

## **Parte I: Introducción.....7**

### **Capítulo 1: Presentación del proyecto..... 9**

- 1.1. Descripción del proyecto ..... 9
- 1.2. Alcance del proyecto ..... 10
- 1.3. Documentación presentada ..... 10

### **Capítulo 2: Objetivos propuestos..... 13**

- 2.1. Objetivos propuestos ..... 13

## **Parte II: Conceptos básicos .....17**

### **Capítulo 3: Contexto de la aplicación: La parálisis cerebral .... 19**

- 3.1. Parálisis Cerebral..... 19
  - 3.1.1. Definición ..... 19
  - 3.1.2. Tipos de parálisis cerebral..... 20
    - 3.1.2.1. Clasificación según el tipo .....20
    - 3.1.2.2. Clasificación según la topografía (según la parte del cuerpo afectada) ...21
    - 3.1.2.3. Clasificación según el tono.....21
    - 3.1.2.4. Clasificación según el grado de afectación .....21
  - 3.1.3. Causas de la Parálisis Cerebral. .... 22
  - 3.1.4. Síntomas de la Parálisis Cerebral ..... 23
- 3.2. Aplicación al proyecto ..... 24

### **Capítulo 4: Las nuevas tecnologías de la comunicación y los alumnos con déficit ..... 27**

- 4.1. Aprendizaje y mejora de las nuevas tecnologías para alumnos deficientes..... 27
- 4.2. Nuevas tecnologías para personas con deficiencia motora..... 27
- 4.3. Avances tecnológicos ..... 28
  - 4.3.1. Sistemas Alternativos y Aumentativos de Acceso a la Información.... 28
  - 4.3.2. Sistemas de Acceso. .... 29
  - 4.3.3. Sistemas Alternativos y Aumentativos de comunicación ..... 30
  - 4.3.4. Sistemas de Movilidad..... 31
  - 4.3.5. Sistemas de Control de Entornos ..... 31
- 4.4. Aplicación al proyecto ..... 31

### **Capítulo 5: Introducción a los dispositivos móviles. .... 33**

- 5.1. ¿Qué es un dispositivo móvil? ..... 33
- 5.2. Personal Digital Assistant o PDA ..... 34
  - 5.2.1. Elementos básicos de una PDA. .... 34
  - 5.2.2. Términos para referirse a los dispositivos móviles..... 36
- 5.3. Tipos de Personal Digital Assistant o P.D.A. .... 37

5.3.1.	Newton .....	37
5.3.2.	Palm .....	38
5.3.3.	Epoc-Symbian .....	39
5.3.4.	Windows CE .....	40

**Capítulo 6: Desarrollo de la aplicación para Pocket Pc ..... 41**

6.1.	Características del lenguaje Java .....	41
6.2.	Maquina Virtual Java (JVM) .....	43
6.3.	El Software Development Kit de Java .....	44
6.3.1.	Utilidades JDK .....	44
6.3.2.	Ediciones del SDK .....	45
6.3.3.	Nociones básicas de J2ME .....	46
6.3.3.1.	Máquinas Virtuales J2ME .....	47
6.3.3.2.	Configuraciones. ....	50
6.3.3.3.	Perfiles. ....	51
6.3.4.	MIDlet .....	53
6.3.4.1.	Herramientas de desarrollo .....	53
6.3.4.2.	Gestor de Aplicaciones .....	54

**Parte III: Máquinas Virtuales Java ..... 57**

**Capítulo 7: Maquinas Virtuales Java para Pocket Pc ..... 59**

7.1.	Introducción .....	59
7.2.	Máquinas virtuales Java. ....	59
7.2.1.	NSIcom CrEme 3.2.2 .....	60
7.2.2.	SavaJe OS 2.0 .....	61
7.2.3.	Sun Personal Java .....	61
7.2.4.	Blackdown J2RE (ARM port) .....	61
7.2.5.	Esemertec Jeode .....	62
7.2.6.	IBM WebSphere Studio .....	62
7.2.7.	Ewe VM .....	63
7.2.8.	SuperWaba VM .....	64
7.2.9.	Kada .....	64
7.2.10.	Kaffe .....	65
7.2.11.	Hewlett Packard ChaiVM .....	65
7.3.	Análisis de las Máquinas Virtuales Java .....	66
7.3.1.	Comparativa entre las Máquinas Virtuales SuperWaba y Ewe .....	66
7.4.	Superwaba .....	68
7.4.1.	Introducción .....	68
7.4.2.	Particularidades de SuperWaba .....	69
7.4.3.	Ventajas competitivas. ....	69
7.4.4.	Características básicas .....	70
7.4.5.	Plataformas soportadas .....	70
7.4.6.	Paquetes incluidos .....	71
7.4.7.	Características avanzadas .....	71
7.4.8.	Características de los datos .....	72
7.4.9.	Conectividad .....	72
7.4.10.	Requisitos mínimos. ....	72

## **Parte IV: Desarrollo de la aplicación ..... 75**

<b>Capítulo 8. Estudio previo.....</b>	<b>77</b>
8.1. Decisiones iniciales.....	77
8.2. Metodología empleada.....	77
<b>Capítulo 9: Análisis.....</b>	<b>79</b>
9.1. Descripción del problema.....	79
9.2. Definición de actores.....	80
9.3. Diagrama de Casos de Uso .....	81
9.4. Descripción de los casos de uso .....	82
9.4.1. Consultar Ayuda .....	82
9.4.2. Modificar Configuración .....	83
9.4.3. Escribir Texto.....	84
9.4.4. Insertar Frase .....	85
9.4.5. Borrar Frase.....	86
9.4.6. Guardar Frase .....	87
9.4.7. Reproducir Texto .....	88
9.4.8. Editar Texto .....	89
9.4.9. Cambiar Color.....	90
9.5. Modelo de objetos.....	91
9.5.1. Identificación de las clases.....	91
9.5.2. Diagrama inicial de clases .....	91
<b>Capítulo 10: Diseño .....</b>	<b>93</b>
10.1. Casos de Uso y Diagramas de secuencia.....	93
10.1.1. Consultar Ayuda .....	93
10.1.2. Modificar Configuración .....	95
10.1.3. Escribir Texto.....	97
10.1.4. Insertar Frase .....	98
10.1.5. Borrar Frase.....	100
10.1.6. Guardar Frase .....	102
10.1.7. Reproducir Texto .....	104
10.1.8. Editar Texto .....	105
10.1.9. Cambiar Color.....	107
10.2. Identificación de las clases.....	109
10.3. Diccionario de datos.....	110
<b>Capítulo 11: Implementación .....</b>	<b>115</b>
11.1. El entorno de programación y el lenguaje .....	115
11.2. Utilización de ficheros .....	115
11.3. Software usado .....	116
11.4. Hardware empleado.....	117
<b>Capítulo 12: Pruebas .....</b>	<b>119</b>
12.1. Pruebas realizadas. ....	119

12.2. Análisis de casos límite ..... 126

**Parte V: Manual de usuario ..... 131**

**Capítulo 13: Manual de Usuario ..... 133**

- 13.1. Descripción de la aplicación ..... 133
- 13.2. Instalación de la aplicación ..... 133
- 13.3. Pantalla de bienvenida ..... 134
- 13.4. Pantalla principal ..... 134
- 13.5. Métodos de escritura ..... 137
  - 13.5.1. Método de las letras agrupadas ..... 137
  - 13.5.2. Método de las vocales ..... 138
- 13.6. Pantalla números..... 140
- 13.7. Pantalla menú..... 140
- 13.8. Pantalla editar..... 143
- 13.9. Pantalla frases..... 144
- 13.10. Pantalla colores ..... 144
- 13.11. Pantalla configuración ..... 145
- 13.12. Pantalla ayuda..... 146

**Parte VI: Conclusiones..... 149**

**Capítulo 14: Conclusiones..... 151**

- 14.1. Objetivos alcanzados..... 151
- 14.2. Conclusiones de tipo técnico ..... 151
  - 14.2.1. Ventajas ..... 151
  - 14.2.2. Inconvenientes ..... 152

**Capítulo 15: Posibles mejoras..... 153**

**Apéndices..... 155**

**Apéndice A. Instalación de la Máquina Virtual SuperWaba..... 157**

- Instalación de la Máquina Virtual SuperWaba en el IDE de desarrollo..... 157
- Instalación de la Máquina Virtual SuperWaba en el Pocket PC ..... 158

**Apéndice B. Contenido del CD-ROM ..... 159**

**Apéndice C. Referencias..... 161**

- Bibliografía..... 161
- Fuentes Web..... 161





# **PARTE I:**

## **- Introducción -**





## **CAPITULO 1: PRESENTACIÓN DEL PROYECTO**

### **1.1. Descripción del proyecto**

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

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

El proyecto consiste en la implementación de un comunicador basado en PDAs que permita la comunicación a personas discapacitadas a través de la introducción de texto.

El comunicador se compondrá de varias pantallas con un número variable de botones cada una. Dispondrá de una pantalla principal en la que aparecerán botones de grupos de caracteres y un menú que permitirá realizar diversas acciones sobre el texto escrito. También dispondrá de pantallas secundarias en la que los caracteres se redistribuirán en botones de forma independiente. En ambas pantallas aparecerá un cuadro de texto en el que se mostrará en todo momento el texto escrito por el usuario. Éste podrá seleccionar el botón con la funcionalidad deseada mediante una pulsación directa sobre él. La pulsación de un botón de caracteres provocará la inserción de dicho carácter en el cuadro de texto.

La aplicación se desarrollará con la intención de que las personas con parálisis cerebral se incorporen lo mejor posible al entorno. En los últimos años, el dominio de Informática para discapacitados ha tenido un importante auge gracias a los avances tecnológicos en este campo, que permiten desarrollar aplicaciones para suplir cada vez una mayor cantidad de discapacidades, facilitando así la comunicación de dichas personas con su entorno.

La aplicación se desarrollara para un dispositivo móvil, en concreto para una PDA Pocket PC. Se podrá ejecutar en varias plataformas, debido a que el lenguaje de programación que usaremos es Java, y éste tiene la característica de ser multiplataforma.

El usuario de la aplicación va a poder aumentar, entre otras cosas, su independencia y su capacidad de comunicación, ya que el dispositivo móvil tiene un peso y tamaño reducido, no requiere un nivel de conocimientos elevado, etc.

## 1.2. Alcance del proyecto

La aplicación está pensada para personas con parálisis cerebral, o personas privadas de la capacidad del habla, pero que sin embargo hayan adquirido a lo largo de su vida conocimientos de lectura y escritura, puesto que la aplicación se basa en el lenguaje escrito. Además para aumentar su utilidad, cada una de las expresiones del usuario podrá ser reproducida, mediante un sintetizador de voz.

## 1.3. Documentación presentada

La memoria del proyecto se organiza en partes, y éstas a su vez se dividen en capítulos:

- **Parte I: Introducción.** En esta parte se hace una breve descripción del proyecto así como su alcance.
- **Parte II: Conceptos básicos.** En este punto se describen los conceptos básicos en los que se basa el proyecto. Éstos son:
  - La parálisis cerebral: definición de la enfermedad, así como los tipos, síntomas, causas, prevención y su tratamiento.
  - Las nuevas tecnologías de la comunicación y los alumnos con déficit: describe los avances tecnológicos de la informática en el ámbito de las personas discapacitadas.
  - Introducción a los dispositivos móviles: describe los tipos y características de estos dispositivos, centrándose en especial en la PDA Pocket PC.
  - Desarrollo de aplicaciones Java: En este capítulo se recorre de manera teórica los conceptos necesarios para llevar a cabo el desarrollo de una aplicación para este tipo de dispositivos.
  -
- **Parte III: Máquinas Virtuales Java.** Se trata de hacer un estudio de las diferentes Máquinas Virtuales Java existentes para Pocket Pc.
  - Maquinas virtuales Java para Pocket PC. En este capítulo se describen las distintas maquinas virtuales Java así como sus características
- **Parte IV: Desarrollo de la aplicación.** Se trata de determinar los puntos de partida del proyecto, para a partir de los mismos, abordar el análisis de la aplicación que vamos a construir.
  - Estudio previo: se realizará un estudio de la metodología que vamos a emplear para la realización del proyecto.
  - Análisis: se realizará un estudio sobre lo que debe hacer la aplicación, teniendo en cuenta las restricciones que se pueden presentar.

- Diseño: en esta fase se realizará el diseño de la aplicación basándonos en las técnicas de la Ingeniería del Software.
  - Implementación: se describirá el entorno de desarrollo así como el lenguaje de programación usado.
  - Pruebas: se presentarán las pruebas realizadas en la aplicación final.
- **Parte V: Manual de Usuario.** Describe el manejo de la aplicación.
  - **Parte VI: Conclusiones.** Estudio sobre los objetivos alcanzados y sobre las posibles mejoras del proyecto.
  - **Apéndices.**



## **CAPÍTULO 2: OBJETIVOS PROPUESTOS**

### **2.1. Objetivos propuestos**

Los objetivos principales que se pretenden obtener con la realización de este proyecto son los siguientes:

- Evaluar las posibilidades de Java como plataforma de implementación para dispositivos móviles.
- Conocer el entorno en el que se desarrollará la aplicación.
- Aplicar los conocimientos obtenidos en los objetivos anteriores para desarrollar un comunicador software que implemente un teclado que permita la introducción de texto por parte de los usuarios.

Como consecuencia del primer objetivo surgen los siguientes objetivos:

- Aprendizaje del lenguaje de programación Java para dispositivos móviles.
- Estudio de las diferentes Máquinas Virtuales Java para PDA.

Como consecuencia del segundo objetivo surgen los siguientes objetivos:

- Valorar y conocer a un colectivo de personas a las que las nuevas tecnologías pueden servir de gran ayuda en su día a día.
- Acercar la Tecnología de la Información a personas con grandes dificultades de comunicación.
- Conocer el entorno de trabajo para la realización de aplicaciones para dispositivos móviles.
- Ayudar a mejorar la calidad de vida de personas con discapacidades en muchos casos severas, así como a las personas de su entorno, debido a que los medios a su alcance normalmente son limitados.

Como consecuencia del tercer objetivo surgen los siguientes objetivos:

- Crear la interfaz del teclado lo más fácil de usar y lo más amigable posible de cara a su utilización por personas discapacitadas.
- Conseguir que el comunicador sea lo más configurable posible, pudiéndose adaptar a las necesidades de cada persona.
- Permitir que en el comunicador se puedan integrar otras aplicaciones creadas sobre el mismo dominio.
- Conseguir que la aplicación desarrollada esté destinada al uso de personas reales.







# **PARTE II:**

## **-Conceptos Básicos-**



## **CAPITULO 3: CONTEXTO DE LA APLICACIÓN: LA PARÁLISIS CEREBRAL**

### **3.1.Parálisis Cerebral**

#### **3.1.1. Definición**

La Parálisis Cerebral es la alteración de la postura y del movimiento que a veces se combina con alteraciones de las funciones superiores, producidas por una lesión no progresiva, a nivel del Sistema Nervioso Central. Esta lesión puede suceder durante la gestación, el parto o durante los primeros años de vida, y puede deberse a diferentes causas, como una infección intrauterina, malformaciones cerebrales, nacimiento prematuro, asistencia incorrecta en el parto, etc.

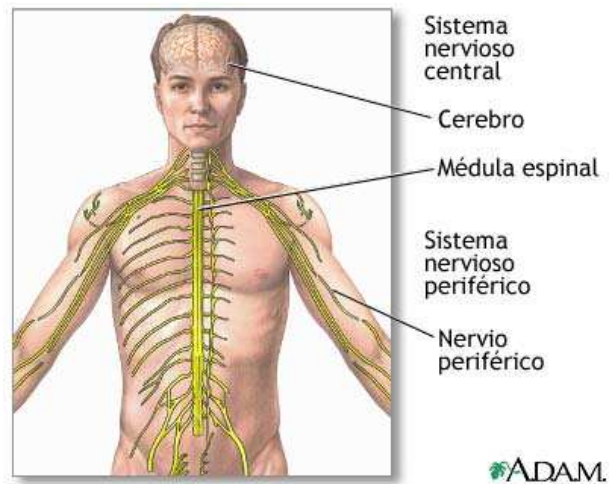


Figura 3.1. Sistema Nervioso Central

Las afectaciones más comunes son la perturbación del tono muscular, postura y movimiento, así como las interferencias producidas en el desarrollo neuropsíquico. A los problemas del movimiento se pueden asociar otros de diversa índole y no menos importantes. Se trata de problemas clínicos, sensoriales, perceptivos y de comunicación. Existe una enorme variedad de situaciones personales, no generalizables que dependen del tipo, localización, amplitud y difusión de la lesión neurológica. Así, en algunas personas la Parálisis Cerebral es apenas apreciable, mientras que otras pueden estar muy afectadas y necesitar de terceras personas para su vida diaria.

### 3.1.2. Tipos de parálisis cerebral

En la clasificación podemos encontrar 4 criterios diferentes, estos criterios son el tipo, la topografía, el tono y el grado.

#### 3.1.2.1. Clasificación según el tipo

- **Espástica:** Espasticidad significa rigidez; las personas que tienen esta clase de parálisis cerebral encuentran mucha dificultad para controlar algunos o todos sus músculos, que tienden a estirarse y debilitarse, y que a menudo son los que sostienen sus brazos, sus piernas o su cabeza.

La parálisis cerebral espástica se produce normalmente cuando las células nerviosas de la capa externa del cerebro o corteza, no funcionan correctamente.

Este es el grupo más grande; afecta a un 75% de las personas con parálisis cerebral.

- **Atetoide:** Las Personas que sufren este tipo de parálisis cerebral tienen unos músculos que cambian rápidamente de flojos a tensos. Sus brazos y sus piernas se mueven de una manera descontrolada, y puede ser difícil entenderles debido a que tienen dificultad para controlar su lengua, la respiración y las cuerdas vocales.

La Parálisis Cerebral atetoide, es el resultado de que la parte central del cerebro no funciona adecuadamente.

Afecta a menos de un 10% de las personas con Parálisis Cerebral.

- **Atáxica:** La Parálisis Cerebral atáxica hace que las personas que la padecen tengan dificultades para controlar el equilibrio, y si aprender a caminar lo harán de una manera bastante inestable.

También son propensos los afectados a tener movimientos en las manos y un hablar tembloroso.

La Parálisis Cerebral atáxica se produce porque el cerebelo, en la base del cerebro, no funciona bien.

Afecta a un 5 o un 10% de las personas con parálisis cerebral.

- **Mixta:** Casi un 10 % presentan un tipo mixto de parálisis cerebral, y un porcentaje reducido, es un tipo especial de tensión muscular como distonía, hipertonia, rigidez y temblores.

### 3.1.2.2. Clasificación según la topografía (según la parte del cuerpo afectada)

El sufijo plejia significa ausencia de movimiento, cuando hay algún tipo de movilidad se utiliza el sufijo paresia (cuadriparesias, tetraparesias, hemiparesias y monoparesias).

- **Cuadriplejía:** Están afectados los cuatro miembros.
- **Tetraplejía:** Afectación global incluyendo tronco y las cuatro extremidades, con un predominio de afectación en miembros superiores.
- **Triplejía:** Afectación de las extremidades inferiores y una superior.
- **Diplejía:** Afectación de las cuatro extremidades con predominio en extremidades inferiores.
- **Hemiplejía:** Está tomado un solo lado del cuerpo (hemicuerpo), y dentro de este el más afectado es el miembro superior.
- **Doble hemiplejía:** Cuando existe una afectación de las cuatro extremidades, pero mucho más evidente en un hemicuerpo, comportándose funcionalmente como una hemiparesia.
- **Paraplejía:** Son muy poco frecuentes, se afectan solo los miembros inferiores.
- **Monoplejía:** Se afecta un solo miembro (brazo o pierna), estos casos son poco comunes.

### 3.1.2.3. Clasificación según el tono

- **Isotónica:** tono normal.
- **Hipertónica:** tono incrementado.
- **Hipotónica:** tono disminuido.
- **Variable.**

### 3.1.2.4. Clasificación según el grado de afectación

- **Grave:** autonomía casi nula.
- **Moderada:** autonomía o a lo sumo necesita alguna ayuda asistente.
- **Leve:** autonomía total.

<b>CRITERIO</b>	<b>NIVELES</b>
<i>Tipo</i>	<ul style="list-style-type: none"><li>• Espástica</li><li>• Atetoide</li><li>• Atáxica</li><li>• Mixta</li></ul>
<i>Topografía</i>	<ul style="list-style-type: none"><li>• Cuadriplejía</li><li>• Tetraiplejía</li><li>• Triplejía</li><li>• Diplejía</li><li>• Hemiplejía</li><li>• Doble hemiplejía</li><li>• Monoplejía</li></ul>
<i>Tono</i>	<ul style="list-style-type: none"><li>• Isotónica</li><li>• Hipertónica</li><li>• Hipotónica</li><li>• Variable</li></ul>
<i>Grado</i>	<ul style="list-style-type: none"><li>• Grave</li><li>• Moderada</li><li>• Leve</li></ul>

Figura 3.2. Clasificación de la Parálisis Cerebral

### **3.1.3. Causas de la Parálisis Cerebral.**

Las causas se clasifican de acuerdo a la etapa en que ha ocurrido el daño a ese cerebro que se esta formando, creciendo y desarrollando. Se clasificarán como causas prenatales, perinatales o posnatales.

<b>Causas Prenatales</b>	<ol style="list-style-type: none"><li>1- Anoxia prenatal. (circulares al cuello, patologías placentarias o del cordón).</li><li>2- Hemorragia cerebral prenatal.</li><li>3- Infección prenatal. (toxoplasmosis, rubéola, etc.).</li><li>4- Factor Rh (incompatibilidad madre-feto).</li><li>5- Exposición a radiaciones.</li><li>6- Ingestión de drogas o tóxicos durante el embarazo.</li><li>7- Desnutrición materna (anemia).</li><li>8- Amenaza de aborto.</li><li>9- Tomar medicamentos contraindicados por el médico.</li><li>10- Madre añosa o demasiado joven.</li></ol>
<b>Causas Perinatales</b> <i>(Son las más conocidas y de mayor incidencia, afecta al 90 % de los casos.)</i>	<ol style="list-style-type: none"><li>1- Prematuridad.</li><li>2- Bajo peso al nacer.</li><li>3- Hipoxia perinatal.</li><li>4- Trauma físico directo durante el parto.</li><li>5- Mal uso y aplicación de instrumentos (fórceps).</li><li>6- Placenta previa o desprendimiento.</li><li>7- Parto prolongado y/o difícil.</li><li>8- Presentación pelviana con retención de cabeza.</li><li>9- Asfixia por circulares al cuello (anoxia).</li><li>10- Cianosis al nacer.</li><li>11- Broncoaspiración.</li></ol>
<b>Causas Posnatales</b>	<ol style="list-style-type: none"><li>1- Traumatismos craneales.</li><li>2- Infecciones (meningitis, meningoencefalitis, etc.).</li><li>3- Intoxicaciones (plomo, arsénico).</li><li>4- Accidentes vasculares.</li><li>5- Epilepsia.</li><li>6- Fiebres altas con convulsiones.</li><li>7- Accidentes por descargas eléctricas.</li><li>8- Encefalopatía por anoxia.</li></ol>

Figura 3.3. Causas de la Parálisis Cerebral

### 3.1.4. Síntomas de la Parálisis Cerebral

Con gran frecuencia, en la Parálisis Cerebral a los problemas del movimiento se asocian otros de diversa índole y no de menor importancia. Se trata de problemas clínicos, sensoriales, perceptivos y de comunicación.

- **Problemas Visuales:** El problema visual más común es el estrabismo que puede necesitar ser corregido con gafas, o en los casos más graves, con una operación.

Los problemas de ojos más serios son menos frecuentes. Algunos niños pueden tener un defecto cortical. Esto quiere decir, que la parte del cerebro que es responsable de la interpretación de las imágenes que el niño ve no funciona con normalidad. En pocos casos, el niño se puede quedar ciego pero en la mayoría de los casos los niños con este defecto sólo tienen dificultad para descifrar los mensajes que reciben desde sus ojos, por ejemplo, cuando aprenden a leer.

- **Percepción Espacial:** Algunos niños con Parálisis Cerebral no pueden percibir el espacio para relacionarlo con sus propios cuerpos (no pueden, por ejemplo calcular las distancias) o pensar espacialmente (como construir visualmente en tres dimensiones). Esto es debido, a una anomalía en una parte del cerebro, y no está relacionado con la inteligencia.
- **Problemas Auditivos:** Los niños que tienen Parálisis Cerebral atetóide son más propensos que otros a tener problemas auditivos graves, aunque no es el caso de los que padecen otro tipo de Parálisis Cerebral.

Sin embargo, estos niños pueden tener infecciones en el oído como cualquier otro niño, que pueden causar un ligero problema auditivo que le provocaría más dificultades para aprender a hablar.

- **Problemas de Comunicación:** La capacidad de comunicarse de un niño afectado por Parálisis Cerebral va a depender en primer lugar de su desarrollo intelectual, que hay que estimular desde el principio. Su capacidad de hablar también dependerá de la habilidad para controlar los pequeños músculos de la boca, la lengua, el paladar y la cavidad bucal.

Las dificultades para hablar que tienen los parálíticos cerebrales suelen ir unidas a las de tragar y masticar, aunque la intervención conjunta de diversos profesionales le ayudarán con estos problemas.

La mayoría de los niños afectados podrán ejercitar en alguna medida la comunicación verbal, mientras que otros podrán beneficiarse de la utilización de sistemas aumentativos de la comunicación, como por ejemplo, comunicadores tableros y otras ayudas tecnológicas.

- **Epilepsia:** La Epilepsia afecta a uno de cada tres niños con Parálisis Cerebral, pero es imposible predecir de qué manera o en qué momento el niño puede desarrollar los ataques. Algunos empiezan a padecerlos de pequeños y otros en edad adulta, pero se pueden controlar los ataques con medicación.
- **Otros problemas:** Algunos niños con Parálisis Cerebral pueden sufrir otros problemas como son: una cierta tendencia a congestionarse, a los constipados, sabañones, dificultades para controlar la temperatura corporal, para coger peso, alteraciones de la conducta y el comportamiento y problemas de sueño.

### 3.2. Aplicación al proyecto

La aplicación se realizará teniendo en cuenta los síntomas que presentan las personas afectadas con parálisis cerebral permitiéndoles una mejor comunicación con el entorno. Por ejemplo, para solventar los problemas visuales, como la pantalla de una PDA es de tamaño



reducido, ésta se aprovechará al máximo utilizando toda su superficie, de forma que los elementos sean lo más grandes posibles y evitando elementos innecesarios. Para solucionar los problemas de comunicación, la aplicación dispondrá de un reproductor de texto.



## **CAPITULO 4: LAS NUEVAS TECNOLOGIAS DE LA COMUNICACIÓN Y LOS ALUMNOS CON DEFICIT**

### **4.1. Aprendizaje y mejora de las nuevas tecnologías para alumnos con déficit.**

Para que los alumnos con parálisis cerebral puedan integrarse mejor en su entorno, sería conveniente introducir las nuevas tecnologías en su proceso de aprendizaje. Esto incluye:

- Que se beneficien de las posibilidades de los medios utilizados en un marco general.
- La necesidad de diseñar y producir medios específicos que puedan ser de ayuda y beneficio para las personas discapacitadas.

Se pretende que este colectivo esté a igualdad de condiciones que el resto de la sociedad, permitiéndoles una mayor integración social y laboral.

### **4.2. Nuevas tecnologías para personas con deficiencia motora**

Los problemas motóricos se pueden neutralizar, en la medida de lo posible, con diversos recursos como: fijar el lápiz a un soporte que, a su vez se unirá a la mano; fijar el papel a la mesa; fijar “perchas” a la silla para sostener el brazo; soportes en la cabeza o la boca, si no pueden usar las manos; máquinas de escribir con modificaciones ;utilización de programas específicos de lecto-escritura, como el Lápiz 4; utilización del ordenador y programas informáticos adecuados, etc.

Existen diversas máquinas y productos elaborados por la industria que podemos considerar como nuevas tecnologías y que pueden ayudar en el proceso rehabilitador de los alumnos con deficiencias motoras, entre ellos citaremos:

- **Aparatos POSUUM:** Permite accionar diferentes aparatos como interruptor de luz, T.V., sonidos, etc., por medio de la mano, el pie o la boca.

- **Microprocesador AUTOCOM:** Funciona como una máquina de escribir para cambiar de renglón, vuelta atrás, corrección, etc. El paciente selecciona mediante una pieza magnética la letra, palabra o frase que quiere emitir.
- **Tableros de letras, palabras, imágenes o símbolos:** Adaptados al nivel del desarrollo infantil. Hay tableros electrónicos que pueden ser accionados apretando un botón, succionando, manejando una palanca o con un movimiento de cabeza.
- **Máquinas de escribir:** Mucho mejor si son eléctricas por ser más silenciosas y permitir el borrado de errores más fácilmente.
- **Ordenadores:** Abren muchas posibilidades para estos niños con parálisis cerebral tanto en el terreno de la comunicación como en el aprendizaje y el juego.

### 4.3. Avances tecnológicos

La función de la tecnología es tratar de resolver distintas necesidades que tienen las personas con discapacidad, adecuando sus capacidades al entorno. Podemos agrupar la tecnología de ayuda en cinco grupos:

#### 4.3.1. Sistemas Alternativos y Aumentativos de Acceso a la Información.

Son ayudas para personas con discapacidad visual y/o auditiva. Entre ellas se pueden destacar:

- **Tecnologías del Habla:** El reconocimiento de voz y la conversión texto-voz ofrecen infinidad de posibilidades, algunas viables actualmente y otras en un futuro más o menos próximo. La expectativa en este campo es una disminución en todas sus restricciones, aunque no se espera que se logre reconocer el habla de forma plenamente satisfactoria.
- **Sistemas Multimedia Interactivos:** Los sistemas ofrecen la posibilidad de actuar sobre los contenidos de los mismos, surgiendo así la interactividad. Las personas con discapacidad se benefician de la existencia de servicios y aplicaciones multimedia que les permitirán, mediante las necesarias adaptaciones, perfeccionar el acceso multimodal en igualdad de condiciones.
- **Comunicaciones Avanzadas:** La conexión exclusiva a través de las computadoras va a dar paso a una amplia gama de dispositivos de acceso y todos los países podrán acceder a una red de alta velocidad. Hay una acelerada tendencia a incluir la videotelefonía, teléfonos de texto, fax y otros. Uno de los efectos que se prevén será la mayor integración social de personas que hasta ahora han tenido un acceso limitado a las telecomunicaciones, lo que directamente revertirá en disminuir las diferencias sociales entre la población con discapacidad.
- **Rehabilitación Cognitiva:** Existen programas sencillos de evaluación y rehabilitación para personas que presentan déficits o deterioros cognitivos, mediante interacción directa del usuario utilizando un sistema multimedia y una pantalla táctil, sin necesidad de utilizar teclado, trackball ni mouse, que permiten en algunos casos detener el deterioro cognitivo y recuperar algunas funciones cerebrales superiores.

### 4.3.2. Sistemas de Acceso.

Son interfaces adaptativas que permiten a las personas con discapacidad física o sensorial utilizar una computadora.

Algunos consisten en elementos microelectrónicos o telemáticos o ayudas técnicas no avanzadas pero necesarias que ayudan a compensar una deficiencia o discapacidad sustituyendo una función o potenciando los restos de la misma. Entre ellos mencionaremos:

- **Telelupas:** Son sistemas de magnificación de imágenes basados en circuitos cerrados de TV que posibilitan la lectura a personas con disminución visual.
- **Sintetizador Braille:** Es un pequeño computador personal con sintetizador de voz ó voz digitalizada que le permite a una persona invidente escribir información simulando a una máquina Perkins y verificar luego la misma.
- **Sistema de Reconocimiento óptico de caracteres:** Este dispositivo permite a una persona con discapacidad visual reproducir la información desde una computadora utilizando un scanner que lee cualquier texto mediante un programa OCR y los retransmite por medio de un sintetizador de voz o una línea Braille.
- **Teclado de conceptos:** consiste en una cuadrícula en blanco que se puede agrupar de acuerdo a varios conceptos temáticos asignados por los terapeutas.
- **Sobreteclados:** los hay de distintos tipos y se pueden adaptar a las necesidades del usuario.
- **Mouses:** Los hay tipo palancas, pedal, esférico (track ball), touch, etc. Entre ellos encontramos:
  - Emuladores de Ratón: Dispositivos inalámbricos que actúan mediante ondas infrarrojas.
  - Ratón Virtual: Pensado para facilitar el control del ratón a personas con discapacidad motora que no tengan control de la voz. Funciona por escaneo de un teclado virtual y se activa por un sonido detectado por un micrófono conectado a la tarjeta de sonido o un conmutador conectado a la misma tarjeta de sonido.
- **Pizarras electrónicas copiadoras:** Facilitan que las personas sordas e hipoacúsicas, o aquellas con dificultades motrices puedan obtener copias de clases presenciales sin perder la observación y atención de las mismas.
- **Pantallas táctiles:** Permiten que personas con dificultades motrices puedan acceder a los movimientos del cursor con la presión de un dedo o mano.
- **Interruptores:** Se pueden adaptar de cualquier tipo, bucales, oculares o de cejas, de muñeca, infrarrojos, fotoeléctricos, etc.
- **Navegadores.** Hay 2 tipos de navegadores:
  - Navegadores de acceso estándar: es el navegador habitual.
  - Navegadores de acceso alternativo: se caracterizan por presentar la información de las páginas Web de manera distinta a la convencional, normalmente en formato sólo texto o mediante audio y síntesis de voz.

### 4.3.3. Sistemas Alternativos y Aumentativos de comunicación

*"Los sistemas alternativos/aumentativos 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).*

Existen dos tipos: los sistemas con ayuda y los sistemas sin ayuda. Se clasifican dependiendo de si hacen uso de soportes o no, y la elección de un tipo de Sistema Alternativo y Aumentativo de Comunicación (SAAC) u otro se verá reflejada dependiendo de la necesidad de la persona.

- **Sistemas con ayuda:** Desarrollados para personas que por su discapacidad, no pueden acceder a un código verbal-oral de comunicación.

Son mecanismos que hacen uso de soportes externos a la persona para poder facilitarle la comunicación con su entorno. En general, estos sistemas incluyen pictogramas, ortografía o escritura en soportes individualizados.

Dentro de los sistemas con ayuda destacan:

- El sistema Bliss: es un sistema que ofrece un conjunto restringido de símbolos pictográficos que permiten representar infinidad de palabras e ideas.
- El sistema SPC (Sistema Pictográfico de Comunicación): es un sistema de comunicación no vocal basado en la representación de símbolos visuales de claro parecido con la realidad

- **Sistemas sin ayuda:** Son mecanismos mediante los cuales las personas con alguna deficiencia o carencia lingüística pueden mejorar su comunicación sin hacer uso de apoyos externos a la persona. Están estrechamente ligados a la educación y reeducación de las personas con sordera profunda prelocutiva con el objetivo principal de desarrollar la lengua oral y, posteriormente, la lengua escrita.

Dentro de los sistemas sin ayuda destacan:

- La lengua de señas: es una lengua de expresión y configuración gesto-espacial y percepción visual
- El sistema bimodal: es un sistema que usa signos simultáneamente al habla.
- El alfabeto dactilológico: es la representación manual del abecedario en el espacio
- La palabra complementada: es un sistema que hace posible percibir el habla completamente a través de la vista mediante el uso simultáneo de la lectura labial y una serie limitada de complementos manuales.

#### 4.3.4. Sistemas de Movilidad

Son aquellos relacionados a la movilidad personal y las barreras arquitectónicas. Una vez superadas estas dificultades se buscan las posibilidades para que las personas con graves discapacidades físicas puedan interactuar o utilizar una computadora. Ej: brazos o soportes articulados, conmutadores adosados a sillas de ruedas, emuladores de mouse, varillas, micro-robots, etc. que se combinan con sistemas alternativos y aumentativos de comunicación.

Existen dos casos de búsqueda alternativos:

- **Chip para parapléjicos.** Es un dispositivo que se implanta en los músculos y nervios y produce una electroestimulación que podría solucionar el problema de la parálisis
- **“Phantom” o dedo-robot para ciegos.** Es una combinación de escenarios de realidad virtual unidos a un dedo-robot que permite sentir el tacto de los objetos representados informáticamente.

#### 4.3.5. Sistemas de Control de Entornos

Permiten la manipulación de dispositivos que ayudan a controlar un entorno.

Se distinguen dos tipos:

- **Control Ambiental:** Interfaces que permiten a las personas con discapacidad motora, el poder controlar dispositivos de uso doméstico. Ej. “*casas inteligentes*”, cuyo software facilita: conectar/desconectar timbres, abrir/cerrar puertas, comunicarse por teléfono, control de luces/aire acondicionado/TV u otros dispositivos.
- **Realidad virtual:** utiliza nuevos dispositivos de entrada y salida avanzados, tales como guantes sensitivos, dispositivos de seguimiento de movimientos oculares, posicionadores de 3 D, etc. con alentadoras y crecientes posibilidades en el diseño de sistemas de asistencia a personas con discapacidad.

### 4.4. Aplicación al proyecto

Las personas con parálisis cerebral tienen dificultades para controlar los músculos del cuerpo. En especial, tienen problemas con el lenguaje, como puede ser una mala articulación de las palabras (problema conocido como disartria), dificultad de secuenciar y ejecutar los movimientos musculares necesarios para poder hablar (problema conocido como apraxia), lentitud en el habla, etc.

Debido a estos problemas y tras el estudio hecho anteriormente sobre los distintos avances tecnológicos, para la realización del proyecto nos basaremos principalmente en los Sistemas Alternativos y Aumentativos de Comunicación.





## **CAPITULO 5: INTRODUCCIÓN A LOS DISPOSITIVOS MÓVILES.**

### **5.1. ¿Qué es un dispositivo móvil?**

Los dispositivos móviles son aquellos suficientemente pequeños para ser transportados y usados durante su transporte. Normalmente se sincronizan con un sistema de sobremesa para actualizar aplicaciones y datos.

Las características básicas de este tipo de dispositivos son:

- Son aparatos pequeños.
- Tienen capacidad de procesamiento.
- Pueden conectarse a la red.
- Tienen memoria limitada.
- Están diseñados específicamente para una función, pero que pueden llevar a cabo otras más generales.
- Normalmente se asocian al uso individual de una persona, tanto en posesión como en operación, el cual puede adaptarlos a su gusto.
- La mayoría de estos aparatos pueden ser transportados en el bolsillo del propietario. Otros están integrados dentro de otros mayores, controlando su funcionalidad.

Algunas de las características que hacen que estos dispositivos sean diferentes de los ordenadores de sobremesa son los siguientes:

- Funcionalidad limitada.
- No necesariamente extensible y actualizable.
- Menos complicado en su manejo.
- Fácil de aprender su operación.
- No se requieren usuarios expertos.

Algunos de estos dispositivos son los siguientes:

- Paginadores.
- Comunicadores de bolsillo.

- Teléfonos con pantalla para Internet (Internet Screen Phones):
- Sistemas de navegación de automóviles.
- Sistemas de entretenimiento.
- Sistemas de revisión e Internet (WebTV).
- Teléfonos móviles.
- Organizadores y asistentes personales digitales (Personal Digital Assistant o PDA)

## 5.2. Personal Digital Assistant o PDA

Un PDA (Personal Digital Assistant o Ayudante personal digital) es un ordenador de mano originalmente diseñado como agenda electrónica. Hoy en día se puede usar como un ordenador doméstico (ver películas, crear documentos, navegar por Internet).

Un PDA puede funcionar como teléfono móvil, fax, explorador de Internet, organizador personal, GPS, etc.

La mayoría de PDAs empezaron a usarse con una especie de bolígrafo en lugar de teclado, por lo que incorporaban reconocimiento de escritura a mano. Hoy en día los PDAs pueden tener teclado y/o reconocimiento de escritura. Algunos PDAs pueden incluso reaccionar a la voz, mediante tecnologías de reconocimiento de voz.

Un PDA es parecido a un PC. Tienen una placa base, un procesador, memoria RAM, memoria permanente (ROM). También tienen sistema operativo en el cual se ejecutan los programas que deseemos (juegos, agendas, hojas de cálculo, navegadores Web...). La forma de comunicarse con el PDA es la propia pantalla, que en algunos modelos es táctil. El lápiz que incorpora el aparato se utiliza a modo de ratón. Para escribir podemos utilizar un teclado virtual que se muestra en la pantalla o bien hacer uso de la característica de reconocimiento de escritura.

### 5.2.1. Elementos básicos de una PDA.

Los elementos principales que forman una PDA son los siguientes:

- **Procesador:** Los ordenadores de bolsillo suelen incluir procesadores de arquitectura diferente a los que encontramos en nuestros ordenadores personales, ya que han de tener un consumo muy reducido y adecuarse a las características físicas de los PDA. Por ejemplo la PDA AcerN30, utilizada para la realización del proyecto, tiene un procesador Samsung 2410 de 266 Mhz, a diferencia de los PC actuales que pueden tener hasta 3,5Ghz .
- **Software:** La funcionalidad de los ordenadores de bolsillo sólo está limitada por la de las aplicaciones que en ellos se instalen. Lo habitual es que los ordenadores de esta clase incorporen de serie las aplicaciones más comunes: agenda de contactos, calendario, notas, gestor de correo electrónico, etc.

Más adelante, podremos añadir cualquier otro programa que adquiramos o descarguemos y que resuelva nuestras necesidades específicas.

- **Sistema operativo:** Los dos sistemas operativos más extendidos del mercado son Palm OS, de Palm Inc., y Windows Mobile, de Microsoft. Existen muchos otros pero la mejor opción sería elegir con un dispositivo que funcione con uno de estos dos, pues son los que tienen a su disposición un mayor número de actualizaciones y programas.
  - **Palm OS:** este sistema operativo se encuentra en los ordenadores comercializados por Palm y Handspring, entre otros. Es muy sencillo de utilizar.
  - **Windows Mobile:** la alternativa de Microsoft cuenta con una mayor difusión, y la variedad de aparatos basados en ella es más amplia.
- **Conectividad:** El ordenador de bolsillo intercambia datos con el exterior a través de los puertos de comunicación. Existen dos tipos: por cable e inalámbrico. Entre los inalámbricos, existen tres vías de conectividad, infrarrojos, Bluetooth y Wifi. Los infrarrojos presentan más limitaciones que los otros dos, como pueden ser la falta de rango y la necesidad de mantener una línea visual entre los dispositivos.

Actualmente, un mismo dispositivo puede incorporar dos o más sistemas simultáneamente.

- **Memoria:** La memoria de estos ordenadores es relativamente reducida. Algunos sólo tienen 2 o 4 MB, pero la mayoría se sitúa sobre los 16, 32 o 64 MB. Los modelos más recientes están equipados con hasta 256 MB, lo que aporta una gran comodidad a la hora de cargar varias aplicaciones, sobre todo las de multimedia (archivos MP3 o Internet).

Por otro lado, algunos dispositivos disponen de un espacio para que se pueda añadir tarjetas de memoria, aumentando así la capacidad del dispositivo.

- **Ranuras de expansión:** para los usuarios que demandan mayores capacidades en movilidad es importante que el PDA disponga de ranura de expansión para los diferentes modelos de tarjetas. Con ellas se solucionan las limitaciones de los dispositivos, en cuanto a tamaño.
- **Teclado o lápiz:** La mayoría de PDAs no tienen un verdadero teclado. Para introducir texto existen dos métodos: el primero, un pequeño teclado que aparece en la pantalla, cuyas letras se seleccionan con el lápiz incorporado. El segundo, un sistema de reconocimiento de letras, cifras y caracteres especiales (signos de puntuación, operaciones aritméticas, etc.). Además, la mayoría de estos ordenadores soporta la conexión de un pequeño teclado externo, que nos facilitará la entrada de datos.
- **Pantalla:** El tamaño de la pantalla es bastante reducido. La mayor parte de los modelos sin teclado presentan una definición comprendida entre 160x160 y 320x240 píxeles, mientras que los modelos con teclado son más ricos en resolución, que puede llegar hasta los 640x240 píxeles.

- **Batería:** Un ordenador de bolsillo puede funcionar gracias a su batería interna o a las pilas, que se pueden cambiar por baterías recargables. En algunas PDA, una pila botón asegura la conservación de los datos en la memoria RAM cuando la alimentación principal se agota.

Actualmente incorporan una batería de Litio-polímetro o Ion-litio, ya que tienen mayor duración, entre 8 y 12 horas.

- **Accesorios integrados:**
  - Altavoces, micrófono y toma de auriculares: no todos los modelos los llevan, pero la tendencia es que los vayan incorporando. Hace algunos años el micro se utilizaba para grabar datos que luego eran recuperados gracias al altavoz. Con la llegada del MP3, apareció la toma de auriculares.
  - Base de conexión: algunos ordenadores de bolsillo se entregan con el llamado Cradle, un dispositivo unido a la PDA, por lo que no es necesario enchufarlo cada vez que hay que hacer una sincronización o un intercambio de datos.
  - Lector biométrico: Algún modelo puede tener un sistema de seguridad para evitar el acceso a los datos contenidos en el ordenador por personas "no autorizadas". Básicamente es un pequeño lector de huellas dactilares que actúa de igual forma que la tradicional protección por contraseña.

### 5.2.2. Términos para referirse a los dispositivos móviles.

Término	Descripción
<i>PDA (Personal Digital Assistant)</i> <i>Asistente Personal</i> <i>Agenda Electrónica</i>	Nombre genérico para cualquier dispositivo portátil de reducido tamaño.
<i>Ordenador de mano</i>	Lo mismo que PDA con la diferencia que aquí se suponen capacidades avanzadas de programación. Quedan excluidas las agendas sin capacidad de programación, como las de gama baja.
<i>Handheld</i> <i>HPD</i>	PDA con teclado tipo QWERTY incorporado, de más peso y con más prestaciones que un PPC

<i>Palmhedl</i>	PDA sin teclado. PocketPc es el nombre de uno de los sistemas operativos, pero actualmente se usa también como sinónimo de ese tipo de dispositivos.
<i>Palm-size PC</i>	
<i>PPC</i>	
<i>Pocketpc</i>	
<i>Pocketpc</i>	PDA con S.O. Windows CE 3.0.  Actualmente se usa este nombre asimismo con nombre genérico para este tipo de dispositivos.

Figura 5.1. Términos para referirse a los dispositivos móviles.

### 5.3. Tipos de Personal Digital Assistant o P.D.A.

Hay muchos tipos diferentes, pero básicamente están divididos en 4 grandes familias.

#### 5.3.1. Newton



Uno de los primeros PDA fue el modelo Newton Message Pad (NMP) de Apple Computers, que apareció en el mercado en agosto de 1993. La Newton traía integrado un procesador ARM 610 de 20 MHz, tenía una capacidad máxima en RAM de 64 Kbytes, una resolución de 336 x 240 píxeles y con dimensiones de 19x11.4 centímetros. La Newton era muy grande, cara y complicada de operar, y su programa de reconocimiento de escritura tenía muchas limitaciones.

Figura 5.2. Macintosh Newton.

### 5.3.2. Palm

Una Palm es un dispositivo móvil, que contiene diferentes utilidades. Algunas de ellas vienen directamente de fábrica, como pueden ser, agenda, calculadora, libreta de direcciones y un anotador de ideas y otras pueden ser bajadas desde Internet.

Esta minicomputadora cuenta con diferentes elementos:

- Tamaño reducido
- Pantalla monocromática
- Mucha autonomía, semanas de uso continuado sin recargarlas.
- Microprocesador : capaz de resolver cálculos complejos
- Memoria: encargada de almacenar los datos.
- Lápiz: se utiliza para introducir los datos. Para meter letras se usa la parte inferior de la pantalla, y se escribía directamente las letras a mano alzada usando el método de escritura Graffiti
- Dispositivo infrarrojo: sirve para intercambiar información con otras Palms (sin ningún tipo de cables) o con otros dispositivos de este tipo, como impresoras, cámaras, etc.
- Cradle: conecta la Palm a cualquier PC, permite bajar los correos electrónicos contenidos en la computadora. Hay Palms que vienen preparadas para conectarse a Internet sin necesidad de módem, en las cuales la conexión se realiza mediante una antena (Palm VII). Además de brindarnos una vía de conexión con nuestras PC, los Cradles hacen una copia de seguridad de todo lo que tengas almacenado en la Palm, dentro de la PC
- Sistema operativo PALM OS, que hace un uso muy eficiente de las limitadas prestaciones y que va por la versión 6. Ha habido una gran evolución entre la versión 5 y la 6 para poder hacer un uso más eficiente de los nuevos procesadores.
- Versiones compatibles entre si.

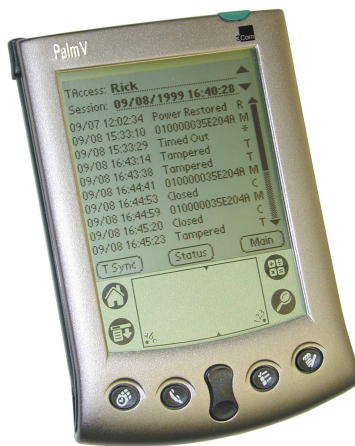


Figura 5.3. palm V



Figura 5.4. Palm tungsten

### 5.3.3. Epoc-Symbian

Este sistema operativo, el *EPOC*, fue inventado y desarrollado por la empresa inglesa *Psion*, y son típicos sus *handheld* con un auténtico teclado, pero con una pantalla monocromática que le permitía una gran autonomía. Las prestaciones eran mayores que las *PALM*, con muchísima mejor pantalla, y con un tamaño y peso muy reducido, y con mucha más autonomía que las *WINDOWS CE*. Hace poco hicieron una alianza entre unas cuantas empresas, como *Psion*, *Ericsson*, *Nokia*, y *Psion* dejó de fabricar sus propios *handhelds*, centrándose en crear el sistema operativo Symbian, tomando como base el S.O. *EPOC*.

Las características principales son:

- Necesita poco espacio
- Usa escasos recursos de memoria de forma dinámica
- Administra de forma eficiente la energía
- Soporta en tiempo real los protocolos de comunicación y telefonía
- Mas tolerante a fallos que los PCs
- Es actualizable



Figura 5.5. Psion Revo



Figura 5.6. Psion 5MX

Este sistema operativo es el que llevan los *nokia* de última generación, el *nokia 3650* y algún otro aparato.



Figura 5.7. Nokia 3650

### 5.3.4. Windows CE

Windows CE es una versión del sistema operativo Windows diseñado para pequeños dispositivos, como en el caso de los Handhelds. La interfaz gráfica del Windows CE es muy similar a la del Windows 95.

El Windows CE tiene sus propias APIs para desarrollo, y necesita sus propios drivers para el hardware con el cual va a interactuar. Windows CE no es un sinónimo de Windows XP en forma pequeña, incrustada o modular.

Las principales características son:

- Tecnologías sin cable (bluetooth, IrDA, 802.1x y EAP, IPv6, OBEX, MediaSense, RTC/SIP) que permiten al dispositivo conectarse a infraestructuras ya existentes
- Amplia gama de arquitecturas (ARM, MIPS, SHx, x86)
- Tiempo real
- Soporte para lenguajes de Internet y XML
- Tecnología de emulación
- Sencillo entorno de desarrollo del Kernel y de aplicaciones



## **CAPITULO 6: DESARROLLO DE APLICACIONES** **JAVA**

El lenguaje de programación Java fue desarrollado por Sun Microsystems en 1991, como parte de un proyecto de investigación para desarrollar software que pudiera ejecutarse en todo tipo de dispositivos.

### **6.1. Características del lenguaje Java**

Las características principales que nos ofrece Java respecto a cualquier otro lenguaje de programación, serían:

- **Simple:** Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos. C++ es un lenguaje que adolece de falta de seguridad, pero C y C++ son lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje.

Java elimina muchas de las características de otros lenguajes como C++, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el garbage collector (reciclador de memoria dinámica). No es necesario preocuparse de liberar memoria, el reciclador se encarga de ello y como es un thread de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que reduce la fragmentación de la memoria.

- **Orientado a objetos:** Java fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. La tendencia del futuro, a la que Java se suma, apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red.
- **Distribuido:** Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como *http* y *ftp*. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales.

Java en sí no es distribuido, sino que proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se difundan en varias máquinas, interactuando.

- **Interpretado y compilado a la vez:** Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador.

Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (run-time).

- **Robusto:** Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria.

También implementa los arrays auténticos, en vez de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria resultado de punteros que señalan a zonas equivocadas. Estas características reducen drásticamente el tiempo empleado en el desarrollo de aplicaciones Java.

- **Seguro:** Java está pensado para ser usado en red, por lo que se ha cuidado mucho la seguridad. En principio, es capaz de evitar que se rebase la pila del sistema en tiempo de ejecución, que corrompa la memoria externa a un proceso, o que se pueda acceder a ficheros locales de un ordenador que está ejecutando un applet en su navegador, por ejemplo.
- **Portable:** La indiferencia a la arquitectura representa sólo una parte de su portabilidad. Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas.
- **Multihilo:** Al ser MultiHilo, Java permite muchas actividades simultáneas en un programa. Los hilos son básicamente pequeños procesos o piezas independientes de un gran proceso. Al estar estos hilos construidos en el mismo lenguaje, son más fáciles de usar y más robustos que sus homólogos en C o C++.

El beneficio de ser multihilo consiste en un mejor rendimiento interactivo y mejor comportamiento en tiempo real. Aunque el comportamiento en tiempo real está limitado a las capacidades del sistema operativo subyacente (Unix, Windows, etc.) de la plataforma, aún supera a los entornos de flujo único de programa (single-threaded) tanto en facilidad de desarrollo como en rendimiento.

- **Indiferente a la arquitectura:** Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows Nt, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos

diversos. Para acomodar requisitos de ejecución tan variopintos, el compilador de Java genera bytecodes: un formato intermedio indiferente a la arquitectura, diseñado para transportar el código eficientemente a múltiples plataformas hardware y software. El resto de problemas los soluciona el intérprete de Java.

- **Dinámico:** El lenguaje Java y su sistema de ejecución en tiempo real son dinámicos en la fase de enlazado. Las clases sólo se enlazan a medida que son necesitadas. Se pueden enlazar nuevos módulos de código bajo demanda, procedente de fuentes muy variadas, incluso desde la Red.

## 6.2. Máquina Virtual Java (JVM)

Java es algo más que un lenguaje, ya que la palabra Java se refiere a dos cosas inseparables: el lenguaje que nos sirve para crear programas y la Máquina Virtual Java que sirve para ejecutarlos. Como vemos en la figura, el API de Java y la Máquina Virtual Java forman una capa intermedia (Plataforma Java) que aísla el programa Java de las especificidades del hardware.

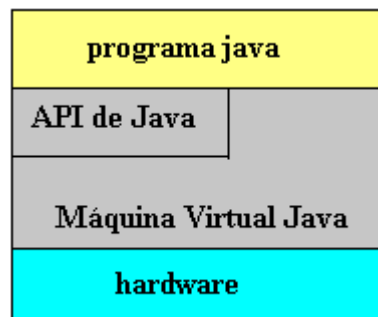


Figura 6.1. Plataforma Java

La Máquina Virtual Java (JVM) es el entorno en el que se ejecutan los programas Java, su misión principal es la de garantizar la portabilidad de las aplicaciones Java. Define esencialmente un ordenador abstracto y especifica las instrucciones (bytecodes) que este ordenador puede ejecutar. El intérprete Java específico ejecuta las instrucciones que se guardan en los archivos cuya extensión es .class. Las tareas principales de la JVM son las siguientes:

- Reservar espacio en memoria para los objetos creados
- Liberar la memoria no usada (garbage collection).
- Asignar variables a registros y pilas
- Llamar al sistema huésped para ciertas funciones, como los accesos a los dispositivos
- Vigilar el cumplimiento de las normas de seguridad de las aplicaciones Java

Esta última tarea, es una de las más importantes que realiza la JVM. Además, las propias especificaciones del lenguaje Java contribuyen extraordinariamente a este objetivo:

- Las referencias a arrays son verificadas en el momento de la ejecución del programa
- No hay manera de manipular de forma directa los punteros
- La JVM gestiona automáticamente el uso de la memoria, de modo que no queden huecos.
- No se permiten realizar ciertas conversiones (casting) entre distintos tipos de datos.

## 6.3. El Software Development Kit de Java

El JDK (Java Development Kit), también llamado SDK (Software Development Kit) es el entorno de base mínimo para efectuar desarrollos en Java, está compuesto por el conjunto de herramientas necesarias para compilar y ejecutar código escrito en Java.

### 6.3.1. Utilidades JDK

En la versión 2 de JDK podemos encontrar las siguientes utilidades:

- **javac:** El compilador de Java. Convierte el código fuente escrito en Java a bytecode.
- **java:** El intérprete de Java. Ejecuta en bytecode de la aplicación en Java, directamente a partir de los archivos class.
- **appletviewer:** Intérprete de Java que ejecuta las clases de los applet que se encuentran dentro de los archivos HTML.
- **avadoc:** Crea documentación HTML basándose en el código fuente de Java y en los comentarios que contiene.
- **jdb:** El analizador de Java. Permite recorrer línea a línea el programa, analizando las variables y los puntos de interrupción.
- **javah:** Genera archivos de cabecera en C que se pueden utilizar para construir rutinas que se encargarán de llamar a los métodos de Java o cancelar sus programas.
- **javap:** Desensamblador de Java. Muestra las funciones a las que puede acceder y, que se encuentran dentro de los archivos class compilados. También muestra el sentido del bytecode.
- **rmic:** Crea diversos archivos class que admiten el método de invocación remota (RMI).
- **rmiregistry:** Registro que se utiliza para acceder a los objetos RMI que se encuentran en un sistema determinado.
- **rmid:** Dominio de activación del sistema para el registro y activación de los objetos RMI.
- **serialver:** Utilidad para la serialización. Gracias a él se pueden hacer versiones con los objetos persistentes.
- **native2ascii:** Programa en serie que se utiliza para efectuar conversiones entre los caracteres de Latin-1 Unicode y otros sistemas de codificación internacional.
- **jar:** Generador de archivos JAR. Estos archivos pueden trabajar con varios recursos y clases de Java que se pueden distribuir a través de un único archivo comprimido.
- **keytool:** Se utiliza en las tareas de administración y generación de las claves de seguridad.
- **jarsigner:** Implementa una firma digital JAR en los archivos class. Permite que los applet se puedan certificar.
- **policytool:** Permite la instalación de una política de seguridad a nivel de seguridad.
- **tnameserv:** Servidor de nombres IDL de Java.

### 6.3.2. Ediciones del SDK

Sun, dispuesto a proporcionar las herramientas necesarias para cubrir las necesidades de todos los usuarios, creó distintas versiones de Java de acuerdo a las necesidades de cada uno. Según esto nos encontramos con que el paquete Java 2 lo podemos dividir en 3 ediciones distintas. J2SE (Java Standard Edition) orientada al desarrollo de aplicaciones independientes de la plataforma, J2EE (Java Enterprise Edition) orientada al entorno empresarial y J2ME (Java Micro Edition) orientada a dispositivos con capacidades restringidas. Veamos cuáles son las características de cada una de las versiones:

1. **Java 2 Platform, Standard Edition (J2SE):** Esta edición de Java es la que en cierta forma recoge la iniciativa original del lenguaje Java. Tiene las siguientes características:
  - Inspirado inicialmente en C++, pero con componentes de alto nivel, como soporte nativo de strings y recolector de basura.
  - Código independiente de la plataforma, precompilado a bytecodes intermedio y ejecutado en el cliente por una JVM (Java Virtual Machine).
  - Modelo de seguridad tipo sandbox proporcionado por la JVM.
  - Abstracción del sistema operativo subyacente mediante un juego completo de APIs de programación.

Esta versión de Java contiene el conjunto básico de herramientas usadas para desarrollar Java Applets, así como las APIs orientadas a la programación de aplicaciones de usuario final: Interfaz gráfica de usuario, multimedia, redes de comunicación, etc.

2. **Java 2 Platform, Enterprise Edition (J2EE):** Esta versión está orientada al entorno empresarial. El software empresarial tiene unas características propias marcadas: está pensado no para ser ejecutado en un equipo, sino para ejecutarse sobre una red de ordenadores de manera distribuida y remota mediante EJBs (Enterprise Java Beans). De hecho, el sistema se monta sobre varias unidades o aplicaciones. En muchos casos, además, el software empresarial requiere que se sea capaz de integrar datos provenientes de entornos heterogéneos. Esta edición está orientada especialmente al desarrollo de servicios Web, servicios de nombres, persistencia de objetos, XML, autenticación, APIs para la gestión de transacciones, etc. El cometido de esta especificación es ampliar la plataforma J2SE para dar soporte a los requisitos de las aplicaciones de empresa.
3. **Java 2 Platform, Micro Edition (J2ME):** Esta versión de Java está enfocada a la aplicación de la tecnología Java en dispositivos electrónicos con capacidades computacionales y gráficas muy reducidas, tales como teléfonos móviles, PDAs o electrodomésticos inteligentes. Esta edición tiene unos componentes básicos que la diferencian de las otras versiones, como el uso de una máquina virtual denominada KVM (Kilo Virtual Machine, debido a que requiere sólo unos pocos Kilobytes de memoria para funcionar) en vez del uso de la JVM clásica, inclusión de un pequeño y rápido recolector de basura y otras diferencias que ya iremos viendo más adelante.

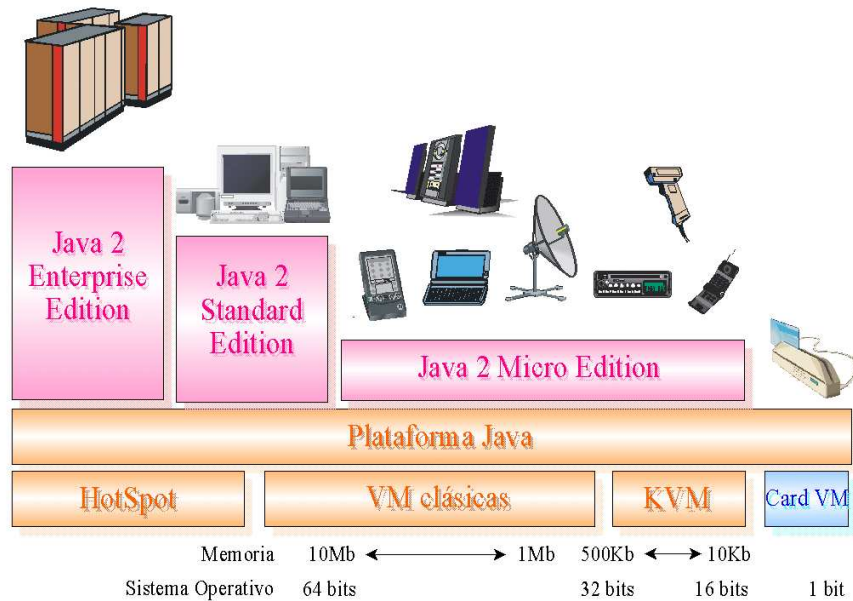


Figura 6.2. Arquitectura de la plataforma Java 2 de Sun

En la actualidad no es realista ver Java como un simple lenguaje de programación, si no como un conjunto de tecnologías que abarca a todos los ámbitos de la computación con dos elementos en común:

- El código fuente en lenguaje Java es compilado a código intermedio interpretado por una Java Virtual Machine (JVM), por lo que el código ya compilado es independiente de la plataforma.
- Todas las tecnologías comparten un conjunto más o menos amplio de APIs básicas del lenguaje, agrupadas principalmente en los paquetes `java.lang` y `java.io`.

J2ME representa una versión simplificada de J2SE. Sun separó estas dos versiones ya que J2ME estaba pensada para dispositivos con limitaciones de proceso y capacidad gráfica. También separó J2SE de J2EE porque este último exigía unas características muy pesadas o especializadas de E/S, trabajo en red, etc. Por tanto, separó ambos productos por razones de eficiencia. Hoy, J2EE es un superconjunto de J2SE pues contiene toda la funcionalidad de éste y más características, así como J2ME es un subconjunto de J2SE (excepto por el paquete `javax.microedition`) ya que contiene varias limitaciones con respecto a J2SE.

### 6.3.3. Nociones básicas de J2ME

Los componentes que forman parte de esta tecnología son los siguientes:

- Una serie de **máquinas virtuales Java** con diferentes requisitos, cada una para diferentes tipos de pequeños dispositivos.
- **Configuraciones**, que son un conjunto de clases básicas orientadas a conformar el corazón de las implementaciones para dispositivos de características específicas. Existen 2 configuraciones definidas en J2ME:
  - Connected Limited Device Configuration (CLDC) enfocada a dispositivos con restricciones de procesamiento y memoria
  - Connected Device Configuration (CDC) enfocada a dispositivos con más recursos.
- **Perfiles**, que son unas bibliotecas Java de clases específicas orientadas a implementar funcionalidades de más alto nivel para familias específicas de dispositivos.

Un entorno de ejecución determinado de J2ME se compone entonces de una selección de:

- a) Máquina virtual.
- b) Configuración.
- c) Perfil.
- d) Paquetes Opcionales.



Figura 6.3 Entorno de ejecución.

### 6.3.3.1. Máquinas Virtuales J2ME

Una máquina virtual de Java (JVM) es un programa encargado de interpretar código intermedio (bytecode) de los programas Java precompilados a código máquina ejecutable por la plataforma, efectuar las llamadas pertinentes al sistema operativo subyacente y observar las reglas de seguridad y corrección de código definidas para el lenguaje Java. De esta forma, la JVM proporciona al programa Java independencia de la plataforma con respecto al hardware y al sistema operativo subyacente. Las implementaciones tradicionales de JVM son, en general, muy pesadas en cuanto a memoria ocupada y requerimientos computacionales. J2ME define varias

JVMs de referencia adecuadas al ámbito de los dispositivos electrónicos que, en algunos casos, suprimen algunas características con el fin de obtener una implementación menos exigente.

Cada una de las configuraciones CLDC y CDC requieren su propia máquina virtual. La VM (Virtual Machine) de la configuración CLDC se denomina KVM y la de la configuración CDC se denomina CVM. Las características principales de cada una de ellas son:

- **KVM**

Se corresponde con la Máquina Virtual más pequeña desarrollada por Sun. Su nombre KVM proviene de Kilobyte (haciendo referencia a la baja ocupación de memoria). Se trata de una implementación de Máquina Virtual reducida y especialmente orientada a dispositivos con bajas capacidades computacionales y de memoria. La KVM está escrita en lenguaje C y fue diseñada para ser:

- Pequeña, con una carga de memoria entre los 40Kb y los 80 Kb, dependiendo de la plataforma y las opciones de compilación.
- Alta portabilidad.
- Modulable.
- Lo más completa y rápida posible y sin sacrificar características para las que fue diseñada.

Sin embargo, esta baja ocupación de memoria hace que posea algunas limitaciones con respecto a la clásica Java Virtual Machine (JVM):

- No hay soporte para tipos en coma flotante. No existen por tanto los tipos double ni float. Esta limitación está presente porque los dispositivos carecen del hardware necesario para estas operaciones.
- No existe soporte para JNI (Java Native Interface) debido a los recursos limitados de memoria.
- No existen cargadores de clases (class loaders) definidos por el usuario. Sólo existen los predefinidos.
- No se permiten los grupos de hilos o hilos daemon. Cuando queramos utilizar grupos de hilos utilizaremos los objetos Colección para almacenar cada hilo en el ámbito de la aplicación.
- No existe la finalización de instancias de clases. No existe el método `Object.finalize()`.
- No hay referencias débiles.
- Limitada capacidad para el manejo de excepciones debido a que el manejo de éstas depende en gran parte de las APIs de cada dispositivo por lo que son éstos los que controlan la mayoría de las excepciones.
- Reflexión.

El verificador de clases estándar de Java es demasiado grande para la KVM. Este verificador de clases es el encargado de rechazar las clases no válidas en tiempo de ejecución. Este mecanismo verifica los bytecodes de las clases Java realizando las siguientes comprobaciones:



- Ver que el código no sobrepase los límites de la pila de la VM.
- Comprobar que no se utilizan las variables locales antes de ser inicializadas.
- Comprobar que se respetan los campos, métodos y los modificadores de control de acceso a clases.

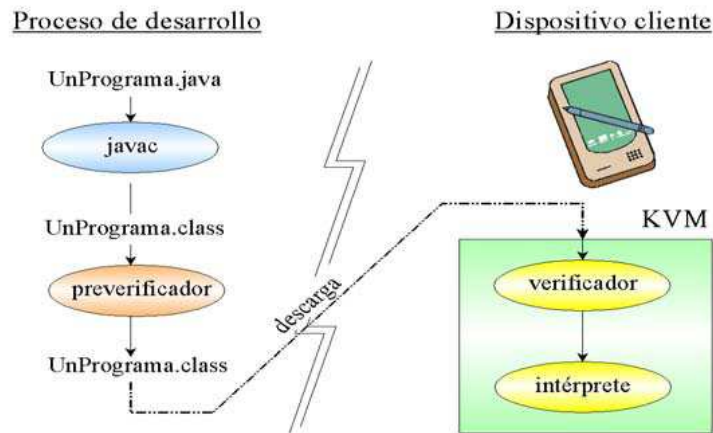


Figura 6.4. Preverificación de clases en CDLC/KVM

#### ▪ CVM

La CVM (Compact Virtual Machine) ha sido tomada como Máquina Virtual Java de referencia para la configuración CDC y soporta las mismas características que la Máquina Virtual de J2SE. Está orientada a dispositivos electrónicos con procesadores de 32 bits de gama alta y en torno a 2Mb o más de memoria RAM. Las características que presenta esta Máquina Virtual son:

- Sistema de memoria avanzado.
- Tiempo de espera bajo para el recolector de basura.
- Separación completa de la VM del sistema de memoria.
- Recolector de basura modularizado.
- Portabilidad.
- Rápida sincronización.
- Ejecución de las clases Java fuera de la memoria de sólo lectura (ROM).
- Soporte nativo de hilos.
- Baja ocupación en memoria de las clases.
- Proporciona soporte e interfaces para servicios en Sistemas Operativos de Tiempo Real.
- Conversión de hilos Java a hilos nativos.
- Soporte para todas las características de Java2 v1.3 y librerías de seguridad, referencias débiles, Interfaz Nativa de Java (JNI), invocación remota de métodos (RMI), Interfaz de depuración de la Máquina Virtual (JVMDI).

### 6.3.3.2. Configuraciones.

Una configuración es el conjunto mínimo de APIs Java que permiten desarrollar aplicaciones para un grupo de dispositivos. Éstas APIs describen las características básicas, comunes a todos los dispositivos:

- Características soportadas del lenguaje de programación Java.
- Características soportadas por la Máquina Virtual Java.
- Bibliotecas básicas de Java y APIs soportadas.

*Existen dos configuraciones en J2ME:*

- **Configuración de dispositivos con conexión, CDC** (Connected Limited Configuration)  
La CDC está orientada a dispositivos con cierta capacidad computacional y de memoria. Por ejemplo, decodificadores de televisión digital, televisores con internet, algunos electrodomésticos y sistemas de navegación en automóviles. CDC usa una Máquina Virtual Java similar en sus características a una de J2SE, pero con limitaciones en el apartado gráfico y de memoria del dispositivo. Ésta Máquina Virtual es la conocida como CVM (Compact Virtual Machine). La CDC está enfocada a dispositivos con las siguientes capacidades:
  - Procesador de 32 bits.
  - Disponer de 2 Mb o más de memoria total, incluyendo memoria RAM y ROM.
  - Poseer la funcionalidad completa de la Máquina Virtual Java2.
  - Conectividad a algún tipo de red.

La CDC está basada en J2SE v1.3 e incluye varios paquetes Java de la edición estándar. Las peculiaridades de la CDC están contenidas principalmente en el paquete `javax.microedition.io`, que incluye soporte para comunicaciones http y basadas en datagramas.

- **Configuración de dispositivos limitados con conexión, CLDC** (Connected Limited Device Configuration).

La CLDC está orientada a dispositivos dotados de conexión y con limitaciones en cuanto a capacidad gráfica, cómputo y memoria. Un ejemplo de éstos dispositivos son: teléfonos móviles, buscapersonas (pagers), PDAs, organizadores personales, etc.

La CLDC está orientada a dispositivos con ciertas restricciones, algunas de éstas vienen dadas por el uso de la KVM, necesaria al trabajar con la CLDC debido a su pequeño tamaño. Los dispositivos que usan CLDC deben cumplir los siguientes requisitos:

- Disponer entre 160 Kb y 512 Kb de memoria total disponible. Como mínimo se debe disponer de 128 Kb de memoria no volátil para la Máquina Virtual Java y las bibliotecas CLDC, y 32 Kb de memoria volátil para la Máquina Virtual en tiempo de ejecución.
- Procesador de 16 o 32 bits con al menos 25 Mhz de velocidad.

- Ofrecer bajo consumo, debido a que éstos dispositivos trabajan con suministro de energía limitado, normalmente baterías. Tener conexión a algún tipo de red, normalmente sin cable, con conexión intermitente y ancho de banda limitado (unos 9600 bps).

La CLDC aporta las siguientes funcionalidades a los dispositivos:

- Un subconjunto del lenguaje Java y todas las restricciones de su Máquina Virtual (KVM).
- Un subconjunto de las bibliotecas Java del núcleo.
- Soporte para E/S básica.
- Soporte para acceso a redes.
- Seguridad.

Un aspecto importante en CLCD es la seguridad. Esta configuración posee un modelo de seguridad sandbox al igual que ocurre con los applets.

Una determinada Configuración no se encarga del mantenimiento del ciclo de vida de la aplicación, interfaces de usuario o manejo de eventos, sino que estas responsabilidades caen en manos de los perfiles.

### 6.3.3.3. Perfiles.

Un perfil es un conjunto de APIs orientado a un ámbito de aplicación determinado. Los perfiles identifican un grupo de dispositivos por la funcionalidad que proporcionan (electrodomésticos, teléfonos móviles, etc.) y el tipo de aplicaciones que se ejecutarán en ellos. Las librerías de la interfaz gráfica son un componente muy importante en la definición de un perfil.

El perfil establece unas APIs que definen las características de un dispositivo, mientras que la configuración hace lo propio con una familia de ellos. Esto hace que a la hora de construir una aplicación se cuente tanto con las APIs del perfil como de la configuración. Un perfil siempre se construye sobre una configuración determinada. De este modo, podemos pensar en un perfil como un conjunto de APIs que dotan a una configuración de funcionalidad específica.

Al igual que en las máquinas virtuales existen varias configuraciones, cada una de éstas tiene un perfil.

*Para la configuración CDC tenemos los siguientes perfiles:*

- **Foundation Profile.** Este perfil define una serie de APIs orientadas a dispositivos que carecen de interfaz gráfica.
- **Personal Profile.** Es un subconjunto de la plataforma J2SE v1.3, y proporciona un entorno con un completo soporte gráfico AWT. El objetivo es el de dotar a la configuración CDC de una interfaz gráfica completa, con capacidades web y soporte de applets Java. Este perfil requiere una implementación del Foundation Profile.

- **RMI Profile.** Este perfil requiere una implementación del Foundation Profile ya que se construye encima de él. El perfil RMI soporta un subconjunto de las APIs J2SE v1.3 RMI. Algunas características de estas APIs se han eliminado del perfil RMI debido a las limitaciones de cómputo y memoria de los dispositivos.

*Para la configuración CLDC tenemos los siguientes perfiles:*

- **PDA Profile.** El PDA Profile está construido sobre CLDC. Pretende abarcar PDAs de gama baja, tipo Palm, con una pantalla y algún tipo de puntero (ratón o lápiz) y una resolución de al menos 20000 pixels con un factor 2:1.
- **Mobile Information Device Profile (MIDP).** MIDP fue el primer perfil definido para esta plataforma. Este perfil está orientado para dispositivos con las siguientes características:
  - Reducida capacidad computacional y de memoria.
  - Conectividad limitada (en torno a 9600 bps).
  - Capacidad gráfica muy reducida (mínimo un display de 96x54 pixels monocromo).
  - Entrada de datos alfanumérica reducida.
  - 128 Kb de memoria no volátil para componentes MIDP.
  - 8 Kb de memoria no volátil para datos persistentes de aplicaciones.
  - 32 Kb de memoria volátil en tiempo de ejecución para la pila Java.

Los tipos de dispositivos que se adaptan a estas características son: teléfonos móviles, buscapersonas (pagers) o PDAs de gama baja con conectividad. El perfil MIDP establece las capacidades del dispositivo, por lo tanto, especifica las APIs relacionadas con:

- La aplicación (semántica y control de la aplicación MIDP).
- Interfaz de usuario.
- Almacenamiento persistente.
- Trabajo en red.
- Temporizadores.

Las aplicaciones que realizamos utilizando MIDP reciben el nombre de MIDlets. Por tanto un MIDlet es una aplicación Java realizada con el perfil MIDP sobre la configuración CLDC.

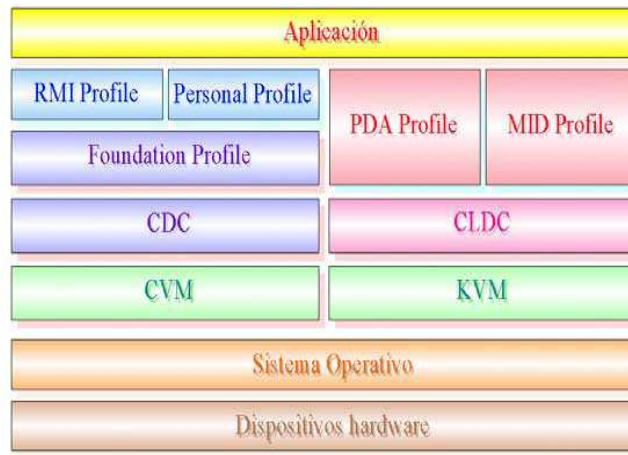


Figura 6.5 Arquitectura del entorno de ejecución de J2ME.

### 6.3.4. MIDlet

Los MIDlets son aplicaciones creadas usando la especificación MIDP. Están diseñados para ser ejecutados, como ya sabemos, en dispositivos con poca capacidad gráfica, de cómputo y de memoria. En estos dispositivos no disponemos de líneas de comandos donde poder ejecutar las aplicaciones que queramos, si no que reside en él un software que es el encargado de ejecutar los MIDlets y gestionar los recursos que éstos ocupan.

#### 6.3.4.1. Herramientas de desarrollo

Una aplicación J2ME está formada por un archivo JAR que es el que contiene a la aplicación en sí y un archivo JAD (Java Archive Descriptor) que contiene diversa información sobre la aplicación.

Para instalar las herramientas necesarias que usaremos para la construcción de los MIDlets, vamos a ver las etapas básicas que han de realizarse con este objetivo:

- **Desarrollo:** En esta fase vamos a escribir el código que conforma el MIDlet.
- **Compilación:** Se compilará la aplicación haciendo uso de un compilador J2SE.
- **Preverificación:** Antes de empaquetar el MIDlet es necesario realizar un proceso de preverificación de las clases Java. En esta fase se realiza un examen del código del MIDlet para ver que no viola ninguna restricción de seguridad de la plataforma J2ME.
- **Empaquetamiento:** En esta fase crearemos un archivo JAR que contiene los recursos que usa la aplicación, y crearemos también un archivo descriptor JAD.
- **Ejecución:** Para esta fase haremos uso de los emuladores que nos permitirán ejecutar el MIDlet.
- **Depuración:** Esta última fase nos permitirá depurar los fallos detectados en la fase anterior del MIDlet.

### **6.3.4.2. Gestor de Aplicaciones**

El gestor de aplicaciones o AMS (Application Management System) es el software encargado de gestionar los MIDlets. Este software reside en el dispositivo y es el que nos permite ejecutar, pausar o destruir nuestras aplicaciones J2ME.

El AMS realiza dos grandes funciones:

- Por un lado gestiona el ciclo de vida de los MIDlets.
- Por otro, es el encargado de controlar los estados por los que pasa el MIDlet mientras está en la memoria del dispositivo, es decir, en ejecución.







# **PARTE III:**

**- Máquinas -**

**- Virtuales Java -**



## **CAPÍTULO 7: MÁQUINAS VIRTUALES JAVA PARA POCKET PC**

### **7.1. Introducción**

El desarrollo del proyecto se llevará a cabo en lenguaje de programación Java, en concreto en la edición J2ME, ya que es la versión de Java orientada a dispositivos móviles.

Para poder ejecutar el “bytecode” de las clases Java es necesaria la instalación de una Máquina Virtual Java (JVM). A lo largo de este capítulo se realizará un estudio de las distintas Máquinas Virtuales Java existentes, en concreto para una PDA Pocket PC, ya que éste será el dispositivo utilizado para la realización del proyecto.

### **7.2. Máquinas virtuales Java.**

<b>NOMBRE</b>	<b>S.O. SOPORTADOS</b>	<b>VELOCIDAD</b>	<b>HARDWARE SOPORTADO</b>	<b>COSTE</b>
<b>NSIcom CrEme 3.2.2</b>	CE.net, PocketPC/2002	Rápido	PocketPC/ARM, Palm, HPC	Pago
<b>SavaJe OS 2.0</b>		Muy Rápido	PocketPC/ARM	Pago
<b>Sun Personal Java</b>	WinCE 2.11	Rápido	MIPS, SH3	Libre
<b>Blackdown J2RE (ARM port)</b>	Familiar Linux	Lento al principio, razonable durante la ejecución	iPAQ H36xx (o superior)	Libre
<b>Esemertec Jeode</b>	WinCE 2.11, PocketPC, Linux	Rápido	Dell Axim X5	Pago

<b>IBM WebSphere Studio</b>	WinCE 2.11, PocketPC	Rápido	PocketPC/ARM, WinCE, PalmOS, Windows, Wind River..	Pago
<b>Ewe VM</b>	Linux, PocketPC/2002, Windows, Palm, HPC Pro	Medio-Rápido	PocketPC/ARM, Sharp Zaurus, Palm, HPC, Linux, Win32	Libre
<b>SuperWaba VM</b>	PocketPC, Palm	Rápido	PocketPC/ARM, Palm	Libre
<b>Kada</b>	WinCE 2.11/3.0	Rápido	PocketPC/ARM	Pago
<b>Kaffe</b>	Familiar Linux		PocketPC/ARM	Libre
<b>jCompiler</b>	PocketPC		PocketPC/ARM	Pago
<b>Hewlett Packard ChaiVM</b>	WinCE 2.11, PocketPC	Lento	PocketPC/ARM	Libre

Figura 7.1. Maquinas Virtuales Java para Pocket PC

### 7.2.1. NSIcom CrEme 3.2.2

La maquina virtual Java CrEme es una buena opción para PDAs y otros productos basados en Windows-CE. Esta disponible para la mayor parte de plataformas Windows-CE y las versiones de evaluación están disponibles para las plataformas actuales. CrEme es fácil de instalar.

La maquina virtual CrEme 4 implementa la especificación J2ME CDC/Personal Profile 1.0. Soporta la librería Swing. La librería de gráficos esta basada en la implementación Truffle de AWT.

Los applets de Java podrán trabajar sin Internet Explorer, instalando el plugin CrEme. Los ordenadores personales soportan las versiones de Internet Explorer de la 4 a la 6. Los Pocket Pc y los dispositivos Windows Mobile soportan todas las versiones de Internet Explorer.

CrEme esta muy integrado con Windows CE y es fácil de configurar para aplicaciones particulares. Por ejemplo, las aplicaciones pueden controlar el dispositivo, pueden ejecutarse en modo pantalla completa, pueden cambiar la forma de ver el teclado, usar puertos de comunicación, el estado de la batería, etc. Todas estas características están disponibles en el paquete opcional de Java y en las extensiones Java-API (la biblioteca cremex.jar).

Algunas aplicaciones pueden requerir el acceso a las características que aún no soporta Java. Tal funcionalidad puede ser implementada en C o C++, como CrEme que soporta el estándar del Interfaz Nativo de Java.

Numerosas librerías de Java trabajan bien con CrEme, tales como la librería de gráficos SWT de Eclipse y varios paquetes de bases de datos.

Esta máquina virtual se puede obtener en <http://www.nsicom.com/>.

### 7.2.2. SavaJe OS 2.0

Ha sido desarrollada por Sun Microsystems' VM, pero ha sido optimizada para el tamaño, la velocidad y baja memoria. La JVM se ejecuta con el intérprete de bytecode optimizado, produciendo eficientemente código mientras conserva memoria.

El recolector de basura es simple, eficiente, y esta optimizado para Java.

La SavaJe OS JVM es un intérprete y no usa las técnicas del JIT (Just In Time). Necesita menos memoria que los sistemas basados en JIT. Las aplicaciones que interactúan con el usuario o la red tienen un orden de magnitud más rápido y con una memoria considerablemente más pequeña que en otras plataformas Java. Las tareas de computación intensivas escritas en Java pueden probarse más despacio que sobre una Máquina Virtual con un compilador JIT.

Esta máquina virtual se puede obtener en <http://www.mochaworks.com>

### 7.2.3. Sun Personal Java

Este producto ya no existe, pero la versión CE 2.11 está disponible en el EOL (End of Life) de Sun. Es rápido y está bien implementado, los componentes AWT están pensados para ordenadores de sobremesa, una consola útil para eliminar fallos etc., además de estar bien documentado y apoyado. El único problema que presenta es que esto no maneja la memoria y hilos muy bien, las opciones de línea de comando pueden ser usadas para pequeñas ayudas (para asignar más memoria), aunque muchos hilos simultáneos y estructuras de datos grandes pueden causar problemas.

Esta máquina virtual se puede obtener en <http://java.sun.com/products/personaljava/index.html>

### 7.2.4. Blackdown J2RE (ARM port)

Blackdown Java es un puerto de la máquina virtual Java de Sun para Linux creada por un grupo de voluntarios dirigidos por Karl Asha. Soporta plataformas que no tienen la versión oficial, incluyendo SPARC (Scalable Processor ARChitecture) y PowerPC. Actualmente, Blackdown Java soporta hasta la versión 1.4.2 de las bibliotecas de Java en tiempo de ejecución.

Blackdown proporciona un SDK (J2SDK, formalmente conocido como JDK) y un entorno de ejecución (J2RE) para Linux en varias arquitecturas (ix86, AMD-64, Sparc, PPC, S/390, ARM, m68k). El J2SDK y el J2RE contienen el Plug-In de Java para Netscape 4.x y Mozilla/Netscape 6.x.

Blackdown Java es un puerto de la máquina virtual Java de Sun para Linux creada por un grupo de voluntarios conducidos por Karl Asha. Soporta plataformas que no tienen la versión oficial, incluyendo SPARC y PowerPC. Actualmente, Blackdown Java soporta hasta la versión 1.4.2 de las bibliotecas de Java en tiempo de ejecución.

Blackdown no es ni Software Libre ni código abierto. La razón fue la falta de Java en Gentoo Linux y muchas otras distribuciones; esto es debido a que la política de redistribución binaria le permite ser preinstalada o distribuida por el sistema de dirección de paquete de la distribución, mientras que la política de redistribución binaria del Java Sun no lo hizo.

Desde Java 5, la Licencia de Distribuidor del Sistema Operativo para Java (DLJ) tiene algunos requisitos de distribución de Linux.

Esta máquina virtual se puede obtener en <http://www.blackdown.org/>

### **7.2.5. Esemertec Jeode**

El entorno de ejecución de Java *Jeode* viene implementado en las PDAs Sharp-Zaurus series SL-5XXX. Mediante la implementación de Java, una plataforma estándar y abierta, se pueden crear aplicaciones usando las herramientas de desarrollo existentes para Java.

El paquete AWT (de herramientas de ventanas abstractas) de Jeode ha sido implementado para manejar el sistema nativo de ventanas de cada plataforma.

Esmertec ha hecho mejoras significantes en el tiempo de arranque del motor de arranque de Jeode EVM por medio de la implementación de librerías de clases precargadas.

Esmertec, ofrece el primer entorno JVM en el mercado que provee soporte de plugins para correr applets de Java dentro del navegador de internet - Pocket Internet Explorer. Jeode además soporta otros navegadores populares para PDAs incluyendo Espial Escape, Netclue Clue.

Esta máquina virtual se puede obtener en <http://www.esmertec.com>

### **7.2.6. IBM WebSphere Studio**

WebSphere Studio proporciona un entorno de desarrollo integrado de próxima generación para la creación, la realización de pruebas, la integración y la implantación de aplicaciones J2EE y servicios Web. Basado en las innovaciones de Eclipse Version 2.0 y escrito según las especificaciones de la norma J2EE. Ayuda a optimizar y simplificar el desarrollo de aplicaciones J2EE con métodos recomendados, herramientas visuales, plantillas y generación de código.

WebSphere Studio proporciona una integración flexible, de tipo portal, para herramientas de desarrollo de aplicaciones que es compatible con múltiples lenguajes, plataformas y dispositivos, y que aprovecha al máximo la productividad de los programadores, incrementa el rendimiento de la inversión y acelera el proceso de comercialización. Disponible en varias configuraciones, con

extensiones de IBM y otros socios comerciales, WebSphere Studio está diseñado para cumplir los requisitos de programación específicos de las empresas en expansión, desde interfaces Web hasta aplicaciones de servidor, desde programación individual hasta entornos avanzados para equipos, desde programación en Java hasta integración de aplicaciones.

Esta máquina virtual se puede obtener en <http://www.ibm.com/websphere>

### 7.2.7. Ewe VM

La Máquina Virtual Ewe es una plataforma de desarrollo que permite escribir aplicaciones usando Java, pudiendo ejecutarse tanto en ordenadores personales como en sistemas móviles.

La máquina virtual Ewe está disponible para:

- PocketPC
- MS SmartPhone
- Casio BE-300
- HandHeldPC Pro
- Sharp Zaurus
- Linux Desktop
- Windows Desktop
- Any Java 1.2 VM

Las aplicaciones Ewe se pueden ejecutar en ordenadores sin la Máquina Virtual Ewe. La máquina virtual Java 1.2 (o superior) puede ejecutar aplicaciones Ewe. La aplicación será más portable que cualquier otra aplicación Java. El código puede ejecutarse como un Applet sin necesidad de hacer cambios.

Ewe tiene un uso eficiente de memoria en ordenadores personales y dispositivos móviles.

Las características que incluye para los dispositivos móviles son:

- Fácil programación del puerto de infrarrojos
- API de comunicación para la sincronización del ordenador con el dispositivo móvil.
- Interfaz de usuario personalizada para cada dispositivo.
- Una biblioteca GUI similar a SWING, pero con un tamaño mas reducido.
- Interfaz de Método Nativa para acceso personalizado y APIs específicos del dispositivo.

El SDK de Ewe se puede descargar y usar de forma gratuita. Contiene todo lo necesario para el desarrollo y despliegue de aplicaciones, incluyendo una utilidad para crear un programa de instalación de la aplicación.

El SDK dispone de un paquete para las aplicaciones ejecutables para las plataformas Windows y WindowsCE. Las aplicaciones pueden verse y ejecutarse como ficheros .exe.

En el SDK se incluye una guía para el desarrollo de aplicaciones Ewe.

Esta máquina virtual se puede obtener en <http://www.ewesoft.com>

### **7.2.8. SuperWaba VM**

Superwaba es un entorno de Programacion Orientado a Objetos que tiene como principal objetivo el de realizar aplicaciones para dispositivos móviles, tales como PDAs, Teléfonos Celulares, Tablet PCs, y muchos otros.

Soporta excepciones, threads, color y librerías Java, entre otras características.

Soporta las siguientes plataformas de ejecución: Palm OS 3.0 o superior, Windows CE 2.11 o superior, Pocket PC, .Net, Symbian 7.0, Windows 98 o superior y Linux.

Se presenta en dos ediciones: Una comunitaria con licencia GPL y otra profesional.

Esta máquina virtual se puede obtener en <http://www.superwaba.com.br>

### **7.2.9. Kada**

Kada Systems dispone de una máquina virtual para Pocket PC Windows 2000, Pocket PC Windows Mobile 2003 y Pocket PC 2002.

Kada es una máquina Virtual Java de pago, en la que hay que hacer una suscripción anual para poder acceder a la red de desarrollo móvil, para que los usuarios tengan acceso al centro de descargas y a la ayuda, entre otras cosas.

Kada proporciona un certificado J2ME para soluciones de software para desarrolladores de dispositivos móviles, proveedores y fabricantes y para la red de desarrollo móvil, con el objetivo de unir a esta comunidad para permitir discusiones en tiempo real sobre el desarrollo.

Kada ha desarrollado una Máquina Virtual Java optimizada y completa para Java 2 Micro Edition (J2ME) de Sun Microsystems Inc. A través del MDN (Mobile Developer Network), los desarrolladores tendrán acceso a la Plataforma Kada Móvil para J2ME (CLDC/MIDP) sobre entornos basados en ventanas y la plataforma Pocket PC 2002.

Actualmente está disponible el soporte para sistemas operativos móviles adicionales, como PalmOS y otras configuraciones J2ME y perfiles.

Esta máquina virtual ofrece soporte de MIDP 2.0 además de:



- Integración con JBuilder 9 Micro Edition
- Integración con JProbe Profiler
- Soporte configurable de CLDC 1.0 o CLDC 1.1
- Nuevos protocolos: Comm, Datagram, HTTP, HTTPS, Socket, Server Socket, SSL

Esta máquina virtual se puede obtener en <http://www.kadasystems.com>

### 7.2.10. Kaffe

Kaffe es una máquina virtual diseñada para ejecutar bytecode de Java (conjunto de instrucciones altamente optimizado).

Las principales características de esta Máquina Virtual son:

- Es software libre
- Posee un compilador JIT (Just in time), cuyo objetivo es traducir bytecode a código nativo del microprocesador de la arquitectura, con lo que se logra la ejecución a velocidades similares a las de los programas compilados clásicos. Este soporte ya está en Kaffe para los microprocesadores de las series Motorola 68000, Intel 386, Sparc y Alpha, plataformas en las que también existe Linux para ellas, por lo que la conjunción Linux y Kaffe JIT está presente en un gran número de arquitecturas.
- No posee una versión oficial de la licencia de la Máquina Virtual Java. No utiliza el código fuente de Sun por lo que kaffe no es similar a Java.
- Está constantemente en desarrollo y no es compatible con los lanzamientos actuales de Java. Carece de muchas características que posee la Máquina Virtual Java como son las relacionadas con la seguridad (no tiene el verificador completo de bytecode para ejecutar código).
- No es la mejor Máquina Virtual para desarrollar aplicaciones Java ya que carece de documentación, compatibilidad, soporte para eliminar errores, etc.
- Carece de la posibilidad de desarrollar entornos gráficos con AWT.

Esta máquina virtual se puede obtener en <http://www.kaffe.org>

### 7.2.11. Hewlett Packard ChaiVM

ChaiVM es una máquina virtual Java integrada en algunas impresoras Hewlett Packard. Soporta oficialmente las impresoras HP 9000 ,4100 y 4550, pero solo se ejecuta en HP 8150.

ChaiVM viene con servidor web, archivos estáticos y objetos, todo esto viene con el sistema de archivos de la impresora.

ChaiVM es bastante inestable.

Esta máquina virtual se puede obtener en <http://www.hp.com>

## 7.3. Análisis de las Máquinas Virtuales Java

A la hora de elegir la máquina virtual con la que trabajar para la realización del proyecto, hemos tenido en cuenta como primer factor el coste de éstas, descartando inicialmente aquellas que son de pago. Por tanto inicialmente hemos descartado las siguientes máquinas virtuales:

- NSIcom CrEme 3.2.2,
- SavaJe OS 2.0
- Esemertec Jeode
- Kada
- jCompiler
- IBM WebSphere Studio

El siguiente factor que hemos tenido en cuenta es el Sistema Operativo soportado por la máquina virtual. Como la realización del proyecto se llevará a cabo bajo un entorno Windows, de las máquinas virtuales restantes hemos descartado aquellas familiarizadas en un entorno Linux. Éstas son:

- Blackdown J2RE (ARM port)
- Kaffe

Otro factor a tener en cuenta es la velocidad de ejecución al utilizar la máquina virtual. En este caso, hemos descartado la máquina virtual Hewlett Packard ChaiVM debido a que tiene una velocidad lenta.

Hemos descartado la maquina virtual Sun Personal Java debido a que solo existe una versión (CE2.11) y es un producto que ya no se encuentra en desarrollo. Debido a esto hemos pensado que podríamos tener problemas a la hora de implementar nuestra aplicación.

### 7.3.1. Comparativa entre las Máquinas Virtuales SuperWaba y Ewe

Vamos a realizar un estudio comparativo entre las Máquinas Virtuales SuperWaba y Ewe, analizando las similitudes y diferencias de usar una plataforma frente a otra.

Dichas plataformas constan:

- **Librerías para programación:** Un API adaptado para éste tipo de dispositivos, similar en funcionalidad al de java y de una potencia mas cercana al de J2SE que al de J2ME.
- **Máquinas virtuales para diferentes arquitecturas:** Ambas plataformas compilan con el compilador *javac* (usando las librerías antes citadas) y como los bytecodes son un lenguaje estandar, el interprete *java* (así como muchos otros) pueden ejecutarlo sin problemas (una vez mas, usando las librería propias). Es por lo tanto posible ejecutar las aplicaciones de las siguientes 3 formas:

- En Máquina Virtual -> plataformas móviles, Windows y GNU/Linux
  - Como aplicación java -> plataformas sobremesa y portátiles (Windows, GNU/Linux, Mac....)
  - Como applet java -> navegadores web con MV java
- **Herramientas para el despliegue:** Software usado para empaquetar las aplicaciones para su uso en las máquinas virtuales sobre plataformas móviles. Generan un formato similar a un .jar (incluye las librerías, archivos de imagen y sonido y, en general, cualquier fichero al que se haga referencia dentro del código). También permiten crear instaladores para algunas plataformas.
  - **Documentación:** La propia de la API similar a la entregada con el JDK de Sun, así como documentación para desarrolladores.

Diferencias entre las Máquinas Virtuales SuperWaba y Ewe:

A continuación se presenta una tabla resumen con las principales características de ambos entornos:

	<b>SuperWaba</b>	<b>Ewe</b>
<b>Documentación API</b>	Si (buena)	Si (regular)
<b>Gratuito</b>	Si	Si
<b>Interfaz de usuario</b>	Normal (similar a AWT)	Normal (similar a SWING)
<b>Despliegue de aplicaciones</b>	Ficheros .pdb (similar jar)	Ficheros .ewe (similar jar)
<b>Utilidades para creación archivos de despliegue</b>	warp (.pdb) y exigen (lanzadores e instaladores)	Jewel Program Builder (aplicación gráfica)
<b>Creación de instaladores</b>	Si (Palm, WindowsCE/Pocket PC y Symbian)	Si (solo para Zaurus VM)
<b>Profundidades de color admitidas</b>	2,4,8 y 16 bits	hasta 32 bits (limitada por el dispositivo)

Figura 7.2. Comparación entre las Máquinas Virtuales SuperWaba y Ewe

Tras este estudio hemos observado que ambas plataformas presentan características similares, por lo que ha sido difícil la elección entre una u otra. Las diferencias más notables entre ambas plataformas son la documentación del API (mejor en la VM SuperWaba) y las profundidades de color que admiten (mejor en Ewe).

Hemos visto que existe un entorno de desarrollo específico para SuperWaba, que es Tauschke MobileCreator Personal, de fácil manejo y que permite instalar dicha maquina virtual de forma muy sencilla, simplemente especificando su ruta de acceso. Debido a esto, la Máquina Virtual elegida para el desarrollo del proyecto ha sido SuperWaba.

## 7.4. Superwaba

### 7.4.1. Introducción

Superwaba es una plataforma para la programación de dispositivos móviles, tales como PDAs, teléfonos celulares, Tablet PCs, etc. Define un lenguaje, una máquina virtual, un formato de ficheros .class y un conjunto de clases base.

El kit del desarrollo del software de SuperWaba (SWSDK) consiste en:

- **La máquina virtual de SuperWaba (SWVM)** que proporciona portabilidad para las aplicaciones de desarrollo.
- **Bibliotecas básicas y de la extensión** - proporciona funcionalidad para el desarrollo de aplicaciones comerciales.
- **Herramientas de la estructura y del despliegue** - construir e instalar aplicaciones de desarrollo en los dispositivos.
- **Ejemplos y documentación**

Es una tecnología reciente, fue creada a principios de 2000 por Hazan de Guilherme Campos, en Brasil.

Es código abierto optimizado para el desarrollo de aplicaciones que se ejecutan en Dispositivos Moviles ya sea bajo PalmOS o Windows CE. La sintaxis que utiliza es un subconjunto de JAVA, lo que permite que los desarrolladores que estén familiarizados con JAVA puedan comenzar rápidamente a utilizar SuperWaba. Sin embargo Superwaba no deriva de JAVA y no esta relacionado de ninguna manera con Sun Microsystems. SuperWaba se extiende de Waba y es compatible con ella; la diferencia entre estos dos lenguajes es que la API de Superwaba esta mas desarrollada en la parte de gráficos y accesos a hardware y mediante esto se pueden lograr mejores aplicaciones.

Existen varios IDE's(entornos de desarrollos) gratuitos que se pueden utilizar para trabajar, uno de ellos es el Proyecto Eclipse, que puede actualizarse con el plugin del IDE de SuperWaba. También esta el Tauschke MobileCreator Personal 1.74 y 1.8, y el Palm OS Developer Suite (utiliza Eclipse como IDE, e incluye todas las herramientas necesarias para el desarrollo de las aplicaciones como pueden ser: Palm Emulator, Palm Simulator, Palm OS debugger, y otras herramientas).

El lenguaje definido por SuperWaba, su máquina virtual y el formato de los ficheros clase han sido diseñados de forma óptima para su uso en PDA's. Las características de JAVA que

usaban mucha memoria o que eran innecesarias para los PDA's han sido omitidas en el diseño del lenguaje y su máquina virtual.

Los programas escritos en SuperWaba pueden ejecutarse en cualquier plataforma mediante el interprete JAVA (PDA's en general, Applets, Windows, Unix, MS-DOS, etc). Como el entorno y su API están basados en tecnología POO, todas las clases y métodos son reutilizables.

A medida que crece el mercado de los dispositivos móviles, mayor se hace la demanda de software en función de cualquier necesidad que se tenga. SuperWaba es una herramienta que simplifica mucho la tarea, en un "idioma" accesible, comprensible, y sobre todo portable.

La comunidad de programadores en SuperWaba esta creciendo muy rápido. Como hijo de C++ y de Java, se independiza de manera muy rápida, heredando muchos hábitos, pero obteniendo nuevos progresos de forma diferente y en dispositivos diferentes.

## 7.4.2. Particularidades de SuperWaba

- **Reducido tamaño:** SuperWaba ha sido diseñado para dispositivos pequeños. Su Máquina Virtual incluyendo las clases base, esta en el orden de los KiloBytes (270 Kb) en su última versión y necesita muy poca memoria para ejecutarse.
- **Funcionalidad:** SuperWaba permite desarrollar programas profesionales en casi todas las plataformas en un lenguaje muy extendido y fácil de usar. Además es orientado a objetos e incluye características de JAVA como el recolector de basura que agiliza el tiempo de desarrollo y permite la creación de aplicaciones más robustas.
- **Seguridad:** Los PDA's normalmente no poseen dispositivos de almacenamientos externos como diskettes o discos duros, lo que significa que si en un programa se corrompe esa memoria seguramente habrá que resetear el dispositivo, perdiéndose así todos los datos almacenados. SuperWaba protege los accesos a memoria para prevenir estos fallos. Además al tener un recolector de basura, son muy raros los gastos de memoria innecesaria comparados con otros programas desarrollados en otros lenguajes.
- **Multiplataforma:** Con SuperWaba, puedes escribir un programa que se ejecute en PalmOs, Windows CE, o cualquier otra plataforma que soporte el JAVA (JDK 1.02, 1.1 , 1.2 o 2.0 ). Las alternativas a programar en SuperWaba es programar en un lenguaje específico de la plataforma. Lo que significa escribir código C nativo y APIS específicas. Estas APIS suelen ser difíciles de programar y pequeños errores de programación suelen llevar a errores difíciles de encontrar y corregir.

## 7.4.3. Ventajas competitivas.

- **Bajo costo de mantenimiento:** SuperWaba proporciona aplicaciones más portables, ejecutándose en un gran número de dispositivos y de sistemas operativos sin modificaciones.

- **Coste total más bajo de propiedad (TCO):** la plataforma de SuperWaba es libre y no tiene licencia por cada dispositivo, aunque en realidad utiliza un modelo de suscripción anual de bajo coste. El servicio de la suscripción incluye servicios importantes, tales como actualizaciones de SWSDK y soporte de desarrollo que es ofrecido opcionalmente por otros vendedores. La suscripción profesional también incluye una base de datos para dispositivos móviles llamada PDBDriver sin coste adicional.
- **Mejor funcionamiento:** comparado con otras plataformas de desarrollo para dispositivos móviles, como la plataforma .NET (Microsoft) y J2ME (Sun Microsystems), las aplicaciones basadas en SuperWaba tienen un buen funcionamiento y SWVM tiene un bajo uso de memoria.
- **Fácil aprendizaje:** las bibliotecas son fáciles de utilizar, centradas en el desarrollo de aplicaciones comerciales, que permiten hacer prototipos rápidamente.
- **Plataforma abierta:** SuperWaba es una plataforma abierta, que tiene acceso al código fuente y a la libre distribución del software incluido en SWSDK. Estas características permiten la personalización para necesidades específicas, garantiza la calidad y evolución constante a través del trabajo de colaboración de la comunidad de usuarios.

#### 7.4.4. Características básicas

- Cualquier Java IDE se puede utilizar para desarrollar aplicaciones basadas en SuperWaba.
- Tiene un gran número de interfaces de usuarios y funcionalidades como son: editbox, combobox, buttons, Grid, menus, sound (wav/midi), graphics, listbox, label, event support, timer, date picker, keyboard, calculator, scrollbars, pop-up windows y un formulario exclusivo que hace que la aplicación funcione en cualquier resolución.
- Funcionamiento gráfico: la implementación SDL de SuperWaba (Simple Direct Media Layer) proporciona mucho poder para el procesamiento gráfico, añadiendo muchas características gráficas a la plataforma.
- Mejor uso de la memoria en los dispositivos Palm OS (permite la asignación de memoria independientemente de los dispositivos).

#### 7.4.5. Plataformas soportadas

- Palm OS 3.0 o superior
- Windows CE 2.11 o superior, Pocket PC, .Net
- Symbian 7.0 y 7.0s
- Windows 98 o superior (Las aplicaciones basadas en SuperWaba se ejecutan en ordenadores de sobremesa sin Java Runtime Environment)
- Linux

### 7.4.6. Paquetes incluidos

- **waba.lang (java.lang):** este paquete contiene algunas clases del java.lang original, y estas clases contienen un subconjunto de los métodos. Este paquete tiene entre otras las siguientes clases: Object, Class, String, StringBuffer, Math, y algunas derivadas de Exception, incluida ella.
- **waba.fx:** Este paquete contiene varias clases usadas para manipular gráficos y sonidos. Este paquete tiene, entre otras, las siguientes clases: Color, Coord, Font, Image, Palette, Rect, Sound, AudioClip y Graphics. Hay que resaltar que la clase Graphics es una de las más importantes de este paquete, ya que en ella encontramos todos los métodos usados para dibujar, como: drawRect, drawLine, drawCircle, drawEllipse.
- **waba.io:** este paquete tiene todas las clases usadas para el acceso de entrada y salida de dispositivos, ya sean Socket, puertos serie, ficheros, etc. Este paquete tiene, entre otras, las siguientes clases: ByteArrayOutputStream, Catalog (representa un archivo de la base de datos de PalmOS - fue usado también para Windows CE/Pocket PC permitiendo una gran portabilidad), DataStream, File (funciona para sistema de archivos en WinCE, y también para Palm, que utilizan tarjetas de memoria), Socket, SerialPort e ResizeStream.
- **waba.sys:** este paquete contiene clases que hacen referencia al Sistema Operativo y a la máquina virtual en los cuales el programa se está ejecutando, así como las clases de conversión de tipos. Este paquete tiene, entre otras, las siguientes clases: Vm, Thread, Settings, Convert (muy importante y útil, ya que contiene métodos para la conversión entre tipos primitivos y Strings. Esta clase permite solucionar muchos problemas.), CharacterConverter (usada para la conversión de Strings a bytes) y UTF8CharacterConverter.
- **waba.ui:** Este es el paquete más importante de SuperWaba, pues dispone de todas las clases de componentes que hacen falta para crear una interfaz de usuario. Este paquete tiene, entre otras, las siguientes clases: Button, Check, ComboBox, Edit (similar a TextField de J2SE), Event (esta clase se usa para el tratamiento de eventos de las aplicaciones; SuperWaba usa el mismo modelo de eventos que Java 1.0), Label, ListBox, MenuBar, PopupMenu, Radio, ScrollBar, TabPanel e MainWindow (esta es la clase que define la ventana principal de la aplicación, que es la clase que será llamada por defecto al inicio de la aplicación. Una clase que hereda de ésta, normalmente tendrá el mismo nombre de la aplicación).
- **waba.util:** las clases contenidas en este paquete son clases que tienen la misma utilidad que las clases de java.util. Este paquete tiene, entre otras, las siguientes clases: Date, Hashtable, IntHashtable, IntVector, Random e Vector.

### 7.4.7. Características avanzadas

- Biblioteca GPS, que procesa la información global.
- Personal Information Management Abstract Layer (PIMAL), permite acceso a la plataforma, al libro de direcciones, a la lista y a la información del calendario.
- Biblioteca para el desarrollo de juegos.
- Soporte Symbol® Barcode Scanners (lector de código de barras) para Pocket PC y Palm OS.

- Biblioteca Cryptography, que implementando los algoritmos siguientes: Blowfish, MD5, SHA1, TEA.
- Permite la creación de bibliotecas (escritas en lenguaje-c) que pueden ser integradas en aplicaciones basadas en SuperWaba.
- Bibliotecas Zlib para el Manejo de ficheros comprimidos
- Soporte para los múltiples formatos gráficos incluyendo BMP, el JPEG, el png y el GIF
- Incluye una utilidad que convierte las fuentes de TrueType a las fuentes de la multi-plataforma soportadas por SuperWaba.
- Soporte para el Manejo de archivos HTML.
- La clase SWUnit, permite generar los casos de prueba para ser ejecutados en los dispositivos.

#### **7.4.8. Características de los datos**

- La clase File para el manejo de archivos comunes, incluye soporte para las tarjetas Secure Digital y Memory Stick.
- El API para Bases de Datos (SWDBC), tiene dos implementaciones disponibles:
  - IBM Db2e driver, tiene integrada la base de datos Db2.
  - PDBdriver 3.0, que permite el manejo de ficheros PDB con los comandos básicos de SQL como son SELECT, INSERT, DELETE, UPDATE y otras funcionalidades de SQL como las funciones agregadas, que son (avg, max, min, sum, group by, order by);
- La clase Catalog para el manejo de archivos PDB (Palm Database) en todas las plataformas

#### **7.4.9. Conectividad.**

- Soporte Socket para la comunicación TCP/IP, infrarrojos, puertos serie, USB y Bluetooth
- Biblioteca Printing
- Soporte para protocolos HTTP y FTP.
- Clase para el envío de Email (SMTP)
- Soporte para servicios Web.

#### **7.4.10.Requisitos mínimos.**

- El SWSDK se puede instalar en Windows 98 /ME/NT/2000/XP.
- Pentium II o equivalente, con 128Mb de memoria disponible
- Mínimo de 200Mb de espacio libre en el disco duro.







## **PARTE IV:**

### **- Desarrollo de la - - aplicación -**



## **CAPITULO 8. ESTUDIO PREVIO**

### **8.1. Decisiones iniciales**

El objetivo de este proyecto es la implementación de una aplicación para PDAs que permita la comunicación de personas discapacitadas a partir de la introducción de texto, mediante la pulsación de botones. El comunicador dispondrá de una serie de frases que el usuario puede utilizar, pudiendo eliminar o añadir nuevas en función de sus necesidades. Permitirá reproducir el texto escrito, así como editarlo. También se podrá modificar la configuración y los colores del comunicador.

Se pretende que la herramienta sea práctica, fácil de manejar y que permita a las personas que posean alguna discapacidad una mejor comunicación con el entorno.

Se ha realizado un estudio de los síntomas que presentan las personas que poseen alguna discapacidad (estudio realizado en el capítulo 3) intentando que el comunicador se adapte lo mejor posible a ellas. Por tanto, la aplicación se realizará de forma que ocupe todo el tamaño de la pantalla de la PDA, ya que ésta es de tamaño reducido y las personas con parálisis cerebral pueden tener problemas de visión.

Se ha realizado un estudio de cómo afectan las nuevas tecnologías a las personas con discapacidades (estudio realizado en el capítulo 4), observando que existen diferentes medios para ayudar a estas personas, de los cuales usaremos los Sistemas Alternativos y Aumentativos. Por tanto la aplicación contará con un reproductor de voz, facilitando la comunicación a aquellas personas que tengan dificultades con la lectura.

### **8.2. Metodología empleada**

El lenguaje utilizado para la realización de este proyecto es Java (plataforma Java SDK), por tanto el desarrollo del proyecto se basará en la Orientación a Objetos.

Inicialmente se determinaran los puntos de partida del proyecto, para realizar un análisis de las necesidades de la aplicación que queremos construir. Posteriormente se implementará la aplicación, según las especificaciones.

Para realizar el análisis del sistema que queremos desarrollar, usaremos el Lenguaje Unificado de Modelado (UML) que es el lenguaje de modelado de sistemas de software más utilizado actualmente.

El modelo seguido para realizar el proyecto es un *modelo en cascada*, ya que las etapas seguidas para la creación de la aplicación han sido: análisis, diseño, implementación y pruebas (como se muestra en la figura 8.1.), de forma que no hemos iniciado ninguna etapa hasta que no haya terminado la anterior.

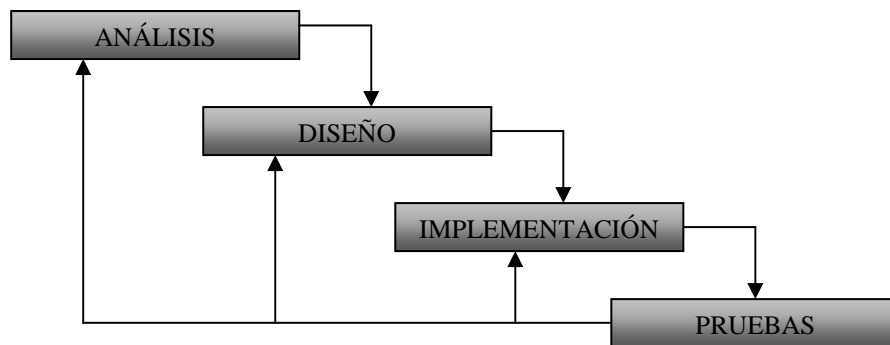


Figura 8.1. Modelo en Cascada

Debido a que la aplicación está orientada a la comunicación de personas con parálisis cerebral, tenemos que tener en cuenta los requisitos del usuario, y orientar el diseño a sus especificaciones.

Se pretenderá que las componentes software que integren la aplicación sean lo más independientes posibles, para poder ser reutilizadas posteriormente y permitir un mejor mantenimiento.

Una vez realizada la aplicación se realizarán una serie de pruebas para poder descubrir y depurar los posibles errores que puedan existir. Para ello probaremos cada uno de los elementos de la aplicación y analizaremos los casos límite, teniendo en cuenta que es donde más errores se suelen producir.

## CAPITULO 9: ANALISIS

El análisis consiste en la comprensión y modelado de la aplicación y del dominio en el cual funciona. La entrada inicial de la fase de análisis es una descripción del problema que hay que resolver y proporciona una visión general conceptual del sistema propuesto. La salida del análisis es un modelo formal que captura los tres aspectos esenciales del sistema: los objetos y sus relaciones, el flujo dinámico de control y la transformación funcional de datos que están sometidos a restricciones.

### 9.1. Descripción del problema

El objetivo del proyecto es desarrollar un componente software que implemente un teclado configurable, es una aplicación que permite al usuario introducir texto mediante la selección de botones.

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

Los botones serán los que permitirán la interacción entre el usuario y el teclado. Éstos serán de caracteres y pueden ser de dos tipos:

- Botones que lleven escrita su funcionalidad asociada (por ejemplo los siguientes botones: MENU, EDITAR, VOLVER, etc)
- Botones que contengan el texto a escribir en el cuadro de texto al ser pulsados (por ejemplo los siguientes botones: A, 1, ?, etc.)

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 carácter 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 hasta el momento 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.
- **Salir de la aplicación:** La pulsación de un botón con esta funcionalidad hará que finalice la aplicación.

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. En este proyecto se desarrollarán dos configuraciones:

- **Método de letras agrupadas:** Deberá existir una pantalla inicial en la que se mostrará un conjunto de botones representado, cada uno de ellos, por un conjunto de caracteres (por ejemplo: AÁBCDE, ÉFGHIÍ, etc.). La selección de uno de estos botones llevará a otra pantalla en la que aparecerán los caracteres del botón pulsado redistribuidos de forma independiente en botones (por ejemplo: si pulsamos el botón AÁBCDE, aparecerá una nueva pantalla con seis botones que contendrán A, Á, B, C, D, E). La selección de alguno de estos botones 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 representa de forma independiente en la pantalla principal, lo que permite agilizar la escritura reduciendo el número de pulsaciones.

## 9.2. Definición de actores

Los actores son personas o entidades externas que intercambian información con el sistema.

En este sistema solo existe un único actor que será el usuario de la aplicación. Éste tiene acceso a todas las funciones que podrá manejar de forma autónoma e independiente, de manera que podrá conseguir una comunicación más natural que con los antiguos sistemas.

<b>ACT-0001</b>	<b>Alumno</b>
<b>Descripción.</b>	Este actor representa a <i>la persona con parálisis cerebral usuaria de la aplicación.</i>

Figura 9.1 Descripción del actor de los casos de uso



### 9.3. Diagrama de Casos de Uso

Un caso de uso representa una interacción típica entre el usuario y el sistema. Los casos de uso se utilizan para capturar los requisitos funcionales del sistema. La descripción de éstos se centra en el comportamiento y no en la implementación de las partes que define.

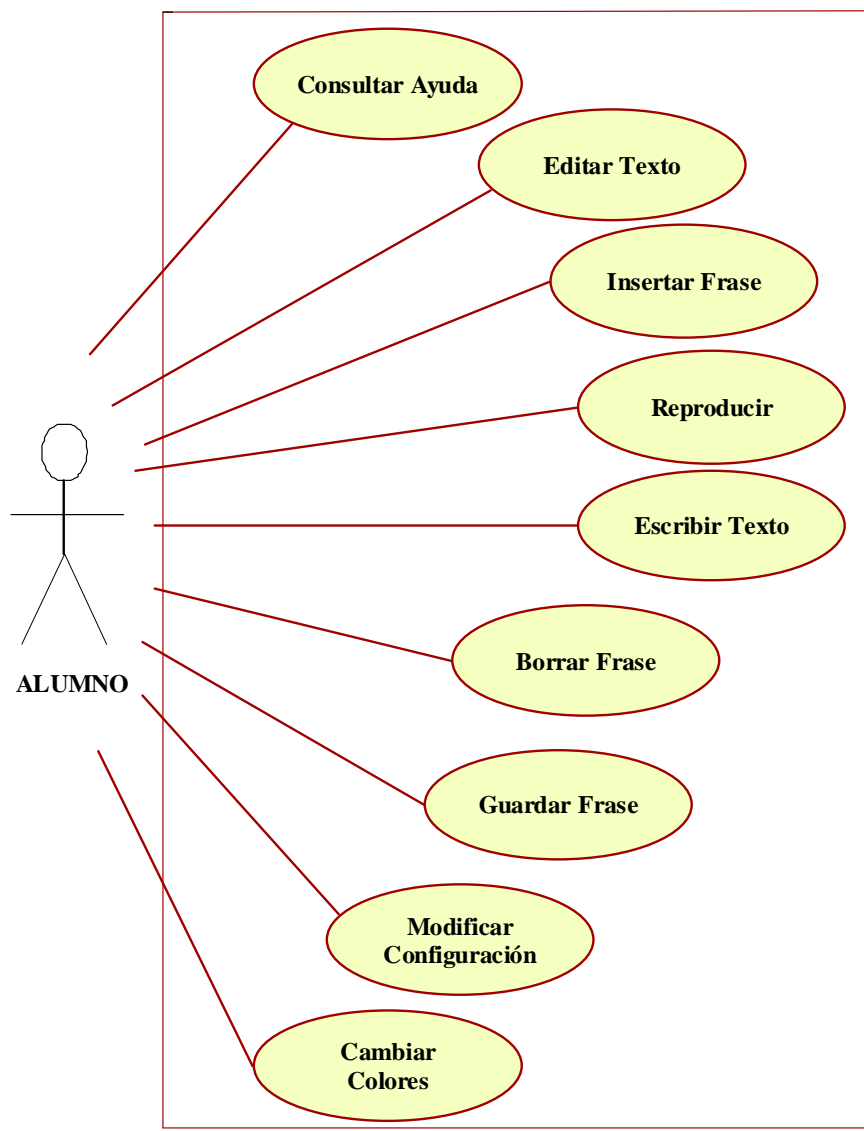


Figura 9.2: Diagrama de Casos de Uso

## 9.4. Descripción de los casos de uso

Un caso de uso describe una funcionalidad más una interacción entre un actor y un sistema en forma de secuencia de acciones.

Un caso de uso debe ser simple, inteligible, claro y conciso.

La descripción debe contener el inicio y el fin del caso de uso, la interacción entre el caso de uso y los actores, el intercambio de datos y la cronología y origen de las interacciones.

### 9.4.1. Consultar Ayuda

UC-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, según se indica en el siguiente caso de uso.</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <a href="#">Alumno (ACT-0001)</a> solicita ayuda al sistema.
	2	El sistema le ofrece una serie de posibilidades sobre las que puede ayudarle.
	3	Actor <a href="#">Alumno (ACT-0001)</a> escoge un tema de ayuda.
4	El sistema le proporciona la ayuda correspondiente sobre el tema seleccionado.	

Figura 9.3. Descripción del caso de uso Consultar Ayuda

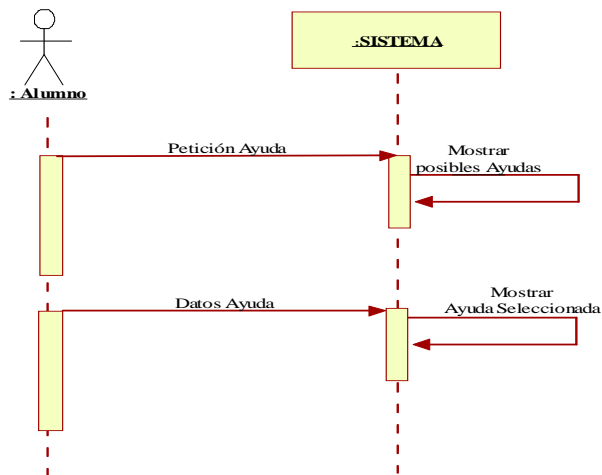


Figura 9.4. Diagrama del caso de uso Consultar Ayuda

### 9.4.2. Modificar Configuración

<b>UC-0002</b>	<b>Modificar configuración</b>	
<b>Descripción</b>	<i>El sistema deberá proporcionar la posibilidad al usuario de cambiar la configuración de los botones, según se indica en el siguiente caso de uso.</i>	
<b>Precondición</b>	Configuración actual	
<b>Secuencia normal</b>	<b>Paso</b> <b>Acción</b>	
	1	Actor <u>Alumno (ACT-0001)</u> solicita modificar la configuración.
	2	El sistema le ofrece la posibilidad de modificar o cambiar la configuración de los botones.
	3	Actor <u>Alumno (ACT-0001)</u> selecciona la configuración deseada.
4	El sistema actualiza los cambios.	
<b>Postcondición</b>	Nueva configuración	

Figura 9.5. Descripción del caso de uso Modificar Configuración

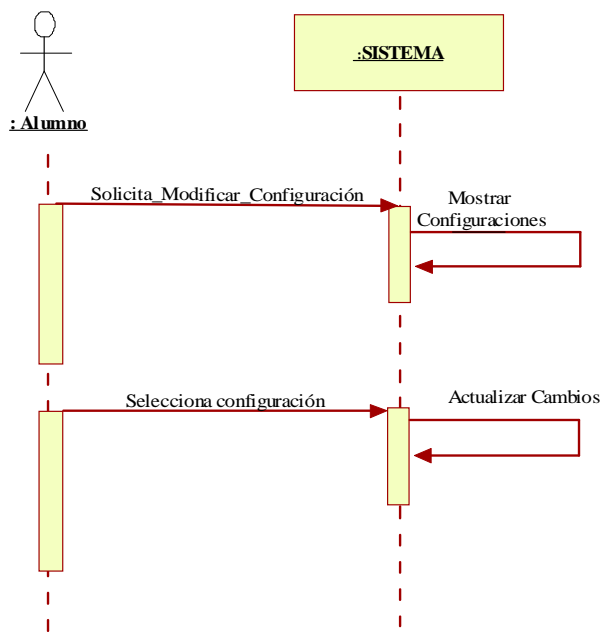


Figura 9.6. Diagrama del caso de uso Modificar Configuración

### 9.4.3. Escribir Texto

<b>UC-0003</b>	<b>Escribir texto</b>	
<b>Descripción</b>	<i>El sistema deberá permitir que el usuario pueda introducir texto, según se indica en el siguiente caso de uso.</i>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <u>Alumno (ACT-0001)</u> selecciona los caracteres que desea escribir.
	2	El sistema actualiza el texto.

Figura 9.7. Descripción del caso de uso Escribir Texto

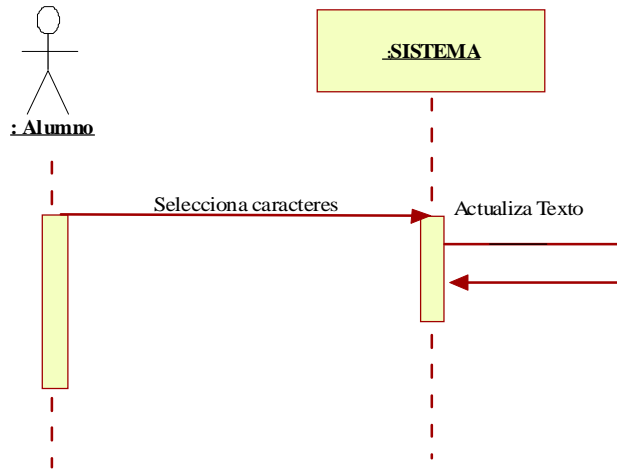


Figura 9.8. Diagrama del caso de uso Escribir Texto

### 9.4.4. Insertar Frase

<b>UC-0004</b>	<b>Insertar frase</b>	
<b>Descripción</b>	<i>El sistema tiene que ofrecer al usuario una lista de frases comunes para que pueda el alumno disponer de ellas en cada momento de manera que se agilice la comunicación, según se indica en el siguiente caso de uso.</i>	
<b>Precondición</b>	texto actual	
<b>Secuencia normal</b>	<b>Paso</b> <b>Acción</b>	
	1	Actor <u>Alumno (ACT-0001)</u> solicita ver las frases disponibles.
	2	El sistema muestra las frases que tiene almacenadas.
	3	Actor <u>Alumno (ACT-0001)</u> escoge la frase con la que quiere comunicarse y pulsa insertar.
	4	El sistema actualiza el texto.
<b>Postcondición</b>	Texto actual mas la frase insertada	

Figura 9.9. Descripción del caso de uso Insertar Frase

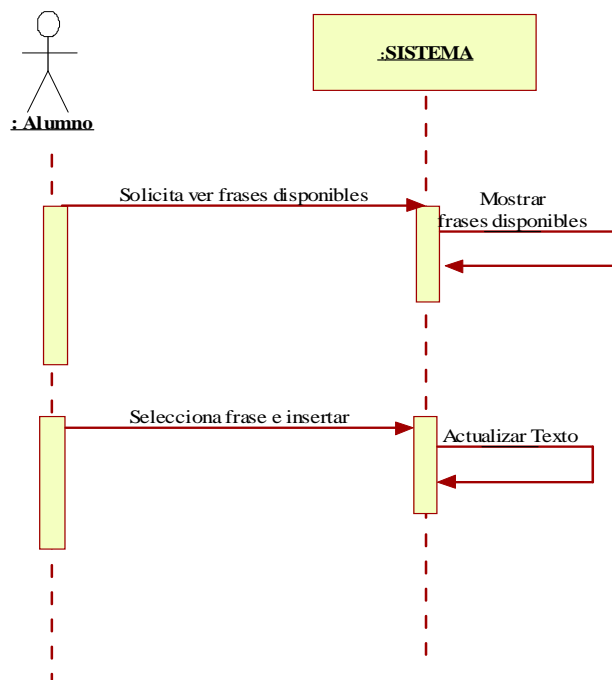


Figura 9.10. Diagrama del caso de uso Insertar Frase

### 9.4.5. Borrar Frase

<b>UC-0005</b>	<b>Borrar frase.</b>	
<b>Descripción</b>	<i>El sistema tiene que ofrecer la posibilidad de borrar las frases que tiene almacenadas si el usuario considera que no son lo suficientemente útiles, según se indica en el siguiente caso de uso</i>	
<b>Precondición</b>	La aplicación dispone de n frases	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <u>Alumno (ACT-0001)</u> solicita ver las frases disponibles.
	2	El sistema muestra la lista de frases de las que dispone.
	3	Actor <u>Alumno (ACT-0001)</u> escoge la frase y pulsa borrar.
	4	El sistema actualiza la lista de frases.
<b>Postcondición</b>	La aplicación dispone de n-1 frases	

Figura 9.11. Descripción del caso de uso Borrar Frase

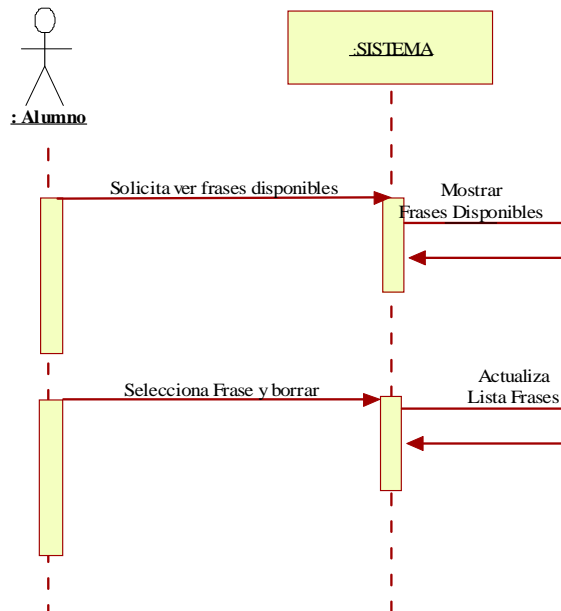


Figura 9.12. Diagrama del caso de uso Borrar Frase

### 9.4.6. Guardar Frase

<b>UC-0006</b>	<b>Guardar frase</b>	
<b>Descripción</b>	<i>El sistema tiene que permitir al usuario que guarde sus frases si lo desea para su posterior uso, para conseguir una comunicación más rápida y natural, según se indica en el siguiente caso de uso.</i>	
<b>Precondición</b>	Hay texto escrito	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <u>Alumno (ACT-0001)</u> solicita guardar la frase.
	2	El sistema actualiza la lista de frases.
<b>Postcondición</b>	En el sistema hay una frase más disponible.	

Figura 9.13. Descripción del caso de uso Guardar Frase

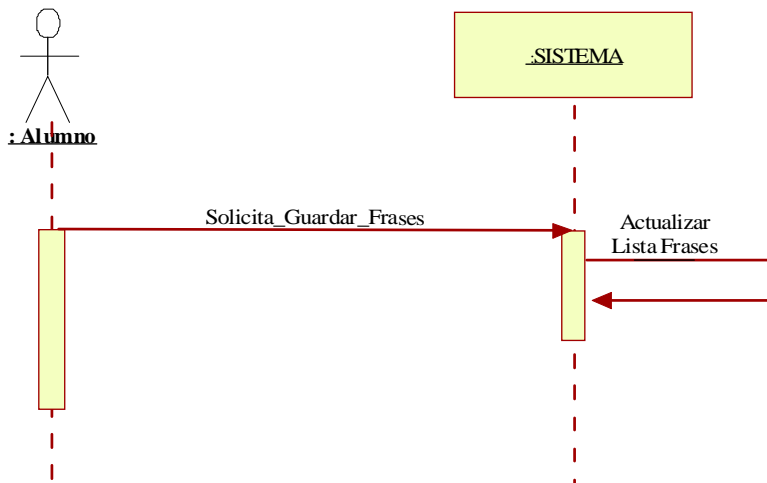


Figura 9.14. Diagrama del caso de uso Guardar Frase

### 9.4.7. Reproducir Texto

<b>UC-0007</b>	<b>Reproducir texto</b>	
<b>Descripción</b>	<i>El sistema tiene que permitir reproducir las frases guardadas y el texto escrito, según se indica en el siguiente caso de uso.</i>	
<b>Precondición</b>	Hay texto escrito	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <a href="#">Alumno (ACT-0001)</a> solicita reproducir el texto escrito.
	2	El sistema reproducirá el texto correspondiente.

Figura 9.15. Descripción del caso de uso Reproducir Texto

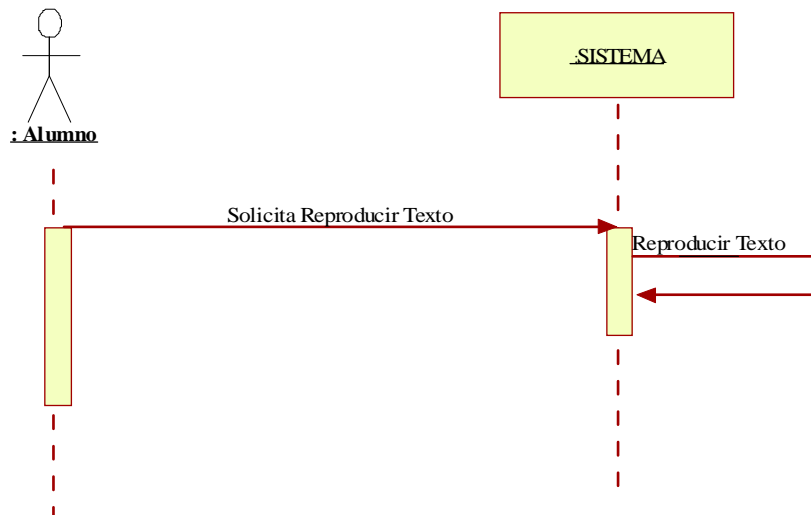


Figura 9.16. Diagrama del caso de uso Reproducir Texto



### 9.4.8. Editar Texto

<b>UC-0008</b>	<b>Editar texto</b>	
<b>Descripción</b>	<i>El sistema tiene que permitir editar el texto escrito, según se muestra en el siguiente caso de uso.</i>	
<b>Precondición</b>	Hay texto escrito	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <u>Alumno (ACT-0001)</u> solicita editar el texto.
	2	El sistema permite modificar el texto.
	3	Actor <u>Alumno (ACT-0001)</u> modifica el texto.
	4	El sistema actualiza los cambios.

Figura 9.17. Descripción del caso de uso Editar Texto

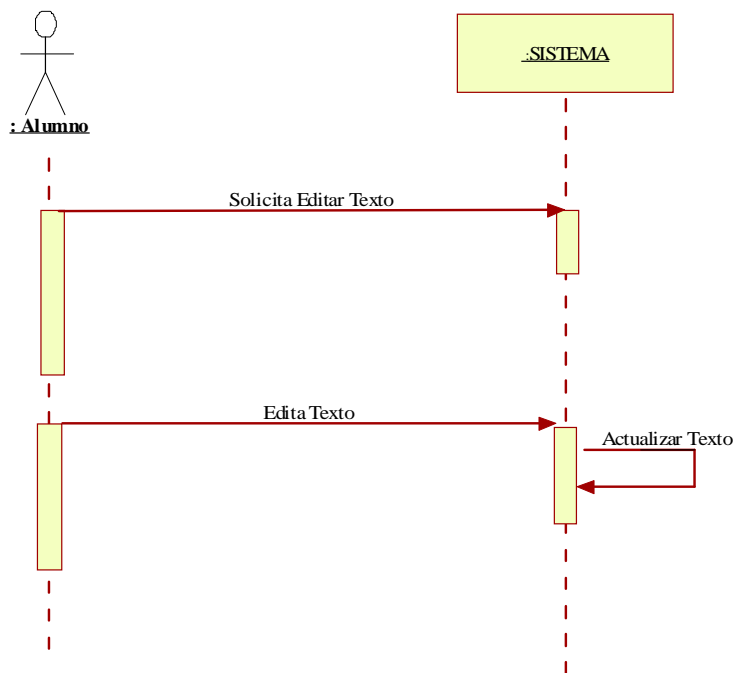


Figura 9.18. Diagrama del caso de uso Editar Texto

### 9.4.9. Cambiar Color

<b>UC-0009</b>	<b>Cambiar color</b>	
<b>Descripción</b>	<i>El sistema deberá permitir al usuario cambiar los colores de todos los elementos que aparecen en la aplicación según se describe en el siguiente caso de uso.</i>	
<b>Secuencia normal</b>	<b>Paso</b> <b>Acción</b>	
	1	Actor <u>Alumno (ACT-0001)</u> solicita elegir colores.
	2	El sistema muestra los elementos de la aplicación a los que se puede cambiar el color.
	3	Actor <u>Alumno (ACT-0001)</u> selecciona el tipo de elemento al que quiere cambiar el color.
	4	El sistema muestra los colores disponibles.
	5	Actor <u>Alumno (ACT-0001)</u> elige el color que desea cambiar.
6	El sistema actualiza los cambios.	

Figura 9.19. Descripción del caso de uso Cambiar Color

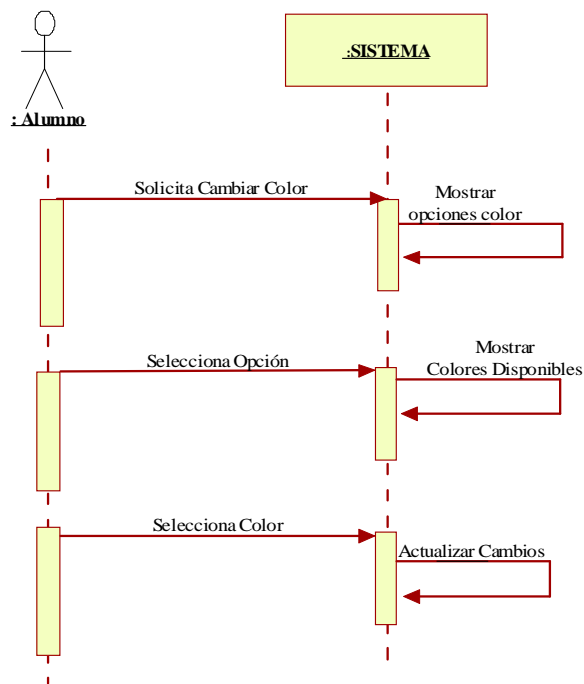


Figura 9.20. Diagrama del caso de uso Cambiar Color

## 9.5. Modelo de objetos

El Modelo de Objetos captura la estructura estática del sistema, mostrando los objetos, las relaciones entre ellos, los atributos y las operaciones. Captura aquellos objetos del mundo real que son importantes para la aplicación.

### 9.5.1. Identificación de las clases

Para realizar el modelo de objetos del sistema es imprescindible encontrar las clases que van a formar parte del dominio del problema.

A continuación se muestran los candidatos a clases obtenidos tras realizar el estudio de la descripción del problema:

- **Teclado:** Clase principal, que representará el teclado virtual.
- **Teclado principal:** Clase que creará el teclado del comunicador.
- **Configuración:** Clase que representará las distintas configuraciones con las que podrá trabajar el comunicador.
- **Reproductor:** Clase que se encargará de reproducir el texto escrito.
- **Frases:** Clase que se encargará de manejar las frases disponibles del comunicador.
- **Caracteres:** Clase que creará los sub-teclados del comunicador, en las que se redistribuyen los caracteres de forma independiente.

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

### 9.5.2. Diagrama inicial de clases

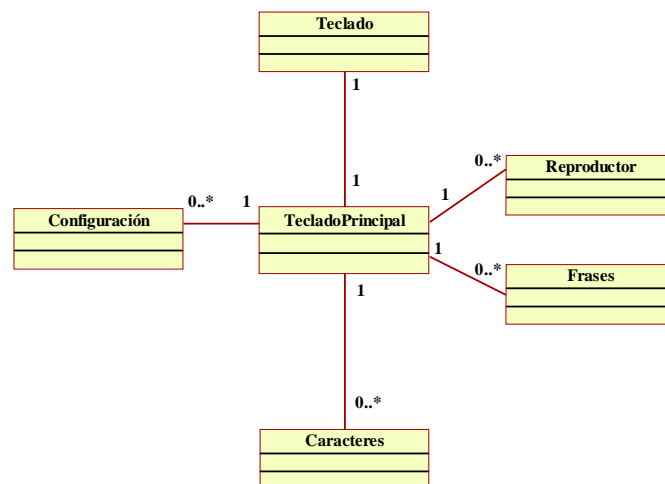


Figura 9.21. Diagrama inicial de clases.



## CAPITULO 10: DISEÑO

Una vez finalizado el análisis del sistema, y antes de realizar la implementación, habrá que hacer un diseño del sistema acorde con la especificación.

El diseño determina la arquitectura general del sistema que mejor satisface los requisitos: componentes físicos del sistema (hardware), el software utilizado y la interacción con el usuario. Define como debe realizar su función el sistema.

### 10.1. Casos de Uso y Diagramas de Secuencia

#### 10.1.1. Consultar Ayuda

UC-0001	Consultar ayuda	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario solicita consultar ayuda</i> .	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <a href="#">Alumno (ACT-0001)</a> pulsa el botón MENU de la pantalla principal del dispositivo móvil.
	2	El sistema muestra las diferentes opciones que contiene el menú principal.
	3	Actor <a href="#">Alumno (ACT-0001)</a> pulsa el botón AYUDA del menú.
	4	El sistema despliega el menú con los posibles temas de ayuda.
	5	Actor <a href="#">Alumno (ACT-0001)</a> pulsa el botón que se corresponde con la ayuda deseada.
6	El sistema muestra la información correspondiente al tema de ayuda solicitado.	
<b>Importancia</b>	importante	
<b>Comentarios</b>	El actor puede hacer las consultas a la ayuda en cualquier momento de uso de la aplicación.	

Figura 10.1. Caso de Uso Consultar Ayuda

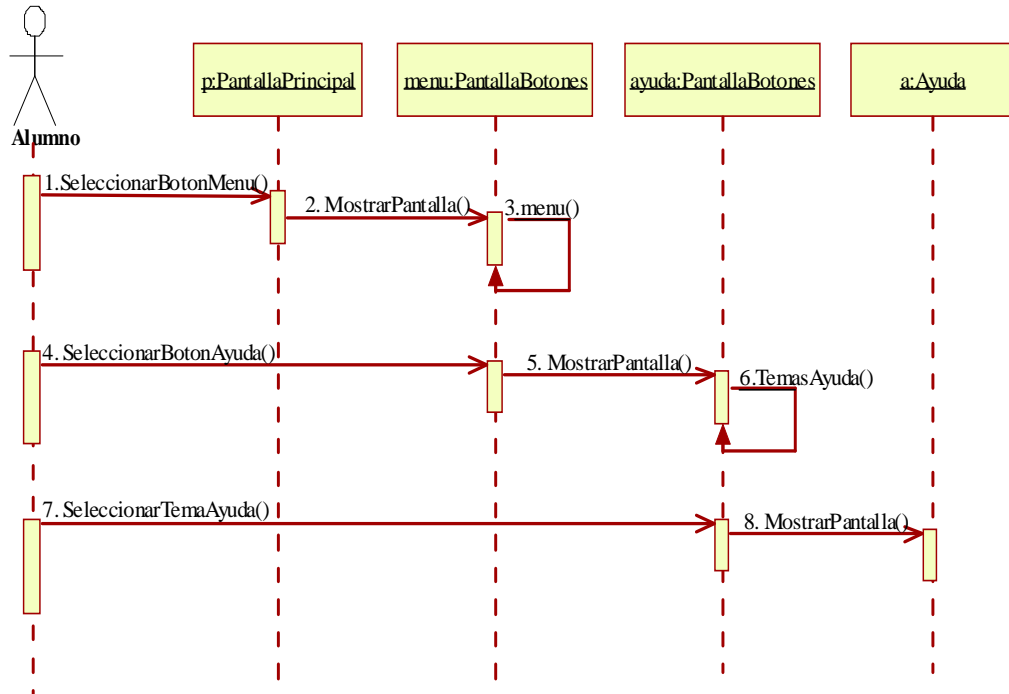


Figura 10.2. Diagrama de Secuencia Consultar Ayuda

### 10.1.2. Modificar Configuración

<b>UC-0002</b>	<b>Modificar configuración</b>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario solicite cambiar la configuración de los botones de la aplicación según sus necesidades.</i>	
<b>Precondición</b>	Configuración actual	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <u>Alumno (ACT-0001)</u> pulsa el botón MENÚ del dispositivo móvil.
	2	<i>El sistema muestra las diferentes opciones que contiene el menú principal.</i>
	3	Actor <u>Alumno (ACT-0001)</u> pulsa el botón CONFIGURACIÓN del menú.
	4	<i>El sistema muestra las configuraciones disponibles de la aplicación.</i>
	5	Actor <u>Alumno (ACT-0001)</u> elige la configuración deseada
	6	<i>El sistema actualiza los cambios</i>
<b>Poscondición</b>	Nueva configuración	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	-	-
<b>Importancia</b>	Importante	
<b>Comentarios</b>	El actor puede hacer el cambio de configuración de los botones en cualquier momento de uso de la aplicación.	

Figura 10.3. Caso de Uso Modificar Configuración

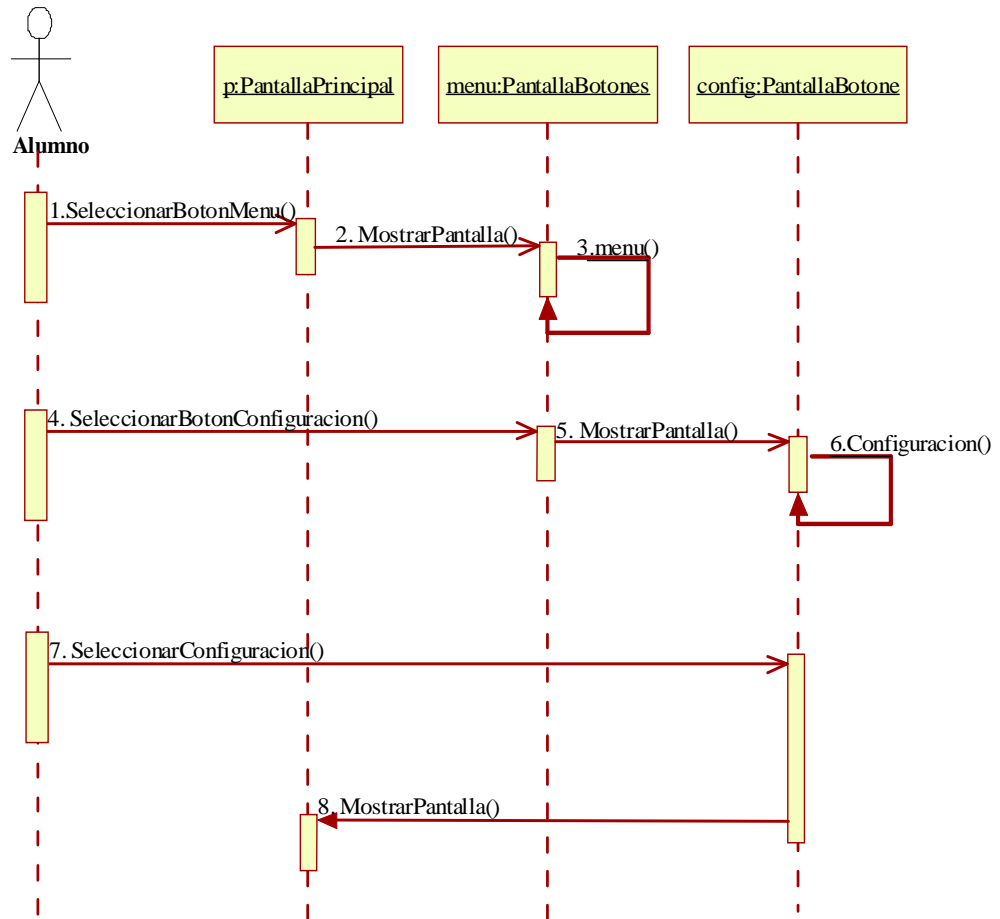


Figura 10.4. Diagrama de Secuencia Modificar Configuración



### 10.1.3. Escribir Texto

<b>UC-0003</b>	<b>Escribir texto</b>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee introducir texto en cualquier momento</i>	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <u>Alumno (ACT-0001)</u> <i>pulsa las letras, números o símbolos que desea escribir.</i>
	2	<i>El sistema actualiza el texto escrito.</i>
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	<i>En caso de equivocación, el actor Alumno (ACT-0001) podrá borrar el carácter incorrecto y volver al paso 1.</i>
<b>Importancia</b>	Vital	
<b>Comentarios</b>	Este caso de uso puede utilizarse en cualquier momento porque forma parte de la comunicación	

Figura 10.5. Caso de Uso Escribir Texto

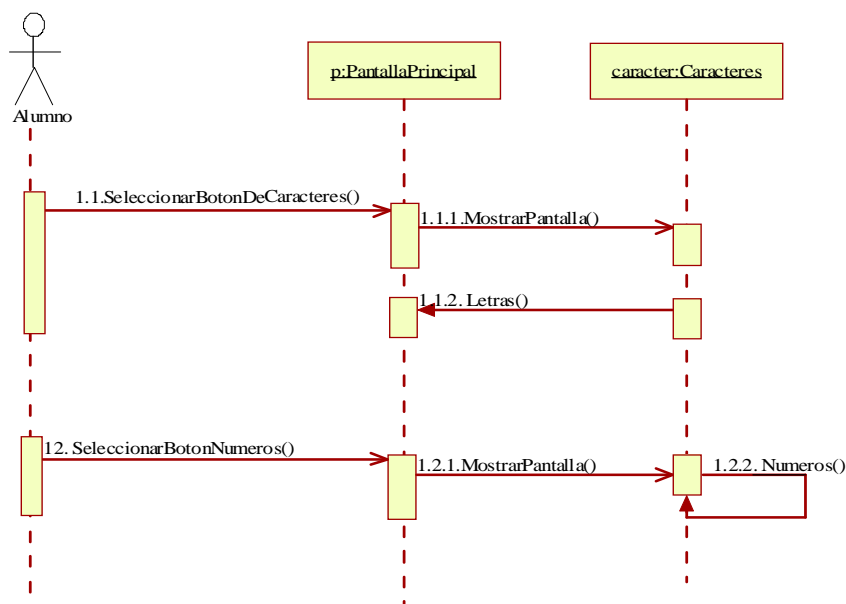


Figura 10.6. Diagrama de Secuencia Escribir Texto

### 10.1.4. Insertar Frase

<b>UC-0004</b>	<b>Insertar frase</b>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario quiera insertar frases en el texto escrito hasta el momento</i>	
<b>Precondición</b>	texto actual	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <u>Alumno (ACT-0001)</u> pulsa el botón MENU del dispositivo móvil.
	2	El sistema muestra las diferentes opciones del menú principal.
	3	Actor <u>Alumno (ACT-0001)</u> pulsa el botón FRASES del menú.
	4	El sistema muestra las frases disponibles en la aplicación y los botones que permiten su uso.
	5	Actor <u>Alumno (ACT-0001)</u> escoge la frase deseada y pulsa el botón INSERTAR.
	6	El sistema actualiza el texto escrito.
<b>Postcondición</b>	texto actual más la frase insertada	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	5	Si no existen frases en el sistema o el usuario no ha seleccionado ninguna de las disponible, el sistema no permitirá insertar una frase en el texto actual.
<b>Importancia</b>	importante	
<b>Comentarios</b>	Este caso de uso se da cuando el usuario quiere insertar una frase de las disponibles en el sistema en el texto escrito para poder hacer uso de ella (reproducirla o editarla)	

Figura 10.7. Caso de Uso Insertar Frase

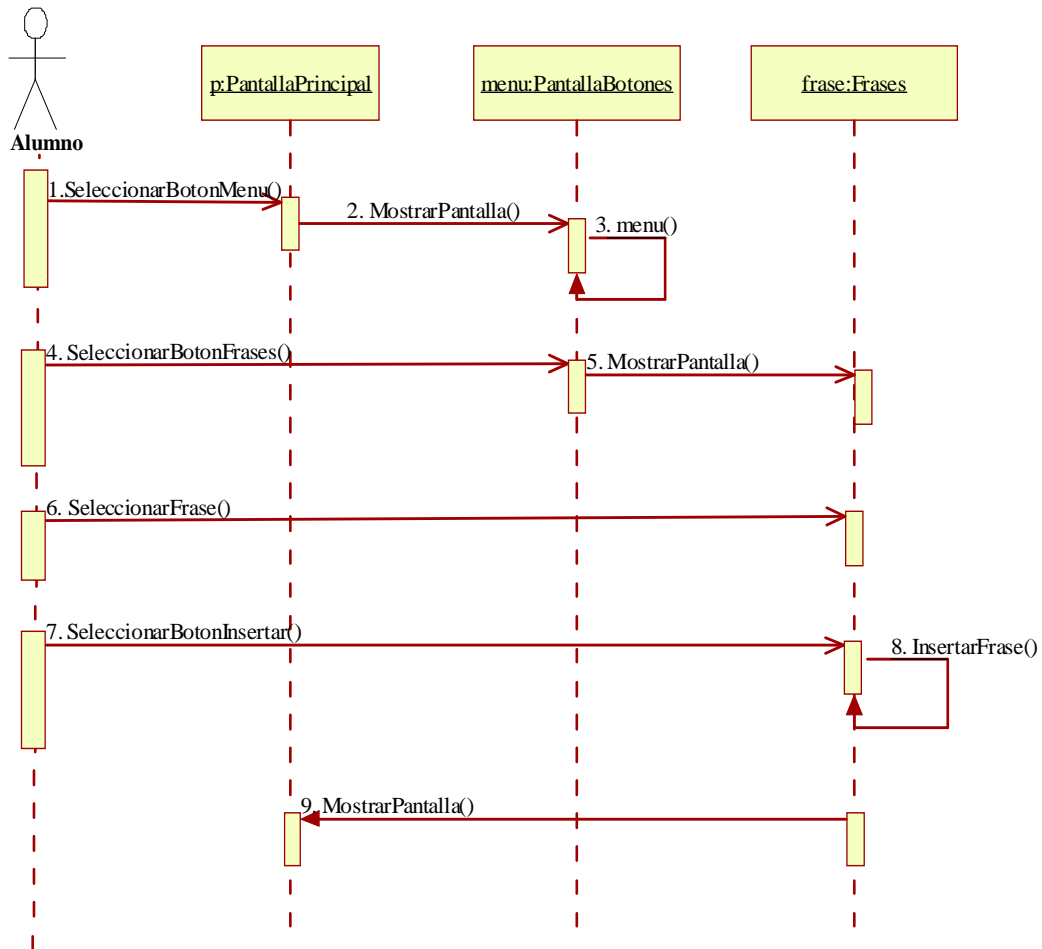


Figura 10.8. Diagrama de Secuencia Insertar Frase

### 10.1.5. Borrar Frase

<b>UC-0005</b>	<b>Borrar frase</b>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee borrar una frase de las disponibles en la aplicación.</i>	
<b>Precondición</b>	La aplicación dispone de n frases	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <u>Alumno (ACT-0001)</u> pulsa el botón MENU del dispositivo móvil.
	2	El sistema muestra las diferentes opciones del menú principal.
	3	Actor <u>Alumno (ACT-0001)</u> pulsa el botón FRASES del menú principal.
	4	El sistema muestra las frases disponibles en la aplicación y los botones que permiten usarlas.
	5	Actor <u>Alumno (ACT-0001)</u> selecciona la frase y pulsa el botón BORRAR.
	6	El sistema actualiza la lista de frases disponibles en la aplicación.
<b>Postcondición</b>	La aplicación dispone de n-1 frases	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	5	Si no existen frases en el sistema, éste no permitirá su borrado
<b>Importancia</b>	Puede quedar bien	
<b>Comentarios</b>	Este caso de uso se dará cuando el actor Alumno (ACT-0001) considere que alguna de las frases almacenadas en la aplicación no son lo suficientemente útiles.	

Figura 10.9. Caso de Uso Borrar Frase

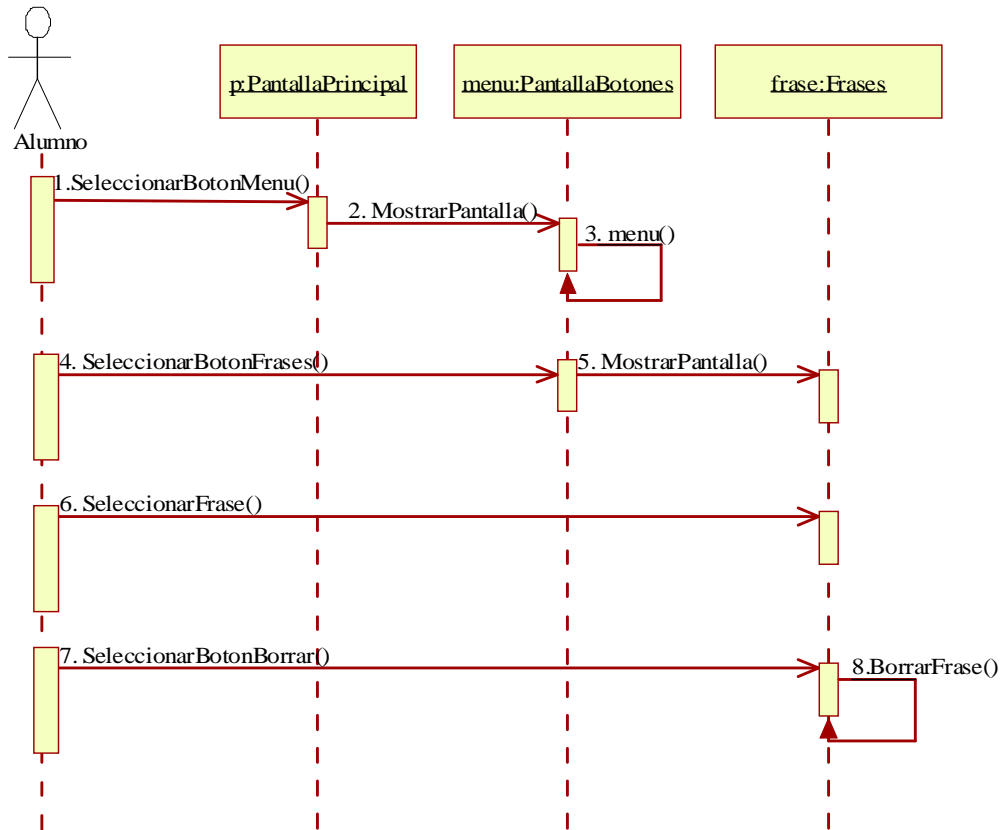


Figura 10.10. Diagrama de Secuencia Borrar Frase

### 10.1.6. Guardar Frase

<b>UC-0006</b>	<b>Guardar frase</b>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario quiera almacenar en la aplicación las frases que desee para su posterior uso.</i>	
<b>Precondición</b>	Hay texto escrito	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <u>Alumno (ACT-0001)</u> pulsa el botón MENU del dispositivo móvil.
	2	El sistema muestra las diferentes opciones del menú principal.
	3	Actor <u>Alumno (ACT-0001)</u> pulsa el botón FRASES.
	4	El sistema muestra las frases almacenadas y varios botones, entre ellos el botón GUARDAR.
	5	Actor <u>Alumno (ACT-0001)</u> pulsa el botón GUARDAR.
	6	El sistema añade el texto escrito como una frase al fichero donde éstas se almacenan.
<b>Poscondición</b>	En el sistema hay una frase más disponible.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	5	Si no hay texto escrito, la aplicación no permitirá guardarlo.
<b>Importancia</b>	Importante	
<b>Comentarios</b>	El sistema pretende tener almacenadas las frases usadas más comúnmente por el usuario para permitir mayor velocidad en la comunicación.	

Figura 10.11. Caso de uso Guardar Frase

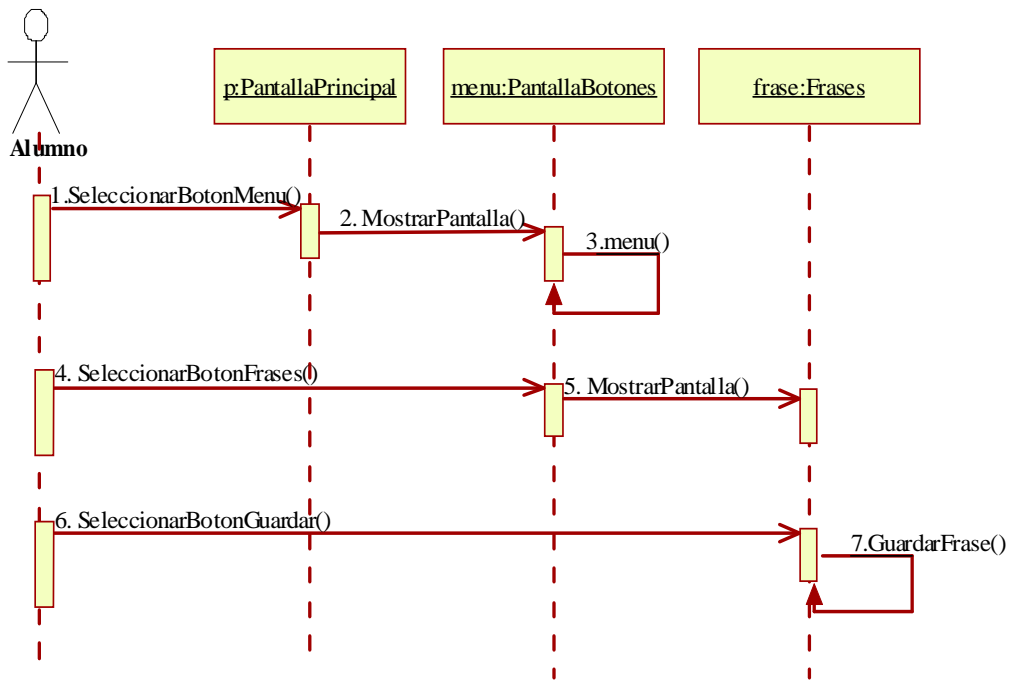


Figura 10.12. Diagrama de Secuencia Guardar Frase

### 10.1.7. Reproducir Texto

<b>UC-0007</b>	<b>Reproducir texto</b>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario quiera reproducir un texto escrito</i> .	
<b>Precondición</b>	Hay texto escrito	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <u>Alumno (ACT-0001)</u> pulsa el botón MENU del dispositivo móvil
	2	El sistema muestra las diferentes opciones del menú principal.
	3	Actor <u>Alumno (ACT-0001)</u> pulsa el botón REPRODUCIR
	4	El sistema reproduce el texto escrito hasta el momento.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si no hay texto escrito, la aplicación no permitirá su reproducción.
<b>Importancia</b>	vital	
<b>Comentarios</b>	El usuario puede reproducir en cualquier momento el texto escrito y/o las frases disponibles en la aplicación. Este caso de uso es importante para aquellas personas con problemas de comunicación y escritura.	

Figura 10.13. Caso de uso Reproducir texto.

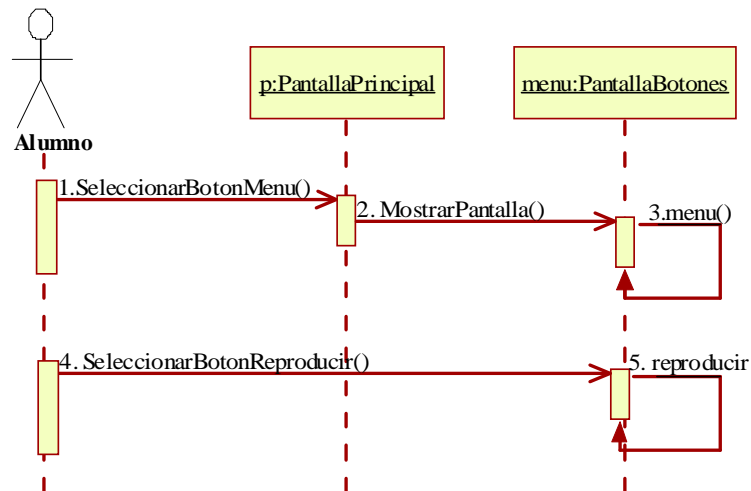


Figura 10.14. Diagrama de Secuencia Reproducir texto.



### 10.1.8. Editar Texto

<b>UC-0008</b>	<b>Editar texto</b>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee editar el texto escrito hasta el momento, pudiendo borrar la última palabra o el texto completo, además de reproducirlo.</i>	
<b>Precondición</b>	Hay texto escrito	
<b>Secuencia ordinaria</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <u>Alumno (ACT-0001)</u> pulsa el botón MENU del dispositivo móvil.
	2	<i>El sistema muestra las diferentes opciones del menú principal</i>
	3	Actor <u>Alumno (ACT-0001)</u> pulsa el boton EDITAR
	4	<i>El sistema muestra el texto escrito y varios botones de edición de texto</i>
	5a	Actor <u>Alumno (ACT-0001)</u> pulsa el botón de edición de texto BORRAR PALABRA
	5b	Actor <u>Alumno (ACT-0001)</u> pulsa el botón de edición de texto BORRAR TODO
	6	<i>El sistema actualiza el texto editado</i>
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	5a,5b	<i>Si no hay texto escrito, la aplicación no permitirá editarlo.</i>
<b>Importancia</b>	vital	
<b>Comentarios</b>	Este caso de uso permite editar el texto escrito en cualquier momento. Resulta útil cuando se produce una equivocación que implique varios caracteres.	

Figura 10.15. Caso de uso Editar Texto.

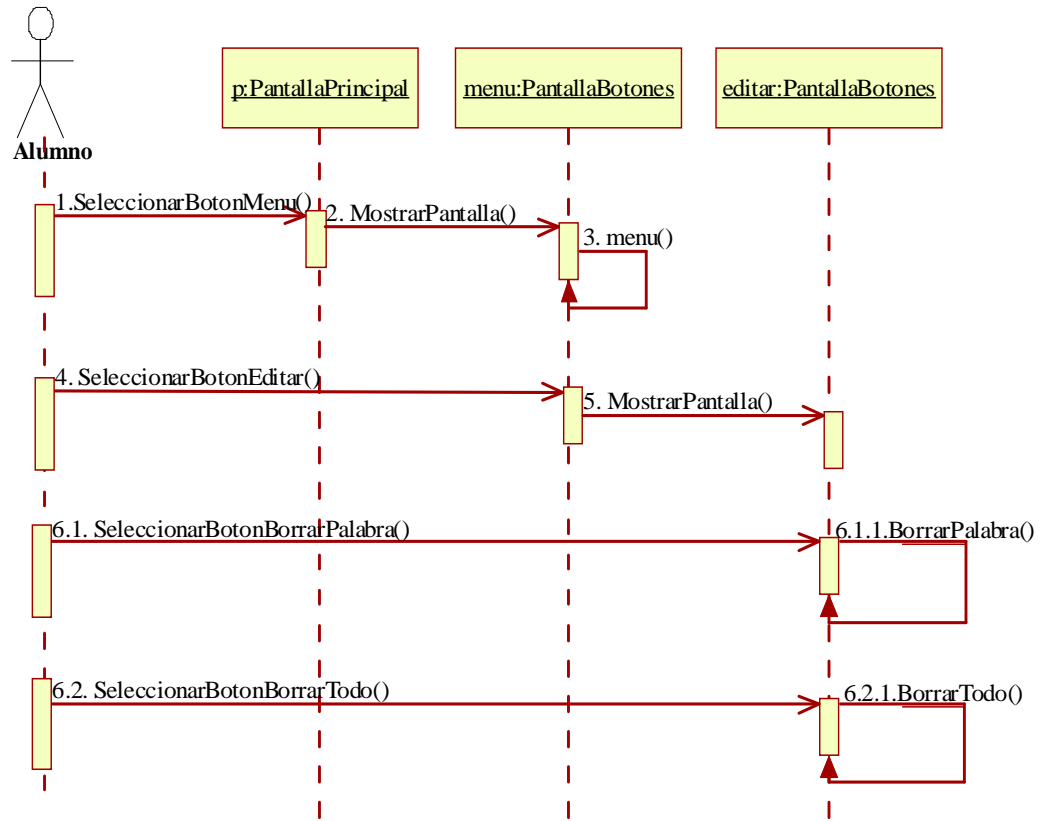


Figura 10.16. Diagrama de Secuencia Editar Texto.

### 10.1.9. Cambiar Color

<b>UC-0009</b>	<b>Cambiar color</b>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario solicite cambiar los colores de los diferentes elementos de la aplicación, como los botones, el fondo o la letra</i>	
<b>Precondición</b>	Color actual	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Actor <u>Alumno (ACT-0001)</u> pulsa el botón MENU del dispositivo móvil
	2	El sistema muestra las diferentes opciones del menú principal
	3	Actor <u>Alumno (ACT-0001)</u> pulsa el botón COLORES
	4	El sistema muestra los elementos de la aplicación a los cuales se les puede cambiar el color
	5a	Actor <u>Alumno (ACT-0001)</u> selecciona el botón CAMBIAR COLOR DE FONDO
	5b	Actor <u>Alumno (ACT-0001)</u> selecciona el botón CAMBIAR COLOR DE LA LETRA
	5c	Actor <u>Alumno (ACT-0001)</u> selecciona el botón CAMBIAR COLOR DEL BOTON
	6	El sistema muestra la lista de colores disponibles
	7	Actor <u>Alumno (ACT-0001)</u> selecciona el color deseado
8	El sistema actualiza los cambios	
<b>Poscondición</b>	Color modificado	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	-	-
<b>Importancia</b>	Puede quedar bien	
<b>Comentarios</b>	Este caso de uso permite al usuario cambiar el color de los elementos de la aplicación ajustado a sus necesidades	

Figura 10.17. Caso de uso Cambiar Color.

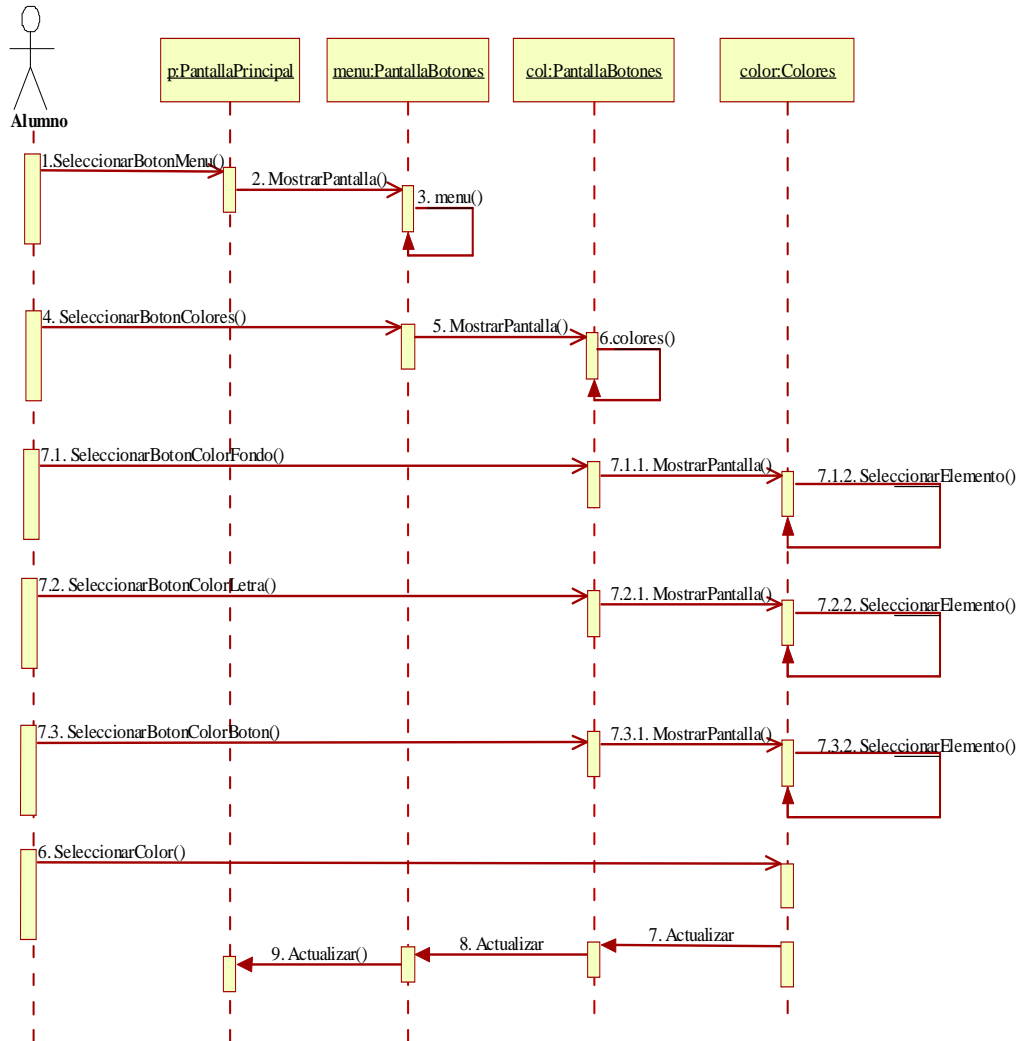


Figura 10.18. Diagrama de Secuencia Cambiar Color.

## 10.2. Diagrama final de clases.

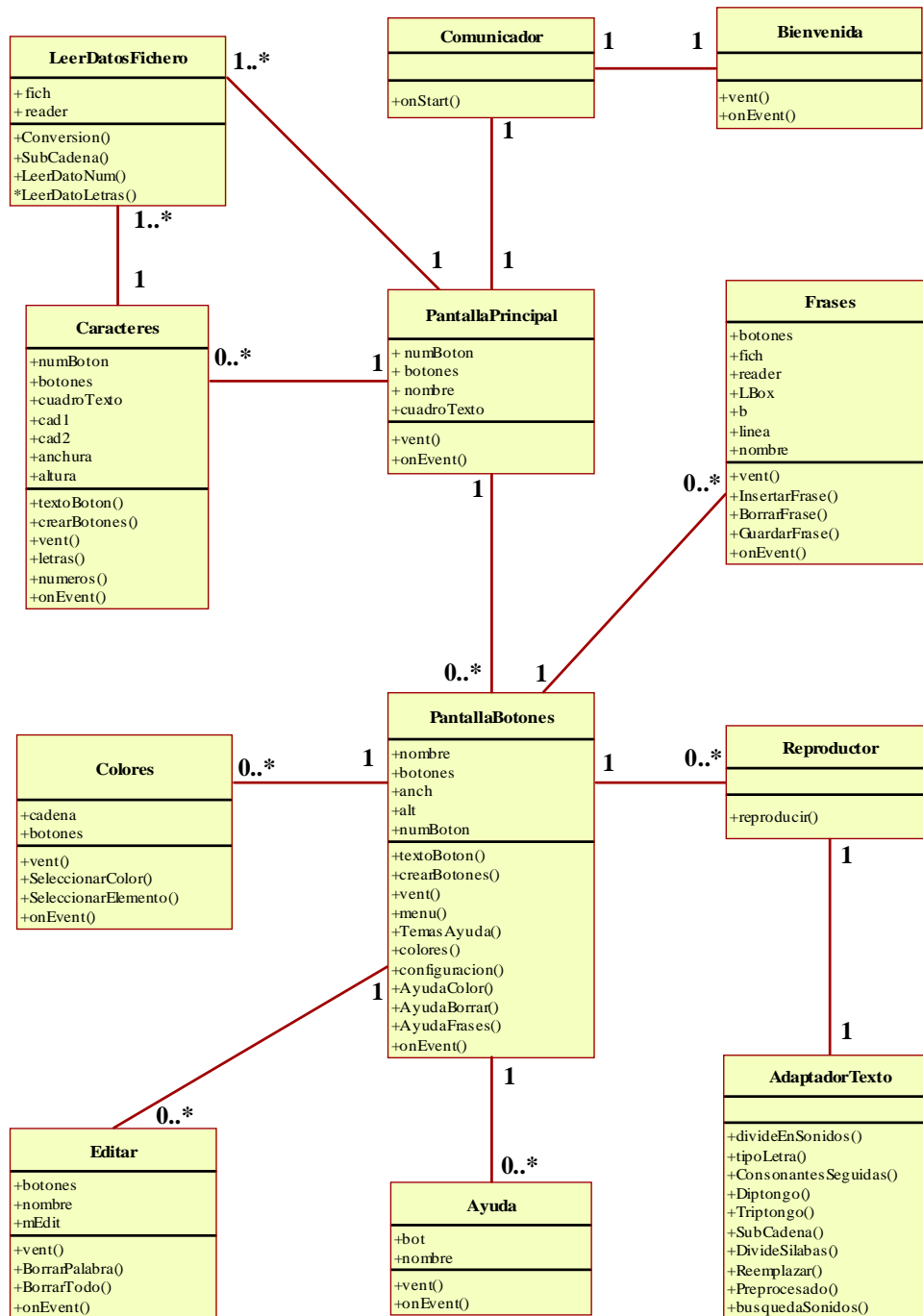


Figura 10.19. Diagrama final de clases.

### 10.3. Diccionario de datos.

A continuación se realizará una descripción de todos los elementos existentes en el sistema mostrados en el diagrama final de clases (Figura 10.19.).

Clase Bienvenida	
<b>Descripción:</b>	Esta clase muestra el titulo de la aplicación así como el nombre de sus autores. Transcurrido un cierto periodo de tiempo esta pantalla se cerrará comenzado la aplicación
<b>Atributos</b>	No tiene atributos
<b>Métodos:</b>	<ul style="list-style-type: none"><li>▪ <i>MostrarPantalla()</i>: crea la ventana con los elementos necesarios</li><li>▪ <i>onEvent(Event e)</i>: este método cierra transcurrido el tiempo indicado.</li></ul>

Figura 10.20. Descripción de la Clase Bienvenida.

Clase Comunicador	
<b>Descripción:</b>	Esta clase crea una ventana principal y lanza la aplicación
<b>Atributos</b>	No tiene atributos
<b>Métodos:</b>	<ul style="list-style-type: none"><li>▪ <i>onStart()</i>: encargado de comenzar la aplicación</li></ul>

Figura 10.21. Descripción de la Clase Comunicador.

Clase PantallaPrincipal	
<b>Descripción:</b>	Esta clase crea la pantalla principal (teclado) de la aplicación con la configuración dada.
<b>Atributos</b>	<ul style="list-style-type: none"><li>▪ <i>numBoton</i>: variable que contiene el numero de botones que tiene la pantalla principal</li><li>▪ <i>botones</i>: array que almacena los botones de la pantalla principal</li><li>▪ <i>nombre</i>: array que contiene el nombre de los botones</li><li>▪ <i>cuadroTexto</i>: variable que muestra el texto escrito</li></ul>
<b>Métodos:</b>	<ul style="list-style-type: none"><li>▪ <i>MostrarPantalla()</i>: método en el que se crea la pantalla principal, es decir, el teclado</li><li>▪ <i>onEvent(Event e)</i>: contiene los eventos asociados a esta pantalla</li></ul>

Figura 10.22. Descripción de la Clase PantallaPrincipal.

### Clase LeerDatosFichero

<b>Descripción:</b>	Esta clase lee los datos necesarios de los ficheros de configuración
<b>Atributos</b>	<ul style="list-style-type: none"><li>▪ <i>fich</i>: fichero que contiene los datos de las configuraciones</li><li>▪ <i>reader</i>: variable utilizada para la lectura de líneas del fichero</li></ul>
<b>Métodos:</b>	<ul style="list-style-type: none"><li>▪ <i>Conversión(String c)</i>: convierte a entero el string dado como parámetro</li><li>▪ <i>SubCadena(int i, String texto)</i>: devuelve el carácter de la posición i de la cadena texto</li><li>▪ <i>LeerDatoNum()</i>: devuelve un dato numérico leído del fichero</li><li>▪ <i>LeerDatoLetras()</i>: devuelve una cadena de caracteres leída del fichero</li></ul>

Figura 10.23. Descripción de la Clase LeerDatosFichero.

### Clase Caracteres

<b>Descripción:</b>	Esta clase crea las pantallas en las que aparecen los caracteres en botones independientes.
<b>Atributos</b>	<ul style="list-style-type: none"><li>▪ <i>numBoton</i>: variable que contiene el numero de botones que tiene la pantalla</li><li>▪ <i>botones</i>: array que almacena los botones de la pantalla</li><li>▪ <i>cad1, cad2</i>: variables que almacenan los nombres de los botones</li><li>▪ <i>cuadroTexto</i>: variable que muestra el texto escrito</li><li>▪ <i>anchura, altura</i>: variable que almacena la anchura y la altura de los botones respectivamente.</li></ul>
<b>Métodos:</b>	<ul style="list-style-type: none"><li>▪ <i>textoBoton()</i>: método que almacena en un array de caracteres el texto que debe aparecer en los botones</li><li>▪ <i>crearBotones()</i>: método encargado de crear los botones que aparecerán en la pantalla</li><li>▪ <i>MostrarPantalla()</i>: método en el que se crea la pantalla</li><li>▪ <i>números(int i)</i>: método encargado de insertar los caracteres numéricos en el cuadro de texto</li><li>▪ <i>letras(int i)</i>: método encargado de insertar los caracteres alfabéticos y signos de puntuación en el cuadro de texto</li><li>▪ <i>onEvent(Event e)</i>: contiene los eventos asociados a esta pantalla</li></ul>

Figura 10.24. Descripción de la Clase Caracteres.

<b>Clase PantallaBotones</b>	
<b>Descripción:</b>	Esta clase crea las pantallas que están compuestas únicamente de botones.
<b>Atributos</b>	<ul style="list-style-type: none"><li>▪ <i>numBoton</i>: variable que contiene el numero de botones que tiene la pantalla</li><li>▪ <i>botones</i>: array que almacena los botones de la pantalla</li><li>▪ <i>nombre</i>: array que contiene el nombre de los botones</li><li>▪ <i>anch,alt</i>: variables que almacenan la anchura y altura de los botones respectivamente</li></ul>
<b>Métodos:</b>	<ul style="list-style-type: none"><li>▪ <i>textoBoton()</i>: este método asigna el numero de botones, la anchura y altura de éstos y almacena en un array el nombre de los botones que componen la pantalla</li><li>▪ <i>crearBotones()</i>: este método crea los botones que componen la pantalla</li><li>▪ <i>MostrarPantalla()</i>: muestra la pantalla</li><li>▪ <i>menu(int i)</i>: método en el que se especifican los eventos asociados a la pantalla menú</li><li>▪ <i>TemasAyuda(int i)</i>: método en el que se especifican los eventos asociados a los temas de ayuda</li><li>▪ <i>Colores(int i)</i>: método en el que se especifican los eventos asociados a la pantalla colores</li><li>▪ <i>Configuración(int i)</i>: método en el que se especifican los eventos asociados a la pantalla configuración</li><li>▪ <i>AyudaColor(int i)</i>: método en el que se especifican los eventos asociados a la pantalla de Ayuda para cambiar el color</li><li>▪ <i>AyudaBorrar(int i)</i>: método en el que se especifican los eventos asociados a la pantalla de ayuda para borrar</li><li>▪ <i>AyudaFrasas(int i)</i>: método en el que se especifican los eventos asociados a la pantalla de ayuda para frases</li><li>▪ <i>onEvent(Event e)</i>: contiene los eventos asociados a esta pantalla</li></ul>

Figura 10.25. Descripción de la Clase PantallaBotones.

<b>Clase Editar</b>	
<b>Descripción:</b>	Esta clase crea la pantalla que permite editar el texto
<b>Atributos</b>	<ul style="list-style-type: none"><li>▪ <i>botones</i>: array que almacena los botones de la pantalla</li><li>▪ <i>nombre</i>: array que almacena el texto de los botones</li><li>▪ <i>mEdit</i>: variable que muestra el texto escrito en múltiples líneas</li></ul>
<b>Métodos:</b>	<ul style="list-style-type: none"><li>▪ <i>MostrarPantalla()</i>: método en el que se crea la pantalla</li><li>▪ <i>BorrarPalabra()</i>: método que se encarga de borrar la última palabra escrita</li><li>▪ <i>BorrarTodo()</i>: método que se encarga de borrar todo el texto escrito</li><li>▪ <i>onEvent(Event e)</i>: contiene los eventos asociados a esta pantalla</li></ul>

Figura 10.26. Descripción de la Clase Editar.



### Clase Frases

<b>Descripción:</b>	Esta clase crea la pantalla que permite manejar frases
<b>Atributos</b>	<ul style="list-style-type: none"><li>▪ <i>botones</i>: array que almacena los botones de la pantalla</li><li>▪ <i>fich</i>: fichero que almacena las frases de la aplicación</li><li>▪ <i>reader</i>: variable utilizada para la lectura de líneas del fichero</li><li>▪ <i>LBox</i>: lista que muestra las frases disponibles</li><li>▪ <i>b</i>: variable de tipo byte utilizada para la escritura de datos en el fichero</li><li>▪ <i>linea</i>: variable en la que se almacena temporalmente la línea leída del fichero</li><li>▪ <i>nombre</i>: array que almacena el texto de los botones</li></ul>
<b>Métodos:</b>	<ul style="list-style-type: none"><li>▪ <i>MostrarPantalla()</i>: método en el que se crea la pantalla</li><li>▪ <i>InsertarFrase()</i>: método que se encarga de insertar la frase seleccionada por el usuario en el cuadro de texto</li><li>▪ <i>BorrarFrase()</i>: método que se encarga de borrar de la lista de frases disponibles, la frase seleccionada por el usuario</li><li>▪ <i>GuardarFrase()</i>: método que se encarga de guardar en la lista, el texto escrito en el cuadro de texto</li><li>▪ <i>onEvent(Event e)</i>: contiene los eventos asociados a esta pantalla</li></ul>

Figura 10.27. Descripción de la Clase Frases.

### Clase Colores

<b>Descripción:</b>	Esta clase crea la pantalla que permite modificar el color de los elementos de la aplicación
<b>Atributos</b>	<ul style="list-style-type: none"><li>▪ <i>botones</i>: array que almacena los botones de la pantalla</li><li>▪ <i>cadena</i>: variable en la que se almacena el nombre del botón</li></ul>
<b>Métodos:</b>	<ul style="list-style-type: none"><li>▪ <i>MostrarPantalla()</i>: método en el que se crea la pantalla</li><li>▪ <i>SeleccionarColor(int i)</i>: método que devuelve el color asociado a cada botón</li><li>▪ <i>SeleccionarElemento(Color col)</i>: método que devuelve el elemento sobre el que se va a realizar un cambio de color</li><li>▪ <i>onEvent(Event e)</i>: contiene los eventos asociados a esta pantalla</li></ul>

Figura 10.28. Descripción de la Clase Colores

### Clase Reproductor

<b>Descripción:</b>	Esta clase permite reproducir el texto escrito
<b>Atributos</b>	No tiene atributos
<b>Métodos:</b>	<ul style="list-style-type: none"><li>▪ <i>reproducir()</i>: se encarga de proporcionar los sonidos</li></ul>

Figura 10.29. Descripción de la Clase Reproductor.

<b>Clase Ayuda</b>	
<b>Descripción:</b>	Esta clase crea la pantalla donde se muestra la ayuda solicitada
<b>Atributos</b>	<ul style="list-style-type: none"><li>▪ <i>bot</i>: array que almacena los botones de la pantalla</li><li>▪ <i>nombre</i>: array que almacena el texto de los botones</li></ul>
<b>Métodos:</b>	<ul style="list-style-type: none"><li>▪ <i>MostrarPantalla()</i>: método en el que se crea la pantalla</li><li>▪ <i>onEvent(Event e)</i>: contiene los eventos asociados a esta pantalla</li></ul>

Figura 10.30. Descripción de la Clase Ayuda.

<b>Clase AdaptadorTexto</b>	
<b>Descripción:</b>	Esta clase lleva a cabo las modificaciones pertinentes sobre el texto para dar lugar a una lista de ficheros reproducibles, y además implementa la síntesis de voz
<b>Atributos</b>	No tiene atributos
<b>Métodos:</b>	<ul style="list-style-type: none"><li>▪ <i>divideEnSonidos()</i>: método que devuelve la lista final de sonidos para ser reproducidos</li><li>▪ <i>tipoLetra(String c)</i>: este método se encarga de comprobar si la cadena es una vocal o una consonante, necesaria para la división en sílabas</li><li>▪ <i>ConsonantesSeguidas(String subcadena)</i>: contiene los eventos asociados a esta pantalla</li><li>▪ <i>Diptongo( String subcadena)</i>: este método se encarga de comprobar si la cadena es un diptongo</li><li>▪ <i>Triptongo(String subcadena)</i>: este método se encarga de comprobar si la cadena es un triptongo</li><li>▪ <i>SubCadena(int i, String texto)</i>: devuelve el carácter de la posición i de la cadena texto</li><li>▪ <i>DivideSilabas()</i>: este método se encarga de dividir el texto escrito hasta el momento en una serie de sílabas</li><li>▪ <i>Reemplazar(String aux)</i>: este método se encarga de reemplazar aux por otro/s caracteres para poder ser reproducibles</li><li>▪ <i>Preprocesado(Vector listaSilabas)</i>: método que elabora una lista de sonidos casi reproducibles</li><li>▪ <i>busquedaSonidos(Vector listaPreprocesada)</i>: método que elabora la lista de sonidos definitiva</li></ul>

Figura 10.31. Descripción de la Clase AdaptadorTexto.

## **CAPITULO 11: IMPLEMENTACIÓN**

La fase de implementación consiste en la traducción del diseño realizado anteriormente a un lenguaje de programación.

### **11.1. El entorno de programación y el lenguaje**

El lenguaje de programación propuesto para la realización del proyecto ha sido Java, creado por Sun Microsystems. Java es un lenguaje de programación moderno, de alto nivel y orientado a objetos. La edición de Java SDK utilizada para la creación de la aplicación ha sido J2ME (Java 2 Micro Edition), ya que está orientada a dispositivos móviles.

Para poder ejecutar aplicaciones Java en dispositivos móviles, en concreto en un Pocket PC, es necesario disponer de una máquina virtual Java. En nuestro caso, después de realizar un estudio sobre las existentes en el capítulo 7 de esta memoria, la máquina virtual elegida ha sido SuperWaba.

Para el desarrollo de la aplicación hemos utilizado el Tauschke MobileCreator, que es una plataforma para PalmOS y Windows CE. Es un IDE de desarrollo para SuperWaba.

Para que la aplicación fuera lo más configurable posible, habíamos pensado usar ficheros XML, ya que este lenguaje permite almacenar los datos con una estructura jerárquica. Esto nos planteo muchos problemas debido a que no disponíamos de la documentación relativa a la parte de XML de la máquina virtual SuperWaba ya que necesitábamos una licencia profesional.

Como una de las características más importantes del teclado es que sea configurable, para solventar el problema explicado anteriormente, nos hemos visto obligadas a utilizar ficheros .txt.

### **11.2. Utilización de ficheros**

Para el desarrollo de la aplicación se han utilizado los siguientes ficheros .txt:

- **Fichero para las frases:** este fichero contiene las frases disponibles en la aplicación, que puede variar en función de que el usuario guarde nuevas frases o elimine las existentes.
- **Ficheros para las configuraciones:** hay un fichero por cada configuración existente. En cada fichero se almacenaran datos, como el número de botones, su anchura, altura, y el texto del mismo.

Para configuración hemos utilizado dos ficheros, uno para la pantalla de primer nivel (la pantalla principal) y otro para las pantallas de segundo nivel (en las que se muestran las letras de forma independiente).

- La estructura del fichero correspondiente a la pantalla de primer nivel es la siguiente: la primera línea corresponde al número de botones que forman la pantalla, el resto del fichero está formado por tantos bloques de tres líneas como número de botones se haya indicado anteriormente. Las tres líneas de cada bloque contienen el texto que debe aparecer en el botón, la anchura y altura de dicho botón respectivamente.
- La estructura del fichero correspondiente a las pantallas de segundo nivel es la siguiente: la primera línea corresponde al número de botones que forman la pantalla, el resto del fichero está formado por tantos bloques de dos líneas como número de botones se haya indicado anteriormente. Las dos líneas de cada bloque contienen la anchura y altura de dicho botón respectivamente.

Si se desea cambiar algún elemento de alguna de las configuraciones existente, no es necesario realizar cambios en el código de la aplicación, bastaría con cambiar el dato deseado en el fichero de configuración.

### 11.3. Software usado

El software usado para la creación de la aplicación es el siguiente:

- Microsoft Windows XP Home Edition
- Tauschke MobileCreator 1.8
- SuperWaba SDK 5.61 GPL
- j2sdk 1.4.1\_07
- Emulador
- PhotoFitre
- Microsoft Office Word 2003
- Microsoft Office Power Point 2003
- REM 1.2.1
- PDF creador
- SmartDraw 7

## 11.4. Hardware empleado

- Equipo principal de desarrollo
  - S.O. Microsoft Windows XP Home Edition Service Pack 2
  - Intel Pentium M 1,73GHz
  - 1024 Mb RAM
  - HD 60Gb
  
- Equipo secundario de desarrollo
  - S.O. Microsoft Windows XP Home Edition Service Pack 2
  - Pentium IV 3,20 GHz
  - 512 Mb RAM
  - HD 200Gb
  
- PDA de prueba de la aplicación (Acer N30)
  - S.O. Windows Mobile 2003
  - Procesador Intel Xscale 266 Mhz
  - 64Mb Ram
  - 32Mb Rom
  
- Otro Hardware:
  - Impresora Epson C-62
  - Impresora HP PSC- 2650



## CAPITULO 12: PRUEBAS

En este capítulo se muestran las pruebas realizadas a la aplicación para comprobar su correcto funcionamiento. Hemos realizado la comprobación analizando los casos límite ya que son los casos en los que se podrían presentar más problemas.

### 12.1. Pruebas realizadas.

<b>Descripción</b>	<b>Comprobación del botón BORRAR</b>
<i>Acción realizada</i>	Pulsación del botón BORRAR en cualquier configuración
<i>Resultado esperado</i>	Borrar el ultimo carácter introducido
<i>Resultado obtenido</i>	Carácter borrado
<i>Observaciones</i>	Correcto

Figura 12.1. Comprobación del botón BORRAR.

<b>Descripción</b>	<b>Comprobación del botón ESPACIO</b>
<i>Acción realizada</i>	Pulsación del botón ESPACIO en cualquier configuración
<i>Resultado esperado</i>	Introducción de un espacio en el texto
<i>Resultado obtenido</i>	Espacio introducido
<i>Observaciones</i>	Correcto

Figura 12.2. Comprobación del botón ESPACIO.

<b>Descripción</b>	<b>Comprobación de inserción de un carácter</b>
<i>Acción realizada</i>	Pulsación del botón que contiene un carácter en cualquier configuración
<i>Resultado esperado</i>	Añadir carácter al texto escrito
<i>Resultado obtenido</i>	Carácter añadido
<i>Observaciones</i>	Correcto

Figura 12.3. Comprobación de inserción de un caracter.

<b>Descripción</b>	<b>Comprobación del botón MENÚ</b>
<i>Acción realizada</i>	Pulsación del botón MENÚ en cualquier configuración
<i>Resultado esperado</i>	Mostrar la pantalla menú
<i>Resultado obtenido</i>	Pantalla menú mostrada
<i>Observaciones</i>	Correcto

Figura 12.4. Comprobación del botón MENÚ.

<b>Descripción</b>	<b>Comprobación del botón REPRODUCIR</b>
<i>Acción realizada</i>	Pulsación del botón REPRODUCIR en el menú
<i>Resultado esperado</i>	Reproducir el texto escrito
<i>Resultado obtenido</i>	Texto reproducido
<i>Observaciones</i>	Correcto

Figura 12.5. Comprobación del botón REPRODUCIR.

<b>Descripción</b>	<b>Comprobación del botón EDITAR</b>
<i>Acción realizada</i>	Pulsación del botón EDITAR en el menú
<i>Resultado esperado</i>	Mostrar la pantalla editar
<i>Resultado obtenido</i>	Pantalla editar mostrada
<i>Observaciones</i>	Correcto

Figura 12.6. Comprobación del botón EDITAR.



<b>Descripción</b>	<b>Comprobación del botón BORRAR PALABRA</b>
<i>Acción realizada</i>	Pulsación del botón BORRAR PALABRA en la pantalla editar
<i>Resultado esperado</i>	Borrar la última palabra del texto escrito
<i>Resultado obtenido</i>	Última palabra borrada
<i>Observaciones</i>	Correcto

Figura 12.7. Comprobación del botón BORRAR PALABRA.

<b>Descripción</b>	<b>Comprobación del botón BORRAR TODO</b>
<i>Acción realizada</i>	Pulsación del botón BORRAR TODO en la pantalla editar
<i>Resultado esperado</i>	Borrar por completo el texto introducido
<i>Resultado obtenido</i>	Texto borrado
<i>Observaciones</i>	Correcto

Figura 12.8. Comprobación del botón BORRAR TODO.

<b>Descripción</b>	<b>Comprobación del botón FRASES</b>
<i>Acción realizada</i>	Pulsación del botón FRASES en el menú
<i>Resultado esperado</i>	Mostrar la pantalla frases
<i>Resultado obtenido</i>	Pantalla frases mostrada
<i>Observaciones</i>	Correcto

Figura 12.9. Comprobación del botón FRASES.

<b>Descripción</b>	<b>Comprobación del botón INSERTAR en la pantalla frases</b>
<i>Acción realizada</i>	Pulsación del botón INSERTAR en la pantalla frases
<i>Resultado esperado</i>	Insertar la frases seleccionada en el texto introducido
<i>Resultado obtenido</i>	Frase insertada en el texto
<i>Observaciones</i>	Correcto

Figura 12.10. Comprobación del botón INSERTAR en la pantalla frases.

<b>Descripción</b>	<b>Comprobación del botón BORRAR en la pantalla frases</b>
<i>Acción realizada</i>	Pulsación del botón BORRAR en la pantalla frases, después de haber seleccionado la frase seleccionada mediante la barra de desplazamiento
<i>Resultado esperado</i>	Borrar la frase seleccionada de la lista
<i>Resultado obtenido</i>	Frase borrada
<i>Observaciones</i>	Correcto

Figura 12.11. Comprobación del botón BORRAR en la pantalla frases.

<b>Descripción</b>	<b>Comprobación del botón GUARDAR en la pantalla frases</b>
<i>Acción realizada</i>	Pulsación del botón GUARDAR en la pantalla frases
<i>Resultado esperado</i>	Guarda la frase seleccionada al final de la lista
<i>Resultado obtenido</i>	Frase guardada
<i>Observaciones</i>	Correcto

Figura 12.12. Comprobación del botón GUARDAR en la pantalla frases.

<b>Descripción</b>	<b>Comprobación del botón COLORES</b>
<i>Acción realizada</i>	Pulsación del botón COLORES en la pantalla menú
<i>Resultado esperado</i>	Mostrar los distintos elementos de la aplicación a los que puede cambiarse el color
<i>Resultado obtenido</i>	Pantalla con las distintas opciones mostrada
<i>Observaciones</i>	Correcto

Figura 12.13. Comprobación del botón COLORES.

<b>Descripción</b>	<b>Comprobación del botón CAMBIAR COLOR BOTON</b>
<i>Acción realizada</i>	Pulsación del botón CAMBIAR COLOR BOTÓN de la pantalla colores
<i>Resultado esperado</i>	Mostrar la pantalla con los colores disponibles
<i>Resultado obtenido</i>	Pantalla mostrada
<i>Observaciones</i>	Correcto

Figura 12.14. Comprobación del botón CAMBIAR COLOR BOTON.

<b>Descripción</b>	<b>Comprobación del botón CAMBIAR COLOR LETRA</b>
<i>Acción realizada</i>	Pulsación del botón CAMBIAR COLOR LETRA de la pantalla colores
<i>Resultado esperado</i>	Mostrar la pantalla con los colores disponibles
<i>Resultado obtenido</i>	Pantalla mostrada
<i>Observaciones</i>	Correcto

Figura 12.15. Comprobación del botón CAMBIAR COLOR LETRA.

<b>Descripción</b>	<b>Comprobación del botón CAMBIAR COLOR FONDO</b>
<i>Acción realizada</i>	Pulsación del botón CAMBIAR COLOR FONDO de la pantalla colores
<i>Resultado esperado</i>	Mostrar la pantalla con los colores disponibles
<i>Resultado obtenido</i>	Pantalla mostrada
<i>Observaciones</i>	Correcto

Figura 12.16. Comprobación del botón CAMBIAR COLOR FONDO

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

Figura 12.17. Comprobación de la pantalla con los colores disponibles.

<b>Descripción</b>	<b>Comprobación del botón CONFIGURACIÓN</b>
<i>Acción realizada</i>	Pulsación del botón CONFIGURACIÓN en la pantalla menú
<i>Resultado esperado</i>	Mostrar configuraciones disponibles
<i>Resultado obtenido</i>	Configuraciones mostradas
<i>Observaciones</i>	Correcto

Figura 12.18. Comprobación del botón CONFIGURACIÓN.

<b>Descripción</b>	<b>Comprobación del botón METODO DE LETRAS AGRUPADAS</b>
<i>Acción realizada</i>	Pulsación del botón METODO DE LETRAS AGRUPADAS de la pantalla configuración
<i>Resultado esperado</i>	Cambiar la configuración actual al método de letras agrupadas
<i>Resultado obtenido</i>	Configuración cambiada
<i>Observaciones</i>	Correcto

Figura 12.19. Comprobación del botón METODO DE LETRAS AGRUPADAS.

<b>Descripción</b>	<b>Comprobación del botón METODO DE LAS VOCALES</b>
<i>Acción realizada</i>	Pulsación del botón METODO DE LAS VOCALES de la pantalla configuración
<i>Resultado esperado</i>	Cambiar la configuración actual al método de las vocales
<i>Resultado obtenido</i>	Configuración cambiada
<i>Observaciones</i>	Correcto

Figura 12.20. Comprobación del botón METODO DE LAS VOCALES.

<b>Descripción</b>	<b>Comprobación del botón AYUDA</b>
<i>Acción realizada</i>	Pulsación del botón AYUDA en la pantalla menú
<i>Resultado esperado</i>	Mostrar temas de ayuda
<i>Resultado obtenido</i>	Temas de ayuda mostrados
<i>Observaciones</i>	Correcto

Figura 12.21. Comprobación del botón AYUDA.

<b>Descripción</b>	<b>Comprobación de los botones de los distintos temas de ayuda</b>
<i>Acción realizada</i>	Pulsación del tema de ayuda deseado
<i>Resultado esperado</i>	Mostrar ayuda
<i>Resultado obtenido</i>	Ayuda mostrada
<i>Observaciones</i>	Correcto

Figura 12.22. Comprobación de los botones de los distintos temas de ayuda.

<b>Descripción</b>	<b>Comprobación del botón SALIR</b>
<i>Acción realizada</i>	Pulsación del botón SALIR en la pantalla menú
<i>Resultado esperado</i>	Cerrar la aplicación
<i>Resultado obtenido</i>	Aplicación cerrada
<i>Observaciones</i>	Correcto

Figura 12.23. Comprobación del botón SALIR.

<b>Descripción</b>	<b>Comprobación del botón VOLVER</b>
<i>Acción realizada</i>	Pulsación del botón VOLVER en la pantalla menú
<i>Resultado esperado</i>	Volver a la pantalla anterior
<i>Resultado obtenido</i>	Pantalla anterior mostrada
<i>Observaciones</i>	Correcto

Figura 12.24. Comprobación del botón VOLVER.

## 12.2. Análisis de casos límite

Descripción	Comprobación del botón BORRAR cuando no hay texto introducido
<i>Acción realizada</i>	Pulsación del botón BORRAR en la pantalla principal
<i>Resultado esperado</i>	No realizar ninguna acción
<i>Resultado obtenido</i>	Excepción
<i>Observaciones</i>	Incorrecto
<i>Modo de actuación</i>	Modificado y corregido en la última versión de la aplicación

Figura 12.25. Comprobación del botón BORRAR cuando no hay texto introducido.

Descripción	Comprobación del botón BORRAR PALABRA cuando no hay texto introducido
<i>Acción realizada</i>	Pulsación del botón BORRAR PALABRA en la pantalla editar
<i>Resultado esperado</i>	No realizar ninguna acción
<i>Resultado obtenido</i>	Ninguna acción realizada
<i>Observaciones</i>	Correcto

Figura 12.26. Comprobación del botón BORRAR PALABRA cuando no hay texto introducido.

Descripción	Comprobación del botón BORRAR TODO cuando no hay texto introducido
<i>Acción realizada</i>	Pulsación del botón BORRAR TODO en la pantalla editar
<i>Resultado esperado</i>	No realizar ninguna acción
<i>Resultado obtenido</i>	Ninguna acción realizada
<i>Observaciones</i>	Correcto

Figura 12.27. Comprobación del botón BORRAR TODO cuando no hay texto introducido.

<b>Descripción</b>	<b>Comprobación del botón REPRODUCIR cuando no hay texto introducido</b>
<i>Acción realizada</i>	Pulsación del botón REPRODUCIR en la pantalla menú y/o editar
<i>Resultado esperado</i>	No realizar ninguna acción
<i>Resultado obtenido</i>	Ninguna acción realizada
<i>Observaciones</i>	Correcto

Figura 12.28. Comprobación del botón REPRODUCIR cuando no hay texto introducido.

<b>Descripción</b>	<b>Comprobación del botón INSERTAR en la pantalla frases cuando no hay frase seleccionada</b>
<i>Acción realizada</i>	Pulsación del botón INSERTAR cuando no hay frase seleccionada
<i>Resultado esperado</i>	No realizar ninguna acción
<i>Resultado obtenido</i>	Vuelve a la pantalla principal
<i>Observaciones</i>	Incorrecto
<i>Modo de Actuación</i>	Modificado y corregido en la última versión de la aplicación

Figura 12.29. Comprobación del botón INSERTAR en la pantalla frases cuando no hay frase seleccionada.

<b>Descripción</b>	<b>Comprobación del botón BORRAR en la pantalla frases cuando no hay frase seleccionada</b>
<i>Acción realizada</i>	Pulsación del botón BORRAR cuando no hay frase seleccionada
<i>Resultado esperado</i>	No realizar ninguna acción
<i>Resultado obtenido</i>	Ninguna acción realizada
<i>Observaciones</i>	Correcto

Figura 12.30. Comprobación del botón BORRAR en la pantalla frases cuando no hay frase seleccionada.

<b>Descripción</b>	<b>Comprobación del botón GUARDAR en la pantalla frases cuando no hay frase seleccionada</b>
<i>Acción realizada</i>	Pulsación del botón GUARDAR cuando no hay texto escrito
<i>Resultado esperado</i>	No realizar ninguna acción
<i>Resultado obtenido</i>	Inserta frases en blanco
<i>Observaciones</i>	Incorrecto
<i>Modo de actuación</i>	Modificado y corregido en la última versión de la aplicación

Figura 12.31. Comprobación del botón GUARDAR en la pantalla frases cuando no hay frase seleccionada.







# **PARTE V:**

## **- Manual de - - Usuario -**



## **CAPÍTULO 13: MANUAL DE USUARIO**

### **13.1. Descripción de la aplicación**

Esta aplicación consiste en un comunicador que permite la introducción de texto.

La aplicación dispone de configuraciones de tipo Caracteres. Las configuraciones se diferencian en el número de botones que las componen. Éstas se caracterizan porque los botones tienen asociados un conjunto de caracteres que se pueden distribuir en una nueva pantalla de forma independiente.

La aplicación va a permitir realizar las siguientes opciones:

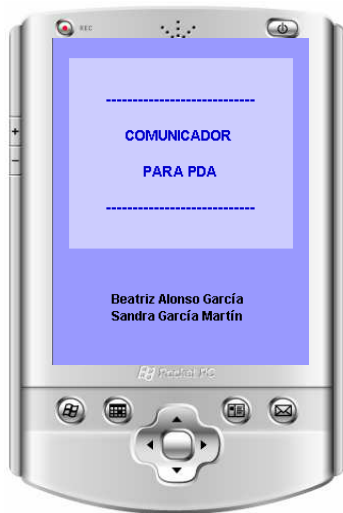
- Escribir texto mediante varios métodos de escritura.
- Borrar el texto escrito
- Reproducir el texto
- Editar el texto escrito
- Disponer de frases almacenadas para agilizar la comunicación

A continuación se detalla el uso y funcionamiento de la aplicación para un eficaz manejo.

### **13.2. Instalación de la aplicación**

Para instalar la aplicación directamente en la PDA, primeramente se conectará la PDA con el PC mediante el programa ActiveSync. Seguidamente, habrá que instalar la Máquina Virtual SuperWaba en la PDA, para ello se debe ejecutar el archivo CEinstall que se encuentra en SuperWabaSDK\bin\installers\vm\ce. Posteriormente hay que ejecutar desde el PC el archivo Comunicador\_Install y copiar el fichero Comunicador en la carpeta SuperWaba de la PDA (generada al instalar la Máquina Virtual) y ejecutarle desde ésta.

### 13.3. Pantalla de bienvenida



Es la pantalla inicial y aparece al arrancar la aplicación. En ella se muestra el título de la aplicación y el nombre de los autores. Esta pantalla se muestra durante unos segundos tras los cuales aparece la pantalla principal de la aplicación.

Figura13.1. Pantalla de Bienvenida.

### 13.4. Pantalla principal

El uso de la aplicación se basa en esta pantalla (método de escritura) y en el uso de todos los botones que aquí se muestran y que se detallan más adelante.



Figura 13.2. Pantalla Principal.



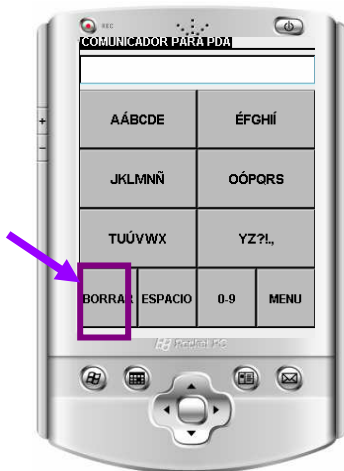
- **Texto introducido:** En esta parte de la pantalla se muestra el texto escrito hasta el momento.

Figura 13.3. Pantalla de introducción de texto.

- **Botones de las letras:** Estos botones serán empleados como teclado para poder escribir el texto deseado. Deberá pulsar sobre el botón donde se encuentra la letra que usted deseará escribir



Figura 13.4. Botones de letras.



- **Botón borrar:** Al pulsar este botón, se borrara el último carácter introducido.

Figura 13.5. Botón Borrar.

- **Botón espacio:** Cuando se pulsa este botón se introduce un espacio en el texto escrito hasta el momento.



Figura 13.6. Botón espacio.



Figura 13.7. Botón números.

- **Botón de los números:** Al pulsar este botón, saldrá una nueva pantalla con los números del 0 al 9 y la coma decimal para que usted pueda introducir números en el texto escrito, así como el botón borrar que le permitirá eliminar dichos números en caso de equivocación.

- **Botón Menú:** Al pulsar dicho botón, se despliega el menú principal de la aplicación con las diferentes opciones.



Figura 13.8. Botón menú.



## 13.5. Métodos de escritura

### 13.5.1. Método de las letras agrupadas

Supongamos que se desea escribir la palabra *casa*. Tendremos como situación de inicio la parte superior de la pantalla en blanco, ya que todavía no hemos introducido ninguna letra.



Para escribir una letra hay que apretar el botón que contiene la letra elegida, en nuestro caso, si queremos escribir la palabra *casa* tendremos que pulsar en el botón que contenga la *c*, como se muestra en la figura 13.9.

Figura 13.9. Ejemplo letras agrupadas.

En el momento que se pulsa el botón deseado, todas las letras se redistribuirán en una nueva pantalla, de manera que cada letra aparece en un botón diferente.

En esta nueva pantalla, usted debería pulsar la letra deseada, en este caso la *c*, volviendo automáticamente a la pantalla principal en la que se mostrará el texto introducido.

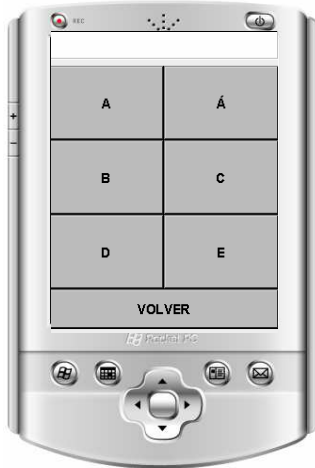


Figura 13.10. Ejemplo letras agrupadas.

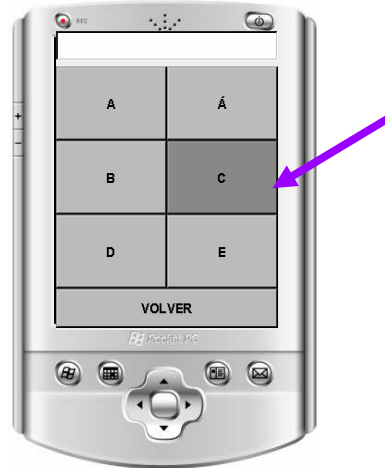


Figura 13.11. Ejemplo letras agrupadas.



Figura 13.12: Ejemplo letras agrupadas.

Ahora habría que hacer lo mismo con el resto de las letras:

- Pulsar en el grupo donde esté situada la letra *a*.
- Las letras se distribuirán como en el caso anterior.
- Repetir con el resto de letras del texto a escribir

### 13.5.2. Método de las vocales

Éste método se puede considerar una optimización del anterior, ya que las vocales están en botones separados de las consonantes de manera que el número de pulsaciones disminuye considerablemente, puesto que en una palabra por cada consonante aproximadamente hay una vocal.

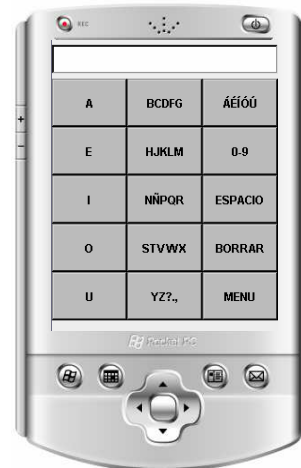


Figura 13.13: Método de las vocales.

Veamos como es su uso, utilizando el mismo ejemplo que con el método anterior, es decir la escritura de la palabra *casa*.

La primera letra que tenemos que escribir es la *c*, por lo tanto tendremos que pulsar en el grupo de letras entre las que se encuentra. Al igual que en el método anterior, la pulsación provocará que las letras correspondientes a ese botón se redistribuyan en una nueva pantalla.



Figura 13.14: Ejemplo método de vocales

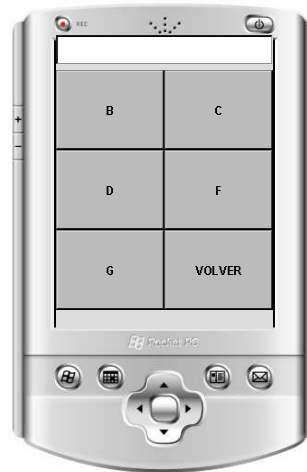


Figura 13.15: Ejemplo método de vocales

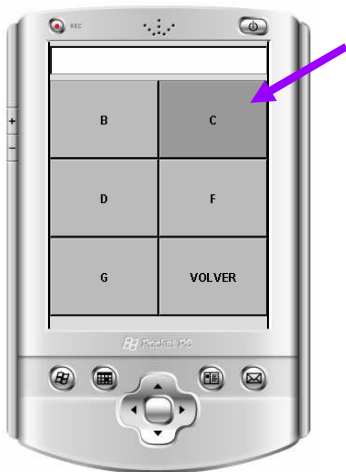


Figura 13.16: Ejemplo método de vocales



Figura 13.17: Ejemplo método de vocales

Lo siguiente sería pulsar la letra *a* y así sucesivamente hasta completar la palabra. La función de los botones Espacio, Menú, Borrar y 0-9 es la misma que en el método de escritura anterior.

### 13.6. Pantalla números

En la pantalla principal de los dos métodos de escritura existe un botón cuya funcionalidad es 0-9. Si se pulsa este botón aparecerá una nueva pantalla, en la que se muestran los números de 0 al 9, la coma decimal y se ofrece la posibilidad de borrar.

Presenta una diferencia con respecto a la pantalla de caracteres, ya que al pulsar un botón se permanece en esta pantalla. Para acceder a la pantalla principal, se debe pulsar el botón VOLVER.



Figura 13.18: Pantalla números

### 13.7. Pantalla menú

En el menú opciones se encuentran las diferentes funcionalidades que ofrece la aplicación para tratar el texto escrito, permitiendo el cambio de configuración, la inserción y almacenamiento de frases para agilizar la comunicación o el abandono de la aplicación, entre otras.



Figura 13.19. Pantalla menú.

Podemos hacer uso de sus opciones en cualquier configuración, y se accede a el pulsando el botón MENU que se encuentra en la Pantalla Principal

Todas las pantallas a las que se accede desde la Pantalla Menú disponen de un botón VOLVER desde el que se puede regresar de nuevo al MENU.

A continuación se describe cada una de las funcionalidades que ofrece esta pantalla.

- **Botón Reproducir**

El Botón Reproducir se encarga de reproducir en sonido el texto que se haya escrito hasta el momento de manera que, aunque no se pueda visualizar todo el texto escrito porque el cuadro de texto limita su aparición, éste será reproducido en su totalidad.



Figura 13.20. Botón reproducir.



Figura 13.21. Botón editar.

- **Botón Editar**

En esta pantalla se trata y edita el texto escrito hasta el momento.

- **Botón Frases**

Permite insertar en el texto escrito las frases guardadas en el dispositivo para agilizar la comunicación, así como introducir frases nuevas o borrar las existentes.



Figura 13.22. Botón frases.



Figura13.23. Botón colores.

- **Botón Colores**

Este botón muestra una pantalla en la que se pueden elegir el elemento al que usted quiera cambiar el color, ya sea el color de los botones, de la letra o del fondo.



Figura13.24. Botón configuración.

- **Botón Configuración**

Este botón muestra una pantalla en la que se puede elegir la configuración deseada. El sistema dispone de dos configuraciones.



Figura13.25. Botón Ayuda.

- **Botón Ayuda**

Este botón muestra una pantalla en la que se muestran los diferentes tipos de ayuda para facilitar el uso de la aplicación.

- **Botón Salir**

Este botón permite finalizar la ejecución de la aplicación.



Figura 13.26. Botón Salir.



- **Botón Volver**

Este botón permite volver a la pantalla anterior.

Figura 13.27: Botón Volver.

## 13.8. Pantalla editar

El menú Editar Texto es el encargado de tratar los textos escritos. Contiene una caja de texto en la que se puede visualizar el texto escrito.



Figura 13.28: Pantalla Editar.

Botones y sus funciones:

- **Botón Reproducir:** Reproduce el texto escrito hasta el momento.
- **Botón Borrar Palabra:** Borra la última palabra escrita.
- **Botón Borrar Todo:** Borra el texto escrito en su totalidad quedando la caja de texto vacía para poder escribir de nuevo.
- **Botón Volver:** Vuelve a la pantalla anterior, en este caso la Pantalla Menú.

## 13.9. Pantalla frases

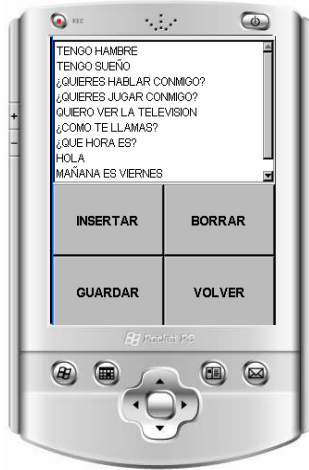


Figura13.29. Pantalla Frases.

El menú frases dispone de un conjunto de frases que permiten agilizar la comunicación permitiendo insertar frases almacenadas en el texto que estamos escribiendo, almacenar frases nuevas para poder usarlas posteriormente o borrar las existentes.

Esta pantalla contiene una lista con las frases disponibles. Esta lista dispone de una barra de desplazamiento que permite recorrer las frases almacenadas. Para escoger la frase deseada se deberá pulsar sobre ella.

Botones y sus funciones:

- **Botón Insertar:** Para usar en cualquier momento una frase de la lista, deberá seleccionar la frase que desee y pulsar el botón insertar. Esta frase se insertará al final del texto escrito.
- **Botón Borrar:** Si no considera necesaria alguna frase de la lista, debe seleccionar dicha frases y posteriormente pulsar el botón Borrar. De esta manera, la frase se eliminará permanentemente de la lista.
- **Botón Guardar:** Si desea almacenar una frase que utilice frecuentemente, deberá pulsar el botón Guardar y el texto que lleve escrito hasta el momento se almacenará al final de la lista.
- **Botón Volver:** vuelve a la pantalla anterior.

## 13.10. Pantalla colores

El menú Colores permite cambiar el color de los diferentes elementos de la aplicación.



Botones y sus funciones:

- **Botón Cambiar Color De Botones**
- **Botón Cambiar Color De La Letra**
- **Botón Cambiar Color De Fondo**
- **Botón Volver**

Deberá seleccionar el elemento al cual desea cambiarle el color y en este momento se desplegará una pantalla con los colores disponibles. Deberá seleccionar el color que desee y automáticamente se cambiará en toda la aplicación. Esta pantalla dispone también de un botón Volver por si se ha entrado erróneamente en ella.



Figura13.30. Pantalla Colores.

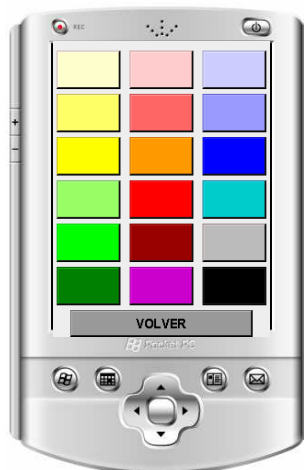


Figura 13.31. Pantalla con lista de colores.

### 13.11.Pantalla configuración

El menú Configuración permite elegir entre las dos configuraciones disponibles, éstas son:

- Método de las letras agrupadas
- Método de las vocales

Para ello tienes que pulsar en el botón configuración, y posteriormente sobre una de las configuraciones mencionadas. Al pulsar sobre el método deseado, automáticamente te situará en la pantalla principal con la configuración elegida.



Figura 13.32. Pantalla Configuración.

## 13.12. Pantalla ayuda



El menú ayuda muestra clasificados los diferentes temas de ayuda. Cada uno de estos temas despliega una nueva pantalla con nuevos temas de ayuda más precisos o bien una pantalla que contiene la información del tema de ayuda escogido.

Figura 13.33. Pantalla Ayuda.

Podemos observar un ejemplo en el que al pulsar botón Colores de la pantalla Ayuda, se despliega una nueva pantalla en la que se ofrecen temas de ayuda relacionados con los elementos de la aplicación a los que podemos cambiarles el color. Finalmente, al seleccionar el tema deseado, en el ejemplo "Color De Botones", aparece una pantalla en la que se detalla su funcionamiento.



Figura13.34: Ejemplo de ayuda



Figura13.35: Ejemplo de ayuda



Figura13.36: Ejemplo de ayuda.





# **PARTE VI:**

## **- Conclusiones -**



## **CAPÍTULO 14: CONCLUSIONES**

### **14.1. Objetivos alcanzados**

- Hemos conseguido realizar una aplicación que puede ser útil para personas con parálisis cerebral, y en general, a las personas que tengan grandes dificultades físicas y sensoriales de comunicación con las personas de su entorno.
- Hemos conseguido que la aplicación sea portable, ya que el lenguaje de programación utilizado ha sido Java, y éste es multiplataforma, aunque solo ha sido probada en Pocket PC.
- Hemos logrado que el texto introducido en el comunicador para PDA pueda reproducirse gracias al sintetizador de voz reutilizado permitiendo una comunicación más natural.
- Hemos conseguido que el comunicador sea configurable, permitiendo al usuario el cambio de configuración en función de sus necesidades.
- Hemos logrado desarrollar una aplicación atractiva, no solo para personas con discapacidad, también puede ser utilizada por niños de corta edad para facilitar el aprendizaje de la escritura.

### **14.2. Conclusiones de tipo técnico**

#### **14.2.1. Ventajas**

- Nos hemos familiarizado con los dispositivos móviles y con este tipo de tecnología, en concreto con la PDA, que al principio era ajena a nosotras. Sabemos que es un campo que está evolucionando muy rápidamente, por lo que en un futuro puede sernos de gran ayuda de cara a un mercado laboral.
- Hemos adquirido conocimientos del lenguaje Java, en concreto su uso para dispositivos móviles. Para ello, hemos realizado un estudio de las diferentes máquinas virtuales Java

para Pocket PC, adquiriendo conocimientos en la máquina virtual SuperWaba, que es la que hemos usado para realizar la aplicación.

- Hemos conseguido aplicar y desarrollar los conocimientos que hemos adquirido durante el transcurso de la carrera en un entorno real, como puede ser el uso de UML.

### **14.2.2. Inconvenientes**

- En la realización del estudio de las máquinas virtuales Java para Pocket PC, nos hemos encontrado con el problema de que la información referente a éstas es escasa.
- A la hora de realizar la parte de configuraciones, no hemos podido utilizar ficheros XML como teníamos pensado inicialmente, ya que la máquina virtual SuperWaba no dispone de algunas clases así como de la documentación relativa a esta parte, debido a que existe otro paquete de uso profesional en el que hacía falta obtener una licencia. Esto fue el principal problema para nosotras, ya que inicialmente en la información obtenida sobre esta máquina virtual aparecía como gratuita, y no descubrimos este inconveniente hasta el desarrollo de las configuraciones de la aplicación.



## **CAPÍTULO 15: POSIBLES MEJORAS**

En este apartado se verán las posibles mejoras que podrían hacerse sobre el producto software así como sobre el proyecto en general:

- Ofrecer la posibilidad de incorporar una nueva configuración con imágenes pertenecientes al sistema de comunicación SPC (Símbolos pictográficos para la comunicación) dando a la aplicación una apariencia más atractiva.
- Añadir un componente en el comunicador que permita la comunicación basada en bluetooth o wifi de la PDA con otra PDA o incluso con un PC con la finalidad de que sea posible el envío de mensajes entre los dos dispositivos.
- Mejora de la sintetización de voz.
- Introducir en el teclado un sistema de barrido que facilite la selección de los botones a los usuarios.
- Usar un mecanismo de predicción de texto, que consiste en la creación de una base de datos con las palabras más usadas en el castellano.
- En la pantalla editar que puedas moverte a cualquier carácter del texto escrito tal como se realiza en los móviles.
- Integrar en el comunicador nuevos componentes que aumenten la funcionalidad del mismo, como por ejemplo un componente que permita al usuario final crear sus propias configuraciones.



# **APÉNDICES**



## APÉNDICE A. INSTALACIÓN DE LA MAQUINA VIRTUAL SUPERWABA

### Instalación de la Máquina Virtual SuperWaba en el IDE de desarrollo utilizado (Tauschke MobileCreator Personal)

Primeramente habrá que descargar el SDK de SuperWaba de la página Web [www.superwaba.com.br](http://www.superwaba.com.br) en la sección Downloads. Habrá que descargar el Software Development Kit. El archivo obtenido habrá que descomprimirlo en una carpeta llamada SuperWabaSDK.

Será necesario instalar el JDK de Java Sun Microsystems de la página Web [java.sun.com](http://java.sun.com).

Habrá que instalar también el IDE de desarrollo siguiendo los pasos necesarios. Una vez hecho esto, aparecerá la pantalla Setup Wizard que se muestra en la Figura A.1.



Figura A.1: Paso 1 de la configuración de SuperWaba.

Se hará click en *Next* para empezar la configuración. Aparecerá la pantalla mostrada en la Figura A.2 en la que habrá que especificar la ruta de acceso del SDK de SuperWaba.



Figura A.2. Paso 2 de la configuración de SuperWaba.

En la siguiente pantalla, la Figura A.3, habrá que indicar la dirección del JDK a utilizar. Las siguientes pantallas son opcionales y seguiremos con la configuración hasta finalizar. En este momento el IDE de desarrollo ya estará listo para comenzar a trabajar con SuperWaba.

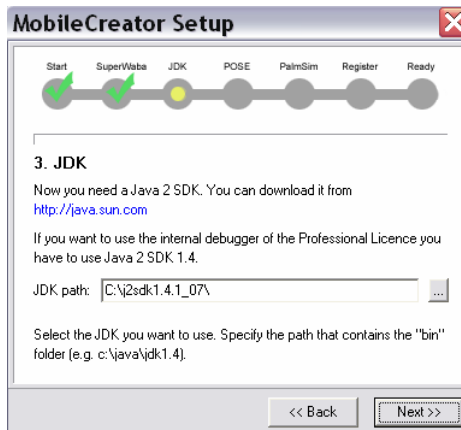


Figura A.3. Paso 3 de la configuración de SuperWaba.

### **Instalación de la Máquina Virtual SuperWaba en el Pocket PC**

Basta con ejecutar el archivo CEinstall que se encuentra en SuperWabaSDK\bin\installers\vm\ce.

## **APÉNDICE B. CONTENIDO DEL CD-ROM.**

Se enumera a continuación los contenidos del CD-ROM que acompaña a esta documentación:

- **Directorio Instalación:** Contiene los archivos necesarios para instalar la aplicación en el dispositivo. Para ello, primeramente se conectará la PDA con el PC mediante el programa ActiveSync. Seguidamente, habrá que instalar la Máquina Virtual SuperWaba en la PDA, para ello se debe ejecutar el archivo CEinstall que se encuentra en SuperWabaSDK\bin\installers\vm\ce. Posteriormente hay que ejecutar desde el PC el archivo Comunicador\_Install y copiar el fichero Comunicador en la carpeta SuperWaba de la PDA(generada al instalar la Máquina Virtual) y ejecutarle desde ésta.
- **Directorio Memoria:** Contiene la memoria en formato PDF.
- **Directorio Manual de Usuario:** Contiene el manual de usuario del comunicador en formato PDF.
- **Directorio de código fuente:** Contiene todos los archivos de código fuente generados para el desarrollo del comunicador. Aquí se incluyen también los siguientes subdirectorios:
  - Ficheros: en el que se almacenan los ficheros de frases y configuración.
  - Sonidos: en el que se almacenan los sonidos que utiliza el reproductor.
  - Ficheros Comunicador: en el que se almacenan los ficheros de código fuente generados para el desarrollo del comunicador.
  - Desarrollo: en el que se almacenan el IDE de desarrollo, así como el emulador utilizado.





## APENDICE C. REFERENCIAS

### Bibliografía

- Grady Booch, James Rumbaugh, Ivan Jacobson, “*El lenguaje unificado de desarrollo software*”, Ed. Addison Wesley.
- Grady Booch, James Rumbaugh, Ivan Jacobson, “*El lenguaje unificado de modelado*”. Ed. Addison Wesley, 2000.
- Herbert Schildt, “*La Biblia de Java 2 v5.0*”, Ed. Anaya Multimedia, 2005
- Jean-François Macary, Cédric Nicolas, “*Programación Java*” Ed. Enrolles
- John Zukowski, “*Programación en Java 2*” Ed. Anaya Multimedia, 1999

### Fuentes Web

- [es.wikipedia.org](http://es.wikipedia.org): es una enciclopedia libre
- [www.lcc.uma.es/~galvez/J2ME.html](http://www.lcc.uma.es/~galvez/J2ME.html). pagina con información sobre dispositivos móviles en Java.
- [java.sun.com](http://java.sun.com): pagina oficial de Sun Microsystems
- [www.aspace.org](http://www.aspace.org): pagina con información sobre la Parálisis Cerebral
- [www.blackdown.org](http://www.blackdown.org): pagina oficial de la Máquina Virtual Blackdown J2RE
- [www.comp.lancs.ac.uk/computing/users/fittond/ppcjava.html](http://www.comp.lancs.ac.uk/computing/users/fittond/ppcjava.html): pagina con información de las distintas Máquinas Virtuales Java para Pocket PC.
- [www.esmertec.com](http://www.esmertec.com) pagina oficial de la Máquina Virtual Jeode
- [www.ewesoft.com](http://www.ewesoft.com): pagina oficial de la Máquina Virtual Ewe
- [www.hp.com](http://www.hp.com) pagina oficial de Hewlett Packard
- [www.ibm.com/websphere](http://www.ibm.com/websphere): pagina oficial de la Máquina Virtual IBM WebSphere Studio
- [www.kadashsystems.com](http://www.kadashsystems.com) pagina oficial de la Máquina Virtual Kada
- [www.kaffe.org](http://www.kaffe.org) pagina oficial de la Máquina Virtual Kaffe
- [www.medline.plus.org](http://www.medline.plus.org): Pagina Web de Información de Salud de la Biblioteca Nacional de Medicina.
- [www.mipcdebolsillo.com](http://www.mipcdebolsillo.com): pagina Web con información general, datos, noticias, publicaciones y todo lo relacionado con los sistemas PocketPC.
- [www.mochaworks.com](http://www.mochaworks.com): pagina oficial de la Máquina Virtual SavaJe XE
- [www.nsicom.com](http://www.nsicom.com) pagina oficial de la Máquina Virtual NSICOM CrEme

- [www.pdaexpertos.com](http://www.pdaexpertos.com): Página Web con información sobre PDA.
- [www.programacion.com](http://www.programacion.com) : página con información acerca de la programación, lenguajes, implementaciones...
- [www.superwaba.com.br](http://www.superwaba.com.br): pagina oficial de la Máquina Virtual SuperWaba.
- [www.todopocketpc.com](http://www.todopocketpc.com): es una página española dedicada a los dispositivos Pocket PC, con un foro muy interesante sobre dichos dispositivos.