliente, orientada por la metodología Extreme Programming, hace que cualquier mala interpretación de parte del desarrollador se corrija rápidamente sin que ocurra un efecto multiplicativo del error en la estructura del sistema.

Así pues, los ajustes se redujeron a errores por parte del programador al pasar por alto algunas situaciones o casos en la utilización de la aplicación por parte de los usuarios y que solo requirieron modificaciones mínimas en el código y en casos más complejos la variación de la estructura de algún formulario.

3.4 TRANSICIÓN

3.4.1 Implantación de las Opciones

Para la implantación del sistema se utilizó un servidor existente en la empresa para alojar la base de datos y los módulos de la aplicación.

La empresa disponía de una infraestructura de red con los suficientes equipos para cada uno de los usuarios del sistema. Estos equipos, por pensarse para las labores de desarrollo de software de la empresa, presentan las siguientes

características: Procesador Intel Pentium 4 de 2.4 GHz, 1 Gbyte de memoria RAM y 70 Gbytes de disco duro.

La herramienta de bases de datos utilizada fue Oracle9i Enterprise Edition Release 9.2.0.6.0.

El entorno de la aplicación corresponde a uno cliente/servidor, razón por la que se escogió la herramienta de desarrollo Oracle Developer 6i mediante la cual a los equipos cliente solo se necesitó instalarles los runtime (archivos de ejecución) para la ejecución de los módulos de las aplicaciones desde el servidor.

3.4.2 Ajustes

En vista de los detalles surgidos en la implantación del sistema se hacen los ajustes necesarios a fin de que su funcionamiento sea el óptimo y a satisfacción del cliente.

3.4.3 Documentación

La documentación, aunque se incluye en la fase final, es un proceso que se lleva a cabo durante todo el desarrollo del proyecto.

Esta se inicia en la definición del modelo de datos al utilizar herramientas como Designer 9i que permiten escribir una descripción de cada uno de los elementos definidos en el diseño lógico y físico de la base de datos.

Se continúa en el desarrollo con los comentarios incluidos en el código que explican las rutinas programadas, junto con los textos emergentes que se muestran cuando el usuario navega por la aplicación y la ayuda del sistema.

Finalmente la documentación se completa con el manual de usuario y la documentación técnica del sistema, cuya recopilación comprendió un tiempo de 1 semana.

3.4.4 Entrega Final de las Opciones y su Documentación

Al terminar el presente proyecto se realizó la entrega de las opciones elaboradas y la documentación del sistema y la empresa certificó dicha labor por medio de una carta en la que expresa la satisfacción con el trabajo realizado. (Véase anexo B)

CONCLUSIONES

En aplicación de los principios de “Mejora Continua” y “Enfoque al Cliente” del Sistema de Gestión de Calidad, uno de los objetivos de Visión es brindar un mejor soporte a los usuarios de sus productos. A través de una aplicación de gestión de las solicitudes de los clientes se buscó este fin en el desarrollo del presente proyecto.

Uno de los requerimientos que se considero para ajustar el sistema de soporte a las políticas de desarrollo internas es el de compatibilidad con los productos desarrollados por la empresa. Este requerimiento hizo que se consideraran los modelos GCI y CRM previamente elaborados.

La revisión del modelo GCI aportó en la definición del componente de Gestión Documental del Sistema de Gestión de Calidad en cuanto a los elementos que estructuraron el modelo de Datos del sistema y la definición de los formularios de la aplicación.

Posterior al diseño del modelo de datos se pasó al diseño detallado de las opciones que se completó tomando como base el avance realizado por el desarrollo del sistema GCI CRM, cuyos formularios se ajustaron a las necesidades de Visión y se elaboraron los expuestos en el presente documento.

Se realizó una evaluación de las herramientas disponibles en la empresa para el desarrollo del proyecto, lo que llevó a la selección de Oracle Developer 6i en vista de que en cuanto a recursos hardware y software resultaba en costos mucho menores a la utilización de Oracle Developer 9i.

En observancia del principio de “Mejora Continua” del Sistema de Gestión de Calidad, la empresa capacitó al personal que laboró en el proyecto en las

herramientas necesarias a fin de responder a las necesidades en cuanto al desarrollo de software.

Seguidamente se elaboraron los formularios y reportes que conformarían el sistema. Los formularios en un sistema de información son los elementos fundamentales de un sistema de información porque tienen interacción directa con los usuarios y recogen la mayor parte de la información. Para su elaboración se requiere realizar etapas detalladas de análisis y diseño. Durante estas etapas es fundamental la comunicación con los usuarios a fin de que la información que requieren y suministran al sistema sea más que un montón de datos almacenados.

“Oracle 6i Forms Developer” resulta ser una herramienta apropiada para la construcción de aplicaciones empresariales por las facilidades que ofrece y la variada gama de posibilidades que presenta entre las que se encuentran:

- Generación automática de formularios con base en el modelo de datos
- Variedad de posibilidades en el diseño de formularios por los múltiples elementos de que dispone
- Estructuras definidas de programación para el manejo de eventos.
- Validación automática de los datos.

Los reportes en un sistema de información también juegan un papel importante para el apoyo de la administración de la organización y al igual que el sistema del cual forma parte, requiere de realizar etapas de análisis y diseño para su elaboración. Durante estas etapas es fundamental la comunicación con los usuarios a fin de que la información que requieren se presente de manera adecuada a sus necesidades.

“Oracle 6i Reports Developer” es una herramienta apropiada para la construcción de reportes por sus características, entre las que se encuentran:

- Generación automática de reportes con base en unos parámetros de diseño
- Reportes tabulares, matriciales y gráficos que se generan dinámicamente a través de parámetros otorgados por el usuario del sistema.
- Múltiples opciones en cuanto al tipo de formato en el que se desea el reporte, bien sea en papel, como un archivo o soportado en web.

El uso de herramientas como estas en el desarrollo de Sistemas de Información hacen que el desarrollo se centre en tener un diseño adecuado y en cuanto a la programación de los componentes reflejar al mayor grado posible los procesos internos de la organización. Se logró, de igual forma, una alta productividad en el desarrollo de los formularios y reportes requeridos en la empresa.

Las actividades realizadas muestran que la gestión de las solicitudes de los clientes abarca un área de la información de la empresa muy amplia que se requiere considerar para cumplir con los niveles de calidad deseados tanto a nivel del desarrollo del presente proyecto como para los productos que se ofrecen a los clientes.

BIBLIOGRAFÍA

BOEHM, Barry. "Software Engineering Economics". Capítulos 2 y 3. Editorial Prentice Hall. Englewood Cliffs, New Jersey. 1981.

Sistema Nacional de Aprendizaje - SENA, Regional Santander. Curso Virtual de Gestión de Calidad ISO 9000. www.senasantander.org. Bucaramanga, Colombia. 2005.

CAZORLA, Javier. "Sistema de gestión de la calidad: ISO 9001". [www.mailxmail.com/curso/ empresa/iso9001](http://www.mailxmail.com/curso/empresa/iso9001). 2004.

WELLS, Don. "Extreme Programming: A gentle introduction". www.extremeprogramming.org. 1999

LETELIER, Patricio, PENADÉS, M^a Carmen. "Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)". Departamento de Sistemas Informáticos y Computación. Facultad de Informática. Universidad Politécnica de Valencia. www.willydev.net/descargas/masyxp.pdf. 2003.

GOMEZ FLOREZ, Luis Carlos. "Planeación de Proyectos – Un enfoque para la Ingeniería de Sistemas e Informática". Bucaramanga, Colombia. 2004.

SÁNCHEZ, Víctor G., RÍOS, Homero V. "Metodología CASE para el desarrollo de sistemas". México. 1996. www.lania.mx/biblioteca/newsletters/1996-otono-invierno/articulo1.html

CANTOR, Murray. "Orient Object Project Management with UML". Editorial Wiley. 1998

Object Management Group, Inc. – OMG. “Introduction to OMG's Unified Modeling Language™ (UML®)”. www.omg.org/gettingstarted/what_is_uml.htm. 2005.

COETZEE, Francois. “UML 2.0”. www.xpdian.com/UML2.0.html. 2005.

NOVOSELSKY, Angel, CHANG, Ben, MACKY, Ian, KARUN, K., SCARDINA, Mark, RAMAKRISHNAN, Nirajan, KIRITZOV, Stefan. “Manual de Oracle 8i XML”. Editorial Mc Graw Hill. 1º Edición. 2001.

BASU, Julie, SALOMON, Martin K., MORISSEAU-LEROY, Nirva. “Oracle 8i Programación de Componentes Java”. Editorial Mc Graw Hill. 1º Edición. 2001.

Documentación de Oracle9i Developer Suite – Getting Started 2002

SARKER, Saonee. Prentice-Hall, Inc., A division of Pearson Education, Upper Saddle River, New Jersey 07458. 1999. <http://www.prenhall.com/divisions/bp/app/hoffer/oracle/tutorial/index.html>

Robert H. Smith School of Business. Diciembre 15 del 2005. <http://www.rhsmith.umd.edu/dit/oracle/formstutorial.html>

Acelerador para la implantación de Oracle Internet Application Server http://www.oracle.com/global/es/consultoria/accel_iAS.pdf

DTD for Customizing Graphs in Oracle Reports. www.oracle.com/technology/products/reports/htdocs/getstart/whitepapers/graphdtd/graph_dtd_technote_2.html

ANEXOS

ANEXO A

www.oracle.com/technology/products/reports/htdocs/getstart/whitepapers/graphdtd/graph_dtd_technote_2.html

4 DTD for Customizing Graphs in Oracle Reports

An Oracle Reports Technical Note
September 2003

| [Introduction](#) | [DTD for the Graph](#) | [Related Materials](#) |

4.1 Introduction

Reports Builder provides the Graph Wizard to insert graphs in both Paper and Web Reports. Reports Builder uses what you specify on each page of the Graph Wizard to create an XML file (graph.xml). This XML file is used to create the look-and-feel of the resulting graph at runtime.

A common requirement of Reports users is that one should be able to customize the graph to suit the particular report's look-and-feel. Although it is possible to customize the graph by re-entering the Graph Wizard and changing the attribute values/layout etc, however, there are a lot of other attributes which could be manipulated to enhance the appearance of the graph but they are not currently exposed by the Graph Wizard. All of these attributes are available in a DTD file graph.dtd.

This Technical Note explains all these attributes using comments within the graph.dtd. The objective is to explain the usage of these attributes to enable more fine-grained customization of graphs than is possible using the Graph Wizard.

Note: Links have been provided to different Elements of the DTD at the top for easy navigability. Colors used: **ELEMENT**, **ATTLIST**

[Back to Top](#)

4.2 DTD for the Graph

[DTD for the Graph](#)

[Graph definition](#)

[Default size of the graph image](#)

[Size of graph image](#)

[Place for paging component](#)

[Text elements](#)

[Title elements](#)

image

- Graph definition of the title; appears in graph image
- Graph definition of the subtitle; appears in graph

bubbles)

- Graph definition of the footnote
- Axis title elements
 - Title for the ordinal axis (O1Axis)
 - Title for the X data axis (on scatters and
- Title for the Y1-axis (left side); reads up
- Title for the Y2-axis (right side); reads down
- Title for the series edge (left edge of the

floor) in a 3-D graph

- Tick label elements
- Marker label and related elements
 - ViewFormats for different number values that can appear
- in marker text and tooltip text
 - Text for pie slices
 - Text for series items in the legend
 - Labels for pies in pie graph; pies are groups
 - Total that appears in the center of a ring in a ring

graph

- Graph version of a Font
- Line elements: axes, tick marks, and so on
 - Axis elements
 - Ordinal axis on bar, line, area graphs
 - Horizontal data axis on bubble and scatter
 - graphs; center point in polar graph
 - Left data axis on a vertical graph
 - Right data axis on a vertical graph
 - Tick mark elements
 - Tick marks along O1Axis; graphs except pies,
- scatter, bubbles
 - Tick marks along X1Axis; bubbles, scatters, and

polar

- Tick marks along Y1Axis
- Tick marks along Y2Axis
- Panel grids: the gridlines in 3-D graphs
 - Lines from left side of floor up the right wall
 - Lines across the walls
 - Lines from right side of floor up the left wall
 - Y Grid carried to the 3D markers
- Zero line elements: a line for zero
 - Zero along Y1Axis
 - Zero along Y2Axis
- Base line elements; base lines are lines from which bars and
- area markers originate
 - Base line for Y1-axis

Base line for Y2-axis
Reference line for X1-axis: includes up to 3 actual
lines

Reference line for Y1-axis: includes up to 3 actual
lines

Reference line for Y2-axis: includes up to 3 actual
lines

Miscellaneous line elements

Line that shows cumulative percentage in a pareto graph
Big blue lines across bubble graphs

Region elements (plot area, legend, etc.)

Region where data is plotted in graphs other than pie
and 3-D

Region where data is plotted in pie and ring graphs

Three-dimensional graph plotting region elements

Floor of region where data is plotted in 3-D graphs
Left wall of region where data is plotted in 3-D graphs
Right wall of region where data is plotted in 3-D

graphs

Graph legend: does not appear on 3-D graphs except 3-D

surface

Size of legend in a 3-D surface graph
Special effect element
Color in a gradient
Fine tuning for 3-D panels

Marker elements

For data on graph types other than pie and stock
For data on pie and ring graphs
For stock data in stock graph
For volume in stock graphs that show volume
For series in the legend (data values in 3-D surface
graph)

Data marker that does not share series attributes
 Create one for each special marker

SeriesItems: series-related attributes
 For a single series

Data-related elements

Data for local (static) data source
Row labels
Column labels
Labels
Grid of data
Row of data in grid
Cell of data in row
Relational data for local data source
Row of relational data
Number of columns the grid should have, when known

Number of rows the grid should have, when known
User-defined element
UserElement attribute: name of element
User-defined property
Rule elements
Imports
ViewFormat definition
ViewStyle definition
DataviewTitles definition
Dataview definition of the title
Dataview definition of the subtitle
Dataview definition of the footnote
QDR definition



ANEXO B

041_06

Bucaramanga, 06 de Febrero de 2006

Ingeniero:
LUIS IGNACIO GONZÁLES RAMÍREZ
Director de Escuela
Escuela de Ingeniería de Sistemas e Informática
Universidad Industrial de Santander
Ciudad

Asunto: practica empresarial

Por medio de la presente me permito informar el cumplimiento de los objetivos de la practica empresarial con el tema: Codificación realización de pruebas y documentación de programas para el registro de información y generación de reportes a través de la metodología Extreme Programming con el fin de consolidar el sistema de gestión de calidad de Visión adelantada en Visión Ltda., Por el estudiante RICARDO ARMANDO PABON PINILLA identificado con la c.c 91.528.186 de Bucaramanga.

Demostrando una labor satisfactoria en todo lo que realizo en un término de 6 meses iniciando en 1 de Agosto del 2005 al 31 de Enero del 2006.

Agradezco su colaboración.

Cordialmente,



GERARDO ARMANDO GARAVITO FLOREZ
Gerente
VISION LTDA

