

Capítulo 4

Diseño e Implementación de MexVox Versión 2.0

4.1 Reingeniería

Como ya se comentó, el sistema DosVox para invidentes tiene 10 años de haber empezado a desarrollarse. En ese entonces fue pensado como un sistema que fuera plano en su totalidad, es decir, que no tuviera ningún tipo de componente gráfico, por lo que se diseñó como un sistema similar a la línea de comandos de MS-DOS, basado en su totalidad en los comandos *write* y *readline*, adaptados a las personas con discapacidad visual, dándoles sonido a través de un sintetizador, o de archivos de sonido pregrabados.

Actualmente los sistemas computacionales, en su mayoría, están diseñados con ventanas y formas que hagan más fácil su manejo y el acceso a las funciones. Normalmente éstas ventanas están compuestas de un conjunto de componentes gráficos, como son los campos de texto, botones, *checkboxes*, *checklists*, entre otros, y que tienen la característica de que el usuario puede controlarlos con la ayuda del *mouse* sin excluir al teclado.

Dados estos cambios en el desarrollo y diseño de los sistemas computacionales a través del tiempo, nuestra propuesta fue el desarrollo de un sistema similar a DosVox, haciendo

un proceso de reingeniería de este sistema, usando herramientas actuales y que sea adaptable para los discapacitados visuales. Para este fin se han tomado en cuenta las mismas especificaciones de DosVox, ya que el sistema ha tenido mucho éxito, aunque para el nuevo sistema se realizó un nuevo análisis, cambiando ligeramente algunas funciones.

Se realizó un diseño distinto al de DoxVox incluyéndose nuevas clases, componentes y formas adecuadas para que un usuario invidente pueda manejarlo. Cabe aclarar que el sistema no elimina los componentes gráficos, sino que les añade componentes auditivos con la finalidad de que el usuario invidente pueda tener interacción con ellos. De esta forma los componentes gráficos emiten sonidos que hacen que el sistema esté en comunicación con el usuario, que el discapacitado visual sepa lo que está ocurriendo en los componentes y que éstos no sean un obstáculo sino una ayuda para el manejo del sistema.

Como se explicó en la sección 3.3. Este proceso de reingeniería es válido e importante ya que DosVox ha sido modificado constantemente, dando como resultado una serie de códigos difíciles de entender y con nula documentación, lo que hace difícil que el sistema sea adaptable a un nuevo idioma e incluso a nuevas tecnologías que pueden ser agregadas, como por ejemplo integrar reconocimiento de voz al sistema.

Esta segunda versión de MexVox incluye sólo un subconjunto de programas de DosVox. La selección de los programas fue analizada por el grupo de servicio social y finalmente

se eligieron los módulos que se consideraron más útiles para empezar una capacitación a personas con discapacidad visual. Asimismo, se buscó formaran una base sólida para que posteriormente seguir con el proyecto desarrollando los módulos restantes y añadiéndoles nuevas funciones al sistema optimizandolo.

Los códigos desarrollados en este proyecto han sido re-diseñados en su totalidad, desarrollando programas con interfaces similares a las de un sistema para videntes. Además, a diferencia de DosVox, ésta segunda versión de MexVox cuenta con documentación para que otras personas puedan continuar el desarrollo del sistema. Se espera que en un futuro se logre tener un sistema tan o más útil y completo que el DosVox original de Brasil, que sea en su totalidad en español, que este bien documentado y que de la misma forma ayude a las invidentes en México.

4.2 Herramientas Utilizadas

4.2.1 Delphi

Para el desarrollo de los códigos de MexVox2.0 se decidió usar Delphi [Borland, 1994], ya que es la herramienta para desarrollo de programas oficial del proyecto DosVox en Brasil y ahora en México. Se usaron la versiones Delphi 6 y Delphi 7. Cabe aclarar que Delphi es una herramienta de desarrollo, no un lenguaje; el lenguaje que usa Delphi es Object Pascal, una evolución de Pascal que lo convirtió en un lenguaje orientado a objetos.

4.2.2 GoldWave

Para la grabación de sonidos del sistema se utilizó el sistema GoldWave [GoldWave, 1998], ya que es un sistema de grabación de sonidos gratuito que se encuentra disponible en Internet. Los sonidos fueron grabados con los siguientes valores:

- *Sample Rate* (número de muestras por segundo): 22, 050 Hz.
- *Bit Depth* (número de bits por muestra): 16 bits.
- Canal: Stereo.

Éstos valores proporcionan al archivo de sonido, una calidad de audio parecida a la de la radio, al cual es aceptable para los fines de este sistema.

4.2.3 SAPI

Speech Application Programming Interface (SAPI) es la plataforma de Microsoft sobre la cual corren la mayoría de las aplicaciones de síntesis y reconocimiento de voz y de telefonía. Más información sobre esta plataforma se encuentra en la sección 2.4 del Capítulo 2 de éste documento.

4.2.4 Lernout and Hauspie TTS3000

Lernout & Hauspie TTS3000 [Caragol, 2003], es el sintetizador elegido para usar en éste sistema por ser el de mejor calidad de los sintetizadores gratuitos que fueron probados. La tabla 4.1 muestra una tabla comparación de varios sintetizadores. Todos los productos mostrados en la tabla son sintetizadores que se encuentran disponibles en español latinoamericano y que usan la plataforma SAPI para su ejecución.

Tabla 4.1. Tabla Comparativa de Sintetizadores.

Producto	Precio	Sistema Operativo	Calidad
AT&T Natural Voices text-to-speech engine (AT&T)	149 dls.	Windows 95 en adelante, LINUX	Muy Buena
DECTalk Software (Fonix)	200 dls.	Windows 95 en adelante, LINUX	Aceptable
L&H TTS3000 (Microsoft)	gratis	Windows 95 en adelante	Buena
Loquendo TTS (Loquendo)	350 dls.	Windows 95 en adelante, LINUX, Solaris	Buena
L&H RealSpeak (Scan Soft)	no comercial	Windows 95 en adelante	Muy Buena
L&H TruVoice (Scan Soft)	gratis	Windows 95 en adelante	Insuficiente
Via Voice OutLoud (IBM)	295 dls.	Windows 95 en adelante	Buena

Las muestras de voz de los sintetizadores de la tabla 4.1 y las ligas a las páginas del distribuidor correspondiente a cada uno se encuentran en la página de la Fundación Privada Catalana para la orientación y soporte Tecnológico de los Ciegos Manuel A. Caragol [Caragol, 2003].

4.3 Organización de Archivos

La carpeta principal se llama MexVox2.0. Ésta debe ser instalada en el disco duro C:\ y ahí están contenidos todos los archivos, tanto códigos, documentos y archivos de sonido que el sistema necesita. Dentro de la Carpeta MexVox2.0 se encuentran las carpetas Codigos, Imagenes, Sintesis y Sonidos (Figura 4.1).

La carpeta de “Codigos” a su vez tiene una carpeta por cada programa del sistema, que tiene los códigos y el ejecutable de cada programa. La carpeta “Imagenes” tiene imágenes que usa la ventana principal del sistema. La carpeta “Sintesis” tiene dentro dos ejecutables de instalación; “SPCHAPI”, que instala los componentes necesarios de la plataforma SAPI para ejecutar procesos que usan los sintetizadores y reconocedores de voz, y el archivo “lhttspe” que instala el sintetizador Lernout and Hauspie TTS3000 en español latinoamericano. La carpeta “Sonidos” contiene también una carpeta por cada programa del sistema, pero aquí se encuentran los sonidos pregrabados que usa cada uno de los programas.

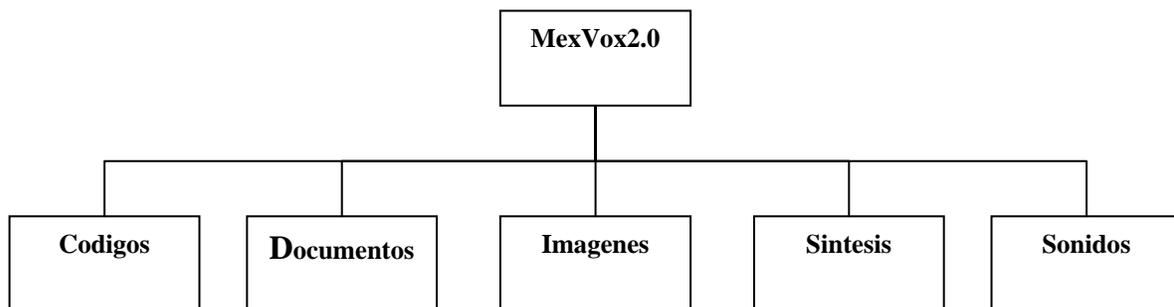


Figura 4.1. Arquitectura de las Carpetas de MexVox V2.0

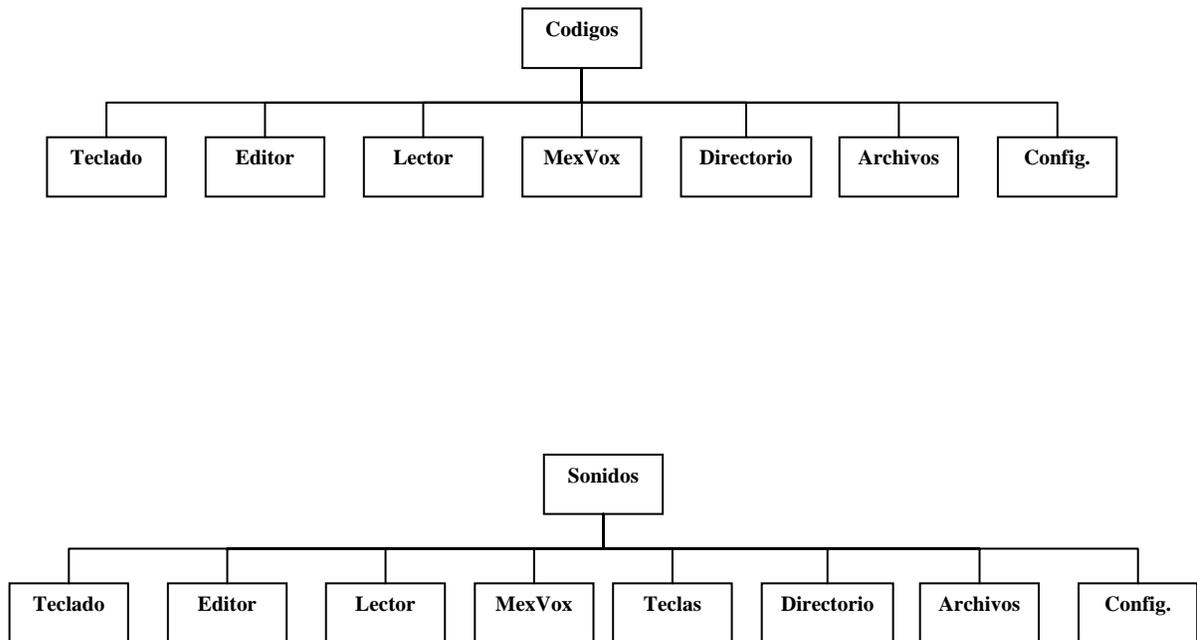


Figura 4.1. Arquitectura de las Carpetas de MexVox V2.0 (continuación)

4.4 Clases Desarrolladas

A continuación se describen las clases que fueron desarrolladas para MexVox V2.0.

MexVox es la clase principal. Engloba todas las clases del sistema, permitiendo su entrada, salida, selección de opciones y la creación de instancias de clases correspondientes.

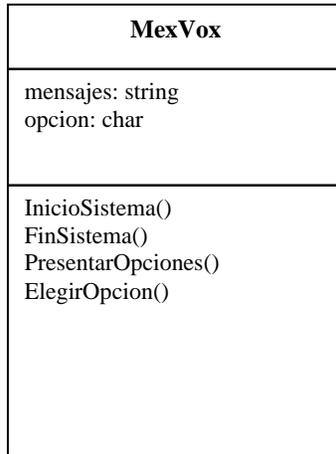


Figura 4.2. Diagrama de la Clase MexVox

La clase **teclado** tiene las funciones del programa más básico del sistema. Ésta clase presenta una forma que reconoce y despliega en pantalla el valor de la tecla que el usuario oprime.

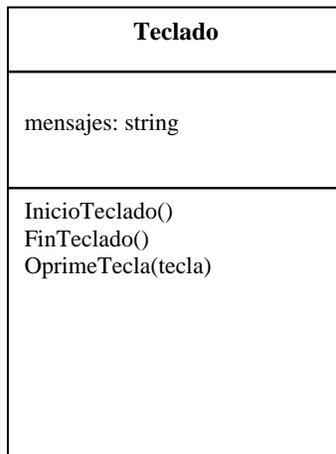


Figura 4.3. Diagrama de la clase Teclado

La clase **teclas** maneja los métodos que se pueden hacer con una tecla una vez que es presionada por el usuario, como obtener su valor, obtener su código, emitir el sonido correspondiente a esa tecla.

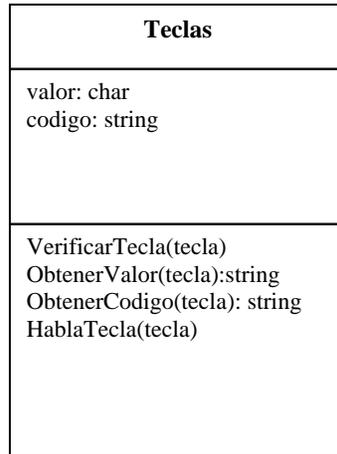


Figura 4.4. Diagrama de la clase Teclas

La clase **editor** contiene las funciones que el usuario puede realizar cuando está editando un texto, las acciones que se pueden hacer con el archivo (abrir, crear, guardar) así como acciones de edición que se hacen sólo sobre una parte del texto (copiar, cortar, pegar, buscar una palabra).



Figura 4.5. Diagrama de la clase Editor

La clase **lector** tiene los métodos que dan la oportunidad al usuario de elegir la forma en que éste deseé leer el documento, así como abrirlo o seleccionarlo.

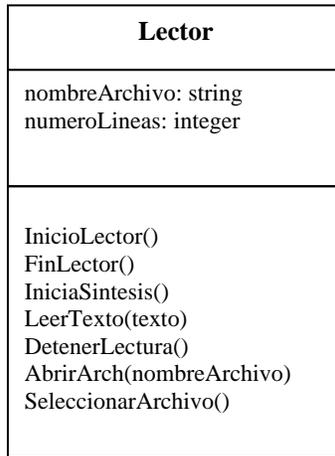


Figura 4.6. Diagrama de la clase Lector

Mensajes es la clase que se encarga de administrar los mensajes pregrabados de cada código. Para hacer esto más fácil existe una clase de este tipo por cada programa, y contiene sólo los mensajes del programa correspondiente, determina a través del código del mensaje cuál es el mensaje grabado y el mensaje escrito que le corresponde, también determina en que Label de la pantalla desplegar el mensaje escrito.

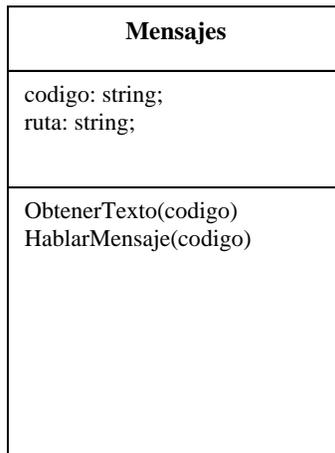


Figura 4.7. Diagrama de la clase Mensajes

La clase **ArchivosDirectorios** tiene las funciones para que el usuario pueda administrar archivos así como navegar por las carpetas que se encuentren en la computadora.



Figura 4.8. Diagrama de la clase ArchivosDirectorios

ConfiguracionHabla es la clase que accesa el archivo de configuración de habla, archivo que contiene tres valores enteros que pertenecen al número de voz, la velocidad y la tonalidad del sintetizador. La clase de las opciones al usuario de consultar los datos o de modificarlos.

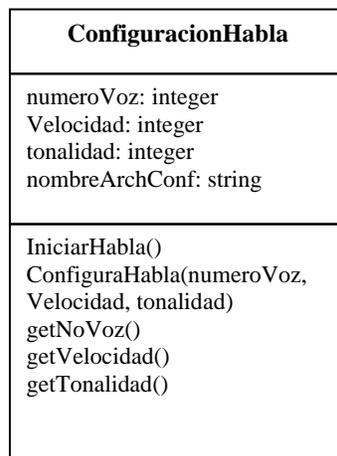


Figura 4.9. Diagrama de la clase ConfiguracionHabla

DirectorioTelefonico da las opciones que tiene el Directorio Telefónico al usuario, que por el momento son Agregar un contacto y hacer búsquedas.

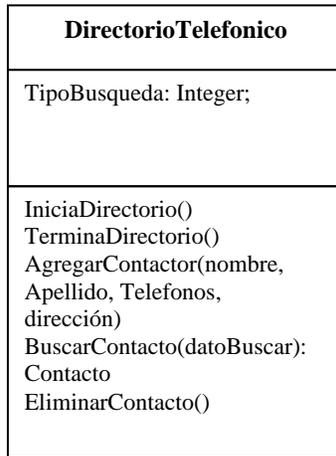


Figura 4.10. Diagrama de la clase DirectorioTelefonico

DirectorioTelefonico usa la clase **Contacto**, que contiene los atributos del contacto, así como los métodos de agregarContacto, eliminarContacto y buscarContacto.

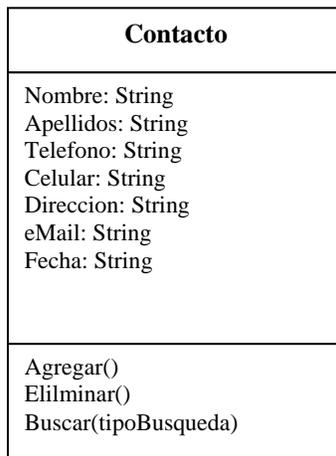


Figura 4.11. Diagrama de la clase Contacto

4.5 Relación entre clases

Las relaciones entre las clases creadas y explicadas en los párrafos anteriores están descritas por los siguientes Diagramas UML mostrado en la Figura 4.12.

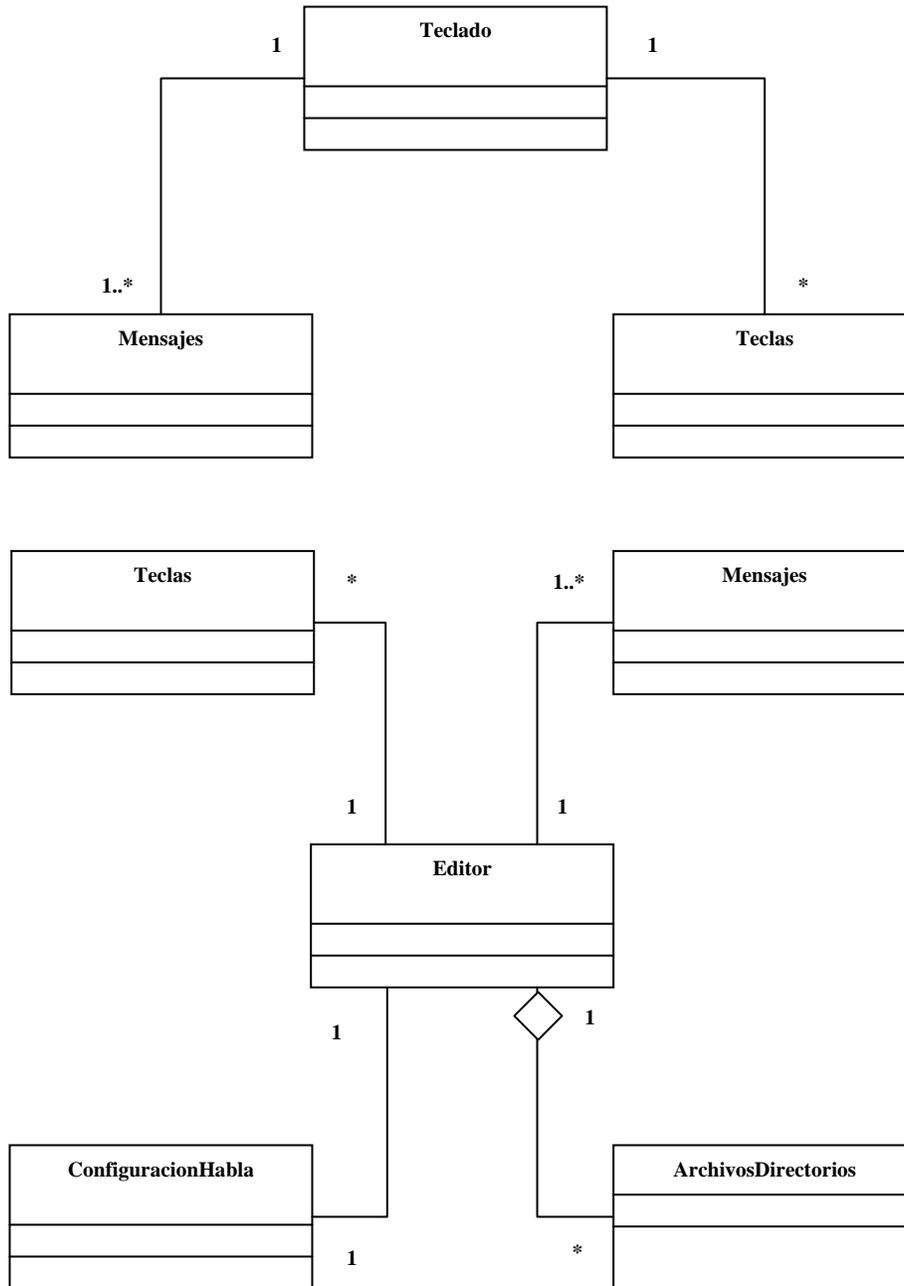


Figura 4.12. Relación entre clases

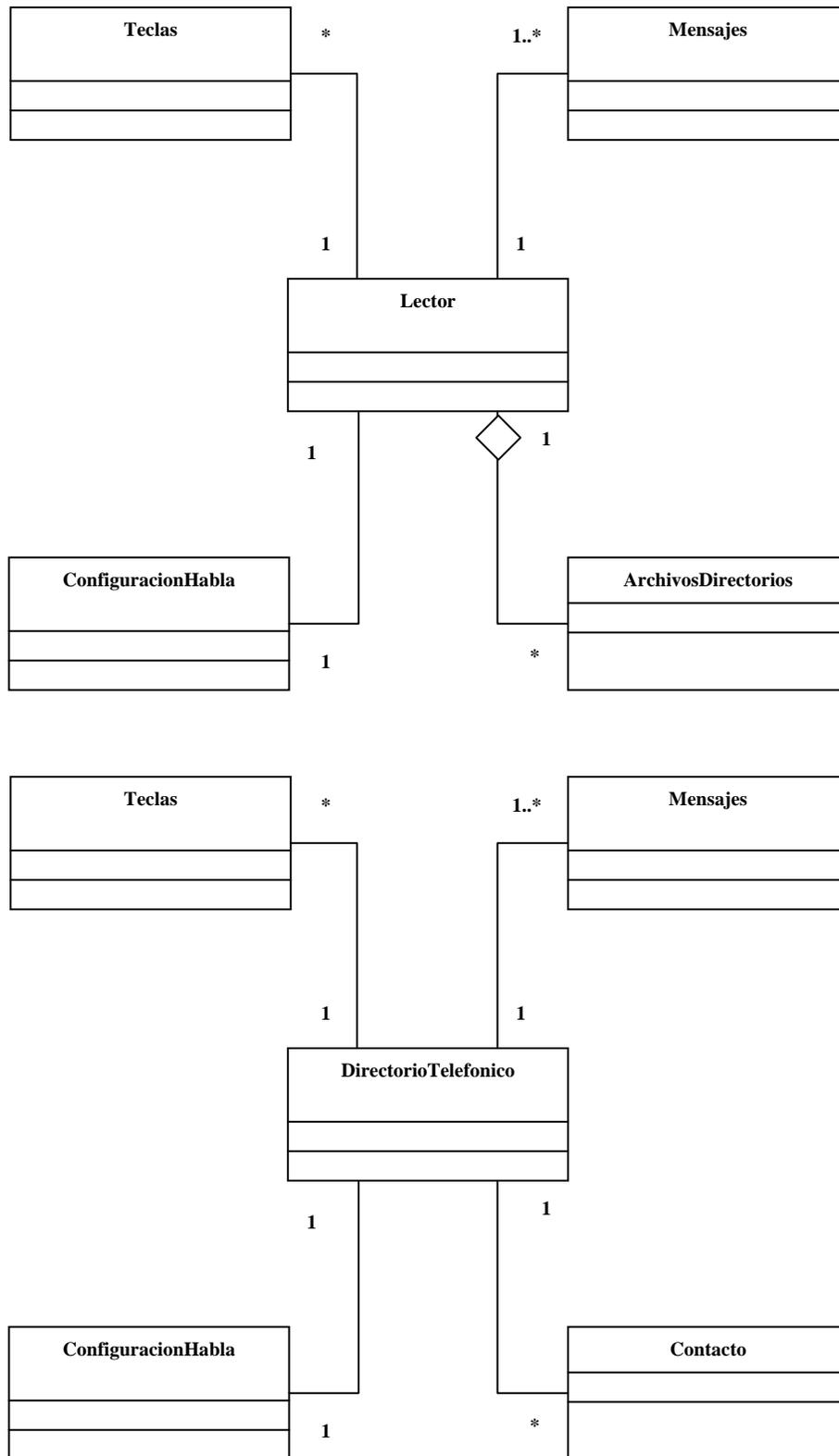


Figura 4.12. Relación entre clases (continuación)

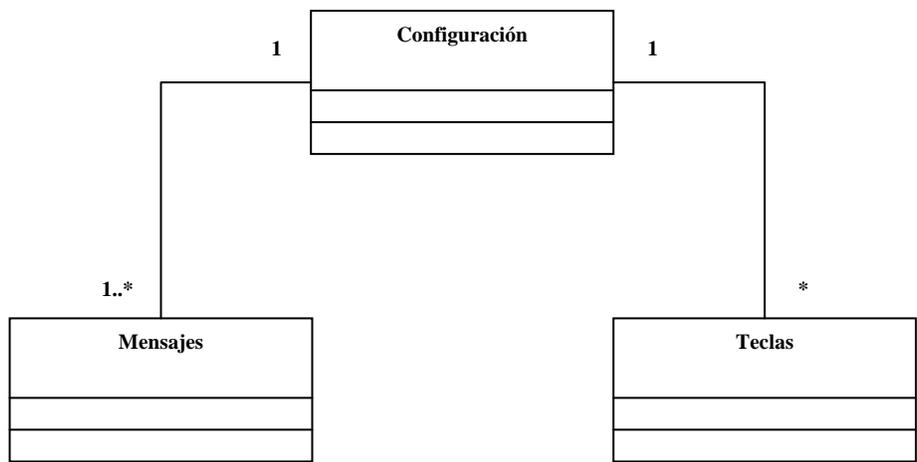
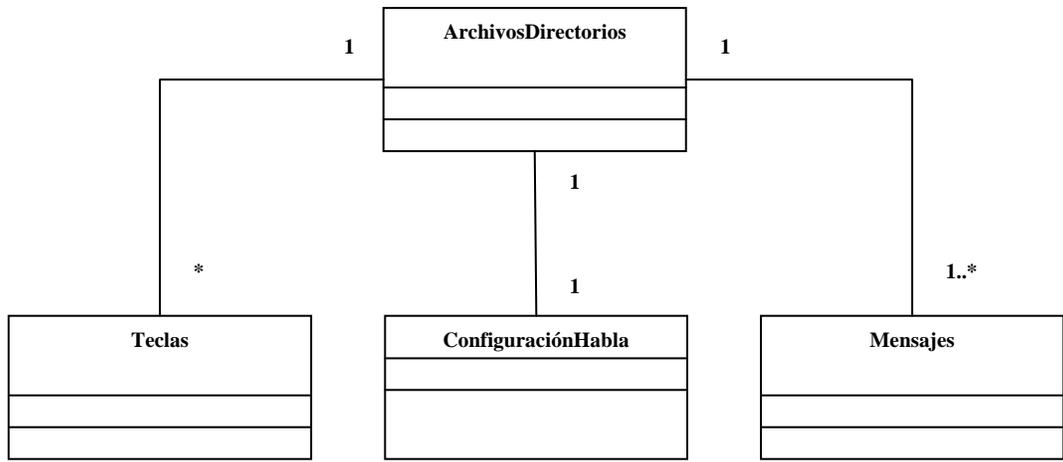


Figura 4.12. Relación entre clases (continuación)

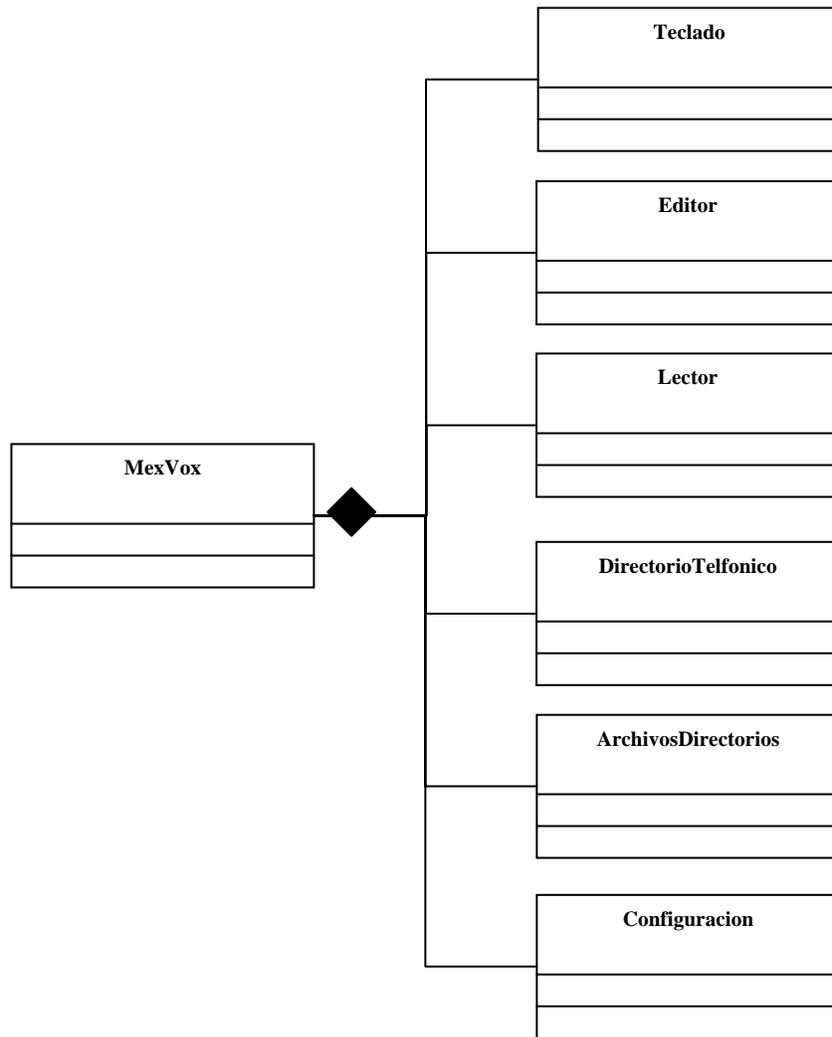


Figura 4.12. Relación entre clases (continuación)

4.6 Colaboración entre Clases

La comunicación entre las clases descritas anteriormente se muestra en la Figura 4.13.

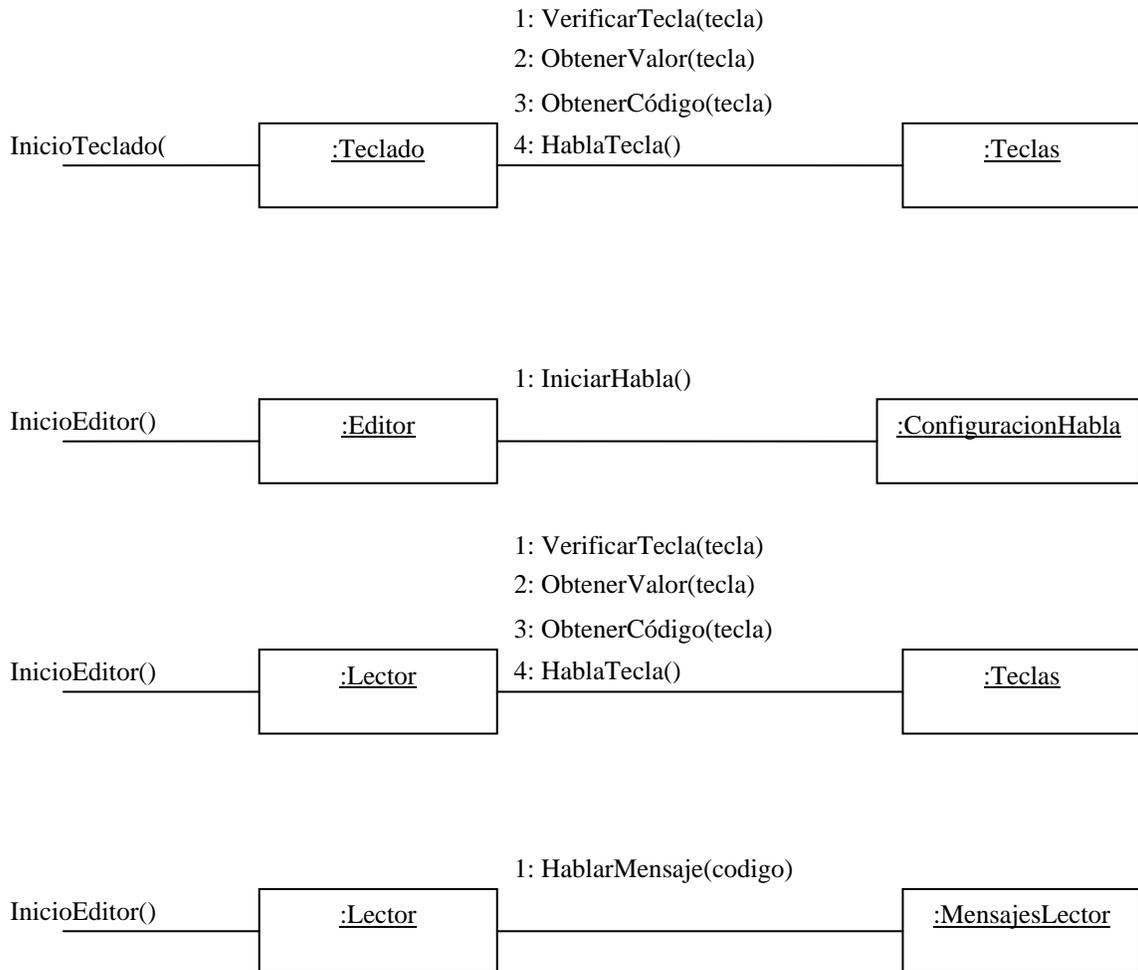


Figura 4.13. Diagramas de Colaboración

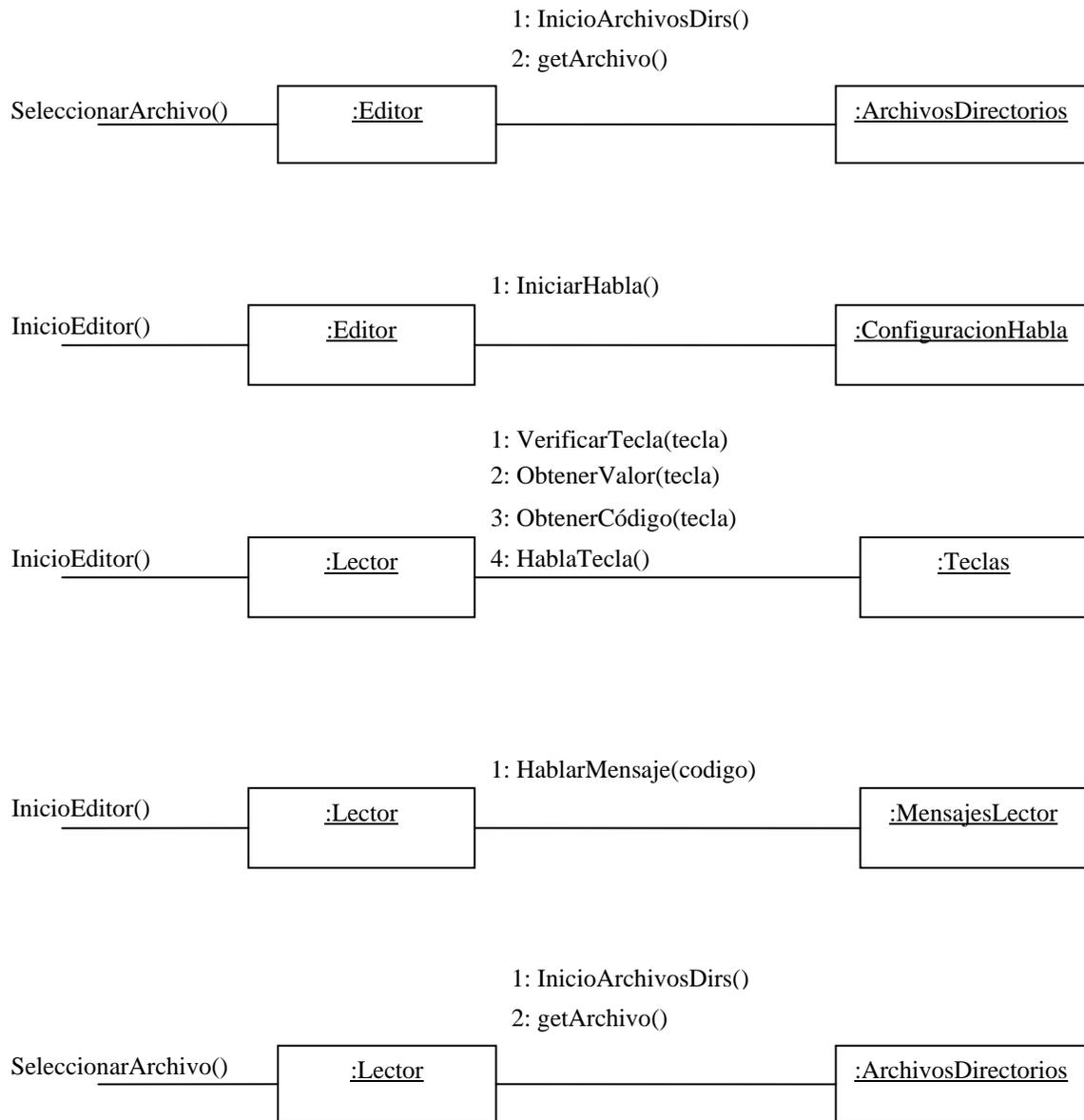


Figura 4.13. Diagramas de Colaboración (continuación)

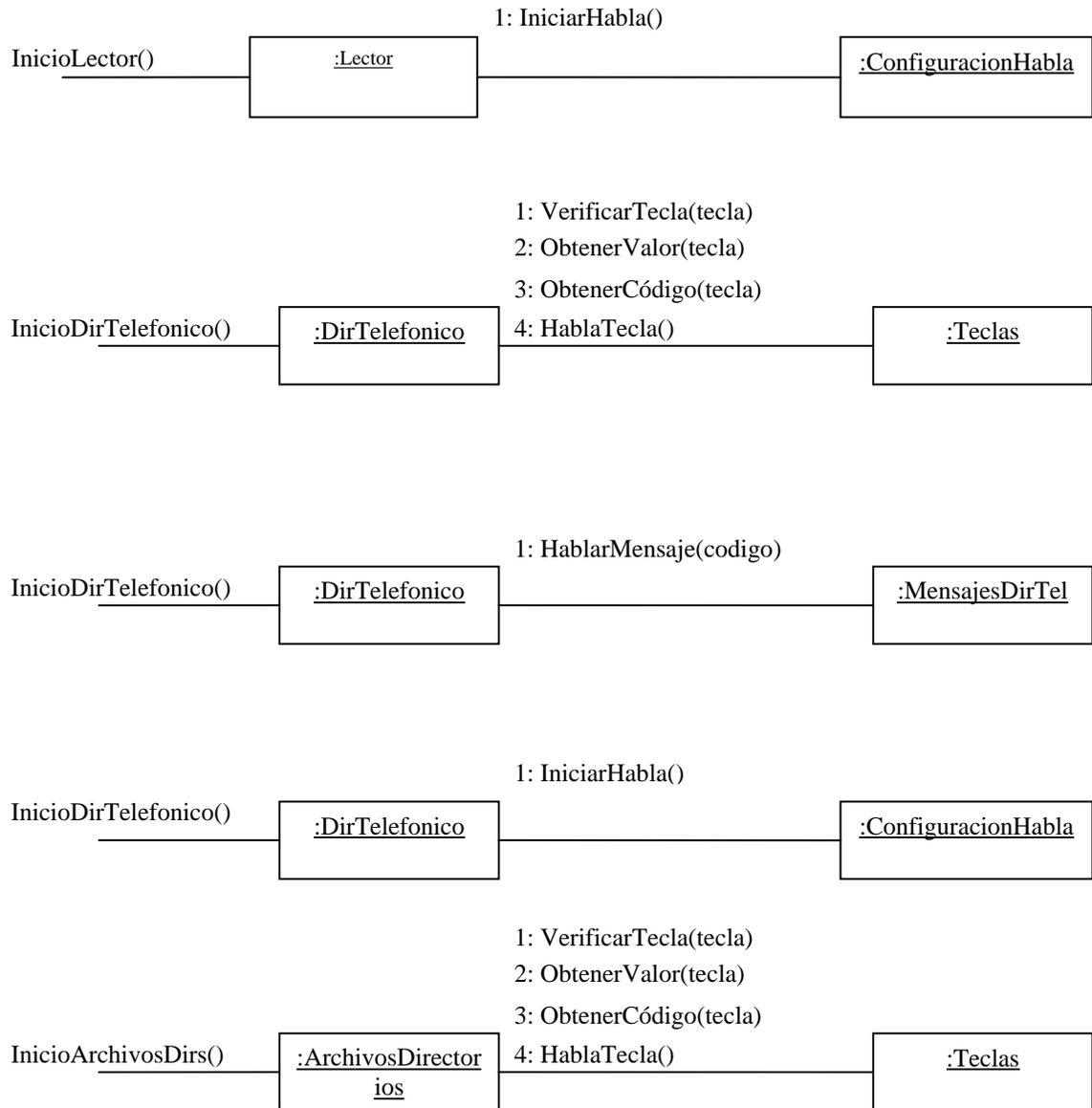


Figura 4.13. Diagramas de Colaboración (continuación)

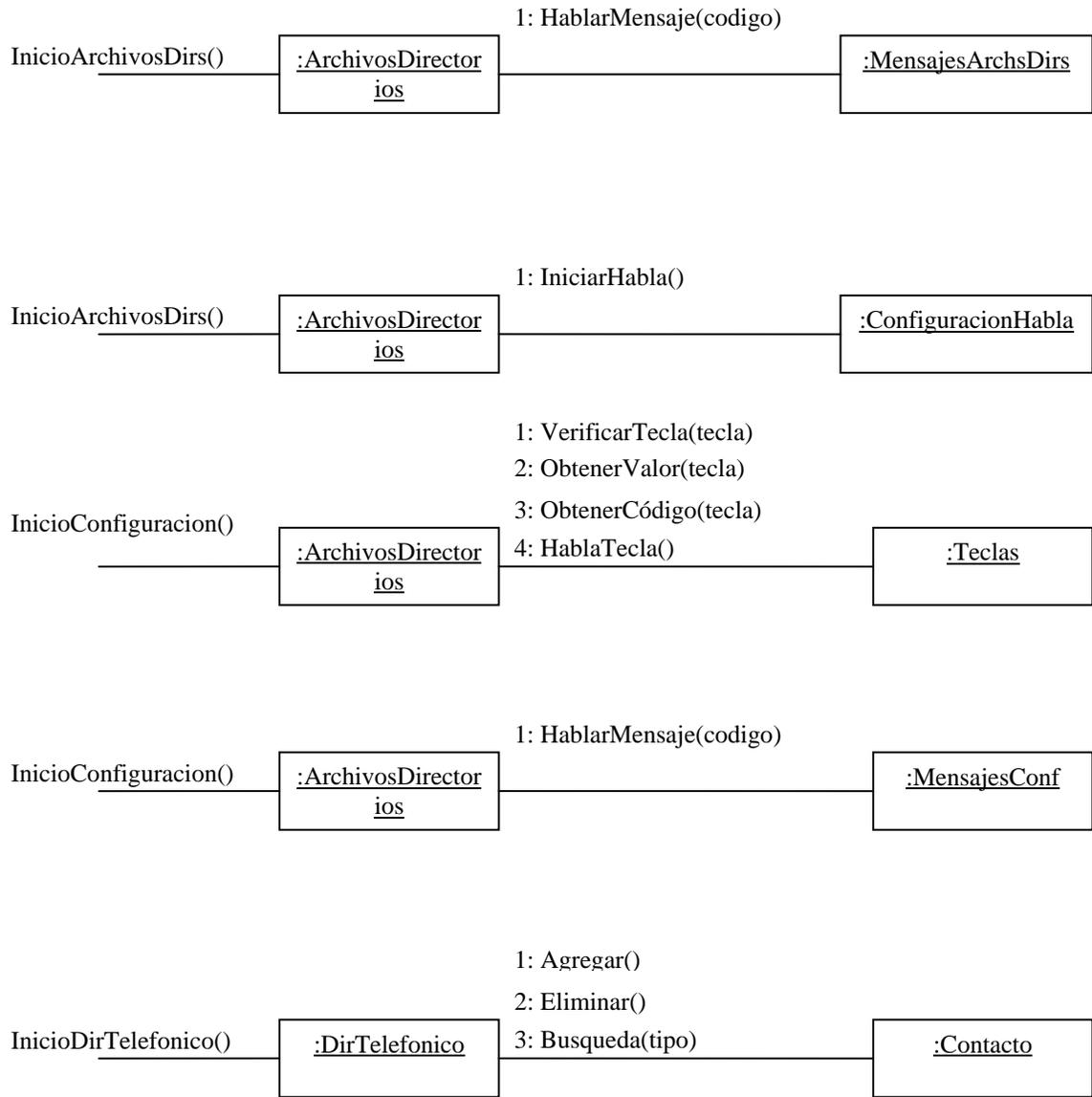


Figura 4.13. Diagramas de Colaboración (continuación)