



**Universidad de Valladolid**

**E. T. S. DE INGENIERÍA INFORMÁTICA**

**Ingeniería Técnica en Informática de Gestión**

---

**Desarrollo de una línea de productos de comunicadores para personas discapacitadas**

---

***Alumnos: David Ortega López  
M<sup>a</sup> Elena Pastor Portela***

**Tutores: Miguel A. Laguna Serrano  
Bruno González-Baixauli**



## *Agradecimientos*

### *David*

A mi padre, mi madre, mis hermanos y mi madrina, que siempre me han guiado y dado impulso para realizar el proyecto, a pesar de las dificultades.

A mis amigos, que han comprendido el trabajo y dedicación que este trabajo ha implicado.

A M<sup>a</sup> Elena, que siempre me ha apoyado en los momentos difíciles, y ha sido la mejor compañera de proyecto que alguien puede tener.

### *M<sup>a</sup> Elena*

A mis padres y mi hermana porque me han animado, han valorado todo mi esfuerzo, se han interesado por el trabajo y alegrado por el resultado, y por toda su comprensión.

A mis amigos y a todos los que han comprendido la difícil situación que hemos pasado a la par de la realización del proyecto, y que han sabido entender el retraso de la presentación. Pues hay ciertas cosas que se necesita un tiempo para su asimilación y forzando, no se logra nada.

Por último a David, un gran compañero y persona que ha aprendido a no tener miedo en mis momentos de ira cuando algo no funcionaba y por abrirme los ojos en muchos aspectos.

*También queríamos dar las gracias a nuestros tutores, Miguel Ángel y Bruno, por todo el tiempo que nos han dedicado y por su orientación en el proyecto.*



# ÍNDICE

---



---

## **PARTE 1. INTRODUCCIÓN ..... 1**

<b>Capítulo 1: PRESENTACIÓN DEL PROYECTO .....</b>	<b>3</b>
1.1 Descripción y ámbito del proyecto .....	3
1.2 Alcance del proyecto.....	4
1.3 Sobre esta documentación.....	5
<b>Capítulo 2: OBJETIVOS PROPUESTOS .....</b>	<b>7</b>
2.1 Objetivos.....	7

---

## **PARTE 2. CONTEXTO DEL PROBLEMA ..... 9**

<b>Capítulo 3: INFORMÁTICA PARA PERSONAS DISCAPACITADAS .....</b>	<b>11</b>
3.1 Introducción.....	11
3.2 Comunicación .....	11
3.3 Problemas en la comunicación.....	12
3.4 Sistemas alternativos de comunicación.....	12
<b>Capítulo 4: LINEAS DE PRODUCTOS SOFTWARE .....</b>	<b>13</b>
4.1 Introducción.....	13
4.2 Terminología empleada.....	15
4.3 Objetivos de la línea de productos .....	16
4.4 Beneficios .....	17

---

## **PARTE 3. PLAN DE PROYECTO..... 19**

<b>Capítulo 5: PLANIFICACIÓN DEL PROYECTO .....</b>	<b>21</b>
5.1 Personal necesario.....	21
5.2 Recursos técnicos necesarios .....	21
5.3 Gestión de riesgos .....	22
5.3.1 Identificación de riesgos del proyecto.....	22
5.3.2 Incidencia de riesgos sobre el proyecto .....	23
5.3.3 Medidas para la reducción y gestión de riesgos .....	24
5.4 Estimaciones temporales.....	25
5.4.1 Fechas de comienzo y fin de proyecto.....	25
5.4.2 Fases de la elaboración del proyecto. Tareas.....	25
5.4.3 Estimación de tiempos.....	26
5.5 Seguimiento del proyecto.....	29
5.5.1 Riesgos aparecidos en el desarrollo del proyecto .....	29
5.5.2 Distribución real de tiempos.....	30
5.5.3 Planificación real del proyecto .....	30

5.6	Horas de trabajo.....	32
5.7	Evaluación de costes.....	32

**PARTE 4. ANÁLISIS Y DESARROLLO DE LA LÍNEA DE PRODUCTOS .....33**

**Capítulo 6: ANÁLISIS DE LA LÍNEA DE PRODUCTOS ..... 35**

6.1	Definición de la variabilidad de la línea de productos.....	35
6.1.1	Definición de la línea de productos .....	36
6.2	Análisis de requisitos. Modelo de características .....	38
6.3	Gestión de la variabilidad con el método de combinación de paquetes.....	45
6.4	Análisis de los casos de uso.....	47
6.4.1	Identificación de los actores .....	47
6.4.2	Diagrama de casos de uso.....	48
6.4.3	Estructura de paquetes de los casos de uso.....	50
6.4.4	Presentación de casos de uso .....	52
6.5	Diagrama inicial de clases.....	69
6.5.1	Estructura de paquetes inicial de las clases .....	70

**Capítulo 7: DISEÑO DE LA LÍNEA DE PRODUCTOS ..... 73**

7.1	Introducción.....	73
7.2	Especificación de casos de uso .....	74
7.3	Diagrama final de clases.....	92
7.4	Diagrama de clases empaquetadas.....	94
7.5	Especificación de las clases.....	99
7.6	Descripción de la interfaz gráfica de usuario del comunicador .....	110

**Capítulo 8: IMPLEMENTACIÓN DE LA LÍNEA DE PRODUCTOS ..... 113**

8.1	Introducción.....	113
8.2	Entorno de programación y lenguaje .....	113
8.2.1	Utilización de los ficheros XML .....	115
8.2.2	Tratamiento de los archivos XML.....	119
8.3	Instanciación manual de productos.....	121
8.4	Software empleado .....	128
8.5	Hardware empleado .....	129

**Capítulo 9: CONFIGURACIÓN DE PRODUCTOS ..... 131**

9.1	Introducción.....	131
9.2	Modelado de características con Eclipse y el plugin FMP .....	132
9.3	Configuración de productos basada en características.....	134

**Capítulo 10: PRUEBAS ..... 135**

10.1	Introducción.....	135
10.2	Pruebas realizadas.....	136
10.3	Análisis de casos límite .....	144

---

**PARTE 5. MANUAL DE USUARIO ..... 147**

**Capítulo 11: MANUAL DE USUARIO ..... 149**

- 11.1 Introducción ..... 149
- 11.2 Instalación de la aplicación ..... 150
- 11.3 Pantalla de bienvenida ..... 153
- 11.4 Funcionamiento del barrido ..... 153
- 11.5 Escritura mediante inserción de caracteres ..... 155
  - 11.5.1 Método de letras agrupadas ..... 155
  - 11.5.2 Método de categorías ..... 158
  - 11.5.3 Método de las vocales ..... 160
- 11.6 Borrar texto ..... 161
- 11.7 Escritura mediante imágenes ..... 163
- 11.8 Borrar una palabra por el método de las imágenes ..... 166
- 11.9 Movimientos entre pantallas ..... 167
- 11.10 Menú ..... 168
- 11.11 Reproducción por voz ..... 171
- 11.12 Configuración de la aplicación ..... 171
- 11.13 Cambiar colores de los elementos de la aplicación ..... 173
- 11.14 Empleo de frases ..... 175
- 11.15 Uso de la ayuda en la aplicación ..... 177
- 11.16 Salir de la aplicación ..... 179

---

**PARTE 6. CONCLUSIONES ..... 181**

**Capítulo 12: CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO ..... 183**

- 12.1 Conclusiones acerca de las Líneas de Productos ..... 183
- 12.2 Dificultades encontradas ..... 185
- 12.3 Objetivos alcanzados ..... 186
- 12.4 Conocimientos adquiridos ..... 186
- 12.5 Futuras líneas de trabajo ..... 187

---

**REFERENCIAS ..... 189**

---

**APÉNDICE A. TECNOLOGÍA UTILIZADA ..... 195**

**A.1 MICROSOFT .NET ..... 197**

- A.1.1 Introducción ..... 197
- A.1.2 .NET Framework ..... 197
- A.1.3 .NET Compact Framework ..... 198

A.1.4	Microsoft Visual Studio.NET .....	198
A.1.5	C# .....	199
<b>A.2</b>	<b>OBJETOS DATASET Y ARCHIVOS .XML .....</b>	<b>199</b>
A.2.1	Introducción .....	199
A.2.2	Objetos DATASET .....	199
A.2.3	XML .....	201
A.2.4	XSD .....	203

---

***APÉNDICE B. ESTRUCTURA DE LOS FICHEROS .XML .....205***

<b>B.1</b>	<b>CONSIDERACIONES PREVIAS .....</b>	<b>207</b>
B.1.1	Introducción .....	207
B.1.2	División y organización de los archivos XML de interfaz.....	208
<b>B.2</b>	<b>ESTRUCTURA DE LOS FICHEROS .XML .....</b>	<b>208</b>
B.2.1	Estructura general de los ficheros .XML .....	208
B.2.2	Información sobre las pantallas (paneles) .....	209
B.2.3	Información sobre las etiquetas.....	210
B.2.4	Información sobre los botones .....	211
	B.2.4.1 Botones de caracteres.....	212
	B.2.4.2 Botones de imágenes .....	216
B.2.5	Información sobre los botones de radio .....	218
B.2.6	Información sobre las listas de selección .....	219
B.2.7	Información sobre el fichero de configuración .....	220

---

***APÉNDICE C. CONTENIDO DEL CD.....223***

# ÍNDICE DE FIGURAS

---



---

## Capítulo 4

Figura 4.1: Evolución de la reutilización software .....	14
Figura 4.2: Esquema conceptual de la línea de productos .....	15
Figura 4.3: Terminología de las líneas de productos .....	16

---

## Capítulo 5

Figura 5.1: Porcentajes de tiempo estimado dedicado por tarea .....	27
Figura 5.2: Diagrama de Gantt del tiempo estimado .....	28
Figura 5.3: Porcentajes de tiempo real dedicado por tarea .....	30
Figura 5.4: Diagrama de Gantt del tiempo real empleado .....	31

---

## Capítulo 6

Figura 6.1: Diagrama de características .....	41
Figura 6.2: Detalle del diagrama de características. Funcionalidades .....	42
Figura 6.3: Detalle del diagrama de características. Métodos de escritura .....	44
Figura 6.4: Diagrama de casos de uso .....	49
Figura 6.5: Estructura de paquetes de los casos de uso .....	50
Figura 6.6: Ejemplo de combinación de paquetes .....	51
Figura 6.7: Diagrama de secuencia del caso de uso: Consultar ayuda .....	52
Figura 6.8: Diagrama de secuencia del caso de uso: Seleccionar colores .....	53
Figura 6.9: Diagrama de secuencia del caso de uso: Seleccionar .....	54
Figura 6.10: Diagrama de secuencia del caso de uso: Seleccionar barrido .....	55
Figura 6.11: Diagrama de secuencia del caso de uso: Seleccionar método de escritura .....	56
Figura 6.12: Diagrama de secuencia del caso de uso: Borrar palabra .....	57
Figura 6.13: Diagrama de secuencia del caso de uso: Insertar frase .....	58
Figura 6.14: Diagrama de secuencia del caso de uso: Guardar frase .....	59
Figura 6.15: Diagrama de secuencia del caso de uso: Borrar frase .....	60
Figura 6.16: Diagrama de secuencia del caso de uso: Reproducir .....	61
Figura 6.17: Diagrama de secuencia del caso de uso: Escribir texto .....	62
Figura 6.18: Diagrama de secuencia del caso de uso: Escribir texto Categorías .....	63
Figura 6.19: Diagrama de secuencia del caso de uso: Escribir texto Vocales .....	64
Figura 6.20: Diagrama de secuencia del caso de uso: Escribir texto Letras agrupadas .....	65
Figura 6.21: Diagrama de secuencia del caso de uso: Escribir método variable .....	66
Figura 6.22: Diagrama de secuencia del caso de uso: Escribir imágenes .....	67
Figura 6.23: Diagrama de secuencia del caso de uso: Salir de la aplicación .....	68
Figura 6.24: Diagrama inicial de clases .....	69
Figura 6.25: Diagrama inicial de clases empaquetado .....	70

---

## Capítulo 7

Figura 7.1: Diagrama de secuencia del caso de uso Consultar ayuda .....	74
Figura 7.2: Diagrama de secuencia del caso de uso Seleccionar colores .....	75
Figura 7.3: Diagrama de secuencia del caso de uso Seleccionar .....	77
Figura 7.4: Diagrama de secuencia del caso de uso Seleccionar barrido .....	78
Figura 7.5: Diagrama de secuencia del caso de uso Seleccionar método de escritura .....	79
Figura 7.6: Diagrama de secuencia del caso de uso Borrar palabra .....	80
Figura 7.7: Diagrama de secuencia del caso de uso Inserta frase .....	81

Figura 7.8: Diagrama de secuencia del caso de uso Guardar frase.....	82
Figura 7.9: Diagrama de secuencia del caso de uso Borrar frase.....	83
Figura 7.10: Diagrama de secuencia del caso de uso Reproducir.....	84
Figura 7.11: Diagrama de secuencia del caso de uso Escribir texto Categorías.....	86
Figura 7.12: Diagrama de secuencia del caso de uso Escribir texto Vocales.....	87
Figura 7.13: Diagrama de secuencia del caso de uso Escribir texto Letras agrupadas.....	88
Figura 7.14: Diagrama de secuencia del caso de uso Escribir método variable.....	89
Figura 7.15: Diagrama de secuencia del caso de uso Escribir imágenes.....	90
Figura 7.16: Diagrama de secuencia del caso de uso Salir de la aplicación.....	91
Figura 7.17: Diagrama final de clases.....	93
Figura 7.18: Diagrama final de clases empaquetadas.....	94
Figura 7.19: Detalle del diagrama de clases empaquetado: Caso base.....	95
Figura 7.20: Detalle del diagrama de clases empaquetado: Escritura.....	96
Figura 7.21: Detalle del diagrama de clases empaquetado: Funcionalidades.....	98
Figura 7.22: Esquema interfaz gráfica de usuario de la pantalla inicial.....	111
Figura 7.23: Esquema interfaz gráfica de usuario de una pantalla de un grupo.....	111
Figura 7.24: Esquema interfaz gráfica de usuario de la pantalla de menú.....	111

---

## Capítulo 8

Figura 8.1: Aspecto de la implementación de la línea de productos en Visual Studio .NET 2005.....	114
Figura 8.2: Esquema XSD de la interfaz gráfica del comunicador.....	120
Figura 8.3: Ejemplo de un fichero .csproj.....	123

---

## Capítulo 9

Figura 9.1: Árbol del modelo de características.....	133
Figura 9.2: Ejemplo de configuración de producto.....	133
Figura 9.3: Ejemplo de .xml generado mediante la configuración de un producto.....	134

---

## Capítulo 11

Figura 11.1. Creación de proyecto CAB.....	150
Figura 11.2. Adición de archivos al .cab.....	151
Figura 11.3. Selección de elementos a añadir.....	151
Figura 11.4. Construcción del proyecto CAB.....	152
Figura 11.5: Pantalla de bienvenida.....	153
Figura 11.6: Pantalla del método de letras agrupadas con un botón señalado por el barrido.....	154
Figura 11.7: Pantalla del método de imágenes con un botón señalado por el barrido.....	154
Figura 11.8: Pantalla del método de imágenes.....	154
Figura 11.9: Pantalla principal de letras agrupadas.....	155
Figura 11.10: Pantalla desplegada del botón 0-9.....	157
Figura 11.11: Pantalla del botón XYZ..?.....	157
Figura 11.12: Pantalla del botón SIMBOLOS.....	157
Figura 11.13: Pantalla principal de CATEGORIAS.....	158
Figura 11.14: Pantalla VOCALES.....	158
Figura 11.15: Pantalla CONSONANTES.....	159
Figura 11.16: Pantalla BCDF.....	159
Figura 11.17: Pantalla principal del método de las VOCALES.....	160
Figura 11.18: Pantalla inicial en la configuración de IMÁGENES.....	163
Figura 11.19: Primera pantalla de inserción de expresiones de la categoría Personas.....	163

Figura 11.20: Segunda pantalla de inserción de expresiones de la categoría Personas .....	164
Figura 11.21: Botón Volver .....	167
Figura 11.22: Botón Volver .....	167
Figura 11.23: Menú básico .....	168
Figura 11.24: Menú con colores .....	169
Figura 11.25: Menú con frases .....	169
Figura 11.26: Menú completo.....	170
Figura 11.27: Menú de imágenes.....	170
Figura 11.28: Botón MENU en la configuración de IMÁGENES .....	171
Figura 11.29: Mensaje reproducción por voz errónea .....	171
Figura 11.30: Pantallas CONFIGURACIÓN .....	172
Figura 11.31: Aviso de reinicio .....	173
Figura 11.32: pantalla COLORES .....	174
Figura 11.33: pantalla de selección de color.....	174
Figura 11.34: Pantalla frases.....	176
Figura 11.35: Temas de ayuda para el caso de disponer de un solo método de escritura .....	178
Figura 11.36: Temas de ayuda para el caso de disponer de más de un método de escritura.....	178
Figura 11.37: Ayuda sobre cómo configurar el barrido.....	178
Figura 11.38: Elección de temas para la ayuda de CONFIGURACION .....	179
Figura 11.39: Ayuda sobre cómo cambiar el método de escritura.....	179
Figura 11.40: Botón salir en la configuración Imágenes .....	179

---

## ***Capítulo 12***

Figura 12.1: Evolución de costes. Línea de productos frente a sistemas simples .....	184
--	-----

---

## ***Apéndice A***

Figura A.1: Modelo de objetos de un DataSet .....	199
Figura A.2: Ejemplo gráfico de esquema XSD.....	204



# **ÍNDICE DE TABLAS**

---



---

## Capítulo 5

Tabla 5.1: Identificación de riesgos del proyecto .....	22
Tabla 5.2: Niveles de riesgo .....	23
Tabla 5.3: Incidencia de los riesgos.....	23
Tabla 5.4: Tabla de riesgos del proyecto .....	24
Tabla 5.5: Medidas de reducción de probabilidad de riesgos.....	24
Tabla 5.6: Medidas de reducción de impacto de riesgos.....	25
Tabla 5.7: Desglose de tiempo estimado por tarea.....	27
Tabla 5.8: Desglose de tiempo real por tarea.....	30

---

## Capítulo 6

Tabla 6.1: Clasificación de las características según los proyectos de fin de carrera estudiados.....	39
Tabla 6.2: Descripción del actor del sistema .....	48
Tabla 6.3: Clasificación de los casos de uso, según su categoría.....	48
Tabla 6.4: Descripción del caso de uso: Consultar ayuda.....	52
Tabla 6.5: Descripción del caso de uso: Seleccionar colores.....	53
Tabla 6.6: Descripción del caso de uso: Seleccionar.....	54
Tabla 6.7: Descripción del caso de uso: Seleccionar barrido.....	55
Tabla 6.8: Descripción del caso de uso: Seleccionar método de escritura.....	56
Tabla 6.9: Descripción del caso de uso: Borrar palabra.....	57
Tabla 6.10: Descripción del caso de uso: Insertar frase.....	58
Tabla 6.11: Descripción del caso de uso: Guardar frase.....	59
Tabla 6.12: Descripción del caso de uso: Borrar frase.....	60
Tabla 6.13: Descripción del caso de uso: Reproducir.....	61
Tabla 6.14: Descripción del caso de uso: Escribir texto.....	62
Tabla 6.15: Descripción del caso de uso: Escribir texto Categorías.....	63
Tabla 6.16: Descripción del caso de uso: Escribir texto Vocales.....	64
Tabla 6.17: Descripción del caso de uso: Escribir texto Letras agrupadas.....	65
Tabla 6.18: Descripción del caso de uso: Escribir método variable.....	66
Tabla 6.19: Descripción del caso de uso: Escribir imágenes.....	67
Tabla 6.20: Descripción del caso de uso: Salir de la aplicación.....	68
Tabla 6.21: Descripción de las clases.....	71

---

## Capítulo 7

Tabla 7.1: Descripción del caso de uso: Consultar ayuda.....	74
Tabla 7.2: Descripción del caso de uso: Seleccionar colores.....	75
Tabla 7.3: Descripción del caso de uso: Seleccionar.....	77
Tabla 7.4: Descripción del caso de uso: Seleccionar barrido.....	78
Tabla 7.5: Descripción del caso de uso: Seleccionar método de escritura.....	79
Tabla 7.6: Descripción del caso de uso: Borrar palabra.....	80
Tabla 7.7: Descripción del caso de uso: Insertar frase.....	81
Tabla 7.8: Descripción del caso de uso: Guardar frase.....	82
Tabla 7.9: Descripción del caso de uso: Borrar frase.....	83
Tabla 7.10: Descripción del caso de uso: Reproducir.....	84
Tabla 7.11: Descripción del caso de uso: Escribir texto.....	85
Tabla 7.12: Descripción del caso de uso: Escribir texto Categorías.....	86
Tabla 7.13: Descripción del caso de uso: Escribir texto Vocales.....	87
Tabla 7.14: Descripción del caso de uso: Escribir texto Letras agrupadas.....	88

Tabla 7.15: Descripción del caso de uso: Escribir método variable.....	89
Tabla 7.16: Descripción del caso de uso: Escribir imágenes.....	90
Tabla 7.17: Descripción del caso de uso: Salir de la aplicación.....	91
Tabla 7.18: Descripción completa de las clases.....	92
Tabla 7.19: Clase componente teclado del paquete Caso base.....	99
Tabla 7.20: Clase CuadroTexto del paquete Caso base.....	100
Tabla 7.21: Clase EstadoConfiguración del paquete Caso base.....	100
Tabla 7.22: Clase Pantalla del paquete Caso base.....	100
Tabla 7.23: Clase BotonSimple del paquete Caso base.....	101
Tabla 7.24: Clase BotonRadio del paquete Caso base.....	101
Tabla 7.25: Clase Botón del paquete Caso base.....	101
Tabla 7.26: Clase Etiquetas del paquete Caso base.....	101
Tabla 7.27: Clase ListasSeleccion del paquete Caso base.....	101
Tabla 7.28: Clase Reproductor del paquete Caso base.....	102
Tabla 7.29: Clase Barrido del paquete Caso base.....	102
Tabla 7.30: Clase AdaptadorTexto del paquete Caso base.....	102
Tabla 7.31: Archivos .xml del paquete Caso base.....	102
Tabla 7.32: Clase ComponenteTeclado del paquete Un metodo.....	103
Tabla 7.33: Archivos .xml del paquete Un metodo.....	103
Tabla 7.34: Clase ComponenteTeclado del paquete Varios métodos.....	103
Tabla 7.35: Archivos .xml del paquete Varios métodos.....	103
Tabla 7.36: Clase ComponenteTeclado del paquete Escritura textual.....	104
Tabla 7.37: Archivos .xml del paquete Escribir método categorías.....	104
Tabla 7.38: Archivos .xml del paquete Escribir método vocales.....	104
Tabla 7.39: Archivos .xml del paquete Escribir método variable.....	104
Tabla 7.40: Clase ComponenteTeclado del paquete Escribir método imágenes.....	105
Tabla 7.41: Clase Pantalla del paquete Escribir método imágenes.....	105
Tabla 7.42: Clase Botón del paquete Escribir método imágenes.....	105
Tabla 7.43: Clase Imagen del paquete Escribir método imágenes.....	105
Tabla 7.44: Clase BotonImagen del paquete Escribir método imágenes.....	106
Tabla 7.45: Archivos .xml del paquete Escribir método imágenes.....	106
Tabla 7.46: Clase ComponenteTeclado del paquete Funcionalidad extra.....	106
Tabla 7.47: Clase ComponenteTeclado del paquete Funcionalidad simple.....	106
Tabla 7.48: Clase ComponenteTeclado del paquete Opción colores.....	107
Tabla 7.49: Clase ComponenteTeclado del paquete Opción frases.....	107
Tabla 7.50: Clase ComponenteTeclado del paquete Opción completa.....	107
Tabla 7.51: Clase ComponenteTeclado del paquete Colores.....	108
Tabla 7.52: Clase Pantalla del paquete Colores.....	108
Tabla 7.53: Clase Etiquetas del paquete Colores.....	108
Tabla 7.54: Archivos .xml del paquete Colores.....	108
Tabla 7.55: Clase ComponenteTeclado del paquete Soporte frases.....	109
Tabla 7.56: Archivos .xml del paquete Soporte frases.....	109
Tabla 7.57: Clase ComponenteTeclado del paquete Listas básicas.....	109

---

## Capítulo 8

Tabla 8.1: Relación de paquetes y archivos XML.....	118
Tabla 8.2: Relación de prototipos y paquetes necesarios para cada uno.....	122
Tabla 8.3: Clases parciales de Program.cs.....	125
Tabla 8.4: Productos, paquetes y constantes definidas.....	127

---

## Capítulo 10

Tabla 10.1: Prueba de escritura (a) .....	136
Tabla 10.2: Prueba de escritura (b) .....	136
Tabla 10.3: Prueba de escritura (c) .....	136
Tabla 10.4: Prueba escribir y volver (a) .....	137
Tabla 10.5: Prueba escribir y volver (b) .....	137
Tabla 10.6: Prueba escribir y volver (c) .....	137
Tabla 10.7: Prueba borrar carácter .....	138
Tabla 10.8: Prueba borrar palabra .....	138
Tabla 10.9: Prueba insertar espacio en texto .....	138
Tabla 10.10: Prueba mostrar menú .....	138
Tabla 10.11: Prueba salir de la aplicación .....	139
Tabla 10.12: Prueba volver a la pantalla anterior .....	139
Tabla 10.13: Prueba ver temas de ayuda .....	139
Tabla 10.14: Prueba consultar un tema de ayuda .....	139
Tabla 10.15: Prueba reproducir texto .....	140
Tabla 10.16: Prueba desactivar barrido .....	140
Tabla 10.17: Prueba cambiar tiempo de barrido .....	140
Tabla 10.18: Prueba cambiar configuración .....	141
Tabla 10.19: Prueba mostrar la pantalla de frases .....	141
Tabla 10.20: Prueba añadir una frase al texto .....	141
Tabla 10.21: Prueba borrar una frase de la pantalla de frases .....	142
Tabla 10.22: Prueba guardar un texto en el listado de frases .....	142
Tabla 10.23: Prueba desplazamiento de las frases en la lista de frases .....	142
Tabla 10.24: Prueba mostrar la pantalla colores .....	142
Tabla 10.25: Prueba mostrar colores disponibles para cambiar el color de fondo .....	143
Tabla 10.26: Prueba mostrar colores disponibles para cambiar el color de los botones .....	143
Tabla 10.27: Prueba mostrar colores disponibles para cambiar el color de texto de los botones .....	143
Tabla 10.28: Prueba mostrar colores disponibles para cambiar el color de texto .....	144
Tabla 10.29: Prueba cambiar color .....	144
Tabla 10.30: Prueba borrar sin texto .....	144
Tabla 10.31: Prueba borrar una palabra sin texto .....	145
Tabla 10.32: Prueba reproducir sin texto .....	145
Tabla 10.33: Prueba reproducir texto no reproducible .....	145
Tabla 10.34: Prueba cambio de imágenes de botones variables .....	146
Tabla 10.35: Prueba desplazamiento hacia arriba con la primera frase marcada .....	146
Tabla 10.36: Prueba desplazamiento hacia abajo con la última frase marcada .....	146



# **PARTE 1. INTRODUCCIÓN**

---



## Capítulo 1

# PRESENTACIÓN DEL PROYECTO

### 1.1 Descripción y ámbito del proyecto

Este proyecto ha sido propuesto por el Grupo de Investigación en Reutilización y Orientación al Objeto (GIRO) y está orientado al desarrollo de componentes software reutilizables en el dominio de la “Informática para personas discapacitadas”.

Una línea de productos software es un conjunto de productos software, los cuales tienen aspectos comunes que son compartidos por todos ellos y aspectos variables que establecen diferencias entre sí. Los productos de dicha línea son desarrollados a partir de un conjunto de activos de software reutilizables.

El proyecto en concreto, consiste en el análisis, documentación y desarrollo de una línea de productos software en dicho dominio.

Para este fin, la línea de productos se basará en cuatro proyectos de fin de carrera ya realizados, dedicados al desarrollo de comunicadores software para personas discapacitadas.

Uno de los principales problemas que plantea el desarrollo de líneas de productos es la representación y gestión tanto de la variabilidad como de las partes comunes de dicha línea de productos. La forma habitual de definir ambos aspectos durante la fase de requisitos es mediante modelos de características o “features”.

Otro de los problemas que plantean las líneas de producto es cómo plasmar dicha variabilidad en las fases posteriores y distinguir entre la variabilidad propia de la línea de productos y de los productos individuales. El proyecto realizado, se basa en una propuesta del grupo GIRO que propone la utilización de mecanismos UML. La propuesta consiste en la utilización del mecanismo de combinación de paquetes (“package merge”) para representar la variabilidad de la línea de productos, restringiendo los mecanismos clásicos de modelado (como sería el caso de la relación <<extiende>>) para expresar las variantes válidas en tiempo de ejecución de cada aplicación concreta.

Es decir, el propósito principal es aprovechar el modelo de desarrollo convencional aplicado al desarrollo de una línea de productos.

Los proyectos anteriores que se han utilizado para desarrollar a partir de ellos la línea de producto mencionada son, por orden cronológico:

**(i) 2003 - 2004:**

- *"Teclado sobre una PDA para Personas con Parálisis Cerebral"*  
**Autores:** Fermín Juan Martínez, Virginia Vilorio Mozo  
**Tutor:** Miguel Ángel Laguna Serrano  
**PFC Ingeniero Técnico en Informática de Gestión**

**(ii) 2004 - 2005:**

- *"Comunicador Basado en Barrido para PDA"*  
**Autores:** Margarita Lucas Barrios, Noelia Manso Tardón  
**Tutores:** Miguel Ángel Laguna Serrano, Bruno González Baixauli  
**PFC Ingeniero Técnico en Informática de Gestión**

**(iii) 2005 - 2006:**

- *"Desarrollo de un Componente Teclado "*  
**Autor:** Francisco Javier Sánchez Ramos  
**Tutores:** Miguel Ángel Laguna Serrano, Bruno González Baixauli  
**PFC Ingeniero Técnico en Informática de Sistemas**

**(iv) 2006 - 2007:**

- *"Desarrollo de un Comunicador Java para PDA "*  
**Autores:** Beatriz Alonso García, Sandra García Martía  
**Tutores:** Miguel Ángel Laguna Serrano, Bruno González Baixauli  
**PFC Ingeniero Técnico en Informática de Gestión**

## 1.2 Alcance del proyecto

Este proyecto intenta facilitar la incorporación en la medida de lo posible de las personas discapacitadas en la sociedad. Para ello, tiene como objetivo la creación de comunicadores adaptables a sus necesidades. Para llevarlo a cabo, se aplicará el concepto de línea de productos software.

Está destinado a todos aquellos que deseen obtener información sobre el estudio y desarrollo de líneas de producto software y el aprovechamiento para ello de herramientas ya existentes como es UML 2 así como el mecanismo de combinación de paquetes o "package merge".

Propone a empresas y particulares un paso más en la reutilización del software, pudiendo tener la capacidad de crear eficientemente varias variaciones de un mismo producto aprovechando las partes comunes a todos, y particulares de cada variación.

### 1.3 Sobre esta documentación

Esta memoria pretende ser de utilidad para la comprensión del estudio y desarrollo de la línea de productos que nos ocupa, siendo rigurosa con los contenidos teóricos y sin extenderse en descripciones exhaustivas. La memoria se organiza en partes y capítulos, para establecer así, un orden lógico de lectura por temas. A continuación se describe brevemente en qué consistirá cada una de las partes de la memoria:

- **Parte 1. Introducción:**  
Breve presentación acerca del tema que aborda el proyecto, así como los objetivos del mismo y estructura de la memoria.
- **Parte 2. Contexto del problema:**  
Acercamiento al software de comunicación para las personas con discapacidad. Estudio del enfoque sobre las líneas de producto software: Orígenes, motivación y terminología usada; objetivos y beneficios de su utilización.
- **Parte 3. Plan del proyecto:**  
Recursos empleados. Análisis de riesgos. Estimaciones temporales. Fases de la elaboración del proyecto. Seguimiento del proyecto. Diagramas de Gantt.
- **Parte 4. Análisis y desarrollo de la línea de productos:**  
Estudio de los proyectos a partir de los cuales se realizará la línea de productos. Análisis, diseño e implementación de la misma. Guía para el desarrollador de aplicaciones basadas en la línea de productos. Elaboración de aplicaciones de ejemplo y pruebas unitarias.
- **Parte 5. Manual de usuario**  
Guía de la correcta instalación y uso para el usuario final de las aplicaciones generadas mediante la línea de productos.
- **Parte 6. Conclusiones:**  
Reflexión sobre las dificultades encontradas, los objetivos alcanzados y las posibles nuevas vías de actuación del proyecto.
- **Referencias:**  
Bibliografía y recursos digitales empleados.
- **Apéndices:**  
Tecnologías utilizadas  
Estructuras de ficheros XML  
Contenido del CD-ROM



## Capítulo 2

# OBJETIVOS PROPUESTOS

### 2.1 Objetivos

Seguidamente, se describen los objetivos que este proyecto de fin de carrera pretende cumplir:

- Objetivos Específicos del proyecto:
  - Conocer y comprender el concepto de Línea de Producto Software, sus ventajas y sus inconvenientes.
  - Realizar un estudio de proyectos fin de carrera ya realizados y combinarlos para así analizar la línea de producto que se deriva de ellos.
  - Desarrollar dicha línea de productos mediante Visual Studio.NET, de manera que, con el menor esfuerzo posible se puedan crear nuevos comunicadores para personas discapacitadas, seleccionando las partes de cada uno más convenientes.
  - Desarrollo de una serie de prototipos a modo de ejemplos en los que podamos tener un comunicador con funcionalidades básicas y opcionales. Para demostrar la calidad de la línea de productos, se desarrollarán los comunicadores implementados en los proyectos estudiados.
- Objetivos Generales:
  - Conocer y utilizar un entorno de trabajo específico, como el desarrollo de aplicaciones para dispositivos móviles con Visual Studio.NET, y más concretamente, el lenguaje C#.
  - Practicar y mejorar nuestras habilidades relacionadas con el trabajo en equipo.
  - Usar correctamente los conocimientos adquiridos durante la carrera para la elaboración de este proyecto.



# **PARTE 2. CONTEXTO DEL PROBLEMA**

---



## Capítulo 3

# INFORMÁTICA PARA PERSONAS DISCAPACITADAS

### 3.1 Introducción

La informática posee la capacidad de ofrecer soluciones totales o parciales a las personas que tienen algún tipo de discapacidad; sin embargo, a menudo aparecen una serie de barreras de acceso, ya que para hacer uso del ordenador existen un conjunto de requerimientos que la persona discapacitada no puede llevar a cabo de forma autónoma. Por suerte, el ordenador es una máquina con gran versatilidad y brinda la posibilidad de adaptarla a las necesidades específicas de cada usuario.

La detección de las barreras de acceso existentes en cada situación, la elección de la ayuda técnica idónea en función de las capacidades y necesidades de ese usuario y el entrenamiento necesario, son algunas de las etapas a tener en cuenta en el momento de dotar a una persona con aquella adaptación que favorecerá o posibilitará el uso de la computadora.

### 3.2 Comunicación

Para evitar el aislamiento, la informática facilita a las personas con algún tipo de discapacidad su integración en la sociedad. Para conseguirlo se adapta tanto el software como el hardware.

Los comunicadores pretenden lograr la interacción con el entorno a personas con discapacidades, en la medida de lo posible dotando así al usuario de independencia.

Los cuatro proyectos de los que partimos están pensados 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.

### 3.3 Problemas en la comunicación

Las personas 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.) lo que se conoce como disartria. El habla de estas personas 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, las personas con parálisis cerebral deben utilizar métodos alternativos para comunicarse.

### 3.4 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).

Para facilitar la comunicación a los usuarios con alguna discapacidad existen hoy en día numerosos dispositivos orientados a este fin. Tal es el caso de las pantallas táctiles, los diferentes tipos de ratón y teclados, software de reconocimiento de voz, emuladores de ratón y teclado, etc.

Este proyecto se centra, de forma especial, en el uso de dispositivos móviles o PDAs para favorecer la comunicación a las personas con discapacidad.

## Capítulo 4

# LINEAS DE PRODUCTOS SOFTWARE

### 4.1 Introducción

En el mundo de la economía y del marketing, una **línea de productos** implica el ofrecer varios productos relacionados entre sí pero de forma individual. Es decir, es un conjunto de productos que están estrechamente relacionados, ya sea bien porque satisfacen una clase de necesidades específicas o bien porque se usan conjuntamente; es un amplio grupo de productos dedicado a usos similares. Por ejemplo: línea de productos electrónicos: televisores, videos, DVDs, Home Cinemas, etc.

La industria lleva mucho tiempo usando técnicas de ingeniería análogas al desarrollo de líneas de productos, mediante fábricas que encajan y configuran partes diseñadas para ser reutilizadas por distintos productos, dentro de un conjunto específico de éstos. Por ejemplo, los fabricantes automovilísticos pueden crear ahora cientos de miles de variantes únicas de un mismo modelo de coche usando un único repositorio de partes y una factoría específicamente diseñada para configurar y ensamblar dichas partes.

La idea de manufacturar software mediante partes reutilizables ha estado vigente durante décadas, pero el éxito no ha sido el esperado. Echando la vista atrás, vemos que la historia de la programación es como una espiral creciente donde el elemento clave que origina cada vuelta de espiral o cada salto es la reutilización. Ya en los últimos años de la década de los sesenta, la idea de construir sistemas mediante la composición de *componentes* software fue presentada como solución a la afamada *Crisis del software* por M.D. McIlroy. Durante la década de los setenta se propugnó la reutilización de los *módulos*, mientras que en los años ochenta la influencia del *paradigma orientado a objetos* hizo que la *clase* se convirtiera en la unidad de reutilización. Sin embargo, todas estas tendencias fallaban en conseguir un enfoque sistemático de reutilización. Todas ellas daban lugar a iniciativas individuales, frecuentemente realizadas a pequeña escala. Para solucionar este problema surgen los *Frameworks* y la *Programación Orientada a Componentes*.

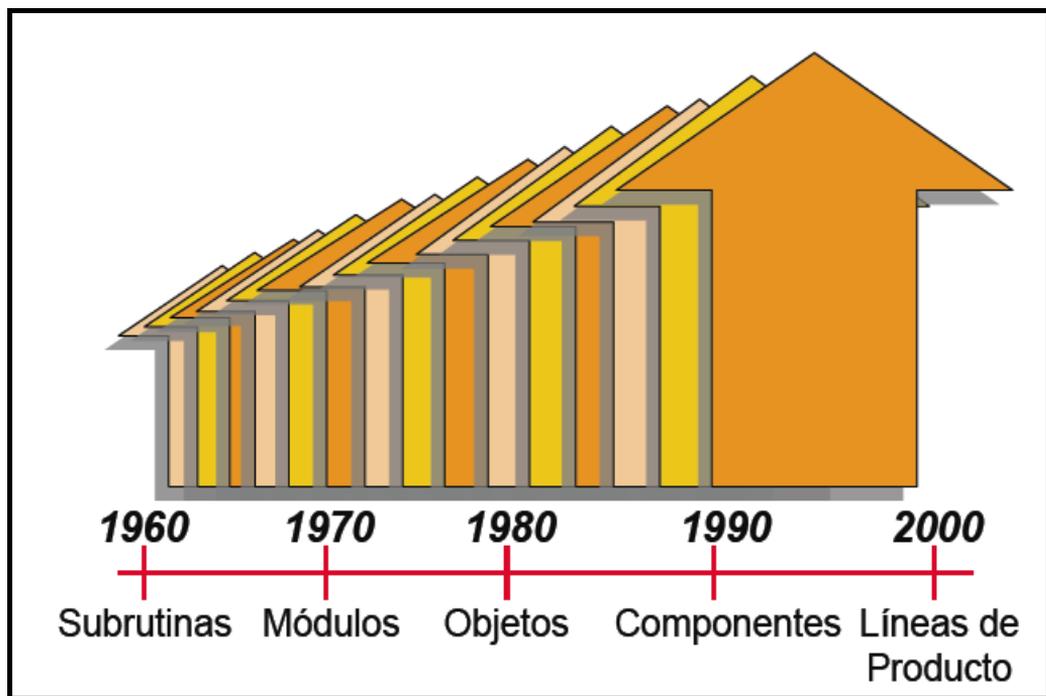


Figura 4.1: Evolución de la reutilización software<sup>1</sup>

Como podemos ver en la Figura 4.1, el siguiente paso en la evolución de la reutilización software son las líneas de productos. Recientes avances en el campo de las líneas de productos software han demostrado que la cuidadosa y estratégica planificación de esos conceptos puede producir mejoras en factores como la calidad, tiempo-al-mercado y el coste de ingeniería de software. El resultado es normalmente una mejora similar a la que se daba con la producción en cadena y la personalización en masa.

La característica principal que distingue las líneas de producto software de los anteriores esfuerzos por la reutilización del software es la reutilización predictiva en lugar de la oportunista. En vez de poner componentes software generales en una librería en espera de que se presente la oportunidad de reutilizarlos, las líneas de producto software sólo crean elementos software cuando la reutilización es predecible en uno o más productos de una línea de productos bien definida.

La producción en masa – la capacidad de crear eficientemente varias copias del mismo producto- representó un gran avance en el mundo de la industria. Crear varias copias de un producto software es una tarea trivial. Por otra parte, la personalización en masa – la capacidad de crear eficientemente varias variaciones de un producto- es un gran avance tanto en el mundo de la industria como en el del desarrollo software. La clave de la personalización en masa es **aprovechar las partes comunes y gestionar eficientemente las variaciones de los productos**, que es justamente el principio en el que se basa el concepto de línea de productos.

<sup>1</sup> De Lara, Juan. Presentación del curso de Postgrado “*Diseño de Software Basado en Modelado y simulación*”, Universidad Autónoma de Madrid, 2006.(Adaptado de Northrop).

Como se ve en la Figura 4.2, los productos en una línea de productos pertenecen al mismo dominio de aplicación, comparten una estructura común y se construyen a partir de componentes software reutilizables.

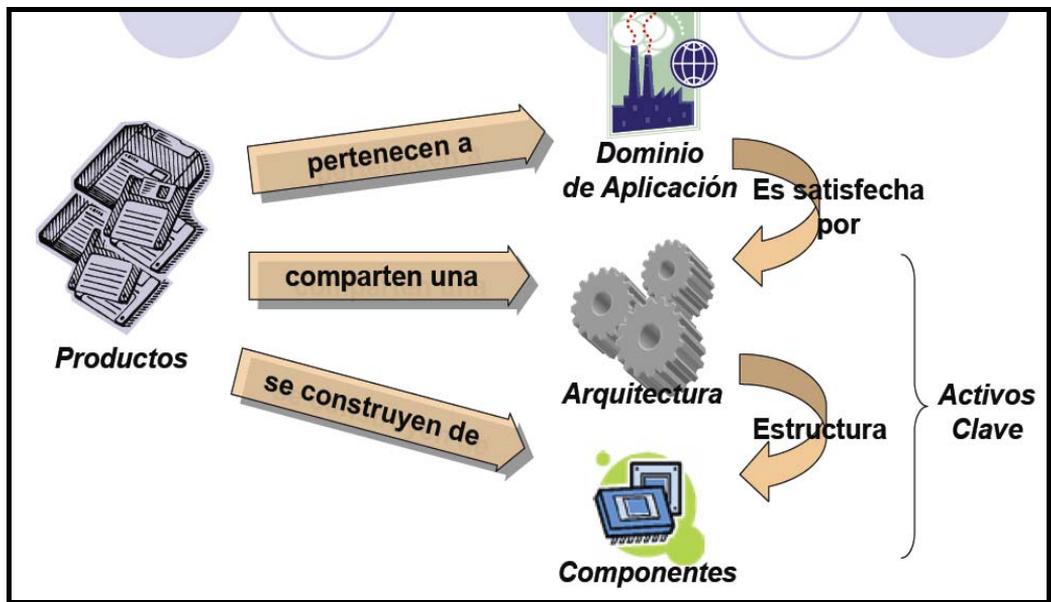


Figura 4.2: Esquema conceptual de la línea de productos<sup>2</sup>

## 4.2 Terminología empleada

Se usan varios **términos** para referirse a líneas de productos software. Estos pueden ser: familias de productos software, familia de sistemas o dominios de aplicación. Los términos más usados en los comienzos de la investigación en este campo eran análisis de dominio y modelado de dominio.

Las líneas de producto software pueden ser descritas en términos de cuatro simples conceptos:

- **Elementos software reutilizables:**  
Una colección de elementos reutilizables –como requisitos, componentes de código fuente, casos de prueba, arquitectura, documentación...- que pueden ser configurados y compuestos de diferentes maneras para crear todos los productos de una línea de productos. Para incluir la variación dentro de los productos, algunos elementos pueden ser opcionales y otros pueden tener puntos de variación internos que pueden dotar al elemento de diferentes comportamientos según se necesite.

<sup>2</sup> De Lara, Juan. Presentación del curso de Postgrado “*Diseño de Software Basado en Modelado y simulación*”, Universidad Autónoma de Madrid, 2006.(Adaptado de Northrop).

- **Decisiones de producto y modelo de decisión:**  
El modelo de decisión describe las características variables y opcionales para los productos de la línea de productos. Cada producto de la línea está definido por sus decisiones de producto (elecciones para cada una de las características opcionales y variables del modelo de decisión).
- **Mecanismo y proceso de producción:**  
Los medios para componer y configurar los productos a partir de los elementos software reutilizables. Las decisiones de producto son usadas en la producción para determinar qué elementos reutilizables usar y cómo configurar los puntos de variación de éstos.
- **Productos software resultantes:**  
La colección de todos los productos que pueden resultar de una línea de productos. El ámbito está determinado por el grupo de productos software resultantes, que pueden originarse a partir de los elementos software y el modelo de decisión.

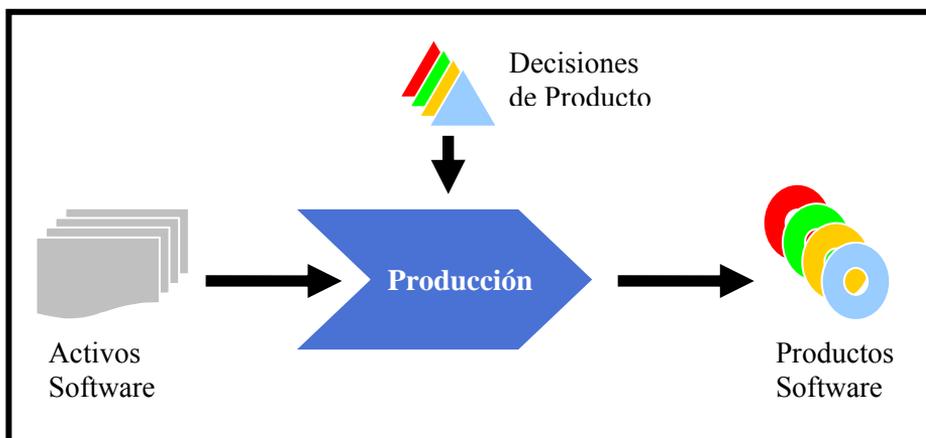


Figura 4.3: Terminología de las líneas de productos<sup>3</sup>

En la figura 4.3 podemos ver los conceptos que intervienen en las líneas de productos. Se dispone de unos activos software reutilizables. Mediante las decisiones de producto se establece cuáles de estos activos se utilizarán y cómo. Después del proceso de producción se obtendrán unos productos software finales acordes con los activos utilizados y las decisiones tomadas.

Un hecho destacable sobre las líneas de producto es que al igual que en la ingeniería del software convencional, los “elementos” de la línea de productos (elementos software reutilizables, decisiones y productos resultantes) están sujetos a mantenimiento y evolución, pero de un modo especial. Debido a la propia naturaleza de la línea de productos y la dependencia entre sus elementos, los cambios evolutivos que se realicen en uno de ellos (por ejemplo, un elemento software reutilizable), pueden tener que propagarse a otros elementos interdependientes en la línea de producto (por ejemplo, una decisión de producto).

<sup>3</sup> Adaptado de <http://www.softwareproductlines.com>. Abril 2007

### 4.3 Objetivos de la línea de productos

Actualmente en el desarrollo de software, existe la necesidad de desarrollar múltiples y similares productos software en lugar de un solo producto individual. Hay numerosas razones para ello. Los productos que son desarrollados para el mercado internacional deben adaptarse a diferentes entornos legales o culturales, así como a distintos lenguajes, proveyendo al usuario de interfaces adaptadas. Debido a que las restricciones de tiempo y coste no permiten al desarrollador crear nuevos productos desde prácticamente cero para cada cliente, la reutilización del software debe alcanzar otro nivel. El desarrollo de líneas de productos software ofrece una solución a estos problemas.

Los conceptos mencionados anteriormente ilustran los objetivos clave de las líneas de productos software. Aprovechar la parte común y gestionar la variabilidad para reducir el tiempo, esfuerzo, coste y complejidad de crear y mantener un conjunto de sistemas software similares:

- **Aprovechar la parte común**, mediante la consolidación y reutilización de los componentes software, evitando la duplicación y la divergencia.
- **Gestionar la variación**, definiendo claramente los puntos de variación y el modelo de decisión, explicitando la localización y dependencias de la variación.

### 4.4 Beneficios

Las líneas de productos software nos proporcionan una serie de beneficios:

- La entrega de productos de software de una manera más rápida, económica y con una mejor calidad.
- Producen mejoras en el tiempo de entrega del producto, costes de ingeniería, reducción de las tasas de defectos y calidad de los productos.
- Incremento en el número total de productos que pueden ser efectivamente desplegados y mantenidos.
- Reducción en el esfuerzo promedio requerido para desarrollar y mantener los productos.
- Mayor agilidad para expandir el negocio a nuevos mercados.



**PARTE 3.**  
**PLAN DE PROYECTO**

---



## Capítulo 5

# PLANIFICACIÓN DEL PROYECTO

### 5.1 Personal necesario

El personal necesario para la elaboración del proyecto fue el siguiente:

- **2** Ingenieros Técnicos Informáticos, cumpliendo los siguientes roles:
  - *Directores de Proyecto*: Recopilación de información previa, definición de necesidades de sistema y selección de proyectos fin de carrera de la línea de productos.
  - *Analistas*: Análisis de la línea de productos y diseño de la misma.
  - *Programadores*: Documentación y adaptación de código existente e implementación de la línea de productos.
  - *Documentalistas*: Documentación de la línea de productos.

### 5.2 Recursos técnicos necesarios

Para la realización del proyecto, se disponen los siguientes recursos hardware:

- Ordenador de sobremesa Intel P4 2.4 GHz.
- Ordenador de sobremesa Intel P4 1.5 GHz.
- PDA cedida por el grupo GIRO.

Se dispone, asimismo de los siguientes recursos software:

- Microsoft Windows XP SP 2
- Microsoft Visual Studio.NET 2005
- .NET Compact Framework 2.0 (SP 2)
- StarUML
- Microsoft Office 2003

Se ofrecerá más información sobre los recursos hardware y software utilizados en la elaboración del proyecto en la parte de implementación de la presente memoria.

## 5.3 Gestión de riesgos

### 5.3.1 Identificación de riesgos del proyecto

Se recogen a continuación en la tabla 5.1, los riesgos que se han identificado para la realización del presente proyecto:

RIESGO	DESCRIPCION
Conflictos entre proyectos	Puede que alguna funcionalidad de un proyecto esté implementada de tal modo que no permita la funcionalidad de algún otro, o requiera alguna solución compleja para poder ser implementada con garantías.
Funcionalidad ausente	En la etapa de implementación se pueden encontrar funcionalidades o características que no hayan sido tenidas en cuenta en la captura inicial de requisitos.
Funcionalidad inviable	Puede que funcionalidades o características planteadas en la captura inicial de requisitos, se consideren inviables, bien porque requieran de una implementación compleja que necesite unos recursos de tiempo de los que no se dispone o porque entre en conflicto con alguna otra característica.
Gestión de la variabilidad de la interfaz inviable	Puede ser que al gestionar la variabilidad intrínseca de la línea de productos, en la teoría no haya problema, pero a la hora de la implementación y debido a la particular naturaleza de los proyectos empleados en la línea de productos, no se dé con un método de implementación realizable.
Estudio y adaptación de Visual Studio .NET 2005 y lenguaje C#.	Los proyectos a estudiar están en su mayoría desarrollados con Visual Studio 2003 e incluso utilizan lenguajes distintos, como C# y JAVA. El adaptar de VS.NET 2003 a VS.NET 2005 debería ser trivial; no así el adaptar código JAVA a C#, a pesar de sus similitudes. Es posible que haya que modificar y adaptar código para poder integrar sus funcionalidades en la línea de productos.
Tiempo insuficiente para cumplir los objetivos.	Puede que determinados aspectos como el estudio de los proyectos previos o la adaptación del código de los mismos, conlleve un coste de tiempo distinto al planteado inicialmente.
Situaciones personales	Debido a la existencia de razones externas a la realización del proyecto, es posible que no se pueda dedicar el tiempo estimado en un principio a la realización del mismo.

Tabla 5.1: Identificación de riesgos del proyecto

### 5.3.2 Incidencia de riesgos sobre el proyecto

Para calcular la incidencia de los riesgos identificados sobre el proyecto se han establecido, como puede verse en la tabla 5.2, niveles cuantitativos para evaluar cada uno de ellos:

Probabilidad	Intervalo	Impacto	Intervalo
Muy Baja	[0.0 – 0.2)	Despreciable	[0 - 2)
Baja	[0.2 – 0.4)	Marginal	[2 - 4)
Media	[0.4 – 0.6)	Crítico	[4 - 8)
Alta	[0.6 – 0.8)	Catastrófico	[8 - 10]
Muy Alta	[0.8 – 1.0]	-----	-----

Tabla 5.2: Niveles de riesgo.

Con la tabla anterior ya se puede establecer la tabla de incidencias de los riesgos que pueden darse en el proyecto:

Impacto Probabilidad	Impacto			
	Despreciable	Marginal	Crítico	Catastrófico
Muy Baja	Baja	Baja	Baja	Moderada
Baja	Baja	Baja	Baja	Moderada
Media	Baja	Baja	Moderada	Alta
Alta	Moderada	Moderada	Alta	Alta
Muy Alta	Moderada	Moderada	Alta	Alta

Tabla 5.3: Incidencia de los riesgos.

Una vez que se tienen los niveles cuantitativos de los riesgos y la tabla de incidencia de los mismos, se puede elaborar la estimación de incidencia de cada uno de los riesgos identificados en el proyecto. Dicha información se recoge en la tabla 5.4:

<b>Riesgo</b>	<b>Probabilidad</b>	<b>Impacto</b>	<b>Incidencia</b>
Conflictos entre proyectos	Baja (0.3)	Crítico(4)	Baja (1.2)
Funcionalidad ausente	Alta(0.7)	Marginal(3)	Moderada(2.1)
Funcionalidad inviable	Baja(0.3)	Crítico(4)	Baja(1.2)
Gestión de la variabilidad de la interfaz inviable	Muy Baja(0.2)	Catastrófico(8)	Moderada(1.6)
Estudio y adaptación de V.S.NET 2005 y C#.	Media(0.4)	Marginal(3)	Baja(1.2)
Tiempo insuficiente para cumplir los objetivos.	Baja(0.3)	Marginal(2)	Baja(0.6)
Situaciones personales	Alta(0.7)	Marginal(2)	Moderada(1.4)

Tabla 5.4: Tabla de riesgos del proyecto.

### 5.3.3 Medidas para la reducción y gestión de riesgos

Se plantean a continuación, en las siguientes tablas, las medidas a tomar para reducir la incidencia de los riesgos del proyecto.

<b>RIESGO</b>	<b>Medidas para reducir la PROBABILIDAD de aparición</b>
Conflictos entre proyectos	Comprobar y clasificar las funcionalidades de cada proyecto, localizando así las incompatibilidades entre ellos.
Funcionalidad ausente	Realizar una exhaustiva captura de requisitos, no sólo centrándose en los proyectos previos, sino ampliando el enfoque a la línea de productos en sí misma.
Funcionalidad inviable	Analizar convenientemente las distintas funcionalidades a incluir en la línea de productos.
Gestión de la variabilidad de la interfaz inviable	Realización de una extensa documentación previa sobre los métodos de gestión de la variabilidad en las líneas de productos.
Estudio y adaptación de Visual Studio .NET 2005 y lenguaje C#.	No se pueden tomar medidas más concretas para reducir esta probabilidad. Se tendrá que adaptar código JAVA.
Tiempo insuficiente para cumplir objetivos.	Aumentar el tiempo de trabajo en la segunda mitad del desarrollo del proyecto. Reorganizar las tareas en las etapas finales del mismo.
Situaciones personales	No se pueden tomar medidas más concretas para reducir esta probabilidad.

Tabla 5.5: Medidas de reducción de probabilidad de riesgos.

<b>RIESGO</b>	<b>Medidas para reducir el nivel de IMPACTO</b>
Conflictos entre proyectos	En la medida de lo posible, adaptar el código de las funcionalidades relacionadas de un mismo proyecto, de modo que se eliminen la mayor parte de las incompatibilidades.
Funcionalidad ausente	Elaborar un diseño adecuado de la línea de productos. Las nuevas funcionalidades necesarias que surjan podrán ser incluidas con relativa facilidad.
Funcionalidad inviable	En el caso de localizar una funcionalidad inviable, se descartará en el desarrollo indicando los motivos.
Gestión de la variabilidad de la interfaz inviable	Se tendrán en cuenta varios métodos alternativos de gestión de variabilidad de interfaz. En caso de inviabilidad se documentará este hecho y se dispondrá de un conjunto de interfaces fijas para poder seguir desarrollando el resto del proyecto.
Estudio y adaptación de Visual Studio .NET 2005 y lenguaje C#.	Utilización del conversor integrado en Visual Studio 2005 .NET para adaptar el código de los proyectos elaborados con Visual Studio 2003.
Tiempo insuficiente para cumplir objetivos.	Dar prioridad a la realización de la estructura de la línea de productos y las características comunes, dejando para más adelante las características opcionales y alternativas.
Situaciones personales	Aprovechar al máximo el tiempo de trabajo en el proyecto.

Tabla 5.6: Medidas de reducción de impacto de riesgos.

## 5.4 Estimaciones Temporales

### 5.4.1 Fechas de comienzo y fin de proyecto

Se reservará una semana para la preparación de la documentación, impresión y encuadernación de la misma, así como para la elaboración del material a entregar en formato digital. Se reservarán asimismo dos semanas para la revisión y finalización de la documentación a presentar. Inicialmente se estima como fecha límite de finalización de proyecto el 26 de mayo de 2008, considerando como fecha de comienzo del proyecto el 2 de Abril de 2007, pudiendo adelantarse o retrasarse dicha fecha dependiendo de los riesgos aparecidos en el proyecto.

### 5.4.2 Fases de la elaboración del proyecto. Tareas

El desarrollo del proyecto creado se ha dividido en varias fases, realizándose en cada una de ellas un trabajo específico acorde a cada fase en cuestión. Las fases son las siguientes:

- **Planificación y Recopilación de información.** En esta fase se busca y asimila información acerca del tema principal de nuestro proyecto. Se recopila documentación sobre el mismo procedente de varias fuentes y se realiza un estudio para seleccionar el mejor enfoque a la hora de realizar el proyecto. Se comienza la redacción de la memoria, entre ella el plan de proyecto, identificando y analizando riesgos, elaborando planes de prevención y contingencia y distribuyendo los recursos disponibles para las diferentes tareas.

- **Selección y análisis de los proyectos.** En esta fase se seleccionan los proyectos de fin de carrera del grupo GIRO que darán lugar a la posterior línea de productos. Asimismo se realiza un estudio exhaustivo de dichos proyectos, poniendo especial atención al análisis de requisitos, diagramas de análisis y de diseño.
- **Análisis de la línea de productos.**
  - **Especificación de requisitos de la línea de productos.**  
Los datos obtenidos de la fase anterior se examinan y combinan para así obtener una especificación de requisitos común a todos los proyectos de fin de carrera implicados. Se elabora el de diagrama de Casos de Uso de la línea de productos.
  - **Listado y diagrama de características de la línea de productos.**  
A partir de los requisitos obtenidos se elabora un listado con las características (features) de la línea de productos. Posteriormente se clasifican éstas y se obtiene el diagrama de características inicial.
- **Diseño de la línea de productos.** Una vez estudiado el problema planteado en la fase de análisis, en esta fase se determina la forma de resolver dicho problema, y de este modo, se estudia y proponen soluciones de implementación y realización.
- **Implementación.** Adaptación del código de los proyectos seleccionados e inclusión de código propio implementando las nuevas funcionalidades añadidas. Mejoras a la funcionalidad del comunicador.
- **Verificación.** Diseño, implementación y ejecución de las pruebas unitarias de cada una de las funcionalidades disponibles en la línea de productos. Desarrollo de una serie de aplicaciones a modo de ejemplo, combinando distintas funcionalidades. Pruebas de dichas aplicaciones tanto en emulador como en dispositivo móvil físico.
- **Documentación.** Redacción del manual de usuario. Guía de instalación y uso de una aplicación desarrollada con la línea de productos.

### 5.4.3 Estimación de tiempos

A continuación presentamos un cronograma de barras o diagrama de Gantt en el cual se detallan las distintas fases previstas del desarrollo del proyecto fin de carrera. Con él se quiere mostrar el tiempo estimado de dedicación a las diferentes tareas de la realización del proyecto. Se ha realizado una estimación del tiempo, teniendo en cuenta la más que probable aparición de alguno de los riesgos anteriormente descritos.

El proyecto de Fin de Carrera se comenzó a realizar en Abril de 2007 y se ha finalizado en Junio de 2008. En cada día laboral se han invertido alrededor de unas 4 o 5 horas diarias de trabajo, lo que supone una media de 23 horas a la semana. Salvo alguna excepción, no se ha trabajado en el proyecto los días no laborables (fines de semana y festivos nacionales) y no ha habido vacaciones en periodo estival ni navidad.

Con el objetivo de ser más realistas, en su realización hemos incluido en el diagrama de Gantt los distintos roles que participan en el desarrollo de un producto software (Jefe de Proyecto, Analista, Programador, etc.), de tal modo que se puedan diferenciar las etapas del desarrollo y los roles que intervienen en cada una de ellas.

La tarea “Reuniones” es una tarea repetitiva que se realiza a lo largo de todo el proyecto y que consiste en mantener reuniones entre los participantes en el proyecto.

Para más información, previamente al diagrama de Gantt, exponemos los porcentajes de tiempo dedicado por tarea:

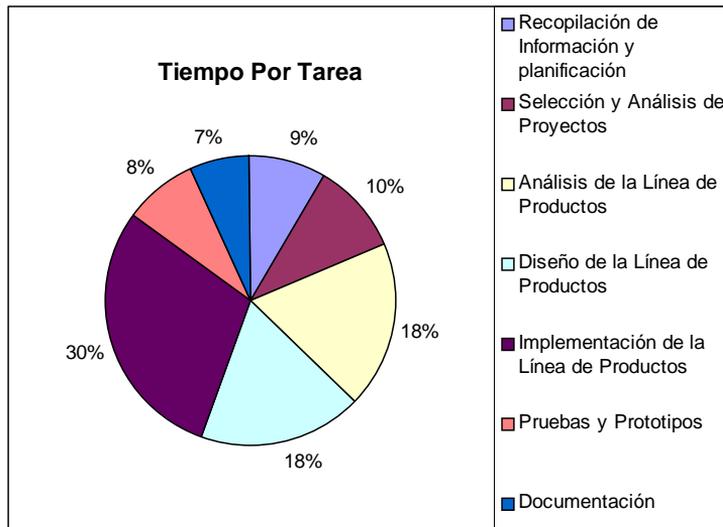


Figura 5.1: Porcentajes de tiempo estimado dedicado por tarea.

La siguiente tabla muestra el desglose del tiempo estimado dedicado:

Porcentaje	Tarea	Días
6%	Recopilación de información	18
3%	Planificación del proyecto	8
10%	Selección y análisis de proyectos fin de carrera	30
18%	Análisis de la línea de productos	55
18%	Diseño de la línea de productos	55
30%	Implementación de la línea de productos	89
4%	Creación de aplicaciones de ejemplo	12
4%	Realización de pruebas	13
7%	Documentación del proyecto	20
<b>TOTAL</b>		<b>300</b>

Tabla 5.7: Desglose de tiempo estimado por tarea.



## 5.5 Seguimiento del proyecto

Finalmente, las estimaciones del tiempo de realización del proyecto han resultado ser optimistas y el desarrollo del proyecto se ha alargado 14 días laborables. Aunque al hacer la estimación inicial se tuvo en cuenta posibles problemas a encontrar y resolver, no se ha sabido identificar la magnitud real de éstos. No se han podido cumplir los plazos previstos por diversos motivos:

- El estudio de los proyectos fin de carrera, así como la adaptación de su código, ha sido más complicado de lo esperado no sólo por la disparidad de criterios a la hora de la elaboración de los mismos, sino también por una documentación menos precisa de lo que se esperaba.
- La documentación se ha extendido más de lo previsto, pues se ha querido plasmar suficiente información tanto de la línea de productos que se desarrolla como de las distintas aplicaciones que ésta puede generar. Aunque se tenía presente que iba a ser un proceso laborioso, ha resultado serlo aún más de lo planificado.

### 5.5.1 Riesgos aparecidos en el desarrollo del proyecto

Durante el desarrollo han aparecido algunos de los riesgos que se habían identificado previamente, retrasando el proyecto considerablemente. Por ello el tiempo de desarrollo planificado ha sido insuficiente y para intentar conseguir los objetivos se alargó la planificación. Los motivos de que han hecho fallar la planificación inicial son:

- Conflictos entre proyectos. Aunque todos los proyectos que se han estudiado tenían como objetivo la creación de un comunicador para personas discapacitadas, sus diseños eran todos muy diversos. Esto ha llevado no sólo a un incremento del tiempo dedicado a su estudio y adaptación de código, sino también a la necesidad de elaboración de mucho más código nuevo del que se planificó inicialmente.
- En la fase de implementación de la línea de productos, se descubrió que para poder adaptar algunas funcionalidades se necesitaban nuevas características que no estaban definidas anteriormente. Afortunadamente, gracias al diseño de la línea de productos se pudieron incluir estas nuevas características sin demasiado esfuerzo.
- Por último, las situaciones personales han hecho que el proyecto se retrase de un modo mayor al que estaba planificado en un principio.

Al final, se ha podido cumplir los objetivos propuestos gracias a haber ampliado el tiempo de planificación del proyecto.

### 5.5.2 Distribución real de tiempos

La distribución de esfuerzos y de tiempo es diferente de la planificada en un principio. Por un lado se redujo el esfuerzo de análisis y diseño de la línea de productos. Esto fue debido al estudio de los proyectos previos y a la documentación acerca de la variabilidad de la línea de productos, que simplificó el diseño de la misma. Del mismo modo, la tarea de creación de prototipos fue prácticamente trivial una vez terminada la línea de productos y redujo mucho su tiempo estimado. La implementación de la línea y la documentación del proyecto, por el contrario, aumentaron su tiempo estimado debido a los motivos ya señalados anteriormente. A continuación, en la figura 5.3 y en la tabla 5.8 se muestra la distribución real de los tiempos empleados en cada tarea:

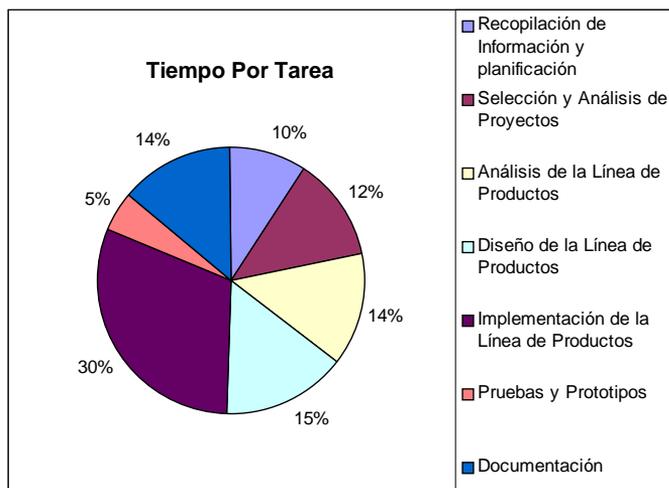


Figura 5.3: Porcentajes de tiempo real dedicado por tarea.

Porcentaje	Tarea	Días
6%	Recopilación de información	24
2%	Planificación del proyecto	6
12%	Selección y análisis de proyectos fin de carrera	39
14%	Análisis de la línea de productos	43
15%	Diseño de la línea de productos	46
30%	Implementación de la línea de productos	97
0.6%	Creación de aplicaciones de ejemplo	2
4%	Realización de pruebas	13
14%	Documentación del proyecto	44
<b>TOTAL</b>		<b>314</b>

Tabla 5.8: Desglose de tiempo real por tarea.

### 5.5.3 Planificación real del proyecto

A continuación en la figura 5.4, se muestra el diagrama de Gantt correspondiente al trabajo real realizado en el proyecto.

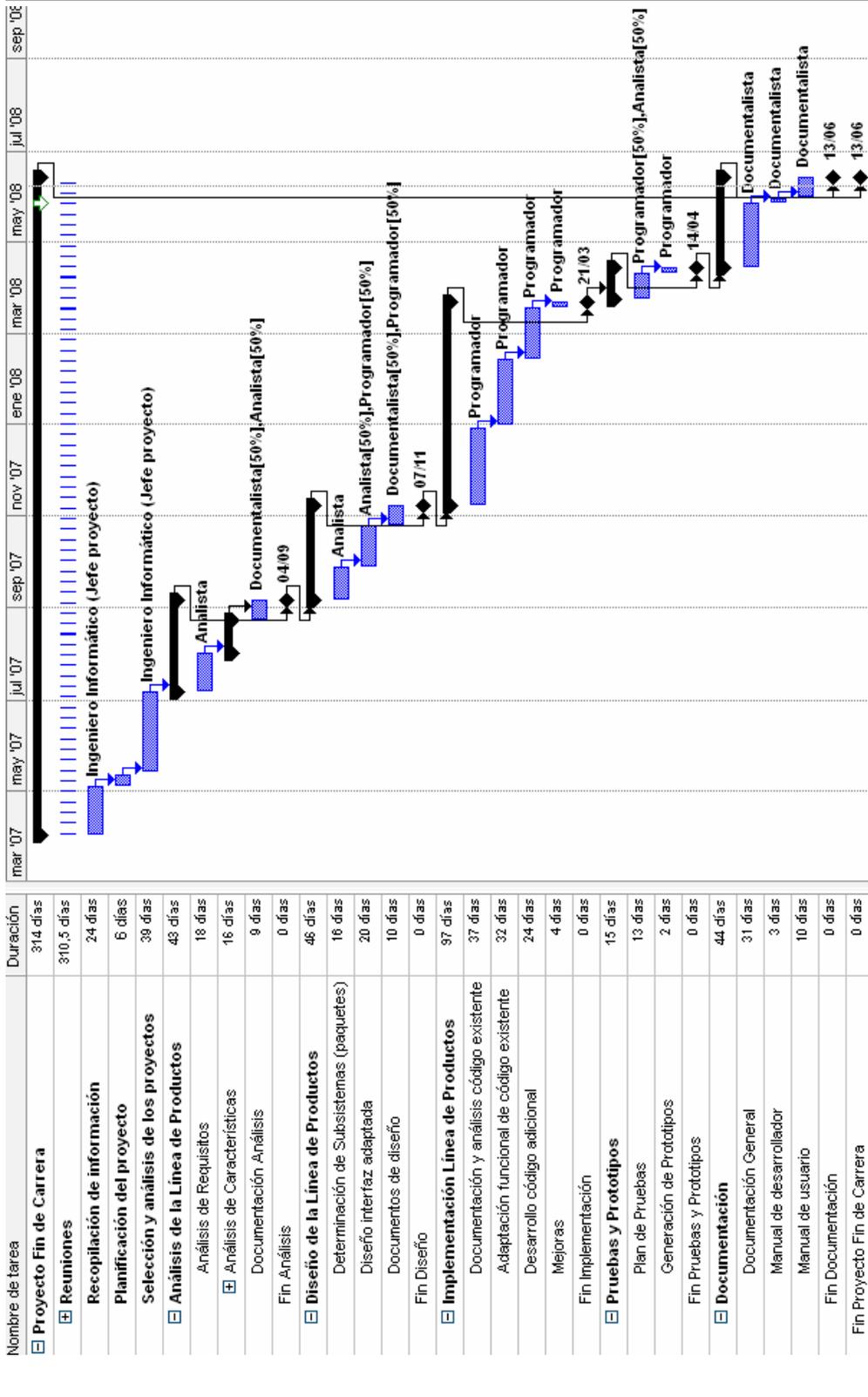


Figura 5.4: Diagrama de Gantt del tiempo real empleado

## 5.6 Horas de trabajo

Teniendo en cuenta que el proyecto Fin de Carrera se comenzó a realizar en Abril de 2007 y se ha finalizado en Junio de 2008 y que en cada día laboral se han invertido alrededor de unas 4 o 5 horas diarias de trabajo, tenemos que las horas para cada fase del proyecto son aproximadamente las siguientes:

Fase	Fechas	Días laborales	Horas de trabajo
Recopilación de información	02 Abril 07 - 03 Mayo 07	24 días	108 horas
Planificación del proyecto	04 Mayo 07 – 11 Mayo 07	6 días	27 horas
Selección y análisis de proyectos	14 Mayo 07 - 05 Julio 07	39 días	175 horas
Análisis de la línea de productos	06 Julio 07 - 04 Septiembre 07	43 días	192 horas
Diseño de la línea de productos	05 Julio 07 - 07 Noviembre 07	46 días	204 horas
Implementación de la línea de productos	08 Noviembre 07 - 21 Marzo 08	97 días	436 horas
Pruebas y prototipos	24 Marzo 08 - 14 Abril 08	15 días	67 horas
Documentación	25 Abril 08 – 13 Junio 08	44 días	198 horas
		<b>314 días</b>	<b>1407 horas</b>

## 5.7 Evaluación de costes

### ▪ Costes profesionales:

Según varias ofertas del mercado laboral actual, establecemos un coste aproximado por hora de 20€ para cada profesional implicado en la elaboración del proyecto, obteniendo los siguientes costes:

	Coste/Hora	Horas empleadas	Coste Total
2 Ingenieros Tec. Informáticos	20€	1407 horas	28.140€

### ▪ Costes de equipamiento y costes indirectos

Debido al hecho de que el proyecto ha sido desarrollado con los equipos, programas e instalaciones de las que dispone la Universidad de Valladolid o los propios autores del proyecto de fin de carrera, no se tendrán en cuenta los diversos costes que podrían derivar de:

- Equipos informáticos utilizados
- Dispositivos móviles utilizados
- Software utilizado
- Local y mantenimiento del mismo

**PARTE 4. ANÁLISIS Y  
DESARROLLO DE LA LÍNEA  
DE PRODUCTOS**

---



## Capítulo 6

# ANÁLISIS DE LA LÍNEA DE PRODUCTOS

El propósito del análisis de la línea de productos es decidir qué productos de esta línea se van a desarrollar. El análisis incluye la definición concreta de la línea de productos, delimitando el dominio del problema y de la solución, así como la evaluación de su alcance.

### 6.1 Definición de la variabilidad de la línea de productos

Durante los últimos años, los sistemas software tienden a soportar una mayor variabilidad, entendida como la habilidad de cambio o de personalización de un sistema. Esta variabilidad se debe a las demandas por parte de los usuarios, de sistemas más adaptables a sus necesidades (sistemas personalizables). También se debe a la presión del mercado, que hace más rentable fabricar líneas de productos software para reutilizar la mayor parte del esfuerzo de desarrollo.

La disciplina de *Ingeniería de Requisitos para Líneas de Productos* es clave en este tipo de desarrollo. Por ello, es preciso mejorar la representación y gestión de requisitos con especial hincapié en los aspectos de variabilidad y trazabilidad.

Para definir esta variabilidad se utilizará el modelo de características y casos de uso.

Hay dos estrategias básicas para desarrollar una línea de productos software: ingeniería evolutiva hacia delante e ingeniería evolutiva inversa. La ingeniería evolutiva hacia delante es mejor cuando se desarrolla una nueva línea de productos y la funcionalidad común puede ser determinada antes que la funcionalidad variable. La inversa se aplica más cuando hay sistemas que son candidatos para la línea de productos y están disponibles para su análisis.

En el caso que aquí se trata, se opta por **la estrategia de la ingeniería evolutiva inversa**. El modelo de casos de uso para cada sistema individual es analizado y documentado primero. Los diferentes modelos de casos de uso son integrados en un solo modelo para la línea de productos. Esto significa que los casos de uso de cada miembro son especificados primero. Si no existe un caso de uso para un miembro hay que desarrollarlo previamente. Después de que los casos de uso para cada sistema han sido desarrollados, se comparan.

### **6.1.1 Definición de la línea de productos**

La definición de la línea de productos, describe los problemas que se espera que esta línea solucione y los productos que se esperan conseguir para resolver esos problemas. También describe el contexto en el que se encuentran estos problemas y en el que se espera que los productos resultantes sean utilizados.

El proyecto a elaborar consiste en la realización de una línea de productos software, enfocada al desarrollo de comunicadores para personas con discapacidad. Concretamente, los comunicadores se centran en la escritura mediante texto o imágenes y la posterior reproducción del mismo.

Estas aplicaciones deberán poder reproducir el texto que escriba el usuario, solicitar ayuda para que le muestre las distintas funcionalidades de la aplicación y su modo de uso y salir de la aplicación. Todas estas son funcionalidades comunes de todos los comunicadores disponibles en la realización de este proyecto.

Todos los proyectos de fin de carrera que forman parte del estudio tienen la posibilidad de escribir, como mínimo, de dos formas diferentes. El método de escritura común en todos ellos es el de escribir mediante letras agrupadas. Como variantes tendríamos: escribir mediante el método de las vocales, escribir mediante categorías o escribir mediante imágenes.

Por tanto, contemplamos las siguientes posibilidades:

- Un solo método de escritura: Letras agrupadas
- Dos métodos de escritura a escoger:
  - Letras agrupadas y Método de vocales.
  - Letras agrupadas y Método categorías.
  - Letras agrupadas y Método imágenes.
- Tres métodos de escritura:
  - Letras agrupadas, Método de vocales y Método de categorías.
  - Letras agrupadas, Método de vocales y Método de imágenes.
  - Letras agrupadas, Método de categorías y Método de imágenes.
- Cuatro métodos de escritura:
  - Letras agrupadas, Método de vocales, Método de categorías y Método de imágenes.

Para poder escribir en cada una de estas opciones, anteriormente se deberá configurar la aplicación para que se adapte al método de escritura deseado.

Debido a que todas las aplicaciones de las cuales nos basamos para formar nuestra línea de productos están aplicadas a dispositivos móviles de pequeño tamaño (PDA), es imposible representar en una única pantalla toda la funcionalidad ofrecida por el componente y, por tanto, se usarán varias pantallas.

Se deberá desarrollar el componente lo más configurable y flexible que sea posible, es decir, que suponga el menor esfuerzo realizar cambios en la funcionalidad y apariencia del teclado o que permita añadir nuevas configuraciones.

---

Recopilando los distintos métodos de escritura de los proyectos de fin de carrera consultados, tenemos:

- **Método de letras agrupadas:** Deberá existir una pantalla inicial en la que se mostrarán varios conjuntos de caracteres (por ejemplo: AÁBCDE, ÉFGHIÍ, etc.). La selección de uno de estos conjuntos llevará a otra pantalla, en la que aparecerán los caracteres que lo forman, redistribuidos de forma independiente (por ejemplo: si pulsamos el conjunto AÁBCDE, aparecerá una nueva pantalla con las letras A, Á, B, C, D, E). La selección de alguna de estas letras es la que hará que se añadan caracteres al cuadro de texto.
- **Método de las vocales:** Este método es similar al anterior, con la diferencia de que las vocales se representan de forma independiente en la pantalla principal, lo que permite agilizar la escritura reduciendo el número de pulsaciones.
- **Método de categorías:** Existirá una pantalla inicial en la que se mostrarán las categorías de los grupos de caracteres (por ejemplo: consonantes, vocales y números). La selección de éstos tendrá dos posibles efectos:
  1. Ir a otra pantalla, en la que aparecerá distribuidos los caracteres de la categoría seleccionada en la pantalla inicial. La selección de estos caracteres es la que hace que se añadan al cuadro de texto.
  2. Ir a otra pantalla, en la que aparecerán grupos de 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 imágenes que pertenecerán a un alfabeto conocido por el usuario. Para facilitar su localización a la hora de escribir, se deberán agrupar por categorías (personas, cosas, animales, etc.). Existirá una pantalla inicial que permitirá al usuario seleccionar la categoría. Una vez elegida, se mostrará una pantalla con un nuevo conjunto de imágenes. La selección de una de estas últimas imágenes hará que se añada en el cuadro de texto su texto asociado. Debido a la existencia de más imágenes dentro de una categoría que imágenes que caben en una pantalla, será necesario que las imágenes de una pantalla vayan cambiando temporalmente, es decir, es necesario que sean variables.

El usuario tendrá la posibilidad de escribir mediante el método de barrido. En este método, en la pantalla salen todas las letras y símbolos que el usuario puede escribir. A través de un cursor de color diferente, se le indica al usuario la tecla que en ese momento está activa. Cuando el cursor se encuentra encima del carácter deseado, el usuario sólo tiene que apretar con cualquier parte del cuerpo un pulsador conectado a la PDA y esa letra será seleccionada. A continuación se vuelve a hacer un barrido de todas las letras.

Todos los proyectos de fin de carrera disponen de un componente software encargado de la reproducción del texto escrito por el usuario.

## 6.2 Análisis de requisitos. Modelo de características

La diferencia entre posibles productos de la línea de productos, puede ser discutida en términos de *features* o *características*.

*“Una característica o feature es una característica relevante para algún interesado y que representa un aspecto común o variable de un producto software”*<sup>4</sup>

Los requisitos de la línea de productos son la suma de los requisitos de todos los miembros que componen la línea.

En estos términos, una característica es un requisito que es satisfecho o proporcionado por uno o más miembros de la línea de productos. En particular, las características se usan para diferenciar los miembros de dicha línea y así determinar y definir la funcionalidad común y variable de una línea de productos software.

Cualquier miembro de la línea de productos, normalmente proporciona sólo algunas de las características de la línea. Al contrario que en un único sistema (en el que todas las características son proporcionadas), en una línea de productos una característica dada puede no ser proporcionada por un determinado miembro de la línea. Es necesario analizar la diversidad de las características de la línea de productos y clasificarlas de acuerdo con su uso. Debido a la dificultad añadida en analizar la variabilidad, el análisis de características para una línea de producto software suele ser más complejo que para un sistema simple.

Se hace una importante distinción entre características comunes, opcionales y alternativas.

- **Comunes:** proporcionadas por todos los miembros de la línea de producto.
- **Opcionales:** sólo por algunos miembros de la línea.
- **Alternativas:** cuando hay que elegir al menos una de las características para un determinado miembro de la línea de productos. Son por lo general mutuamente excluyentes, queriendo decir que sólo se puede seleccionar una de esas características.

El énfasis en el análisis de características está en modelar la variabilidad proporcionada por las características alternativas y opcionales, porque esas características son en lo que difiere un miembro de otro en la línea de productos.

Como primer paso para el análisis de variabilidad hemos analizados los distintos proyectos previos. Con el objetivo de hacer más visual y más fácil el análisis y clasificación de los diferentes tipos de características, se han agrupado todos los requisitos encontrados en la tabla 6.1, indicando las aplicaciones que contienen tales requisitos:

---

<sup>4</sup> K.Czarnecki and U.W. Eisenecker. *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley, Boston, MA, 2000.

<b>CARACT.</b>	<b>PROYECTOS DE FIN DE CARRERA</b>			
	<b>Comunicador Java para PDA (2006)</b>	<b>Componente teclado (2006)</b>	<b>Comunicador basado en barrido para PDA (2005)</b>	<b>Teclado sobre PDA (2004)</b>
Reproducir texto	✓	✓	✓	✓
Consultar ayuda	✓	✓	✓	✓
Escribir por agrupación de letras	✓	✓	✓	✓
Escribir por categorías	✗	✓	✓	✗
Escribir por imágenes	✗	✓	✓	✗
Escribir por el método de vocales	✓	✗	✗	✓
Cambiar el color de los elementos	✓	✓	✓	✗
Elegir tiempo de barrido	✗	✓	✓	✗
Disponer y gestionar frases	✓	✗	✓	✓
Borrar última palabra	✓	✓	✓	✗

Tabla 6.1: Clasificación de las características según los proyectos de fin de carrera estudiados.

Consideraciones:

- Además de los cuatro posibles métodos de escritura dispuestos por los proyectos de fin de carrera estudiados, con el fin de hacer la aplicación más reutilizable y útil en un futuro, se incorpora un nuevo método de escritura: **método variable**. Este método permitirá dos cosas: la posibilidad de introducir un nuevo método de escritura, simplemente rellenando con los datos deseados la disposición de los botones y su contenido, y ofrecer la posibilidad de crear un método de escritura totalmente personalizado según las exigencias del usuario sin ningún tipo de complicación.

- Un hecho destacable es el de la capacidad de disponer de varios métodos de escritura. Siempre, en cualquier aplicación resultante de la línea de productos, se dispondrá del método de escritura mediante letras agrupadas. Ahora bien, también se da la posibilidad de disponer de varios métodos de escritura. Es decir, se contemplan dos características alternativas mutuamente excluyentes: **Un método de escritura** y **Varios métodos de escritura**. Si se dispone de algún otro método de escritura adicional, se deberá permitir cambiar de un método de escritura a otro dentro de la aplicación. Es por esto, que surge una nueva característica: **Configurar método de escritura**, que permite tal operación. Hay diferencias fundamentales de funcionalidad en cuanto al tipo de escritura: si es **Escritura Textual** o es mediante imágenes.
- Como se puede ver en la tabla 6.1, tenemos como característica “Disponer y gestionar frases”. Este soporte de frases atiende a tres características comunes que se deben de dar en tal situación, que serían: **Borrar frase, Guardar Frase e Insertar Frase**. Además, hay que tener en cuenta que las frases deben aparecer en un listado. Originalmente se cuenta con un listado para el tiempo de barrido y otro para los métodos de escritura disponibles en la configuración: **Listas básicas**. Si se cuenta con la opción de manejo de frases, estas listas básicas se tendrían que ampliar con las de frases.
- Para poder añadir una funcionalidad a la aplicación, ya sea colores, frases o ambas, es necesario una nueva característica **Añadir Opción**, que estará destinada a tal fin.
- Dependiendo de si lo que se añade es la funcionalidad para frases o colores, el comunicador resultante tendrá una apariencia distinta en cuanto a lo que cada característica ofrece. Igualmente, si se diera el caso en el que ambas opciones aparecen disponibles, la apariencia sería distinta a las anteriores, dando la posibilidad del empleo de ambas características. Así tendremos: **Apariencia Frases, Apariencia Colores y Apariencia completa**.
- Las funcionalidades de poder emplear el barrido y poder borrar la última palabra escrita, son opcionales. Pero en esta línea de productos se considerarán a partir de ahora como funcionalidades comunes. Consideramos que la posibilidad de disponer de la función de barrido es deseable para cualquier comunicador orientado a las personas con parálisis cerebral, cuya minusvalía limita su movilidad. De igual forma, el poder borrar la última palabra escrita, agiliza la comunicación mediante el comunicador de manera notable. Además, esta consideración supone una reducción en la complejidad de la línea de productos. Se ha considerado que se disponen de suficientes funcionalidades comunes, opcionales y alternativas, como para alcanzar los objetivos del proyecto.

De esta forma se ven los requisitos comunes, opcionales y alternativos que tendría nuestra línea de productos. A partir de éstos, se puede ya esbozar el diagrama de características resultante:

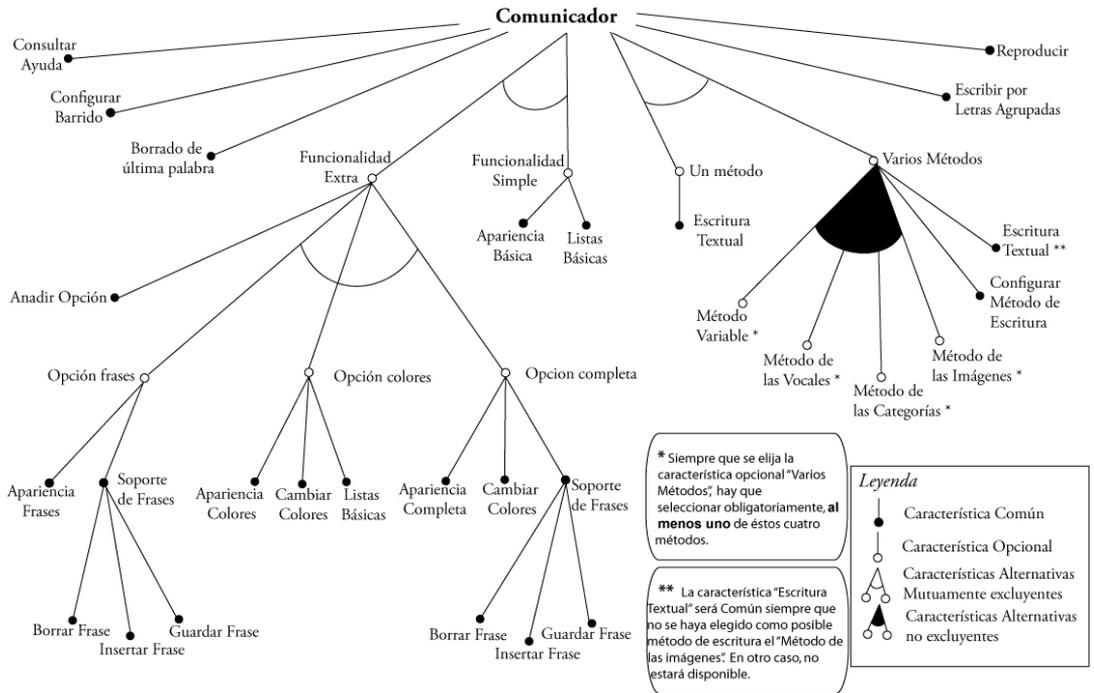


Figura 6.1: Diagrama de características.

A continuación, se explican más detenidamente algunas de las partes de este diagrama:

En la figura 6.2 se puede apreciar con detalle una parte del diagrama de características. A continuación se ofrece una breve descripción de las características que en él aparecen:

**Características comunes:**

Las siguientes características son consideradas comunes y, por tanto, estarán presentes en cualquier aplicación que se genere mediante la línea de productos:

**Consultar Ayuda:**

Atiende a la posibilidad de consultar la ayuda de la aplicación en todo momento.

**Configurar Barrido:**

Permite establecer la existencia o no del barrido y su el tiempo de intervalo.

**Borrado de última palabra:**

Permite borrar la última palabra que se haya escrito en el cuadro de texto del comunicador.

**Funcionalidad Extra – Funcionalidad Simple:**

Se dispone, en el diagrama de características, de dos características alternativas mutuamente excluyentes: Funcionalidad Extra y Funcionalidad Simple.

El ser alternativas mutuamente excluyentes significa que cada aplicación concreta que desarrollemos mediante la línea de productos que nos ocupa, dispondrá de una u otra característica pero nunca de ambas.

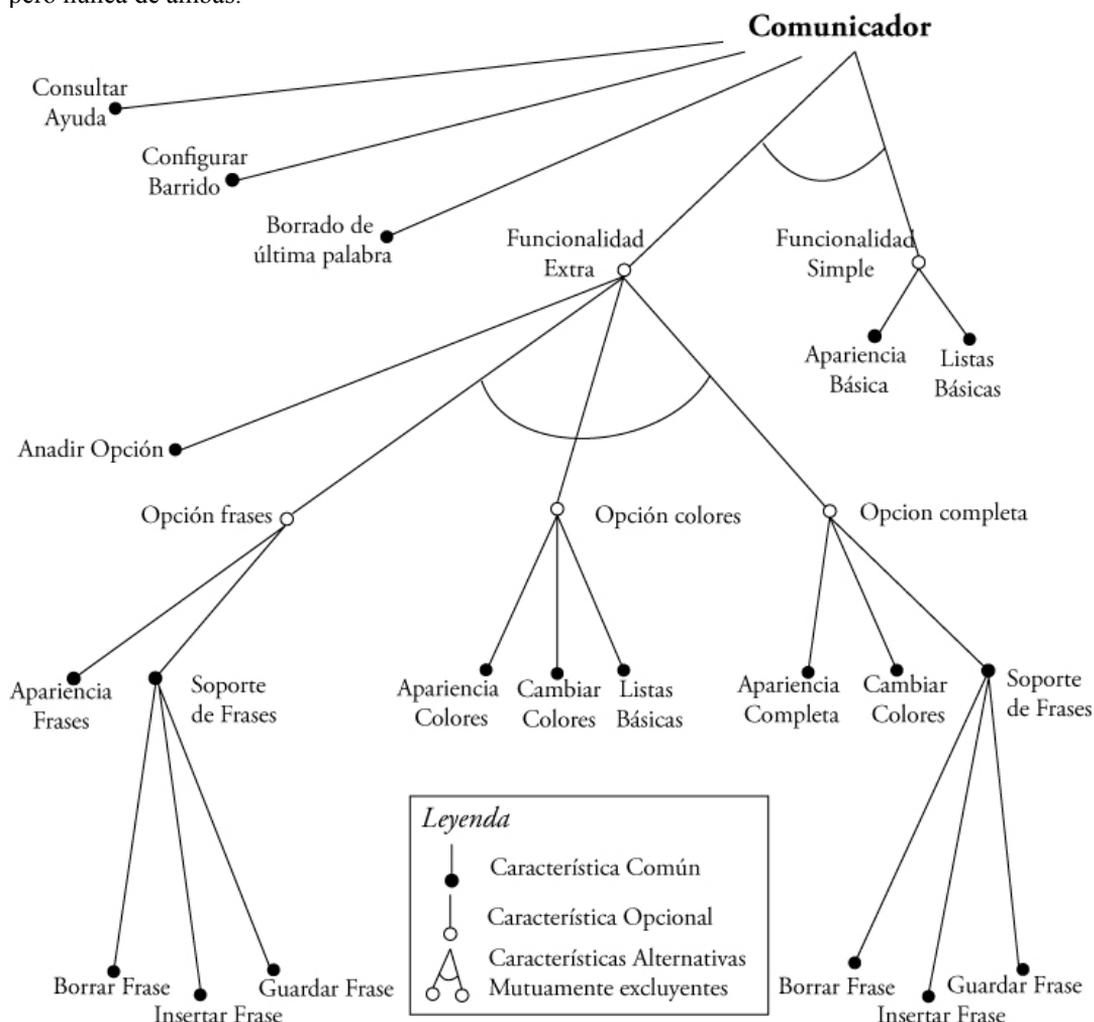


Figura 6.2: Detalle del diagrama de características. Funcionalidades.

**Funcionalidad Extra:** Recoge la posibilidad de que la aplicación final disponga de la opción de elegir colores, de gestionar y usar frases, o ambas. Por tanto, contamos con tres alternativas mutuamente excluyentes:

1. *Opción frases:* si queremos incluir la posibilidad de gestión de frases.
2. *Opción Colores:* si queremos incluir la posibilidad de cambiar de color los elementos de la aplicación.
3. *Opción Completa:* si queremos incluir ambas posibilidades.

Si se escoge *Opción frases* se necesitarán: una interfaz acorde con la incorporación de frases, esto es, *Apariencia Frases*; y una funcionalidad para el tratamiento de frases, que

se tendría con *Soporte de frases*. Así se permite borrar, insertar y guardar frases a través de sus correspondientes características.

Si se escoge *Opción Colores* se necesitarán: *Cambiar Colores*, para poder cambiar los colores de los elementos de la aplicación; *Apariencia Colores*, para integrar dicha opción en la interfaz de la aplicación; y *Listas básicas*, que incluye la versión original del elemento de diseño lista, necesaria para la aplicación.

Si se escoge *Opción Completa*, se incluirán las características anteriormente explicadas *Cambiar Colores* y *Soporte de frases*. Con esta opción, la interfaz de la aplicación tendrá que contemplar ambas posibilidades, por lo que se añade la característica *Apariencia completa*. En este caso y aunque se incluya el soporte para colores, no se añadirá la característica *Listas básicas*, puesto que al contener también la posibilidad de gestionar y utilizar frases, dichas listas se verán modificadas.

Asimismo, *Funcionalidad Extra* incluye una característica obligatoria: *Añadir Opción*, que es necesaria para el correcto funcionamiento de cualquiera de estas opciones cuando se seleccionen.

***Funcionalidad Simple:*** Al seleccionar esta característica excluyente, estamos descartando las posibilidades de cambiar el color de los elementos de la aplicación y de gestionar y usar frases. Se incluyen dos características obligatorias:

*Listas Básicas*, necesaria para el correcto funcionamiento de cualquier aplicación que no disponga de la característica *Soporte de Frases*.

*Apariencia Simple*, que prepara la interfaz básica de la aplicación sin ninguna funcionalidad añadida.

En la figura 6.3 se puede apreciar con detalle la otra parte del diagrama de características. A continuación, se ofrece una breve descripción de las características que en él aparecen:

### **Características comunes:**

Las siguientes características son consideradas comunes y, por tanto, estarán presentes en cualquier aplicación que se genere mediante la línea de productos:

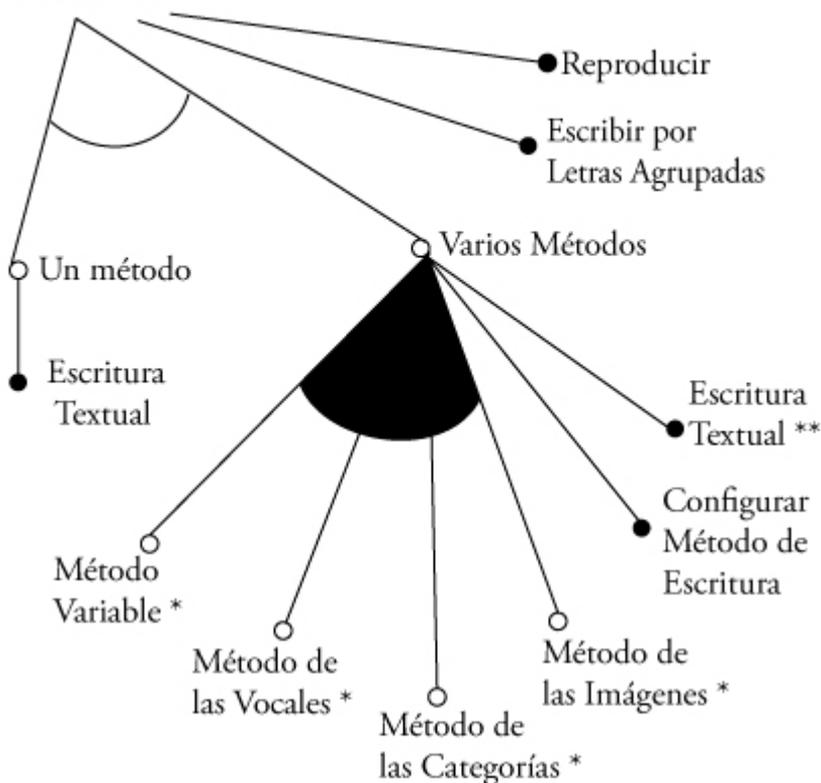
#### ***Reproducir:***

Permite la reproducción del texto escrito.

#### ***Escribir por Letras Agrupadas:***

Permite la escritura de texto en el comunicador, mediante el método de las letras agrupadas. Éste es el método de escritura por defecto, y aunque se escojan otras formas de escritura adicionales, el método de las letras agrupadas siempre estará disponible para su uso.

## Comunicador



\* Siempre que se elija la característica opcional "Varios Métodos", hay que seleccionar obligatoriamente, **al menos uno** de éstos cuatro métodos.

\*\* La característica "Escritura Textual" será Común siempre que no se haya elegido como posible método de escritura el "Método de las imágenes". En otro caso, no estará disponible.

### Leyenda

-  Característica Común
-  Característica Opcional
-  Características Alternativas Mutuamente excluyentes
-  Características Alternativas no excluyentes

Figura 6.3: Detalle del diagrama de características. Métodos de escritura.

### **Uno o Varios Métodos de Escritura:**

La figura 6.3 muestra otro caso de dos características alternativas mutuamente excluyentes: *Varios Métodos de Escritura* y *Un Método de Escritura*.

***Varios Métodos de Escritura:*** Consta de una característica obligatoria, *Configurar Método de Escritura*, que sirve para, como su nombre indica, poder pasar de uno a otro método de escritura dentro de la aplicación. También consta de cuatro métodos de escritura distintos, pudiendo seleccionarse de uno a cuatro métodos al mismo tiempo. Para los métodos basados en texto, tenemos la característica *Escritura Textual*, que es necesaria para su correcto funcionamiento.

***Un Método de Escritura:*** Al seleccionar esta característica no se añaden nuevos métodos de escritura a la aplicación y sólo se podrá escribir por el método de Letras Agrupadas. Al ser éste un método basado en texto, se añade la característica *Escritura Textual* como obligatoria.

## **6.3 Gestión de la variabilidad con el método de combinación de paquetes**

Existe un amplio consenso en el mundo del desarrollo de líneas de productos software, en torno a la necesidad de expresar la variabilidad mediante modelos orientados a features o características.

En el desarrollo de una línea de productos software, una vez que se obtienen los requisitos funcionales y no funcionales por parte del cliente, se acude al modelo de características y se seleccionan aquellas que son más apropiadas a los requisitos obtenidos. El subconjunto de características es el que generará toda o la mayor parte del código de la aplicación concreta, mediante un framework. Dicho framework orientado a objetos es la forma más común de expresar el diseño de la arquitectura de una línea de productos.

A la hora de desarrollar la línea de productos, nos encontramos con dos dificultades principales. En primer lugar, está el hecho de que los mecanismos comunes de modelado de variabilidad se utilizan para dos cosas diferentes: para el diseño de la línea de productos (que una aplicación permita una forma concreta de escritura o no) y para el diseño de una aplicación concreta (que un usuario emplee un método de escritura u otro). Es decir, el problema está en que el mismo mecanismo sirve para mostrar variaciones en tiempo de configuración y en tiempo de ejecución. El segundo problema tiene que ver con la gestión de la trazabilidad de las características hasta el código. Una característica opcional puede originar varios elementos en un modelo de diseño, y uno de estos elementos, puede ser compartido por varias características, lo que complica el modelo de manera notable.

Las soluciones más recientemente propuestas a estos problemas pasan por anotar o modificar los modelos estructurales, funcionales y dinámicos, teniendo que cambiar o extender el meta-modelo de UML mediante estereotipos.

El grupo GIRO ha estado realizando un estudio sobre los puntos débiles y fuertes de las distintas propuestas para el desarrollo de una línea de productos software. Tras este estudio, se pudo establecer un conjunto mínimo de requisitos que debe cumplir una técnica útil de representación y gestión de la variabilidad en el nivel de diseño de una línea de productos software:

1. Localizar en un solo punto del modelo de diseño, todas las variaciones que origina cada característica opcional, de forma que se mantenga una correspondencia uno a uno y se facilite la gestión de la trazabilidad.
2. Separar la variabilidad originada en el nivel de la línea de productos, de la variabilidad intrínseca en el nivel de las aplicaciones concretas, eliminando ambigüedades.
3. Mantener inalterado el meta-modelo de UML, para eliminar la barrera de entrada a este paradigma para cualquier desarrollador, además de permitir el uso de herramientas CASE convencionales.
4. Conectar con los modelos de implementación, para acercarnos al ideal de desarrollo sin costuras o “*seamless development*”, propugnado por el paradigma de orientación a objetos por desechado muchas veces por irrealizable.

El desarrollo de la línea de productos de este proyecto, se basa en el mecanismo de combinación de paquetes de UML 2 que cumple los requisitos arriba indicados.

El mecanismo de combinación de paquetes consiste fundamentalmente en añadir detalles de forma incremental. Según la especificación UML 2, éste consiste en una relación entre dos paquetes que indica que los contenidos de ambos se combinan. Es similar a la generalización y se utiliza cuando elementos en distintos paquetes tienen el mismo nombre y representan el mismo concepto. Dicho concepto se extiende incrementalmente en cada paquete añadido.

La filosofía general es que el elemento resultante tiene que ser al menos tan capaz como el original antes de la combinación.

Este mecanismo nos permite establecer una trazabilidad clara entre los modelos de características y los artefactos UML. La aplicación a nuestro problema consiste en establecer, en primer lugar y para cada modelo UML del diseño, un paquete base que recoge la parte común de la línea de productos. A este paquete base, se añade un paquete por cada característica opcional, de modo que queden localizadas en ese paquete todas las modificaciones necesarias en el modelo de diseño asociadas a la característica. El paquete añadido se conecta mediante la relación <<merge>> con su paquete base, en el punto preciso de la jerarquía de paquetes.

Cada punto de variación detectado en el modelo de características, hace que aparezca un paquete que se puede combinar o no en tiempo de desarrollo del producto, según la configuración de características elegida.

Aunque en este trabajo nos interesan, sobre todo, los diagramas de clases, el mecanismo se puede extender a cualquier diagrama de UML. En concreto, en el proyecto realizado, se aplicará al diagrama de clases y al de casos de uso.

## 6.4 Análisis de los casos de uso

En el análisis orientado a objetos para los sistemas simples, los casos de uso determinan los requisitos funcionales del sistema: representan las funciones que el sistema va a poder ejecutar. También sirven para esto en las líneas de productos.

Al aplicar el método de modelado de los casos de uso, para especificar los requisitos funcionales de las líneas de productos software, es necesario definir distintos tipos de casos de uso:

- **Casos de uso comunes:** son necesarios en todos los miembros de la línea de productos.
- **Casos de uso opcionales:** son necesarios en sólo algunos de los miembros de la línea de productos.
- **Casos de uso alternativos:** donde diferentes miembros de la línea de productos necesitan diferentes casos de uso. Éstos son, normalmente, exclusivos.

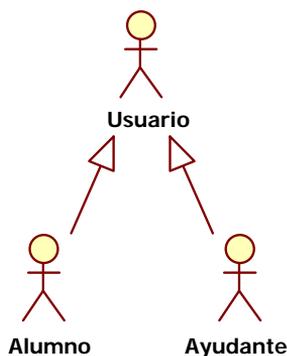
La distinción entre estos tres tipos de casos de uso es, a menudo, llamado análisis de variabilidad.

### 6.4.1 Identificación de los actores

Dependiendo del proyecto de fin de carrera en el que nos fijemos, encontramos unos actores u otros:

- En el desarrollo de un componente teclado (2006), tenemos dos actores: discapacitado y ayudante.
- En el comunicador basado en barrido para PDA (2005), tenemos tres actores: alumno, profesor y administrador.
- En el desarrollo de un comunicador java para PDA (2006), tenemos un único actor: alumno.
- En el teclado sobre una PDA para personas con parálisis cerebral (2004), tenemos también un único actor: alumno.

Todos estos actores tienen roles en común. Por tanto, se considera que tienen que ser generalizados, de modo que la parte común es caracterizada por un actor generalizado.



<b>ACT-0001</b>	<b>Usuario</b>
<b>Descripción</b>	Este actor es la persona que va a usar la aplicación.

Tabla 6.2: Descripción del actor del sistema.

#### 6.4.2 Diagrama de casos de uso

Una vez recopilados los requisitos funcionales de la línea de productos, podemos agrupar los distintos casos de uso obtenidos según sea su categoría: comunes, opcionales o alternativos.

COMUNES	OPCIONALES	ALTERNATIVOS
Consultar Ayuda	Seleccionar método de escritura	
Reproducir	Seleccionar colores	
Salir de la aplicación	Escribir texto categorías	
Escribir texto letras agrupadas	Escribir texto vocales	
Escribir texto	Escribir imágenes	
Borrar palabra	Escribir modo variable	
Seleccionar	Insertar frase	
Seleccionar barrido	Guardar frase	
	Borrar frase	

Tabla 6.3: Clasificación de los casos de uso según su categoría.

Dentro del caso de uso Seleccionar, tenemos que distinguir dos posibilidades: Seleccionar barrido y Seleccionar método de escritura. La primera es la que permite modificar el tiempo de barrido y si dicha función queda activada o desactivada. Por defecto vendrá activada, por lo que es un caso de uso común. La segunda, es la que permite seleccionar el método de escritura que quiere utilizar de entre los disponibles. Estos dos casos de uso, extienden la funcionalidad de Seleccionar.

Lo mismo ocurre con el caso de uso Escribir Texto. Este caso de uso permite la escritura mediante texto en la aplicación. Como ya hemos dicho anteriormente, hay tres tipos de escritura que son de texto: el método de letras agrupadas, categorías y vocales. Por tanto, estos tres casos de uso, extienden al caso de uso Escribir texto.

Con esto se puede conseguir el diagrama de casos de uso, el cual documentará el comportamiento de un sistema desde el punto de vista del usuario.

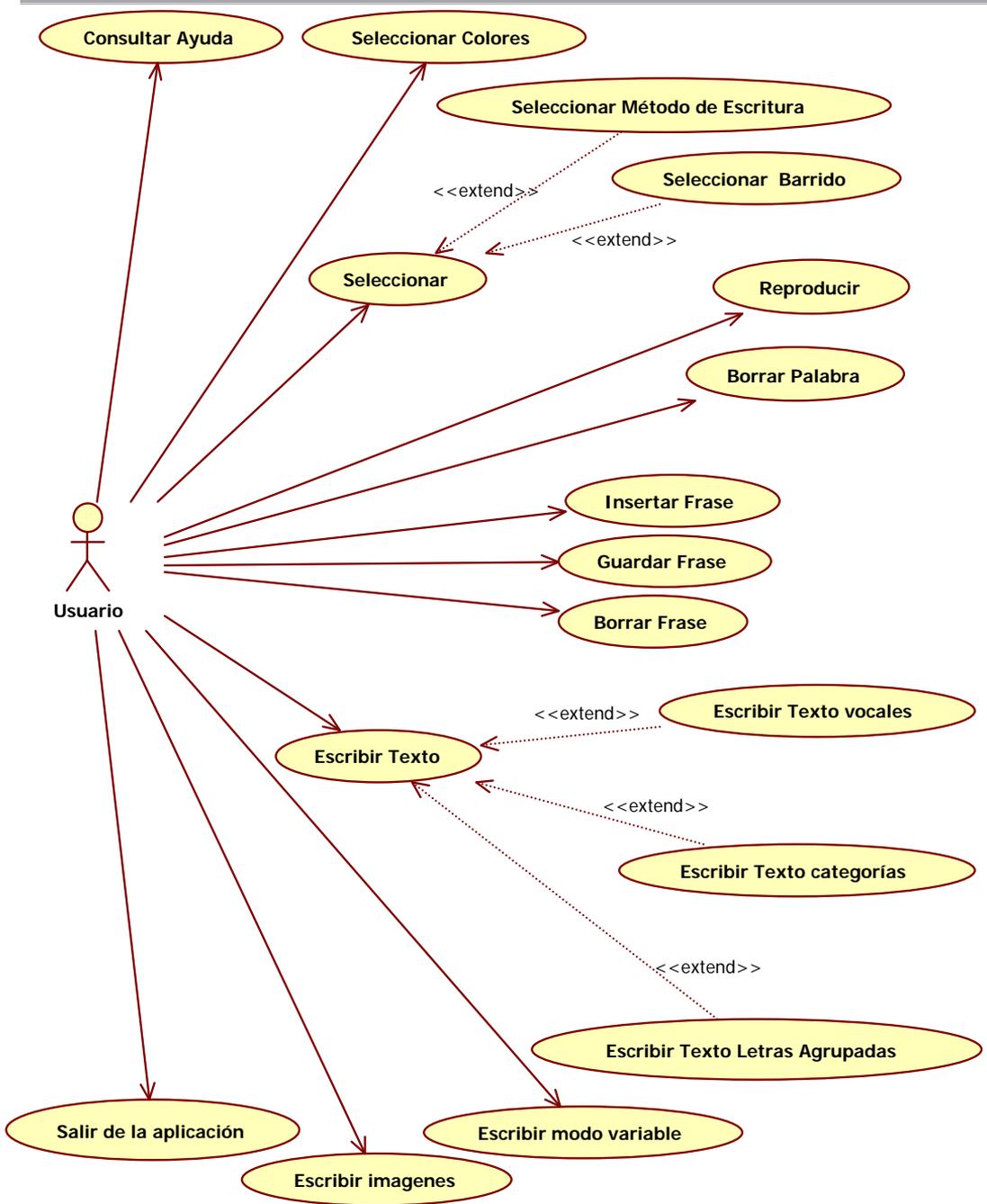


Figura 6.4: Diagrama de casos de uso.

Como se puede ver en la figura 6.4, se muestra un diagrama de casos de uso que interpreta el sistema completo. Es decir, no hay distinción entre los casos de uso opcionales y comunes.

### 6.4.3 Estructura de paquetes de los casos de uso

Siguiendo el diagrama de características anteriormente explicado en la figura 6.1, se agrupan en un único paquete al que llamaremos caso base, todos los casos de uso comunes de la línea de productos. En él aparece el conjunto de todas las características obligatorias y deberá estar incluido siempre en cualquier aplicación que se derive de esta línea.

Asimismo se dispone de varios paquetes opcionales, que se combinarán o no con el paquete Caso Base. Como se puede ver en la figura 6.5, estos paquetes opcionales serían: varios métodos, colores y soporte de frases. Del mismo modo, los paquetes correspondientes a los casos de uso relacionados con los métodos de escritura, se pueden o no combinar con el paquete varios métodos.

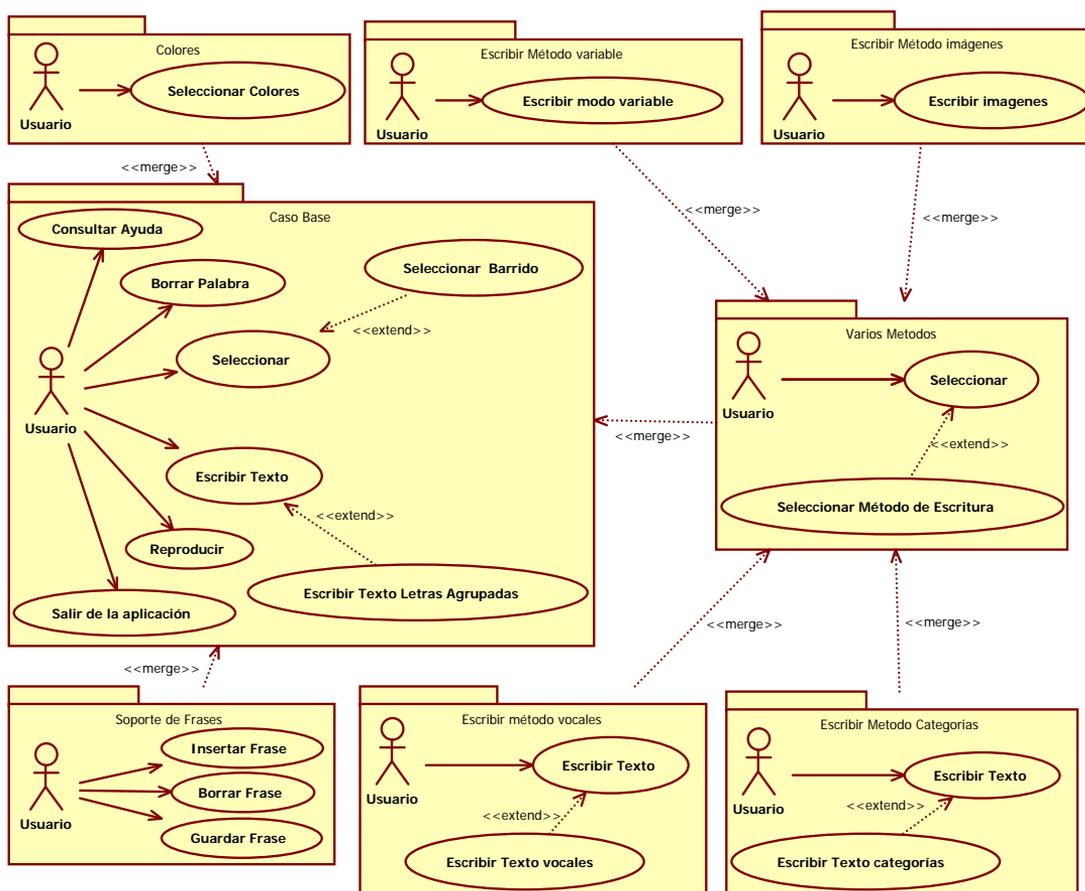


Figura 6.5: Estructura de paquetes de los casos de uso.

En la figura 6.6 se puede ver un ejemplo de la combinación de paquetes. En este caso, la aplicación resultante tendría la posibilidad de escribir mediante el método de categorías y letras agrupadas, así como de cambiar los colores por defecto de la aplicación. Para ello, habría que agregar al paquete Varios Métodos el paquete Método Categorías, para que se pueda escribir mediante esa forma de escritura. Al paquete base habría que agregarle el paquete Varios Métodos para que se pueda alternar entre los dos métodos de escritura disponibles y el paquete Opción Cambiar Colores, para poder modificar el color de los elementos que se deseen.

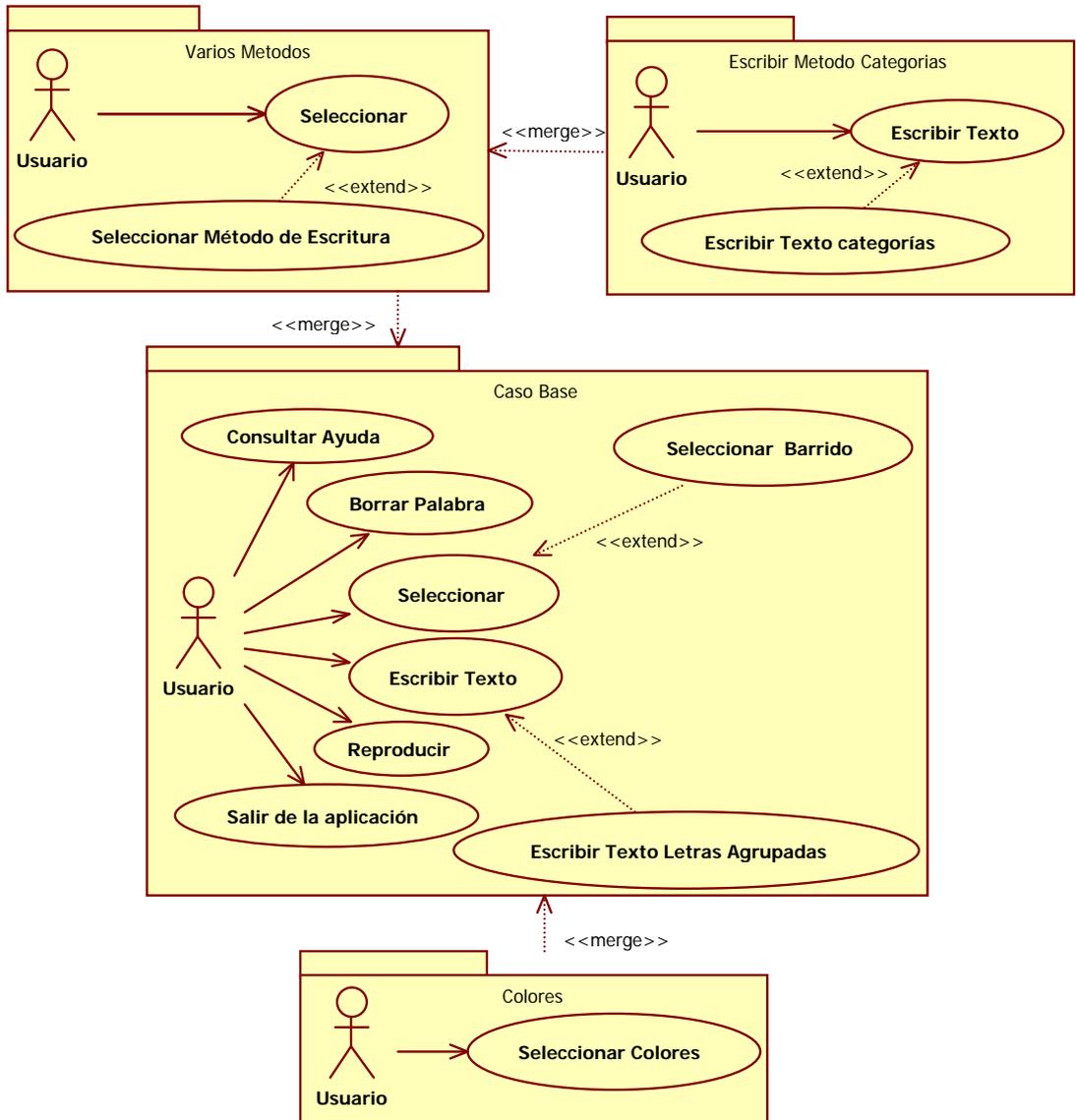


Figura 6.6: Ejemplo de combinación de paquetes.

#### 6.4.4 Presentación de casos de uso

A continuación se describirán, de forma general, los casos de uso representados en la figura 6.4. En esta fase no se entrará en detalles relacionados con la parte de diseño, como puede ser la interfaz mostrada al usuario para realizar una función. Ese nivel de detalle se mostrará en la especificación de los casos de uso donde se hablará de la apariencia del teclado. También se representará los diagramas de secuencia de cada caso de uso, mostrando el orden de las llamadas del sistema.

C-0001	Consultar ayuda	
<b>Descripción</b>	<i>El sistema deberá permitir que el usuario pueda consultar la ayuda sobre el uso de determinadas funciones de la aplicación en cualquier momento.</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) solicita consultar la ayuda.
	2	El sistema muestra los temas de ayuda.
	3	El actor usuario (Act-001) selecciona el tema sobre el que desea recibir la ayuda.
4	El sistema muestra la ayuda relacionada con el tema seleccionado.	

Tabla 6.4: Descripción del caso de uso: Consultar ayuda.

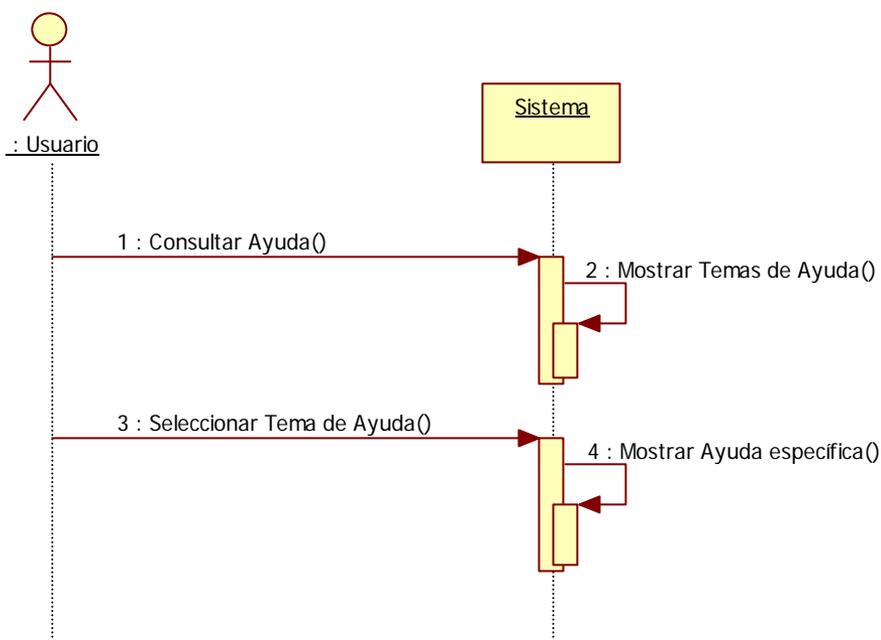


Figura 6.7: Diagrama de secuencia del caso de uso: Consultar ayuda.

C-0002	Seleccionar colores	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en este caso de uso, cuando se deseen cambiar los colores de apariencia.</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) solicita cambiar poder cambiar los colores de los elementos de la aplicación.
	2	El sistema muestra los colores disponibles.
	3	El actor usuario (Act-001) elige el color.
4	El sistema realiza los cambios de color solicitados.	

Tabla 6.5: Descripción del caso de uso: Seleccionar colores.

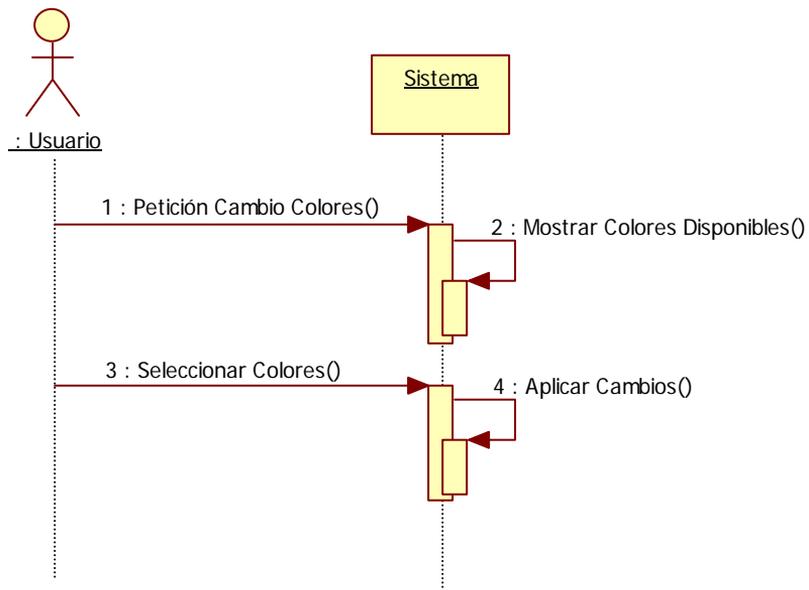


Figura 6.8: Diagrama de secuencia del caso de uso: Seleccionar colores.

C-0003		Seleccionar	
<b>Descripción</b>	<i>El sistema deberá permitir que el usuario seleccione y configure el barrido, así como la otra posible opción de selección (método de escritura)</i>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El actor usuario (Act-001) solicita cambiar la configuración.	
	2	El sistema muestra la opción de configurar la aplicación.	
	3	El actor usuario (Act-001) selecciona dicha opción.	
	4	El sistema muestra las configuraciones disponibles.	
<b>Extensiones</b>	<i>C-0004: Seleccionar Barrido.                      C-0005: Seleccionar Método de Escritura.</i>		

Tabla 6.6: Descripción del caso de uso: Seleccionar.

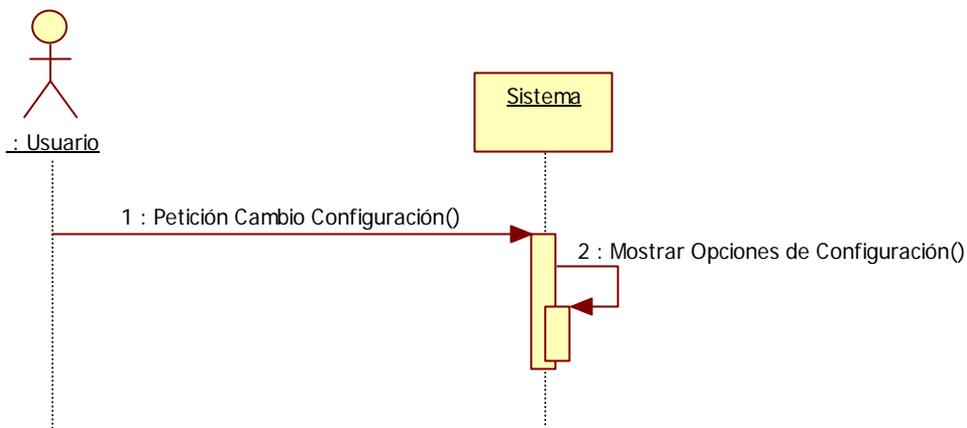


Figura 6.9: Diagrama de secuencia del caso de uso: Seleccionar.

<b>C-0004</b>	<b>Seleccionar barrido</b>	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en este caso de uso, cuando se desee activar o desactivar el barrido, así como configurar su tiempo.</i>	
<b>Precondición</b>	<i>En el método de escritura basado en imágenes, no está permitido trabajar sin barrido porque éste es necesario para el paso automático de pantallas.</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor Usuario (Act-001) selecciona las opciones de configuración de barrido que desee.
	2	El sistema actualizará dichos cambios.
<b>Caso de uso extendido</b>	<i>C-0003 : Seleccionar</i>	

Tabla 6.7: Descripción del caso de uso: Seleccionar barrido.

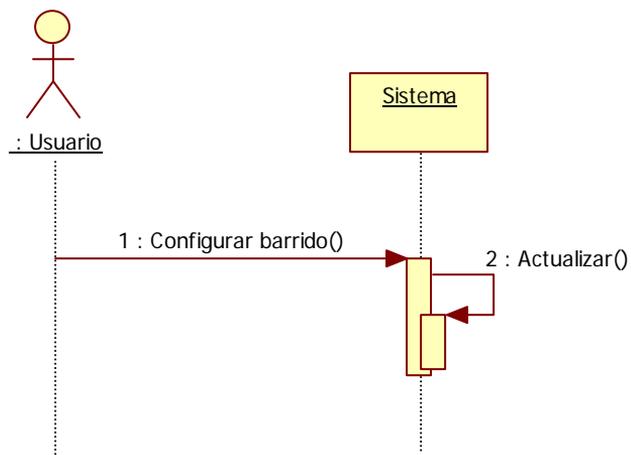


Figura 6.10: Diagrama de secuencia del caso de uso: Seleccionar barrido.

<b>C-0005</b>	<b>Seleccionar método de escritura</b>	
<b>Descripción</b>	<i>El sistema permitirá seleccionar el método de escritura a utilizar, de entre todos los disponibles.</i>	
<b>Precondición</b>	<i>El sistema deberá tener al menos un método de escritura disponible.</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) selecciona un método de escritura.
	2	El sistema actualiza la aplicación con el nuevo método de escritura.
<b>Caso de uso extendido</b>	<i>C-0003 : Seleccionar</i>	

Tabla 6.8: Descripción del caso de uso: Seleccionar método de escritura.

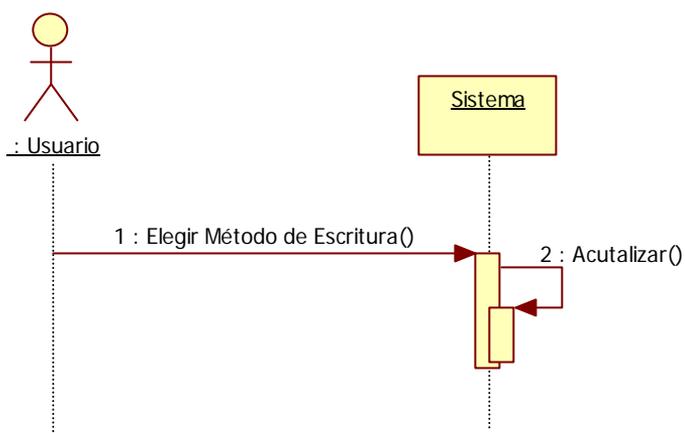


Figura 6.11: Diagrama de secuencia del caso de uso: Seleccionar método de escritura.

C-0006		Borrar palabra
Descripción	<i>El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario desee borrar la última palabra del texto escrito hasta el momento.</i>	
Precondición	<i>El campo de texto debe tener algún texto para poder borrarlo.</i>	
Secuencia normal	Paso	Acción
	1	El actor usuario (Act-001) selecciona la opción de borrar palabra.
	2	El sistema borra la última palabra del cuadro de texto.

Tabla 6.9: Descripción del caso de uso: Borrar palabra.

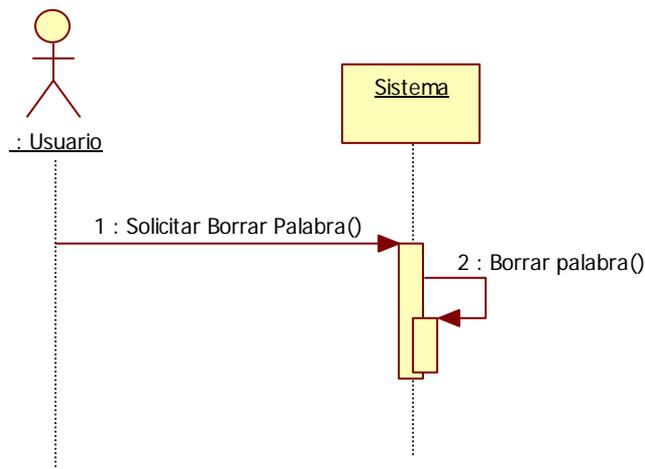


Figura 6.12: Diagrama de secuencia del caso de uso: Borrar palabra.

C-0007	Insertar frase	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiera insertar frases en el texto escrito hasta el momento.</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) solicita ver las frases disponibles.
	2	El sistema muestra las frases que tiene almacenadas.
	3	El actor usuario (Act-001) escoge la frase con la que quiere comunicarse y selecciona la opción de <i>insertar</i> .
4	El sistema actualiza el texto.	
<b>Postcondición</b>	<i>Texto actual más la frase insertada.</i>	

Tabla 6.10: Descripción del caso de uso: Insertar frase.

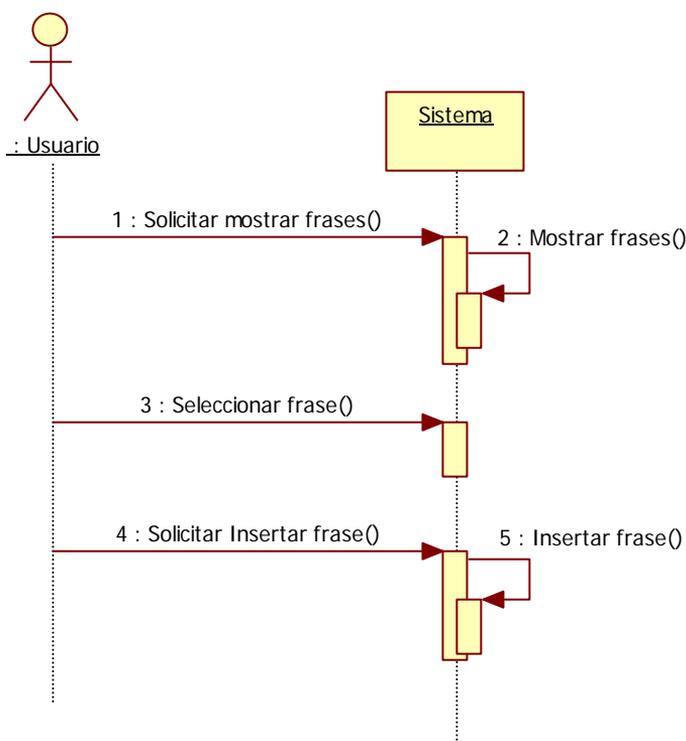


Figura 6.13: Diagrama de secuencia del caso de uso: Insertar frase.

C-0008		Guardar frase
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiera almacenar en la aplicación las frases que desee para su posterior uso.</i>	
<b>Precondición</b>	<i>Hay almacenadas n frases (n&gt;=0) El campo de texto debe contener algún texto escrito.</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) solicita ver las frases disponibles.
	2	El sistema muestra las frases que tiene almacenadas.
	3	El actor usuario (Act-001) solicita guardar el texto como frase.
4	El sistema actualiza la lista de frases.	
<b>Postcondición</b>	<i>Hay almacenadas n+1 frases (n&gt;=0).</i>	

Tabla 6.11: Descripción del caso de uso: Guardar frase.

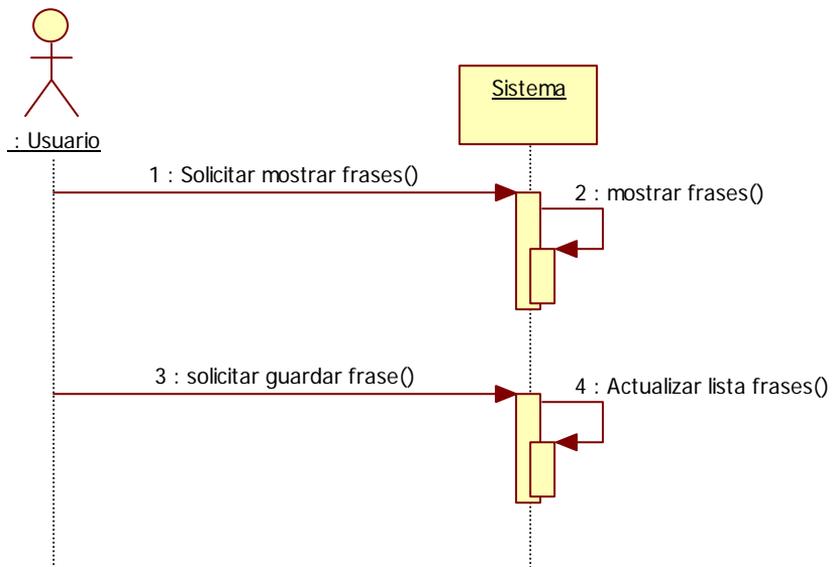


Figura 6.14: Diagrama de secuencia del caso de uso: Guardar frase.

C-0009	Borrar frase	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiera borrar alguna frase almacenada.</i>	
<b>Precondición</b>	<i>Hay almacenadas m frases (m &gt;= 1)</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) solicita ver las frases disponibles.
	2	El sistema muestra la lista de frases de las que se dispone.
	3	El actor usuario (Act-001) escoge la frase y selecciona la opción <i>borrar</i> .
4	El sistema actualiza la lista de frases.	
<b>Postcondición</b>	<i>Hay almacenadas m-1 frases (m &gt;= 1)</i>	

Tabla 6.12: Descripción del caso de uso: Borrar frase.

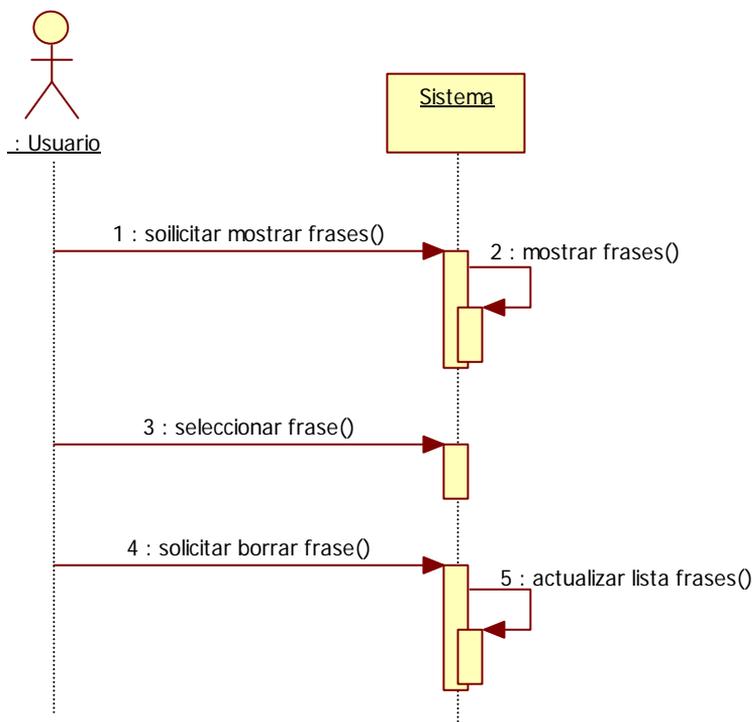


Figura 6.15: Diagrama de secuencia del caso de uso: Borrar frase.

C-0010	Reproducir	
Descripción	<i>El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiera reproducir un texto escrito.</i>	
Precondición	<i>El campo de texto debe tener algún texto para poder reproducirlo.</i>	
Secuencia normal	Paso	Acción
	1	El actor usuario (Act-001) solicita reproducir el texto escrito.
	2	El sistema reproduce el texto correspondiente.

Tabla 6.13: Descripción del caso de uso: Reproducir.

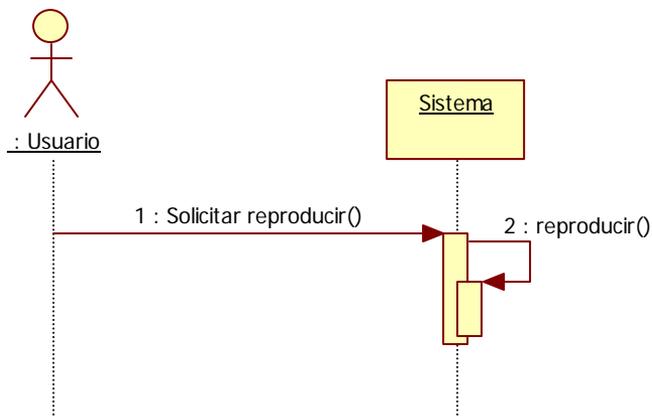


Figura 6.16: Diagrama de secuencia del caso de uso: Reproducir.

C-0011	Escribir texto	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe, cuando se desee escribir mediante caracteres tipográficos. La secuencia se completará con los casos de uso extendidos a partir de este mismo caso de uso.</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) solicita escribir.
	2	El sistema muestra la configuración de escritura correspondiente para que el usuario pueda escribir.
<b>Extensiones</b>	<i>C-0012 : Escribir Texto Categorías</i> <i>C-0013: Escribir Texto Vocales</i> <i>C-0014: Escribir Texto Letras Agrupadas</i>	

Tabla 6.14: Descripción del caso de uso: Escribir texto.

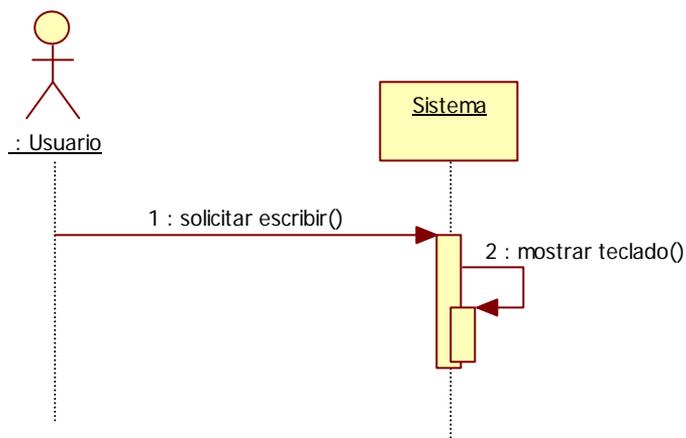


Figura 6.17: Diagrama de secuencia del caso de uso: Escribir texto.

C-0012		Escribir texto Categorías	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en este caso de uso, cuando se desee añadir un texto por el método de escritura “categorías”.</i>		
<b>Precondición</b>	<i>El método de escritura debe ser el de “categorías”.</i>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El actor Usuario (Act-001) selecciona el conjunto de letras donde está la que desea escribir.	
	2	El sistema muestra un conjunto reducido de letras.	
	3	El actor Usuario (Act-001) selecciona el carácter que desea escribir.	
4	El sistema actualiza el texto.		
<b>Caso de uso extendido</b>	<i>C-0011 : Escribir</i>		

Tabla 6.15: Descripción del caso de uso: Escribir texto Categorías.

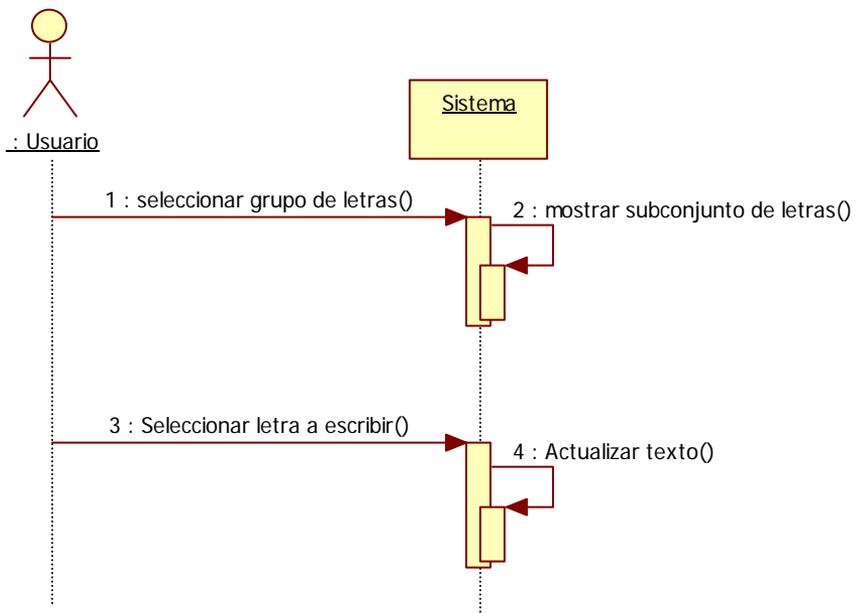


Figura 6.18: Diagrama de secuencia del caso de uso: Escribir texto Categorías.

C-0013	Escribir texto Vocales	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en este caso de uso, cuando se desee añadir un texto por método de escritura “método de las vocales”</i>	
<b>Precondición</b>	<i>El método de escritura debe ser el de “método de las vocales”</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor Usuario (Act-001) selecciona el conjunto de consonantes donde está la que desea escribir, o selecciona la vocal que desea escribir.
	2a	Si el actor Usuario (Act-001) ha seleccionado un grupo de consonantes, el sistema muestra un conjunto reducido de éstas.
	2b	Si el actor Usuario (Act-001) ha seleccionado una vocal, ir al paso 4.
	3	El actor Usuario (Act-001) selecciona la consonante que desea escribir.
	4	El sistema actualiza el texto.
<b>Caso de uso extendido</b>	<i>C-0011 : Escribir</i>	

Tabla 6.16: Descripción del caso de uso: Escribir texto Vocales.

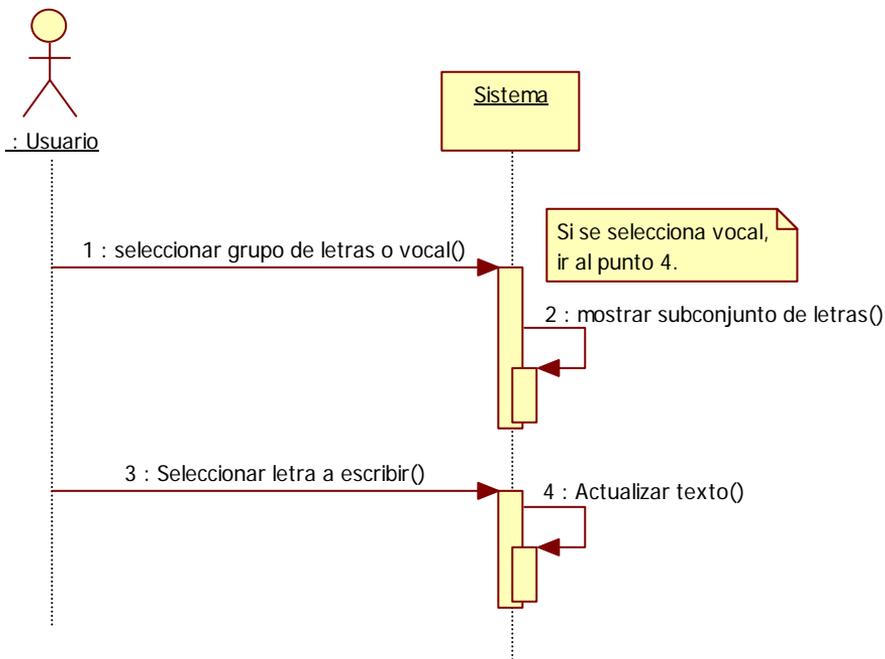


Figura 6.19: Diagrama de secuencia del caso de uso: Escribir texto Vocales

<b>C-0014</b>	<b>Escribir texto Letras agrupadas</b>	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en este caso de uso, cuando se desee añadir un texto por el método de escritura “letras agrupadas”.</i>	
<b>Precondición</b>	<i>El método de escritura debe ser el de “letras agrupadas”</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor Usuario (Act-001) selecciona el conjunto de letras donde está la que desea escribir.
	2	El sistema muestra un conjunto reducido de letras.
	3	El actor Usuario (Act-001) selecciona el carácter que desea escribir.
4	El sistema actualiza el texto.	
<b>Caso de uso extendido</b>	<i>C-0011 : Escribir</i>	

Tabla 6.17: Descripción del caso de uso: Escribir texto Letras agrupadas.

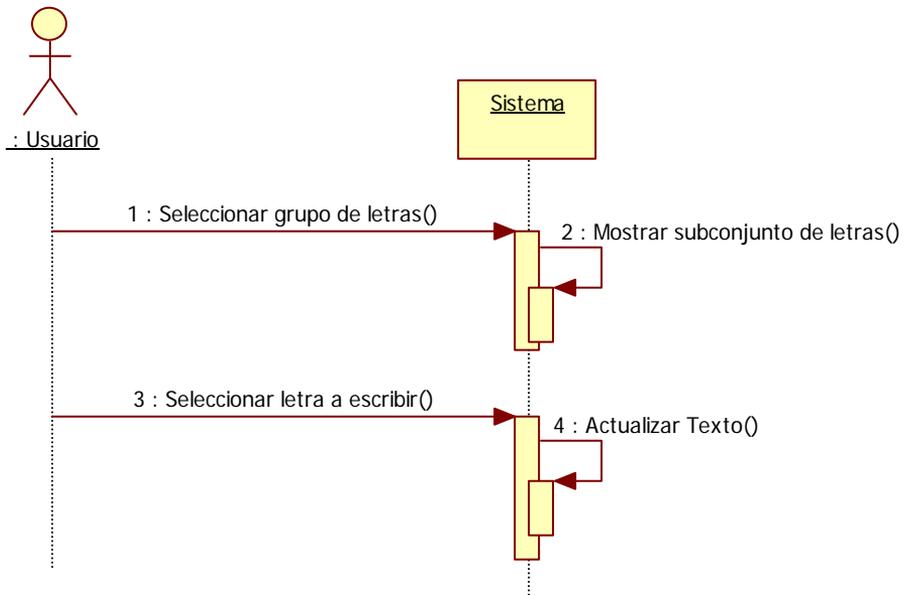


Figura 6.20: Diagrama de secuencia del caso de uso: Escribir texto Letras agrupadas.

C-0015		Escribir método variable	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en este caso de uso, cuando se desee añadir un texto por método de escritura “método variable”</i>		
<b>Precondición</b>	<i>El método de escritura debe ser el de “método variable”</i>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El actor Usuario (Act-001) selecciona los caracteres que desea escribir.	
	2	El sistema actualiza el texto.	

Tabla 6.18: Descripción del caso de uso: Escribir método variable.

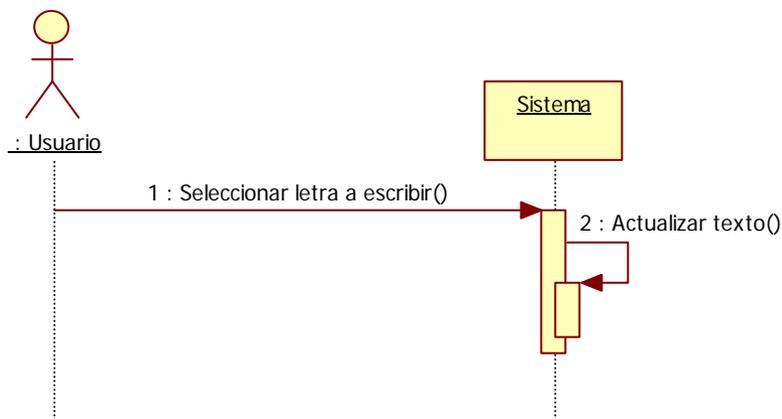


Figura 6.21: Diagrama de secuencia del caso de uso: Escribir método variable.

C-0016		Escribir Imágenes	
<b>Descripción</b>		<i>El sistema deberá comportarse tal como se describe, cuando se desee añadir caracteres asociados a una imagen.</i>	
<b>Precondición</b>		<i>El método de escritura debe ser el de "Imágenes"</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El actor Usuario (Act-001) selecciona el grupo de imágenes que se corresponde con el tipo de palabra que desea escribir.	
	2	El sistema muestra el subconjunto de imágenes.	
	3	El actor Usuario (Act-001) selecciona la imagen que representa la palabra que desea escribir.	
	4	El sistema actualiza el texto.	

Tabla 6.19: Descripción del caso de uso: Escribir Imágenes.

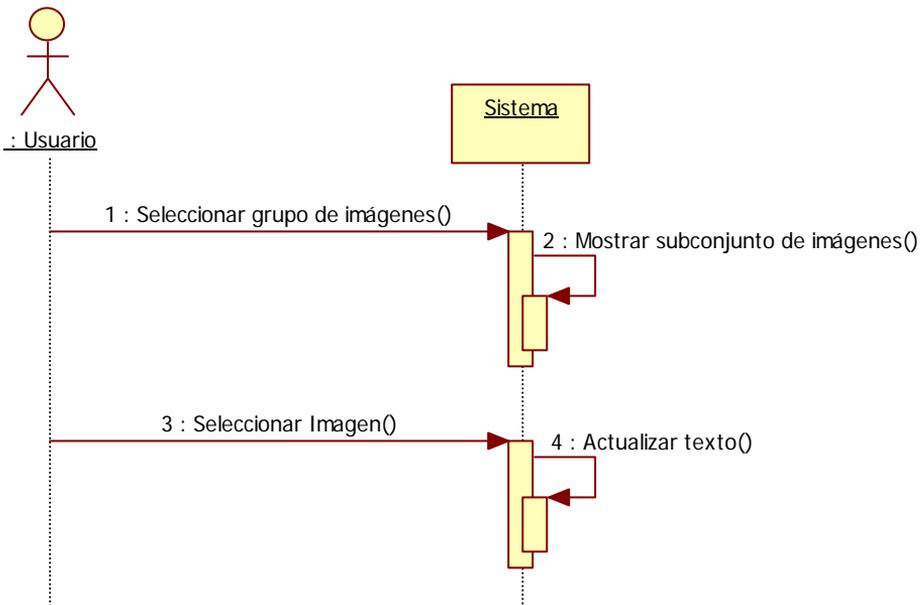


Figura 6.22: Diagrama de secuencia del caso de uso: Escribir Imágenes.

C-0017		Salir de la aplicación
Descripción	<i>El sistema deberá permitir que el usuario pueda salir de la aplicación.</i>	
Secuencia normal	Paso	Acción
	1	El actor usuario (Act-001) selecciona el Salir.
	2	El sistema termina la aplicación.

Tabla 6.20: Descripción del caso de uso: Salir de la aplicación.

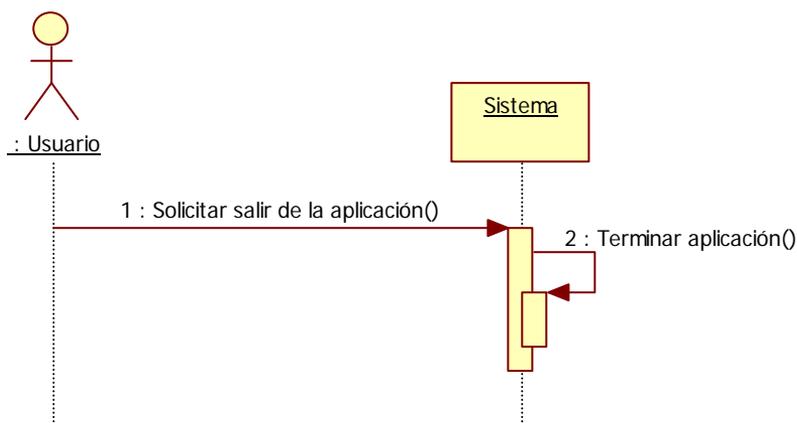


Figura 6.23: Diagrama de secuencia del caso de uso: Salir de la aplicación.

## 6.5 Diagrama inicial de clases

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones (incluyendo herencia, agregación, asociación, etc.). Los diagramas de clase son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido y describe los tipos de objetos en el sistema, los atributos y operaciones de las clases.

A continuación se muestra el diagrama inicial de clases, representando las mismas y sus relaciones. En el capítulo 7, “Diseño de la línea de productos”, se representará de manera más específica y se explicará con más detalle, así como las clases que lo componen.

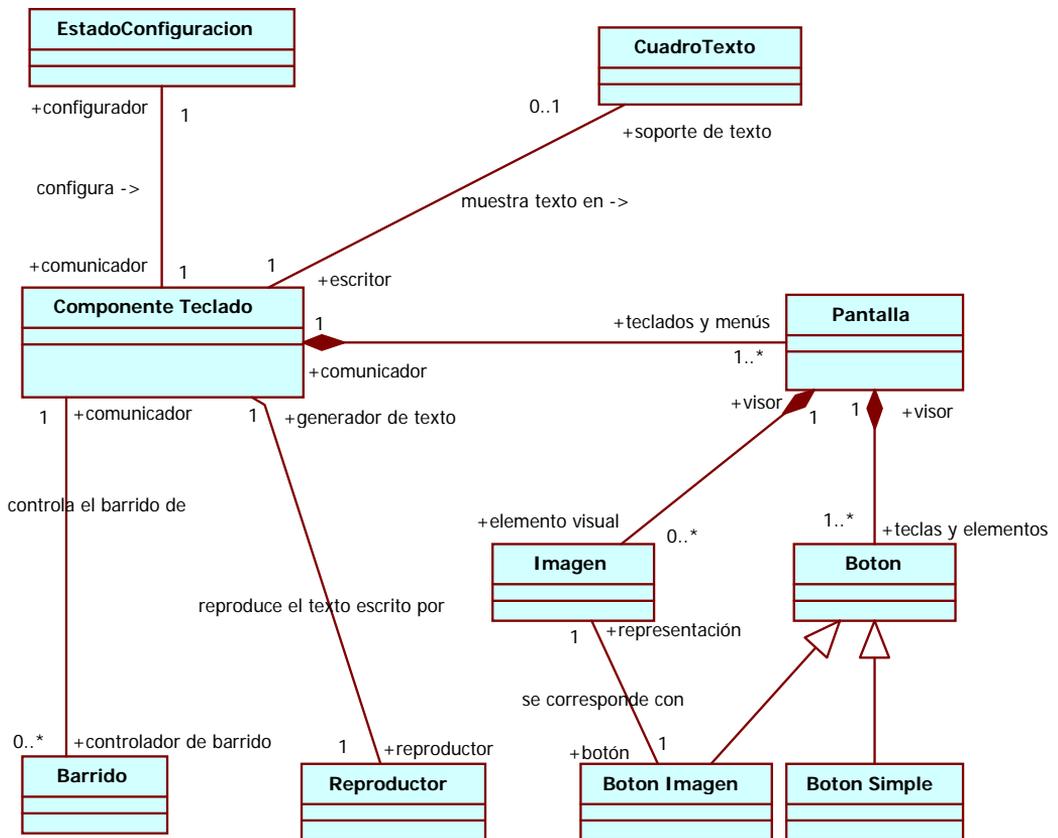


Figura 6.24: Diagrama inicial de clases.

### 6.5.1 Estructura de paquetes inicial de las clases

A continuación, en la figura 6.25, se presenta el diagrama de clases inicial empaquetado. Los paquetes se relacionan entre sí mediante el mecanismo de combinación de paquetes o *package merge*, del cual se ha hablado en el punto 6.3.

En el capítulo 7, “Diseño de la línea de productos”, se representará de manera más específica y se explicará con más detalle, así como las clases y paquetes que lo componen.

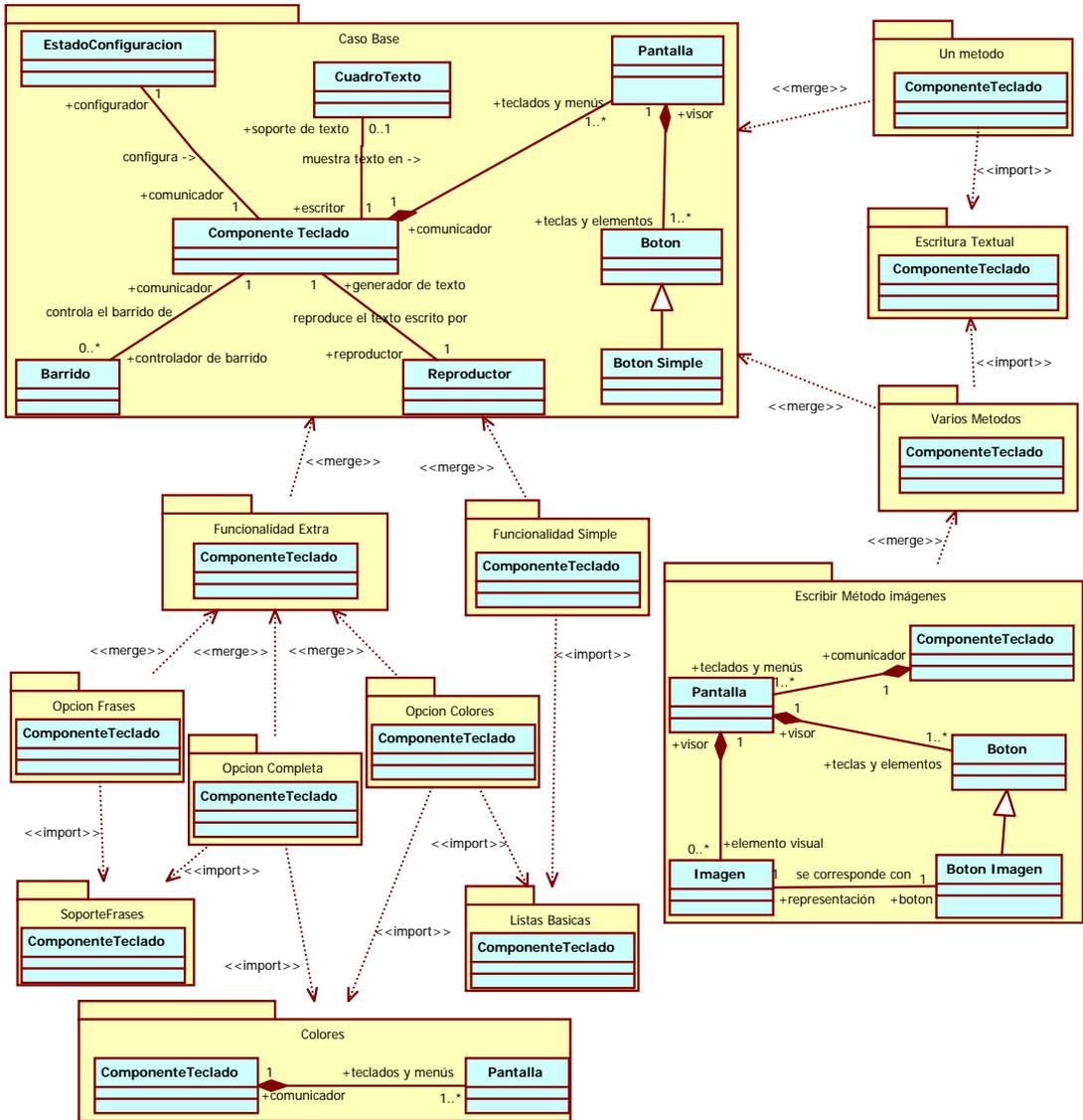


Figura 6.25: Diagrama inicial de clases empaquetado.

## Descripción de las clases

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

CLASE	DESCRIPCIÓN
ComponenteTeclado	Esta clase será la 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.
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.
CuadroTexto	Clase encargada de almacenar y mostrar el texto escrito por el usuario.
Barrido	Clase encargada de controlar el tiempo que cada botón, de una determinada pantalla, está señalado a la espera de una pulsación del usuario.
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.
Boton	Almacenará la información (posición, tamaño, apariencia y funcionalidad) de los botones que formarán el teclado.
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.
BotonSimple	Almacenará la información relativa a los botones de caracteres que aparecerán en las pantallas del teclado.
Imagen	Almacenará información (imagen normal, imagen con foco de barrido y texto a escribir por su botón asociado) sobre las imágenes que aparecerán en las pantallas cuando se esté trabajando con alguna configuración de imágenes.
Reproductor	Clase encargada de la reproducción del texto escrito.

Tabla 6.21: Descripción de las clases.



## Capítulo 7

# DISEÑO DE LA LÍNEA DE PRODUCTOS

### 7.1 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 del desarrollo de la línea de productos, se hará una ampliación de la especificación de los casos de uso que han sido descritos anteriormente. Esto se debe a que los proyectos de fin de carrera estudiados, contienen una fuerte interacción entre usuario y aplicación, afectando en cada paso al diseño de la misma. Además, se representará un diagrama de secuencia por cada caso de uso particular, mostrando el orden de las llamadas del sistema en funcionamiento.

Se presentará el diagrama de clases definitivo y organizado en paquetes. Para finalizar, se muestra la tabla 6.21 actualizada con las clases que intervienen en el sistema. También se hará una descripción más detallada de cada una de estas clases.

## 7.2 Especificación de casos de uso

C-0001	Consultar ayuda	
<b>Descripción</b>	<i>El sistema deberá permitir, que el usuario pueda consultar la ayuda sobre el uso de determinadas funciones de la aplicación.</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) selecciona el botón “MENU”.
	2	El sistema muestra una nueva pantalla (pantalla de Menú) compuesta por un conjunto de botones.
	3	El actor usuario (Act-001) selecciona el botón con el texto “AYUDA”.
	4	El sistema mostrará los temas sobre los que puede ofrecer ayuda.
	5	El actor usuario (Act-001) seleccionará el tema que desee.
6	El sistema ofrecerá la ayuda correspondiente al tema seleccionado	

Tabla 7.1: Descripción del caso de uso: Consultar ayuda.

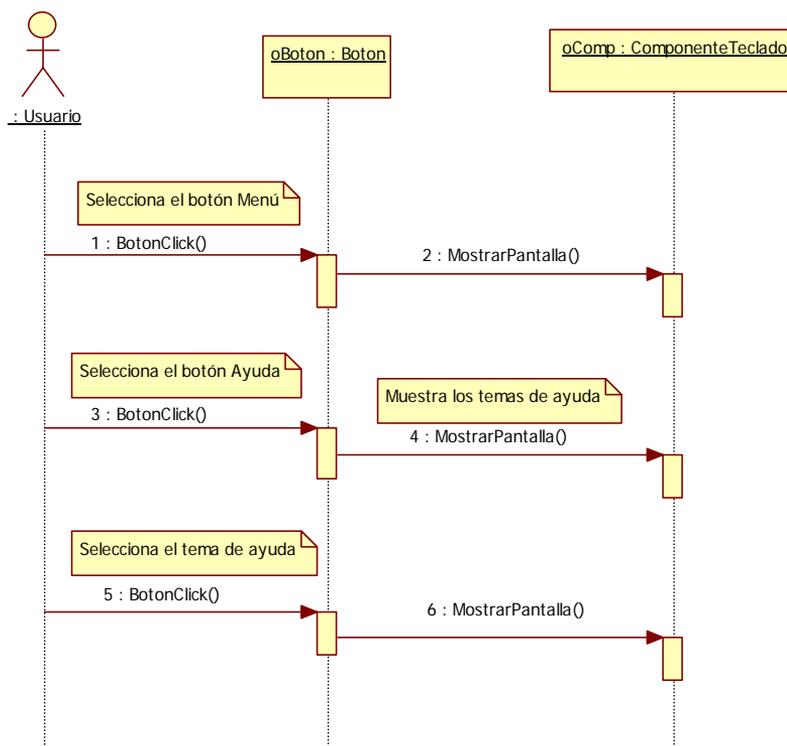
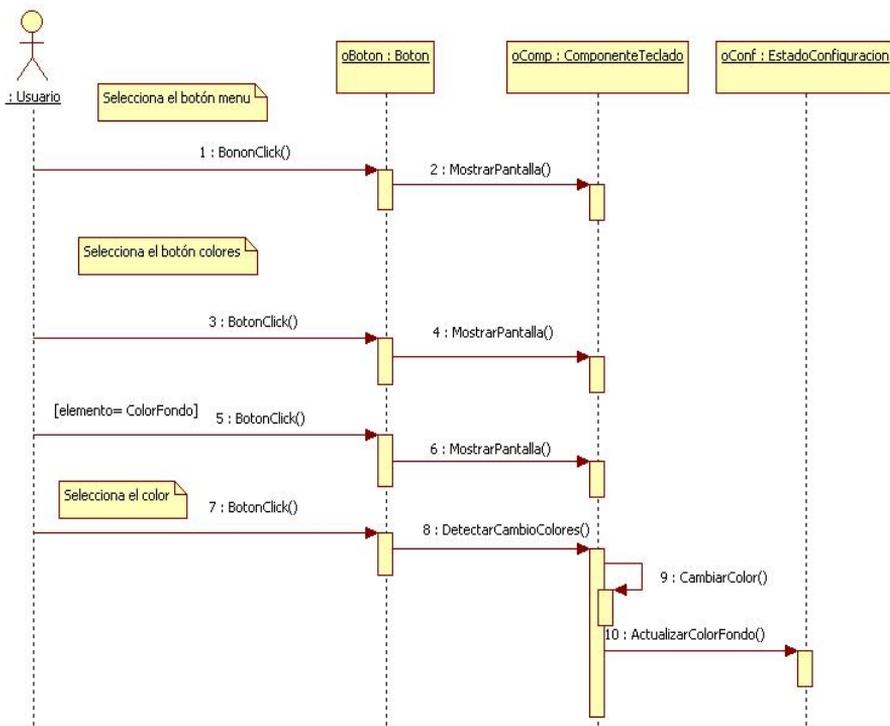


Figura 7.1: Diagrama de secuencia del caso de uso Consultar ayuda.

C-0002	Seleccionar colores	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en este caso de uso, cuando se deseen cambiar los colores de apariencia.</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) selecciona el botón “MENU”.
	2	El sistema muestra una nueva pantalla (pantalla de Menú) compuesta por un conjunto de botones.
	3	El actor usuario (Act-001) selecciona el botón con el texto “COLORES”.
	4	El sistema muestra los elementos de la aplicación a los cuales se pueden modificar el color.
	5	El actor Usuario (Act-001) selecciona el elemento al que desee cambiar el color.
	6	El sistema muestra un conjunto de botones, cada uno de ellos con un color.
	7	El actor Usuario (Act-001) selecciona el color que desee.
8	El sistema actualiza los cambios de color.	

Tabla 7.2: Descripción del caso de uso: Seleccionar colores.



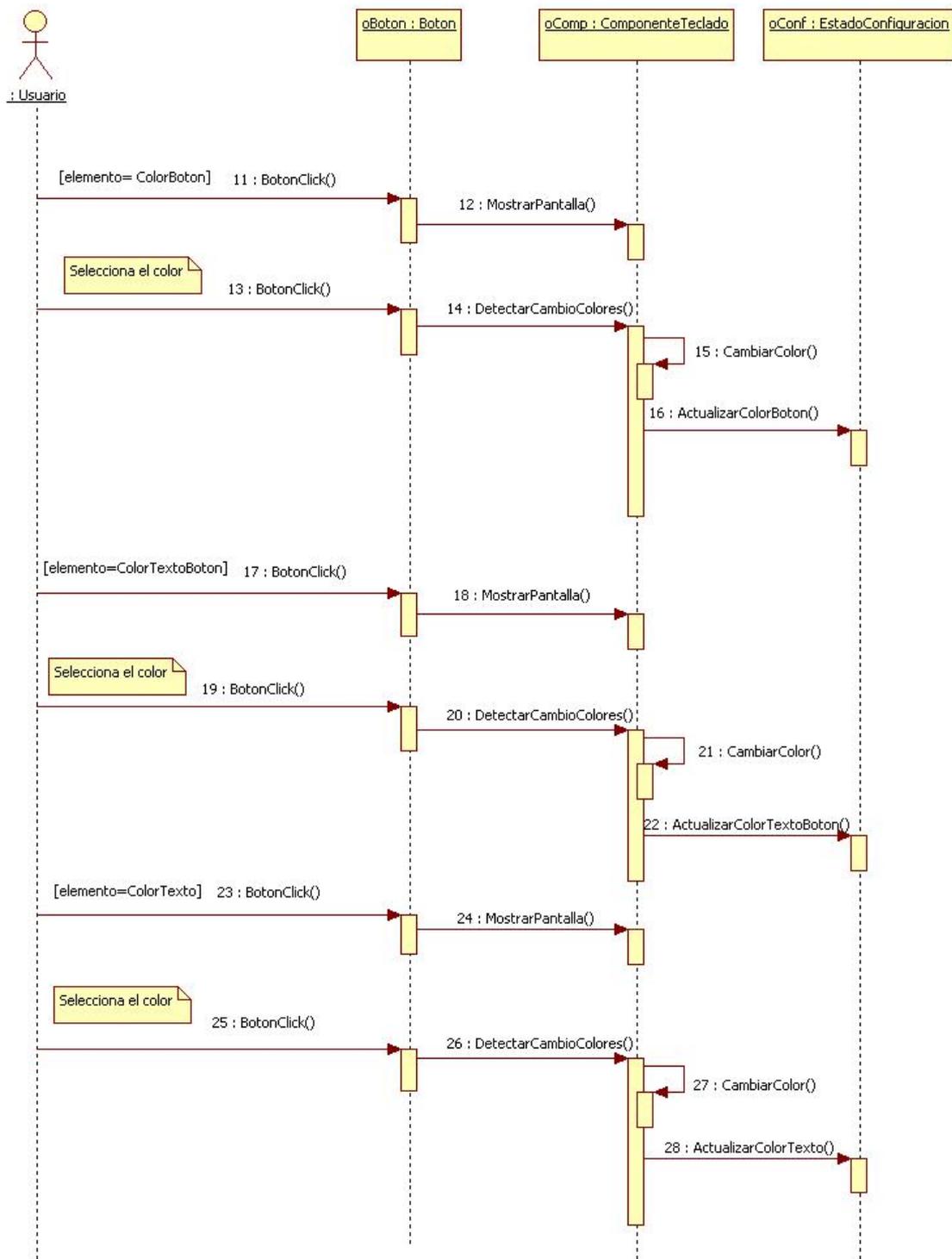


Figura 7.2: Diagrama de secuencia del caso de uso Seleccionar colores.

C-0003	Seleccionar	
<b>Descripción</b>	<i>El sistema deberá permitir que el usuario seleccione y configure el barrido, así como la otra posible opción de selección (método de escritura)</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) selecciona el botón “MENU”.
	2	El sistema muestra una nueva pantalla (pantalla de Menú), compuesta por un conjunto de botones.
	3	El actor usuario (Act-001) selecciona el botón con el texto “CONFIGURACIÓN”.
4	El sistema muestra las opciones de configuración disponibles.	
<b>Extensiones</b>	C-0004 : <i>Seleccionar Barrido</i> C-0005 : <i>Seleccionar Método de Escritura</i>	

Tabla 7.3: Descripción del caso de uso: Seleccionar.

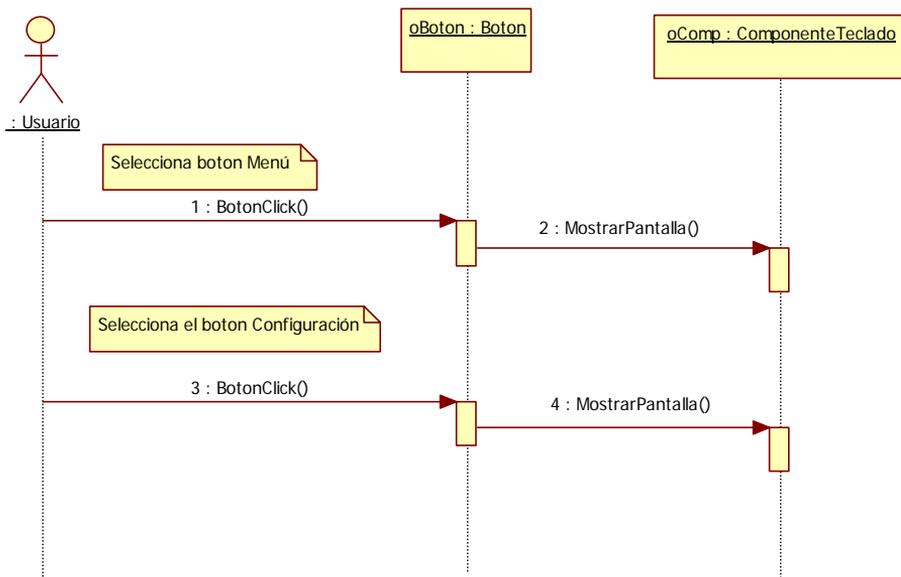


Figura 7.3: Diagrama de secuencia del caso de uso Seleccionar.

C-0004		Seleccionar barrido	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en este caso de uso, cuando se desee activar o desactivar el barrido, así como configurar su tiempo.</i>		
<b>Precondición</b>	<i>En el método de escritura basado en imágenes, no está permitido trabajar sin barrido, porque éste es necesario para el paso automático de pantallas.</i>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El actor Usuario (Act-001) selecciona las opciones de barrido que desee y pulsa "ACEPTAR".	
	2	El sistema muestra un mensaje de aviso para reiniciarse y actualizar la configuración.	
	3	El actor Usuario (Act-001) acepta el mensaje.	
	4	El sistema se reinicia con la situación del barrido actualizada.	
<b>Caso de uso extendido</b>	C-0003 : <i>Seleccionar</i>		

Tabla 7.4: Descripción del caso de uso: Seleccionar barrido.

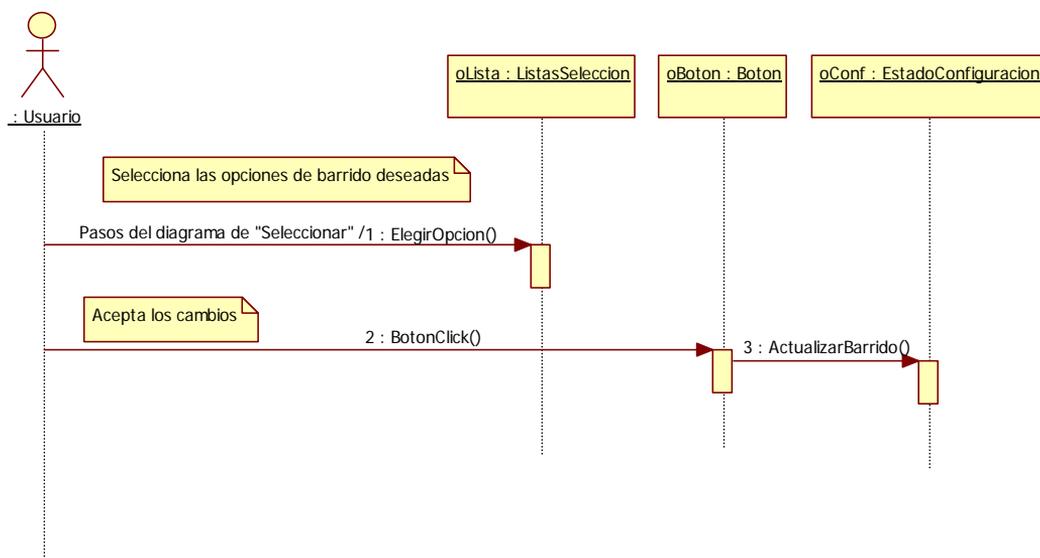


Figura 7.4: Diagrama de secuencia del caso de uso Seleccionar barrido.

<b>C-0005</b>	<b>Seleccionar método de escritura</b>	
<b>Descripción</b>	<i>El sistema permitirá seleccionar el método de escritura a utilizar, de entre todos los disponibles.</i>	
<b>Precondición</b>	<i>El sistema deberá tener al menos un método de escritura disponible.</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) selecciona el método de escritura que desea y pulsa "ACEPTAR".
	2	El sistema muestra un mensaje de aviso para reiniciarse y actualizar la configuración.
	3	El actor usuario (Act-001) acepta el mensaje.
4	El sistema se reinicia con el nuevo método de escritura.	
<b>Caso de uso extendido</b>	<i>C-0003 : Seleccionar</i>	

Tabla 7.5: Descripción del caso de uso: Seleccionar método de escritura.

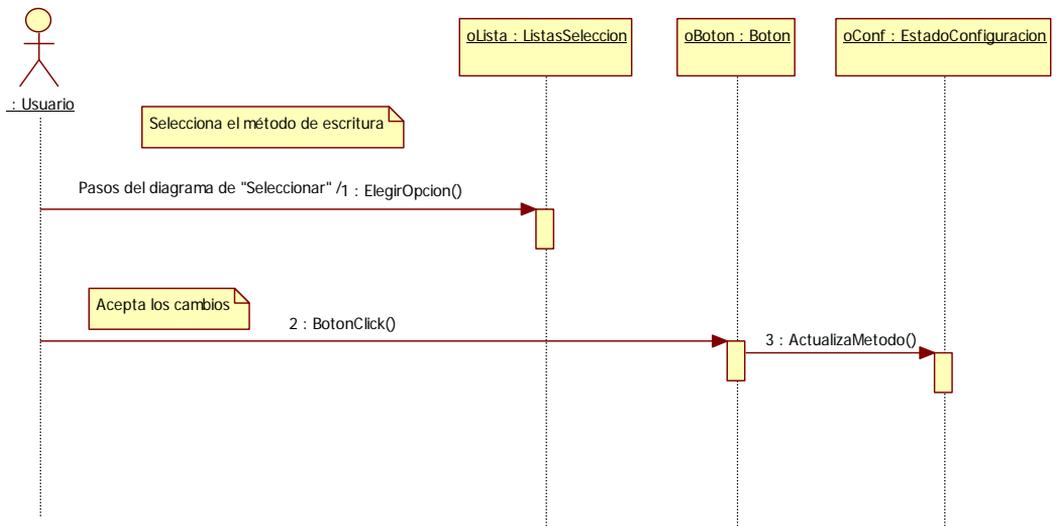


Figura 7.5: Diagrama de secuencia del caso de uso Seleccionar método de escritura.

C-0006		Borrar palabra	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario desee borrar la última palabra del texto escrito en el cuadro de texto.</i>		
<b>Precondición</b>	<i>El campo de texto debe tener algún texto para poder borrarlo.</i>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El actor usuario (Act-001) selecciona el botón “B.Palabra”.	
	2	El sistema borra la última palabra del cuadro de texto.	
<b>Caso de uso extendido</b>	<i>C-0006 : Borrar Texto</i>		

Tabla 7.6: Descripción del caso de uso: Borrar palabra.

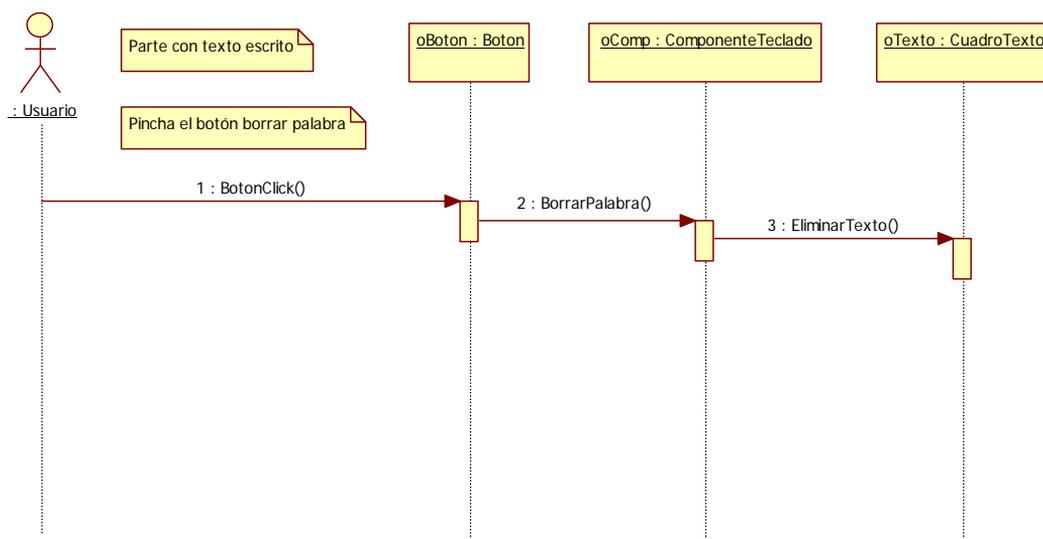


Figura 7.6: Diagrama de secuencia del caso de uso Borrar palabra.

C-0007	Insertar frase	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiera insertar frases en el texto escrito hasta el momento</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) selecciona el botón “MENU”.
	2	El sistema muestra una nueva pantalla (pantalla de Menú) compuesta por un conjunto de botones.
	3	El actor usuario (Act-001) selecciona el botón con el texto “FRASES”.
	4	El sistema muestra una nueva pantalla (pantalla de Frases) compuesta por un campo de texto, un menú de selección con las frases almacenadas disponibles y un conjunto de botones para su inserción, borrado y almacenado.
	5	El actor usuario (Act-001) selecciona una de las frases y pulsa el botón “INSERTAR”
6	El sistema inserta la frase a continuación del texto escrito en el campo texto (haya o no texto escrito)	

Tabla 7.7: Descripción del caso de uso: Insertar frase.

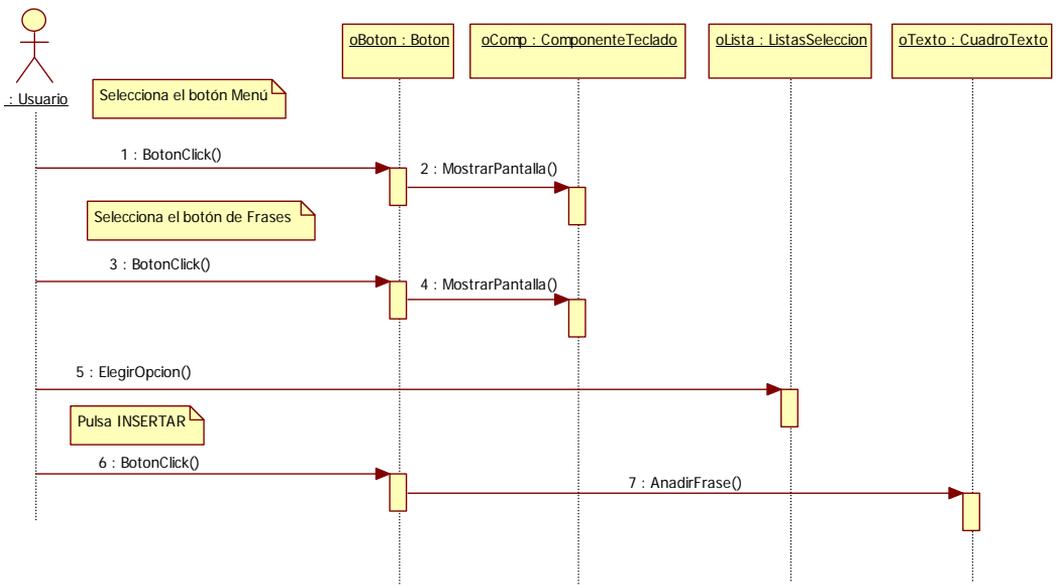


Figura 7.7: Diagrama de secuencia del caso de uso Inserta frase.

C-0008		Guardar frase	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiera almacenar en la aplicación las frases que desee para su posterior uso.</i>		
<b>Precondición</b>	<i>Hay almacenadas n frases (n&gt;=0) El campo de texto debe contener algún texto escrito.</i>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El actor usuario (Act-001) selecciona el botón “MENU”.	
	2	El sistema muestra una nueva pantalla (pantalla de Menú) compuesta por un conjunto de botones	
	3	El actor usuario (Act-001) selecciona el botón con el texto “FRASES”.	
	4	El sistema muestra una nueva pantalla (pantalla de Frases) compuesta por un campo de texto, un menú de selección con las frases almacenadas disponibles y un conjunto de botones para su inserción, borrado y almacenado.	
	5	El actor usuario (Act-001) pulsa el botón “GUARDAR”	
	6	El sistema almacena el texto escrito en el campo de texto como una nueva frase disponible.	
<b>Postcondición</b>	<i>Hay almacenadas n+1 frases (n&gt;=0)</i>		

Tabla 7.8: Descripción del caso de uso: Guardar frase.

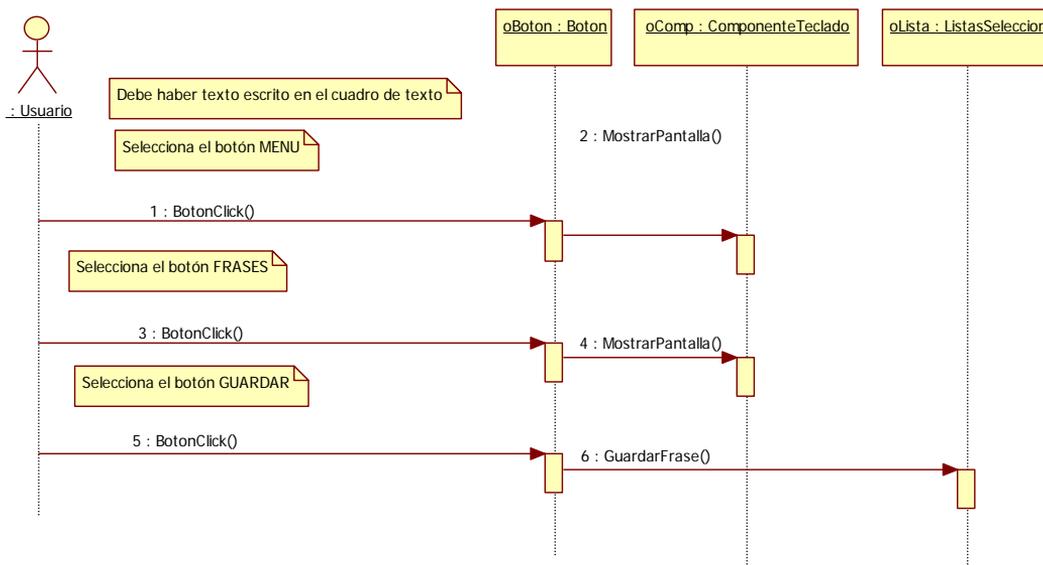


Figura 7.8: Diagrama de secuencia del caso de uso Guardar frase.

C-0009	Borrar frase	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiera borrar alguna frase almacenada</i>	
<b>Precondición</b>	<i>Hay almacenadas m frases (m &gt;= 1)</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) selecciona el botón "MENU".
	2	El sistema muestra una nueva pantalla (pantalla de Menú) compuesta por un conjunto de botones
	3	El actor usuario (Act-001) selecciona el botón con el texto "FRASES".
	4	El sistema muestra una nueva pantalla (pantalla de Frases) compuesta por un campo de texto, un menú de selección con las frases almacenadas disponibles y un conjunto de botones para su inserción, borrado y almacenado.
	5	El actor usuario (Act-001) selecciona una de las frases y pulsa el botón "BORRAR"
6	El sistema borra dicha la frase seleccionada.	
<b>Postcondición</b>	<i>Hay almacenadas m-1 frases (m &gt;= 1)</i>	

Tabla 7.9: Descripción del caso de uso: Borrar frase.

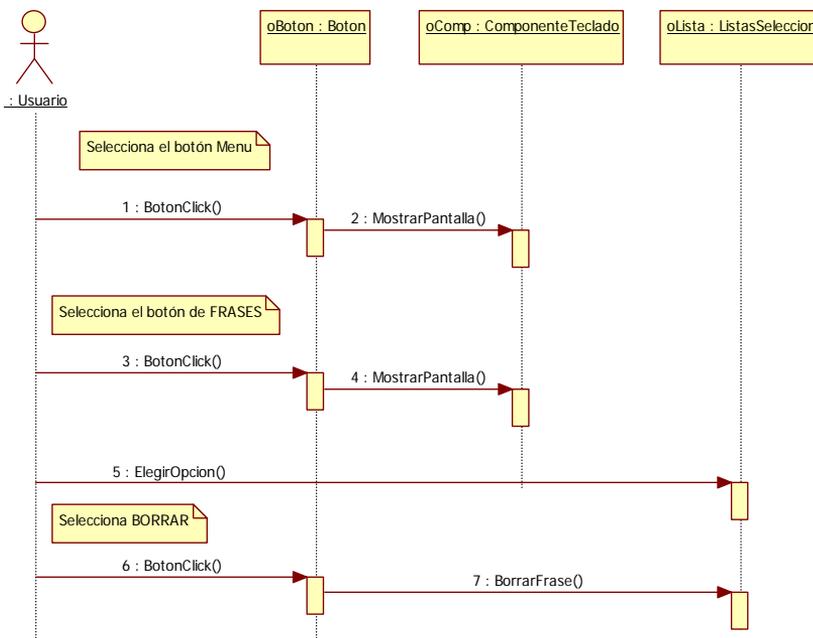


Figura 7.9: Diagrama de secuencia del caso de uso Borrar frase.

C-0010	Reproducir	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiera reproducir un texto escrito.</i>	
<b>Precondición</b>	<i>El campo de texto debe tener algún texto para poder reproducirlo.</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) selecciona el botón “MENU”.
	2	El sistema muestra una nueva pantalla (pantalla de Menú) compuesta por un conjunto de botones.
	3	El actor usuario (Act-001) selecciona el botón con el texto “PLAY”.
	4.a	El sistema reproduce el texto que haya escrito en el campo de texto y sigue mostrando la pantalla de menú.
4.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>	4a,4b	<i>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.</i>

Tabla 7.10: Descripción del caso de uso: Reproducir.

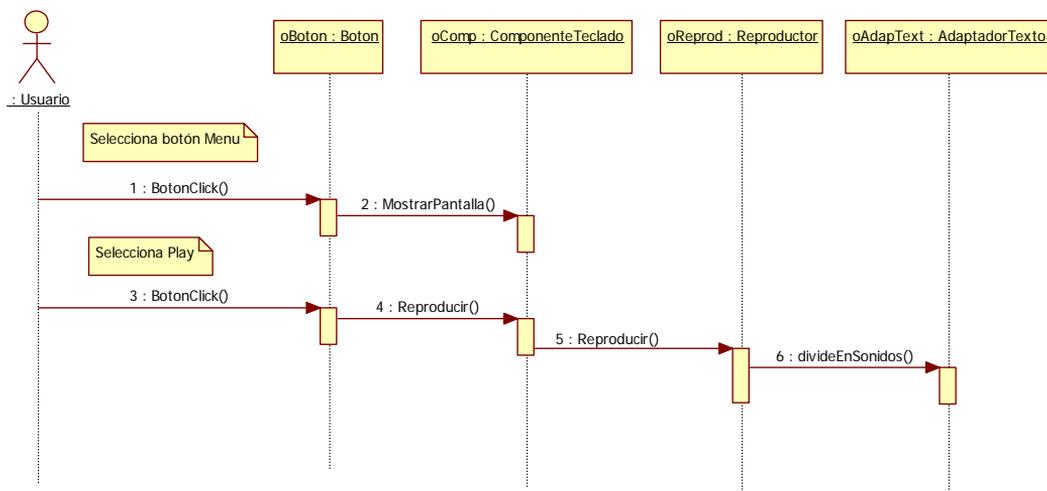


Figura 7.10: Diagrama de secuencia del caso de uso Reproducir

<b>C-0011</b>		<b>Escribir texto</b>	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe, cuando se desee añadir caracteres al cuadro de texto, de forma tal, que los botones de la pantalla, estén asociados a caracteres tipográficos. La secuencia se completará con los casos de uso extendidos a partir de este mismo caso de uso.</i>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El sistema mostrará una pantalla compuesta por un campo de texto y un conjunto de botones asociados a uno o varios caracteres tipográficos, que tendrán una disposición sobre la pantalla y una funcionalidad particular dependiendo del método de escritura que se esté utilizando.	
<b>Extensiones</b>	<i>C-0012 : Escribir Texto Categorías</i> <i>C-0013: Escribir Texto Vocales</i> <i>C-0014: Escribir Texto Letras Agrupadas</i>		

Tabla 7.11: Descripción del caso de uso: Escribir texto

C-0012		Escribir texto Categorías	
<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 por método de escritura "categorías"		
<b>Precondición</b>	El método de escritura debe ser el de "categorías" Cada botón de la pantalla representa a una categoría de caracteres (vocales, consonantes, números, signos).		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El actor Usuario (Act-001) seleccionará el botón del grupo de caracteres que contenga el caracter que desea escribir.	
	2a	Si el sistema muestra una pantalla formada por el cuadro de texto y botones de escritura, ir al paso 3.	
	2b	Si el sistema muestra una pantalla formada por el cuadro de texto y botones de selección de grupo, ir al paso 1.	
	3	El actor Usuario (Act-001) seleccionará el botón que represente el carácter que desea escribir.	
4	El sistema añade el carácter asociado al botón a lo que ya hubiera escrito en el campo de texto.		
<b>Caso de uso extendido</b>	C-0011 : Escribir		

Tabla 7.12: Descripción del caso de uso: Escribir texto Categorías.

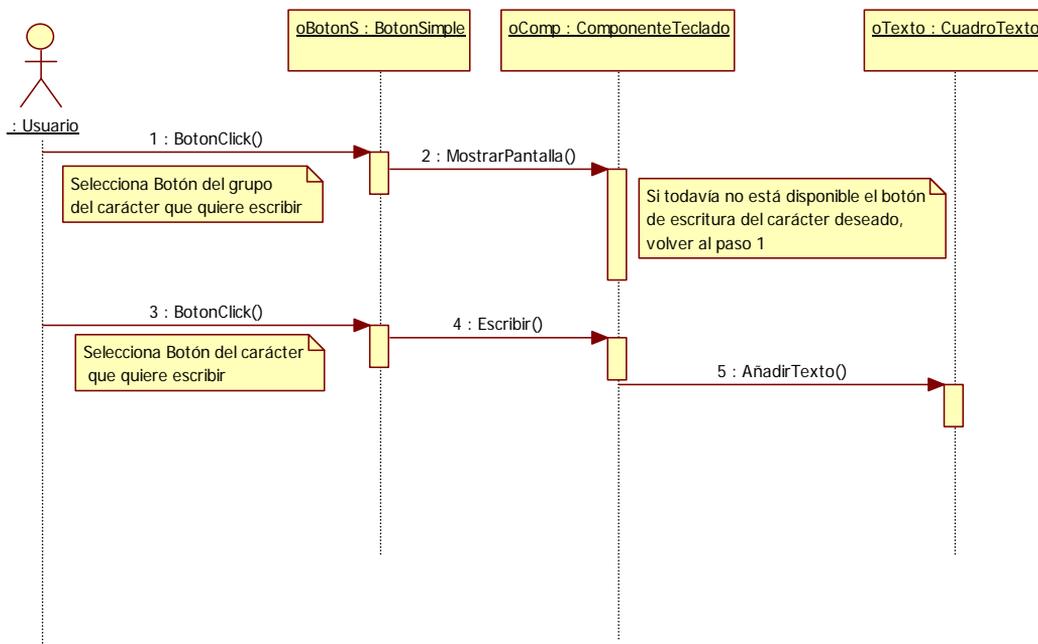


Figura 7.11: Diagrama de secuencia del caso de uso Escribir texto Categorías.

C-0013		Escribir texto Vocales	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en este caso de uso, cuando se desee añadir un texto al cuadro de texto por método de escritura “método de las vocales”</i>		
<b>Precondición</b>	<i>El método de escritura debe ser el de “método de las vocales” Cada botón de la pantalla representará a un conjunto de caracteres, excepto 5 botones especiales que representarán cada uno a una vocal.</i>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1a	Si el actor Usuario(Act-001) selecciona un botón correspondiente a una vocal , ir al paso 5	
	1b	El actor Usuario (Act-001) seleccionará el grupo que contenga el carácter que desea escribir.	
	2	El sistema mostrará una pantalla formada por el cuadro de texto y botones de escritura para cada carácter dentro del grupo.	
	3	El actor Usuario (Act-001) selecciona un botón correspondiente al carácter que desea escribir.	
5	El sistema añade el carácter asociado al botón a lo que ya hubiera escrito en el campo de texto.		
<b>Caso de uso extendido</b>	<i>C-0011 : Escribir</i>		

Tabla 7.13: Descripción del caso de uso: Escribir texto Vocales.

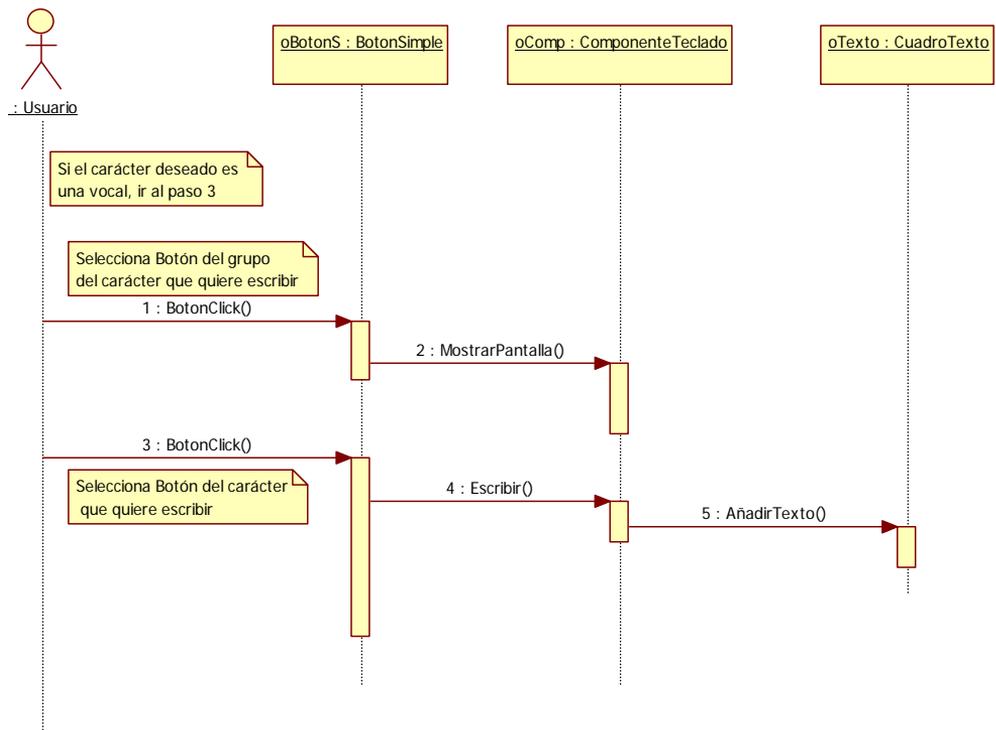


Figura 7.12: Diagrama de secuencia del caso de uso Escribir texto vocales.

C-0014		Escribir texto Letras agrupadas
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en este caso de uso, cuando se desee añadir un texto al cuadro de texto por método de escritura "letras agrupadas"</i>	
<b>Precondición</b>	<i>El método de escritura debe ser el de "letras agrupadas" Cada botón de la pantalla representa a un conjunto de caracteres (abcde, fghij,klmnñ, etc...).</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor Usuario (Act-001) seleccionará el grupo que contenga el carácter que desea escribir.
	2	El sistema mostrará una pantalla formada por el cuadro de texto y botones de escritura para cada carácter dentro del grupo.
	3	El actor Usuario (Act-001) selecciona un botón correspondiente al carácter que desea escribir.
4	El sistema añade el carácter asociado al botón a lo que ya hubiera escrito en el campo de texto.	
<b>Caso de uso extendido</b>	C-0011 : <i>Escribir</i>	

Tabla 7.14: Descripción del caso de uso: Escribir texto Letras agrupadas.

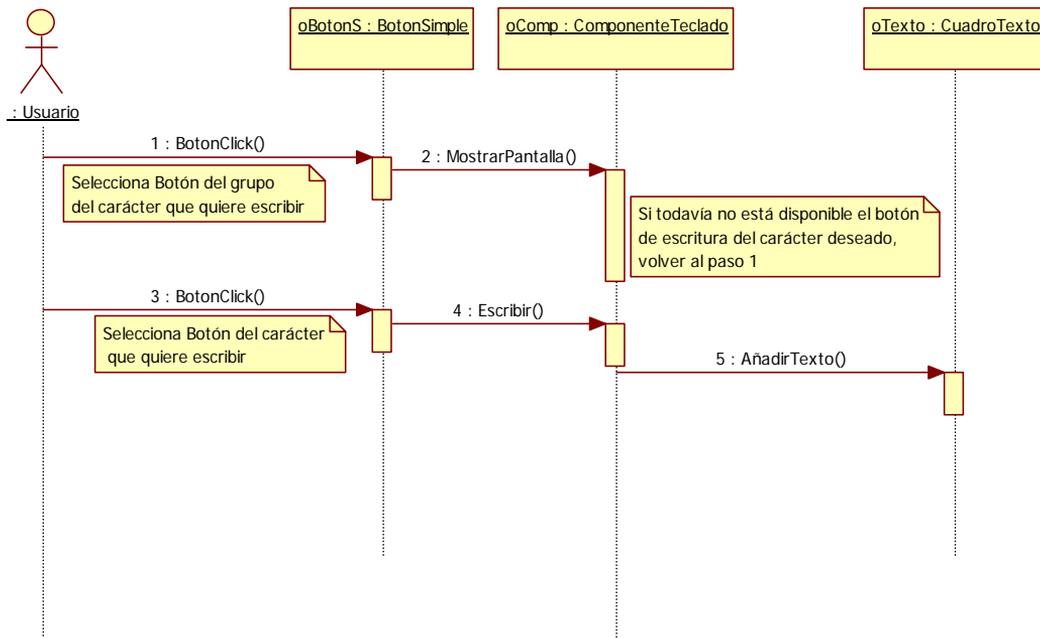


Figura 7.13: Diagrama de secuencia del caso de uso Escribir texto Letras agrupadas.

C-0015	Escribir método variable	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe en este caso de uso, cuando se desee añadir un texto al cuadro de texto por método de escritura “método variable”</i>	
<b>Precondición</b>	<i>El método de escritura debe ser el de “método variable”</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor Usuario (Act-001) seleccionará el botón relacionado con el carácter que quiera escribir.
	2a	Si el botón que selecciona el usuario tiene como funcionalidad “escribir” ir al paso 4
	2b	Si el botón que selecciona el usuario no tiene como funcionalidad “escribir”, el sistema mostrará una pantalla formada por un conjunto de botones que tendrán una disposición sobre la pantalla y una funcionalidad particular dependiendo del contenido del fichero xml correspondiente al método configurable.
	4	Ir al paso 1
5	El sistema añade la palabra o la expresión asociada al botón a lo que ya hubiera escrito en el campo de texto	

Tabla 7.15: Descripción del caso de uso: Escribir método variable.

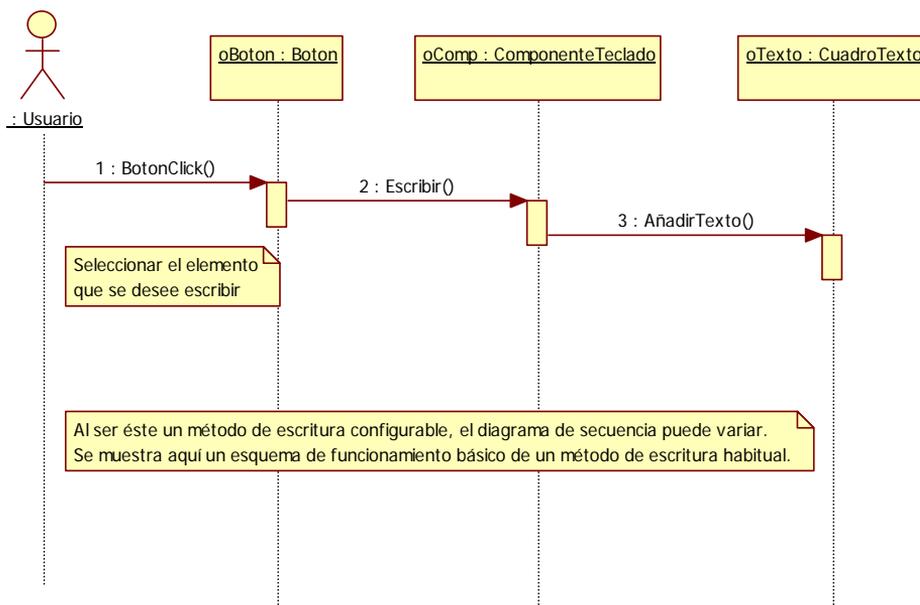


Figura 7.14: Diagrama de secuencia del caso de uso Escribir método variable.

C-0016	Escribir imágenes	
<b>Descripción</b>	<i>El sistema deberá comportarse tal como se describe, cuando se desee añadir caracteres al cuadro de texto, de forma tal, que los botones de la pantalla, estén asociados a una imagen.</i>	
<b>Precondición</b>	<i>El método de escritura debe ser el de "Imágenes"</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor Usuario (Act-001) seleccionará el botón que represente al grupo de palabras que contenga el texto que desea escribir.
	2	El sistema mostrará una pantalla formada por un conjunto de botones-imágen con su palabra, así como un conjunto de botones para su navegación por las páginas de imágenes.
	3	El actor Usuario (Act-001) seleccionará el botón imagen que represente la palabra que desea escribir
4	El sistema añade la palabra asociada al botón a lo que ya hubiera escrito en el campo de texto	

Tabla 7.16: Descripción del caso de uso: Escribir imágenes.

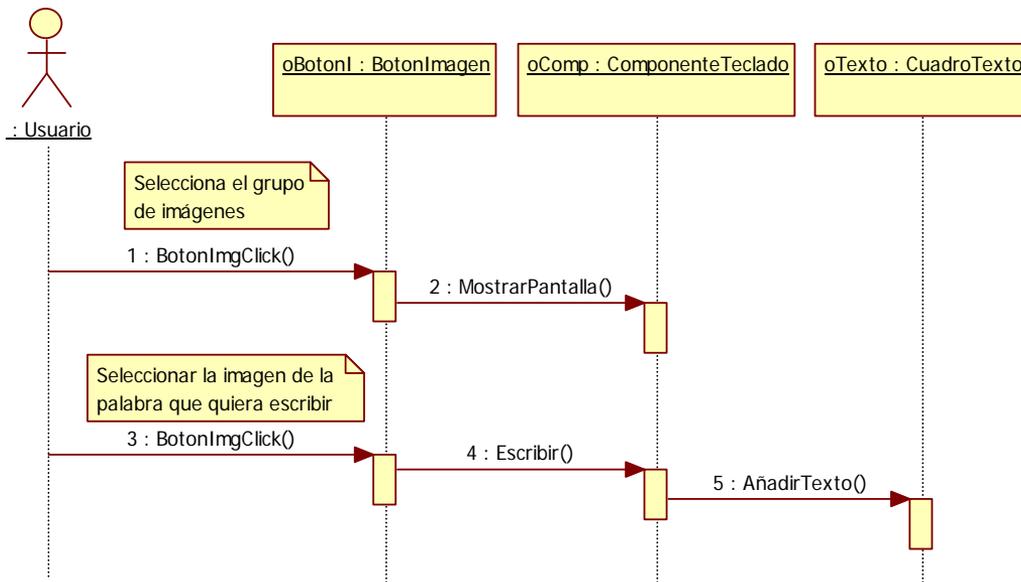


Figura 7.15: Diagrama de secuencia del caso de uso Escribir imágenes.

C-0017	Salir de la aplicación	
<b>Descripción</b>	<i>El sistema deberá permitir que el usuario pueda salir de la aplicación.</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor usuario (Act-001) selecciona el botón “MENU”.
	2	El sistema muestra una nueva pantalla (pantalla de Menú) compuesta por un conjunto de botones.
	3	El actor usuario (Act-001) selecciona el botón con el texto “SALIR”.
4	El sistema termina la aplicación.	

Tabla 7.17: Descripción del caso de uso: Salir de la aplicación.

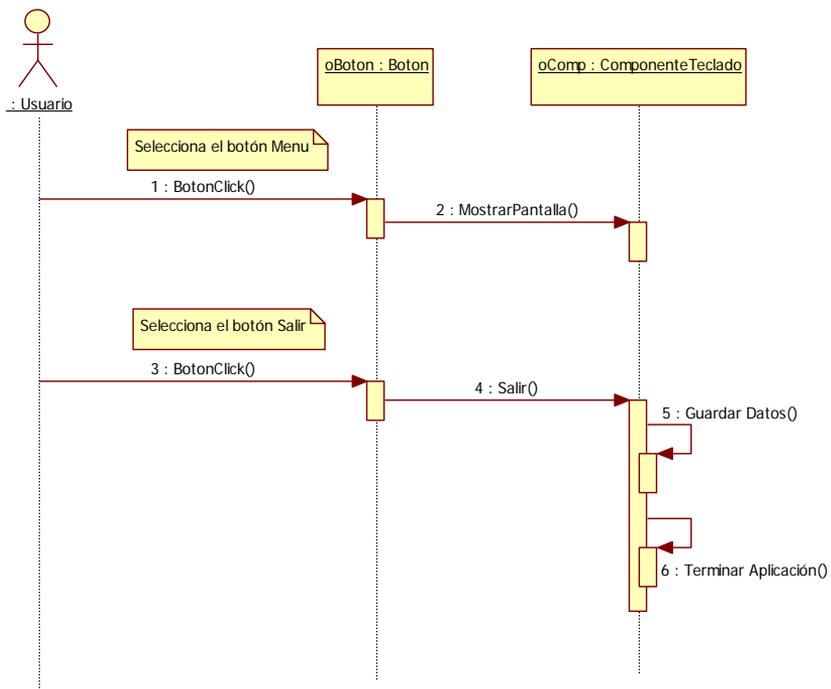


Figura 7.16: Diagrama de secuencia del caso de uso Salir de la aplicación.

### 7.3 Diagrama final de clases

Tras tener el primer esbozo del diagrama de clases que puede verse en la figura 6.18, se muestra en la figura 7.17, el diagrama completo al que se le han añadido cuatro clases: BotonRadio, ListasSeleccion, Etiquetas y Adaptador de Texto. Se amplía por tanto la descripción de las clases de la tabla 6.21 en la tabla 7.18 que se puede apreciar a continuación.

CLASE	DESCRIPCIÓN
Componente Teclado	Esta clase será la 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.
Estado Configuración	Almacenará la información necesaria para que la clase ComponenteTeclado conozca cuál es la configuración que debe utilizar para crear el teclado.
CuadroTexto	Clase encargada de almacenar y mostrar el texto escrito por el usuario.
Barrido	Clase encargada de controlar el tiempo que cada botón, de una determinada pantalla, está señalado a la espera de una pulsación del usuario.
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.
Boton	Almacenará la información (posición, tamaño, apariencia y funcionalidad) de los botones que formarán el teclado.
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.
BotonSimple	Almacenará la información relativa a los botones de caracteres que aparecerán en las pantallas del teclado.
Imagen	Almacenará información (imagen, imagen con foco de barrido y texto a escribir por su botón asociado) sobre las imágenes que aparecerán en las pantallas cuando se esté trabajando con alguna configuración de imágenes.
Reproductor	Clase encargada de la reproducción del texto escrito.
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.
ListasSeleccion	Almacenará la información relativa a las listas de selección donde se mostrará al usuario las posibilidades de selección. Estarán presentes en la pantalla de configuración, posibilitando el elegir el tiempo de barrido y el método de escritura a emplear. También almacenará la lista de selección relativa a las frases almacenadas y con las que podrá comunicarse.
Etiquetas	Almacenará información (posición, tamaño, texto y colores) relativa el texto informativo que aparecerá en las distintas pantallas. El texto 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.).
AdaptadorTexto	Clase que lleva a cabo las modificaciones sobre el texto, para dar lugar a una lista de ficheros reproducibles, y además, implementa la síntesis de voz.

Tabla 7.18: Descripción completa de las clases.



## 7.4 Diagrama de clases empaquetadas

En este diagrama ya se incluyen los archivos .xml de donde se tomará toda la información necesaria para crear la interfaz de la aplicación, de acuerdo a las características requeridas. Dichos archivos se explicarán más detenidamente en el punto 8.2.1.

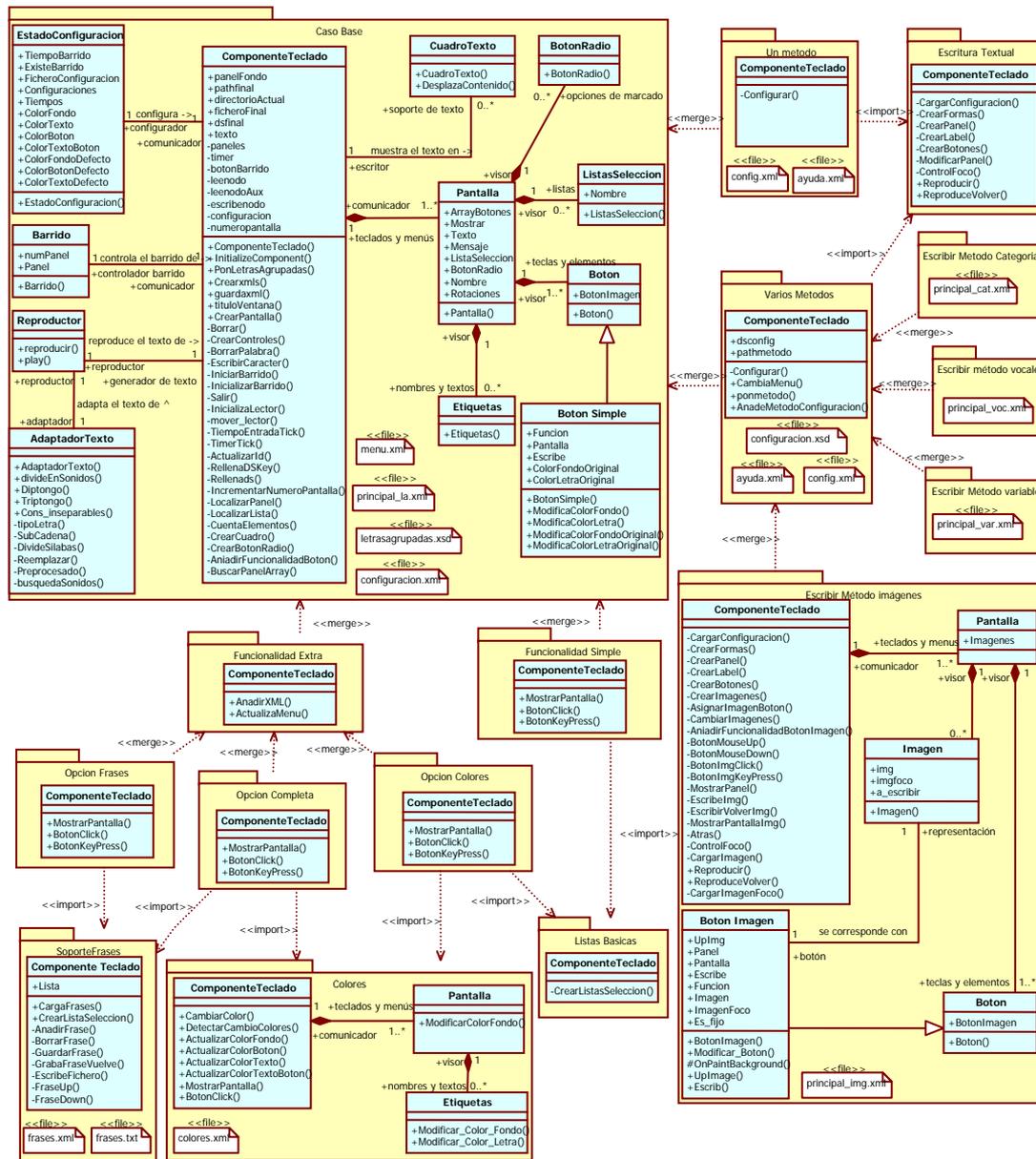


Figura 7.18: Diagrama final de clases empaquetadas.

A continuación se procederá a describir el diagrama de clases empaquetadas de la Figura 7.18 por partes. Para su mejor visualización, se dividirá en:



**Paquetes de Escritura:**

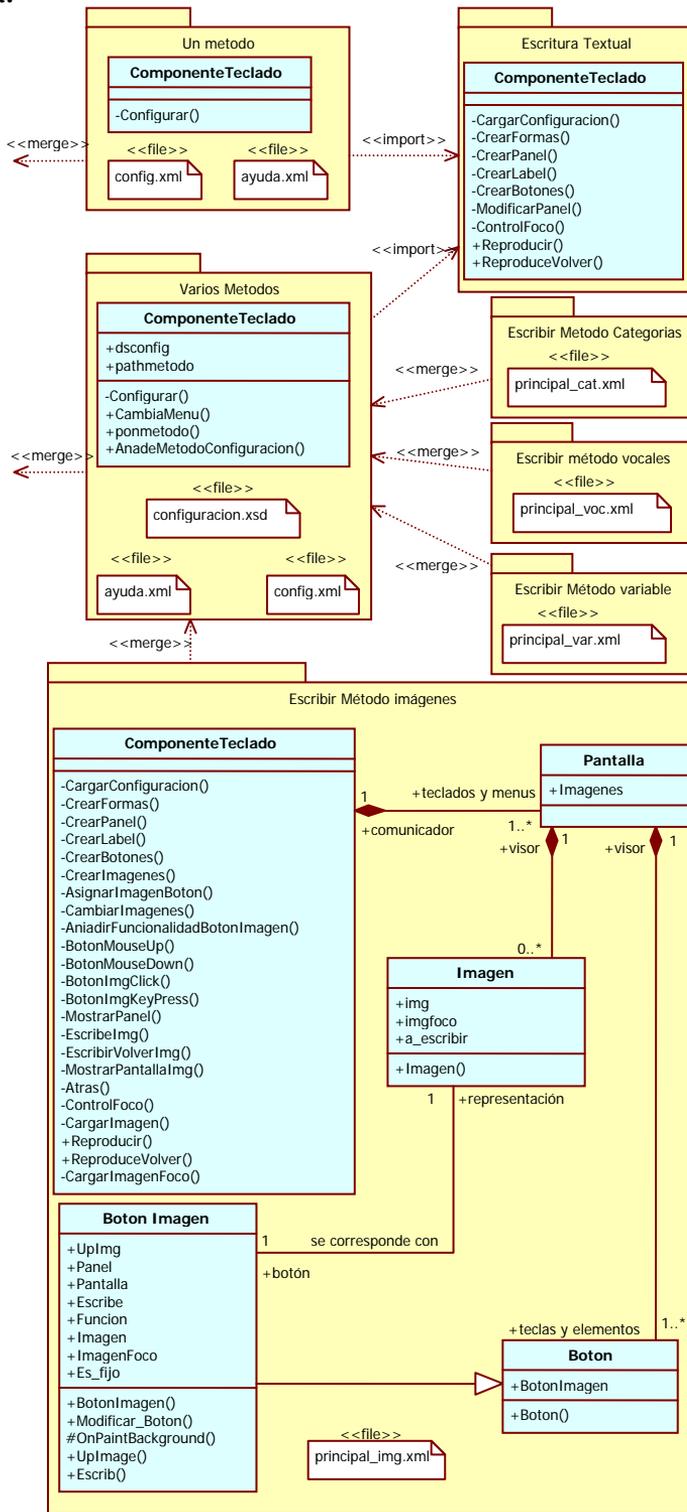


Figura 7.20: Detalle del diagrama de clases empaquetado: Escritura

Si se desea que la aplicación a desarrollar contenga sólo el método de escritura por defecto, es decir, el método de letras agrupadas, se deberá incluir el paquete “Un Método”, que optimiza la pantalla de configuración de la aplicación para tal fin. También contiene archivos .xml correspondientes al método de escritura por defecto.

De igual forma, se deberá incluir en la aplicación el paquete “Escritura Textual”, ya que el método de letras agrupadas está basado en texto y no se incluye ningún otro método de escritura. Por este motivo, las relaciones del paquete “Escritura Textual”, en lugar de “merge” son “import”: Es un paquete necesario para el buen funcionamiento de la aplicación. Para poder llevar a cabo la relación “import” basta con añadir el paquete importado junto al paquete que lo importa.

Cuando se desee disponer de algún método de escritura adicional, se deberá incluir por el contrario, el paquete “Varios Métodos” en la aplicación, que incorpora a la clase ComponenteTeclado las funciones necesarias para tal fin. Asimismo, contiene archivos .XML y XSD para la correcta representación de la interfaz asociada a cada método de escritura que se incluya.

De igual manera, deberemos incluir aquellos paquetes que correspondan a los métodos de escritura que queramos que estén disponibles en la aplicación (ver Figura 7.20). Los métodos de escritura basados en texto sólo contienen el archivo .XML donde se detalla cada método. Éstos son: “Escribir Método Categorías”, “Escribir Método Vocales” y “Escribir Método Variable”. Al añadirlos, únicamente, se debe incluir también el paquete “Escritura Textual”. Si por el contrario, añadimos el método de escritura mediante imágenes, no hará falta agregarlo.

El método de escritura mediante imágenes presenta diferencias con los demás métodos, ya que la interfaz del comunicador debe ser adaptada para soportar un nuevo tipo de botón: el BotónImagen, así como su imagen asociada. Al igual que los demás paquetes de métodos, cuenta con su archivo XML donde se detalla el método de escritura.

Para una descripción de cada clase, ver la tabla 7.18, o el apartado 7.5 “Especificación de las clases”.

**Paquetes de Funcionalidades:**

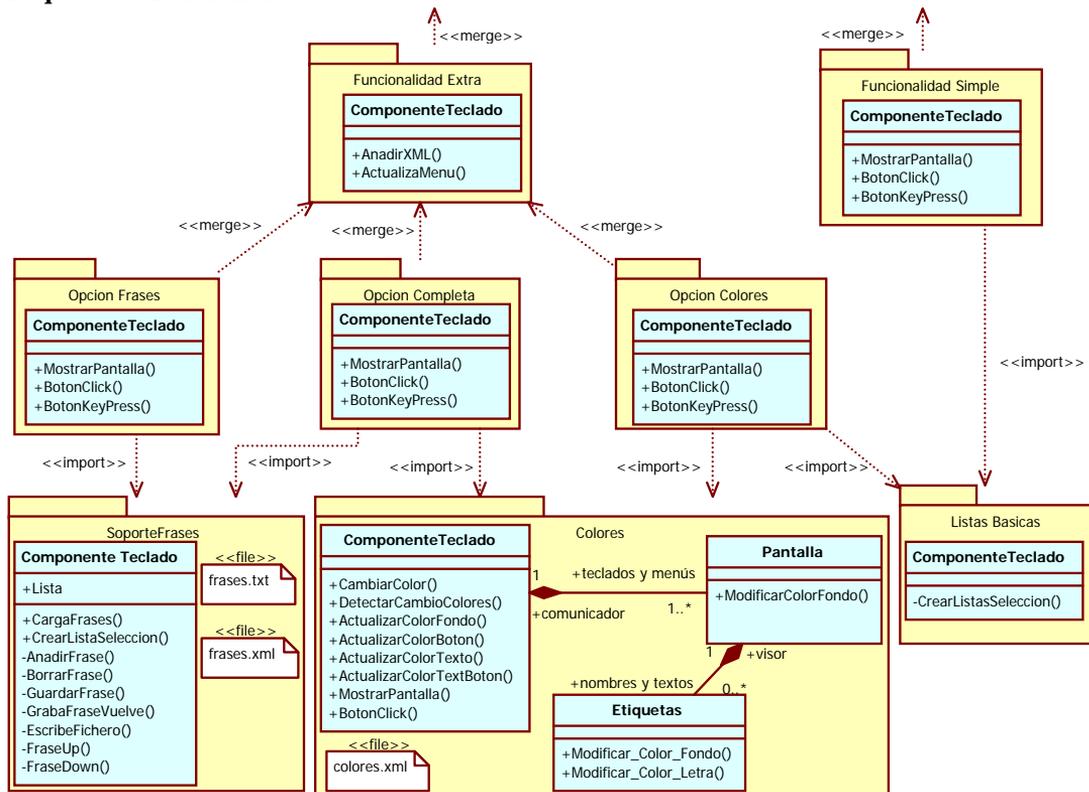


Figura 7.21: Detalle del diagrama de clases empaquetado: Funcionalidades

En la figura 7.21 se detallan los paquetes relacionados con las funcionalidades de la aplicación a desarrollar.

Hay dos funcionalidades extra de las que podemos dotar a nuestra aplicación: Poder cambiar los colores a los elementos de la aplicación e insertar, borrar y guardar frases.

Si no queremos disponer de ninguna de estas posibilidades, bastará con añadir el paquete “Funcionalidad Simple”, que establece la funcionalidad básica a los botones y pantallas de nuestra aplicación. Este paquete importa el paquete “Listas Básicas”, que permitirá el empleo de listas para los métodos de escritura y los tiempos de barrido. Hay que recordar que para que un paquete importe a otro, bastará con incluir ambos al compilar y construir la aplicación.

Si queremos tener alguna funcionalidad extra en nuestra aplicación, se deberá añadir el paquete “Funcionalidad Extra” y los paquetes que se consideren de la funcionalidad o funcionalidades que se deseen incluir. Disponemos de 3 opciones:

1. Funcionalidad de Gestión y uso de frases: Se deberá incluir los paquetes “Opción Frases” y “Soporte de Frases”. El primero adapta los botones y pantallas de la aplicación para que soporten dicha funcionalidad, y el segundo paquete implementa el uso y gestión de las frases en sí. Un hecho a destacar, es que este último paquete también permite el empleo de listas, ampliando su uso a las frases, por lo que no es necesario el paquete “Listas Básicas”.

2. Funcionalidad de Cambio de colores: Se deberá incluir los paquetes “Opción Colores”, “Colores” y “Listas Básicas”. El primero adapta los botones y pantallas de la aplicación para que soporten dicha funcionalidad, el segundo implementa la posibilidad de cambiar los colores de los elementos de la aplicación, y el último permite el empleo de listas.
3. Funcionalidades de Gestión de frases y Cambio de colores: En este caso se incluirán los paquetes “Opción completa”, que adapta los botones y pantallas de la aplicación para que soporten ambas funcionalidades y “ Soporte de Frases” y “Colores”, que las implementan.

## 7.5 Especificación de las clases

A continuación se muestran los atributos y métodos más importantes de cada clase organizada por los distintos paquetes explicados anteriormente. La descripción completa de todos los métodos de cada clase se puede ver en el CD adjunto.

### PAQUETE CASO BASE

<b>Clase ComponenteTeclado</b>	
<b>Atributos</b>	<ul style="list-style-type: none"> <li>• <i>panelFondo</i>: el contenedor de todos los controles</li> <li>• <i>texto</i>: cuadro de texto donde aparecerá el texto escrito</li> <li>• <i>paneles</i>: paneles que existirán en la aplicación</li> <li>• <i>timer</i>: controlará el tiempo del barrido</li> <li>• <i>botonBarrido</i>: posición del siguiente botón que recibirá el foco</li> <li>• <i>configuracion</i>: contiene la configuración actual</li> </ul>
<b>Métodos</b>	<ul style="list-style-type: none"> <li>• <i>InitializeComponent()</i>: lanza la aplicación</li> <li>• <i>PonLetrasAgrupadas()</i>: método que crea el archivo “letrasagrupadas.xml”</li> <li>• <i>CrearPantalla()</i>: crea todas las pantallas que se van a utilizar en la aplicación y su información correspondiente</li> <li>• <i>CrearCuadro()</i>: crea el cuadro de texto rellenándolo o no con texto dependiendo de la información del xml.</li> <li>• <i>AniadirFuncionalidadBoton(Boton boton, int numpanel)</i>: método que da a cada botón que aparecerá en la pantalla su funcionalidad.</li> <li>• <i>Salir()</i>: Graba en el xml la información de la configuración actual y sale de la aplicación.</li> <li>• <i>EscribirCaracter()</i>: refleja en el cuadro de texto el texto que se escribe</li> <li>• <i>Borrar()</i>: Borra el último carácter escrito en el cuadro de texto.</li> <li>• <i>Reproducir()</i>: Reproduce el texto escrito en el cuadro de texto.</li> <li>• <i>BorrarPalabra()</i>: método que borra la última palabra escrita del cuadro de texto.</li> </ul>

Tabla 7.19: Clase Componenteteclado del paquete Caso base.

## Clase CuadroTexto

### Atributos

### Métodos

- *CuadroTexto*: Crea el objeto CuadroTexto.
- *DesplazaContenido*: sitúa el cursor al final del texto escrito.

Tabla 7.20: Clase CuadroTexto del paquete Caso base.

## Clase EstadoConfiguración

### Atributos

- *TiempoBarrido*: Indica el tiempo o intervalo de barrido.
- *ExisteBarrido*: Indica si el barrido está o no activo.
- *Configuraciones*: Listado de métodos de escritura disponibles.
- *Tiempos*: Listado de tiempos de barrido disponibles.
- *ColorFondo*: Color del fondo de la aplicación.
- *ColorTexto*: Color del texto en la aplicación.
- *ColorBoton*: Color de los botones en la aplicación.
- *ColorTextoBoton*: Color del texto en los botones de la aplicación.

### Métodos

- *EstadoConfiguracion*: Crea el objeto EstadoConfiguración y almacena la información de la configuración actual.

Tabla 7.21: Clase EstadoConfiguración del paquete Caso base.

## Clase Pantalla

### Atributos

- *mostrar*: Si la pantalla se muestra o no.
- *texto*: cuadros de texto asociados a la pantalla.
- *Mensaje*: Etiquetas asociadas a la pantalla.
- *listasSeleccion*: Listas de selección asociadas a la pantalla.
- *botonRadio*: Botones de radio asociados a la pantalla
- *nombre*: Nombre de la pantalla.

### Métodos

- *Pantalla*: Crea el objeto Pantalla

Tabla 7.22: Clase Pantalla del paquete Caso base.

## Clase BotonSimple

### Atributos

- *Función:* descripción breve si quieres
- *Pantalla:* más de lo mismo
- *Escribe:* más de lo mismo
- *Panel:* más de lo mismo

### Métodos

- *BotonSimple:* Crea el objeto BotonSimple
- *ModificarColorFondo.*
- *ModificarColorLetra.*

Tabla 7.23: Clase BotonSimple del paquete Caso base.

## Clase BotonRadio

### Atributos

### Métodos

- *BotonRadio:* Crea el objeto BotonRadio.

Tabla 7.24: Clase BotonRadio del paquete Caso base.

## Clase Botón

### Atributos

### Métodos

- *Boton:* Crea el objeto Boton.

Tabla 7.25: Clase Botón del paquete Caso base.

## Clase Etiquetas

### Atributos

### Métodos

- *Etiquetas:* Crea el objeto Etiquetas.

Tabla 7.26: Clase Etiquetas del paquete Caso Base.

## Clase ListasSeleccion

### Atributos

- *Nombre.*

### Métodos

- *ListasSeleccion:* Crea el objeto ListasSeleccion.

Tabla 7.27: Clase ListasSeleccion del paquete Caso Base.

## Clase Reproductor

### Atributos

### Métodos

- *Sound*: Reproduce el texto escrito.

Tabla 7.28: Clase Reproductor del paquete Caso Base.

## Clase Barrido

### Atributos

- *NumPanel*: Panel en el que está activo el barrido

### Métodos

- *Barrido*: Establece el intervalo del tiempo de barrido.

Tabla 7.29: Clase Barrido del paquete Caso Base.

## Clase AdaptadorTexto

### Atributos

### Métodos

- *AdaptadorTexto*: Crea el objeto AdaptadorTexto.
- *divideEnSonidos*.
- *tipoLetra*: Devuelve si la letra es consonante o vocal.
- *Cons\_inseparables*: Devuelve si la subcadena es un sonido inseparable.
- *Diptongos*.
- *Triptongos*.
- *DivideSilabas*.
- *Preprocesado*.
- *BusquedaSonidos*

Tabla 7.30: Clase AdaptadorTexto del paquete Caso Base.

## ARCHIVOS

- *Configuración.xml*
- *LetrasAgrupadas.xsd*
- *Principal\_la.xml*
- *Menu.xml*

Tabla 7.31: Archivos .xml del paquete Caso Base.

---

## PAQUETE UN METODO

### Clase Componente Teclado

#### Atributos

#### Métodos

- *Configurar()*: Guarda los datos de la nueva configuración y reinicia la aplicación.

Tabla 7.32: Clase ComponenteTeclado del paquete Un metodo.

### ARCHIVOS

- *Config.xml*
- *Ayuda.xml*

Tabla 7.33: Archivos .xml del paquete Un metodo.

## PAQUETE VARIOS MÉTODOS

### Clase Componente Teclado

#### Atributos

#### Métodos

- *Configurar()*: Guarda los datos de la nueva configuración y reinicia la aplicación.
- *cambiamenu()*: actualiza los botones del menú para que concuerden con un método de escritura distinto.
- *ponmetodo()*: Posibilita cambiar a un método de escritura distinto.
- *anadeMetodoConfiguracion()*: Añade el método de escritura a la lista de métodos de la pantalla de configuración.

Tabla 7.34: Clase ComponenteTeclado del paquete Varios métodos.

### ARCHIVOS

- *Configuración.xml*
- *Config.xml*
- *Ayuda.xml*

Tabla 7.35: Archivos .xml del paquete Varios métodos.

---

## PAQUETE ESCRITURA TEXTUAL

Clase Componente Teclado	
<b>Atributos</b>	
<b>Métodos</b>	<ul style="list-style-type: none"><li>• <i>CargarConfiguracion()</i>: Guarda los datos de la nueva configuración y reinicia la aplicación.</li><li>• <i>CrearFormas</i>: Crea los elementos gráficos de la aplicación.</li><li>• <i>CrearPanel</i>: Crea las pantallas.</li><li>• <i>CrearLabel</i>: Crea las etiquetas.</li><li>• <i>CrearBotones</i>: Crea los botones.</li></ul>

Tabla 7.36: Clase ComponenteTeclado del paquete Escritura textual.

## PAQUETE ESCRIBIR MÉTODO CATEGORÍAS

ARCHIVOS
<ul style="list-style-type: none"><li>• <i>Principal_cat.xml</i></li></ul>

Tabla 7.37: Archivos .xml del paquete Escribir método categorías.

## PAQUETE ESCRIBIR MÉTODO VOCALES

ARCHIVOS
<ul style="list-style-type: none"><li>• <i>Principal_voc.xml</i></li></ul>

Tabla 7.38: Archivos .xml del paquete Escribir método vocales.

## PAQUETE ESCRIBIR MÉTODO VARIABLE

ARCHIVOS
<ul style="list-style-type: none"><li>• <i>Principal_var.xml</i></li></ul>

Tabla 7.39: Archivos .xml del paquete Escribir método variable.

## PAQUETE ESCRIBIR MÉTODO IMAGENES

### Clase Componente Teclado

#### Atributos

#### Métodos

- *CargarConfiguracion()*: Guarda los datos de la nueva configuración y reinicia la aplicación.
- *CrearFormas*: Crea los elementos gráficos de la aplicación.
- *CrearPanel*: Crea las pantallas.
- *CrearLabel*: Crea las etiquetas.
- *CrearBotones*: Crea los botones.
- *CrearImagen*: Crea las imágenes.
- *AsignarImagenBoton*: Asigna cada imagen a su botón correspondiente.
- *CambiarImagenes*: Cambia la imagen a los botones, cuando la pantalla tiene rotaciones.
- *AniadirFuncionalidadBotonImagen*: Método que da a cada botón que aparecerá en la pantalla su funcionalidad.

Tabla 7.40: Clase ComponenteTeclado del paquete Escribir método imágenes.

### Clase Pantalla

#### Atributos

#### Métodos

- *Pantalla*: Crea el objeto Pantalla

Tabla 7.41: Clase Pantalla del paquete Escribir método imágenes.

### Clase Botón

#### Atributos

#### Métodos

- *Boton*: Crea el objeto Boton.

Tabla 7.42: Clase Botón del paquete Escribir método imágenes.

### Clase Imagen

#### Atributos

#### Métodos

- *Img*: path de la imagen.
- *Imgfoco*: path de la imagen con foco.
- *Imagen*: Crea el objeto Imagen.

Tabla 7.43: Clase Imagen del paquete Escribir método imágenes.

## Clase BotonImagen

<b>Atributos</b>	<ul style="list-style-type: none"><li>• <i>Escribe</i>: Crea el objeto Boton.</li><li>• <i>Panel</i>: número de panel donde está el botón.</li><li>• <i>Funcion</i>: Funcionalidad del botón.</li><li>• <i>Pantalla</i>: Pantalla a la que dirige el boton.</li><li>• <i>Imagen</i>: Imagen asociada al botón.</li><li>• <i>ImagenFoco</i>: Imagen con foco asociada al botón.</li></ul>
<b>Métodos</b>	<ul style="list-style-type: none"><li>• <i>BotonImagen</i>: Crea el objeto BotonImagen.</li><li>• <i>ModificarBoton</i>: Permite modificar ciertos atributos del botón.</li></ul>

Tabla 7.44: Clase BotonImagen del paquete Escribir método imágenes.

## ARCHIVOS

- *principal\_img.xml*

Tabla 7.45: Archivos .xml del paquete Escribir método imágenes.

## PAQUETE FUNCIONALIDAD EXTRA

### Clase Componente Teclado

<b>Atributos</b>	<ul style="list-style-type: none"><li>• <i>cambios</i>: Devuelve si hay cambios o no</li></ul>
<b>Métodos</b>	<ul style="list-style-type: none"><li>• <i>AnadirXML</i>: Permite la inclusión de funcionalidades extra</li><li>• <i>ActualizaMenu</i>: actualiza el menú de la aplicación con los botones de las funcionalidades extra.</li></ul>

Tabla 7.46: Clase ComponenteTeclado del paquete Funcionalidad extra.

## PAQUETE FUNCIONALIDAD SIMPLE

### Clase Componente Teclado

<b>Atributos</b>	
<b>Métodos</b>	<ul style="list-style-type: none"><li>• <i>MostrarPantalla</i>: Muestra la pantalla indicada.</li><li>• <i>BotonClick</i>: Permite las funcionalidades básicas de los botones</li></ul>

Tabla 7.47: Clase ComponenteTeclado del paquete Funcionalidad simple.

---

## PAQUETE OPCIÓN COLORES

Clase Componente Teclado	
<i>Atributos</i>	
<i>Métodos</i>	<ul style="list-style-type: none"><li>• <i>MostrarPantalla</i>: Muestra la pantalla indicada.</li><li>• <i>BotonClick</i>: Permite las funcionalidades básicas de los botones, así como las funcionalidades de cambio de color de los elementos.</li></ul>

Tabla 7.48: Clase ComponenteTeclado del paquete Opción colores.

## PAQUETE OPCIÓN FRASES

Clase Componente Teclado	
<i>Atributos</i>	
<i>Métodos</i>	<ul style="list-style-type: none"><li>• <i>MostrarPantalla</i>: Muestra la pantalla indicada.</li><li>• <i>BotonClick</i>: Permite las funcionalidades básicas de los botones, así como las funcionalidades relacionadas con la gestión de frases.</li></ul>

Tabla 7.49: Clase ComponenteTeclado del paquete Opción frases.

## PAQUETE OPCIÓN COMPLETA

Clase Componente Teclado	
<i>Atributos</i>	
<i>Métodos</i>	<ul style="list-style-type: none"><li>• <i>MostrarPantalla</i>: Muestra la pantalla indicada.</li><li>• <i>BotonClick</i>: Permite las funcionalidades básicas de los botones, así como las funcionalidades de cambio de colores y gestión de frases.</li></ul>

Tabla 7.50: Clase ComponenteTeclado del paquete Opción completa.

---

## PAQUETE COLORES

### Clase Componente Teclado

<b>Atributos</b>	<ul style="list-style-type: none"><li>• <i>cambios</i>: Devuelve si hay cambios o no</li></ul>
<b>Métodos</b>	<ul style="list-style-type: none"><li>• <i>DetectarCambioColores</i>.</li><li>• <i>CambiarColor</i>.</li><li>• <i>ActualizarColorFondo</i>.</li><li>• <i>ActualizarColorBoton</i>.</li><li>• <i>ActualizarColorTexto</i>.</li><li>• <i>ActualizarColorTextoBoton</i>.</li></ul>

Tabla 7.51: Clase ComponenteTeclado del paquete Colores.

### Clase Pantalla

<b>Atributos</b>	
<b>Métodos</b>	<ul style="list-style-type: none"><li>• <i>ModificarColorFondo</i>.</li></ul>

Tabla 7.52: Clase Pantalla del paquete Colores.

### Clase Etiquetas

<b>Atributos</b>	
<b>Métodos</b>	<ul style="list-style-type: none"><li>• <i>ModificarColorFondo</i>.</li><li>• <i>ModificarColorLetra</i>.</li></ul>

Tabla 7.53: Clase Etiquetas del paquete Colores.

## ARCHIVOS

- *Colores.xml*

Tabla 7.54: Archivos .xml del paquete Colores.

## PAQUETE SOPORTE FRASES

### Clase Componente Teclado

<b>Atributos</b>	<ul style="list-style-type: none"><li>• <i>ListaFrases</i>: Lista de frases disponibles..</li></ul>
<b>Métodos</b>	<ul style="list-style-type: none"><li>• <i>CargaFrases</i>: Se cargan las frases del fichero a la lista.</li><li>• <i>CrearListaSelección</i>: Permite la creación de una lista que contenga frases, no sólo métodos de escritura o tiempos de barrido.</li><li>• <i>AñadirFrase</i>: Permite añadir una frase de la lista al cuadro de texto.</li><li>• <i>GuardaFrase</i>: Permite guardar el contenido del cuadro de texto como una nueva frase.</li><li>• <i>borrarFrase</i>: Permite borrar una frase de la lista.</li><li>• <i>escribeFichero</i>: Graba las frases de la lista al fichero.</li></ul>

Tabla 7.55: Clase ComponenteTeclado del paquete Soporte frases.

### ARCHIVOS

- *frases.xml*
- *frases.txt*

Tabla 7.56: Archivos .xml del paquete Soporte frases.

## PAQUETE LISTAS BÁSICAS

### Clase Componente Teclado

<b>Atributos</b>	
<b>Métodos</b>	<ul style="list-style-type: none"><li>• <i>CrearListasSelección</i>: Permite la creación de listas que contengan tiempos de barrido o métodos de escritura.</li></ul>

Tabla 7.57: Clase ComponenteTeclado del paquete Listas básicas.

---

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

El componente software estará formado por pantallas con un número variable de botones en cada una, mediante los cuales el usuario interactuará con la aplicación. Son los siguientes:

- **Botones 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: botón A, botón B, botón 7, etc.).
- **Botones 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 se reproduzca mediante voz el texto escrito en el cuadro de texto.
- **Cambiar la configuración:** La pulsación de un botón con esta funcionalidad hará que cambie la apariencia de las pantallas de la aplicación.
- **Cambiar el color de los elementos:** La pulsación de un botón con esta funcionalidad hará que cambie el color de los botones, la letra o el fondo de la aplicación según las necesidades del usuario.
- **Usar frases:** La pulsación de un botón con esta funcionalidad permitirá el uso de frases ya establecidas, agregar nuevas frases escritas por el usuario para su posterior reutilización y eliminar aquellas frases almacenadas que considere el usuario que ya no van a serle de utilidad.
- **Salir de la aplicación:** La pulsación de un botón con esta funcionalidad hará que finalice una sesión de trabajo.

A continuación 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.

### Pantalla inicial

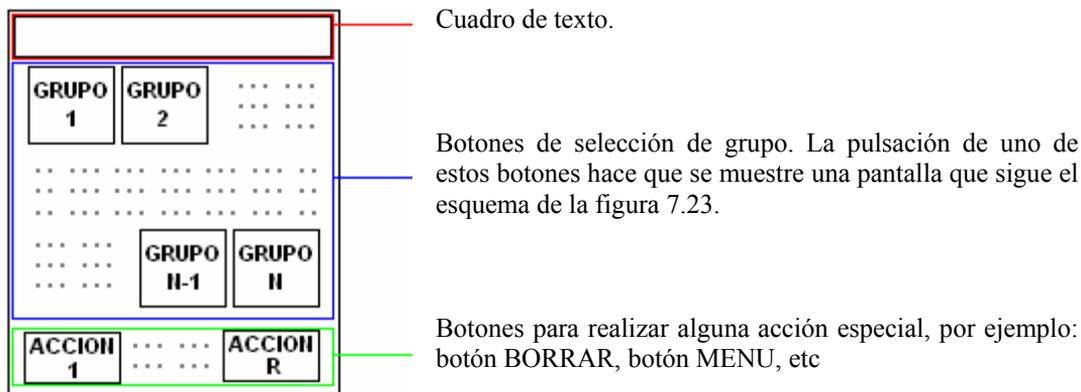


Figura 7.22: Esquema interfaz gráfica de usuario de la pantalla inicial.

### Pantalla de un grupo

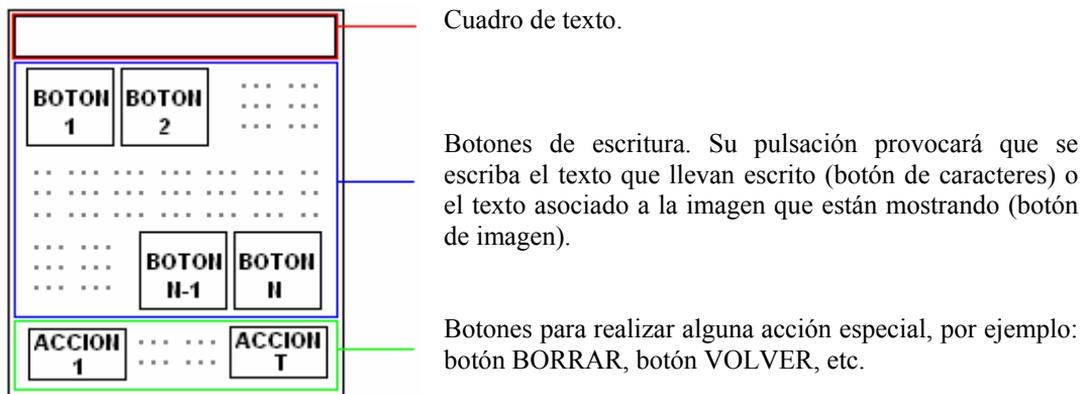


Figura 7.23: Esquema interfaz gráfica de usuario de una pantalla de un grupo

### Pantalla menú

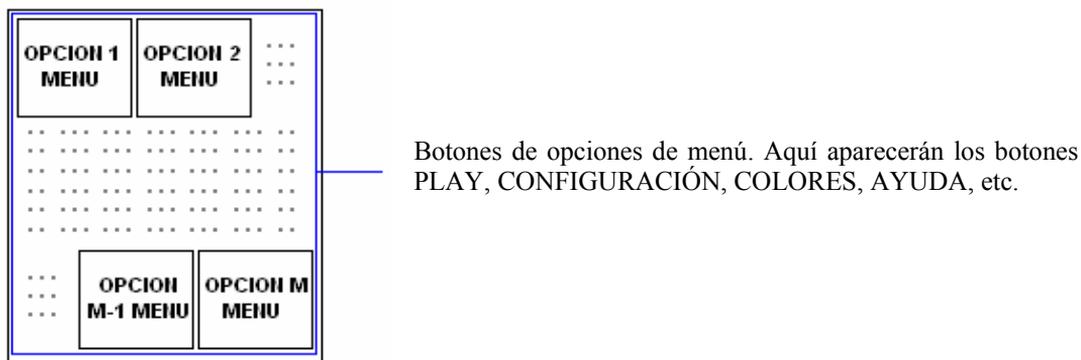


Figura 7.24: Esquema interfaz gráfica de usuario de la pantalla de menú



## Capítulo 8

# IMPLEMENTACIÓN DE LA LÍNEA DE PRODUCTOS

### 8.1 Introducción

Durante esta fase de implementación de la línea de productos, se plasma el diseño de la arquitectura de la misma mediante un framework orientado a objetos. Este framework dará origen a las distintas aplicaciones que podrán generarse mediante la línea de productos, dependiendo de las características que hayan sido elegidas para dicha aplicación en concreto.

En este capítulo de la memoria, se expondrán las técnicas usadas para llevar a cabo esta fase de manera satisfactoria. Asimismo, se ofrecerá información acerca del entorno de programación y el hardware y software utilizado para ello.

A la hora de integrar las funcionalidades del proyecto en un único desarrollo, se encontraron varias dificultades. La idea era reutilizar código ya existente en los proyectos de fin de carrera estudiados. Sin embargo, nos encontramos un código en el que se mezclaban, en cada uno de sus métodos, todas las funcionalidades. Es decir, en un mismo método se trabajaba con imágenes, colores, varios métodos de escritura, etc. Esto nos supuso un problema, pues tuvimos que estudiar a fondo el código y realizar todos los cambios convenientes para aislar cada funcionalidad. Pero no sólo adaptamos código, también tuvimos que generar código nuevo con el fin de enlazar algunas funcionalidades, agregar nuevas y mejorar algunos aspectos.

### 8.2 Entorno de programación y lenguaje

Para extender la trazabilidad de la línea de productos hasta los modelos de implementación, utilizamos el concepto de clases parciales, que permiten que la declaración de una clase se realice en varios archivos de código fuente. De esta forma, será tarea del compilador tomar las porciones de los distintos archivos y fundirlas en una única entidad. La intención es mantener la correspondencia, no sólo entre características y paquetes de diseño, sino también con la estructura del código.

En nuestro modelo de desarrollo utilizamos el mecanismo de clases parciales de C# para implementar la línea de productos dentro de la estructura de soluciones y paquetes de la plataforma MS Visual Studio 2005. Si el *framework* que implementa la arquitectura de la línea de productos está organizado en paquetes de clases parciales (un paquete base y tantos paquetes auxiliares como variaciones existen), para derivar una aplicación concreta, basta con importar o referenciar los paquetes que se correspondan directamente con la configuración elegida en el modelo de características. Estos paquetes opcionales pueden añadirse o no al proyecto que representa una

aplicación concreta de la línea de productos utilizando los ficheros de configuración del compilador.

En la figura 8.1 se puede ver un fragmento del Visual Studio 2005 en el que se aprecia la organización en paquetes de las clases parciales.

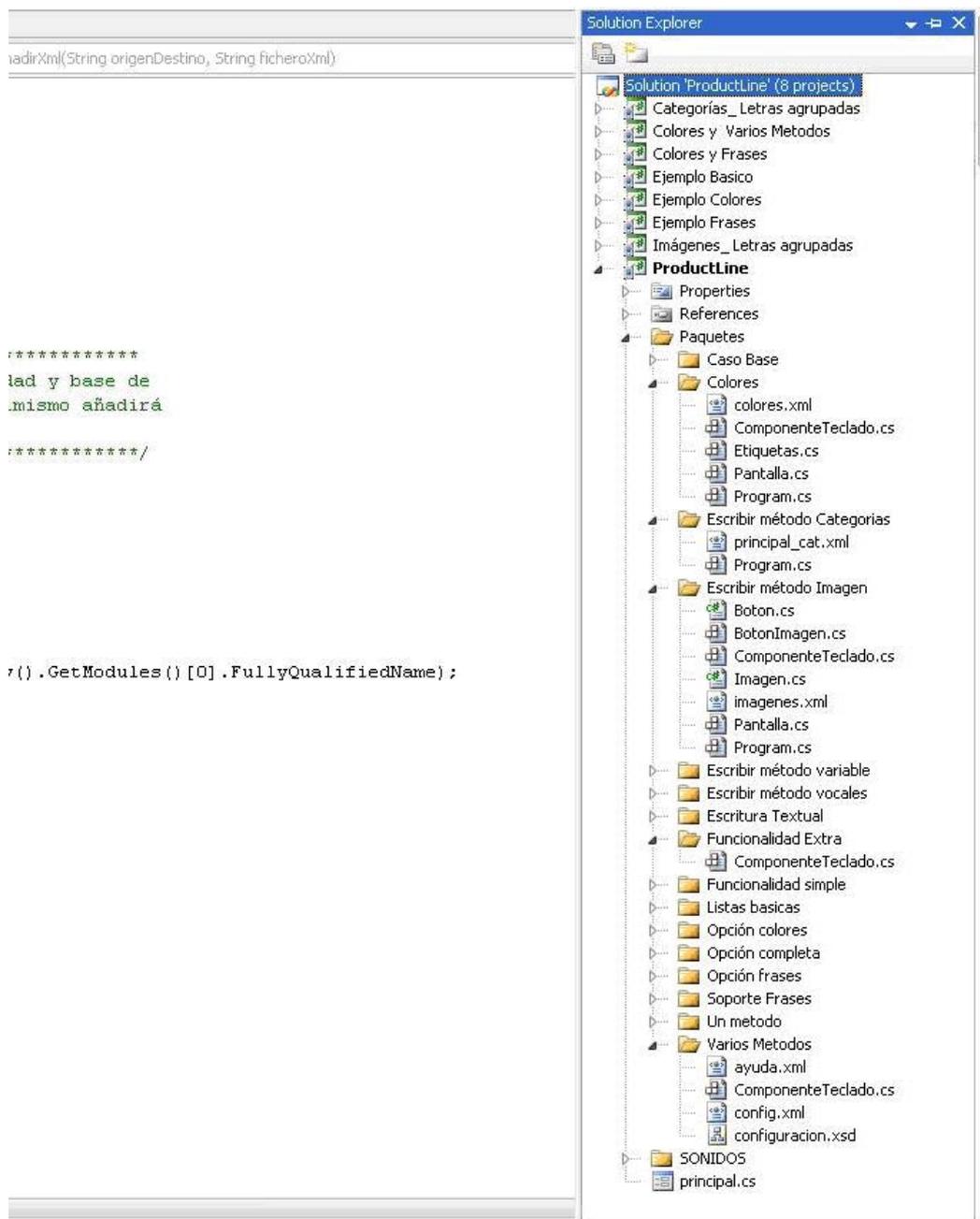


Figura 8.1: Aspecto de la implementación de la línea de productos en Visual Studio .NET 2005

Para la realización del proyecto se ha decidido usar ficheros XML como almacenamiento de la información que necesitará la aplicación. El contenido de estos ficheros determinará tanto la apariencia del teclado como la funcionalidad del mismo.

La elección de los XML se ha tomado por diversos motivos:

- 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: Tecnología empleada).

### 8.2.1 Utilización de los ficheros XML

El principal motivo de su utilización es hacer la aplicación lo más configurable posible. 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 desplegadas.
- Etiquetas de texto.
- Botones de radio.

Varios de los proyectos de fin de carrera estudiados utilizaban un único fichero .xml para almacenar toda la información sobre la interfaz del comunicador. En la línea de productos, la interfaz es un elemento variable, pues se podrán crear aplicaciones que necesiten distintas interfaces.

Para poder expresar esta variabilidad, se decidió dividir el fichero .xml en varias partes, de modo que pudieran combinarse para conformar la interfaz que se necesitase en cada aplicación. ¿Cuántos ficheros XML se van a necesitar? Esa sería la pregunta clave. La línea de productos está basada en paquetes, esto es, en partes comunes, opcionales y alternativas. Haremos algo similar con el archivo .xml de la interfaz, dividiéndolo en las siguientes partes diferenciadas:

- Pantalla de Bienvenida + Método de escritura.
- Pantalla de Menú.
- Pantalla de Configuración.
- Pantallas de Ayuda.
- Pantallas de Frases (si las hay)
- Pantallas de Selección de Colores (si las hay)

Hagamos un estudio de las combinaciones posibles:

⇒ **Parte Común:**

- La aplicación mostrará un mensaje de bienvenida.
- Habrá un único método de escritura: método de letras agrupadas.
- El menú de la aplicación constará de los siguientes botones:
  - PLAY
  - CONFIGURACION
  - AYUDA
  - VOLVER
  - SALIR
- Al pulsar el botón CONFIGURACION habrá las siguientes posibilidades:
  - Barrido Activado (posibilidad de marcar o desmarcar el barrido).
  - Tiempo de barrido.
- Ayuda: textos de ayuda correspondientes a los elementos anteriormente señalados.

⇒ **Parte Opcional:**

- *Colores*: se añadirá la posibilidad de modificar los colores de las pantallas y botones. Para ello se agrega un nuevo botón al menú:
  - PLAY
  - CONFIGURACION
  - COLORES
  - AYUDA
  - VOLVER
  - SALIR

El tamaño ahora de los botones será diferente a los de la parte común, puesto que hay un botón más.

- Añadir la posibilidad de escoger entre *varios métodos de escritura*: para ello se añade en Configuración una lista de selección indicando los posibles métodos de escritura disponibles, quedando entonces:
  - Barrido Activado.
  - Tiempo de barrido.
  - Diseño de pantalla (lista de selección de los métodos de escritura).

En cuanto a la Ayuda, la información sobre configurar se verá modificada, puesto que deberá informar sobre cómo cambiar de un método de escritura a otro.

- Añadir la posibilidad de incluir y tratar *frases*. Para ello se añade un botón a menú:
  - PLAY
  - CONFIGURACION
  - FRASES
  - AYUDA
  - VOLVER
  - SALIR

- Como un caso especial de método de escritura, se añadirá la posibilidad de escritura mediante el *método de las imágenes*, en el cual, las pantallas del teclado estarán formadas por botones de imágenes. Estas imágenes pertenecen a un alfabeto conocido por el usuario. En este método de escritura el barrido es obligatorio y las opciones de tratamiento de frases y cambio de colores están desactivadas. Por tanto, el menú quedaría:
  - PLAY
  - CONFIGURACION
  - AYUDA
  - VOLVER
  - SALIR

Y la configuración:

- Tiempo de barrido.
- Diseño de pantalla (lista de selección de los métodos de escritura).

Hay que tener en cuenta, que puede haber múltiples combinaciones de estas opciones o características: Puede haber un componente que permita varios métodos de escritura, la gestión de frases y la posibilidad de cambiar los colores, por ejemplo. En este caso, el menú se vería ahora con dos botones más que en el caso original:

- PLAY
- CONFIGURACION
- COLORES
- FRASES
- AYUDA
- VOLVER
- SALIR

Por tanto, si por ejemplo, se desea el método de letras agrupadas y categorías, la información obtenida del XML sobre la pantalla de configuración de la aplicación será diferente al caso original, en el que no se contemplaba la posibilidad de cambiar de un método de escritura a otro. Por consiguiente, se harán dos archivos .xml: uno que contenga la pantalla de configuración original y otro con una pantalla de configuración que atienda a varios métodos de escritura para elegir. El archivo .xml que se elija dependerá de la aplicación que se desee crear.

Cada archivo .xml estará ubicado en el paquete que le corresponda según su funcionalidad. Se muestra a continuación, una tabla en la que aparecen los paquetes que contienen archivos .xml y la información que aportan.

Para ver de una forma detallada las estructuras de los ficheros consúltese el Apéndice B: Estructura de los ficheros XML.

PAQUETES	FICHEROS XML
<b>Caso Base</b>	<ul style="list-style-type: none"> <li>• <i>configuracion.xml</i>: define qué método de escritura lee la aplicación para crear el teclado, si hay barrido o no, posibles configuraciones en que podrá trabajar y aspectos globales de apariencia como el color de fondo.</li> <li>• <i>menu.xml</i>: forma el menú correspondiente a la parte común.</li> <li>• <i>principal_la.xml</i>: define la pantalla de bienvenida y crea el teclado de acuerdo al método de escritura letras agrupadas.</li> </ul>
<b>Colores</b>	<ul style="list-style-type: none"> <li>• <i>colores.xml</i>: crea las pantallas donde se facilita el cambio de colores de determinados elementos de la aplicación.</li> </ul>
<b>Escribir método Categorías</b>	<ul style="list-style-type: none"> <li>• <i>principal_cat</i>: define la pantalla de bienvenida y crea el teclado de acuerdo al método de escritura por categorías.</li> </ul>
<b>Escribir método Imagen</b>	<ul style="list-style-type: none"> <li>• <i>imagenes.xml</i>: define la pantalla de bienvenida y crea el teclado de acuerdo al método de escritura por imágenes.</li> </ul>
<b>Escribir método Variable</b>	<ul style="list-style-type: none"> <li>• <i>principal_var</i>: define la pantalla de bienvenida y crea el teclado de acuerdo al método de escritura por el método variable.</li> </ul>
<b>Escribir método vocales</b>	<ul style="list-style-type: none"> <li>• <i>principal_voc</i>: define la pantalla de bienvenida y crea el teclado de acuerdo al método de escritura por el método de las vocales.</li> </ul>
<b>Soporte Frases</b>	<ul style="list-style-type: none"> <li>• <i>frases.xml</i>: crea las pantallas donde poder usar y gestionar frases.</li> </ul>
<b>Un método</b>	<ul style="list-style-type: none"> <li>• <i>ayuda.xml</i>: define la ayuda que ofrecerá la aplicación en cuanto al manejo del método de letras agrupadas, únicamente.</li> <li>• <i>config.xml</i>: creará la pantalla proveniente del botón CONFIGURAR. Se mostrará la posibilidad de marcar o no el barrido y un listado de los tiempos de barrido disponibles.</li> </ul>
<b>Varios métodos</b>	<ul style="list-style-type: none"> <li>• <i>ayuda.xml</i>: define la ayuda que ofrecerá la aplicación. Entre otras cosas, el de cómo cambiar de un método de escritura a otro.</li> <li>• <i>config.xml</i>: creará la pantalla proveniente del botón CONFIGURAR. Se mostrará la posibilidad de marcar o no el barrido, un listado de los tiempos de barrido disponibles y los posibles métodos de escritura.</li> </ul>

Tabla 8.1: Relación de paquetes y archivos XML

## 8.2.2 Tratamiento de los archivos XML

A la hora de trabajar con el mecanismo de combinación de paquetes (package merge), se dispone de la gran ventaja de su implementación directa gracias a las clases parciales de C#. Así, teniendo una parte de código de una clase parcial 'A' en un paquete P1, y otra parte de código distinto de esa misma clase parcial 'A' en un paquete P2, el compilador de Visual Studio 2005, combina automáticamente el código de ambos paquetes en una misma clase.

Nos encontramos con el problema de que no podemos aplicar el mismo procedimiento con los ficheros XML, ya que al no ser éstos código C#, el compilador no nos hace el trabajo automáticamente. Tenemos que idear un método para unir los fragmentos de XML necesarios en un único fichero XML, de acuerdo con los paquetes que se hayan seleccionado previamente.

Para ello, se contempló en principio dos soluciones:

- Utilizar las librerías XML de .NET para ir leyendo nodo por nodo cada fragmento de XML e ir escribiendo nodo por nodo el fichero XML final.
- Utilizar objetos DATASET para almacenar en ellos los fragmentos de XML, creando así una base de datos de los elementos del comunicador.

Se decidió utilizar la opción del almacenamiento por medio de DATASETS debido a que nos proporciona un método de tratamiento mucho más potente, práctico y simple. Al tener almacenados los elementos de la aplicación en forma de base de datos relacional, podemos modificarlos según lo que necesitemos en cada momento, al contrario que la opción descartada. Dado el carácter variable de las interfaces de los miembros de la línea de productos, el método de DATASETS nos permite tratar con esa variabilidad de un modo ágil y eficiente.

Además, se quería que la línea de productos fuera lo más fiel posible a los proyectos fin de carrera de los cuales se deriva. Por ello, los datos almacenados en los dataset, darán lugar a un fichero .xml final, desde donde la aplicación leerá los datos que necesite, como en los proyectos estudiados. El proceso consistiría en:

1. Cargar los DATASETS con los .xml correspondientes.
2. Modificar si es preciso esos DATASETS, bien sea añadiendo nuevos fragmentos de archivos .xml, bien sea modificando los elementos del mismo directamente, para adaptar la interfaz del comunicador.
3. Grabar el contenido del DATASET a un fichero .xml conservando su estructura, para que la aplicación pueda utilizarlo sin mayor problema.

Para llevar a cabo el método de paso de información a través de los dataset, se necesitaron los esquemas .xsd de los archivos .xml de configuración e interfaz gráfica.

Un DATASET se puede cargar de forma automática con un fichero .xml. Al cargar dicho fichero, se carga en el mismo DATASET su esquema xsd, bien sea de forma implícita o explícita. En nuestro caso, se tenía que, de alguna manera, combinar esos .xml para lo que se necesitaba disponer del esquema final del fichero combinado para evitar la pérdida de información en el proceso.

El esquema .xsd de la figura 8.20 muestra de forma visual las relaciones existentes entre los distintos componentes de la interfaz del comunicador:

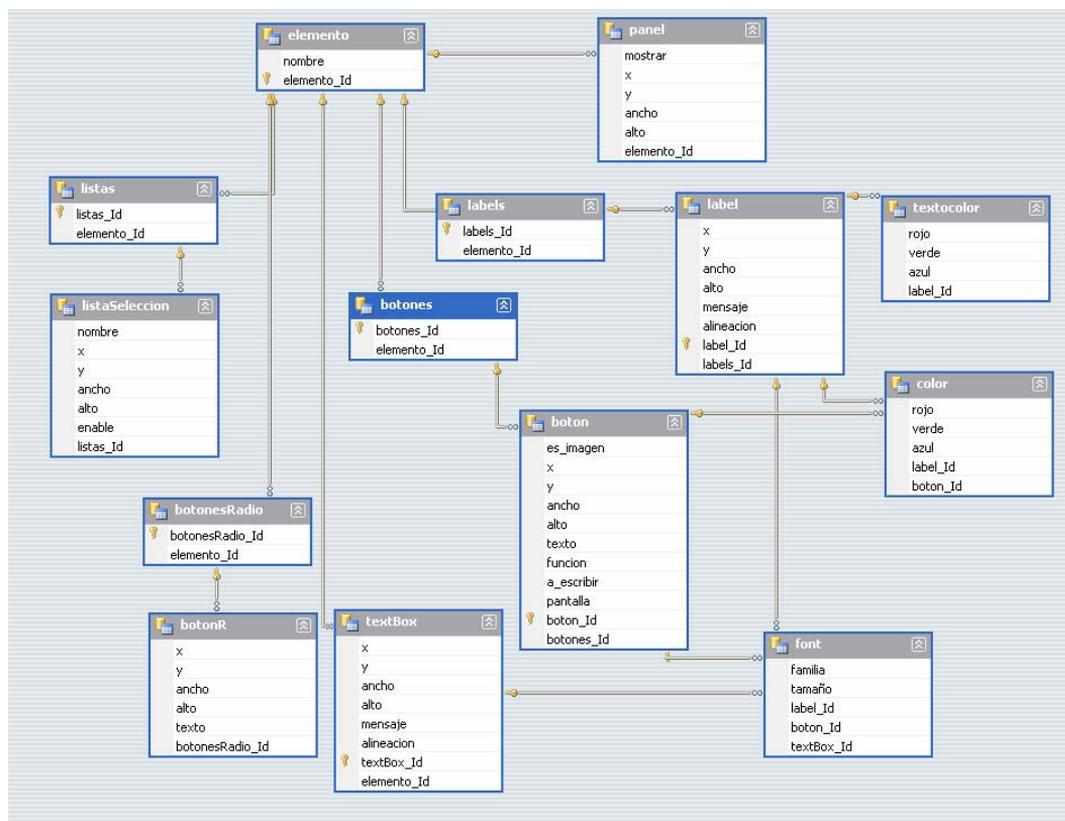


Figura 8.2: Esquema XSD de la interfaz gráfica del comunicador.

Se podrá obtener más información sobre objetos DATASET y su diferencia con los objetos XmlDocument en el apéndice A: Tecnología empleada.

### 8.3 Instanciación manual de productos

Una vez obtenidos todas las características, paquetes y ficheros .xml necesarios para producir cualquier aplicación dentro de esta línea de productos software de comunicadores para personas discapacitadas, se han elaborado a través de la línea de productos resultante, una serie de productos a modo de ejemplos o prototipos, para comprobar el correcto funcionamiento del desarrollo anterior. Éstos son:

- **Categorías\_Letras agrupadas:** Este proyecto contempla la posibilidad de cambiar el método de escritura de la aplicación. Parte del método de escritura de letras agrupadas o conjuntos, que es el común de todos los proyectos, y da la posibilidad de cambiar al método de escritura de categorías.
- **Colores y varios métodos:** Contempla la posibilidad de cambiar de color los elementos de la aplicación. Asimismo, se puede modificar la configuración, permitiendo alternar entre el método de letras agrupadas y el de categorías.
- **Colores y Frases:** Éste únicamente tendrá el método de escritura mediante letras agrupadas. Sin embargo, tendrá funcionalidades añadidas: colores, que permitirá cambiar el color de los elementos de la aplicación; y frases, que permitirá agregar frases predefinidas al cuadro de texto, así como gestionar dichas frases (borrarlas, guardar frases nuevas...).
- **Ejemplo básico:** Contendrá, únicamente, toda la funcionalidad común, es decir, todas las características comunes de la línea de productos.
- **Ejemplo colores:** Es el caso anterior, características comunes, más la posibilidad de modificar los colores de varios elementos de la aplicación.
- **Ejemplo frases:** Contendrá la funcionalidad común a la línea de productos, más la posibilidad de utilizar y gestionar frases.
- **Imágenes letras agrupadas:** sería parecido al caso de Categorías\_Letras agrupadas, en cuanto a disponer de varios métodos de escritura a elegir. Sin embargo, este caso combina un método de escritura textual, con caracteres tipográficos, con un método de escritura mediante imágenes.
- **Letras agrupadas y vocales:** Parte del método de escritura de letras agrupadas y da la posibilidad de cambiar al método de escritura mediante vocales.

Se puede ver en la tabla 8.2 los paquetes que serían necesarios para llevar a cabo cada uno de estos prototipos.

	Ejemplo Básico	Ejemplo Colores	Ejemplo Frases	Categ. y letras agrup.	Imágenes Letras agrup.	Colores y varios métodos	Colores y Frases	Letras agrup. y vocales
Funcionalidad Extra		✓	✓			✓	✓	
Caso Base	✓	✓	✓	✓	✓	✓	✓	✓
Colores		✓				✓	✓	
E. método Imagen					✓			
E. método variable								
E. método vocales								✓
Escritura Textual	✓	✓	✓	✓		✓	✓	✓
Listas básicas	✓	✓		✓	✓	✓		✓
E. método Categorías				✓		✓		
Funcionalidad Simple	✓			✓	✓			✓
Opción colores		✓				✓		
Opción completa							✓	
Opción frases			✓					
Soporte frases			✓				✓	
Un método	✓	✓	✓				✓	
Varios Métodos				✓	✓	✓		✓

Tabla 8.2: Relación de prototipos y paquetes necesarios para cada uno.

Ahora bien, ¿cómo añadir los paquetes necesarios a nuestro proyecto? Para ello, habrá que descargar el proyecto y editar el archivo .csproj del mismo.

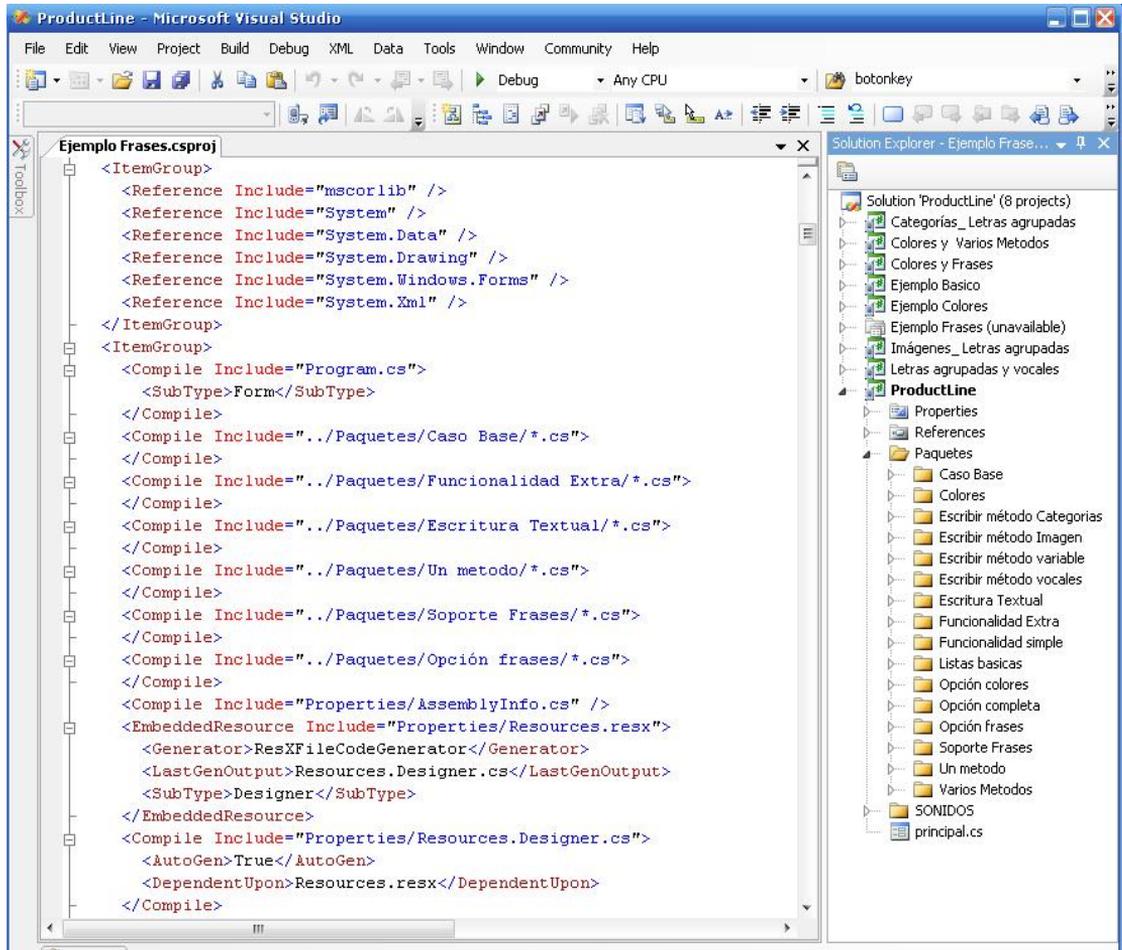


Figura 8.3: Ejemplo de un fichero .csproj.

En la figura 8.3, podemos ver un ejemplo del resultado tras editar el .csproj del proyecto Ejemplo Frases. Se puede ver, como basta con añadir la ruta de los paquetes necesarios:

```
<Compile Include="..\Paquetes/Caso Base/*.cs">
</Compile>
<Compile Include="..\Paquetes/Funcionalidad Extra/*.cs">
</Compile>
<Compile Include="..\Paquetes/Escritura Textual/*.cs">
</Compile>
<Compile Include="..\Paquetes/Un metodo/*.cs">
</Compile>
<Compile Include="..\Paquetes/Soporte Frases/*.cs">
</Compile>
<Compile Include="..\Paquetes/Opción frases/*.cs">
</Compile>
```

De igual forma se hará con los archivos .xml necesarios:

```
<Content Include="..\Paquetes\Soporte Frases\frases.xml">  
  <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>  
  <Link>frases.xml</Link>  
</Content>
```

```
<Content Include="..\Paquetes\Soporte Frases\frases.txt">  
  <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>  
  <Link>frases.txt</Link>  
</Content>
```

```
<Content Include="..\Paquetes\Caso Base\configuracion.xml">  
  <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>  
  <Link>configuracion.xml</Link>  
</Content>
```

```
<Content Include="..\Paquetes\Un metodo\ayuda.xml">  
  <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>  
  <Link>ayuda.xml</Link>  
</Content>
```

```
<Content Include="..\Paquetes\Un metodo\config.xml">  
  <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>  
  <Link>config.xml</Link>  
</Content>
```

```
<Content Include="..\Paquetes\Caso Base\menu.xml">  
  <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>  
  <Link>menu.xml</Link>  
</Content>
```

```
<Content Include="..\Paquetes\Caso Base\principal_la.xml">  
  <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>  
  <Link>principal_la.xml</Link>  
</Content>
```

```
<Content Include="..\Paquetes\Caso Base\letrasagrupadas.xsd">  
  <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>  
  <Link>letrasagrupadas.xsd</Link>  
  <SubType>Designer</SubType>  
</Content>
```

Con la idea de hacer este proyecto de fin de carrera lo más reutilizable posible, se han aplicado clases parciales y compilación condicional. De esta forma, un futuro desarrollador de un producto de esta línea de productos, simplemente deberá de añadir unas pocas líneas de código a la clase que inicia el sistema.

La idea es usar el mismo código en todos los casos, como si de una plantilla se tratara. Para lograrlo, se ha aplicado clases parciales a la clase lanzadora, Program.cs. De esta forma, el paquete Caso Base contendrá un Program.cs en el que se incluirá el código necesario para la parte común de la aplicación. El paquete Escribir método de Categorías contendrá un Program.cs que contendrá las líneas de código para indicarle que agregue el método de categorías, entre otras. El paquete Opción completa, tendrá un Program.cs con las llamadas a métodos para que se añadan los xml de colores y de frases, y que actualice el menú con los dos botones de más necesarios. Y así, con el resto de posibilidades.

A continuación se muestra la tabla 8.3, donde indicamos en qué paquetes se han necesitado crear clases parciales de Program.cs, el nombre del método y su función.

PAQUETE	NOMBRE DEL MÉTODO	DESCRIPCIÓN
Caso base	principal_1()	Código necesario para la creación del componente y la formación del método de letras agrupadas.
Escribir método Categorías	principal_2()	Código necesario para disponer de un segundo método de escritura: método categorías.
Colores	principal_3()	Código necesario para agregar el xml correspondiente a los elementos necesarios para gestionar los colores, y para añadir el botón "COLORES" al menú de la aplicación.
Soporte Frases	principal_4()	Código necesario para agregar el xml correspondiente a los elementos necesarios para gestionar frases y adaptar el menú con un nuevo botón llamado "FRASES".
Escribir método Imágenes	principal_5()	Código necesario para crear el método de escritura mediante imágenes.
Opción completa	principal_6()	Código necesario para añadir los xml de colores y frases y agregar dos botones en el menú.
Escribir método vocales	principal_7()	Código necesario para disponer de un segundo método de escritura: método de las vocales.

Tabla 8.3: Clases parciales de Program.cs

Una vez repartido el código, necesitaremos un gestor condicional, de forma que sólo se compilen aquellas líneas de código que serán necesarias para lanzar la aplicación.

Las directivas #if, #elif, #else y #endif, ofrecen la posibilidad de compilación condicional controlando el comportamiento del preprocesador, de forma que éste puede ignorar o compilar determinadas líneas del código en función de ciertas condiciones que son evaluadas durante el preproceso.

Por tanto, un fragmento de esta clase lanzadora sería:

```
this.princ = new Paquetes.principal_0();
this.princ.principal_1();

#if CATEGORIAS
    this.princ.principal_2();
#endif

#if VOCALES
    this.princ.principal_7();
#endif

#if COLORES
    this.princ.principal_3();
#endif

#if FRASES
    this.princ.principal_4();
    this.componente.cargafrases();
#endif

#if IMAGENES
    this.princ.principal_5();
#endif

#if MENU_COMPLETO
    this.princ.principal_6();
    this.componente.cargafrases();
#endif
```

Es decir, mediante constantes logramos que se compile un código u otro. Así, un desarrollador que quiera tener una aplicación con dos posibles métodos de escritura: letras agrupadas y categorías, y además, que se pueda cambiar los colores de dicha aplicación, únicamente deberá agregar al comienzo de la clase lanzadora de su proyecto:

```
#define CATEGORIAS
```

```
#define COLORES
```

A continuación se muestra la tabla 8.4, donde se exponen los distintos productos o ejemplos elaborados mediante la línea de productos desarrollada, así como los paquete que componen cada uno de ellos y las constantes que se han definido para su total y correcto funcionamiento.

PROYECTOS	PAQUETES NECESARIOS	CONSTANTES A AÑADIR
<b>Ejemplo Básico</b>	<ul style="list-style-type: none"> <li>• Caso base</li> <li>• Escritura Textual</li> <li>• Un método</li> <li>• Listas básicas</li> <li>• Funcionalidad simple</li> </ul>	
<b>Ejemplo Colores</b>	<ul style="list-style-type: none"> <li>• Caso base</li> <li>• Escritura Textual</li> <li>• Un método</li> <li>• Listas básicas</li> <li>• Funcionalidad Extra</li> <li>• Colores</li> <li>• Opción colores</li> </ul>	COLORES
<b>Ejemplo Frases</b>	<ul style="list-style-type: none"> <li>• Caso base</li> <li>• Escritura Textual</li> <li>• Un método</li> <li>• Funcionalidad Extra</li> <li>• Soporte Frases</li> <li>• Opción frases</li> </ul>	FRASES
<b>Categorías y letras agrupadas</b>	<ul style="list-style-type: none"> <li>• Caso base</li> <li>• Escritura Textual</li> <li>• Varios métodos</li> <li>• Listas básicas</li> <li>• Funcionalidad simple</li> <li>• Escribir método Categorías</li> </ul>	CATEGORIAS
<b>Imágenes y letras agrupadas</b>	<ul style="list-style-type: none"> <li>• Caso base</li> <li>• Listas básicas</li> <li>• Funcionalidad simple</li> <li>• Varios métodos</li> <li>• Escribir método Imagen</li> </ul>	IMAGENES
<b>Colores y varios métodos</b>	<ul style="list-style-type: none"> <li>• Caso base</li> <li>• Escritura Textual</li> <li>• Funcionalidad Extra</li> <li>• Listas básicas</li> <li>• Varios métodos</li> <li>• Colores</li> <li>• Escribir método Categorías</li> <li>• Opción colores</li> </ul>	CATEGORIAS COLORES
<b>Colores y frases</b>	<ul style="list-style-type: none"> <li>• Caso Base</li> <li>• Escritura Textual</li> <li>• Funcionalidad Extra</li> <li>• Un método</li> <li>• Colores</li> <li>• Soporte frases</li> <li>• Opción completa</li> </ul>	MENU_COMPLETO
<b>Letras agrupadas y vocales</b>	<ul style="list-style-type: none"> <li>• Caso base</li> <li>• Escritura Textual</li> <li>• Varios métodos</li> <li>• Listas básicas</li> <li>• Funcionalidad simple</li> <li>• Escribir método vocales</li> </ul>	VOCALES

Tabla 8.4: Productos, paquetes y constantes definidas

Tenemos que señalar una seria dificultad con la que nos encontramos mientras implementábamos las aplicaciones de ejemplo:

A la hora de probar cualquier aplicación que contuviera el método de escritura de imágenes, el emulador integrado en el Visual Studio daba un error de “Dispositivo no preparado”. En primer lugar pensamos que era un error del emulador, por lo que intentamos crear un archivo CAB de la aplicación en cuestión para probarlo directamente en la PDA. A la hora de construir el .cab, aparecía el siguiente error: “*.inf contains DirIDs, which are not supported*”.

El error parecía proveer del CabWiz (que es invocado internamente por Visual Studio a la hora de crear proyectos CAB para dispositivos móviles o al pasar un proyecto al emulador). Después de algunas pruebas, descubrimos que el problema radicaba en la cantidad de archivos distintos que teníamos que añadir al proyecto. Con 262 archivos, Visual Studio podía crear un archivo .cab satisfactoriamente, pero con 263 archivos daba el error antes mencionado. Examinando el fichero .inf que Visual Studio pasa al CabWiz, descubrimos que tenía en la sección [DestinationDirs] una entrada por cada archivo añadido, incluso si el directorio era siempre el mismo para varios archivos.

Nos dimos cuenta de que este error era muy común entre los desarrolladores de software para dispositivos móviles. Afortunadamente, Windows Mobile 6 SDK incorpora una nueva versión del CabWiz que soluciona esta limitación, por lo que sólo tuvimos que descargarnos el SDK y reemplazar los archivos correspondientes con los del SDK. Los archivos que hubo que sustituir fueron “CabWiz.exe” y “CabWiz.ddf”, que se encuentran en la carpeta C:\Archivos de Programa\Microsoft Visual Studio 8.0\SmartDevices\SDK\SDKTools\.

Después de realizar estos cambios, pudimos crear y probar los proyectos realizados de manera satisfactoria.

## 8.4 Software empleado

Para la elaboración de este proyecto se han empleado los siguientes recursos software:

### Entorno de desarrollo

- Microsoft Windows XP Service Pack 2
- Microsoft Visual Studio.NET 2005
- .NET Compact Framework 2.0 (SP 2)
- Eclipse v.3.2.0
- Eclipse Modelling Framework v2.2.0
- Java Runtime Environment v1.4.2

### Conexión con Dispositivo móvil

- Microsoft ActiveSync 4.5

### Modelado

- StarUML v 5.0.2.1570

### **Realización de la memoria**

- Microsoft Office Word 2003
- Adobe Acrobat 8 Professional
- Adobe Illustrator 13.0.0

## **8.5 Hardware empleado**

Para la elaboración de este proyecto se han empleado los siguientes recursos hardware:

### **Equipos de desarrollo**

- Ordenador portátil Packard Bell EasyNote M258
  - Procesador Intel Core Duo 2 GHz.
  - 1024 MB de RAM.
  - 60 GB HDD.
- Ordenador de sobremesa
  - Procesador Intel P4 2.4 GHz.
  - 512 MB de RAM.
  - 80 GB HDD.
- Ordenador de sobremesa
  - Procesador Intel P4 1.5 GHz.
  - 640 MB de RAM.
  - 140 GB HDD.

### **Dispositivo móvil**

- PDA ACER N30
- Procesador Samsung s3c2410 266 mhz
- Memoria Rom 32 mb flash
- Memoria ram 64 mb sdram
- Microsoft Windows Mobile 2003 premium



## Capítulo 9

# CONFIGURACIÓN DE PRODUCTOS

### 9.1 Introducción

En el desarrollo de aplicaciones mediante una línea de productos, un miembro de dicha línea es construido mediante un conjunto de elementos reutilizables. El modelado de características permite modelar la variabilidad de los miembros de la línea de productos a través de todas las fases del desarrollo de la misma.

A la hora de la creación de aplicaciones determinadas, el diagrama de características sirve como base para estimar el coste del desarrollo, así como para derivar dichas aplicaciones, ya sea de forma manual o automatizada.

Para desarrollar un determinado miembro de la línea de productos, se necesita seleccionar previamente las funcionalidades que dicho miembro tendrá, es decir, configurar sus características. En líneas de productos con un número elevado de las mismas, no es algo sencillo debido a la posible existencia de restricciones y relaciones entre características.

Con el objetivo de facilitar dicha navegación por el diagrama de características y la configuración de las mismas, se están desarrollando algunas herramientas como el “*Kumbang Configurator*”<sup>5</sup> o el “*DOPLERUCon*”<sup>6</sup>. Actualmente está disponible de forma gratuita un plug-in para la plataforma de desarrollo Eclipse llamado “Feature Modeling Plug-in (fmp)”, que es la herramienta que se utilizará en este proyecto para la configuración de características. Dicho plugin ha sido desarrollado por Michał Antkiewicz y Krzysztof Czarnecki, de la Universidad de Waterloo, en Canadá.

---

<sup>5</sup> *Kumbang Configurator—A Configuration Tool for Software Product Families.*

Varvana Myllärniemi and Timo Asikainen and Tomi Mannisto and Timo Soininen. Helsinki University of Technology

<sup>6</sup> *Product Configuration in Large-Scale Product Lines.*

Mag. Rick Rabiser. Johannes Kepler University Linz

---

## 9.2 Modelado de características con Eclipse y el plugin FMP

El diagrama de características puede representarse en forma de árbol. Como se puede apreciar en la figura 9.1, en este caso, el modelo de características (  ) contiene un diagrama de características indicado por el símbolo  .

Las características comunes u obligatorias del diagrama, están señaladas con el símbolo  , que indica que su cardinalidad es [1..1]. La cardinalidad de una característica indica cuántas veces dicha característica (y cualquier posible subárbol) puede y debe estar presente en una misma aplicación. Todas las características comunes u obligatorias que cuelgan de otra característica opcional, sólo estarán presentes en un determinado miembro de la línea de producto, si la característica “padre” de la que cuelgan, es seleccionada para dicho miembro. Si no colgaran de ninguna otra característica, como sucede en la figura 3.1 con las cinco primeras características del árbol, estarán siempre presentes en cualquier aplicación de la línea y nunca estarán repetidas.

El símbolo de grupo  , indica una cardinalidad (1-1), es decir, de todas sus características “hijas” marcadas con  , se deberá escoger una y solamente una de ellas para estar presente en la línea de productos. De este modo podemos ver en la figura 9.1, que para la línea de productos se puede escoger entre *funcionalidad simple* o *funcionalidad extra*, y entre *un método de escritura* y *varios métodos de escritura*.

En el caso de la característica *varios métodos de escritura*, encontramos dentro de ella el símbolo de grupo  , que indica una cardinalidad (1-k), donde k es el tamaño del grupo. En este caso, k tiene valor 4, por lo que se pueden seleccionar de una a cuatro de sus características “hijas” para que estén presentes en un determinado miembro de la línea de productos.

Por último, mencionar que aunque en la figura 9.1 no se ha dado el caso, queda otro tipo de característica a representar, que es la característica opcional simple. Su cardinalidad es [0..1] y se representaría con el símbolo  .

Una vez que se representa el modelo de características de la línea de productos en la plataforma Eclipse, con el FMP, podemos pasar a la fase de creación de configuraciones de productos. A continuación se expone una imagen del árbol de características que crea el fmp.



Figura 9.1: Árbol del modelo de características



Figura 9.2: Ejemplo de configuración de producto.

### 9.3 Configuración de productos basada en características

La configuración es el proceso de derivar un subconjunto de características concreto, de acuerdo al diagrama de las mismas, mediante un proceso de selección. La figura 9.2 muestra una configuración del diagrama de características de la figura 9.1. Cajas de selección aparecen al lado de características opcionales y grupos de características, permitiendo su selección o descarte.

De esta forma, vamos seleccionando o descartando aquellas características que queremos para nuestra aplicación y vamos restringiendo las posibilidades hasta quedarnos con una configuración de entre todas las posibles. En la línea de productos que trata este proyecto, tenemos inicialmente un conjunto de 64 configuraciones disponibles, es decir, podemos crear 64 aplicaciones similares, pero distintas, combinando las características entre sí. Según se van seleccionando características, reducimos el número de configuraciones disponibles. Por ejemplo, al seleccionar entre dos alternativas, el número de aplicaciones que podremos crear ahora con esa alternativa seleccionada será de 32. El número de configuraciones restantes aparece siempre señalado a lado del nombre del árbol de características.

Una vez que tenemos lista la configuración deseada, podemos exportar a un fichero .XML las características escogidas (Figura 9.3). A la hora de crear una aplicación completa de la línea de productos de este proyecto, se debe editar el fichero .csproj del proyecto para añadir los paquetes correspondientes a cada característica que queramos incluir en nuestro producto. Como futura línea de trabajo sobre este proyecto, se propondrá la utilización del fichero .XML que resulta de crear la configuración con Eclipse y FMP para poder añadir los paquetes de manera automática, sin tener que modificar el fichero .csproj.



```
<feature name="Comunicador" type="NONE" id="comunicador">
  <feature name="Consultar Ayuda" type="NONE" id="consultarAyuda">
  </feature>
</feature>
<feature name="Configurar Barrido" type="NONE" id="configurarBarrido">
</feature>
<feature name="Borrado de ultima palabra" type="NONE" id="borradodeúltimapalabra">
</feature>
<feature name="Escribir por letras agrupadas" type="NONE" id="escribirporletrasagrupadas">
</feature>
<feature name="Reproducir" type="NONE" id="reproducir">
</feature>
<feature name="Funcionalidad Extra" type="NONE" id="funcionalidadExtra">
  <feature name="Añadir Opcion" type="NONE" id="añadirOpcion">
  </feature>
  <feature name="BotonesMenu" type="NONE" id="botonesMenu">
  </feature>
  <feature name="Opcion Frases" type="NONE" id="opcionFrases">
    <feature name="Apariencia Frases" type="NONE" id="aparienciaFrases">
    </feature>
    <feature name="Soporte de Frases" type="NONE" id="soportedeFrases">
      <feature name="Borrar Frase" type="NONE" id="borrarFrases">
      </feature>
      <feature name="Insertar Frase" type="NONE" id="insertarFrase">
      </feature>
      <feature name="Guardar Frase" type="NONE" id="guardarFrase">
      </feature>
    </feature>
  </feature>
</feature>
</feature>
```

Figura 9.3: Ejemplo de .xml generado mediante la configuración de un producto.

## Capítulo 10

### PRUEBAS

#### 10.1 Introducción

En este apartado se presentarán las pruebas realizadas a las distintas aplicaciones obtenidas en la línea de productos desarrollada. La finalidad de las pruebas es detectar los posibles errores para su corrección.

Para la realización de las pruebas se ha utilizado, tanto el emulador de Pocket PC que ofrece el entorno de desarrollo de .NET, como una PDA cedida por el Grupo GIRO para la realización de este proyecto. Las especificaciones técnicas de la misma pueden verse en el punto 8.5 de la memoria: “Hardware empleado”.

Debido a la especial naturaleza del presente proyecto de fin de carrera, para probar la correcta integración de los distintos elementos de los que puede disponer una aplicación, se han instanciado varias aplicaciones a modo de ejemplo. La información sobre dichas aplicaciones puede verse en el punto 8.3 de la memoria: “Instanciación manual de productos”.

Sobre dichas aplicaciones de ejemplo, se han realizado todas y cada una de las pruebas que se presentan a continuación, señalando, en caso de error, la aplicación donde se dio el mismo.

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 el usuario (escribir, borrar y reproducir).
- De casos límite: Pruebas encaminadas a encontrar errores que el desarrollador sospecha que pueden existir en un determinado caso.

## 10.2 Pruebas realizadas

### PRUEBA 001: Escritura usando botón con funcionalidad de ESCRIBIR (a)

<i>Acción realizada</i>	Selección de botón de escritura de manera directa estando el barrido desactivado.
<i>Resultado esperado</i>	El texto asociado al botón se añadirá al cuadro de texto.
<i>Resultado obtenido</i>	Texto añadido.
<i>Observaciones</i>	Correcto.

Tabla 10.1: Prueba de escritura (a)

### PRUEBA 002: Escritura usando botón con funcionalidad de ESCRIBIR (b)

<i>Acción realizada</i>	Selección de botón de escritura de manera directa estando el barrido activado.
<i>Resultado esperado</i>	El texto asociado al botón se añadirá al cuadro de texto y el barrido seguirá su camino.
<i>Resultado obtenido</i>	Texto añadido y el barrido sigue su camino.
<i>Observaciones</i>	Correcto.

Tabla 10.2: Prueba de escritura (b)

### PRUEBA 003: Escritura usando botón con funcionalidad de ESCRIBIR (c)

<i>Acción realizada</i>	Selección del botón de escritura señalado por el barrido.
<i>Resultado esperado</i>	El texto asociado al botón se añadirá al cuadro de texto y el barrido seguirá su camino.
<i>Resultado obtenido</i>	Texto añadido y el barrido sigue su camino.
<i>Observaciones</i>	Correcto.

Tabla 10.3: Prueba de escritura (c)

---

**PRUEBA 004: Escritura usando botón de caracteres con funcionalidad de ESCRIBIR Y VOLVER (a)**

<i>Acción realizada</i>	Selección de botón de escritura de manera directa estando el barrido desactivado.
<i>Resultado esperado</i>	El texto asociado al botón se añadirá al cuadro de texto y se mostrará la página anterior a la actual.
<i>Resultado obtenido</i>	Texto añadido y mostrado de la anterior página.
<i>Observaciones</i>	Correcto.

Tabla 10.4: Prueba escribir y volver (a)

**PRUEBA 005: Escritura usando botón de caracteres con funcionalidad de ESCRIBIR Y VOLVER (b)**

<i>Acción realizada</i>	Selección de botón de escritura de manera directa estando el barrido activado.
<i>Resultado esperado</i>	El texto asociado al botón se añadirá al cuadro de texto y se mostrará la página anterior a la actual.
<i>Resultado obtenido</i>	Texto añadido y mostrado de la anterior página.
<i>Observaciones</i>	Correcto.

Tabla 10.5: Prueba escribir y volver (b)

**PRUEBA 006: Escritura usando botón de caracteres con funcionalidad de ESCRIBIR Y VOLVER (c)**

<i>Acción realizada</i>	Selección del botón de escritura señalado por el barrido.
<i>Resultado esperado</i>	El texto asociado al botón se añadirá al cuadro de texto y se mostrará la página anterior a la actual.
<i>Resultado obtenido</i>	Texto añadido y mostrado de la anterior página.
<i>Observaciones</i>	Correcto.

Tabla 10.6: Prueba escribir y volver (c)

### PRUEBA 007: Borrado con funcionalidad de BORRAR

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

Tabla 10.7: Prueba borrar carácter

### PRUEBA 008: Borrado con funcionalidad de BORRAR PALABRA

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

Tabla 10.8: Prueba borrar palabra

### PRUEBA 009: Insertar un espacio en el texto

<i>Acción realizada</i>	Pulsación del botón ESPACIO.
<i>Resultado esperado</i>	Se introducirá un espacio en el texto.
<i>Resultado obtenido</i>	Espacio introducido.
<i>Observaciones</i>	Correcto.

Tabla 10.9: Prueba insertar espacio en texto

### PRUEBA 010: Mostrado de la pantalla de MENU

<i>Acción realizada</i>	Selección del botón de la pantalla destinado a mostrar la pantalla de menú.
<i>Resultado esperado</i>	Se dejará de mostrar la pantalla actual y se mostrará la pantalla de menú.
<i>Resultado obtenido</i>	Pantalla de menú mostrada correctamente, pero el barrido no recorre en orden los botones en la aplicación “Colores y frases”.
<i>Observaciones</i>	Incorrecto en “Colores y frases”.
<i>Modo de actuación</i>	Modificado y corregido en la versión final del proyecto.

Tabla 10.10: Prueba mostrar menú

### PRUEBA 011: Salir de la aplicación

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

Tabla 10.11: Prueba salir de la aplicación

### PRUEBA 012: Volver a la pantalla anterior

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

Tabla 10.12: Prueba volver a la pantalla anterior

### PRUEBA 013: Ver temas de ayuda

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

Tabla 10.13: Prueba ver temas de ayuda

### PRUEBA 014: Consultar ayuda sobre un tema

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

Tabla 10.14: Prueba consultar un tema de ayuda

### PRUEBA 015: Reproducir texto escrito

<b><i>Acción realizada</i></b>	Selección del botón destinado a la reproducción de manera oral del texto.
<b><i>Resultado esperado</i></b>	Se deberá reproducir el texto escrito.
<b><i>Resultado obtenido</i></b>	Texto reproducido oralmente.
<b><i>Observaciones</i></b>	Correcto.

Tabla 10.15: Prueba reproducir texto

### PRUEBA 016: Desactivar barrido

<b><i>Acción realizada</i></b>	Selección de la opción “NO” (barrido no activado) en la pantalla configuración y selección del botón “ACEPTAR”.
<b><i>Resultado esperado</i></b>	Mostrado de mensaje advirtiendo de la necesidad de reiniciar. Al reiniciar el usuario la aplicación, el barrido estará desactivado.
<b><i>Resultado obtenido</i></b>	Mostrado de mensaje y barrido activado al reiniciar.
<b><i>Observaciones</i></b>	Correcto.

Tabla 10.16: Prueba desactivar barrido

### PRUEBA 017: Cambiar tiempo de barrido

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

Tabla 10.17: Prueba cambiar tiempo de barrido

### PRUEBA 018: Cambiar la configuración

<i>Acción realizada</i>	Selección de una de las configuraciones que se muestran en la pantalla configuración.
<i>Resultado esperado</i>	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.
<i>Resultado obtenido</i>	Mostrado de mensaje y configuración actualizada.
<i>Observaciones</i>	Correcto.

Tabla 10.18: Prueba cambiar configuración

### PRUEBA 019: Mostrar la pantalla de FRASES

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

Tabla 10.19: Prueba mostrar la pantalla de frases

### PRUEBA 020: Añadir una frase al cuadro de texto

<i>Acción realizada</i>	Selección del botón AÑADIR de la pantalla de frases.
<i>Resultado esperado</i>	Se insertará la frase seleccionada en el texto introducido.
<i>Resultado obtenido</i>	Frase añadida en el texto.
<i>Observaciones</i>	Correcto.

Tabla 10.20: Prueba añadir una frase al texto

### PRUEBA 021: Borrar una frase de la pantalla de frases

<b><i>Acción realizada</i></b>	Selección del botón BORRAR de la pantalla de frases después de haber elegido una frase de la lista de frases.
<b><i>Resultado esperado</i></b>	La frase seleccionada se borrará de la lista.
<b><i>Resultado obtenido</i></b>	Frase borrada.
<b><i>Observaciones</i></b>	Correcto.

Tabla 10.21: Prueba borrar una frase de la pantalla de frases

### PRUEBA 022: Guardar un texto en el listado de frases

<b><i>Acción realizada</i></b>	Selección del botón GUARDAR de la pantalla de frases.
<b><i>Resultado esperado</i></b>	Guarda el texto escrito como una nueva frases en la lista de frases.
<b><i>Resultado obtenido</i></b>	Frase guardada.
<b><i>Observaciones</i></b>	Correcto.

Tabla 10.22: Prueba guardar un texto en el listado de frases

### PRUEBA 023: Desplazamiento de las frases en la lista de frases

<b><i>Acción realizada</i></b>	Pulsación de los botones arriba y abajo de la pantalla de frases.
<b><i>Resultado esperado</i></b>	Desplazamiento por las frases de la lista.
<b><i>Resultado obtenido</i></b>	No hay desplazamiento alguno en aplicaciones “Ejemplo Frases” y “Colores y frases”.
<b><i>Observaciones</i></b>	Incorrecto en “Ejemplo Frases” y “Colores y frases”.
<b><i>Modo de actuación</i></b>	Modificado y corregido en la versión final del proyecto.

Tabla 10.23: Prueba desplazamiento de las frases en la lista de frases

### PRUEBA 024: Mostrar la pantalla colores

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

Tabla 10.24: Prueba mostrar la pantalla colores

### **PRUEBA 025: Mostrar colores disponibles para cambiar el color de fondo**

<i>Acción realizada</i>	Pulsación del botón CAMBIAR junto al texto “Color fondos” de la pantalla colores.
<i>Resultado esperado</i>	Se mostrará una pantalla con la paleta de colores disponibles.
<i>Resultado obtenido</i>	Pantalla mostrada.
<i>Observaciones</i>	Correcto.

Tabla 10.25: Prueba mostrar colores disponibles para cambiar el color de fondo

### **PRUEBA 026: Mostrar colores disponibles para cambiar el color de los botones**

<i>Acción realizada</i>	Pulsación del botón CAMBIAR junto al texto “Color botones” de la pantalla colores.
<i>Resultado esperado</i>	Se mostrará una pantalla con la paleta de colores disponibles.
<i>Resultado obtenido</i>	Pantalla mostrada.
<i>Observaciones</i>	Correcto.

Tabla 10.26: Prueba mostrar colores disponibles para cambiar el color de los botones

### **PRUEBA 027: Mostrar colores disponibles para cambiar el color de texto de los botones**

<i>Acción realizada</i>	Pulsación del botón CAMBIAR junto al texto “Color texto botón” de la pantalla colores.
<i>Resultado esperado</i>	Se mostrará una pantalla con la paleta de colores disponibles.
<i>Resultado obtenido</i>	Pantalla mostrada.
<i>Observaciones</i>	Correcto.

Tabla 10.27: Prueba mostrar colores disponibles para cambiar el color de texto de los botones

### PRUEBA 028: Mostrar colores disponibles para cambiar el color del texto

<i>Acción realizada</i>	Pulsación del botón CAMBIAR junto al texto “Color texto” de la pantalla colores.
<i>Resultado esperado</i>	Se mostrará una pantalla con la paleta de colores disponibles.
<i>Resultado obtenido</i>	Pantalla mostrada.
<i>Observaciones</i>	Correcto.

Tabla 10.28: Prueba mostrar colores disponibles para cambiar el color de texto

### PRUEBA 029: Cambiar el color

<i>Acción realizada</i>	Pulsación del color deseado en la pantalla de colores disponibles.
<i>Resultado esperado</i>	Cambiar el color del elemento elegido.
<i>Resultado obtenido</i>	Cambio del color del elemento elegido.
<i>Observaciones</i>	Correcto.

Tabla 10.29: Prueba cambiar color

## 10.3 Análisis de casos límite

### PRUEBA 030: Borrado cuando el cuadro de texto está vacío

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

Tabla 10.30: Prueba borrar sin texto

### PRUEBA 031: Borrado de palabra cuando el cuadro de texto está vacío

<i>Acción realizada</i>	Selección del botón de borrado de palabra cuando el cuadro de texto no contiene ningún texto.
<i>Resultado esperado</i>	No ocurrirá nada.
<i>Resultado obtenido</i>	Error. Se termina la aplicación en todas las aplicaciones.
<i>Observaciones</i>	Incorrecto en todas las aplicaciones.
<i>Modo de actuación</i>	Modificado y corregido en la versión final del proyecto.

Tabla 10.31: Prueba borrar una palabra sin texto

### PRUEBA 032: Reproducción cuando el cuadro de texto está vacío

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

Tabla 10.32: Prueba reproducir sin texto

### PRUEBA 033: Reproducción cuando el texto no es reproducible

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

Tabla 10.33: Prueba reproducir texto no reproducible

### **PRUEBA 034: Cambio de las imágenes de botones variables**

<b><i>Acción realizada</i></b>	Esperar a que los botones de imágenes variables de una pantalla cambien sus imágenes cuando el barrido termine de recorrerlos.
<b><i>Resultado esperado</i></b>	Los botones cambiarán sus imágenes por las siguientes en la lista de imágenes de la pantalla.
<b><i>Resultado obtenido</i></b>	Los botones han cambiado de imágenes.
<b><i>Observaciones</i></b>	Correcto.

Tabla 10.34: Prueba cambio de imágenes de botones variables

### **PRUEBA 035: Desplazamiento hacia arriba con la primera frase marcada**

<b><i>Acción realizada</i></b>	Pulsación de la flecha subir cuando estemos situados en la primera frase de la lista.
<b><i>Resultado esperado</i></b>	Situación en la última frase de la lista.
<b><i>Resultado obtenido</i></b>	Última frase de la lista marcada.
<b><i>Observaciones</i></b>	Correcto.

Tabla 10.35: Prueba desplazamiento hacia arriba con la primera frase marcada

### **PRUEBA 036: Desplazamiento hacia abajo con la última frase marcada**

<b><i>Acción realizada</i></b>	Pulsación de la flecha bajar cuando estemos situados en la última frase de la lista.
<b><i>Resultado esperado</i></b>	Situación en la primera frase de la lista.
<b><i>Resultado obtenido</i></b>	Primera frase de la lista marcada.
<b><i>Observaciones</i></b>	Correcto.

Tabla 10.36: Prueba desplazamiento hacia abajo con la última frase marcada

# **PARTE 5. MANUAL DE USUARIO**

---



## Capítulo 11

# MANUAL DE USUARIO

### 11.1 Introducción

En este capítulo se muestra el manual de usuario de las aplicaciones que se pueden generar mediante la línea de productos desarrollada.

En el presente manual se explicará, paso a paso, la correcta instalación y uso de cualquier aplicación generada. Se mostrarán, como apoyo, las pantallas que mostrará el comunicador, en los distintos pasos a realizar. Se dará una clara explicación de los elementos que conforman cada pantalla, explicando su función y apariencia.

Con la línea de productos generada, se pueden desarrollar un elevado número de aplicaciones distintas, por lo que el manual de usuario contemplará dicha variabilidad. En caso de que una pantalla pueda ser diferente dependiendo de las características de la aplicación seleccionadas, se mostrarán todas las posibles opciones explicando detalladamente sus elementos.

A continuación se muestra un breve resumen de los contenidos del manual:

- Instalación de la aplicación.
- Pantalla de Bienvenida.
- Funcionamiento del barrido.
- Escritura mediante inserción de caracteres.
  - Método de letras agrupadas.
  - Método de categorías.
  - Método de las vocales.
- Borrar.
- Escritura mediante imágenes.
- Borrar una palabra por el método de imágenes.
- Movimientos entre pantallas.
- Menú.
- Reproducción por voz.
- Configuración de la aplicación.
- Cambiar colores de los elementos de la aplicación.
- Empleo de frases.
- Uso de la ayuda en la aplicación.
- Salir de la aplicación.

## 11.2 Instalación de la aplicación

Después de haber desarrollado una aplicación mediante la línea de productos, se puede proceder a su instalación en un dispositivo móvil. Recordemos que para el desarrollo, se ha configurado dicha aplicación con la herramienta Eclipse y el FMP, se han añadido los paquetes y constantes necesarias para su correcto funcionamiento, y se ha construido (“build”) mediante Microsoft Visual Studio.NET.

En primer lugar, crearemos un archivo autoinstalable de la aplicación. Para ello, se debe crear un nuevo “Smart Device CAB Project” en MS. Visual Studio .NET. En su creación, hay que marcar la opción de “añadir a la solución” de la línea de productos, que es de donde obtendremos el contenido del .cab.

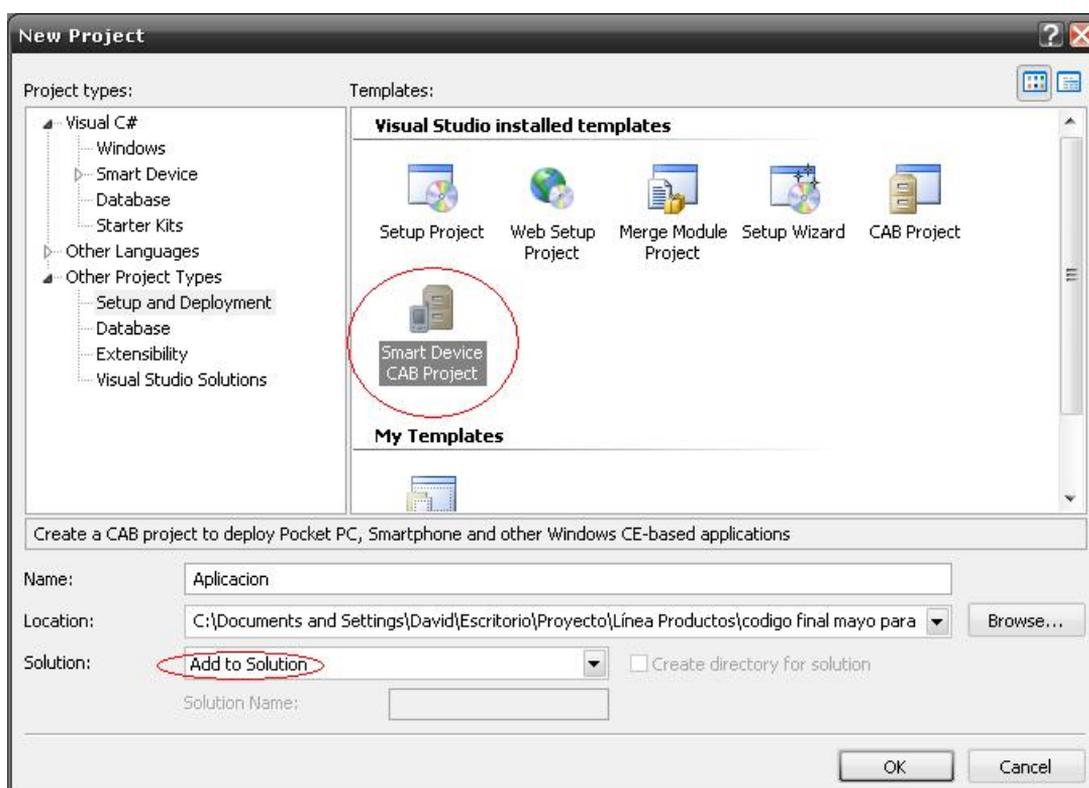


Figura 11.1. Creación de proyecto CAB

Una vez creado, se mostrará el sistema de ficheros del proyecto CAB. Para que las aplicaciones de la línea de productos funcionen adecuadamente, deberemos añadir los archivos de las mismas en una carpeta que cuelgue del directorio “Program Files” del dispositivo móvil. Para ello, hacemos click con el botón secundario en la carpeta “Application Folder”, y seleccionamos “Add” → “Project Output”.

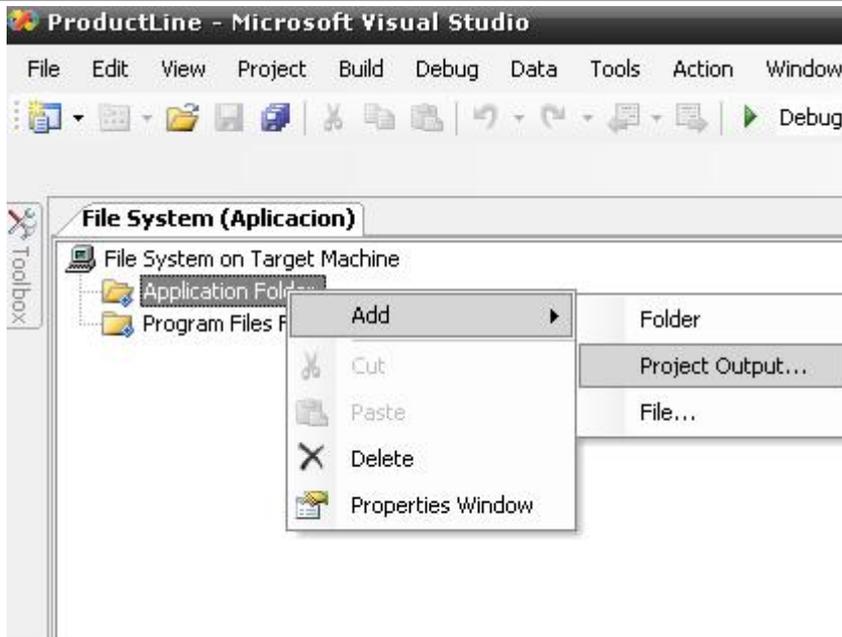


Figura 11.2. Adición de archivos al .cab

Aparecerá un menú donde se podrá escoger el proyecto que se desea incluir en el archivo .cab. Seleccionamos el proyecto de la aplicación que queremos instalar, y posteriormente, se marcan los elementos “Primary output”, “Localized resources” y “Content files”. A continuación, se pulsa el botón “OK”.

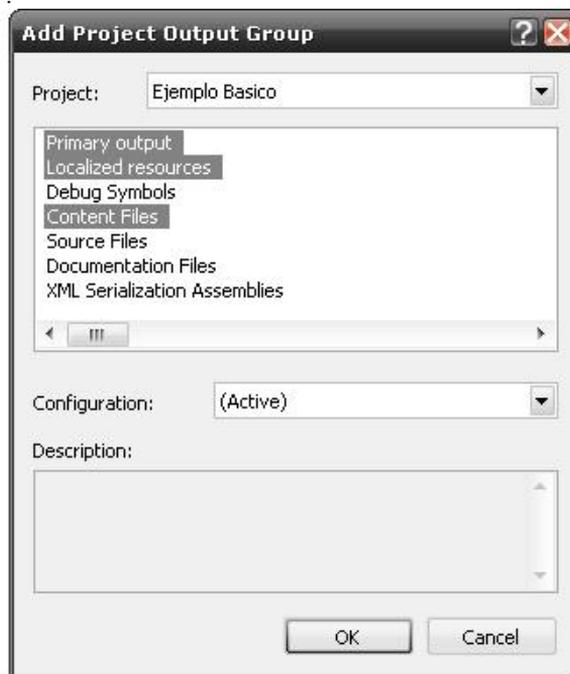


Figura 11.3. Selección de elementos a añadir.

Ya sólo queda construir el proyecto CAB. Para ello, primeramente deberemos escoger la opción “Release” en el desplegable “Solution Configurations” de la barra de herramientas estándar. Después hay que seleccionar el proyecto en el explorador de la solución, y con el botón secundario, pulsamos en “Build”.

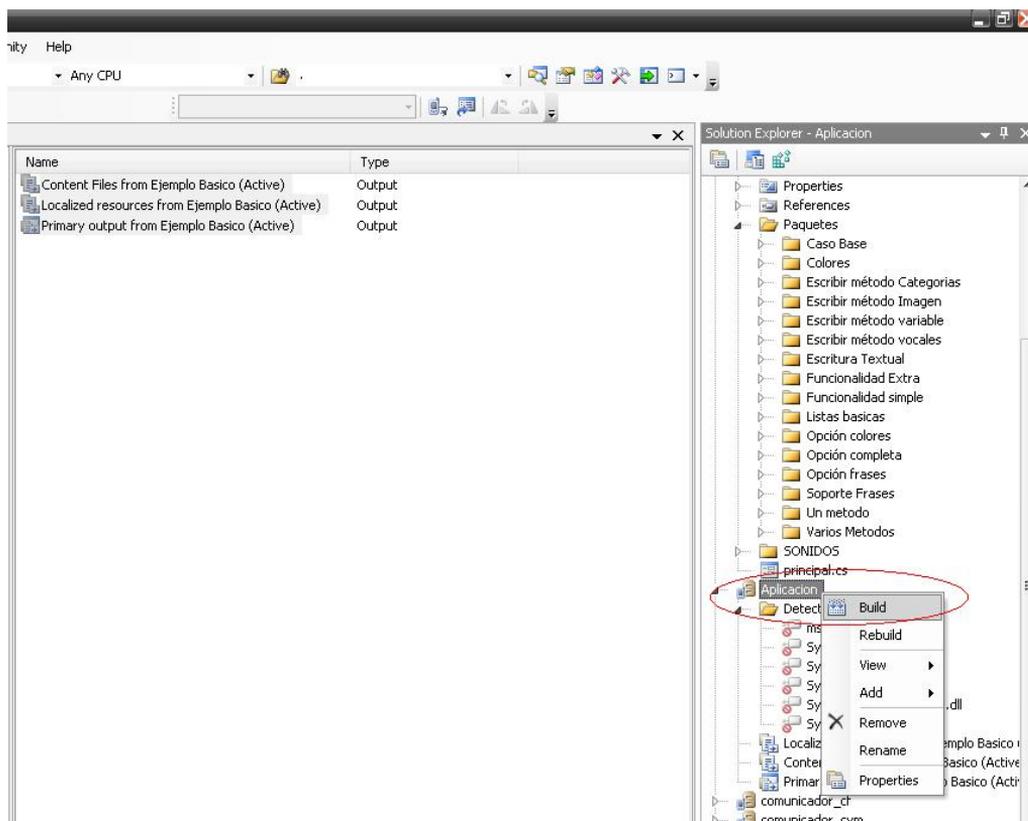


Figura 11.4. Construcción del proyecto CAB.

Por último, nos dirigiremos al directorio donde está el proyecto CAB, y en la carpeta “Release” podremos encontrar el archivo .cab de la aplicación.

Mediante Microsoft Active Sync, podemos copiar dicho archivo .cab en nuestro dispositivo móvil, y al ejecutarlo, ya se dispondrá de la aplicación correctamente instalada.

### 11.3 Pantalla de bienvenida



Figura 11.5: Pantalla de bienvenida

Al iniciarse la aplicación aparecerá esta pantalla de bienvenida (Figura 11.5). Contiene el título del proyecto y el nombre de los autores 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.

### 11.4 Funcionamiento del barrido

Toda aplicación generada admite la posibilidad de escritura con barrido. El barrido es un método de ayuda a la escritura que consiste en señalar (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 se seleccione el botón señalado por el barrido, cuando se accione con cualquier parte del cuerpo, un pulsador conectado a la PDA.

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, el botón señalado por el barrido será de color negro y con el texto blanco (Figura 11.6).
- Si la configuración es de teclado formado por botones de imágenes habrá dos opciones:
  - El botón es de selección de categorías de imágenes: Será del color representativo de la categoría, pero el borde del botón será de color morado. (Figura 11.7).
  - 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 11.8).



Figura 11.6: Pantalla del método de letras agrupadas con un botón señalado por el barrido

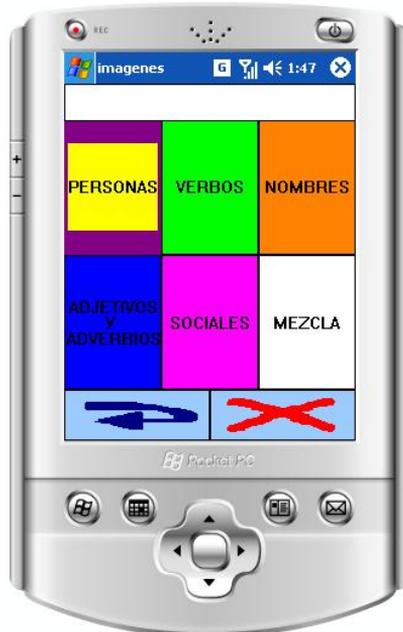


Figura 11.7: Pantalla del método de imágenes con un botón señalado por el barrido



Figura 11.8: Pantalla del método de imágenes con un botón que no es de selección de categorías señalado por el barrido.

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.

## 11.5 Escritura mediante inserción de caracteres

Se dispone de tres metodologías diferentes: método de letras agrupadas, método de categorías y método de las vocales. A continuación describimos su funcionamiento.

### 11.5.1 Método de letras agrupadas

La pantalla de la figura 11.9 será la pantalla principal cuando se esté ejecutando la aplicación con la configuración de letras agrupadas.

En la parte superior de la pantalla se muestra un cuadro de texto (vacío de un primer 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”, “B. PALABRA”, “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:



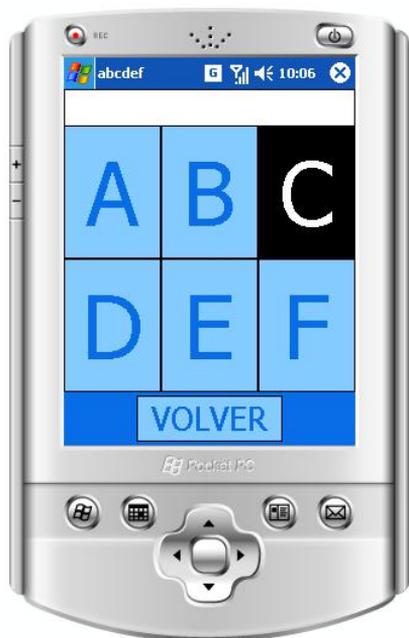
Figura 11.9: Pantalla principal de letras agrupadas.

- BORRAR: Borra el último carácter escrito.
- B. PALABRA: Borra la última palabra escrita.
- 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 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 11.1:** Supóngase que se desea añadir la letra “C” al cuadro de texto. Para ello se deberá pulsar el botón representado por los caracteres “ABCDEF” (Figura 11.9). A continuación se mostrará la pantalla de la Figura a) del ejemplo. Después, se deberá pulsar el botón representado por el carácter “C”. La aplicación mostrará la pantalla inicial, pero con la letra “C” escrita en el cuadro de texto (Figura b) del ejemplo).



a) Pantalla ABCDEF



b) Pantalla inicial con texto escrito

Ejemplo 11.1: Escritura de un carácter mediante el método de letras agrupadas.

- **Botón Números:**

Si se selecciona el botón de la figura 11.9 que contiene el grupo de caracteres “0-9”, se mostraría la pantalla de la figura 11.10, en la que aparecen los números del 0 al 9, el punto decimal y varios operadores matemáticos. De igual forma que en el caso anterior, al pulsar uno de los botones se añadiría al texto escrito el carácter asociado.

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; para acceder a la pantalla principal se debe pulsar el botón “VOLVER”.

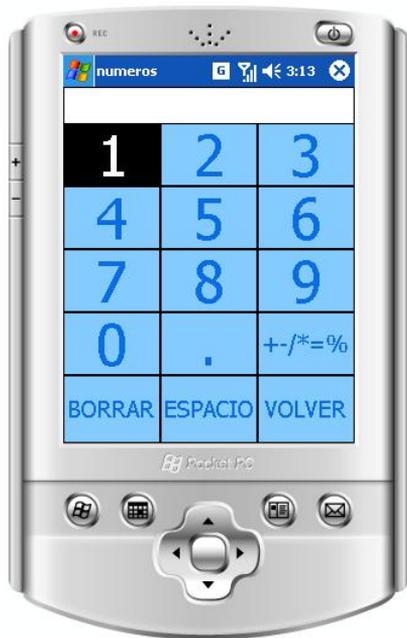


Figura 11.10: Pantalla desplegada del botón 0-9

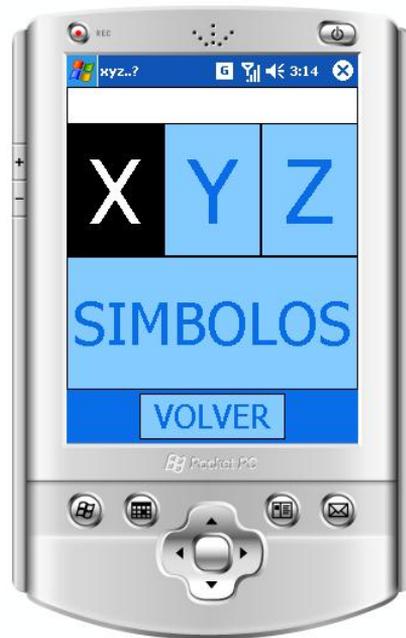


Figura 11.11: Pantalla del botón XYZ..?

- **Botón Símbolos:**

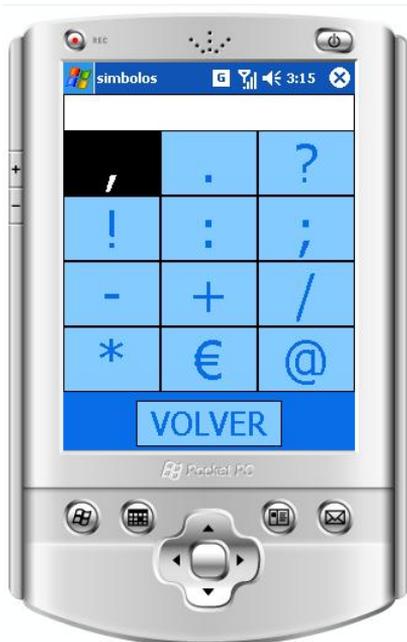


Figura 11.12: Pantalla del botón SIMBOLOS.

Si se selecciona el botón de la figura 11.9 que contiene el grupo de caracteres “XYZ..?”, se mostraría la figura 11.11, que no sólo contiene las letras “X”, “Y” y “Z”, sino también un botón más llamado “SIMBOLOS”. Al presionarlo, nos mostrará la pantalla que se puede ver en la figura 11.12 que contendrá varios signos ortográficos y otros símbolos.

### 11.5.2 Método de categorías



Figura 11.13: Pantalla principal de CATEGORIAS.

La pantalla de la figura 11.13 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 en un primer 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”, “B. PALABRA”, “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.

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:

- Si la categoría elegida es la de “VOCALES” (Figura 11.14), 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 11.12).
- Si la categoría elegida es la de “CONSONANTES”, cada botón estará representado por un conjunto de consonantes (Figura 11.15). Al pulsar uno de estos botones se mostrará una nueva pantalla donde cada botón está representado por una única consonante como por ejemplo, la figura 11.16 que se obtiene tras pulsar el botón “BCDF”. En esta nueva pantalla ya se puede escribir una consonante mediante la selección del botón que la represente.

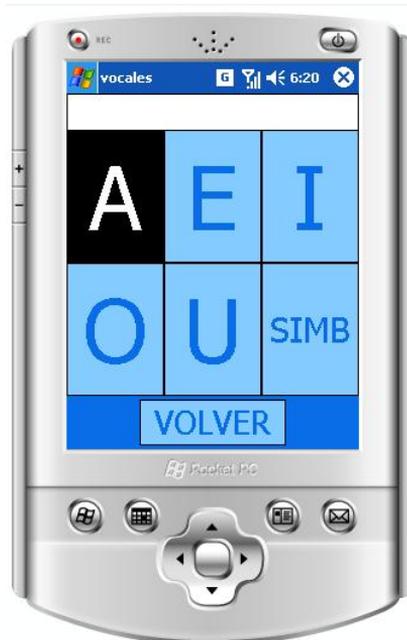


Figura 11.14: Pantalla VOCALES

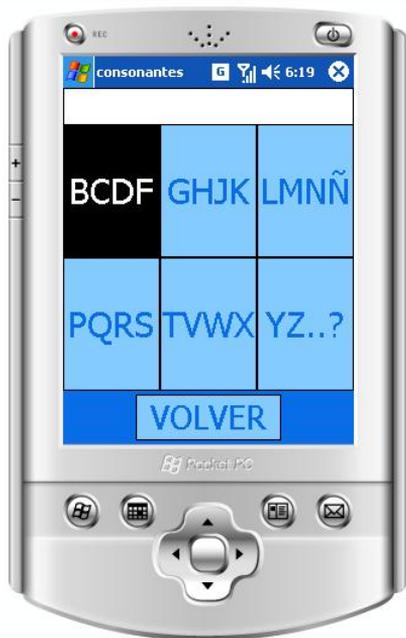


Figura 11.15: Pantalla CONSONANTES

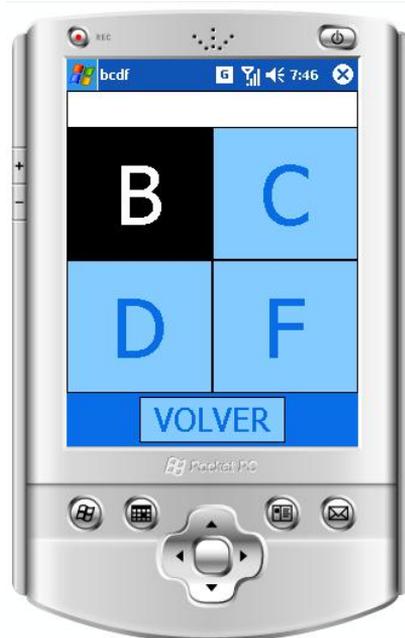


Figura 11.16: Pantalla BCDF

- Si la categoría elegida es la de NUMEROS, se mostrará una pantalla formada por botones que representarán los números del 0 al 9, el punto decimal y varios operadores matemáticos (Figura 11.10).

**Ejemplo 11.2:** Supóngase el mismo ejemplo que antes. Se desea añadir la letra “C” al cuadro de texto. Para ello se deberá pulsar el botón que representa la categoría “CONSONANTES” (Figura 11.13). A continuación se mostrará la pantalla de consonantes representada por la figura 11.15. Seleccionamos el botón que contiene la letra deseada, en nuestro caso la “C”, es decir, el botón “BCDF”. A continuación aparecerá la pantalla de la figura 11.16. Para finalizar, se pulsa el botón representado por el carácter “C”. La aplicación mostrará la pantalla inicial, pero con la letra “C” escrita en el cuadro de texto.

### 11.5.3 Método de las vocales



Figura 11.17: Pantalla principal del método de las VOCALES.

La figura 11.17 muestra la pantalla inicial del método de las vocales. Este método se puede considerar una optimización del de letras agrupadas.

Las vocales se localizan en botones independientes, de forma que pulsándolos se escribirá directamente la vocal deseada en el cuadro de texto. De esta forma, el número de pulsaciones disminuye considerablemente, puesto que en una palabra, por cada consonante hay, aproximadamente, una vocal.

Las consonantes se agrupan de la misma forma que en método de categorías. Es decir, pulsando un botón con un grupo de consonantes, nos llevará a una pantalla donde cada consonante aparecerá en un botón de manera independiente.

El botón representado por los caracteres “0-9” se corresponderá, como en los métodos de escritura anteriores, con la figura 11.10.

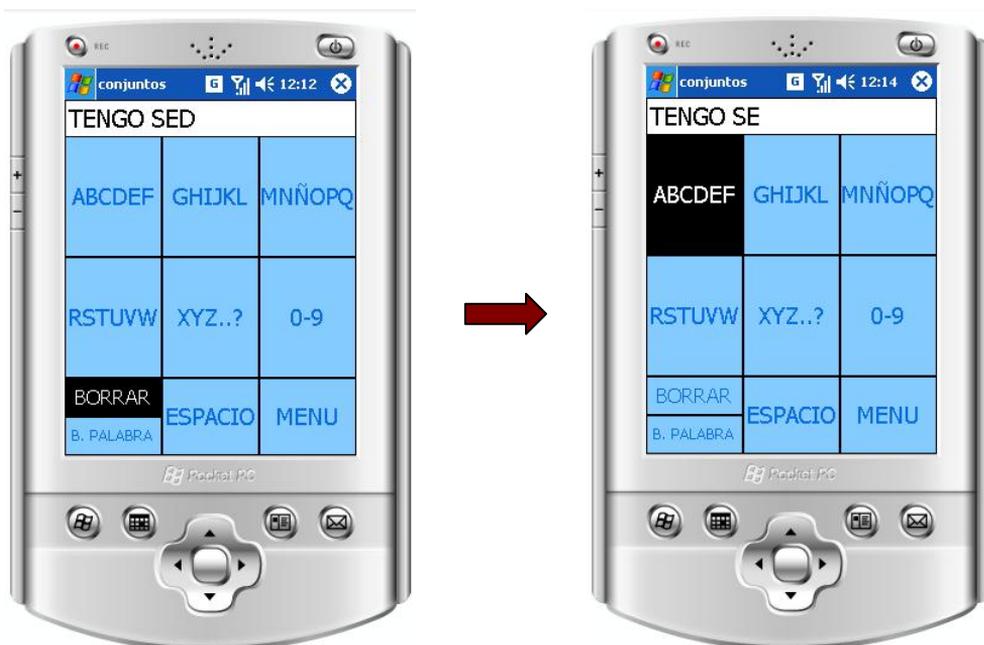
**Ejemplo 11.3:** Para escribir la sílaba “CA” habrá que hacer lo siguiente: Se seleccionará el botón “BCDF” de la figura 11.17, que nos mostrará una pantalla con cuatro botones con las letras “B”, “C”, “D” y “F” respectivamente (figura 11.16). Seleccionamos el botón con la letra “C”. Automáticamente volveremos a la pantalla principal (figura 11.17) pero con la letra “C” escrita en el cuadro de texto. A continuación pulsaremos el botón “A” agregándose directamente al cuadro de texto teniendo así formada la sílaba “CA”.

## 11.6 Borrar texto

Para borrar texto escrito mediante inserción de caracteres se dispone de dos formas posibles:

- **Borrar un único 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 letras agrupadas, categorías y vocales.

**Ejemplo 11.4:** 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.



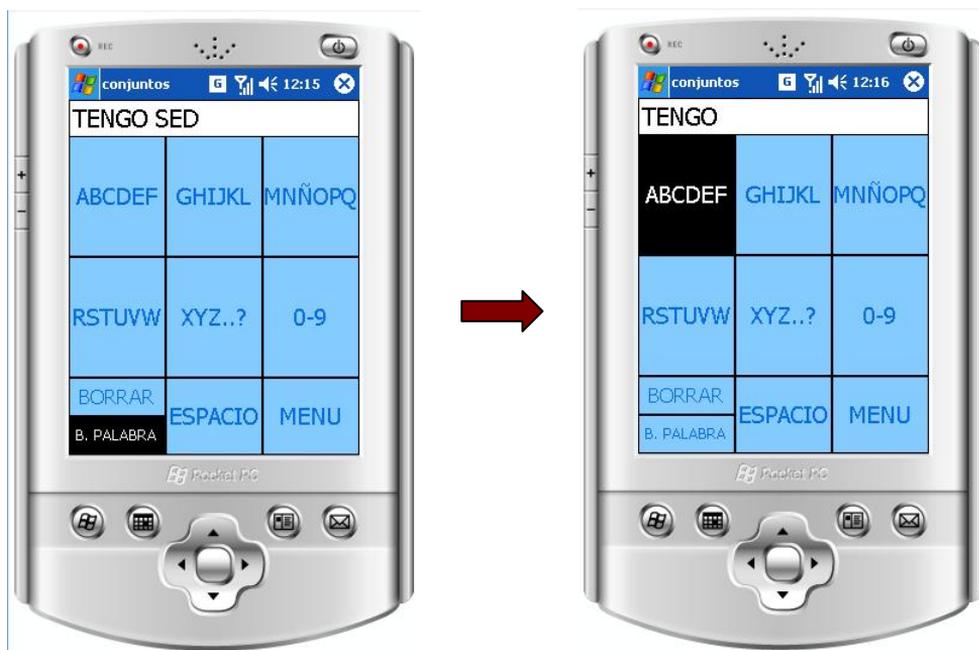
a) Pantalla principal de la configuración letras agrupadas con texto escrito.

b) Pantalla principal de la configuración letras agrupadas con el último carácter borrado.

Ejemplo 11.4: Borrado de un carácter

- **Borrar una palabra:** Para borrar un carácter simplemente se debe pulsar el botón con el texto asociado “B. PALABRA” que aparecerá en las pantallas principales para las configuraciones de letras agrupadas, categorías y vocales.

**Ejemplo 11.5:** Supóngase que se tiene un texto escrito en el cuadro de texto (Figura a) del ejemplo) y se desea borrar la última palabra. Para ello se debe pulsar el botón B. PALABRA.



a) Pantalla principal de la configuración letras agrupadas con texto escrito.

b) Pantalla principal de la configuración letras agrupadas con la última palabra borrada.

Ejemplo 11.5: Borrado de una palabra.

## 11.7 Escritura mediante imágenes



Figura 11.18: Pantalla inicial en la configuración de IMÁGENES

En la figura 11.18 se muestra la pantalla inicial cuando la aplicación se está ejecutando bajo la configuración de IMÁGENES.

La pantalla inicial en esta configuración, está formada, 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 11.19: Primera pantalla de inserción de expresiones de la categoría Personas

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 de una pantalla.

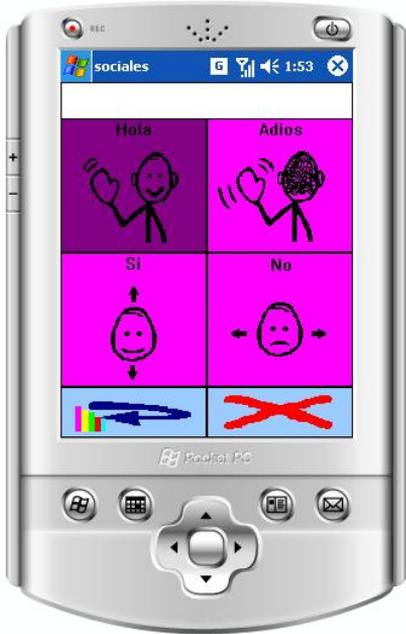
Para que la aplicación conozca el momento en el que debe cambiar las imágenes de los botones se utiliza el barrido. Así cuando el barrido está seleccionando el último botón de la pantalla (botón borrar palabra), la aplicación cambia las imágenes y las expresiones asociadas de los botones variables.

Por ejemplo, la categoría de “PERSONAS” dispone de varias pantallas, de entre ellas, tenemos como primera la figura 11.19, a la que le sigue la figura 11.20, que aparecerá una vez que pase el barrido por el aspa roja que representa el botón de borrar palabra.



Figura 11.20: Segunda pantalla de inserción de expresiones de la categoría Personas

**Ejemplo 11.6:** Supongamos que queremos escribir: “Hola abuela”. Para ello, estando en la pantalla principal (figura 11.18) seleccionamos el botón “SOCIALES” que nos llevará a la figura a) del ejemplo. Buscaremos la expresión deseada. Si no estuviera en esta pantalla esperaríamos a las posteriores hasta localizarla. En nuestro caso se encuentra en esta primera pantalla. Al pulsarlo, la aplicación mostrará de nuevo la pantalla inicial, eso sí, con el texto “HOLA” en el cuadro de texto (figura b) del ejemplo). A continuación escogeremos el botón “PERSONAS”. Haremos la misma operación que en el caso anterior: esperaremos a encontrar la imagen deseada. Esta vez habrá que esperar a una segunda pantalla (figura c) del ejemplo). Pulsaremos el botón “Abuela” y la aplicación nos devolverá a la pantalla inicial con el texto “HOLA ABUELA” en el cuadro de texto (figura d) del ejemplo).



a) Primera pantalla de la categoría "SOCIALES"



b) Pantalla inicial con el texto "HOLA" escrito



c) Segunda pantalla de la categoría "PERSONAS"



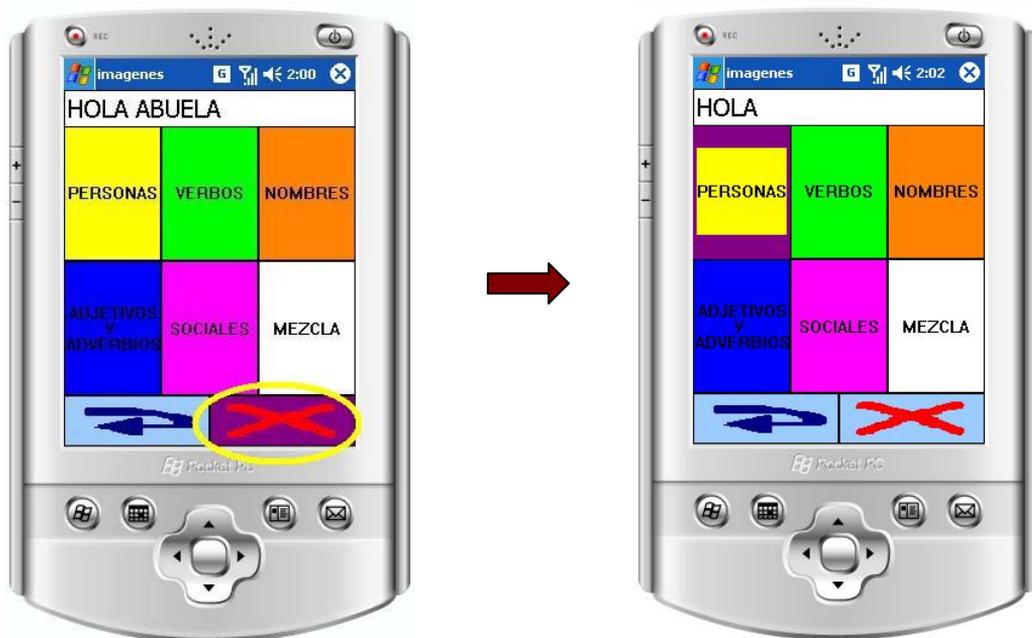
d) Pantalla inicial con el texto "HOLA ABUELA" escrito

Ejemplo 11.6: Escribir mediante imágenes

## 11.8 Borrar una palabra por el método de las imágenes

A diferencia de los métodos de escritura mediante caracteres, con el de imágenes únicamente se puede borrar una palabra completa. El botón que realiza tal operación es el representado mediante un aspa roja.

**Ejemplo 11.7:** Siguiendo el ejemplo anterior, si queremos ahora borrar la palabra “ABUELA” del cuadro de texto, pulsaremos el último botón de la pantalla (botón rodeado con una elipse amarilla en la figura a)).



a) Pantalla antes del borrado

b) Pantalla después del borrado

Ejemplo 11.7: Borrado de una palabra por el método de las imágenes.

Además de realizar el borrado desde la pantalla principal, cómo en el ejemplo 11.7, es posible hacerlo desde cualquiera de las pantallas de inserción de expresiones, ya que todas ellas poseen el botón de borrado.

## 11.9 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”. 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 11.21: 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.

- 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 11.22: Botón 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.

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.

## 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.

### 11.10 Menú

En todas las configuraciones posibles existe una pantalla MENÚ en la que pueden realizar varias funciones. Dependiendo de las características elegidas tendremos un formato u otro.

- **Menú básico:** Constará de cinco botones con las siguientes funcionalidades:
  - *PLAY*: Reproduce por voz el texto escrito en el cuadro de texto.
  - *CONFIGURACIÓN*: Cambia la configuración de la aplicación.
  - *AYUDA*: Consulta la ayuda que está integrada en la aplicación.
  - *VOLVER*: Vuelve a la pantalla principal.
  - *SALIR*: Sale de la aplicación.



Figura 11.23: Menú básico.

- **Menú básico más colores:** Será el menú básico anterior más un botón más correspondiente a los colores. Las funcionalidades de los botones son:
  - *PLAY*: Reproduce por voz el texto escrito en el cuadro de texto.
  - *CONFIGURACIÓN*: Cambia la configuración de la aplicación.
  - *COLORES*: Cambia los colores de apariencia de la aplicación.
  - *AYUDA*: Consulta la ayuda que está integrada en la aplicación.
  - *VOLVER*: Vuelve a la pantalla principal.
  - *SALIR*: Sale de la aplicación.



Figura 11.24: Menú con colores.

- **Menú básico más frases:** Será el menú básico anterior más un botón más correspondiente a las frases. Las funcionalidades de los botones son:
  - *PLAY*: Reproduce por voz el texto escrito en el cuadro de texto.
  - *CONFIGURACIÓN*: Cambia la configuración de la aplicación.
  - *FRASES*: Utiliza y gestiona frases ya predefinidas.
  - *AYUDA*: Consulta la ayuda que está integrada en la aplicación.
  - *VOLVER*: Vuelve a la pantalla principal.
  - *SALIR*: Sale de la aplicación.



Figura 11.25: Menú con frases.

- **Menú completo (básico, colores y frases):**

Será el menú básico anterior más un botón más correspondiente a los colores y otro más para las frases. Las funcionalidades de los botones son:

- *PLAY*: Reproduce por voz el texto escrito en el cuadro de texto.
- *CONFIGURACIÓN*: Cambia la configuración de la aplicación.
- *FRASES*: Utiliza y gestiona frases ya predefinidas.
- *COLORES*: Cambia los colores de apariencia de la aplicación.
- *AYUDA*: Consulta la ayuda que está integrada en la aplicación.
- *VOLVER*: Vuelve a la pantalla principal.
- *SALIR*: Sale de la aplicación.



Figura 11.26: Menú completo.

- **Menú de imágenes:** Esta configuración constará de cinco botones. El orden de aparición y la funcionalidad de cada uno de ellos será igual que en el menú básico. La diferencia radica en la apariencia de los botones.



Figura 11.27: Menú de imágenes.

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, este botón está representado por la palabra “MENU”, mientras que en la configuración de IMAGENES este botón está representado por una flecha azul, ovalada y apuntando hacia la izquierda (Figura 11.28).



Figura 11.28: Botón MENU en la configuración de IMAGENES

### 11.11 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. 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 11.29). 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.

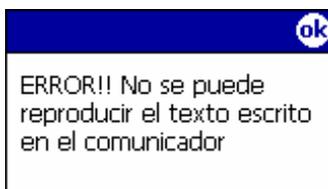


Figura 11.29: Mensaje reproducción por voz errónea.

### 11.12 Configuración de la aplicación

Se podrá configurar varios aspectos de la aplicación. Éstos variarán dependiendo de las características que se desea que tenga el comunicador: se puede disponer de un único método de escritura, que será el que viene por defecto (letras agrupadas), o bien tener un listado de posibles métodos de escritura a elegir.

Para realizar modificaciones en la configuración se deberá pulsar el botón “CONFIGURACIÓN” de la pantalla MENÚ. A continuación se mostrará una pantalla en función de la configuración con la que se esté trabajando en ese momento. En la figura 11.30 se muestra un listado de posibles pantallas de configuración:



a) Configuración con un único método de escritura.



b) Configuración con dos métodos de escritura disponibles (letras agrupadas y categorías).



c) Configuración con dos métodos de escritura disponibles (letras agrupadas e imágenes).



d) Configuración con dos métodos de escritura disponibles (letras agrupadas y vocales).

Figura 11.30: Pantallas CONFIGURACIÓN.

Como puede verse en la figura 11.30 existen diferencias entre las posibles configuraciones. El caso a) no dispone de cambio entre métodos de escritura ya que está configurado para que únicamente se pueda escribir mediante el método de letras agrupadas. Por tanto, sólo se podrá activar o desactivar el barrido e indicar el tiempo del mismo. En cambio, los casos b), c) y d) disponen de un listado de métodos disponibles indicado mediante el texto “Diseño de pantalla”: letras agrupadas y categorías para el caso b); letras agrupadas e imágenes para el caso c); y letras agrupadas y vocales para el caso d).

En el caso c), la pantalla de CONFIGURACIÓN se corresponde al método de escritura de imágenes y por tanto, no aparece la posibilidad de activar o desactivar el barrido. Esto es debido a la obligación de trabajar en la configuración de IMÁGENES con el barrido activado.

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 los segundos que el barrido está señalando a un botón.

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.

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 11.31).

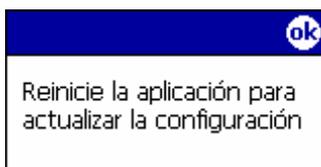


Figura 11.31: Aviso de reinicio

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.

### 11.13 Cambiar colores de los elementos de la aplicación

Cuando el usuario desee modificar el color de algún elemento de la pantalla deberá acceder al MENU y pulsar el botón “COLORES” (Figura 11.24 o Figura 11.26, según sea la configuración elegida). A continuación se mostrará la pantalla de la figura 11.32. En ella, el usuario pulsará el botón “CAMBIAR” situado al lado de los elementos que desee modificar.



Figura 11.32: pantalla COLORES

Los posibles cambios de color son los siguientes:

- **Color fondos:** para cambiar el fondo de las pantallas.
- **Color botones:** para cambiar el color de los botones de los métodos de escritura por caracteres.
- **Color texto botón:** para cambiar el color de texto de los botones de los métodos de escritura por caracteres.
- **Color texto:** para cambiar el color del texto informativo escrito en las pantallas.

Una vez pulsado el botón “CAMBIAR” del elemento deseado, se mostrará la figura 11.33, donde aparecerá una paleta de colores. El usuario pulsará en el botón del color deseado.

Al pulsar sobre el botón, la aplicación mostrará la pantalla representada en la Figura 11.32 (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 11.33: pantalla de selección de color

**Ejemplo 11.8:** Supongamos que se quiere cambiar el color de fondo de la aplicación estando en el método de letras agrupadas. Para ello, pulsaremos el botón “MENU” de la pantalla principal que nos llevará a la Figura 11.24 o Figura 11.26, según sea la configuración elegida, y se pulsará el botón “COLORES”. A continuación seleccionaremos el botón “CAMBIAR” situado a la derecha del texto “Color fondos”. Aparecerá la pantalla de selección del color (Figura b) del ejemplo) donde se escogerá el color deseado para el fondo de la pantalla, tras lo cual, se mostrará de nuevo la pantalla de configuración de colores con el cambio ya realizado (Figura c) del ejemplo).



a) Selección del elemento

b) Selección del color

c) Cambio realizado

Ejemplo 11.8: Cambio de color del fondo de pantalla.

## 11.14 Empleo de frases

La pantalla frases (Figura 11.34) dispone de un conjunto de frases que permiten agilizar la comunicación. Esta pantalla aparecerá tras pulsar el botón “FRASES” del menú de la figura 11.25 o de la figura 11.26.

Las posibilidades que tenemos son: insertar en el cuadro de texto frases que ya están almacenadas, guardar frases nuevas para poder emplearlas posteriormente y borrar las frases que no utilicemos o no creamos conveniente tener almacenadas.

La pantalla de frases consta del cuadro de texto con el texto escrito por el usuario, un listado de frases disponibles para su utilización y seis botones cuya funcionalidad se describirá a continuación por orden de aparición:



Figura 11.34: Pantalla frases.

- **Botón AÑADIR:** Insertará al final del texto escrito la frase que actualmente esté marcada (fondo azul).
- **Botón ^ :** Permite recorrer las frases del listado de forma ascendente. Cuando se sitúe en la primera frase, la siguiente pulsación nos llevará a la última del listado.
- **Botón GUARDAR:** Añadirá el texto escrito hasta el momento a la lista de frases. Se almacenará al final de la lista. De esta forma se podrá guardar aquellas frases que el usuario utilice frecuentemente.
- **Botón BORRAR:** Eliminará de la lista de frases aquella que se encuentre seleccionada en ese momento.
- **Botón V :** Permite recorrer las frases del listado de forma descendente. Cuando se sitúe en la última frase, la siguiente pulsación nos llevará a la primera de la lista.
- **Botón VOLVER:** vuelve a la pantalla anterior.

**Ejemplo 11.9:** Si el usuario quiere insertar la frase “QUIERO IR AL SERVICIO” tras situarse en la figura a) del ejemplo, deberá seleccionar la frase deseada mediante los botones de desplazamiento (**^**, **v**) y a continuación pulsar el botón “AÑADIR”. La frase se mostrará en el cuadro de texto (Figura b) del ejemplo).



a) Elección de la frase a añadir.

b) Frase añadida al cuadro de texto.

Ejemplo 11.9: Añadir una frase predefinida

## 11.15 Uso de la ayuda en la aplicación

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.

Para consultar la ayuda se debe acceder al MENU y pulsar el botón “AYUDA”. Una vez hecho esto, se mostrará una de las dos pantallas correspondientes a las figuras 11.35 y 11.36. Si el comunicador sólo dispone de un método de escritura disponible (letras agrupadas), se visualizará la pantalla de la figura 11.35. Si, por el contrario, la aplicación tuviera la posibilidad de comunicarse mediante varios métodos de escritura, la pantalla sería la de la figura 11.36. La diferencia entre estas dos radica en la configuración. En el primer caso sólo mostrará información sobre cómo configurar el barrido mientras que en el segundo caso, la ayuda sobre cómo configurar atenderá a cómo modificar el barrido y cómo cambiar el método de escritura de entre los posibles en la aplicación.

En estas pantallas se elegirá la ayuda mediante la pulsación del botón correspondiente. A continuación se mostrará una pantalla compuesta por un título, un cuadro de texto en el que explica brevemente cómo se puede realizar la acción determinada y un botón “VOLVER” para regresar a la pantalla anterior (Figura 11.37).



Figura 11.36: Temas de ayuda para el caso de disponer de más de un método de escritura.



Figura 11.35: Temas de ayuda para el caso de disponer de un solo método de escritura.



Figura 11.37: Ayuda sobre cómo configurar el barrido.

Si el tema elegido es BARRIDO de la figura 11.35, se mostrará la pantalla de la figura 11.37. Ahora bien, si el tema es CONFIGURACION de la figura 11.36, en vez de mostrarse una pantalla de ayuda, se visualizará una nueva pantalla de elección de temas (Figura 11.38) compuesta por los siguientes temas:

- **TIPOS:** Ayuda sobre la manera de elegir otra configuración distinta de la actual.
- **BARRIDO:** Ayuda sobre cómo activar o desactivar el barrido y sobre cómo se puede cambiar el tiempo de barrido.

De esta forma, pulsando en el botón “TIPOS” se mostrará la pantalla de la figura 11.39 mientras que el con el botón “BARRIDO” aparecerá la pantalla de la figura 11.37.



Figura 11.38: Elección de temas para la ayuda de CONFIGURACION.



Figura 11.39: Ayuda sobre cómo cambiar el método de escritura.

## 11.16 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. 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 11.40).



Figura 11.40: Botón salir en la configuración IMAGENES



# **PARTE 6. CONCLUSIONES**

---



## Capítulo 12

# CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO

En este capítulo se expondrán las conclusiones derivadas de la realización del presente proyecto, las principales dificultades encontradas, los objetivos alcanzados tras el mismo, los conocimientos adquiridos y las futuras líneas de trabajo propuestas.

### 12.1 Conclusiones acerca de las líneas de productos

En el proyecto realizado se han estudiado cuatro proyectos de fin de carrera ya elaborados, y se ha derivado de ellos una línea de comunicadores para personas discapacitadas. Una vez desarrollada dicha línea, se han instanciado a modo de ejemplo una serie de comunicadores con distintas funcionalidades.

Las líneas de productos, son un paso más en la reutilización del software. Así como la producción en cadena y la personalización de los productos, supuso un cambio en la industria de los productos manufacturados, las líneas de productos suponen, a nuestro juicio, un cambio en el desarrollo del software al aplicar esos mismos conceptos. Aportan muchas mejoras, como ya vimos en el capítulo 4:

- Entrega de productos software de manera más rápida y económica.
- Mejoras en tiempo de entrega del producto y costes de ingeniería.
- Incremento en el número total de productos que pueden ser elaborados y mantenidos.
- Reducción en el esfuerzo promedio requerido para desarrollar y mantener los productos.
- Mayor agilidad para expandir el negocio a nuevos mercados.

Estas ventajas se aprecian de un modo mayor, cuantos más proyectos se tengan que desarrollar con dicha línea. Para crear una única aplicación, las líneas de productos no son eficientes. Su verdadero potencial está en la creación de varias aplicaciones similares, que se distinguen dependiendo de las preferencias o necesidades del cliente.

La figura 12.1 ilustra el modelo de coste de las líneas de productos frente a los sistemas simples. Sin líneas de productos, el coste acumulativo empieza en cero y crece de manera uniforme. Con líneas de productos, el coste acumulativo comienza con el coste de los elementos comunes y luego crece a un ritmo menor según se desarrollan los proyectos. El punto en el que se cruzan las dos líneas es el punto donde las líneas de productos empiezan a ser verdaderamente rentables. Ese punto, puede ser expresado en términos del número de sistemas creados.

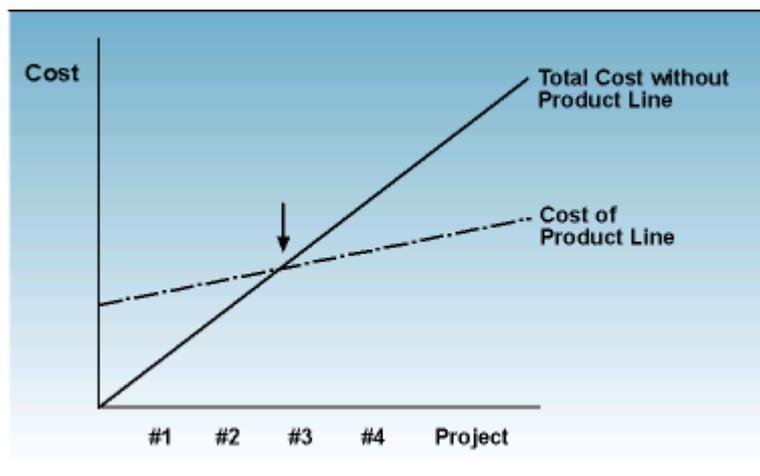


Figura 12.1: Evolución de costes. Línea de productos frente a sistemas simples<sup>7</sup>

El punto en el que las líneas de productos empiezan a resultar rentables, es cuando se tienen que elaborar más de dos sistemas similares. A partir de ahí, el coste inicial de desarrollar los componentes reutilizables, y establecer la estructura de la línea de productos, empieza a verse superado por los beneficios de la reutilización.

Este dato pone de manifiesto el increíble avance que supone la utilización de líneas de productos, ya que, cuando se establece que se va a diseñar una línea de productos se hace, en la gran mayoría de los casos, con el objetivo de crear más de dos aplicaciones similares. Por tanto, la rentabilidad de las líneas de productos está asegurada.

Otro hecho que queremos destacar es la importancia de la **gestión de la variabilidad** en las líneas de productos. El mecanismo de combinación de paquetes (package merge), es un excelente método para gestionar la variabilidad en la capa lógica de la línea de productos. El problema principal viene en la interfaz de los comunicadores. En nuestro caso, dicha interfaz estaba almacenada en un fichero .xml y tuvimos que encontrar un método eficaz para poder captar, de algún modo, la variabilidad dentro del fichero. Finalmente se optó por seguir con la filosofía del método de combinación de paquetes, y se dividió el archivo .xml en varios ficheros. Cada fichero nuevo creado, correspondía a una parte bien diferenciada de la interfaz. De este modo, y mediante objetos DataSet, pudimos ir combinando los archivos que se necesitaban dependiendo de la interfaz final de la aplicación a desarrollar. El empleo de objetos DataSet nos proporcionó un método simple, organizado y efectivo de gestionar la variabilidad de un fichero .xml.

<sup>7</sup> Clements, Paul. "It Takes Two". News@sei. Volume 6, Number 4. Fourth quarter 2003. <http://www.sei.cmu.edu>. Abril 2007.

Sin embargo, debemos señalar que la gestión de dicha variabilidad es el principal problema que se encuentra en el desarrollo de una línea de productos.

También tenemos que destacar la importancia de una buena labor de planificación y documentación, especialmente cuando se va a derivar una línea de productos de unos proyectos ya elaborados previamente.

## 12.2 Dificultades encontradas

A continuación se detallan las principales dificultades que han surgido durante la elaboración y desarrollo del presente proyecto de fin de carrera:

- La documentación de los proyectos fin de carrera que empleamos para derivar de ellos la línea de productos, no cumplía las expectativas que teníamos. Ello alargó el tiempo de desarrollo del proyecto.
- Los proyectos, aunque todos tenían una finalidad similar (el desarrollo de un comunicador para personas discapacitadas), ofrecían un diseño e implementación muy dispar. Por otra parte, las funcionalidades que éstos implementaban presentaban cierto acoplamiento. Tuvimos que estudiar a fondo la implementación de los proyectos utilizados, y elaborar una considerable cantidad de código de adaptación.
- La lista y el diagrama de características tuvieron que revisarse durante el diseño y la implementación, pues descubrimos características que no se tuvieron en cuenta en la fase de análisis. Afortunadamente, el buen diseño de la línea permitió la inclusión de dichas características de una manera sencilla.
- El gestionar la variabilidad de la interfaz. El método de combinación de paquetes es excelente en la capa lógica, pero en nuestro caso no se podía aplicar directamente a la capa de la interfaz, pues ésta se almacenaba en un único archivo .xml. Para respetar en la medida de lo posible la naturaleza de los proyectos estudiados, se decidió conservar el método de almacenamiento por XML. Y se intentó buscar algún método que fuera compatible con la filosofía de la combinación de paquetes. Gracias al empleo de objetos DataSet pudimos superar esta dificultad de una manera directa, rápida y eficaz.
- La programación para dispositivos móviles en C# y el empleo de clases parciales nos era desconocida. Por suerte, la documentación sobre estos temas es extensa y pudimos adquirir conocimientos en estas áreas de un modo relativamente sencillo.
- El error que apareció al probar cualquier proyecto que contuviera el método de imágenes. Visual Studio no permitía incluir tantos ficheros en un archivo CAB. En el capítulo 8: Implementación, se dispone de más información sobre el problema y la solución que empleamos.
- Intentar establecer una forma de crear aplicaciones con la línea de productos, reduciendo al máximo la cantidad de código necesario a escribir. Se consiguió de una manera muy efectiva mediante compilación condicional.

### **12.3 Objetivos alcanzados**

Con el desarrollo del presente proyecto, se han conseguido la gran mayoría de los objetivos propuestos al inicio y que a continuación se detallan:

- Se ha trabajado con el concepto de Línea de Productos, y se han analizado sus características y ventajas.
- Se ha realizado un estudio de cuatro proyectos fin de carrera ya elaborados, y se han combinado para analizar y diseñar la línea de productos que se deriva de ellos.
- Se ha desarrollado dicha línea de productos utilizando el concepto de combinación de paquetes, y aplicando la capacidad de clases parciales de C#, de manera que, con el menor esfuerzo posible se pueden crear nuevos comunicadores para personas discapacitadas, seleccionando las funcionalidades o características más convenientes.
- Se han desarrollado con la línea de productos de manera satisfactoria, una serie de comunicadores a modo de ejemplo, cada uno con distintas características.
- Se han empleado conocimientos adquiridos a lo largo de la carrera.

Aparte de cumplir los objetivos propuestos en el proyecto, también se han conseguido logros que ha mejorado en parte el resultado del mismo.

- Mediante la elaboración de la línea de productos, no sólo se pueden desarrollar las cuatro aplicaciones correspondientes a cada proyecto de fin de carrera estudiado, sino que tenemos la posibilidad de obtener 64 combinaciones posibles y por tanto, 64 aplicaciones distintas.
- Pensando en posibles ampliaciones y uso de futuros desarrolladores de comunicadores para discapacitados, se ha implementado la línea de productos de manera que con el uso de una simple “plantilla” de código y mediante el uso de constantes, simplemente tenga que indicar qué característica quiere añadir a su aplicación final. De esta forma, no tendrá necesidad de crear código nuevo haciendo un buen uso, por tanto, de las ventajas que la línea de productos nos ofrece.
- De igual manera que se puede gestionar la variabilidad lógica de la aplicación mediante el método de combinación de paquetes y las clases parciales de C#, hemos obtenido un método similar válido para gestionar la variabilidad del XML mediante objetos DataSet.

### **12.4 Conocimientos adquiridos**

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

- Nos hemos familiarizado con la tecnología .NET, en especial con el lenguaje C#, el empleo de clases parciales y el desarrollo de aplicaciones para dispositivos móviles.
- Hemos realizado la planificación de un proyecto de tamaño considerable, estimando tiempos y costes, analizando riesgos y elaborando planes de contingencia de los mismos.

- Hemos conocido, desarrollado y aplicado el concepto de línea de productos, y conocido sus características y beneficios.
- Hemos conocido el mundo de la informática para personas discapacitadas, y comprendido la gran ayuda que la informática puede prestar a esas personas.

## **12.5 Futuras líneas de trabajo**

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

- Integración mediante Visual Studio, de un mecanismo de configuración automática, similar al plugin fimp de eclipse, de modo que al configurar un determinado producto, se agreguen los paquetes automáticamente, sin tener que modificar el archivo .csproj.
- Añadir nuevas funcionalidades posibles a los comunicadores, por ejemplo: Poder usar un mecanismo de predicción de texto, para que la escritura sea más ágil. Posibilidad de comunicación remota entre dos dispositivos móviles, o de uno de éstos a un ordenador, para enviar y recibir los mensajes creados con los comunicadores.



# REFERENCIAS

---



## BIBLIOGRAFÍA

- Clements, Paul; Northrop, Linda. *Software Product Lines – Practices and Patterns*. Addison-Wesley, 2001.608pp. ISBN: 0-201-70332-7
- Gomaa, Hassan. *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*. Addison-Wesley, 2004. 736pp. ISBN: 0-201-77595-6.
- Greenfield, Jack; Short, Keith *et al.* *Software Factories: Asembling Applications with Patterns, Models, Frameworks and tools*. John Wiley & Sons, 2004. 666pp. ISBN:0-471-20284-3
- Hoffman, Kevin, *Visual C# 2005*, Anaya Multimedia, 2007.732pp. ISBN: 84-415-2098-4
- Laguna, Miguel A.; González-Baixauli, Bruno; López, Oscar. *Gestión de la Variabilidad en Líneas de Productos*. En: XXXIII Conferencia Latinoamericana de Informática, San José - Costa Rica, oct. 2007
- Laguna, Miguel A.; González-Baixauli, Bruno; Marqués Corral, José Manuel. *Seamless Development of Software Product Lines: Feature Models to UML Traceability*. En: Sixth International Conference on Generative Programming and Component Engineering (GPCE 07). Salzburg, Austria - oct 2007
- Laguna, Miguel A.; González-Baixauli, Bruno. *Variabilidad, Trazabilidad y Líneas de Productos: una Propuesta basada en UML y Clases Parciales*. En: XII Jornadas Ingeniería del Software y Bases de Datos (JISBD 2007), Zaragoza, Spain - sep 2007. Disponible en: <http://www.sistedes.es/sistedes/pdf/2007/JISBD-07-laguna-variabilidad.pdf>
- Muthig, Dirk et al. *GoPhone – A Software Product Line in the Mobile Phone Domain*. Fraunhofer IESE, 2004. 113p. IESE-Report No. 025.04/E
- Pressman, Roger S., *Ingeniería del Software. Un enfoque práctico*, Mc Graw Hill, 2002.640pp. ISBN: 84-481-3214-9
- Sommerville, Ian, *Ingeniería del Software*, Addison Wesley, 2002.687pp. ISBN: 84-782-9074-5

## RECURSOS DIGITALES

### C# y Clases parciales

- Bazuzi, Jay. *Winforms designed code and C# partial classes*, [http://blogs.msdn.com/jaybaz\\_ms/archive/2004/04/28/122392.aspx](http://blogs.msdn.com/jaybaz_ms/archive/2004/04/28/122392.aspx), Febrero, 2008
- Microsoft MSDN. *Definiciones de Clases Parciales (Guía de programación en C#)*, [http://msdn.microsoft.com/es-es/library/wa80x488\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/wa80x488(VS.80).aspx), Enero, 2008
- OnDOtNet.com. McDEonald, Matthew. *New Language Features in C# 2.0 Part 1*, <http://www.ondotnet.com/pub/a/dotnet/2004/04/05/csharpwhidbeypt1.html>, Diciembre, 2007.
- Torres Tatis, Alvaro. *Tipos parciales en C # 2.0*, [http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ\\_1903/default.aspx](http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_1903/default.aspx), Enero, 2008.
- Wikibooks.org. *Manual de C sharp NET*, [http://es.wikibooks.org/wiki/C\\_sharp\\_NET/\\_Texto\\_Completo](http://es.wikibooks.org/wiki/C_sharp_NET/_Texto_Completo), Noviembre, 2007.

### Documentación y Planificación

- JMRR – *Proyectos Fin Carrera. Guía de creación de un cuaderno de bitácora para la realización de Proyectos de Fin de Carrera de ITI*, <http://pfc-jmrr.blogspot.com/>, Mayo, 2007.
- Montesa Andrés, J., *Evaluación Organización y Gestión de Proyectos Informáticos*, <http://www.upv.es/~jmontesa/eog-ind.html>, Mayo, 2007

### Líneas de productos software

- Clements, Paul. *It Takes Two*. Software Engineering Institute de la Universidad Carnegie Mellon. news@sei, Volume 6. Number 4. Fourth quarter 2003. <http://www.sei.cmu.edu/news-at-sei/columns/software-product-lines/2003/4q03/software-product-lines-4q03.htm>. Abril 2007.
- Comunidad de desarrolladores interesados en líneas de productos. <http://www.softwareproductlines.com/>, Abril, 2007.
- De Lara, Juan. Introducción al Curso de Postgrado: *Diseño de software basado en Modelado y simulación*. Universidad Autónoma de Madrid. 2006. [http://astreo.ii.uam.es/~jlara/doctorado.2006/presentacion\\_curso.pdf](http://astreo.ii.uam.es/~jlara/doctorado.2006/presentacion_curso.pdf), Abril, 2007.

- 
- Montilva, Jonás A. Desarrollo de Software basado en Líneas de Producto Software. Dentro de las Conferencias del Programa DVP del IEEE Computer Society Sección Argentina. 2006, <http://www.ieee.org.ar/downloads/2006-montilva-productos.pdf>, Abril, 2007.
  - Página de la Software Product Line Conference 2008 del Software Engineering Institute de la Universidad Carnegie Mellon, <http://www.sei.cmu.edu/productlines/>, Mayo, 2008.

## Visual Studio y .NET Compact Framework

- Elguille, la Web del Visual Basic, C#, .NET y más...*.NET Compact Framework y Smart Device Extension*, <http://www.elguille.info/NET/netCF/netCF.htm>, Diciembre, 2007.
- Microsoft MSDN *.NET Framework Developer Center. .NET Compact Framework*. En: <http://msdn.microsoft.com/en-us/netframework/aa497273.aspx>, Noviembre, 2007.
- *Microsoft Visual Studio .NET Tutorials (Español)*. Disponible en: <http://www.vtc.com/products/Microsoft-Visual-Studio-.NET-Espanol-tutorials.htm>, Diciembre, 2007.

## XML y DataSets

- Curso de ASP NET Avanzado.adrformacion.com. Capítulo 6.- DataSet. <http://www.adrformacion.com/cursos/aspnet35av/leccion3/tutorial5.html>. Febrero 2008.
- DaniWeb. *Comparar dos archivos xml con Csharp*. Hilo de DaniWeb Home > Forums > Software Development > C#, <http://www.daniweb.com/forums/thread46345.html>, Enero, 2008.
- Java2s.com. *Creación de un documento XML en C#: Elemento xml y propiedades*, <http://www.java2s.com/Code/CSharp/XML/CreateXMLdocumentxmlelementandproperties.htm>, Enero, 2008.
- Dev Shed Forums. *Escribir un archivo XML usando código C#*. Hilo de Dev Shed Forums > Programming Languages - More > .Net Development, <http://forums.devshed.com/net-development-87/writing-an-xml-file-using-c-code-197961.html>, Enero, 2008.
- Narayanaswamy, Anand. *Manipular un fichero de datos XML usando C#*, <http://www.developer.com/net/csharp/article.php/3489611>, Enero, 2008.



**APÉNDICE A.  
TECNOLOGÍA UTILIZADA**

---



## **A.1 MICROSOFT .NET**

### **A.1.1 Introducción**

Microsoft .NET es un conjunto 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.1.2 .NET Framework**

.NET Framework es un 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). Incorpora servicios Web XML, que 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 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.

### **A.1.3 .NET Compact Framework**

.NET Compact Framework aporta la eficacia del entorno de programación .NET Framework a los dispositivos compactos. 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, dispositivos de computación para automóviles, etc.

.NET Compact Framework es un subconjunto de la biblioteca de clases .NET Framework y también contiene clases diseñadas expresamente para él. Hereda la arquitectura completa de Common Language Runtime y la ejecución de código administrado. 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.1.4 Microsoft Visual Studio.NET**

Es una familia de herramientas de desarrollo de software de Microsoft, orientadas hacia su entorno de programación .Net Framework. 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.

Visual Studio.Net 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 2005, hay perfiles que permiten crear aplicaciones para Pocket PC 2003, Smartphone 2003, Windows CE 5.0 y Windows Mobile 6 Standard. Estos perfiles permiten crear aplicaciones que incluyen Windows Forms y ADO.NET.

### A.1.5 C#

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de la plataforma .NET. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes. 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.

## A.2 OBJETOS DATASET Y ARCHIVOS .XML

### A.2.1 Introducción

En la elaboración de la línea de productos, se han utilizado objetos DATASET y archivos de tipo .xml para poder representar la variabilidad de la interfaz de los distintos componentes de la línea. A continuación, se explicarán brevemente estos conceptos.

### A.2.2 Objetos DataSet

El DataSet de .NET Framework es una representación de datos residente en memoria, que proporciona un modelo de programación relacional coherente, independientemente del origen de datos que contiene (gracias a xml). Un DataSet representa un conjunto completo de datos, incluyendo las tablas que contienen, ordenan y restringen los datos, así como las relaciones entre las tablas.

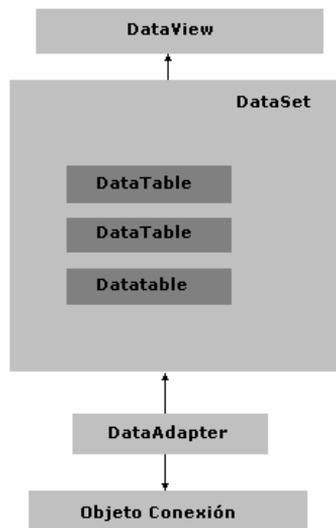


Figura A.1: Modelo de objetos de un DataSet<sup>8</sup>

<sup>8</sup> Curso de ASP NET avanzado. Capítulo 6: DataSet. adrformacion.com.  
<http://www.adrformacion.com/cursos/aspnet35av/leccion3/tutorial5.html>. Febrero 2008.

Como puede verse en la figura A.1, un DataSet es un modelo de datos relacional, con lo que tenemos tablas, con sus filas y columnas, relaciones y propiedades extendidas.

- **Tablas.** Se compone de un objeto DataTableCollection el cual puede formarse por una o más DataTable. Un objeto DataTable representa una de las tablas de nuestra fuente de datos. Un DataTable está formado a su vez por colecciones de filas (Rows) y columnas (Columns) que podrán contener una o más filas (DataRow) o columnas (DataColumn).
- **Relaciones.** Se trata de colecciones de relaciones que pueden asociar tablas con claves foráneas.
- **Propiedades extendidas.** Hace referencia a colecciones de propiedades que pueden ser asignadas al DataSet en el momento de ser creado.

Tenemos tres modos diferentes de trabajar con los objetos DataSet:

- Crear objetos DataTable dentro del DataSet para almacenar los datos mediante programación.
- Recoger la información de una base de datos, de modo que llenemos el DataSet con estos datos mediante un objeto "puente" entre el DataSet y la base de datos que recibe el nombre de DataAdapter.
- Trabajar con la información mediante fuentes de datos XML.

En este proyecto se ha utilizado el último método, de modo que se rellenan las tablas del DataSet utilizando información contenida en fichero .XML para su posterior tratamiento. Para llevarlo a cabo, se puede utilizar la secuencia o el documento XML para suministrar datos al DataSet, suministrar información de esquema o ambas cosas.

La información aportada desde la secuencia o el documento XML puede combinarse con datos o información de esquema existente ya presente en el DataSet. En una representación XML de un DataSet, los datos se escriben en XML y el esquema, si está incluido implícitamente, se escribe utilizando el lenguaje de definición de esquemas XML (XSD).

Para cargar en un DataSet los datos XML y/o su esquema, se puede utilizar el método ReadXML del DataSet. Estos datos se pueden leer directamente desde un archivo, un objeto Stream, un objeto XmlWriter o un objeto TextWriter. Además, este método admite un parámetro adicional, XMLReadMode, que permite especificar cómo cargar el esquema y/o los datos xml en el DataSet. Los valores posibles del parámetro son:

- **DiffGram.** Lee un DiffGram y aplica los cambios del DiffGram al DataSet. (Un DiffGram es un formato XML que identifica las versiones actual y original de los elementos de datos).
- **Fragment.** Lee documentos XML que contienen fragmentos de un esquema reducido de datos XML (XDR, *XML-Data Reduced*).
- **IgnoreSchema.** Pasa por alto cualquier esquema interno y lee los datos en el esquema DataSet actual.

- *InferSchema*. Pasa por alto cualquier esquema interno, deduce el esquema a partir de los datos y carga los datos. Si el DataSet ya contiene un esquema, InferSchema extiende el esquema actual; para ello, agrega columnas a las tablas existentes y agrega tablas nuevas si no existen.
- *ReadSchema*. Lee cualquier esquema interno y carga los datos.
- *Auto*. Es la opción predeterminada. Realiza la acción más apropiada.

En el proyecto, se han tenido que unir varios y distintos archivos .XML en un único archivo, por lo cual, se ha prestado especial atención a los esquemas .XSD para poder asegurar la integridad de los datos empleados. Se ha elaborado un esquema .XSD general que contemplara los esquemas de los distintos .XML, y con él, se han cargado los datos en los objetos DataSet que se utilizaron.

A continuación se hablará sobre los archivos .XML y los esquemas XSD más detenidamente.

### A.2.3 XML

XML es un acrónimo cuyo significado en inglés es eXtensible Markup Language (Lenguaje de formato ampliable). Es decir, es un lenguaje de marcas, 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. Sus principales características son:

- 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.

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.
- 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, etc. 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. 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 los agentes usuarios de xml: navegadores, sistemas “push”, etc.

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, y compatible con estándares. 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.

El espacio de nombres System.XML, 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 (W3C) DOM. También se explican las consultas XPath (XML Path Language) y las transformaciones XSLT (Extensible Stylesheet Language Transformations). Las clases principales del espacio de nombres XML son:

- 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 **XsltTransform** 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.2.4 XSD

Los archivos con extensión .XSD (Xml Schema Definition) son esquemas XML. Un esquema XML es una herramienta compleja y eficaz para crear y validar la estructura de los documentos XML compatibles. De forma parecida al modelado de datos de una base de datos relacional, un esquema proporciona una forma de definir la estructura de los documentos XML al especificar los elementos que se pueden utilizar en ellos, así como la estructura y tipos que dichos elementos deben tener para ser válidos con respecto al esquema específico.

En el esquema, se describe el contenido de los elementos del XML al que alude, mediante código XML a su vez. Los elementos y atributos se declaran con las etiquetas element y attribute, y la estructura se crea con las etiquetas simpleType y complexType.

Los esquemas .XSD proporcionan las siguientes mejoras sobre las DTD (Document Type Definition – Definiciones de tipo de documento):

- Con el esquema se disponen de tipos de datos adicionales.
- Se pueden crear tipos de datos personalizados.
- Utiliza sintaxis XML.
- Admite conceptos de la programación orientada a objetos, como polimorfismo y herencia.

Mediante Visual Studio .NET, podemos ver un esquema .XSD de una manera gráfica, donde se puede apreciar claramente, la correspondencia de los datos contenidos en el XML con una base de datos relacional, como se muestra en la figura A.2:

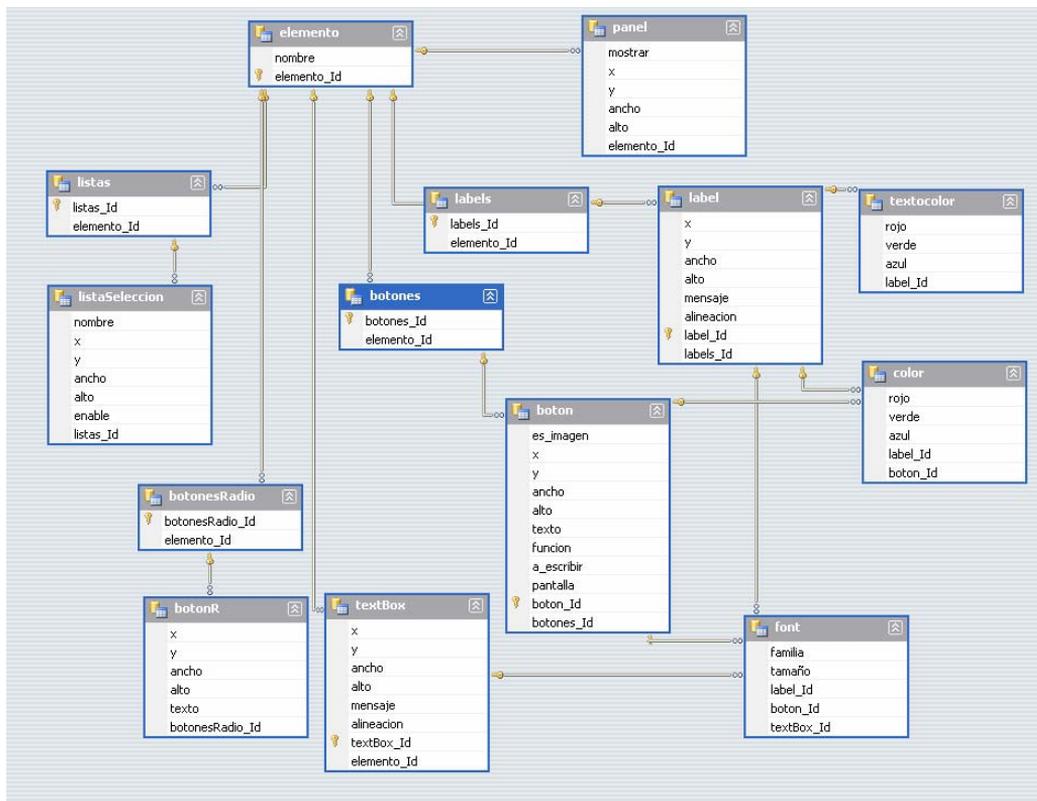


Figura A.2: Ejemplo gráfico de esquema XSD.

**APÉNDICE B.  
ESTRUCTURA DE LOS  
FICHEROS .XML**

---



## B.1 CONSIDERACIONES PREVIAS

### B.1.1 Introducción

La mayoría de los proyectos empleados en la elaboración de esta línea de productos utilizan, como método de almacenamiento de la información, los archivos .XML. En ellos se guarda información, no sólo de la configuración de la aplicación, sino también de la interfaz de la misma.

Al desarrollar una aplicación mediante la línea de productos, su interfaz dependerá de las características propias de la aplicación desarrollada. Esto significa que en la elaboración de la línea de productos hay que transformar las interfaces de los proyectos de donde se infiere en una interfaz “variable”, de modo que, construyendo distintas aplicaciones, éstas tengan distintas interfaces adaptadas a sus necesidades.

Para poder tratar esta variabilidad en la interfaz, se ha optado por seguir con la filosofía del método de combinación de paquetes o *package merge*, de modo que, el paquete base tendrá un archivo .xml “base”, con los elementos de la interfaz comunes a todas las aplicaciones que puedan elaborarse. Del mismo modo, se tendrán paquetes que se irán añadiendo o no al paquete base, dependiendo de las características elegidas para la aplicación a desarrollar mediante la línea de productos. En estos paquetes, se hallarán archivos XML que implementen las partes de la interfaz necesarias para las características que se correspondan con tales paquetes. Esto significa que del mismo modo que se combinan las clases parciales de C#, se combinarán también los distintos archivos .XML en uno único que la aplicación final utilizará para guardar la información de su interfaz final.

Dicha combinación de ficheros .XML se realizará mediante objetos DataSet, como se explica en el punto 8.2.2 (Tratamiento de los archivos XML) de la presente memoria. Para más información sobre archivos .XML y objetos DataSet, ver el apéndice A: Tecnología Utilizada.

Hay que aclarar, que se dispondrán de dos tipos básicos de ficheros XML: los ficheros de interfaz ya nombrados y el fichero de configuración, llamado configuración.xml, donde se guardará información sobre los datos específicos de configuración de la aplicación: tiempo de barrido, método de escritura, etc.

## **B.1.2 División y organización de los archivos XML de interfaz**

En los archivos .XML de interfaz, se almacena información sobre los elementos que componen dicha interfaz. Como se ha explicado anteriormente, para poder expresar la variabilidad de la interfaz del comunicador, se ha dividido la interfaz en las siguientes partes diferenciadas:

- Pantalla de Bienvenida + Método de escritura.
- Pantalla de Menú.
- Pantalla de Configuración.
- Pantallas de Frases (si las hay)
- Pantallas de Selección de Colores (si las hay)
- Pantallas de Ayuda.

Hay que tener presente que para el correcto funcionamiento de la aplicación, la combinación de los ficheros .XML en uno sólo debe realizarse siguiendo estrictamente el orden señalado anteriormente.

En el punto 8.2.1 (Utilización de los ficheros .XML) de la memoria, se puede encontrar información sobre la división de los ficheros XML originales y su organización en paquetes.

A continuación se explicarán los esquemas generales, tanto del fichero de configuración de la aplicación, como de cualquier fichero .XML de interfaz empleado en este proyecto. También se explicará el esquema detallado para cada elemento de la interfaz:

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

De este modo, la edición de cualquier archivo .XML del proyecto se realizará de manera sencilla pudiendo crear, por ejemplo, gracias al método de escritura configurable, un nuevo modo de escritura definido a gusto del usuario final.

## **B.2 ESTRUCTURA DE LOS FICHEROS .XML**

### **B.2.1 Estructura general de los ficheros .XML**

La definición de los datos de una interfaz 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.

El esquema que sigue un fichero de interfaz completo 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>
```

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.2.2 Información sobre las pantallas (paneles)

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 o pantalla.

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

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

### B.2.3 Información sobre 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 configuración.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.2.4 Información sobre los botones

Se disponen en el proyecto, dos tipos de botones básicos bien diferenciados: Botones de Caracteres y Botones de Imágenes. Los últimos tienen una imagen asociada. 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.2.4.1 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 muestran y explican los diferentes formatos.

```
<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) Función: Ir a otra pantalla

```
<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>
  <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>
  <font>
    <familia></familia>
    <tamaño></tamaño>
  </font>
</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 los anteriores esquemas 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 al 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.2.4.2 Botones de Imágenes

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 muestran y explican los diferentes formatos.

```
<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 carácter

```
<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>ATRAS</funcion>
  <pantalla></pantalla>
  <imagennormal></imagennormal>
  <imagenfoco></imagenfoco>
</boton>
```

d) Función: Ir a pantalla anterior

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

e) Función: Borrar Palabra

```
<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_Y_VOLVER</funcion>
  <pantalla></pantalla>
  <imagennormal></imagennormal>
  <imagenfoco></imagenfoco>
</boton>
```

g) Función: Borrar Palabra

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

h) Función: Salir de la aplicación

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 “si” 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>`. 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 >
```

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.2.5 Información sobre los botones de radio**

Los botones de radio 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>
```

## B.2.6 Información sobre 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. Además, cuando se esté trabajando con frases, dichas frases se dispondrán en una lista de selección para poder seleccionar aquella que se quiera añadir al texto o borrar. 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>
```

Entre las etiquetas <enable></enable> puede aparecer la palabra 'si', si se desea que la lista esté activa, o 'no', si se desea que la lista no esté activa. Sirve para, por ejemplo, si sólo se trabaja con un método de escritura, deshabilitar la lista de selección de métodos en la pantalla de configuración.

## B.2.7 Información sobre el fichero de configuración

El Esquema que sigue el fichero configuración.xml 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>
    </colorBotonDefecto>
  </configuracion>
</aplicacion>
```

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

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 “sí” 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.**  
**CONTENIDO DEL CD**

---



---

A continuación se indican el contenido del CD incluido en la memoria, organizado por carpetas:

- **Memoria**

En este directorio se encuentra la memoria íntegra, en formato PDF.

- **Línea de productos**

Aquí se dispone de la solución de Visual Studio .NET 2005 correspondiente a la línea de productos, con todos los paquetes y archivos necesarios, y lista para desarrollar aplicaciones a partir de ella.

- **Configuración de productos**

En este directorio se dispone todo lo necesario para la configuración de productos. Se incluye el Feature-Modelling Plug-in, así como Eclipse 3.2 y demás archivos necesarios para su correcto funcionamiento: Java 1.4.2, Eclipse Modelling Framework (emf) y xsd runtime. También se encuentra disponible en este directorio, el modelo de características elaborado para el proyecto y un archivo léeme con las instrucciones para la instalación de todos estos elementos.

- **Manual de usuario**

En este directorio se dispone, en formato PDF, del manual de usuario de una aplicación cualquiera generada con la línea de productos.

- **Aplicaciones de ejemplo**

Aquí se disponen las aplicaciones de ejemplo creadas en el desarrollo del proyecto mediante la línea de productos. Están en formato .cab listas para ser instaladas en cualquier dispositivo móvil compatible. A continuación, se listan los archivos .cab y sus correspondientes aplicaciones ejemplo:

Archivo CAB	Aplicación
Comunicador_basico.CAB	Ejemplo Básico
Comunicador_imgla.CAB	Imágenes y Letras Agrupadas
Comunicador_c.CAB	Ejemplo Colores
Comunicador_cf.CAB	Colores y Frases
Comunicador_cla.CAB	Categorías y Letras Agrupadas
Comunicador_cvm.CAB	Colores y Varios Métodos
Comunicador_f.CAB	Ejemplo Frases
Comunicador_lav.CAB	Letras Agrupadas y Vocales

- **Archivos utilizados**

En este directorio se disponen los archivos utilizados en la implementación de la línea de productos organizados en las siguientes categorías o subdirectorios:

- *SONIDOS*. Los archivos de sonido utilizados en las aplicaciones para reproducir el texto escrito.
- *IMÁGENES*. Los archivos de imagen utilizados en las aplicaciones que permiten la escritura mediante imágenes.
- *Paquetes*. Los distintos paquetes de la línea de productos. Contienen código fuente en C#, así como archivos XML y XSD, y ficheros de texto para almacenar las frases.

- **Diagramas empleados**

Contiene el archivo “componentes.uml”, creado con StarUML, en el cual se disponen los diagramas de casos de uso, los diagramas de clases y los diagramas de secuencia de la línea de productos desarrollada. Asimismo, contiene el archivo “caracteristicas.ai”, generado en Adobe Illustrator, que contiene el diagrama de características de la línea.

- **Otros**

En este directorio se encuentran los archivos “cabwiz.ddf” y “cabwiz.exe” que hay que reemplazar en el directorio de Visual Studio.Net 2005 para poder generar archivos CAB de las aplicaciones de la línea de productos que permitan la escritura mediante imágenes. Para una mejor comprensión, se incluye un archivo léeme explicativo.