



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DESARROLLO DEL SISTEMA PARA EL CONTROL DE
PROYECTOS ANTARES

T E S I S

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN

P R E S E N T A

ANDRÉS PUGA LUGO

DIRECTOR DE TESIS:
M. C. ALEJANDRO VELÁZQUEZ MENA



CIUDAD UNIVERSITARIA 2009

Agradecimientos:

A Gisela y Gabina, mi madre y abuelita respectivamente, por darme su vida para poder vivir la mía, gracias por mostrarme el verdadero significado de la familia, el concepto de disciplina y el valor de la constancia; ustedes son las dos personas que más admiro en la vida.

Estoy muy agradecido con ustedes mis dos madres por regalarme su tiempo, su espacio, porque a lo largo de mi vida me han colmado de cariño, respeto y confianza, gracias por todos los sacrificios, esfuerzos y dificultades que, junto con mi abuelito Natividad tuvieron que sortear para proveerme de una educación y un hogar; jamás podré compensar todo el amor derrochado, su infinita paciencia y apoyo incondicional, que Dios las bendiga.

A mi gran maestro: Natividad, que desde el cielo se mantiene al alba y donde sin duda, sigue al pendiente de toda su familia. Muchas gracias abuelito por permitirme ser tu compañero, por enseñarme a ser responsable y por mostrarme que el trabajo ennoblece al hombre, gracias por compartir toda tu sabiduría con los tuyos, sin duda tu código ético es un gran legado.

A Felipe, Adriancito y Almita, gracias por ser mi familia, por todo lo que aprendo de ustedes día a día, por ser las personas más brillantes y nobles que conozco. Gracias por contagiarme de su felicidad, por soportar mi mal humor y por ayudarme a crecer, ustedes son un aliciente que me ayuda a seguir adelante.

A mis hermanos del alma, Ponchito, Alex, Pepe, Saúl, Erika, Adrian por cuidar de mí prácticamente toda la vida, por estar a mi lado en las buenas, en las malas y en las peores, pero sobre todo, gracias por ayudarme a creer en mí. En verdad les digo que ustedes han sido y serán, el mejor ejemplo a seguir y mis hermanos mayores.

A los Puga, por dejarme entrar en sus vidas, por todo el aprendizaje y cariño incondicional brindado. Mi más profundo agradecimiento a mi abuela Chuy, a mis tíos Marco, Kay, Memo y los primos Adrian, Marina, Ajax, Marquito, Mochis, a los que ya no están, Andrés, Juan y Alejandro, gracias por todas sus enseñanzas, por la calidez humana, por las risas, por los regaños y por ser unas excelentes personas, siempre estaré en deuda con ustedes.

Al tesoro más grande en la vida, mi familia: mis tíos Tino, Benigno, Nachito, Bertina, Luisa, Seras, Cunda, Celsa, Petra, a todos los primos y sobrinos, gracias por cuidar de mí, por sus sabios consejos y sus valiosos regaños, por todas las alegrías y travesuras, por los días de fiesta y los días difíciles, en los que siempre nos mantenemos unidos. Gracias por los incontables momentos de felicidad que me han regalado y por ofrecerme siempre un techo y sustento, por abrirme su casa,

pero sobre todo gracias por creer en mí. En verdad agradezco a Dios pues no pude tener una mejor familia que ustedes.

A la familia Mendoza Caballero por permitirme invadir su casa, por dejarme contagiado de su buena vibra, por las incontables charlas de sobremesa, por compartir tantas tazas de café conmigo mientras discutimos de política y fútbol, por las risas, por los de manita y por permitirme formar parte del club de cinéfilos de la sala.

Y a ti mi Lyndita, gracias por todos los momentos felices, por todos tus consejos y enseñanzas, por hacerme absolutamente feliz, por las desveladas, por cuidar de mí, por abrirme tu corazón, por caminar a mi lado y por ser la persona más maravillosa del mundo.

A Dios, le agradezco infinitamente que hoy me permita ver culminada mi carrera profesional y a todos ustedes, gracias por tocar mi vida y marcarla para siempre.

Sinceramente: Andrés Puga Lugo.

ÍNDICE

ÍNDICE.....	4
INTRODUCCIÓN.....	7
CAPÍTULO I: CONCEPTOS GENERALES.....	9
1.1 LAS APLICACIONES WEB.....	9
1.2 PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE.....	10
1.2.1 NECESIDAD DE UTILIZAR UNA METODOLOGÍA.....	10
1.2.2 PROCESO DE DESARROLLO DE SOFTWARE.....	10
1.2.3 APLICACIÓN DEL RUP A ANTARES.....	11
1.2.4 DEFINICIÓN DE UN CASO DE USO.....	11
1.2.5 LA IMPORTANCIA DE UNA CORRECTA ARQUITECTURA.....	12
1.2.6 PROCESO DE DESARROLLO ITERATIVO E INCREMENTAL.....	13
1.2.7 LA VIDA DEL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE (RUP).....	15
1.2.8 EL PRODUCTO ANTARES.....	16
1.2.9 FASES DEL CICLO.....	17
1.3 ESTÁNDAR J2EE.....	20
1.3.1 VISIÓN GENERAL DE J2EE.....	20
1.3.2 ARQUITECTURA J2EE.....	21
1.3.3 APLICACIÓN DE J2EE A ANTARES.....	23
1.4 PATRONES DE DISEÑO.....	25
1.4.1 IMPORTANCIA DEL USO DE PATRONES.....	25
1.4.2 PATRONES DE CREACIÓN.....	26
1.4.3 PATRONES DE COMPORTAMIENTO.....	28
1.4.4 PATRONES ESTRUCTURALES.....	30
1.4.5 PATRONES DE SISTEMA.....	30
1.5 MARCOS DE TRABAJO.....	32
1.5.2 FRAMEWORK STRUTS.....	33
1.5.3 FRAMEWORK SPRING.....	37
1.5.4 FRAMEWORK HIBERNATE.....	39
CAPÍTULO II: FASE DE INICIO.....	43
2.1 REQUISITOS FUNCIONALES DE ANTARES.....	43
2.1.1 PROBLEMÁTICA A RESOLVER.....	43
2.1.2 REQUISITOS PARA ANTARES.....	44
2.1.3 MODELO DE DOMINIO.....	46
2.2 ACUERDOS DE ADMINISTRACIÓN DEL PROYECTO.....	48
2.2.1 INTROUCCIÓN.....	48
2.2.2 DESCRIPCIÓN DEL PROYECTO.....	48
2.2.3 ORGANIZACIÓN Y COMUNICACIÓN.....	50
2.2.4 RESPONSABILIDADES DE LOS GRUPOS DE TRABAJO.....	52
2.2.5 PROCEDIMIENTOS DE ADMINISTRACIÓN DEL PROYECTO ANTARES.....	58
2.3 METODOLOGÍA DE DESARROLLO.....	61
2.3.1 PROCESOS.....	61
2.3.2 REVISIONES.....	62
2.3.4 HERRAMIENTAS DE MODELADO Y DESARROLLO.....	62
CAPÍTULO III: FASE DE ELABORACIÓN.....	65
3.1 MODELO DE CASOS DE USO.....	65
3.1.1 ACTORES.....	65
3.1.2 DIAGRAMA DE CASOS DE USO DE ANTARES.....	67

ÍNDICE

3.2 ARQUITECTURA DE ANTARES.....	68
3.2.1 ALCANCE	68
3.2.2 MECANISMOS Y RESTRICCIONES	68
3.3 ANÁLISIS DE ANTARES	76
3.3.1 ESPECIFICACIÓN DE CASOS DE USO	76
3.4 DISEÑO.....	97
3.4.1 DIAGRAMAS DE DISEÑO DEL MÓDULO DE CUENTAS CORPORATIVAS	98
3.4.2 ESPECIFICACIÓN DEL DISEÑO DE LOS CASOS DE USO	106
3.4.3 MODELO DE DATOS DE ANTARES	132
CAPÍTULO IV: FASE DE CONSTRUCCIÓN	141
4.1 DESCRIPCIÓN DE LA FASE DE CONSTRUCCIÓN.....	141
4.2 IMPLEMENTCIÓN DE ANTARES.....	142
4.2.1 CONVENCIONES DE CODIFICACIÓN.....	142
4.2.2 DESCRIPCIÓN DEL CÓDIGO FUENTE DEL SISTEMA ANTARES	147
4.3 CRITERIOS DE EVALUACIÓN	152
4.3.1 REQUISITOS NO FUNCIONALES.....	153
CAPÍTULO V: FASE DE TRANSICIÓN	157
5.1 LIBERACIÓN DEL SISTEMA ANTARES	157
5.2 CONFIGURACIÓN DE AMBIENTE	159
5.2.1 CONFIGURACIÓN DE TOMCAT 5.0	159
5.3 INSTALACIÓN DE SOFTWARE.....	162
5.3.1 COMPILACIÓN DEL PROYECTO ANTARES.....	162
5.3.2 DEPLOYMENT DE ANTARES: PUBLICACIÓN	166
5.4 IMPLEMENTACIÓN DE BASE DE DATOS	169
5.4.1 CREAR LA BASE DE DATOS DE LA GCSR9	169
5.4.2 CARGA DE INFORMACIÓN	175
CONCLUSIONES	178
Anexo A: GLOSARIO DE TÉRMINOS.....	180
Anexo B: FORMATO DE CONTROL DE CAMBIOS DE ALCANCE.....	183
Anexo C: FORMATO DE EVALUACIÓN DE SATISFACCIÓN DEL CLIENTE	184
Anexo D: FORMATO DE CONTROL DE ASUNTOS.....	187
Anexo E: FORMATO DE CONTROL DE ACTIVOS	188
REFERENCIAS BIBLIOGRÁFICAS.....	189

INTRODUCCIÓN

La Gerencia de Calidad de Servicio Región 9 (GCSR9), es un área operativa de una empresa dedicada al negocio de las telecomunicaciones y que se enfoca principalmente en el rubro de la Telefonía Celular. Pertenece a la Dirección de Ingeniería de Red cuyo principal objetivo es, la Optimización General de la Red GSM en las bandas 1900 y 800 [MHz]; así como garantizar la disponibilidad y calidad en los servicios de voz y datos para sus clientes y cuentas corporativas con las cuales mantiene un contrato.

Al interior de la GCSR9 se desarrollan una serie de proyectos (procesos, tareas y procedimientos) en conjunto con otras áreas de la empresa; la información generada de dichos proyectos se concentra en distintas y variadas fuentes, que van desde servidores de bases de datos, hasta unidades compartidas de disco, y en general la misma es presentada en diversos formatos. En virtud de lo anterior, en las labores diarias, la GCSR9 carece de un mecanismo de control riguroso que permita una interacción fluida y eficaz con las otras áreas; así también al día de hoy, no se cuenta con un estándar para el seguimiento de dichos procesos.

Los objetivos que se pretenden abordar con este proyecto de desarrollo de software son los siguientes:

- Desarrollar un sistema que satisfaga la necesidad de centralizar y controlar la información generada de la atención y seguimiento de los proyectos en los que intervienen diversas áreas de la empresa, tales como Mesa de Ayuda, Ventas, Centrales, Troncales, Nuevas Tecnologías, Compras; y cuya responsabilidad corre a cargo de la GCSR9.
- Construir una arquitectura de desarrollo robusta y fácilmente extensible, que permita implementar y administrar un sistema WEB para la gestión de los proyectos de la GCSR9, mediante el cual sea posible tener acceso vía remota, a la información de los mismos.
- Centralizar la información del negocio en una base de datos y realizar un plan de respaldos de la misma, reduciendo con esto los riesgos de la posible pérdida de información de la empresa; y de esa manera contribuir con la GCSR9 en la tarea de optimizar recursos, tiempo y costos.

- Proveer de una herramienta a los usuarios (analistas, jefes de área y personal general de la empresa) que permita la consulta de status y el seguimiento de los proyectos; así como automatizar las notificaciones y actualizaciones más trascendentes durante la atención de cada proyecto de la GCSR9.
- Mantener el sistema Antares en un ambiente de producción fiable, robusto y estable, que permita la disponibilidad de la información de los proyectos de la GCSR9 en todo momento.

Al finalizar el proyecto Antares, se estima que la GCSR9 cuente con los siguientes beneficios en su operación diaria:

- Obtención de la información en tiempo, con consistencia, independencia e integridad requerida para las labores de atención, seguimiento y consulta de los proyectos de la GCSR9.
- Conocer las acciones emprendidas en tiempo real, por parte de los usuarios responsables de la atención de cada proyecto; para permitir así a los jefes y directivos la evaluación del desempeño del personal a su cargo.
- Apoyar a los jefes de área de la GCSR9 en la planeación y distribución de cargas de trabajo entre el personal a su cargo.

CAPÍTULO I: CONCEPTOS GENERALES

1.1 LAS APLICACIONES WEB

El mundo del desarrollo de aplicaciones se encuentra inmerso en una nueva etapa de transformación y evolución hacia nuevos esquemas de trabajo.

Los factores determinantes de dicho cambio los podemos encontrar en la necesidad de utilizar Internet como vehículo de intercambio por parte de diversos sectores laborales.

Particularmente, la GCSR9 requiere establecer mecanismos de interacción con sus clientes más dinámicos, de modo que el volumen de negocio se incremente al explotar los recursos disponibles, tal es el caso del canal electrónico de comunicación. Por otro lado, también necesitan unas relaciones empresariales más ágiles aprovechando al máximo este mismo marco tecnológico. Aunado a los elementos mencionados anteriormente, nos encontramos con que, una aplicación WEB representa una poderosa herramienta para el usuario de la GCSR9, que le permite desplazarse y tener a su disposición información en línea, mediante un acceso rápido y sencillo. Esto le permite tener un mayor grado de interacción con otras áreas de la empresa, y obteniendo información de una fuente centralizada de datos; todo esto naturalmente, sin los tradicionales esfuerzos de configuración que requieren algunas aplicaciones.

Con el paso del tiempo la Internet se ha convertido en el principal entorno de trabajo para el desarrollo de aplicaciones que gestionan información, haciendo que su alcance sea mayor que ningún otro medio hasta el momento, y para muestra tenemos que, mediante cualquier dispositivo que tenga acceso a Internet y un programa navegador, es posible acceder a infinidad de sitios Web basados en este paradigma.

Sin embargo, en la actualidad la comunicación entre servidores es complicada, sobre todo si los sistemas que desean comunicarse residen en plataformas distintas. Es por ello que surge la necesidad de contar con sistemas que permitan la integración de aplicaciones y/o herramientas, teniendo como objetivos, el mejorar los procesos y funciones del negocio, reducir tiempos y costos, teniendo disponible en todo momento, información en línea, integral, completa y confiable.

1.2 PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE

1.2.1 NECESIDAD DE UTILIZAR UNA METODOLOGÍA

Es evidente que todas las empresas profesionales, para lograr la consecución de sus objetivos, requieren dirección organizativa para el desarrollo de sus procesos. El ámbito de sistemas y particularmente, la GCSR9 no queda exenta a esta situación, de tal manera que los recursos que laboramos en ella, debemos apegarnos a una serie de políticas y lineamientos, para el desempeño de nuestras tareas diarias. Así mismo, para tener un control de métricas y estimaciones acerca de un proceso y contar con un adecuado seguimiento del mismo, implica necesariamente contar con una adecuada dirección sobre las políticas y procedimientos de la organización.

Con base en lo anterior, el objetivo de este apartado es mostrar los beneficios que aporta, el desarrollo de un proyecto de software apegado a una metodología estable, probada y muy aceptada en la comunidad de ingenieros de software, como lo es el Proceso Unificado de Desarrollo.

1.2.2 PROCESO DE DESARROLLO DE SOFTWARE

En la ingeniería de software, y particularmente para la GCSR9 la meta del proceso a abordar, consiste en construir el producto de software Antares, definiendo quién está haciendo qué, cuándo y cómo se deben alcanzar o cumplir con las metas establecidas por la GCSR9.

El proceso fungirá como guía a usuarios y ejecutivos, además de permitirles comprender el papel que desempeñan en el desarrollo en el que se encuentran implicados. Para lograr lo anterior, el responsable del proceso deberá equilibrar cuatro conjuntos de circunstancias:

- Tecnología propuesta.
- Herramientas.
- Personas relacionadas. (Ya que el proceso descansa en una variedad extensa de actividades).
- Patrones de organización. (El proceso debe adaptarse a la realidad del momento de la organización).

El proceso de software debe ante todo, reducir las dificultades que afrontan los integrantes de la GCSR9, para afrontar las cargas de actividades, además de proveer una forma coordinada de trabajo, mediante tareas comunes.

1.2.3 APLICACIÓN DEL RUP A ANTARES

Tal como lo especifica el RUP, para el desarrollo de Antares se utiliza el Lenguaje de Modelado Unificado (UML) para preparar todos los esquemas del sistema. A grandes rasgos, la aplicación del Proceso Unificado de Desarrollo de Software en el proyecto Antares, consistió en definir el conjunto de actividades necesarias para transformar los requerimientos tecnológicos de la GCSR9. Véase la Fig. 1.1



Fig. 1.1 El proceso RUP

En este punto, es conveniente hablar acerca de los casos de uso, ya que estos dirigen todas las actividades y esfuerzos involucrados para transformar los requerimientos tecnológicos de la GCSR9, en un producto de Software.

1.2.4 DEFINICIÓN DE UN CASO DE USO

Un caso de uso es una descripción del conjunto de secuencias de acciones (incluyendo variaciones), que un sistema lleva a cabo, para conseguir un resultado específico para un usuario del sistema.

Según Ivar Jacobson *“Los casos de uso son las funciones que proporciona un sistema para añadir valor a los usuarios. Tomando la perspectiva de cada tipo de usuario, podemos capturar dichas funciones necesarias para hacer su trabajo”* [1].

Bajo la perspectiva anteriormente descrita, la construcción de Antares y en general de cualquier sistema, se necesita conocer a los usuarios y las necesidades y/o expectativas que estos tienen. El usuario representa a “alguien” o “algo”, (como puede ser un sistema externo), que interactúa con el sistema Antares. Para ejemplificar el tipo de interacción al que se hace alusión, se tiene la siguiente situación:

Una persona que utiliza un cajero automático. La persona inserta su tarjeta, ingresa ciertas claves y responde algunas preguntas que le hace el cajero y recibe una cantidad de dinero.

Dicho en pocas palabras, los casos de uso son el mapeo que representan los requisitos funcionales que el sistema deberá satisfacer. Más adelante se describirá el Modelo de Casos de Uso del sistema, el cual constituye la funcionalidad total del sistema Antares. Respondiendo así a la pregunta: ¿Qué debe hacer el sistema?

Finalmente mencionar que, la importancia de los casos de uso radica en qué aparte de fungir como una herramienta mediante la que se especifican los requisitos del sistema; también guían el proceso de desarrollo:

- El analista genera el modelo de casos de uso.
- El programador construye el sistema con el objetivo de satisfacer los casos de uso que le fueron asignados.
- Un “tester” o ingeniero de prueba, evalúa la correcta implementación del producto, mediante casos de prueba, los cuales consisten en probar la funcionalidad del sistema, contra las especificaciones de los casos de uso.

1.2.5 LA IMPORTANCIA DE UNA CORRECTA ARQUITECTURA

El papel que juega la arquitectura para cualquier sistema es de vital importancia; a este respecto y haciendo una analogía con la arquitectura en la construcción de edificios, tenemos que en ese rubro, la primera tarea a realizar es evaluar varias perspectivas como la estructura general, los servicios a cubrir, la electricidad y demás sistemas de que se deberá componer. Para el rubro de sistemas, el equipo de trabajo debe tener una imagen completa de lo que será el producto final, antes de que se comience con la construcción del mismo. Con base en lo anterior, la arquitectura de Antares se describe mediante diferentes vistas del sistema en construcción.

La arquitectura surge de las necesidades de la empresa, en este caso se verá que los requerimientos funcionales de la GCSR9 jugaron un papel muy importante en la definición de la arquitectura del sistema. El resto de los factores que influyen en la definición de la arquitectura, tienen que ver con la plataforma en la que debe operar Antares (Arquitectura hardware, sistema operativo, sistema de gestión de base de datos, protocolos de comunicación en red, marcos de trabajo o frameworks de que se dispone, entre otros), así como con otro tipo de requisitos no

funcionales como el rendimiento esperado por el sistema, fiabilidad y facilidad de uso para los usuarios.

La arquitectura de Antares debe evolucionar al paralelo de la definición de los casos de uso, esto porque a los casos de uso les corresponde describir la funcionalidad a implementar, mientras que a la arquitectura le corresponde la forma en la que se efectuará dicha implementación. Para trabajar en esa forma, el arquitecto de software debe enfocarse en la comprensión general de las funciones o casos de uso claves para el sistema; generalmente constituyen el 10% del total del modelo de casos de uso y es, a partir de ellos que, se crea un esquema de borrador sin mucho nivel de detalle, pero que sí incluye los datos de la plataforma y una comprensión general de la funcionalidad que el sistema deberá cubrir.

Posteriormente el arquitecto deberá trabajar con un subconjunto mayor de casos de uso de las funciones claves del sistema, esto con la finalidad de adoptar una perspectiva de casos de uso como subsistemas a los que se requiere definir a detalle. Finalmente, en la medida en que maduran los casos de uso, madura y sufre modificaciones menores la arquitectura del sistema hasta alcanzar una arquitectura estable.

1.2.6 PROCESO DE DESARROLLO ITERATIVO E INCREMENTAL

El Proceso Unificado de Desarrollo de Software propone abordar cada proyecto de desarrollo de Software como varios proyectos más pequeños, o mini proyectos generados de forma iterativa y para los que cada iteración constituye un incremento.

Dicho de otra forma las iteraciones se refieren a los pasos del flujo de trabajo, mientras que los incrementos se refieren al crecimiento del software visto como producto.

Ahondando en el tema, la iteración se enfoca en un cierto grupo de casos de uso que componen el producto de software. Posteriormente se identifican los riesgos más importantes, de tal forma que el final de una iteración sea el punto de comienzo de la siguiente, esto con el objetivo de ir refinando el producto final, por lo que no todo incremento es aditivo en cuestión de código fuente y funcionalidad; también puede abordar cambios y/o mejoras a las casos de uso existentes.

El objetivo a cubrir con esta perspectiva, es que tanto el analista como el programador aborden cada iteración identificando los casos de uso más relevantes, creen un diseño utilizando la

arquitectura seleccionada como guía, implementen el diseño mediante componentes de software; y finalmente que verifiquen que los componentes satisfacen en su totalidad, a los casos de uso que representan. En resumen cada iteración incluye las fases de análisis, diseño, implementación y prueba de los procesos de desarrollo de software convencionales.

Finalmente si una iteración cumple con sus objetivos, se continúa con el desarrollo de la siguiente iteración.

La razón por la que la identificación de riesgos juega un papel preponderante en el proceso de desarrollo de software, se justifica por que contribuye a minimizar problemas inesperados y en consecuencia, a tener una mejor planificación inicial, lo cual a su vez ayuda a seleccionar un número menor de iteraciones para el proyecto. El impacto es directamente en la cuestión económica de la empresa, pues solo se tendrán las iteraciones necesarias, evitando así consumir más esfuerzo y tiempo. En general, los beneficios que aporta un proceso iterativo son:

- La iteración controlada reduce el costo de riesgo en los incrementos innecesarios, ayudando a la organización a no perder el valor del producto interno, en caso de ocurrir fallas.
- La iteración reduce el riesgo de no lanzar el producto de software a producción, en el calendario previsto. Esto porque la identificación de riesgos en fases tempranas de desarrollo permite planificar el tiempo en el que los programadores deberán dar solución a los mismos. Evitando así que los problemas y errores de software sean detectados hasta la etapa de pruebas del sistema, que es una etapa crítica del proyecto, en la que no se tiene tanto tiempo para que los programadores dediquen su tiempo en atender nuevas incidencias. En consecuencia de lo anterior, contribuye a no retrasar las entregas programadas.
- Los programadores trabajan de manera más eficiente pues se enfocan en conseguir resultados objetivos a corto plazo.
- Se tiene una visión más objetiva respecto a las necesidades del usuario, ya que los requerimientos no pueden definirse completamente al inicio del proyecto y necesariamente se van refinando en iteraciones sucesivas.

En resumen, según Ivar Jacobson *“La arquitectura proporciona la estructura sobre la cual se deben guiar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración”* [2].

1.2.7 LA VIDA DEL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE (RUP)

El ciclo de vida del RUP, consiste en repetirse a lo largo de un conjunto de ciclos, que a su vez constituyen la vida de un sistema. Al finalizar cada ciclo, podemos hablar de que se ha concluido una versión del producto Antares. Véase las Fig. 1.2.

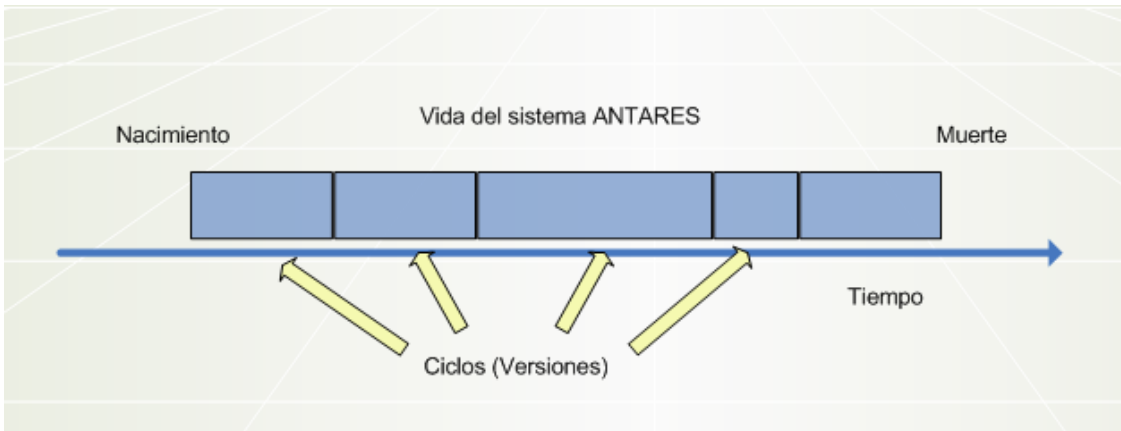


Fig. 1.2 Vida del Sistema Antares

Cada ciclo consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. A su vez cada fase se subdivide en iteraciones. Véase las Fig. 1.3.

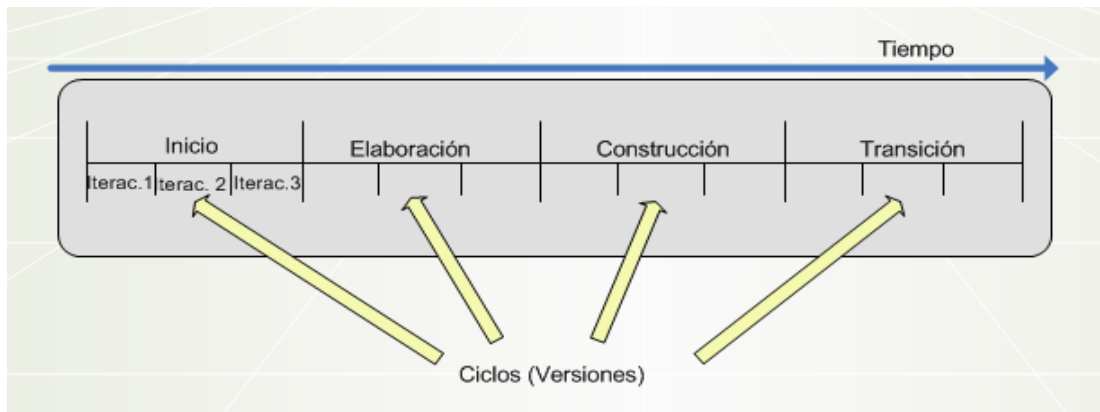


Fig. 1.3 Un ciclo con sus fases e iteraciones

1.2.8 EL PRODUCTO ANTARES

Bajo la perspectiva que plantea el RUP, al finalizar cada ciclo de desarrollo de Antares, estamos produciendo una nueva versión del mismo, de tal manera que cada versión constituye un producto listo para entregarse a la GCSR9, quienes a su vez pueden comenzar a utilizarlo en su operación diaria. Como tal, el producto consta de un cuerpo de código fuente incluido en componentes independientes que pueden compilarse, en el caso del código java interpretarse y ejecutarse; además de manuales técnicos y de usuario.

El usuario aceptará el producto como terminado, en el momento que este cumpla con el alcance establecido; es decir, que satisfaga los requerimientos planteados durante la fase de análisis del proyecto de desarrollo.

El producto terminado incluye:

- Requerimientos funcionales.
- Casos de uso.
- Modelo de la arquitectura.
- Modelo visual.
- Diagrama Entidad Relación.
- Especificaciones no funcionales.
- Casos de prueba.
- Manuales técnicos y de usuario.

La anterior lista muestra los elementos del proyecto en desarrollo, ya que permiten a clientes, usuarios, analistas, programadores, ingenieros de prueba y líderes:

Especificar, diseñar, implementar, probar y en general, utilizar el sistema. En lo que respecta a entregables hacia el cliente u organización para la que se está desarrollando el sistema en este caso, constituyen el mínimo grupo de entregables que deberán proporcionarse a la GCSR9.

El modelo Visual de Antares muestra todas las representaciones del producto de software, además de que su labor es la de representar el sistema como un todo. El modelo visual incluye:

- Una representación de la arquitectura utilizada.
- Un modelo con todos los casos de uso.

- Un modelo de análisis, para refinar los casos de uso, así como también para fijar el alcance de la funcionalidad del sistema, mediante un conjunto de objetos que proporcionan los comportamientos (Métodos en el caso del paradigma Orientado a Objetos).
- El modelo de diseño que define:
 - La estructura estática de Antares, a modo de subsistemas, clases e interfaces.
 - Los casos de uso en términos de colaboraciones (Diagramas de Colaboración).
 - Diagramas de secuencia.
- El modelo de implementación que incluye los componentes de código fuente y las clases correspondientes a cada componente de que se compone Antares.
- El modelo de despliegue o de distribución, (Diagramas de Despliegue y Diagrama de Interfaces), los cual define los nodos físicos, así como la correspondencia que estos tienen con los componentes.
- El modelo de casos de prueba, que consiste en implementar clases que verifiquen el correcto funcionamiento de los casos de uso.

1.2.9 FASES DEL CICLO

Dentro del marco de tiempo asignado al proyecto Antares, se efectúan varios ciclos para lograr la obtención del producto esperado. A lo largo de cada ciclo, identificamos una división que consta de 4 fases, mismas que pueden ser visualizadas mediante una secuencia de modelos.

En cada fase los miembros del grupo de trabajo, pueden descomponer el trabajo en iteraciones con sus respectivos incrementos resultantes. La finalidad de dividir los ciclos en fases radica en permitir un mejor control en el progreso del trabajo, ya que al finalizar cada fase se presenta un producto entregable, que consiste en la toma de decisiones que redefinirán el rumbo del proyecto; y para lograr lo anterior, tanto desarrolladores del proyecto, como directivos y clientes deben examinar el proceso de desarrollo con base a un determinado grupo de artefactos o “entregables” que, pueden ser documentos y/o modelos que han sido desarrollados para cubrir el alcance del proyecto en la fase respectiva.

En este sentido, se estableció un compromiso con la GCSR9, en el cual se especifica la entrega puntual de una lista de documentación y diagramas que deberán acompañar la liberación de cada versión de Antares.

Al concluir cada fase, se obtiene información importante para la planificación del proceso de desarrollo, entre la que destaca, la estimación de tiempos y asignación de recursos que deberá emplearse en el proyecto para abordar la siguiente fase de desarrollo.

FASE DE INICIO:

Se desarrolla una descripción del producto final a partir de la lluvia de ideas acerca de los requerimientos funcionales de la GCSR9. En respuesta a esto, un servidor debe presentar un análisis del negocio para el producto. En esta fase se responden las siguientes preguntas:

- ¿Cuáles son las principales funciones del sistema para sus usuarios más importantes?
La respuesta consiste en el Modelo de Casos de Uso simplificado, con los casos de uso críticos para Antares. En este caso estaremos describiendo los casos de uso referentes a los flujos de trabajo de la Atención a las Cuentas Corporativas de la GCSR9.
- ¿Cómo podría ser la arquitectura del sistema?
Realizar un esbozo que muestre a Antares como subsistemas inter relacionados entre sí. También es necesario identificar y clasificar riesgos.
- ¿Cuál es el plan de trabajo y cuánto costará el desarrollo del producto?
Realizar una aproximación de la estimación del proyecto de desarrollo.

FASE DE ELABORACIÓN:

Se especifican a detalle la mayoría de los casos de uso que el sistema Antares deberá incluir, y para lo cual se estarán describiendo las principales tareas involucradas con el proceso y seguimiento de las cuentas corporativas. Es en esta fase donde se diseña la arquitectura que soportará el desarrollo de la aplicación empresarial bajo estándares J2EE. La arquitectura se expresa en forma de vistas de todos los modelos del sistema (análisis, diseño, implementación y despliegue). Para el caso de la vista del modelo de implementación se deben incluir los componentes necesarios a fin de probar que la arquitectura es ejecutable.

El resultado de esta fase es la obtención de una línea base de la arquitectura; así como la adecuada toma de decisiones por parte de los directivos del proyecto, para planificar las actividades y estimar los recursos necesarios para concluir el proyecto. Básicamente se toman como directriz los siguientes puntos:

- Verificar que se cumple con la satisfacción de los requerimientos funcionales de la GCSR9.
- Describir la funcionalidad y las ventajas que ofrece el desarrollo a partir de los Frameworks (marcos de trabajo): Struts 1.2, Spring 2.0 y Hibernate 3.
- Se debe fijar un alcance para el proyecto, de tal manera que sea factible el establecer un contrato que considere y controle los riesgos como para ser capaces de comprometernos con la GCSR9 y la Dirección de Ingeniería de Red.

FASE DE CONSTRUCCIÓN:

En esta fase se construye el producto en lo que a componentes de software se refiere. Es en esta fase, donde la Arquitectura de Antares es puesta a prueba, con la finalidad de crecer y extenderse hasta convertirse en el sistema completo, y listo para ser entregado a la comunidad de usuarios.

Al concluir esta fase se debe garantizar que el producto contenga todos los casos de uso, que los directivos y usuarios han acordado fijándolos como alcance de esta versión de Antares.

FASE DE TRANSICIÓN:

Esta fase se refiere al periodo en el que Antares deberá convertirse en una versión beta, destinada a ser utilizada por un número reducido de usuarios con experiencia; esto con la finalidad de probar y ratificar el impacto y la correcta funcionalidad del sistema en sus procesos diarios.

Es en esta fase en la que deben identificarse los fallos aún existentes, tanto en funcionalidad del sistema como de flujo de trabajo. Posteriormente se informa acerca de las incidencias detectadas a analistas y programadores, quienes estarán dando solución a los problemas; así como también estarán incorporando mejoras sugeridas a la versión del producto. Finalmente otra tarea de suma importancia y que debe abordarse en esta fase, es la elaboración de manuales técnicos y de usuarios; así como cualquier tipo de actividad encaminada a formar o capacitar al usuario final de Antares.

1.3 ESTÁNDAR J2EE

Java comienza con el lanzamiento de un “Kit” de desarrollo conocido como JDK (Java Development Kit) mismo que fué extendiéndose con interfaces y bibliotecas según el mundo corporativo lo exigía. El éxito obtenido por este lenguaje repercute en el mundo empresarial, por lo que Sun Microsystems es motivado a incorporar nuevas extensiones a las API (Application Programming Interfaces), dando como resultado la iniciación del Programa Comunitario de Java: JCP, el cual reunió a usuarios corporativos, proveedores y técnicos para desarrollar una forma estándar de las API empresariales de Java. El resultado de la evolución de este programa fué la plataforma Java 2 Enterprise Edition, mejor conocida como J2EE.

Los técnicos de Sun Microsystems reescribieron la forma en que los desarrolladores construyen sistemas distribuidos centrados en la Web a gran escala mediante el uso de J2EE; esto con la finalidad de atender y resolver problemas comunes a los que se enfrentan los desarrolladores.

1.3.1 VISIÓN GENERAL DE J2EE

Según Jim Keogh, en su manual de referencia J2EE: *“Las tecnologías desorganizadas dan lugar a tecnologías de soporte, las cuales se convierten en componentes integrales de ellas y hacen que la tecnología desorganizada opere con mayor efectividad”* [3]. Bajo esta perspectiva J2EE es una tecnología de soporte asociada de manera muy cercana a las redes, sobre todo a la Internet; misma que permite a los ingenieros de software y programadores crear una infraestructura de software robusta y en general, permite que la construcción de aplicaciones basadas en Web sea rentable. Cabe mencionar que J2EE surge en medio del auge de desarrollos por parte de los departamentos de informática, cuyo enfoque tiene sus cimientos en la arquitectura cliente-servidor, en la que los clientes solicitan servicios de software que se ejecuta del lado del servidor. Y es precisamente a partir de ese paradigma de desarrollo, que los sistemas servidores y la infraestructura han crecido cada vez más, convirtiéndose de una arquitectura de dos capas, hasta convertirse en una arquitectura multicapas para la cual, una petición de un cliente a un servidor, puede generar a su vez peticiones a otros servidores conectados a través de redes. Adicionalmente, para asegurar el crecimiento de la Internet, es necesario contar con aplicaciones Web que puedan proveer servicios, además de interactuar con muchos otros, como por ejemplo bases de datos y servicios de mensajería.

La razón por la que Antares sigue el estándar J2EE es porque, permite integrar sistemas y servicios de los servidores, con la finalidad de producir un entorno más escalable y robusto para aplicaciones centradas en Web, como lo es el sistema requerido por la GCSR9. Para atender a las necesidades de la GCSR9, J2EE permite el diseño de la aplicación Antares, de forma tal que atienda a todos los usuarios de la GCSR9, las 24 horas al día y siete días a la semana:

- 44 Analistas: Ingenieros de Soporte.
- 4 jefes de áreas.
- 1 gerente.
- Alrededor de 120 usuarios más de Centros de Atención a Clientes).

Otra ventaja que provee el apego al estándar J2EE consiste en que permite encapsular la funcionalidad en componentes; situación que permite al programador dividir la aplicación de acuerdo a las funciones principales y posteriormente distribuir esas funciones entre los componentes a desarrollar. Todos los componentes J2EE están escritos en Java y se incluyen interfaces y bibliotecas necesarias para abordar necesidades empresariales comunes, siendo los principales componentes los Enterprise Java Beans (EJB's), Servlets y las Java Server Pages (JSP's), además de incluir siete servicios:

- 1) Compatibilidad con CORBA para interactuar con sistemas heredados (Legacy systems).
- 2) JavaMail que permite comunicarse mediante el envío y recepción de mensajes de correo electrónico.
- 3) JMS Servicios de Mensajería Java, para incluir en los programas java, un enlace de transmisión entre componentes.
- 4) JNDI Interfaz de nombres de directorios para Java. Este servicio permite que los objetos puedan ubicarse en distintos puntos de servidores conectados a la infraestructura de la empresa.
- 5) JTA API Java para la Gestión de Transacciones. Una transacción puede involucrar varios componentes.
- 6) JDBC Para permitir el acceso a información contenida en bases de datos.
- 7) Descriptores de despliegue XML. Indican al contenedor cómo y dónde desplegar los componentes.

1.3.2 ARQUITECTURA J2EE

De acuerdo con la definición de Jim Keogh, *“J2EE es una Arquitectura de cuatro capas: Cliente, mejor conocida como Capa de Presentación o Aplicación, la capa Web, la Capa de Negocio y la Capa de sistemas de información empresarial. Cada una de ellas se enfoca en proporcionar a una aplicación, un tipo específico de funcionalidad” [4].*

De lo anterior solo vale la pena agregar que el estándar J2EE constituye una opción para satisfacer las necesidades actuales del mercado. Necesidades que se encaminan a satisfacer las expectativas de ejecutivos, clientes y demás usuarios que demandan acceso instantáneo a la información, a cualquier hora y día de la semana y desde cualquier lugar. Los fines de la información son muy variados, y van desde consultar algún catálogo en línea de una empresa como la GCSR9 por ejemplo, colocar una solicitud de proceso o recuperar información de cuentas; o simplemente enviar y recibir correo electrónico.

La comunidad informática tuvo que darse a la tarea de reforzar las redes y sistemas para dar servicio a una población de usuarios que crecía a pasos agigantados. La atención se enfocó en el modelo de la arquitectura tecnológica, siendo ampliamente utilizado el modelo cliente-servidor de dos capas. Uno de los inconvenientes a esta arquitectura es que depende en gran medida de mantener actualizado el software de los clientes, situación que se torna costosa de mantener para las empresas que cuentan con varias aplicaciones o ambientes de intranets.

Como respuesta a la necesidad anterior, los sistemas multicapa basados en Web que siguen la arquitectura J2EE, no exigen la actualización del software cliente, cada vez que se modifica la presentación y funcionalidad de la aplicación. El objetivo fundamental de la arquitectura multicapa es compartir recursos entre clientes, siendo esta la filosofía del diseño para desarrollar software.

Para abordar el tema de la Arquitectura, enunciaremos los siguientes componentes que la constituyen:

- 1) Capa. Es un concepto abstracto que define un grupo de tecnologías que proporcionan uno o más servicios a sus clientes.
- 2) Cliente. Es todo recurso que envía una petición de un servicio a un proveedor de servicios.
- 3) Servicio. Es cualquier recurso que percibe y atiende peticiones de clientes. En este punto es importante mencionar que dicho recurso a su vez, puede hacer peticiones a otros recursos para atender la petición inicial.
- 4) Componente. Es parte de una capa consistente de un conjunto de clases o programas que realiza una función para proporcionar el servicio. Los componentes son los responsables de implementar la lógica del negocio.

- 5) Contenedor. Es el software que gestiona el componente y le proporciona servicios de sistema. El contenedor debe encargarse de tareas como la persistencia, la gestión de recursos, la seguridad y otros servicios a nivel de sistema.

Así tenemos que en una arquitectura multicapa como la J2EE, cada capa contiene servicios, mismos que incluyen objetos de software, sistemas de gestión de bases de datos, ó conexiones con sistemas heredados. Con esto la funcionalidad de la aplicación se divide en componentes lógicos asociados a cada capa; de tal manera que, cada componente es un servicio que se implementa y mantiene de forma independiente de los otros servicios.

1.3.3 APLICACIÓN DE J2EE A ANTARES

Para el caso particular del desarrollo de Antares, se emplean los siguientes componentes:

- Servlets. Programas java que se ejecutan en un servidor Web, en el Sistema Antares realizan la labor del control en el patrón de diseño MVC, fungiendo como una capa intermedia entre la petición realizada por usuarios de la GCSR9 mediante el navegador Web y las clases de negocio de Antares.
- JSP's. Son páginas Web codificadas en HTML con código java embebido. Son la tecnología lanzada por Sun, para desarrollar aplicaciones Web con contenido dinámico. En el ámbito de Antares son utilizados en la capa de presentación (vistas) de la arquitectura Web del Sistema.
- Spring. Se emplea este framework para la gestión y transacciones de Beans.

En lo que respecta a los servicios de J2EE, Antares utiliza:

- JDBC que permite a las clases de negocio invocar servicios de conexión e interactuar con el DBMS MySQL que es el encargado de gestionar el repositorio de información de Antares.
- JavaMail como recurso eficiente para el intercambio de información entre miembros de la GCSR9 y contactos con las Cuentas Corporativas de la Región 9.
- Descriptores de despliegue propios de la arquitectura de Antares: WEB.xml, applicationContext.xml, struts-config.xml y hibernate.cfg.xml como una forma de almacenar, manipular e intercambiar información contenida en los documentos e indicar al contenedor cómo y dónde desplegar los componentes.

- JNDI para configurar y localizar el datasource (Acceso a datos) configurado con nuestro repositorio en el DBMS MySQL.

Breve descripción de las capas:

- 1) Capa cliente: Está compuesta por los programas que interactúan directamente con el usuario de Antares; en este caso nos referimos a los navegadores de Internet como Mozilla Firefox 3, Internet Explorer 5 o 6 u Opera por ejemplo. Estos programas piden datos a los usuarios de Antares y convierten la respuesta en peticiones que son enviadas a un componente encargado de procesar peticiones. Otra tarea fundamental del programa cliente es, traducir la respuesta del servidor a pantallas y texto personalizado que se presenta al usuario.
- 2) La capa Web que proporciona a Antares y en general, a las aplicaciones J2EE la funcionalidad de interacción entre redes, incluso con la Internet. Los componentes de esta capa utilizan el protocolo de transferencia de Hipertexto HTTP para recibir peticiones y enviar respuestas a clientes de cualquier capa. Esta capa está representada por la interfaz Web de Antares (vistas en JSP's), que se ejecuta en un servidor de aplicaciones.
- 3) La capa de negocio encapsula los datos de la empresa y como tal, contiene la lógica del negocio de las aplicaciones J2EE. Particularmente, para la arquitectura de Antares es en esta capa donde se encuentran las clases que implementan a través de servicios, la funcionalidad de los procesos de la GCSR9 referentes al seguimiento de las Cuentas Corporativas. Cabe mencionar que se utilizan clases tipo Bean para transferir información entre las capas. Este tema será abordado a detalle en el siguiente apartado de patrones de diseño.

Esta capa juega un papel muy importante dentro de la aplicación, pues es a través de ella, como se puede tener acceso a los datos del negocio, en forma concurrente y de forma tal, que no se afecte el rendimiento.

- 4) La capa de acceso a datos (DAO Data Access Object) de la GCSR9, es la capa que permite y controla el acceso al repositorio de información de la GCSR9, que es en este caso, la base de datos en MySQL. Esta capa sirve como puente entre la capa de negocio (servicios) y la base de datos, y su objetivo fundamental es la gestión final de los datos del negocio.

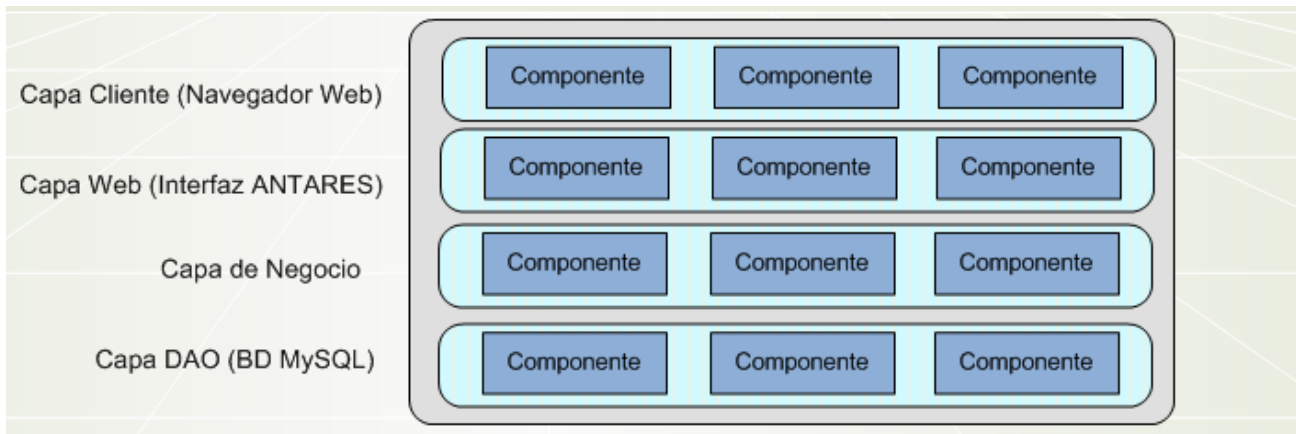


Fig. 1.4 Capas J2EE del sistema Antares.

1.4 PATRONES DE DISEÑO

1.4.1 IMPORTANCIA DEL USO DE PATRONES

El objetivo de los patrones de diseño es desarrollar una forma estandarizada para representar soluciones generales a problemas generales que se presentan durante las fases de desarrollo de software.

Al afrontar el proyecto de software requerido por la GCSR9, las expectativas por parte de los directivos del área de sistemas eran muy puntuales, y giran en torno a las siguientes indicadores: Automatización de procesos relacionados con el seguimiento de cuentas corporativas, eficiencia, flexibilidad, facilidad de uso, extensibilidad y por supuesto bajo costo.

Para atender las necesidades y expectativas de la GCSR9, los patrones de diseño han servido de guía para la construcción de Antares, ya que son soluciones documentadas y probadas para situaciones y problemáticas comunes. Dicho de otra manera, los patrones de diseño, nos permitieron adaptar soluciones generales a los requerimientos y necesidades de la GCSR9.

Una causa más que justifica su utilización radica en que contribuyen a responder a las siguientes preguntas: ¿Cómo podemos diseñar software de manera efectiva?, ¿Cómo podemos aplicar la experiencia obtenida en el pasado a los proyectos futuros?. Citando la obra de Stephen Stelting: *“Los patrones de diseño representan una evolución importante en la abstracción y reutilización del software”* [5].

Es importante abordar estos conceptos ya que representan dos directrices esenciales para la programación de sistemas. La abstracción representa la forma en la que el ingeniero de software aborda un problema complejo y lo divide en otros más simples. El enfoque es dar la solución a esos problemas simples a modo de bloques de sistemas. El concepto de reutilización de código es de sumo interés para el desarrollo de sistemas, ya que constituye una manera eficiente de reducir costos y tiempos en un proyecto, buscando también estandarizar la construcción y en general permiten una mayor productividad del código. Reutilizar código que previamente ha sido utilizado y probado implica reducir tiempo en actividades propias de la construcción de sistemas como depuración, documentación y soporte.

1.4.2 PATRONES DE CREACIÓN

En el paradigma orientado a objetos, estos patrones cubren la función de facilitar la creación de objetos en un sistema. Para Antares, así como para cualquier sistema orientado a objetos necesitan crear instancias de las muchas clases de las que se componen. A este respecto, estos patrones cuentan con las siguientes características:

- Instanciación genérica para permitir la creación de objetos dentro del sistema, sin tener que identificar una clase específica dentro del código.
- Simplicidad: Para poder instanciar un objeto no se requiere escribir gran cantidad de código fuente.
- Para un mejor control del sistema, permite restricciones sobre la creación de cierto tipo o número de objetos.

En el desarrollo de Antares se emplearon algunos conceptos del patrón Abstract Factory. A continuación se presenta la definición del mismo.

PATRÓN ABSTRACT FACTORY

El patrón Fábrica Abstracta también es conocido como “Kit Toolkit” o “Grupo de Herramientas”.

- a) Propiedades del patrón: Es para crear objetos a nivel Componente.
- b) Propósito: Proporcionar un contrato para crear familias de objetos relacionados o dependientes, sin tener que especificar su clase concreta.

- c) Introducción: Con este patrón se define una clase que funge de Fábrica para generar objetos que siguen un patrón general. En tiempo de ejecución, esta fábrica se asocia con un número concreto de fábricas que tienen su propia versión de clases; de tal forma que, en vez de añadir lógica funcional a las clases, se extiende de las mismas y se tiene mayor libertad para extender el código; esto sin tener que realizar grandes modificaciones en la estructura del sistema.
- d) Aplicabilidad: Es utilizado en Antares debido a que la aplicación debe configurarse con varias familias de productos. Se desea proporcionar una colección de clases en la capa de presentación, sin tener que conocer sus implementaciones.
- e) Descripción: Se aplica a Antares, porque se desea que la aplicación sea lo suficientemente flexible como para que sea escalable sin tener que reescribir mucho código adicional, cuando se introduzcan nuevos módulos funcionales.
- f) Implementación:

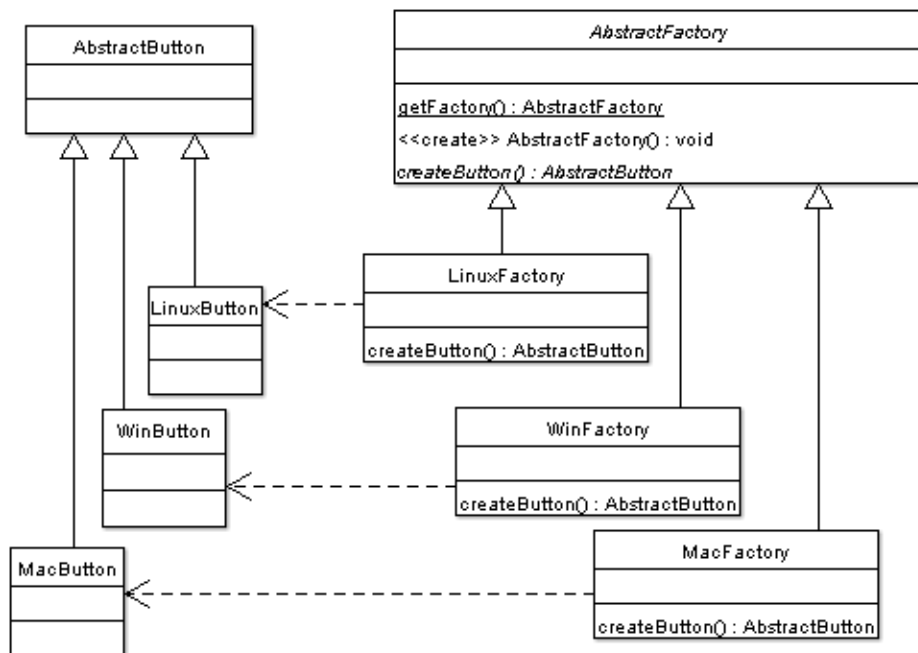


Fig. 1.5 Diagrama de Clases del Patrón Abstract Factory.

Elementos que se utilizan para implementar el patrón:

- **AbstractFactory:** Es la clase abstracta o interfaz que define los métodos de creación de los productos abstractos.
- **AbstractProduct:** Es la clase abstracta o interfaz que describe el comportamiento general de los recursos utilizados por la aplicación.

- ConcreteFactory: Clase derivada de la fábrica que implementa los métodos para crear uno o más productos concretos.
 - ConcreteProduct: Clase derivada del producto abstracto que proporciona una implementación para un recurso final específico.
- g) Ventajas e inconvenientes: Ayuda a incrementar la flexibilidad general de una aplicación, misma que se manifiesta tanto en tiempo de ejecución, como en tiempo de compilación. Durante el diseño no es necesario predecir todos los usos futuros de la aplicación. Simplifica la comprobación del resto de la aplicación. Esto se hace mediante clases de prueba
- h) Patrones relacionados: Factory Method.

1.4.3 PATRONES DE COMPORTAMIENTO

Este tipo de patrones se caracterizan por relacionarse con el flujo de control en un sistema. De aquí que ciertas formas de organizar el control en un sistema pueden traer muchos beneficios para la eficiencia y el funcionamiento del sistema.

En el desarrollo de Antares se emplearon algunos conceptos del patrón Chain of Responsibility, para la cuestión de atención a múltiples peticiones de clientes realizadas al servlet controlador de Antares. A continuación se presenta la definición del mismo.

PATRÓN CHAIN OF RESPONSABILITY

- a) Propiedades del patrón: Es de comportamiento a nivel componente.
- b) Propósito: Establecer una cadena de responsabilidades para una aplicación mediante un mensaje que se manejará en el momento de recibirse, se analizará y será redirigido a un objeto o código que pueda atenderlo.
- c) Introducción: Se referencia uno o más objetos centrales que almacenan las responsabilidades y las acciones.
- d) Aplicabilidad: Es utilizado en Antares como parte del controlador, para especificar criterios de atención a las diversas peticiones realizadas por los usuarios de la GCSR9. En general, se recomienda utilizar cuando haya un grupo de objetos en un sistema que puedan responder potencialmente al mismo tipo de mensajes.

- e) Descripción: El patrón consiste en una cadena de redirección de mensajes, de tal forma que si un objeto no puede manejar un determinado mensaje, lo pasa a algún otro objeto.
- f) Implementación:

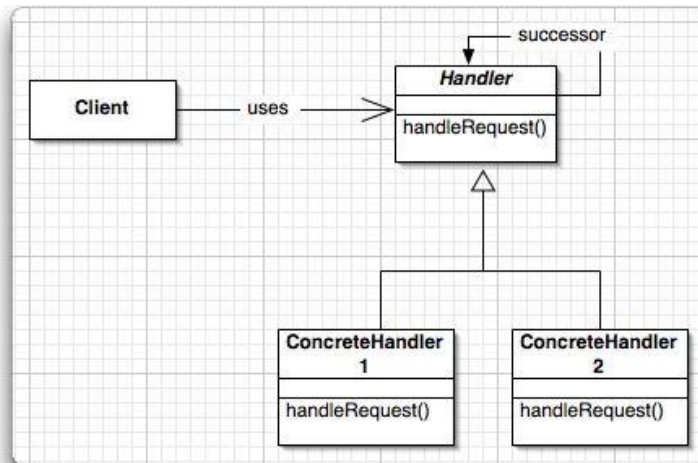


Fig. 1.6 Diagrama de Clases del Patrón Chain of Responsibility.

Elementos que se utilizan para implementar el patrón:

- Handler: Es la interfaz en la que se define el modelo utilizado para pasar un mensaje al siguiente manejador.
 - ConcreteHandler: Es la clase que implementa la interfaz Handler. Mantiene una referencia a la siguiente instancia de Handler. En Antares esta referencia es establecida mediante un método.
- g) Ventajas e inconvenientes: Ofrece gran flexibilidad para el procesamiento de eventos de una aplicación, tal es el caso de Antares, para la que provee atención en una misma vista a diversos tipos de eventos (Botones de búsqueda, actualización de información, navegación entre la aplicación). La facilidad se debe a que maneja la complejidad de la gestión de eventos dividiendo las responsabilidades entre elementos más simples. Como desventaja cabe mencionar el costo de desarrollo, comprobación y depuración de este patrón, debido a que conforme aumenta la complejidad en el redireccionamiento de la cadena hay que controlar si los eventos son enviados correctamente.
- h) Patrones relacionados: Composite.

1.4.4 PATRONES ESTRUCTURALES

Los patrones estructurales ayudan a dividir y/o combinar los elementos de una aplicación. Permiten que sistemas incompatibles entre sí, puedan comunicarse; así como simplificar la interfaz para un grupo de subsistemas o un sistema complejo.

1.4.5 PATRONES DE SISTEMA

Este tipo de patrones aplican su arquitectura en un nivel muy alto de abstracción. Pueden aplicarse tanto a procesos principales como entre aplicaciones.

Antares se construyó bajo las especificaciones del patrón Modelo-Vista-Controlador puesto que se buscaba desacoplar la presentación de la lógica del negocio de la GCSR9; de aquí la importancia de abordar este patrón de diseño.

PATRÓN MODELO-VISTA-CONTROLADOR

- a) Propiedades del patrón: Es de comportamiento a nivel componente o arquitectura.
- b) Propósito: Divide un componente o sistema en tres partes lógicas, el modelo, la vista y el controlador. Su finalidad es facilitar la modificación o personalización de cada parte desacoplando la lógica del negocio de la manera en que se presenta la información.
- c) Introducción: Para controlar la complejidad de las aplicaciones así como la previsión de sus cambios el patrón Model View Controller (MVC) plantea dividir las clases en tres partes funcionales. Una clase que representa la información del negocio, otra la representación visual y la tercera finalmente sería la encargada de controlar el flujo de información entre la interfaz gráfica y la información de negocio.
- d) Aplicabilidad: La arquitectura del sistema Antares se encuentra construida bajo las especificaciones del patrón MVC, esto porque se requiere que los componentes del producto sean flexibles y fáciles de mantener. En el contexto del sistema construido para la GCSR9, el patrón MVC permite que se gestionen las responsabilidades de recibir peticiones, a continuación ejecutar alguna lógica de negocio y luego determinar la siguiente vista a mostrar. Aunado a lo anterior, Antares cubre los requisitos en los que se hace la recomendación a utilizar el patrón MVC.
 - La representación interna del sistema es muy diferente a la representación en la pantalla.

- Existen diferentes tipos de comportamiento, por lo que múltiples fuentes pueden invocar el mismo comportamiento en el mismo componente.
 - Se tiene la capacidad de adaptar o reutilizar cierto componente bajo distintas circunstancias, con el mínimo de trabajo extra.
- e) Descripción: Este patrón define de manera elegante a un elemento complejo en términos de tres subunidades lógicas:
- Modelo: Representa el estado del elemento de negocio además de ocuparse de los cambios de estado del mismo, dicho de otra manera es el responsable de la lógica del negocio. Para el caso particular de Antares, el modelo es gestionado por servicios y la información del negocio adopta la forma de DTO (Data Transfer Objects) u Objetos de Transferencia de Datos, que son JavaBeans que son encargados de transferir información entre las capas de la arquitectura.
 - Vista: Es el responsable de la vista de representación de la lógica del negocio. Para Antares, las vistas constan de páginas HTML para mostrar contenido estático; así como de JSP's que se utilizan para presentar contenido tanto estático como dinámico.
 - Controlador: Responsable de controlar el flujo de la aplicación, responde a los eventos generados por la vista e invoca acciones pertinentes en el modelo y devuelve el control al usuario. En el contexto de Antares, el controlador del patrón MVC es un servlet y tiene las siguientes funciones:
 1. Intercepta peticiones HTTP desde un cliente.
 2. Traduce cada petición en una operación específica de negocio a llevarse a cabo
 3. Invoca la operación de negocio o la delega a un manejador.
 4. Ayuda a seleccionar la siguiente vista a mostrar al cliente.
 5. Devuelve la vista al cliente.
- Descomponiendo los componentes en esta forma, cada parte puede ser tratada independientemente de las otras dos. Para que el componente se comporte como un todo, cada parte deberá tener una interfaz hacia las otras dos.
- f) Implementación: Se utiliza un diagrama de componentes para describir este patrón. Cada una de las tres partes del patrón MVC constituye un componente, el cual a su vez, consta de varias clases e interfaces java.

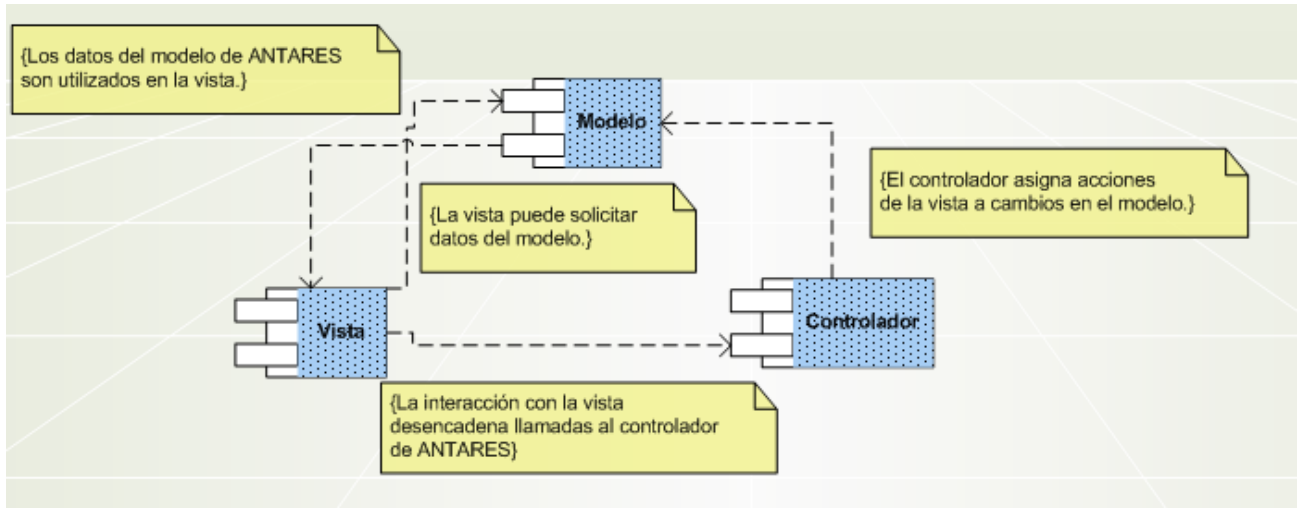


Fig. 1.7 Diagrama de Componentes del Patrón Model View Controller.

- g) Ventajas e inconvenientes: Ofrece buen nivel de encapsulamiento. Proporciona una forma excelente de hacer que un componente sea flexible y adaptable a distintas situaciones. La flexibilidad puede ser utilizada de forma estática y dinámica.
- h) Patrones relacionados: Observer y Strategy.

1.5 MARCOS DE TRABAJO

Un marco de trabajo o “framework” es un conjunto de clases e interfaces que cooperan entre sí, para solucionar un tipo específico de problema de software. Tiene las siguientes características:

- Consta de varias clases o componentes, cada una de las cuales puede proporcionar una abstracción de algún concepto determinado.
- Define la manera en la que las abstracciones trabajan conjuntamente para solucionar un problema.
- Sus componentes son reutilizables.
- Organiza patrones de diseño.

Como tal un framework proporciona un comportamiento genérico que pueden utilizar múltiples aplicaciones, de tal manera que aplicación amplía la funcionalidad provista por el framework, para cumplir con las tareas requeridas.

1.5.2 FRAMEWORK STRUTS

Struts fue creado por Craig R. McClanahan y donado a la ASF (Fundación de Software Apache) en el año 2000. A partir de ahí muchos desarrolladores expertos se han sumado al proyecto para mejorar el framework y ampliar su valor. El framework Struts está basado principalmente en la tecnología Java Servlet y en menor grado en Java Server Pages, situación que lo hace dependiente de un contenedor Web.

El motivo por el cual Antares utiliza en su arquitectura el framework Struts, se debe principalmente a que permite dividirla en términos de capas, esto con la finalidad de cumplir con los objetivos esenciales del estándar J2EE como los son, el separar la funcionalidad de la aplicación en niveles, para proporcionar una separación de responsabilidad, reutilización de código y escalabilidad.

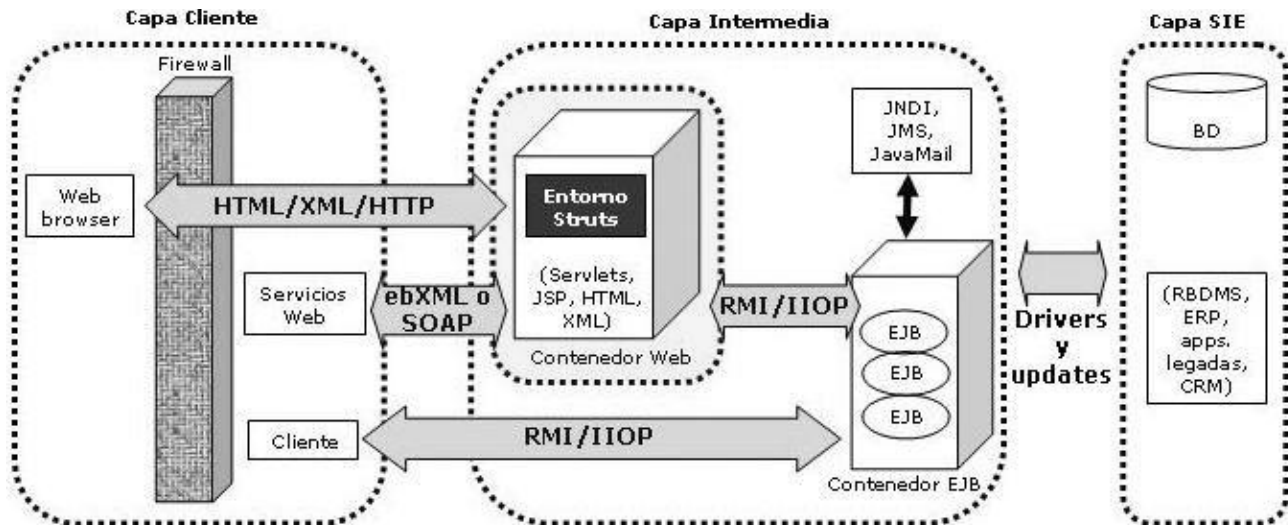


Fig. 1.8 Niveles funcionales de aplicaciones construidas con Struts.

DEFINICIÓN DE CONTENEDOR

Un contenedor es un mecanismo de software que proporciona un entorno de almacenamiento desde el que es posible la ejecución de componentes de software. Los contenedores proporcionan servicios generales que pueden utilizar los componentes dentro del entorno y que son de gran utilidad para el desarrollo de aplicaciones. El caso particular del sistema Antares maneja un contenedor Web, el cual permite que se desarrollen y ejecuten tanto Servlets, JSP's y clases java dentro del contenedor. El contenedor provee el servicio de JNDI (Java Naming and Directory

Interface), así como servicios de transacción (tratados en el punto 1.3.4), para la gestión de recursos como servidores de bases de datos y que se pueden configurar desde el contenedor.

COMPONENTES DE STRUTS

El framework de Struts se compone de más de 300 clases java y se divide en 8 paquetes principales:

- Action: Contiene las clases del controlador. (Utilizado por Antares).
- Actions: Contiene las clases Action que son susceptibles de ser ampliadas. (Utilizado por Antares).
- Config: Incluye las clases de configuración en tiempo de ejecución del archivo de configuración de Struts.
- Taglib: Contiene las clases del manejador de etiquetas para las librerías de Struts. (Utilizado por Antares).
- Tiles: Clases utilizadas para referenciar documentos.
- Upload: Clases utilizadas para enviar y descargar archivos desde el sistema local. Para Antares, incluyó las clases necesarias para gestionar archivos de forma local, en un sistema operativo Windows 2003 Server. (Utilizado por Antares).
- Util: Contiene clases de apoyo y utilidad para el desarrollo de aplicaciones. (Utilizado por Antares).
- Validator: Clases que permiten hacer validaciones para el manejo del flujo de información entre las capas de Struts.

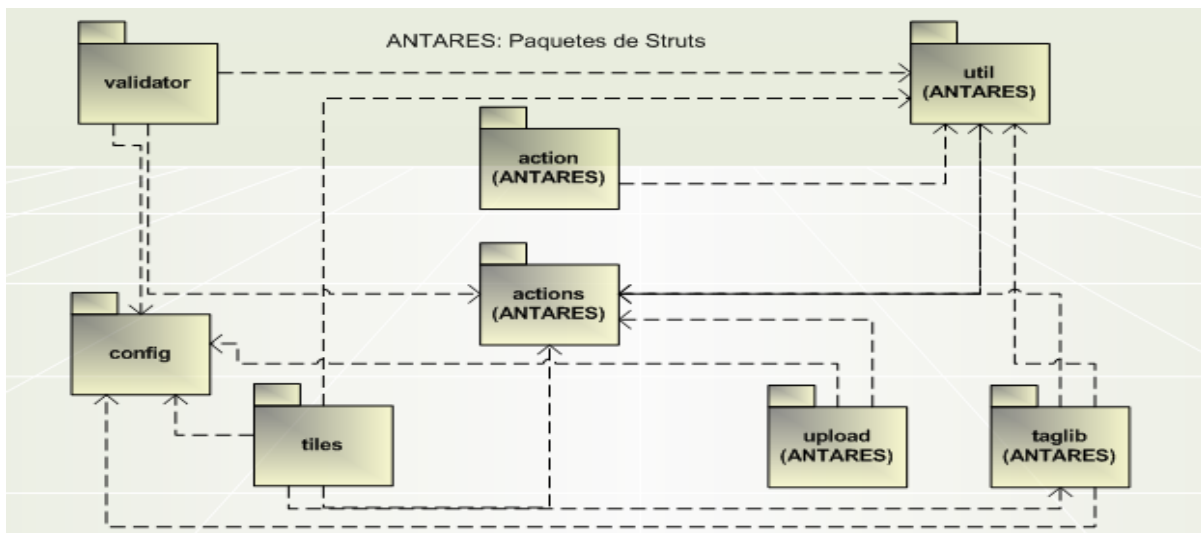


Fig. 1.9 Paquetes de nivel superior de Struts.

CONTROLADOR DE STRUTS PARA ANTARES

El componente de controlador en una aplicación MVC tiene varias responsabilidades, incluidas recibir peticiones de los clientes, invocar operaciones de negocio y coordinar la vista a devolver al cliente. En Antares que es una aplicación creada utilizando Struts, el controlador es un java Servlet, cuya labor principal es mapear las acciones de los usuarios de la GCSR9, en operaciones de negocio sobre las Cuentas Corporativas para luego seleccionar la vista a devolver, esto basándose en la petición e información de estado.

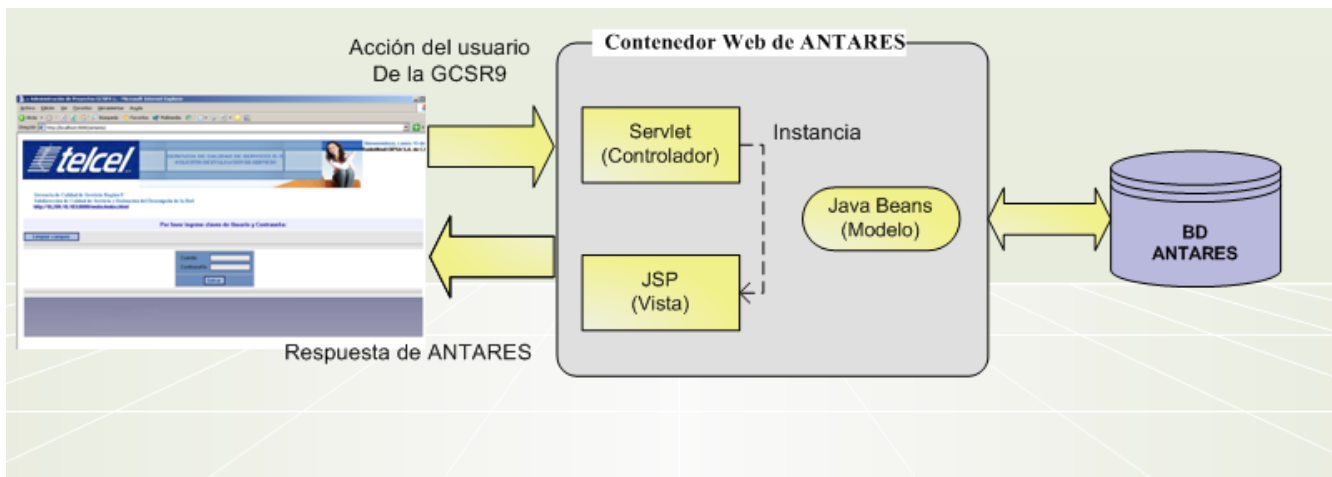


Fig. 1.10 El Framework de Struts en la Arquitectura de Antares.

Las clases que proporciona Struts para la capa del Controlador:

- **ActionServlet:** Es la clase que extiende `javax.servlet.http.HttpServlet` responsable del empaquetado y enrutado del tráfico HTTP hacia el manejador apropiado dentro del entorno (framework). Esta clase no es abstracta, por lo que se puede utilizar directamente en cualquier aplicación.
- **RequestProcessor:** Esta clase permite desacoplar el proceso de petición (request process) del `ActionServlet` y así poder modificar cómo se procesa la petición (haciendo un subclase de `RequestProcessor`).
- **Action:** Es la clase que independiza la petición del cliente del modelo de negocio. Es una extensión del componente de control (capa C) y permite llevar a cabo funciones como autorización, logging, o validación de sesión, antes de invocar la lógica de negocio. Su método más importante es:


```
public ActionForward execute(ActionMapping mapping, HttpServletRequest request,
    HttpServletResponse response) throws Exception;
```

- **ActionMapping:** Clase que representa una acción de mapeado que se define en el fichero de configuración de Struts. Indica al controlador que instancia de Action se debe invocar en cada petición.
- **ActionForward:** Clase que representa el destino al cual el controlador debe enviar el control una vez que una Action se ha completado.

MODELO DE STRUTS PARA ANTARES

El modelo de Antares, está conformado por servicios que se comunican con la capa de presentación y con la capa de acceso a datos, a través de los DTO (Objetos de transferencia de datos). Con esto logramos uno de los objetivos del patrón MVC, que es mantener desacoplados la capa de negocio de la presentación.

Al utilizar los DTO, también conocido como VO's (Objetos de Valor) se están representando objetos de negocio remotos, mediante java beans que contienen los datos necesarios, pero que no contemplan lógica de negocio alguna. También son utilizados para actualizar el objeto de negocio, ya que funcionan como una foto instantánea del objeto remoto en un momento determinado. Otras ventajas por las que son utilizados por Antares consisten en que ayudan a reducir el ancho de banda de red; así como mejorar el tiempo de respuesta, facilitando el mantenimiento de la aplicación.

VISTA DE STRUTS PARA ANTARES

Los componentes de la vista empleados para el sistema Antares son los siguientes:

- HTML
- DTO o Value Objects (Objetos de Transferencia de Datos).
- ActionForms, clases que permiten el flujo de datos de entrada hacia delante y hacia atrás entre el usuario y la capa de negocio. El framework recopila automáticamente la entrada de datos de la petición y pasa esos datos a una clase Action utilizando un form bean, el cual puede ser pasado posteriormente a la capa de negocio.
- JavaServer Pages: Conforman la mayoría de las vistas que deben mostrarse a los usuarios de la GCSR9. Combinadas con librerías de etiquetas, hojas de estilo, código java Script y HTML e imágenes constituyen la capa de la vista de Antares.
- Struts Tags (librería de etiquetas)

- Paquete de recursos Java

1.5.3 FRAMEWORK SPRING

Spring es un framework Java/J2EE desarrollado usando licencia de OpenSource y tiene como directriz en su configuración, el uso de los JavaBeans. Se caracteriza por ser fácilmente ampliable, así como sumamente poderoso en la gestión de clases y componentes, aplica conceptos de la “Inversión de Control”, propio de la programación orientada a aspectos (AOP).

De acuerdo con Chuck Cavaness, *“Spring es un framework de aplicación J2EE desarrollado por Rod jonson. Es un framework Web, pero sus características ofrecen la posibilidad de integrarse a otros componentes y marcos de trabajo”* [6].

PRINCIPIO DE INVERSIÓN DE CONTROL (IoC)

Spring se basa en IoC. IoC que es lo que se conoce como “Principio de Inversión de Dependencia”, Inversion of Control" (IoC) y consiste en:

- Un Contenedor encargado de gestionar los objetos, deslindando al programador de esta labor.
- Controla la creación de estos objetos; por decirlo de alguna manera, el contenedor hace los “new” de las clases java, ahorrando esta labor al programador.
- El contenedor resuelve dependencias entre los objetos que contiene.
- Estos puntos son suficientes y necesarios para poder hablar de una definición básica de IoC. Spring proporciona un contenedor que maneja todo lo que se hace con los objetos del IoC. Debido a la naturaleza del IoC, el contenedor define el ciclo de vida de los objetos.
- Finalmente, el contenedor resuelve las dependencias entre los servicios que él controla.

Toda esta funcionalidad puede usarse en cualquier servidor J2EE, y la mayoría de ella ni siquiera requiere su uso. El objetivo central de Spring es permitir que objetos de negocio y de acceso a datos sean reutilizables, no atados a servicios J2EE específicos.

Se le llama inversión de control porque el mecanismo responsable de un proceso o actividad delega el control o la responsabilidad a otro. En este caso, el mecanismo de creación es el que se está invirtiendo, es decir que el mecanismo responsable de instanciar y referenciar el objeto, es externo a la clase que hace la declaración. En consecuencia se tiene nueva funcionalidad

totalmente desacoplada. Las clases a las cuales se les introduzca nueva funcionalidad mediante AOP de Spring sólo podrán ser inicializadas mediante IoC de Spring, ya que de otra forma los nuevos métodos y atributos no serán introducidos.

APLICACIÓN A ANTARES

- La principal razón, por la que el framework de Spring es aplicado a Antares, es separar o desacoplar los accesos a la capa de datos, de los aspectos relacionados con las transacciones de la capa de negocio; de tal manera que los objetos de la capa de negocio no dependen de ninguna estrategia particular de clases DAOs.
- Funge como una potente herramienta para manejar objetos de Factoría, ya que proporciona un ligero contenedor de beans que aloja a los objetos de las clases de negocio, DAOs y sesiones con los frameworks Hibernate 3 y Struts 1.2. Esto se hace mediante la configuración de un archivo XML, en el que se define el contexto de la aplicación.
- Aplica los principios de Inversión de Control (IoC), lo cual permite que la configuración de Antares sea más rápida, pues solo se deben definir los beans en el contexto de aplicación.
- Las aplicaciones configuradas para usar beans mediante Spring no necesitan depender de interfaces o clases propias de Spring, solo basta con obtener su configuración a partir de las propiedades de sus beans. Este concepto se aplica a Antares, pero también es factible para todo tipo de aplicaciones Java.
- Para la capa DAO implementada con los servicios JDBC para Antares, ofrece un a gama muy completa para el manejo de excepciones SQL, con lo que se simplifica el manejo y la solución de errores surgidos durante la codificación de casos de uso.
- Funcionalidad AOP, integrada en la gestión de configuración de Spring. Se puede aplicar AOP a cualquier objeto gestionado por Spring, añadiendo aspectos como gestión de transacciones declarativa. Con Spring se puede tener gestión de transacciones declarativa sin EJB, incluso sin JTA, si se utiliza una única base de datos en un contenedor Web sin soporte JTA.

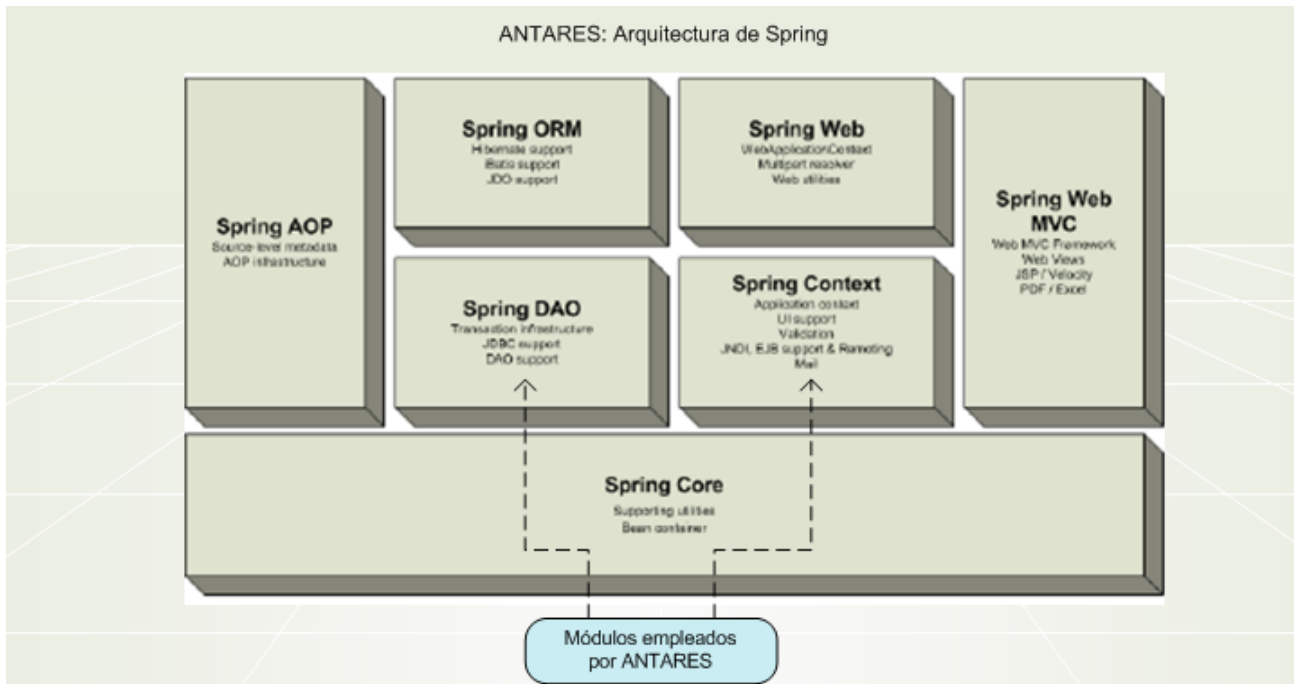


Fig. 1.11 Módulos de la Arquitectura de Spring utilizados por Antares.

1.5.4 FRAMEWORK HIBERNATE

Hibernate es un framework que provee un mecanismo de persistencia relacional para Java a partir de la base de datos Antares. En nuestro contexto representa la capa intermedia que comunica las clases de negocio con las clases DAOs.

Permite desacoplar el código SQL para los accesos a la base de datos de Antares, de los servicios (clases java encargadas de implementar la lógica del negocio). De esta forma, mediante objetos java, se realizan las peticiones de acceso y gestión a la base de datos; mientras que Hibernate es el encargado de ejecutar el código SQL; así como manejar las transacciones y control de excepciones.

IMPLANTACIÓN EN ANTARES

La manera tradicional de acceder sería a través de JDBC directamente conectado a la base de datos Antares mediante ejecuciones de sentencias SQL. Esta solución es factible en arquitecturas sin clases de negocio, sin embargo el mantenimiento del código está altamente ligado a los cambios en el modelo de datos relacional de la BD, lo que implica que un mínimo cambio derive en una revisión de casi todo el código así como su compilación y nuevo despliegue en servidor.

En virtud de lo anterior, Hibernate constituye la alternativa a Antares, sistema que alberga gran cantidad de clases, y le permite que el acceso a través de SQL directas, pueda ser utilizado de manera puntual para realizar operaciones a través del lenguaje SQL lo cual es mucho más efectivo que la carga de gran cantidad de objetos en memoria.

La aplicación de este framework en la arquitectura de Antares se apoya en la creación de clases de acceso a datos (DAO Data Acces Object). De esta manera nuestra capa de negocio interactúa con la capa DAO y siendo esta última la encargada de realizar las operaciones sobre la BD Antares. Por lo tanto Hibernate es el puente entre la aplicación y la BD Antares, y sus funciones van desde la ejecución de sentencias SQL a través de JDBC hasta la creación, modificación y eliminación de objetos persistentes.

Finalmente es de suma importancia mencionar que, la utilización de este framework conllevará un proceso de desarrollo más costoso en lo que respecta a la adecuación de la capa de persistencia de la arquitectura propia de Antares.

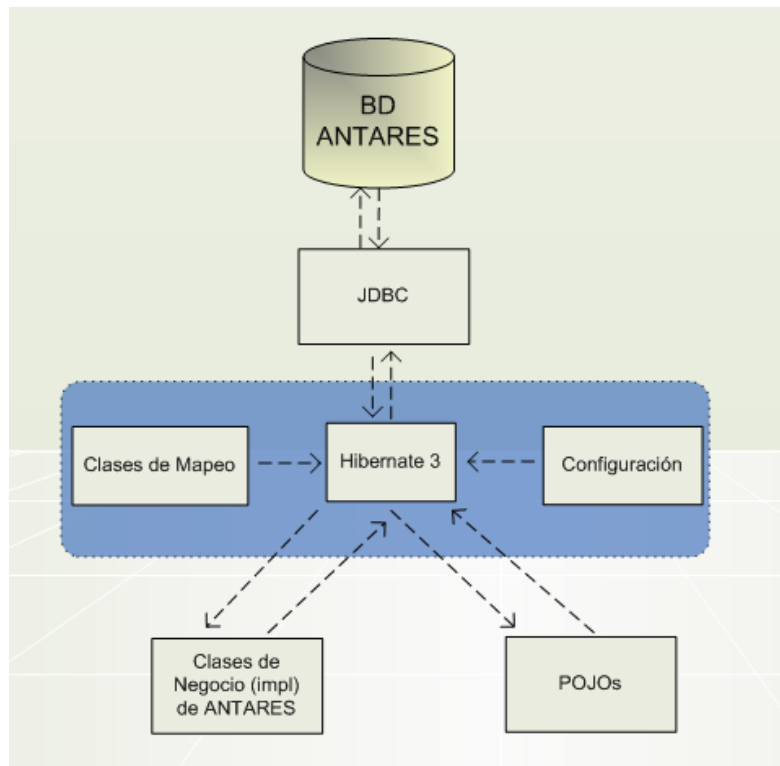


Fig. 1.12 Hibernate como capa de Persistencia en la Arquitectura de Antares.

COMPONENTES DE HIBERNATE

Los componentes de Hibernate necesarios durante el desarrollo de Antares, son los siguientes:

- 1) Tablas propias del Diagrama Entidad Relación de Antares.
- 2) POJOs o Clases JavaBean que representa las Entidades de Antares como objetos java.
- 3) Archivos de mapeo que le permiten a Hibernate saber las características del Bean que se mapean a campos de las Entidades de Antares.
- 4) Archivo de Configuración de Hibernate “hibernate.cfg.xml” el cual le permite a Hibernate conocer la configuración JDBC para acceder a la base de datos Antares.

CAPÍTULO II: FASE DE INICIO

El proyecto de desarrollo de software Antares comienza con la captura de requisitos de los usuarios de la GCSR9. El objetivo de esta actividad es comenzar a guiar el desarrollo hacia el sistema correcto, comprendiendo las verdaderas necesidades de los usuarios, que se traducirán en las condiciones o capacidades que Antares deberá cumplir.

Una vez identificados, se hará una detallada descripción de los principales requisitos del sistema, en la que sean incluidos todos los detalles sobre el funcionamiento esperado por Antares. Esta descripción se hace en el lenguaje del cliente, es decir en el lenguaje del negocio de la GCSR9, ya que es vital conocer a la perfección el contexto en el que operará el sistema. Como se mencionó en el capítulo anterior, el Proceso de desarrollo de Antares estará regido por RUP, por lo que los requerimientos funcionales pasarán a formar los casos de uso.

2.1 REQUISITOS FUNCIONALES DE ANTARES

2.1.1 PROBLEMÁTICA A RESOLVER

La GCSR9 ha realizado un análisis minucioso acerca de las tareas que desempeñan sus integrantes, relacionadas con el seguimiento de los proyectos; y al respecto concluyen que los proyectos son susceptibles a extenderse en tiempo y costo, esto debido al manejo inadecuado de la información, tal como los errores provocados por no contar con un estándar de intercambio de información, o la desactualización de archivos compartidos e incluso la presencia de archivos corruptos. Estas situaciones en definitiva, ponen en riesgo la información de la empresa pues disminuyen la posibilidad de garantizar la veracidad, integridad y disponibilidad en tiempo real de la misma.

El desarrollo del sistema Antares surge en respuesta a las necesidades operativas de la GCSR9 y tiene como directriz, el mantener íntegra, segura y accesible, la información de la empresa generada a partir de la atención y seguimiento de sus proyectos.

Un área más de oportunidad es, liberar en gran parte la presencia de los recursos de la GCSR9 asignados para el seguimiento y atención de los proyectos, para fines de reportar tanto el status de los proyectos que tienen a su cargo, hasta exponer a detalle las tareas relacionadas con un proyecto en particular. Con base en lo anterior, se debe contar con una herramienta que permita mantener informados a jefes y altos directivos, de los avances que van surgiendo derivados de la atención a los proyectos.

2.1.2 REQUISITOS PARA ANTARES

Los requisitos establecidos dentro del alcance del desarrollo del sistema Antares son los siguientes:

- 1) El sistema a través de interfaces Web utilizará Internet para la Gestión de proyectos surgidos como parte del proceso de Atención a Cuentas Corporativas de Radio Móvil DIPSA (Telcel).
- 2) El sistema deberá centralizar y controlar la información generada de la atención y seguimiento de los proyectos de Atención a Cuentas Corporativas. El repositorio de datos de Antares almacenará información de las diversas áreas de la empresa que intervienen en la operación diaria:
 - a. Mesa de Ayuda.
 - b. Ventas.
 - c. Centrales MSC.
 - d. Troncales Celulares.
 - e. Radiobases BSC.
 - f. Nuevas Tecnologías.
 - g. Mantenimiento RAN.
- 3) La arquitectura de Antares deberá permitir el crecimiento paulatino del sistema, de tal manera que reduzca el retrabajo y el tiempo de desarrollo en la inclusión de nuevos módulos de funcionalidades.
- 4) El sistema a implementar deberá ser utilizado y administrado vía WEB. En lo que respecta a la gestión de los proyectos de la GCSR9, permitirá la consulta en línea de la información referente al proyecto, así como el seguimiento durante cada etapa del mismo.
- 5) Proveer de una herramienta a los usuarios (Analistas, Jefes de área y personal general de la empresa) que permita la consulta de estatus y el seguimiento de los proyectos; así como automatizar las notificaciones y actualizaciones más trascendentes durante la atención de cada proyecto de la GCSR9. Las tareas principales que debe cubrir son las siguientes:

- a. Autenticar usuarios del sistema y proveerles recursos y funcionalidades que el rol de cada uno amerita.
- b. Generar una Solicitud de Servicio para atención de una Cuenta Corporativa en particular: El usuario ingresará mediante un formulario, información que detalle tanto las necesidades de servicio requerido por parte de la compañía, como información referente tecnología empleada en las instalaciones. Al capturar la solicitud, el sistema deberá asignarle un folio al usuario, con el que podrá darle seguimiento durante el tiempo de solución del mismo. Así mismo se deberá notificar al usuario Administrados por medio de un correo electrónico, sobre la creación del nuevo requerimiento.
- c. Consulta de estatus de los requerimientos: Los diferentes tipos de usuarios (Empleado General de Telcel, Gerente de Calidad de Servicio Región 9, Jefe de Optimización de Red Región 9 y Analista de Optimización de Red Región 9) podrán consultar en todo momento el estatus en el que se encuentra un Requerimiento específico; solo basta con que cuenten con el número de Folio asignado. Aunado a esto, el sistema debe permitir consultar los Requerimientos asignados por cada Ingeniero responsable de dar solución (Analista de Optimización de Red Región 9).
- d. El sistema debe permitirle al tipo de usuario Administrador, asignar cada requerimiento recibido a un ingeniero con puesto “Analista de Optimización de Red Región 9”, quien será el responsable de dar atención y solución al mismo. Cuando el requerimiento sea asignado, el sistema deberá notificar por medio de un correo electrónico, sobre la asignación al usuario responsable.
- e. Seguimiento de Requerimientos: El sistema deberá permitirle a los usuarios, adjuntar información que contribuya a la solución del requerimiento de servicio, tal como planes de trabajo, minutas, acuerdos, así como cualquier tipo de documento que funja como evidencia en la atención a la Cuenta Corporativa; así como la captura de comentarios. Particularmente al usuario con puesto Analista de Optimización de Red Región 9, deberá permitírsele el cambio de estatus del requerimiento, dependiendo del avance que se tenga en la solución del mismo. A este respecto, los estatus de un requerimiento son los siguientes:
 - Creado
 - Asignado
 - En proceso por Área Comercial
 - En proceso por Área Jurídica
 - En proceso por CS Corporativo (REG 1-8)

- En proceso por Diseño de Radio Frecuencia R9
 - En proceso por la Gerencia de Calidad de Servicio R9
 - En espera de proyecto
 - En observación
 - No se encontró falla
 - Cancelado
 - Solucionado OK
 - Cerrado
- f. Reportes: El sistema debe permitir la consulta de reportes que permitan una mejor planeación de la atención de los proyectos, así como un mejor balance en la carga de trabajo por cada departamento. En los reportes son los siguientes:
- Número de Requerimientos atendidos por cada usuario
 - Número de Requerimientos por cada estatus
 - Requerimientos asignados/atendidos por departamento
- g. Gestión de usuarios: El sistema debe permitir actualizar información de los usuarios como su perfil o cambio de contraseña, así como generar usuarios y eliminarlos permanentemente del sistema.
- 6) Mantener el sistema Antares en un ambiente de producción fiable, robusto y estable, que permita la disponibilidad de la información de los proyectos de la GCSR9 en todo momento.
- 7) Deberá controlar el acceso y privilegios sobre la funcionalidad del sistema a los siguientes tipos de usuarios:
- a. Empleado General de Telcel.
 - b. Gerente de Calidad de Servicio Región 9.
 - c. Jefe de Optimización de Red Región 9.
 - d. Analista de Optimización de Red Región 9.
 - e. Administrador.

2.1.3 MODELO DE DOMINIO

A continuación se presentan los objetos de dominio obtenidos a partir de las entrevistas con los usuarios de la GCSR9. Los objetos de dominio representan las cosas que existen o los eventos que suceden en el entorno en el que trabaja el sistema, para posteriormente dar lugar a las clases de que se constituirá el sistema Antares. Su objetivo principal es mostrar a todos los miembros de

los equipos de trabajo (Radio Móvil DIPSA y Área Provedora), las clases principales así como la relación y asociación que existen entre ellas.

CLASES DE DOMINIO DE ANTARES

- Cuenta Corporativa: Persona moral (Empresa o Corporativo) que celebra un contrato con Radio Móvil DIPSA en el que esta última, le provee servicios de telecomunicaciones así como la renta de equipo de infraestructura. Son los clientes finales de la GCSR9.
- Área Solicitante: Es el departamento de Radio Móvil DIPSA que representa a la Cuenta Corporativa ante la GCSR9, para el seguimiento de los proyectos.
- Folio o Requerimiento: Es una solicitud de atención de Servicio realizada por parte de una cuenta corporativa y que se ha convertido en proyecto para ser gestionado mediante Antares.
- Bitácora: Es el molde a partir del cual se registran las transacciones realizadas por los usuarios de Antares.
- Usuario: Un usuario se define por su rol que desempeña en la GCSR9 y es a partir del rol, que se definen los privilegios que el usuario tendrá sobre Antares.
- Mail: Un correo electrónico es el medio informativo acerca de las transacciones realizadas por los usuarios de Antares, durante la gestión de un proyecto de atención a cuenta corporativa.
- Archivo Evidencia: Es un archivo que se adjunta al proyecto y que funge como evidencia durante el seguimiento del proyecto.

2.2 ACUERDOS DE ADMINISTRACIÓN DEL PROYECTO

2.2.1 INTROUCCIÓN

Propósito del documento:	Establecer los términos y condiciones base de referencia para una administración efectiva del proyecto Antares.
Acuerdo de Confidencialidad:	La información contenida en este documento no deberá ser divulgada fuera de la Gerencia de Calidad de Servicio Región 9, adscrita la Dirección de Ingeniería de Red de Radio Móvil DIPSA. La información no será duplicada o dada a conocer, parcial o totalmente, sin la autorización de Radio Móvil DIPSA.
Alcance y vigencia del Acuerdo de Administración:	<p>Alcance:</p> <p>Los acuerdos y procedimientos establecidos en el presente documento aplicarán a la administración del proyecto Antares de la GCSR9 y durante el desarrollo del mismo.</p> <p>Control de Cambios:</p> <p>Los cambios entrarán en vigencia a la fecha de emisión de la versión que los incorpora, siempre y cuando hayan sido aprobados por las mismas personas que aprobaron el documento original.</p>

2.2.2 DESCRIPCIÓN DEL PROYECTO

Propósito del Proyecto:	De acuerdo a las metas operativas de la Dirección de Ingeniería de Red, el objetivo que persigue la GCSR9 a través de este sistema, es el de mejorar los mecanismos que permitan optimizar el control de proyectos asignados a cuentas corporativas con quienes se mantiene algún contrato de servicio vigente; así como también, exponer de manera oportuna los resultados de productividad de cada departamento de la GCSR9.
-------------------------	--

Alcance:

Para el sistema a desarrollar se considera como parte de su propuesta técnica de solución, los siguientes alcances que de manera general se enlistan, para satisfacer el módulo de Cuentas Corporativas:

- Solicitud de Atención a Cuenta Corporativa
 - Consulta Estatus de Requerimiento
 - Seguimiento de Requerimientos
 - Administración de Usuarios
 - Reportes
-

Exclusiones específicas:

Para la siguiente versión de desarrollo, el sistema ANATARES deberá gestionar las tareas derivadas de los procesos de la GCSR9:

- Fallas Puntuales
 - Proyecto Q de datos
 - Administración de viáticos y vacaciones
-

Necesidades tecnológicas:

El sistema Antares deberá operar bajo la siguiente plataforma tecnológica:

SOFTWARE DE SERVIDORES:

Sistema Operativo: Windows 2003 Server SP2

Base de datos: MySQL Server 4.1, con facilidad de migración a SQL Server 2005.

Servidor de Aplicaciones: Tomcat 5.5

SOFTWARE DE CLIENTE O DEL USUARIO FINAL:

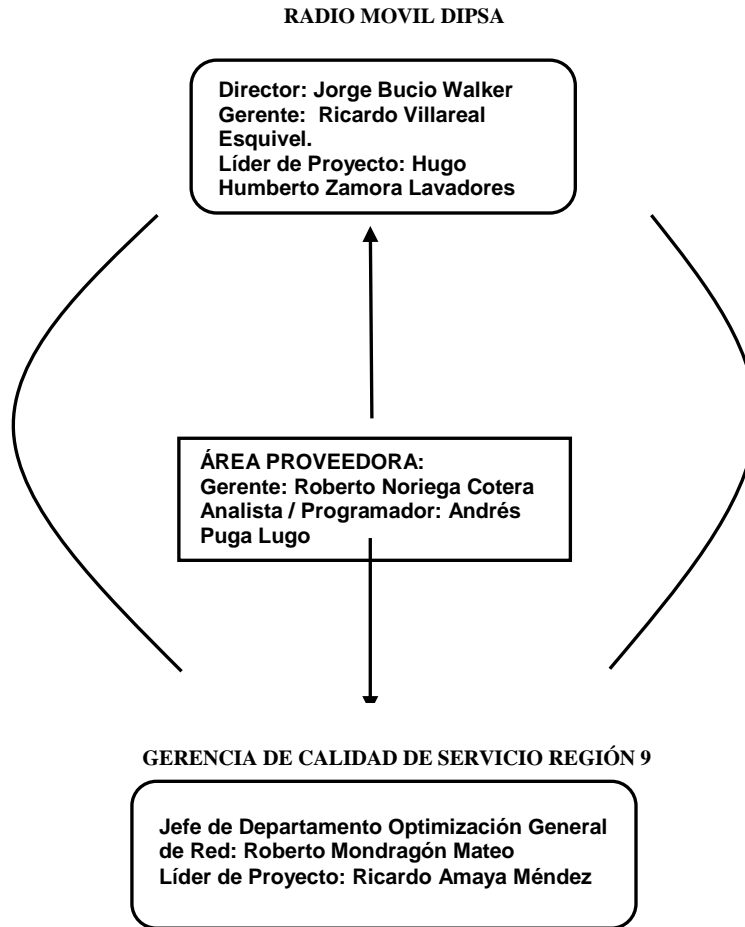
Requerimientos mínimos del equipo: Procesador Pentium 4 o superior, a 1.8 Ghz, 512 MB en RAM.

Navegador: Internet: Microsoft Internet Explorer 6.0.2 o superior.

2.2.3 ORGANIZACIÓN Y COMUNICACIÓN

Organigrama
Funcional :

El siguiente diagrama muestra la organización y líneas de autoridad de las distintas funciones establecidas en los grupos de trabajo:



ÁREA PROVEEDORA:

Nombre	Puesto
Roberto Noriega Cotera	Gerente de área proveedora.
Andrés Puga Lugo	Líder de Proyecto, Analista Programador.

RADIO MÓVIL DIPSA / DIRECCIÓN DE INGENIERÍA DE RED:

Nombre	Puesto
Jorge Bucio Walker	Director de Ingeniería de Red
Ricardo Villareal Esquivel	Gerente de Calidad de Servicio Región 9
Hugo Humberto Zamora	Líder de Proyecto

Lavadores	
-----------	--

GERENCIA DE CALIDAD DE SERVICIO REGIÓN 9:	
Nombre	Puesto
Roberto Mondragón Mateo	Jefe de Departamento de Optimización General de Red.
Ricardo Amaya Méndez	Líder de Proyecto

Sistema de comunicación:

Comunicación Formal:

Esta se va a llevar a cabo por medio de juntas o reuniones, en estas participaran los gerentes y líderes de cada equipo de trabajo, según sea el caso y el objetivo.

Cada vez que se realice una junta o reunión, será necesario generar una minuta de la misma y esta deberá ser firmada por los participantes.

La periodicidad con que se realizarán las juntas o reuniones dependerá de los requerimientos de cada proyecto y se llevará a efecto a petición de los involucrados en los mismos.

Comunicación Informal:

Se podrá dar este tipo de comunicación en cualquier momento, sin embargo, se deberá generar una minuta que refleje los acuerdos y/o compromisos por parte de los participantes.

2.2.4 RESPONSABILIDADES DE LOS GRUPOS DE TRABAJO

Área Provedora:

Descripción de Funciones A continuación se describen los roles y funciones de los integrantes del equipo de trabajo del Área Provedora.

Gerente de

Proyecto:

Roberto Noriega
Cotera

Funciones y tareas con respecto a la calidad :

- Administrar el Alcance del proyecto Antares.
- Acordar con la GCSR9 los procedimientos a seguir para los cambios al alcance del proyecto y el seguimiento al avance.
- Evaluar los cambios al contrato.
- Efectuar la evaluación de la satisfacción del cliente al término del contrato.

Administrar Trabajo:

- Dar seguimiento a las quejas de la GCSR9.
- Atender los requerimientos del personal del área proveedora asignado al equipo de trabajo.
- Dar seguimiento a las acciones y compromisos que se establecieron en el plan de trabajo.

Administrar Personal:

- Establecer los perfiles esperados de los recursos humanos requeridos para el proyecto y, si es necesario, establecer y dar seguimiento al plan de capacitación.
- Solicitar los recursos humanos requeridos al área proveedora.
- Asignar los recursos humanos requeridos.
- Evaluar el desempeño de los integrantes del equipo al terminar el proyecto.
- Notificar la disponibilidad de los miembros del equipo.

Administrar Organización y Comunicación:

- Identificar la estructura de los grupos y personas involucradas en el
-

proyecto.

- Establecer los mecanismos de comunicación y asegurar que la información este disponible para los integrantes del proyecto.

Administrar Recursos y Servicios:

- Establecer y dar seguimiento a un plan de adquisiciones en caso requerido.
 - Solicitar y revisar a la recepción los recursos y servicios requeridos para el proyecto Antares.
 - Registrar y controlar los activos de la GCSR9 y del área proveedora que se tienen bajo custodia.
 - Asegurarse de que las condiciones del medio ambiente sean las adecuadas para el desarrollo del proyecto.
-

Líder de Proyecto,
Analista y
Programador.:
Andrés Puga Lugo

Funciones y tareas con respecto a la calidad :

- Administrar el Alcance de Antares.
- Acordar con la GCSR9 los procedimientos a seguir para los cambios al alcance del proyecto y el seguimiento al avance.
- Efectuar la evaluación de la satisfacción del cliente al término del contrato.

Administrar Trabajo:

- Definir el plan de trabajo para cada fase del proyecto Antares.
- Dar seguimiento a las acciones y compromisos que se establecieron en el plan de trabajo.
- Identificar las desviaciones con respecto al plan y definir acciones correctivas.
- Dar seguimiento a las quejas de la GCSR9.

Administrar Organización y Comunicación:

- Identificar la estructura de los grupos y personas involucradas en el proyecto Antares.
 - Establecer los mecanismos de comunicación y asegurar que la información este disponible para los integrantes del proyecto.
-

Administrar Recursos y Servicios:

- Establecer y dar seguimiento a un plan de adquisiciones en caso requerido.
- Solicitar y revisar a la recepción los recursos y servicios requeridos para el proyecto.
- Registrar y controlar los activos de la GCSR9 y del área proveedora que se tienen bajo custodia.
- Asegurarse de que las condiciones del medio ambiente sean las adecuadas para el desarrollo del proyecto.

Administrar Configuración de Productos:

- Controlar los productos de trabajo para que sean identificados, controlados y se encuentren disponibles.
- Establecer el control de cambios de los productos de trabajo bajo configuración.
- Establecer mecanismos de almacenamiento, respaldo y seguridad de los productos de trabajo durante el proceso y hasta la entrega en las condiciones contractuales.

Actividades del Proceso de Desarrollo:

- Fase de Inicio: Realizar documentos entregables derivados del análisis y diseño de Antares:
 - 1) Requisitos Funcionales de Antares
 - 2) Modelo de Análisis
 - 3) Modelo de Diseño
 - 4) Modelo de Casos de Uso
- Fase de Elaboración: Completar la documentación correspondiente al análisis y diseño de Antares, así como establecer la arquitectura y ambientes para la implementación:
 - 1) Completar los casos de uso
 - 2) Modelo de despliegue
 - 3) Establecer Arquitectura de Antares
- Fase de Construcción: Realizar la codificación y documentación

correspondiente a los flujos de trabajo de implementación y pruebas de Antares:

- 1) Modelo de implementación
 - 2) Implementación del código fuente (mediante clases java) de los componentes de Antares.
 - 3) Garantizar la correcta funcionalidad de los componentes de Antares.
 - 4) Mantener la integridad y disponibilidad de los componentes de Antares.
 - 5) Implementación del código fuente para las clases correspondientes a las pruebas unitarias, para los componentes de Antares.
- Fase de Transición: Realizar las tareas correspondientes a pruebas finales, así como establecer planes de capacitación:
 - 1) Realizar de integración de los componentes de Antares.
 - 2) Realizar planeación e implementación de pruebas integrales, de stress y de volumen para Antares.
 - 3) Elaborar manual de usuario, así como un manual técnico con las especificaciones del ambiente en el que opera Antares.
-

Radio Móvil DIPSA: Dirección de Ingeniería de Red.

Descripción de Funciones A continuación se describen los roles y funciones de los integrantes del equipo de trabajo de la Dirección de Ingeniería de Red.

Gerente de Proyecto: Ricardo Villareal Esquivel. Funciones y tareas con respecto a la calidad:

- Administrar el Alcance de Antares.
- Acordar con el Gerente del área proveedora los procedimientos a seguir para los cambios al alcance del proyecto y el seguimiento al avance.

Administrar Trabajo:

- Dar seguimiento a las acciones y compromisos que se establecieron en el plan de trabajo del proyecto.

Administrar Organización y Comunicación:

- Identificar la estructura de los grupos y personas involucradas en el proyecto.
- Respetar los mecanismos de comunicación establecidos y asegurar que la información este disponible para los integrantes del proyecto.

Administrar Recursos y Servicios:

- Asegurarse de que las condiciones del medio ambiente sean las adecuadas para el desarrollo del proyecto.
- Proveer el ambiente de trabajo y recursos requeridos para la ejecución exitosa del proyecto.
- Proporcionar espacio de trabajo y tiempo de servicio de cómputo en sus instalaciones, de acuerdo a los recursos involucrados en el proyecto.
- Apoyo a requerimientos de ambientes de desarrollo, producción, cambios y en general de soporte técnico y base de datos.

- Líder de Proyecto: Funciones y tareas con respecto a la calidad :
- Ricardo Amaya
Méndez y Hugo
Humberto Zamora
Lavadores.
- Administrar el Alcance
 - Acordar con el área proveedora los procedimientos a seguir para los cambios al alcance del proyecto y el seguimiento al avance.
 - Evaluar los cambios al alcance y al contrato mediante el procedimiento de control de cambios establecido para el proyecto.

Administrar Trabajo:

- Dar seguimiento a las acciones y compromisos que se establecieron en el plan de trabajo.
- Identificar las desviaciones con respecto al plan y definir acciones correctivas.
- Proveer de información, datos y decisiones en un plazo que no afecte el calendario de trabajo establecido.
- Tomar las acciones directas que resulten de los planes de acción cuando sea apropiado ratificar desviaciones del plan, programa de actividades o procedimientos, que sean responsabilidad de las áreas involucradas de la GCSR9.
- Designar al personal necesario para la entrega y explicación de la documentación e información inherente al proyecto.

Jefe de
Departamento de
Optimización
General de Red:
Roberto Mondragón
Mateo

Administrar Organización y Comunicación:

- Identificar la estructura de los grupos y personas involucradas en el proyecto.
- Respetar los mecanismos de comunicación establecidos y asegurar que la información este disponible para los integrantes del proyecto.
- Proveer de información, datos y decisiones en un plazo que no afecte el calendario establecido de trabajo.
- Proveer la información necesaria para poder cumplir completamente con las obligaciones del área proveedora, definidos en la propuesta de servicios en un plazo mínimo que no afecte la continuidad del proyecto.

Administrar Recursos y Servicios

- Conseguir las facilidades, materiales, manuales y suministros que se requieran para la continuidad de los servicios.
 - Asegurarse de que las condiciones del medio ambiente sean las adecuadas para el desarrollo del proyecto.
-

2.2.5 PROCEDIMIENTOS DE ADMINISTRACIÓN DEL PROYECTO ANTARES

Control de
Cambios al
Alcance

Objetivo del Procedimiento:

Otorgar a las partes (GCSR9 y área proveedora), un mecanismo controlado y ordenado para solicitar cambios a los Requerimientos definidos originalmente en el alcance del proyecto estipulado, así como cualquier otro tipo de cambio que afecte al costo, al cronograma o al contenido del proyecto.

Reglas Generales:

- 1) Como en las demás negociaciones del proyecto, la interfase entre las partes para el manejo de cambios, serán los Gerentes de Proyecto de las organizaciones, quienes juntos forman el Comité de evaluación de cambios. El Comité es encabezado por el Gerente de Proyecto de la GCSR9.
 - 2) El Gerente de Proyecto de la GCSR9 y el área proveedora lleva el control de los cambios.
 - 3) Los cambios pueden ser identificados y solicitados tanto por la GCSR9 como por el área proveedora, aunque en la mayoría de los casos, serán originados por la GCSR9.
 - 4) Todo cambio identificado por el personal de la GCSR9, deberá ser canalizado a su correspondiente Líder de Proyecto, quien lo entregará al Líder de proyecto del área proveedora y estos después de validar los cambios los transfieren al Gerente de Proyecto respectivos para su evaluación en el comité.
 - 5) Todas las formas de solicitud de cambio, sean aceptadas o no, serán archivadas en el archivo del proyecto como parte de su documentación.
 - 6) Las Solicitudes de cambio, tienen validez legal, ya que en cada una
-

de ellas, se hace referencia al número de contrato que ampara este proyecto y una vez aprobadas, forman parte del contrato.

Pasos del Procedimiento:

- 1) Cuando el Líder de Proyecto de una de las partes identifica un cambio, presenta la forma de "Control de Cambios al Alcance" al Gerente de Proyecto del área proveedora, llenando la Sección I del formato del Anexo "B" .
- 2) El Gerente del área proveedora asignará un número consecutivo a la solicitud y determinará conjuntamente con el comité de evaluación de cambios, (Formado por los gerentes de las partes), si el cambio afecta el cronograma del Proyecto y/o si tiene un costo adicional para el cliente. La evaluación de cada solicitud de cambio no debe exceder los tres (3) días hábiles.
- 3) Después de hecho el análisis, el Gerente de Proyecto del área proveedora llenará la Sección II del formato que se localiza en el Anexo "B".
- 4) Si la solicitud resulta sin costo y sin impactos en el proyecto, se procede inmediatamente a las firmas de aprobación, esta corresponde a la sección III del formato del Anexo "B".
- 5) Si la solicitud resulta en cambio de fechas en el cronograma o en costo adicional para el cliente, el Gerente de Proyecto del área proveedora, presentará la Solicitud de Cambio y su costo al Comité Ejecutivo del Proyecto o al Gerente de Proyecto de la GCSR9 (quien tenga delegada la autoridad) para su aprobación, rechazo o aplazamiento para una etapa posterior del proyecto (en cuyo caso será archivado y revalidado posteriormente).
- 6) En caso de que el cambio sea aprobado, se llena la sección III del formato que se encuentra en el Anexo "B" y se hacen los ajustes necesarios al cronograma del proyecto. La forma de Control de Cambios al Alcance pasa entonces a formar parte integral del contrato y se procede a su facturación.
- 7) En caso de ser rechazado o ser diferido para una etapa posterior, se llena la sección III y se archivará en la carpeta de control del proyecto.

-
- 8) El Gerente de Proyecto del área proveedora informará al Comité Gerencial del avance de las solicitudes previamente aprobadas en las juntas de revisión.
-

Control de No Conformidades

Objetivo

Este procedimiento es obligatorio para manejar quejas y no conformidades del cliente.

Todos los asuntos que se originen en el proyecto, ya sean errores, fallas, problemas, incidentes, necesidades del cliente, riesgos, etc.

Tipos de asuntos:

- 1) Problema: Es un incidente que requiere discusión con una persona o grupo que está fuera del proyecto, antes de que se tomen acciones para resolverlo.
- 2) Incidente: Es una situación no prevista que requiere una acción dentro del proyecto para resolverlo.
- 3) Defecto: Es una falla o error que se presenta en los productos entregables del proyecto.
- 4) Necesidad del cliente: Cualquier necesidad expresada por el cliente, ya sea explícita o implícitamente.
- 5) No conformidad del cliente: Cualquier expresión de insatisfacción por parte del cliente.

Registro y seguimiento de asuntos

- 1) El registro y seguimiento se efectúa de forma integrada.
- 2) El registro de asuntos proporciona una buena base para el control de la calidad del proyecto, mostrando los tipos de asuntos, en que etapa del proyecto fueron detectados, cual área del proyecto está generando asuntos, cuántos asuntos ocurrieron, que tan rápido deben ser corregidos.
- 3) Una vez detectado el asunto, se le da seguimiento y solución a través de este procedimiento.
- 4) Con el propósito de firmar el cierre del asunto, el Gerente de Proyecto de la GCSR9 y el área proveedora deberán firmar una minuta. Con esto se da por concluido el asunto.

2.3 METODOLOGÍA DE DESARROLLO

Para las fases de desarrollo del sistema Antares, se dispuso apoyarme en la metodología definida por el Proceso Racional Unificado (RUP por sus siglas en inglés), las razones principales son las siguientes:

- Se centra en los modelos, permitiendo comunicar las necesidades del negocio mediante diagramas que, posteriormente se convertirán en los casos de uso del sistema Antares.
- Una vez obtenidos los casos de uso, contamos con poderosos elementos evaluadores tanto de la Arquitectura del Sistema, como de la funcionalidad esperada por el cliente.
- Por la constante interacción con los usuarios, evalúo como mejor opción, un desarrollo iterativo e incremental que me permita generar varias versiones evolutivas del sistema, teniendo al final de cada una, retroalimentación y observaciones por parte de los usuarios, mismas que me permitan reencausar los esfuerzos y recursos empleados.

El desarrollo del sistema estará regido por el paradigma Orientado a Objetos, teniendo como estándar para los diagramas, al Lenguaje de Modelado Unificado (UML) ya que permite mostrar diversas vistas o modelos del sistema.

Se ha optado por ese paradigma de programación, debido al rápido progreso en las etapas de desarrollo de software que este permite. Algunas ventajas notables, radican en la posibilidad de agrupar y organizar los programas en clases dependiendo de su funcionalidad; además de facilitar la reutilización de código fuentes, y contar con características como la herencia, las interfaces y en general el tratamiento de objetos como piezas fundamentales que encapsulan comportamientos y atributos propios. La tecnología de Antares sigue el estándar para sistemas empresariales J2EE y a ese respecto, cuenta con el desarrollo de los siguientes componentes:

- Servicio HTTP donde el API del lado del servidor está definido por los API Servlet y JSP.
- Acceso a datos y persistencia (Hibernate, JDBC, Criteria).
- Mensajería y e-mail con JavaMail.
- Utilización de Spring como modelo de componente para desarrollo empresarial en Java, sustituyendo la funcionalidad de transacciones de objetos EJBs.

2.3.1 PROCESOS

a) Los procesos de inicio, elaboración, construcción y transición se mapean a las fases del plan de trabajo como sigue:

1. Inicio → Inicio
2. Elaboración → Análisis de Requerimientos y Diseño
3. Construcción → Construcción del software y pruebas unitarias del mismo
4. Transición → Pruebas de integración y estabilización, implementación y capacitación

b) Dado que el tiempo es muy reducido para el alcance total del proyecto, se examinará la factibilidad de realizar iteraciones. Esto dependerá tanto de la complejidad funcional vista a detalle durante el análisis y diseño, como de la premura de tiempo para finalizar el proyecto, el cual ya tiene una fecha de término de antemano.

2.3.2 REVISIONES

1. El área proveedora realizará peer reviews (puntos de revisión) para todos los productos entregables antes de entregarlos a la GCSR).
2. Todos los artefactos que requieran firma de aceptación se entregarán listos para su Vo.Bo. y una vez firmados se archivarán en carpetas para cada fase del proyecto.

2.3.4 HERRAMIENTAS DE MODELADO Y DESARROLLO

1. Microsoft Visio para el modelado con UML.
2. StarUML v5.0.2 (<http://staruml.sourceforge.net/>): StarUML es una herramienta para el diseño de sistemas en UML. Tiene algunas características de Rational Rose como lo son el control por unidades, ingeniería de reversa y generación de código. El objetivo de utilizar StarUML es operar una herramienta de plataforma abierta para el diseño orientado a objetos.
3. DBVisualizer y AquaData para modelado y construcción de la base de datos
4. Tomcat 5.5 y WebLogic 8 como servidores de aplicación
5. Eclipse v3.2.1 (<http://www.eclipse.org/>): Eclipse es un IDE de Java para desarrollar aplicaciones. El objetivo de utilizar Eclipse es tener una herramienta de desarrollo que permite la integración de nueva funcionalidad a través de plug-in's que están disponibles abiertamente en la red.

Plug-in	Descripción
MyEclipse	MyEclipse es un conjunto de plug-ins para UML, AJAX, Web, J2EE, JSP, XML, Struts, Spring, JSF, y Hibernate. En este proyecto se estarán utilizando las facilidades que dan los plug-ins para: XML, Spring, y Hibernate.

En este apartado se mencionó brevemente la Metodología de Desarrollo de software empleado, sin embargo, posteriormente se entrará en el detalle de los casos de uso de que se compone el sistema Antares, así como en la especificación de los Servicios J2EE, los diagramas UML (Lenguaje de Modelado Unificado) involucrados en la construcción del sistema, así como el proceso de desarrollo RUP seguido hasta la obtención del producto final operando en producción.

CAPÍTULO III: FASE DE ELABORACIÓN

Los objetivos de este capítulo para el proceso de desarrollo de Antares son los siguientes:

- Elaborar el modelo de casos de uso definitivo, recopilando la mayor parte de los requisitos de la GCSR9.
- Establecer la arquitectura inicial del sistema con la cual, se guíe el trabajo en las fases de construcción y transición, así como las posteriores versiones del sistema Antares.

El cumplimiento de los objetivos anteriormente descritos, permiten tener una comprensión global del sistema Antares, ya que se conocerá su ámbito, así como los requisitos funcionales y no funcionales que la operación de la GCSR9 demanda.

3.1 MODELO DE CASOS DE USO

El modelo de casos de uso de Antares permite fijar el alcance de los requisitos funcionales de la GCSR9, para el proceso de atención a Cuentas Corporativas. Aborda las condiciones y posibilidades que debe cumplir el sistema.

3.1.1 ACTORES

El modelo de casos de uso describe lo que hace el sistema Antares para cada tipo de usuario, a su vez cada usuario es representado por uno o más actores quienes constituyen el entorno externo al sistema. Un actor es un tercero externo al sistema Antares pero que colabora con él, a continuación se presentan los actores identificados para el sistema Antares:

- 1) Empleado General de Telcel: Es el usuario que levantará las solicitudes de Servicio para la atención de una Cuenta Corporativa en Particular. Este usuario es de vital importancia en el seguimiento de los proyectos, ya que es el contacto que funge como frente con el cliente propietario o responsable de la cuenta corporativa.
- 2) Gerente de Calidad de Servicio Región 9: Es el usuario responsable de todas las cuentas corporativas. De él depende el cierre, Visto Bueno (VoBo) y finalización de un proyecto. Es la autoridad mayor en la toma de decisiones acerca de los proyectos.

- 3) Jefe de Optimización de Red Región 9: Es el usuario encargado de dar seguimiento a los proyectos, verificando los estatus de los mismos y que se cumplan las fechas compromiso establecidas. Debe anticipar riesgos, negociar las tareas necesarias para la atención con las áreas involucradas y servir de frente único con el Empleado General de Telcel que inició el requerimiento.
- 4) Analista de Optimización de Red Región 9: Será el usuario responsable del requerimiento durante todos los estatus del proyecto. Es el responsable de ejecutar las tareas necesarias para dar solución satisfactoria al requerimiento levantado por la Cuenta Corporativa. Deberá coordinar a las áreas involucradas en la atención del requerimiento o proyecto. Es el único autorizado para asignarle un estatus a los proyectos que tiene a su cargo. Establecerá fechas compromiso de solución con base en planes de trabajo surgidos del análisis realizado para cada requerimiento en particular.
- 5) Administrador: Es el usuario responsable de recibir todos los requerimientos levantados por las áreas usuarias (Empleados Generales de Telcel). Una vez analizado el requerimiento deberá asignarlo al Ingeniero responsable de la zona geográfica (Analista de Optimización de Red Región 9). Será el responsable de la administración de usuarios del sistema Antares. Solo este usuario puede modificar el rol de un usuario, así como permitir dar altas, bajas definitivas sobre el sistema.

3.1.2 DIAGRAMA DE CASOS DE USO DE ANTARES

La finalidad del diagrama de casos de uso es mostrar en conjunto, los casos de uso, actores y relaciones entre ellos y que componen la funcionalidad del sistema Antares. Cabe mencionar que “Los diagramas de casos de uso nos muestran la funcionalidad mediante los casos de uso de un sistema, desde el punto de vista estático” [8].

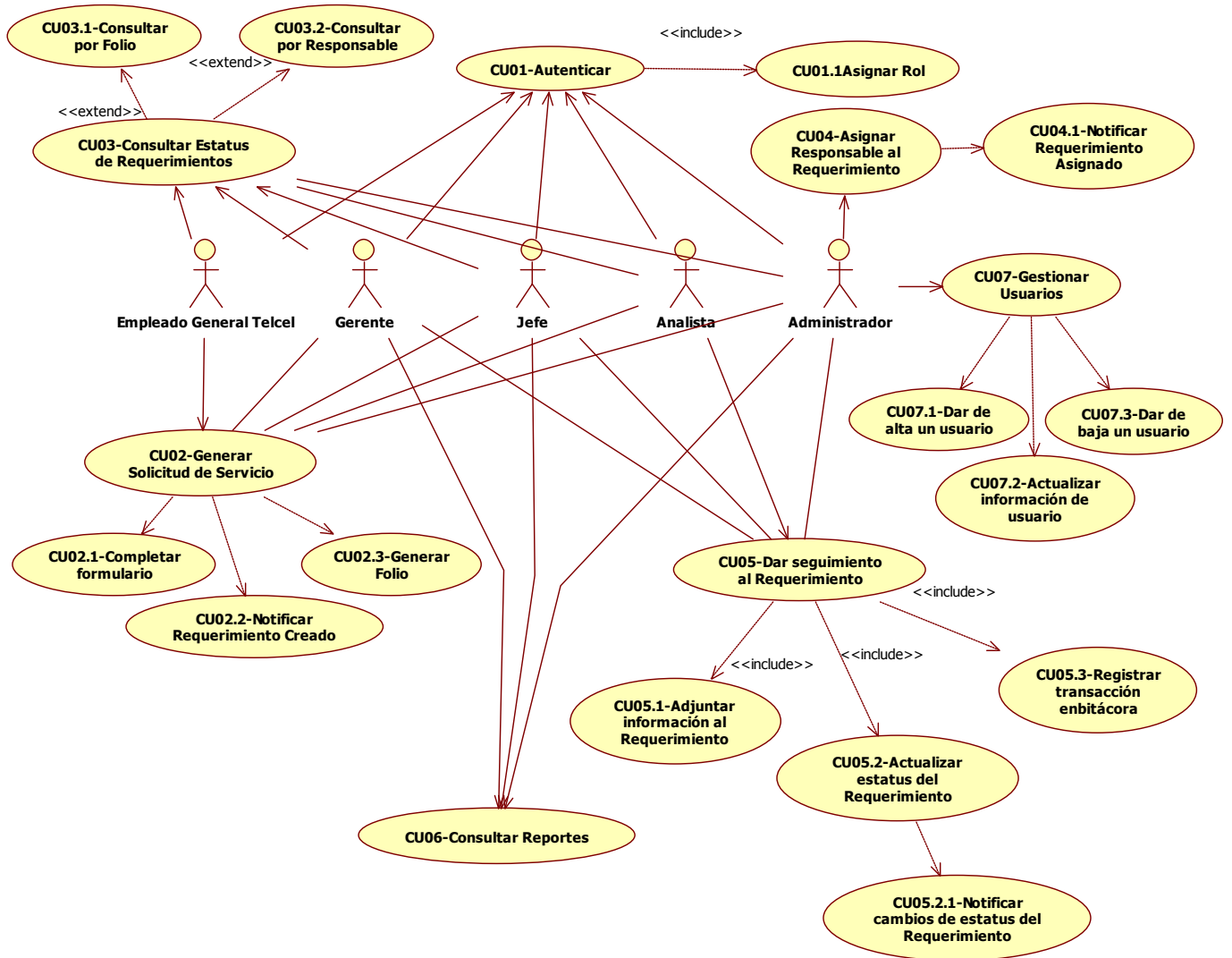


Fig. 3.1 Diagrama de casos de uso de Antares

3.2 ARQUITECTURA DE ANTARES

3.2.1 ALCANCE

Los objetivos de este apartado son muy puntuales:

- Presentar un análisis de los componentes principales del sistema Antares, la conducta de esos componentes según se les percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema.
- Definir la organización fundamental del sistema Antares encarnada en sus componentes arquitectónicos fundamentales, las relaciones entre ellos y el ambiente, así como los principios que orientan su diseño y evolución.
- Sentar las bases del diseño y la implementación, documentando de forma organizada las decisiones técnicas del proyecto, permitiendo a través del mismo entender mejor y solventar las diversas problemáticas que se presenten durante el desarrollo del sistema.

3.2.2 MECANISMOS Y RESTRICCIONES

A continuación se presentan los diferentes mecanismos empleados para este proyecto, así mismo se documenta y justifica la estructura y organización de la solución propuesta a los requerimientos de la GCSR9, mismos que dieron lugar al desarrollo de este proyecto.

Para realizar la descripción de los distintos componentes se utilizarán:

- Fragmentos de código
- Diagramas de UML.

Los mecanismos de que se compone la Arquitectura de Antares y para los cuales se argumentará acerca de sus ventajas e importancia en el proyecto son:

- 1) La Persistencia.
- 2) Creación y distribución de objetos.
- 3) Gestión de Transacciones.

Para facilitar la implementación de estos mecanismos se utilizarán los siguientes marcos de trabajo:

- a) Hibernate v3.2.1: Debido a que el sistema requiere remover código de SQL de la capa del negocio, y en general de cualquier tipo de clase que no pertenezca a la capa del modelo. Utilizando Hibernate en conjunto con Spring, dan paso al mecanismo que implementa el patrón de diseño DAO el cual permite desacoplar y encapsular los datos del negocio.
- b) Spring v2.0.2: Dado que el sistema requiere reducir el acoplamiento entre las distintas capas e introducir nueva funcionalidad de una forma totalmente desacoplada, se utilizará Spring y sus módulos para IoC y AOP. Esto facilitará tanto el desarrollo, como los futuros mantenimientos y ajustes que la aplicación demande.
- c) Struts v1.2.9: Como parte de la implementación de una arquitectura J2EE, y mediante la aplicación de los patrones MVC y Front Controller, Struts tiene la función de gestionar las capas de control y vista, de tal forma que se logre la separación de la vista del negocio y el acceso a datos de Antares. Adicionalmente debe proveer al sistema la habilidad de mostrar múltiples vistas para cada uno de los usuarios, ayudando a presentar la información de un modo personalizado; esto se logra mediante un acceso centralizado de las peticiones cliente (requests) provenientes de la capa de presentación, aprovechando el potencial de la clase RequestProcessor. Así mismo mencionar que, el uso de Struts promoverá la reutilización del código.

Las características para Hibernate3, Spring2 y Struts 1.2 mencionadas en este apartado son únicamente las utilizadas en la arquitectura de Antares y que no se encuentran en las versiones Hibernate2, Spring1 y Struts 1.1. Esto porque el propósito de utilizar esta tecnología es satisfacer los objetivos funcionales y de negocio de la GCSR9 más otros que son, inherentes a todo buen diseño y arquitectura, de tal suerte que también se reduzca el tiempo de desarrollo.

Características de Hibernate 3:

- Flexibilidad en el Mapeo.
- Soporte a los atributos insert, update, y fórmula para las propiedades de los componentes.
- Las entidades pueden ser mapeadas a subconsultas de SQL en lugar de a una tabla. Funcionalidad útil si se desea tener vistas que no estén en la base de datos, como es el caso de una versión mejorada del sistema Antares, que permita explotar información de las

bases de datos de otras áreas como Radio Frecuencia, Compras, Centros de Atención a Clientes (CAC Telcel), entre otras.

- Soporte para Stored Procedures y Queries SQL en Mapping
- Hibernate 3 permite escribir queries en SQL dentro del Mapping. También es posible mapear entidades a los resultados que genere un Stored Procedure.
- Incluye soporte para retornar valores escalares y cargar colecciones.
- Implementa métodos CRUD para entidades permitidas en el ambiente de J2EE.

Características de Spring 2:

- Spring2 provee soporte para Hibernate3.
- Fácil migración de versiones, en este caso Spring 1.2 a Spring 2.X: En el mejor de los casos la migración es transparente, sólo hay que reemplazar el JAR en la ruta apropiada.
- Spring 2.0 remueve muchos de los métodos y clases marcados como deprecated. En caso de que la aplicación utilice métodos o clases marcados como deprecated será necesario reemplazarlos y volver a compilar los archivos fuente.

MECANISMO DE PERSISTENCIA

En la capa de Persistencia se implementará el patrón DAO (Data Access Object) para el acceso a los datos. La configuración se realizará a través de Hibernate v 3.2.1 y Spring v 2.0.2. El patrón DAO describe una implementación genérica y contiene los métodos básicos CRUD (Create, Read, Update, Delete). La siguiente interfaz define los métodos para el DAO genérico:

```
public interface GenericDAO {
    /**
     * Persiste un Nuevo objeto en la base de datos
     */
    public Serializable create(Object obj);
    /**
     * Recupera un objeto previamente persistido en la base
     * de datos usando el identificador como llave primaria.
     */
    public Object read(Serializable id);
    /**
     * Guarda las modificaciones hechas a un objeto persistido.
     */
    public void update(Object obj);
    /**
     * Remueve un objeto persistido de la base de datos.
     */
    public void delete(Object obj);
}
```

```
}
```

En la implementación del DAO genérico se invocan los métodos correspondientes de Hibernate y a Spring se encarga de la creación de la sesión y el control de las transacciones.

```
public class GenericDAOHibernateImpl
    implements GenericDAO {

    public GenericDaoHibernateImpl(Class type) {
        this.type = type;
    }
    public Serializable create(Object obj) {
        return getSession().save(obj);
    }
    public Object read(Serializable id) {
        return getSession().get(type, id);
    }
    public void update(Object obj) {
        getSession().update(obj);
    }
    public void delete(Object obj) {
        getSession().delete(obj);
    }
    public Session getSession() {
        boolean allowCreate = true;
        return SessionFactoryUtils.
            getSession(sessionFactory, allowCreate);
    }
    /**
     * El objeto sessionFactory se inicializa a través de IoC
     * de Spring
     */
    public void setSessionFactory(SessionFactory sessionFactory) {
        this.sessionFactory = sessionFactory;
    }
}
```

El patrón se complementa con la asociación de un objeto bean de java denominada POJO, el cual representa a una entidad en específico de la base de datos.

```
// Objeto POJO que representa una entidad de la base de datos
class Catalog implements java.io.Serializable {

    private Integer id;
    private String name;
    private String description;

    public Catalog() {
    }

    public Integer getId(){
```

```
        return this.id;
    }
    public void setId(Integer id){
        this.id = id;
    }
    public String getName(){
        return this.name;
    }
    public void setName(String name){
        this.name = name;
    }
    public String getDescription(){
        return this.description;
    }
    public void setDescription(String description){
        this.description = description;
    }
}
}
```

Esta representación de la entidad debe mapearse en un archivo XML, Catalog.hbm.xml, que contendrá lo siguiente:

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping
PUBLIC "-//Hibernate/Hibernate Mapping DTD//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-2.0.dtd">
<hibernate-mapping>
    <class
        name="com.gpo.modelo.model.pojos.Catalog"
        table="CATALOG">

        <id
            name="id"
            column="CATALOG_ID"
            type="long">
                <generator class="native"/>
        </id>
        <property
            name="name"
            column="NAME"
            type="string"/>
        <property
            name="description"
            column="DESCRIPTION"
            type="string"/>
    </class>
</hibernate-mapping>
```

Se utiliza la propiedad Introduction (Inter-type Declaration) en Spring (AOP) para agregar funcionalidad a los objetos DAO. Estos son instanciados encapsulándolos en un Proxy. La finalidad es que todos los métodos que no son soportados por el DAO genérico, es decir, que no pertenezcan a los métodos CRUD, y que puedan ser utilizados por las especializaciones, se

controlen por una sola clase. Para ilustrar la configuración, se utilizará el ejemplo de una clase de genérica y su DAO correspondiente.

Se definen las interfaces de los DAO's especializados:

```
package com.gpo.modelo.model.daos;

import com.gpo.modelo.model.pojos.Usr;

public interface UsuarioDao {
    public void guardarUsuario();
    public Usr autenticar(String login, String password);
}
```

La siguiente configuración se agrega en Spring para declarar la especialización del DAO genérico y se describe a detalle más adelante:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">

<beans>
  <!-- Modulo de AUTENTICACION de usuario -->
  <bean id="usuario"
class="com.gpo.modelo.negocio.servicios.impl.UsuarioImpl">
    <property name="usuarioDao">
      <ref local="usuarioDao" />
    </property>
  </bean>

  <bean id="usuarioDao"
class="com.gpo.modelo.model.daos.impl.UsuarioDaoImpl">
    <property name="sessionFactory">
      <ref local="sessionFactory" />
    </property>
  </bean>

  <!-- Modulo de CUENTAS CORPORATIVAS -->
  <bean id="cuentasCorp"
class="com.gpo.modelo.negocio.servicios.impl.CuentasCorpImpl">
    <property name="cuentasCorpDao">
      <ref local="cuentasCorpDao" />
    </property>
  </bean>

  <bean id="cuentasCorpDao"
class="com.gpo.modelo.model.daos.impl.CuentasCorpDaoImpl">
    <property name="sessionFactory">
      <ref local="sessionFactory" />
    </property>
  </bean>
```

```
<!-- Hibernate sessionFactory -->
<bean id="sessionFactory"

class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
    <property name="configLocation">
        <value>WEB-INF/hibernate.cfg.xml</value>
    </property>
</bean>

</beans>
```

La clase UsuarioDaoImpl permite que una instancia de esta sea inyectada al servicio (clase de negocio) mediante uno o más AOP.

MECANISMOS DE CREACIÓN Y DISTRIBUCIÓN DE OBJETOS

Declaración de Servicios en Spring: Los DAO's y servicios se inicializan mediante el patrón IoC (Inversion of Control) de Spring. Para crear una instancia de un servicio que utiliza un DAO, éste se inyecta al servicio mediante IoC de la siguiente forma:

```
<bean id="usuario"
class="com.gpo.modelo.negocio.servicios.impl.UsuarioImpl">
    <property name="usuarioDao">
        <ref local="usuarioDao" />
    </property>
</bean>
```

La clase UsuarioImpl es la implementación de UsuarioInterface:

```
package com.gpo.modelo.negocio.servicios;

import com.gpo.modelo.model.pojos.Usr;

public interface UsuarioInterface {
    public void guardar();
    public Usr autenticar(String login, String password);
}
```

MECANISMO DE GESTIÓN DE TRANSACCIONES

El Control de Transacciones con Spring y Hibernate: Mediante Spring se realiza el control de transacciones utilizando la configuración transaccional de Hibernate, la cual aplica sobre un solo objeto de sesión de Hibernate, el sessionFactory. Es posible configurar el nivel de transacción por

operación mediante el interceptor TransactionProxyFactoryBean. De esta clase se derivaran los servicios de la aplicación.

```
<!-- Hibernate SessionFactory -->
<bean id="sessionFactory"
      class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
  <property name="configLocation">
    <value>WEB-INF/hibernate.cfg.xml</value>
  </property>
</bean>
```

Para integrar una clase de servicio al control de transacciones, se realiza la declaración en el archivo applicationContext.xml de la siguiente forma, considerando que el servicio encapsula a un DAO, y la transacción contempla la ejecución de la clase del servicio más el DAO:

```
<bean id="usuario"
      class="com.gpo.modelo.negocio.servicios.impl.UsuarioImpl">
  <property name="usuarioDao">
    <ref local="usuarioDao" />
  </property>
</bean>

<bean id="usuarioDao"
      class="com.gpo.modelo.model.daos.impl.UsuarioDaoImpl">
  <property name="sessionFactory">
    <ref local="sessionFactory" />
  </property>
</bean>
```

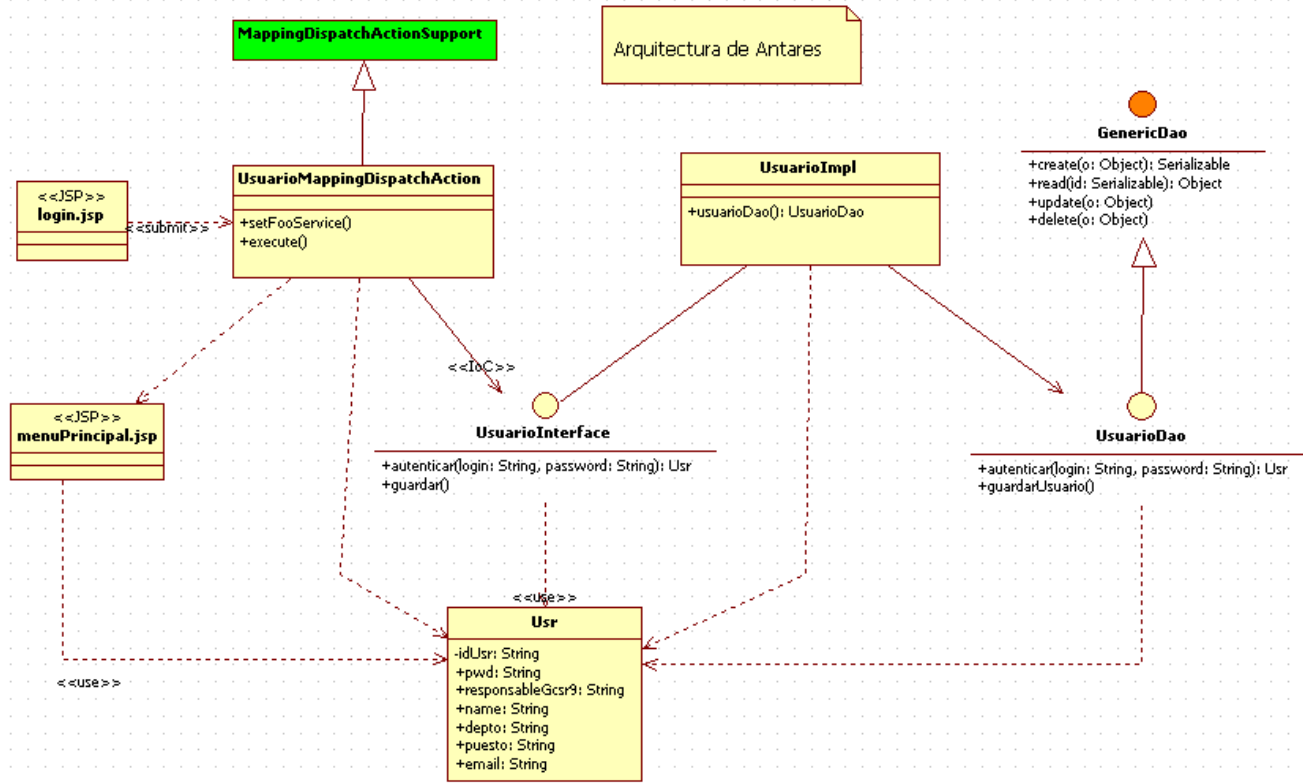


Fig. 3.2 Diagrama de clases de la arquitectura de Antares

3.3 ANÁLISIS DE ANTARES

El objetivo del modelo de análisis de Antares es, ayudar a refinar y estructurar los requisitos de la GCSR9, así como también razonar sobre los aspectos internos del sistema, de tal manera que el proyecto se encamine al desarrollo de un sistema Antares de fácil mantenimiento, flexible a cambios y reutilizable para las posteriores versiones de desarrollo planeadas por la GCSR9.

La estrategia utilizada según la metodología del proceso unificado de desarrollo de software, es la realización de los casos de uso – análisis para los requerimientos de la GCSR9, esto con la finalidad de contribuir con el modelo de análisis para describir como se llevan a cabo los casos de uso, en términos de las clases y objetos en interacción. La especificación de cada caso de uso posee una descripción textual del flujo de sucesos o actividades, diagramas de clases y diagramas de flujo de datos que muestran la realización de cada flujo básico o escenario de la operación diaria de la GCSR9.

3.3.1 ESPECIFICACIÓN DE CASOS DE USO

CU01-AUTENTICAR

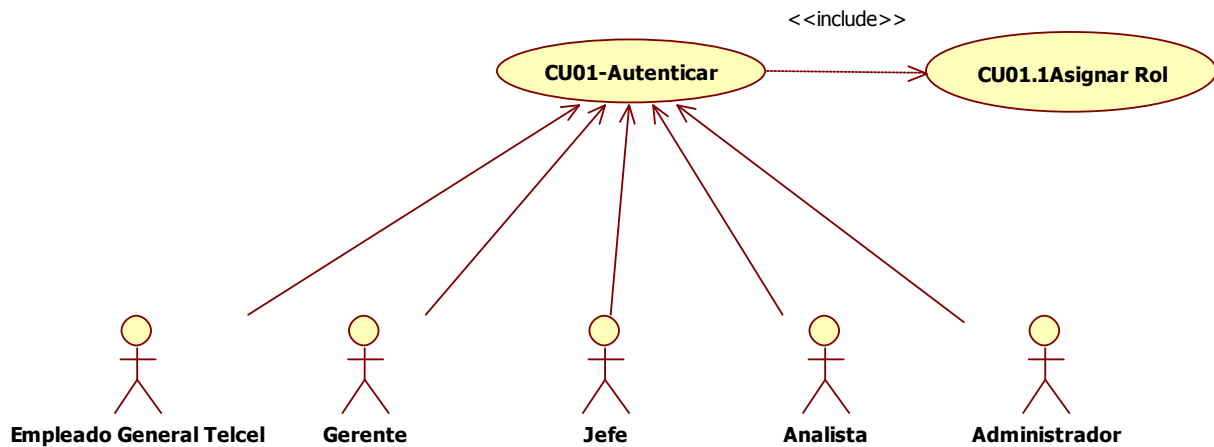


Fig. 3.3 Diagrama del Caso de Uso CU01-Autenticar

a. Descripción breve: Autenticar el acceso al sistema Antares, así como asignar privilegios a usuarios sobre el sistema dependiendo del rol que desempeñen.

b. Actores involucrados:

- Empleado general Telcel
- Analista
- Administrador
- Gerente
- Jefe

c. Flujo básico:

1. El caso de uso inicia cuando el actor <Empleado general Telcel, Analista, administrador, gerente o jefe> ingresa mediante el navegador de Microsoft Internet Explorer 6 (o posterior), a la URL <http://localhost:8080/antares/> con lo que se mostrará la pantalla de inicio de sesión del sistema Antares.
2. En la pantalla de inicio de sesión aparecen dos campos, Cuenta y Contraseña, para los cuales el actor <Empleado general Telcel, Analista, administrador, gerente o jefe> deberá ingresar la información correspondiente para finalmente solicitar el acceso al sistema Antares.

3. Si los datos que ingresó al usuario son inválidos, se ejecuta lo correspondiente al Flujo alternativo 1 <Error de Autenticación>.
 4. Si los datos ingresados por el usuario son correctos, el sistema Antares asigna los privilegios correspondientes al rol del usuario en cuestión y le concede el acceso a la pantalla principal del sistema Antares.
 5. Finaliza el caso de uso.
- d. Flujos alternos:
- Flujo alternativo 1 <Error de Autenticación>: Si los datos ingresados por el usuario son inválidos, el sistema redirecciona a la página de “Error de Autenticación”, en la que mostrará el siguiente mensaje “Error al autenticar usuario, intente nuevamente”. Acto seguido deberá permitir el regreso a la pantalla de inicio de sesión para que el actor pueda intentar el acceso al sistema Antares una vez más.
- e. Reglas de negocio:
1. El actor que desee ingresar al sistema Antares, deberá estar registrado en la base de datos de la GCSR9, también deberá contar con un puesto y deberá estar asignado a un departamento reconocido por Radio Móvil DIPSA.
 2. El sistema Antares deberá asignar al usuario solamente los privilegios que le correspondan a su rol. Debe validar que no tenga privilegios o funciones que no sean propias de su perfil.
- f. Precondiciones:
1. Debe existir en la base de datos de la GCSR9 la relación de usuarios que utilizarán el sistema Antares.
- g. Pos condiciones:
- N/A
- h. Diagrama de Actividades:

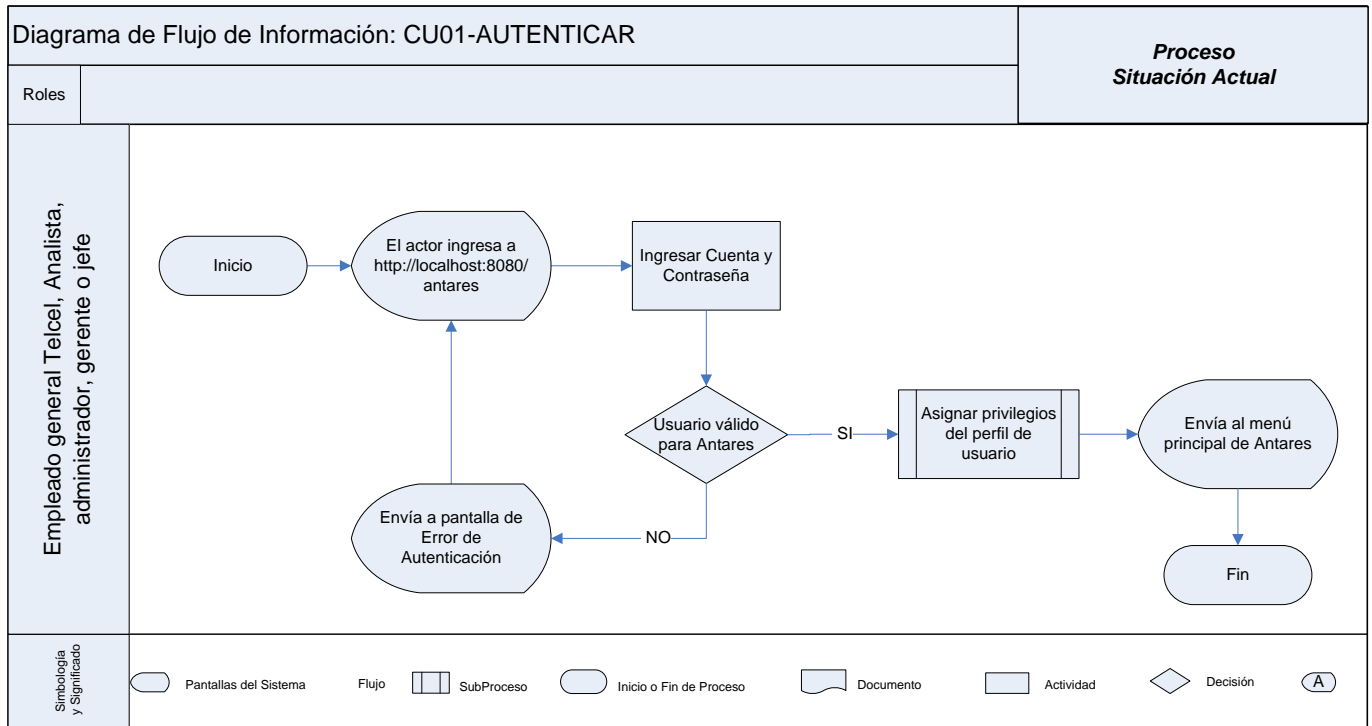


Fig. 3.4 Diagrama de actividades CU01-Autenticar

CU02-GENERAR SOLICITUD DE SERVICIO

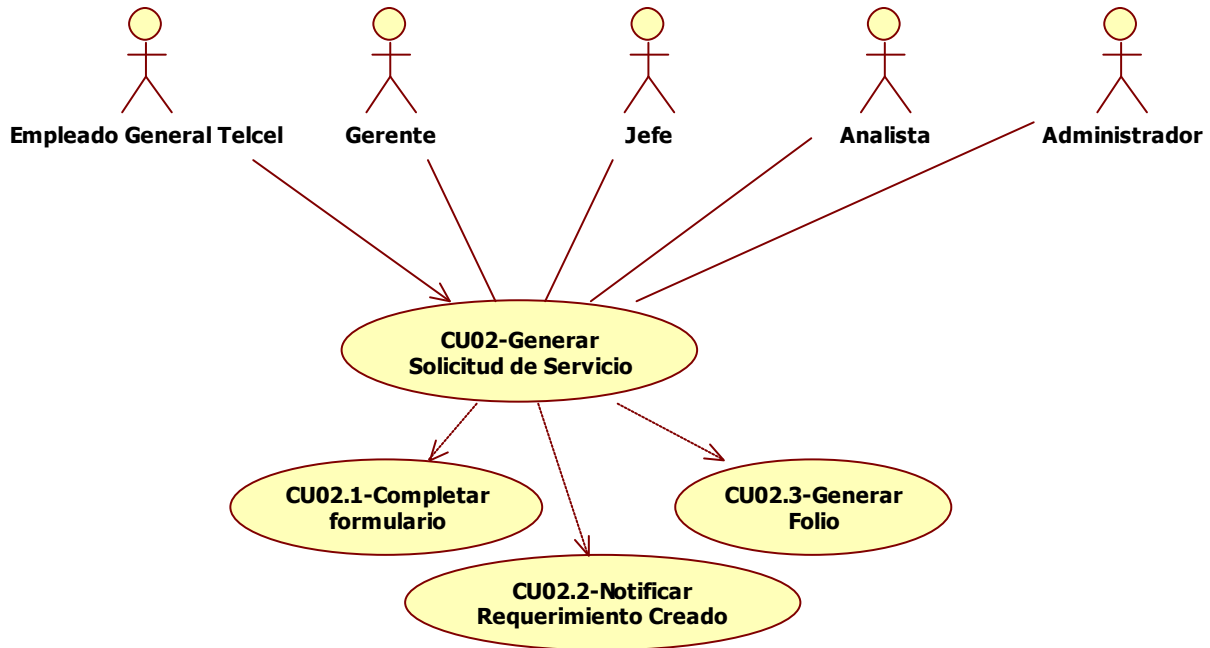


Fig. 3.5 Diagrama del Caso de Uso CU02-Generar Solicitud de Servicio

- a. Descripción breve: Generar una Solicitud de Servicio para atención de una Cuenta Corporativa que reporta una falla en el servicio proporcionado por Radio Móvil DIPSA.
- b. Actores involucrados:
- Empleado general Telcel
 - Analista
 - Administrador
 - Gerente
 - Jefe
- c. Flujo básico:
1. El caso de uso inicia cuando el actor <Empleado general Telcel, Analista, administrador, gerente o jefe> previamente autenticado en el sistema Antares, selecciona del menú principal la opción “Cuentas Corporativas” y posteriormente invoca la generación de una “Nueva Solicitud”.
 2. El sistema Antares muestra un formulario que deberá ser llenado por el actor en cuestión. Los datos presentados por el formulario son indispensables para la generación del requerimiento de atención a Cuenta Corporativa. La información recabada por el formulario debe incluir lo siguiente:
 - Fecha de generación del requerimiento.
 - Información de la persona solicitante, que permita mantener una comunicación inmediata con los representantes de la cuenta corporativa.
 - Información que detalle la falla presentada en la cuenta corporativa, contemplando las necesidades de servicio requerido por parte de la compañía, así como información referente a la tecnología empleada en las instalaciones..
 - Características técnicas de la infraestructura instalada en la cuenta corporativa.
 3. El actor deberá poder adjuntar a la solicitud de atención, información adicional que funja como evidencia para sustentar la falla reportada a la GCSR9.
 4. Al capturar toda la información solicitada por el formulario, el usuario debe invocar la generación de la solicitud de atención, y si uno o más campos son incorrectos se ejecuta lo correspondiente al Flujo alterno 1 <Completar formulario>.
 5. Una vez generada la solicitud de atención, Antares deberá notificar al usuario Administrador por medio de un correo electrónico, sobre la creación del nuevo requerimiento.

6. Finaliza el caso de uso.

d. Flujos alternos:

- Flujo alternativo 1 <Completar formulario>: Si los datos ingresados por el usuario al formulario son inválidos, el sistema Antares enviará un mensaje de alerta indicando el campo en conflicto. Posteriormente deberá permitir la corrección del campo y seguir adelante con la generación de la solicitud de atención.

e. Reglas de negocio:

1. Todos los datos solicitados por el formulario deberán ser capturados por completo y en un formato establecido por la GCSR9; el sistema Antares debe contribuir con el usuario para la correcta captura de los datos.
2. Adjuntar información adicional al requerimiento es opcional, y no impide la generación de la solicitud de atención a la cuenta corporativa, si el resto de los campos están completos y correctos.

f. Precondiciones:

1. Que alguna cuenta corporativa tenga problemas con el servicio proporcionado por Radio Móvil DIPSA y levante una solicitud de atención con alguno de los frentes: GCSR9, Mesa de ayuda, Radiobases BSC o centrales MSC.

g. Pos condiciones:

N/A

h. Diagrama de Actividades:

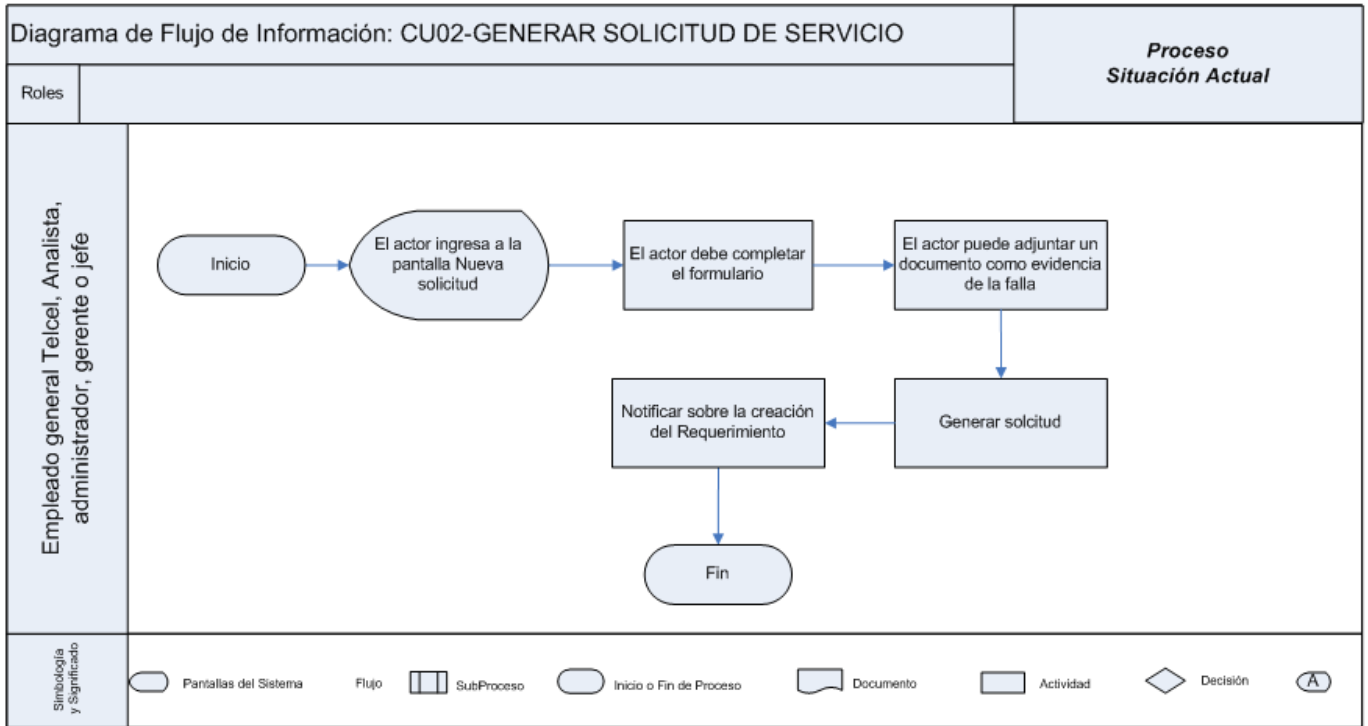


Fig. 3.6 Diagrama de actividades CU02-Generar Solicitud de Servicio

CU03-CONSULTAR ESTATUS DE REQUERIMIENTOS

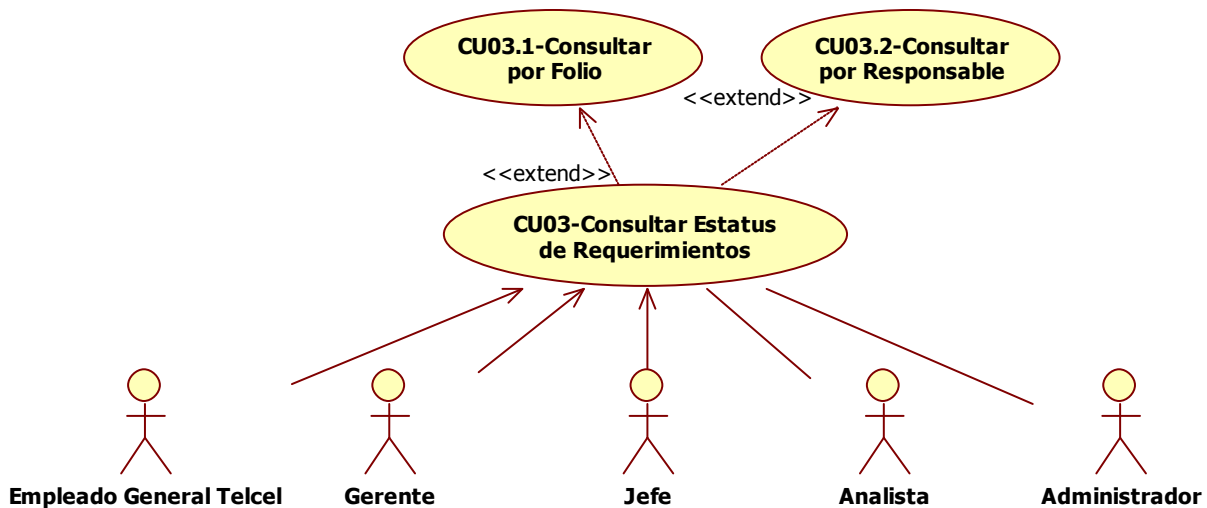


Fig. 3.7 Diagrama del Caso de Uso CU03-Consultar estatus de requerimientos

- Descripción breve: Consultar el estatus, así como las acciones emprendidas por parte de la GCSR9 para el tratamiento de una Solicitud de Servicio para la atención de una Cuenta Corporativa.

b. Actores involucrados:

- Empleado general Telcel
- Analista
- Administrador
- Gerente
- Jefe

c. Flujo básico:

1. El caso de uso inicia cuando el actor <Empleado general Telcel, Analista, administrador, gerente o jefe> previamente autenticado en el sistema Antares, selecciona del menú principal la opción “Cuentas Corporativas” y posteriormente invoca la opción de “Consulta de Estatus”.
2. A continuación el actor deberá seleccionar el criterio de búsqueda del Requerimiento:
 - Por folio: El usuario podrá realizar la búsqueda por el número del folio asignado al requerimiento en cuestión.
 - Por responsable: El usuario podrá realizar la búsqueda a partir de una lista, al usuario responsable (Analista de Optimización de Red Región 9) de la atención del requerimiento.
3. El sistema Antares deberá presentar en una pantalla el resumen del (de los) requerimiento(s) que cumplan con el criterio de búsqueda. Los datos mostrados deberán incluir el folio, la información del empleado que está atendiendo el requerimiento, la descripción del error o falla reportada por la cuenta corporativa y el estatus.
4. Finalmente se podrá consultar a detalle las acciones realizadas por las áreas involucradas en la atención del requerimiento en cuestión. Dichas acciones serán coordinadas por la GCSR9, área que en todo momento dará seguimiento al proceso de solución de los requerimientos.
5. Dentro de la pantalla de consulta del detalle del requerimiento, se podrá consultar el documento adjunto al requerimiento en el momento de su creación.
6. Finaliza el caso de uso.

d. Flujos alternos:

N/A

e. Reglas de negocio:

N/A

f. Precondiciones:

N/A

g. Pos condiciones:

N/A

h. Diagrama de Actividades:

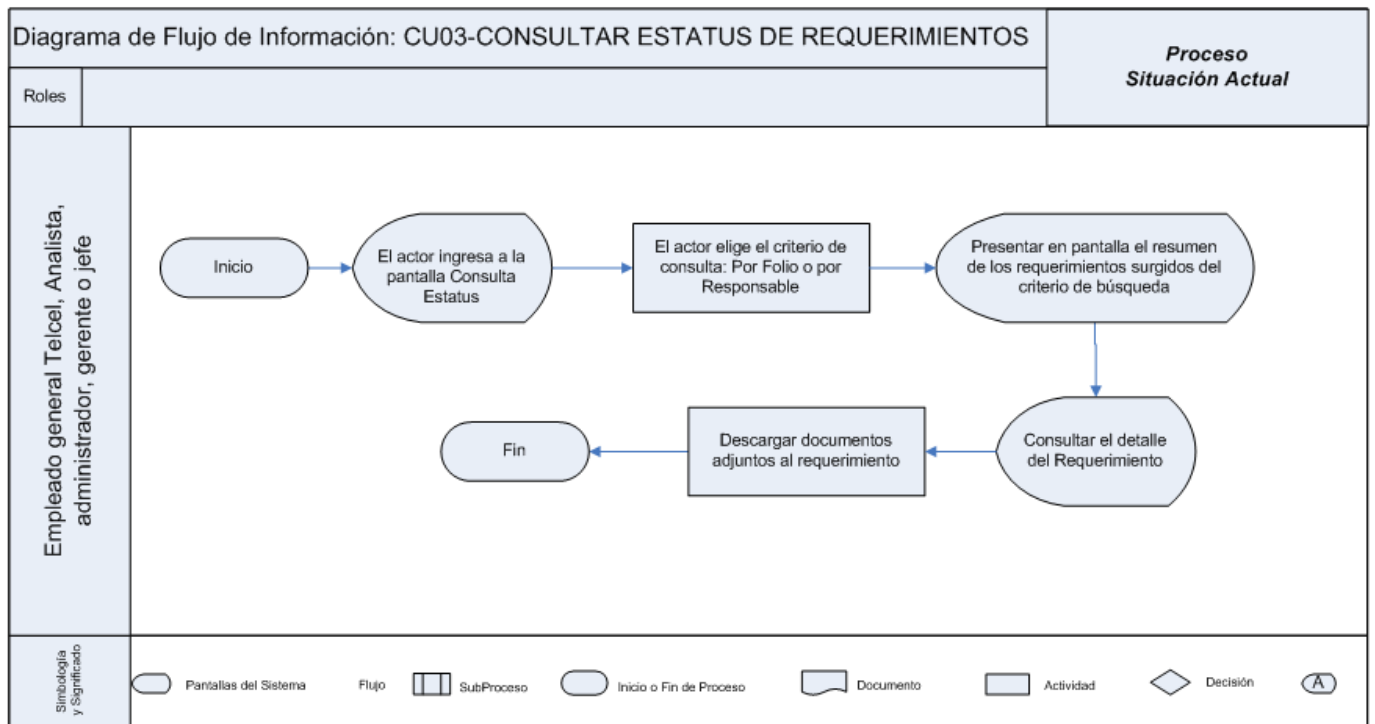


Fig. 3.8 Diagrama de actividades CU03-Consultar estatus de requerimientos

CU04-ASIGNAR RESPONSABLE AL REQUERIMIENTO

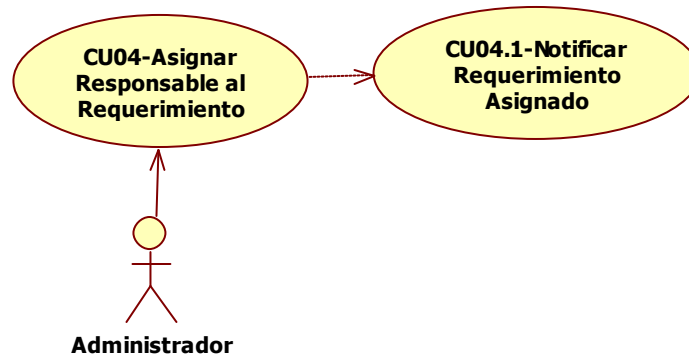


Fig. 3.9 Diagrama del Caso de Uso CU04-Asignar responsable al requerimiento

- a. Descripción breve: Asignar el requerimiento recién generado a un Analista de Optimización de Red Región 9, quien será responsable de dar solución y respuesta a la cuenta corporativa.
- b. Actores involucrados:
- Administrador
- c. Flujo básico:
1. El caso de uso inicia cuando el actor <Administrador > previamente autenticado en el sistema Antares, selecciona del menú principal la opción “Cuentas Corporativas” y posteriormente invoca la opción “Seguimiento de Folios”.
 2. A continuación el sistema Antares deberá mostrarle en su bandeja operativa, una lista con los requerimientos recién creados y que presentan un estatus de “CREADO”. En caso de que no existan requerimientos pendientes por asignar, se ejecuta lo correspondiente al Flujo alternativo 1 <Consulta y reasignación de folios>.
 3. El actor deberá gestionar cada uno de los requerimientos, siendo su tarea principal, la de consultar el detalle de la solicitud, evaluar la severidad de la falla reportada, así como identificar la zona geográfica en donde se ubica la cuenta corporativa, a fin de tener los criterios necesarios para asignar el requerimiento a un ingeniero (Analista de Optimización de Red región 9), quien será el responsable de dar solución y respuesta a la cuenta cliente de Radio Móvil DIPSA.

4. Posteriormente a la asignación del requerimiento, Antares envía un correo electrónico notificando al ingeniero sobre la asignación y responsabilidad que le fue delegada para la atención de la falla reportada por la cuenta corporativa
 5. Finaliza el caso de uso.
- d. Flujos alternos:
- Flujo alternativo 1 < Consultar y reasignación de folios >: En la pantalla del actor puede realizar una consulta y reasignación de un folio previamente creado. El actor deberá tener a su disposición los criterios de consulta de folios por número de folio y por responsable del mismo. Una vez en la pantalla detalle del folio, el actor estará en facultad de reasignar el requerimiento a otro analista de optimización de red región 9, teniendo que capturar una breve justificación para tal acción.
- e. Reglas de negocio:
1. Solo se podrán asignar aquellos requerimientos cuyo folio tenga un estatus de "CREADO".
 2. El criterio de asignación de cada requerimiento depende de la región geográfica de la cuenta corporativa, así como de la carga de trabajo que los departamentos (Sur, norte, centro, Optimización general) tengan al momento de la elaboración de la solicitud de servicio.
 3. Durante el tiempo en que el requerimiento permanezca con estatus "CREADO", es responsabilidad del administrador de Antares dar seguimiento del mismo. Es por ello que debe ser delegado a la brevedad posible.
 4. Solo se podrá efectuar la acción de reasignación de requerimiento si se captura una justificación en la que se detallen los motivos por los cuales, el requerimiento pasa a manos de otro ingeniero responsable de la GCSR9.
- f. Precondiciones:
1. Los requerimientos a asignar deben tener estatus de "CREADO".
 2. La reasignación se realiza solo si, el ingeniero actual responsable del requerimiento solicita expresamente apoyo, y lo documenta como una observación en el folio. La solicitud de apoyo puede deberse a una excesiva carga de trabajo, vacaciones o incapacidad médica para el ingeniero responsable.
- g. Pos condiciones:

1. Aún cuando el requerimiento sea reasignado, las acciones realizadas por el primer responsable permanecerán registradas en la base de datos de la GCSR9; a estas se sumarán las posteriores acciones surgidas del seguimiento del requerimiento.

h. Diagrama de Actividades:

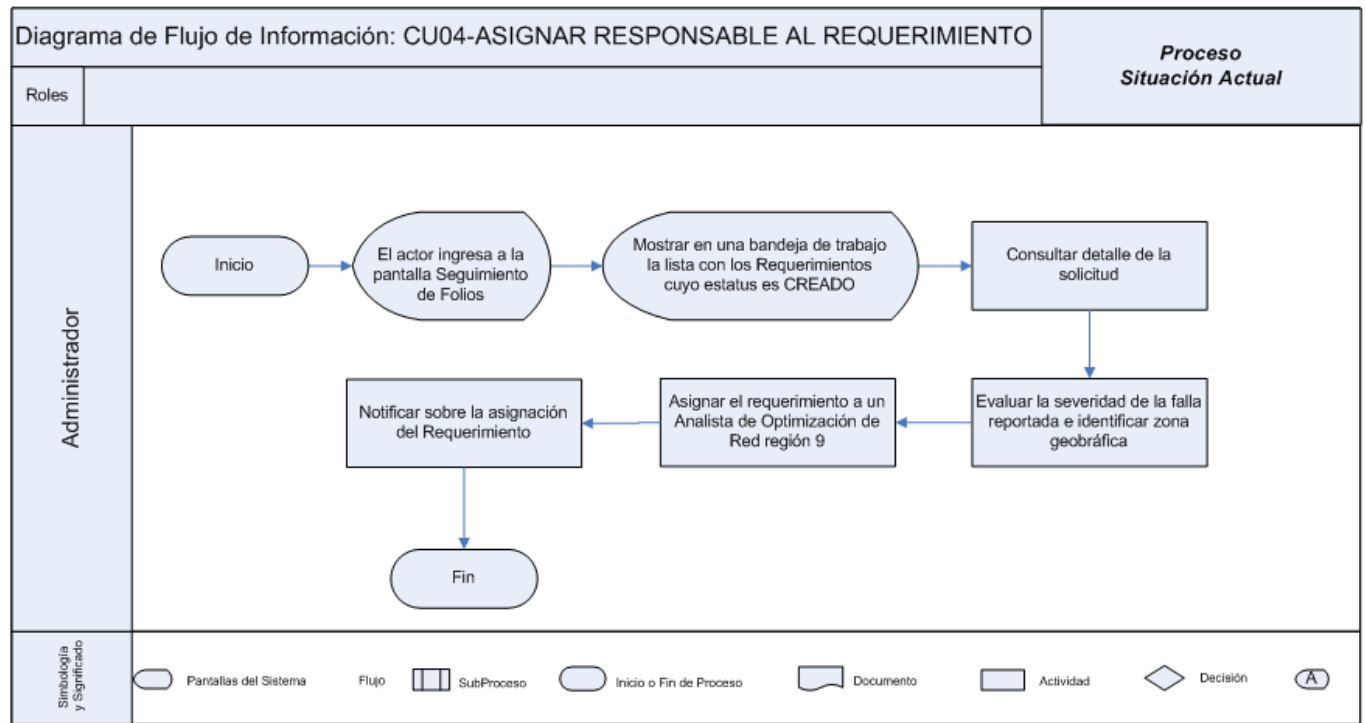


Fig. 3.10 Diagrama de actividades CU04-Asignar responsable al requerimiento

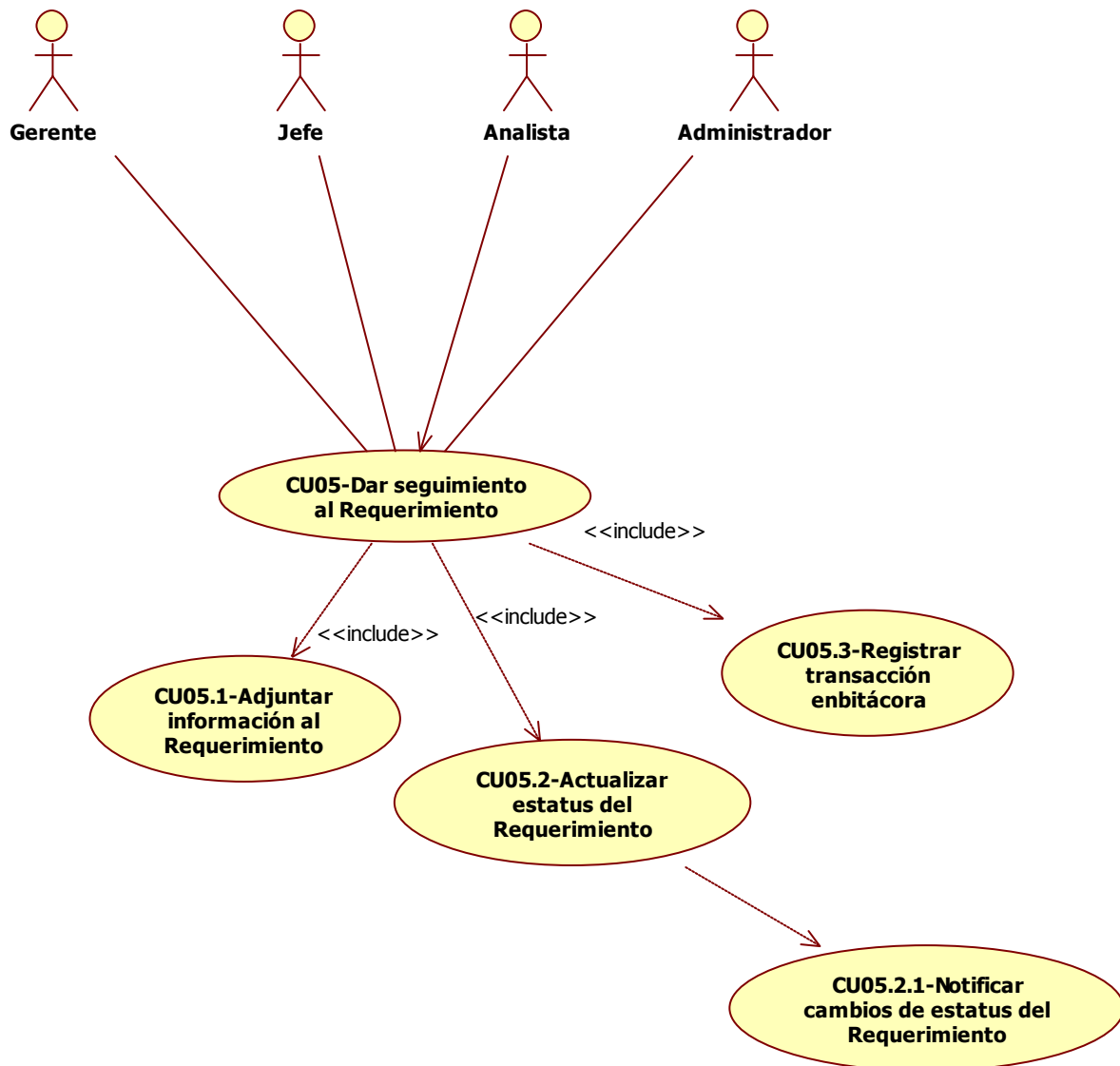
CU05-DAR SEGUIMIENTO AL REQUERIMIENTO

Fig. 3.11 Diagrama del Caso de Uso CU05-Dar seguimiento al requerimiento

- a. Descripción breve: Dar seguimiento a los requerimientos generados para la atención y evaluación del servicio de las cuentas corporativas que sostienen un contrato con Radio Móvil DIPSA.
- b. Actores involucrados:
- Analista
 - Administrador
 - Gerente

- Jefe

c. Flujo básico:

1. El caso de uso inicia cuando el actor <Analista, administrador, gerente o jefe> previamente autenticado en el sistema Antares, selecciona del menú principal la opción “Cuentas Corporativas” y posteriormente invoca la acción de “Seguimiento de Folios”.
2. El sistema Antares muestra una vista a cada actor, dependiendo de su perfil.
 - a. Analista: Se ejecuta lo correspondiente al Flujo alterno 1 <Seguimiento de Requerimientos perfil Analista>.
 - b. Administrador: Se ejecuta lo correspondiente al Flujo alterno 2 <Seguimiento de Requerimientos perfil Administrador >.
 - c. Gerente y Jefe: Se ejecuta lo correspondiente al Flujo alterno 3 <Seguimiento de Requerimientos perfil Supervisor>.
3. Finaliza el caso de uso.

d. Flujos alternos:

- Flujo alterno 1 <Seguimiento de Requerimientos perfil Analista>:
 1. Antares muestra al actor <Analista> una lista con los requerimientos que tiene bajo su responsabilidad.
 2. En esa bandeja de trabajo, el actor inicialmente consulta la información más relevante de cada requerimiento.
 3. El actor ingresa al detalle del requerimiento, pantalla en la que puede realizar el seguimiento del mismo.
 4. El actor realiza la gestión del requerimiento, registrando cada acción emprendida para la solución del mismo. Cada transacción realizada por el actor, es registrada en la bitácora de seguimiento del requerimiento.
 5. El actor en todo momento, puede adjuntar al requerimiento, algún documento relevante que sirva para sustentar las acciones emprendidas como parte del seguimiento en la solución al requerimiento.
 6. Después de las acciones realizadas al requerimiento, el actor deberá reflejar el avance del requerimiento mediante el establecimiento de un estatus representativo. Los estatus son los siguientes:
 - CANCELADO
 - CERRADO
 - CREADO

- EN ESPERA DE PROYECTO
 - EN PROCESO POR AREA COMERCIAL
 - EN PROCESO POR AREA JURIDICA
 - EN PROCESO POR CS CORPORATIVO (REG 1-8)
 - EN PROCESO POR DISEÑO DE RF R9
 - EN PROCESO POR GCSR9
 - EN OBSERVACION
 - SOLUCIONADO OK
7. Al actualizar el estatus del requerimiento, Antares enviará al responsable, así como al jefe del responsable, un correo electrónico de notificación acerca del cambio de estatus. Este correo representa un documento vivo de evidencia acerca del seguimiento y avance en la atención a la solicitud de la cuenta corporativa cliente.
- Flujo alternativo 2 <Seguimiento de Requerimientos perfil Administrador >:
 1. Se ejecuta lo correspondiente al Caso de Uso <CU04-Asignar responsable al requerimiento>.
 - Flujo alternativo 3 <Seguimiento de Requerimientos perfil Supervisor >:
 1. Antares muestra al actor <Supervisor> una lista con los requerimientos que no han sido atendidos. También le presenta los criterios de búsqueda por número de folio o por responsable.
 2. Antares muestra al actor, una bandeja de trabajo, con la información de los requerimientos resultantes de la consulta.
 3. El actor ingresa al detalle del requerimiento, pantalla en la que puede realizar las acciones de adjuntar archivo de evidencia y/o comentario; así como cancelar el requerimiento.
 4. Cuando un requerimiento es atendido satisfactoriamente, y es sometido a un periodo de prueba establecido por la GCSR, el actor puede establecer el estatus de CERRADO con Visto Bueno, lo cual indica que el término del proceso de seguimiento para ese folio.
- e. Reglas de negocio:
1. Solo se podrán gestionar requerimientos, cuyo estatus sea distinto a CERRADO o CANCELADO.

2. Las notificaciones vía correo electrónico se realizan solamente para el cambio de estatus en el requerimiento y no para el resto de las acciones de seguimiento del mismo.
3. Adjuntar información adicional al requerimiento es opcional, para todos los actores de este caso de uso.
4. Los cambios de estatus se realizan con base al avance registrado al requerimiento. Antares deberá validar esta situación.
5. Cada acción emprendida para el seguimiento del requerimiento, deberá quedar registrada en una bitácora.

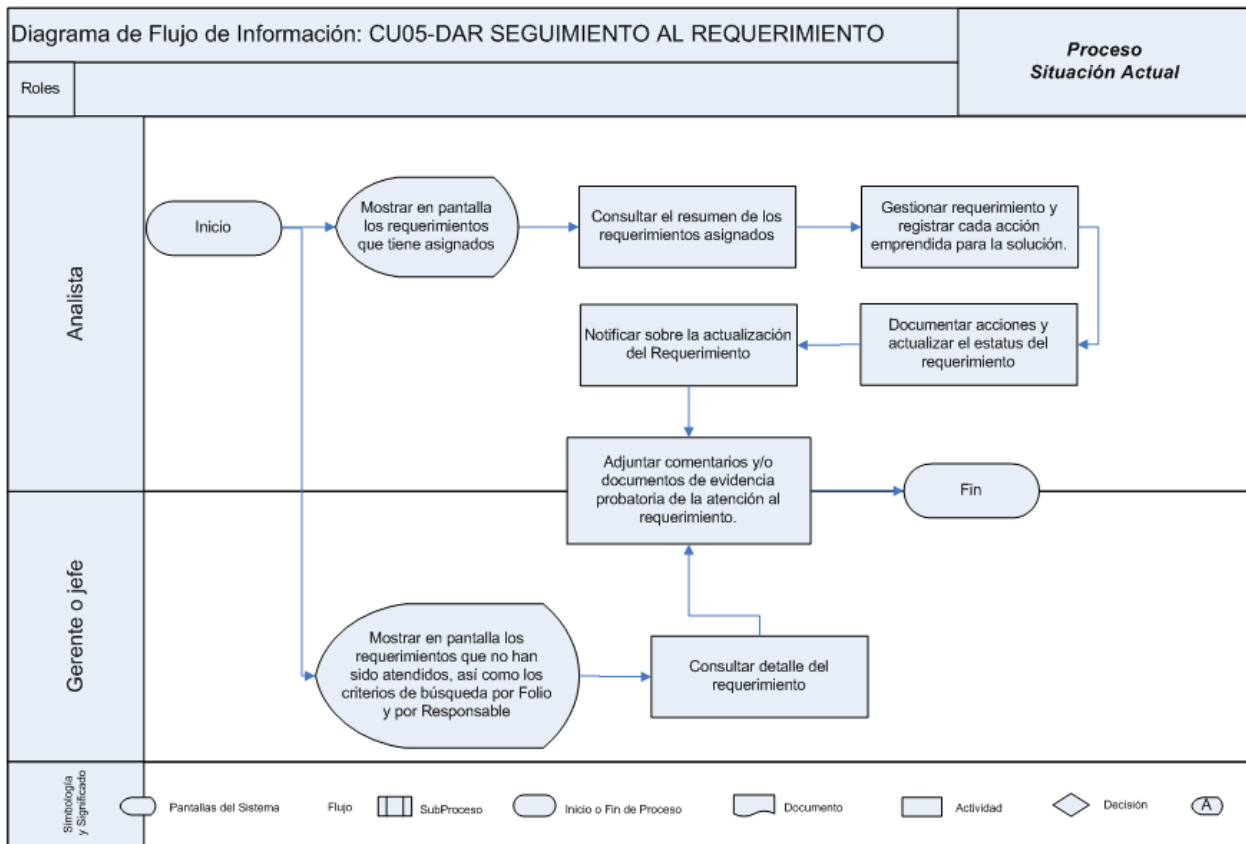
f. Precondiciones:

N/A

g. Pos condiciones:

N/A

h. Diagrama de Actividades:



3.12 Diagrama de actividades CU05-Dar seguimiento al requerimiento

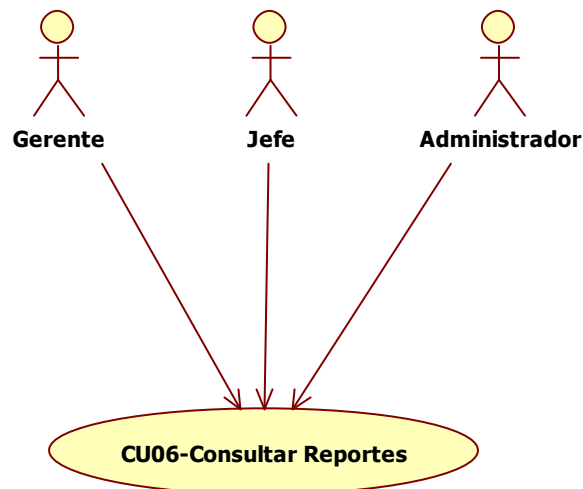
CU06-CONSULTAR REPORTES

Fig. 3.13 Diagrama del Caso de Uso CU06-Consultar Reportes

- a. Descripción breve: Consultar reportes planeados que permitan una mejor planeación de la atención de los proyectos, así como un mejor balance en la carga de trabajo por cada departamento.
- b. Actores involucrados:
- Administrador
 - Gerente
 - Jefe
- c. Flujo básico:
1. El caso de uso inicia cuando el actor <Administrador, gerente o jefe> previamente autenticado en el sistema Antares, selecciona del menú principal la opción "Cuentas Corporativas" y posteriormente invoca la opción "Reportes".
 2. A continuación el sistema Antares deberá mostrarle en pantalla al actor, las opciones disponibles de reportes planeados:
 - Número de Requerimientos atendidos por cada usuario.
 - Número de Requerimientos por cada estatus.
 - Requerimientos asignados/atendidos por departamento.
 3. El actor elige la opción y realiza la consulta de los requerimientos solicitados.

4. El actor puede consultar el detalle de un requerimiento en particular y realizar la gestión correspondiente a su perfil de usuario.
5. Finaliza el caso de uso.

d. Flujos alternos:

N/A

e. Reglas de negocio:

N/A

f. Precondiciones:

N/A

g. Pos condiciones:

N/A

h. Diagrama de Actividades:

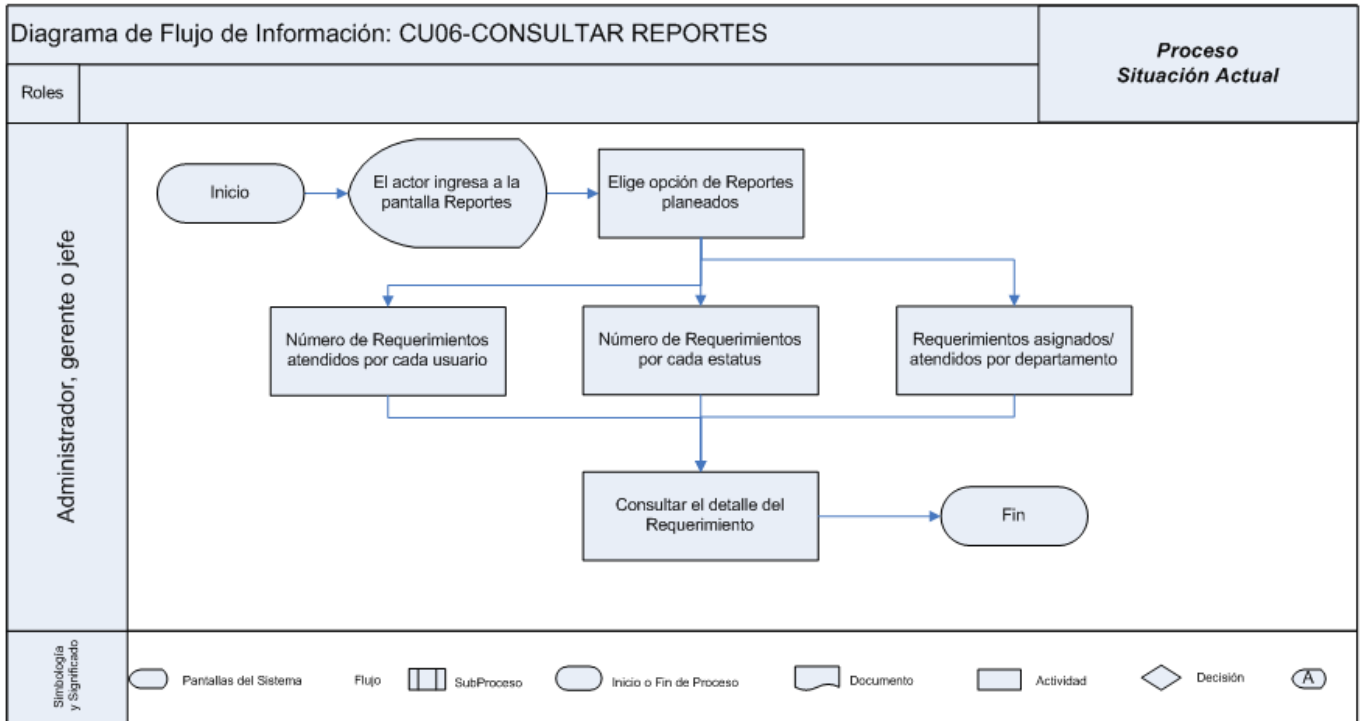


Fig. 3.14 Diagrama de actividades CU06-Consultar Reportes

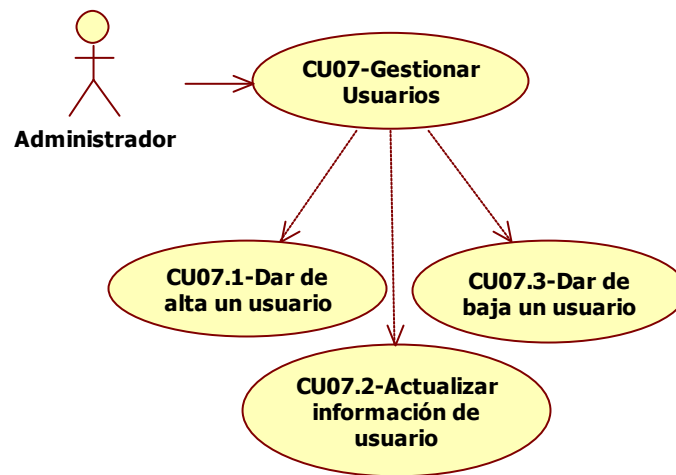
CU07-GESTIONAR USUARIOS

Fig. 3.15 Diagrama del Caso de Uso CU07-Gestionar usuarios

- a. Descripción breve: La gestión de usuarios consiste en permitir actualizar información de los usuarios como su perfil o cambio de contraseña, así como generar usuarios y eliminarlos permanentemente del sistema.
- b. Actores involucrados:
- Administrador
- c. Flujo básico:
1. El caso de uso inicia cuando el actor <Administrador > previamente autenticado en el sistema Antares, selecciona del menú principal la opción “Cuentas Corporativas” y posteriormente invoca la opción “Usuarios”.
 2. A continuación el sistema Antares deberá mostrarle en pantalla al actor, las opciones disponibles para la gestión de usuarios:
 - Agregar un nuevo usuario: Se ejecuta lo correspondiente al Flujo alternativo 1 <Agregar un nuevo usuario>.
 - Editar la información de un usuario existente: Se ejecuta lo correspondiente al Flujo alternativo 2 <Editar usuario>.
 - Eliminar permanentemente a un usuario de Antares: Se ejecuta lo correspondiente al Flujo alternativo 1 <Eliminar usuario>.
 3. El actor confirma la acción realizada sobre el usuario.
 4. Finaliza el caso de uso.

d. Flujos alternos:

- Flujo alternativo 1 <Agregar un nuevo usuario>:
 1. Antares muestra al actor un formulario para la captura de la información indispensable para el alta del usuario:
 - a. Nombre completo
 - b. Identificador asignado por la GCSR9
 - c. Departamento
 - d. Número de empleado
 - e. Puesto
 - f. Correo electrónico
 2. El actor confirma el alta del usuario en Antares.
- Flujo alternativo 2 <Editar usuario>:
 1. Antares muestra al actor un formulario que presenta la información actual del usuario, misma que podrá ser editable, salvo el campo del Identificador asignado por la GCSR9, ya que este es único.
 2. El actor confirma la actualización del usuario en Antares.
- Flujo alternativo 3 <Eliminar usuario>:
 1. El actor debe seleccionar de una lista, al usuario a eliminar del sistema Antares.
 2. El actor confirma la eliminación del usuario en Antares.

e. Reglas de negocio:

1. Debe existir un usuario administrador encargado de la recepción y asignación de requerimientos. Este usuario es vital en el modo de operación de la GCSR9, en lo que respecta a la atención a cuentas corporativas.
2. Los perfiles de Gerente, Analista de Optimización de Red y Jefes de área deben corresponder con sus respectivos departamentos, esta validación es indispensable para evitar la duplicidad de perfiles, así como la redundancia en las acciones efectuadas durante el seguimiento de requerimientos.
3. Al eliminar un usuario de Antares, se perderán los privilegios de acceso al mismo, sin embargo, las gestiones realizadas por el mismo, sobre los requerimientos a su cargo persistirán en la base de datos de la GCSR9 a modo de información histórica.

f. Precondiciones:

N/A

g. Pos condiciones:

N/A

h. Diagrama de Actividades:

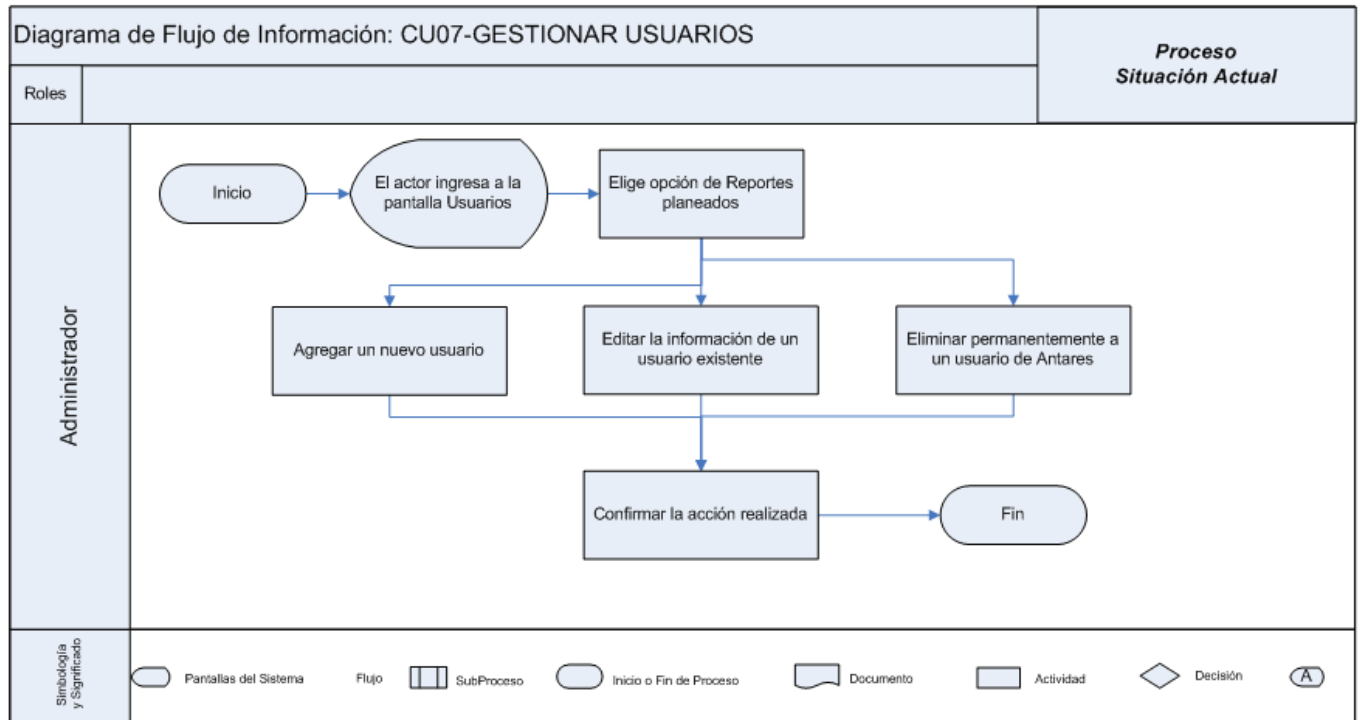


Fig. 3.16 Diagrama de actividades CU07-Gestionar usuarios

3.4 DISEÑO

Una vez finalizada la etapa de análisis, se tiene como resultado el modelo de análisis descrito en el apartado anterior mediante el cual, se logra la comprensión detallada de los requisitos de la GCSR9. A continuación, en la etapa de diseño, se modela en su totalidad el sistema Antares obteniendo como resultado, la identificación de los requerimientos funcionales que serán necesarios implementarse en la solución propuesta; es decir, es esta etapa se construye la entrada apropiada y que será el punto partida para las actividades de la implementación.

Concretamente, esta etapa persigue los siguientes objetivos:

- Proporciona el diseño de la interfaz gráfica de usuarios y reportes.
- Definir la funcionalidad del nuevo sistema de la GCSR9, por lo que es la referencia fundamental de la funcionalidad para la construcción de Antares, las pruebas y los criterios de aceptación final.
- Proporciona la especificación detallada de los casos de uso.
- Sirve como entrada para la elaboración de un manual de usuarios y una guía de capacitación.

El resto de esta sección contiene en primer lugar algunas características generales del sistema y en seguida, por cada módulo de Antares contiene las páginas web que lo conforman, las reglas del proceso de negocio subyacente. En estos últimos se ha puesto el énfasis en el flujo de la interacción usuario – sistema y no se han detallado los casos de uso del módulo de reportes dado que el contenido de estos es bastante familiar para los usuarios, esto es, no requieren mayor explicación, y desde el punto de vista de desarrollo, se cuenta con una carpeta con todos los formatos que la GCSR9 ha proporcionado.

3.4.1 DIAGRAMAS DE DISEÑO DEL MÓDULO DE CUENTAS CORPORATIVAS

- Diagrama de clases: Su finalidad es mostrara un conjunto de clases, interfaces, colaboraciones que existen en el contexto del sistema Antares; así como las relaciones entre estos. “Los diagramas de clases muestran el diseño de un sistema desde el punto de vista estático y es a su vez, un diagrama que muestra una colección de elementos estáticos declarativos”[9].

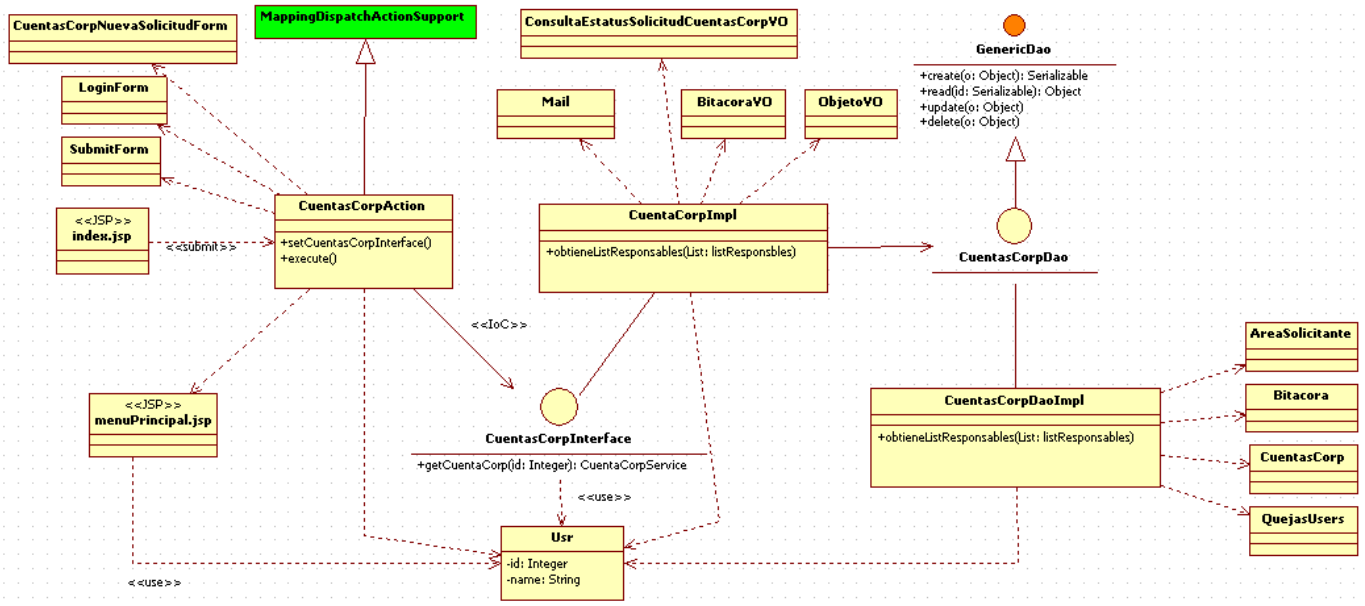


Fig. 3.17 Diagrama de clases del módulo Cuentas Corporativas.

- Diagramas de interacción: Con base en la especificación de UML tanto los diagramas de secuencia como de colaboración son tipos de diagramas de interacción; particularmente para el desarrollo del sistema Antares se emplearon diagramas de secuencia. Es mediante estos diagramas que, se muestra una interacción ordenada según la secuencia temporal de eventos, para cada actividad permitida dentro del módulo de Cuentas Corporativas. En particular, muestran los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado.

1. Diagrama de secuencia para el escenario principal del caso de uso CU01: Autenticar.

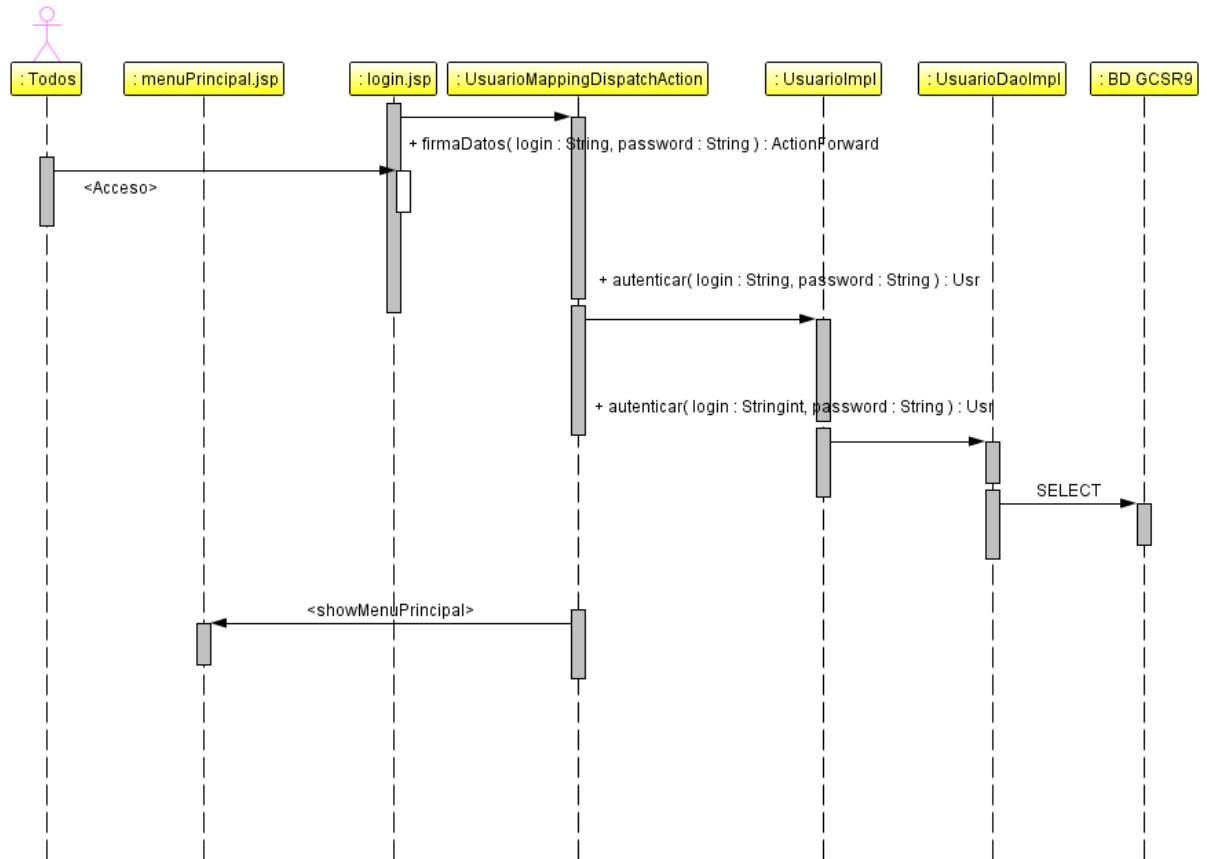


Fig. 3.18 Diagrama de secuencia: CU01-Autenticar.

2. Diagrama de secuencia para el escenario principal del caso de uso CU02: Generar Solicitud de Servicio.

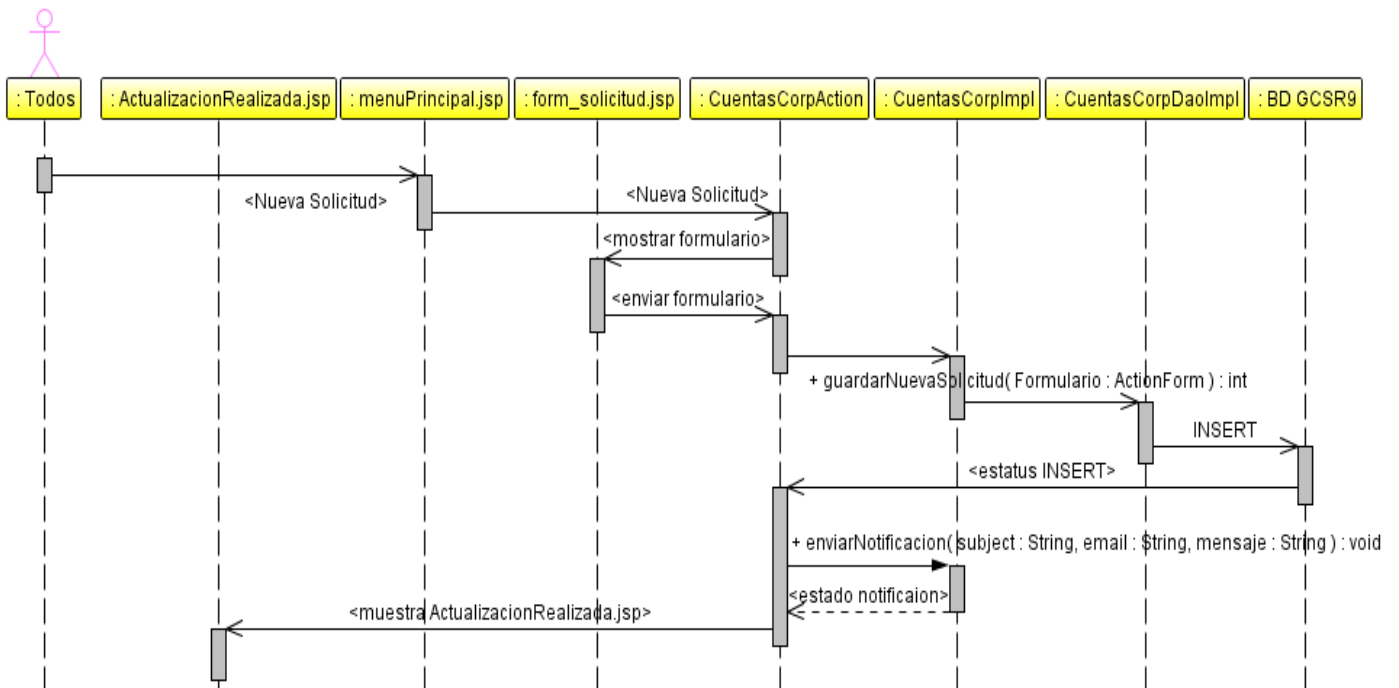


Fig. 3.19 Diagrama de secuencia: CU02-Generar Solicitud de Servicio.

3. Diagrama de secuencia para el escenario principal del caso de uso CU03: Consultar estatus de requerimientos.

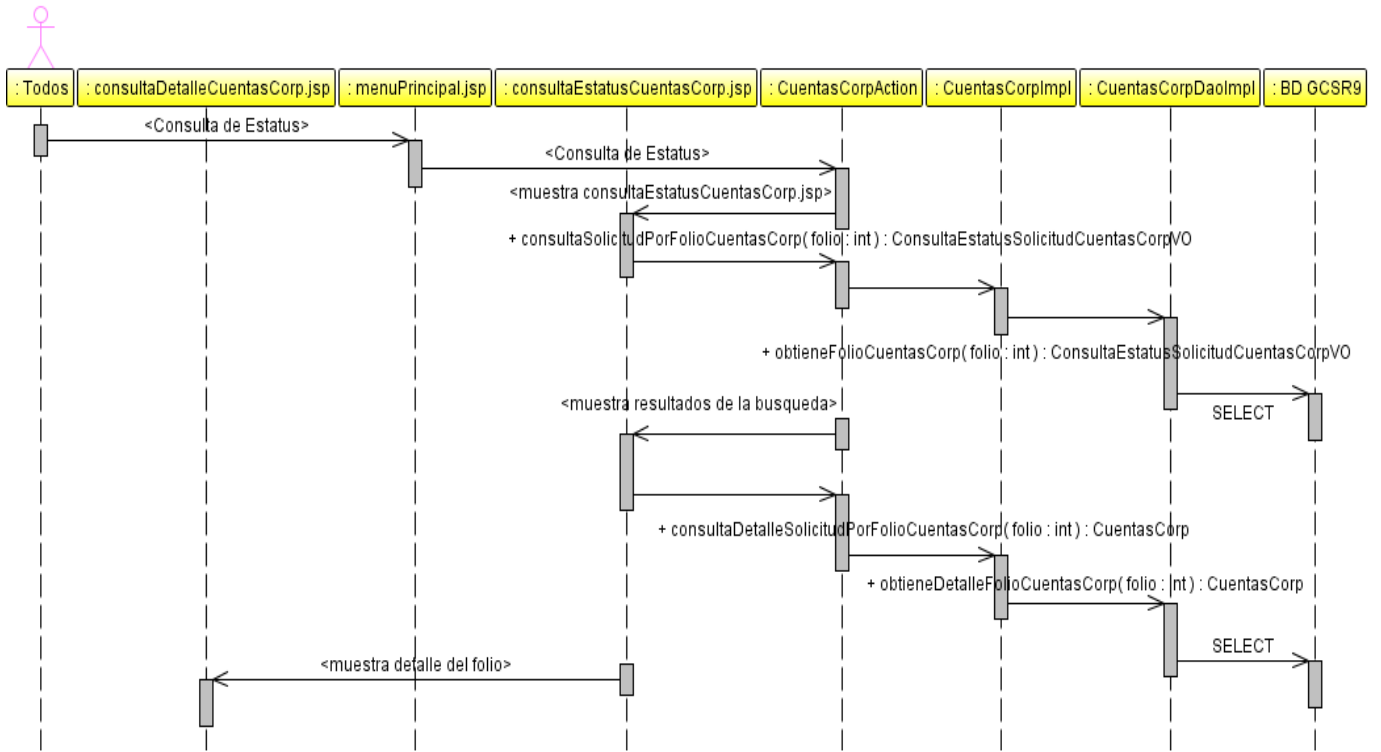


Fig. 3.20 Diagrama de secuencia: CU03-Consultar estatus de requerimientos.

4. Diagrama de secuencia para el escenario principal del caso de uso CU04: Asignar responsable al requerimiento.

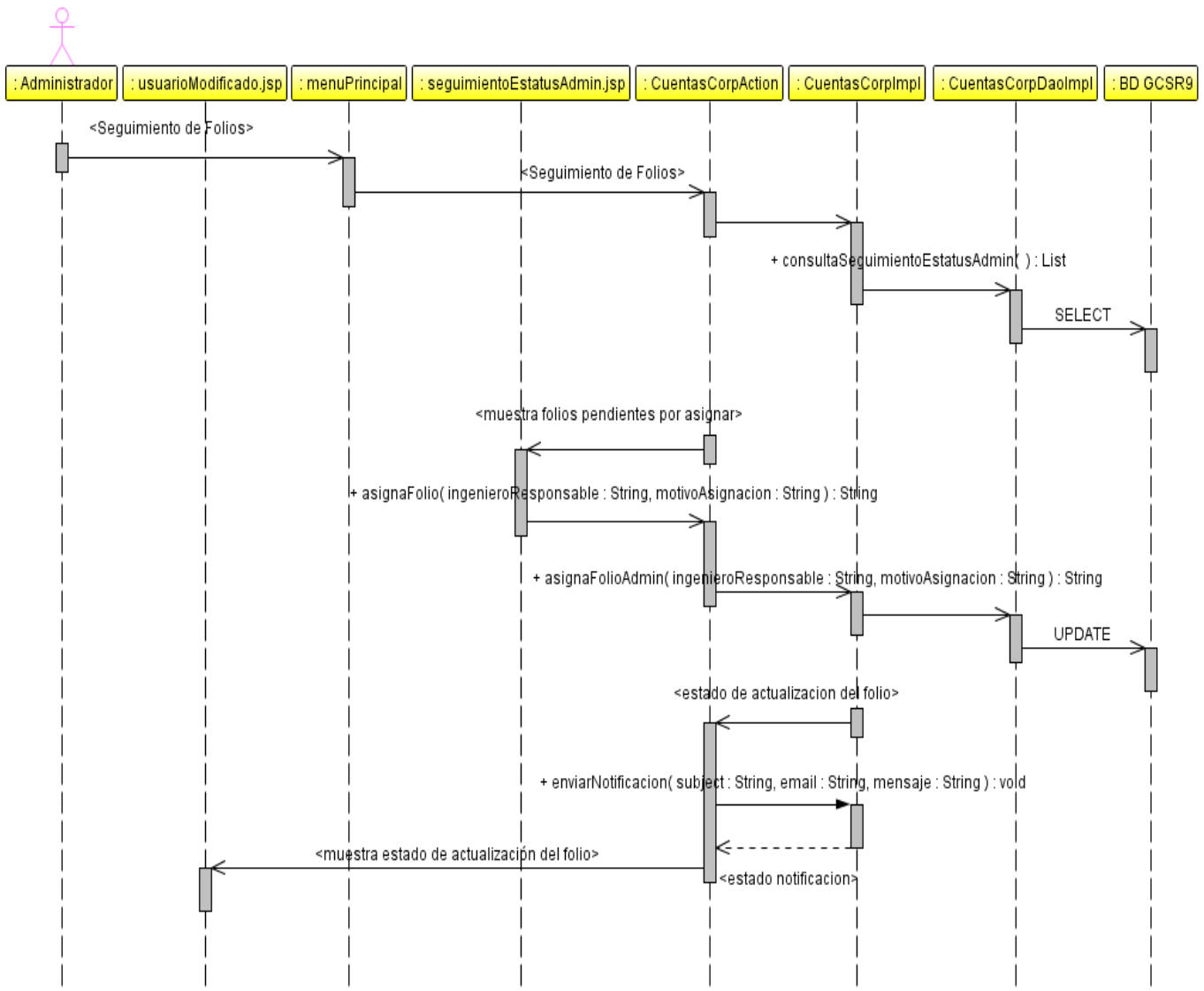


Fig. 3.21 Diagrama de secuencia: CU04-Asignar responsable al requerimiento.

5. Diagrama de secuencia para el escenario principal del caso de uso CU05: Dar seguimiento al requerimiento.

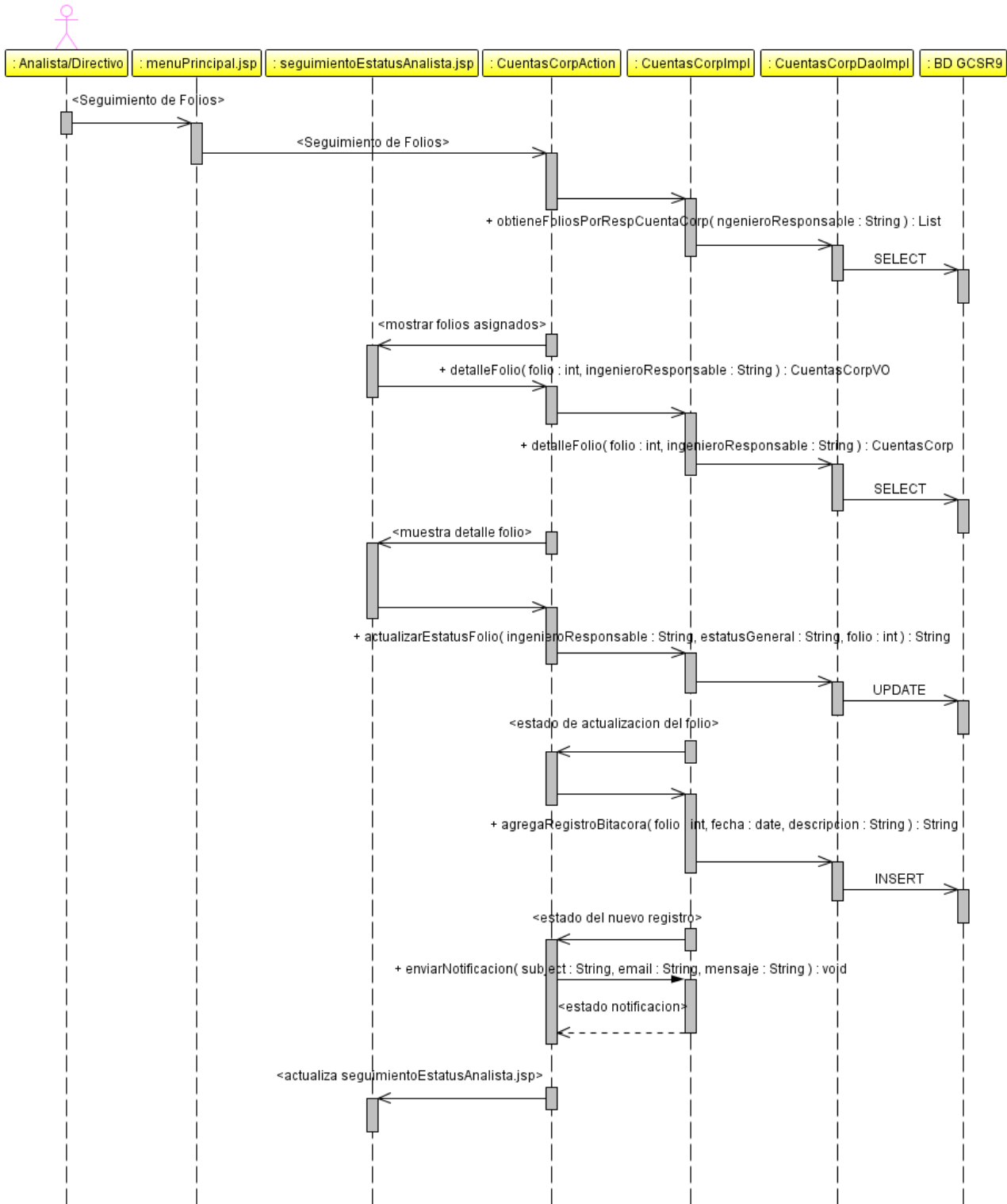


Fig. 3.22 Diagrama de secuencia: CU05-Dar seguimiento al requerimiento.

6. Diagrama de secuencia para el escenario principal del caso de uso CU06: Consultar reportes.

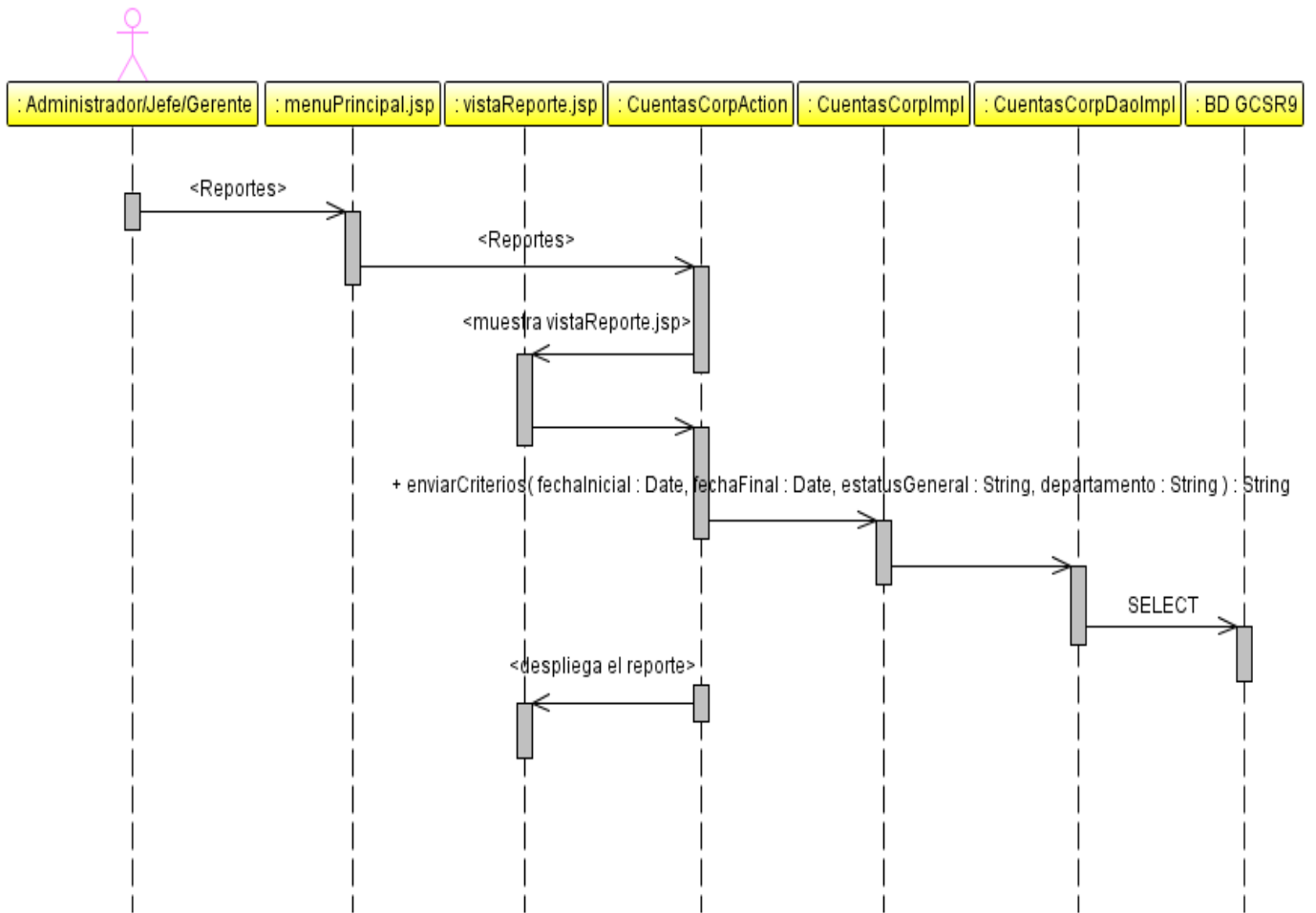


Fig. 3.23 Diagrama de secuencia: CU06-Consultar reportes.

7. Diagrama de secuencia para el escenario principal del caso de uso CU07: Gestionar usuarios → Agregar usuario.

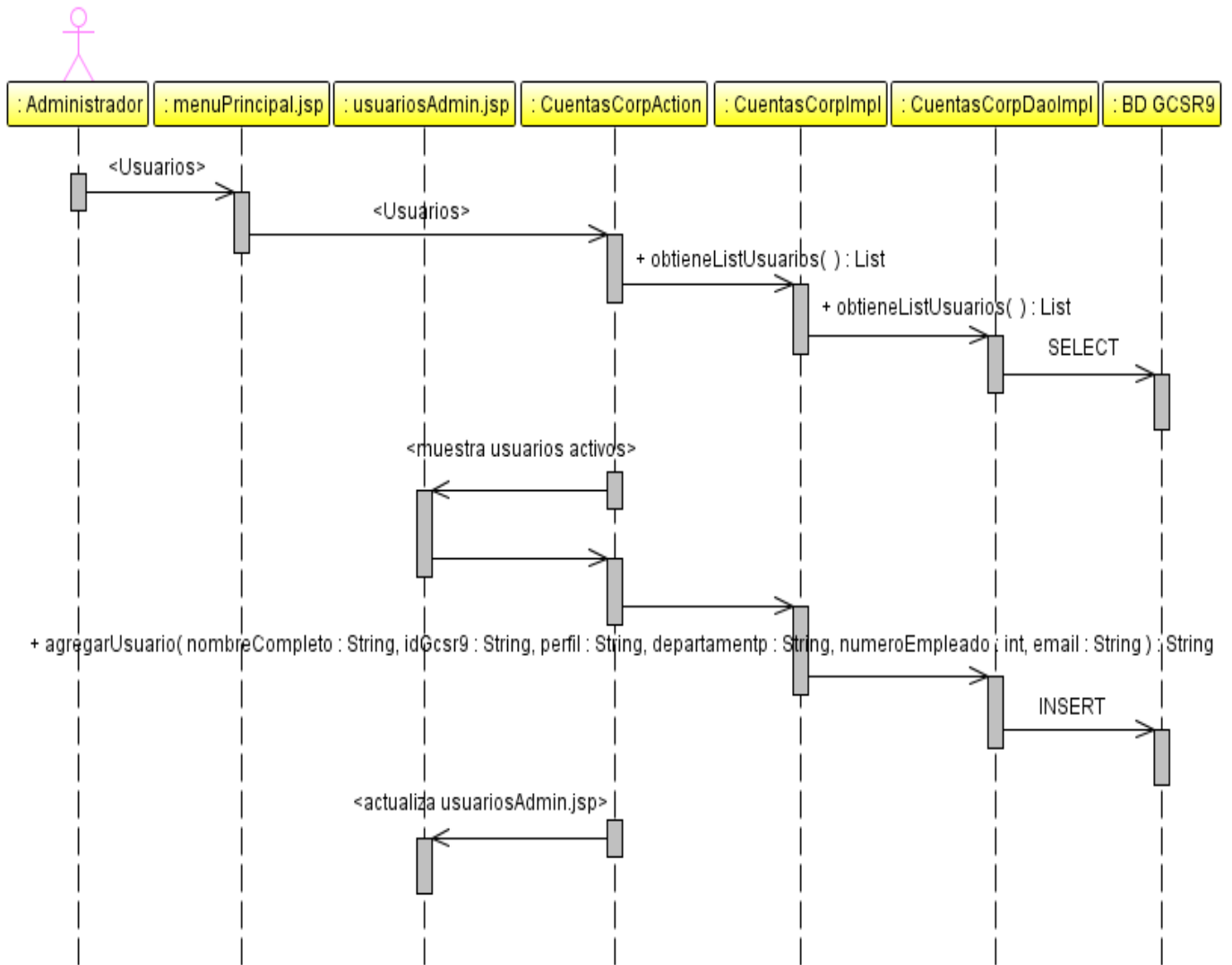
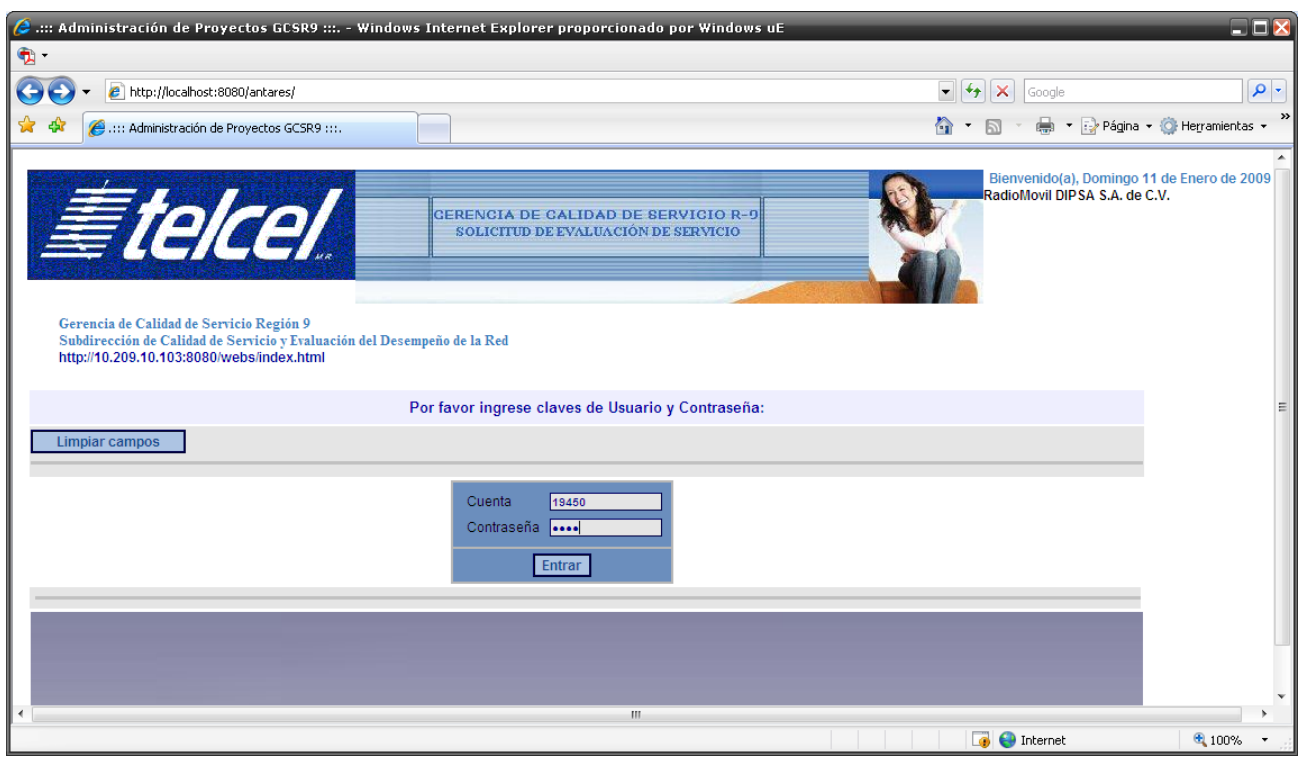


Fig. 3.24 Diagrama de secuencia: CU07-Gestionar usuarios.

3.4.2 ESPECIFICACIÓN DEL DISEÑO DE LOS CASOS DE USO

CU01-AUTENTICAR

Nombre	Inicio de sesión.
Descripción	Permite el acceso al sistema utilizando autenticación integrada.
Pantalla	
	

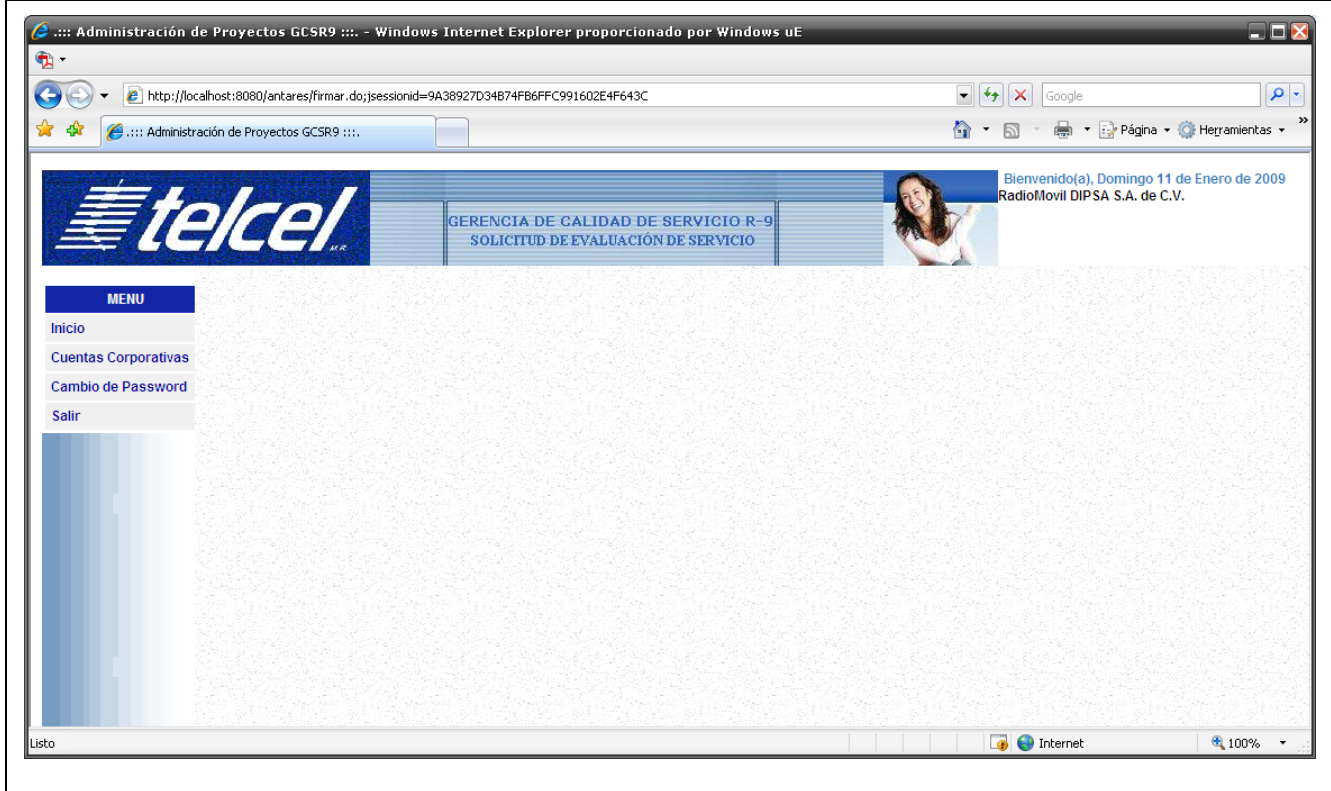
Reglas funcionales:

Clave	Especificaciones
ANT_RF_001	Para fines de seguridad de la información propiedad de Radio Móvil DIPSA, el usuario <Empleado general Telcel, Analista, administrador, gerente o jefe> debe ingresar sus claves de acceso (cuenta y contraseña) en la URL http://localhost:8080/antares/ mediante el navegador de Microsoft Internet Explorer 6 (o posterior).
ANT_RF_002	El actor que desee ingresar al sistema Antares, deberá estar registrado en la base de datos de la GCSR9, también deberá contar con un puesto y deberá estar

Clave	Especificaciones
	asignado a un departamento reconocido por Radio Móvil DIPSA.

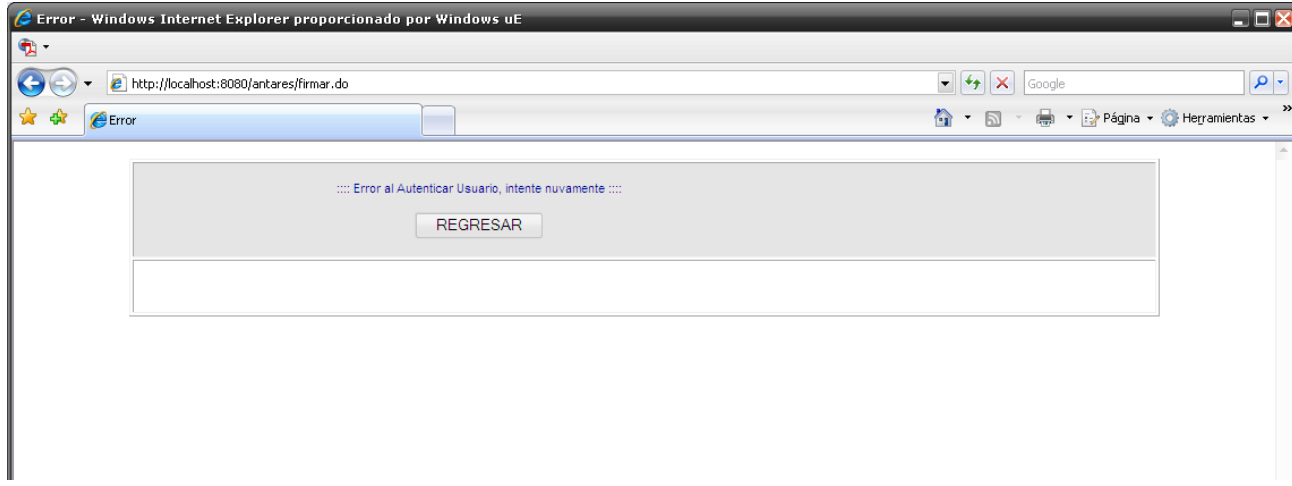
Nombre	Menú principal.
Descripción	Son las opciones del módulo Cuentas Corporativas.

Pantalla



Reglas funcionales:

Clave	Especificaciones
ANT_RF_003	El sistema deberá mostrar las opciones de menú de acuerdo al perfil de cada usuario. El menú principal muestra las opciones Inicio, Cuentas Corporativas, Cambio de Password y Salir.
ANT_RF_004	El sistema deberá asignar al usuario solamente los privilegios que le correspondan a su rol. Debe validar que no tenga privilegios o funciones que no sean propias de su perfil

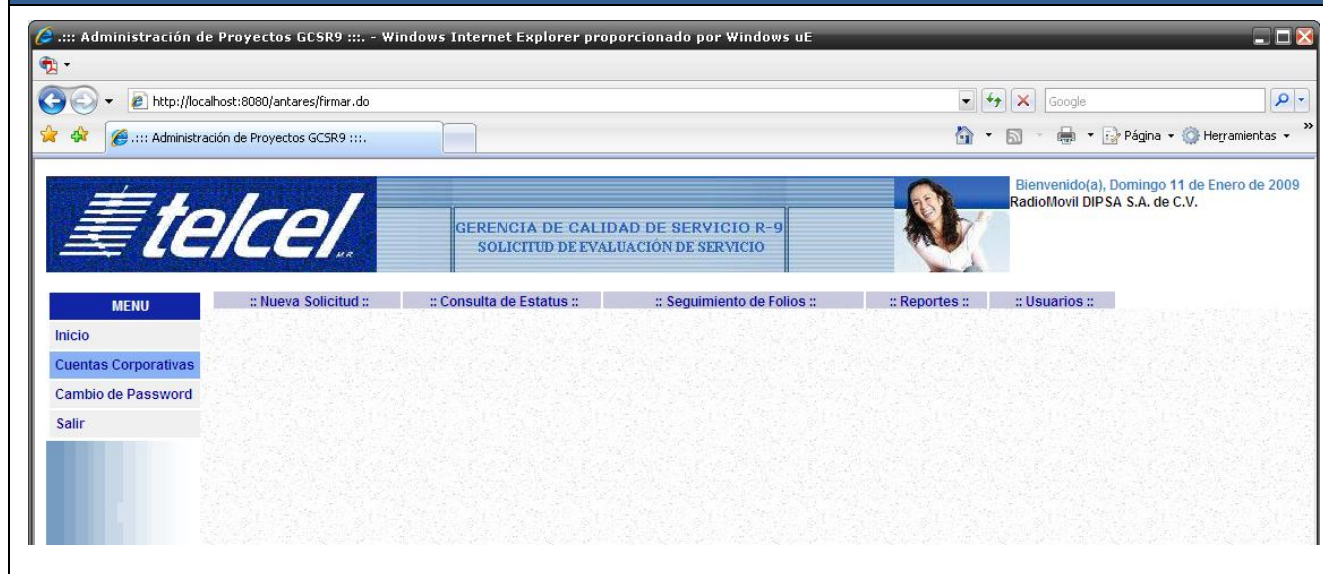
Nombre	Error de Autenticación.
Descripción	Es la vista que muestra el sistema cuando no reconoce al usuario que intenta ingresar.
Pantalla	
	

Reglas funcionales:

Clave	Especificaciones
ANT_RF_005	Si los datos ingresados por el usuario son inválidos, el sistema redirecciona a la página de “Error de Autenticación”, en la que mostrará el siguiente mensaje “Error al autenticar usuario, intente nuevamente”.
ANT_RF_006	Mediante el botón REGRESAR, el sistema deberá permitir el regreso a la pantalla de inicio de sesión para que el actor pueda intentar el acceso al sistema Antares una vez más.

Nombre	Menú Cuentas Corporativas.
Descripción	Pantalla que muestra las opciones que se tienen disponibles en el módulo de Cuentas Corporativas.

Pantalla



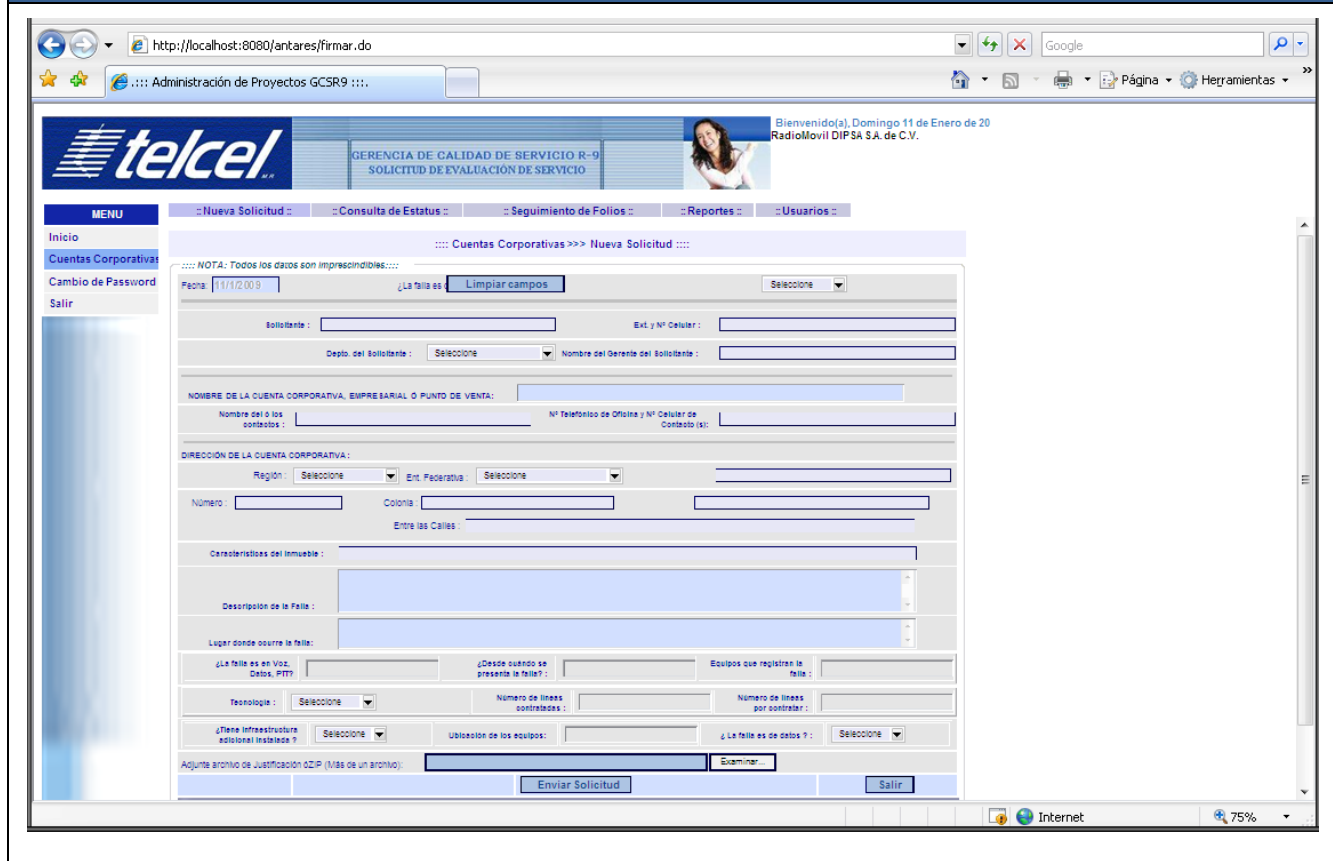
Reglas funcionales:

Clave	Especificaciones
ANT_RF_007	Esta pantalla muestra las opciones Nueva Solicitud, Consulta de Estatus, Seguimiento de Folios, Reportes y Usuarios. Dichas opciones están habilitadas dependiendo del perfil de usuario.
ANT_RF_008	Perfil administrador: Tiene habilitadas las funciones de Nueva Solicitud, Consulta de Estatus, Seguimiento de Folios, Reportes y Usuarios.
ANT_RF_009	Perfil Jefe: Tiene habilitadas las funciones de Nueva Solicitud, Consulta de Estatus, Seguimiento de Folios y Reportes.
ANT_RF_010	Perfil Gerente: Tiene habilitadas las funciones de Nueva Solicitud, Consulta de Estatus, Seguimiento de Folios y Reportes.
ANT_RF_011	Perfil Empleado General de Telcel: Tiene habilitadas las funciones de Nueva Solicitud y Consulta de Estatus.
ANT_RF_012	Perfil Analista de Optimización de Red Región 9: Tiene habilitadas las funciones de Nueva Solicitud, Consulta de Estatus y Seguimiento de Folios.

CU02-GENERAR SOLICITUD DE SERVICIO

Nombre	Nueva solicitud.
Descripción	Pantalla que muestra el formulario a llenar por parte del usuario, para solicitar atención a una cuenta corporativa.

Pantalla



Reglas funcionales:

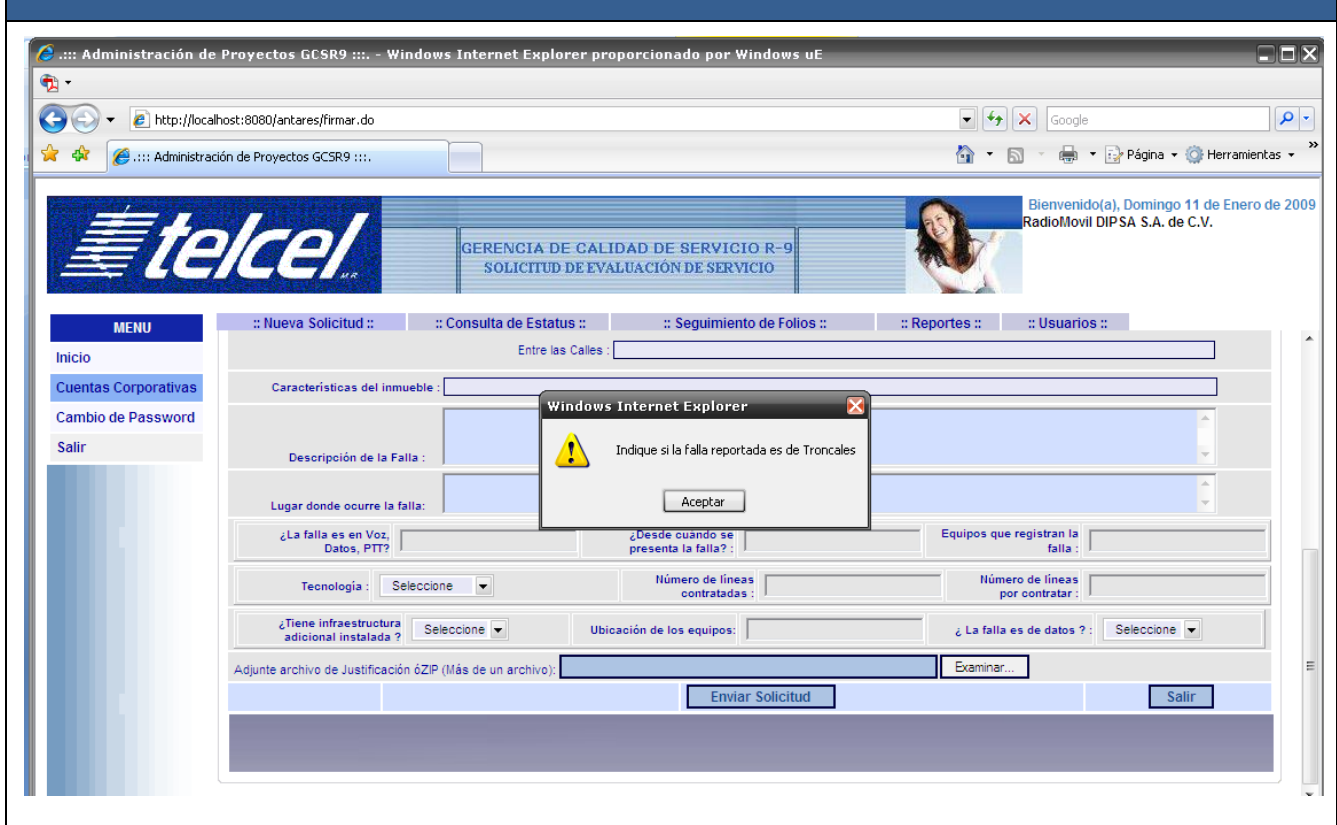
Clave	Especificaciones
ANT_RF_013	El usuario <Empleado general Telcel, Analista, administrador, gerente o jefe> previamente autenticado en el sistema Antares, selecciona del menú principal la opción “Cuentas Corporativas” y posteriormente invoca la generación de una “Nueva Solicitud”, con lo que se muestra la pantalla del formulario.
ANT_RF_014	Para generar una Solicitud de Servicio para atención de una Cuenta Corporativa, es necesario capturar la información requerida por el formulario en la que se describirá la falla en el servicio proporcionado por Radio Móvil DIPSА.

Clave	Especificaciones
ANT_RF_015	<p>Los campos a capturar son los siguientes:</p> <ol style="list-style-type: none"> 1. Fecha: Este campo se llena automáticamente y se refiere a la fecha en la que se realiza la solicitud a la GCSR9. 2. ¿La falla es de Troncales Celulares? 3. Solicitante 4. Teléfono extensión y N° Celular 5. Depto. del Solicitante 6. Nombre del Gerente del Solicitante 7. Nombre de la cuenta corporativa 8. Nombre del contacto con la cuenta corporativa 9. N° Telefónico de Oficina y N° Celular de Contacto 10. Región 11. Entidad Federativa 12. Calle 13. Número 14. Colonia 15. Del. / Municipio 16. Entre las Calles 17. Características del inmueble 18. Descripción de la Falla 19. Lugar donde ocurre la falla 20. ¿La falla es en Voz, Datos, PTT? 21. ¿Desde cuándo se presenta la falla? 22. Equipos que registran la falla 23. Tecnología 24. Número de líneas contratadas 25. Número de líneas por contratar 26. ¿Tiene infraestructura adicional instalada? 27. Ubicación de los equipos 28. ¿La falla es de datos?
ANT_RN_016	<p>El usuario puede si así lo desea, adjuntar a la solicitud de atención, información adicional que funja como evidencia para sustentar la falla reportada a la GCSR9.</p>

Clave	Especificaciones
ANT_RN_017	Adjuntar información adicional al requerimiento es opcional, y no impide la generación de la solicitud de atención a la cuenta corporativa, si el resto de los campos están completos y correctos.
ANT_RN_018	Al capturar toda la información solicitada por el formulario, el usuario debe invocar la generación de la solicitud de atención mediante el botón "Enviar Solicitud".
ANT_RN_019	Una vez generada la solicitud de atención, Antares deberá notificar al usuario Administrador por medio de un correo electrónico, sobre la creación del nuevo requerimiento.

Nombre	Captura incorrecta de datos del formulario.
Descripción	Muestra un mensaje de alerta indicándole al usuario el campo ingresado incorrectamente.

Pantalla

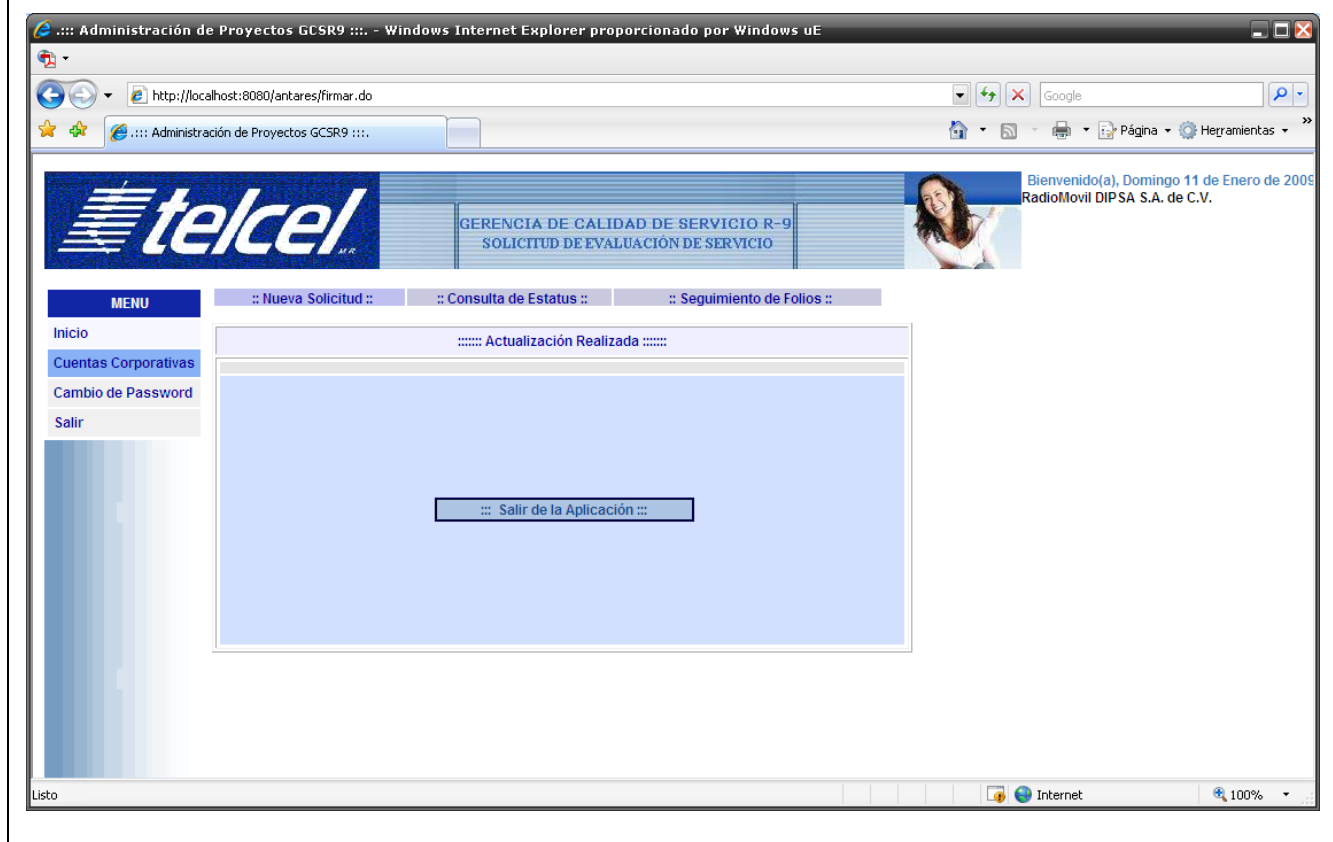


Reglas funcionales:

Clave	Especificaciones
ANT_RN_020	Si alguno de los datos ingresados por el usuario al formulario es inválido o no ha sido capturado por el usuario, el sistema Antares enviará un mensaje de alerta indicando el campo en conflicto.
ANT_RN_021	Al dar clic en el botón de “Aceptar”, deberá permitir la corrección del campo y seguir adelante con la generación de la solicitud de atención.

Nombre	Pantalla de Confirmación de Solicitud Generada.
Descripción	Muestra un mensaje de confirmación indicándole al usuario que se generó exitosamente la solicitud de atención a la cuenta corporativa correspondiente.

Pantalla

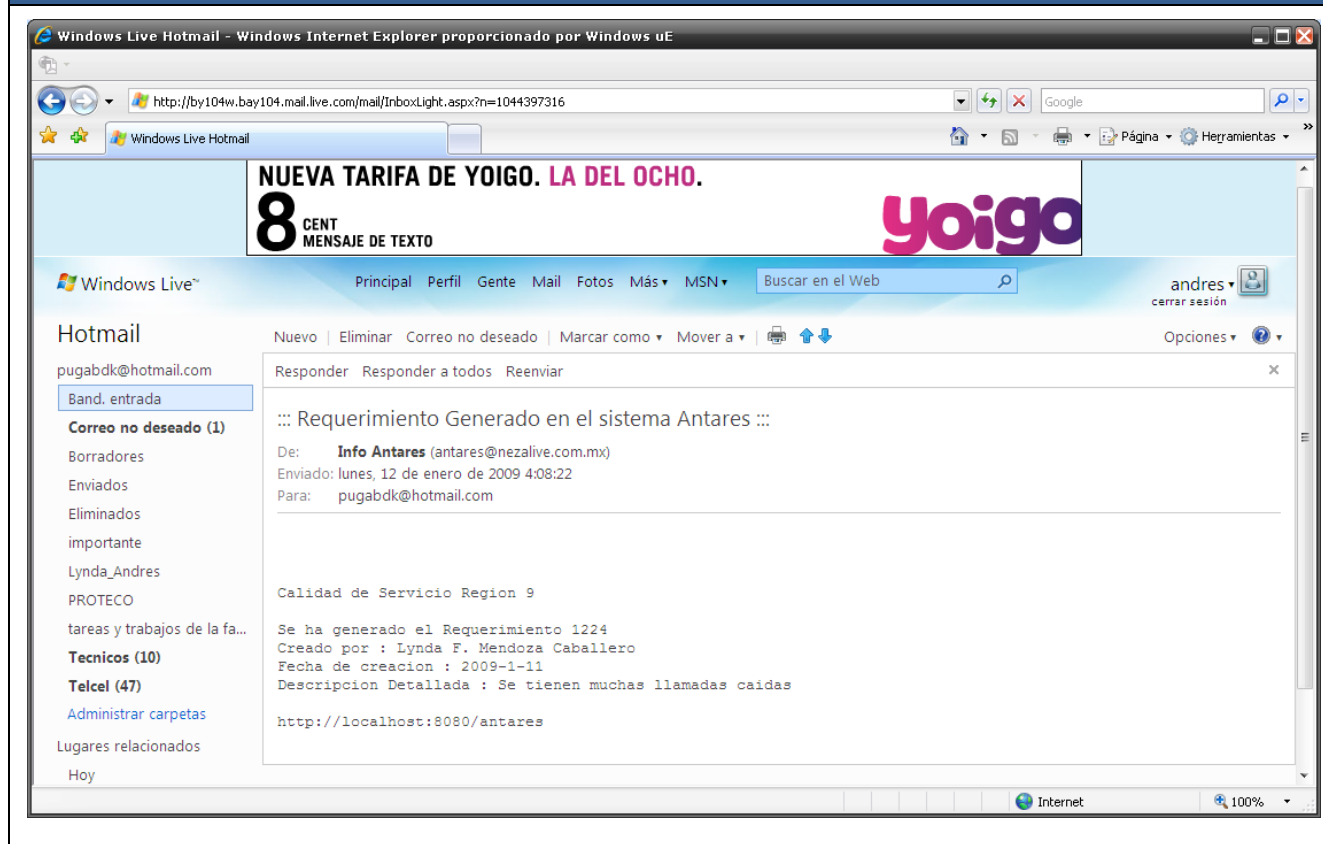


Reglas funcionales:

Clave	Especificaciones
ANT_RN_022	El usuario ha generado exitosamente la solicitud de atención a la cuenta corporativa, por lo que puede continuar trabajando con el sistema o bien, salir de la aplicación.

Nombre	Pantalla de notificación de Solicitud Generada.
Descripción	Notificación vía correo electrónico a la GCSR9, de que se generó una solicitud para atender a una cuenta corporativa en convenio.

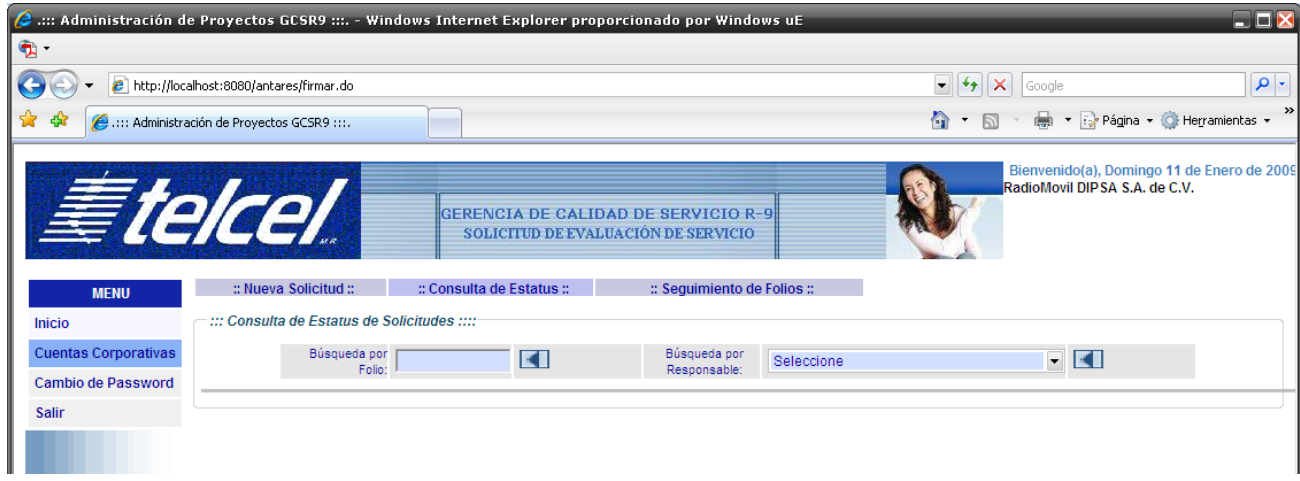
Pantalla



Reglas funcionales:

Clave	Especificaciones
ANT_RN_023	Cuando es generado el nuevo requerimiento, se debe enviar un correo electrónico de notificación, al administrador de Antares quien a su vez es responsable de analizar el caso y comenzar con la gestión y el seguimiento del requerimiento. La notificación debe ser automática ya que los tiempos de respuesta en la atención a clientes son de vital importancia para la GCSR9 y en general para Radio Móvil DIPSA.

CU03-CONSULTAR ESTATUS DE REQUERIMIENTOS

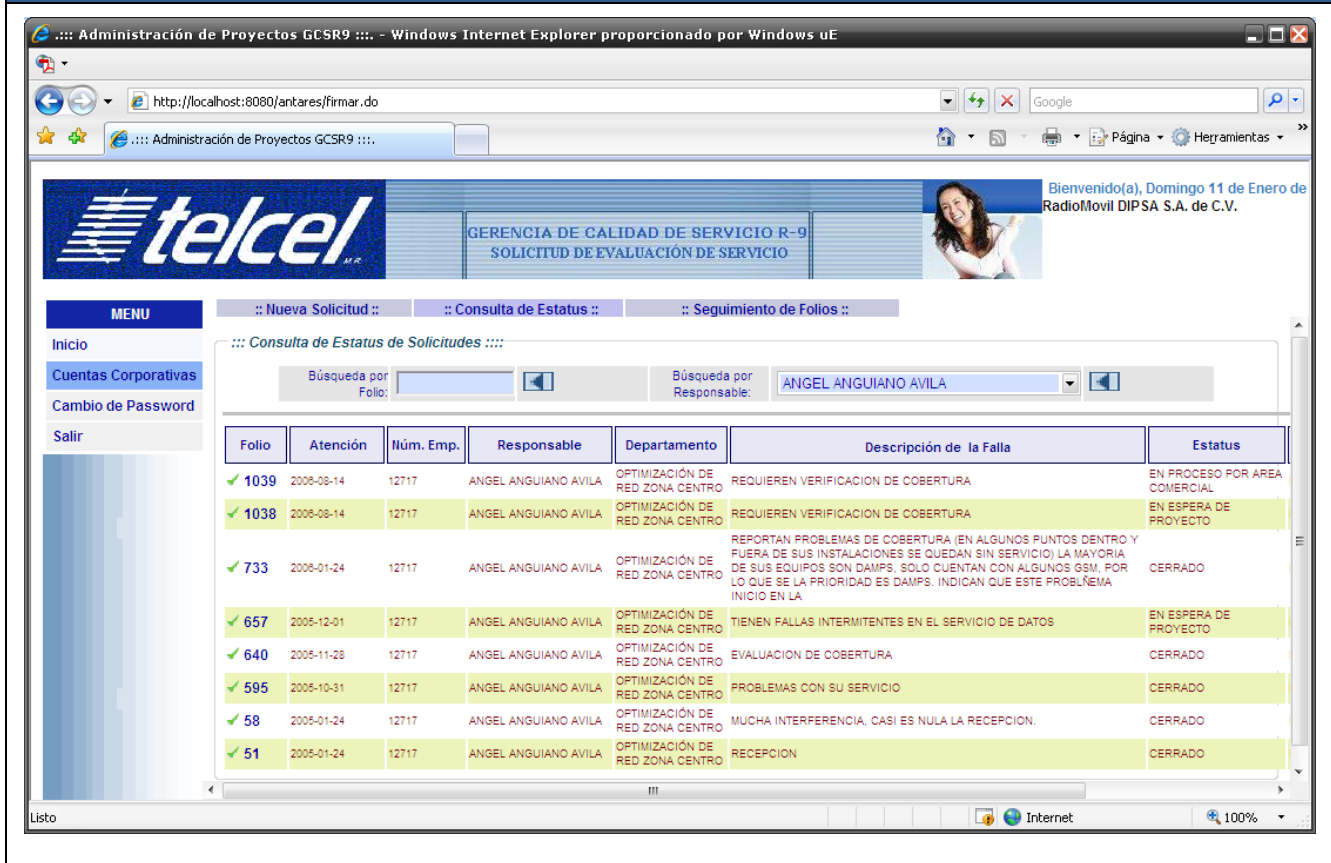
Nombre	Pantalla de Consulta de estatus de las solicitudes.
Descripción	Muestra las opciones disponibles para la búsqueda de requerimientos.
Pantalla	
	

Reglas funcionales:

Clave	Especificaciones
ANT_RN_024	El usuario <Empleado general Telcel, Analista, administrador, gerente o jefe> previamente autenticado en el sistema Antares, selecciona del menú principal la opción “Cuentas Corporativas” y posteriormente invoca la opción de “Consulta de Estatus”.
ANT_RN_025	A continuación el actor deberá seleccionar el criterio de búsqueda del Requerimiento: <ol style="list-style-type: none"> 1. Por folio: El usuario podrá realizar la búsqueda por el número del folio asignado al requerimiento en cuestión. 2. Por responsable: El usuario podrá realizar la búsqueda al seleccionar de una lista, al usuario responsable (Analista de Optimización de Red Región 9) de la atención del requerimiento.

Nombre	Resumen de requerimientos.
Descripción	Muestra el resumen de los requerimiento que cumplen con el criterio de búsqueda.

Pantalla

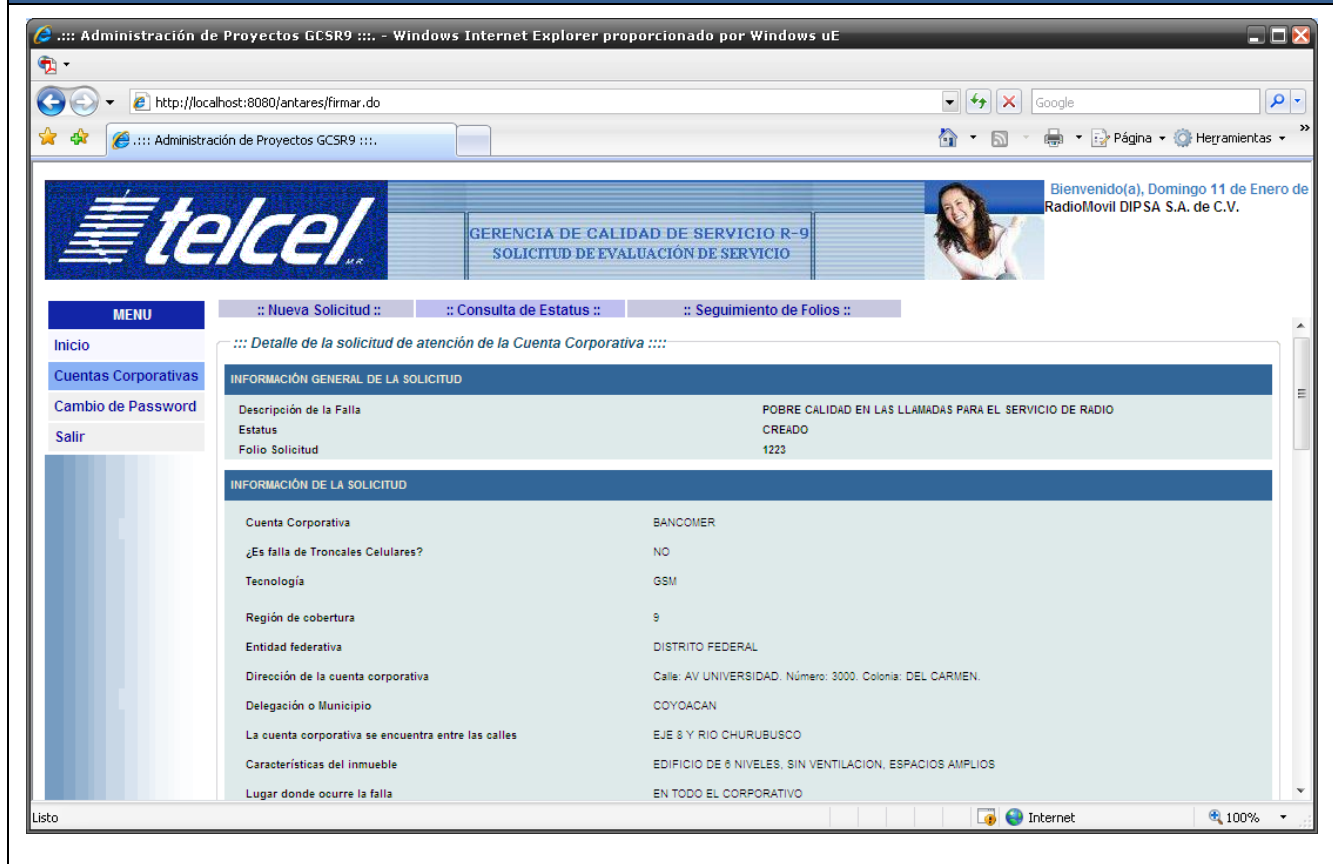


Reglas funcionales:

Clave	Especificaciones
ANT_RN_026	Los datos mostrados deberán incluir el folio, la información del empleado que está atendiendo el requerimiento, la descripción del error o falla reportada por la cuenta corporativa y el estatus.
ANT_RN_027	El campo "Folio" permite que en cada registro sea posible consultar a detalle las acciones realizadas por las áreas involucradas en la atención del requerimiento en cuestión. Dichas acciones serán coordinadas por la GCSR9, área que en todo momento dará seguimiento al proceso de solución de los requerimientos.

Nombre	Consulta detalle del requerimiento.
Descripción	Muestra el detalle del requerimiento, con todas las acciones realizadas durante su atención por parte de la GCSR9.

Pantalla



Reglas funcionales:

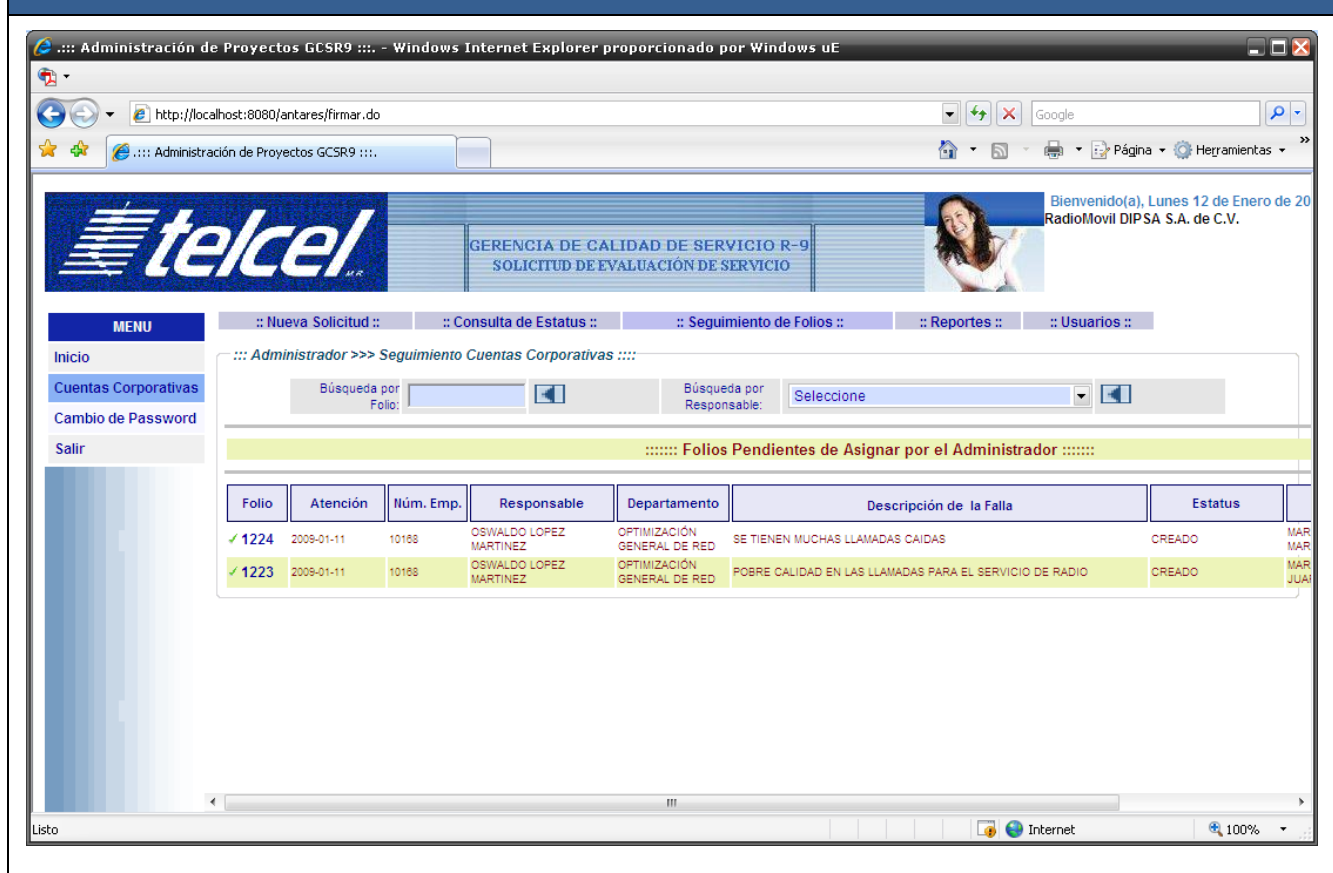
Clave	Especificaciones
ANT_RN_028	Los datos mostrados por esta pantalla son de carácter informativo y no se permite la modificación de alguno de ellos.
ANT_RN_029	La información que se muestra en la pantalla, se divide entre los siguientes apartados: <ol style="list-style-type: none"> 1. Información General de la Solicitud. 2. Información de la Solicitud. 3. Archivos adjuntos a la solicitud. 4. Campos de trabajo.

Clave	Especificaciones
	<ol style="list-style-type: none">5. Seguimiento de la solicitud.6. Acciones implementadas para su solución,7. Observaciones y comentarios.8. Solución y cierre de la solicitud.9. Información del Contacto.
ANT_RN_030	Dentro de la pantalla de consulta del detalle del requerimiento, mediante la liga “Descargar Evidencia” se podrá consultar el documento adjunto al requerimiento en el momento de su creación.
ANT_RN_031	El botón “Regresar” nos permite ir nuevamente a la pantalla de “Resumen de requerimientos”.

CU04-ASIGNAR RESPONSABLE AL REQUERIMIENTO

Nombre	Requerimientos pendientes de asignación.
Descripción	Muestra una lista con los requerimientos cuyo estatus es CREADO y que están pendientes por ser asignados a un analista de optimización de red.

Pantalla

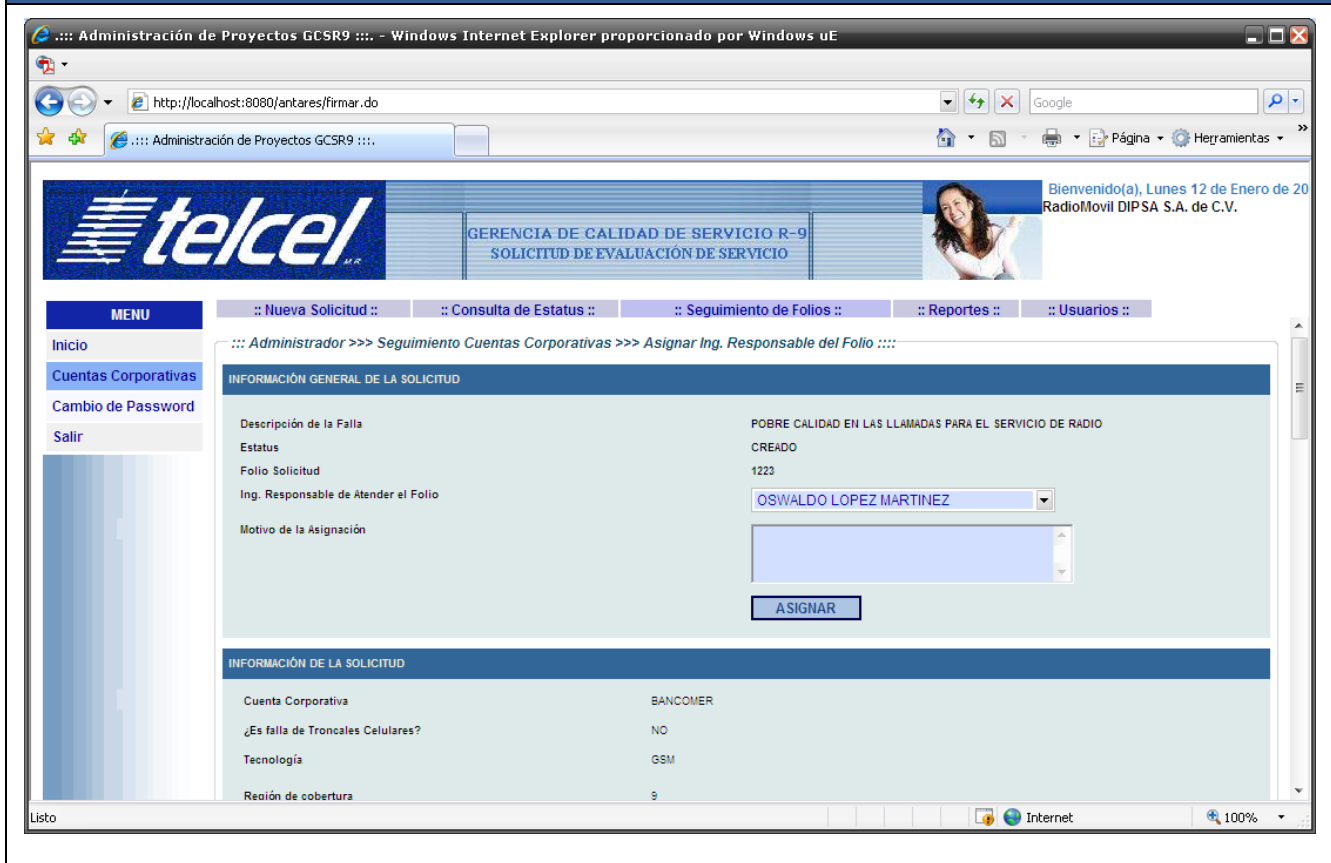


Reglas funcionales:

Clave	Especificaciones
ANT_RN_032	El usuario <Administrador > previamente autenticado en el sistema Antares, selecciona del menú principal la opción “Cuentas Corporativas” y posteriormente invoca la opción “Seguimiento de Folios”.
ANT_RN_033	A continuación el sistema Antares deberá mostrarle en su bandeja operativa, una lista con los requerimientos recién creados y que presentan un estatus de “CREADO”.

Nombre	Consultar el detalle del Requerimiento.
Descripción	Muestra el detalle del requerimiento que está pendiente de asignación.

Pantalla



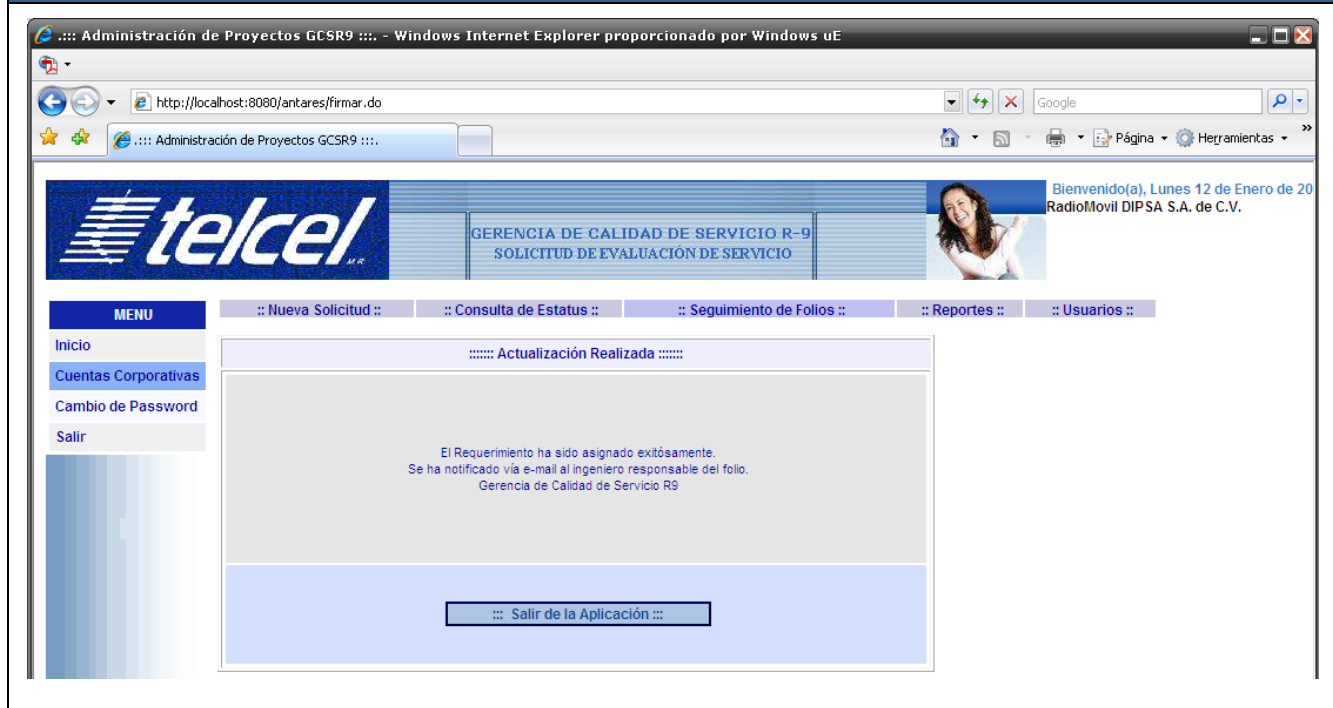
Reglas funcionales:

Clave	Especificaciones
ANT_RN_034	El actor deberá gestionar cada uno de los requerimientos, siendo su tarea principal, la de consultar el detalle de la solicitud.
ANT_RN_035	Solo se podrán asignar aquellos requerimientos cuyo folio tenga un estatus de "CREADO".
ANT_RN_036	El administrador debe evaluar la severidad de la falla reportada, así como identificar la zona geográfica en donde se ubica la cuenta corporativa, a fin de tener los criterios necesarios para asignar el requerimiento a un ingeniero (Analista de Optimización de Red región 9), quien será el responsable de dar

Clave	Especificaciones
	solución y respuesta a la cuenta cliente de Radio Móvil DIPSA.
ANT_RN_037	El administrador debe seleccionar de la lista de analistas de optimización de red, al responsable de dar solución al requerimiento; también se debe capturar un mensaje indicando el motivo de la asignación del requerimiento. Dicho mensaje llegará al usuario responsable de atender el requerimiento,
ANT_RN_038	Durante el tiempo en que el requerimiento permanezca con estatus "CREADO", es responsabilidad del administrador de Antares dar seguimiento del mismo. Es por ello que debe ser delegado a la brevedad posible.
ANT_RN_039	Solo se podrá efectuar la acción de reasignación de requerimiento si se captura una justificación en la que se detallen los motivos por los cuales, el requerimiento pasa a manos de otro ingeniero responsable de la GCSR9.
ANT_RN_040	La reasignación se realiza solo si, el ingeniero actual responsable del requerimiento solicita expresamente apoyo, y lo documenta como una observación en el folio. La solicitud de apoyo puede deberse a una excesiva carga de trabajo, vacaciones o incapacidad médica para el ingeniero responsable.

Nombre	Confirmación de asignación de requerimiento
Descripción	Muestra un mensaje de confirmación indicándole al usuario que se ha asignado exitosamente el requerimiento.

Pantalla



Reglas funcionales:

Clave	Especificaciones
ANT_RN_041	El administrador ha asignado exitosamente la solicitud de atención a la cuenta corporativa, por lo que puede continuar trabajando con el sistema o bien, salir de la aplicación. El mensaje que muestra en pantalla es “El Requerimiento ha sido asignado exitosamente. Se ha notificado vía e-mail al ingeniero responsable del folio. Gerencia de Calidad de Servicio R9”.

Nombre	Pantalla de notificación de requerimiento asignado.
Descripción	Notificación vía correo electrónico al ingeniero de la GCSR9, quien será el responsable de dar solución a la falla reportada por la cuenta corporativa.

Pantalla



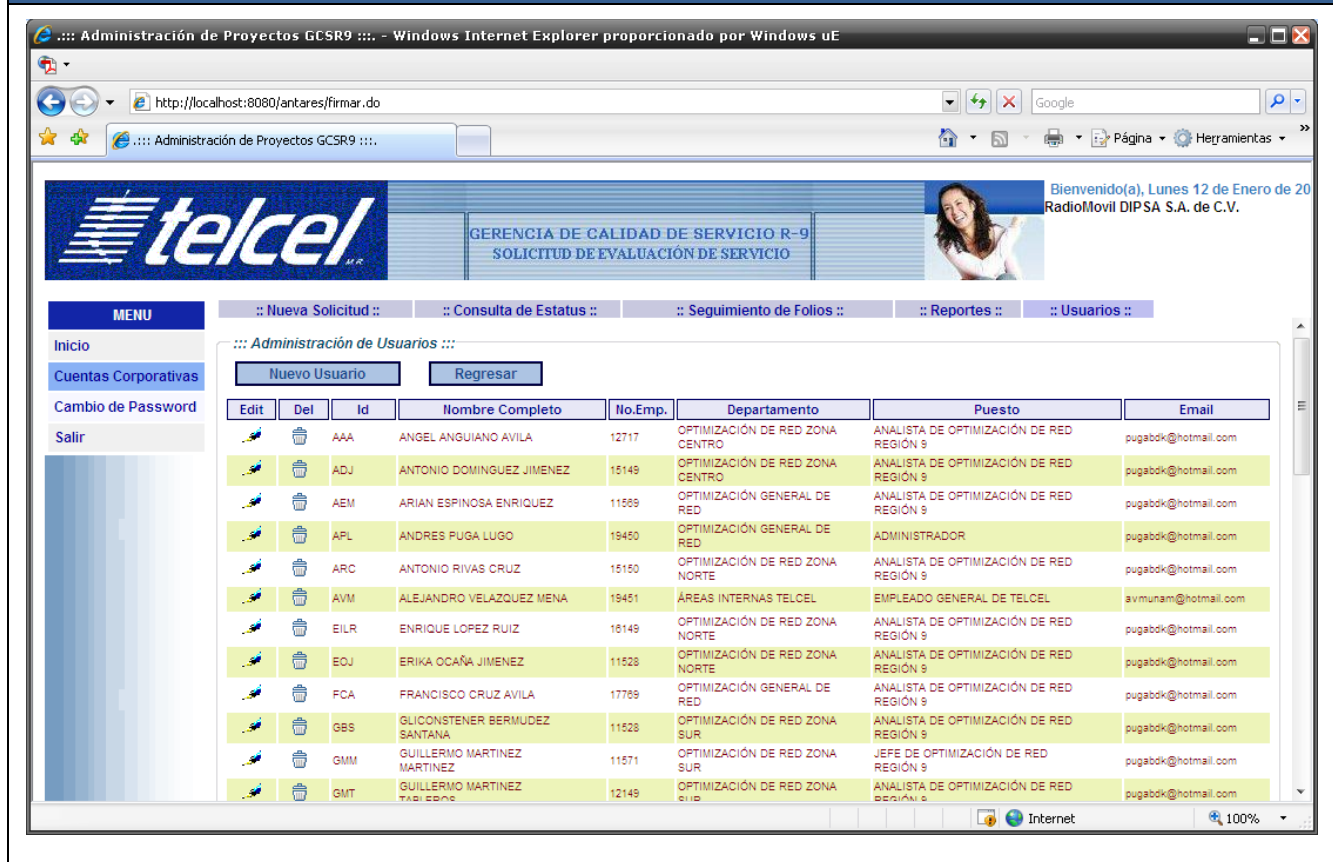
Reglas funcionales:

Clave	Especificaciones
ANT_RN_042	Posteriormente a la asignación del requerimiento, el sistema debe enviar un correo electrónico notificando al ingeniero sobre la asignación y responsabilidad que le fue delegada para la atención de la falla reportada por la cuenta corporativa. La notificación debe ser automática ya que los tiempos de respuesta en la atención a clientes son de vital importancia para la GCSR9 y en general para Radio Móvil DIPSA.

CU07-GESTIONAR USUARIOS

Nombre	Pantalla de administración de usuarios.
Descripción	Muestra una lista con los usuarios de Antares activos.

Pantalla



Reglas funcionales:

Clave	Especificaciones
ANT_RN_043	El actor <Administrador > previamente autenticado en el sistema Antares, selecciona del menú principal la opción “Cuentas Corporativas” y posteriormente invoca la opción “Usuarios”.
ANT_RN_044	A continuación el sistema Antares deberá mostrarle en pantalla al actor, las opciones disponibles para la gestión de usuarios: 1. Agregar un nuevo usuario.

Clave	Especificaciones
	2. Editar la información de un usuario existente. 3. Eliminar permanentemente a un usuario de Antares.
ANT_RN_045	La gestión de usuarios consiste en permitir actualizar información del perfil de los usuarios, así como generar usuarios y eliminarlos permanentemente del sistema.
ANT_RN_046	Debe existir un usuario “administrador” encargado de la recepción y asignación de requerimientos. Este usuario es vital en el modo de operación de la GCSR9, en lo que respecta a la atención a cuentas corporativas.
ANT_RN_047	Los perfiles de Gerente, Analista de Optimización de Red y Jefes de área deben corresponder con sus respectivos departamentos, esta validación es indispensable para evitar la duplicidad de perfiles, así como la redundancia en las acciones efectuadas durante el seguimiento de requerimientos.

Nombre	Agregar un nuevo usuario.
Descripción	Muestra el formulario para completar la alta de un usuario en Antares.
Pantalla	

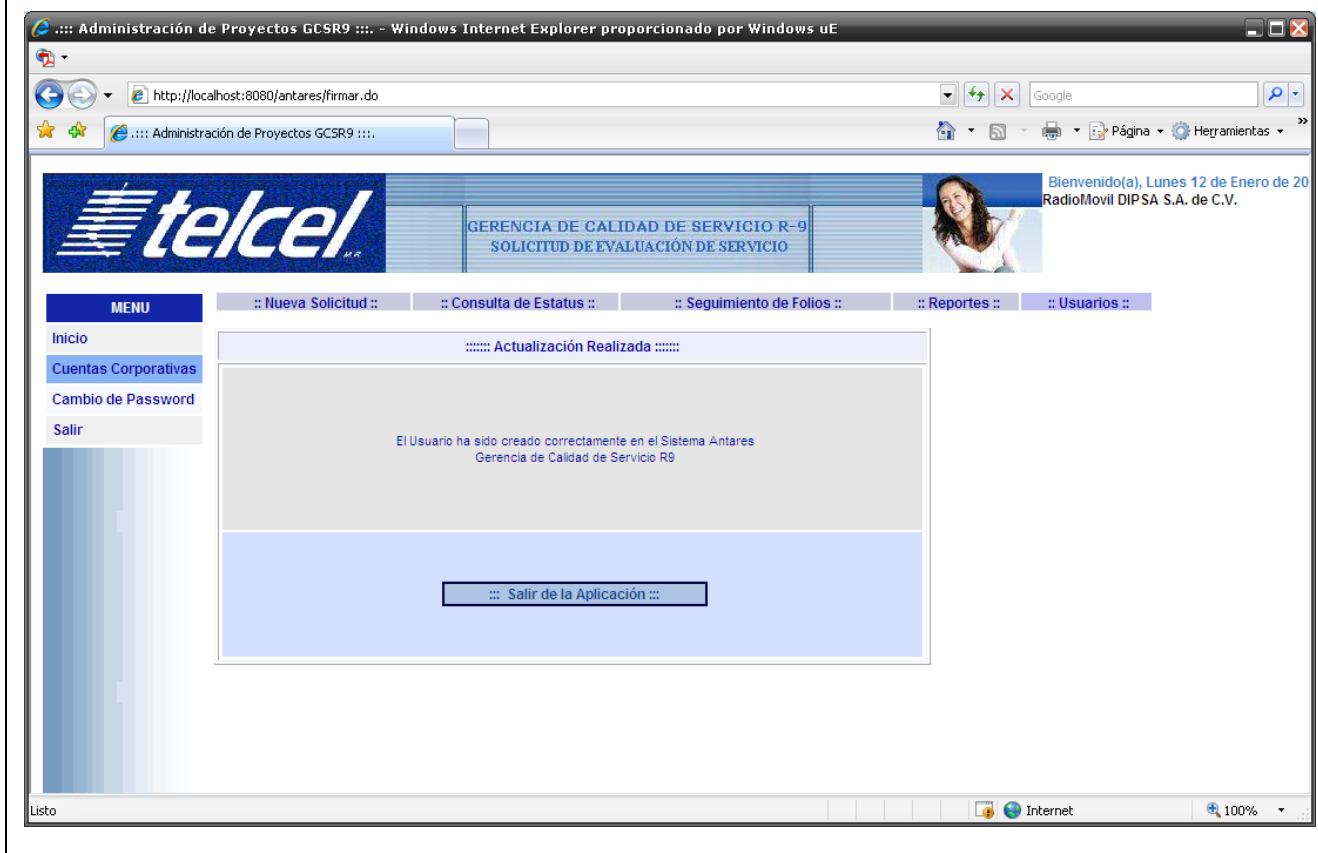
Reglas funcionales:

Clave	Especificaciones
ANT_RN_048	<p>Se debe realizar captura de la información del formulario, ya que es indispensable para el alta del usuario:</p> <ol style="list-style-type: none"> 1. Nombre completo 2. Identificador asignado por la GCSR9 3. Departamento 4. Número de empleado 5. Puesto 6. Correo electrónico

Clave	Especificaciones
ANT_RN_049	Al capturar todos los campos, debe dar clic en el botón "Agregar Usuario".

Nombre	Confirmación de usuario agregado.
Descripción	Muestra la pantalla de confirmación de que se ha agregado un nuevo usuario al sistema Antares.

Pantalla

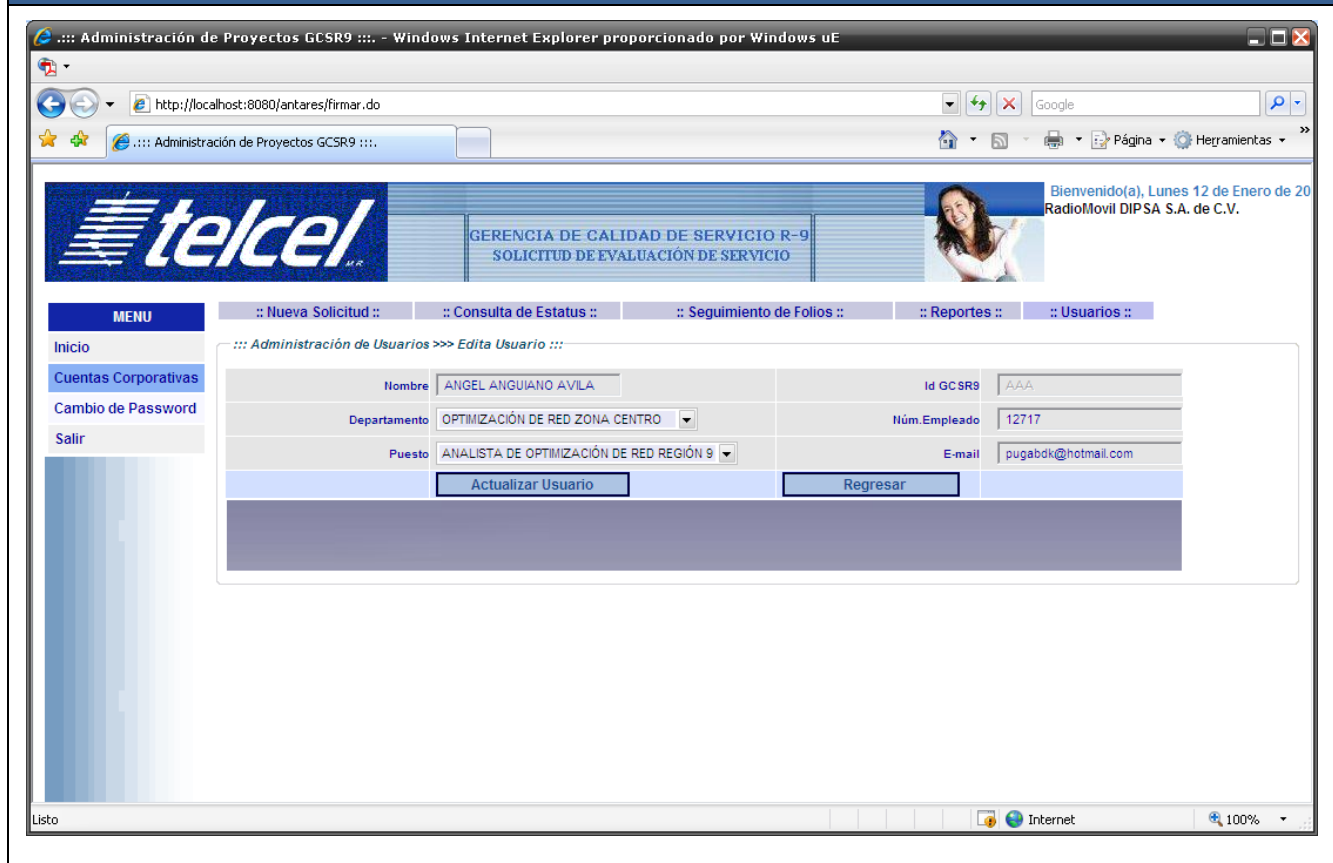


Reglas funcionales:

Clave	Especificaciones
ANT_RN_050	El actor confirma el alta del usuario en Antares. Puede continuar utilizando el sistema o bien, salir de la aplicación.

Nombre	Actualizar el perfil de un usuario.
Descripción	Muestra la pantalla de edición del perfil de usuario.

Pantalla

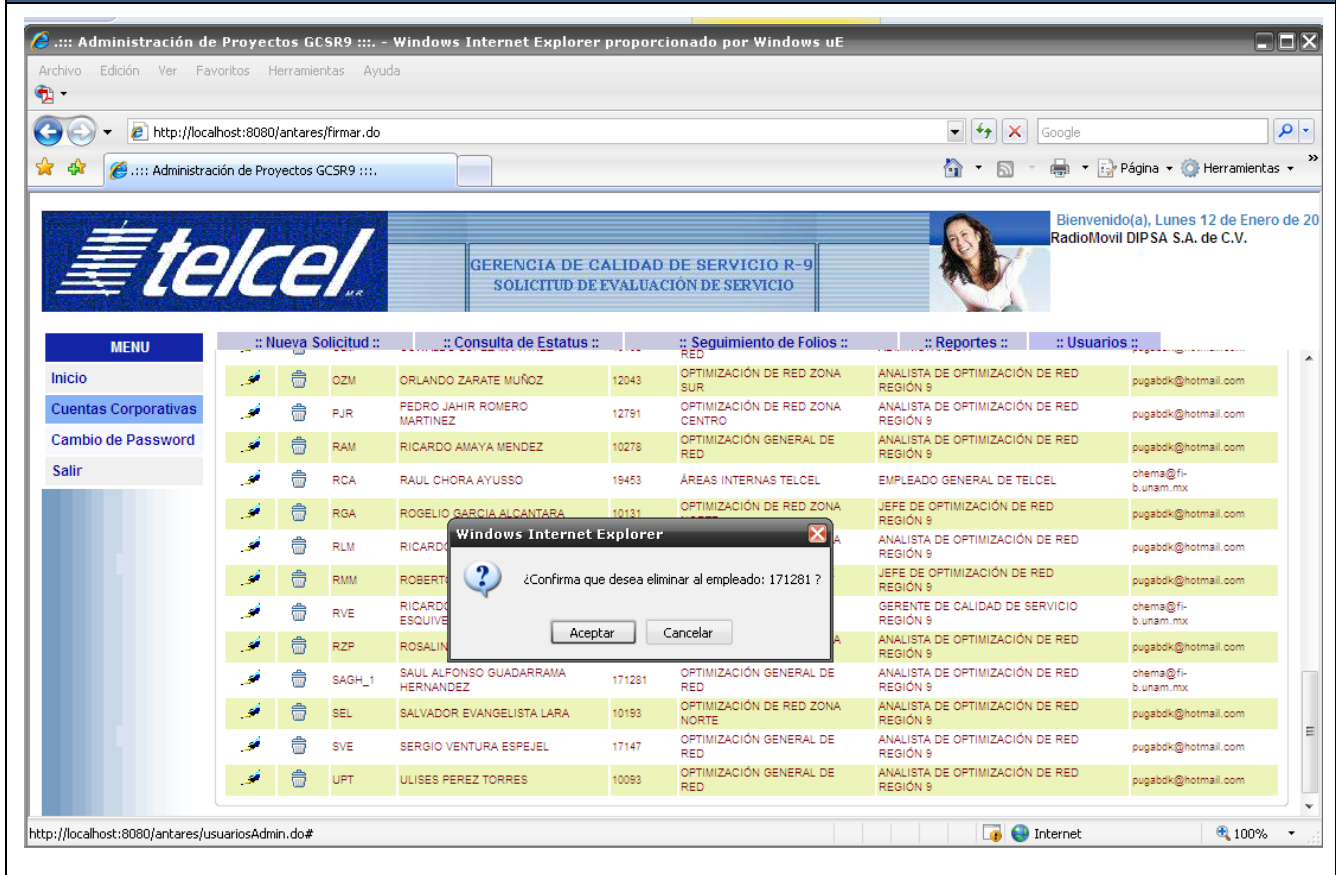


Reglas funcionales:

Clave	Especificaciones
ANT_RN_051	Se puede actualizar la información del usuario: <ol style="list-style-type: none"> 1. Nombre completo 2. Departamento 3. Número de empleado 4. Puesto 5. Correo electrónico
ANT_RN_052	Al actualizar los campos, debe dar clic en el botón “Actualizar Usuario”.

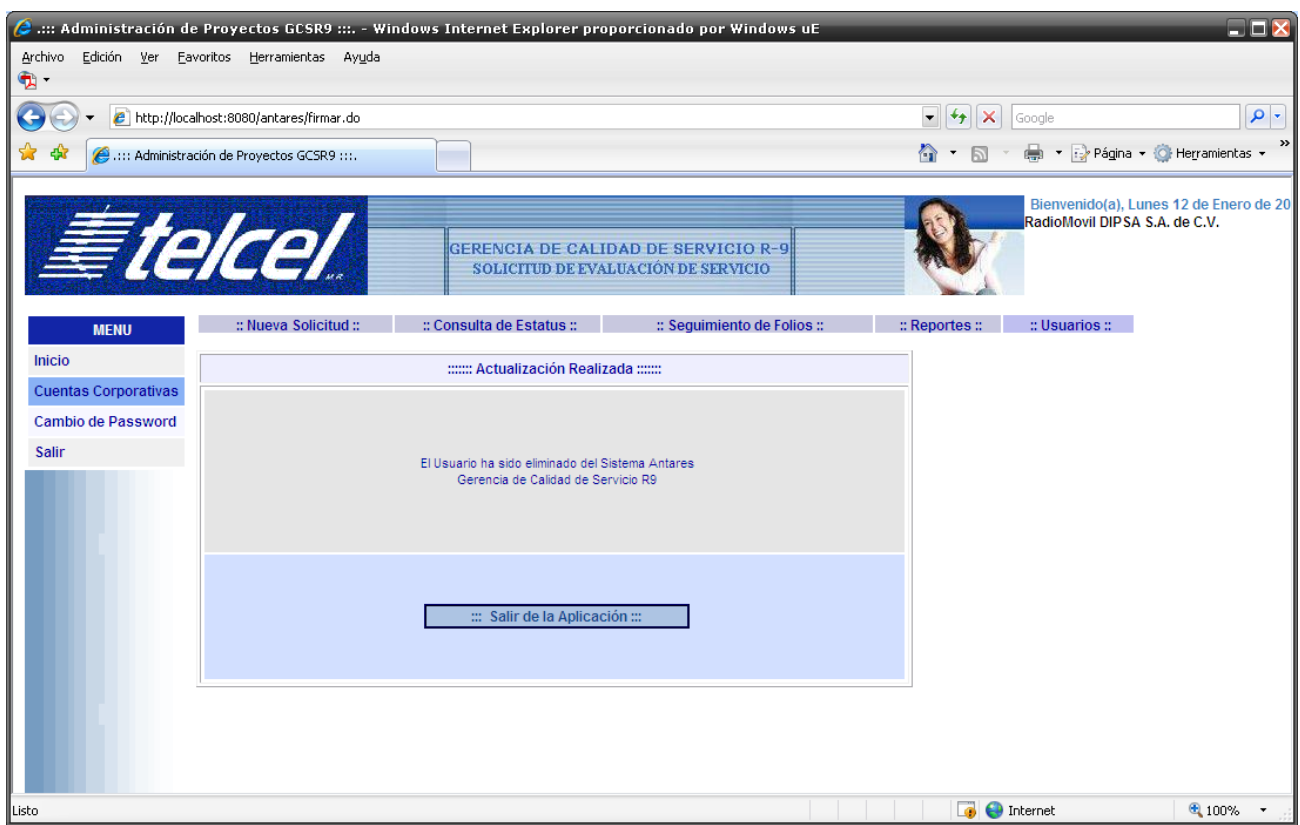
Nombre	Eliminar un usuario.
Descripción	Muestra la pantalla donde se pregunta si se desea eliminar al usuario de Antares.

Pantalla



Reglas funcionales:

Clave	Especificaciones
ANT_RN_053	El administrador debe seleccionar de una lista, al usuario a eliminar del sistema Antares.
ANT_RN_054	El administrador confirma la eliminación del usuario en Antares, al dar clic en "Aceptar".

Nombre	Confirmación de un usuario eliminado.
Descripción	Muestra la pantalla de confirmación de un usuario eliminado de Antares.
Pantalla	
 <p>The screenshot shows a web browser window titled 'Administración de Proyectos GCSR9'. The address bar shows 'http://localhost:8080/antares/firmar.do'. The page header includes the Telcel logo and the text 'GERENCIA DE CALIDAD DE SERVICIO R-9 SOLICITUD DE EVALUACIÓN DE SERVICIO'. A navigation menu on the left includes 'Inicio', 'Cuentas Corporativas', 'Cambio de Password', and 'Salir'. The main content area displays a message: 'Actualización Realizada' followed by 'El Usuario ha sido eliminado del Sistema Antares Gerencia de Calidad de Servicio R9'. A button labeled 'Salir de la Aplicación' is visible at the bottom of the message box.</p>	

Reglas funcionales:

Clave	Especificaciones
ANT_RN_055	El sistema muestra un mensaje ratificando la eliminación permanente del usuario. El mensaje es “El usuario ha sido eliminado del sistema Antares”.
ANT_RN_056	Al eliminar un usuario de Antares, se perderán los privilegios de acceso al mismo, sin embargo, las gestiones realizadas por el mismo, sobre los requerimientos a su cargo persistirán en la base de datos de la GCSR9 a modo de información histórica.

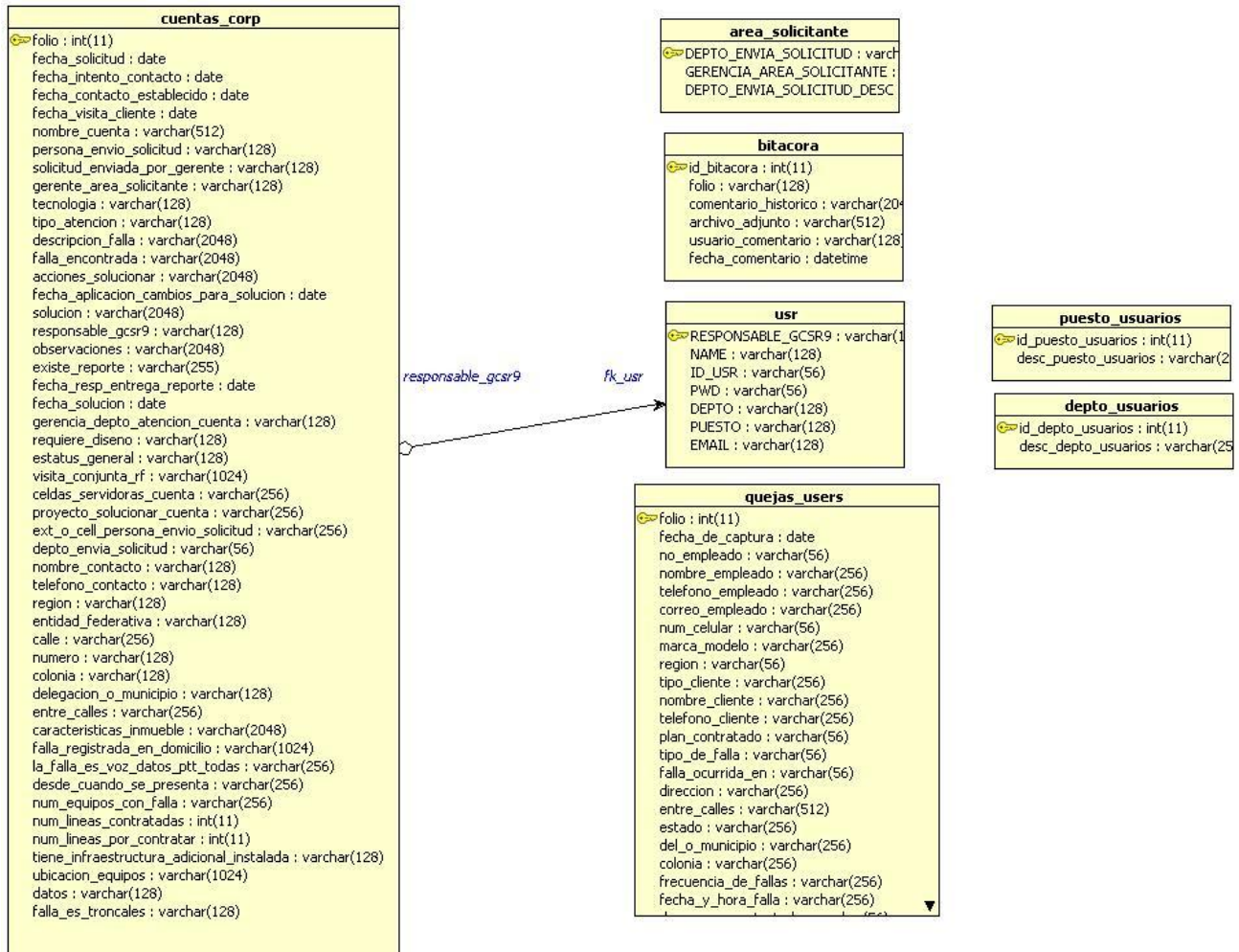
3.4.3 MODELO DE DATOS DE ANTARES

DESCRIPCIÓN DE ENTIDADES DEL MODELO

Nombre de la entidad	Descripción
USR	Es el catálogo de usuarios del sistema Antares, contiene información general, así como los privilegios con los que cuentan dichos usuarios.
AREA_SOLICITANTE	<p>Es el catálogo con la información general de los departamentos de Radio Móvil DIPSA, que tienen relación directa con los clientes. Son los responsables de levantar solicitudes de atención. Las áreas actuales son las siguientes:</p> <ol style="list-style-type: none"> 1.- Help Desk 2.- Ventas 3.- Centrales 4.- Troncales 5.- Radiobases 6.- Nuevas tecnologías 7.- Mantenimiento RAN
DEPTO_USUARIOS	<p>Es el catálogo con la información general de los departamentos que dan atención y seguimiento a los requerimientos del sistema Antares. Los departamentos considerados por el sistema son los siguientes:</p> <ol style="list-style-type: none"> 1.- GERENCIA DE CALIDAD DE SERVICIO R9 2.- OPTIMIZACIÓN GENERAL DE RED 3.- OPTIMIZACIÓN DE RED ZONA NORTE 4.- OPTIMIZACIÓN DE RED ZONA CENTRO 5.- OPTIMIZACIÓN DE RED ZONA SUR 6.- ÁREAS INTERNAS TELCEL
PUESTO_USUARIOS	<p>Es el catálogo con los puestos de los usuarios, contiene la información necesaria para asignar los roles y privilegios sobre el sistema Antares. Los puestos actuales son los siguientes:</p> <ol style="list-style-type: none"> 1.- ANALISTA DE OPTIMIZACIÓN DE RED REGIÓN 9 2.- GERENTE DE CALIDAD DE SERVICIO REGIÓN 9 3.- JEFE DE OPTIMIZACIÓN DE RED REGIÓN 9

Nombre de la entidad	Descripción
	4.- EMPLEADO GENERAL DE TELCEL 5.- ADMINISTRADOR
QUEJAS_USERS	Entidad donde se almacenan los requerimientos por concepto de "Quejas de Clientes Telcel".
BITACORA	Entidad donde se almacenan todas las transacciones realizadas sobre el sistema Antares. El registro de las transacciones es indispensable para el control, seguimiento y rastreabilidad de los requerimientos levantados a la GCSR9.
CUENTAS_CORP	Entidad que almacena el detalle de cada requerimiento de las cuentas corporativas clientes de Radio Móvil DIPSA. Es en esta entidad, donde se registran todas las solicitudes de incidencias y atención a nuevos proyectos realizadas a la GCSR9.

DIAGRAMA ENTIDAD RELACIÓN



DICCIONARIO DE DATOS

Nombre entidad:	USR		
Tipo de entidad:	Catálogo		
Tipo de dato:	Nombre campo:	Nulos:	Descripción:
Varchar	ID_USR	NO	Es el identificador que se le asigna a cada empleado de Radio Móvil DIPSA
Varchar	NAME	NO	Nombre completo del usuario.
Varchar	RESPONSABLE_GCSR9	NO	Es el identificador de usuario, utilizado por el departamento al que está adscrito el usuario.
Varchar	PWD	NO	Contraseña de acceso al sistema.
Varchar	DEPTO	NO	Departamento al que está adscrito el

CAPÍTULO III

			usuario.
Varchar	PUESTO	NO	Puesto que ostenta el usuario dentro de Radio Móvil DIPSA.
Varchar	EMAIL	NO	Correo electrónico institucional del usuario dominio mail.telcel.com

Nombre entidad:	AREA_SOLICITANTE		
Tipo de entidad:	Catálogo		
Tipo de dato:	Nombre campo:	Nulos:	Descripción:
Int	DEPTO_ENVIA_SOLICITUD	NO	Identificador del departamento al que pertenece el área solicitante de apoyo.
Varchar	GERENCIA_AREA_SOLICITANTE	NO	Gerencia a la que pertenece el área solicitante de apoyo.
Varchar	DEPTO_ENVIA_SOLICITUD_DESC	NO	Departamento al que pertenece el área solicitante de apoyo.

Nombre entidad:	DEPTO_USUARIOS		
Tipo de entidad:	Catálogo		
Tipo de dato:	Nombre campo:	Nulos:	Descripción:
Int	ID_DEPTO_USUARIOS	NO	Identificador de los departamentos de Radio Móvil DIPSA.
Varchar	DESC_DEPTO_USUARIOS	NO	Nombre de los departamentos de Radio Móvil DIPSA.

Nombre entidad:	PUESTO_USUARIOS		
Tipo de entidad:	Catálogo		
Tipo de dato:	Nombre campo:	Nulos:	Descripción:
Int	ID_PUESTO_USUARIOS	NO	Identificador correspondiente a los puestos de laGCSR9.
Varchar	DESC_PUESTO_USUARIOS	NO	Descripción correspondiente a los puestos de laGCSR9

Nombre entidad:	QUEJAS_USERS		
Tipo de entidad:	Transaccional		
Tipo de dato:	Nombre campo:	Nulos:	Descripción:
Int	FOLIO	NO	Identificador de la solicitud de la queja registrada.
Date	FECHA_DE_CAPTURA	NO	Fecha en que se genera la solicitud de la queja registrada.
Int	NO_EMPLEADO	NO	Número del empleado que apoya a registrar la queja en el sistema.
Varchar	NOMBRE_EMPLEADO	NO	Nombre del empleado que apoya a registrar la queja en el sistema.
Varchar	TELEFONO_EMPLEADO	NO	Teléfono de la oficina del empleado que apoya a registrar la queja en el sistema.
Varchar	CORREO_EMPLEADO	NO	Correo electrónico del empleado que apoya a registrar la queja en el sistema.
Varchar	NUM_CELULAR	NO	Número celular del equipo que reporta la falla.
Varchar	MARCA_MODELO	NO	Marca y modelo del equipo que reporta la falla.
Int	REGION	NO	Región en la que se reporta la queja.
Varchar	TIPO_CLIENTE	SI	Tipo de cliente que reporta la queja (interno, VIP, plan tarifario, plan amigo).
Varchar	NOMBRE_CLIENTE	SI	Nombre completo del cliente.
Varchar	TELEFONO_CLIENTE	NO	Teléfono del cliente.
Varchar	PLAN_CONTRATADO	NO	Plan contratado por el cliente.
Varchar	TIPO_DE_FALLA	NO	Tipo de falla reportada.
Varchar	FALLA_OCURRIDA_EN	NO	Lugar físico en que se presentó la falla.
Varchar	DIRECCION	NO	Dirección en donde se presentó la falla.
Varchar	ENTRE_CALLES	NO	Calles aledañas a la dirección.
Varchar	ESTADO	NO	Entidad federativa en donde se presentó la falla.
Varchar	DEL_O_MUNICIPIO	NO	Delegación o municipio en donde se presentó la falla.
Varchar	COLONIA	NO	Colonia en donde se presentó la falla.
Varchar	FRECUENCIA_DE_FALLAS	NO	Frecuencia de fallas.
Varchar	FECHA_Y_HORA_FALLA	NO	Fecha y hora en la que ocurrió la falla.
Varchar	DESEA_SER_CONTACTADO	NO	Desea ser contactado por un

			responsable de Racio Móvil DIPSA.
Varchar	DESCRIPCION_FALLA	NO	Descripción de la falla presentada.
Varchar	ASIGNACION	SI	Asignación del folio.
Varchar	RESPONSABLE	SI	Ingeniero responsable del folio.
Int	NO_EMPLEADO_RESPONSABLE	SI	Número de empleado del ingeniero responsable de atender el folio.
Varchar	DEPARTAMENTO	SI	Departamento del ingeniero responsable de atender el folio.
Date	FECHA_ASIGNACION	SI	Fecha en que se asignó el folio.
Varchar	NUM_CELULAR_RESPONSABLE	SI	Número celular del ingeniero responsable de atender el folio.
Varchar	ESTATUS	SI	Estatus de atención del folio.
Varchar	TECNOLOGÍA	SI	Tecnología de la falla reportada.
Varchar	CELDA	SI	Celda servidora del equipo que reporta falla.
Varchar	SECTOR	SI	Sector (A,B,C) de la celda servidora del equipo que reporta falla.
Varchar	NOMBRE_CELDA	SI	Nombre de la celda servidora del equipo que reporta falla.
Varchar	MSC_BSC	SI	Central de la celda servidora del equipo que reporta falla.
Varchar	DESCRIPCION_TECNICA	SI	Diagnóstico técnico realizado por el ingeniero responsable.
Varchar	ACCIONES_EFECTUADAS	SI	Acciones efectuadas para solucionar la queja reportada.
Varchar	COMENTARIOS	SI	Comentarios acerca del seguimiento del folio de atención.
Date	FECHA_ULTIMA_MODIFICACION	SI	Fecha en la que el folio sufrió alguna modificación.

Nombre entidad:	BITACORA		
Tipo de entidad:	Transaccional		
Tipo de dato:	Nombre campo:	Nulos:	Descripción:
Int	ID_BITACORA	NO	Identificador de la transacción registrada.
Int	FOLIO	NO	Folio al que está asociada la transacción.
Varchar	COMENTARIO_HISTORICO	NO	Comentario descriptivo de la transacción.
Varchar	ARCHIVO_ADJUNTO	SI	Ruta en servidor de archivo adjunto a la transacción en caso de existir.
Varchar	USUARIO_COMENTARIO	NO	Usuario que genera la transacción.
Varchar	FECHA_COMENTARIO	NO	Fecha en la que se realiza la transacción.

FASE DE ELABORACIÓN

Nombre entidad:	CUENTAS_CORP		
Tipo de entidad:	Transaccional		
Tipo de dato:	Nombre campo:	Nulos:	Descripción:
Int	FOLIO	NO	Folio asignado al requerimiento de atención a la cuenta corporativa.
Date	FECHA_SOLICITUD	NO	Fecha en la que se registró la solicitud.
Date	FECHA_INTENTO_CONTACTO	SI	Fecha en la que se intenta contactar por primera vez a la cuenta corporativa.
Date	FECHA_CONTACTO_ESTABLECIDO	SI	Fecha en la que se establece una cita para evaluación y seguimiento del folio.
Date	FECHA_VISITA_CLIENTE	SI	Fecha en que se realiza la visita para comenzar con la solución del folio.
Varchar	NOMBRE_CUENTA	NO	Nombre de la cuenta corporativa.
Varchar	PERSONA_ENVIO_SOLICITUD	NO	Empleado de Radio Móvil DIPSA que apoya con el envío de la solicitud.
Varchar	SOLICITUD_ENVIADA_POR GERENTE	NO	La solicitud tiene el VoBo del Gerente del área solicitante.
Varchar	GERENTE_AREA_SOLICITANTE	NO	Nombre del Gerente del área solicitante.
Varchar	TECNOLOGIA	NO	Tecnología en la que ocurre la falla. (DAMPS, GSM, GPRS, EDEG).
Varchar	TIPO_ATENCION	NO	Tipo de atención que requiere la cuenta corporativa.
Varchar	DESCRIPCION_FALLA	NO	Descripción de la falla registrada.
Varchar	FALLA_ENCONTRADA	SI	Diagnóstico realizado por el ingeniero responsable del folio.
Varchar	ACCIONES_SOLUCIONAR	SI	Acciones realizadas para solucionar el problema en la Cuenta corporativa.
Date	FECHA_APLICACION_CAMBIOS_PARA_SOLUCION	SI	Fecha en la que se aplican los cambios en la infraestructura.
Varchar	SOLUCION	SI	Descripción final de la solución.
Varchar	RESPONSABLE_GCSR9	SI	Identificador del ingeniero responsable de atender el folio.
Varchar	OBSERVACIONES	SI	Observaciones sobre el seguimiento en la atención al folio.
Varchar	EXISTE_REPORTE	SI	Existe reporte adjunto a la solicitud.
Date	FECHA_RESP_ENTREGADA	SI	Fecha en la que se entrega respuesta de solución a la cuenta corporativa.
Date	FECHA_SOLUCION	SI	Fecha en la que se soluciona y se da por concluido el folio.
Varchar	GERENCIA_DEPTO_ATENCION_CUENTA	SI	Gerencia a la que pertenece el departamento que atiende el folio.
Varchar	REQUIERE_DISENO	SI	La cuenta corporativa requiere diseño por parte de la Gerencia de Radio frecuencia.
Varchar	ESTATUS_GENERAL	NO	Estatus general del folio.
Varchar	VISITA_CONJUNTA_RF	SI	Se requiere visita en conjunto con Radio

CAPÍTULO III

			frecuencia a la cuenta corporativa.
Varchar	CELDAS_SERVIDORAS_CUENTA	SI	Celdas servidoras en el área de cobertura de la cuenta corporativa.
Varchar	PROYECTO_SOLUCIONAR_CUENTA	SI	Se generará un nuevo proyecto para atender el folio de la cuenta corporativa.
Varchar	EXT_O_CEL_PERSONA_ENVIO_SOLICITUD	SI	Extensión telefónica del empleado que envió la solicitud.
Varchar	DEPTO_ENVIA_SOLICITUD	SI	Departamento al que pertenece el empleado que envió la solicitud.
Varchar	NOMBRE_CONTACTO	NO	Nombre del contacto con la cuenta corporativa.
Varchar	TELEFONO_CONTACTO	NO	Teléfono del contacto con la cuenta corporativa.
Int	REGION	NO	Región en la que se localiza la cuenta corporativa.
Varchar	ENTIDAD_FEDERATIVA	NO	Entidad federativa en la que se localiza la cuenta corporativa.
Varchar	CALLE	NO	Calle en la que se localiza la cuenta corporativa.
Varchar	NUMERO	NO	Número de la calle en la que se localiza la cuenta corporativa.
Varchar	COLONIA	NO	Colonia en la que se localiza la cuenta corporativa.
Varchar	DELEGACION_O_MUNICIPIO	NO	Delegación o municipio.
Varchar	ENTRE_CALLES	NO	Calles aledañas a la cuenta corporativa.
Varchar	CARACTERISTICAS_INMUEBLE	NO	Características físicas del inmueble.
Varchar	FALLA_REGISTRADA_EN_DOMICILIO	NO	La falla se registra en el domicilio de la cuenta corporativa.
Varchar	LA_FALLA_ES_VOZ_DATOS_PT_T_TODAS	NO	Tipo de falla presentada.
Varchar	DESDE_CUANDO_SE_PRESENTA	NO	Periodo a partir del que se presenta la falla.
Varchar	NUM_EQUIPOS_CON_FALLA	NO	Número de equipos que presentan la falla.
Varchar	NUM_LINEAS_CONTRATADAS	NO	Número de líneas contratadas por la cuenta corporativa.
Varchar	NUM_LINEAS_POR_CONTRATAR	NO	Número de líneas por contratar por la cuenta corporativa.
Varchar	TIENE_INFRAESTRUCTURA_ADICIONAL_INSTALADA	NO	La cuenta corporativa tiene infraestructura instalada.
Varchar	UBICACIÓN_EQUIPOS	NO	Distribución de los equipos, en las instalaciones de la cuenta corporativa.
Varchar	DATOS	NO	La falla es de datos.
Varchar	FALLA_ES_TRONCALES	NO	La falla se presenta en las troncales celulares.

CAPÍTULO IV: FASE DE CONSTRUCCIÓN

Al final de la fase de elaboración del sistema en proceso de construcción Antares, se encuentra en un estado en el que su arquitectura es completamente ejecutable y más importante aún, estable. Se han establecido los fundamentos significativos del negocio de la GCSR9 mediante los modelos de análisis y diseño, se tienen contemplados los subsistemas, componentes e interfaces con las que se comunica, así como la realización de los casos de uso principales, correspondiendo con las principales actividades de los procesos de negocio de los usuarios de Radio Móvil DIPSA.

El propósito principal de esta fase, es convertir al sistema Antares en un producto de software operativo, garantizando que cumple con los requisitos funcionales de la GCSR9 y teniendo la calidad adecuada para ser presentado a los usuarios finales.

4.1 DESCRIPCIÓN DE LA FASE DE CONSTRUCCIÓN

A medida que el proceso de desarrollo de software pasa de la fase de elaboración a la de construcción, también se modifica el enfoque y el rumbo del proyecto. Esto se debe a que en las fases de inicio y elaboración el énfasis se sitúa en adquirir el conocimiento del entorno del negocio en el que se desempeñará el sistema, mientras que en la fase de construcción, los esfuerzos se centran en la construcción de un sistema funcional, teniendo como directrices: el costo, esfuerzo y un plan de tiempos para el desarrollo del mismo.

Una definición muy adecuada acerca de la fase de construcción de un sistema, es la de Ivar Jacobson en su obra “El proceso unificado de desarrollo de software”, en la que detalla: *“La fase de construcción en pocas palabras consiste en desarrollar un producto de software listo para su operación inicial en un entorno de usuario, mejor conocido como versión Beta. Para ello se detallan los escenarios por completo y se continúan los flujos de trabajo a través de iteraciones, dejando cerrados los modelos de análisis, diseño e implementación; integrando además todos los subsistemas” [10].*

4.2 IMPLEMENTCIÓN DE ANTARES

En esta etapa del proceso de desarrollo de Antares, los requisitos de la GCSR9, así como la arquitectura del sistema son estables, es momento entonces de poner énfasis en completar el desarrollo de los casos de uso, esto se hace mediante la codificación de clases java que irán extendiendo los marcos de trabajo (Struts 1.2, Spring 2 y Hibernate 3) y que gradualmente irán dando forma a los componentes requeridos por el módulo de Cuentas Corporativas.

Al finalizar la construcción de todos los componentes de Cuentas Corporativas, se procede con las pruebas unitarias de cada uno de ellos, si estas resultan exitosas es momento entonces de realizar la integración de los componentes que en conjunto completan al sistema Antares. Una vez concluida la integración la versión del sistema es proporcionada al usuario de la GCSR9 para que realice sus pruebas funcionales, pudiendo otorgar una retroalimentación con base en una lista de incidencias y/o mejoras que se tengan respecto al sistema.

4.2.1 CONVENCIONES DE CODIFICACIÓN

ESTÁNDARES PARA EL NOMBRAMIENTO DE CLASES DEL SISTEMA ANTARES

Los nombres de las clases deberán ser lo suficientemente descriptivos, como para poder conocer las funciones que realizan en el sistema, aunado a esto se siguen las siguientes convenciones:

- El nombramiento de las clases se realiza tomando como mayúscula a la primer letra, así como el inicio de cada palabra que complementa el nombre de la clase, por ejemplo: “ClaseDeEjemplo.java”.
- Las clases que modelarán el comportamiento de las reglas del negocio de la GCSR9 tienen el sufijo “Interface”, por ejemplo “UsuarioInterface.java”.
- Las clases responsables de implementar la funcionalidad de las reglas del negocio de la GCSR9 tienen el sufijo “Impl” por ejemplo: “UsuarioImpl.java”.
- Las clases que independizan la petición del cliente del modelo de negocio, son servlets que funcionan como controladores en la arquitectura de Antares, quienes a su vez son responsables del empaquetado y enrutado del tráfico HTTP hacia el manejador del marco de trabajo Struts. Los servlets tienen el sufijo “Action” por ejemplo: “UsuarioMappingDispatchAction.java”.

- Las clases que permite manipular los datos de entrada entre el usuario y la capa de negocio de la aplicación se conocen como ActionForms y tienen el sufijo “Form”, por ejemplo “LoginForm.java”.
- Las clases encargadas de la persistencia de la información del negocio y en general de realizar los accesos a las bases de datos, aparecen con el sufijo DAO (Data Access Object), teniendo la siguiente notación: “UsuarioDao.java”.
- El patrón de diseño VO (Value Objects), también conocido como DTOs (Data Transfer Objects) es utilizado para facilitar la comunicación entre las capas de la arquitectura de Antares; de tal forma que las clases encargadas de manejar los Beans, aparecen con el sufijo VO, teniendo la siguiente notación: ClaseVO.java.
- Las vistas del sistema como tal, se componen de páginas HTML, imágenes y JSP's, cuyos nombres describen el tipo de información que muestran.

ORGANIZACIÓN DEL CÓDIGO FUENTE DEL SISTEMA ANTARES

La organización del código fuente es la siguiente:

- El proyecto Antares de la GCSR9 se encuentra estructurado en dos carpetas principales:
 1. La carpeta de código fuente llamada “src”, la cual contiene el paquete **com.gpo.modelo**, que es el paquete principal del proyecto de software.
 2. La carpeta “WebRoot” que contiene los archivos necesarios para la publicación y el adecuado despliegue de la aplicación Antares como una aplicación empresarial J2EE.

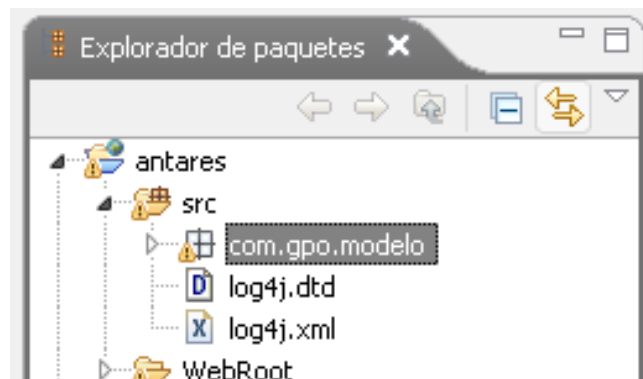


Fig. 4.1 Estructura del proyecto Antares.

- La carpeta src organiza el código fuente de la aplicación en 3 estructuras de paquetes, respetando la separación de capas de una aplicación J2EE basada en el patrón Model View Controller:

1. `model`: Contiene el paquete “`daos`” que aloja las clases encargadas de realizar los accesos a las bases de datos; así como las clases POJO que son los JavaBean que representan a cada entidades de la base de datos de Antares como objetos java.

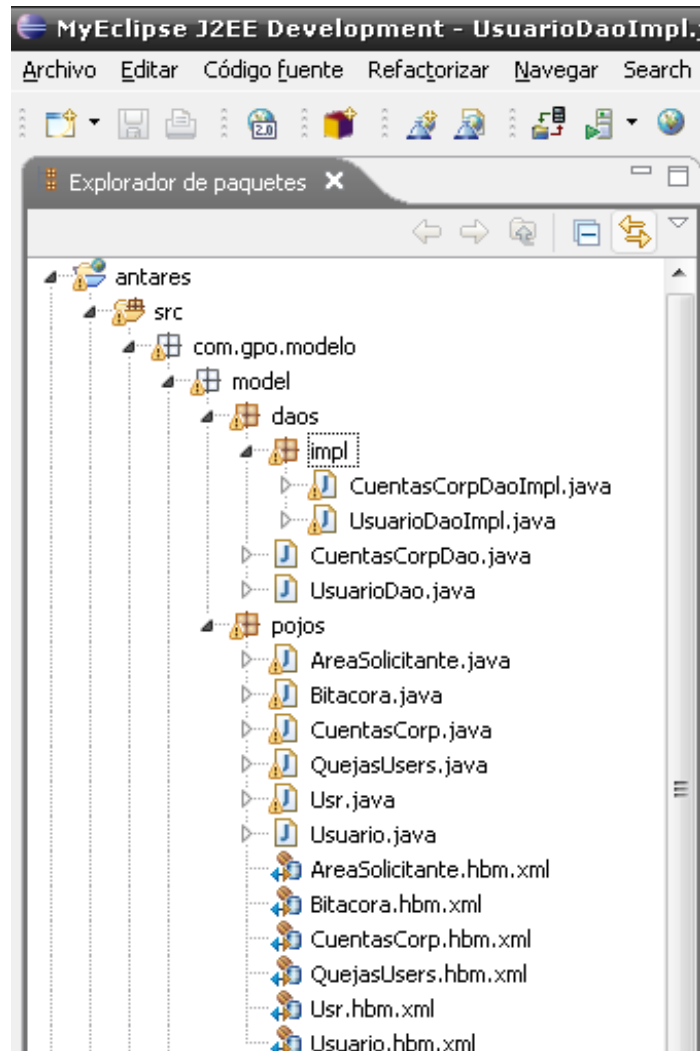


Fig. 4.2 Estructura del paquete `model`.

2. `negocio.servicios`: El paquete “`negocio.servicios`” contiene las clases encargadas de atender las reglas del negocio de la GCSR9. La funcionalidad de la aplicación se provee mediante el patrón business delegate, el cual plantea la creación de servicios pertenecientes a la capa de negocio, que servirán de nexo de unión con la capa de control y mediante estos, Struts pedirá y recibirá los únicos objetos que entiende los VOs, clases incluidas también en este paquete y cuya labor es manejar los Beans u objetos de valor del sistema.

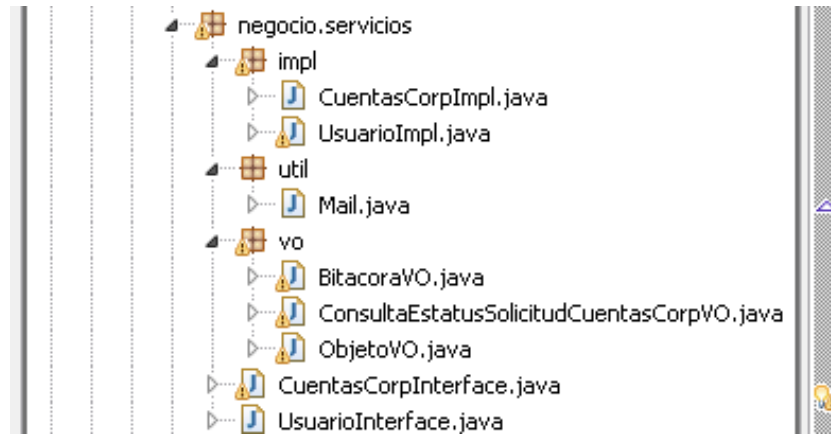


Fig. 4.3 Estructura del paquete negocio.servicios.

3. vista: El paquete “vista” contiene las clases action (servlets que implementan la funcionalidad del controlador), así como los form (clases que permiten el flujo de datos de entrada hacia delante y hacia atrás entre el usuario y la capa de negocio).

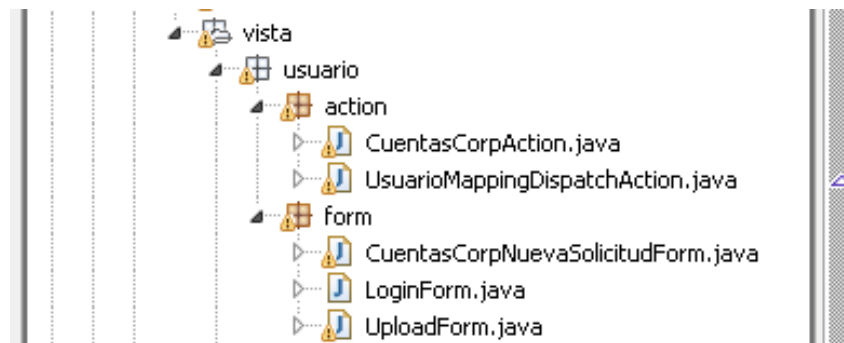


Fig. 4.4 Estructura del paquete vista.

- La carpeta “WebRoot” que contiene los archivos que componen las vistas de la aplicación (HTML, JSP, CSS e imágenes), así como los archivos de configuración de la aplicación Web (applicationContext.xml, hibernate.cfg.xml, struts-config.xml, web.xml) necesarios para la publicación y el despliegue del proyecto como una aplicación Web. Adicionalmente los archivos DTD de descripción y sintaxis para documentos XML.

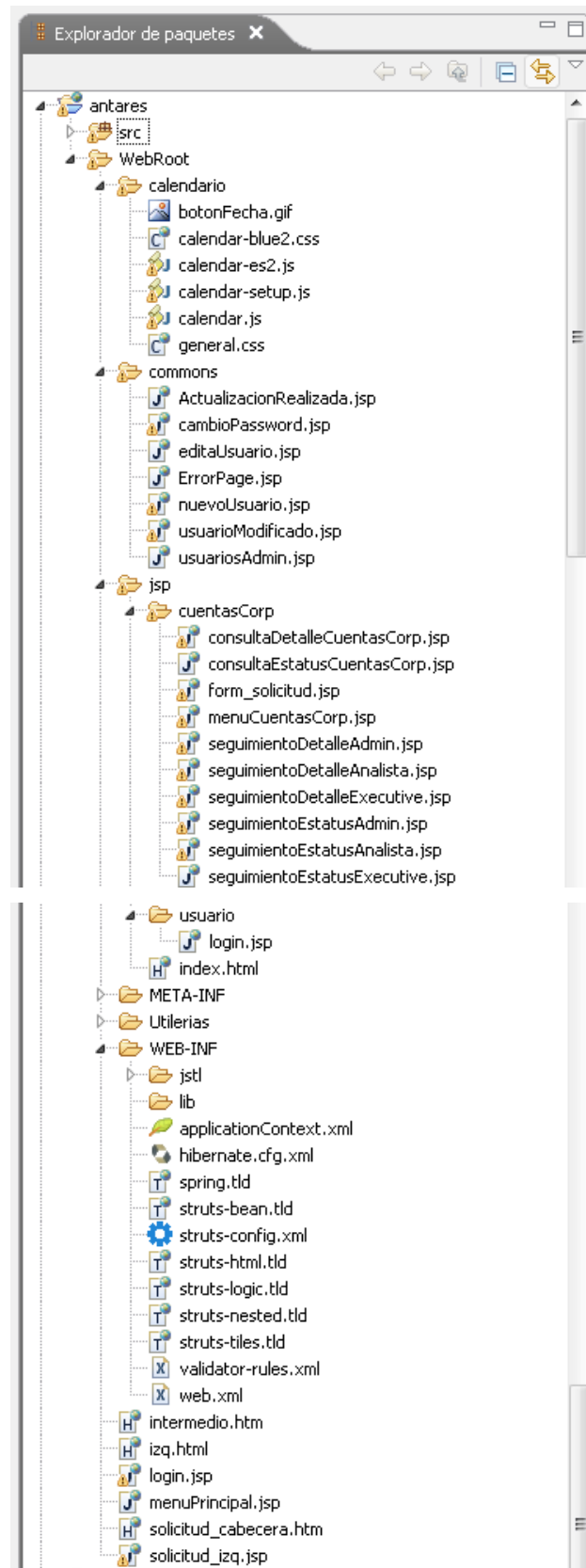


Fig. 4.5 Estructura de la carpeta WebRoot.

4.2.2 DESCRIPCIÓN DEL CÓDIGO FUENTE DEL SISTEMA ANTARES

Las clases de que se compone Antares son de 8 tipos diferentes:

1. JSP: Java Server Pages: Estas clases componen la interfaz gráfica al usuario o vista del sistema, constan de código html y java embebidos en un archivo con extensión .jsp las cuales muestran las pantallas del sistema donde el usuario podrá interactuar con el mismo.

Ruta	Nombre	Descripción
/antares/	login.jsp	Punto de acceso inicial del sistema Antares. En esta vista es en donde se le solicita al usuario proporcionar su clave y contraseña para autenticarse e ingresar al sistema
/antares/	intermedio.htm	Página html que contiene el fondo principal de las vistas.
/antares/	izq.html	Página html que contiene el diseño de los menús de la aplicación.
/antares/	menuPrincipal.jsp	Despliega el menú principal de la aplicación.
/antares/	solicitud_cabecera.htm	Contiene las imágenes y texto que conforman la cabecera de la aplicación.
/antares/jsp/ cuentasCorp	menuCuentasCorp.jsp	Menú principal del módulo de Cuentas Corporativas.
/antares/jsp/ cuentasCorp	form_solicitud.jsp	Presenta el formulario de captura de los requisitos para la atención y servicio a una cuenta corporativa.
/antares/jsp/ cuentasCorp	consultaEstatusCuentasCorp.jsp	Permite la consulta de estatus e información general para algún requerimiento en particular.
/antares/jsp/ cuentasCorp	consultaDetalleCuentasCorp.jsp	Permite la consulta detallada de algún requerimiento en particular. Mediante esta pantalla es factible descargar los documentos que han sido adjuntos mediante el seguimiento del requerimiento.
/antares/jsp/ cuentasCorp	seguimientoDetalleAdmin.jsp	Esta vista permite realizar el seguimiento de los requerimientos de atención a cuentas corporativas, para el perfil de usuario Administrador.

Ruta	Nombre	Descripción
/antares/jsp/cuentasCorp	seguimientoDetalleAnalista.jsp	Esta vista permite realizar el seguimiento de los requerimientos de atención a cuentas corporativas, para el perfil de usuario Analista.
/antares/jsp/cuentasCorp	seguimientoDetalleEjecutivo.jsp	Esta vista permite realizar el seguimiento de los requerimientos de atención a cuentas corporativas, para el perfil de usuario Ejecutivo (Jefes y Gerente de la GCSR9).
/antares/jsp/cuentasCorp	seguimientoEstatusAdmin.jsp	Esta vista permite realizar la consulta del seguimiento de los requerimientos de atención a cuentas corporativas, para el perfil de usuario Administrador.
/antares/jsp/cuentasCorp	seguimientoEstatusAnalista.jsp	Esta vista permite realizar la consulta del seguimiento de los requerimientos de atención a cuentas corporativas, para el perfil de usuario Analista.
/antares/jsp/cuentasCorp	seguimientoEstatusEjecutivo.jsp	Esta vista permite realizar la consulta del seguimiento de los requerimientos de atención a cuentas corporativas, para el perfil de usuario Ejecutivo (Jefes y Gerente de la GCSR9).

2. DAOs: Clases de acceso a datos: Estas clases se encargan de realizar las consultas requeridas por el sistema a las bases de datos MySql, cada módulo del sistema tiene sus propias clases de acceso a datos asociadas con el propósito de modularizar el sistema y reducir el mantenimiento al mismo.

Ruta	Nombre	Descripción
com/gpo/modelo/model/daos	UsuarioDao.java	Interface que modela el comportamiento de los accesos a la base de datos para los servicios referentes al proceso de autenticación de usuarios a Antares.
com/gpo/modelo/model/daos	CuentasCorpDao.java	Interface que modela el comportamiento de los accesos a la base de datos para los servicios

Ruta	Nombre	Descripción
		referentes a los procesos del módulo de cuentas corporativas.
com/gpo/modelo/model/daos/impl	UsuarioDaolImpl.java	Clase que implementa las transacciones con la base de datos para los servicios referentes al proceso de autenticación de usuarios a Antares.
com/gpo/modelo/model/daos/impl	CuentasCorpDaolImpl.java	Clase que implementa las transacciones con la base de datos para los servicios referentes a los procesos del módulo de cuentas corporativas.

3. Pojos: Son las clases que representa las entidades de la base de datos de Antares como objetos java y realizan lo que se conoce como mapeo de la entidad a objeto.

Ruta	Nombre	Descripción
com/gpo/modelo/model/pojos	AreaSolicitante.java	Clase JavaBean que contiene los atributos y métodos set y get para encapsular las columnas de la tabla area_solicitante.
com/gpo/modelo/model/pojos	Bitacora.java	Clase JavaBean que contiene los atributos y métodos set y get para encapsular las columnas de la tabla bitacora.
com/gpo/modelo/model/pojos	CuentasCorp.java	Clase JavaBean que contiene los atributos y métodos set y get para encapsular las columnas de la tabla cuentas_corp.
com/gpo/modelo/model/pojos	QuejasUsers.java	Clase JavaBean que contiene los atributos y métodos set y get para encapsular las columnas de la tabla quejas_users.
com/gpo/modelo/model/pojos	Usr.java	Clase JavaBean que contiene los atributos y métodos set y get para encapsular las columnas de la tabla usr.
com/gpo/modelo/model/pojos	AreaSolicitante.hbm.xml	Es el archivo XML para el mapeo de la entidad area_solicitante de la base de datos en un objeto java.

Ruta	Nombre	Descripción
com/gpo/modelo/model/pojos	Bitacora.hbm.xml	Es el archivo XML para el mapeo de la entidad bitacora de la base de datos en un objeto java.
com/gpo/modelo/model/pojos	CuentasCorp.hbm.xml	Es el archivo XML para el mapeo de la entidad cuentas_corp de la base de datos en un objeto java.
com/gpo/modelo/model/pojos	QuejasUsers.hbm.xml	Es el archivo XML para el mapeo de la entidad quejas_users de la base de datos en un objeto java.
com/gpo/modelo/model/pojos	Usr.hbm.xml	Es el archivo XML para el mapeo de la entidad usr de la base de datos en un objeto java.

4. Clases de negocio: Estas clases mantienen y ejecutan las reglas de negocio de Antares asociadas a un modulo en especifico. Cada modulo del sistema cuenta con sus propias clases de negocio asociadas, mediante las cuales hace los cálculos y operaciones requeridas por el sistema.

Ruta	Nombre	Descripción
com/gpo/modelo/negocio/servicios	UsuarioInterface.java	Interface que modela el comportamiento de los servicios referentes al proceso de autenticación de usuarios a Antares.
com/gpo/modelo/negocio/servicios	CuentasCorpInterface.java	Interface que modela el comportamiento de los servicios referentes a los procesos del módulo de cuentas corporativas.
com/gpo/modelo/negocio/servicios/impl	UsuarioImpl.java	Clase que implementa las reglas de negocio para el proceso de autenticación al sistema Antares.
com/gpo/modelo/negocio/servicios/impl	CuentasCorpImpl.java	Clase que implementa las reglas de negocio y la funcionalidad requerida en las operaciones diarias de la GCSR9 para la atención a cuentas corporativas.

5. Clases de utilerías: Las clases de utilerías son clases planas de Java utilizadas para generar funciones comunes al sistema con el propósito de reutilizar dichas funciones y evitar duplicidad de código en la aplicación.

Ruta	Nombre	Descripción
com/gpo/modelo/negocio/servicios/util	Mail.java	Clase encargada de establecer los métodos de comunicación con el proveedor del servicio de correo para la GCSR9 mediante el protocolo SMTP.
com/gpo/modelo/negocio/servicios/util	Reporteador.java	Clase responsable de las tareas de reportes planeados por la GCSR9 para el seguimiento mensual de productividad de los departamentos adscritos a la Gerencia.

6. Clases VO: Estas clases representan a los “Objetos de Valor” para el negocio de Antares y se encargan transportar los valores del sistema a través de sus capas(modelo-vista-controlador).

Ruta	Nombre	Descripción
com/gpo/modelo/negocio/servicios/vo	ObjetoVO.java	Es la clase que contiene las propiedades y métodos de acceso a las mismas de un objeto tipo Map (llave - valor).
com/gpo/modelo/negocio/servicios/vo	ConsultaEstatusSolicitudCuentasCorpVO.java	Es la clase que contiene las propiedades y métodos de acceso a las mismas para la consulta a detalle de la información correspondiente una solicitud de atención expedida por Antares.
com/gpo/modelo/negocio/servicios/vo	BitacoraVO.java	Es la clase que contiene las propiedades y métodos de acceso a las mismas para las transacciones efectuadas durante el seguimiento de un requerimiento de atención a cuentas corporativas.

7. Action servlets: Los actions son servlets que se encargan de administrar el flujo de los procesos del sistema con el propósito de que cuando un usuario haga una petición al sistema este obtenga una respuesta del mismo.

Ruta	Nombre	Descripción
com/gpo/modelo/vista/usuario/action	UsuarioMappingDispatchAction.java	Esta clase procesa las peticiones de autenticación y permite controlar el flujo de navegación del usuario.

Ruta	Nombre	Descripción
com/gpo/mode lo/vista/usuario /action	CuentasCorpAction.java	Esta clase procesa las peticiones referentes a los servicios del módulo Cuentas Corporativas y permite controlar el flujo de navegación del usuario.

8. Action forms: Clases que permiten el flujo de datos de entrada hacia delante y hacia atrás entre el usuario y la capa de negocio. Deben mantener la estructura de un JavaBean e implementar los métodos set y get para sus atributos.

Ruta	Nombre	Descripción
com/gpo/mode lo/vista/usuario /form	LoginForm.java	Form encargado de cachar los parámetros clave y contraseña de usuario en la página de login en el proceso de autenticación.
com/gpo/mode lo/vista/usuario /form	UploadForm.java	Form encargado de adjuntar archivos durante el proceso de seguimiento y atención a cuentas corporativas.
com/gpo/mode lo/vista/usuario /form	CuentasCorpNuevaSolicit udForm.java	Form encargado de cachar los parámetros requeridos en la página de Nueva solicitud de atención a cuentas corporativas.

4.3 CRITERIOS DE EVALUACIÓN

Durante la fase de construcción, se deben cubrir con los criterios específicos establecidos con base en los requisitos funcionales de la GCSR9 y que se detallaron en los casos de uso; dicho de otra manera, los criterios de evaluación de Antares giran en torno al apego que este tenga con los:

- Requisitos funcionales (Casos de uso de la GCSR9).
- Requisitos no funcionales (Rendimiento, disponibilidad, facilidad de uso).
- Mitigación de riesgos.

La evaluación de estos criterios permite visualizar cuando se ha completado la construcción de una iteración, y es precisamente en esta fase en la que se prepara el material adicional para capacitar a los diferentes usuarios que utilizarán el sistema (administradores del sistema y usuarios finales de Radio Móvil DIPSA):

- Material de usuario: GUIAS DE USUARIO (TECNICAS)
- Material de cursos: CAPACITACION

4.3.1 REQUISITOS NO FUNCIONALES

a. Cotas de volúmenes de operación:

El nuevo sistema Antares para la GCSR9 debe soportar los siguientes volúmenes como cotas para su correcto desempeño. Los datos han servido como la referencia tanto para dimensionar la base de datos, como para diseñar los esquemas y algoritmos propios del sistema para procesar esa cantidad de información:

No.	Aspecto	Mín.	Máx.	Comentarios e información adicional
1	Espacio en disco en el servidor de aplicaciones Tomcat 5.0 y/o Weblogic 8.1 estimado para la realización del despliegue del sistema Antares. (Referentes a las capas de negocio y de interface con el usuario).	500 MB	N/A	Este espacio en disco debe considerar las posibles extensiones a Antares, como la inclusión de nuevos módulos funcionales, así como las mejoras o versiones posteriores.
2	Espacio en disco estimado para la base de datos de la GCSR9.	12 GB	N/A	El almacenamiento incluye los datos generados por Antares, así como los Logs de la operación diaria.
3	Crecimiento mensual estimado de la base de datos de la GCSR9, durante el primer año.	25 MB	40 MB	El crecimiento de la base de datos depende en gran medida, de la carga de trabajo de las áreas de atención a cuentas corporativas de la GCSR9.
4	Conexiones concurrentes esperadas a la base de datos de la GCSR9.	5	20	

No.	Aspecto	Mín.	Máx.	Comentarios e información adicional
5	Duración promedio estimada de las sesiones de usuario en la aplicación.	10 min	25 min	La excepción ocurre con el usuario administrador de Antares quien permanece en línea en un horario de 9:00 – 18:00 horas.
6	Número esperado de usuarios para la aplicación.	75	N/A	El número de usuarios aumentará paulatinamente y depende de la adición de cuentas corporativas, así como del registro de colaboradores de los centros de atención a clientes Telcel.
7	Número esperado de usuarios concurrentes en Antares.	5	20	
8	Procesamiento dedicado para el servidor de aplicaciones Weblogic 8.1.	5%	15%	Servidor Microsoft Windows 2003 server, con 4 GB de memoria RAM y 4 procesadores a 400 MHz.
9	Procesamiento dedicado para el DBMS MySQL 5.0	30%	45%	Servidor Microsoft Windows 2003 server, con 4 GB de memoria RAM y 4 procesadores a 400 MHz.
10	Productos de software y versiones requeridas para el máximo desempeño de Antares.	N/A	N/A	1.- Sistema operativo: Windows 2003 server. 2.- Servidor de aplicaciones Tomcat 5.0 y/o Weblogic 8.1. 3.- DBMS: MySQL 5.0. 4.- Máquina virtual de java 1.5.0.09.

b. Tiempos de respuesta:

Los beneficios de utilizar los marcos de trabajo, Struts, Hibernate y Spring repercuten directamente en el rendimiento de la aplicación, así como en los tiempos de respuesta en las peticiones realizados por los usuarios. El tiempo de respuesta del sistema depende de una gran variedad de factores, tanto a nivel de la infraestructura de telecomunicaciones, redes y de equipo de cómputo, como a nivel del software base de sistema operativo,

servidores de aplicaciones y servidores de base de datos, entre otros muchos factores y finalmente de la efectividad con la que la programación atiende el procesamiento de las cotas de volúmenes de operación. A este respecto los tiempos de respuesta aceptables para la GCSR9 y con los que deberá cumplir Antares son los siguientes:

1. Cuando haya hasta 10 usuarios conectados accediendo simultáneamente al sistema, su tiempo de respuesta no será en ningún momento superior a los 10 segundos por cada transacción realizada.
2. Ante un fallo en el software del sistema (Servidor de aplicaciones Tomcat 5.0 y/ó Weblogic 8.1), el administrador del sistema no tardará más de 60 minutos en restaurar los servicios que permitan el correcto funcionamiento de Antares.

c. Disponibilidad y horarios de servicio:

La disponibilidad de la aplicación (Las horas del día en las que se puede tener acceso al sistema y trabajar en él) estará solamente limitada por la normativa propia de Radio Móvil DIPSA en cuanto a los servicios ofrecidos para la atención a las Cuentas Corporativas. Antares está preparado para funcionar las 24 horas al día por los 7 días de la semana.

d. Plan de respaldos:

Inicialmente la planeación para la realización de respaldos de la base de datos de la GCSR9, en la que reside la información generada por Antares tendrá una periodicidad de 2 meses, realizándose el primer día de cada mes par del año en cuestión: febrero, abril, junio, agosto, octubre, diciembre.

Los respaldos son almacenados en discos duros portátiles, teniendo una carpeta por cada respaldo, las carpetas deberán respetar el formato "GCSR9_Antares_dd-mm-aaaa". Los respaldos deberán conservarse por los últimos 5 años, esto como apego a las políticas sobre el manejo de información propiedad de Radio Móvil DIPSA.

e. Control de acceso:

El sistema Antares tendrá su propio esquema de perfiles de usuarios, con acceso restringido a diferentes tipos de información y con permisos adecuados a la función que cada usuario tenga en el sistema. Los usuarios serán administrados por la GCSR9, entidad responsable de generar nuevos usuarios, así como la asignación del perfil y privilegios que los usuarios tendrán sobre el sistema.

CAPÍTULO V: FASE DE TRANSICIÓN

Siguiendo la metodología propuesta por el Proceso Unificado de Desarrollo de Software de Rational, en la fase de transición se persigue como objetivo esencial, la obtención de un sistema que ha alcanzado la capacidad operativa inicial; es decir, llegado a este punto el sistema constituye una versión del producto de software, misma que deberá ser liberada en un ambiente productivo. *“El líder de proyecto considera que el sistema ofrece la confianza suficiente como para operar en un entorno de usuario, teniendo en cuenta que el sistema es perfectible y mejorable en sus módulos, esto implica ser conscientes de que pueden surgir problemas, riesgos y defectos que no se han puesto en evidencia durante las pruebas realizadas al sistema”.* [11]

En esta fase los usuarios suelen agregar nuevas funcionalidades o cambios sobre el sistema, por lo que de acuerdo a su importancia, se evalúa si es factible agregarlos, o si dichas características propuestas afectan la planeación del proyecto y en cuyo caso conviene más integrarlos a una lista de requerimientos y características para el siguiente ciclo de desarrollo, lo que se considera como la siguiente versión del sistema.

5.1 LIBERACIÓN DEL SISTEMA ANTARES

La liberación del sistema Antares se centra en implantar la aplicación Web en su entorno de operación, específicamente es necesario generar el archivo “antares.war” que será publicado en los servidores de aplicaciones Tomcat 5.0 y Weblogic 8 de la GCSR9.

Una vez publicado el sistema los usuarios de la GCSR9 tienen la responsabilidad de probar la aplicación de tal forma que se reciba una retroalimentación que incluya los siguientes puntos:

1. Antares hace lo que demanda el negocio de Cuentas Corporativas de Radio Móvil DIPSA.
2. Detectar riesgos inesperados y no cubiertos por Antares.
3. Elaborar lista de incidencias y problemas no resueltos, enviarla a la GCSR9.
4. Revisión de este documento con la finalidad de detectar lagunas de documentación de usuario.
5. Recomendaciones para la formación y capacitación de los usuarios de Antares.

Una vez recibida esta información, se debe evaluar la factibilidad de modificar inmediatamente los componentes involucrados con estas incidencias o recomendaciones (JSP's, Servlets, Servicios, DAO's) y preparar nuevamente el producto para producción (empaquetar y publicar en servidor de aplicaciones). Dichas correcciones deberán ser corregidas en el marco de la línea base de la arquitectura existente.

En caso de considerar que los cambios tendrán un impacto superior a las 10 horas de programación, los mismos se agregarán a la lista de requerimientos a atender en el siguiente ciclo de desarrollo, esto debido a que el objetivo de esta fase del proceso de desarrollo es, encontrar y solucionar pequeñas deficiencias que pasaron desapercibidas durante la fase de construcción.

Finalmente, en atención al cliente Radio Móvil DIPSA, se le proporciona ayuda a los usuarios de la GCSR9 para utilizar el sistema Antares en forma efectiva, y se establece de común acuerdo con la GCSR9, que los usuarios utilizarán la aplicación y llevarán su operación en paralelo con las actividades a las que Antares reemplaza, durante un periodo de un mes a partir de la liberación. Tiempo suficiente para evaluar desempeño, así como la estabilización de la información en la base de datos.

5.2 CONFIGURACIÓN DE AMBIENTE

5.2.1 CONFIGURACIÓN DE TOMCAT 5.0

Las tareas que se mencionan a continuación requieren de tener acceso con rol de administrador al servidor de aplicaciones Web Tomcat 5.0, en el que se llevará a cabo la instalación de la aplicación; por lo que este debe encontrarse en ejecución y de esta manera poder acceder a la consola de administración del mismo. Los pasos se presentan a continuación:

1. Iniciar el servidor Tomcat 5.0: Ir a la ruta de instalación del producto, en este caso: **C:\java\Tomcat 5.0** e ingresar a la carpeta **bin**. Una vez en ella se debe ejecutar el script “**startup.bat**”, con lo que se abrirá la consola del Tomcat 5.0. Si el servidor inicia adecuadamente mostrará el mensaje “**Server startup**”.



```

inestampsCache
WARN [main] org.hibernate.cache.EhCacheProvider (EhCacheProvider.java:97) - Could not find configuration [org.hibernate.cache.UpdateTimestampsCache]; using defaults.
DEBUG [main] net.sf.ehcache.store.DiskStore (DiskStore.java:194) - Deleting data file org.hibernate.cache.UpdateTimestampsCache.data
DEBUG [main] net.sf.ehcache.store.MemoryStore (MemoryStore.java:147) - org.hibernate.cache.UpdateTimestampsCache Cache: Using SpoolingLinkedHashMap implementation
DEBUG [main] net.sf.ehcache.store.MemoryStore (MemoryStore.java:128) - initialized MemoryStore for org.hibernate.cache.UpdateTimestampsCache
DEBUG [main] net.sf.ehcache.Cache (Cache.java:277) - Initialised cache: org.hibernate.cache.UpdateTimestampsCache
DEBUG [main] org.hibernate.cache.EhCacheProvider (EhCacheProvider.java:100) - started EhCache region: org.hibernate.cache.UpdateTimestampsCache
INFO [main] org.hibernate.cache.StandardQueryCache (StandardQueryCache.java:50) - starting query cache at region: org.hibernate.cache.StandardQueryCache
WARN [main] org.hibernate.cache.EhCacheProvider (EhCacheProvider.java:97) - Could not find configuration [org.hibernate.cache.StandardQueryCache]; using defaults.
DEBUG [main] net.sf.ehcache.store.DiskStore (DiskStore.java:194) - Deleting data file org.hibernate.cache.StandardQueryCache.data
DEBUG [main] net.sf.ehcache.store.MemoryStore (MemoryStore.java:147) - org.hibernate.cache.StandardQueryCache Cache: Using SpoolingLinkedHashMap implementation
DEBUG [main] net.sf.ehcache.store.MemoryStore (MemoryStore.java:128) - initialized MemoryStore for org.hibernate.cache.StandardQueryCache
DEBUG [main] net.sf.ehcache.Cache (Cache.java:277) - Initialised cache: org.hibernate.cache.StandardQueryCache
DEBUG [main] org.hibernate.cache.EhCacheProvider (EhCacheProvider.java:100) - started EhCache region: org.hibernate.cache.StandardQueryCache
INFO [main] org.hibernate.impl.SessionFactoryImpl (SessionFactoryImpl.java:379) - Checking 0 named queries
29-mar-2009 11:19:53 org.apache.catalina.core.StandardHostDeployer install
INFO: Instalando aplicaciones web en trayectoria de contexto /jsp-examples desde URL file:C:\java\Tomcat 5.0\webapps\jsp-examples
29-mar-2009 11:19:53 org.apache.catalina.core.StandardHostDeployer install
INFO: Instalando aplicaciones web en trayectoria de contexto desde URL file:C:\java\Tomcat 5.0\webapps\ROOT
29-mar-2009 11:19:53 org.apache.catalina.core.StandardHostDeployer install
INFO: Instalando aplicaciones web en trayectoria de contexto /servlets-examples desde URL file:C:\java\Tomcat 5.0\webapps\servlets-examples
29-mar-2009 11:19:53 org.apache.catalina.core.StandardHostDeployer install
INFO: Instalando aplicaciones web en trayectoria de contexto /tomcat-docs desde URL file:C:\java\Tomcat 5.0\webapps\tomcat-docs
29-mar-2009 11:19:53 org.apache.catalina.core.StandardHostDeployer install
INFO: Instalando aplicaciones web en trayectoria de contexto /webdav desde URL file:C:\java\Tomcat 5.0\webapps\webdav
29-mar-2009 11:19:54 org.apache.catalina.core.StandardHostDeployer install
INFO: Instalando aplicaciones web en trayectoria de contexto /yui desde URL file:C:\java\Tomcat 5.0\webapps\yui
29-mar-2009 11:19:54 org.apache.coyote.http11.Http11Protocol start
INFO: Arrancando Coyote HTTP/1.1 en puerto http-8080
29-mar-2009 11:19:54 org.apache.jk.common.ChannelSocket init
INFO: JK2: ajp13 listening on /0.0.0.0:8009
29-mar-2009 11:19:54 org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=42/16 config=C:\java\Tomcat 5.0\conf\jk2.properties
29-mar-2009 11:19:54 org.apache.catalina.startup.Catalina start
INFO: Server startup in 24422 ms

```

Fig. 5.1 Consola de Tomcat 5.0.

- Ingresar a la URL del servidor por el puerto 8080, en este caso: <http://192.168.0.12:8080/>

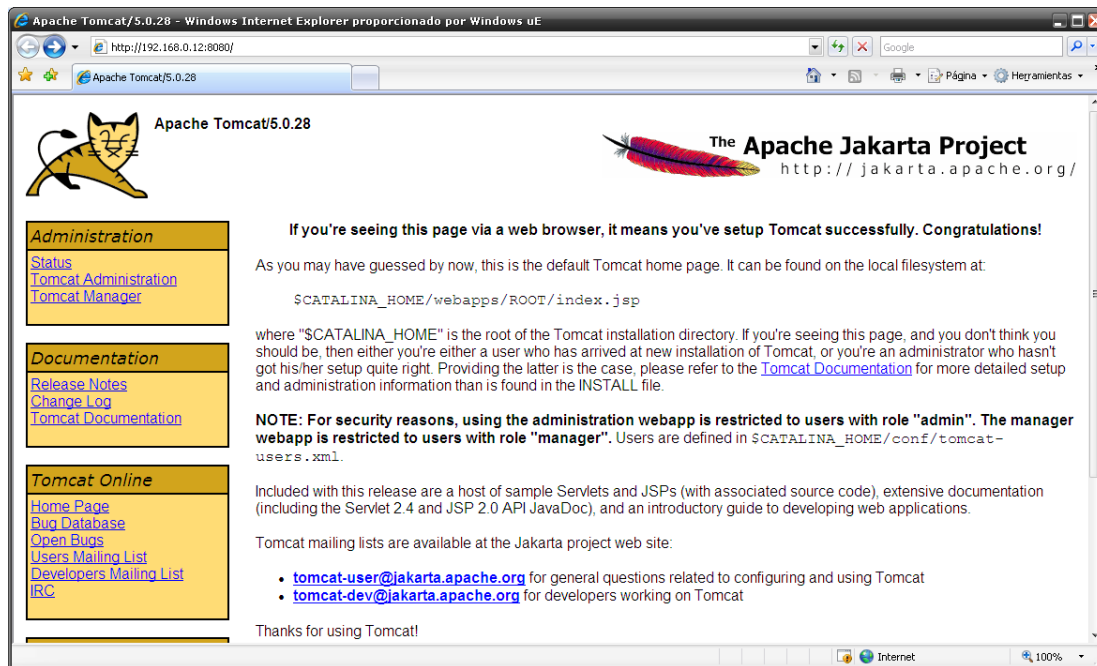


Fig. 5.2 Página de inicio del servidor de aplicaciones Tomcat 5.0.

- Ir a la liga "Tomcat Administration" en la que será necesario autenticarse con rol de usuario administrador.

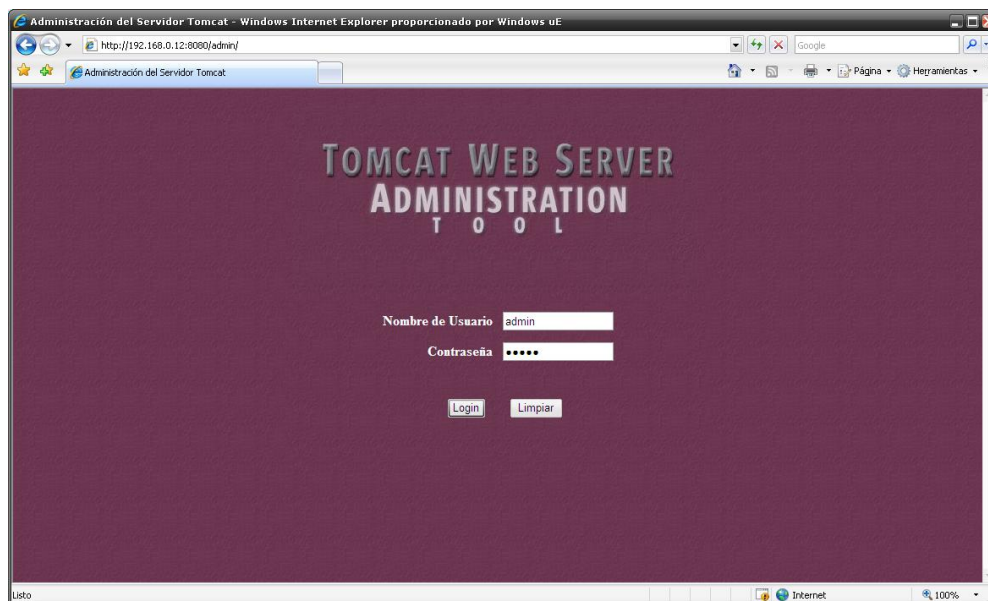


Fig. 5.3 Página de autenticación para la consola de administración.

4. Configurar la fuente de datos con la base de datos MySQL: Una vez en la consola de administración de Tomcat 5.0, ir a la sección correspondiente a “Fuentes de datos”. Posteriormente seleccionar del combo “Data Source Actions” la opción “Crear nueva fuente de datos” e ingresar la siguiente información de configuración:

- Nombre JNDI: antares_JNDI
- URL de fuente de datos: jdbc:mysql://localhost/antares
- Clase de manejador JDBC: org.gjt.mm.mysql.Driver
- Nombre de usuario: root
- Contraseña:
- Máx. Conexiones activas: 20
- Máx. Conexiones ociosas: 10

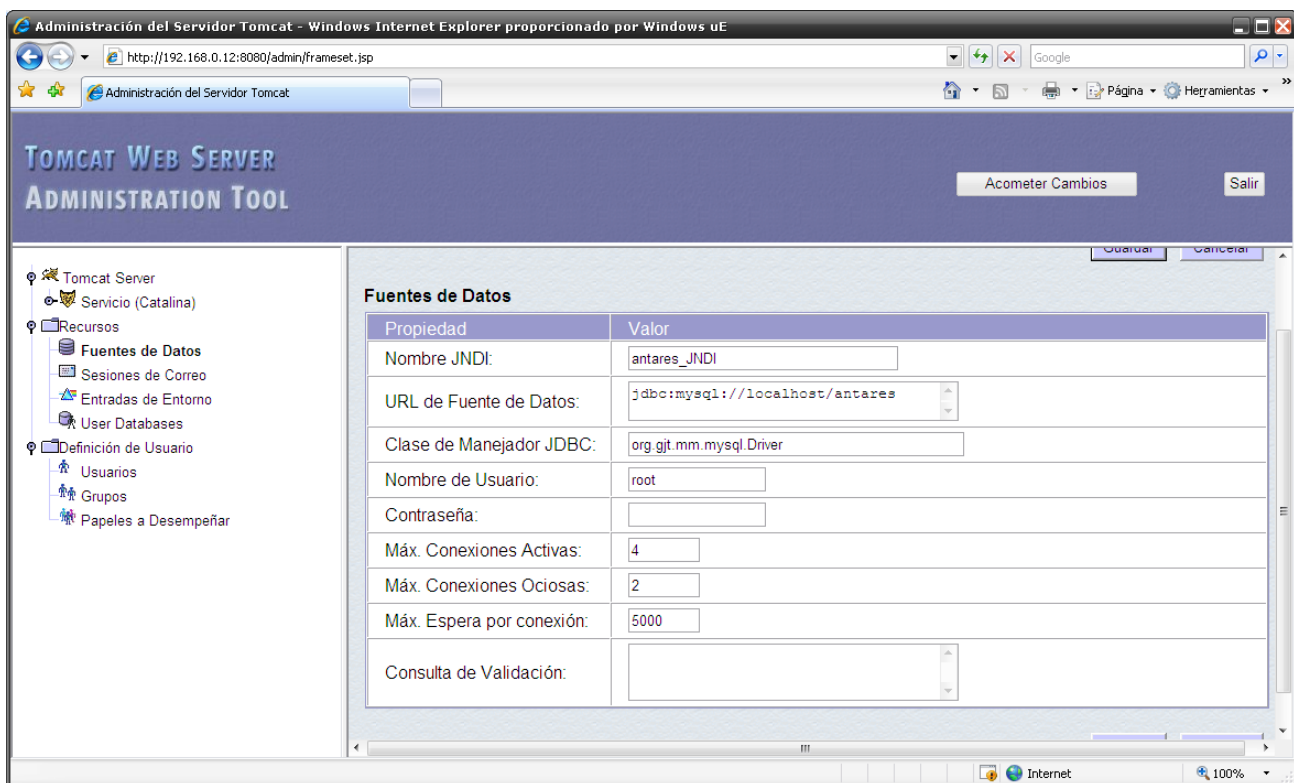


Fig. 5.4 Configuración de la fuente de datos con MySQL.

5.3 INSTALACIÓN DE SOFTWARE

La publicación del sistema Antares como plataforma J2EE se llevó a cabo utilizando la herramienta MyEclipse 5.0 de Genuitec. Esta herramienta permite:

- Administrar los archivos de código fuente que componen el proyecto WEB.
- Realizar el proceso de compilación del código fuente.
- Integración del proyecto para generar el archivo WAR o EAR mediante el cual se realiza el deployment (instalación) de la aplicación.

5.3.1 COMPILACIÓN DEL PROYECTO ANTARES

A continuación se presenta el ambiente de trabajo de MyEclipse, en donde se cargará el proyecto de Antares y se procederá a compilarlo para generar el archivo WAR requerido.

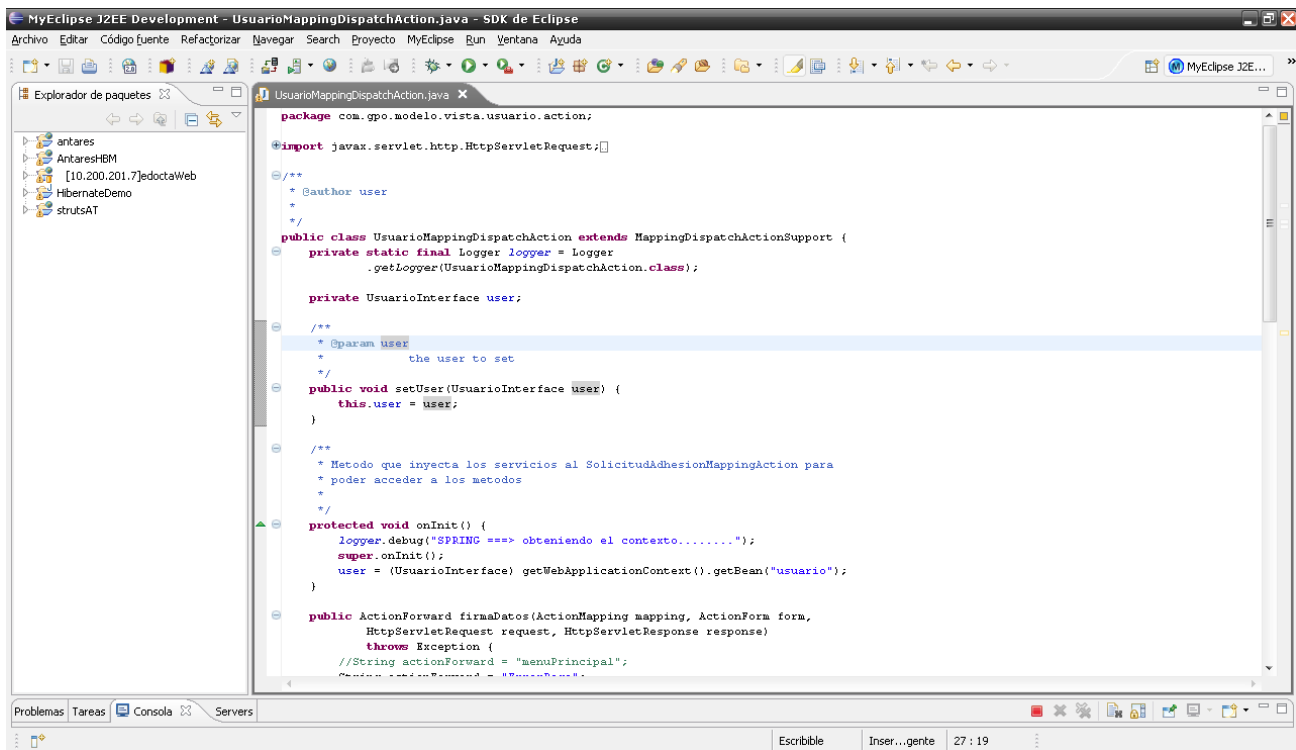


Fig. 5.5 Ambiente de trabajo de MyEclipse 5.0.

El procedimiento es el siguiente:

1. Importar el directorio local Antares en la maquina en la que se va a realizar la compilación. Este directorio contiene todo el código fuente, además de los archivos de configuración generados para poder ejecutar la aplicación en MyEclipse.
2. Ir a la opción “Archivo” de la barra superior de herramientas de MyEclipse y seleccionar la opción “Importar”. De la ventana que se muestra deberá seleccionar “Sistema de Archivos” y a continuación dar click en el botón “Siguiente”.

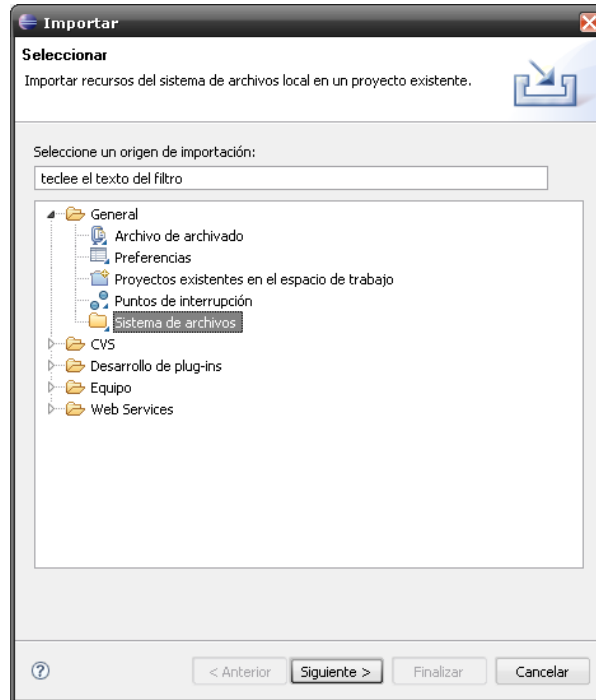


Fig. 5.6 Importar proyecto Antares.

3. Con el botón “Examinar” ir a la carpeta del proyecto y seleccionarla. Posteriormente debe elegir la opción “Crear estructura de carpeta completa” y dar click en finalizar.

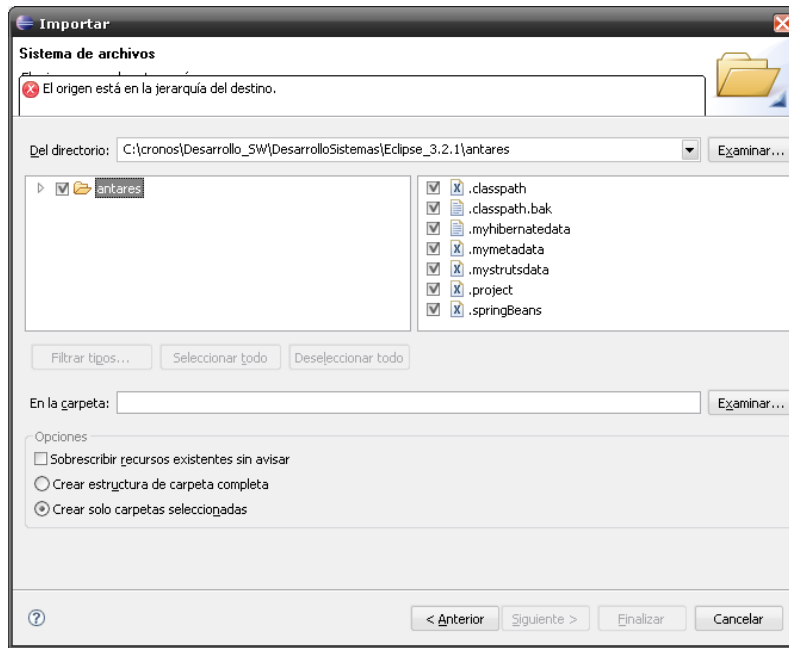


Fig. 5.7 Seleccionar la carpeta del proyecto Antares.

4. A continuación se cargará todo el ambiente de trabajo en MyEclipse. Mostrando en el recuadro superior izquierdo una estructura como la que se muestra en la siguiente figura.

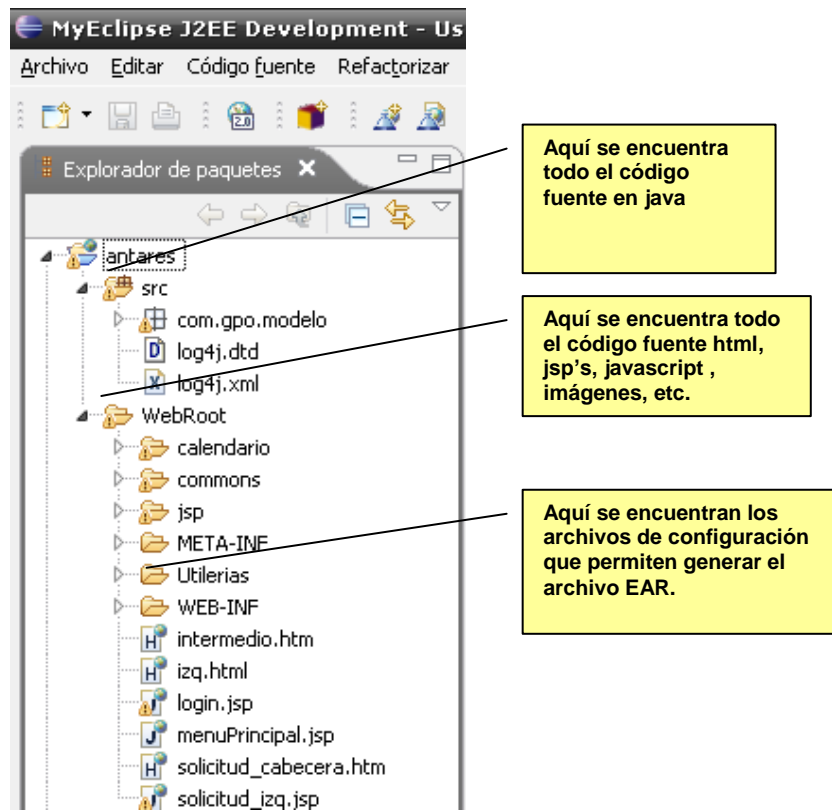


Fig. 5.8 Estructura del proyecto Antares.

5. Ir a la opción “Proyecto” de la barra superior de herramientas de MyEclipse y seleccionar “Construir proyecto”. Esto iniciará la compilación completa del proyecto Antares.

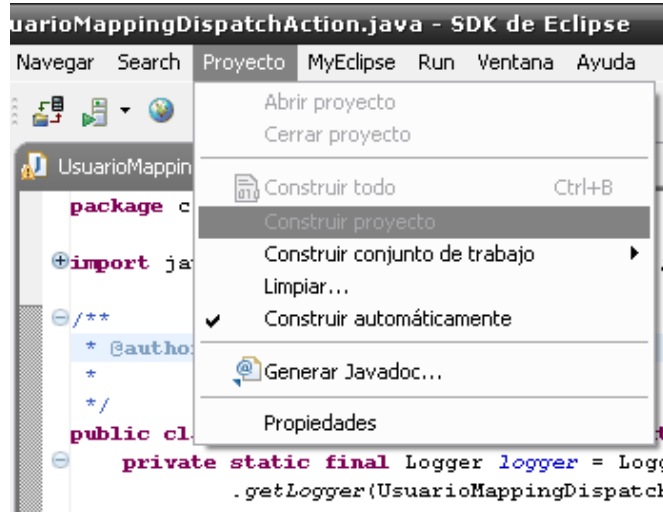


Fig. 5.9 Compilación del proyecto Antares.

6. En la parte inferior se mostrara una ventana de log indicando el progreso de la compilación. Al terminar ir otra vez al recuadro superior izquierdo y dar click derecho en el nodo global del proyecto Antares, ir a MyEclipse y elegir la opción “Add and Remove Project Deployments”. Esto iniciará el proceso que generará el archivo WAR necesario para la instalación.

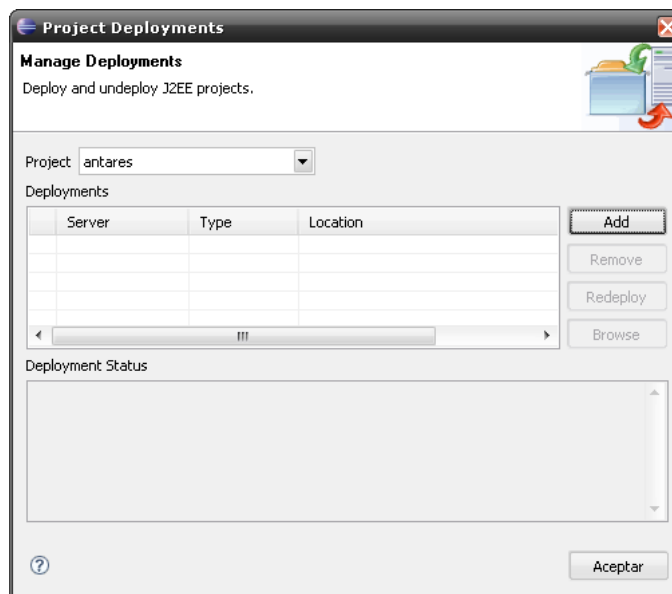


Fig. 5.10 Generar el archivo WAR necesario para la publicación de la aplicación.

7. Dar click en el botón “Add” y seleccionar el Servidor para el que se generará el archivo WAR necesario para la publicación del sistema Antares. Elegir la opción “Packaged Archive”.



Fig. 5.11 Elegir el servidor de aplicaciones para el que se generará el archivo WAR.

5.3.2 DEPLOYMENT DE ANTARES: PUBLICACIÓN

Una vez que se tiene el archivo WAR, se procede a realizar la instalación de la aplicación en el servidor Tomcat 5.0 en el que correrá la aplicación. Cabe mencionar que las tareas que se mencionan a continuación requieren de tener acceso con privilegios de administrador, al servidor de Tomcat 5.0 en el que se llevará a cabo la instalación de la aplicación. Por lo que este debe encontrarse en ejecución, para de esta manera poder acceder a la consola de administración del mismo. Procediendo con la siguiente secuencia de pasos:

1. Ingresar a la consola de administración del servidor Tomcat, para esto es necesario autenticarse:

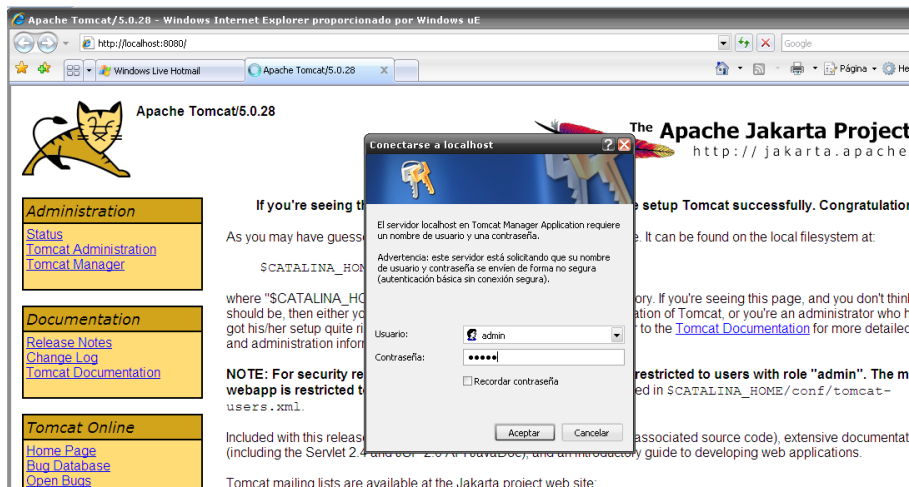


Fig. 5.12 Acceso a la consola de administración de Tomcat.

- Una vez accediendo a la consola de administración, ir al módulo de “Desplegar”.



Fig. 5.13 Desplegar el proyecto Antares.

- En la sección de “Archivo WAR a desplegar”, dar clic en el botón “Examinar”, elegir el archivo WAR generado previamente y dar clic en “Desplegar”.

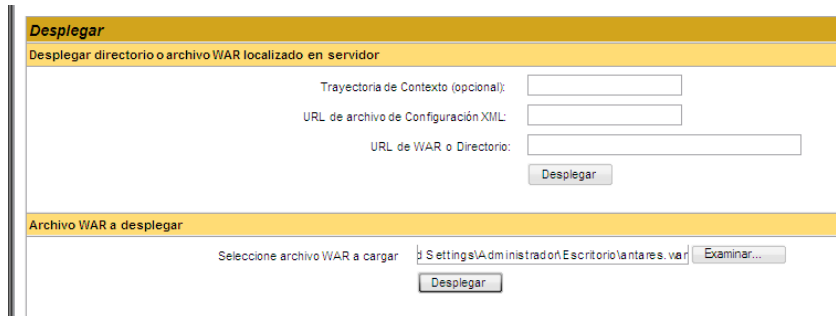


Fig. 5.14 Completar la publicación del proyecto Antares.

- Finalmente aparecerá el mensaje “OK - Desplegada aplicación en trayectoria de contexto /antares”, que confirma la correcta publicación del sistema Antares. En la sección “Aplicaciones”, aparecerá la aplicación Antares, teniendo como acciones disponibles en la consola de administración: Arrancar, Parar, Recargar y Replegar.

FASE DE TRANSICIÓN

The screenshot shows the Tomcat Manager web interface in a browser window. The title bar indicates it's running on Windows uE. The address bar shows the URL `http://localhost:8080/manager/html/upload`. The main heading is "Gestor de Aplicaciones Web de Tomcat". A message box displays: "Mensaje: OK - Desplegada aplicación en trayectoria de contexto /antares". Below this, there are navigation links: "Listar Aplicaciones", "Ayuda HTML de Gestor", "Ayuda de Gestor", and "Estado de Servidor". A table titled "Aplicaciones" lists various web applications with their status and session counts.

Trayectoria	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar
/admin	Tomcat Administration Application	true	0	Arrancar Parar Recargar Replegar
/antares		true	0	Arrancar Parar Recargar Replegar
/balancer		true	0	Arrancar Parar Recargar Replegar
/jsp-examples	JSP 2.0 Examples	true	0	Arrancar Parar Recargar Replegar
/manager	Tomcat Manager Application	true	0	Arrancar Parar Recargar Replegar
/servlets-examples	Servlet 2.4 Examples	true	0	Arrancar Parar Recargar Replegar
/tomcat-docs	Tomcat Documentation	true	0	Arrancar Parar Recargar Replegar
/webdav	Webdav Content Management	true	0	Arrancar Parar Recargar Replegar
/wui		true	0	Arrancar Parar Recargar Replegar

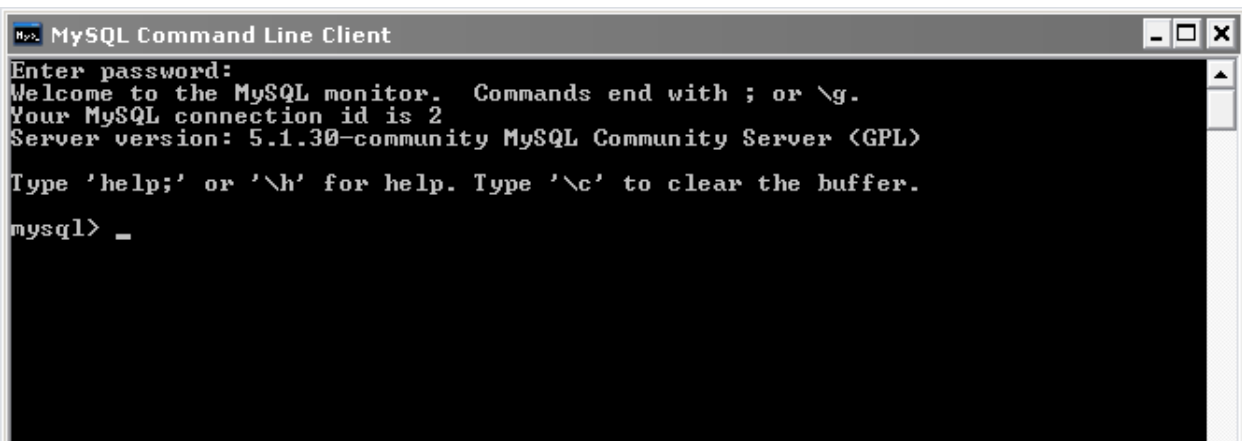
Fig. 5.15 Publicación del proyecto Antares exitosa

5.4 IMPLEMENTACIÓN DE BASE DE DATOS

5.4.1 CREAR LA BASE DE DATOS DE LA GCSR9

MySQL 5.1 es el manejador de base de datos (DBMS) utilizado por el sistema Antares, por lo que es necesario crear la base de datos de la GCSR9 en dicho servidor de base de datos. Para tal efecto se deben seguir los siguientes pasos:

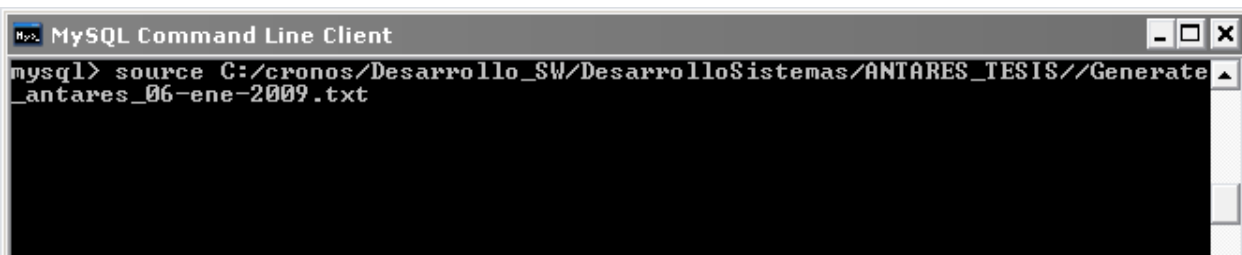
1. Ingresar a la consola de SQL de MySQL 5.1: En el servidor de la GCSR9, Windows 2003 Server, se debe dar click en Inicio → Todos los programas → MySQL → MySQL Server 5.1 → MySQL Command Line Client.



```
MySQL Command Line Client
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.30-community MySQL Community Server (GPL)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> _
```

Fig. 5.16 Consola del cliente MySQL 5.1

2. Ejecutar en la consola y mediante el comando source el script de instalación de la base de datos de la GCSR9, "Generate_antares_06-ene-2009.txt" que se encuentra en la ruta "C:\cronos\Desarrollo_SW\DesarrolloSistemas\ANTARES":



```
MySQL Command Line Client
mysql> source C:/cronos/Desarrollo_SW/DesarrolloSistemas/ANTARES_TESIS//Generate_antares_06-ene-2009.txt
```

Fig. 5.17 Ejecución del script de creación de la base de datos de la GCSR9 en MySQL 5.1

3. El contenido del script es el siguiente:

```
/*
Script generated by Aqua Data Studio 4.5.2 on ene-06-2009 06:07:18 PM
Database: antares
Schema: < All Schema >
Objects: TABLE, VIEW, PROCEDURE, FUNCTION
*/
ALTER TABLE usuario
    DROP CONSTRAINT PRIMARY
GO
ALTER TABLE usr
    DROP CONSTRAINT PRIMARY
GO
ALTER TABLE quejas_users
    DROP CONSTRAINT PRIMARY
GO
ALTER TABLE puesto_usuarios
    DROP CONSTRAINT PRIMARY
GO
ALTER TABLE depto_usuarios
    DROP CONSTRAINT PRIMARY
GO
ALTER TABLE cuentas_corp
    DROP CONSTRAINT PRIMARY
GO
ALTER TABLE cuentas_corp
    DROP CONSTRAINT fk_usr
GO
ALTER TABLE cuentas_corp
    DROP CONSTRAINT fk_area_solicitante
GO
ALTER TABLE bitacora
    DROP CONSTRAINT PRIMARY
GO
ALTER TABLE area_solicitante
    DROP CONSTRAINT PRIMARY
GO
DROP INDEX PRIMARY ON usuario
GO
DROP INDEX PRIMARY ON usr
GO
DROP INDEX PRIMARY ON quejas_users
GO
DROP INDEX PRIMARY ON puesto_usuarios
GO
DROP INDEX PRIMARY ON depto_usuarios
GO
DROP INDEX PRIMARY ON cuentas_corp
GO
DROP INDEX fk_usr ON cuentas_corp
GO
DROP INDEX fk_area_solicitante ON cuentas_corp
```

```

GO
DROP INDEX PRIMARY ON bitacora
GO
DROP INDEX PRIMARY ON area_solicitante
GO
DROP TABLE usuario
GO
DROP TABLE usr
GO
DROP TABLE quejas_users
GO
DROP TABLE puesto_usuarios
GO
DROP TABLE depto_usuarios
GO
DROP TABLE cuentas_corp
GO
DROP TABLE bitacora
GO
DROP TABLE area_solicitante
GO

CREATE TABLE area_solicitante (
    DEPTO_ENVIA_SOLICITUD          varchar(56) NOT NULL,
    GERENCIA_AREA_SOLICITANTE     varchar(128) NOT NULL,
    DEPTO_ENVIA_SOLICITUD_DESC    varchar(128) NOT NULL
)

GO
CREATE TABLE bitacora (
    id_bitacora                    int(11) AUTO_INCREMENT NOT NULL,
    folio                          varchar(128) NULL,
    comentario_historico          varchar(2048) NULL,
    archivo_adjunto               varchar(512) NULL,
    usuario_comentario            varchar(128) NULL,
    fecha_comentario              datetime NULL
)

GO
CREATE TABLE cuentas_corp (
    folio                          int(11) AUTO_INCREMENT NOT
NULL,
    fecha_solicitud                date NULL,
    fecha_intento_contacto         date NULL,
    fecha_contacto_establecido    date NULL,
    fecha_visita_cliente           date NULL,
    nombre_cuenta                  varchar(512) NULL,
    persona_envio_solicitud        varchar(128) NULL,
    solicitud_enviada_por_gerente  varchar(128) NULL,
    gerente_area_solicitante      varchar(128) NULL,
    tecnologia                     varchar(128) NULL,
    tipo_atencion                  varchar(128) NULL,
    descripcion_falla              varchar(2048) NULL,
    falla_encontrada               varchar(2048) NULL,
    acciones_solucionar            varchar(2048) NULL,
    fecha_aplicacion_cambios_para_solucion date NULL,
    solucion                       varchar(2048) NULL,
    responsable_gcsr9              varchar(128) NULL,

```

FASE DE TRANSICIÓN

```
observaciones                varchar(2048) NULL,
existe_reporte                varchar(255) NULL,
fecha_resp_entrega_reporte   date NULL,
fecha_solucion                date NULL,
gerencia_depto_atencion_cuenta varchar(128) NULL,
requiere_diseno               varchar(128) NULL,
estatus_general               varchar(128) NULL,
visita_conjunta_rf            varchar(1024) NULL,
celdas_servidoras_cuenta      varchar(256) NULL,
proyecto_solucionar_cuenta    varchar(256) NULL,
ext_o_cell_persona_envio_solicitud varchar(256) NULL,
depto_envia_solicitud         varchar(56) NULL,
nombre_contacto               varchar(128) NULL,
telefono_contacto             varchar(128) NULL,
region                         varchar(128) NULL,
entidad_federativa            varchar(128) NULL,
calle                         varchar(256) NULL,
numero                        varchar(128) NULL,
colonia                       varchar(128) NULL,
delegacion_o_municipio        varchar(128) NULL,
entre_calles                   varchar(256) NULL,
caracteristicas_inmueble      varchar(2048) NULL,
falla_registrada_en_domicilio varchar(1024) NULL,
la_falla_es_voz_datos_ptt_todas varchar(256) NULL,
desde_cuando_se_presenta      varchar(256) NULL,
num Equipos_con_falla         varchar(256) NULL,
num_lineas_contratadas        int(11) NULL,
num_lineas_por_contratar      int(11) NULL,
tiene_infraestructura_adicional_instalada varchar(128) NULL,
ubicacion_equipos             varchar(1024) NULL,
datos                         varchar(128) NULL,
falla_es_troncales            varchar(128) NULL
)
GO
CREATE TABLE depto_usuarios (
  id_depto_usuarios          int(11) NOT NULL,
  desc_depto_usuarios        varchar(255) NOT NULL
)
GO
CREATE TABLE puesto_usuarios (
  id_puesto_usuarios         int(11) NOT NULL,
  desc_puesto_usuarios       varchar(255) NOT NULL
)
GO
CREATE TABLE quejas_users (
  folio                      int(11) AUTO_INCREMENT NOT NULL,
  fecha_de_captura            date NOT NULL,
  no_empleado                 varchar(56) NOT NULL,
  nombre_empleado             varchar(256) NOT NULL,
  telefono_empleado           varchar(256) NOT NULL,
  correo_empleado             varchar(256) NOT NULL,
  num_celular                 varchar(56) NOT NULL,
  marca_modelo                varchar(256) NOT NULL,
  region                      varchar(56) NOT NULL,
  tipo_cliente                varchar(256) NOT NULL,
  nombre_cliente              varchar(256) NOT NULL,
```

```

telefono_cliente          varchar(256) NOT NULL,
plan_contratado           varchar(56) NOT NULL,
tipo_de_falla             varchar(56) NOT NULL,
falla_ocurrida_en        varchar(56) NOT NULL,
direccion                 varchar(256) NOT NULL,
entre_calles              varchar(512) NOT NULL,
estado                    varchar(256) NOT NULL,
del_o_municipio           varchar(256) NOT NULL,
colonia                   varchar(256) NOT NULL,
frecuencia_de_fallas     varchar(256) NOT NULL,
fecha_y_hora_falla        varchar(256) NOT NULL,
desea_ser_contactado     varchar(56) NOT NULL,
descripcion_falla         varchar(2048) NOT NULL,
asignacion                varchar(256) NULL,
responsable               varchar(256) NULL,
no_empleado_responsable  varchar(56) NULL,
departamento             varchar(256) NULL,
fecha_asignacion          date NULL,
num_celular_responsable  varchar(56) NULL,
estatus                   varchar(256) NULL,
tecnologia                varchar(256) NULL,
celda                     varchar(256) NULL,
sector                    varchar(56) NULL,
nombre_celda              varchar(256) NULL,
msc_bsc                   varchar(256) NULL,
descripcion_tecnica       varchar(2048) NULL,
acciones_efectuadas       varchar(2048) NULL,
comentarios               varchar(2048) NULL,
fecha_ultima_modificacion date NULL
)
GO
CREATE TABLE usr (
RESPONSABLE_GCSR9 varchar(128) NOT NULL,
NAME               varchar(128) NOT NULL,
ID_USR             varchar(56) NOT NULL,
PWD                varchar(56) NOT NULL,
DEPTO              varchar(128) NOT NULL,
PUESTO             varchar(128) NOT NULL,
EMAIL              varchar(128) NULL
)
GO
CREATE TABLE usuario (
ID_USUARIO int(11) AUTO_INCREMENT NOT NULL,
NOMBRE     varchar(254) NULL,
LOGIN      varchar(254) NULL,
PASSWORD   varchar(254) NULL
)
GO
CREATE UNIQUE INDEX PRIMARY
ON area_solicitante(DEPTO_ENVIA_SOLICITUD)
GO
CREATE UNIQUE INDEX PRIMARY
ON bitacora(id_bitacora)
GO
CREATE UNIQUE INDEX PRIMARY
ON cuentas_corp(folio)

```

```

GO
CREATE INDEX fk_usr
    ON cuentas_corp(responsable_gcsr9)
GO
CREATE INDEX fk_area_solicitante
    ON cuentas_corp(depto_envia_solicitud)
GO
CREATE UNIQUE INDEX PRIMARY
    ON depto_usuarios(id_depto_usuarios)
GO
CREATE UNIQUE INDEX PRIMARY
    ON puesto_usuarios(id_puesto_usuarios)
GO
CREATE UNIQUE INDEX PRIMARY
    ON quejas_users(folio)
GO
CREATE UNIQUE INDEX PRIMARY
    ON usr(RESPONSABLE_GCSR9)
GO
CREATE UNIQUE INDEX PRIMARY
    ON usuario(ID_USUARIO)
GO
ALTER TABLE area_solicitante
    ADD CONSTRAINT PRIMARY
    PRIMARY KEY (DEPTO_ENVIA_SOLICITUD)
GO
ALTER TABLE bitacora
    ADD CONSTRAINT PRIMARY
    PRIMARY KEY (id_bitacora)
GO
ALTER TABLE cuentas_corp
    ADD CONSTRAINT PRIMARY
    PRIMARY KEY (folio)
GO
ALTER TABLE cuentas_corp
    ADD CONSTRAINT fk_usr
    FOREIGN KEY(responsable_gcsr9)
    REFERENCES usr(RESPONSABLE_GCSR9)
GO
ALTER TABLE cuentas_corp
    ADD CONSTRAINT fk_area_solicitante
    FOREIGN KEY(depto_envia_solicitud)
    REFERENCES area_solicitante(DEPTO_ENVIA_SOLICITUD)
GO
ALTER TABLE depto_usuarios
    ADD CONSTRAINT PRIMARY
    PRIMARY KEY (id_depto_usuarios)
GO
ALTER TABLE puesto_usuarios
    ADD CONSTRAINT PRIMARY
    PRIMARY KEY (id_puesto_usuarios)
GO
ALTER TABLE quejas_users
    ADD CONSTRAINT PRIMARY
    PRIMARY KEY (folio)
GO

```

```
ALTER TABLE usr
  ADD CONSTRAINT PRIMARY
  PRIMARY KEY (RESPONSABLE_GCSR9)
GO
ALTER TABLE usuario
  ADD CONSTRAINT PRIMARY
  PRIMARY KEY (ID_USUARIO)
GO
```

5.4.2 CARGA DE INFORMACIÓN

Previo a la liberación del sistema Antares en un ambiente de producción, se requiere realizar una carga inicial de información en la base de datos de la GCSR9, misma que se compone de los catálogos y entidades indispensables para el adecuado funcionamiento de la aplicación.

La carga se hace a partir del script “inserts.sql” y su contenido es el siguiente:

```
INSERT INTO area_solicitante(DEPTO_ENVIA_SOLICITUD,
GERENCIA_AREA_SOLICITANTE, DEPTO_ENVIA_SOLICITUD_DESC)
  VALUES('1', 'HELP DESK', 'HELP DESK')
GO
INSERT INTO area_solicitante(DEPTO_ENVIA_SOLICITUD,
GERENCIA_AREA_SOLICITANTE, DEPTO_ENVIA_SOLICITUD_DESC)
  VALUES('2', 'VENTAS', 'VENTAS')
GO
INSERT INTO area_solicitante(DEPTO_ENVIA_SOLICITUD,
GERENCIA_AREA_SOLICITANTE, DEPTO_ENVIA_SOLICITUD_DESC)
  VALUES('3', 'CENTRALES', 'CENTRALES')
GO
INSERT INTO area_solicitante(DEPTO_ENVIA_SOLICITUD,
GERENCIA_AREA_SOLICITANTE, DEPTO_ENVIA_SOLICITUD_DESC)
  VALUES('4', 'TRONCALES', 'TRONCALES')
GO
INSERT INTO area_solicitante(DEPTO_ENVIA_SOLICITUD,
GERENCIA_AREA_SOLICITANTE, DEPTO_ENVIA_SOLICITUD_DESC)
  VALUES('5', 'RADIOBASES', 'RADIOBASES')
GO
```

FASE DE TRANSICIÓN

```
INSERT INTO area_solicitante(DEPTO_ENVIA_SOLICITUD,
GERENCIA_AREA_SOLICITANTE, DEPTO_ENVIA_SOLICITUD_DESC)
VALUES('6', 'NUEVAS TECNOLOGÍAS', 'NUEVAS TECNOLOGÍAS')
GO
INSERT INTO area_solicitante(DEPTO_ENVIA_SOLICITUD,
GERENCIA_AREA_SOLICITANTE, DEPTO_ENVIA_SOLICITUD_DESC)
VALUES('7', 'MANTENIMIENTO RAN', 'MANTENIMIENTO RAN')
GO
INSERT INTO usr(RESPONSABLE_GCSR9, NAME, ID_USR, PWD, DEPTO, PUESTO,
EMAIL)
VALUES('OLM', 'OSWALDO LOPEZ MARTINEZ', '10168', 'martinez',
'OPTIMIZACIÓN GENERAL DE RED', 'ADMINISTRADOR',
'pugabdk@hotmail.com')
GO
INSERT INTO usr(RESPONSABLE_GCSR9, NAME, ID_USR, PWD, DEPTO, PUESTO,
EMAIL)
VALUES('APL', 'ANDRES PUGA LUGO', '19450', 'lugo', 'OPTIMIZACIÓN
GENERAL DE RED', 'ADMINISTRADOR', 'pugabdk@hotmail.com')
GO
INSERT INTO depto_usuarios(id_depto_usuarios, desc_depto_usuarios)
VALUES(1, 'GERENCIA DE CALIDAD DE SERVICIO R9')
GO
INSERT INTO depto_usuarios(id_depto_usuarios, desc_depto_usuarios)
VALUES(2, 'OPTIMIZACIÓN GENERAL DE RED')
GO
INSERT INTO depto_usuarios(id_depto_usuarios, desc_depto_usuarios)
VALUES(3, 'OPTIMIZACIÓN DE RED ZONA NORTE')
GO
INSERT INTO depto_usuarios(id_depto_usuarios, desc_depto_usuarios)
VALUES(4, 'OPTIMIZACIÓN DE RED ZONA CENTRO')
GO
INSERT INTO depto_usuarios(id_depto_usuarios, desc_depto_usuarios)
VALUES(5, 'OPTIMIZACIÓN DE RED ZONA SUR')
GO
INSERT INTO depto_usuarios(id_depto_usuarios, desc_depto_usuarios)
VALUES(6, 'ÁREAS INTERNAS TELCEL')
GO
```



```
INSERT INTO puesto_usuarios(id_puesto_usuarios,
desc_puesto_usuarios)
VALUES(1, 'ANALISTA DE OPTIMIZACIÓN DE RED REGIÓN 9')
GO
INSERT INTO puesto_usuarios(id_puesto_usuarios,
desc_puesto_usuarios)
VALUES(2, 'GERENTE DE CALIDAD DE SERVICIO REGIÓN 9')
GO
INSERT INTO puesto_usuarios(id_puesto_usuarios,
desc_puesto_usuarios)
VALUES(3, 'JEFE DE OPTIMIZACIÓN DE RED REGIÓN 9')
GO
INSERT INTO puesto_usuarios(id_puesto_usuarios,
desc_puesto_usuarios)
VALUES(4, 'EMPLEADO GENERAL DE TELCEL')
GO
INSERT INTO puesto_usuarios(id_puesto_usuarios,
desc_puesto_usuarios)
VALUES(5, 'ADMINISTRADOR')
GO
```

CONCLUSIONES

El sistema Antares se desarrolló con base en la metodología RUP mediante la cual, se aborda la construcción de un sistema informático a través de una serie de iteraciones incrementales o versiones que permiten la evolución del sistema, incluyendo en cada una de ellas, mejoras y nuevos funcionamientos como resultado de la evaluación parcial del cliente (usuario final). Cada versión entregable de Antares cumplió con las cuatro fases fundamentales propuestas por la metodología: Inicio, elaboración, construcción y transición.

Antares, se sustenta en la plataforma empresarial de Java (J2EE), y considera la revisión de los principales componentes de esta tecnología, además de aplicar las mejores prácticas actualmente utilizadas para la construcción de aplicaciones empresariales robustas, confiables y portables, basadas en un proceso de desarrollo iterativo.

Para automatizar los procesos de negocio de la GCSR9 se utilizaron principalmente los patrones de diseño Modelo-Vista-Controlador, Delegación de Negocio y Composición, esto con la finalidad de lograr que Antares sea, un sistema empresarial flexible, fiable, fácil de extender y mantener.

Una prioridad para la GCSR9 es reducir costos, tanto operativos como de licenciamientos de software, en virtud de lo anterior la arquitectura de Antares está soportada por software libre:

1. Contenedor Web Tomcat 5.0 para publicar la aplicación y hacerla accesible vía internet.
2. Lenguaje de programación Java empleado para desarrollar el código fuente que implementa la lógica de negocio de los procesos de la GCSR9.
3. MySQL Server 5.1 como gestor de base de datos responsable de persistir la información de las cuentas corporativas de Radio Móvil DIPSA.

Finalmente mencionar que, se cumplieron satisfactoriamente los siguientes objetivos de la GCSR9:

- El sistema Antares permite centralizar y controlar la información generada de la atención y seguimiento de los proyectos de la GCSR9, permitiendo la colaboración de otras áreas de la empresa: Mesa de Ayuda, Ventas, Centrales, Troncales, Nuevas Tecnologías, Compras.

- Antares está construido sobre una arquitectura de desarrollo robusta y fácilmente extensible, lo cual permitirá automatizar e integrar en el menor tiempo posible, otros módulos operativos de la GCSR9.
- Dado que el sistema Antares es una aplicación WEB, provee una interfaz gráfica de usuario accesible vía remota que permite la gestión, seguimiento y consulta de información de los proyectos de la GCSR9. Apoya a altos mandos en la planeación y distribución de cargas de trabajo entre el personal a su cargo.
- Antares gestiona la información del negocio en una base de datos y no en archivos planos, tal y como lo hacía en el pasado la GCSR9. Situación que permite la disponibilidad, confiabilidad e integridad de la información en todo momento, aunado a que el acceso a la misma es más veloz, reduciendo con esto los riesgos de la posible pérdida de información de la empresa.
- Con el desarrollo de Antares, se provee a la GCSR9 de una herramienta destinada a Analistas, Jefes de área y personal general de la empresa, que les permite conocer el estatus y el seguimiento de los proyectos; así como automatizar las notificaciones y actualizaciones más trascendentes durante la atención de cada proyecto.
- Se permite conocer las acciones emprendidas en tiempo real por parte de los usuarios responsables de la atención de cada proyecto; para permitir así a los jefes y directivos la evaluación del desempeño del personal a su cargo.

Es para mí una gran satisfacción, el poder culminar el sistema Antares, debido a que se superaron diversas dificultades, tanto en el ámbito académico, como profesional. se cumplieron las expectativas funcionales de la GCSR9 y a nivel personal, el desarrollo de este sistema me ofreció la posibilidad de perfilar mi carrera profesional hacia el área de desarrollo de software; un área muy interesante que abordé con mucho entusiasmo a lo largo de mi estancia en la Facultad de Ingeniería de la UNAM.

Anexo A: GLOSARIO DE TÉRMINOS

1. API: Application Programming Interface (Interfaz de programación de aplicaciones).
2. B2B: Business to business (Negocio a negocio).
3. DAO: Data Access Object (Objetos de Acceso a Datos utilizados para interactuar con la capa de persistencia de Antares).
4. DTD: Document Type Definition (Definición de formato en XML para algún documento).
5. DBMS: Database Management System (Sistema gestor de base de datos).
6. DTO: Data Transfer Object (Objeto de transferencia de datos) Son objetos java empleados en la comunicación de las capas del sistema Antares, también se conocen como VO (Value Objects).
7. EAR: Enterprise Archive (Archivo que empaqueta una aplicación empresarial JEE, puede contener cero o varios archivos Web (WAR) y/o cero o varios archivos EJB (jar).
8. Framework: Marco de desarrollo de software que permite establecer reglas y comportamientos a los que deberá ajustarse una aplicación.
9. GoF: Gang of four (Banda de los cuatro, se refiere al grupo de desarrolladores que aportaron los patrones de diseño Orientados a Objetos).
10. Hibernate: Marco de desarrollo de software que establece las reglas de comunicación y persistencia de una aplicación empresarial, con un repositorio de información, tal como una base de datos.
11. HTML: Hyper-Text Markup Language (Lenguaje estático para marcado de hipertexto).

12. IDE: Integrated Development Environment (Entorno integrado de Desarrollo de Software).
13. J2EE: Java™ Platform, Enterprise Edition (Plataforma de desarrollo de Java para aplicaciones empresariales).
14. JAR: Java Archive (Paquete que almacena clases Java).
15. JDBC: Java DataBase Connectivity (API Java para la comunicación con gestores de bases de datos).
16. JNDI: Java Naming and Directory Interface (API Java que proporciona el servicio de directorio de búsqueda de recursos).
17. JSP: Java Server Pages (Tecnología Java que permite incrustar código Java en páginas HTML, para que sean procesadas en un contenedor Web).
18. JVM: Java Virtual Machine (Máquina Virtual de Java).
19. MVC: Acrónimo del patrón de diseño Modelo Vista Controlador.
20. PK: Primary Key (Columna de una tabla de base de datos que identifica de manera única a cada registro de dicha tabla).
21. POJO: Plain Old Java Object (Java Bean utilizado para transferir información entre las capas del sistema Antares. Utilizado también para el mapeo de entidades de la base de datos en objetos Java).
22. Spring: Es un marco de desarrollo de software Java/J2EE desarrollado usando licencia de OpenSource y tiene como directriz en su configuración, el uso de los JavaBeans. Es sumamente poderoso en la gestión de clases y componentes, aplica conceptos de la “Inversión de Control”.
23. SQL: Structured query language (Lenguaje estructurado para consultas y acceso a una base de datos).

24. Struts: Es un marco de desarrollo de software basado principalmente en la tecnología Java Servlet y en menor grado en Java Server Pages, situación que lo hace dependiente de un contenedor Web. Permite el desarrollo de una aplicación sujetándolo al patrón de diseño Modelo-Vista-Controlador.
25. TLD: Tag library descriptor (Librería de descriptores de etiquetas a utilizar en documentos HTML).
26. UML: Unified Modeling Language (Lenguaje de modelado unificado utilizado para documentar mediante diagramas, las fases del desarrollo de Software).
27. URI: Uniform Resource Identifier (Identificador uniforme de recursos, utilizado para identificar recursos dentro de un contenedor Web).
28. URL: Universal Resource Locator (Localizador universal de recursos. Permite identificar algún recurso residente en un servidor de aplicaciones y diferenciarlo al interior de una red).
29. WAR: Web Archive (Archivo que empaqueta una aplicación Web que ha de publicarse en un servidor de aplicaciones).
30. WWW: Acrónimo de "World Wide Web" (Red Global Mundial, es un sistema de documentos de hipertexto enlazados y accesibles a través de Internet).
31. XML: Extensible Markup Language (Lenguaje de marcado extensible).

Anexo B: Formato de Control de Cambios al Alcance

CONTROL DE CAMBIOS AL ALCANCE

I. REGISTRO DE SOLICITUD

Identificación del Proyecto	Clave
	Nombre
Fecha de Detección	
Reportado por	
Clave	
Resumen	
Descripción Detallada	

II. EVALUACION

Fecha Requerida	
Prioridad	0 (alta), 1 (media), 2(baja).
Costo Total	
Comentarios	
Resumen	

Subproyecto	Costo	Comentarios
Nombre del subproyecto	<i>Costo que implicaría el implantar el cambio solicitado en el subproyecto</i>	

III. IMPLANTACION

Decisión	Aceptado () Rechazado () Diferido ()
Autorizaciones	Nombre, fecha y firma de los responsables de autorizar para que proceda el cambio. Aquí van las firmas del cliente y del Líder de Proyecto o Gerente de Proyecto del área proveedora.

Anexo C: Formato de Evaluación de Satisfacción del Cliente

EVALUACION DE SATISFACCION DEL CLIENTE

I

Con el propósito de mejorar nuestros servicios, le agradeceremos se tome unos minutos para evaluar nuestro desempeño en el proyecto:_____.

¿Qué tan satisfecho está:

1. Considerando los aspectos relacionados con el contrato, proyecto, la solución, los recursos y la entrega de los servicios, por favor indique si nivel general de satisfacción.

- | | |
|--|---|
| <input type="checkbox"/> 5 Muy Satisfecho | <input type="checkbox"/> 2 Insatisfecho |
| <input type="checkbox"/> 4 Satisfecho | <input type="checkbox"/> 1 Muy Insatisfecho |
| <input type="checkbox"/> 3 Ni Satisfecho ni Insatisfecho | <input type="checkbox"/> 0 Sin Opinión |

Comentarios:

2. ¿El área proveedora, cubrió sus expectativas en todo lo relacionado al contrato, proyecto o servicios?

- | | |
|--|---|
| <input type="checkbox"/> 5 Muy Satisfecho | <input type="checkbox"/> 2 Insatisfecho |
| <input type="checkbox"/> 4 Satisfecho | <input type="checkbox"/> 1 Muy Insatisfecho |
| <input type="checkbox"/> 3 Ni Satisfecho ni Insatisfecho | <input type="checkbox"/> 0 Sin Opinión |

Comentarios:

3. ¿Se demostró (a través de nuestros servicios) que el área proveedora entendió las necesidades de su negocio en el diseño de la solución a través del contrato o proyecto?

- | | |
|--|---|
| <input type="checkbox"/> 5 Muy Satisfecho | <input type="checkbox"/> 2 Insatisfecho |
| <input type="checkbox"/> 4 Satisfecho | <input type="checkbox"/> 1 Muy Insatisfecho |
| <input type="checkbox"/> 3 Ni Satisfecho ni Insatisfecho | <input type="checkbox"/> 0 Sin Opinión |

Comentarios:

4. ¿De que se mantuvo una línea de comunicación estrecha y clara durante todas las fases del contrato o proyecto?

- | | |
|--|---|
| <input type="checkbox"/> 5 Muy Satisfecho | <input type="checkbox"/> 2 Insatisfecho |
| <input type="checkbox"/> 4 Satisfecho | <input type="checkbox"/> 1 Muy Insatisfecho |
| <input type="checkbox"/> 3 Ni Satisfecho ni Insatisfecho | <input type="checkbox"/> 0 Sin Opinión |

Comentarios:

5. ¿Con las habilidades y experiencia de los recursos humanos que trabajaron en este contrato o proyecto?

- | | |
|---|---|
| <input type="checkbox"/> 5 Muy Satisfecho | <input type="checkbox"/> 2 Insatisfecho |
|---|---|

CAPÍTULO V

- 4 Satisfecho 1 Muy Insatisfecho
 3 Ni Satisfecho ni Insatisfecho 0 Sin Opinión

Comentarios:

6. ¿Con la inversión realizada en este contrato?. Tratándolo al valor recibido en su organización con el proyecto.

- 5 Muy Satisfecho 2 Insatisfecho
 4 Satisfecho 1 Muy Insatisfecho
 3 Ni Satisfecho ni Insatisfecho 0 Sin Opinión

Comentarios:

7. ¿De que este contrato o proyecto fue administrado eficazmente?

- 5 Muy Satisfecho 2 Insatisfecho
 4 Satisfecho 1 Muy Insatisfecho
 3 Ni Satisfecho ni Insatisfecho 0 Sin Opinión

Comentarios:

8. ¿Del cumplimiento de los compromisos que hicimos respecto a la ejecución de los servicios o entrega e instalación de los productos?

- 5 Muy Satisfecho 2 Insatisfecho
 4 Satisfecho 1 Muy Insatisfecho
 3 Ni Satisfecho ni Insatisfecho 0 Sin Opinión

Comentarios:

9. En caso de requerir otros servicios en el futuro, escogería esta consultora como proveedor?

- 5 Casi Seguro 2 Probablemente No
 4 Probablemente 1 Seguramente No
 3 Indeciso 0 Sin Opinión

Comentarios:

10. En base a su experiencia en este contrato o proyecto ¿Recomendaría al área proveedora a otras organizaciones?

- 5 Casi Seguro 2 Probablemente No
 4 Probablemente 1 Seguramente No
 3 Indeciso 0 Sin Opinión

Comentarios:

Nombre:

Puesto:

Número Telefónico:

Firma:

Fecha:

A ser completado por el entrevistador:

Razón social del cliente:

Proyecto:

Responsable del Proyecto:

Fechas del Proyecto:

dd/mm/aaaa dd/mm/aaaa

Número de contrato (de aplicar):

Calificación Promedio:

Anexo D: Formato de Control de Asuntos

CONTROL DE ASUNTOS

Identificación del Proyecto Clave
Nombre

REGISTRO DE ASUNTO

Fecha de Detección Fecha de detección.
 Reportado por Persona que identifico el asunto y lo reporta.
 Clave Número Consecutivo.
 Descripción Detalle del asunto.

RESOLUCION

Responsable de Resolución Persona responsable de resolver la no conformidad.
 Resultado de la Evaluación Resultado de la evaluación del problema o asunto reportado.
 Plan de Acción Punto por punto lo que se va hacer para darle solución.

SEGUIMIENTO A ACCIONES

Fecha Seguimiento	% Avance	Reporte de Avance	Fecha Término	Fecha Revisión	No. Nota

Anexo E: Formato de Control de Activos

CONTROL DE ACTIVOS

Identificación del Proyecto Clave
Nombre

Registro de Activo

Clave No. asignado al activo.
 Fecha de Recepción Fecha en la que se recibe el activo.
 Descripción Características del activo.
 Origen Indicar quién proporciona el activo
 Entregado por Responsable de entregar el activo y de recibirlo al terminar el período de custodia
 Recibido por Responsable de recibir el activo.
 Ubicación Lugar en donde estará ubicado el activo.

Estatus de Recepción

Descripción	Especificaciones de Mantenimiento	Aprobación
<i>Condiciones del activo al recibirlo</i>	<i>Cuidados que se deben tener con el activo</i>	<i>Nombre y Firma de la Persona que recibe el activo</i>

Estatus de Entrega

Condiciones de Entrega	Aprobación
<i>Condiciones del activo al entregarlo</i>	<i>Nombre y Firma de la Persona que recibe el activo por parte del cliente.</i>

REFERENCIAS BIBLIOGRÁFICAS

- 1.- Jacobson, Ivar, Grady Booch, James Rmbaugh. El proceso unificado de desarrollo de software / Trad. Salvador Sánchez [et al] Madrid : Peason Addison Wesley, 2000. Pág. 6.
- 2.- Jacobson, Ivar, Grady Booch, James Rmbaugh. El proceso unificado de desarrollo de software / Trad. Salvador Sánchez [et al] Madrid : Peason Addison Wesley, 2000. Pág. 8.
- 3.- J. Keogh. J2EE Manual de referencia. Mc Graw-Hill, Madrid 2003. Pág. 3.
- 4.- J. Keogh. J2EE Manual de referencia. Mc Graw-Hill, Madrid 2003. Pág. 30.
- 5.- Stephen Stelting. Patrones de diseño aplicados a java. Pearson Educacional, Madrid 2003. Pág 27.
- 6.- Chuck Cavaness. JAKARTA STRUTS. O'Reilly Media, Madrid 2004. Pág 47.
- 7.- Jacobson, Ivar, Grady Booch, James Rmbaugh. El proceso unificado de desarrollo de software / Trad. Salvador Sánchez [et al] Madrid : Peason Addison Wesley, 2000. Pág. 170.
- 8.- Jacobson, Ivar, Grady Booch, James Rmbaugh. El proceso unificado de desarrollo de software / Trad. Salvador Sánchez [et al] Madrid : Peason Addison Wesley, 2000. Pág. 414.
- 9.- Jacobson, Ivar, Grady Booch, James Rmbaugh. El proceso unificado de desarrollo de software / Trad. Salvador Sánchez [et al] Madrid : Peason Addison Wesley, 2000. Pág. 414.
- 10.- Jacobson, Ivar, Grady Booch, James Rmbaugh. El proceso unificado de desarrollo de software / Trad. Salvador Sánchez [et al] Madrid : Peason Addison Wesley, 2000. Pág. 367.
- 11.- Jacobson, Ivar, Grady Booch, James Rmbaugh. El proceso unificado de desarrollo de software / Trad. Salvador Sánchez [et al] Madrid : Peason Addison Wesley, 2000. Pág. 378.

OBRAS CONSULTADAS

- Antipatrón de diseño [en línea] Consultado 13 de marzo 2008. Disponible en: http://es.wikipedia.org/wiki/Antipatr%C3%B3n_de_dise%C3%B1o
- Developin applications for the java TM EE platform : student guide with instructor notes. Santa Clara, Calif. : Sun Microsystems, 2006. 350 p.
- Como manejar transacciones con AOP Spring 2.5 y Hibernate [en línea] Consultado 24 de febrero 2009. Disponible en: <http://www.springhispano.org/?q=node/417>
- Java Pet Store 2.0 : reference Application, Early Access [en línea] Consultado 01 de junio 2008. Disponible en: <https://blueprints.dev.java.net/petstore/netbeans.html>
- Bobadilla Sancho, Jesús, Adela Sancho Hernández. Comunicaciones y bases de datos con JAVA a través de ejemplos. México : Alfaomega, 2003. 233 p.
- Brown,Dave, Peter Bahrs. Arquitectura empresarial para Ingenieros de Sistemas [en línea] Consultado 01 de junio 2008. Disponible en: <http://www.ibm.com/developerworks/ssa/rational/library/edge/09/jun09/enterprisearchitetur/index.html>
- Canales Mora, Roberto. Setup de Struts [en línea] Consultado 16 de abril 2007. Disponible en: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=struts>
- González Sánchez, Javier. Anti-patrones : la mejor forma de hacer un pésimo sistema de software [en línea] Consultado 17 de septiembre 2007. Disponible en: <http://www.sg.com.mx/content/view/407>
- Hall, Marty. Servlets y java server pages : guía práctica / Trad. David Garza Marín. México : Pearson Educación, 2001. 580 p.

- Jacobson, Ivar, Grady Booch, James Rumbaugh. El proceso unificado de desarrollo de software / Trad. Salvador Sánchez [et al] Madrid : Pearson Addison Wesley, 2000. 438 p.
- Johnson, Rod [et al] The Spring Framework - Reference Documentation [en línea] Consultado 10 de enero 2008. Disponible en: <http://static.springsource.org/spring/docs/2.0.x/reference/index.html>
- Keogh, Jim. J2EE : manual de referencia / Trad. Mario Camou Riveroll. Madrid : Mc Graw Hill, 2002. 803 p.
- Loney, Kevin, George Koch. Oracle 9i : manual de referencia / Trad. Vuelapluma. México : McGraw Hill, 2003. 1328 p.
- Medina Montenegro, Enrique. Cómo crear una aplicación con struts paso a paso [en línea] Consultado 16 de abril 2007. Disponible en: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=strutsb>
- Patzer, Andrew. JSP : ejemplos prácticos. Madrid : Anaya Multimedia, 2003. 317 p.
- Seta, Leonardo De. Transacciones con Spring Framework y EJB 3.0 [en línea] Consultado 10 de febrero 2009. Disponible en: <http://www.dosideas.com/java/414-transacciones-con-spring-framework-y-ejb-30.html>
- Stelting, Stephen, Olav Maassen. Patrones de diseño aplicados a Java / trad. Manuel Campos Martínez. México : Pearson : Prentice Hall, 2002. 586 p.
- Vicente, Raúl. Spring 1.2 : manejo de transacciones declarativo [en línea] Consultado 7 de junio 2008. Disponible en: <http://www.programania.net/otros/spring-12-manejo-de-transacciones-declarativo/>
- Vicente, Raúl. Spring 2.0 : manejo declarativo de transacciones [en línea] Consultado 7 de junio 2007. Disponible en: <http://www.programania.net/otros/spring-20-manejo-declarativo-de-transacciones/>