

CAPÍTULO 4

Implementación y personalización del sistema

4.1 Metodologías de diseño

Los diagramas o representaciones gráficas son una parte fundamental en el desarrollo de sistemas. En general, se utilizan para definir las especificaciones de un programa y muestran el diseño de los sistemas (programas). Proporcionan el proyecto original para la implantación del diseño en código y además son partes esenciales para la documentación del sistema y posteriormente para su mantenimiento.

Los diagramas claros son una parte fundamental en el diseño de sistemas y en el desarrollo de programas durante el ciclo de vida del software. Una técnica de diagramación pobre puede provocar la inhibición en el razonamiento, en cambio, una buena puede acelerar el trabajo y aumentar la calidad de los resultados. Por ejemplo, si sólo hay una persona que desarrolla el diseño de un sistema o de un programa, los diagramas ayudarán al razonamiento claro. Cuando hay varias personas que participan en el desarrollo del sistema los diagramas son una poderosa herramienta esencial para la comunicación y el desarrollo del sistema.

Con una técnica de diagramación formal, los desarrolladores tendrán la posibilidad de intercambiar ideas y podrán reunir con precisión los distintos componentes del sistema. Una de las características con que deben contar los diagramas es con el uso de una simbología estandarizada para facilitar la comunicación y evitar las confusiones que puedan conducir a un error dentro del desarrollo del sistema por parte del equipo de trabajo.

En términos generales, podemos decir que los diagramas forman un lenguaje que permite el modelado del software, debido a que proporcionan una forma de representación concisa y sin ambigüedades. Además, forman una parte fundamental en el ciclo de vida, análisis y el diseño del software. Ver figura 4.1.1.

Ciclo de vida típico de los sistemas informáticos

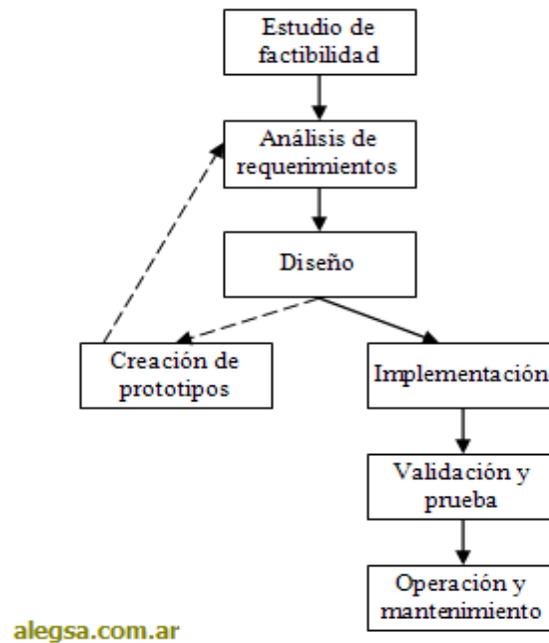


Figura 4.1.1, Ejemplo del ciclo de vida de un sistema informático.

Importancia de la visión en los diagramas

Las técnicas de diagramación se utilizan para dar una visión de alto nivel y detallada de un sistema. Para representar un sistema completo se necesita un diagrama de estructura de datos de alto nivel y otro detallado del sistema. La visión estructurada del sistema identifica los componentes del proceso y las relaciones entre ellos mostrando el flujo de datos, el flujo de control, así como el tiempo y la secuencia de la información. La visión de la estructura de datos describe principalmente las entidades y atributos, así como sus posibles relaciones o asociaciones. La importancia de ambas visiones radica en que los ingenieros pueden ver cómo los datos se derivan o utilizan en un determinado proceso. Existen varias técnicas de diagramación (estructurada), pero los cuatro tipos de diagramas fundamentales para representar un sistema informático son:

1.- Diagrama de procesos (DP): Un diagrama conocido y utilizado durante el análisis para definir los componentes del problema (sistema) y diseñar un primer

esbozo de los componentes del programa y del paso de datos entre ellos. Describe la transformación que los datos experimentan en su flujo a través del sistema. Los Diagramas de Procesos (DP) utilizan 4 símbolos, ver figura 4.1.2.



Figura 4.1.2 Simbología de Diagrama de Procesos

Otros símbolos empleados en los diagramas de flujos de datos son los siguientes, ver figura 4.1.3

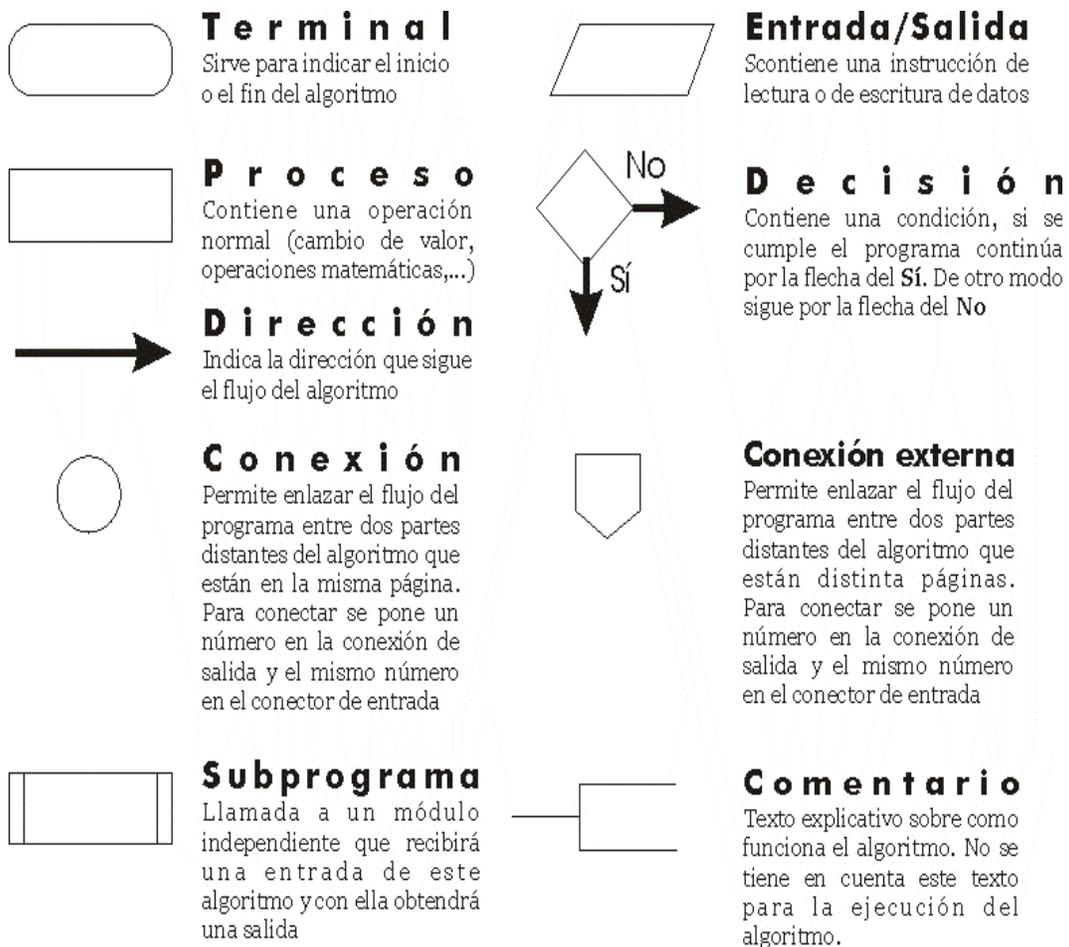


Figura 4.1.3. Algunos símbolos usados en los DFD.

Metodología Vs Ciclo de vida

Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. La metodología indica cómo hay que obtener los distintos productos parciales y finales.

La Metodología es el conjunto de pasos, métodos, procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevas aplicaciones informáticas. Una metodología, por tanto, representa el camino para desarrollar aplicaciones informáticas de una manera sistemática.

Normalmente consistirá en un conjunto de fases descompuestas en subfases (módulos, etapas, pasos, etc.). Esta descomposición del proceso de desarrollo guía a los ingenieros en la elección de las técnicas que debe elegir para cada estado del proyecto y facilita la planificación, gestión, control y evaluación de los proyectos.

Evolución de las metodologías

En la siguiente tabla podemos ver cómo han surgido las metodologías más representativas en la historia de la Ingeniería del Software.

Años	Metodologías
1968	Concepción sobre la programación estructurada de DIJKDTRA
1974	Técnicas de programación estructurada de WARNIER y JACKSON
1975	Primeros conceptos sobre diseño estructurado de MYERS y YOURDON
1977	Primeros conceptos sobre el análisis estructurado GANE y SARSON
1978	Análisis estructurado: DEMARCO y WIINBERG. Nace MERISE
1981	SSADM. Information Engineering
1985	Análisis y Diseño estructurado para sistemas de tiempo real de WARD y MELLOR
1986	SSADM Versión 3

1987	Análisis y Diseño estructurado para sistemas de tiempo real de HATLEY Y PIRHBAY
1989	METRICA
1990	SSADM Versión 4
1993	METRICA Versión 2
1995	METRICA Versión 2.1

Tipos de metodologías

Entre los tres tipos de metodologías más utilizados para la creación de bases de datos, están la de: Merise, SSADM y la de Yourdon/DeMarco.

Estructura general de MERISE

Las bases de MERISE comenzaron en 1972 por un equipo universitario de ingenieros de Aix-en-Provence. La primera versión salió a finales de 1976. El proyecto partió del Centre Technique Informatique del Ministerio de Industria Francés en Septiembre de 1977, para cubrir las necesidades tanto de la administración como de las empresas. El proyecto finalizó en mayo de 1978 dando lugar a MERISE como metodología de Análisis y Diseño de Sistemas de Información.

Esta metodología aporta un ciclo de vida más largo a los existentes hasta entonces que se materializa en un conjunto definido de etapas. Introducen dos ciclos complementarios: ciclo de abstracción y ciclo de decisión. El ciclo de abstracción se basa en la percepción de tres niveles de abstracción: conceptual, organizativo y físico. Además se definen dos niveles para cada nivel: un modelo de datos y otro de tratamientos.

Las fases de la metodología MERISE son:

1. Estudio preliminar.
2. Estudio detallado.

3. Implementación.
4. Realización y puesta en marcha.

Estructura general de SSADM

El gobierno británico plantea la necesidad de crear una metodología y se desarrolló entre el Central Computing and Telecommunications Agency (CCTA) y Learmonth and Burchett Management Systems (LBMS), dando como resultado la metodología SSADM (Structures Systems Analysis and Design Method). Los aspectos claves de SSADM son:

- Énfasis en los usuarios: sus requisitos y participación.
- Definición del proceso de producción: qué hacer, cuándo y cómo.
- Tres puntos de vista: datos, eventos, procesos.
- Máxima flexibilidad en herramientas y técnicas de implementación.

SSADM, ver figura 4.1.4, proporciona un conjunto de procedimientos para llevar a cabo el análisis y diseño, pero no cubre aspectos como la planificación estratégica ni entra en la construcción del código.

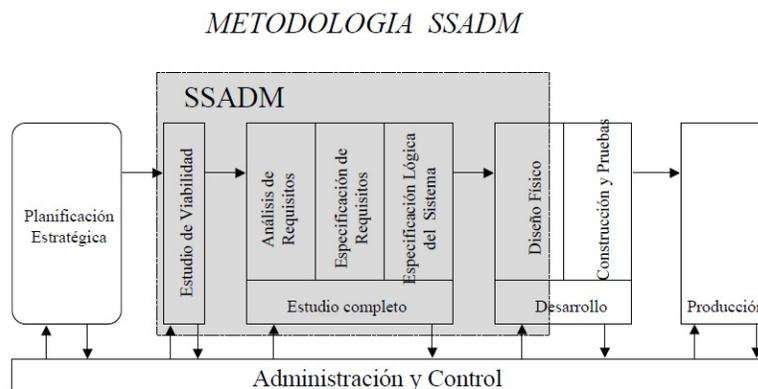


Figura 4.1.4. Metodología SSADM

En definitiva las fases son distintas pero a la vez muy similares ya que el proceso de creación de aplicaciones siempre tiene pasos por los que todas las metodologías han de pasar. En general las metodologías pretenden: Obtener mejores aplicaciones, mejorar el proceso de desarrollo, tener un proceso estándar en la organización.

Estructura general de Yourdon/DeMarco

El análisis estructurado de Tom DeMarco 1979 & diseño estructurado de Yourdon en 1979, propusieron un análisis y un diseño de sistema estructurados. Varios poderes del gobierno de los Estados Unidos de América lo estandarizaron en base al modelo DoD 2167A el cual requiere que los sistemas se compongan de una jerarquía de elementos de configuración de hardware y software. La comunidad DBA, con frecuencia, aplica niveles de detalle para modelar bases de datos relacionales.

La Técnica de Análisis y Diseño Estructurado propuesta por Edward Yourdon y Tom de Marco, se basa en una aproximación top-down al problema del diseño de los sistemas de información: comienza con el diagrama más general a lo particular. Ésta metodología está basada en los siguientes conceptos:

- Usa la organización jerarquizada descendente, por medio de la descomposición funcional para definir los requerimientos del sistema.
- Herramientas gráficas de comunicación y documentación. Ver la figura 4.1.5 es un ejemplo de Metodología Yourdon / De Marco.

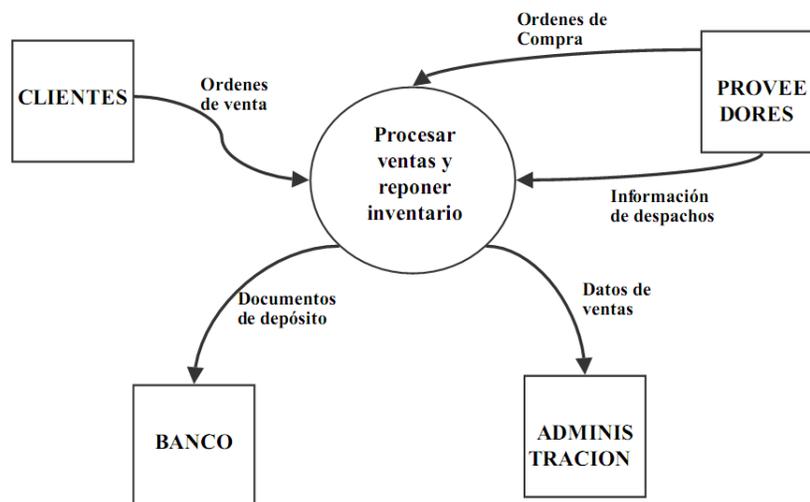


Figura 4.1.5. Metodología de diseño estructurado de Yourdon/DeMarco

Proponen la creación de modelos del sistema que representan los procesos, los flujos y la estructura de los datos de una manera descendente. Hay dos tipos: Orientada a Procesos: Se basan en la utilización de un método descendente de descomposición funcional para definir los requisitos del sistema, dan lugar a un nuevo concepto que es la especificación estructurada a un modelo gráfico particionado, descendente y jerárquico de los procesos del sistema. Esta metodología se compone de:

- Diagrama de procesos: Son diagramas que representan los procesos de datos que deben llevar a cabo un sistema a distintos niveles de abstracción y los datos que hay entre las funciones.
- Diagramas de flujo: muestran como se llevan a cabo cada una de las actividades de un proceso.
- Diagrama entidad relación: muestra como se asocian los procesos y relaciona la información en las diferentes tablas.
- Diccionario de datos: Es el conjunto de las definiciones de todos los datos que aparecen en el DFD.
- Especificaciones de procesos: como se obtienen las salidas del proceso a partir de sus entradas.

Esta metodología proporciona una manera para diseñar paso a paso sistemas y programas detallados. Cabe mencionar que unos pasos involucran el análisis, otros el desarrollo del diseño y otros más la medición y la mejora de la calidad del diseño. La principal herramienta generada en el diseño estructurado es el “diagrama de estructura” donde muestra los componentes de procedimientos del programa, su ordenación jerárquica y los datos conectados a ellos. El diagrama de estructura es un diagrama de árbol o jerárquico que, en términos generales, define la arquitectura global de un programa que muestra los procedimientos y sus interrelaciones. En dicho diagrama se utilizan bloques básicos, como son cajas que representan los componentes de procedimientos y las flechas que muestran como se conectan. Estos procesos y los datos que los enlazan forman los cimientos para definir los componentes del programa. Ver figura 4.1.6.

La política de refinamiento Yourdon/DeMarco recomiendan que un DFD no debería contener más de siete procesos, y que los procesos complejos deberían ser refinados hasta que cada proceso primitivo del DFD de menor nivel pueda ser documentado en una página o menos de lenguaje natural estructurado.

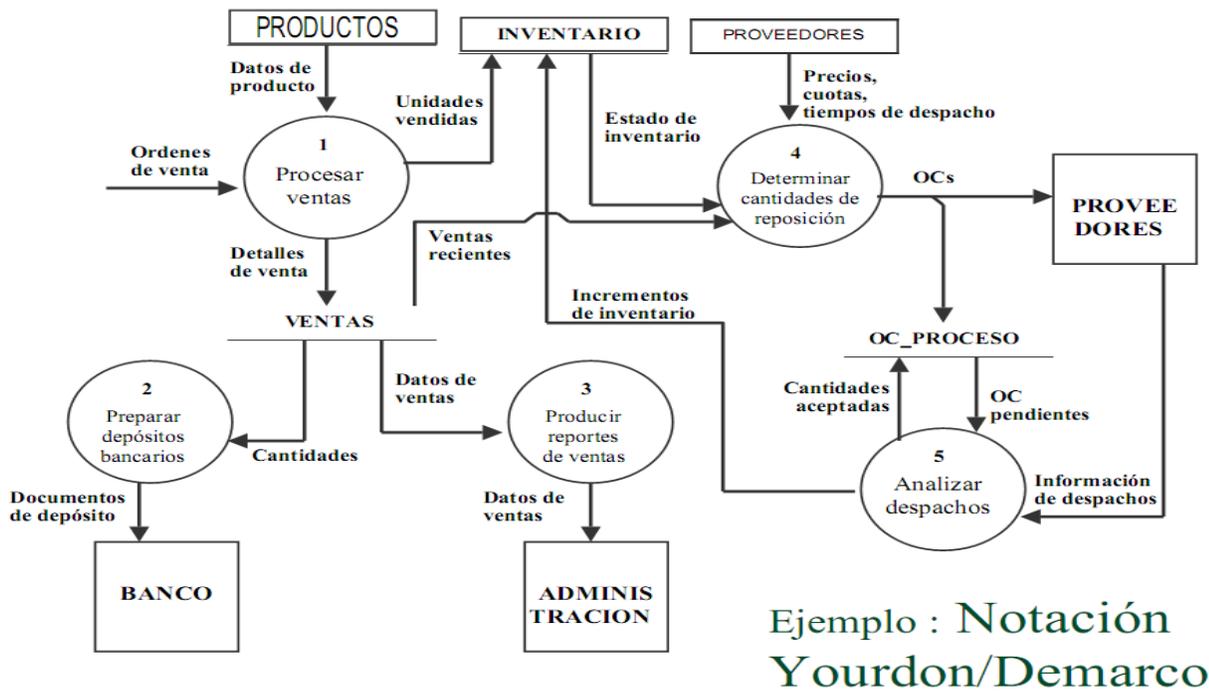


Figura 4.1.6 Ejemplo del Método Yourdon & DeMarco.

La metodología seleccionada es la de Yourdon/De Marco por los siguientes motivos:

- Se diseñará previamente un diagrama del contexto, en donde se definen las características generales del proceso en investigación,
- Luego se desarrollará el primer nivel en un diagrama físico de flujo de datos, en donde se definirá el sistema tal como es entendido y
- Finalmente se desarrollará cada uno de los procesos de primer nivel con todos los detalles que éste presenta. Con un menor tiempo de desarrollo.

Es decir, comenzaremos con el diagrama más general e iremos descomponiendo el sistema hasta lo más particular. Paso a paso con el análisis, el desarrollo del diseño y la medición y mejora del sistema.

Características de la Principales Metodologías:

MERISSE	SSADM	YOURDON/DEMARCO
<ul style="list-style-type: none"> • Ciclo de vida más largo que los existentes • Introducción de 2 ciclos complementarios <ul style="list-style-type: none"> • De decisión • De abstracción: Se basa en la percepción de 3 niveles de abstracción en el que se definen dos modelos: <ul style="list-style-type: none"> • Modelo de datos y • Modelo de tratamientos • <i>Nivel Conceptual:</i> Se ocupa de definir el qué a través de un conjunto de reglas de gestión que pesan sobre la empresa. • <i>Nivel Organizativo:</i> Corresponde a la organización a implementar en la empresa para alcanzar los objetivos asignados al sistema. • <i>Nivel Físico u Operativo:</i> Corresponde a la integración de los medios técnicos para el proyecto. 	<p>Introduce una técnica llamada técnica del diseño del diálogo que se utiliza para diseñar la interfaz del usuario, los aspectos fundamentales son:</p> <ul style="list-style-type: none"> • Enfatiza mucho el nivel de usuario, • Analiza y diseña centrándose en 3 aspectos: datos, procesos y eventos, • Aporta máxima flexibilidad. 	<ul style="list-style-type: none"> • Construir el modelo físico actual (DFD físico actual). • Construir el modelo lógico actual (DFD lógico actual). • Crear un conjunto de modelos físicos alternativos. • Estimar los costes y tiempos de cada opción. • Seleccionar un modelo. • Empaquetar la especificación.

4.2 Diagramación

Diagrama de contexto

Un diagrama de contexto indica las interacciones del sistema con entidades externas a este, como lo muestra la figura 4.2.1

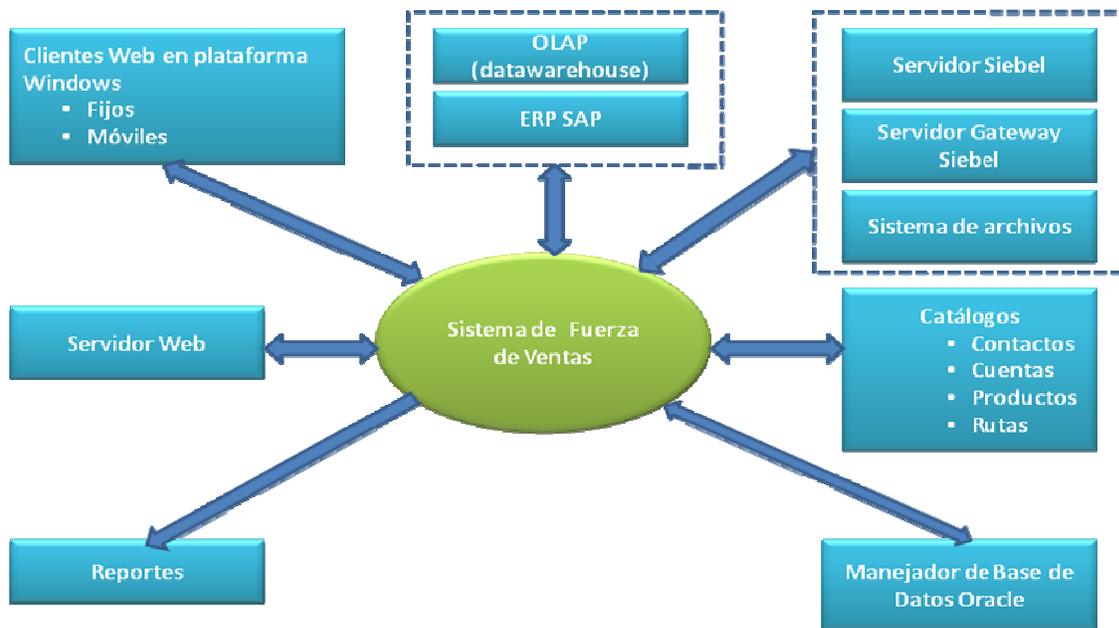


Figura 4.2.1 Diagrama de Contexto

Diagrama de flujo

Un diagrama de flujo es la funcionalidad que se espera sea satisfecha por el sistema y que ha sido solicitada por un rol de este. Analizando los requerimientos solicitados, se tiene lo siguiente:

Roles identificados:

- Visitador médico profesional (Usuario)
- Gerente de Distrito y/o Regional
- Administradores

Tareas que los usuarios del sistema deben realizar con la ayuda de este:

- Alta Médico - Cuenta
- Baja Médico - Cuenta
- Cambio de perfil básico Médico - Cuenta
- Pre-Visita Médico
- Pre-Visita Cuenta
- Visita Médico
- Visita Cuenta
- Proceso Administrativo

Todo esto está modelado en los diagramas de flujo de las figuras 4.2.2 a la 4.2.10 En el diagrama de flujo de la figura 4.2.2 muestra la ruta a seguir de cada uno de los procesos involucrados en el Sistema de Fuerza de Ventas, como en el caso de los representantes de ventas, el papel que tiene los gerentes y la parte que controla los flujos de datos que es el proceso administrativo.

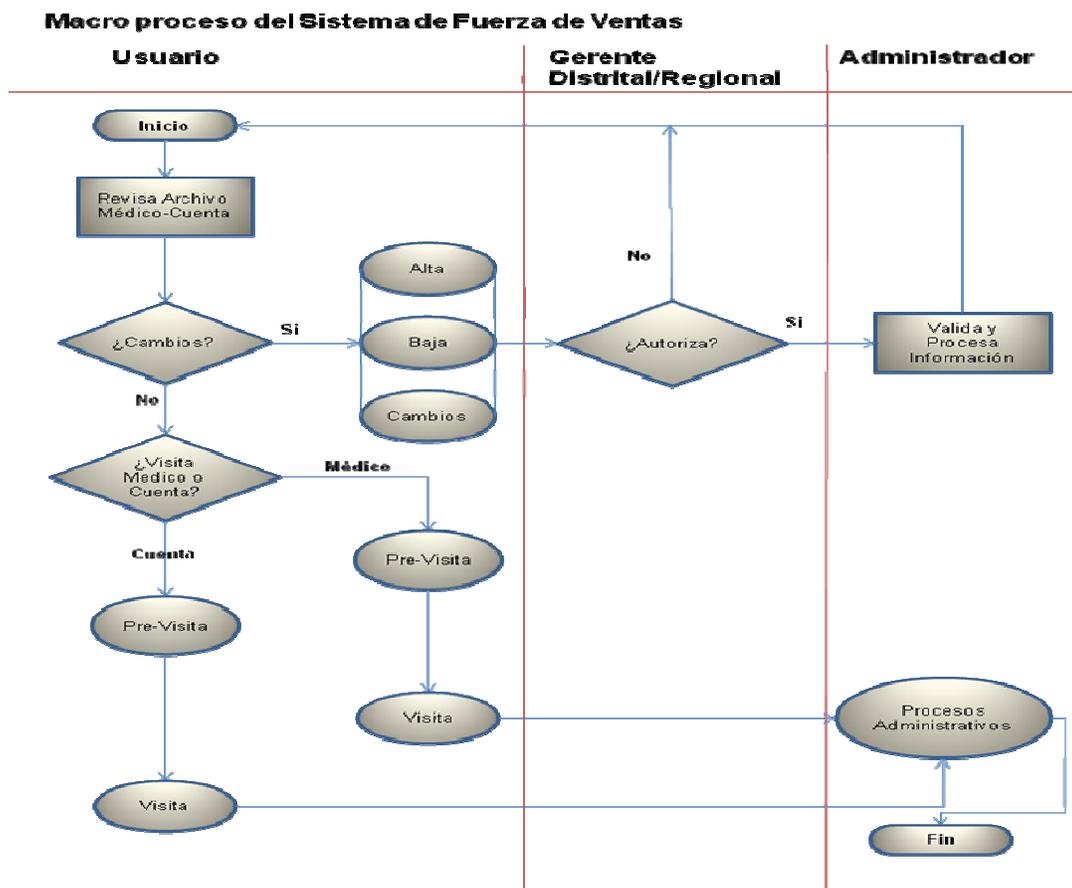


Figura 4.2.2 Diagrama de Flujo Macro del Sistema de Fuerza de Ventas

La figura 4.2.3, 4.2.4 y 4.2.5 muestra una de las partes principales del Sistema de Fuerza de Ventas que es la Médico-Cuenta, la cual representa a las instituciones que los Representantes visitan con acciones comerciales; Siendo en esta parte donde se capturan los datos básicos del Médico y la Cuenta asociados.

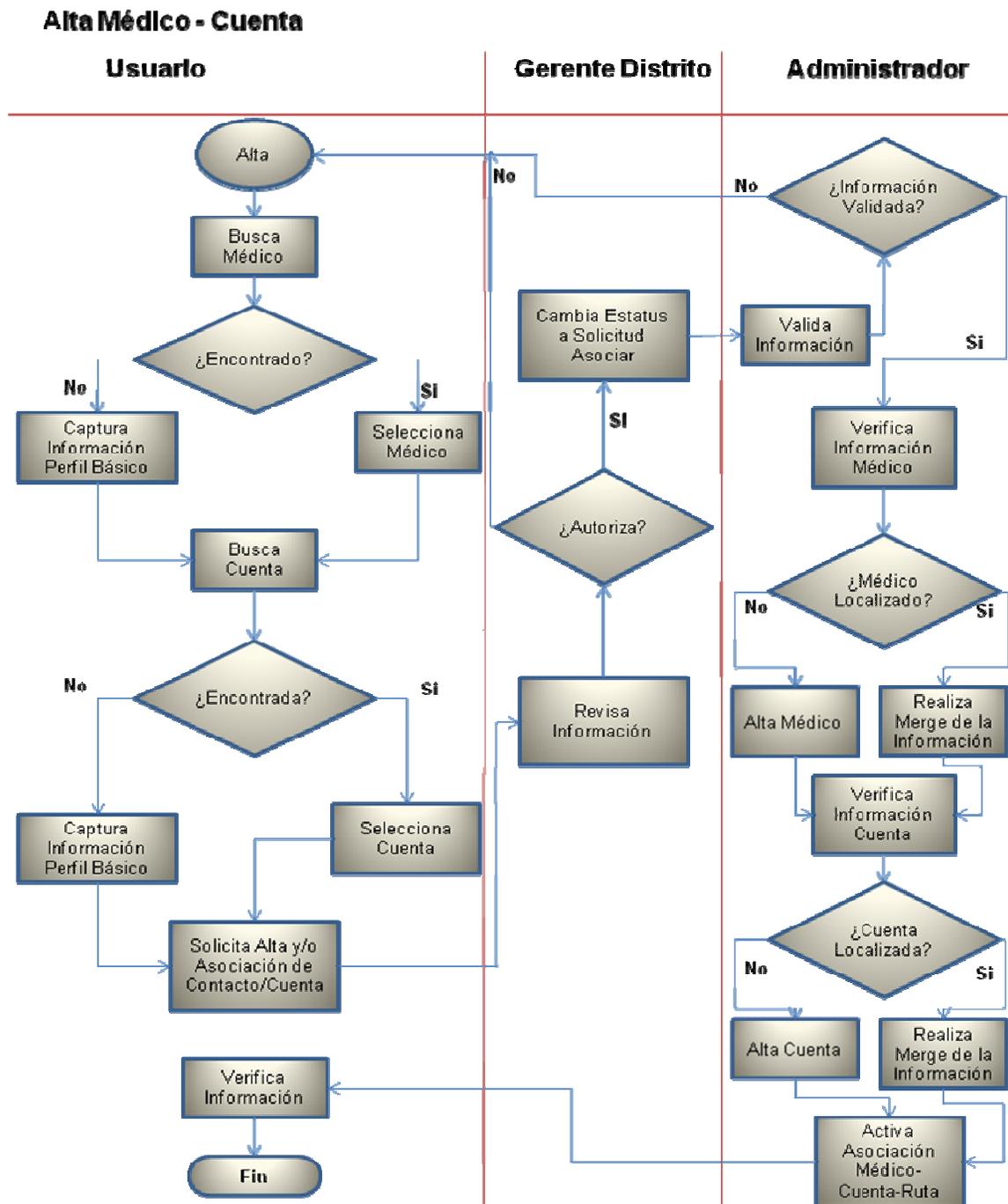


Figura 4.2.3 Diagrama de Flujo Alta Médico-Cuenta

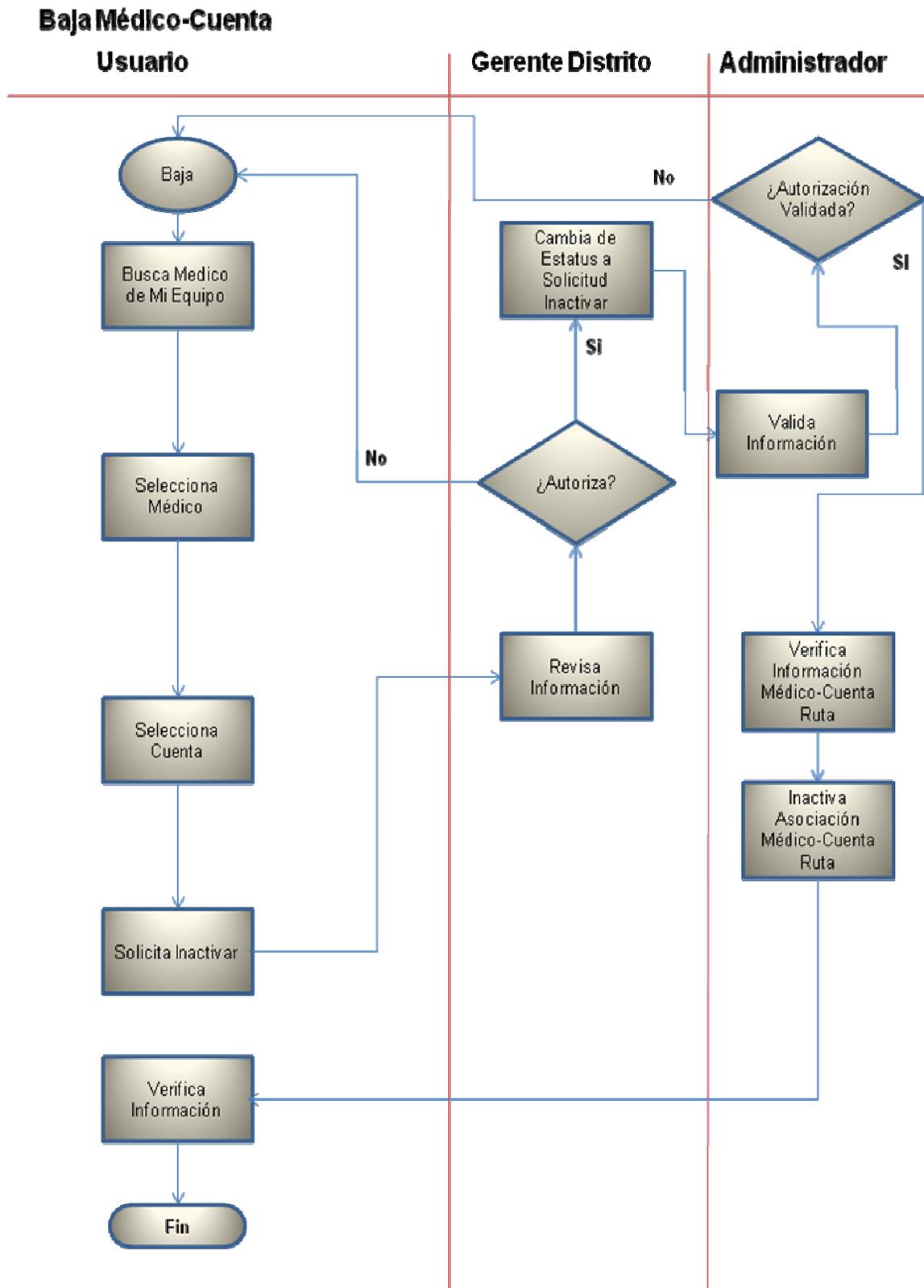


Figura 4.2.4 Diagrama de Flujo Bajas

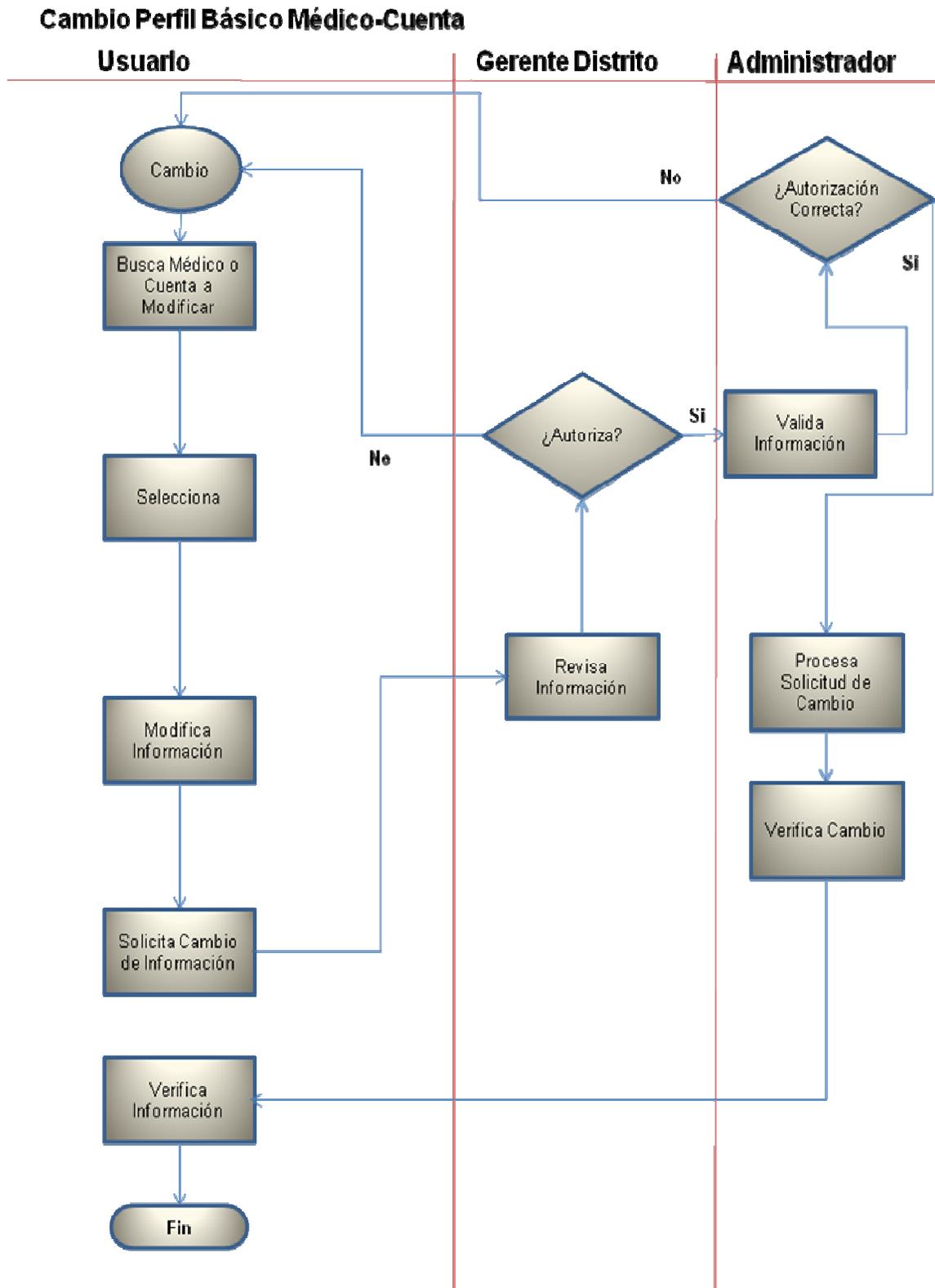


Figura 4.2.5 Diagrama de flujo Cambios

En el diagrama que se muestra en la figura 4.2.6 se selecciona el médico y la cuenta al cual se le realizará la visita, revisándose el historial de visitas, la parrilla promocional, las muestras que se tienen para este médico, así como la captura de algunas notas importantes para la visita.

Pre-Visita Médico

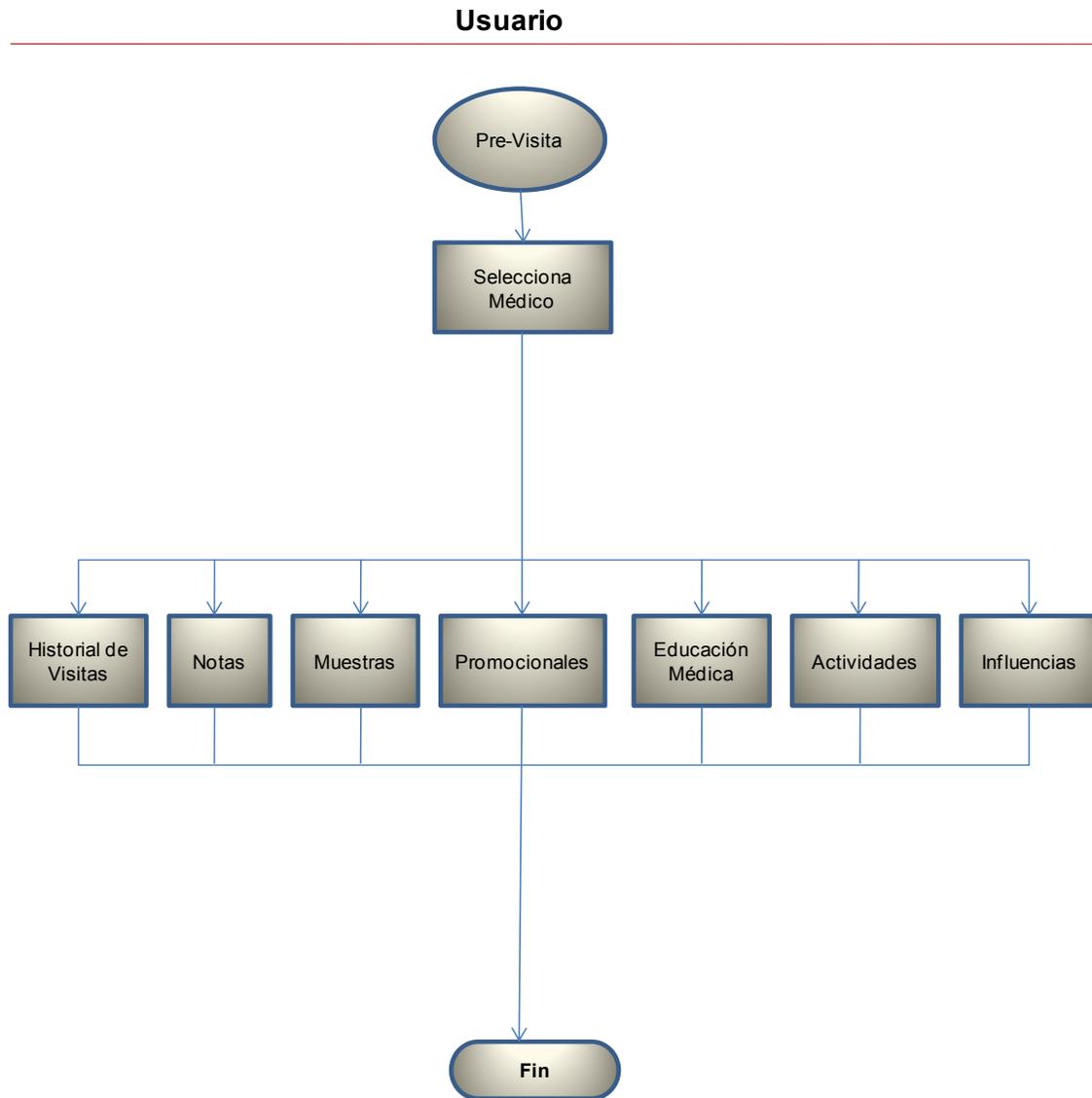


Figura 4.2.6 Diagrama de Flujo Pre-Visita Médico

En el diagrama de la figura 4.2.7 se muestra la **Pre-Visita**, en la que podemos revisar los datos que el Representante ingresa en sus visitas. Se muestran las siguientes opciones para revisar:

- Historial de visitas. Se muestra la lista de visitas que el Representante ha realizado al Profesional.
- Notas de visita. Notas relevantes de la visita que son **Públicas**.
- Educación médica. Se muestra si el Profesional está asignado a algún evento de educación médica.
- Actividades. Se muestran las actividades alternas a una visita médica.
- Parrila promocional. En esta sección se muestran los promocionales para la relación Profesional-Ruta (Fuerza de ventas, Sub-Especialidad). Sólo está disponible para el Representante la visualización de la información.

Pre-Visita Cuenta

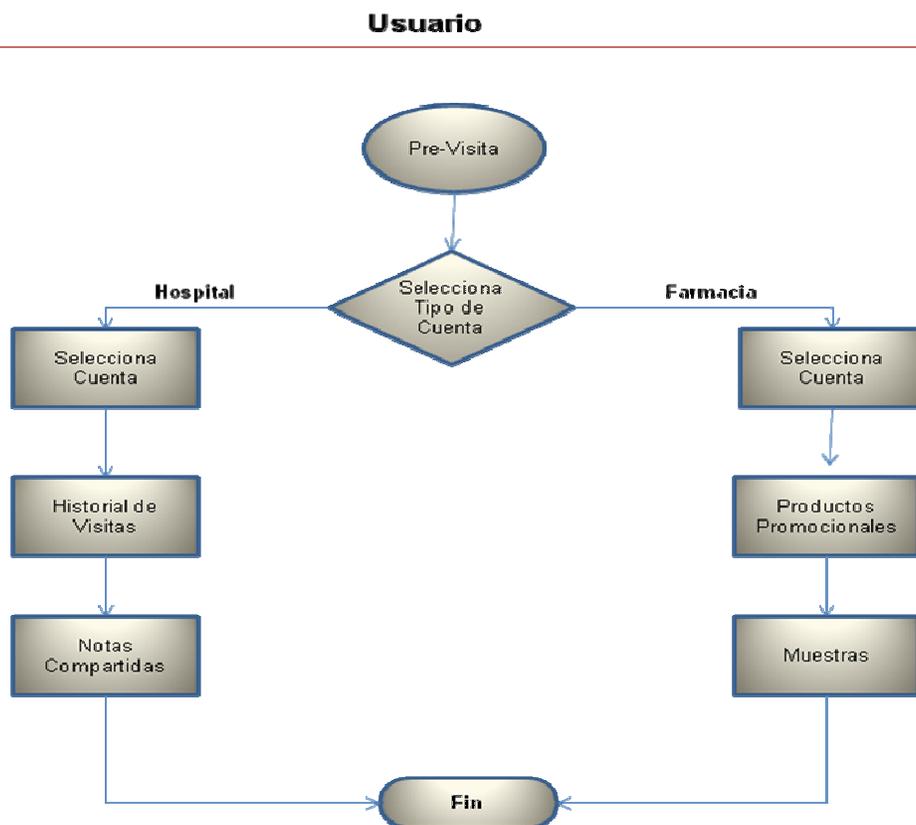


Figura 4.2.7 Diagrama de Flujo Pre-Visita Cuenta

En el diagrama de la figura 4.2.8 se muestra el historial de las visitas a la Cuenta, mostrando el status y la ruta a la que pertenece la visita, además se capturan las notas pertinentes y el objetivo de la siguiente visita.

Visita Médico

Usuario



Figura 4.2.8 Diagrama de Flujo Visita Médico

En el diagrama de la figura 4.2.9 se muestra la preparación de una visita, determinándose el tipo de visita a realizar, ya sea a hospital o farmacia, se captura el objetivo de la visita, los productos promocionales, etc.

Visita Cuenta

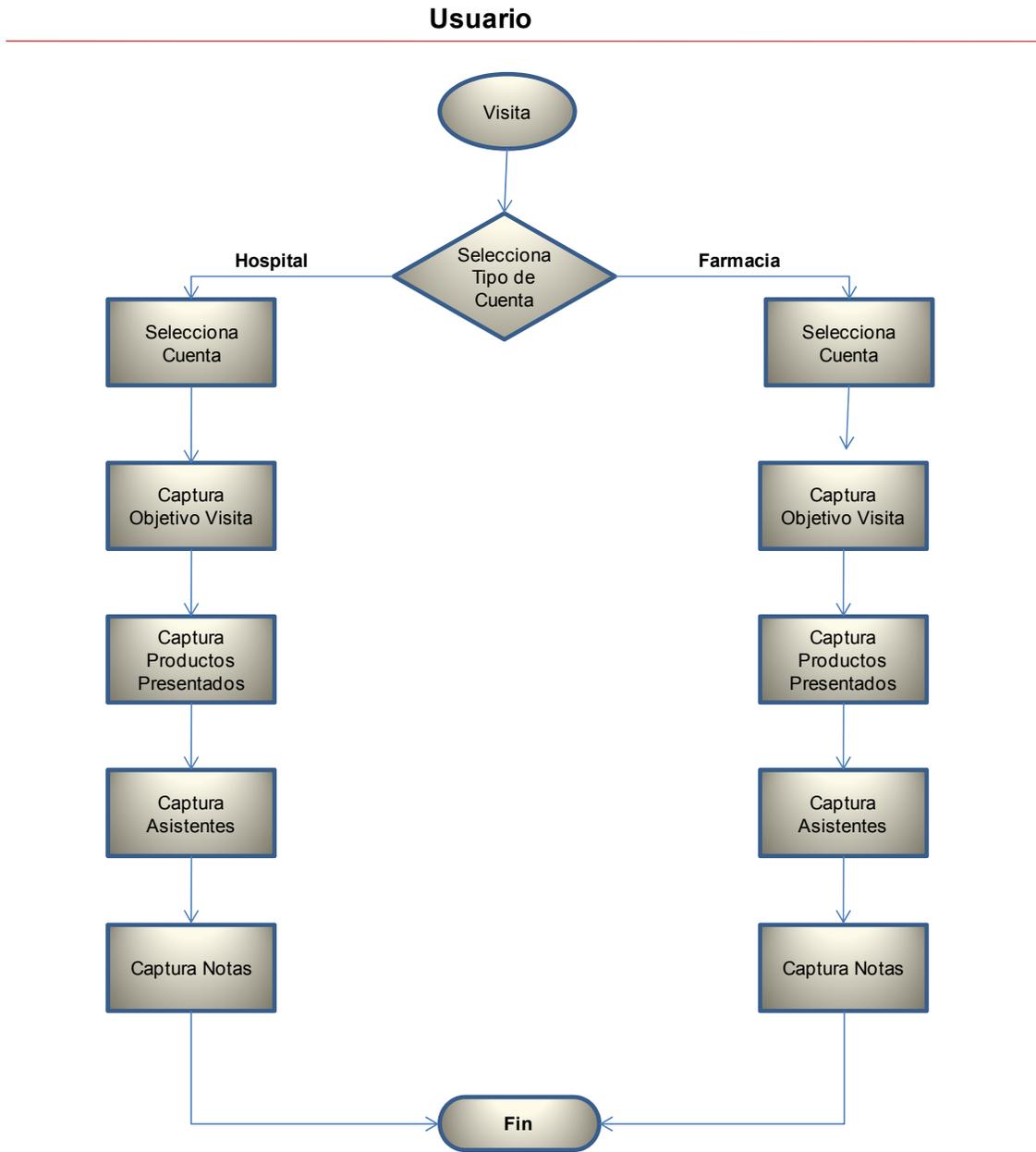


Figura 4.2.9 Diagrama de Flujo Visita Cuenta

El diagrama de flujo de la figura 4.2.10 muestra todos los procesos administrativos de la aplicación, como es la administración de grupos la cual está enfocada a definir la jerarquía organizacional de la compañía, que determina el acceso de los empleados a vistas y registros; La administración de usuarios proporciona al Administrador mayor control sobre Altas y Bajas de los empleados que utilizan la aplicación. Además de identificar fácilmente el rol que desempeñan en la compañía; La administración de datos involucra la creación de catálogos disponibles en la aplicación para ser utilizados en otros procesos por los usuarios.

Proceso Administrativo

Administrador

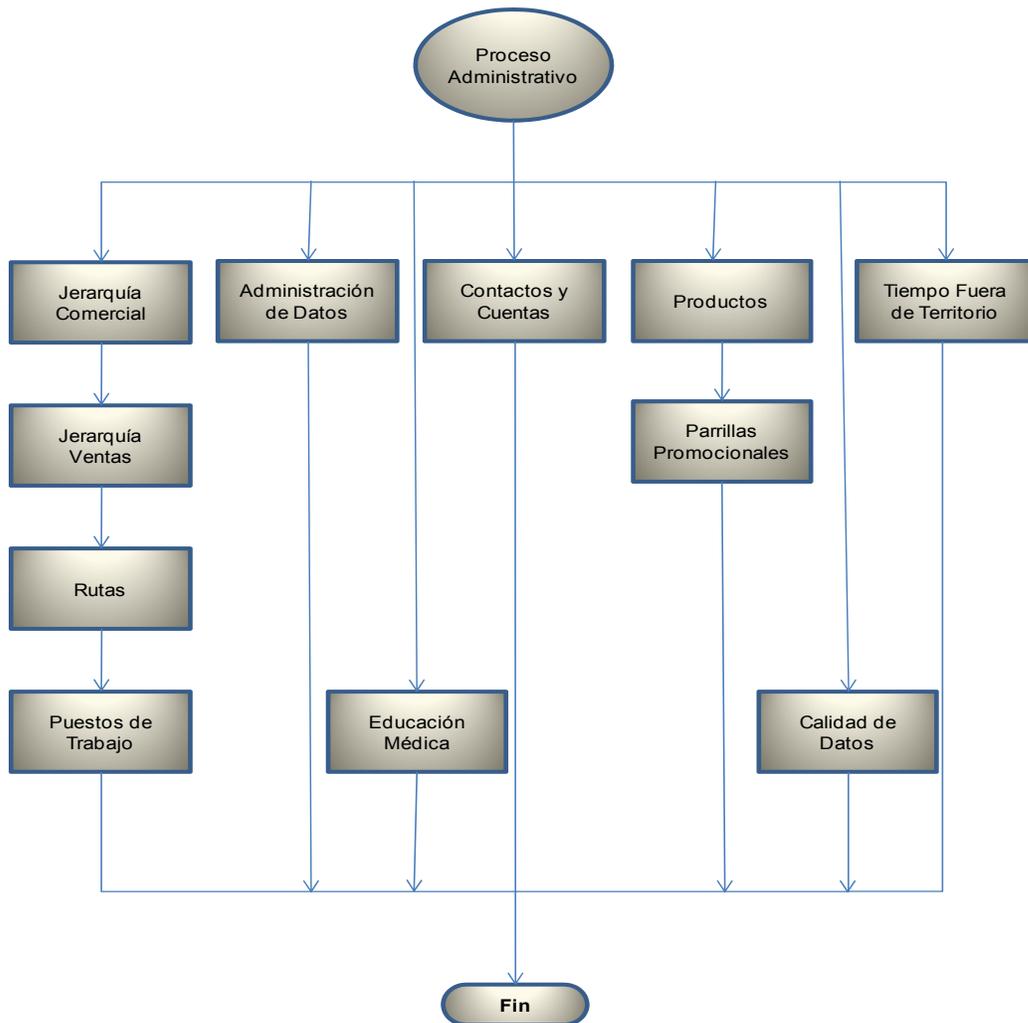


Figura 4.2.10 Diagrama de Flujo Administrativo

4.2.1 Diagrama de Proceso

Los Diagramas de procesos representan la funcionalidad que se espera sea satisfecha por el sistema, indicando que tabla está involucrada, como lo muestra la figura 4.2.1.1. Analizando los requerimientos solicitados, se tiene los siguientes procesos:

- Proceso de Altas, bajas y cambio de Cuenta –Contacto.
- Proceso de Visita.
- Proceso de asignación de Cuenta a Contacto.
- Proceso de Asignación de Ruta a empleado

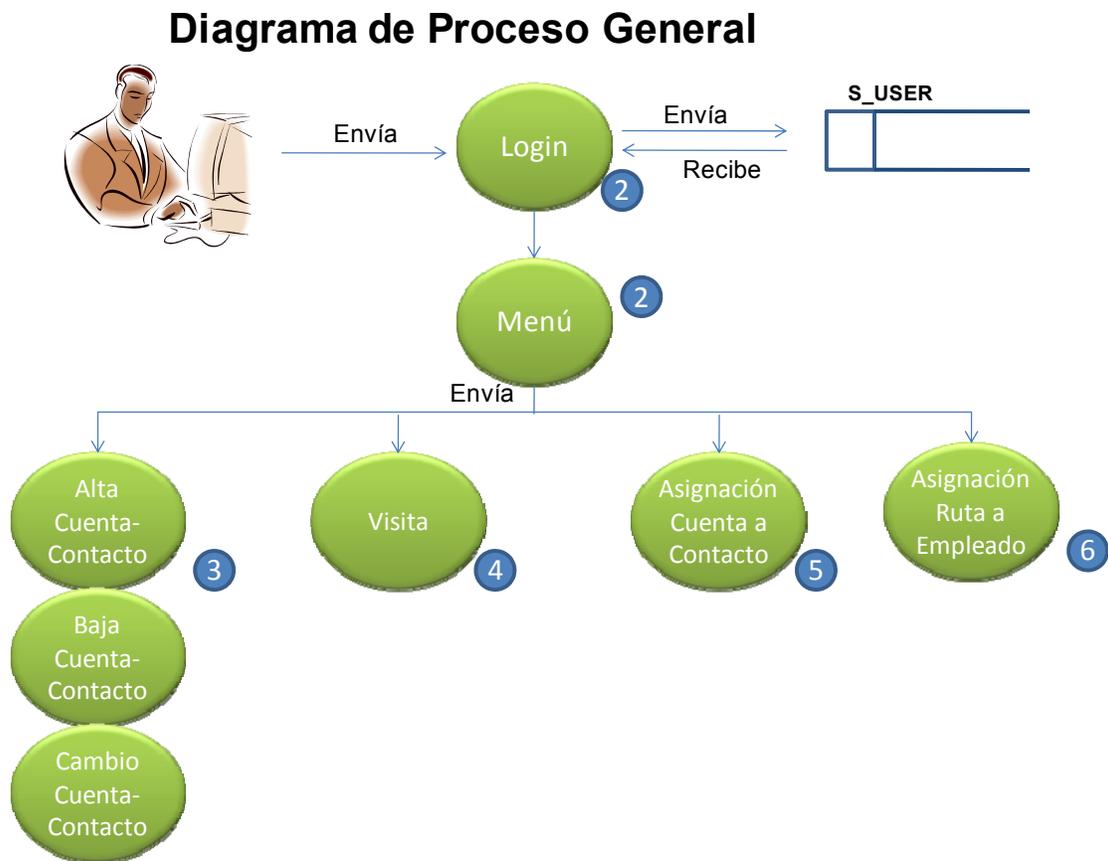


Figura 4.2.1.1 Diagrama de Proceso General

La figura 4.2.1.2 representa el proceso de inicio de sesión en el Sistema de Fuerza de ventas.

Proceso Login 2

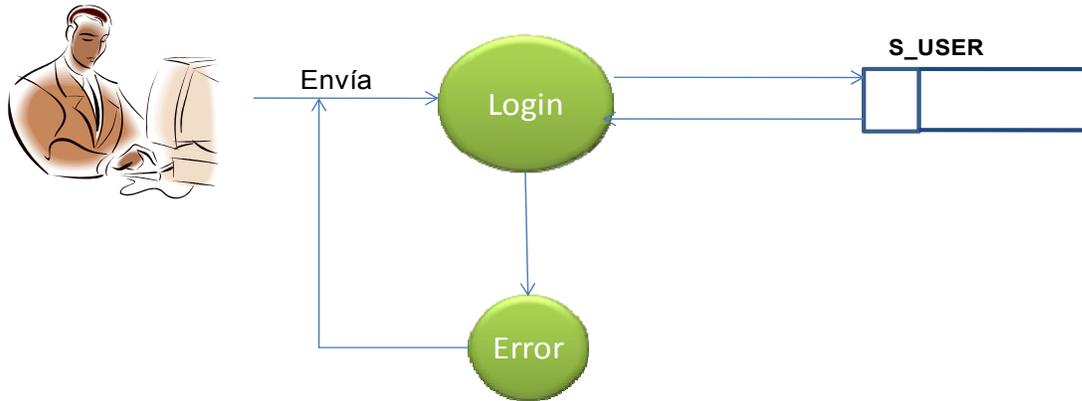


Figura 4.2.1.2 Diagrama de Proceso de inicio del sistema

Las figuras 4.2.1.3 a la 4.2.1.5 representan los procesos a seguir y las tablas que se consultan o se modifican para realizar altas bajas y cambios en una Cuenta y qué relación se tiene con el Contacto, pudiéndose asignar una Ruta.

Proceso Alta Cuenta - Contacto 3

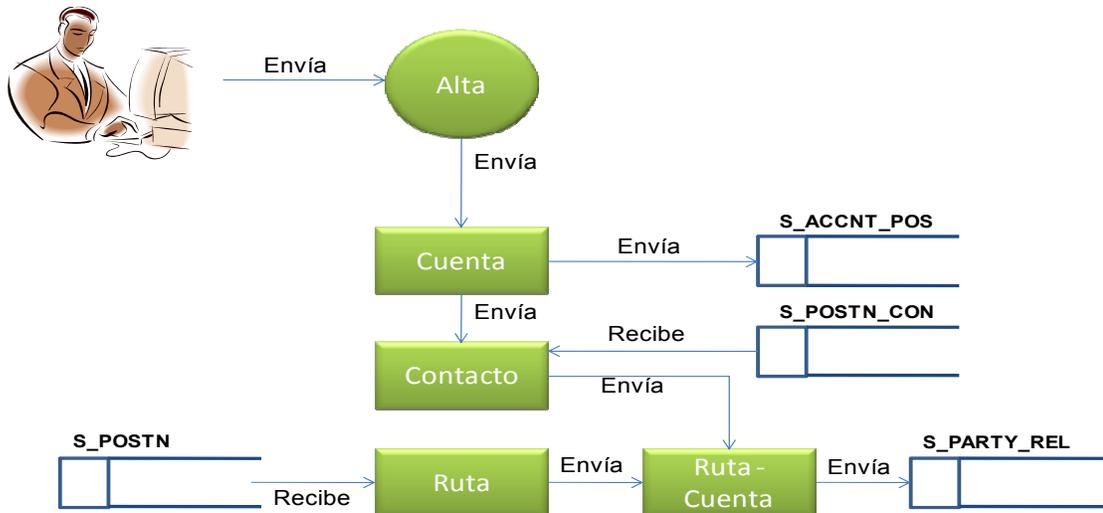


Figura 4.2.1.3 Diagrama de Proceso de Alta Cuenta-Contacto

Proceso Baja Cuenta - Contacto 3

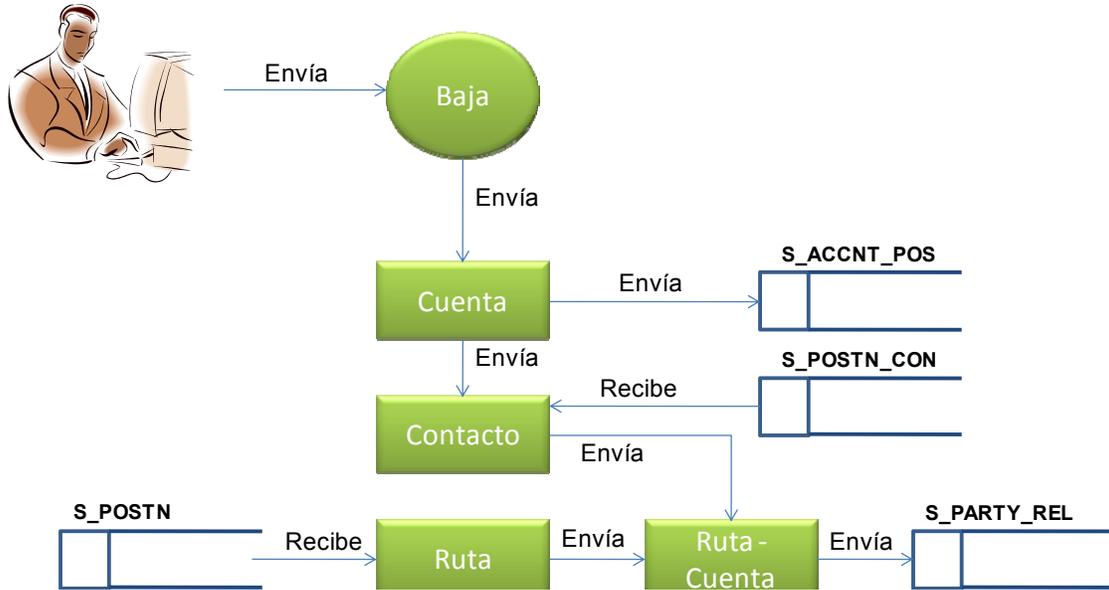


Figura 4.2.1.4 Diagrama de Proceso de Baja Cuenta-Contacto

Proceso Cambio Cuenta - Contacto 3

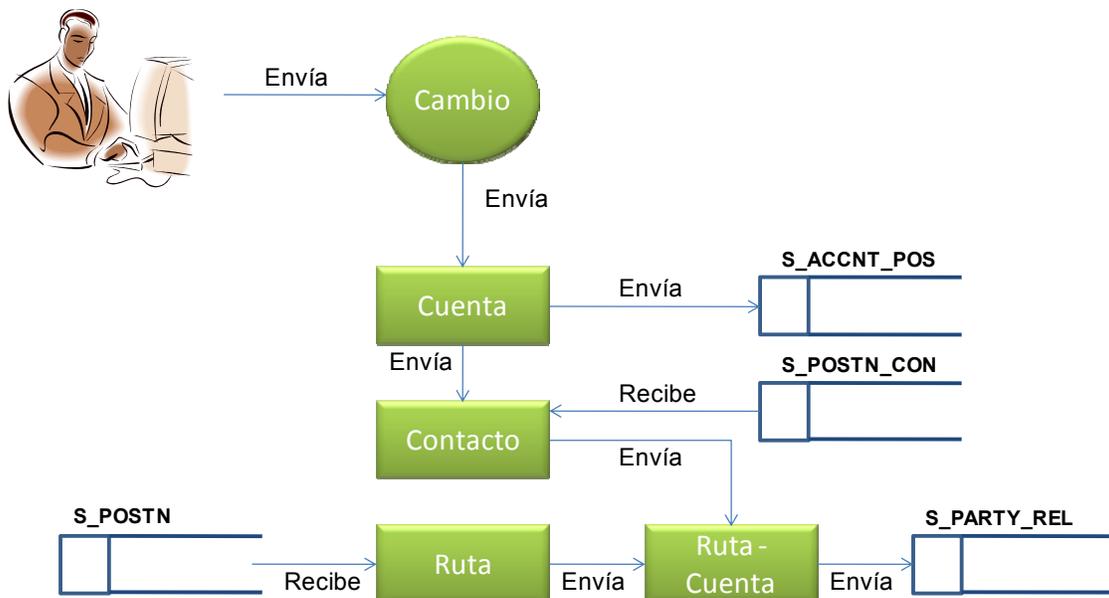


Figura 4.2.1.5 Diagrama de Proceso de Cambio Cuenta-Contacto

La figura 4.2.1.6 representa el proceso a seguir y las tablas que se consultan o se modifican para preparar las visitas que se van a realizar, tomando en cuenta las relaciones de Cuenta y Contacto, además de los productos que serán presentados.

Proceso Visita 4

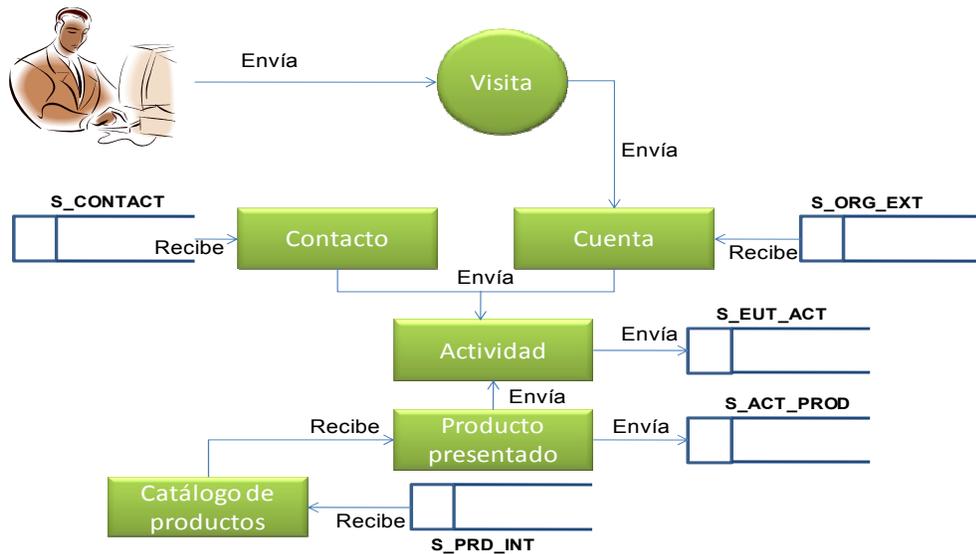


Figura 4.2.1.6 Diagrama de Proceso de Visita

La figura 4.2.1.7 representa el proceso a seguir y las tablas que se consultan o se modifican para realizar la asignación de una Cuenta a un Contacto.

Proceso Asignación Cuenta a Contacto 5

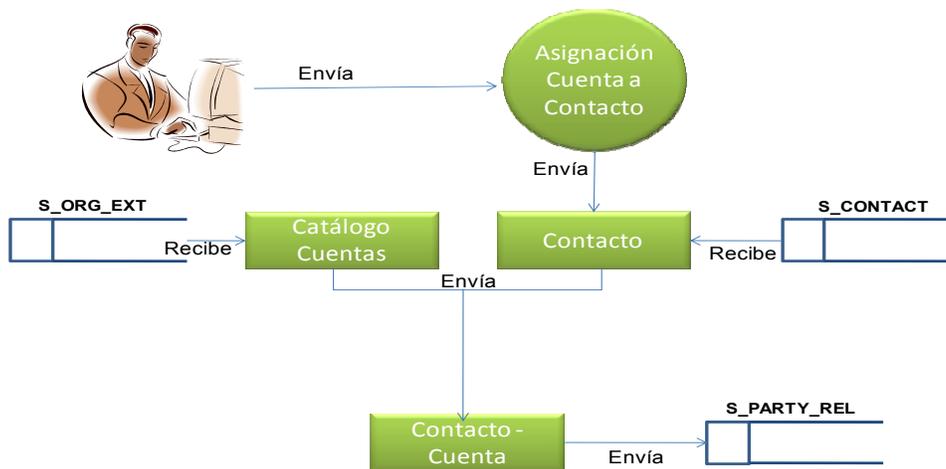


Figura 4.2.1.7 Diagrama de Proceso de Asignación Cuenta a Contacto

La figuras 4.2.18 representa el proceso a seguir y las tablas que se consultan o se modifican para realizar la asignación las Ruta a seguir por un Empleado.

Proceso Asignación Ruta a Empleado 6

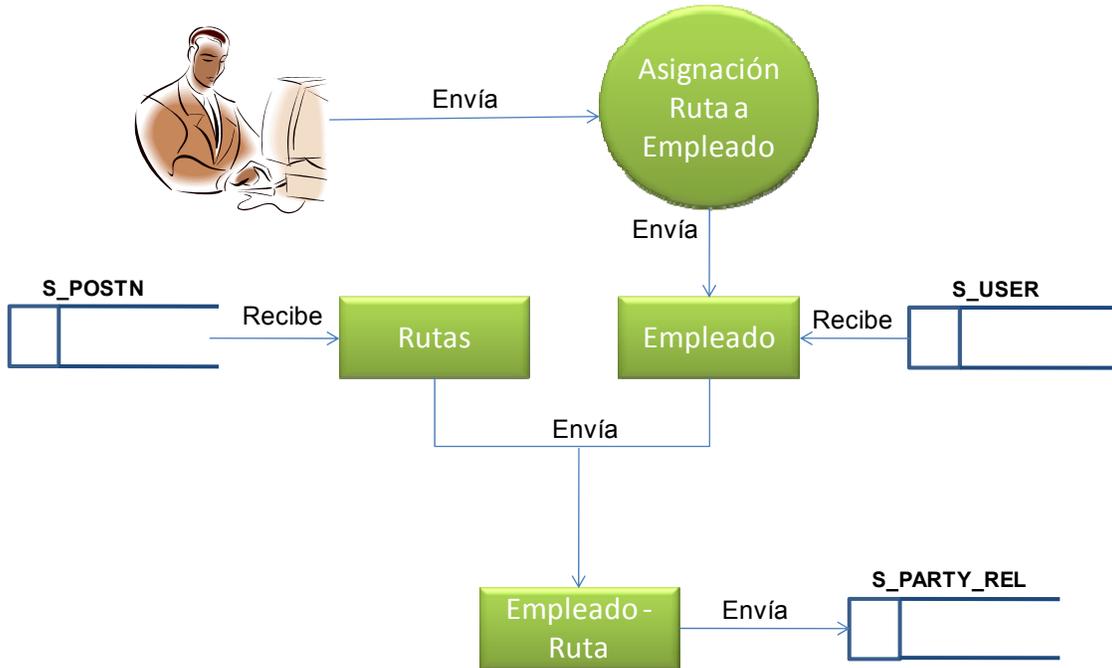


Figura 4.2.1.8 Diagrama de Proceso de Asignación Ruta a Empleado

4.2.2 Diagrama de Entidad – Relación

Los diagramas entidad relación (DER) están basados en el modelo de nombre ídem. Se basa en la percepción del mundo real, que consiste en un conjunto de objetos llamados entidades y las relaciones entre ellas.

Representa la estructura lógica general de la base de datos gráficamente.

Conceptos

Entidad: es un objeto concreto como una persona o abstracto como un día festivo o evento, siempre son sustantivos tangibles o intangibles.

Atributos: características propias que definen a la entidad

Dominio de Atributo: rango de valores permitidos para definir un atributo.

Conjunto de entidades: grupo de entidades del mismo tipo.

Relación: asociación entre entidades.

Llave candidato: Conjunto de uno o más atributos que junto identifican a una entidad como única.

Llave primaria: Es la llave candidato elegida para identificar una entidad como única (es la llave candidato más pequeña posible).

Llave Foránea: Es llave primaria en la Entidad "A" y la que sirve para relacionar con la Entidad "B"

Entidad fuerte o dominante: Entidad que tiene suficientes atributos para formar una llave primaria

Entidad débil o subordinada: Entidad que no tiene suficientes atributos para formar una llave primaria.

CUENTA

En esta entidad se almacenarán las cuentas y se observa la relación con direcciones y cuentas principalmente.

ENTIDAD	TABLAS
CUENTA	S_ORG_EXT, S_PARTY
PARTY	S_PARTY
UNIDAD DE NEGOCIO	S_BU, S_ORG_EXT_S_PARTY
PERSONA	S_CONTACT,S_PARTY
PERSONA CUENTA	S_PARTY_REL
EMPLEADO	S_EMP_PER, S_PARTY
PUESTO DE TRABAJO	S_POSTN, S_PARTY
ACTIVIDAD	S_EVT_ACT
MIEMBROS DE EQUIPO DE VENTAS	S_ACCNT_POSTN
SEGMENTO DE MERCADO	S_MKT_SEG
CUENTA SECTOR MERCADO	S_ACCNT_MKT_SEG
DIRECCION	S_ADDR_PER
NOMBRE JERARQUÍA	S_DYN_HRCHY
NOMBRE JERARQUÍA CUENTA	S_DYN_HRCHY_REL
SINONIMO DE CUENTA	S_ORG_SYN

Tabla 4.2.2.1 Tabla de entidad Cuenta

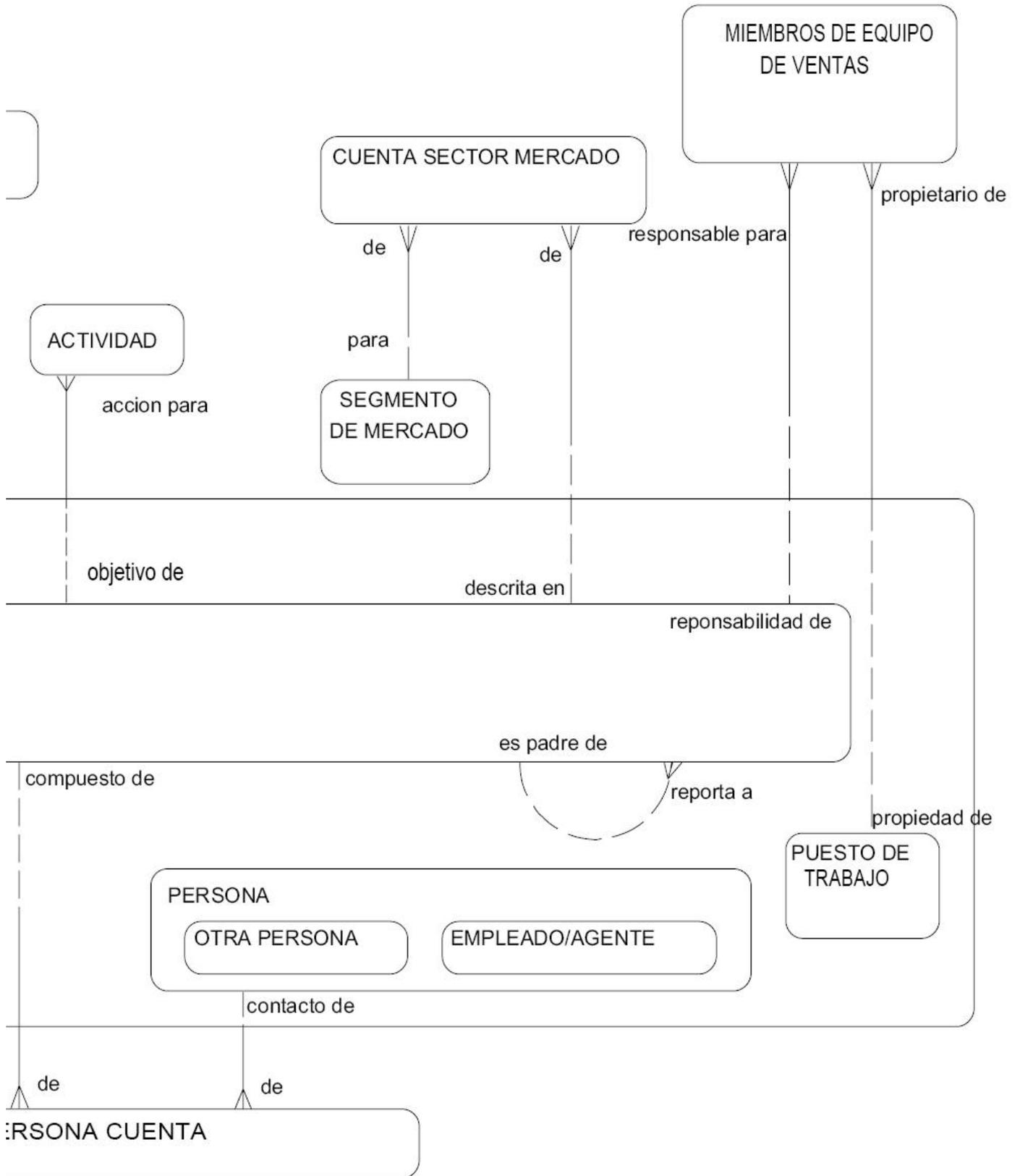


Figura 4.2.2.2 Diagrama Entidad Relación Cuenta (Continuación)

ACTIVIDAD

En esta entidad se almacenarán las actividades. Se le asocian a las Cuentas y a los Contactos, y se relaciona con productos.

ENTIDAD	TABLA
CUENTA	S_ORG_EXT, S_PARTY
CONTACTO CUENTA	S_PARTY_REL
ACTIVIDAD	S_EVT_ACT
CONTACTO	S_CONTACT
ACTIVIDAD PERSONA	S_ACT_CONTACT
EMPLEADO	S_EMP_PER, S_CONTACT, S_PARTY
EMPLEADO ACTIVIDAD	S_ACT_EMP
INDICIACION	S_PROD_APPLCTN
OBJECION	S_PROD_ISS
OBJECION ACTIVIDAD	S_ACT_ISS
PERSONA	S_CONTACT, S_PARTY
PRODUCTO	S_PROD_INT
PRODUCTO ACTIVIDAD	S_ACT_PRDINT

Tabla 4.2.2.2 Tablas de entidad Cuenta

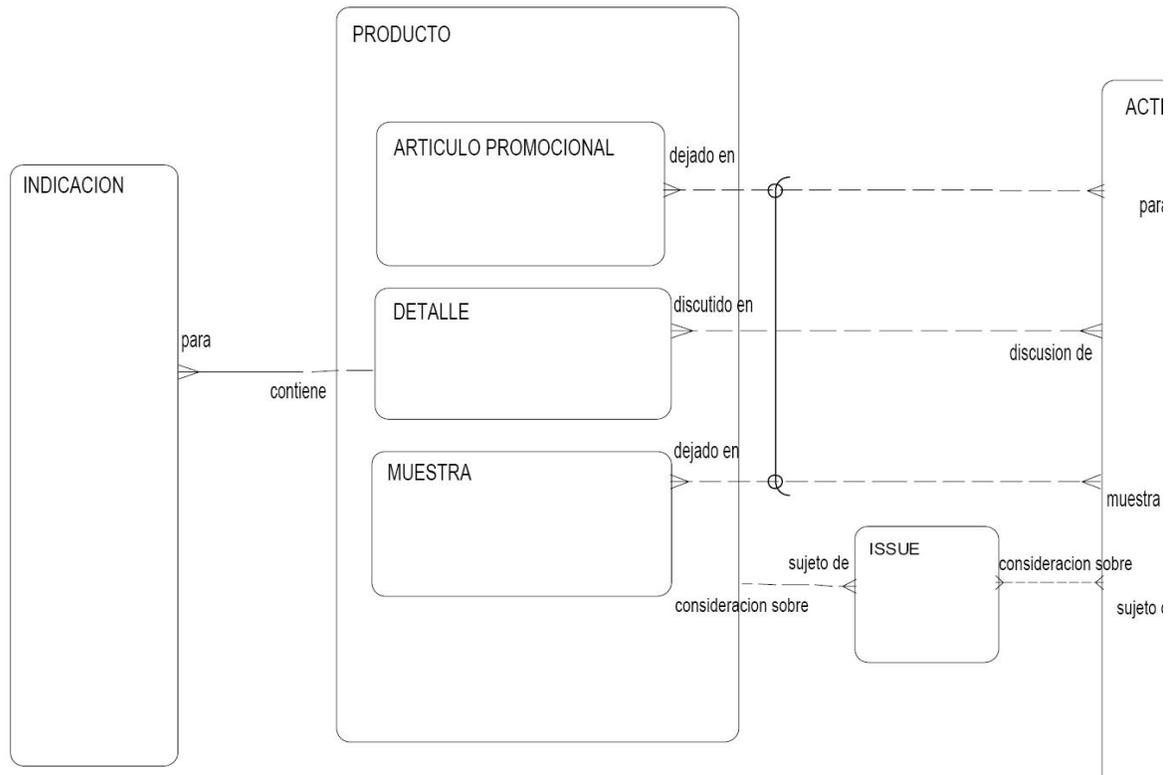


Figura 4.2.2.3. Diagrama Entidad Relación Actividad

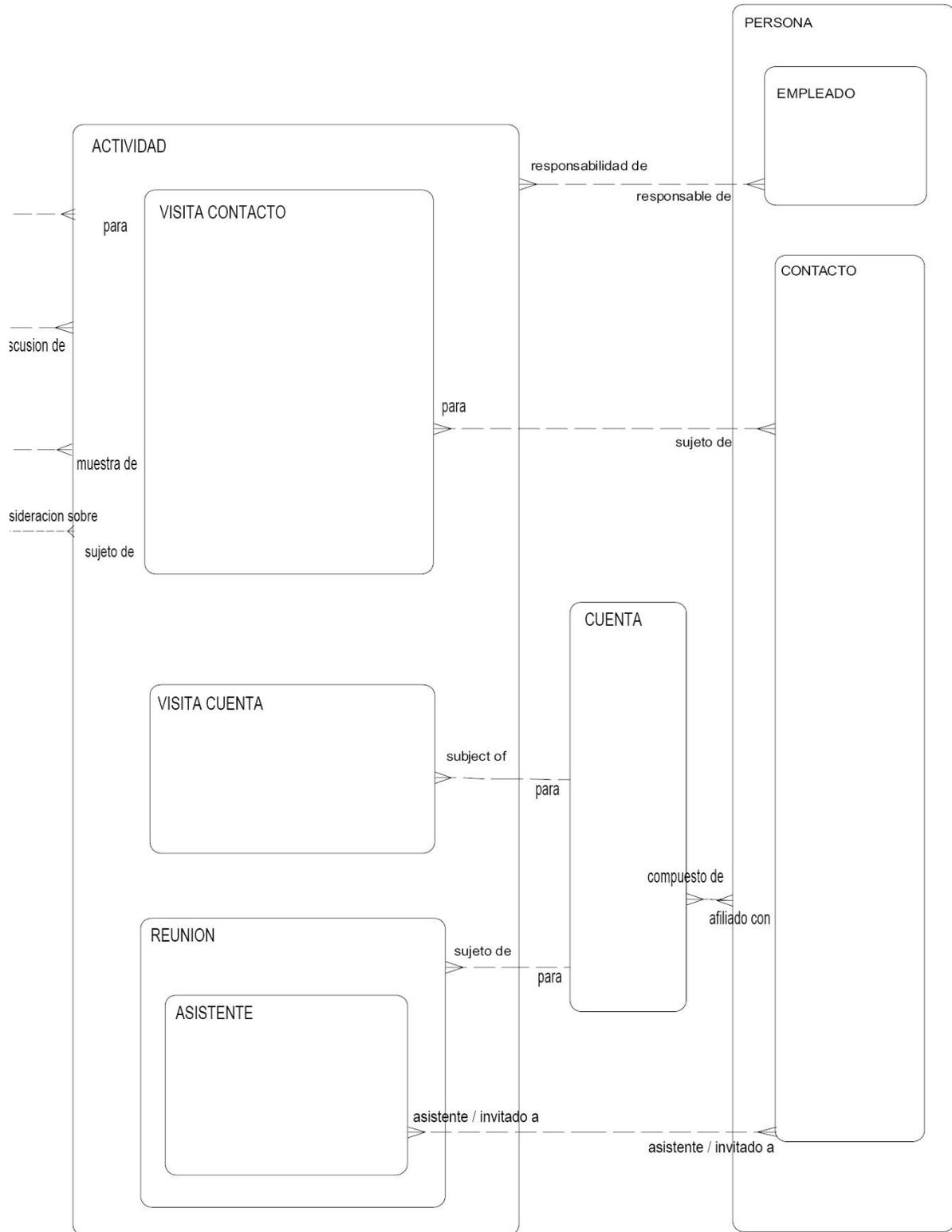


Figura 4.2.2.4. Diagrama Entidad Relación Actividad (Continuación)

CONTACTO

En esta entidad se almacenarán todos los contactos y personas en general como usuarios. Se relaciona con un puesto de trabajo o Ruta.

ENTIDAD	TABLA
RELACION PARTY	S_PARTY_REL
PARTY	S_PARTY
PERSONA	S_EMP_PER, S_CONTACT, S_PARTY
USER LOGIN	S_USER
PUESTO DE TRABAJO	S_POSTN
RELACION PUESTO CONTACTO	S_POSTN_CON
UNIDAD DE NEGOCIO	S_BU, S_PARTY
CUENTA	S_ORG_EXT, S_PARTY
DIRECCION CUENTA	S_ADDR_PER

Tabla 4.2.2.3 Tablas Entidad Contacto.

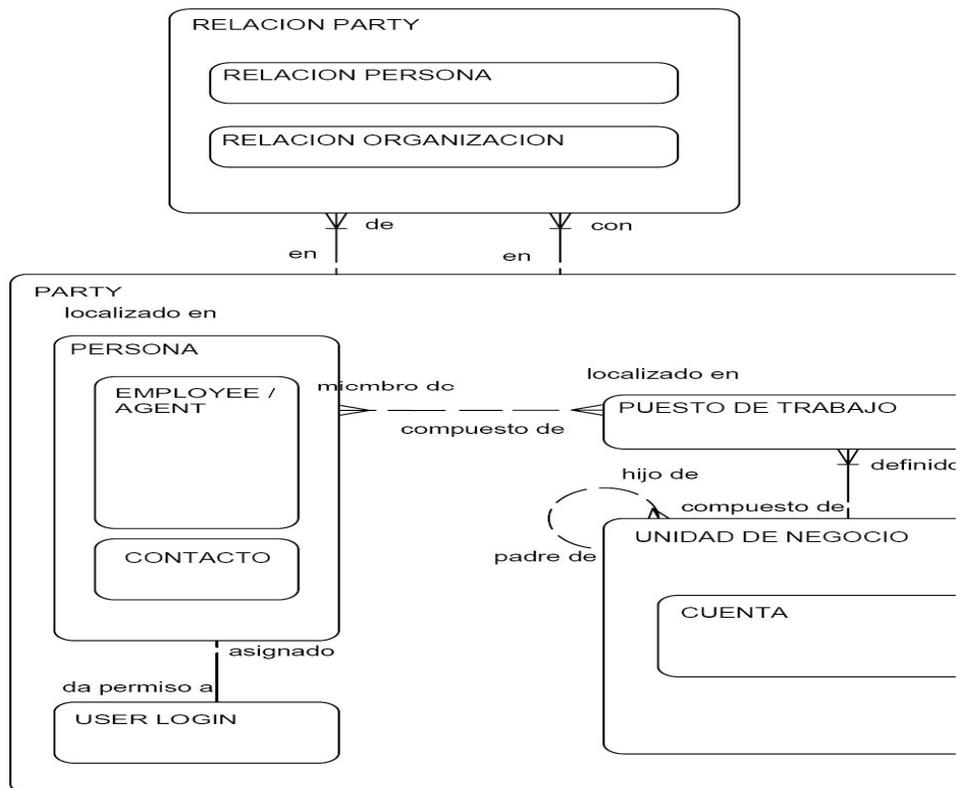


Figura 4.2.2.5 Diagrama Entidad Relación Contactos

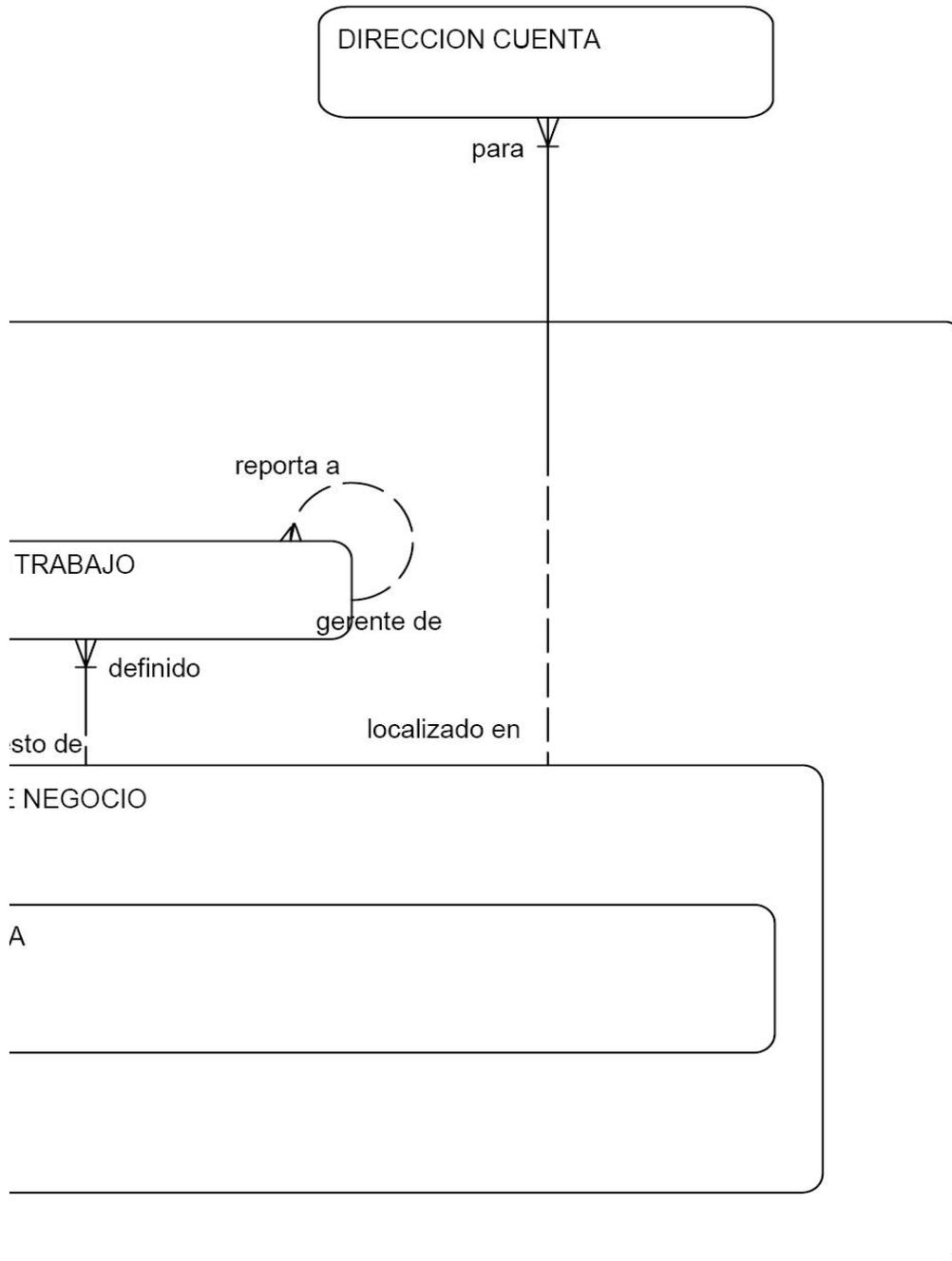


Figura 4.2.2.6 Diagrama Entidad Relación Contactos (Continuación)

PRODUCTO

En esta entidad se almacena el catalogo de productos y líneas de producto están asociadas a una unidad de negocio

ENTIDAD	TABLA
PRODUCTO INTERNO	S_PROD_INT
CATALOGO	S_CTLG
PUESTO DE TRABAJO	S_POSTN
UNIDAD DE NEGOCIO	S_BU, S_PARTY
CUENTA	S_ORG_EXT, S_PARTY

Tabla 4.2.2.4 Tablas Entidad Producto

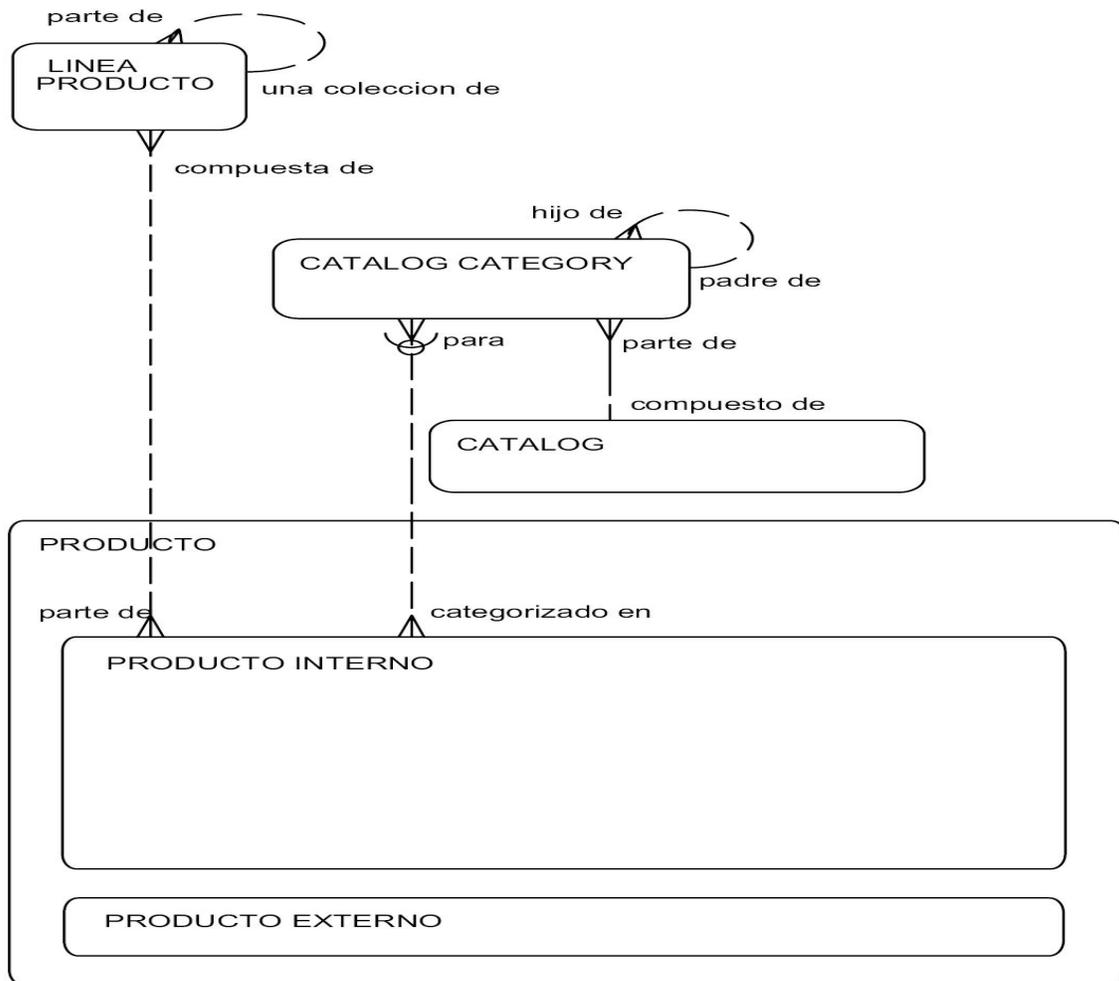


Figura 4.2.2.7 Diagrama Entidad Relación Producto

VISIBILIDAD DE DATOS

Esta entidad controla el acceso a datos mediante puesto de trabajo y usuario

ENTIDAD	TABLA
CUENTA	S_ORG_EXT, S_PARTY
CONTACTOS	S_CONTACT, S_PARTY,
ACTIVIDAD	S_EVT_ACT
PARTY	S_PARTY
EMPLEADO	S_EMP_PER,S_CONTACT, S_PARTY
PUESTO	S_POSTN
UNIDAD DE NEGOCIO	S_BU, S_PARTY
UNIDAD DE NEGOCIO INTERNA	S_ORG_EXT, S_PARTY
PRODUCTO	S_PROD_INT

Tabla 4.2.2.5 Tablas para Visibilidad de Datos.

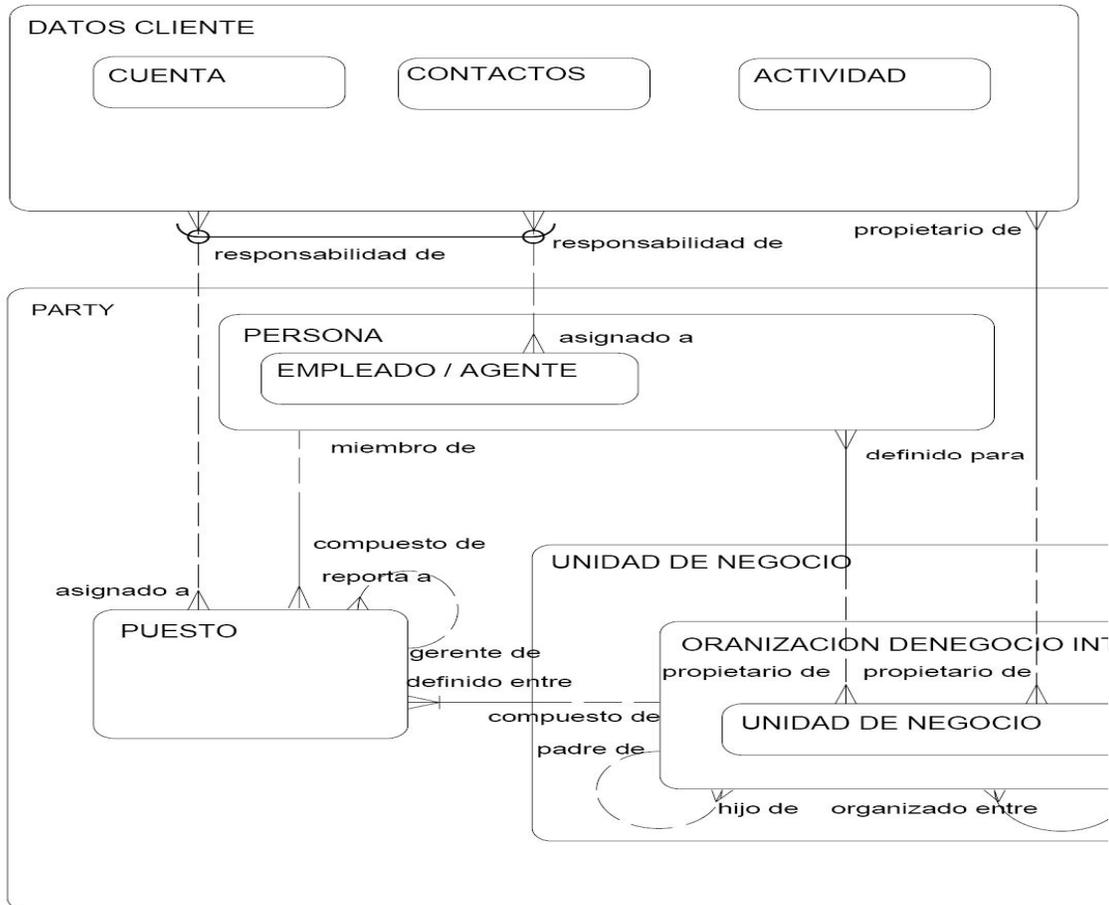


Figura 4.2.2.8 Diagrama Entidad Relación Visibilidad de Datos



Figura 4.2.2.9 Diagrama Entidad Relación Visibilidad de Datos (Continuación).

4.3 Diccionario de Datos.

Conjunto central de tablas y vistas de la base de datos de solo lectura. (BD Maestra)

Un diccionario de datos proporciona la siguiente información:

- Estructura lógica y física de la BD.
- Nombre de los usuarios
- Privilegios y roles que tienen los usuarios
- Definiciones de los objetos de la BD (Tablas, Vistas, índices, Sinónimos, Clúster, Secuencias, Procedimientos, Triggers, Tablespaces, Funciones, etc.)
- Auditoría de información, como los accesos a los objetos
- Espacio ocupado por los objetos

Un diccionario de datos debe cumplir las siguientes características:

- Debe soportar las descripciones de los modelos conceptual, lógico, interno y externo de la BD.
- Debe estar integrado dentro del Sistema Gestor de BD
- Debe apoyar la transferencia eficiente de información al Sistema Gestor de BD. La conexión entre los modelos interno y externo debe ser realizada en tiempo de ejecución.
- Debe comenzar con la reorganización de versiones de producción de la BD. Además debe reflejar los cambios en la descripción de la BD. Cualquier cambio a la descripción de programas ha de ser reflejado automáticamente en la librería de descripción de programas con la ayuda del diccionario de datos.
- Debe estar almacenado en un medio de almacenamiento con acceso directo para la fácil recuperación de información.

A continuación se mostrará la descripción de tablas más representativas de los modelos Entidad Relación descritos. Debido a lo extenso de los campos solo se muestran algunos campos como forma representativa de cada tabla.

S_BU

Tabla de Unidad de Negocio

NOMBRE	ACRONIMO	PK O FK	NOT NULL	TABLA FORANEA	TIPO DATO	LONG	DESC.
ID UNICO	ROW_ID	PK	Y		Varchar	15	ID UNICO TABLA
ID CONFLICTO	CONFLICT_ID		Y		Varchar	15	RESERVADO SISTEMA
FECHA CREACION	CREATED		Y		UTC Date Time	7	
CREADO POR	CREATED_BY	FK	Y	S_USER	Varchar	15	
FECHA ULTIMA ACT DB	DB_LAST_UPD		N		UTC Date Time	7	RESERVADO SISTEMA
FUENTE ULTIMA ACT DB	DB_LAST_UPD_SRC		N		Varchar	50	RESERVADO SISTEMA
ID JERARQUÍA DINAMICA	DYN_HRCHY_ID	FK	N	S_DYN_HRCHY	Varchar	15	ID PARA JERARQUÍA DE UNIDADES DE NEGOCIO
FECHA ULTIMA ACT	LAST_UPD		Y		UTC Date Time	7	RESERVADO SISTEMA
ULTIMA ACT POR	LAST_UPD_BY	FK	Y	S_USER	Varchar	15	RESERVADO SISTEMA
NUM MODIFICACION	MODIFICATION_NUM		Y		Number	22	RESERVADO SISTEMA
NOMBRE	NAME		Y		Varchar	100	NOMBRE ASIGNADO A LA UNIDAD DE NEGOCIO
ID PADRE	PAR_ROW_ID	FK	Y	S_PARTY	Varchar	15	ID DE LA UNIDAD DE NEGOCIO PADRE

Tabla 4.3.1. Descripción Tabla S_BU

S_CONTACT. Tabla de contactos

NOMBRE	ACRONIMO	PK O FK	NOT NULL	TABLA FORANEA	TIPO DATO	LONG	DESC.
ID UNICO	ROW_ID	PK	Y		Varchar	15	ID UNICO TABLA
ID PADRE	PAR_ROW_ID	FK	Y	S_PARTY	Varchar	15	ID TABLA PADRE
ID DE PERSONA	PERSON_UID		Y		Varchar	100	ID UNICO DE CONTACTO USADO PARA INTEGRACION
TITULO PERSONAL	PER_TITLE		N		Varchar	15	POR EJEMPLO SR, SRA, LIC, ING, DR.
LUGAR NACIMIENTO	PLACE_OF_BIRTH		N		Varchar	100	LUGAR DE NACIMIENTO DEL CONTACTO
BANDERA PRIVADO	PRIV_FLG		Y		Character	1	CONTACTO PRIVADO DEL CREADOR
BANDERA DE PROSPECTO	PROSPECT_FLG		Y		Character	1	INDICA SI EL CONTACTO ES PROSPECTO
BANDERA DE PROVEEDOR	PROVIDER_FLG		N		Character	1	INDICA SI LA PERSONA ES PROVEEDOR
CÓDIGO DE ESTATUS	STATUS_CD		N		Varchar	30	ESTATUS DEL CONTACTO, ACTIVO, INACTVO, ETC.
ID ESPECIALIDAD	SUB_SPEC_ID	FK	N	S_MED_SPEC	Varchar	15	ID DE LA ESPECIALIDAD
BANDERA NO LLAMAR	SUPPRESS_CALL_FLG		N		Character	1	INDICA NO LLAMAR TELEF.
BANDERA NO E-MAIL	SUPPRESS_EMAIL_FLG		Y		Character	1	INDICA NO MANDAR E-MAIL

Tabla 4.3.2 Descripción Tabla S_CONTACT

S_EMP_PER. Tabla de empleados

NOMBRE	ACRONIMO	PK O FK	NOT NULL	TABLA FORANEA	TIPO DATO	LONG	DESC.
ID CONFLICTO	CONFLICT_ID		Y		Varchar	15	RESERVADO SISTEMA
FECHA CREACION	CREATED		Y		UTC Date Time	7	RESERVADO SISTEMA
CREADO POR	CREATED_BY		Y	S_USER	Varchar	15	RESERVADO SISTEMA
CÓDIGO DISPONIBLE	CURR_AVAIL_CD		N		Varchar	30	INDICA EMPLEADO DISPONIBLE
DISPONIBLE HASTA	CURR_AVAIL_UNTIL		N		UTC Date Time	7	INIDICA FECHA PARA DISPONIBILIDAD
PORCENTAJE BONO	CURR_BONUS_TGT_PC T		N		Number	22	INDICA EL PORCENTAJE DE BONO PARA COMPENSACION
SUELDO ACTUAL	CURR_SALARY_AMT		N		Number	22	INDICA EL SUELDO DEL EMPLEADO
FECHA ULTIMA ACT DB	DB_LAST_UPD		N		UTC Date Time	7	RESERVADO SISTEMA
FUENTE ULTIMA ACT DB	DB_LAST_UPD_SRC		N		Varchar	50	RESERVADO SISTEMA
	EMERG_NOTIFY_CD		N		Varchar	30	
TIPO EMPLEADO	EMPLOYEE_TYPE_CD		N		Varchar	30	INIDICA EL TIPO DE EMPLEADO
ESTATUS	EMP_STAT_CD		N		Varchar	30	INDICA EL ESTATUS DEL EMPLEADO
FECHA CONTRATACION	HIRE_DT		N		Date	7	INDICA LA FECHA DE CONTRATACION
FECHE ULTIM ACT	LAST_UPD		Y		UTC Date Time	7	RESERVADO SISTEMA
ULTIM ACT POR	LAST_UPD_BY	FK	Y	S_USER	Varchar	15	RESERVADO SISTEMA

Tabla 4.3.3. Descripción Tabla S_EMP_PER

S_EVT_ACT. Tabla de actividades

NOMBRE	ACRONIMO	PK O FK	NOT NULL	TABLA FORANEA	TIPO DATO	LONG	DESC.
ID ACTIVIDAD	ACTIVITY_UID		Y		Varchar	30	ID ACTIVIDAD
ACTIVIDAD CREADA POR	ACT_CREATED_BY	FK	N	S_USER	Varchar	15	CREACION ACTIVIDAD POR USUARIO
FECHA CREACION ACTIVIDAD	ACT_CREATED_DT		N		UTC Date Time	7	FECHA CREACION ACTIVIDAD
ID PLANTILLA ACTIVIDAD	ACT_TMPL_ID	FK	N	S_EVT_ACT	Varchar	15	ID PLANTILLA DE ACTIVIDAD
BANDERA ALARMA	ALARM_FLAG		Y		Character	1	INDICA QUE TIENE ALARMA
DURACION VISITA MINUTOS	APPT_DURATION_MIN		N		Number	22	INDICA LOS MINUTOS DE DURACION DE VISITA
FECHA INICIO	APPT_START_DT		N		UTC Date Time	7	INDICA LA FECHA DE INICIO ACTIVIDAD
HORA INICIO	APPT_START_TM		N		Time	7	INDICA LA HORA DE INICIO ACTIVIDAD
FECHA ENVIO VISITA	CALL_SUBMIT_DT		N		UTC Date Time	7	INDICA LA FECHA DE PRESENTACION DE VISITA.
TIPO ACTIVIDAD	CAL_TYPE_CD		N		Varchar	30	INDICA EL TIPO DE ACTIVIDAD
COMENTARIOS	COMMENTS_LONG		N		Varchar	1,500	INDICA LOS COMENTARIOS
FECHA CREACION	CREATED		Y		UTC Date Time	7	RESERVADO SISTEMA
CREADO POR	CREATED_BY	FK	Y	S_USER	Varchar	15	RESERVADO SISTEMA
FECHA ULTIMA ACT DB	DB_LAST_UPD		N		UTC Date Time	7	RESERVADO SISTEMA
FUENTE ULTIMA ACT DB	DB_LAST_UPD_SRC		N		Varchar	50	RESERVADO SISTEMA

Tabla 4.3.4. Descripción Tabla S_EVT_ACT

S_ORG_EXT. Tabla de organizaciones internas y externas.

NOMBRE	ACRONIMO	PK O FK	NOT NULL	TABLA FORANEA	TIPO DATO	LONG	DESCRIPCION
ID UNICO	ROW_ID	PK	Y		Varchar	15	RESERVADO SISTEMA
CUENTA PRIVADA	PRIVACY_CD		N		Varchar	30	
GRUPO PRIMARIO	PRI_GRP_CD		N		Varchar	30	
BANDERA PROSPECTO	PROSPECT_FLG		Y		Character	1	SI LA CUENTA ES PROSPECTO
BANDERA PARTNER	PRTNR_FLG		Y		Character	1	
ID PARTNER	PRTNR_ORG_INT_ID	FK	N	S_ORG_EXT	Varchar	15	
TIPO PARTNER	PRTNR_TYPE_CD		N		Varchar	30	
PAQUETERIA PREFERIDA	PREF_SHIP_CARR_CD		N		Varchar	30	INDICA EL CÓDIGO DE PAQUETERIA A UTILIZAR A LA CUENTA
PRIORIDAD ENTREGA	PREF_CARR_PRIO_CD		N		Varchar	30	INDICA EL CÓDIGO DE PRIORIDAD
E MAIL PRIMARIA	MAIN_EMAIL_ADDR		N		Varchar	50	EMAIL PRIMARIO DE LA CUENTA
FAX PRIMARIO	MAIN_FAX_PH_NUM		N		Varchar	40	NÚMERO DE FAX PRIMARIO
TELEFONO PRIMARIO	MAIN_PH_NUM		N		Varchar	40	TELEF. PRIMARIO CUENTA
CLASE MERCADO	MARKET_CLASS_CD		N		Varchar	30	CLASE MERCADO CUENTA
TIPO MERCADO	MARKET_TYPE_CD		N		Varchar	30	TIPO MERCADO CUENTA
CUENTA PRINCIPAL	MASTER_OU_ID	FK	N	S_ORG_EXT	Varchar	15	CUENTA PADRE
NÚMERO MODIFICACION	MODIFICATION_NUM		Y		Number	22	RESERVADO SISTEMA
NOMBRE DE LA CUENTA	NAME		Y		Varchar	100	NOMBRE ORGANIZACIÓN

Tabla 4.3.5. Descripción Tabla S_ORG_EXT

S_PARTY. Tabla para almacenar instancias Party como son personas, puestos organizaciones.

NOMBRE	ACRONIMO	PK O FK	NOT NULL	TABLA FORANEA	TIPO DATO	LON G	DESC
ID UNICO	ROW_ID	PK	Y		Varchar	15	ID UNICO TABLA
ID CONFLICTO	CONFLICT_ID		Y		Varchar	15	RESERVADO SISTEMA
FECHA CREACION	CREATED		Y		UTC Date Time	7	RESERVADO SISTEMA
CREADO POR	CREATED_BY	FK	Y	S_USER	Varchar	15	RESERVADO SISTEMA
FECHA ULTIM ACT BD	DB_LAST_UPD		N		UTC Date Time	7	RESERVADO SISTEMA
FUENTE ULTIM ACT BD	DB_LAST_UPD_SRC		N		Varchar	50	RESERVADO SISTEMA
FECHA ULTIM ACT	LAST_UPD		Y		UTC Date Time	7	RESERVADO SISTEMA
ULTIM ACT POR	LAST_UPD_BY	FK	Y	S_USER	Varchar	15	RESERVADO SISTEMA
NUM MODIFICACION	MODIFICATION_NUM		Y		Number	22	RESERVADO SISTEMA
NOMBRE	NAME		Y		Varchar	150	NOMBRE A DESPLEGAR DEL PARTY
CÓDIGO TIPO PARTY	PARTY_TYPE_CD		Y		Varchar	30	TIPO DE PARTY
ID DE PARTY	PARTY_UID		Y		Varchar	100	ID DE INTEGRACION DE PARTY
ID PARTY PADRE	PAR_PARTY_ID	FK	N	S_PARTY	Varchar	15	ID DE PARTY PADRE

Tabla 4.3.6. Descripción Tabla S_PARTY

S_PARTY_REL

Tabla intersección para relaciones entre personas, organizaciones y puestos (PARTY)

NOMBRE	ACRONIMO	PK O FK	NOT NULL	TABLA FORANEA	TIPO DATO	LONG	DESC
ID UNICO	ROW_ID	PK	Y		Varchar	15	ID UNICO TABLA
ID CONFLICTO	CONFLICT_ID		Y		Varchar	15	RESERVADO SISTEMA
FECHA CREACION	CREATED		Y		UTC Date Time	7	RESERVADO SISTEMA
CREADO POR	CREATED_BY	FK	Y	S_USER	Varchar	15	RESERVADO SISTEMA
FECHA ULTIM ACT DB	DB_LAST_UPD		N		UTC Date Time	7	RESERVADO SISTEMA
FUENTA ULTIM ACT DB	DB_LAST_UPD_SRC		N		Varchar	50	RESERVADO SISTEMA
FECHA ULTIM ACT	LAST_UPD		Y		UTC Date Time	7	RESERVADO SISTEMA
ULTIM ACT POR	LAST_UPD_BY	FK	Y	S_USER	Varchar	15	RESERVADO SISTEMA
NUM MODIFICACION	MODIFICATION_NUM		Y		Number	22	RESERVADO SISTEMA
ID PARTY	PARTY_ID		Y		Varchar	100	ID
ID DE PARTY	PARTY_ID	FK	Y	S_PARTY	Varchar	15	ID DE PARTY AL QUE SE RELACIONA
FECHA INICIO	STAQRT_DT		N		UTC Date Time	7	FECHA INICIO RELACION
FECHA FIN	END_DT		N		UTC Date Time	7	FECHA FIN RELACION
ID RELACION PARTY	REL_PARTY_ID	FK	Y	S_PARTY	Varchar	15	ID DEL PARTY RELACIONADO

Tabla 4.3.7. Descripción Tabla S_PARTY_REL

S_POSTN. Tabla de puestos de trabajo o rutas

NOMBRE	ACRONIMO	PK O FK	NOT NULL	TABLA FORANEA	TIPO DATO	LONG	DESC
ID UNICO	ROW_ID	PK	Y		Varchar	15	ID UNICO TABLA
FECHA INICIO	START_DT		N		Date	7	FECHA EFECTIVA DE COMIENZO DEL PUESTO
ID CUENTA PRIMARIA	PR_OU_EXT_ID	FK	N	S_ORG_EXT	Varchar	15	ASIGNACION PRIMARIA DEL PUESTO EN LA CUENTA
ID EMPLEADO PRIMARIO	PR_EMP_ID	FK	N	S_USER	Varchar	15	EMPLEADO PRIMARIO QUE OCUPA EL PUESTO
CÓDIGO TIPO PUESTO	POSTN_TYPE_CD		N		Varchar	30	CÓDIGO DE TIPO DE PUESTO
ID TABLA PADRE	PAR_ROW_ID	FK	Y	S_PARTY	Varchar	15	ID TABLA PADRE PRINCIPAL
ID PUESTO PADRE	PAR_POSTN_ID	FK	N	S_POSTN	Varchar	15	ID PUESTO PADRE
ID DIVISION	OU_ID	FK	Y	S_ORG_EXT	Varchar	15	DIVISION EN LA QUE ESTA ASIGNADO EL PUESTO
NOMBRE	NAME		Y		Varchar	50	NOMBRE PUESTO DE TRABAJO
ULTIM ACT POR	LAST_UPD_BY	FK	Y	S_USER	Varchar	15	RESERVADO SISTEMA
FECHA ULTIM ACT	LAST_UPD		Y		UTC Date Time	7	RESERVADO SISTEMA
ID INTEGRACION	INTEGRATION_ID		N		Varchar	30	RESERVADO SISTEMA

Tabla 4.3.8. Descripción Tabla S_POSTN

S_POSTN_CON. Tabla de intersección entre S_CONTACT y S_POSTN

NOMBRE	ACRONIMO	PK O FK	NOT NULL	TABLA FORANEA	TIPO DATO	LONG	DESC
BANDERA ACTIVO	ACTIVE_FLG		N		Character	1	INDICA SI EL CONTACTO ESTA ACTIVO EN EL PUESTO
ID CONFLICTO	CONFLICT_ID		N		Varchar	30	RESERVADO SISTEMA
ID CONTACTO	CON_ID	FK	Y	S_CONTACT	Varchar	15	ID DEL CONTACTO
FECHA CREACION	CREATED		Y		Varchar	15	RESERVADO SISTEMA
CREADO POR	CREATED_BY		Y		UTC Date Time	7	RESERVADO SISTEMA
FEHCA ULTIM ACT BD	DB_LAST_UPD	FK	Y	S_USER	Varchar	15	RESERVADO SISTEMA
FUENTE ULTIM ACT BD	DB_LAST_UPD_SRC		N		UTC Date Time	7	RESERVADO SISTEMA
	LAST_CALL_DT		N		Character	1	
FECHA ULTM ACT	LAST_UPD		N		Date	7	RESERVADO SISTEMA
ULTIM ACT POR	LAST_UPD_BY		Y		UTC Date Time	7	RESERVADO SISTEMA
NUM MODIFICACION	MODIFICATION_NUM	FK	Y	S_USER	Varchar	15	RESERVADO SISTEMA
ID PUESTO	POSTN_ID	FK	Y	S_POSTN	Number	22	ID PUESTO
CÓDIGO RELACION	RELATION_CD	FK	N		Varchar	15	TIPO DE RELACION ENTRE PUESTO Y CONTACTO
ID UNICO	ROW_ID	PK	Y		Varchar	30	ID UNICO TABLA
ESPECIALIDAD	SPECIALTY		N		Varchar	10	ESPECIALIDAD DEL PUESTO
ESTADO	STATUS		N		Varchar	30	ESTATUS RELACION

Tabla 4.3.9. Descripción Tabla S_POSTN_CON

S_PROD_INT. Tabla para almacenar productos y/o servicios

NOMBRE	ACRONIMO	PK O FK	NOT NULL	TABLA FORANEA	TIPO DATO	LONG	DESC
BANDERA ACTIVO	ACTIVE_FLG		Y		Character	1	ESTATUS DE ACTIVO
NOMBRE ALIAS	ALIAS_NAME		N		Varchar	100	NOMBRE ALTERNO
CÓDIGO DE BARRAS	BAR_CODE_NUM		N		Varchar	50	NÚMERO DE CÓDIGO DE BARRAS
BANDERA FACTURABLE	BILLABLE_FLG		Y		Character	1	PRODUCTO FACTURABLE
UNIDAD NEGOCIO	BU_ID	FK	Y	S_BU	Varchar	15	UNIDAD DE NEGOCIO PRINCIPAL
CAPACIDAD	CAPACITY		N		Number	22	CAPACIDAD DEL PRODUCTO
CATEGORIA	CATEGORY_CD		N		Varchar	30	CATEGORIA DEL PRODUCTO
COMENTARIOS	COMMENTS		N		Varchar	250	COMENTARIOS
ID CONFLICTO	CONFLICT_ID		Y		Varchar	15	RESERVADO SISTEMA
FECHA CREACION	CREATED		Y		UTC Date Time	7	RESERVADO SISTEMA
CREADO POR	CREATED_BY	FK	Y	S_USER	Varchar	15	RESERVADO SISTEMA
FECHA ULTIM ACT BD	DB_LAST_UPD		N		UTC Date Time	7	RESERVADO SISTEMA
FUENTE ULTIM ACT BD	DB_LAST_UPD_SRC		N		Varchar	50	RESERVADO SISTEMA
INTERVALO DE ENTREGA	DELIV_INTERVAL		N		Number	22	TIEMPO DE ENTREGA
DESCRIPCION	DESC_TEXT		N		Varchar	255	DESCRIPCION DEL PRODUCTO
TIPO PRODUCTO	DETAIL_TYPE_CD		N		Varchar	30	TIPO DE PRODUCTO

Tabla 4.3.10. Descripción Tabla S_PORD_INT

S_PROD_LN

Tabla para almacenar Líneas de productos

NOMBRE	ACRONIMO	PK O FK	NOT NULL	TABLA FORANEA	TIPO DATO	LONG	DESC
ID UNICO	ROW_ID	PK	Y		Varchar	15	RESERVADO DE SISTEMA
ID CONFLICTO	CONFLICT_ID		Y		Varchar	15	RESERVADO DE SISTEMA
FECHA CREACION	CREATED		Y		UTC Date Time	7	RESERVADO DE SISTEMA
CREADO POR	CREATED_BY	FK	Y	S_USER	Varchar	15	RESERVADO DE SISTEMA
FECHA ULTIM ACT BD	DB_LAST_UPD		N		UTC Date Time	7	RESERVADO DE SISTEMA
FUENTA ULTIM ACT BD	DB_LAST_UPD_SRC		N		Varchar	50	RESERVADO DE SISTEMA
DESCRIPCION	DESC_TEXT		N		Varchar	255	TEXTO DE DESCRIPCION
ID INTEGRACION	INTEGRATION_ID		N		Varchar	30	ID PARA INTERFASES
FECHA ULTIMA ACT	LAST_UPD		Y		UTC Date Time	7	RESERVADO DE SISTEMA
ULTIMA ACT POR	LAST_UPD_BY	FK	Y	S_USER	Varchar	15	RESERVADO DE SISTEMA
NUM MODIFICACION	MODIFICATION_NUM		Y		Number	22	RESERVADO DE SISTEMA
NOMBRE LINEA	NAME		Y		Varchar	100	NOMBRE DE LA LINEA
ID LINEA PADRE	PAR_PROD_LN_ID	FK	N	S_PROD_LN	Varchar	15	ID DE LA LINEA PADRE

Tabla 4.3.11. Descripción Tabla S_PROD_LN

S_USER

Tabla para almacenar usuarios, tabla 1:1 de S_PARTY

NOMBRE	ACRONIMO	PK O FK	NOT NULL	TABLA FORANEA	TIPO DATO	LONG	DESC
COMENTARIOS	COMMENTS		N		Varchar	250	
ID CONFLICTO	CONFLICT_ID		Y		Varchar	15	RESERVADO SISTEMA
FECHA CREACION	CREATED		Y		UTC Date Time	7	RESERVADO SISTEMA
CREADO POR	CREATED_BY	FK	Y	S_USER	Varchar	15	RESERVADO SISTEMA
FECHA ULTIMA ACT DB	DB_LAST_UPD		N		UTC Date Time	7	RESERVADO SISTEMA
FUENTE ULTIMA ACT DB	DB_LAST_UPD_SRC		N		Varchar	50	RESERVADO SISTEMA
FECHA ULTIMO LOGIN	LAST_LOGIN_TS		N		UTC Date Time	7	FECHA DEL ULTIMO LOGIN
FECHA ULTIMA ACTUALIZACION	LAST_UPD		Y		UTC Date Time	7	RESERVADO SISTEMA
ULTIMA ACTUALIZACION POR	LAST_UPD_BY	FK	Y	S_USER	Varchar	15	RESERVADO SISTEMA
ID DE USUARIO	LOGIN		Y		Varchar	50	
NUM MODIFICACION	MODIFICATION_NUM		Y		Number	22	RESERVADO SISTEMA
ID PADRE	PAR_ROW_ID	FK	Y	S_PARTY	Varchar	15	
CONTRASEÑA	PASSWORD		N		Varchar	30	
ULTIMA ACTUALIZACION PASSWORDS	PW_LAST_UPD		N		UTC Date Time	7	FECHA DE UTILMA ACT PASS
ID UNICO	ROW_ID	PK	Y		Varchar	15	ID UNICO TABLA
BANDERA USUARIO	USER_FLG		Y		Character	1	BANDERA DE USUARIO

Tabla 4.3.12. Descripción Tabla S_USER

4.4 Diseño y construcción del Back-End

- Bienvenido a Oracle - haga clic en Install/Deinstall Products
- En la ventana Método de Instalación, elija Instalación Básica y haga clic siguiente, vea figura 4.4.1

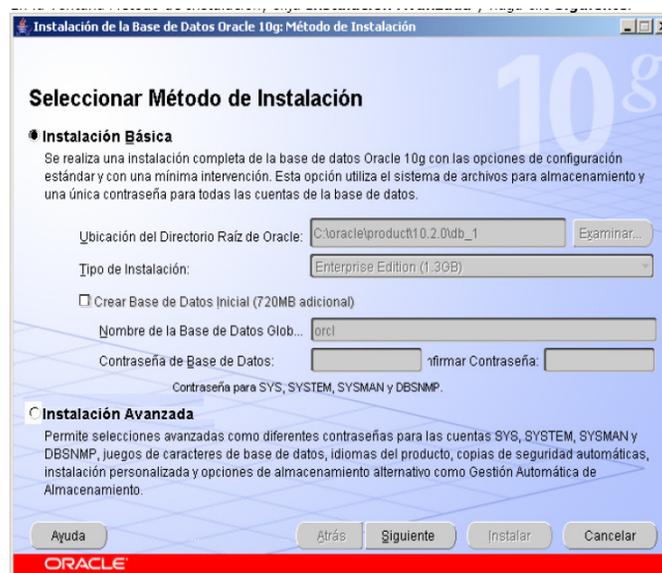
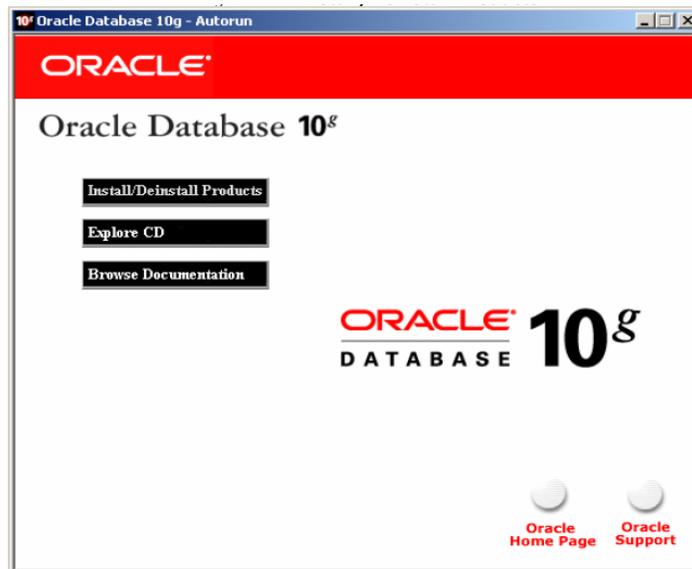


Figura 4.4.1 Pantallas de instalación Oracle

- En la ventana Selección Tipo de Instalación, elija Enterprise Edition y haga clic siguiente

- En el siguiente paso habrá que especificar Detalles de Directorio Raíz y continúe.
- En seguida deberá de Comprobar los Requisitos Específicos del Producto que propone la instalación, verifique y continúe.
- En la ventana Seleccionar Opción de Instalar sólo software de bases de datos y haga clic Siguiente, vea figura 4.4.2

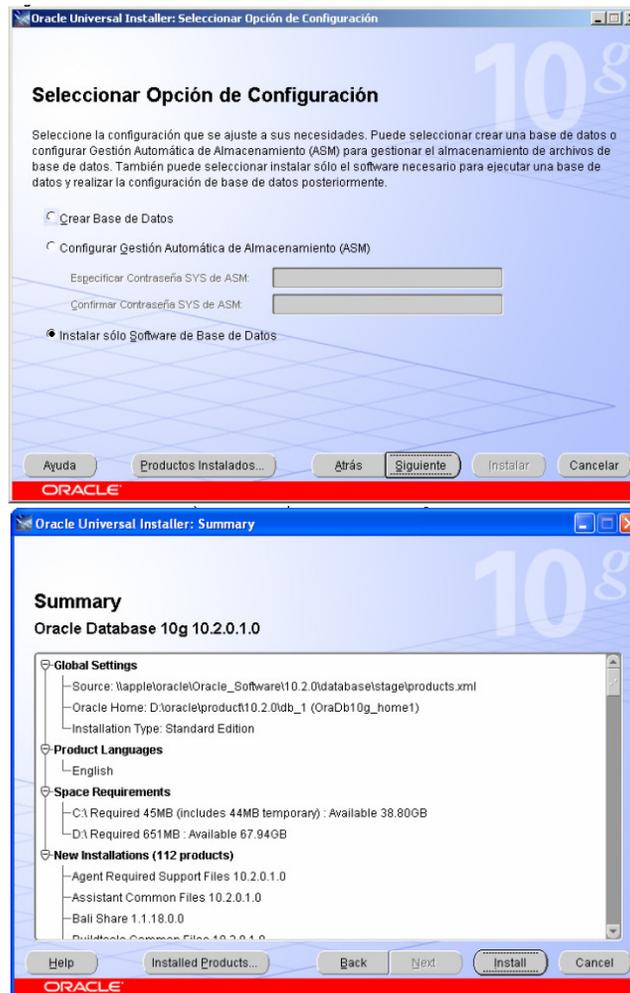
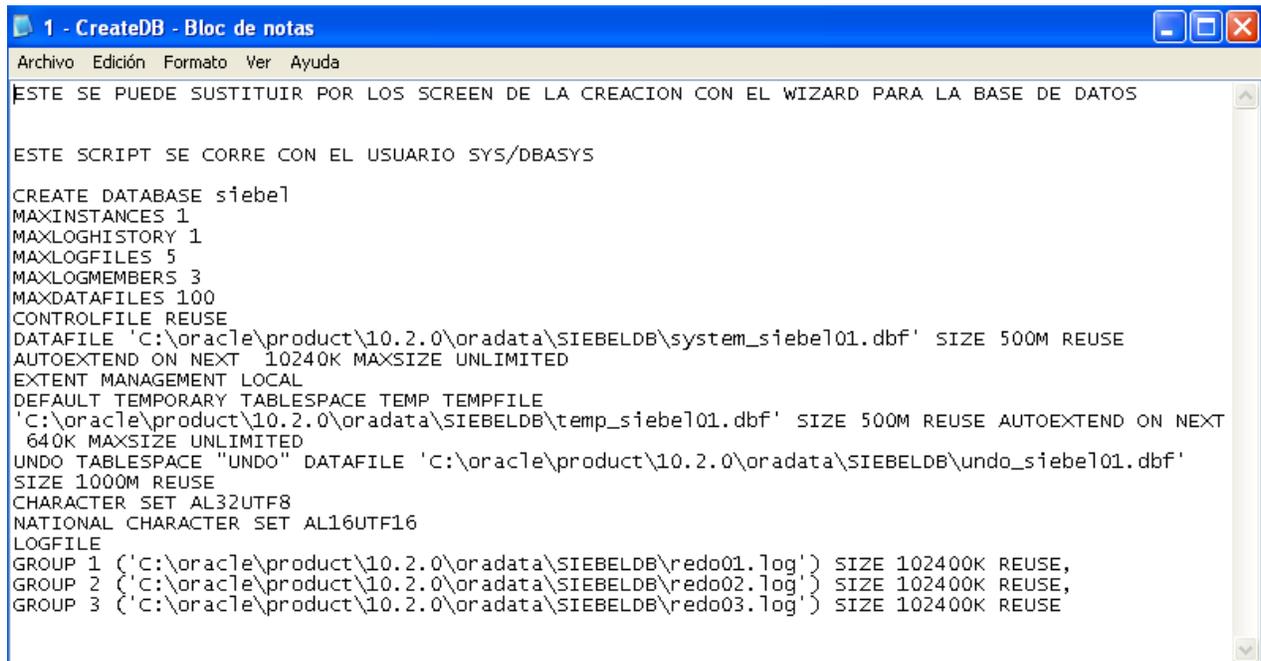


Figura 4.4.2 Pantallas de configuración de Oracle

- En la ventana Seleccionar Configuración de Base de Datos, elija Avanzadas y continúe.
- En la ventana Resumen (mostrado aquí de la versión ingles), haga clic Instalar.

- En la ventana Instalar, mire como la instalación sigue, al finalizar instalaremos la base de datos.
- Este script se puede sustituir por los screen de la creación con el Wizard para la base de datos. Este script se corre con el usuario SYS/DBASYS, vea figura 4.4.3



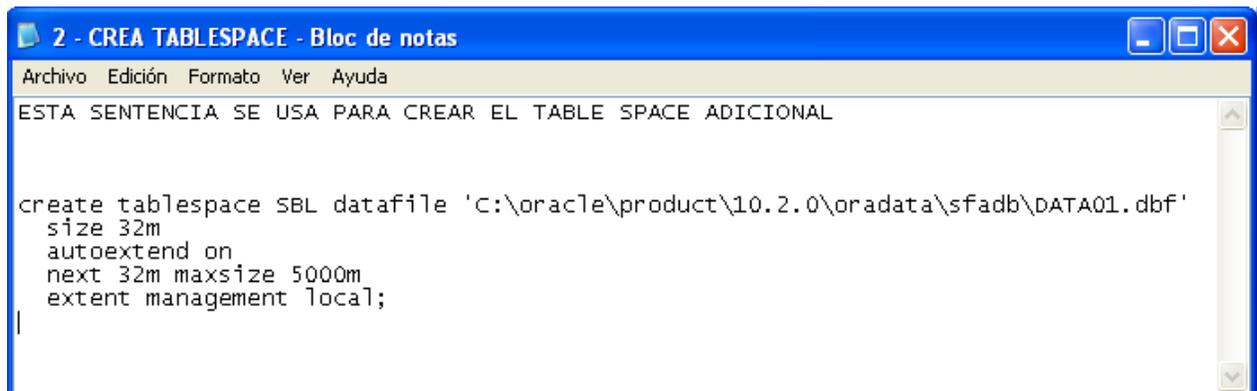
```
1 - CreateDB - Bloc de notas
Archivo Edición Formato Ver Ayuda
ESTE SE PUEDE SUSTITUIR POR LOS SCREEN DE LA CREACION CON EL WIZARD PARA LA BASE DE DATOS

ESTE SCRIPT SE CORRE CON EL USUARIO SYS/DBASYS

CREATE DATABASE siebel
MAXINSTANCES 1
MAXLOGHISTORY 1
MAXLOGFILES 5
MAXLOGMEMBERS 3
MAXDATAFILES 100
CONTROLFILE REUSE
DATAFILE 'C:\oracle\product\10.2.0\oradata\SIEBELDB\system_siebel01.dbf' SIZE 500M REUSE
AUTOEXTEND ON NEXT 10240K MAXSIZE UNLIMITED
EXTENT MANAGEMENT LOCAL
DEFAULT TEMPORARY TABLESPACE TEMP TEMPFILE
'C:\oracle\product\10.2.0\oradata\SIEBELDB\temp_siebel01.dbf' SIZE 500M REUSE AUTOEXTEND ON NEXT
640K MAXSIZE UNLIMITED
UNDO TABLESPACE "UNDO" DATAFILE 'C:\oracle\product\10.2.0\oradata\SIEBELDB\undo_siebel01.dbf'
SIZE 1000M REUSE
CHARACTER SET AL32UTF8
NATIONAL CHARACTER SET AL16UTF16
LOGFILE
GROUP 1 ('C:\oracle\product\10.2.0\oradata\SIEBELDB\redo01.log') SIZE 102400K REUSE,
GROUP 2 ('C:\oracle\product\10.2.0\oradata\SIEBELDB\redo02.log') SIZE 102400K REUSE,
GROUP 3 ('C:\oracle\product\10.2.0\oradata\SIEBELDB\redo03.log') SIZE 102400K REUSE
```

Figura 4.4.3 Código para la creación del wizard para la base de datos

- Este script se usa para crear el TABLE SPACE adicional, vea figura 4.4.4



```
2 - CREA TABLESPACE - Bloc de notas
Archivo Edición Formato Ver Ayuda
ESTA SENTENCIA SE USA PARA CREAR EL TABLE SPACE ADICIONAL

create tablespace SBL datafile 'C:\oracle\product\10.2.0\oradata\sfadb\DATA01.dbf'
size 32m
autoextend on
next 32m maxsize 5000m
extent management local;
|
```

Figura 4.4.4 Código para crear el table space

- Habiendo hecho lo anterior se corre este script, vea figura 4.4.5

```
3 - GRANTUSER.SQL - Bloc de notas
Archivo Edición Formato Ver Ayuda
&&& UNA VEZ CREADA LA BASE DE DATOS SE CORRE ESTE SCRIPT

create role sse_role;
grant create session to sse_role;

rem Create Role tblo_role
create role tblo_role;
grant ALTER SESSION, CREATE CLUSTER, CREATE DATABASE LINK, CREATE INDEXTYPE,
CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE SESSION,
CREATE SYNONYM, CREATE TABLE, CREATE TRIGGER, CREATE TYPE, CREATE VIEW,
SELECT_CATALOG_ROLE, EXECUTE_CATALOG_ROLE, ADVISOR, CREATE DIMENSION,
CREATE MATERIALIZED VIEW, QUERY REWRITE, ON COMMIT REFRESH,
CREATE ANY SYNONYM
to tblo_role;

rem Create SIEBEL user
create user SIEBEL identified by SIEBEL;
grant tblo_role to SIEBEL;
alter user SIEBEL quota 0 on SYSTEM;
alter user SIEBEL default tablespace SBL;
alter user SIEBEL temporary tablespace TEMP;
alter user SIEBEL quota unlimited on SBL;

rem =====
rem Create db accounts for Siebel users
rem

create user SADMIN identified by SADMIN;
grant sse_role to SADMIN;
alter user SADMIN default tablespace SBL;
alter user SADMIN temporary tablespace TEMP;

create user LDAPUSER identified by LDAPUSER;
grant sse_role to LDAPUSER;
alter user LDAPUSER default tablespace SBL;
alter user LDAPUSER temporary tablespace TEMP;
```

Figura 4.4.5 Código para crear la base de datos

Se comienza a generar la estructura de la base de datos cargándose al final los datos semilla sobre la estructura ya creada para tener la aplicación funcional. Metadatos y Datos para entrar a la aplicación, vea figura 4.4.6

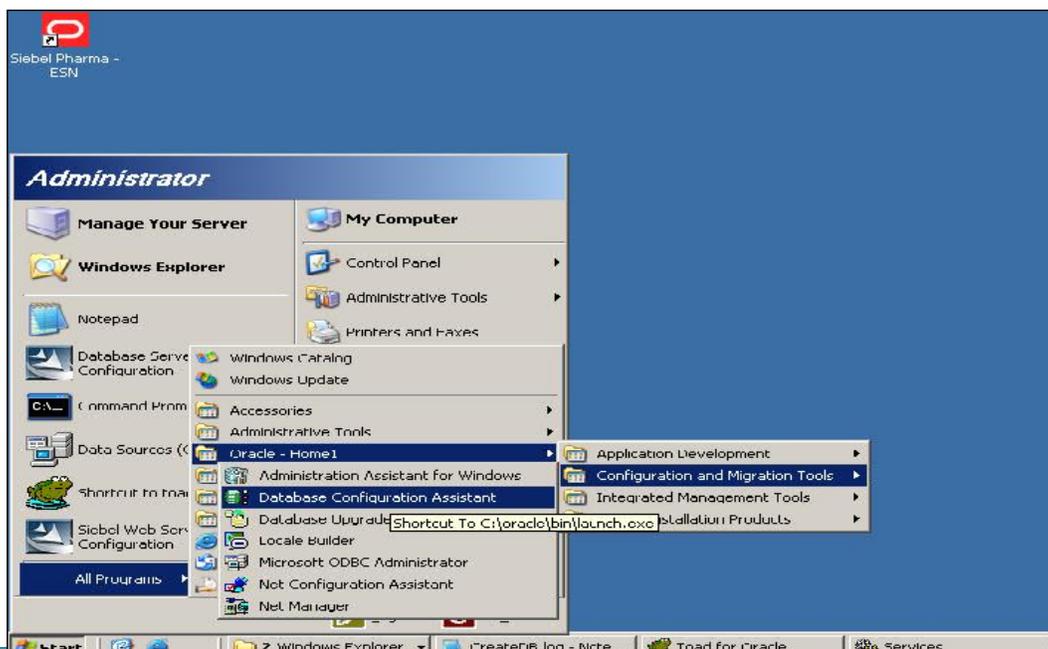


Figura 4.4.6 Selección configuración de la Base de datos

- Se crea la base de datos, se selecciona el template, se da el nombre a la base de datos, el nombre del dominio.
- Continuando con la instalación y por razones de seguridad, debe especificar las contraseñas de las siguientes cuentas de usuario en la base de datos, recordando utilizar la misma contraseña para todas las cuentas.
- En el siguiente paso deberá seleccionar el mecanismo de almacenamiento que se desea utilizar para la base de datos. Seleccionar la opción File System y continuar.
- En el siguiente paso se especificaran las ubicaciones de la base de datos de archivo que se creará. Seleccionar opción Use Database file locations from template y continuar.
- En las siguientes ventanas nos muestra las opciones de recuperación de la base de datos, ver figura 4.4.7

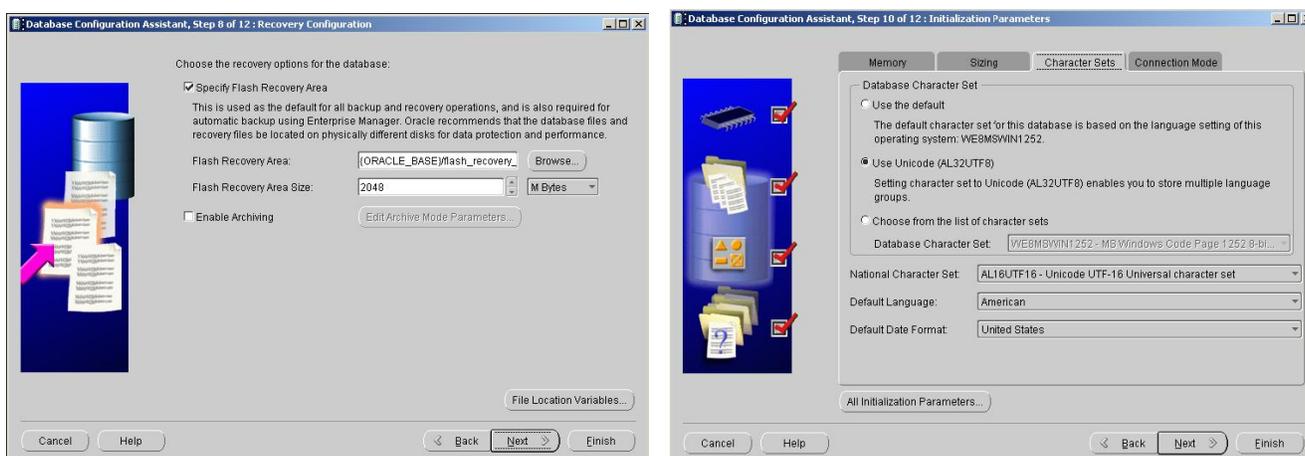


Figura 4.4.7 Pantallas de instalación de la base de datos

- En el siguiente paso se configurara las conexiones remotas que deben introducirse en la base de datos de Oracle debe configurar una red de escucha de Oracle.
- Posteriormente se introduce el nombre del Listener que desea crear y continuar

- Continuando solicita el número de puerto TCP/IP que debe utilizar el listener? el número de puerto seleccionado no debe ser utilizado por otro software en este equipo. Seleccionar la opción de usar el número de puerto estándar de 1521 y continuar.
- Para finalizar pregunta se desea configurar otro listener? Seleccione la opción de NO y continuar.

INSTALAR SIEBEL ENTERPRISE SERVER CONFIGURATION 8.0

- **DATABASE SERVER CONFIGURATION, ver figura 4.4.8**



Figura 4.4.8 Selección de Database Server Configuration

Pasos para configurar: Se pide la dirección del host, el directorio donde será instalado, la plataforma, el script que correrá en la base de datos, el idioma, etc. Además se instalara la seguridad de la base de datos introduciendo nombre y password, se sigue con la configuración de la tabla quedando el siguiente resumen, como se muestra en la siguiente pantalla, ver figura 4.4.9

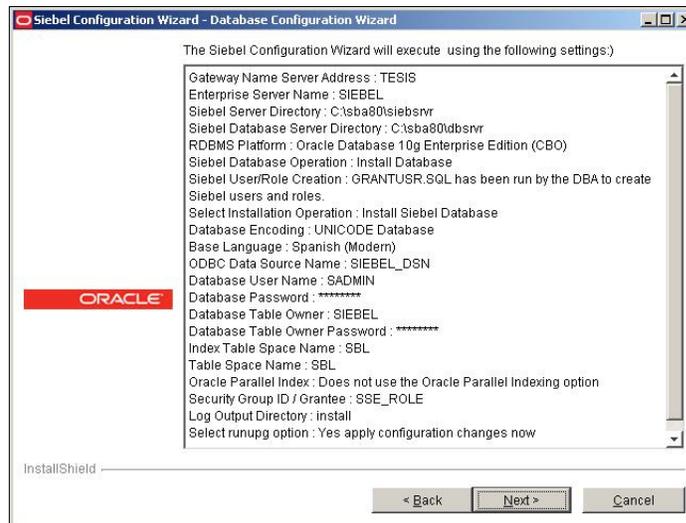


Figura 4.4.9 Pantalla resumen de la instalación

- **SIEBEL ENTERPRISE CONFIGURATION ver figura 4.4.8 1**

En este segmento se configura el nombre de Gateway y el puerto TCP/IP como se muestra en la pantalla, vea figura 4.4.9

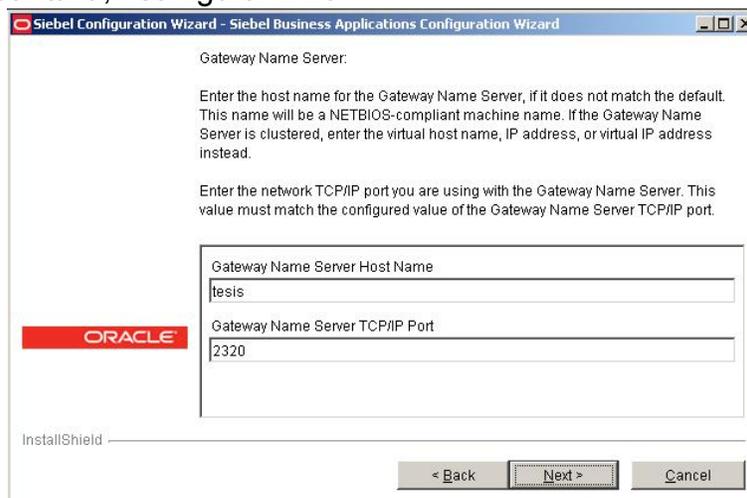


Figura 4.4.9 Nombre de Gateway y TCP/IP

Siebel utiliza el ODBC registrado durante la configuración de la base de datos.

En este caso el ODBC a utilizar es SIEBEL_DSN, el driver utilizado se genera con la instalación del Siebel Enterprise Server, ver figura 4.4.10

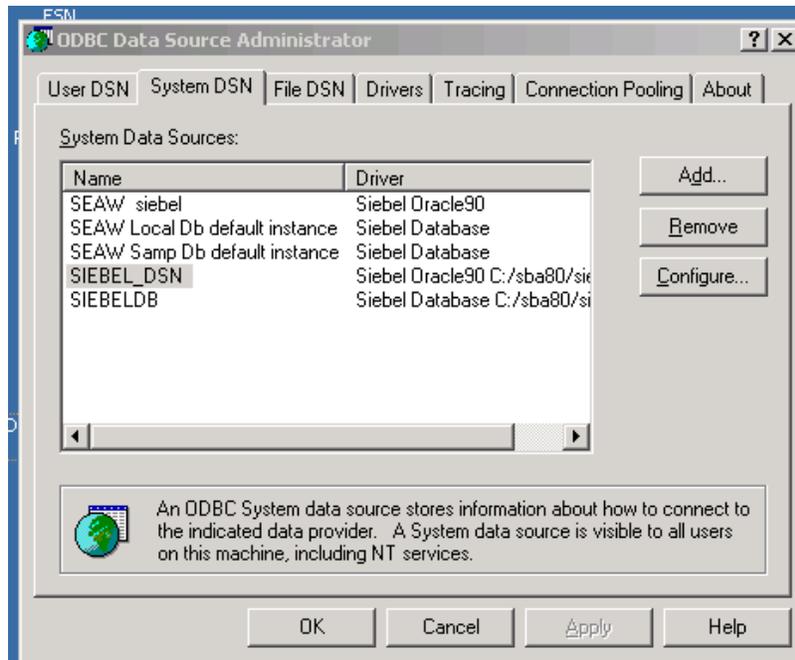


Figura 4.4.10 Pantalla de ODBC

Se muestra la configuración del setup manejador de la Base de datos, ver figura 4.4.11

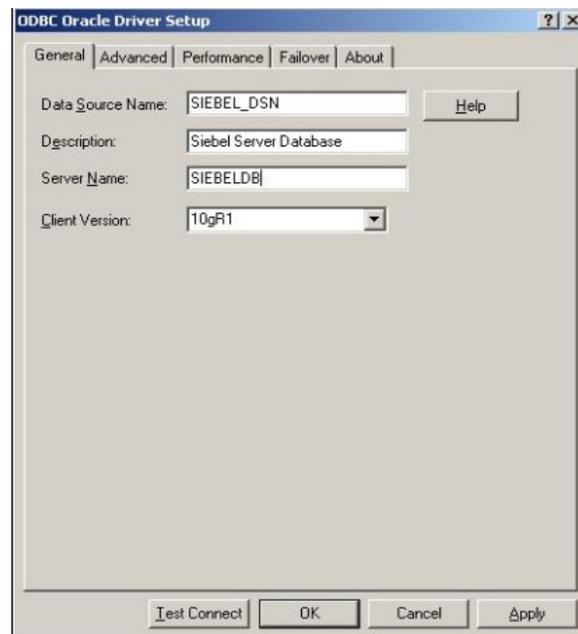


Figura 4.4.11 Pantalla Configuration del setup

SIEBELDB es el nombre del servicio asociado la base de datos y el cual es resuelto a través del tnsnames.ora de Oracle.

De esta manera toda la comunicación que se realiza entre Oracle Siebel y la base de datos de Oracle utiliza esta conexión.

Esta comunicación se realiza dentro Oracle Siebel por el Object Manager que es el programa principal cuyas tareas ejecutan la aplicación. Se muestra un resumen de la aplicación, ver figura 4.4.12

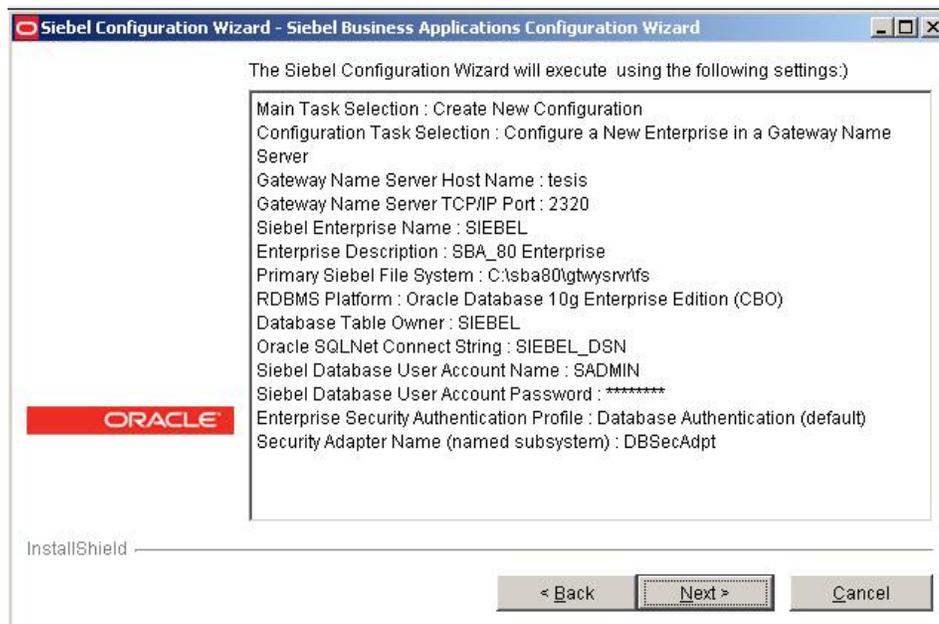


Figura 4.4.12 Pantalla resumen de configuración

Resumen de la configuración del servidor de Siebel, se muestra en la figura 4.4.13

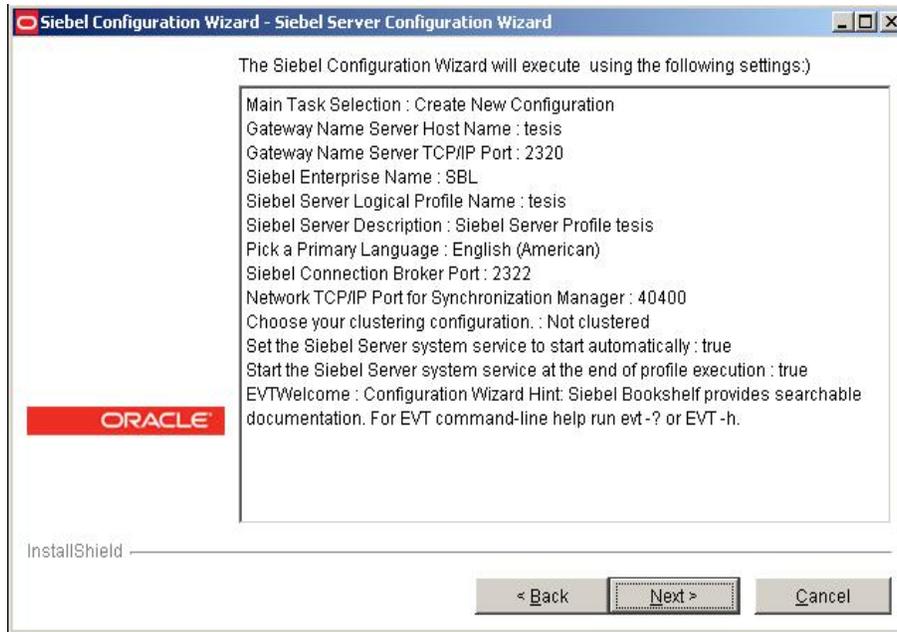


Figura 4.4.13 Pantalla resumen de configuración

Se configura la aplicación webserver y se muestra un resumen, ver figura 4.4.14

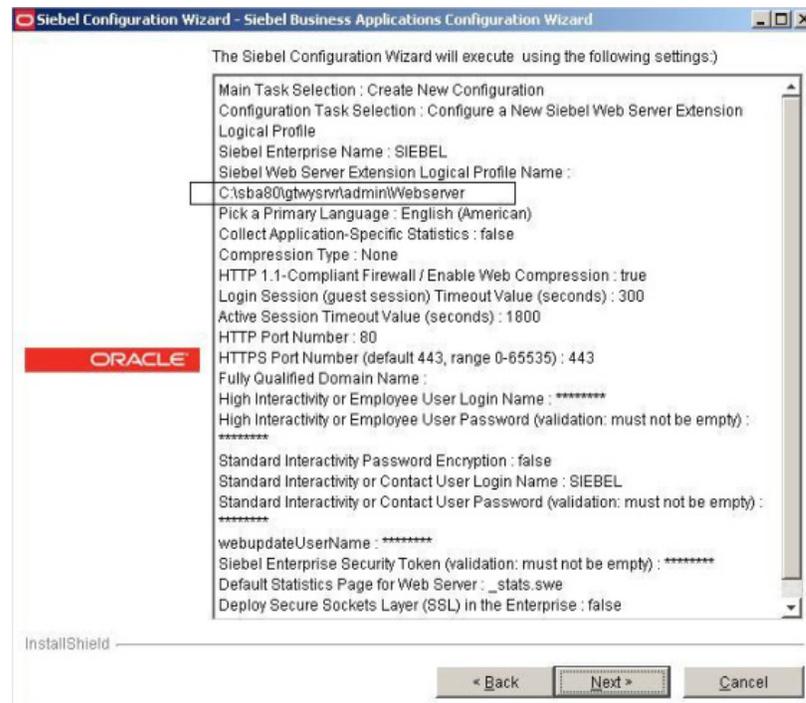


Figura 4.4.14 Pantalla resumen de configuración

- **Sibel Server Configuración, ver figura 4.4.8**



Resumen de la configuración del servidor de Siebel Applications, se muestra en la figura 4.4.15

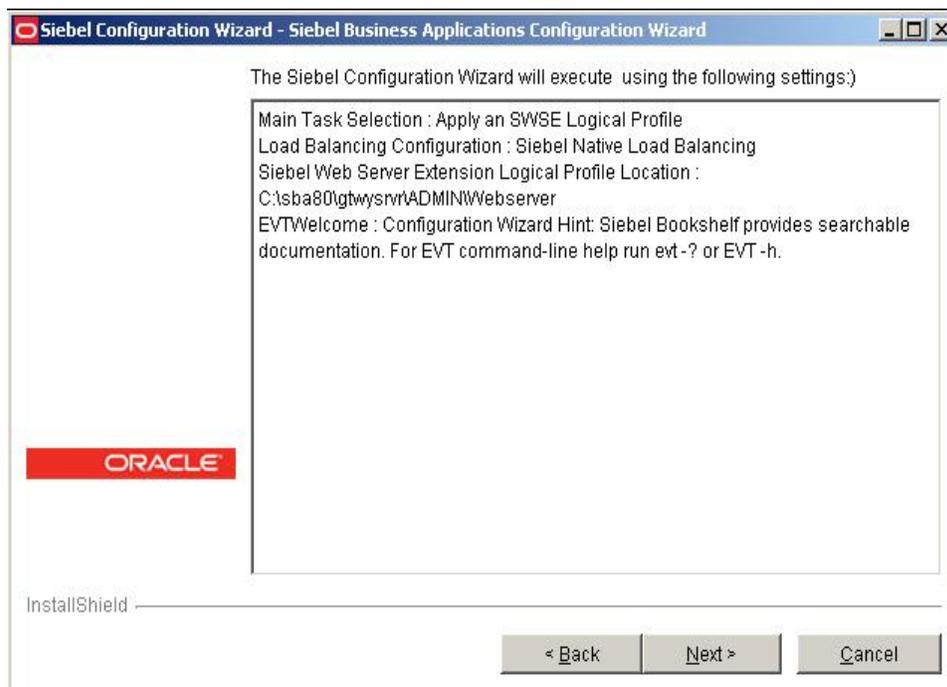


Figura 4.4.15 Pantalla resumen de configuración



- **Web Server Extension Configuration, ver figura 4.4.8**

Oracle Siebel está estructurado en tres capas que permiten su configuración y construcción. Estas capas se mencionan a continuación:

- **Capa de Base de datos:** En esta capa se indican las definiciones de las tablas, relaciones entre ellas, columnas, e índices.
- **Capa de Negocio:** Capa que permite construir las consultas a partir de las Capa de base de datos. En esta capa se incluyen reglas y validaciones que deben de cumplirse para entidades específicas.
- **Capa de Interfaz de usuario:** Esta permite la visualización de los datos principalmente mediante formularios y listados, Vistas y Pantallas compuestos

de controles que contienen la información ejecutada por las consultas de la Capa de Negocio.

La figura siguiente 4.4.16 muestra como se relaciona la capa de Base de Datos con la Capa de Negocio.

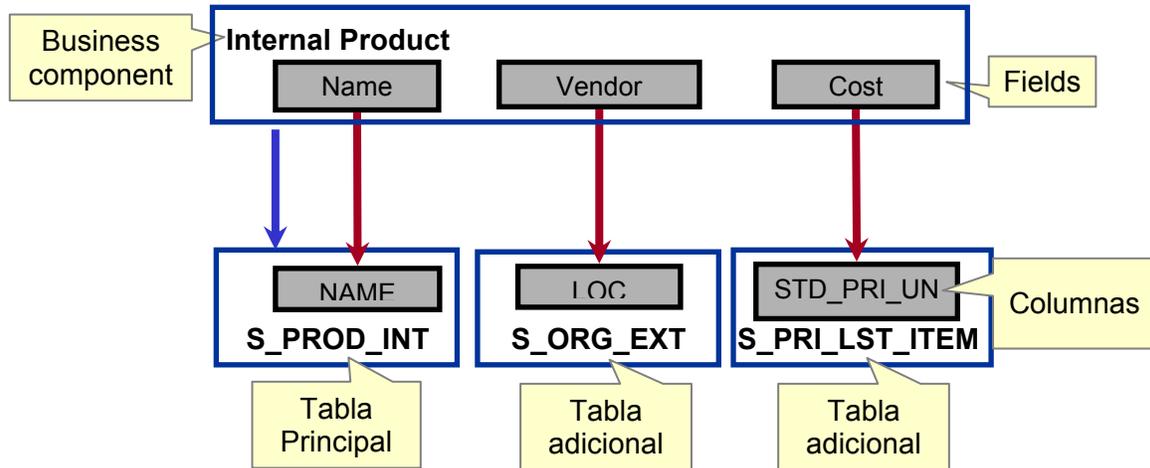


Figura 4.4.16 Relación entre Capa de Base de Datos y Capa de Negocio

La relación entre la capa de negocio y la capa de Interfaz de usuario la podemos apreciar en la figura 4.4.17

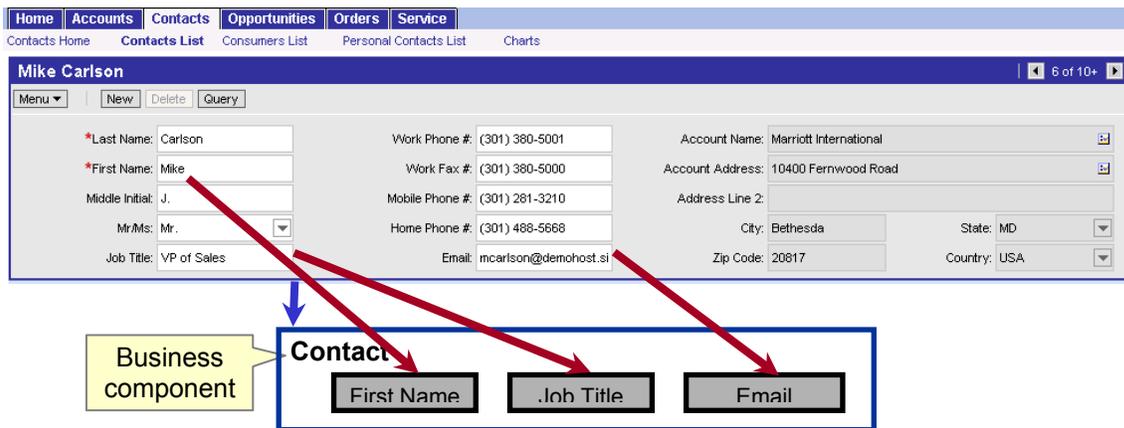


Figura 4.4.17 Relación entre Capa de Negocio y capa de Interfaz de usuario.

Oracle Siebel mediante el cliente de configuración **Siebel Tools** permite la modificación y adecuación de las tres capas anteriormente descritas para ajustarse a los requerimientos que son solicitados en la implantación del sistema.

La construcción de las tres capas se describe a continuación.

Construcción del Back - End: Capa de Base de Datos

Creación de tablas: Las tablas dentro de Siebel Tools son generadas mediante un asistente que gestiona la creación de la tabla para que posteriormente de forma manual se completen las columnas que son requeridas.

Creación de Tabla en Oracle Siebel.

La creación de una Tabla se realiza con Siebel Tools mediante un asistente de la siguiente manera, vea figura 4.4.18.

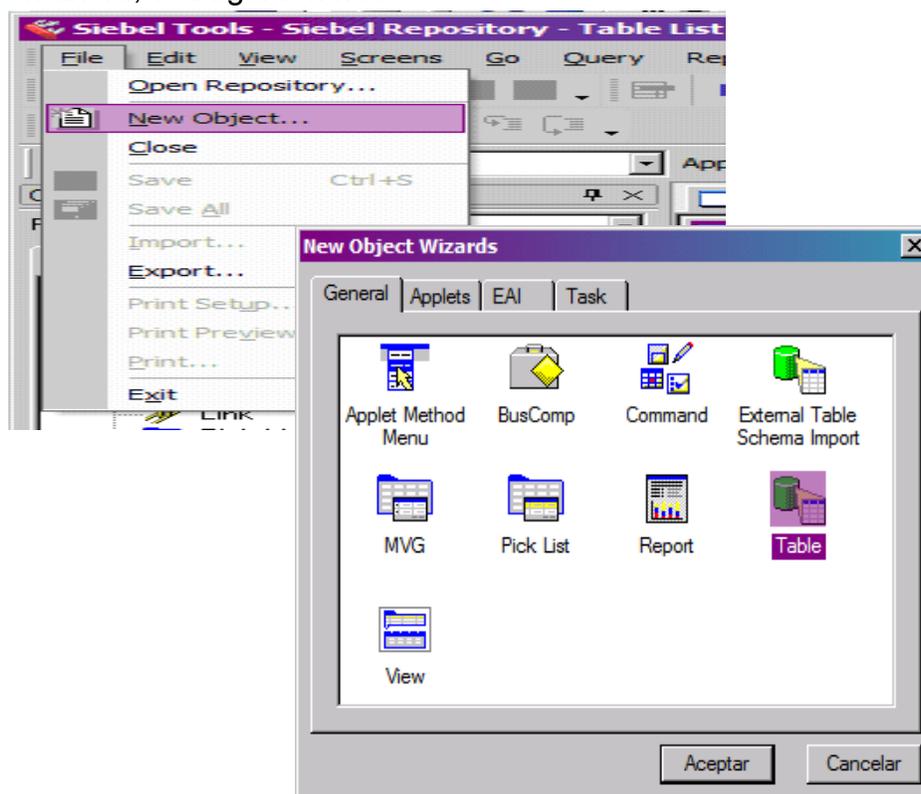


Figura 4.4.18 Creación de tablas en Oracle Siebel

Seleccionando Table de los objetos disponibles:

A continuación seleccionamos el tipo de tabla que vamos a utilizar, esto es si se trata de una tabla con cardinalidad 1:M, M:M (tabla de intersección), 1:1 o tabla sin relación.

Seleccionamos a continuación la o las tablas con las que se relacionará, ver figura 4.4.19

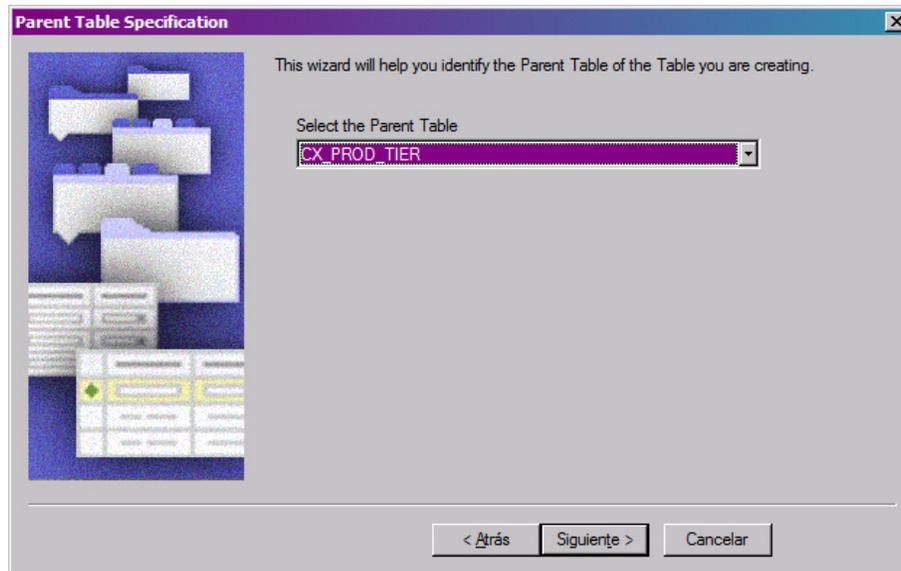


Figura 4.4.19 Seleccionando tipo de tabla

Siebel Mostrará un resumen con los cambios a aplicar:

La tabla se creará dentro del repositorio de metadatos tal como se muestra en la figura 4.4.20

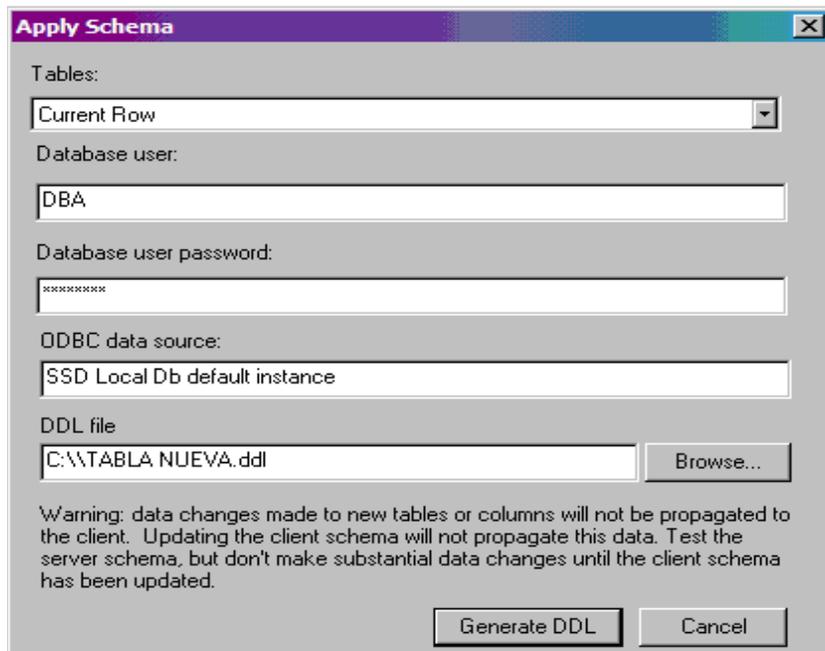


Figura 4.4.20 Repositorio de metadatos

Agregamos atributos a la tabla de ser necesario y seleccionamos el tipo de dato y longitud necesaria, ver figura 4.4.21

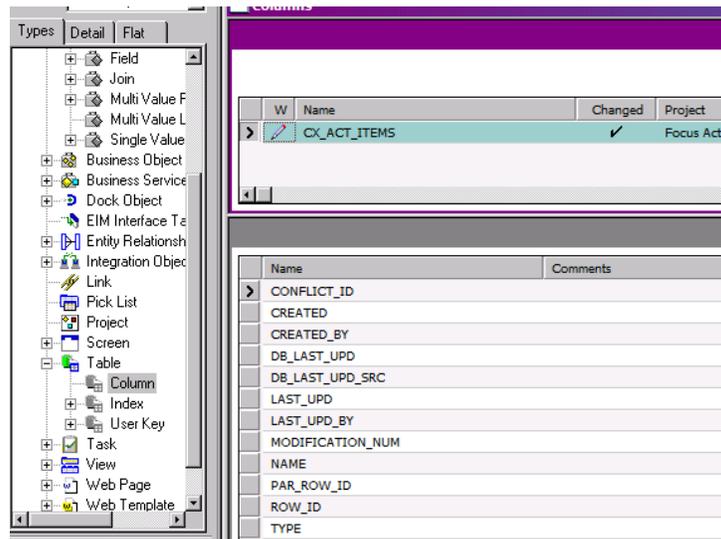


Figura 4.4.21 Atributos de la tabla

Se generará el código SQL correspondiente en el archivo DDL en la ruta que indiquemos, para ser ejecutado con SQL Plus o alguna otra herramienta con el usuario SIEBEL (propietario del esquema).

```
-- Please review product documentation in the Siebel Install Guide
-- on supportable modifications to the DDL file
create table SIEBEL.CX_ACT_ITEMS (
"ROW_ID"          varchar(45)      not null,
"CREATED"         timestamp        default current timestamp not null,
"CREATED_BY"      varchar(45)      not null,
"LAST_UPD"        timestamp        default current timestamp not null,
"LAST_UPD_BY"     varchar(45)      not null,
"MODIFICATION_NUM" numeric(10, 0)  default 0 not null,
"CONFLICT_ID"     varchar(45)      default '0' not null,
"PAR_ROW_ID"      varchar(45)      not null,
"NAME"            varchar(300)     not null,
"TYPE"            varchar(90)      not null,
"ATRIBUTO3"       numeric(10, 0),
"DB_LAST_UPD"     timestamp,
"ATRIBUTO1"       varchar(90),
"ATRIBUTO2"       varchar(90),
"DB_LAST_UPD_SRC" varchar(150))
;
commit
;
grant select,insert,update,delete on SIEBEL.CX_ACT_ITEMS to SSE_ROLE
;
commit
;
create unique index CX_ACT_ITEMS_U1 on SIEBEL.CX_ACT_ITEMS
(PAR_ROW_ID, TYPE, NAME, CONFLICT_ID)
;
commit
;
create index CX_ACT_ITEMS_M1 on SIEBEL.CX_ACT_ITEMS
(TYPE, NAME)
;
commit
;
create unique index CX_ACT_ITEMS_P1 on SIEBEL.CX_ACT_ITEMS
(ROW_ID)
;
commit
;

-- Use the [Activate] button in Tools or the SQL provided to increment
-- the schema version when the update is ready for production.
-- UPDATE SIEBEL.S_APP_VER SET
-- DB_LAST_UPD = current utc timestamp,
-- LAST_UPD = current utc timestamp,
-- CUSTOM_SCHEMA_VER = 84;
```

Árbol de Tablas: Un ejemplo del árbol de tablas se observa en la figura 4.4.22 con sus respectivas columnas mostrando tipos de dato y longitudes. (Se muestran solo algunas tablas que son representativas del proyecto).

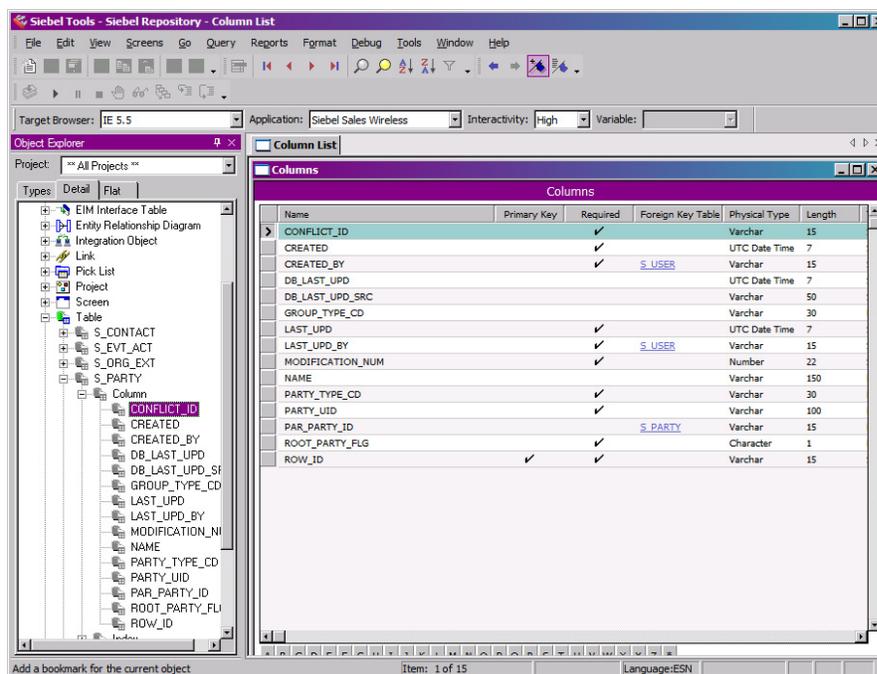


Figura 4.4.22 Árbol de tablas y columnas

Construcción del Back – End: Capa de Negocio

La construcción de los objetos de Capa de negocio se puede realizar mediante un asistente o de forma manual.

Los objetos de capa de negocio principales son Business Objects, Business Components y Links.

- **Business Object:** es una colección de Business Components que permite su agrupación.
- **Business Component:** Representa una entidad del negocio.
- **Link:** Relaciona de directamente diferentes Business Components permitiendo establecer jerarquías entre ellos dentro de un Business Object

Árbol de consultas: Las consultas son generadas a través de instancias de Business Component, los cuales mantienen tanto campos que apuntan a

columnas de tablas como campos calculados, que permiten realizar validaciones y operaciones Figura 4.4.23

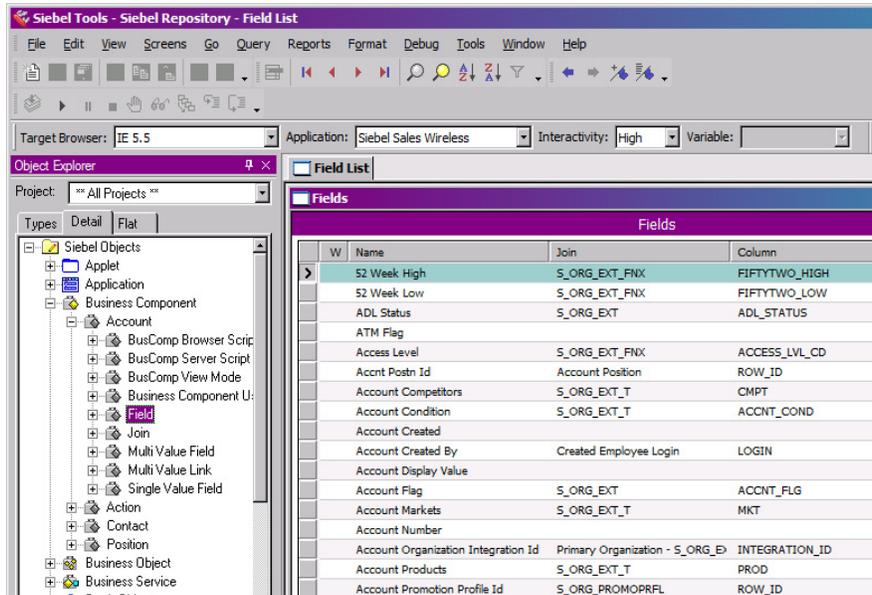


Figura 4.4.23 Árbol de consulta

Los campos o Fields del Business Component, invocan columnas de diversas tablas mediante Joins; estos contienen la información de cómo se construye, indicando un campo fuente y una columna destino tal como se muestra en la siguiente Figura 4.4.24

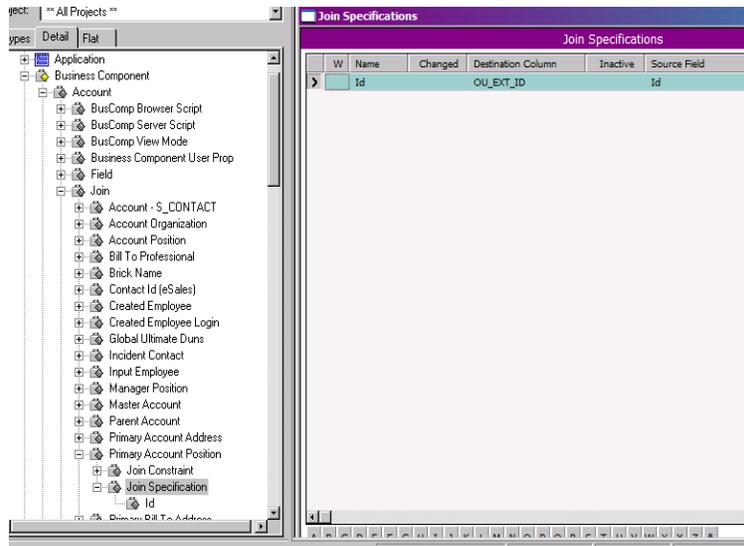


Figura 4.4.24 Definición de Joins entre tablas de un Business Component

Construcción de Business Component:

En la figura 4.4.25 se muestra la construcción de Business Component de forma manual.

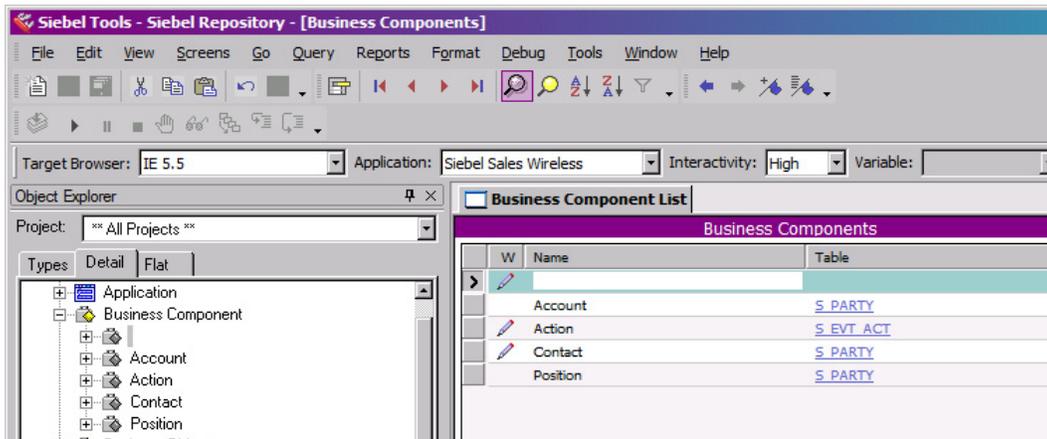


Figura 4.4.25 Creación manual de un Business Component

Se teclea el nombre y la tabla principal sobre el cual se ejecutarán las consultas.

A continuación se agregan los campos requeridos asociando la columna correspondiente así como los Joins necesarios. Ver Figura 4.4.26

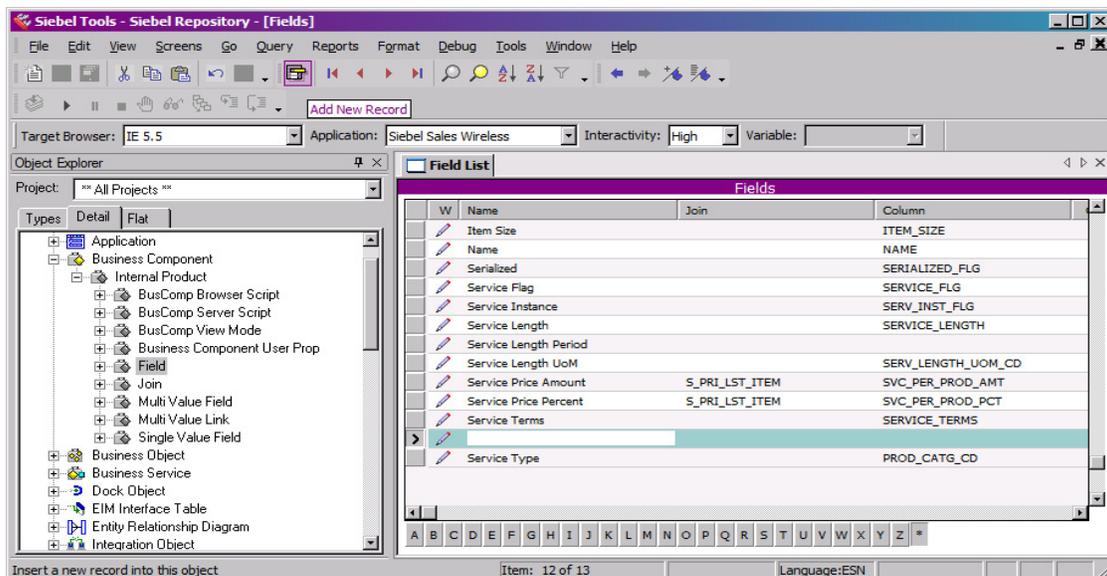


Figura 4.4.26 Creación de Campos para un Business Component

Construcción de Link:

La construcción de un link se puede generar manualmente o mediante un asistente. La forma manual se describe a continuación.

Para el tipo de objeto Link, se establecen que Business Components se relacionarán y mediante que campos. General mente representa una relación 1:M pero en ocasiones puede representar una relación M:M. Para tal caso se debe indicar cuál es la tabla de intersección y las llaves foráneas que se utilizan.

Un ejemplo figura 4.4.27 el Business Component Padre es Action mientras que el hijo es Internal Product.

Link List					
Links					
Name	Parent Business Component	Child Business Component	Inter Table	Inter Parent Column	Inter Child Column
Action/Internal Product	Action	Internal Product	S ACT PRDINT	ACTIVITY_ID	PRDINT_ID

Figura 4.4.27 Construcción de Link con relación M:M

Construcción Business Object:

La construcción de un Business Object involucra Links y Business Components (deben de existir previamente) ambos se asignan de manera conjunta y con ello se indicará la jerarquía de las entidades. En el ejemplo mostrado en la Figura 4.4.28 se muestra el Business Object **Action** que contiene los Business Components **Action**, **Contact** e **Internal Product**. Los dos últimos son hijos de Action debido a la definición del Link.

Business Object Component List				
Business Objects				
W	Name	Changed	Project	Primary Business Component
	Action		Focus Activities NV	Action

Business Object Components				
W	Bus Comp	Changed	Inactive	Link
	Action			
	Contact			Action/Contact
	Internal Product			Action/Internal Product

Figura 4.4.28 Construcción de un Business Object

4.4.1 Diseño y construcción del Front-end

Construcción de Front End: Capa de Interfaz de Usuario:

La capa de interface consiste en diferentes elementos que se relacionan con la Capa de Negocio:

- **Applet:** Relacionados directamente con los Business Components. Los controles de los Applets están relacionados con los campos de los Business Components.
- **Vista:** Es conjunto de Applets ya sean de lista o de forma. Está relacionado con un Business Object.
- **Pantalla:** Es un conjunto de vistas que permite agruparlas de acuerdo a un grupo de entidades relacionadas.

Construcción de Applets:

Se genera mediante un asistente que permite elegir el tipo de Applet ya sea de Lista, Formulario, Multivalor, Selección o Árbol. Ver figura 4.4.1.1

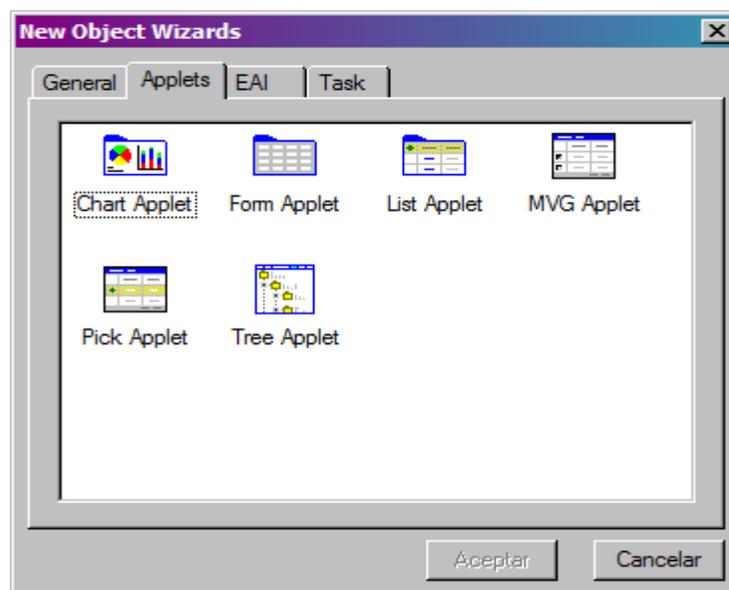


Figura 4.4.1.1 Tipos de Applet para el asistente de creación.

Luego de seleccionar el tipo de Applet, se eligen:

- Nombre único del Applet
- Business Component asociado
- Nombre que se desplegará dentro de la aplicación

A continuación se eligen que campos del Business Object se incluirán dentro del Applet, esto es que campos se desean visualizar. Ver figura 4.4.1.2

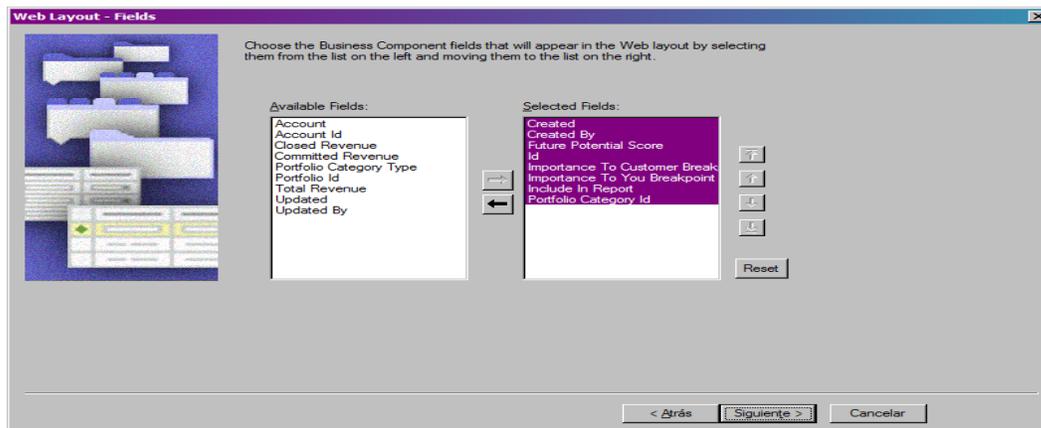


Figura 4.4.1.2 Elecciones de Campos a mostrar dentro del Applet

De igual manera que el paso anterior, se eligen los controles que permitirán realizar las opciones de creación, eliminación, consultas y navegación. Finalmente se procede a editar el layout del Applet esto es como organizar los campos que serán mostrados. Ver Figura 4.4.1.3

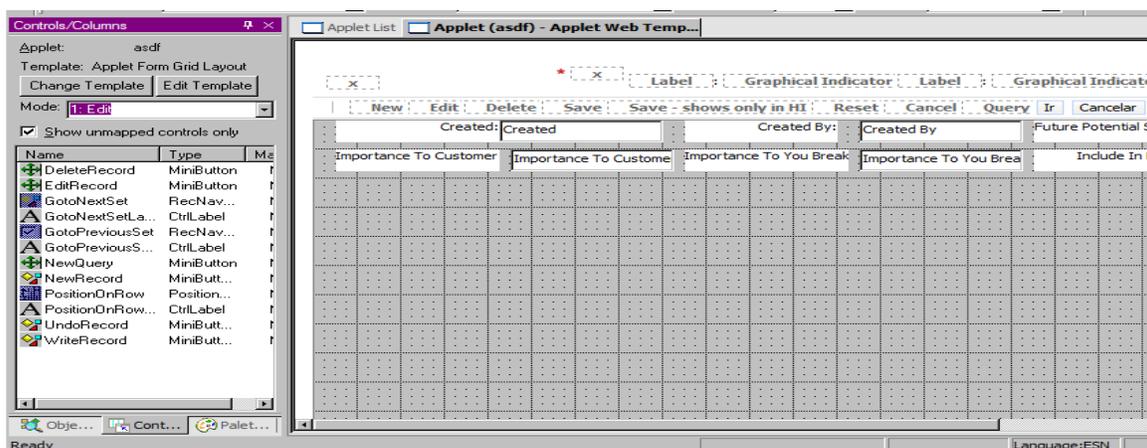


Figura 4.4.1.3 Mapeo de controles en layout de Applet.

Construcción de Vistas:

La construcción de la vista involucra la existencia previa de Applet pues estos serán el contenido. Se puede generar manual o mediante asistente. En el segundo caso se crea como un objeto nuevo y se selecciona Vista. Ver Figura 4.4.1.4

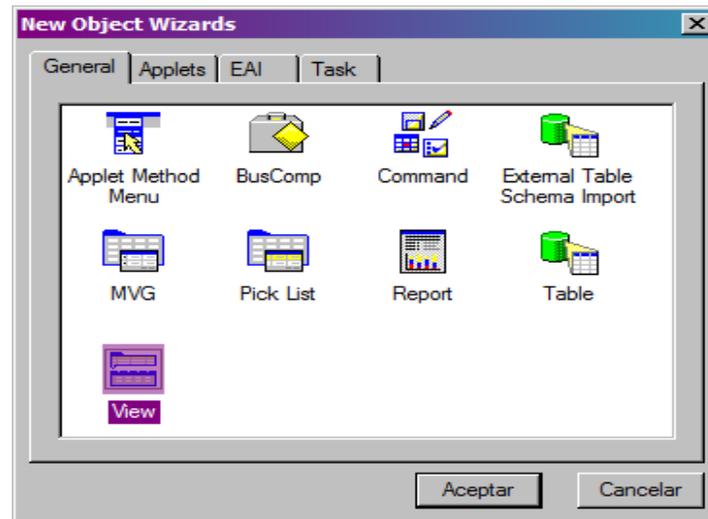


Figura 4.4.1.4 Selección de Vista en el asistente de creación.

El paso siguiente es seleccionar:

- Nombre único de vista
- Business Object en el cual está basado
- Nombre que presentará dentro de la aplicación

A continuación se elige el layout que tendrá la vista, esto es, que aspecto y posición tendrá el o los Applets que se incluirán dentro de la vista.

Finalmente se procede a completar el layout con los Applets que se requieren. Cabe aclarar que los Applets que se pueden agregar están basados en Business Components contenidos en el Business Object elegido para la Vista, ver figura 4.4.1.5

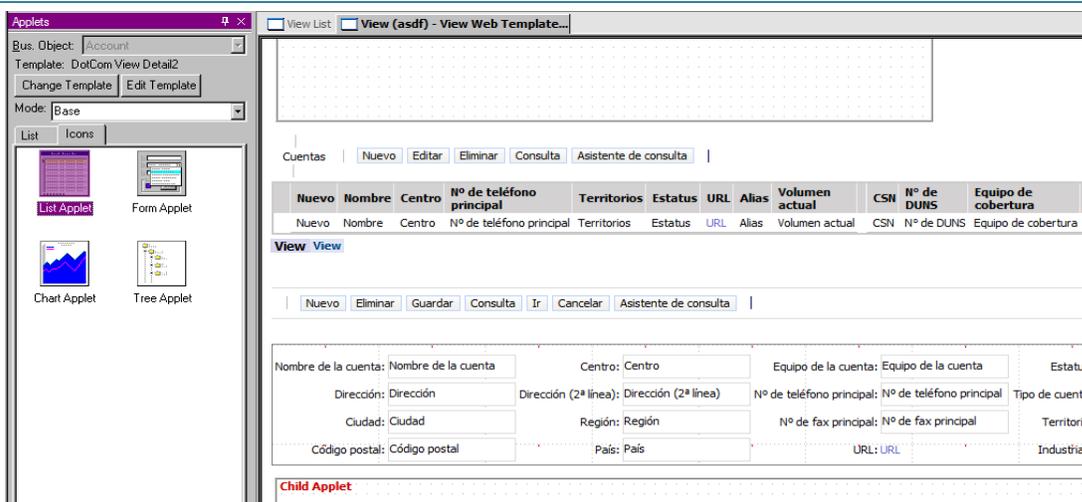


Figura 4.4.1.5 Mapeo de Applets en layout de Vista

Construcción de Pantallas:

La construcción de pantallas se realiza solo manualmente. Estas agruparán Vistas. En este caso es necesario crear el registro correspondiente y agregar las vistas requeridas. En la Pantalla es necesario indicar diferentes tipos de niveles que es la forma en cómo se agruparán las vistas dentro de la aplicación. Los niveles son los siguientes:

- Aggregate Category: Se utiliza como identificador para agrupar Vistas Padres
- Aggregate View: Son las vistas que contienen la Aggregate Category
- Detail Category: Identifica grupo de vistas hijas.
- Detail View: Son las vistas contenido de la Detail Category

Un ejemplo de esta configuración se aprecia en la figura 4.4.1.6

W	Name	Changed	Project	Viewbar Text	Viewbar Text - String Reference	Viewbar Text - String Override
	Account Screen (Show Viewbar Text)		Account (SSE)			
	Account Targeting Screen		CS CG Account Target	Cuentas objetivo	SBL_TARGET_ACCOUNTS-100909053	
	Accounts Detail Screen (SCW)		Account (SCW)	Detalles de la cuenta	SBL_ACCOUNT_DETAIL-1004224417	
	Accounts Screen	✓	Focus Accounts V	Cuentas	SBL_ACCOUNTS-1004224419-01T	

W	Name	Category Name	View	Type	Changed	Sequence
	Account Detail - Revenue Schedule View DC		Account Detail - Revenue Schedule View DC	Detail View		221
	Account Detail - Tasks View		Account Detail - Tasks View	Detail View		119
	Account Detail View		Account Detail View	Detail View		18
	Account Duplicates Detail View		Account Duplicates Detail View	Detail View		308
	Account Entitlement List View		Account Entitlement List View	Detail View		125
	Account Explorer View		Account Explorer View	Aggregate View		9
	Account Hierarchy	Account Hierarchy		Detail Category		295
	Account Iguales View		Account Iguales View	Aggregate View		400
	Account Issue View		Account Issue View	Detail View		159
	Account List	Account List		Aggregate Category		292
	Account List View		Account List View	Aggregate View		4

Figura 4.4.1.6 Mapeo de Vistas en Pantalla

El orden de las vistas contenidas dentro la Pantalla se puede editar tal como se muestra en la siguiente figura 4.4.1.7. Esto se logra haciendo click derecho sobre el nombre de la Pantalla requerida y seleccionando la opción de **“Edit Screen View sequence”**

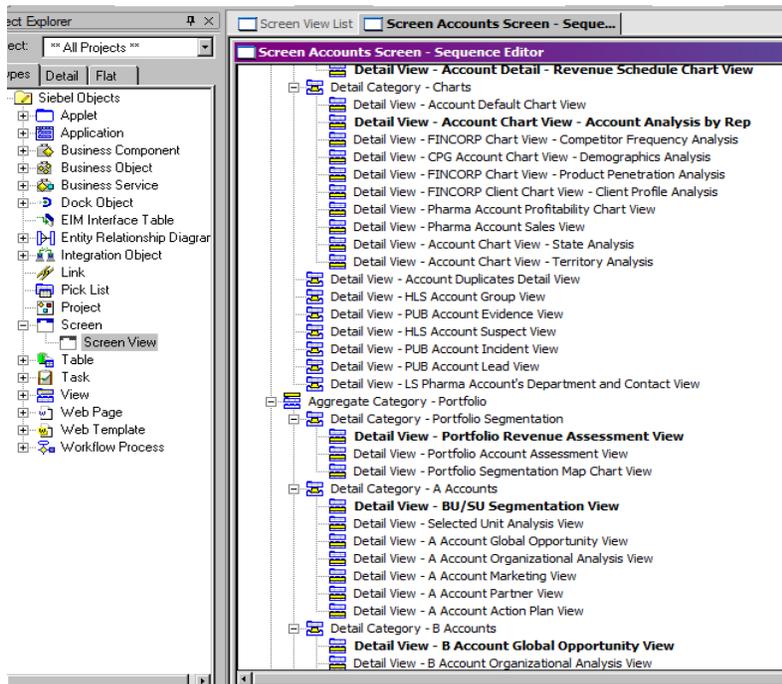


Figura 4.4.1.7 Edición de secuencia de Vistas en Pantalla.

Ejemplo de Front End Completo. Ver Figura 4.4.1.8

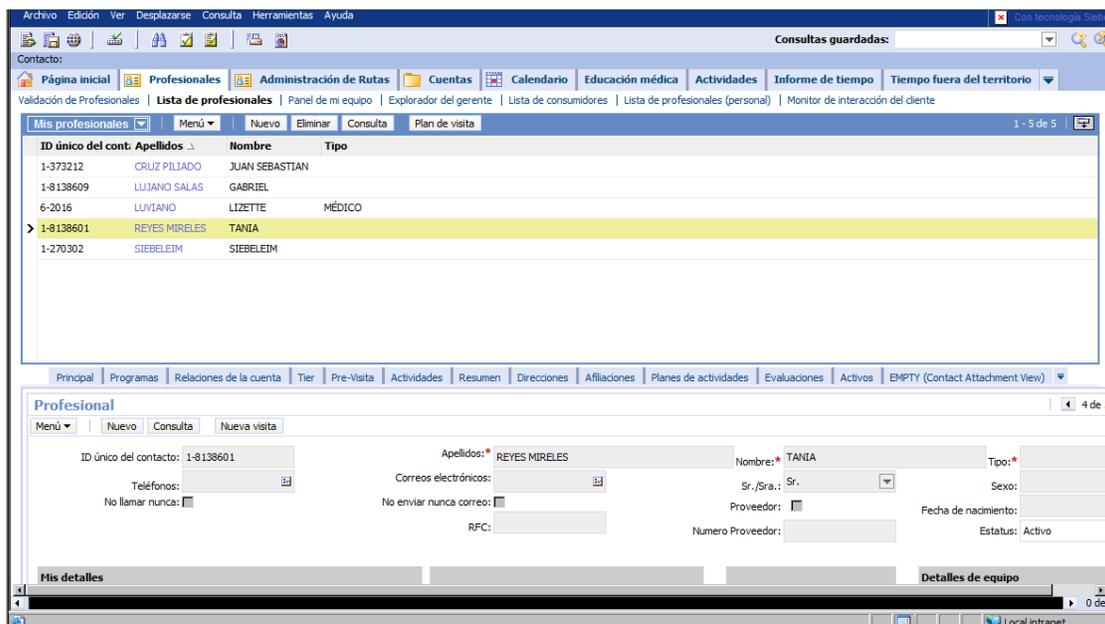


Figura 4.4.1.8 Ejemplo de Front End Siebel.

4.5 Integración y Pruebas del Sistema.

Como preámbulo en esta sección de pruebas del sistema, y sin caer en un glosario técnico, se dan aquí algunas definiciones con la intención de presentar la teoría de pruebas y aseguramiento de calidad, así como también para homologar los conceptos necesarios utilizados durante esta fase de pruebas del sistema.

Defecto: Es una deficiencia que resulta del proceso de desarrollo del sistema.

Error: Es una deficiencia que se encuentra durante el uso del sistema en producción.

Pruebas Unitarias: Se utilizan para validar una pequeña parte del código. Usualmente prueban un método particular en un contexto específico.

Pruebas Funcionales: Son aquellas pruebas que permiten validar el adecuado funcionamiento del aplicativo de acuerdo a los requerimientos definidos por el cliente.

Pruebas de Regresión: Son aquellas pruebas que permiten validar las correcciones realizadas a las fallas detectadas en las iteraciones.

Pruebas No Funcionales: Son aquellas pruebas que someten al aplicativo a condiciones extremas de rendimiento.

Pruebas de Carga: Comportamiento de una aplicación ante una carga determinada. Esta carga puede ser el número de usuarios esperado ejecutando o un número de transacciones durante un tiempo determinado.

Pruebas de Estrés: Son utilizadas normalmente para someter a la aplicación al límite de su funcionamiento, mediante la ejecución de un número de usuarios muy superior al esperado.

Pruebas de resistencia: Determinar si la aplicación puede mantener la carga esperada de manera continua y durante un largo tiempo.

Pruebas de Picos: Se realizan insertando la carga en el sistema en forma de “picos” que se irán lanzando en distintos momentos de la prueba y que permitirán comprender el comportamiento de la aplicación ante cambios bruscos de carga.

Pruebas de caja blanca: (White-Box Testing). Son pruebas estructurales. Conociendo el código y siguiendo su estructura lógica, se pueden diseñar pruebas destinadas a comprobar que el código hace correctamente lo que el diseño de bajo nivel indica y otras que demuestren que no se comporta adecuadamente ante determinadas situaciones.

Pruebas de caja negra: (Black-Box Testing) son pruebas funcionales. Se parte de los requisitos funcionales, a muy alto nivel, para diseñar pruebas que se aplican sobre el sistema sin necesidad de conocer como está construido por dentro (Caja

negra). Las pruebas se aplican sobre el sistema empleando un determinado conjunto de datos de entrada y observando las salidas que se producen para determinar si la función se está desempeñando correctamente por el sistema bajo prueba.

Prioridad: Requerimiento de corrección de la falla en el software (Alta, Media, Baja)

- Alta: Error que necesita ser corregido lo antes posible debido a su relevancia.
- Media: Error que puede o no ser corregido inmediatamente.
- Baja: Error que necesita ser corregido sin premura.

Severidad: Impacto causado por la falla en el software (Alta, Media, Baja)

- Alta: Error considerado como bloqueante, es decir, que impide continuar con el flujo de la prueba.
- Media: Error que no impide que se continúe con el flujo de la prueba.
- Baja: Errores considerados como ortográficos y/o redacción.

Por cuestiones de espacio, y dado que el alcance de este trabajo de tesis no pretende agotar toda la teoría de Pruebas y aseguramiento de Calidad del software, solo se incluyen algunos ejemplos de pruebas realizadas al sistema, una de caja blanca y dos de caja negra.

Ejemplos de Prueba de Caja Negra.

Inicio de sesión en el sistema.

En la figura 4.5.1 se muestra la pantalla de acceso al sistema:

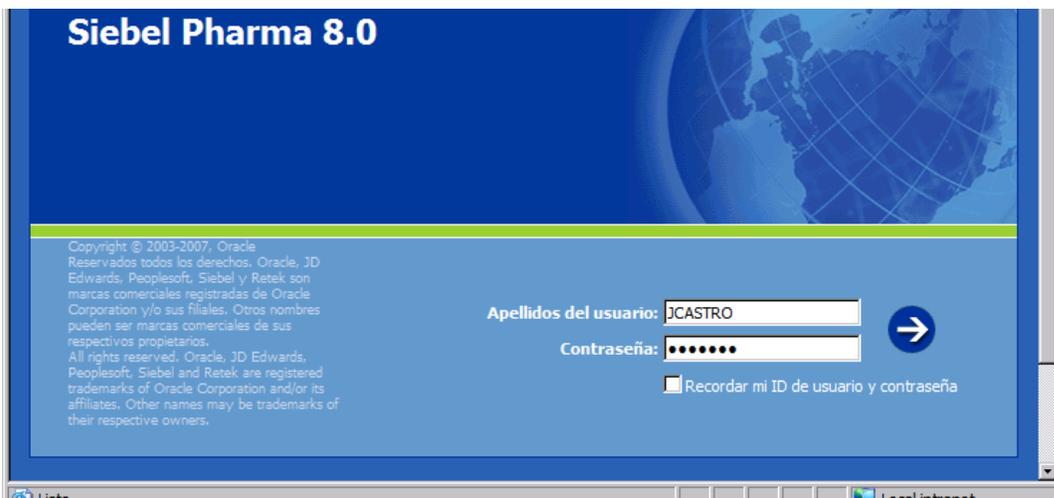


Figura 4.5.1 Pantalla de inicio (login) del sistema.

En orden de continuar con la prueba de caja negra sobre esta parte del sistema, si se introducen valores incorrectos para el usuario y/o password, el sistema le niega el acceso al usuario y vacía los campos del formulario (cajas de texto) indicando además con un mensaje el error del que se trata, Véase Figura 4.5.2.

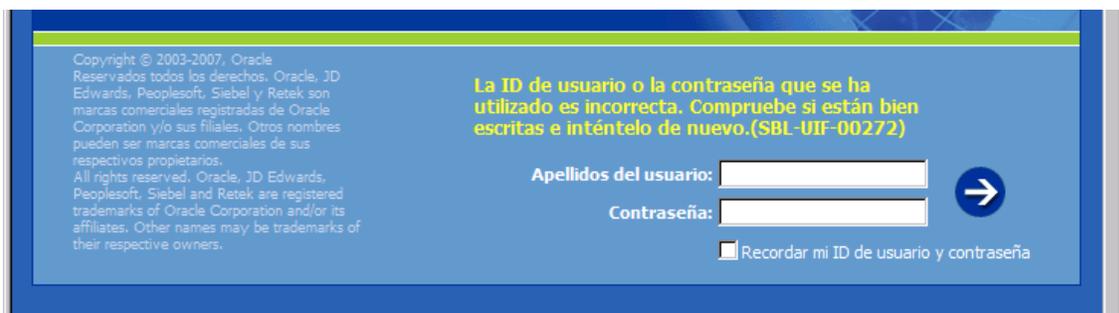


Figura. 4.5.2 Ejemplo de acceso denegado debido a un incorrecto Usuario y/o password.

La otra alternativa en esta misma prueba, es cuando el usuario y el password proporcionados en el formulario son correctos, la figura 4.5.3 muestra el nuevo intento de acceder al sistema partiendo del error mostrado en la figura 4.5.2:

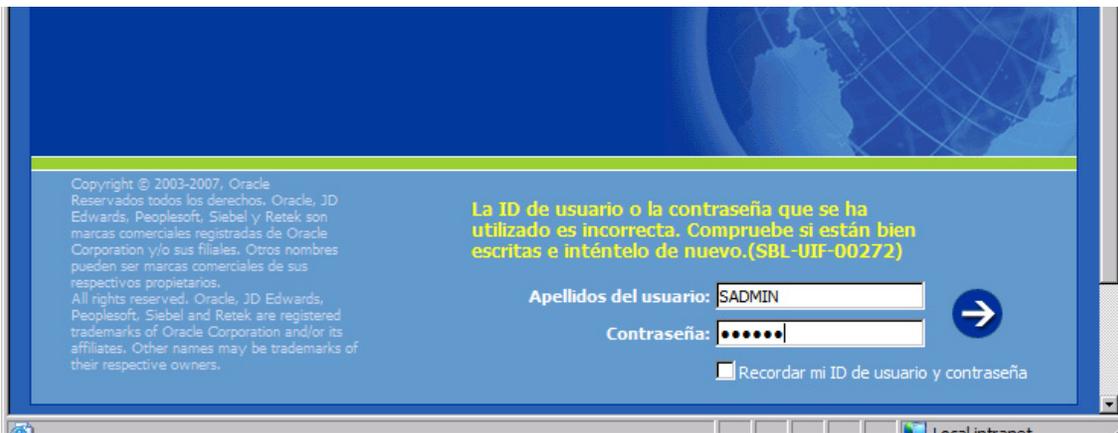


Figura 4.5.3 Reintento de login a partir de un error en la autenticación.

La figura 4.5.4 muestra la pantalla de inicio (principal) del sistema, una vez que se podido iniciar una sesión en el mismo partiendo de un usuario y un password válido.

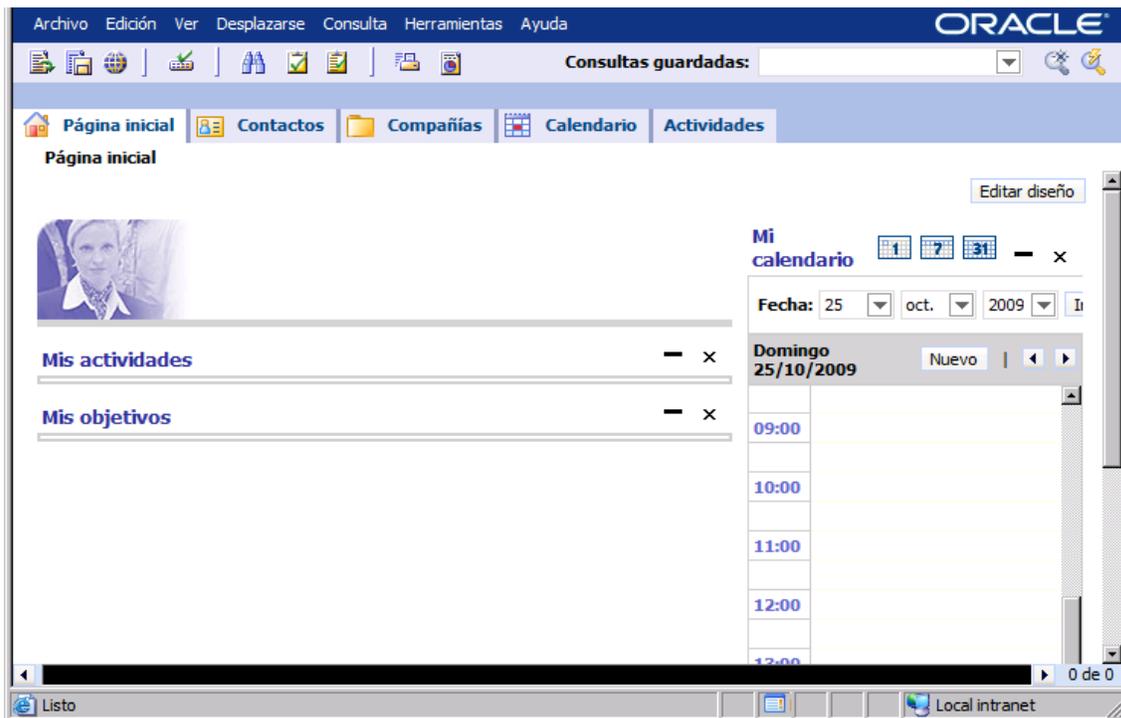


Figura 4.5.4 Pantalla principal del sistema.

Inserción en BD.

En este ejemplo, demostraremos como se agrega un nuevo registro a una categoría de las entidades del sistema, específicamente agregaremos datos de una Cuenta. La figura 4.5.5 muestra el listado (vacío) para la tabla que almacena la información de las entidades “Cuenta”. En esta pantalla se pueden apreciar los diferentes campos que son requeridos para la inserción del nuevo registro en la entidad, que son los campos marcados con un asterisco rojo en la parte inferior de la pantalla.

Para proseguir con la prueba, hacemos clic en el botón “Nuevo” señalado en la figura 4.5.5, y proporcionamos los datos que se nos piden.

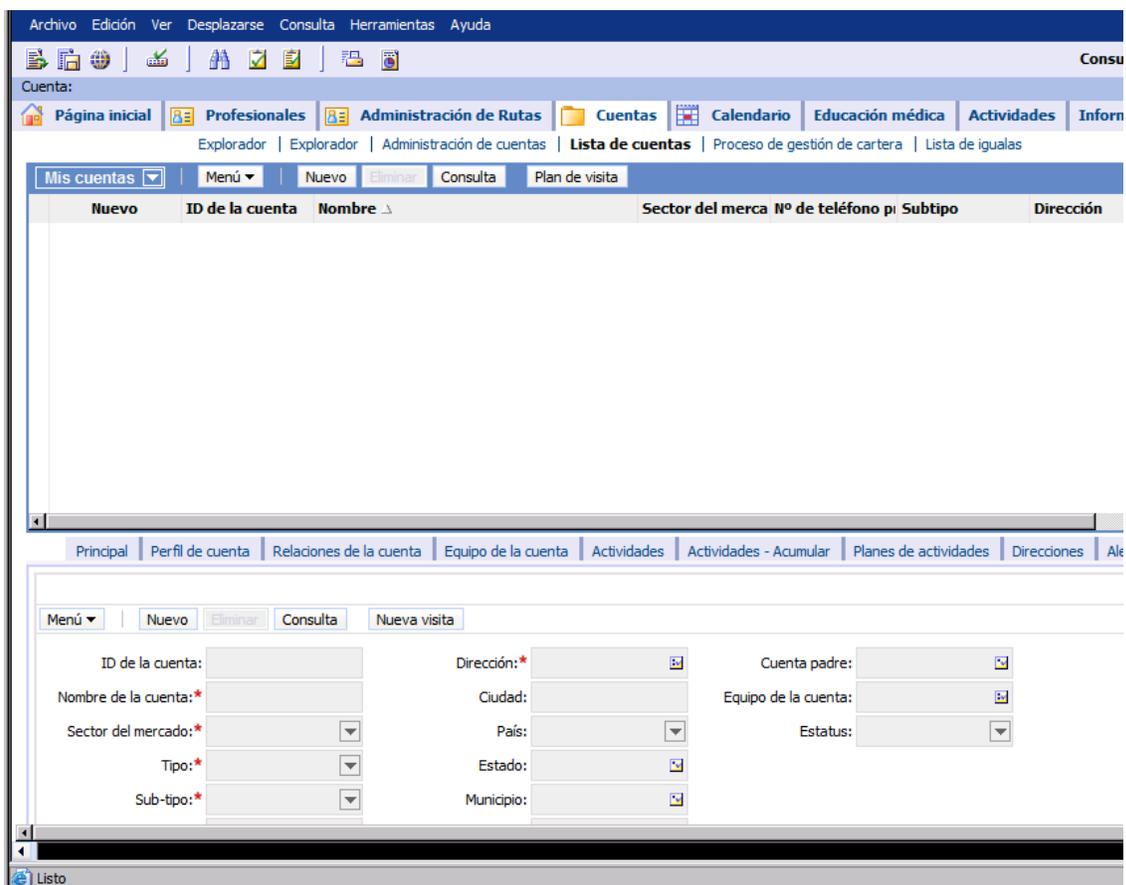


Figura 4.5.5 Lista de entidades Cuenta.

La figura 4.5.6 muestra el resultado de haber introducido los valores necesarios en los campos del formulario, y se aprecia el registro recién ingresado como parte del listado de registros para la entidad Cuenta.

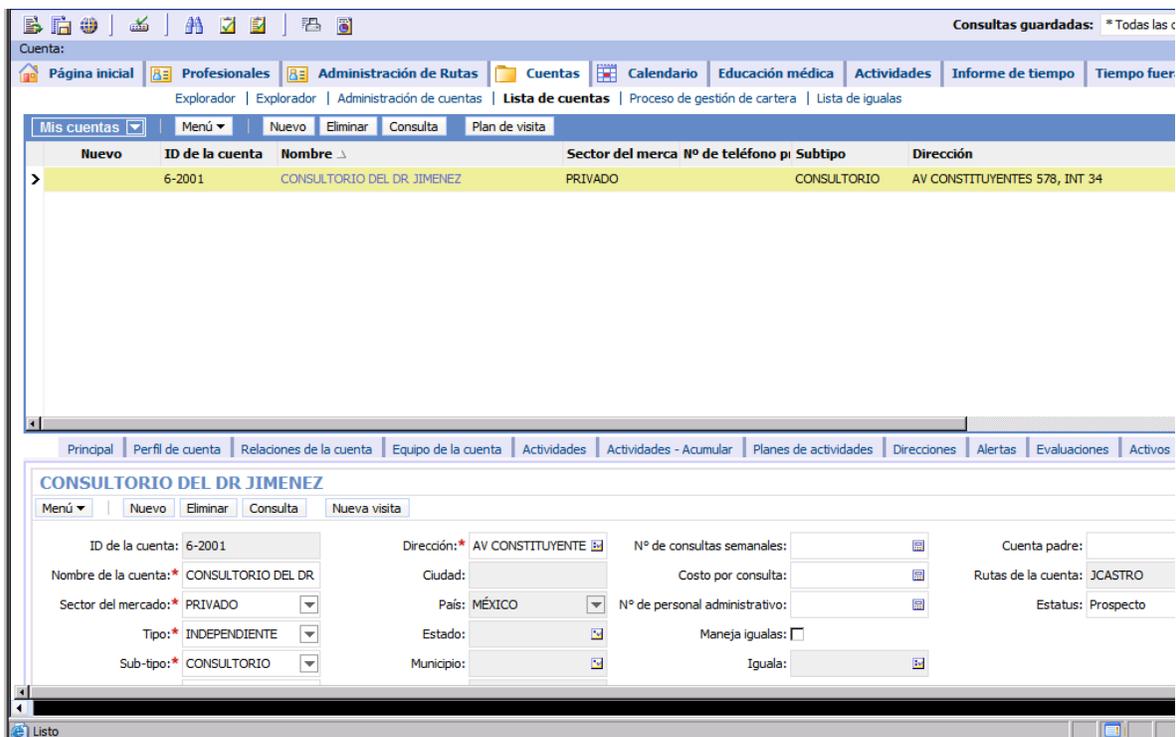


Figura 4.5.6 Nuevo registro agregado en BD y consultado por el sistema.

Prueba de Caja Blanca.

Validación de un campo requerido.

A continuación con fines ilustrativos, en la pantalla 4.5.7. se muestra un **snippet (término en programación que se refiere a una pequeña porción de código fuente)** para una validación “interna” de la lógica de negocio del sistema, particularmente en el caso de que se quiera dar de alta un profesional (Médico) en la lista de profesionales, con una especialidad sin proporcionar el número de cedula profesional. Es decir, se trata de hacer una inserción sin proporcionar un campo obligatorio por las reglas de negocio (y/o de integridad de datos en el esquema de BD).

```
function f_ced_prof()
{
try{
this.ActivateField("Id");
var sContactId = this.GetFieldValue("Id");
var sIsCedNull = "";
var sSpec = "";
var m_busObject = TheApplication().GetBusObject("Nov Cedula Profesionales");
var BCCed = m_busObject.GetBusComp("Nov Cedula Profesionales");

BCCed.ActivateField ("Nov Ced Prof");
BCCed.ActivateField ("Contact Id");
BCCed.ActivateField ("Specialty Id");
BCCed.ActivateField ("Specialty Name");

BCCed.ClearToQuery ();
BCCed.SetSearchSpec("Contact Id", sContactId);
BCCed.ExecuteQuery ();
var isFirst = BCCed.FirstRecord()
if (isFirst)
{
while(isFirst)
{
sIsCedNull = BCCed.GetFieldValue("Nov Ced Prof");
sSpec = BCCed.GetFieldValue ("Specialty Name");
if (sIsCedNull == null || sIsCedNull == "")
{
TheApplication().RaiseErrorText("La especialidad "+sSpec+" no tiene cedula profesional");
}
else
{
isFirst = BCCed.NextRecord();
}
}
}
else
{
TheApplication().RaiseErrorText("El Contacto no tiene especialidades");
}
}
}
```

Figura 4.5.7 Código que realiza una validación sobre un campo requerido.

En la pantalla 4.5.8 en orden de continuar con la prueba el sistema valida (al guardar) el registro y detecta que tiene especialidad sin cedula profesional asociada.

The screenshot shows a web-based interface for managing medical contacts. The top navigation bar includes options like 'Administración de Rutas', 'Cuentas', 'Calendario', 'Educación médica', 'Actividades', 'Informe de tiempo', and 'Tiempo fuera del territorio'. Below this, there's a breadcrumb trail: 'e la cuenta > Tier > Pre-Visita > Actividades > Resumen > Direcciones > Afiliaciones > Planes de actividades > Evaluaciones > Activos > EMPTY (Contact Attachment View)'. The main form displays details for a contact named 'LIZETTE'. Fields include 'Nombre' (LIZETTE), 'Sr./Sra.' (dropdown), 'Tipo' (MÉDICO), 'Especialidades' (ALERGOLOGIA), 'Cédula profesional' (empty), 'Status del contacto' (Prospecto), 'Fecha de nacimiento', 'Proveedor', 'Numero Proveedor', 'Tendencia prescriptiva', 'Estado', 'Ciudad', 'Código postal', 'Brick', 'Sub-especialidad de la Ruta', 'Mi fecha de la última visita', 'Status Ruta Contacto', 'Solicitud activar', and 'Motivo del movimiento'. At the bottom, there's a table header with columns: 'Nombre', 'Ubicación de visit', 'Dirección', 'Ciudad', 'Código postal', 'Sub-especialidad', 'Fecha de la últim', 'Status ruta cont', and 'Puesto de trabajo'. The table currently shows 'Sin registros' and '0 de 0'.

Figura 4.5.8 Contacto Médico con especialidad sin Cédula Profesional Asociada.

En la figura 4.5.9 se puede observar el correspondiente gráfico a la validación implementada en código que se observó en la figura 4.5.7, en esta pantalla lo que se realiza es el mensaje del sistema indicando el error, y la implicación de no permitir avanzar con la inserción hasta que todas las condiciones de información del registro hayan sido satisfechas.

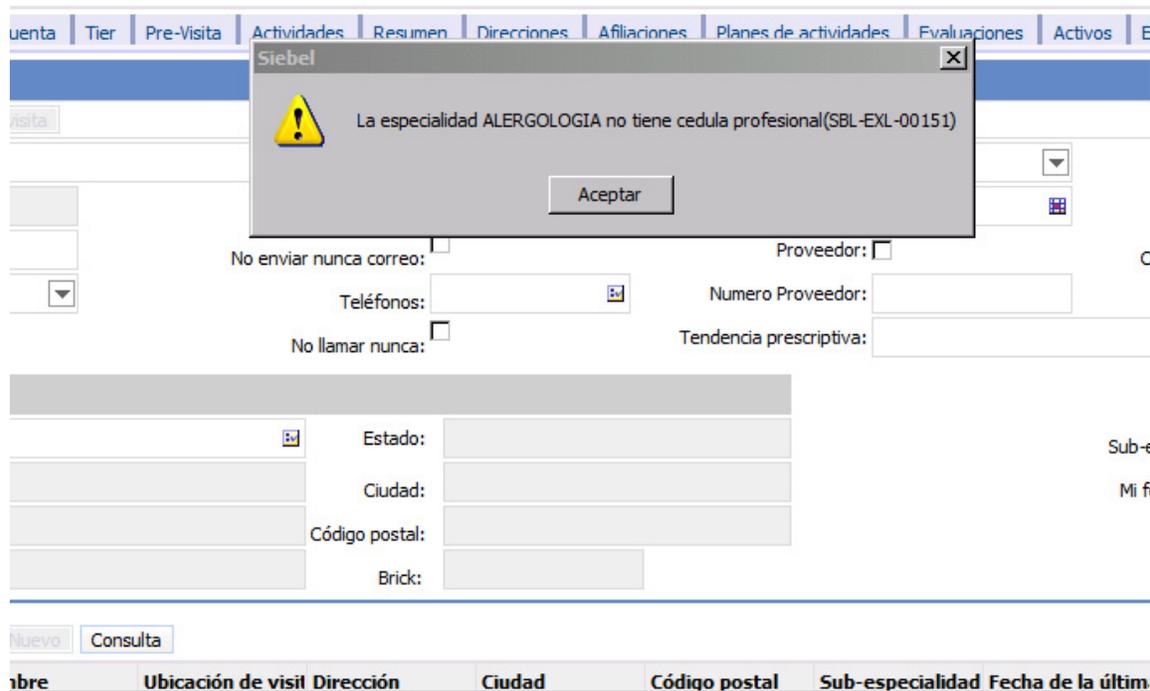


Figura 4.5.9 Mensaje de error, al realizar validación.

En este caso para poder realizar la inserción sería necesario introducir un valor en el campo de cédula profesional de la página del sistema.

4.6 Obtención de reportes

Durante el análisis de requerimientos y el trabajo de campo acerca de cómo se obtiene actualmente la información, se determinó que los registros que más información útil poseen para fines estadísticos, decisivos y de análisis son 4 rubros:

- Reporte de Frecuencia de Visitas de los últimos 6 meses.
- Reporte de Contactos por Representantes de Venta.
- Reporte de visitas a contacto y cuentas
- Reporte de Frecuencia de visitas por cuenta

En la siguiente figura 4.6.1 se muestra un ejemplo del **Contact Call Frequency, Reporte de Frecuencia de Visitas** de los últimos 6 meses. En este se plasman en columnas cada mes, indicando el número de llamadas/visitas realizadas al contacto, el nombre del contacto, y una categoría de la frecuencia de llamadas o visitas.

Contact Call Frequency			SIEBEL					
Contact Name	Rep Specialty	Call Frequency	Number of Calls Per Month					
			May	Jun.	Jul.	Aug.	Sept.	Oct.
ANDREOU, DIANE			0	0	0	0	0	0
Aamos, Ray		>5 times/Year	0	0	0	0	0	0
Aaron, Felix			0	0	0	0	0	0
Aaron, Mary			0	0	0	0	0	0
Aaron, Meredith			0	0	0	0	0	0
Aarons, Maisha			0	0	0	0	0	0
Aarons, Mary			0	0	0	0	0	0
Abadia, Gerardo			0	0	0	0	0	0
Abbot, Bobby			0	0	0	0	0	0
Abbot, Lisa			0	0	0	0	0	0
Abbott, Diana			0	0	0	0	0	0
Abby, George			0	0	0	0	0	0
Abel, Henry			0	0	0	0	0	0
Able, Carol			0	0	0	0	0	0
Able, Carol			0	0	0	0	0	0
Able, Janet			0	0	0	0	0	0
Abrams, Joan			0	0	0	0	0	0
Abrams, Michael			0	0	0	0	0	0
Abrams, Peter			0	0	0	0	0	0
Accinni, Kathy			0	0	0	0	0	0
Ackerly, Brian			0	0	0	0	0	0
Ackerly, Sandra			0	0	0	0	0	0
Ackerly, To			0	0	0	0	0	0

4.6.1 Reporte de Frecuencia de Visitas de los últimos 6 meses.

El reporte **Contacts – by account** o **Reporte de Contactos por Cuenta (Representantes de Venta)**, muestra campos como apellidos, nombre, teléfono, fax, dirección etc. Es decir, de todos los datos relevantes para el contacto. Véase figura 4.6.2.

Contacts - by account											SIEBEL
Last Name	First Name	M/M	Work Phone	Work Fax	Position	Account	Address	City	State	Zip	
ANDREOU	DIANE	Mr.	(972) 555-1234				100 ANDREOU Street	Dallas	PA	75277	
Abadia	Gerardo										
AbeI	Henry	Mr.									
Able	Carol				ContactCenter Agent						
Abrams	Joan	Mrs.			Housewife		99 Hightower Lane	San Leandro	CA	94651	
Abrams	Michael	Mr.			Student		99 Hightower Lane	San Leandro	CA	94651	
Ackerly	Brian	Mr.			Student						
Adcox	Bradley	Mr.	(415) 345-6010	(415) 345-6015	Executive Vice President						
Adams	Robert	Mr.			Research Analyst						
Adelman	Deborah		(973) 540-2001	(973) 568-2345			10 Gloucester Circle	Hartford	CT	06103	
Adler	Suzanne	Mrs.	(415) 345-8805	(800) 331-4283							
Adler	Jim	Mr.	(408) 853-7000 x201	(408) 853-7001	NE Executive VP of Marketing						
Administrador	Siebel				Sys Admin		1855 South Grant St.	San Mateo	CA	94403	
Administrador - AU	Siebel				GM Sys Admin - AU						
Administrador - CDM	Siebel		(415) 766-9076		Sys Admin - CDM						
Administrador - CS	Siebel				Sys Admin - CS						
Administrador - ENVY	Siebel				Sys Admin - ENVY						
Administrador - ER	Siebel				Sys Admin - ER						
Administrador - FIN	Siebel				Sys Admin - FIN						
Administrador - FIN	Siebel				Sys Admin - FIN						
DEU											
Administrador - GOV	Siebel				Sys Admin - GOV						
Administrador - HRZ	Siebel				Sys Admin - HRZ						
Administrador - HT	Siebel				Sys Admin - HT						
Administrador - III	Siebel				Sys Admin - III						
Administrador - INS	Siebel				Sys Admin - INS						
Administrador - LS	Siebel				Sys Admin - LFS						
Administrador - MAN	Siebel				Sys Admin - MAN						
MAN											

Report Generated for ADMIN on 10/24/00 Page 1 of 17

4.6.2 Reporte de Contactos por Representantes de Venta.

El reporte de actividades **Activity List (Reporte de visitas y/o llamadas a contacto y cuentas)** muestra el detalle de actividades y llamadas realizadas capturadas en el sistema. Véase Figura 4.6.3.

No.	Activity Type	Name	Phone No.	Due	Done	Description	Opportunity	Account	Account Site	Status
1	Account Call			8/6/1999 05:00:00 PM		Meet and visit decision maker	Jamba - \$2.5MM EXIM	Jamba-Wear Corp.	HQ	Planned
2	Professional Call	Jonah Alexander	(201) 874-2362	11/6/2002 04:00:00 PM		First visit				Planned
3	Professional Call	Jonah Alexander	(201) 874-2362	1/6/2004 04:00:00 PM		Follow up visit				Planned
4	Professional Call	Jonah Alexander	(201) 874-2362	2/6/2004 04:00:00 PM		Final call				Planned

Figura 4.6.3 Reporte de visitas a contacto y cuentas

En el reporte **Accounts Visits by Salesperson**, **Reporte de Frecuencia de visitas por cuenta (Vendedor)**, mostrado en la figura 4.6.4, se ilustra un claro ejemplo de cómo se concentran las visitas por mes, trimestre o año, por vendedor y por cuenta, así como también el lugar en el que fue realizada la visita.

Account	Site	Dist. Code	Visit Freq.	# Visits past month	# Visits past Qtr.	# Visits past Yr.
Com	HQ-Distribution			0	0	0
E.Sorenson & Associates	Charlotte			0	0	0
G Edwards & Sons, Inc	HQ-Distribution			0	0	0
G Edwards & Sons, Inc	San Francisco			0	0	0
aron-Jones Dry Cleaning	San Francisco			0	0	0
bbey General Hospital	NJ			0	0	0
bbot Designs	San Francisco			0	0	0
bbot School for Medicine	San Francisco			0	0	0
bbtisi Consolidated Inc.				0	0	0
brams Insurance Carriers	Costa Mesa			0	0	0
ccess Unlimited Financial Services	San Francisco			0	0	0
me New England Division	Boston			0	0	0
cupuncture Institute of Northern California	HQ-Distribution			0	0	0
dam Marketing Resources, Inc.	West			0	0	0
ddison Fleet Services	West			0	0	0
ddison Manufacturing	HQ-Distribution			0	0	0
ddison Real Estate	California			0	0	0
dohe Systems, Inc.	California			0	0	0
guillar	San Francisco			0	0	0
ikon AG	Salo, Finland			0	0	0
ir Canada				0	0	0
irtime Hobby Shop, Inc.	Boston			0	0	0
irtouch Communications	West			0	0	0
lameda Healthcare Systems	HQ			0	0	0
lan Baker Licensing Agency				0	0	0
lbany General Hospital	Albany			0	0	0

4.6.4 Reporte de frecuencia de visitas por cuenta