

INDICE

1.	Introducción.....	5
1.1.	Presentación del proyecto.....	7
1.2.	Objetivos.....	8
1.3.	Contexto de aplicación.....	9
1.3.1.	Introducción.....	9
1.3.2.	Parálisis cerebral.....	9
1.3.2.1.	Síntomas.....	9
1.3.2.2.	Problemas en la comunicación.....	10
1.3.3.	Sistemas alternativos de comunicación.....	10
1.3.3.1.	SPC.....	11
1.3.3.2.	Sistema Bliss.....	11
2.	Análisis.....	15
2.1.	Introducción.....	17
2.2.	Descripción del problema.....	17
2.3.	Identificación de los actores.....	20
2.4.	Diagrama de casos de uso.....	20
2.5.	Presentación de casos de uso.....	21
2.6.	Estructura estática del sistema.....	25
2.6.1.	Identificación de las clases.....	25
2.6.2.	Diagrama inicial de clases.....	26
3.	Diseño.....	29
3.1.	Introducción.....	31
3.2.	Especificación de casos de uso y diagramas de secuencia.....	31
3.3.	Diagrama final de clases del componente.....	44
3.4.	Datos en las configuraciones.....	47
3.5.	Diccionario de datos.....	48
3.6.	Descripción de la interfaz gráfica de usuario del comunicador.....	51
4.	Implementación.....	55
4.1.	Introducción.....	57
4.2.	Ficheros utilizados.....	57
4.3.	Clases de .NET utilizadas para la implementación.....	59
5.	Pruebas.....	61
5.1.	Introducción.....	63
5.2.	Pruebas de comprobación de funcionalidad.....	63
5.3.	Pruebas para casos límite.....	67

6.	Conclusiones.....	71
6.1.	Objetivos alcanzados.....	73
6.2.	Conocimientos adquiridos.....	73
6.3.	Líneas de trabajo futuras.....	74
7.	Apéndices.....	77
	Apéndice A. Tecnología utilizada.....	78
	Apéndice B. Estructura de los ficheros.....	87
	Apéndice C. Glosario de términos.....	101
	Apéndice D. Manual de usuario del comunicador.....	103
	Apéndice E. Hardware y software utilizado.....	127
	Apéndice F. Contenido del CD.....	129
	Bibliografía.....	131





# 1. Introducción



## 1.1 Presentación del proyecto

Este proyecto forma parte de una serie de proyectos propuestos por el Grupo de Investigación en Reutilización y Orientación al Objeto (GIRO) orientados al desarrollo de componentes software reutilizables en el dominio de la Comunicación Alternativa y Aumentativa (CAA).

Estos componentes software se desarrollan con la finalidad de elaborar un conjunto de elementos reutilizables que permitan el desarrollo de aplicaciones futuras a partir de la reutilización de los mismos.

Este proyecto en particular consiste en el desarrollo de un componente software reutilizable que implemente un teclado adaptable que, integrado en una aplicación, sirva para facilitar la comunicación de las personas con parálisis cerebral con su entorno.

La idea central es que el sistema tenga el mayor grado de movilidad y autonomía posible, por eso y por la mayor facilidad de programación se utilizará la plataforma Pocket PC. Sin embargo, será posible integrar el componente en aplicaciones para PC sin realizar para ello un gran esfuerzo.

El componente estará compuesto por distintas pantallas, éstas a su vez formadas por un teclado adaptable de botones (posibilita la interacción del usuario con el sistema) y por una parte dedicada a mostrar la información que el usuario desea transmitir (cuadro de texto).

Con la finalidad de satisfacer las necesidades de un mayor número de usuarios, el componente ofrecerá varias formas de introducir la información a transmitir. A continuación se detallan:

- Introducción de texto carácter a carácter. Esta forma va destinada a usuarios que sepan leer y escribir. La pulsación de un botón provocará la introducción de un único carácter en el cuadro de texto.
- Introducción del texto asociado a imágenes de algún alfabeto conocido por el usuario: Esta forma va destinada a usuarios que no sepan leer, ni escribir; pero que conozcan algún alfabeto basado en imágenes como por ejemplo SPC (Sistema Pictográfico de Comunicación), Bliss, etc. La pulsación de un botón provocará la introducción de una palabra o de una expresión en el cuadro de texto.

El usuario tendrá dos formas de seleccionar el botón deseado de los que formarán el teclado:

- Pulsación directa del botón: Consistirá en pulsar directamente el botón que represente la función que se desea realizar (por ejemplo escribir un carácter).
- Barrido: El teclado irá señalando, a intervalos de tiempo iguales, todos los botones que aparecen en la pantalla que se esté mostrando. La finalidad de este método es que cuando el usuario desee seleccionar el botón que está señalado, deba realizar una acción muy simple (por ejemplo accionar un pulsador que esté conectado a la PDA).

El proyecto también consiste en la utilización del componente para crear un comunicador. Un comunicador es una aplicación que permite al usuario hacer más natural y sencilla la comunicación con su entorno.

### 1.2 Objetivos

Los objetivos principales que se persiguen son los siguientes:

- Desarrollar un componente software reutilizable que implemente un teclado adaptable para permitir la introducción de información por parte de los usuarios.
- Desarrollar, a modo de demostración de uso del componente, una aplicación que utilice el componente para facilitar la comunicación de las personas con parálisis cerebral con su entorno (un comunicador).

Como consecuencia del primero de los objetivos principales aparecen los siguientes objetivos secundarios:

- Familiarizarse con el desarrollo de software basado en componentes.
- Crear el componente con un grado de configurabilidad tal que se pueda modificar la apariencia y funcionalidad del teclado que implementa de una manera sencilla.
- Crear el componente con un grado de extensibilidad tal que se puedan añadir nuevos métodos de escritura (por ejemplo un nuevo alfabeto de imágenes) de una manera sencilla.
- Introducir en el teclado adaptable un sistema de barrido que facilite la selección de los botones a los usuarios.
- Crear la interfaz del teclado lo más fácil de usar y lo más amigable posible de cara a su utilización por personas discapacitadas.
- Introducir en el componente un sistema de escritura basado en la inserción de caracteres.
- Introducir en el componente un sistema de escritura basado en la inserción del texto asociado a imágenes de cualquier alfabeto conocido por el usuario (por ejemplo alfabeto SPC, Bliss, etc.).

Como consecuencia del segundo de los objetivos principales aparecen los siguientes objetivos secundarios:

- Aprender a integrar unos componentes con otros en una aplicación real.
- Añadir un reproductor de voz que exprese, de manera oral, la información escrita por el usuario.
- Permitir que el usuario elija el método de escritura a utilizar desde la aplicación creada (el comunicador).

## 1.3 Contexto de aplicación

### 1.3.1 Introducción

El componente está pensado para su integración en aplicaciones destinadas a ser utilizadas por el colectivo de personas con parálisis cerebral que, debido a su enfermedad, tienen importantes discapacidades motoras y problemas para comunicarse de manera oral. En las siguientes subsecciones se hablará sobre los problemas en la comunicación provocados por la parálisis cerebral y sobre algún método de comunicación aumentativa y alternativa.

### 1.3.2 Parálisis cerebral

La parálisis cerebral es un grupo de trastornos motores que alteran los movimientos voluntarios y la postura, ocasionados por una afección en el cerebro en desarrollo del niño. En muchos de estos niños no puede identificarse la causa de la parálisis cerebral y, cuando puede ser identificada, generalmente responde a algún trastorno padecido previamente al nacimiento.

Entre las causas más frecuentemente asociadas se encuentran los traumatismos maternos durante el embarazo, ictericia neonatal, asfixia durante el parto y cualquier trastorno padecido durante el embarazo. También puede aparecer un cuadro de parálisis como consecuencia de una infección del sistema nervioso central, un traumatismo o algún tipo de asfixia.

#### 1.3.2.1 Síntomas

Los síntomas de la parálisis cerebral varían considerablemente ya que existen muchos tipos y niveles diferentes. El primer síntoma de que un niño tiene parálisis cerebral es un retraso en alcanzar los indicadores importantes en el área motora o del movimiento. A continuación se detallan otros posibles síntomas:

- Convulsiones: contracciones súbitas, violentas e incontrolables de un grupo de músculos, aunque también pueden ser más sutiles involucrando solamente una breve "pérdida de contacto" o unos breves momentos de lo que pareciera ser un "soñar despierto".
- Espasmos musculares: Contracciones involuntarias de los músculos, generalmente dolorosas, que hacen que el músculo se endurezca y se abulte.
- Dificultad para succionar o comer
- Respiración irregular
- Desarrollo retardado de las destrezas motoras, tales como alcanzar objetos, sentarse, gatear, etc.
- Retardo mental
- Anormalidades del habla: Disartria: se manifiesta en la conversación diaria por la dificultad de expresar ciertos sonidos o palabras.
- Anormalidades visuales
- Anormalidades auditivas
- Contracciones espásticas de las articulaciones
- Rango limitado del movimiento
- Dientes en forma de clavija

### 1.3.2.2 Problemas en la comunicación

Los niños con parálisis cerebral a menudo tienen problemas con el lenguaje. Esto es porque la parálisis cerebral afecta a los músculos que se usan para producir el habla (lengua, garganta, pulmones, etc.) esto se conoce como disartria. El habla de estos niños puede ser lenta y confusa. Sus voces pueden tener un sonido nasal si entra demasiado aire por la nariz o pueden ser como que han tenido un resfriado si muy poco aire entra por la nariz.

Debido a estos problemas los niños con parálisis cerebral deben utilizar métodos alternativos para comunicarse. La aplicación desarrollada intenta puede considerarse uno de estos métodos.

### 1.3.3 Sistemas alternativos de comunicación

A continuación se presentan varias definiciones hechas por expertos en sistemas alternativos de comunicación y comunicación aumentativa:

*"Los sistemas alternativos de comunicación son instrumentos de intervención destinados a personas con alteraciones diversas de la comunicación y/o lenguaje, y cuyo objetivo es la enseñanza mediante procedimientos específicos de instrucción de un conjunto estructurado de códigos no vocales necesitados o no de soporte físico, los cuales, mediante esos mismos u otros procedimientos específicos de instrucción, permiten funciones de representación y sirven para llevar a cabo actos de comunicación (funcional, espontánea y generalizable), por sí solos, o en conjunción con códigos vocales, o como apoyo parcial a los mismos, o en conjunción con otros códigos no vocales". (Tamarit, 1988, 4).*

*"Llamamos comunicación alternativa a cualquier forma de comunicación distinta del habla y empleada por una persona en contextos de comunicación cara a cara. El uso de signos manuales y gráficos, el sistema Morse, la escritura, etc., son formas alternativas de comunicación para una persona que carece de la habilidad de hablar (...) Comunicación <aumentativa> subraya el hecho de que la enseñanza de las formas alternativas de comunicación tiene un doble objetivo, a saber: promover y apoyar el habla, y garantizar una forma de comunicación alternativa si la persona no aprende a hablar". (Von, y Martisen, 1993, 24).*

*"En la actualidad se ha ido hacia el concepto más amplio de comunicación aumentativa (CA), que incluye todas aquellas opciones, sistemas o estrategias que se pueden utilizar para facilitar la comunicación de toda persona que tiene dificultades graves para la ejecución del habla". (Torres, 2001, 25).*

### 1.3.3.1 SPC

El SPC (Sistema Pictográfico de Comunicación) es un alfabeto que está formado por unos símbolos icónicos sencillos y esquemáticos que representan imágenes reales. Estos dibujos tienen escrita, en su parte superior, la palabra que representa la imagen. Cada símbolo, entonces, representa una palabra. Está especialmente indicado para niños y personas con graves dificultades en el lenguaje oral (Parálisis cerebral, autismo...).

En España, es el sistema alternativo de comunicación más utilizado. Permite desarrollar las principales funciones de la comunicación: expresar deseos, peticiones, sentimientos, afectos... y para intercambiar experiencias, proporcionando y pidiendo información. Ayudan y facilitan la lectura y la escritura a través de la formación de frases y cuentos.

Los símbolos, para facilitar su discriminación, están divididos en categorías de palabras en función de colores:

- Amarillo: personas
- Verde: verbos
- Naranja: nombres
- Morado o rosa: términos sociales
- Azul: términos descriptivos/adverbios
- Blanco: artículos, conjunciones, preposiciones, colores, alfabeto y números.

Dado a su alta utilización en España, el sistema SPC es el que se ha elegido para el desarrollo de la parte de la aplicación dedicada a la comunicación mediante la utilización de imágenes. No obstante, se pueden añadir otros sistemas a la aplicación sin la necesidad de realizar un gran esfuerzo.

### 1.3.3.2 Sistema Bliss

El método de Charles Bliss, es un sistema de comunicación gráfico. Los símbolos Bliss son de una gran simplicidad y no es necesario saber leer para usarlos. En la pre-lectura, el Bliss puede usarse para identificar objetos sencillos y expresar ideas y sentimientos.

Las personas con discapacidad, con un adiestramiento previo, son capaces de utilizar este método con mucha habilidad, por medio de tableros convencionales, que presentan entre 150 y 400 símbolos, según la necesidad del usuario de éste método. Con la ayuda de un sistema informático, el rendimiento en el uso del Bliss y la comunicación con el exterior se pueden extender.

Se utiliza muy frecuentemente en personas que presentan dificultades de la expresión oral y que sean susceptibles de nuevos aprendizajes.

Se ha aplicado a personas con parálisis cerebral, como así también a personas deficientes mentales, afásicas y sordas.

### ➤ Características del Bliss

- Indicado para personas que aún no están preparadas para la utilización del alfabeto gramatical y necesitan un vocabulario extenso.
- Permite crear nuevos símbolos combinando un vocabulario extenso.
- Es de fácil memorización, dado que incluye símbolos pictóricos.
- Contribuye a mejorar el desarrollo global del niño o del sujeto que lo utilice.
- Como buen sistema aumentativo de comunicación, posibilita la interacción en el medio, con lo cual aumenta la estima personal social del usuario.
- Al igual que en el SPC, las palabras se agrupan en categorías, que se identifican por colores:
  - Nombres: naranja.
  - Personas: amarillo.
  - Verbos: verde.
  - Adjetivos: azul.
  - Sociales: rosa.





## **2. Análisis**



## 2.1 Introducción

En este apartado se detallarán las acciones que deberá llevar a cabo el sistema. Para ello se debe realizar un análisis de requisitos (proceso del estudio de las necesidades de los usuarios para llegar a una definición de la funcionalidad que debe ofrecer el componente, de las restricciones del mismo, etc.).

Una vez terminada esta fase, se debe tener un mayor grado de conocimiento sobre las funcionalidades que debe ofrecer el componente a desarrollar.

## 2.2 Descripción del problema

Se debe desarrollar un componente software que implemente un teclado adaptable, es decir, una aplicación que permite al usuario introducir texto mediante la selección de botones. Estos botones representan el texto a introducir si son seleccionados.

El teclado adaptable deberá estar formado por pantallas con un número variable de botones cada una. Las pantallas dedicadas a la introducción de texto por parte del usuario deberán tener un cuadro de texto que muestre, en todo momento, el texto escrito por el mismo.

Los botones serán los que permitirán la interacción entre el usuario y el teclado. Existirán dos tipos de botones:

- Botón de caracteres: Botones que llevan escrita su funcionalidad asociada (por ejemplo: VOLVER, BORRAR, CONFIGURAR, etc.) o el texto que se añadirá en el cuadro de texto al ser seleccionados (por ejemplo: boton A, boton B, boton 7, etc.).
- Botón de imagen: Botones que representan su funcionalidad asociada o el texto que escribirán, al ser seleccionados, mediante imágenes de un alfabeto conocido por el usuario. Existirán dos tipos de botones de imagen:
  - Fijos: Siempre mostrarán la misma imagen y su selección siempre tendrá el mismo efecto (por ejemplo: Botón de selección de una categoría de imágenes, botón de menú en una configuración de imágenes).
  - Variables: La imagen que muestran y el efecto provocado al seleccionarlos puede cambiar (por ejemplo: la imagen de los botones de inserción de texto en una configuración de imágenes y el texto escrito al seleccionarlos varían a lo largo del tiempo).

Las posibles funcionalidades de estos botones serán las siguientes:

- Ir a otra pantalla: La pulsación de un botón con esta funcionalidad hará que el teclado muestre otra pantalla distinta a la actual.
- Escribir texto: La pulsación de un botón con esta funcionalidad hará que se añada en el cuadro de texto el texto asociado al botón.

- Borrar texto: La pulsación de un botón con esta funcionalidad hará que se elimine una porción del texto escrito en el cuadro de texto (podrá ser un único carácter o una palabra entera).
- Reproducir un texto escrito: La pulsación de un botón con esta funcionalidad hará que el componente pida a la aplicación en la que esté integrado (por ejemplo un comunicador) que reproduzca mediante voz el texto escrito en el cuadro de texto.
- Salir de la aplicación: La pulsación de un botón con esta funcionalidad hará que finalice una sesión de trabajo.

La existencia de varias pantallas en el teclado es debida al reducido tamaño de la pantalla del dispositivo en el que generalmente se utilizará (PDA Pocket PC). Por esta circunstancia es imposible representar en una única pantalla toda la funcionalidad ofrecida por el componente.

El componente funcionará bajo una determinada configuración. La configuración será la encargada de facilitar toda la información necesaria para que se creen todas las pantallas del teclado y sus respectivos botones. La información facilitada por la configuración será la siguiente:

- Tamaño de cada una de las pantallas.
- Ubicación de cada pantalla dentro de la pantalla del dispositivo en el que se ejecute (PDA o PC).
- Aparición o no del cuadro de texto en cada pantalla.
- Tamaño y ubicación de los botones dentro de la pantalla.
- Funcionalidad asociada a cada uno de los botones.
- Texto o imagen que representará cada botón.
- Existencia o no de barrido.
- Tiempo de barrido.
- Colores de apariencia (del fondo de pantalla, de los botones de texto, etc.).

Se debe desarrollar el componente lo más configurable y flexible que sea posible, es decir, que suponga el menor esfuerzo realizar cambios en la funcionalidad y en la apariencia del teclado que implementa, añadir nuevas configuraciones (por ejemplo con algún alfabeto de imágenes distinto) o quitar configuraciones que no sean necesarias para un usuario determinado con la finalidad de que el teclado consuma menos recursos de almacenamiento (por ejemplo, puede ser posible que un usuario que sabe leer y escribir no desee utilizar las configuraciones de imágenes que haya en el sistema). En este proyecto se desarrollarán tres configuraciones:

- Configuración de conjuntos: Las pantallas del teclado estarán formadas por botones de caracteres. Existirá una pantalla inicial en la que se mostrará un conjunto de botones representado, cada uno de ellos, por un conjunto de caracteres (por ejemplo: ABCDEF, GHIJKL, etc.). La selección de uno de estos botones llevará a otra pantalla formada también por botones de caracteres. Cada uno de estos botones representará un carácter del botón seleccionado en la pantalla inicial. La selección de estos últimos botones es la que hace que se añadan caracteres al cuadro de texto.

- Configuración de categorías: Las pantallas del teclado estarán formadas por botones de caracteres. Existirá una pantalla inicial en la que se mostrarán las categorías en las que se hayan agrupado los caracteres (por ejemplo: consonantes, vocales y números). La selección de estos botones tendrá dos posibles efectos:
  - Ir a otra pantalla en la que aparecerá un botón por cada uno de los caracteres del botón seleccionado en la pantalla inicial. La selección de estos últimos botones es la que hace que se añadan caracteres al cuadro de texto.
  - Ir a otra pantalla en la que aparecerán botones representados por varios caracteres. A partir de aquí se procederá de forma similar a la configuración de conjuntos.
- Configuración de imágenes: Las pantallas del teclado estarán formadas por botones de imágenes. Estas imágenes pertenecerán a un alfabeto conocido por el usuario. Para facilitar su localización a la hora de escribir, se deberán agrupar por categorías por ejemplo (personas, cosas, animales, etc.). Existirá una pantalla inicial que permitirá al usuario seleccionar la categoría. Una vez elegida se mostrará una pantalla con un conjunto de botones de imagen. La selección de uno de estos últimos botones hará que se añada en el cuadro de texto el texto asociado a la imagen que representa el botón. Debido a la existencia de más imágenes dentro de una categoría que botones de imagen en una pantalla, será necesario que la imagen que representan los botones de una pantalla vaya cambiando temporalmente, es decir, es necesario que los botones de imagen dedicados a la inserción de texto sean variables.

El usuario tendrá dos formas de seleccionar el botón que desee dentro de los que conformarán el teclado. A continuación se detallan:

- Pulsación directa del botón: Consistirá en pulsar directamente el botón que represente la función que se desea realizar (por ejemplo escribir un carácter). Existirán dos formas de seleccionar un botón directamente en función de la plataforma utilizada:
  - Si el comunicador que utiliza el teclado está integrado en una aplicación para Pocket PC, la forma de hacer que se cumpla la función asociada a un botón es pulsando sobre el botón deseado, por ejemplo, con el lápiz de la PDA.
  - Si el comunicador que utiliza el teclado está integrado en una aplicación para PC, la forma de hacer que se cumpla la función asociada a un botón es pulsando con el ratón sobre el botón deseado.
- Barrido: El teclado irá señalando a intervalos de tiempo iguales todos los botones que aparecen en la pantalla. La forma que tendrá el teclado de señalar qué botón está señalado por el barrido es cambiando su color, de manera que sea el único distinto dentro de la pantalla. Existirán dos formas de seleccionar un botón en función de la plataforma utilizada:
  - Si el comunicador que utiliza el teclado está integrado en una aplicación para Pocket PC, se deberá conectar un pulsador a la PDA de manera que para hacer cumplir la función asociada a un botón deba golpear ese dispositivo cuando dicho botón este señalado.

- Si el comunicador que utiliza el teclado está integrado en una aplicación para PC, la forma de hacer que se cumpla la función asociada a un botón es pulsando cualquier tecla del teclado del PC cuando dicho botón está señalado.

**Nota:** El barrido podrá ser activado y desactivado por el usuario. Aunque el barrido esté activado, el usuario podrá seleccionar los botones de forma directa.

Debe existir la posibilidad de que el usuario, utilizando el propio teclado, elija la configuración deseada, la existencia o no del barrido, el tiempo de barrido y los colores de apariencia. Es decir, una parte del teclado debe estar destinada a su propia configuración.

El componente debe ser integrado, a modo de ejemplo, en una aplicación (comunicador) que permita a las personas con parálisis cerebral comunicarse con su entorno.

Debe ser posible integrar el componente tanto en aplicaciones para la plataforma Pocket PC como en aplicaciones para PC.

Por último, se deberá añadir en el comunicador otro componente (ya desarrollado en otro proyecto) que se encargará de reproducir el texto escrito por el usuario del teclado adaptable.

### 2.3 Identificación de los actores

Un actor representa cualquier elemento que intercambia información con el sistema, por eso está fuera de él. Un actor puede ser una persona, un sistema, una máquina, etc.

En el estudio del problema se han identificado los siguientes actores:

- Discapacitado (ACT-001): Es el principal usuario del sistema. Será la persona que, debido a su discapacidad, utilizará la aplicación para comunicarse con el entorno.
- Ayudante (ACT-002): Será el encargado de configurar el sistema en función del grado de parálisis que tenga el usuario que va a utilizar el teclado. Este actor se encargará también de ayudar al actor Discapacitado (ACT-001) con el manejo del teclado en su fase de aprendizaje.

### 2.4 Diagrama de casos de uso

Una vez estudiada la descripción del problema, se procede a plasmar, de manera escrita, la funcionalidad del sistema. Mediante el diagrama de casos de uso se muestran, de manera gráfica, las funcionalidades que ofrece el sistema para cada actor.

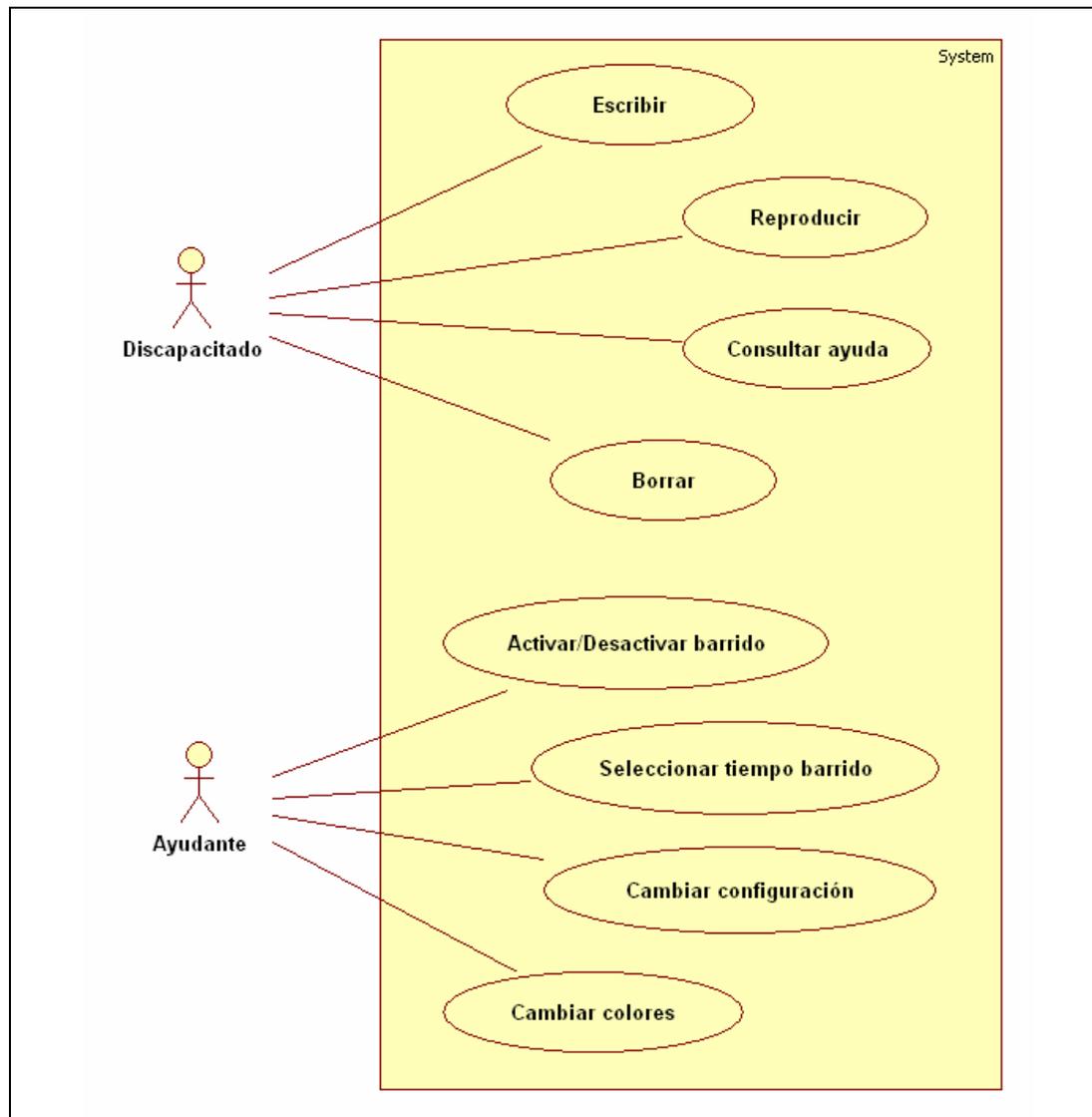


Figura 2.1 Diagrama de casos de uso

## 2.5 Presentación de casos de uso

A continuación se describirán, de forma general, los casos de uso representados en el diagrama anterior. En esta fase no se entrará en detalles relacionados con la parte de diseño, como puede ser, por ejemplo, la interfaz mostrada al usuario para realizar una función. Ese nivel de detalle se mostrará en la especificación de los casos de uso (apartado 3.2) donde se hablará de la apariencia del teclado.

UC-001		Escribir	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se desee escribir texto en el cuadro de texto.		
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El sistema mostrará una pantalla compuesta por los grupos en que están agrupados los botones de escritura (ya sean de caracteres o de imágenes).	
	2	El actor Discapacitado (Act-001) seleccionará el grupo que desee.	
	3 a	Si el sistema muestra una pantalla formada por botones de escritura, ir al paso 4.	
	3 b	Si el sistema muestra una pantalla formada por botones de selección de grupo, ir al paso 2.	
	4	El actor Discapacitado (Act-001) seleccionará el botón asociado al carácter, la palabra o expresión que desea escribir.	
5	El sistema escribe el texto en el cuadro de texto y muestra todo el texto escrito hasta el momento.		

Figura 2.2 Descripción del caso de uso Escribir

UC-002		Reproducir	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se desee reproducir un texto escrito.		
<b>Precondición</b>	Para que el caso de uso tenga algún efecto debe existir texto escrito en el campo de texto.		
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El sistema mostrará el botón dedicado a reproducir el texto que esté escrito en el cuadro de texto.	
	2	El actor Discapacitado (Act-001) seleccionará dicho botón.	
3	El sistema reproduce el texto.		

Figura 2.3 Descripción del caso de uso Reproducir

UC-003		Consultar ayuda	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se desee consultar la ayuda.		
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El sistema muestra los temas de ayuda.	
	2	El actor Discapacitado (Act-001) selecciona el tema sobre el que desea recibir la ayuda.	
	3	El sistema muestra la ayuda relacionada con el tema seleccionado.	

Figura 2.4 Descripción del caso de uso Consultar ayuda

UC-004		Borrar	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se desee borrar un carácter.		
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El actor Discapacitado (Act-001) selecciona borrar.	
	2 a	El sistema borra el último carácter que se haya escrito.	
	2 b	El sistema borra la última palabra escrita en el cuadro de texto.	
<b>Comentarios</b>	El paso por los puntos 2 a o 2 b dependerá de la configuración con la que se esté trabajando.		

Figura 2.5 Descripción del caso de uso Borrar

UC-005		Activar/Desactivar barrido	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se desee activar o desactivar el barrido.		
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El sistema ofrecerá la posibilidad de elegir una u otra opción.	
	2	El actor Ayudante (Act-002) elegirá la opción que desee.	
	3	El sistema actualizará el estado del barrido (activado o desactivado).	

Figura 2.6 Descripción del caso de uso Activar/Desactivar barrido

<b>UC-006</b>		<b>Seleccionar tiempo barrido</b>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se desee seleccionar un tiempo barrido.		
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El sistema mostrará los tiempos de barrido que se pueden elegir.	
	2	El actor Ayudante (Act-002) seleccionará uno de los tiempos disponible.	
	3	El sistema actualizará el tiempo de barrido.	

Figura 2.7 Descripción del caso de uso Seleccionar tiempo barrido

<b>UC-007</b>		<b>Cambiar configuración</b>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se desee cambiar la configuración.		
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El sistema mostrará las configuraciones disponibles.	
	2	El actor Ayudante (Act-002) seleccionará la configuración deseada.	
	3	El sistema actualizará la configuración.	

Figura 2.8 Descripción del caso de uso Cambiar configuración

<b>UC-008</b>		<b>Cambiar colores</b>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se deseen cambiar los colores de apariencia.		
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El sistema mostrará los colores disponibles.	
	2	El actor Ayudante (Act-002) elegirá un color.	
	3	El sistema realizará los cambios de color solicitados.	

Figura 2.9 Descripción del caso de uso Cambiar colores

## 2.6 Estructura estática del sistema

### 2.6.1 Identificación de las clases

Para realizar el modelo de objetos del sistema es imprescindible encontrar las clases que van a formar parte del dominio del problema. Para ello, se suele realizar un estudio de la descripción textual del problema:

- Los sustantivos generalmente se corresponderán con las clases.
- Los verbos transitivos suelen corresponderse con métodos de una clase (guardar, escribir, etc.).
- Los verbos de posesión dan lugar a estructuras de composición (tener, estar formado por, etc.).

Haciendo el estudio de la descripción del problema se han obtenido los siguientes candidatos a clases:

- **ComponenteTeclado:** Clase principal del teclado, será la encargada de crear todas las pantallas con los respectivos botones y con todas sus funcionalidades.
- **Pantalla:** Clase que representará a las pantallas del teclado.
- **Boton:** Clase que representará a los botones.
- **BotonCaracter:** Especificación de la clase Boton, para representar los botones de caracteres.
- **BotonImagen:** Especificación de la clase Boton, para representar los botones de imágenes.
- **Configuracion:** Clase que representará a las distintas configuraciones con la que puede trabajar la clase ComponenteTeclado. En función de la información dada por esta clase se crearán las distintas pantallas y los distintos botones que formarán el teclado.
- **Imagen:** Clase que representa la información de las imágenes que puede mostrar un BotonImagen. Estas imágenes serán las que aparecerán, de manera rotativa en una determinada pantalla.
- **Texto:** Clase que almacena el texto escrito en el cuadro de texto.
- **Barrido:** Clase encargada de controlar el tiempo que pasa señalado cada botón a la espera de una pulsación del usuario.

En la siguiente fase del estudio (fase de diseño) se decidirá si estas clases son suficientes o si es necesario añadir alguna más.

### 2.6.2 Diagrama inicial de clases

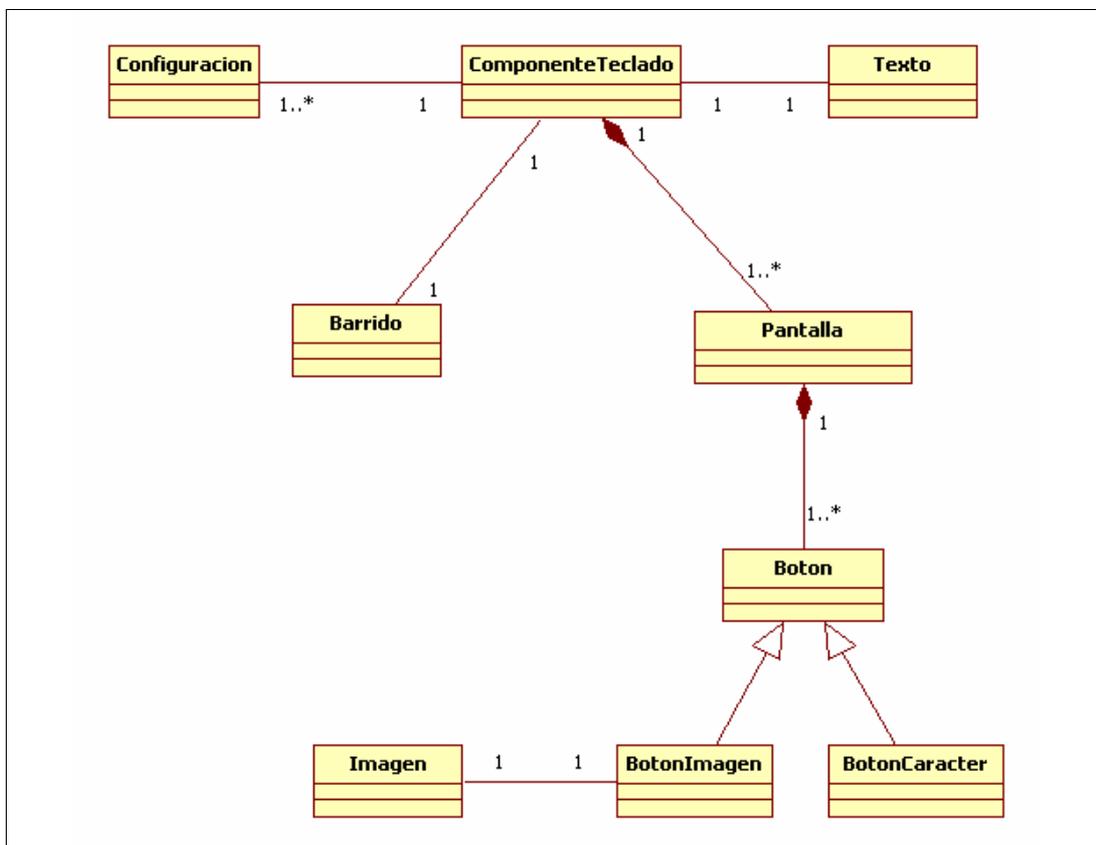


Figura 2.10 Diagrama inicial de clases





## **3. Diseño**



### 3.1 Introducción

El diseño es el proceso por el cual se traducen las especificaciones de los requisitos en una representación del software que se desea construir. El diseño representa un puente entre el análisis del problema y la implementación de la solución a ese problema.

En esta parte, se desarrollan los casos de uso que han sido descritos en apartados anteriores. Además, se representará un diagrama de secuencia por cada caso de uso mostrando el orden de las llamadas del sistema en funcionamiento.

También se mostrarán esquemas que representaran la forma de estructurar la información relativa a las distintas configuraciones que se incluirán en el componente.

Además de lo anterior, se mostrará el diagrama de clases definitivo junto con el diccionario de datos, donde se explicarán los términos que aparecerán en el diagrama de clases y un esquema de de la apariencia que tendrá la interfaz de la aplicación.

### 3.2 Especificación de casos de uso y diagramas de secuencia

#### Caso de uso Escribir

UC-001	Escribir	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se desee añadir un texto al cuadro de texto.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema mostrará una pantalla compuesta por un campo de texto y un teclado formado por una combinación de botones (de imagen o de caracteres) representando, cada uno de ellos, a otro grupo de botones.
	2	El actor Discapacitado (Act-001) seleccionará el grupo que contenga el texto que desea escribir.
	3 a	Si el sistema muestra una pantalla formada por el cuadro de texto y botones de escritura, ir al paso 4.
	3 b	Si el sistema muestra una pantalla formada por el cuadro de texto y botones de selección de grupo, ir al paso 2.
	4	El actor Discapacitado (Act-001) seleccionará el botón que represente el texto que desea escribir.
	5	El sistema añade el carácter, la palabra o la expresión asociada al botón a lo que ya hubiera escrito en el campo de texto.
<b>Importancia</b>	Vital	
<b>Comentarios</b>	El número de pantallas por las que se debe pasar hasta encontrar el botón de escritura dependerá de la configuración con que se esté trabajando. Este caso de uso es el que asegura la funcionalidad más básica del componente, que es permitir la comunicación del actor Discapacitado (Act-001) con su entorno mediante la escritura.	

Figura 3.1 Especificación del caso de uso Escribir

### Diagrama de secuencia Escribir

El diagrama de secuencia que se encarga de la realización del caso de uso Escribir se ha dividido en dos diagramas para mayor claridad. En el primero de ellos se ve las acciones que realiza el usuario hasta que se muestre la pantalla que contiene el carácter, texto o expresión que desea escribir y en el segundo de ellos se muestra la acción que realiza para escribirlo (para añadirlo al cuadro de texto).

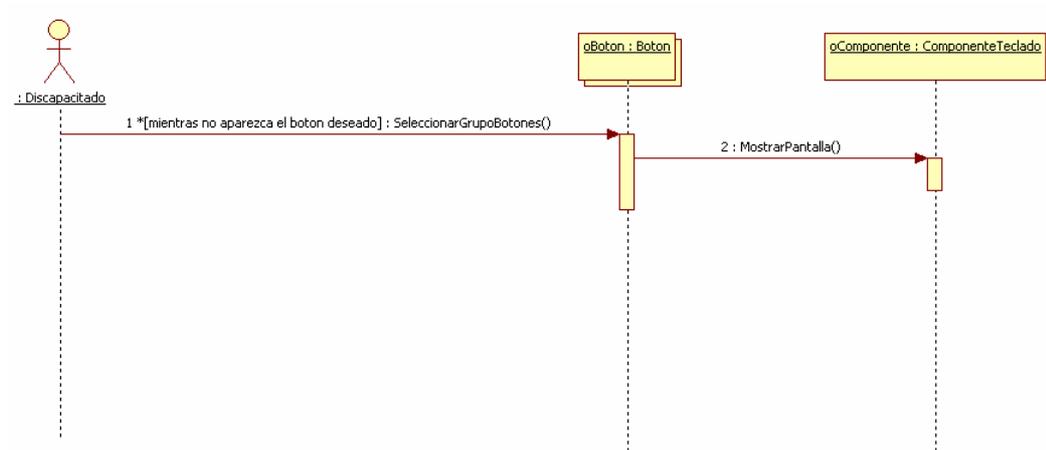


Diagrama de secuencia Seleccionar grupo botones

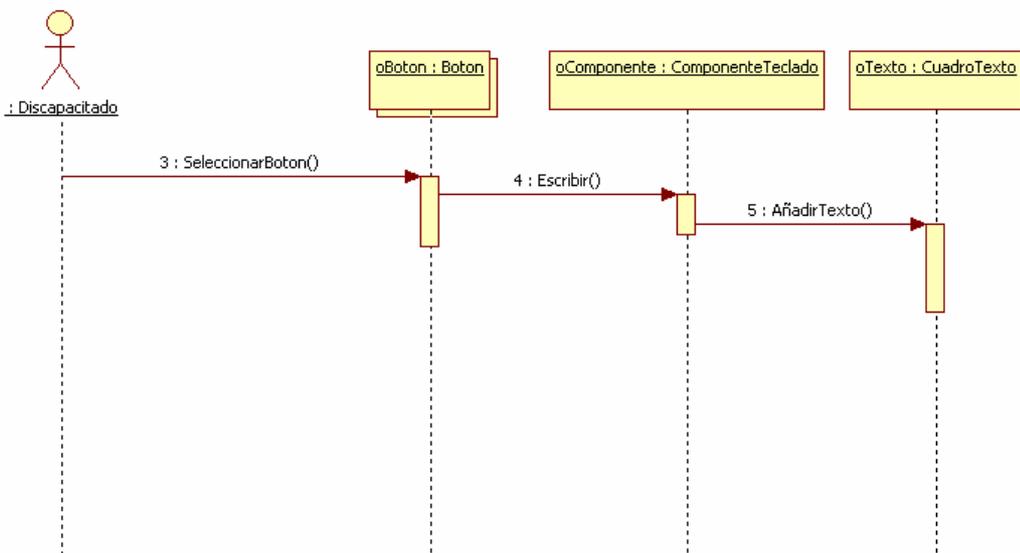


Diagrama de secuencia Seleccionar botón

Figura 3.2 Diagrama de secuencia Escribir

## Caso de uso Reproducir

UC-002	Reproducir	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se desee reproducir un texto escrito en el campo de texto.	
<b>Precondición</b>	El campo de texto debe contener texto para que el caso de uso tenga algún efecto.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema muestra la pantalla inicial compuesta por un campo de texto y un teclado de botones (botones de caracteres o de imágenes)
	2	El actor Discapacitado (Act-001) selecciona el botón "MENU".
	3	El sistema muestra una nueva pantalla (pantalla de Menú) compuesta por un conjunto de botones.
	4	El actor Discapacitado (Act-001) selecciona el botón con el texto "PLAY".
	5 a	El sistema reproduce el texto que haya escrito en el campo de texto y sigue mostrando la pantalla de menú.
5 b	El sistema reproduce el texto que haya escrito en el cuadro de texto, vuelve a la pantalla anterior y borra el contenido del cuadro de texto.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	5a',5b'	El sistema muestra un mensaje de error si el contenido del campo de texto no se puede reproducir en el idioma para el se desarrolló el componente de reproducción.
<b>Importancia</b>	Vital	
<b>Comentarios</b>	El caso de uso hace que el actor Discapacitado (Act-001) se comunique de forma oral con su entorno. El paso por los puntos 5 a o 5 b dependerá de la configuración con la que se esté trabajando.	

Figura 3.3 Especificación del caso de uso Reproducir

**Diagrama de secuencia Reproducir**

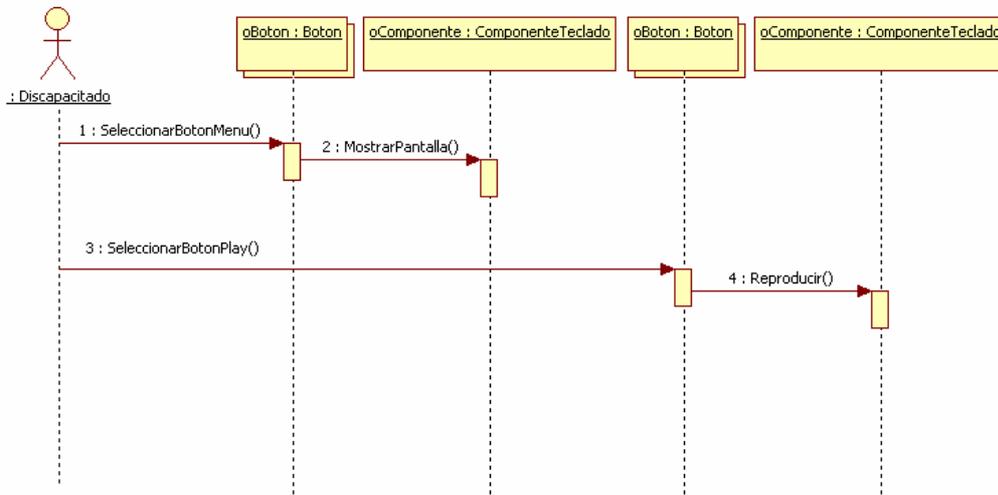


Figura 3.4 Diagrama de secuencia Reproducir

**Caso de uso Consultar ayuda**

UC-003	Consultar ayuda	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se desee consultar la ayuda.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor Discapacitado (Act-001) accederá al menú de la aplicación.
	2	El sistema mostrará las opciones del menú.
	3	El actor Discapacitado (Act-001) accederá a la opción de ayuda.
	4	El sistema mostrará los temas sobre los que puede ofrecer ayuda.
	5	El actor Discapacitado (Act-001) seleccionará el tema que desee.
	6	El sistema ofrecerá la ayuda correspondiente al tema seleccionado
<b>Importancia</b>	Muy importante	
<b>Comentarios</b>	Mediante este caso de uso es posible aprender a utilizar la aplicación de manera rápida.	

Figura 3.5 Especificación del caso de uso Consultar Ayuda

**Diagrama de secuencia Consultar ayuda**

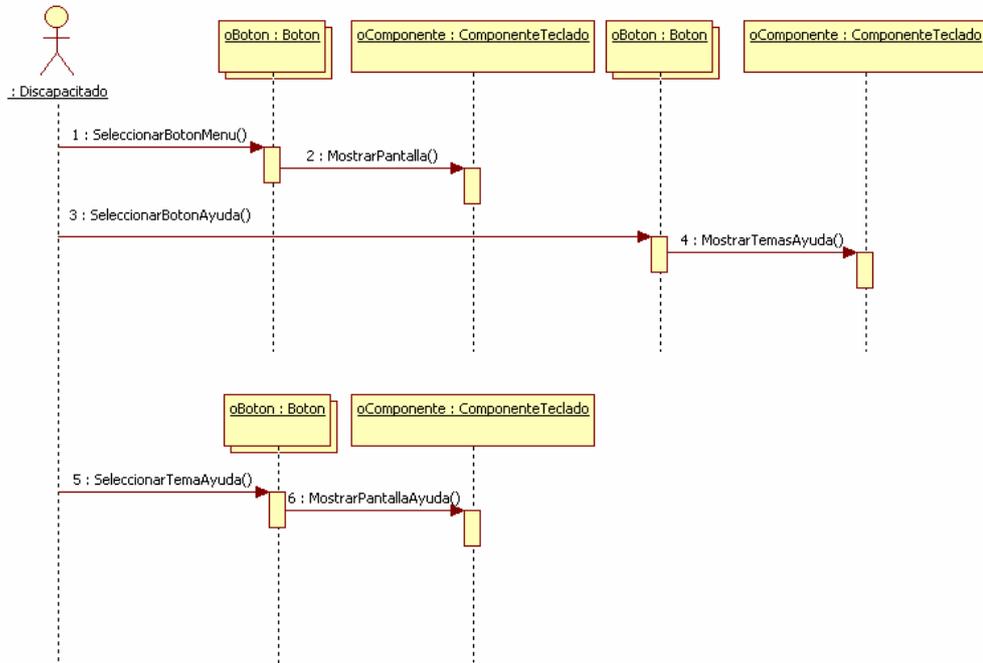


Figura 3.6 Diagrama de secuencia Consultar ayuda

Caso de uso Borrar

<b>UC-004</b>	<b>Borrar</b>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se desee borrar una porción de texto.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor Discapacitado (Act-001) selecciona el botón de borrado.
	2 a	El sistema borra el último carácter que se haya escrito en el campo de texto.
	2 b	El sistema borra la última palabra escrita en el campo de texto.
<b>Importancia</b>	Vital	
<b>Comentarios</b>	Este caso de uso es de vital importancia porque es el que permite al usuario corregir posibles fallos en la escritura. El botón de borrado aparecerá en todas las pantallas salvo en las pantallas que pertenecen al MENU. El paso por los puntos 2 a o 2 b dependerá de la configuración con la que se esté trabajando.	

Figura 3.7 Especificación del caso de uso Borrar

Diagrama de secuencia Borrar

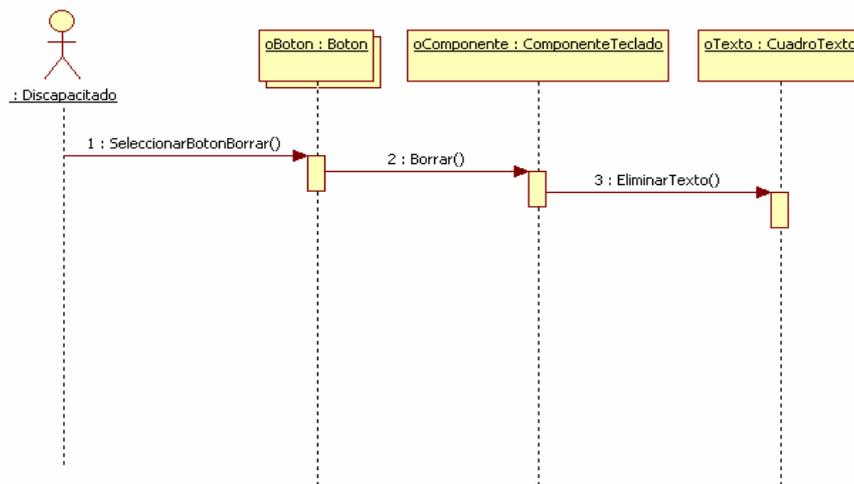


Figura 3.8 Diagrama de secuencia Borrar

**Caso de uso Activar/Desactivar barrido**

UC-005	Activar/Desactivar barrido	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se desee activar o desactivar el barrido.	
<b>Precondición</b>	La aplicación se debe estar ejecutando con alguna configuración de escritura basada en inserción de caracteres, ya que en las configuraciones de imágenes no está permitido trabajar sin barrido porque éste es necesario para el paso automático de pantallas.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor Ayudante (Act-002) accede al menú.
	2	El sistema muestra las opciones del menú.
	3	El actor Ayudante (Act-002) selecciona CONFIGURACION.
	4	El sistema muestra las opciones de configuración, entre ellas la opción de decidir si se quiere tener el barrido activado o no.
	5	El actor Ayudante (Act-002) selecciona la opción que desee y pulsa "ACEPTAR".
	6	El sistema muestra un mensaje de aviso para reiniciarse y actualizar la configuración.
	7	El actor Ayudante (Act-002) acepta el mensaje.
8	El sistema se reinicia con la situación del barrido actualizada.	
<b>Importancia</b>	Vital	
<b>Comentarios</b>	El caso de uso es de importancia vital porque permite al actor Ayudante (Act-002) decidir, en función del grado de parálisis del usuario final del sistema, si necesita de la utilización del barrido para utilizar la aplicación. Aunque el barrido no esté activado, es posible configurar el tiempo del mismo.	

Figura 3.9 Especificación del caso de uso Activar/Desactivar barrido

**Diagrama de secuencia Activar/Desactivar barrido**

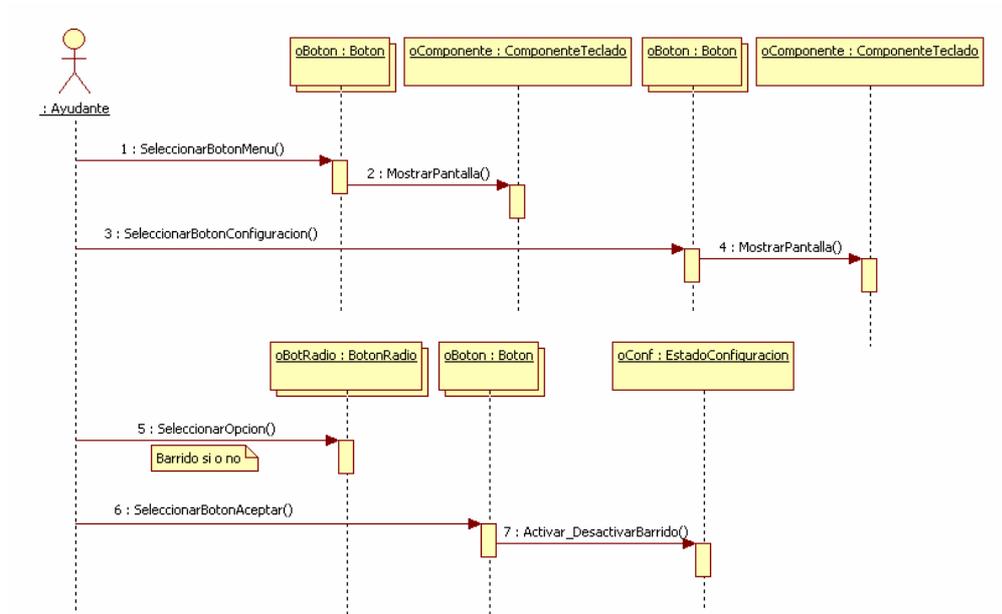


Figura 3.10 Diagrama de secuencia Activar/Desactivar barrido

## Caso de uso Seleccionar tiempo barrido

UC-006	Seleccionar tiempo barrido	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se desee seleccionar un tiempo barrido.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor Ayudante (Act-002) accede al menú.
	2	El sistema muestra las opciones del menú.
	3	El actor Ayudante (Act-002) selecciona CONFIGURACION.
	4	El sistema muestra las opciones de configuración, entre ellas una lista de tiempos de barrido.
	5	El actor Ayudante (Act-002) selecciona el tiempo que desee y pulsa "ACEPTAR".
	6	El sistema muestra un mensaje de aviso para reiniciarse y actualizar la configuración.
	7	El actor Ayudante (Act-002) acepta el mensaje.
8	El sistema se reinicia con el tiempo de barrido actualizado.	
<b>Importancia</b>	Vital	
<b>Comentarios</b>	El caso de uso es de vital importancia porque permite al actor Ayudante (Act-002) elegir el tiempo que cada botón del teclado permanece señalado a la espera de una pulsación. Así este tiempo podrá ser aumentado o disminuido en función del grado de parálisis del actor Discapacitado (ACT-001) que vaya a utilizar la aplicación.	

Figura 3.11 Especificación del caso de uso Seleccionar tiempo barrido

**Diagrama de secuencia Seleccionar tiempo barrido**

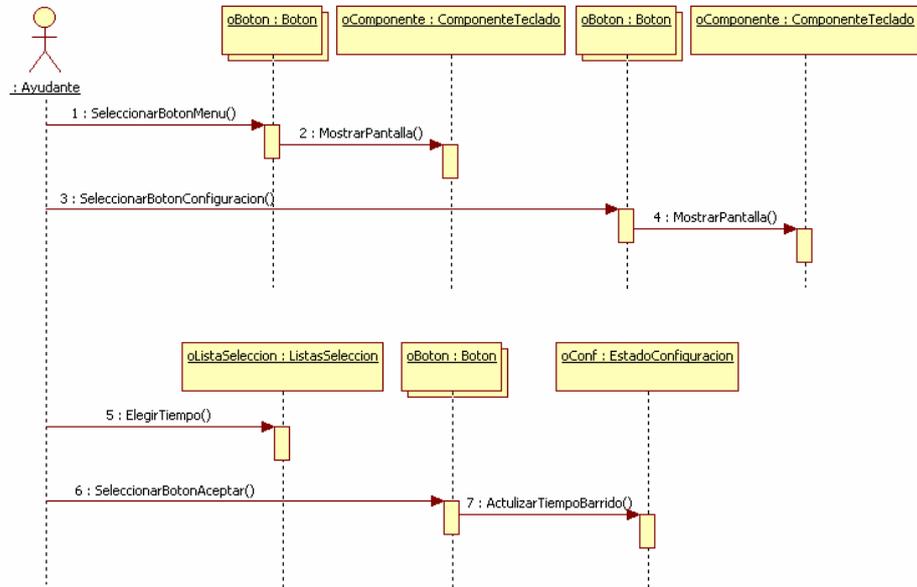


Figura 3.12 Diagrama de secuencia Seleccionar tiempo barrido

## Caso de uso Cambiar configuración

UC-007	Cambiar configuración	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se desee seleccionar otra apariencia y funcionalidad del teclado, dentro de las posibles.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor Ayudante (Act-002) accede al menú.
	2	El sistema muestra las opciones del menú.
	3	El actor Ayudante (Act-002) selecciona "CONFIGURACION".
	4	El sistema muestra las opciones de configuración, entre ellas una lista con las configuraciones disponibles.
	5	El actor Ayudante (Act-002) selecciona la configuración que desee y pulsa "ACEPTAR"
	6	El sistema muestra un mensaje de aviso para reiniciarse y actualizar la configuración.
	7	El actor Ayudante (Act-002) acepta el mensaje.
	8	El sistema se reinicia con la nueva configuración cargada.
<b>Importancia</b>	Vital	
<b>Comentarios</b>	El caso de uso es de importancia vital porque permite al actor Ayudante (Act-002) seleccionar la configuración más adecuada para el actor Discapacitado (Act-001) que vaya a utilizar la aplicación en función de sus conocimientos (por ejemplo si sabe leer o no)	

Figura 3.13 Especificación del caso de uso Cambiar configuración

### Diagrama de secuencia Cambiar configuración

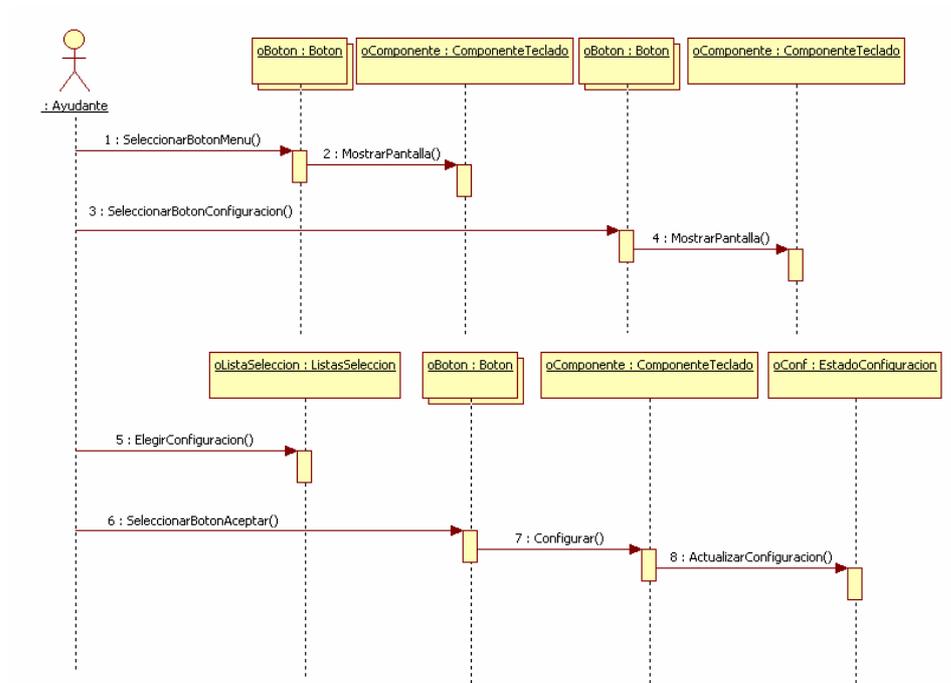


Figura 3.14 Diagrama de secuencia Cambiar configuración

Caso de uso Cambiar colores

UC-008	Cambiar colores	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en este caso de uso cuando se deseen cambiar los colores de apariencia.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor Ayudante (Act-002) accede al menú.
	2	El sistema muestra las opciones del menú.
	3	El actor Ayudante (Act-002) selecciona "COLORES".
	4	El sistema muestra los elementos de las pantallas a los que se les puede modificar el color.
	5	El actor Ayudante (Act-002) selecciona el elemento al que desee cambiar el color.
	6	El sistema muestra un conjunto de botones coloreados cada uno de ellos con un color.
	7	El actor Ayudante (Act-002) selecciona el color que desee.
8	El sistema actualiza los cambios de color.	
<b>Importancia</b>	Poca	

Figura 3.15 Especificación del caso de uso Cambiar colores

Diagrama de secuencia Cambiar colores

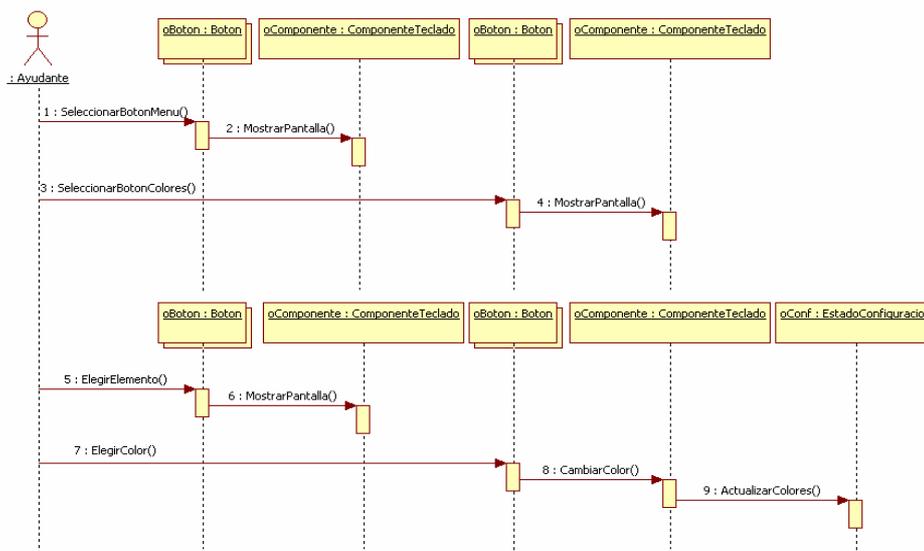


Figura 3.16 Diagrama de secuencia Cambiar colores

### 3.3 Diagrama final de clases del componente

Como puede verse en el diagrama (figura 3.17), se han añadido algunas clases al diagrama presentado en la parte de análisis (figura 2.10), como son la clase BotonRadio, la clase ListasSeleccion y la clase Etiquetas. Además de esto, se ha añadido una nueva asociación de tipo composición entre la clase imagen y la clase paneles. Esta nueva asociación surge de la necesidad de tener botones de imagen variables en las pantallas formadas por botones de escritura de las configuraciones de imágenes. Así, estas pantallas contendrán un conjunto de imágenes que se irán asociando a sus botones de imagen variables a lo largo del tiempo.

Un cambio importante es el que se ha realizado con la clase Configuracion. Ésta ha sido sustituida por la clase EstadoConfiguracion. Esta nueva clase se ha creado con la finalidad de que no sea necesario crear una instancia de la clase Configuración por cada configuración existente en el componente (ya que puede existir un gran número de configuraciones), sino que se creará únicamente una instancia de la clase EstadoConfiguracion que contendrá la información necesaria para que el componente conozca cuál es la configuración con la que debe trabajar.

A continuación se muestra una breve explicación de cada una de las clases:

- Clase ComponenteTeclado: Esta clase será la clase principal del componente. Estará compuesta por todas las pantallas del teclado. Será la encargada de crear todas las pantallas (con sus respectivos botones, cuadro de texto, etc.) y la que añade toda la funcionalidad al componente. Para ello, se basará en la configuración determinada por la clase EstadoConfiguracion.
- Clase EstadoConfiguracion: Esta clase almacenará toda la información necesaria para que la clase ComponenteTeclado conozca cuál es la configuración que debe utilizar para crear el teclado.
- Clase CuadroTexto: Clase encargada de almacenar y mostrar el texto escrito por el usuario.
- Clase Barrido: Clase encargada de controlar el tiempo que cada botón, de una determinada pantalla, está señalado a la espera de una pulsación del usuario.
- Clase Pantalla: Esta clase será la que almacenará la información de cada pantalla del teclado. Estará compuesta por objetos de las clases Boton, BotonRadio, ListasSeleccion, Etiquetas e Imágenes.
- Clase Boton: Almacenará la información (posición, tamaño, apariencia y funcionalidad) de los botones que formarán el teclado.
- Clase BotonImagen: Almacenará la información relativa a los botones de imagen que estarán presente en las pantallas del teclado cuando se esté trabajando con alguna configuración de imágenes.
- Clase BotonCaracter: Almacenará la información relativa a los botones de caracteres que aparecerán en las pantallas del teclado.

- Clase BotonRadio: Almacenará la información relativa a los botones de selección que estarán presente en la pantalla de configuración para dar la posibilidad al usuario de elegir si quiere trabajar con barrido o no.
- Clase ListasSeleccion: Almacenará la información relativa a las listas de selección que estarán presente en la pantalla de configuración. Se utilizarán para dar la posibilidad al usuario de elegir el tiempo de barrido y para elegir la configuración con la que desee trabajar.
- Clase Etiquetas: Almacenará información (posición, tamaño, texto y colores de apariencia) relativa el texto informativo que aparecerá en las distintas pantallas. El texto informativo está dedicado a guiar las acciones del usuario (por ejemplo: “Tiempo de barrido” aparecerá justo encima de la lista de selección que permitirá al usuario elegir un tiempo de barrido, etc.).
- Clase Imagenes: Almacenará información (imagen normal, imagen a mostrar al ser señalada por el barrido y texto a escribir por el botón de escritura que la tenga como imagen asociada) sobre las imágenes que aparecerán en las pantallas cuando se esté trabajando con alguna configuración de imágenes.

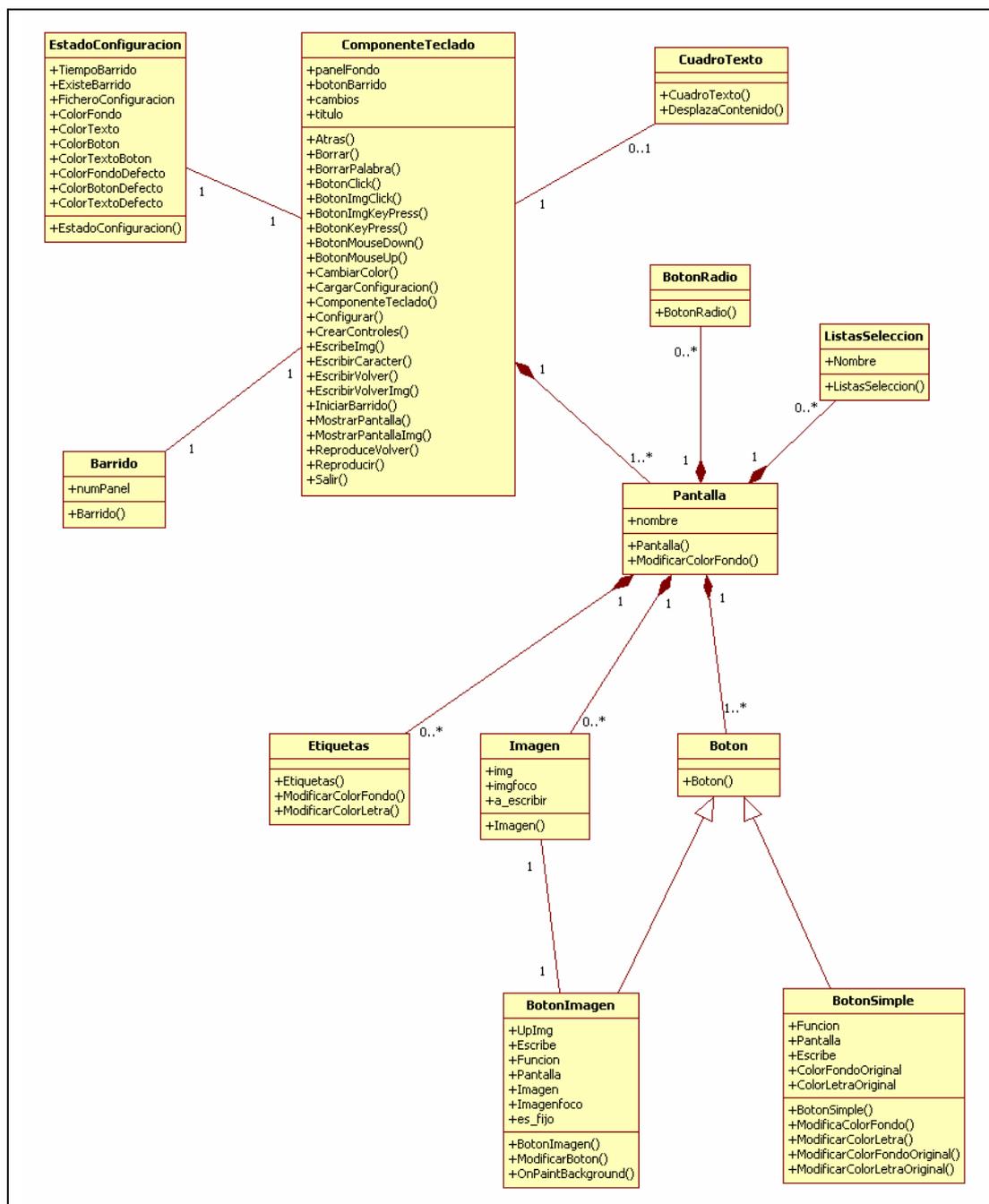


Figura 3.17 Diagrama final de clases

### 3.4 Datos de las configuraciones

Para representar la información con la que trabajará cada una de las configuraciones que tenga el componente se utilizará un diagrama entidad relación (E-R). Los motivos de esta elección son los siguientes:

- Familiarización: Después de cursar asignaturas como Bases de Datos, Ingeniería del Software I o Ingeniería del Software II se considera este tipo de diagrama como básico para el modelado de datos de los sistemas software.
- Facilidad de comprensión: El diagrama muestra de una manera muy eficaz los datos con los que trabajará el componente y las relaciones entre ellos. Incluso una persona no familiarizada con la informática podría entender fácilmente lo que se quiere decir con él.

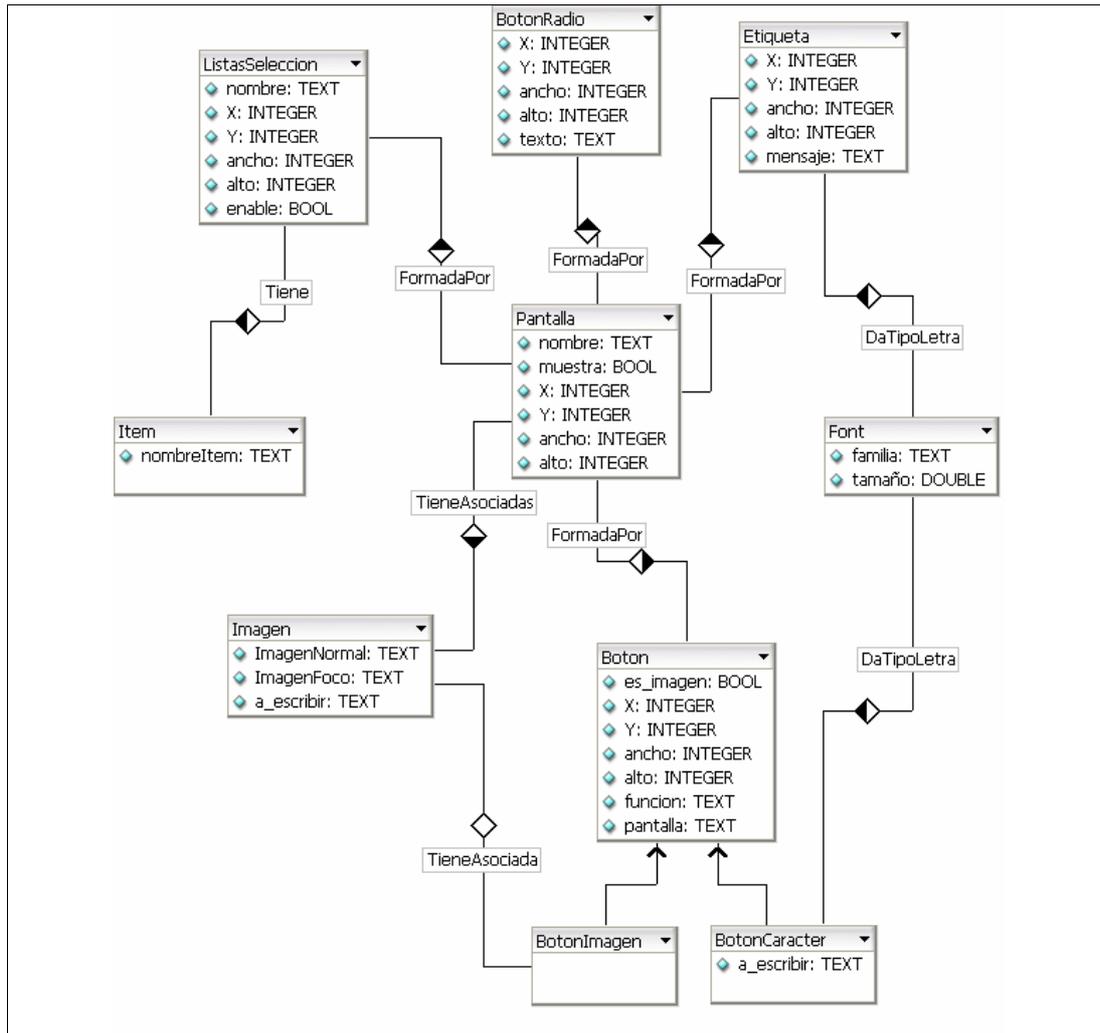
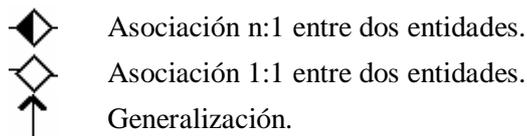


Figura 3.18 Entidad Relación de las configuraciones

Leyenda:



### 3.5 Diccionario de datos

En este apartado se explicará el significado de todos los términos que aparecen en el anterior diagrama entidad relación (figura 3.18).

<b>Entidad Pantalla</b>		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>
nombre	Contiene el nombre de la pantalla.	Texto
muestra	Contiene un valor que indica si la pantalla muestra o no el cuadro de texto.	Booleano
X	Contiene la coordenada x del borde superior izquierdo de la pantalla.	Número
Y	Contiene la coordenada y del borde superior izquierdo de la pantalla.	Número
ancho	Contiene el ancho de la pantalla.	Número
alto	Contiene el alto de la pantalla.	Número

Figura 3.19 Datos de la entidad Pantalla

<b>Entidad Boton</b>		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>
es_imagen	Contiene un valor que indica si el botón es de tipo imagen o es de tipo caracteres	Booleano
X	Contiene la coordenada x del borde superior izquierdo del botón.	Número
Y	Contiene la coordenada y del borde superior izquierdo del botón.	Número
ancho	Contiene el ancho del botón.	Número
alto	Contiene el alto del botón.	Número
funcion	Contiene la funcionalidad asociada a la pulsación del botón.	Texto
pantalla	Si el botón tiene la funcionalidad de ir a otra pantalla al ser pulsado, pantalla contiene el nombre de dicha pantalla.	Texto

Figura 3.20 Datos de la entidad Boton

<b>Entidad BotonCaracter</b>		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>
a_escribir	Si el botón tiene la funcionalidad de escribir al ser pulsado, a_escribir contiene el texto a añadir en el cuadro de texto.	Texto

Figura 3.21 Datos de la entidad BotonCaracter

<b>Entidad Imagen</b>		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>
ImagenNormal	Contiene el nombre de la imagen que debe mostrar el botón que la tenga como imagen asociada, cuando el botón no está siendo señalado por el barrido.	Texto
ImagenFoco	Contiene el nombre de la imagen que debe mostrar el botón que la tenga como imagen asociada, cuando el botón está siendo señalado por el barrido.	Texto
a_escribir	Si el botón tiene la funcionalidad de escribir al ser pulsado, a_escribir contiene el texto a añadir en el cuadro de texto.	Texto

Figura 3.22 Datos de la entidad Imagen

<b>Entidad Font</b>		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>
familia	Contiene el nombre de la familia de tipos de letra.	Texto
tamaño	Contiene un valor que indica el tamaño de la letra.	Número

Figura 3.23 Datos de la entidad Font

<b>Entidad Etiqueta</b>		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>
X	Contiene la coordenada x del borde superior izquierdo de la etiqueta.	Número
Y	Contiene la coordenada y del borde superior izquierdo de la etiqueta.	Número
ancho	Contiene el ancho de la etiqueta.	Número
alto	Contiene el alto de la etiqueta.	Número
mensaje	Contiene el texto que debe mostrar la etiqueta.	Texto

Figura 3.24 Datos de la entidad Etiqueta

<b>Entidad BotonRadio</b>		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>
X	Contiene la coordenada x del borde superior izquierdo del botón de selección.	Número
Y	Contiene la coordenada y del borde superior izquierdo del botón de selección.	Número
ancho	Contiene el ancho del botón de selección.	Número
alto	Contiene el alto del botón de selección.	Número
mensaje	Contiene el texto que debe mostrar el botón de selección.	Texto

Figura 3.25 Datos de la entidad BotonRadio

<b>Entidad ListasSeleccion</b>		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>
nombre	Contiene el nombre de la lista de selección.	Texto
X	Contiene la coordenada x del borde superior izquierdo de la lista de selección.	Número
Y	Contiene la coordenada y del borde superior izquierdo de la lista de selección.	Número
ancho	Contiene el ancho de la lista de selección.	Número
alto	Contiene el alto de la lista de selección.	Número
enable	Contiene un valor que indica si la lista de selección está activada o no.	Booleano

Figura 3.26 Datos de la entidad ListasSeleccion

<b>Entidad Item</b>		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>
nombreltem	Contiene el nombre de un ítem de una lista de selección.	Texto

Figura 3.27 Datos de la entidad Item

### 3.6 Descripción de la interfaz gráfica de usuario del comunicador

En este apartado se mostrará, de manera esquemática, la interfaz gráfica de usuario que se desea tenga la aplicación a desarrollar (el comunicador).

Como puede verse en las siguientes figuras, el número de botones de las pantallas es variable. La configuración con la que se esté trabajando indicará el número de botones de cada pantalla, así como su tamaño, su posición, su funcionalidad y su apariencia (como ya se ha explicado en apartados anteriores).

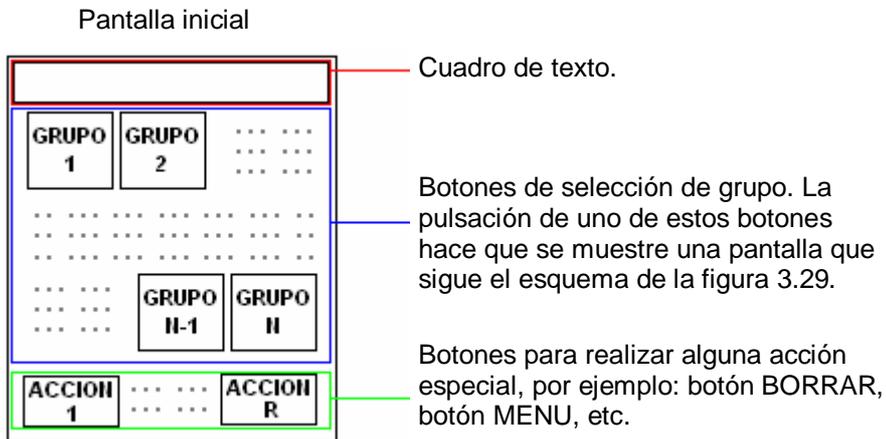


Figura 3.28 Esquema interfaz gráfica de usuario de la pantalla inicial

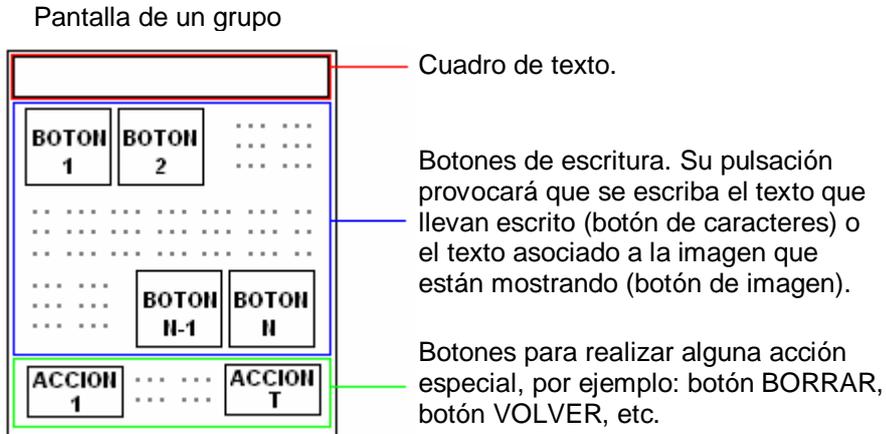


Figura 3.29 Esquema interfaz gráfica de usuario de una pantalla de un grupo

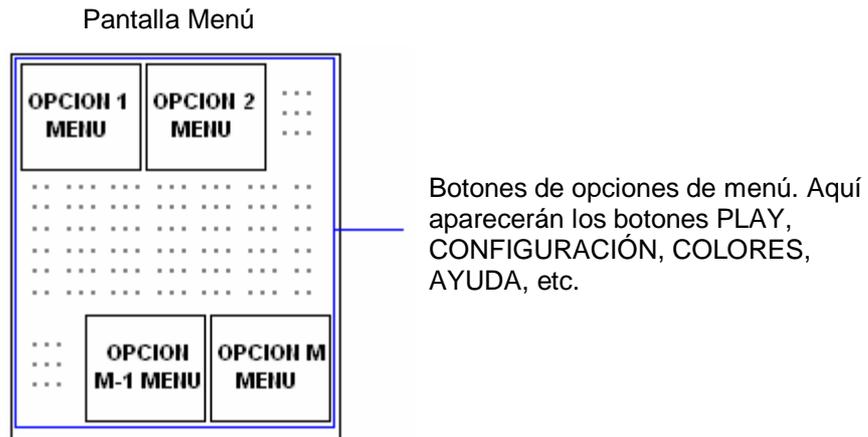


Figura 3.30 Esquema interfaz gráfica de usuario de la pantalla de menú





## 4. Implementación



## 4.1 Introducción

En este apartado se convertirán todas las especificaciones descritas hasta ahora en un sistema software real. Para conseguirlo se deben tomar las siguientes decisiones:

- Tecnología a utilizar.
- Entorno de desarrollo.
- Lenguaje de programación.
- Mecanismo de almacenamiento de información a utilizar para almacenar los datos de las configuraciones.
- Forma de acceder a la fuente de información para el manejo de los datos.

Debido al actual auge de la tecnología .NET se ha decidido desarrollar este proyecto ayudándose de esta tecnología. El entorno de desarrollo utilizado será Visual Studio.NET 2003, que es el entorno de desarrollo que ofrece .NET y se utilizará el lenguaje de programación C# que es el lenguaje nativo de .NET (para más información sobre la tecnología utilizada ver apéndice A).

La decisión que se ha tomado para el almacenamiento de las distintas configuraciones es crear un fichero XML por cada una de ellas. Además de los ficheros de las distintas configuraciones, existirá otro fichero XML que proporcionará al componente la información necesaria para crear todas las pantallas del teclado, como por ejemplo el nombre de la configuración a cargar, si se va a trabajar con barrido o no, el tiempo del barrido, etc.

La elección de XML como fuente de almacenamiento de las configuraciones se ha tomado por distintas razones:

- Visual Studio.NET ofrece clases optimizadas para la lectura y escritura de ficheros XML. Estas clases serán las que permitirán manejar los datos de las distintas configuraciones.
- Visual Studio.NET proporciona un editor para la creación de los documentos XML.
- Facilidad de crear una estructura jerárquica (mediante etiquetas) que contenga los datos relativos a las pantallas que forman una configuración.
- Propiedades que ofrece XML, como por ejemplo la extensibilidad, la posibilidad de compartir datos entre aplicaciones escritas en distintos lenguajes, etc. (ver apéndice A).

## 4.2 Ficheros utilizados

Para ayudar a la labor de implementación de esta aplicación se ha utilizado una serie de ficheros XML, como se decía anteriormente. El contenido de estos ficheros determinará tanto la apariencia del teclado, como la funcionalidad del mismo.

El motivo de su utilización es hacer la aplicación lo más configurable posible (uno de los objetivos propuestos). Así, en estos ficheros se han almacenado los datos necesarios para la creación de los elementos que formarán la interfaz de la aplicación:

- Pantallas.
- Botones.
- Campos de texto.
- Listas desplegables.
- Etiquetas de texto.

De esta manera, para realizar algún cambio en la interfaz, no es necesario modificar ningún archivo de código fuente, basta con modificar el fichero que contenga la definición del elemento que se quiere modificar (por ejemplo, posición de un botón, tamaño de un botón, etc.).

Además de la apariencia de la interfaz, en los ficheros está definida la funcionalidad de los botones que formarán el teclado. De la misma forma que antes, para cambiar la funcionalidad de un botón basta con hacer un cambio mínimo en el fichero correspondiente.

Como ya se ha dicho en apartados anteriores, se han introducido, a modo de ejemplo, tres configuraciones en el componente.

- Configuración de conjuntos (fichero conjuntos.xml)
- Configuración de categorías (fichero categorias.xml)
- Configuración de imágenes (fichero imagenes.xml)

Además de los tres ficheros de configuraciones, ha sido necesario añadir otro fichero XML (configuracion.xml) que será el encargado de:

- Definir que fichero, de los anteriores, debe leer el componente para crear el teclado.
- Decir al componente si el barrido estará activado o no.
- Definir el tiempo de barrido.
- Almacenar los posibles tiempos de barrido para que se pueda dar la posibilidad, desde la aplicación, de elegir uno de ellos cuando se desee cambiar el tiempo de barrido.
- Almacenar un listado con las posibles configuraciones con las que puede trabajar el componente para que se pueda dar la posibilidad, desde la aplicación, de elegir una de ellas cuando se desee modificar la configuración.
- Definir aspectos globales de apariencia para todo el teclado (colores de fondo, de botones, etc.).

Este fichero, será el que asegure la persistencia de la configuración del componente. Es decir, cada vez que se apague la aplicación, se escribirán en este fichero los datos necesarios para que la próxima vez que se inicie la aplicación, se trabaje con la misma configuración que lo hacía en la última sesión de trabajo.

Para ver de una forma detallada las estructuras de los ficheros consúltese al apéndice B.

### 4.3 Clases de .NET utilizadas para la implementación

En este apartado se hablará de las clases de .NET que se han utilizado para implementar las clases que forman el diagrama de clases del componente (figura 3.17).

➤ Clase `System.Windows.Form.Panel`: Clase de la que heredará la clase `Pantalla`. La clase `Panel` es un control que contiene otros controles, de ahí su uso para la implementación de las pantallas porque como se viene comentando a lo largo de este desarrollo “*el teclado estará formado por pantallas que contendrán botones, etc.*”.

➤ Clase `System.Windows.Form.Button`: Clase que se utilizará para la implementación de los botones. Esta elección se debe a que la clase `Button` proporciona eventos que se ajustan muy bien a los requisitos de este sistema como son el evento clic, el evento `KeyPress`, etc.

➤ Clase `System.Windows.Form.Timer`: Clase que se utilizará para la implementación de la clase `Barrido`. La causa de esta elección es que la clase `Timer` implementa un temporizador que provoca un evento en los intervalos que se hayan definido. Esta situación se ajusta perfectamente a la filosofía del barrido como se decía en la parte de análisis “*Se irán señalando a intervalos de tiempo iguales todos los botones que aparecen en la pantalla*”.

➤ Clase `System.Windows.Form.RadioButton`: Clase que se utilizará para la implementación de la clase `BotonRadio`. La elección de esta clase se debe a que permite al usuario seleccionar una única opción entre un grupo de elecciones, que es justamente lo que se quiere conseguir para ofrecer al usuario la posibilidad de activar o desactivar el barrido.

➤ Clase `System.Windows.Form.Label`: Clase que se utilizará para la implementación de la clase `Etiquetas`. Se ha elegido esta clase porque proporciona un texto descriptivo que es lo que se busca con la clase `Etiquetas`, mostrar un texto que guíe al usuario en sus acciones.

➤ Clase `System.Windows.Form.ListBox`: Clase que se utilizará para la implementación de la clase `ListasSeleccion`. La clase `ListBox` permite mostrar una lista de elementos para que el usuario los seleccione haciendo clic en ellos. Este es el motivo de su elección.

➤ Clase `System.Windows.Form.TextBox`: Clase que se utilizará para la implementación de la clase `CuadroTexto`. La clase `TextBox` muestra el texto escrito por el usuario, esta es la funcionalidad más básica que se persigue con el desarrollo del teclado como se decía en la parte de análisis “*la selección de un botón de escritura provocará que el texto asociado al mismo aparezca escrito en un cuadro de texto*”.



## 5. Pruebas



## 5.1 Introducción

En este apartado se presentarán las pruebas realizadas en el la aplicación final (comunicador). La finalidad de las pruebas es detectar los posibles errores para su corrección.

Para la realización de las pruebas se ha utilizado el emulador de Pocket PC que ofrece el entorno de desarrollo de .NET.

Se han realizado dos tipos de prueba:

- De comprobación de funcionalidad: Pruebas encaminadas a corregir los errores que hacen que no se cumpla la funcionalidad más básica pedida por los usuarios (en este caso escribir, borrar y reproducir).
- De casos límite: Pruebas encaminadas a encontrar errores que el desarrollador sospecha que pueden existir en un determinado caso.

## 5.2 Pruebas de comprobación de funcionalidad

<b>PRUEBA 001</b>	<b><i>Escritura usando botón con funcionalidad de ESCRIBIR (a)</i></b>
Acción realizada	Selección de botón de escritura de manera directa estando el barrido desactivado.
Resultado esperado	El texto asociado al botón se añadirá al cuadro de texto.
Resultado obtenido	Texto añadido
Observaciones	Correcto

Figura 5.1 Prueba de escritura (a)

<b>PRUEBA 002</b>	<b><i>Escritura usando botón con funcionalidad de ESCRIBIR (b)</i></b>
Acción realizada	Selección de botón de escritura de manera directa estando el barrido activado.
Resultado esperado	El texto asociado al botón se añadirá al cuadro de texto y el barrido seguirá su camino.
Resultado obtenido	Texto añadido y el barrido sigue su camino.
Observaciones	Correcto

Figura 5.2 Prueba de escritura (b)

<b>PRUEBA 003</b>	<b><i>Escritura usando botón con funcionalidad de ESCRIBIR (c)</i></b>
Acción realizada	Selección del botón de escritura señalado por el barrido.
Resultado esperado	El texto asociado al botón se añadirá al cuadro de texto y el barrido seguirá su camino.
Resultado obtenido	Texto añadido y el barrido sigue su camino.
Observaciones	Correcto

Figura 5.3 Prueba de escritura (c)

<b>PRUEBA 004</b>	<b><i>Escritura usando botón de caracteres con funcionalidad de ESCRIBIR Y VOLVER (a)</i></b>
Acción realizada	Selección de botón de escritura de manera directa estando el barrido desactivado.
Resultado esperado	El texto asociado al botón se añadirá al cuadro de texto y se mostrará la página anterior a la actual.
Resultado obtenido	Texto añadido y mostrado de la anterior página.
Observaciones	Correcto

Figura 5.4 Prueba escribir y volver (a)

<b>PRUEBA 005</b>	<b><i>Escritura usando botón de caracteres con funcionalidad de ESCRIBIR Y VOLVER (b)</i></b>
Acción realizada	Selección de botón de escritura de manera directa estando el barrido activado.
Resultado esperado	El texto asociado al botón se añadirá al cuadro de texto y se mostrará la página anterior a la actual.
Resultado obtenido	Texto añadido y mostrado de la anterior página.
Observaciones	Correcto

Figura 5.5 Prueba escribir y volver (b)

<b>PRUEBA 006</b>	<b><i>Escritura usando botón de caracteres con funcionalidad de ESCRIBIR Y VOLVER (c)</i></b>
Acción realizada	Selección del botón de escritura señalado por el barrido.
Resultado esperado	El texto asociado al botón se añadirá al cuadro de texto y se mostrará la página anterior a la actual.
Resultado obtenido	Texto añadido y mostrado de la anterior página.
Observaciones	Correcto

Figura 5.6 Prueba escribir y volver (c)

<b>PRUEBA 007</b>	<b><i>Borrado con funcionalidad de BORRAR</i></b>
Acción realizada	Selección del botón de la pantalla destinado al borrado.
Resultado esperado	El último carácter del texto que aparece en el cuadro de texto se borrará.
Resultado obtenido	Carácter borrado.
Observaciones	Correcto.

Figura 5.7 Prueba borrar carácter

<b>PRUEBA 008</b>	<b><i>Borrado con funcionalidad de BORRAR PALABRA</i></b>
Acción realizada	Selección del botón de la pantalla destinado al borrado.
Resultado esperado	La última palabra del texto que aparece en el cuadro de texto se borrará.
Resultado obtenido	Palabra borrada.
Observaciones	Correcto.

Figura 5.8 Prueba borrar palabra

<b>PRUEBA 009</b>	<b><i>Mostrado de la pantalla de MENU</i></b>
Acción realizada	Selección del botón de la pantalla destinado a mostrar la pantalla de menú.
Resultado esperado	Se dejará de mostrar la pantalla actual y se mostrará la pantalla de menú.
Resultado obtenido	Pantalla de menú mostrada.
Observaciones	Correcto.

Figura 5.9 Prueba mostrar menú

<b>PRUEBA 010</b>	<b><i>Salir de la aplicación</i></b>
Acción realizada	Selección del botón destinado a provocar el cierre de la aplicación.
Resultado esperado	La aplicación se cerrará.
Resultado obtenido	Aplicación cerrada.
Observaciones	Correcto.

Figura 5.10 Prueba salir de la aplicación

<b>PRUEBA 011</b>	<b><i>Volver a la pantalla anterior</i></b>
Acción realizada	Selección del botón destinado a mostrar la pantalla anterior.
Resultado esperado	Dejará de mostrarse la pantalla actual y se mostrará la pantalla anterior.
Resultado obtenido	Pantalla anterior mostrada.
Observaciones	Correcto.

Figura 5.11 Prueba volver a la pantalla anterior

<b>PRUEBA 012</b>	<b><i>Ver temas de ayuda</i></b>
Acción realizada	Selección del botón destinado a mostrar los temas de ayuda existentes.
Resultado esperado	Mostrado de los temas de ayuda.
Resultado obtenido	Se muestran los temas de ayuda.
Observaciones	Correcto.

Figura 5.12 Prueba ver temas de ayuda

<b>PRUEBA 013</b>	<b><i>Consultar ayuda sobre un tema</i></b>
Acción realizada	Selección de un tema de ayuda.
Resultado esperado	Se mostrará la ayuda correspondiente al tema elegido.
Resultado obtenido	Ayuda mostrada.
Observaciones	Correcto.

Figura 5.13 Prueba consultar un tema de ayuda

<b>PRUEBA 014</b>	<b><i>Reproducir texto escrito (a)</i></b>
Acción realizada	Selección del botón destinado a la reproducción de manera oral del texto.
Resultado esperado	Se deberá reproducir el texto y se seguirá mostrando la pantalla que contiene el botón de reproducción.
Resultado obtenido	Texto reproducido oralmente sin cambio de pantalla.
Observaciones	Correcto.

Figura 5.14 Prueba reproducir texto (a)

<b>PRUEBA 015</b>	<b><i>Reproducir texto escrito (b)</i></b>
Acción realizada	Selección del botón destinado a la reproducción de manera oral del texto.
Resultado esperado	Se deberá reproducir el texto y se mostrará la pantalla anterior con el cuadro de texto vacío.
Resultado obtenido	Texto reproducido oralmente con cambio de pantalla y cuadro de texto vacío.
Observaciones	Correcto.

Figura 5.15 Prueba reproducir texto (a)

<b>PRUEBA 016</b>	<b><i>Desactivar barrido</i></b>
Acción realizada	Selección de la opción "NO" (barrido no activado) en la pantalla configuración y selección del botón "ACEPTAR".
Resultado esperado	Mostrado de mensaje advirtiendo de la necesidad de reiniciar. Al reiniciar el usuario la aplicación, el barrido estará desactivado.
Resultado obtenido	Mostrado de mensaje y barrido activado al reiniciar.
Observaciones	Incorrecto.
Actuación	Problema solucionado en la última versión del componente.

Figura 5.16 Prueba desactivar barrido

<b>PRUEBA 017</b>	<b><i>Cambiar tiempo de barrido</i></b>
Acción realizada	Selección de un tiempo de barrido en la pantalla configuración.
Resultado esperado	Mostrado de mensaje advirtiendo de la necesidad de reiniciar. Al reiniciar el usuario la aplicación, el tiempo de barrido habrá cambiado.
Resultado obtenido	Mostrado de mensaje y tiempo de barrido actualizado.
Observaciones	Correcto.

Figura 5.17 Prueba cambiar tiempo de barrido

<b>PRUEBA 018</b>	<b><i>Cambiar configuración</i></b>
Acción realizada	Selección de una de las configuraciones que se muestran en la pantalla configuración.
Resultado esperado	Mostrado de mensaje advirtiendo de la necesidad de reiniciar. Al reiniciar el usuario la aplicación, ésta presentará la apariencia y funcionalidad dada por la nueva configuración.
Resultado obtenido	Mostrado de mensaje y configuración actualizada.
Observaciones	Correcto.

Figura 5.18 Cambiar configuración

### 5.3 Pruebas para casos límites

<b>PRUEBA 019</b>	<b><i>Borrado cuando el cuadro de texto está vacío</i></b>
Acción realizada	Selección del botón de borrado cuando el cuadro de texto no contiene ningún texto.
Resultado esperado	No ocurrirá nada.
Resultado obtenido	No ha ocurrido nada.
Observaciones	Correcto.

Figura 5.19 Prueba borrar sin texto

<b>PRUEBA 020</b>	<b><i>Reproducción cuando el cuadro de texto está vacío</i></b>
Acción realizada	Selección del botón de reproducción cuando el cuadro de texto no contiene ningún texto.
Resultado esperado	No ocurrirá nada.
Resultado obtenido	No ha ocurrido nada.
Observaciones	Correcto.

Figura 5.20 Prueba reproducir sin texto

<b>PRUEBA 021</b>	<b><i>Reproducción cuando el texto no es reproducible.</i></b>
Acción realizada	Selección del botón de reproducción cuando el texto no es reproducible en el idioma para el que esta creado el componente reproductor.
Resultado esperado	Se mostrará un mensaje de error advirtiendo de tal circunstancia.
Resultado obtenido	Mostrado de mensaje de error.
Observaciones	Correcto.

Figura 5.21 Prueba reproducir texto no reproducible

<b>PRUEBA 022</b>	<b><i>Cambio de las imágenes de botones variables.</i></b>
Acción realizada	Esperar a que los botones de imágenes variables de una pantalla cambien sus imágenes cuando el barrido termine de recorrerlos.
Resultado esperado	Los botones cambiarán sus imágenes por las siguientes en la lista de imágenes de la pantalla.
Resultado obtenido	Los botones no han cambiado de imágenes.
Observaciones	Incorrecto.
Actuación	Problema solucionado en una versión posterior.

Figura 5.22 Prueba cambio de imágenes de botones variables





## **6. Conclusiones**



## 6.1 Objetivos alcanzados

Con el desarrollo del componente teclado se ha conseguido la gran mayoría de los objetivos propuestos al principio de este proyecto, a continuación se detallan:

- Se ha desarrollado el componente fácilmente reutilizable.
- El componente implementa un teclado amigable y fácil de utilizar que permite realizar la escritura a los usuarios.
- El componente incluye un sistema de escritura basado en caracteres y otro basado en imágenes.
- Es posible realizar cambios en la apariencia y funcionalidad del teclado de una manera sencilla, e incluso se pueden añadir nuevas configuraciones, como por ejemplo una configuración que cree un teclado cuyas pantallas estén formadas por botones con imágenes del alfabeto Bliss. Este objetivo se logra gracias a la incorporación de los ficheros XML.
- El teclado posee un sistema de barrido que, mediante la conexión de un pulsador a la PDA, facilita la utilización del sistema a los usuarios cuyo grado de movilidad sea muy bajo. El barrido puede ser desactivado y activado por el usuario del teclado.
- Se ha conseguido integrar el componente en una aplicación real (comunicador) que permite la comunicación de las personas con parálisis cerebral con su entorno.
- Se ha conseguido integrar en la aplicación el componente desarrollado en este proyecto con otro componente (ya desarrollado) que se encarga de reproducir de manera oral el texto que se haya escrito utilizando el teclado.
- Se da la posibilidad de que el usuario elija, desde la propia aplicación, la configuración con la que desea que el componente trabaje.

## 6.2 Conocimientos Adquiridos

Con el desarrollo de este proyecto se han adquirido gran cantidad de conocimientos de tipo técnicos y ha sido posible aplicar multitud de metodologías, estudiadas a lo largo de la carrera, a un sistema real.

En cuanto a los conocimientos técnicos, el más importante es el aprendizaje de la filosofía de trabajo de la tecnología .NET y de su entorno de desarrollo. Dado el gran auge de esta tecnología en la actualidad, el haberse familiarizado con su entorno de desarrollo y con sus lenguajes de

programación (en especial C#) puede suponer una buena oportunidad para la entrada en el mercado laboral.

Además de los conocimientos adquiridos, se ha comprobado lo útil que puede llegar a ser la informática en el colectivo de personas con parálisis cerebral.

### 6.3 Líneas de trabajo futuras

A continuación se comentaran propuestas para ampliar y mejorar este proyecto:

- Integrar en el comunicador nuevos componentes que aumenten la funcionalidad del mismo, como por ejemplo un componente que permita al usuario final crear sus propias configuraciones.
- Añadir una nueva funcionalidad al componente que permita al usuario elegir la posición, apariencia y funcionalidad de los botones que componen el teclado.
- Añadir un componente en el comunicador que permita la comunicación basada en bluetooth o wifi de la PDA con otra PDA o incluso con un PC con la finalidad de que sea posible el envío de mensajes entre los dos dispositivos.
- Desarrollar una aplicación de escritorio que permita al propio usuario la creación de los ficheros XML de una manera gráfica.





# Apéndices



## APÉNDICE A. Tecnología utilizada

### A.1 Microsoft.NET

Microsoft .NET es un programa de software que conecta información, usuarios, sistemas y dispositivos. Incluye clientes, servidores y herramientas para programadores, y está formado por:

- Windows .NET Framework que permite generar y ejecutar todo tipo de software, incluidas aplicaciones basadas en Web, aplicaciones cliente inteligentes y servicios Web XML. Estos componentes facilitan la integración, ya que comparten datos y funcionalidad a través de una red mediante protocolos estándar independientes de la plataforma, como XML, SOAP y HTTP.
- Varias herramientas para programadores, como Microsoft Visual Studio.NET, que ofrece un entorno de desarrollo integrado (IDE) para sacar el máximo partido a la productividad de los programadores con Windows .NET Framework.
- Un conjunto de servidores, incluidos Microsoft Windows Server 2003, Microsoft SQL Server y Microsoft BizTalk Server, que integran, ejecutan, operan y administran servicios Web XML y aplicaciones basadas en Web.
- Software cliente, como Windows XP, Windows CE o Microsoft Office XP, que ayuda a los programadores a ofrecer una experiencia positiva para el usuario a través de la amplia familia de dispositivos y productos existentes.

#### A.1.1 .NET FRAMEWORK

.NET Framework es el nuevo modelo de programación de Microsoft para desarrollar, implementar y ejecutar servicios Web XML y todos los tipos de aplicaciones (de escritorio, para dispositivos móviles o basadas en Web). Uno de los objetivos que Microsoft perseguía con la creación de la plataforma .NET era ofrecer un marco de trabajo moderno de tercera generación para crear e implementar servicios Web XML. Con .NET Framework, Microsoft mejora enormemente la productividad del programador, la facilidad de desarrollo y la ejecución de aplicaciones confiables. .NET Framework incorpora servicios Web XML, un concepto totalmente nuevo en informática. Los servicios Web XML integran aplicaciones y componentes poco complementados diseñados para el heterogéneo entorno informático actual mediante la comunicación con protocolos de Internet estándar como SOAP, WSDL (Lenguaje de descripción de servicios Web) y UDDI (Integración, descubrimiento y descripción universal).

.NET Framework está formado por tres partes principales. La primera es Common Language Runtime (CLR), que asume la responsabilidad de ejecutar la aplicación. CLR garantiza que se cumplan todas las dependencias de la aplicación, administra la memoria y controla cuestiones como la seguridad y la integración de lenguajes. Common Language Runtime proporciona muchos servicios que simplifican el desarrollo del código y la implementación de la aplicación a la vez que

aumenta la fiabilidad de la aplicación. La mayor parte de este trabajo se controla de manera transparente, simplificando las tareas de los desarrolladores y administradores.

La segunda parte es la de las clases principales unificadas. Estas clases proporcionan todos los recursos que requiere un desarrollador para generar una aplicación moderna, incluida la compatibilidad con XML, las conexiones de red y el acceso a datos. Tener estas clases unificadas significa que un desarrollador que desarrolle cualquier tipo de aplicación, basada en Windows o en Web, utiliza las mismas clases. Esta coherencia aumenta la productividad del desarrollador y la reutilización de código.

La tercera y última parte es la de las clases de presentación, que incluyen ASP.NET para el desarrollo de aplicaciones Web, así como servicios WEB XML y Windows Forms para el desarrollo de aplicaciones basadas en Windows o de "cliente inteligente".

### **A.1.1.2 .NET COMPACT FRAMEWORK**

.NET Compact Framework aporta la eficacia del entorno de programación .NET Framework a los dispositivos. .NET Compact Framework es un entorno independiente del hardware para la ejecución de programas en dispositivos de computación con limitaciones de recursos, entre los que se encuentran los asistentes de datos personales (PDA) como Pocket PC, teléfonos móviles, descodificadores de televisión, dispositivos de computación para automóviles y dispositivos incrustados de diseño personalizado, que están integrados en el sistema operativo Windows CE .NET.

.NET Compact Framework es un subconjunto de la biblioteca de clases .NET Framework y también contiene clases diseñadas expresamente para él. Hereda la arquitectura .NET Framework completa de Common Language Runtime y la ejecución de código administrado.

.NET Compact Framework ofrece las siguientes funciones principales:

- Ejecuta programas independientes del hardware y el sistema operativo.
- Admite protocolos de red comunes y se conecta perfectamente con servicios XML Web.
- Proporciona a los desarrolladores un modelo para orientar sus aplicaciones y componentes ya sea a una amplia gama de dispositivos o a una categoría específica de éstos.
- Facilita el diseño y la optimización de los recursos de sistema limitados.
- Obtiene un rendimiento óptimo en la generación de código nativo cuando se utiliza compilación Just-In-Time (JIT).

### **A.1.1.3 Visual Studio.NET**

Es la nueva versión de la familia de herramientas de desarrollo de software de Microsoft, orientadas naturalmente hacia su nuevo entorno de programación: .NET Framework.

Es posible la escritura de programas empleando sólo el SDK de .NET Framework, pero al estar compuesto de herramientas independientes, constituye un medio más incómodo de trabajo.

Visual Studio.NET, al tratarse de un entorno de desarrollo integrado (Integrated Development Environment o IDE) incluye todas las herramientas del SDK: compiladores, editores, ayuda, etc., facilitando en gran medida la creación de programas.

#### **A.1.1.3.1 Visual Studio.NET 2003 para dispositivos móviles**

Visual Studio .NET 2003 ofrece un entorno de desarrollo robusto para la creación de aplicaciones destinadas a .NET Compact Framework. Junto con Visual Studio.NET se incluyen un conjunto de perfiles de dispositivos ya creados. Un perfil de dispositivo contiene la información necesaria para crear aplicaciones destinadas a determinados dispositivos. Con Visual Studio .NET, hay perfiles que permiten crear aplicaciones para Pocket PC, Pocket PC 2002 y Windows CE .NET 4.1 y posterior.

Estos perfiles le permiten crear aplicaciones que incluyen Windows Forms y ADO.NET, y ofrecen la posibilidad de consumir servicios Web.

Los perfiles pueden ser específicos de los dispositivos, como los destinados a Pocket PC, plataformas menos específicas destinadas a la plataforma WINDOWS CE en general o perfiles genéricos destinados a cualquier plataforma a la que se haya transferido .NET Compact Framework.

#### **A.1.1.4 Visual C#.NET**

Visual C# .NET proporciona a los programadores un lenguaje moderno y orientado a componentes con el que pueden crear con rapidez soluciones interesantes controladas por datos. Gracias a la rapidez de diseño, desarrollo e implementación de servicios Web XML, las soluciones controladas por datos de alto rendimiento y los diseñadores visuales en servidor, Visual C# .NET ofrece una funcionalidad superior para optimizar los procesos empresariales.

Permite a los programadores generar soluciones para una gama muy amplia de clientes, incluidas aplicaciones Web, aplicaciones basadas en Windows Forms y dispositivos de cliente ligero e inteligentes. Además, con un lenguaje de programación elegante y moderno, los programadores de C# pueden aprovechar sus conocimientos de C++ y Java para desarrollar con el entorno de Microsoft .NET.

##### ➤ Principales características

- C# es un lenguaje de programación simple pero eficaz, diseñado para escribir aplicaciones empresariales.
- El lenguaje C# es una evolución de los lenguajes C y C++. Utiliza muchas de las características de C++ en las áreas de instrucciones, expresiones y operadores.
- C# presenta considerables mejoras e innovaciones en áreas como seguridad de tipos, control de versiones, eventos y recolección de elementos no utilizados (liberación de memoria).
- C# proporciona acceso a los tipos de API más comunes: .NET Framework, COM, Automatización y estilo C.

### A.1.2 XML

#### A.1.2.1 ¿Qué es XML?

XML es un acrónimo cuyo significado en inglés es EXtensible Markup Language (Lenguaje de formato ampliable). Es decir, es un lenguaje de marcado, como el lenguaje HTML, lo cual significa que utiliza etiquetas.

Es un lenguaje usado para estructurar información en un documento o en general en cualquier fichero que contenga texto (ficheros de configuración de un programa...) y distribuirla en un formato independiente de la plataforma.

Ha ganado muchísima popularidad en los últimos años debido a ser un estándar abierto y libre, creado por el W3C (los creadores de la www), en colaboración con las principales compañías productoras de software.

XML es un sistema independiente de la plataforma porque no usa un lenguaje específico. Las etiquetas de XML no están predefinidas, lo cual significa que cada uno escribe sus propias etiquetas.

#### ➤ Características

- Lenguaje ampliable: se puede definir cualquier conjunto de etiquetas adicionales sin que se bloquee la aplicación.
- Plataformas distintas, un único lenguaje: La portabilidad de XML es consecuencia de que es el propio desarrollador el que define las etiquetas y los atributos. No se necesitan bibliotecas ni servidores de aplicaciones especiales para leer un documento XML. Los documentos XML son archivos de texto normal, por lo que no requieren un software propietario para interpretarlos, como ocurre con la mayoría de los archivos binarios.
- El contenido es independiente de la presentación: Con XML, se puede reducir el riesgo de incluir contenido redundante. Sus clientes se concentrarán en usar HTML y CSS para definir el diseño y la presentación, lo cual no se verá afectado por cambios en la información, que se almacena por separado en un archivo XML.

#### ➤ Principales usos de XML

- XML aplicado a los **sitios web**: permite separar contenido y presentación, y que los mismos datos se puedan mostrar de varias formas distintas sin demasiado esfuerzo.
- XML para la **comunicación** entre aplicaciones: representación de los datos muy simple, fácil de transmitir por la red, estándar. En los últimos tiempos este uso se está haciendo muy popular con el surgimiento de los Servicios web.
- XML para la **configuración** de programas: representación de los datos simple y estándar.

### A.1.2.2 Estructura de XML

XML consta de cuatro especificaciones (el propio XML sienta las bases sintácticas y el alcance de su implementación):

- DTD (Document Type Definition): Definición del tipo de documento. Es en general, un archivo que encierra una definición formal de un tipo de documento y, a la vez, especifica la estructura lógica de cada documento. Define tanto los elementos de una página como sus atributos. El DTD del XML es opcional. En tareas sencillas no es necesario construir una DTD, entonces se trataría de un documento "bien formado" y si lleva DTD será un documento "validado".
- XSL (eXtensible Stylesheet Language): Define o implementa el lenguaje de estilo de los documentos escritos para XML. Permite modificar el aspecto de un documento. Se pueden lograr múltiples columnas, texto girado, orden de visualización de los datos de una tabla, múltiples tipos de letra con amplia variedad en los tamaños. Se considera más potente que las hojas de estilo en cascada, usado en un principio con el lenguaje DHTML.
- XLL (eXtensible Linking Language): Define el modo de enlace entre diferentes enlaces. Desde marzo de 1998 el W3C trabaja en los enlaces y direccionamientos del XML. Provisionalmente se le renombró como Xlink y a partir de junio se le denomina XLL. Este lenguaje de enlaces extensible tiene dos importantes componentes: Xlink y el Xpointer. Va más allá de los enlaces simples que sólo soporta el HTML. Se podrá implementar con enlaces extendidos.
- XUA (XML User Agent): Estandarización de navegadores XML. Todavía está en proceso de creación de borradores de trabajo. Se aplicará a los navegadores para que compartan todas las especificaciones XML.

### A.1.2.3 XML en .NET FRAMEWORK

El diseño de XML para trabajar con el .NET Framework, estuvo basado en los siguientes objetivos:

- Compatibilidad con los estándares del W3C.
- Extensibilidad.
- Arquitectura conectable.
- Rendimiento e integración estrecha con ADO.NET.

XML proporciona a los programadores la capacidad de ofrecer datos estructurados desde muchas aplicaciones al sistema local con el fin de trabajar localmente con ellos. Además, por su parte, .NET Framework nos proporciona un conjunto de clases que están categorizadas de acuerdo a la funcionalidad que ofrecen, como lectura y escritura de documentos XML, validación de documentos XML, navegación y selección de nodos, administración de esquemas, y transformación de documentos XML.

.NET Framework ofrece la posibilidad de diseñar un conjunto integrado de clases XML e innovar en el entorno XML. Las clases XML que se suministran son elementos básicos de .NET Framework. Estas clases ofrecen una solución abierta, interoperable y compatible con estándares

para superar las dificultades a las que se enfrentan actualmente los programadores. Toda esta funcionalidad se encuentra dentro del espacio de nombres `system.xml.dll`, entre los más usados tenemos: `System.xml`, `System.Xml.Schema`, `System.Xml.XPath`, `System.Xml.Xsl`.

Una funcionalidad muy útil que nos brinda el .NET Framework, es la posibilidad de trabajar con serialización de objetos. Todas las clases que tienen esta funcionalidad se encuentran agrupadas en el espacio de nombres `System.Xml.Serialization`. La serialización XML escribe instancias de objetos como documento XML y lee objetos a partir de documentos XML. La serialización es usada en la programación Web en combinación con SOAP y dentro del marco del .NET Framework XML Web Services.

### ➤ Espacio de nombres `System.Xml`

Este espacio de nombres tiene un conjunto completo de clases XML para análisis, validación y manipulación de datos XML mediante sistemas de lectura, sistemas de escritura y componentes compatibles con el consorcio World Wide Web (W3C) DOM. También se explican las consultas XPath (XML Path Language) y las transformaciones XSLT (Extensible Stylesheet Language Transformations).

La lista siguiente contiene las clases principales del espacio de nombres XML:

- La clase **XmlTextReader** proporciona acceso rápido de lectura, sin almacenamiento en caché y con desplazamiento sólo hacia delante a datos XML.
- La clase **XmlNodeReader** proporciona un objeto `XmlReader` a través del subárbol de nodo DOM dado.
- La clase **XmlValidatingReader** proporciona validación de esquemas DTD, XDR y XSD.
- La clase **XmlTextWriter** proporciona una forma rápida y de desplazamiento sólo hacia delante para generar código XML.
- La clase **XmlDocument** implementa las especificaciones W3C, Document Object Model Level 1, Core y Core DOM Level2.
- La clase **XmlDataDocument** proporciona una implementación de un objeto `XmlDocument` que se puede asociar a un objeto `DataSet`. Los datos XML estructurados se pueden ver y manipular simultáneamente a través de la representación relacional del objeto `DataSet` o de la representación de árbol del objeto `XmlDataDocument`.
- La clase **XPathDocument** proporciona una caché rápida y de alto rendimiento con el fin de procesar documentos XML para XSLT.
- La clase **XPathNavigator** proporciona un modelo de datos W3C XPath 1.0 sobre un almacén con un modelo de desplazamiento de tipo cursor.
- La clase **XslTransform** corresponde a un procesador XSLT compatible con la especificación W3C XSLT 1.0 con el fin de transformar documentos XML.

- Las clases del modelo de objetos **XmlSchema** proporcionan un conjunto de clases que se pueden examinar y que reflejan directamente la especificación W3C XSD. Proporcionan la capacidad de crear esquemas XSD mediante programación.
- La clase **XmlSchemaCollection** proporciona una biblioteca de esquemas XDR y XSD. Estos esquemas, almacenados en memoria, proporcionan validación rápida en tiempo de análisis para el objeto XmlValidatingReader.

#### A.1.2.4 Clases utilizadas para la lectura y escritura de ficheros XML

En este apartado se explicarán brevemente las clases ofrecidas por la tecnología .NET para el manejo de los datos de los ficheros XML que se han utilizado para la implementación de la solución.

##### **XmlReader**

La clase XmlReader es una clase base abstracta que proporciona acceso a datos XML de sólo avance y de sólo lectura sin almacenamiento en caché. Cumple con el Lenguaje de marcado extensible (XML) 1.0 de W3C y los espacios de nombres en recomendaciones XML.

La clase XmlReader admite la lectura de datos XML a partir de una secuencia o archivo. Define métodos y propiedades que permiten desplazarse a través de los datos y leer los contenidos de un nodo. El nodo actual hace referencia al nodo en el que está situado el sistema de lectura. Para avanzar el sistema de lectura, se podrá utilizar cualquiera de los métodos de lectura, existirán propiedades que devuelven el valor del nodo actual.

La clase XmlReader permite:

- Verificar que los caracteres son caracteres legales XML y que los nombres de elemento y atributo son nombres XML válidos.
- Verificar que el documento XML tiene el formato correcto.
- Validar los datos frente a la DTD o esquema.
- Devolver datos de secuencia XML u omitir registros no deseados mediante un modelo de extracción.

En este proyecto se ha utilizado, para la lectura, la clase XmlNodeReader que hereda de XmlReader. Esta clase representa un lector que proporciona acceso rápido, sin almacenamiento en caché y con desplazamiento sólo hacia delante, a los datos XML de un objeto XmlNode (representa un único nodo del documento XML). La clase XmlNodeReader no es compatible con la definición de tipo de documento (DTD) ni con la validación de esquemas y, por tanto, no valida los datos XML que lee.

### **XmlTextWriter**

Representa un sistema de escritura que proporciona un medio rápido, sin almacenamiento en caché y con desplazamiento sólo hacia delante para generar secuencias o archivos con datos XML que satisface las recomendaciones relativas a espacios de nombres en XML y Extensible Markup Language (XML) 1.0 del Consorcio W3C.

El objeto XmlTextWriter es una implementación de la clase XmlWriter que proporciona la API y que escribe datos XML en un archivo, una secuencia o un objeto TextWriter. Esta clase proporciona muchas reglas de validación y comprobación para garantizar que la sintaxis de los datos XML escritos es correcta. Cuando se producen determinadas infracciones, se provocan excepciones que podrán detectarse. El objeto XmlTextWriter tiene distintos constructores, cada uno de los cuales especifica un tipo de ubicación diferente en la que se van a escribir los datos XML.

## APÉNDICE B. Estructura de los ficheros

### B.1 Estructura de los ficheros

El esquema que siguen todos los ficheros de definición de una configuración es el siguiente:

```
<aplicacion>
  <elemento>
    <nombre></nombre>
    <panel>
    </panel>
    <labels>
      <label>
      </label>
      .....
    </labels>
    <botones>
      <boton>
      </boton>
      .....
    </botones>
    <imagenes_boton>
      <imagen>
      </imagen>
      .....
    </imagenes_boton>
    <textBox>
    </textBox>
    <botonesRadio>
      <botonR>
      </botonR>
      .....
    </botonesRadio>
    <listas>
      <listaSeleccion>
      </listaSeleccion>
      .....
    </listas>
  </elemento>
  .....
</aplicacion>
```

### ➤ Explicación

La definición de los datos de una configuración comienza con la etiqueta `<aplicacion>` a partir de la cual y antes de llegar a la etiqueta de cierre `</aplicacion>`, se considerará que todo el contenido pertenece a la configuración que se quiere crear. En el siguiente nivel de profundidad, aparecerán las etiquetas de apertura y cierre de la información correspondiente a una pantalla de la aplicación (`<elemento>` `</elemento>`), se podrán añadir tantas pantallas como se desee. En el siguiente nivel de profundidad y dentro de las etiquetas `<elemento>``</elemento>` aparecerán las etiquetas `<nombre>``</nombre>` que contendrán el nombre de la pantalla.

En los siguientes apartados se explica la forma en la que se define la información correspondiente a cada elemento que puede ser introducido en una pantalla.

### **B.1.1 Información de las pantallas**

De cara a la creación de las pantallas que formarán el teclado, se almacena en los ficheros de apariencia y funcionalidad la siguiente información:

```
<panel>
  <mostrar></mostrar>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
</panel>
```

Dentro de las etiquetas `<mostrar>``</mostrar>` se escribirá “si” o “no” en función de si la pantalla a crear va a mostrar en su parte superior el campo de texto que muestra el texto que el usuario ha escrito hasta el momento. Dentro de las etiquetas `<x>``</x>` aparecerá la coordenada ‘x’ del vértice superior izquierdo de la pantalla y dentro de las etiquetas `<y>``</y>` aparecerá la coordenada ‘y’ del vértice superior izquierdo de la pantalla. En las etiquetas `<ancho>``</ancho>` y `<alto>``</alto>` aparecerán respectivamente el ancho y el alto del panel.

### **B.1.2 Información de las etiquetas**

La información de las etiquetas de texto se almacenará dentro de las etiquetas `<labels>``</labels>`, la información de cada una de las etiqueta de texto que se desee añadir (pueden ser más de uno) debe ir almacenada de una de las siguientes formas:

```

<label>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <mensaje></mensaje>
  <font>
    <familia></familia>
    <tamaño></tamaño>
  </font>
  <alineacion></alineacion>
</label>

```

a) No define colores

```

<label>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <mensaje></mensaje>
  <font>
    <familia></familia>
    <tamaño></tamaño>
  </font>
  <alineacion></alineacion>
  <textocolor>
    <rojo></rojo>
    <verde></verde>
    <azul></azul>
  </textocolor>
  <color>
    <rojo></rojo>
    <verde></verde>
    <azul></azul>
  </color>
</label>

```

b) define colores

En las etiquetas `<x></x>`, `<y></y>` aparecerán respectivamente las coordenadas 'x' e 'y' del vértice superior izquierdo de la etiqueta. Dentro de las etiquetas `<ancho></ancho>`, `<alto></alto>` aparecerán respectivamente el ancho y el alto de la etiqueta. En las etiquetas `<mensaje></mensaje>` aparecerá el texto asociado a la etiqueta, es decir, el texto que se mostrará en la pantalla. En las etiquetas situadas dentro de las etiquetas `<font></font>` se escribirá la familia a la que se desea pertenezca el texto asociado a la etiqueta (en las etiquetas `<familia></familia>`) y el tamaño del mismo (en las etiquetas `<tamaño></tamaño>`). Dentro de las etiquetas `<alineacion></alineacion>` podrá aparecer una de las siguientes palabras:

- "centro": el texto de la etiqueta aparecerá centrado.
- "izquierda": el texto de la etiqueta aparecerá alineado a la izquierda.
- "derecha": el texto de la etiqueta aparecerá alineado a la derecha.

A partir de ahora se explicarán las etiquetas correspondientes al tipo b). Las diferencias entre ambas definiciones es que en la definición del tipo b) se define de manera explícita los colores de apariencia de la etiqueta, mientras que en la definición del tipo a) no se especifican colores porque a la hora de crear las etiquetas se cogerán los colores que están definidos por defecto en el fichero `configuracion.xml`; es por esto que si ejecutando la aplicación el usuario cambia los colores de apariencia de las etiquetas, únicamente cambiarán los colores de las etiquetas que no se definen de manera explícita.

En el tipo de definición b) se definirá el color que se desea que tenga el texto y el fondo de la etiqueta en función del modo RGB (Red Green Blue), para ello en las etiquetas `<rojo></rojo>`, `<verde></verde>` y `<azul></azul>` contenidas tanto en las etiquetas `<textocolor></textocolor>`

(color del texto), como en las etiquetas `<color></color>` (color de fondo de la etiqueta) deberá aparecer un número (entre 0 y 255) que indique la intensidad de cada uno de los colores (rojo, verde y azul) a introducir en un píxel de la pantalla.

### B.1.3 Información de los botones

Antes de comenzar con la forma de almacenar la información correspondiente a los botones, se explicarán las distintas funcionalidades que puede tener asociadas un botón. Es decir, los efectos que tendrán la selección de los botones del teclado:

- **PANTALLA:** Al seleccionar un botón con esta funcionalidad, dejará de mostrarse la pantalla actual y se mostrará otra pantalla (será el botón el encargado de decir cuál es la nueva pantalla a mostrar).
- **ESCRIBIR:** Al seleccionar un botón con esta funcionalidad, se añadirá al cuadro de texto de la pantalla el texto asociado al botón. Si sobre el botón aparece la palabra ESPACIO, al pulsarlo se introducirá un espacio en blanco.
- **BORRAR:** Al pulsar un botón con esta funcionalidad, se borrará el último carácter escrito en el cuadro de texto.
- **BORRAR\_PALABRA:** Al pulsar un botón con esta funcionalidad, se borrará la última palabra escrita en el cuadro de texto.
- **ESCRIBIR\_Y\_VOLVER:** Al pulsar un botón con esta funcionalidad, el efecto sobre el cuadro de texto es el mismo que el que se tenía con la funcionalidad de ESCRIBIR, pero la pantalla actual dejará de mostrarse y se mostrará una nueva pantalla (también indicada por el botón).
- **EXIT:** Al pulsar un botón con esta funcionalidad, se cierra la aplicación guardando los cambios de configuración realizados durante la sesión de trabajo para que permanezcan en la siguiente sesión.
- **PLAY:** Al pulsar un botón con esta funcionalidad, se reproduce el texto que esté escrito en el cuadro de texto.
- **PLAY\_Y\_VOLVER:** Al pulsar un botón con esta funcionalidad, el efecto es el mismo que con la funcionalidad de PLAY salvo que se deja de mostrar la pantalla actual y se muestra una nueva pantalla con el cuadro de texto vacío (de nuevo el botón será el encargado de decir qué pantalla se debe mostrar). Esta funcionalidad debe su existencia a que rara vez se dice la misma expresión de maneras consecutivas.
- **CONFIGURAR:** El botón con esta funcionalidad asociada aparecerá en la pantalla en la que se permita realizar cambios en la configuración (barrido y configuración utilizada). El efecto que tiene el pulsar este botón es que se aplican los cambios realizados en la configuración.
- **CAMBIO\_COLOR\_Y\_PANTALLA:** Los botones con esta funcionalidad, aparecerán en la pantalla de selección de colores para un determinado elemento de la pantalla (labels, botones, fondo de la pantalla...). Al pulsar sobre alguno de estos botones, se modificará el

color de apariencia del elemento que se hubiera seleccionado antes de entrar a esta pantalla (el color lo debe definir el botón) y después se muestra otra pantalla (definida también por el botón).

- ATRAS: Esta funcionalidad sólo podrá estar asociada a un botón de imagen que esté situado en una pantalla formada por botones de escritura. Al pulsar un botón de imagen con esta funcionalidad cambiarán las imágenes de los botones, que así estén definidos (botones de imagen variable), por las que tenían anteriormente o si la pantalla actual es la primera dentro de una categoría, se mostrará entonces la pantalla de selección de categorías.

### B.1.3.1 Botones de caracteres

En una pantalla se incluirán objetos de la clase Button de Visual Studio.NET. Para definir los botones de caracteres que aparecerán en una determinada pantalla se debe añadir, por cada botón que se desee introducir, la información que a continuación se detalla introducida entre las etiquetas <botones></botones>.

En base a la funcionalidad asociada al botón existen diferentes formatos para almacenar su información. A continuación se explican:

```
<boton>
  <es_imagen>no<es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <texto></texto>
  <funcion>PANTALLA</funcion>
  <pantalla></pantalla>
  <font>
    <familia></familia>
    <tamaño></tamaño>
  </font>
</boton>
```

a) Esquema para las funciones

```
<boton>
  <es_imagen>no<es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <texto></texto>
  <funcion>ESCRIBIR</funcion>
  <a_escribir></a_escribir>
  <font>
    <familia></familia>
    <tamaño></tamaño>
  </font>
</boton>
```

b) Función: Escribir carácter

```
<boton>
  <es_imagen>no<es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <texto></texto>
  <funcion></funcion>
  <font>
    <familia></familia>
    <tamaño></tamaño>
  </font>
</boton>
```

c) Esquema para las funcionalidades de BORRAR y BORRAR\_PALABRA

```
<boton>
  <es_imagen>no<es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <texto></texto>
  <funcion>ESCRIBIR_Y_VOLVER</funcion>
  <a_escribir></a_escribir>
  <pantalla></pantalla>
  <font>
    <familia></familia>
    <tamaño></tamaño>
  </font>
</boton>
```

d) Función: Escribir e ir a otra pantalla

```
<boton>
  <es_imagen>no<es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <texto></texto>
  <funcion>EXIT</funcion>
  <font>
    <familia></familia>
    <tamaño></tamaño>
  </font>
</boton>
```

e) Función: Salir de la aplicación

```
<boton>
  <es_imagen>no<es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <texto></texto>
  <funcion>PLAY</funcion>
  <font>
    <familia></familia>
    <tamaño></tamaño>
  </font>
</boton>
```

f) Función: Reproducir un texto escrito

```

<boton>
  <es_imagen>no<es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <texto> </texto>
  <funcion>PLAY_Y_VOLVER</funcion>
  <pantalla></pantalla>
  <font>
    <familia></familia>
    <tamaño></tamaño>
  </font>
</boton>

```

g) Función: Reproducir e ir a otra pantalla

```

<boton>
  <es_imagen>no<es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <texto></texto>
  <funcion>CONFIGURAR</funcion>
  <font>
    <familia></familia>
    <tamaño></tamaño>
  </font>
</boton>

```

h) Función: Guardar configuración.

```

<boton>
  <es_imagen>no<es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <funcion>CAMBIO_COLOR_Y_PANTALLA</funcion>
  <pantalla></pantalla>
  <color>
    <rojo></rojo>
    <verde></verde>
    <azul></azul>
  </color>
</boton>

```

i) Función: Cambiar el color de un elemento de la pantalla e ir a otra pantalla

En las anteriores alternativas para almacenar la información de los botones de texto hay partes comunes, como:

- <es\_imagen></es\_imagen> aquí aparecerá la palabra “si” en el caso de que la información corresponda a un botón de imagen o la palabra “no” si la información corresponde a un botón de texto. En lo anteriores esquema aparece escrito “no” por ser esquemas de almacenamiento de información de botones de texto.
- <x></x> aquí se almacenará la coordenada ‘x’ del borde superior izquierdo del botón.
- <y></y> aquí se almacenará la coordenada ‘y’ del borde superior izquierdo del botón.
- <ancho></ancho> aquí se almacenará la anchura del botón.
- <alto></alto> aquí se almacenará la altura del botón.

- `<funcion></funcion>` aquí deberá ir escrita la funcionalidad asociada la botón. Todas las funcionalidades posibles se muestran escritas en los esquemas anteriores (salvo BORRAR Y BORRAR\_PALABRA que se muestran en el pie del esquema c).
- `<font></font>` Dentro de sus etiquetas `<familia></familia>` y `<tamaño></tamaño>` se podrá definir, respectivamente, la familia y el tamaño del texto que aparecerá escrito en el botón.

A continuación se explican las partes no comunes de los anteriores esquemas:

- `<pantalla></pantalla>` Entre estas etiquetas deberá aparecer el nombre de la pantalla que se desee mostrar cuando se pulse el botón. El nombre deberá coincidir con alguno de los nombres de pantalla definidos en el propio fichero entre las etiquetas `<nombre></nombre>` (que, como ya se ha visto, aparecerán al comienzo de la definición de la información de cada una de las pantallas).
- `<a_escribir></a_escribir>` Entre estas etiquetas aparecerá el texto que se desee añadir al cuadro de texto cuando se pulse el botón. Si se desea introducir un espacio en blanco al pulsar el botón, aquí deberá aparecer el texto “espacio”.
- En las etiquetas contenidas en `<color></color>` se especifica según el modelo RGB un color. Estos botones servirán para realizar los cambios en los colores de los distintos elementos de la pantalla.

### B.1.3.2 Botones de imagen

Para definir los botones de imagen que aparecerán en una determinada pantalla se debe añadir, entre las etiquetas `<botones></botones>`, por cada botón que se desee introducir la información que a continuación se detalla. En base a la funcionalidad asociada al botón y a si el botón es fijo o variable, existen diferentes formatos para almacenar su información. A continuación se explican:

```
<boton>
  <es_imagen>si</es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <funcion>PANTALLA</funcion>
  <pantalla></pantalla>
  <imagennormal></imagennormal>
  <imagenfoco></imagenfoco>
</boton>
```

a) Función: Ir a otra pantalla

```
<boton>
  <es_imagen>si</es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <funcion>ESCRIBIR</funcion>
  <pantalla></pantalla>
</boton>
```

b) Función: Escribir texto asociado a imagen.

```
<boton>
  <es_imagen>si</es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <funcion>ESCRIBIR_Y_VOLVER</funcion>
  <pantalla></pantalla>
</boton>
```

c) Función: Escribir e ir a otra pantalla.

```
<boton>
  <es_imagen>si</es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <funcion>BORRAR_PALABRA</funcion>
  <imagennormal></imagennormal>
  <imagenfoco></imagenfoco>
</boton>
```

d) Función: Borrar palabra

```
<boton>
  <es_imagen>si</es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <funcion>ATRAS</funcion>
  <pantalla></pantalla>
  <imagennormal></imagennormal>
  <imagenfoco></imagenfoco>
</boton>
```

e) Función: Ir a otra pantalla

```
<boton>
  <es_imagen>si</es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <funcion>EXIT</funcion>
  <imagennormal></imagennormal>
  <imagenfoco></imagenfoco>
</boton>
```

f) Función: Salir de la aplicación

```
<boton>
  <es_imagen>si</es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <funcion>PLAY</funcion>
  <imagennormal> </imagennormal>
  <imagenfoco> </imagenfoco>
</boton>
```

g) Reproducir texto escrito

```
<boton>
  <es_imagen>si</es_imagen>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <funcion>PLAY_Y_VOLVER</funcion>
  <pantalla></pantalla>
  <imagennormal> </imagennormal>
  <imagenfoco> </imagenfoco>
</boton>
```

h) Reproducir texto escrito e ir a otra pantalla

En los esquemas anteriores se puede ver la forma en que se almacena la información para la creación de botones de imagen. Los distintos tipos de esquemas tienen partes comunes, como son la localización y el tamaño de los botones, que se hace de idéntica manera que para los botones de texto, las etiquetas `<es_imagen></es_imagen>` que, por tratarse de esquemas de almacenamiento de información de los botones de imagen, contienen la palabra “sí” y las etiquetas `<funcion></funcion>` que, como en los botones de texto, contendrá la funcionalidad asociada al mismo.

La gran diferencia entre unos esquemas (a, d, e, f, g, h) y otros (b, c) son las etiquetas `<imagennormal></imagennormal>` e `</imagenfoco></imagenfoco>`. Para comenzar la explicación sobre esta diferencia, se empezará diciendo que las etiquetas `<imagennormal></imagennormal>` contendrán el nombre con el que está almacenada en memoria la imagen que deberá aparecer en el botón y las etiquetas `</imagenfoco></imagenfoco>` contendrán el nombre con el que está almacenada en memoria la imagen que deberá aparecer en el botón cuando éste esté señalado por el barrido. La causa por la que unos botones definen explícitamente sus imágenes y otros no es que los que las definen son botones que sólo tienen asociada una imagen, es decir, que la imagen no cambia durante la ejecución de la aplicación, mientras que los que no las definen son botones que dan la posibilidad de modificar su imagen durante la ejecución. Para que estos botones conozcan la imagen que deben mostrar en cada momento de la ejecución el componente debe consultar el siguiente fragmento de texto XML donde aparece una lista de las imágenes que deben aparecer sobre los botones de imágenes variables de la pantalla correspondiente:

```
<imagenes_boton>
  <imagen>
    <imagennormal></imagennormal>
    <imagenfoco></imagenfoco>
    <a_escribir></a_escribir>
  </imagen>
  .....
</imagenes_boton >
```

### a) Listado de imágenes rotatorias de una pantalla

En el esquema anterior se puede ver la forma en la que se debe introducir la información de cada imagen que se desee aparezca sobre algún botón de imagen variable de una determinada pantalla. Pueden existir muchas más imágenes que botones o incluso pueden existir menos imágenes que botones, en este caso los botones que no lleguen nunca a recibir una imagen a mostrar no se representarán en la pantalla (aparecerán “huecos en blanco”). Como los botones variables van a tener asociada la funcionalidad de escribir texto, se define aquí también el texto que se deberá añadir al cuadro de texto al pulsar el botón.

### B.1.3.3 Botones de selección

Los botones de selección son los pequeños botones redondos que permiten elegir una única opción que se esté tratando. En esta aplicación se han utilizado para dar la posibilidad al usuario de decidir, si en una determinada configuración, desea trabajar con barrido o no.

El esquema que se ha seguido en los ficheros es el siguiente:

```
<botonesRadio>
  <botonR>
    <x></x>
    <y></y>
    <ancho></ancho>
    <alto></alto>
    <texto>Si</texto>
  </botonR>
  <botonR>
    <x></x>
    <y></y>
    <ancho></ancho>
    <alto></alto>
    <texto>No</texto>
  </botonR>
</botonesRadio>
```

a) Esquema seguido para los RadioButtons

### B.1.4 Información de las listas de selección

En la pantalla CONFIGURACIÓN (presente en todas las configuraciones) se incluirán listas de selección para permitir seleccionar uno de los tiempos de barrido disponibles y también una configuración distinta de la actual.

El esquema que se ha seguido en los ficheros para almacenar la información de las listas de selección es el siguiente:

```
<listaSeleccion>
  <nombre></nombre>
  <x></x>
  <y></y>
  <ancho></ancho>
  <alto></alto>
  <enable></enable>
</listaSeleccion>
```

a) Esquema seguido para las listas de selección

A lo largo de este capítulo se ha explicado el significado de todas las etiquetas que aparecen en el esquema salvo el de la etiqueta <enable></enable>. Entre estas etiquetas puede aparecer la palabra 'si', si se desea que la lista esté activa, o 'no', si se desea que la lista no esté activa.

### B.2 Fichero para la persistencia de la configuración del componente

El Esquema que sigue el fichero que permite la persistencia de la configuración del componente es el siguiente:

```
<aplicacion>
  <barrido>
    <tiemposDisponibles>
      <tiempoD></tiempoD>
      .....
    </tiemposDisponibles>
    <activado></activado>
    <tiempo></tiempo>
  </barrido>
  <configuracion>
    <fichero></fichero>
    <colorFondo>
      <rojo></rojo>
      <verde></verde>
      <azul></azul>
    </colorFondo>
    <colorBoton>
      <rojo></rojo>
      <verde></verde>
      <azul></azul>
    </colorBoton>
    <colorTextoBoton>
      <rojo></rojo>
      <verde></verde>
      <azul></azul>
    </colorTextoBoton>
    <colorTexto>
      <rojo></rojo>
      <verde></verde>
      <azul></azul>
    </colorTexto>
    <colorFondoDefecto>
      <rojo></rojo>
      <verde></verde>
      <azul></azul>
    </colorFondoDefecto>
    <colorBotonDefecto>
      <rojo></rojo>
      <verde></verde>
      <azul></azul>
  </configuracion>
</aplicacion>
```

```

        </colorBotonDefecto>
        <colorTextoDefecto>
            <rojo></rojo>
            <verde></verde>
            <azul></azul>
        </colorTextoDefecto>
        <confDisponibles>
            <conf></conf>
            .....
        </confDisponibles>
    </configuracion>
</aplicacion>

```

Esquema fichero “configuracion.xml”

Como se puede ver, el fichero comienza la etiqueta <aplicacion> a partir de la cual y antes de la etiqueta de cierre correspondiente </aplicacion> se introducirá la información deseada.

Entre las etiquetas <barrido></barrido> aparecerá la información correspondiente al barrido:

- Tiempos de barrido disponibles: Entre las etiquetas <tiemposDisponibles></tiemposDisponibles> se pueden introducir tantas etiquetas <tiempoD></tiempoD> como opciones se desea que aparezcan en la lista de selección de tiempo de barrido que aparece en la pantalla CONFIGURACIÓN. Entre las etiquetas <tiempoD></tiempoD> debe aparecer el tiempo en segundos.
- Existe o no barrido y el tiempo de barrido: Para decidir si existe o no barrido se debe introducir, entre las etiquetas <activado></activado> un “si” o un “no” según se desee.
- Definición del tiempo de barrido: Para definir el tiempo de barrido (tiempo que cada botón estará señalado a la espera de ser seleccionado) se debe introducir entre las etiquetas <tiempo></tiempo> el número de milisegundos que se desee.

Entre las etiquetas <configuracion></configuracion> aparecerá la información necesaria para la creación de las distintas pantallas. Esta información será la siguiente:

- Entre las etiquetas <fichero></fichero> aparecerá el nombre del fichero que se utilizará la próxima vez que se reinicie la aplicación.
- Entre las etiquetas <colorFondo></colorFondo>, <colorBoton></colorBoton>, <colorTextoBoton></colorTextoBoton> y <colorTexto></colorTexto> aparecerán respectivamente y en base al modo RGB, los colores de fondo de pantalla, de fondo de los botones, del texto de los botones y del texto de las pantallas. Cada vez que se ejecute la aplicación y se necesite conocer el color de algún elemento se recurrirá a esta información. Cuando el usuario, mediante el uso de la aplicación modifique el color de apariencia de alguno de los anteriores elementos se modificará esta parte del fichero.
- Entre las etiquetas <colorFondoDefecto></colorFondoDefecto>, <colorBotonDefecto></colorBotonDefecto> y <colorTextoDefecto></colorTextoDefecto> aparecerán

respectivamente y en base al modo RGB, los colores que el diseñador del componente ha dado al fondo de las pantallas, a los botones y al texto de las pantallas.

- Entre las etiquetas `<confDisponibles>` `</confDisponibles>` aparecerán tantas etiquetas `<conf>``</conf>` como configuraciones posibles haya en el componente. Entre las etiquetas `<conf>``</conf>` deberá aparecer el nombre del fichero de configuración sin extensión.

## APÉNDICE C. Glosario de términos

En este apartado se muestra un listado de palabras y acrónimos que aparecen a lo largo de la memoria cuyo significado es conveniente explicar.

- **CAA:** Comunicación Alternativa y Aumentativa.
- **SPC:** Sistema Pictográfico de Comunicación. Es un sistema de comunicación alternativo y aumentativo.
- **PDA:** Personal Digital Assistant.
- **E-R:** Entidad-relación. Diagrama utilizado para la representación de datos.
- **.NET:** Proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma y que permita un rápido desarrollo de aplicaciones.
- **W3C:** World Wide Web Consortium.
- **DTD:** Document Type Definition. Define los tipos de elementos, atributos y entidades permitidas en un determinado fichero XML, y puede expresar algunas limitaciones para combinarlos.
- **API:** Application Programming Interface. conjunto de especificaciones de comunicación entre componentes software.
- **IDE:** Integrated Development Environment. Es un programa compuesto por un conjunto de herramientas para un programador.
- **SQL:** Structured Query Language. Lenguaje estructurado de consultas, estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.
- **SOAP:** Simple Object Access Protocol. Protocolo estándar creado por Microsoft, IBM y otros, está actualmente bajo el auspicio de la W3C que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web.
- **WSDL:** Web Service Definition Language. Es un formato XML que se utiliza para describir servicios Web. Describe la interfaz pública a los servicios Web.
- **UDDI:** Universal Description, Discovery, and Integration. UDDI es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.

- **CRL:** Common Language Runtime. Motor de ejecución común que sirve de traductor entre el MSIL (Microsoft Intermediate Language) y el código máquina nativo.
- **ADO.NET:** Nueva interfaz de bases de datos. No se trata de una evolución de ADO (ActiveX Data Objects), sino que se trata de una interfaz completamente nueva.
- **ASP.NET:** Nueva tecnología para páginas web dinámicas completamente integrada dentro del entorno .NET. Representa una autentica revolución en el desarrollo Web (Internet e Intranet).
- **SDK:** Software Development Kit. conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema bastante concreto.
- **HTML:** HyperText Markup Language. Lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.
- **CSS:** Lenguaje de hojas de estilo en cascada (cascade style sheets) utilizadas para definir la presentación de un documento HTML o XML.
- **XML:** eXtensible Markup Language. Lenguaje de marcado para la edición de ficheros con una estructura.
- **XSL:** Extensible Stylesheet Language. Es una familia de lenguajes basados en el estándar XML que permite describir cómo la información contenida en un documento XML cualquiera debe ser transformada o formateada para su presentación en un medio específico.
- **XSLT:** XSL Transformaciones. Es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML.
- **XLL:** eXtensible Linking Language. Abarca un conjunto de tecnologías relacionadas con los enlaces y el direccionamiento en lenguaje XML.
- **XUA:** (XML User Agent): Es la estandarización de navegadores XML. Todavía está en proceso de creación. Se aplicará a los navegadores para que compartan todas las especificaciones XML.
- **XPath:** XML Path Language. Es un lenguaje para direccionar partes de un documento XML.
- **XDR:** eXternal Data Representation.
- **XSD:** XML Schema Definition. Un XSD se utiliza para la definición de un tipo de documentos o datos que sirven en su mayoría para especificar y ofrecer el contenido de cualquier documento.
- **RGB:** Reed Green Blue. Sistema de síntesis aditiva basado en rojo, verde y azul.

## APÉNDICE D. Manual de usuario del comunicador

### D.1 Introducción

En este manual de usuario se explicará el uso de la aplicación con la configuración dada por el desarrollador del sistema. Se debe tener en cuenta que la funcionalidad del sistema puede variar cuando se desee.

La aplicación busca facilitar la comunicación de las personas con parálisis cerebral mediante la utilización de distintos métodos de escritura. Los métodos de escritura existentes son dos:

- Escritura mediante la inserción de caracteres.
- Escritura basada en el alfabeto SPC (Sistema Pictográfico de Comunicación).

La diferencia entre ambos métodos es que en el primero las diferentes pantallas de la aplicación estarán formadas por botones con caracteres escritos. Este método permitirá ir escribiendo carácter a carácter. En el segundo de los métodos, las diferentes pantallas estarán formadas por botones de imagen. Este método permitirá escribir expresiones asociadas a los botones de imagen.

En los siguientes apartados se explicará de manera detallada y gráfica cómo se pueden utilizar los distintos métodos de escritura de la aplicación para comunicarse con el entorno.

### D.2 Bienvenida



Figura D.1 Pantalla de bienvenida

Al iniciarse la aplicación aparecerá esta pantalla de bienvenida (Figura D.1). Contiene el título del proyecto y el nombre del autor del mismo. Los colores de esta pantalla no pueden ser modificados.

Mientras se muestra esta pantalla, la aplicación estará cargando la configuración determinada. Es decir, este es el momento en el que la aplicación “conocerá” el método de escritura que va a utilizar:

- Basado en caracteres
- Basado en imágenes

Tras esta pantalla aparecerá la pantalla principal que corresponde a la configuración determinada.

### D.3 Funcionamiento del barrido

El barrido es un método que se ha empleado en el desarrollo del sistema que, cuando está activado, hace que se señalen (a intervalo de tiempos iguales y configurables por el usuario) todos los botones que aparecen en las distintas pantallas de la aplicación. La finalidad es que el usuario no esté obligado a pulsar el botón que desee utilizando para ello el lápiz de la PDA, sino que mediante un simple golpe, con cualquier parte de su cuerpo, en un pulsador conectado a la PDA, la aplicación seleccione el botón señalado por el barrido. Este método de escritura está especialmente diseñado para su uso por personas con discapacidades motoras severas.

La manera de mostrar al usuario el botón que está señalado por el barrido es distinta en función de la configuración con la que se esté trabajando:

- Si la configuración es de teclado formado por botones de caracteres (configuraciones de CONJUNTOS o de CATEGORIAS) el botón señalado por el barrido será de color negro y con el texto blanco (Figura D.2).
- Si la configuración es de teclado formado por botones de imágenes (configuración de IMAGENES) habrá dos opciones:
  - El botón es de selección de categorías de imágenes: Será del color representativo de la categoría (según el alfabeto SPC), pero el borde del botón será de color morado. (Figura D.3).
  - El botón no es de selección de categorías de imágenes (botón de inserción de texto, botón de menú...): Será de color morado y el color de la imagen no cambiará (en este caso seguirá siendo negro). (Figura D.4).



Figura D.2 Pantalla de configuración de CONJUNTOS con un botón señalado por el barrido.



Figura D.3 Pantalla de configuración de IMAGENES con un botón de selección de categoría, señalado por el barrido.



a) Botón de inserción de texto señalado por el barrido.



b) Botón de vuelta atrás señalado por el barrido.

Figura D.4 Pantallas de configuración de IMAGENES con un botón, que no es de selección de categoría, señalado por el barrido

Este método de escritura es un método lento, ya que el usuario está la mayor parte del tiempo esperando a que se señale el botón que desea.

La única habilidad que debe tener el usuario para poder comunicarse utilizando este método consiste en poder golpear el pulsador con cualquier parte de su cuerpo con la que tenga movilidad (barbilla, codo, cabeza...).

Aunque el barrido esté activado, se podrá utilizar indistintamente el lápiz de la PDA, es decir, si una persona del entorno del usuario habitual de la aplicación (profesor, familiar, tutor...) quiere ayudar al mismo o quiere configurar la aplicación, no necesitará desactivar el barrido para aumentar la velocidad de sus acciones; ya que, si desea pulsar un botón que no esté señalado por el barrido, lo podrá pulsar ayudándose del lápiz y funcionará correctamente.

**Ejemplo D.1:** Supóngase que se desea acceder a una pantalla cuando el barrido está señalando un botón que mediante su pulsación mostrará otra pantalla distinta a la deseada y que la persona que lo desea hacer tiene la capacidad para ello, es decir, no tiene una discapacidad que le evite pulsar un botón con el lápiz de la PDA.

En la Figura a) del ejemplo se muestra una pantalla en la que el barrido está señalando el botón ABCDEF y con un círculo rojo se señala el botón que, supóngase, el usuario desea pulsar. Si el usuario pulsa con el lápiz el botón 0-9, la aplicación mostrará la pantalla de números, que es la que corresponde a dicho botón (Figura b) del ejemplo).



a) Pantalla antes de la pulsación.



b) Pantalla después de la pulsación.

Ejemplo D.1: Acceso a una pantalla utilizando el lápiz de la PDA.

En las configuraciones de escritura mediante inserción de caracteres, se puede trabajar con o sin barrido y esta decisión la podrá tomar el usuario configurando la aplicación. Sin embargo, no está permitido trabajar sin barrido en la configuración de inserción de texto mediante imágenes. Esto es así porque, generalmente, para una determinada categoría existen más imágenes a mostrar que las que caben en una pantalla de la PDA y la aplicación utiliza la posición de barrido para saber cuando debe modificar las imágenes de los botones que está mostrando. Este método se explicará más detalladamente en el apartado D.6 *Escritura mediante imágenes*.

## D.4 Escritura mediante inserción de caracteres

### D.4.1 Agrupación de caracteres por conjuntos



Figura D.5 Pantalla principal de CONJUNTOS

Esta pantalla será la pantalla principal cuando se esté ejecutando la aplicación con la configuración CONJUNTOS.

En la parte superior de la pantalla se muestra un cuadro de texto (vacío de momento), a continuación dos filas y tres columnas de botones con caracteres escritos y una fila de botones en la parte inferior con las palabras escritas BORRAR, ESPACIO Y MENU.

Los botones de las dos filas superiores son los que están dedicados a la inserción de texto en el cuadro de texto. Los botones de la fila inferior tienen las funcionalidades siguientes:

- BORRAR: Borra el último carácter escrito.
- ESPACIO: Inserta un espacio en el cuadro de texto.
- MENU: Muestra la pantalla de menú.

Cuando la aplicación esté mostrando esta pantalla y se desee añadir un carácter, se debe pulsar el botón que contiene el carácter a escribir. De esta manera la aplicación mostrará una pantalla formada por el cuadro de texto y un nuevo conjunto de botones. Cada botón estará representado por un carácter de tal forma que al pulsar el botón que contiene el carácter deseado, éste se añadirá al cuadro de texto.

Una vez añadido el carácter, la aplicación puede volver a la pantalla principal (Figura D.5) o quedarse en la misma pantalla. Esta distinción depende de la manera en que se han creado los ficheros de configuración. Con la finalidad de agilizar la escritura, en este desarrollo, se han creado los botones de inserción de caracteres, de todas las configuraciones, con la funcionalidad de volver a la pantalla inicial después de insertar el carácter, pero se podría cambiar sin ningún esfuerzo por parte del desarrollador.

**Ejemplo D.2:** Supóngase, por ejemplo, que se desee añadir la letra “A” al cuadro de texto. Para ello se deberá pulsar el botón representado por los caracteres “ABCDEF” (Figura D.5). A continuación se mostrará la pantalla representada en la Figura a) del ejemplo. Después, se deberá pulsar el botón representado por el carácter “A” (Representado por un círculo rojo en la figura). La aplicación mostrará la pantalla inicial, pero con la letra “A” escrita en el cuadro de texto (Figura b) del ejemplo).



a) Pantalla ABCDEF



b) Pantalla inicial con texto escrito

Ejemplo D.2: Escritura de un carácter con la configuración de COJUNTOS

## D.4.2 Agrupación de caracteres por categorías

Esta pantalla será la pantalla principal cuando se esté ejecutando la aplicación con la configuración CATEGORIAS.

En la parte superior de la pantalla se muestra un cuadro de texto (vacío de momento), a continuación tres filas con un botón por fila representando una categoría y una fila de botones en la parte inferior con las palabras escritas BORRAR, ESPACIO Y MENU.

Los tres botones de categorías son los que están dedicados a la inserción de texto en el cuadro de texto. Los botones de la fila inferior tienen las mismas funcionalidades que se describieron anteriormente.



Figura D.6 Pantalla principal de CATEGORIAS

Cuando la aplicación esté mostrando esta pantalla y se desee añadir un carácter, se debe pulsar el botón que represente la categoría del carácter a escribir. De esta manera la aplicación mostrará una pantalla formada por el cuadro de texto y un conjunto de botones. Dependiendo de la categoría elegida se tiene lo siguiente:



Figura D.7 Pantalla VOCALES

- Si la categoría elegida es la de VOCALES (Figura D.7), cada botón de la nueva pantalla está representado por un carácter de tal forma que al pulsar el botón que contiene el carácter deseado, éste se añadirá al cuadro de texto. En esta pantalla existe también un botón representado por los caracteres "SIMB" que al ser pulsado hará que se muestre una pantalla para poder escribir símbolos (Figura D.10).

- Si la categoría elegida es la de CONSONANTES, cada botón estará representado por un conjunto de consonantes (Figura D.8). Al pulsar uno de estos botones se mostrará una nueva pantalla donde cada botón está representado por una única consonante. En esta nueva pantalla ya se puede escribir una consonante mediante la selección del botón que la represente.



Figura D.8 Pantalla CONSONANTES



Figura D.9: Pantalla NUMEROS

- Si la categoría elegida es la de NUMEROS, se mostrará una pantalla formada por botones que representarán, cada uno de ellos, a un número (Figura D.9). Mediante la pulsación del botón correspondiente se produce la escritura. Además, se podrá acceder a otra pantalla de inserción de símbolos matemáticos (Figura D.11). Los botones de esta pantalla de escritura de números están definidos con la funcionalidad de escribir y no volver a la pantalla principal, esto es así debido a la gran probabilidad que hay de escribir varios números seguidos.

**Ejemplo D.3:** Supóngase el mismo ejemplo que antes. Se desea añadir la letra “A” al cuadro de texto. Para ello se deberá pulsar el botón que representa la categoría VOCALES (Figura D.6). A continuación se mostrará la pantalla de vocales representada en la Figura D.7. Después, se deberá pulsar el botón representado por el carácter “A”. La aplicación mostrará la pantalla inicial, pero con la letra “A” escrita en el cuadro de texto (Figura a) del ejemplo).



a) Pantalla inicial con texto escrito

Ejemplo D.3: Escritura de un carácter con la configuración de CATEGORIAS



Figura D.10 Pantalla de símbolos



Figura D.11 Pantalla de símbolos matemáticos

## D.5 Borrar carácter

Para borrar un carácter simplemente se debe pulsar el botón con el texto asociado BORRAR que aparecerá en las pantallas principales y en las pantallas de números para las configuraciones de CONJUNTOS Y CATEGORIAS.

**Ejemplo D.4:** Supóngase que se tiene un texto escrito en el cuadro de texto (Figura a) del ejemplo) y se desea borrar el último carácter. Para ello se debe pulsar el botón BORRAR.



a) Pantalla principal de la configuración CONJUNTOS con texto escrito.



b) Pantalla principal la configuración CONJUNTOS con último carácter borrado.

Ejemplo D.4: Borrado de un carácter

## D.6 Escritura mediante imágenes



Figura D.12 Pantalla inicial en la configuración de IMÁGENES

En la figura D.12 se muestra la pantalla inicial cuando la aplicación se está ejecutando bajo la configuración de IMÁGENES.

La pantalla inicial, en esta configuración, está formada, como antes, por un cuadro de texto, por dos filas de botones que permiten elegir la categoría de la expresión a insertar y por una fila de dos botones de imagen en la parte inferior.

Cada botón de selección de categoría se representa con un color representativo de la misma según el alfabeto SPC. Además, los botones llevan escrito el nombre de la categoría.

Los botones de la fila inferior tienen las siguientes funcionalidades:

- La flecha ovalada hacia la izquierda es el botón que permite mostrar la pantalla de menú.
- El aspa roja permite borrar una palabra.

Para escribir una expresión es necesario pulsar el botón que representa la categoría a la que pertenece dicha expresión (Personas, verbos, nombres, etc.). Después, la aplicación mostrará una nueva pantalla formada por un cuadro de texto situado en la parte superior, por un número de botones (cuatro según la configuración de IMÁGENES) que representan, mediante imágenes SPC, las expresiones que se pueden escribir y por dos botones, situados en la parte inferior, que tendrán las siguientes funcionalidades:

- Volver a la pantalla anterior (La flecha azul)
- Borrar la última palabra escrita en el cuadro de texto (el aspa roja).



Figura D.13 Primera pantalla de inserción de expresiones de la categoría Personas

**Ejemplo D.5:** Supóngase que se desea escribir la palabra AGUA. Para ello será necesario pulsar el botón Nombres de la página principal (Figura D.12). La aplicación mostrará la primera pantalla de la categoría Nombres (Figura a) del ejemplo). Si la palabra AGUA está representada en esta pantalla, pulsaremos el botón correspondiente y la aplicación mostrará, de nuevo, la pantalla inicial, pero con el texto ya añadido (Figura b) del ejemplo).



a) Primera pantalla en la categoría Nombres.

b) Pantalla principal con texto añadido.

Ejemplo D.5: Inserción de una expresión que aparece en la primera pantalla de la categoría.

Los botones de inserción de expresiones serán botones variables, es decir, sus imágenes y sus expresiones asociadas variarán durante la ejecución de la aplicación. Esto es debido a que, dentro de una categoría, existen más imágenes que botones dentro una pantalla. Para que la aplicación conozca el momento en el que debe cambiar las imágenes de los botones se utiliza el barrido, así cuando el barrido está seleccionando el último botón de la pantalla (botón borrar palabra), la aplicación cambia las imágenes y las expresiones asociadas de los botones variables.

**Ejemplo D.6:** Supóngase el ejemplo anterior, pero con la diferencia de que la expresión que se quiere escribir no aparece en la primera pantalla de la categoría. Por ejemplo la palabra BUENO. Para comenzar se debe pulsar el botón ADJETIVOS Y ADVERBIOS de la pantalla principal (figura D.12). A continuación se muestran las pantallas que aparecerán y los botones que va señalando el barrido:



a) Primera pantalla de la categoría



b) El barrido selecciona el siguiente botón.



c) El barrido selecciona el siguiente botón.



d) El barrido selecciona el siguiente botón.



e) El barrido selecciona el siguiente botón.



f) Selecciona el último botón y cambio de imágenes.



g) El barrido selecciona el botón de se desea.



h) Pantalla principal con el texto añadido.

Este ejemplo muestra de manera gráfica lo lenta que se puede hacer la escritura de una expresión a medida que ésta pertenece a una pantalla más profunda dentro de una categoría.

## D.7 Borrar palabra

La forma de borrar texto es ligeramente diferente cuando se trabaja con la configuración de IMAGENES a la forma de hacerlo trabajando con otra configuración. Esta diferencia no es sólo por la distinta apariencia del botón de borrado, sino por la distinta funcionalidad de ambos; ya que, como se comentó en el apartado D.5 *Borrar carácter*, en las configuraciones de escritura mediante inserción de caracteres cada pulsación del botón BORRAR elimina el último carácter escrito en el cuadro de texto, sin embargo, trabajando con la configuración de IMAGENES cada pulsación del botón BORRAR provoca la eliminación de la última palabra que esté escrita en el cuadro de texto.

**Ejemplo D.7:** Supóngase que se desea borrar la palabra PAPA del cuadro de texto. Para ello se deberá pulsar el botón de borrado (Rodeado por una elipse roja en el ejemplo).



a) Pantalla antes del borrado

b) Pantalla después del borrado

Ejemplo D.7 Borrado de una palabra

Además de realizar el borrado desde la pantalla principal, como en el ejemplo D.7, es posible hacerlo desde cualquiera de las pantallas de inserción de expresiones, ya que todas ellas poseen el botón de borrado.

## D.8 Movimientos entre pantallas

En este apartado se va a explicar cómo el usuario puede moverse entre las distintas pantallas de la aplicación. Existen distintos motivos por los que la aplicación puede cambiar la pantalla que está mostrando:

1. Pulsación de un botón cuya funcionalidad es la de ir a otra pantalla
2. Pulsación de un botón cuya funcionalidad es la de realizar una función y después ir a otra pantalla.
3. Cambio de pantalla provocado por el barrido.

1. Pulsación de un botón cuya funcionalidad es la de ir a otra pantalla: aquí están englobados los cambios de pantalla que provoca el usuario conscientemente, es decir, el usuario desea ver una determinada pantalla y pulsa el botón correspondiente para ello.

Para realizar estos movimientos existen muchos botones, como son por ejemplo los de selección de una categoría (CONSONANTES, VOCALES, PERSONAS, VERBOS...), los de selección de un conjunto de caracteres (ABCDEF, SIMBOLOS...) y, por supuesto, los botones de VOLVER. En todas las pantallas (salvo en la principal) de todas las configuraciones aparece un botón VOLVER que permite volver hacia atrás. En las configuraciones de escritura mediante inserción de texto este botón esta representado por la propia palabra VOLVER (Ver Ejemplo D.8), en la configuración de imágenes existen dos tipos de representación para este botón:

- a) En las pantallas de escritura aparece un botón con una flecha azul ovalada y apuntando hacia la izquierda (hacia unas barras de colores)



Figura D.14 Botón Volver

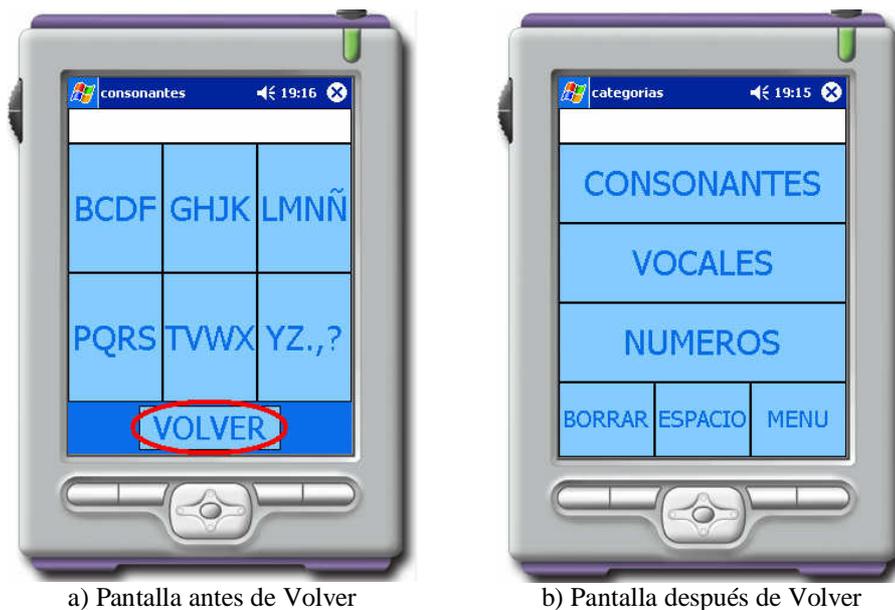
Si la pantalla actual no es la primera dentro de una categoría de imágenes, la pulsación de este botón hará que se muestre la pantalla de imágenes anterior. Si la pantalla actual es la primera, se mostrará la pantalla principal (Ver Ejemplo D.9).

- b) En la pantalla de menú aparece un botón con una flecha azul ovalada y apuntando hacia la izquierda también, pero no tiene las barras de colores. Su pulsación provocará ir a la pantalla principal.



Figura D.15 Botón Volver

**Ejemplo D.8:** Volver hacia atrás en una configuración de escritura mediante inserción de caracteres:



Ejemplo D.8: Ejemplo de Volver

**Ejemplo D.9:** Volver hacia atrás en una pantalla de escritura de la configuración de IMAGENES:



Ejemplo D.9: Ejemplo de Volver

2. Pulsación de un botón cuya funcionalidad es la de realizar una función y después volver a la pantalla principal: En este apartado se engloban los botones que realizan una función conocida por el usuario (escribir, reproducir un texto escrito, fijar un determinado color para un elemento de la pantalla...) y cuando la terminan se muestra la pantalla que corresponda.

Según se ha creado la aplicación, los botones que tienen esta funcionalidad añadida son:

- Los botones de escritura de un carácter (que no sea un número).
- Los botones de escritura de una expresión asociada a una imagen.
- Los botones PLAY (de reproducción por voz del texto escrito).
- Los botones de selección de un color.

Como ya se ha explicado en alguna ocasión, si el usuario prefiriera cambiar quitar estas funcionalidades añadidas simplemente debería comunicárselo al desarrollador y en un breve período de tiempo (prácticamente en el instante) obtendría la aplicación con sus nuevas funcionalidades. La elevada velocidad con la que se soluciona este problema es debida a la gran capacidad de configuración que tiene la aplicación.

3. Cambio de pantalla provocado por el barrido: Este punto hace referencia a los cambios que se producen en las imágenes de los botones y en el texto asociado a las mismas cuando se trabaja con la configuración de IMAGENES. Este aspecto ya se comentó en el apartado D.6 *Escritura mediante imágenes* y en el mismo apartado se representó mediante el Ejemplo D.6.

## D.9 Menú

En todas las configuraciones de la aplicación existe una pantalla que permite realizar ciertas funciones que no están íntimamente relacionadas con la escritura. Las funciones que se pueden realizar a través de la pantalla MENU son las siguientes:

- Reproducir por voz el texto escrito en el cuadro de texto.
- Cambiar la configuración de la aplicación.
- Cambiar los colores de apariencia de la aplicación (si se trabaja con la configuración de IMAGENES esta opción no aparecerá)
- Consultar la ayuda que está integrada en la aplicación.
- Salir de la aplicación.

A continuación (Figura D.16) se muestran las dos posibles apariencias de la pantalla MENU en función de la configuración con la que se esté trabajando.



a) Menú para las configuraciones de CONJUNTOS Y CATEGORIAS

b) Menú para la configuración de IMAGENES

Figura D.16 Pantallas de MENÚ

La forma de llegar a la pantalla MENÚ consiste en la pulsación de un botón situado en la página principal de cada configuración. En las configuraciones de escritura mediante inserción de caracteres (CONJUNTOS Y CATEGORIAS) este botón está representado por la palabra MENU, mientras que en la configuración de IMAGENES este botón esta representado por una flecha azul, ovalada y apuntando hacia la izquierda (Figura D.17).



Figura D.17 Botón MENU en la configuración de IMAGENES

## D. 10 Reproducción por voz

Con la finalidad de mejorar la comunicación del usuario de la aplicación con su entorno, se ha incluido en el sistema un reproductor de voz. Para reproducir el texto que esté escrito en el cuadro de texto bastará con pulsar el botón PLAY situado en la pantalla de MENU (Figura D.16). Una vez hecho esto, la aplicación mostrará la pantalla principal con el cuadro de texto en blanco.

Si por un error en la escritura el reproductor de voz no puede reproducir un texto determinado, la aplicación mostrará un mensaje de error avisando de tal problema (Figura D.18). Cuando el usuario cierre el mensaje de error, la aplicación mostrará la pantalla principal con el cuadro de texto vacío para que el usuario reescriba la expresión que quiere comunicar.

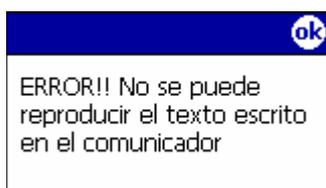


Figura D.18 Mensaje reproducción por voz errónea.

## D.11 Configuración de la aplicación

Como se viene comentando a lo largo de este manual, la aplicación puede funcionar bajo distintas configuraciones (las configuraciones de CONJUNTOS, CATEGORIAS E IMÁGENES están incluidas, pero dada la configurabilidad de la aplicación se podrían añadir más). La elección sobre la configuración a utilizar la debe tomar una persona del entorno del usuario (profesor, tutor, familiar...) en función de la discapacidad del mismo.

Para elegir una u otra configuración, se debe pulsar el botón CONFIGURACION de la pantalla MENU. A continuación y en función de la configuración con la que se esté trabajando en este momento se mostrará una de las siguientes pantallas (Figura D.19).



a) Configuración en CONJUNTOS Y CATEGORIAS

b) Configuración en IMAGENES

Figura D.19 Pantallas CONFIGURACIÓN

Una particularidad de esta pantalla y de todas las que están dedicadas a configurar la aplicación (por ejemplo: cambio de colores) es que en ellas no funciona el barrido, esto es debido a que la persona que configurará la aplicación será una persona sin discapacidad que, por lo tanto, puede utilizar el lápiz de la PDA para moverse por la aplicación.

Como se puede en ver la Figura D.19 la diferencia entre ambas pantallas es que en la pantalla de CONFIGURACIÓN correspondiente a la configuración de IMAGENES no aparece la posibilidad de activar o desactivar el barrido, esto es debido a la obligación existente, ya comentada, de trabajar en la configuración de IMÁGENES con el barrido activado.

A excepción de lo comentado anteriormente, ambas pantallas son similares. Debajo del texto “Tiempo de Barrido” aparece una lista donde se puede seleccionar el intervalo de tiempo que un botón estará señalado por el barrido. Los números representan el número de segundos que el barrido está señalando a un botón. Debajo del texto “Diseño de Pantalla” aparece otra lista donde se puede seleccionar la configuración que se desee (CONJUNTOS, CATEGORIAS o IMAGENES).

Una vez se hayan seleccionado la nueva configuración, se debe pulsar el botón ACEPTAR. A continuación la aplicación mostrará un mensaje pidiendo al usuario que reinicie la aplicación para que los cambios de configuración tengan efecto (Figura D.20).

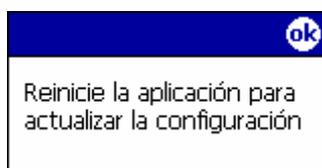


Figura D.20 Aviso de reinicio

En el momento en que el usuario cierre el mensaje, la aplicación se cerrará automáticamente. Cuando el usuario abra de nuevo la aplicación, ésta conservará la configuración guardada la última vez que se utilizó.

### **D.12 Colores de apariencia:**

La aplicación, cuando está funcionando bajo las configuraciones de CONJUNTOS o de CATEGORIAS, da la posibilidad al usuario de modificar los colores apariencia de los elementos de la pantalla. Cuando la aplicación funciona con la configuración de IMAGENES no está permitido el cambio de los colores porque el propio color de los botones es representativo de la funcionalidad asociada al mismo (Botón amarillo: Categoría de Personas...).

Cuando el usuario desee modificar el color de algún elemento de la pantalla deberá acceder al MENU y pulsar el botón COLORES (Figura D.16 a). Después de esto la aplicación mostrará la pantalla que aparece en la Figura D.21.



Figura D.21 Pantalla COLORES

En la pantalla representada en esta figura, el usuario debe seleccionar el tipo de elemento de pantalla al que desea cambiar el color. Para ello debe pulsar el botón CAMBIAR que esté situado al lado del correspondiente elemento.

Los posibles cambios de color son los siguientes:

- Color de fondo de las pantallas.
- Color de los botones en las configuraciones CONJUNTOS Y CATEGORIAS.
- Color del texto de los botones de las anteriores configuraciones.
- Color del texto informativo escrito en las pantallas.

Una vez seleccionado el elemento, la aplicación mostrará la pantalla representada en la Figura D.22. Aquí el usuario elegirá, mediante la pulsación del botón correspondiente, el color que desee para el elemento de pantalla elegido anteriormente (en la pantalla de la Figura D.21).

Al pulsar sobre el botón, la aplicación mostrará la pantalla representada en la Figura D.21 (para dar la posibilidad de realizar más cambios de color), ya con los cambios de color que se hayan realizado.

Los cambios de color realizados en una sesión de trabajo permanecerán al cerrar la aplicación, es decir, si el usuario cambia el color de apariencia de algún elemento, al reiniciar la aplicación (en cualquiera de las configuraciones salvo en la de IMAGENES) ese elemento tendrá el nuevo color.



Figura D.22 Pantalla de selección de color

**Ejemplo D.10:** En este ejemplo se muestra un posible cambio de color de apariencia. Supóngase que se está trabajando con la configuración de CONJUNTOS. El usuario para realizar un cambio de color accede al MENU (Figura D.16 a) y después pulsa el botón COLORES, accediendo a la pantalla de selección del elemento a modificar (Figura a) del ejemplo). Supóngase, que el usuario desea cambiar el fondo de la pantalla, para ello deberá pulsar el botón CAMBIAR que está situado junto el texto “Color Fondos”. Una vez pulsado, aparecerá la pantalla de selección del color (Figura b) del ejemplo). Supóngase, que el usuario desea que el fondo de sus pantallas sea amarillo, entonces pulsará sobre el color y la aplicación mostrará la pantalla representada en la Figura c) del ejemplo.



Ejemplo D.10: Cambio de color del fondo de la pantalla.

### D.13 Consulta de ayuda

La aplicación lleva incorporado un pequeño resumen de este manual que explica de manera muy breve cómo se pueden realizar todas las funciones que se ofrecen.



Figura D.23 Temas de ayuda

Para consultar la ayuda se debe acceder al MENU y pulsar el botón AYUDA (Figura D.16). Una vez hecho esto, se mostrará la pantalla representada en la Figura D.23. Aquí se deberá elegir el tema sobre el que se quiere consultar la ayuda mediante la pulsación del botón correspondiente. Los temas posibles de ayuda son:

- Reproducir
- Configuración
- Escribir
- Borrar
- Salir

Al elegir uno de estos temas, se mostrará una pantalla compuesta por un título, un cuadro de texto en el que se explica de manera muy breve cómo se puede realizar la acción determinada y un botón de VOLVER (Figura D.24).

Si el tema elegido es CONFIGURACIÓN, en vez de mostrarse una pantalla de ayuda, se mostrará una nueva pantalla de elección de temas (Figura D.25) compuesta por los siguientes temas:

- Tipos: Ayuda sobre la manera de elegir otra configuración distinta de la actual.
- Colores: Ayuda sobre cómo cambiar los colores de apariencia de la pantalla
- Barrido: Ayuda sobre cómo activar o desactivar el barrido y sobre cómo se puede cambiar el tiempo de barrido.



Figura D.24 Ayuda Borrar en una configuración de escritura mediante inserción de caracteres



Figura D.25 Elección de temas para la ayuda de CONFIGURACIÓN

Las pantallas de ayuda son similares en todas las configuraciones (sólo cambian pequeños matices en las explicaciones), es decir, no existen pantallas de ayuda en las que sus botones estén formados por imágenes. Esto es así porque se ha supuesto que la ayuda será consultada por una persona del entorno del usuario final como puede ser un familiar, un profesor, etc.

### D.14 Salir de la aplicación

Una vez terminada una sesión de trabajo el usuario podrá cerrar la aplicación pulsando el botón SALIR, que se encuentra situado en la pantalla MENU (Figura D.16). El botón SALIR cuando se está trabajando con la configuración de IMÁGENES está representado por una flecha azul y recta apuntando hacia la derecha (Figura D.26).



Figura D.26 Botón salir en la configuración IMAGENES

## Apéndice E. Hardware y software utilizado

En este apéndice se comentará brevemente el hardware y el software empleado para el desarrollo de este proyecto.

Los recursos hardware utilizados han sido los siguientes:

- Equipo de desarrollo:
  - Ordenador portátil Intel Pentium M 730.
  - 512 MB de memoria RAM.
  - 80 GB HDD.
  - lector/grabador de DVD.
  - Sistema operativo Windows XP Professional.
  
- PDA para realizar algunas comprobaciones:
  - Acer N-30.
  - Procesador Intel Xcale 266 MHz.
  - 64 MB de memoria RAM.
  - 32 MB de memoria ROM.
  - Sistema operativo Windows Mobile 2003.

Los recursos software utilizados han sido los siguientes:

- Visual Studio.NET 2003
- StarUML.
- Microsoft ActiveSync.
- DB Designer 4.
- Microsoft Office Word 2003.
- Nero Burning Rom 6.
- PDF redirect.



## APÉNDICE F. Contenido del CD

A continuación se enumeran los contenidos del CD que acompaña a esta memoria:

➤ Directorio de instalación:

- Subdirectorio de instalación en PDA: Contiene los archivos CAB necesarios para instalar el comunicador en el dispositivo. Para instalarlo directamente en una PDA Pocket PC hay que copiar estos archivos al dispositivo y ejecutarlos uno a uno desde allí. Para ello se puede emplear el explorador de ficheros de la propia PDA.

➤ Directorio de memoria: Contiene la memoria en formato PDF.

➤ Directorio de manual de usuario: Contiene el manual de usuario del comunicador en formato PDF.

➤ Directorio de código fuente: Contiene todos los archivos de código fuente generados para el desarrollo del componente y de la aplicación que lo utiliza. Aquí se incluyen también subdirectorios con los ficheros XML utilizados, las imágenes utilizadas y los sonidos utilizados por el componente que reproduce el texto escrito. Los ficheros de código fuente generados para el desarrollo del componente se encuentran en el subdirectorio *Ficheros Componente* y los utilizados para reutilizar el componente reproductor se encuentran en *Ficheros Reproductor*.



## Bibliografía

1. Grady Booch, James Rumbaugh, Ivan Jacobson, *El lenguaje unificado de desarrollo de software*. Ed. Addison Wesley
2. Grady Booch, James Rumbaugh, Ivan Jacobson, *El lenguaje unificado de modelado*. Ed. Addison Wesley, 2000.
3. John Chessman, John Daniels, *UML Components. A Simple Process for Specifying Component-Based Software*. Ed. Addison-Wesley, 2000 (1ª edición)
4. Larman, C. *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Ed. Prentice Hall , 2003 (2ª edición)
5. Presman, R. S. *Ingeniería del Software. Un enfoque práctico*. Ed. McGraw Hill, 2005 (4ª edición)
6. Sommerville. *Ingeniería de Software*, Ed. Addison-Wesley, 2005, (7ª edición)

## Referencias Web

- Microsoft [www.microsoft.com](http://www.microsoft.com)
- MSDN [www.msdn.microsoft.com/library](http://www.msdn.microsoft.com/library)
- Desarrolla con MSDN [www.desarrollaconmsdn.com](http://www.desarrollaconmsdn.com)
- [www.lawebdelprogramador.com](http://www.lawebdelprogramador.com)
- El guille [www.elguille.info](http://www.elguille.info)
- El Gurú del programador [www.elgurudelprogramador.com](http://www.elgurudelprogramador.com)
- SymbolWorld.org [www.symbolworld.org](http://www.symbolworld.org)
- [www.healthsystem.virginia.edu](http://www.healthsystem.virginia.edu)