ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"Sistema Computarizado De Comunicación Y Control De Ingreso A Oficinas"

TESIS DE GRADO

Previo a la obtención del Título de:

INGENIERO EN COMPUTACIÓN
ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS

INGENIERO EN COMPUTACIÓN
ESPECIALIZACIÓN SISTEMAS DE INFORMACIÓN

Presentada por:

Priscilla Fernanda Jiménez Pazmiño

Galo Fernando Solís Vargas

Sara Piedad Soriano González

GUAYAQUIL - ECUADOR

2005

AGRADECIMIENTO

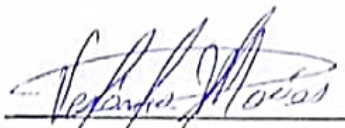
A Dios, a nuestros padres, y a todas las personas que de una u otra manera colaboraron con nosotros en la realización de este proyecto de tesis; especialmente, al Ing. Xavier Ochoa, al Sr. Juan Moreno, y a todo el personal del Laboratorio de Computación de la FIEC.

DEDICATORIA

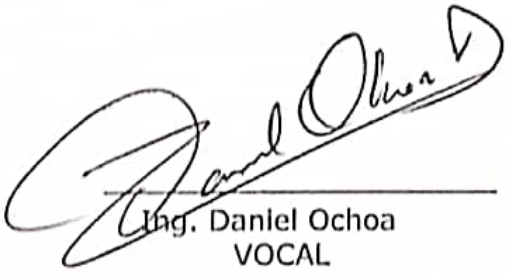
A Dios, a nuestra familia y a nuestros compañeros.

TRIBUNAL DE GRADUACIÓN

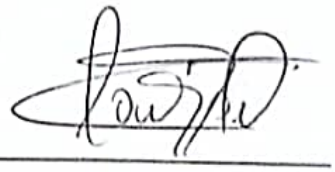
Ing. Miguel Yapur
SUBDECANO DE LA FIEC
PRESIDENTE



Ing. María V. Macías
DIRECTOR DE TESIS



Ing. Daniel Ochoa
VOCAL



Ing. Dennys Paillacho
VOCAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la **Escuela Superior Politécnica del Litoral**"

Priscilla Jiménez Pazmiño

Galo Solís Vargas

Sara Soriano González

RESUMEN

Este proyecto surgió al observar las necesidades que existen dentro de la facultad, en cuanto a comunicación con los profesores que no disponen de oficinas o se encuentran ausentes la mayor parte del tiempo, y a la seguridad y control de ingreso a estas.

El sistema pretende solucionar estos problemas, apoyándose en el uso de dispositivos de captura de audio y vídeo. Para esto, presentará inicialmente una lista de los usuarios, con su respectivo estado de conexión ("disponible", "vuelvo enseguida", etc.); y en caso de que el usuario esté disponible, tendrá la posibilidad de recibir mensajes en tiempo real de parte de la persona que lo esté solicitando, y de observarla por pantalla. De este modo, el usuario podrá decidir si le permite el ingreso, en cuyo caso podrá accionar desde su computadora el mecanismo que abre la puerta. En caso de que el usuario no se encuentre disponible, esté ausente, o no disponga de una oficina, el sistema permitirá que se le dejen mensajes. Si estos son escritos, se

enviarán vía correo electrónico al usuario; de lo contrario, se mandará un e-mail indicándole que ha recibido un nuevo mensaje de vídeo o audio; en ambos casos, se adjuntará una foto instantánea del remitente.

En el Capítulo 1 de la Tesis se describen los objetivos del sistema, y su justificación.

En el Capítulo 2 se explican los fundamentos teóricos en los que se basa el proyecto, tales como la compresión de audio y vídeo, y la transmisión de estos datos por red.

En el Capítulo 3 se presenta el análisis de los distintos componentes del sistema, en términos de requerimientos funcionales, viabilidad e interacción Hombre-Máquina.

En el Capítulo 4 se explica cómo están diseñadas la arquitectura y la interacción del sistema, los diferentes módulos que lo componen, y la comunicación entre estos.

En el Capítulo 5 se describen la fase de implementación y las pruebas que se realizaron en el sistema, para verificar la obtención de los resultados esperados.

Y en el Capítulo 6, se presentan las conclusiones, y algunas recomendaciones que pueden ser de utilidad para implementar funcionalidades adicionales.

ÍNDICE GENERAL

Pág.

CAPÍTULO 3

CAPÍTULO 4

ABREVIATURAS

ADSL	Asimetric Digital Suscriber Line
AOC	Audio Object Component
API	Application Program Interface
ASF	Advanced Streaming Format
ASP	Active Server Pages
AVI	Audio Video Interleaved
BMP	Bit MaP
CCTV	Closed Circuit Television
CD	Compact Disk
DAB	Digital Audio Broadcast
DIO	Diagrama de Interacción de Objetos
DNS	Domain Name System
dpi	Dots per inch
DVB	Digital Video Broadcast
DV	Digital Video
DVD	Digital Video Disk
FIEC	Facultad de Ingeniería en Electricidad y Computación
FPSE	FrontPage Server Extensions
FTP	File Transfer Protocol
GB	Giga Bytes
GHz	Giga Hertz
GIF	Graphical Interchange Format
GOP	Group of Pictures
HTTP	Hyper Text Transfer Protocol
IANA	Internet Assigned Numbers Authority
IIS	Internet Information Services
IP	Internet Protocol
JFIF	JPEG File Interchange Format
JMF	Java Media Framework
JPEG	Joint Photographic Experts Group
JSP	Java Server Pages
Kbps	Kilo bits por segundo
KHz	Kilo Hertz

LAN	Local Area Network
MAC	Media Access Control
MB	Mega Bytes
MBR	Marshal By Reference
MPEG	Moving Picture Experts Group
OGG	Ogg Vorbis
ODBC	Open Database Connectivity
OSI	Open Systems Interconnect
PASC	Precision Adaptative Subband Coding
PHP	Hypertext Preprocessor
PNG	Portable Network Graphics
RDSI	Red digital de servicios integrados
RIFF	Resource Interchange File Format
SCCIO	Sistema Computarizado de Comunicación y Control de Ingreso a Oficinas
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol
TIFF	Tag Image File Format
UDP	User Datagram Protocol
URI	Uniform Resources Identifier
VfW	Video for Windows
VOC	Video Object Component
WAV	Waveform Audio File
XML	Extensible Markup Language

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

1. INTRODUCCIÓN Y JUSTIFICACIÓN

1.1. Aspectos de seguridad y comunicación al interior de la Facultad de Ingeniería en Electricidad y Computación

1.1.1. Situación actual de la seguridad y control de ingreso a oficinas

En la actualidad no se lleva a cabo un control de las personas que se acercan a las oficinas buscando a algún profesor ni de la hora en la que llegan, de modo que no se dispone de un medio fiable para saber si alguien los ha solicitado. Las personas que ingresan son aquellas que disponen de la llave para abrir la cerradura, o aquellas a las que les dan acceso los que se encuentran dentro del área de oficinas. Además, debido a que no se lleva un registro de quiénes ingresan y

quiénes les dan acceso, se hace muy difícil determinar qué personas estuvieron involucradas y cómo ingresaron, cuando se sustraen o se pierden objetos de las oficinas, o cuando ingresan personas no autorizadas a esta área.

1.1.2. Situación actual de la comunicación con el profesor

Las herramientas de comunicación utilizadas dentro de la FIEC son: el Sistema de Comunicación Alumno-Profesor, "Cursos-Web"; el Sistema Interactivo de Desarrollo para el Web, "SidWeb"; y el correo electrónico.

Las dos primeras son herramientas de colaboración, diseñadas para dar soporte Web a los cursos y facilitar la comunicación entre alumnos y profesores. Sin embargo, no involucran comunicaciones fuera del ámbito del curso, lo que deja fuera a los alumnos que no forman parte de éste y a las personas ajenas a la universidad, que pudieran necesitar contactarse con el profesor. De este modo, las herramientas de colaboración mencionadas, no siempre resultan el mejor mecanismo para comunicarse.

El correo electrónico, por otra parte, permite el envío y recepción de mensajes dentro de la red local y a través de

Internet. Sin embargo, es necesario tener acceso a Internet, poseer una cuenta de correo electrónico y conocer el email del destinatario, para poder emplearlo. Esto podría ser un obstáculo en el caso de que personas ajenas a la universidad vayan a buscar a algún profesor, que se encuentre ausente en ese momento, y necesiten hacerle llegar un mensaje, ya que ellos quizás desconocozcan su dirección de correo electrónico o incluso no estén familiarizados con el uso del email.

Por otro lado, algunos comunicados dirigidos a los profesores se hacen llegar a sus oficinas por medio de circulares y avisos por escrito. Esto hace que sea más difícil mantener informados a los profesores que pasan la mayor parte del tiempo fuera de sus lugares de trabajo, o que no tienen oficina.

1.2. Objetivos

El objetivo general de este sistema es:

- Diseñar un sistema que utilice herramientas tecnológicas, orientadas a multimedia, para satisfacer las necesidades tanto de control en el acceso al área de oficinas como de comunicación.

Los objetivos específicos de este sistema son:

- Aumentar la seguridad y controlar el ingreso al área de oficinas.
- Proveer un enlace de comunicación entre los empleados de oficina y las personas que los vayan a buscar a sus lugares de trabajo.
- Proveer un mecanismo que permita a los usuarios que no disponen de oficinas, o se encuentran ausentes de ellas la mayor parte del tiempo, mantenerse al tanto de las personas que los han solicitado.

1.3. Sistemas computarizados afines, que se utilizan en la actualidad

1.3.1. Sistemas de seguridad

Dentro de la amplia gama de sistemas de seguridad disponibles actualmente en el mercado, se encuentran los sistemas de control de acceso y de monitoreo por CCTV (Closed Circuit Television). Se han seleccionado estos dos tipos de sistemas de seguridad por las semejanzas que existen entre las tareas que estos realizan, y las que se llevan a cabo en el sistema que hemos diseñado.

Sistemas de control de acceso

Estos sistemas tienen como objetivo principal controlar el ingreso del personal a un establecimiento determinado; y la tecnología en la que se fundamentan varía, dependiendo de factores como el entorno físico, el número de usuarios, el grado de seguridad requerido, etc.

En sus inicios, el control de acceso se llevaba a cabo por medio de sistemas que requerían el ingreso por teclado de una clave, compuesta generalmente por caracteres alfanuméricos; la clave ingresada era luego comparada con la información registrada en una base de datos, y dependiendo del resultado de esta comparación, el sistema determinaba si debía concederse acceso al usuario. Esta técnica dio paso al surgimiento de otras metodologías, que se basaban en el mismo concepto, pero hacían el proceso más cómodo, seguro y ágil para el usuario. Entre esos sistemas se encuentran, por ejemplo, los que utilizan tarjetas y lectoras de proximidad. Estos permiten controlar el ingreso a áreas restringidas o de alta seguridad, activando cerraduras, barreras, etc.; las chapas y llaves son reemplazadas respectivamente por dispositivos abre puertas o unidades de control, y tarjetas o llaves electrónicas con códigos únicos. De este modo, la

entrada y la salida solamente son permitidas al personal autorizado. Adicionalmente, este equipo suele contar con circuitos que activan alarmas y puertos de entrada para sensores magnéticos, que permiten determinar si alguna puerta ha sido forzada, o si ha permanecido abierta demasiado tiempo. Estos eventos suelen ser registrados en la computadora que controla el sistema, para posteriores consultas y presentaciones de reportes. El software de control de acceso permite obtener información detallada de los movimientos de cada persona incluyendo, fecha, hora, entrada o salida y lugar al que accedió.

Otro tipo de sistemas de control de acceso, cuya tecnología se fundamenta en el uso de biometría. Ésta provee quizás el mecanismo de identificación más preciso, ya que se basa en el reconocimiento de un rasgo corporal único como la huella dactilar, el iris, la córnea, la voz, la conformación de la cabeza, entre otros. Muchos de los sistemas de este tipo aún se encuentran en desarrollo, pero otros ya han sido acogidos, con gran éxito, en numerosas organizaciones, como por ejemplo en bancos, oficinas de aduanas, aeropuertos, etc. Entre los más populares se encuentran las lectoras de mano, y de huella dactilar.

Sistemas de monitoreo por CCTV

Este tipo de vigilancia consiste en ubicar cámaras de CCTV a lo largo de las instalaciones, cerca de las puertas, o en determinados puntos estratégicos, y conectarlas a una computadora especial que funciona como punto centralizado de vigilancia. Las cámaras de CCTV suelen estar asociadas con sensores de cerraduras para tomar fotos en puntos críticos, y tanto las imágenes capturadas como los datos de los eventos que se hayan presentado, se almacenan en discos duros de gran capacidad, para consultas posteriores.

1.3.2. Sistemas de comunicación

Existe una variedad de sistemas computarizados que se utilizan en la actualidad para mantener comunicadas a las personas; entre ellos se encuentran el correo electrónico, la mensajería instantánea, la teleconferencia, la videoconferencia, entre otros.

El correo electrónico permite a los usuarios de Internet, que dispongan de una cuenta de correo, enviar y recibir mensajes de texto (con la opción de adjuntar archivos), que puedan ser revisados en cualquier momento.

La mensajería instantánea, es un punto intermedio entre los sistemas de Chat y los mensajes de correo electrónico; las herramientas de mensajería instantánea son programas regularmente gratuitos y versátiles, que funcionan mientras el computador se encuentre conectado a Internet. Los mensajes escritos se envían a uno o varios destinatarios que los reciben en tiempo real; de modo, que cuando el receptor lo lee, puede contestar en el acto. A las últimas versiones se les han añadido una serie de funciones adicionales como la posibilidad de entablar conversaciones telefónicas, utilizando la infraestructura de Internet; visualizar a la persona con la que se está comunicando; compartir diferentes tipos de archivos y programas, etc.

La videoconferencia y la teleconferencia son sistemas que permiten mantener una comunicación simultánea entre dos o más puntos habilitados y conectados a las redes de transmisión de datos. Con estos sistemas es posible comunicar a varias personas ubicadas en sitios distantes, y establecer una conversación como lo harían si todas se encontraran reunidas en una misma sala.

1.4. Descripción del sistema computarizado de comunicación y control de ingreso a oficinas

Este sistema permite controlar el acceso al área de oficinas, y facilita la comunicación entre el personal interno y cualquier persona que requiera comunicarse.

El sistema consta de tres grandes módulos:

- uno módulo de administración,
- uno de comunicación y
- uno de interacción con los usuarios internos.

El módulo de administración es una aplicación que permite la creación, modificación, consulta y eliminación de los elementos que constituyen el sistema (usuarios, porteros, estados de conexión, etc.). También permite consultar e imprimir reportes de las transacciones registradas en la bitácora, y establecer los datos correspondientes a la configuración del sistema.

El módulo de comunicación, es una aplicación que permite el establecimiento de la comunicación entre usuarios internos y externos, la creación y el envío de mensajes a los usuarios internos, controlar el acceso a las oficinas, y mantener un registro de las transacciones que se realizan. Las computadoras que

ejecuten el sistema portero se encontrarán ubicadas a la entrada del área de oficinas. La interfaz de la aplicación mostrará inicialmente una lista de los usuarios internos con su respectivo estado de conexión, y dependiendo de este estado, el usuario externo podrá establecer una comunicación con él o dejarle un mensaje. Adicionalmente, el sistema portero se comunica con un mecanismo que permite abrir la puerta, en el momento en que se le concede acceso a un usuario.

El módulo que interactúa con los usuarios internos, les permitirá consultar y modificar sus datos (nombre, correo electrónico, teléfonos, etc.) y configuración personal (tipo de mensajes a recibir, contraseñas, etc.); cambiar su estado de conexión; revisar los mensajes que ha recibido; y visualizar el vídeo que está siendo capturado por un portero.

1.5. Justificación de la tesis

Este proyecto de tesis provee una alternativa para solucionar los problemas planteados en el análisis de la situación actual y se enfoca principalmente en los siguientes puntos:

- El sistema es útil para entornos de trabajo en los que parte del personal trabaja fuera de oficinas, ya que permite que los

empleados pueden mantenerse al tanto de las personas que los han estado solicitando, por medio de los mensajes que estos les dejan al utilizar el sistema portero.

- Dado que las funciones que realiza este sistema son incluso más seguras y eficientes que las que llevaría a cabo un portero, es posible prescindir de una persona que se encargue de estas tareas.

- La implementación de este sistema resulta escalable ya que se adapta a los recursos tecnológicos y a los diferentes entornos de trabajo de las organizaciones. Esto se ha logrado al permitir que se configuren un gran número de parámetros, como por ejemplo los dispositivos de captura de audio o vídeo, los tipos de compresores, entre otros.

1.6. Posibles usuarios

Los posibles usuarios para este sistema son aquellas organizaciones con entornos de trabajo en los que no todo el personal dispone de una oficina, o se encuentra ausente la mayor parte del tiempo; con una estructura organizacional clara, y necesidades de seguridad altas relacionadas al control de ingreso. Por ejemplo, las universidades, empresas que brindan atención personalizada a sus

clientes como compañías de bienes raíces, asesorías o consultoría jurídica, entre otros.

CAPÍTULO 2

2. FUNDAMENTOS TEÓRICOS

2.1. Formatos de archivos gráficos, de audio, y de vídeo

Un archivo de audio, de vídeo o de tipo gráfico, consiste en un arreglo de números. Un arreglo es una matriz de datos, todos del mismo tipo, posicionados desde el índice cero por un entero. Cada uno de estos números representa, por ejemplo, en el caso del audio, el volumen y la frecuencia de sonido en un instante de tiempo; puestos todos estos números juntos y ejecutados por el reproductor apropiado, generarán un flujo cambiante de frecuencias y volúmenes, que serán entonces voz, música o efectos de sonido. Los archivos de vídeo se comportan de manera similar, aunque utilizan los números para definir colores, brillo, contraste o coordenadas de los cuadros (*frames*) que componen una "película" o secuencia de imágenes.

La manera en que los números son usados para representar la información de un archivo es denominada "formato". Existen varios formatos para los gráficos, el audio y el vídeo. Cada formato utiliza una fórmula única para determinar el orden y el comportamiento de los números del arreglo que serán usados para representar el gráfico, el audio o el vídeo.

2.1.1. Formatos más comunes de archivos gráficos

En la manipulación de archivos gráficos se utilizan dos tipos de formato: los gráficos de mapas de bits y los vectoriales.

El formato de mapa de bits representa a la imagen como filas y columnas de píxeles. En estos gráficos, el orden de cada píxel se define por su posición en la matriz, por medio de la especificación de una fila y una columna. Los formatos de gráficos basados en mapas de bits más comunes son el GIF (Graphical Interchange Format), el TIFF (Tag Image File Format) y el BMP (Bit MaP).

En el formato GIF, los archivos se almacenan con un máximo de 8 bits por píxel, por lo que su paleta se limita a 256 colores, y ofrece la posibilidad de especificar un color como transparente. El GIF es un formato ideal para utilizar en la

Web, en imágenes pequeñas o de pocos colores, en el dibujo de líneas, imágenes con bloques de color sólido e imágenes con límites definidos entre colores. Los archivos GIF se comprimen, sin que se pierda información durante el proceso de compresión; es decir, una imagen descomprimida es exactamente igual que la imagen original. La resolución estándar de este formato es de 72 dpi, y no es recomendable utilizarlo en impresiones, ya que su calidad se limita al uso en pantalla.

El formato TIFF permite generar imágenes de tipo mapa de bits en alta resolución, y es ideal para fotografías e imágenes complejas que vayan a utilizarse en impresión. La información relacionada con la imagen (marca del escáner, equipo host, tipo de compresión, orientación, muestras por píxel, etc.) puede almacenarse en un archivo y organizarse mediante el uso de etiquetas. El formato TIFF puede extenderse cuando se precise, con la aprobación y adición de nuevas etiquetas.

BMP es un formato estándar que Windows utiliza para almacenar imágenes independientes del dispositivo y de la aplicación. El número de bits por píxel (1, 4, 8, 15, 24, 32 o 64) de un archivo BMP determinado se especifica en un

encabezado. Los archivos BMP con 24 bits por píxel son muy comunes, pero debido a que se comprimen, no son muy apropiados para transferirlos a través de Internet.

Por otro lado, los gráficos vectoriales, se basan en algoritmos matemáticos para recrear la imagen que se desea guardar y manipular. A diferencia de los gráficos de mapa de bits, en los gráficos vectoriales, los puntos no están representados por su posición de fila y columna, sino por la relación espacial que tienen entre sí. Algunos de los formatos más comunes de gráficos vectoriales son el JPEG (Joint Photographic Experts Group) y el PNG (Portable Network Graphics).

JPEG es un esquema de compresión, no un formato de archivo, que funciona muy bien para escenas naturales como fotografías escaneadas. Durante el proceso de compresión se pierde algo de información, pero la pérdida suele ser imperceptible para el ojo humano. Los archivos JPEG almacenan 24 bits por píxel, por lo que son capaces de mostrar más de 16 millones de colores, pero no admiten transparencias ni animaciones. El formato de intercambio de archivos JPEG (JFIF) es comúnmente utilizado para almacenar y transferir imágenes que se han comprimido conforme al

esquema JPEG. Los archivos JFIF que muestran los exploradores Web utilizan la extensión .jpg.

El formato PNG se utiliza principalmente para la transmisión de imágenes por red, y está basado en un algoritmo de compresión sin pérdida. Este formato permite crear imágenes con color verdadero (*TrueColor*), escala de grises y paleta de 8 bits. Para imágenes con color verdadero, el formato utiliza 48 bits por píxel, pudiendo representar imágenes de hasta 281, 474, 656, 710, 976 colores distintos. PNG también utiliza canales alfa para especificar transparencias. A diferencia de GIF, que sólo permite especificar si un píxel es transparente o no, PNG emplea 254 grados de transparencia. Además puede manejar imágenes entrelazadas en 2-D, que permiten tener una visión general de la imagen antes de terminar de descargarla.

2.1.2. Formatos más comunes de archivos de audio

Existen dos tipos de formatos para la grabación y reproducción de archivos de audio. El primero no utiliza compresión para la grabación, y el espectro audible que captura se almacena tal como se recibe; un ejemplo de este tipo de formato es el WAV (Waveform Audio File). El segundo tipo de formato sí utiliza

compresión y sus objetivos principales son ahorrar espacio en el almacenamiento y realizar transmisiones eficaces por la red; un ejemplo de este tipo de formato es el MP3.

El formato WAV es originario de Microsoft Windows 3.1, y es uno de los más utilizados para almacenar sonido. Es flexible, debido a que puede ser comprimido y grabado en distintas calidades y tamaños. Aunque los archivos WAV pueden tener un excelente sonido, comparable al del CD (16 bits y 44,1 KHz. estéreo), el tamaño necesario para obtener esta calidad es demasiado grande (de 20 a 30 MB). Sin embargo, si se utiliza la opción que permite ahorrar más espacio (4 bits y los KHz lo más bajo posible), se presentan problemas tales como baja calidad, ruidos, estática, cortes en el sonido, entre otros. Por esta razón casi siempre se lo usa para muestras de sonido. Su ventaja más grande es la compatibilidad que tiene para convertirse en varios formatos, utilizando el software adecuado.

El formato de compresión de audio MP3 fue creado por el MPEG (Moving Picture Expert Group), un grupo de diseñadores y programadores de normas de compresión de audio y vídeo. La calidad de sonido del MP3 y su pequeño tamaño lo han

hecho muy popular en Internet; su algoritmo se basa en la forma de escuchar que tiene el oído humano, pues las frecuencias que quedan fuera de la audición (mayores a 20KHz y menores que 20Hz) no son registradas en el archivo. Al usar el formato MP3 se puede reducir la pista de un CD en un factor de 12 a 1, (1 minuto de calidad CD en formato MP3 equivale a 1MB, aproximadamente), pero lo más importante es que no pierde calidad de sonido. Al igual que los archivos .zip, los MP3 deben descomprimirse para poder escucharse, y por esto requieren más procesamiento.

2.1.3. Formatos más comunes de archivos de vídeo

Los formatos para almacenar y reproducir vídeo digital se basan en la codificación, compresión, transferencia de la imagen, y sonido capturado. Dentro de los formatos de vídeo más comunes están el AVI (Audio Video Interleaved), el MOV (QuickTime Movie) y el MPEG (Moving Pictures Experts Group).

El formato AVI es un caso especial de archivos RIFF (Resource Interchange File Format). Fue definido por Microsoft e IBM, y es un formato de propósito general para el intercambio de datos multimedia. AVI quiere decir Audio y Vídeo Intercalado; esto significa, que en un archivo AVI los datos de audio y

vídeo son almacenados consecutivamente en capas, es decir, un segmento de datos de vídeo es seguido inmediatamente por otro de audio. Un AVI en VfW (Video for Windows) puede almacenar sólo audio, sólo vídeo o ambos, pero en flujos separados para cada tipo. Para cada flujo se puede utilizar un codec con un factor de compresión distinto, y tanto esta como otra información de definición de formato, se especifican en las cabeceras (*headers*) del flujo.

El formato MOV se originó inicialmente en Macintosh, y es el estándar propietario de la aplicación Quicktime de Apple, que almacena simultáneamente audio y vídeo. Entre 1993 y 1995, Quicktime fue superior al formato AVI de Microsoft tanto en funcionalidad como en calidad. La funcionalidad de la última generación (Quicktime 4.0) también incluye el streaming de vídeo por Internet (la transmisión en tiempo real de vídeos sin la necesidad de primero descargar el archivo completo a la computadora), pero a pesar de esto, el formato MOV está continuamente perdiendo popularidad con el aumento del uso del formato MPEG. Sin embargo, ciertos vídeos codificados con este formato todavía se pueden encontrar en algunos CDs gracias a la habilidad de Quicktime para correr tanto en Macintosh como en computadoras x86.

El formato MPEG proviene de "Moving Pictures Experts Group", una organización internacional que desarrolla estándares para la codificación de imágenes en movimiento. Este formato produce una compresión de datos con una pequeña pérdida de calidad; y es el estándar de compresión de vídeo más popular. Un archivo con extensión MPEG puede ejecutarse bajo cualquier plataforma, y existen cuatro tipos: MPEG1, MPEG2, MPEG3 (obsoleto) y MPEG4.

2.2. Compresión

Los archivos de audio y vídeo digital llegan a tener tamaños considerables, y requieren gran velocidad de transferencia de datos, tanto en la lectura como en la reproducción; por esta razón, se han desarrollado numerosas técnicas para comprimir este tipo de información, y transmitirla de manera más eficiente.

Compresión es el proceso de eliminación o reestructuración de los datos, con el fin de disminuir su tamaño. En la actualidad existe una gran cantidad de algoritmos de compresión/descompresión para audio y vídeo digital, que se ponen a disposición de las diferentes aplicaciones a través del uso de codecs. Un codec (codificador-decodificador) es un algoritmo capaz de comprimir y codificar audio/vídeo en su propio formato, y también de

decodificarlo y descomprimirlo. Son usados en la reproducción, captura y edición de audio/vídeo digital; y pueden ser implementados tanto en software como en hardware. Entre las características más importantes de un codec se encuentra la simetría, y se dice que éste es simétrico cuando sus velocidades de codificación y decodificación son iguales. Del mismo modo, se dice que un codec es altamente asimétrico cuando estas velocidades son muy distintas; esto significa, por ejemplo, que existen codecs que tardan mucho en comprimir/codificar, pero que son muy rápidos decodificando.

Los codecs utilizan distintos sistemas para comprimir audio/vídeo que se pueden distinguir básicamente en dos tipos: codecs sin pérdida y codecs con pérdida.

En los codecs sin pérdida, los datos que se obtienen de la decodificación son idénticos, bit a bit, a los de la fuente original, pero los factores de compresión conseguidos son bajos (menores que 10:1). Una codificación sin pérdidas no puede garantizar un factor de compresión determinado, pues depende de la cantidad de redundancia de la información original.

En los codecs con pérdidas, los datos de la salida de la decodificación no son idénticos, comparando bits, a los de la fuente original; y lo que se pretende, en este caso, es que esta diferencia sea lo menos perceptible posible. Los factores de compresión, sin embargo, son altos, y pueden estar entre 40:1 y 100:1. Sin embargo, este tipo de compresión no debe realizarse en cascada, especialmente si se utilizan distintos algoritmos para comprimir los datos. La cantidad de información perdida depende del grado de compresión y es proporcional a la disminución de la calidad, es decir:

Máxima calidad, máximo tamaño de datos = mínima compresión

Mínima calidad, mínimo tamaño de datos = máxima compresión

2.2.1. Compresión de audio

El oído humano puede detectar cantidades minúsculas de distorsión y aceptar un enorme rango dinámico. Debido a que analiza el sonido con bandas de frecuencia (bandas críticas), algunas técnicas de compresión de audio se aprovechan de este hecho, dividiendo el espectro de frecuencias en bandas para disminuir el flujo de bits.

Toda compresión de datos de audio se basa en la comprensión del mecanismo auditivo, por lo que constituye una forma de

codificación perceptual. El oído es sólo capaz de extraer una cierta proporción de la información contenida en un determinado sonido; a este fenómeno se lo denomina entropía perceptual. Un sistema ideal debe eliminar toda información redundante, dejando únicamente la entropía.

Los algoritmos empleados en la compresión truncan el archivo original, de modo que el sonido obtenido no es el mismo. Lo que sucede es que las frecuencias recortadas son aquellas no perceptibles por el oído humano, de forma que se eligen una tasa de transferencia y una frecuencia determinadas para conseguir que en la extracción de audio, el sonido sea aparentemente como el del original.

La velocidad de transferencia de datos de salida de un codificador es prácticamente independiente de la frecuencia de muestreo de entrada. Esto se debe a que la entropía del sonido se encuentra en la forma de onda y no en el número de muestras que la llevan. A mayor tasa de transferencia resultará un mayor tamaño del archivo y mayor similitud con el original. Por el contrario, cuanto más baja sea la tasa de transferencia, menor será el tamaño del archivo resultante, al igual que su calidad.

Cuando se cuantifica una señal, lo que se hace es asignar un único valor de reconstrucción a un rango de niveles. Esto hace más fácil discernir entre niveles de amplitud, reduciendo el efecto del ruido que se pueda añadir en una transmisión o en un proceso de lectura. Sin embargo, esto trae consigo una distorsión de la señal, debido a que ésta no recupera su amplitud original en todos los puntos, sino un valor próximo, que se le ha asignado. Esta distorsión puede verse como ruido añadido, en una proporción que podemos controlar variando el número de niveles de cuantificación: cuantos más niveles, menos ruido.

En esto se basa la compresión de audio. Cuando en una zona del espectro se puede introducir ruido sin que se oiga, se realiza una cuantificación menos fina (escalones de cuantificación más grandes, que se traduce en menos bits), mientras que en las zonas donde el ruido se hace audible, se asignan más bits. Es en este punto donde se diferencian unos codificadores de otros. El cálculo de la cantidad de ruido que se puede admitir es un dato basado en lo que se llama el "Modelo psicoacústico". Este modelo es completamente experimental, y se realiza promediando la respuesta de muchas personas frente a determinados estímulos. Un buen

modelo permitirá estimar con precisión la cantidad de ruido admisible y la banda en la que puede introducirse con pérdidas mínimas, mientras que las estimaciones de un mal modelo no permitirán comprimir tanto o con tanta calidad.

El procedimiento básico de compresión es el siguiente:

1. *Enventanado de la señal:* tomar muestras durante unos 10 ms (alrededor de 512 muestras). A este intervalo temporal se le denomina ventana de análisis.

2. *Análisis espectral de la ventana:* se divide la señal en sub-bandas, generalmente unas 32, que suelen distribuirse de manera uniforme en frecuencia. Hay que calcular un umbral de enmascaramiento para cada una de estas bandas.

3. Generalmente se usa una FFT (Transformada rápida de Fourier), pero pueden utilizarse otras transformaciones, como por ejemplo la DCT (Transformada Discreta del Coseno). Al aplicar esta transformación, el espectro se divide en bandas de anchura creciente con la frecuencia, lo que simula el comportamiento del oído, que tiene más resolución espectral en baja frecuencia.

4. *Cálculo de los umbrales de enmascaramiento:* esta parte puede hacerse de dos formas. La más simple, y la que se usa para factores de compresión pequeños, es utilizar la energía de las sub-bandas para estimar los umbrales, lo que resulta en un costo computacional bajo. Para elevar los factores de compresión, se necesita afinar más en la estimación, lo que se hace calculando una FFT (o DCT) de muchos puntos (más de 512) o de cada una de las sub-bandas. La decisión de usar uno u otro método es un compromiso entre prestaciones y coste computacional.

5. *Cuantificación:* según los umbrales de enmascaramiento y la velocidad binaria de salida se realiza la cuantificación de los coeficientes de cada banda con un número determinado de bits.

Uno de los mecanismos de compresión más comúnmente utilizados para el audio es la codificación MPEG. Las normas MPEG de audio definen tres capas de codificación, que se distinguen por su tasa de compresión para una calidad de audio percibida determinada. La norma de televisión digital DVB prescribe para el sonido la utilización de las capas 1 y 2

de la especificación MPEG-1 de audio, que prevé cuatro modos principales de transmisión:

- *Stereo:* los canales izquierdo y derecho se codifican de manera completamente independiente.
- *Joint_stereo:* aprovechamiento de la redundancia entre los canales izquierdo y derecho, a fin de reducir el flujo.
- *Dual_channel:* los dos canales son independientes (sonido bilingüe, por ejemplo)
- *Mono:* un solo canal de sonido

La capa 1, también llamada "pre-MUSICAM", usa el algoritmo PASC, desarrollado por PHILIPS. Utiliza una velocidad fija entre las 14 posibles (de 32 a 448 Kbps); la calidad Hi-Fi necesita 192 Kbps por canal de audio (384 Kbps en estéreo). Su principal ventaja es la relativa sencillez para implementar el codificador y el decodificador. La cuantificación de los coeficientes de sub-banda está definida para toda la duración de la trama por un número de 4 bits, permitiendo una codificación de 0 a 15 bits para cada sub-banda.

La capa 2 utiliza un algoritmo que se conoce como MUSICAM; éste es el estándar adoptado para la radio (DAB) y televisión (DVB) digitales europeas. Permite obtener una calidad

equivalente con un flujo menor que el de la capa 1 (reducción del 30% al 50%), a costa de un incremento moderado en la complejidad tanto del codificador como del decodificador. El flujo constante puede escogerse entre 32 y 192 Kbps por canal; la calidad subjetiva Hi-Fi se obtiene a partir de 128 Kbps, es decir, 256 Kbps en estéreo.

La capa 3 utiliza una codificación Huffman y un análisis de la señal basado en la Transformada discreta del coseno. Permite un flujo variable y una tasa de compresión aproximadamente dos veces más elevada que la capa 2, a costa de una complejidad claramente mayor en el codificador y el decodificador, así como de un tiempo de codificación y decodificación más largo. La calidad Hi-Fi se obtiene a partir de los 64 Kbps por canal (128 Kbps en estéreo). Está destinada principalmente a aplicaciones de redes de baja velocidad (por ejemplo, Internet).

2.2.2. Compresión de vídeo

Los métodos que se utilizan para comprimir vídeo recurren a los procedimientos generales de compresión de datos, aprovechando además la redundancia espacial de una imagen (áreas uniformes), la correlación entre puntos cercanos, la

menor sensibilidad del ojo a los detalles finos de las imágenes, y la redundancia temporal entre imágenes sucesivas.

Cuando las imágenes individuales son comprimidas sin referencia a las demás, el eje del tiempo no entra en el proceso de compresión. A esto se denomina *codificación intra o espacial*, y se basa en explorar las redundancias dentro de una imagen, empleando ciertas técnicas desarrolladas para las imágenes fijas, como el estándar de compresión JPEG.

Por otro lado, cuando se involucra en el proceso al eje del tiempo, es posible obtener grandes factores de compresión, ya que se toma en cuenta la redundancia que existe entre imágenes sucesivas, y a este proceso se lo denomina *codificación inter o temporal*. En este caso, una imagen individual existe en términos de la diferencia entre imágenes previas, de modo que si una de éstas es eliminada en la edición, los datos de diferencia pueden resultar insuficientes para recrear la siguiente imagen. Esto se debe a que en lugar de enviar la información de cada imagen por separado, el codificador envía la diferencia existente entre la imagen previa y la actual en forma de codificación diferencial. Para esto, el codificador/decodificador necesita una imagen almacenada con

anterioridad, que le sirva de base para hacer comparaciones con las imágenes sucesivas; el estándar MPEG utiliza esta técnica.

Otro tipo de codificación que suele emplearse es la *codificación bidireccional,* que permite tomar información de imágenes anteriores y posteriores a una imagen observada. Este concepto se ilustra en la figura 2.1. En el centro del diagrama un objeto se mueve revelando su fondo, pero éste no se conoce hasta la siguiente imagen. Entonces se toman los datos de las imágenes anteriores y posteriores, o incluso se utiliza el promedio de los datos, para lograr que el fondo sea descubierto.

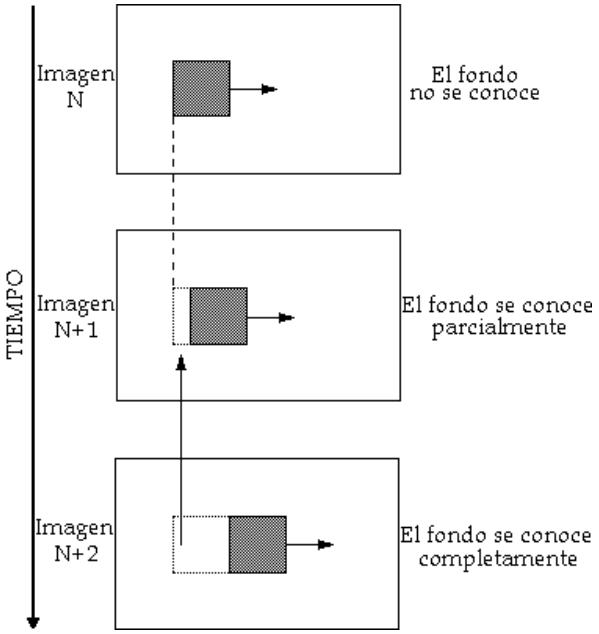


Figura 2.1.- Codificación bidireccional

En la actualidad, se han hecho muchos esfuerzos por desarrollar estándares para la reproducción y transmisión de vídeo. El formato MPEG para la representación codificada y comprimida de imágenes en movimiento y audio asociado, es uno de los estándares de compresión más populares. MPEG aplica la compresión temporal y la espacial; y permite obtener tasas desde 50:1 hasta 200:1. Los estándares MPEG fueron desarrollados para ser independientes de la red específica, proporcionando un punto de interoperabilidad en entornos de red heterogéneos. Este estándar define tres tipos de imágenes que se encadenan según el esquema mostrado en la figura 2.2.

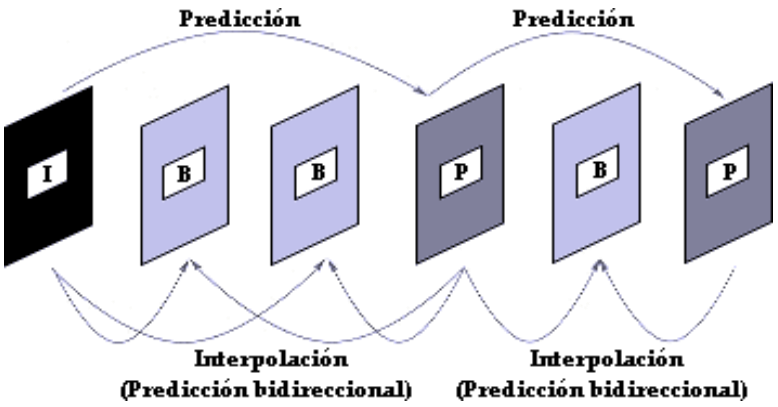


Figura 2.2.- Encadenamiento de los 3 tipos de imágenes MPEG

Las imágenes I o intra no requieren información adicional para su decodificación y representan el punto de partida para una secuencia de imágenes. Debido a que son codificadas sin referencias a otras imágenes, contienen todos los elementos

necesarios para su reconstrucción por el decodificador y su tasa de compresión es pequeña.

Las imágenes P o previstas, por otra parte, se codifican con respecto a las imágenes de tipo I o P anteriores, gracias a las técnicas de predicción con compensación de movimiento. Su tasa de compresión es claramente mayor que la de las imágenes I, ya que requieren aproximadamente la mitad de los datos que éstas.

Las imágenes B o bidireccionales, por último, se codifican por interpolación entre dos imágenes de tipo I o P, la precedente y la posterior. Como no se utilizan para describir otras imágenes, las imágenes B no propagan posibles errores de codificación. Este tipo de imagen es el que ofrece el factor de compresión más alto, que generalmente representa una cuarta parte de los datos de las imágenes I. Por lo general, se utiliza un número limitado de imágenes previstas para garantizar una transmisión de los datos fidedigna, y periódicamente se envía una imagen que no haya sido tratada con ningún método de compresión con pérdidas, idéntica a la imagen original, con el fin de refrescar los datos en la secuencia de transmisión.

Las imágenes suelen ser combinadas para producir un GOP ("Group of Pictures" o grupo de imágenes), que comienza con una imagen I, seguida de un número variable de imágenes P y/o B. El GOP es la unidad fundamental de codificación temporal, y puede ser abierto o cerrado; en un GOP cerrado, las últimas imágenes B de la secuencia requieren de una imagen I, que viene en el siguiente GOP por decodificar.

Adicionalmente a esto, se han definido otros parámetros denominados M y N, que indican la manera en que las imágenes I, P y B se encadenan; M es la distancia (en número de imágenes) entre dos imágenes P sucesivas, y N es la distancia entre dos imágenes I. En el caso de MPEG-1, una de las variedades de MPEG que se mencionan más adelante, para alcanzar un flujo de vídeo de 1.15 Mbps, con una calidad satisfactoria, los parámetros comúnmente utilizados son M=3 y N= 12 como se muestra en la figura 2.3.

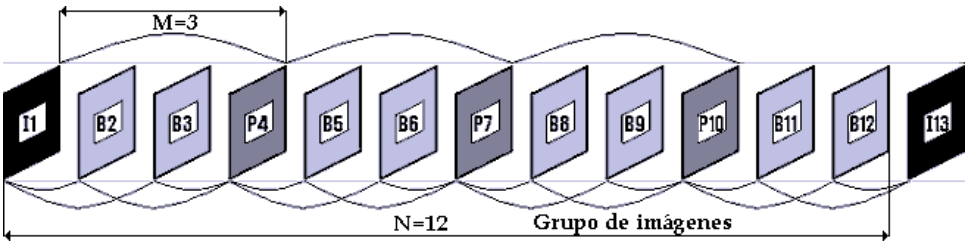


Figura 2.3.- Ejemplo de grupo de imágenes, para M=3, N=12

En este caso, una secuencia de vídeo se compone de 1/12 (8.33%) de imágenes I, 1/4 (25%) de imágenes P, y 2/3 (66.66%) de imágenes B. El factor de compresión global se ve favorecido por el hecho de que las imágenes más frecuentes son las que tienen un factor de compresión más alto. Es evidente que las imágenes de la secuencia de vídeo, al ser visualizadas, deben ser reproducidas en el mismo orden en que se captaron.

Con los parámetros definidos anteriormente (M=3, N=12), el modo de codificación de imágenes sucesivas se traduce por la correspondencia siguiente: 1(I) 2(B) 3(B) 4(P) 5(B) 6(B) 7(P) 8(B) 9(B) 10(P) 11(B) 12(B) 13(I) 14(B) 15(B) 16(P)...y así sucesivamente. Sin embargo, para codificar o decodificar una imagen B, el codificador/decodificador necesitará la imagen I o P precedente y la posterior.

El orden de las imágenes será, por tanto, modificado antes de la codificación, de modo que el codificador y el decodificador dispongan con anterioridad de las imágenes I y/o P necesarias para el tratamiento de las imágenes B, es decir: 1(I) 4(P) 2(B) 3(B) 7(P) 5(B) 6(B) 10(P) 8(B) 9(B) 13(I) 11(B) 12(B) 16(P) 14(B) 15(B)…y así sucesivamente.

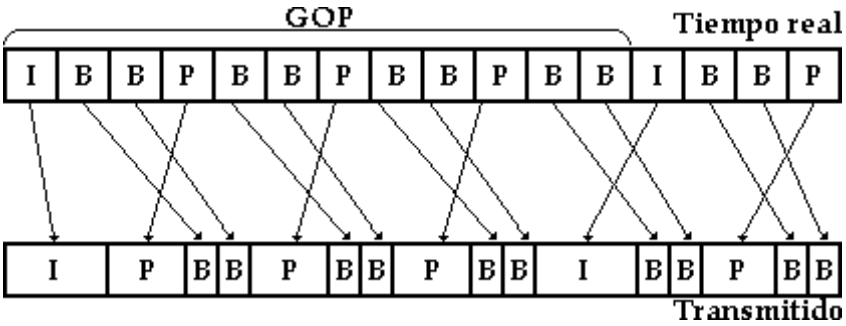


Figura 2.4.- Comparación de las imágenes, antes y después de la compresión, mostrando el cambio de secuencia

De este modo, el aumento en el factor de compresión proporcionado por el uso frecuente de las imágenes B, trae consigo un incremento en el tiempo de codificación y decodificación, y un aumento en el tamaño de la memoria necesaria.

Existen diferentes tipos de formato MPEG, que se describen a continuación.

MPEG-1, que guarda una imagen, la compara con la siguiente y almacena sólo las diferencias (usa tanto redundancia espacial como temporal) y se utiliza para audio y vídeo de CD-ROM.

MPEG-2, que permite transmitir vídeo digital comprimido con velocidades mayores a 1 Mbps y conseguir tasas de hasta

100:1, y se utiliza en televisión satelital, DVD y aplicaciones de vídeo de alta calidad.

MPEG-4, un estándar orientado principalmente a las videoconferencias y a Internet, que define objetos audiovisuales con los que se puede interactuar, mezclando sonido, imagen real, texto y gráficos en dos y tres dimensiones.

En MPEG-4, las imágenes se encuentran divididas en "componentes de vídeo-objetos" (VOC) y "componentes de audio-objetos" (AOC), que son tratados de forma independiente, pero se encuentran relacionados entre sí. En lugar de comprimir una imagen completamente, MPEG-4 utiliza un enfoque basado en capas, en el que se separa el primer plano del entorno. MPEG-4 trata a estos elementos como dos capas diferentes y utiliza distintas compresiones para cada una de ellas. Es rápido codificando vídeo de alta calidad, para contenidos que deben visualizarse en tiempo real y bajo demanda.

2.3. Transmisión de audio y vídeo por red

Las aplicaciones basadas en audio y vídeo cada vez están siendo más utilizadas en diferentes ámbitos, tales como, la educación, comunicación, seguridad, entre otros. Debido al crecimiento masivo del uso común de las redes en organizaciones, se ha vuelto una necesidad en estos últimos años, la transmisión de audio y vídeo sobre redes.

Los usuarios que requieran del contenido multimedia pueden estar conectados a una LAN aislada o distribuidos por todo el mundo. Existen diversos factores que hay que considerar en la distribución vía Internet como son el rango de velocidad de los dispositivos y las líneas de comunicación utilizadas. La conexión del usuario puede ser por módem, con velocidades de hasta 56 Kbps, o por conexión de banda ancha (ADSL, RDSI y otras), pudiendo llegar hasta 2 Mbps.

Los contenidos multimedia son almacenados en Servidores de Streaming o Servidores de Aplicaciones, y su transmisión a través de la red puede ocurrir en dos tiempos de reproducción, en tiempo real (*streaming*), y bajo demanda; y en cualquier caso, los datos viajan por alguno de dos modos de distribución: unicast o

multicast. La relación de los tiempos de reproducción y los métodos de distribución de datos se ilustra en la siguiente tabla:

Tiempo de reproducción	***Método de Distribución***	
	Unicast	*Multicast*
En Tiempo Real	Internet - intranets	sólo en intranets
Bajo Demanda	Internet - intranets	no aplica

Tabla 2.1.- Relación entre tiempos de reproducción y métodos de distribución de vídeo.

2.3.1. Tiempos de Reproducción

La transmisión de multimedia a través de la red puede ocurrir en dos tiempos de reproducción: tiempo real (*streaming*), y bajo demanda.

2.3.1.1 Transmisión en tiempo real

Streaming es la tecnología que permite transmisión de audio y vídeo desde una computadora a otra. El sonido y las imágenes son codificados en datos de computador (bits) y transmitidos de manera continua y sincronizada sobre un canal (LAN, Internet) como "flujo de datos". La computadora receptora toma los bits y los ensambla en sonidos e imágenes que el usuario puede escuchar y ver.

Esta tecnología permite apreciar el contenido conforme se va teniendo acceso a la información del archivo que se encuentra en un servidor, de esta forma no hay necesidad de una descarga previa. Los servidores de streaming transmiten el contenido casi en tiempo real (con un retardo mínimo) y el usuario dispone del contenido de forma prácticamente inmediata.

El servidor de streaming permite visualizar el vídeo de forma continua porque hace uso de un buffer, donde van cargándose algunos segundos de la secuencia antes de que sean mostrados. Entonces cuando se detecta un periodo de congestión de red, se visualizarán los datos que se tengan ya almacenados en el buffer. De esta forma el cliente obtiene los datos tan rápido como el servidor y la red lo permitan. Hay pocos formatos hoy en día que soporten este tipo de visualización progresiva, probablemente en el futuro, el estándar para el streaming vídeo estará en Advanced Streaming Format (ASF).

El streaming puede decirse que funciona de forma inteligente ya que asegura al usuario que recibirá la más alta calidad posible dependiendo de la velocidad de

conexión o de los problemas de conexión de la red. Tradicionalmente, la congestión de la red forzaba al usuario a detener la visualización del vídeo y almacenarlo en un buffer, para posteriormente continuar mostrando la secuencia. Con los nuevos formatos de streaming como el MPEG-4, el cliente y el servidor pueden degradar la calidad de forma inteligente para asegurar una reproducción continua del vídeo.

Si se dan problemas de congestión de red, primeramente el servidor de vídeo disminuye el número de fotogramas que está enviando para mantener la calidad del audio e ir llenando el buffer en lo mínimo. Si las condiciones empeoran, el servidor deja de mandar tramas (*frames*) de vídeo, pero mantiene la calidad del audio. Finalmente, si la calidad del audio empieza a degradarse, el cliente reconstruye de manera inteligente las secuencias que tiene almacenadas para no perder calidad.

Para poder realizar streaming es necesario que la tasa de bits sea menor que el ancho de banda de la red. Entendiéndose por tasa de bits, la velocidad a la que se envían los datos (datos/seg.).

2.3.1.2. Transmisión bajo demanda

En la transmisión bajo demanda es necesario esperar que el archivo sea descargado en el equipo cliente para la reproducción de su contenido; este archivo quedará almacenado y disponible para volver a consultar en cualquier momento. Los servidores de aplicaciones (por ejemplo, Servidores Windows Media) almacenan los contenidos multimedia y el usuario los descarga a su estación.

2.3.2. Métodos de Distribución

2.3.2.1 Unicast

Este servicio consiste en un servidor que envía paquetes de datos a cada computador que solicita un stream. Con unicast el servidor tiene que procesar cada solicitud y despacharla; cada stream toma una pequeña porción de poder de procesamiento del servidor, y el efecto que tiene este método de distribución sobre los recursos de la red es de consumo acumulativo. Cada usuario que se conecta a una transmisión multimedia consume tantos Kbps como la codificación del contenido lo permita.

2.3.2.2 Multicast

La transmisión multimedia dentro de un ambiente corporativo puede alcanzar niveles de audiencia ilimitadas gracias al método de transmisión multicast. Con el método multicast, el consumo de ancho de banda en una red Ethernet es equivalente al de un único usuario, independientemente si se conectan a la transmisión cinco, quinientas, o el número que sea de computadoras simultáneamente. Esta eficiencia se consigue con instrucciones de la capa 3 del modelo OSI, que convierte a cada computadora de un grupo determinado en destinataria de los paquetes de datos multicast que viajan a lo largo de la red Ethernet.

Las transmisiones en vivo pueden aprovechar la eficiencia de este método porque cada computadora recibe exactamente la misma información, al mismo tiempo. Si una nueva computadora se une tarde a la transmisión en vivo, el nuevo usuario sólo puede ver el contenido a partir del momento en que se unió. Por otro lado, en transmisiones bajo demanda este método no aplica porque cada usuario espera ver o escuchar el contenido a su gusto y conveniencia, por lo tanto un mismo paquete

de datos se debe enviar en instantes diferentes a cada nueva computadora que lo requiera.

CAPÍTULO 3

3. ANÁLISIS DEL SISTEMA

3.1. Requerimientos funcionales

El sistema computarizado de comunicación y control de ingreso a oficinas está constituido por tres partes: un componente de administración, uno de comunicación, y una aplicación que interactúa con los usuarios internos.

El componente de administración o *sistema administrador* cumple con los siguientes requerimientos funcionales:

- Creación de los elementos del sistema tales como máquinas, usuarios, porteros, horarios, categorías, y estados.
- Modificación de los datos de los elementos del sistema.
- Consulta de los datos de los elementos del sistema.
- Eliminación de los elementos del sistema.

- Presentación e impresión de reportes basados en la bitácora. En ésta se registran las transacciones que se realizan, tales como la comunicación entre un usuario interno y uno externo, la apertura de las puertas, y el envío de mensajes por parte de los usuarios externos.
- Establecer parámetros de la configuración del sistema como el espacio asignado a los usuarios para el almacenamiento de los mensajes que reciben, el tiempo de duración de los mensajes de audio y vídeo, la dirección de correo electrónico del administrador, la dirección IP del servidor que realizará el envío de correo electrónico en el sistema, etc.

El componente de comunicación o *sistema portero* cumple con los siguientes requerimientos funcionales:

- Permitir el establecimiento de una comunicación en tiempo real entre un usuario externo y uno interno, siempre y cuando éste disponga de una oficina y se encuentre disponible. La comunicación se establecerá de acuerdo a lo que el usuario interno haya especificado en su configuración y puede estar basada en mensajes de texto, audio o vídeo.
- Crear y enviar mensajes de texto, audio o vídeo, que van dirigidos a los usuarios internos que no se encuentren disponibles o no posean una oficina. El tipo de mensaje que

recibe el usuario dependerá de lo que éste haya indicado en su configuración.

- Permitir o denegar el acceso al área de oficinas a los usuarios internos, por medio de la verificación de su clave de acceso; si ésta es correcta el sistema portero abrirá la puerta, y tomará una foto instantánea de la persona que ingresa, para registrarla en la bitácora.
- Apagarse de manera automática cuando su horario de actividad haya finalizado.
- Registrar las transacciones de "Abrir Puerta", "Comunicación" y "Guardar Mensaje" en la bitácora.

La aplicación que interactúa con los usuarios internos o *sistema usuario* cumple con los siguientes requerimientos funcionales:

- Establecer la configuración de los datos personales del usuario tales como correo electrónico, cargo, teléfono, y foto; la contraseña para el ingreso a la aplicación; y los datos de configuración del sistema como el tipo de comunicación (texto, vídeo, o audio), tipo de mensajes que desea recibir de los usuarios externos (texto, vídeo, o audio), y un comentario que se presentará junto con sus datos en el sistema portero. Esta funcionalidad también está disponible en el sistema usuario Web.

- Presentar los mensajes nuevos que el usuario interno ha recibido para que puedan ser descargados y revisados. Esta funcionalidad también está disponible en el sistema usuario Web.
- Permitir la visualización del vídeo que esté siendo capturado por los porteros a los que el usuario se encuentre asociado.
- Aceptar o denegar la solicitud de establecer una comunicación de un usuario externo desde un sistema portero.
- Permitir el ingreso al área de oficinas de un usuario externo con el que el usuario interno se haya comunicado previamente por medio del sistema portero.
- Permitir cambiar el estado de conexión del usuario interno.

3.2. Requerimientos Técnicos

El sistema debe cumplir con los siguientes requerimientos técnicos:

- Controlar la asignación de espacio en disco para los usuarios con el fin de llevar un control más eficiente del crecimiento de los datos en el dispositivo de almacenamiento.
- Funcionar independientemente del dispositivo de captura de vídeo que vaya a utilizarse.
- Brindar flexibilidad en el uso de compresores de audio y vídeo.
- Funcionar dentro de un esquema distribuido.

3.3. Análisis de la Interacción Hombre-Máquina

Los tres componentes que conforman el sistema, deben permitir que los usuarios puedan llevar a cabo sus tareas de manera ágil y eficiente. Para esto, deben cumplir con los principios de usabilidad definidos en la Interacción Hombre-Máquina, tales como consistencia, familiaridad, flexibilidad, visibilidad, tiempo de respuesta y retroalimentación. A continuación se describen estos principios para cada uno de los componentes del sistema.

- *Consistencia*.- Tanto la presentación como la distribución de las diferentes funciones en cada uno de los componentes deben ser consistentes en todo el sistema. Esto permite que los usuarios puedan aprender rápidamente a realizar las tareas más comunes.

- *Familiaridad*.- Aplicar el principio de familiaridad permite aprovechar la experiencia que los usuarios tengan con sistemas similares, para llevar a cabo las tareas de manera más eficiente. Pensando en esto, el sistema administrador, deberá presentar una interfaz y una interacción que se apeguen a los estándares establecidos por las aplicaciones más comúnmente usadas, en cuanto a presentación de menús, distribución de botones, barras de tareas, etc. El

sistema usuario, deberá asemejarse en algunas de las funcionalidades que ofrece, a los diversos sistemas de mensajería instantánea, para que la interacción con la aplicación sea más intuitiva; en cuanto a la parte Web de este componente, cualquier persona que esté familiarizada con Internet y las aplicaciones desarrolladas para el Web, deberá poder interactuar satisfactoriamente con él. Por último, el sistema portero, deberá utilizar para la grabación de mensajes de audio o vídeo, funciones similares a las que presenta cualquier equipo diseñado para grabar y reproducir este tipo de contenido, como videograbadoras, equipos de sonido, etc.; tomando en consideración que este componente podría ser utilizado incluso por personas que no tienen experiencia en el uso de un computador.

- *Flexibilidad*.- El sistema debe permitir que los usuarios puedan acceder a las mismas funciones, de diferentes maneras. Este principio debe aplicarse especialmente cuando existen funciones que se realizan con mucha frecuencia, como es el caso de las tareas que se llevan a cabo en el sistema administrador, donde existen diversos elementos (porteros, usuarios, máquinas, etc.) sobre los que

se llevan a cabo las mismas operaciones (crear, consultar, modificar, y eliminar).

- *Visibilidad*.- Este principio hace referencia a la forma correcta en que deben estar dispuestos los elementos y las diferentes opciones que componen una interfaz, con el fin de facilitar a los usuarios el acceso a las funciones. La aplicación del principio de visibilidad tiene especial importancia en el caso del sistema portero, en el que deberán tomarse en cuenta factores como el tamaño, color y ubicación de los elementos de la interfaz, y la cantidad de información que se presenta al usuario. Esto se debe a que este componente del sistema puede ser utilizado por usuarios que no se encuentran familiarizados con el uso de un computador.

- *Tiempo de respuesta y retroalimentación*.- El sistema debe proveer una respuesta al usuario dentro de un periodo aceptable y utilizar indicadores que permitan determinar cuando un requerimiento está siendo procesado o en espera de ser atendido. Esto último podrá observarse en el caso de las solicitudes de comunicación desde un sistema portero. Una vez hecha la solicitud, el usuario externo deberá esperar

una respuesta por parte del usuario interno, en un lapso determinado. Durante este tiempo el sistema deberá proveer información, que permita al usuario conocer en qué estado se encuentra su solicitud.

3.4. Análisis de la Viabilidad

El sistema es viable de acuerdo a los siguientes factores:

Escalabilidad

El sistema está diseñado para funcionar en entornos organizacionales de cualquier tamaño, y soportar futuros crecimientos tanto en la cantidad de usuarios internos, como de computadores que ejecuten el sistema usuario o portero. Adicionalmente, el ingreso de nuevos elementos del sistema (usuarios, porteros, máquinas, horarios, etc.) es un proceso que puede llevarse a cabo con facilidad.

Adaptabilidad

La adaptabilidad del sistema permite que éste pueda funcionar con un gran número de tecnologías existentes y soportar cambios a tecnologías posteriores. Por ejemplo, el sistema funciona con cualquier dispositivo de captura de vídeo, base de datos, y compresor de audio y vídeo. Adicionalmente, el sistema brinda la

posibilidad de especificar el tipo de canal de comunicación que se ha de utilizar entre los sistemas cliente y el servidor, con el objetivo de facilitar, en caso de que se requiera, la posterior implementación de las aplicaciones cliente del sistema en un entorno multiplataforma.

Flexibilidad

El sistema es flexible porque permite configurar un gran número de parámetros tales como el espacio en disco que se asigna a cada usuario en el sistema, diferenciando entre los que tienen una oficina y los que no; la duración de los mensajes de audio y vídeo que se envían a los usuarios; el tipo de comunicación que se desea establecer (texto, audio, o vídeo), entre otros.

Utilidad

Las funciones que lleva a cabo el sistema, en lo que respecta a seguridad y comunicación, lo hacen muy conveniente para implementarse en entornos de trabajo en los que se necesita controlar el ingreso de personas a las oficinas, y mantener al personal informado acerca de quiénes los han estado solicitando, principalmente a aquellos que no disponen de una oficina o que se encuentran ausentes la mayor parte del tiempo.

3.5. Modelos de Análisis

Los modelos y diagramas de análisis que se presentan a continuación permiten obtener una visión más clara del funcionamiento del Sistema Computarizado de Comunicación y Control de Ingreso a Oficinas.

3.5.1. Diagrama general del sistema

El diagrama general del sistema está compuesto por tres niveles: un nivel de presentación, uno de procesamiento, y otro de almacenamiento de datos.

En el nivel de presentación se encuentran los cuatro componentes que permiten llevar a cabo la administración de datos, el establecimiento de comunicaciones, y la interacción con los usuarios internos; estos son, respectivamente, el sistema administrador, el sistema portero, el sistema usuario y el sistema usuario Web. Los requerimientos funcionales exigen que el sistema portero esté equipado con dos dispositivos de captura de vídeo (uno que enfoque el área de la puerta y otro que capte la vista frontal del usuario externo) y uno de audio, para la comunicación en tiempo real y la grabación de mensajes. El sistema usuario, por otro lado, puede o no estar equipado con dispositivos de

audio/vídeo, y de esto dependerá el tipo de comunicación que se pueda establecer entre el usuario interno y el usuario externo.

En el nivel de procesamiento, por otro lado, se encuentra el servidor de aplicaciones que se encarga de procesar los requerimientos de los sistemas que componen el nivel de presentación.

Por último, en el nivel de almacenamiento, se encuentra la base de datos, que guarda la información permanente de los elementos del sistema, y los directorios que contienen los archivos que se utilizan. El servidor se comunica con este nivel para procesar los requerimientos de las diferentes aplicaciones.

En la figura 3.1, que se muestra a continuación, se presenta el diagrama general del sistema, con sus respectivos componentes.

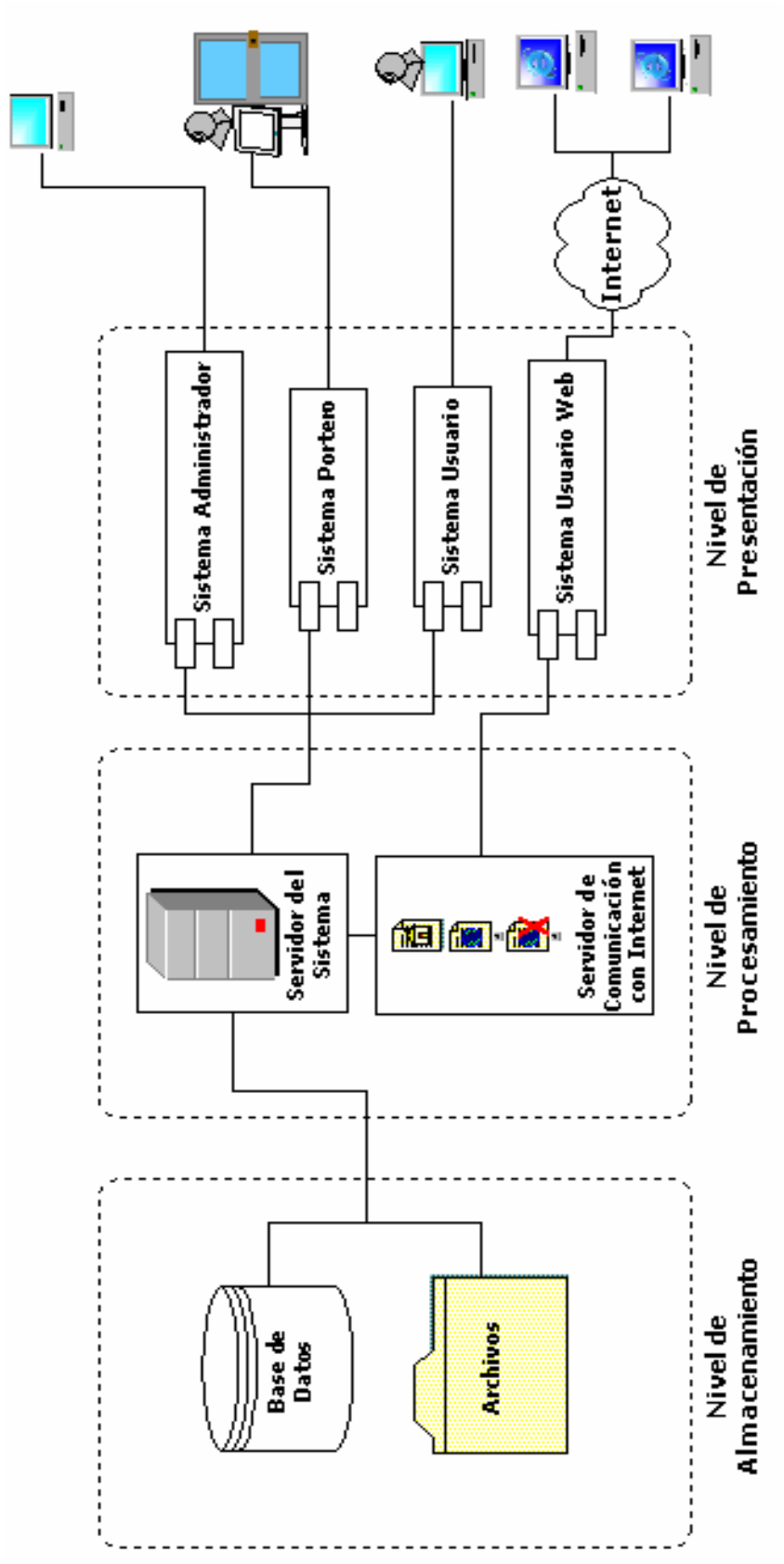


Figura 3.1.- Diagrama General del Sistema

3.5.2. Modelo conceptual

Las entidades que conforman el sistema se describen a continuación:

- *usuario*.- Entidad que representa a un usuario interno del sistema, en términos de sus datos personales (que sirven para identificarlo) y sus datos de configuración (que le permiten interactuar con los sistemas usuario y portero).
- *portero*.- Entidad representativa de una computadora que tenga instalado y ejecute el sistema portero. Sus campos describen los datos que identifican a un portero, y que le permiten funcionar a éste dentro del sistema.
- *mensaje*.- Entidad que representa a un mensaje de audio o vídeo, creado y enviado desde el sistema portero, y que va dirigido a alguno de los usuarios internos.
- *maquina*.- Entidad representativa de una computadora que tenga instalado y ejecute el sistema usuario. Sus campos identifican a una computadora autorizada para funcionar dentro del sistema.
- *categoria*.- Entidad que representa a una clasificación que permite catalogar y organizar de mejor manera a los usuarios internos. Por ejemplo, en una institución educativa, podrían utilizarse categorías asociadas con

las diferentes áreas académicas, como "Área de Informática", "Área de Ciencias Sociales", etc.

- *horario*.- Entidad que engloba un grupo de días, que permite a los usuarios y/o porteros registrados acceder al sistema dentro de un rango de fechas y horas determinado.
- *dias*.- Entidad que representa a un día de la semana con una hora de inicio y de fin, asociada a un horario determinado.
- *configuracion_sistema*.- Entidad que contiene toda la información general necesaria para poner en funcionamiento el sistema.
- *log*.- Entidad que permite mantener un historial de las transacciones que se llevan a cabo en el sistema ("Abrir Puerta", "Guardar Mensaje" y "Comunicación").
- *estado*.- Entidad que representa a un estado de conexión de un usuario interno.

Las relaciones establecidas entre las entidades del sistema son:

- *usuario-categoria*.- Relación con cardinalidad de muchos a muchos, que asocia a los usuarios con las categorías.

- *usuario-horario*.- Relación de muchos a uno, que asocia a un usuario con alguno de los horarios registrados.
- *usuario-mensaje*.- Relación de uno a muchos, que establece la asociación entre un usuario y los mensajes que recibe.
- *usuario-maquina*.- Relación de muchos a muchos, que representa la asociación entre los usuarios internos, con oficina, y las computadoras registradas que le han sido asignadas.
- *usuario-portero*.- Relación de muchos a muchos, que establece la asociación entre los usuarios internos, sin oficina, y los porteros registrados en el sistema.
- *horario-dias*.- Relación de uno a muchos, que permite asociar días a los horarios existentes.
- *portero-maquina*.- Relación de muchos a muchos, que representa la asociación entre los porteros y las computadoras pertenecientes a los usuarios internos que tienen oficina.
- *portero-horario*.- Relación de uno a muchos, que asocia a un portero con alguno de los horarios registrados.

Las entidades y las relaciones que existen entre ellas se muestran en la figura 3.2:

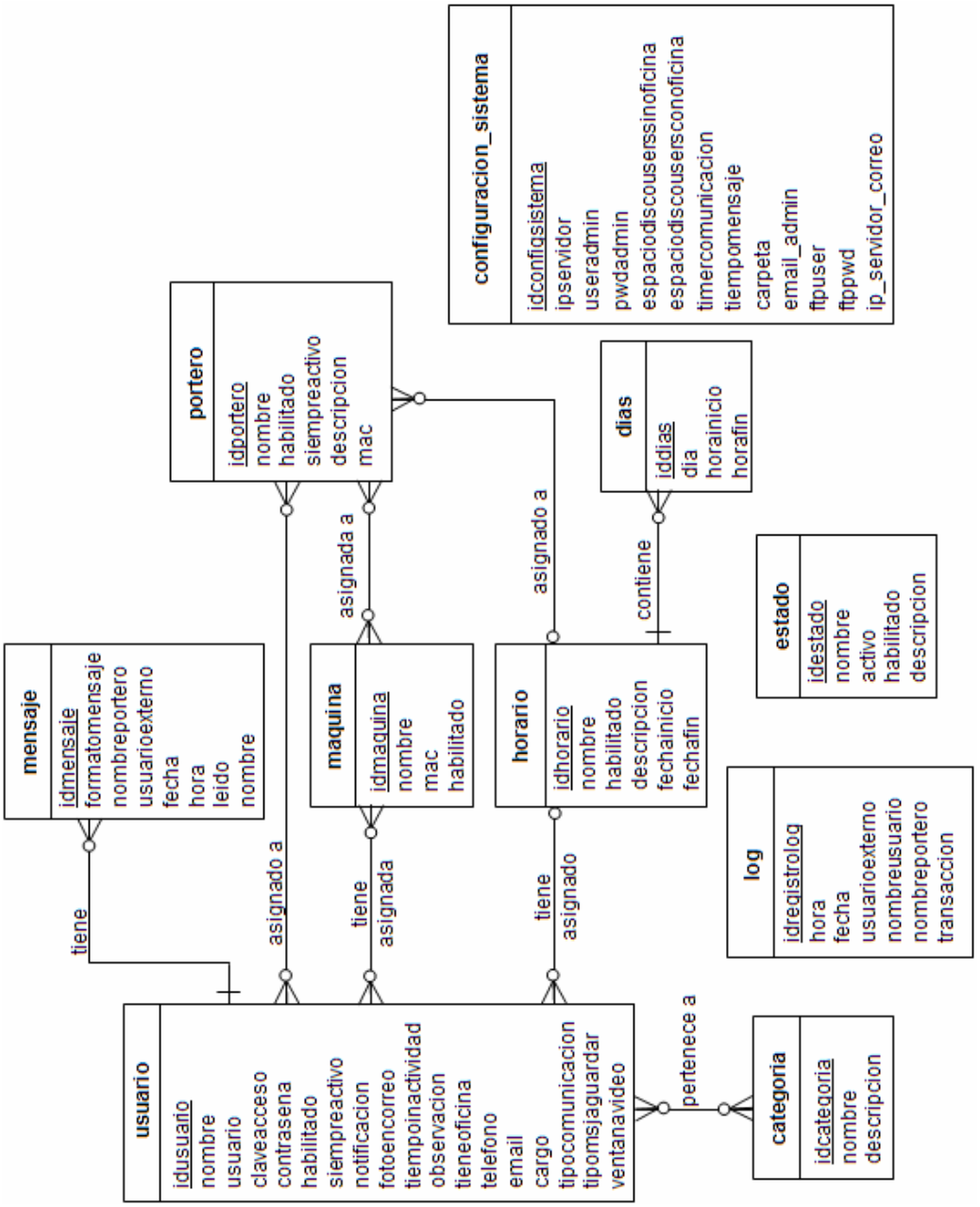


Figura 3.2.- Modelo Conceptual del Sistema

3.5.3. Casos de uso y escenarios

A continuación se presentarán los casos de uso y escenarios más representativos de cada componente del sistema.

Sistema Administrador

Caso de Uso: Administración de usuarios internos.

Actores: Administrador; persona que se encarga de la creación, modificación, eliminación y configuración de los elementos que constituyen el sistema (administrador).

Escenario 1: Creación exitosa de un nuevo usuario interno.

Supuestos:

- Se ingresan los datos requeridos para la creación de un nuevo usuario interno (nombre, e-mail, usuario, contraseña, clave de acceso, etc.)
- Las categorías a las que pertenece el usuario, el horario, y las máquinas o porteros que han de asignársele, han sido registrados con anterioridad.
- El nombre, el usuario y la clave de acceso ingresados son únicos en el sistema.

Salidas:

- El nuevo usuario interno es registrado exitosamente.
- Se crea la carpeta de mensajes recibidos para el usuario.

Escenario 2: Eliminación exitosa de un usuario interno.

Supuestos:

- El administrador selecciona el usuario interno que desea eliminar.
- El usuario seleccionado no está conectado.

Salidas:

- El usuario es eliminado del sistema.
- La carpeta de mensajes recibidos del usuario es eliminada.

Escenario 3: Modificación exitosa de un usuario interno.

Supuestos:

- El administrador selecciona el usuario interno cuyos datos desea modificar, e ingresa los nuevos datos válidos correspondientes.
- El usuario seleccionado, no se encuentra conectado.

Salidas:

- Los datos modificados del usuario son almacenados.

Caso de Uso: Configuración del sistema por el administrador.

Actores: Persona que se encarga de la creación, modificación, eliminación y configuración de los elementos que constituyen el sistema (administrador).

Escenario 1: Configuración exitosa del sistema.

Supuestos:

- Se ingresan todos los datos requeridos para la configuración del sistema. (dirección IP del servidor de correo, usuario del administrador, y contraseña, etc.).

Salidas:

- Se almacena la nueva configuración del sistema.

Sistema Usuario

Caso de Uso: Comunicación usuario interno-servidor

Actores: Persona que solicita establecer comunicación con el servidor desde su máquina (usuario interno).

Escenario 1: Acceso exitoso del usuario interno al sistema.

Supuestos:

- El usuario interno ha ingresado su usuario y contraseña.
- El usuario y contraseña son válidos, y la máquina desde la que se conecta corresponde a alguna de las asignadas al usuario interno.
- Tanto el usuario como la máquina se encuentran habilitados.
- El usuario ingresa dentro del horario de actividad establecido.

Salidas:

- El sistema concede acceso al usuario interno.
- Se despliega en la ventana principal la lista de mensajes recibidos.
- Se actualiza el estado de conexión del usuario interno.

- Se envía una notificación al usuario interno conectado desde la página Web, indicándole que se acaba de conectar desde la aplicación instalada en su computadora.

Escenario 2: Revisión de los mensajes recibidos.

Supuestos:

- El usuario interno tiene uno o más mensajes recibidos.
- El usuario interno selecciona el mensaje nuevo que desea revisar, y presiona el botón "Descargar".

Salidas:

- El mensaje es descargado en el directorio local de mensajes del usuario.
- El mensaje es descartado de la lista de mensajes nuevos, y pasa a formar parte de la lista de mensajes almacenados.
- Se elimina el mensaje del servidor.

Nota.- Cuando se descarga un mensaje de audio o vídeo, se crearán dos archivos; el del contenido multimedia, y uno con extensión .txt, que indique datos como fecha, hora, portero desde el que se envío, etc.

Escenario 3: Configuración del sistema por el usuario interno.

Supuestos:

- El usuario interno ingresa todos los datos personales requeridos.
- En los datos de la aplicación, el usuario interno especifica el tipo de notificación que desea utilizar (con o sin sonido); si desea recibir una foto adjunta con los mensajes escritos; ingresa el tiempo de inactividad, la observación, en caso de que utilice alguna; el tipo de mensaje que desea usar en la comunicación y en la recepción de mensajes enviados.
- Ingresa la contraseña con la que va a acceder al sistema.

Salidas:

- Los datos son configurados correctamente.

Sistema Portero

Caso de Uso: Abrir la puerta usando el portero

Actores: Persona que ingresa su clave al solicitar acceso desde el sistema portero (usuario interno).

Escenario 1: Un usuario interno solicita acceso, utilizando el Sistema Portero, y se le concede.

Supuestos:

- La clave de acceso ingresada existe.
- El usuario está habilitado e intenta ingresar dentro del horario permitido.

Salidas:

- Se muestra un mensaje de éxito.
- Se abre la puerta para permitir el acceso.
- Se almacena un nuevo registro en la bitácora.

Caso de Uso: Comunicación usuario externo-usuario interno por medio del Sistema Portero.

Actores: Persona que solicita establecer comunicación desde el portero (usuario externo); persona a la que va dirigida la solicitud de comunicación (usuario interno).

Escenario 1: Solicitud exitosa de comunicación con el usuario interno.

Supuestos:

- El usuario interno solicitado se encuentra disponible.

- Se notificó al usuario que alguien lo solicita, utilizando un mensaje de tipo pop-up.
- El usuario acepta la solicitud de comunicación.
- Se identifica el tipo de comunicación que el usuario tiene configurado (texto, audio, o vídeo).

Salidas:

- Se establece la comunicación entre el usuario externo y el usuario interno.
- Se añade el registro correspondiente a la bitácora.

Escenario 2: El usuario externo deja un mensaje de audio o vídeo al usuario interno.

Supuestos:

- El usuario interno seleccionado no se encuentra disponible.
- Se identifica cuál es el tipo de mensajes almacenados que el usuario interno especificó en su configuración.
- El usuario externo escoge la opción de dejar un mensaje de audio o vídeo al usuario interno.
- Hay espacio suficiente en el servidor para almacenar el mensaje

Salidas:

- El mensaje es almacenado.
- Se envía una notificación vía e-mail al usuario interno.
- Se añade el registro correspondiente a la bitácora.

Escenario 3: El usuario externo deja un mensaje de texto al usuario interno.

Supuestos:

- El usuario interno seleccionado no se encuentra disponible.
- Se identifica cuál es el tipo de mensajes almacenados que el usuario interno especificó en su configuración.
- El usuario externo escoge la opción de dejar un mensaje de texto al usuario interno.

Salidas:

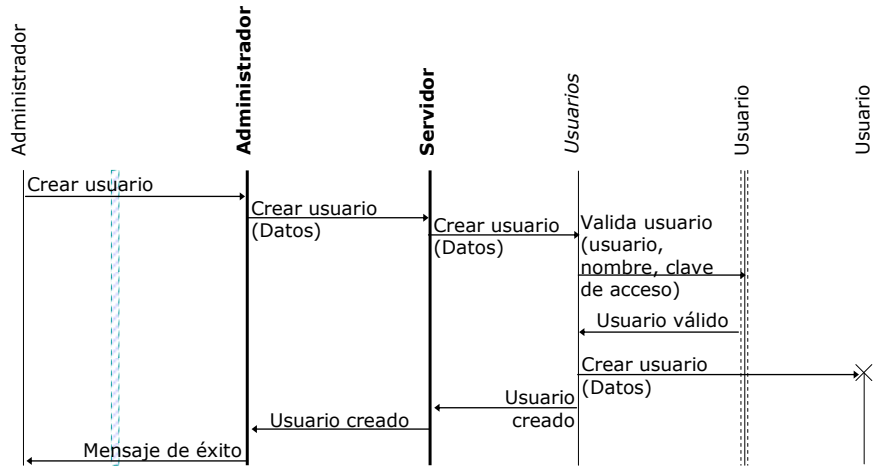
- Se envía un e-mail al usuario interno con el mensaje del usuario externo.
- Se añade el registro correspondiente a la bitácora.

3.5.4. Diagramas de Interacción de Objetos

A continuación se presentarán los diagramas de interacción de objetos correspondientes a los casos de uso y escenarios anteriormente mencionados.

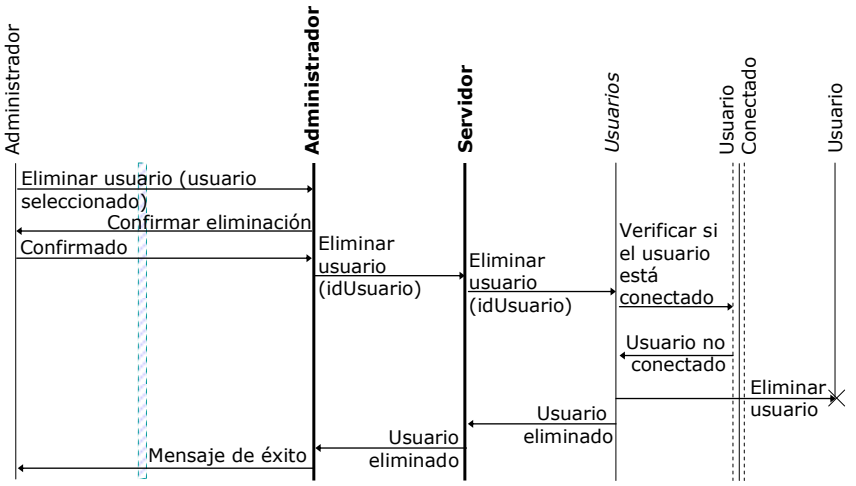
Sistema Administrador

Caso de uso:	Administración de usuarios internos
Escenario:	Creación exitosa de un nuevo usuario interno



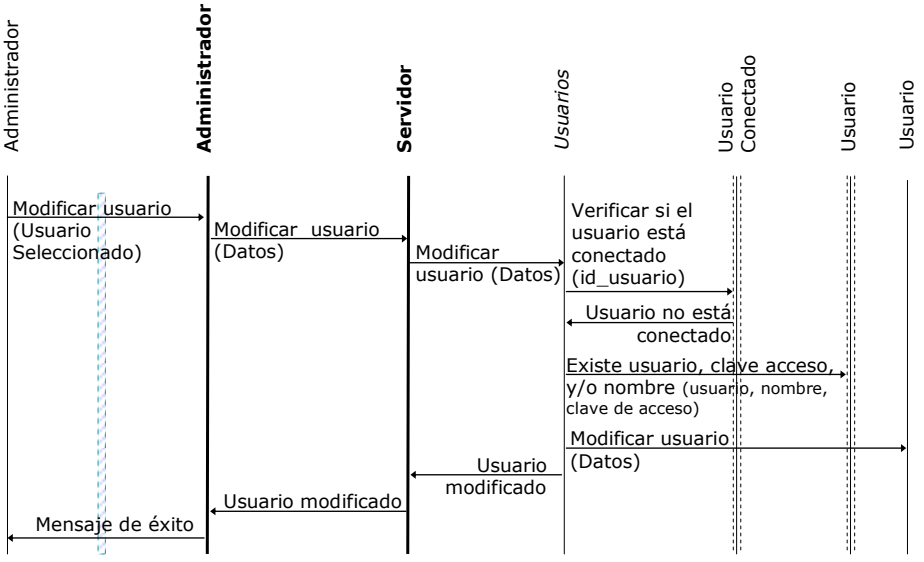
Datos: Nombre, cargo, teléfonos, e-mail, usuario, contraseña, clave de acceso, lista de categorías asociadas, lista de máquinas o porteros relacionados, habilitado

Caso de uso:	Administración de usuarios internos
Escenario:	Eliminación exitosa de un usuario interno



Si el usuario está conectado, no se lo elimina.

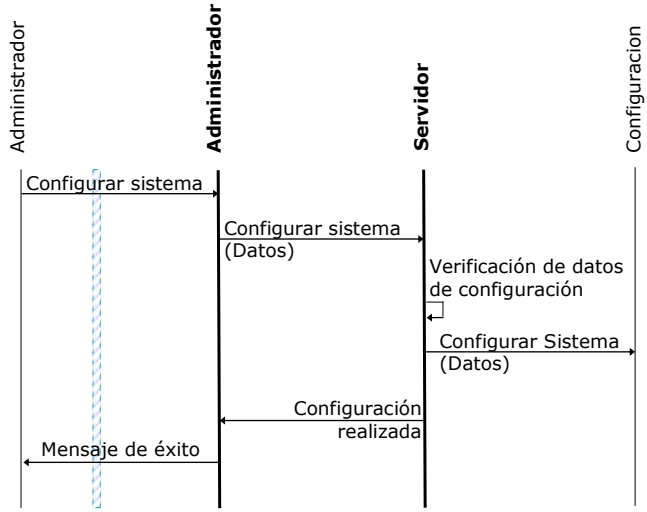
Caso de uso:	Administración de usuarios internos
Escenario:	Modificación exitosa de un usuario interno



Datos: Nombre, cargo, teléfonos, e-mail, usuario, contraseña, lista de categorías asociadas, lista de máquinas o porteros asociados, clave de acceso, habilitado.

Si el usuario está conectado, no se modifica.

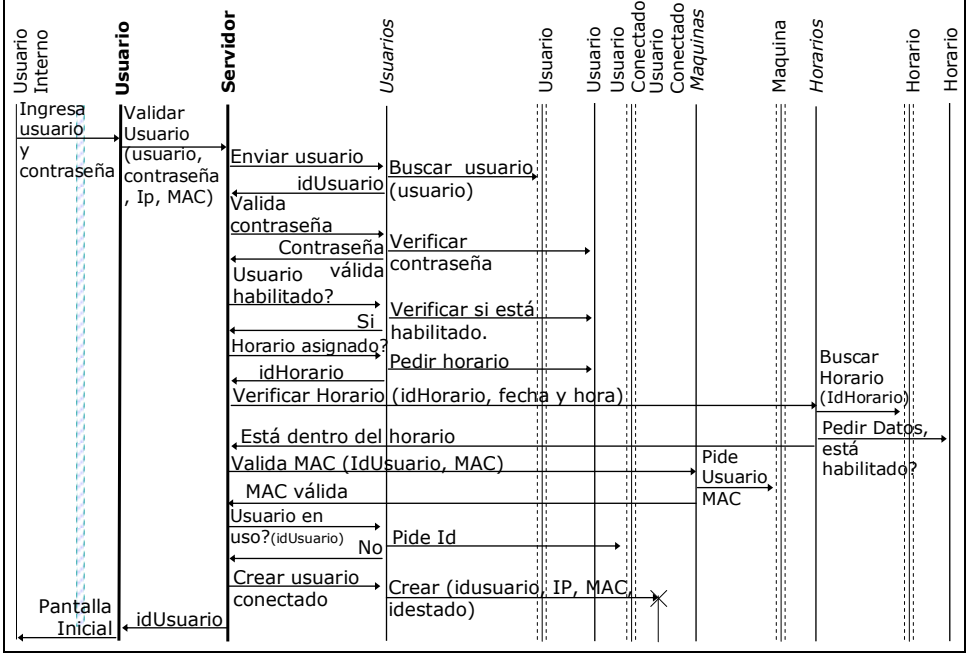
Caso de uso:	Configuración del sistema por el administrador
Escenario:	Configuración exitosa del sistema



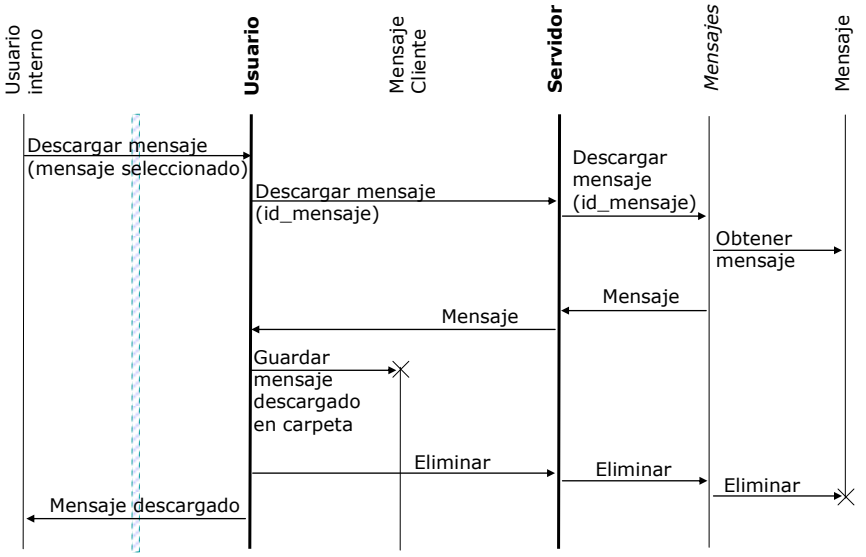
Datos: IP servidor de correo, tiempo de espera antes de cancelar una comunicación, usuario y contraseña del administrador, e-mail del administrador, espacio disco usuario con oficina, espacio disco usuario sin oficina, nombre de la carpeta que guardará los mensajes para los usuarios, tiempo de duración de un mensaje, usuario y contraseña para cuenta FTP

Sistema Usuario

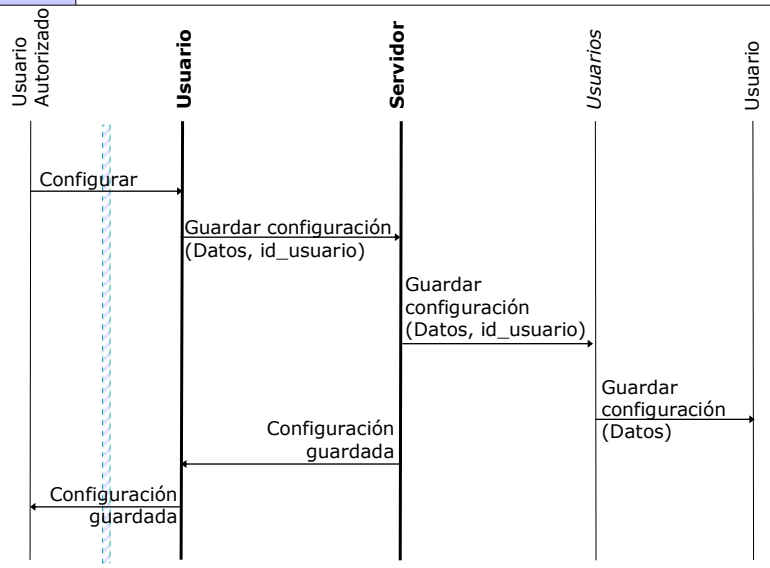
Caso de uso:	Comunicación usuario interno - servidor
Escenario:	Acceso exitoso del usuario interno al sistema



Caso de uso:	Comunicación usuario interno - servidor
Escenario:	Revisión de los mensajes recibidos



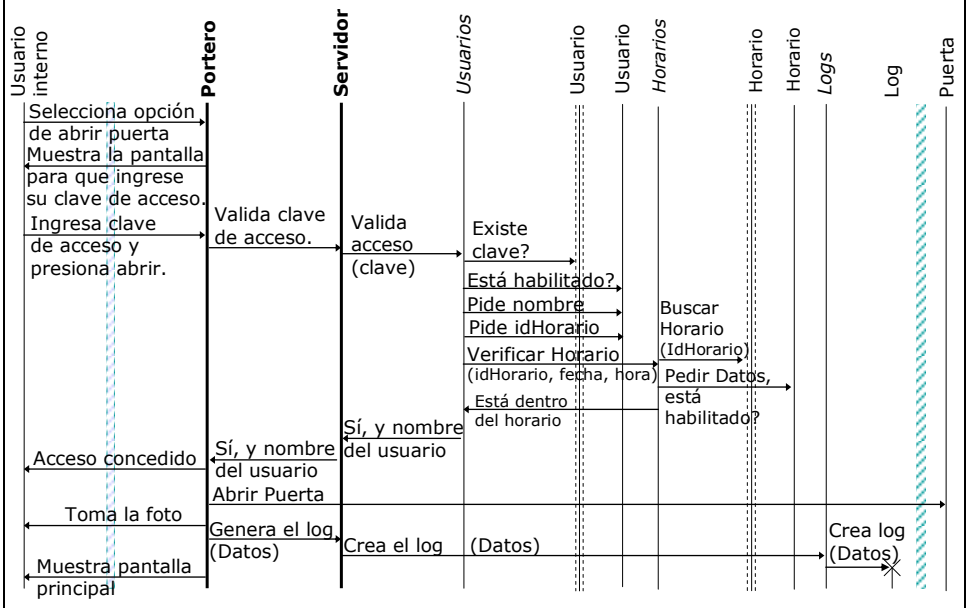
Caso de uso:	Comunicación usuario interno - servidor
Escenario:	Configuración del sistema por el usuario interno



Datos: Nombre, cargo, teléfonos, e-mail, foto, contraseña, notificación, foto en Correo, tiempo de inactividad, observación, tipo mensaje para comunicación, tipo mensaje a guardar por el portero, ver usuario externo en comunicaciones de texto y audio.

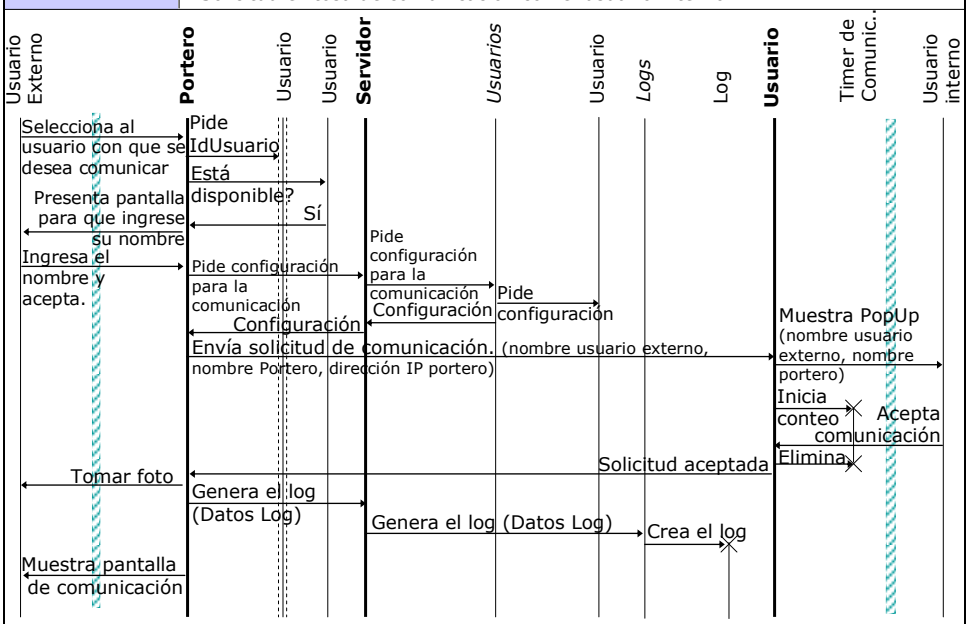
Sistema Portero

Caso de uso:	Abrir la puerta usando el portero
Escenario:	Un usuario interno solicita acceso, utilizando el Sistema Portero, y se le concede



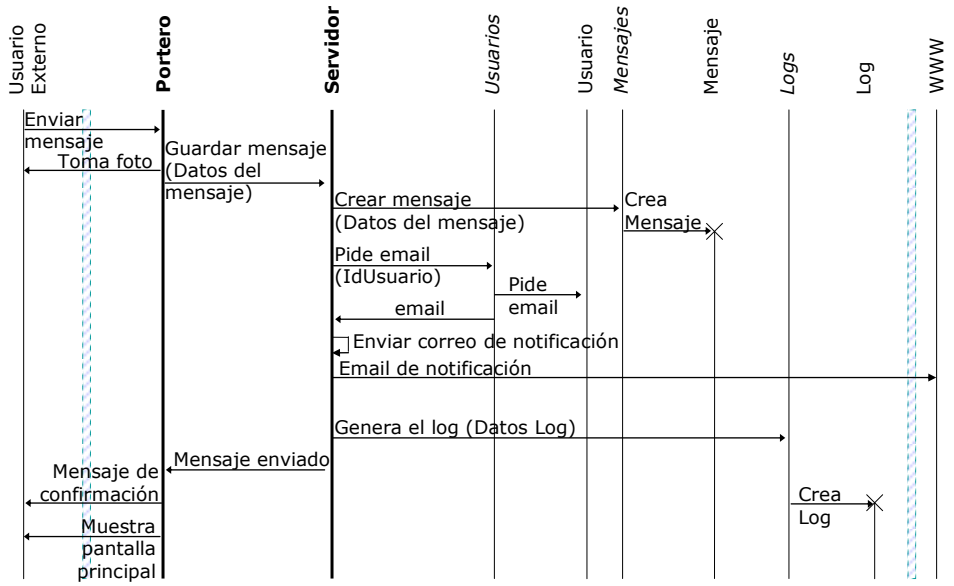
Datos: foto, transaccion, fecha, hora, nombre del usuario, nombre del Portero

Caso de uso:	Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	Solicitud exitosa de comunicación con el usuario interno.



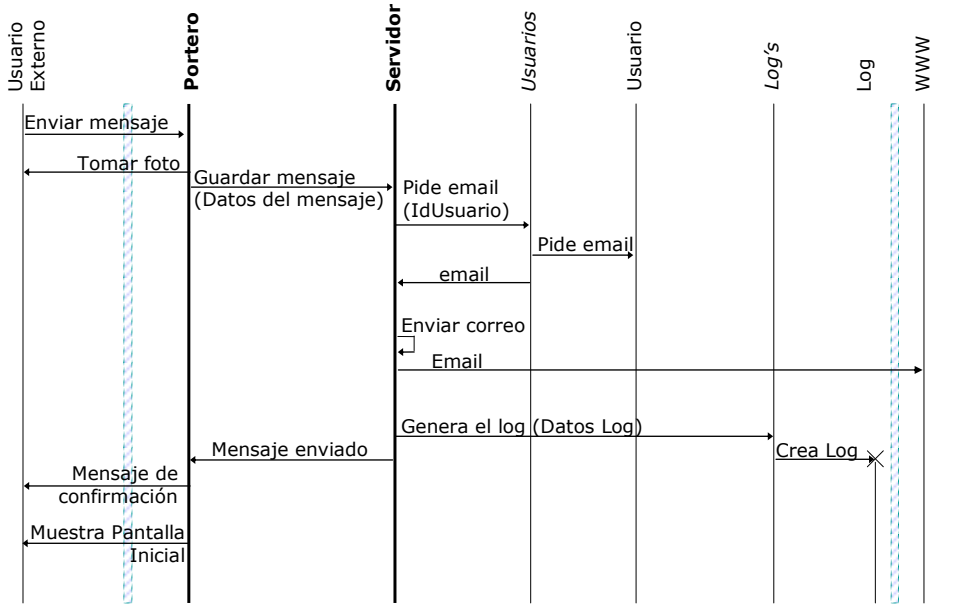
Datos Log: foto, tipo de transacción, fecha, hora, nombre del usuario, nombre del Portero, nombre del usuario externo

Caso de uso:	Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	El usuario externo deja un mensaje de vídeo o de audio al usuario interno



Datos Log: foto, transacción, fecha, hora, nombre del usuario, nombre del portero, usuario externo
Datos del mensaje: mensaje, idUsuario, tipo de mensaje, idPortero, nombre usuario externo, foto

Caso de uso:	Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	El usuario externo deja un mensaje de texto al usuario interno



Datos Log: foto, transacción, fecha, hora, nombre del usuario, nombre del portero, usuario externo
Datos del mensaje: mensaje, idUsuario, tipo de mensaje, idPortero, usuario externo, foto

3.6. Análisis de las aplicaciones y herramientas de desarrollo

3.6.1. Plataforma

Los sistemas operativos que son más utilizados en la actualidad, en lo que se refiere a computadores de escritorio, son Windows, Linux, y Mac OS, en sus diferentes versiones.

Para la aplicación cliente, que interactúa con los usuarios internos, y para el sistema portero, se seleccionó la plataforma Windows. Esto se debe al grado de investigación y de pruebas que se han realizado en esta plataforma, en cuanto a la manipulación de contenido multimedia; a que la mayoría de los usuarios utilizan este sistema operativo; y a que es la plataforma que mejor conocemos.

En el caso del servidor de aplicaciones y servidor Web, también se optó por utilizar la plataforma de Windows, en cualquier versión que trabaje con IIS (Internet Information Services), con el objetivo de proveer mayor naturalidad en la comunicación entre las aplicaciones cliente y el servidor.

Adicionalmente, la selección de las plataformas para las distintas aplicaciones que componen el sistema, se justifica

también por la herramienta de desarrollo utilizada, como se explicará en el siguiente punto.

3.6.2. Herramientas de desarrollo

Los principales atributos que se han considerado al seleccionar el lenguaje de programación, que se utilizó para desarrollar esta aplicación son: claridad, sencillez y unidad; naturalidad para la aplicación; facilidad para verificar programas; y costo de uso. Estas características permitieron que se minimice el tiempo invertido tanto en la creación como en la depuración y prueba del sistema.

Entre las herramientas más comúnmente utilizadas para desarrollar un proyecto de este tipo, y con las que hemos tenido mayor experiencia en la programación, se encuentran básicamente Java y los componentes de Visual Studio.

Java provee una librería que permite manipular contenido multimedia, conocida como JMF (Java Media Framework); inicialmente, se trataba de código cerrado, no disponible gratuitamente, que hace poco tiempo fue liberado. Ésta provee un marco de trabajo, en el que es necesario implementar toda la funcionalidad que se requiera en el caso

de la obtención de vídeo a partir de dispositivos de captura, y del envío de contenido multimedia en tiempo real a través de la red. Los codecs con los que trabaja son abiertos lo cual es una desventaja en la calidad del resultado al compararlos con los propietarios; esto es crítico en el caso de la videoconferencia, que requiere velocidades de compresión/descompresión muy altas que no afecten la calidad del resultado. Java se caracteriza por ser un lenguaje de programación portable, sin embargo, en este contexto su portabilidad es cuestionable ya que es necesario conseguir los drivers de los dispositivos en cada plataforma, para que pueda funcionar en todos los sistemas operativos. Por otro lado, el acceso a los dispositivos de hardware en Java es más lento en comparación con los lenguajes de plataformas nativas, y esto se debe a que utiliza una máquina virtual como paso intermedio.

En cuanto a Visual Studio, se consideraron las dos versiones más actuales que existían en el momento en que surgió la idea de desarrollar este sistema; éstas eran Visual Studio 6.0 y Visual Studio .NET. Se seleccionó Visual Studio .NET, en vista de que la versión 6.0 presenta problemas en cuanto a la portabilidad al encontrarse muy ligada al entorno de

desarrollo, de modo que los resultados de la ejecución de un código pueden variar de una máquina a otra. Finalmente, dentro de Visual Studio .NET, las dos mejores opciones que se presentaban eran Visual C++, y Visual C#. Se eligió C# porque se apega más a los principios que se mencionaron anteriormente para la selección de un lenguaje de programación, ya que su sintaxis facilita la escritura, prueba, compresión y depuración de los programas. Adicionalmente, provee librerías que hacen más sencillo el desarrollo de sistemas cliente-servidor, la manipulación y la transferencia de vídeo y audio por red (parte de la funcionalidad del sistema se fundamenta en el uso de librerías basadas en DirectShow, para el manejo de contenido multimedia), y el establecimiento de comunicación con dispositivos de hardware. En este lenguaje el acceso a estos dispositivos es directo; se permite utilizar codecs propietarios como el Windows Media 9, que es idóneo para aplicaciones de vídeo-conferencia; y se puede acceder a las dll propias del sistema e importar librerías desarrolladas en algún otro lenguaje dentro del entorno .NET. La selección de este lenguaje de programación para el desarrollo, va de la mano con el sistema operativo de Windows, y por tanto las aplicaciones

que constituyen el sistema deberán ejecutarse sobre esta plataforma.

Por otro lado, en lo que respecta a la creación del sitio Web dinámico del sistema, los lenguajes de programación que se presentaron como mejores alternativas son ASP, PHP, y JSP. Se optó por utilizar ASP, debido a la afinidad que presenta con la plataforma seleccionada, y a su facilidad de implementación y mantenimiento. Este lenguaje usa IIS como servidor Web para alojar las páginas Web que componen la aplicación sistema usuario Web; éstas no contienen ninguna lógica de negocio en su código, sino que se comunican con el servidor de aplicaciones para llevar a cabo las tareas requeridas. Esto permite que el sistema funcione dentro de una arquitectura centralizada, lo que a su vez facilita la detección de errores, y brinda seguridad en la manipulación del código.

CAPÍTULO 4

4. DISEÑO DEL SISTEMA

4.1. Modelos de Diseño

Los modelos y diagramas de diseño que se presentan a continuación permiten obtener una idea más detallada del Sistema Computarizado de Comunicación y Control de Ingreso a Oficinas.

4.1.1. Diseño de la arquitectura del sistema

El sistema está compuesto por tres niveles o capas: nivel de almacenamiento de datos, nivel de procesamiento y nivel de presentación, tal como se muestra en la figura 4.1.

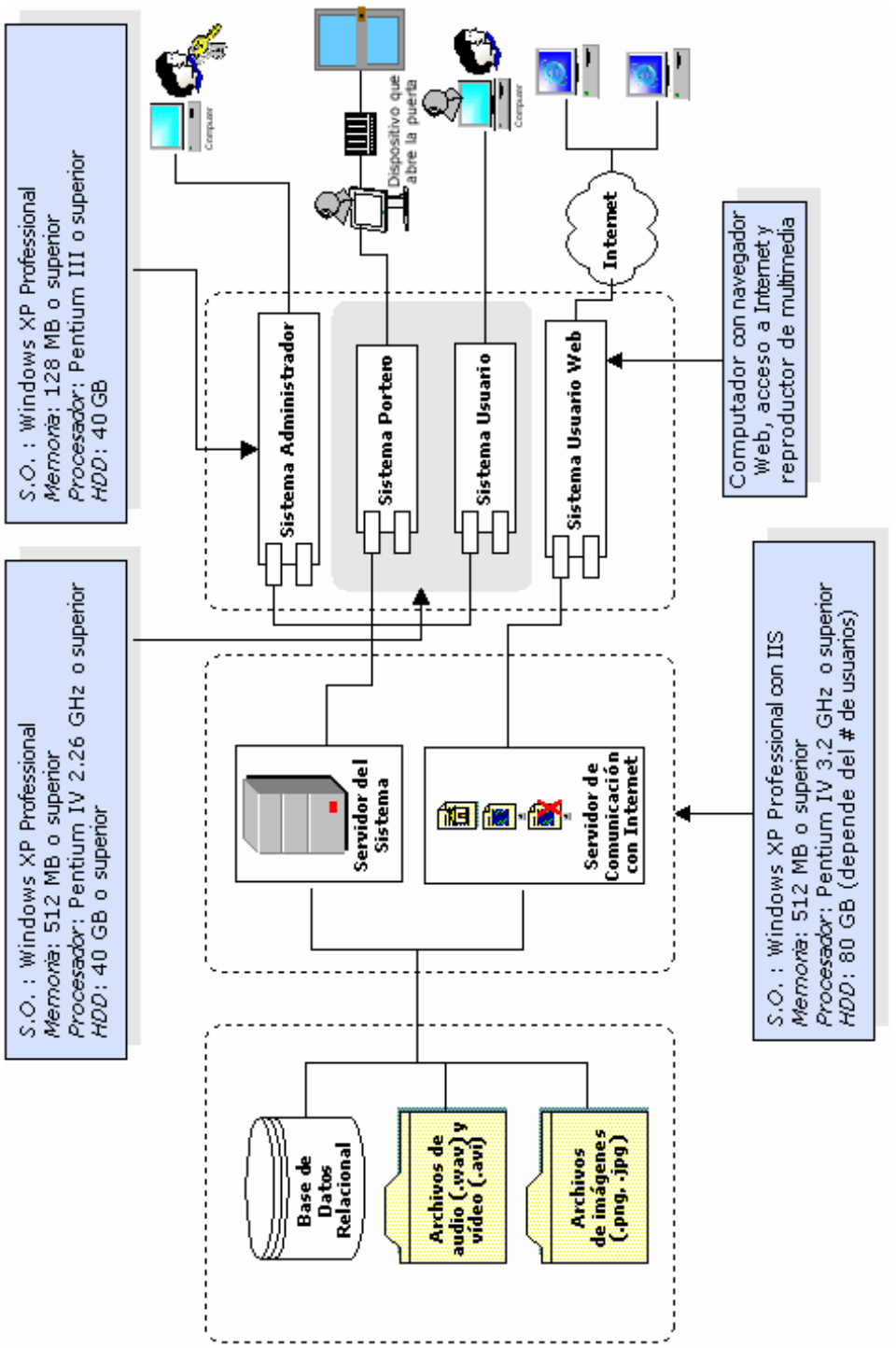


Figura 4.1.- Diseño detallado de la arquitectura del sistema

El nivel de almacenamiento se compone por una base de datos relacional; por los directorios de usuarios, en los que se almacenan archivos de audio (.wav) y de vídeo (.avi); y por los directorios donde se guardan las fotos de los usuarios, las imágenes que representan los estados de conexión y las fotos que son tomadas por el portero cada vez que se realiza una transacción ("Abrir Puerta", "Comunicación", y "Guardar Mensaje"). Los archivos de las imágenes de los usuarios y de los estados de conexión se guardan con el nombre correspondiente al ID con el que fueron registrados en la base de datos. Los nombres para los directorios de los usuarios se construyen a partir del contenido del campo "usuario", que se guarda en una de las tablas de la base.

El nivel de procesamiento está compuesto por un computador que ejecutará el servidor de aplicaciones y el servidor Web. Los requerimientos de hardware y software para que el servidor pueda desempeñarse de la mejor manera son:

- Memoria: 512 MB o superior.
- Procesador: Pentium IV 3.2 GHz o superior.
- Capacidad de disco duro: 80 GB.

- Sistema Operativo: Windows XP Professional
- Internet Information Services (IIS) con FrontPage Server Extensions (FPSE) y File Transfer Protocol (FTP) Service.
- .NET Framework 1.1

Adicionalmente, debe configurarse una cuenta con permisos de lectura y escritura sobre la carpeta que contendrá los mensajes destinados a los usuarios internos (esta carpeta estará dentro de la ruta C:\Inetpub\ftproot), para realizar las transferencias de archivos de audio y vídeo entre los sistemas usuario o portero, y el servidor. Además, deberá instalarse el componente FTP Service del IIS; y habilitar el servicio FTP Publishing del sistema operativo.

Por otro lado, la conexión con el nivel de almacenamiento se realiza por medio de ODBC, leyendo de un archivo la cadena de conexión específica, lo que permite, que el sistema pueda comunicarse y trabajar con cualquier motor de base de datos; también se deberá indicar el número máximo de registros que la tabla "Log" (bitácora) podrá almacenar sin que se envíe un mensaje de advertencia al administrador.

El nivel de presentación se compone del sistema administrador, del sistema usuario con su componente Web y del sistema portero. Los requerimientos de hardware y software para los tres sistemas se detallan a continuación.

Sistema Administrador

- Memoria: 128 MB o superior.
- Procesador: Pentium III o superior.
- Capacidad de disco duro: 40 GB o incluso menor (se requieren 10 MB para la instalación).
- Sistema Operativo: Windows XP
- .NET Framework 1.1

Sistema Usuario

- Memoria: 512 MB o superior.
- Procesador: Pentium IV 2.26 GHz o superior.
- Capacidad de disco duro: 40 GB.
- Sistema Operativo: Windows XP
- .NET Framework 1.1

Sistema Portero

- Memoria: 512 MB o superior.
- Procesador: Pentium IV 2.26 GHz o superior.
- Capacidad de disco duro: 40 GB.
- Sistema Operativo: Windows XP
- .NET Framework 1.1
- DirectX 8.1

Los requerimientos detallados para el Sistema Usuario y Sistema Portero son necesarios para la transmisión de contenido multimedia a través de la red. Sin embargo, si en la implementación del sistema no se utilizan estos dispositivos para las operaciones anteriormente mencionadas, los requerimientos de hardware pueden ser inferiores.

Sistema Usuario Web

- Computador con navegador Web, acceso a Internet y reproductor de multimedia.

En los tres sistemas, se debe especificar el parámetro correspondiente a la dirección IP del servidor. Adicionalmente, tanto en el sistema usuario como en el

sistema portero, se deben indicar los números de puerto que han de utilizarse para establecer comunicaciones instantáneas y enviar mensajes entre ellos. Finalmente, en el sistema portero deben especificarse los parámetros referentes a los dispositivos de captura de vídeo y audio con sus respectivos compresores.

4.1.2. Modelo lógico de la base de datos

El modelo lógico, mostrado en la figura 4.2, consta de cuatro tablas adicionales, que resultan de las relaciones especificadas en el modelo conceptual detallado en la sección 3.5.2.

- *usuario_categoria*.- Está formada por los campos idusuario e idcategoria, que permiten asociar a los usuarios con las distintas categorías existentes.
- *usuario_maquina*.- Los campos que contiene son idusuario e idmaquina, que constituyen la relación entre los usuarios internos, con oficina, y las computadoras.
- *portero_maquina*.- Los campos que la conforman son idportero e idmaquina, que relacionan a los porteros con las computadoras que les hayan sido asignadas. Los usuarios con oficina y los porteros se relacionan por medio de esta tabla y la anterior, del siguiente modo: a

un usuario se le asigna una máquina, relación que se registra en la tabla usuario_maquina, y luego esta máquina es asociada a un portero, ingresando un nuevo registro en la tabla portero_maquina.

- *usuario_portero*.- Está compuesta por los campos idusuario e idportero, que permiten relacionar a los usuarios, sin oficina, con los porteros.

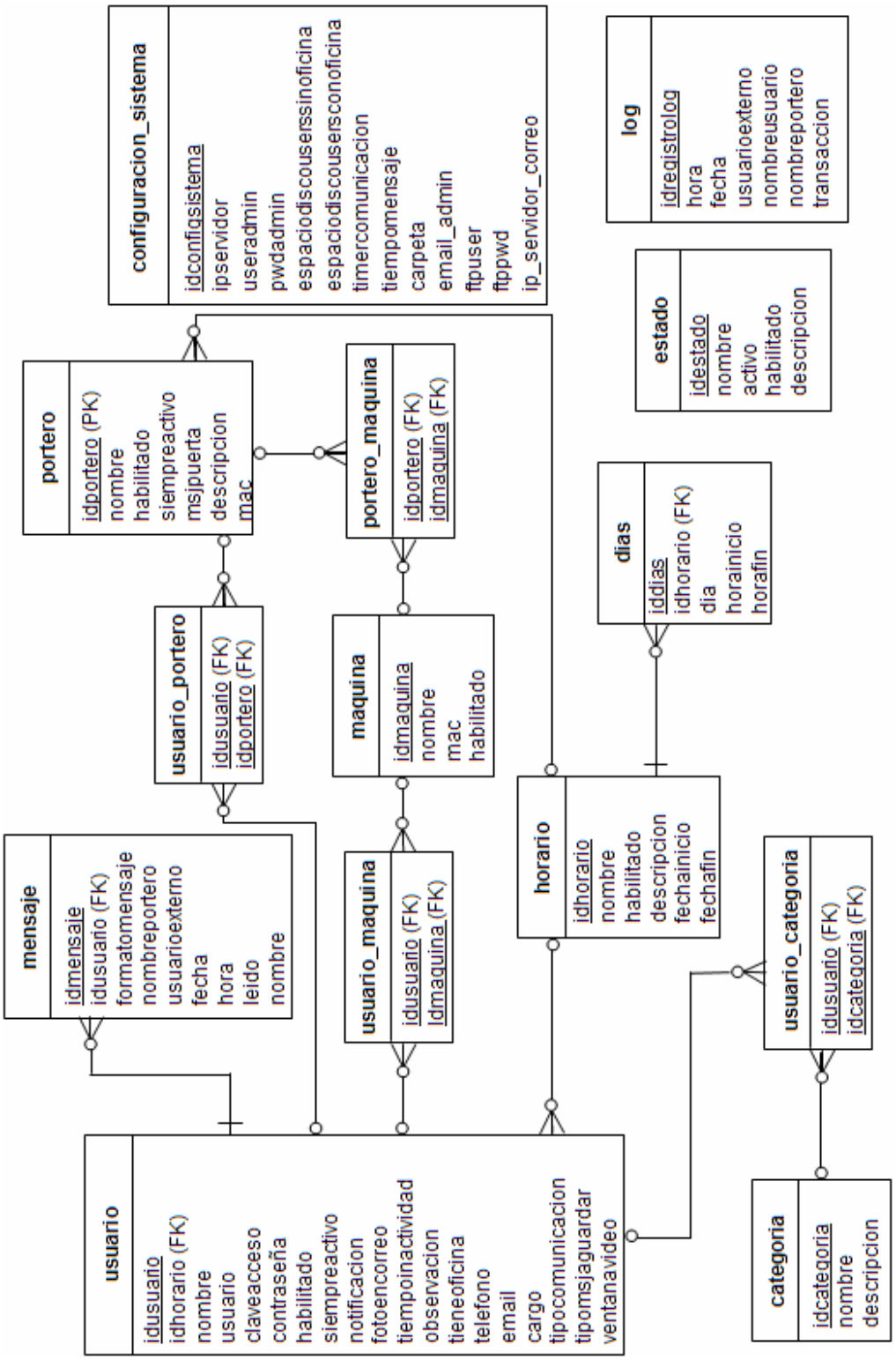


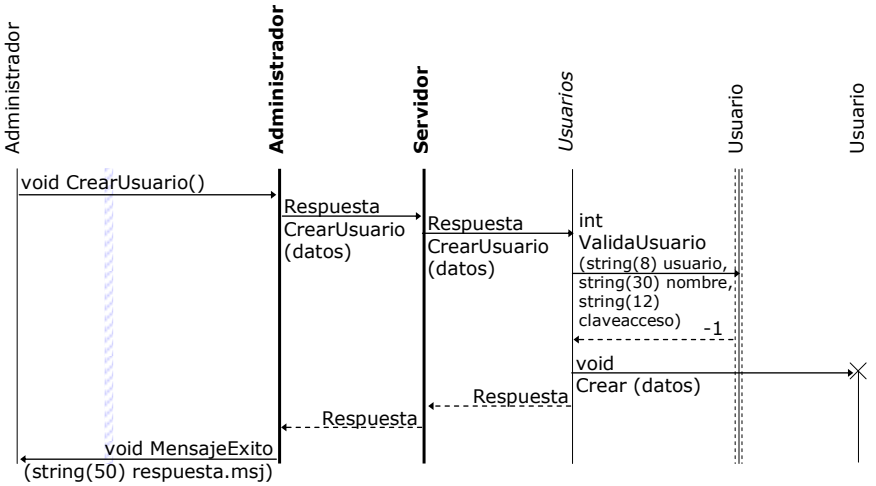
Figura 4.2.- Modelo Lógico del Sistema

4.1.1. Diagramas de Interacción de Objetos

A continuación se presentarán los diagramas de interacción de objetos de diseño, obtenidos a partir de los DIOs de análisis presentados en la sección 3.5.4.

Sistema Administrador

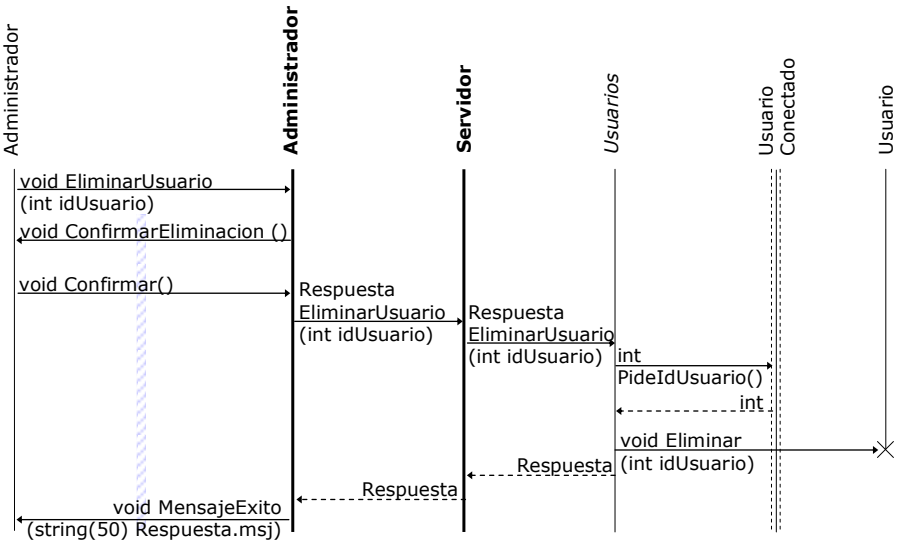
Caso de uso:	Administración de usuarios internos
Escenario:	Creación exitosa de un nuevo usuario interno



datos: string(30) nombre, string(50) cargo, string(50) telefono , string(50) email, boolean habilitado, string(8) usuario, string(8) contrasena, string(12) claveacceso, boolean siempreactivo, int idhorario, string[] categorias, boolean tieneoficina, string [] maquinas, string[] porteros

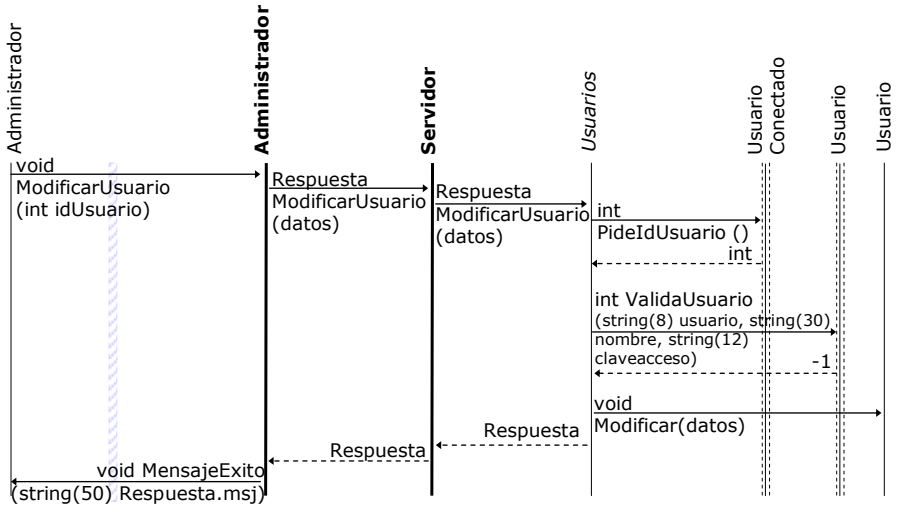
Respuesta: [string , string (msj)] → "1", "El usuario ha sido creado"

Caso de uso:	Administración de usuarios internos
Escenario:	Eliminación exitosa de un usuario interno



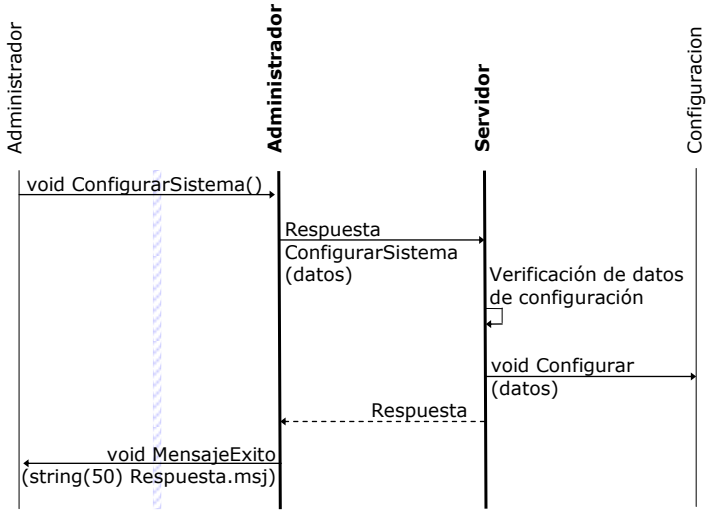
Respuesta: [string , string (msj)] → "1", "El usuario ha sido eliminado"
Si el usuario está conectado, no se lo elimina.

Caso de uso:	Administración de usuarios internos
Escenario:	Modificación exitosa de un usuario interno



datos: int idUsuario, string(30) nombre, string(50) cargo, string(50) telefono , string(50) email, boolean habilitado, string(8) usuario, string(8) contrasena, string(12) claveacceso, boolean siempreactivo, int idhorario, string[] categorias, boolean tieneoficina, string [] maquinas, string[] porteros
Respuesta: [string , string (Msj)] → "1", "El usuario ha sido modificado"
Si el usuario está conectado, no puede ser modificado.

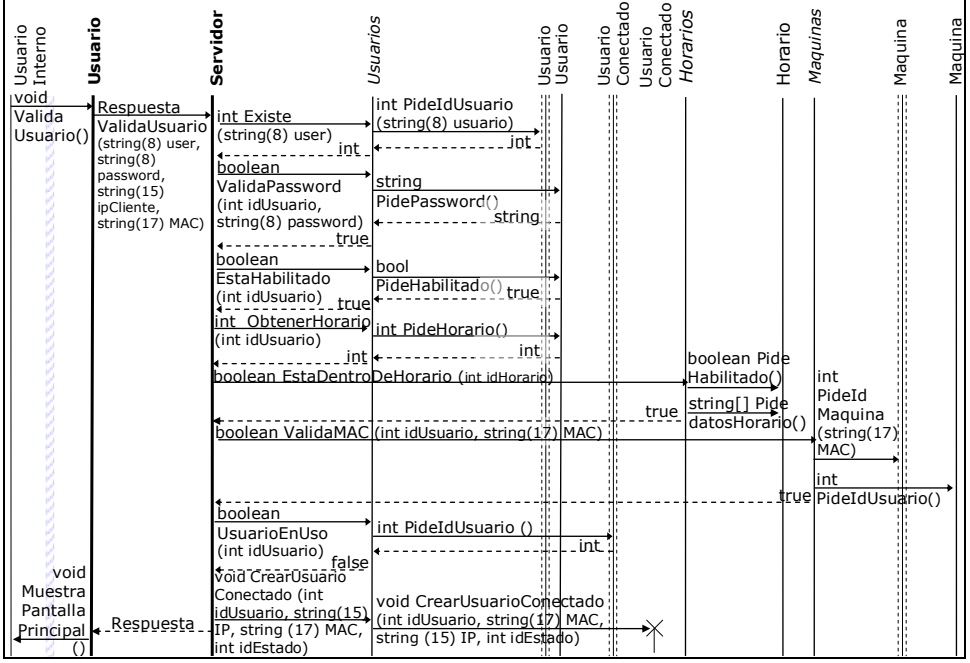
Caso de uso:	Configuración del Sistema por el administrador
Escenario:	Configuración exitosa del sistema



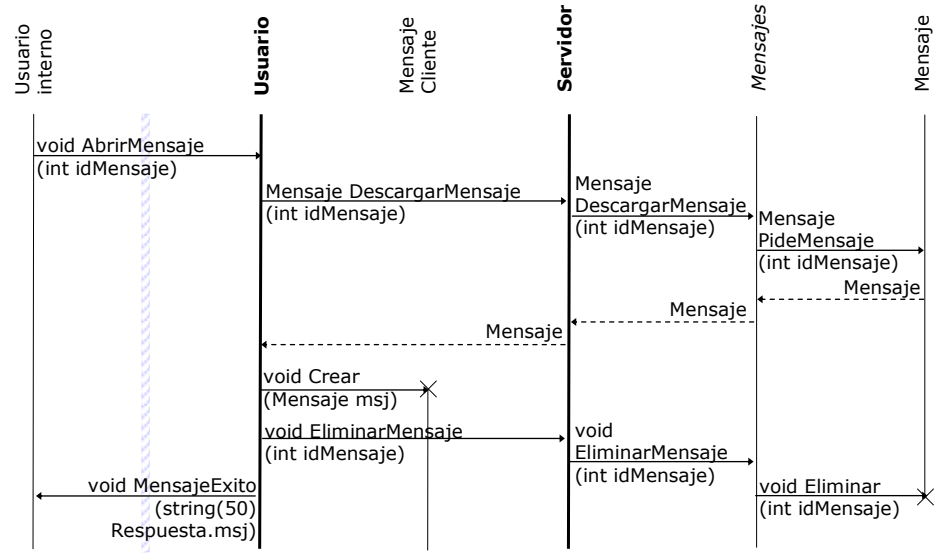
datos: string(15) ip_servidor_correo, int timercomunicacion, string(8) useradmin, string(8) pwdadmin, string(50) email_admin, string(20) carpeta, int espaciodiscousersconoficina, int espaciodiscouserssinoficina, int tiempomensaje, string(8) ftpuser , string(8) ftppwd
Respuesta: [string , string (Msj)] → "1", "La Configuración del sistema ha sido guardada"

Sistema Usuario

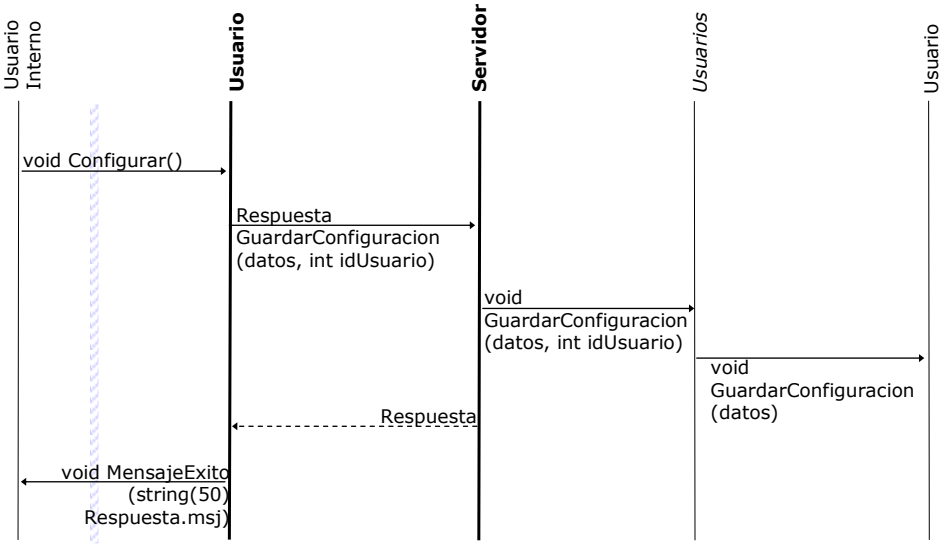
Caso de uso:	Comunicación usuario interno - servidor
Escenario:	Acceso exitoso del usuario interno al sistema



Caso de uso:	Comunicación usuario interno - servidor
Escenario:	Revisión de los mensajes recibidos



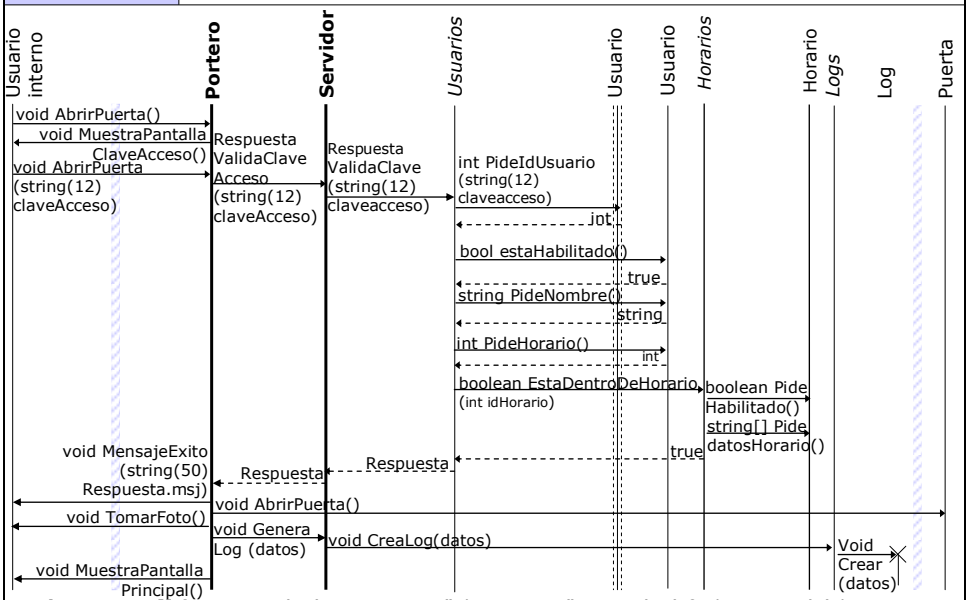
Caso de uso:	Comunicación usuario interno - servidor
Escenario:	Configuración del sistema por el usuario interno



datos: string(50) cargo, string (50) telefono, string(50) email, byte[] foto, string(4) notificacion, boolean fotoEnCorreo, int tiempoInactividad, string(100) observacion, string(5) tipoComunicacion, string(15) tipoMsjAGuardar, string(8) password, boolean ventanavideo
Respuesta: [string , string (Msj)] → "1", "La Configuración del usuario ha sido guardada"

Sistema Portero

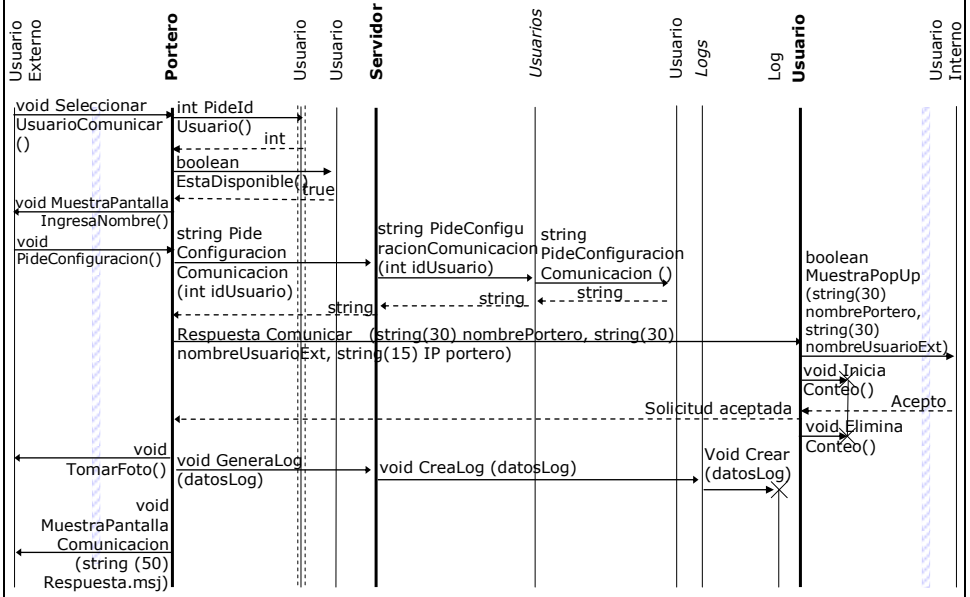
Caso de uso:	Abrir la puerta usando el portero
Escenario:	Un usuario interno solicita acceso, utilizando el Sistema Portero, y se le concede



datos: Byte[] foto, string(15) transaccion "Abrir Puerta", string(10) fecha, string(8) hora, string(12) claveAcceso, string(30) nombrePortero, string(30) nombreUsuarioExt

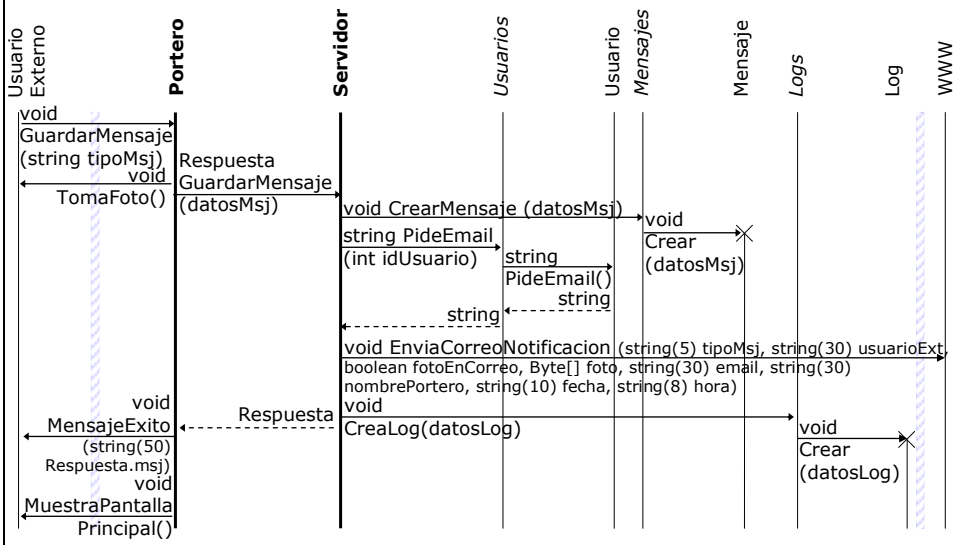
Respuesta: [string , string (Msj)] → "1", "nombre"

Caso de uso:	Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	Solicitud exitosa de comunicación con el usuario interno.



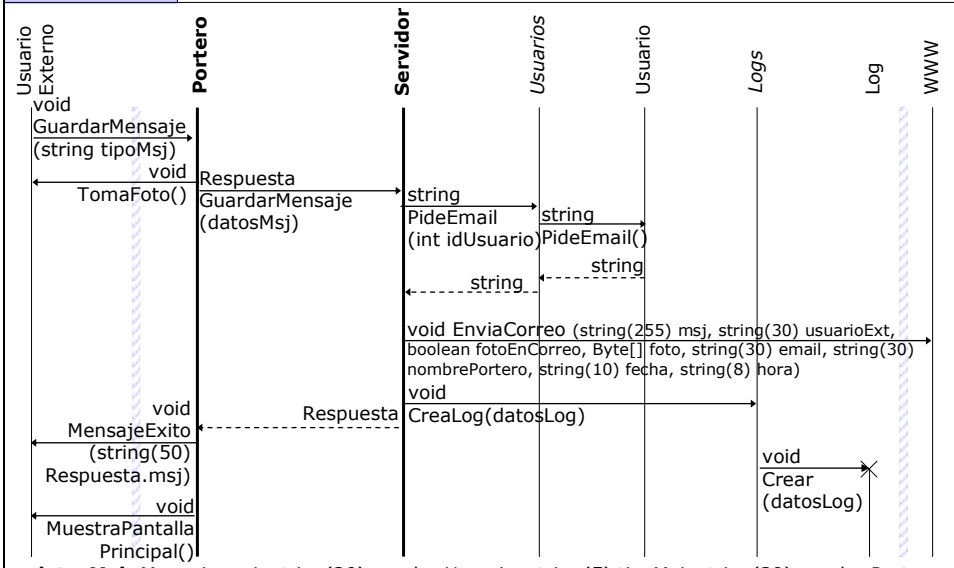
datosLog: Byte[] foto, string(15) transaccion "Comunicación", string(10) fecha, string(8) hora string(30) nombreUsuario, string(30) nombrePortero, string(30) nombreUsuarioExt

Caso de uso:	Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	El usuario externo deja un mensaje de vídeo o de audio al usuario interno



datosMsj: Mensaje msj, int idUsuario, string(5) tipoMsj, int idPortero, string(30) usuarioExt, Byte[] foto, string(10) fecha, string(8) hora.
datosLog: Byte[] foto, string(15) transaccion "Guardar Mensaje", string(10) fecha, string(8) hora string(30) nombreUsuario, string(30) nombrePortero, string(30) nombreUsuarioExt

Caso de uso:	Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	El usuario externo deja un mensaje de texto al usuario interno



datosMsj: Mensaje msj, string(30) nombreUsuario, string(5) tipoMsj, string(30) nombrePortero, string(30) usuarioExt, Byte[] foto, string(10) fecha, string(8) hora.
datosLog: Byte[] foto, string(15) transaccion "Guardar Mensaje", string(10) fecha, string(8) hora string(30) nombreUsuario, string(30) nombrePortero, string(30) nombreUsuarioExt

4.2. Diseño de los módulos del sistema

A continuación se describen los cuatro módulos de diseño que componen el sistema: módulo administrador, módulo portero, módulo usuario, y módulo Web.

4.2.1. Módulo administrador

Este módulo representa al componente de administración o sistema administrador, que es una aplicación que podrá instalarse en cualquier computador de escritorio que tenga las características descritas en la sección 4.1.1.

Los parámetros requeridos para este módulo son la dirección IP del servidor y el tipo de canal de comunicación que ha de utilizarse; estos datos se almacenan en un archivo de configuración, y se leen de éste al momento de iniciar la aplicación.

En cuanto al proceso de acceder al sistema, primeramente se solicita el ingreso del usuario y la contraseña del administrador, que se envían al servidor para que los verifique; éste compara los datos ingresados con los que se encuentran registrados en la tabla "configuracion_sistema" de la base de datos. Una vez realizada esta verificación y

concedido el acceso, este módulo envía al servidor la dirección IP que se utilizó en la conexión con éste, y que se obtuvo a partir de un archivo, para que sea registrada en la tabla "configuracion_sistema". Luego de esto, el administrador puede proceder a manipular los datos de los elementos del sistema (crear, consultar, modificar, eliminar), cambiar la configuración, y solicitar reportes de la bitácora o historial. Adicionalmente, este módulo deberá permitir obtener información relevante de los elementos del sistema, de manera general o detallada, sin necesidad de realizar consultas.

Entre los elementos del sistema que el administrador puede manipular se encuentran:

- *Porteros*.- En los procesos de creación o modificación de un portero se verifica que tanto el nombre como la dirección MAC ingresados sean únicos. Por otro lado, para que el portero pueda estar "habilitado", es necesario que se le asigne por lo menos una máquina, una dirección MAC, y un horario. Para proceder a modificar o eliminar un portero, primeramente se verifica que éste no se encuentre en uso en ese momento, y que ninguno de los

usuarios con oficina, asociados al portero, se encuentre conectado.

- *Máquinas*.- Al crear o modificar una máquina se verifica que los datos ingresados (nombre y dirección MAC) sean únicos. Para que la máquina pueda estar habilitada es necesario que se especifiquen estos datos. En los procesos de eliminación o modificación, por otro lado, se comprueba inicialmente que la máquina no se encuentre en uso para poder realizar la operación; y en el caso de que esté asociada a más de un usuario, debe mostrarse una advertencia, indicando lo que sucede y solicitando que el administrador confirme si desea realizar la operación.

- *Usuarios*.- Al momento de crear o modificar un usuario se verifica que el nombre, usuario y clave de acceso sean únicos; y que haya suficiente espacio en el disco, utilizado por el servidor, para crear una nueva carpeta en la que se almacenarán los mensajes dirigidos a este usuario. Adicionalmente, los campos correspondientes al usuario y la contraseña tendrán una longitud mínima de cinco

caracteres alfanuméricos. En cuanto a la clave de acceso (mínimo ocho caracteres alfanuméricos), el módulo brinda dos posibilidades de asignarla: aleatoriamente (sugerida por el servidor) o por ingreso manual. Por otro lado, para que el usuario pueda estar "habilitado" es necesario especificar su nombre, correo electrónico, usuario, contraseña, clave de acceso, horario, y asignarle por lo menos una máquina o portero. Finalmente, al eliminar o modificar un usuario, se verifica que éste no se encuentre conectado antes de realizar la operación.

- *Horarios*.- En la creación y modificación de un horario, se verifica que el nombre sea único. Para que pueda estar "habilitado" se deben ingresar el nombre y las fechas de inicio y fin, y se le deberá asignar por lo menos un día de actividad (día de la semana, hora inicio y hora fin). Por último, para eliminar o modificar un horario, se debe comprobar que los usuarios o porteros, a los que haya sido asignado, no se encuentren conectados. En caso de que esté relacionado con algún usuario o portero, se mostrará una advertencia indicando lo sucedido al administrador y solicitando que confirme si desea realizar la operación.

- *Categorías*.- En los procesos de creación o modificación de una categoría se verifica que el nombre sea único; y en los de eliminación y modificación, que ninguno de los usuarios a los que está asociada se encuentre conectado. En caso de que la categoría esté asociada a algún usuario, se deberá mostrar una advertencia indicando lo sucedido, y solicitando una confirmación por parte del administrador.

- *Estados*.- Al crear o modificar un estado se debe comprobar que el nombre sea único, y para que éste pueda estar "habilitado" debe asignársele también una imagen. Adicionalmente, la entidad estado consta de un campo denominado "activo", que permite determinar si el usuario que se encuentra en ese estado de conexión, puede o no recibir solicitudes de comunicación. En los procesos de eliminación y modificación se determina si el estado se encuentra en uso, en cuyo caso no se permite que se lleve a cabo la operación; y que no se trate de alguno de los tres estados fijos del sistema (disponible, vuelvo enseguida y desconectado), en cuyo caso se mostrará una advertencia indicando que está por modificarse uno de estos tres estados básicos, y

solicitando confirmación. Cuando esto sucede, no es posible modificar el campo "activo" de dichos estados, y si se trata de una eliminación, la operación no es permitida.

Este módulo también permite que se realicen cambios en la configuración del sistema. Los datos que se pueden modificar son: la dirección IP del servidor de correo; el tiempo máximo de espera antes de dar por rechazada una solicitud de comunicación; el usuario y la contraseña del administrador; la dirección del correo electrónico del administrador, que es utilizada por el sistema para enviar los mensajes de advertencia cuando la bitácora ha superado el número máximo de registros, y cuando se envían a los usuarios internos notificaciones de que han recibido un nuevo mensaje; el nombre de la carpeta en la que se guardan los directorios que contienen los mensajes de los usuarios; el espacio asignado, en mega bytes, a los usuarios que disponen de una oficina y a los que no; y el usuario y la contraseña de la cuenta para acceder vía FTP al servidor. Cuando los datos modificados corresponden al espacio asignado a los usuarios, el servidor realiza una verificación, antes de guardar los datos, que permite determinar si hay

suficiente espacio disponible para aplicar los cambios solicitados; en caso de que no sea así, se muestra un mensaje al administrador, indicando que no ha sido posible modificar estos datos.

Adicionalmente, el módulo de administración dispone de una opción que permite importar usuarios, a partir de un archivo de texto. Este archivo deberá contener el nombre completo, usuario, y contraseña de cada persona que vaya a utilizar el sistema; y los campos deberán ir separados en cada línea, por un carácter que se especifica antes de dar comienzo al proceso de importación. Durante este proceso, primeramente se comprueba que el archivo suministrado cumpla con el formato descrito, y antes de proceder a la creación de un nuevo usuario se verifica que los datos correspondientes al nombre completo y usuario de éste sean únicos en el sistema, y que el número de caracteres en cada campo no sea inferior ni superior a los límites establecidos. Esta opción se incluyó con el objetivo de proveer cierto grado de integración con los sistemas que ya existan dentro de la organización, y de facilitar la tarea de registro de usuarios al administrador.

Por último, el módulo permite realizar consultas en base a las operaciones registradas en la bitácora. Entre los criterios de búsqueda que pueden utilizarse para la presentación de reportes están los siguientes: de acuerdo a la transacción ("Abrir Puerta", "Comunicación", "Guardar Mensaje"), al portero desde el que se realizó la transacción, y al usuario interno que intervino. Los resultados anteriores deben limitarse al rango de fechas ingresado por el usuario que solicita la presentación del reporte, y pueden estar ordenados indistintamente de acuerdo a la transacción, al portero, y al usuario interno, para facilitar la visualización de los datos de interés. El reporte que se presentará constará de los datos utilizados en la consulta, y adicionalmente, mostrará para cada transacción, la hora en que se realizó, y el nombre y la foto del usuario externo.

4.2.2. Módulo portero

Este módulo representa al componente de comunicación o sistema portero, que es una aplicación que estará instalada en un computador, ubicado a la entrada del área de oficinas. Este computador estará equipado con un teclado, un dispositivo de captura de audio, y dos de vídeo.

Los parámetros requeridos para este módulo son la dirección IP del servidor, el tipo de canal de comunicación que ha de utilizarse, y los números de puerto que se emplearán para establecer comunicaciones instantáneas con los usuarios internos. Estos parámetros se almacenan en un archivo de configuración, y se leen de éste al momento de iniciar la aplicación.

El módulo trabaja con dos dispositivos de captura de vídeo y uno de audio. En el caso de los de vídeo, uno se utiliza para capturar fotos de los usuarios externos, para la comunicación instantánea, y para grabar mensajes; y el otro, para capturar la foto de las personas que ingresan al área de oficinas por la puerta. El primero estará ubicado cerca del monitor, y enfocará hacia el usuario externo, mientras que el segundo estará ubicado en el techo y enfocará hacia la puerta.

El funcionamiento de estos dispositivos de captura que el sistema utiliza, se verifica previamente mediante una prueba, que se proveerá como parte de este módulo. Esto permite asegurar que los dispositivos funcionen correctamente. Los datos de la configuración de estos

dispositivos se almacenan en un archivo, en el que se indica cuáles son las cámaras que enfocarán hacia la puerta y hacia el usuario, respectivamente; el dispositivo de captura de audio; y los compresores tanto de vídeo como de audio que se utilizarán.

Este módulo está diseñado para verificar inicialmente que los dispositivos de captura hayan sido configurados, y que los datos de su configuración se encuentren registrados en el archivo correspondiente. Una vez realizada la comprobación anterior, se solicita acceso al servidor, enviando tanto la dirección MAC como la dirección IP de la computadora portero. La primera sirve para determinar si se trata de un portero registrado y habilitado para el uso, mientras que la segunda se requiere para el establecimiento de comunicaciones instantáneas. Dentro de esta verificación, se comprueba que el portero se encuentre funcionando dentro del horario que se le haya asignado. Finalmente, terminado este proceso de validación, el servidor envía una respuesta al sistema portero indicando si se le concede acceso o no, y en caso de que la respuesta sea afirmativa se envía también la ID (dato que identifica al portero) del portero correspondiente.

Este módulo presentará al usuario externo una lista a manera de árbol, que muestra los usuarios internos asociados con el portero. Cada usuario interno estará representado inicialmente por su nombre y una imagen que indique su estado de conexión (por ejemplo, "Disponible", "Vuelvo Enseguida", "Desconectado", etc.). Posteriormente, si el usuario externo selecciona un elemento de la lista, se presentarán los datos detallados asociados a dicho usuario interno (nombre, observación, foto, estado de conexión, cargo, teléfonos, y una lista de categorías a las que está asociado).

El sistema también dispone de una opción que permite realizar búsquedas de usuarios internos, en caso de que el usuario externo no desee recorrer la lista anteriormente mencionada. El criterio de búsqueda que deberá ingresarse puede consistir en unos cuantos caracteres que formen parte del nombre; y para limitar el número de resultados que se obtengan, se puede especificar adicionalmente el nombre de alguna categoría a la que ese usuario pertenezca.

Una vez que el usuario externo haya seleccionado al usuario interno con el que se desea comunicar, tendrá la opción de

dejar un mensaje (texto, vídeo, audio) o de establecer una comunicación, dependiendo del estado de conexión de éste.

En caso de que el usuario interno se encuentre disponible, el usuario externo tendrá la posibilidad de enviar mensajes escritos en tiempo real, o de establecer una comunicación de vídeo o audio con él; esto dependerá del tipo de comunicación (texto, vídeo, o audio) que éste haya especificado en su configuración. El usuario interno podrá entonces decidir si permite el ingreso al usuario externo, en cuyo caso podrá mandar a abrir la puerta, desde la aplicación usuario instalada en su computadora (el mensaje de "abrir puerta" es enviado al sistema portero, que se encarga de accionar el mecanismo que abre la cerradura).

En caso de que el usuario interno no se encuentre disponible, esté ausente, o no disponga de una oficina, el sistema permitirá dejar mensajes (texto, audio, o vídeo), dependiendo de lo que el usuario haya especificado en su configuración y del espacio disponible en su directorio de mensajes.

Si los mensajes son escritos, el sistema portero solicitará al servidor que se envíen automáticamente a la dirección de correo electrónico del usuario interno; en caso contrario, el e-mail que enviará el servidor únicamente consistirá en una indicación de que el usuario interno ha recibido un nuevo mensaje de vídeo o de audio, y de que debe descargarlo por medio del sistema usuario o sitio Web. En ambos casos, si el usuario interno lo ha especificado en su configuración, se adjuntará una "foto instantánea" del remitente, que será tomada al momento de solicitar el envío. Con esto será posible identificar de manera inequívoca a la persona que envía el mensaje, aunque ésta sea ajena al ámbito de la organización.

Cada vez que se envía un nuevo mensaje de vídeo o de audio, el servidor realiza una verificación del espacio disponible para que el sistema pueda continuar funcionando correctamente.

Otra de las operaciones que se realizan por medio del portero es la apertura de la puerta para permitir el ingreso a los usuarios internos. Para esto, el sistema solicita el ingreso de la clave de acceso que se le haya asignado al usuario, y la

envía al servidor para que éste se encargue de validar que se trate de una clave registrada, y que el usuario se encuentre habilitado y dentro del horario de acceso asignado. Una vez realizada esta verificación, el servidor envía una respuesta al sistema portero, indicando si se debe o no conceder acceso al usuario.

Cada vez que se lleva a cabo una operación dentro del sistema ("Abrir Puerta", "Comunicación", "Guardar Mensaje"), el sistema portero solicita al servidor que se registre la nueva transacción en la bitácora. Los datos que se guardan para cada registro son: la hora y la fecha en la que la transacción se llevó a cabo; el nombre del portero desde el que se la realizó; los nombres del usuario interno y del usuario externo que intervinieron en el proceso; el nombre que identifica a la transacción; y la imagen capturada por el portero del usuario externo.

Cuando un portero concluye su periodo de actividad de acuerdo al horario que se le haya asignado, se envía una solicitud al servidor indicando que su periodo de trabajo ha concluido, y que va a apagarse. El servidor entonces se encarga de actualizar la información correspondiente, eliminándolo de la lista de porteros conectados. Si por

alguna razón el portero se apaga o deja de funcionar repentinamente, la información de conexión se mantiene actualizada por medio del envío de notificaciones de conexión periódicas al servidor. En estas notificaciones se envía la ID del portero, la dirección MAC y la dirección IP.

Dado que el computador en el que se instale el sistema portero estará destinado únicamente a ejecutar esta aplicación, el acceso a las funciones que ofrece el sistema operativo, tales como la manipulación de las teclas de control y sus combinaciones, y el acceso a la barra de tareas, debe quedar restringido para los usuarios.

En caso de que el administrador necesite acceder a estas funciones deberá teclear un código, mientras se encuentre en la pantalla inicial de la aplicación, que luego se compara con el que se encuentra registrado en uno de los archivos de configuración. Si la verificación resulta correcta, se mostrará una ventana en la que se solicita el ingreso del usuario y la contraseña del administrador, durante un lapso. Esta información se envía al servidor para que sea verificada, y en caso de que los datos sean correctos, se activarán las funciones del sistema operativo, que se hayan bloqueado

anteriormente. Para volver a bloquear estas funciones se deberá efectuar el mismo procedimiento.

4.2.3. Módulo usuario

Este módulo representa al componente que interactúa con los usuarios internos o sistema usuario. Ésta es una aplicación que estará instalada en la computadora de escritorio de cada usuario interno que se encuentre dentro de una determinada área de oficinas. Esta computadora puede estar equipada con dispositivos de captura de audio o vídeo; de ser así, el usuario interno podrá, si lo desea, establecer comunicaciones de este tipo con los usuarios externos.

Los parámetros requeridos para este módulo son la dirección IP del servidor, el tipo de canal de comunicación que ha de utilizarse, y los números de puerto que se emplearán para establecer comunicaciones instantáneas con los usuarios externos por medio del sistema portero. Estos parámetros se almacenan en un archivo de configuración, y se leen de éste al momento de iniciar la aplicación. Adicionalmente, el dispositivo de captura de vídeo utilizado por el sistema en el proceso de comunicación (en caso de que se encuentre

instalado alguno) se debe especificar como parte de la información de configuración necesaria para que la aplicación funcione correctamente.

En cuanto al proceso de acceder al sistema usuario, el módulo está diseñado para verificar inicialmente que los datos de la configuración se encuentren registrados en el archivo correspondiente. Una vez realizada esta comprobación, se solicita el ingreso de un usuario, y una contraseña, por parte del usuario interno, y se obtiene la dirección MAC de la computadora desde la que éste solicita acceso; una vez que se obtienen estos datos, se envía esta información al servidor, que se encarga de llevar a cabo ciertas verificaciones.

Primeramente, comprueba que se trate de alguno de los usuarios existentes, es decir, que el usuario ingresado se encuentre registrado. Como segundo paso, se procede a validar que la contraseña ingresada sea correcta, o lo que es lo mismo, que coincida con la que se haya registrado previamente para el usuario. Una vez comprobados estos dos puntos, se verifica que esté "habilitado", que no se encuentre conectado, y se confirma que esté ingresando dentro del horario que se le haya asignado; luego, se

determina si la máquina desde la que está solicitando acceso corresponde a alguna de las que le hayan sido asignadas, por medio de la verificación de la dirección MAC. Por último el servidor retornará una respuesta al sistema usuario indicando si se debe conceder acceso o no. En caso de que la respuesta sea afirmativa, se retornará además la ID (dato que identifica al usuario en el sistema) del usuario correspondiente.

Si el usuario ha sobrepasado el 80% del espacio asignado para su directorio de mensajes, se le mostrará una advertencia, indicando que debe descargar sus mensajes del servidor.

Este módulo está diseñado para permitir al usuario descargar y revisar mensajes; modificar sus datos personales, sus contraseñas, y la configuración de la aplicación; visualizar el vídeo que está siendo capturado por los porteros a los que el usuario se encuentre asociado; cambiar su estado de conexión; y aceptar o denegar solicitudes de comunicación de parte de usuarios externos.

Para descargar y revisar mensajes el sistema presenta inicialmente una lista de los mensajes recibidos que no han sido descargados, especificando para cada uno el asunto (obtenido a partir del nombre del archivo en el que se encuentra almacenado), la hora y la fecha en que fue enviado, desde qué portero se realizó el envío, el tipo de mensaje (audio, o vídeo), y su estado, que indica si ha sido descargado vía Web. El usuario debe seleccionar los mensajes que desea descargar de entre la lista de mensajes recibidos, y una vez descargados, puede proceder a revisarlos seleccionándolos de la lista de mensajes guardados.

Cuando un mensaje es descargado por medio de la aplicación, es eliminado automáticamente del servidor y de la lista de mensajes recibidos, y pasa a formar parte de la lista de mensajes guardados. Estos mensajes se almacenan en una carpeta, cuyo nombre es igual al "usuario" registrado para el usuario interno, y que se encuentra dentro de un directorio denominado "mensajes", ubicado en la ruta donde está instalada la aplicación. Cuando los mensajes han sido revisados, el sistema cambia el valor del campo "leído" a "1", y da la opción de eliminar los archivos del directorio.

Este módulo permite que el usuario modifique sus datos personales tales como el cargo, el teléfono, la dirección de correo electrónico y la foto; su contraseña para ingresar al sistema (mínimo cinco caracteres); y los datos de la configuración de la aplicación tales como el tipo de comunicación, el tipo de mensajes a recibir, etc. También brinda la opción al usuario de visualizar la clave de acceso que le permite abrir la puerta usando el módulo portero; para esto se solicita que el usuario interno ingrese nuevamente su usuario y su contraseña, por motivos de seguridad.

Otra de las operaciones que este módulo permite llevar a cabo, es la de visualizar el vídeo que está siendo capturado por algún portero. Para esto, se mostrará inicialmente una lista de los porteros que se encuentren conectados y que estén asociados al usuario, y éste podrá seleccionar aquel cuyo vídeo desea visualizar. Esta opción estará habilitada para los usuarios que en la instalación del sistema indicaron que el procesador de su computador es equivalente o superior a un Pentium IV.

Este módulo brinda la opción al usuario de cambiar su estado de conexión, mostrando la lista de los estados registrados y habilitados en el sistema, como por ejemplo "Disponible", "Vuelvo enseguida", entre otros, para que él pueda seleccionar de entre estos. Por otro lado, este módulo realiza un cambio automático de estado cada vez que no se detecta actividad por parte del usuario durante un lapso determinado, que se especifica en la configuración de la aplicación.

Adicionalmente, este módulo permite al usuario interno aceptar o denegar solicitudes de establecer comunicaciones de texto (a manera de un Chat), de audio o de vídeo (a manera de una videoconferencia), de parte de los usuarios externos.

Cada vez que un usuario interno recibe una solicitud de comunicación, se le muestra un mensaje emergente indicando el nombre del usuario externo y el portero desde el que se hace la solicitud, lo que le brinda la opción de aceptarla o denegarla; adicionalmente, se inicializa un temporizador encargado de controlar que el tiempo de espera por una respuesta de parte del usuario interno, no

sea mayor a lo establecido en el campo "timercomunicacion" de la configuración del sistema.

En caso de que se acepte la solicitud de comunicación, el usuario interno tendrá la posibilidad de visualizar al usuario externo que se encuentra utilizando el sistema portero (si es que el campo "*ventanavideo*" tiene el valor de "1", para las comunicaciones de audio o texto), establecer una comunicación con él, y mandar a abrir la puerta desde el sistema, si así lo desea.

En caso de que el temporizador expire antes de que se envíe una respuesta a la solicitud, el sistema usuario cambia el estado de conexión del usuario interno al de ID=2, que representa al estado ausente; y a continuación, envía automáticamente una respuesta al usuario externo, indicándole que la persona solicitada no responde.

En caso de que el usuario interno rechace la solicitud de comunicación, se le cambiará el estado de conexión al de ID=2, que representa al estado ausente, y se enviará una respuesta al sistema portero, indicando que el usuario interno ha denegado la solicitud de comunicación.

Durante el tiempo que se mantenga la conexión, si es que el usuario accede también desde el módulo usuario Web, se le enviará un mensaje de notificación indicando esto, con el fin de mantenerlo informado acerca de las conexiones que tiene abiertas, y de mejorar la seguridad en el acceso al sistema.

Finalmente, cuando el usuario interno concluye su período de actividad de acuerdo al horario que se le haya asignado, se envía una solicitud al servidor indicando que su periodo de trabajo ha concluido, y que se va a cerrar su aplicación. El servidor entonces, para mantener actualizada la información correspondiente a la lista de usuarios conectados, lo elimina de la lista. Si por alguna razón la computadora del usuario interno se apaga o el sistema deja responder repentinamente, la información de conexión se mantiene actualizada por medio del envío de notificaciones de conexión periódicas al servidor. En estas notificaciones se envía la ID del usuario, la dirección MAC y la dirección IP.

4.2.4. Módulo Web

Este módulo representa al componente Web del sistema usuario. Está diseñado para que lo utilicen los usuarios internos, especialmente aquellos que no tienen oficina, y

puede accederse a él desde cualquier computador que tenga conexión a Internet y disponga de un navegador Web.

Los parámetros que este módulo requiere para funcionar son la dirección IP del servidor, y el tipo de canal de comunicación que ha de utilizarse. Estos parámetros se almacenan en un archivo de configuración, y se obtienen de éste al momento de solicitar ingreso al sistema.

Al iniciar el proceso de acceso al sistema, se solicita al usuario interno, el ingreso de un usuario, y una contraseña, que serán enviados al servidor, para que realice las verificaciones correspondientes.

Primeramente, se comprueba que se trate de alguno de los usuarios existentes, es decir, que el usuario ingresado se encuentre registrado. Como segundo paso, se procede a validar que la contraseña ingresada sea correcta, o lo que es lo mismo, que coincida con la que se haya registrado previamente para el usuario. Una vez comprobados estos dos puntos, se verifica que el usuario esté "habilitado"; y, por último, el sistema usuario Web recibe una respuesta del servidor indicando si se debe conceder acceso o no. En caso

de que la respuesta sea afirmativa, se retornará además la ID del usuario; y se mostrará la página Web que se encarga de presentar los mensajes que éste ha recibido.

Cuando el tamaño del directorio de mensajes del usuario haya sobrepasado el 80% del espacio asignado para él, se le mostrará una advertencia, indicando que debe descargar y borrar sus mensajes del servidor.

Este módulo está diseñado para permitir al usuario descargar y revisar sus mensajes; y modificar sus datos personales, contraseñas, y ciertos parámetros de configuración, requeridos por el sistema portero.

Una vez concedido el acceso al sitio Web, para que el usuario pueda descargar y revisar sus mensajes, el sistema presenta una lista de los mensajes recibidos que no han sido descargados y eliminados, especificando para cada uno de éstos el asunto (obtenido a partir del nombre del archivo en el que se encuentra almacenado); la hora y la fecha en que fue enviado; desde qué portero se realizó el envío; el tipo de mensaje (audio, o vídeo); y el estado en el que se encuentra, que indica si ha sido descargado vía Web o no. El

usuario debe seleccionar los mensajes que desea descargar de entre la lista de mensajes recibidos, y una vez descargados, puede proceder a borrarlos del servidor si así lo desea, y a revisarlos desde su computador. Cuando un mensaje es descargado por medio del módulo Web, al campo "leído" de éste se le asigna el valor de "1".

Este módulo permite que el usuario modifique sus datos personales tales como el cargo, el teléfono, la dirección de correo electrónico y la foto; su contraseña para ingresar al sistema (mínimo cinco caracteres); y los datos de la configuración de la aplicación tales como el tipo de mensajes a recibir, el mensaje que se presentará a los porteros asociados, etc. También brinda la opción al usuario de recordar la clave de acceso que le permite abrir la puerta usando el módulo portero; para esto se solicita que él ingrese nuevamente su usuario y su contraseña, por motivos de seguridad, y si estos datos son correctos, se le enviará un correo electrónico adjuntando la respectiva clave de acceso.

Mientras dure la sesión, si es que el usuario también se conecta desde el módulo usuario instalado en la computadora que utiliza dentro del área de oficinas, se le enviará un mensaje de notificación indicando esto, con el fin

de mantenerlo informado acerca de las conexiones que tiene abiertas, y de mejorar la seguridad en el acceso al sistema.

Finalmente, cuando el usuario interno se extiende del periodo de sesión asignado, se envía una solicitud al servidor indicando que su estado de conexión debe cambiarse a "desconectado". El servidor entonces, para mantener actualizada la información, lo elimina de la lista correspondiente a los usuarios Web conectados.

4.3. Diseño de la comunicación entre los componentes

Las aplicaciones que componen el Sistema Computarizado de Comunicación y Control de Ingreso a Oficinas, hacen requerimientos al servidor, que a la vez se comunica con la base de datos, para poder llevar a cabo sus tareas, como se explicó en la sección 3.5.1; para esto es necesario que cada componente conozca la dirección IP del servidor.

Todas las comunicaciones dentro del sistema siguen este esquema, excepto las que se establecen entre una aplicación portero y una aplicación usuario, por ejemplo, cuando un usuario externo solicita el establecimiento de una comunicación con un usuario interno, ya sea ésta de texto, audio o vídeo. En estos casos, los componentes

se comunican en un esquema punto a punto, y ya no se usa al servidor como intermediario. Las comunicaciones instantáneas de texto se llevan a cabo por medio de sockets, especificando los números de puerto y las direcciones IP respectivas, para crearlos. El número de puerto que se utiliza desde uno de los extremos para enviar mensajes, corresponde en el otro, al número de puerto utilizado para recibirlos; y viceversa. Tal como se muestra en la figura 4.3.

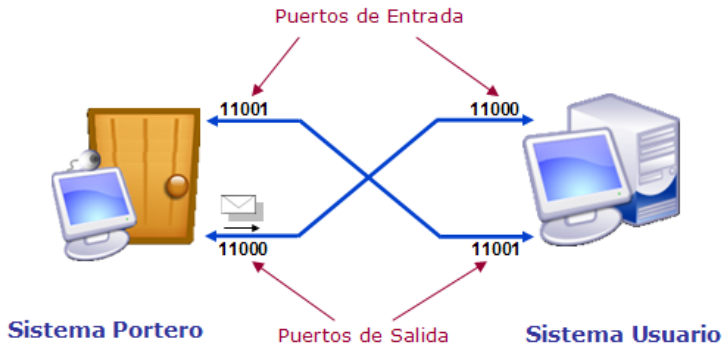


Figura 4.3.- Esquema de comunicaciones de texto entre el sistema portero y sistema usuario.

Adicionalmente, los mensajes enviados entre el emisor y el receptor siguen un formato predeterminado, que se muestra en la figura 4.4. Estos mensajes viajan encriptados a través de la red.

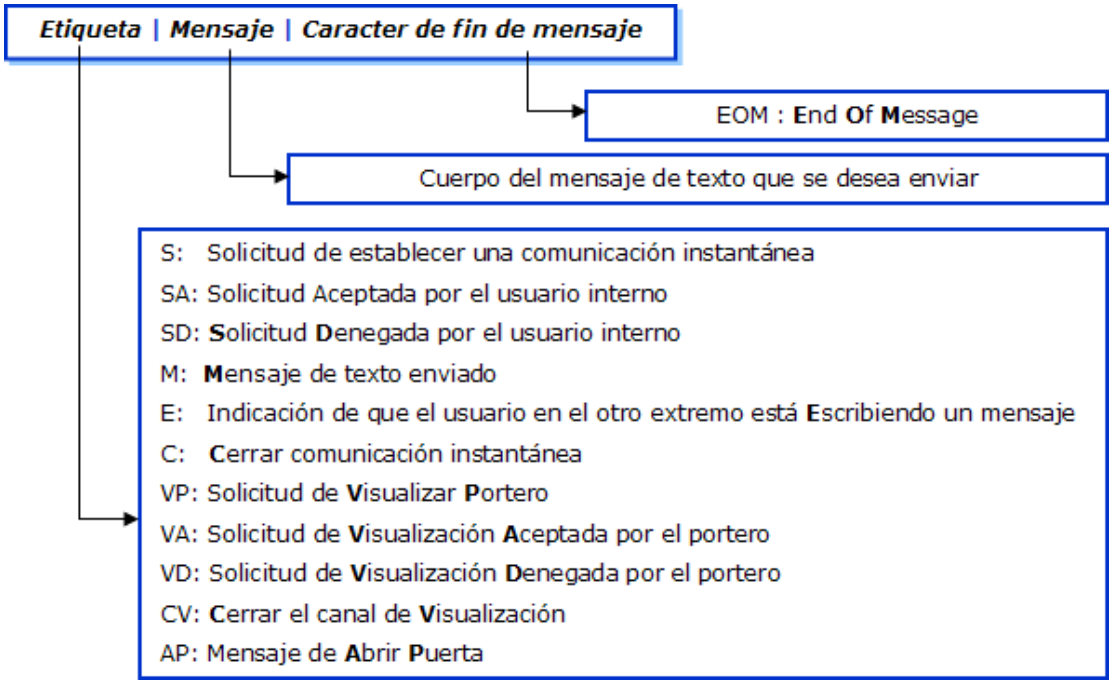


Figura 4.4.- Formato de los mensajes enviados entre el sistema usuario y el sistema portero.

4.4. Diseño de la interacción

Con el objetivo de diseñar un sistema con el que el usuario pueda interactuar fácilmente y de manera intuitiva, se han tomado en consideración los principios de usabilidad de la interacción hombre-máquina, descritos anteriormente en la sección 3.3, para el diseño de cada uno de los componentes del sistema.

4.4.1. Interacción del sistema administrador

Con el objetivo de cumplir con el principio de *consistencia*, en el sistema administrador, las funciones que permiten manipular los datos (crear, consultar, modificar y eliminar) y

que son comunes para los diferentes elementos del sistema, deberán estar colocadas, agrupadas y presentadas del mismo modo en las ventanas respectivas. Esto se logra al colocar una barra de tareas, que permita acceder a las diferentes funciones de manipulación de datos, en cada una de las ventanas secundarias de la aplicación, que representan a los elementos del sistema, como se muestra en la figura 4.5. Esto permite que los usuarios puedan aprender rápidamente a realizar las tareas más comunes.

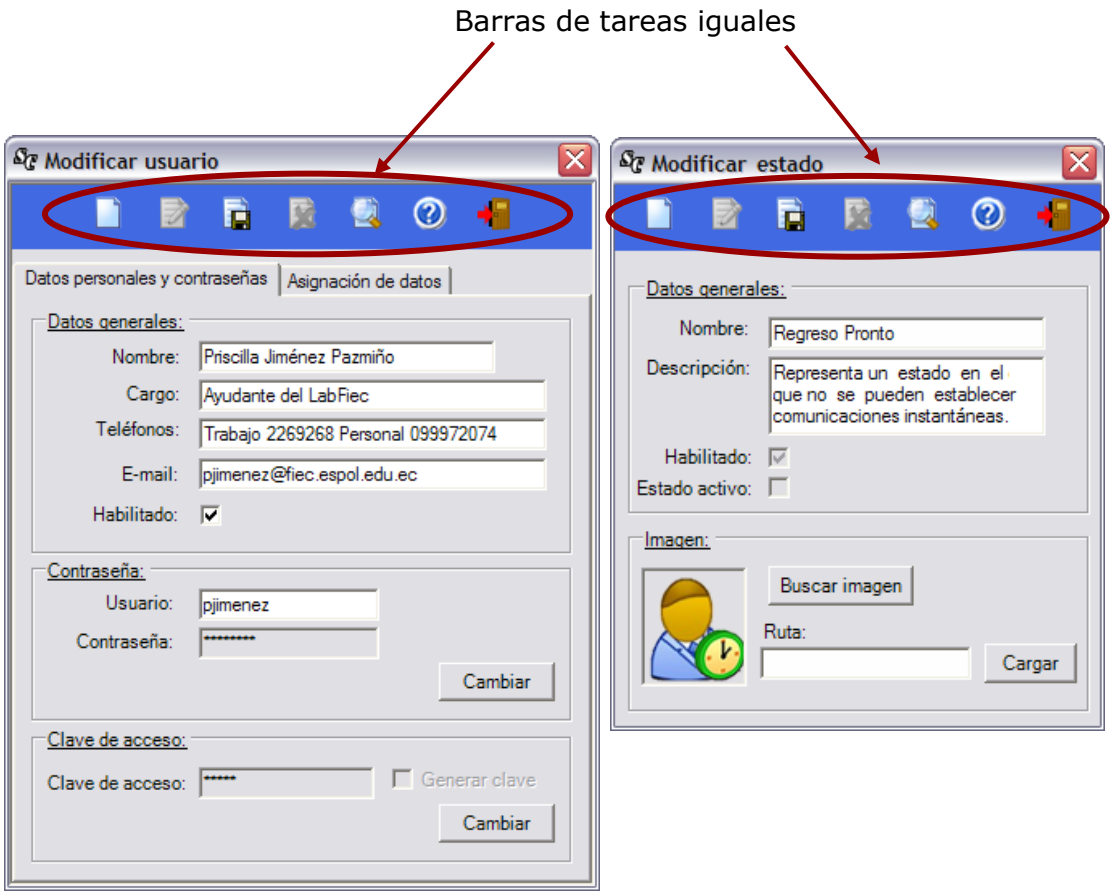
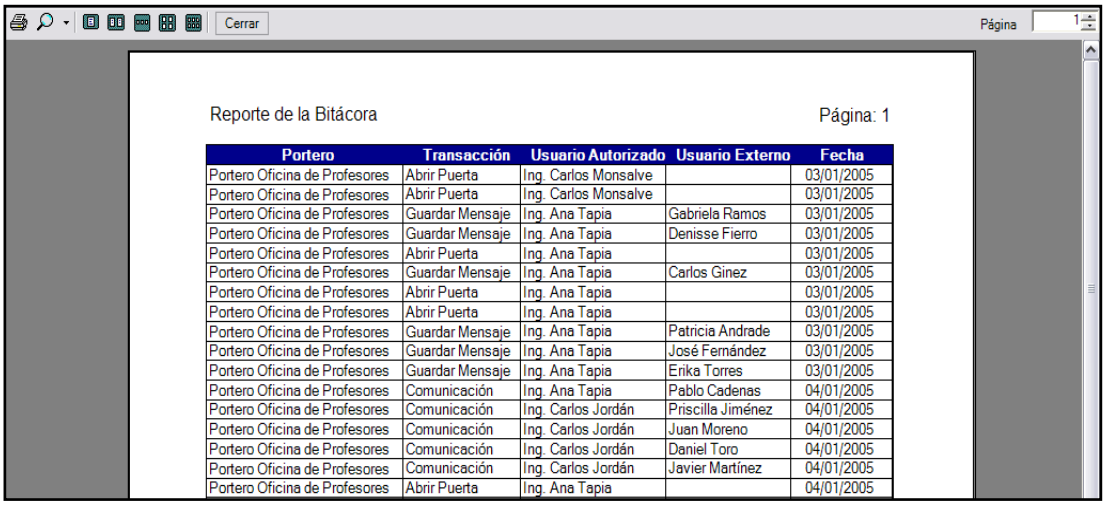


Figura 4.5.- Barras de tareas iguales en las ventanas secundarias

La aplicación del principio de *familiaridad* se observará en la utilización de componentes de interfaz comunes para la mayoría de las aplicaciones de uso frecuente, como por ejemplo, en el empleo de un menú desde el que se puede acceder a todas las funciones del sistema; y en el uso de una barra de tareas horizontal y una vertical que brindan acceso a un grupo específico de funciones, tal como se muestra en la figura 4.6.

Figura 4.6.- Elementos principales de la interfaz del sistema administrador

Adicionalmente, el sistema deberá proveer herramientas de uso habitual, como por ejemplo, la vista preliminar para la impresión, en el caso de los reportes basados en la bitácora, y la ventana de ayuda del sistema, tal como se presenta en las figuras 4.7 y 4.8, respectivamente.



Portero	Transacción	Usuario Autorizado	Usuario Externo	Fecha
Portero Oficina de Profesores	Abrir Puerta	Ing. Carlos Monsalve		03/01/2005
Portero Oficina de Profesores	Abrir Puerta	Ing. Carlos Monsalve		03/01/2005
Portero Oficina de Profesores	Guardar Mensaje	Ing. Ana Tapia	Gabriela Ramos	03/01/2005
Portero Oficina de Profesores	Guardar Mensaje	Ing. Ana Tapia	Denisse Fierro	03/01/2005
Portero Oficina de Profesores	Abrir Puerta	Ing. Ana Tapia		03/01/2005
Portero Oficina de Profesores	Guardar Mensaje	Ing. Ana Tapia	Carlos Ginez	03/01/2005
Portero Oficina de Profesores	Abrir Puerta	Ing. Ana Tapia		03/01/2005
Portero Oficina de Profesores	Abrir Puerta	Ing. Ana Tapia		03/01/2005
Portero Oficina de Profesores	Guardar Mensaje	Ing. Ana Tapia	Patricia Andrade	03/01/2005
Portero Oficina de Profesores	Guardar Mensaje	Ing. Ana Tapia	José Fernández	03/01/2005
Portero Oficina de Profesores	Guardar Mensaje	Ing. Ana Tapia	Erika Torres	03/01/2005
Portero Oficina de Profesores	Comunicación	Ing. Ana Tapia	Pablo Cadenas	04/01/2005
Portero Oficina de Profesores	Comunicación	Ing. Carlos Jordán	Priscilla Jiménez	04/01/2005
Portero Oficina de Profesores	Comunicación	Ing. Carlos Jordán	Juan Moreno	04/01/2005
Portero Oficina de Profesores	Comunicación	Ing. Carlos Jordán	Daniel Toro	04/01/2005
Portero Oficina de Profesores	Comunicación	Ing. Carlos Jordán	Javier Martínez	04/01/2005
Portero Oficina de Profesores	Abrir Puerta	Ing. Ana Tapia		04/01/2005

Figura 4.7.- Vista preliminar de reportes de la bitácora.

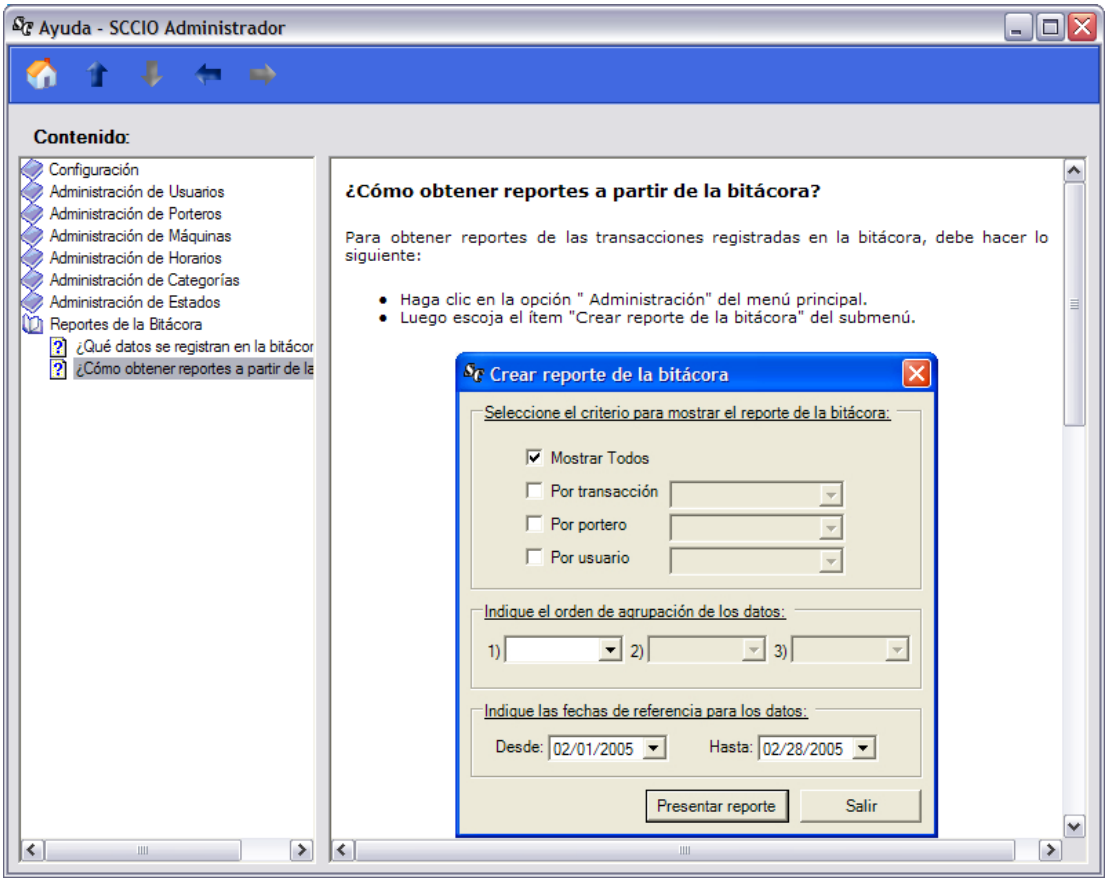


Figura 4.8.- Ventana de ayuda del sistema administrador.

En cuanto al principio de *flexibilidad*, su aplicación se observará en el uso de menús contextuales, de vistas (ver iconos, ver lista, y ver detalles), y en general, en las diferentes formas de acceso que el sistema proveerá para las funciones que vayan a ser realizadas. Por ejemplo, en el menú se incluye una opción para "modificar usuarios", a la que también se puede acceder haciendo doble clic en el icono representativo del usuario, cuyos datos se desea modificar; haciendo uso de la función "modificar" que se incluye en la

barra de tareas horizontal; o utilizando la opción "modificar" del menú contextual respectivo.

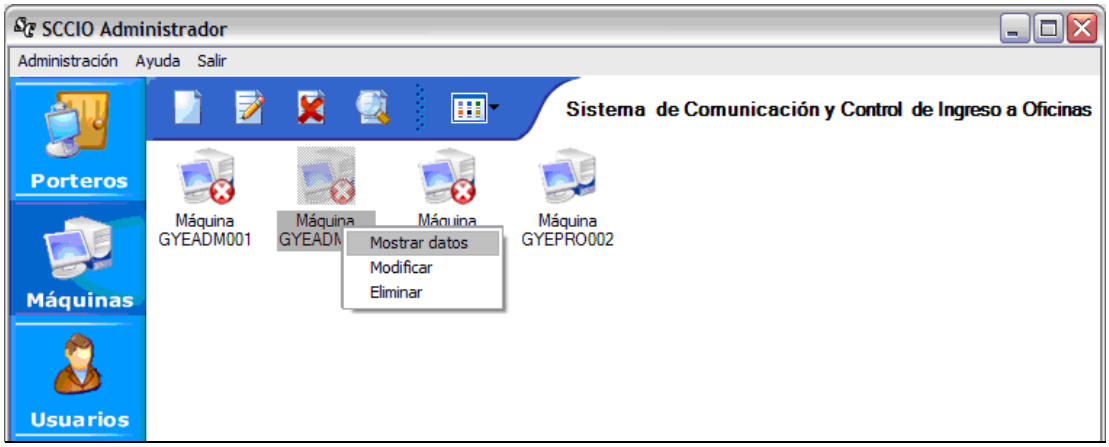


Figura 4.9.- Uso de menús contextuales.

Otros de los principios considerados en el diseño de este módulo, es el de *visibilidad*. Este principio deberá observarse en la distribución y disposición de las funciones dentro de la interfaz. Por ejemplo, es conveniente presentar agrupadas las funciones, de acuerdo al elemento del sistema con el que se relacionan (porteros, usuarios, máquinas, categorías, horarios, y estados), para que los usuarios sepan exactamente a dónde deben dirigirse, cuando quieran crear un usuario, eliminar un máquina, consultar los datos de un portero, etc.

En cuanto a la generación de reportes, la interfaz deberá proveer controles que faciliten al usuario la tarea de creación y visualización de resultados. Estos controles deberán evitar el ingreso de datos erróneos, limitando al usuario a seleccionar entre las posibles opciones, y brindando valores por defecto que le sirvan de guía.

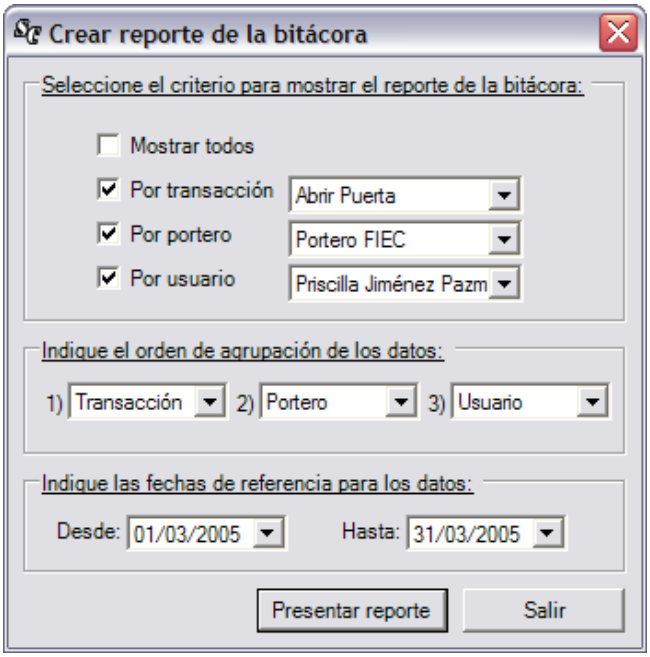


Figura 4.10.- Uso de controles que facilitan las tareas a los usuarios.

Finalmente, para facilitar la visualización del reporte de la bitácora, la interfaz proveerá controles que agiliten la navegación entre los datos, y la presentación de la información respectiva, incluyendo la imagen asociada a cada uno de los registros, tal como se muestra en la figura 4.11.

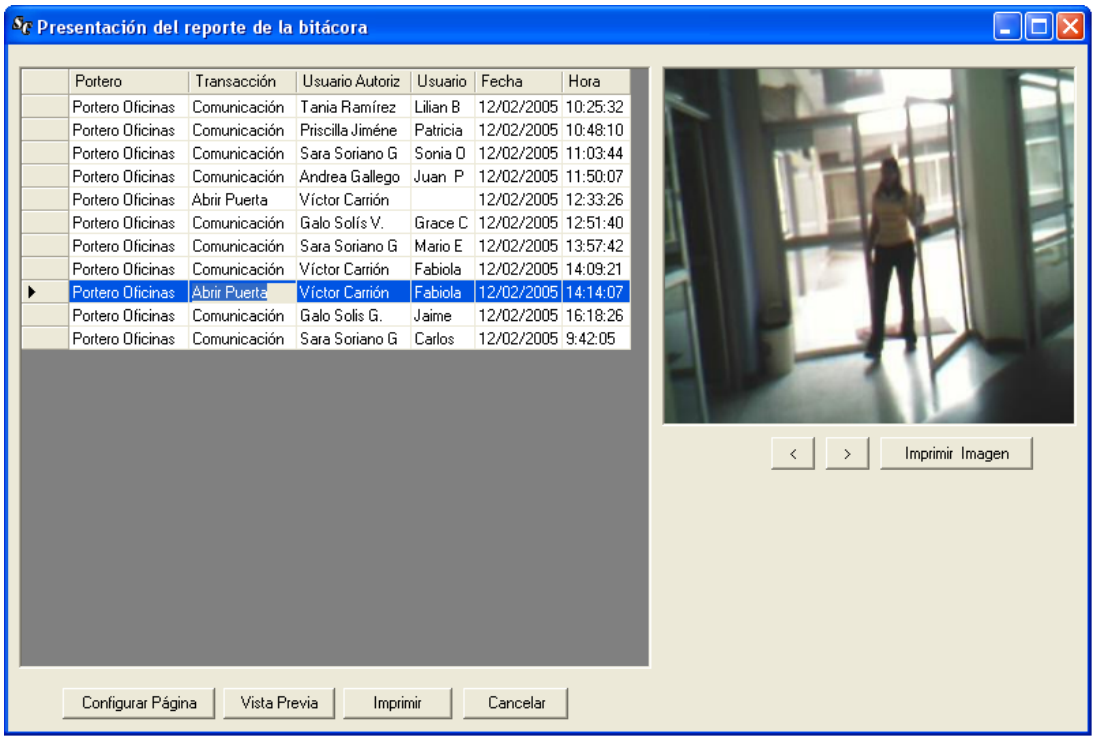


Figura 4.11.- Uso de controles que facilitan la navegación y presentación de datos en el reporte de la bitácora.

4.4.2. Interacción del sistema portero

La aplicación del principio de *consistencia* en el sistema portero se observará a través del flujo de ventanas, ya que en estas se mantendrán la misma distribución y presentación de las funciones (ir a la pantalla de inicio, abrir puerta, comunicación, dejar mensajes, etc.), a las que el usuario puede acceder, tal como se muestra en la figura 4.12.

Figura 4.12.- Distribución de las funciones en la interfaz del sistema portero, representadas por los botones.

Debido a que este sistema será utilizado incluso por personas que no hayan tenido experiencia en el uso de un computador, es importante aplicar el principio de *familiaridad*, principalmente en el proceso más complejo, que es el de grabación de mensajes de audio o vídeo. En este caso, las funciones que se provean deberán ser similares a las que presenta cualquier equipo diseñado para grabar y reproducir este tipo de contenido, como videograbadoras, equipos de sonido, etc., con el objetivo, de que los usuarios

puedan reconocer e intuir qué funciones utilizar y cómo hacerlo.

Figura 4.13.- Interfaz para la grabación de mensajes.

En el diseño de las interfaces de este sistema deberá hacerse hincapié en la aplicación del principio de *visibilidad*, de modo que deberán tomarse en cuenta factores como el tamaño de los botones y las letras, el color y la ubicación de los diferentes elementos que componen la interfaz, y la cantidad de información que se presente al usuario. Dado que la computadora que tenga instalado este sistema, no llevará a

cabo ninguna otra tarea que no sea la de ejecutar esta aplicación, la interfaz será diseñada para ocupar todo el tamaño de la pantalla.

Es importante mantener informado al usuario en todo momento acerca del estado en que se encuentra el sistema, y del resultado de las operaciones que se hayan realizado, a este principio se lo conoce como *retroalimentación,* y su aplicación en el sistema portero deberá observarse principalmente en el establecimiento de comunicaciones instantáneas. Cada vez que un usuario externo solicite establecer una comunicación con un usuario interno, deberá esperar un determinado tiempo hasta que éste acepte o rechace la solicitud, o hasta que se sobrepase el tiempo de espera por respuesta. Durante este lapso, el sistema portero le mostrará un mensaje, tal como se muestra en la figura 4.14.

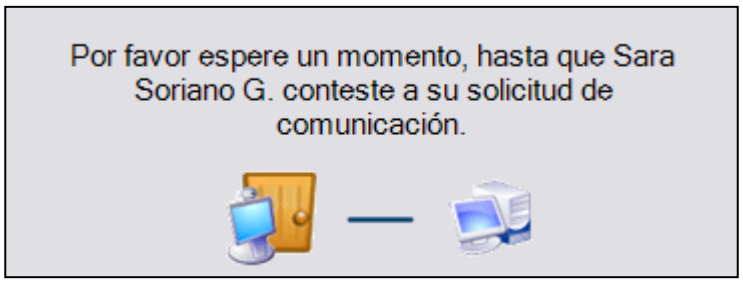


Figura 4.14.- Mensaje presentado a un usuario, mientras espera por una respuesta a su solicitud de comunicación.

4.4.3. Interacción del sistema usuario

El sistema usuario tiene diferentes partes en las que se aprovecha el principio de *familiaridad* para hacer más intuitivas las tareas que realiza el usuario. Por ejemplo, en el caso de la comunicación instantánea de texto, la interfaz dispone de los componentes básicos de cualquier sistema de mensajería instantánea, tal como se muestra en la figura 4.15.

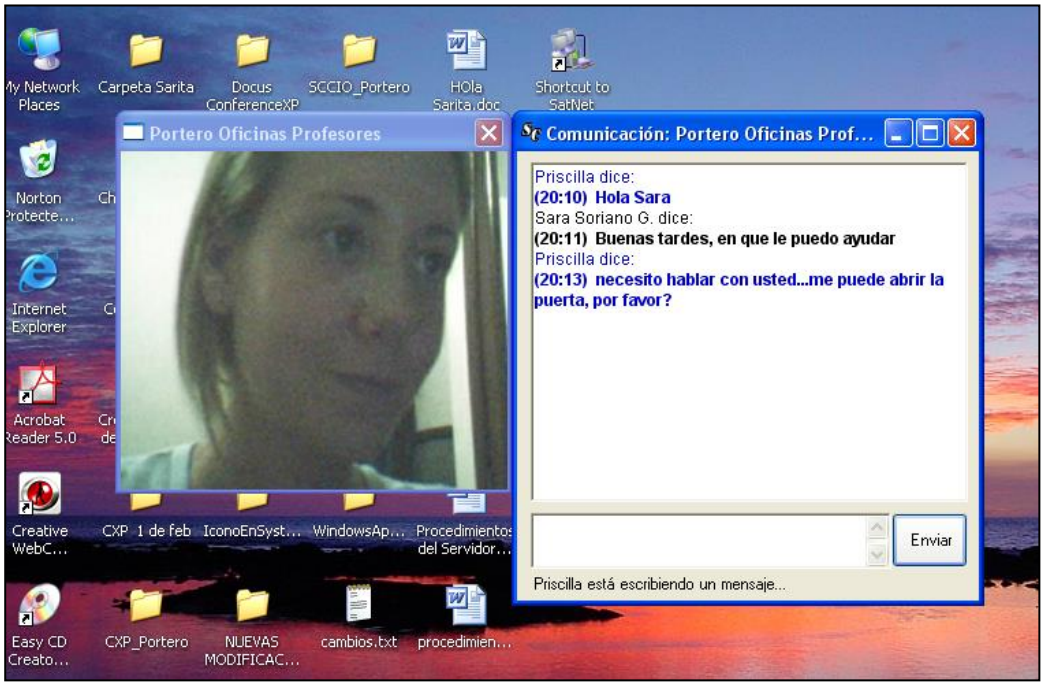


Figura 4.15.- Ventana usada para las comunicaciones instantáneas de texto.

Otra de las tareas típicas que se lleva a cabo en este tipo de sistemas, es el cambio, manual o automático, del estado de conexión. Se recurre al principio de familiaridad en este caso, al presentar, como suele hacerse, una lista de los estados disponibles, para que el usuario pueda seleccionar al que desea cambiarse.

Figura 4.16.- Interfaz con la lista de los estados de conexión disponibles.

Otro aspecto en el que se ve aplicado este principio, es en el uso de controles tab y de un menú, para distribuir de mejor manera la información y las funciones que se presentan al usuario. Esta también es una práctica común en el diseño de las interfaces de la mayoría de los sistemas.

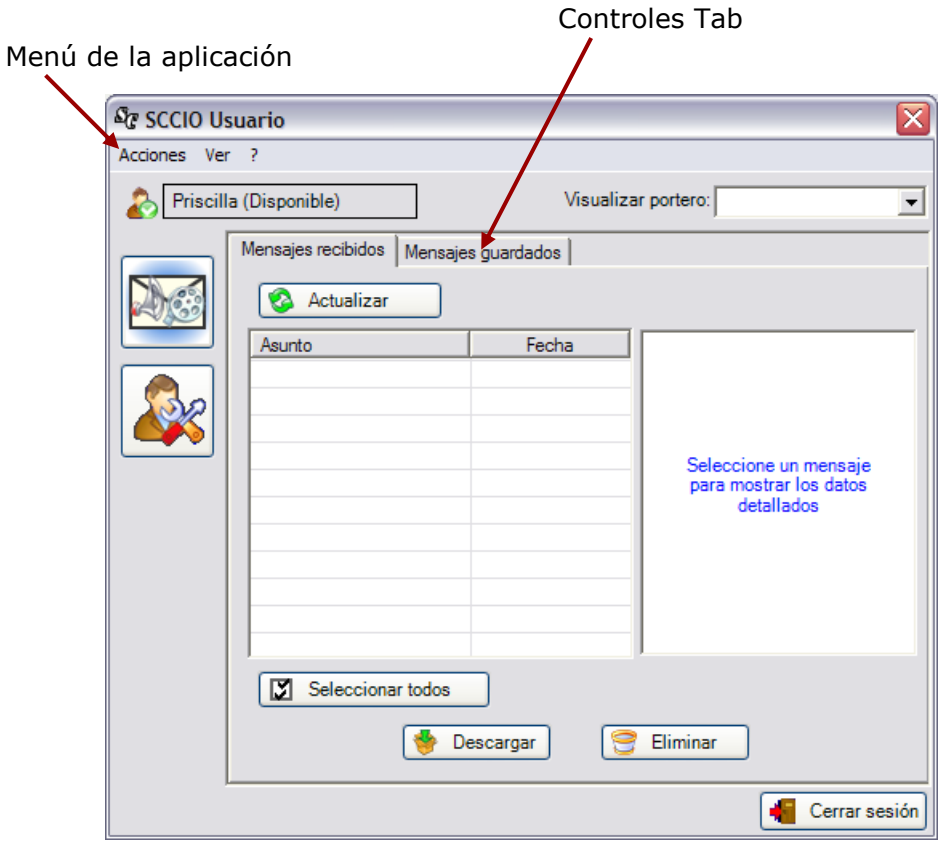


Figura 4.17.- Uso de controles comunes en la interfaz del sistema usuario

La aplicación del principio de *flexibilidad* se observa en el uso de un menú desde el que se puede acceder a todas las

funciones que ofrece este sistema; en el empleo de otros controles distribuidos en la interfaz que permiten acceder directamente a determinadas funciones; y en la utilización de un menú contextual para el icono de la aplicación que se coloca en la barra del sistema, y que permitirá realizar algunas de las funciones más frecuentes que se pueden llevar a cabo a través de la pantalla principal de la aplicación. Por ejemplo, en caso de que el usuario desee cambiar su estado de conexión manualmente, existen tres formas diferentes de hacerlo: utilizando el menú de la aplicación, accediendo al control que se provee en la interfaz, o haciendo uso del menú contextual mencionado anteriormente.

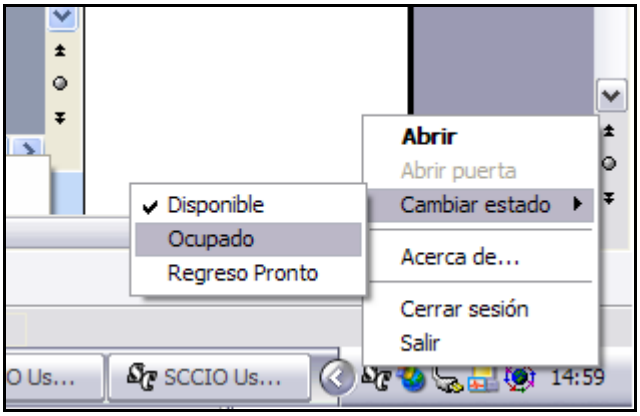


Figura 4.18.- Uso del menú contextual del icono mostrado en la barra del sistema.

El principio de *retroalimentación* se aplica principalmente en el cambio de estado de conexión, ya que se muestra el estado actual del usuario en respuesta al cambio efectuado, como se presenta en la figura 4.19.

Figura 4.19.- Principio de retroalimentación en el cambio de estado de conexión.

4.4.4. Interacción del sistema Web

Para aprovechar el principio de *familiaridad*, la interfaz de revisión de los mensajes recibidos deberá asemejarse en su formato, a las listas de mensajes que se presentan en los sistemas de correo electrónico. Por ejemplo, esto se observará en el uso de paginación para controlar la cantidad de mensajes que se presenten al usuario por página, en el empleo de una tabla para organizar los datos, en la adición de un CheckBox junto a cada mensaje que permita

seleccionarlos, y en el uso de links que permitan realizar las descargas.

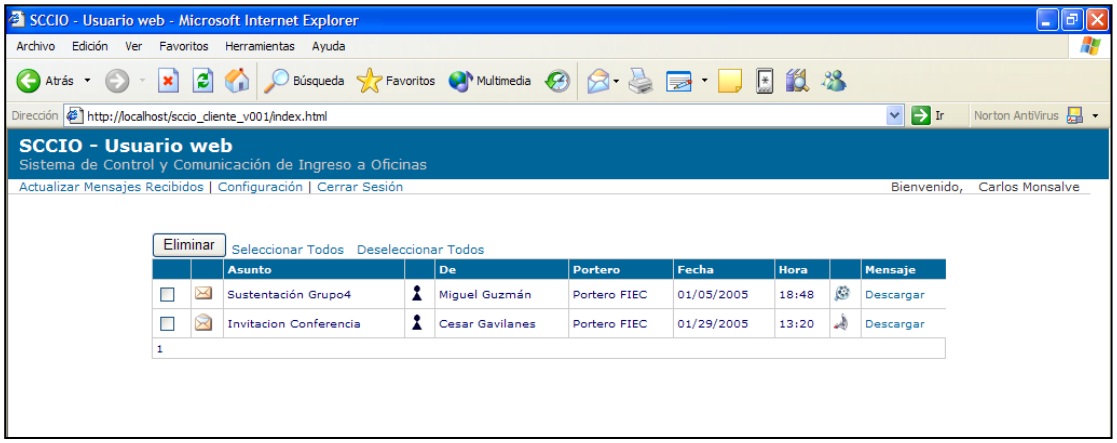


Figura 4.20.- Interfaz de revisión de mensajes recibidos

Adicionalmente, en la interfaz de configuración de datos del usuario, se utilizará un formulario que facilite el ingreso de la información y que suele emplearse en la mayoría de sistemas de este tipo. La aplicación del principio de *familiaridad* en las maneras que hemos mencionado, permitirá que cualquier usuario que tenga experiencia en el uso de Internet y de aplicaciones Web, pueda interactuar de manera intuitiva con el sistema.

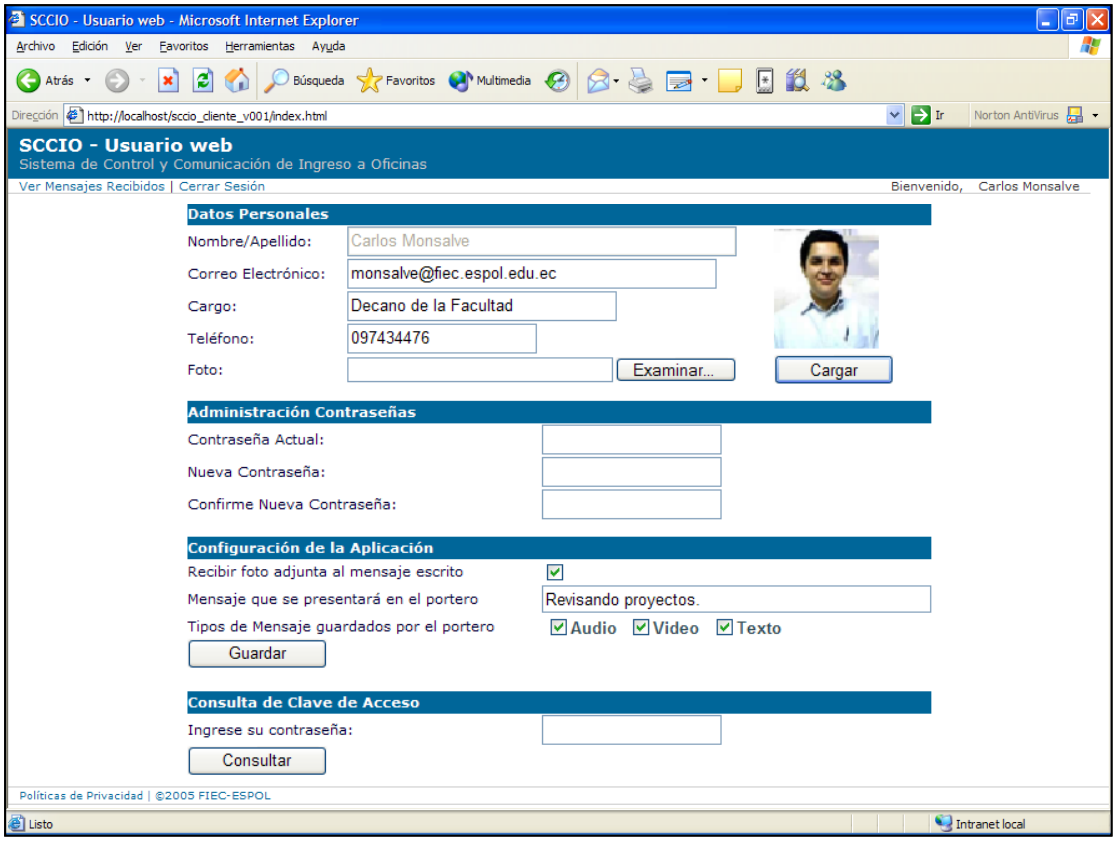


Figura 4.21.- Interfaz de configuración de datos del usuario

En todo momento, el usuario recibirá *retroalimentación* de las operaciones que realiza, ya que junto a cada botón de acción se presentará un mensaje que indique el resultado de éstas. Por ejemplo, cuando el usuario modifica los datos de su configuración, le aparecerá un mensaje indicando si la operación fue realizada con éxito, o si alguno de los datos ingresados era incorrecto, tal como se muestra en la figura 4.22.

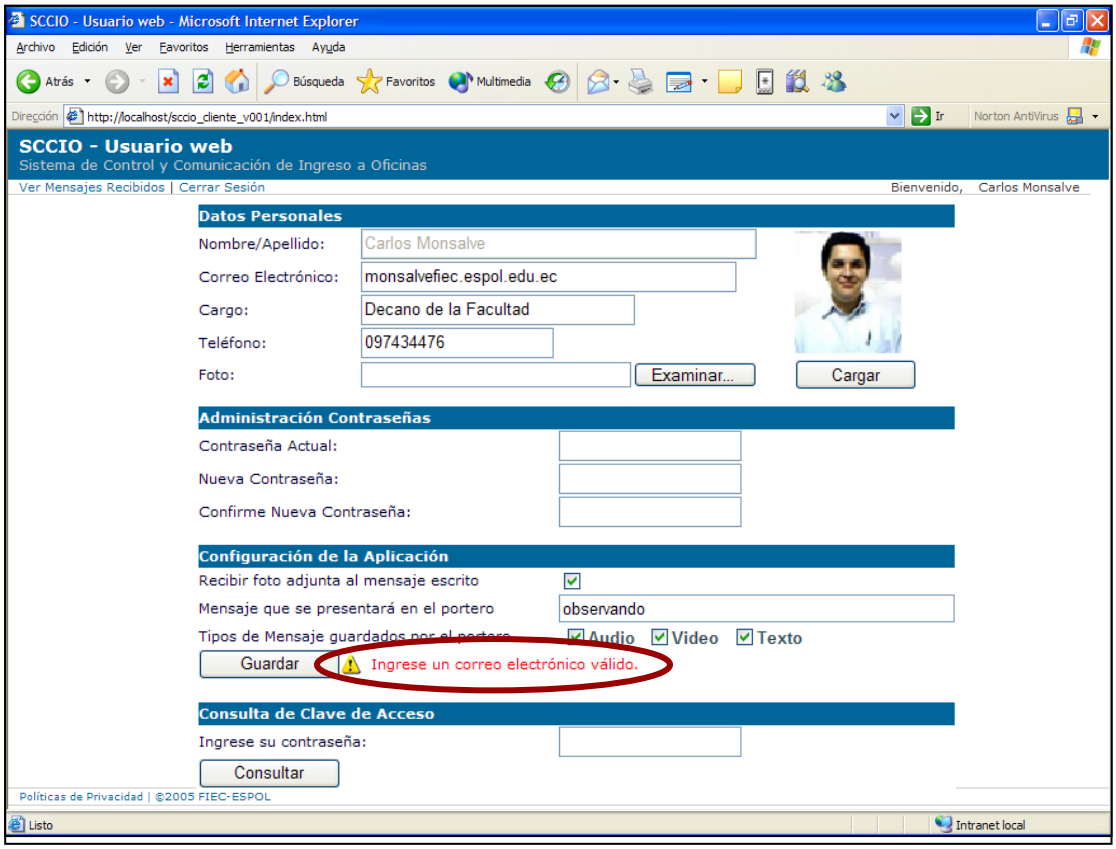


Figura 4.22.- Aplicación del principio de retroalimentación.

CAPÍTULO 5

5. IMPLEMENTACIÓN Y PRUEBAS

5.1. Herramientas utilizadas de los lenguajes de programación

Dentro de las herramientas provistas por el lenguaje de programación que tuvieron mayor importancia en la implementación del Sistema Computarizado de Comunicación y Control de Ingreso a Oficinas, se encuentran: la tecnología .Net Remoting, la clase Socket, y la arquitectura DirectShow; que se describirán a continuación.

.Net Remoting

La tecnología *.NET Remoting* se utiliza en escenarios de implementación distribuida. Permite crear fácilmente aplicaciones de cliente que utilicen objetos en otros procesos del mismo computador o en cualquier otro computador disponible en la red. A

continuación se describen algunos conceptos que se utilizan en el entorno remoto de .NET:

- *Canales*: Un canal es un tipo de objeto que toma una secuencia de datos, crea un paquete según un determinado protocolo de red y lo envía a otro computador. .NET Framework suministra dos tipos de canales: HTTP y TCP; tanto *TcpChannel* como *HttpChannel*, se pueden utilizar para enviar y recibir información. La clase *TcpChannel* representa a un canal combinado de *TcpServerChannel* y *TcpClientChannel*, y utiliza un formateador binario de manera predeterminada para serializar todos los mensajes en una secuencia binaria y transportar la secuencia al identificador uniforme de recursos (URI) de destino mediante el protocolo TCP. *TcpChannel* permite la comunicación entre el remitente y el destinatario mediante sockets, que se cierran en el cliente después de 15 a 20 segundos de inactividad. La clase *HttpChannel*, por otro lado, representa a un canal combinado de *HttpServerChannel* y *HttpClientChannel*, que transporta mensajes a objetos remotos utilizando el protocolo SOAP de manera predeterminada. Todos los mensajes salientes pasan por el formateador SOAP, donde se convierten en XML y se serializan. La secuencia de datos se

transporta entonces a la dirección URI utilizando el protocolo HTTP; y para los mensajes entrantes se invierte el proceso.

- *Formateadores*: Los formateadores de serialización .NET codifican y decodifican los mensajes entre las aplicaciones y los dominios de aplicación. .NET Runtime tiene dos formateadores nativos: Binario y SOAP. En realidad el formateador es un componente conectable de un canal y se puede utilizar un formateador personalizado para reemplazar el estándar XML (SOAP) o los formateadores binarios. En la serialización binaria, los campos públicos y privados del objeto, el nombre de la clase, y el ensamblado de ésta, se convierten en una secuencia de bytes que luego se escribe en una secuencia de datos. Cuando, después, el objeto se deserializa, se crea una copia exacta del objeto original. La serialización XML, por otra parte, convierte en una secuencia XML las propiedades y los campos públicos de un objeto, o los parámetros y valores devueltos de los métodos. Como XML es un estándar abierto, cualquier aplicación puede procesar la secuencia XML si es necesario, independientemente de la plataforma.

- *Archivos de configuración*: Normalmente, cada dominio de aplicación tiene un archivo de configuración propio. La principal

ventaja de usar este tipo de archivos es que la información de configuración está separada del código, y, en consecuencia, cuando haya necesidad de hacer cambios, se podrá recurrir al archivo de configuración, en lugar de volver a compilar el código fuente.

- *Activación del servidor*: Los objetos activados en el servidor son aquellos cuya duración está controlada directamente por éste. El dominio de aplicación del servidor crea estos objetos únicamente cuando el cliente llama a un método en el objeto, y no cuando llama a *new* o a *Activator.GetObject*; así se evita la necesidad de una acción de ida y vuelta por la red con el único propósito de crear una instancia. Hay dos modos de activación para los objetos activados por el servidor: *Singleton* y *SingleCall*. Los tipos *Singleton* nunca tienen más de una instancia al mismo tiempo; si existe una instancia, todas las solicitudes de clientes son atendidas por esa instancia; si no existe ninguna, el servidor crea una y todas las solicitudes de cliente posteriores serán atendidas por ella. Dado que los tipos *Singleton* tienen asociada una duración predeterminada, los clientes no siempre recibirán una referencia a la misma instancia de la clase utilizable de forma remota. Los tipos *SingleCall*, por otro lado, siempre tienen una instancia por cada solicitud de

cliente; de este modo, la siguiente llamada a un método será atendida por otra instancia del servidor, aunque el sistema no haya reciclado todavía la instancia anterior. Los tipos *SingleCall* no participan en el sistema de concesión del período de duración.

.NET Remoting permite reemplazar un protocolo de comunicación o un formato de serialización por otro sin tener que recompilar el cliente ni el servidor. Además, el sistema de interacción remota no utiliza ningún modelo de aplicación en particular, es decir, se puede comunicar desde una aplicación Web, una aplicación de consola, un servicio de Windows, etc.

Los servidores de interacción remota también pueden ser de cualquier tipo de dominio de aplicación. Si se desea utilizar .NET Remoting para crear una aplicación en la que dos componentes se comunican directamente más allá de los límites de los dominios de aplicación, sólo se deberá crear lo siguiente: un objeto que se puede utilizar de forma remota, un dominio de aplicación host para escuchar las solicitudes de dicho objeto, y un dominio de aplicación de cliente que realice solicitudes para dicho objeto.

El uso de .NET Remoting en la implementación del sistema SCCIO, se centra en la comunicación entre las aplicaciones clientes y el servidor. La creación de una aplicación que utilice .NET Remoting para comunicarse más allá de los límites de los dominios de aplicación es muy sencilla, y los elementos que se necesitan son tres: un tipo de objeto utilizable de manera remota, un dominio de aplicación de escucha o host, y un dominio de aplicación cliente que realiza la llamada. Este proceso se aplica independientemente de la complejidad del escenario remoto, y a continuación se explicará la implementación de este esquema en el desarrollo del sistema SCCIO.

Como se mencionó anteriormente, es necesario disponer de un tipo de objeto de uso remoto, si se requiere que los objetos de otros dominios de aplicación puedan utilizar instancias de alguna clase específica, y para esto la clase debe heredarse de *MarshalByRefObject.* En el siguiente ejemplo de código se muestra un fragmento de la clase *Servidor*; ésta contiene todos los métodos que los clientes pueden acceder, desde diferentes interfaces (*IAdministrador, IUsuario, IPortero, IUsuarioWeb*).

```
//Servidor.cs
using System;
using System.Data;
using System.Data.Odbc;

public class Servidor: MarshalByRefObject, IAdministrador.Interfaz,
IUsuario.Interfaz, IPortero.Interfaz, IUsuarioWeb.Interfaz
{   …
    public string[] crearUsuario(string nombre, string cargo, …)
    {  …
        Respuesta=Usuarios.crearUsuario(nombre,cargo,…);
        …
        return Respuesta;
    }
    …
}
```

Las interfaces presentan, a manera de menú, los diferentes procedimientos y funciones del servidor, a los que pueden tener acceso las aplicaciones cliente; y para utilizarlas con ese propósito es necesario que la clase *Servidor* las implemente, tal como se muestra en el código anterior.

El segundo paso en la aplicación del esquema mencionado al inicio consistía en la generación de una aplicación host. Para que los objetos de otros dominios de aplicación puedan crear instancias de la clase *Servidor* de manera remota, es preciso generar una aplicación host o de escucha con el fin de realizar dos acciones: elegir y establecer un canal, y registrar el modo que éste puede utilizar para escuchar las solicitudes de dicho tipo. Dado que la configuración remota se realiza por cada dominio de aplicación, éste debe ejecutarse para poder escuchar las solicitudes. El

siguiente código muestra la manera en que ha sido implementado el dominio de aplicación host, *Servidor,* en el sistema SCCIO, utilizando un archivo de configuración.

```
//LevantarServidor.cs
public class LevantarServidor
{
  public LevantarServidor()
  {
    string rutaArchivoConfig=ruta del archivo de configuración
    RemotingConfiguration.Configure(rutaArchivoConfig);
  }
}
```

La clase *LevantaServidor* debe obtener la ruta del archivo que contiene la información de configuración de la clase *Servidor*; este archivo se encuentra por lo general en el mismo directorio que el ejecutable, y a continuación, se muestra la información que contiene.

```
<configuration>
  <system.runtime.remoting>
    <application>
      <service>
        <wellknown
          type="SCCIOServidor.Servidor,SCCIOServidor.Servidor"
          mode="Singleton"
          objectUri="SCCIO"/>
      </service>
      <lifetime leaseTime="0M"
            sponsorshipTimeOut="5M"
            renewOnCallTime="5M"
            leaseManagerPollTime="2M"/>
      <channels>
        <channel ref="tcp" port="8085">
          <serverProviders>
                <formatter ref="binary"/>
          </serverProviders>
        </channel>
      </channels>
    </application>
  </system.runtime.remoting>
</configuration>
```

El sistema de interacción remota utiliza la información de este archivo para escuchar y enrutar las solicitudes hacia una instancia de un objeto de tipo de uso remoto. En el archivo se especifica el modo de activación de servidor (*Singleton)*, el nombre de tipo y el ensamblado del tipo en cuyo nombre debe escucharse (*SCCIOServidor.Servidor*), así como el identificador URI (Identificador uniforme de recursos) o el nombre externo del objeto (*SCCIO*). En el archivo se indica asimismo al sistema de interacción remota que escuche las solicitudes en el puerto 8085 mediante el *TcpChannel* proporcionado por el sistema.

Por otro lado, el elemento <lifetime>, que se observa en el archivo de configuración, contiene información sobre el período de duración del objeto. Los objetos MBR (Marshal-By-Reference, cálculo por referencia) no residen en la memoria eternamente, a no ser que se reemplacen sus directivas referentes a la duración; de lo contrario, cada objeto MBR tendrá una duración controlada por una combinación de concesiones, un administrador de concesiones y una serie de patrocinadores. Una concesión es el período de tiempo que un determinado objeto estará activo en la memoria antes de que el sistema .NET Remoting comience el proceso para eliminarlo y reclamar la memoria.

El administrador de concesiones del dominio de aplicación de servidor es el objeto que determina cuándo el objeto remoto debe ser reclamado por el recolector de elementos no utilizados. Un patrocinador es un objeto que solicita una nueva concesión para un determinado objeto, para lo que se registra él mismo en el administrador de concesiones.

Siempre que se utiliza de forma remota un objeto MBR fuera de su dominio de aplicación, se crea una concesión de período de duración para ese objeto. Puesto que la vida útil de un objeto remoto es independiente de la de sus clientes, la concesión para un objeto sencillo o pequeño puede ser muy larga, la pueden utilizar varios clientes y la puede renovar periódicamente un administrador o un cliente.

Este enfoque utiliza las concesiones de manera eficaz, porque se necesita muy poco tráfico en la red para la recolección distribuida de elementos no utilizados. Sin embargo, los objetos remotos que utilizan muchos recursos pueden tener una concesión para un período de duración breve, que el cliente renueva frecuentemente a intervalos cortos. Cuando todos los clientes han terminado con el objeto remoto, el sistema .NET Remoting elimina rápidamente el objeto; con esta táctica, se utilizan de manera más eficaz los

recursos del servidor. Las propiedades de una concesión de período de duración se pueden modificar especificando que las instancias del tipo correspondiente tengan un período de duración infinito, en el elemento <*lifetime*> del archivo de configuración.

Esto es justamente lo que se hace en el sistema SCCIO, como se puede observar en la porción de código anterior, al configurar el atributo *leaseTime* en 0. En la siguiente tabla, se describen brevemente las funciones de los atributos del elemento <*lifetime*> que se incluyen en el archivo de configuración.

Atributo	Descripción
leaseTime	Especifica el tiempo de concesión de la aplicación. El valor predeterminado es 5 minutos. El valor cero configura la concesión con un período de duración infinito.
sponsorshipTimeout	Especifica el tiempo que aguarda el administrador de concesiones la respuesta del patrocinador cuando se ha notificado el vencimiento de una concesión. Si el patrocinador no responde en el tiempo especificado, el recolector de elementos no utilizados eliminará el objeto remoto. El valor predeterminado es 2 minutos.
renewOnCallTime	Especifica la medida en que se extiende el tiempo de concesión con cada llamada a función en el objeto. El valor predeterminado es 2 minutos.
leaseManagerPollTime	Especifica el tiempo durante el cual permanece inactivado el administrador de concesiones tras comprobar la presencia de concesiones vencidas. El valor predeterminado es 10 segundos.

Tabla 5.1.- Atributos del elemento <lifetime>

El tercer y último paso en la implementación del esquema consistía en la generación de una aplicación cliente del tipo remoto *Servidor*, albergado por la clase *LevantaServidor*. Para esto, se registra a la aplicación como cliente de dicho objeto y, luego se lo invoca como si éste estuviese dentro de su propio dominio. El sistema .NET Remoting interceptará las llamadas al cliente, las enviará al objeto remoto, y devolverá los resultados al cliente. En el siguiente ejemplo se muestra la porción de código del sistema administrador que permite acceder al método *crearUsuario* de la clase *Servidor,* por medio de la interfaz *IAdministrador*.

```
/*devuelve un objeto de tipo IAdministrador.Interfaz, que permite acceder a
una porción de los métodos que ofrece la clase Servidor*/
public static IAdministrador.Interfaz ConexionServidor()
{
    ...
    if (canal=="tcp")
    {
        chanTcp = new TcpChannel();
        ChannelServices.RegisterChannel(chanTcp);
    }
    if (canal=="http")
    {
         chanHttp= new HttpChannel();
         ChannelServices.RegisterChannel(chanHttp);
     }
    dirServidor=obtener la dirección IP del servidor
    System.Object o = System.Activator.GetObject (typeof
     (IAdministrador.Interfaz),
    canal+"://"+dirServidor+":"+numPuerto +"/SCCIO");
    IAdministrador.Interfaz obj= (IAdministrador.Interfaz) o;
    return obj;
}
...
//se accede al método "crearUsuario" del servidor
ConexionServidor().crearUsuario(nombre, cargo,…);
```

Como se puede ver en este fragmento de código, una vez que se obtiene la información de configuración necesaria (tipo de canal, dirección IP del servidor, número de puerto, y URI), ya es posible establecer la conexión con el servidor y acceder a una parte de los métodos que éste ofrece, por medio de la interfaz *IAdministrador.Interfaz*. La información de configuración indica al sistema remoto, de manera resumida, que el cliente debe intentar crear y utilizar un objeto de tipo *Servidor* (porque este objeto implementa la interfaz utilizada), ubicado en *tcp://192.168.0.3:8085/SCCIO*, por ejemplo.

Clase Socket de .NET Framework

Las implementaciones de sockets proporcionan un extenso conjunto de métodos y propiedades para las comunicaciones en red. La clase Socket de .NET Framework es una versión de código administrado de los servicios de socket que proporciona la API de Winsock32. Admite dos modos básicos: sincrónico y asincrónico.

En el modo sincrónico, las llamadas a funciones que realizan operaciones de red esperan a que la operación concluya antes de devolver el control al programa desde el que se realiza la llamada. Por esta razón, no resultan apropiados para aplicaciones que usan

la red de forma intensiva en su funcionamiento, pero sí para aplicaciones más sencillas.

En el modo asincrónico, por otro lado, estas llamadas devuelven el control inmediatamente. Antes de poder usar un socket para establecer comunicación con dispositivos remotos, debe inicializarse con información sobre la dirección de red y el protocolo. El constructor de la clase Socket tiene parámetros que especifican la familia de direcciones, el tipo de socket, y el tipo de protocolo que se utiliza para realizar conexiones. Por medio de la siguiente porción de código se crean los sockets empleados en el establecimiento de comunicaciones entre el sistema usuario y el sistema portero; estos sockets se pueden utilizar en una red basada en TCP/IP.

```
Socket listener = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp );
```

La enumeración *AddressFamily* especifica las familias de direcciones estándar que utiliza la clase Socket para resolver direcciones de red (por ejemplo, el miembro *AddressFamily.InterNetwork* especifica la familia de direcciones de IP versión 4). La enumeración *SocketType* especifica el tipo de socket (por ejemplo, el miembro *SocketType.Stream* establece un socket estándar para enviar y recibir datos con control de flujo). La enumeración *ProtocolType*

indica el protocolo de red que hay que utilizar al establecer comunicación en el Socket (por ejemplo, *ProtocolType.Tcp* indica que el socket utiliza TCP; *ProtocolType.Udp* indica que el socket utiliza UDP). Una vez que el socket haya sido creado, se puede iniciar la conexión con el otro extremo o recibir conexiones de dispositivos remotos.

Los sockets de servidor o de escucha abren un puerto en la red y después esperan a que un cliente se conecte con ese puerto. La dirección única de un servicio de TCP/IP se define mediante la combinación de la dirección IP del host con el número de puerto para crear un extremo para el servicio. A continuación, se presenta una porción del código que muestra cómo se crea un *IPEndPoint* combinando la primera dirección IP devuelta por *Dns* para el equipo host, con un número de puerto elegido entre el intervalo de números de puerto registrados, en este caso, 11000.

```
IPHostEntry ipHostInfo = Dns.Resolve(Dns.GetHostName());
IPAddress ipAddress = ipHostInfo.AddressList[0];
IPEndPoint localEndPoint = new IPEndPoint(ipAddress, 11000);
```

La clase *Dns* proporciona métodos que devuelven información acerca de las direcciones de red admitidas por el dispositivo de red local. Cuando el dispositivo de red local tiene más de una dirección de red, o si el sistema local admite más de un dispositivo de red, la

clase *Dns* devuelve información acerca de todas las direcciones de red, y la aplicación debe elegir la dirección correcta para el servicio. *Internet Assigned Numbers Authority (IANA)* define los números de puerto para servicios comunes; otros servicios pueden tener números de puerto registrados en el intervalo de 1,024 a 65,535.

Una vez determinado el extremo local, el socket es asociado al extremo mediante el método *Bind* y se establece "en escucha" mediante el método *Listen*. *Bind* inicia una excepción si la combinación específica de dirección y puerto ya se está utilizando. En el siguiente fragmento de código se muestra cómo se asocia el socket al *IPEndPoint* creado anteriormente.

```
listener.Bind(localEndPoint);
listener.Listen(32);
```

El método *Listen* recibe un único parámetro, que especifica cuántas conexiones más se permiten al socket antes de que se devuelva un error de servidor ocupado al cliente que se conecta. En este ejemplo, se colocan hasta 32 clientes en la cola de conexión antes de que se devuelva una respuesta de servidor ocupado al cliente 33. En el caso del sistema SCCIO, debido a que el número de conexiones simultáneas depende de la cantidad de porteros que se encuentren conectados en un momento dado, también se optó por utilizar un valor máximo fijo de 32. Para esto, se tomó en

consideración que es muy poco probable que a un usuario le lleguen al mismo tiempo, tal cantidad de solicitudes de comunicación por parte de los sistemas porteros, pero si llegara a suceder, el sistema únicamente responderá que se encuentra ocupado, tal como se mencionó en el ejemplo anterior.

Los sockets que se utilizan en los dos extremos de comunicación del sistema son de tipo asincrónico. Después de que el socket se establece en escucha en un extremo, ya puede empezar a aceptar solicitudes de conexión entrantes mediante los métodos *BeginAccept* y *EndAccept*. Un socket de servidor asincrónico requiere un método para comenzar la aceptación de solicitudes de conexión de la red, un método de respuesta para controlar las solicitudes de conexión y comenzar la recepción de datos de la red, y un método de respuesta para finalizar la recepción de los datos.

Los fragmentos de código que se presentarán a continuación, muestran cómo se da comienzo a la aceptación de solicitudes de conexión, tanto en el caso del sistema portero como del sistema usuario. El método *atenderMensajesEntrantes* inicializa el socket y después utiliza el método *BeginAccept* para comenzar la aceptación de nuevas conexiones. Cuando se recibe una solicitud de conexión nueva en el socket, se llama al método de respuesta a la

aceptación. Este método se encarga de obtener la instancia de socket que controlará la conexión y de entregar ese socket al subproceso que se hará cargo de la solicitud. Permite indicar al subproceso de aplicación principal que continúe el procesamiento, establecer la conexión, y comenzar la lectura asincrónica de datos del cliente. El método de respuesta a la aceptación implementa el delegado de *AsyncCallback*, no devuelve ningún valor, y toma un único parámetro de tipo *IAsyncResult*.

A continuación, se describe cómo se ha implementado el método de respuesta a la aceptación o *AcceptCallback*, en el sistema.

```
public void AcceptCallback(IAsyncResult ar)
{
    // Indicar al hilo principal que puede continuar
    allDone.Set();

    // Obtener el socket que maneja los requerimientos del cliente
    Socket listener = (Socket) ar.AsyncState;
    Socket handler = listener.EndAccept(ar);

    // Crear el objeto estado
    StateObject state = new StateObject();
    state.workSocket = handler;
    handler.BeginReceive( state.buffer, 0, StateObject.BufferSize,
    0,new AsyncCallback(ReadCallback), state);
}
```

El método *BeginAccept*, que es llamado dentro del procedimiento *atenderMensajesEntrantes*, toma dos parámetros, un delegado de *AsyncCallback* que señala al método de respuesta a la aceptación, y un objeto que se utiliza para proveer información de estado al

método de respuesta; en este caso, se pasa el socket de escucha, como el segundo parámetro.

La porción de código que se muestra a continuación describe cómo se ha implementado el método *atenderMensajesEntrantes* en el sistema.

```
public void atenderMensajesEntrantes()
{
  byte[] bytes = new Byte[1024];
  IPHostEntry ipHostInfo =     Dns.Resolve(Dns.GetHostName());
  IPAddress ipAddress = ipHostInfo.AddressList[0];
  IPEndPoint localEndPoint = new IPEndPoint(ipAddress,
  puertoEntrada);
  Socket listener = new Socket(AddressFamily.InterNetwork,
  SocketType.Stream, ProtocolType.Tcp );
  listener.Bind(localEndPoint);
  listener.Listen(32);
  while (true)
  {
   allDone.Reset();
   listener.BeginAccept(new AsyncCallback(AcceptCallback),listener);
   // Esperar hasta que la conexión esté hecha, antes de continuar
   allDone.WaitOne();
  }
}
```

La lectura de datos de un socket de cliente requiere un objeto de estado que pase valores entre llamadas asincrónicas, y en el siguiente fragmento de código se muestra la implementación de este objeto, que se ha creado para recibir una cadena del cliente remoto.

```
public class StateObject
{
  // Socket cliente
  public Socket workSocket = null;
  // Tamaño del buffer de recepción
  public const int BufferSize = 1024;
  // buffer de recepción
  public byte[] buffer = new byte[BufferSize];
  // String de datos recibidos
  public StringBuilder sb = new StringBuilder();
}
```

Contiene campos para el socket de cliente, un búfer para los datos que se reciben y un *StringBuilder* para crear la cadena de datos que envía el cliente. La colocación de estos campos en el objeto de estado permite que se conserven los valores a lo largo de varias llamadas para leer datos del socket de cliente. La porción de código que se presentó anteriormente y que describía la implementación del método *acceptCallback*, encargado de dar comienzo a la recepción de datos del socket de cliente, primero inicializa una instancia de la clase *StateObject* y después llama al método *BeginReceive* para comenzar la lectura de los datos de forma asincrónica.

El método final que hay que implementar para el servidor de socket asincrónico es el método de respuesta a la lectura que devuelve los datos enviados por el cliente. Este método lee uno o más bytes del socket de cliente en el búfer de datos y después vuelve a llamar al método *BeginReceive* hasta que se completen los datos enviados

por el cliente. Una vez que se ha obtenido el mensaje entero, se lo pasa al procedimiento *nuevoMensaje* para que se encargue de darle el tratamiento correspondiente, y se cierra el socket de servidor que controla la conexión con el cliente.

```
public void ReadCallback(IAsyncResult ar)
{
  string mensaje = String.Empty;
  StateObject estado = (StateObject) ar.AsyncState;
  Socket handler = estado.workSocket;
   // leer datos del socket del cliente
  int bytesRead = handler.EndReceive(ar);
  if (bytesRead > 0)
  { // la información entrante se va almacenando
   estado.sb.Append(Encoding.Unicode.GetString(state.buffer,0,
   bytesRead));
   mensaje = estado.sb.ToString();
   // chequear el EOM. Si se encuentra, el mensaje está completo
   if (mensaje.IndexOf("|EOM") > -1)
      nuevoMensaje(content);
   else
   { // Si no se encuentra, continuar obteniendo datos
      handler.BeginReceive(estado.buffer, 0, StateObject.BufferSize, 0,
      new AsyncCallback(ReadCallback), state);
    }
  }
}
```

Arquitectura DirectShow

DirectShow es un API que permite a aplicaciones de Windows controlar una gran variedad de dispositivos de captura de audio y vídeo; y que está integrado con la tecnología Microsoft DirectX. Detecta automáticamente el uso de aceleradores de hardware de audio y vídeo; y utiliza filtros para manejar y manipular datos multimedia. DirectShow provee varios filtros para soportar vídeo digital (DV):

- MSDV Driver.- Representa a los dispositivos de captura de vídeo digital.
- DV Splitter.- Divide un flujo intercalado de vídeo digital en un flujo de audio y uno de vídeo.
- DV Video Decoder.- Decodifica vídeo digital en vídeo descomprimido.
- DV Video Encoder.- Codifica vídeo descomprimido en vídeo digital.
- DV Muxer.- Combina audio y vídeo digital en un mismo flujo intercalado.

El DV Splitter y el DV Video Decoder trabajan juntos. El DV Splitter toma el flujo intercalado y los separa en un flujo de audio y otro de vídeo. El DV Video Decoder convierte el vídeo digital en un vídeo descomprimido, tal como se muestra en la figura 5.1.

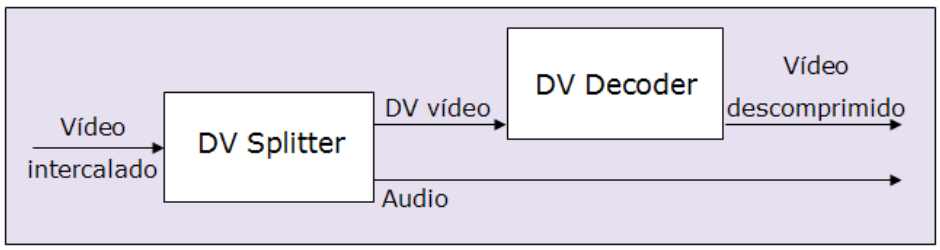


Figura 5.1.- Funciones del DV Splitter y DV Decoder

El DV Video Encoder y el DV Muxer invierten el proceso: el decodificador convierte el vídeo descomprimido a vídeo digital, y el

DV Mux combina el audio y el vídeo para crear un simple flujo intercalado, como se muestra en la figura 5.2.

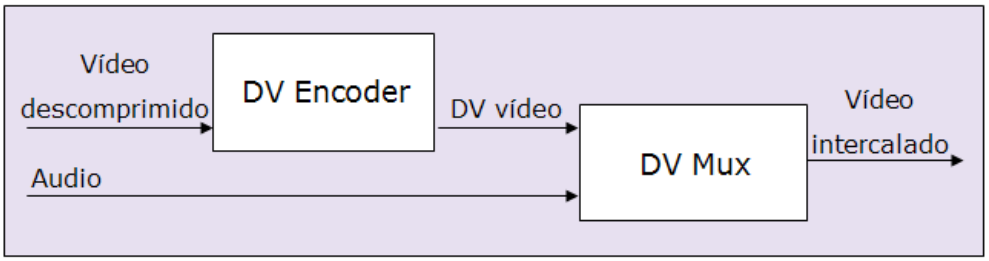


Figura 5.2.- Funciones del DV Encoder y DV Mux

Un filtro podrá desempeñar acciones como codificar, decodificar, dar formato, dividir flujo de datos multimedia, entre otros. Esto permite que la salida de un filtro vaya directamente a la entrada de otro.

Por otro lado, DirectShow soporta una gran variedad de formatos de archivos multimedia, tales como ASF, MPEG, AVI, MP3, WAV, etc.; y de compresores como Microsoft Windows Media Video codec versión 7.0, ISO MPEG-4 video versión 1.0, Microsoft MPEG-4 versión 3, Cinepak, entre otros.

Microsoft ha especificado el formato para almacenamiento de vídeo digital en archivos AVI; de esta manera, se asegura que sean compatibles con futuras versiones de la arquitectura de vídeo

digital Microsoft DirectShow para la plataforma de Microsoft Windows.

El sistema SCCIO Portero utiliza las librerías DShowNET.dll, DirectX.Capture.dll, que están basadas en DirectShow para capturar, grabar y reproducir archivos de audio (.wav) y vídeo (.avi). DirectX.Capture.dll provee funciones que permiten capturar el vídeo y audio de los dispositivos de entrada respectivos, encapsulando y facilitando la tarea de especificar qué dispositivos se van a utilizar, qué compresores de audio y vídeo, y el nombre del archivo .avi o .wav, que contendrá el respectivo contenido multimedia, tal como se muestra a continuación.

```
private Capture capture = null;
private Filters filters = new Filters();
Capture = new Capture (filters.VideoInputDevices[i],
                filters.AudioInputDevices[i]);
capture.VideoCompressor = filters.VideoCompressors[i];
capture.AudioCompressor = filters.AudioCompressors[i];
capture.Filename = "mensajeTmp.avi";
capture.PreviewWindow = PanelCaptura;
capture.Start(); // Inicia la grabación del contenido multimedia.
capture.Stop(); // Detiene la grabación.
```

El objeto *capture* es utilizado para capturar tanto audio como vídeo; en caso de que se desee capturar estos dos tipos de contenido, se deberán especificar los dispositivos de audio y vídeo a utilizar; y, en caso, de que se desee trabajar sólo con audio, se deberá colocar *null* en el primer argumento del constructor. El

objeto *filters*, por otro lado, es el que encapsula los filtros disponibles, y el índice "i" de los arreglos correspondientes representa a un filtro específico.

En el proceso de instalación del sistema SCCIO Portero el administrador deberá indicar, los tipos de dispositivos de captura de vídeo y compresores que vayan a utilizarse. Estos parámetros serán especificados en un archivo; la aplicación leerá los nombres de los diferentes dispositivos de captura de vídeo y los compresores, y luego, por medio de una función los buscará por su nombre en el arreglo correspondiente de cada filtro, y devolverá el índice "i" respectivo. Como último parámetro se debe especificar el objeto donde se mostrará el vídeo capturado, que será *PanelCaptura*, siempre y cuando el contenido a presentar sea vídeo. En caso de que se trate sólo de audio, deberá asignarse a este objeto el valor de *null.* Para poder iniciar la grabación del contenido multimedia deberán estar configurados los parámetros anteriormente mencionados. Se utilizarán los métodos *Start* y *Stop* para iniciar la grabación y detenerla, respectivamente.

Por otro lado, también se utiliza una clase denominada *Captura*, que permite obtener de manera sencilla imágenes instantáneas, tal como se muestra a continuación.

```
Image imagen = Captura.GetImage (índice del dispositivo de captura de
                                                    vídeo);
```

5.2. Descripción de procedimientos principales

A continuación se describen los procedimientos que llevan a cabo las tareas más relevantes dentro del sistema SCCIO.

Servidor

- *Verificación de espacio en la creación de usuarios*

  Al momento de crear un nuevo usuario se llevan a cabo algunas verificaciones, como por ejemplo, que haya suficiente espacio libre en la unidad correspondiente. Esto permite asegurar que el usuario, una vez creado, pueda disponer de la totalidad del espacio que se le haya asignado para su directorio de mensajes recibidos. El código que ejecuta el servidor para este proceso de verificación se describe a continuación.

```
public bool hayEspacioParaNuevoUsuario(bool tieneOficina)
{
  bool hayEspacioEnUnidad=false;
  long tamañoNuevoDir=0;
  long tamañoNuevoDir=
        obtenerEspacioEnUnidadAsignado(tieneOficina);
  long espacioLibreEnUnidad=obtenerEspacioLibreEnUnidad();
  long espacioLibreEnDirs=obtenerEspacioLibreEnDirectorios()
                              +tamañoNuevoDir;
  long espacioMinimo=espacio mínimo configurado;
  long espacioLibreTotal=espacioLibreEnUnidad-espacioLibreEnDirs;
  if(espacioLibreTotal<espacioMinimo)
    hayEspacioEnUnidad =false;
  else
    hayEspacioEnUnidad =true;
  return hayEspacioEnUnidad;
}
```

La función *hayEspacioParaNuevoUsuario* recibe como parámetro un dato que permite determinar si el usuario que ha de crearse dispondrá o no de una oficina. Este dato es necesario, porque de él dependerá el espacio asignado al directorio de mensajes del usuario, que se guardará en la variable *tamañoNuevoDir*.

Otro de los datos que debe obtenerse es el espacio disponible en la unidad correspondiente, y para esto se llama a la función *obtenerEspacioLibreEnUnidad*.

Adicionalmente, se necesita conocer el espacio libre que tienen en total los directorios de los usuarios que ya existen en el sistema. Este dato es importante, porque si en la configuración consta, por ejemplo, que los usuarios con oficina deber tener acceso a 100 MB de espacio para su carpeta de mensajes, y uno de los usuarios existentes ha ocupado apenas 20 MB, los 80 MB restantes de su directorio no se considerarán como espacio libre en la unidad, sino como parte de lo ocupado, y deberán restarse del espacio libre total.

La suma de los valores correspondientes al espacio disponible en cada directorio de usuario, se obtiene a partir de la función

obtenerEspacioLibreEnDirectorios y se almacena en la variable *espacioLibreEnDirs*.

Finalmente, el último dato que hace falta es el espacio mínimo configurado, que representa el espacio libre, requerido como mínimo, para la unidad. Una vez que se han obtenido todos estos valores, se procede a efectuar el cálculo del espacio libre total del que se puede disponer, tal como se muestra en el código. Si el resultado es inferior al espacio mínimo configurado, el valor devuelto por la función será verdadero (hay suficiente espacio para crear un nuevo usuario), y viceversa.

- *Validación del ingreso de un usuario al sistema*

  Esta validación se efectúa cada vez que alguna persona solicita acceso al sistema usuario desde alguna de las computadoras del área de oficinas.

  La función *validaUsuario* se encarga de realizar las verificaciones necesarias, y para esto recibe como parámetros el usuario y la contraseña de la persona que intenta ingresar, y las direcciones MAC e IP de la máquina desde la que se solicita acceso. El valor devuelto por este método es un arreglo de cadenas de

caracteres, con capacidad para dos ítems; el primero tomará únicamente un valor de "0" o "1", que indica el resultado de la validación ("0", negativo; "1", positivo) y de la concesión de acceso; y el segundo contendrá el mensaje de respuesta que ha de presentarse al solicitante.

Como primer paso, se obtienen las IDs tanto del usuario como de la computadora, llamando a las funciones *pideIdUsuarioUser* y *pideIdMaquinaMAC*, respectivamente. Ambas devuelven valores de tipo entero, y cuando el parámetro recibido no coincide con ninguno de los datos registrados, el valor retornado es -1; lo que significa que el acceso debe ser denegado.

Posteriormente, se comprueba que la contraseña ingresada por el usuario sea válida, y que éste se encuentre habilitado (*estaHabilitado* devuelve "verdadero") y desconectado en ese momento (*usuarioEnUso* devuelve "falso").

La siguiente verificación que se lleva a cabo permite determinar si la computadora desde la que el usuario ha solicitado acceso se encuentra asignada a él, y esto se hace llamando a la función *validaMac*, que comprueba la existencia de una relación entre ellos. Adicionalmente, a cada usuario registrado y habilitado se

le asigna un tiempo de acceso, que rige tanto para el ingreso al sistema usuario, como para la solicitud de apertura de la puerta desde el sistema portero; este tiempo se determina, en parte, por medio del campo *siempreActivo* que forma parte de los datos del usuario.

Si el valor de *siempreActivo* es "1", el sistema deberá permitir el ingreso del usuario en todo momento; de lo contrario, su acceso se verá restringido a los días y las horas especificadas en el horario que se le haya asignado.

Si el acceso concedido al usuario es de tipo ilimitado (*siempreActivo= "1"*), ya no es necesario comprobar ningún otro dato, y la función concluye creando un nuevo registro de usuario conectado, al llamar a la función *crearUsuarioConectado*; pero si el valor de *siempreActivo* es "0", todavía debe verificarse que el horario asignado se encuentre habilitado y que la solicitud de acceso del usuario esté dentro de los días y las horas correspondientes; esto último se determina al llamar a la función *estaDentroDeHorario*. Una vez que se han completado estas validaciones, el sistema concede acceso al usuario, y crea el nuevo registro de usuario conectado respectivo.

```
public string[] validaUsuario(string user, string password, string direccMAC, string ip)
{
  string[] Respuesta=new string[2];
  int idMaquina=Maquinas.pideIdMaquinaMAC(direccMAC);
  int idUsuario=Usuarios.pideIdUsuarioUser(user);
  if(idUsuario==-1)
    Respuesta→"0";"El usuario no está registado."
  else
  {
   if(Usuarios.validaPassword(idUsuario,password)==false)
       Respuesta→"0";"La contraseña es incorrecta."
   else
   {
     if(Usuarios.estaHabilitado(idUsuario)==false)
         Respuesta→"0";"El usuario está deshabilitado."
     else
     {
       if(Usuarios.usuarioEnUso(idUsuario)==true)
          Respuesta→"0";"El usuario está conectado."
       else
       {
         if(idMaquina==-1)
             Respuesta→"0";"La máquina no está registrada."
         else
         {
           if(Maquinas.validaMac(idUsuario,direccMAC)==-1)
               Respuesta→"0";"Máquina no está asignada a usuario."
           else
           {
             if(Maquinas.estaHabilitada(idMaquina)==false)
                 Respuesta→"0";"Máquina está deshabilitada."
             else
             {
               if(Usuarios.usuarioSiempreActivo(idUsuario)==false)
               {
                  if(Usuarios.tieneHorarioHabilitado(idUsuario)==false)
                     Respuesta→"0";"Horario de acceso deshabilitado."
                   else
                  {
                       if(Usuarios.estaDentroDeHorario(idUsuario)==false)
                          Respuesta→"0";"Usuario fuera de horario."
                       else
                       {
                           Respuesta→"1"; idUsuario.ToString()+","+idMaquina.ToString()
                           Usuarios.crearUsuarioConectado(idUsuario,mac,1,ip);
                        }
                  }
               }
                else
               {
                    Respuesta→"1"; idUsuario.ToString()+","+idMaquina.ToString()
                    Usuarios.crearUsuarioConectado(idUsuario,mac,1,ip);
               }
    }}}}}}}
  return Respuesta;
}
```

- *Registro de una nueva transacción en la bitácora*

  El procedimiento que se encarga de registrar un nueva transacción en la bitácora, es llamado cada vez que se manda a abrir la puerta; que se establece una comunicación instantánea entre el sistema usuario y el sistema portero; y que un usuario externo deja un mensaje de texto, audio o vídeo a alguno de los usuarios internos, utilizando el sistema portero.

  Los parámetros que recibe este método son los datos que se necesitan para registrar la nueva información en la base de datos (el tipo de transacción, que puede ser "Abrir Puerta", "Comunicación" o "Guardar Mensaje"; la fecha y la hora en que se realizó; el nombre del usuario externo y del usuario interno que intervinieron; el nombre del portero desde el que se efectuó; y la foto capturada del usuario externo, que se envía por red en forma de un arreglo de bytes).

  El código que ejecuta el servidor por medio de este método, se muestra a continuación.

```
public void bool registrarNuevaTransaccion(string transaccion, string fecha, string
hora, string usuarioExterno, string nombreUsuario, string nombrePortero, byte[]
foto)
{
  string sentenciaSQL="INSERT INTO log (transaccion, fecha, hora,
  usuarioexterno, nombreusuario, nombreportero) VALUES
  ('"+transaccion+"', '"+fecha+"','"+hora+"', '"+usuarioExterno+"',
  '"+nombreUsuario+"', '"+nombrePortero+"')";
  OdbcConnection odbcConexion = conectarABase();
  OdbcTransaction trans=
              odbcConexion.BeginTransaction(IsolationLevel.ReadCommitted);
  OdbcCommand myCommand=new
                                  OdbcCommand(sentenciaSQL,odbcConexion);
  trans = odbcConexion.BeginTransaction(IsolationLevel.ReadCommitted);
  myCommand.Transaction = trans;
  myCommand.ExecuteNonQuery();
  trans.Commit();
  int idLog=obtener la ID del registro que se acaba de crear;
  crearArchivoFoto(foto,"log_img",idLog);
  //número máximo de registros
  int numMaxRegistrosLog=obtenerNumMaxRegistrosLog();
  //número actual de registros
  int numRegistrosLog=pideNumRegistrosLog();
  if(numRegistrosLog>numMaxRegistrosLog)
     enviarEmailAdvertencia(numMaxRegistrosLog,numRegistrosLog);
  ChequeaEspacio chqEspacio=new ChequeaEspacio();
  chqEspacio.suficienteEspacioEnDisco();
  cerrarConexionBase(odbcConexion);
```

Una vez que los datos de la nueva transacción han sido guardados, se obtiene la ID que representa a este nuevo registro, *idLog*, y se crea en la computadora servidor, en la ruta *Unidad:\Inetpub\ftproot\log_img*, el archivo correspondiente a la imagen capturada del usuario externo; esto se lleva a cabo por medio de la función *crearArchivoFoto*, que recibe como parámetros el arreglo de bytes que constituye la foto, el nombre de la carpeta en la que ha de guardarse, y el ID de la nueva transacción, que vendrá a ser el nombre con el que se guardará el nuevo archivo JPEG *("idLog".jpg*).

Después de esto se realiza una verificación que permite determinar si el tamaño de la tabla "log" amerita que se saque un respaldo de la información y que se eliminen los registros más antiguos para liberar espacio; este proceso requiere que se obtengan tanto el número actual de registros en la tabla, como el número máximo de registros que se haya configurado. Si el primer valor sobrepasa al segundo, la tabla habrá crecido más de lo establecido, y se enviará por correo electrónico un mensaje de advertencia al administrador del sistema, indicando lo sucedido.

Por último, se realiza un chequeo que permite determinar si hay suficiente espacio en el disco, como para que el sistema SCCIO funcione correctamente, y los usuarios puedan disponer de la totalidad del espacio asignado para sus carpetas de mensajes recibidos; para esto se crea una instancia de la clase *ChequeaEspacio* que se encarga de ejecutar métodos muy similares a los que se analizaron para la función *hayEspacioParaNuevoUsuario*, llamando al método *suficienteEspacioEnDisco*. Este método lleva a cabo la comprobación de espacio, y si determina que es insuficiente, también envía un email de advertencia al administrador, para

ponerlo al tanto del problema e indicarle que debe liberar espacio en la unidad respectiva.

- *Solicitud de envío de un nuevo mensaje para un usuario interno*

  Cuando un usuario interno no se encuentra disponible, los usuarios externos tienen la opción de dejarle un mensaje de texto, audio o vídeo. Para esto, el servidor debe primero verificar si el usuario dispone de suficiente espacio libre en su carpeta de mensajes, para guardar uno nuevo, y posteriormente, enviar un correo electrónico que contenga una notificación, indicando que el usuario ha recibido un nuevo mensaje (en caso de que éste sea de audio o vídeo), o el mensaje en sí, cuando es de texto. A continuación se presentan los fragmentos de código que llevan a cabo estas dos tareas.

```
//verifica si hay espacio en el directorio del usuario para un nuevo mensaje
public string[] hayEspacioEnDirParaNuevoMsj(int idUsuario)
{
  string[] Respuesta=new string[2];
  long
     espacioLibreEnDir=Usuarios.obtenerEspacioLibreEnDirectorio(idUsuario);
  if(espacioLibreEnDir<tamañoMsjMax)
     Respuesta→"0";"El directorio del usuario no tiene suficiente espacio
                    disponible."
  else
     Respuesta→"1";""
  return Respuesta;
}
```

```
//envía una notificación de nuevo mensaje recibido o el mensaje de texto en sí
public void enviarEmail(int idUsuario,int idPortero,string fecha, string hora, string
subjectMsj,string usuarioExterno,byte[] foto,string mensaje, string tipoMensaje,
string nombreMensaje)
{
  string nombrePortero=Porteros.pideNombre(idPortero);
  string nombreUsuario=Usuarios.pideNombre(idUsuario);
  string emailUsuario=Usuarios.pideEmail(idUsuario);
  string ipServidorCorreo= obtener la IP del servidor de correo electrónico
  string emailServidor=SCCIO@sccio.com, encabezadoMsj="";
  string user=Usuarios.pideUser(idUsuario);
  string rutaImg= Application.StartupPath+@"\imgTemp.jpg";
  MemoryStream MS = new MemoryStream(foto, false);
  Image imagen = Image.FromStream(MS);
  imagen.Save(rutaImg,ImageFormat.Jpeg);
  switch (tipoMensaje)
  {
    case "texto":
      encabezadoMsj ="Este mensaje fue enviado desde el portero'"+nombrePortero
     +"', por '"+usuarioExterno+"', a las '"+hora+"' del '"+fecha+"'.";
     break;
    case "audio":
      encabezadoMsj ="Ha recibido un nuevo mensaje de audio, enviado desde el
     portero'"+nombrePortero+"', por '"+usuarioExterno+"', a las '"+hora+"' del
     '"+fecha+"'.";
     break;
    case "video":
     encabezadoMsj ="Ha recibido un nuevo mensaje de vídeo, enviado desde el
     portero'"+nombrePortero+"', por '"+usuarioExterno+"', a las '"+hora+"' del
     '"+fecha+"'.";
     break;
    }
    string msjAEnviar= encabezadoMsj+mensaje;
    MailMessage msjEmail = new MailMessage();
    msjEmail.From = emailServidor;
    msjEmail.To = emailUsuario;
    msjEmail.Subject = subjectMsj;
    msjEmail.Body = msjAEnviar;
    bool fotoEnCorreo=Usuarios.fotoEnCorreo(idUsuario);
    if (foto!=null && fotoEnCorreo==true)
     msjEmail.Attachments.Add(new MailAttachment(rutaImg, MailEncoding.Base64));
    SmtpMail.SmtpServer=ipServidorCorreo;
    SmtpMail.Send(msjEmail);
    if(tipoMensaje!="texto")
    {
          Mensajes.crear(idUsuario, nombrePortero, formatoMensaje, usuarioExterno,
          fecha,hora, nombreMensaje, foto);
          long tamañoMsj=obtenerTamañoMensaje(user,nombreMensaje);
          if(tamañoMsj>tamañoMsjMax)
          {
             tamañoMsjMax=tamañoMsj;
            modificarArchivoTamañoMsjMax(tamañoMsjMax);
          }
     }
    Logs.crearLog("Guardar Mensaje", fecha, hora, usuarioExterno, nombreUsuario,
    nombrePortero, foto);
}
```

La primera función, *hayEspacioEnDirParaNuevoMsj*, recibe como parámetro el ID del usuario al que estaría dirigido el mensaje; por medio de este dato, se obtiene la cantidad de espacio libre que hay actualmente en el directorio del usuario, y se lo almacena en la variable *espacioLibreEnDir*. Este dato se compara luego con el valor contenido en la variable *tamañoMsjMax*, que guarda el tamaño del mensaje más grande que se haya enviado hasta ese momento. Si el espacio libre en el directorio resulta superior al tamaño de mensaje máximo, significa que el usuario todavía tiene espacio disponible en su carpeta para albergar un nuevo archivo de mensaje.

La función *enviarEmail*, por otro lado, recibe como parámetros ciertos datos que se requieren para guardar un nuevo registro en la tabla "mensajes", y otros que permiten obtener la información necesaria. Para poder efectuar el envío de un correo electrónico se requieren básicamente datos tales como el email del destinatario (*emailUsuario*), el email del remitente (*emailServidor*), y la dirección IP del servidor de correo (*ipServidorCorreo*); esta función se encarga de recopilar todos esos datos, a la vez que construye el cuerpo de mensaje que ha de enviarse (*msjAEnviar*). Para hacer esto último, se toma en cuenta el tipo de mensaje que se requiere enviar (*tipoMensaje*),

es decir, si se trata de una notificación de que al usuario le han enviado un nuevo mensaje (*tipoMensaje="audio"* o *tipoMensaje="video"*) o si, por el contrario, se trata del mensaje en sí (*tipoMensaje="texto"*), tal como puede observarse en el fragmento de código anterior. En ambos casos, se incluirá una foto del usuario externo que envía el mensaje, si es que así lo ha especificado el usuario interno en su configuración (*fotoEnCorreo= "1"*). La foto que llega como un arreglo de bytes es convertida en un archivo, que se guarda temporalmente en la ubicación especificada por *rutaImg*; esto permite que pueda ser luego adjuntada al mensaje de correo electrónico que se envía al usuario interno.

Si el tipo de mensaje que se envía es de "vídeo" o "audio" (*tipoMensaje!="texto"*), se creará un nuevo registro asociado al usuario interno en la tabla "mensaje". También se llevará a cabo una verificación para determinar si el archivo denominado de acuerdo al valor de la variable *nombreMensaje*, que se ha guardado en el directorio dado por el dato *user*, es de mayor tamaño que *tamañoMsjMax*. De ser así, el valor de esta variable se reemplaza por el tamaño en bytes del nuevo archivo de mensaje, y debido a que este valor se mantiene adicionalmente de forma permanente en un archivo, el

contenido de éste también es modificado por medio del procedimiento *modificarArchivoTamañoMsjMax*. Finalmente, la transacción *Guardar Mensaje*, que se acaba de llevar a cabo es registrada en la tabla "log" que representa a la bitácora, llamando al método *crearLog*.

- *Reporte de conexión*

  Con el objetivo de proveer consistencia en cuanto al estado de conexión tanto de usuarios como de porteros, se han incluido ciertas funciones en el código del servidor, que llevan a cabo un monitoreo de verificación, y el código de algunos de estos procedimientos se describe a continuación.

```
public void reportarConexionUsuario(int idUsuario, int idMaquina, string
ip)
{
  for(int i=0;i<usuariosReportados.Count;i++)
  {
    if(usuariosReportados[i].id==idUsuario)
          usuariosReportados.RemoveAt(i);
  }
  registrarNuevoReporte(idUsuario,"u",idMaquina,ip);
}
```

  El procedimiento *reportarConexionUsuario* es llamado cada minuto desde los sistemas usuario y portero que se encuentren corriendo, y lo que hace, tal como su nombre lo indica, es reportar al servidor la conexión para que éste pueda mantener información actualizada con respecto a los usuarios y porteros

conectados. Cada vez que el servidor recibe un nuevo reporte, busca en el arreglo correspondiente (*usuariosReportados* o *porterosReportados*, de acuerdo al caso) el elemento cuya ID coincida con la del usuario o portero que se está reportando; si encuentra una coincidencia, el elemento es eliminado del arreglo, tal como se muestra en el fragmento de código anterior.

```
public void registrarNuevoReporte(int id, string tipo, int idMaquina,
string ip)
{
  DateTime horaRecepcion=DateTime.Now;
  DateTime horaDesconexion=horaRecepcion.AddMinutes(1.25);
  ReportarConectado nuevoReporte=new
          ReportarConectado(horaRecepcion,horaDesconexion,id);
  switch(tipo)
  {
    case "u":
    {
      usuariosReportados.Add(nuevoReporte);
      if(Usuarios.verificarUsuarioEnUso(id)==false)
           validarReporteConexionUsuario(id,idMaquina,ip);
     }
     break;
     case"p":
     {
          porterosReportados.Add(nuevoReporte);
          if(Porteros.verificarPorteroEnUso(id)==false)
            Porteros.crearPorteroConectado(id,ip);
      }
      break;
    }
}
```

El procedimiento *registrarNuevoReporte* agrega un nuevo elemento al arreglo de *usuariosReportados* o *porterosReportados*, según sea el caso, asignándoles una hora de desconexión que sirve como referencia para poder determinar cuándo un elemento debe ser desconectado.

Adicionalmente, si por ejemplo, algún usuario que ha enviado su reporte no se encuentra registrado en el arreglo de usuarios conectados, este procedimiento se encarga de realizar el registro correspondiente, valiéndose de los datos que recibe como parámetros; esto vale también para el caso de los porteros.

```
public void desconectar()
{
  DateTime horaActual=DateTime.Now;
  for(int i=0;i<usuariosReportados.Count;i++)
  {
    if(usuariosReportados[i].debeSerDesconectado(horaActual))
    {
          eliminarUsuarioConectado(usuariosReportados[i].id);
          usuariosReportados.RemoveAt(i);
    }
  }
  for(int i=0;i<porterosReportados.Count;i++)
  {
    if(porterosReportados[i].debeSerDesconectado(horaActual))
    {
          eliminarPorteroConectado(porterosReportados[i].id);
          porterosReportados.RemoveAt(i);
    }
  }
}
```

El procedimiento desconectar se ejecutada cada minuto con diez segundos, y lo que hace es remover del arreglo de usuarios conectados o porteros conectados, aquellos elementos cuya hora de desconexión ha sido superada y no han enviado reportes de conexión hasta ese momento.

Módulo Usuario

- *Visualizar un portero*

  El procedimiento *VisualizarPortero,* permite visualizar el vídeo capturado por un portero específico, identificado por su nombre, que es recibido como parámetro del método.

  Este procedimiento es llamado después de haber enviado previamente una solicitud de visualización al respectivo portero y de haber recibido su confirmación; esto se debe a que el portero no puede realizar la transmisión y captura del vídeo si la cámara, en ese instante, está siendo utilizada para la captura de imágenes instantáneas, o para el establecimiento de comunicaciones en tiempo real.

  A continuación se muestra el código del procedimiento.

```
private void VisualizarPortero(string nombrePortero)
{
   if(CanalComunicOcupado)
         mensajeError("No puede visualizar el portero, mientras usted se
        encuentre en una comunicación.");
   else
   {
       string[] datosCom=dameDatosComPortero(nombrePortero);
       CanalComunicOcupado=true;
       nombreCanalActivo=nombrePortero;
       string direccionIP=datosCom[0];
       int puerto=Convert.ToInt32(datosCom[1]);
       System.Net.IPEndPoint ipEndPoint=new System.Net.IPEndPoint
                          (System.Net.IPAddress.Parse(direccionIP),puerto);
```

```
        //creación del canal
        MSR.LST.ConferenceXP.VenueData vData=new
        MSR.LST.ConferenceXP.VenueData(nombreCanalActivo,ipEndPoint,
       128,MSR.LST.ConferenceXP.VenueType.PublicVenue,null);
          //configuración de la vídeo-conferencia
          MSR.LST.ConferenceXP.Conference.AutoSendVideo = false;
          MSR.LST.ConferenceXP.Conference.AutoSendAudio = false;
          MSR.LST.ConferenceXP.Conference.AutoPlayLocal = false;
          MSR.LST.ConferenceXP.Conference.AutoPlayRemote = true;
         MSR.LST.ConferenceXP.Conference.AutoPosition =
                MSR.LST.ConferenceXP.Conference.AutoPositionMode.FourWay;
          MSR.LST.ConferenceXP.Conference.CallingForm=this;
          //unirse al canal de vídeo-conferencia
          MSR.LST.ConferenceXP.Conference.JoinCustomVenue(vData);
    }
}
```

El procedimiento, para poder realizar la visualización, deberá primero crear un canal de comunicación; para esto necesita de una dirección IP, un número de puerto, y el nombre del portero; éste último dato será utilizado para especificar el nombre del canal por el que viajará el flujo de vídeo. Una vez que se haya creado el canal, deberán configurarse ciertos atributos de la conferencia; estos son, los parámetros *AutoSendAudio* y *AutoSendVideo*, que indican si se debe enviar audio y vídeo respectivamente, por medio del canal. En este caso, estos atributos serán colocados en "falso", porque el extremo del portero no debe visualizar al extremo del usuario interno. Por esta misma razón el atributo *AutoPlayLocal* estará en "falso", y *AutoPlayRemote* en "verdadero". Para que la visualización se lleve a cabo, en el extremo del portero deberá estar definida

una función muy similar, pero con una configuración de la conferencia diferente; por ejemplo, *AutoSendAudio*, y *AutoSendVideo* tendrán el valor de "verdadero", porque lo que se capture deberá ser enviado a través del canal de comunicación; y los parámetros *AutoPlayLocal*, y *AutoPlayRemote* estarán en "falso", porque el portero no deberá mostrar el vídeo capturado.

Por otro lado, para establecer comunicaciones tanto de audio como de vídeo, se utilizarán funciones similares. En el sistema usuario, y en el sistema portero se deberá crear un canal de comunicación con los mismos datos requeridos en el procedimiento de *VisualizaciónPortero*; los atributos que cambiarían serán los de la configuración de la conferencia, anteriormente mencionados, que tendrán los valores de "verdadero" o "falso", de acuerdo a lo que se requiera.

- *Mensajes enviados por el sistema portero que se reciben por el socket cliente.*

  El procedimiento que identifica los diferentes mensajes recibidos por el socket, se llama *nuevoMensaje.* Este procedimiento, recibe como parámetro el socket cliente, por medio del cual se obtendrán los mensajes recibidos. Luego de obtener el

mensaje, se identifican sus partes para determinar de qué tipo es y la acción que deberá realizarse, tal como se muestra en el código a continuación.

```
public void nuevoMensaje(Socket client)
{
 string datosRecibidos=mensajeSocket, msjRecibido="";
 string[] datosInicioSesion=new string[3];
 if(datosRecibidos.IndexOf("|EOM")!=-1)
 {
   datosRecibidos=
   datosRecibidos.Remove(datosRecibidos.IndexOf("|EOM"),4);
   string [] split = datosRecibidos.Split('|');
   string msjAEnviar="";
   msjRecibido=split[1];
   mensajeSocket=split[1];
   if(datosRecibidos.StartsWith("S|"))
   {
        //Presenta PopUp preguntando si desea aceptar la comunicación
        if (PopUpSCCIO.result==DialogResult.OK)
            //Establece la comunicación
        else if (PopUpSCCIO.result==DialogResult.Cancel)
            //Envia mensaje al portero de solicitud denegada
    }else if(datosRecibidos.StartsWith("C|"))
        // Cierra la comunicación
   else if(datosRecibidos.StartsWith("E|"))
       comunicacion.usuarioRemotoEstaEscribiendoMsj(msjRecibido);
   else if(datosRecibidos.StartsWith("M|"))
       comunicacion.presentarMsjEnCartelera
       (msjRecibido,usuarioExterno,"azul");
   else if (datosRecibidos.StartsWith("VA|"))
        VisualizarPortero(nombrePortero);
   else if (datosRecibidos.StartsWith("VD|"))
        //Cierra canal de visualización
   else if (datosRecibidos.StartsWith("CV|"))
      //Cierra canal de visualización
}
```

Módulo Portero

- *Mostrar vídeo*

  La función *StartupVideo*, nos permite mostrar el vídeo que está siendo capturado por una cámara. Recibe como parámetro una interfaz en la que se especifica la cámara que se va a utilizar

para obtener el vídeo digital. Esta función crea un dispositivo de captura basado en lo que indica su parámetro de entrada, luego prepara las interfaces que permiten controlar el audio y el vídeo; después se establecen los parámetros para obtener y mostrar el vídeo, y si todo se inicializa correctamente, la función retornará "verdadero", y se podrá visualizar el vídeo; en caso contrario, si falla en alguno de los pasos previos, la función retornará el valor de "falso". A continuación se muestra parte del código que permite que el vídeo capturado por la cámara configurada sea mostrado al usuario.

```
bool StartupVideo( UCOMIMoniker mon )
{
        int hr;
if( ! CreateCaptureDevice( mon ) )
                return false;
        if( ! GetInterfaces() )
                return false;
        if( ! SetupGraph() )
                return false;
        if( ! SetupVideoWindow() )
                return false;
        hr = mediaCtrl.Run();
        if( hr < 0 )
                Marshal.ThrowExceptionForHR( hr );
        return true;
}
```

- *Enviar un mensaje de texto, vídeo o audio a un usuario interno*

  El procedimiento que sirve para enviar al usuario un mensaje, que anteriormente fue grabado en el sistema portero, se denomina *EnviarMensaje*.

Este procedimiento primeramente obtendrá los bytes de la imagen instantánea capturada por la cámara, validará que todos los datos requeridos sean válidos (nombre de quien envía el mensaje; el asunto del mensaje; el cuerpo del mensaje, en caso de que éste sea de texto; o la existencia del archivo, en caso de que sea de audio o vídeo), y finalmente se envían todos los datos al servidor.

Si el mensaje es de audio o de vídeo, antes de enviar los datos, se transferirá al servidor vía FTP el archivo que contiene el mensaje grabado, para que sea guardado en la respectiva carpeta de usuario. Este archivo contendrá un nombre único compuesto por el asunto, el identificador del portero desde el que se grabó, la fecha, y la hora; este nombre se construirá por medio de la función *crearNombreMensaje*.

A continuación, se muestra el código del procedimiento *EnviarMensaje.*

```
private void EnviarMensaje()
{
   byteFoto= new byte[0];
   if (pbFoto.Image!=null)
          byteFoto=ObtenerBytesImagen("usuarioExt.jpg");
   if (tipoMsj=="texto" && datosTextoValidos())
  {
     Respuesta=
         servidor.enviarEmail(frmInicio.idUsuario,frmInicio.idPortero,
         fecha(),hora(),asunto,nombre,byteFoto, txtMensaje.Text,tipoMsj,"");
          mensajeError(Respuesta[1],"");
  }
  else if ((tipoMsj=="video" || tipoMsj=="audio") &&
          datosValidos())
 {
         nombreMensaje=
                  crearNombreMensaje(frmInicio.idPortero,asunto);
         File.Move(MsjTemporal,nombreMensaje);
         SubirMensaje();
        Respuesta=
        servidor.enviarEmail(idUsuario,idPortero,fecha(),hora(),
        asunto,nombre,byteFoto,"",tipoMsj,nombreMensaje)
  }
}
```

- *Salir del Canal de Comunicación*

  El procedimiento *CerrarCanal*, cierra el canal de comunicación que se creó al momento de establecer una comunicación o de visualizar el vídeo capturado por el portero. Este procedimiento es llamado cuando se cierra una comunicación, o cuando se va a efectuar alguna operación con las cámaras, como la captura de imágenes instantáneas al grabar mensajes o al ingresar por la puerta; esta función también es llamada cuando al estar en una visualización, el sistema cliente que la ha solicitado cierra la ventana respectiva, indicando que ya no desea visualizar el vídeo capturado por el portero. La acción anterior sólo llamará a este procedimiento si se trata del único usuario que se

encuentra en la visualización; ya que si son varios, el canal no se cerrará sino hasta que no haya usuarios conectados a él. A continuación se muestra el código del procedimiento.

```
public void CerrarCanal()
{
   if(nombreCanalActivo.Length>0)
   {
         string canal=nombreCanalActivo;
         nombreCanalActivo="";
         MSR.LST.ConferenceXP.Conference.Venues[canal].Leave();
         MSR.LST.ConferenceXP.Conference.Venues.Remove(canal
         CamaraPuertaOcupada=false;
    }
    if (visualizando)
    {
        visualizando=false;
         while(usuariosVisualizando.Count>0)
         {
              enviarMensaje("CV||EOM",usuariosVisualizando[0]);
              usuariosVisualizando.RemoveAt(0);
         }
     }
}
```

La variable *nombreCanalActivo*, contiene el nombre del portero que se utiliza para identificar el canal de comunicación; la variable *CamaraPuertaOcupada* indica que la cámara que se usa para la visualización está siendo utilizada cuando su valor es "verdadero"; la variable *visualizando* indica si el canal que el portero tiene abierto está siendo utilizado para permitir la visualización de una de sus cámaras; por último, el arreglo *usuariosVisualizando*, contiene las direcciones IP de los computadores de los usuarios internos que están visualizando en ese momento el vídeo capturado por el portero. Cada vez

que un usuario solicita esta visualización, su dirección IP será agregada a este arreglo, y cuando el usuario cierre la ventana de vídeo, su dirección IP será removida de él. Esta variable es útil cuando se solicita cerrar el canal para establecer una comunicación o utilizar la cámara para capturar fotos instantáneas, ya que el portero tiene que enviar un mensaje a cada usuario que estaba visualizando el vídeo capturado por el portero, indicando que ha sido necesario cerrar el canal, y que intente realizar nuevamente la visualización más tarde.

5.3. Plan de pruebas

A continuación se presenta el resumen de los resultados obtenidos en las pruebas que se realizaron para seleccionar los compresores de audio y de vídeo que se utilizan en el sistema.

Compresor de video	Compresor de audio	Espacio utilizado (MB)	Calidad
Sin compresor	Sin compresor	350	Vídeo Original
Cinepak Codec by Radius	MPEG Layer-3	14.78	El vídeo resultante es de buena calidad, pero el sonido se escucha entrecortado.
Cinepak Codec by Radius	Windows Media Audio V2	16.15	Hay mucho desfase entre el audio y vídeo; y el sonido se escucha entrecortado.
Indeo® video 5.04 Compression Filter	MPEG Layer-3	18.31	La calidad del resultado es aceptable. El audio y el vídeo no se encuentran sincronizados.
Indeo® video 5.04 Compression Filter	Windows Media Audio V2	16.82	La calidad del resultado es aceptable. El audio y el vídeo no se encuentran sincronizados.
MJPEG Compressor	MPEG Layer-3	332.29	La calidad del resultado es aceptable. El audio y el vídeo no se encuentran sincronizados.
MJPEG Compressor	Windows Media Audio V2	333.65	La calidad del resultado es aceptable. El audio y el vídeo no se encuentran sincronizados.
Microsoft MPEG-4 VKI Codec V3	MPEG Layer-3	3.26	La calidad del resultado es aceptable. El audio y el vídeo no se encuentran sincronizados.
Microsoft MPEG-4 VKI Codec V3	Windows Media Audio V2	4.64	La calidad del resultado es aceptable. El audio y el vídeo no se encuentran sincronizados.
Microsoft MPEG-4 VKI Codec V2	MPEG Layer-3	4.61	La calidad del vídeo no es muy buena. El audio y el vídeo no se encuentran sincronizados.
Microsoft MPEG-4 VKI Codec V2	Windows Media Audio V2	3.28	La calidad del vídeo no es muy buena. El audio y el vídeo no se encuentran sincronizados.
Microsoft MPEG-4 VKI Codec V1	MPEG Layer-3	3.24	La calidad del resultado es aceptable. El audio y el vídeo no se encuentran sincronizados.
Microsoft MPEG-4 VKI Codec V1	Windows Media Audio V2	4.62	La calidad del resultado es aceptable. El audio y el vídeo no se encuentran sincronizados.

Tabla 5.2.- Resultados obtenidos en las pruebas de selección de compresores

Con el fin de asegurar que el sistema funcione de acuerdo a lo esperado se ha preparado el siguiente plan de pruebas, en base a las operaciones más relevantes, que se realizan con mayor frecuencia.

Número	1
Nombre de la prueba	Solicitud de ingreso al área de oficinas
Instrucciones para la prueba	1.- Un usuario registrado solicita que se le conceda acceso al área de oficinas, por medio de la opción "Abrir Puerta" del sistema portero. 2.- El usuario ingresa su clave de acceso, y presiona el botón "Aceptar". 3.- El sistema portero envía el dato al servidor y éste verifica que la clave exista. 4.- El usuario espera por una respuesta.
Comportamiento aceptable	Si la clave ingresada existe, el sistema portero mandará a abrir la puerta, y tomará la foto instantánea del usuario que ingresa, para registrar la transacción en la bitácora. La transacción deberá constar en la bitácora. De lo contrario, se presentará un mensaje indicando que la clave ingresada es incorrecta.

Número	2
Nombre de la prueba	Establecimiento de Comunicación de vídeo
Instrucciones para la prueba	1.- El usuario externo selecciona al usuario interno con el que se desea comunicar, y presiona el botón "Comunicar" (el usuario seleccionado se encuentra disponible). 2.- El usuario externo ingresa su nombre y presiona el botón "Aceptar". Espera la respuesta a su solicitud. 3.- El usuario interno acepta la solicitud de comunicación.
Comportamiento aceptable	Tanto en el sistema usuario como en el sistema portero deberán aparecer dos ventanas mostrando el vídeo que se está capturando en ambos extremos, y los dos usuarios podrán escucharse entre sí. La transacción deberá constar en la bitácora.

Número	3
Nombre de la prueba	Establecimiento de Comunicación de texto
Instrucciones para la prueba	1.- El usuario externo selecciona al usuario interno con el que se desea comunicar, y presiona el botón "Comunicar" (el usuario seleccionado se encuentra disponible). 2.- El usuario externo ingresa su nombre y presiona el botón "Aceptar". Espera la respuesta a su solicitud. 3.- El usuario interno acepta la solicitud de comunicación.
Comportamiento aceptable	Tanto en el sistema usuario como en el sistema portero deberá aparecer una ventana mostrando el vídeo que se está capturando en el sistema portero, y el usuario interno podrá escuchar al usuario externo. También se presentará en ambos extremos la interfaz correspondiente a la comunicación de texto. La transacción deberá constar en la bitácora.

Número	4
Nombre de la prueba	El usuario externo envía un mensaje de texto al usuario interno
Instrucciones para la prueba	1.- El usuario externo selecciona al usuario interno al que está solicitando. El usuario seleccionado no se encuentra disponible. 2.- Presiona el botón "Enviar Mensaje". 3.- Ingresa su nombre, y el asunto y cuerpo del mensaje. 4.- Se toma una foto instantánea, presionando el botón "Tomar Foto". 5.- Presiona el botón "Enviar".
Comportamiento aceptable	El sistema manda los datos al servidor, y éste envía el mensaje por correo electrónico al usuario interno. El usuario interno deberá recibir por email el mensaje enviado por el usuario externo desde el sistema portero. La transacción deberá constar en la bitácora.

Número	5
Nombre de la prueba	El usuario externo envía un mensaje de vídeo al usuario interno
Instrucciones para la prueba	1.- El usuario externo selecciona al usuario interno al que está solicitando. El usuario seleccionado no se encuentra disponible. 2.- Presiona el botón "Enviar Mensaje". 3.- Ingresa su nombre, y el asunto del mensaje. 4.- Graba el mensaje de vídeo. 5.- Se toma una foto instantánea, presionando el botón "Tomar Foto". 6.- Presiona el botón "Enviar".
Comportamiento aceptable	El archivo correspondiente al mensaje de vídeo debe encontrarse en la carpeta de mensajes recibidos del usuario interno. El sistema manda los datos al servidor, y éste envía una notificación por correo electrónico al usuario interno, indicándole que ha recibido un nuevo mensaje. La transacción deberá constar en la bitácora.

Número	6
Nombre de la prueba	Apertura manual de la puerta
Instrucciones para la prueba	1.- Una persona presiona el botón de apertura manual de la puerta.
Comportamiento aceptable	La puerta se deberá abrir, y el dispositivo electrónico que controla la puerta enviará una señal al sistema portero indicando que la puerta ha sido abierta manualmente. El sistema portero deberá mandar a tomar la foto para el registro de la transacción en la bitácora. La transacción deberá constar en la bitácora.

5.4. Resultados de las pruebas

En esta sección se presentarán los resultados que se obtuvieron al realizar las pruebas anteriormente mencionadas, en el sistema SCCIO. En cada caso, el grado de éxito se ha medido en comparación al comportamiento aceptable descrito para la prueba.

Número	1
Nombre de la prueba	Solicitud de ingreso al área de oficinas
Resultados	En el primer intento, la clave ingresada no coincidió con ninguna de las claves registradas, y el sistema presentó un mensaje que decía: *"Acceso denegado. La clave ingresada es incorrecta."* En el segundo intento, se ingresó una clave de acceso registrada, y la puerta del área de oficinas se abrió. Se comprobó que constaran en la bitácora la información y la foto correspondientes a la transacción.

Número	2
Nombre de la prueba	Establecimiento de Comunicación de vídeo
Resultados	Las dos ventanas de vídeo aparecieron tanto del lado del usuario interno como del usuario externo, y se pudieron ver y escuchar satisfactoriamente. Se verificó que constaran en la bitácora la información y la foto correspondientes a la transacción.

En la figura 5.3 que se presenta a continuación se puede observar el resultado del establecimiento de la comunicación de vídeo.

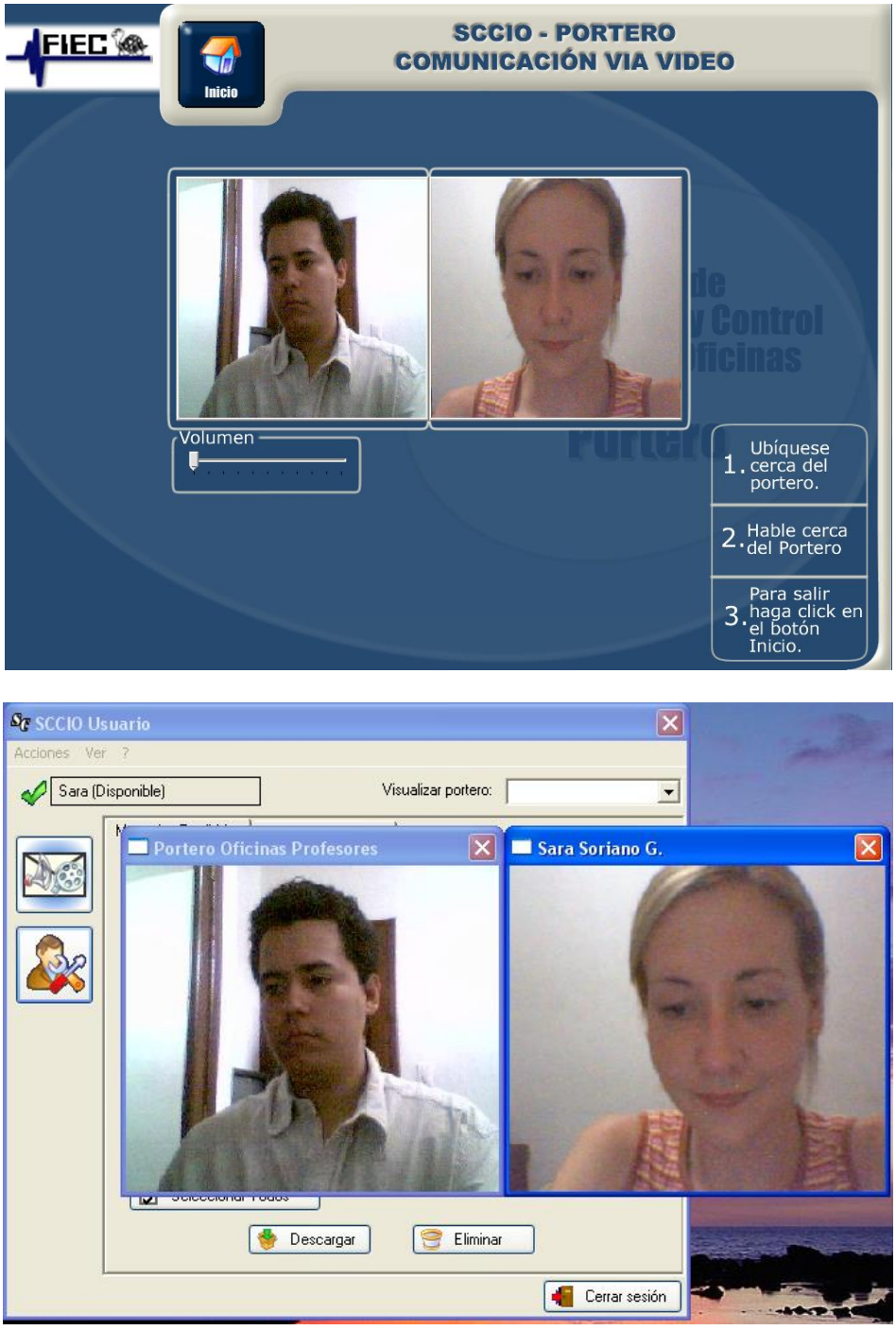


Figura 5.3.- Ventanas de la comunicación de vídeo

Número	3
Nombre de la prueba	Establecimiento de Comunicación de texto
Resultados	El vídeo capturado por el sistema portero se mostró tanto del lado del usuario interno como del usuario externo. En ambos extremos se presentó la interfaz correspondiente a la comunicación de texto, y se pudieron comunicar satisfactoriamente. Se comprobó que constaran en la bitácora la información y la foto correspondientes a la transacción.

Número	4
Nombre de la prueba	El usuario externo envía un mensaje de texto al usuario interno
Resultados	El usuario interno recibió el mensaje de texto escrito por el usuario externo, vía correo electrónico. El usuario interno había especificado en su configuración que deseaba que se adjuntara la foto del remitente a los mensajes, de modo que también recibió la foto capturada del usuario externo. Se verificó que constaran en la bitácora la información y la foto correspondientes a la transacción.

Número	5
Nombre de la prueba	El usuario externo envía un mensaje de vídeo al usuario interno
Resultados	El usuario interno visualizó en la lista de mensajes recibidos, que presenta el sistema usuario, el mensaje de vídeo enviado por el usuario externo. También recibió la notificación vía correo electrónico de que había recibido un nuevo mensaje de vídeo. Se comprobó que constaran en la bitácora la información y la foto correspondientes a la transacción.

Número	6
Nombre de la prueba	Apertura manual de la puerta
Resultados	Se abrió la puerta y se verificó que constaran en la bitácora la información y la foto correspondientes a la transacción.

En la figura 5.4, se puede observar el resultado de la prueba de apertura manual de la puerta tal como fue registrado en la bitácora, junto con la foto que se capturó del usuario que realizó la operación.

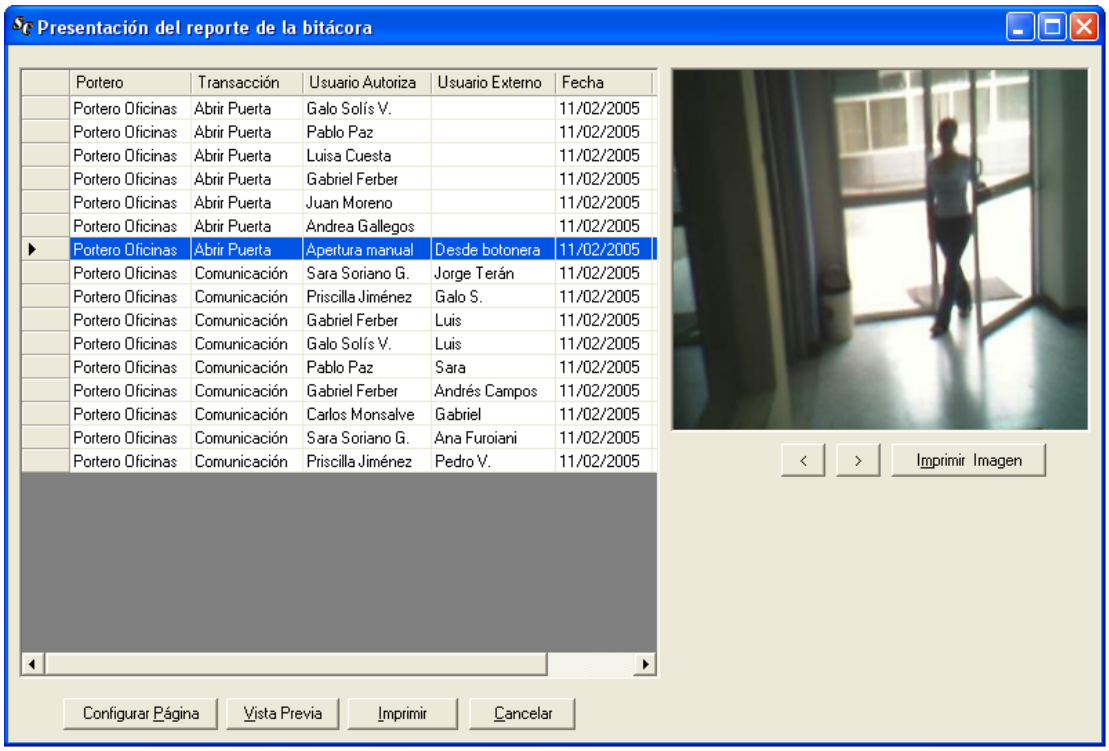


Figura 5.4.- Presentación de la transacción de apertura manual de la puerta en la bitácora

CAPÍTULO 6

6. CONCLUSIONES Y RECOMENDACIONES

Como resultado del desarrollo de esta tesis, tenemos un sistema que se fundamenta en el uso de multimedios, para llevar a cabo sus tareas principales; estos hacen más natural para el usuario la interacción con el sistema y por tanto, agilitan y hacen más eficiente el proceso de comunicación. Adicionalmente, constituyen el mecanismo más eficaz en lo que respecta a brindar seguridad, ya que permiten identificar de manera inequívoca a las personas que acceden al área de oficinas.

Por medio de este sistema, los usuarios de la organización que no dispongan de una oficina o que no pasen la mayor parte del tiempo en ella, pueden mantenerse informados con respecto a las personas que los han estado solicitando, mediante los mensajes que estos les dejan, y que pueden ser revisados en cualquier momento desde el sistema usuario Web. Adicionalmente, tanto la seguridad, como el control de acceso, se

ven mejorados, principalmente mediante el registro de todas las transacciones en la bitácora, y la captura de fotos instantáneas de las personas que ingresan o salen del área de oficinas.

Este sistema puede implementarse en entornos organizacionales de cualquier tamaño, y soportar futuros crecimientos, lo que lo hace escalable. Su adaptabilidad permite que pueda funcionar con diversas tecnologías existentes y soportar cambios a tecnologías posteriores. Por ejemplo, el sistema funciona con varios dispositivos de captura de vídeo, base de datos, y compresores de audio y vídeo.

La herramienta de desarrollo seleccionada, Microsoft Visual C#, proporciona facilidades en la manipulación de contenido multimedia, y en la comunicación entre sistemas dentro de un entorno distribuido. Dado que estas dos características eran fundamentales en el desarrollo del proyecto y que el lenguaje de programación se encuentra ligado al entorno Windows, la plataforma seleccionada tanto para el servidor como para las aplicaciones clientes fue ésta. .NET Framework suministra dos tipos de canales: HTTP y TCP; y nuestro proyecto presenta la opción de trabajar con cualquiera de los dos. Cuando el canal es de tipo HTTP, los mensajes que se envían entre aplicaciones están en XML, un estándar abierto, que cualquier aplicación puede procesar, independientemente de

la plataforma. Esto representa una facilidad para futuras implementaciones de las aplicaciones clientes, en otras plataformas.

El espacio de disco duro asignado a los usuarios se utiliza eficientemente, mediante el uso de compresores tanto para el audio como para el vídeo, que ayudan a reducir considerablemente el tamaño de los mensajes de este tipo que se envían a los usuarios internos. Adicionalmente, las comunicaciones instantáneas de vídeo o de audio que se llevan a cabo entre el sistema usuario y el sistema portero, se realizan eficientemente con baja latencia, y sin congestionar la red.

Entre las mejoras que podrían realizarse posteriormente, estarían la introducción de un sistema biométrico o de lectora de tarjetas de acceso para identificar al usuario interno de manera más rápida, sin necesidad de que éste tenga que ingresar manualmente su clave de acceso; y la utilización de una pantalla sensible al tacto para facilitar la interacción del usuario con el sistema portero.

APÉNDICE A

CASOS DE USO Y ESCENARIOS

Caso de Uso 1: Abrir la puerta usando el portero

Actores: Persona que ingresa su clave al solicitar acceso desde el sistema portero (usuario interno).

Escenario 1.1: Un usuario interno solicita acceso, utilizando el sistema portero y se le concede.

Supuestos:

- La clave de acceso ingresada existe.
- El usuario está habilitado e intenta ingresar dentro del horario permitido.

Salidas:

- Se muestra un mensaje de éxito.

- Se abre la puerta para permitir el acceso.
- Se almacena un nuevo registro en la bitácora.

Caso de Uso 2: Comunicación usuario externo - usuario interno por medio del sistema portero.

Actores: Persona que solicita establecer comunicación desde el portero (usuario externo); persona a la que va dirigida la solicitud de comunicación (usuario interno).

Escenario 2.1: Solicitud exitosa de comunicación con el usuario interno.

Supuestos:

- El usuario interno solicitado se encuentra disponible.
- Se notificó al usuario que alguien lo solicita, utilizando un mensaje de tipo pop-up.
- El usuario acepta la solicitud de comunicación.
- Se identifica el tipo de comunicación que el usuario tiene configurado (texto, audio, o vídeo).

Salidas:

- Se establece la comunicación entre el usuario externo y el usuario interno.
- Se añade el registro correspondiente a la bitácora.

Escenario 2.2: Un usuario externo solicita comunicación con un usuario interno y éste **NO** responde.

Supuestos:

- El usuario interno solicitado se encuentra disponible.
- Se ha notificado al usuario que alguien lo solicita, utilizando un mensaje de tipo pop-up.
- El usuario no responde a la solicitud de comunicación, y el tiempo de espera expira.

Salidas:

- No se establece la comunicación entre el usuario externo y el usuario interno.
- Se presenta un mensaje al usuario externo, indicándole lo sucedido.

Escenario 2.3: Se establece comunicación y se le da acceso al usuario externo.

Supuestos:

- El usuario interno y el usuario externo lograron comunicarse con éxito
- El usuario interno manda a abrir la puerta desde su computadora.

Salidas:

- Se le concede al usuario externo acceso a las oficinas.

- Se añade el registro correspondiente a la bitácora.

Escenario 2.4: El usuario externo deja un mensaje de audio o vídeo al usuario interno.

Supuestos:

- El usuario interno seleccionado no se encuentra disponible.
- Se identifica cuál es el tipo de mensajes almacenados que el usuario interno especificó en su configuración.
- El usuario externo escoge la opción de dejar un mensaje de audio o vídeo al usuario interno.
- Hay espacio suficiente en el servidor para almacenar el mensaje

Salidas:

- El mensaje es almacenado.
- Se envía una notificación vía e-mail al usuario interno.
- Se añade el registro correspondiente a la bitácora.

Escenario 2.5: El usuario externo deja un mensaje de texto al usuario interno.

Supuestos:

- El usuario interno seleccionado no se encuentra disponible.
- Se identifica cuál es el tipo de mensajes almacenados que el usuario interno especificó en su configuración.

- El usuario externo escoge la opción de dejar un mensaje de texto al usuario interno.

Salidas:

- Se envía un e-mail al usuario interno con el mensaje del usuario externo.
- Se añade el registro correspondiente a la bitácora.

Caso de Uso 3: Comunicación usuario interno-servidor

Actores: Persona que solicita establecer comunicación con el servidor desde su máquina (usuario interno).

Escenario 3.1: Acceso exitoso del usuario interno al sistema.

Supuestos:

- El usuario interno ha ingresado su usuario y contraseña.
- El usuario y contraseña son válidos, y la máquina desde la que se conecta corresponde a alguna de las asignadas al usuario interno.
- Tanto el usuario como la máquina se encuentran habilitados.
- El usuario ingresa dentro del horario de actividad establecido.

Salidas:

- El sistema concede acceso al usuario interno.

- Se despliega en la ventana principal la lista de mensajes recibidos.
- Se actualiza el estado de conexión del usuario interno.
- Se envía una notificación al usuario interno conectado desde la página Web, indicándole que se acaba de conectar desde la aplicación instalada en su computadora.

Escenario 3.2: Revisión de los mensajes recibidos.

Supuestos:

- El usuario interno tiene uno o más mensajes recibidos.
- El usuario interno selecciona el mensaje nuevo que desea revisar, y presiona el botón "Descargar".

Salidas:

- El mensaje es descargado en el directorio local de mensajes del usuario.
- El mensaje es descartado de la lista de mensajes nuevos, y pasa a formar parte de la lista de mensajes almacenados.
- Se elimina el mensaje del servidor.

Nota.- Cuando se descarga un mensaje de audio o vídeo, se crearán dos archivos; el del contenido multimedia, y uno con extensión .txt, que indique datos como fecha, hora, portero desde el que se envío, etc.

Escenario 3.3: Configuración del sistema por el usuario interno.

Supuestos:

- El usuario interno ingresa todos los datos personales requeridos.
- En los datos de la aplicación, el usuario interno especifica el tipo de notificación que desea utilizar (con o sin sonido); si desea recibir una foto adjunta con los mensajes escritos; ingresa el tiempo de inactividad, la observación, en caso de que utilice alguna; el tipo de mensaje que desea usar en la comunicación y en la recepción de mensajes enviados.
- Ingresa la contraseña con la que va a acceder al sistema.

Salidas:

- Los datos son configurados correctamente.

Caso de Uso 4: Comunicación del usuario interno con el sistema usuario.

Actores: Persona que interactúa con el sistema usuario instalado en su máquina (usuario interno).

Escenario 4.1: Cambio manual del estado de conexión.

Supuestos:

- El usuario interno selecciona la opción de cambiar de estado de conexión que presenta la aplicación.

- De la lista de estados disponibles, el usuario interno selecciona el estado que desea.

Salidas:

- El estado de conexión del usuario interno se modifica, y aparece el icono correspondiente al nuevo estado.

Escenario 4.2: Eliminación de mensajes de la lista de almacenados.

Supuestos:

- El usuario interno selecciona el mensaje que desea eliminar de la lista de mensajes almacenados, y presiona el botón "Eliminar".
- El mensaje seleccionado ya ha sido leído.

Salidas:

- El mensaje es eliminado de la lista de mensajes almacenados, y del disco duro de la máquina.

Caso de Uso 5: Administración de usuarios internos.

Actores: Persona que se encarga de la creación, modificación, eliminación y configuración de los elementos que constituyen el sistema (administrador).

Escenario 5.1: Creación exitosa de un nuevo usuario interno.

Supuestos:

- Se ingresan los datos requeridos para la creación de un nuevo usuario interno (nombre, e-mail, usuario, contraseña, clave de acceso, etc.)
- Las categorías a las que pertenece el usuario, el horario, y las máquinas o porteros que han de asignársele, han sido registrados con anterioridad.
- El nombre, el usuario y la clave de acceso ingresados son únicos en el sistema.

Salidas:

- El nuevo usuario interno es registrado exitosamente.
- Se crea la carpeta de mensajes recibidos para el usuario.

Escenario 5.2: Eliminación exitosa de un usuario interno.

Supuestos:

- El administrador selecciona el usuario interno que desea eliminar.
- El usuario seleccionado no está conectado.

Salidas:

- El usuario es eliminado del sistema.
- La carpeta de mensajes recibidos del usuario es eliminada.

Escenario 5.3: Modificación exitosa de un usuario interno.

Supuestos:

- El administrador selecciona el usuario interno cuyos datos desea modificar, e ingresa los nuevos datos válidos correspondientes.
- El usuario seleccionado, no se encuentra conectado.

Salidas:

- Los datos modificados del usuario son almacenados.

Caso de Uso 6: Administración de máquinas

Actores: Persona que se encarga de la creación, modificación, eliminación y configuración de los elementos que constituyen el sistema (administrador).

Escenario 6.1: Creación exitosa de una nueva máquina.

Supuestos:

- Se ingresan los datos requeridos para la creación de una nueva máquina: nombre, MAC, y estado (habilitada o deshabilitada).
- El nombre y la dirección MAC de la máquina son únicos en el sistema.

Salidas:

- La nueva máquina es registrada exitosamente.

Escenario 6.2: Eliminación de una máquina cuando está asignada a uno o más usuarios internos.

Supuestos:

- El administrador selecciona la máquina que desea eliminar.
- La máquina seleccionada no se encuentra conectada.
- La máquina seleccionada está asignada a uno o más usuarios internos.

Salidas:

- Mensaje de Advertencia
- La máquina es eliminada del sistema.

Escenario 6.3: Modificación exitosa de una máquina.

Supuestos:

- El administrador selecciona la máquina cuyos datos desea modificar, e ingresa los nuevos datos válidos correspondientes.
- La máquina no se encuentra conectada.

Salidas:

- Los datos modificados de la máquina son almacenados.

Caso de Uso 7: Administración de porteros

Actores: Persona que se encarga de la creación, modificación, eliminación y configuración de los elementos que constituyen el sistema (administrador).

Escenario 7.1: Creación exitosa de un nuevo portero.

Supuestos:

- Se ingresan los datos requeridos para la creación de un nuevo portero (nombre, MAC, etc.).
- El nombre del portero es único en el sistema.
- Las máquinas y el horario asignado al portero ya han sido registrados con anterioridad.

Salidas:

- El nuevo portero es registrado exitosamente.

Escenario 7.2: Eliminación exitosa de un portero.

Supuestos:

- El administrador selecciona el portero que desea eliminar.
- El portero seleccionado no se encuentra conectado.

Salidas:

- El portero es eliminado del sistema.
- Las maquinas que no pertenecen a otro portero son deshabilitadas.

Escenario 7.3: Modificación exitosa de un portero.

Supuestos:

- El administrador selecciona el portero cuyos datos desea modificar, e ingresa los nuevos datos válidos correspondientes.
- El portero no se encuentra conectado.

Salidas:

- Los datos modificados del portero son almacenados.

Caso de Uso 8: Administración de horarios

Actores: Persona que se encarga de la creación, modificación, eliminación y configuración de los elementos que constituyen el sistema (administrador).

Escenario 8.1: Creación exitosa de un nuevo horario.

Supuestos:

- Se ingresan los datos requeridos para la creación de un nuevo horario (nombre, fecha de inicio, fecha de fin, etc.).
- El nombre del horario es único en el sistema.

Salidas:

- El nuevo horario es registrado exitosamente.

Escenario 8.2: Eliminación de un horario asignado a uno o más usuarios internos o porteros.

Supuestos:

- El administrador selecciona el horario que desea eliminar.
- El horario seleccionado no está asignado a ninguno de los porteros o usuarios conectados.
- El horario seleccionado está asignado a uno o más usuarios internos o porteros.

Salidas:

- Mensaje de Advertencia
- El horario es eliminado del sistema.

Escenario 8.3: Modificación exitosa de un horario.

Supuestos:

- El administrador selecciona el horario cuyos datos desea modificar, e ingresa los nuevos datos válidos correspondientes.
- El horario seleccionado no está asignado a ninguno de los porteros o usuarios conectados.

Salidas:

- Los datos modificados del horario son almacenados.

Caso de Uso 9: Administración de categorías

Actores: Persona que se encarga de la creación, modificación, eliminación y configuración de los elementos que constituyen el sistema (administrador).

Escenario 9.1: Creación exitosa de una nueva categoría.

Supuestos:

- Se ingresan los datos requeridos para la creación de una nueva categoría: nombre y descripción.
- El nombre de la categoría es único en el sistema.

Salidas:

- La nueva categoría es registrada exitosamente.

Escenario 9.2: Eliminación de una categoría que está asignada a uno o más usuarios internos.

Supuestos:

- El administrador selecciona la categoría que desea eliminar.
- La categoría está asignada a uno o más usuarios internos.
- La categoría no está asignada a ninguno de los usuarios conectados.

Salidas:

- Mensaje de advertencia
- La categoría es eliminada del sistema.

Escenario 9.3: Modificación exitosa de una categoría.

Supuestos:

- El administrador selecciona la categoría cuyos datos desea modificar, e ingresa los nuevos datos válidos correspondientes.
- La categoría no está asignada a ninguno de los usuarios conectados.

Salidas:

- Los datos modificados de la categoría son almacenados.

Caso de Uso 10: Administración de estados

Actores: Persona que se encarga de la creación, modificación, eliminación y configuración de los elementos que constituyen el sistema (administrador).

Escenario 10.1: Creación exitosa de un nuevo estado.

Supuestos:

- Se ingresan los datos requeridos para la creación de un nuevo estado (nombre, descripción, etc.).
- El nombre del nuevo estado es único en el sistema.

Salidas:

- El nuevo estado es registrado exitosamente.

Escenario 10.2: Eliminación exitosa de un estado.

Supuestos:

- El administrador selecciona el estado que desea eliminar.
- El estado no está asignado a ninguno de los usuarios conectados.
- El estado no es ninguno de los tres estados principales del sistema.

Salidas:

- El estado es eliminado del sistema.

Escenario 10.3: Modificación exitosa de un estado.

Supuestos:

- El administrador selecciona el estado cuyos datos desea modificar, e ingresa los nuevos datos válidos correspondientes.
- El estado no está asignado a ninguno de los usuarios conectados.
- El estado no es ninguno de los tres estados principales del sistema.

Salidas:

- Los datos modificados del estado son almacenados.

Caso de Uso 11: Configuración del sistema por el administrador

Actores: Persona que se encarga de la creación, modificación, eliminación y configuración de los elementos que constituyen el sistema (administrador).

Escenario 11.1: Configuración exitosa del sistema.

Supuestos:

- Se ingresan todos los datos requeridos para la configuración del sistema. (dirección IP del servidor de correo, usuario del administrador, y contraseña, etc.).

Salidas:

- Se almacena la nueva configuración del sistema.

Caso de Uso 12: Administración del Log

Actores: Persona que se encarga de la creación, modificación, eliminación y configuración de los elementos que constituyen el sistema (administrador).

Escenario 12.1: Consulta del Log

Supuestos:

- Los datos requeridos han sido especificados en el criterio de búsqueda.

Salidas:

- Se presenta el reporte solicitado.

Caso de Uso 13: Comunicación portero-servidor

Actores: Persona que se encarga de la creación, modificación, eliminación y configuración de los elementos que constituyen el sistema (administrador).

Escenario 13.1: Inicialización exitosa de un portero.

Supuestos:

- La máquina que solicita acceso es uno de los porteros registrados del sistema.
- El portero está habilitado y se encuentra dentro del horario de actividad que se le ha asignado.

Salidas:

- Se presenta la pantalla principal del portero.

Caso de Uso 14: Comunicación administrador-servidor

Actores: Persona que se encarga de la creación, modificación, eliminación y configuración de los elementos que constituyen el sistema (administrador).

Escenario 14.1: Solicitud de acceso exitosa del administrador.

Supuestos:

- El administrador solicita acceso al sistema, ingresando un usuario y una contraseña válidos.

Salidas:

- Se le concede al administrador acceso al sistema.
- Se presenta la pantalla principal del sistema mostrando la lista de porteros.

Escenario 14.2: Búsqueda de usuarios internos.

Supuestos:

- El administrador ingresa el nombre completo o parte del nombre del usuario interno que desea consultar.

Salidas:

- Los resultados de la búsqueda son presentados por pantalla.

Escenario 14.3: Consulta de un usuario interno.

Supuestos:

- El administrador solicita consultar los datos de un usuario específico.

Salidas:

- Los resultados de la consulta son presentados por pantalla.

Escenario 14.4: Consulta exitosa de la contraseña y clave de acceso de un usuario interno.

Supuestos:

- El administrador solicita consultar la contraseña y clave de acceso de un usuario específico.
- El administrador ingresa nuevamente su usuario y su contraseña correctamente.

Salidas:

- Los resultados de la consulta son presentados por pantalla.

Caso de Uso 15: Comunicación usuario interno-servidor vía Internet

Actores: Persona que solicita establecer comunicación con el servidor desde un navegador de Internet (usuario interno).

Escenario 15.1: Acceso exitoso del usuario interno al sistema vía Internet.

Supuestos:

- El usuario interno ha ingresado su usuario y su contraseña correctamente
- El usuario y la contraseña son válidos, y el usuario se encuentra habilitado.
- El usuario interno también se encuentra conectado a través de la aplicación instalada en su computadora.

Salidas:

- El sistema concede acceso al usuario interno.
- Se abre una página presentando los mensajes pendientes de revisión.
- Se muestra una notificación al usuario interno conectado desde la aplicación instalada en su computadora, indicándole que se acaba de conectar vía Web.

Escenario 15.2: Eliminación de mensajes vía Internet.

Supuestos:

- El usuario interno selecciona de entre la lista de mensajes, el mensaje que desea eliminar.
- El mensaje ya ha sido leído.

Salidas:

- El mensaje es eliminado tanto de la lista de mensajes, como del servidor.

Escenario 15.3: Configuración del sistema por el usuario interno vía Internet.

Supuestos:

- El usuario interno ingresa sus datos personales.
- En los datos de la aplicación, el usuario interno especifica si desea recibir una foto adjunta con los mensajes escritos, la observación, en caso de que utilice alguna; el tipo de mensaje en la recepción de mensajes enviados.
- Ingresa la contraseña con la que va a acceder al sistema.

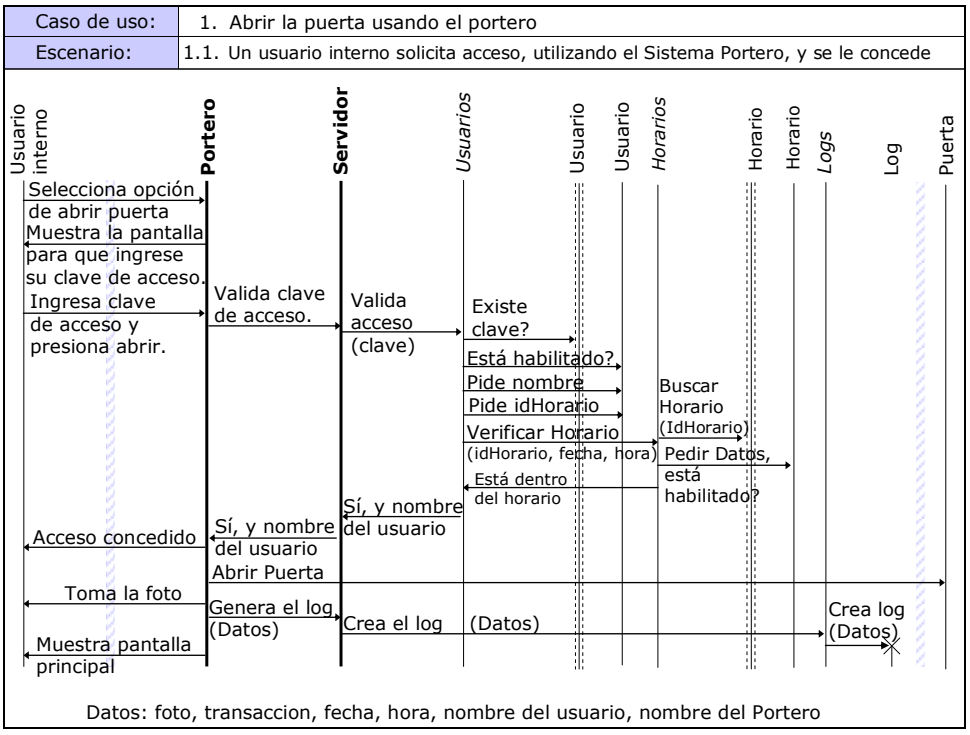
Salidas:

- Los datos son guardados correctamente.

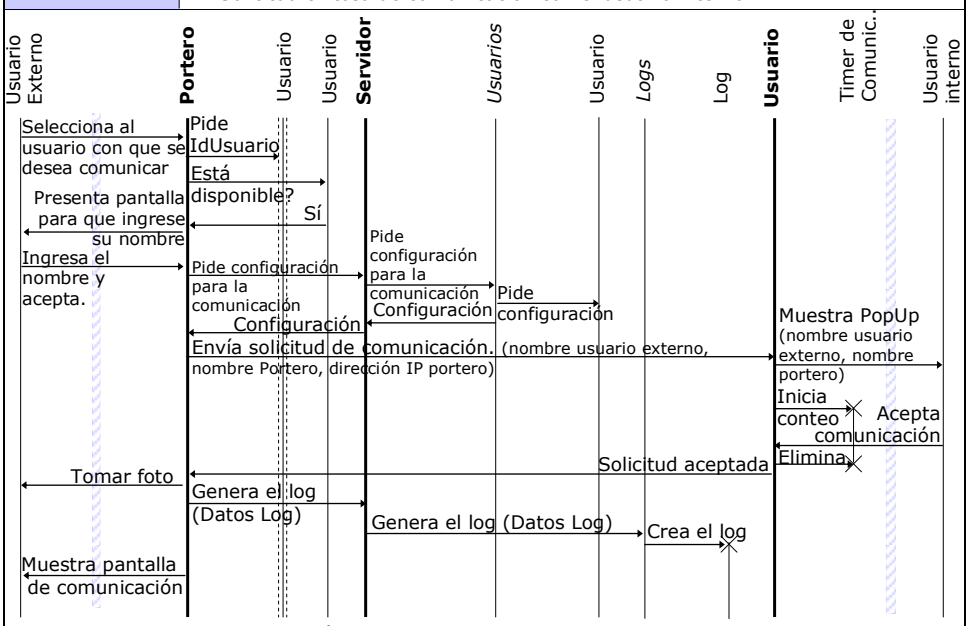
APÉNDICE B

DIAGRAMAS DE INTERACCIÓN DE OBJETOS DE ANÁLISIS

Los siguientes diagramas de interacción de objetos de análisis corresponden a los escenarios descritos en el apéndice A.

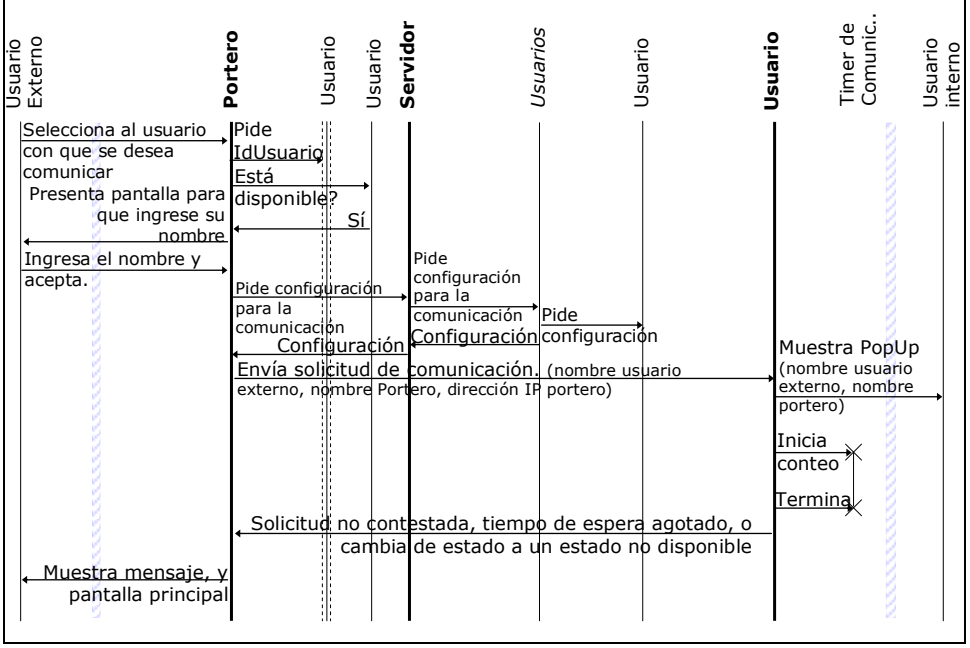


Caso de uso:	2. Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	2.1. Solicitud exitosa de comunicación con el usuario interno.

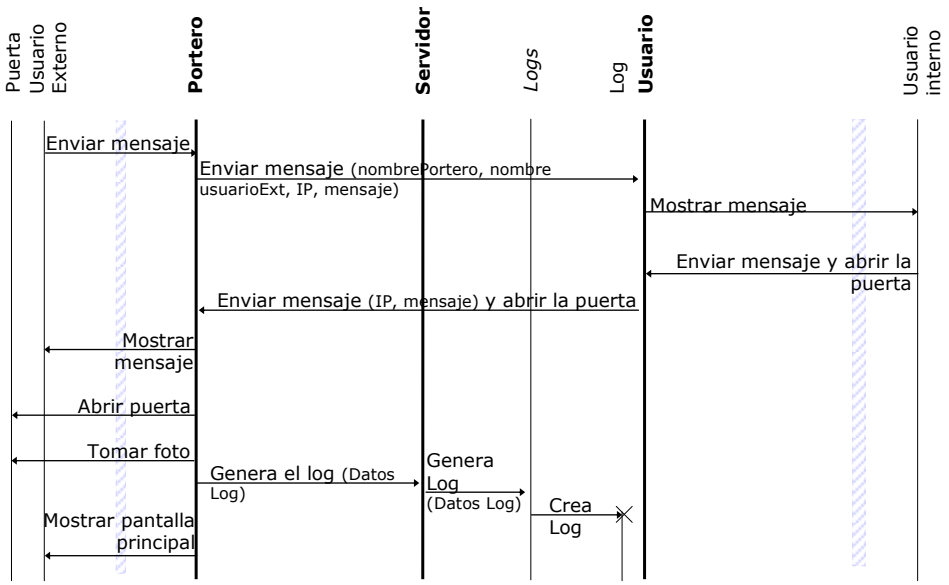


Datos Log: foto, tipo de transacción, fecha, hora, nombre del usuario, nombre del Portero, nombre del usuario externo

Caso de uso:	2. Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	2.2. Un usuario externo solicita comunicación con un usuario interno y **NO** responde

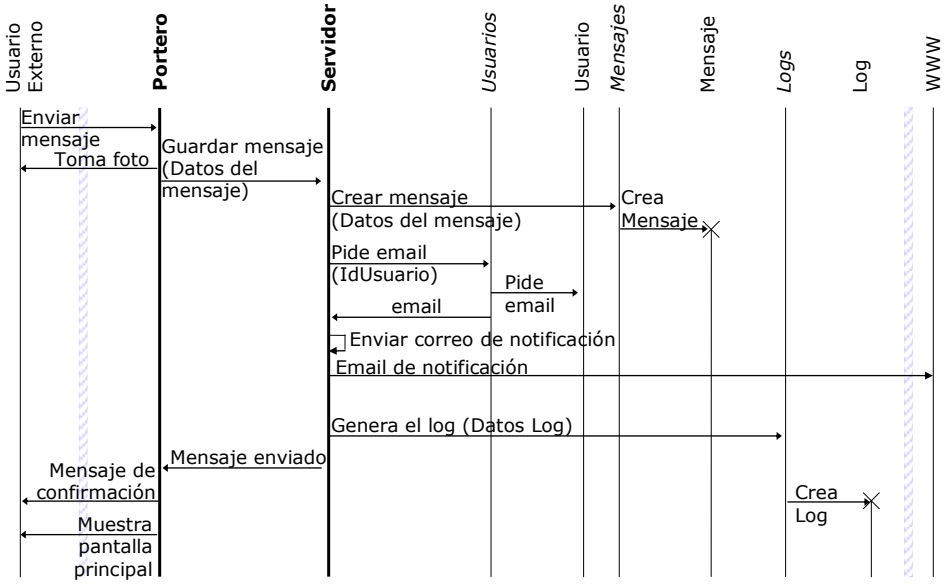


Caso de uso:	2. Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	2.3. Se establece comunicación y se le da acceso al usuario externo



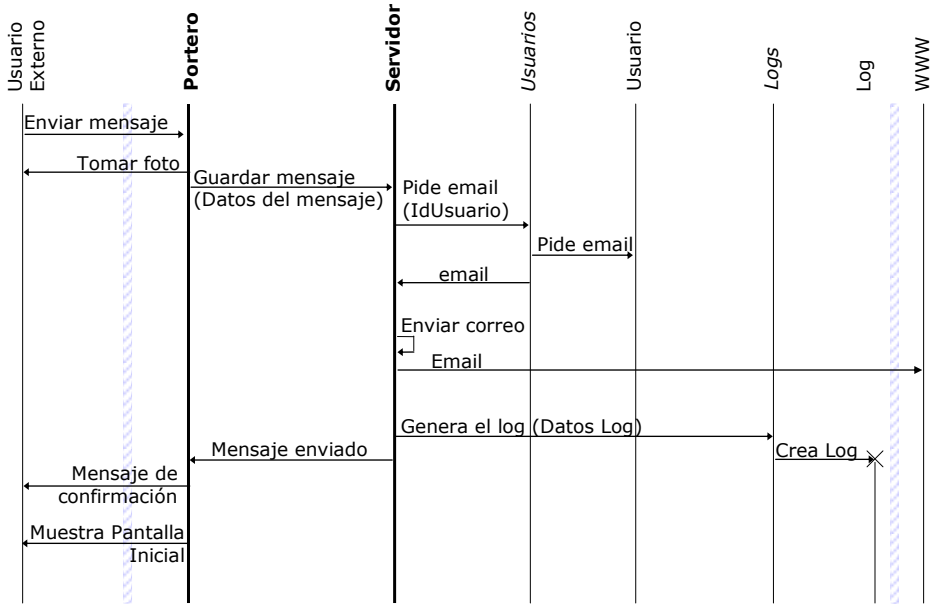
Datos Log: foto, transacción, fecha, hora, nombre del usuario interno, nombre del Portero, nombre del usuario externo

Caso de uso:	2. Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	2.4. El usuario externo deja un mensaje de vídeo o de audio al usuario interno



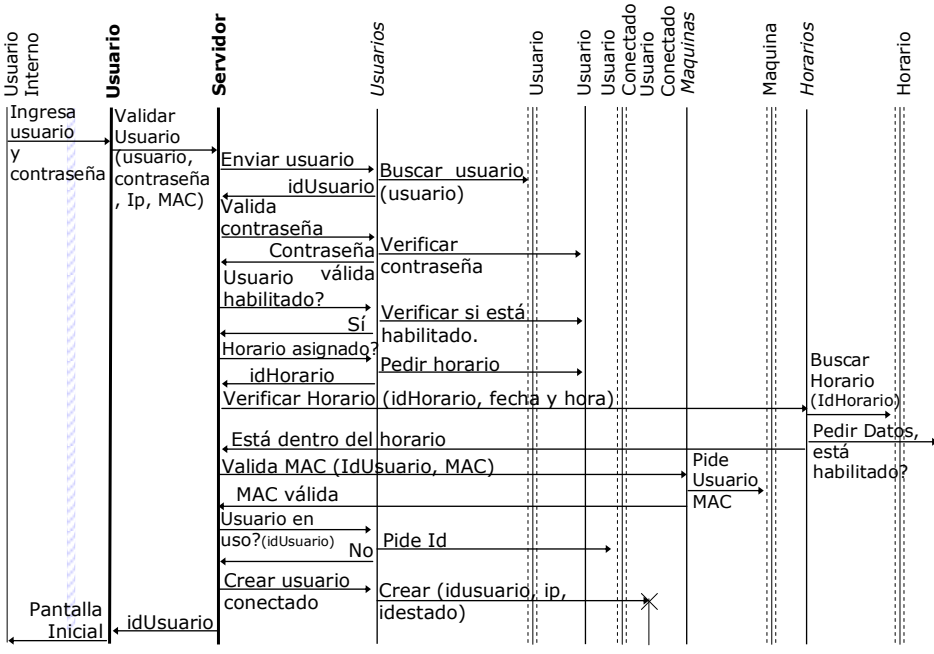
Datos Log: foto, transacción, fecha, hora, nombre del usuario, nombre del portero, usuario externo
Datos del mensaje: mensaje, idUsuario, tipo de mensaje, idPortero, nombre usuario externo, foto

Caso de uso:	2. Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	2.5. El usuario externo deja un mensaje de texto al usuario interno

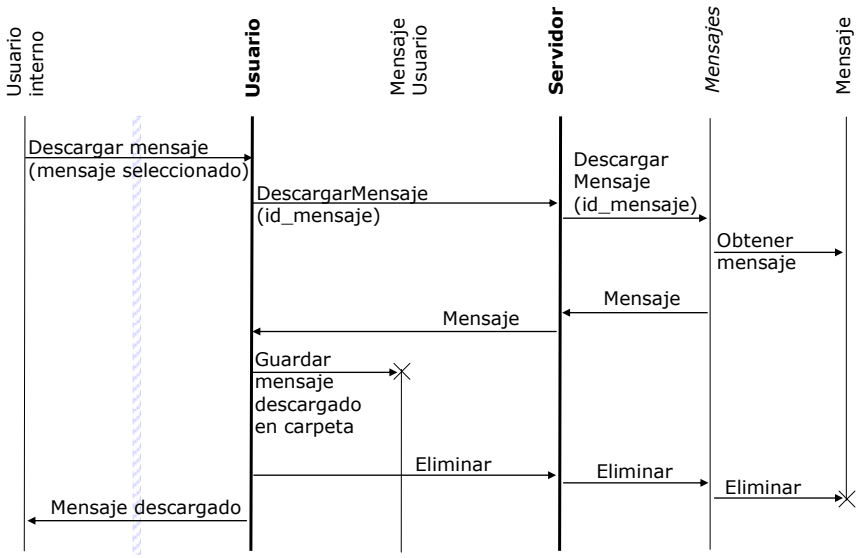


Datos Log: foto, transacción, fecha, hora, nombre del usuario, nombre del portero, usuario externo
Datos del mensaje: mensaje, idUsuario, tipo de mensaje, idPortero, usuario externo, foto

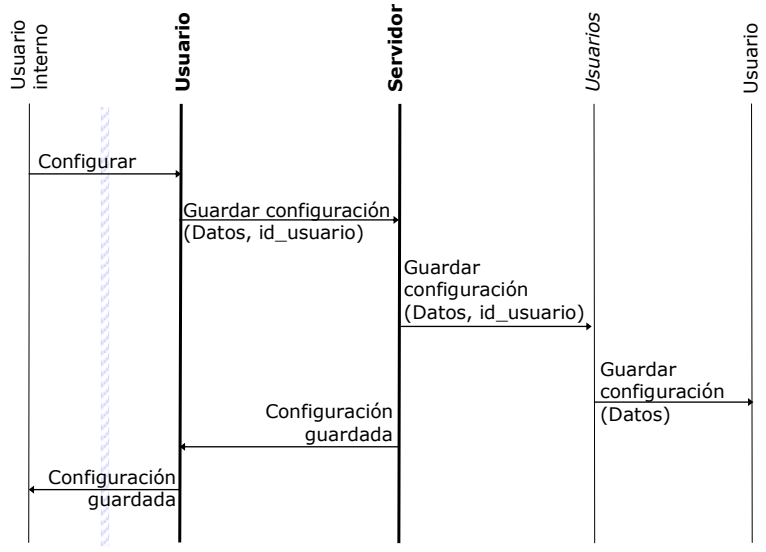
Caso de uso:	3. Comunicación usuario_interno - servidor
Escenario:	3.1. Acceso exitoso del usuario interno al sistema



Caso de uso:	3. Comunicación usuario interno - servidor
Escenario:	3.2. Revisión de los mensajes recibidos

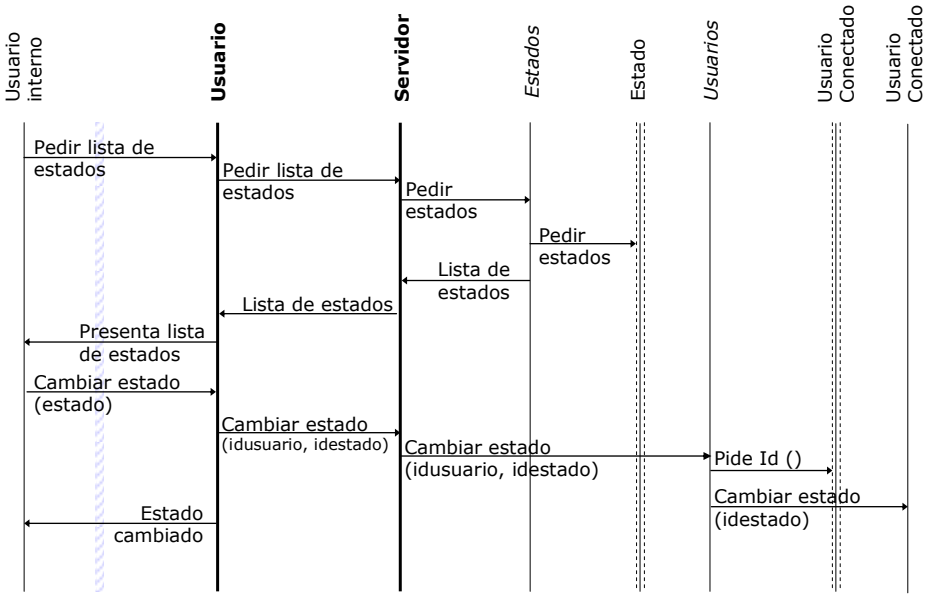


Caso de uso:	3. Comunicación usuario_interno - servidor
Escenario:	3.3. Configuración del sistema por el usuario interno

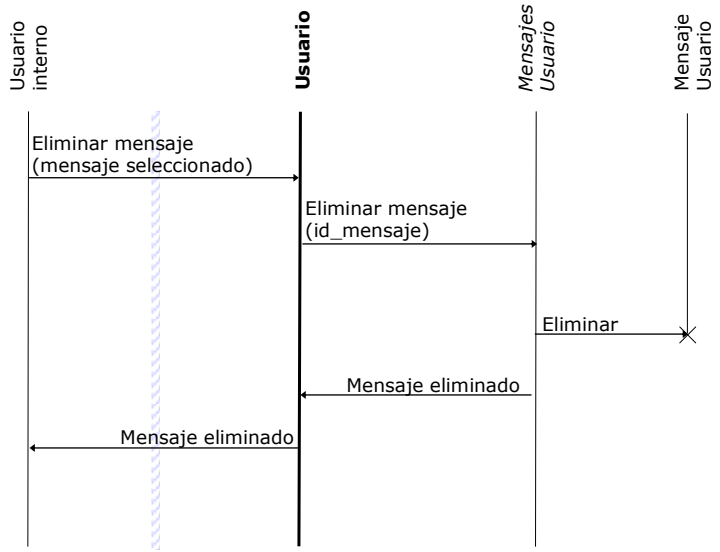


Datos: Nombre, cargo, teléfonos, e-mail, foto, contraseña, notificación, foto en Correo, tiempo de inactividad, observación, tipo mensaje para comunicación, tipo mensaje a guardar por el portero, ver usuario externo en comunicaciones de texto y audio.

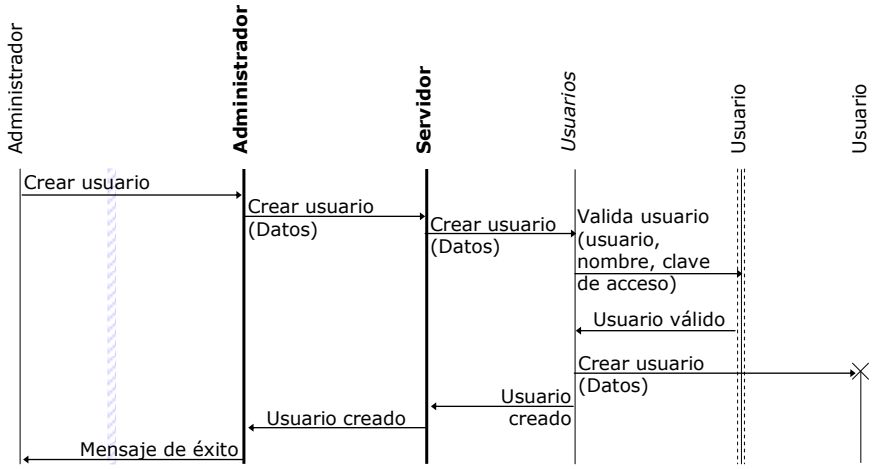
Caso de uso:	4. Comunicación del usuario interno con el sistema usuario.
Escenario:	4.1. Cambio manual del estado de conexión



Caso de uso:	4. Comunicación del usuario interno con el sistema usuario.
Escenario:	4.2. Eliminación de mensajes de la lista de almacenados

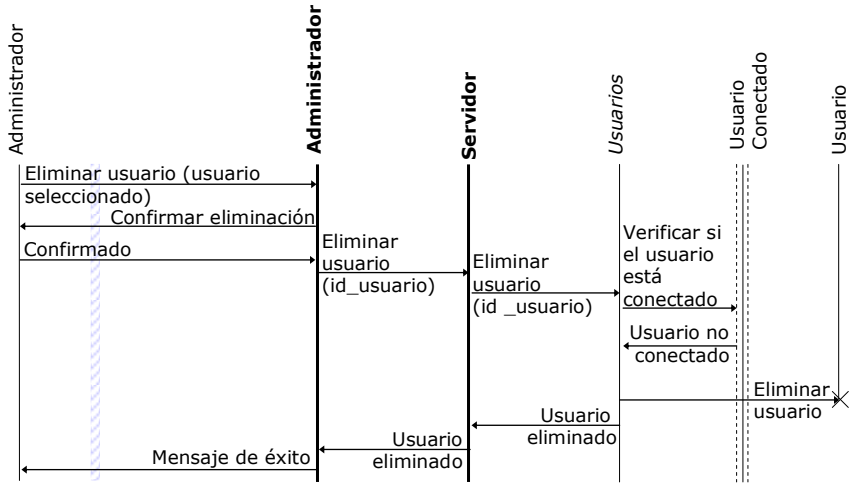


Caso de uso:	5. Administración de usuarios internos
Escenario:	5.1. Creación exitosa de un nuevo usuario interno



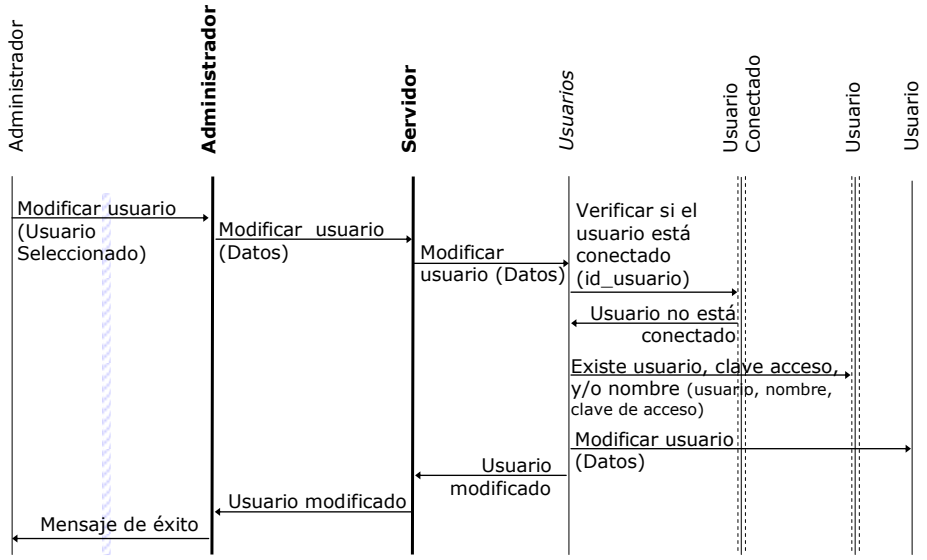
Datos: Nombre, cargo, teléfonos, e-mail, usuario, contraseña, clave de acceso, lista de categorías asociadas, lista de máquinas o porteros relacionados, habilitado

Caso de uso:	5. Administración de usuarios internos
Escenario:	5.2. Eliminación exitosa de un usuario interno



Si el usuario está conectado, no se lo elimina.

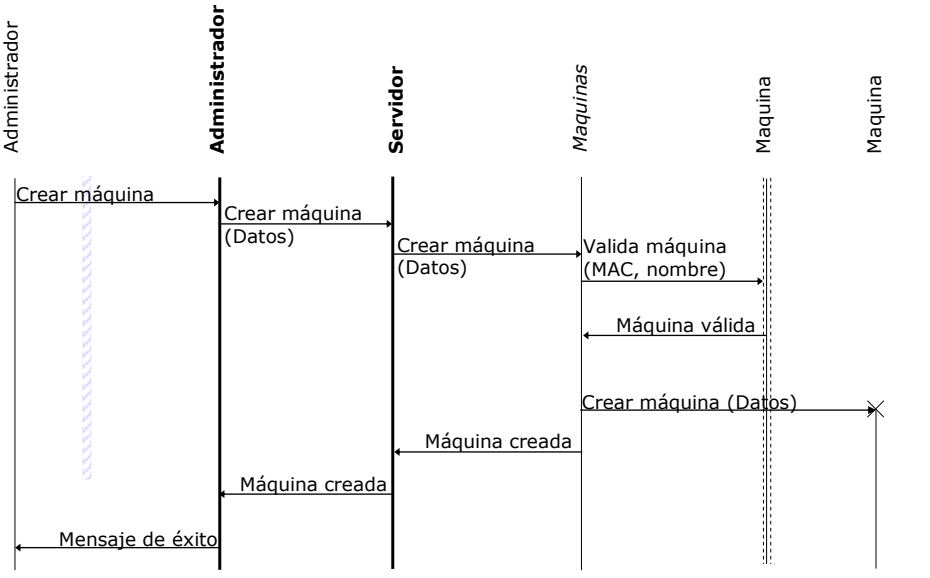
Caso de uso:	5. Administración de usuarios internos
Escenario:	5.3. Modificación exitosa de un usuario interno



Datos: Nombre, cargo, teléfonos, e-mail, usuario, contraseña, lista de categorías asociadas, lista de máquinas o porteros asociados, clave de acceso, habilitado.

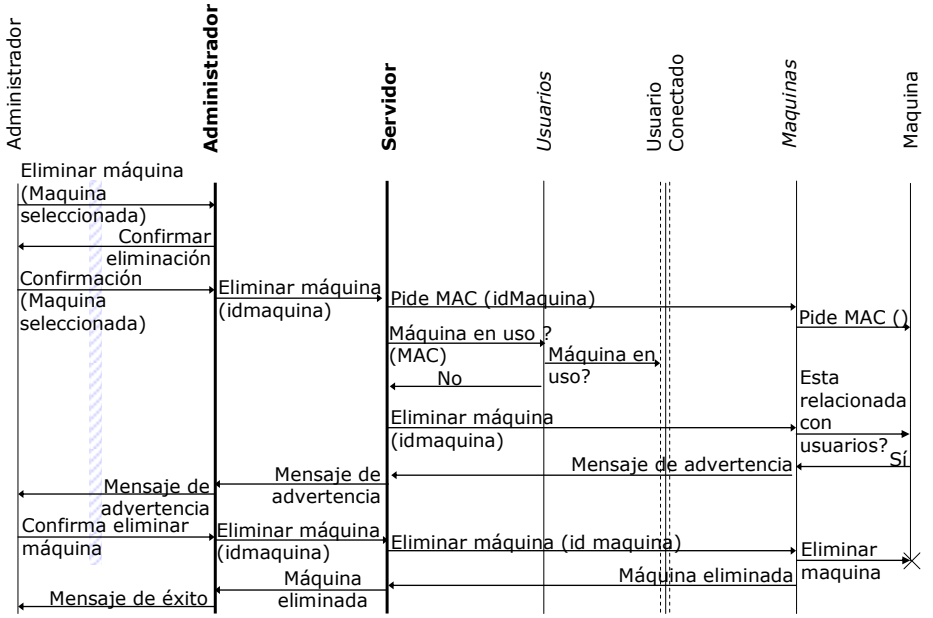
Si el usuario está conectado, no se lo elimina.

Caso de uso:	6. Administración de máquinas
Escenario:	6.1. Creación exitosa de una nueva máquina



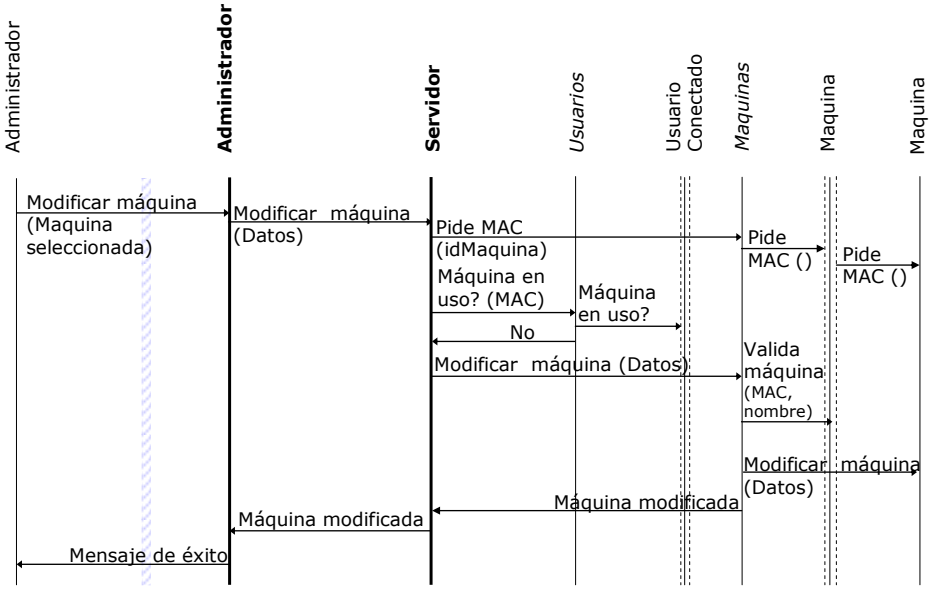
Datos: Nombre, MAC, Habilitada (deshabilitada por defecto)

Caso de uso:	6. Administración de máquinas
Escenario:	6.2. Eliminación de una máquina cuando esta asignada a uno o más usuarios



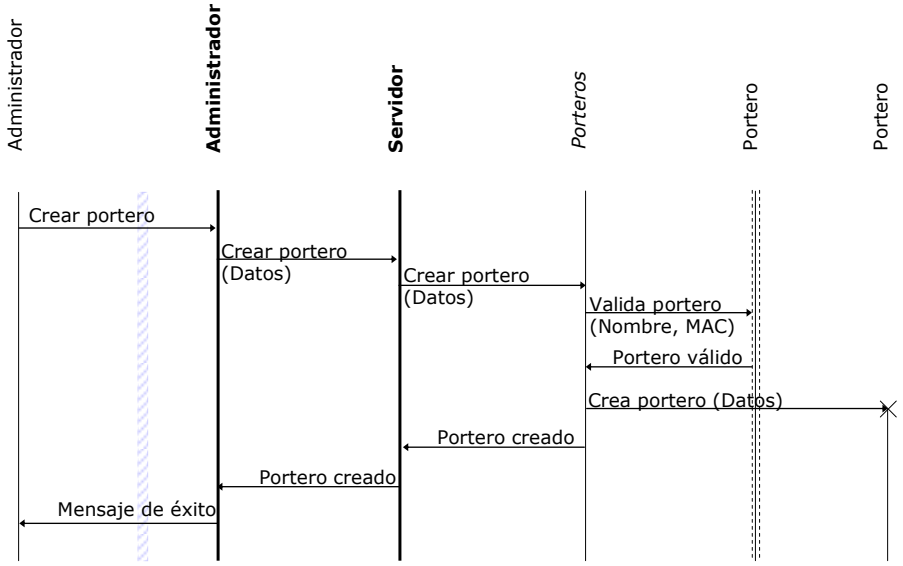
Si la máquina está en uso no puede ser eliminada, si no está relacionada se la elimina.

Caso de uso:	6. Administración de máquinas
Escenario:	6.3. Modificación exitosa de una máquina



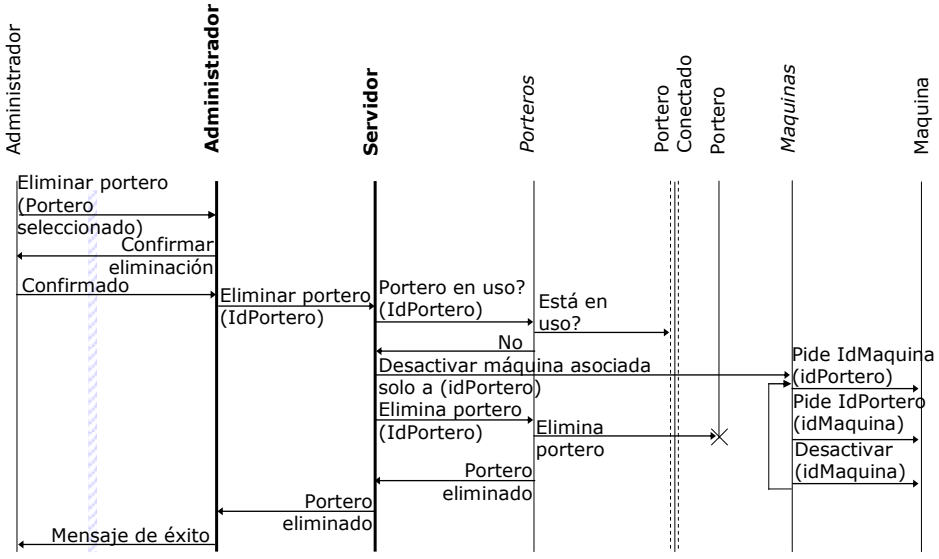
Datos: Nombre, MAC, Habilitado
Si la máquina está en uso no puede ser modificada; es modificada si la MAC y el nombre no existen

Caso de uso:	7. Administración de Porteros
Escenario:	7.1. Creación exitosa de un nuevo portero



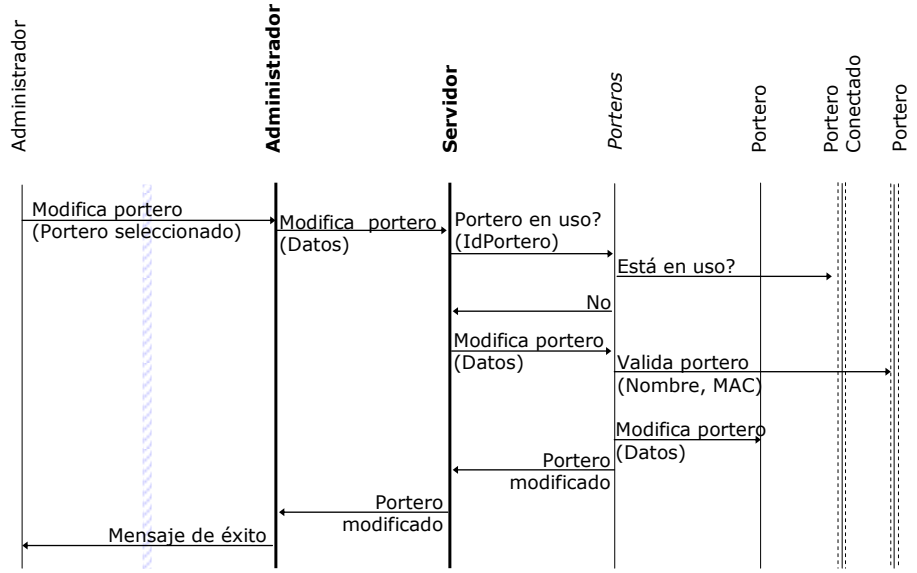
Datos: nombre, descripción, MAC, habilitado, horario asignado, lista de máquinas asignadas

Caso de uso:	7. Administración de Porteros
Escenario:	7.2. Eliminación exitosa de un portero



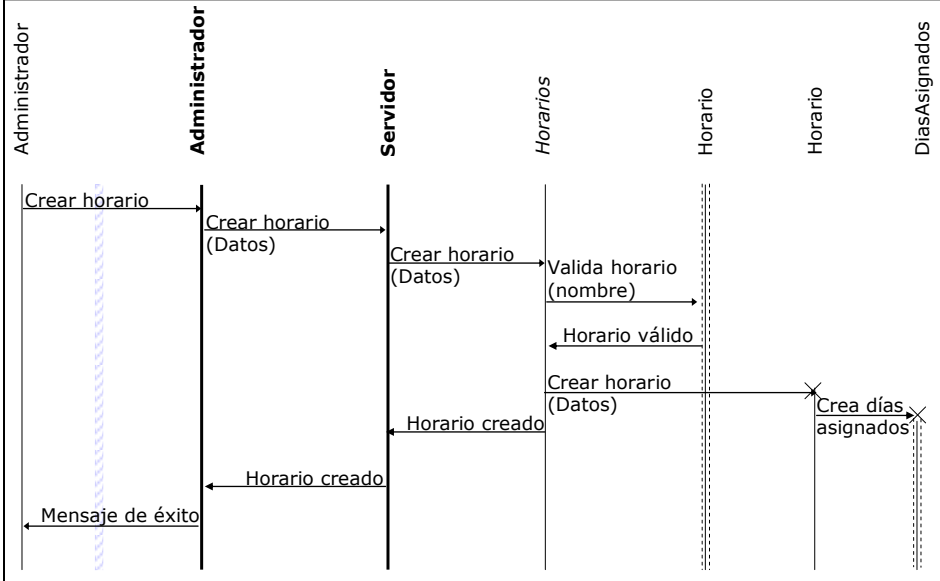
Si el portero está en uso no puede ser eliminado

Caso de uso:	7. Administración de Porteros
Escenario:	7.3. Modificación exitosa de un portero



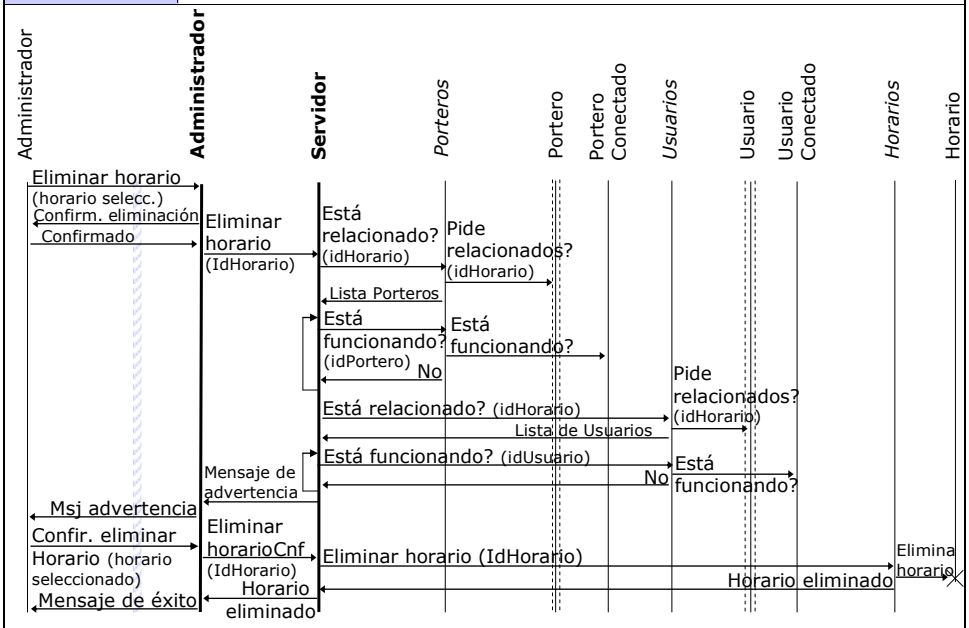
Datos: nombre, descripción, MAC, estado, horario asignado, lista de máquinas asignadas.
Si el portero está en uso, o la MAC o el nombre ya existen no puede ser modificado

Caso de uso:	8. Administración de Horarios
Escenario:	8.1. Creación exitosa de un nuevo horario



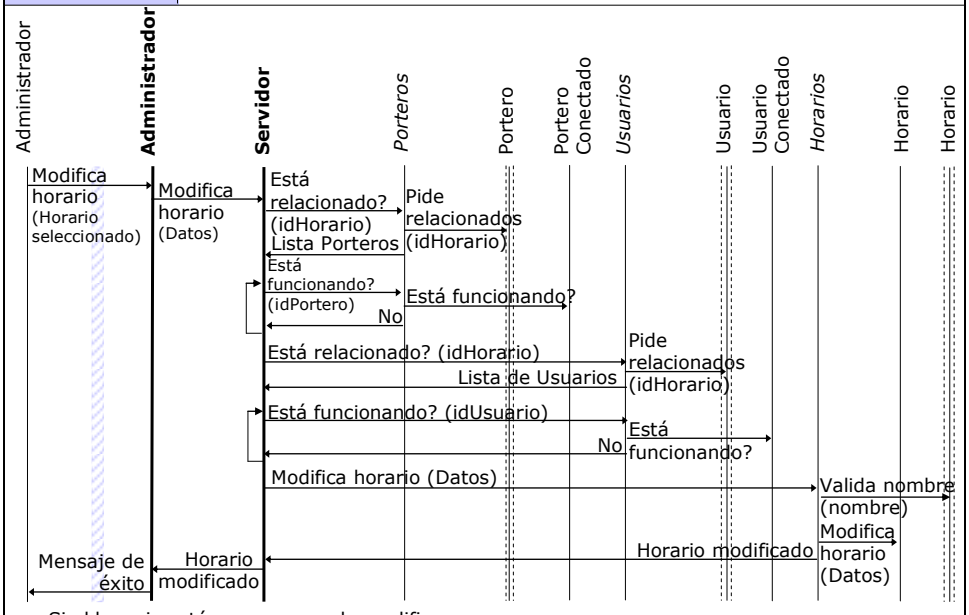
Datos: Nombre, fecha de inicio, fecha de finalización, habilitado, Dias asignados (día, hora inicio, hora fin).

Caso de uso:	8. Administración de Horarios
Escenario:	8.2. Eliminación de un horario asignado a uno o más usuarios internos o porteros



Si no está relacionado, lo elimina. Si está en uso no lo elimina.

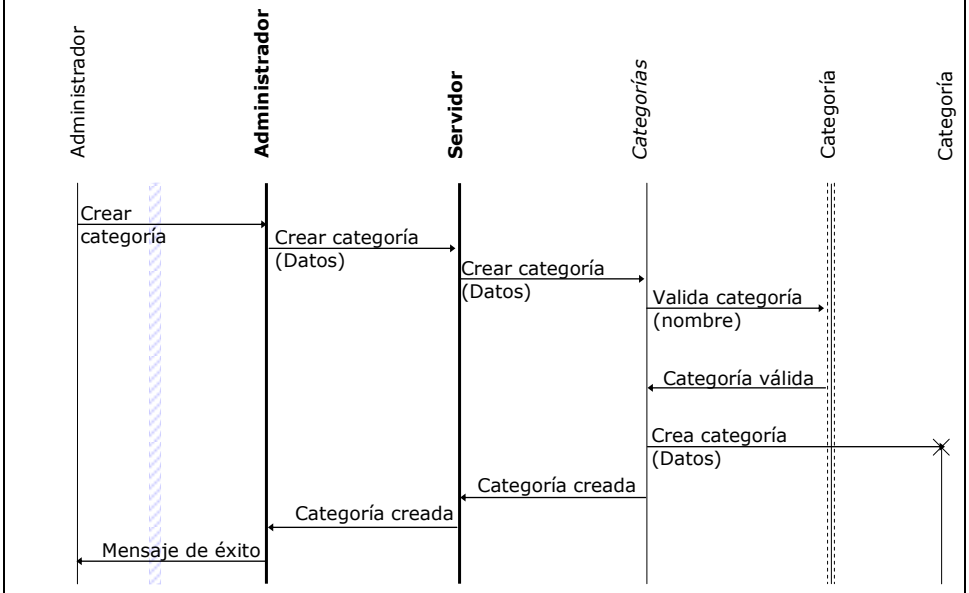
Caso de uso:	8. Administración de Horarios
Escenario:	8.3. Modificación exitosa de un horario



Si el horario está en uso no se lo modifica.

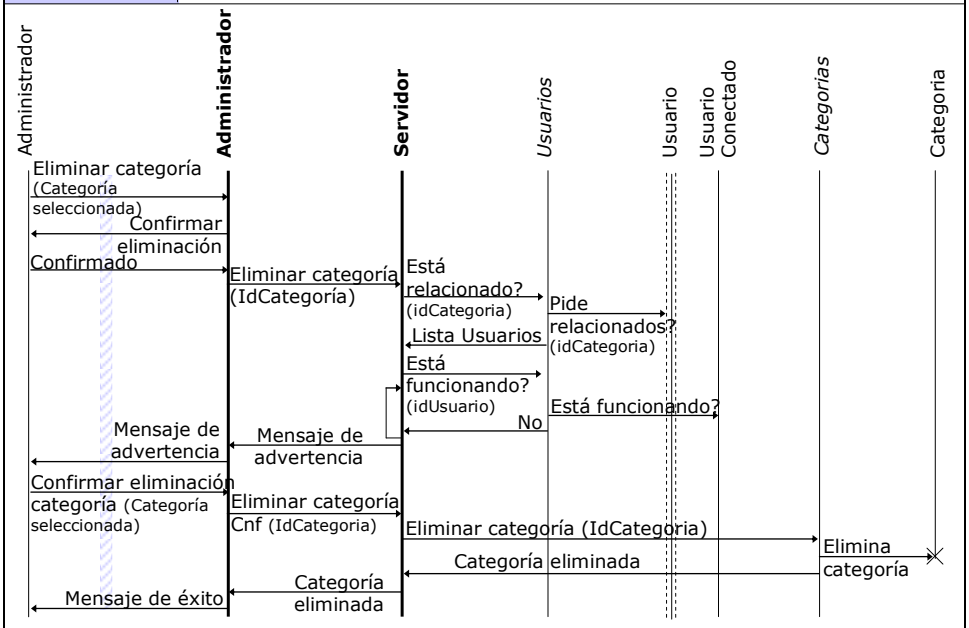
Datos: Nombre, fecha inicio, fecha fin, habilitado, días asignados (día, hora inicio, hora fin).

Caso de uso:	9. Administración de Categorías
Escenario:	9.1. Creación exitosa de una nueva categoría



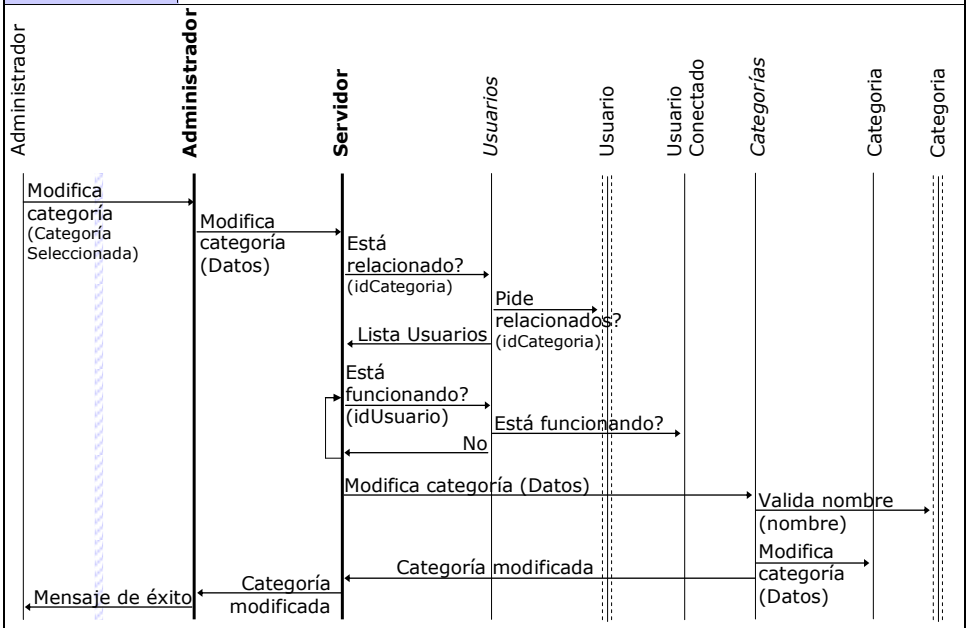
Datos: Nombre, descripción

Caso de uso:	9. Administración de Categorías
Escenario:	9.2. Eliminación de una categoría que está asignada a uno o más usuarios internos



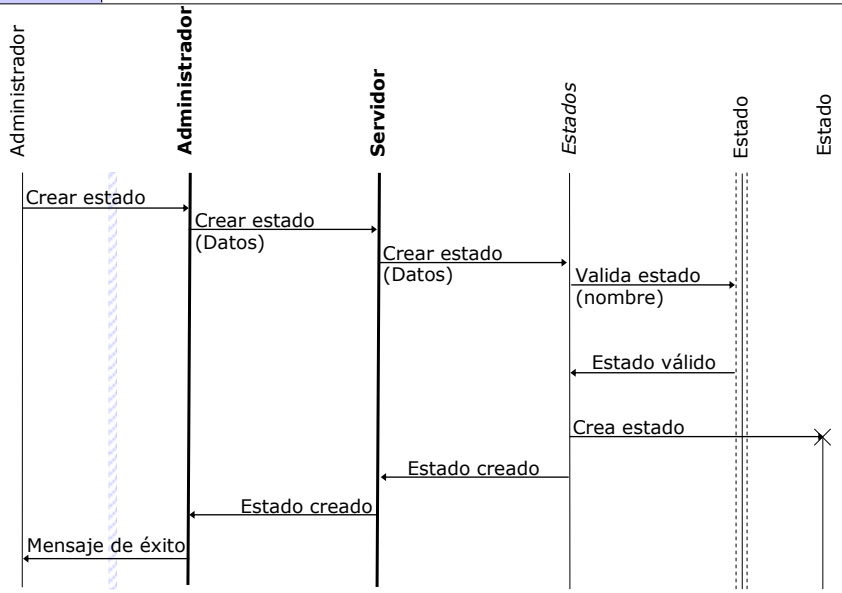
Si la categoría está en uso no se la elimina

Caso de uso:	9.	Administración de Categorías
Escenario:	9.3.	Modificación exitosa de una categoría



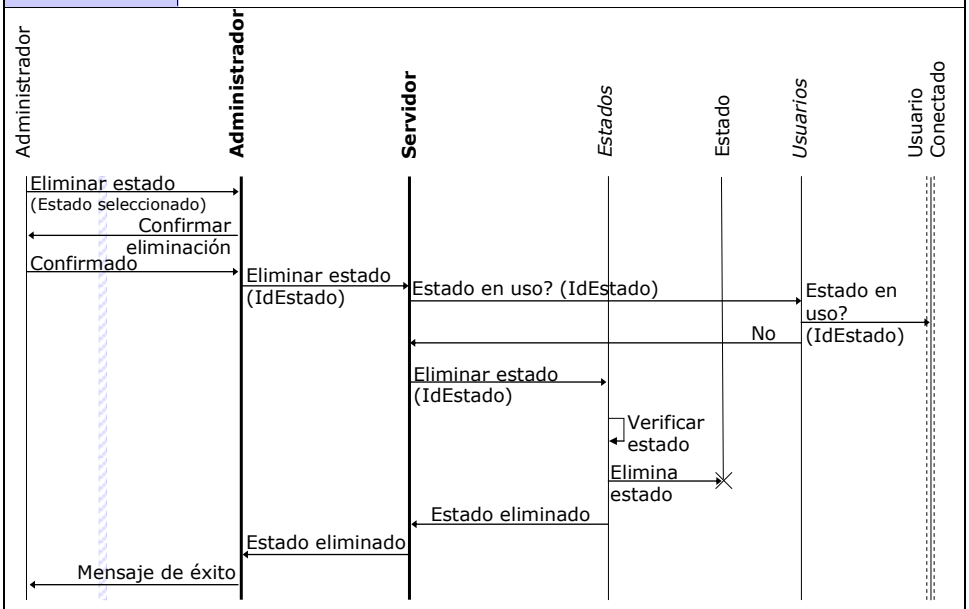
Si está en uso no se la elimina. Datos: Nombre, descripción

Caso de uso:	10.	Administración de Estados
Escenario:	10.1.	Creación exitosa de un nuevo estado



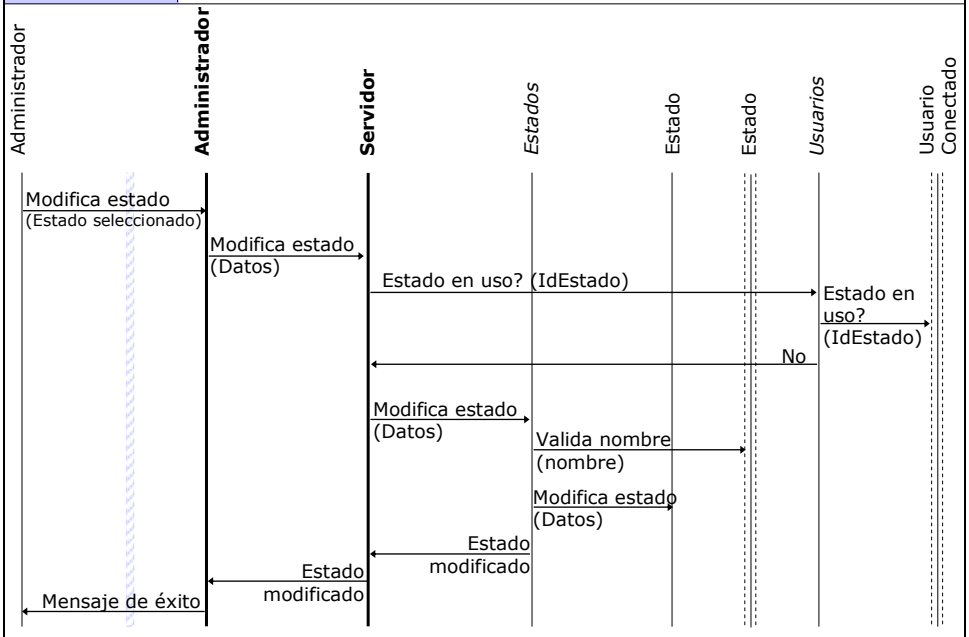
Datos: Nombre, descripción, imagen, habilitado, activo (si es verdadero, el usuario puede recibir solicitudes de comunicación al encontrarse en este estado)

Caso de uso:	10. Administración de Estados
Escenario:	10.2. Eliminación exitosa de un estado



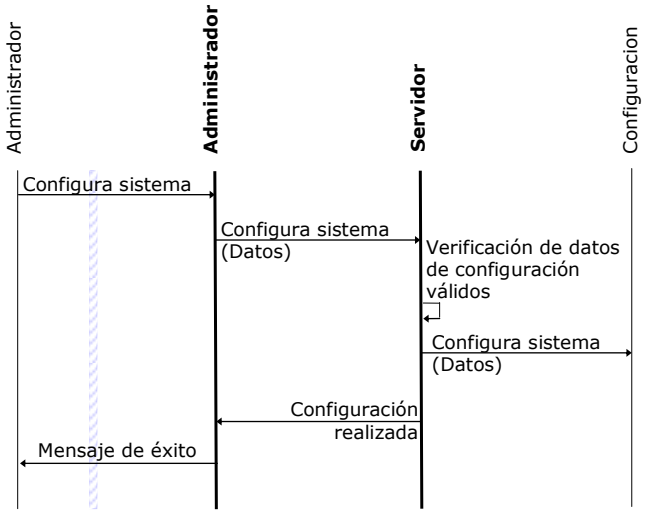
Si está en uso no se lo elimina. Verificar estado: Verifica que no se trate del estado "disponible", "ausente" o "desconectado" ya que estos son los estados básicos del sistema.

Caso de uso:	10. Administración de Estados
Escenario:	10.3. Modificación exitosa de un estado



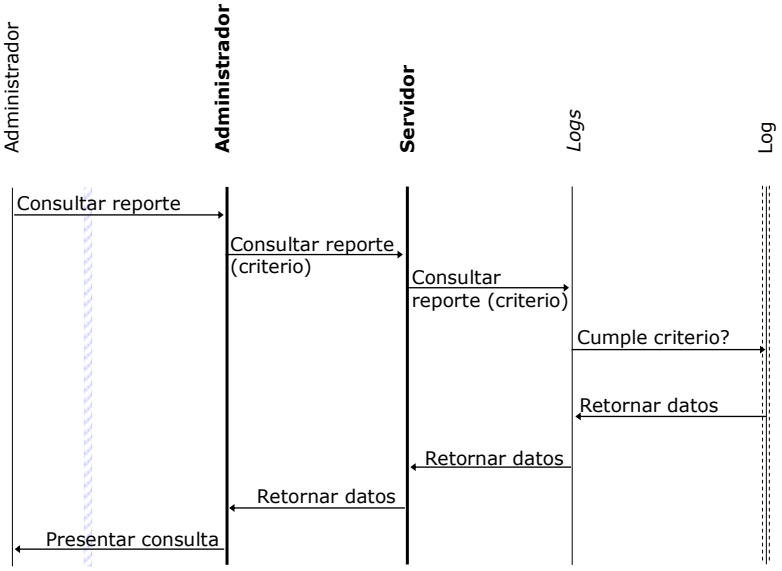
Si está en uso no se lo elimina; Datos: Nombre, descripción, imagen, habilitado, activo

Caso de uso:	11. Configuración del Sistema por el administrador
Escenario:	11.1. Configuración exitosa del sistema



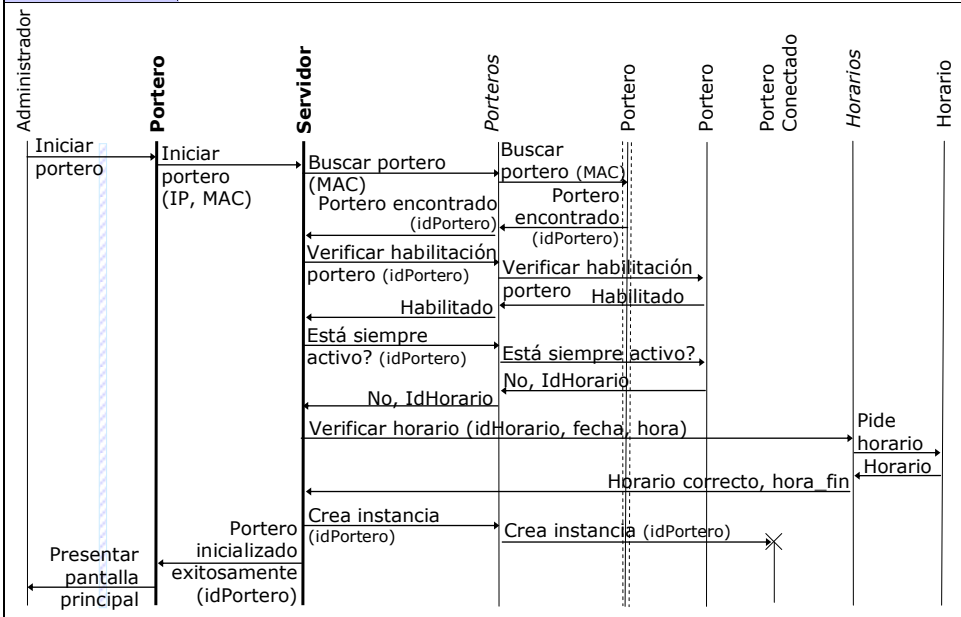
Datos: IP servidor de correo, tiempo de espera antes de cancelar una comunicación, usuario y contraseña del administrador, e-mail del administrador, espacio disco usuario con oficina, espacio disco usuario sin oficina, nombre de la carpeta que guardará los mensajes para los usuarios, tiempo de duración de un mensaje, usuario y contraseña para cuenta FTP

Caso de uso:	12. Administración del log
Escenario:	12.1. Consulta del Log



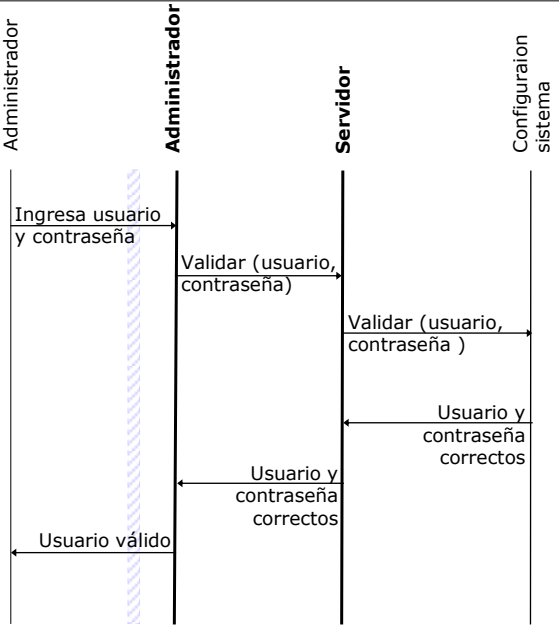
Criterio: tipo de transacción, nombre del portero, nombre del usuario interno, fecha inicio, fecha fin, orden de agrupación

Caso de uso:	13. Comunicación portero-servidor
Escenario:	13.1. Inicialización exitosa de un portero

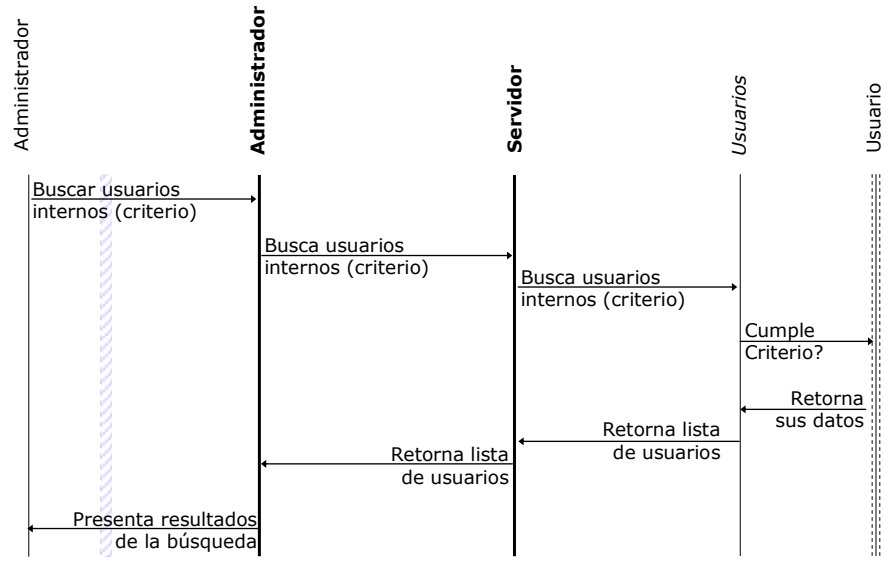


Si el portero tiene el campo siempreactivo en verdadero no verifica un horario.

Caso de uso:	14. Comunicación administrador - servidor
Escenario:	14.1. Solicitud de acceso exitosa del administrador

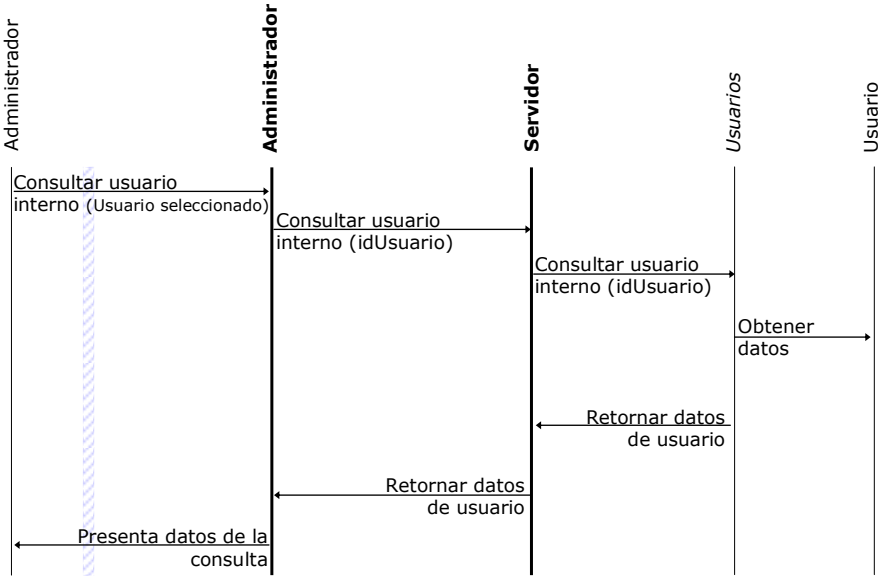


Caso de uso:	14. Comunicación Administrador-Servidor
Escenario:	14.2. Búsqueda de Usuarios internos



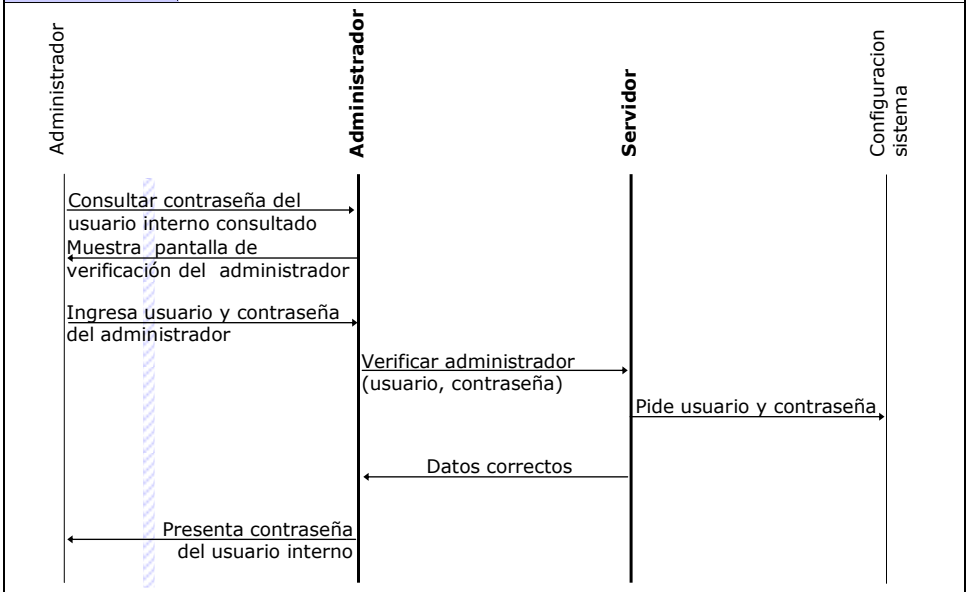
El mismo procedimiento se realiza para todos los elementos de sistema, cada uno con sus respectivos objetos.

Caso de uso:	14. Comunicación Administrador-Servidor
Escenario:	14.3. Consulta de un Usuario interno



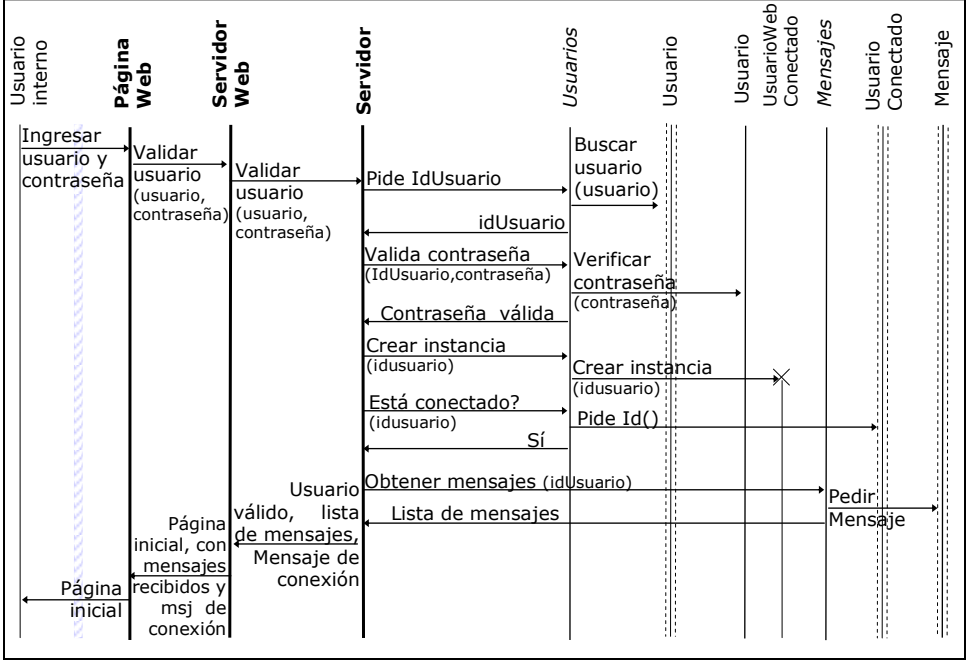
El mismo procedimiento se realiza para todos los elementos de sistema, cada uno con sus respectivos objetos.

Caso de uso:	14. Comunicación Administrador-Servidor
Escenario:	14.4. Consulta exitosa de la contraseña o clave de acceso de un usuario interno

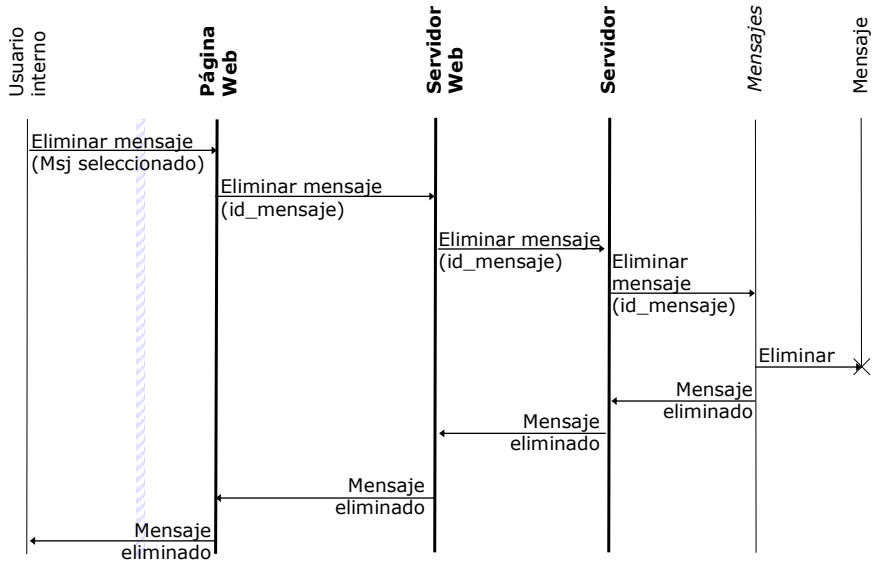


El mismo procedimiento se realiza para consultar la clave de acceso del usuario interno

Caso de uso:	15. Comunicación usuario interno – servidor vía Internet
Escenario:	15.1. Acceso exitoso del usuario interno al sistema vía Internet

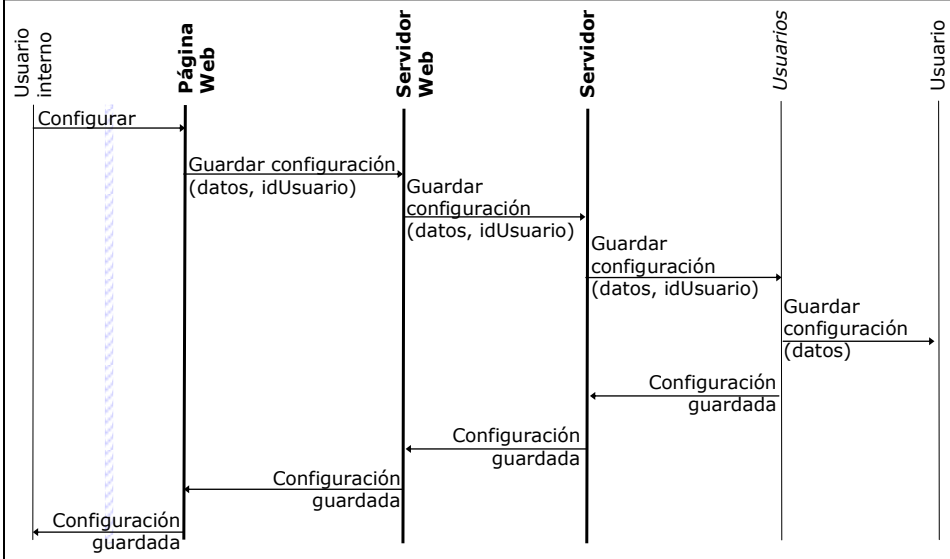


Caso de uso:	15. Comunicación usuario interno – servidor vía Internet
Escenario:	15.2. Eliminación de mensajes vía Internet



Los mensajes ya descargados (con campo leído igual a verdadero), pueden ser eliminados.

Caso de uso:	15. Comunicación usuario interno – servidor vía Internet
Escenario:	15.3. Configuración del sistema por el usuario interno vía Internet

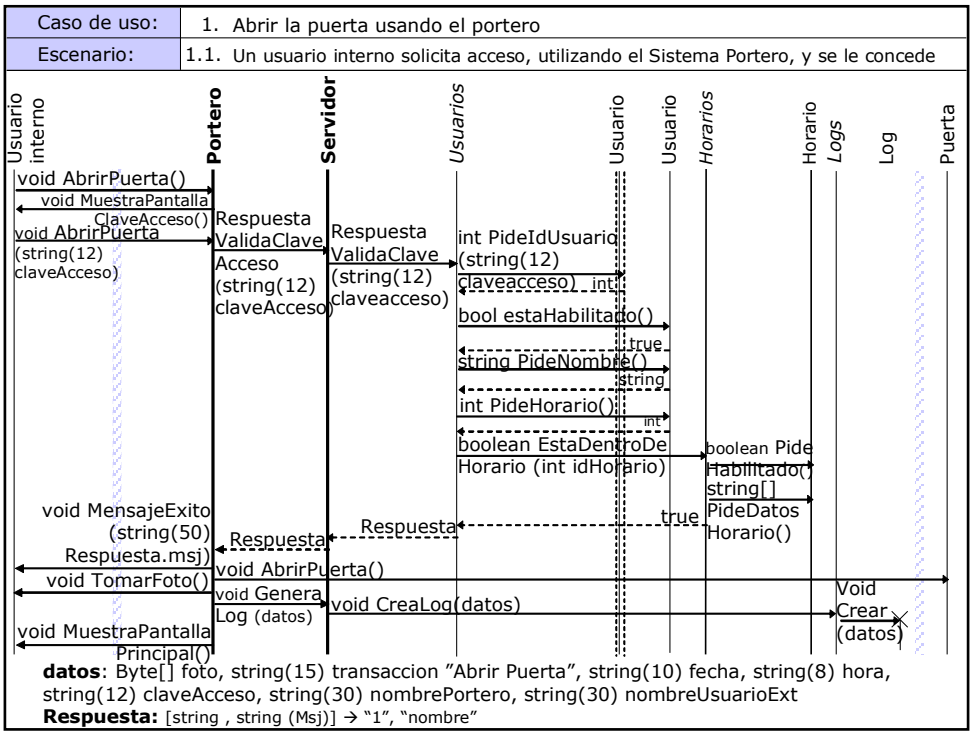


datos: cargo, teléfonos, e-mail, foto, contraseña, recibe foto en correo, observación, tipo de mensajes que dejarán los usuarios externos por medio del portero.

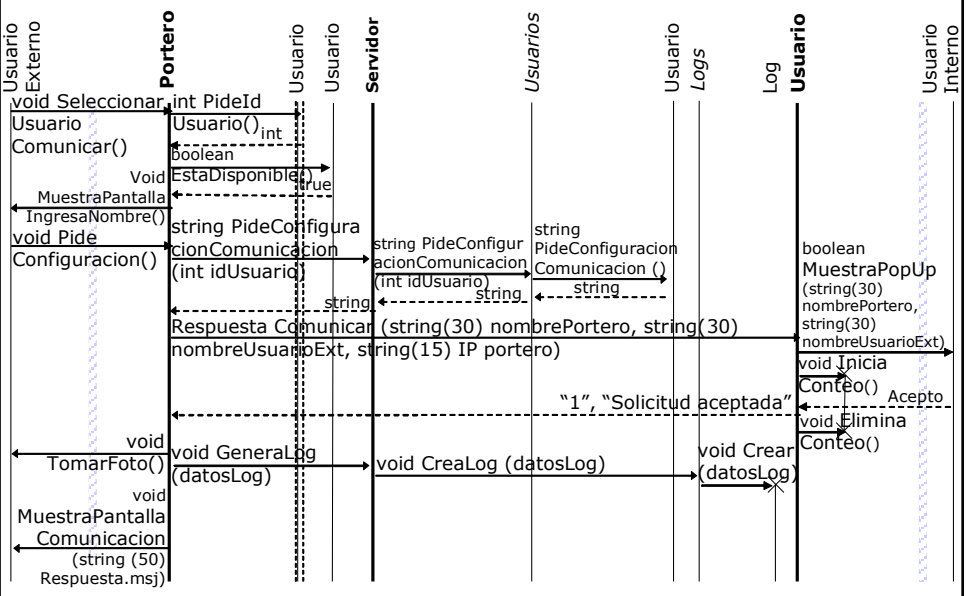
APÉNDICE C

DIAGRAMAS DE INTERACCIÓN DE OBJETOS DE DISEÑO

Los siguientes diagramas de interacción de objetos de diseño corresponden a los diagramas mostrados en el apéndice B.

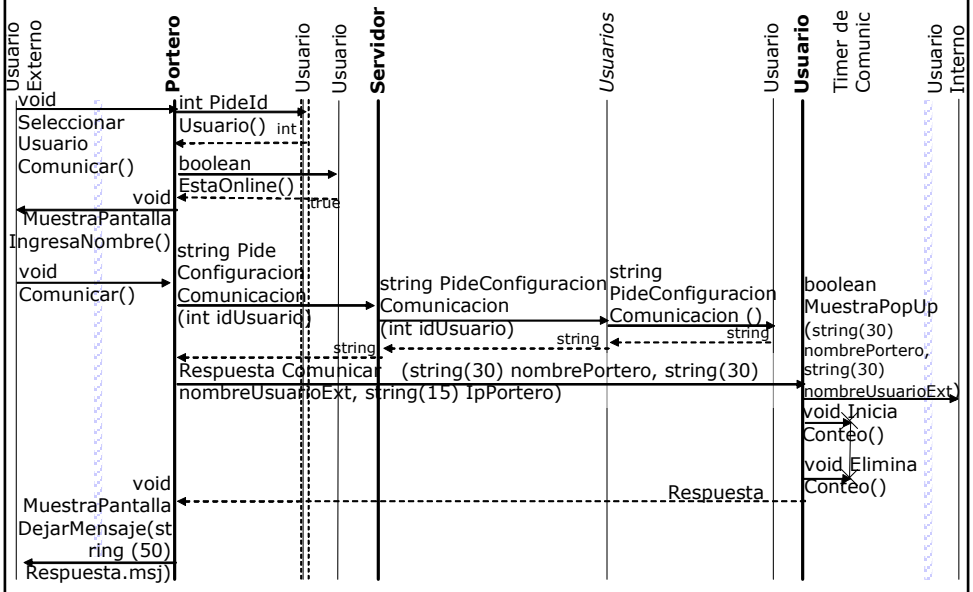


Caso de uso:	2. Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	2.1. Solicitud exitosa de comunicación con el usuario interno.



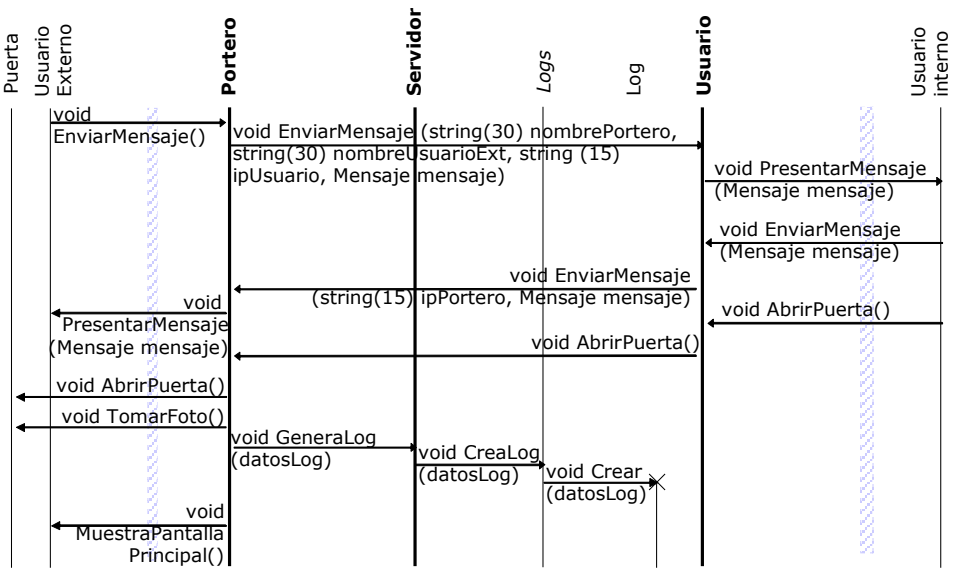
datosLog: Byte[] foto, string(15) transaccion "Comunicación", string(10) fecha, string(8) hora string(30) nombreUsuario, string(30) nombrePortero, string(30) nombreUsuarioExt

Caso de uso:	2. Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	2.2. Un usuario externo solicita comunicación con usuario interno y **NO** responde.



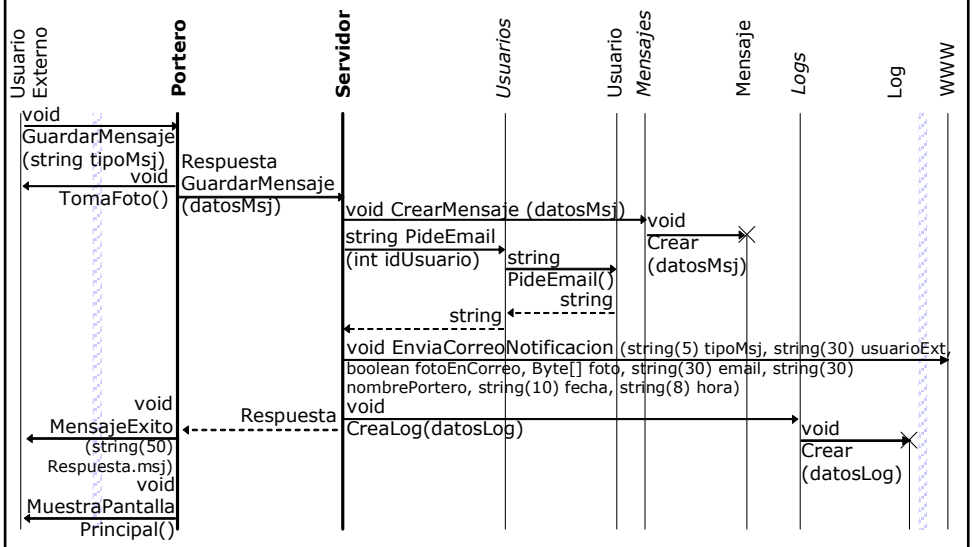
Razones de Solicitud no contestada: "tiempo de espera agotado" | "cambió a un estado no disponible"

Caso de uso:	2. Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	2.3. Se establece comunicación y se le da acceso al usuario externo



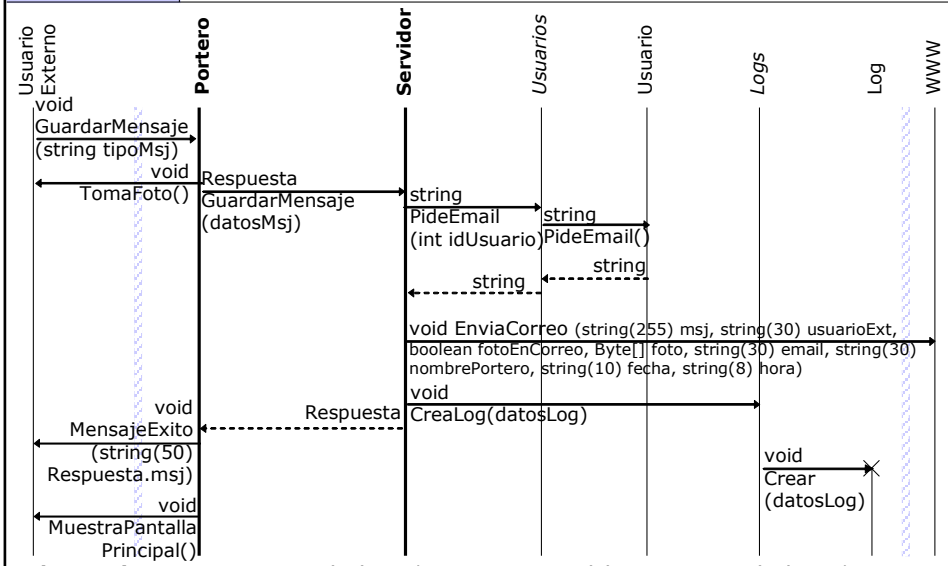
datosLog: Foto foto, string(15) transaccion "Abrir Puerta", string(10) fecha, string(8) hora, string(30) nombreUsuario, string(30) nombrePortero

Caso de uso:	2. Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	2.4. El usuario externo deja un mensaje de vídeo o de audio al usuario interno



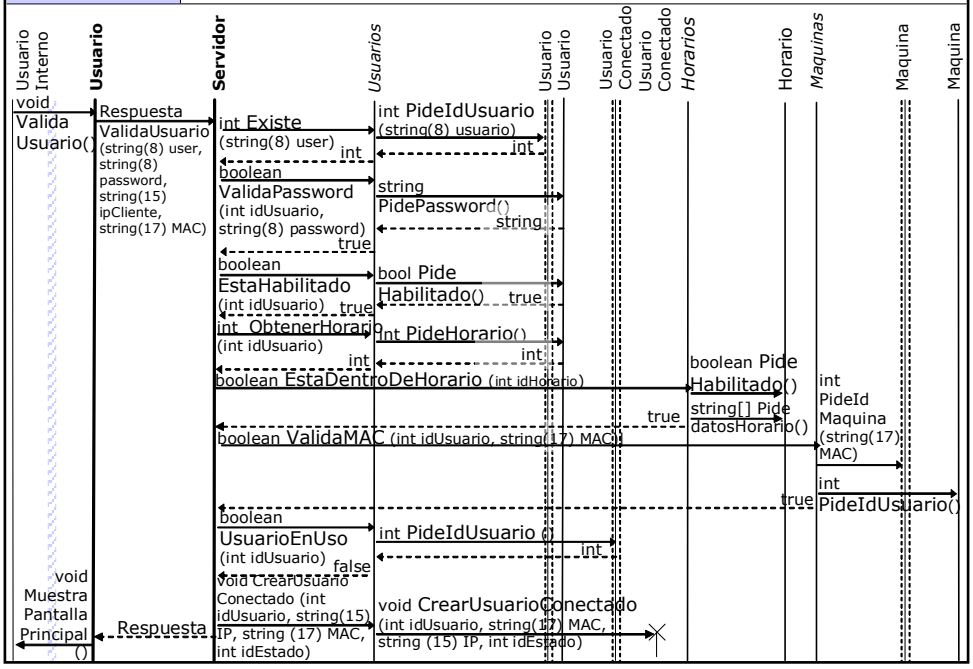
datosMsj: Mensaje msj, int idUsuario, string(5) tipoMsj, int idPortero, string(30) usuarioExt, Byte[] foto, string(10) fecha, string(8) hora.
datosLog: Byte[] foto, string(15) transaccion "Guardar Mensaje", string(10) fecha, string(8) hora string(30) nombreUsuario, string(30) nombrePortero, string(30) nombreUsuarioExt

Caso de uso:	2. Comunicación usuario externo – usuario interno por medio del sistema portero
Escenario:	2.5. El usuario externo deja un mensaje de texto al usuario interno

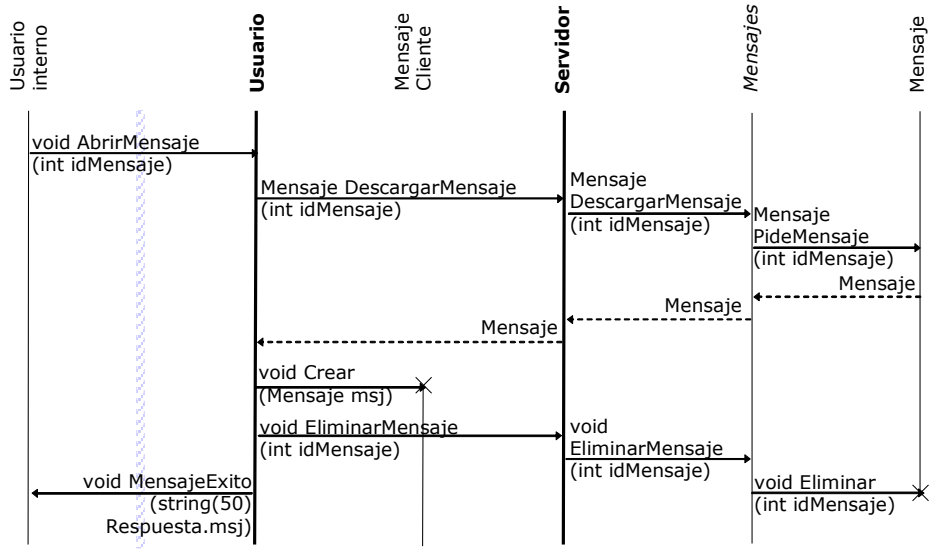


datosMsj: Mensaje msj, string(30) nombreUsuario, string(5) tipoMsj, string(30) nombrePortero, string(30) usuarioExt, Byte[] foto, string(10) fecha, string(8) hora.
datosLog: Byte[] foto, string(15) transaccion "Guardar Mensaje", string(10) fecha, string(8) hora string(30) nombreUsuario, string(30) nombrePortero, string(30) nombreUsuarioExt

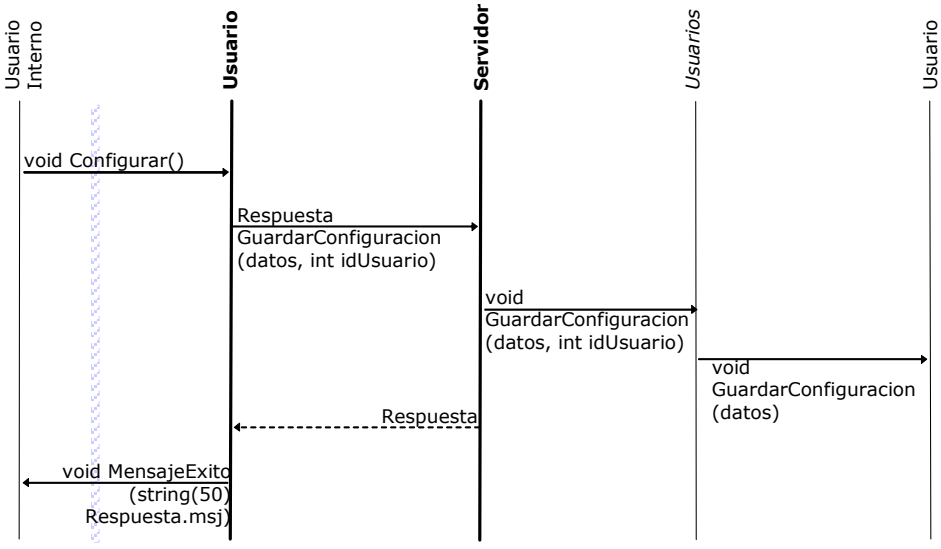
Caso de uso:	3. Comunicación usuario_interno - servidor
Escenario:	3.1. Acceso exitoso del usuario interno al sistema



Caso de uso:	3. Comunicación usuario interno - servidor
Escenario:	3.2. Revisión de los mensajes recibidos

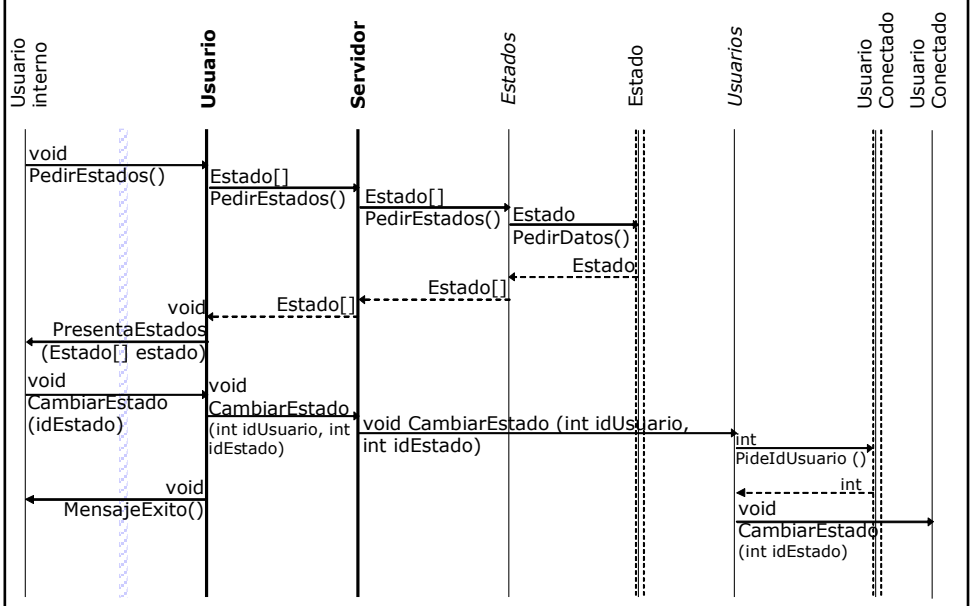


Caso de uso:	3. Comunicación usuario_interno - servidor
Escenario:	3.3. Configuración del sistema por el usuario interno

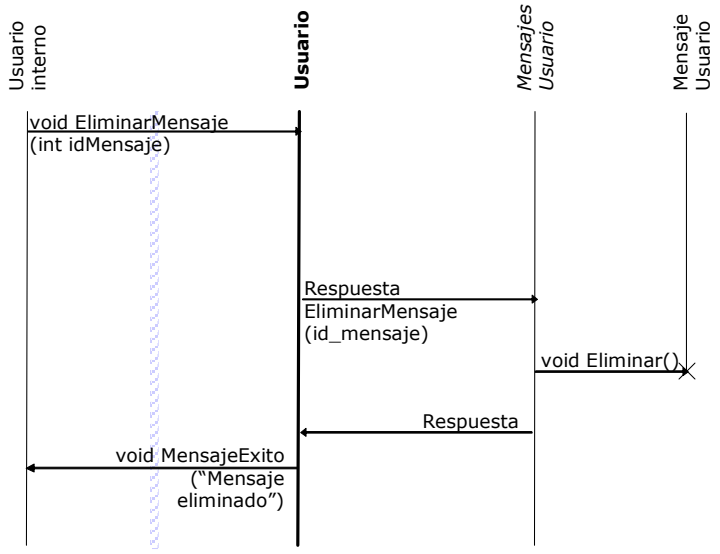


datos: string(50) cargo, string (50) telefono, string(50) email, byte[] foto, string(4) notificacion, boolean fotoEnCorreo, int tiempoInactividad, string(100) observacion, string(5) tipoComunicacion, string(15) tipoMsjAGuardar, string(8) password, boolean ventanavideo
Respuesta: [string , string (Msj)] → "1", "La Configuración del usuario ha sido guardada"

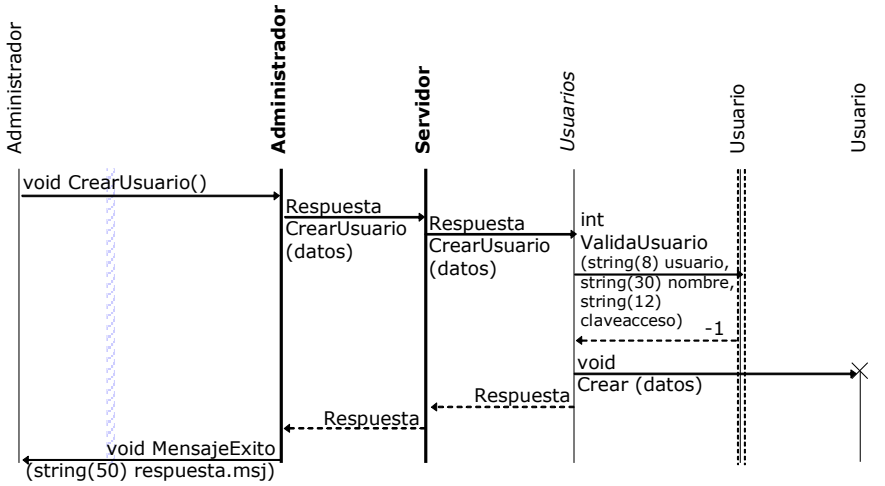
Caso de uso:	4. Comunicación del usuario interno con el sistema usuario.
Escenario:	4.1. Cambio manual del estado de conexión



Caso de uso:	4. Comunicación del usuario interno con el sistema usuario.
Escenario:	4.2. Eliminación de mensajes de la lista de almacenados



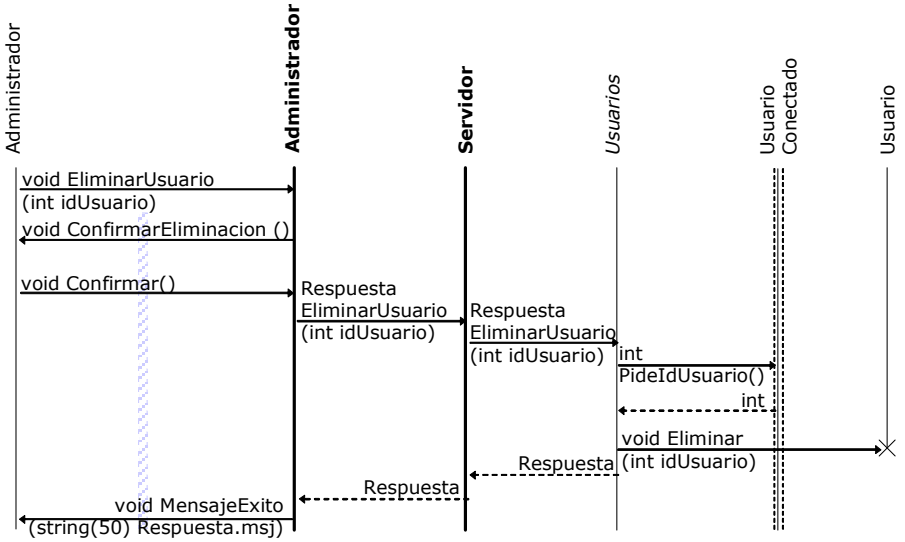
Caso de uso:	5. Administración de usuarios internos
Escenario:	5.1. Creación exitosa de un nuevo usuario interno



datos: string(30) nombre, string(50) cargo, string(50) telefono , string(50) email, boolean habilitado, string(8) usuario, string(8) contrasena, string(12) claveacceso, boolean siempreactivo, int idhorario, string[] categorias, boolean tieneoficina, string [] maquinas, string[] porteros

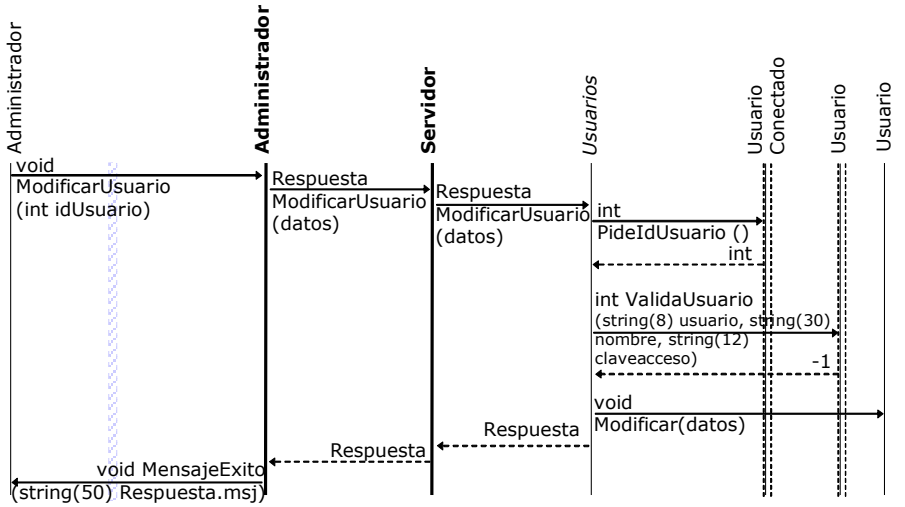
Respuesta: [string , string (msj)] → "1", "El usuario ha sido creado"

Caso de uso:	5. Administración de usuarios internos
Escenario:	5.2. Eliminación exitosa de un usuario interno



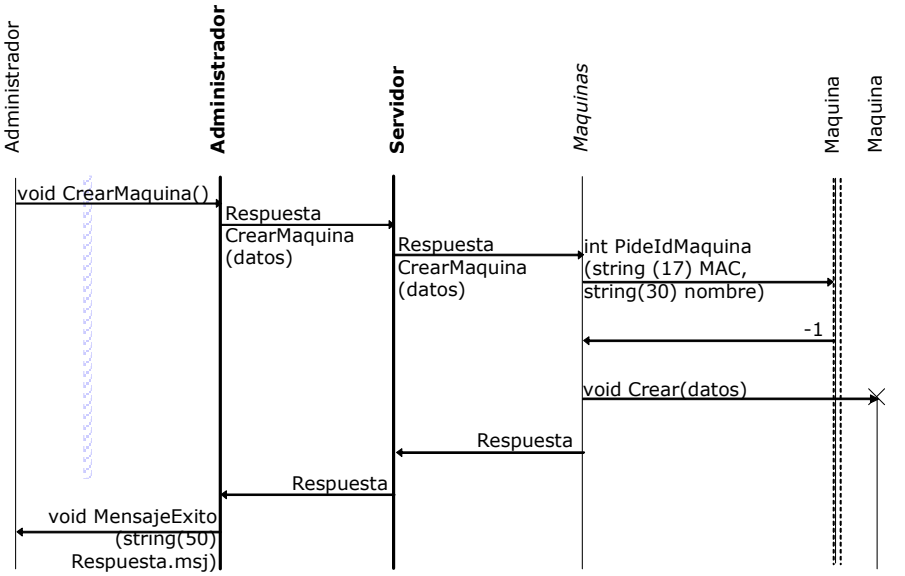
Respuesta: [string , string (msj)] → "1", "El usuario ha sido eliminado"
Si el usuario está conectado, no se lo elimina.

Caso de uso:	5. Administración de usuarios internos
Escenario:	5.3. Modificación exitosa de un usuario interno



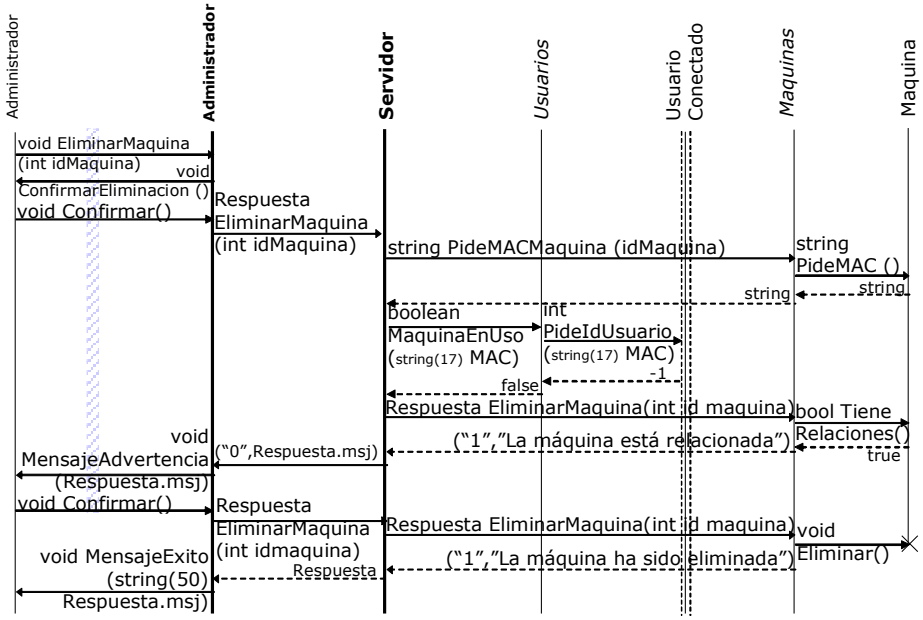
datos: int idUsuario, string(30) nombre, string(50) cargo, string(50) telefono , string(50) email, boolean habilitado, string(8) usuario, string(8) contrasena, string(12) claveacceso, boolean siempreactivo, int idhorario, string[] categorias, boolean tieneoficina, string [] maquinas, string[] porteros
Respuesta: [string , string (Msj)] → "1", "El usuario ha sido modificado"
Si el usuario está conectado, no puede ser modificado.

Caso de uso:	6. Administración de máquinas
Escenario:	6.1. Creación exitosa de una nueva máquina



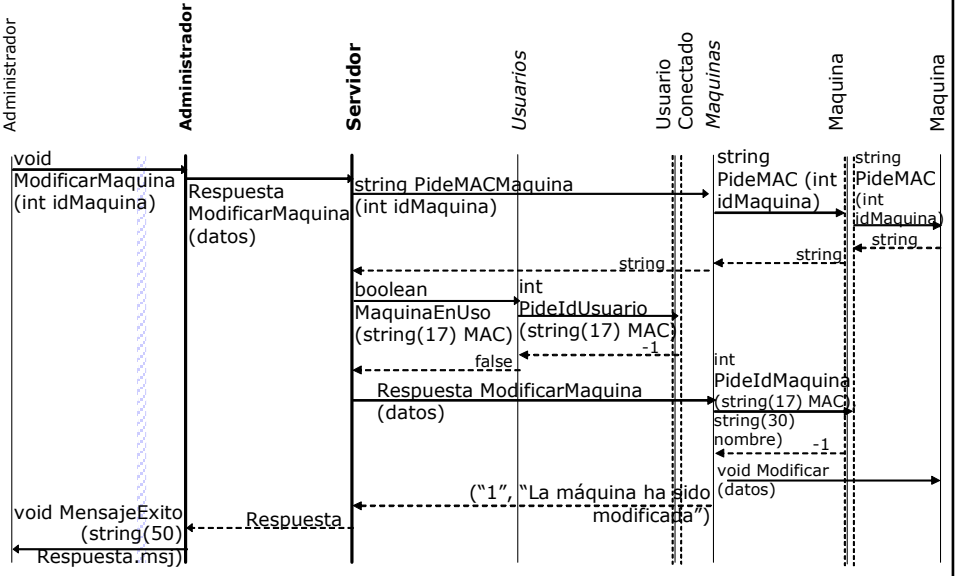
datos: string(30) nombre, string(17) MAC, boolean habilitado (por defecto false)
Respuesta: [string , string (Msj)] → "1", "La máquina ha sido creada"

Caso de uso:	6. Administración de máquinas
Escenario:	6.2. Eliminación de una máquina cuando esta asignada a uno o más usuarios



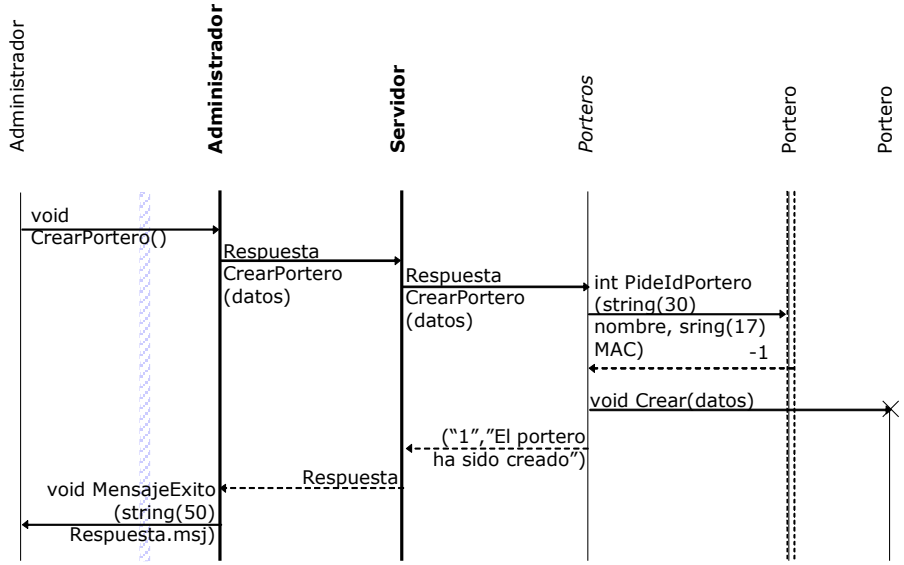
Si la máquina está en uso no puede ser eliminada, si no está relacionada se la elimina.

Caso de uso:	6. Administración de máquinas
Escenario:	6.3. Modificación exitosa de una máquina



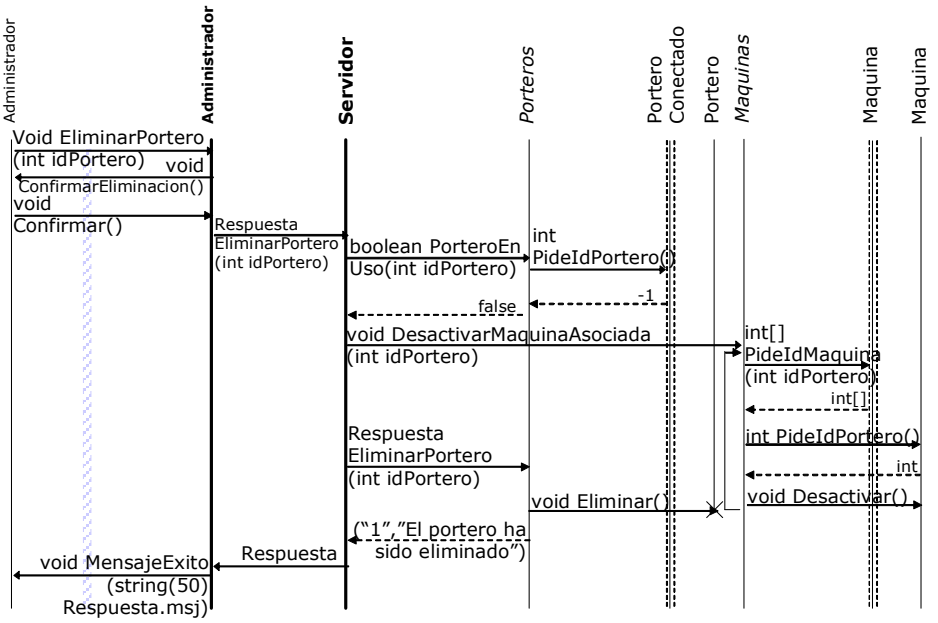
datos: string(30) nombre, string(17) MAC, boolean habilitado
Si la máquina está en uso no puede ser modificada; es modificada si la MAC y el nombre no existen

Caso de uso:	7. Administración de Porteros
Escenario:	7.1. Creación exitosa de un nuevo portero



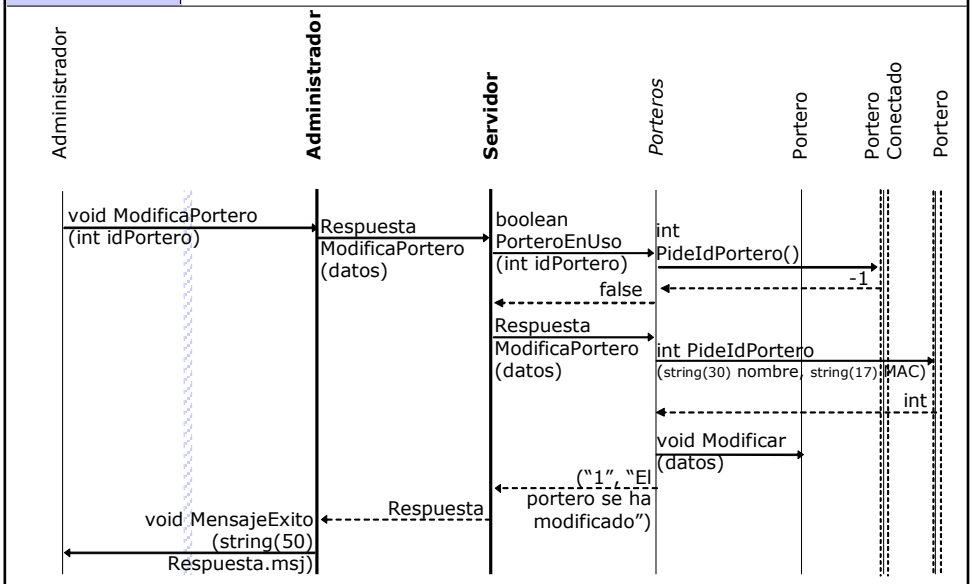
datos: string(30) nombre, string(100) descripcion, string(17) MAC, boolean habilitado, int idHorario, int[] idMaquinasAsociadas

Caso de uso:	7. Administración de Porteros
Escenario:	7.2. Eliminación exitosa de un portero



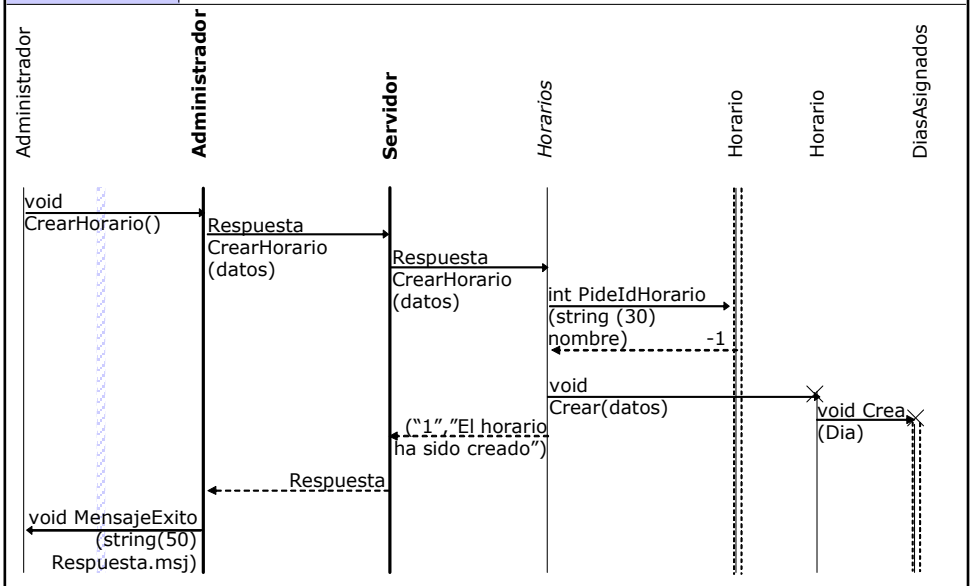
Si el portero está en uso no puede ser eliminado

Caso de uso:	7. Administración de Porteros
Escenario:	7.3. Modificación exitosa de un portero



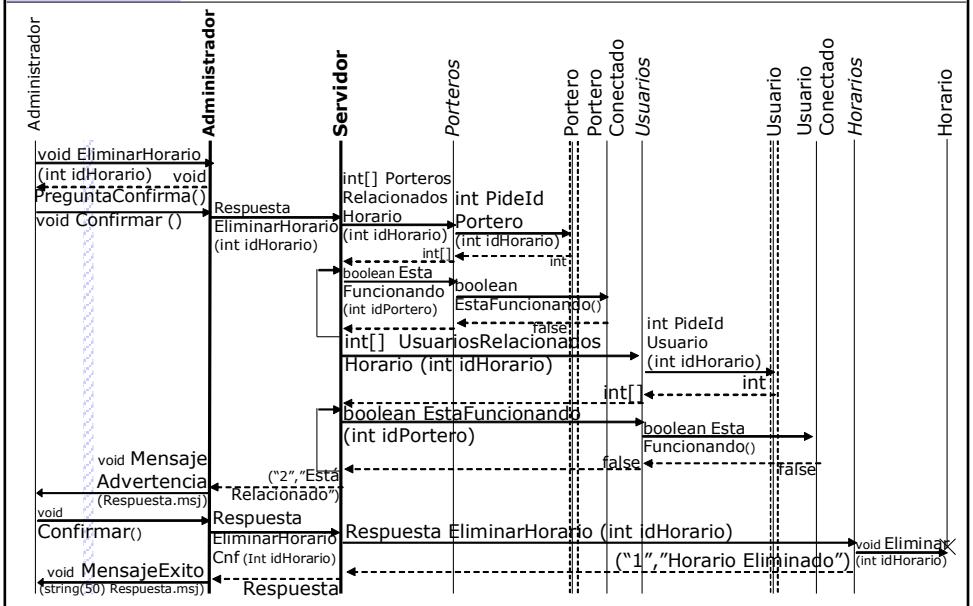
datos: string(30) nombre, string(100) descripcion, string(17) MAC, boolean estado, int idHorario, int[] idMaquinasAsociadas
Si el portero está en uso o la MAC o el nombre ya existen no puede ser modificado

Caso de uso:	8. Administración de Horarios
Escenario:	8.1. Creación exitosa de un nuevo horario



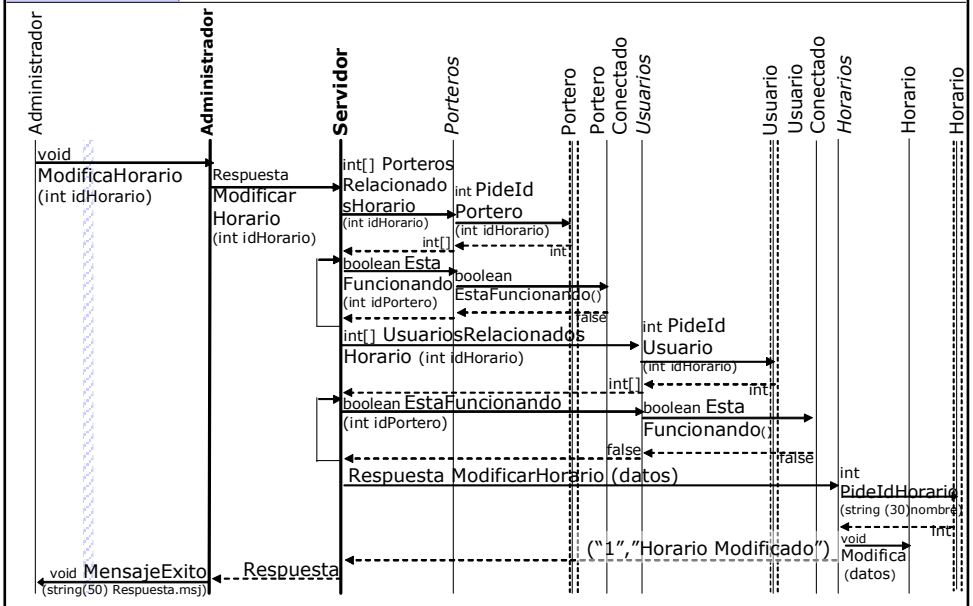
datos: string(30) nombre, string fechaIni, string fechaFin, boolean habilitado, Dia[] diasAsignados

Caso de uso:	8. Administración de Horarios
Escenario:	8.2. Eliminación de un horario asignado a uno o más usuarios internos o porteros



Si no está relacionado, lo elimina. Si está en uso no lo elimina.

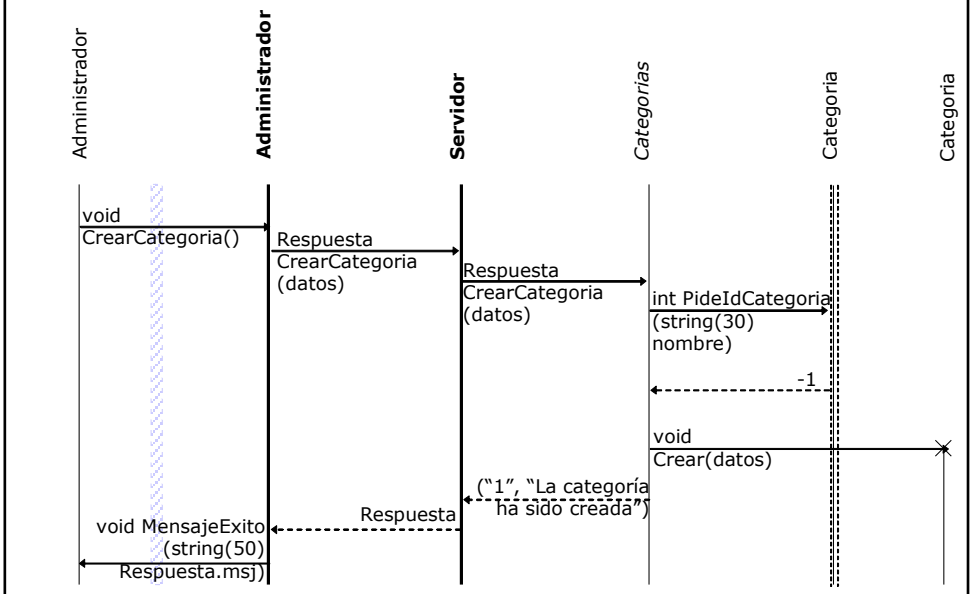
Caso de uso:	8. Administración de Horarios
Escenario:	8.3. Modificación exitosa de un horario



Si está en uso no lo modifica.

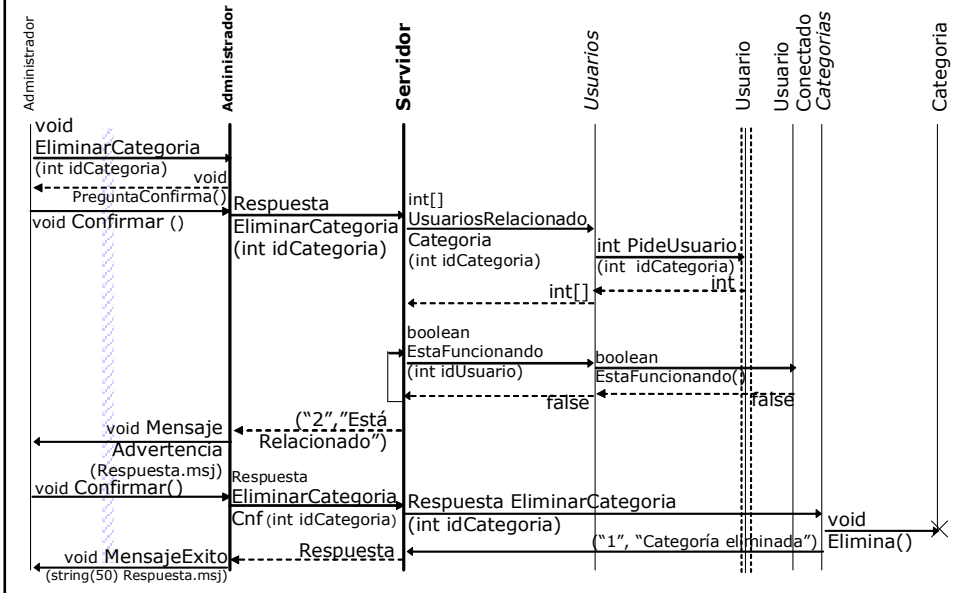
datos: string(30) nombre, string fechaIni, string fechaFin, boolean habilitado, Dia[] diasAsignados

Caso de uso:	9. Administración de Categorías
Escenario:	9.1. Creación exitosa de una nueva categoría



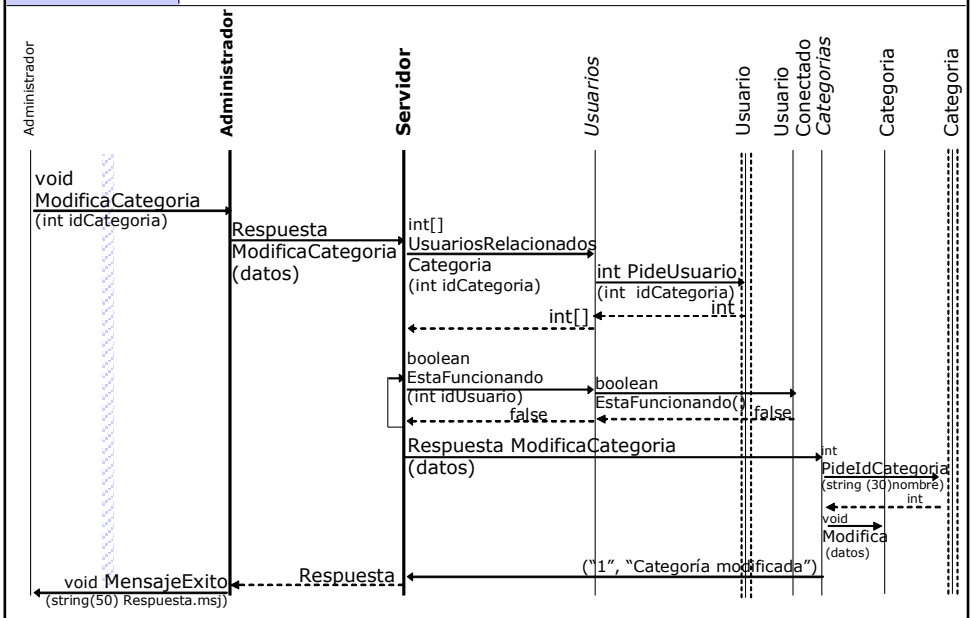
datos: string(30) nombre, string(100) descripcion

Caso de uso:	9. Administración de Categorías
Escenario:	9.2. Eliminación de una categoría que está asignada a uno o más usuarios internos



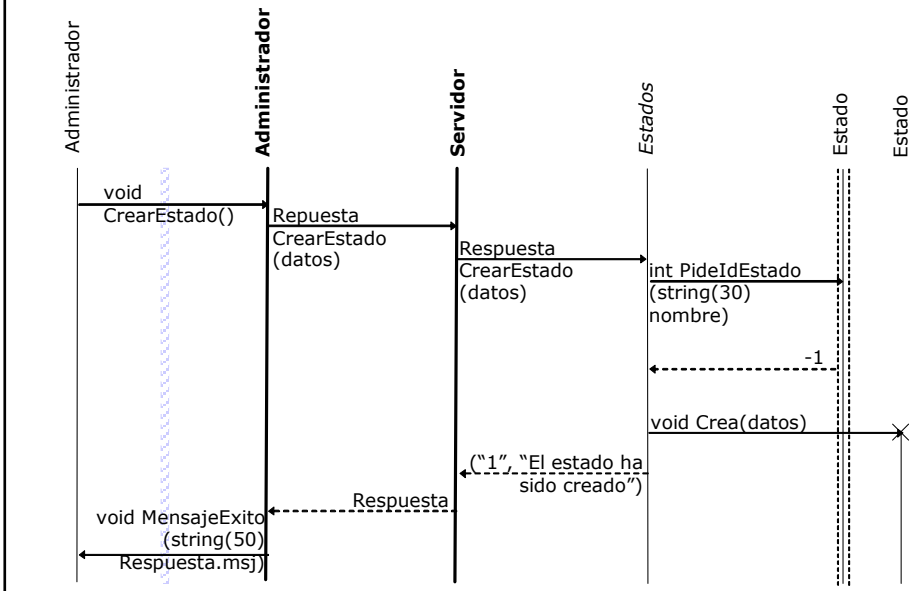
Si está en uso no la elimina

Caso de uso:	9. Administración de Categorías
Escenario:	9.3. Modificación exitosa de una categoría



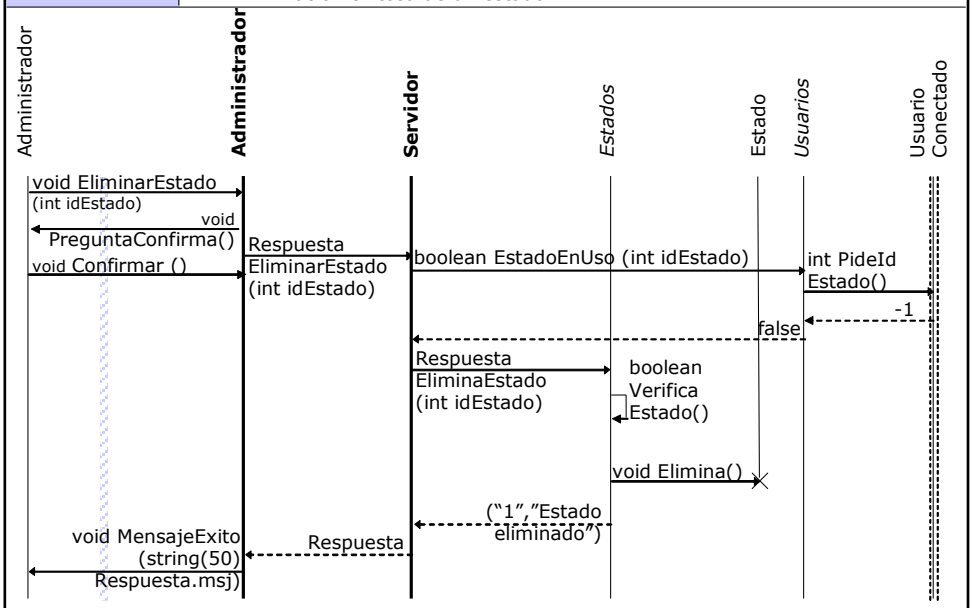
Si está en uso no lo elimina; **datos**: string(30) nombre, string(100) descripcion

Caso de uso:	10. Administración de Estados
Escenario:	10.1. Creación exitosa de un nuevo estado



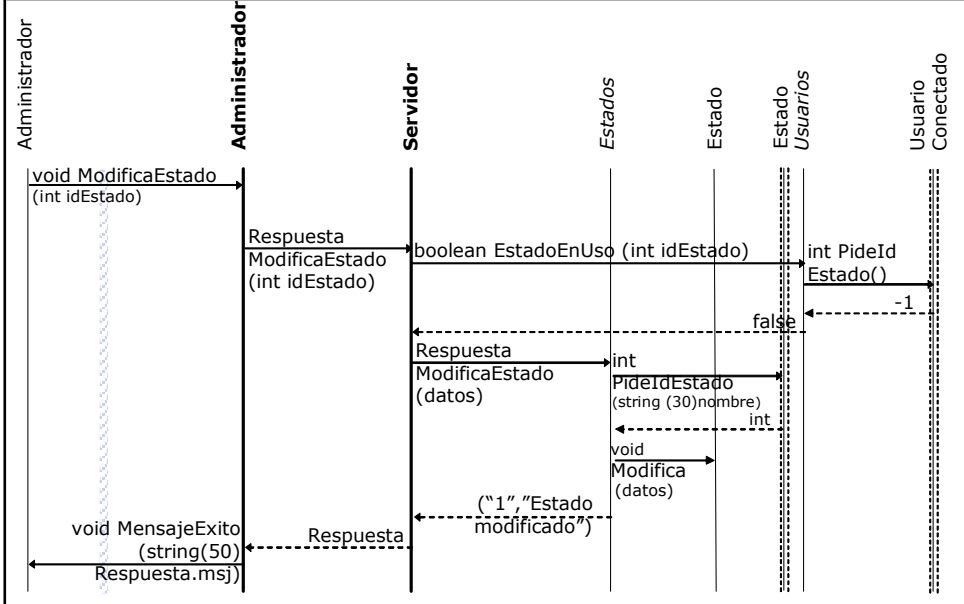
datos: string(30) nombre, string(100) descripcion, byte[] imagen, boolean habilitado, boolean activo (*si es verdadero, el usuario puede recibir solicitudes de comunicación al encontrarse en este estado*)

Caso de uso:	10. Administración de Estados
Escenario:	10.2. Eliminación exitosa de un estado



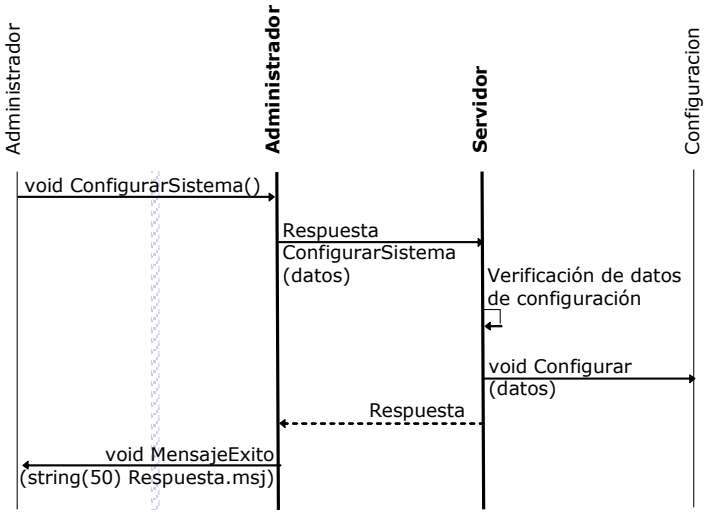
Si está en uso no lo elimina. Verificar Estado: Verifica que no se trate del estado "disponible", "ausente" o "desconectado" ya que estos son los estados básicos del sistema.

Caso de uso:	10. Administración de Estados
Escenario:	10.3. Modificación exitosa de un estado



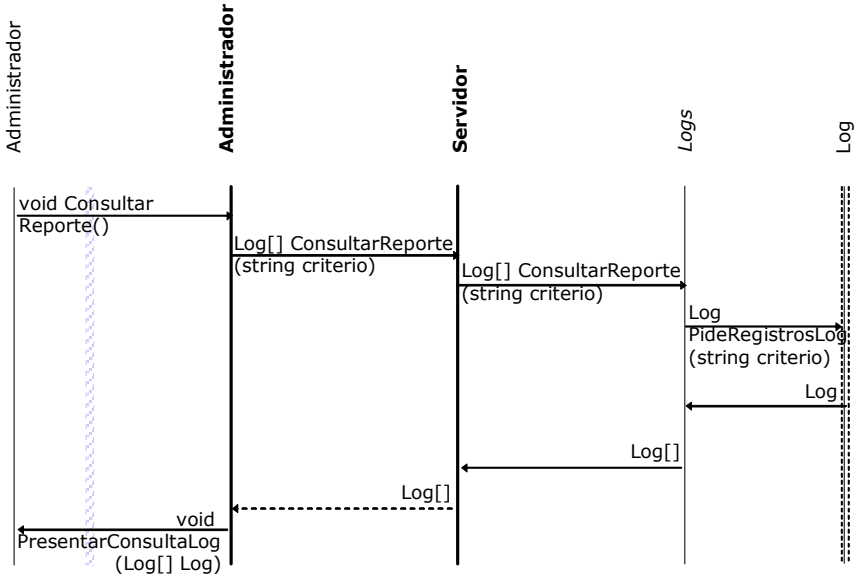
Si está en uso no lo elimina; **Datos:** string(30) nombre, string(100) descripcion, byte[] imagen, boolean habilitado, boolean activo

Caso de uso:	11. Configuración del Sistema por el administrador
Escenario:	11.1. Configuración exitosa del sistema



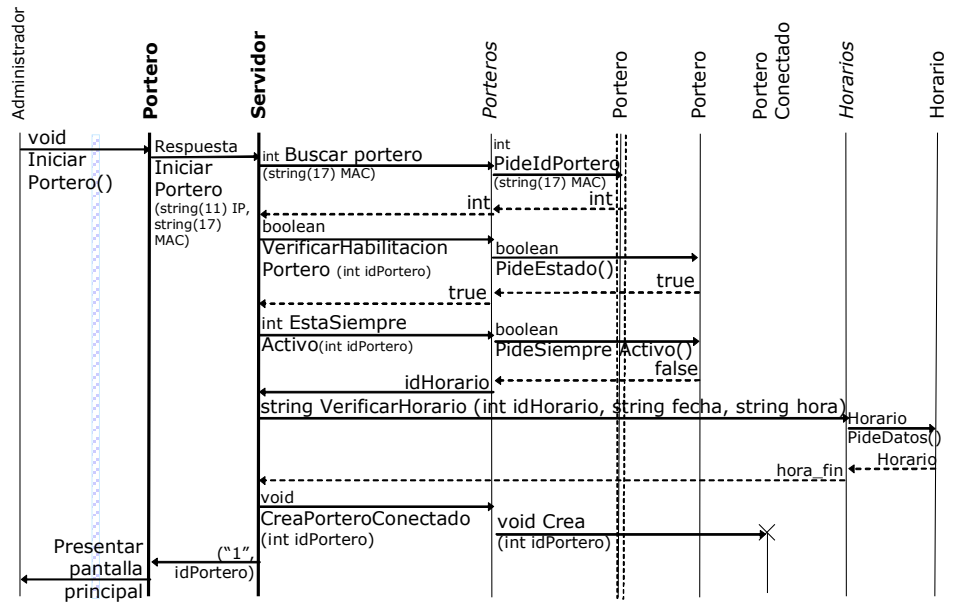
datos: string(15) ip_servidor_correo, int timercomunicacion, string(8) useradmin, string(8) pwdadmin, string(50) email_admin, string(20) carpeta, int espaciodiscousersconoficina, int espaciodiscouserssinoficina, int tiempomensaje, string(8) ftpuser , string(8) ftppwd
Respuesta: [string , string (Msj)] → "1", "La configuración del sistema ha sido guardada"

Caso de uso:	12. Administración del log
Escenario:	12.1. Consulta del Log



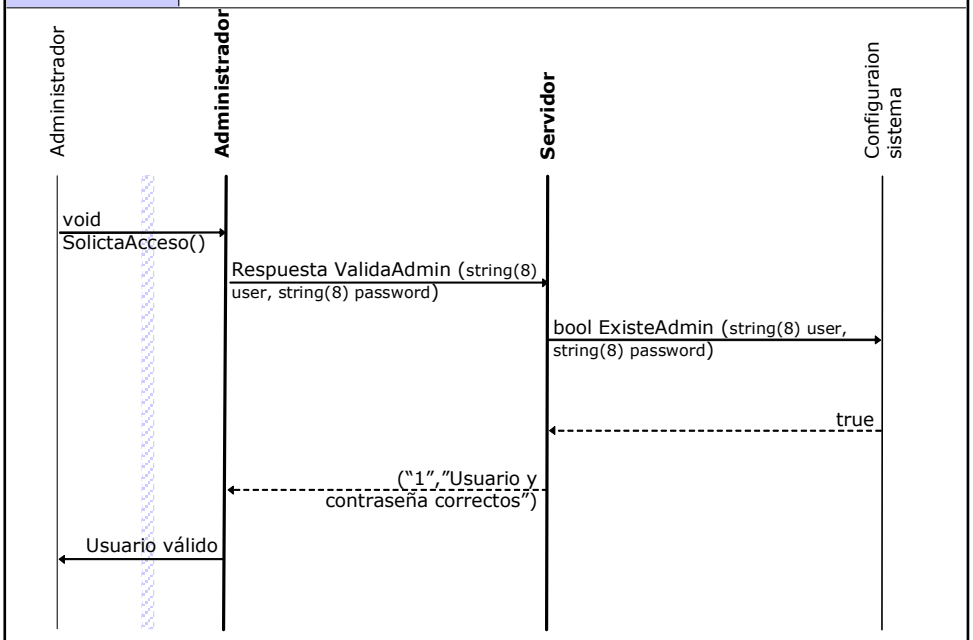
criterio: string(15) transaccion, string(30) nombreportero, string(30) nombreusuario, string fecha_inicio, string fecha_fin, string orden_agrupacion

Caso de uso:	13. Comunicación portero-servidor
Escenario:	13.1. Inicialización exitosa de un portero

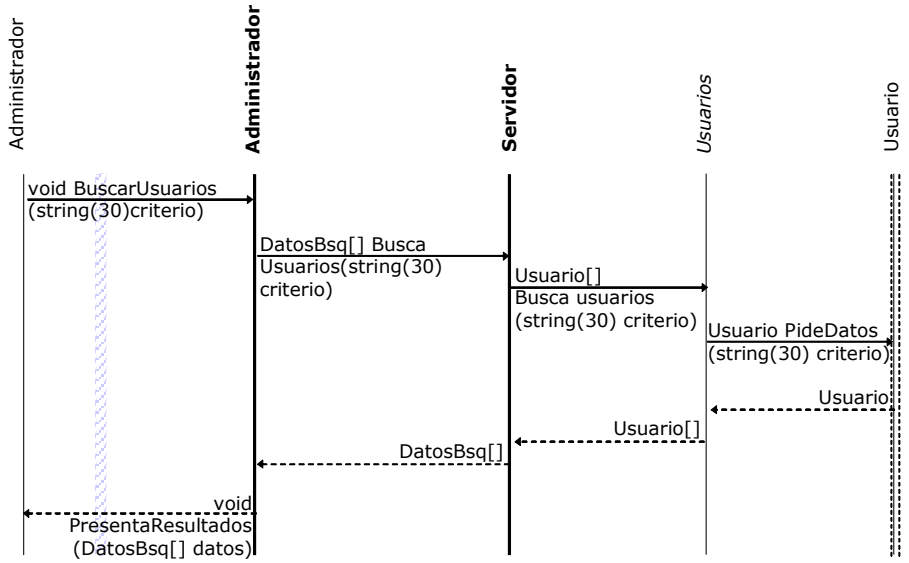


Si el portero tiene el campo siempreactivo en verdadero no verifica un horario.

Caso de uso:	14. Comunicación administrador - servidor
Escenario:	14.1. Solicitud de acceso del administrador exitosa

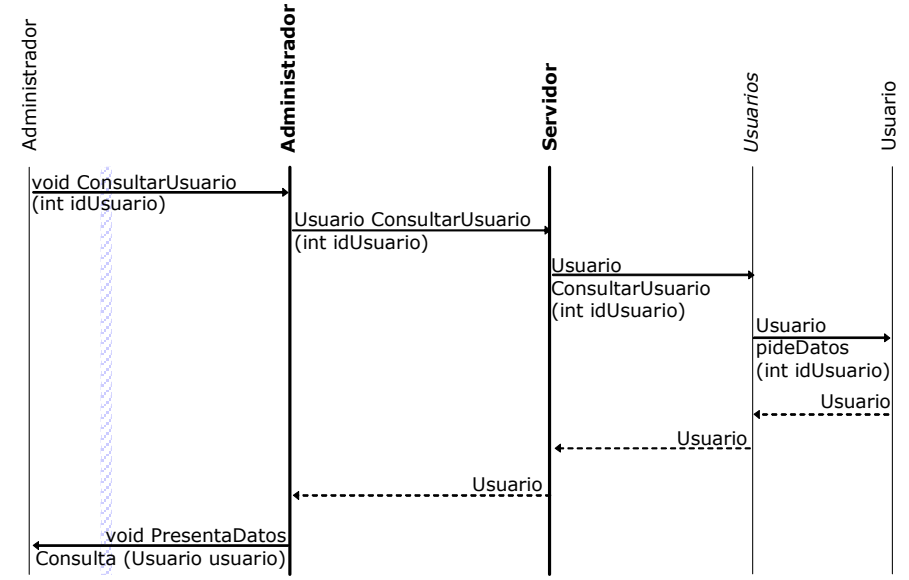


Caso de uso:	14. Comunicación Administrador-Servidor
Escenario:	14.2. Búsqueda de Usuarios internos



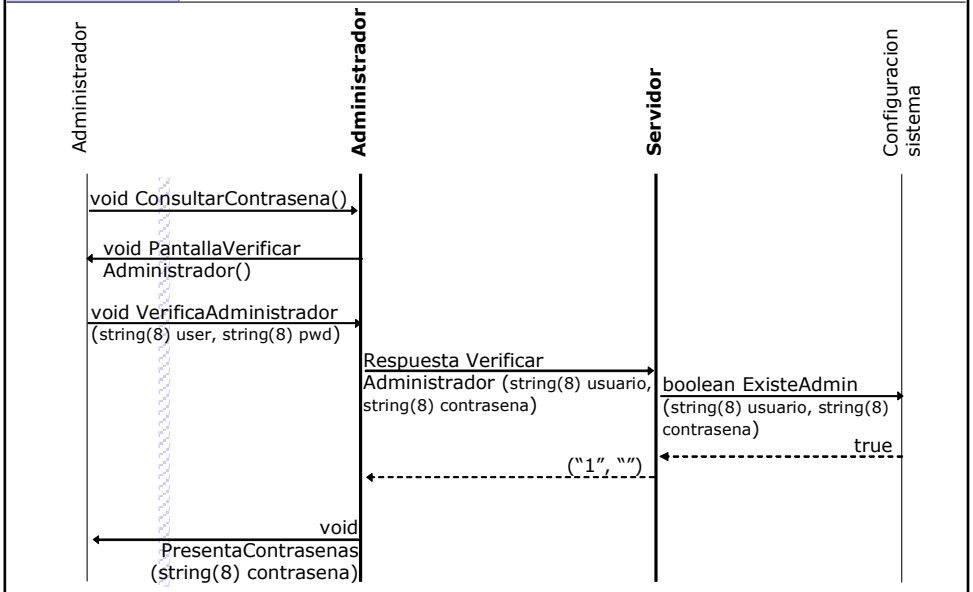
El mismo procedimiento se realiza para todos los elementos de sistema, cada uno con sus respectivos objetos.

Caso de uso:	14. Comunicación Administrador-Servidor
Escenario:	14.3. Consulta de un Usuario interno



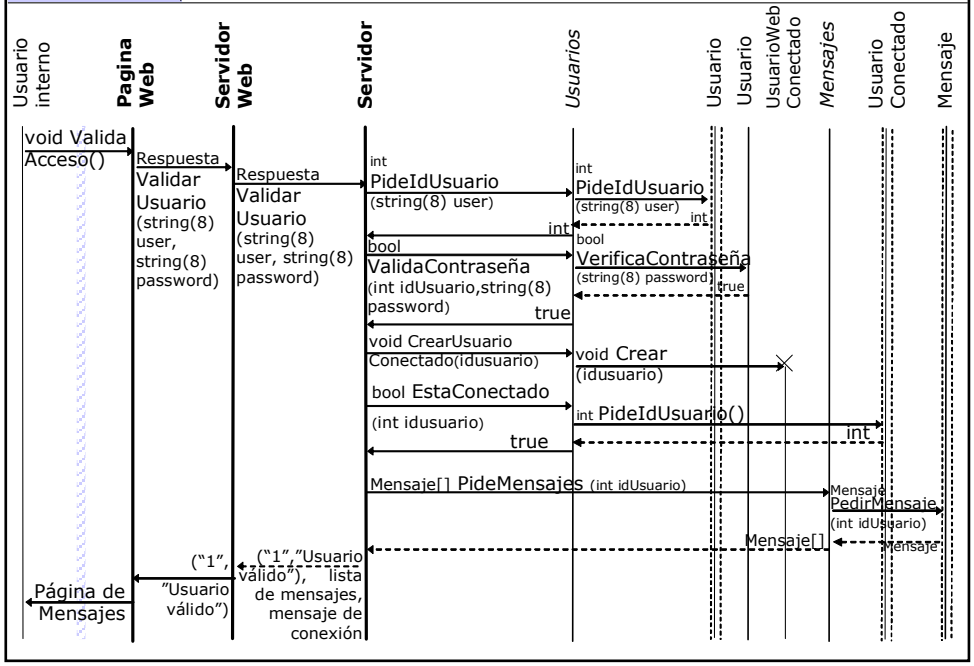
El mismo procedimiento se realiza para todos los elementos de sistema, cada uno con sus respectivos objetos.

Caso de uso:	14. Comunicación Administrador-Servidor
Escenario:	14.4. Consulta exitosa de la contraseña o clave de acceso de un usuario interno

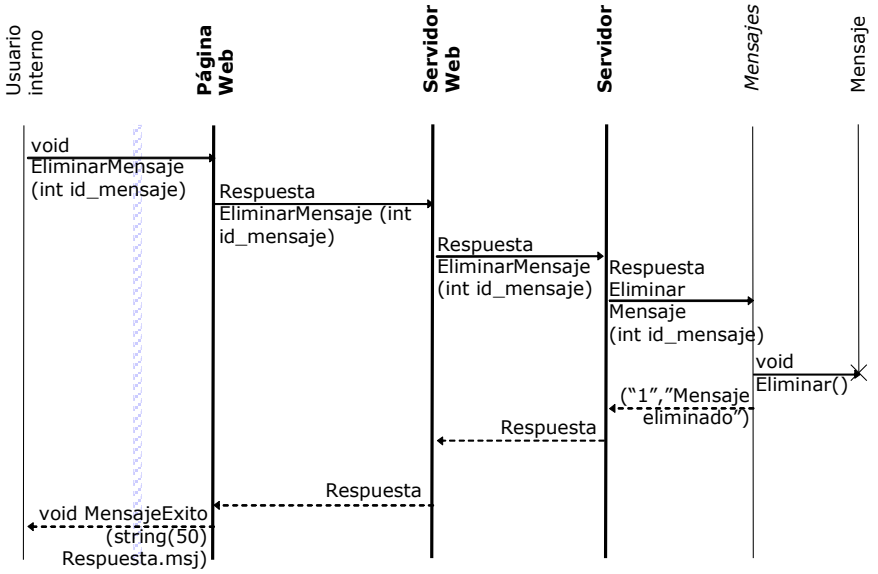


El mismo procedimiento se realiza para consultar la clave de acceso del usuario interno

Caso de uso:	15. Comunicación usuario interno – servidor vía Internet
Escenario:	15.1. Acceso exitoso del usuario interno al sistema vía Internet

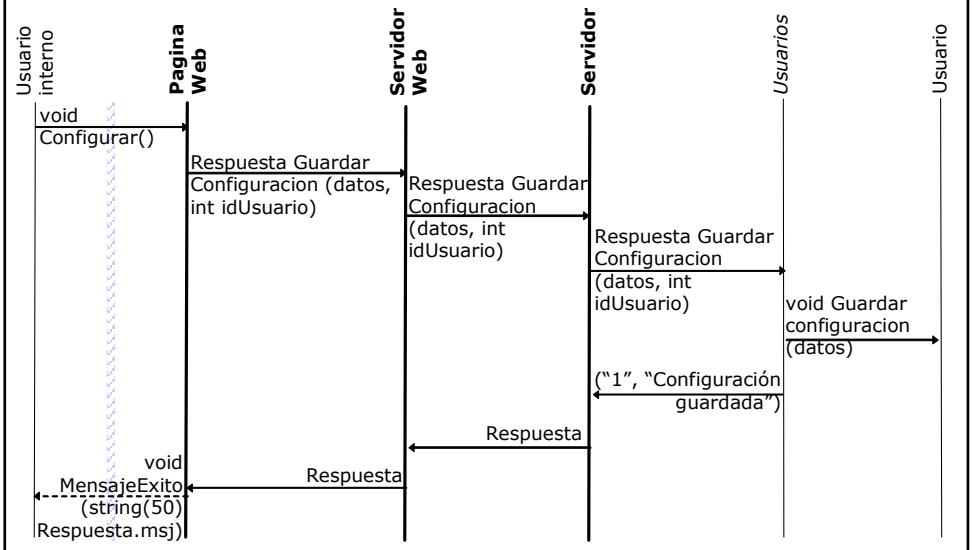


Caso de uso:	15. Comunicación usuario interno – servidor vía Internet
Escenario:	15.2. Eliminación de mensajes vía Internet



Los mensajes ya descargados (con campo leído igual a verdadero), pueden ser eliminados.

Caso de uso:	15. Comunicación usuario interno – servidor vía Internet
Escenario:	15.3. Configuración del sistema por el usuario interno vía Internet



datos: string(50) cargo, string (50) telefono, string(50) email, byte[] foto , string(8) password, boolean fotoEnCorreo, int string(100) observacion, string(15) tipoMsjAGuardar

APÉNDICE D

DICCIONARIO DE DATOS

A continuación se detallará el diccionario de datos correspondiente al modelo conceptual del sistema mostrado en la figura 3.2.

***usuario*.-** Representa a los usuarios que se encuentran registrados en el sistema, con el objetivo de que se pueda llevar un control de las distintas transacciones que realizan.

- *idusuario*.- Número entero consecutivo, que representa al identificador del usuario. No puede ser administrado manualmente, porque su valor es asignado automáticamente por la base de datos.
- *nombre*.- Nombre del usuario. Este atributo no es opcional y no se permite que contenga caracteres especiales. Es asignado por el administrador del sistema, y no puede ser modificado por el usuario. No se permite que dos o más usuarios registrados en el sistema compartan el mismo nombre.

- *usuario*.- Atributo con el que se identifica al usuario que desea acceder a las aplicaciones SCCIO Usuario o SCCIO Usuario Web. Puede tener entre 5 y 8 caracteres, y no se permite que contenga espacios ni caracteres especiales. Es manejado exclusivamente por el administrador del sistema. No se permite que dos o más usuarios registrados en el sistema tengan el mismo usuario.
- *contrasena*.- Atributo que permite autentificar al usuario que intenta conectarse al sistema usando SCCIO Usuario o SCCIO Usuario Web. El usuario puede modificar este dato.
- *claveacceso*.- Palabra secreta con la que se concede acceso al usuario cuando desea abrir la puerta utilizando el portero. Solo puede ser asignada por el administrador del sistema.
- *habilitado*.- Atributo lógico que toma valores de "0" o "1". Permite identificar a los usuarios que pueden utilizar el sistema. Uno de los requisitos para que estos puedan ingresar a las aplicaciones SCCIO Usuario y SCCIO Usuario Web, y para que puedan tener acceso a las oficinas desde el SCCIO Portero, es que este atributo tenga un valor de "1". Este dato es modificado exclusivamente por el administrador del sistema.
- *siempreactivo*.- Atributo lógico que toma valores de "0" o "1". Este campo indica si el acceso del usuario a las aplicaciones del sistema y al área de oficinas se verá restringido de acuerdo a un horario específico o no. Si este atributo es "1", el usuario tendrá acceso en todo

momento; de lo contrario, se le deberá asignar un horario para controlar su ingreso. Este campo es manejado exclusivamente por el administrador del sistema.

- *notificacion*.- Si el valor de este campo es "cpop", cada vez que un usuario externo intente comunicarse con un usuario interno por medio de la aplicación SCCIO Portero, al usuario interno le aparecerá un mensaje emergente acompañado de un sonido, indicándole que alguien se desea comunicar con él. En cambio, si el valor de este campo es "spop", se le presentará el mensaje emergente sin reproducir ningún sonido. Este atributo es administrado solamente por el usuario, en su configuración de aplicación, y su valor por defecto es "spop".
- *fotoencorreo*.- Atributo lógico que toma valores de "0" o "1". Si el usuario interno ha especificado un valor de "1" para este campo, cada mensaje de texto y cada notificación que se le envíe desde el SCCIO Portero contendrá una foto del remitente como dato adjunto. Este atributo es administrado solamente por el usuario, y su valor por defecto es "0".
- *tiempoinactividad*.- Es un valor entero, que representa el tiempo en minutos que el usuario interno puede permanecer inactivo, antes de que el sistema automáticamente cambie su estado de conexión al de ID=2 (ausente). Este atributo es administrado solamente por el

usuario; cada vez que se crea un usuario nuevo, este atributo toma el valor predeterminado de 15.

- *observacion*.- Es el mensaje que el usuario desea que se muestre en el portero, junto con sus demás datos. No se le permite contener caracteres especiales y su longitud máxima es de 100. Este atributo es administrado solamente por el usuario.
- *tieneoficina*.- Atributo lógico que toma valores de "0" o "1". Este campo permite determinar si el usuario posee una oficina o un lugar físico de trabajo dentro del área controlada por el portero. Si el usuario no tiene oficina, solamente se le permitirá revisar sus mensajes desde SCCIO Usuario Web. Este atributo es administrado exclusivamente por el administrador del sistema.
- *telefono*.- Representa los números telefónicos del usuario. Tiene una máxima longitud de 50 caracteres. Es administrado exclusivamente por el usuario.
- *email*.- Atributo que indica al sistema la dirección de correo electrónico del usuario; se lo utiliza cada vez que se desea enviar un mensaje o notificación al usuario interno desde el SCCIO Portero. Es un atributo requerido y su longitud máxima es de 50 caracteres.
- *cargo*.- Es el cargo o título que el usuario posee. Tiene una longitud máxima de 50 caracteres.
- *tipocomunicacion*.- Es un atributo que representa el tipo de comunicación que el usuario está dispuesto a aceptar cuando un

usuario externo solicite establecer una comunicación instantánea con él desde el SCCIO Portero. Los valores que puede tener son: "audio", "vídeo" o "texto". Este atributo es administrado exclusivamente por el usuario y su valor por defecto es "texto".

- *tipomsjaguardar*.- Es un atributo multivaluado que representa el tipo de mensaje que un usuario externo puede dejar al usuario interno cuando éste no se encuentre conectado o rechace una solicitud de comunicación. Puede tomar combinaciones de los siguientes valores: "audio", "vídeo" y "texto". Este atributo es administrado exclusivamente por el usuario y su valor por defecto es "texto".
- *ventanavideo*.- Atributo lógico que toma valores de "0" o "1". Permite determinar si el usuario interno desea visualizar o no, al usuario externo, cuando la comunicación que se establezca sea de "texto" o "audio".

mensaje.- Representa a los mensajes de audio o vídeo que los usuarios externos han dejado a los usuarios internos utilizando el sistema portero.

- *idmensaje*.- Número entero consecutivo, que representa al identificador del mensaje. No puede ser administrado manualmente, porque su valor es asignado automáticamente por la base de datos.
- *formatomensaje*.- Indica al sistema el tipo de mensaje que representa; puede tomar los siguientes valores: "audio", "vídeo".

- *nombreportero*.- Indica el nombre del portero desde el que se realizó el envío del mensaje.
- *usuarioexterno*.- Especifica el nombre ingresado por la persona que envió el mensaje. Es un atributo requerido.
- *fecha*.- Representa la fecha en la que el mensaje fue enviado al usuario.
- *hora*.- Representa la hora en la que el mensaje fue enviado al usuario.
- *leido*.- Atributo lógico que puede tomar valores de "0" o "1". Es "1" solamente si el usuario ha descargado el mensaje utilizando SCCIO Usuario Web.
- *nombre*.- Especifica el nombre del archivo que contiene el mensaje enviado al usuario. Tiene una longitud máxima de 80 caracteres.

***maquina*.-** Representa a las computadoras registradas en el sistema.

- *idmaquina*.- Número entero consecutivo, que representa al identificador de la máquina. No puede ser administrado manualmente, porque su valor es asignado automáticamente por la base de datos.
- *nombre*.- Nombre con la que se desea representar a la máquina. Es un campo requerido y único. Tiene una longitud máxima de 15 caracteres.
- *mac*.- Campo que representa la dirección MAC de la computadora. El objetivo de este atributo es poder autentificar a la máquina, cuando se

intenta acceder desde ella al sistema. Este campo es requerido y único.

- *habilitado*.- Atributo lógico que toma valores de "0" o "1". Este campo debe tener el valor de "1" para que una máquina pueda ser utilizada para ingresar al sistema.

***horario*.-** Representa los distintos horarios que pueden ser utilizados para restringir el acceso de los usuarios y porteros al sistema.

- *idhorario*.- Número entero consecutivo, que representa al identificador del horario. No puede ser administrado manualmente, porque su valor es asignado automáticamente por la base de datos.
- *nombre*.- Nombre que permite identificar a un horario. Es un campo requerido y único.
- *habilitado*.- Atributo lógico que toma valores de "0" o "1". En caso de que sea "0", los usuarios y/o porteros que tengan asignado ese horario, no podrán acceder al sistema.
- *descripcion*.- Atributo que describe al horario. Tiene una longitud máxima de 100 caracteres.
- *fechainicio*.- Especifica desde qué fecha rige el horario. Es un atributo requerido y debe ser una fecha anterior al atributo *fechafin*.
- *fechafin*.- Especifica hasta qué fecha rige el horario. Es un atributo requerido y debe ser una fecha posterior al atributo *fechainicio*.

portero.- Representa a las computadoras que ejecutan el SCCIO Portero y se encuentran registradas en el sistema.

- *idportero*.- Número entero consecutivo, que representa al identificador del portero. No puede ser administrado manualmente, porque su valor es asignado automáticamente por la base de datos.
- *nombre*.- Nombre que permite identificar al portero. Es un campo requerido y único. Tiene una longitud máxima de 15 caracteres.
- *habilitado*.- Atributo lógico que toma valores de "0" o "1". Sólo si el valor del atributo es "1", el portero tiene autorización para ejecutarse.
- *siempreactivo*.- Atributo lógico que toma valores de "0" o "1". Este atributo tiene la función de indicar si al portero se le restringe el acceso al sistema de acuerdo a un horario específico. Si este atributo es "1", el portero podrá ejecutarse en cualquier momento.
- *descripcion*.- Atributo que describe al portero. Tiene una longitud máxima de 100 caracteres.
- *mac*.- Campo que representa la dirección MAC de la computadora. El objetivo de este atributo es poder autentificar al portero. Este campo es requerido y único.

***dias*.-** Representa los días de la semana que componen a los horarios.

- *iddias.*- Número entero consecutivo, que representa al identificador del día. No puede ser administrado manualmente, porque su valor es asignado automáticamente por la base de datos.

- *dia*.- Representa un día de la semana. Los valores que puede tomar son de "Lunes" a "Domingo".
- *horainicio*.- Especifica desde qué hora en ese día se concede acceso al sistema, este atributo debe ser menor a la hora de fin.
- *horafin*.- Especifica hasta qué hora en ese día se concede acceso al sistema, este atributo debe ser mayor a la hora de inicio.

***categoria*.-** Representa las clasificaciones en las que están organizados los usuarios registrados.

- *idcategoria*.- Número entero consecutivo, que representa al identificador de la categoría. No puede ser administrado manualmente, porque su valor es asignado automáticamente por la base de datos.
- *nombre*.- Nombre que permite identificar la categoría. Es un campo requerido y único, que no admite caracteres especiales.
- *descripcion*.- Atributo que describe a la categoría. No acepta caracteres especiales, y tiene una longitud máxima de 100 caracteres.

***log*.-** Representa el registro de las transacciones que se han realizado en el sistema.

- *idregitrolog*.- Número entero consecutivo, que representa al identificador de la transacción. No puede ser administrado

manualmente, porque su valor es asignado automáticamente por la base de datos.

- *hora*.- Hora en la que se realizó la transacción.
- *fecha*.- Fecha en la que se realizó la transacción.
- *usuarioexterno*.- Indica el nombre del usuario externo que ha intervenido en una comunicación instantánea. Tiene una longitud máxima de 30 caracteres.
- *nombreusuario*.- Especifica el nombre del usuario interno que ha intervenido de alguna manera en la transacción. Este atributo tiene una longitud máxima de 30 caracteres.
- *nombreportero*.- Representa el nombre del portero desde el que se llevó a cabo la transacción. Este atributo tiene una longitud máxima de 15 caracteres.
- *transaccion*.- Representa el tipo de transacción que se ha registrado. Puede tomar alguno de los siguientes valores: "Comunicación", "Abrir Puerta" o "Guardar Mensaje".

***estado*.-** Representa los distintos estados de conexión que puede tener un usuario interno. Existen tres estados predeterminados, que son creados en el momento de instalar el sistema. El estado cuyo identificador es 1, que representa el estado de conexión disponible; el estado cuyo identificador es 2, que representa el estado ausente; y el estado con identificador 3, que representa el estado desconectado.

- *idestado*.- Número entero consecutivo, que representa al identificador del estado. No puede ser administrado manualmente, porque su valor es asignado automáticamente por la base de datos.
- *nombre*.- Nombre con el cual se desee identificar al estado. Es un campo requerido y único, que no acepta caracteres especiales.
- *activo*.- Atributo de tipo lógico que puede tomar valores de "0" o "1". Si el atributo tiene el valor de "1", todo usuario que tenga asociado este estado puede recibir solicitudes de comunicación desde el portero que lo requiera (es equivalente al estado disponible). Este atributo solamente puede modificado para los estados que se han creado manualmente, y su valor por defecto es "0".
- *habilitado*.- Atributo de tipo lógico que puede tomar valores de "0" o "1". Permite determinar si el estado de conexión puede ser utilizado en el sistema. Su valor por defecto es "0".
- *descripcion*.- Atributo que describe al estado. No admite caracteres especiales y su longitud máxima es de 100.

***configuracion_sistema*.-** Representa el conjunto de valores con los que el sistema ha sido configurado.

- *idconfigsistema*.- Número entero consecutivo, que representa al identificador del registro de configuración. No puede ser administrado manualmente, porque su valor es asignado automáticamente por la base de datos.

- *ipservidor*.- Representa la dirección IP de la máquina que ejecuta el servidor del sistema.
- *useradmin*.- Representa al usuario del administrador del sistema. Es un campo que no admite caracteres especiales, y su longitud puede ser de cinco a ocho caracteres.
- *pwdadmin*.- Atributo que representa a la contraseña del administrador del sistema.
- *espaciodiscousersinoficina*.- Es un valor entero, que se expresa en MB. Representa el espacio que se asigna a cada uno de los directorios de mensajes recibidos, de los usuarios que no disponen de oficinas.
- *espaciodiscousersconoficina*.- Es un valor entero, que se expresa en MB. Representa el espacio que se asigna a cada uno de los directorios de mensajes recibidos, de los usuarios que disponen de oficinas.
- *timercomunicacion*.- Es un valor entero, que representa el tiempo que un usuario externo puede esperar respuesta por parte del usuario interno, en el proceso de solicitar el establecimiento de una comunicación instantánea.
- *tiempomensaje*.- Es un valor entero, entre 1 y 120, que representa los segundos que una mensaje de audio o vídeo puede durar.
- *carpeta*.- Especifica el nombre de la carpeta del servidor FTP, en la que se alojarán los mensajes de audio o vídeo que se hayan enviado a los usuarios internos.

- *email_admin*.- Indica la dirección de correo electrónico del administrador del sistema. Este atributo es requerido y puede tener una longitud máxima de 50 caracteres.
- *ftpuser*.- Representa el usuario de la cuenta configurada en el servidor FTP, con permiso para descargar y subir los archivos correspondientes a los mensajes al servidor.
- *ftppwd*.- Representa la contraseña de la cuenta configurada en el servidor FTP, con permiso para descargar y subir los archivos correspondientes a los mensajes al servidor.
- *ip_servidor_correo*.- Especifica la dirección IP de la máquina que ejecuta el servidor de correo electrónico.

APÉNDICE E

DISPOSITIVO ELECTRÓNICO QUE CONTROLA LA APERTURA DE LA CERRADURA ELÉCTRICA

El sistema SCCIO requiere un dispositivo electrónico para poder abrir la puerta del área de oficinas, o detectar cuando ésta es abierta manualmente. Éste dispositivo se comunica con el computador, que tiene instalado el sistema portero, por medio del puerto serial COM1.

El dispositivo electrónico manda a abrir la puerta por medio de un relay, que permite o impide el paso de la corriente eléctrica según se haya configurado. Es decir, el circuito, brinda la opción de especificar si el relay ha de funcionar como NA (normalmente abierto, al ser accionado se cierra el relay) o NC (normalmente cerrado, al ser accionado se abre el relay); con la finalidad de que pueda funcionar utilizando distintos tipos de cerraduras eléctricas. El tiempo que el relay permanecerá abierto o

cerrado, de acuerdo al modo de operación especificado, es configurable; y deberá indicarse por medio de una señal enviada desde el computador, que mandará a incrementar o decrementar el valor actual en 0.5 segundos. El valor mínimo para el tiempo es de 0.5 y el máximo, de 5 segundos.

El siguiente diagrama de bloques muestra el esquema de funcionamiento del circuito electrónico.

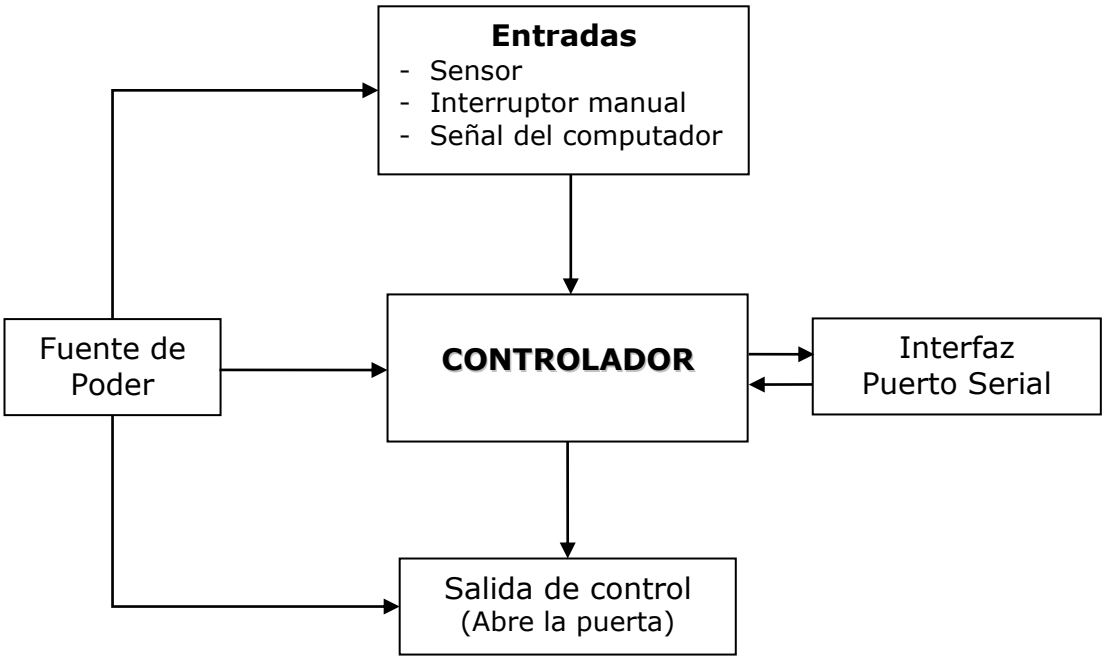


Figura 1.- Diagrama de bloques del circuito electrónico que controla la apertura de la cerradura eléctrica.

Las entradas del circuito son las señales que vienen del computador, del interruptor manual (botón que abre la puerta), y de un sensor que

detectará cuando la puerta sea abierta o cerrada (permite determinar si ésta se abrió usando la llave, sin hacer uso del computador o interruptor manual). La salida del circuito hacia la cerradura se representa simplemente por el paso u obstrucción de la corriente eléctrica. Finalmente la interfaz del puerto serial es la que se encarga de recibir datos del computador para que el circuito realice alguna operación, o de enviárselos cuando se necesite indicar el estado actual del sistema que controla la puerta.

El dispositivo electrónico se comunica con el computador a través del puerto COM1, utilizando un cable de tres hilos de alambre de cobre, con un conector macho tipo DB-9. Los pines utilizados se muestran a continuación.

PIN1	NC
PIN2	**TX**
PIN3	**RX**
PIN4	NC
PIN5	**GND**
PIN6	NC
PIN7	NC
PIN8	NC
PIN9	NC

El pin2 es usado para transmitir las señales, el pin3 para recibirlas, y el pin5 se lo utiliza como el común; los demás pines, marcados como NC, no se conectan. Las señales enviadas y recibidas entre el dispositivo electrónico y el computador son datos interpretados en código ASCII.

Las instrucciones que se envían del computador hacia el circuito electrónico, para que éste realice alguna operación, se detallan a continuación:

Dato	Orden a ejecutarse
R	Incrementar el tiempo de activación de la cerradura
L	Decrementar el tiempo de activación de la cerradura
O – o	Activar o abrir la cerradura
D	Consultar el tiempo de activación de la cerradura

Tabla 1.- Instrucciones que se envían del computador hacia el circuito electrónico.

La información que se envía del circuito electrónico hacia el computador, indicando el estado del sistema, se detalla a continuación:

Dato	Significado
C	La puerta fue cerrada (detectado por el sensor magnético)
A	La puerta fue abierta (detectado por el sensor magnético)
0-9	Valor actual del tiempo de activación de la cerradura. Enviado después de que el circuito haya recibido una "R", "L", o "D"
P	Se ha presionado el botón de activación manual *

Tabla 2.- Información que se envía del circuito electrónico hacia el computador.

* El tiempo de activación de la cerradura al presionar el interruptor manual, será el que haya sido ajustado por el usuario mediante los comandos "R" o "L".

Por otro lado, para que el computador se comunique con el puerto COM, a nivel de software, se utiliza la librería *SerialPorts.dll*. Para que se pueda transmitir y recibir datos por este puerto se deberá crear un objeto del tipo *SerialPort* que haga referencia a todas las propiedades y métodos que permiten trabajar con los puertos COM. Una vez creado este objeto, deberá llamarse al método *Open,* que recibe como parámetro el número del puerto COM con el que se desea establecer comunicación (en este caso es "1").

La configuración utilizada por defecto es "9600,N,8,1"; lo que significa respectivamente, 9600 baudios, sin bit de paridad, 8 bits de datos y un bit de parada. Una vez que se haya indicado con qué puerto se va a trabajar se podrán enviar los datos por medio de la función *Send,* que recibe como parámetro un arreglo de bytes o el byte correspondiente al código ASCII del dato que se requiere enviar al dispositivo electrónico.

Adicionalmente, se utilizará el evento *OnRecvI,* para obtener los bytes, correspondientes al código ASCII del dato, que se haya enviado al computador desde el dispositivo electrónico. Finalmente, se llamará al método *Close* para cerrar el puerto con el que se estableció comunicación.

APÉNDICE F

CONFERENCEXP

ConferenceXP es una plataforma de investigación desarrollada por Microsoft, con el objetivo de proporcionar a las instituciones académicas, y en especial a las universidades, un medio para construir aplicaciones colaborativas, principalmente encaminadas a mejorar la experiencia del aprendizaje, tanto dentro como fuera de los salones de clase.

Esta herramienta integra transferencias de audio y vídeo de alto desempeño, y tecnologías de red, permitiendo establecer conexiones de alta calidad, y baja latencia, y proporcionando un mecanismo para el desarrollo de aplicaciones de vídeo-conferencia y colaboración en tiempo real. Al tratarse de una plataforma de investigación abierta, es posible diseñar e implementar aplicaciones de este tipo, que se apeguen a las circunstancias y requerimientos específicos de cada caso. A continuación

se mencionan algunas de las características más importantes del ConferenceXP.

- *Facilidad de uso*.**-** Las conferencias locales o en línea que se establecen por medio del ConferenceXP se basan en el uso de canales de comunicación, que representan lugares virtuales de reunión; para que dos o más personas puedan verse y escucharse de manera simultánea, deben primeramente pertenecer a un mismo canal. Los participantes de una conferencia pueden ser fácilmente añadidos y removidos, por medio de dos métodos simples (*Join* y *Leave*, respectivamente); y la configuración de la conferencia se puede establecer del lado de cada participante, dando valores de verdadero o falso a propiedades como *AutoSendAudio*, *AutoSendVideo*, *AutoPlayRemote*, *AutoPlayLocal*, entre otras. De este modo, en unas cuantas líneas de código es posible especificar todo lo que hace falta para establecer una comunicación, sea ésta de audio o de vídeo.

- *Escalabilidad*.- Soporta una arquitectura simple de "una computadora por nodo", de modo que no requiere una infraestructura basada en un servidor complejo, para compartir audio y vídeo de alta calidad entre múltiples puntos locales y remotos. Puede utilizar multicast para acomodar cantidades grandes de usuarios, con alto desempeño.

- *Tecnologías avanzadas*.- Soporta redes inalámbricas y Tablet PCs; integra tecnologías tales como Microsoft DirectShow, Microsoft Windows Media, y Microsoft DirectX.

Arquitectura del ConferenceXP

La arquitectura del ConferenceXP es de tipo punto a punto. Debido a que no involucra un servidor, se pueden prevenir tanto los cuellos de botella en el tráfico de la red como los puntos de falla únicos. Esta arquitectura permite a los desarrolladores crear sistemas que tomen ventaja de los protocolos subyacentes de red, conferencia, y aplicaciones distribuidas. Se encuentra divida en cuatro capas lógicas: Aplicación ConferenceXP, Capacidad ConferenceXP, API de Conferencia, y Transporte de Red.

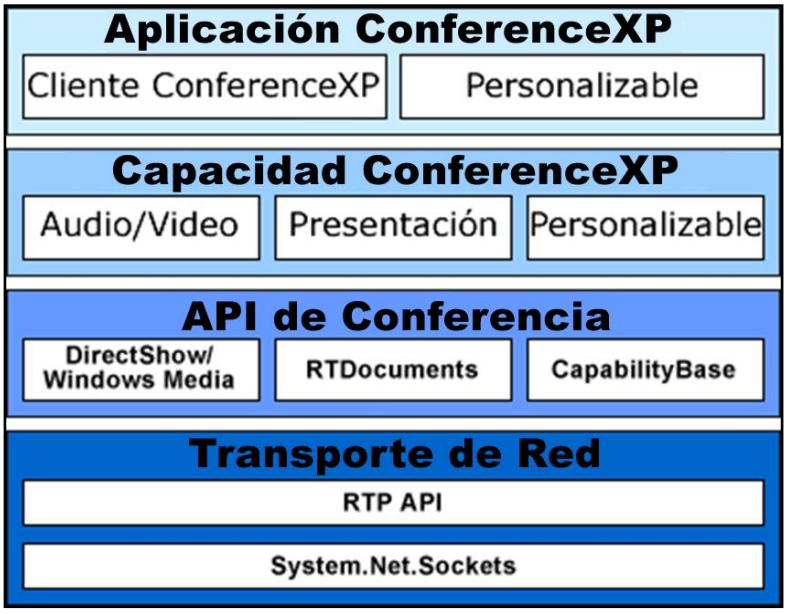


Figura 1.- Capas del ConferenceXP.

Las capas Aplicación y Capacidad proveen la interfaz de usuario para esta herramienta. Las "capacidades" son componentes que pueden ser añadidos y que proporcionan diferentes funcionalidades a la aplicación con el objetivo de que sean compartidas entre los participantes de un canal de comunicación, como por ejemplo, enviar y visualizar vídeo, PowerPoint distribuido, pizarrón compartido y trazos de tinta (ink), foros de preguntas y respuestas, etc. Estas dos capas usan el API de conferencia. La capa de Capacidad incluye las funcionalidades de *Audio/Vídeo*, que permite a una aplicación ConferenceXP enviar y recibir este tipo de contenido; y de *Presentación*, que permite compartir documentos y trazos de tinta (ink).

La capa del API de Conferencia, por otro lado, permite que los desarrolladores puedan crear fácil y rápidamente una aplicación ConferenceXP o capacidad. Por ejemplo, las capacidades pueden heredarse de *CapabilityBase*, que es la clase encargada de llevar a cabo la mayor parte del trabajo, y que permite a los desarrolladores convertir la mayoría de las aplicaciones en una capacidad en menos de cien líneas de código. El *API RTDocuments* provee un protocolo que permite realizar transferencias de documentos y de "trazos de tinta" (ink strokes), para que las aplicaciones que utilicen estas herramientas puedan interoperar. El protocolo *RTDocuments* es una implementación Microsoft .NET de la especificación de intercambio IMS/SCORM. Las APIs de DirectShow y

Windows Media proveen acceso a características de audio y vídeo en Windows; permiten, por ejemplo, que las aplicaciones y capacidades ConferenceXP puedan conectar dispositivos con codecs, así como también enviar audio y vídeo por red.

La capa de transporte de red provee tecnología escrita a la medida, para asegurar que los flujos de audio, vídeo e información, en general, sean transmitidos con una mínima pérdida de datos por la red. Para poder llevar a cabo esta tarea, se utiliza una implementación del protocolo de transporte en tiempo real (RTP), que está basado a su vez en la implementación administrada de los sockets de Windows. El transporte de red RTP, punto a punto, es un estándar de la IETF para la transmisión de audio y vídeo; y está diseñado para escenarios en lo que se requiere baja latencia, como la conferencia de alto desempeño.

El uso de esta plataforma en el desarrollo del Sistema Computarizado de Comunicación y Control de Ingreso a Oficinas, proporcionó confiabilidad y eficiencia al proceso de establecer comunicaciones de audio y vídeo en tiempo real. Adicionalmente, la flexibilidad con que se ha desarrollado el código de esta herramienta permitió que pueda ser adaptada a las necesidades específicas de este proyecto.

APÉNDICE G

MANUAL DE INSTALACIÓN DEL SISTEMA

SCCIO Servidor

Prerrequisitos:

- Memoria: 512 MB o superior.
- Procesador: Pentium IV 3.2 GHz o superior.
- Capacidad de disco duro: 80 GB.
- Sistema Operativo: Windows XP Professional
- .NET Framework 1.1
- Servicios de Internet Information Server 5.1 (IIS) con el subcomponente *Servicios de Protocolo de transferencia de archivos (FTP) y FrontPage Server Extensions(FPSE)*

Pasos previos:

1. *Crear una cuenta de usuario para acceder vía FTP*:

   Deberá crear una cuenta, asignándole un usuario y una contraseña, para que las demás aplicaciones puedan acceder con ella al servidor, vía FTP.

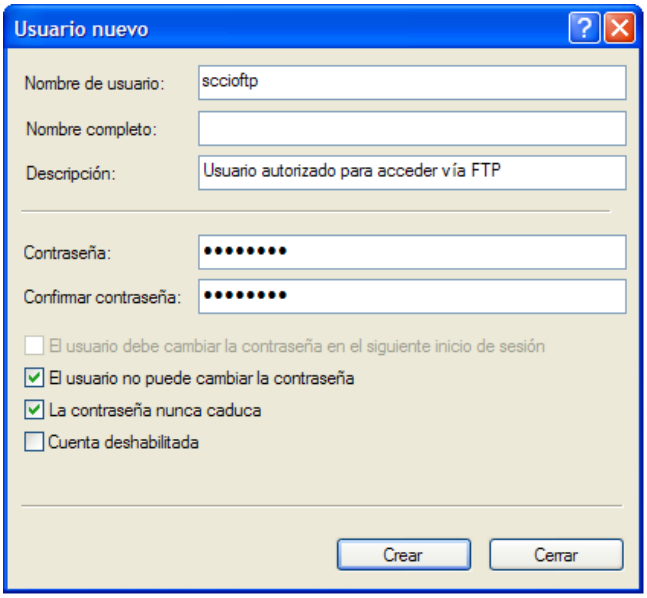


Figura 1.- Creación de la cuenta de usuario para acceder vía FTP

2. *Conceder permisos de lectura y escritura sobre el directorio FTP al nuevo usuario*:

   En el *Panel de control* haga doble clic sobre el ítem *Herramientas administrativas*, y luego sobre la opción *Servicios de Internet Information Server*. En esta ventana, haga clic con el botón derecho del ratón en el ítem *Sitio FTP predeterminado*, y escoja la opción *Propiedades*. A continuación aparecerá la ventana de propiedades, en

la que deberá hacer clic sobre la pestaña *Directorio particular*. Aquí usted podrá marcar las casillas de verificaciones de Lectura y Escritura, para permitir que se lleven a cabo estas operaciones sobre el directorio FTP, que aparece en la ruta de acceso local, tal como se muestra en la figura 2.

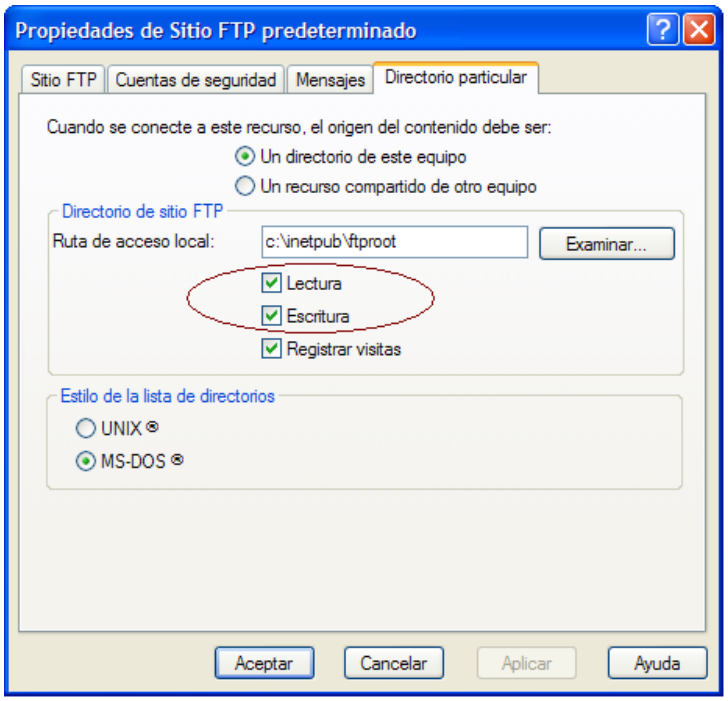


Figura 2.- "Directorio particular" en la ventana de propiedades de Sitio FTP predeterminado

Una vez que haya realizado el paso anterior, haga clic sobre la pestaña *Cuentas de seguridad*. En el campo Nombre de usuario, ingrese el usuario de la cuenta que creó anteriormente, y marque las casillas de verificación tal como se muestra en la figura 3.

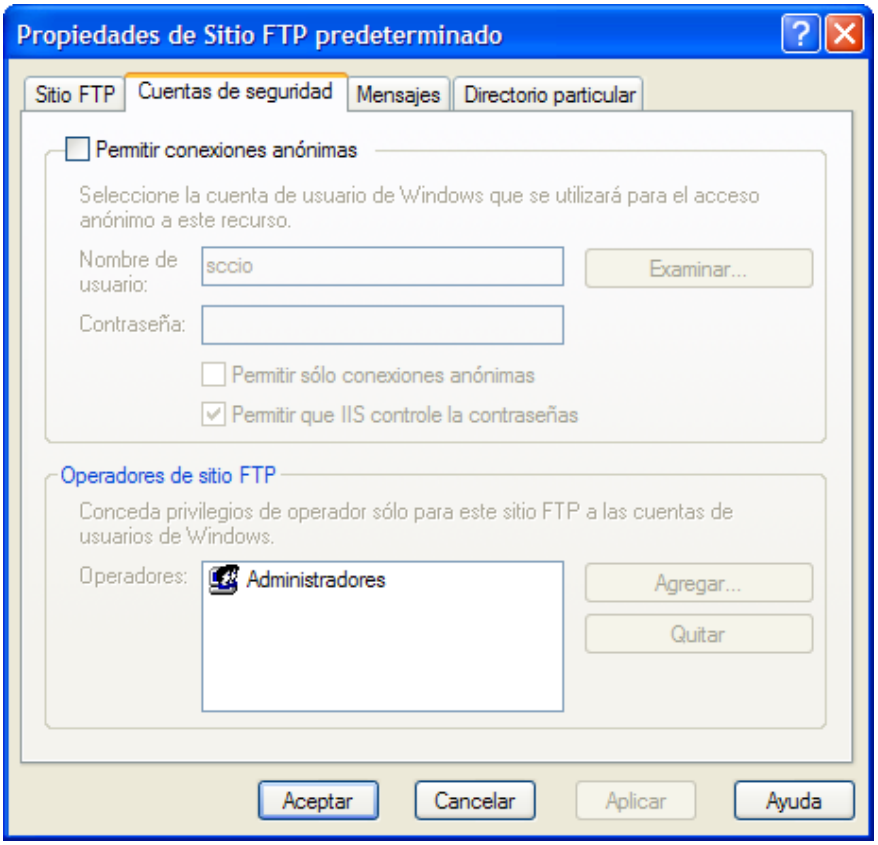


Figura 3.- "Cuentas de seguridad" en la ventana de propiedades de Sitio FTP predeterminado

Instalación:

Una vez completados los pasos previos, usted ya está listo para instalar SCCIO Servidor. Para dar comienzo al proceso de instalación, debe ejecutar el archivo *Setup.exe*. Esto hará que aparezca la primera ventana del asistente para la instalación, que lo guiará a través de los pasos requeridos. De clic en el botón que indica *Siguiente* para continuar.

En la siguiente ventana del asistente, que se muestra en la Figura 4, se solicitará el ingreso de los datos que permitirán al servidor establecer

comunicación con la base de datos y especificar el puerto de comunicación.

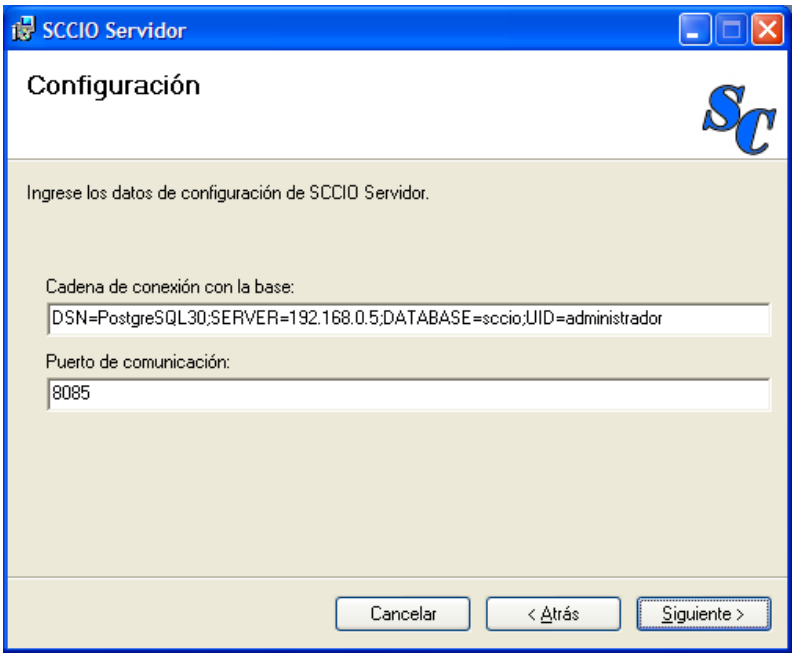


Figura 4.- Configuración de la cadena ODBC de conexión con la base de datos, y del puerto de comunicación.

- En el campo *Cadena de conexión con la base*, deberá especificar la cadena ODBC de conexión con la base de datos, compuesta por el nombre DSN, la dirección IP del servidor que contendrá la base de datos, el usuario, y la contraseña de acceso.
- En el campo *Puerto de comunicación*, se indicará el número de puerto por medio del cual el servidor ofrecerá sus servicios a las demás aplicaciones del sistema. Debe asegurarse de que este dato coincida con el que se especifique en la instalación de las demás aplicaciones del sistema.

Una vez que los datos solicitados hayan sido ingresados, puede dar clic en el botón que indica *Siguiente* para continuar con la instalación.

En la siguiente ventana que presenta el asistente de instalación se deberá indicar el tipo de canal de comunicación que SCCIO Servidor utilizará, seleccionando alguna de las dos opciones que se ofrecen: TCP (por defecto) o HTTP. Debe tener en cuenta que las aplicaciones sólo podrán establecer comunicación con el servidor, si el tipo de canal configurado en ellas es el mismo que el del servidor. La ventana del asistente que permite especificar este dato se muestra a continuación:

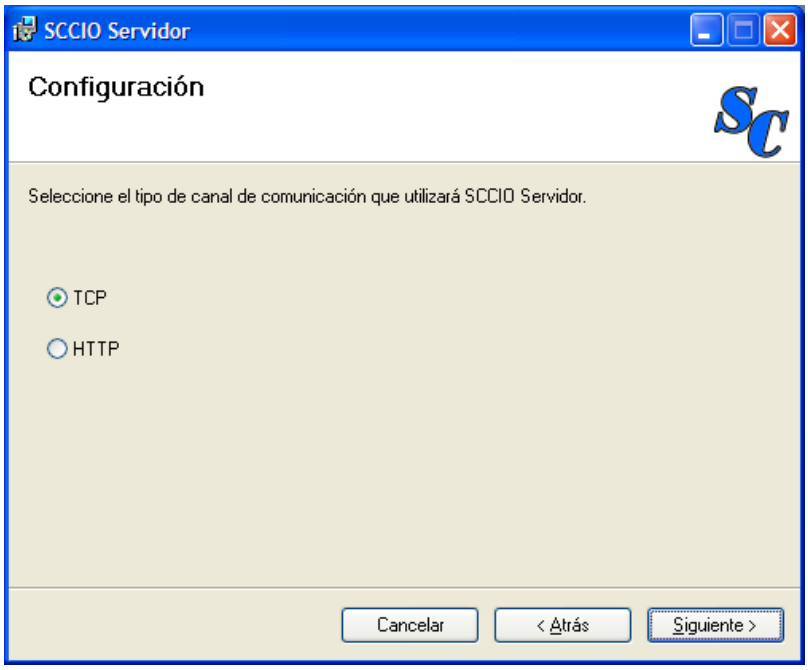


Figura 5.- Selección del tipo de canal de comunicación a utilizar.

Una vez que haya seleccionado el tipo de canal de comunicación que utilizará SCCIO Servidor, haga clic en el botón que indica *Siguiente* para continuar.

En la ventana a continuación deberá ingresar los siguientes datos:

- En el campo *Número máximo de registros en la bitácora*, deberá especificar la cantidad de registros que podrá llegar a contener la bitácora, en condiciones normales. Cuando este valor sea superado, el sistema enviará un email de advertencia al administrador de SCCIO.

- En el campo *Dirección IP del servidor de correo electrónico*, deberá especificar, tal como lo sugiere el nombre, la dirección IP del servidor encargado de manejar el correo electrónico dentro de la organización.

- En el campo *Dirección de correo electrónico del administrador de SCCIO Servidor*, deberá especificar la dirección de correo electrónico que se deberá utilizar para enviar todos los mensajes del sistema SCCIO, que vayan dirigidos al administrador.

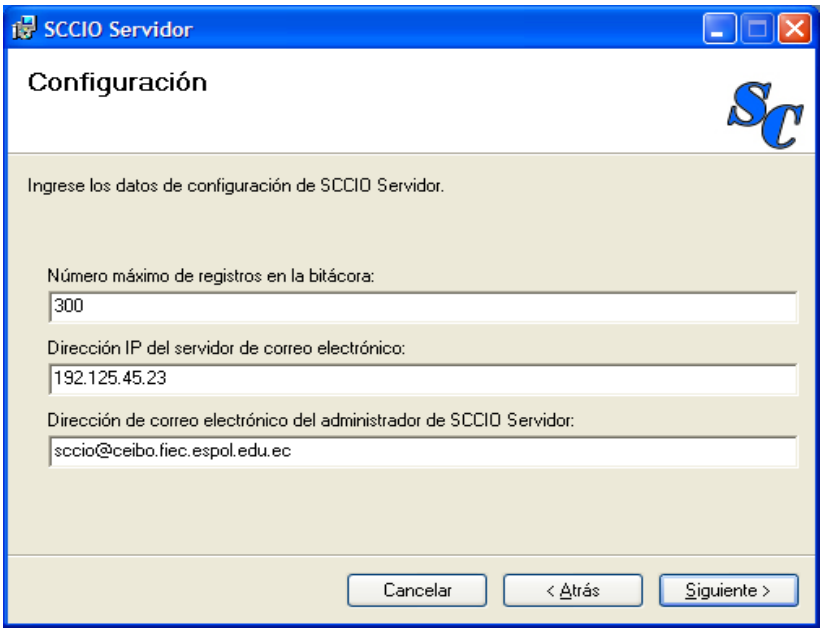


Figura 6.- Configuración de los datos del número máximo de registros en la bitácora, la dirección IP del servidor de correo electrónico y dirección de email del administrador.

En la siguiente ventana, deberá ingresar los datos correspondientes al usuario y la contraseña del administrador del sistema y de la cuenta FTP. En el caso de la cuenta FTP, los datos ingresados deberán coincidir con los de la cuenta que se especificó anteriormente.

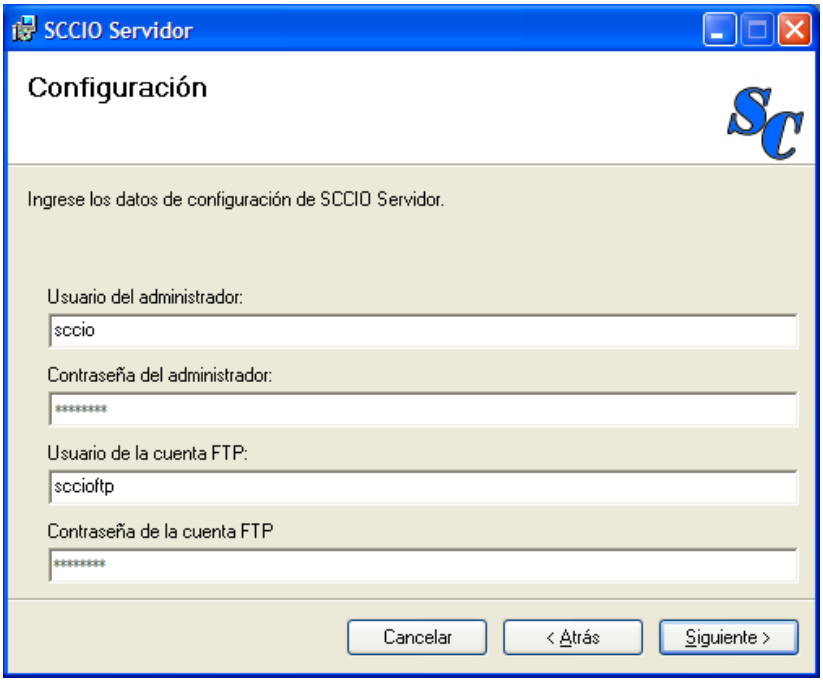


Figura 7.- Ingreso de usuarios y contraseñas del administrador y de la cuenta FTP.

En esta etapa del proceso deberá especificar la ruta en la que desea que se instale la aplicación SCCIO Servidor. Por defecto, el sistema se instalará dentro del directorio *SCCIO*, en la carpeta *SCCIO Servidor*, en la ruta predeterminada en la que se instalan los programas.

Como puede observarse en la figura 8, junto al botón *Examinar*, que permite buscar y seleccionar la ruta de instalación deseada, se encuentra un botón que indica *Espacio en disco*. Al hacer clic sobre este botón, se presentará una pequeña ventana en la que podrá visualizar la información correspondiente a cada una de las unidades del disco (espacio asignado, espacio disponible, espacio requerido por la instalación, espacio libre en

caso de que se instalara la aplicación allí). El objetivo de esta información es que le sirva de ayuda para decidir en qué unidad resultará más conveniente instalar la aplicación, en caso de que existan varias.

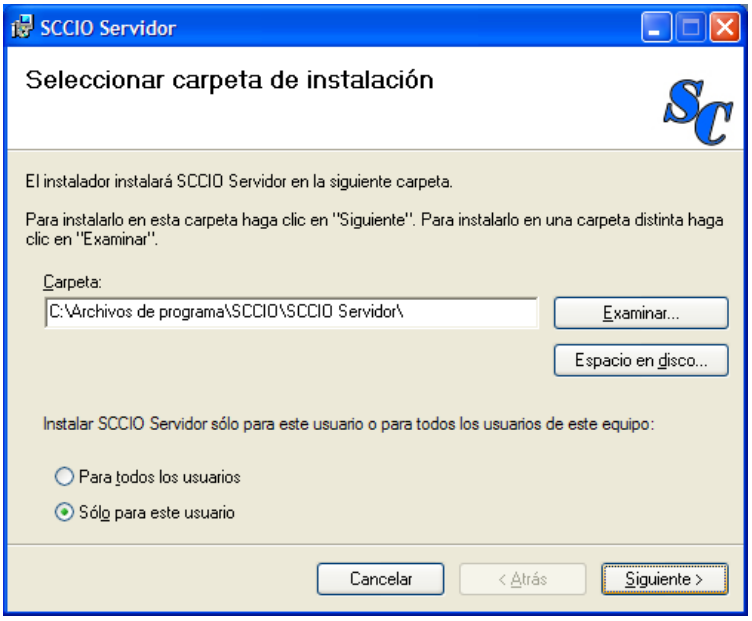


Figura 8.- Selección de la ruta de instalación para la aplicación.

Adicionalmente, deberá especificar si desea que SCCIO Servidor se instale únicamente para el usuario de la sesión actual, o para todos los usuarios que utilicen la computadora. Una vez, que haya indicado todos los datos solicitados, puede hacer clic en el botón *Siguiente*, para continuar.

Ahora que todos los datos solicitados para instalar SCCIO Servidor han sido ingresados, se mostrará la ventana de confirmación, en la que deberá presionar el botón *Siguiente*, para que se lleve a cabo la

instalación de la aplicación. Con esto, el asistente procederá a copiar los archivos necesarios para ejecutar la aplicación, en la ruta especificada anteriormente. Cuando el proceso haya concluido, aparecerá una ventana indicando que la instalación de SCCIO Servidor se ha completado, y solicitando que se presione el botón *Cerrar* para salir del asistente.

Finalmente, usted podrá ejecutar la aplicación, haciendo doble clic en el acceso directo que se habrá creado en el escritorio, o en el archivo *SCCIOServidor.exe*, ubicado en la ruta de instalación.

SCCIO Usuario Web

Esta aplicación deberá instalarse en la misma computadora que SCCIO Servidor, deberá cumplir con los mismos prerrequisitos.

Instalación:

Para dar comienzo a la instalación de la aplicación SCCIO Usuario Web, debe ejecutar el archivo *Setup.exe*. Esto hará que aparezca la primera ventana del asistente para la instalación, que lo guiará a través de los pasos requeridos. De clic en el botón que indica *Siguiente* para continuar.

En la siguiente ventana del asistente se solicitará el ingreso de los datos que permitirán a la aplicación establecer comunicación con el servidor del sistema.

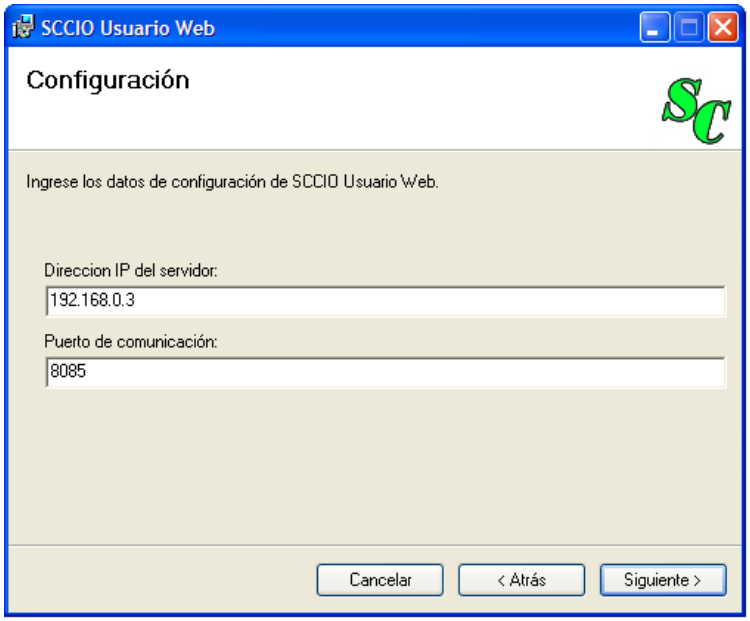


Figura 9.- Configuración de la dirección IP y del puerto de comunicación requeridos para la comunicación con el servidor.

- En el campo *Dirección IP del servidor* deberá especificar la dirección IP de la computadora que tendrá instalada y ejecutará la aplicación SCCIO Servidor.

- En el campo *Puerto de comunicación*, se indicará el número de puerto por medio del cual la aplicación podrá acceder a los servicios que ofrece el servidor. Debe asegurarse de que este dato coincida con el que se especificó en la instalación de SCCIO Servidor.

Una vez que los datos solicitados hayan sido ingresados, puede dar clic en el botón que indica *Siguiente* para continuar con la instalación.

En la siguiente ventana que presenta el asistente de instalación se deberá indicar el tipo de canal de comunicación que la aplicación utilizará, seleccionando alguna de las dos opciones que se ofrecen: TCP (por defecto) o HTTP. Debe tener en cuenta que las aplicaciones sólo podrán establecer comunicación con el servidor, si el tipo de canal configurado en éste es el mismo. La ventana del asistente que permite especificar este dato se muestra a continuación:

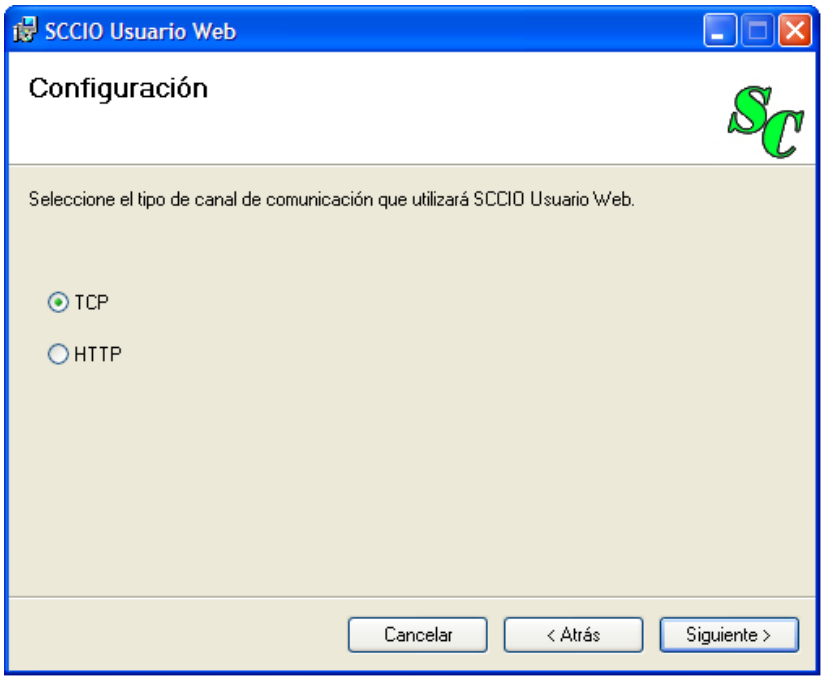


Figura 10.- Selección del tipo de canal de comunicación a utilizar.

Una vez que haya seleccionado el tipo de canal de comunicación que utilizará la aplicación, haga clic en el botón que indica *Siguiente* para continuar.

Ahora que todos los datos solicitados para instalar SCCIO Usuario Web han sido ingresados, se mostrará la ventana de confirmación, en la que deberá presionar el botón *Siguiente*, para que se lleve a cabo la instalación de la aplicación. Con esto, el asistente procederá a copiar las páginas ASP y los archivos necesarios para la aplicación, en la ruta *C:\Inetpub\wwwroot\SCCIOUsuarioWeb*.

Cuando el proceso haya concluido, aparecerá una ventana indicando que la instalación de SCCIO Usuario Web se ha completado, y solicitando que se presione el botón *Cerrar* para salir del asistente.

Usted podrá verificar que las páginas ASP, que constituyen la aplicación, y los archivos correspondientes requeridos, se habrán copiado en la ruta especificada anteriormente. Para cargar la página inicial de SCCIO Usuario Web, puede abrir el Internet Explorer, y escribir la dirección *http://localhost/SCCIOUsuarioWeb/*, tal como se muestra en la figura 11.

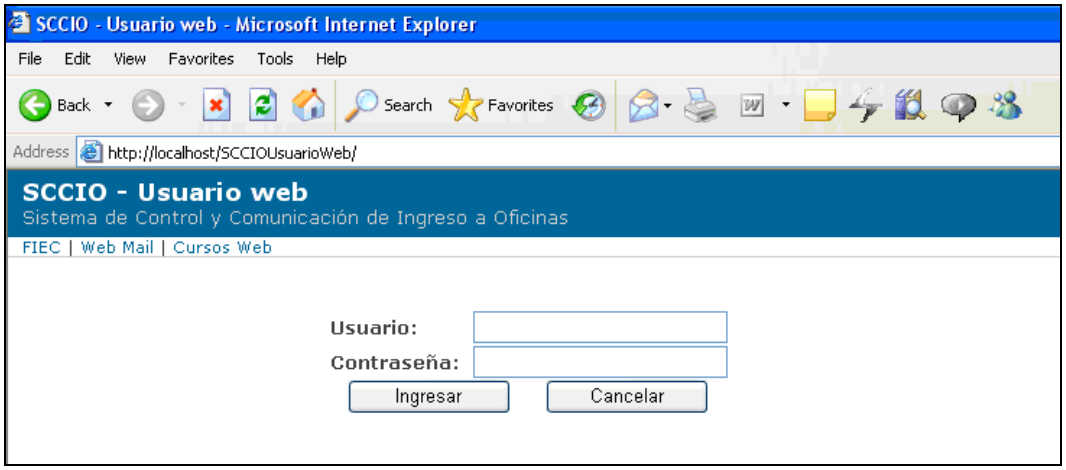


Figura 11.- Página inicial de SCCIO Usuario Web.

SCCIO Administrador

Prerrequisitos:

- Memoria: 128 MB o superior.
- Procesador: Pentium III o superior.
- Capacidad de disco duro: 40 GB o incluso menor (se requieren 10 MB para la instalación).
- Sistema Operativo: Windows XP
- .NET Framework 1.1

Instalación:

Para dar comienzo a la instalación de la aplicación SCCIO Administrador, debe ejecutar el archivo *Setup.exe*. Esto hará que aparezca la primera

ventana del asistente para la instalación, que lo guiará a través de los pasos requeridos. De clic en el botón que indica *Siguiente* para continuar.

En la siguiente ventana del asistente, se solicitará el ingreso de los datos que permitirán a la aplicación establecer comunicación con el servidor del sistema.

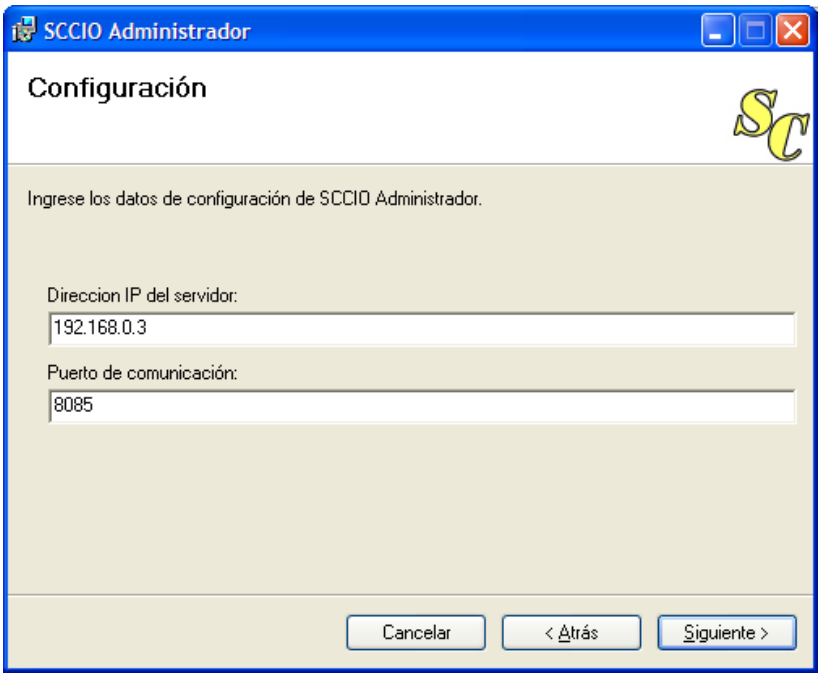


Figura 12.- Configuración de la dirección IP y del puerto de comunicación requeridos para la comunicación con el servidor.

- En el campo *Dirección IP del servidor* deberá especificar la dirección IP de la computadora que tendrá instalada y ejecutará la aplicación SCCIO Servidor.

- En el campo *Puerto de comunicación*, se indicará el número de puerto por medio del cual la aplicación podrá acceder a los servicios que ofrece el servidor. Debe asegurarse de que este dato coincida con el que se especificó en la instalación de SCCIO Servidor.

Una vez que los datos solicitados hayan sido ingresados, puede dar clic en el botón que indica *Siguiente* para continuar con la instalación.

En la siguiente ventana que presenta el asistente de instalación se deberá indicar el tipo de canal de comunicación que la aplicación utilizará, seleccionando alguna de las dos opciones que se ofrecen: TCP (por defecto) o HTTP. Debe tener en cuenta que las aplicaciones sólo podrán establecer comunicación con el servidor, si el tipo de canal configurado en éste es el mismo. La ventana del asistente que permite especificar este dato se muestra a continuación:

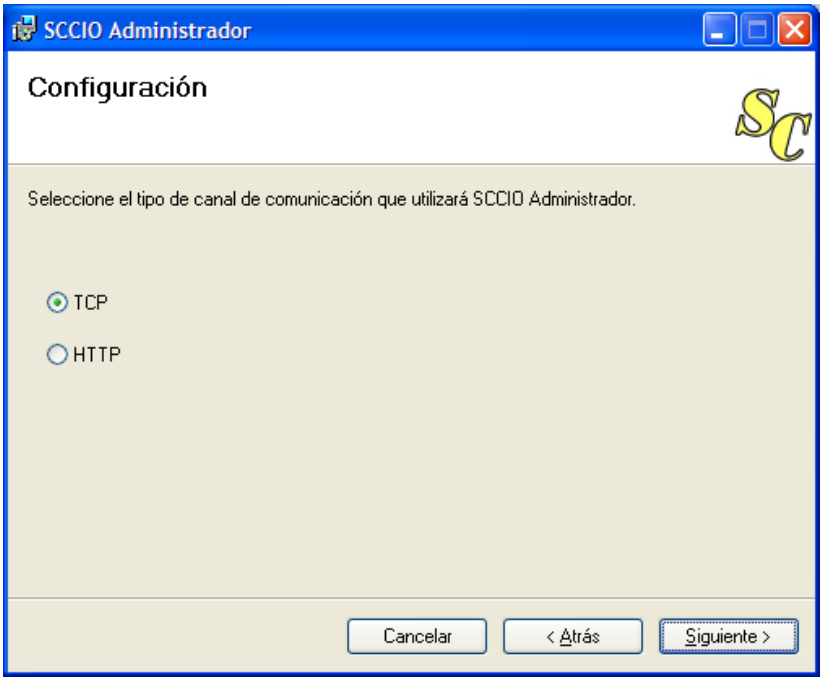


Figura 13.- Selección del tipo de canal de comunicación a utilizar.

Una vez que haya seleccionado el tipo de canal de comunicación que utilizará la aplicación, haga clic en el botón que indica *Siguiente* para continuar.

En esta etapa del proceso deberá especificar la ruta en la que desea que se instale la aplicación SCCIO Administrador. Por defecto, el sistema se instalará dentro del directorio *SCCIO*, en la carpeta *SCCIO Administrador*, en la ruta predeterminada en la que se instalan los programas.

Como puede observarse en la figura 14, junto al botón *Examinar*, que permite buscar y seleccionar la ruta de instalación deseada, se encuentra un botón que indica *Espacio en disco*. Al hacer clic sobre este botón, se

presentará una pequeña ventana en la que podrá visualizar la información correspondiente a cada una de las unidades del disco (espacio asignado, espacio disponible, espacio requerido por la instalación, espacio libre en caso de que se instalara la aplicación allí).

El objetivo de esta información es que le sirva de ayuda para decidir en qué unidad resultará más conveniente instalar la aplicación, en caso de que existan varias.

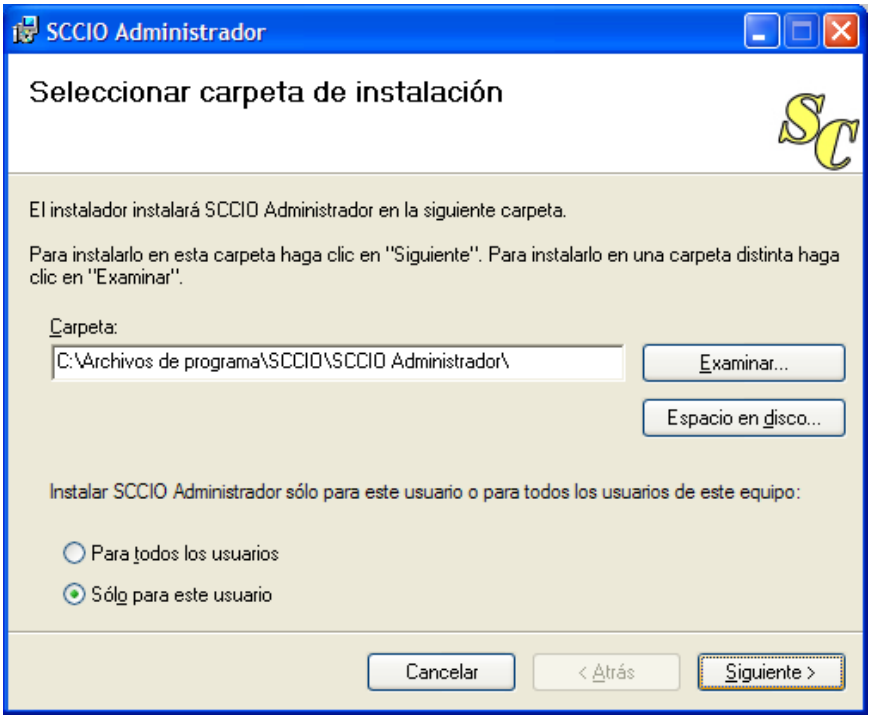


Figura 14.- Selección de la ruta de instalación para la aplicación.

Adicionalmente, deberá especificar si desea que SCCIO Administrador se instale únicamente para el usuario de la sesión actual, o para todos los usuarios que utilicen la computadora. Una vez, que haya indicado todos

los datos solicitados, puede hacer clic en el botón *Siguiente*, para continuar.

Ahora que todos los datos solicitados para instalar SCCIO Administrador han sido ingresados, se mostrará la ventana de confirmación, en la que deberá presionar el botón *Siguiente*, para que se lleve a cabo la instalación de la aplicación. Con esto, el asistente procederá a copiar los archivos necesarios para ejecutar la aplicación, en la ruta especificada anteriormente. Cuando el proceso haya concluido, aparecerá una ventana indicando que la instalación de SCCIO Administrador se ha completado, y solicitando que se presione el botón *Cerrar* para salir del asistente.

Finalmente, usted podrá ejecutar la aplicación, haciendo doble clic en el acceso directo que se habrá creado en el escritorio, o en el archivo *SCCIOAdministrador.exe*, ubicado en la ruta de instalación.

SCCIO Usuario

Prerrequisitos:

- Memoria: 512 MB o superior.
- Procesador: Pentium IV 2.26 GHz o superior.
- Capacidad de disco duro: 40 GB.

- Sistema Operativo: Windows XP
- .NET Framework 1.1

Instalación:

Para dar comienzo a la instalación de la aplicación SCCIO Usuario, debe ejecutar el archivo *Setup.exe*. Esto hará que aparezca la primera ventana del asistente para la instalación, que lo guiará a través de los pasos requeridos. De clic en el botón que indica *Siguiente* para continuar.

En la siguiente ventana del asistente, se solicitará el ingreso de los datos que permitirán a la aplicación establecer comunicación con el servidor del sistema, y con las aplicaciones SCCIO Portero.

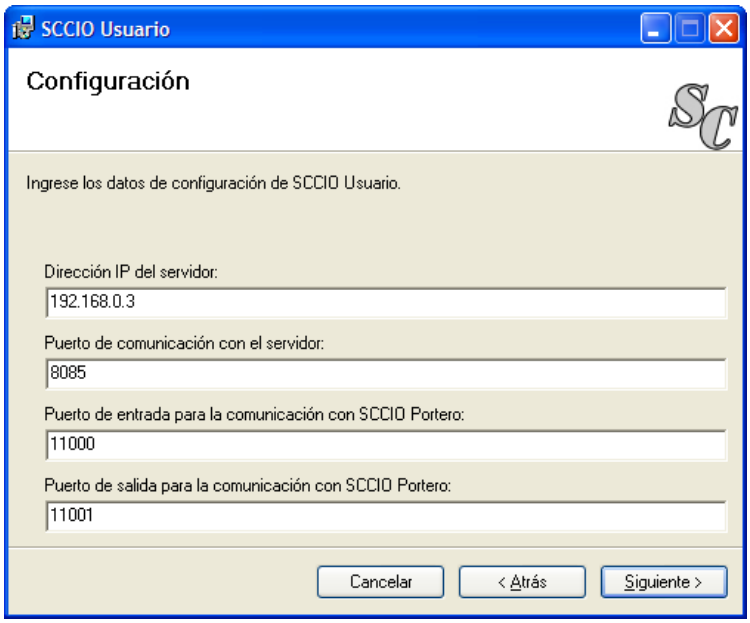


Figura 15.- Configuración de los datos requeridos para la comunicación con el servidor y con las aplicaciones SCCIO Portero.

- En el campo *Dirección IP del servidor* deberá especificar la dirección IP de la computadora que tendrá instalada y ejecutará la aplicación SCCIO Servidor.

- En el campo *Puerto de comunicación*, se indicará el número de puerto por medio del cual la aplicación podrá acceder a los servicios que ofrece el servidor. Debe asegurarse de que este dato coincida con el que se especificó en la instalación de SCCIO Servidor.

- En el campo *Puerto de entrada para la comunicación con SCCIO Portero*, se debe ingresar el número del puerto de "escucha", por el que la aplicación atenderá los mensajes que se le envíen desde las aplicaciones *SCCIO Portero*. El valor por defecto de este campo es 11000, pero si desea modificarlo, deberá verificar que el nuevo valor coincida con el del puerto de salida que se haya configurado en la instalación de las aplicaciones SCCIO Portero.

- En el campo *Puerto de salida para la comunicación con SCCIO Portero*, se debe ingresar el número del puerto por el que se enviarán los mensajes dirigidos a las aplicaciones *SCCIO Portero*. El valor por defecto de este campo es 11001, pero si desea modificarlo, deberá verificar que el nuevo valor coincida con el del puerto de entrada que se haya configurado en la instalación de las

aplicaciones SCCIO Portero. En general, el esquema que debe respetarse para que las aplicaciones puedan comunicarse es el siguiente:

Puerto de entrada en SCCIO Usuario = Puerto de salida en SCCIO Portero Puerto de salida en SCCIO Usuario = Puerto de entrada en SCCIO Portero

Una vez que los datos solicitados hayan sido ingresados, puede dar clic en el botón que indica *Siguiente* para continuar con la instalación.

En la siguiente ventana que presenta el asistente de instalación se deberá indicar el tipo de canal de comunicación que la aplicación utilizará, seleccionando alguna de las dos opciones que se ofrecen: TCP (por defecto) o HTTP. Debe tener en cuenta que las aplicaciones sólo podrán establecer comunicación con el servidor, si el tipo de canal configurado en éste es el mismo.

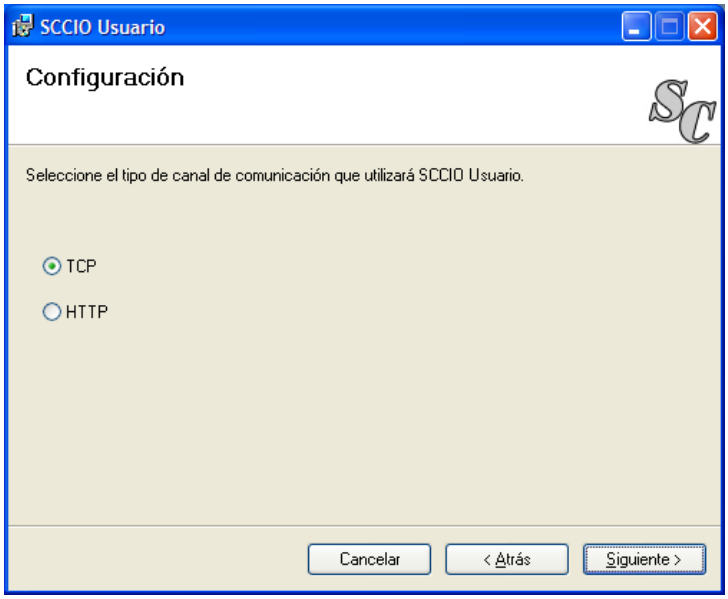


Figura 16.- Selección del tipo de canal de comunicación a utilizar.

Una vez que haya seleccionado el tipo de canal de comunicación que utilizará la aplicación, haga clic en el botón que indica *Siguiente* para continuar.

A continuación, tal como se muestra en la figura 17, se deberá especificar si la computadora en la que se está instalando la aplicación SCCIO Usuario, tiene un procesador equivalente a Pentium 4 o superior. El sistema utilizará esta información para determinar si debe permitir al usuario interno establecer comunicaciones de audio o vídeo, y visualizar el vídeo capturado por algún portero. Para continuar, haga clic en el botón que indica *Siguiente*.

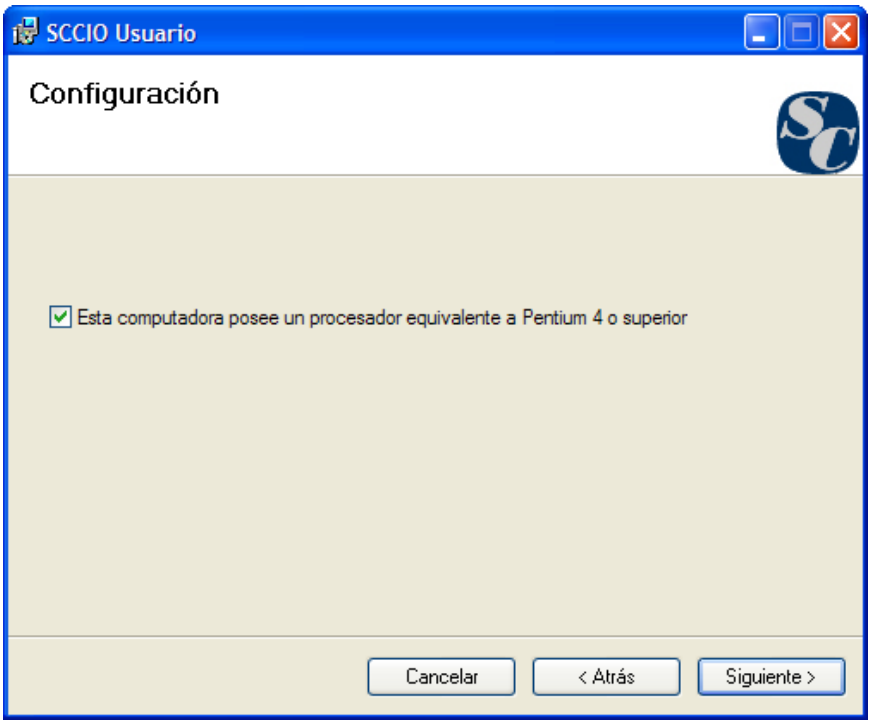


Figura 17.- Especificación del tipo de procesador.

En esta etapa del proceso deberá especificar la ruta en la que desea que se instale la aplicación SCCIO Usuario. Por defecto, el sistema se instalará dentro del directorio *SCCIO*, en la carpeta *SCCIO Usuario*, en la ruta predeterminada en la que se instalan los programas.

Como puede observarse en la figura 18, junto al botón *Examinar*, que permite buscar y seleccionar la ruta de instalación deseada, se encuentra un botón que indica *Espacio en disco*. Al hacer clic sobre este botón, se presentará una pequeña ventana en la que podrá visualizar la información correspondiente a cada una de las unidades del disco (espacio asignado, espacio disponible, espacio requerido por la instalación, espacio libre en caso de que se instalara la aplicación allí). El objetivo de esta información es que le sirva de ayuda para decidir en qué unidad resultará más conveniente instalar la aplicación, en caso de que existan varias.

Adicionalmente, deberá especificar si desea que SCCIO Usuario se instale únicamente para el usuario de la sesión actual, o para todos los usuarios que utilicen la computadora. Una vez, que haya indicado todos los datos solicitados, puede hacer clic en el botón *Siguiente*, para continuar.

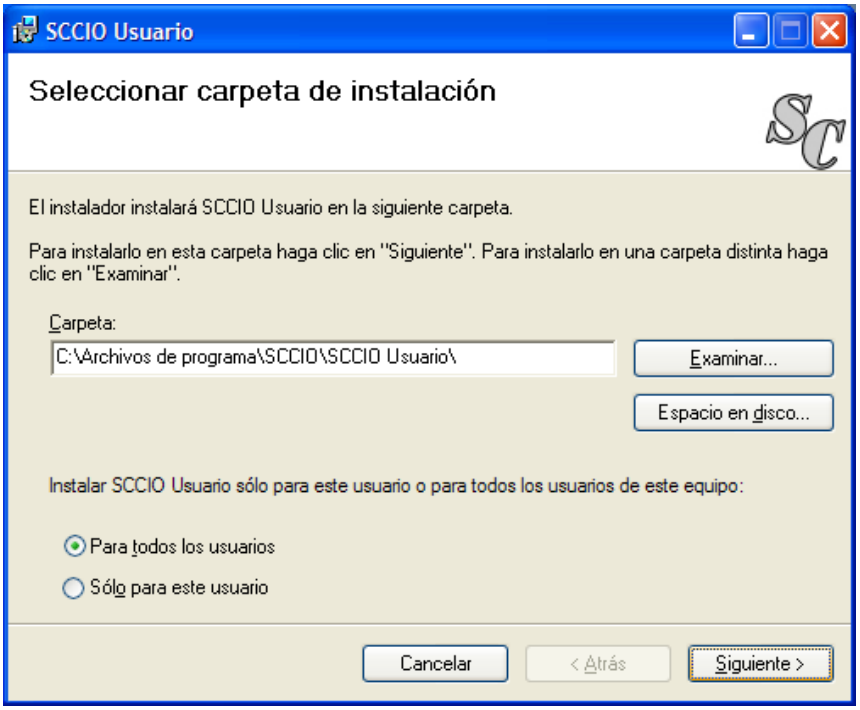


Figura 18.- Selección de la ruta de instalación para la aplicación.

Ahora que todos los datos solicitados para instalar SCCIO Usuario han sido ingresados, se mostrará la ventana de confirmación, en la que deberá presionar el botón *Siguiente*, para que se lleve a cabo la instalación de la aplicación. Con esto, el asistente procederá a copiar los archivos necesarios para ejecutar la aplicación, en la ruta especificada anteriormente. Cuando el proceso haya concluido, aparecerá una ventana indicando que la instalación de SCCIO Usuario se ha completado, y solicitando que se presione el botón *Cerrar* para salir del asistente.

Finalmente, usted podrá ejecutar la aplicación, haciendo doble clic en el acceso directo que se habrá creado en el escritorio, o en el archivo *SCCIOUsuario.exe*, ubicado en la ruta de instalación.

SCCIO Portero

Prerrequisitos:

- Memoria: 512 MB o superior.
- Procesador: Pentium IV 2.26 GHz o superior.
- Capacidad de disco duro: 40 GB.
- Sistema Operativo: Windows XP
- .NET Framework 1.1
- DirectX 8.1

Instalación:

Para dar comienzo a la instalación de la aplicación SCCIO Portero, debe ejecutar el archivo *Setup.exe*. Esto hará que aparezca la primera ventana del asistente para la instalación, que lo guiará a través de los pasos requeridos. De clic en el botón que indica *Siguiente* para continuar.

En la siguiente ventana del asistente, se solicitará el ingreso de los datos que permitirán a la aplicación establecer comunicación con el servidor del sistema, y con las aplicaciones SCCIO Usuario.

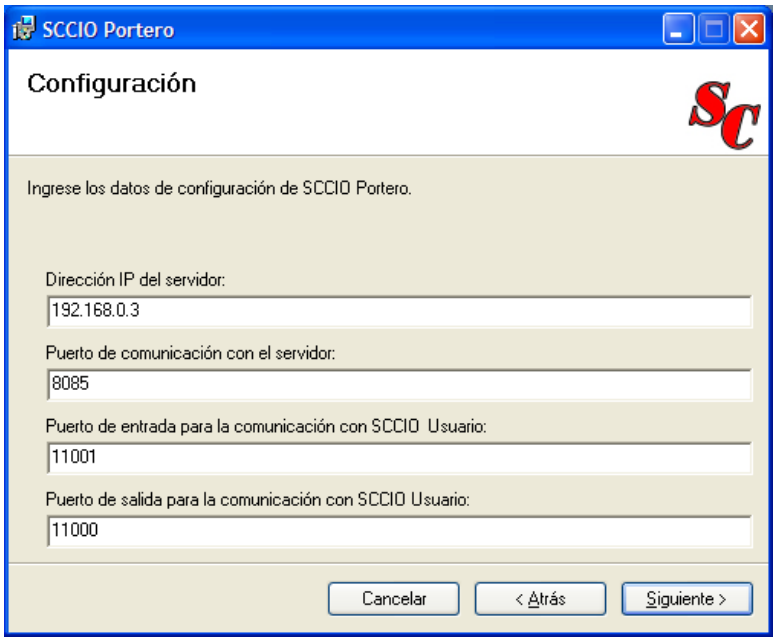


Figura 19.- Configuración de los datos requeridos para la comunicación con el servidor y con las aplicaciones SCCIO Usuario.

- En el campo *Dirección IP del servidor* deberá especificar la dirección IP de la computadora que tendrá instalada y ejecutará la aplicación SCCIO Servidor.

- En el campo *Puerto de comunicación*, se indicará el número de puerto por medio del cual la aplicación podrá acceder a los servicios que ofrece el servidor. Debe asegurarse de que este dato coincida con el que se especificó en la instalación de SCCIO Servidor.

- En el campo *Puerto de entrada para la comunicación con SCCIO Usuario*, se debe ingresar el número del puerto de "escucha", por el

que la aplicación atenderá los mensajes que se le envíen desde las aplicaciones *SCCIO Usuario*. El valor por defecto de este campo es 11001, pero si desea modificarlo, deberá verificar que el nuevo valor coincida con el del puerto de salida que se haya configurado en la instalación de las aplicaciones SCCIO Usuario.

- En el campo *Puerto de salida para la comunicación con SCCIO Usuario*, se debe ingresar el número del puerto por el que se enviarán los mensajes dirigidos a las aplicaciones *SCCIO Usuario*. El valor por defecto de este campo es 11001, pero si desea modificarlo, deberá verificar que el nuevo valor coincida con el del puerto de entrada que se haya configurado en la instalación de las aplicaciones SCCIO Usuario. En general, el esquema que debe respetarse para que las aplicaciones puedan comunicarse es el siguiente:

Puerto de entrada en SCCIO Usuario = Puerto de salida en SCCIO Portero Puerto de salida en SCCIO Usuario = Puerto de entrada en SCCIO Portero

Una vez que los datos solicitados hayan sido ingresados, puede dar clic en el botón que indica *Siguiente* para continuar con la instalación.

En la siguiente ventana que presenta el asistente de instalación se deberá indicar el tipo de canal de comunicación que la aplicación utilizará,

seleccionando alguna de las dos opciones que se ofrecen: TCP (por defecto) o HTTP. Debe tener en cuenta que las aplicaciones sólo podrán establecer comunicación con el servidor, si el tipo de canal configurado en éste es el mismo.

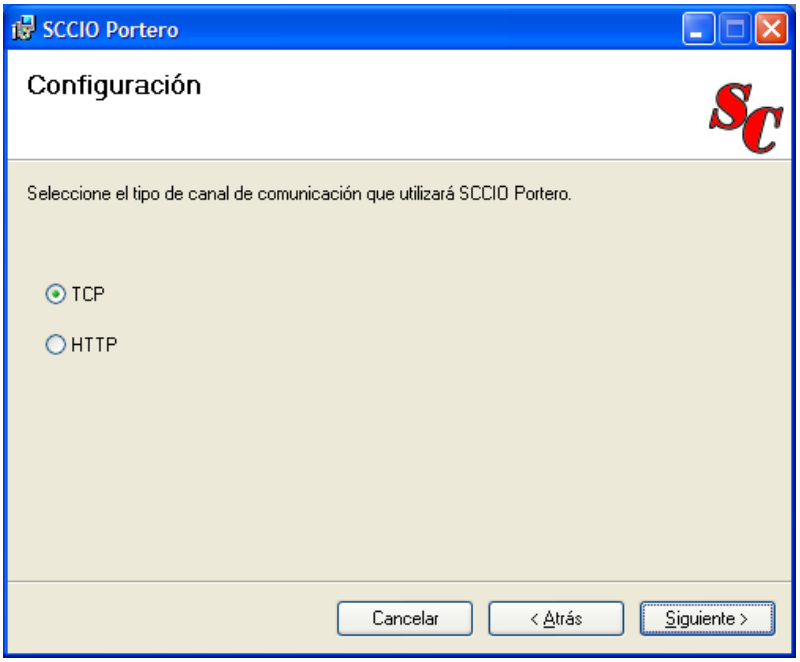


Figura 20.- Selección del tipo de canal de comunicación a utilizar.

Una vez que haya seleccionado el tipo de canal de comunicación que utilizará la aplicación, haga clic en el botón que indica *Siguiente* para continuar.

En la siguiente ventana del asistente, se solicitará el ingreso de una clave que servirá para bloquear o desbloquear el sistema. Debido a que las computadoras que ejecuten la aplicación SCCIO Portero sólo deberán permitir que se lleve a cabo esta única tarea, ciertas funciones del

sistema operativo deberán ser bloqueadas, como por ejemplo, el acceso al escritorio, a la barra de herramientas, y la utilización de determinadas combinaciones de teclas y funciones de control. Sin embargo, previendo que el administrador de SCCIO podría necesitar alguna de estas herramientas en algún momento, se incluye en el proceso de instalación el ingreso de la clave que permitirá deshabilitar el bloqueo.

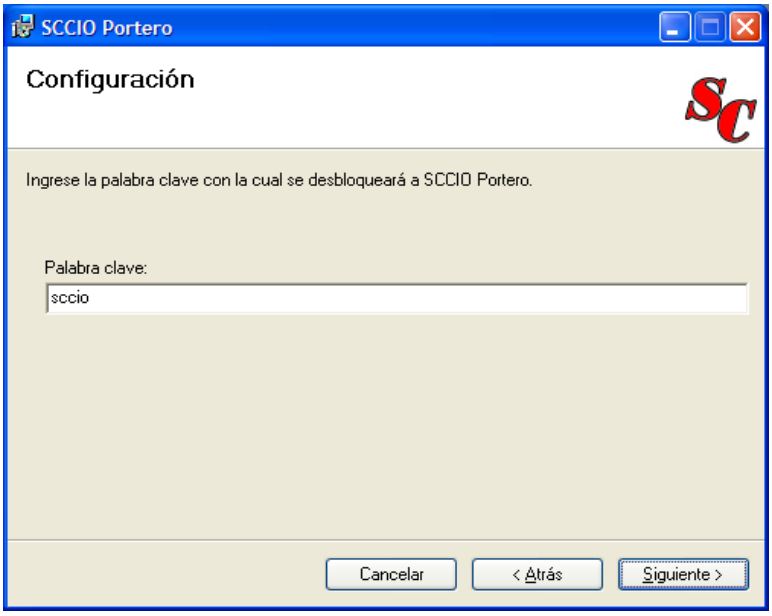


Figura 21.- Ingreso de la palabra clave para bloquear o desbloquear el sistema.

En esta etapa del proceso deberá especificar la ruta en la que desea que se instale la aplicación SCCIO Portero. Por defecto, el sistema se instalará dentro del directorio *SCCIO*, en la carpeta *SCCIO Portero*, en la ruta predeterminada en la que se instalan los programas.

Como puede observarse en la figura 22, junto al botón *Examinar*, que permite buscar y seleccionar la ruta de instalación deseada, se encuentra un botón que indica *Espacio en disco*. Al hacer clic sobre este botón, se presentará una pequeña ventana en la que podrá visualizar la información correspondiente a cada una de las unidades del disco (espacio asignado, espacio disponible, espacio requerido por la instalación, espacio libre en caso de que se instalara la aplicación allí). El objetivo de esta información es que le sirva de ayuda para decidir en qué unidad resultará más conveniente instalar la aplicación, en caso de que existan varias.

Adicionalmente, deberá especificar si desea que SCCIO Portero se instale únicamente para el usuario de la sesión actual, o para todos los usuarios que utilicen la computadora. Una vez, que haya indicado todos los datos solicitados, puede hacer clic en el botón *Siguiente*, para continuar.

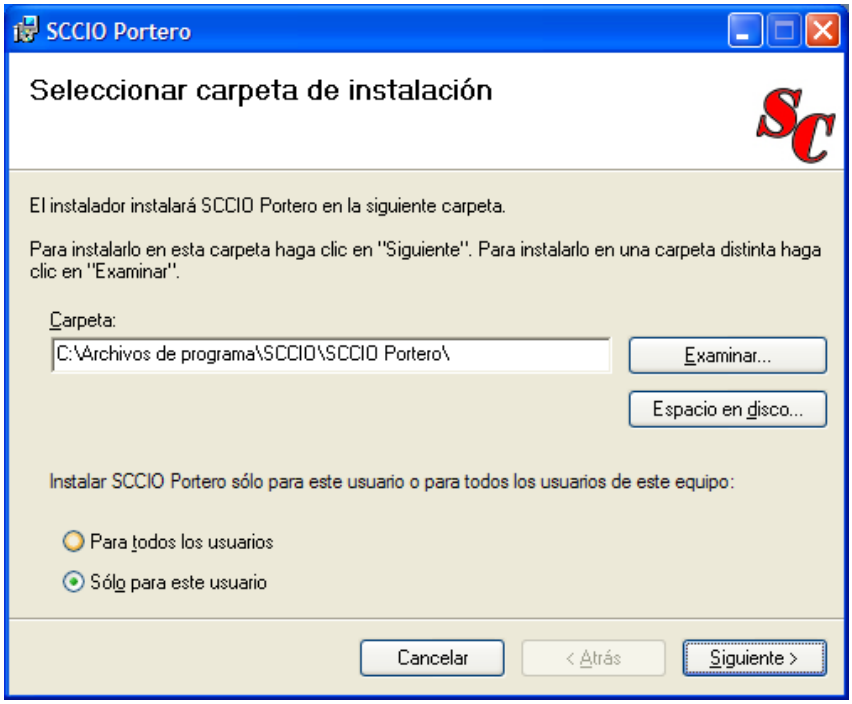


Figura 22.- Selección de la ruta de instalación para la aplicación.

Ahora que todos los datos solicitados para instalar SCCIO Portero han sido ingresados, se mostrará la ventana de confirmación, en la que deberá presionar el botón *Siguiente*, para que se lleve a cabo la instalación de la aplicación. Con esto, el asistente procederá a copiar los archivos necesarios para ejecutar la aplicación, en la ruta especificada anteriormente. Cuando el proceso haya concluido, aparecerá una ventana indicando que la instalación de SCCIO Portero se ha completado, y solicitando que se presione el botón *Cerrar* para salir del asistente.

Finalmente, usted podrá ejecutar la aplicación, haciendo doble clic en el acceso directo que se habrá creado en el escritorio, o en el archivo *SCCIOPortero.exe*, ubicado en la ruta de instalación.

Sin embargo, aún hace falta que se indique tanto el dispositivo de captura de vídeo como los compresores que utilizará la aplicación en el proceso de grabación de mensajes. Estos datos se especifican en la ventana que aparece cuando la instalación se ha completado.

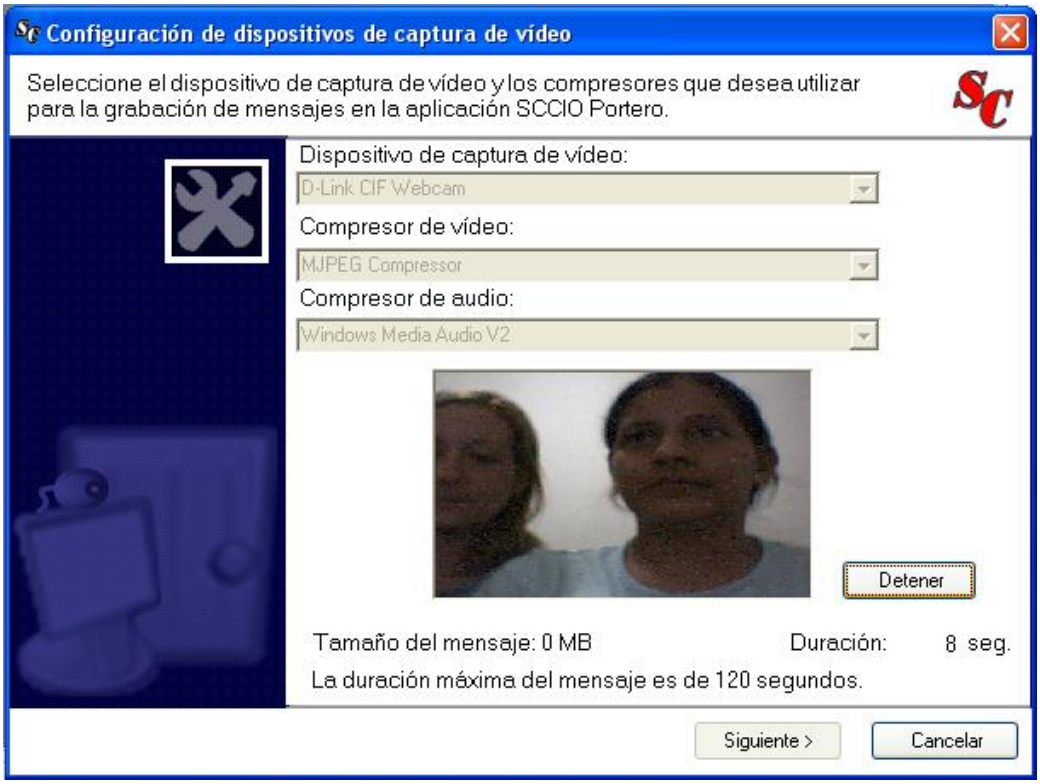


Figura 23.- Selección del dispositivo de captura de vídeo, y de los compresores que utilizará la aplicación en la grabación de mensajes.

Una vez que hayan seleccionado los dispositivos y el compresor que desea utilizar, puede grabar un mensaje de prueba presionando el botón *Grabar*. Esto le permitirá determinar si la combinación de compresores seleccionada logra reducir considerablemente el tamaño del archivo que se genera, o si es preferible escoger una diferente, que comprima los datos de manera más eficiente.

Con este objetivo, en la interfaz se incluyen dos etiquetas, que muestran tanto la duración como el tamaño final del mensaje de prueba que se haya generado y comprimido, utilizando el dispositivo de captura de vídeo y los compresores seleccionados, respectivamente. Una vez que haya encontrado la combinación que más le satisfaga, haga clic en el botón *Siguiente* para continuar.

Finalmente, aparecerá una ventana en la que se ha de especificar el dispositivo de vídeo que enfocará hacia la puerta y capturará las fotos instantáneas de los usuarios que accedan al área de oficinas. Puede probar el buen funcionamiento y el enfoque del dispositivo de captura seleccionado, haciendo clic en el botón *Tomar foto*.

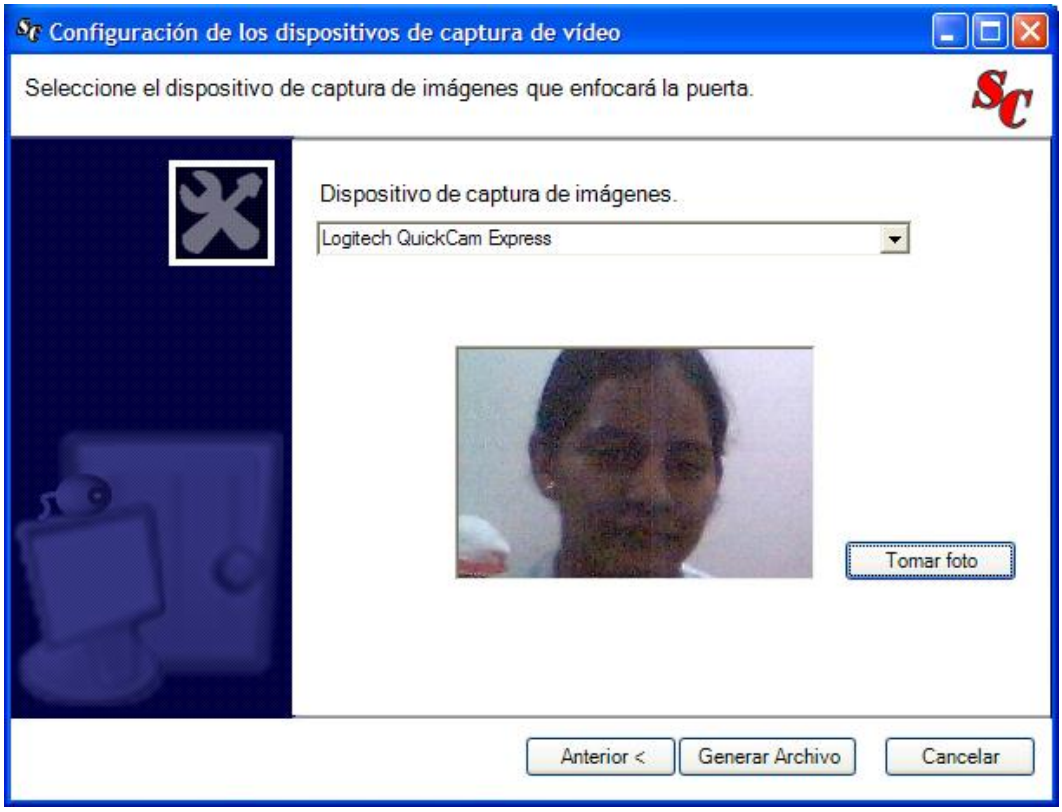


Figura 24.- Selección del dispositivo de captura de imágenes.

Una vez que haya indicado todos los datos anteriormente mencionados, presione el botón *Generar Archivo*, para mandar a guardar los datos de configuración que se acaban de especificar. De este modo, podrán ser recuperados más adelante, cuando sea preciso, durante la ejecución de SCCIO Portero.

BIBLIOGRAFÍA

- Microsoft Corporation, 2004, "ConferenceXP Research Platform", http://www.conferencexp.net/Community/Default.aspx?tabindex=12&tabid=27
- Microsoft Corporation, 2004, "ConferenceXP Architecture", http://www.conferencexp.net/Community/Default.aspx?tabindex=12&tabid=22
- Dr. Ze-Nian Li y Yingchen Yang, 2000, "Video Compression", http://www.cs.sfu.ca/CourseCentral/365/li/material/notes/Chap4/Chap4.3/Chap4.3.html#MPEG
- Fredy Cuello y Juan Rueda, 2000, "Compresión de vídeo en el estándar MPEG-1 (Aplicaciones Multimedia)", http://www.fuac.edu.co/autonoma/pregrado/ingenieria/ingelec/proyectosgrado/compresvideo/MPEG1.htm
- Fredy Cuello y Juan Rueda, 2000, "Compresión de vídeo en el estándar MPEG-2 (Aplicaciones Broadcast)", http://www.fuac.edu.co/

autonoma/pregrado/ingenieria/ingelec/proyectosgrado/compresvideo/MPEG2.htm

- Fredy Cuello y Juan Rueda, 2000, "Compresión de audio", http://www.fuac.edu.co/autonoma/pregrado/ingenieria/ingelec/proyectosgrado/compresvideo/compresion_audio.htm
- Anónimo, "CODECs y arquitecturas de vídeo", http://www.video-computer.com/codecs.htm
- Miguel Fernández, 1999, **"**Compresión de vídeo. Los CODECs de vídeo", http://ingenieria.udea.edu.co/~marthac/multimedia/codecvideo .html
- Anónimo, "Comparativa de CODECs de vídeo", http://www.mundo hardware.com/article/articleview/134/1/3/
- A. J. Millán, Septiembre del 2004, "Tratamiento digital del sonido", http://www.zator.com/Hardware/H10_2.htm
- Fredy Cuello y Juan Rueda, 2000, "Introducción a la compresión de vídeo", http://www.fuac.edu.co/autonoma/pregrado/ingenieria/ingelec/proyectosgrado/compresvideo/int_comp_video.htm
- Wave Report, Noviembre del 2004, **"**Video Compression Technology", http://www.wave-report.com/tutorials/VC.htm
- Rob Koenen, 2002, "Overview of the MPEG-4 standard", http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm#1.1
- Anónimo, "Vídeo sobre redes", http://www.monografias.com/trabajos10/vire/vire.shtml

- Juan Cattaneo, Agosto del 2002, "Multimedia", http://animania.com.ar/multimedia/
- Anónimo, 2004, "Transmisión de audio y vídeo por Internet", http://www.internetmultimedia.com.mx/bloques/transmision.htm
- Fernando Pazmiño, "Vídeo-conferencia", http://www.monografias.com/trabajos/videoconferencia/videoconferencia.shtml
- Anónimo, "Acerca de los formatos", http://www.labondiola.com/records/formatos/
- Lic. Virgilio Victoria, "Formatos utilizados en la tecnología multimedia", http://www.utp.ac.pa/seccion/topicos/multimedia/formatos.html#video
- Anónimo, Diciembre del 2003, "Formatos de imágenes", http://www.elpomeloradiactivo.com/paginas/diseno/grafico/art7.html
- Sergio de los Santos, 2004, "Biometría: El adiós a las contraseñas", http://www.portalmundos.com/mundoinformatica/comunicaciones/biometria.htm
- Anónimo, 2004, "¿Cómo funciona la biometría?", http://mouse.latercera.cl/2004/rep/02/04/01.asp
- Grupo INSYS, "Soluciones biométricas", http://www.insys.com.mx/biometria/biometria.htm
- Grupo INSYS, "Lectores biométricos Recognition Systems", http://www.insys.com.mx/biometria/lectores.htm
- Anónimo, "Sistema de control de presencia y accesos", http://www.revenga.com/sistemas/virtualkey.htm

- AMTEL, 2002, "Proximity card readers and cards", http://www.amtel-security.com/Components/Cutsheets/proximity_card.pdf
- Siemon, "CCTV y vigilancia por vídeo sobre 10G IP", http://www.siemon.com/la/white_papers/SD-03-08-CCTV.asp
- "Sistemas CCTV", http://www.superinventos.com/kitscctv.htm
- Bruce Momjian, Marzo del 2000, "Preguntas de uso frecuente (FAQ) sobre PostgreSQL", http://es.tldp.org/Postgresql-es/web/navegable /faq/faq-es.htm#BM1_1
- Damond Walker, "Connecting to PostgreSQL from Windows Platforms", http://www.developer.com/open/article.php/10930_631 251_1
- Anónimo, Julio del 2002, "DirectShow.NET", http://www.code project.com /cs /media/directshownet.asp
- Brian Low, Marzo del 2003, "DirectX.Capture Class Library", http://www.codeproject.com/cs/media/directxcapture.asp
- Microsoft Corporation, Julio del 2002, "DirectShow: Core Media Technology in Windows XP Empowers You to Create Custom Audio/Video Processing Components", MSDN Magazine
- Microsoft Corporation, 2002, "DirectShow DV Filters", MSDN Library
- Microsoft Corporation, Julio del 2001, "Microsoft .NET Remoting: introducción técnica", MSDN Library
- Microsoft Corporation, Noviembre del 2002, "Generar una aplicación .NET Remoting básica", Manual del programador de .NET Framework

- Microsoft Corporation, Noviembre del 2002, "Generar un tipo de uso remoto", Manual del programador de .NET Framework
- Microsoft Corporation, Noviembre del 2002, "Generar una aplicación host", Manual del programador de .NET Framework
- Microsoft Corporation, Noviembre del 2002, "Generar una aplicación cliente", Manual del programador de .NET Framework
- Microsoft Corporation, Noviembre del 2002, "Compilar y ejecutar la aplicación básica", Manual del programador de .NET Framework
- Microsoft Corporation, Noviembre del 2002, "Configuración de objeto remoto, Manual del programador de .NET Framework
- Microsoft Corporation, Noviembre del 2002, "Realizar una escucha con sockets", Manual del programador de .NET Framework
- Microsoft Corporation, Noviembre del 2002, "Utilizar un socket de servidor asincrónico", Manual del programador de .NET Framework
- Microsoft Corporation, Noviembre del 2002, "Utilizar un socket de cliente asincrónico", Manual del programador de .NET Framework