

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA INFORMÁTICA

**Mecanismos para el modelado y monitorización  
de actividades vinculadas a prescripción en  
educación infantil**

Realizado por

Alberto de Diego Cottinelli

Dirigido por

Beatriz Barros Blanco

Departamento de Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA

MÁLAGA, diciembre de 2008



UNIVERSIDAD DE MÁLAGA  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA  
INGENIERÍA INFORMÁTICA SUPERIOR

Reunido el tribunal examinador en el día de la fecha, constituido por:

Presidente/a D<sup>o</sup>/D<sup>a</sup>. \_\_\_\_\_

Secretario/a D<sup>o</sup>/D<sup>a</sup>. \_\_\_\_\_

Vocal D<sup>o</sup>/D<sup>a</sup>. \_\_\_\_\_

para juzgar el proyecto Fin de Carrera titulado:

**Mecanismos para el modelado y monitorización de actividades vinculadas a prescritura en educación infantil**

del alumno/a D<sup>o</sup>/D<sup>a</sup>. Alberto de Diego Cottinelli

dirigido por D<sup>o</sup>/D<sup>a</sup>. Beatriz Barros Blanco

ACORDÓ POR \_\_\_\_\_ OTORGAR LA CALIFICACIÓN DE \_\_\_\_\_  
Y PARA QUE CONSTE, SE EXTIENDE FIRMADA POR LOS COMPARECIENTES DEL  
TRIBUNAL, LA PRESENTE DILIGENCIA.

Málaga, a \_\_\_\_ de \_\_\_\_\_ del 200\_

El/La Presidente/a

El/La Secretario/a

El/La Vocal

Fdo:

Fdo:

Fdo:



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Contexto . . . . .	2
1.2. Motivación . . . . .	2
1.3. Objetivos . . . . .	3
1.4. Organización de la Memoria . . . . .	3
1.5. Material utilizado . . . . .	5
<b>2. Motricidad, escritura infantil y tecnología</b>	<b>7</b>
2.1. Motricidad . . . . .	7
2.1.1. Motricidad gruesa o global . . . . .	8
2.1.2. Motricidad fina . . . . .	8
2.2. Fases y edades de la escritura . . . . .	8
2.2.1. Desarrollo grafomotriz . . . . .	8
2.2.2. Aprendizaje de la escritura . . . . .	9
2.3. Las disgrafías . . . . .	10
2.4. La preescritura . . . . .	12
2.5. Sistemas Informáticos de ayuda al aprendizaje de la escritura . . . . .	13
<b>3. Modelado</b>	<b>17</b>
3.1. Introducción . . . . .	17
3.2. Modelo de datos del sistema . . . . .	18
3.2.1. Modelo del alumno . . . . .	20
3.2.2. Modelo de tareas . . . . .	20
3.3. La entrada de datos . . . . .	22
3.3.1. Impresión de ejercicios . . . . .	23

---

3.3.2.	Ejercicios realizados sobre la pantalla . . . . .	23
3.3.3.	Crear un modelo . . . . .	24
3.4.	Procesado de ejercicios escaneados . . . . .	24
3.4.1.	Filtrado de la imagen . . . . .	25
3.4.2.	Detección de las marcas . . . . .	26
3.4.3.	Recolocación de la imagen . . . . .	26
3.4.4.	Extracción de subimágenes . . . . .	27
3.4.5.	Decodificación del código de barras . . . . .	27
3.5.	Extracción y vectorización de trazos . . . . .	27
3.5.1.	Extracción de los trazos . . . . .	28
3.5.2.	Reconstrucción de los trazos . . . . .	28
3.5.3.	Esqueletonización de los trazos . . . . .	29
3.5.4.	Separación y vectorización de los trazos . . . . .	29
3.6.	Corrección de los ejercicios . . . . .	29
3.6.1.	Asignación de los trazos del ejercicio a los del modelo . . . . .	29
3.6.2.	Comparación de los trazos con los del modelo . . . . .	29
3.6.3.	Cálculo de la nota de trazo . . . . .	31
3.6.4.	Ejemplo de corrección de un ejercicio . . . . .	32
3.7.	Almacenamiento de ejercicios . . . . .	33
3.8.	La salida de resultados . . . . .	33
<b>4.</b>	<b>Funcionamiento del sistema</b> . . . . .	<b>35</b>
4.1.	Barra de herramientas . . . . .	35
4.2.	Añadir un nuevo modelo . . . . .	36
4.2.1.	Propiedades que deben tener los modelos . . . . .	37
4.3.	Gestión de alumnos . . . . .	37
4.4.	Búsqueda . . . . .	38
4.4.1.	Lista de imágenes . . . . .	40
4.4.2.	Lista de modelos . . . . .	41
4.4.3.	Imprimir modelos . . . . .	42
4.4.4.	Lista de ejercicios . . . . .	43
4.4.5.	Reproductor de ejercicios . . . . .	44
4.5.	Interfaz de dibujo . . . . .	45
4.6.	Añadir ejercicios escaneados . . . . .	47
4.7.	Ventana de corrección . . . . .	48
4.8.	Gráficas de resultados . . . . .	49

<b>5. Diseño e implementación del sistema</b>	<b>51</b>
5.1. Conceptos previos . . . . .	51
5.1.1. Modelo de color RGB . . . . .	51
5.2. Descripción tecnológica . . . . .	52
5.2.1. Java . . . . .	53
5.2.2. MySQL . . . . .	54
5.2.3. MySQL Connector/JDBC . . . . .	55
5.2.4. Otros paquetes . . . . .	55
5.3. La Base de Datos . . . . .	56
5.3.1. Tabla del Alumno . . . . .	56
5.3.2. Tabla de ejercicios . . . . .	57
5.3.3. Tabla de modelo . . . . .	58
5.4. Procesado de ejercicios escaneados . . . . .	58
5.4.1. Detección de las marcas . . . . .	58
5.4.2. Filtrado de fondo . . . . .	62
5.5. Decodificación del código de barras . . . . .	62
5.5.1. Codificación EAN-13 . . . . .	63
5.5.2. Tabla de codificación . . . . .	64
5.5.3. Tabla de paridad . . . . .	64
5.5.4. Ejemplo de codificación . . . . .	64
5.6. Extracción de los trazos . . . . .	67
5.6.1. Reconstrucción de los trazos . . . . .	67
5.6.2. Esqueletonización de los trazos . . . . .	68
5.6.3. Filtrado del esqueleto . . . . .	69
5.6.4. Separación y vectorización de los trazos . . . . .	70
5.6.5. Vectorización . . . . .	71
5.7. Corrección de los ejercicios . . . . .	72
5.7.1. Asignación de subtrazos . . . . .	73
5.7.2. Cálculo de características . . . . .	73
5.8. El almacenamiento de ejercicios . . . . .	74
5.9. Arquitectura general . . . . .	76
5.10. Acceso a la base de datos . . . . .	76
5.11. La Interfaz . . . . .	78
5.11.1. Internacionalización . . . . .	78
5.11.2. Listas . . . . .	78
5.11.3. Reproducción de ejercicios . . . . .	79
5.11.4. Extracción y vectorización de trazos . . . . .	81
5.11.5. Corrección de un ejercicio . . . . .	83

<b>6. Evolución del sistema</b>	<b>87</b>
6.1. Ciclos de desarrollo . . . . .	87
6.2. Usuarios . . . . .	87
6.3. Ciclo 0 . . . . .	88
6.3.1. Características implementadas . . . . .	88
6.3.2. Objetivos . . . . .	88
6.4. Ciclo 1 . . . . .	88
6.4.1. Características implementadas . . . . .	89
6.4.2. Objetivos . . . . .	89
6.4.3. Evaluación . . . . .	89
6.4.4. Problemas detectados . . . . .	90
6.4.5. Ajustes . . . . .	90
6.5. Ciclo 2 . . . . .	90
6.5.1. Características implementadas . . . . .	91
6.5.2. Evaluación . . . . .	91
6.5.3. Objetivos . . . . .	91
6.5.4. Mejoras . . . . .	91
6.5.5. Problemas detectados . . . . .	92
6.6. Ciclo 3 . . . . .	92
6.6.1. Características implementadas . . . . .	92
6.6.2. Evaluación . . . . .	92
6.6.3. Objetivos . . . . .	92
6.6.4. Problemas detectados . . . . .	93
6.6.5. Mejoras . . . . .	93
6.7. Evaluación dispositivos de entrada . . . . .	93
6.7.1. Papel . . . . .	93
6.7.2. Tableta . . . . .	94
6.7.3. Tablet-pc . . . . .	94
6.7.4. Pizarra interactiva . . . . .	94
6.7.5. Pantalla táctil . . . . .	95
6.8. Conclusiones sobre el funcionamiento del sistema . . . . .	95
6.8.1. Detección de marcas . . . . .	95
6.8.2. Extracción de trazos . . . . .	95

---

6.8.3. Herramientas de escritura . . . . .	95
6.9. Resultados método de corrección . . . . .	96
6.9.1. Ejercicios tablet . . . . .	98
6.9.2. Ejercicios táctil . . . . .	102
6.9.3. Ejercicios escaneados . . . . .	106
6.9.4. Gráficas de los resultados . . . . .	110
<b>7. Conclusiones y futuros trabajos</b>	<b>113</b>
7.1. Resumen y aportaciones . . . . .	113
7.2. Futuros trabajos . . . . .	114



# Índice de tablas

2.1. Características de los sistemas . . . . .	16
3.1. Tabla de resultados . . . . .	33
5.1. Etiquetado de símbolos . . . . .	59
5.2. Tabla de características de las marcas . . . . .	60
5.3. Tabla de codificación EAN-13 . . . . .	65
5.4. Tabla de paridad EAN-13 . . . . .	65
5.5. Tabla de cálculo dígito verificación . . . . .	66
5.6. Construcción del código . . . . .	66
6.1. Resultados ejercicios tablet-pc (1) . . . . .	99
6.2. Resultados ejercicios tablet-pc (2) . . . . .	101
6.3. Resultados ejercicios táctil (1) . . . . .	103
6.4. Resultados ejercicios táctil (2) . . . . .	105
6.5. Resultados ejercicios escaneados (1) . . . . .	107
6.6. Resultados ejercicios escaneados (2) . . . . .	109



# Índice de figuras

1.1. Áreas de la educación infantil . . . . .	2
2.1. CoWriter . . . . .	13
2.2. Write-On . . . . .	14
2.3. Dos de los tests usados en “Remi” . . . . .	14
2.4. Captura del programa MedDraw . . . . .	15
3.1. Flujo general del sistema . . . . .	19
3.2. Modelo de un ejercicio . . . . .	21
3.3. Entrada de datos . . . . .	22
3.4. Ejercicio etiquetado con código de barras . . . . .	23
3.5. Creación de un modelo . . . . .	24
3.6. Diagrama de detección de marcas . . . . .	25
3.7. Imagen con el fondo filtrado . . . . .	26
3.8. Detección de marcas en una imagen . . . . .	27
3.9. Diagrama de filtrado . . . . .	28
3.10. Ejemplo de extracción y vectorización . . . . .	30
3.11. Corrección de un ejercicio . . . . .	32
4.1. Barra de herramientas . . . . .	36
4.2. Pantalla de modelos . . . . .	37
4.3. Listado alumno . . . . .	38
4.4. Pantalla de búsqueda . . . . .	39
4.5. Lista de imágenes . . . . .	41
4.6. Pantalla de búsqueda de modelos . . . . .	42
4.7. Pantalla de búsqueda de ejercicios . . . . .	43

---

4.8. Pantalla de reproducción de ejercicios . . . . .	44
4.9. Pantalla de realización de ejercicios . . . . .	46
4.10. Pantalla de ejercicios escaneados . . . . .	47
4.11. Ventana de corrección . . . . .	49
4.12. Gráfica de notas medias . . . . .	50
4.13. Gráfica de progreso . . . . .	50
5.1. Esquema de la base de datos . . . . .	57
5.2. Filtro de ruido aplicado a una marca . . . . .	59
5.3. Marca izquierda por encima de la derecha . . . . .	61
5.4. Marca izquierda esté por debajo de la derecha . . . . .	61
5.5. Cálculo del ángulo de rotación . . . . .	62
5.6. Histograma de un ejercicio . . . . .	63
5.7. Código de barras de un ejercicio . . . . .	63
5.8. Código de barras EAN-13 creado . . . . .	66
5.9. Extracción de los trazos . . . . .	67
5.10. Detalle reconstrucción de los trazos . . . . .	68
5.11. Reconstrucción de los trazos aplicado a un ejercicio . . . . .	68
5.12. Esqueleto de una imagen . . . . .	69
5.13. Vecindad 8 . . . . .	70
5.14. Filtrado del esqueleto de una imagen . . . . .	70
5.15. Cruce de dos trazos . . . . .	71
5.16. Proceso de vectorización . . . . .	71
5.17. Quadtree . . . . .	72
5.18. Partición de los trazos . . . . .	73
5.19. Comparación forma modelo y ejercicio . . . . .	74
5.20. Log y trazo resultante . . . . .	75
5.21. Esquema paquetes . . . . .	76
5.22. Diagrama de secuencia de la base de datos . . . . .	77
5.23. JLists . . . . .	79
5.24. Diagrama de los componentes de las listas . . . . .	80
5.25. Diagrama de clases representando los estados del ejercicio . . . . .	81
5.26. Diagrama de reproducción . . . . .	82

---

5.27. Secuencia de extracción de los trazos . . . . .	84
5.28. Secuencia de corrección . . . . .	85
6.1. Versión básica del sistema . . . . .	89
6.2. Ejercicio en papel . . . . .	93
6.3. Ejercicio en tableta . . . . .	94
6.4. Ejercicio en tablet-pc . . . . .	94
6.5. Ejercicio en pizarra . . . . .	94
6.6. Ejercicio en pantalla táctil . . . . .	95
6.7. Ejercicio hecho a lápiz . . . . .	96
6.8. Ejercicios tablet-pc (1) . . . . .	98
6.9. Ejercicios tablet-pc (2) . . . . .	100
6.10. Ejercicios pantalla táctil (1) . . . . .	102
6.11. Ejercicios pantalla táctil (2) . . . . .	104
6.12. Ejercicios pantalla táctil (1) . . . . .	106
6.13. Ejercicios escaneados (2) . . . . .	108
6.14. Media de resultados por alumnos de ejercicios de tablet-pc . . . . .	110
6.15. Media de resultados por alumnos de ejercicios de pantalla táctil . . . . .	110
6.16. Media de resultados por alumnos de ejercicios escaneados . . . . .	111



# Capítulo 1

## Introducción

El aprendizaje de la lectoescritura ha ocupado gran parte de la educación infantil siendo la preescritura una de las disciplinas encargadas de que los alumnos mejoren sus capacidades grafomotoras. Para ello es importante detectar lo antes posible desórdenes en la escritura infantil y corregirlas, facilitando así el desarrollo psicomotriz alumno. Las tareas de preescritura consisten básicamente en copiar un modelo con o sin pautas. Durante el proceso de aprendizaje el alumno realiza los ejercicios que se corresponden con su edad y la dificultad aumenta progresivamente. El profesor se encarga de observar que el alumno progrese adecuadamente, desde los primeros años en los que el niño realiza garabatos y formas geométricas simples hasta que llega a copiar letras y finalmente escribir sus propios textos.

Este proyecto está pensado para ayudar al educador en dicha tarea. Tradicionalmente, el profesor proponía una serie de ejercicios y los corregía. Dicha tarea se ha llevado siempre a cabo de forma manual. Con el nuevo sistema se pretende automatizar parte del proceso y apoyar la labor del docente, que ahora podrá monitorizar la evolución de cada alumno de manera detallada y además tener almacenados digitalmente los ejercicios.

Existen otros trabajos de software relacionados con la escritura infantil, como por ejemplo uno sobre la usabilidad del reconocimiento óptico de caracteres para la entrada de texto en un ordenador personal en niños [1]. En este proyecto se comparan distintos métodos para que los niños introduzcan datos en el ordenador. En otro trabajo se estudia un sistema para corregir automáticamente ejercicios que consisten en rellenar huecos de un texto usando un conjunto de palabras [2]. Estos trabajos centran su atención principalmente en el reconocimiento óptico de caracteres y en sus distintas aplicaciones. Existen otros enfocados a intentar detectar dificultades grafomotoras, llamadas disgrafías, a partir del estudio de diferentes dibujos y trazos realizadas por los niños, y aplicando métodos gráficos [3] e identificando características iniciales (numero de trazos, velocidad media, punto de inicio del trazo, etc. ) para poder hacer estudios posteriores [4]. En esta misma línea está orientado este proyecto, y en él se explorarán estas técnicas, entre otras, para detectar su nivel de escritura y las posibles disgrafías que tenga el niño.

## 1.1. Contexto

El presente trabajo forma parte del proyecto de investigación educativa PATIO para un colegio de educación infantil. Dicho proyecto engloba una serie de subproyectos todos ellos relacionados con la educación, que incluyen herramientas de ayuda para la realización de tareas relacionadas con el desarrollo cognitivo del niño, para el modelado de usuario y la monitorización de las sesiones de trabajo. El PFC que se propone pretende cubrir la parte de preescritura (3-4 años).

La educación infantil promueve el desarrollo integral del individuo, tanto a nivel motriz, cognitivo, lingüístico y social (Figura 1.1). Dentro de la primera, se engloba la psicomotricidad y la práctica de la escritura en papel. Es en este punto en el que se sitúa este trabajo fin de carrera.

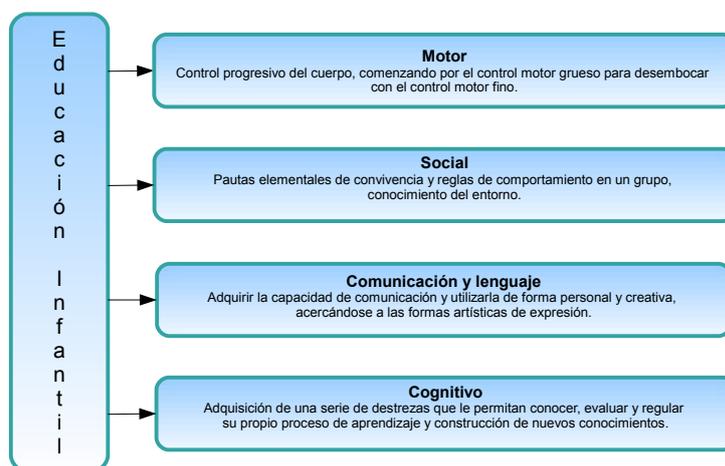


Figura 1.1: Áreas de la educación infantil

## 1.2. Motivación

Para la enseñanza de la escritura el educador infantil debe preparar una serie de ejercicios de preescritura en papel para distintos tipos de trazo que debe realizar el alumno. Mientras los alumnos realizan estos ejercicios el profesor comprueba que los hagan correctamente. Sin embargo no es posible que supervise el trabajo de todos alumnos, por lo que el proceso por el cual los alumnos llegaron a la solución del ejercicio se pierde. La cantidad de ejercicios que realizan complica la monitorización del desarrollo del alumno y el educador tampoco tiene una herramienta de apoyo que le ayude a identificar fallos en el aprendizaje de los niños. Otro

problema es que el alumno debe repetir varias veces el mismo ejercicio, por lo que hay que presentarlos de distintas formas para hacer esta tarea más agradable .

También es importante para el desarrollo psicomotriz del niño que utilice distintas herramientas para escribir y que lo haga sobre diversas superficies. El hecho de que los alumnos puedan hacer los ejercicios utilizando distintos soportes hace que esta tarea sea más atractiva.

### **1.3. Objetivos**

El propósito de este proyecto es implementar una herramienta que ayude al profesor en las tareas mencionadas anteriormente. Para ello se pretende introducir el ordenador en los procesos de preescritura y observar la influencia que tiene en la enseñanza de la escritura. El sistema intentará facilitar la tarea del educador y no suplantarla. Con este fin se plantean los siguientes subobjetivos:

1. Crear un sistema para imprimir y almacenar los ejercicios de preescritura que realicen los alumnos. Incluir los mecanismos que permitan a los profesores guardar los resultados de su corrección.
2. Implementar los métodos necesarios para que el sistema corrija automáticamente los ejercicios realizados por los alumnos en papel, cuyos resultados se compararán con las evaluaciones emitidas del profesor en sus correcciones manuales.
3. Añadir la funcionalidad necesaria para que los alumnos puedan hacer los ejercicios directamente sobre una pantalla táctil y comparar el resultado con el obtenido en los apartados anteriores.
4. Incorporar a la corrección un mecanismo que permita caracterizar los niveles de conocimiento del alumno que permitan detectar posibles disgrafías de este.
5. Definir un entorno que integre todas las funcionalidades descritas y que permita monitorizar los resultados según diferentes puntos de vista.

### **1.4. Organización de la Memoria**

Este capítulo de la memoria ha puesto de manifiesto los principales objetivos del proyecto, el contexto y la motivación del mismo. Los restantes capítulos de la memoria serán descritos a continuación:

- Motricidad, escritura infantil y tecnología

Este capítulo explica aspectos y términos importantes relacionados con la educación de la escritura y el desarrollo de los niños. Se explican los problemas que pueden tener los alumnos en el aprendizaje. Dichos conceptos son importantes para entender la finalidad de este proyecto. Finalmente se describen algunos de los programas que existen actualmente y están destinados a la educación.

- **Modelado de un sistema de apoyo a las primeras fases de preescritura infantil**

En este capítulo se explica el modelado del sistema. Esto incluye los modelos de datos que tendremos y cómo se hará la entrada de datos, que incluye la entrada de modelos ejercicios escaneados y realizados con un dispositivo táctil. También se definen el tratamiento que se hace a los ejercicios escaneados así como los procesos de corrección. Se explica como se almacenarán los ejercicios y los resultados obtenidos en la corrección automática.

- **Funcionamiento del sistema**

Este capítulo hace un recorrido por la interfaz del sistema haciendo una descripción detallada. Se explica la funcionalidad de todos los controles y los pasos a seguir para realizar varias acciones, como añadir un nuevo modelo, corregir un ejercicio, hacer un ejercicio sobre la pantalla táctil o ver los resultados de los alumnos.

- **Diseño e implementación del sistema**

En este capítulo se describen detalladamente todos los módulos de los que está compuesto el sistema. Se explican como están implementados y los paquetes de software que se han usado para ello. En los apartados de este capítulo encontramos la descripción de la base de datos, el funcionamiento detallado de la entrada de ejercicios y de cómo se realiza la corrección.

- **Evolución del sistema**

Aquí se describe la metodología que se ha seguido y las distintas evaluaciones a las que se ha sometido el sistema en las que han intervenido los profesores así como los alumnos. El proceso ha constado de varios ciclos en los que se han identificado nuevas necesidades y errores del sistema. Finalmente se hace una evaluación de los resultados obtenidos con el sistema.

- **Conclusiones y futuros trabajos**

En este capítulo se hace un resumen de los objetivos alcanzados con el proyecto así como las limitaciones que presenta el sistema. También se plantean varias mejoras del mismo y la posibilidad de extender la funcionalidad para que por ejemplo sea aplicable a alumnos de más edad.

## **1.5. Material utilizado**

Para realizar el proyecto se ha usado el siguiente material. Como entorno de programación se ha usado NetBeans. La memoria se ha escrito usando el lenguaje  $\text{\LaTeX}$  y el entorno TeXnicCenter para editar los archivos fuente de  $\text{\LaTeX}$ . Las figuras se han realizado con el programa Draw de OpenOffice y el editor de imágenes GIMP.



## Capítulo 2

# Motricidad, escritura infantil y tecnología

En este capítulo se tratarán los temas relacionados con la educación infantil en materia de escritura. En la primera sección 2.1 se explicará el concepto de motricidad, haciendo una distinción entre la grafomotricidad fina y la gruesa y explicando las diferencias. La sección 2.2 explica las fases del aprendizaje de la escritura. En la sección 2.3 se exponen los problemas que pueden tener los niños durante este proceso. Se explicará el concepto de disgrafía y los distintos tipos que hay. En la siguiente sección se enumeran las condiciones necesarias para que el niño tenga un desarrollo normal que posibilite el aprendizaje de la escritura. Finalmente se expondrán una serie de programas orientados a la educación infantil.

### 2.1. Motricidad

La motricidad es la capacidad del hombre que le permite moverse por sí mismo. Para lograr esto tiene que haber una coordinación y sincronización entre el sistema nervioso, los sentidos, el sistema muscular y los demás elementos que intervienen en el proceso.

Juan Jiménez da la siguiente definición [5]:

La motricidad es definida como el conjunto de funciones nerviosas y musculares que permiten la movilidad y coordinación de los miembros, el movimiento y la locomoción. Los movimientos se efectúan gracias a la contracción y relajación de diversos grupos de músculos. Para ello entran en funcionamiento los receptores sensoriales situados en la piel y los receptores propioceptivos de los músculos y los tendones. Estos receptores informan a los centros nerviosos de la buena marcha del movimiento o de la necesidad de modificarlo.

La motricidad puede clasificarse en motricidad gruesa y motricidad fina. La primera hace referencia a movimientos amplios como caminar, correr, saltar y la segunda a movimientos precisos que requieren cierta destreza, como la escritura.

### **2.1.1. Motricidad gruesa o global**

La motricidad gruesa se refiere al control de los movimientos musculares generales del cuerpo que llevan al niño desde la dependencia absoluta a desplazarse solos. Algunos de estos movimientos son sentarse, girar sobre sí mismo, gatear, mantenerse de pie, caminar, saltar, etc. El control motor grueso es un paso muy importante en el desarrollo de un bebé, el cual puede refinar los movimientos descontrolados, aleatorios e involuntarios a medida que su sistema neurológico madura. Y de tener un control motor grueso pasa a desarrollar un control motor fino perfeccionando los movimientos pequeños y precisos.

### **2.1.2. Motricidad fina**

Al contrario que la motricidad gruesa, la fina comprende todas las actividades que necesitan precisión y un alto nivel de coordinación. Requiere de un elevado grado de maduración y un aprendizaje largo para la dominación de cada uno de sus aspectos.

La motricidad fina resulta del proceso de refinamiento del control de la motricidad gruesa y va asociado a la maduración del sistema neurológico. Las destrezas de la motricidad fina se desarrollan a través del tiempo, de la experiencia y del conocimiento y requieren un nivel intelectual normal para poder planear y ejecutar una tarea así como fuerza muscular, coordinación y sensibilidad [6].

## **2.2. Fases y edades de la escritura**

El aprendizaje de la escritura se desarrolla de forma gradual en el niño, que necesita adquirir las suficientes destrezas motoras que le permitan sujetar el lápiz con la suficiente coordinación y precisión para poder comenzar a hacer los primeros trazos. Para ello deben cumplirse una serie de condiciones como la independencia entre los dedos de la mano dominante y la independencia funcional del brazo.

### **2.2.1. Desarrollo grafomotriz**

La habilidad para orientar y formar las letras correctamente está relacionado con el conocimiento y el control motriz. Para producir unos escritos limpios, el niño debe ser capaz de llevar a cabo movimientos motrices finos y gruesos de manera controlada. Se estima que entre el 5 y 30 por ciento de todos los niños tienen algunas dificultades con la motricidad fina que puede influir en su habilidad para escribir correctamente [7]. Esto se puede manifestar en la adición de líneas o elementos innecesarios o en letras demasiado pequeñas o grandes.

Durante el desarrollo grafomotriz del alumno se pueden distinguir tres etapas:

- En la primera etapa el niño aprende a coger el instrumento de escritura para hacer sus primeros trazos. Éstos consisten en simples garabatos sin ningún significado, es una actividad básicamente motriz.
- En la siguiente etapa se adquieren progresivamente el control visual que se asocia con el control del brazo y la mano. De esta forma se llega al establecimiento de un sistema de guía del movimiento de la mano de manera que el alumno es capaz de trazar formas.
- A partir de los cuatro años la coordinación entre mano y visión está completamente desarrollada y se puede comenzar con el aprendizaje de la escritura y la lectura.

### 2.2.2. Aprendizaje de la escritura

Los niños comienzan a escribir a temprana edad haciendo simples dibujos que representan cosas de su alrededor y haciendo formas en papel que representan letras. Cuando su habilidad de dibujar mejora y su conocimiento sobre el mundo incrementa, sus dibujos se vuelven más representativos y comienzan a etiquetarlos (“esto es una casa”). Al mismo tiempo que comienzan a nombrar sus dibujos los niños empiezan a hacer dibujos especiales (normalmente compuestos por líneas y pequeños trazos) que consideran como palabras [8]. Sobre la edad de tres años, los niños comienzan a distinguir entre palabras y dibujos pero asumen que sus palabras tienen un significado (como una lengua ideográfica), creyendo por ejemplo que las cosas grandes tienen palabras grandes [9].

En una investigación sobre las producciones escritas de niños entre cuatro y siete años, E. Ferreiro y A. Teberosky estudiaron cómo el niño construye sus propios conocimientos sobre la escritura. Describieron la existencia de determinadas fases en el aprendizaje de la escritura [10].

1. La primera fase se corresponde con el periodo en el que los niños diferencian ya la escritura del dibujo: los grafismos que tienen una semejanza icónica con su referente son identificados como dibujo, y los que no la tienen, como escritura. Ahora bien, en su intento de representación de la escritura, los niños todavía no llegan a realizar letras convencionales: es la etapa de la escritura indiferenciada.
2. En la segunda fase, utilizan un repertorio variado de grafías convencionales: es la fase de la escritura diferenciada. En ella las producciones escritas están reguladas por determinadas hipótesis que los niños manejan: linealidad, unión y discontinuidad, número mínimo de letras, variedad interna entre las mismas...
3. En la tercera, la fase silábica, los niños comienzan a establecer relaciones entre sus grafismos y los aspectos sonoros de la palabra, pero es una producción conducida por la segmentación silábica de la palabra. Identifican la sílaba, pero ésta suele ser representada mediante una sola letra. Así, por ejemplo, MARIPOSA podría ser representada por A I O A.

4. En la fase silábico-alfabética, la cuarta, los niños se dan cuenta de la existencia de correspondencias intrasilábicas, pero no son capaces de segmentar todos los elementos sonoros de la palabra; en consecuencia, en sus producciones el número de letras es inferior al número de consonantes y vocales de la palabra, porque algunas letras quedan sin reflejar.
5. En la quinta fase, los niños reconocen una correspondencia alfabética exhaustiva: a cada consonante y vocal de la palabra corresponde una letra. Se encuentran ya en la etapa alfabética; ahora bien, esto no significa que se trate necesariamente de escritura correcta en cuanto a ortografía, lo cual llegará más adelante.

### 2.3. Las disgrafías

La disgrafía es un trastorno que provoca que el alumno tenga dificultades para escribir correctamente. Estas alteraciones no son provocadas por una lesión neuronal o una deficiencia intelectual y puede presentarse en niños con adecuada estimulación ambiental. Ajuriaguerra lo define de la siguiente manera en su “Manual de Psiquiatría Infantil” [11]: “será disgráfico todo niño cuya escritura sea defectuosa, si no tiene ningún déficit neurológico o intelectual que lo justifique”. Este tipo de disgrafía se conoce como disgrafía primaria, disgrafía evolutiva o disgrafía funcional.

La disgrafía primaria la podemos clasificar en dos tipos. Cuando la disgrafía afecta a la forma de las letras se habla de disgrafía motriz o caligráfica. La disgrafía disléxica está relacionada con la omisión de letras, inversiones o uniones y separaciones de palabras.

María Nosty define estos dos tipos de disgrafías de la siguiente manera [12]:

- **Disgrafía motriz**

Se trata de trastornos psicomotores. El niño disgráfico motor comprende la relación entre sonidos los escuchados, que el mismo pronuncia perfectamente, y la representación gráfica de estos sonidos, pero encuentra dificultades en la escritura como consecuencia de una motricidad deficiente. Se manifiesta en lentitud, movimientos gráficos disociados, signos gráficos indiferenciados, manejo incorrecto del lápiz y postura inadecuada al escribir.

- **Disgrafía específica**

La dificultad para reproducir las letras o palabras no responden a un trastorno exclusivamente motor, sino a la mala percepción de las formas, a la desorientación espacial y temporal, a los trastornos de ritmo, etc., compromete a toda la motricidad fina.

A continuación se enumeran algunos de los errores que pueden ser consecuencia de la disgrafía motriz o caligráfica:

- Escritura en espejo
- Trastorno de la forma de la letra
- Trastorno del tamaño de la letra
- Deficiente espaciamiento entre las letras dentro de una palabra, entre palabras y renglones.
- Inclinação defectuosa de las palabras y renglones
- Ligamentos defectuosos de las palabras y de los renglones
- Ligamentos defectuosos entre las letras que conforman cada palabra
- Trastornos de la presión o color de la escritura, bien por exceso o por defecto.
- Trastorno de la fluidez y del ritmo escritor

Algunos de los errores propios de la disgrafía de simbolización son los siguientes:

- Omisión de las letras, sílabas o palabras
- Confusión de letras con sonido semejante
- Inversión o transposición del orden de las sílabas
- Invención de palabras
- Uniones y separaciones indebidas de sílabas, palabras o letras.
- Los textos que componen son de inferior calidad que sus pares
- Utilización de oraciones más cortas y con mayor número de errores gramaticales.

La disgrafía que está provocada por otros problemas que no son funcionales se conoce como disgrafía secundaria o adquirida. Algunos de los problemas que pueden ser el origen de este trastorno pueden ser una deficiencia intelectual, visual o motriz.

También se habla de Predisgrafía, cuando encontramos dificultades gráficas en niños menores de seis años, que podrían posteriormente presentar trastornos disgráficos.

## 2.4. La preescritura

La preescritura es una parte de la educación psicomotriz del alumno, que permite la adquisición de las destrezas motoras que son necesarias para la escritura. Para adquirir estas destrezas el alumno realiza ejercicios relacionados con la construcción de grafismos.

El objetivo directo es obtener las cualidades básicas para el aprendizaje de la escritura. Para que el alumno realice grafismos de forma correcta se deben cumplir una serie de condiciones neurológicas, perceptivas y motrices, antes de que adquiera significado y se convierta en lenguaje escrito. Así se evitan problemas posteriores como la dislexia y las disgrafías.

Siguiendo a P. Vayer [13], estas condiciones son las siguientes:

- El control neuro-muscular
- Independencia funcional del brazo
- Coordinación entre vista y mano
- Organización espacio-temporal

Para lograr una independencia de los dedos y la mano hay que partir de la independencia funcional del brazo. Así el alumno logrará progresivamente una coordinación más precisa y a partir de aquí el alumno podrá continuar con el aprendizaje de la escritura.

- Independencia de los dedos
- Visión y transcripción de la izquierda hacia la derecha
- Rotación habitual de los bucles en sentido contrario al de las agujas del reloj
- Mantenimiento correcto del útil de escritura

Durante la educación en la etapa preescolar no se contempla directamente el aprendizaje de la escritura, sin embargo si se crean las bases para poder lograr este objetivo más adelante. Para el desarrollo psicomotriz del alumno y el aprendizaje de la escritura se usan tareas de preescritura. Estos ejercicios están destinados al entrenamiento de distintos tipos de trazo y se van complicando conforme avanza la edad del alumno. Al principio los ejercicios consisten en líneas horizontales o verticales, después se sigue con trazos oblicuos, ondas hasta formar trazos más complejos. En general, van dirigidos a que el alumno adquiera las bases para realizar letras posteriormente. Se suelen integrar en dibujos para hacer el trabajo más atractivo para los niños y convertirlo de forma lúdica. El maestro deberá prestar especial atención al movimiento realizado en el trazo de las líneas, controlando aspectos tales como la dirección, la presión del lápiz, las separaciones, las inversiones, los tamaños, etc.

## 2.5. Sistemas Informáticos de ayuda al aprendizaje de la escritura

El software que se usa actualmente para el aprendizaje de la escritura está orientado a alumnos de más edad que ya componen textos. Estos paquetes incluyen una variedad de herramientas creativas que hacen que la composición sea más fácil, ya sea corrigiendo la ortografía y la gramática o ayudando en la planificación de los escritos. También pueden ayudar ofreciendo vocabulario o frases hechas que los alumnos pueden utilizar. Otras herramientas ayudan en el diagnóstico de problemas motrices de los alumnos. A continuación se explicarán algunos de estos programas.



Figura 2.1: CoWriter

CoWriter [14] es un paquete de software para ayudar a los alumnos en la composición de textos en inglés. Incorpora software que ayuda a escribir palabras averiguando la que intenta escribir el alumno buscando otras palabras que son parecidas fonéticamente. En la figura 2.1 aparece una captura en la que el programa muestra como se escribe la palabra correctamente. También ayuda en la fase de planificación del escrito ayudando a los niños a organizar sus tramas y hacer notas sobre lo que será colocado en cada sección antes de empezar con la composición.

Write-On handwriting [15] (Figura 2.2) es un programa que enseña a hacer las letras. Primero muestra al alumno una animación del recorrido y después el alumno tiene que repetirlo sobre una tableta. El software comprueba si el trazo pasa por ciertos puntos en la dirección correcta al escribir la letra, que se pueden hacer en cursiva o en imprenta.

Scriptot [16] está orientado también a la enseñanza de las letras. Para ello las reproduce en la pantalla del ordenador mostrando el recorrido para que los niños aprendan a hacerlas. Los alumnos escriben sobre una tableta gráfica y el sistema realiza un análisis de la forma de las letras para evaluarlas y examina la fluidez de los trazos haciendo un análisis cinemático.

En el artículo “Automatic analysis of the structuring of children’s drawings and writing” [3] se presenta una herramienta en la que los alumnos tienen que hacer varios tests con

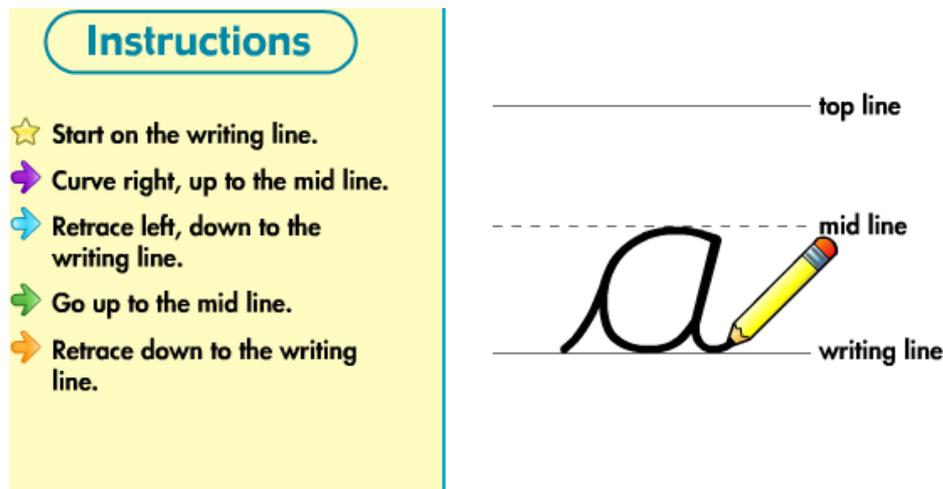


Figura 2.2: Write-On

los que se pretende identificar dificultades motoras. Estos tests incluyen la comprensión de frases escritas, tests de dibujo y de escritura. A partir de ahí se hace un análisis de algunas características de la escritura para determinar si el alumno tiene algún problema grafomotriz. En la figura 2.3 se muestran dos de los tests de dibujo.

<i>Model</i>	<i>Production (a)</i>	<i>Production (b)</i>	<i>Production (c)</i>

Figura 2.3: Dos de los tests usados en “Remi”

El programa Meddraw[17] es un sistema para el diagnóstico automático de pacientes con problemas neurológicos como dispraxia (falta de organización en el movimiento). Para lograrlo se proponen una serie de tests que consisten en la copia de dibujos que se analizarán posteriormente. Algunas de las características que analiza el programa son la aceleración, la velocidad, número de segmentos, presión etc. La captura de datos se hace también a través de una tableta de escritura digital. En la figura 2.4 se muestra una pantalla del programa con el análisis de algunos dibujos hechos.

En la tabla 2.1 se resumen las características de todos estos sistemas y proyectos. Los

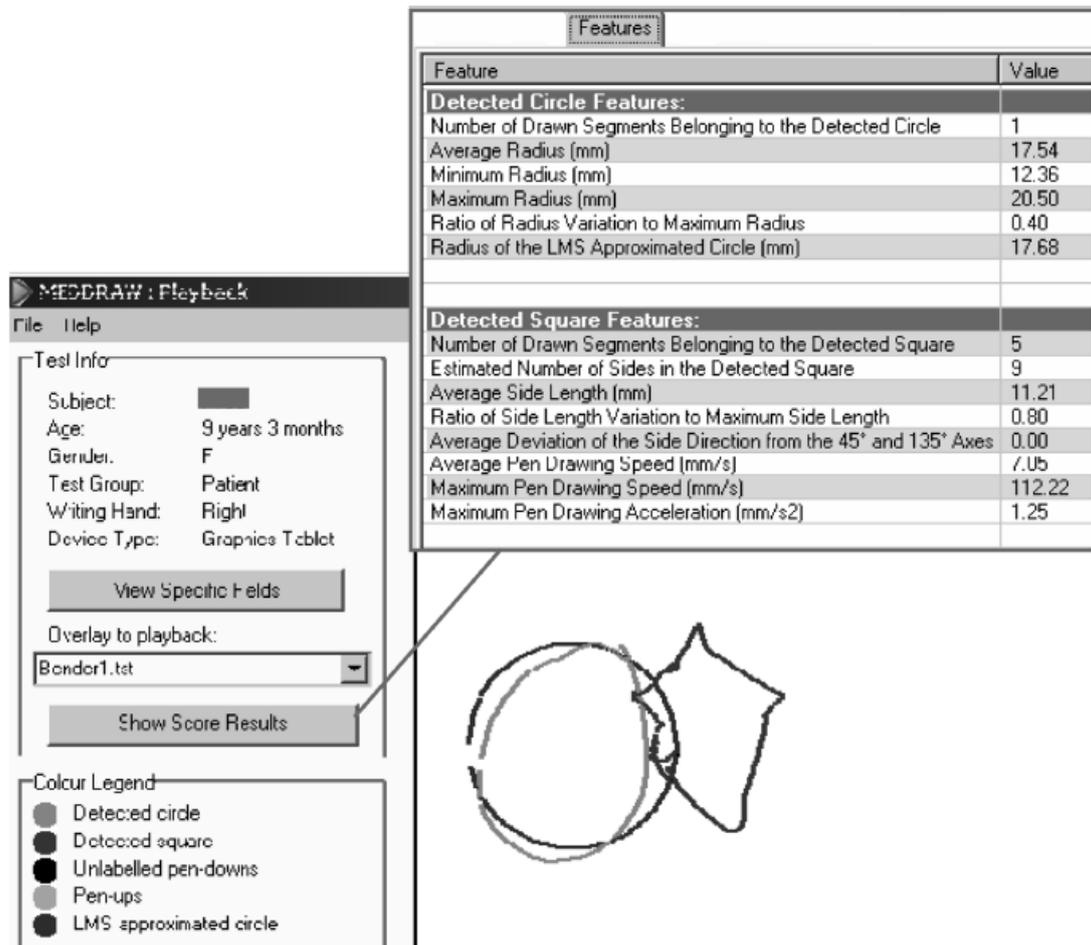


Figura 2.4: Captura del programa MedDraw

conceptos estudiados son los siguientes:

- Trazabilidad: se refiere a si el sistema es capaz de almacenar el progreso de los alumnos.
- Evaluación aut.: muestra si el programa hace algún tipo de corrección automática
- Estado: indica el estado de implementación de la herramienta (prototipo o final)
- Analiza escritura: indica si se realiza un análisis de escritura manuscrita
- Nuevos ejercicios: especifica si el sistema admite nuevos tipos de ejercicios
- Finalidad: el uso que se le da a la herramienta (educación o diagnóstico)
- Fase de la educación: la edad de los usuarios finales para los que está pensada la herramienta

	CoWriter	Write On	Remi	Scriptot	Meddraw
Trazabilidad	×	×	✓	✓	✓
Evaluación aut.	×	✓	✓	✓	✓
Implementado	✓	✓	×	×	✓
Analiza escritura	×	✓	✓	✓	✓
Nuevos ejercicios	✓	×	×	×	×
Finalidad	Edu.	Edu.	Diag.	Diag.	Diag.
Fase de la ed.	Primaria	Preesc	Todas	Preesc	Todas

Tabla 2.1: Características de los sistemas

Entre todos los sistemas podemos distinguir dos grupos, uno orientado a la composición de textos y otro al análisis de los trazos que componen la escritura para hacer diagnósticos. El presente proyecto está orientado en ésta línea, pero centrándose en la etapa de preescritura. Se pretende crear un sistema orientado a la educación de los trazos de la escritura que además evalúe automáticamente los ejercicios analizando los trazos de los alumnos y almacene su progreso. Con ayuda de los resultados obtenidos el educador podrá determinar si el alumno progresa adecuadamente.

## Capítulo 3

# Modelado de un sistema de apoyo a las primeras fases de preescritura infantil

En este capítulo se definen los mecanismos del sistema para gestionar las tareas asociadas a actividades de preescritura infantil. Se modela el flujo global de datos del sistema, desde la impresión de los ejercicios sobre papel hasta la evaluación de los mismos. Estas tareas incluyen la corrección automática de los ejercicios así como el almacenamiento de los ejercicios y el tratamiento de las imágenes escaneadas. También se explican los modelos de datos necesarios para almacenar los resultados.

Existen tres maneras de introducir ejercicios, en la primera la entrada de datos se realiza a través de el escaneado de los ejercicios típicos que realizan los alumnos. La segunda posibilidad es la de realizarlos a través de una pantalla táctil y el último caso sería en el que los ejercicios se hacen en una tableta.

En la sección 3.2 se definen los modelos de datos presentes en el sistema. A continuación se explican en las secciones 3.4 y 3.5 los mecanismos para procesar y posicionar los ejercicios correctamente mediante el uso de marcas y extraer los trazos de los ejercicios para poder introducirlos en el sistema. En la sección 3.6 se define el procedimiento usado para evaluar los ejercicios y los criterios que se utilizarán para interpretar los resultados obtenidos mediante el análisis de los ejercicios, que se pretende que sean lo más parecido a los criterios que usan los profesores para corregir los ejercicios. A continuación se define en la sección 3.7 el procedimiento para almacenar los ejercicios. Finalmente, en la última sección se explica como se interpretan los resultados obtenidos con los ejercicios y como el sistema apoyará la labor del docente mediante la presentación de resultados y el seguimiento de la evolución de los niños.

### 3.1. Introducción

El objetivo de este proyecto es hacer del ordenador una herramienta más dentro del aula, como el papel y lápiz. Para ello el sistema debe permitir que se puedan realizar ejercicios

de formas diversas, es decir ejercicios sobre papel o bien a través de varios dispositivos electrónicos, como la pantalla táctil, el tablet-pc o la pizarra interactiva.

Cada uno de estos distintos tipos de ejercicios debe ser procesado de distinta manera para poder ser introducido en el sistema. El ejercicio escaneado necesita ser dividido en las distintas partes y recolocado para poder ser corregido. Los ejercicios que se hagan sobre la pantalla táctil u otro dispositivo necesitan de una interfaz especial para poder ser realizados y guardados. Una vez que se realiza esta primera fase de pre-procesado pasan todos el mismo proceso de corrección. Después son guardados en el sistema y se actualiza el modelo del usuario que los realiza. Los resultados de estos ejercicios pueden ser visualizados mediante la interfaz de monitorización.

Para poder corregir de forma automática un ejercicio es necesario haber introducido previamente en el sistema el modelo con el que contrastarlo. Para ello se generan una presentación y un patrón para el ejercicio, que servirán para crear el modelo y para extraer los trazos de la imagen del ejercicio realizado en papel. En la figura 3.1 podemos ver todo este proceso de forma simplificada.

## 3.2. Modelo de datos del sistema

En este proyecto se van a usar metadatos para etiquetar los ejercicios realizados por los alumnos, lo cual permitirá conocer qué realiza bien y qué no, así como cualquier otra información que pudiera ser útil. En sentido genérico se definen los metadatos como “datos de los datos” y surgieron de la necesidad de describir, identificar y caracterizar objetos de información (documentos impresos, sonoros, etc.) a través de campos que identifican cada dato con el propósito de facilitar la gestión, búsqueda, acceso y localización de los documentos. Ejemplo: título, autor, materia, resumen, editor [18].

Las funciones de los metadatos son fundamentalmente [18]:

1. Proporcionar una descripción de un objeto de información junto con otra información necesaria para su manejo y preservación.
2. Proporcionar los puntos de acceso a esa descripción.
3. Codificar esa descripción.

El propósito se puede resumir en tres puntos [18]:

1. Ayudar en la determinación de la relevancia de una información relacionada con una consulta concreta.
2. Analizar las características de un conjunto de información.
3. Enviar información de un sistema a otro diferente.

Si se detalla el esquema de la figura 3.1 en el que se representa todo el proceso que sigue el sistema puede observarse que se tienen fundamentalmente las siguientes entidades:

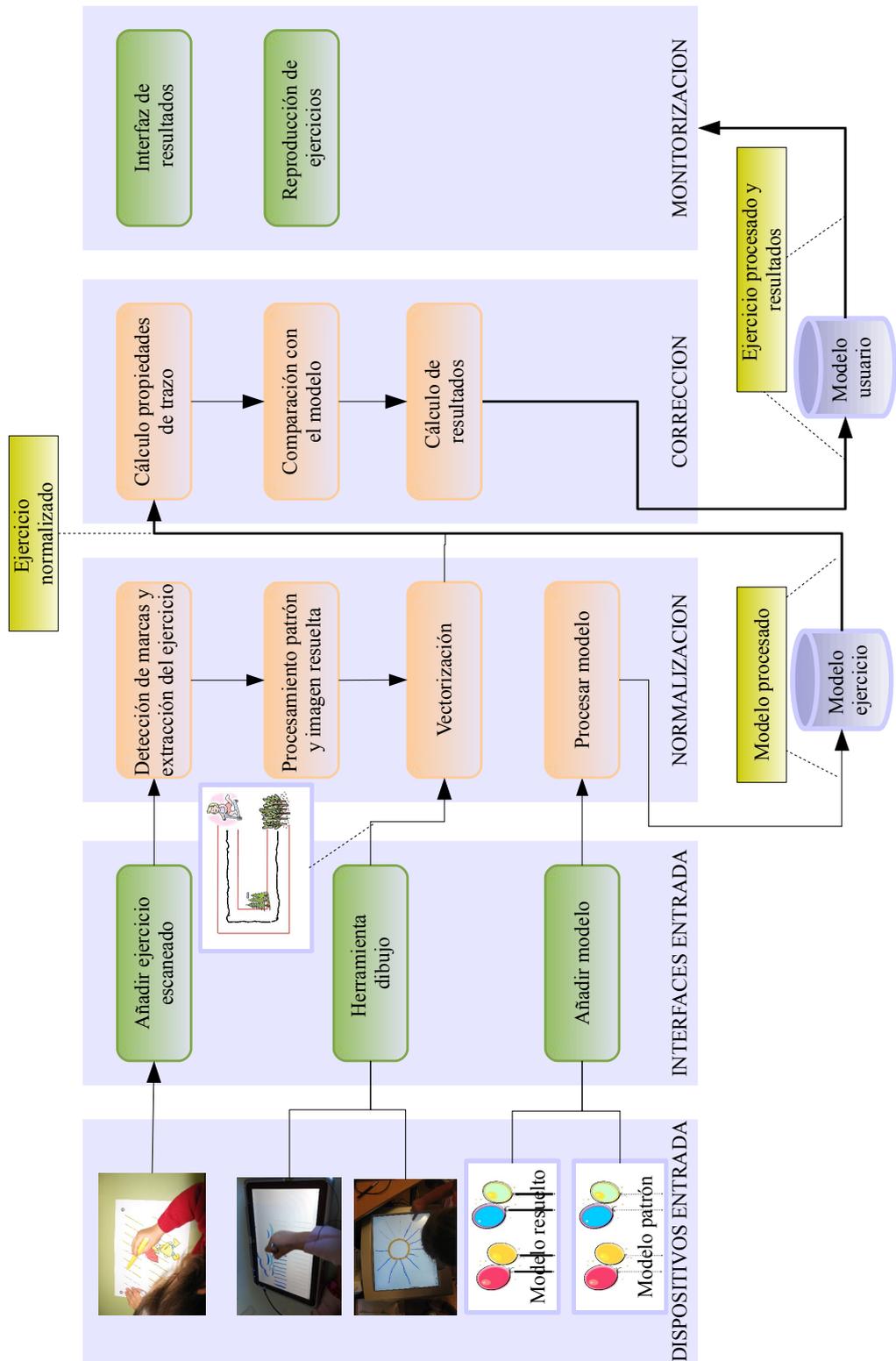


Figura 3.1: Flujo general del sistema

### 3.2.1. Modelo del alumno

El modelo del alumno debe almacenar una representación de su estado de conocimiento así como los datos que definen su perfil completo. Es decir, es necesario guardar todos los datos relativos al alumno así como los resultados que ha ido obteniendo en los distintos tipos de ejercicios para poder hacer un seguimiento de su desarrollo y ver los aspectos en los que podría mejorar. Para ello se almacenan todos los resultados que el alumno va obteniendo en los ejercicios ordenándolos por categorías (dificultad, tipo de trazo, ...) y así poder inferir a partir de estos datos dicha información.

### 3.2.2. Modelo de tareas

Los alumnos realizan un tipo de trazo en función de la edad, a medida que se van haciendo mayores éstos se hacen más complejos. La finalidad de cada ejercicio es que los alumnos practiquen estos distintos tipos de trazos. Por lo tanto, el modelo para cada ejercicio deberá contener que trazo tiene el ejercicio, la dificultad y la edad para la que está pensado.

Además, para realizar la corrección es necesario introducir el modelo correspondiente a la solución del ejercicio. Para ello se debe crear una imagen con el ejercicio punteado y otra con el ejercicio resuelto, que corresponde a una realización perfecta del ejercicio. A partir de estas dos imágenes se crea el modelo con el que se contrastarán los ejercicios realizados por los alumnos. Los modelos se etiquetan con los trazos que contiene.

Para la creación de este modelo se necesitan dos entidades, como se muestra en la figura 3.2:

- Ejercicio vacío

Un ejercicio vacío consta de una imagen y un conjunto de líneas punteadas sobre las que los alumnos tendrán que trazar líneas.

- Patrón ejercicio

A partir del ejercicio vacío se realiza una imagen patrón que se tomará como solución perfecta del ejercicio y se contrastarán con los ejercicios resueltos por los alumnos.

Los ejercicios corregidos se almacenan en el sistema asociados a su autor y constarán de una imagen de con el ejercicio resuelto (escaneado o realizado a través de la pantalla táctil) y también una representación matemática del recorrido de los trazos y de la evaluación obtenida.

A continuación se listan los atributos de las entidades del sistema:

1. Perfil del alumno

- a) Nombre: el nombre del alumno

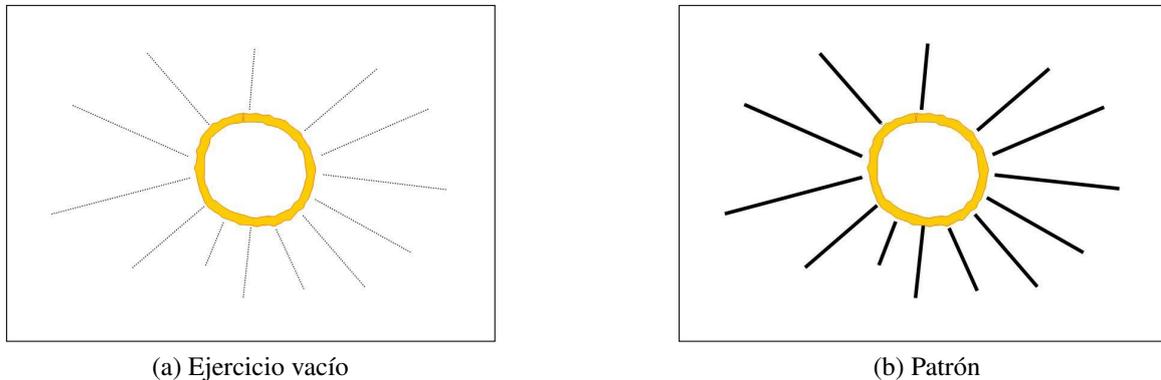


Figura 3.2: Modelo de un ejercicio

- b) Apellidos: los apellidos del alumno
- c) Fecha nacimiento: la fecha de nacimiento del alumno

## 2. Ejercicio

- a) Alumno: el alumno que ha realizado el ejercicio
- b) Modelo: el modelo del ejercicio
- c) N<sup>o</sup> relativo: el numero relativo de ejercicio que ha hecho el alumno
- d) Fecha: la fecha en que se realizó del ejercicio
- e) Escaneado: indica si se trata de un ejercicio escaneado o hecho con la pantalla táctil
- f) Resultados: los resultados obtenidos con el ejercicio
  - 1) Distancia: la calificación obtenida por la distancia de los trazos
  - 2) Diferencia de longitud: la calificación obtenida por la diferencia de longitud de los trazos
  - 3) Completitud: la calificación obtenida por el estado de completitud del ejercicio
  - 4) Nota global: el resultados final del ejercicio

## 3. Modelo de ejercicio

- a) Imagen punteada: la imagen punteada del ejercicio
- b) Patrón: patrón con el que se contrastarán los ejercicios realizados
- c) Edad: la edad para la que está pensada el ejercicio
- d) Dificultad: la dificultad del ejercicio
- e) Trazos: los trazos que contiene el modelo

#### 4. Estado de conocimiento del alumno

- a) Forma: la calificación media obtenida por la realización de los trazos
- b) Distancia: la calificación media obtenida por la distancia de los trazos
- c) Diferencia de longitud: la calificación media obtenida por la diferencia de longitud de los trazos
- d) Nota media: la calificación media global

### 3.3. La entrada de datos

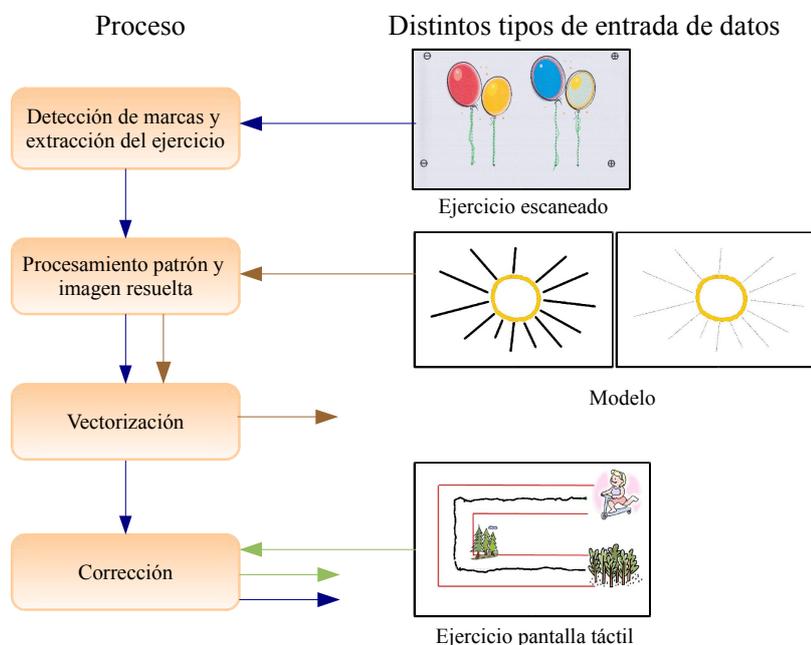


Figura 3.3: Entrada de datos

En el sistema se contemplan dos maneras distintas de introducir ejercicios, desde imágenes escaneadas o bien dibujando directamente sobre la pantalla táctil. Para ello el programa debe ser capaz de tratar las imágenes escaneadas contenidas en ficheros y extraer los trazos realizados por los alumnos así como implementar un sistema para que se puedan realizar ejercicios sobre una pantalla táctil. Éstos serán contrastados con su correspondiente modelo que debe haber sido introducido previamente en el sistema.

Por lo tanto tenemos tres tipos distintos de entrada de datos, los ejercicios escaneados, los modelos y los ejercicios hechos con la pantalla táctil. Cada uno tiene que ser preprocesado de

distinta manera, sin embargo todos comparten parte del proceso. Por ejemplo, para incorporar nuevos modelos se sigue esencialmente el mismo proceso que con los ejercicios escaneados, sin embargo los trazos extraídos se guardan en el sistema y no se corrigen. En el caso de los ejercicios realizados con la pantalla táctil no es necesario extraer los trazos de la imagen porque se almacenan automáticamente, sin embargo el proceso de corrección es igual al de los escaneados. En la figura 3.3 podemos ver qué parte del proceso comparte cada uno.

### 3.3.1. Impresión de ejercicios

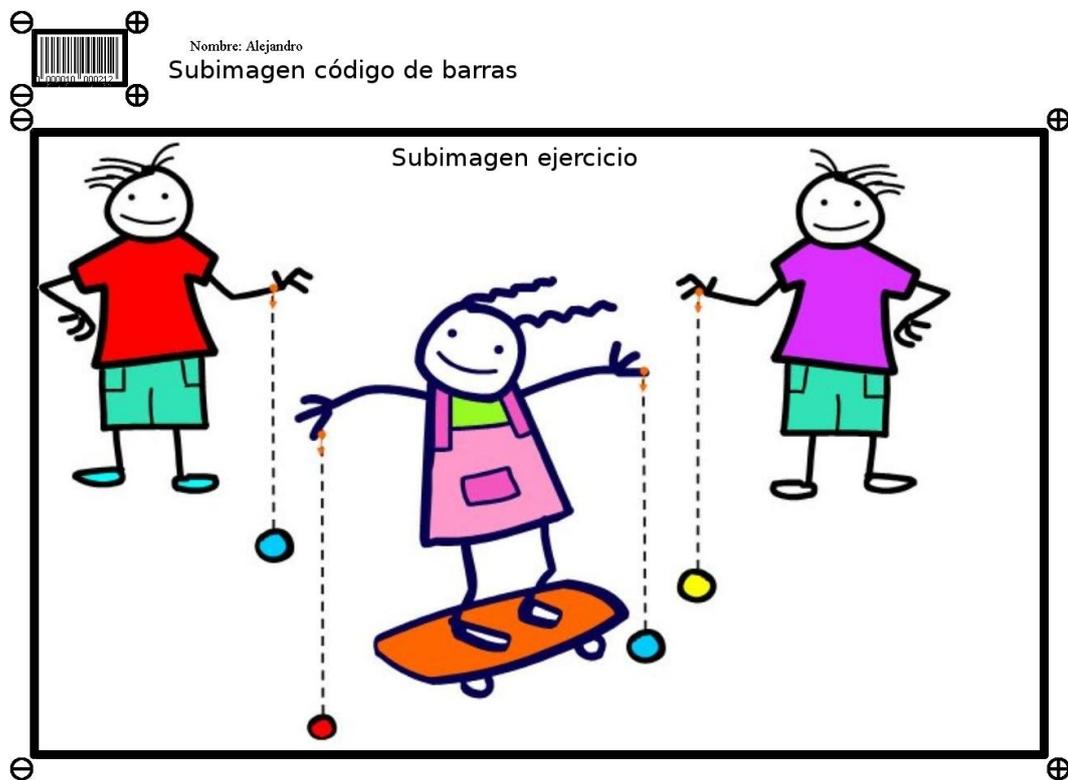


Figura 3.4: Ejercicio etiquetado con código de barras

Cuando se imprimen los ejercicios para los alumnos es necesario añadir un código de barras para que el sistema detecte posteriormente el alumno al que se le ha asignado y el modelo del que se trata. Además se añaden cuatro marcas alrededor del código y la imagen principal para poder segmentar el ejercicio posteriormente. En la figura 3.4 vemos un ejemplo de un ejercicio con código de barras y marcas delimitando el código y la imagen del ejercicio.

### 3.3.2. Ejercicios realizados sobre la pantalla

Los ejercicios que se realicen mediante la pantalla táctil serán guardados automáticamente y no será necesario ningún preprocesamiento para detectar marcas y segmentar el ejercicio.

Estos ejercicios se realizan gracias al módulo de dibujo que además de la imagen generará un fichero de log de modo que se podrán recrear sin necesidad de la imagen guardada. En él se guardan todas las acciones del usuario de forma que para corregirlo posteriormente sólo será necesario el log y no se tendrá que procesar la imagen.

### 3.3.3. Crear un modelo

Para crear un modelo el proceso seguido es similar al de los ejercicios escaneados, hay que extraer los trazos y vectorizarlos para posteriormente almacenarlos en el sistema. En la figura 3.5 se puede ver un ejemplo del resultado de crear un modelo.

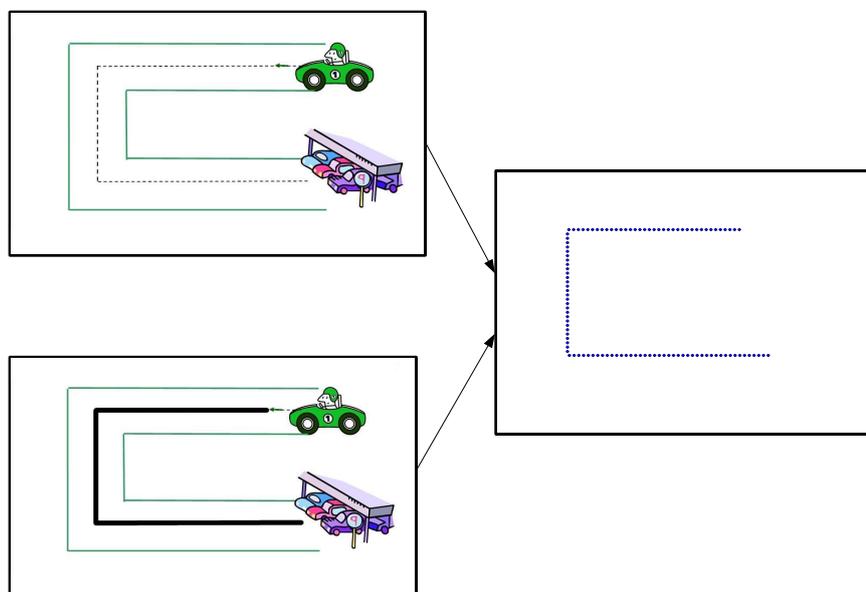


Figura 3.5: Creación de un modelo

## 3.4. Procesado de ejercicios escaneados

Una vez que los alumnos han hecho el ejercicio y se ha escaneado, el primer paso consistirá en detectar la posición de las marcas de los ejercicios para recolocarlos correctamente en el caso de que al escanearlos hayan sido girados. El programa colocará la imagen en la posición correcta y separará las partes que contienen el ejercicio y el código de barras para poder procesarlos. Para ello se realizarán una serie de acciones que se muestran en la figura 3.6 y cuyo funcionamiento se explica en las subsecciones que se incluyen a continuación:

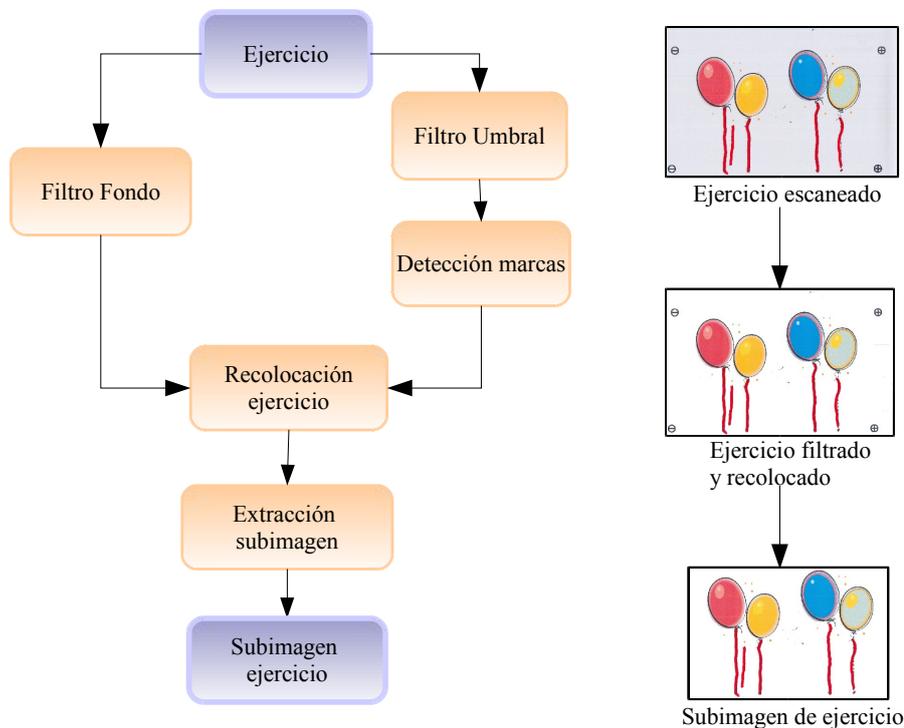


Figura 3.6: Diagrama de detección de marcas

### 3.4.1. Filtrado de la imagen

A partir de la imagen de entrada que contiene el ejercicio se generan dos imágenes distintas aplicando dos filtros diferentes a dicha imagen. Una de las imágenes resultantes se usará para detectar las marcas y la otra para extraer los trazos.

Para generar la imagen que detecta las marcas se aplica un filtro de umbral a la imagen original para pasarla a blanco y negro y después otro filtro para reducir el ruido que pueda contener y evitar problemas en la detección de las marcas. De esta forma evitamos por ejemplo que aparezcan píxeles blancos dentro de las marcas. Los filtros usados se explican en la sección 5.4.1.

A la imagen de la que se extraerán los trazos se le aplica un filtro de fondo. En la figura 3.7 podemos ver la aplicación de este filtro a un ejercicio escaneado. De este modo se pretende conservar todos los trazos que haya en la imagen y se evita que trazos con un color muy claro se pierdan. Al escanear la imagen se puede generar ruido y el color blanco del papel puede aparecer como gris. Este proceso de filtrado se explica en la sección 5.4.2.

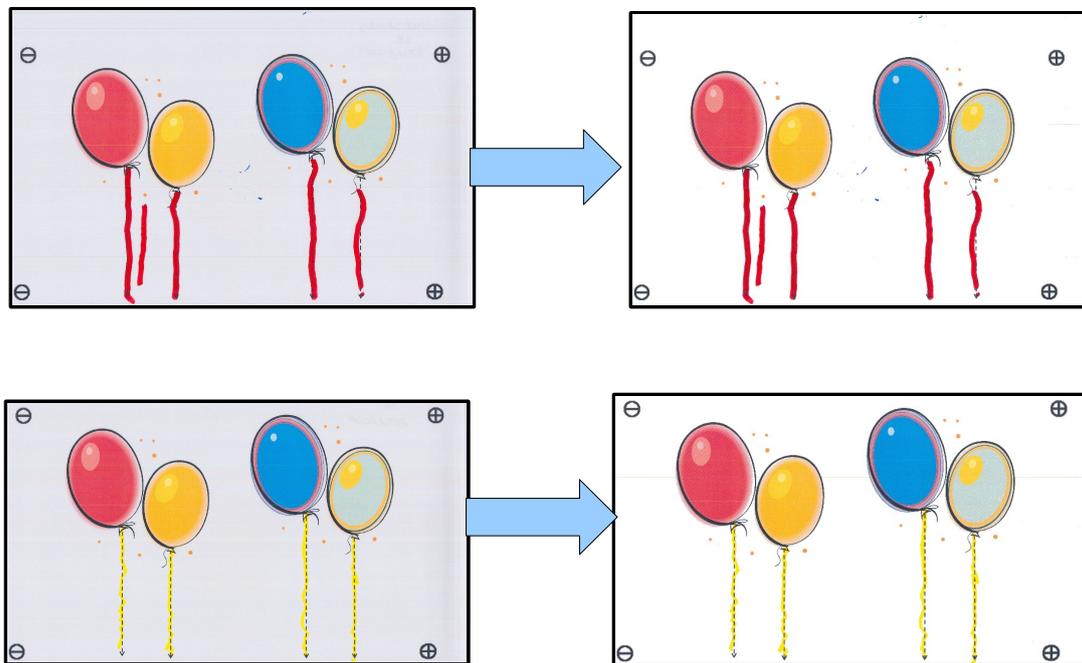


Figura 3.7: Imagen con el fondo filtrado

### 3.4.2. Detección de las marcas

Las marcas que cada ejercicio lleva en las esquinas sirven para recolocar la imagen en el caso de que esté girada y para delimitar la zona central para poder contrastarla con el modelo. Para detectar las marcas y diferenciar las de la izquierda de las de la derecha se extraen una serie de características de todos los elementos que componen la imagen para ver si encajan con las de las marcas. Se comprueba el tamaño, la relación entre altura y anchura, el porcentaje de área en negro y el número de agujeros del símbolo. Para identificarlas correctamente estas características deben ser independientes de la rotación de las marcas. En la figura 3.8 se ve como el sistema ha detectado las dos marcas izquierdas y derechas.

### 3.4.3. Recolocación de la imagen

A partir de la posición de las marcas izquierdas y derechas se determina en que sentido está girada la imagen y el número de grados que es necesario girarla para colocarla correctamente. Una vez que está bien colocada se recorta el área comprendida entre las cuatro marcas y así se obtiene el dibujo que contiene el ejercicio realizado.



Figura 3.8: Detección de marcas en una imagen

#### 3.4.4. Extracción de subimágenes

Con la ayuda de las posiciones de las marcas detectadas se extraen las dos subimágenes del ejercicio, una contiene el código de barras y otra la imagen con los trazos. La subimagen que contiene el código de barras servirá para que después de ser decodificada indique el modelo y el alumno que realizó el ejercicio y no tener que introducir esta información manualmente. La segunda subimagen que contiene los trazos pasará al siguiente paso del proceso.

#### 3.4.5. Decodificación del código de barras

El código de barras contiene el modelo del ejercicio y el alumno al que se ha asignado. Esta información es necesaria para continuar con la corrección y en el caso de que no se pudiera decodificar por algún motivo habría que agregarla manualmente.

### 3.5. Extracción y vectorización de trazos

Este paso se aplica tanto a los ejercicios como a los modelos. El objetivo es extraer los trazos de los ejercicios o del modelo y convertirlos a una representación geométrica. En el caso de los modelos la imagen patrón hará de “ejercicio resuelto”. El ejercicio escaneado llega a este paso después de habersele retirado las marcas y recolocado en la posición correcta. Los pasos del procedimiento son extraer los trazos, reconstruirlos y vectorizarlos. En la figura 3.9 se muestra el proceso seguido.

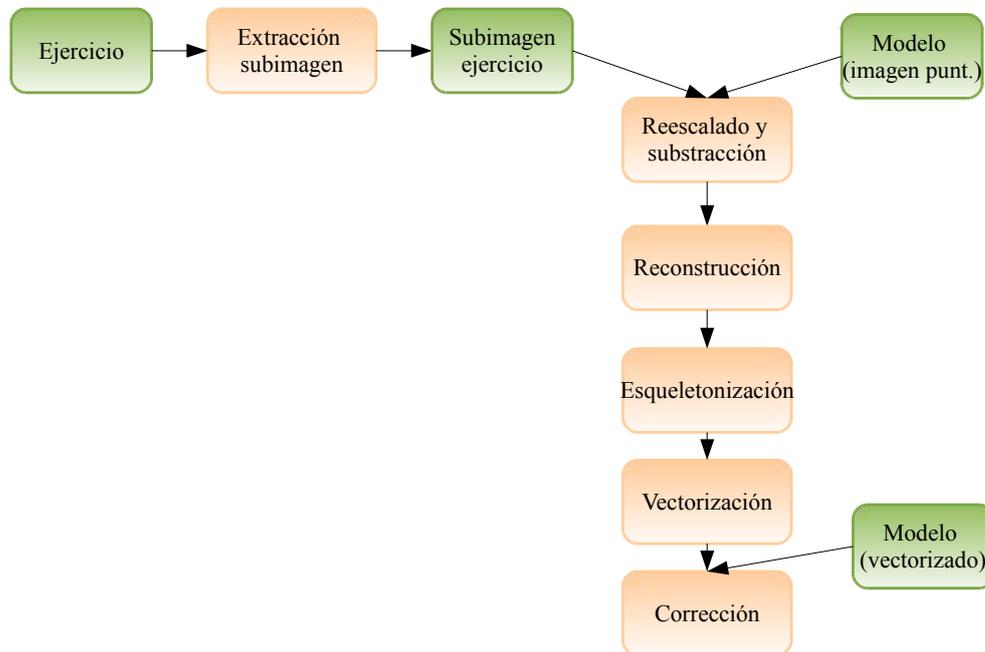


Figura 3.9: Diagrama de filtrado

### 3.5.1. Extracción de los trazos

Para poder corregir los ejercicios es necesario separar los trazos del alumno del resto de la imagen escaneada. Para ello se usa la imagen punteada del modelo y se contrasta con la imagen del ejercicio. Cualquier elemento del ejercicio que no esté en el modelo se considerará como un trazo del alumno.

### 3.5.2. Reconstrucción de los trazos

La imagen que se crea durante la extracción de los trazos no está completa. En el proceso se han eliminado las partes de los trazos que corresponden a las guías del ejercicio vacío, por lo que se debe realizar un proceso de reconstrucción para rellenar las partes de los trazos que se hayan escrito sobre las guías del ejercicio. Para ello se obtienen las posiciones de las guías a partir del modelo punteado y se comprueba en el ejercicio resuelto si se ha dibujado alrededor de ellas.

### 3.5.3. Esqueletonización de los trazos

Para poder convertir los trazos a una representación de polilínea y poder compararla con el modelo se debe reducir el ancho de los trazos hasta una anchura de un píxel. Idealmente dicha línea recorrerá el centro del trazo que realizó el alumno. Como consecuencia de este proceso es posible que se generen ramas pequeñas en los lugares donde el trazo presente esquinas y deberán ser eliminadas puesto que no representan el trazo original. Para ello se recorre toda la imagen y se eliminan las ramas cuya longitud no supera un cierto umbral.

### 3.5.4. Separación y vectorización de los trazos

A partir de la imagen que contiene el esqueleto de los trazos hay que transformar las líneas que contiene a una representación geométrica para poder tratarlas computacionalmente. Lo primero es separar los trazos, ya que puede haber varios cruzados. El proceso de separación puede convertirse en una tarea imposible de resolver si dichos trazos están muy mezclados o han seguido trayectorias extrañas. Por lo tanto para la separación se intentará tomar los trazos que presenten una continuidad más lógica. Para ello en los puntos en los que se crucen se considerarán las ramas que sean más continuas como un único trazo.

En la figura 3.10 se puede ver un ejemplo de la aplicación del proceso de extracción de trazos a una imagen concreta y para cada paso intermedio se muestra el resultado.

## 3.6. Corrección de los ejercicios

A partir del conjunto de trazos obtenidos de los ejercicios escaneados o bien de la interfaz de dibujo y de los modelos, se realiza la corrección. El proceso de corrección tiene una serie de fases que se detallan en las siguientes subsecciones.

### 3.6.1. Asignación de los trazos del ejercicio a los del modelo

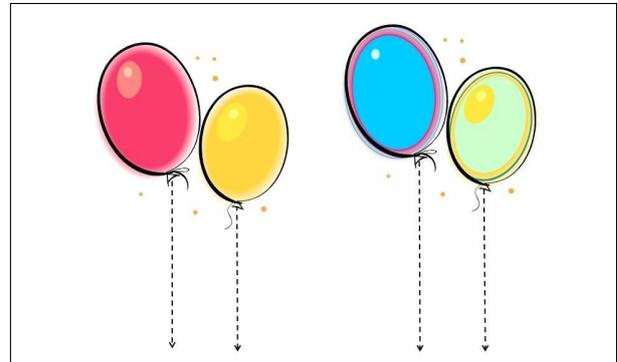
El primer paso será el realizar una correspondencia entre los trazos del ejercicio y los del modelo. Esta asignación se hace en base a la distancia, por lo tanto hay que calcular que trazo del modelo está más próximo al del ejercicio. Si no hubiera ninguno cerca se consideraría que es un trazo no válido y se descartará a la hora de hacer la corrección.

### 3.6.2. Comparación de los trazos con los del modelo

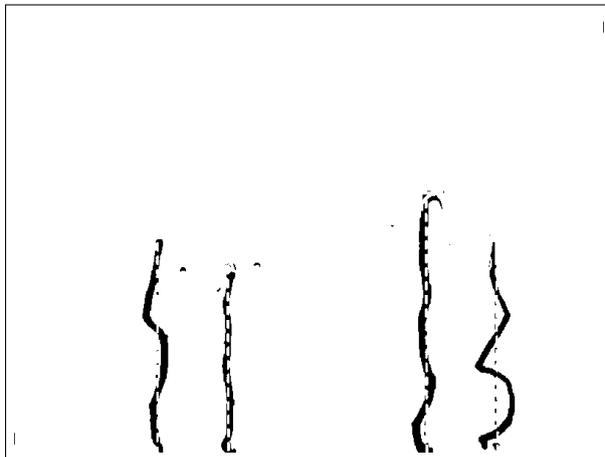
A continuación se compara el trazo escrito con el del modelo y en función de una serie de criterios se evaluará la corrección. Los características que se examinan son los siguientes:



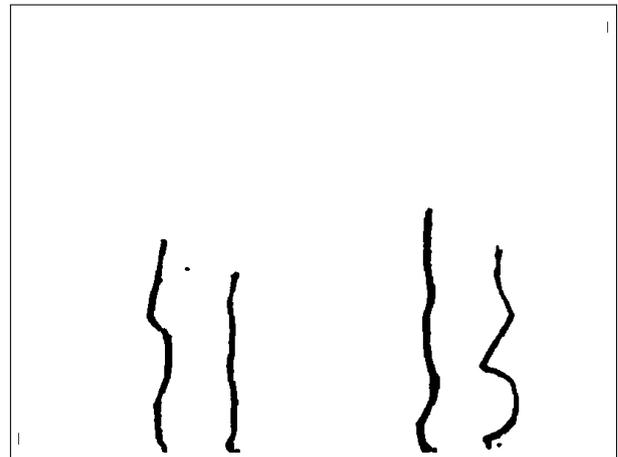
(a) Ejercicio



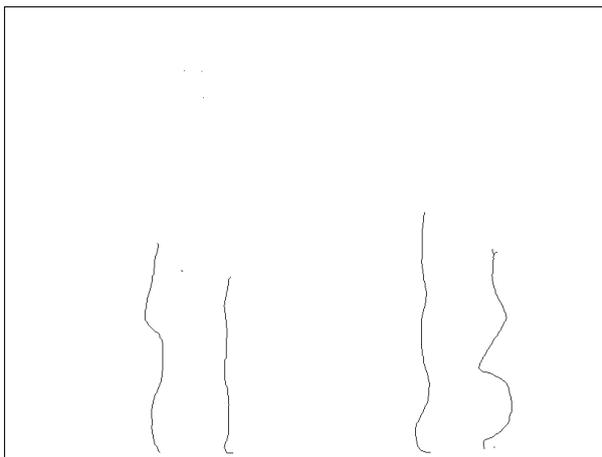
(b) Modelo



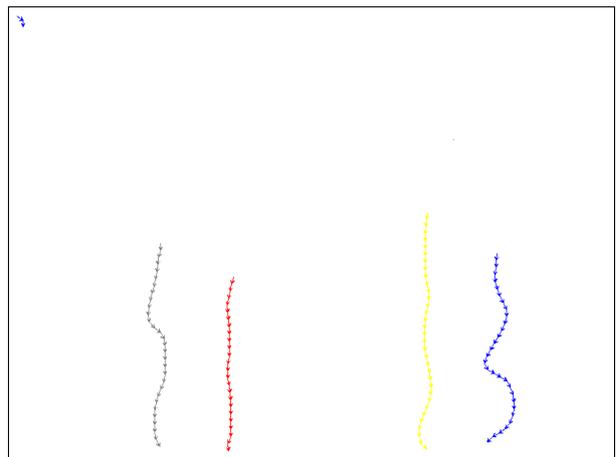
(c) Trazos extraídos



(d) Reconstrucción



(e) Esqueleto



(f) Trazos vectorizados

Figura 3.10: Ejemplo de extracción y vectorización

### 1. Completitud

Mientras más partes del trazo modelo cubra el del ejercicio más completo estará éste. Si el alumno realiza trazos demasiado cortos o discontinuos menor será el valor de completitud.

### 2. Completitud total del ejercicio

Este cálculo se realiza comprobando si hay alguna parte de los trazos realizados por los alumnos dentro de una distancia máxima al modelo, si no la hay esa parte se considerará no realizada. Por lo tanto, si el alumno no ha completado alguna parte del ejercicio este valor será menor.

### 3. Longitud del trazo

Si el alumno ha realizado un trazo demasiado corto o largo en comparación con el modelo, esto se reflejará en que la diferencia de longitudes entre los dos será mayor. Sin embargo el hecho de que las longitudes sean similares no quiere decir que el trazo sea correcto. Para hacer este cálculo se suman las longitudes de los segmentos que forman un trazo.

### 4. Proximidad del trazo

La proximidad es otro aspecto que se examina. Cuanto menor sea la distancia entre los trazos del modelo y los del ejercicio más puntuación obtendrá. Para ello se calcula para cada punto del trazo del ejercicio la distancia al más cercano del modelo. La media de estos valores nos dará el valor de proximidad.

### 5. Forma

Esta característica mide si el trazo se asemeja en cuanto a su curvatura al del modelo. Si está desplazado pero conserva la forma del modelo obtendrá toda la puntuación en esta característica. Para hacer esta medición se crean dos funciones que representen la forma del trazo del y del ejercicio y se examinan cuantitativamente las diferencias entre ellas.

Después de realizar este análisis para cada trazo se guardarán los resultados y se actualizarán en la base de datos.

## 3.6.3. Cálculo de la nota de trazo

Cada trazo realizado por el alumno se evaluará en función de las propiedades extraídas y de sus respectivas puntuaciones. Cada característica tiene un peso asociado en función de su importancia. La más importante es la forma y por tanto es la que más se tiene en cuenta a la hora de puntuar el trazo.

A cada característica extraída del trazo se le asignará una nota que se calculará de la siguiente manera: cada propiedad tiene un valor mínimo y un máximo, si el valor de la

propiedad no llega al mínimo recibirá una puntuación de 0 y si sobrepasa el máximo la puntuación será de 10. El rango que hay entre el valor mínimo y el máximo se divide en 10 niveles y la puntuación se calcula en función de en que nivel esté el valor de la propiedad.

### Cálculo de la nota global

El ejercicio se evalúa en función de las puntuaciones obtenidas en los trazos y las características globales del ejercicio, que son la completitud global del ejercicio y el número de trazos que ha necesitado el alumno para realizar el ejercicio. A cada trazo se le asigna un peso en función de su longitud y así se calculan las notas medias de cada propiedad de los trazos.

#### 3.6.4. Ejemplo de corrección de un ejercicio

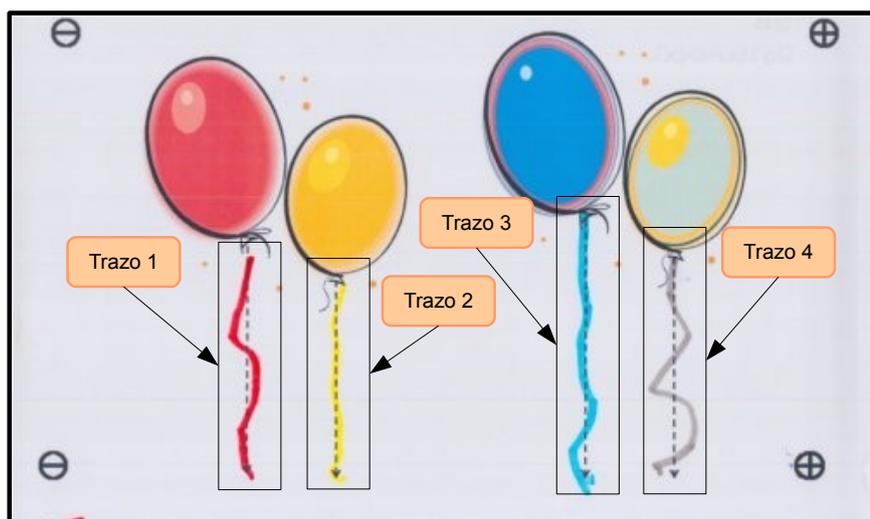


Figura 3.11: Corrección de un ejercicio

En la figura la figura 3.11 se muestra el ejercicio de ejemplo realizado por un alumno y en la tabla 3.1 vemos los resultados individuales que han obtenido cada trazo así como la nota global del ejercicio. En este ejercicio se pueden ver claras diferencias entre los trazos y sirve para demostrar el método de evaluación. Los resultados mostrados son la diferencia de longitud, la distancia, la diferencia de forma y la nota del trazo. La diferencia de forma refleja la desviación media de cada trazo con respecto al modelo. Por lo tanto los trazos con forma más parecida al modelo serían de menor a mayor el 2, 3, 1 y 4. Ésta es la característica más importante y la que más peso tiene en la evaluación, por lo que el orden de las notas de trazo no cambia con respecto al de la forma y el que mejor nota ha obtenido ha sido el segundo

Diferencia de longitud	7.41	10	Diferencia de longitud	14.29	9
Diferencia de forma	13.06	2	Diferencia de forma	6.66	7
Distancia	3.98	7	Distancia	3.59	8
Nota global		5	Nota global		7
(a) Trazo 1			(b) Trazo 2		
Diferencia de longitud	10.34	10	Diferencia de longitud	20.83	8
Diferencia de forma	8.43	6	Diferencia de forma	24.41	0
Distancia	3.51	8	Distancia	5.97	5
Nota global		7	Nota global		3
(c) Trazo 3			(d) Trazo 4		
Diferencia de longitud	14.49	9			
Diferencia de forma	14.67	1			
Distancia	4.41	7			
Nota global		5			
(e) Evaluación global					

Tabla 3.1: Tabla de resultados

trazo, que es el que mejor ha realizado el alumno y más se parece al modelo. Obviamente el trazo peor conseguido es el cuarto.

Las distintas notas globales del ejercicio se calculan multiplicando las notas de cada trazo por su longitud y dividiéndola entre la suma de todas las longitudes. Por lo tanto trazos largos tendrán más peso que los trazos cortos a la hora de hacer el cómputo. Además entra en juego el porcentaje que se ha completado del ejercicio a la hora de hacer la evaluación global.

### 3.7. Almacenamiento de ejercicios

Un ejercicio se guarda como una imagen más un fichero con el recorrido de los trazos. En el caso de los ejercicios realizados en la pantalla táctil este trazado es perfectamente conocido y además será posible reproducirlos más tarde. Para ello será necesario almacenar los puntos en que comienzan y terminan cada trazo así como una marca de tiempo para los ejercicios realizados con la interfaz de dibujo. En el caso de los escaneados estas marcas de tiempo no existen y simplemente se almacenan los puntos que describen la forma de los trazos.

### 3.8. La salida e interpretación de resultados

Todos los datos obtenidos en las evaluaciones se almacenan en el modelo del alumno. A través del análisis de los ejercicios se pretende obtener datos sobre su realización e identificar

los aspectos en los que falla el alumno para poder corregirlos. Así se tendrá información precisa sobre su estado de conocimiento, se podrá ver que alumnos hacen peor los ejercicios o que tipo de trazo es el que más le cuesta a un alumno determinado. A partir de esta información el profesor podrá intervenir y ver por qué motivo realiza los ejercicios incorrectamente o podrá hacer que el alumno repita el ejercicio en el que ha obtenido mala puntuación para que mejore sus capacidades grafomotrices.

# Capítulo 4

## Funcionamiento del sistema

En este capítulo se describen los pasos a seguir para preparar tareas para los alumnos con ayuda del sistema implementado. Además se explicarán con detalle todas las funciones de la interfaz.

En primer lugar se deberá añadir el modelo de los ejercicios que se quieran preparar e introducir los datos de los alumnos para poder asociar los ejercicios. A partir de este punto existen dos opciones, que los alumnos hagan los ejercicios en papel o a través de un dispositivo táctil. En cualquier caso hay que acceder a los modelos almacenados en el sistema a través de la búsqueda de modelos.

Si queremos que los ejercicios se realicen sobre papel habrá que imprimirlos a través del diálogo de impresión. Una vez realizados los ejercicios se escanean y a través de la ventana de entrada de ejercicios se pueden corregir y almacenar en el sistema. También se pueden realizar a través de la pantalla táctil mediante la interfaz de dibujo. Este tipo de ejercicios se corrigen automáticamente al finalizar y se pueden guardar en el sistema.

Una vez guardados los ejercicios se pueden ver más tarde a través de la interfaz de búsqueda. De esta forma se tiene una relación de los ejercicios que ha hecho cada alumno y los resultados que ha obtenido. Se podrá ver el estado del conocimiento de los alumnos para cada tipo de ejercicio así como la media global. Así se podrán preparar ejercicios para reforzar los tipos de trazo que el alumno realiza peor.

### 4.1. Barra de herramientas

A través de la barra de herramientas se puede acceder a las principales pantallas del programa. Éstas son la pantalla de búsqueda, la pantalla de dibujo y las pantallas de entrada de modelo y de ejercicios. Esta barra se muestra en la figura 4.1.

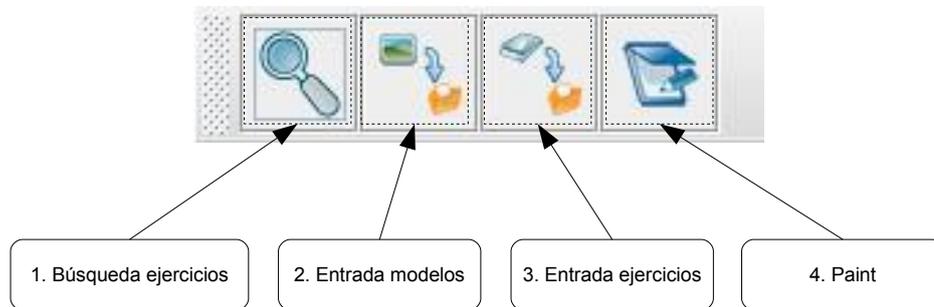


Figura 4.1: Barra de herramientas

## 4.2. Añadir un nuevo modelo

Para activar la ventana de entrada de modelos (Figura 4.2) se pulsa el botón de entrada de modelos en la barra de herramientas. En la pantalla para añadir modelos se selecciona la carpeta donde se encuentran las imágenes de los modelos que se quieren añadir. A continuación se deben seleccionar la imagen que servirá de plantilla y la imagen que contiene la solución del ejercicio. Pulsando el botón para generar el modelo el sistema lo creará. Después se marcan las propiedades del modelo y se pulsa el botón de añadir modelo para guardarlo.

### 1. Elegir imagen punteada

Pulsando este botón se selecciona la imagen que corresponde al modelo sin rellenar.

### 2. Elegir imagen resuelta

Selecciona la imagen que corresponde al modelo resuelto.

### 3. Propiedades de modelo

Aquí se deben introducir las propiedades que tenga el modelo y que servirán para etiquetarlo, la dificultad, la edad y los trazos que contiene el modelo.

### 4. Generar modelo

El sistema generará el fichero que contendrá los trazos vectorizados del modelo para compararlos después con los ejercicios.

### 5. Ver modelo

Muestra el modelo generado por el sistema para comprobar si lo ha hecho correctamente.

### 6. Guardar modelo

El modelo creado se guardará en la base de datos.

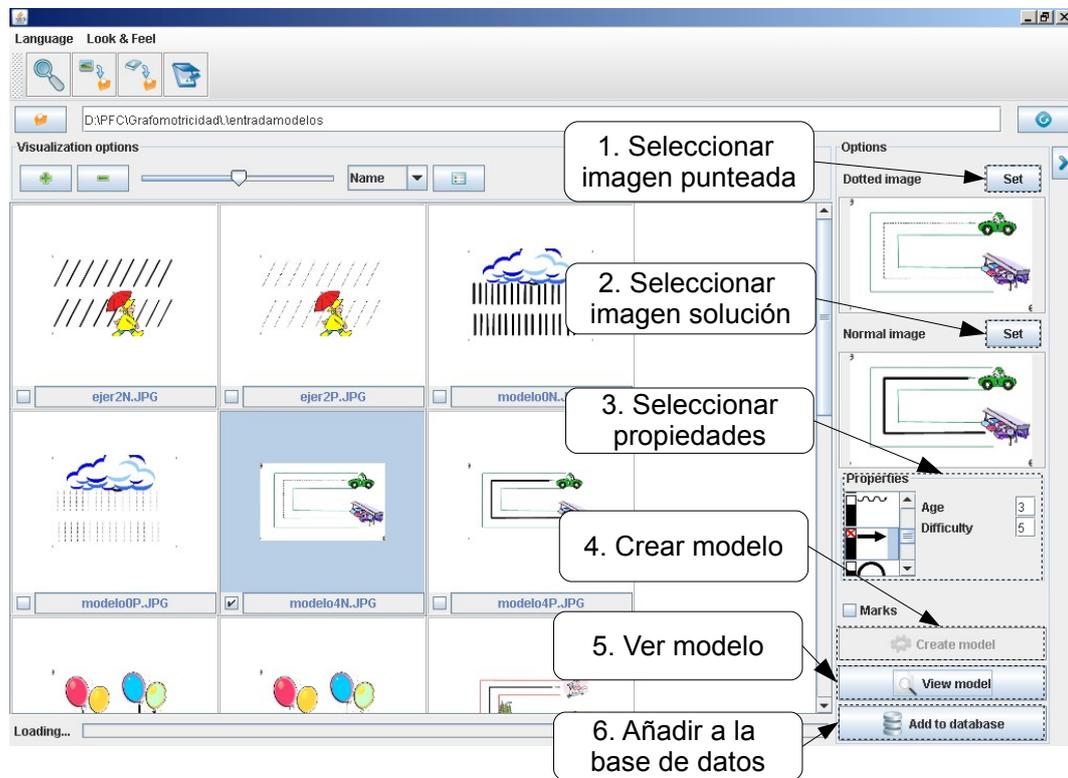


Figura 4.2: Pantalla de modelos

### 4.2.1. Propiedades que deben tener los modelos

Para que los modelos puedan ser procesados e introducidos en el sistema deben cumplir una serie de características. Las guías deben estar separadas de los dibujos. Tampoco se permite que los trazos se crucen o cubran parte de los dibujos del ejercicio. La versión resuelta del modelo debe ser exactamente igual que la versión sin completar y además, los trazos de la versión resuelta deben tener los finales redondeados.

## 4.3. Gestión de alumnos

Para añadir nuevos alumnos hay que hacerlo usando los botones que hay en la lista de alumnos (Figura 4.3) dentro de la interfaz de búsqueda, a la que se accede través de la barra de herramientas. Estos controles permiten añadir un alumno a través de un diálogo. También se podrán borrar alumnos o modificar sus datos. Las acciones que se pueden realizar son las siguientes:

1. Añadir un alumno nuevo

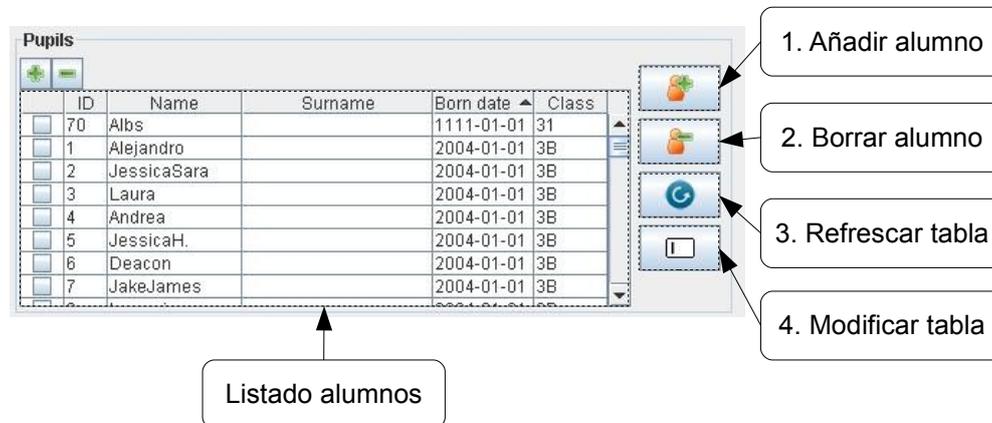


Figura 4.3: Listado alumno

Pulsando este botón se abrirá un diálogo en el que habrá que introducir los datos del alumno en los campos de texto. Una vez completados los datos se de pulsar el botón de añadir. Si la fecha de nacimiento no es una fecha válida el sistema mostrará un mensaje de error.

#### 2. Borrar un alumno

Este botón borra los alumnos que estén seleccionados en la tabla, así como los ejercicios que hubieran realizado.

#### 3. Refrescar tabla

Refresca los datos de los alumnos en la tabla.

#### 4. Modificar datos de un alumno

Para modificar un dato de la tabla se activa este botón y se hace doble-click sobre el campo que se desea modificar. A continuación se introduce el nuevo valor en la tabla.

## 4.4. Búsqueda

Una vez se ha añadido el modelo al sistema debemos acceder al menú de búsqueda. La ventana de búsqueda se activa pulsando sobre el botón de búsqueda de la barra de herramientas. Aquí se pueden ver los ejercicios que han realizado los alumnos y los modelos disponibles. También se pueden generar gráficas con los resultados obtenidos por los alumnos. Con las opciones de filtrado se pueden reducir los resultados de las búsquedas.

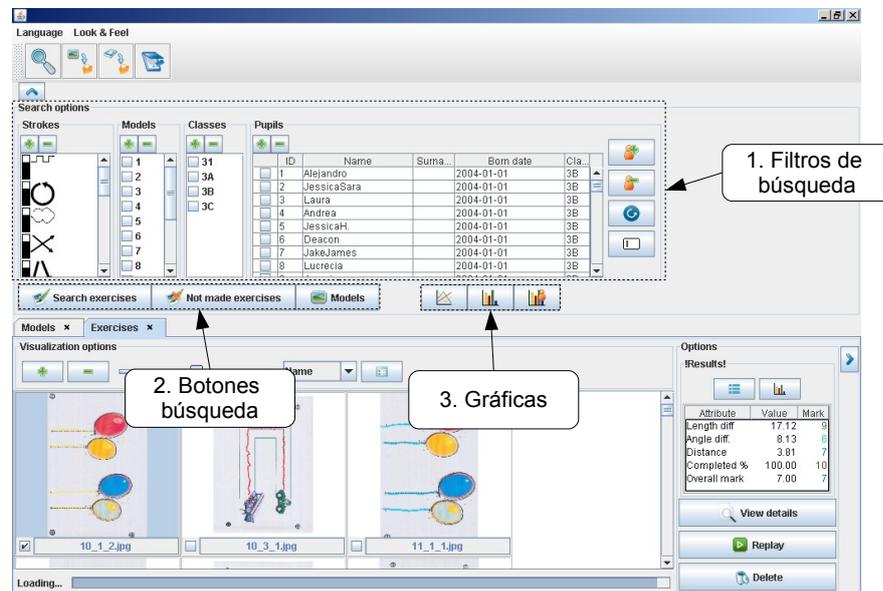


Figura 4.4: Pantalla de búsqueda

## 1. Filtros

Hay varias opciones para filtrar los ejercicios:

### a) Modelo

Sólo muestra los ejercicios o modelos marcados en esta lista.

### b) Trazos

Con esta opción de filtrado sólo se mostrarán los ejercicios que contengan algún tipo de trazo especificado.

### c) Clase

Sólo muestra los resultados de los alumnos que pertenezcan a las clases marcadas.

### d) Alumnos

Para mostrar los ejercicios que han realizado un conjunto de alumnos concretos se marcan en esta lista.

## 2. Botones búsqueda

### a) Ver ejercicios

Para que se muestren en la lista los ejercicios realizados hay que pulsar sobre el botón “Mostrar ejercicios” y se abrirá un panel con los resultados de la búsqueda realizada.

b) Ver modelos

Para buscar modelos se debe presionar el botón “Ver modelos”. Dependiendo de si se ha realizado una búsqueda de modelos o ejercicios se presentarán las diferentes acciones que se pueden realizar.

3. Botones gráficas

Estos botones sirven para generar gráficas con datos relativos a los alumnos y a los ejercicios que han realizado. Estas gráficas se explican en la sección 4.8.

a) Media

Muestra la media de los resultados agrupados por alumno. Para ello se deben seleccionar de la lista de alumnos y marcar los filtros que queramos para que los resultados sólo representen los ejercicios que cumplan las condiciones de filtrado.

b) Progresión

Muestra la progresión para un solo alumno usando los filtros para reducir los datos de los ejercicios que se muestren en la gráfica. Estos datos se muestran en orden cronológico.

c) Perfil

Muestra el perfil para un solo alumno, es decir la media para cada modelo que haya hecho. También se pueden filtrar estos resultados.

#### 4.4.1. Lista de imágenes

Cada vez que se realice una búsqueda se abrirá una nueva lista de imágenes. Todas estas listas tienen una serie de características comunes y un conjunto de acciones que se puede realizar sobre cada una de ellas. En la figura 4.5 vemos una lista de imágenes.

1. Seleccionar y deseleccionar todos los elementos

A través de estos botones se pueden seleccionar y deseleccionar todos los elementos de la lista.

2. Barra de escalado

Permite escalar las imágenes, hacerlas más pequeñas para que quepan más en la misma pantalla o bien agrandarlas para poder ver los detalles.

3. Opción de ordenar

Dependiendo de los elementos que contenga la lista se podrán ordenar por cualquiera de sus atributos.

4. Ver detalles

Pulsando este botón se mostrarán los atributos de cada elemento de la lista.

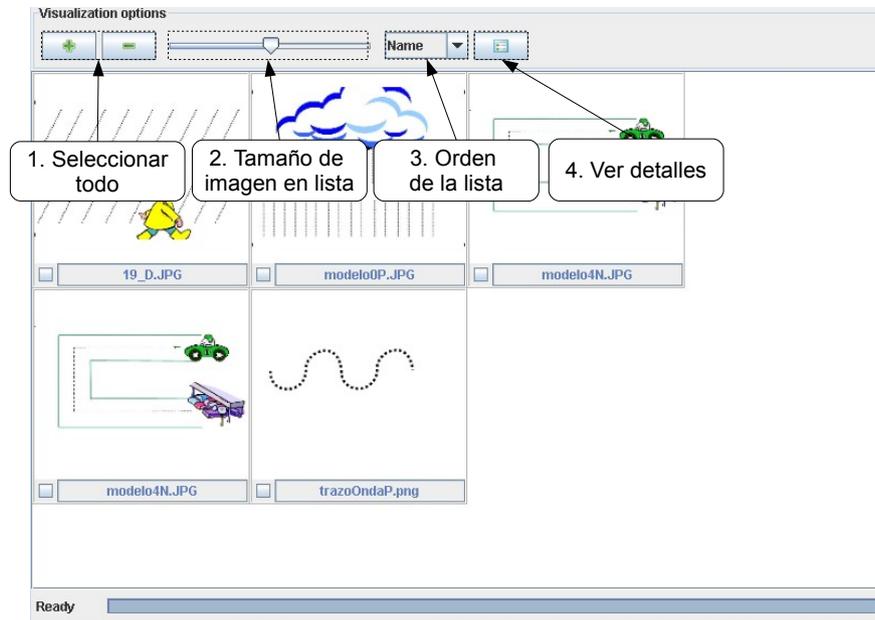


Figura 4.5: Lista de imágenes

#### 5. Abrir una imagen

Si se hace doble-click sobre una imagen se abrirá una ventana para ver la imagen al detalle. La figura 4.5 muestra un ejemplo de una imagen mostrada con el visor. Se pueden realizar el siguiente conjunto de acciones:

- a) Zoom: Con la rueda del ratón se puede incrementar o decrementar el zoom sobre la imagen y se hará sobre el punto en el que esté situado el cursor del ratón.
- b) Desplazamiento: Para desplazar la imagen se debe pinchar sobre ella y arrastrarla.

### 4.4.2. Lista de modelos

Para que los alumnos realicen un ejercicio hay que realizar una búsqueda y seleccionar los modelos que se quiera de la lista. Si queremos imprimir los ejercicios hay que pulsar el botón de impresión y para que lo realicen sobre la pantalla táctil se pulsa el botón de dibujo. También se pueden borrar los modelos a través de esta interfaz, lo que provocará que se borren todos los ejercicios asociados a ellos.

#### 1. Imprimir modelos

Si pulsamos sobre este botón se abrirá un diálogo que permitirá imprimir los modelos que se hayan seleccionado en la lista.

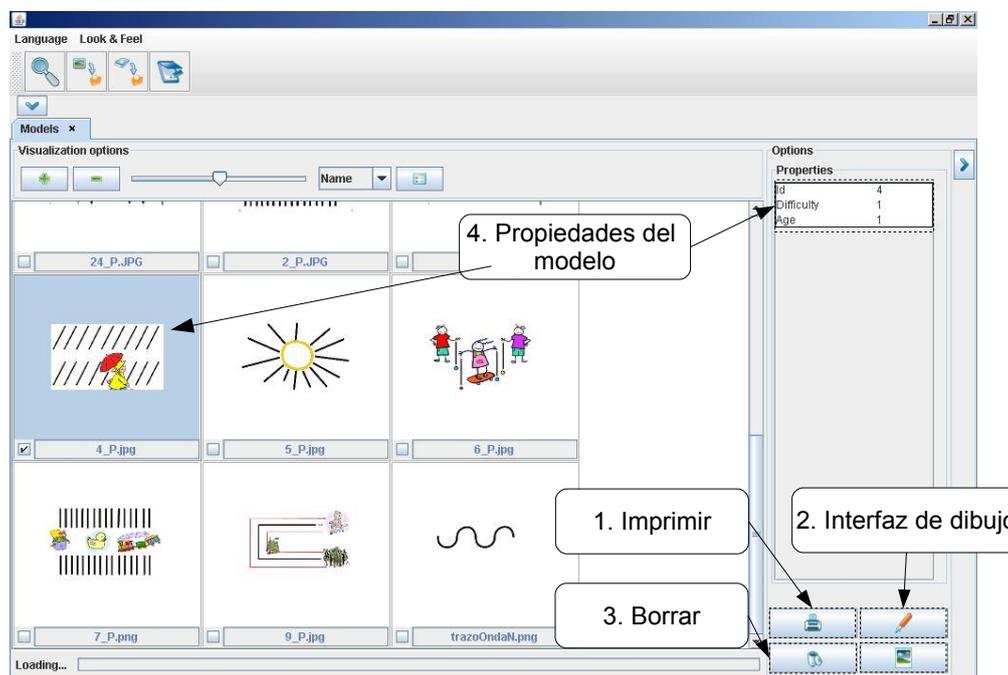


Figura 4.6: Pantalla de búsqueda de modelos

## 2. Interfaz de dibujo

Si se pulsa este botón se cargará el modelo en la ventana de dibujo explicada en la sección 4.5 y se activará dicha ventana para realizar los modelos que se hayan marcado a través de la pantalla táctil.

## 3. Borrar un modelo

Se borrarán los modelos seleccionados en la lista y todos los ejercicios asociados a ese modelo.

### 4.4.3. Imprimir modelos

A través de este diálogo se pueden imprimir los ejercicios a los que se le añadirán un código de barras que identifique el modelo y el alumno al que se le ha asignado, para que posteriormente se reconozca automáticamente y no sea necesario indicar estos datos.

Acompañado de cada modelo aparecerá la lista de alumnos. Hay que seleccionar de la lista los alumnos que queremos que hagan los ejercicios. Después existen dos opciones, guardar los ejercicios generados con el código de barras para cada alumno que se haya seleccionado en el disco duro o bien imprimirlos. Cuando los alumnos hayan hecho los ejercicios se introducen en el sistema con ayuda de la interfaz para añadir ejercicios escaneados 4.6

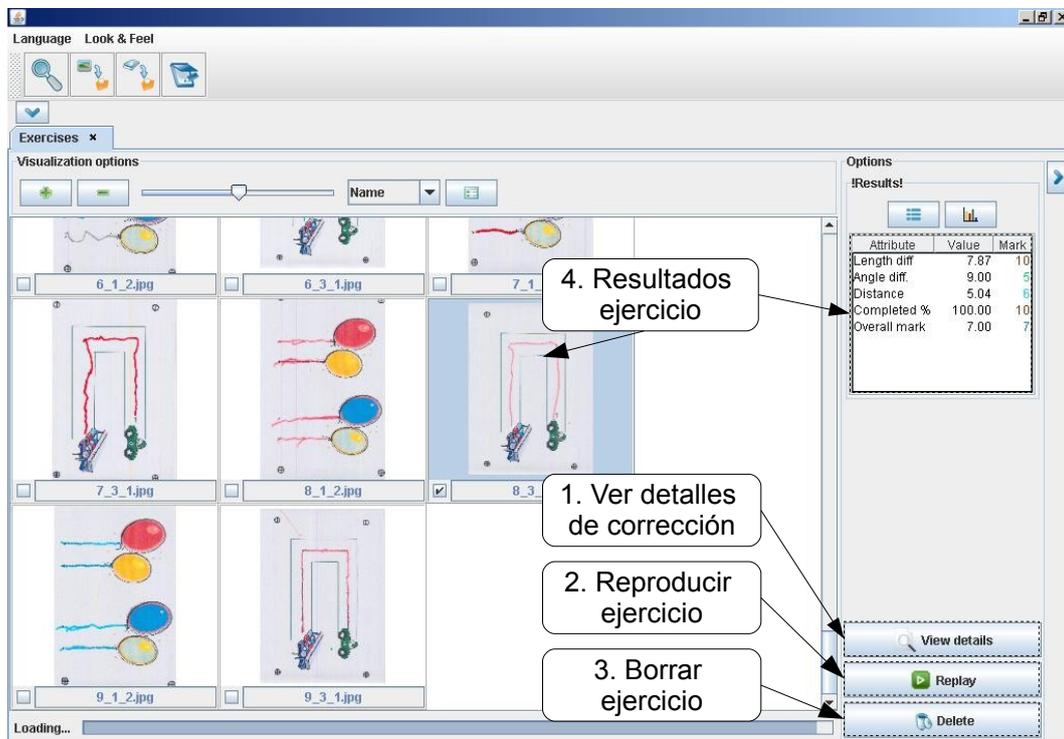


Figura 4.7: Pantalla de búsqueda de ejercicios

#### 4.4.4. Lista de ejercicios

Una vez que los alumnos han realizado ejercicios sobre papel o con la pantalla táctil y se han guardado en el sistema, se pueden hacer búsquedas para ver el conjunto de ejercicios que ha hecho un alumno en concreto o según otros criterios. En la figura 4.7 se puede ver una lista de ejercicios. A través de esta lista se pueden realizar varias acciones, como ver los resultados del ejercicio en detalle, reproducirlo si se ha hecho con la interfaz de dibujo o bien borrarlo del sistema.

##### 1. Ver detalles

A través de esta acción se abrirá una ventana de corrección que se explica en la sección 4.7 para examinar los resultados obtenidos por cada trazo.

##### 2. Reproducir ejercicios

Los ejercicios realizados sobre la pantalla táctil a través de la interfaz de dibujo pueden ser reproducidos para ver exactamente cómo el alumno hizo el ejercicio. La reproducción se realizará sobre la interfaz de dibujo 4.4.5.

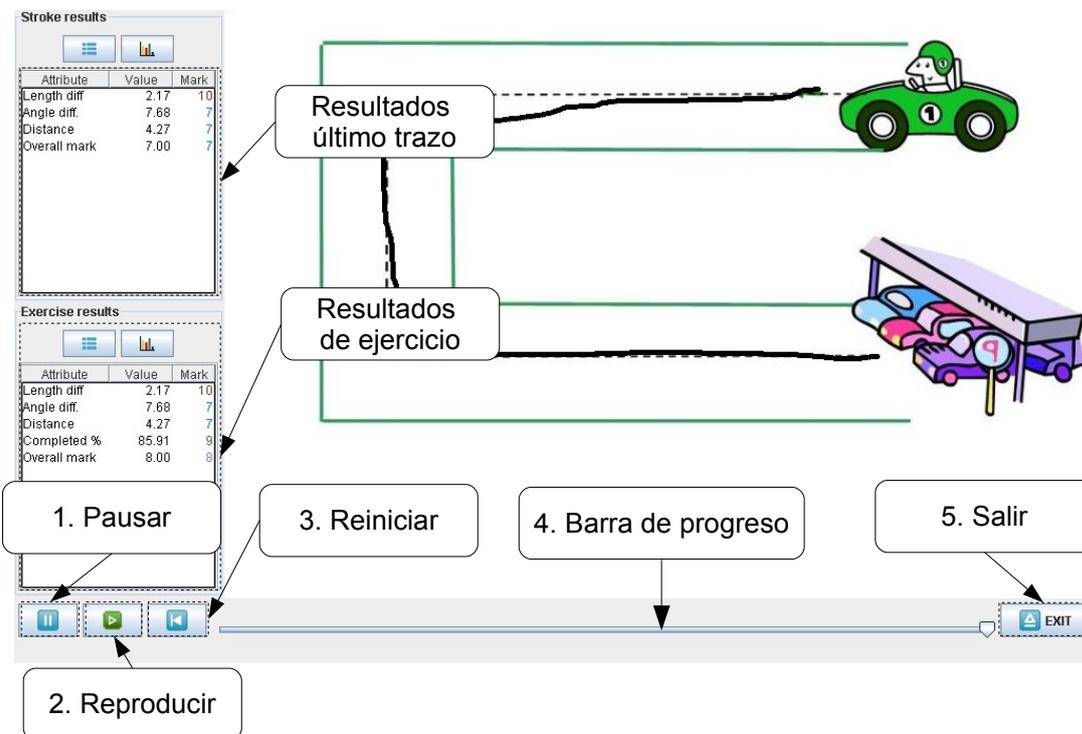


Figura 4.8: Pantalla de reproducción de ejercicios

### 3. Borrar ejercicio

Se borra el ejercicio seleccionado de la lista.

#### 4.4.5. Reproductor de ejercicios

Esta interfaz sirve para reproducir los ejercicios realizados a través de la interfaz de dibujo. Para ello se debe pulsar el botón correspondiente en la lista de ejercicios de la ventana de búsqueda de ejercicios, explicada en la sección 4.4.4. La ventana de reproducción contiene controles para ver el progreso de la reproducción, pausarla y posicionarla en cualquier instante. En la figura 4.8 se muestra la ventana de reproducción.

##### 1. Pausar

Pulsando este botón se interrumpe la reproducción.

##### 2. Reproducir

Sirve para continuar con la reproducción si se ha pausado.

### 3. Reinicio

Vuelve a comenzar la reproducción desde cero.

### 4. Barra de progreso

La barra muestra el progreso de la reproducción del ejercicio y sirve para posicionarse desplazándola en cualquier punto de la reproducción.

### 5. Salir

Si se quiere abandonar el modo de reproducción se pulsa sobre este botón y se volverá al modo de dibujo.

## 4.5. Interfaz de dibujo

Mediante esta interfaz que se muestra en la figura 4.9 se realizan los ejercicios a través de la pantalla táctil y se accede a través de la barra de herramientas. Pulsando el botón para cambiar ejercicios se abre un diálogo del que se pueden seleccionar varios modelos para hacer. Para avanzar de ejercicio se pulsa sobre el botón “Siguiente ejercicio”. Con el botón de herramientas se abre un diálogo que permite seleccionar una brocha de dibujo distinta y cambiar el color del trazo. Para almacenar los ejercicios se debe pulsar el botón de guardar ejercicios.

### 1. Resultados de trazo

Muestra los resultados del último trazo dibujado.

### 2. Resultados de ejercicio

Muestra los resultados globales del ejercicio.

### 3. Herramientas de dibujo

El diálogo de herramientas se abre pulsando sobre el botón de herramientas de dibujo. En este diálogo se pueden realizar las siguientes acciones:

#### a) Grosor del trazo

Desplazando la barra deslizante se puede cambiar el grosor del trazo.

#### b) Panel de color

Pulsando sobre un botón de color se cambia el color del trazo.

#### c) Tipo de trazo

El tipo de trazo se puede cambiar pulsando sobre los botones.

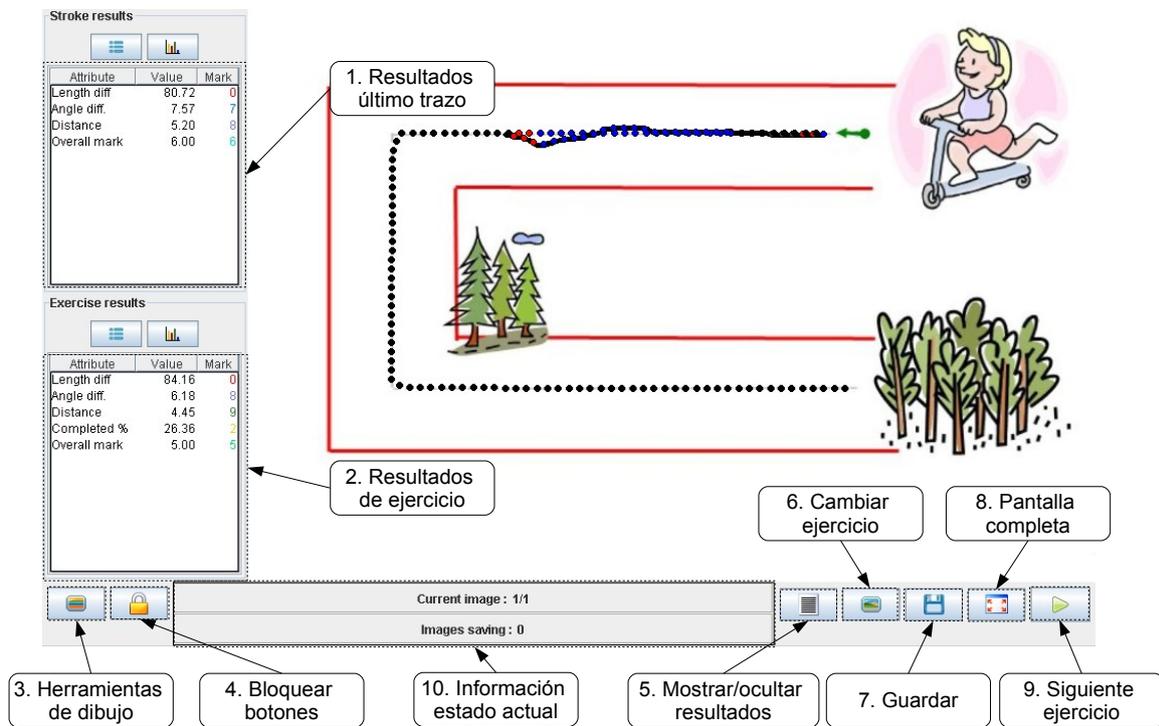


Figura 4.9: Pantalla de realización de ejercicios

#### 4. Bloquear botones

Con este botón se bloquean todos los botones excepto el de pasar de ejercicio y los de selección de color. Si se quiere desbloquear hay que pulsar la tecla CTRL a la vez que se pulsa este botón.

#### 5. Mostrar/ocultar el panel de resultados

Este botón oculta el panel de resultados a la izquierda del ejercicio.

#### 6. Cambiar de ejercicio

Pulsando este botón se abre un panel que permite seleccionar varios modelos para realizar sobre la pantalla táctil y anula el ejercicio que se estuviera haciendo anteriormente.

#### 7. Guardar ejercicios

Abre un diálogo que muestra los ejercicios que se han hecho con la interfaz de dibujo y pulsando sobre ellos se podrán ver los resultados de la corrección de cada uno de ellos. Para almacenarlos en la base de datos se debe seleccionar al alumno que los realizó y pulsar sobre el botón guardar todos o guardar seleccionados.

## 8. Pantalla completa

Este botón pondrá la interfaz en modo a pantalla completa para aprovechar todo el espacio de la pantalla. Si estamos en pantalla completa y se pulsa este botón se volverá al estado normal.

## 9. Pasar al siguiente ejercicio

Pulsando el botón de siguiente ejercicio se realiza la corrección del ejercicio y se almacena temporalmente el resultado. Se cargará el siguiente ejercicio de la lista que se haya seleccionado para hacer. Para almacenar permanentemente los ejercicios se tiene que pulsar el botón de guardar ejercicios.

## 10. Información estado actual

Muestra información sobre el número de ejercicios realizados.

## 4.6. Añadir ejercicios escaneados

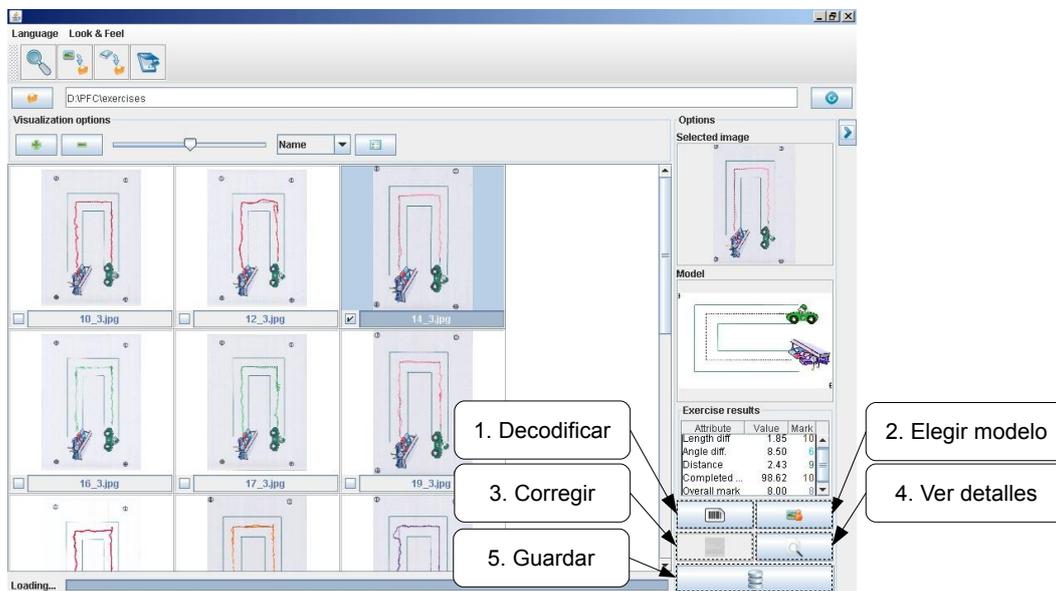


Figura 4.10: Pantalla de ejercicios escaneados

Si se quieren corregir y añadir ejercicios realizados sobre papel hay que usar la pantalla de añadir ejercicios escaneados, a la que se accede a través de la barra de herramientas. A continuación se selecciona la carpeta en la que se encuentran. Si los ejercicios contienen códigos de barra se identificará de que modelo se trata y el alumno que lo ha realizado

automáticamente pulsando el botón de decodificación. Si no contienen códigos de barras o no se pudiera reconocer se pueden introducir estos datos de forma manual.

1. Decodificar

El sistema intentará decodificar el código de barras y asignar un alumno y un modelo al ejercicio. Si no se puede decodificar se podrá introducir estos datos manualmente.

2. Seleccionar un modelo y un alumno para un ejercicio escaneado

Si los ejercicios no incluyen un código de barras se pueden seleccionar estos datos pulsando sobre un ejercicio de la lista y después sobre el botón de selección de modelo. Se abrirá un diálogo con el que se podrán seleccionar un modelo de la lista y el alumno de la tabla.

3. Corregir

Para corregir los ejercicios se seleccionan en la lista y se pulsa el botón de corregir, lo que iniciará el proceso de corrección. Una vez finalizado se podrán ver los resultados en el panel de resultados.

4. Ver detalles

Pulsando en este botón se abre una ventana con los detalles de la corrección. Esta ventana se explica en la sección 4.7.

5. Guardar

Se guarda la imagen del ejercicio con los resultados y el log con los trazos que se han extraído, asociándose al alumno que se haya seleccionado o que se haya obtenido a través del código de barras.

## 4.7. Ventana de corrección

En esta ventana se muestran los resultados de la corrección detalladamente. Los trazos se dibujan con un color en función de la nota obtenida. Si se pulsa con el ratón cerca de uno se ven los resultados de la corrección para ese trazo.

1. Leyenda de notas

En este panel se muestra la relación entre las notas y los colores de los trazos.

2. Resultados de trazo

En este panel podemos ver los resultados que cada trazo ha obtenido en la corrección del ejercicio. Para mostrar los resultados hay que pulsar cerca del trazo en la imagen central.

3. Resultados de ejercicio

Aquí se ven los resultados globales para el ejercicio.

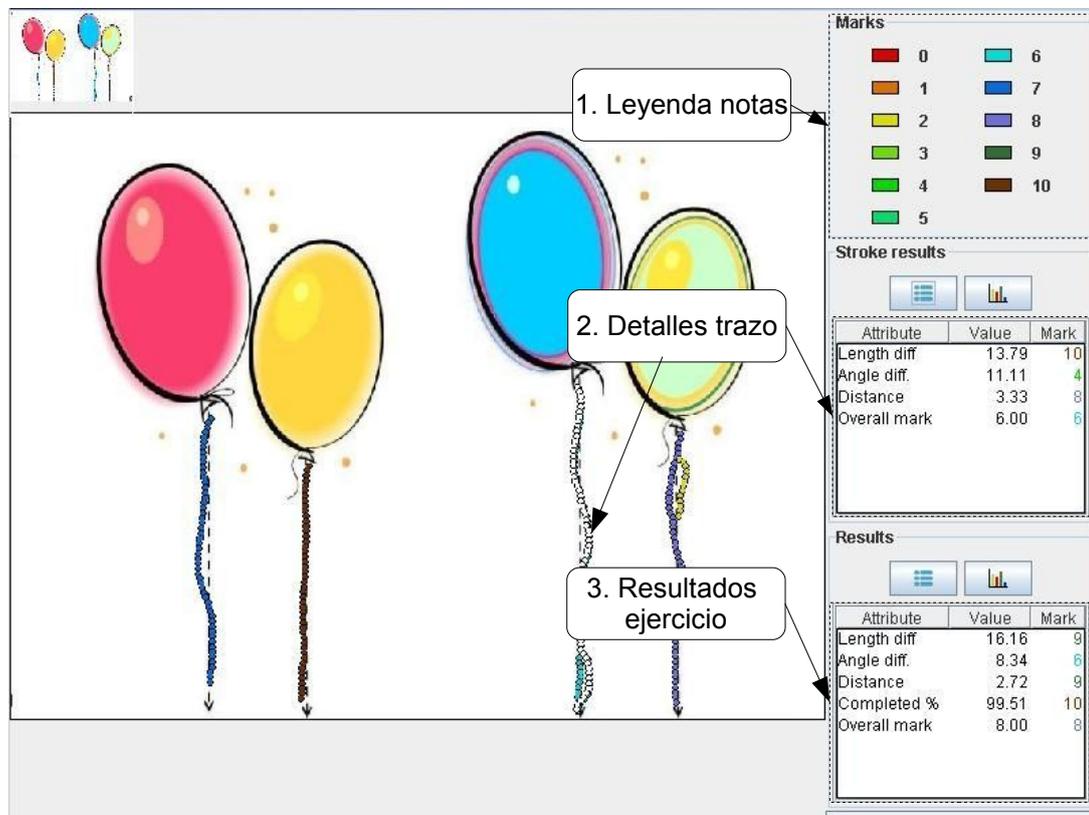


Figura 4.11: Ventana de corrección

## 4.8. Gráficas de resultados

Si queremos ver un resumen de las calificaciones que han obtenido los alumnos se dispone de dos tipos de gráficas, una que muestra la nota media de todos los ejercicios que han realizado y otra que muestra el progreso del alumno a lo largo del tiempo. Con estas gráficas se puede ver si algún alumno no tiene un progreso adecuado o tiene dificultades con algún tipo de ejercicio o trazo. Para que se muestren estas gráficas hay que pulsar sobre los botones de gráficas en la pantalla de búsqueda. En la figura 4.12 se muestra una gráfica con las notas medias de varios alumnos y en la figura 4.13 la gráfica muestra el progreso de un único alumno.

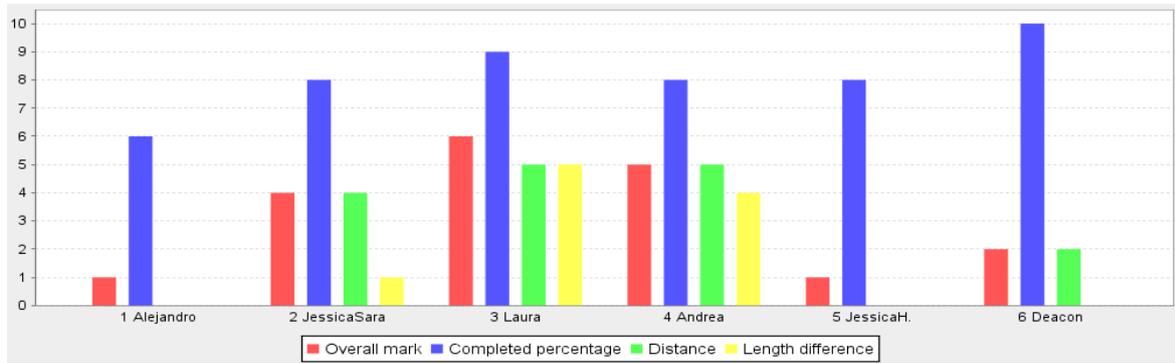


Figura 4.12: Gráfica de notas medias

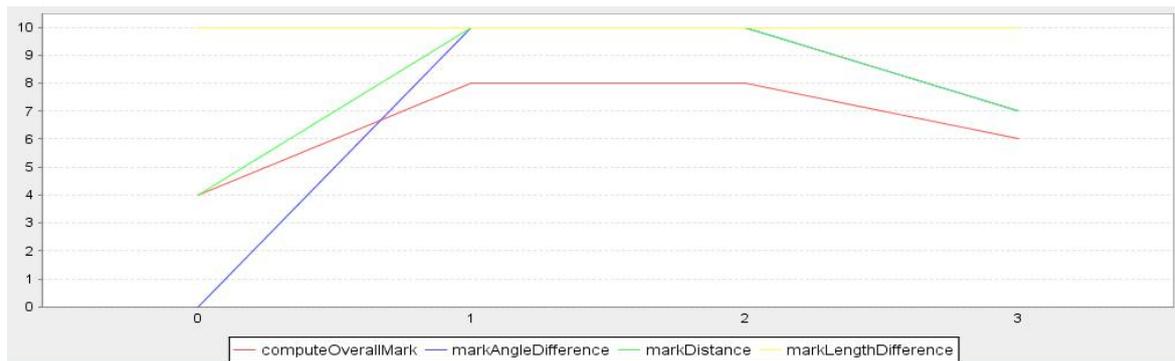


Figura 4.13: Gráfica de progreso

# Capítulo 5

## Diseño e implementación del sistema

Siguiendo los objetivos enunciados en el capítulo 3, se ha diseñado e implementado un sistema que permite que se puedan realizar ejercicios y corregirlos de forma automática. En este capítulo se describen todos los algoritmos creados.

Para poder abordar este sistema es preciso conocer una serie de conceptos que se explican en la sección 5.1. En la sección 5.2 se explican las tecnologías y los paquetes de software que se han utilizado para desarrollar este proyecto. Los algoritmos que creados para el procesado de los ejercicios escaneados son definidos en el capítulo 5.4. En la sección 5.5 se explica como se codifica el código de barras usado en los ejercicios. Todos los algoritmos utilizados para extraer los trazos de los ejercicios escaneados y corregirlos son definidos en las secciones 5.6 y 5.7. En la última sección 5.8 se explican las estructuras de datos usadas en el almacenamiento de los ejercicios.

### 5.1. Conceptos previos

En esta sección se explican algunos conceptos básicos necesarios para comprender el funcionamiento del sistema.

#### 5.1.1. Modelo de color RGB

La descripción RGB (del inglés Red, Green, Blue; “rojo, verde, azul”) de un color hace referencia a la composición del color en términos de la intensidad de los colores primarios con que se forma: el rojo, el verde y el azul. Es un modelo de color basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores luz primarios. Indicar que el modelo de color RGB no define por sí mismo lo que significa exactamente rojo, verde o azul, razón por la cual los mismos valores RGB pueden mostrar colores notablemente diferentes en diferentes dispositivos que usen este modelo de color. Aunque utilicen un mismo modelo de color, sus espacios de color pueden variar considerablemente [19].

Para indicar con qué proporción mezclamos cada color, se asigna un valor a cada uno de los colores primarios, de manera, por ejemplo, que el valor 0 significa que no interviene en la mezcla y, a medida que ese valor aumenta, se entiende que aporta más intensidad a la mezcla. Aunque el intervalo de valores podría ser cualquiera (valores reales entre 0 y 1, valores enteros entre 0 y 37, etc.), es frecuente que cada color primario se codifique con un byte (8 bits). Así, de manera usual, la intensidad de cada una de las componentes se mide según una escala que va del 0 al 255.

Por lo tanto, el rojo se obtiene con (255,0,0), el verde con (0,255,0) y el azul con (0,0,255), obteniendo, en cada caso un color resultante monocromático. La ausencia de color, lo que nosotros conocemos como color negro, se obtiene cuando las tres componentes son 0, (0,0,0). La combinación de dos colores a nivel 255 con un tercero en nivel 0 da lugar a tres colores intermedios. De esta forma el amarillo es (255,255,0), el cian (0,255,255) y el magenta (255,0,255). Obviamente, el color blanco se forma con los tres colores primarios a su máximo nivel (255,255,255).

### Codificación hexadecimal del color

La codificación hexadecimal del color permite expresar fácilmente un color concreto de la escala RGB, utilizando la notación hexadecimal. Este sistema utiliza la combinación de tres códigos de dos dígitos para expresar las diferentes intensidades de los colores primarios RGB (Red, Green, Blue, rojo, verde y azul).

El blanco y el negro se codifican de la siguiente manera:

- Negro 0x000000  Los tres canales están al mínimo 00, 00 y 00
- Blanco 0xfffff  Los tres canales están al máximo ff, ff y ff

En el sistema de numeración hexadecimal, además de los números del 0 al 9 se utilizan seis letras con un valor numérico equivalente; a=10, b=11, c=12, d=13, e=14 y f=15. La correspondencia entre la numeración hexadecimal y la decimal u ordinaria viene dada por la siguiente fórmula:

$$\text{decimal} = \text{primera cifra hexadecimal} * 16 + \text{segunda cifra hexadecimal}$$

La intensidad máxima es ff, que se corresponde con  $(15*16)+15= 255$  en decimal, y la nula es 00, también 0 en decimal. De esta manera, cualquier color queda definido por tres pares de dígitos.

## 5.2. Descripción tecnológica

Para implementar el sistema se ha utilizado el lenguaje de programación Java y MySQL como gestor de base de datos. Además el sistema se basa en otros proyectos que se detallarán más adelante. Las descripciones que se incluyen en las siguientes secciones sobre Java, JDBC y bases de datos relacionales son un resumen de las que aparecen en la Wikipedia para estas tecnologías (véanse las referencias [20], [21], [22]).

### 5.2.1. Java

La Plataforma Java se compone de un amplio abanico de tecnologías, cada una de las cuales ofrece una parte del complejo de desarrollo o del entorno de ejecución en tiempo real. Por ejemplo, los usuarios finales suelen interactuar con la máquina virtual de Java y el conjunto estándar de bibliotecas. Además, las aplicaciones Java pueden usarse de forma variada, como por ejemplo ser incrustadas en una página Web. Para el desarrollo de aplicaciones, se utiliza un conjunto de herramientas conocidas como JDK (Java Development Kit, o herramientas de desarrollo para Java).

#### Java Runtime Environment

Un programa destinado a la Plataforma Java necesita dos componentes en el sistema donde se va a ejecutar: una máquina virtual de Java (JVM), y un conjunto de librerías para proporcionar los servicios que pueda necesitar la aplicación. La JVM que proporciona Sun Microsystems, junto con su implementación de las librerías estándar, se conocen como Java Runtime Environment (JRE) o Entorno en tiempo de ejecución para Java. El JRE es lo mínimo que debe contener un sistema para poder ejecutar una aplicación Java sobre el mismo.

#### Máquina Virtual de JAVA

El corazón de la Plataforma Java es el concepto común de un “procesador virtual” que ejecuta programas escritos en ese lenguaje de programación. En concreto, ejecuta el código resultante de la compilación del código fuente, conocido como bytecode. Este “procesador” es la máquina virtual de Java o JVM (Java Virtual Machine), que se encarga de traducir (interpretar o compilar al vuelo) el bytecode en instrucciones nativas de la plataforma destino. Esto permite que una misma aplicación Java pueda ser ejecutada en una gran variedad de sistemas con arquitecturas distintas, siempre que haya una implementación adecuada de la JVM. Este hecho es lo que ha dado lugar a la famosa frase: “write once, run anywhere” (escribir una vez, ejecutar en cualquier parte). La condición es que no se utilicen llamadas nativas o funciones específicas de una plataforma.

#### Librerías de Java

En la mayoría de los sistemas operativos actuales, se ofrece una cantidad de código para simplificar la tarea de programación. Este código toma la forma, normalmente, de un conjunto de librerías dinámicas que las aplicaciones pueden llamar cuando lo necesiten. Pero la Plataforma Java está pensada para ser independiente del sistema operativo subyacente, por lo que las aplicaciones no pueden apoyarse en funciones dependientes de cada sistema en concreto. Lo que hace la Plataforma Java, es ofrecer un conjunto de librerías estándar, que contiene mucha de las funciones reutilizables disponibles en los sistemas operativos actuales.

Las librerías de Java tienen tres propósitos dentro de la Plataforma Java. Al igual que otras librerías estándar, ofrecen al programador un conjunto bien definido de funciones para realizar tareas comunes, como manejar listas de elementos u operar de forma sofisticada sobre cadenas de caracteres. Además, las librerías proporcionan una interfaz abstracta para tareas que son altamente dependientes del hardware de la plataforma destino y de su sistema operativo. Tareas tales como manejo de las funciones de red o acceso a ficheros, suelen depender fuertemente de la funcionalidad nativa de la plataforma destino. En el caso concreto anterior, las librerías `java.net` y `java.io` implementan el código nativo internamente, y ofrecen una interfaz estándar para que aplicaciones Java puedan ejecutar tales funciones. Finalmente, no todas las plataformas soportan todas las funciones que una aplicación Java espera. En estos casos, las librerías bien pueden emular esas funciones usando lo que esté disponible, o bien ofrecer un mecanismo para comprobar si una funcionalidad concreta está presente.

### 5.2.2. MySQL

MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales que se desarrolla como software de código abierto. A continuación se explicará brevemente el concepto de base de datos relacional y el lenguaje SQL, que sirve para acceder a ella.

#### Base de datos relacional

Una base de datos relacional es un conjunto de una o más tablas estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común, en ambos casos posee las mismas características como por ejemplo el nombre de campo, tipo y longitud; a este campo generalmente se le denomina ID, identificador o clave. A esta manera de construir bases de datos se le denomina modelo relacional.

Estrictamente hablando el término se refiere a una colección específica de datos pero a menudo se le usa, en forma errónea como sinónimo del software usado para gestionar esa colección de datos. Ese software se conoce como SGBD (sistema gestor de base de datos) relacional o RDBMS (del inglés relational database management system).

Su idea fundamental es el uso de “relaciones”. Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados “tuplas”. Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar, esto es, pensando en cada relación como si fuese una tabla que está compuestas por registros (cada fila de la tabla sería un registro), que representarían las tuplas, y campos (las columnas de una tabla).

#### Lenguaje SQL

El Lenguaje de consulta estructurado (SQL Structured Query Language ) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de

operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos, así como también hacer cambios sobre la misma.

### 5.2.3. MySQL Connector/JDBC

MySQL Connector/J es un driver de Java que convierte las llamadas JDBC (Java Database Connectivity) al protocolo de red usado por la base de datos MySQL.

#### JDBC

Java Database Connectivity, más conocida por sus siglas JDBC, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.

El API JDBC se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de base de datos. Un manejador de conexiones hacia un modelo de base de datos en particular es un conjunto de clases que implementan las interfaces Java y que utilizan los métodos de registro para declarar los tipos de localizadores a base de datos (URL) que pueden manejar. Para utilizar una base de datos particular, el usuario ejecuta su programa junto con la librería de conexión apropiada al modelo de su base de datos, y accede a ella estableciendo una conexión, para ello provee en localizador a la base de datos y los parámetros de conexión específicos. A partir de allí puede realizar con cualquier tipo de tareas con la base de datos a las que tenga permiso: consulta, actualización, creación, modificación y borrado de tablas, ejecución de procedimientos almacenados en la base de datos, etc.

### 5.2.4. Otros paquetes

Para implementar el sistema se ha hecho uso de varias librerías de código abierto, algunas de las cuales se han modificado para aumentar su funcionalidad o bien para adaptarlas al sistema.

- **JBarcode**

Para generar los códigos de barras que se imprimen con los ejercicios se ha utilizado la librería JBarcode. JBarcode es una librería basada en Java que permite generar imágenes de códigos de barras (EAN13, EAN8, UPCA, UPCE, Code 3 of 9, Codabar, Code 11, Code 93, Code 128, MSI/Plessey, Interleaved 2 of 5, PostNet, etc) usando la API de Java Java2D. El código que se usa en el sistema es el EAN13, que se explica detalladamente en el capítulo 5.5.

- **Lector código de barras**

El paquete que se encarga de decodificar los códigos de barras está basado en el proyecto ReadBarJ, que es una implementación de reconocimiento para códigos de barras EAN para móviles. Dicho proyecto se ha modificado para que funcione con el JRE y se ha adaptado a las necesidades del programa.

- **Paint**

La interfaz para pintar a través de la pantalla táctil está basada en la applet de Java SinglePaint. Se le han añadido funciones para cargar imágenes de fondo, reescalar automáticamente la imagen de fondo para adaptarla al tamaño de la pantalla y para guardar el progreso de dibujo en un fichero para poder reproducirlo después.

- **Filtros**

Parte de los filtros para el tratamiento de las imágenes están tomados del editor de imágenes Java Image Editor. Estos filtrados son usados principalmente para procesar las imágenes escaneadas.

- **Freechart**

JFreeChart es una librería libre para crear gráficas en la plataforma Java. Puede generar diagramas de barras, diagramas circulares, gráficas de funciones, histogramas etc. Esta librería se ha usado para crear las gráficas que muestran los resultados de los ejercicios.

- **Tulliana**

Los iconos usados en el programa se han tomado del pack de iconos Tulliana o han sido creados a partir de ellos.

## 5.3. La Base de Datos

En la base de datos se almacena el modelo del alumno y todos los datos relacionados con los ejercicios y los modelos. En la tabla ejercicio se almacena el autor del ejercicio, el modelo al que pertenece, la fecha, el nombre del fichero en el que está guardado el ejercicio etc. La tabla modelo contiene los nombres de los ficheros que contienen la imagen punteada y normal del modelo, la dificultad, la edad para lo que está pensado etc. La tabla alumno contiene todos los datos relativos al alumno. En la figura 5.1 se puede ver el esquema de la base de datos.

### 5.3.1. Tabla del Alumno

El perfil del alumno se almacena en la tabla alumno dentro de la base de datos. En esta tabla se almacenarán sus datos personales y los ejercicios que realicen se asociarán con esta tabla.

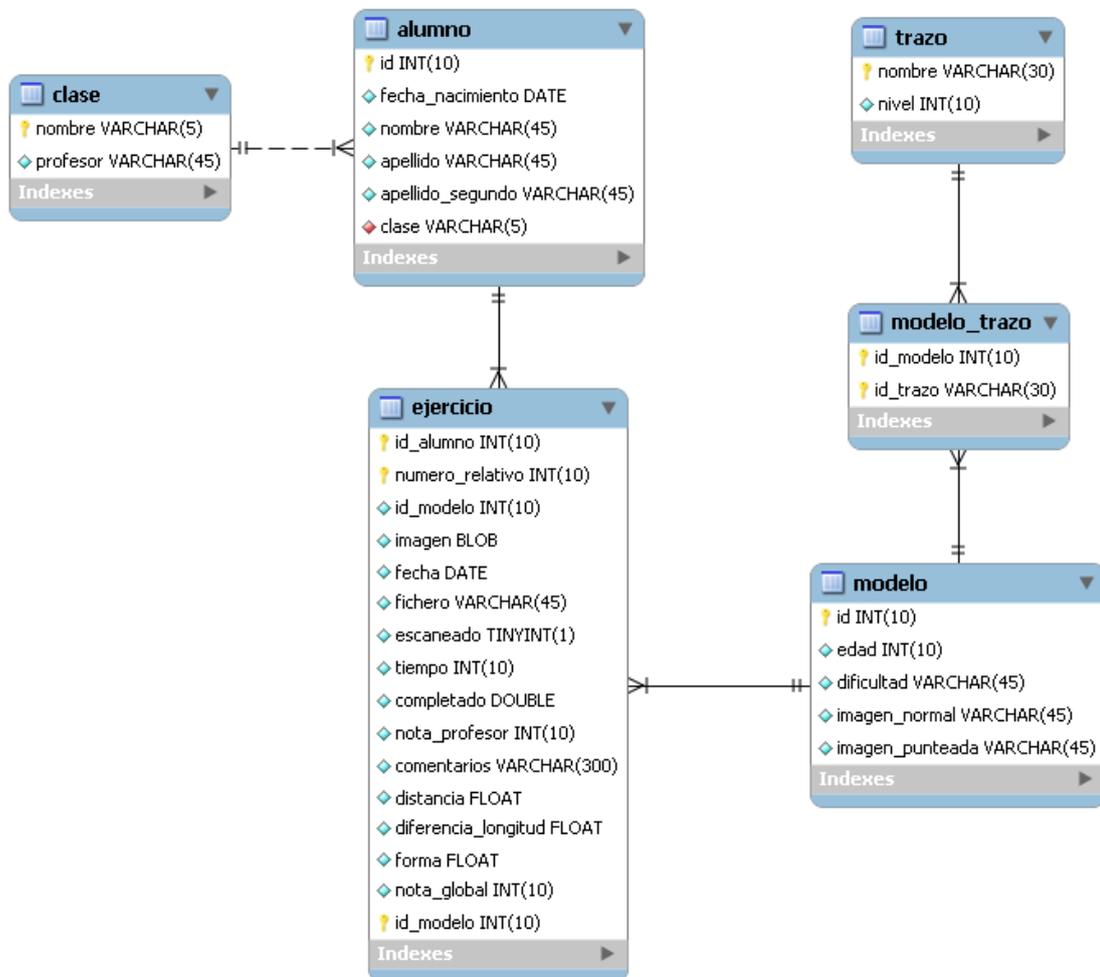


Figura 5.1: Esquema de la base de datos

### 5.3.2. Tabla de ejercicios

Para almacenar las imágenes con los ejercicios se guarda una referencia al fichero que contiene la imagen en la tabla de ejercicios. También se guardarán los resultados de las calificaciones. A partir de estas tablas se podrá inferir el estado de conocimiento del alumno, las calificaciones obtenidas para cada ejercicio y notas medias para los distintos tipos de trazo o modelo.

- Almacenamiento de log

Los ejercicios se almacenarán en ficheros de texto. Cada línea del fichero constituirá un punto del trazo, y llevará un identificador y una marca de tiempo asociada para cada tipo de punto, es decir si es un comienzo, final o punto intermedio.

- Almacenamiento imagen

Las imágenes se almacenarán en ficheros JPG, y el nombre del fichero se formará a partir del identificador del alumno, el del modelo y el número relativo de ejercicio del alumno. Por ejemplo, si un alumno con id 1 realiza un ejercicio del modelo con id 3 y es su cuarto ejercicio el nombre con el que se almacena el fichero será 1\_3\_4.

### 5.3.3. Tabla de modelo

Para almacenar los modelos se tienen en la tabla modelo dos referencias, una a la imagen punteada y otra al log de modelo. Además la tabla contiene la dificultad y los trazos que contiene el modelo.

## 5.4. Procesado de ejercicios escaneados

En esta sección se detallan los procesos descritos en el capítulo 3. Esto incluye los algoritmos de detección de marcas, filtrado de imágenes y extracción y vectorización de los trazos.

### 5.4.1. Detección de las marcas

Para detectar la posición de las marcas se usa un algoritmo de “inundación” con el que se etiqueta cada símbolo que aparece en la imagen escaneada. Un símbolo es cualquier elemento conexo de la imagen, ya sea un trazo o dibujo, y serán etiquetados con un número distinto. Se obtendrán gran cantidad de símbolos, de entre los que son marcas del resto.

Antes de poder detectar las marcas hay que pasar la imagen escaneada a blanco y negro y filtrar el ruido que se haya podido crear al escanear la imagen. Para ello se usan el filtro de umbral y de ruido.

- Filtrado de umbral

Este filtro transforma una imagen en color a una en blanco y negro. Para cada píxel de la imagen se comprueba si sumando las tres componentes RGB se supera un cierto valor, en ese caso el píxel se pasa a negro, si no lo superara se convertiría en un píxel blanco obteniendo de esta forma una imagen binaria.

- Filtrado de ruido

Este filtro elimina el ruido que pudiera dificultar la detección de las marcas. Si encontramos un píxel en la imagen que tenga un número de vecinos de distinto color al suyo el píxel adoptaría el color de sus vecinos. En la figura 5.2 podemos ver un ejemplo de una marca filtrada.

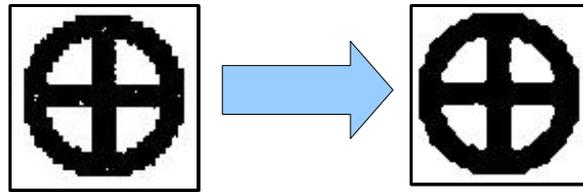


Figura 5.2: Filtro de ruido aplicado a una marca

-1	-1	0	-1	-1	-1	1	-1	0	-1	-1	-1	1	1	0	-1	-1	-1
-1	-1	0	-1	0	0	-1	-1	0	-1	0	0	-1	-1	0	-1	0	0
0	-1	0	-1	0	0	0	-1	0	-1	0	0	0	-1	0	-1	0	0
(a) Paso 1						(b) Paso 2						(c) Paso 3					
1	1	0	-1	-1	-1	1	1	0	-1	-1	-1	1	1	0	2	-1	-1
1	-1	0	-1	0	0	1	1	0	-1	0	0	1	1	0	-1	0	0
0	-1	0	-1	0	0	0	1	0	-1	0	0	0	1	0	-1	0	0
(d) Paso 4						(e) Paso 5						(f) Paso 6					
1	1	0	2	2	-1	1	1	0	2	2	-1	1	1	0	2	2	2
1	1	0	-1	0	0	1	1	0	2	0	0	1	1	0	2	0	0
0	1	0	-1	0	0	0	1	0	-1	0	0	0	1	0	-1	0	0
(g) Paso 7						(h) Paso 8						(i) Paso 9					
1	1	0	2	2	2												
1	1	0	2	0	0												
0	1	0	2	0	0												
(j) Paso 10																	

Tabla 5.1: Etiquetado de símbolos

El algoritmo de inundación funciona de la siguiente manera [23]:

En primer lugar debemos convertir la imagen en una imagen binaria aplicando el filtro de umbral y de ruido. La imagen generada se recorre de izquierda a derecha y de arriba hacia abajo, de forma que cuando se encuentre un píxel negro se marcarán ese píxel y los vecinos con un número. A los vecinos se les aplicará el mismo algoritmo recursivamente. Por cuestiones de eficiencia se usa una pila para simular la recursión. Una vez finalizado se seguirá recorriendo la imagen y no se tendrán en cuenta los píxeles marcados. En la tabla 5.1 se muestra el algoritmo aplicado a una imagen pequeña. Las casillas con valor -1 representan píxeles negros y las que tienen valor 0 blancos.

Cuando el proceso de marcado ha finalizado se filtran de entre todos los símbolos los que

	Marca izquierda	Marca derecha
Ratio	1	1
Agujeros	4	2
Área	0.431111	0.511111

Tabla 5.2: Tabla de características de las marcas

son marcas. Para ello tenemos que comprobar si cumplen las propiedades que hacen de un símbolo una marca:

- Relación altura / anchura: si se divide la altura por la anchura del símbolo debe dar un valor próximo a 1 para que se trate de una marca.
- Porcentaje de negro: es el porcentaje de área cubierta por píxeles negros. Las marcas izquierdas tienen menos porcentaje de negro que las derechas.
- N° de agujeros: Las marcas izquierdas tienen dos y las derechas cuatro agujeros.

En la tabla 5.2 se muestran los valores de las características mencionadas anteriormente.

Puesto que el cálculo de agujeros es computacionalmente más costoso que el del resto de atributos se realizarán antes el cálculo de estos últimos, con lo que se descartarán la mayoría de símbolos y el algoritmo tardará menos.

### Recolocar la imagen

En el caso de que se hayan detectado el mismo número de marcas izquierdas y derechas podremos continuar. Ahora debemos calcular los grados que está girada la imagen y en que sentido. Para calcular los grados en que hay que girar la imagen en el sentido de las agujas del reloj hay que tener en cuenta las cuatro posibles estados en que puede estar girada la imagen. Para ello se usan las siguiente fórmulas:

Marca izquierda esté por encima de la derecha (Figura 5.3):

- Marca izquierda está a la izquierda de la marca derecha

En este caso la imagen está rotada entre 0 y 90 grados.

$$\alpha = 2 * \pi - \arctan(a/c)$$

- Marca izquierda está a la derecha de la marca derecha:

La imagen está rotada entre 90 y 180 grados.

$$\alpha = \arctan(a/c) + 2 * (\pi/2)$$

Marca izquierda esté por debajo de la derecha (Figura 5.4):

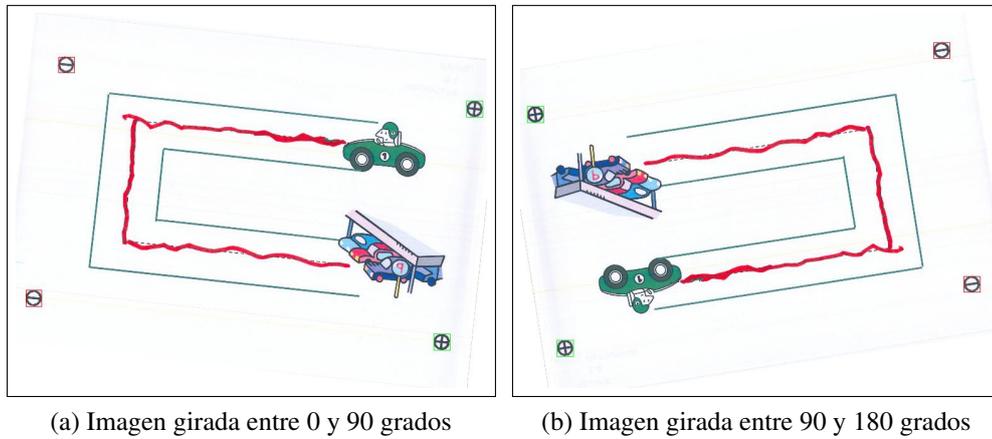


Figura 5.3: Marca izquierda por encima de la derecha

- Marca izquierda está a la izquierda de la marca derecha  
 En este caso la imagen está rotada entre 270 y 360 grados.  
 $\alpha = \arctan(a/c)$
- Marca izquierda está a la derecha de la marca derecha:  
 La imagen está rotada entre 180 y 270 grados.  
 $\alpha = (\pi/2) - \arctan(a/c) + (\pi/2)$

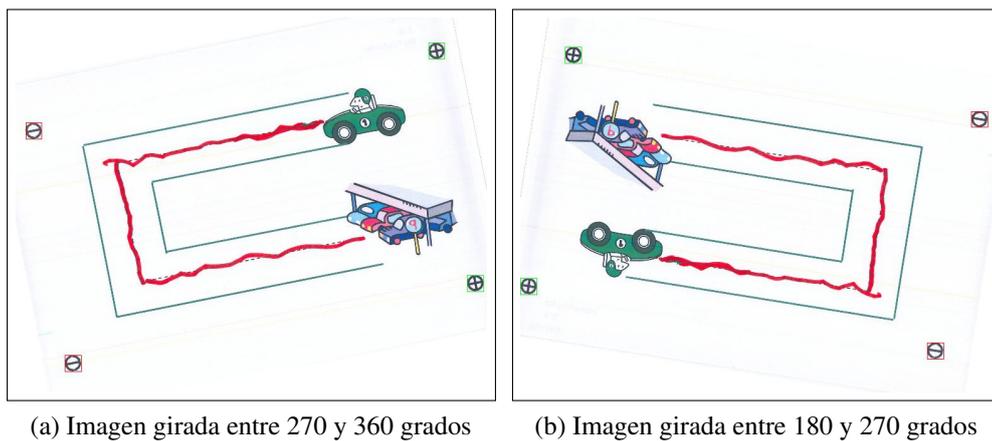


Figura 5.4: Marca izquierda esté por debajo de la derecha

Una vez recolocada la imagen se recortan las imágenes comprendidas entre las marcas. Para ella se toman las subimágenes entre la esquina inferior derecha de la marca superior izquierda y la esquina superior izquierda de la marca inferior derecha.

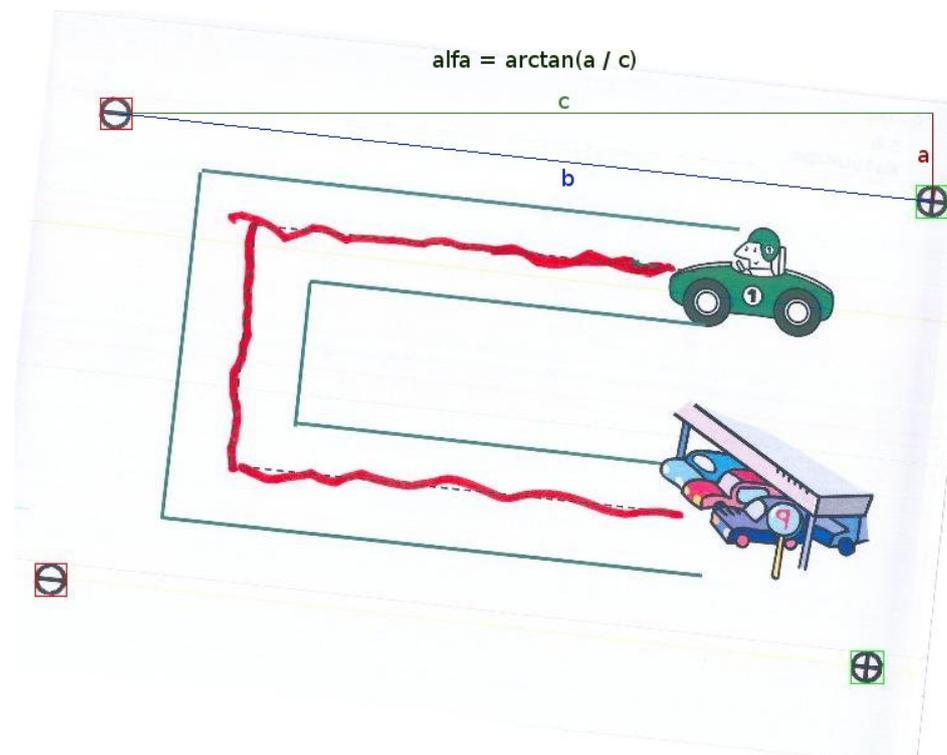


Figura 5.5: Cálculo del ángulo de rotación

### 5.4.2. Filtrado de fondo

Para realizar este filtrado se examina la imagen y se crea su histograma, que representa la frecuencia con la que cada valor de los canales RGB aparece en la imagen. Los píxeles que tengan un valor que se aproxime más al valor máximo del histograma se pasarán a blanco, ya que se considerarán como píxeles de fondo. En la figura 3.7 vemos como queda la imagen después de haber aplicado el filtro de fondo sobre ella. Este filtro conserva todos los trazos, incluso los colores claros como amarillos o grises.

En la figura 5.6 vemos una representación del histograma de una imagen escaneada de un ejercicio. En el eje X aparecen los valores de cada canal entre un rango de 0 a 255. En el eje Y vemos la frecuencia de cada valor en los píxeles de la imagen. Vemos que la frecuencia máxima para el canal rojo y verde está aproximadamente entre 225 y 230, y de entre 235 y 240 para el azul. Los valores son parecidos entre sí, por lo que estamos ante una tonalidad de gris. El color representa al color de fondo que es el que queremos eliminar, por lo tanto todos los píxeles que tengan un valor próximo a éste serán convertidos a blanco.

## 5.5. Decodificación del código de barras

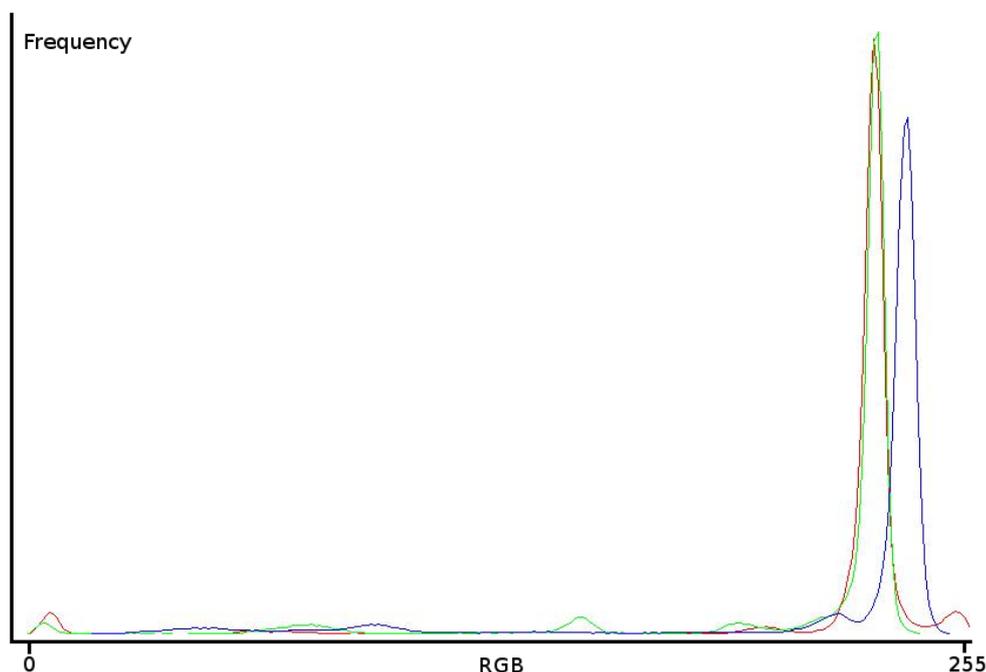


Figura 5.6: Histograma de un ejercicio

El código de barras utilizado es del tipo EAN-13. Para decodificar el código de barras se escanea la imagen línea por línea y se estima el código tomando las mejores muestras, ya que el código de barras puede estar deformado. El dígito de verificación sirve para comprobar que la decodificación se haya hecho correctamente.

Gracias a la información contenida en el código de barras sabrá que alumno ha hecho el ejercicio y de que modelo se trata. El código puede contener 12 dígitos que contendrán la id del modelo y del alumno al que se le ha asignado el ejercicio. Si no se pudiera decodificar habría que introducir estos datos manualmente para poder realizar la corrección. En la figura 5.7 vemos un código de barras de un ejercicio que contiene el identificador del modelo del ejercicio y del alumno que lo ha realizado.



Figura 5.7: Código de barras de un ejercicio

### 5.5.1. Codificación EAN-13

La información sobre este tipo de codificación ha sido recopilada de la web (véase la referencia [24]). Estos códigos tienen las siguientes características:

- Cada carácter se representa con 7 elementos que consisten en 2 barras y dos espacios. Ningún espacio o barra puede ser mayor de 4 elementos. La única excepción son las zonas de guarda izquierdas y derechas (3 elementos) y la barra central (5 elementos).

- Los caracteres se codifican dependiendo de la posición del caracter dentro del código de barras, izquierda o derecha, y del primer dígito del código.
- Todos los caracteres de la parte izquierda del código comienzan con 0 mientras que los de la parte derecha con un 1.
- La codificación de los caracteres de la parte derecha es exactamente igual que la codificación impar de la parte izquierda, salvo que los ceros están cambiados por unos y viceversa.

Un ejemplo de código EAN-13 se ve en la figura 5.8, en la que se pueden observar las partes que componen el código. Algunas características generales son las siguientes:

1. Zona de guarda: espacio blanco al comienzo y al final del código.
2. Código de comienzo y fin: es el código 101 o barra-espacio-barra que indica donde comienza y termina el código.
3. Indicador central: se forma con 01010 y divide el código por la mitad.
4. Dígito de verificación: es el ultimo dígito, se calcula en función de los restantes, y sirve para verificar que el código leído es correcto.

### 5.5.2. Tabla de codificación

Esta tabla 5.3 indica como codificar cada dígito de un código de barras EAN-13 dependiendo en qué mitad esté. En el caso de que esté a la izquierda, la codificación (par o impar) se basa en el valor del primer dígito (ver tabla 5.4). Para codificar un numero hay que mirar el primer dígito y después la tabla de paridad para ver como se codifica la primera mitad. Después se mira la tabla de codificación para codificar cada uno de los caracteres. El último dígito de verificación se calcula en función de los 12 primeros.

### 5.5.3. Tabla de paridad

La tabla de paridad 5.4 indica como se codifica cada caracter de la parte izquierda del código. Ésta se basa en el primer dígito del código EAN-13.

### 5.5.4. Ejemplo de codificación

En este ejemplo se codificará el valor “7501031311309” como un código de barras EAN-13. Para calcular el dígito de verificación se multiplican los número en posición impar por tres y se suman a los dígitos en posición par. En la tabla 5.5 se muestran los cálculos necesarios:

Dígito	Codificación izquierda		Codificación derecha
	Paridad impar	Paridad par	Todos
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

Tabla 5.3: Tabla de codificación EAN-13

1º DIGITO	Codificación izquierda					
	2º DIGITO	Resto del código				
		1	2	3	4	5
0	impar	impar	impar	impar	impar	impar
1	impar	impar	par	impar	par	par
2	impar	impar	par	par	impar	par
3	impar	impar	par	par	par	impar
4	impar	par	impar	impar	par	par
5	impar	par	par	impar	impar	par
6	impar	par	par	par	impar	impar
7	impar	par	impar	par	impar	par
8	impar	par	impar	par	par	impar
9	impar	par	par	impar	par	impar

Tabla 5.4: Tabla de paridad EAN-13

Si se suman los dígitos se produce el siguiente resultado  $7 + 15 + 0 + 3 + 0 + 9 + 1 + 9 + 1 + 3 + 3 + 0 = 51$ . Hay que sumar 9 para que sea múltiplo de 10, por lo tanto el dígito de verificación será 9.

Lo siguiente es observar el primer dígito que en este caso es un 7. En la tabla de paridad hay que mirar la codificación a usar, en este caso “Impar/Par/Impar/Par/Impar/Par”. El código se construye concatenando todos los códigos. En la tabla 5.6 se muestra como se construye el código.

El código que se genera se muestra en la figura 5.8, así como las distintas partes que componen el código:

Código	7	5	0	1	0	3	1	3	1	1	3	0
Posición	E	O	E	O	E	O	E	O	E	O	E	O
Peso	1	3	1	3	1	3	1	3	1	3	1	3
Cálculo	$7*1$	$5*3$	$0*1$	$1*3$	$0*1$	$3*3$	$1*1$	$3*3$	$1*1$	$1*3$	$3*1$	$0*3$
Resultado	7	15	0	3	0	9	1	9	1	3	3	0

Tabla 5.5: Tabla de cálculo dígito verificación

Dígito	Posición	Valor	Codificación	Valor
1.	Guarda izquierda	(const.)		101
2.	1. Dígito izq.	[5]	Izq. paridad impar	0110001
3.	2. Dígito izq.	[0]	Izq. paridad par	0100111
4.	3. Dígito izq.	[1]	Izq. paridad impar	0011001
5.	4. Dígito izq.	[0]	Izq. paridad par	0100111
6.	5. Dígito izq.	[3]	Izq. paridad impar	0111101
7.	6. Dígito izq.	[1]	Izq. paridad par	0110011
8.	Barras centrales	(const.)		01010
9.	1. Dígito der.	[3]	Derecha	1000010
10.	2. Dígito der.	[1]	Derecha	1100110
11.	3. Dígito der.	[1]	Derecha	1100110
12.	4. Dígito der.	[3]	Derecha	1000010
13.	5. Dígito der.	[0]	Derecha	1110010
14.	Dígito de verificación	[9]	Derecha	1110100
15.	Guarda derecha	(const.)		101

Tabla 5.6: Construcción del código



Figura 5.8: Código de barras EAN-13 creado

## 5.6. Extracción de los trazos

A partir de la imagen del ejercicio y del modelo se extraen los trazos que ha realizado el alumno. Para ello se le aplica al ejercicio y al modelo un filtro de umbral que convierte las dos imágenes a blanco y negro y se superponen, para extraer los píxeles negros que se encuentran en la imagen escaneada y no en el modelo. En la figura 5.9 vemos como a partir de una imagen sin resolver y una imagen resuelta se genera la imagen de los trazos. El paso intermedio muestra como se superponen las dos imágenes.

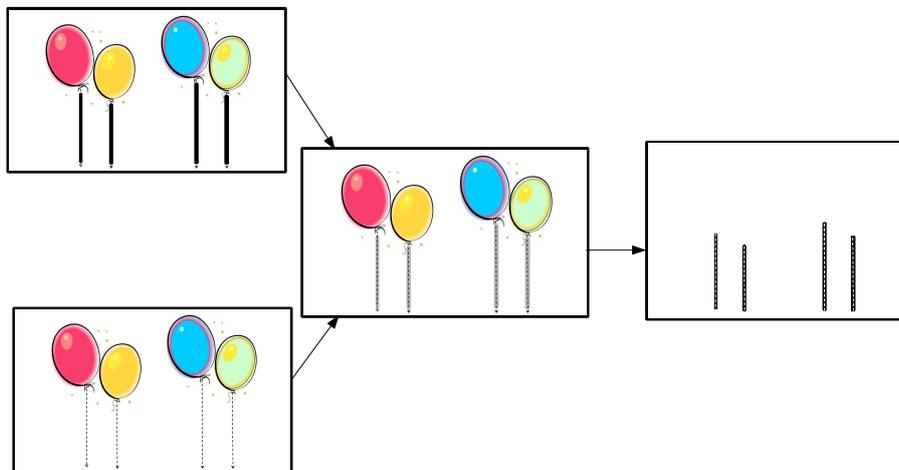


Figura 5.9: Extracción de los trazos

Para extraer los trazos de la imagen del ejercicio primera se crea una imagen en blanco del mismo tamaño que el ejercicio. Después se recorren las dos imágenes, que previamente han sido reescaladas, punto por punto. Si el  $pixel_{ij}$  del ejercicio es un píxel negro y el  $pixel_{ij}$  del modelo es blanco ese punto se considerará parte de un trazo dibujado por el alumno y en la imagen de salida se pintará dicho píxel en negro.

### 5.6.1. Reconstrucción de los trazos

Esto se realiza de la siguiente forma, en el ejercicio resuelto se comprueba si los píxeles que hay alrededor de la guía son de color negro. En el caso de que así sea quiere decir que parte del trazo del alumno se ha dibujado sobre la guía, por lo que debemos reconstruir el trazo. La reconstrucción del trazo funciona como se muestra en la figura 5.10. El objetivo es crear un polígono que represente el área comprendida en el trozo de trazo que falta y rellenarlo del color del trazo.

Para obtener la posición de las guías aplicamos el algoritmo de marcado explicado en la sección 5.4.1 al modelo del ejercicio. Una vez obtenidas las posiciones de las guías y los

rectángulos que las envuelven, se comprueba si alrededor de esos rectángulos en la imagen de ejercicio hay píxeles negros, en cuyo caso habrá que reconstruir el trazo.

Para ello se recorren los píxeles alrededor de la guía y en el caso de que sea un píxel límite, es decir un píxel negro que tenga un vecino blanco, se añade al polígono. Si el píxel está en una esquina y es negro también se añade. Cuando se ha hecho el recorrido completo se rellena el polígono resultante de negro.

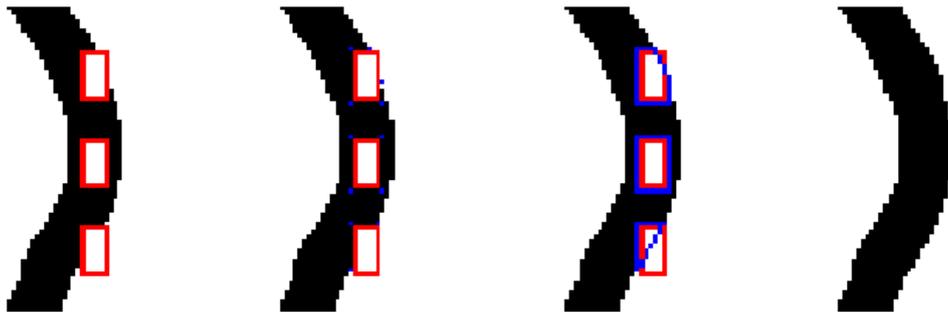


Figura 5.10: Detalle reconstrucción de los trazos

Este proceso debemos repetirlo para todas las guías. En la figura 5.11 vemos el proceso aplicado a toda la imagen de un ejercicio. La imagen que se obtiene contiene todos los trazos reconstruidos y ya se puede continuar con el proceso de eskeletonización.

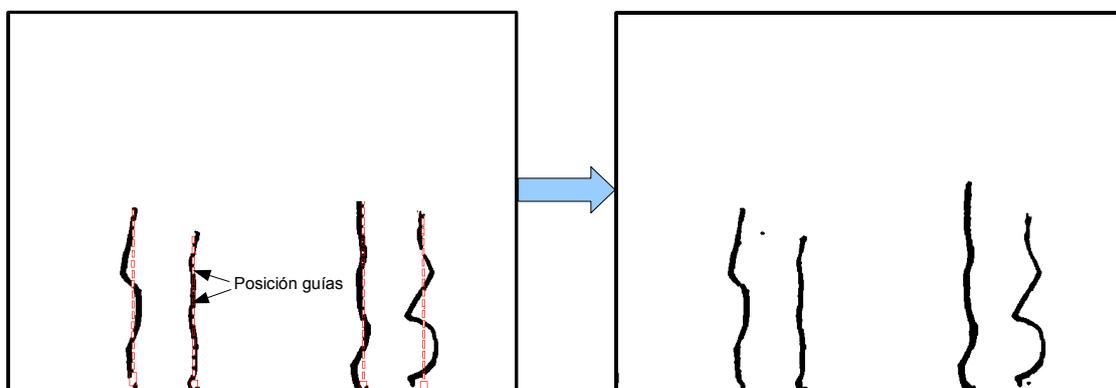


Figura 5.11: Reconstrucción de los trazos aplicado a un ejercicio

### 5.6.2. Eskeletonización de los trazos

Este proceso está inspirado en el que se propone en el artículo “Piecewise linear skeletonization using principal curves” [25]. El algoritmo usado para generar el esqueleto es el propuesto por Zhang-Suen [26]:

Este método es un método paralelo, es decir, cada píxel se puede calcular utilizando el valor de la iteración anterior. El método consta de dos subiteraciones. En cada una de ellas se eliminarán aquellos píxeles que cumplan todas las reglas definidas para la subiteración. La primera subiteración consiste en un conjunto de reglas que se comprueban para cada píxel de la imagen, y son las siguientes:

1. La conectividad debe ser uno.
2. Tiene al menos dos vecinos negros y no más de seis.
3. Al menos uno de  $I(i,j+1)$ ,  $I(i-1,j)$  e  $I(i,j-1)$  es blanco.
4. Al menos uno de  $I(i-1,j)$ ,  $I(i+1,j)$  e  $I(i,j-1)$  es blanco.

La segunda subiteración es igual que la primera salvo por las dos últimas reglas, que son las siguientes:

1. Al menos uno de  $I(i-1,j)$ ,  $I(i,j+1)$  e  $I(i+1,j)$  es blanco.
2. Al menos uno de  $I(i,j+1)$ ,  $I(i+1,j)$  e  $I(i,j-1)$  es blanco.

Si alguna de estas dos subiteraciones se completan sin haber eliminado ningún píxel, el esqueleto estará completo y el programa se parará. En la figura 5.12 se muestra la aplicación del algoritmo a una imagen.

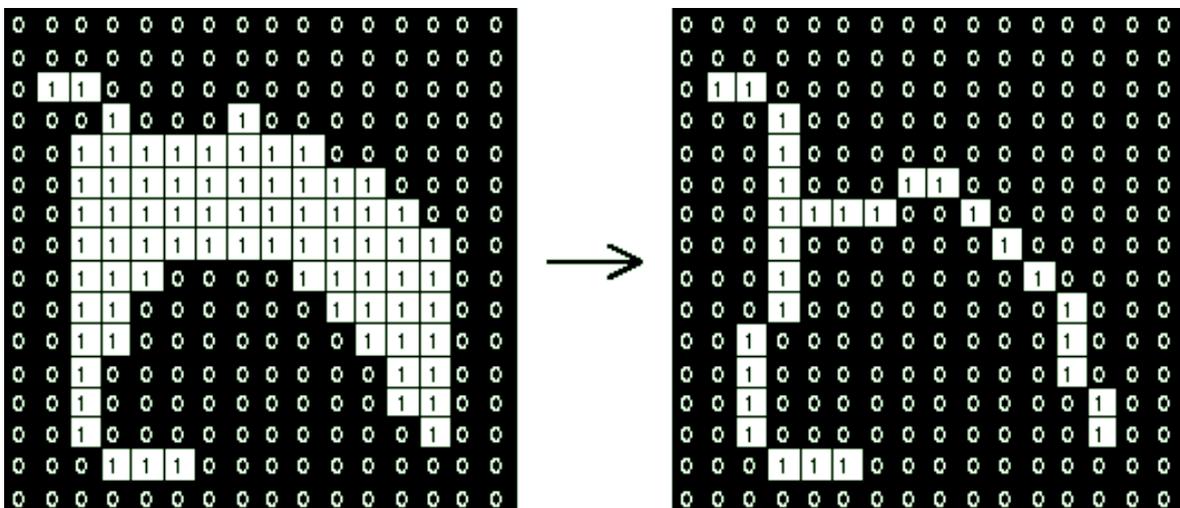


Figura 5.12: Esqueleto de una imagen

Con ayuda del esqueleto generado que discurre por el centro de los trazos de los alumnos podemos convertirlos a una representación matemática en forma de polilíneas.

### 5.6.3. Filtrado del esqueleto

Para filtrar el esqueleto se recorre toda la imagen en busca del comienzo de un trazo. El comienzo de trazo es un píxel negro que tiene un sólo vecino. La vecindad que se considera es vecindad-8. A partir de ahí se recorrerá el trazo hasta llegar a una intersección, que es un píxel con más de un vecino negro sin contar el píxel por el se llega. Si la rama recorrida no supera una cierta longitud se pintarán todos sus puntos de blanco. En la imagen 5.14 podemos ver las ramas filtradas del esqueleto creado a partir de un ejercicio.

1	2	3
4	*	5
6	7	8

Figura 5.13: Vecindad 8

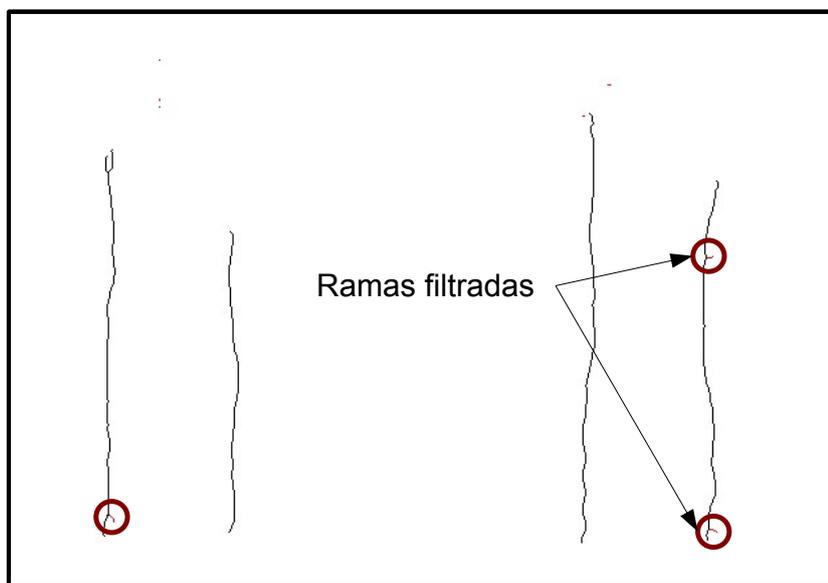


Figura 5.14: Filtrado del esqueleto de una imagen

#### 5.6.4. Separación y vectorización de los trazos

Tras haber aplicado el proceso de esqueletonización a la imagen tendremos una imagen binaria en la que los trazos pueden estar cruzados. Para convertirlos en polilíneas se recorre toda la imagen buscando el comienzo de un trazo y una vez encontrado se recorren los píxeles que lo forman añadiendo las posiciones de los puntos a la polilínea hasta que se llega a una intersección o al final, en cuyo caso se habría acabado con ese trazo y se buscaría otro. Si se llega a una intersección se continuará por el camino que presente una mayor continuidad para el trazo. Los píxeles que se hayan recorrido se marcarán como “visitados” y no se tomarán en cuenta en las sucesivas iteraciones del algoritmo, exceptuando los puntos de cruce, que si pueden pertenecer a varios trazos.

Para decidir por qué píxel continuar en un cruce de trazos se comprueba cual de los vecinos está mejor situado. Para ello se calcula la diferencia en el eje x e y se coge el menor en mínimos cuadrados. Si hay dos vecinos que tienen el mismo valor, se comprueba si uno de ellos es la continuación de otro trazo y se descartaría en ese caso.

En la figura 5.15 podemos ver un ejemplo de un cruce de dos trazos. Si se viene recorriendo el trazo por la rama 1, el camino más lógico sería seguir por la 3. En el caso de se comience por la rama 2, al llegar al punto marcado como cruce podríamos continuar por la 3 o la 4. Sin embargo en este punto se comprueba si hay otra rama que tenga mayor continuidad por alguna de estas ramas, y efectivamente se ve que la 1 encaja mejor con la 3 que con la 4. Por lo tanto, el recorrido que comience por la rama 2 acabará en la 4.

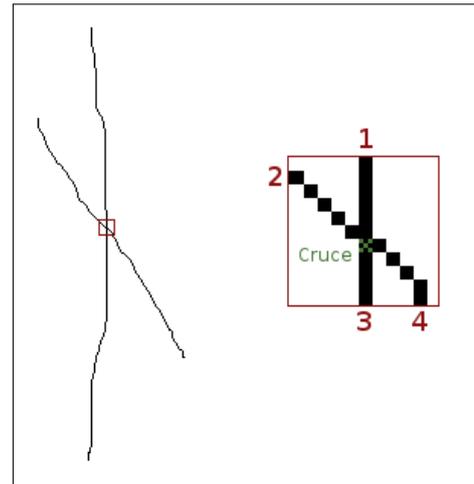


Figura 5.15: Cruce de dos trazos

### 5.6.5. Vectorización

Es necesario para poder aplicar el algoritmo de corrección que los segmentos que forman los trazos generados tengan la misma longitud. Para esto se utiliza un algoritmo que genera a partir de la imagen una polilínea como resultado que cumple esa propiedad. Para ello se interseca un círculo de la longitud que se quiere obtener con la polilínea y se toma el punto de la intersección para añadirlo al nuevo trazo. Se avanza hasta ese nuevo punto y se repite otra vez el mismo paso hasta llegar al final. En la figura 5.16 podemos ver como funciona este proceso.

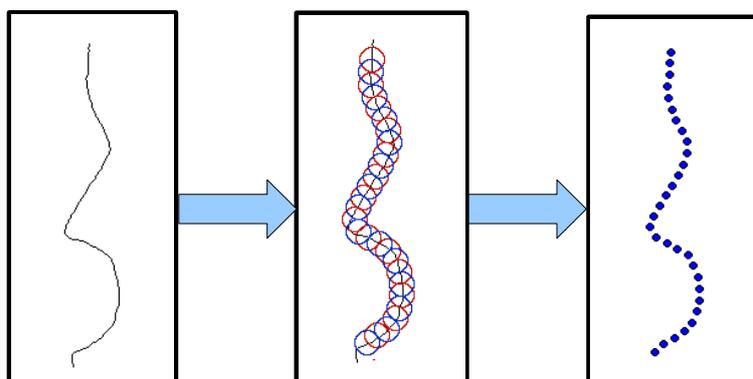


Figura 5.16: Proceso de vectorización

## 5.7. Corrección de los ejercicios

Una vez realizada la extracción de los trazos se procede a la corrección de los mismos. Todos los trazos que se han extraído de la imagen escaneada se contrastan con el modelo. Para hacer la corrección hay que asignar cada trazo del ejercicio a un trazo del modelo, esta asignación se realiza en función de la distancia. Por lo tanto se asignará al trazo del modelo al que esté más próximo. Para ello se recorren todos los puntos del trazo del ejercicio y se comprueba que trazo del modelo está más próximo a ese punto. Si no hay ningún punto dentro un radio máximo no se asignará dicho punto a ninguno del modelo. Al final se contabilizan a que trazo del modelo pertenecen la mayor parte de los puntos asignados, si la mayor parte pertenece a ningún trazo del modelo se considerará un trazo inválido y no se corregirá.

Los trazos del modelo se cargan a partir del fichero en el que están contenidos. Estos puntos se insertan en una estructura de Quadtree para obtener el trazo del modelo más cercano a un punto del trazo del ejercicio eficientemente.

Un Quadtree es una estructura de datos jerárquicas basada en el principio de descomposición recursiva del espacio. En un QuadTree de puntos, el centro de una subdivisión está siempre en un punto. Al insertar un nuevo elemento, el espacio queda dividido en cuatro cuadrantes, y al repetir el proceso, el espacio igual se divide en cuatro cuadrantes, lo que limitado por la división superior y así sucesivamente [27]. En la figura 5.17 se muestra un conjunto de puntos sobre un cuadrado y como se organizan gracias al QuadTree.

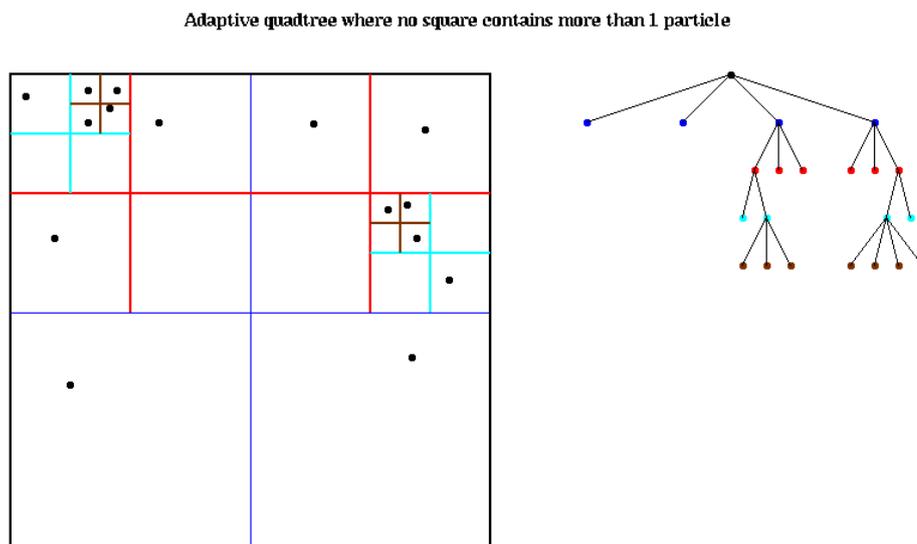


Figura 5.17: Quadtree

### 5.7.1. Asignación de subtrazos

Hay que tener en cuenta que los trazos realizados por los alumnos pueden ser más largos en ciertas zonas que el modelo si no se ajusta exactamente a su forma. Por lo tanto hay que hacer una partición del trazo y asignar cada parte a una parte del modelo para poder compararlos. Este proceso intenta asignar a cada parte del trazo del alumno la parte equivalente del trazo del modelo. Para ello se intenta “averiguar” que parte del modelo estaba intentado hacer el alumno al hacer un trazo. En primer lugar se asigna a cada punto del trazo el punto más cercano del modelo. Se recorre esa asignación y se comprueba si entre algún par de puntos hay una discontinuidad. En ese caso se entenderá que el alumno se ha saltado una parte del modelo. Para la comparación de forma se asignará el trozo que se ha saltado al trozo de trazo de ejercicio. De forma análoga si el alumno realiza una parte demasiado grande con respecto al modelo, ésta se asignará a un trozo del modelo. En la figura 5.18 se muestra cada asignación delimitada por una línea discontinua.

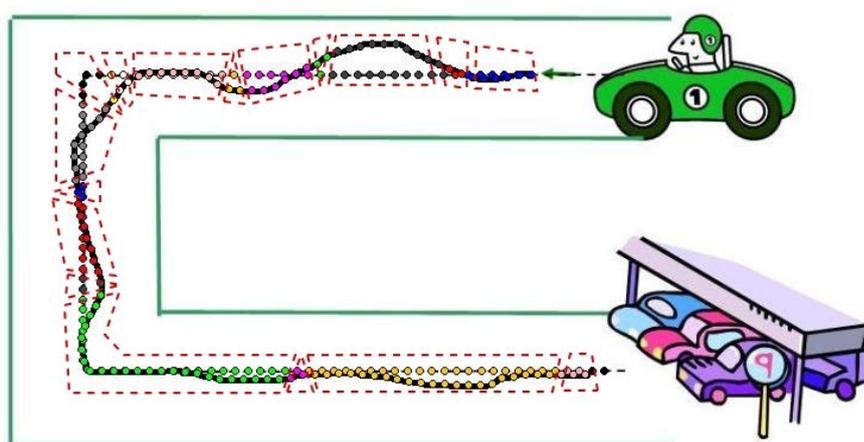


Figura 5.18: Partición de los trazos

### 5.7.2. Cálculo de características

- Cálculo de los grados

Para cada segmento de cada subtrazo se calculará el ángulo que forma con la horizontal. Al final del proceso anterior tendremos una lista de pares de subtrazos que pueden ser de distinto tamaño. Cada par de trazos se hace del mismo tamaño multiplicando la longitud de cada segmento del par más pequeño por un factor.

- Cálculo de la diferencia de forma

Para cada par de trazos se crea una función que representa la forma de cada uno de ellos. Estas funciones representan el ángulo que forma con la horizontal cada segmento.

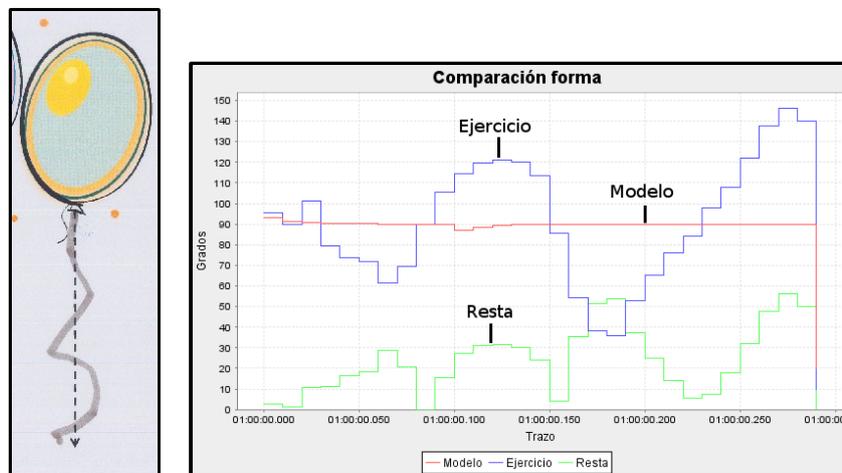


Figura 5.19: Comparación forma modelo y ejercicio

La diferencia de forma se calcula restando una función de la otra y tomando el valor absoluto. Este método está inspirado en el propuesto en el artículo “Partial Matching of Planar Polylines Under Similarity Transformations” [28]. Después esta nueva función se integra y se divide entre la longitud del trazo. En la figura 5.19 vemos los valores de los ángulos que forman el trazo de un ejercicio y el del modelo. En la parte más baja de la gráfica está la función que expresa la diferencia de forma entre el modelo y el ejercicio, esta función es la que se integra y divide entre la suma de las longitudes para obtener la desviación media del trazo.

- Cálculo de la distancia

El cálculo de la distancia del trazo del ejercicio se realiza asignando a cada punto del trazo del ejercicio el punto del modelo más cercano y se computa la media aritmética.

- Cálculo de la diferencia de longitud

La diferencia de las longitudes de los trazos se calculan sumando las longitudes de los segmentos del trazo del ejercicio y restando la longitud del modelo.

## 5.8. El almacenamiento de ejercicios

Los resultados obtenidos en la corrección se almacenan en la base de datos y la imagen con el ejercicio se guarda en un fichero. Además se crea un log que almacena los trazos en un fichero de texto. El log se construye de la siguiente forma, cada línea constituye una acción del usuario y está compuesta por un número que especifica la acción realizada, dos

números con la coordenada de la pantalla en la que se realizó y un número que especifica el tiempo relativo en que se realizó la acción en milisegundos. De esta forma también se puede reproducir el ejercicio más tarde. Cada línea del log consta de cuatro campos: el tipo de evento, las dos componentes de la coordenada y una marca de tiempo. En la figura 5.20 se muestra un ejemplo.

ID	X	Y
1	62.86	152.49
2	65.71	163.06
2	73.29	169.94
2	81.03	176.58
2	90.71	176.85
2	100.29	175.22
2	109.18	170.83
2	117.49	165.15
2	126.97	162.92
2	136.59	161.64
2	146.20	160.28
2	154.87	165.21
2	160.08	174.53
2	162.58	185.21
2	165.35	195.80
2	167.52	206.57
2	169.42	217.41
2	176.95	224.35
2	185.84	228.76
2	195.48	227.72
2	205.11	226.62
2	214.80	226.60
3	224.48	226.60

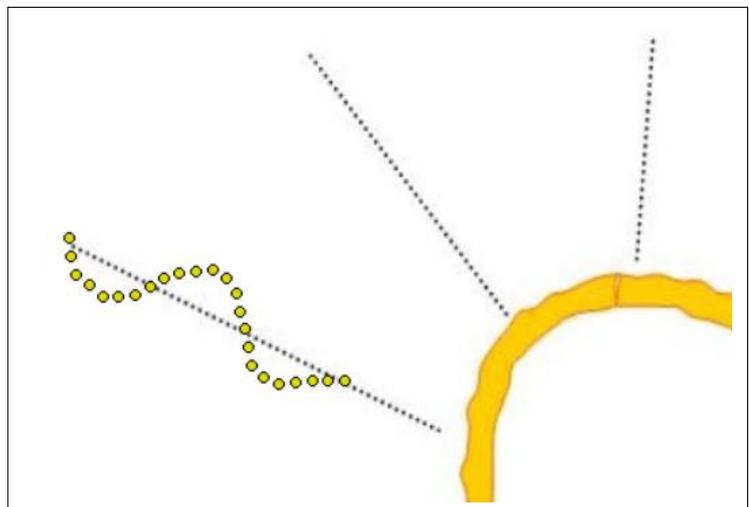


Figura 5.20: Log y trazo resultante

El significado de los tipos de eventos es el siguiente:

ID	Significado
1	Comienzo de trazo
2	Punto intermedio de trazo
3	Fin de trazo
4	Tamaño de la pantalla

## 5.9. Arquitectura general

El sistema está dividido en una serie de módulos diferenciados. Estos son la base de datos, la interfaz, el paquete de tratamiento de imágenes y el paquete de corrección. El paquete de base de datos contiene todas las clases y métodos necesarios para acceder a la información almacenada en la base de datos y oculta las operaciones de bajo nivel a las capas superiores. El módulo de tratamiento de imágenes contiene los métodos necesarios para la detección de marcas, lectura del código de barras, operaciones para filtrar las imágenes (filtro de umbral, eliminar ruido, rotación etc.) y extraer los trazos del modelo. El módulo de trazos se encarga de vectorizar los trazos y contrastarlos con el modelo para evaluarlos en función de una serie de características. El paquete de interfaz contiene todas las ventanas, diálogos, paneles y demás componentes que sirven para que el usuario interactúe con el programa.

En la figura 5.21 se pueden ver los paquetes de los que está compuesto el sistema y las dependencias entre ellos. A continuación se describirán detalladamente todos los módulos que componen el sistema:

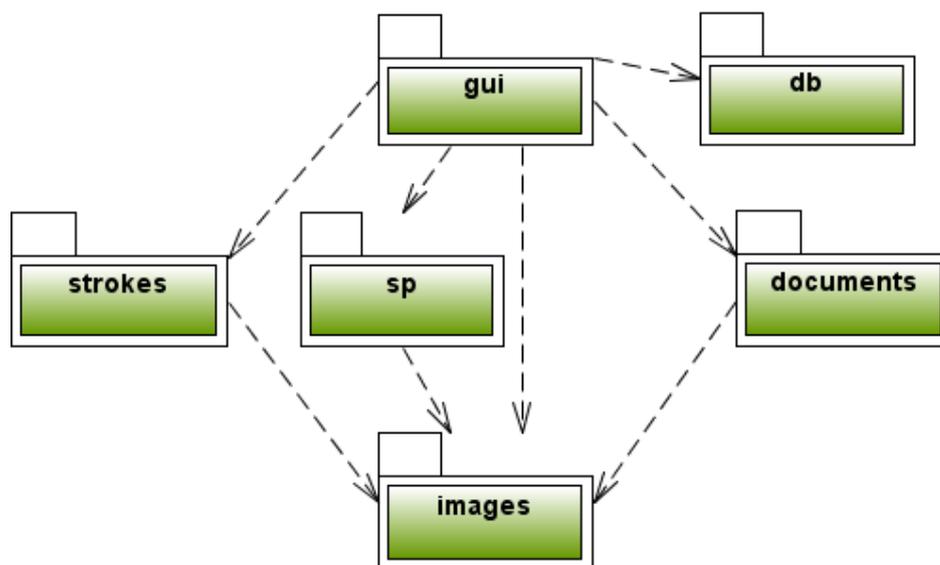


Figura 5.21: Esquema paquetes

## 5.10. Acceso a la base de datos

El acceso a la base de datos se hará a través de una única clase. Para que sólo exista una instancia de dicha clase se usa el patrón singleton. La clase `DatabaseService` actúa como una capa intermedia entre la base de datos y el sistema y oculta la generación de las consultas SQL y se comunica con la API para acceder a bases de datos de JAVA. A modo de ejemplo se puede ver en la figura 5.22 un acceso a la base de datos.

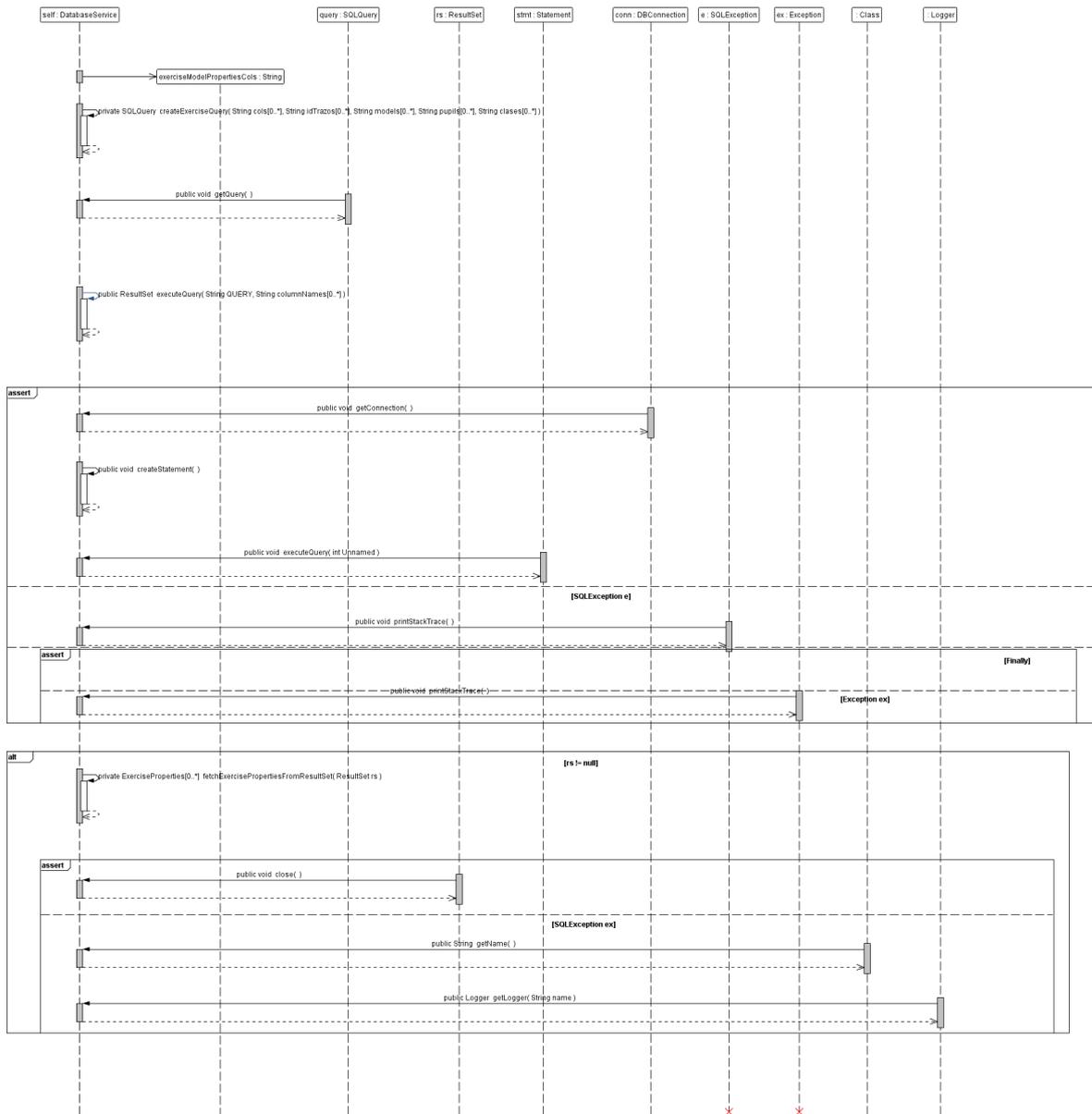


Figura 5.22: Diagrama de secuencia de la base de datos

## 5.11. La Interfaz

La interfaz se ha implementado usando la biblioteca gráfica SWING de Java. En esta sección se explicará el diseño de las partes más importantes de la interfaz.

### 5.11.1. Internacionalización

Para facilitar la internacionalización todos los mensajes de la interfaz se almacenan en un fichero de texto. Si se quiere añadir otro idioma sólo habrá que crear un nuevo fichero con las traducciones de los mensajes.

### 5.11.2. Listas

Para seleccionar modelos y ejercicios se ha implementado una lista que carga ficheros de imágenes y las muestra. En la figura 5.23 se ven las relaciones entre los distintos tipos de listas y la jerarquía. La lista de imágenes tiene métodos comunes a todas las listas para cargar una colección de imágenes en segundo plano, para abrir un visor de imágenes, para cambiar el tamaño de las imágenes en la lista etc. Se ha implementado una clase que renderiza los elementos de las listas para que muestren las imágenes y un checkbox que se activa cuando se selecciona una imagen.

Las listas que muestran modelos y ejercicios son subclases de dicha lista y cada una muestra diferentes acciones para realizar a través del menú contextual. La lista de visión de modelos tiene métodos para abrir una interfaz de dibujo, imprimir ejercicios etc. La lista de ejercicios escaneados métodos para seleccionar el modelo, decodificar el código de barras y corregir los ejercicios. Cada lista de imágenes tiene un comparador distinto para ordenar los elementos en función de sus atributos. Para implementar esto se ha usado el patrón factory [29], de esta forma cada lista reimplementa el método para obtener el comparador.

La clase ImageComponent contiene cada imagen que se muestra en las listas y las clases ModelComponent y ExerciseComponent heredan directamente de ésta. Mientras que la clase ImageComponent contiene métodos para cargar imágenes, las subclases contienen las implementaciones de los métodos que se pueden realizar sobre los modelos y ejercicios. En la figura 5.24 se ven los métodos y las relaciones entre los componentes.

El estado de los ejercicios se ha modelado siguiendo el patrón estado. En la figura 5.25 vemos el diagrama de clases para la implementación del patrón estado. En las clases que representan los estados se define el siguiente estado para cada una de las operaciones disponibles. De esta forma las clases que implementan cada estado son responsables de las transiciones.



Figura 5.23: JLists

### 5.11.3. Reproducción de ejercicios

En la figura 5.26 se muestra el diagrama de secuencia de esta operación. Las clases que se usan en esta operación son las siguientes:

- FullscreenPaintPanel

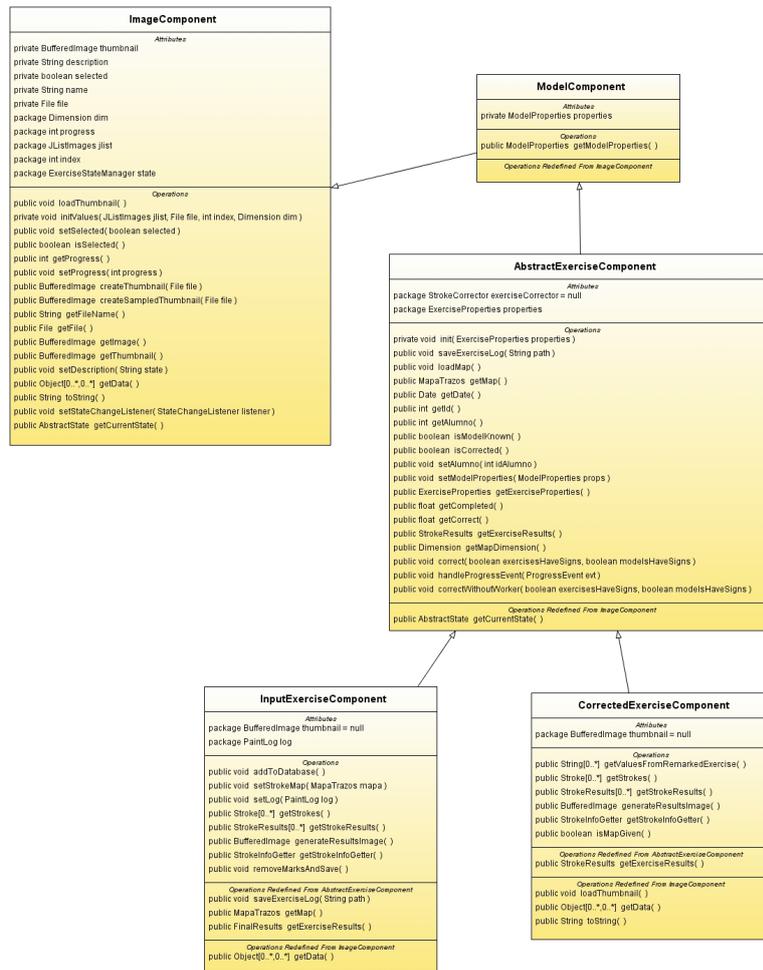


Figura 5.24: Diagrama de los componentes de las listas

Panel de la interfaz que muestra la interfaz de dibujo

- Painter

Clase que se encarga de renderizar los trazos sobre la interfaz de dibujo

- Player

Clase para leer los ficheros de log y reproducir los ejercicios almacenados. También los graba y almacena.

Para reproducir un ejercicio en la pantalla de dibujo hay que cargar la imagen del modelo correspondiente como fondo. Después se obtiene el fichero que contiene el log del ejercicio y se llama al método loadLog de la clase Player para que lea y cargue el fichero. Si se estuviera reproduciendo otro ejercicio hay que detenerla. Después se puede iniciar la reproducción llamando al método replay de esa clase.

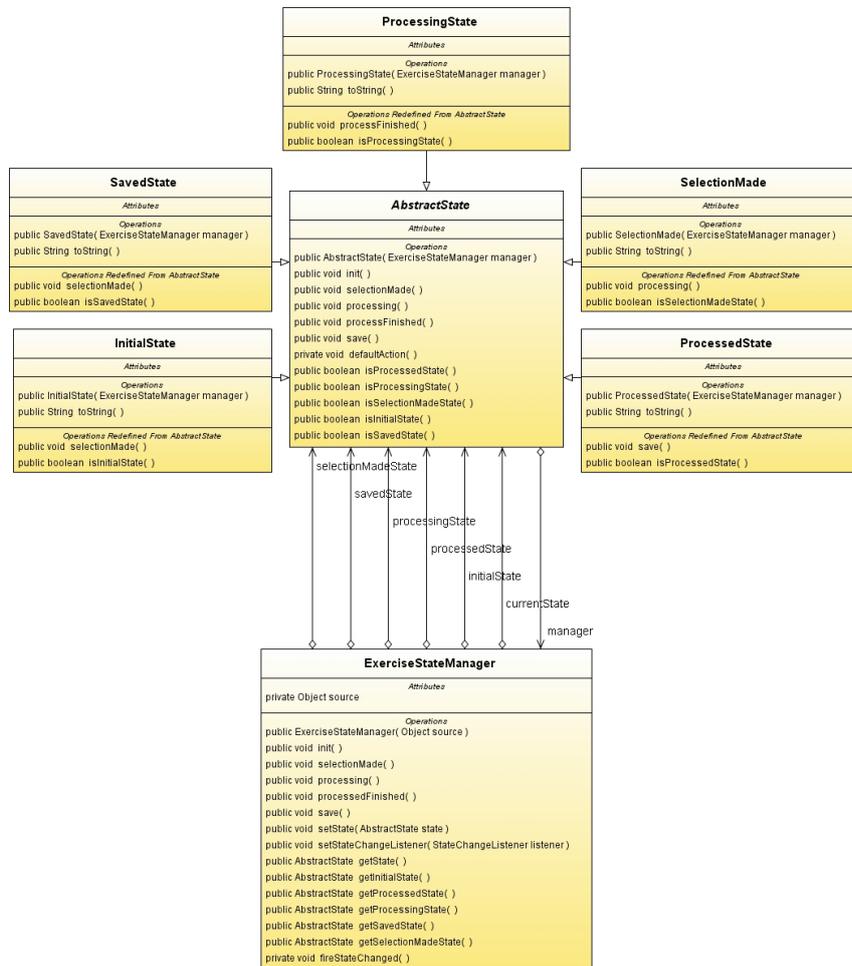


Figura 5.25: Diagrama de clases representando los estados del ejercicio

#### 5.11.4. Extracción y vectorización de trazos

Las clases usadas para este proceso son como podemos ver en la figura 5.27 las siguientes:

- **StrokeExtractor**

Es la clase encargada de realizar todo el proceso.

- **ImageUtil**

Clase que contiene métodos básicos para el tratamiento digital de imágenes, como lectura de ficheros que contienen imágenes y reescalado.

- **Analyzer**

Esta clase detecta la posición de las marcas y extrae las subimágenes que contienen el ejercicio.

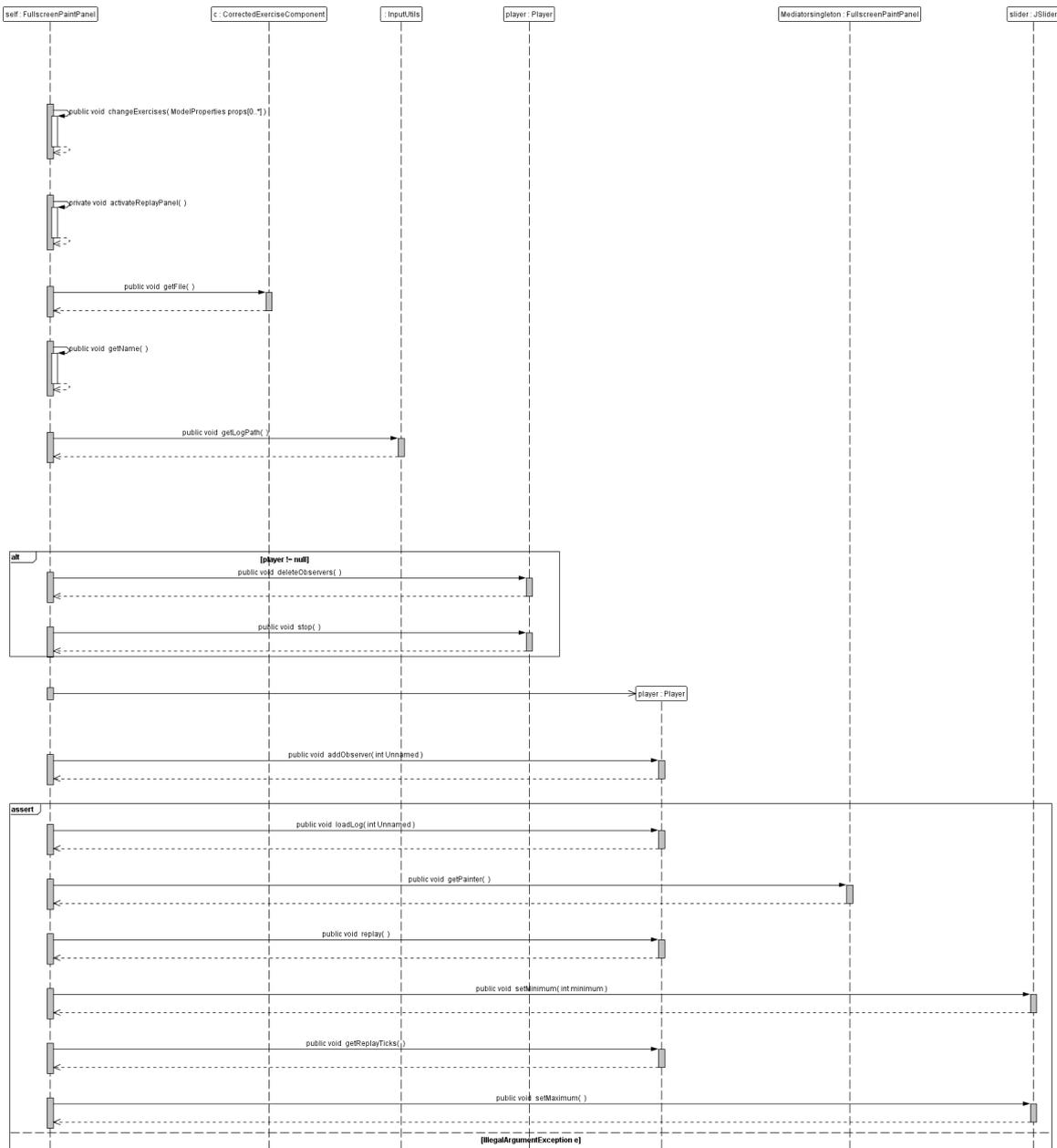


Figura 5.26: Diagrama de reproducción

- Filtros

Las clases que implementan los filtros para tratar las imágenes.

- LabelImage

Esta clase detecta y etiqueta símbolos dentro de una imagen.

En primer lugar se leen las imágenes del modelo y el ejercicio escaneado. Después se detectan las marcas del ejercicio escaneado con la clase *Analyzer*. Para extraer los trazos del ejercicio a partir de la imagen escaneada y el modelo punteado se usa la clase *SubstractFilter*. Después se aplican una serie de filtros para reducir el ruido que pueda contener la imagen y poder aplicar el algoritmo de vectorización. Finalmente se aplica el algoritmo de vectorización que transforma las líneas de un pixel que contiene la imagen en polilíneas.

### **5.11.5. Corrección de un ejercicio**

En la figura 5.28 vemos el diagrama de secuencia para la corrección de un ejercicio y las clases implicadas en el proceso. Primero se carga el modelo almacenado en un fichero de log con la clase *StrokeMap*. Esta clase inserta los puntos de los trazos en una estructura de *Quadtree*. Después con la clase *StrokeExtractor* se extraen los trazos del ejercicio escaneado. Es necesario hacer los dos mapas de la misma dimensión para poder hacer la corrección. Después se recorren todos los trazos que se han extraído del ejercicio y se corrigen comparándolos con los del modelo, utilizando el método *markStroke* de la clase *StrokeMap*. Finalmente se invoca el método *computeFinalResults* para obtener el resultado del ejercicio.



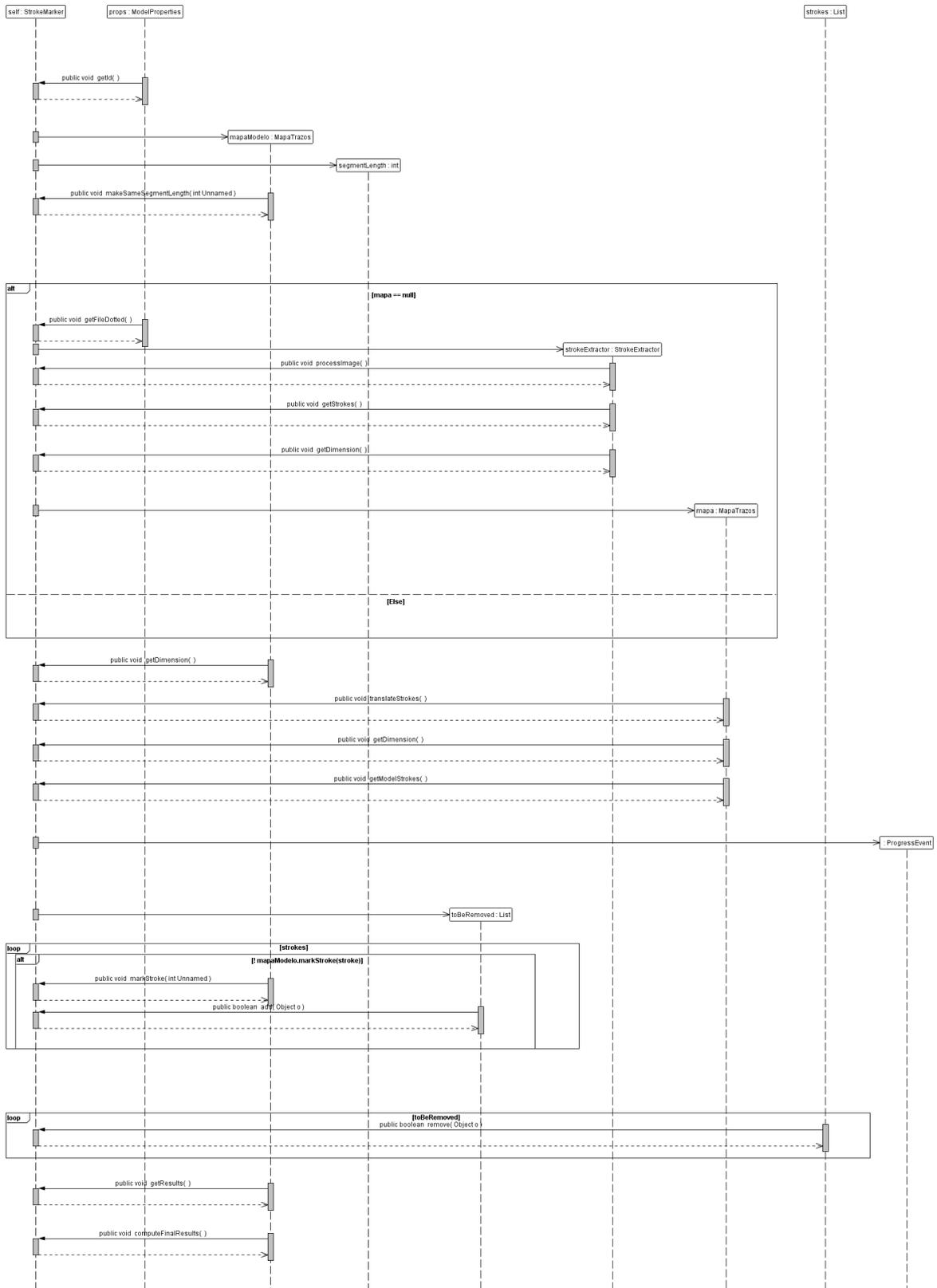


Figura 5.28: Secuencia de corrección



# Capítulo 6

## Evolución del sistema

Para el desarrollo de este PFC se ha seguido un proceso basado en generar de forma incremental sucesivos prototipos que han sido sometidos a evaluación formativa por parte de los usuarios. La evaluación formativa [30] ocurre durante el diseño y en las primeras fases de desarrollo de un proyecto. Está orientada a las necesidades inmediatas de los que trabajan para mejorar el diseño y, por consiguiente, el funcionamiento del sistema. Sirve para obtener información detallada de lo que se puede modificar y mejorar en la funcionalidad del sistema. Su función es encontrar fallos en el sistema lo suficientemente pronto como para que los cambios que haya que hacer no afecten al proceso. En este tipo de evaluación se consideran tanto características generales como detalles específicos (como diseño de la interfaz, rendimiento, etc.). Este método de trabajo nos lleva a un proceso de diseño en el que los usuarios intervienen desde las primeras fases [31].

En las siguientes secciones se explicará cada ciclo detalladamente y se hará un resumen de los resultados obtenidos durante la evaluación, los problemas detectados y las medidas que se han aplicado para solucionarlos.

### 6.1. Ciclos de desarrollo

Se han realizado cuatro prototipos que se han ido desarrollando de forma iterativa e incremental. En cada versión que se ha realizado del sistema se han añadido nuevas funcionalidades o bien se han corregido defectos del prototipo anterior. Para ello se ha sometido cada uno a una evaluación que ha servido para detectar problemas y mejorar el programa.

### 6.2. Usuarios

Los prototipos se han evaluado con alumnos de tres y cuatro años del colegio de educación infantil “Los Claveles” de Mijas, Málaga. Las evaluaciones han servido para ver las necesidades de los niños y corregir los problemas que han ido surgiendo.

## **6.3. Ciclo 0. Filtrado, detección de marcas**

Durante el ciclo inicial del proyecto se realizó el modelado de ejercicios y de usuarios del sistema así como el flujo general del sistema. Se decidió utilizar ejercicios de trazos para el sistema, destinados a niños de entre 3 y 4 años. Para poder recolocar los ejercicios se definieron los mecanismos para añadir marcas en las esquinas del ejercicio, diferenciando el lado izquierdo del derecho. Se implementaron los métodos necesarios para realizar todas estas operaciones.

### **6.3.1. Características implementadas**

- Se modela el flujo del sistema.
- Se detecta la necesidad de añadir marcas a los ejercicios para colocarlos correctamente para poder realizar la corrección.
- Se implementan filtros para convertir las imágenes a blanco y negro y los algoritmos para detectar las marcas correctamente.

### **6.3.2. Objetivos**

En esta evaluación se pretendía estudiar el funcionamiento de la detección de marcas y de filtros de imágenes, así como familiarizarse con otras herramientas para el tratamiento digital de imágenes.

## **6.4. Ciclo 1. Entrada de datos en papel y módulo de corrección**

Durante este ciclo se definió el proceso de corrección y se realizó una implementación básica de un método de corrección (Figura 6.1). Para poder contrastar el ejercicio se requería incorporar el modelo al sistema. También durante este segundo ciclo del proyecto se desarrollaron los paquetes más básicos y sirvió también para definir nuestros objetivos, ver los fallos y realizar los cambios pertinentes para mejorar el proyecto. En este ciclo se definieron las características que debían tener los ejercicios para poder ser analizados, así como un método de corrección para los ejercicios. También filtros para eliminar ruido generado durante el escaneo de las imágenes y facilitar la detección de marcas.

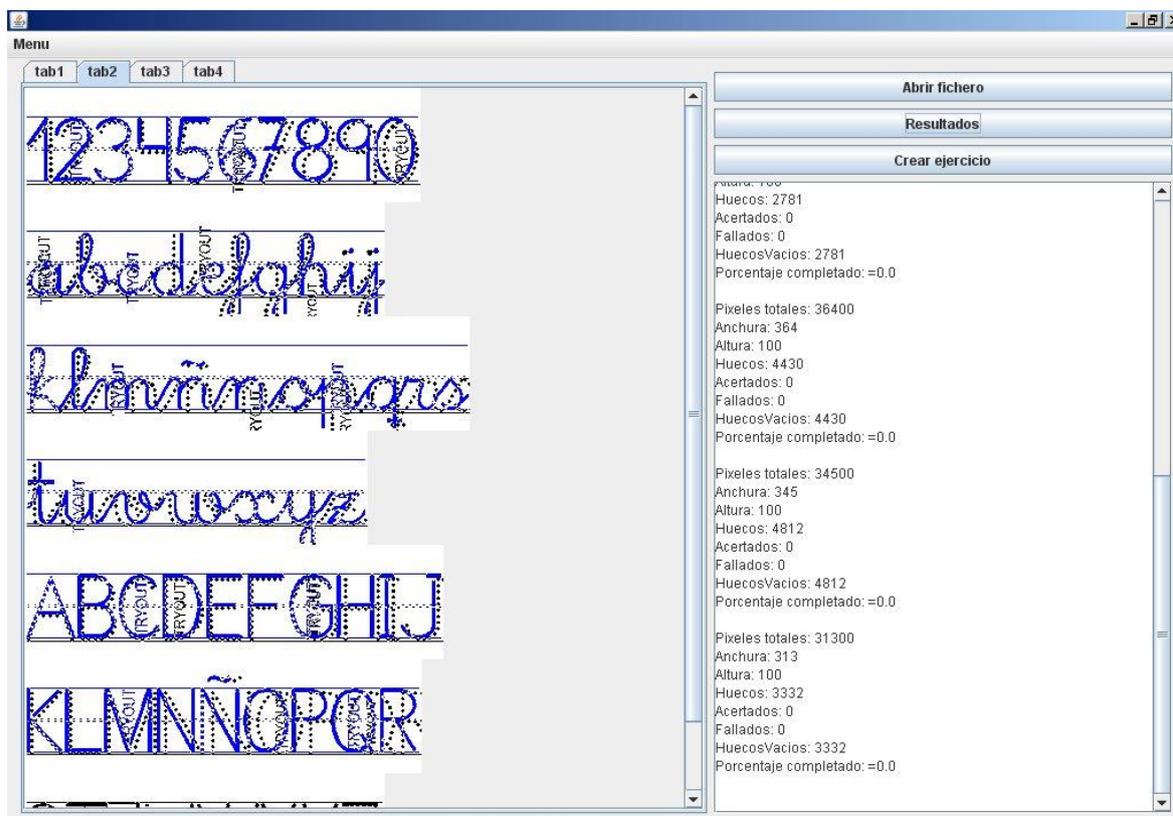


Figura 6.1: Versión básica del sistema

### 6.4.1. Características implementadas

- El algoritmo de corrección implementado en este ciclo contabilizaba cuantitativamente el grado de semejanza que hay entre el modelo y el ejercicio realizado por el alumno
- Se implementan filtros para la eliminación de ruido
- Se definen los mecanismos para añadir un modelo al sistema

### 6.4.2. Objetivos

En esta evaluación se pretendía estudiar el funcionamiento del sistema con ejercicios reales y detectar problemas que pudieran surgir. También se quería probar el método de corrección definido en este ciclo.

### 6.4.3. Evaluación

Para esta fase se prepararon una serie de ejercicios para que los realizaran los alumnos y tener ejemplos con los que poder trabajar. Para ello se pidió que los alumnos hicieran

diferentes ejercicios en papel con distintas herramientas y colores. A estos ejercicios se le añadieron marcas para delimitar la zona en la que se dibujaban los trazos para poder corregirlo después. Cada modelo tiene definido un patrón que luego servirá para comparar con los realizados por el alumno. Los alumnos realizaron los ejercicios con diferentes colores y herramientas (boli, lápiz, rotulador etc.) para comprobar con cuales de ellas se podía trabajar y cuales presentaban problemas a la hora de ser tratadas. Los ejercicios realizados se escanearon y se probaron las técnicas de corrección implementadas.

#### **6.4.4. Problemas detectados**

- Estos ejercicios sirvieron para realizar ajustes en la detección de marcas y en el algoritmo de corrección, así como en el proceso de filtrado de las imágenes.
- Se comprobó que el algoritmo de corrección implementado no se adecuaba al problema.
- Tras imprimir y escanear el ejercicio sufre ciertas deformaciones, lo que hace que no se ajuste exactamente con el modelo.
- Los ejercicios escaneados no conservan los colores de la imagen original, algunos píxeles negros aparecen con colores distintos y los colores blancos aparecen grisáceos.
- Cuando los niños dibujan con colores claros o lápiz el proceso de filtrado elimina parte de los trazos.
- La gran cantidad de ejercicios realizados hace que asignar un modelo y un alumno a cada uno de ellos sea una tarea tediosa.

#### **6.4.5. Ajustes**

Ante los resultados obtenidos se decide rediseñar el algoritmo de corrección, así como descartar herramientas de dibujo como el lápiz, ya que los ejercicios realizados con esta herramienta son difícilmente tratables.

### **6.5. Ciclo 2. Entrada de datos papel, pantalla táctil y monitor**

En este ciclo se modeló un nuevo proceso de corrección que se adecuara mejor al problema. También se pensó en la posibilidad de realizar ejercicios sobre otros dispositivos además de los ejercicios realizados sobre papel. Como nuevo dispositivo se escogió un tablet-pc. Para ello se implementó una interfaz que permitiera realizar dibujos y mostrar los ejercicios en la pantalla. Además se definió un modelo para almacenar el proceso de dibujado con este nuevo

sistema de realizar ejercicios, de forma que se pudieran reproducir más tarde. Así además del resultado se podría inspeccionar el proceso por el que el alumno llegó a ese resultado. Otra ventaja de almacenar el ejercicio es que se podrían evaluar de nuevo en el caso de que se realizaran cambios en el algoritmo de corrección. También se sustituyó el método de corrección por uno más complejo, lo que llevó a añadir nuevos algoritmos al ese proceso para poder examinar la forma de los trazos.

### **6.5.1. Características implementadas**

- Se implementa una interfaz sencilla para añadir modelos y guardar los ejercicios.
- Interfaz para dibujar a través de pantallas táctiles.
- Se crea un log para guardar cada ejercicio en un fichero.
- Se modela e implementa un nuevo algoritmo de corrección y se le añade la capacidad de comparar la forma del ejercicio con el modelo.
- El nuevo modelo de corrección requiere de nuevos filtros: filtro de esqueleto.
- Se añade la capacidad de leer códigos de barras.

### **6.5.2. Evaluación**

Para evaluar esta nueva interfaz se realizaron pruebas con los alumnos con un tablet-pc. Se prepararon varios ejercicios que cada alumno debía realizar. Además se incluyó una tableta gráfica como nuevo dispositivo. También se usaron los ejercicios del ciclo anterior para evaluar el nuevo algoritmo de corrección

### **6.5.3. Objetivos**

En este ciclo se pretendía evaluar como se adaptan los alumnos a escribir con los nuevos dispositivos, así como el nuevo modelo de corrección y de almacenamiento de ejercicios.

### **6.5.4. Mejoras**

- El nuevo proceso de corrección supone una mejora considerable respecto a la versión anterior.
- Los nuevos dispositivos de entrada permiten examinar el proceso de dibujado del alumno.
- Con los nuevos dispositivos el recorrido de los trazos es más exacto que infiriéndolo de los ejercicios escaneados.

### 6.5.5. Problemas detectados

- La tableta presentó diversos problemas, hay que fijar el papel perfectamente, lo que para cada una cantidad elevada de ejercicios resulta poco práctico.
- Las marcas de tiempo de los logs no son del todo precisas.
- El tablet-pc necesita que se aplique algo más de presión para escribir que sobre papel.
- El algoritmo de extracción de trazos de los ejercicios escaneados es ineficiente

## 6.6. Ciclo 3. Entrada de datos papel, pantalla, tableta y monitor ampliado

Los resultados del segundo ciclo sirvieron para realizar mejoras sobre el mecanismo de capturar las acciones sobre la pantalla táctil. También se hicieron ajustes sobre el sistema de corrección para mejorar la evaluación que daba. Además se añadieron dos nuevos tipos de dispositivos para la entrada de datos, una pantalla táctil y una pizarra interactiva.

### 6.6.1. Características implementadas

- Se añaden nuevas capacidades al algoritmo de corrección. Además de contrastar la forma del trazo del ejercicio con la del modelo, se comprueba la distancia, la diferencia de longitud
- Nuevos filtros para el algoritmo de corrección: filtro de fondo para preservar los trazos en caso de estén hecho con un color muy claro.
- Se añaden nuevas capacidades a la interfaz: imprimir ejercicios con códigos de barras
- Se mejora la implementación del almacenaje de ejercicios.

### 6.6.2. Evaluación

Se prepararon una serie de ejercicios nuevos que cada alumno realizó sobre la pantalla táctil. Algunos alumnos realizaron ejercicios sobre la pizarra interactiva. Se comprueban los resultados del algoritmo con los de los profesores

### 6.6.3. Objetivos

En esta evaluación se pretende observar como se adaptan los alumnos a nuevos dispositivos de escritura y recopilar más datos para seguir refinando el sistema.

#### 6.6.4. Problemas detectados

- La pantalla táctil no es adecuada para escribir sobre ella, pero sí para hacer los trazos con los dedos, convirtiéndose en una herramienta más para el desarrollo motor, añadida a pizarra, suelo...
- La extracción de los trazos de las imágenes es un proceso bastante complejo y a veces surgen errores.
- Las guías de los modelos deben separarse de los dibujos

#### 6.6.5. Mejoras

- El nuevo algoritmo de corrección produce mejores resultados
- Se mejora el rendimiento
- Se aumenta el acierto en la detección de marcas
- Se mejora la interfaz, añadiendo la capacidad de monitorizar los resultados de los alumnos a través de gráficas

### 6.7. Evaluación dispositivos de entrada

Para la realización de los ejercicios se han utilizado distintos dispositivos de entrada. Cada uno de ellos presenta inconvenientes y ventajas frente al resto. A continuación se describirán los dispositivos usados.

#### 6.7.1. Papel

El papel es el medio que usan los alumnos normalmente y por tanto les resulta mucho más sencillo realizar los ejercicios sobre él. Para escribir sobre papel pueden usar distintas herramientas de escritura como lápiz, rotulador y cera. Presenta muchas ventajas: se puede girar, mover etc. Sin embargo también tiene algunos inconvenientes respecto a los otros dispositivos. Es imposible determinar el sentido y el orden en el que el alumno ha realizado los trazos. Si el alumno ha cruzado trazos o los ha “repintado” no es posible saberlo. Además, la extracción de los trazos para el posterior tratamiento dentro del ordenador es un proceso complejo y relativamente lento, por lo que la corrección de estos ejercicios requiere más tiempo para el sistema.



Figura 6.2: Ejercicio en papel

### 6.7.2. Tableta

Este dispositivo de entrada consiste en un papel que se coloca sobre la tableta. Cuando el alumno escribe sobre el papel la tableta detecta los trazos y los guarda en una tarjeta de memoria SD. Después se pueden pasar al pc a través del puerto USB. Las ventajas que presenta este dispositivo es que se escribe sobre papel, que es a lo que están acostumbrados los niños. También se puede mover y girar fácilmente. Sin embargo presenta varios problemas: es necesario fijar el papel a la tableta de manera que no se desplace, ya que sino la imagen que se crea en la tableta aparecerá desplazada. Este hecho hace que el dispositivo no sea muy práctico y se haya descartado.



Figura 6.3: Ejercicio en tableta

### 6.7.3. Tablet-pc

Con este dispositivo los alumnos escriben directamente sobre la pantalla del portátil con un lápiz especial que tiene la punta de goma. Para que puedan escribir correctamente el portátil debe estar bien calibrado. Los alumnos deben adecuarse a este nuevo medio ya que se necesita aplicar una cierta presión a la pantalla para que se escriba bien. Las ventajas del tablet es que los trazos del alumno quedan perfectamente almacenados, por lo que no es necesario extraer los trazos de la imagen. Así se evitan los errores y se conoce el sentido y el recorrido de los trazos perfectamente. La corrección de estos ejercicios es mucho más rápida porque al contrario que con los ejercicios sobre papel no es necesario extraer trazos.



Figura 6.4: Ejercicio en tablet-pc

### 6.7.4. Pizarra interactiva

La pizarra interactiva tiene las mismas ventajas que el tablet-pc y además no es necesario aplicar tanta presión. Es un medio completamente distinto por el tamaño y las herramientas de escritura que se utilizan, y por lo tanto es interesante para que los alumnos practiquen. Tiene las mismas ventajas que el tablet-pc, pero también necesita ser calibrada correctamente. Este medio es el que menos se ha probado.

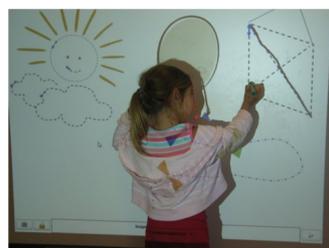


Figura 6.5: Ejercicio en pizarra

### 6.7.5. Pantalla táctil

La pantalla táctil tiene el problema de que no dispone de un lápiz especial para dibujar sobre ella, por lo que no es adecuada para escribir. Sin embargo si se puede utilizar para que los niños dibujen con los dedos. Tiene las mismas ventajas que los otros dispositivos electrónicos, los ejercicios se almacenan automáticamente. El mayor inconveniente de este dispositivo de entrada es que los niños no pueden apoyar la mano sobre la pantalla.

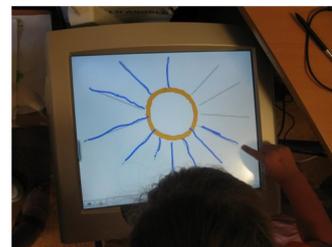


Figura 6.6: Ejercicio en pantalla táctil

## 6.8. Conclusiones sobre el funcionamiento del sistema

En esta sección se evalúan los resultados obtenidos por el sistema y se hará una evaluación de los distintos mecanismos de procesado de ejercicios del sistema.

### 6.8.1. Detección de marcas

En los ejercicios escaneados que se han evaluado se han producido errores en la detección de marcas en los casos en que los niños habían pintado parcialmente alguna de las marcas. Por lo tanto es indispensable que las marcas estén intactas, porque sino no será posible realizar la corrección.

### 6.8.2. Extracción de trazos

La extracción de trazos de una imagen escaneada presenta varios problemas. Si el alumno ha pintado varios trazos pegados formando una línea más gruesa será imposible separar esos trazos y el sistema sólo detectará un único trazo que pasa por el centro. Del mismo modo, si tiene algunas zonas más anchas se considerará que discurre por el centro de esa zona.

Otro problema es la deformación que sufre la imagen al ser impresa y escaneada y que hace que no coincida exactamente con el modelo. Esto provoca que con la extracción se creen elementos que no corresponden con trazos y pese a que la imagen se filtra para eliminarlos es posible que quede algún resto. Suelen ser trazos pequeños por lo que no afectan al resultado de la corrección.

### 6.8.3. Herramientas de escritura

Los ejercicios evaluados han sido realizados con distintos colores y herramientas de escritura para ver como se comporta el sistema. De entre todas el lápiz es la herramienta

menos adecuada, ya que si los alumnos no realizan los trazos con la suficiente presión aparecen de color gris muy débil y con los filtros que se aplican para tratar las imágenes escaneadas muchos de estos trazos se pierden. Todos los demás tipos de utensilios de escritura como rotuladores o bolígrafos no presentan problemas. En la figura 6.7 se muestran un ejemplo en el que se han perdido gran parte de los trazos realizados por el alumno.

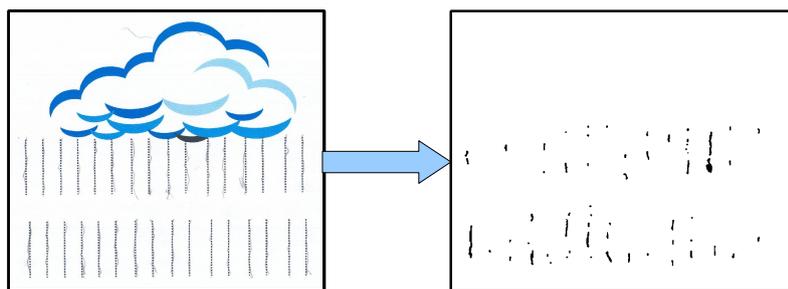


Figura 6.7: Ejercicio hecho a lápiz

## 6.9. Resultados método de corrección

El sistema ha sido evaluado con niños de un colegio de educación primaria que han realizado ejercicios sobre papel, con un tablet-pc y una pantalla táctil. Para cada uno de estos tres tipos de ejercicios se mostrarán las imágenes con los ejercicios y los resultados obtenidos. Los tres grupos de ejercicios se muestran por separado, y para cada uno se muestra una lista de imágenes y una tabla.

Así en las figuras 6.8 y 6.9 se muestran las imágenes de los ejercicios hechos con el tablet-pc y en las tablas 6.1 y 6.2 sus resultados. En las figuras 6.10 y 6.11 se listan los ejercicios realizados con la pantalla táctil y los resultados en las tablas 6.3 y 6.4. El último grupo de ejercicios que se muestra es el de los ejercicios escaneados en las figuras 6.12 y 6.13 y los resultados obtenidos con su evaluación en 6.5 y 6.6.

Para cada tabla de resultados cada fila contiene los resultados de un ejercicio y las columnas tienen el siguiente significado:

En la primera columna aparece el número que identifica a cada imagen. Los resultados que se presentan son la forma, distancia diferencia de longitud, completitud y nota global y se muestran en pares, primero el valor numérico y después la nota que se ha asignado entre 0 y 10. El valor de la forma es la desviación media de cada trazo con respecto al modelo, la distancia representa la media de distancia de cada trazo al modelo. Cada imagen se etiqueta con tres números, el número de identificación del alumno, el número del modelo y el número relativo del ejercicio. Las imágenes están ordenadas por alumno.

A continuación se hará una interpretación de algunos de los resultados mostrados en la primera tabla de los ejercicios realizados con el tablet-pc. En ella se observa que los ejercicios 4\_4\_3 y 5\_5\_4 han obtenido una buena puntuación lo que concuerda con las imágenes de dichos ejercicios. Otros ejercicios como el 9\_4\_3, el 1\_5\_7 o el 9\_1\_1 han obtenido una puntuación muy baja ya que como se puede apreciar en las imágenes de estos ejercicios no están hechos correctamente. En el ejercicio 2\_3\_2 o el 5\_3\_2 se puede observar que la puntuación de forma es alta, sin embargo la puntuación de diferencia de longitud baja. Como se observa en las imágenes de estos ejercicios la forma de los trazos es correcta, pero son discontinuos.

Se observa que algunos alumnos tienen problemas con los ejercicios sobre la pantalla táctil, por ejemplo el 1 y el 9. Requieren de un periodo de adaptación, como se observa con el ejercicio 17\_3\_2 y 17\_3\_3, donde se muestra el mismo ejercicio hecho por un alumno, apreciándose una notable mejoría después de repetirlo. Los mejores resultados se han obtenido con el ejercicio 1, puesto que es el ejercicio más sencillo. Los alumnos tienen más problemas con el ejercicio 3 que mezcla trazos de distintas direcciones. Además se puede apreciar que los alumnos tienden a sacar resultados proporcionales, es decir si tiene puntuación baja en un ejercicio es probable que también tenga una puntuación baja en el resto. Los resultados de los ejercicios en papel son mejores que con el resto de dispositivos, puesto que es a lo que están acostumbrados los alumnos.

### 6.9.1. Ejercicios tablet

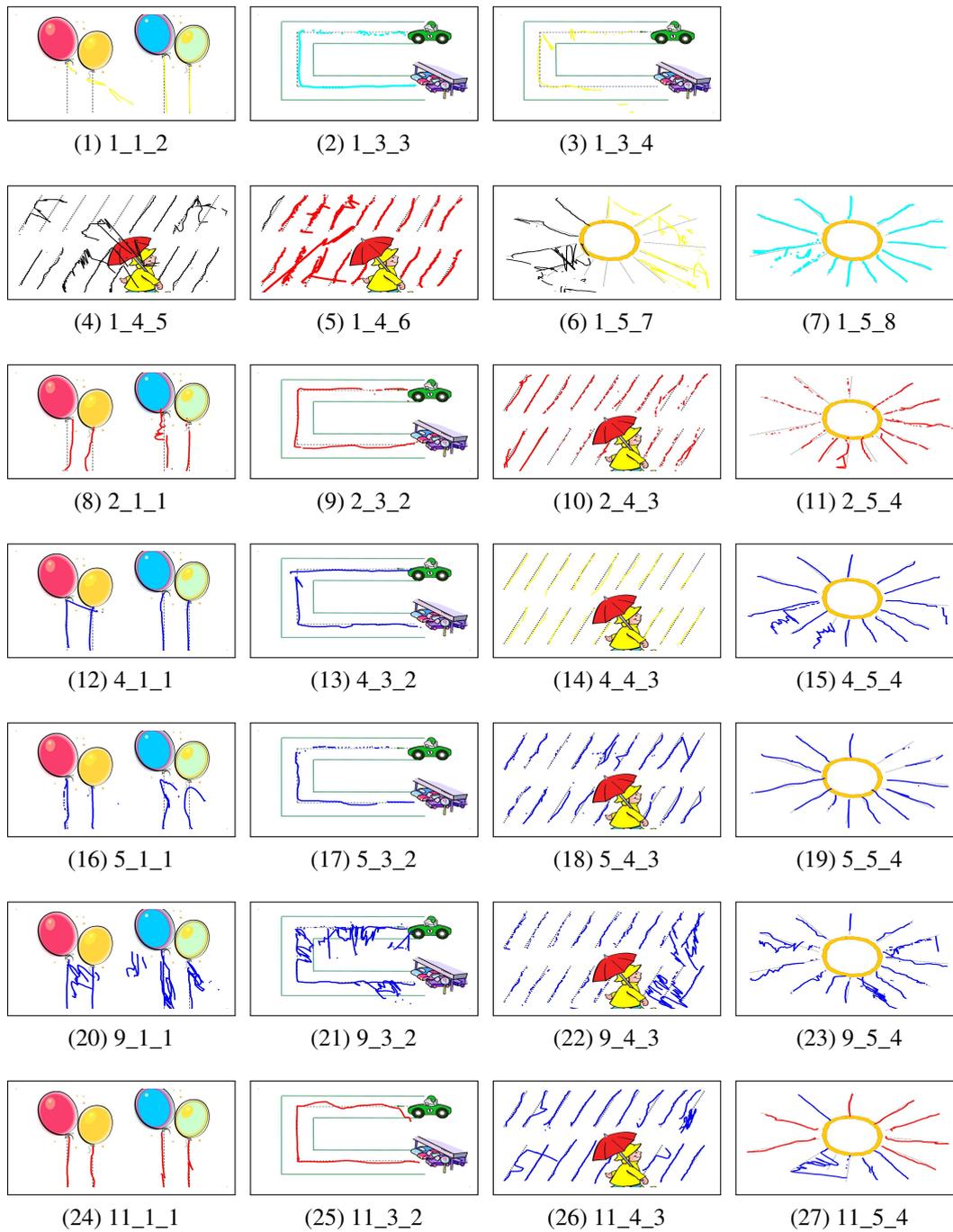


Figura 6.8: Ejercicios tablet-pc (1)

Nombre	Comp.		Dist.		Forma		Longitud		Nota
1_1_1	0.00	0	0.00	10	0.00	10	0.00	10	8
1_1_2	60.58	6	6.64	4	18.38	0	24.81	7	3
1_3_3	70.91	7	4.64	6	7.40	7	34.65	4	6
1_3_4	61.82	6	8.78	2	23.94	0	69.94	0	1
1_4_5	73.11	7	14.95	0	28.53	0	111.39	0	1
1_4_6	88.22	9	5.35	6	15.55	0	36.25	4	3
1_5_7	68.33	7	17.89	0	40.03	0	93.54	0	1
1_5_8	88.75	9	5.43	6	10.92	4	19.82	8	6
2_1_1	96.15	10	9.00	2	11.43	3	22.09	7	5
2_3_2	92.73	10	6.35	5	6.89	7	55.13	0	5
2_4_3	62.54	6	7.40	3	7.47	7	60.42	0	4
2_5_4	57.50	5	6.10	5	18.66	0	55.92	0	2
4_1_1	99.04	10	7.83	3	9.58	5	31.53	5	5
4_3_2	94.55	10	7.40	3	11.39	4	36.81	4	5
4_4_3	81.27	8	5.10	6	6.62	7	33.37	5	6
4_5_4	89.17	9	6.05	5	9.06	5	30.96	5	5
5_1_1	94.23	10	8.06	3	11.17	4	41.16	3	4
5_3_2	69.09	7	4.52	7	5.59	8	58.65	0	6
5_4_3	77.04	8	6.51	4	8.85	6	24.49	7	6
5_5_4	82.08	9	4.85	6	6.14	8	23.01	7	7
9_1_1	100.00	10	13.12	0	16.57	0	45.33	2	2
9_3_2	88.18	9	13.51	0	42.71	0	51.39	0	1
9_4_3	54.98	5	16.59	0	34.70	0	89.90	0	1
9_5_4	84.58	9	9.98	1	25.28	0	48.23	1	2
11_1_1	99.04	10	8.01	3	9.69	5	43.61	2	5
11_3_2	98.18	10	7.82	3	12.08	3	17.59	9	5
11_4_3	93.66	10	7.99	3	14.76	1	39.83	3	3
11_5_4	83.33	9	12.26	0	23.24	0	110.37	0	1

Tabla 6.1: Resultados ejercicios tablet-pc (1)

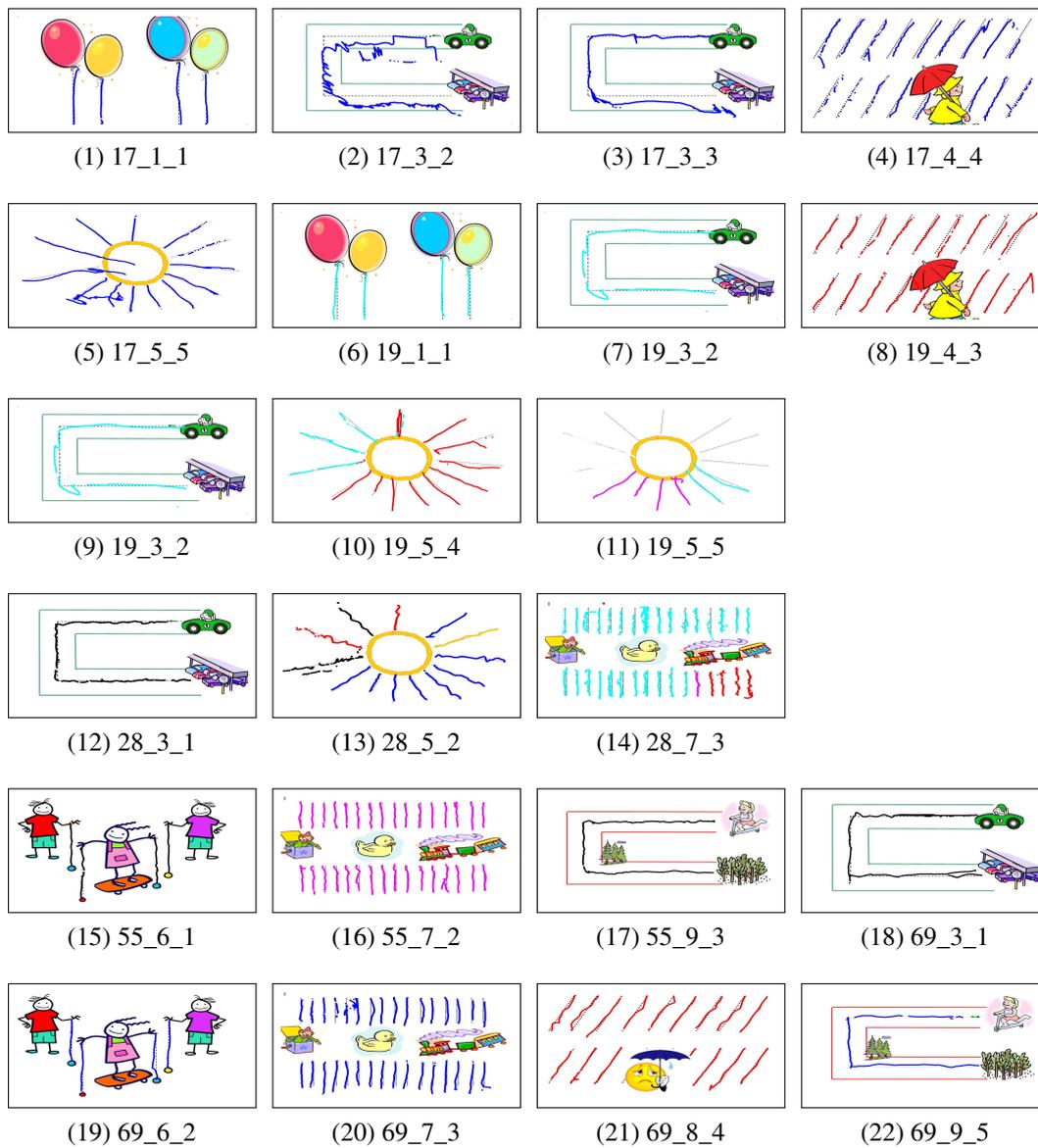


Figura 6.9: Ejercicios tablet-pc (2)

Nombre	Comp.		Dist.		Forma		Longitud		Nota
17_1_1	99.04	10	3.83	7	3.34	10	12.66	10	9
17_3_2	63.64	6	23.31	0	35.47	0	57.41	0	1
17_3_3	87.27	9	15.99	0	21.32	0	35.13	4	2
17_4_4	75.23	8	6.37	5	11.37	4	29.67	6	5
17_5_5	85.83	9	10.68	0	16.51	0	45.18	2	2
19_1_1	100.00	10	3.48	8	2.89	10	14.18	9	9
19_3_2	91.82	10	7.56	3	12.82	2	53.93	0	3
19_4_3	85.50	9	5.84	5	6.99	7	30.09	5	6
19_5_4	85.83	9	5.82	5	12.05	3	57.26	0	4
19_5_5	60.42	6	5.74	5	9.68	5	20.58	8	5
28_3_1	86.36	9	5.00	6	13.25	2	78.36	0	3
28_5_2	88.33	9	3.86	7	12.91	2	25.72	7	5
28_7_3	79.58	8	3.24	8	11.17	4	45.92	2	5
55_6_1	96.30	10	3.82	7	5.27	9	42.02	2	7
55_7_2	95.31	10	2.69	9	6.55	8	18.47	8	8
55_9_3	87.39	9	5.98	5	8.39	6	75.01	0	5
69_3_1	98.18	10	4.78	6	8.14	6	37.65	4	6
69_6_2	100.00	10	4.16	7	4.28	9	29.77	6	8
69_7_3	93.19	10	3.94	7	5.82	8	18.01	8	8
69_8_4	7.78	0	20.33	0	30.54	0	96.63	0	0
69_9_5	88.24	9	5.22	6	6.17	8	73.81	0	6

Tabla 6.2: Resultados ejercicios tablet-pc (2)

## 6.9.2. Ejercicios táctil

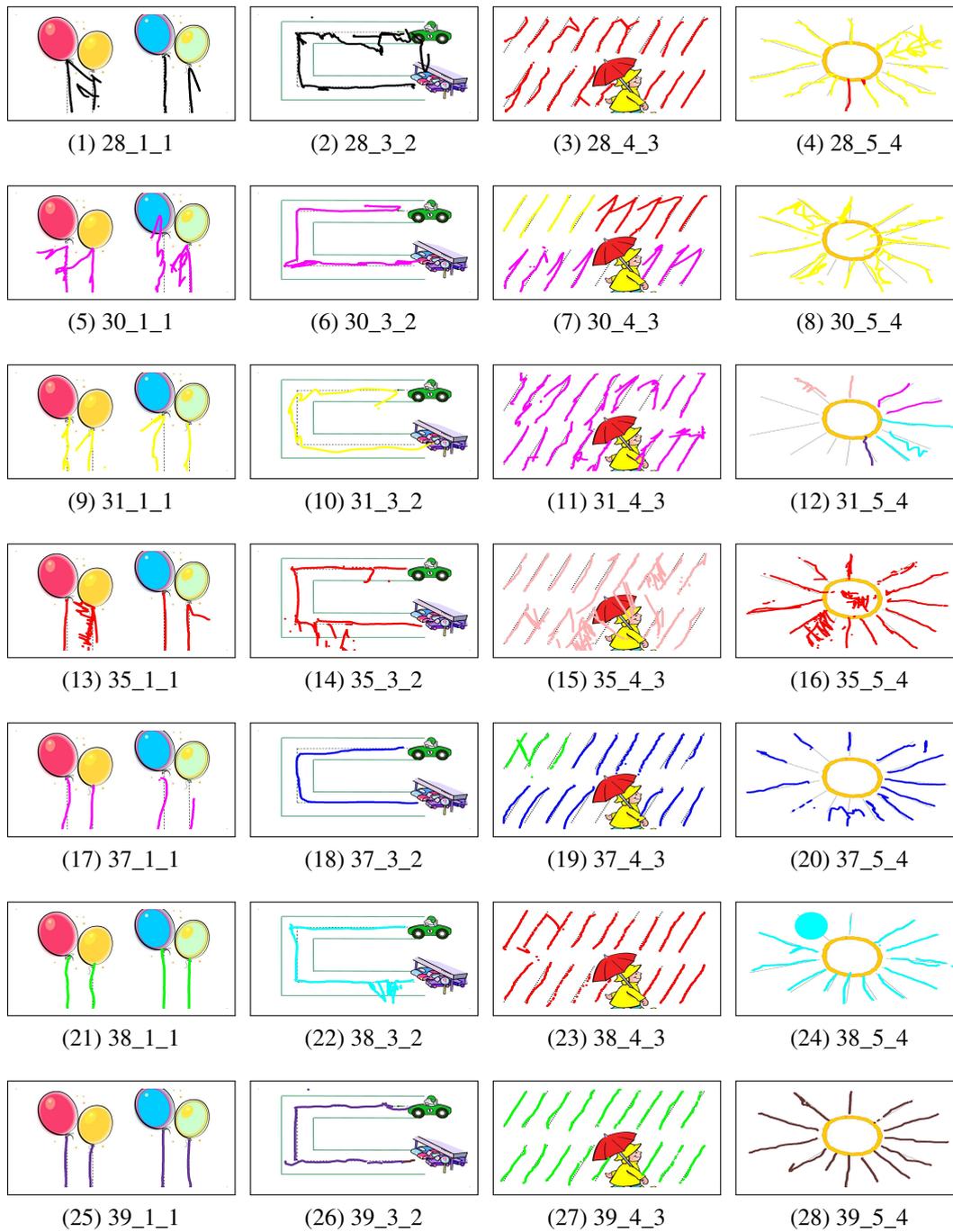


Figura 6.10: Ejercicios pantalla táctil (1)

Nombre	Comp.		Dist.		Forma		Longitud		Nota
28_1_1	99.04	10	9.93	1	15.22	0	46.77	1	2
28_3_2	90.91	10	18.01	0	26.59	0	47.53	1	2
28_4_3	87.92	9	7.35	3	11.67	3	26.45	6	4
28_5_4	88.33	9	11.49	0	26.30	0	62.80	0	1
30_1_1	98.08	10	18.70	0	28.46	0	110.98	0	2
30_3_2	99.09	10	7.33	3	17.16	0	67.59	0	2
30_4_3	91.54	10	7.65	3	10.33	4	66.70	0	4
30_5_4	84.17	9	14.31	0	24.41	0	117.28	0	1
31_1_1	99.04	10	15.08	0	20.30	0	74.87	0	2
31_3_2	92.73	10	14.46	0	21.79	0	34.26	4	2
31_4_3	93.96	10	13.93	0	18.44	0	95.39	0	2
31_5_4	53.33	5	7.00	4	18.04	0	24.40	7	3
35_1_1	99.04	10	12.47	0	20.33	0	51.32	0	2
35_3_2	98.18	10	8.81	2	23.93	0	73.14	0	2
35_4_3	93.96	10	15.59	0	22.36	0	60.77	0	2
35_5_4	88.33	9	8.40	2	19.18	0	51.97	0	2
37_1_1	89.42	9	18.10	0	5.87	8	51.17	0	5
37_3_2	90.91	10	7.25	4	9.58	5	1.85	10	6
37_4_3	89.43	9	7.85	3	8.38	6	23.53	7	6
37_5_4	72.08	7	7.79	3	12.92	2	23.42	7	4
38_1_1	100.00	10	3.92	7	3.55	10	19.30	8	9
38_3_2	99.09	10	11.75	0	29.33	0	66.46	0	2
38_4_3	79.15	8	5.89	5	6.87	7	35.33	4	6
38_5_4	90.83	10	5.81	5	10.53	4	21.38	8	6
39_1_1	99.04	10	4.45	7	2.03	10	19.09	8	9
39_3_2	95.45	10	6.31	5	17.03	0	78.96	0	3
39_4_3	86.10	9	4.62	6	5.89	8	24.31	7	7
39_5_4	92.08	10	4.10	7	8.59	6	21.30	8	7

Tabla 6.3: Resultados ejercicios táctil (1)

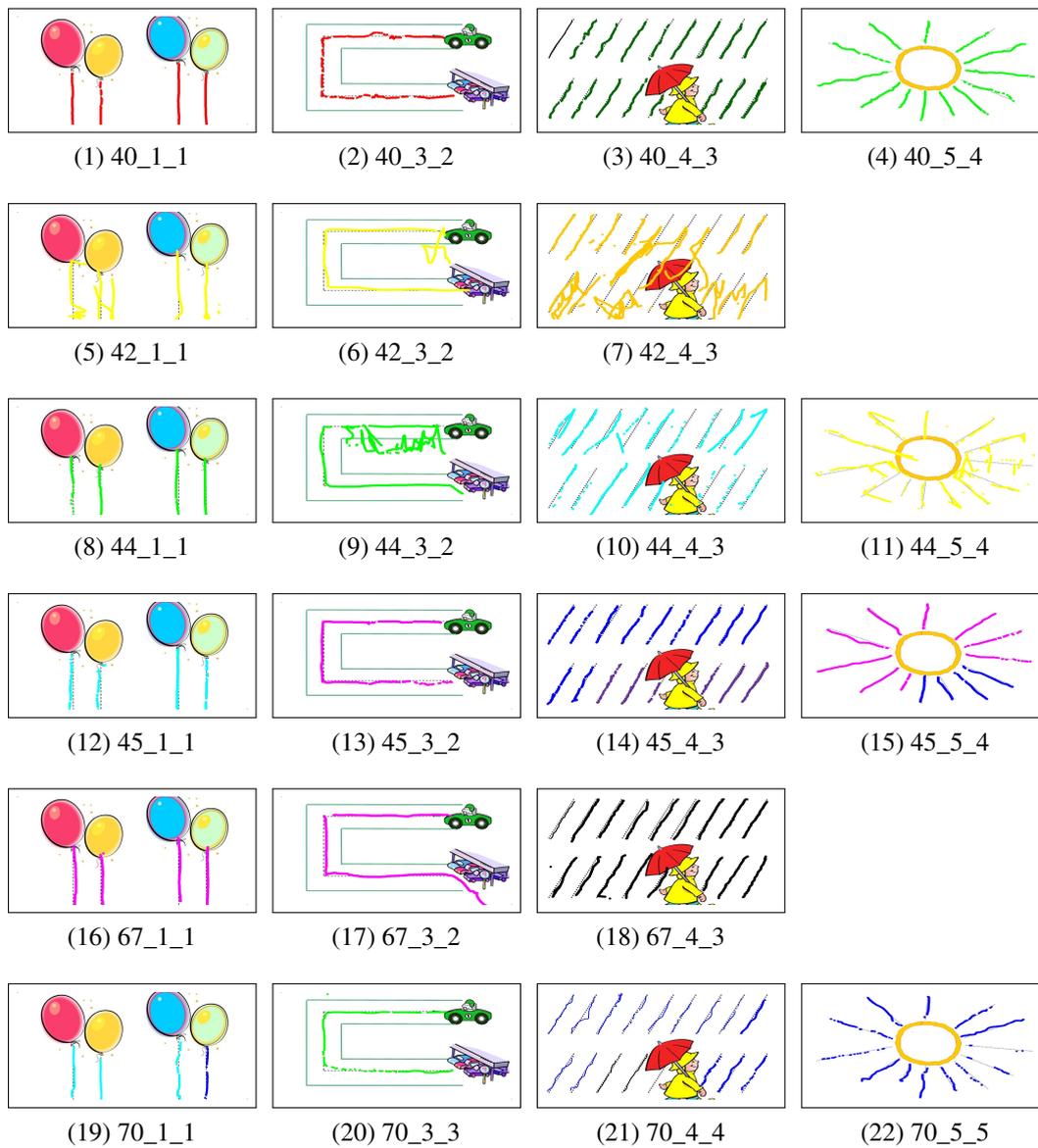


Figura 6.11: Ejercicios pantalla táctil (2)

Nombre	Comp.		Dist.		Forma		Longitud		Nota
40_1_1	91.35	10	2.09	9	1.71	10	28.39	6	9
40_3_2	56.36	5	5.87	5	8.33	6	95.33	0	4
40_4_3	81.87	8	4.14	7	6.13	8	23.27	7	7
40_5_4	76.67	8	3.15	8	9.37	5	22.03	7	6
42_1_1	100.00	10	14.54	0	19.68	0	49.15	1	2
42_3_3	96.36	10	6.85	4	4.88	9	8.33	10	8
42_4_4	77.34	8	22.93	0	24.76	0	133.58	0	1
44_1_1	83.65	9	6.22	5	2.95	10	40.31	3	7
44_3_2	94.55	10	6.57	4	9.16	5	48.03	1	5
44_4_3	55.89	5	7.84	3	8.57	6	47.03	1	4
44_5_4	58.75	6	11.45	0	46.05	0	50.06	0	1
45_1_1	72.12	7	3.90	7	1.78	10	54.68	0	6
45_3_2	80.00	8	5.06	6	5.22	9	59.49	0	6
45_4_3	59.21	6	3.78	7	5.62	8	41.21	3	6
45_5_4	78.75	8	3.69	8	6.84	7	19.13	8	7
67_1_1	100.00	10	6.41	4	2.54	10	24.78	7	8
67_3_2	94.55	10	16.35	0	12.47	3	27.28	6	4
67_4_3	96.68	10	6.31	5	5.69	8	22.87	7	7
67_5_4	91.67	10	14.77	0	22.38	0	49.28	1	2
70_1_1	88.46	9	3.63	8	3.61	10	34.76	4	8
70_3_2	0.00	0	0.00	10	0.00	10	0.00	10	8
70_3_3	70.91	7	5.71	5	19.86	0	75.76	0	2
70_4_4	77.34	8	4.28	7	8.95	6	30.71	5	6
70_5_5	40.00	3	4.16	7	9.52	5	37.40	4	4

Tabla 6.4: Resultados ejercicios táctil (2)

### 6.9.3. Ejercicios escaneados

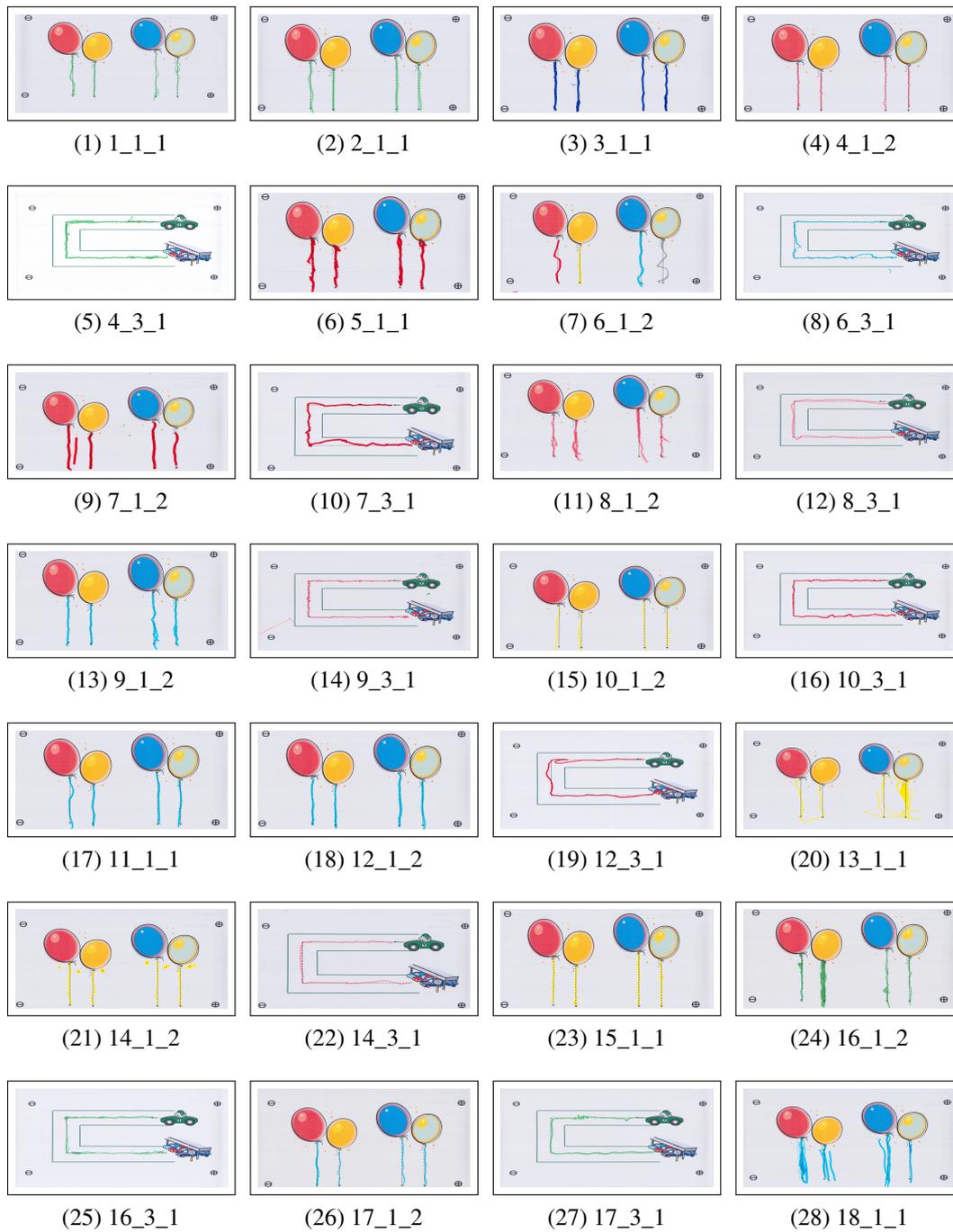


Figura 6.12: Ejercicios pantalla táctil (1)

Nombre	Comp.	Dist.	Forma	Longitud	Nota
1_1_1	100.00 10	2.74 9	10.19 5	18.61 8	7
2_1_1	100.00 10	2.82 8	3.93 10	12.14 10	9
3_1_1	100.00 10	2.25 9	6.28 8	13.14 10	9
4_1_2	100.00 10	1.85 10	5.55 8	5.60 10	9
4_3_1	100.00 10	2.48 9	11.20 4	16.93 9	7
5_1_1	99.51 10	2.69 9	12.08 3	22.25 7	6
6_1_2	100.00 10	4.46 7	15.12 0	20.59 8	5
6_3_1	98.16 10	4.44 7	12.97 2	42.75 2	4
7_1_2	99.01 10	4.67 6	5.68 8	13.21 10	8
7_3_1	98.62 10	5.48 6	12.86 2	44.90 2	4
8_1_2	100.00 10	3.83 7	12.14 3	27.43 6	5
8_3_1	100.00 10	5.04 6	9.00 5	7.87 10	7
9_1_2	100.00 10	3.02 8	7.82 6	16.81 9	7
9_3_1	97.70 10	1.51 10	8.11 6	0.00 10	8
10_1_2	100.00 10	3.81 7	8.13 6	17.12 9	7
10_3_1	98.62 10	2.90 8	11.36 4	7.87 10	7
11_1_1	100.00 10	3.17 8	6.73 7	14.65 9	8
12_1_2	100.00 10	2.35 9	5.24 9	13.14 10	9
12_3_1	98.16 10	4.48 7	16.03 0	38.90 3	4
13_1_1	98.52 10	11.45 0	29.21 0	27.87 6	3
14_1_2	100.00 10	5.12 6	9.79 5	17.95 9	7
14_3_1	98.62 10	2.43 9	8.50 6	1.85 10	8
15_1_1	100.00 10	2.43 9	7.71 7	11.33 10	8
16_1_2	99.01 10	2.51 9	7.02 7	14.88 9	8
16_3_1	97.24 10	2.80 8	10.57 4	62.19 0	5
17_1_2	93.60 10	2.76 9	6.15 8	21.27 8	8
17_3_1	97.70 10	3.74 7	12.76 2	37.60 4	5
18_1_1	100.00 10	6.43 4	9.65 5	37.90 4	5

Tabla 6.5: Resultados ejercicios escaneados (1)

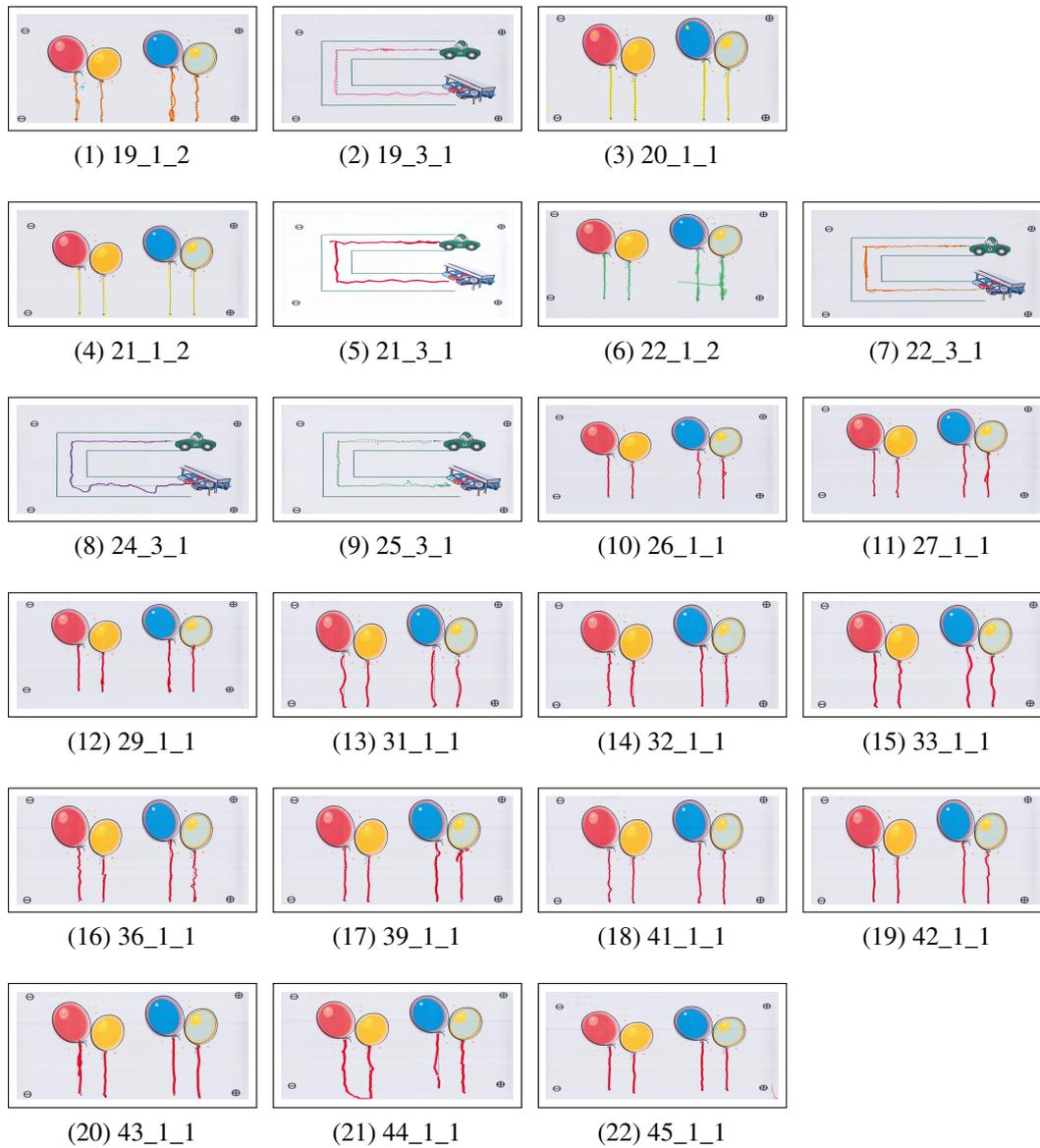


Figura 6.13: Ejercicios escaneados (2)

Nombre	Comp.		Dist.		Forma		Longitud		Nota
19_1_2	100.00	10	3.10	8	12.25	3	26.78	6	6
19_3_1	99.08	10	3.02	8	12.79	2	10.19	10	6
20_1_1	100.00	10	2.56	9	6.20	8	9.63	10	9
21_1_2	100.00	10	3.27	8	5.87	8	15.59	9	8
21_3_1	99.54	10	3.08	8	11.38	4	10.56	10	7
22_1_2	100.00	10	5.00	6	15.57	0	17.01	9	5
22_3_1	97.70	10	2.18	9	9.35	5	73.97	0	5
24_3_1	99.54	10	4.99	6	16.87	0	12.96	10	5
25_3_1	94.93	10	3.07	8	16.13	0	59.03	0	3
26_1_1	99.51	10	1.85	10	5.84	8	9.27	10	9
27_1_1	100.00	10	2.09	9	6.10	8	9.09	10	9
29_1_1	100.00	10	1.67	10	4.26	9	8.12	10	9
31_1_1	98.03	10	4.55	7	9.68	5	14.74	9	7
32_1_1	100.00	10	2.47	9	6.34	8	9.76	10	9
33_1_1	100.00	10	2.96	8	8.45	6	16.11	9	7
36_1_1	100.00	10	2.04	9	8.34	6	9.16	10	8
39_1_1	100.00	10	2.68	9	8.49	6	15.17	9	8
41_1_1	100.00	10	2.33	9	6.28	8	10.12	10	9
42_1_1	100.00	10	1.93	9	6.22	8	10.67	10	9
43_1_1	100.00	10	2.52	9	4.51	9	12.82	10	9
44_1_1	97.54	10	3.77	7	7.53	7	23.14	7	7
45_1_1	100.00	10	2.20	9	3.62	10	13.68	10	9

Tabla 6.6: Resultados ejercicios escaneados (2)

### 6.9.4. Gráficas de los resultados

El sistema dispone de la facilidad de mostrar estos resultados gráficamente, lo cual ayuda al profesor a entender mejor los resultados obtenidos por los alumnos. Estos resultados los puede filtrar de manera que por ejemplo sólo se muestren los de un trazo o modelo concreto. Además puede comparar de esta manera los resultados de los alumnos. Las figuras 6.14, 6.15 y 6.16 muestran los datos de las tablas del capítulo anterior gráficamente. Los datos están agrupados por alumno y se muestran la media de las notas obtenidas para todos los ejercicios realizados. El identificador de cada alumno se encuentra debajo de la tabla. Así se puede ver en la figura 6.14 fácilmente por ejemplo que el alumno 19 y 55 han sacado la mejor puntuación y otros como el 1, 9 y 17 tienen más problemas a la hora de hacer los ejercicios.

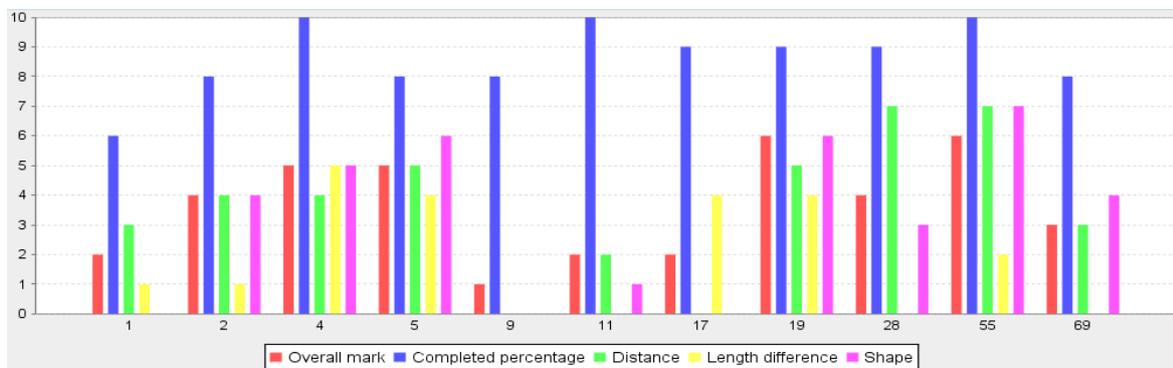


Figura 6.14: Media de resultados por alumnos de ejercicios de tablet-pc

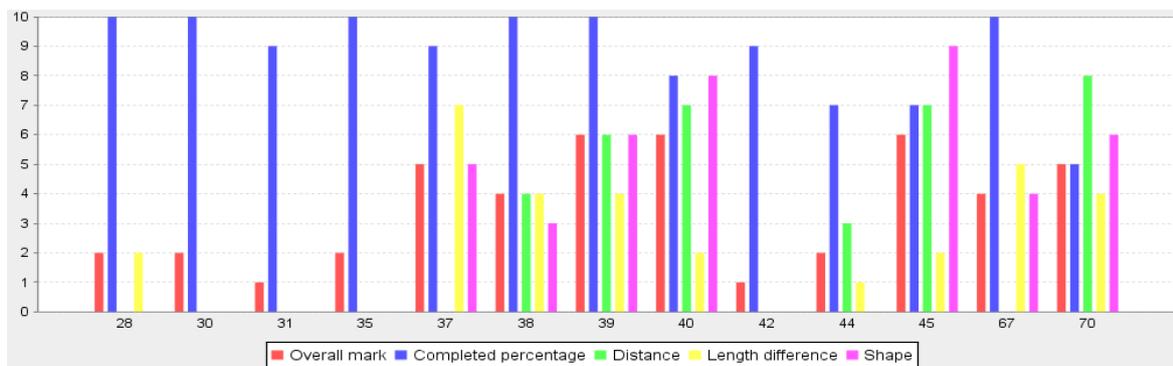


Figura 6.15: Media de resultados por alumnos de ejercicios de pantalla táctil

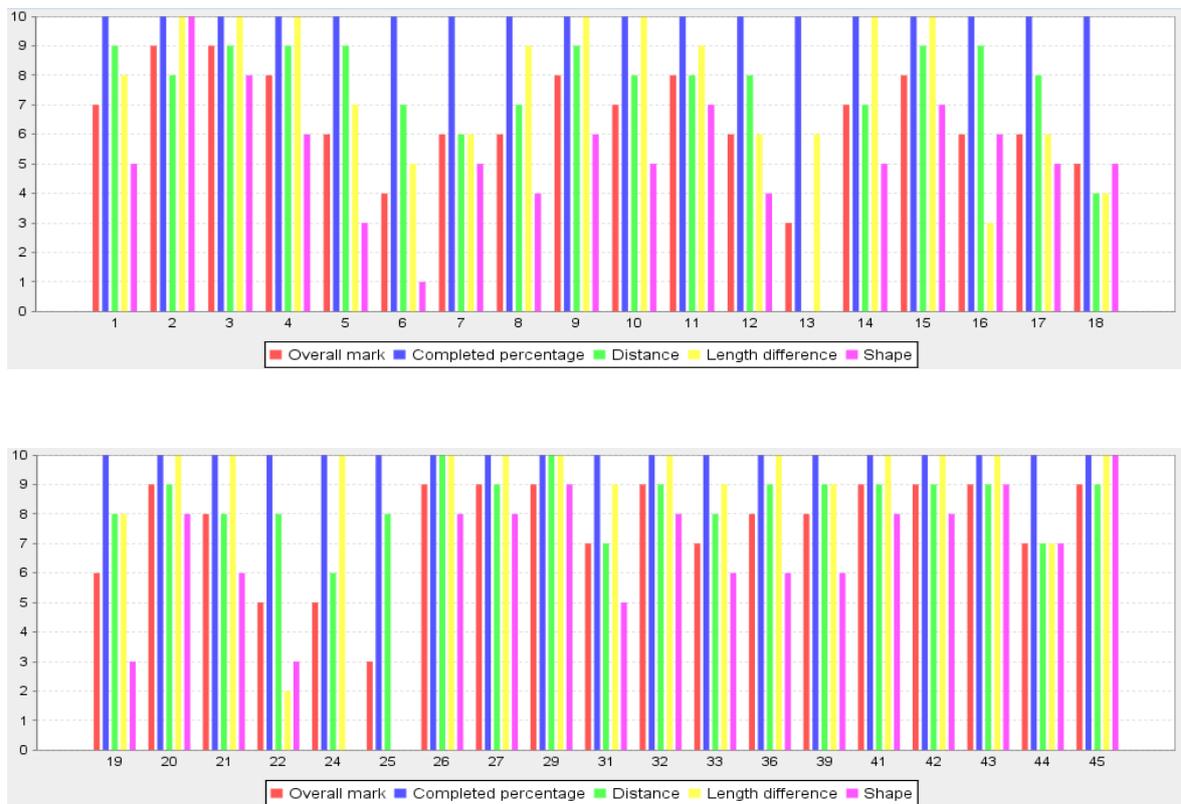


Figura 6.16: Media de resultados por alumnos de ejercicios escaneados



# Capítulo 7

## Conclusiones y futuros trabajos

En este capítulo se describen los resultados obtenidos con el proyecto, haciendo un resumen de sus características. Además se propondrán posibles mejoras y funcionalidad que se podría añadir para aumentar sus capacidades.

En la sección 7.1 se hace un resumen de las principales propiedades del proyecto. En la última sección se hace referencia a las limitaciones que tiene el proyecto y finalmente se proponen una serie de mejoras y nuevas funcionalidades que se pueden añadir al sistema.

### 7.1. Resumen y aportaciones

El sistema que se ha creado permite imprimir y almacenar los ejercicios que realizan los alumnos en papel. Esto se realiza a través de una interfaz de impresión mediante la cual se asigna un código de barras a cada ejercicio para que se pueda identificar el modelo y el alumno al que se le ha asignado posteriormente más fácilmente. Una vez resueltos y escaneados dichos ejercicios se pueden incorporar al sistema.

Además se han implementado los métodos necesarios para corregir los ejercicios automáticamente. El algoritmo los evalúa según diferentes puntos de vista: forma, distancia del trazo del ejercicio al modelo, longitud del trazo en comparación con la del modelo y completitud del ejercicio. A partir de estos datos se realiza la corrección. Para poder corregir los ejercicios realizados sobre papel se ha definido un mecanismo para incorporar marcas en ellos de forma que se puedan recolocar y segmentar posteriormente. Además se han implementado una serie de algoritmos que permiten extraer los trazos del ejercicio y convertirlos a una representación geométrica.

Se han incorporado los mecanismos para realizar los ejercicios sobre una pantalla táctil. Para ello se ha diseñado una interfaz que permite cargar los distintos modelos que haya almacenados en el sistema. Los ejercicios se corrigen sobre la marcha y se almacenan en el sistema cuando se finaliza la realización. Además se pueden reproducir posteriormente, de manera que el proceso por el que el alumno realiza el ejercicio puede ser inspeccionado y no

se pierde. El sistema permite visualizar como los alumnos los alumnos llegan a una solución de un ejercicio, lo cual no había sido nunca posible con las herramientas tradicionales, es decir escribiendo sobre papel. Todas las teorías de enseñanza destacan la importancia de que los estudiantes reflexionen sobre cómo llegaron al resultado final y, por lo tanto, es importante tanto la solución como el proceso que permite al grupo llegar a ella [32].

Se han incorporado a la corrección los mecanismos para caracterizar el nivel de conocimiento del alumno que permita detectar posibles disgrafías de este. En el sistema se almacena la relación de ejercicios realizada por cada alumno y a partir de los resultados obtenidos se puede inferir el nivel del alumno. Para poder visualizar estos datos se ha creado una interfaz que los muestra en forma de tabla o de gráficas.

Finalmente se ha definido un entorno que integra todas las funcionalidades descritas y que permite monitorizar los resultados según diferentes puntos de vista. Para ello se ha diseñado una interfaz que engloba todas las características e interfaces definidas anteriormente.

Cada fase de este proceso se ha realizado con usuarios reales lo que ha servido para mejorar las funcionalidades de la arquitectura así como la interfaz del sistema. Esta aproximación ha permitido, además, comprobar la utilidad práctica y la validez de las propuestas que sustentan este trabajo.

Parte del trabajo realizado se ha presentado y publicado en el workshop “ECTEL 2008 - Workshop on Inclusive E-Learning” [33].

## 7.2. Futuros trabajos

El sistema ha sido diseñado para abarcar la enseñanza de la escritura entre los tres y cuatro años. Se centra en ejercicios de preescritura pautados y no admite que los trazos sean demasiado “complejos”. Tampoco se permite que haya varios tipos de trazos distintos dentro de una imagen.

Para corregir esto, se podría aumentar la capacidad de corregir ejercicios con múltiples trazos distintos identificando automáticamente de que trazo se trata en cada ocasión. Además se podría aumentar la funcionalidad para abarcar otras edades y otro tipo de ejercicios, como pueden ser copiar un dibujo sin pautas o realizar formas más complejas como letras. Esto requeriría extender el sistema de corrección para adaptarlo a estos nuevos ejercicios. También se podría aprovechar la interfaz de dibujo y el sistema de corrección ya existente para realizar otro tipo de ejercicios relacionados con la psicomotricidad.

Además se podría mejorar las características del sistema añadiendo nuevas funcionalidades, como la que se propone en el sistema Scriptot [16] para que se muestre a los alumnos una animación de como deben escribir las letras. Sería interesante añadir nuevas capacidades al algoritmo de corrección y hacer un análisis cinemático de los trazos, de esta forma se examinaría la velocidad con la que el alumno hace los trazos además de si la forma es correcta.

Una de las dificultades que tienen los niños al trabajar con las pantallas táctiles es que al apoyar la mano el cursor puede saltar a otra posición y no responder correctamente. Esto

---

ocurre sobre todo con los monitores y rara vez con el tablet-pc. Para corregir esto se podría mejorar la interfaz de dibujo para evitar que el cursor pegue saltos cuando el alumno apoya la mano.

En cuanto a los dispositivos de entrada se puede investigar el uso de nuevos dispositivos o de alternativas a los actuales. Por ejemplo buscar una alternativa al tablet-pc en el que no haya que hacer tanta presión sobre la pantalla o un nuevo tipo de dispositivo.



# Bibliografía

- [1] J. C. Read, “A study of the usability of handwriting recognition for text entry by children,” *Interact. Comput.*, vol. 19, no. 1, pp. 57–69, 2007. 1
- [2] J. Allan, T. Allen, and N. Sherkat, “Confident assessment of children’s handwritten responses,” in *IWFHR '02: Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, (Washington, DC, USA), p. 508, IEEE Computer Society, 2002. 1
- [3] C. Rémi, C. Frelicot, and P. Courtellemont, “Automatic analysis of the structuring of children’s drawings and writing,” *Pattern Recognition*, no. 35, pp. 1059–1069, 2002. 1, 13
- [4] R. Guest, S. Chindaro, M. Fairhurst, and J. Potter, “Optimisation procedures for diagnostic processing of hand-drawn geometric figures,” *Electronic letters*, vol. 39, no. 2, pp. 205–206, 2003. 1
- [5] J. Jiménez Vargas, *Neurofisiología psicológica fundamental*. Ed. Científico médica, 1982. 7
- [6] J. A. GarcíaÑúñez and P. P. Berruelo, *Psicomotricidad y Educación Infantil*. Madrid: Editorial CEPE, 1994. 8
- [7] S. S. Gubbay, *The Clumsy Child*. Philadelphia: Saunders, 1975. 8
- [8] E. Adi Japha and N. H. Freeman, “Development of differentiation between writing and drawing systems.,” *Developmental Psychology*, vol. 37, pp. 101–114, 2001. 9
- [9] E. Bialystok, “A study of the usability of handwriting recognition for text entry by children,” *Journal of Experimental Child Psychology*, vol. 76, pp. 173–179, 2000. 9
- [10] E. Ferreiro and A. Teberoski, *Los sistemas de escritura en el desarrollo del niño*. Madrid: Siglo XXI, 1972. 9
- [11] J. d. Ajuriaguerra, *Manual de psiquiatría infantil*. España: Elsevier, 1997. 10
- [12] M.Ñosty, *La buena letra 3ª Ed.* México, DF: Fernández Editores, 1992. 10

- [13] L. Picq and P. Vayer, *Educación psicomotriz y retraso mental*. Barcelona: Ed. Científico médica, 1969. 12
- [14] <http://www.donjohnston.com/products/cowriter/index.html>. 13
- [15] [http://www.writeonhandwriting.com/demo/c\\_0main.html](http://www.writeonhandwriting.com/demo/c_0main.html). 13
- [16] S. Djeziri, W. Guerfali, R. Plamondon, and J. M. Robert, “Learning handwriting with pen-based systems: computational issues,” *Pattern Recognition*, no. 35, pp. 1049–1057, 2002. 13, 114
- [17] M. Fairhurst, T. Linnell, S. Glénat, R. Guest, L. Heutte, and T. Paquet, “Developing a generic approach to online automated analysis of writing and drawing tests in clinical patient profiling,” *Behavior Research Methods*, vol. 40, no. 1, pp. 290–303, 2008. 14
- [18] J. C. Sánchez, “Qué son los metadatos,” *SIEL*, 2002. 18
- [19] Wikipedia. [http://es.wikipedia.org/wiki/Modelo\\_de\\_color\\_RGB](http://es.wikipedia.org/wiki/Modelo_de_color_RGB). 51
- [20] Wikipedia. [http://es.wikipedia.org/wiki/Lenguaje\\_de\\_programación\\_Java](http://es.wikipedia.org/wiki/Lenguaje_de_programación_Java). 52
- [21] Wikipedia. <http://es.wikipedia.org/wiki/JDBC>. 52
- [22] Wikipedia. [http://es.wikipedia.org/wiki/Base\\_de\\_datos\\_relacional](http://es.wikipedia.org/wiki/Base_de_datos_relacional). 52
- [23] L. G. Shapiro and G. C. Stockman, *Computer Vision*. Prentice Hall, January 2001. 59
- [24] <http://www.barcodeisland.com/ean13.phtml>. 63
- [25] B. Kégl and A. Krzyżak, “Piecewise linear skeletonization using principal curves,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 59–74, 2002. 68
- [26] U. d. D. Grupo PAS, “Métodos para el cálculo del esqueleto.” <http://www.pas.deusto.es/recursos/esqueleto.pdf>. 68
- [27] Wikipedia. <http://es.wikipedia.org/wiki/Quadtree>. 72
- [28] S. D. Cohen and L. J. Guibas, “Partial matching of planar polylines under similarity transformations,” in *SODA '97: Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*, (Philadelphia, PA, USA), pp. 777–786, Society for Industrial and Applied Mathematics, 1997. 74
- [29] R. J. Erich Gamma, Richard Helm and J. M. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series. 78
- [30] M. Scriven, “The methodology of evaluation,” *Curriculum evaluation*, 1967. 87

- 
- [31] B. Barros Blanco, “Aprendizaje colaborativo en enseñanza a distancia: entorno genérico para configurar, realizar y analizar actividades en grupo.,” *Tesis doctoral, Universidad Politécnica de Madrid.*, 1999. 87
- [32] J. S. Brown, “Process versus product: a perspective on tools for communal and informal electronic learning,” vol. 1, no. 2, pp. 179–201, 1983. 114
- [33] B. Barros, R. Conejo, A. deDiego Cottinelli, and J. Garcia-Herreros, “Modelling pre-writing tasks to improve graphomotricity processes,” *ECTEL 2008 - Workshop on Inclusive E-Learning*, 2008. 114
- [34] J. Bloch, *Effective Java programming language guide*. Mountain View, CA, USA: Sun Microsystems, Inc., 2001.
- [35] M. Parsian, *JDBC Metadata, MySQL, and Oracle Recipes: A Problem-Solution Approach (Expert’s Voice in Java)*. Berkely, CA, USA: Apress, 2006.