

UNIVERSIDAD SIMÓN BOLÍVAR

Ingeniería de la Computación

**SISTEMA PARA EL REGISTRO, CONTROL Y SEGUIMIENTO
DE SOLICITUDES DE UN HELP DESK**

POR

SERGIO AGUILERA DAMAS

INFORME FINAL DE CURSOS EN COOPERACIÓN
Presentado ante la Ilustre Universidad Simón Bolívar
como Requisito Parcial para Optar al Título de
Ingeniero en Computación

Sartenejas, Enero de 2005

UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

ACTA FINAL DE CURSOS EN COOPERACIÓN

**SISTEMA PARA EL REGISTRO, CONTROL Y SEGUIMIENTO
DE SOLICITUDES DE UN HELP DESK**

Presentado por:
Sergio Aguilera Damas

Este trabajo de cursos en cooperación ha sido aprobado en nombre de la
Universidad Simón Bolívar por el siguiente jurado examinador:

Prof. Carolina Chang
Jurado

Prof. Ángela DiSerio
Tutor Académico

Sartenejas, Enero de 2005

SISTEMA PARA EL REGISTRO, CONTROL Y SEGUIMIENTO DE SOLICITUDES DE UN HELP DESK

POR
SERGIO AGUILERA DAMAS

RESUMEN

En esta pasantía se desarrolló un producto de *software* para apoyar y automatizar las labores diarias y de administración de organizaciones dedicadas a la atención de solicitudes de soporte, caracterizado por ser completamente configurable y ajustarse considerablemente a la realidad de organizaciones diversas. *Net2Client Help Desk* - como fue llamado- es un sistema que permite el registro de peticiones de servicio por parte de los solicitantes y su posterior documentación y atención por parte de los analistas; proveyendo mecanismos para la toma de acciones sobre cada solicitud, durante su ciclo de vida, hasta ser solucionada. Igualmente, permite asociar esta actividad con el manejo de los recursos para los cuales se ofrece soporte y provee herramientas para hacer seguimiento al desempeño de la organización, controlando acuerdos y niveles de servicio y midiendo tiempos de atención y de respuesta.

El proceso de desarrollo siguió un ciclo de vida de tipo evolutivo e incremental, adaptando la metodología *Rational Unified Process* (RUP) a las circunstancias particulares de este proyecto.

Este sistema está orientado al ambiente Web y su diseño sigue el estilo arquitectónico de tres capas. El principal rasgo de su estructura es que está integrado a una plataforma de generación de aplicaciones que a su vez sirvió de herramienta para su elaboración. Está desarrollado en el lenguaje PHP con una interfaz de usuario en HTML y *JavaScript*, y apoyado sobre una base de datos *MySQL*. También se emplea un repositorio de archivos XML para la configuración del proceso definido por el sistema en cada instalación.

A Bernardo Aguilera Ballesteros (+),
por soñarme la vida todos tus días hasta el final.

y

A Mireya Damas,
mi ejemplo de persistencia y superación.

AGRADECIMIENTOS

A mi padre por haberme dado el privilegio de ser su laboratorio humano, por dedicarse a construirme como persona, por enseñarme tantas cosas y darme tanto. Gracias BAB por siempre soñar con mi éxito y felicidad, y hacer todo lo que estaba en tu poder por ayudarme a alcanzarlos. Gracias porque, tras tu partida, has sido el motor de todos mis esfuerzos, y tu orgullo la recompensa a perseguir en cada meta.

A mi madre, Mireya Damas, porque la palabra “apoyo” no es suficiente para expresar todo el aliento que me ha dado en cada paso de mi carrera y mi vida. Gracias por tener una parte de todos de mis logros; por impulsarme, guiarme, aconsejarme y estar siempre allí dispuesta a brindarme tu ayuda. Este éxito también te lo debo.

A Karina, Alexis y Eliana por ser desde un comienzo más que hermanos y haberme ayudado a llegar hasta acá desde aquellos días en que me enseñaban a leer.

A Karen por acompañarme siempre y apoyarme con paciencia, por estar siempre allí para escucharme y tenderme la mano.

A Hernán Rosas, Valmore Riera y Giovanni Savoca por darme tantas oportunidades de aprender y desarrollarme. Gracias por el apoyo incondicional, la confianza recibida y las enseñanzas que me han dejado.

A Olga Lorenzo por su preocupación y aliento, por servirme a la vez de supervisora y promotora durante el desarrollo de este proyecto.

A Jamile Track por guiarme y darme ese importante empujón inicial.

A Fernando López por cumplir una promesa indirecta y asumir la responsabilidad de apoyarme hasta el final. Gracias por ser el instrumento en la tierra de la mano protectora de mi padre.

A Virna por responder pacientemente todas mis preguntas y ayudarme a estar orientado durante este proyecto.

A Beta por ofrecerme toda la ayuda posible y estar siempre dispuesta a guiarme - y hasta presionarme- en este último tramo.

A Isis, por -sin saberlo- haberme enseñado tanto sobre la vida y haberme ayudado a poder enfrentar toda la experiencia de mi carrera universitaria.

A “Dios” -por falta de un mejor nombre- porque el mundo sería un lugar muy duro sin un puerto seguro al que podemos regresar en busca de fuerza. Gracias por tantas bendiciones, tanta suerte y tantas oportunidades de vivir.

ÍNDICE GENERAL

INTRODUCCIÓN	1
CAPÍTULO 1.- ENTORNO EMPRESARIAL	4
CAPÍTULO 2.- MARCO TEÓRICO	6
2.1 - <i>Help Desk</i> o Escritorio de Ayuda	6
2.2 - Acuerdos de Niveles de Servicio	9
2.3 - Uso de <i>software</i> para la administración de un <i>Help Desk</i>	10
2.4 - Modelo de Red de Servicios al Cliente	12
2.5 - Arquitectura de tres capas y aplicaciones Web	14
CAPÍTULO 3.- PLATAFORMA DE DESARROLLO <i>NET2CLIENT</i>	17
3.1 - <i>Net2Client</i> como herramienta de desarrollo	18
3.2 - Arquitectura de una aplicación generada	21
3.1 - <i>Net2Client</i> como plataforma	22
CAPÍTULO 4.- PLANTEAMIENTO DEL PROBLEMA	25
4.1 - Motivación	25
4.2 - ¿Por qué un <i>Help Desk</i> ?	26
4.3 - Descripción del problema	26
4.4 - Descripción de los objetivos	28

CAPÍTULO 5.- METODOLOGÍA DE DESARROLLO	30
5.1 - RUP: <i>Rational Unified Process</i>	30
5.2 - Ciclo de vida de RUP: Fases e Iteraciones	33
5.3 - Conceptos claves de RUP en el desarrollo de este proyecto	36
CAPÍTULO 6.- <i>NET2CLIENT HELP DESK</i>	37
6.1 - Fase de Inicio	37
6.2 - Fase de Elaboración	51
6.3 - Fase de Construcción	69
6.4 - Estado actual del sistema	71
CONCLUSIONES Y RECOMENDACIONES	75
REFERENCIAS BIBLIOGRÁFICAS	79
APÉNDICE A.- MANEJO DE REQUERIMIENTOS	80
APÉNDICE B.- REGLAS DE ENRUTAMIENTO DE SOLICITUDES	88

ÍNDICE DE TABLAS Y FIGURAS

TABLAS

Tabla 6.1 - Principales requerimientos de <i>Net2Client Help Desk</i>	39
Tabla 6.2 - Principales riesgos identificados	40
Tabla 6.3 - Plan de desarrollo de <i>software</i>	50
Tabla 6.4 - Plan revisado para la Fase de Construcción	68

FIGURAS

Figura 1.1 - Organigrama estructural de Global Web Technologies	5
Figura 3.1 - Arquitectura básica de una aplicación generada por <i>Net2Client</i>	21
Figura 3.2 - Ejemplos de sistemas integrados a <i>Net2Client</i>	23
Figura 5.1 - Arquitectura general de RUP	32
Figura 6.1 - Diagrama de actores	41
Figura 6.2 - Diagrama de Casos de Uso preliminares	44
Figura 6.3 - Diagrama de Estados general de una solicitud	47
Figura 6.4 - Diagrama de Estados de una solicitud para casos no exitosos	49
Figura 6.5 - Modelo Conceptual	52
Figura 6.6 - Modelo de Casos de Uso principales	56
Figura 6.7 - Modelo de Casos de Uso secundarios	59
Figura 6.8 - Diseño de la interfaz gráfica del prototipo evolutivo	60
Figura 6.9 - Arquitectura física de <i>Net2Client Help Desk</i>	62

LISTA DE ACRÓNIMOS Y ABREVIATURAS

- **CRM** Customer Relationship Management
- **GWT** Global Web Technologies (GlobalWebTek.com)
- **HTML** Hypertext Markup Language
- **N2C** Net2Client
- **N2C-HD** Net2Client Help Desk
- **PHP** PHP: Hypertext Preprocessor
- **RUP** Rational Unified Process
- **SFA** Sales Force Automation
- **SLA** Service Level Agreement
- **SLM** Service Level Management
- **SQL** Structured Query Language
- **UML** Unified Modeling Language
- **XML** Extensible Markup Language

INTRODUCCIÓN

Cuando la misión crítica de una organización es la de proveer un servicio de atención a las solicitudes de sus relacionados, cobra especial importancia que su interacción con ellos sea lo más eficiente posible. Por esto es necesario que todos los procesos que se realicen sean documentados apropiadamente para contar en todo momento con la información necesaria para lograr este objetivo.

Ese es el caso que se presenta en los Escritorios de Ayuda o *Help Desks*, por lo que cada vez se hace más indispensable el uso de herramientas de *software* para soportar su funcionamiento e incluso ayudar a mejorar el proceso mediante el cual se da soporte a sus clientes. Además, los avances en las nuevas tecnologías *Web* han ocasionado que surja en el mercado una abundante oferta de sistemas y herramientas para el manejo de *Help Desk*, orientadas a este ambiente. Sin embargo, el grueso de estos sistemas se enfoca hacia sectores muy específicos entre el conjunto de posibles escritorios de ayuda, a la vez que especifican un proceso de atención de solicitudes rígido que puede no ajustarse al funcionamiento interno de la mayoría de ellos.

El presente informe busca dar una descripción del producto *Net2Client Help Desk*, un sistema elaborado -como objetivo de este proyecto de pasantía- para proporcionar una solución a este problema, ofreciendo una alternativa a la oferta actual de *software* que resulte más flexible y adaptable a la realidad de un gran número de escritorios de ayuda. De allí que esta herramienta se caracterice por ser completamente configurable y ajustable para cada instalación.

La motivación principal detrás del desarrollo de este producto -y el especial enfoque hacia su adaptabilidad a *Help Desks* de diversos tipos- es que se construyó integrado a *Net2Client*, una plataforma de generación de sistemas Web y soluciones orientadas al soporte de las interacciones entre las organizaciones y sus clientes.

Net2Client Help Desk, a pesar de que es aún un sistema en la etapa final de su desarrollo, se ha convertido en un paquete perteneciente al conjunto de aplicaciones que acompaña a esta plataforma. Entre sus principales funcionalidades está el ofrecer mecanismos para el registro, control y seguimiento del proceso de atención a las solicitudes de un escritorio de ayuda cualquiera, así como también para hacer seguimiento a su desempeño interno e introducirle mejoras.

Debido a que la base de su elaboración fue una herramienta de generación de sistemas -orientada a la producción de prototipos rápidos y flexibles-, el ciclo de vida de este producto ha seguido un modelo evolutivo e incremental, aplicándose para su desarrollo una metodología que constituye una adaptación de un proceso de carácter iterativo como lo es el *Rational Unified Process (RUP)*.

Es precisamente la explicación detallada de este proceso de desarrollo, las circunstancias de su aplicación y los resultados que arrojó lo que compone el tema principal del presente informe. Adicionalmente se ofrecerá una descripción del producto elaborado, sus características, su estado actual y la evolución que seguirá. Todo este proceso será enfocado desde la perspectiva de lo que representó como proyecto de pasantía.

Para lograr tales objetivos el cuerpo de este informe se ha estructurado de la siguiente manera:

- **Capítulo 1 - Entorno empresarial:** En él se describen las circunstancias bajo las cuales se llevó a cabo este proyecto de pasantía; estableciendo las características de la empresa que la propuso y el ambiente dentro de ésta donde fue desarrollado.

- **Capítulo 2 - Marco teórico:** Los conceptos especiales más básicos, que deben manejarse para poder comprender el problema y las soluciones propuestas, son brevemente definidos y descritos en este capítulo.

- **Capítulo 3 - Plataforma de desarrollo *Net2Client*:** En él se ofrece una descripción general de la plataforma, sus características como herramienta de desarrollo, y la estructura básica de los sistemas que genera.

- **Capítulo 4 - Planteamiento del problema:** El problema del uso de *software* para la administración de *Help Desks* y las motivaciones detrás de esta pasantía son discutidas en este capítulo, donde también se definirán los objetivos de la misma.

- **Capítulo 5 - Metodología de desarrollo:** En este capítulo se describe la metodología que se aplicó y los elementos que se toman del proceso sobre el cual está basada. Igualmente se darán luces sobre el efecto que ocasiona en el ciclo de vida, el desarrollo a través de herramientas de generación de sistemas.

- **Capítulo 6 - *Net2Client Help Desk*:** Finalmente, en este capítulo se describirá el grueso de las actividades que se llevaron a cabo durante este proyecto de pasantía y los resultados que se lograron durante su desarrollo hasta la elaboración del producto final.

CAPÍTULO 1.- ENTORNO EMPRESARIAL

Global Web Technologies (GWT) es una empresa joven dedicada al desarrollo de *software* y soluciones de Tecnología de Información que buscan acercar a las organizaciones a sus clientes y relacionados.

Su misión es “...la creación de soluciones que permitan que las empresas den saltos cuánticos en la transformación de sus relaciones de negocio centrados en sus clientes” [GWT 2004].

Todos los desarrollos de GWT son completamente orientados al ambiente Web y las soluciones que ofrecen buscan destacarse por su bajo costo y por su facilidad y rapidez de implantación. Adicionalmente, están enfocados hacia el manejo de las “interacciones” con los clientes en oposición al manejo de las “transacciones” internas de las organizaciones.

GWT ofrece servicios de consultoría y desarrollos de proyectos a organizaciones y empresas de varias ramas en todo el país y en el exterior. Igualmente elabora productos de *software* de interés general.

Adicionalmente, la empresa ha desarrollado una plataforma de generación de sistemas Web que se ha convertido en el núcleo de casi la totalidad de sus proyectos. Esta plataforma, llamada *Net2Client*, representa una herramienta de desarrollo de *software* no sólo utilizada por la empresa como recurso tecnológico sino también por distribuidores asociados fuera de Venezuela. Su funcionamiento y estructura básica son descritos más adelante en el capítulo 3.

En la figura 1.1 se puede ver la estructura organizativa de la empresa; las divisiones que la componen y las dependencias entre ellas.

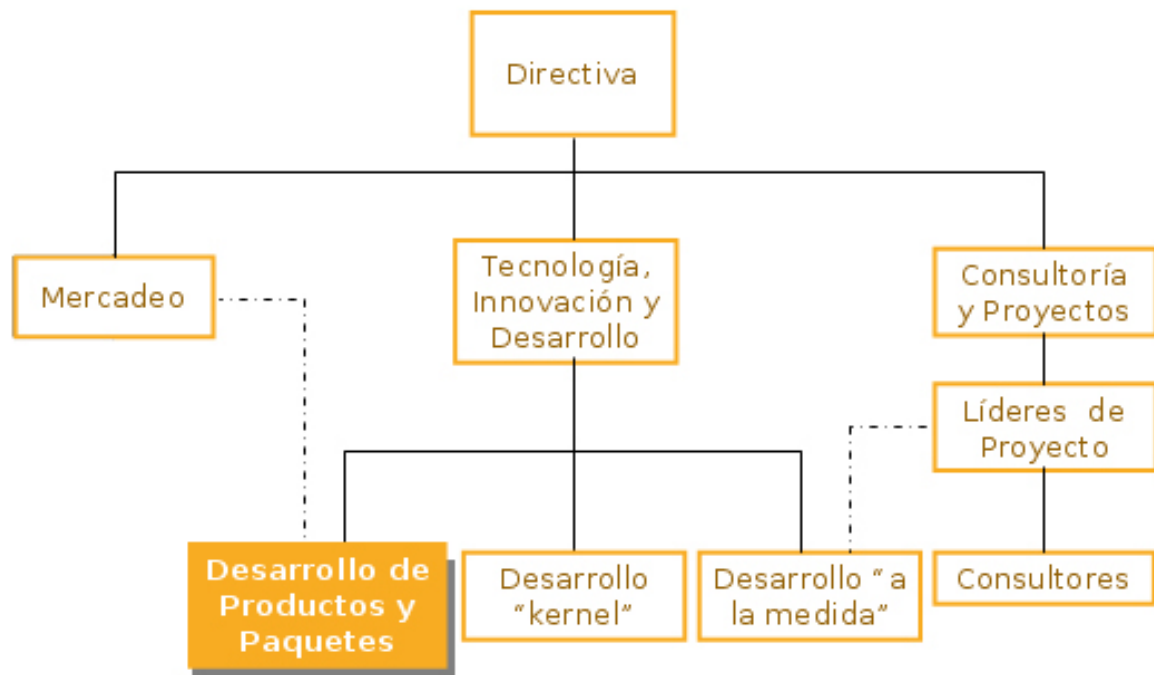


Figura 1.1 - Organigrama estructural de Global Web Technologies

El presente proyecto de pasantía se realizó en la división de “Desarrollo de Productos y Paquetes”, resaltada con fondo anaranjado en el organigrama presentado. En esta área se lleva a cabo el desarrollo de *software* que no está asociado a proyectos de consultoría. Es decir, aquí se elaboran los sistemas de información no hechos a la medida, que más que buscar satisfacer las necesidades de los clientes específicos tratan de cubrir una necesidad del mercado. Estos sistemas se integran a la plataforma de desarrollo creada por la empresa y se convierten en productos y paquetes que ésta ofrece a sus clientes, bien sea como valor agregado a los proyectos y servicios de consultoría, o simplemente como herramientas de interés general para automatizar sus procesos.

CAPÍTULO 2.- MARCO TEÓRICO

En este capítulo se ofrece una breve descripción de los aspectos teóricos asociados con el problema planteado en esta pasantía, que se deben manejar para una mejor comprensión de su desarrollo. Se busca proveer una definición concreta de lo que es un *Help Desk* y los conceptos e ideas relacionados, incluyendo un análisis superficial sobre la aplicación de *software* en su administración. Además se describirá el modelo de desarrollo hacia el cual se orientan los objetivos del sistema a desarrollar y la arquitectura bajo la cual fue concebido.

2.1 - *Help Desk* o Escritorio de Ayuda

Un *Help Desk* -o Escritorio de Ayuda- puede ser entendido en líneas generales como un departamento o una división dentro de una organización que se encarga de proveer ayuda a sus miembros para facilitar o garantizar el desempeño apropiado de sus funciones dentro de la misma. Típicamente, el *Help Desk* se encarga de dar mantenimiento y soporte técnico para el uso de ciertos bienes o recursos que una organización proporciona a sus integrantes como herramientas de trabajo.

El concepto de *Help Desk* también abarca los casos en que la ayuda se presta a los relacionados externos a la organización, generalmente sus clientes. Esto puede ser válido tanto para organizaciones cuya única misión es la de prestar el propio servicio de atención del *Help Desk*, como para aquellas que ofrecen a sus relacionados determinados recursos o bienes (típicamente productos que un cliente compra) para los cuales provee cierto soporte como valor agregado o como garantía de que se hará responsable de su mantenimiento. En ambos casos la ayuda ofrecida forma parte

integral del servicio o bien proveído y el *Help Desk* cumple la función de mantener el vínculo entre la organización y sus relacionados.

Así, el *Help Desk* funciona como un punto central donde los problemas, asuntos, quejas o solicitudes son reportadas, controladas, manejadas y atendidas [HDW 2004]. De manera que la ayuda que proporciona consiste en resolver cada una de las solicitudes que los clientes o empleados de la organización puedan hacer sobre los servicios que ella presta o los bienes que provee.

Dependiendo de la naturaleza de la organización, el tipo de solicitudes -y la forma de atenderlas y resolverlas- será distinto. El ejemplo más común es el de un *Help Desk* de Tecnología de Información (TI) dentro de una empresa, responsable de que los recursos tecnológicos asignados a cada empleado funcionen apropiadamente y de que cada quien cuente con las herramientas que requiere para realizar su trabajo. Usualmente, este tipo de *Help Desk* se constituye dentro del departamento de TI (o su equivalente) y los empleados de la empresa acuden a su personal para reportar fallas, solicitar actualizaciones o nuevos equipos, e incluso realizar consultas en torno al uso apropiado de los recursos y herramientas. Muy similares a este tipo de *Help Desk* son los comúnmente conocidos como Centros de Atención al Cliente, que reciben el mismo tipo de solicitudes que en el caso anteriormente descrito, sólo que ahora los solicitantes son los clientes de la empresa y los recursos o equipos son productos que le compraron, o para los cuales ella ofrece algún tipo de garantía o acuerdo de mantenimiento. El concepto también pudiese abarcar los casos en que esta ayuda se presta como un servicio independiente y no como un valor agregado a la compra del producto. Por ejemplo: talleres especializados de reparación de equipos.

Pero además de estos ejemplos, existen otros menos tradicionales que también pudieran verse desde la perspectiva de un *Help Desk*, y que están siendo tomados en cuenta como parte del concepto global manejado en esta pasantía. Tal es el caso de, por ejemplo, un centro de recepción de donativos de una organización cualquiera o el sistema de reservación de un restaurante.

De todo esto se puede inferir que existen ciertos elementos básicos que entran en funcionamiento en un concepto genérico de *Help Desk*. La clave, entonces, está en prestar ayuda o atención a un grupo de interesados a través del registro, control y resolución de solicitudes que van cumpliendo un ciclo de vida. Adicionalmente, está el soporte y mantenimiento ofrecido a determinados bienes, recursos o servicios proporcionados a estos interesados.

Si bien el proceso interno a través del cual un *Help Desk* atiende las solicitudes -y el ciclo de vida que llevan las mismas- varía para cada organización, existen ciertos conceptos y nociones básicas que son comunes a la mayoría de ellos, a saber:

- Las solicitudes son atendidas por uno o más **analistas**, quienes son las personas encargadas de prestar la ayuda que ofrece el *Help Desk*.
- Las solicitudes suelen ser **asignadas** a uno o más entes responsables de su resolución durante su ciclo de vida.
- Un *Help Desk* puede contar con varios **grupos** de analistas, cada uno atendiendo las tareas de un área específica. Estos pueden entenderse como **colas** de atención en las que aquellas solicitudes de tipos similares esperan a ser asignadas a alguno de los analistas que integran el grupo correspondiente.

- La atención de las solicitudes suele darse a través de frentes o **niveles** de servicio. Así, una solicitud en todo momento se encuentra en un nivel, y dependiendo de su progreso deberá o no escalar al siguiente, lo que puede ocasionar que sea atendida de una manera distinta o por un grupo diferente de analistas.
- Los *Help Desks* suelen contar con un conjunto de entes externos, a veces llamados contratistas o simplemente **terceros**, a quienes encargan la atención de cierto tipo de solicitudes.

2.2 - Acuerdos de Niveles de Servicio

Un concepto que a menudo se maneja dentro de la administración de los *Help Desks* es el de los Acuerdos de Niveles de Servicio, comúnmente conocidos como SLA's por sus siglas en inglés (*Service Level Agreements*). Un SLA es un contrato que se establece entre un ente que provee un servicio y aquel que se beneficia del mismo. En este contrato, el proveedor se compromete a ofrecer determinados bienes, recursos o servicios bajo un conjunto de condiciones que son debidamente especificadas. Igualmente, el beneficiado se compromete a cumplir con ciertas exigencias (en caso de que las hubiere) para poder recibir este servicio, y entre ambos acuerdan el proceso a través del cual se garantiza que sus interacciones en un futuro se darán bajo estas condiciones y exigencias.

Los SLA's son aplicados en muchas organizaciones que prestan servicios, pero cobran especial importancia en los *Help Desks*. En ellos, los SLA's se convierten en los documentos que dictan las pautas del funcionamiento entero de la organización. Generalmente, un *Help Desk* establece con cada cliente o interesado un SLA (o varios:

para cada tipo de solicitud o para cada recurso ofrecido) en el cual se especifican tiempos máximos de respuesta, sanciones o retribuciones para cuando no se cumplen y costo del servicio prestado.

De esta manera, los SLA's indican al personal del *Help Desk* qué tipo de solicitudes pueden recibir y cómo deben atenderlas. Más aun, a través de estos contratos se pueden especificar actividades de mantenimiento periódico a determinado conjunto de recursos que el *Help Desk* se compromete a realizar, lo que se traduce en solicitudes que su personal deberá atender. Adicionalmente, muchos administradores de *Help Desk* utilizan SLA's particulares para imponer controles y expectativas sobre el desempeño del personal encargado de las solicitudes. De ahí la importancia de la Gerencia de Niveles de Servicio o SLM (por sus siglas en inglés *Service Level Management*), una práctica comúnmente aplicada como herramienta para hacer seguimiento del funcionamiento global del *Help Desk* como organización, así como la satisfacción de sus clientes, y para aumentar su eficacia en la resolución priorizada de las solicitudes.

2.3 - Uso de *software* para la administración de un *Help Desk*

La labor de administrar un *Help Desk* requiere de una gran cantidad de actividades de gerencia. Entre ellas se cuentan el establecimiento adecuado de las solicitudes, el manejo de los SLA's, el seguimiento del desempeño del personal, el control de la eficacia en la respuesta, el aprovechamiento de la información y la experiencia (gerencia del conocimiento), el control del historial de vida de los bienes o recursos y el manejo apropiado de las relaciones con los clientes, entre otras.

Para lograr un control apropiado de estas actividades, el uso de *software* especializado se ha convertido en una herramienta para el soporte de todas las tareas asociadas a la administración de un *Help Desk*. Los distintos procesos involucrados en esta disciplina suelen estar apoyados por uno o más sistemas que optimizan el manejo de la información, promueven la generación y aplicación de conocimiento, y automatizan el control y seguimiento.

Esta creciente aplicación de *software* para gerencia de *Help Desk* responde a varias necesidades que han ido surgiendo progresivamente entre los administradores de estos Escritorios de Ayuda, su personal operativo y sus usuarios externos, y que pueden resumirse en esta lista de motivaciones:

→ Los solicitantes desean contar con un mecanismo que les permita conocer el estado de sus peticiones sin necesidad de interactuar de nuevo -probablemente a través de una llamada- con personal del *Help Desk*. Los administradores desean reducir el volumen de llamadas para proporcionar un acceso más sencillo y aumentar la productividad de su personal.

→ El personal del *Help Desk* desea contar con toda la información posible al momento de atender una solicitud o de contactar al interesado, y desea poder acceder a ella rápidamente para reducir sus tiempos de atención.

→ El solicitante no desea tener que describir su problema repetidas veces y explicarlo de nuevo a personas que no lo manejan.

→ El administrador desea poder hacer un seguimiento detallado del ciclo de vida de cada solicitud y contrastar los lapsos que tomó su resolución contra los acuerdos de niveles de servicio que dictan su funcionamiento esperado.

→ Resulta imperativo el poder llevar un control de los recursos sobre los que se presta el servicio de ayuda, a manera de poder realizar diagnósticos más informados y detectar problemas recurrentes.

→ El administrador y el personal desean contar con mecanismos de alertas para detectar vencimientos de plazos para la atención de solicitudes y otros sucesos que puedan requerir su atención y resolución.

→ Todos los actores buscan que cada solicitud tome el menor tiempo posible para ser resuelta, por lo que se desea contar con herramientas para aprovechar el conocimiento y experiencias previas, evitando que problemas recurrentes signifiquen tiempo de análisis y resolución innecesario, cuando podrían simplemente aplicarse las correcciones que han funcionado en el pasado.

2.4 - Modelo de Red de Servicios al Cliente

Con este nombre se denota a un mecanismo de gerencia de sistemas de la Tecnología de Información, basado en la aplicación de soluciones apoyadas en Internet, que consta de cuatro componentes principales:

◆ El ***Auto-servicio por parte del cliente*** se refiere a proveer al relacionado externo la capacidad de obtener la información más actualizada sobre los bienes, recursos y servicios ofrecidos. Generalmente, esta tendencia es asociada con la aplicación de comercio electrónico y el aprovechamiento de recursos como catálogos de productos en línea. Así, el comerciante puede actualizar los datos de su negocio directamente a través de un sistema automatizado que será el mismo responsable de interactuar con el cliente externo, quien cuenta, entonces, con un acceso mucho más

cómodo y disponible las 24 horas del día, evitando tener que contactar personal de ventas (lo que resulta más engorroso y además representa un gasto de tiempo innecesario para la organización).

◆ El concepto de ***Automatización de la Fuerza de Venta*** (o SFA, por sus siglas en inglés *Sales Force Automation*) implica la aplicación de soluciones de *software* para automatizar los procesos internos de las ventas, comúnmente conocido como *back office*. Esto trae consigo una disminución en el tiempo de respuesta del personal, mayor información para la toma de decisiones, estandarización de los mecanismos aplicados y una gran facilidad de seguimiento y evaluación para la gerencia. Todo esto se traduce en una mayor productividad.

◆ La ***Gerencia de las Relaciones con los Clientes*** (comúnmente llamada CRM por sus siglas en inglés *Customer Relationship Management*) a través de sistemas de información se ha convertido en una práctica influyente entre las empresas más exitosas. La idea general consiste en contar con mecanismos de automatización que permitan llevar un estricto control y seguimiento a la relación de la organización con sus clientes. Esto implica el registro detallado de cada interacción eventual y la disponibilidad, por parte del personal y en un tiempo mínimo, de historiales que reflejen toda la comunicación con un cliente. El poder administrar esta información y estas relaciones hace que sea la organización la que “recuerde” sus interacciones con el relacionado, independientemente de cuántos o cuáles miembros del personal lo hayan atendido. Esto disminuye la dependencia en los empleados, a la vez que mejora la experiencia de los clientes. Así se satisface el principio que afirma que para una

organización el poder conservar un cliente es mucho menos costoso y más valioso que adquirir uno nuevo.

◆ Finalmente, el ***Mercadeo Personalizado*** hace referencia a la aplicación de *software* para el manejo apropiado de la información disponible de cada cliente, que permita analizar sus intereses y ofrecerle un proceso de venta mucho más ajustado a sus necesidades.

2.5 - Arquitectura de tres capas y aplicaciones Web

Cuando se habla de la arquitectura de un sistema se hace referencia a su estructura en términos de los elementos que lo componen y las relaciones entre ellos. El concepto clave está en la definición del *software* en función de un conjunto de componentes -separados e independientes- que interactúan para cumplir las funciones del sistema que integran.

Así, de la misma manera que ocurre con las obras que produce la ingeniería civil, los sistemas y herramientas de *software* también pueden clasificarse según el estilo arquitectónico que poseen. Uno de estos estilos es el que identifica los componentes de *software* como capas o estratos que se superponen, de manera que cada uno se encarga de realizar ciertas operaciones y mediar entre su ambiente externo y el interno [RSP 2002].

El creciente auge que ha tenido el desarrollo de aplicaciones para ambiente Web ha demostrado la existencia de una tendencia en esta área a aplicar un estilo arquitectónico estratificado; más específicamente lo que se ha definido como “Arquitectura de tres capas”.

Este término no es más que la referencia a un caso específico dentro de las posibles arquitecturas de *software*. Y es que gran parte de los nuevos sistemas, especialmente los orientados al ambiente Web, están coincidiendo en definir los componentes de su estructura como tres capas independientes que se comunican entre sí por mecanismos mediadores. En estos casos los estratos identificados son siempre los mismos, independientemente de cómo sean nombrados, a saber:

- Capa de datos: responsable del almacenamiento y organización de toda la información persistente.

- Capa de lógica: agrupa a todos los componentes que representan las reglas del negocio del sistema, por lo que se encarga de la ejecución de todas las acciones.

- Capa de Interfaz: es la más superior y, por tanto, la que entra en contacto directo con el usuario. Se encarga de recopilar los datos que éste introduce, así como de activar las acciones que solicite y presentarle los resultados de forma apropiada.

Las razones por las cuales este estilo arquitectónico ha predominado en el desarrollo de aplicaciones Web probablemente tengan su origen en dos aspectos inherentes a este mecanismo de comunicación entre *software* y usuario:

- ☑ El usuario se encuentra en una locación remota, interactuando únicamente con una aplicación de navegación Web que sólo permite el despliegue de datos en el formato HTML. Siendo éste un lenguaje únicamente de presentación, es natural que el aspecto de interfaz sea separado en un componente independiente de los procesos de lógica, los cuales deben implementarse utilizando otros lenguajes para ser activados

por esta capa de interfaz y ejecutarse en un ambiente completamente separado a su ámbito de acción.

☑ Para poder activarse a través de mecanismos Web, los procesos de lógica deben encontrarse dentro de algún servidor. Sin embargo, pocos integran a sus opciones el manejo de datos, pues esta funcionalidad escapa a los intereses básicos de la prestación de servicio Web. Por ello, el control de la información persistente suele extraerse y dejarse a cargo de algún manejador de base de datos, al cual se accede desde la capa de lógica a través de interfaces apropiadas.

CAPÍTULO 3 – PLATAFORMA DE DESARROLLO *Net2Client*

Net2Client es una plataforma de generación de aplicaciones y sistemas orientados al ambiente Web. Puede entenderse como un manejador de base de datos de alto nivel que permite -con gran facilidad y rapidez- implementar estructuras de datos orientadas a un modelo relacional integrando a ellas el manejo de seguridad y control de acceso a la información. Provee también de manera automática todos los mecanismos, aptos para usuarios finales, de modificación y consulta de estas estructuras. Igualmente, ofrece herramientas para agregar aspectos de lógica de negocios al manejo de todos los datos (a través de la generación de reportes, búsquedas y tareas de administración) y lo encapsula todo tras una interfaz gráfica, generando así automáticamente una aplicación Web.

A un nivel más avanzado, *Net2Client* cuenta con un conjunto de interfaces, librerías y utilitarios que permiten la ampliación de las funcionalidades básicas generadas por la plataforma para implementar sistemas más complejos que se apoyen sobre su estructura. A través de estas herramientas se puede acceder a los datos persistentes de manera segura y el control de acceso, en lo concerniente a niveles de autorización y perfiles de usuario, es manejado por *Net2Client* de forma transparente.

Net2Client está desarrollado con herramientas de *software* libre: sus procesos están codificados utilizando el lenguaje PHP y las estructuras de datos están soportadas por el manejador de base de datos *MySQL*. La interfaz gráfica está conformada por un conjunto de páginas HTML, que a menudo cuentan con cierto grado de dinamismo ofrecido por instrucciones del lenguaje *JavaScript*.

3.1 - *Net2Client* como herramienta de desarrollo

Para comprender mejor el funcionamiento de *Net2Client* veamos paso a paso cómo es el proceso de generación de una aplicación básica:

→ En primera instancia, el *asesor* (término que utiliza *Net2Client* para referirse a los desarrolladores) accede a la plataforma y define la existencia de la nueva aplicación a generar, asignándole un código que la representará en todo momento.

→ Posteriormente, entra al módulo de administración a través del cual “crea” la aplicación, lo que en verdad se traduce en la generación automática de una serie de estructuras (tablas) de control dentro de la base de datos de la plataforma, que almacenarán toda la configuración especificada para el nuevo sistema.

→ A partir de este momento, el *asesor* puede comenzar a definir los conceptos asociados al problema específico de su aplicación. Esto significa que puede indicar tanto la estructura interna como los valores de definición de las tablas relacionales que conformarán la base de datos del sistema. La especificación de estas estructuras se hace a muy alto nivel, pudiendo, por ejemplo, definir campos tipo “*Email*” -que garantizará que sólo se introduzcan valores que cumplan con el formato de una dirección de correo electrónico- o incluso tipo “*Lista*” -que permite asociarle un conjunto de valores correctos que puede ajustarse en cualquier momento-. Igualmente se pueden dar indicaciones sobre el formato de presentación de los listados y pantallas de introducción de datos de cada tabla. Todo esto se hace a través de una sencilla interfaz gráfica que permite que esas estructuras sean modificadas cualquier cantidad de veces durante todo el proceso de desarrollo.

→ Esta definición de las tablas pasa a engrosar un conjunto de parámetros que incluye también aspectos de configuración de la aplicación Web propiamente dicha (como por ejemplo nombres y colores). Son estos parámetros la verdadera esencia de la aplicación, pues *Net2Client*, más que generar código nuevo para cada aplicación desarrollada, reutiliza los mismos programas y procesos básicos, que en todo momento son informados del código de la aplicación para la cual están siendo utilizados y ajustan así su comportamiento para adaptarse a lo que indique su configuración.

→ De esta manera, el asesor puede ya acceder al punto único de ingreso de la plataforma *Net2Client*, indicando el código de su aplicación nueva. Una vez que entre, ya estará en funcionamiento, reflejando toda la configuración que se realizó en el módulo de administración.

→ A partir de entonces pasa a tener importancia el rol del *administrador* (nuevamente un término que hace referencia al desarrollador, pero que en este caso no genera la aplicación sino que la ajusta y mantiene), quien en primera instancia definirá los usuarios del nuevo sistema y sus perfiles de seguridad. Esta definición de perfiles permitirá agrupar varios usuarios y controlar su acceso a las estructuras de datos definidas, tanto por campos -mediante niveles de restricción que se asocian a ellos- como por registros -a través de condiciones lógicas específicas-. También se puede indicar para cada perfil si tendrá acceso a ciertas funcionalidades que la plataforma integra al sistema automáticamente, como por ejemplo el envío de correos electrónicos y la generación de reportes.

→ Ya para este momento puede considerarse que la aplicación fue completamente generada. Cada vez que uno de sus usuarios entre al nuevo sistema, se

creará un identificador de sesión que lo acompañará en todo momento al pasar entre las distintas pantallas de la plataforma, indicándole a éstas en todo momento cuál es la aplicación que se está mostrando (es decir, cuál es el valor actual del conjunto de parámetros que regirán su comportamiento) y cuáles son los niveles de acceso y permisos vigentes para dicha sesión (es decir, la especificación del perfil que está asociado al usuario en este momento).

→ Finalmente nos encontramos con el paso en que la herramienta le da independencia al desarrollador y le permite definir su lógica de negocios con libertad para integrarla a la aplicación generada. Esto, en dos niveles: el primero es indicando, a través de parámetros, que se debe ejecutar un cierto código (en PHP) cuando ocurran determinadas operaciones sobre un conjunto específico de datos; el segundo es desarrollando programas completamente independientes -utilizando los mismos lenguajes de programación de *Net2Client* (*JavaScript*, HTML, PHP e incluso SQL)- que pueden ser invocados por los programas de la plataforma desde el punto en que se le indique en la configuración de los parámetros.

Es importante destacar que estos programas elaborados para cada aplicación funcionarán siempre dentro del ambiente de *Net2Client* (a través del mecanismo de control de sesiones), por lo que en todo momento podrán acceder a las estructuras de datos y a la configuración mediante un conjunto de rutinas, librerías e interfaces que garantizan siempre el control de acceso y los niveles de seguridad que habían sido definidos desde un comienzo.

3.2 - Arquitectura de una aplicación generada

En la figura 3.1 se puede observar la arquitectura básica de una aplicación generada por *Net2Client*. Cada una de las tres capas que la conforman (datos, procesos y presentación) pertenece a la plataforma como tal, pero ajusta su comportamiento en función de la configuración y parámetros específicos de la aplicación correspondiente, representada en la gráfica por el recuadro gris que las contiene en todo momento.

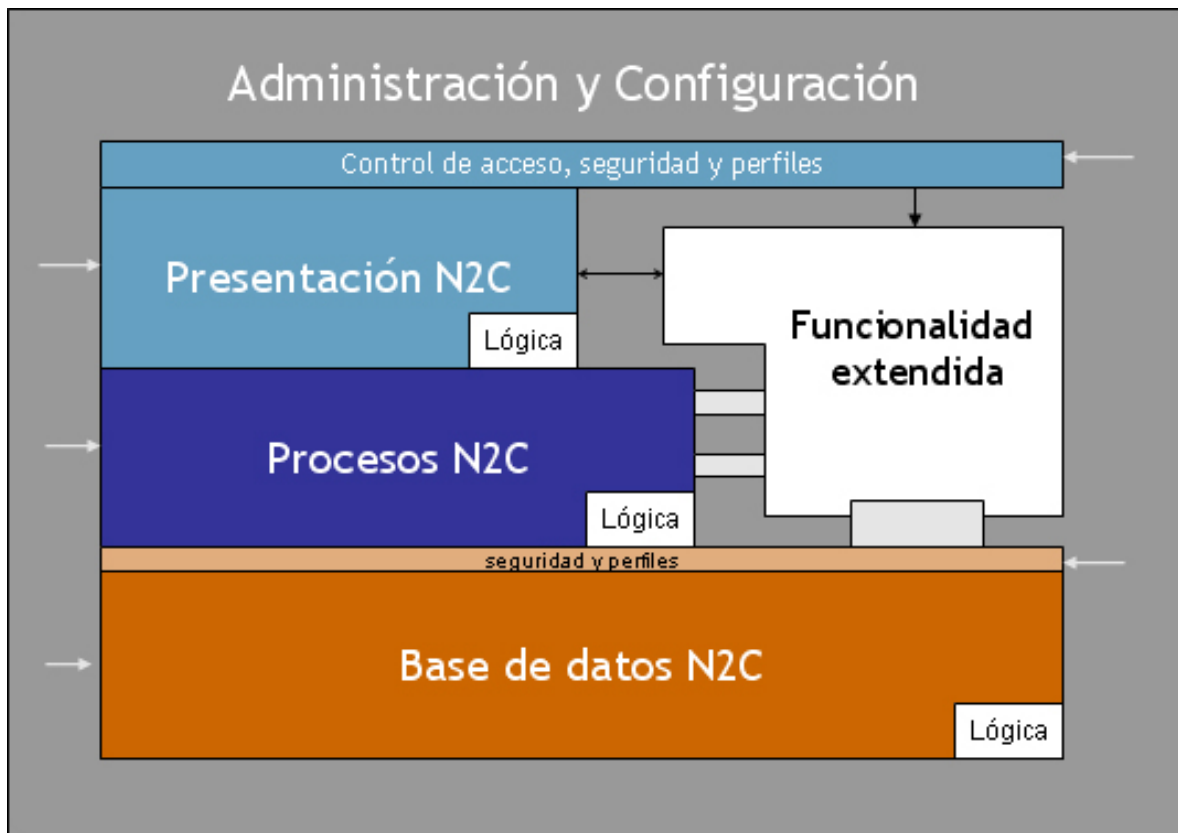


Figura 3.1 - Arquitectura básica de una aplicación generada por Net2Client

Adicionalmente, las piezas de color blanco nos muestran los componentes que pueden o no desarrollarse para una aplicación y que se integran con la estructura

generada de tres posibles maneras: insertándose como módulos de lógica de negocio en cualquiera de las capas, comunicándose con los procesos de *Net2Client* a través de las interfaces o accediendo directamente a los datos mediante librerías y mediadores que garantizan el cumplimiento de los niveles de seguridad y perfiles de acceso definidos para la aplicación en su configuración.

Cada aplicación generada cuenta con dos espacios asociados, a saber:

- El espacio de la herramienta, común a todas las aplicaciones, representa al conjunto de programas, rutinas y utilitarios pertenecientes a *Net2Client*, que están involucrados en el despliegue de los sistemas generados. En la figura 3.1 está representado por los recuadros azules que marcan las capas superiores.

- El espacio de la aplicación, que es particular a cada una de ellas, hace referencia a todo código que haya sido desarrollado de manera particular e independiente y que entra en funcionamiento dentro del sistema generado. Está representado por los recuadros de color blanco en la figura 3.1.

También existe la posibilidad de que varios sistemas compartan su espacio de aplicación, cuando constituyen distintas instalaciones de un mismo producto. En este caso se habla del espacio del paquete.

3.1 - *Net2Client* como plataforma

Net2Client está pensado para la elaboración de sistemas aplicados a la gerencia de las interacciones con los clientes y su relación con los procesos internos de cada organización, por lo que sus modelos de datos se ajustan a la representación de estos

conceptos. Esto se debe a que está orientado al modelo de Red de Servicio al Cliente que fue descrito en el capítulo 2.

Su diseño busca facilitar y promover la integración entre las distintas aplicaciones generadas. De allí que su utilización ideal a nivel empresarial consista en la implementación de varios sistemas que cubran todas las necesidades de la organización y se integren con transparencia dentro de la plataforma global.

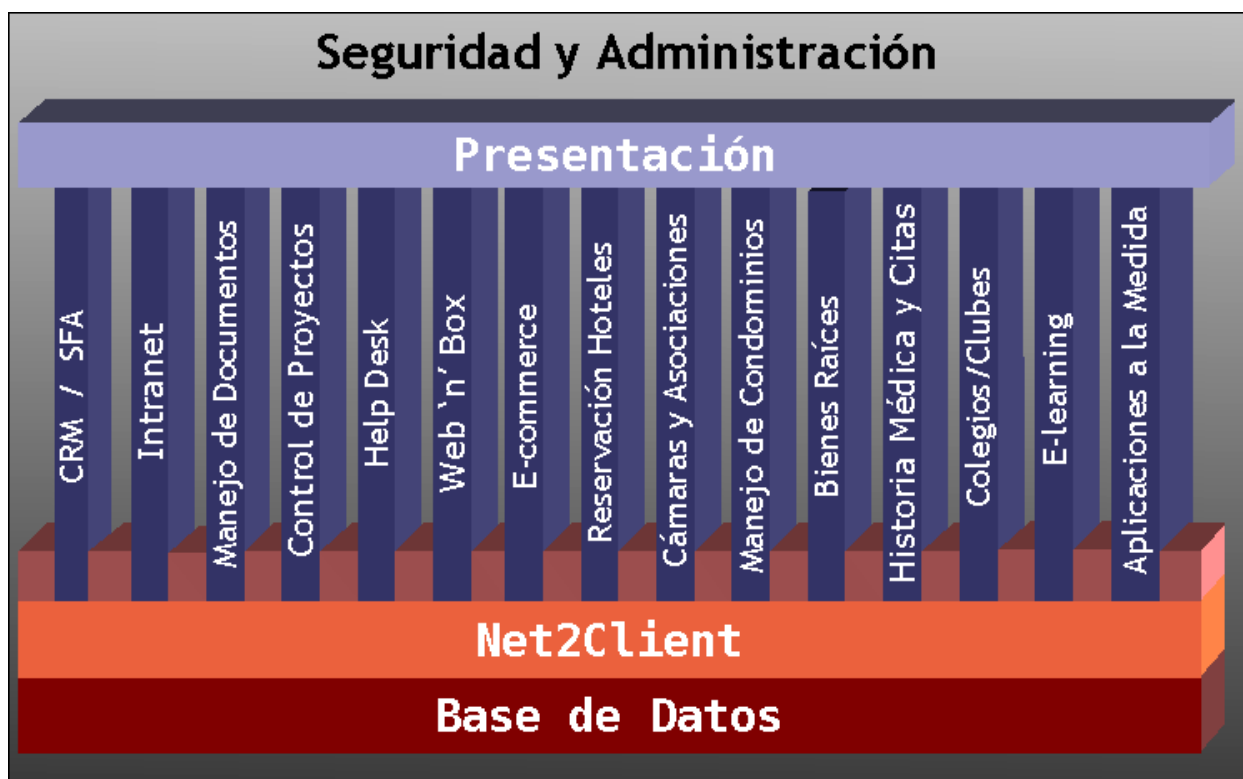


Figura 3.2 - Ejemplos de sistemas integrados a Net2Client

Adicionalmente, esta plataforma cuenta con una serie de aplicaciones “prefabricadas” (e igualmente configurables), que buscan cubrir necesidades genéricas y comunes que suelen presentarse en las organizaciones. Éstas representan un conjunto de productos desarrollados a través de *Net2Client* que se comportan

como paquetes que permiten la generación de aplicaciones ya predefinidas. Esto quiere decir que los sistemas que se declaren como instancias de un producto no partirán de cero en su configuración y estructuras, sino que se ajustarán a lo que fue definido para el paquete al cual fueron asociados.

En la figura 3.2 se puede apreciar cómo cada uno de estos paquetes (marcados en color azul oscuro) representa un producto que se integra a la plataforma completa agregando un nivel mayor de lógica de negocios. El objeto de este proyecto de pasantía es, precisamente, el desarrollo de uno de estos sistemas: la solución de *Net2Client* para la administración de un *Help Desk*.

CAPÍTULO 4.- PLANTEAMIENTO DEL PROBLEMA

En este capítulo se busca describir la motivación de la empresa detrás del planteamiento de este proyecto de pasantía, así como delimitar más claramente su objetivo y lo que se desea obtener como resultado de su desarrollo.

4.1 - Motivación

Anteriormente hemos descrito de manera general el funcionamiento y estructura de *Net2Client*, la plataforma de desarrollo de sistemas creada por GWT. Vimos, entonces, cómo una parte importante de ella es la serie de “paquetes” que son desarrollados utilizándola como herramienta y que integran el conjunto de aplicaciones que se ofrece a los clientes como producto independiente o como valor agregado a otros desarrollos. GWT está constantemente en la búsqueda de expandir las capacidades de *Net2Client*, lo que se traduce -entre otras cosas- en el fortalecimiento de este conjunto de aplicaciones, desarrollando cada vez nuevos productos de interés general que ataquen problemas comunes de una manera innovadora, aprovechando las facilidades que ofrece el estar integrado a una plataforma de generación de sistemas Web.

Analizando los problemas de sus clientes, las tendencias del mercado, e incluso su propio funcionamiento interno, GWT identificó la necesidad de un sistema de Información que permita el registro, control y seguimiento de todas las actividades involucradas en la administración de un *Help Desk* (o incluso en otros procesos de comportamiento similar en las organizaciones), y que esté orientado bajo el modelo de *Red de Servicios al Cliente*.

4.2 - ¿Por qué un *Help Desk*?

Ya hemos visto cómo la labor de administración de un *Help Desk* requiere de una gran cantidad de actividades que pueden optimizarse a través del uso de *software* especializado. La tendencia actual del mercado es producir cada vez más herramientas y sistemas sofisticados para la automatización de este tipo de tareas. Sin embargo, la existencia de esta variedad de opciones no significa que la necesidad esté cubierta por completo.

Debido a la naturaleza del problema y lo amplio del concepto de *Help Desk*, la gran cantidad de productos disponibles fallan al ofrecer soluciones igual de prácticas para organizaciones de distintos tipos. Es decir, las alternativas existentes tienden a estar íntimamente ligadas con una forma específica de llevar a cabo el proceso de administración del *Help Desk*, o a estar asociadas a un conjunto de conceptos que solamente son aplicables en una cierta rama de esta extensa área. Esto les impide ajustarse a realidades distintas a sus casos “ideales”. En muchos ocasiones incluso obligan a la organización a reformular sus procesos -a un gran costo- para ajustarse a las tecnologías adoptadas.

4.3 - Descripción del problema

Veamos ahora más en detalle cuáles son los problemas que se han detectado en la opción existente de *software* para el manejo de *Help Desk* y cuáles son las necesidades que se quieren satisfacer:

☒ La mayoría de los productos están dirigidos a la administración de Escritorios de Ayuda de Tecnología de la Información. Sus conceptos básicos y herramientas sólo

se ajustan a los *Help Desks* que presten soporte a equipos informáticos, por lo que no resultan prácticos para aquellos que no se enfoquen en esta área.

☒ El proceso de atención a las solicitudes varía considerablemente entre los distintos *Help Desks*; sin embargo, la mayoría de las aplicaciones imponen un mecanismo específico para su funcionamiento. Esto ocasiona que las organizaciones que las adoptan deban tratar de ajustar la utilización de la herramienta para que refleje su flujo de trabajo, aun cuando éste difiera de lo que el producto ha preestablecido. Un ejemplo está en el uso de niveles que en muchos paquetes de administración de *Help Desk* se restringe a un número fijo: típicamente dos.

☒ En el funcionamiento de un *Help Desk* pueden estar involucrados muchos conceptos distintos y a veces alternativos. Un ejemplo claro está en el uso de frentes o niveles y la subdivisión en grupos y colas: típicamente, las organizaciones que especifican su proceso a través de niveles, dividen a sus analistas en función de estos y no utilizan la noción de grupos de atención por tipos de solicitudes, y viceversa. Por ello la tendencia en el mercado de *software* para *Help Desks* es incluir sólo uno de estos conceptos, sin permitir la posibilidad de que una organización adopte el alternativo, o ambos.

☒ En el caso contrario están los productos que sí incluyen una gran cantidad de conceptos, pero no permiten que el proceso de atención de solicitudes sea ajustable para cada organización. Esto no beneficia a los *Help Desks* pequeños que se ven forzados a utilizar una herramienta demasiado sofisticada para sus tareas sencillas, sin

poder especificar cuáles de las opciones disponibles son las que verdaderamente necesitan para su funcionamiento básico.

☒ Esta misma situación se ve reflejada en las organizaciones que no son propiamente un *Help Desk*, pero que de igual forma manejan un proceso de atención y seguimiento a ciertos “entes” durante su ciclo de vida. Debido a que sus problemas son muy específicos, difícilmente ubicarán una herramienta que se ajuste a sus necesidades, y si utilizan *software* para *Help Desk*, deberán adaptarse al uso de conceptos que le son ajenos.

4.4 - Descripción de los objetivos

En respuesta a estos problemas, GWT propone el desarrollo de un sistema de control y seguimiento de solicitudes de un *Help Desk*, completamente configurable e integrado a su plataforma de generación de aplicaciones. Más específicamente, se busca que la herramienta a elaborar cuente con las siguientes características:

☒ Permitir el registro y manipulación de todas las solicitudes recibidas por un escritorio de ayuda, así como el seguimiento del proceso de atención a las mismas. Esto implica, además, el manejo apropiado de los datos de los clientes, recursos y contratistas asociados al *Help Desk*.

☒ Ofrecer mecanismos para que el proceso de atención de solicitudes pueda ajustarse, de manera rápida y sencilla, lo más posible al funcionamiento de cada organización que utilice el sistema. Esto significa que se deben incluir todas las nociones más comunes que entran en juego en el funcionamiento de un *Help Desk*,

pero a la vez permitir que se pueda especificar en cada caso cuáles se quieren incluir. De igual manera, todo lo que puede variar de una organización a otra (como por ejemplo, cantidad de niveles) debe ser completamente configurable.

☑ El sistema debe estar diseñado para prestar servicio a diversos tipos de *Help Desk* u organización con procesos similares. De ahí que los conceptos empleados y las maneras de hacer referencia a ellos deben ser configurables en cada instalación.

☑ Además, la herramienta debe integrarse completamente a la plataforma de *Net2Client*, por lo que su diseño debe estar orientado a la interconexión con otros paquetes y aplicaciones generadas por la herramienta.

☑ Finalmente, el sistema debe orientarse hacia el Modelo de Red de Servicio al Cliente. En el caso de un paquete para administración de un *Help Desk*, esto implica hacer énfasis en ofrecer una herramienta de “autoservicio” para los solicitantes, y de automatización de las labores diarias para los analistas.

CAPÍTULO 5.- METODOLOGÍA DE DESARROLLO

La metodología de desarrollo utilizada en esta pasantía se vio influenciada por el hecho de que su objeto fuera el desarrollo de un sistema a través de una plataforma generadora de aplicaciones. El uso de este tipo de herramientas impone cierto flujo de trabajo en las actividades de implementación. En el caso específico de *Net2Client*, su utilización fomenta el desarrollo de prototipos evolutivos, pues permite elaborar módulos entregables de *software* en plazos cortos e integrar cambios a los mismos con relativa facilidad.

Así, el proceso de desarrollo utilizado en este proyecto debía ajustarse a un modelo de vida de tipo incremental. En consecuencia, se utilizó una adaptación de la metodología RUP, la cual -entre otras cosas- cumple con esta característica. Veamos ahora con más detalle en qué consiste esta metodología y cómo se adapta al proceso definido para el presente desarrollo.

5.1 - RUP: *Rational Unified Process*

RUP es una metodología de desarrollo de *software*¹ caracterizada por proveer un enfoque estructurado a la asignación de roles y planificación de actividades dentro de una organización, para la ejecución de proyectos [RSC 2002]. Sin embargo, su clave está en que es en sí un proceso configurable, por lo que puede adaptarse con igual facilidad a proyectos más pequeños, como en el caso de esta pasantía, donde el equipo de desarrollo cuenta con un solo miembro, único responsable de todos los roles y actividades a lo largo del proyecto.

¹ En la actualidad, RUP conforma una plataforma de desarrollo ofrecida como un producto por la empresa IBM. Sin embargo, en este texto haremos referencia sólo al proceso de *software* que dicha plataforma permite implementar.

Los creadores de RUP lo definen como un proceso de desarrollo de *software* orientado por casos de uso, centrado en la arquitectura, y de carácter iterativo e incremental [JRB 2000]. Este último concepto hace referencia a la “subdivisión” del proyecto en varias partes o iteraciones “independientes”, que se van culminando consecutivamente, resultando cada una en un incremento. Así, el desarrollo se va realizando cíclicamente y el producto va incrementándose con cada iteración, lo cual se ajusta perfectamente con el modelo de vida que debía seguir este proyecto.

En la figura 5.1 podemos observar la arquitectura general de RUP. En ella se distinguen dos dimensiones:

- El eje vertical es la dimensión estática del proceso, pues representa las **disciplinas** -entendidas éstas como agrupaciones lógicas de actividades asociadas- en las que se incurre durante el proyecto [RSC 2002]. En esta dimensión se agrupan los conceptos de RUP más asociados al flujo de trabajo dentro del equipo de desarrollo (roles, *workflows* y artefactos intermedios), por lo que no resulta de interés para este proyecto describirla en detalle.
- El eje horizontal es la dimensión dinámica del proceso, pues representa el progreso del proyecto durante el tiempo, desde su inicio hasta su culminación. Es aquí donde se refleja el carácter evolutivo e iterativo del proceso, elemento clave en la implementación que de la metodología se realizó durante esta pasantía. Más adelante describiremos en detalle las fases y demás elementos que componen esta dimensión.

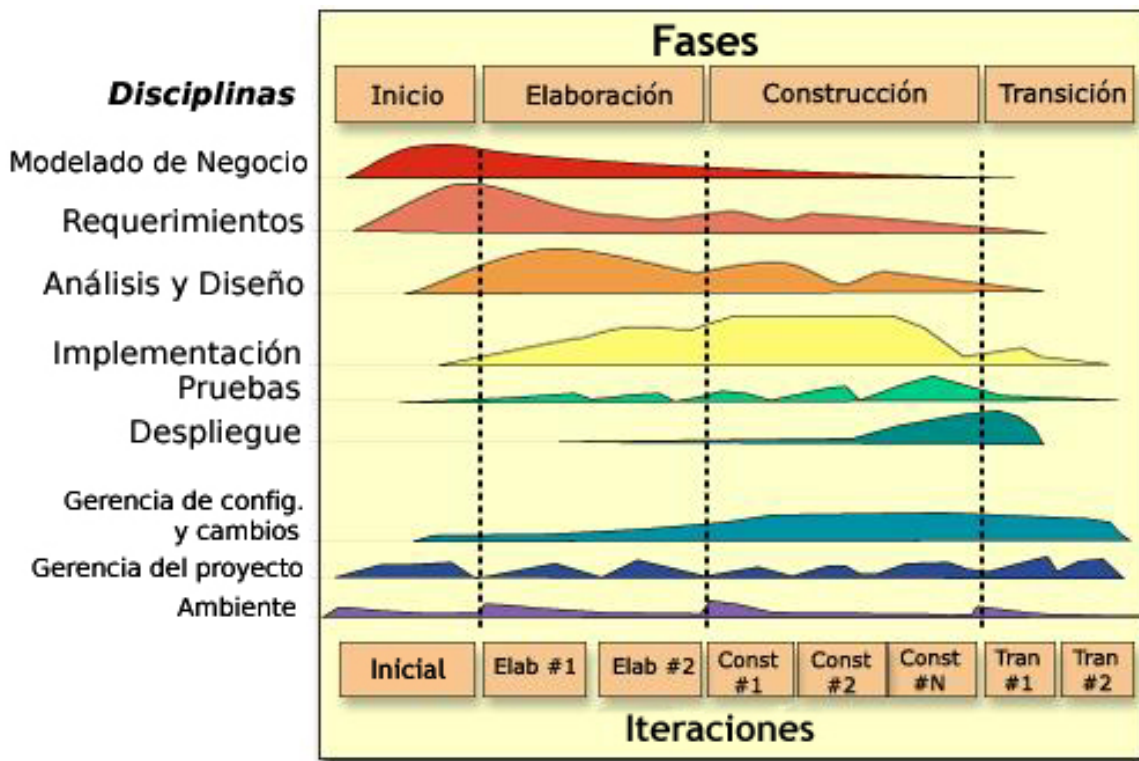


Figura 5.1 - Arquitectura general de RUP

El gráfico de la figura 5.1 nos muestra cómo el énfasis en las distintas disciplinas durante el proyecto va variando conforme se avanza en el tiempo. Podemos ver cómo, a diferencia de metodologías tradicionales, RUP establece un alto nivel de coexistencia entre distintas actividades que usualmente son tratadas como excluyentes. Más aun, es notorio cómo desde etapas iniciales se muestra un esfuerzo considerable en actividades de implementación e incluso en etapas avanzadas se estipula un énfasis en tareas de análisis, diseño y modelado de negocio. Esto resulta altamente compatible con el estilo de desarrollo incremental impuesto por el uso de herramientas de generación de sistemas.

5.2 - Ciclo de vida de RUP: Fases e Iteraciones

La metodología RUP divide el ciclo de vida del desarrollo de *software* en cuatro fases secuenciales. El final de cada una de ellas está marcado por el alcance de un *hito* que representa el cumplimiento de sus objetivos. Dentro de cada fase ocurren una o más *iteraciones*, entendidas éstas como un conjunto de actividades que se llevan a cabo siguiendo un plan y resultan en la producción de un *incremento* [RSC 2002]. Dichos incrementos representan versiones estables del producto completo y en esta implementación de RUP están conformados por los prototipos evolutivos que se van produciendo con la herramienta generadora. RUP establece que cada iteración se enfoque en un subconjunto de las funcionalidades (casos de uso) para incluirlas en el siguiente incremento. Adicionalmente, indica que la elección de estos subconjuntos se debe hacer basándose en un estudio del nivel de riesgo que cada uno representa en el desarrollo global. Así, los primeros casos de uso que son implementados -y por tanto, las primeras funcionalidades en aparecer en el producto- son los más cruciales, y las subsiguientes iteraciones van agregando aspectos menos críticos que representan un menor riesgo de cambios en la arquitectura o, incluso, fracaso del proyecto.

Veamos ahora en qué consiste cada una de estas cuatro fases, enfocándolas desde los conceptos de interés para esta pasantía:

Fase de Inicio:

El objetivo principal de esta fase es lograr un acuerdo entre todos los interesados en cuanto a los objetivos globales y el alcance del desarrollo. Es también en esta fase que se deben identificar los riesgos que enfrenta el proyecto y basándose en ellos determinar su viabilidad. Otras actividades que se llevan a cabo incluyen la

determinación de los casos de uso principales, la definición de una arquitectura candidata preliminar y la preparación del ambiente de desarrollo.

Hito: Definición de los objetivos del ciclo de vida.

Fase de Elaboración:

Los esfuerzos durante esta fase están orientados a la definición de una arquitectura estable que pueda soportar sin riesgo el grueso del desarrollo que se llevará a cabo en la siguiente fase. Para alcanzar esto, se debe probar y evaluar la arquitectura contra los escenarios que mayor riesgo representan. Esto se logra a través del desarrollo del primer prototipo evolutivo que cubre los casos de uso más críticos. Esta primera versión se convierte en el primer incremento y la línea base sobre la cual se irá construyendo el sistema.

Hito: Arquitectura estable y línea base para la construcción

Fase de Construcción:

El objetivo de esta fase es completar el desarrollo del sistema partiendo del prototipo evolutivo elaborado para definir la arquitectura. En ella se van aclarando los casos de uso restantes e incrementando de forma iterativa el producto a elaborar. Esta fase es considerada un proceso más de manufactura que de creación y está enfocada a la producción de múltiples versiones consecutivas del sistema. Esto se hace especialmente significativo en el presente proyecto, debido al uso de una herramienta como *Net2Client*, lo que se traduce en un mayor número de iteraciones de construcción y una menor duración para cada una de ellas.

Hito: Capacidad operacional inicial.

Fase de Transición:

La fase de transición se inicia cuando la versión del producto está lo suficientemente madura como para ser entregada a sus usuarios finales. En esta fase se agrupan todas las actividades para hacer esa transición. Esto incluye típicamente pruebas exhaustivas formales y ajustes al sistema basados en la respuesta del usuario.

Hito: Producto finalizado y lanzado al mercado.

Es importante destacar que esta última fase no está contemplada dentro del alcance de esta pasantía, puesto que las restricciones de tiempo sólo permiten que su objetivo prevea la elaboración de un prototipo funcional con un subconjunto importante de las características deseadas para el producto final. De cualquier manera es importante destacar que el trabajo con la plataforma *Net2Client* permitió que cada iteración de la fase de construcción incluyera al final algunas actividades propias de la fase de transición, pues todo incremento que se produce con la herramienta pasa automáticamente a ambiente de producción. Esto significa que cada uno de ellos puede ser utilizado y probado inmediatamente por usuarios finales, en este caso internamente por los empleados de GWT, proveyendo un valioso *feedback* para las subsiguientes iteraciones de construcción que podían así ir realizando de manera paralela los ajustes que tradicionalmente se hubieran efectuado al final del desarrollo, durante la transición.

5.3 - Conceptos clave de RUP en el desarrollo de este proyecto

Como ya hemos descrito anteriormente, la metodología RUP incluye una gran cantidad de conceptos que abarcan todos los posibles aspectos involucrados en la realización de un proyecto de desarrollo *software* de cualquier envergadura. Sin embargo, cada implementación o adaptación del proceso se debe basar únicamente en las nociones que más se ajusten a sus circunstancias. Recapitulemos, entonces, los conceptos de RUP que resultaron clave en el desarrollo de esta pasantía:

- El ciclo de vida está dividido en fases. Éstas, a su vez, en iteraciones.
- El avance del desarrollo se da a través del alcance de hitos.
- Cada iteración está orientada a la implementación de un subconjunto de las funcionalidades del producto final.
- Se utiliza un prototipo evolutivo que va creciendo con cada incremento.
- Se producen varias versiones consecutivas del sistema muy rápidamente.
- Los casos de uso críticos se enfrentan primero y con cada iteración se va disminuyendo el nivel de riesgo.

CAPÍTULO 6 - *NET2CLIENT HELP DESK*

En este capítulo se busca describir el proceso de desarrollo que se llevó a cabo durante esta pasantía y los resultados obtenidos. Para ello recorreremos el ciclo de vida del sistema desarrollado, revisando en cada una de las fases que establece la metodología las actividades realizadas y los objetivos alcanzados, y ofrecemos una descripción de su estado actual. Los resultados descritos son ilustrados con un conjunto de diagramas y modelos expresados en el lenguaje UML.

6.1 - Fase de Inicio

Esta fase se concluyó en una iteración que tuvo una duración de 4 semanas. La principal actividad durante este período fue el análisis detallado del problema de la administración de un *Help Desk* y los aspectos asociados al uso de *software* en apoyo a esta tarea. Esto incluyó una investigación sobre el tema y entrevistas a profesionales con experiencia en el área. Igualmente se realizó un estudio de la oferta de *software* para *Help Desks*, probando una gran cantidad de productos de distintos enfoques e identificando los puntos más comunes y las diferencias entre ellos.

Adicionalmente, durante esta primera fase se llevó a cabo el curso inductivo a la plataforma *Net2Client* y posteriormente se estudió detalladamente la herramienta y se realizaron pequeñas pruebas -generando aplicaciones menores- para poder comprender su funcionamiento y proponer una estructura para el nuevo sistema.

El producto de estas actividades permitió, finalmente, definir los objetivos concretos del desarrollo que se iba a comenzar y tener una visión clara del producto a elaborar, su posicionamiento y su alcance. Además, se pudieron identificar los requerimientos y funcionalidades -a un nivel macro- del sistema.

Resultados:

a) Se delineó la **visión del producto** a elaborar durante la pasantía, el cual se convertirá en un paquete integrado al conjunto de aplicaciones de la plataforma *Net2Client*, y estará identificado por el nombre “*Net2Client Help Desk*” (N2CHD). Buscará posicionarse como una herramienta configurable de apoyo para la gestión de escritorios de ayuda de cualquier tipo, o de organizaciones que tengan una misión similar. Sin embargo, estará principalmente orientado al sector de los *Help Desks* pequeños y medianos, que son los más comunes y a la vez menos atendidos por la oferta actual de *software*.

Net2Client Help Desk se deberá caracterizar por ser completamente genérico y permitir que cada aspecto del proceso que soporta sea fácilmente configurable y ajustable para adaptarse con la mayor precisión posible al funcionamiento de la organización que lo utilice. También deberá afincarse en facilitar la interacción entre el *Help Desk* y sus clientes, promoviendo el autoservicio y apoyando al analista con herramientas para mantenerse en contacto con su solicitante.

b) Se identificaron los **requerimientos** básicos del sistema a desarrollar, los cuales fueron tipificados y calificados según un formato definido para el proyecto (dicho formato puede consultarse en el apéndice A: “Manejo de requerimientos”). Estos requerimientos permitieron delimitar el alcance del sistema y empezar a definir más exactamente las funcionalidades que deberá ofrecer y la arquitectura que habrá de soportarlas. Es importante destacar que debido a que el desarrollo completo del sistema continuará después de concluida esta pasantía, no todos los requerimientos

identificados formaron parte del alcance de la misma, pues correspondían a funcionalidades que se agregarán en futuros incrementos o versiones del producto.

En la tabla 6.1 se pueden ver algunos de los requerimientos más importantes que se identificaron.

Código	Descripción
FUNC-01	Permitir la recepción de nuevas solicitudes y el registro de sus características (así como documentación asociada)
FUNC-02	Facilitar la clasificación de solicitudes según tipos y prioridades configurables.
FUNC-03	Controlar el estado (y progreso) de cada solicitud registrada durante cada etapa de su ciclo de vida
FUNC-04	Permitir el registro de eventos varios asociados con cada solicitud por parte de los analistas responsables
FUNC-06	Permitir a los clientes consultar el estado de sus solicitudes y los eventos asociados a la misma
SEG-02	Restringir al cliente la información pertinente al manejo interno de las solicitudes, según niveles de privacidad configurables
FUNC-10	Permitir la configuración de niveles de respuesta (SLA's) asociados con tipos y características de solicitudes, con usuarios, con grupos de analistas o con contratistas.
FUNC-14	Mantener la información de los analistas y grupos que atienden las solicitudes
FUNC-22	Llevar un control de la asignación de solicitudes a cada analista, grupo o contratista y los eventos que ocurran en el transcurso de la misma.
FUNC-27	Llevar un control de las solicitudes asociadas a cada recurso
NOF-03	Ofrecer mecanismos de búsqueda rápida de solicitantes como apoyo al personal que registra solicitudes en vivo
NOF-04	Permitir la configuración de los nombres de los conceptos principales
TEC-03	Permitir que los datos de solicitantes, analistas y recursos puedan ser compartidos con otras aplicaciones de la plataforma <i>Net2Client</i>

Tabla 6.1 - Principales requerimientos de *Net2Client Help Desk*

c) Se identificó un conjunto de **riesgos** que podían amenazar tanto el desenvolvimiento del proyecto como la calidad del producto elaborado. Para cada uno de ellos se determinó el grado en que podía afectar negativamente el alcance de los objetivos y se estimó la probabilidad de que ocurriera.

Nombre Descripción	
#1	Utilización de herramientas desconocidas para el desarrollador Para el desenvolvimiento de este proyecto es crucial que el pasante conozca y aprenda a utilizar apropiadamente la herramienta de desarrollo <i>Net2Client</i> en un tiempo prudencial. Esto para evitar que la curva de aprendizaje retrase el proyecto o que el desconocimiento de la herramienta se traduzca en sub-utilización de la misma y desaprovechamiento de sus posibilidades, lo que eventualmente puede implicar una baja calidad en el producto elaborado.
#2	Posibles restricciones técnicas para el desarrollo impuestas por la herramienta de generación El uso de este tipo de <i>software</i> en la elaboración de un sistema pudiera implicar que el desarrollador deba ajustarlo a algún modelo específico o no tenga libertad de implementar las soluciones de la manera más eficiente.
#3	Presión sobre el tiempo de desarrollo El alcance del presente proyecto y la amplitud del problema a resolver son bastante grandes y el desarrollo del mismo está delimitado a un período estrecho de tiempo. Esto ocasiona que la planificación sea muy ajustada y no tenga flexibilidad para tolerar retrasos.
#4	Énfasis en la adaptabilidad del sistema desarrollado puede redundar en pérdida de simplicidad Es posible que para permitir que el proceso de atención de solicitudes modelado sea completamente configurable, sea necesario implementar mecanismos de adaptación que resulten excesivamente complicados, deteriorando así la flexibilidad de la herramienta.

Tabla 6.2 - Principales riesgos identificados

En la tabla 6.2 se listan los principales riesgos identificados junto con una breve descripción. Son de especial importancia los señalados con los números 2 y 3, pues en las fases subsiguientes del desarrollo se presentaron con una magnitud moderada trayendo algunas consecuencias para la ejecución del proyecto.

d) Se especificaron -a un nivel macro- las **funcionalidades básicas** que debía ofrecer el sistema a desarrollar. El resultado obtenido abarca dos aspectos, a saber:

- En primer lugar se determinaron los **actores** que iban a interactuar con el sistema, quienes representan a los entes que activarán sus funcionalidades. Para el caso específico de *Net2Client Help Desk*, la especificación de los actores sirve como guía para identificar los distintos tipos de usuarios finales de la herramienta. En la figura 6.1 se pueden apreciar estos actores y las relaciones entre ellos.

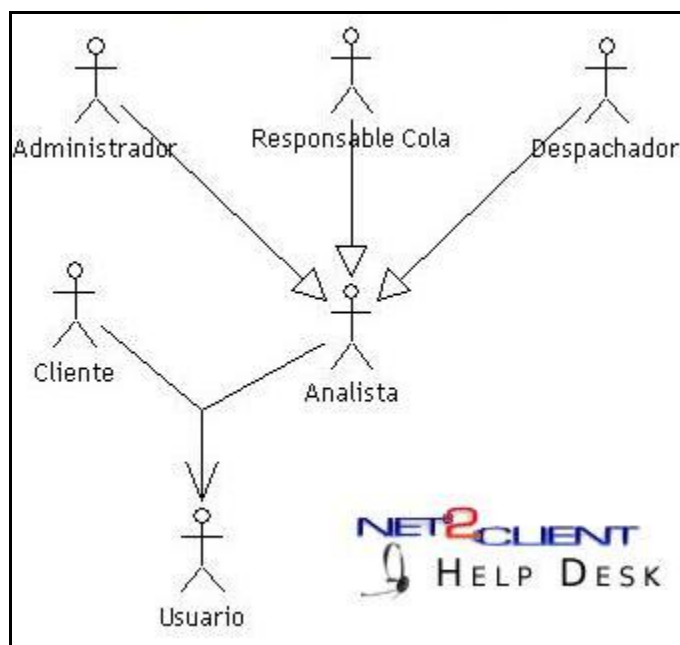


Figura 6.1 - Diagrama de actores

Veamos ahora más en detalle el tipo de usuarios representados por cada uno de estos, pasando de los más generales a los más específicos:

- El actor Usuario representa, como su nombre lo indica, a todos los usuarios del sistema sin distinción alguna. Su vista de la aplicación será la más genérica e incluirá todas las funcionalidades que son comunes para todos. Todo *Usuario* necesariamente deberá ser *Analista* o *Cliente*.

- El actor Cliente representa a los entes que emiten las solicitudes de servicio, los usuarios externos del sistema. Cada *Cliente* será un solicitante para *Net2Client Help Desk* e interactuará con él para originar nuevas solicitudes y consultar sus datos.

- El actor Analista representa a los usuarios internos del sistema; es decir, el personal del escritorio de ayuda. Un *Analista* es el principal interlocutor de la herramienta y su visión del sistema incluye todas las funcionalidades básicas a las que pueden acceder la mayoría de los usuarios. Un subconjunto de los analistas puede tomar más acciones en el sistema y está representado por los actores *Despachador*, *Administrador* y *Responsable Cola*.

- El actor Despachador representa a los analistas del *Help Desk* que se encargan de la recepción, enrutamiento y despache de las solicitudes. Esto significa que su visión del sistema debe incluir un mayor conjunto de funcionalidades para permitir estas tareas que no pueden acceder otros analistas.

- El actor Responsable Cola representa a aquellos analistas que tienen a su cargo alguno de los grupos o colas que funcionen en el *Help Desk*. Por ello deben poder acceder a información protegida que no estaría disponible para el común de los usuarios internos.

- El actor Administrador representa al usuario más avanzado del sistema. Este grupo incluye a aquellos analistas que deben encargarse de las labores administrativas de la herramienta; es decir, configuración del proceso de atención de solicitudes, actualización de datos básicos, autorización de usuarios, entre otras. Su visión de la herramienta es la más amplia pues incluye un gran conjunto de funcionalidades adicionales a las ya accesibles por los *Analistas*.

- En segundo lugar se identificaron las funcionalidades generales que definen el sistema a desarrollar y su alcance. Estas están expresadas en un conjunto de **casos de uso preliminares** que a lo largo del desarrollo serán revisados y especificados hasta convertirse en documentaciones precisas de cada aspecto del funcionamiento de *Net2Client Help Desk*.

En la figura 6.2 se presenta el Diagrama de Casos de Uso Preliminares mostrándonos las relaciones existentes entre ellos y con los actores ya descritos. Como ya hemos explicado anteriormente, no todas las funcionalidades identificadas para el desarrollo completo entran dentro del alcance de esta pasantía. La figura 6.2 incluye sólo a aquellas que corresponden a la porción del desarrollo prevista para este período de tiempo. Veamos ahora un poco más en detalle lo que representan:

- Los casos de uso Administrar... hacen referencia, como es de esperarse, a todas las funcionalidades asociadas a las labores administrativas y de configuración de la herramienta, por lo cual vemos que son activados únicamente por el actor “Administrador”. Si bien son los más abundantes del diagrama, estos casos de uso no representan la misión crítica del sistema y son típicamente los que se presentarán con menor frecuencia en el funcionamiento cotidiano de la herramienta.

Cada uno de ellos está asociado con algún concepto básico del problema de atención de un *Help Desk* y básicamente agrupa las funcionalidades de manejo de datos; es decir, agregar nuevos entes de ese concepto, buscar y consultar los registrados, modificarlos e incluso eliminarlos. Para algunos casos específicos es posible que hayan funcionalidades (típicamente las consultas) que sean accesibles a otros actores además del “Administrador”, pero esto se verá con mayor detalle una vez que estos casos de uso globales sean refinados y especificados en la próxima fase.

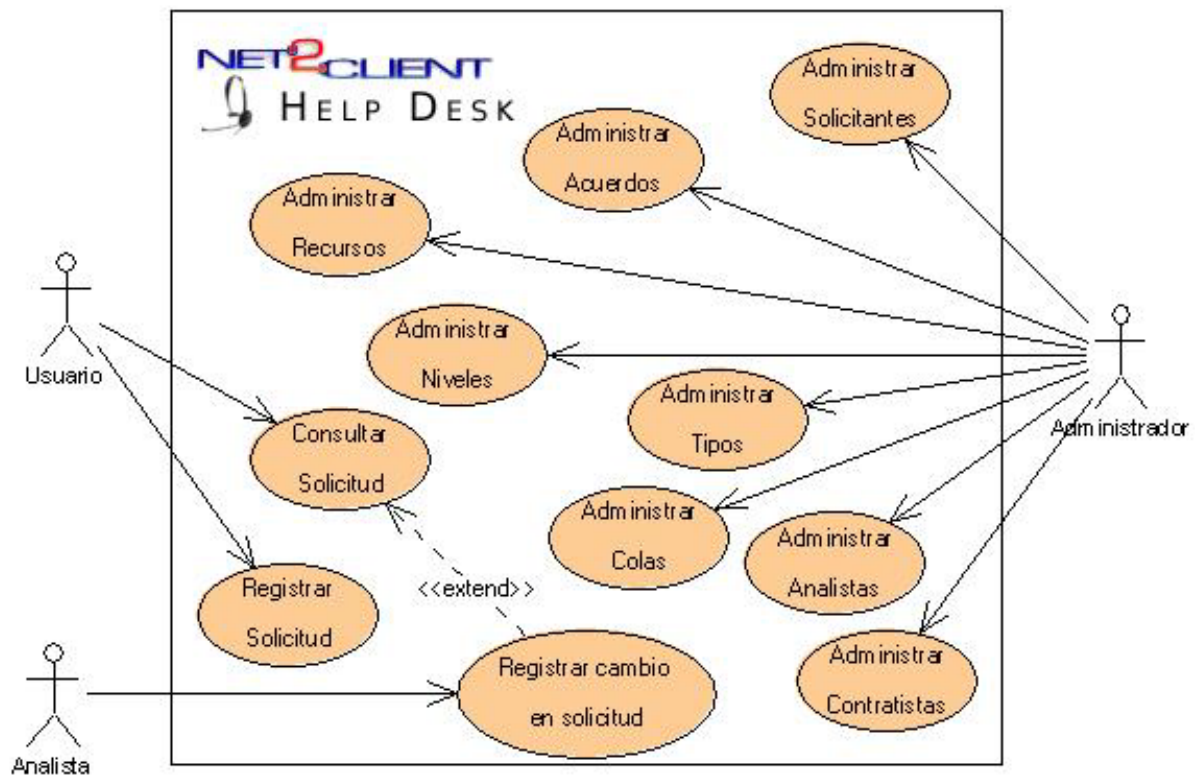


Figura 6.2 - Diagrama de Casos de Uso Preliminares

- El caso de uso Registrar Solicitud es el más público -siendo accedido directamente por el actor “Usuario”- y es de gran importancia porque representa el

inicio de todas las actividades del *Help Desk*: la recepción y/o apertura de las solicitudes de soporte.

- El objetivo principal del sistema está contenido en los casos de uso Consultar Solicitud y Registrar cambio en Solicitud, que agrupan las funcionalidades más importantes y que probablemente se presenten con mayor frecuencia, en la herramienta. Están relacionados a través de una “extensión” que indica que el registro de cambios de solicitud es una acción que usualmente ocurrirá como complemento a una consulta de sus datos.

Consultar Solicitud implica acceder a toda la información asociada a una petición de servicio y los avances durante su ciclo de vida, por ello puede ser accedida tanto por analistas como por solicitantes, con vistas menos o más reducidas de los datos según sea el caso. De ahí que el caso de uso sea activado directamente por el actor “Usuario”. Sin embargo, el caso de uso Registrar cambio en Solicitud ya implica la toma de acciones para la atención de la solicitud, actividad que sólo debe estar disponible para el actor “Analista”.

e) El estudio realizado acerca de los escritorios de ayuda y su funcionamiento permitió contrastar distintas formas de estructuración de estas organizaciones para alcanzar así una **definición genérica del proceso** de atención de solicitudes y los conceptos involucrados en las operaciones diarias de un *Help Desk*. Esto implica la identificación de las distintas tendencias aplicadas en su administración -uso de niveles, división por colas, entre otras- y la determinación de cómo estas se

interrelacionan en la misión crítica de un escritorio de ayuda cualquiera: la atención de las solicitudes.

El resultado de esta definición puede expresarse a través de un diagrama de estados de una solicitud que busca representar un proceso de atención a un nivel muy general; es decir, abarcando la mayor cantidad de conceptos posibles, buscando garantizar que el funcionamiento de una gran cantidad de *Help Desks* pueda definirse como un subconjunto de los estados y transiciones presentes en dicho diagrama.

Para facilitar su lectura y comprensión, el diagrama se presenta en dos partes. La primera, que puede verse en la figura 6.3, describe el proceso de atención de una solicitud para los casos en que el resultado es exitoso, es decir el curso principal de eventos. Igualmente por razones visuales se omitieron de este diagrama las transiciones referentes al pase de un nivel a otro que fueron modeladas como reflexivas (es decir, que no implican cambio de estado) pudiendo presentarse en los tres estados resaltados con un tono de azul más oscuro. Es importante destacar que un cambio de nivel puede ocurrir un número variable de veces dependiendo del mecanismo interno definido por cada *Help Desk*, e incluso puede presentarse como consecuencia automática de ciertos eventos reflejados en este diagrama, como por ejemplo la asignación a un grupo o a un ente externo. Sin embargo, en términos de las acciones que pueden tomarse o de los eventos que puedan ocurrir, el nivel en que se encuentre una solicitud no representa un estado particular de ella sino más bien una propiedad que la acompaña en cualquier estado en el que se encuentre.

Veamos ahora para cada uno de los estados definidos en el primer diagrama, qué representa y cuál es el mecanismo por el cual una solicitud entra o sale de él:

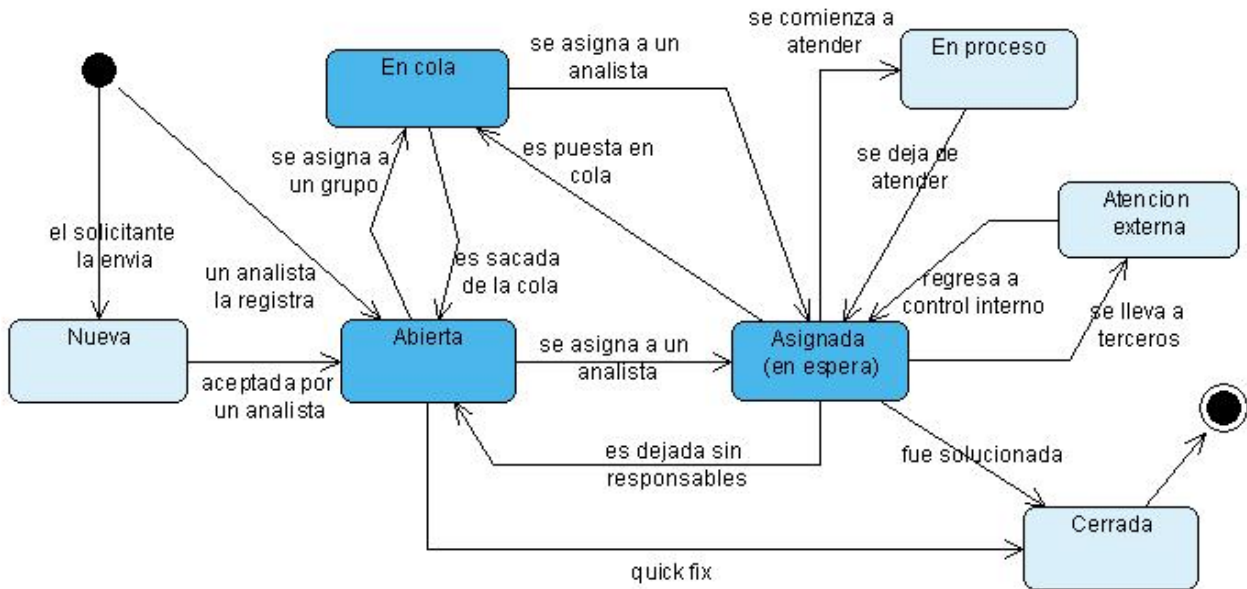


Figura 6.3 - Diagrama de Estados general de una Solicitud

- Una solicitud se considera Nueva cuando fue registrada a través de mecanismos externos al *Help Desk* y aún no ha sido aceptada por la organización; es decir, ningún analista ha entrado en contacto con ella.

- Cuando una solicitud es registrada por mecanismos internos (como por ejemplo aquellas que son reportadas al personal de atención telefónica) pasa automáticamente a encontrarse Abierta. Este es el estado por defecto de una solicitud que si bien se considera aceptada por el *Help Desk*, no se encuentra a cargo de ningún analista, contratista o cola. Una vez que le es asignado al menos un responsable, la solicitud sale de este estado, al cual puede regresar si en el futuro queda nuevamente sin ningún ente encargado de ella.

- Una solicitud se encuentra En Cola cuando ha sido asignada a un grupo y está a la espera de que algún analista se haga responsable por ella.

- Una vez que la solicitud se asigna a algún analista pasa a encontrarse Asignada (en espera). Este estado representa que si bien no se está trabajando por solucionarla en el momento, la solicitud cuenta con un responsable que responde por su evolución. Las solicitudes Asignadas están listas para ser atendidas en cualquier momento y regresar a este estado cualquier cantidad de veces hasta que sean solucionadas.

- Una solicitud está En Proceso cuando se está atendiendo propiamente; es decir, se están realizando acciones específicas por su resolución.

- Cuando se asigna la responsabilidad de trabajar en la solicitud a un ente externo al *Help Desk* esta pasa a estar en Atención Externa, donde permanece hasta que el control de su evolución regrese a la organización.

- Finalmente, cuando la solicitud es resuelta exitosamente puede considerarse como Cerrada, estado a partir del cual culmina su ciclo de vida.

La segunda parte del diagrama de estados, que puede verse en la figura 6.4, representa los cursos de eventos alternos que pueden resultar en un ciclo de vida no exitoso. Los estados marcados en color gris son aquellos, ya presentes en el diagrama de la figura 6.3, en los cuales pueden ocurrir los nuevos eventos que alterarían el ciclo de vida de la solicitud para llevarla a un flujo alternativo. La diferencia está básicamente en el paso por alguno de los dos estados marcados en azul, a saber:

- El estado Suspendida indica que la solicitud aún forma parte de las tareas pendientes del *Help Desk* pero que no estará siendo considerada para atenderse o asignarse durante un período de tiempo. Se puede llegar a este estado independientemente de si la solicitud en cuestión tenía un responsable, estaba en cola para algún grupo, o simplemente se encontraba abierta a la espera. Sin embargo, una

vez que se decide retomarla, regresa únicamente al estado de Abierta para comenzar su ciclo de vida desde cero.



Figura 6.4 - Diagrama de Estados de una Solicitud para casos no exitosos

- Finalmente una solicitud Cancelada es aquella para la cual no se consiguió una resolución exitosa y aún así dejará de formar parte de las tareas del *Help Desk* que dejará de atenderla definitivamente. A partir de este estado la solicitud culmina su ciclo de vida automáticamente, y puede llegar a él proveniente desde prácticamente cualquier punto en su evolución (menos aquellos estados dinámicos que implican atención efectiva: En Proceso y Atención Externa).

f) Se elaboró un **plan de desarrollo de software**, organizando las actividades del proyecto en función de las fases que especifica la metodología. Esto implicó la determinación de las iteraciones a realizar, sus objetivos, duraciones y fechas

estimadas de inicio y culminación. En la tabla 6.3 se presentan las actividades de este plan para el período de duración de esta pasantía:

Período	Semana Inicio	Semana Final	Duración (semanas)
Fase de Inicio	1	4	4
Iteración preliminar <i>Investigación del problema, identificación de requerimientos y delineamiento del alcance.</i>	1	4	4
Fase de Elaboración	5	9	5
Iteración E1 - Diseño Conceptual <i>Modelado conceptual y de datos, establecimiento de la arquitectura.</i>	5	6	2
Iteración E2 - Refinamiento y desarrollo de prototipo <i>Refinamiento de modelos, implementación de casos de uso críticos en el prototipo evolutivo.</i>	7	9	3
Fase de Construcción	10	19	10
Iteración C1 - Primer entregable <i>Desarrollo del primer entregable → manejo del ciclo de vida de las solicitudes.</i>	10	11	2
Iteración C2 - Expansión (niveles) <i>Desarrollo del segundo entregable → utilización de niveles y evaluación de solicitudes “vencidas”</i>	12	14	3
Iteración C3 - Expansión (recursos) <i>Desarrollo del tercer entregable → asociación de solicitudes con equipos o recursos.</i>	15	16	2
Iteración C4 - Expansión (enrutamiento) <i>Desarrollo del cuarto entregable → facilidades para asignación y enrutamiento de solicitudes</i>	17	19	3

Tabla 6.3 - Plan de desarrollo de software

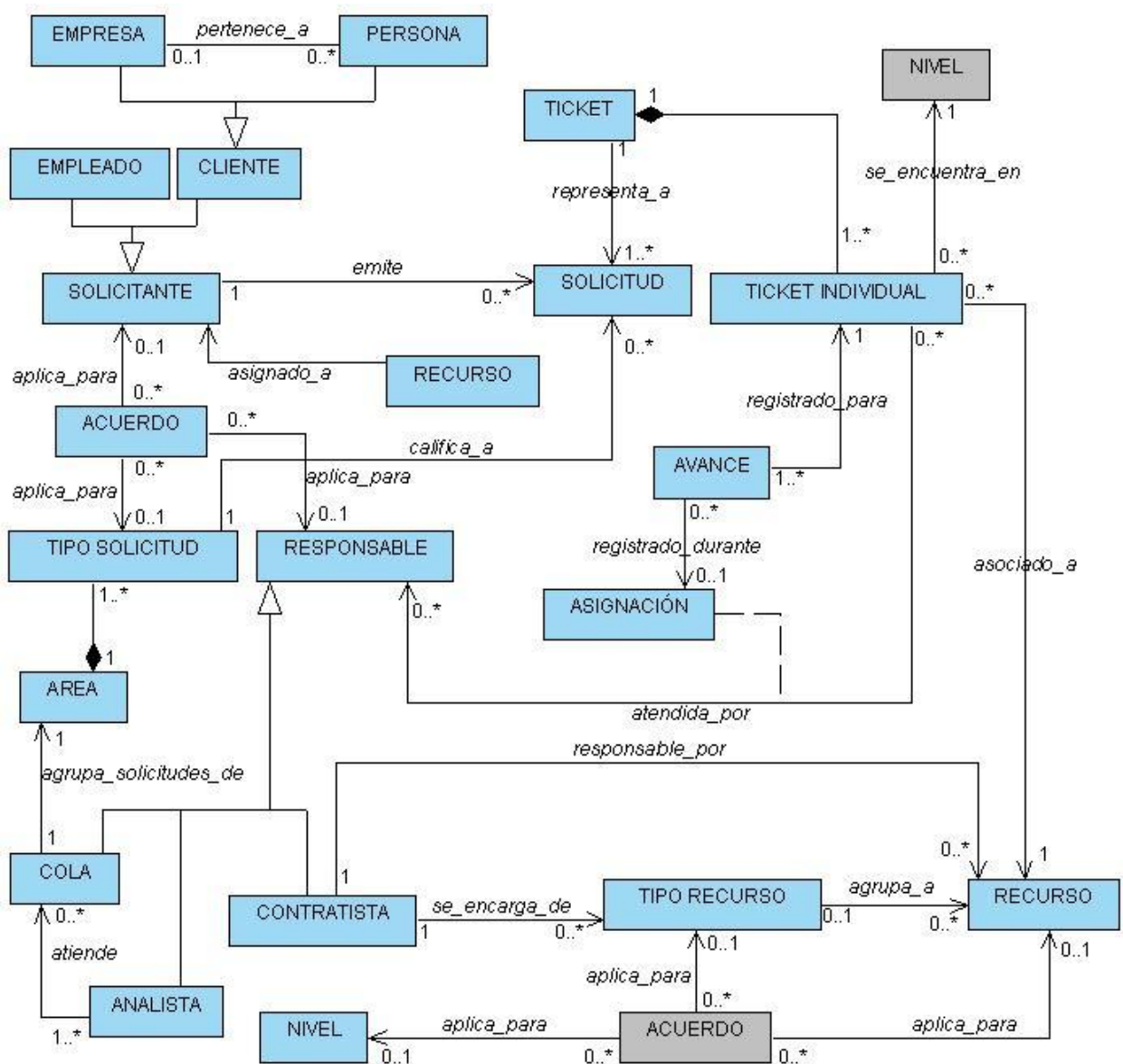
6.2 - Fase de Elaboración

Esta fase tuvo una duración de 7 semanas y abarcó dos iteraciones. En ella se presentaron ciertos factores que ocasionaron su extensión con respecto a lo planificado en la fase anterior. Esto será descrito más adelante cuando se hable sobre la incidencia de los riesgos y se presente el plan de desarrollo revisado para la siguiente fase.

En la primera iteración se aplicó la información derivada de los resultados de la fase anterior para iniciar el diseño del sistema, tanto en los aspectos de modelado del problema, como en los referentes a la estructuración de la solución. Todo esto enfocado en la definición de la arquitectura que habría de soportar el desarrollo, por lo que también entraban en consideración los factores asociados con los mecanismos a emplear durante la implementación.

En la siguiente iteración se iniciaron las actividades para la elaboración del primer prototipo evolutivo, que representaba la implementación de los casos de uso críticos -y por ende, las funcionalidades más importantes- del sistema. Paralelamente se realizaron las actividades de acondicionamiento del entorno de desarrollo para poder soportar la arquitectura diseñada en la iteración anterior. Esto incluyó la realización de modificaciones varias a la herramienta/plataforma de desarrollo, así como la investigación y prueba de mecanismos de almacenamiento alternativos. Este último aspecto será descrito más adelante cuando se presente la arquitectura de implementación de *Net2Client Help Desk*.

El producto de todas estas actividades fue la elaboración del prototipo evolutivo sobre el cual se basaron los futuros incrementos al sistema.



En la figura 6.5 puede consultarse este modelo, excluyendo del mismo los pocos conceptos asociados a aspectos del manejo de un *Help Desk* que no entran dentro del alcance de esta pasantía. A continuación se dará una breve descripción de estos conceptos identificados y lo que representan las asociaciones entre ellos:

- El concepto más importante es el de Solicitud. Cada una de ellas equivale a una petición que debe ser atendida, para lo cual es representada por exactamente un Ticket (que a su vez puede corresponder a varias solicitudes en los casos que éstas se agrupen por contar con una solución común). Estos *tickets* están compuestos por un Ticket Individual (o más en los casos en que la atención de un problema implique la solución de varios asociados) que no es más que el verdadero elemento de trabajo del *Help Desk*, cuyo funcionamiento está orientado a la resolución exitosa del mismo. Cada *ticket* individual se encuentra en algún Nivel específico en todo momento de su ciclo de vida; proceso durante el cual se le registran además varios Avances asociados, que van a representar sus cambios de estado, escalamientos, asignaciones y hasta resolución final.

- Los avances de los *tickets* pueden estar también asociados con alguna de sus Asignaciones. Estas representan el período de tiempo y conjunto de eventos que transcurren mientras el *ticket* en cuestión se encuentra asignado a algún Responsable. Estos pueden estar representados por los Analistas que ejecutan las labores del *Help Desk*, los Contratistas o entes externos que le presten servicio, e incluso las Colas de atención que se hayan definido. Los *tickets* asignados a una cola se encuentran a la espera de ser pasados a un analista que atienda dicha cola (de entre las varias que puede atender). Esto se debe a que cada una de ellas corresponde a una determinada

Área, entendida esta como un conjunto de categorías o Tipos de solicitudes según los cuales pueden calificarse estas peticiones recibidas.

- Los Solicitantes (aquellos que originan las solicitudes) pueden ser tanto internos -es decir, Empleados de la misma organización en la que opera el *Help Desk*- o externos, representados por los Clientes. Estos a su vez pueden ser Personas naturales o Empresas.

- Cada *ticket* individual puede representar la atención de un problema presentado por un Recurso o equipo para el cual el *Help Desk* presta servicio. Estos pueden estar asignados a algún solicitante y se clasifican en Tipos de Recursos. Los contratistas pueden estar encargados del soporte de algunos recursos específicos o de todos los que sean de un mismo tipo.

- Finalmente, cada Acuerdo representa un conjunto de condiciones mínimas exigidas para la atención de una solicitud bajo un conjunto específico de circunstancias de la misma, que pueden incluir el encontrarse en algún nivel, ser de un determinado tipo, haber sido originada por un solicitante en particular, estar asignado a algún responsable específico o corresponder a la atención de un recurso determinado o de un tipo particular.

b) Se alcanzó una **especificación detallada de las funcionalidades** del sistema, partiendo del conjunto de Casos de Uso preliminares identificados durante la fase de Inicio y refinándolos progresivamente hasta delimitar cada escenario de utilización que se implementará para *Net2Client Help Desk*.

Este resultado puede representarse a través de un diagrama de casos de uso que incluye estas funcionalidades, sus interrelaciones y los mecanismos de acceso a ellas por parte de los distintos actores asociados.

A continuación se presentará el subconjunto de estos casos de uso que hace referencia a funcionalidades que forman parte del alcance de esta pasantía, separándolos en dos diagramas para facilitar su lectura y explicando brevemente las funcionalidades más importantes y los mecanismos presentes en sus asociaciones.

En la figura 6.6 podemos observar el Modelo de Casos de Uso principales, es decir, aquellos que forman parte de la misión crítica del sistema. En este modelo se aprecian los casos de uso asociados con el registro, control, evolución y seguimiento de las solicitudes durante su ciclo de vida. Igualmente se encuentran las funcionalidades correspondientes a las labores administrativas de los conceptos más estrechamente asociados con una versión básica del proceso de atención de peticiones: los tipos y las colas.

De este modelo se desprenden los casos de uso identificados como **críticos** para ser implementados en el primer prototipo evolutivo, los cuales serán descritos brevemente a continuación:

- Registrar Solicitud: Este caso de uso representa la funcionalidad a través de la cual las solicitudes que serán atendidas por el *Help Desk*, entran al sistema. No se ahonda en su descripción puesto que no fue refinado desde su inclusión en el modelo de Casos de Uso preliminares que se realizó en la fase de Inicio, por lo que ya ha sido explicado anteriormente.



- Listar Solicitudes: Representa el punto de partida del trabajo diario del personal del *Help Desk*, pues incluye la posibilidad de listar las solicitudes de servicio recibidas en función de determinadas condiciones (por ejemplo, con un status específico o que se encuentren asignadas a algún analista determinado) ofreciendo el mecanismo de ubicarlas para luego pasar a consultarlas.

Este caso de uso también es activado por los solicitantes (de ahí que se asocie con el actor más general, “Usuario”) que según los niveles de seguridad que se definan podrán listar, por ejemplo, todas las solicitudes originadas por ellos mismos.

- Consultar Solicitud: Este caso de uso implica el poder acceder a toda la información asociada a una solicitud, lo que incluye entre otras cosas el histórico de avances que para ella se han registrado y la información pertinente a su estado actual y su origen. Nuevamente es una funcionalidad accesible por todos los usuarios quienes podrán ver distintos elementos de cada solicitud según su tipo. Esta funcionalidad representa el corazón del *Help Desk* y, como puede verse en la figura gran parte de los demás casos de uso pueden ser activados a través de ella.

- Registrar Avance: Esta funcionalidad, disponible sólo para los analistas, hace referencia al registro de acciones, hallazgos, recomendaciones o notas de cualquier tipo que pasan a formar parte del historial de una solicitud.

- Enviar comunicación: Representa los mecanismos que soportan la interacción entre el solicitante y el *Help Desk*, permitiendo el envío de mensajes a través de medios varios o su registro en caso de que ocurran por vías alternas al sistema. Por esto, puede ser activado tanto por un analista como por un cliente.

- Cerrar solicitud: Como indica su nombre, es la funcionalidad a través de la cual se registra la resolución de una solicitud junto con la información pertinente.

En la figura 6.7 podemos observar los casos de uso no incluidos en el diagrama principal por no están directamente asociados con el proceso de atención de las solicitudes. Casi la totalidad de ellos son producto del refinamiento de las funcionalidades que durante la fase de Inicio fueron identificadas y representadas con los casos de uso “Administrar...” por lo que pueden ser activados únicamente por el actor “Administrador”. Incluyen las posibilidades de ingreso y actualización de datos para los conceptos involucrados en el funcionamiento del *Help Desk*: analistas, acuerdos, niveles, recursos, contratistas y solicitantes.

Adicionalmente se incluyen en este modelo los casos de uso que permiten la consulta de los datos de estos mismos conceptos antes listados. Estos, como puede apreciarse en la figura, en la mayoría de los casos representan un nuevo mecanismo de acceso al caso de uso más importante del sistema, Consultar Solicitud, pues típicamente se querrán conocer todas las peticiones asociadas a un solicitante, o asignadas a determinado analista, etc.

Este grupo de casos de uso “Consultar...” son los que permiten llevar a cabo las actividades de seguimiento del funcionamiento del *Help Desk*. Algunos de ellos pueden también ser accedidos por los analistas quienes en su labor diaria pueden requerir información acerca de, por ejemplo, los niveles y su estado actual o los solicitantes que puedan estarles contactando para pedir respuestas o emitir solicitudes.



c) Se elaboró un **diseño de la interfaz gráfica** que dictó los lineamientos a seguir en la implementación del prototipo evolutivo. En él se reflejan los mecanismos

de interacción entre el sistema y sus usuarios derivados de la especificación de casos de uso, así como de los requerimientos identificados durante la fase de inicio.

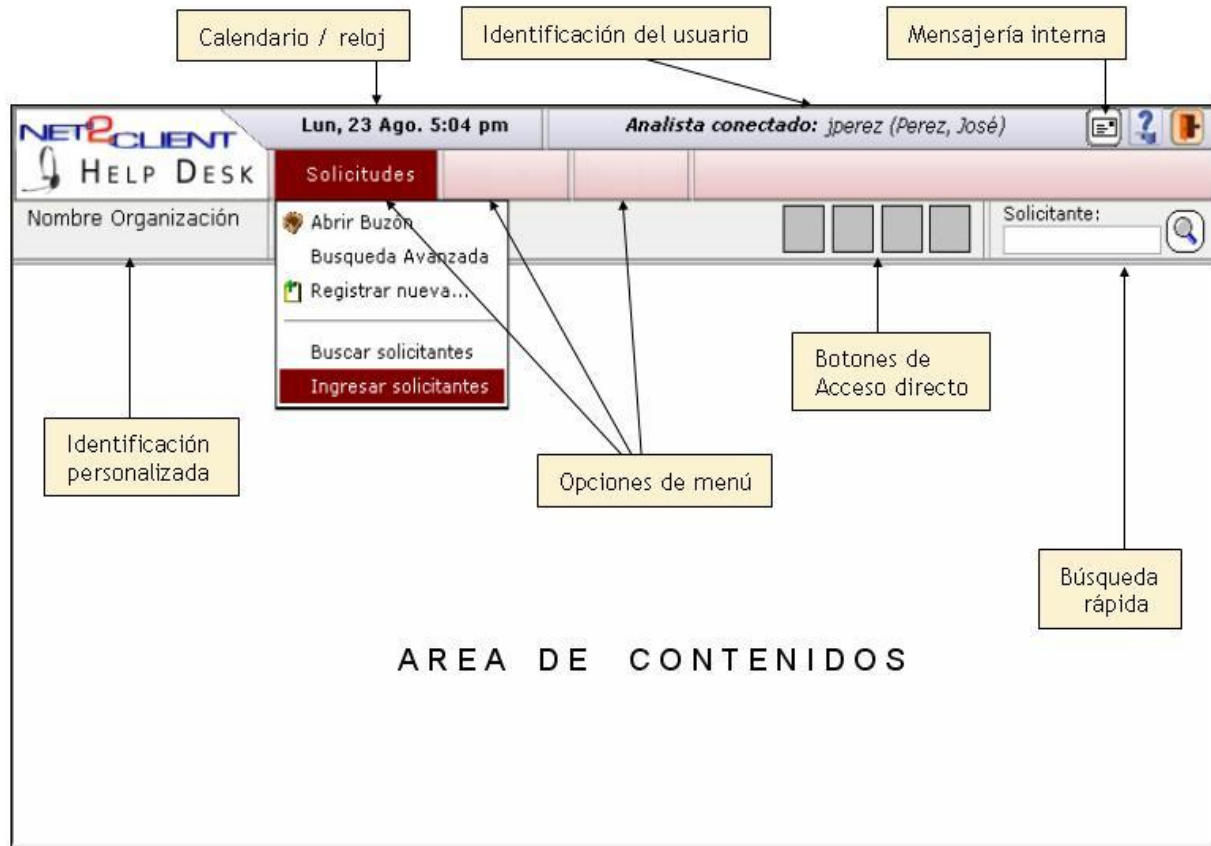


Figura 6.8 - Diseño de la interfaz gráfica del prototipo evolutivo

En la figura 6.8 se nos presenta una pantalla genérica de ejemplo de *Net2Client Help Desk*, donde pueden observarse la apariencia básica del sistema y los elementos que componen su interfaz gráfica a la luz de este diseño concebido durante la fase de elaboración para el prototipo inicial.

Cabe destacar que el *look & feel* que exhibe la pantalla está pensado para asemejarse lo más posible a las pantallas que *Net2Client* provee automáticamente para las aplicaciones que son generadas a través del mismo. Igualmente se concibió un

diseño sencillo de logotipo que se basa en la marca ya existente de la plataforma para apoyar la noción de presentar al producto elaborado como perteneciente a un conjunto de sistemas integrados.

También es importante la presencia de elementos como los *botones de acceso directo* y el *panel de búsqueda rápida*. Ambos buscan acelerar los mecanismos de activación de las funciones más comunes en el uso de la herramienta, siguiendo la idea de que *Net2Client Help Desk* es un sistema que maneja una función crítica y muy dinámica dentro de la organización, por lo que debe permitir a sus usuarios realizar gran cantidad de acciones exigiendo de ellos la información mínima y ofreciéndoles los tiempos de respuesta más bajos.

d) Se diseñó la estructura básica de la **arquitectura del sistema**, y se estableció su línea base -sobre la cual se apoyó el grueso del desarrollo- al probar su viabilidad a través de la elaboración del primer prototipo evolutivo de *Net2Client Help Desk*, el cual implementó las funcionalidades básicas incluidas en los Casos de Uso identificados como críticos que fueron explicados anteriormente.

El diseño de esta arquitectura responde directamente al producto de las distintas actividades que se llevaron a cabo durante las primeras dos fases del presente desarrollo. Así, es consecuencia directa de los requerimientos identificados e incluso de las funcionalidades especificadas. Igualmente se encuentra influenciado en gran medida por el hecho de ser un sistema orientado al ambiente Web, y especialmente el pertenecer y ser desarrollado a través de una plataforma de generación de aplicaciones como *Net2Client*.

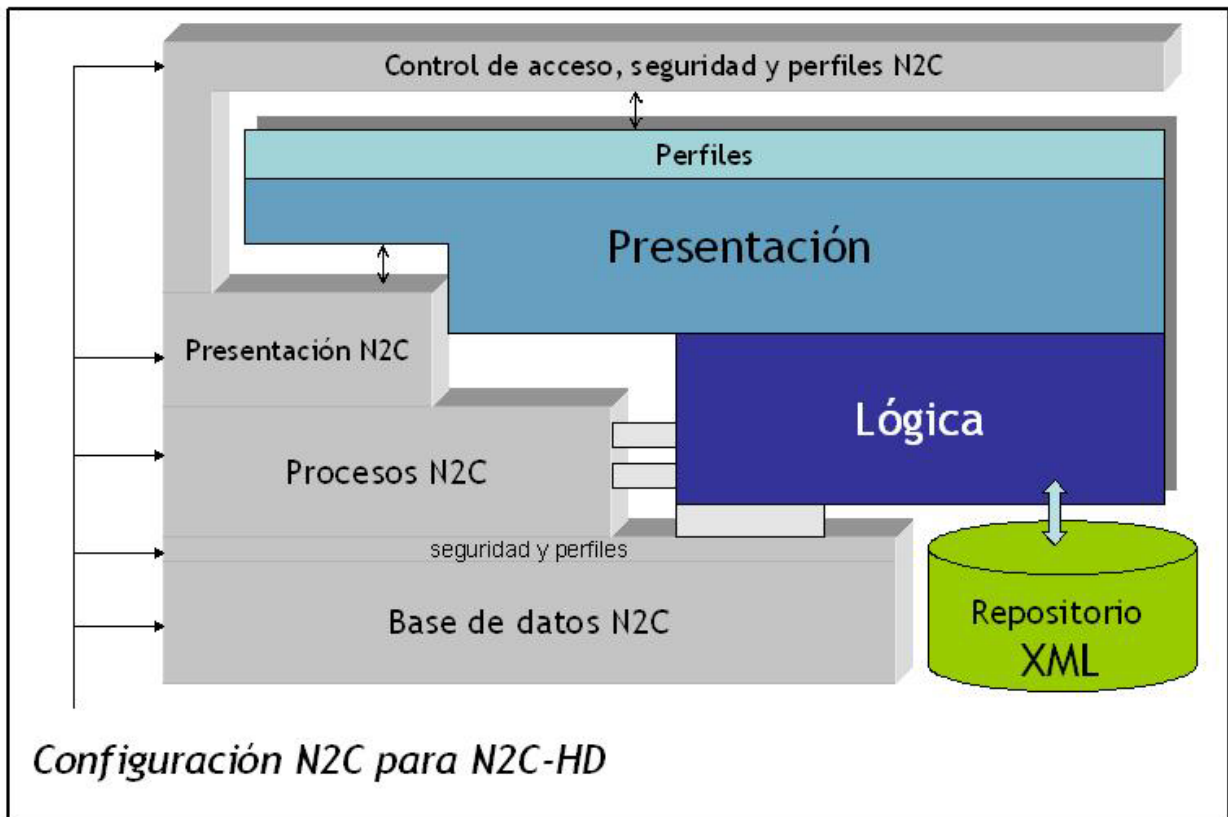


Figura 6.9 - Arquitectura física de Net2Client Help Desk

El diagrama de la figura 6.9 nos presenta los distintos elementos que componen el diseño de la arquitectura física de *Net2Client Help Desk*, así como sus interrelaciones y mecanismos de integración con los componentes básicos de una aplicación generada por *Net2Client* (que ya fueron descritos en el capítulo 3). Veamos ahora un poco más en detalle estos aspectos.

- Ambiente y configuración: Siendo un sistema integrado a la plataforma *Net2Client*, el producto elaborado necesariamente debía incluir los componentes de las aplicaciones generadas por la misma. En la figura se presentan marcados en color gris oscuro y puede verse cómo el comportamiento de cada uno de ellos está afectado

por la configuración específica de *Net2Client Help Desk*, señalado por el recuadro blanco que en todo momento contiene a la arquitectura completa.

- Estructuración por capas: Los componentes del sistema se agrupan en tres capas básicas que enfocan aspectos distintos de su estructura y se comunican entre sí para su funcionamiento. Cada una de ellas incluye un conjunto de elementos que son tanto propios (elaborados exclusivamente para *Net2Client Help Desk*) como generados en cierto nivel por la herramienta de desarrollo. Estos elementos y los mecanismos a través de los cuales interactúan serán descritos a continuación:

→ ***Capa de Presentación***

Esta capa comprende todos los mecanismos responsables de interactuar de manera directa con el usuario, bien sea para presentarle datos o para permitirle ejecutar acciones. Sus componentes pueden verse en la figura 6.9 como las dos piezas azules de tono claro -marcadas como “Perfiles” y “Presentación”- y los dos bloques grises superiores -marcados como “Control de acceso, seguridad y perfiles N2C” y “Presentación N2C”- pertenecientes estos últimos a la arquitectura básica de las aplicaciones *Net2Client*.

El acceso al sistema se da siempre a través del primer bloque de “Control de acceso...” que encapsula todo el manejo de autorización de usuarios y manejo de sesiones de manera transparente. Igualmente añade ya cierto nivel de lógica al restringir ciertas opciones en función de los perfiles que sean definidos en la configuración del sistema dentro de la plataforma.

Este bloque se comunica con los elementos que integran la sub-capa de “Presentación” (en color azul claro en la figura 6.9) la cual interpone en su parte

superior un nuevo nivel de lógica de “Perfiles” que actúa como un nuevo filtro que se integra al ya proveído por el bloque de “Control de acceso...” de *Net2Client*. Los elementos de esta sub-capa son los que implementan la interfaz gráfica que se presenta al usuario y que recoge sus respuestas para en función de ellas activar los distintos procesos en la capa inferior.

Adicionalmente, para ciertas funcionalidades de administración y otras de consulta genéricas, la sub-capa de “Presentación” puede apoyarse en los elementos ofrecidos por el bloque de “Presentación N2C” (esto está señalizado en la figura 6.9 por la flecha bidireccional que les une). Este bloque no sólo se encuentra configurado de una manera específica según la lógica de negocio de *Net2Client Help Desk*, sino que adicionalmente se presenta al usuario siempre dentro de la interfaz gráfica cuyo diseño se expuso anteriormente en este mismo capítulo; más específicamente, en la porción de la figura 6.8 marcada como “Área de Contenidos”.

→ *Capa de Procesos*

En esta capa se realiza todo el procesamiento de los datos de la capa inferior antes de ser enviados a la superior. Igualmente se ejecutan en ella todas las acciones solicitadas por el usuario según la lógica de negocios desarrollada.

Está compuesta por una sola sub-capa, señalizada en la figura 6.9 en tono azul oscuro como “Lógica”, la cual es accedida únicamente por la sub-capa de “Presentación” que se apoya directamente sobre ella. La capa de “Lógica” incluye un conjunto de rutinas, librerías, utilitarios, clases y procesos que son los que verdaderamente implementan las acciones de las funcionalidades como fueron definidas durante el diseño del sistema.

Adicionalmente puede hacer uso de los elementos presentes en el bloque de “Procesos N2C” con el cual se comunica mediante los mecanismos de interfaz que este provee. Así, la sub-capas de “Lógica” puede activar ciertos procesos en este bloque bien sea para que éste después se comuniquen de manera directa con la capa inferior, o para que sus resultados sean pasados a la capa superior a través de la vía interna de *Net2Client* -es decir, comunicándose de forma directa con el bloque de “Presentación N2C”- para ser mostrados luego por la sub-capas de “Presentación”.

→ *Capa de Datos*

En esta capa se incluyen todos los mecanismos de almacenamiento y acceso a la información perdurable del sistema. Su componente principal es una base de datos relacional, la cual corresponde al bloque inferior (llamado “Base de Datos N2C”) en la figura 6.9. Este componente es accedido en todo momento pasando a través de la configuración de “seguridad y perfiles”, bien sea de manera directa por parte del bloque de “Procesos N2C” o indirecta por parte de la sub-capas de “Lógica” a través de las rutinas que este bloque ofrece para ello.

Adicionalmente se integra a esta capa un repositorio alternativo (representado en la figura 6.9 por un cilindro de color verde) conformado por un conjunto de archivos en formato XML que, a diferencia de los demás componentes, se ubican en el “espacio” de cada aplicación² pues serán particulares a cada instalación. El mismo es accedido directamente por la sub-capas de “Lógica” a través de rutinas que procesan datos en este formato y se utiliza para almacenar información

² Los demás componentes que integran *Net2Client Help Desk* se ubican en el “espacio de paquetes”. Estos espacios hacen referencia a la estructura de una aplicación generada por *Net2Client* tal y como fue descrita en el capítulo 3.

relacionada con la configuración del sistema y del proceso de atención de solicitudes, para cada instalación de la herramienta.

e) Durante todo el proceso de diseño y estabilización de la arquitectura se realizó una constante **evaluación de incidencia de los riesgos** que habían sido identificados durante la primera fase de este desarrollo. Efectivamente, algunos de ellos se presentaron trayendo consecuencias tanto para el diseño elaborado como para el cronograma inicial del proyecto. Los riesgos que se presentaron en una magnitud considerable fueron básicamente dos, incluidos en la tabla 6.2 con los números 2 y 3.

En primer lugar estuvo la existencia de restricciones técnicas al desarrollo impuestas por la herramienta. En efecto, debido a la naturaleza compleja del problema de administración de un *Help Desk* y a la necesidad de que el producto a elaborar se adaptara a cualquier realidad e integrara con otros sistemas pertenecientes a la plataforma, el diseño que se iba construyendo para definir el funcionamiento del sistema exigía la aplicación de mecanismos que no eran permitidos por la herramienta de generación que se estaba utilizando.

Más específicamente, la necesidad de orientar el diseño al cumplimiento de ciertos estándares entre los paquetes integrantes de la plataforma forzaba a que se manejaran completamente por separado a los usuarios internos del sistema -los analistas del *Help Desk*- y los usuarios externos -sus clientes-. Sin embargo, la herramienta sólo prevé un conjunto de usuarios, y el mecanismo de control de acceso, seguridad y perfiles sólo está diseñado para esa estructura. De manera que se restringían mucho las posibilidades de acceso al sistema para aquellos usuarios no analistas, impidiendo entonces la implementación apropiada de los casos de uso que

podían ser accedidos por los actores “Usuario” y “Cliente”. Siendo que esto era crucial para los objetivos del sistema a desarrollar y aprovechando que la herramienta en cuestión pertenece a la misma empresa y se encuentra en constante renovación, se tomó la decisión de ir realizando modificaciones a la misma como actividad paralela a la implementación del primer prototipo evolutivo. El producto de esto fue que el sistema pudiese contar con mecanismos que permitan la existencia de otro grupo de usuarios alternativo, al haberse eliminado esta restricción de la herramienta de desarrollo. Sin embargo, esto ocasionó un retraso de una semana y media en la segunda iteración de la fase.

Por otro lado, también existió un pequeño retraso de media semana durante la primera iteración producto de la incidencia de este mismo riesgo. Más específicamente, la plataforma exige que la integración con otros paquetes se haga compartiendo porciones de sus bases de datos que pasan a ser comunes, a la vez que restringe la cantidad de tablas relacionales que puede contener un sistema. Esto obligó a que *Net2Client Help Desk* utilizara, por ejemplo, tablas distintas para los clientes que son personas naturales y los que son personas jurídicas (siguiendo el estándar de otros paquetes donde estos conceptos si ameritan un tratamiento por separado). Esto redujo el conjunto de tablas disponibles para la implementación de los demás abundantes conceptos que si eran particulares al problema de los *Help Desk*. Así, las estructuras de datos diseñadas debían replantearse cada vez para tratar de hallar opciones que permitiesen cumplir con los estándares y no perder conceptos básicos; lo cual se resolvió a través de mecanismos como la unión de conceptos

distintos en tablas comunes, la eliminación de redundancias menores y el uso de repositorios alternativos, como lo son los archivos en formato XML.

Finalmente, y en segundo lugar, el riesgo de la presión sobre el tiempo de desarrollo significó que el retraso de dos semanas que se presentó en esta fase no pudiera ser amortiguado durante las fases siguientes, pues el cronograma era muy ajustado e inflexible. Esto ocasionó que el plan de desarrollo para la fase de construcción tuviera que ser reestructurado, como se describirá a continuación.

f) Se replanteó el **Plan de Desarrollo de Software** de la Fase de Construcción, ajustándolo al número de semanas disponibles y redefiniendo ligeramente las funcionalidades asociadas a cada incremento a elaborar para reflejar las prioridades identificadas durante la elaboración del prototipo evolutivo inicial. El plan elaborado puede verse en la tabla 6.4.

Iteraciones	Semana Inicio	Semana Final	Duración (semanas)
Iteración C1 - Primer entregable <i>Desarrollo del primer entregable → manejo del ciclo de vida de las solicitudes + asignaciones.</i>	12	14	3
Iteración C2 - Expansión (niveles) <i>Desarrollo del segundo entregable → utilización de niveles y evaluación de solicitudes “vencidas”</i>	15	17	3
Iteración C3 - Expansión (recursos) <i>Desarrollo del tercer entregable → asociación de solicitudes con equipos o recursos.</i>	18	19	2
Iteración C4 (Inicio) - Expansión (enrutamiento) <i>Investigación y diseño de reglas de asignación para desarrollo del cuarto entregable → enrutamiento de solicitudes</i>	20	20	1

Tabla 6.4 - Plan revisado para la Fase de Construcción

6.3 - Fase de Construcción

Esta fase transcurrió durante las últimas nueve semanas previstas para el desarrollo de esta pasantía, cubriendo por completo tres iteraciones. Igualmente se tomó una semana adicional para el inicio de una cuarta iteración.

En cada iteración se llevaron a cabo básicamente dos actividades; primero, una revisión detallada de los casos de uso correspondientes a las funcionalidades asociadas a la iteración en cuestión -así como de aquellos que aunque ya formasen parte del prototipo evolutivo pudieran ser afectados de alguna manera por las nuevas opciones a incluir en el sistema- y segundo, la implementación de estos casos de uso para producir un incremento.

Esta última actividad consistía en la configuración de los parámetros del sistema dentro de la plataforma *Net2Client*, la creación o actualización de las estructuras de datos a través de esta misma herramienta, la codificación de los procesos de negocio utilizando el lenguaje PHP (junto a las rutinas de acceso a los archivos en formato XML) y la elaboración de las pantallas correspondientes a la interfaz gráfica en el lenguaje HTML, agregándoles instrucciones en el lenguaje *JavaScript*.

Resultados:

a) Basándose sobre el prototipo evolutivo que se obtuvo como resultado de la fase de elaboración, se construyó el **primer entregable** del sistema *Net2Client Help Desk*. Este incluye prácticamente todas las funcionalidades identificadas como principales en el diagrama de casos de uso de la figura 6.6. Básicamente, permite el manejo del ciclo de vida de una solicitud. Esto abarca desde el proceso de registro de

la misma -proveyendo a los analistas herramientas que permitan atender rápidamente a los solicitantes y registrar sus peticiones de la manera más eficiente-, hasta el acceso fácil a sus datos y el historial de avances y comunicaciones que en su ciclo de vida se hayan registrado. Igualmente permite realizar acciones básicas de asignación de las solicitudes a los posibles responsables dentro del *Help Desk*.

b) El primer entregable fue expandido para incluir la posibilidad del **manejo de niveles y seguimiento de desempeño** en la herramienta *Net2Client Help Desk*. Esto significó la implementación de un incremento que se incluyó al producto final permitiéndole ahora ofrecer a las organizaciones herramientas para el manejo de niveles de servicio con configuraciones específicas. Esto implica igualmente el poder hacer un seguimiento de los tiempos de respuesta y resolución para cada solicitud al calificarla como “vencida” cuando se excedan los parámetros establecidos para el nivel en que se encuentre actualmente.

c) Una nueva expansión añadió al producto las funcionalidades asociadas con el **manejo de recursos** de la organización. Esto incluye no sólo la posibilidad de llevar un pequeño inventario de cada elemento para el cual el *Help Desk* presta soporte, sino también el poder controlar para cada recurso a quién está asignado, de cuál contratista depende y cuáles solicitudes se han presentado que lo involucran.

d) Finalmente, se diseñó un conjunto de **reglas de asignación** de solicitudes para ser implementadas posteriormente, expandiendo así el manejo de asignaciones que ofrece *Net2Client Help Desk*. La descripción de dichas reglas, junto con el mecanismo mediante el cual han de aplicarse, puede consultarse en el apéndice B: “Reglas de Enrutamiento de Solicitudes”.

6.4 - Estado actual del sistema

El sistema desarrollado se encuentra en la etapa final de su fase de construcción y listo para una breve transición a su lanzamiento al mercado. De hecho, la metodología aplicada durante su elaboración permitió que aún antes de culminado el primer ciclo de vida del *software*, ya se tenga un entregable de calidad que puede ser considerado como un producto final; la versión preliminar de *Net2Client Help Desk*.

→ **Funcionalidades**

Las funcionalidades incluidas en el sistema pueden resumirse -de manera bastante global- en el manejo del ciclo de vida de las solicitudes, el seguimiento del desempeño del proceso de atención a las mismas y el control de los recursos para los cuales se presta el soporte; desarrolladas éstas tal y como fueron descritas en los resultados de la fase de construcción. Esto satisface los requerimientos funcionales y los casos de uso que fueron identificados al comienzo del desarrollo y que entraban dentro del alcance previsto para esta pasantía.

→ **Otras características**

El sistema desarrollado está integrado a la plataforma *Net2Client*, y su diseño cumple con los estándares establecidos para la interconexión entre los paquetes que integran su conjunto de aplicaciones asociadas.

Tanto el proceso de atención a las solicitudes como los formatos de presentación de los datos, así como el nombre de los conceptos más importantes, son completamente configurables de manera que cada instalación de *Net2Client Help Desk*

puede adaptarse en gran medida a la realidad de la organización en la que es implantada, mediante un sencillo y rápido proceso de ajuste. Por ejemplo, pueden existir dos escritorios de ayuda utilizando este producto y en uno pueden manejarse “solicitudes” asociadas a “clientes” que sólo pueden ser empresas y estas pueden avanzar a través de niveles, mientras que en el otro se manejan “reservas” originadas por “pasajeros” de cualquier tipo y que son asignadas a “agentes”.

Este proceso mediante el cual el sistema se va ajustando desde la definición genérica hacia la realidad de cada *Help Desk* que lo adopte consta de: la definición de los parámetros de configuración de la instancia específica de la aplicación dentro de la plataforma *Net2Client*; el ajuste del proceso de atención de solicitudes, ciertos conceptos utilizados y algunos formatos de presentación de datos mediante la actualización de los archivos XML; la definición de los aspectos particulares a cada organización (tipos, grupos y niveles a usar) a través del proceso de poblado de las tablas relacionales; y -en algunos casos- la definición en alto nivel de reportes específicos asociados a la aplicación dentro de la plataforma.

→ **Pruebas**

Como hemos visto antes, la naturaleza de la plataforma *Net2Client* permitió que cada incremento elaborado desde el prototipo inicial pasara a ambiente de producción de manera automática, pudiendo ser sometido a distintos tipos de pruebas de desempeño y adaptación a ambientes reales, paralelamente al desarrollo de más funcionalidades.

En particular, *Net2Client Help Desk* estuvo disponible en su primera versión “Ligera” desde el final de la primera iteración de la fase de construcción. En ese momento, comenzó a ser utilizado internamente por el personal de GWT lo cual representó un gran escenario de prueba generando observaciones y sugerencias que eran incorporadas al desarrollo y facilitaban la especificación de las funcionalidades aún no implementadas.

Adicionalmente, al final de esta pasantía, GWT preparó rápidamente esta primera versión y la ofreció a un importante cliente dentro del sector bancario del país, quien comenzó a utilizarla para apoyar el funcionamiento de su departamento de *Help Desk*, que presta servicio a sus empleados y a los trabajadores pertenecientes a otras empresas asociadas. Esta instalación buscaba representar tanto una inducción al uso del producto final, como un escenario real de prueba del sistema. El resultado fue bastante exitoso, con una implantación sencilla y en un período de más de un mes las únicas observaciones recibidas sobre el *software* eran para solicitar la inclusión de funcionalidades que ya estaban siendo integradas a los incrementos posteriores.

→ ***Evolución***

El desarrollo del sistema continuará después de esta pasantía para incluir todos los aspectos que si bien fueron detectados desde el inicio -y se planificó su inclusión- no podían ser abarcados por ella debido a los lapsos de tiempo.

En primer lugar se culminará la implementación de las reglas diseñadas al final de este proyecto cerrando así la cuarta iteración de la fase de construcción. Sobre ese último incremento se realizarán en principio dos iteraciones más. La primera buscará

expandir el sistema para que incluya el manejo de bases de conocimiento que utilicen los registros tomados de solicitudes pasadas y la posibilidad de elaborar FAQ's basados en archivos XML que puedan ser ofrecidos a los solicitantes tratando de reducir el número de solicitudes y promoviendo el autoservicio. La segunda sería para desarrollar un módulo que ofrezca funcionalidades de configuración de la herramienta; es decir, facilidades que permitan modificar el contenido de los archivos XML en el repositorio secundario de manera transparente.

CONCLUSIONES Y RECOMENDACIONES

El desarrollo de este proyecto de pasantía abarcó muchos aspectos relacionados con la ingeniería de computación. Varios de ellos representaron la aplicación en la realidad de las nociones teóricas adquiridas por el pasante a lo largo de sus estudios, mientras que tantos otros correspondieron a conceptos nuevos que pudieron ser aprendidos y ejercitados. De toda esta experiencia se desprenden una serie de conclusiones que son enumeradas a continuación, indicando cuando sea el caso las recomendaciones que de ella se deriven:

→ Los procesos organizacionales que involucran atención al cliente, y en especial los asociados al funcionamiento de un *Help Desk*, deben manejarse haciendo un uso inteligente de la información que permita ofrecerle respuestas óptimas a dicho cliente. Para esto se hace indispensable que sean **soportados por software** especializado en el apoyo a la gerencia y automatización de estos procesos.

→ El diseño de sistemas bajo el modelo de arquitectura de tres capas, combinado con las tendencias existentes para las **aplicaciones Web**, fomenta la elaboración de sistemas enfocados al soporte de las interacciones entre el ámbito interno y el externo de una organización. Por ello suelen presentarse con tanta frecuencia en las herramientas para manejo de servicios de **atención a clientes**, como la que fue desarrollada durante esta pasantía.

→ La **utilización de herramientas de generación** de aplicaciones puede traer grandes ventajas en el desarrollo de sistemas, disminuyendo considerablemente el

tiempo necesario para su implementación y ofreciendo un nuevo paradigma para la concepción y diseño de sus estructuras. Esto se debe a que se promueve la creación de prototipos evolutivos -que permiten un mejor conocimiento del problema en etapas iniciales del desarrollo, y que pueden funcionar como un mecanismo de comunicación directo entre el desarrollador y los interesados- para luego irlos adaptando conforme el sistema y su diseño maduren. Sin embargo, es siempre muy importante tener en cuenta que es posible que estas herramientas presenten restricciones a los sistemas que en ella se desarrollen, por lo que debe tenerse cautela al emplearlas.

Para el caso específico de *Net2Client* como herramienta de desarrollo, se recomienda que se trabaje en la eliminación o flexibilización de las restricciones que impone sobre las estructuras de datos manejados; tal y como fueron descritas en la fase de elaboración del capítulo 5.

→ Para sistemas grandes que atacan un espacio de problema bastante amplio (como es el caso de la administración de un *Help Desk*) es recomendable la aplicación de **modelos de vida incrementales** en su desarrollo. Esto permite ir resolviendo los distintos aspectos que lo conforman poco a poco, a la vez que se va elaborando una herramienta cada vez más sólida. Esto es perfectamente compatible con el uso de herramientas de generación cuyo principal objetivo es el de producir múltiples incrementos consecutivos de manera sencilla y rápida.

→ La **aplicación de metodologías de desarrollo** y la **gerencia apropiada** de un proyecto es vital para que alcance el éxito. Igualmente se recomienda que más allá de las circunstancias atenuantes de cada caso, las planificaciones de los proyectos de

desarrollo de *software* incluyan la suficiente holgura y flexibilidad como para poder absorber los retrasos que puedan darse en sus fases iniciales.

→ La metodología de desarrollo aplicada demostró que el proceso **RUP** es **totalmente ajustable** a proyectos de intereses variados; aún cuando estos no involucren un extenso equipo de desarrolladores ni correspondan a la elaboración de sistemas orientados a objetos (como pareciese ser su objetivo). Lo único indispensable es que el *software* se construya a través de un ciclo de vida de tipo incremental.

→ El uso de repositorios basados en archivos de texto bajo **formato XML** se está convirtiendo en una alternativa importante a la dependencia exclusiva de bases de datos relacionales en ciertos sistemas. Especialmente cuando se trata del almacenamiento de datos utilizados para indicar parámetros de configuración, como es el caso de *Net2Client Help Desk*. El hecho de que el formato XML represente prácticamente un estándar global facilita mucho el desarrollo de mecanismos que procesen información bajo este formato, pues pueden apoyarse en la gran cantidad de librerías y rutinas públicas que se encuentran disponibles para ese fin. También otorga un grado alto de flexibilidad para introducir cambios en la estructura de los datos sin afectar considerablemente al sistema.

→ El enfoque que se tiene al desarrollar una aplicación para satisfacer los problemas específicos de un único interesado es totalmente diferente a la perspectiva que se tiene con la **elaboración de paquetes o herramientas genéricas**. Si bien se tiene más libertad al no tener que responder directamente a las demandas de un usuario exigente, se cuenta con mucho menos apoyo pues no hay una delimitación

clara de las necesidades y el diseño debe tratar de enfocar un mismo problema desde cada ángulo posible que pudiera corresponder a la realidad de algún usuario potencial. Por ello se recomienda en estos casos que una prioridad sea la inclusión del mayor nivel posible de **adaptabilidad y configuración** a partir de la etapa inicial del desarrollo del sistema.

→ Finalmente, con respecto al sistema desarrollado, *Net2Client Help Desk*, puede afirmarse que se **cumplieron los objetivos** planteados originalmente en el capítulo 3. Esto, al obtener un producto bastante configurable que ofrece las funcionalidades principales de la administración de un *Help Desk* tal y como fueron descritas en el último capítulo de este informe.

En este aspecto se recomienda que se continúe el desarrollo sobre el producto hasta implementar las reglas diseñadas para la asignación de solicitudes y todas las funcionalidades adicionales (base de conocimientos y módulo de administración) que se identificaron durante este proyecto, para luego liberar su primera versión oficial. Igualmente sería aconsejable que en el futuro se iniciara un nuevo ciclo de desarrollo sobre el producto tanto para evaluar la posibilidad de su integración con otras herramientas ajenas a *Net2Client*, como para corregir errores y ampliar funcionalidades una vez que la plataforma vaya superando sus posibles restricciones técnicas (como por ejemplo un manejo de recursos y de contratistas más independiente y detallado).

REFERENCIAS BIBLIOGRÁFICAS

- [GWT 2004]: GlobalWebTek.com. “Global Web Technologies” Disponible en <http://www.globalwebtek.com> - Última fecha de revisión: enero 2005
- GlobalWebTek.com. “*Net2Client*: Una solución integral”. Documento interno de la empresa (sin publicar), 2003
- GlobalWebTek.com. “*Net2Client* White Paper V1”. Documento interno de la empresa (sin publicar), 2004
- [HDW 2004]: Help Desk Ware Group. “The HelpDesk Software and Information Portal” Disponible en <http://www.help-desk-world.com> - Última fecha de revisión: enero 2005
- [JRB 2000]: Jacobson, Ivar, Rumbaugh, James y Booch, Grady. El Proceso Unificado de Desarrollo de Software. Madrid, Addison-Wesley, 2000
- [RSP 2002]: Pressman, Roger S. Ingeniería del Software: Un enfoque práctico. Madrid, McGraw Hill, 2002
- [RSC 2002]: Rational Software Corporation. Documentación digital de la herramienta “Rational Unified Process (Version 2002.05.00)”. 2002

APÉNDICE A: ESPECIFICACIÓN DE REQUERIMIENTOS

▪ Atributos:

Cada uno de los requerimientos identificados fue caracterizado con ciertos valores para determinados atributos, permitiendo así un seguimiento más sencillo del progreso en el cumplimiento de los mismos durante el desarrollo del sistema. Estos valores permitieron hasta cierto punto medir el éxito y la completitud del proceso. Más adelante serán descritos los tipos de requerimientos que fueron identificados y cuáles atributos se están teniendo en cuenta para cada uno de ellos.

A continuación se especificarán más en detalle cada uno de estos atributos, los valores que puede tomar cada uno y su significado:

Atributo Descripción	Tipo	Valores	Significado
Prioridad <i>Indicación del nivel de urgencia para la satisfacción de este requerimiento.</i>	Lista	Crítico	Su implementación es esencial. Su ausencia implica que el sistema no refleje las necesidades de los interesados y sea rechazado.
		Importante	Debe estar presente en la versión final y puede afectar la satisfacción de los interesados pero no interferirá en la planificación
		Útil	Aunque ofrecen una funcionalidad, no forman parte del objetivo principal del uso del sistema y no afectan otras funcionalidades. No producirán un impacto significativo en la satisfacción del cliente.
		Prescindible	Su inclusión es deseable, pero puede omitirse sin afectar a los interesados.
Status <i>Estado del requerimiento</i>	Lista	Propuesto	Fue sugerido pero aun no se ha aprobado su inclusión.
		Pendiente	En espera para ser satisfecho.
		Satisfecho	Las funcionalidades asociadas a este requerimiento ya fueron implementadas
		Descartado	Se excluyó del alcance del sistema.

Impacto <i>Grado en que afecta al sistema completo</i>	Lista	Alto	Afecta al sistema de manera significativa.
		Medio	Afecta al sistema ligeramente.
		Bajo	Prácticamente no afecta al sistema.
Afecta la arquitectura <i>Determina si este requerimiento influye en la arquitectura del sistema</i>	Booleano		
Complejidad <i>Dificultad (a nivel técnico) para satisfacerlo</i>	Lista	Alta	Requiere gran esfuerzo para poder satisfacerse.
		Media	Puede satisfacerse con un esfuerzo relativamente moderado.
		Baja	Puede satisfacerse de manera sencilla.
Iteración Planeada <i>Numero que identifica en cual iteración se satisfecerá este requerimiento</i>	Texto		

▪ Tipos de requerimientos:

Los requerimientos son identificados en función de ciertos aspectos que permiten clasificarlos según una serie de tipos que asocian características comunes a los mismos. Esto posibilita la asignación de distintos niveles de prioridad a conjuntos globales de requerimientos según sea su tipo, facilitando también la trazabilidad entre éstos y las características identificadas en la visión del producto y los casos de uso determinados para el modelo inicial. Para el desarrollo de *Net2Client Help Desk* se han identificado los siguientes tipos de requerimientos (para cada uno se especifica la nomenclatura a emplear):

Tipo	Nomenclatura	Descripción	Atributos
Requerimientos Funcionales	FUNCC- xx	Características observables desde afuera que debe poseer N2C-HD para prestar la funcionalidad básica y satisfacer una necesidad directa.	- Status - Prioridad - Afecta la arquitectura - Complejidad - Iteración planeada
Requerimientos No Funcionales	NOF- xx	Características no observables de forma directa pero si medibles que representan un punto de interés.	- Status - Prioridad - Afecta la arquitectura - Complejidad - Impacto
Requerimientos Técnicos	TEC- xx	Características relacionadas con aspectos técnicos (ej: infraestructura, apego a estándares, conectividad con otros sistemas, etc.) necesarias para el desarrollo e implantación del sistema.	- Status - Prioridad - Afecta la arquitectura
Requerimientos de Seguridad	SEG- xx	Especificaciones en cuanto a niveles de seguridad y políticas para el manejo de datos y acceso a funcionalidades que debe implementar el sistema.	- Status - Prioridad - Complejidad - Impacto
Requerimientos de Interfaz	INT- xx	Características pertinentes a la presentación visual de los datos, operaciones y la navegación entre funcionalidades.	- Status - Prioridad - Complejidad
Requerimientos de Desempeño	PERF- xx	Solicitudes específicas sobre aspectos medibles del desempeño del sistema para proveer mayor eficacia (ej: tiempos de respuesta para ciertas operaciones, tolerancia a fallas, etc.)	- Status - Complejidad

▪ Estado inicial de requerimientos:

A continuación se listarán los principales requerimientos identificados, identificando para cada uno el valor de cada uno de sus atributos al comienzo del presente desarrollo.

Código	Descripción	Valores	
FUNC-01	Permitir la recepción de nuevas solicitudes y el registro de sus características (así como documentación asociada).	Status	Pendiente
		Prioridad	Crítico
		Afecta Arq.	Si
		Complejidad	Baja
		Iterac. plan	C1
FUNC-02	Facilitar la clasificación de solicitudes según tipos y prioridades configurables	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	Si
		Complejidad	Media
		Iterac. plan	C1
FUNC-03	Controlar el estado (y progreso) de cada solicitud registrada durante cada etapa de su ciclo de vida.	Status	Pendiente
		Prioridad	Crítico
		Afecta Arq.	No
		Complejidad	Baja
		Iterac. plan	C1
FUNC-04	Permitir el registro de eventos varios asociados con cada solicitud por parte de los analistas responsables.	Status	Pendiente
		Prioridad	Crítico
		Afecta Arq.	Si
		Complejidad	Baja
		Iterac. plan	C1
FUNC-05	Permitir la opción de emitir al cliente una encuesta de satisfacción predefinida de manera automática al cerrarse una solicitud.	Status	Propuesto
		Prioridad	Prescindible
		Afecta Arq.	No
		Complejidad	Baja
		Iterac. plan	C2
FUNC-06	Permitir a los clientes consultar el estado de sus solicitudes y los eventos asociados a la misma	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	Si
		Complejidad	Media
		Iterac. plan	C1
FUNC-07	Facilitar la documentación de sugerencias de soluciones para solicitudes de cierto tipo.	Status	Pendiente
		Prioridad	Útil
		Afecta Arq.	Si
		Complejidad	Baja
		Iterac. plan	C5

FUNC-08	Ofrecer al cliente la posibilidad de consultar soluciones de auto-ayuda antes de emitir la solicitud	Status	Propuesto
		Prioridad	Útil
		Afecta Arq.	Si
		Complejidad	Media
		Iterac. plan	--
FUNC-09	Permitir la elaboración de FAQ's configurables que puedan ser consultados por cualquier actor	Status	Propuesto
		Prioridad	Útil
		Afecta Arq.	Si
		Complejidad	Media
		Iterac. plan	--
FUNC-10	Permitir la configuración de niveles de respuesta (SLA's) asociados con tipos y características de solicitudes, con usuarios, con grupos de analistas o con contratistas.	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	Si
		Complejidad	Alta
		Iterac. plan	C2
FUNC-11	Hacer un seguimiento del progreso de cada solicitud comparándolo con los niveles estipulados para sus acuerdos.	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	No
		Complejidad	Media
		Iterac. plan	C2
FUNC-12	Generar alarmas automáticas que notifiquen a los responsables de solicitudes fuera de sus niveles.	Status	Pendiente
		Prioridad	Útil
		Afecta Arq.	Si
		Complejidad	Media
		Iterac. plan	C2
FUNC-13	Permitir programar recordatorios o actividades pendientes.	Status	Propuesto
		Prioridad	Prescindible
		Afecta Arq.	No
		Complejidad	Media
		Iterac. plan	--
FUNC-14	Mantener la información de los analistas y grupos que atienden las solicitudes y las características inherentes a cada uno de ellos.	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	Si
		Complejidad	Baja
		Iterac. plan	C1
FUNC-15	Permitir especificar una topología de niveles o líneas en las que puedan distribuirse las solicitudes y/o los responsables.	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	Si
		Complejidad	Media
		Iterac. plan	C2
FUNC-16	Permitir la asignación de solicitudes a uno o más analistas.	Status	Pendiente
		Prioridad	Crítico
		Afecta Arq.	Si
		Complejidad	Media
		Iterac. plan	C1

FUNC-17	Permitir la asignación de solicitudes a colas de espera para ser atendidas.	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	Si
		Complejidad	Media
		Iterac. plan	--
FUNC-18	Permitir la asignación de solicitudes a entes terceros o contratistas para su resolución.	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	Si
		Complejidad	Media
		Iterac. plan	--
FUNC-20	Permitir el escalamiento automático de solicitudes entre niveles según eventos asociados	Status	Pendiente
		Prioridad	Útil
		Afecta Arq.	Si
		Complejidad	Media
		Iterac. plan	--
FUNC-21	Registrar asociaciones entre analistas o grupos de analistas y colas a las que prestan servicio.	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	No
		Complejidad	Baja
		Iterac. plan	C1
FUNC-22	Llevar un control de la asignación de solicitudes a cada analista, grupo o contratista y los eventos que ocurran en el transcurso de la misma.	Status	Pendiente
		Prioridad	Crítico
		Afecta Arq.	Si
		Complejidad	Media
		Iterac. plan	C1
FUNC-23	Apoyar el enrutamiento de las solicitudes a colas (y al ser tomadas, a analistas) basándose en reglas de asignación que optimicen el trabajo de todos.	Status	Pendiente
		Prioridad	Útil
		Afecta Arq.	Si
		Complejidad	Alta
		Iterac. plan	C4
FUNC-24	Ofrecer reportes generados para cada SLA que se establezca, que compare los tiempos estipulados con los reales para las solicitudes asociadas.	Status	Pendiente
		Prioridad	Útil
		Afecta Arq.	No
		Complejidad	Media
		Iterac. plan	C2
FUNC-25	Permitir la definición de actividades programadas que se inicien automáticamente generando solicitudes.	Status	Pendiente
		Prioridad	Útil
		Afecta Arq.	Si
		Complejidad	Media
		Iterac. plan	--
FUNC-26	Permitir el registro de los recursos sobre los cuales se hacen solicitudes	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	Si
		Complejidad	Baja
		Iterac. plan	C3

FUNC-27	Llevar un control de las solicitudes asociadas a cada recurso	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	Si
		Complejidad	Media
		Iterac. plan	C3
FUNC-28	Facilitar la detección de fallas recurrentes en los recursos manejados (en función de las solicitudes que se le asocien)	Status	Pendiente
		Prioridad	Útil
		Afecta Arq.	No
		Complejidad	Media
		Iterac. plan	C3
FUNC-29	Facilitar la agrupaciones de múltiples solicitudes comunes que comparten un único ciclo de vida	Status	Pendiente
		Prioridad	Útil
		Afecta Arq.	Si
		Complejidad	Alta
		Iterac. plan	--
FUNC-30	Facilitar la subdivisión de solicitudes en ciclos de vida paralelos.	Status	Pendiente
		Prioridad	Útil
		Afecta Arq.	Si
		Complejidad	Alta
		Iterac. plan	--
FUNC-31	Permitir la definición de tipos configurables para las solicitudes	Status	Pendiente
		Prioridad	Útil
		Afecta Arq.	Si
		Complejidad	Media
		Iterac. plan	--
FUNC-32	Ofrecer buzones de búsqueda configurables para listar las solicitudes según su estatus, tipo, nivel, responsables y solicitantes.	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	Si
		Complejidad	Alta
		Iterac. plan	--
NOF-01	Permitir que se configure para cada organización el uso o no de niveles de servicio.	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	Si
		Complejidad	Alta
		Impacto	Alto
NOF-02	Permitir que se configure para cada organización el uso o no de grupos de analistas.	Status	Pendiente
		Prioridad	Útil
		Afecta Arq.	Si
		Complejidad	Alta
		Impacto	Medio
NOF-03	Ofrecer mecanismos de búsqueda rápida de solicitantes como apoyo al personal que registra solicitudes en vivo.	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	No
		Complejidad	Media
		Impacto	Medio

NOF-04	Permitir la configuración de los nombres de los conceptos principales y su presentación.	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	Si
		Complejidad	Alta
		Impacto	Medio
NOF-05	Permitir configurar la presentación de una ficha de datos de los solicitantes.	Status	Pendiente
		Prioridad	Útil
		Afecta Arq.	Si
		Complejidad	Alta
		Impacto	Bajo
NOF-06	Ofrecer búsquedas configurables entre los solicitantes.	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	Si
		Complejidad	Alta
		Impacto	Medio
TEC-01	Permitir que el registro del cierre de solicitudes por parte de los analistas pueda hacerse a través de mensajería de telefonía celular	Status	Propuesto
		Prioridad	Útil
		Afecta Arq.	Si
TEC-02	El sistema deberá integrarse a la plataforma <i>Net2Client</i> .	Status	Pendiente
		Prioridad	Crítico
		Afecta Arq.	Si
TEC-03	Permitir que los datos de solicitantes, analistas y recursos puedan ser compartidos con otras aplicaciones de la plataforma <i>Net2Client</i> .	Status	Pendiente
		Prioridad	Importante
		Afecta Arq.	Si
SEG-01	Restringir las acciones que los analistas puedan tomar sobre las solicitudes según perfiles configurables.	Status	Pendiente
		Prioridad	Importante
		Complejidad	Media
		Impacto	Alto
SEG-02	Restringir al cliente la información pertinente al manejo interno de las solicitudes, según niveles de privacidad configurables.	Status	Pendiente
		Prioridad	Útil
		Complejidad	Media
		Impacto	Medio
INT-01	Ofrecer mecanismos para acceder de forma directa a las funciones más críticas del <i>Help Desk</i> .	Status	Pendiente
		Prioridad	Importante
		Complejidad	Baja
INT-02	Seguir los estándares de <i>look & feel</i> de <i>Net2Client</i> para la presentación de las funciones.	Status	Pendiente
		Prioridad	Crítico
		Complejidad	Media

APÉNDICE B: REGLAS DE ENRUTAMIENTO DE SOLICITUDES

Para apoyar el proceso de asignación de las solicitudes dentro de un escritorio de ayuda, *Net2Client Help Desk* debe ofrecer un mecanismo de enrutamiento que ayude a sus usuarios a determinar qué analista o grupo de analistas debe ser responsable de cada una de ellas. Tal mecanismo debe aplicar una o más reglas que garanticen que los recursos se aprovechen lo mejor posible y que cada petición se resuelva de una manera eficiente y sin incumplir ningún acuerdo de nivel de servicio que la organización tenga con sus clientes.

El mecanismo que acá se propone busca precisamente alcanzar estos objetivos, abstrayendo el proceso de enrutamiento de una solicitud en varios pasos desde su recepción hasta su asignación final a un analista. En cada paso se determina cual será el siguiente tramo en la ruta de la solicitud, aplicando una serie de reglas, a saber:

→ **Paso 1:** Al ser recibida la solicitud en el *Help Desk*, debe determinarse a cuál área pertenece; es decir, cuál es el grupo de analistas que deberá encargarse de su resolución y por lo tanto **en cuál cola de espera** será ubicada. Este paso es bastante automático pues la única regla aplicada es la de que cada solicitud debe asignarse al área a la cual pertenece el tipo asociado a ella. Es decir, que en todo momento, cada petición que ha sido tipificada está asociada directamente con un único grupo de analistas, independientemente del estado actual de las distintas colas en el *Help Desk*.

Si tomamos el ejemplo de un escritorio de ayuda tecnológico, donde sus analistas se dividan en “Grupo de *Hardware*” y “Grupo de *Software*”, una solicitud que al

registrarse sea calificada con el tipo “Instalación de nuevos sistemas” quedaría automáticamente asignada al segundo grupo mencionado.

Los casos en que el *Help Desk* en cuestión no utilice subdivisión por áreas, se tratarán como si existiese un único grupo, al cual se encargarán todas las solicitudes.

→ **Paso 2:** Una vez que una solicitud se encuentra asignada a un grupo, el siguiente paso es determinar cuál debe ser su **posición dentro de la cola** asociada a este. Esto implica poder calificar cada una de las peticiones con un valor numérico (o alguna opción equivalente) que permita determinar unívocamente para cada par de ellas, cuál debe ser despachada primero. Según el proceso que defina cada *Help Desk*, este paso puede consistir únicamente en comparar la petición nueva con todas las que están en la cola para ubicar la posición que le corresponde, o reordenar por completo dicha cola para reflejar cambios en las prioridades de las solicitudes que puedan haberse dado con el tiempo.

Los factores que se tengan en cuenta para asignar un valor numérico a cada solicitud van a determinar la dinámica con la cual funcionará el *Help Desk*; es decir, cuál será la tendencia en el servicio de las solicitudes y bajo que circunstancias se le da más prioridad a una petición sobre otra.

El mecanismo acá propuesto contempla que a cada solicitud se le calcule una serie de valores y su calificación final se obtenga sumando dichos valores, ponderados según un conjunto de pesos configurables. Esto permite que en cada *Help Desk* se ajuste la importancia que se da a determinado factor en la evaluación de las solicitudes. Los valores en cuestión se obtendrán según los siguientes aspectos:

- En principio, cada solicitud debe contar con un nivel de *urgencia*, expresado por el cliente que la emite. Sin embargo, dado que su juicio no siempre es el más acertado, un analista que evalúe la petición deberá asignarle una *prioridad interna* de mayor importancia para el *Help Desk*. En función de estos dos valores se calculará la Prioridad de Resolución de la solicitud, que no es más que un factor que representa tanto los intereses del solicitante (que por ejemplo puede necesitar que la solución sea inmediata) como los de la organización (que por ejemplo puede querer asegurarse de que todas las solicitudes del presidente de la empresa sean atendidas apenas se reciben).

- Sobre cada solicitud actúan una serie de acuerdos de niveles de servicio según sus características y las distintas circunstancias en que se encuentre en el momento de su evaluación. Cada uno de ellos puede especificar tanto el tiempo máximo que puede tardar el *Help Desk* en solucionarla (lo que se traduce en una fecha tope para su resolución calculada en función de su fecha de registro) como la cantidad de horas máxima que los analistas pueden dedicar a atenderla (que en muchos casos es utilizado como garantía a los clientes de que el cobro de los servicios no excederá el monto correspondiente a esa cantidad de tiempo). En ambos casos se debe tener en cuenta solamente el mínimo de estos valores, partiendo de la base de que el *Help Desk* no puede incumplir ninguno de sus acuerdos. A estos dos valores se les debe descontar el tiempo que ha pasado desde el registro inicial de la solicitud, y la cantidad de horas que ya se han dedicado a resolverla, respectivamente.

La diferencia entre estos dos valores representa la Holgura de la solicitud; es decir, qué tanto puede permanecer en espera sin estar consumiendo tiempo de atención de los analistas. Mientras menos sea la holgura de una solicitud, menor tiempo puede pasar antes de despacharse, por lo que debe calificarse con un mayor valor final para garantizar que sea tomada por un analista más pronto.

- Al dar prioridad a solicitudes con menor holgura se garantiza que ninguna solicitud sea alienada, pues este valor disminuye conforme las peticiones pasan más tiempo en cola sin ser despachadas. Sin embargo, es posible que esta dinámica provoque que se atienda primero a solicitudes sencillas y se vaya retrasando la resolución de otras más complejas que a pesar de tener una mayor holgura, representan un mayor riesgo para la organización pues requieren de más tiempo de atención por parte de los analistas. Este efecto puede minimizarse si se tiene en consideración también el Porcentaje de Holgura, entendido este como el cociente entre el tiempo máximo de atención que pueda dedicarse a la petición y su holgura calculada como se explico anteriormente. Este valor garantiza que una gran holgura represente menos para solicitudes más complejas y viceversa. A mayor porcentaje, menor será la holgura en proporción con el tiempo que se estima deberá dedicarse, por lo que más pronto deberá despacharse la solicitud.

→ **Paso 3:** Una vez que las colas de atención de cada grupo están ordenadas según se ha descrito, el proceso se transforma en ubicar un analista para asignarle la solicitud que haya resultado en la primera posición. Para ello se deben comparar las circunstancias de cada uno de los analistas que prestan servicio al grupo en cuestión, y

determinar cuál de ellos es el más apropiado, tanto para distribuir las cargas uniformemente dentro del *Help Desk*, como para garantizar el cumplimiento -si es posible- de los acuerdos de niveles de servicio establecidos con los clientes. Se propone que esta comparación se realice en función de los siguientes aspectos:

- Se debe favorecer al analista más disponible, dando prioridad al que tenga un menor tiempo efectivo de trabajo asignado. Ciertamente este valor no puede conocerse con certeza, pero puede estimarse como la suma del tiempo máximo de atención restante de cada una de las solicitudes que tiene asignada (calculado este como se explicó en el paso anterior).

- Para poder reflejar que distintos analistas tienen distintas capacidades de respuesta, a cada uno de ellos se les asocia un valor mínimo y máximo de solicitudes asignadas concurrentemente. Así, se dará prioridad a quienes estén más lejos de alcanzar su mínimo y se descartará a quienes ya sobrepasen su máximo.

- Independientemente de cuán libre este un analista, se debe garantizar que al asignársele una solicitud esté en capacidad de resolverla a tiempo para evitar que se incumplan los acuerdos asociados a ella. Por eso, además de considerar cuánto tiempo de trabajo tiene por delante, se debe tener en cuenta que tan libre es de reacomodar su cronograma para ajustar las horas necesarias para resolver la solicitud en cuestión. Esto puede estimarse a través del cálculo del Grado de libertad del analista. Este valor representa el promedio ponderado de los Porcentajes de Holgura de cada una de las solicitudes asignadas al analista. Es decir la suma del tiempo máximo de atención efectiva de cada petición

multiplicado por su porcentaje de holgura (calculado este como se describió en el paso anterior), y todo esto dividido entre el tiempo total ya estimado.

- Teniendo en cuenta que un analista puede prestar servicio a distintas colas de atención, se puede querer garantizar un equilibrio entre las áreas de las solicitudes a su cargo, para evitar que se enfoque más en las tareas de un grupo que en las de los demás. Para ello debe tenerse en cuenta el porcentaje de atención a la cola correspondiente; entendido éste como el cociente entre el número total de solicitudes asignadas y la cantidad de ellas que provinieron del grupo en cuestión.

→ **Paso 4:** Finalmente, una vez que la solicitud ha llegado a un analista, sólo queda que la misma sea atendida efectivamente por él. Para ello, puede desearse que las peticiones de las cuales es responsable se organicen en una cola de manera similar a lo que ocurría en los grupos. Esto permitiría que en todo momento el analista pueda conocer a cuál solicitudes debe comenzar a dedicarle más tiempo para buscar que todas sean resueltas dentro de los parámetros de tiempo esperados.

El mecanismo que acá se propone, contempla que el conjunto de solicitudes asignadas a un analista sea ordenado de manera análoga al proceso comparativo que se describió en el paso 2, teniendo en cuenta los mismos aspectos.

Cada uno de estos pasos debe poder ajustarse lo más posible a la realidad de diversos *Help Desks*. Esto no sólo hace referencia al ajuste de los pesos numéricos empleados en las fórmulas, sino que también incluye la posibilidad de asignar de

manera dinámica parámetros distintos a los analistas (como por ejemplo un mayor mínimo de solicitudes concurrentes) y de modificar y ampliar los acuerdos establecidos para reflejar las circunstancias de cada caso atendido.

Todo esto además debe presentarse como un simple apoyo a la tarea de los usuarios, permitiendo que este realice cualquiera de los pasos de manera manual y utilice el sistema sólo como una fuente de sugerencia, en caso de que no quiera confiar el enrutamiento por completo a la herramienta. Así, las solicitudes deben poder pertenecer a grupos no correspondientes con los tipos asociados o asignarse a los analistas menos indicados. Igualmente, los despachadores deben poder elegir cualquier solicitud dentro de las colas -así no sea prioritaria- para buscarle responsable y cada analista está en la libertad de atender las peticiones a su cargo en un orden cualquiera.