

Grabador Continuo de Audio

María Eugenia Aguirre
Guillermo Martony

22 de julio de 2005

*Departamento de Telecomunicaciones
Facultad de Ingeniería
Universidad de la República
Tutor: Ing. Juan Pechiar*

*A nuestros seres queridos
por su apoyo y comprensión
Juan, Elena, Verónica, Rafael
Carmen y Sebastian*

ÍNDICE GENERAL

Parte I Presentación	1
1.. Introducción	2
1. Contexto y motivación	2
1.1. Introducción	2
1.2. Métodos de grabación de larga duración y sus problemas	2
1.3. Necesidades adicionales al registro del audio	2
2. Sistema propuesto	3
2.1. Funcionalidades	3
3. Ingeniería de software	3
2.. Documento de visión	5
1. Posicionamiento	5
2. Relevamiento de necesidades	5
3. Perspectiva general del producto	5
4. Descripción de los usuarios	6
5. Rangos de calidad	7
6. Requerimientos sobre la documentación	7
6.1. Manual del usuario	7
6.2. Guía de instalación y archivo Léame.	7
3.. Funcionalidades destacables	9
1. Multi-plataforma y estándar	9
2. Programación versátil	9
3. Robusto y Seguro	10
4. Administración remota	10
5. Recuperación del audio con software no propietario	11
4.. Descripción del GCA	12
1. Diseño básico	12
2. Funcionamiento del proceso grabador	12
3. Control de Grabación	13
4. Mezclador	13
5. Control de Procesos	13
6. Control de Fallas	14
7. Consola de Administración y Operación	15
8. Recuperación y Reproducción del audio almacenado	15

5.. Proceso de investigación y desarrollo	16
1. Adquisición sin pérdidas	16
2. Independencia e intercomunicación de procesos	17
3. Necesidades adicionales a la biblioteca STL	18
3.1. Sincronización de procesos	18
3.2. Persistencia	18
3.3. Otras necesidades	19
4. Interfaz gráfica	19
5. Recuperación y reproducción de audio	20
5.1. Reproducción en tiempo real	20
5.2. Solicitud del audio	20
6. Archivos externos ejecutables	21
7. Servicios y registro de sucesos	21
8. Instaladores	22
6.. Estudio del mercado	23
1. Mercado genérico de los grabadores	23
2. Grabadores de alta calidad de audio	24
3. Funciones principales	24
3.1. Detección de fallas	24
3.2. Programas de grabación	25
3.3. Formatos de audio	25
3.4. Administración y clientes	25
3.5. Licenciamiento	25
4. Matriz de comparación	25
7.. Mediciones y ensayos realizados	27
1. Mediciones del GCA	27
1.1. Entorno de ensayo	27
1.2. Resultados obtenidos	28
2. Ensayo de operación continua	29
2.1. Radio El Espectador	29
2.2. Mesa de ensayos	29
3. Costo del GCA	29
8.. Perspectiva de desarrollo posterior	30
1. Motivos para un desarrollo posterior	30
2. Mejoras en la administración	30
2.1. Reproducción en vivo del audio	30
2.2. Alarma por falta de audio	30
2.3. Forzar cambio de archivo	31
2.4. Eliminación automática de audio antiguo	31
2.5. Módulo SNMP para monitorización de estado	31
3. Funciones adicionales para radiodifusión	31
3.1. Detección de señal externa de encendido	31
3.2. Separación de audio útil/audio de tandas	31
3.3. Módulo para programas de automatización radial	32

4.	Ampliación a otros segmentos del mercado de grabadores	32
4.1.	Comando de grabación por presencia de audio	32
4.2.	Ajuste automático de volumen	32
4.3.	Módulo para detección de corriente de bucle	32
4.4.	Registro de abonado llamante y de tonos digitados	33
5.	Otras funciones	33
5.1.	Internacionalización	33
5.2.	Soporte para unidades múltiples de almacenamiento	33
Parte II Desarrollo		34
9..	Manual del proyecto	35
1.	Objetivos	35
2.	Proceso de desarrollo	35
2.1.	Principio	36
2.2.	Elaboración	36
2.3.	Construcción	37
2.4.	Transición	37
3.	Documentos de las distintas fases del desarrollo	37
10..	Especificación de los casos de uso	39
1.	Objetivos	39
2.	Actores	39
3.	Funcionalidades comunes a todos los casos de uso	40
3.1.	Registro de errores	40
3.2.	Acceso remoto	40
4.	Caso de uso Recuperar Audio	40
4.1.	Descripción breve	40
4.2.	Flujo de eventos	40
5.	Caso de uso Reproducir Audio	41
5.1.	Descripción breve	41
5.2.	Flujo de eventos	41
6.	Caso de uso Grabar	41
6.1.	Descripción breve	41
6.2.	Flujo de eventos	41
7.	Caso de uso Administrar el Cronograma	43
7.1.	Descripción breve	43
7.2.	Flujo de eventos	43
8.	Caso de uso Monitorizar Canal	44
8.1.	Descripción breve	44
8.2.	Flujo de eventos	44
9.	Caso de uso Administrar los Canales	44
9.1.	Descripción breve	44
9.2.	Flujo de eventos	44
10.	Caso de uso Monitorizar Estado del GCA	46
10.1.	Descripción breve	46

10.2. Flujo de eventos	46
11..Especificación suplementaria	47
1. Objetivos	47
2. Alcance	47
3. Usabilidad	47
3.1. Compatibilidad	47
3.2. Diseño	47
3.3. Ayuda al usuario	47
4. Confiabilidad	47
4.1. Disponibilidad	47
5. Mantenimiento	48
5.1. Nuevas versiones	48
6. Restricciones de diseño	48
6.1. Requerimientos de la plataforma	48
12..Lista de riesgos - principio	49
1. Objetivo	49
2. Lista de riesgos	49
13..Guía de estilo del código fuente	51
1. Objetivo	51
2. Referencias	51
3. Sangría	51
4. Ubicación y uso de las llaves	51
5. Nomenclatura de variables, funciones, clases, etc.	52
6. Funciones	52
7. Clases	52
8. Comentarios	53
9. Instrucciones	53
10. Orden de las secciones de un archivo	53
11. Nombres de archivos	54
12. Abreviaciones de palabras comunes	54
13. Prefijos de controles de la interfaz gráfica	54
14..Lista de riesgos - elaboración	56
1. Objetivo	56
2. Lista de riesgos	56
15..Documento de diseño	57
1. Tecnología y bibliotecas utilizadas	57
1.1. Lenguajes de programación utilizados	57
1.2. Uso de CORBA	57
1.3. Bibliotecas utilizadas	57
2. Diseño	58
2.1. Perspectiva general del sistema	58
2.2. Biblioteca de clases comunes	61

2.3.	Grabador	64
2.4.	Control del Grabación y Cronograma	68
2.5.	Mezclador	70
2.6.	Control de procesos	71
2.7.	Control de fallas	72
2.8.	Consola de operación y monitorización	74
16..	Descripción de interfaces CORBA	76
1.	Control de procesos	76
1.1.	Propósito	76
1.2.	Métodos	76
1.3.	Tipos de datos	77
1.4.	Archivo de descripción de la interfaz (IDL)	77
2.	Control de Grabación	78
2.1.	Propósito	78
2.2.	Métodos	78
2.3.	Archivo de descripción de la interfaz (IDL)	78
3.	Grabador	79
3.1.	Propósito	79
3.2.	Métodos	79
3.3.	Tipos de datos	80
3.4.	Archivo de descripción de la interfaz (IDL)	80
4.	Cronograma	80
4.1.	Propósito	80
4.2.	Métodos	80
4.3.	Tipos de datos	81
4.4.	Archivo de descripción de la interfaz (IDL)	82
5.	Mezclador	83
5.1.	Propósito	83
5.2.	Métodos	83
5.3.	Tipos de datos	84
5.4.	Archivo de descripción de la interfaz (IDL)	84
6.	Tipos de datos compartidos	84
6.1.	Propósito	84
6.2.	Tipos de datos	84
6.3.	Archivo de descripción de la interfaz (IDL)	85
Parte III	Apéndices	86
17..	Herramientas del entorno de desarrollo	87
1.	Diseño	87
1.1.	Rational Rose	87
2.	Codificación	87
2.1.	KDevelop	87
2.2.	Microsoft Visual Studio	87
2.3.	Subversion y TortoiseSVN	87

2.4.	Meld y WinMerge	88
3.	Documentación	88
3.1.	Lyx	88
3.2.	Doxygen	88
18..	Relevamiento y elección de codificadores de audio	89
1.	Motivación	89
2.	Atributos de los códecs de audio	89
3.	Clasificación de códecs de audio	90
4.	Codificación perceptual y otras técnicas de reducción de datos	93
5.	Códec elegido	93
19..	Manual del GCA	95
1.	Capacidades del producto	95
2.	Requisitos mínimos del sistema	96
2.1.	Sistema principal GCA	96
2.2.	Control de Fallas del GCA	96
2.3.	Consola del GCA	96
3.	Instrucciones de instalación	97
3.1.	Instalación en Linux	97
3.2.	Instalación en Microsoft Windows	98
3.3.	Programas adicionales	102
4.	Puesta en marcha	104
4.1.	Configuración del servicio de nombres CORBA (nsd)	104
4.2.	Configuración inicial de archivos del sistema GCA	104
4.3.	Inicio de servicios	105
4.4.	Configuración inicial de canales	105
4.5.	Configuración para la recuperación del audio	105
5.	Manual de la Consola	107
5.1.	Funciones	107
5.2.	Inicio de la Consola	107
5.3.	Selección de GCA	108
5.4.	Configuración de canales	111
5.5.	Administración de canales	117
5.6.	Monitorización de grabación y operación manual de un canal	117
5.7.	Administración del cronograma de un canal	119
6.	Manual del Control de Fallas	125
6.1.	Descripción del funcionamiento	125
6.2.	Descripción del archivo de configuración	125
7.	Interfaz Web para la búsqueda del audio	128
20..	Descripción de archivos del sistema	129
1.	Archivo de configuraciones del GCA	129
1.1.	Propósito	129
1.2.	Claves	129
1.3.	Archivo de definición (gca_conf.xml)	131
2.	Cronograma de un canal	131

2.1.	Propósito	131
2.2.	Claves de un elemento de cronograma	132
2.3.	Claves de un elemento formato_audio	132
2.4.	Claves de un elemento fechas	133
2.5.	Archivo de definición (canal1.xml)	133
3.	Archivo de configuraciones del Control de Fallas	134
3.1.	Propósito	134
3.2.	Claves	134
3.3.	Archivo de definición (gcactrlfallas_conf.xml)	135
	Bibliografía	137
	. Bibliografía	138

Parte I

PRESENTACIÓN

1. Introducción

1. Contexto y motivación

1.1. Introducción

Las emisoras de radiodifusión deben registrar su programación durante varios días de acuerdo a las reglamentaciones impuestas por la Unidad Reguladora (URSEC). A menudo tienen además la necesidad de editar grabaciones de la programación previamente emitida para producir otros programas pre-grabados (por ejemplo: recopilaciones para el programa del fin de semana).

1.2. Métodos de grabación de larga duración y sus problemas

El método clásico utilizado es el registro en cinta magnética de $\frac{1}{4}$ de pulgada a baja velocidad (1.875 in/s) en grabadores de carrete abierto. Este método es engorroso principalmente en dos aspectos. En primer lugar es necesaria la supervisión de un operador que debe estar atento al cambio de carrete, elección de pista y ajuste del nivel de grabación. En segundo lugar, la reproducción de una selección de audio implica buscar entre las múltiples cintas y luego, eventualmente, copiar el contenido en otro medio magnético, lo que requiere un tiempo considerable.

Adicionalmente las máquinas grabadoras son costosas y necesitan un mantenimiento constante de las cabezas lectoras y mecanismos. Los repuestos son exclusivos y mantenerlos en depósito agrega un costo más. Los carretes de cinta son caros, de duración limitada y actualmente difíciles de encontrar en plaza.

Por último, la calidad de audio obtenida no es satisfactoria, por lo cual se hace necesario grabar paralelamente en otro formato (casete), en caso de que se requiera una edición posterior para producir otro programa.

Otros dispositivos utilizados para realizar grabaciones de audio de larga duración son los grabadores de vídeo VHS o los grabadores DAT. Ambos sistemas mejoran la calidad del audio pero mantienen la necesidad de supervisión.

Una mejora sustancial frente a los sistemas mencionados antes son los grabadores continuos digitales de cinta DDS-2 o DDS-3 (Digital Data Storage) y disco rígido. Estos aparatos permiten almacenar una gran cantidad de horas y varios canales, así como reproducir desde la cinta mientras continúa la grabación en el disco duro. Sin embargo su costo es varias veces superior al del sistema propuesto y no dispone de algunas funciones que se exponen a continuación.

1.3. Necesidades adicionales al registro del audio

Hoy en día es cada vez más frecuente y útil la edición digital por computadora utilizando aplicaciones diseñadas para tal fin. En caso de que se desee realizar un programa a partir de audio almacenado en otro medio que no sea un archivo (por ej.: casete de audio compacto, DAT, etc), es necesario

transferir el audio a la computadora y realizar la composición. Luego, si no se dispone de un sistema que permita reproducir el archivo producido directamente al aire, hay que transferir a otro medio nuevamente. El tiempo requerido en las dos transferencias hace que muchas veces sea preferible editar utilizando los métodos tradicionales, salvo que se desee realizar ediciones más finas (quitar ruidos, aplicar filtros, etc.). Adicionalmente en el proceso mencionado hay varias conversiones A/D, D/A que deterioran el audio agregando ruido.

Las emisoras de radio modernas utilizan computadoras y software específico para la asistencia al aire del operador para emitir las pautas publicitarias y demás archivos de audio, ubicados en una biblioteca de archivos de audio o “audioteca”. Las discotecas musicales tienden a transferirse completamente a la audioteca, eliminándose la utilización de discos compactos. El propio disco compacto tiende a dejarse de usar y en su lugar la música, cuñas publicitarias y otros tipos de contenido de audio provenientes de terceros son transferidos directamente a través de redes de datos. La audioteca es utilizada internamente para la producción de los programas y parte de ella publicada en la Internet. La edición de audio por computadora utilizando aplicaciones diseñadas para tal fin ha desplazado casi completamente los demás métodos de edición.

En este ambiente totalmente digital, se hace prácticamente indispensable poder disponer a través de la red, de un acceso aleatorio al audio almacenado digitalmente y en archivo en formato estándar utilizado por las aplicaciones de edición mencionadas antes.

2. Sistema propuesto

Se propone implementar un sistema utilizando una computadora personal (PC) con dispositivo de audio estándar, que tenga las siguientes funcionalidades:

2.1. Funcionalidades

1. Adquisición y almacenamiento continuo de un canal estereofónico de audio comprimido en disco rígido. Con la posibilidad de selección de su formato de compresión de acuerdo a un cronograma preestablecido, de modo de grabar con mayor o menor calidad según la forma en que se planee utilizar la grabación a posteriori.
2. Seguridad de que la grabación se está realizando correctamente, utilizando un sistema de verificación y alarma.
3. Recuperación del audio almacenado en formato digital o analógico. Esto se realizará mediante una aplicación servidor que recibirá solicitudes de clientes que indicarán la hora y fecha de comienzo y opcionalmente la de finalización.

La recuperación en formato digital estándar se realizará a través de una aplicación cliente que podrá funcionar a través de una red local o en la misma PC grabadora y se implementará una interfaz de acceso a través de un explorador web.

3. Ingeniería de software

Uno de los objetivos principales de este desarrollo es aplicar procesos de ingeniería de software orientados a objetos, junto con herramientas que los apoyen, como por ejemplo: herramientas de mo-

delado y visualización de la arquitectura, control de versiones, documentación, gestión de proyectos, entorno de desarrollo integrado, etc.

Al utilizar estos procesos establecidos se busca: aprender los procesos y las herramientas, asegurar la calidad del software y lograr un sistema que cumpla con los requerimientos.

Durante el proyecto se busca utilizar las técnicas establecidas, pero sin ceñirse estrictamente, sino utilizando un criterio de practicidad y utilidad, de modo de poner las técnicas al servicio del proyecto y no viceversa.

La metodología de desarrollo de software orientado a objetos empleada utiliza el lenguaje de modelado UML (Unified Modeling Language) y fue derivada de los trabajos de Grady Booch [1], Ivar Jacobson [2] y James Rumbaugh [3]

2. Documento de visión

Se definen en esta sección los requerimientos de alto nivel del sistema de Grabador Continuo de Audio (GCA) en términos de las necesidades de los usuarios finales.

1. Posicionamiento

Existen numerosas aplicaciones para un grabador continuo de audio, algunas de las cuales son:

- Grabación de conversaciones telefónicas. Por ejemplo en servicios de atención de emergencias, servicios financieros, servicios de atención a clientes, etc.
- Grabación de emisiones de difusión de radio o televisión.
- Grabación del audio proveniente de sistemas de alarma.
- Grabación de sesiones parlamentarias o juicios orales, des-grabación y asistencia a taquígrafos.

El presente proyecto se basa en el caso de una emisora de radiodifusión. Las demás aplicaciones serán tenidas en cuenta al definir la arquitectura, de modo que el desarrollo pueda eventualmente ser reutilizado para ellas.

2. Relevamiento de necesidades

Para el caso de una emisora de radio, se identifican las siguientes necesidades que requieren un grabador continuo de audio:

1. De acuerdo a las reglamentaciones impuestas por la Unidad Reguladora de Servicios de Comunicaciones (URSEC), las emisoras de radiodifusión deben registrar su programación durante 10 días.
2. Reutilización posterior con los objetivos de:
 - Edición, clasificación y archivo en una biblioteca de audio.
 - Publicación en Internet.
 - Reemisión en programas de recopilación.

3. Perspectiva general del producto

El producto a desarrollar es un software que consta de un conjunto de módulos servidores y clientes que se detallan a continuación:

Grabador

El módulo grabador se encarga de adquirir un canal de audio analógico o digital y almacenarlo en un medio de almacenamiento masivo de acceso aleatorio, normalmente un disco rígido. Un cronograma determina cuándo y con qué formato de audio debe realizarse la grabación. El sistema permite ejecutar un proceso al finalizar el almacenamiento de cada archivo de audio PCM. Este proceso puede comprimir el audio, copiarlo a través de la red, o realizar cualquier tarea automatizable que se requiera. El sistema está preparado para administrar varios grabadores y por lo tanto permitir la adquisición de múltiples fuentes de audio.

Servidor de audio

Este módulo es el responsable de atender las solicitudes de acceso al audio almacenado provenientes de los distintos clientes.

Consola de operación y monitorización

Esta consola se utiliza para configurar y supervisar el GCA.

Cliente con interfaz web

Esta interfaz permite acceder al audio almacenado desde un explorador de Internet estándar y reproducirlo. Se reproduce el audio que fue almacenado posteriormente a una fecha y hora determinadas por el usuario. También permite recuperar los archivos de audio utilizando hipervínculos.

Detección de fallas

El sistema utiliza este módulo para detectar faltas en el correcto funcionamiento del grabador. En caso que se detecten fallas pueden ejecutarse acciones programables. El módulo de detección de fallas funciona a través de la red lo que permite situarlo en una computadora independiente.

4. Descripción de los usuarios

Editor de audio

El editor de audio selecciona una parte del audio almacenado en el GCA para escucharlo o recuperarlo en forma de archivo de audio para su posterior edición.

Usuario de Internet

El usuario de Internet utiliza un explorador de Internet como interfaz para seleccionar el momento a partir del cual desea escuchar el audio almacenado en el GCA.

Operador

El operador es el responsable de:

- Controlar que el GCA funcione correctamente, a) verificando que grabe cuando lo debe hacer y que lo haga con un volumen adecuado y b) analizando los registros de funcionamiento.
- Especificar el cronograma que determina en qué momento y con cuál formato se debe grabar el audio.
- Habilitar o deshabilitar los grabadores del GCA.

Técnico

El técnico se encarga de instalar y configurar el sistema de modo que quede listo para ser operado por los demás usuarios. Luego es responsable de dar soporte de primer nivel a los mismos.

5. Rangos de calidad

Disponibilidad El GCA debe poder grabar durante las 24 horas del día, todos los días.

Confiabilidad En caso que se detecten condiciones de mal funcionamiento, debe registrarse y activarse un indicador de alarma.

6. Requerimientos sobre la documentación

6.1. Manual del usuario

El manual del operador debe describir el uso del sistema desde los puntos de vista del operador.

- Requisitos mínimos del sistema.
- Capacidades del producto.
- Descripción de la interfaz de usuario.
- Ejemplos de operación del sistema.
- Información de los contactos de soporte.

6.2. Guía de instalación y archivo Léame.

La guía de instalación debe incluir:

- Requerimientos mínimos del sistema.
- Instrucciones de instalación.
- Información de los contactos de soporte.

El archivo Léame debe incluir:

- Nuevas características de la versión.
- Defectos conocidos y formas de evitarlos.

3. Funcionalidades destacables

A continuación se describen las funcionalidades más destacables del GCA.

1. Multi-plataforma y estándar

- Todos los módulos del sistema pueden operar tanto en Linux como en Microsoft Windows.
- El sistema principal y el sistema de control de fallas operan como servicios, lo que permite que los procesos se ejecuten independientemente de la sesión de usuario activa. Además, no necesitan de interfaz gráfica lo que minimiza los recursos requeridos para su ejecución.
- El GCA adquiere múltiples canales de audio utilizando una o varias tarjetas de audio estándar conectadas a la PC.
- El sistema está diseñado de forma distribuida donde las aplicaciones de supervisión, administración y recuperación son independientes del servidor que realiza las grabaciones.
- El servidor utiliza protocolos de comunicación públicos, lo que permite que el cliente pueda desarrollar sus propias aplicaciones que interactúen con el servidor, escogiendo el sistema operativo y el lenguaje que prefiera.

2. Programación versátil

- Cada canal tiene su propio cronograma que indica cuándo grabar y con qué calidad adquirir el audio. Esto permite reducir el espacio en disco adaptándose a las necesidades del usuario.
- La programación del cronograma es versátil permitiendo un formato similar la tabla del proceso “cron” de Unix. Las entradas del cronograma permiten entonces una programación por fechas fijas, por días de la semana o por eventos de repetición (por ejemplo: todos los primeros de mes de 20:00 a 21:00, o los primeros 10 minutos de cada hora, etc.)
- Los parámetros que definen la calidad de audio son configurables por el usuario de acuerdo a las posibilidades de la tarjeta de audio que disponga.
- El audio se graba en disco en archivos de duración programable por canal y en formato RIFF-PCM.
- El audio se graba sin pérdida de muestras de uno a otro archivo gracias a un cuidado diseño basado en colas.

- Luego que cada archivo se guarda en disco, se puede lanzar un proceso externo que permite actuar sobre el archivo guardado.
- El proceso externo post-grabación es totalmente personalizable por el usuario permitiéndole por ejemplo comprimir el audio, copiarlo a través de la red, etc.
- El proceso externo es asignado por canal por lo que cada canal puede tener un proceso diferente.

3. Robusto y Seguro

- El proceso de grabación de cada canal es independiente de los demás y del proceso central de control, lo que torna al sistema robusto ante fallas de algún módulo.
- El GCA dispone de un módulo independiente de control de fallas que se comunica con el proceso central del GCA para verificar que no existan fallas de funcionamiento.
- El control de fallas puede funcionar en otra PC y por lo tanto totalmente independiente del sistema central. Esto permite detectar problemas generales del servidor central o de su entorno.
- Al detectar una falla se pueden disparar procesos en base única y/o recurrente. Al igual que los procesos de post-grabación, son totalmente personalizables. Esto permite gran versatilidad en el manejo de fallas que puede ir desde el envío de un correo al operador de guardia hasta el inicio automático de un servidor GCA de respaldo.

4. Administración remota

- El GCA dispone de un módulo independiente que es la consola de administración.
- La consola de administración también funciona a través de la red por lo que no requiere la operación de una interfaz gráfica sobre el sistema en que se ejecuta el GCA.
- La consola permite configurar los canales, iniciarlos, detenerlos y monitorizarlos.
- Para cada canal el cliente puede definir la duración de los archivos de audio, el directorio donde guardarlos, el identificador del dispositivo de audio y su mezclador, el mínimo espacio en disco admisible, y los tamaños máximos admisibles en las colas de proceso. Estos últimos parámetros permiten el ajuste fino del sistema de control de fallas.
- Para cada canal, la consola permite monitorizar el nivel de grabación. El monitor muestra el estado del grabador, los niveles de grabación en tiempo real y un indicador de saturación.
- Para cada canal, la consola permite ajustar el nivel de grabación.
- Para cada canal, la consola permite ajustar el cronograma de grabación con las calidades de audio deseadas.
- El operador de la consola puede forzar de forma manual e independientemente de la programación del cronograma, una detención o inicio de grabación con un formato escogido.

5. Recuperación del audio con software no propietario

- La recuperación y reproducción del audio puede realizarse a través de la red compartiendo simplemente el directorio donde está almacenado.
- Con un navegador web estándar, el usuario puede buscar el audio indicando el canal, fecha y hora. Se devuelven hipervínculos para recuperar los archivos de audio.
- A partir de la consulta efectuada utilizando el navegador, la reproducción se realiza con un reproductor que soporte el lenguaje SMIL, como por ejemplo RealPlayer.
- Cualquier sistema que disponga de un navegador web estándar y opcionalmente un reproductor de audio (tipo RealPlayer) puede actuar como cliente del GCA sin requerir otros programas adicionales.

4. Descripción del GCA

1. Diseño básico

El GCA es un software codificado en C++ utilizando la biblioteca estándar y bibliotecas multi-plataforma públicas desarrolladas por terceros. El código que es específico a la interfaz con el sistema operativo (como la administración de procesos y la comunicación con el hardware) se codificó con directivas de compilación. Esta técnica permite mantener la funcionalidad multi-plataforma para los sistemas operativos Linux y Microsoft Windows.

El principal objetivo es que la grabación se realice con la menor probabilidad de fallas y asegurando la adquisición continua del audio (sin pérdida de muestras). Por este motivo el diseño consiste en módulos que se manifiestan como procesos independientes y que se comunican entre ellos con el protocolo de comunicación CORBA. Esto permite que el grabador de cada canal sea un proceso independiente de los demás grabadores u otros procesos de control. Si un proceso falla, por ejemplo por un problema en una tarjeta adquisidora, esto no causa la falla del proceso de grabación de otro canal. La comunicación CORBA permite que los procesos de control que se ocupan de la gestión del cronograma o la supervisión del buen funcionamiento del sistema puedan igualmente comunicarse los unos con los otros y con los procesos grabadores. El CORBA es un protocolo que permite la comunicación de procesos incluso en un entorno de red lo que facilita la realización de clientes de red para administrar el GCA.

2. Funcionamiento del proceso grabador

Para asegurar que la adquisición se realice sin pérdida de muestras se utiliza un sistema de colas de adquisición y procesamiento. Para esto se escogió una arquitectura multi-thread lo que permite que exista dentro del proceso grabador varios hilos de ejecución, cada cual encargado de una etapa de la tarea de adquisición y almacenamiento. El diseño utiliza dos colas de muestras y tres bucles de ejecución.

El primer bucle se encarga de adquirir las muestras de audio del dispositivo de audio y colocarlas en la primera cola. Al iniciar la grabación el primer bucle inicializa el dispositivo de audio de acuerdo a la calidad de adquisición que se requiera.

El segundo bucle puede realizar medidas en caso que se solicite por ejemplo para monitorizar los niveles del audio adquirido. El segundo bucle toma los búfer de la primera cola, los procesa y los coloca en la segunda cola.

El tercer bucle realiza la grabación en disco del audio tomando los búfer de la segunda cola. Si algún bucle se retrasa realizando alguna actividad las colas se incrementan. Esta elasticidad es lo que permite que no se pierdan muestras cuando se cambia de archivo en el tercer bucle. La grabación a disco se realiza en archivos de audio en formato WAV que tienen un tamaño dado configurable por canal. El tercer bucle graba en disco hasta completar el tamaño configurado y luego que el archivo se

ha guardado en disco, lanza un proceso independiente. Este proceso toma como parámetro el nombre del archivo que se ha guardado. Este archivo ejecutable es configurable por el operador del sistema y cada canal puede tener asignado uno diferente o ninguno (en este caso no se lanza un proceso al finalizar la grabación del archivo WAV). Como el proceso se lanza independiente al proceso grabador, el tercer bucle prosigue su funcionamiento abriendo un nuevo archivo y reanudando la grabación de las muestras. El archivo de audio puede cerrarse antes de completar el tamaño establecido en caso que la adquisición deba detenerse.

Si la grabación debe detenerse el primer bucle detiene la adquisición, el segundo bucle termina de procesar todos los búfer de la primera cola y finalmente el tercer bucle completa la grabación a disco de las muestras pendientes y si corresponde lanza el proceso externo.

El proceso grabador descrito puede entonces estar en uno de dos estados: detenido o adquiriendo y grabando a disco el audio. El estado del grabador está dado por la instrucción que se envíe utilizando el protocolo CORBA.

3. Control de Grabación

Para cada canal existe, asociado al proceso grabador, un proceso de control de grabación. El proceso de control de grabación tiene acceso al cronograma del canal que le corresponde. Este proceso es un bucle que trabaja por eventos programados, el bucle comienza consultando el estado en que debe estar el grabador de acuerdo con el cronograma e instruye al grabador usando CORBA, luego consulta en el cronograma cuál sería la fecha del próximo cambio de estado del grabador y programa un evento para ese momento. En este punto el bucle queda detenido a la espera del evento que se programó y se mantiene recibiendo eventos externos.

Los eventos externos se reciben por el protocolo CORBA. Un evento externo se recibe ante un cambio realizado al cronograma, en este caso el control de grabación consulta en el cronograma si el evento programado debe modificarse. Otro evento externo es la instrucción explícita de un comando de grabación o detención (independiente del cronograma). La consola de administración y operación permite al operador forzar un cambio de estado en el grabador lo que se realiza enviando la instrucción por CORBA al control de grabación del canal correspondiente. En este caso, el control de grabación instruye al grabador y se mantiene esperando el evento ya programado.

4. Mezclador

Para cada canal, existe un proceso de manejo del mezclador de audio. El proceso mezclador tiene acceso al dispositivo mezclador del canal y permite mediante su interfaz CORBA, modificar el nivel de adquisición del audio y obtener el nivel definido para el canal. Esto permite regular la sensibilidad de adquisición del audio.

5. Control de Procesos

De acuerdo a lo explicado antes, cada canal que esté operativo produce tres procesos independientes comunicados por CORBA. Para gestionar el conjunto de canales activos, se requiere de un tercer proceso que también es independiente de los otros. Este proceso lleva el nombre de Control de

Procesos porque es el que inicializa los procesos de cada canal y supervisa que estén operando sin fallas.

El Control de Procesos inicialmente obtiene la información de todos los canales configurados a partir de un archivo de configuración. Para cada canal que debe estar operativo lanza los procesos de Control de Grabación y Grabador. Luego permanece en un bucle que consulta periódicamente a cada proceso si está funcionando correctamente. Esta consulta se realiza por CORBA. En caso que algún proceso no responda a la consulta o responda indicando que tiene una falla, se registra la falla ocurrida en el registro de sucesos del sistema.

El Control de Procesos también presenta una interfaz CORBA que puede ser accedida por otros módulos de administración, operación y supervisión.

Es posible instruir por CORBA al Control de Procesos para que inicie o detenga el funcionamiento de un canal, agregue nuevos canales, modifique o elimine canales existentes. El Control de Procesos implementa lo necesario para detener adecuadamente los procesos de un canal dado o iniciarlos según sea requerido. Los cambios en las configuraciones de los canales son reflejados por el Control de Procesos en el archivo de configuración del sistema que se guarda en el disco luego de cada cambio.

También se puede consultar por CORBA al Control de Procesos si el sistema está funcionando sin fallas u obtener el detalle de las fallas activas si existen.

El Control de Procesos se registra en el sistema operativo como un servicio el cual puede iniciarse o detenerse con el método usual de administración de servicios. Esto permite que el GCA pueda iniciarse automáticamente al iniciar el sistema operativo. Como las interfaces de usuario son procesos independientes, el Control de Procesos y los procesos de cada canal pueden ejecutarse en ambiente no gráfico lo que reduce los requerimientos de recursos del sistema.

6. Control de Fallas

La gestión del buen funcionamiento del GCA se realiza con un proceso independiente. El Control de Fallas es un bucle de ejecución que periódicamente consulta por CORBA al Control de Procesos el estado del GCA. En caso que existan fallas, el detalle de las mismas se registra en un archivo en disco y se lanza un proceso independiente que consiste en un archivo ejecutable definido por el usuario. Este proceso se lanza cada vez que la consulta de estado señalice una falla. Esto significa que puede considerarse una proceso recurrente de periodicidad igual al bucle de ejecución del Control de Fallas. Además, cuando se detecta la primera falla (o dentro de un estado de falla se observa una falla nueva que no había sido reportada antes), se lanza otro proceso externo diferente al anterior. Ante la ocurrencia de una única falla que se mantiene al cabo de varias consultas, se lanzará entonces un proceso externo singular y luego varias veces el proceso recurrente (para todas las siguientes consultas que reporten la misma falla).

Si el Control de Fallas no puede comunicarse por CORBA con el Control de Procesos, lanza un proceso diferente a los anteriores por única vez. Luego, si la falta de comunicación permanece en las próximas consultas, se lanza un cuarto proceso de forma recurrente (para cada consulta que no sea respondida).

Al restablecerse el correcto funcionamiento del GCA luego de una detección de falla o pérdida de comunicación, el Control de Fallas lanza una vez un proceso para señalar la condición de buen funcionamiento.

Los cinco procesos involucrados en la dinámica del control de fallas son configurables por el usuario y se establecen en un archivo de configuraciones junto con la periodicidad del bucle de consulta.

Como la comunicación CORBA funciona a través de la red, lo conveniente es ubicar el Control de Fallas en una computadora independiente, en un lugar separado, para poder detectar fehacientemente fallas generales del sistema y entorno en que se encuentra el GCA. Dentro de los procesos programables pueden configurarse, según el requerimiento del cliente, alarmas sonoras, mensajes de correo y tareas automáticas de contingencia como por ejemplo iniciar una sistema GCA de respaldo.

El Control de Fallas también se registra en el sistema operativo como un servicio y al igual que los procesos descritos antes, tampoco requiere de soporte gráfico del sistema operativo.

7. Consola de Administración y Operación

La consola de administración y operación es gráfica y también opera en Linux y en Microsoft Windows a través de una conexión de red. Como biblioteca gráfica se escogió WxWidgets por ser multi-plataforma y compatible con un conjunto extenso de compiladores.

La consola se conecta con los distintos procesos según la tarea que se requiera.

La consola permite consultar el estado de funcionamiento del sistema al Control de Procesos y obtener la lista de errores activos si hay fallas. También permite agregar nuevos canales, modificar canales existentes, eliminarlos, iniciar y detener la operación de los mismos. Para estas tareas se comunica con el Control de Procesos.

Al escoger un canal, la consola permite monitorizar los niveles de grabación para lo que se comunica específicamente con el proceso Grabador del canal escogido. Además permite el ajuste del nivel de adquisición del canal de audio para lo cual se comunica con el proceso Mezclador del canal escogido. También permite la edición del cronograma comunicándose con la interfaz CORBA del Cronograma del canal escogido.

8. Recuperación y Reproducción del audio almacenado

La recuperación o reproducción del audio se realiza con base en un servidor de páginas web y un servidor de streaming. Como servidor de páginas web se propone un Apache[4] y como servidor de streaming un Helix[5] aunque pueden utilizarse otras aplicaciones según la preferencia del cliente. El servidor de streaming debe configurarse para que pueda acceder a los archivos de audio de cada canal.

La comunicación se establece realizando una consulta por canal y fecha usando un navegador estándar. La página ejecuta un script PHP que consulta en el disco si existen archivos de audio para los criterios indicados en la consulta. Si existen archivos se genera una página dinámica con los hipervínculos de los archivos. Para la reproducción se genera un archivo de definiciones SMIL[6] que es interpretada por el reproductor RTSP (RFC 2326) [7] que solicita el audio al servidor de streaming.

5. Proceso de investigación y desarrollo

Este capítulo describe el proceso de investigación efectuado para la resolución de los temas relevantes del proyecto.

1. Adquisición sin pérdidas

Como primera aproximación a la cuestión de la adquisición de audio se realizaron varios grabadores simplificados con el propósito de tomar familiaridad con la interfaz de programación de los dispositivos de audio y mezcladores. En este proceso se observó que se producían pérdidas de muestras por falta de procesamiento o demoras en los accesos a disco. Especialmente se observó este problema al iniciar en simultáneo otras aplicaciones o durante el cambio de archivo de audio (al cerrar un archivo y abrir uno nuevo).

Para solucionar el problema de pérdida de muestras se propuso y ensayó un sistema con dos hilos de ejecución (threads) que compartían una cola de búfers de memoria donde se guardaban temporalmente las muestras de audio. Un hilo se encargaba de adquirir las muestras del dispositivo de audio y colocarlas en la cola, mientras que el otro hilo se encargaba de tomar las muestras de la cola y grabarlas al disco. El resultado del ensayo resultó muy satisfactorio por lo que se desarrolló y perfeccionó el sistema con base en este diseño.

La división en hilos de ejecución se realizó con el propósito de separar e independizar las diferentes tareas que componen la adquisición y grabación del audio. Este método permite un mejor uso del tiempo del procesador ya que la asignación de recursos la realiza el despachador de tareas del sistema operativo, que es un sistema eficiente y confiable. Además, el despachador realiza el manejo necesario para brindar más atención a las tareas que se asignen con mayor prioridad, lo que en el caso del GCA, permite aumentar la prioridad al hilo de adquisición reduciendo las probabilidades de pérdida de muestras.

El diseño final está compuesto por tres hilos de ejecución ya que se identificaron tres tareas de importancia: la adquisición de muestras de audio, el procesamiento intermedio y el almacenamiento en disco de las muestras. El proceso intermedio se utiliza actualmente para realizar cálculos de nivel con fines de supervisión. Se utilizan dos colas de intercambio de muestras o búfers. La primera es llamada cola de adquisición, donde el hilo de adquisición coloca las muestras tomadas del dispositivo de audio y el hilo de procesamiento intermedio las retira para realizar los cálculos necesarios. Luego de procesado, cada búfer o muestra es colocado en la segunda cola, la cola de grabación, de donde el hilo de almacenamiento las toma para generar los archivos de audio en disco.

Para mejorar el manejo de solicitud y liberación de espacio de memoria, el diseño utiliza una tercera cola a modo de repositorio. Cuando un espacio de memoria correspondiente a una muestra de la cola de grabación se libera, se asocia al repositorio y, cuando el hilo de adquisición requiere insertar una nueva muestra en la cola de adquisición, toma la memoria ya existente en la cola de reposición. Al iniciar, el sistema asigna espacios de memoria para el repositorio y las dos colas. El repositorio tiene

un margen de crecimiento lo que permite que el adquisidor siga operando normalmente aunque algún hilo de ejecución se retrase. En estos casos las colas crecen hasta que se recupere la operación normal del sistema. En condiciones normales, las colas se mantienen estables ya que los hilos de proceso y almacenamiento son capaces de realizar sus tareas en un tiempo menor que la adquisición del audio. Con el diseño de colas, la cantidad y lugar de memoria utilizada por el grabador es normalmente constante.

Por su importancia y vinculación a la capacidad de procesamiento, el largo de las colas es un indicador que se utiliza para supervisar el correcto funcionamiento del sistema.

2. Independencia e intercomunicación de procesos

Para reducir las probabilidades que la grabación del audio no se realice a causa de fallas, se buscó un diseño en el cual el módulo elemental de grabación fuese lo más independiente posible del resto del sistema. El módulo básico de grabación para un canal de audio es el proceso de tres hilos descrito en 1. Existen otros módulos que son necesarios para el completo funcionamiento del sistema como aquellos que se encargan de gestionar el conjunto de canales que deben grabar audio o los que manejan la lógica de cuando debe realizarse la adquisición de cada canal. Se propuso un diseño basado en procesos independientes para cada tarea, de este modo la falla de un proceso no genera la falla de los otros. Por ejemplo, ante una falla de acceso al hardware o cualquier falla del sistema que resulte en la terminación de un proceso de un canal, las tareas de otro canal continúan ejecutándose normalmente.

Existen por canal tres procesos: el módulo básico de grabación, un proceso que controla la grabación (es decir cuándo hay que adquirir el audio y con qué formato) y un proceso que maneja el mezclador asociado al canal. Luego existe un proceso central que se encarga de lanzar, terminar y supervisar los procesos de cada canal y un proceso que controla el correcto funcionamiento del sistema.

Para un diseño como el planteado es de máxima importancia la forma de comunicación entre procesos. El sistema de comunicación debe ser confiable y cumplir con ser multi-plataforma.

Se realizó una investigación de las alternativas, siendo las más relevantes DCOM, manejo de puertos (sockets) y CORBA. El sistema DCOM de Microsoft se descartó por ser específico de este sistema operativo. El manejo de puertos es un sistema de bajo nivel y su implementación es dependiente del sistema operativo. Optar por esta alternativa requiere desarrollar todo el manejo necesario para la administración de las conexiones y utilizar directivas de compilación y código independiente según la plataforma.

CORBA[8] es un estándar abierto que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos. Es una solución bien lograda, de alto nivel y orientada a objetos, que fue desarrollada para comunicar procesos pensando en brindar el mayor grado de independencia de las tecnologías. La comunicación CORBA opera localmente o a través de una red de datos y es independiente del lenguaje de programación que cada proceso utilice o del sistema operativo en que cada proceso se esté ejecutando. Para ello se definen interfaces de comunicación de forma que un proceso puede operar como un servidor que implementa métodos y otro como un cliente que utiliza dichos métodos. Pueden desarrollarse interfaces CORBA hacia múltiples lenguajes de programación y durante la investigación se realizaron experimentos con procesos codificados en Java y C++ que se comunicaban entre sí.

Para efectuar intercambios entre dos procesos, naturalmente es necesario que establezcan contacto. En CORBA se dispone de dos alternativas: la primera es que cada proceso tenga una referencia estática hacia el proceso con quien quiere comunicarse y la segunda es que se utilice un servidor de nombres. La primera alternativa requiere que cada proceso de alguna forma obtenga una referencia al otro, lo cual es fácil de obtener accediendo a un archivo compartido si se trata de una operación local, pero cuya implementación en la red se torna poco práctica. La segunda alternativa es un sistema concebido dentro del estándar de CORBA y se trata de un servidor de registro de nombres. Esta entidad es un proceso que está en ejecución y mantiene el conjunto de referencias necesarias de todos los procesos que se hayan registrado con ella. Luego de su registro con el servidor de nombres, un proceso puede solicitar la referencia de otro y así iniciar la comunicación. Por su versatilidad y conveniencia en un ambiente de red se optó por la utilización de CORBA con servicio de nombres (NSD).

Existen varias soluciones que implementan CORBA, entre ellas se optó por el paquete MICO[9] porque es una alternativa pública, de uso libre, multi-plataforma, con bibliotecas para C++ las cuales están bien documentadas. MICO implementa las funcionalidades necesarias para su utilización en el GCA como lo son: disponer de una NSD e implementar manejo de tipos de datos como vectores y estructuras. Además se contaba con documentación específica, libros, guías y ejemplos.

3. Necesidades adicionales a la biblioteca STL

3.1. Sincronización de procesos

Para la implementación del diseño planteado era necesario contar con manejo de hilos de ejecución y primitivas de sincronización de procesos (sección crítica, exclusión mutua, etc.). Para ello era necesario contar con bibliotecas que suministrasen las clases necesarias y cumpliesen con ser públicas, gratuitas y multi-plataforma. La biblioteca estándar STL es la utilizada para todas las funciones básicas por cumplir con las condiciones de licencia y aplicabilidad, pero carece de los manejos adicionales requeridos.

Se realizó un estudio de las bibliotecas existentes y se seleccionó la biblioteca CommonC++. Esta biblioteca se utilizó hasta que más tarde se descubrió que el conjunto de bibliotecas Boost tenía un mejor desarrollo y prestaciones.

Las bibliotecas Boost[10] siguen la filosofía de la biblioteca estándar de C++ y en su inicio el grupo de trabajo estaba integrado en su mayoría por miembros del comité de estandarización (C++ Standards Committee Library Working Group). Son bibliotecas de uso general, completas y muy bien logradas, varias de las cuales están propuestas al comité.

3.2. Persistencia

Uno de los objetivos del diseño del GCA es que el operador pueda administrar cómodamente el sistema. Por este motivo se diseñó una consola de administración que permite la gestión de los canales y sus cronogramas. Esto permite al operador configurar la mayor parte del sistema, especialmente las tareas frecuentes, sin necesidad de detener el sistema o editar manualmente archivos de configuración. Es necesario entonces que el sistema mantenga actualizados los archivos que describen el estado de las clases.

Se buscaron bibliotecas que implementasen la persistencia de las clases y que en lo posible los archivos de sistema que se generasen fuesen de tipo XML ya que es un formato estándar y portable.

Inicialmente se trabajó con la biblioteca SXP que cumplía con los requisitos. Pero en el momento en que se comenzaron a utilizar las bibliotecas de sincronización de procesos de Boost, también se cambió la biblioteca de persistencia.

Boost es preferible porque funciona correctamente, reduce la cantidad de proveedores de bibliotecas para el código, y cumple satisfactoriamente con los requerimientos. Esta biblioteca implementa la persistencia de todos los contenedores de la biblioteca estándar (STL), hace un manejo inteligente de punteros y trabaja de forma constructiva para facilitar la persistencia de estructuras de clases.

3.3. Otras necesidades

Adicionalmente existen bibliotecas Boost para manejo de estructuras de directorio y tipos de datos fecha y hora. El diseño requería disponer de estas funciones por lo que el uso de las bibliotecas Boost evitó el desarrollo de clases propias.

4. Interfaz gráfica

Para que la consola de operación fuese agradable y cómoda debía implementarse de forma gráfica. Se realizó un estudio de las bibliotecas gráficas disponibles con el propósito de seleccionar una que brindase las funcionalidades necesarias, fuese de uso libre y multi-plataforma. Las bibliotecas más avanzadas son Qt, GTKmm, SWT+Java y wxWidgets.

La biblioteca Qt si bien tiene un desarrollo completo y avanzado tiene restricciones de licenciamiento ya que no es de uso gratuito en Microsoft Windows o para aplicaciones comerciales.

La biblioteca GTKmm no está suficientemente madura para su utilización en Microsoft Windows. Además no es compatible con el compilador de Microsoft.

La utilización de Java compilado con la biblioteca SWT resulta en una interfaz bastante ágil y agradable. Esta opción es adecuada pero requiere realizar el código gráfico en lenguaje Java. Se prefirió optar por otra solución que no requiriese dedicar un tiempo adicional al estudio de un nuevo lenguaje de programación y un nuevo entorno de desarrollo. Las interfaces a desarrollar eran simples lo que apoyó la decisión tomada.

La biblioteca wxWidgets[11] es un desarrollo muy maduro, de uso libre, que fue diseñado con el propósito de ser portable y compatible con la mayoría de los compiladores. Está desarrollado originalmente para C++ y puede trabajar con archivos externos de definición de la interfaz gráfica. El archivo de definición es de formato XRC que se trata de un formato delimitado tipo XML con nombres de etiquetas dadas para cada objeto gráfico.

Se optó por la utilización de la biblioteca wxWidgets con archivos XRC. Los archivos XRC pueden ser generados por varias aplicaciones gráficas de diseño de interfaces. En estos archivos quedan definidos los objetos gráficos a utilizar junto con sus propiedades, relación y ubicación respecto a los otros objetos. En el código se instruye la lectura del archivo XRC lo que instancia los objetos para su utilización. Esta técnica permite separar la forma en que se presenta la interfaz gráfica, del comportamiento que adopta durante su utilización (definido por el código C++). La biblioteca tiene previsto el uso de tamaños relativos para los objetos, esto permite que puedan cambiar sus dimensiones en la medida en que se modifican las dimensiones de los contenedores (cuadros de diálogo o ventanas).

5. Recuperación y reproducción de audio

Para que un usuario pueda recuperar o reproducir a través de la red el audio almacenado, es necesario disponer de una arquitectura de tipo cliente-servidor. Ante la solicitud del cliente, el servidor debe ser capaz de devolver el audio o archivo solicitado.

Inicialmente se hizo un diseño que incluía un servidor y clientes propios al GCA. Esto requería que se resolviese la transmisión del audio a través de la red, lo que en el caso de la reproducción, implicaba resolver de forma eficiente la generación de un flujo continuo de audio. Además debería diseñarse la gestión de conexiones múltiples con las precauciones de reserva de recursos que esto implica. Finalmente, la necesidad de usar clientes propios obligaría a los usuarios a instalar localmente la aplicación cliente, lo que es una restricción incómoda en caso que se desee atender un conjunto de clientes amplio o distribuido. Se buscó entonces una solución que ya estuviese desarrollada con el propósito de transferir audio y adaptarla para su utilización en el GCA.

5.1. Reproducción en tiempo real

Para la transmisión de audio a través de la red con el propósito de reproducción, la solución ideal es utilizar la tecnología de “streaming” en la que se genera una conexión entre el cliente y el servidor dentro de la cual el servidor va enviando el audio en la medida que el cliente lo reproduce. Esto evita descargar de antemano el archivo de audio que se desee reproducir y reduce al mínimo el tiempo de espera entre la solicitud del audio y el comienzo de la reproducción. El protocolo utilizado para streaming es el estándar RTSP (RFC 2326) [7] para la transmisión bajo demanda de contenido de tiempo real (como audio y vídeo), asumiendo su utilización en entornos con conexiones concurrentes. Existen servidores de streaming gratuitos como el Helix y clientes populares como el Real Player [12]. Esta tecnología está extensamente difundida y es utilizada por ejemplo para la transmisión de radio en vivo a través de Internet.

Para la reproducción del audio, es necesario que el programa cliente de RTSP, luego de conectarse al servidor de streaming, indique cuál es el archivo de audio a reproducir y si corresponde, el corrimiento desde el principio del mismo. Los clientes RTSP son capaces de interpretar varios archivos de definición que existen con este objetivo.

El lenguaje SMIL [6] es un estándar que establece una forma de definir presentaciones multimedia. Se genera un archivo de tipo XML con directivas que permiten establecer secuencias de contenidos multimedia. Es un lenguaje fácil de utilizar que permite desarrollar presentaciones complejas de audio y vídeo. Maneja indicación de desplazamiento inicial o final para cada archivo multimedia a reproducir y permite la reproducción en secuencia y en paralelo. Varios clientes RTSP, entre ellos el Real Player, son capaces de interpretar SMIL.

Se optó entonces por utilizar un servidor de streaming externo al GCA el cual debe configurarse para que acceda a los directorios donde están almacenados los archivos de audio. Para instruir al cliente RTSP se optó por utilizar el formato SMIL estableciendo la secuencia de los archivos de audio a reproducir y el corrimiento inicial y final para ajustar exactamente la reproducción de acuerdo a la solicitud requerida.

5.2. Solicitud del audio

Para completar la solicitud, es decir establecer el canal y marcos de fecha del audio que se desea recuperar, se optó por una solución basada en una página web. Esto evita que el cliente requiera

aplicaciones especiales, ya que para solicitar el audio sólo necesita acceder, utilizando un navegador estándar de Internet, a una página web que contiene un formulario de consulta.

Para la tarea de búsqueda de los archivos que cumplen con la consulta realizada, se optó por desarrollar un código en lenguaje PHP. PHP[13] es un lenguaje de automatización estándar y público que puede ser integrado en el código HTML de las páginas web. Los servidores de páginas web más conocidos disponen de intérpretes para PHP, lo que permite la interacción y realización de tareas desde una interfaz web. El código PHP puede permanecer privado, en el servidor web y ejecutarse en el servidor para luego entregar el resultado necesario al cliente.

Se desarrolló en PHP un código que se ejecuta en el servidor de páginas web cuando el cliente confirma el formulario de búsqueda. Como resultado, se obtiene una página de contenido dinámico que indica si la consulta fue exitosa y contiene los hipervínculos a los archivos de audio que cumplen el criterio. Además, se puede descargar el archivo SMIL que contiene la definición de la secuencia de reproducción correspondiente a la consulta.

6. Archivos externos ejecutables

Durante el diseño, se buscó brindar un producto que fuese adaptable, tratando de minimizar las decisiones que resultasen en limitación de prestaciones. Por este motivo se decidió dejar a cargo de programas externos la realización de las tareas que por su naturaleza fuese más posible que cada cliente deseara realizar según sus necesidades y el avance de nuevas tecnologías.

Según las configuraciones que realice el operador del sistema, se pueden lanzar programas externos en momentos claves del funcionamiento del GCA. Cuando se completa el almacenamiento de un archivo de audio de un canal, el sistema puede lanzar un proceso independiente con el archivo ejecutable y parámetros definidos. Asimismo, en el módulo de control de fallas, al detectar una falla o pérdida de comunicación, el sistema puede lanzar un proceso independiente.

7. Servicios y registro de sucesos

Durante el proceso de investigación se desarrollaron numerosos programas simples que se ejecutaban desde la línea de comandos. Los procesos quedan en estos casos vinculados a la sesión de usuario o a la terminal de inicio. Para un sistema como el GCA, que debe operar de forma continua y autónoma, es imprescindible que los procesos se ejecuten independientemente de la sesión de usuario iniciada o los cambios de sesión que se realicen. Para ello el proceso central debe iniciar como un servicio del sistema operativo.

Se investigó y probó el modo de implementar servicios en Linux y Microsoft Windows. En este último, la interfaz de programación es bastante compleja lo que motivó el desarrollo de una clase con varios métodos para manejar las distintas etapas de creación, instalación y ejecución del servicio. Estos conceptos y estilo se portaron para los métodos necesarios en Linux, con lo que se pudo encapsular toda la lógica de servicios de forma independiente de la plataforma. Se desarrolló entonces una clase básica, con directivas de compilación según el sistema operativo, de la cual se derivan aquellas que implementan los procesos del GCA.

Los servicios y demás procesos normalmente utilizan el registro de sucesos del sistema operativo para informar y dejar archivado los eventos significativos que ocurren durante su ejecución. Linux

dispone para esto de un servidor syslog mientras que en Microsoft Windows se usa el registro de sucesos propio. Aplicando el mismo criterio que para los servicios, se encapsularon las particularidades de cada interfaz de programación y se unificaron por correspondencia los niveles de criticidad, obteniendo una clase que registra sucesos y presenta un único método de utilización para ambos sistemas operativos.

8. Instaladores

Una vez avanzada la implementación del código, naturalmente surgió la necesidad de definir la forma en que se realizaría la instalación del mismo. Con el propósito de facilitar esta tarea era necesario seleccionar una solución basada en la extracción y ubicación de los programas ya compilados. Se debió utilizar una técnica distinta para cada sistema operativo.

En Linux se optó por la utilización de paquetes RPM[14] que son procesados por el programa “rpm”. Este programa es de uso público y la técnica de instalación es muy difundida, de operación simple y conocida (todos los paquetes RPM tienen las mismas directivas para instalar, actualizar y desinstalar el programa). Un paquete RPM se genera a partir de un archivo de definiciones (de formato y extensión SPEC) que instruye la compilación del código, selección de los archivos a incluir en el paquete de instalación y tareas a realizar durante la instalación o desinstalación del programa. Los archivos indicados se guardan dentro del archivo de extensión rpm, junto con el archivo de definiciones. El formato del archivo de definiciones es versátil y está bien documentado.

En Microsoft Windows se genera un archivo de instalación ejecutable. Para generarlo se optó por utilizar el programa Inno Setup[15] que es gratuito y bien difundido. Este programa gráfico tiene un módulo que produce el ejecutable y una herramienta que permite editar el archivo de definiciones con las instrucciones para la generación del ejecutable. El archivo de definiciones utilizado es de formato ISS que es un formato público y debidamente documentado.

6. Estudio del mercado

1. Mercado genérico de los grabadores

En el mercado pueden encontrarse sistemas de grabación de tipo portátil, por ejemplo para grabación de entrevistas, y sistemas no portátiles para grabación de fuentes de audio fijas. Dentro de los sistemas fijos, existen familias de productos orientados a diferentes necesidades.

Comunicaciones por radio

Los grabadores para el mercado de radiocomunicaciones graban solamente cuando existe audio, utilizan baja calidad y por ende alta tasa de compresión. Algunos grabadores permiten barrido de frecuencias y tienen medidores de potencia y presencia de señal.

Registro de llamadas telefónicas

Estos grabadores en su mayoría están destinados a centros de atención de llamadas (Call-Center) y están provistos de una gran cantidad de canales. Normalmente son equipos en un hardware dedicado y permiten la grabación y respaldo en cinta. Tienen la banda limitada en 3400 Hz y utilizan códecs de voz para minimizar el tamaño del audio grabado.

Conferencias y registros legales

Se trata de grabadores pensados para registrar sesiones donde hablan varias personas frecuentemente en simultáneo o con gran interacción. Requieren para ello varios canales (al menos cuatro) y la calidad de audio es mejor que la utilizada en los grabadores de llamadas telefónicas. Los grabadores para registros legales suelen suministrarse con accesorios como pedales de control de marcha para ayudar al operador que debe pasar por escrito las sesiones. Frecuentemente los grabadores de este segmento suelen ofrecerse como grabadores de alta calidad para radiodifusión pero normalmente no tienen implementado sistemas de detección de fallas de operación.

Radiodifusión

Finalmente en este sector se encuentran los grabadores que permiten por un lado la operación continuada para registro legal del audio emitido y a la vez la grabación en alta calidad que permita edición y reutilización adecuadas. Estos grabadores están pensados para una operación ininterrumpida e implementan normalmente sistemas de detección de fallas.

2. Grabadores de alta calidad de audio

El GCA es un grabador que permite el registro en alta calidad y fue pensado para aplicaciones de radiodifusión. Se estudió en mayor detalle entonces dicho mercado analizando las diferentes ofertas. Dentro de este mercado existen productos con diferentes niveles de prestaciones. Existen productos simples que son normalmente aplicaciones diseñadas para adquirir audio de un único canal y que, aunque puedan tener funciones avanzadas como streaming, análisis del audio o detección externa, no tienen implementados controles para asegurar el correcto e ininterrumpido funcionamiento del sistema. Tales productos son de muy bajo costo, pero por su baja confiabilidad y limitación de canales, se han excluido del estudio.

Entre los grabadores de alta calidad para uso profesional existen productos que incluyen el hardware de adquisición y/o procesamiento y almacenamiento, mientras que otros productos son únicamente el software que debe ser instalado en un equipo tipo PC que ya disponga de tarjetas de audio.

Los grabadores que incluyen hardware suelen ser menos versátiles y más escasos en funciones, pero supuestamente son más confiables ya que no están sujetos a fallas del sistema operativo u otras aplicaciones que puedan estar ejecutándose en la misma PC. Estos productos entregan dentro de sus especificaciones los parámetros eléctricos de interferencia entre canales y otros datos de respuesta. Frecuentemente tienen clientes propietarios para permitir la obtención del audio grabado en ellos, los cronogramas de grabación son limitados a pocos programas y carecen de opciones para automatización de tareas o interacción con otras aplicaciones.

Los grabadores por software suelen ser más versátiles y permiten mejor acceso al audio ya que el mismo suele grabarse en algún formato estándar en un disco de la PC. Al disponer de acceso directo al audio es posible la interacción con otros programas que disponga el cliente. A su vez, es frecuente que estos grabadores aprovechen las ventajas del sistema operativo para realizar copias por la red, respaldos o generar correos electrónicos. Los grabadores por software operan sobre Microsoft Windows. No hemos encontrado grabadores dentro de este segmento que funcionen en Linux u otros sistemas operativos.

3. Funciones principales

3.1. Detección de fallas

La mayoría de los grabadores profesionales tiene algún sistema para asegurar la continuidad de la operación.

Uno de los productos estudiados (Arbor LogDepot) está especialmente desarrollado para garantizar la continuidad de grabación. Este producto tiene adquisidores propietarios con capacidad de almacenamiento local de varias horas para evitar la pérdida de registro en caso de fallas del procesador. Adicionalmente pueden instalarse adquisidores que se configuren como redundantes.

Los demás productos pueden detectar fallas de funcionamiento y escriben un registro o envían un correo o mensaje de aviso. No hemos encontrado que los demás productos puedan ejecutar otro tipo de tareas ante la detección de una falla.

Algunos grabadores pueden programarse para detectar como condición de falla la ausencia de audio (audio por debajo de un umbral dado).

3.2. Programas de grabación

La gran mayoría de los productos no tiene un cronograma abierto, la programación suele tener una base semanal donde se indica cada día en qué intervalos se debe grabar el audio en cada canal.

Muchos de los grabadores en este segmento permiten la detección de presencia de audio y pueden activar la grabación o detención en función de la existencia de audio. Adicionalmente, algunos grabadores detectan la entrada de joystick o del puerto RS-232 para comandar el inicio o detención de grabador. Esto se utiliza por ejemplo para sistemas de recepción por satélite.

3.3. Formatos de audio

La frecuencia de muestreo y el formato en que se guarda el audio normalmente se configura por canal y es independiente del cronograma de grabación. Para resolver la necesidad de disponer de audio en alta calidad para edición, etc. y además el registro legal en calidad mínima, algunos grabadores permiten generar varias copias en distinta calidad del mismo audio adquirido. La ventaja que reporta esto es poder devolver el audio rápidamente sin necesidad de convertirlo en caso que un cliente solicite audio de menor calidad que la requerida para tareas de edición.

De los productos relevados, algunos pueden tener tareas programadas a intervalos periódicos o comandadas asincrónicamente para realizar copia de archivos o respaldo pero no se ha encontrado que puedan disparar las tareas al momento en que se cierra cada archivo de audio.

3.4. Administración y clientes

La capacidad de administración remota y el cliente para reproducción o recuperación del archivo de audio suele venderse como una aplicación propietaria con un costo adicional. En el caso de Arbor y WinJay el cliente es web pero en los demás grabadores la aplicación funciona únicamente en Microsoft Windows.

3.5. Licenciamiento

Los grabadores tienen distintos parámetros de licenciamiento. Varios de los grabadores se licencian por cantidad de canales y algunos por la antigüedad deseada del audio (borran el audio al cabo de una cantidad dada de días y si se desea una mayor permanencia debe adquirirse otra licencia). Frecuentemente los clientes para administración por la red o de recuperación del audio tienen un costo adicional y se licencian por simultaneidad de uso. Algunos grabadores tienen licencias diferentes en la medida que se requiere mayor calidad de adquisición.

4. Matriz de comparación

La siguiente matriz representa las funciones principales de los productos que hemos encontrado competitivos en el mercado de radiodifusión y cuyo nivel de prestaciones los enmarca dentro del mismo perfil que el apuntado por el GCA.

Producto	Precio US\$	Tipo	Nº Canales	Almacenamiento	Calidades y Formatos	Cronograma	Adm. y Clientes	Det. Fallas	Otras Funciones
InnesCorp Flash-Log 5	7000	HW	4	42/90/180 d.	13 y 18 kHz. Formato propietario.	Poco versátil	Local o remota con cliente prop.	Sí	
Sonifex Net-Log	3500	HW	2	200 GB	Varias calidades. MP2.	Poco versátil	Remota con cliente prop.	Sí	Detección de audio y RS-232
Arbor LogDepot Audio		HW	esc.	Disco en PC remoto	Varias calidades. PCM, MP2 y MP3.	Poco versátil	Web	Sí	
Eventide VR208HF		HW	8	72 GB	3.5, 13 y 18 kHz. Formato Propietario.	Poco versátil	Local u Opcional remoto con cliente prop.	Sí	
OMT iMedialogger	900/1200/1600	SW	1/2/4 (12)	Ilim.	Varias calidades. MP3, MP2, WAV, WMA, Real.	Versátil	Web	Sí	Detección de audio y Joystick
P Squared Power-Log	3400	SW	8	Ilim.	Varias calidades. MP2, OGG, WAV, WMA.	Versátil	Local o Lan con cliente prop.	Sí	Detección de audio, RS-232 y otros. Alarma por falta de audio
IPShare AudioStream 2000 Pro	1500	SW	5	Ilim.	Varias calidades. MP3.	Versátil	Local, no tiene cliente remoto	No	Detección de audio y joystick. Incl. analizador de onda
Winjay LogJay		SW	4	Ilim.	Todos los soportados por MS Windows.	Versátil	Admin. local o web, cliente de reprod. prop. (opc.)	Opc.	Detección de audio y joystick
GWS NovaLog	650	SW	1	Ilim.	Calidad media. Formato Real.	Poco versátil	Admin. sólo local, cliente prop. (opc.)	No	

Tab. 1: Comparación de grabadores del mercado

7. Mediciones y ensayos realizados

Este capítulo describe los ensayos y mediciones realizadas de donde se deduce el costo de implementación del GCA en una computadora.

1. Mediciones del GCA

1.1. Entorno de ensayo

Equipamiento utilizado

Se ha medido el consumo del GCA tanto en Linux como en Microsoft Windows para tres computadoras de diferentes características. Se ha evaluado el consumo de CPU y memoria de cada proceso para poder realizar una estimación de los recursos necesarios con que se debe contar para la utilización del sistema.

Las medidas se han realizado con el siguiente equipamiento:

Procesador	Memoria RAM	Sistema Operativo
AMD Athlon 1333 MHz	512 MB	Windows XP Pro
AMD Athlon 1913 MHz	512 MB	Linux Fedora Core 2
Pentium II 333 MHz	384 MB	Linux Fedora Core 3

Mediciones en Linux

En Linux se ha utilizado de forma básica la herramienta OProfile[16] para estimar el consumo de CPU por tarea, la herramienta “top” y las estadísticas de los archivos de procesos (archivos “/proc/pid/status”) para evaluar el consumo de memoria junto con la herramienta “vmstat” para verificar en detalle la presencia de faltas de página (eventos “page fault” que causan “swap-in”).

La estimación de utilización de memoria en Linux es compleja porque muchos procesos realizan “caché” en memoria lo que resulta en un aparente agotamiento de la memoria al guiarse por la estadística básica de memoria libre RAM. Sin embargo, la memoria utilizada para “caché” es liberada automáticamente cuando se requiere espacio para memoria de ejecución o datos. Además, la memoria compartida de bibliotecas que se utilizan en común para los procesos del GCA se presentan repetidas por cada proceso lo que, en una primera vista, devolvería un consumo altísimo de memoria RAM que incorrecto.

Se intentó estimar la memoria activa que efectivamente utilizan los procesos para su funcionamiento basándose en el espacio utilizado para el código (parámetro RSS, Resident Set Size) y teniendo en cuenta un valor adicional para el uso de memoria compartida de bibliotecas.

El consumo de CPU se estimó para los procesos básicos sin medir procesos externos como los de post-grabación.

Mediciones en Microsoft Windows

En Microsoft Windows se utilizaron las herramientas propias del visor de tareas repitiendo el inicio y detención de procesos para confirmar la consistencia de los resultados.

1.2. Resultados obtenidos

Se observó que el consumo estimado para Linux es ligeramente inferior que en Microsoft Windows lo cual puede explicarse por diferencias de codificación en las bibliotecas utilizadas, el código específico del GCA que maneja los controladores y las propias llamadas de cada sistema operativo (por ejemplo las llamadas de acceso a los dispositivos de audio, disco, etc.). Las medidas de la Consola gráfica dan un estimado superior para Linux en cuanto a consumo de memoria.

El consumo de los procesos relativos a un canal se estimaron de forma conjunta ya que lo relevante es disponer de una medida que permita la previsión de un factor de escalado.

De forma general para ambos sistemas operativos se puede estimar que el consumo de memoria RAM del sistema principal que controla los procesos es de 10 MB y debe contarse de 10 a 12 MB por cada canal activo. El servicio de nombres de CORBA que es necesario para el funcionamiento del sistema no consume más de 6 MB. El servicio de Control de Fallas en su funcionamiento básico consume menos de 5 MB. La Consola del sistema consume entre 10 y 20 MB de memoria RAM.

Un sistema completo con un canal requiere entonces un mínimo de 50 MB de memoria RAM libres para evitar el uso de paginado a disco. Por cada canal adicional deben contarse no más de 12 MB adicionales. Estas cotas son superiores y garantizan un funcionamiento correcto del sistema. Es necesario considerar memoria adicional para la ejecución de procesos externos y el funcionamiento de los servidores de páginas web y streaming.

El consumo de CPU es muy bajo en cualquier computadora moderna. Las medidas más críticas pudieron verse en el procesador de 333 MHz donde se observa que el proceso Grabador de cada canal utiliza un máximo de 5 % del tiempo de procesamiento. Los demás procesos tienen un consumo mínimo. La Consola es el otro proceso con más consumo cuando se activa la monitorización de un canal, en este caso se observa una utilización de hasta un 5 %. En las computadoras con procesador más veloz, la utilización no supera el 2 % para cualquier proceso del GCA.

La siguiente tabla muestra los estimados para cada computadora ensayada, la memoria RAM se expresa en MB y el uso de procesamiento en porcentaje:

Computadora	512 MB	1333 MHz	512 MB	1913 MHz	384 MB	333 MHz
Sistema Operativo	MS XP	MS XP	FC 2	FC 2	FC 3	FC 3
Datos por Proceso	RAM	CPU	RAM	CPU	RAM	CPU
CORBA NSD	6	<0,2	3	<0,2	4	<0,5
Control de Proceso	7	<0,2	5	<0,2	7	<0,5
Canal	12	<1	10	<1	12	<5
Control de Fallas	4	<0,2	4,5	<0,2	5	<0,5
Consola	9	<2	18	<2	20	<5

Tab. 1: Resultados de mediciones de consumo del GCA

2. Ensayo de operación continua

2.1. Radio El Espectador

En la radio “El Espectador” se instaló una versión preliminar del sistema GCA que está adquiriendo un canal de audio con la programación que se emite. El servidor es de propósito general, utiliza sistema operativo Linux y tiene como proceso post-grabación un archivo de comandos que comprime el audio en formato MP3 utilizando el programa “lame” y luego borra el archivo de extensión “.wav”. El servidor está funcionando desde el 17/05/2005 sin fallas ni deterioro de recursos.

2.2. Mesa de ensayos

El sistema GCA se ensayó de forma extensiva en una de las computadoras que se utiliza para el desarrollo del código. Esta computadora tiene un procesador AMD Athlon de 1913 MHz y 512 MB de RAM, utiliza sistema operativo Linux Fedora Core 2.

Se instalaron 3 tarjetas de sonido y se ensayó la operación de los 3 canales utilizando un cronograma continuo y un cronograma exigente que generaba una grabación de 30 segundos por cada minuto. Se utilizaron procesos de post-grabación para comprimir en formato OGG (utilizando el programa “oggenc”) y en formato MP3 (utilizando el programa “lame”). En todos los casos se comprobó un funcionamiento correcto durante más de cinco días sin agotamiento o deterioro de recursos inclusive estando activa la interfaz gráfica X y utilizándose normalmente la computadora para otras tareas exigentes como compilación y aplicaciones gráficas.

3. Costo del GCA

De los ensayos realizados se puede concluir que puede implementarse un sistema GCA completo de por lo menos cuatro canales en una computadora general moderna con 512 MB de memoria RAM y 1500 MHz de procesador. El costo de una computadora de estas características es de aproximadamente 500 US\$ a la fecha actual (quinientos dólares americanos en julio de 2005).

8. Perspectiva de desarrollo posterior

1. Motivos para un desarrollo posterior

El GCA es un sistema de grabación que fue desarrollado con el propósito de satisfacer las necesidades de un grabador para radiodifusión pero teniendo en cuenta otras aplicaciones futuras en su arquitectura. Tiene características que lo distinguen de su competencia pero carece de algunas funciones que están presentes en otros productos. Incorporar estas funciones mejoraría la competitividad respecto a los otros grabadores estudiados. Además existen otras funciones que pueden desarrollarse para mejorar la administración del sistema, y que contribuirían en distinguirlo del resto. Por último puede pensarse en extender las funciones del GCA para cubrir las necesidades de otros segmentos del mercado como por ejemplo los grabadores de conversaciones.

2. Mejoras en la administración

2.1. Reproducción en vivo del audio

La consola de administración permite iniciar y detener los canales, editar el cronograma y monitorizar gráficamente el nivel del audio que está siendo adquirido. Pero no dispone de la facilidad de escuchar en vivo el audio que se ha adquirido y que está siendo procesado para grabarse en el disco. Esta función es importante para ayudar al operador a ajustar el volumen además de mejorar el nivel de confianza del operador al monitorizar el sistema. Como la consola está pensada para operar a través de la red es necesario que esta función preserve dicha facilidad. Para esto podría emplearse un servidor de streaming y la consola se conectaría para solicitar el audio que está siendo grabado.

En general los servidores de streaming especifican un protocolo para que clientes de captura de audio en vivo se conecten a ellos vía IP. El módulo grabador del GCA podría transformarse en un cliente como el mencionado.

2.2. Alarma por falta de audio

Existe el riesgo que se desconecte físicamente la fuente de audio por accidente o se produzca una falla en los sistemas que la generan. En estos casos el canal estaría en silencio lo que podría considerarse una condición de alarma. Dentro del grabador, a nivel del procesamiento intermedio entre la adquisición y el registro a disco, puede calcularse el nivel del búfer de audio y compararlo con un nivel predefinido, para clasificarlo como silencio o señal. El umbral y duración del silencio estarían dados por canal y establecidos por el operador. Si el silencio excede el tiempo configurado como admisible, se señalaría una condición de alarma.

2.3. Forzar cambio de archivo

El GCA adquiere el audio, lo guarda a disco en formato WAV y luego inicia un proceso externo para procesarlo. Por algún motivo el operador puede desear procesar o disponer del archivo de audio actual sin tener que esperar que el mismo se complete a la duración establecida para el canal. Para realizar esto se puede agregar una instrucción al grabador que fuerce un cambio de archivo bajo comando.

2.4. Eliminación automática de audio antiguo

Actualmente el GCA graba el audio en disco y detiene la grabación señalizando una falla cuando el espacio libre en disco llega a un mínimo establecido. Podría desarrollarse una función que elimine periódicamente el audio de antigüedad superior a una cantidad dada de días. Tal función ayudaría a mantener el espacio libre en disco sin necesidad de recurrir a una tarea externa programada. Adicionalmente podría incorporarse una función que actúe en caso que se llegue al mínimo de disco y elimine los archivos de audio más antiguos hasta obtener un margen dado, de esta forma no se señalaría la falla y el sistema seguiría grabando normalmente.

2.5. Módulo SNMP para monitorización de estado

Hoy día es cada vez más común que las aplicaciones dispongan de un servidor SNMP para que otras aplicaciones de supervisión puedan consultar el estado o mantener un registro a lo largo del tiempo de las variables significativas del sistema. En el caso de los servidores Web se les puede consultar si están operativos o cuantas conexiones están activas, en los servidores de correo se puede consultar la cantidad de mensajes en cola para ser enviados, etc. Del mismo modo el GCA podría informar por SNMP el largo de las colas de procesamiento, el estado de grabación o enviar mensajes trap ante condiciones de falla.

3. Funciones adicionales para radiodifusión

3.1. Detección de señal externa de encendido

Otros grabadores son capaces de detectar que se ha activado un comando externo utilizando el puerto de joystick de la tarjeta de audio o el puerto paralelo. Según estas indicaciones son capaces de iniciar o detener la grabación de audio. Estos puertos (especialmente el paralelo) son capaces de recibir más de una señal diferente de encendido (ON/OFF) lo que permite comandar varios canales. Podría desarrollarse una extensión para el control de procesos que monitorice un puerto externo a la espera de un comando de activación e instruya cada canal según sea necesario.

3.2. Separación de audio útil/audio de tandas

Cuando se toma parte del audio del grabador continuo con el fin de editarlo y almacenarlo en la audioteca, generalmente se busca excluir las tandas publicitarias. El trabajo de excluir las tandas publicitarias usando un editor de audio es tedioso y repetitivo. El software de asistencia de operador al aire que emite las tandas publicitarias a menudo está diseñado para cadenas de emisoras geográficamente dispersas. Una estación maestra envía señales de comienzo y fin de tanda que reciben las

esclavas para emitir las tandas locales sincronizadas con las de la maestra. Esta señal bien podría recibirla un módulo del GCA que la transmita al grabador para que éste vaya cortando los archivos en los instantes en que comienzan y terminan las tandas. Incluso podría dejar de grabar durante la tanda.

3.3. Módulo para programas de automatización radial

Los programas de automatización radial generalmente almacenan los archivos de audio en directorios específicos y permiten asociar un conjunto de información adicional por cada archivo. Es habitual que esa información se almacene en forma de registros de una base de datos relacional. Es interesante poder reproducir archivos almacenados por el GCA desde el programa de automatización radial, por ejemplo para consulta o re-emisión. Entonces podría agregarse al proceso de post-grabación, un cliente ODBC que agregue un registro por cada archivo creado por el grabador a la base de datos del software de automatización.

4. Ampliación a otros segmentos del mercado de grabadores

Incorporando ciertas funciones puede extenderse el uso del GCA para la grabación de conversaciones telefónicas, grabación de conferencias, sesiones jurídicas o parlamentarias. Además, desarrollando hardware complementario puede obtenerse un producto más completo o profesional orientado a estos segmentos de mercado.

4.1. Comando de grabación por presencia de audio

El GCA inicia la grabación según un cronograma por canal pero podría habilitarse un modo de operación en que se adquiera el audio de forma continua y se evalúe la presencia de silencio o señal. El modo de realizarlo se describe en la sección 2.2. Evaluando cada búfer de muestras se registrarían aquellas que se clasifiquen como señal, mientras que los silencios se registrarían siempre que su duración no exceda un valor dado. Esta facilidad elimina la necesidad de operación manual, permite la reducción del audio almacenado y a la vez preserva el contenido significativo.

4.2. Ajuste automático de volumen

A diferencia del audio producido por una emisora de radio que es procesado para mantener el nivel dentro de un rango, las conversaciones telefónicas son de nivel variable y está fuera del control de los administradores del sistema. Además, esta atenuación es variable de una a otra llamada. Lo mismo ocurre en asambleas donde diferentes participantes intervienen, las condiciones no están tan controladas como en un estudio de grabación profesional. Es útil entonces un proceso que sea capaz de medir el volumen del audio adquirido y corregir el nivel del canal según sea necesario. Este proceso puede incluirse en el control de grabación donde ya está disponible la conexión con el grabador para tomar las medidas de nivel.

4.3. Módulo para detección de corriente de bucle

En el caso específico de grabación de llamadas telefónicas podría implementarse un hardware para detectar la corriente de bucle y determinar exactamente el inicio y final de cada llamada. Esta señal comandaría la grabación del audio sin necesidad de activar un detector de silencio. Esta función permite asegurar que una conversación que tiene un silencio intermedio (producido tal vez por la

ausencia momentánea del cliente) quede guardada de forma continua en un único archivo de audio. La señal proveniente del hardware de detección de bucle puede ingresar por algún puerto externo como puede ser el paralelo, serie o de joystick. Debe disponerse entonces de un módulo como el descrito en 3.1.

4.4. Registro de abonado llamante y de tonos digitados

Si se ha desarrollado el hardware de detección de corriente de bucle puede incorporársele un módulo para identificación de DTMF que detecte los tonos digitados en la línea y un módulo que interprete el identificador de abonado llamante enviado por la central telefónica.

5. Otras funciones

5.1. Internacionalización

Para ampliar el mercado objetivo del GCA podría traducirse su interfaz de usuario y manuales a otros idiomas. Para ello habría que utilizar una herramienta como Gettext o similar.

5.2. Soporte para unidades múltiples de almacenamiento

Actualmente el GCA dispone de un atributo por canal que es el directorio donde debe almacenarse el audio de dicho canal. El grabador se inicia con esta información y guarda a disco en el directorio establecido. Si se dispone de varias unidades de almacenamiento se podría ampliar la capacidad manejando una lista ordenada de directorios por canal. Para esto el grabador debería cambiar el directorio destino del audio en la medida que los medios alcanzan un espacio libre establecido como mínimo. Como alternativa también pueden utilizarse las herramientas provistas por los sistemas operativos para tratar varias unidades como una sola.

Parte II

DESARROLLO

9. Manual del proyecto

1. Objetivos

El propósito de este documento es presentar un modelo del proceso de desarrollo utilizado para realizar el GCA. También especifica los documentos que serán preparados en las distintas etapas del desarrollo.

2. Proceso de desarrollo

El proyecto atraviesa cuatro fases de desarrollo: Principio, Elaboración, Construcción y Transición. Cada fase contiene una o más iteraciones. El flujo de una iteración típica se ilustra en la figura 1. Las fases del proceso, sus iteraciones y la cantidad de esfuerzo empleado en los distintos flujos de trabajo se ilustran en la figura 2.

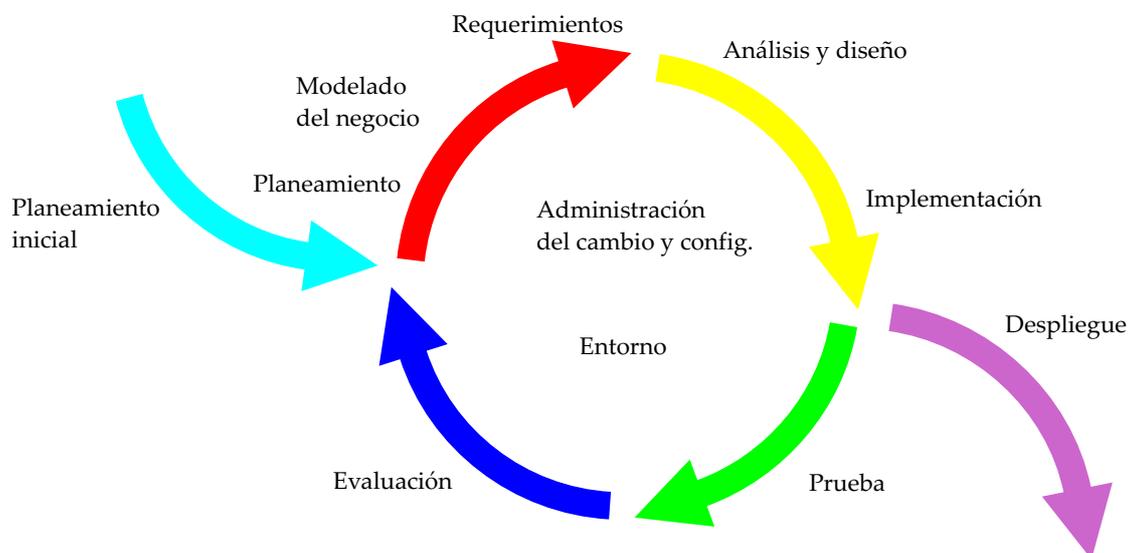


Fig. 1: Flujo de una iteración típica.

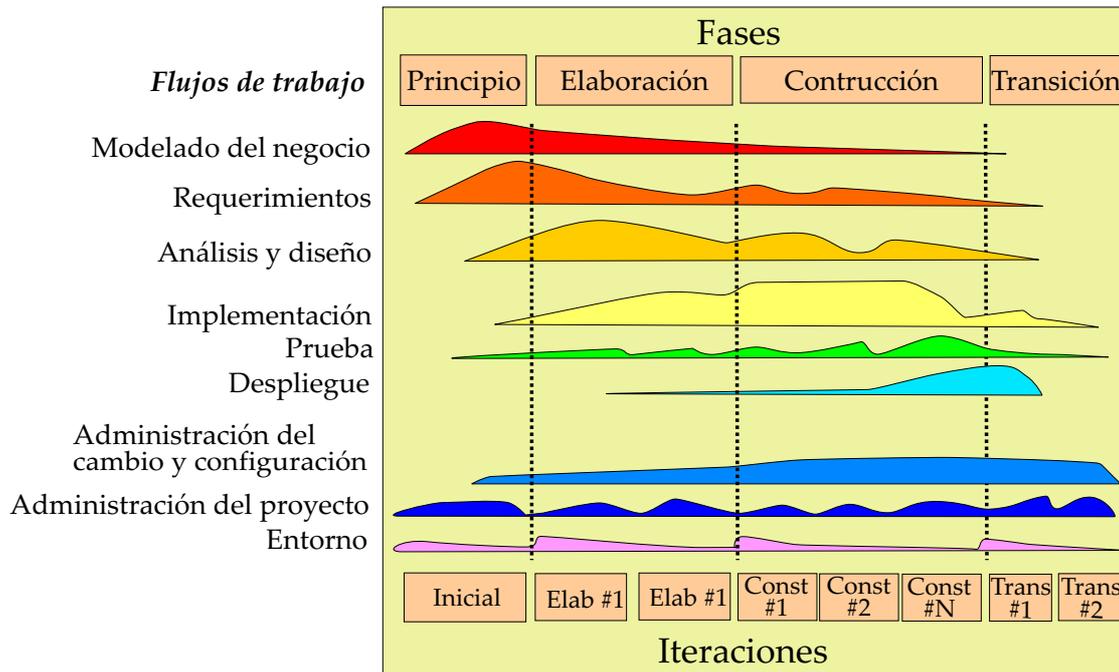


Fig. 2: Fases del desarrollo y flujos de trabajo de un proyecto típico

2.1. Principio

En el principio del desarrollo se define qué será el sistema, posiblemente quién lo usará y por qué, qué funciones y restricciones deben estar presentes.

Actividades

Formular el alcance del proyecto Definir qué es el sistema a desarrollar y cómo satisfacer a las partes involucradas. Se debe capturar el contexto y los requerimientos más importantes en suficiente detalle como para derivar un criterio de aceptación del producto.

Planear y preparar los motivos y beneficios del proyecto Con el documento de visión como guía, definir la estrategia de mitigación de riesgos, desarrollar un plan inicial e identificar costos conocidos si corresponde.

Sintetizar una arquitectura candidata Seleccionar tecnologías, sintetizar una arquitectura candidata y eventualmente desarrollar un prototipo para mitigar riesgos mayores del proyecto.

Preparar el entorno del proyecto Seleccionar y preparar las herramientas de software, recursos físicos y procedimientos que utilizará el equipo de desarrollo.

2.2. Elaboración

El objetivo de esta fase es elaborar la arquitectura del sistema para proveer una base estable para el grueso del esfuerzo de diseño e implementación de la fase de construcción.

La arquitectura evoluciona a partir de la consideración de los requerimientos más importantes y una valoración del riesgo. La estabilidad de la arquitectura puede evaluarse a través de uno o más prototipos arquitectónicos.

Actividades

Definir, validar y crear una base de la arquitectura

Refinar la visión

Crear los planes de iteración para la fase de construcción

2.3. Construcción

El objetivo de esta fase es completar el desarrollo del sistema. Esto se logra mediante sucesivas iteraciones que expanden e integran los módulos definidos durante la elaboración hasta llegar a un conjunto estable y funcional que forma el sistema. En esta etapa se elabora el software y también parte de la documentación del sistema (descripción interna del sistema, manuales de usuario, ensayos).

Actividades

Administrar recursos y controlar el proceso

Desarrollar y probar componentes

Valorar la iteración

2.4. Transición

El foco de la transición es asegurar que el software esté disponible para sus usuarios finales. Incluye pruebas del producto y realización de ajustes menores basados en la realimentación de los usuarios.

Actividades

Finalizar el material de soporte del usuario

Probar el producto en el ambiente del cliente

Afinar el producto basándose en la realimentación del cliente

Entregar el producto final

3. Documentos de las distintas fases del desarrollo

En el cuadro 1 se muestran los documentos a producir a lo largo del desarrollo.

	Principio	Elaboración	Construcción	Transición
Requerimientos	Documento de visión			
	Especificación de los casos de uso	Especificación de los casos de uso		
	Especificación suplementaria	Especificación suplementaria		
Diseño		Investigación de bibliotecas a utilizar		
		Documento de diseño		
Implementación		Reporte de la prueba de evaluación	Reporte de la prueba evaluación	
Despliegue				Notas de la versión
Administración	Plan del proyecto (incl. Plan de iteraciones de elaboración)	Plan del proyecto (incl. Plan de iteraciones de construcción)	Plan del proyecto (Plan de despliegue)	Plan del proyecto (Plan de despliegue)
	Lista de riesgos - principio	Lista de riesgos - elaboración	Lista de riesgos - construcción	
	Manual del proyecto	Plan de prueba	Plan de prueba del sistema	
Estándares y guías	Guía de estilo del código fuente	Guía de estilo del código fuente		

Tab. 1: Documentos de las distintas fases del proyecto

10. Especificación de los casos de uso

1. Objetivos

El propósito de este documento es definir los requerimientos funcionales del sistema a través de sus casos de uso y actores.

2. Actores

Usuario Es el que solicita al sistema la recuperación o reproducción del audio almacenado a través de diferentes interfaces.

Operador Es el que administra el GCA y controla que funcione correctamente.

Fuente de audio Representa los canales de audio que se almacenan en el GCA.

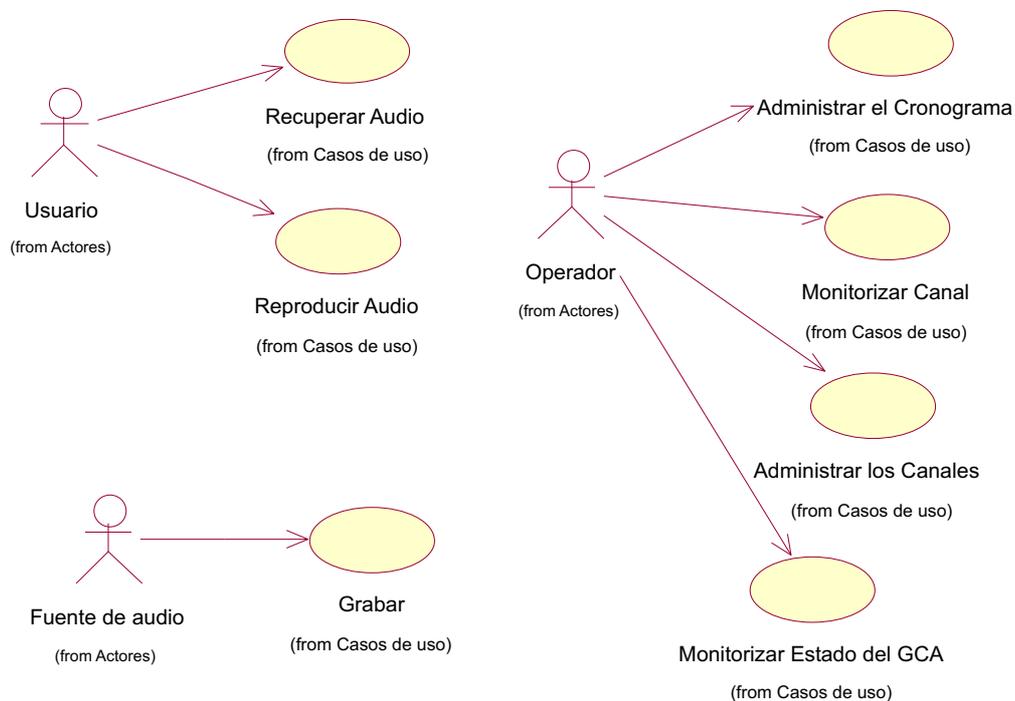


Fig. 1: Diagrama general de casos de uso y actores

3. Funcionalidades comunes a todos los casos de uso

3.1. Registro de errores

Los errores del sistema deben ser registrados. Según la gravedad o el alcance del módulo, errores fatales deben recuperarse, escalarse o resumirse de forma ordenada.

El registro de errores debe incluir una descripción, el módulo en que se detectó el error y la fecha en que se produjo el error.

3.2. Acceso remoto

Todas las funciones que se realicen por personas (actores humanos) deben ser accesibles a través de una conexión de red. Esto requerirá eventualmente la utilización de aplicaciones cliente en las computadoras remotas.

4. Caso de uso Recuperar Audio

4.1. Descripción breve

Este caso de uso permite al usuario recuperar el audio almacenado entre dos instantes en forma de archivo de audio estándar.

4.2. Flujo de eventos

El caso de uso comienza cuando el usuario inicia una interfaz web habilitada para este caso de uso. La interfaz se utiliza para especificar el intervalo a recuperar y el canal.

Flujo básico

1. El usuario define el canal y la fecha/hora inicial.
2. El sistema verifica que exista audio almacenado para el canal y fecha especificados por el usuario. Si existe audio almacenado, el sistema genera una página HTML con la lista de archivos de audio que cumplen con el criterio de consulta.
3. El usuario elige de la lista generada cuales archivos desea guardar y la ubicación destino para la operación.

Flujos alternativos

No existe audio almacenado en el intervalo

Si el servidor no tiene audio almacenado para la fecha y canal seleccionados, el sistema informa al usuario sobre el error detectado.

5. Caso de uso Reproducir Audio

5.1. Descripción breve

Este caso de uso permite al usuario reproducir el audio almacenado a partir de un instante que él especifica.

5.2. Flujo de eventos

El caso de uso comienza cuando el usuario inicia una interfaz web habilitada para este caso de uso. La interfaz se utiliza para especificar el momento a partir del cual se inicia la reproducción y el canal.

Flujo básico

1. El usuario define el canal y la fecha/hora inicial.
2. El sistema verifica que exista audio almacenado para el canal a partir de la fecha/hora especificada. Si existe audio almacenado, el sistema genera una lista de reproducción.
3. El usuario confirma el inicio de la reproducción y también puede indicar que desea pausar o detener la reproducción.

Flujos alternativos

No existe audio almacenado

Si el servidor no tiene audio almacenado posterior a la fecha/hora y para el canal especificados, el sistema informa al usuario sobre el error detectado.

6. Caso de uso Grabar

6.1. Descripción breve

Este caso de uso realiza la grabación del audio en un medio de almacenamiento masivo de acceso aleatorio. El cronograma especifica durante qué períodos y con qué formato de audio se graba.

El sistema permite grabar múltiples canales de audio simultáneamente.

6.2. Flujo de eventos

Las siguientes descripciones de los flujos de eventos se realizan para un canal de los varios que pueden estar grabándose simultáneamente.

El caso de uso comienza cuando el operador habilita un canal correspondiente a una fuente de audio (ver caso de uso Administrar los Canales).

Flujos básicos

Operación Automática

Al habilitar un canal correspondiente a una fuente de audio, el sistema consulta el cronograma del canal y determina si actualmente debe estar iniciada la grabación en cuyo caso inicia la adquisición. Además el sistema determina el próximo cambio de estado para este canal y programa un evento para operar el grabador en ese momento. Los cambios de estado son de tres tipos: inicio de grabación con un formato de audio dado, continuación de la grabación pero con un nuevo formato de audio o detención de la grabación. Estos flujos se describen a continuación.

Grabar

1. El sistema inicia la adquisición del audio proveniente de la fuente de audio con el formato de audio indicado en el cronograma
2. El audio adquirido se procesa para adaptarlo al formato de compresión especificado por el cronograma.
3. El audio procesado se almacena en disco.

Detener grabación El sistema detiene la adquisición y termina de completar la grabación a disco del audio adquirido.

Cambiar formato de audio El sistema termina de completar la grabación a disco del audio adquirido y procede con el flujo Grabar.

Forzar grabación

1. El operador indica que desea que se grabe con un determinado formato de audio que define utilizando la interfaz habilitada para tal fin.
2. El sistema resume el estado actual de grabación e inicia la grabación con el formato indicado. La grabación proseguirá hasta que se llegue al próximo evento programado en el sistema (según lo que indique el cronograma). A partir de ese momento se sigue según el flujo Operación Automática.

Forzar detención

1. El operador indica que desea que se detenga la grabación.
2. El sistema resume el estado actual de grabación y detiene la grabación. La grabación estará detenida hasta que se llegue al próximo evento programado en el sistema (según lo que indique el cronograma). A partir de ese momento se sigue según el flujo Operación Automática.

Flujos alternativos

Disco lleno

Si el sistema detecta que el disco está lleno, la grabación se detiene y el sistema genera una alarma.

Falta de disponibilidad de procesamiento

Si la adquisición, codificación o almacenamiento en disco no se está realizando con el ritmo correcto, el sistema genera una alarma.

7. Caso de uso Administrar el Cronograma**7.1. Descripción breve**

Este caso de uso permite al operador realizar cambios en el cronograma.

7.2. Flujo de eventos

El caso de uso comienza cuando el operador inicia la consola de operación y selecciona el canal y elige la opción Cronograma.

Flujos básicos**Agregar entrada**

1. El sistema muestra una grilla con el cronograma actual.
2. El operador marca el renglón donde desea insertar la entrada de cronograma.
3. El sistema genera un renglón vacío.
4. El operador completa la acción-grabador, la fecha/hora en que debe realizarse y el formato de audio si corresponde.
5. El operador confirma el ingreso de la entrada (el operador puede cancelar el flujo en cuyo caso no se agrega la entrada).
6. El sistema agrega la entrada al cronograma e inicia el flujo alternativo Cambio en el cronograma.

Eliminar entrada

1. El sistema muestra una grilla con el cronograma actual.
2. El operador marca uno o varios renglones del cronograma y selecciona la opción Eliminar.
3. El sistema elimina la entrada del cronograma e inicia el flujo alternativo Cambio en el cronograma.

Flujos alternativos**Cambio en el cronograma**

Si el operador modifica el cronograma de un canal, el sistema verifica si el evento programado se mantiene o debe ser modificado, en cuyo caso cambia la programación del evento de operación automática. El sistema no realiza acciones sobre el estado actual de operación.

8. Caso de uso Monitorizar Canal

8.1. Descripción breve

Este caso de uso permite al operador monitorizar la grabación de los canales.

8.2. Flujo de eventos

El caso de uso comienza cuando el operador inicia la consola de operación y selecciona el canal y elige la opción Monitorizar Canal.

Flujo básico

1. El operador elige el canal a monitorizar (ver el flujo alternativo Elegir canal).
2. El sistema presenta un indicador de saturación, un medidor visual y una medida numérica del nivel del audio que se está reproduciendo.
3. El operador puede forzar el inicio o detención de la grabación.

Flujos alternativos

Modificar nivel de grabación

La ventana presenta un control gráfico de ajuste de nivel que permite modificar el nivel con el que se adquiere el audio de la fuente de audio. Si el operador modifica la posición del control de nivel, el sistema modifica en tiempo real el valor real del mezclador del canal.

Forzar grabación

Ver flujo Grabar.

Forzar detención

Ver flujo Grabar.

9. Caso de uso Administrar los Canales

9.1. Descripción breve

Este caso de uso permite al operador habilitar o deshabilitar los distintos canales disponibles, eliminar canales y agregar nuevos canales.

9.2. Flujo de eventos

El caso de uso comienza cuando el operador inicia la consola de operación y se conecta a un GCA.

Flujos básicos**Iniciar canal**

1. El operador elige el canal de acuerdo al flujo Elegir Canal.
2. El operador elige la opción Iniciar Canal.
3. El sistema inicia el flujo Grabar para el canal especificado.

Terminar canal

1. El operador elige el canal de acuerdo al flujo Elegir Canal.
2. El operador elige la opción Terminar Canal.
3. El sistema resume el estado actual de grabación, detiene la grabación y concluye el flujo Grabar.

Eliminar canal

1. El operador elige el canal de acuerdo al flujo Elegir Canal.
2. El operador elige la opción Eliminar Canal.
3. El operador confirma la eliminación del canal (el operador puede cancelar el flujo en cuyo caso no se elimina el canal).
4. El sistema elimina el canal pero mantiene la información guardada en disco incluyendo el archivo que contiene el cronograma.

Agregar canal

1. El operador elige la opción Agregar Canal.
2. El sistema presenta la lista de campos con los parámetros para definir un nuevo canal.
3. El operador completa la información.
4. El operador confirma el ingreso del nuevo canal (el operador puede cancelar el flujo en cuyo caso no se agrega el canal).
5. El sistema agrega el nuevo canal a la lista de los canales configurados, genera un cronograma predeterminado y deja el canal en estado deshabilitado.

Flujo alternativo**Elegir canal**

1. El operador selecciona la opción Elegir Canal.
2. El sistema recupera y despliega la lista de canales.
3. El operador elige un canal de la lista.
4. El sistema asigna el canal como activo para operaciones de inicio, detención o eliminación del canal, monitorización de estado o cambios de su cronograma.

10. Caso de uso Monitorizar Estado del GCA

10.1. Descripción breve

Este caso de uso permite al operador monitorizar el funcionamiento del GCA y ver la lista de errores activos en caso que existan.

10.2. Flujo de eventos

El caso de uso comienza cuando el operador inicia la consola de operación y se conecta a un GCA.

Flujo básico

1. El operador selecciona la opción Ver Estado.
2. El sistema muestra una ventana donde indica si el funcionamiento del GCA es correcto o hay fallas. Si hay fallas el sistema lista las fallas.

11. Especificación suplementaria

1. Objetivos

El propósito de este documento es definir los requerimientos del GCA. Esta especificación suplementaria lista los requerimientos que no se derivan inmediatamente del documento de especificación de casos de uso. La especificación suplementaria y el documento de casos de uso forman juntos un conjunto completo de los requerimientos del sistema.

2. Alcance

El sistema graba audio proveniente de distintas fuentes de acuerdo a un cronograma. El audio grabado puede accederse de forma aleatoria para su reproducción o recuperación utilizando diferentes interfaces. Esta especificación define requerimientos no funcionales.

3. Usabilidad

3.1. Compatibilidad

El sistema completo debe ser compatible con los sistemas operativos Linux y Microsoft Windows, en sus versiones de 32 bit.

3.2. Diseño

La interfaz de usuario debe ser diseñada para que pueda ser utilizada por un usuario con conocimientos básicos del manejo de la computadora.

3.3. Ayuda al usuario

La ayuda debe incluir: definiciones para los términos y funciones propios al sistema GCA, descripciones de las pantallas e instrucciones paso a paso acerca de cómo realizar las tareas típicas. Cada operación que pueda utilizar el usuario debe disponer de una ayuda o guía.

4. Confiabilidad

4.1. Disponibilidad

El sistema debe operar de forma continua sin fallas propias al código y notificar si existen fallas de funcionamiento.

5. Mantenimiento

5.1. Nuevas versiones

Las actualizaciones de los programas clientes deben poderse descargar a través de la red.

6. Restricciones de diseño

6.1. Requerimientos de la plataforma

La aplicación cliente debe operar en cualquier computadora personal con los siguientes requerimientos mínimos:

- Procesador Intel Pentium II o equivalente
- Memoria RAM de 64 Mbytes
- 40 Mbytes libres en el disco rígido para instalar la aplicación cliente

Adicionalmente se requerirá espacio en disco suficiente para recuperar el audio que solicite el usuario.

12. Lista de riesgos - principio

1. Objetivo

El propósito de este documento es listar los riesgos más significativos del desarrollo de modo de evitar sus consecuencias negativas en una etapa temprana del mismo.

2. Lista de riesgos

Clasificación ¹	Descripción del riesgo e impacto	Estrategia de mitigación o plan de contingencia
10	Un proyecto anterior de la Universidad fracasó en su intento de grabar audio por computadora. El proyecto de la asignatura C++ realizado por nuestro grupo logró grabar audio pero con fallas que producían que el ejecutable provocara errores de violación de segmento.	Se realizará un grabador de audio simple que al ejecutarlo grave un archivo. Este archivo debe poderse reproducir para verificar que la grabación se haya realizado correctamente.
9	El grupo tiene poca experiencia y formación en procesos de desarrollo orientados a objetos. Esto puede conducir a una eficiencia reducida y baja calidad del producto.	Definir un manual de proyecto que guíe el proceso de desarrollo. Determinar la literatura que ilustra los procesos a seguir. Guía, ayuda y seguimiento del tutor.
8	El grupo sólo tiene conocimientos básicos sobre los codificadores/decodificadores de audio disponibles y las interfaces que poseen.	Se realizará un estudio y clasificación de los codificadores / decodificadores de audio de uso más generalizado. Se elegirán los que se consideren adecuados para realizarle pruebas y estudiar sus interfaces.

¹ Los riesgos se clasifican del 1 al 10. 1 es el mínimo y 10 el máximo. La clasificación se basa en cuán crítico es el riesgo y su probabilidad de ocurrencia

3	En proyectos anteriores que el grupo realizó se notaron carencias en la forma en que se administraron las diferentes versiones de los archivos de documentación y de código fuente.	Elegir e implementar un sistema de versiones desde el comienzo.
2	En proyectos anteriores que el grupo realizó, se encontraron defectos y limitaciones del sistema de documentación utilizado que perjudicaron la calidad de la presentación e hicieron bajar la productividad.	Elegir y validar otro sistema de documentación.

13. Guía de estilo del código fuente

1. Objetivo

El propósito de este documento es establecer los lineamientos de estilo del código fuente a seguir durante el desarrollo del GCA de forma resumida y práctica.

2. Referencias

Este documento especializa y resume los lineamientos de la bibliografía revisada ([17] y [18]).

3. Sangría

Para establecer la sangría del código, utilizar tabuladores de 4 caracteres.

4. Ubicación y uso de las llaves

Utilizar llaves aunque sólo se utilice una sentencia dentro de ellas.

Abrir las llaves al final de las líneas y cerrarlas en una línea aparte. Ejemplo:

```
if (x es verdadero) {
    hacer y
}
```

Sin embargo, en el caso especial de las funciones, abrir las llaves en el comienzo de la línea siguiente:

```
int funcion(int x)
{
    cuerpo de la funcion
}
```

Dejar la llave de cierre sola en una línea propia, salvo que se continúe con la misma instrucción:

```
do {
    cuerpo del bucle do
} while (condicion);
if (x == y) {
    ..
} else if (x > y) {
    ..
} else {
    ..
}
```

5. Nomenclatura de variables, funciones, clases, etc.

- Los nombres deben ser descriptivos y preferentemente breves. Por ejemplo no utilizar "de" en ArchivoDeAudio y en cambio usar ArchivoAudio.
- Utilizar minúsculas para los nombres de variables. Separar las palabras con "_".
- Comenzar los nombres de las variables globales con "g_".
- Comenzar los nombres de punteros con "p_".
- Comenzar los nombres de las variables miembro de una clase con "m_". Colocar este prefijo antes que cualquier otro. Ejemplo: m_nombre, m_p_nombre.
- Nombrar las clases, estructuras y enumerados utilizando mayúsculas/minúsculas. Comenzar por una mayúscula y separar palabras con mayúsculas (Ej.: EstaEsUnaClase)
- Nombrar las funciones igual que el caso anterior pero comenzando por minúscula (Ej.: estaEsUnaFuncion).
- Utilizar todas mayúsculas para los nombres de las macros (#define).

6. Funciones

Las funciones deben ser cortas y hacer sólo una cosa. Deben entrar en una o dos pantallas de texto, salvo que se trate de una función simple con por ejemplo una larga instrucción *case*. Separar las tareas más complejas en más funciones y en caso que el rendimiento sea crítico incluir las funciones como *inline*.

Evitar que el número de las variables locales supere la decena.

7. Clases

Ordenar las definiciones de clases como en el siguiente ejemplo:

```
Class ClaseEjemplo {
    int variable_privada1;
    char variable_privada2;
    void funcion_privada1();
public:
    int variable_publica1;
    int variable_publica2;
    Ejemplo();
    Ejemplo~();
    void funcion_publica1();
}
```

8. Comentarios

Incluir comentarios en el código para facilitar su comprensión. Comentar qué es lo que hace el código y a la vez escribirlo de modo que sea fácil de entender.

9. Instrucciones

Utilizar una instrucción por línea.

10. Orden de las secciones de un archivo

Los archivos tienen las secciones ordenadas como se indica a continuación:

1. Prólogo
Ejemplo:

```
/**
@file nombre_archivo.hpp
@short Define la interfaz ....
@date Tue Jul 2 2004
@author M. E. Aguirre, G. Martony (correo-e: gca@cue.com.uy)
*/
```

1. Instrucciones #include
2. Instrucciones #define
3. Definiciones y declaraciones globales
4. Funciones

11. Nombres de archivos

Utilizar las siguientes extensiones de archivos asociadas a los contenidos de los mismos.

Extensión	Contenido del archivo
.c	fuelle en lenguaje C
.cpp	fuelle en lenguaje C++
.h	encabezado en lenguaje C
.hpp	encabezado en lenguaje C++
.idl	encabezado de definición de interfaz CORBA
.java	fuelle en lenguaje Java

Los archivos que contienen el encabezado y la implementación de una clase deben tener el mismo nombre que la clase.

12. Abreviaciones de palabras comunes

Abreviación	Significado
ant	Anterior
arch	Archivo
conf	Configuración
ctrl	Control
disp	Dispositivo
elem	Elemento
est	Establecer valor de propiedad o variable
evt	Evento
id	Identificador
mm	Multimedios
num	Número
obt	Obtener valor de propiedad o variable
pos	Posición
prox	Próximo
reg	Registro
tmp	Temporario/a

13. Prefijos de controles de la interfaz gráfica

Abreviación	Tipo de control	Descripción
b	Button	Botón
cb	Combo Box	Cuadro combinado
et	Static Text	Texto estático
chk	Check Box	Cuadro de verificación
ct	Text	Texto
g	Gauge	Barra
opc	Menu	Opción de menú

14. Lista de riesgos - elaboración

1. Objetivo

El propósito de este documento es listar los riesgos más significativos del desarrollo de modo de evitar sus consecuencias negativas en una etapa temprana de la construcción.

2. Lista de riesgos

Clasificación ¹	Descripción del riesgo e impacto	Estrategia de mitigación o plan de contingencia
10	Los métodos de desarrollo de software pueden resultar difíciles y engorrosos de aplicar, probablemente por falta de experiencia y/o conocimientos del grupo. Esto provoca pérdida de motivación.	Utilizar un enfoque más práctico. Utilizar la parte de los métodos que se comprenda mejor y muestre utilidad clara. Realizar experimentos de implementación de pequeñas partes del sistema.

¹ Los riesgos se clasifican del 1 al 10. 1 es el mínimo y 10 el máximo. La clasificación se basa en cuan crítico es el riesgo y su probabilidad de ocurrencia

15. Documento de diseño

1. Tecnología y bibliotecas utilizadas

1.1. Lenguajes de programación utilizados

Se utiliza el lenguaje C++ para todos los desarrollos salvo para la interfaz web para la cual se utiliza PHP. Se utiliza como referencia del lenguaje C++ la publicación [19].

1.2. Uso de CORBA

El sistema a desarrollar es distribuido tanto en cuanto a los usuarios como a la computación y los datos. CORBA provee un marco de desarrollo de aplicaciones distribuidas orientadas a objetos que es independiente de la plataforma y del lenguaje de programación. Habilita la comunicación entre objetos existentes en distintos procesos o subprocesos del sistema de forma transparente a la red de comunicaciones.

1.3. Bibliotecas utilizadas

El la figura siguiente se ilustran las dependencias de bibliotecas externas.

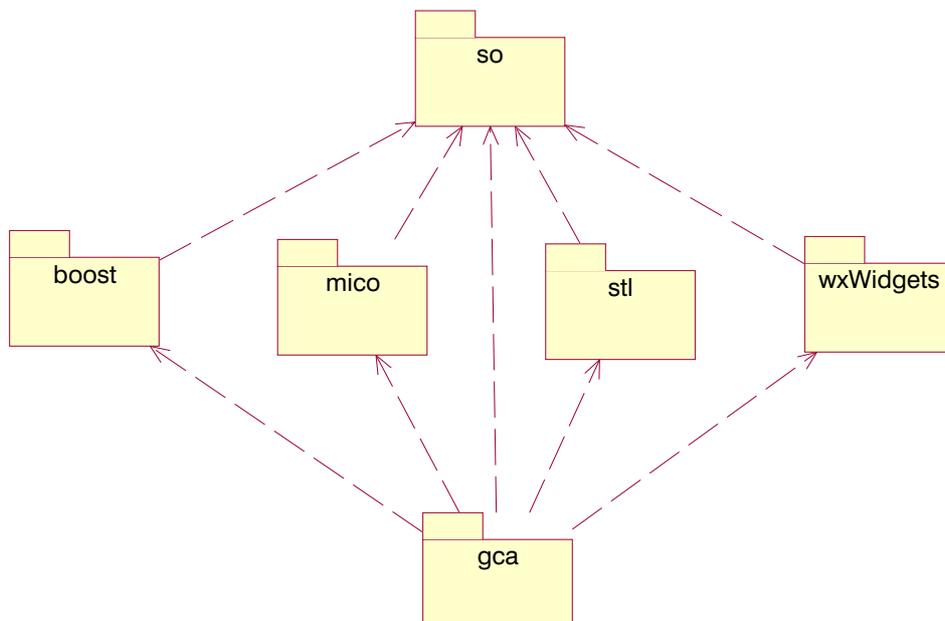


Fig. 1: Dependencia de bibliotecas externas

Sistema operativo

Se utilizan llamadas directas al sistema operativo para controlar los dispositivos de adquisición, mezcla de audio y lanzamiento de procesos.

Biblioteca estándar de plantillas (STL)

Se utilizan las clases de la biblioteca STL a lo largo de todo el sistema. Se destacan las clases `std::string`, `std::deque` y `std::vector`.

Biblioteca Boost

Para hacer la programación más eficiente, portable y segura se utilizan las bibliotecas de código abierto Boost, que proveen clases de objetos de sincronización de tareas, subprocesos, fecha/hora y persistencia.

Se utilizan las siguientes bibliotecas de Boost:

Clase	Descripción
<code>date_time</code>	Define clases para manejo de fechas y horas.
<code>thread</code>	Encapsula los métodos genéricos de subprocesos para los distintos sistemas operativos disponibles. Provee métodos para iniciar y detener subprocesos de forma controlada. Incluye clases para realizar exclusión mutua entre subprocesos y otras clases de sincronización.
<code>filesystem</code>	Manejo del sistema de archivos
<code>serialization</code>	Provee las funciones necesarias para realizar la persistencia de las clases.
<code>bind</code>	Asociadores genéricos de funciones, objetos y punteros.

Biblioteca Mico

Se utiliza la implementación MICO 2.3.11 para C++ como entorno de CORBA.

Biblioteca wxWidgets

Esta biblioteca es un marco para desarrollar interfaces gráficas que utiliza los controles nativos de cada sistema operativo.

2. Diseño

2.1. Perspectiva general del sistema

La organización lógica de los paquetes de clases del sistema se ilustra en la figura siguiente.

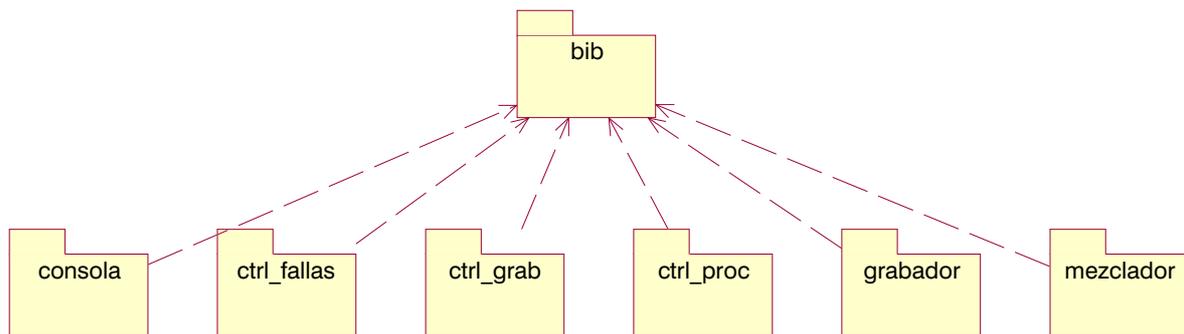


Fig. 2: Vista lógica global

Para facilitar la comprensión del sistema se muestra un ejemplo de despliegue en la figura 4. El sistema consta de:

- Un servidor que contiene el servidor de nombres de CORBA, el servidor web y opcionalmente servidores para compartir archivos y de streaming.
- Computadoras grabadoras encargadas de realizar la grabación de múltiples canales.
- Computadora cliente con una consola de operación y mantenimiento y clientes con interfaz gráfica y de línea de comando.
- Computadora con un navegador de Internet para acceder al sistema a través de su interfaz web.
- Computadora de respaldo que contiene el servicio de Control de Fallas y lanza una grabadora al detectarse una condición de error.

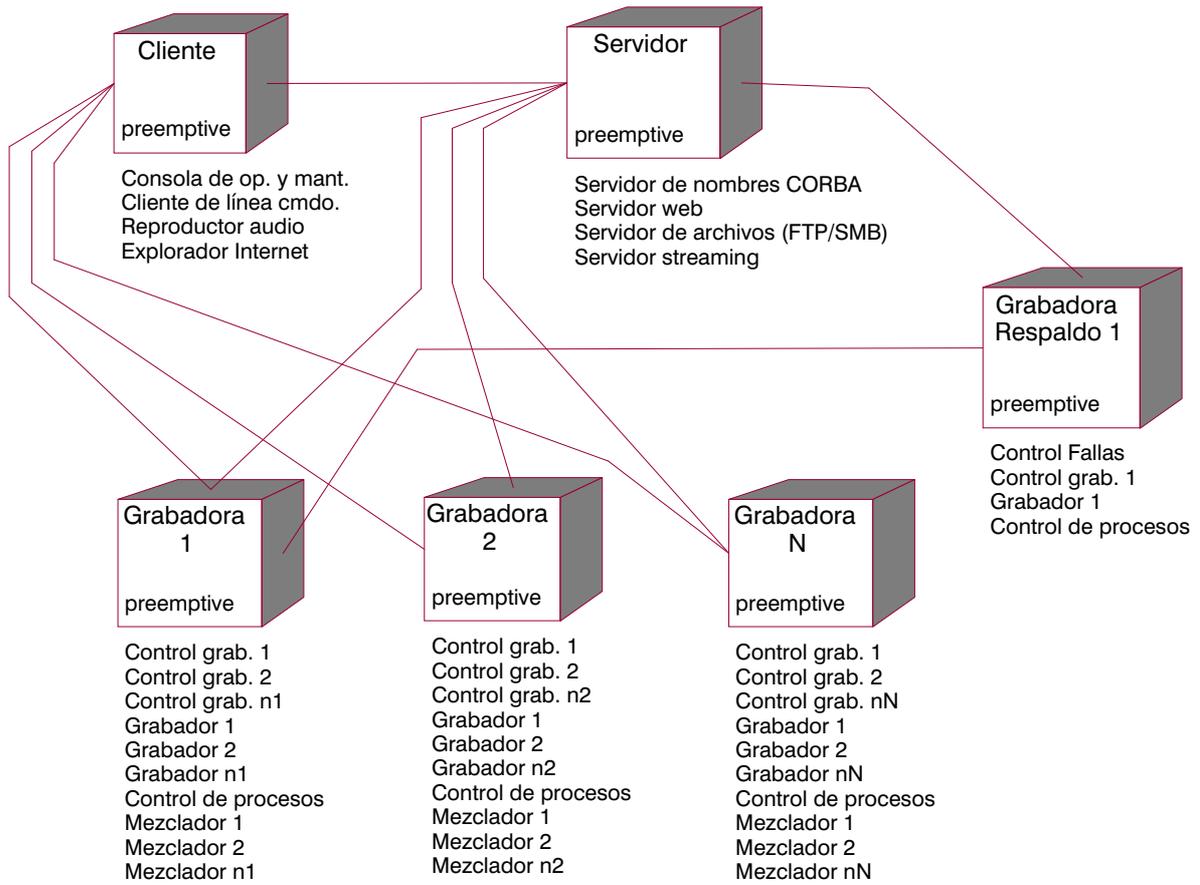


Fig. 3: Diagrama de despliegue

2.2. Biblioteca de clases comunes

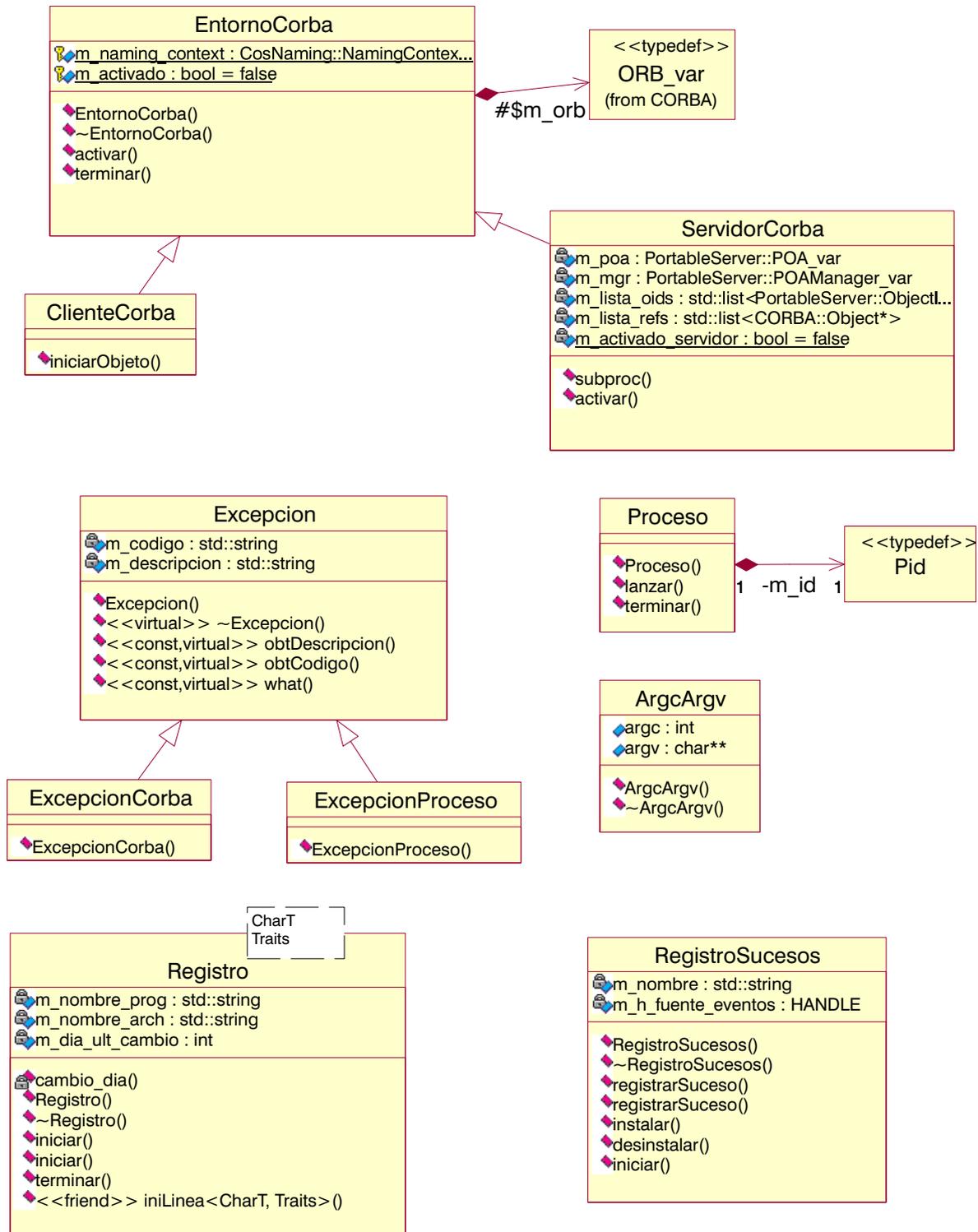


Fig. 4: Diagrama de clases de la biblioteca

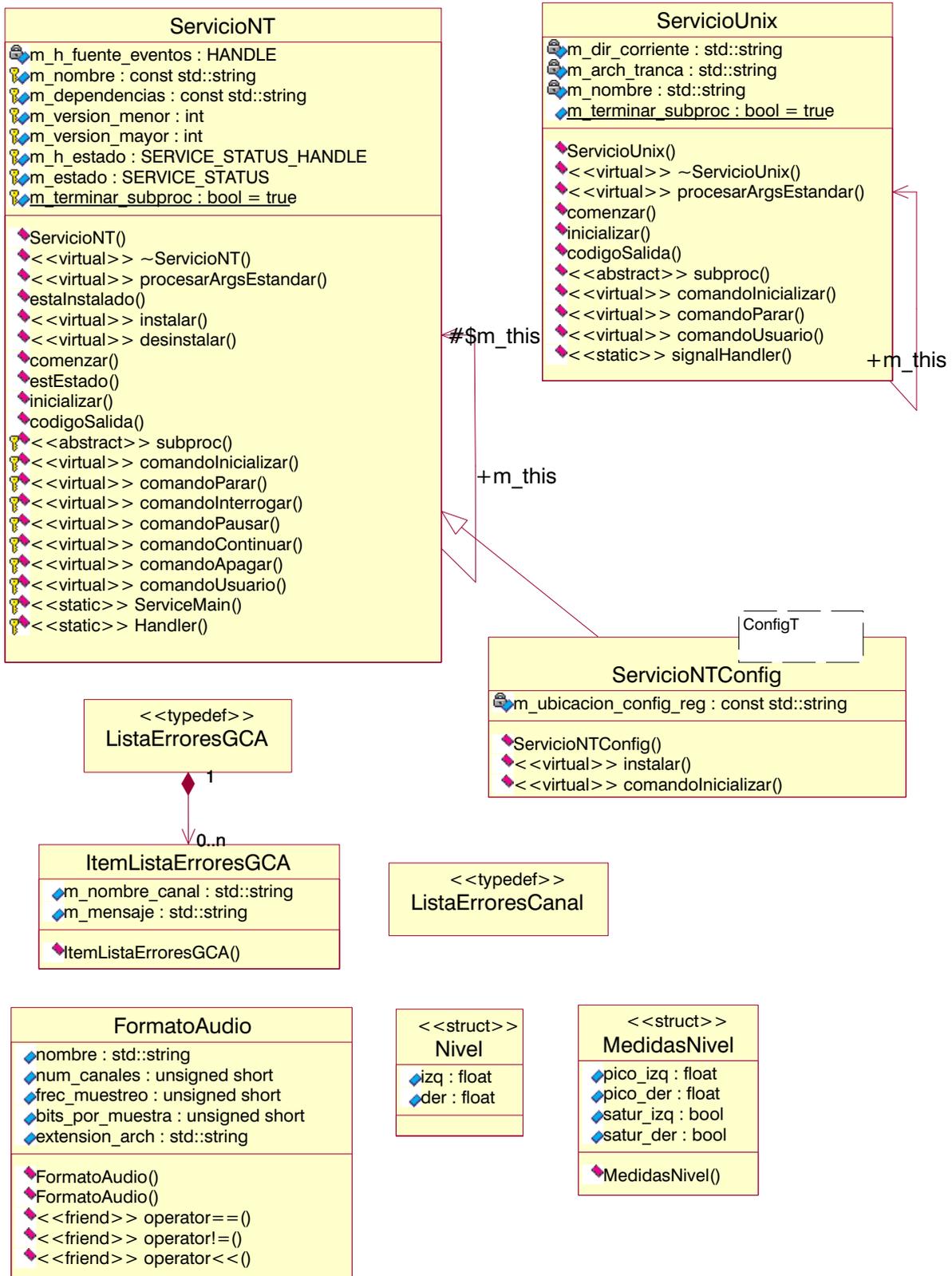


Fig. 5: Diagrama de clases de la biblioteca (continuación)

En este paquete se agrupa una biblioteca estática de clases que se utilizan en los otros módulos. Se detallan a continuación las clases principales.

Comunicación CORBA

La clase `EntornoCorba` contiene referencias a objetos comunes tanto a un servidor como a un cliente CORBA. Mantiene una referencia al servidor de nombres.

La clase `ClienteCorba` deriva de `EntornoCorba` y permite conectarse y obtener la referencia a un servidor CORBA.

La clase `ServidorCorba` deriva de `EntornoCorba`, registra la interfaz del servidor con el servidor de nombres y recibe las solicitudes de los clientes.

Interfaces CORBA

Los módulos que presentan un servidor CORBA ubican en esta biblioteca las clases generadas a partir del archivo de definiciones IDL.

Manejo de excepciones

La clase `Excepcion` es la clase base para las excepciones, deriva de la clase `std::exception` de la biblioteca STL y almacena un código y una descripción. De esta clase se derivan las clases específicas para el manejo de excepciones relativas al dispositivo de audio, almacenamiento en disco, etc.

Registro de Sucesos

La clase `RegistroSucesos` se instancia como una variable global para ser utilizada por las distintas clases de un proceso. Implementa métodos para el registro de mensajes en el sistema operativo. Para encapsular las diferencias de los sistemas operativos, se utilizan directivas de compilación. Interactúa con el registro de sucesos de Microsoft o con el servicio Syslog.

Registro de depuración

La clase `Registro` se instancia como una variable global para ser utilizada por las distintas clases de un proceso. Genera archivos por fecha donde se guardan los mensajes de depuración que se requieran. Contiene directivas para el compilador de modo tal que en la versión optimizada no realiza ninguna acción.

Servicios del sistema

La clase `ServicioNT` interactúa con el controlador de servicios de Microsoft (SCM, Service Control Manager) para instalar o desinstalar el ejecutable como un servicio. Además procesa los mensajes que envía el sistema operativo para actuar sobre el servicio (iniciarlo, detenerlo, interrogarlo, etc.). La clase `ServicioNTConfig` deriva de `ServicioNT` y agrega la capacidad de guardar y consultar atributos en el registro de Windows.

La clase `ServicioUnix` desprende el proceso que se desea ejecutar como servicio del proceso padre, luego cierra los descriptores y programa los manejadores que se utilizarán para procesar las señales que envía el sistema operativo para actuar sobre el servicio.

Lanzamiento de procesos

La clase Proceso se encarga de crear y terminar procesos. Para cada sistema operativo, encapsula las llamadas necesarias.

Atributos de audio

La clase FormatoAudio contiene la lista de atributos que se utilizan para programar el dispositivo de audio con una determinada calidad de adquisición e implementa la persistencia de los mismos.

2.3. Grabador

El grabador es el responsable de adquirir las muestras de la fuente de audio, procesarlas y almacenarlas en el disco. Es el elemento más crítico en cuanto a la confiabilidad del sistema. Se debe evitar la pérdida de muestras de audio aún cuando el procesador o el disco del sistema se encuentren por momentos muy cargados y no alcancen a mantener la capacidad mínima para procesar el flujo de audio que se adquiere. En consecuencia, el diseño adoptado divide el proceso de grabación en tres subprocesos según indica la figura 6: 1) la adquisición, que debe tener la máxima prioridad para evitar la pérdida de muestras, 2) un subproceso de codificación para análisis del audio y 3) la grabación, que se encarga de guardar el audio en el disco¹. Las colas de adquisición y de grabación son de longitud variable. La cola de adquisición crece cuando el procesador está muy exigido por procesamiento en el flujo intermedio y la cola de grabación crece cuando el disco está exigido.

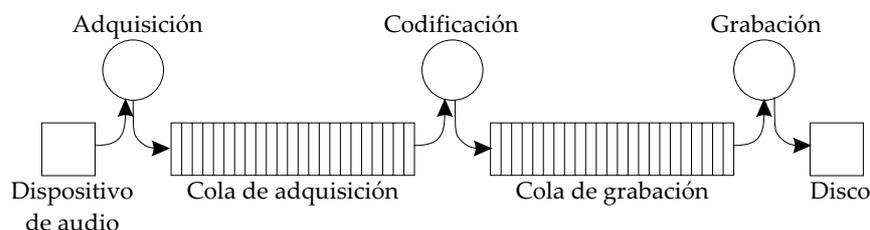
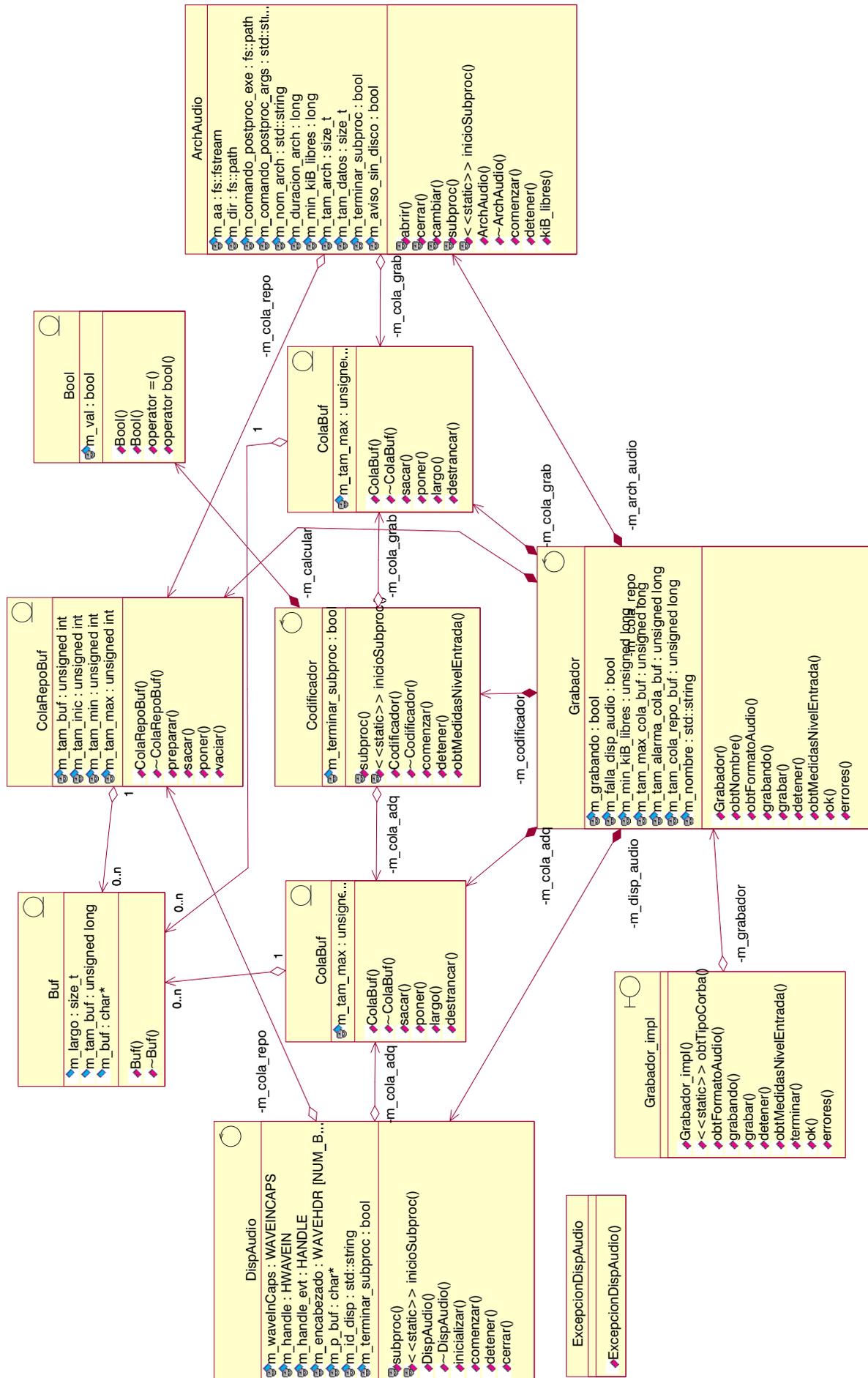


Fig. 6: Esquema del grabador

En la figura siguiente se puede observar el diagrama de clases del grabador.

¹ Más generalmente, el almacenamiento puede realizarse en disco o en cualquier medio persistente masivo de acceso aleatorio.



La clase Grabador crea y contiene a las demás clases. Inicia las colas de búfers y el repositorio. Además, implementa las funciones de la interfaz CORBA. De acuerdo a las instrucciones recibidas por CORBA, el Grabador inicia o detiene el funcionamiento de los sub-procesos de los objetos de tipo DispAudio, Codificador y ArchAudio.

DispAudio permanece en un bucle que solicita búfers al repositorio, los completa con lo que lee del dispositivo de audio y los coloca en la cola de adquisición. Este sub-proceso eleva su prioridad de ejecución para reducir la probabilidad de pérdida de muestras.

Codificador también permanece en un bucle esperando que se coloque un búfer en la cola de adquisición. Retira el búfer, lo procesa y lo coloca en la cola de grabación. El proceso consiste en calcular los valores máximos y verificar si hay saturación. Esta información se utiliza para monitorizar el audio.

ArchAudio, en su bucle de proceso, espera que se coloque un búfer en la cola de adquisición, lo retira, guarda en disco la información contenida en él y lo retorna al repositorio. Al comenzar, abre un archivo en disco con la fecha y hora corriente. El archivo se cierra ante un evento de detención de grabación o cuando se alcanza un tamaño que se calcula a partir de la duración configurada para el canal. Al cerrar el archivo, se cambia el nombre anexando la fecha y hora corriente y, si está configurado, se lanza el proceso post-grabación.

2.4. Control del Grabación y Cronograma

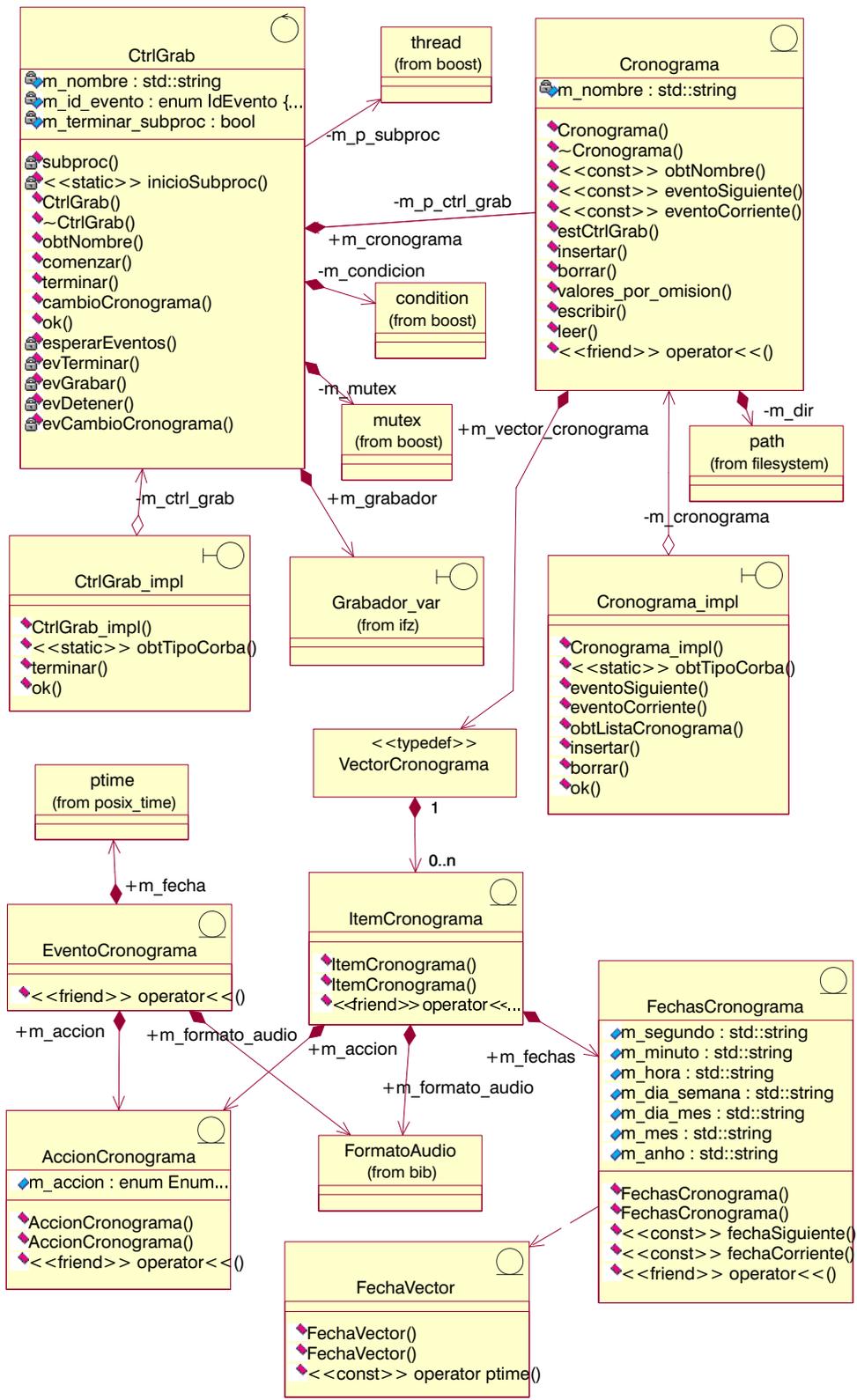


Fig. 8: Diagrama de clases del Control de Grabación y Cronograma

Cuando se inicia el proceso de Control de Grabación se construyen los objetos de tipo Cronograma y CtrlGrab con referencias cruzadas para comunicarse entre sí y se inician las interfaces CORBA respectivas.

La clase Cronograma es persistente y guarda en disco un archivo de formato XML con las entradas del cronograma del canal. Al construirse, lee del disco la información para crear la lista de entradas de cronograma. Tiene métodos para calcular, dada una fecha corriente, el estado en que debería estar el grabador y la fecha del próximo cambio de estado. Como las entradas de cronograma pueden tener valores libres para la fecha o la hora, el cronograma debe resolver el cálculo de fechas reales a partir de las entradas.

CtrlGrab tiene una variable de condición que se utiliza para dejar en espera el bucle de proceso hasta que ocurra un evento. Al comenzar, CtrlGrab consulta al Cronograma en qué estado debería estar el Grabador y lo comanda por CORBA. Luego consulta cuál es la fecha del próximo cambio de estado y programa un temporizador para la variable de condición. El bucle de proceso queda esperando la notificación de la variable de condición. Al transcurrir el tiempo programado, la variable de condición se libera, CtrlGrab comanda nuevamente al Grabador y vuelve a consultar al Cronograma el próximo evento para programar el temporizador. El Cronograma también puede notificar la variable de condición cuando es modificado, lo que causa que CtrlGrab vuelva a programar el próximo evento.

El algoritmo del Cronograma que localiza el evento corriente dada una fecha, recorre todas las entradas calculando para cada una cuál es la fecha más reciente que le corresponde y es anterior a la fecha dada. De todas las fechas así obtenidas elige la entrada que devuelva la fecha más cercana a la fecha dada. Para calcular la fecha de cada entrada, se sustituye para cada una de las posiciones libres el respectivo valor que tiene la fecha dada. En caso de obtener una fecha válida y menor a la fecha dada, se sustituyen las posiciones libres por el máximo posible verificando en cada caso si la fecha obtenida sigue siendo anterior a la fecha dada; en caso contrario se mantiene el valor anterior de la posición. Si al sustituir las posiciones libres por los valores de la fecha dada como corriente, se obtiene una fecha posterior, entonces se recorren las posiciones libres comenzando por la menos significativa, reduciendo su valor en una unidad y comparando si la fecha así obtenida es anterior; si es anterior a la fecha dada se mantiene el nuevo valor y si es posterior se establece al máximo posible. Finalmente, en caso que la entrada tenga establecido un día de semana, se revisa la coincidencia y eventualmente se busca hacia atrás en el tiempo cuál fecha coincide con el día de semana establecido.

El algoritmo del Cronograma que localiza el próximo evento dada una fecha es similar al que calcula el evento corriente pero con la lógica inversa. Este algoritmo busca hacia adelante y emplea mínimos en lugar de máximos para las posiciones libres.

2.5. Mezclador

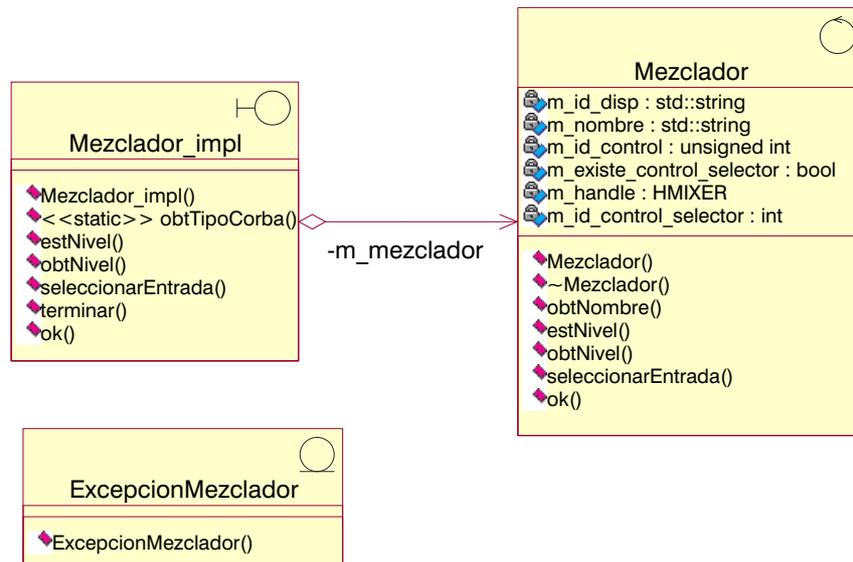


Fig. 9: Diagrama de clases del Mezclador

La clase `Mezclador` permite seleccionar un control de un mezclador y modificar su valor.

2.6. Control de procesos

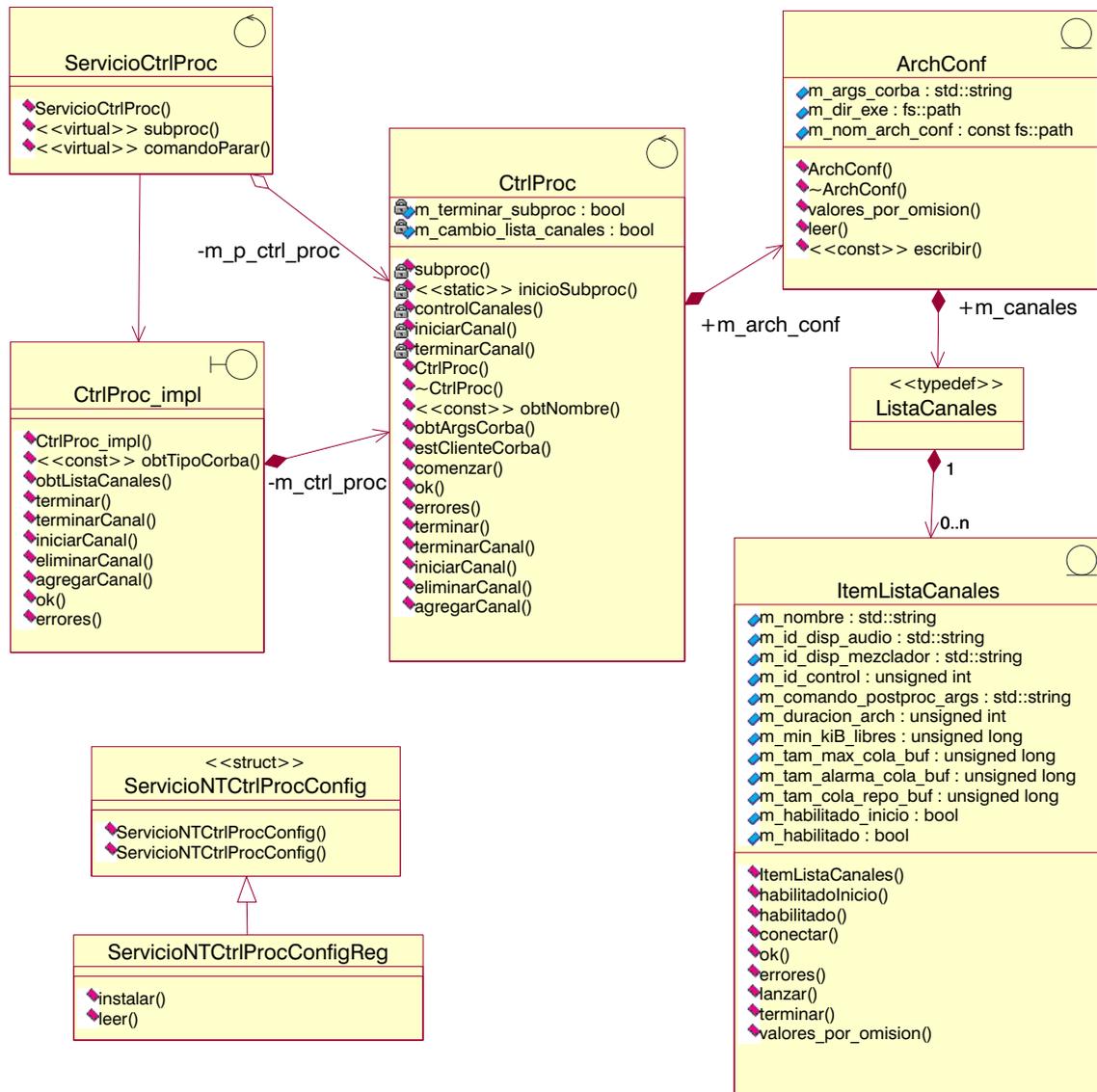


Fig. 10: Diagrama de clases del Control de Procesos

El Control de Procesos utiliza la clase ServicioCtrlProc que es derivada de la clase Servicio de la biblioteca de clases comunes. ServicioCtrlProc se encarga de ejecutarse como un servicio, registrar la interfaz CORBA del control de procesos y comenzar el proceso de la clase CtrlProc que permanecerá en un bucle hasta que se detenga el servicio.

Durante la construcción del objeto de tipo CtrlProc se crea el objeto de tipo ArchConf que implementa la persistencia del archivo de configuraciones. Este archivo contiene la lista de los canales configurados. Cuando se construye, ArchConf lee del archivo de configuraciones y construye el objeto de tipo ListaCanales. Además de mantener la lista de canales, el objeto ListaCanales tiene métodos

para manejar los procesos de un canal y conectarse usando CORBA. ListaCanales usa la clase Proceso de la biblioteca de clases comunes para lanzar y terminar el grabador, control de grabación y mezclador de cada canal.

Al comenzar, CtrlProc inicia los canales que están configurados para iniciarse automáticamente y se mantiene en un bucle donde consulta por CORBA el estado de cada canal. Si se detectan errores se registran en una lista y en el registro de sucesos del sistema usando para ello los métodos provistos por la clase RegistroSucesos.

La interfaz CORBA de CtrlProc permite modificar la lista de canales, iniciar, terminar canales y consultar el estado del sistema obteniendo la lista de errores.

2.7. Control de fallas

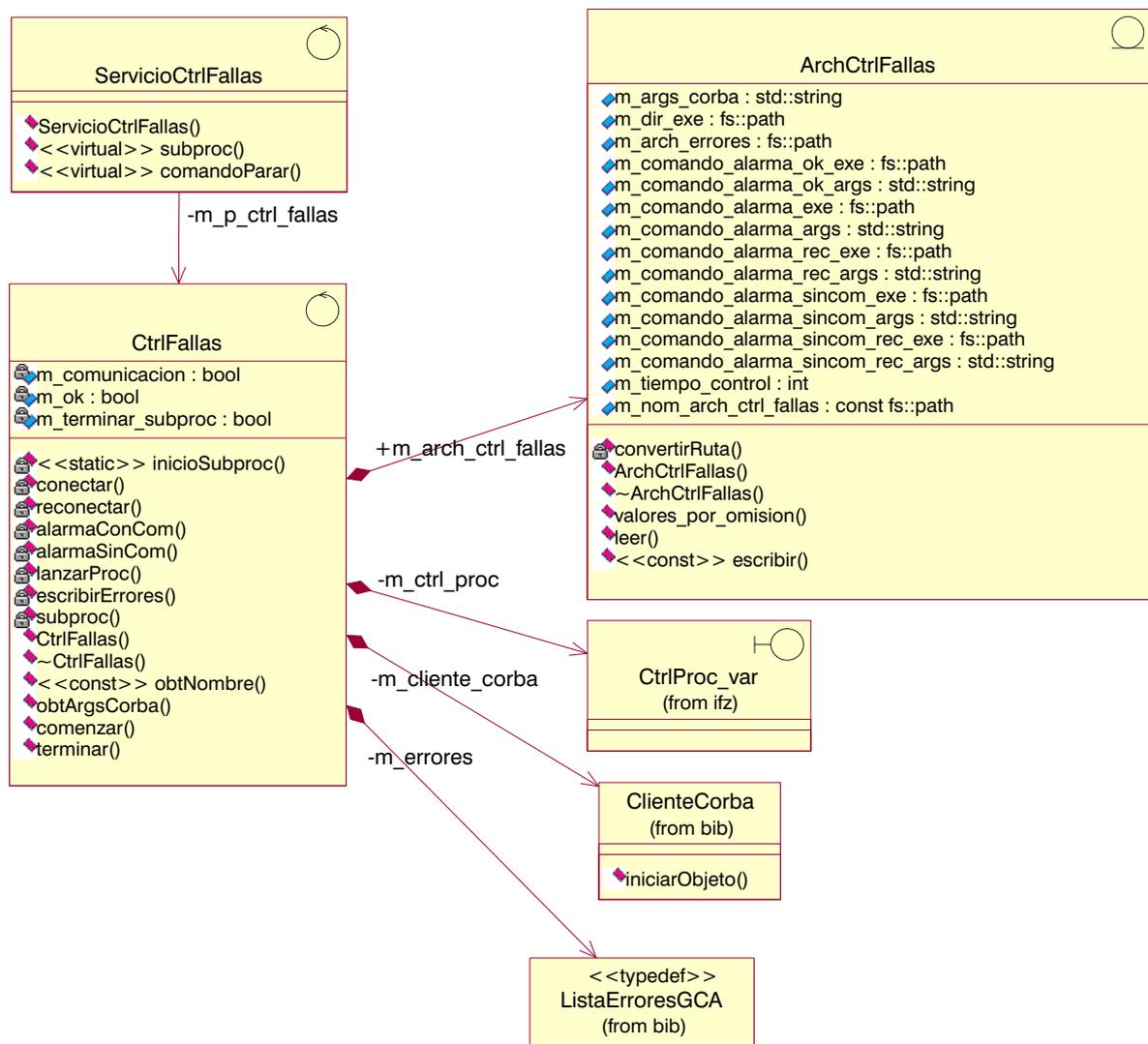


Fig. 11: Diagrama de clases del Control de Fallas

El Control de Fallas utiliza la clase `ServicioCtrlFallas` que es derivada de la clase `Servicio` de la biblioteca de clases comunes. `ServicioCtrlFallas` se encarga de ejecutarse como un servicio y comenzar el proceso de la clase `CtrlFallas` que permanecerá en un bucle hasta que se detenga el servicio.

Durante la construcción del objeto de tipo `CtrlFallas` se crea el objeto de tipo `ArchCtrlFallas` que implementa la persistencia del archivo de configuraciones del control de fallas. Este archivo contiene los comandos y argumentos que definen los procesos a lanzar ante las condiciones de error que se detecten.

`CtrlFallas` se mantiene en un bucle consultando periódicamente por CORBA a `CtrlProc` si el estado de funcionamiento es correcto. Si la consulta reporta fallas, `CtrlFallas` solicita la lista de errores y guarda un archivo de texto plano en disco. Además lanza un proceso según se especifique en el archivo de configuraciones. Se puede configurar un proceso de ejecución única y otro recurrente. De este modo, `CtrlFallas` lanza el proceso de ejecución única ante un error de funcionamiento y si en las sucesivas consultas el mismo error se mantiene, lanza el proceso de ejecución recurrente. En caso que exista un error de comunicación y no se pueda acceder por CORBA a `CtrlProc`, se lanza un proceso diferente de ejecución única y otro recurrente. Al restaurarse la comunicación y correcto funcionamiento, `CtrlFallas` lanza de forma única un proceso de notificación de funcionamiento normal. Alguno de los procesos descritos pueden no estar definidos en el archivo de configuraciones.

2.8. Consola de operación y monitorización

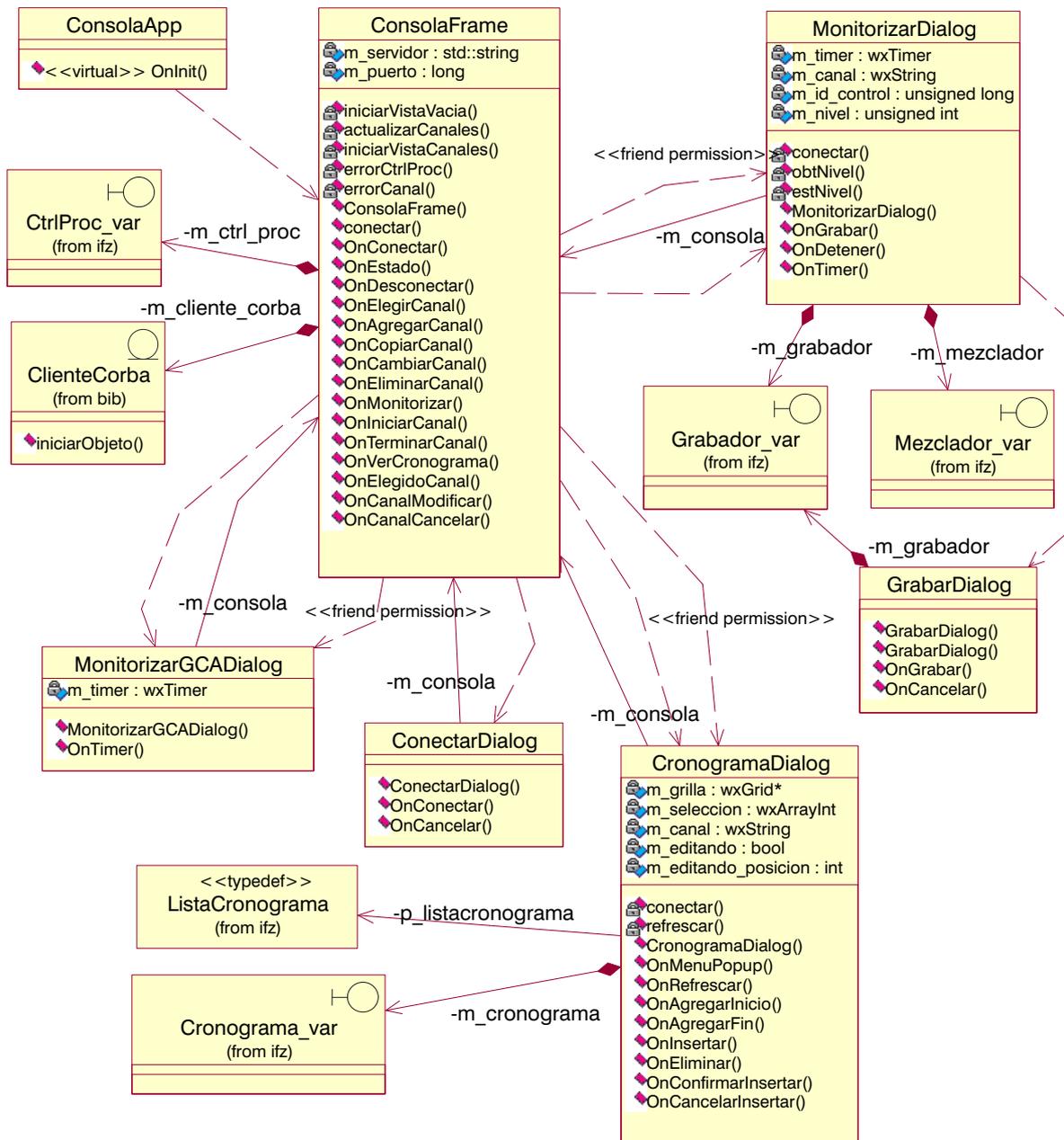


Fig. 12: Diagrama de clases de la Consola

La Consola es una aplicación gráfica que utiliza y deriva clases de la biblioteca wxWidgets. El proceso principal comienza implementando la clase `ConsolaApp` que es una clase que se deriva de `wxApp`, esto causa un llamado al método `OnInit()` que carga en memoria los objetos gráficos que se utilizarán. A partir de este punto el sistema funciona en base a eventos.

El conjunto de ventanas, cuadros de diálogos, menú y controles que se utilizan en la Consola están

definidos en un archivo externo de formato similar al XML que es un archivo de definición XRC. La biblioteca wxWidgets interpreta el contenido del archivo XRC y de esta manera crea los objetos en memoria. Los métodos propios a la consola que realizan las diferentes tareas están implementados en las clases derivadas de las clases wxFrame y wxDialog y que son específicas de la Consola. De este modo, por ejemplo, la clase ConsolaFrame (derivada de wxFrame) implementa los métodos para cada opción del menú y para cada acción que debe iniciarse al apretar los botones de la interfaz gráfica principal. Asimismo, la clase MonitorizarDialog (derivada de wxDialog) implementa los métodos del temporizador para mostrar en tiempo real los niveles del audio y los métodos para iniciar y detener la grabación de forma manual cuando el operador aprieta los botones respectivos.

La biblioteca wxWidgets trabaja con eventos vinculados a los objetos de modo similar a la clásica operación de Microsoft Visual Basic. De este modo, cada control es capaz de activar eventos que se vinculan, según el control que lo dispara, a distintos métodos de las clases derivadas de los cuadros de diálogo o ventanas de wxWidgets. En la Consola se utiliza frecuentemente el evento "EVT_BUTTON" que se activa al apretar un botón o el evento "EVT_MENU" que se activa cuando el operador elige una opción del menú.

La Consola se comunica mediante CORBA con la interfaz de CtrlProc para obtener la lista de los canales, administrarlos, configurarlos y monitorizar el estado del sistema. Cuando el operador elige un canal y selecciona la opción de monitorización del canal, la Consola se comunica con la interfaz CORBA de Grabador del canal elegido para consultar los niveles de audio y con la interfaz Mezclador para obtener y modificar el nivel del mezclador. Asimismo, si el operador confirma un comando manual de inicio o detención del grabador, la Consola se comunica directamente con la interfaz de Grabador del canal elegido. Finalmente para la consulta y modificación del cronograma, la Consola accede a la interfaz CORBA de Cronograma y completa una grilla con la lista de las entradas.

16. Descripción de interfaces CORBA

1. Control de procesos

1.1. Propósito

El Control de Procesos se registra en el Servicio de Nombres de CORBA, utilizando el identificador “gcactrlproc”. Esta interfaz permite administrar los canales y monitorizar el estado del GCA.

1.2. Métodos

obtListaCanales()

Este método no toma parámetros y devuelve la lista de los canales definidos en el GCA usando el tipo de datos ListaCanales.

terminarCanal()

Este método corresponde al flujo Terminar Canal y toma como parámetro el nombre del canal. El sistema cierra de forma ordenada los módulos relativos al canal y configura el atributo habilitado del canal al valor falso.

iniciarCanal()

Este método corresponde al flujo Iniciar Canal y toma como parámetro el nombre del canal. El sistema inicia los módulos relativos al canal y configura el atributo habilitado del canal al valor verdadero y comienza el caso de uso Grabar.

eliminarCanal()

Este método corresponde al flujo Eliminar Canal y toma como parámetro el nombre del canal. El sistema verifica que el canal no esté habilitado y elimina las definiciones relativas a él en el archivo de configuraciones y en memoria. Si el canal está habilitado el sistema no realiza acciones.

agregarCanal()

Este método corresponde al flujo Agregar Canal y toma como parámetro el canal a ser agregado usando el tipo de datos ItemListaCanales. El sistema agrega las definiciones relativas al nuevo canal en el archivo de configuraciones y en memoria y configura el atributo habilitado del canal al valor falso.

ok()

Este método no toma parámetros y devuelve verdadero si el GCA está funcionando correctamente o falso si existen fallas.

errores()

Este método no toma parámetros y devuelve la lista de errores actuales que tiene el GCA usando el tipo de datos ListaErroresGCA.

1.3. Tipos de datos**ItemListaErroresCGA**

Es una estructura con los atributos: nombre del canal y mensaje de error.

ListaErroresGCA

Es una secuencia simple de errores definidos según el tipo ItemListaErroresCGA.

ItemListaCanales

Es una estructura con los atributos: nombre del canal, descriptor del dispositivo de audio (por ejemplo /dev/dsp), descriptor del dispositivo mezclador asociado al dispositivo de audio anterior (por ejemplo /dev/mixer), ruta completa en el sistema de archivos donde se guarda el audio adquirido, ruta completa en el sistema de archivos donde se encuentra el archivo ejecutable a utilizar cada vez que se termina de guardar un archivo de audio en formato WAV, argumento a enviar junto con el archivo ejecutable, duración de cada archivo de audio, mínimo espacio libre de disco necesario para que la grabación prosiga para este canal (en kiB), tamaño máximo en búfers de las colas de adquisición y grabación, tamaño en búfers para dar aviso de falla de estas colas, tamaño en búfers para el repositorio de las colas, estado predeterminado del canal cuando inicia el GCA y estado actual del canal.

ListaCanales

Es una secuencia simple de canales definidos según el tipo ItemListaCanales.

1.4. Archivo de descripción de la interfaz (IDL)

```
module ifz {
    struct ItemListaErroresGCA {
        string nombre_canal;
        string mensaje;
    };
    typedef sequence<ItemListaErroresGCA> ListaErroresGCA;
    struct ItemListaCanales {
        string nombre;
        string id_disp_audio;
        string id_disp_mezclador;
        string dir;
    };
};
```

```

    string comando_postproc_exe;
    string comando_postproc_args;
    unsigned long duracion_arch;
    unsigned long duracion_arch;
    unsigned long min_kiB_libres;
    unsigned long tam_max_cola_buf;
    unsigned long tam_alarma_cola_buf;
    unsigned long tam_cola_repo_buf;
    boolean habilitado_inicio;
    boolean habilitado;
};
typedef sequence<ItemListaCanales> ListaCanales;
interface CtrlProc {
    ListaCanales obtListaCanales();
    void terminarCanal(in string canal);
    void iniciarCanal(in string canal);
    void eliminarCanal(in string canal);
    void agregarCanal(in ItemListaCanales itemcanal);
    boolean ok();
    ListaErroresGCA errores();
};
};

```

2. Control de Grabación

2.1. Propósito

Para cada canal habilitado el sistema registra en el Servicio de Nombres de CORBA un servidor de Control de Grabación con identificador coincidente con el nombre del canal. Esta interfaz permite terminar la operación del módulo de Control de Grabación o consultarle si su estado de operación es correcto.

2.2. Métodos

terminar()

Este método no toma parámetros y termina el proceso del Control de Grabación.

ok()

Este método no toma parámetros y devuelve el estado del Control de Grabación. Este método permite verificar que tanto la interfaz CORBA como el proceso de Control de Grabación están respondiendo correctamente.

2.3. Archivo de descripción de la interfaz (IDL)

```

module ifz {
    interface CtrlGrab {

```

```
        void terminar();  
        boolean ok();  
    };  
};
```

3. Grabador

3.1. Propósito

Para cada canal habilitado el sistema registra en el Servicio de Nombres de CORBA un servidor de Grabador con identificador coincidente con el nombre del canal. Esta interfaz permite iniciar o detener la grabación, consultar el estado de grabación, obtener los niveles actuales del audio y consultar el funcionamiento del grabador.

3.2. Métodos

obtFormatoAudio()

Este método no toma parámetros y devuelve el formato de audio que se está empleando actualmente para la grabación usando el tipo de datos FormatoAudio.

grabando()

Este método no toma parámetros y devuelve verdadero si se está adquiriendo audio o falso si la adquisición está detenida.

grabar()

Este método toma como parámetro de entrada un formato de audio según el tipo de datos FormatoAudio. El método instruye al Grabador iniciar la adquisición y grabación según el formato especificado.

detener()

Este método no toma parámetros e instruye al Grabador detener la adquisición y grabación a disco.

obtMedidasNivelEntrada()

Este método no toma parámetros y devuelve los niveles actuales del audio adquirido usando el tipo de datos MedidasNivel.

terminar()

Este método no toma parámetros y termina el proceso del Grabador.

ok()

Este método no toma parámetros y devuelve el estado del Grabador. Este método permite verificar que tanto la interfaz CORBA como el proceso Grabador están respondiendo correctamente. Se devuelve falso en el caso en que exista una falla de inicialización en el dispositivo de audio, se supere la cuota de disco definida o las colas de adquisición y grabación sean excesivas lo que indica que faltan recursos de procesamiento.

errores()

Este método no toma parámetros y devuelve la lista de errores actuales que tiene el Grabador usando el tipo de datos ListaErrores.

3.3. Tipos de datos

Ver Tipos de datos compartidos.

3.4. Archivo de descripción de la interfaz (IDL)

```
#include "tipos_datos_ifz.idl"
module ifz {
    interface Grabador {
        FormatoAudio obtFormatoAudio();
        boolean grabando();
        void grabar(in FormatoAudio f);
        void detener();
        MedidasNivel obtMedidasNivelEntrada();
        void terminar();
        boolean ok();
        ListaErrores errores();
    };
};
```

4. Cronograma**4.1. Propósito**

Para cada canal habilitado el sistema registra en el Servicio de Nombres de CORBA un servidor de Cronograma con identificador coincidente con el nombre del canal. Esta interfaz permite obtener y modificar el cronograma y consultar el funcionamiento del servidor de Cronograma.

4.2. Métodos**obtListaCronograma()**

Este método no toma parámetros y devuelve la lista de las entradas que definen el cronograma usando el tipo de datos ListaCronograma.

eventoSiguiente()

Este método toma como parámetro una cadena que representa una fecha-hora y una entrada de cronograma usando el tipo de datos EventoCronograma. Este método devuelve verdadero y la entrada de cronograma modificada o devuelve falso. La cadena de fecha-hora debe cumplir con el formato ISO YYYYMMDDTHHMMSS (por ejemplo 20050131T235930 representa el 31 de enero del año 2005 a las 23 horas 59 minutos y 30 segundos). El sistema recorre el cronograma asumiendo que la fecha actual es la especificada en el parámetro de entrada y buscando cuál sería el evento de cronograma a programarse como próximo evento para comandar el Grabador. El método devuelve verdadero si encuentra tal evento o falso si no lo encuentra.

eventoCorriente()

Este método toma como parámetro una cadena que representa una fecha-hora y una entrada de cronograma usando el tipo de datos EventoCronograma. Este método devuelve verdadero y la entrada de cronograma modificada o devuelve falso. La cadena de fecha-hora debe cumplir con el formato ISO YYYYMMDDTHHMMSS (por ejemplo 20050131T235930 representa el 31 de enero del año 2005 a las 23 horas 59 minutos y 30 segundos). El sistema recorre el cronograma asumiendo que la fecha actual es la especificada en el parámetro de entrada y buscando cuál sería el evento que define el estado actual que debe tener el Grabador. El método devuelve verdadero si encuentra tal evento o falso si no lo encuentra.

insertar()

Este método corresponde al flujo Agregar Entrada y toma como parámetro un indicador de posición y una entrada de cronograma usando el tipo de datos ItemCronograma. El sistema agrega la entrada en la posición señalada o al final si la posición excede el largo del cronograma. Se escribe el nuevo cronograma en el archivo de cronograma del canal. El sistema comienza el flujo Cambio en el Cronograma.

borrar()

Este método corresponde al flujo Eliminar Entrada y toma como parámetro un indicador de posición. El sistema elimina la entrada de la posición señalada o elimina la última entrada si la posición excede el largo del cronograma. Se escribe el nuevo cronograma en el archivo de cronograma del canal. El sistema comienza el flujo Cambio en el Cronograma.

ok()

Este método no toma parámetros y devuelve verdadero si el servidor de Cronograma está funcionando correctamente.

4.3. Tipos de datos**AccionCronograma**

Es una enumeración con identificadores para los dos estados del grabador.

FechasCronograma

Es una estructura con los atributos que definen una fecha. Los miembros son cadenas ya que pueden coincidir con el valor numérico deseado o con la cadena "*" que representa cualquiera de todos los valores posibles. El miembro dia_semana representa el día de la semana de lunes (1) a domingo (7).

ItemCronograma

Es una estructura compuesta por los tipos de datos AccionCronograma, FormatoAudio y FechasCronograma y define una entrada de cronograma.

EventoCronograma

Es una estructura compuesta por los tipos de datos AccionCronograma, FormatoAudio y una cadena de fecha-hora. La cadena de fecha-hora debe cumplir con el formato ISO YYYYMMDDT-HHMMSS (por ejemplo 20050131T235930 representa el 31 de enero del año 2005 a las 23 horas 59 minutos y 30 segundos).

ListaCronograma

Es una secuencia simple de entradas de cronograma definidas según el tipo ItemCronograma.

4.4. Archivo de descripción de la interfaz (IDL)

```
#include "tipos_datos_ifz.idl"
module ifz {
    enum AccionCronograma {
        GRABAR,
        DETENER
    };
    struct FechasCronograma {
        string segundo; // 0-59
        string minuto; // 0-59
        string hora; // 0-23
        string dia_semana; // 0-7
        string dia_mes; // 1-31
        string mes; // 1-12
        string anho;
    };
    struct ItemCronograma {
        AccionCronograma accion;
        FormatoAudio formato_audio;
        FechasCronograma fechas;
    };
    struct EventoCronograma {
        AccionCronograma accion;
        FormatoAudio formato_audio;
    };
};
```

```
        string fecha;
    };
    typedef sequence<ItemCronograma> ListaCronograma;
    interface Cronograma {
        ListaCronograma obtListaCronograma();
        boolean eventoSiguiente(out EventoCronograma e, in string fecha);
        boolean eventoCorriente(out EventoCronograma e, in string fecha);
        void insertar(in unsigned long indice, in ItemCronograma i);
        void borrar(in unsigned long indice);
        boolean ok();
    };
};
```

5. Mezclador

5.1. Propósito

Para cada canal habilitado el sistema registra en el Servicio de Nombres de CORBA un servidor de Mezclador con identificador coincidente con el nombre del canal. Esta interfaz permite modificar y consultar el nivel con el que se adquiere el audio del canal y consultar el funcionamiento del mezclador.

5.2. Métodos

seleccionarEntrada()

Este método toma como parámetro un identificador de control de mezclador y lo configura como control asociado a la adquisición del audio.

obtNivel()

Este método no toma parámetros y devuelve los niveles con los que se está adquiriendo el audio (ganancia configurada para el control de mezclador asignado al canal) que se está empleando actualmente para la adquisición. Este método devuelve los niveles usando el tipo de datos Nivel.

estNivel()

Este método toma como parámetros un identificador de control de mezclador y un Nivel y establece los niveles para la adquisición del canal de audio.

terminar()

Este método no toma parámetros y termina el proceso del Mezclador.

ok()

Este método no toma parámetros y devuelve el estado del Mezclador. Este método permite verificar que tanto la interfaz CORBA como el proceso Mezclador están respondiendo correctamente.

5.3. Tipos de datos

Ver Tipos de datos compartidos.

5.4. Archivo de descripción de la interfaz (IDL)

```
#include "tipos_datos_ifz.idl"
module ifz {
    interface Mezclador {
        Nivel obtNivel(in long id_control);
        void estNivel(in long id_control, in Nivel nivel);
        void terminar();
        boolean ok();
    };
};
```

6. Tipos de datos compartidos

6.1. Propósito

Este archivo de definiciones solamente incluye tipos de datos que son utilizados por otras interfaces CORBA.

6.2. Tipos de datos

ListaErrores

Es una secuencia simple de cadenas que contienen mensajes de error.

FormatoAudio

Es una estructura con los atributos: nombre del formato de audio, cantidad de canales (por ejemplo 1 para mono, 2 para estéreo), frecuencia de muestreo (en muestras por segundo), cantidad de bits por muestra y la extensión del archivo que escribe en disco el Grabador.

Nivel

Es una estructura con los atributos de nivel de adquisición de los canales izquierdo y derecho. Son valores entre 0 y 1.

MedidasNivel

Es una estructura con los atributos: valor de pico y saturación de los canales izquierdo y derecho. Los atributos de pico se miden como el máximo nivel registrado dentro del último búfer adquirido, son valores entre 0 y 1. Los atributos de saturación indican si alguna muestra llegó al máximo valor adquirible.

6.3. Archivo de descripción de la interfaz (IDL)

```
module ifz {
    typedef sequence<string> ListaErrores;
    struct FormatoAudio {
        string nombre;
        unsigned short num_canales;
        unsigned short frec_muestreo;
        unsigned short bits_por_muestra;
        unsigned long rapidez_bits;
        string extension_arch;
    };
    struct Nivel {
        float izq, der;
    };
    struct MedidasNivel {
        float pico_izq, pico_der;
        boolean satur_izq, satur_der;
    };
};
```

Parte III

APÉNDICES

17. Herramientas del entorno de desarrollo

A lo largo del desarrollo del GCA se utilizaron numerosas herramientas que ayudaron en las diferentes etapas (diseño, codificación, documentación). A continuación se hace un breve resumen de las herramientas utilizadas y sus beneficios.

1. Diseño

1.1. Rational Rose

Rational Rose de IBM[20] es una plataforma de asistencia para el desarrollo de software. Como herramienta de diseño permite trabajar con modelos de objetos y generar varios tipos de diagramas UML. En nuestro caso fue utilizado para generar los diagramas de casos de uso y realizar los diagramas de clases. Tiene un módulo de ingeniería inversa que permite verificar la relación de los módulos y dependencia de las bibliotecas.

2. Codificación

2.1. KDevelop

Para la codificación y compilación en Linux se escogió KDevelop[21] por ser un programa simple y cómodo.

2.2. Microsoft Visual Studio

Para la codificación y compilación en Microsoft Windows se escogió Microsoft Visual Studio[22] por ser una plataforma de manejo intuitivo y cómodo con un módulo de depuración versátil y potente. Además el compilador es el más difundido para este sistema operativo y el recomendado por el fabricante.

2.3. Subversion y TortoiseSVN

Subversion[23] ayuda a que los desarrolladores lleven un seguimiento de los cambios en los ficheros de código fuente u otros documentos de un proyecto. Permite recuperar versiones anteriores, mantener ramas de desarrollo y todo lo necesario para consolidar un código que pueda ser modificado por varios usuarios. El programa tradicional para esta tarea es el CVS pero Subversion funciona mejor (tiene mejor manejo de archivos y estructuras de directorio) y es más fácil de utilizar.

Como cliente de Subversion, en Microsoft Windows, se utilizó el TortoiseSVN[24] que es una aplicación gráfica que se integra con el Explorador de Archivos. En Linux se utilizaron los programas de línea de comando provistos por Subversion.

2.4. Meld y WinMerge

Para revisar las diferencias entre dos archivos o dos estructuras de directorios, se utilizaron el Meld[25] en Linux y el WinMerge[26] en Microsoft Windows, ambos permiten una visualización clara de las diferencias (modificaciones, eliminaciones y agregados) utilizando colores y marcas. Estas aplicaciones permiten la resolución de conflictos de forma sencilla y constituyen una importante herramienta de apoyo al Subversion.

3. Documentación

3.1. Lyx

Lyx[27] es un programa gráfico multi-plataforma que permite la edición de texto usando L^AT_EX, por lo que permite el uso de las facilidades corrientes del mismo (notación científica, edición de ecuaciones, creación de apéndices, etc.). Se trata de un procesador de texto de tipo WYSIWYM (What you see is what you MEAN) en el que el usuario no necesita pensar en el formato final del trabajo, sino en el contenido, por lo que puede ser utilizado con facilidad para editar documentos grandes (libros) o con formato riguroso (tesis, artículos científicos). Lyx realiza un manejo avanzado y confiable de sub-documentos, referencias, numeración y paginación y permite exportar el texto en varios formatos (PDF, HTML, DVI).

3.2. Doxygen

Doxygen[28] permite generar la documentación del código en formato HTML a partir de comentarios que se insertan con una sintaxis dada en los archivos de encabezado. Doxygen interpreta estos comentarios y genera automáticamente un conjunto de páginas sin requerir que el usuario deba producir documentos adicionales para esta tarea. La ventaja es que puede mantenerse la documentación al día, sin riesgo de omisiones por falta de sincronización entre los cambios realizados del código y la documentación, sólo es necesario verificar que los comentarios de los encabezados sean los correctos.

18. Relevamiento y elección de codificadores de audio

1. Motivación

A los efectos de aprovechar eficientemente el espacio de almacenamiento en memoria no volátil y el ancho de banda de las redes involucradas, es necesario codificar el audio para reducir los datos. Actualmente existe una gran cantidad de codificadores/decodificadores (códecs) y la elección de uno adecuado requiere conocer sus atributos, los de la fuente de audio y del objetivo a lograr.

Por ejemplo, si la fuente de audio es mayormente voz, se puede utilizar un códec especializado que tome en cuenta sus características logrando mayor eficiencia. Si el objetivo de almacenar el audio es disponer de un testigo de una transmisión radial con fines legales, bastará con que la voz sea inteligible y reconocible. Si en cambio el audio almacenado va a ser editado posteriormente, habrá que tomar en cuenta la disponibilidad de herramientas de edición para ese formato.

2. Atributos de los códecs de audio

La calidad de un códec de audio es función de la velocidad de transferencia de bits, el retardo y la complejidad. Existe una fuerte interacción entre estos atributos y hay que lograr un compromiso entre ellos. Por ejemplo, los códecs con baja tasa de bits tienden a tener mayor retardo y pueden tener mayor complejidad, mientras que logran una calidad inferior a los de mayor tasa de bits.

Velocidad de transferencia de bits (bits/s)

Normalmente se trata de que sea la menor posible. Puede ser fija o variable según la señal de entrada. Existen distintos mecanismos para variar la tasa según la entrada. Por ejemplo, algunos códecs utilizan el mecanismo de comprimir silencios, enviando tasas menores o nulas y haciendo que el decodificador genere ruido suave para mejorar la comodidad.

Retardo

El retardo incide más o menos dependiendo de la aplicación. Para aplicaciones de almacenamiento, reproducción o difusión tiene una baja incidencia, mientras que es alta para una comunicación interactiva entre personas. En este caso el retardo consiste usualmente de tres componentes principales. La mayoría de los codificadores procesan una trama a la vez y en ocasiones requieren analizar datos más allá de los límites de la trama. Por lo tanto necesitan almacenar una trama o más datos en un búfer, lo que resulta en el *retardo algorítmico*. Este es el único retardo en el cual no incide la implementación.

La segunda contribución la da el *retardo de procesamiento* compuesto por el tiempo que requiere el codificador para analizar la voz y el decodificador para reconstruirla. Depende de la rapidez del

hardware usado en la implementación del códec. La suma de los retardos algorítmico y de procesamiento es llamada *retardo de un sentido del códec*.

La tercera contribución es el *retardo de comunicación* necesario para transmitir una trama completa entre el codificador y el decodificador. El total de la suma de las tres componentes es el *retardo de un sentido del sistema*. En caso de no haber eco, se toleran valores de 400 ms para este total, aunque preferiblemente debe estar por debajo de los 200 ms para facilitar la comunicación. En condiciones de eco el máximo tolerable baja a sólo 25 ms.

Complejidad

Los códecs se implementan utilizando procesadores digitales de señales (DSP) que se caracterizan por su velocidad de procesamiento dado en millones de instrucciones por segundo (MIPS) y el tamaño de las memorias RAM y ROM necesarias. Mayores valores de estos parámetros resultan en mayores costos del hardware y mayor consumo de energía.

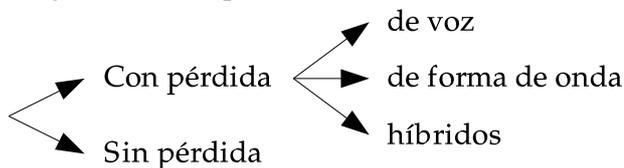
Robustez ante errores del canal

Calidad

De los atributos considerados, la calidad es el que tiene más dimensiones y es más difícil de caracterizar.

3. Clasificación de códecs de audio

La siguiente es una posible clasificación de los códecs de audio:



Sin pérdida

Logran una reducción de aproximadamente un 55 % del tamaño original sin perder información de la fuente.

Con pérdida

Toman en cuenta las características de los sistemas auditivo y/o fonético humanos para determinar la información que puede desecharse, afectando lo mínimo posible la calidad del audio percibida. En consecuencia el método es llamado codificación perceptual.

Códecs de voz

En general se basan en un modelo de generación de voz. Logran tasas de bits del orden de 0,5 a 2 bits/muestra equivalentes a 4 y 16 Kbit/s respectivamente. Las propiedades del oído humano también son tomadas en cuenta.

Códecs de forma de onda

A diferencia de los códecs de voz, no pueden beneficiarse de un modelo general de la fuente. Incorporan modelos perceptuales derivados de la propiedad de enmascaramiento del oído. Logran tasas de bits del orden de 2 a 4 bits/muestra.

Códecs híbridos

Constituyen una clase intermedia entre los códecs de forma de onda y los de voz. Actualmente presentes en las soluciones de vanguardia para códecs de voz con tasas de bits medias (0,5 a 2 bit/muestra) y alta calidad.

Nombre	Tecnología	Retardo (ms)	f_s (KHz)	B (KHz)	bits/(s canal)	SNR (dB)	Canales
LPC-10e	LPC				2,4		1
ITU-T G.711	PCM Ley A y μ				64		1
G.722		<30	16	7,5	56 o 64		1
ITU-T G.723.1	MP-MLQ/ACELP	<40			6,3 y 5,3		1
ITU-T G.726 (incl. G.721)	ADPCM				32	30 a 35	1
G.728	LD-CELP				16		
ITU-T G.729	CS-ACELP				8		1
ETSI GSM 06.10	RPE-LTP				13		1
ETSI GSM 06.20	VSELP				5,6		1
ETSI GSM 06.60	ACELP				12,2		1
Speex							
ISO/MPEG-1 Layer I	Subbandas, MDCT	<50	16, 24, 32, 48		32-192		1, 2
ISO/MPEG-1 Layer II		100	16, 24, 32, 48		32-192		1, 2
ISO/MPEG-1 Layer III		150	16, 24, 32, 48		8-192		1, 2
ISO/MPEG-4 AAC			8 a 96				1 a 48
AC-1	Mod. delta adapt.			20	512		
AC-2			32; 44,1; 48				
AC-3					384		5.1
Ogg Vorbis			8 a 192				1-255, 5.1

Tab. 1: Comparación entre códecs de audio

4. Codificación perceptual y otras técnicas de reducción de datos

La codificación perceptual se basa en el modelado psicoacústico del sistema auditivo humano. Usando el modelo, el codificador calcula qué información puede desechar manteniendo aceptable o nula la percepción de la degradación de calidad.

El oído humano y la parte del cerebro que percibe los sonidos forman un sistema enormemente complejo que selecciona parte de la información contenida en el sonido. Por ejemplo, es sorprendentemente aguda la habilidad de percibir un defecto en una señal, en cambio, no detecta un tono enmascarado por otro más fuerte cercano. A continuación se describen las técnicas utilizadas para la reducción de datos.

Umbral de audición y efecto de enmascaramiento

El oído es capaz de detectar sonidos cuyo nivel esté por encima de un umbral que es dependiente de la frecuencia. A su vez, una señal de cierto nivel mayor, enmascara haciendo inaudibles otras señales cercanas en frecuencia y de nivel menor, como si aumentara el umbral de audición. Ambos fenómenos son fundamentales para el diseño de los codificadores perceptuales.

El efecto de enmascaramiento ocurre también entre señales no simultáneas temporalmente cercanas, lo que sugiere que el cerebro integra las señales durante un período de tiempo. El enmascaramiento temporal ocurre tanto hacia adelante como hacia atrás en el tiempo.

Teniendo en cuenta estas características es posible por ejemplo, adaptar dinámicamente la cantidad de bits de cuantización de una banda, manteniendo el ruido de cuantización por debajo del umbral o dejar de codificar una señal enmascarada.

Codificación de Huffman

Esta codificación es utilizada para reducir la cantidad de bits promedio por símbolo, codificando con menos bits los símbolos más probables y con más los menos probables.

Codificación corrida-largo (run-length)

Consta en codificar el número de veces que se repite un símbolo. Por ejemplo: 555555 se puede codificar como 65.

Redundancia entre canales

Es frecuente que se utilicen señales estéreo o multi-canales que tienen redundancia de información entre ellos. Esta redundancia es aprovechada por los codificadores para mejorar la compresión.

5. Códec elegido

Se consideran los siguientes requerimientos para la elección del códec:

- El códec debe ser un estándar utilizado comúnmente en la industria y cuya documentación se pueda obtener.

- Debe disponerse de bibliotecas que permitan realizar la codificación y decodificación.
- Debe permitir distintos niveles de compresión para tener desde un mínimo equivalente a la calidad telefónica hasta un máximo equivalente a la calidad de un disco compacto.
- El acceso al audio debe ser aleatorio, es decir que se pueda obtener eficientemente un segmento intermedio de un archivo de audio.
- La velocidad de procesamiento requerida por el algoritmo de compresión debe ser tal que permita realizarse por software, en una computadora personal sin características especiales.

Varios de los códecs estudiados satisfacen los requerimientos. Sin embargo la mayor disponibilidad de información y bibliotecas de codificación y decodificación y la gran difusión que tiene en el mercado el códec ISO/MPEG Layer III favorecen su elección. De todos modos se buscará independizar lo más posible el código a desarrollar del códec elegido. Los puntos dependientes del códec se separarán en módulos que eventualmente puedan ser sustituidos por otros desarrollados para un códec diferente.

19. Manual del GCA

1. Capacidades del producto

El GCA es un sistema de adquisición y almacenamiento de audio que puede operar simultáneamente con varias tarjetas de audio. Cada tarjeta de audio se asocia a un canal con un cronograma propio que establece los momentos en que se ha de adquirir y grabar el audio a disco y la calidad de la adquisición.

El GCA incluye un sistema que supervisa el correcto funcionamiento del conjunto y ejecuta archivos externos al detectar condiciones de falla. Además se incluye una consola gráfica de administración y monitorización del sistema que funciona a través de la red. En combinación con un servidor de páginas web y opcionalmente un servidor de streaming, se puede buscar el audio almacenado por canal y por fecha utilizando una sencilla interfaz web. Como resultado de la búsqueda, la interfaz web permite descargar los archivos a la computadora del cliente o escucharlos en tiempo real.

Las características principales del GCA son:

- Todos los módulos del sistema pueden operar tanto en Linux como en Microsoft Windows.
- El sistema principal y el sistema de control de fallas operan como servicios y no requieren de interfaz gráfica.
- Cada canal tiene su propio cronograma que indica cuándo grabar y con qué calidad adquirir el audio.
- El cronograma es versátil, permite una programación por fechas fijas, por días de la semana o por eventos de repetición.
- El audio se graba sin pérdida de muestras de uno a otro archivo gracias a un cuidado diseño basado en colas.
- Luego que cada archivo se guarda en disco, se puede lanzar un proceso externo, configurable según las necesidades y por canal, que permite actuar sobre el archivo guardado.
- El proceso externo es asignado por canal por lo que cada canal puede tener un proceso diferente.
- El GCA dispone de un módulo independiente de control de fallas, que puede funcionar en otra PC y que se comunica con el proceso central del GCA para verificar que no existan fallas de funcionamiento.
- Al detectar una falla se pueden disparar procesos en base única y/o recurrente. Al igual que los procesos de post-grabación, son ajustables a las necesidades del cliente.
- Además del funcionamiento basado en el cronograma, el operador de la consola puede forzar de forma manual una detención o inicio de grabación con un formato escogido.

2. Requisitos mínimos del sistema

2.1. Sistema principal GCA

El sistema principal del GCA necesita para su operación básica una computadora con los siguientes requerimientos mínimos:

- Procesador Intel Pentium II o equivalente
- Memoria RAM de 256 Mbytes
- 40 Mbytes libres en el disco rígido
- Tarjeta de sonido

Se debe disponer de espacio en disco adicional de acuerdo a la cantidad de audio que se desee almacenar. Se requieren aproximadamente 12 MBytes de memoria RAM por cada tarjeta de audio adicional. De acuerdo a los procesos de post-grabación que se configuren, pueden requerirse mayores capacidades de procesamiento o memoria.

2.2. Control de Fallas del GCA

El sistema de Control de Fallas del GCA puede funcionar en una computadora separada de la que utiliza el sistema principal GCA, mientras pueda mantener con ésta una conexión de red. El sistema de Control de Fallas necesita para su operación una computadora con los siguientes requerimientos mínimos:

- Procesador Intel Pentium II o equivalente
- Memoria RAM de 64 Mbytes
- 40 Mbytes libres en el disco rígido

Pueden existir requerimientos mayores de acuerdo a las acciones que se programen para realizarse ante la detección de fallas.

2.3. Consola del GCA

La Consola de la operación necesita para su operación una computadora con los siguientes requerimientos mínimos:

- Procesador Intel Pentium II o equivalente
- Memoria RAM de 64 Mbytes
- 40 Mbytes libres en el disco rígido para instalar la aplicación cliente

Se requiere espacio en disco suficiente para recuperar el audio que solicite el usuario.

3. Instrucciones de instalación

La instalación básica del sistema requiere las bibliotecas de los paquetes Mico, Boost y wxWidgets que se incluyen en el disco de instalación. Como resultado de la instalación se registrarán tres servicios en el sistema operativo:

- Servicio de nombres CORBA (nsd)
- Sistema principal GCA (gca)
- Sistema de Control de Fallas del GCA (gcactrlfallas)

El servicio de nombres CORBA es un servicio que mantiene el registro de todos los procesos del GCA que requieren comunicarse entre sí. Este servicio actúa como una agenda de referencia y la comunicación se realiza utilizando un puerto de comunicación de protocolo TCP cuyo número es configurable por el usuario.

El sistema principal GCA es el servicio que administra y realiza la grabación del audio en los múltiples canales configurados.

El sistema de Control de Fallas del GCA es el servicio que, mediante consultas periódicas al sistema principal, controla que la operación se esté realizando sin errores. En caso de detectar errores o perder comunicación con el GCA, ejecuta programas externos configurables por el usuario.

Además de los servicios, se instalará la aplicación Consola del GCA que permite la administración, configuración y monitorización de los canales.

3.1. Instalación en Linux

Biblioteca Mico

Instalar la biblioteca Mico usando el paquete “rpm” que se encuentra en el disco de instalación:

```
rpm -ivh mico-2.3.11-1.i386.rpm
```

Registrar el servicio de nombres (nsd) de CORBA en el sistema operativo. Se deben copiar los archivos “script” al directorio donde se encuentran los demás archivos “script” para otros servicios (normalmente “/etc” es el directorio base):

```
cp nsd.init /etc/rc.d/init.d/nsd
cp nsd.sysconfig /etc/sysconfig/nsd
chkconfig --add nsd
```

Biblioteca Boost

Instalar la biblioteca Boost usando el paquete “rpm” que se encuentra en el disco de instalación:

```
rpm -ivh boost-1.32.0-1.fc3.i386.rpm
```

Biblioteca wxWidgets

Descomprimir las bibliotecas wxWidgets que se encuentran en el disco de instalación dentro de un directorio vacío y copiar los archivos descomprimidos al directorio donde se encuentran las demás bibliotecas instaladas en el sistema operativo (normalmente “/usr/lib”):

```
tar -xvzf wxwidgets_2.6.0.tar.gz
cp * /usr/lib
```

Sistema GCA

Instalar el sistema GCA sin revisión de dependencias:

```
rpm -ivh --nodeps gca-1.0-1.i386.rpm
```

3.2. Instalación en Microsoft Windows

Para Microsoft Windows se incluye en el disco de instalación un programa que ejecuta la instalación completa del sistema GCA y las bibliotecas necesarias. El avance de la instalación se realiza por ventanas que se describen a continuación. Para avanzar entre las ventanas se debe presionar el botón “Next” y para retroceder el botón “Back”.

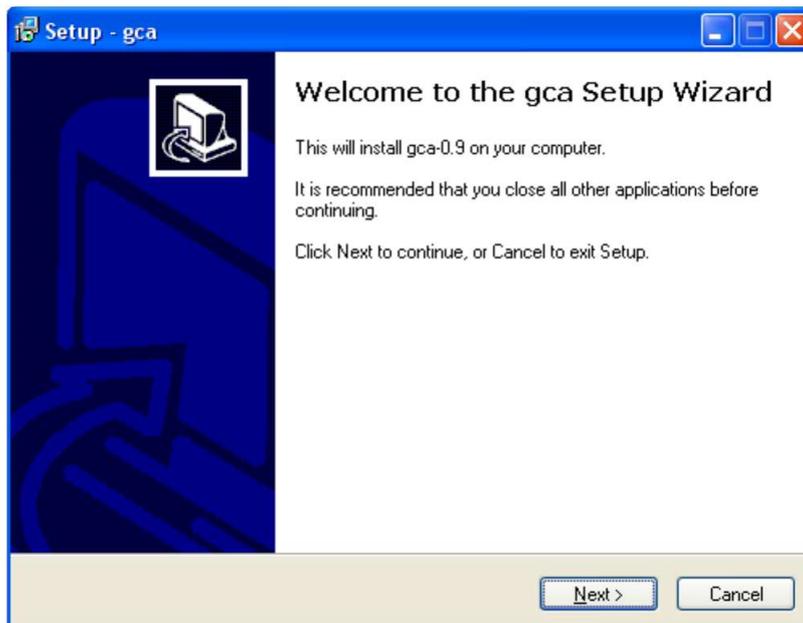


Fig. 1: Pantalla inicial de instalación

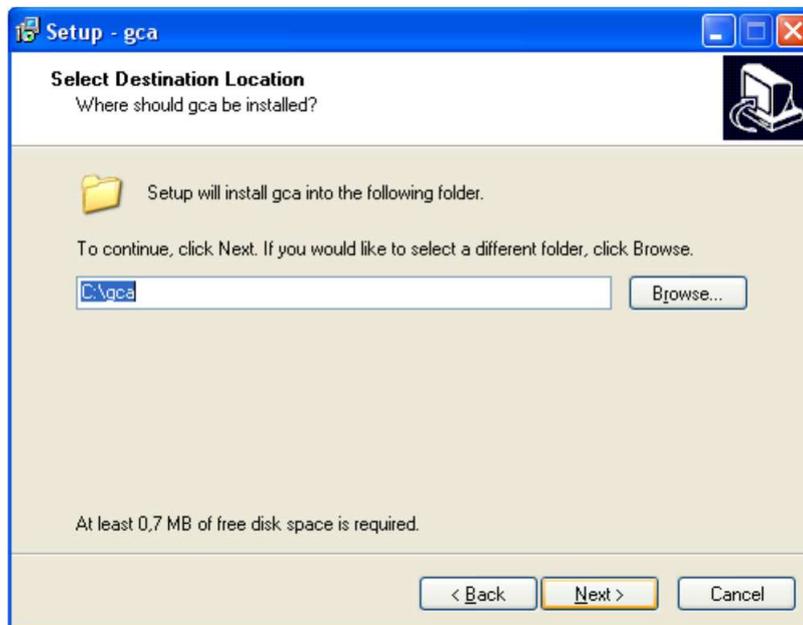


Fig. 2: Directorio de instalación del sistema GCA

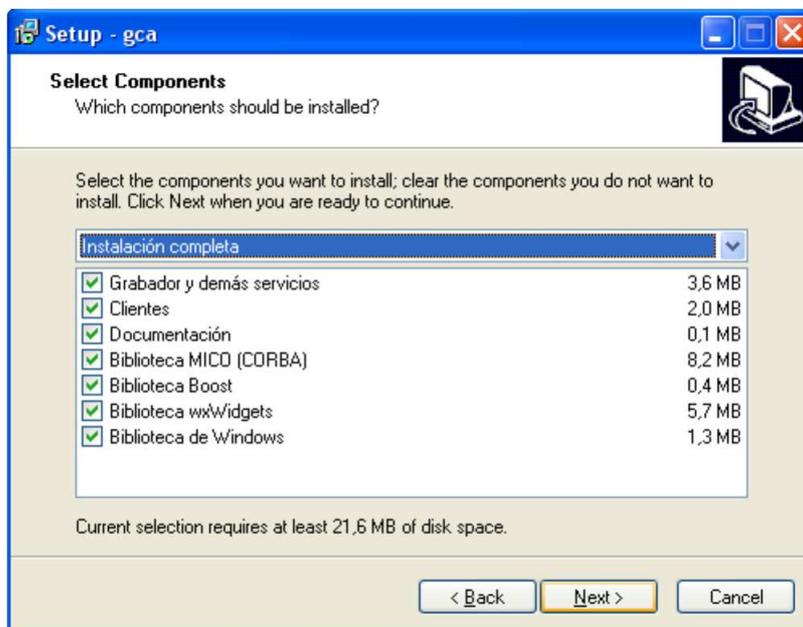


Fig. 3: Selección de paquetes a instalar

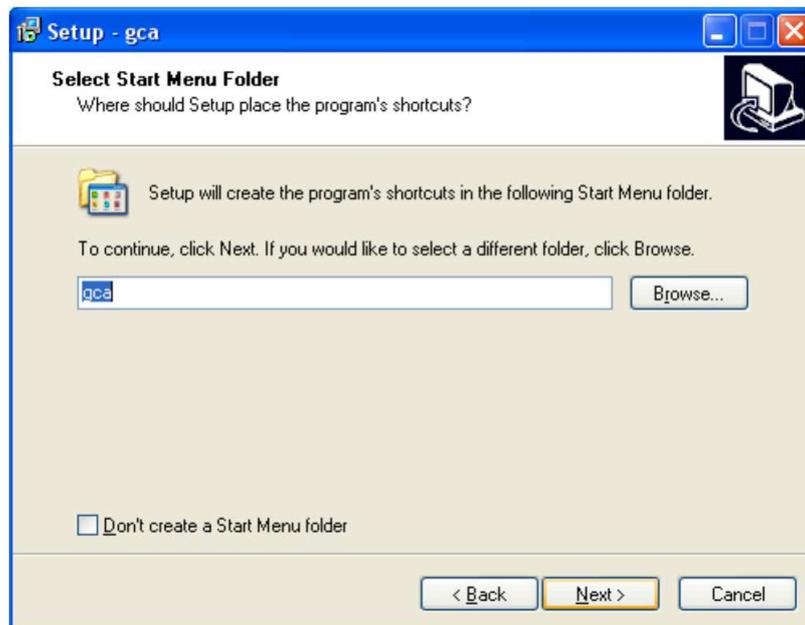


Fig. 4: Nombre de la carpeta del menú Inicio

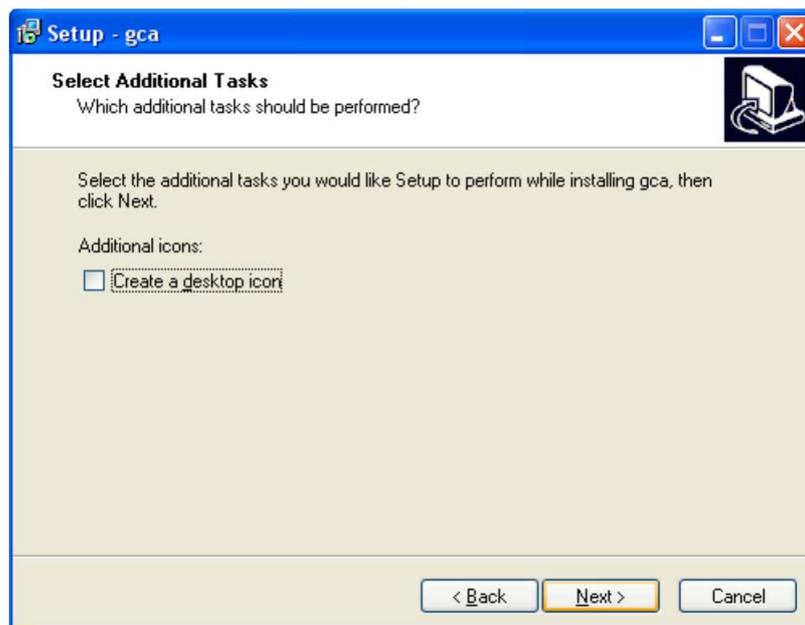


Fig. 5: Confirmación para crear acceso directo en el Escritorio

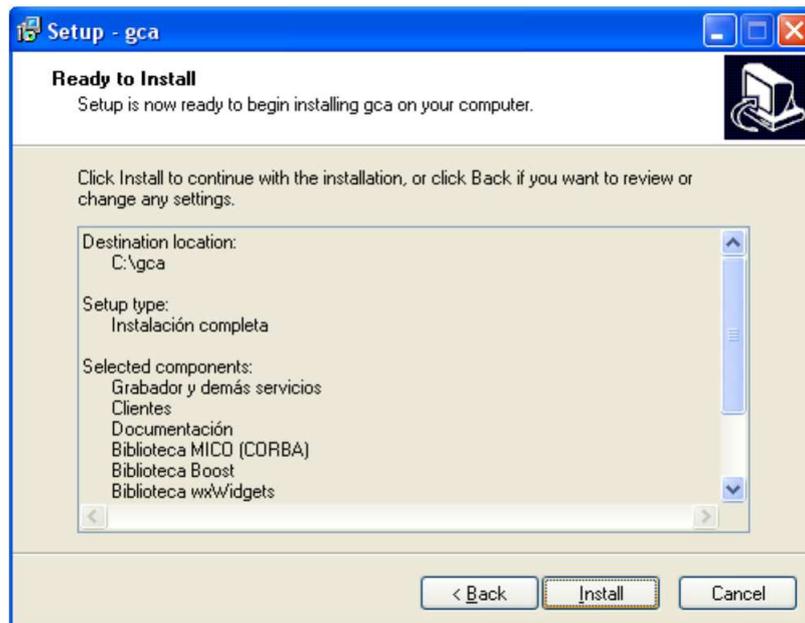


Fig. 6: Confirmación de opciones de instalación

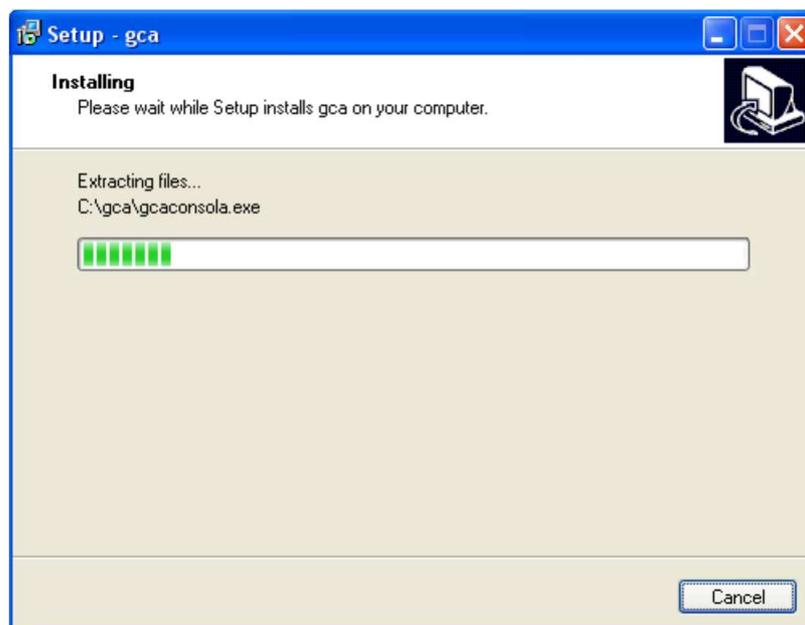


Fig. 7: Avance de la instalación

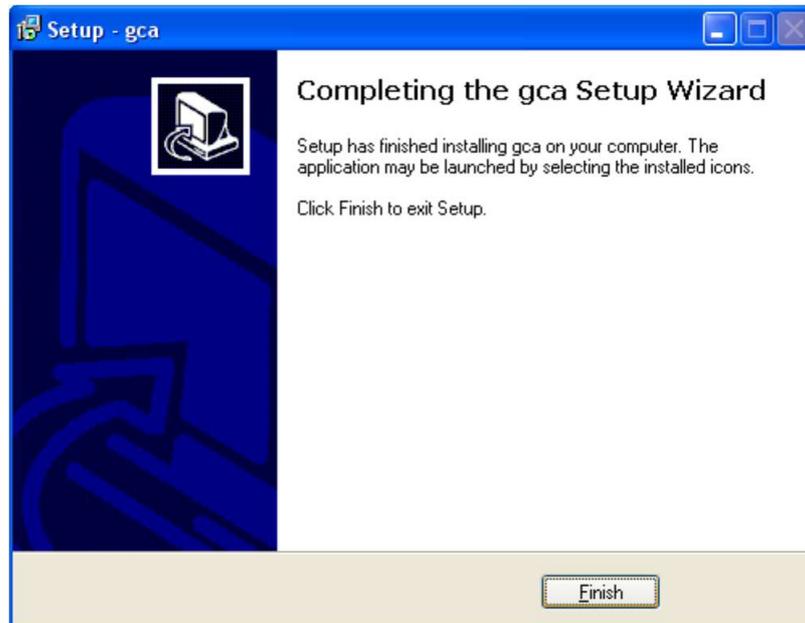


Fig. 8: Finalización de la instalación

3.3. Programas adicionales

Para poder buscar el audio almacenado a través de la red, se debe configurar un servidor de páginas web con soporte para PHP. Para reproducir el audio en tiempo real utilizando el protocolo de streaming RTSP, se debe instalar un servidor de streaming.

Servidor de páginas web

Las páginas que efectúan la búsqueda del audio se encuentran en el disco de instalación y son las siguientes:

- `gcarecuperar.php`
- `gcarecuperar_busca.php`
- `gcarecuperar_conf.php`

Se deben copiar las páginas de búsqueda de audio en la ubicación que se considere conveniente. Se trata de archivos de extensión “.php” que deben ser procesados por el intérprete PHP del servidor de páginas web.

En el servidor de páginas web, se debe configurar una referencia hacia el directorio base donde se encuentran las carpetas que contienen los archivos de audio. El nombre que se asigne a este camino debe configurarse en el archivo de configuraciones cuya estructura se explica en la sección 4.5. Por ejemplo, si las carpetas correspondientes a los canales de audio se encuentran en “/var/lib/gca”, se puede crear una referencia en el servidor de páginas web llamada “audio_gca”. De esta forma, al utilizar un vínculo de tipo “http://servidor/audio_gca” se debe poder acceder al directorio real “/var/lib/gca”.

Servidor de streaming

En el servidor de streaming se debe configurar una referencia hacia el directorio base donde se encuentran las carpetas que contienen los archivos de audio. El nombre que se asigne a este camino debe configurarse en el archivo de configuraciones cuya estructura se explica en la sección 4.5. Por ejemplo, si las carpetas correspondientes a los canales de audio se encuentran en “/var/lib/gca”, se puede crear una referencia en el servidor de streaming llamada “audio_gca”. De esta forma, al utilizar un vínculo de tipo “rtsp://servidor/audio_gca” se debe poder acceder al directorio real “/var/lib/gca”.

Cliente RTSP

Para reproducir el audio usando el protocolo de streaming RTSP, se debe seleccionar un cliente con soporte para archivos de definición SMIL. La página de búsqueda devuelve un archivo SMIL con la lista de vínculos correspondiente a los archivos que cumplen el criterio establecido.

Otros servidores

Si la recuperación del audio se desea realizar utilizando por ejemplo el protocolo FTP, se debe configurar un servidor FTP con un camino hacia los archivos de audio, al igual que lo tiene el servidor de páginas web. También puede realizarse la recuperación utilizando el protocolo SMB para lo cual debe configurarse correctamente el servicio de compartir archivos. En la sección 4.5 se explica cómo modificar el archivo de configuraciones para que los hipervínculos de la página de resultados de búsqueda refieran al protocolo deseado.

4. Puesta en marcha

4.1. Configuración del servicio de nombres CORBA (nsd)

El servicio de nombres de CORBA se instala de forma predeterminada utilizando el puerto TCP 12456. Para modificar este puerto el procedimiento depende del sistema operativo utilizado.

Linux

Modificar el archivo copiado en “/etc/sysconfig” que fue instalado con el nombre “nsd” según lo explicado en la sección 3.1. Se debe sustituir el número de puerto en la línea que inicia con “PORT”:

```
# Escuchar en el puerto TCP siguiente (port)
PORT=12456
```

Microsoft Windows

Modificar la clave del registro de Windows llamada “parametros_nsd”. La clave se encuentra en: HKEY_LOCALMACHINE\SYSTEM\CurrentControlSet\Services\GCA - Servidor de nombres de Corba

Se debe modificar el puerto que figura al final de la clave cuya contenido es similar a:

```
parametros_nsd = -ORBIIOPAddr inet::12456
```

4.2. Configuración inicial de archivos del sistema GCA

Si se modificó el puerto que utilizará el servidor de nombres de CORBA, se deben realizar las respectivas correcciones en los archivos de configuración del sistema principal GCA y el sistema de Control de Fallas. Los archivos a modificar son respectivamente “gca_conf.xml” y “gcactrlfallas_conf.xml”. La clave a modificar es la siguiente:

```
<args_corba>
-ORBInitRef NameService=corbaloc::localhost:12456/NameService
</args_corba>
```

Además, en caso que el servidor de nombres de CORBA no esté ejecutándose en la misma computadora que el sistema GCA o el sistema de Control de Fallas, se debe corregir el valor “localhost” con la dirección de red correspondiente o un nombre válido.

En Microsoft Windows se debe además verificar que el directorio de ejecución coincida con el especificado durante la instalación. La clave correspondiente para esto es:

```
<dir_exe>c:\gca</dir_exe>
```

En la sección 6 se encuentra la explicación detallada del funcionamiento y configuraciones del sistema de Control de Fallas.

4.3. Inicio de servicios

Para que el sistema GCA y el Control de Fallas funcionen correctamente debe estar iniciado el servicio de servidor de nombres de CORBA (nsd). Puede configurarse el sistema operativo para que se inicien automáticamente los servicios necesarios.

4.4. Configuración inicial de canales

La instalación del GCA configura un canal de ejemplo llamado “canal1” que utiliza la tarjeta de audio predeterminada del sistema operativo. El funcionamiento de este canal no se inicia automáticamente al iniciar el GCA, por lo que es recomendable iniciar el sistema y utilizar la consola para verificar que los atributos sean correctos. En la sección 5 se encuentra el manual de la Consola que explica cómo modificar los atributos de un canal (sección 5.4), iniciar su funcionamiento (sección 5.5) y configurar el cronograma (sección 5.7).

4.5. Configuración para la recuperación del audio

El archivo “gcarecuperar_conf.php” contiene la configuración de los caminos virtuales que se usan como base para generar las referencias a los archivos de audio. Este archivo debe ser ajustado según la configuración realizada en los servidores de streaming y páginas web. Se muestra la estructura completa con rutas de ejemplo a continuación.

```
<? // Extension de archivo a buscar
$extension = ".mp3";
// URL base de streaming
$prefijo_escuchar = "rtsp://servidor/gca/";
// Camino donde buscar los archivos
$ruta_audio = '/home/gca/';
$prefijo_bajar = 'http://servidor/gca/';
?>
```

La siguiente tabla explica el significado de cada clave.

Clave	Descripción
extension	Extensión de los archivos a buscar.
ruta_audio	Directorio base real en el sistema de archivos, donde se encuentran las carpetas con el audio de cada canal. El algoritmo de búsqueda de archivos utiliza como directorio de búsqueda el resultante de la concatenación de la clave "ruta_audio" con el nombre del canal elegido en las condiciones de búsqueda. Por ejemplo, si se define como ruta_audio el valor "/home/gca/", el audio del "canal1" se buscará dentro del directorio "/home/gca/canal1".
prefijo_bajar	<p>Prefijo que se utiliza para generar los hipervínculos de la página de resultados de búsqueda. Este prefijo debe ser válido para el servidor y debe estar asociado al directorio base donde se encuentran las carpetas con el audio de cada canal.</p> <p>Se puede elegir el protocolo que se utilizará para la descarga. Se puede por ejemplo realizar la recuperación usando el servicio de archivos compartidos ("file://"), por HTTP ("http://"), por transferencia FTP ("ftp://"), etc.</p> <p>La ruta definida luego del protocolo debe ser válida en el contexto donde se ubiquen los clientes. Puede ser una dirección IP o un nombre que los clientes puedan resolver. Además de la denominación del servidor, si corresponde, se debe agregar, el camino virtual definido en el servidor (camino vinculado al directorio base).</p>
prefijo_escuchar	<p>Prefijo que se utiliza para generar los hipervínculos del archivo SMIL que contiene la definición de la secuencia de reproducción. Este prefijo debe ser válido para el servidor y debe estar asociado al directorio base donde se encuentran las carpetas con el audio de cada canal.</p> <p>Se puede elegir el protocolo que se utilizará. Existen clientes de reproducción que interpretan archivos SMIL y que pueden descargar el audio en modo streaming usando RTSP o con una copia común. En este caso puede cambiarse el prefijo "rtsp://" por el prefijo "file://".</p> <p>Al igual que para prefijo_bajar, la ruta definida luego del protocolo debe ser válida.</p>

5. Manual de la Consola

5.1. Funciones

La Consola del GCA es una aplicación gráfica que permite la administración de las funciones principales del GCA como la administración, monitorización y configuración de los canales de audio. Sus funciones principales son:

- Operación a través de la red de datos
- Conexión a diferentes sistemas GCA remotos
- Configuración de canales (altas, bajas y modificación de canales)
- Acceso a todos los atributos de un canal
- Inicio y detención de la operación de un canal
- Monitorización en tiempo real de los niveles del audio
- Ajuste en tiempo real del nivel de volumen de adquisición de un canal
- Comando manual de detención de grabación de un canal
- Comando manual de inicio de grabación de un canal con formato personalizable
- Consulta y modificación del cronograma de un canal

5.2. Inicio de la Consola

La Consola es un archivo binario llamado “gcaconsola” en el sistema operativo Linux o “gcaconsola.exe” en Microsoft Windows. Para su utilización en Linux se requiere tener funcionado la interfaz gráfica X. Para iniciar la Consola basta ejecutar el archivo binario sin parámetros adicionales.

Al iniciar la Consola se muestra la pantalla principal que es un panel con una barra de menú en la parte superior y una barra de estado en la parte inferior.

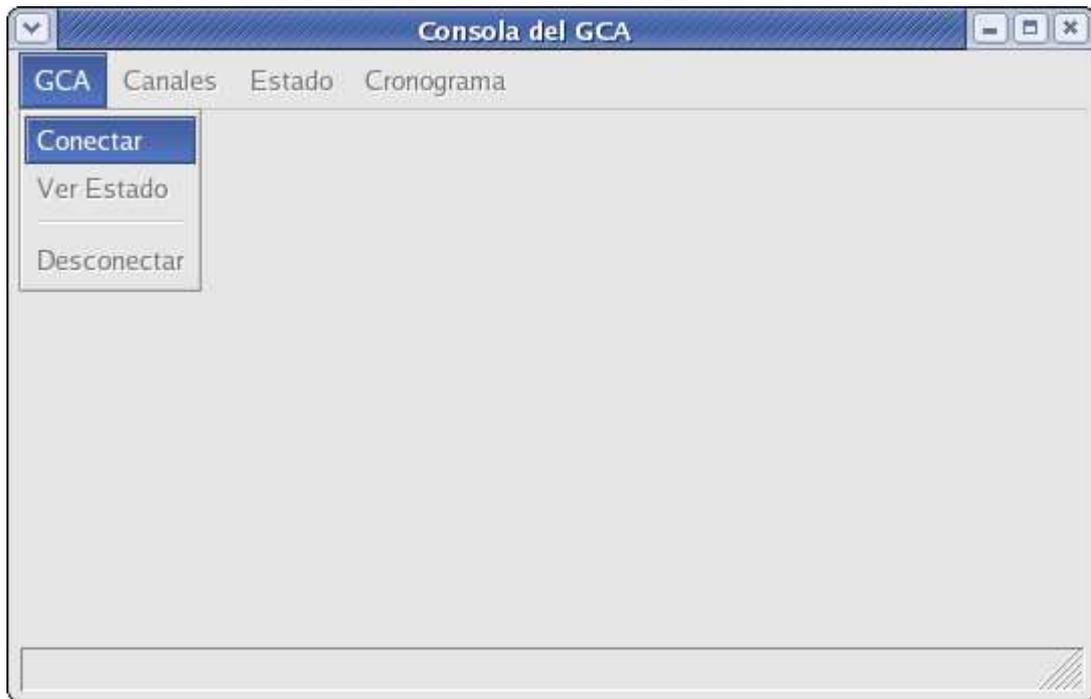


Fig. 9: Pantalla principal de la Consola

5.3. Selección de GCA

Conexión a un GCA

La Consola puede conectarse a través de la red a un sistema GCA remoto. Para iniciar la conexión se debe utilizar la opción “Conectar” del menú “GCA”, que abre un diálogo de conexión.

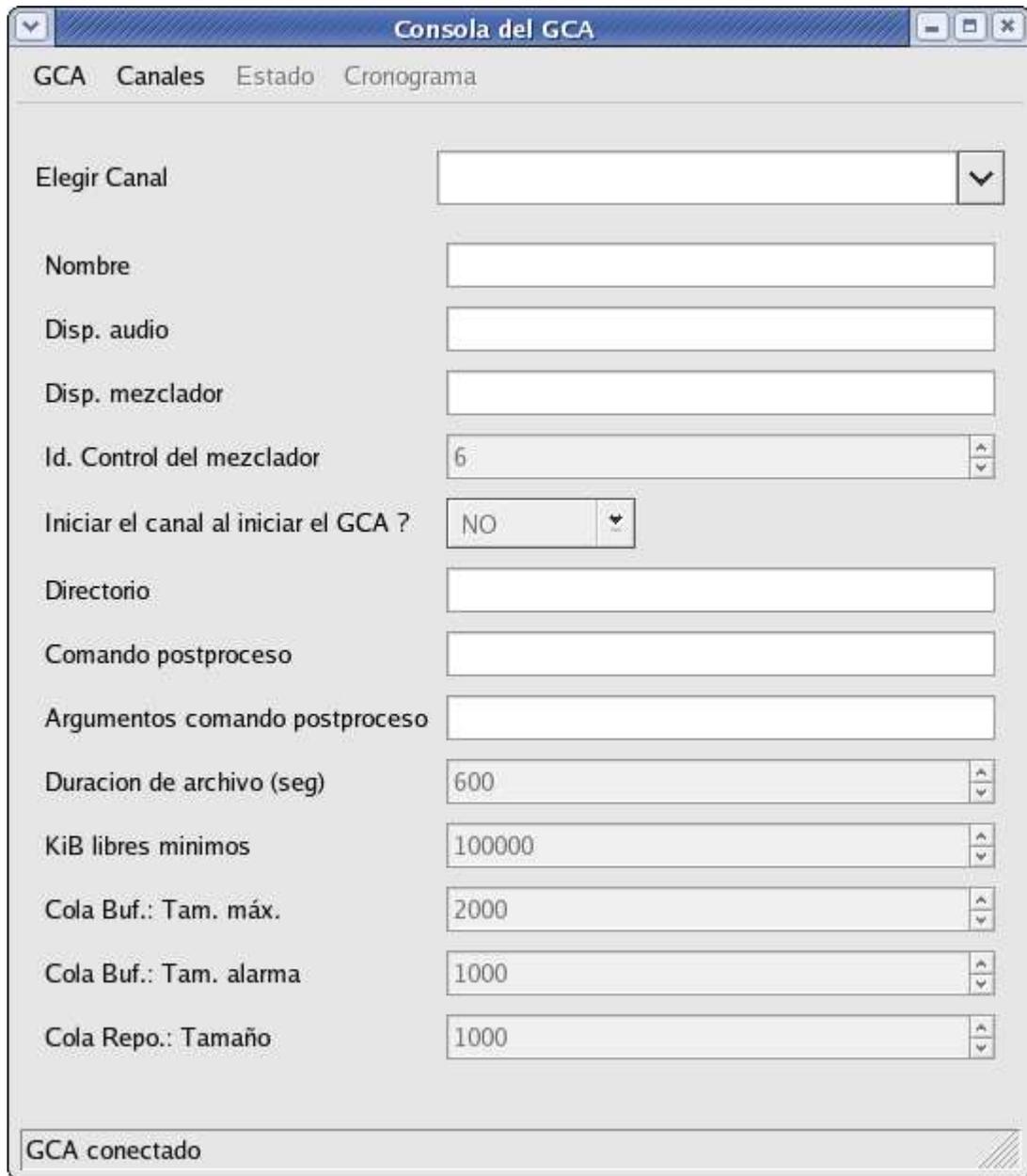


Fig. 10: Diálogo para la conexión al GCA

En este diálogo se deben completar los datos para acceder al servidor donde está ejecutándose el servicio NSD de CORBA al cual está registrado el GCA que se desea contactar.

Dirección del servidor NSD	Nombre válido de red o dirección IP del servidor que ejecuta el servicio NSD
Puerto del servidor NSD	Número de puerto TCP que utiliza el servicio NSD

Para efectuar la conexión se deben confirmar los datos y presionar el botón “Conectar”. La Consola se conecta con el servidor NSD especificado y luego consulta al GCA la lista de canales configurados. La pantalla principal habilita el menú “Canales”. Si no se puede establecer la conexión se mostrará un error en la barra de estado.



The screenshot shows a window titled "Consola del GCA" with a menu bar containing "GCA", "Canales", "Estado", and "Cronograma". The main area contains several configuration fields:

Elegir Canal	<input type="text"/>
Nombre	<input type="text"/>
Disp. audio	<input type="text"/>
Disp. mezclador	<input type="text"/>
Id. Control del mezclador	<input type="text" value="6"/>
Iniciar el canal al iniciar el GCA ?	<input type="text" value="NO"/>
Directorio	<input type="text"/>
Comando postproceso	<input type="text"/>
Argumentos comando postproceso	<input type="text"/>
Duracion de archivo (seg)	<input type="text" value="600"/>
KiB libres minimos	<input type="text" value="100000"/>
Cola Buf.: Tam. máx.	<input type="text" value="2000"/>
Cola Buf.: Tam. alarma	<input type="text" value="1000"/>
Cola Repo.: Tamaño	<input type="text" value="1000"/>

At the bottom of the window, a status bar displays "GCA conectado".

Fig. 11: Pantalla principal luego de conectar al GCA

Para desconectar la consola del GCA actual se debe utilizar la opción "Desconectar" del menú "GCA". Esta opción vuelve a habilitar la opción "Conectar".

Monitorización del GCA

Para monitorizar el estado del GCA se debe utilizar la opción "Ver Estado" del menú "GCA", que abre un diálogo donde se muestra el estado del GCA, y si hay fallas muestra la lista de las fallas. La Consola consulta el estado del GCA periódicamente y actualiza el estado de la pantalla.



Fig. 12: Diálogo de monitorización del GCA

5.4. Configuración de canales



Fig. 13: Menú "Canales"

Selección de canal

Al conectarse al GCA, la consola habilita el menú “Canales” y el cuadro combinado “Elegir Canal” que permite seleccionar el canal sobre el cual se desea operar. Las funciones que operan sobre un canal, como son la monitorización o edición de cronograma, se realizan para el canal que fue elegido usando este cuadro combinado.

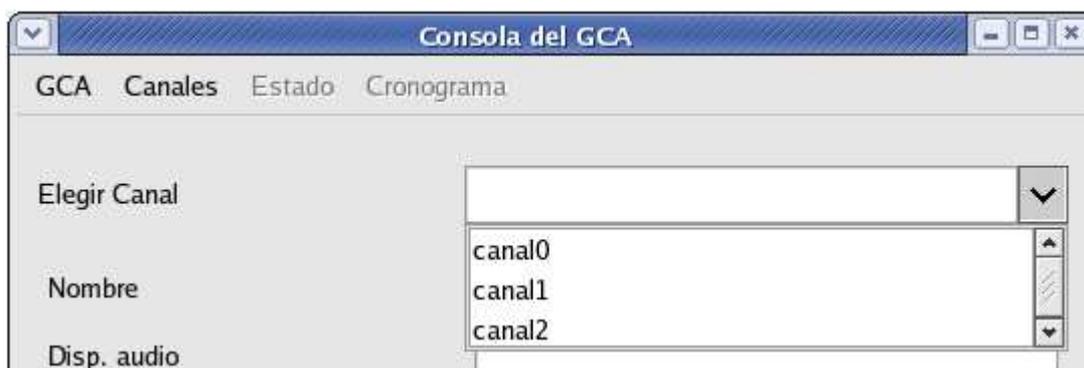


Fig. 14: Cuadro combinado “Elegir Canal”

Además, la opción “Elegir Canal” del menú “Canales”, restaura la pantalla a su valor inicial y vuelve a generar el cuadro combinado con la lista de canales configurados en el GCA.

Agregar un canal nuevo

Para agregar un canal nuevo a la lista de canales configurados en el GCA, se debe utilizar la opción “Agregar Nuevo Canal” del menú “Canales”. Al elegir esta opción, la pantalla dispone todos los cuadros con los atributos de un canal completados con los valores sugeridos por omisión. En la parte inferior de la pantalla aparecen dos botones para confirmar o cancelar la operación de inserción del nuevo canal. Se deben completar todos los cuadros (exceptuando los relativos al post-proceso que no son imprescindibles), para luego confirmar la inserción del nuevo canal.

The screenshot shows a window titled "Consola del GCA" with a menu bar containing "GCA", "Canales", "Estado", and "Cronograma". The main area is a form with the following fields:

Nombre	<input type="text"/>
Disp. audio	<input type="text" value="/dev/dsp"/>
Disp. mezclador	<input type="text" value="/dev/mixer"/>
Id. Control del mezclador	<input type="text" value="6"/>
Iniciar el canal al iniciar el GCA ?	<input type="text" value="NO"/>
Directorio	<input type="text"/>
Comando postproceso	<input type="text"/>
Argumentos comando postproceso	<input type="text"/>
Duracion de archivo (seg)	<input type="text" value="600"/>
KiB libres minimos	<input type="text" value="100000"/>
Cola Buf.: Tam. máx.	<input type="text" value="2000"/>
Cola Buf.: Tam. alarma	<input type="text" value="1000"/>
Cola Repo.: Tamaño	<input type="text" value="1000"/>

At the bottom of the form are two buttons: "Agregar" and "Cancelar".

Complete la información de los controles y pulse el botón Agregar

Fig. 15: Pantalla para agregar un canal nuevo

En la tabla se detallan los atributos de un canal, estos datos se citarán como referencia en las secciones siguientes.

Atributo	Descripción
Nombre	Cadena alfanumérica que describe el canal, se utiliza como identificador, por lo tanto debe ser una cadena única para cada canal.
Disp. audio	Cadena con la ruta del dispositivo de audio o el número de dispositivo de audio según como cada sistema operativo lo identifique. En Linux normalmente son “/dev/dsp0”, “/dev/dsp1”. En Microsoft Windows son un número entero entre 0 y la cantidad de dispositivos totales menos 1.
Disp. mezclador	Cadena con la ruta del dispositivo mezclador asociado al dispositivo de audio o el número de dispositivo mezclador según como cada sistema operativo lo identifique. En Linux normalmente son “/dev/mixer0”, “/dev/mixer1”. En Microsoft Windows son un número entero entre 0 y la cantidad de dispositivos totales menos 1.
Id. Control del mezclador	Número que identifica al control del mezclador que maneja la amplificación del canal por donde se conecta la fuente de audio. Por ejemplo, en Linux, normalmente el control 6 representa el volumen de la línea de entrada y el 7 el volumen del micrófono.
Iniciar el canal al iniciar el GCA ?	En este cuadro se elige si en el momento en que se inicia el servicio del GCA, el canal inicia automáticamente su operación (de acuerdo al cronograma vigente).
Directorio	Cadena con la ruta completa en el sistema de directorios donde se guardarán los archivos de audio de este canal. La sintaxis depende del sistema operativo, por ejemplo en Linux: “/var/lib/gca/canal1” o en Microsoft Windows: “c:\audio\canal1”.

Comando postproceso	Cadena con la ruta completa a un archivo ejecutable. Este archivo se ejecutará al finalizar la grabación en disco de cada archivo de audio. Si se requieren parámetros, se deben especificar en el próximo cuadro. Además, en tiempo de ejecución, el sistema establece como último parámetro el nombre del archivo de audio recién finalizado. Es posible utilizar este cuadro para establecer la ruta a un intérprete y luego enviarle como parámetro el archivo a ejecutar. De este modo se puede utilizar un intérprete de línea de comandos para procesar un archivo de procesamiento por lotes, o un intérprete de lenguaje Perl para ejecutar un “script” en Perl, etc.
Argumentos comando postproceso	Cadena con los parámetros necesarios para utilizar en conjunto con el comando postproceso establecido en el cuadro anterior.
Duración de archivo (seg)	Duración máxima de cada archivo de audio generado por el grabador del canal. Cuando el archivo de audio supera este tamaño se cierra y se continúa el registro en un archivo nuevo. Un archivo puede ser de duración menor a la establecida si hay una pausa en el cronograma u otro evento que genere una detención de la adquisición.
KiB libres mínimos	Espacio libre en disco medido en KiB que debe tener la unidad de disco donde se guardan los archivos de audio. Si no hay suficiente espacio, la grabación se detiene y se reporta una alarma de funcionamiento. Es conveniente configurar este valor con el margen suficiente, teniendo en cuenta el espacio adicional que requiera el post-proceso de un archivo de audio, el tamaño máximo del propio archivo según la duración especificada para el canal y la cantidad de canales que dependen de la misma unidad de disco.
Cola Buf.: Tam. máx.	Este control permite el ajuste fino del sistema de grabación. Al usar la opción “Agregar Nuevo Canal”, el cuadro se completa con el valor sugerido por los desarrolladores. Es el tamaño máximo en cantidad de búfers que pueden alcanzar las colas de adquisición y grabación. Si se alcanza este valor, el sistema descarta el búfer más antiguo al incorporar un búfer nuevo.

Cola Buf.: Tam. alarma	Este control permite el ajuste fino del sistema de grabación. Al usar la opción “Agregar Nuevo Canal”, el cuadro se completa con el valor sugerido por los desarrolladores. Cuando la cantidad de búfers de las colas de adquisición y grabación supera este valor el sistema reporta una falla en el canal. La grabación no obstante prosigue y el tamaño máximo que pueden alcanzar las colas está dado por el valor del cuadro anterior.
Cola Repo.: Tamaño	Este control permite el ajuste fino del sistema de grabación. Al usar la opción “Agregar Nuevo Canal”, el cuadro se completa con el valor sugerido por los desarrolladores. Es el tamaño en cantidad de búfers de la cola de reposición del canal.

Copiar un canal

Para agregar un canal nuevo a la lista de canales configurados en el GCA, pero utilizando como base los valores de los atributos de un canal ya existente, se debe elegir el canal base y utilizar la opción “Copiar Canal” del menú “Canales”. Al elegir esta opción, la pantalla dispone todos los cuadros con los atributos completados con los valores del canal que se había elegido pero estableciendo en blanco el nombre. En la parte inferior de la pantalla aparecen dos botones para confirmar o cancelar la operación de inserción del nuevo canal. Se deben completar todos los cuadros (exceptuando los relativos al post-proceso que no son imprescindibles), para luego confirmar la inserción del nuevo canal. La descripción de los atributos del canal se encuentra en la sección anterior (5.4 - Agregar un canal nuevo).

Cambiar atributos de un canal

Para modificar algún atributo de un canal ya existente, se debe elegir el canal a modificar y utilizar la opción “Cambiar Canal” del menú “Canales”. Al elegir esta opción, la pantalla dispone todos los cuadros con los atributos completados con los valores del canal que se había elegido pero deshabilita la opción de cambiar el nombre. En la parte inferior de la pantalla aparecen dos botones para confirmar o cancelar la operación de modificación del canal. La descripción de los atributos del canal se encuentra en la sección anterior (5.4 - Agregar un canal nuevo).

Eliminar un canal

Para eliminar un canal, se debe elegir el canal deseado y utilizar la opción “Eliminar Canal” del menú “Canales”. Al elegir esta opción, la pantalla dispone todos los cuadros con los atributos completados con los valores del canal que se había elegido, pero deshabilitados. En la parte inferior de la pantalla aparecen dos botones para confirmar o cancelar la operación de eliminación del canal.

5.5. Administración de canales

Al seleccionar un canal del cuadro combinado “Elegir Canal” se habilita el menú “Estado”. Este menú contiene las opciones para iniciar o detener el grabador del canal, monitorizar la grabación, e iniciarla o detenerla manualmente.



Fig. 16: Menú “Estado”

Iniciar el funcionamiento de un canal

Si el proceso de grabación del canal seleccionado no está activo, la opción “Iniciar canal” está habilitada y permite iniciar el funcionamiento del canal. Al seleccionar la opción, los procesos relativos a la grabación del canal seleccionado se inician y, si lo indica el cronograma, la grabación comienza.

Detener el funcionamiento de un canal

Si el proceso de grabación del canal seleccionado está activo, la opción “Terminar canal” está habilitada y permite detener el funcionamiento del canal. Al seleccionar la opción, los procesos relativos a la grabación del canal seleccionado se terminan ordenadamente, cerrando el archivo de audio si existe uno abierto.

5.6. Monitorización de grabación y operación manual de un canal

Para monitorizar los niveles del audio que se está adquiriendo en un canal, se debe elegir el canal deseado y utilizar la opción “Monitorizar Canal” del menú “Estado”, que abre un diálogo de monitorización.

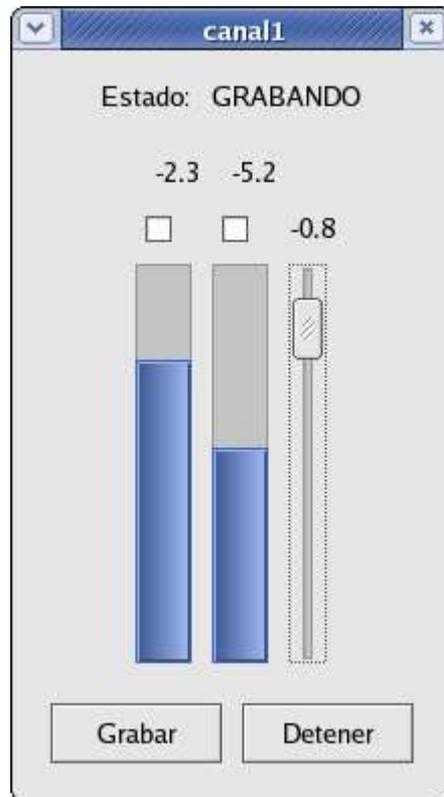


Fig. 17: Diálogo de monitorización de un canal

La ventana de monitorización muestra el estado del grabador (si está grabando o está detenido) y dos indicadores visuales de nivel para representar el canal izquierdo y derecho del audio. Sobre éstos se encuentran dos indicadores que señalan la presencia de saturación en la adquisición y el valor en decibeles del audio adquirido.

Modificar el volumen de adquisición del audio

A la derecha de los indicadores de adquisición se ubica un cuadro deslizable que permite ajustar en tiempo real la ganancia de adquisición. Al modificar la posición del cursor, se modifica el nivel de ganancia asociado al control del mezclador configurado para el canal. Se puede ver en la pantalla la alteración que se produce en el volumen del audio adquirido y ajustar de esta manera el nivel óptimo para evitar saturación.

Iniciar manualmente la grabación

Para inicial manualmente una grabación independiente del cronograma, se debe utilizar el botón “Grabar”, que muestra un diálogo que permite establecer el formato de audio que debe utilizarse para la grabación. Puede utilizarse la función de grabación manual tanto para iniciar una grabación cuando el grabador está detenido, como para modificar el formato de audio utilizado cuando una grabación ya está en curso.



Fig. 18: Diálogo para establecer formato de audio para grabación manual

El cuadro de diálogo permite modificar los atributos del formato de audio a emplear en la grabación y en la parte inferior tiene dos botones para confirmar el inicio de grabación o cancelar la operación. El detalle de los atributos del formato de audio se encuentra en la tabla de la sección 5.7.

Detener manualmente la grabación

Para detener manualmente una grabación en curso, de forma independiente del cronograma, se debe utilizar el botón “Detener”.

5.7. Administración del cronograma de un canal

Al seleccionar un canal del cuadro combinado “Elegir Canal”, si este se encuentra activo, se habilita el menú “Cronograma”. Este menú contiene una opción que abre un diálogo que permite ver o modificar el cronograma del canal.

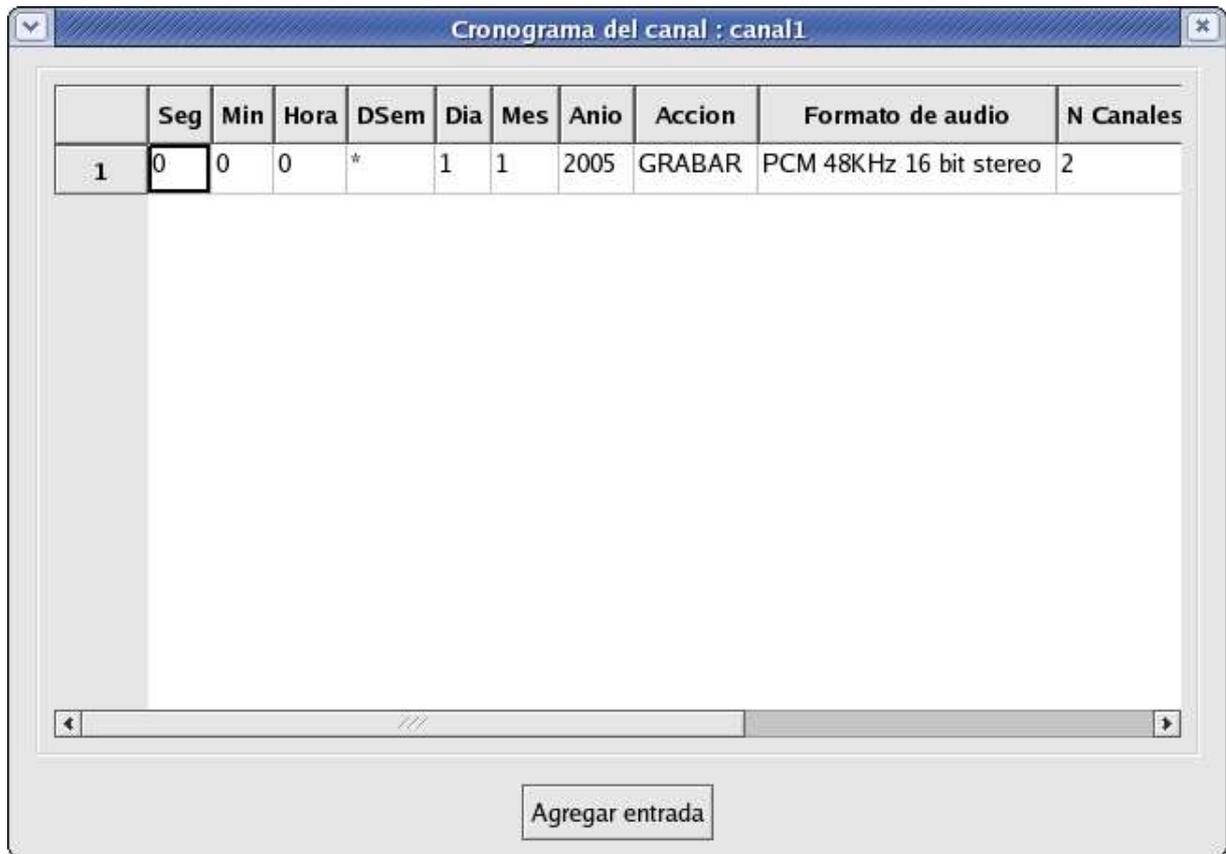


Fig. 19: Diálogo para mostrar y editar el cronograma

El cuadro de diálogo muestra por renglones las entradas existentes en el cronograma. Una entrada se compone de una definición de fecha, una acción (que es una instrucción para iniciar o detener la grabación) y en caso que la acción sea de grabación, un formato de audio. En la parte inferior existe un botón para agregar una entrada nueva al cronograma. Además, la administración se realiza con un menú contextual que aparece al presionar el botón auxiliar del ratón sobre las etiquetas de los renglones.

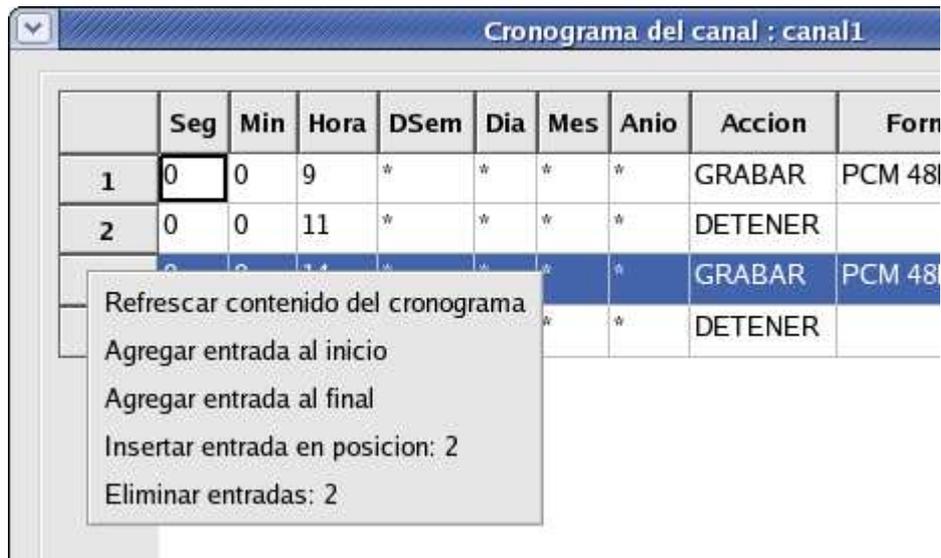


Fig. 20: Menú contextual del cronograma

Cuando el cronograma se modifica, los cambios se aplican de forma inmediata y el proceso encargado de controlar la grabación revisa el nuevo cronograma. El estado actual del grabador no cambia, pero sí puede cambiar el próximo evento programado para el grabador. Por ejemplo, si la modificación realizada al cronograma altera tanto el estado en que debería estar actualmente como el próximo evento a realizar hacia el futuro, el control de grabación corrige la programación para el próximo evento, pero no altera el estado actual del grabador. Si se desea modificar el estado actual se puede realizar manualmente usando el diálogo de monitorización explicado en 5.6.

Agregar una entrada de cronograma

Existen varios modos posibles de agregar una entrada al cronograma. El botón "Agregar entrada" permite agregar una entrada al final del cronograma. El menú contextual permite agregar una entrada al inicio, al final del cronograma, o en una posición intermedia. Para agregar una entrada en una posición intermedia se debe marcar el renglón de la ubicación deseada presionando el botón de la etiqueta del número de renglón y luego abriendo el menú contextual. En este caso aparece una opción adicional que se lee como "Insertar entrada en posición: x".

Al agregar una entrada se genera un nuevo renglón vacío donde se deben completar todas las celdas necesarias. En la tabla se detallan los atributos de una entrada de cronograma (incluye la descripción de una definición de fecha, la acción y el formato de audio), estos datos se citarán como referencia en otras secciones.

Atributo	Descripción
Seg	Número de 0 a 59 que establece el segundo en el cual debe realizarse la acción definida. También se puede completar este atributo con un caracter de asterisco "*" que representa cualquiera de todos los valores posibles, es decir que la acción se realizaría en principio en todos los segundos. Si se deja el atributo en blanco, el sistema asume el valor "*".
Min	Número de 0 a 59 que establece el minuto en el cual debe realizarse la acción definida. También se puede completar este atributo con un caracter de asterisco "*" que representa cualquiera de todos los valores posibles, es decir que la acción se realizaría en principio en todos los minutos. Si se deja el atributo en blanco, el sistema asume el valor "*".
Hora	Número de 0 a 23 que establece la hora en la cual debe realizarse la acción definida. También se puede completar este atributo con un caracter de asterisco "*" que representa cualquiera de todos los valores posibles, es decir que la acción se realizaría en principio en todas las horas. Si se deja el atributo en blanco, el sistema asume el valor "*".
DSem	Número de 1 a 7 que representa el día de la semana de lunes (1) a domingo (7). También puede completarse con asterisco "*" o dejarse en blanco.
Dia	Número de 1 a 31 que representa el día del mes. También puede completarse con asterisco "*" o dejarse en blanco.
Mes	Número de 1 a 12 que representa el mes. También puede completarse con asterisco "*" o dejarse en blanco.
Año	Número de año, debe ingresarse con cuatro dígitos (por ejemplo: 2005). También puede completarse con asterisco "*" o dejarse en blanco.
Acción	Acción que debe realizar el cronograma. Para indicar una detención de la grabación se debe ingresar el caracter "d" o "D". En caso contrario, si el texto ingresado no comienza con la letra "D", o se deja el atributo en blanco, se asume que la acción es un inicio de grabación.
Nombre del formato	Cadena alfanumérica que describe el formato de audio. Puede dejarse este atributo en blanco.

Bits por muestra	Número de bits por muestra de audio, normalmente se utilizan los valores 8 o 16 de acuerdo con las posibilidades de la tarjeta de audio. Si se deja el atributo en blanco, se asumen 16 bits por muestra.
Extensión de archivos	Cadena alfanumérica con la extensión con que se guardan en disco los archivos de audio. Si se deja el atributo en blanco, se asume la extensión “.wav”.
Número de canales	Número que representa si el canal es mono (1) o estéreo (2). Si se deja el atributo en blanco, se asume el valor 2 (estéreo).
Frecuencia de muestreo	Número de muestras por segundo que serán adquiridas por el grabador. Si se deja el atributo en blanco, se asume el valor 48000.

A continuación se presentan algunos ejemplos de definición de fecha para explicar mejor el uso del caracter asterisco “*”.

Seg = 0 Min = 0 Hora = 9 DSem, Dia, Mes, Año = *	El evento puede realizarse todos los días a las 9:00 horas.
Seg = 0 Min = 30 Hora = 9 DSem = 6 Dia, Mes, Año = *	El evento puede realizarse todos los sábados a las 9:30 horas.
Seg = 0 Min = 15 Hora = * DSem = * Dia = 1 Mes, Año = *	El evento puede realizarse el primer día de cada mes y a lo largo de este día, al cumplirse el cuarto de todas las horas (0:15hs, 1:15hs, etc.)

El control de grabación, cuando procesa el cronograma, realiza los cálculos correspondientes para resolver la fechas de las entradas. Cuando existen varios eventos que coinciden en la misma fecha, las acciones de inicio de grabación toman prioridad sobre las de detención. Si existen varios eventos que instruyen inicio de grabación para el mismo momento, se compone un nuevo formato de audio con las mejores características de los eventos coincidentes. En este caso, para cada atributo del formato de audio, se utiliza el mayor de los valores establecidos en los eventos coincidentes (máximo número de canales, bits por muestra y frecuencia de muestreo).

Eliminar una entrada de cronograma

Para eliminar una o varias entradas de cronograma, se debe marcar el o los renglones correspondientes y seleccionar la opción “Eliminar entradas: x” del menú contextual. Para seleccionar varios renglones se debe presionar la tecla “control” y presionar los botones de las etiquetas correspondientes a las entradas a eliminar.

Refrescar la vista del cronograma

La opción “Refrescar contenido del cronograma” vuelve a solicitar el cronograma y completa nuevamente la grilla con el cronograma actual.

6. Manual del Control de Fallas

6.1. Descripción del funcionamiento

El sistema de Control de Fallas se comunica utilizando CORBA con el sistema principal del GCA. El sistema de Control de Fallas funciona a través de la red de datos por lo que puede estar ejecutándose en una computadora independiente del sistema principal del GCA.

Periódicamente, el Control de Fallas consulta al GCA si existen errores de funcionamiento. Si el GCA informa que hay errores de funcionamiento, el Control de Fallas recupera la lista de errores y la guarda en un archivo de texto plano en un directorio local. El Control de Fallas puede configurarse para que inicie programas externos ante la detección de fallas.

Descripción de programas externos

El Control de Fallas puede programarse con cinco programas externos que se ejecutan ante distintas condiciones de estado. Cuando la operación del GCA es normal, no se ejecutan programas externos.

- Programas que se ejecutan ante errores de funcionamiento

Cuando se reporta un error de funcionamiento, se ejecuta un programa externo configurado por el usuario. Si en las próximas consultas no se reporta un error diferente al que ya había sido informado, el programa externo no se vuelve a ejecutar. También puede configurarse un segundo programa para ejecución recurrente. El sistema ejecutará este segundo programa cada vez que la consulta de estado informe que existen errores.

- Programas que se ejecutan ante pérdidas de conexión

Ante la detección de una pérdida de comunicación con el sistema principal (en caso que la consulta de estado no sea respondida), se ejecutan otros dos programas, uno de forma única y el otro de forma recurrente mientras se mantenga la situación de falta de conexión.

- Programa que se ejecuta al retomar el funcionamiento correcto

Cuando se detecta que el sistema principal GCA está funcionando sin errores luego que se habían reportado errores o se había perdido la comunicación, se ejecuta por única vez un programa de aviso de funcionamiento correcto.

6.2. Descripción del archivo de configuración

Ejemplo de archivo de definición (gcactrlfallas_conf.xml)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE boost_serialization>
<boost_serialization signature="serialization::archive" version="3">
<arch_ctrl_fallas class_id="0" tracking_level="0" version="0">
  <args_corba>-ORBInitRef NameService=corbaloc::
    localhost:12456/NameService</args_corba>
  <dir_exe>/usr/libexec/gca</dir_exe>
  <arch_errores>/var/lib/gca/errores.txt</arch_errores>
```

```

    <comando_alarma_ok_exe>/bin/bash</comando_alarma_ok_exe>
    <comando_alarma_ok_args>/var/lib/gca/ok
</comando_alarma_ok_args>
    <comando_alarma_exe></comando_alarma_exe>
    <comando_alarma_args></comando_alarma_args>
    <comando_alarma_rec_exe></comando_alarma_rec_exe>
    <comando_alarma_rec_args></comando_alarma_rec_args>
    <comando_alarma_sincom_exe>/bin/programa
</comando_alarma_sincom_exe>
    <comando_alarma_sincom_args>argumentos
</comando_alarma_sincom_args>
    <comando_alarma_sincom_rec_exe>/usr/bin/perl
</comando_alarma_sincom_rec_exe>
    <comando_alarma_sincom_rec_args>/ruta/script_perl
</comando_alarma_sincom_rec_args>
    <tiempo_control>5</tiempo_control>
</arch_ctrl_fallas>
</boost_serialization>

```

Descripción de las claves

args_corba

Esta clave contiene una cadena que es la lista de parámetros que se requieren para conectarse al servicio de nombres de CORBA (NSD). La cadena incluye el servidor donde está ejecutándose el NSD y el puerto TCP al que está asignado el servicio. En el ejemplo, el servidor es local (localhost) y el puerto es 12456. El servidor “localhost” es apropiado si el Control de Fallas se está ejecutando en la misma computadora que el servicio de nombres de CORBA. De lo contrario debe configurarse la dirección IP de la computadora donde se está ejecutando el servidor de nombres o un nombre que se resuelva correctamente a dicha dirección. El puerto configurado debe coincidir con el establecido durante la configuración de CORBA.

arch_errores

Esta clave es una cadena que apunta a la ruta completa y nombre del archivo de errores generado por el proceso de control de fallas.

claves comando_*_exe y comando_*_args

Las claves “_exe” son una cadena que apunta a la ruta completa y nombre del archivo a ejecutar según la situación de falla detectada. Si se requieren parámetros adicionales se registran en la clave “_args” correspondiente. Se puede ejecutar la ruta a un intérprete para luego enviarle como parámetro el archivo a ejecutar. De este modo se puede utilizar un intérprete de línea de comandos para procesar un archivo de procesamiento por lotes, o un intérprete de lenguaje Perl para ejecutar un “script” en Perl, etc.

La tabla detalla en qué casos se ejecuta cada comando:

Clave	Situación en que se utiliza
-------	-----------------------------

comando_alarma	Se ejecuta al detectar por primera vez una falla nueva
comando_alarma_rec	Se ejecuta cada vez que se detecta una falla aunque ya se haya detectado la misma falla en la consulta anterior
comando_alarma_sincom	Se ejecuta la primera vez que se detecta que no hay comunicación con el GCA
comando_alarma_sincom_rec	Se ejecuta toda vez que se detecta que no hay comunicación con el GCA
comando_alarma_ok	Se ejecuta cuando se restablece el funcionamiento correcto del GCA luego de que habían errores o se había perdido la comunicación.

tiempo_control

Esta clave establece la frecuencia en segundos con que el sistema de control de fallas consulta al GCA su estado de funcionamiento.

7. Interfaz Web para la búsqueda del audio

Si se realizaron las configuraciones según lo explicado en la sección 4.5 y el servidor de páginas web está funcionando correctamente, se puede acceder a la página de búsqueda del audio. La misma tiene un presentación como la mostrada a continuación y permite establecer las condiciones de búsqueda por canal y por fecha.

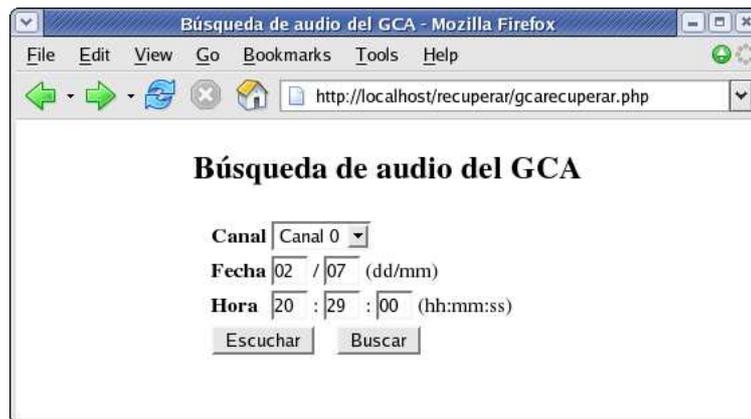


Fig. 21: Pantalla de búsqueda del audio

Utilizando el botón “Escuchar” se descarga un archivo de secuencia de reproducción en formato SMIL para ser utilizado por reproductores con este soporte.

Utilizando el botón “Buscar” se accede a una página con hipervínculos que permite descargar los archivos de audio a la computadora del cliente.

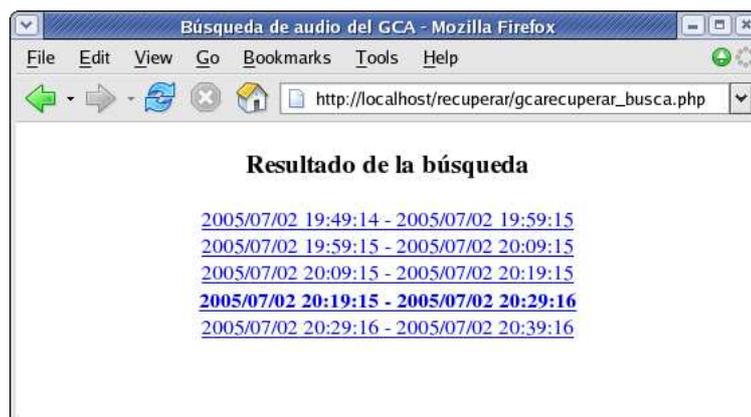


Fig. 22: Pantalla de resultados de la búsqueda del audio

20. Descripción de archivos del sistema

Los archivos de configuraciones utilizados por los diferentes módulos del GCA tienen formato XML y son el resultado de la persistencia de las clases relevantes del sistema. La generación de los archivos XML se realiza automáticamente con la biblioteca de persistencia de Boost. A continuación se detalla la estructura de los archivos del sistema y el significado de cada clave.

También se adjunta la descripción de los archivos externos, cuándo se ejecutan y qué parámetros toman de la línea de comando.

1. Archivo de configuraciones del GCA

1.1. Propósito

El archivo de configuraciones del GCA contiene la información esencial para la operación del sistema principal, como los parámetros para la comunicación CORBA de los procesos y la lista de canales definidos.

1.2. Claves

args_corba

Esta clave contiene una cadena que es la lista de parámetros que se requieren para conectarse al servicio de nombres de CORBA (NSD). La cadena incluye el servidor donde está ejecutándose el NSD y el puerto TCP al que está asignado el servicio. En el ejemplo el servidor es local (localhost) y el puerto es 12456.

dir_exe

Directorio donde se ubican los programas del sistema (exceptuando el programa principal que opera como servicio, gcctrlproc). Este parámetro se utiliza para la ejecución en Microsoft Windows, es el directorio de instalación de los programas como el grabador (gcgrabador) o el control de grabación (gcctrlgrab).

canales

Esta clave inicia la lista de definición de canales. La clave 'count' indica la cantidad total de canales definidos. Cada canal se enmarca dentro de las claves 'item' y tiene una lista de atributos que se detallan a continuación.

nombre

Esta clave contiene una cadena que es el nombre del canal.

id_disp_audio

Esta clave contiene según el sistema operativo la identificación del acceso a la tarjeta de audio para el canal.

id_disp_mezclador

Esta clave contiene según el sistema operativo la identificación del acceso al mezclador de la tarjeta de audio para el canal.

id_control

Esta clave es un número que identifica el control del mezclador que regula el volumen de adquisición del canal.

dir

Esta clave es una cadena que apunta a la ruta completa del directorio donde deben guardarse los archivos de audio WAV adquiridos para el canal. La sintaxis depende del sistema operativo.

comando_postproc_exe

Esta clave es una cadena que apunta a la ruta completa y nombre del archivo a ejecutar cuando se finaliza la grabación a disco de un archivo de audio. Adicionalmente la siguiente clave se envía como parámetro.

comando_postproc_args

Esta clave es una cadena que se envía como parámetro al archivo ejecutable definido antes, además se anexa como parámetro final el nombre del archivo recién guardado.

duracion_arch

La duración máxima en segundos que tiene cada archivo de audio WAV.

min_kiB_libres

La mínima cantidad de kilobytes (kiB) libres que debe tener la unidad referida por la clave 'dir' para que la grabación a disco prosiga sin alteraciones.

tam_maxCola_buf

El tamaño máximo en cantidad de búfers que pueden alcanzar las colas de adquisición y grabación. Si se alcanza este valor, el sistema descarta el búfer más antiguo al incorporar un búfer nuevo.

tam_alarmaCola_buf

El tamaño en cantidad de búfers para las colas de adquisición y grabación a partir del cual el sistema reporta una falla en el canal. La grabación no obstante prosigue y el tamaño máximo que pueden alcanzar las colas está dado por el valor de la clave 'tam_maxCola_buf'.

tamColaRepoBuf

El tamaño en cantidad de búfers de la cola de reposición del canal.

habilitado

Esta clave toma valores 0 o 1. El valor 1 indica que el canal debe iniciarse automáticamente al iniciarse el sistema.

1.3. Archivo de definición (gca_conf.xml)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE boost_serialization>
<boost_serialization signature="serialization::archive"
version="3">
<arch_conf class_id="0" tracking_level="0" version="0">
  <args_corba>-ORBInitRef NameService=corbaloc::localhost:12456
/NameService</args_corba>
  <dir_exe>/usr/libexec/gca</dir_exe>
  <canales class_id="1" tracking_level="0" version="0">
    <count>1</count>
    <item class_id="2" tracking_level="0" version="0">
      <nombre>canal0</nombre>
      <id_disp_audio>/dev/dsp0</id_disp_audio>
      <id_dispmezclador>/dev/mixer0</id_dispmezclador>
      <id_control>6</id_control>
      <dir>/home/maria/audio0</dir>
      <comando_postproc_exe>/bin/bash
</comando_postproc_exe>
      <comando_postproc_args>/var/lib/gca/comprimir.sh
</comando_postproc_args>
      <duracion_arch>600</duracion_arch>
      <min_kiB_libres>500</min_kiB_libres>
      <tam_maxColaBuf>2000</tam_maxColaBuf>
      <tam_alarmaColaBuf>1000</tam_alarmaColaBuf>
      <tamColaRepoBuf>1000</tamColaRepoBuf>
      <habilitado>1</habilitado>
    </item>
  </canales>
</arch_conf>
</boost_serialization>
```

2. Cronograma de un canal**2.1. Propósito**

Para cada canal existe un cronograma que se guarda en el directorio del canal junto con los archivos de audio.

2.2. Claves de un elemento de cronograma

El cronograma está representado por la clave 'vector_cronograma' y está compuesto por una lista de elementos de cronograma cada uno de ellos enmarcado por la clave 'item'. La cantidad de elementos de cronograma es variable y está dada por el valor configurado en la clave 'count'. Cada elemento de cronograma se compone esencialmente por una acción, un formato de audio (en caso que la acción indique grabar, de lo contrario el formato de audio se omite) y una definición de fecha.

accion

Esta clave contiene otra de igual nombre que puede tomar los valores 0 o 1. El valor 0 indica que el elemento de cronograma es una indicación de grabación, mientras que el valor 1 representa una entrada con instrucción de detención de la grabación.

formato_audio

Esta clave contiene una estructura con múltiples claves que define el formato de audio con que debe programarse la tarjeta adquisidora. Esta clave está presente si la acción fue establecida en 0 (grabación) de lo contrario se omite la totalidad de la clave y su contenido. En 2.3 se detallan las claves del formato de audio.

fechas

Esta clave contiene una estructura con múltiples claves que define el momento en que debe realizarse la tarea indicada por este elemento de cronograma. La indicación no necesariamente es una fecha absoluta ya que el cronograma tiene una concepción similar al programador de tareas 'cron' de Linux. Esta clave es obligatoria en todos los elementos de cronograma y en 2.4 se detallan las claves que integran la misma.

2.3. Claves de un elemento formato_audio

nombre

Cadena de caracteres que define el nombre descriptivo del formato de audio

num_canales

Cantidad de canales a adquirir. Se establece 1 para mono y 2 para estéreo.

frec_muestreo

Frecuencia de muestreo en kHz, por ejemplo 48000.

bits_por_muestra

Cantidad de bits por cada muestra, por ejemplo 16.

extension_arch

Extensión con que se guardan en disco los archivos PCM de formato WAV (normalmente ".wav")

2.4. Claves de un elemento fechas

Cualquiera de las claves de fecha puede contener un valor numérico adecuado o un asterisco (*) que representa la condición “todos los valores válidos” que pueda tomar la clave. Las claves “segundo”, “minuto” y “hora” toman valores de 0 a 59. La clave “dia_mes” representa el día del mes del 1 al 31 y la clave “mes” el número de mes del 1 al 12. La clave “año” es el valor de año válido en cuatro dígitos (por ejemplo 2005). Finalmente, la clave “dia_semana” representa el día de la semana de lunes (valor 1) a domingo (valor 7).

2.5. Archivo de definición (canal1.xml)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE boost_serialization>
<boost_serialization signature="serialization::archive"
version="3">
<cronograma class_id="0" tracking_level="0" version="0">
  <vector_cronograma class_id="1" tracking_level="0"
version="0">
    <count>2</count>
    <item class_id="2" tracking_level="0" version="0">
      <accion class_id="3" tracking_level="0"
version="0">
        <accion>0</accion>
      </accion>
      <formato_audio class_id="4" tracking_level="0"
version="0">
        <nombre>PCM 48KHz 16 bit stereo</nombre>
        <num_canales>2</num_canales>
        <frec_muestreo>48000</frec_muestreo>
        <bits_por_muestra>16</bits_por_muestra>
        <extension_arch>.wav</extension_arch>
      </formato_audio>
      <fechas class_id="5" tracking_level="0" version="0">
        <segundo>0</segundo>
        <minuto>0</minuto>
        <hora>0</hora>
        <dia_semana>*</dia_semana>
        <dia_mes>*</dia_mes>
        <mes>*</mes>
        <anho>*</anho>
      </fechas>
    </item>
    <item>
      <accion>
        <accion>1</accion>
      </accion>
      <fechas>
```

```
<segundo>0</segundo>
<minuto>0</minuto>
<hora>23</hora>
<dia_semana>*</dia_semana>
<dia_mes>*</dia_mes>
<mes>*</mes>
<anho>*</anho>
</fechas>
</item>
</vector_cronograma>
</cronograma>
</boost_serialization>
```

3. Archivo de configuraciones del Control de Fallas

3.1. Propósito

El archivo de configuraciones del Control de Fallas del GCA contiene la información esencial para la operación del sistema de control de fallas, como los parámetros para la comunicación CORBA de los procesos y la ruta hacia los archivos de comando a ejecutar en caso de fallas.

3.2. Claves

args_corba

Esta clave contiene una cadena que es la lista de parámetros que se requieren para conectarse al servicio de nombres de CORBA (NSD). La cadena incluye el servidor donde está ejecutándose el NSD y el puerto TCP al que está asignado el servicio. En el ejemplo el servidor es local (localhost) y el puerto es 12456.

arch_errores

Esta clave es una cadena que apunta a la ruta completa y nombre del archivo de errores generado por el proceso de control de fallas.

claves comando_*_exe y comando_*_args

Las claves “_exe” son una cadena que apunta a la ruta completa y nombre del archivo a ejecutar según la situación de falla detectada. Si se requieren argumentos adicionales se registran en la clave “_args” correspondiente. Se puede ejecutar la ruta a un intérprete para luego enviarle como parámetro el archivo a ejecutar. De este modo se puede utilizar un intérprete de línea de comandos para procesar un archivo de procesamiento por lotes, o un intérprete de lenguaje Perl para ejecutar un “script” en Perl, etc.

La tabla detalla en qué casos se ejecuta cada comando:

Clave	Situación en que se utiliza
comando_alarma	Se ejecuta al detectar por primera vez una falla nueva
comando_alarma_rec	Se ejecuta cada vez que se detecta una falla aunque ya se haya detectado la misma falla en la consulta anterior
comando_alarma_sincom	Se ejecuta la primera vez que se detecta que no hay comunicación con el GCA
comando_alarma_sincom_rec	Se ejecuta toda vez que se detecta que no hay comunicación con el GCA
comando_alarma_ok	Se ejecuta cuando se restablece el funcionamiento correcto del GCA luego de que habían errores o se había perdido la comunicación.

tiempo_control

Esta clave establece la frecuencia en segundos con que el sistema de control de fallas consulta al GCA si está funcionando correctamente.

3.3. Archivo de definición (gcactrfallas_conf.xml)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE boost_serialization>
<boost_serialization signature="serialization::archive" version="3">
<arch_ctrl_fallas class_id="0" tracking_level="0" version="0">
  <args_corba>-ORBInitRef NameService=corbaloc::
    localhost:12456/NameService</args_corba>
  <dir_exe>/usr/libexec/gca</dir_exe>
  <arch_errores>/var/lib/gca/errores.txt</arch_errores>
  <comando_alarma_ok_exe>/bin/bash</comando_alarma_ok_exe>
  <comando_alarma_ok_args>/var/lib/gca/ok
</comando_alarma_ok_args>
  <comando_alarma_exe></comando_alarma_exe>
  <comando_alarma_args></comando_alarma_args>
  <comando_alarma_rec_exe></comando_alarma_rec_exe>
  <comando_alarma_rec_args></comando_alarma_rec_args>
  <comando_alarma_sincom_exe>/bin/programa
</comando_alarma_sincom_exe>
  <comando_alarma_sincom_args>argumentos
</comando_alarma_sincom_args>
  <comando_alarma_sincom_rec_exe>/usr/bin/perl
</comando_alarma_sincom_rec_exe>
  <comando_alarma_sincom_rec_args>/ruta/script_perl
</comando_alarma_sincom_rec_args>
  <tiempo_control>5</tiempo_control>
</arch_ctrl_fallas>
```

</boost_serialization>

BIBLIOGRAFÍA

- [1] Grady Booch. Object-oriented analysis and design with applications. 2nd ed. Redwood City, CA : Benjamin/Cummings, 1994.
- [2] Ivar Jacobson. Object-oriented software engineering : a use case driven approach. Wokingham : Addison-Wesley, 1993.
- [3] James Rumbaugh. Object-oriented modeling and design. Englewood Cliffs, NJ : Prentice-Hall, 1991.
- [4] Apache Software Foundation. Apache Software Foundation Homepage. <http://www.apache.org>, c1999-2005.
- [5] Realnetworks. Helix Community. <http://helixcommunity.org>, c1995-2005.
- [6] W3C. SMIL. <http://www.w3.org/AudioVideo>, c2003.
- [7] H. Schulzrinne ; A. Rao ; R. Lanphier. Real Time Streaming Protocol (RTSP). Network Working Group, RFC 2326. <http://www.ietf.org/rfc/rfc2326.txt?number=2326>, April 1998.
- [8] Object Management Group. The OMG's Corba Website. <http://www.corba.org>, c1997-2005.
- [9] ObjectSecurity, Mico Project Team. MICO CORBA. <http://www.mico.org>, c1996-2004.
- [10] Beman Dawes ; David Abrahams. Boost C ++ Libraries. <http://www.boost.org>, c1998-2003, rev. Oct. 2004.
- [11] Anthemion Software. wxWidgets : cross-platform GUI library. <http://www.wxwidgets.org/>, c.2004.
- [12] Realnetworks. RealPlayer 10 : the best audio and video player. <http://mexico.real.com/player>, c2005.
- [13] The PHP Group. PHP : hypertext preprocessor. <http://www.php.net>, c2001-2005.
- [14] Owl River Company. RPM Homepage. <http://www.rpm.org>, c2002.
- [15] Jordan Russell. Inno Setup. <http://www.jrsoftware.org/isinfo.php>, c1997-2005.
- [16] OProfile Developers. OProfile : a system profiler for Linux. <http://oprofile.sourceforge.net>, 21/07/2005 (ult. rev.).
- [17] Los Alamos National Laboratory (W-7405-ENG-36). C Style and Coding Standards for the SDM Proyect. http://iie.fing.edu.uy/ense/asign/str/str/SDM_C_Style_Guide.html, 01/07/2005 (ult. rev.).

-
- [18] Linus Trovalds. Linux Kernel Coding Style. http://www.lnl.gov/linux/slurm/coding_style.pdf, 01/07/2005 (ult. rev.).
- [19] Bjarne Stroustrup. The C++ programming language. 3rd ed. Wokingham : Addison-Wesley, 1997.
- [20] IBM. Rational : software. <http://www-306.ibm.com/software/rational/>, c2005.
- [21] KDevelop. KDevelop : Un entorno de desarrollo integrado. Tr. Alejandro Exojo, Miguel Angel De Blas Burdalo. <http://www.kdevelop.org>, c1999-2005.
- [22] Microsoft Corporation. Microsoft Visual Studio Developer Center. <http://msdn.microsoft.com/vstudio>, c2005.
- [23] Collabnet. Subversion. <http://subversion.tigris.org>, c2005.
- [24] Collabnet. Tortoissvn. <http://tortoissvn.tigris.org>, c2005.
- [25] Meld : Diff and merge tool. Meld Homepage. <http://meld.sourceforge.net>, 01/07/2005 (ult. rev.).
- [26] WinMerge. WinMerge : users. <http://winmerge.sourceforge.net/index.php>, [s.d.].
- [27] Lyx. LyX : the document processor. <http://www.lyx.org>, 17/07/2005 (ult. rev.).
- [28] Dimitri van Heesch. Doxygen. <http://www.doxygen.org>, c1997-2005.