

Universidad  
Politécnica  
de Cartagena



**industriales**  
etsii UPCT

# **APLICACIÓN INFORMÁTICA PARA ENTORNO CLÍNICO CON ACCESO A BASE DE DATOS.**

## **Desarrollo del test Wisconsin para detección de trastornos psicológicos.**

**Titulación:** Ingeniería Industrial.

**Alumno/a:** Plácido Aroca Valverde

**Director/a/s:** Juan Suardiaz Muro/Juan  
Antonio López Riquelme

**Departamento:** Tecnología Electrónica.

Cartagena 10 de Octubre de 2014.

## ÍNDICE

	Nº Pag.
1. INTRODUCCIÓN.....	6
1.1. Denominación del proyecto .....	6
1.2. Naturaleza del proyecto.....	6
1.2.1. Objetivo general : Descripción.....	6
1.2.2. Objetivos específicos: Resumen.....	7
1.2.3. Justificación.....	7
1.2.4. Fases del proyecto.....	8
2. FUNDAMENTOS Y ADAPTACIÓN DEL TEST .....	9
2.1. Descripción del test de clasificación de tarjetas Wisconsin.....	9
2.1.1. Datos del test.....	11
2.1.2. Materiales del test “Clásico” .....	11
2.1.3. Descripción general.....	13
2.1.4. Requisitos profesionales.....	13
2.2. Descripción del test de clasificación de tarjetas Wisconsin .....	14
2.2.1. Instrucciones de preparación y aplicación antes del test.....	14
2.2.2. Instrucciones de aplicación durante el test.....	16
2.2.3. Anotación tradicional de las respuestas.....	17
2.3. Dimensiones valoradas y evaluación del test.....	18
2.3.1. Dimensiones valoradas.....	18
2.3.2. Evaluación del test.....	24
3. PREPARACIÓN DE HERRAMIENTAS Y ENTORNO DE DESARROLLO .....	28
3.1. Herramientas de desarrollo.....	28
3.1.1. Lenguaje de programación java.....	28
3.1.2. Servidor de base de datos MySQL: XAMPP.....	30
3.2. Instalación de paquetes de software .....	31
3.2.1. Instalación y configuración de XAMPP.....	31
3.2.2. Instalación del Java Development Kit (JDK).....	34
3.2.3. Descarga e Instalación del Id de NetBeans .....	35
3.3. Breve descripción del entorno de trabajo.....	37
3.3.1. Entorno de trabajo de NetBeans .....	38
3.3.2. Entorno de trabajo de MySQL.....	38
4. INICIO Y ORGANIZACIÓN DEL PROYECTO.....	41
4.1. Esquema general de la aplicación:”Boceto”.....	41
4.2. Inicio en la creación del proyecto.....	43
4.3. Diseño de menú principal.....	45
5. REGISTRO DE PACIENTES Y BASES DE DATOS.....	51
5.1. Configuración de la conexión deMySQL.....	51
5.1.1 Preparación de la conexión .....	51
5.1.2 Requisitos de almacenamiento en el registro de un sujeto .....	52
5.1.3 Creación y configuración de las tablas en la base de datos.....	53

5.1.4 Descripción de la clase que realiza la conexión a la base de datos: ConexionMySQL.....	56
5.2. Diseño de la interface: Registro.....	59
5.2.1 Diseño de la interface Registro: Zona I .....	59
5.2.2 Programación de Registro: Zona II.....	62
6. FORMULARIO DE PREPARACIÓN E INICIO DEL TEST.....	77
6.1. Interface: Inicio.....	77
6.1.1. Zona I: Formulario de inicio.....	79
6.1.2. Zona II: Ventana de respuestas.....	86
7. EJECUCIÓN DEL TEST.....	88
7.1. Diseño de la interface.....	88
7.1.1. Diseño de la Interface en Idr.....	88
7.1.2. Zona II: Ventana de respuestas.....	90
7.1.3. Programación de los botones en la interface.....	93
7.2. Descripción de la matriz salida[][].....	101
7.3. Conversión y almacenamiento de la matriz de respuestas en la base de datos...	104
8. HISTORIAL.....	107
8.1. Diseño de la interface de historiales.....	107
8.1.1. Interface de búsqueda de test.....	107
8.1.2. Interface-Ventana de muestra de resultados, Clase:.....	110
8.2. Forma de ejecución en el cálculo y muestra de resultados.....	112
9. RESULTADOS ESTADÍSTICOS.....	114
9.1. Transformación de las puntuaciones directas en típicas .....	114
9.1.1. Ingreso de ábacos y tablas tabuladas en la base de datos.....	116
9.1.2. Funcionamiento del programa en la muestra de resultados estadísticos ...	119
9.2. Muestra normativa.....	121
9.2.1. Muestra de tipificación.....	121
9.2.2. Variables normativas .....	121
9.2.3. Normas por categorías.....	122
9.2.4. Estadísticos de la muestra general .....	122
9.3. Interpretación.....	122
9.3.1. Comparación con los baremos.....	123
9.3.2. Comparación con los valores de referencia.....	124
9.4. Programación del Visor de tablas y métodos.....	126
9.4.1. Descripción del Visor.....	126
10. REVISIÓN FINAL.....	129
10.1. Esquema general de la estructura final de la aplicación .....	129
10.2. Bibliografía.....	132
10.3. Futuras mejoras.....	132
10.4. Resumen.....	133
10.5. Contenido CD del proyecto .....	134
ANEXO I. Tablas y ábacos.....	135
ANEXO II. Manual de usuario.....	141
ANEXO III. Ejemplos de Aplicación.....	171
ANEXO IV. Código de programa.....	180





## INTRODUCCIÓN

---

---

### 1.1. DENOMINACIÓN DEL PROYECTO

*Aplicación informática para entorno clínico con acceso a base de datos. Desarrollo del test Wisconsin para detección de trastornos psicológicos.*

### 1.2. NATURALEZA DEL PROYECTO

#### **1.2.1. Objetivo general: Descripción del proyecto**

Realización de una aplicación informática para el diagnóstico y seguimiento médico de pacientes. Basada en un test, denominado Wisconsin o WCST, desarrollado originalmente para evaluar el razonamiento abstracto y la habilidad para cambiar las estrategias cognitivas como respuesta a eventuales modificaciones ambientales. Puede ser considerado como una medida de la función ejecutiva que requiere habilidad para desarrollar y mantener las estrategias de solución de problemas que resultan adecuadas para conseguir unos objetivos, a través de condiciones que implican cambios de estímulos. El WCST requiere estrategias de planificación, indagaciones organizadas, utilizando el “feedback” ambiental para cambiar esquemas, orientación de la conducta hacia el logro de un objetivo y modulación de las respuestas impulsivas. Este, a diferencia de otras medidas de razonamiento abstracto, proporciona puntuaciones objetivas del total de acierto y además, las correspondientes a otras fuentes.

El WCST se ha empleado cada vez en mayor escala como instrumento clínico neurológico. El test tiene sus raíces en anteriores investigaciones sobre la abstracción y los efectos de las lesiones en el lóbulo frontal, en el mantenimiento de la actitud cognitiva, realizadas con personas y animales.

El objetivo final de esta aplicación es el desarrollo del propio test, para que pueda ser resuelto por los pacientes de forma interactiva, ahorrando trabajo al profesional médico, eliminando errores de corrección, así como el tratamiento y almacenamiento de los resultados y de los propios diagnósticos médicos.

### **1.2.2. Objetivos específicos: Resumen**

- Es una aplicación desarrollada en lenguaje de programación Java, que mantiene una conexión a base de datos mediante MySQL. La aplicación contiene un menú principal como los submenús correspondientes para la interacción del paciente con el PC así como el del profesional médico con el PC.
- Se implementará el Test Winconsin con una interfaz gráfica que captará la selección del paciente ante dicho test, para posteriormente ser almacenadas, tratadas y dar un diagnóstico clínico.
- El programa será capaz de captar las respuestas ante los diferentes estímulos de los pacientes, cálculo de resultados de test así como almacenarlos para tener un historial del paciente.
- Se trata de diseñar un programa que intente ser modular, con vista a posibles futuras mejoras y/o ampliaciones
- Se pretende diseñar una herramienta médica de apoyo, que permitirá ayudar al profesional médico (psicólogo-psiquiatra) a realizar su tarea en la búsqueda de un diagnóstico adecuado y su almacenamiento para futuras comparaciones y tratamientos de información.
- Se trata de crear un entorno atractivo para aquellas personas que ponerse frente a un médico pueda variar su estado de concentración y de esta forma que puedan realizar la prueba en un estado más relajado.
- Para los niños se enfoca en la simulación de un juego, mejorando la predisposición de estos a su ejecución.

### **1.2.3. Justificación**

En los últimos años la medicina ha evolucionado de forma exponencial, esta gran evolución no hubiera sido posible sin el apoyo y el desarrollo paralelo de la ingeniería.

Gracias a esta asociación medicina-ingeniería, es posible desarrollar herramientas que ayuden al profesional médico en su labor. Hoy en día existe maquinaria médica tanto para el diagnóstico, como para llevar a cabo operaciones quirúrgicas invasivas.

La labor de un profesional médico en la búsqueda de un diagnóstico lo más acertado posible es de gran complejidad, sobre todo en campos relacionados con el cerebro o la mente.

Este proyecto pretende ser un apoyo para el médico en su diagnóstico, ahorrándole tiempo e incluso facilitando su trabajo para que este pueda centrarse en otros puntos.

**1.2.4. Fases del proyecto**

- Preparación de la información necesaria relativa al test.
- Creación de la interface gráfica para la interacción del paciente frente al test.
- Desarrollo del código y algoritmos necesarios capaces de tomar como entradas las respuestas del paciente ante el test, tratarlas y dar un diagnóstico objetivo a partir de dicha información.
- Desarrollo de una interface completa, compuesta por los menús necesarios de acceso a los datos de los pacientes (históricos de resultados sobre la prueba).
- Conexión a base de datos de la aplicación para el almacenamiento eficiente de datos, resultados e históricos del paciente.
- Conexión de todas las partes en un programa herramienta.
- Comprobación de los resultados mediante la realización de un test ejemplo.

## ***CAPÍTULO 2***

### **FUNDAMENTOS Y ADAPTACIÓN AL TEST**

---

---

En este capítulo se describe todo lo relacionado al test Wisconsin, como la funcionalidad, los elementos que contiene, la población a la que se destina, los materiales que se usan, los fines perseguidos, así como los requisitos profesionales necesarios para su correcta aplicación. Además se apuntan adaptaciones y modificaciones que se van a realizar respecto al test original.

Estos puntos se deben tratar y comprender para una correcta adaptación a la simulación informática que se verá en los siguientes capítulos. Dicha adaptación es un software que permite tanto el desarrollo del test de forma interactiva como las interfaces y formularios necesarios para el almacenamiento y posterior tratamiento específico de los resultados.

Para almacenar los datos, será necesario vincular la aplicación con una base de datos la cual contendrá todos los registros de los test realizados por los correspondientes individuos.

#### **2.1 DESCRIPCIÓN DEL TEST DE CLASIFICACIÓN DE TARJETAS WISCONSIN.**

Test desarrollado originalmente para evaluar el razonamiento abstracto y la habilidad para cambiar las estrategias cognitivas como respuesta a eventuales modificaciones ambientales. Puede ser considerado como una medida de la función ejecutiva que requiere habilidad para desarrollar y mantener las estrategias de solución de problemas que resultan adecuadas para conseguir unos objetivos, a través de condiciones que implican cambios de estímulos

El WCST requiere estrategias de planificación, indagaciones organizadas, utilizando el “feedback ambiental” para cambiar esquemas, orientación de la conducta hacia el logro de un objetivo y modulación de las respuestas impulsivas. Este a diferencia de otras medidas de razonamiento abstracto, proporciona puntuaciones objetivas del total de aciertos y además las correspondientes a otras fuentes específicas de dificultad en la tarea. (por ejemplo, conceptualización inicial deficiente, fallos para mantener la actitud o el curso cognitivo, perseveración y deficiencia de aprendizaje a lo largo de las etapas del test) .

Durante su desarrollo se utilizó como medida del razonamiento abstracto en poblaciones de adultos normales, El WCST se ha empleado cada vez en mayor escala como instrumento clínico neurológico. El test tiene sus raíces en anteriores investigaciones sobre la abstracción y los efectos de las lesiones en el lóbulo frontal, en el mantenimiento de la actitud cognitiva, realizadas con personas y animales.

Gran parte de la popularidad alcanzada entre los clínicos radica en su sensibilidad para proporcionar información sobre disfunciones cerebrales que afecten a los lóbulos frontales. El interés respecto a los efectos que produce el daño temprano del lóbulo frontal, en aspectos cognitivos y de desarrollo, en los niños, hizo surgir también un considerable interés hacia la utilización de WCST como medida potencial de la función ejecutiva en la edad escolar.

A causa de su aparente sensibilidad a los efectos de las lesiones en el lóbulo frontal, El WCST se menciona frecuentemente como una medida del funcionamiento frontal o pre-frontal. Pero esto representa una excesiva simplificación. Si bien se considera que algunas de las dimensiones cognitivas que aprecia el test son particularmente vulnerables las condiciones neurológicas que afectan a las zonas frontales del cerebro, cualquier irregularidad médica o psicológica que desorganice las funciones ejecutivas, en todo o en parte, puede dar lugar como resultado un deterioro en la realización del test. Por esta razón, la interpretación clínica de la realización de un sujeto en el test debe ser considerada en el contexto de una evaluación neuropsicológica integrada por datos neuropsicológicos, datos médicos, información psicosocial e historia del sujeto.

Básicamente, el test consiste en 4 tarjetas modelo y 128 tarjetas de prueba (divididas en dos grupos de 64 tarjetas), que contienen dibujos que varían en la forma (cruces, círculos, triángulos o estrellas), color (rojo, azul, amarillo o verde), y número de figuras (uno, dos, tres o cuatro). El examinador coloca las 4 tarjetas modelo delante del paciente y le entrega un grupo de 64 tarjetas. A continuación, se le pide que clasifique cada una de las tarjetas según el criterio que quiera el propio paciente. El examinador le dirá si la respuesta es correcta o no. El orden de clasificación de las categorías es invariable, color, forma y número. El examinador debe marcar en la hoja de respuestas las categorías (color, forma y número) correctas de clasificación. Así por ejemplo, teniendo en cuenta que el criterio inicial es el color, si el paciente coloca su primera tarjeta (v.g. una estrella de color azul) sobre la tarjeta modelo “una estrella de color amarillo”, el examinador tachará la categoría “forma” y “número”, sin embargo la respuesta es “incorrecto” para el paciente. En el siguiente ensayo, el paciente puede optar por seguir clasificando las tarjetas en relación con algunas de las categorías anteriores o cambiar de criterio y seleccionar el “color”. En este caso la respuesta sería “correcto”. Cuando el sujeto consigue 10 respuestas correctas se cambia de criterio sin previo aviso.

Una ejecución adecuada implica que el sujeto determine en primer lugar el principio de clasificación correcto en función de la retroalimentación proporcionada por el examinador, y mantener después este principio de clasificación o “set” (color), a través de los estímulos que cambian e ignorar las demás dimensiones irrelevantes (forma y número). Por el contrario, se considera un fracaso o una incapacidad para mantener el criterio cuando el paciente comete 5 o más respuestas correctas consecutivas y después comete 1 error antes de conseguir con éxito el criterio (10 respuestas correctas consecutivas). Ahora bien, cuando el paciente clasifica 10

respuestas correctas dentro de una categoría, y el examinador cambia el criterio, el sujeto debe inhibir la tendencia a persistir o perseverar el criterio anterior. Se registra el número de respuestas correctas, el número de categorías completadas, el número de errores, y el de perseveraciones en el error.

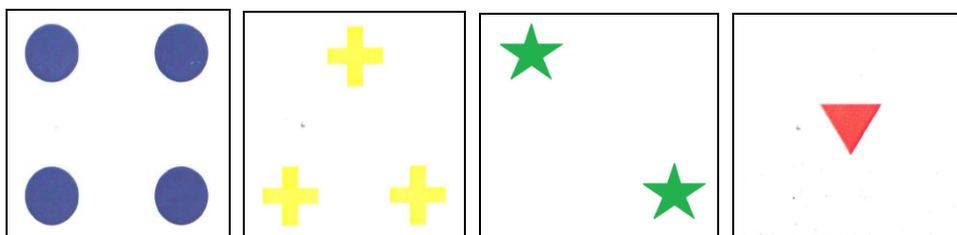
### 2.1.1 Datos del Test

- 1- Administración: Individual
- 2- Duración: Variable
- 3- Aplicación: Niños, adolescentes y adultos con edades comprendidas entre 6,5 años y 89 años.
- 4- Significación: Medida de la función ejecutiva que requiere estrategias de planificación, indagaciones organizadas y utilización del feedback ambiental para cambiar esquemas. Dada su posible sensibilidad a los efectos de las lesiones en el lóbulo frontal, se menciona frecuentemente como una medida del funcionamiento frontal y prefrontal.
- 5- Baremación: Puntuaciones directas y típicas por edades para cada una de las variables de la prueba.

### 2.1.2 Materiales del Test “Clásico”

- Manual
- 4 Tarjetas estímulo
- Dos conjuntos de bloques de tarjetas-respuesta con 64 tarjetas cada uno.
- Hoja de anotación y resultados

En las cuatro tarjetas-estímulo se presentan figuras que constituyen los estímulos del test: un triángulo rojo, dos estrellas verdes, tres cruces amarillas y cuatro círculos azules. En ellas se representan tres y sólo tres parámetros: Color, Forma y Número. Estas tarjetas se pueden ver a continuación en la figura 2.1



*Figura 2.1: Tarjetas-estímulo.*

Cada conjunto o bloque de tarjetas-respuesta contienen 64 tarjetas en las que aparecen figuras de diversas formas (cruces, círculos, triángulos o estrellas), colores

(rojo, azul, amarillo o verde), el número de figuras (una, dos, tres o cuatro). Cada tarjeta-respuesta se puede emparejar con una tarjeta estímulo atendiendo a una de las mencionadas características o a una combinación de las tres.

Las tarjetas-respuesta de cada bloque están numeradas del 1 al 64, en el ángulo inferior izquierdo del reverso de la tarjeta, para asegurar el mismo orden de presentación en todos los casos. Siguiendo este orden se garantiza que en la presentación no aparezcan tarjetas consecutivas con el mismo color, forma o número de elementos.

Antes de comenzar la aplicación, deben asegurarse de que las tarjetas-respuesta están colocadas de forma que la figura que queda en la parte superior del bloque se corresponde con el número.

La hoja de anotación se ha diseñado de modo que permita anotar la información necesaria sobre el sujeto y las respuestas de éste a todos los ítems de la prueba; también incluye espacios para calcular y registrar las puntuaciones obtenidas en el test.

La primera página de la Hoja de anotación contiene espacios para registrar los datos demográficos, la fecha del examen, el nombre del sujeto y también algunas informaciones complementarias, por ejemplo, observaciones sobre la conducta durante la aplicación y juicios sobre la realización en algunos aspectos del test o en otras pruebas.

La segunda página se utiliza para anotar las respuestas del sujeto y las dimensiones utilizadas en los que se anotará el cálculo de las puntuaciones del WCST. En la parte superior de la segunda página de la Hoja de anotaciones aparece la sucesión de categorías: C, F, N, C, F, N, que corresponde a la secuencia de seis categorías de clasificación correctas. Debajo de esta secuencia aparecen los espacios donde se registran las respuestas del sujeto a los 128 ítems (cuatro columnas de 32 ítems cada una).

Las dos primeras columnas contienen los espacios numerados del 1 al 64 en los que se anotarán las respuestas a los elementos correspondientes al segundo conjunto de tarjetas. Lo mismo sucede con las dos últimas en las que se anotarán las respuestas correspondientes al segundo conjunto de tarjetas. En ambos casos la numeración se corresponde con el número impreso en el reverso de las tarjetas en cada uno de los dos bloques. Para cada ítem, se presentan como opciones de respuesta las letras C, F, N, O. A la izquierda de cada ítem aparece un espacio para el que el examinador anote el número de respuestas consecutivas (hasta 10) en cada categoría. A la derecha de cada ítem hay un lugar para indicar algunas dimensiones de la puntuación, que serán registrados después de la aplicación.

La tercera página de la Hoja de anotación contiene espacios para anotar las puntuaciones directas y transformadas obtenidas por el sujeto en la prueba.

### **2.1.3 Descripción General**

Mediante los cuatro puntos siguientes se puede hacer una breve descripción de la forma de aplicación del test.

1. La aplicación consiste en colocar las cuatro tarjetas-estímulo ante el sujeto ordenándolas de izquierda a derecha de la forma: 1º Un triángulo rojo, 2º Dos estrellas verdes, 3º Tres cruces amarillas, 4º cuatro círculos azules.
2. Se entrega al sujeto un bloque de 64 tarjetas y se le indica que tiene que emparejar cada una de las tarjetas de ese bloque con una de las tarjetas-estímulo que se han colocado ante él y que puede emparejarlas de cualquier modo que le parezca posible. Cada vez que el sujeto coloque una tarjeta se le indicará si lo ha hecho de forma correcta o errónea, pero no se le dice nunca cuál es la categoría que se tiene en cuenta para clasificar.
3. Cuando el sujeto ha logrado un número determinado de respuestas correctas consecutivas emparejando las tarjetas con arreglo a la categoría de clasificación establecida inicialmente (Color), se cambia dicha categoría de clasificación pasando a Forma o Número, sin avisar, esto requiere que el sujeto utilice el “feedback” que recibe del examinador para establecer una nueva estrategia de clasificación.
4. La aplicación avanza de esta forma cambiando las categorías establecidas como criterios de clasificación, seleccionando un nuevo criterio de clasificación entre los posibles.

### **2.1.4 Requisitos profesionales**

Aunque cualquier persona con experiencia en la aplicación de pruebas puede llevar a cabo la aplicación del WCST, los procedimientos de administración, registro de respuestas y puntuación que se detallan en apartados siguientes deben ser cuidadosamente estudiados y ensayados antes de utilizar la prueba en una situación clínica. El entrenamiento y supervisión en estos procedimientos debe estar a cargo de un psicólogo cualificado para ello.

La interpretación clínica del WCST debe ser realizada por un psicólogo especializado en psicología clínica o neuropsicología. La utilidad y la validez de este test como medida clínica de la habilidad cognitiva están relacionadas directamente con la formación y los conocimientos del profesional que lo utilice, así como con el conocimiento de la información que contiene este manual. La interpretación de las puntuaciones del WCST con fines clínicos o de diagnóstico no deben abordarse si no se posee una clara comprensión de la relación conducta-cerebro y de los factores médicos y psicológicos que la afectan.

## 2.2 APLICACIÓN Y PUNTUACIÓN

El test se utiliza en variedad de situaciones, tanto con niños como con adultos, dado que se trata de una prueba de aplicación individual, el WCST se puede utilizar como instrumento clínico o de investigación combinándolo con pruebas que evalúen otros aspectos del funcionamiento cognitivo y neuropsicológico.

Las normas de aplicación y puntuación se presentan en los siguientes apartados de este capítulo, estos serán sistematizados mediante algoritmos en la aplicación informática ahorrando tiempo al profesional que lo desarrolla, así como evitando errores de valoración.

Es necesario que la aplicación se realice en un lugar tranquilo, con iluminación adecuada para que el sujeto pueda percibir correctamente los estímulos del test. En la aplicación tradicional se necesitarían una mesa y dos sillas, una para el sujeto y otra para el examinador, en la aplicación informática es solo el paciente el que se sitúa delante del PC.

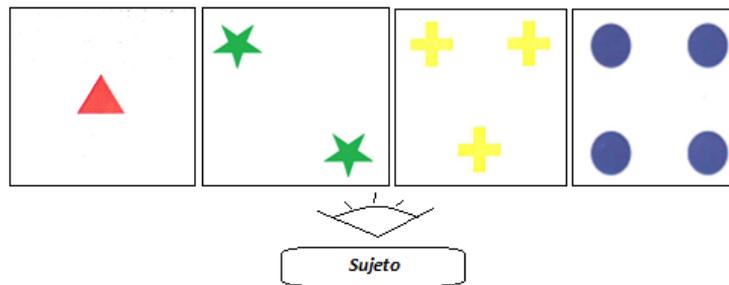
En el proceso tradicional de aplicación el examinador deberá anotar las respuestas en la hoja de anotación y posteriormente tratarlas para llegar al resultado o diagnóstico en este caso. La aplicación informatizada recibe mediante estímulos captados por los periféricos (ratón, teclado del ordenador) la información a anotar. Con los datos recibidos realiza de forma instantánea y sin ningún tipo de error los resultados requeridos por el profesional para realizar y evaluar un diagnóstico.

En la ejecución de la aplicación desaparecen las comprobaciones iniciales por parte del examinador, como es la correcta colocación de las cartas estímulo, así como su orientación y la correcta secuencia de las figuras del mazo.

### **2.2.1 Instrucciones de preparación y aplicación antes del test.**

Antes de empezar con la aplicación del test al sujeto, el examinador debe registrar los datos o informaciones de carácter médico o emocional que puedan afectar a la realización del test. A continuación se pedirá al sujeto que se siente frente al a él o frente al PC en la aplicación informática. Durante la ejecución del programa del test el examinador puede estar o no presente.. En caso que esté presente la aplicación servirá para evitar fallos de anotación y que este se pueda centrar de una forma más específica en observar conducta del sujeto, así como sus emociones, aunque el test está diseñado para que durante esta parte no sea necesaria la presencia del examinador.

El examinador debe tradicionalmente debe tomar los dos bloque de tarjetas dejando la caja a un lado y luego abrir la Hoja de anotación, sin permitir que el sujeto vea esa página. Las tarjetas-estímulo se colocarán sobre la mesa en fila, frente al sujeto, asegurándose de que entre ellas y el límite de la mesa queda espacio suficiente para colocar adecuadamente las tarjetas-respuesta. En la figura 2.2 se muestra la forma en que se deben colocar las tarjetas-estímulo frente al sujeto, comenzando por el lado izquierdo de éste y siguiendo el orden establecido: primero el triángulo rojo (que quedará a la izquierda del sujeto) luego las dos estrellas verdes, las tres cruces amarillas y los cuatro círculos azules



**Figura 2.2:** Forma correcta de colocar las Tarjetas-estímulo frente al sujeto.

Al inicio de la aplicación de forma tradicional el examinador debe trasladar al sujeto las indicaciones sobre el desarrollo del test, es decir el examinador narrará lo expuesto en la Figura 2.3.

*“ Este test es poco usual porque yo no voy a dar mucha información sobre lo que hay que hacer: la tarea consiste emparejar estas tarjetas (señalando los bloque de tarjetas-respuestas) con estas cuatro que he colocado aquí y que sirven de clave (señalar las tarjetas-estímulo sucesivamente, empezando por el triángulo rojo) hay que tomar siempre una tarjeta de este bloque, la de arriba, (señalar) y colocarla debajo de la tarjeta clave con la que se crea que hace pareja (señalar la fila de tarjetas estímulo). Yo no puedo decirle cómo se deben emparejar las tarjetas, pero cada vez que Vd, coloque una; le diré si lo ha hecho correctamente o si se ha equivocado. En los casos en que la haya colocado mal, simplemente dejará la tarjeta donde la haya puesto e intentará colocar la siguiente correctamente. Para esta prueba no hay tiempo límite, ¿está Vd. preparado? Puede comenzar”*

**Figura 2.3:** Texto narrado por el examinador.

En la aplicación informática, el texto anterior debe ser modificado según figura 2.4, ya que ahora la interacción es directamente con el PC, y es este el que debe transmitir al sujeto las indicaciones necesarias sobre la realización del mismo.

*“ Este test es poco usual porque yo no voy a dar mucha información sobre lo que hay que hacer: La tarea consiste en emparejar la tarjeta de abajo con una de las cuatro que hay colocada arriba y que sirven de clave, cada vez que se seleccione una de las tarjetas de arriba que se empareja directamente con la de abajo y a continuación aparece una nueva tarjeta en la parte inferior para emparejarla de nuevo con una de la parte superior. No puedo decir cómo se deben emparejar las tarjetas, pero cada vez que Vd, coloque una; se indicará en pantalla y mediante un sonido si lo ha hecho correctamente o si se ha equivocado. Si es correcta aparecerá una V verde acompañada del siguiente sonido (sonido correcta), y si es incorrecta aparecerá una X en rojo y el sonido (sonido incorrecta). Para esta prueba no hay tiempo límite, ¿está Vd. preparado? ”*

**Figura 2.4:** Texto narrado por el PC.

A causa de la naturaleza ambigua del test los sujetos pueden formular diversas preguntas al examinador, pero es adecuada la aplicación informática ya que no habría que hacer ninguna aclaración debido a que no se puede vulnerar la integridad del WCST dando indicaciones sobre las categorías de clasificación o sobre la naturaleza de los cambios entre una categoría y la siguiente. Generalmente, la repetición de las instrucciones es suficiente para la mayoría de los sujetos.

En las aplicaciones realizadas por los autores del test se probó que, incluso los niños más pequeños tenían pocas dificultades para seguir las instrucciones normales del WCST. Sin embargo, con algunos de los más pequeños, se observó que la prueba resultaba menos amenazante si se presenta como un juego y no como un test. En este caso de aplicación informática por su propia naturaleza no se presenta este problema, incluso es posible que el niño se sienta en mejor disposición a su realización al ser de forma informatizada y visualmente más atractiva.

### **2.2.2 Instrucciones de aplicación durante el test**

- La primera categoría correcta de clasificación es **Color**. Cuando el sujeto comience a colocar las tarjetas-respuesta, a medida que las vaya poniendo o en este caso seleccionando los estímulos, se indicará mediante un sonido y una animación que se asocian a correcto o bien cada vez que las clasifique atendiendo a la característica color y se le indicará incorrecto o mal mediante otra animación y otro sonido cada vez que empareje una tarjeta atendiendo a otra categoría que no sea Color. La aplicación continúa de este modo hasta que el sujeto haya colocado correctamente diez tarjetas consecutivas, atendiendo a la característica Color.
- Sin dar ninguna otra indicación, se cambiará en ese momento la categoría correcta que a partir de entonces será Forma; es decir, si el sujeto coloca la tarjeta siguiente atendiendo a Color, que hasta entonces era lo correcto, se indicará que no es correcto.
- Forma se mantiene como categoría correcta hasta que el sujeto haya alcanzado diez respuestas correctas consecutivas; cuando esto sucede, sin aviso ni indicación de lo que está sucediendo, se cambiará la categoría de clasificación a Número.
- Cuando el sujeto haya dado diez respuestas correctas consecutivas en la categoría Número se volverá a establecer de nuevo Color como categoría correcta de clasificación, después Forma y después Número, actuando siempre del modo descrito anteriormente.
- La prueba continúa hasta que el sujeto haya completado con éxito seis categorías o hasta que haya utilizado todas las tarjetas, en el caso de que esto ocurra antes de completar las categorías indicadas.

*“No se proporcionará al sujeto ninguna información distinta de la incluida anteriormente en ningún momento, ni durante el desarrollo de la misma, ni después de finalizarla.”*

El WCST no es una prueba con tiempo limitado y en las instrucciones iniciales se informa de ello al sujeto. Aunque el tiempo empleado en las aplicaciones realizadas es bastante variable, la mayoría de las veces se ha completado en un período de 20 o 30 minutos. Si la realización fuese mediante cartas podría suceder que el sujeto comience a colocar las tarjetas muy deprisa dificultando la anotación de las respuestas; en estos casos el examinador pedirá que lo haga más despacio, y deberá dejar constancia de ello en la Hoja de anotación. También puede hacer más lento el “feedback” con el sujeto (por ejemplo no diciendo si la respuesta es correcta o errónea hasta su anotación en la Hoja.

Los problemas anteriores que le surgen al examinador, mediante la aplicación informática se eliminan, y ahora el test puede ser realizado de una forma más precisa en cuanto a la captación de los datos o respuestas de sujeto e indicaciones hacia el paciente, así como podrá ser más rápida su ejecución.

### **2.2.3 Anotación tradicional de las respuestas**

Es importante una correcta captación de las respuestas del sujeto. A medida que el sujeto responda a cada ítem, se marcará con una barra (/) cada categoría que coincida en las dos tarjetas (estímulo - respuesta). Si la tarjeta-respuesta y la tarjeta-estímulo se emparejan solamente en una dimensión, la hoja de anotación se pondrá la barra solamente sobre el símbolo que corresponda a esa dimensión. Por ejemplo si la tarjeta-respuesta se empareja con la tarjeta-estímulo solamente en cuanto a Color, en la hoja de anotación se marcará este ítem poniendo la barra solamente sobre la C (C,F,N,O).

Si la tarjeta-respuesta se empareja con la tarjeta estímulo en dos dimensiones, se marcarán con una barra los símbolos correspondientes a estas dos dimensiones. Por ejemplo si se emparejan en Color y Número se colocará sobre estas dos dimensiones una “/”.

Finalmente, si la respuesta no se empareja con la tarjeta estímulo en ninguna dimensión se marcará en la hoja de anotación la opción Otros (C,F,N,Ø). Las respuestas se anotarán de esta forma en cada ítem, con independencia de que sean correctas o incorrectas respecto a la categoría de clasificación que esté considerando.

En la parte superior de la hoja de anotación se presenta la secuencia correcta en que se han de completar las categorías de clasificación (C,F,N,C,F,N). Al comenzar la aplicación se escribe la letra “C” en la parte izquierda de la columna de respuestas, frente al ítem 1, para indicar la primera categoría correcta de clasificación (Color). Durante la aplicación se marcará en la parte superior de la página la letra correspondiente a cada categoría que se haya completado correctamente ( C , F , N , C , F , N ). Para determinar cuándo un sujeto ha completado con éxito una categoría y como ayuda para el posterior cálculo de las puntuaciones, se numerarán, correlativamente las respuestas correctas (de 1 a 10) anotando los números en el espacio destinado a este fin

en la hoja de anotación, a la izquierda de cada ítem; esto puede hacerse a medida que el sujeto va colocando las tarjetas.

No hay que olvidar que solamente se deben numerar las respuestas correctas. Si el sujeto interrumpe la secuencia con un error; cuando vuelva a dar una respuesta correcta se volverá a comenzar la numeración con el número 1, cuando se alcance el criterio de diez respuestas consecutivas, se trazará una línea horizontal por debajo del último ítem contestado, de forma que en la parte superior de la página la letra correspondiente a la categorías superada y se escribirá el símbolo de la próxima categoría de clasificación en la línea horizontal en el lado izquierdo de la columna. Este proceso se muestra gráficamente en la siguiente figura. 1.5

## 2.3 DIMENSIONES VALORADAS Y EVALUACIÓN DEL TEST

La puntuación del WCST ha sido una fuente de dificultades para muchos profesionales. En tanto que algunos investigadores han informado de una alta fiabilidad interpuntuaciones (Axelrod, Goldman y Wodard, 1992; Huettner; Wolfe y Hyud, 1989) otros (Flashman, Madir, Horner y Frids, 1991), han encontrado baja fiabilidad en las puntuaciones a causa de la variada o incorrecta aplicación de las normas de puntuación diseñadas originalmente por Heaton (1981). En esta sección se clarifican las fuentes de dificultades más comunes, respecto a la puntuación y se incluyen algunas sugerencias útiles de Flashman y Axelrod que mantienen las reglas de puntuación originales de Heaton.

### 2.3.1 Dimensiones que se valoran

Cada respuesta del sujeto debe ser considerada como si ocurriese en tres dimensiones separadas y, por tanto, evaluada en cada una de ellas. Estas dimensiones son:

**1. Correcto-incorrecto:** el examinador (Pc) proporciona al sujeto “feedback” en cada respuesta, tanto si es correcta como si es incorrecta, esto depende de que empareja las tarjetas considerando la categoría establecida, cada vez que da una respuesta. La categoría de clasificación establecida se puntúan como correctas y las que no corresponden, como erróneas.

De forma tradicional el examinador deberá numerar correlativamente en la Hoja de anotación las respuestas correctas dadas a ítems consecutivos durante la aplicación. Si el sujeto da una o varias respuesta erróneas, se comienza de nuevo la numeración cuando vuelva a dar respuestas correctas; Una vez finalizada la aplicación, se rodearán los ítems que no se hayan numerado. Los ítems rodeados representan las respuestas incorrectas, es decir, los errores. Estos procedimientos de anotación se muestran en la Figura 1.5:

		CFNCFN				
<b>C</b>	1	C	F	N	O	Los errores aparecen rodeados.
	2	C	F	N	O	
	3	C	F	N	O	
	4	C	F	N	O	
	5	C	F	N	O	
	6	C	F	N	O	
	7	C	F	N	O	
	8	C	F	N	O	
	9	C	F	N	O	
	10	C	F	N	O	
<b>F</b>	1	C	F	N	O	Cambio de Categoría
	13	C	F	N	O	
	14	C	F	N	O	
	15	C	F	N	O	
	1	C	F	N	O	
	17	C	F	N	O	
	18	C	F	N	O	
	1	C	F	N	O	
	19	C	F	N	O	
	20	C	F	N	O	
20		C	F	N	O	

Figura 2.5: Puntuación en categoría correctas- erróneas

**2. Ambigua-no ambigua.** Cuando una tarjeta-respuesta se empareja con una tarjeta-estímulo atendiendo a una, y sólo una de sus características (por ejemplo, Color), para el examinador resulta obvia, no ambigua, la categoría utilizada por el sujeto para realizar el emparejamiento. Esto sucede si, por ejemplo, una tarjeta-respuesta con tres círculos rojo se ha emparejado con la tarjeta-estímulo que tiene un triángulo rojo. Sin embargo las tarjetas pueden emparejarse en más de una dimensión. Un ejemplo de esto ocurre cuando una tarjeta-respuesta con tres triángulos rojos se empareja con el estímulo que sólo tiene un triángulo rojo. En esta situación, la respuesta es ambigua para el examinador pues no puede estar seguro de que el sujeto haya clasificado esa tarjeta considerando la categoría Color o la categoría Forma.

Por tanto, cuando la tarjeta-respuesta se empareja con la tarjeta estímulo en una, y sólo una, característica se considera una respuesta no ambigua y cuando la tarjeta respuesta se empareja con la tarjeta-estímulo en dos o más características se considerará ambigua. Por definición la categoría otras contienen siempre respuestas ambiguas. La dimensión ambigua-no ambigua es independiente de que la respuesta sea correcta o errónea.

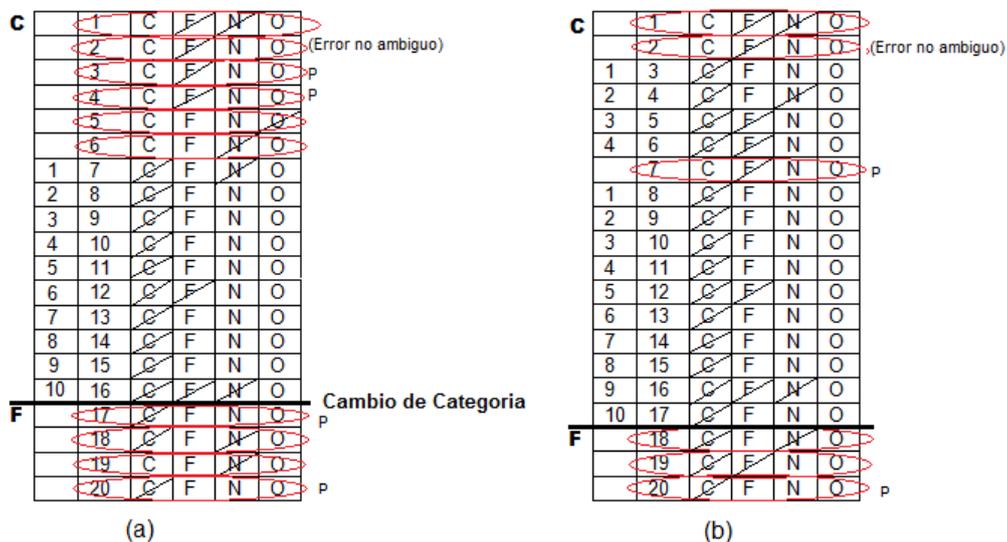
**3. Perseverativa-no perseverativa:** Cuando el sujeto persiste en responder una característica del estímulo que no es correcta, la respuesta se considera una perseveración hacia ese criterio y se puntúa como “perseverativa”. El sujeto puede perseverar en Color, Forma o Número. Sin embargo, no es posible que esto suceda en la categoría Otras porque el examinador no puede estar seguro del criterio de clasificación seguido por el sujeto. El principio de perseveración es operativo una vez establecido. De hecho las respuestas que se emparejen con el principio de perseveración establecido se consideran perseverativas, prescindiendo de que sean correctas o incorrectas. Las respuestas que no se emparejen siguiendo el principio de perseveración se consideran no perseverativas. Las respuestas de forma tradicional se señalarían en la Hoja de anotación, después de terminar la prueba poniendo “p” en la derecha del ítem a que se refieren.

“Hay tres situaciones distintas que delimitan el principio de perseveración en la puntuación de las respuestas perseverativas:

**3.1. El principio de perseveración se establece al comienzo de la prueba,** la primera vez que el sujeto comete un error no ambiguo. Pero el primer error no-ambiguo solamente establece el principio de perseveración y no se puntúa como perseverativo. Cualquier error no ambiguo que aparezca después y que sea igual a este principio de perseveración, establecido en el primer error no ambiguo, se puntúa como perseverativo, incluso si se incluyen en la secuencia otras respuestas que no se igualen a este principio perseverativo.

De este modo, puede suceder que el sujeto dé respuestas perseverativas antes de completar con éxito una categoría. La figura 2.6.a muestra el caso de puntuación más sencillo: En el segundo intento el sujeto ha emparejado la tarjeta que tiene cuatro cruces rojas con la tarjeta estímulo que tienen tres cruces amarillas. Esta respuesta es un error no ambiguo que se sitúa el principio de clasificación en la categoría Forma (C, F, N, O) y la establece como principio perseverativo. Cada vez que el sujeto cometa un error no ambiguo en la categoría Forma, por ejemplo en los intentos 3º y 4º de la Figura X, la respuesta se puntuará como perseverativa hasta que cambie el principio de perseveración.

En la figura 2.6.b se muestra esta regla de puntuación cuando intervienen en la secuencia otras respuestas que no corresponden al principio de perseveración establecido. El sujeto de este ejemplo ha cometido un error no ambiguo respecto a la categoría Forma en el intento 2, luego ha clasificado correctamente las cuatro tarjetas siguientes atendiendo a Color y después ha dado la primera respuesta perseverativa a Forma en el intento 7.



**Figuras 2.6.a y 2.6.b:** Casos de puntuación en categoría de perseveración.(Situación 1)

**3.2. El momento en el que se producen normalmente respuestas perseverativas es después de haber completado una categoría con diez respuestas correctas consecutivas.** Como el sujeto no sabe que el examinador ha cambiado la categoría de clasificación, es muy probable que continúe respondiendo de acuerdo con la regla que

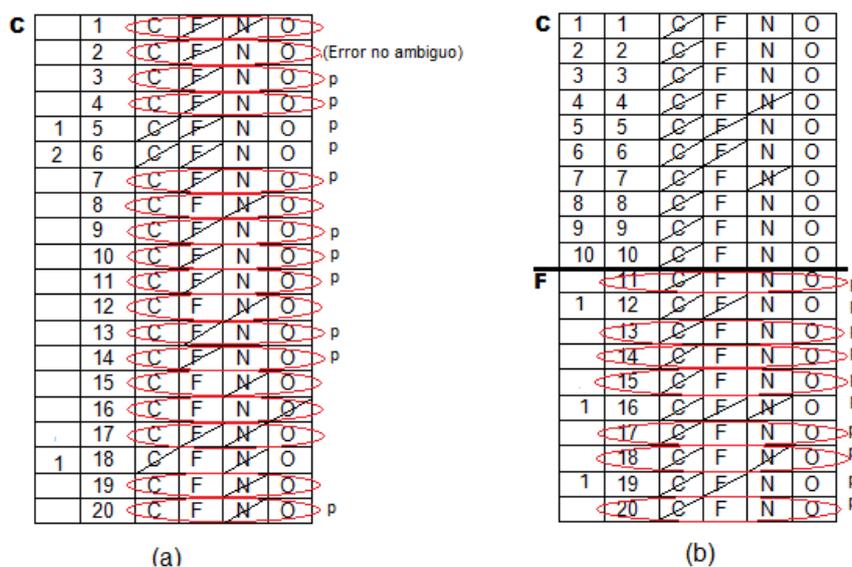
en los ítems anteriores era correcta. Pero, según las reglas de puntuación, después de haber completado con éxito una categoría, cuando el examinador cambia el criterio de clasificación, la categoría que antes era correcta se convierte en principio de perseveración. Algunos investigadores consideran que esta regla es contraintuitiva.

La primera respuesta no ambigua que constituye el nuevo principio de perseveración (el sujeto sigue clasificando las tarjetas atendiendo a la categoría correcta en los elementos anteriores) es un error perseverativo no ambiguo. Por ejemplo podemos ver esta regla en el intento 17 en la figura 2.7.a y en el intento 11 en la figura 2.7.b que se mostrará a continuación; en ambos casos se trata de errores perseverativos no ambiguos porque las respuestas Color siguen el principio de perseveración vigente hasta ese momento (la categoría que se acaba de completar).

Como se muestra en el intento 12 de la figura 2.7.b una respuesta ambigua, podrá ser puntuada también como perseverativa cuando reúne las condiciones a y b, citadas en los párrafos siguientes, que aseguran que forma parte de un patrón coherente de respuestas perseverativas.

Algunas respuestas ambiguas pueden ser perseverativas si cumplen las condiciones a, b y también c, que se citan a continuación. Estas condiciones se deben tener en cuenta tanto si la respuesta ambigua es correcta como si es incorrecta.

- a) La respuesta ambigua debe cumplir el principio de perseveración que está vigente
- b) Las respuestas no ambiguas más próximas a la respuesta ambigua (intento 12 figura 2.7.b), tanto la correspondiente al intento precedente (intento 11) como la correspondiente al siguiente (intento 13) deben ser perseverativas y deben seguir el principio de perseveración establecido. Esta se regla se conoce con el nombre de “regla de bocadillo”, las respuestas ambiguas deben estar situadas entre dos respuestas perseverativas no ambiguas.



Figuras 2.7.a y 2.7.b: Casos de puntuación en categoría de perseveración. (Situación 2)

- c) Para que una respuesta ambigua se considere perseverativa es preciso que todas las respuestas comprendidas entre dos no ambiguas sigan el principio de perseveración vigente. Esto es, el principio debe seguir sin interrupción en todas las respuestas comprendidas entre dos perseverativas no ambiguas; la secuencia no debe estar interrumpida por ninguna respuesta que no siga este principio. En la Figura 2.7.a los intentos 5 y 6, se han puntuado como respuestas perseverativas sin tener en cuenta si son respuestas correctas o errores. Los intentos 12, 16, 18 y 19 de la figura 2.7.a no se han puntuado como perseverativos porque no están precedidos por una respuesta perseverativa no ambigua.

Como se muestra en los intentos 5 y 6 de 2.7.a las respuestas ambiguas pueden ser puntuadas como perseverativas, antes de que el sujeto haya completado la primera categorías, si reúnen las tres condiciones mencionadas anteriormente.

Primera, las respuesta o respuestas deben seguir la regla de perseveración vigente hasta entonces (en el ejemplo, el principio de perseveración es Forma, como se ha definido en el primer error no ambiguo del segundo intento).

Segunda, las respuesta o respuestas deben estar precedidas y seguidas (“en bocadillo”) por respuestas perseverativas no ambiguas (intentos 4 y 7). Finalmente, el principio de perseveración debe estar presente en seguirse, sin ser interrumpido, en todas las respuestas que aparezcan entre dos perseverativas no ambiguas.

**3.3.** El final de un principio de perseveración ocurre cuando el sujeto cambia su forma de respuesta perseverativa. Dentro de cualquier categoría el principio de perseveración cambia cuando:

➤ El sujeto comete tres errores no ambiguos con una categoría de clasificación, que en ese momento no es ni correcta ni perseverativa.

➤ Todas las respuestas comprendidas entre los errores no ambiguos primero y tercero siguen el principio de clasificación que se mantiene en tres errores no ambiguos. cuando esto ocurre, la categoría vigente en esos tres errores no ambiguos se convierte en el nuevo principio de perseveración. Sin embargo, este nuevo principio de perseveración no es operativa para propósitos de puntuación hasta que se produzca el segundo error no ambiguo. La figura 2.8.a muestra el ejemplo más sencillo de este caso de puntuación.

➤ La figura 2.8.a presenta las respuestas de un sujeto que clasificó la tarjeta atendiendo a Forma en el intento 2 y con ello estableció Forma como principio de perseveración al comienzo del test. Los emparejamientos no ambiguos con Forma en los intentos 4,7 y 8 se han puntuado como errores perseverativos. Sin embargo, comenzando en el intento 9, el sujeto hizo los tres emparejamientos siguientes con Número, por tanto los intentos 9,10 y 11 son errores ininterrumpidos no ambiguos respecto a otro principio que no es Forma. Número pasa a ser el nuevo principio de perseveración en el intento 10. Por

tanto, las respuestas dadas en los intentos 10 y 11 se puntúan como errores perseverativos. La Figura 2.8.b muestra estas reglas de puntuación para una larga serie de errores perseverativos donde el principio de perseveración ha pasado de ser Forma (intento2) a ser Número (intento 10).

Aparece una situación de puntuación más compleja cuando las respuestas ambiguas que siguen el nuevo principio de perseveración están intercaladas en una secuencia de tres errores no ambiguos. Siempre que las series no estén rotas por alguna respuesta que no se empareje, con el nuevo principio de perseveración, la segunda respuesta incorrecta no ambigua de la secuencia siquiera las respuestas ambiguas “en bocadillo” entre los errores no ambiguos primero y segundo deben seguir el nuevo principio de perseveración; este nuevo principio no es operativo hasta que se produzca un segundo error no ambiguo, las respuestas ambiguas no se puntúan como perseverativas.

c		1	C	F	N	O	
	2	C	F	N	O		(Error no ambiguo)
	3	C	F	N	O		
	4	C	F	N	O		p
	5	C	F	N	O		
1	6	C	F	N	O		
	7	C	F	N	O		p
	8	C	F	N	O		p
	9	C	F	N	O		(Error no ambiguo)
	10	C	F	N	O		p (Error no ambiguo)
	11	C	F	N	O		p (Error no ambiguo)
1	12	C	F	N	O		
	13	C	F	N	O		
	14	C	F	N	O		
	15	C	F	N	O		p
1	16	C	F	N	O		
	17	C	F	N	O		
1	18	C	F	N	O		
2	19	C	F	N	O		
3	20	C	F	N	O		

(a)

c		1	C	F	N	O	
	2	C	F	N	O		(Error no ambiguo)
1	3	C	F	N	O		
2	4	C	F	N	O		
3	5	C	F	N	O		
4	6	C	F	N	O		
5	7	C	F	N	O		
6	8	C	F	N	O		
7	9	C	F	N	O		
	10	C	F	N	O		(Error no ambiguo)
	11	C	F	N	O		p (Error no ambiguo)
	12	C	F	N	O		p (Error no ambiguo)
	13	C	F	N	O		p
	14	C	F	N	O		p
	15	C	F	N	O		
1	16	C	F	N	O		
	17	C	F	N	O		
1	18	C	F	N	O		
	19	C	F	N	O		p
	20	C	F	N	O		p

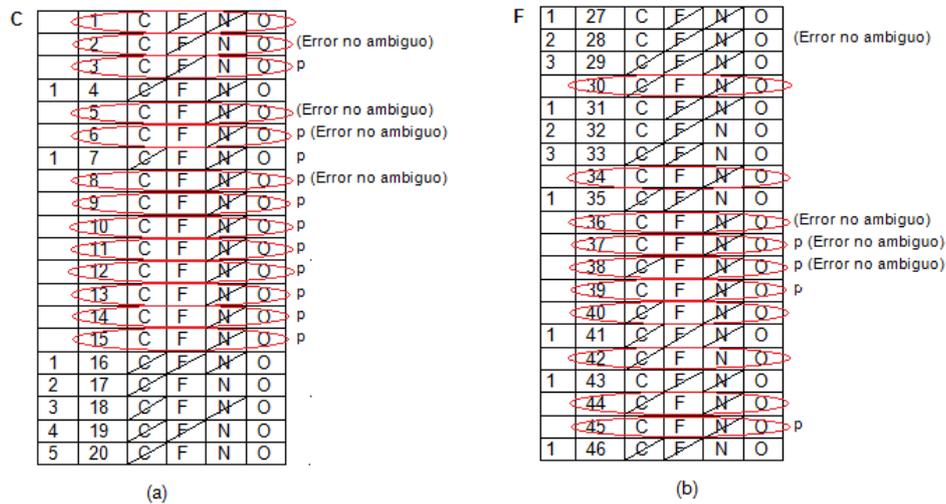
(b)

**Figuras 2.8.a y 2.8.b:** Casos de puntuación: Final del principio de perseveración.

Por ejemplo, en la figura 2.9.a el sujeto cometió el primer error no ambiguo en el intento 2 y por tanto se considera que el primer error perseverativo es la respuesta del intento 3. Sin embargo, el sujeto pasa después a clasificar las tarjetas atendiendo a la categoría Número y comete el primer error no ambiguo con Número en el intento 5. El intento 6 es también un error no ambiguo correspondiente a Número, lo mismo que el intento 8, que constituye el tercer error no ambiguo con la categoría Número. Dado que la respuesta ambigua del intento 7 sigue manteniendo la categoría Número, todas las respuestas de la secuencia desde el intento 5 al intento 8, corresponden a la categoría Número sin interrupción, definiendo Número como el nuevo principio de perseveración, en el intento 5.

Por consiguiente, comenzando en el segundo error no ambiguo de la secuencia (intento 6) se puntúan como perseverativas las respuestas 6 a 15. Se considera perseverativa la respuesta del intento 7, aunque es correcta, porque aparece precedida y seguida por errores perseverativos no ambiguos.

En el ejemplo de la Figura 2.9.b se muestra una situación similar. El error no ambiguo que aparece en la secuencia correspondiente a Número comienza en el intento 36. En este caso la respuesta ambigua del intento 38, que separa los errores no ambiguos segundo y tercero, es también errónea y, por tanto, aparece rodeada. El intento 40 no se ha puntuado como perseverativo por no seguir la “regla bocadillo”.



Figuras 2.9.a y 2.9.b: Casos de puntuación: Final del principio de perseveración.

**2.3.1 .Evaluación del test**

**Categorías completas:** El número de categorías completa es, simplemente, el número de secuencia de diez respuestas correctas consecutivas, que siguen el criterio de clasificación de una categoría, dadas por el sujeto en la prueba. En este apartado, las puntuaciones van desde un mínimo de 0 a un máximo de 6. A medida que el sujeto las vaya completando; el examinador marcará cada categoría en la parte superior de la Hoja de anotación, utilizando por ello la secuencia escrita en ella (C F N C F N).

**Intentos para completar la primera categoría:** El número total de intentos para completar correctamente la primera categoría proporciona una indicación de la conceptualización inicial antes de que se produzca un cambio de ordenación. Los ejemplos de puntuación de la figura anteriores (buscar cual es), muestran que el sujeto (a) completó la primera categoría en 16 intentos, el (b) en 17 intentos el (d) solamente solo necesito 10. Es raro que un sujeto a quién se le apliquen las 128 tarjetas no complete con éxito la primera categoría. En los casos en que esto sucede, se concede al sujeto la puntuación directa 129 intentos para completar la primera categoría.

**Porcentaje de errores perseverativos:** el porcentaje de errores perseverativos refleja la densidad o concentración de los errores perseverativos en la realización del test. Se calcula dividiendo el número de errores perseverativos por el número de

intentos realizados. El resultado de esta operación se multiplica por 100 y se redondea al entero más próximo. Por ejemplo, para un sujeto que haya completado el WCST en 106 intentos y con 15 errores perseverativos, el porcentaje de errores perseverativos será el 14%.

**Fallos para mantener la actitud:** Otra puntuación básica del WCST es la correspondiente a los fallos en el mantenimiento de la actitud. Este tipo de fallo se produce cuando el sujeto da cinco o más repuestas correctas consecutivas y luego comete un error antes de completar la categoría con éxito. El fallo en mantenimiento de la actitud no se puntúa si, antes de finalizar el test, el sujeto coloca correctamente cinco tarjetas o más y luego simplemente pasa las tarjetas hasta llegar a la 128.

La figura siguiente presenta el ejemplo de un sujeto que tenía considerable dificultad para mantener el principio de clasificación Color. En el segundo intento (no se muestra el dato), el sujeto cometió un error no ambiguo en forma, estableciendo así Forma como principio de perseveración. A pasar de recibir “feedback” positivo del examinador para numerosas respuestas correctas no ambiguas a Color, este sujeto, profesor de escuela de 59 años, no pudo continuar respondiendo coherentemente con respecto a esa categoría. En menos de 25 intentos (del 15 al 38), este sujeto tuvo tres fallos en mantenimiento de la actitud (intentos 23,30 y 38).

**c**

1	15	C	F	N	O
2	16	C	F	N	O
	17	C	F	N	O
1	18	C	F	N	O
2	19	C	F	N	O
3	20	C	F	N	O
4	21	C	F	N	O
5	22	C	F	N	O
	23	C	F	N	O
	24	C	F	N	O
1	25	C	F	N	O
2	26	C	F	N	O
3	27	C	F	N	O
4	28	C	F	N	O
5	29	C	F	N	O
	30	C	F	N	O
1	31	C	F	N	O
2	32	C	F	N	O
3	33	C	F	N	O
4	34	C	F	N	O
5	35	C	F	N	O
6	36	C	F	N	O
7	37	C	F	N	O
	38	C	F	N	O

Fallo actitud p

p Fallo actitud

p Fallo actitud

**Figura 2.10:** Puntuación de fallos para mantener la actitud.

**Porcentaje de respuestas** de nivel conceptual: presumiblemente, refleja la comprensión de los principios de clasificación. El nivel conceptual de respuestas está definido por series de tres o más respuestas correctas consecutivas. Estas repuestas son llamadas respuestas de nivel conceptual porque se presume que requieren alguna comprensión de la estrategia necesaria para hacer colocaciones correctas y que es poco probable conseguir una serie correcta de esta longitud por azar. El porcentaje de respuestas de nivel conceptual se calcula dividiendo el número total de respuestas de

este tipo entre el número total de intentos, multiplicando el resultado por 100 y redondeando al número obtenido al entero más próximo.

Ejemplo, de los 128 elementos del test, el sujeto dio 70 respuestas de Nivel Conceptual en diez series de tres o más respuestas consecutivas correctas (10,10,10,3,10,3,4,4,10 y 6). Dividiendo 70 (número de respuestas de nivel conceptual) entre 128, multiplicando el resultado por 100 y redondeándolo al entero más próximo se obtiene el porcentaje de respuestas de Nivel Conceptual, que en este caso es 55%.

Para un sujeto dio 79 respuestas de nivel conceptual en 116 intentos (8,3,13,11,10,10,3,11,10). En este caso, el porcentaje de respuestas de nivel conceptual es 68%.

**Aprender a aprender:** esta refleja el promedio de los cambios en eficacia conceptual del sujeto a través de las diversas etapas de su realización en el WCST. El cálculo de esta puntuación requiere de cuatro pasos:

1- Determinar si es posible o no calcular la puntuación Aprender a aprender. Para ello es preciso contar el número de categorías que el sujeto ha completado o intentado. Se considera que se ha “intentado” una categoría cuando en ella se han realizados por lo menos 10 intentos, incluso si dicha categoría no se ha completado correctamente (por ejemplo, al final del test). La puntuación Aprender a aprender solamente se puede calcular en los casos en que el sujeto ha completado tres categorías o más o si ha completado dos e intentado la tercera.

2- El segundo paso es calcular el porcentaje de errores en cada categoría que se ha completado o intentado (total de errores cometidos en esa categoría dividido por el total de intentos realizados en ella y multiplicado por 100).

3- Calcular las diferencias entre el porcentaje de errores de dos categorías consecutivas o adyacentes. La diferencia entre porcentajes de errores se obtiene restando el correspondiente a la categoría 2 (Forma) del correspondiente a la categoría 1 (Color); el correspondiente a la categoría 3 (Número) del correspondiente a la categoría 2 (Forma) y así sucesivamente. Para finalizar, los valores de las diferencias en los porcentajes de errores se suman y se calcula la media que constituye la puntuación directa en Aprender a aprender. Una puntuación positiva en este factor indica una mejora en eficacia a través de la realización de categorías consecutivas, presumiblemente a causa del aprendizaje. Sin embargo es, probable que la mayoría de los sujetos, obtengan una puntuación negativa en ese aspecto porque se incluye en el cálculo la diferencia del porcentaje de errores entre la categoría 1 y la categoría 2 y esta inclusión hace más difícil obtener una puntuación positiva en Aprender a aprender dada la mayor dificultad que representa realizar el primer cambio de la categoría de clasificación. Sin embargo, en los datos normativos se ha tenido en cuenta este hecho; es decir, los baremos permiten comparar la puntuación de un sujeto dado con la obtenida por su grupo de referencia (que tiene sus mismas características) y así es posible identificar una realización relativamente buena incluso si la puntuación total no es positiva.

. Un caso en que el sujeto ha completado correctamente cinco categorías e intentando una sexta (realizando en ella 17 intentos). Se puede calcular la puntuación Aprender a aprender.

Para obtener las seis puntuaciones de porcentaje de error se divide el número de errores de cada categoría entre el número de intentos correspondiente a esa categoría y este resultado se multiplicó por 100, los datos obtenidos se especifican a continuación: 16,67% (2/12), 36,84% (7/19), 38,10% (8/21), 45,83% (11/24), 31,43 (11/35) y 58,82% (10/17). Las puntuaciones correspondientes a diferencias entre los porcentajes de errores en pares de categorías adyacentes son: -20,17 (16,57-36,84); -1,26 (36,84 -38,10); -7,73 (38,10-45-83), +14,40 (45,83-31,43) y -27,39 (31,43 – 58,82). La suma de los valores de las puntuaciones diferencia de porcentaje de errores es igual a -42,15 y esta cifra corresponde a una diferencia media, o una puntuación directa en Aprender a aprender de -8,43 (-42,15 dividido entre 5).

**Porcentaje de errores, Porcentaje de respuestas perseverativas y Porcentaje de errores no perseverativos:** Estas puntuaciones se presentan principalmente como ayuda para posibles investigaciones. Se calculan dividiendo la puntuación directa de cada uno de esos aspectos entre el número de intentos realizados, multiplicando el resultado por 100 y redondeando la cifra resultante al entero más próximo. Las puntuaciones de porcentaje de errores pueden resultar muy útiles en estudios de investigación en los que sea necesario controlar las diferencias en el número de intentos administrados. Sin embargo, no se recomienda su uso para la interpretación clínica del WCST porque implican valores corregidos para una medida del éxito en todo el test (el número de intentos empleados para completar la prueba), y porque los índices de fiabilidad de estos porcentajes son más bajos que los de las puntuaciones que los generan.

## CAPÍTULO 3

### PREPARACIÓN DE LAS HERRAMIENTAS Y ENTORNO DE DESARROLLO

---

---

En este capítulo, se describen las herramientas informáticas necesarias para el desarrollo de la aplicación programada y el almacenamiento de datos recogidos y tratados.

Como es de esperar todas estas herramientas son de distribución libre, requisito fundamental para el desarrollo de la aplicación. De esta forma posteriormente no existirá ninguna inferencia en futuros desarrollos de la misma.

Durante el capítulo se va a ir explicando el proceso paso a paso de preparación e instalación de los paquetes informáticos, todo acompañado de las imágenes de cada una de las etapas de instalación. Se irán incluyendo los parámetros necesarios de configuración de cada paquete así como una descripción gráfica detallada

Al final del capítulo se hace una breve descripción del entorno de trabajo, enfocada en las partes principales de las herramientas de desarrollo que a continuación se describen.

#### 3.1 HERRAMIENTAS DE DESARROLLO

Se pretende establecer la base y recursos necesarios para la instalación de nuestra plataforma de desarrollo java y la base de datos MySql. Para ello se realizará la descarga, instalación y configuración de JDK , NETBEANS y XAMP. Pero antes de esto se describen cada uno de ellos y sus funciones.

##### 3.1.1 Lenguaje de programación Java

El entorno de desarrollo Java consta de dos partes fundamentales:

- **Java Development Kit** o **JDK** (compilador Java), se trata de un conjunto de programas y librerías que permiten desarrollar, compilar y ejecutar programas en java, incorpora además la posibilidad de la ejecución parcial del programa,

deteniendo la ejecución en el punto deseado y estudiando en cada momento el valor de las distintas variables con el denominado debugger.

Puede instalarse en una computadora local o en una unidad de red. En la unidad de red se pueden tener las herramientas distribuidas en varias computadoras y trabajar como una sola aplicación.

- **NetBeans** es un entorno de desarrollo integrado libre, escrito en java y por tanto hecho principalmente para este, pero puede servir para cualquier otro lenguaje de programación. Es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Existe además un número importante de módulos para extenderlo.

La plataforma *NetBeans* IDE es de código abierto, un producto libre y gratuito sin restricciones de uso, programado completamente en Netbeans, esto lo convierte en un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente.

La usada inicialmente en el desarrollo del proyecto es la versión: [NetBeans 7.2](#) noviembre de 2012 y al finalizar el proyecto está disponible la 8.0

Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente.

La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:

- Administración de las interfaces de usuario (ej. menús y barras de herramientas)

- Administración de las configuraciones del usuario
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato)
- Administración de ventanas
- Framework basado en asistentes (diálogos paso a paso)

### 3.1.2 Servidor de base de datos MySql: XAMPP

**XAMPP** es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl.

El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y MacOS X.

#### ➤ *Características y requisitos*

XAMPP solamente requiere descargar y ejecutar un archivo zip, tar o exe, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin.

Para instalar XAMPP se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas por separado, en nuestro caso solamente requeriremos configura MySQL que es el que vamos a utilizar.

#### ➤ *Aplicaciones*

Oficialmente, los diseñadores de XAMPP sólo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios webs y programadores testear su trabajo en sus propios ordenadores sin ningún acceso a Internet. En la práctica, sin embargo, XAMPP es utilizado actualmente como servidor de sitios Web, ya que, con algunas modificaciones, es generalmente lo suficientemente seguro para serlo. Con el paquete se incluye una herramienta especial para proteger fácilmente las partes más importantes.

## 3.2 INSTALACIÓN: PAQUETES DE SOFTWARE

### 3.2.1 Instalación y configuración de XAMPP

Xampp es el primer paquete se debe instalar en el PC, ya que como se ha comentado en el apartado anterior este nos pone a disposición dos herramientas necesarias para la correcta puesta en funcionamiento de la base de datos. Estas en concreto son la base de datos de MySQL y el servidor Web Apache, este último necesario para poder trabajar en modo local en el PC. Los pasos seguidos hasta la correcta instalación de XAMPP son los siguientes:

#### 1-. Descarga de XAMPP

Se puede descargar la versión más actual de la página o enlace siguiente: <http://www.apachefriends.org/en/xampp-windows.html>

Aunque solo se va a hacer uso de dos de los módulos, (los dos primeros), en las siguientes líneas se nombran todos los módulos de los que dispone este.

- Apache 2.4.3
- MySQL 5.5.27
- PHP 5.4.7
- phpMyAdmin 3.5.2.2
- FileZilla FTP Server 0.9.41
- Tomcat 7.0.30 (con mod\_proxy\_ajp como conector)
- Strawberry Perl 5.16.1.1 Portable
- XAMPP Control Panel 3.1.0 (de hackattack142)

#### 2-. Instalación de XAMPP

- a. Se selecciona la versión del instalador que es la forma más fácil de instalar XAMPP. (La versión siguiente es la utilizada en la realización del proyecto y la incluida en el Cd, para una versión más actualizada descargar de la página anterior indicada o página del autor.)

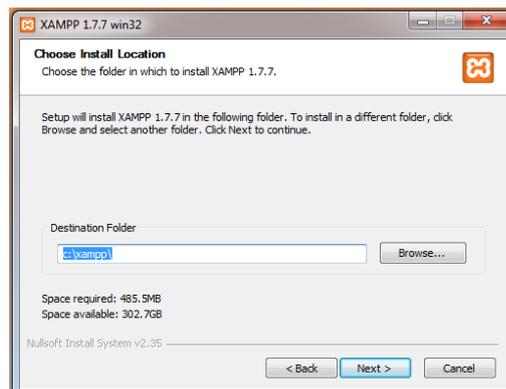


- b. En la pantalla de Bienvenida de XAMPP, recomienda cerrar otras aplicaciones antes de iniciar la instalación, una vez hecho esto se procede a Aceptar “Next”, (*Figura 3.1*)



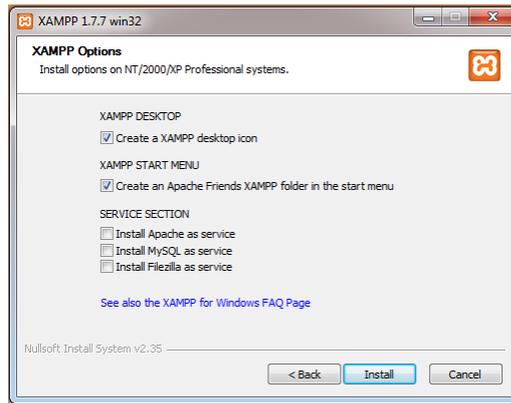
**Figura 3.1:** Segundo paso en la instalación de XAMPP

- c. En este punto se selecciona el directorio de instalación o se deja el que viene por defecto, además nos informa sobre el tamaño requerido para la instalación así como del que disponemos en el PC. (Figura 3.2)



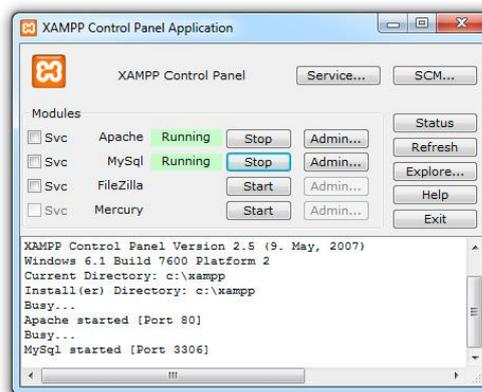
**Figura3.2:** Paso c en la instalación de XAMM

- d. En la figura 3.3 que muestra una ventana “Opciones” de la instalación, en esta se configura como se desea que arranque XAMPP, mediante la marcación de las tres últimas casillas se podrán seleccionar que los servicios web, bases de datos MySQL y FTP se inicien de forma automática al arrancar el sistema operativo. En caso contrario, será necesario arrancarlos cuando se quieran utilizar. (Figura 3.3)



**Figura 3.3 :** Selección de servicios en instalación de XAMPP

- e. Tras unos minutos de instalación, aparece la ventana que avisa el final de la instalación. Pulsando finish, termina esta. A continuación para mostrar el panel de control de XAMPP, pulsar “Si”.
- f. En el panel de control se pueden arrancar y parar los diferentes servicios, así como especificar cuáles de ellos se arrancan de forma automática al inicio de arrancar el sistema operativo (casillas con la etiqueta “Svc”).



**Figura 3.4:** Activación de servicios de XAMPP

Para nuestra aplicación, como muestra figura anterior deben estar activos los servicios Apache y MySQL. También es posible desde este menú ingresar en el administrador de MySQL mediante el navegador que esté instalado en el PC.

### 3-. Configuración de XAMPP

- a. Si no está abierto el panel de control, hacer clic en el icono del escritorio “XAMPP Control Panel” y se abrirá la ventana mostrada en la figura del punto anterior

- b. Arrancar los servicios de interés por medio de los botones Start. En nuestro caso MySQL. Es recomendable este se arranque con el inicio de Windows, para lo cual se deben activar la casilla de dicho servicio etiquetado como Svc.
- c. Arrancar los servicios de interés por medio de los botones Start. En nuestro caso MySQL. Es recomendable este se arranque con el inicio de Windows, para lo cual se deben activar la casilla de dicho servicio etiquetado como Svc.



**Figura 3.5:** Comprobación de la correcta instalación de XAMPP mediante el navegador.

- d. Seleccionando el idioma entramos en: [Http://localhost/xampp/index.php](http://localhost/xampp/index.php)



**Figura 3.6:** Pantalla de bienvenida de XAMPP.

### 3.2.2 Instalación del Java Development Kit – JDK

Para la instalación de NetBeans es necesario que esté instalado el JDK de java más reciente. Se puede descargar de la página del autor en el siguiente enlace: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>, en la sección que en el momento de la redacción de este documento se llamaba: Java SE 7u3, concretamente en el botón “DOWNLOAD JDK”. Para Windows 7, es el fichero siguiente (figura 3.7)

**Java SE Development Kit 7u9**

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	120.63 MB	<a href="#">jdk-7u9-linux-i586.rpm</a>
Linux x86	92.85 MB	<a href="#">jdk-7u9-linux-i586.tar.gz</a>
Linux x64	118.82 MB	<a href="#">jdk-7u9-linux-x64.rpm</a>
Linux x64	91.59 MB	<a href="#">jdk-7u9-linux-x64.tar.gz</a>
Mac OS X	143.47 MB	<a href="#">jdk-7u9-macosx-x64.dmg</a>
Solaris x86	135.14 MB	<a href="#">jdk-7u9-solaris-i586.tar.Z</a>
Solaris x86	91.51 MB	<a href="#">jdk-7u9-solaris-i586.tar.gz</a>
Solaris SPARC	135.7 MB	<a href="#">jdk-7u9-solaris-sparc.tar.Z</a>
Solaris SPARC	95.15 MB	<a href="#">jdk-7u9-solaris-sparc.tar.gz</a>
Solaris SPARC 64-bit	22.8 MB	<a href="#">jdk-7u9-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	17.51 MB	<a href="#">jdk-7u9-solaris-sparcv9.tar.gz</a>
Solaris x64	22.48 MB	<a href="#">jdk-7u9-solaris-x64.tar.Z</a>
Solaris x64	14.94 MB	<a href="#">jdk-7u9-solaris-x64.tar.gz</a>
Windows x86	88.35 MB	<a href="#">jdk-7u9-windows-i586.exe</a>
Windows x64	90.03 MB	<a href="#">jdk-7u9-windows-x64.exe</a>



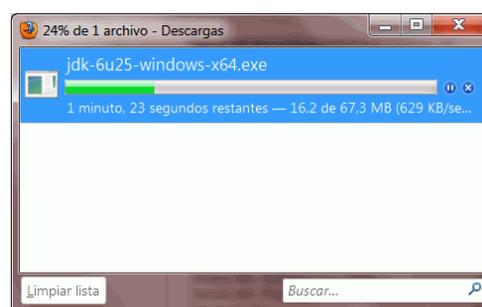
**Figura 3.7:** Instalación del JDK en Windows - Archivo de descarga.

Entonces empezaremos a recibir un único fichero de gran tamaño (cerca de 70 Mb, según versiones, figura 3.8).



**Figura 3.8:** Guardado del Jdk.

Al tener un tamaño tan grande, la descarga puede ser lenta, según la velocidad de nuestra conexión a Internet, a continuación se muestra el gestor de descargas figura 3.9.



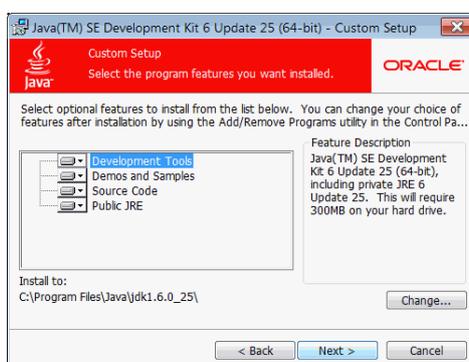
**Figura 3.9:** Gestor de descargas, iniciando la descarga.

Tras haber descargado, hacemos doble clic en el fichero para comenzar la instalación propiamente dicha, debe aparecer la ventana de bienvenida de Oracle según figura 3.10.



**Figura 3.10:** Pantalla de bienvenida de XAMMP.

A continuación se pueden afinar detalles como la carpeta de instalación, o qué partes no queremos instalar (por ejemplo, podríamos optar por no instalar los ejemplos). Para no complicarnos, si tenemos suficiente espacio (posiblemente unos 400 Mb en total), podemos hacer una instalación típica, sin cambiar nada: (Figura 3.1)



**Figura 3.11:** Pantalla de sección de opciones en la instalación del Jdk

Ahora se descomprime todo en la localización indicada y se instala, En cierto punto se nos preguntará si queremos instalar la máquina virtual Java (Java Runtime Environment, JRE). Lo razonable será responder que sí, ya que esta será necesaria para la posterior ejecución del programa sin necesidad de abrir NetBeans. En otro capítulo se describirán los requisitos necesarios para la ejecución de la aplicación, que por su puesto no serán los mismo que para su desarrollo. Igual que para el JDK en este caso es posible cambiar la carpeta de instalación, pero siguiendo las mismas pautas que anteriormente se mantiene el directorio que nos asigna el asistente de instalación.

Tras la extracción de los archivos el asistente continúa con la instalación de los mismos en el sistema operativo. En el caso de que todo el proceso haya transcurrido de forma correcta, al final se debe obtener un mensaje de confirmación.

Finalmente se nos propondrá registrar la copia en la página de Oracle con fines de mejora para el desarrollador y posibilidad de soporte, según figura 3.12. (No es necesario.)



**Figura 3.12:** Pantallas de programas de instalación y finalización.

I

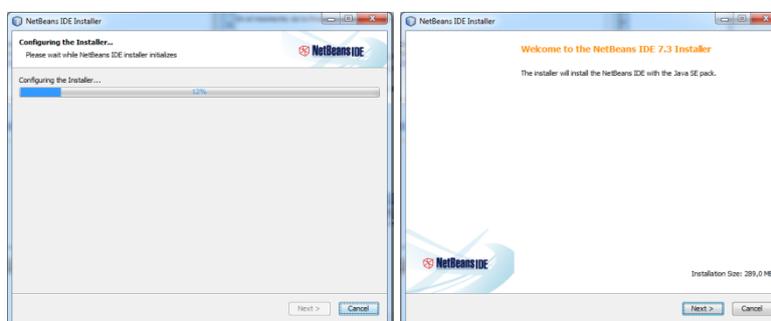
### 3.2.3 Descarga e Instalación del IDE de Netbeans.

Al igual que ocurre con las herramientas anteriores, es posible que en el momento de instalación de NetBeans haya cambiado la versión de este respecto a la utilizada durante el desarrollo del presente proyecto, por ello además de incluir en este los archivos de instalación correspondientes a continuación se da la dirección de descarga de la versión más actualizada de la web del autor [netbeans.org](https://netbeans.org/downloads/index.html). <https://netbeans.org/downloads/index.html>

- En el momento del inicio de la realización de la aplicación la versión: 7.2.1
- En el momento de la finalización del mismo ya estaba desarrollada la versión 8.0.

Una vez descargado el instalador se procede a:

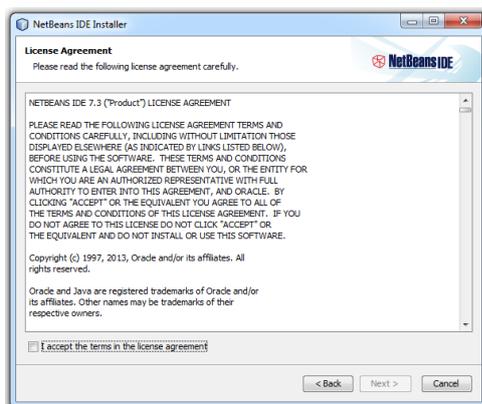
- a. Ejecutar el fichero descargado, (la versiones anteriores son las incluidas en el cd del proyecto), y esperar hasta que nos permita continuar. (figura 3.13)



**Figura 3.13:** Pantallas iniciales del instalador de Netbeans.

- b. Tras la configuración del instalador, se pulsa “Next” (continuar), si hubiésemos querido cambiar la dirección de instalación también podría haber sido posible.

- c. Aceptamos los términos de la licencia con la casilla de verificación y pulsar “Next”. (figura 3.14)



**Figura 3.14:** *Terminos de licencia de NetBeans IDE.*

- *A fecha de finalización del proyecto el autor también da la posibilidad de una instalación conjunta del JDK de Java y NetBeans. –*

### 3.3 BREVE DESCRIPCIÓN DEL ENTORNO DE TRABAJO

Una vez finalizada la instalación y comprobación del correcto funcionamiento de todas las herramientas, se procede en este último apartado del capítulo a realizar una breve descripción de las zonas de trabajo de las mismas. Esto es necesario para que cuando en capítulos posteriores se vaya describiendo como se ha desarrollado la aplicación no sea necesario tener que hacer muchas aclaraciones en relación a la zona a la que hay que acceder para proceder.

#### 3.3.1 Entorno de trabajo de NetBeans.

Para la creación de las interfaces vamos a trabajar con elementos del tipo JFrame, JPanel, JDialog, JInternalFrame y otras que son parte del AWT de NetBeans, esto se irá viendo conforme avanzamos en el montaje de la aplicación.

En la siguiente figura 3.15, se muestran las zonas de trabajo del IDE de NetBeans, para cuando posteriormente sean referenciadas su localización sea directa. A continuación

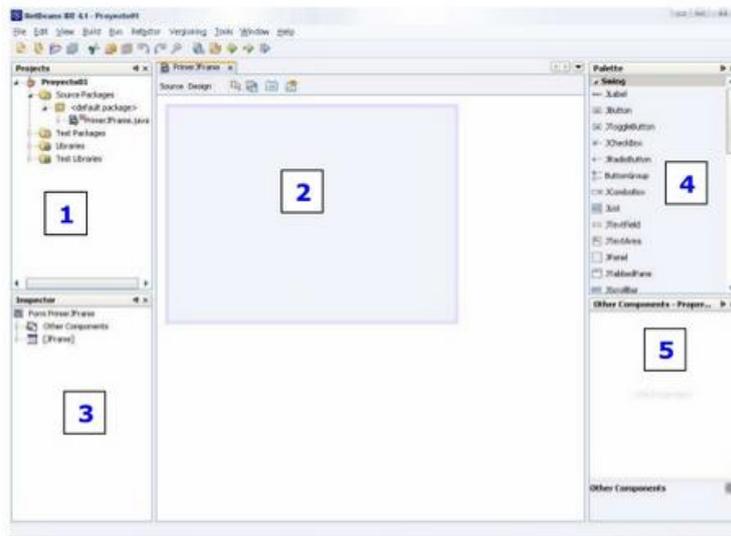


Figura 3.15. Zonas de trabajo del IDE.

#### Descripción de las zonas del IDE:

**Zona 1-** Proyectos: como vemos el nombre del proyecto aparece en la esquina superior y en negrita, a partir de este se ramifica el árbol de las subcarpetas, las cuales agrupan los códigos fuente de las clases junto con sus respectivos enlaces a sus interfaces gráficas (en caso de tenerlas), las librerías usadas y algunos otros.

En la primera de ellas (Source Package) estará básicamente todo lo que desarrollaremos, desde las clases más sencillas de solo código hasta las complejas interfaces gráficas. Un click derecho sobre esta carpeta source permitirá la creación de nuevas clases, ya sean gráficas o no.

**Zona 2-** Form: (Pizarra de diseño y código). Esta será la zona que mostrará todo el trabajo que realicemos en el proceso de diseño de nuestra interfaz. En la parte más superior de ella, se encuentra una pestaña la cual indica el nombre de la clase gráfica y una pequeña X para poder cerrar las clases que no necesitemos.

**Zona 3-**Inspector: Esta zona funciona a modo de árbol; es el navegador de los objetos presentes en nuestra interfaz gráfica; es decir que allí se mostrará una lista jerárquica de todos los componentes colocados en el formulario, y desde ahí se controlará el orden de estos, sus respectivos nombres y con el clic derecho sobre cada objeto, se podrá modificar las propiedades que cada uno posee.

**Zona 4-**Palette: (Herramientas) Aquí se encuentran los componentes gráficos que tienen las librerías Swing y AWT de Java. Con solo hacer clic sobre el objeto deseado y luego clic sobre el área de trabajo se irá diseñando la interfaz y añadiendo componentes.

**Zona 5-** Properties: Tal vez sea esta la zona más importante del entorno de programación gráfica de este EDE. En ella se encuentran las propiedades especiales y atributos que tiene cada componente gráfico. En una interfaz todo componente, desde el más externo como el JFrame hasta el más simple un JSeparator tienen ciertas

propiedades y de comportamiento ante los eventos que permiten una alta personalización del diseño de los objetos colocados en el área de trabajo.

### 3.3.2 Entorno de trabajo de MySql.

Para poder empezar a trabajar en un pc con una base de datos en MySql, como se ha mencionando en párrafos anteriores, es necesario activar los módulos del servidor XAMPP: MySql y Apache. Esto no será necesario si durante la instalación se marco la opción de inicio automático con el arranque del Sistema Operativo.

Para ello se accede al panel de control de Xampp mediante el acceso creado durante su instalación, se pulsa la acción: “Start” sobre ambos módulos. Una vez estos estén corriendo en el PC, se puede acceder al administrados de bases de datos figura 3.16b, (donde se van a crear), pulsando sobre el botón “Admin”, justo a la derecha del nombre del módulo MySql. Estos botones son resaltados en la figura 3.16a

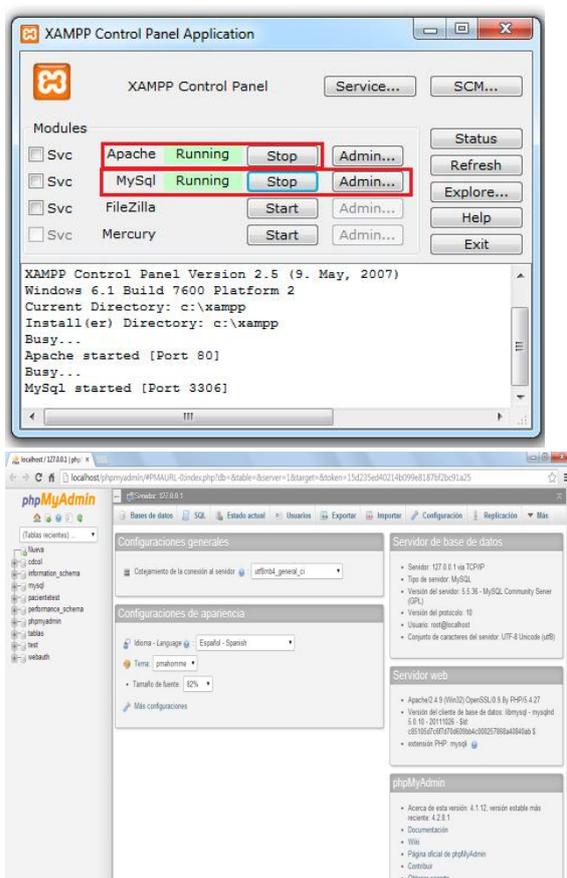


Figura 3.16a. Panel de Control, servicios activados.

Figura 3.16.b. Administrador phpMyAdmin

El Módulo Apache solo es necesario activarlo durante el diseño de la aplicación, no siendo así, cuando el trabajo con la base de datos se realice a través de la aplicación informática.

## ***CAPÍTULO 4***

### **INICIO Y ORGANIZACIÓN DEL PROYECTO**

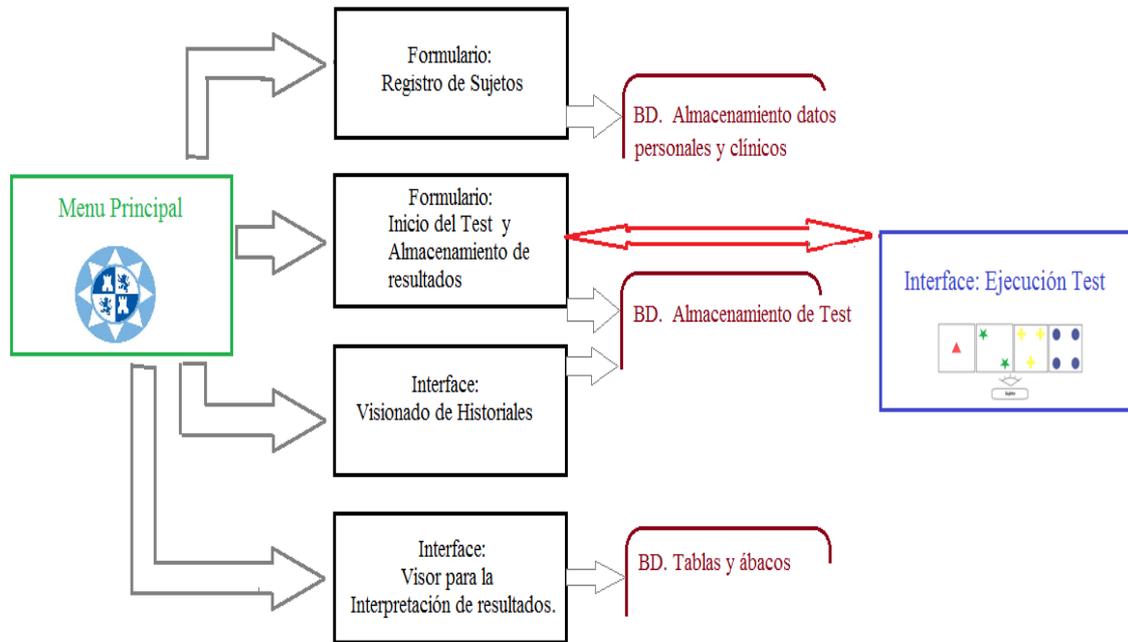
---

---

Una vez revisada toda la documentación relevante a la comprensión del desarrollo del test, se procede a crear las interfaces de los distintos menús, formularios, así como las interconexiones entre ellas. También se creará otro módulo que lleve a la ejecución del test de forma interactiva. Antes del diseño de estos módulos se presenta en un punto de este capítulo el esquema general perseguido (“boceto de la aplicación”).

También se abordará en el presente capítulo, los pasos en la creación de un nuevo proyecto, así como la estructura, interface y comunicación con el menú principal. A partir de aquí se irán desarrollando el resto de módulos y clases (java) que van a dar forma a la aplicación en posteriores capítulos.

#### **4.1 ESQUEMA GENERAL DE LA APLICACIÓN. “BOCETO”**



**Figura 4.1.** Esquema general de la aplicación WCST.

*Descripción del esquema:*

Partiendo de un menú principal se podrá acceder a las interfaces siguientes:

- **Formulario de pacientes:** en esta se procederá al almacenamiento de datos relativo a los sujetos, tanto personales como clínicos.
- **Formulario de inicio:** una vez registrados los sujetos, en esta interface se procede a la selección del sujeto a examinar y a la llamada de la interface para la Ejecución del propio test. Esta debe estar diseñada para poder complementar otros datos relativos a la evolución de la prueba, una vez que esta ha sido realizada.
- **Interface para el visionado de historiales.** Como su nombre indica accediendo a esta será posible rescatar datos de la base de datos y mostrarlos de una forma atractiva, dando también la posibilidad de imprimir resultados.
- **Interface para la Interpretación de resultados:** constituida por un visor que permita mostrar tablas, ábacos, métodos y otros datos de interés para poder comprobar resultados de la prueba.

Existirán tres bases de datos diferentes para almacenar la información necesaria, estas son descritas a continuación.

- Base de datos para **almacenamiento de datos personales y clínicos.**
- Base de datos para **almacenamientos de test y resultados.**

- Base de datos de **tablas y ábacos** utilizados en el cálculo de resultados estadísticos. (No son la utilizadas en el visor).

Además de los formularios y bases de datos se debe diseñar la ejecución del test con una interface atractiva para el paciente, acercándose lo máximo posible a la forma de aplicación Tradicional.

En el esquema anterior se pueden ver las relaciones que mantienen todos los elementos anteriores. Es un esquema general, que sufre pequeñas modificaciones durante el desarrollo de la aplicación y no muestra información sobre las clases y métodos que son utilizados en cada una de las partes que lo componen. Esta información será descrita durante la evolución de la memoria de proyecto.

## 4.2 INICIO EN LA DEL CREACIÓN PROYECTO.

Instalados los programas descritos en el capítulo dos, ya estamos preparados para empezar a diseñar y programar la aplicación.

Por lo tanto, lo primero que se debe hacer, es crear en el IDE de NetBeans el correspondiente proyecto bajo el nombre de Wisconsin. De esta forma, se creará el proyecto bajo el cual se desarrollarán las distintas partes y clases que van a formar la aplicación. Una vez hecho esto tenemos el proyecto abierto para comenzar a trabajar sobre él.

Para la creación del proyecto, una vez abierto el IDE, se selecciona en menú superior la opción Archivo → "Proyecto Nuevo" apareciendo una ventana en la que deberemos seguir dos pasos sencillos para la creación del mismo, el primer paso consiste en seleccionar el tipo de proyecto que deseamos realizar, en nuestro caso, de la categoría Java seleccionamos "JavaApplication" (Aplicación Java), y pulsamos botón siguiente. En la figura 4.2. se ve esto de forma gráfica.



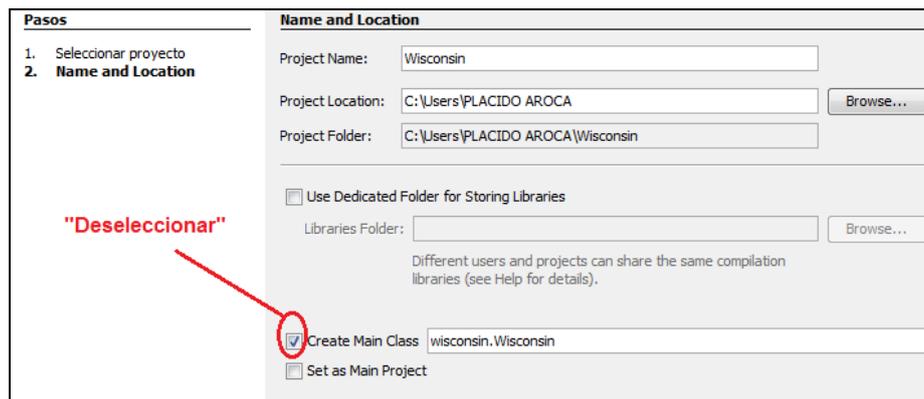
**Figura 4.2.** Paso 1 en la creación del proyecto: Seleccionar tipo de proyecto

El segundo paso consiste en darle nombre al proyecto, el cual será el mismo de la carpeta en la que se alojará todo lo que hagamos. Asegúrese de revisar la carpeta en la cual quedará almacenado el proyecto para que después recuerde en donde puede acceder a sus ficheros sin necesidad de abrir nuevamente el IDE. En nuestro caso, como hemos mencionado antes el nombre del proyecto será “Wisconsin”. Antes de dar clic a siguiente “Next”, hay que revisar dos opciones:

- “Set a Main Project”
- “Create a main class”

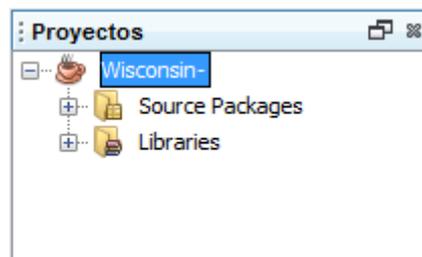
La primera de ella solo se deja activa si se desea que el proyecto que va crear actualmente sea la aplicación principal dentro del IDE. En nuestro caso la deseccionaremos. Y la segunda opción indica si el asistente creará una clase inicial la cual contendrá el método main() para la ejecución del proyecto (como vamos a usar NetBeans como IDE para la creación de interfaces gráficas, es necesario que este desactivada).

En la siguiente figura se muestra la ventana correspondiente a este paso.



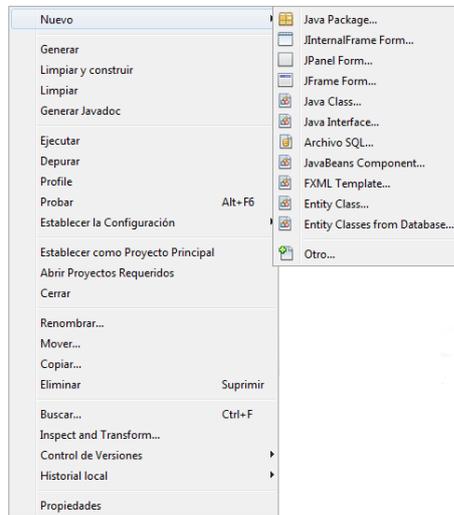
**Figura 4.3.** Paso 2 en la creación del proyecto: Nombre y localización.

Al finalizar el asistente, encontramos en la parte izquierda una división que mostramos en la figura 4.4.



**Figura 4.4:** Zona proyecto del IDE.

Tras tener correctamente montado el inicio del proyecto, se puede crear todo lo que necesitemos a partir de un click derecho sobre el nombre del proyecto y escoger una de las interfaces gráficas o tipos de clases que brinda el asistente de NetBeans: Figura 4.5.



**Figura 4.5.** Tipos de clases que brinda el asistente de NetBeans

### 4.3 DISEÑO DEL MENÚ PRINCIPAL

Este menú será el primero que nos encontramos al iniciar la aplicación, desde aquí seleccionaremos las acciones que nos interesen en cada momento, incluida la ejecución del test.

El contenido de este menú y de sus respectivos submenús aparece a continuación, posteriormente se explicarán con más detalle las opciones que nos van a permitir cada uno. Y en los capítulos siguientes se irán desarrollando las clases o bloques a las que accedemos desde este menú.

#### Componentes del menú:

- *Barra de menús.*
- *Menús:*
  - ***Test***
    - *Registro Paciente*
    - *Test*
    - *Salir*
  - ***Historial***
    - *Buscar Paciente*
  - ***Info***
    - *Muestra de tipificación (Valores de referencia)*
    - *Versión*

Antes de empezar a diseñar la clase que implementará el Menú Principal, se cambia el nombre del package que viene por defecto por “Test”. Este es el que va a contener las clases principales de la aplicación al mismo tiempo que la hace ser más modular. Con la idea de que posteriormente cuando se empiece con la programación de la base de datos, sean separadas. Al mismo tiempo se crearán packages para almacenar las imágenes de las cartas, las de la animación, los sonidos, y otras imágenes utilizadas en la aplicación, aunque esto se verá en siguientes capítulos.

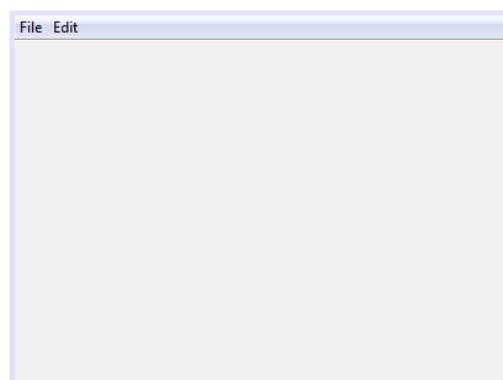
La componente básica que requerimos para implementar una interfaz visual con la librería Swing es la clase JFrame. Esta clase encapsula una Ventana clásica de cualquier sistema operativo con entorno gráfico (Windows, OS X, Linux etc.)

Se encuentra en el paquete javax.swing pero desde el IDE al crear una clase tipo JFrame Form lo que realmente se está haciendo es crear una clase que herede de la clase JFrame y extienda sus responsabilidades, pudiendo agregar botones, etiquetas, editores de líneas etc, esta es el “MenuPrincipal”.

La importación del paquete donde se encuentra la clase JFrame la incorpora directamente el IDE (`import javax.swing.*;`), además en la Main crea directamente un objeto de dicha clase, esto se podrá ver a continuación cuando se pase en lapizarra de diseño del IDE de desing a source.

Para crearla en el IDE procedemos a ir a la zona proyectos y situándonos sobre el package test, pulsamos botón derecho y añadir JFrame Form. Esta se podría decir que es la base sobre la que se van a situar los menús y al mismo tiempo lo que nos va a aparecer cuando ejecutemos la aplicación

Ahora se procede a añadir los menús, para esto situándonos sobre el JFrame del Menú principal y dirigiéndonos a la paleta de herramientas, seleccionamos de los posibles tipos elementos de menús, una barra de menú llamada “Menu Bar”. Esta es trasladada sobre el JFrame y contiene por defecto dos menús, File y Edit según la figura 4.5.



**Figura 4.5.** JFrame en el que se va a implementar el menú principal

Para configurar esta barra de menús se procede de la siguiente forma:

- 1- Se deben renombrar los dos menús ya existentes en la barra por “Test” e “Historial”, además debe añadirse otro titulado “Info”, para ello dirigiéndose a la paleta de herramientas se selecciona el elemento “Menu” y se desliza sobre la barra de menús.
- 2- Se insertan los submenús para cada menú anterior. En este caso el elemento adecuado de la paleta es “Menu ítem”, que igual que se hizo con los menús anteriores debe ser deslizado hacia el menú correspondiente, tantas veces como submenús procedan.
- 3- Renombrar los submenús según definiciones de la página anterior.

Una vez completada la barra de menús, para que la interface inicial quede más atractiva, se incluye un fondo a esta, este se puede ver en la figura x.x.

Para realizar esta operación se ha optado por llamar a la clase denominada “fondoMenuPrincipal” (mostrada en la tabla de código siguiente) ,desde la clase JFrame que implementa el “Menú principal”. Esta clase “fondoMenuPrincipal” se encarga de añadir un fondo al JPanel con Imagen de Fondo auto-ajustado a la ventana.

Se han suprimidos líneas de código de las librerías, solo se muestra el código de los métodos que realizan esta operación. El código completo al igual que todo el de la aplicación, se puede consultar en el anexo correspondiente.

```
1. public class fondoMenuPrincipal implements Border {
2.     BufferedImage fondo;
3.
4.     public fondoMenuPrincipal(){
5.         try {
6.
7.             URL url = new URL(getClass().getResource +
8.                 ("estimulos/logoupct.gif").toString());
9.
10.            fondo = ImageIO.read(url);
11.
12.        } catch (IOException ex) {
13.            Logger.getLogger(fondoMenuPrincipal.class.getName()) +
14.                .log(Level.SEVERE, null, ex);
15.        }
16.    }
17.    public void paintBorder(Component c, Graphics g, int x, int y,
18.        int width, int height){g.drawImage(fondo, (x + (width -
19.        fondo.getWidth())/2), (y + (height - fondo.getHeight())/2), null);
20.    }
21.    public Insets getBorderInsets(Component c){
22.        return new Insets(0,0,0,0);
23.    }
24.    public boolean isBorderOpaque(){
```

```
25.     return true;
26.     }
27.     }
```

**Tabla 4.1.** Código correspondiente a la clase “fondoMenuPrincipal”

Descripción del código:

” Implements” sirve para implementar clases abstractas en este caso de Border, estas tienen declaración de funciones sin ser implementadas. El” implements” obliga a la clase a implementar los métodos que estaban definidos (línea 1)

Se crea una variable “fondo” de tipo BufferedImage donde se va a almacenar la imagen. (línea 2)

En el constructor de clase, se crea una variable “url” del tipo URL, esta se inicia con la dirección relativa donde se encuentra el archivo imagen, para ello se utilizan los métodos getClass().getResource() y .toString. (líneas 7 y 8).

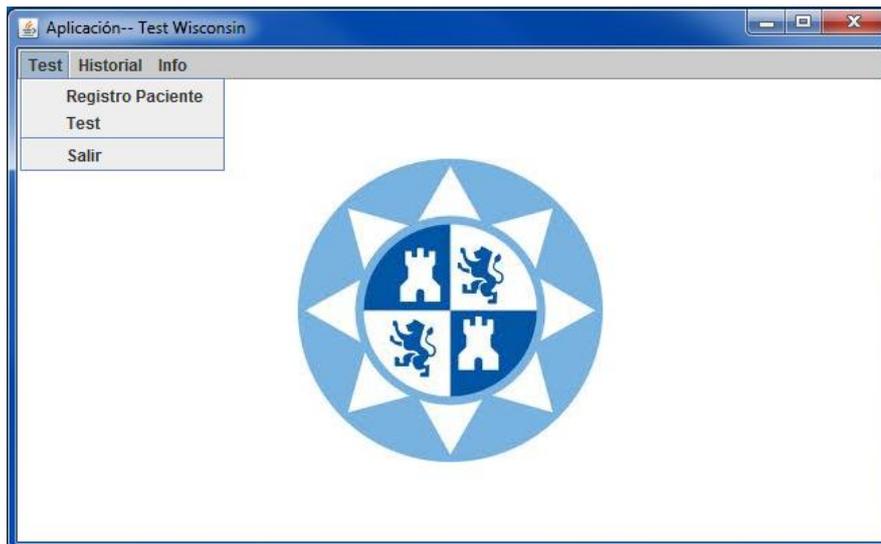
A continuación, de la clase IO se llama al método estático read(), donde enviamos por parámetro un archivo mediante la dirección del mismo, en este caso el de la imagen: “logoupct.gif” (línea 10)

Las líneas de código anteriores son incluidas en una estructura de control try...catch., esta estructura se usa para capturar el flujo de datos cuando por la intervención del usuario o de algún dispositivo externo sucede un error. En este caso el método read() puede tener un error de entrada y salida de datos (IOException).

Seguido del try...catch se procede a sobrescribir los métodos propios de border. El método paintBorder(), es configurado de forma que dibuje la imagen de fondo en el centro del contenedor y que cada vez que este sea redimensionado dicha imagen automáticamente se vuelva colocar en el centro. (línea 17)

Finalmente, tendremos una interface como la mostrada a continuación en la figura 4.6. Debemos notar que aún no se le han asignado las acciones a cada uno de los submenús, solamente se ha creado la interface gráfica y se han editado los correspondientes nombres o forma de nombrar a los menús.

Ahora estamos en disposición de asignarle acciones a cada uno de los submenús, lo normal sería hacer esto una vez creadas el resto de “JInternalFrames” que van a implementar las Ventanas (Interfaces), para el Registro de Paciente, realización del test etc, ya que estas van a ser llamadas durante la ejecución del programa desde el “Menu Principal”.



**Figura 4.6:** Interface del Menú principal.

Aunque el código de la clase “MenuPrincipal” no es completado hasta el final del diseño de la aplicación, a continuación se muestra una parte para darnos a conocer su funcionamiento general. El código corresponde a la llamada de una sola ventana o formulario desde este principal, en este caso el de “Registro”. Se ha omitido el resto, porque es demasiado repetitivo y sería menos ilustrativo.

Todo el código sería muy extenso para mostrarlo aquí. Se dispone del código completo en el anexo correspondiente a códigos de programa.

```

1. package Test;
2.
3. import javax.swing.JInternalFrame;
4. import javax.swing.JOptionPane;
5. import javax.swing.UIManager;
6.
7. public class MenuPrincipal extends javax.swing.JFrame {
8.
9.     private Registro p;
10.    private Iniciar i;
11.    private Historial h;
12.    private Visor ab;
13.
14.    public MenuPrincipal() {
15.
16.        initComponents();
17.        this.setTitle("Aplicación-- Test Wisconsin");
18.        panel.setBorder(new fondoMenuPrincipal());
19.        //formulario al centro de la pantalla
20.        this.setLocationRelativeTo(null);
21.
22.    }

```

**Tabla 4.2.** Código correspondiente a la clase “MenuPrincipal”(1/2)

Al comienzo de la clase deben ser instanciadas cuatro clases correspondientes a cada una de las “JInternalFrames” que van a constituir los formularios-interface de la aplicación, de esta forma podrán ser llamadas en cualquier momento. Las clases instanciadas correspondientes a Registro, Iniciar, Historial y Visor.

Dentro del constructor de clase, se inicializan los componentes de diseño y a continuación se instancia e inicializa la clase “fondoMenuPrincipal” descrita anteriormente, configura para que se muestre como fondo de la aplicación el logo de la universidad, siempre en el medio y autoredimensionable.

```
23. private boolean estacerrado(Object obj){
24.     JInternalFrame[] activos=panel.getAllFrames();
25.     boolean cerrado=true;
26.     int i=0;
27.     while (i<activos.length && cerrado){
28.         if(activos[i]==obj){
29.             cerrado=false;
30.         }
31.         i++;
32.     }
33.     return cerrado;
34. }
35.
36. private void jMIRegistroActionPerformed(java.awt.event.ActionEvent evt){
37.     if(estacerrado(p)){
38.         p = new Registro();
39.         panel.add(p);
40.         p.show();
41.     }
42.     else{
43.         JOptionPane.showMessageDialog(this,"La ventana ya
44.                                     Esta abierta...");
45.     }
46. }
47.
48. public static void main(String args[]) {
49.
50.
51.     java.awt.EventQueue.invokeLater(new Runnable() {
52.         @Override
53.         public void run() {
54.
55.             new MenuPrincipal().setVisible(true);
56.         }
57.     });
58. }
```

**Tabla 4.3.** Código correspondiente a la clase “MenuPrincipal”(2/2)

El primer método mostrado en la tabla 4.3 “estacerrado”, se usa en cada uno de los cuerpos de código que implementan las llamadas a los formularios de Registro, Historial, Inicio y Visor, mediante eventos para comprobar si la ventana de dicho

formulario está abierta, en cuyo caso nos devolvería el mensaje: “La ventana ya está abierta”

A continuación en el código del evento (línea 38), se ejecuta la clase instanciada, dando lugar a que se muestre el formulario de registro. Para que este aparezca en pantalla es necesario utilizar los métodos incorporados: `panel.add(p)` y `p.show()`;

De la línea 48 a la 59 aparece el método `main`, necesario en esta clase para que sea el motor de la aplicación. En este simplemente se instancia e inicializa la propia clase menú principal y mediante el método incorporado `setVisible(true)`, se muestra por pantalla.

## ***CAPÍTULO 5***

### **REGISTRO DE PACIENTES Y BASES DE DATOS**

---

---

En este capítulo se desarrollan todos los aspectos relativos al registro de pacientes, desde la configuración de la base de datos que los va a contener, hasta la interface que formará parte de la aplicación de escritorio. Dicha interface será la que permita al profesional recoger los datos personales y médicos relativos al paciente.

Aunque el contenido de este capítulo podría haber sido estructurado en dos, uno para el diseño de la interface de registro y otro para la base de datos, se ha optado por desarrollarlos de forma conjunta y así servir de referencia para los siguientes capítulos.

## 5.1 CONFIGURACIÓN DE LA CONEXIÓN EN MYSQL

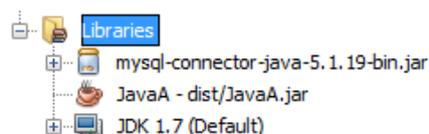
Además de ejecutar el test y visualizar los resultados, la aplicación debe ser capaz de almacenar los datos. Para ello, a continuación se presenta el procedimiento de conexión entre el lenguaje de programación Java y el sistema de gestión de bases de datos MySQL.

### 5.1.1 Preparación de la conexión

Inicialmente es necesario añadir al proyecto una librería que permite realizar un puente entre la base de datos MySQL y el lenguaje de programación java. Consiste en un conector que permite la interacción entre ambas. Este es descargado de la dirección electrónica: <http://dev.mysql.com/download>, el archivo viene en formato comprimido, tras descomprimirlo se tiene una carpeta con el mismo nombre, de la cual solo vamos a necesitar el “mysql-connector-java-5-1-16-bin” ó “mysql-connector-java-5-1-19-bin” según actualización de la versión, dicho archivo viene en formato .jar.

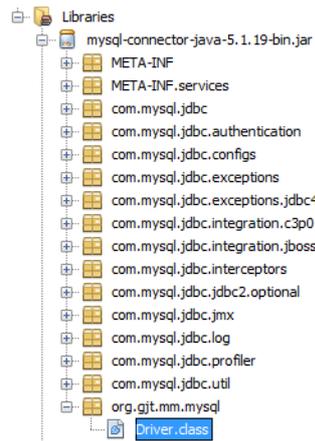
Para incluir el conector descargado y que nos permita la conexión con MySQL, nos dirigimos a donde está localizado el archivo, lo copiamos y lo llevamos a la carpeta donde está guardado el proyecto Wisconsin.

A continuación desde el IDE, entrando en bibliotecas se adiciona dicho elemento al proyecto para realizar la conexión de manera correcta. Para ello situándose encima de la carpeta bibliotecas correspondiente y con click derecho sobre esta, se selecciona la opción añadir .jar, en la ventana que aparece se busca la dirección del proyecto y se selecciona, una vez hecho esto en la estructura de nuestro programa deberá aparecer dicho archivo. En la siguiente figura se muestran las librerías que tendríamos disponibles para trabajar en nuestro programa tras añadir el conector java.



**Figura 5.1:** Hipotética hoja de anotación.

De nuestro conector vamos a usar la clase Drivers.class situada en la carpeta: “org.gjt.mm.mysql” mostrada a continuación en la figura x. Como se ha descrito antes, esta permitirá la conexión con nuestro servidor de base de datos MySQL.



**Figura 5.2:** Hipotética hoja de anotación.

Una vez que tenemos las herramientas necesarias para comunicación con la base de datos, el siguiente paso consistirá en crearla y configurarla.

### 5.1.2 Requisitos de almacenamiento en el registro de un sujeto

Para crear la base de datos, se debe conocer con exactitud los datos que esta va a contener y la forma de organizarlos.

Primero se procede a crear la tabla asociada al registro del paciente, esta deberá incluir los campos necesarios para almacenar tanto los datos personales como otros datos de interés para el evaluador. Para mostrar esto gráficamente se muestra en la figura 5.3. un ejemplo de hoja de anotación para el registro de paciente de forma tradicional.

Esta es la información necesaria para registrar un sujeto antes de proceder a su evaluación, posteriormente se ampliará con registros referidos a la evaluación del propio test, es decir a los resultados.

**Hoja de anotación**

Nombre		Año	Mes	Día
Sexo: V M    Lateralidad: D I		Fecha examen		
Ocupación		Fecha nacimiento		
Nivel de estudios		Edad Cronológica		
Examinado por				

---

**Problema de referencia:**

Antecedentes. Enfermedades actuales:

Medicación/Dosis:

Observaciones sobre la conducta:

---

Actitud durante la aplicación:	
RELACIÓN:	Excelente <input type="checkbox"/> Buena <input type="checkbox"/> Pasable <input type="checkbox"/> Mala <input type="checkbox"/>
COOPERACIÓN:	Excelente <input type="checkbox"/> Adecuada <input type="checkbox"/> Variable <input type="checkbox"/> Forzada <input type="checkbox"/> Inadecuada <input type="checkbox"/>
ESFUERZO:	Excelente <input type="checkbox"/> Adecuada <input type="checkbox"/> Pasable <input type="checkbox"/> Variable <input type="checkbox"/> Mala <input type="checkbox"/>

**Figura 5.3:** Hipotética hoja de anotación.

Primero se crea la base de datos mediante phpmyadmin, que es una herramienta que trae el servidor Web XAMPP, para ello lo único que debemos hacer es acceder a phpMyAdmin mediante el panel de control de XAMMP.

Una vez abierto, se crea la **base de datos** con nombre: “**pacientetest**”, para ello en la parte superior de la página principal, entre las posibles opciones, se selecciona “Bases de datos”, a continuación se designa un nombre y dejamos el cotejamiento que viene por defecto. Una vez hecho esto se procede a crear la tabla que contendrá todos los datos que aparecen en la figura anterior, correspondientes al registro del paciente.

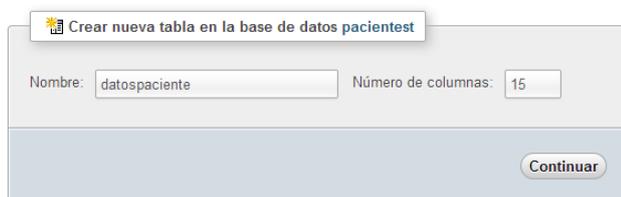
La **tabla** que contendrá los campos del registro se llama “**datospaciente**”, esto es así para diferenciarla posteriormente de otra tabla que se creará para guardar los resultados de cada test y que será llamada “**datostest**”.

El proceso para crear cada uno de los campos de la tabla es el mismo, simplemente se cambiarán las características de cada uno de ellos según los requisitos. A continuación se muestra descriptivamente el proceso de creación de la tabla, detallando solamente como se adicionan alguno de los campos, ya que el resto como hemos comentado en párrafos anteriores solo variarán el tipo, el cotejamiento o las funciones extra.

### 5.1.3 Creación y configuración de tablas en la base de datos

#### a. *Creación de la tabla:*

Se selecciona la base de datos que se acaba de crear “**pacientetest**”, (inicialmente vacía), y se procede a crear una primera tabla. La herramienta phpMyAdmin requiere que le introduzcamos el nombre de la tabla y el número de columnas que esta va a contener, se puede ver la ventana de configuración en la figura 5.4.



**Figura 5.4:** Ventana creación de tabla.

En nuestro caso para el registro de paciente, referenciando la hoja de anotación de la figura x. son necesarias 15 columnas, ya que contendrá los campos de registro que son 14 y además se deberá añadir otro para asignar un código a cada paciente, dicho código será necesario para posteriormente acceder a los datos de forma más directa y es denominado “Id\_paciente”.

Campos que debe contener la tabla “**datospaciente**”:

- 1- Id\_paciente
- 2- Dni
- 3- Nombre
- 4- Primer Apellido
- 5- Segundo Apellido
- 6- Fecha de nacimiento
- 7- Sexo
- 8- Ocupación
- 9- Nivel de estudios
- 10- Lateralidad
- 11- Edad
- 12- Problema de referencia
- 13- Antecedentes
- 14- Medicación, dosis
- 15- Observaciones

**b. Configuración de los campos:**

Una vez creada la tabla, se deben crear y configurar cada uno de los campos, el primero de estos “**Id\_paciente**” será diferente a los demás. Las características de este son:

- N° Entero
- Autoincremental
- Primario

El segundo campo será el que corresponde al DNI, en este caso será un varchar, con una longitud de 11. Para ello situado en la tabla creada en el apartado anterior debemos rellenar los requisitos de dichos campos, esto se muestra mejor mediante la siguiente figura 5.5, que es una impresión directa de lo que veríamos en el administrador de bases de datos.

Lo que vamos haciendo es ir configurando los campos que necesitamos según los requisitos. Aquí como hemos comentado anteriormente, solo se muestra la creación de los dos primeros, el resto hasta llegar a los 15 campos, se irá haciendo posteriormente con el mismo procedimiento.

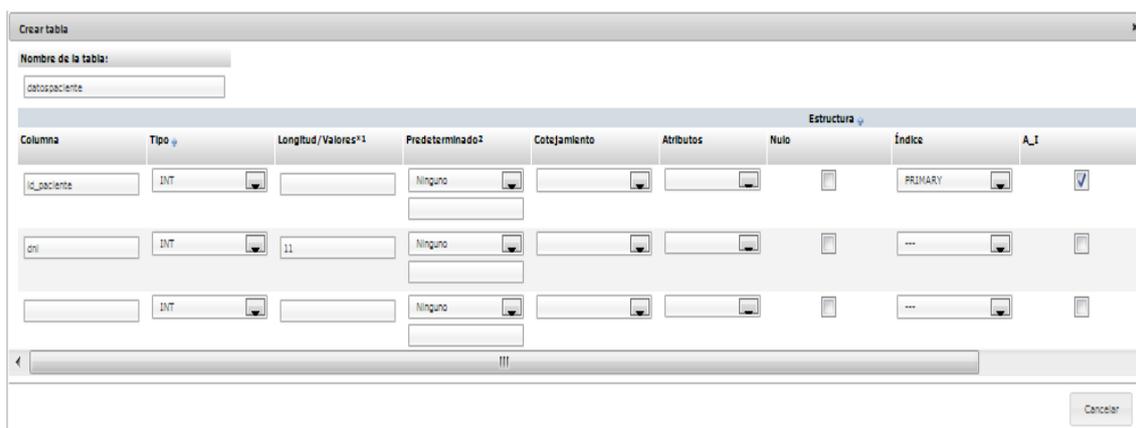


Figura 5.5: Ventana configuración campos de la tabla.

Para los restantes campos la configuración necesaria será la siguiente:

- Nombre, Primer y segundo apellido serán de tipo varchar y tamaño 50 caracteres.
- Fecha: varchar y tamaño 20 caracteres
- Sexo y lateralidad: char de un solo carácter.
- Edad: entero de tamaño once caracteres.
- Observaciones, nivel de estudios, medicación, Ocupación y Antecedentes serán del tipo text, ya que a este no tenemos que dar el tamaño, la propia herramienta se lo asigna según necesidades.

Finalmente debemos seleccionar el motor de almacenamiento InnoDB, ya que brinda muchas ventajas, como por ejemplo permitir realizar transacciones y la relación de la tabla. Una vez completos todos los campos o requisitos de cada una de las columnas, tendremos una estructura de tabla como la mostrada en la figura 5.6.

#	Columna	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input checked="" type="checkbox"/>	1 id_paciente	int(11)			No	Ninguna	AUTO_INCREMENT
<input checked="" type="checkbox"/>	2 dni	int(11)			No	Ninguna	
<input checked="" type="checkbox"/>	3 nombre	varchar(50)	latin1_swedish_ci		No	Ninguna	
<input checked="" type="checkbox"/>	4 apellido1	varchar(50)	latin1_swedish_ci		No	Ninguna	
<input checked="" type="checkbox"/>	5 apellido2	varchar(50)	latin1_swedish_ci		No	Ninguna	
<input checked="" type="checkbox"/>	6 fecha	varchar(20)	latin1_swedish_ci		No	Ninguna	
<input checked="" type="checkbox"/>	7 sexo	char(1)	latin1_swedish_ci		No	Ninguna	
<input checked="" type="checkbox"/>	8 Ocupacion	text	latin1_swedish_ci		No	Ninguna	
<input checked="" type="checkbox"/>	9 nivel_estudios	text	latin1_swedish_ci		No	Ninguna	
<input checked="" type="checkbox"/>	10 lateralidad	char(1)	latin1_swedish_ci		No	Ninguna	
<input checked="" type="checkbox"/>	11 edad	int(11)			No	Ninguna	
<input checked="" type="checkbox"/>	12 problema	text	latin1_swedish_ci		No	Ninguna	
<input checked="" type="checkbox"/>	13 antecedentes	text	latin1_swedish_ci		No	Ninguna	
<input checked="" type="checkbox"/>	14 medicacion	text	latin1_swedish_ci		No	Ninguna	
<input checked="" type="checkbox"/>	15 observaciones	text	latin1_swedish_ci		No	Ninguna	

Figura5.6: Estructura de la tabla: "datospaciente".

#### 5.1.4 Descripción de la clase que realiza la conexión a la Base de datos: ConexionMySQL

Antes de empezar con el diseño de la interface del registro del paciente, se crea una clase que se incluye en el package BaseDatos, cuya finalidad es ejecutar las consultas.

Esta es la que prepara la conexión a MySQL, ya que para realizar dicha conexión son necesarios una serie de datos que la definan o identifiquen, estos se comentan a continuación y corresponden con variables tipo string en el código de la clase.

- 1-. Nombre de la base de datos:” db = "pacientetest" (línea 6)
- 2-. Localización del conector y base datos: “url =”jdbc:mysql://localhost/”+db”
- 3-. Nombre usuario: “user=”root” (administrador, línea 8)
- 4-. Password: user=”root” (línea 9). Por defecto se tendrá una clave en blanco.

Dichos valores de usuario y password pueden ser modificados en estas líneas, pudiendo limitar el acceso a la base de datos a usuarios registrados. En este caso se deja sin password.

En la tabla 5.1 se pueden observar las librerías utilizadas así como lo expuesto anteriormente mediante el código de programación, son las primeras líneas de dicha clase.

```
1. package BasedeDatos;
2. import java.sql.*;
3. import javax.swing.JOptionPane;
4. public class ConexionMySQL {
5.
6.     public String db = "pacientetest";
7.     public String url = "jdbc:mysql://localhost/" + db;
8.     public String user = "root";
9.     public String user = "root"
10.
11.     public ConexionMySQL()
12.     {
13.
14.     }
```

**Tabla 5.1:** Clase *ConexionMySQL* (1/2)

En la siguiente parte de código de la clase (Tabla 5.2), se implementa un método que devuelve un objeto de tipo Connection llamado conectar.

En la línea 23 cargamos el driver, indicamos el driver que nos va a permitir la conexión con la base de datos mediante la ruta donde se encuentra este. Link es un objeto de tipo connection que nos va a permitir cargar o crear un objeto que va a contener el identificador de conexión, al cual le mandamos la url, el usuario y el password con el que se conectará a esa base de datos. Todo esto se ejecuta dentro de una

estructura de control try...catch., esta estructura que captura el flujo de datos en caso de error.

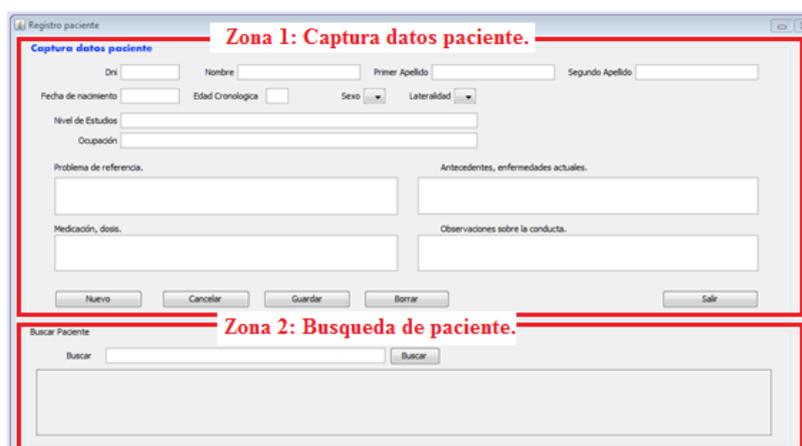
```

15. public Connection Conectar()
16.     {
17.
18.         Connection link=null;
19.         try
20.         {
21.             Class.forName("org.gjt.mm.mysql.Driver");
22.
23.             link = DriverManager.getConnection(this.url,this.user,this.pass);
24.         }
25.         catch (Exception e)
26.         {
27.             JOptionPane.showMessageDialog(null, e);
28.
29.         }
30.         return link;
    }
    
```

**Tabla 5.2:** Clase ConexiónMySQL (2/2)

## 5.2 DISEÑO DE LA INTERFAZ: REGISTRO

Esta interface esta dividida en dos zonas, una zona superior correspondientes a la captura de pacientes, y otra inferior destinada a la búsqueda de pacientes ya ingresados en la base de datos. En la siguiente figura se muestra el resultado que tendrá al final del diseño, diferenciando ambas zonas. En los siguientes apartados, se describen los pasos dados en el diseño y programación de cada una de cada una de las zonas.



**Figura 5.6:** Formulario: Registro de paciente.

### 5.2.1 Diseño de la interface de Registro: Zona 1 (Captura de Pacientes)

Una vez definida la tabla, configurada la base de datos y creada la clase auxiliar la cual va a permitir ejecutar las consultas, se procede al diseño de la interface que permite al profesional registrar los datos.

Se trabaja en el IDE, lo primero es crear una clase llamada Registro en el Package Test la cual heredará del tipo JInternalFrame. Recordamos que dichos packages no son más que carpetas que sirven para organizar el programa.

Como base de la interface del registro se selecciona una clase del tipo “JInternalFrame form”, la cual durante la ejecución del programa podremos llamar desde el menú principal, es decir desde una JFrame.

Para poder colocar los elementos necesarios sobre dicha JInternalFrame se necesita lo que llamamos en java un contenedor, en este caso un JPanel, al cual es necesario modificarle las propiedades con las que viene por defecto y darle la apariencia que nos interesa.

Accediendo a propiedades con botón derecho sobre el JPanel y seleccionando “Border” se procede a la inclusión de un borde de texto, de esta forma quedará todo más visible y mejor organizado. Las posibles opciones de tipos de bordes se muestran en la figura siguiente. En ella se puede ver el tipo de letra y características de dicho borde. Esto debe hacerse cada vez que estemos diseñando formularios para darle la apariencia deseada.

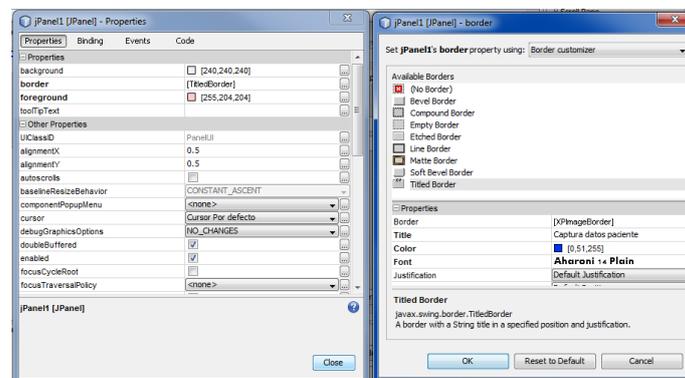


Figura 5.7: Ventana de configuración JPanel: bordes de panel.

Ahora se está en disposición de colocar los controles, como botones, campos de texto, etc. Accediendo a la paleta, dentro de las opciones de “Swing Control”, se selecciona e introduce una etiqueta asociada a cada campo de captación de datos. De los elementos de los que disponemos tenemos varias opciones, pero para que se ajuste de una forma más adecuada a las necesidades debemos recordar el tipo y cantidad de información que deben ser capaces de captar.

Haciendo referencia a las columnas de la tabla “**datospaciente**” creada en el apartado anterior (5.1), se van seleccionando cada uno de los controles de texto que van a ser requeridos. Estos son especificados a continuación según los requisitos de cada uno de ellos y de las opciones que nos permite la paleta de controles del IDE.

- Dni, Nombre, Primer apellido, Segundo apellido, Fecha de nacimiento, Edad cronológica, nivel de estudios, ocupación se usa un control tipo ”textField”.
- Problema de referencia, antecedentes, enfermedades actuales, medicación, dosis y observaciones sobre la conducta, se usa un “textArea”, ya que en estos es posible escribir descripciones más largas.
- Sexo y lateralidad, se usa un control tipo “Combo-box” que nos permita seleccionar entre varias opciones. En caso de sexo: Hombre o mujer, y para Lateralidad: Izquierda o derecha.

Una vez introducidos los controles de texto, se procede a nombrar todos los elementos anteriores. Para las etiquetas simplemente situándose en ellas y con click derecho, seleccionamos editar texto y cambiamos el texto que viene por defecto. Para estas no es necesario cambiar el nombre de variable ya que solo las estamos usando de forma gráfica.

En cuanto a los controles de texto, una vez borrado el contenido que viene por defecto, se procede a cambiar el nombre de la variable que los identifica en el código. Para quitar el texto por defecto de los combo-box se accede a propiedades y en la opción “model” se nos permite realizar esta acción, una vez vacíos procedemos a cargar los combos sexo y lateralidad.

Esto se realiza desde el código fuente, para ello situándose en dicho código y localizando el método con el mismo nombre del JFrame (Registro()) , se añaden las opciones a cada uno de los combos mediante el código mostrado en la tabla de código 5.3.

```

1. public Registro() {
2.
3.   initComponents();
4.   CargarTablaPacientes("");
5.   cboSexo.addItem("H");
6.   cboSexo.addItem("M");
7.   cboLateralidad.addItem("D");
8.   cboLateralidad.addItem("I");
9.   inhabilitar();    }
    
```

**Tabla 5.3:** *Codigo para cargar los combos en la clase Registro.*

Para las acciones, Nuevo, Cancelar, Guardar y Salir se introducen 4 botones. Dichas acciones se describen a continuación.

- Nuevo: Pulsando este se habilitarán los campos de datos, pudiéndose iniciar su captación.

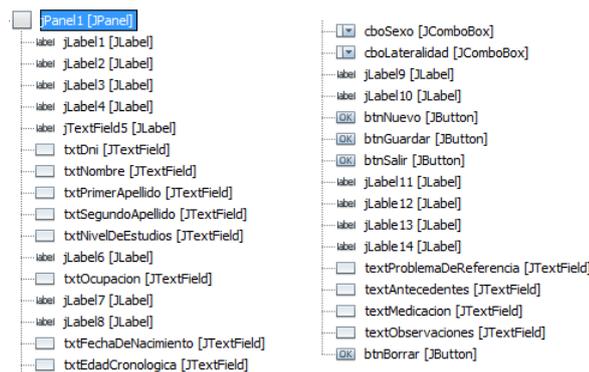
- Cancelar: Anula la creación de un nuevo paciente.
- Guardar: Envía a la base de datos todo el formulario del paciente.
- Salir: Sale del formulario de registro, es decir se cierra el JInternalFrame

Una vez introducidos los botones, también será necesario cambiar el texto que contienen por defecto, así como el nombre de la variable a la que haremos referencia en el código para referirnos a ella, al igual que se ha hecho los controles de texto.

Para nombrar las variables descritas hasta ahora, las cuales identifican en código cada uno de los elementos, se ha seleccionado ha seleccionado la siguiente nomenclatura; Consiste en colocar un sufijo diferente para cada tipo de elemento. De esta forma será más sencillo referirse posteriormente a ellos.

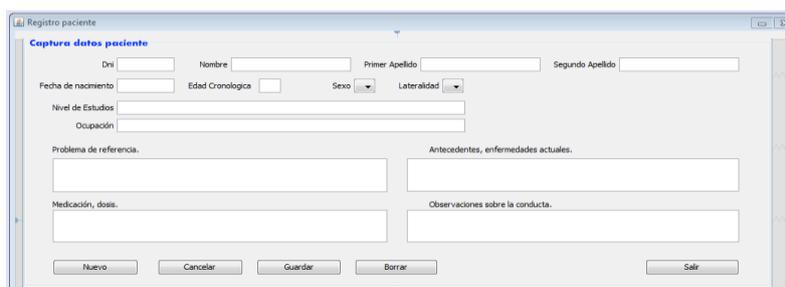
- cbo- para combo Box, así en el caso de Sexo se llamará “cboSexo”.
- txt- para campos jText, por ejemplo “txtDni”.
- btn- para las variables correspondientes a los botones, “btnGuardar”.

En la siguiente figura se muestran los elementos que forman parte de la interface y que están contenidos en el JPanel1.



**Figura 5.8:** Ventana configuración campos de la tabla.

Una vez introducidos y renombrados todos los controles swing, se procede a la programación del código que permita, tanto captar dichos datos como enviarlos a la base de datos. A continuación se van a ir mostrando y describiendo los trozos de código en orden de la programación, para así tener una visión completa del funcionamiento. Antes de esto se muestra en la figura 5.9 como queda el diseño de la interface del registro de la parte que ha sido desarrollada hasta ahora.



**Figura 5.9:** Zona I: Captura datos paciente. (Interface: Registro)

### 5.2.2 Programación botones de registro: Zona I (Captura de Registros)

➤ Botón **Salir**.

Al botón se le asocia una acción, para ello situándose encima de este y con click derecho, aparece un menú en el que se selecciona la opción Eventos, dentro de esta se selecciona el tipo concreto que viene por defecto: ActionPerformed. Una vez hecho esto, cuando se pulse sobre el botón en la aplicación, se ejecutarán las líneas que aparecen en la siguiente tabla. En este caso ha sido suficiente con añadir a esta la línea this.dispose(), la cual se encarga de cerrar la ventana interna. Tabla de código para este botón 5.4.

```

1. private void btnSalirActionPerformed(java.awt.event.ActionEvent evt) {
2.     this.dispose();
3. }
    
```

**Tabla5.4:** Configuración del botón- Salir-

Antes de proceder a programar el resto de botones, se presentan unos métodos que van a ser usados posteriormente, estos métodos son creados a continuación del constructor en el código y son “**habilitar()**” e “**inhabilitar()**”.

El método **inhabilitar** permite cargar los formularios con algunos elementos deshabilitados, es decir al pulsar salir o cancelar se deshabilitarán ciertos controles, además aparecerán los campos deshabilitarlos cuando cargue el programa inicialmente.

En la tabla de código aparece el código correspondiente a este método, en general se usan simplemente tres sentencias o métodos predefinidos. Estos son:

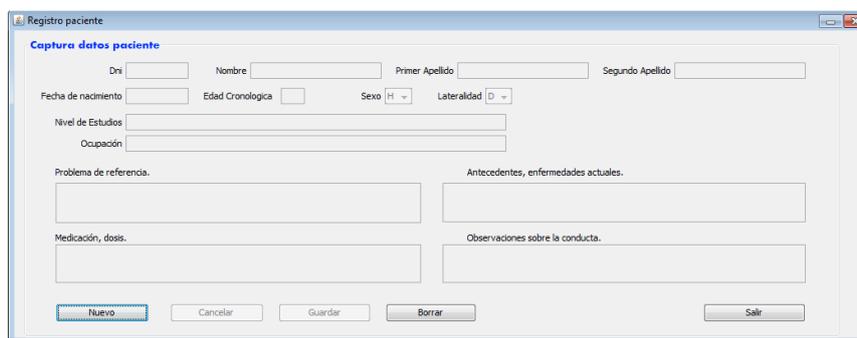
- txt.PrimerApellido.SetEnable(false), se deshabilita el campo Primer Apellido.
- txtPrimerApellido.setText(“”) se limpia el campo, de esta forma el campo aparecerá deshabilitado y en blanco.
- En las líneas finales de este código la sentencia para deshabilitar los botones es btnGuardar.setEnabled(false).

```
1. void inhabilitar()
2. {
3.     txtPrimerApellido.setEnabled(false);
4.     txtSegundoApellido.setEnabled(false);
5.     txtNombre.setEnabled(false);
6.     txtDni.setEnabled(false);
7.     txtFechaDeNacimiento.setEnabled(false);
8.     txtOcupacion.setEnabled(false);
9.     txtNivelDeEstudios.setEnabled(false);
10.    txtEdadCronologica.setEnabled(false);
11.    cboSexo.setEnabled(false);
12.    cboLateralidad.setEnabled(false);
13.    textMedicacion.setEnabled(false);
14.    textProblemaDeReferencia.setEnabled(false);
15.    textObservaciones.setEnabled(false);
16.    textAntecedentes.setEnabled(false);

17.    txtPrimerApellido.setText("");
18.    txtSegundoApellido.setText("");
19.    txtNombre.setText("");
20.    txtDni.setText("");
21.    txtFechaDeNacimiento.setText("");
22.    txtOcupacion.setText("");
23.    txtNivelDeEstudios.setText("");
24.    txtEdadCronologica.setText("");
25.    textMedicacion.setText("");
26.    textProblemaDeReferencia.setText("");
27.    textObservaciones.setText("");
28.    textAntecedentes.setText("");
29.    btnCancelar.setEnabled(false);
30.    btnGuardar.setEnabled(false);
31. }
```

**Tabla 5.5:** Metodo inhabilitar() del formulario de registro.

Compilando y ejecutando el programa, podemos ver en la figura 5.10 el efecto de este método sobre la interface.

The image shows a screenshot of a Java Swing window titled "Registro paciente". Inside the window, there is a section titled "Captura datos paciente" with several input fields. The fields are: "Dni", "Nombre", "Primer Apellido", "Segundo Apellido", "Fecha de nacimiento", "Edad Cronologica" (with a checkbox), "Sexo" (with a dropdown menu showing "H"), "Lateralidad" (with a dropdown menu showing "D"), "Nivel de Estudios", "Ocupación", "Problema de referencia.", "Antecedentes, enfermedades actuales.", "Medicación, dosis.", and "Observaciones sobre la conducta.". At the bottom of the window, there are five buttons: "Nuevo" (highlighted in blue), "Cancelar", "Guardar", "Borrar", and "Salir".

**Figura 5.10:** Efecto del método inhabilitar sobre el formulario: Registro.

En cuanto al método **habilitar()**, es similar al deshabilitar con la diferencia que en este caso al método “setEnabled()”, se le pasa como argumento “true” en vez de

“false”. De forma que cuanto se pulse “nuevo”, se habiliten y pongan en blanco todos los campos, y que los botones se habiliten. A continuación se presentan estas líneas de código para alguno de los ellos, tabla de códigos 5.6. El código completo se incluye en anexo de códigos de programa.

```

1. void inhabilitar()
2.     {
3.         txtPrimerApellido.setEnabled(true);
4.         cboSexo.setEnabled(true);
5.         textMedicacion.setEnabled(true);
6.         //Limpiamos los campos cuando se habilitan
7.         txtPrimerApellido.setText("");
8.         textProblemaDeReferencia.setText("");
9.         btnCancelar.setEnabled(false);
10.    }

```

**Tabla5.6:** Método habialitar() del formulario:Registro.(Solo algunos campos)

➤ **Botones Nuevo y Cancelar.**

Al pulsar el botón **Nuevo** se habilitan todos los campos, siendo el encargado de iniciar la captación de los datos referidos a un nuevo sujeto. El código correspondiente a la programación de este se completa con una llamada al método habilitar() y con la asignación de la cadena “Insertar a la variable acción” y la cadena “ “ a la variable idactualizar. El cometido de estas se ve más avanzado este capítulo. De la misma forma pero llamando al método inhabilitar se programará el botón **Cancelar**.

A esto botones, al igual que todos irá asociado un evento, que se crea como se ha descrito anteriormente. A continuación en la tabla 5.7, se presenta la programación de estos botones.

```

1. private void btnNuevoActionPerformed(java.awt.event.ActionEvent evt) {
2.     habilitar();
3.     accion="Insertar";
4.     idactualizar=" ";}
5.
6. private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {
7.     inhabilitar();
8. }

```

**Tabla 5.7:** Configuración de los botones Nuevo y Cancelar.

En el ingreso de datos sobre los campos de texto contenidos en un panel, es posible desplazar el cursor de uno a otro pulsando la tecla <tab> ó <ctrl-tab>. Sin

embargo es interesante añadir la cualidad de que sea el botón <enter> el que permita dicho desplazamiento.

Para conseguir esto, se dispone del método *transferFocus()*. Llamando a este en cada uno de los componentes (campos de texto, combos...), el foco se transmitirá al siguiente componente del panel que pueda aceptar dicho foco. Este proceso se debe realizar en el mismo orden en el que posteriormente se realice el avance en la introducción de los datos. La programación se debe ir haciendo en el orden que interesa, es decir: primero Dni, segundo Nombre, tercero Primer apellido, etc.

Esto es mostrado en la tabla 5.8, donde aparece el código necesario para llamar a este método. Para asignar el evento al campo, como hemos comentado anteriormente botón derecho>> Eventos>> Action >> actionPerfomed y nos lleva directamente al código ejecutado tras dicho evento.

```
1. private void txtNombreActionPerformed(java.awt.event.ActionEvent evt) {  
2.     txtNombre.transferFocus();  
3. }
```

**Tabla 5.8:** Inclusión del método *transferFocus()*, en el campo de texto Nombre..

“Esto se hace para cada uno de los campos de texto y de los combos en el orden que se van a ir ingresando los datos”

### ➤ Botón **Guardar**

Encargado de dar la orden de ingresar los datos en la base de datos. Al pulsar el botón guardar, no se seguirá el mismo procedimiento en caso de que el almacenamiento de datos corresponda a un nuevo registro o a una modificación en un registro previamente almacenado. Dentro de la programación de este botón se debe diferenciar entre ambas opciones, para ello se crea la variable global auxiliar, tipo string, denominada acción, la cual podrá ser cargada con la cadena “insertar” en el caso de haber pulsado el botón Nuevo (como se vio en el punto de su configuración), con el fin de continuar en el guardado de un nuevo sujeto. Posteriormente cuando se proceda a su modificación desde la tabla en la Zona II de este formulario (Zona de Búsqueda), se verá que a la variable acción se le asignará la cadena “modificar”, y así dentro del código del botón guardar se procederá de otra forma. Es decir, seguirá otras instrucciones, otro código.

El código incluido en las siguientes tablas corresponde a la programación del botón “guardar”, pero solo incluyendo el guardado de un campo de texto y el contenido de un combo. Para el resto de datos a almacenar se realiza de la misma forma, pero no se incluyen en el código por no hacerlo demasiado extenso para su descripción. El código completo se puede revisar en el anexo correspondiente de códigos de programa. Dicho proceso es el siguiente:

- 1- Se instancia la clase creada anteriormente que contiene el método usado para conectar a la base de datos: "ConexionMySQL".(Línea 3)
- 2- Creamos un objeto de tipo "connection" con el que nos vamos a conectar a la base de datos llamado "cn", se asigna al objeto con el que instanciamos la clase. (Línea 4)
- 3- Para poder usar este método debemos importar el paquete java.sql, para ello debajo de la línea inicial del código importamos este mediante la palabra reservada "import". Esta parte de código no está incluida en la tabla de código siguiente.
- 4- Se crea una variable llamada "sSQL" de tipo string y se inicializa en blanco, donde posteriormente guardaremos una sentencia sql. (línea 6), también deben ser creadas las variables locales que van a contener los las cadenas del campo de texto dni y del combo sex. (estas son las seleccionadas para mostrar el funcionamiento de la clase)
- 5- Se procede a capturar los valores presentes en los campos y en los combos (persianas desplegadas con diferentes opciones), antes de esto se deben haber declarado las variables que los contendrán. Una vez capturados dichos valores, serán almacenados en las variables correspondientes.

Para capturar texto se usa el método "getText()". En el combo esto se hará de forma diferente, al ser una lista desplegable es necesario usar el método getItem(). Como este devuelve un elemento de tipo objeto, para poder almacenarlo en una variable de tipo cadena, antes debe ser casteado mediante el método toString(). (líneas 10 y 11)

```

1. private void btnGuardarActionPerformed(java.awt.event.ActionEvent evt) {
2.
3. ConexionMySQL mysql = new ConexionMySQL();
4. Connection cn = mysql.Conectar();
5.
6. String sSQL = "";
7. String mensaje = "";
8. String dni,sex;

9.
10.     dni = txtDni.getText();
11.     sex = cboSexo.getSelectedItem().toString();

```

**Tabla 5.9:** Código botón Guardar (1/)

- 6- En la siguiente tabla de código del botón guardar, inicialmente aparece una sentencia "if-else", esta se encarga de comprobar si la opción a realizar es la de insertar datos o la de modificarlos. Para ellos se utiliza el método equal(), el cual comprueba el contenido de la variable acción.(Línea 12). Si en la variable acción viene cargada la cadena "Insertar" se proce al código de la tabla 5.10.

```
12. if(accion.equals("Insertar")){
13. sSQL = "INSERT INTO datospaciente( dni,sexo) VALUES(?, ?)";
14.
15. mensaje = "Los datos se han insertado de manera satisfactoria...";
```

**Tabla 5.10:** Código botón Guardar (1/)

- Se define la cadena sSQL, para ello se usan las clausulas “*INSERT INTO*” que permiten introducir datos y colocarlos en cada uno de los campos que vamos a insertar.
  - Se inicializa una variable string llamada “mensaje” que posteriormente contendrá el mensaje que indicara si ha sido correcta o no la inserción de los datos.
- 7- En caso de que la variable acción haya sido cargada con la cadena de caracteres “Modificar” (Mediante la asignación en el evento correspondientes a la modificación de registros), el código seguido es el mostrado en la tabla 5.10.

```
16. else if(accion.equals("Modificar")){
17. sSQL="UPDATE datospaciente(dni,sexo)" WHERE id_paciente = idactualizar;
18.
19. mensaje = "Los datos se han Modificado correctamente";
20. }
```

**Tabla 5.11:** Código botón Guardar (2/4)

- Se define la cadena sSQL, para ello se usan las clausulas “*UPDATE*” que permiten recuperar datos y colocarlos en cada uno de los campos que vamos a modificar., para ello mediante la cláusula WHERE asigna el número de paciente almacenado en la variable id\_paciente (correspondiente al campo de formulario id) a la variable idactualizar. Este el criterio de búsqueda para obtener los datos almacenados en dni y sexo en este caso. El proceso completo se puede revisar en el anexo correspondiente a códigos de programa.
- 8- A continuación se realizan los procedimientos que vamos a usar para insertar los datos en la base de datos, para ello se utiliza la interface llamada PreparedStatement, de la forma pst =objeto cn, como esa sentencia es muy recurrente, se ejecuta dentro de una estructura de control try...catch, para capturar el flujo de datos en caso de error.
- 9- Ingresamos cada uno de los datos en esta cadena, es decir se van a introducir los valores que ya están contenidos en las variables en los signos de interrogación. Para esto se usan el objeto pst y el método setString; Al que se le pasa como argumento el número de la posición del elemento que se va a insertar y el valor a insertar el cual está contenido en la variable.

```
21. try {
22.
23.     PreparedStatement pst = cn.prepareStatement(sSQL);
24.
25.     pst.setString(1,dni);
26.     pst.setString(6,sex);
27. }
```

**Tabla 5.12:** Código botón Guardar (3/4).

10-Para finalizar y poder ejecutar esa consulta, antes se crea la variable n, cuya función es saber si se insertaron los datos. Si n es mayor que uno se insertaron de forma correcta y si n es cero no se insertaron. Es decir “n” almacena el contenido de la ejecución de la consulta. El catch nos permite capturar el error que se genera en su caso. En cuanto a la línea que instancia o llama al método “CargarTablaPacientes”, se verá en la siguiente parte del desarrollo del código de la interface de Registro

```
28. int n = pst.executeUpdate();
29. if(n > 0){
30.     JOptionPane.showMessageDialog(null,mensaje);
31.     CargarTablaPacientes("");
32. }
33. }
34. catch (SQLException ex) {
35.     JOptionPane.showMessageDialog(null, ex);
36. }
```

**Tabla 5.12:** Código botón Guardar (4/4)

Hasta el momento se ha creado la interface para el ingreso de los datos del paciente. A continuación se muestra un ejemplo gráfico con la captación de un paciente y el ingreso de los datos personales en la base de datos. La finalidad de esto es mostrar el correcto funcionamiento de lo que se ha desarrollado hasta este punto.

Una vez compilada la clase Registro() en el IDE de Netbeans, debe aparecer la ventana mostrada en la siguiente figura, entonces pulsado el botón “Nuevo” se habilitan los campos, momento en el que se está en disposición de introducir los datos correspondientes al ejemplo. Tras ser introducidos dichos datos se procede a su almacenamiento.

**Registro paciente**

**Captura datos paciente**

Dni: 48498789    Nombre: Francisco    Primer Apellido: Liusan    Segundo Apellido: Martinez

Fecha de nacimiento: 09-02-1985    Edad Cronologica: 28    Sexo: H    Lateralidad: D

Nivel de Estudios: Primarios y Fp

Ocupación: Mecánico en Automoción.

Problema de referencia: problemas con la memoria

Antecedentes, enfermedades actuales: ninguna

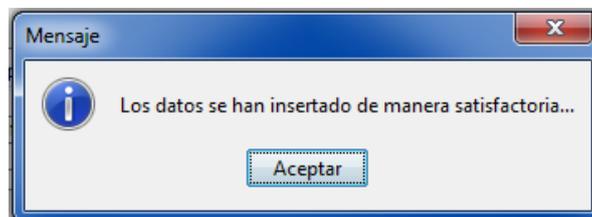
Medicación, dosis: no mantiene medicacion, en la actualidad

Observaciones sobre la conducta: su conducta es correcta

Botones: Nuevo, Cancelar, Guardar, Borrar, Salir

**Figura 5.11.:** Captación de datos personales y médicos de un nuevo sujeto

Tras pulsar el botón guardar si todos los datos se han sido insertados de forma correcta aparecerá en pantalla el siguiente mensaje.



**Tabla:** Captación de datos personales y médicos de un nuevo sujeto

Finalmente queda por comprobar que estos datos han pasado a la base de datos, para ello accediendo al administrador de base de datos MySQL, seleccionando la base de datos “patientest” y dentro de esta la tabla “datospaciente”, aparecerá ya su inserción. Esto se ve reflejado en la siguiente figura 5.12, que es una captura de pantalla del administrados de MySql.

```
SELECT *
FROM `datospaciente`
LIMIT 0 30
```

Perfilando [En línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar: 30 fila(s) iniciando en la fila # 0 en modo horizontal y repetir los encabezados cada 100 celdas

Ordenar según la clave: Ninguna

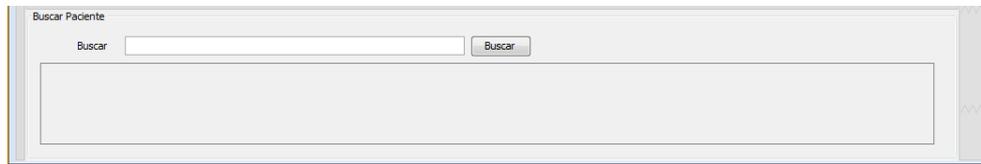
	id_paciente	dni	nombre	apellido1	apellido2	fecha	sexo	Ocupacion	nivel_estudios	lateralidad	edad	problema	antecedent
	18	48498789	Francisco	Liusan	Martinez	09-02-1985	H	Mecánico en Automoción	Primarios y Fp	D	28	problemas con la memoria	ninguna

**Figura 5.12:** Captación de datos personales y médicos de un nuevo sujeto.

### 5.2.3 Diseño de la interface de Registro: Zona 2 (Búsqueda)

En este apartado se crea la zona de la interface para llevar a cabo consultas u modificaciones en los datos de pacientes que han sido ingresados con anterioridad. Los métodos y estructuras elegidas para el diseño de la búsqueda, siempre van a ser los mismos, con esto se quiere aclarar que solo van a ser descritos en este capítulo, y cuando sean utilizados en capítulos posteriores se hará referencia a este apartado.

El contenido de esta se colocará sobre un nuevo panel (jpanel2), el cual estará situado debajo del de captura de paciente, el título de este nuevo panel será Buscar Paciente y el procedimiento para insertarlo en el Frame, será similar al seguido en el panel de captura descrito en el apartado anterior.



*Figura 5.13: Diseño zona de búsqueda en la interface de Registro.*

A continuación se incluirán los elementos necesarios para realizar la consulta, estos son:

- 1- Una tabla para la navegación entre los diferentes sujetos.
- 2- Etiqueta y campo de texto donde se ingresa el criterio de búsqueda.
- 3- Botón buscar, para que una vez ingresado el criterio, proceda a cargar en la zona de captura todos los datos almacenados en la base de datos relativos a este.

Una vez hecho el diseño del segundo panel o panel de consulta, queda de la forma mostrada en la figura 5.13 anterior. En este caso no se describe como se han insertado la etiqueta, el campo de texto y el botón; Ya que el proceso, es similar al descrito en el apartado anterior.

A continuación se describe la configuración de la tabla para que realice su función especificada, así como el código destinado a que tras el ingreso del criterio y pulsar el botón “Buscar”, aparezcan en la zona superior todos los datos almacenados correspondientes a dicho sujeto.

Pasos:

- 1- Se inserta un panel (contenedor), en este caso jpanel2, seguidamente se configura, ajustando al tamaño adecuado, añadiendo un borde y nombrándolo como “Buscar Paciente”.
- 2- Se inserta un elemento de la paleta llamado tabla, (jTable)
- 3- Se procede a configurar la tabla. Como viene con un contenido por defecto se debe borrar dicho contenido. Sobre la tabla con botón derecho aparece la cortina desplegable correspondiente a las propiedades de la misma, accediendo a contenido tabla aparece una ventana de diálogo, y en la pestaña columnas dentro de este nos permite la opción eliminar columna. Se debe nombrar la variable correspondiente a la tabla, el nombre de este txtTablaPaciente.
- 4- Insertar etiqueta, campo de texto y botón. Configurar estos según los pasos seguidos anteriormente. A la variable del campo de texto se le asigna el nombre txtBuscar y al botón btnBuscar, esto es así según la nomenclatura descrita en el capítulo anterior. Una vez hecho esto tenemos totalmente configurada la apariencia de la interface Registro, quedando de la forma mostrada en la siguiente figura.

**Figura5.14:** Interface del formulario: Registro

- 5- Además de los elementos incluidos hasta el momento, se procede a insertar un menú emergente de Selección-Modificación sobre la tabla de búsqueda, que aparezca dando click derecho al ratón. Este se puede ver desplegado en la figura 5.15.

id_paciente	DNI	Nombre	P.Apellido	S.Apellido	Fecha de nac.	Sexo
7	23232323M	Sergio	Almoradi	Fernandez	08-03-1960	H
8	6789788C	Fuentsanta	Perez	Alcantara	23-01-2001	H
9	99999999Z	Julia	Zam	Selecc	13-12-1970	M
12	23232323F	Francisco	Alegria	busman	09-06-1982	H
13	45454545N	Luis	Alisado	Pelsan	09-09_-999	H
14	78787878I	Luis	Alicado	Barona		H

**Figura5.15:** Menú emergente: Seleccionar-Modificar.

La función de pulsar esta opción en el menú emergente, consiste en seleccionar un registro de paciente y proceder a su modificación, cargando los datos personales y clínicos correspondientes a este, sobre los campos de captación de datos situados en la zona superior del formulario.

En este punto del capítulo solo se muestra como se inserta este elemento en la interface, en el siguiente se describirá la programación que ejecuta esta función.

Para crear el menú emergente este debe ser introducido desde la paleta como el resto de componentes. En este caso se utiliza el componente `jPopupMenu`.

Una vez introducido se procede a su configuración. Para ello, en el Navegador de elementos, pulsando sobre el `JPopupMenu` con el botón derecho, se añadirán los elementos del menú. En este caso `Seleccionar-Modificar`. A este elemento o Item del Menú se le llama con el nombre `mnEditar` y se le coloca el texto en la propiedad `text` >> “Seleccionar modificar”

Finalmente se debe asociar este `jPopupMenu` a la `tblConsultaPaciente`, entrando en las propiedades de esta última y seleccionándolo en la opción “`componentPopupMenu`” según figura 5.16.

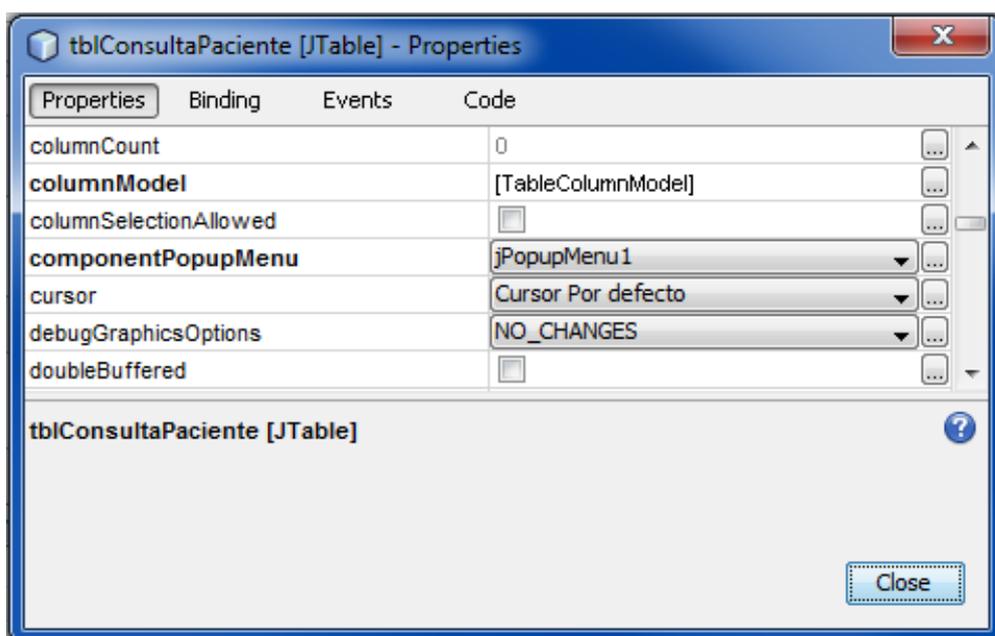


Figura5.16: Ventana propiedades de la tabla: tblConsultaPaciente.

#### 5.2.4 Programación de los elementos de la interface de Registro: zona 2 (Búsqueda)

➤ **Botón Buscar:**

El examinador introduce los criterios de búsqueda en el campo de texto denominado txtBuscar, tras esto dichos datos son captados para realizar la búsqueda de coincidencias en la base de datos. En las siguientes líneas de código está implementada la acción ante el evento de pulsar este botón.

Previamente se habrá creado una variable tipo string llamada “valor” en la que se captura el contenido que en ese momento está en el campo de texto txtBuscar, que corresponderá con el criterio de búsqueda. A continuación se realiza una llamada al método CargarTablaPacientes() pasándole como argumento el contenido de esa variable (criterio de búsqueda).

```

1. private void btnBuscarActionPerformed(java.awt.event.ActionEvent evt) {
2.     String valor = txtBuscar.getText();
3.     CargarTablaPacientes(valor);
4. }
    
```

Tabla 5.13: Código botón Buscar (1/4)

Método *CargarTablaPacientes(String valor)*

Este método permite buscar o comprobar si existe un paciente en la base de datos. Para ello recibirá como criterio de búsqueda el contenido de la variable valor siendo este: el dni, nombre, apellidos o combinación de estos, que hayan sido introducidos en el campo de texto con la etiqueta “Introducir criterio de búsqueda”.

A continuación el método realiza una conexión a la base de datos, comprobando si existen coincidencias con los criterios; En caso afirmativo, recuperará los datos personales del mismo y los organizará sobre una tabla. De esta forma, podrá ser seleccionado para mostrar las pruebas que hayan sido registradas en la aplicación con anterioridad.

Se debe tener en cuenta que durante el desarrollo del proyecto, este método se va a utilizar, para obtener en la tabla diferentes registros según el formulario en el que se esté ejecutando. Para la descripción del código siguiente se han tenido en cuenta los campos, id\_paciente, dni, nombre, apellido1, apellido2, fecha y sexo; Que son los utilizados en el formulario registro.

```
1. void CargarTablaPacientes(String valor)
2.     {
3.     Stringtitulos[]={ "id_paciente", "DNI", "Nombre", "P.Apellido", +
4.                       "S.Apellido", "Fecha de nac.", "Sexo"};
5.     String registro[] =new String[7];
6.     String sSQL=" ";
7.     modelo = new DefaultTableModel(null, titulos);
8.
9.     ConexionMySQL mysql = new ConexionMySQL();
10.    Connection cn = mysql.Conectar();
11.
12.    sSQL = "SELECT id_paciente,dni, nombre, apellido1, apellido2, fecha,
13.          sexo FROM datospaciente "
14.          + "WHERE CONCAT(dni, ' ',nombre, ' ', apellido1, ' ', apellido2) LIKE
15.          '%" +valor+"%'";
16.
17.    try
18.    {
19.        Statement st = cn.createStatement();
20.        ResultSet rs = st.executeQuery(sSQL);
21.
22.        while(rs.next())
23.        {
24.            registro[0] = rs.getString("id_paciente");
25.            registro[1] = rs.getString("dni");
26.            registro[2] = rs.getString("nombre");
27.            registro[3] = rs.getString("apellido1");
28.            registro[4] = rs.getString("apellido2");
29.            registro[5] = rs.getString("fecha");
30.            registro[6] = rs.getString("Sexo");
31.
32.            modelo.addRow(registro);
33.        }
34.
35.        tblConsultaPaciente.setModel(modelo);
36.    }
37.    catch (SQLException ex)
38.    {
39.        JOptionPane.showMessageDialog(null, ex);
40.    }
```



**Tabla 5.14:** CódigoMetodo CargarTablaPacientes.

Descripción del código: (Tabla 5.14)

- 1- Crea a los títulos de la tabla: para esto se crea un vector tipo string llamado títulos, en este caso la tabla dispondrá de los elementos o columnas que aparecen en el párrafo anterior(línea 3)
- 2- Crea otro vector tipo string llamado “registro” para agregar cada uno de los elementos a la tabla. Este estará inicializado con una cantidad de columnas requeridas que coincide con los títulos. (línea 5)
- 3- Se crea una variable llamada “sSQL” de tipo string y se inicializa en blanco, donde posteriormente guardaremos una sentencia sql. (línea 6).
- 4- Se Colocan los títulos al modelo tabla, para esto creamos una variable modelo y se le asigna el objeto DefeaultTableModel y se le envían los títulos. Una vez hecho esto , configurado el modelo de tabla. (línea 7)
- 5- Se instancia la clase creada anteriormente que contiene el método usado para conectar a la base de datos: “ConexionMySQL”.(Línea 9 )
- 6- Creamos un objeto de tipo “connection” con el que nos vamos a conectar a la base de datos llamado “cn”, se asigna al objeto con el que instanciamos la clase. (Línea 10)
- 7- Se definirá la cadena sSQL que contendrá la llamada a la base de datos, la cual nos proporcionara el contenido de los campos que hayan sido indicados tras la clausula SELECT , estos deben tener el mismo nombre con el que aparecen en la base de datos, a continuación mediante FROM se selecciona la tabla de búsqueda, en este caso “datospaciente”

Mediante el WHERE CONCAT que trae el motor de búsqueda de SQL, concatenamos los elementos Dni, Nombre, Apellido1 y Apellido2 para que cuando se busque se utilicen estos como criterio. Utilizando el LIKE, se consigue que todo lo contenido en la variable “valor” (criterio de búsqueda) sea comparado con la concatenación.

Para lo que se busque pueda estar en cualquier posición de la cadena, se pone +valor+. De esta forma si el criterio de búsqueda es el nombre lo buscará en cualquier posición del contenido de la cadena. (líneas 12 y 13)

- 8- A continuación se realizan los procedimientos que vamos a usar para rescatar los datos de la base de datos: este proceso se realiza dentro de un try..catch para poder reportar la excepción en su caso.

Se crea un stamento con la interface “Statement”, llamado st. (línea 17). Se almacenan los datos en un objeto resulset, llamado rs. Para esto se usa la función `executeQuery`, a la cual es mandada una cadena sSQL como argumento, así en este objeto obtenemos todos los datos. (línea 18)

Ahora se debe recorrer la matriz que contiene el objeto resulset para poder rescatar los datos que nos interesen. Esto se consigue mediante un ciclo `while` y el método `next()` para pasar de elemento en elemento hasta llegar al final.

Para obtener estos valores se creo una variable tipo string que a vu vez es un vector: `registro[]`. Se procede a almacenar cada una de las posiciones de ese vector mediante el método `getString()`. En las líneas de la 22 a la 28. A este método se le pasa como argumento el nombre de la columna en la que están almacenados en la base de datos y el resultado es recuperado en cada una de las posiciones del vector `registro`.

- 9- Una vez recuperados estos datos, debemos almacenarlos en el modelo de tabla que hemos creado con el método `addRow()`. (fila 30) Tras el `while`, se agregan los datos a la tabla para que los muestre mediante el método `setModel()` de la forma: `tblConsultaPaciente.setModel(modelo)`. (fila 33)

Se finaliza el `try-catch`, en el `catch` tratamos la excepción mediante la muestra de un mensaje mediante `JOptionPane.showMessageDialog(null,ex)`.

Nota: Para que cuando se inicie el formulario carguen al mismo tiempo directamente los datos en la tabla, lo que hacemos es una llamada al método `cargarTablaPaciente()` y se le manda una cadena en blanco. Esto se hace en el constructor.

#### ➤ **Menú emergente: Seleccionar-Modificar.**

En este apartado se describen los métodos necesarios para que: al presionar `click` derecho y pulsar en el ítem “seleccionar-modificar” del menú emergente, sea capturado el id del registro seleccionado para modificarlo, y este id sea enviado a un método que buscará esos datos en la base de datos y los asignará a los campos correspondientes del formulario de registro.

Este método será similar al `CargarTablaPacientes()`, pero con algunas modificaciones.

Aunque en este caso los campos a recuperar son todos los de la tabla “`datosPaciente`”, para demostrar el funcionamiento, solo se va a mostrar el código correspondiente a los campos: `dni` y `sexo`. En este caso no se hace una descripción tan minuciosa, simplemente se comenta las variaciones respecto a la programación anterior.

- 1- No se van a necesitar las variables tipo string `dni`, `nom...` ya que ahora no vamos a cargar en una tabla sino que los vamos a cargar en campos de texto, tampoco se va a utilizar el modelo de tabla.
- 2- Se modifica el `where` para que busque lo que tenga la variable `id`.

- 3- Por tanto ahora en el while ya no vamos a utilizar un vector sino unas variables, para ello las hemos creados al inicio del método. Por tanto en vez de cargar en un vector cargamos en variables individuales.
- 4- En este caso no existirá modelo de tabla.
- 5- Mediante setText(), o asignar texto conseguimos mandar el contenido de cada uno de las variables a los campos de texto correspondientes

```
1. void BuscarPacienteEditar(String id)
2.     {
3.         String sSQL=" ";
4.         String dni="", sex="";
5.
6.         ConexionMySQL mysql = new ConexionMySQL();
7.         Connection cn = mysql.Conectar();
8.
9.         sSQL = "SELECT id_paciente, dni, ,sexo, FROM datospaciente"
10.        + "WHERE id_paciente = " +id;
11.        try
12.        {
13.            Statement st = cn.createStatement();
14.            ResultSet rs = st.executeQuery(sSQL);
15.
16.            while(rs.next())
17.            {
18.                dni = rs.getString("dni");
19.                sex = rs.getString("sexo");
20.
21.            }
22.
23.            txtDni.setText(dni);
24.
25.            idactualizar=id;
26.        }
27.        catch (SQLException ex)
28.        {
29.            JOptionPane.showMessageDialog(null, ex);
30.        }
31.    }
```

*Tabla 5.15: CódigoMetodo : BuscarPacienteEditar()*

Finalmente queda relacionar el método anterior con el correspondiente ítem del menú. Para ello se procede a programar un evento de la siguiente forma mnEditar>>Eventos>>Action>>ActionPerformed, y ahora se programa las acciones tras el evento.

```
1.     private void mnEditarActionPerformed(java.awt.event.ActionEvent evt) {
2.
3.         int filasel;
4.         String id;
5.
6.         try{
7.             filasel = tblConsultaPaciente.getSelectedRow();
8.             if(filasel!=-1){
9.                 JOptionPane.showMessageDialog(null,"Nosehaseleccionado ninguna fila");
10.            }
11.            else
12.            {
13.                accion = "Modificar";
14.                modelo = (DefaultTableModel) tblConsultaPaciente.getModel();
15.                id = (String) modelo.getValueAt(filasel,0);
16.                habilitar();
17.                BuscarPacienteEditar(id);
18.            }
19.        }
20.    catch(Exception e)
21.    {
22.        }
23.
24.    }
25.
26.
```

*Tabla 5.16: Programación del evento en el Item mnEditar.*

Se utiliza un try...catch para las excepciones, dentro de este es capturada la fila seleccionada mediante getSelectRow(). Antes es necesario preguntar si esta fila ha sido seleccionada, esto se hace mediante una sentencia if-else. Con dicha sentencia se comprueba si la variable “sel” es igual a -1, en cuyo caso no se habrá seleccionado ninguna fila, momento en el que se mostrará mediante un panel de diálogo, el mensaje: “no se ha seleccionado ninguna fila”.

Si se ha seleccionado alguna fila pasaría a ejecutarse el código dentro del “else” de la estructura. Con anterioridad se creó una variable tipo cadena, denominada “acción”, cuya función era diferenciar entre si se procedía a ingresar un nuevo registro o a modificarlo. En este caso como se desea llevar a cabo una modificación, a dicha variable es asignada la cadena “Modificar”.

Ahora obtenemos los datos de la fila seleccionada en la variable “modelo”. Como solo es necesario el contenido de la columna id\_Paciente. Para asignar este a la variable id que será el argumento del siguiente método a utilizar, se hace un cast de string y se utiliza el método getValueAT(). Lo que venga de la columna 0 de la fila seleccionada se convierte a string y se almacena en id.

El contenido de la variable “id” es mandado como argumento al método BuscarPaciente() creado anteriormente, que servía para buscar datos en la base de datos y mandarlos a los campos de texto en la zona superior, y así poder proceder a su modificación

## CAPÍTULO 6

### **FORMULARIO DE PREPARACIÓN E INICIO TEST**

---

---

En este capítulo se desarrolla el funcionamiento de la interface que permite la preparación, inicio, finalización, mostrado de resultados y almacenamiento de los mismos en la base de datos. (Del test ejecutado)

En el proceso descrito en el capítulo dos, para la aplicación de la prueba de forma correcta, se explica que eran necesarias unas operaciones previas para el desarrollo posterior del test de forma adecuada, así como para obtener unos resultados compatibles con los baremos seleccionados.

Una vez que ha sido registrado en paciente en la base de datos, se procede a la ejecución del test. Para ello se ha creado en el programa el denominado formulario de Inicio.

Dicho formulario contiene campos a rellenar antes de la ejecución del test y campos a rellenar tras la ejecución del mismo. Por tanto se pueden considerar 3 etapas:

- 1- *Etapa Pre-Test*: Búsqueda del sujeto a evaluar e ingreso de datos iniciales.
- 2- *Ejecución del test* por el sujeto.
- 3- *Etapa Post-Test*: El examinador completa datos relativos al comportamiento del sujeto durante la ejecución de la prueba y finaliza el proceso visualizando y guardando los resultados.

#### **6.1 INTERFACE: INICIO**

En el diseño de esta se ha optado por dividir la pantalla en dos zonas, una izquierda o formulario de inicio donde se localiza al paciente que va a realizar la prueba y se introducen los parámetros de configuración necesarios para obtener unos resultados adecuados. La zona derecha o zona de respuestas, está constituida por cuatro columnas, las cuales serán auto-rellenadas con las respuestas del test tras la finalización del mismo.

El contenido de las columnas, es el mismo que va a aparecer en el historial del paciente y será descrito en el siguiente capítulo. A continuación se puede observar la pantalla de inicio del test en la figura 5.1. Esta figura muestra cómo va a quedar el formulario de inicio tras su diseño y programación

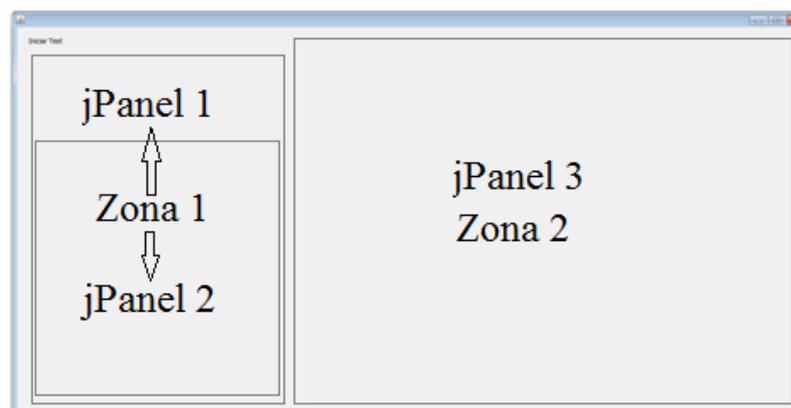
The screenshot shows a Java Swing window titled "Iniciar Test". On the left, there is a search bar "Buscar Paciente" with a "Buscar" button. Below it is a table with columns "DNI", "Nombre", "P.Apellido", and "S.Apellido". The table contains three rows of patient data. Below the table is a "Seleccionar Sujeto" button. The "Test" section includes fields for "NOMBRE", "DNI", and "Fecha Test", along with an "EXAMINADOR" field. There is a "BARENO DE PUNTAJACIÓN" dropdown menu set to "General". Under "Instrucciones", there are radio buttons for "Hombre", "Mujer", and "Desactivar", with an "INICIAR" button. Below this are dropdown menus for "RELACION", "COOPERACION", and "ESFUERZO", all set to "Excelente". There is an "OBSERVACIONES" text area and a "GUARDAR RESULTADOS TEST" button at the bottom. The right side of the window is a large, empty grid area for displaying results.

**Figura 6.1.** Formulario de inicio del test y zona de resultados

A continuación se describe el diseño para este formulario, dividiendo dicha descripción en las zonas que los componen, así como en el proceso seguido para su creación. El diseño vendrá acompañado de los métodos principales utilizados para el funcionamiento del mismo.

El primer paso seguido consiste en crear la clase que implementa este formulario en el Package Test, la cual heredará del tipo `JInternalFrame`.

Esta “`JInternalFrame form`”, va a ser la base del formulario de Inicio, podrá ser llamada desde el menú principal durante la ejecución, es decir desde una `JFrame`. (MenúPrincipal). Para poder colocar los elementos necesarios sobre dicha “`JInternalFrame`” se disponen los contenedores tipo `jPanel` según se muestra en la siguiente figura. Los bordes de estos paneles deben ser modificados al igual que se hizo en formulario de Registro.



**Figura 6.2.** Zonas del formulario de inicio, implementadas sobre elementos `jPanel`.

Los paneles 1 y 2 van a constituir el formulario de Inicio, el `jPanel 3` contendrá la hoja de respuestas, formada por cuatro columnas de 32 filas cada una.

**6.1.1 Zona 1: Formulario de inicio.**

➤ ETAPA PRE-TEST:

Como se ha comentado en el primer apartado, antes de realizar el test se deben haber completados las operaciones de:

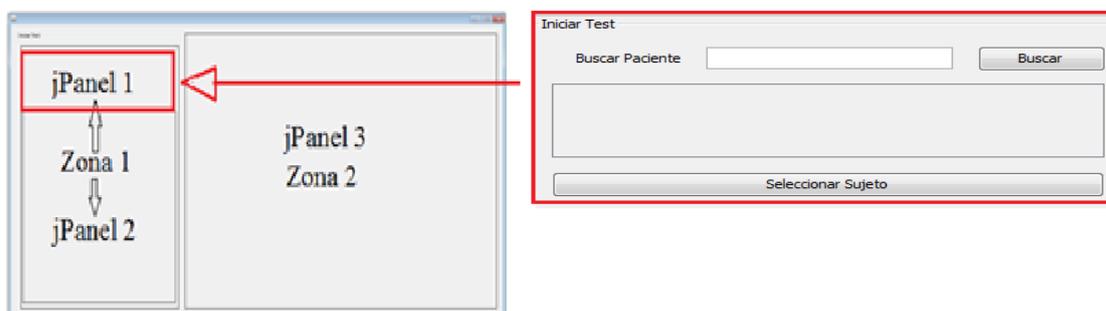
- 1- Selección del sujeto examinado.
- 2- Ingreso del nombre del Examinador si procede.
- 3- Selección del modo de instrucción.
- 4- Selección de la baremación.

Para la descripción del diseño y programación de esta zona, se siguen los puntos anteriores.

**1- Selección del sujeto examinado.**

Para iniciar un test y asignarlo a un paciente, es necesario buscar dicho paciente en la zona superior del formulario, una vez localizado debe ser marcado sobre la tabla y a continuación pulsando el botón “seleccionar paciente” los datos son trasladados a los campos inferiores.

La única diferencia respecto al proceso seguido anteriormente, que ahora es un botón el que permite la selección (en vez de un ítem correspondiente a un menú desplegable). En la siguiente figura 6.3, se muestra el contenido del panel 1 en el formulario de Inicio.



**Figura 6.3.** Selección dentro de la Zona 1 del formulario de Inicio. jPanel1.

El código utilizado para la selección del paciente, implementando mediante un evento es que se muestra en la siguiente tabla de código 6.1. En este caso las funciones son las de trasladar los datos de la tabla anterior a los campos de texto que están justo debajo de estos. Para ello simplemente se crea una variable por campo, a continuación mediante una estructura if-ese se comprueba si alguna fila ha sido seleccionada, y en caso afirmativo recupera el valor de la tabla según la posición que este ocupe mediante la siguiente línea:

`idSujeto = (String) tablaInicio.getValueAt(filasel,0);` Se asigna el valor contenido en la columna 0 de la fila seleccionada correspondiente a `tablaInicio`, este valor es casteado en una cadena antes de su almacenamiento. Seguidamente se muestra en el campo de texto reservado para ello ("`txtId`"), mediante la sentencia "`txtId.setText(idSujeto)`". Esto se repite para cada una de las filas de la tabla.

Quedan dos campos cuya información no procede de la tabla de búsqueda, el primero "`txtFecha`" se autocompleta utilizando para ello el método `Date()`, (predefinido en java). Este obtiene la fecha en la que se está realizando el test, para ello utiliza como fuente la fecha que tiene ajustada el sistema operativo. Las líneas que implementan este proceso son las siguientes

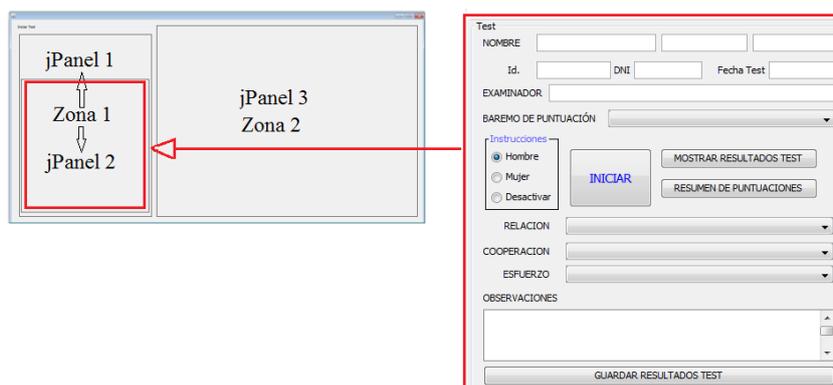
- 1- `Date fecha1 = new Date();` Se instancia la clase `Date`, en el objeto `fecha1`.
- 2- `fecha= fecha1.toLocaleString();` se almacena el contenido de `fecha1` en la variable `fecha`, mediante el método `toLocaleString()`.
- 3- `txtFecha.setText(fecha);` Finalmente se muestra el contenido de `fecha` en el campo de texto "`txtFecha`".

El otro campo es el denominado en la interface "Examinador" (`txtExaminador`), este es posible rellenarlo antes de almacenar los resultados.

```
1. private void btnSeleccionarActionPerformed(java.awt.event.ActionEvent
   evt) {
2.         int filasel;
3.         String idSujeto;
4.         String dniTest;
5.         String nombreTest;
6.         String fecha;
7.         String apel;
8.         String ape2;
9.         Date fecha1 = new Date();
10.        try{
11.            filasel = tablaInicio.getSelectedRow();
12.            if(filasel!=-1){
13.                JOptionPane.showMessageDialog(null,"No se ha +
14.                    seleccionado ninguna fila");
15.            }
16.            else {
17.                ;
18.                idSujeto = (String) tablaInicio.getValueAt(filasel,0);
19.                dniTest = (String) tablaInicio.getValueAt(filasel,1);
20.                nombreTest = (String) tablaInicio.getValueAt(filasel,2);
21.                apel = (String) tablaInicio.getValueAt(filasel,3);
22.                ape2 = (String) tablaInicio.getValueAt(filasel,4);
23.                fecha= fecha1.toLocaleString();
24.                txtId.setText(idSujeto);
25.                txtdni2.setText(dniTest);
26.                txtnombre2.setText(nombreTest);
27.                txtFecha.setText(fecha);
28.                txtApel.setText(apel);
29.                txtApe2.setText(ape2);
30.            }
31.        }
32.        catch(Exception e)
33.        {
34.        }
```

**Tabla 6.1:** Código: Botón seleccionar en el formulario Inico.

Dentro del jPanel1, se introduce otro jPanel2 para diferenciar los formularios de búsqueda y de selección dentro del formulario general “Inico”. Esta parte de la zona I (Iniciar), está dedicada a recoger los datos necesarios para que el programa lleve a cabo un correcto cálculo de resultados y las observaciones sobre la conducta que ha tenido el sujeto durante la ejecución del test.



**Figura 6.4.** Selección dentro de la Zona 1 del formulario de Inicio. jPanel1.

### 2- Ingreso del nombre del Examinador si procede

Una vez seleccionado el paciente, es posible ingresar el nombre del examinador que será almacenado junto a los demás datos y los resultados del test.

### 3- Selección del modo de instrucción.

Para la ejecución del test, es necesario trasladarle al sujeto las instrucciones. El programa da la opción de que estas sean narradas por la voz de un hombre, de una mujer o simplemente que no sean narradas. Las instrucciones narradas por el interlocutor virtual se pueden ver en el capítulo 2, y la clase que implementa el reproductor se puede revisar en el anexo de códigos.

Se ha dispuesto en la interface una zona reservada de selección, implementada mediante elementos del tipo JRadioButton. Como se observa en la figura 6.5, se han incluido tres correspondientes a las opciones. La programación de estos debe ser de tal forma que no sea posible la elección de dos opciones simultáneamente. Con este fin se han reorganizado los tres “JRadioButton” en un elemento denominado “ButtonGroup”, este es añadido de forma independiente al panel que contiene estos botones, es decir; debe ser anexo al Form principal. La Figura 6.5, muestra los elementos utilizados en la implementación de la selección de instrucciones.



**Figura 6.3.** Elementos del grupo de botones: Instrucciones.

#### 4- Selección del Baremo.

El único paso que queda antes de ejecutar el test, es seleccionar la baremación con la que se van a obtener los resultados estadísticos. Las puntuaciones estadísticas dependen de la edad y nivel cultural del sujeto examinado, ya que estas son obtenidas a partir de las puntuaciones directas y de las tablas de baremación, las cuales contiene muestras tipificadas de la prueba para individuos con características similares al ejecutor del test.

- *El funcionamiento de la baremación de resultados se verá en el capítulo: Resultados estadísticos.*

En este punto, si procede mostrar el código implementado en la clase “Inicio” necesario para llevar a cabo la selección del baremo y del narrador de instrucciones, pudiéndose ver este en la siguiente tabla de código. Como cabe de esperar este será captado tras pulsar el botón de “Iniciar” que lleva a la ejecución del test.

```

1. private void btnIniciarActionPerformed(java.awt.event.ActionEvent evt)
2. {
3.     Baremo=cboBaremo.getSelectedItem().toString();
4.     if(rbHombre.isSelected()){
5.         Instrucciones="Hombre";
6.     }
7.     else if(rbMujer.isSelected()){
8.         Instrucciones="Mujer";
9.     }
10.    else {
11.        Instrucciones="SinSonido";
12.    }
13.    new Test.EjecucionTest().setVisible(true);
14. }

```

**Tabla 6.2:** Código: Botón Inicar en el formulario Inico.

Como se puede observar en el código anterior, para la captación del narrador se utilizan estructuras if-else y el método isSelected(), el cual devuelve un 1 al if en el caso de ser la opción elegida, la variable “Instrucciones” de tipo string será cargada con la opción seleccionada.

El contenido de esta variable será utilizado como argumento en la llamada al método que implementa el reproductor de sonido, para que este reproduzca la opción seleccionada.

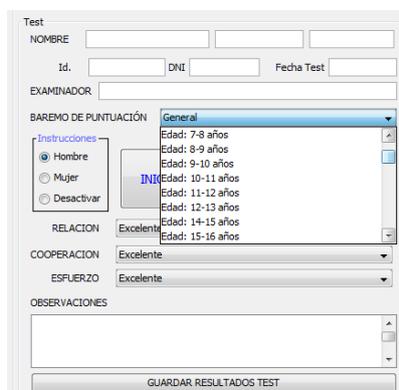
La variable “Baremo” será cargada con una cadena que indique el baremo seleccionado. Lo primero que hay que hacer es cargar el combo-box que permite la selección de la baremación, esto se hace con todas las posibles opciones de baremos en el constructor de clase. A continuación se muestran unas líneas (Tabla 6.3) con algunas de las opciones de baremación incluidas en el combo-box. No se incluye todo el código por ser demasiado extenso, además está incluido en el anexo de códigos de programa.

```

1.  cboBaremo.addItem("Edad: 18-19 años");
2.  cboBaremo.addItem("Edad: 19-20 años");
3.  cboBaremo.addItem("Edad: 20-30 años y 8 años escolarización");
4.  cboBaremo.addItem("Edad: 20-30 años y 9-11 años escolarización");
5.  cboBaremo.addItem("Edad: 20-30 años y 12 años escolarización");
    
```

**Tabla 6.3:** Código de carga del combo-box correspondiente a la selección del baremo de puntuación.

El resultado de esta carga sobre la interface de inicio se muestra en la figura 6.4 donde está desplegado el combo-box.



**Figura 6.4.** Elementos del grupo de botones: Instrucciones.

Para capturar la selección del combo, se utiliza el método: `getSelectedItem()` encargado de seleccionar el contenido deseado del elemento combo-box. Es necesario la conversión del `Item` seleccionado en un `String`, ya que debe ser del mismo tipo que la variable `Baremo` (`string`). Esto nos lo permite el método `toString()`.

Esta cadena contenida en la variable `Baremo`, será almacenada junto a los datos personales y la matriz que contiene las respuestas. Para que cuando se proceda a la obtención de los resultados estadísticos, la aplicación utilice en esta labor la tabla de baremos correspondiente.

El proceso para mostrar los resultados estadísticos que corresponden al baremo seleccionado se podrá ver en el capítulo : Resultados estadísticos.

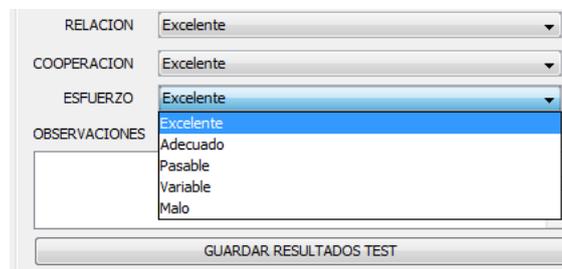
Tras completar todos los campos anteriores se orienta la pantalla hacia el sujeto que va realizar la prueba, cuando esté preparado se le da al Botón “Iniciar”. Desde este momento el examinador puede observar, pero no es necesaria su intervención, ya que el test está diseñado para auto-finalizarse cuando se completen los intentos correspondientes.

➤ ETAPA POST-TEST:

Una vez se ha ejecutado el test, la aplicación vuelve a mostrar la ventana de Inicio, dando la posibilidad de mostrar los resultados y guardar el test.

Antes de guardar el test, se puede incorporar al formulario y por tanto a la base de datos asociada a este paciente, observaciones del examinador sobre: La Relación, Cooperación y Esfuerzo, estas serán seleccionadas en los desplegados correspondientes, eligiendo entre las opciones descritas a continuación.

Para ello como se muestra en la siguiente figura 6.5, se dispone de tres elementos combo-box cargados con las posibles opciones que definan el comportamiento del sujeto durante la prueba. Este proceso es similar al que lleva a cabo durante la evaluación del test de forma tradicional, anotando esta información en la hoja de anotaciones descrita en el capítulo 2.



The image shows a screenshot of a software interface with four dropdown menus and a list box. The first three are labeled 'RELACION', 'COOPERACION', and 'ESFUERZO', each with 'Excelente' selected. The fourth is labeled 'OBSERVACIONES' and has a list of options: 'Adecuado', 'Pasable', 'Variable', and 'Malo'. Below the list box is a button labeled 'GUARDAR RESULTADOS TEST'.

**Figura 6.5.** Elementos combo-box cargados con las opciones que describen el comportamiento del sujeto durante la ejecución del test.

Las opciones que describirán el comportamiento son las siguientes:

- Relación: Excelente, Buena, Pasable y mala
- Cooperación: Excelente, Adecuada, Variable, Forzada e Inexistente.
- Esfuerzo: Excelente, Adecuado, Pasable y Malo.

La carga de estos combos se realiza de la misma forma que se ha hecho con el que contenía los baremos de puntuaciones. Finalmente se incluye un elemento jtext, para poder incluir observaciones que no quedan descritas mediante las opciones anteriores u otra información relevante. Una vez completada esta información se procede al guardado de toda la información en la tabla de la base de datos, “datostest”

Para el guardado de esta información, se ha optado por la programación directa de dicho proceso sobre el evento del botón. Cuando se pulsa el botón guardar Resultados del test la aplicación ejecuta la siguiente tabla de código

```
1. Private void btnGuardarResultadosTestActionPerformed(java.awt.event.ActionEvent
   evt) {
2.     String matrizComoString = toString( EjecucionTest.salida );
3.
4.     ConexionMySQL mysql = new ConexionMySQL();
5.
6.     Connection cn = mysql.Conectar();
7.
8.     String idSuj,dni2,nom2,ap1,ap2,Exa,FechaT,Rel,Cop,Esf,Observ;
9.
10.    String sSQL = " ";
11.    String mensaje = " ";
12.
13.    idSuj = txtId.getText();
14.    dni2 = txtdni2.getText();
15.    nom2 = txtnombre2.getText();
16.    ap1 = txtApel.getText();
17.    ap2 = txtApe2.getText();
18.    Exa = txtExaminador.getText();
19.    FechaT=txtFecha.getText();
20.    Baremo = cboBaremo.getSelectedItem().toString();
21.    Rel = cbo1.getSelectedItem().toString();
22.    Cop = cbo2.getSelectedItem().toString();
23.    Esf = cbo3.getSelectedItem().toString();
24.    Observ= txtObserv.getText();
25.
26.    sSQL = "INSERT INTO datostest(dni, nombre2, apellido1,+
27.        apellido2, Examinador, Fecha,""Relacion,Cooperacion,+
28.        Esfuerzo,Observaciones,salida,Baremo,idSujeto )"
29.        + "VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
30.
31.    mensaje = "Los resultados del test han sido guardados";
32.
33.    try {
34.        PreparedStatement pst = cn.prepareStatement(sSQL);
35.
36.        pst.setString(1,dni2);
37.        pst.setString(2,nom2);
38.        pst.setString(3,ap1);
39.        pst.setString(4,ap2);
40.        pst.setString(5,Exa);
41.        pst.setString(6,FechaT);
42.        pst.setString(7,Rel);
43.        pst.setString(8,Cop);
44.        pst.setString(9,Esf);
45.        pst.setString(10,Observ);
46.        pst.setString(11,matrizComoString);
47.        pst.setString(12,Baremo);
48.        pst.setString(13,idSuj);
49.
50.        int n = pst.executeUpdate();
51.
52.        if(n > 0)
53.        {
54.            JOptionPane.showMessageDialog(null,mensaje);
55.            CargarTablaPacientes2("");
56.        }
```

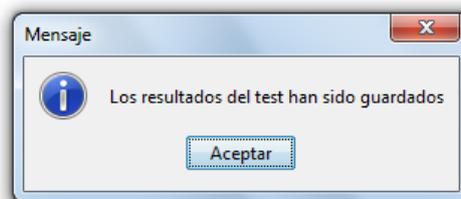
```
57.     }
58.     catch (SQLException ex)
59.     {
60.         JOptionPane.showMessageDialog(null, ex);
61.     }
```

**Tabla 6.4:** Programación del botón: Guardar Resultados del Test.

No procede la descripción de dicho código, ya que este es el mismo que el utilizado para registrar los pacientes desarrollado en el capítulo anterior.

La novedad respecto al registro de los datos clínicos del paciente almacenados en la base de datos “datospaciente”, es que ahora se almacena un array que almacena las respuestas dadas al test y el baremo correspondiente.

Si todo el proceso se ha realizado de forma correcta tras pulsar el botón de guardado del test se mostrará el mensaje mostrado en la figura 6.6.



**Figura 6.6.** Mensaje, resultados guardados de forma correcta.

### **6.1.2 ZONA 2: VENTANA DE RESPUESTAS**

Esta interface también nos permite mostrar las respuestas y resultados obtenidos tras la ejecución del test y antes de ser guardados. Esta corresponde a la zona 2 definida al principio de este capítulo.

Para mostrar las respuestas en las columnas, se debe pulsar el botón “Mostrar Resultados Test” una vez este finalice. También es posible ver la hoja de resultados del mismo modo pulsando el botón “Resumen de puntuaciones”.

Como se puede ver en la figura 6.7 al pulsar el botón “Mostrar Resultados Test”, las cuatro tablas son rellenadas con los intentos realizados durante la ejecución.



## CAPÍTULO 7

### EJECUCIÓN DEL TEST

---

---

En este capítulo se describen los pasos dados en el diseño y código utilizado en la creación de la aplicación del propio test, se comenzará creando la interface. La cual deberá ser lo más adecuada posible al objetivo de la aplicación, que no produzca modificaciones en la conducta o forma de proceder en el paciente, con respecto a si este se realizase de forma tradicional. Es decir no será demasiado decorada, simplemente se ajustará a cómo sería el tratamiento en una mesa y con los elementos correspondientes (mazos, estímulos...) en su forma tradicional.

#### 7.1 DISEÑO DE LA INTERFACE

De forma simplificada este capítulo, aborda la parte de software destinado a la ejecución del test así como su funcionamiento interno para la captación de las respuestas y preparación de datos a incluir en la base de datos. El primer paso consiste en diseñar la interface basándose en los requisitos de la prueba, y en conseguir una apariencia adecuada y lo más aproximada posible a una interacción clínico-paciente.

##### 7.1.1 Descripción y requisitos

Antes de proceder a crear la interface en el IDE, se presenta a continuación un esquema de lo que debe aparecer en pantalla durante la ejecución del test, es decir lo que visualizará el sujeto evaluado. Como se ha comentado anteriormente debe ser sencillo, intuitivo y que no presente dudas al sujeto que puedan modificar su conducta.

En ella se deben mostrar y localizar los siguientes elementos:

- 4 Cartas estímulo
- 4 Posiciones para la última tarjeta emparejada a cada estímulo.
- Mazo de 128 cartas
- Animación y señal sonora: “feedback”

Además de los estímulos y la carta actual será necesario otro elemento para comunicar al ejecutor del test, si es o no correcta su elección ante cada estímulo, esto se

hará mediante una animación. La animación mostrará V en verde si es correcta y X en rojo si es incorrecta.

En conjunto con la animación, aparecerá una señal sonora que también servirá para indicarle al paciente si correcta o no la respuesta, por ello en realidad serán dos sonidos diferentes, uno para cuando sea correcta y otro para incorrecta.

Por tanto el conjunto del sonido y la animación son los que transmiten el “feedback” al sujeto, estos deberán ser sencillos y adecuados a su función.

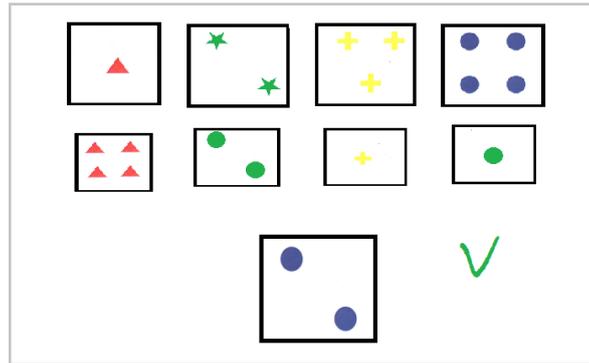
Al igual que estos sonidos, al iniciar la interface del test debe aparecer una pequeña narración dirigida hacia el sujeto con las indicaciones que daría el profesional que aplica el test si su ejecución fuese de forma tradicional.

Estas son las únicas instrucciones que va a recibir el sujeto, a partir de este momento no debe haber ninguna interacción del sujeto con otra fuente que no sea el test. Las instrucciones dadas son las que aparecen en la figura 7.1. Se debe notar que hay una pequeña variación respecto a las instrucciones dadas de forma clásica por el profesional, ya que en este caso es el propio sonido del Pc el que narra dichas instrucciones.

*“ Este test es poco usual porque yo no voy a dar mucha información sobre lo que hay que hacer: La tarea consiste en emparejar la tarjeta de abajo con una de las cuatro que hay colocada arriba y que sirven de clave, cada vez que se seleccione una de las tarjetas de arriba que se empareja directamente con la de abajo y a continuación aparece una nueva tarjeta en la parte inferior para emparejarla de nuevo con una de la parte superior. No puedo decir cómo se deben emparejar las tarjetas, pero cada vez que Vd, coloque una; se indicará en pantalla y mediante un sonido si lo ha hecho correctamente o si se ha equivocado. Si es correcta aparecerá una V verde acompañada del siguiente sonido(sonido correcta), y si es incorrecta aparecerá una X en rojo y el sonido (sonido incorrecta). Para esta prueba no hay tiempo límite, ¿está Vd. preparado? Puede comenzar “*

**Figura 7. 1:** Instrucciones dadas al inicio de la prueba.

Con todo lo dicho anteriormente la forma aproximada en la que debe quedar la interface del test debe ser la mostrada en la Figura 2. En esta se pueden observar todos los elementos antes mencionados. La primera fila de cartas corresponde a las tarjetas-estímulo, la segunda corresponde con la última tarjeta emparejada a cada estímulo que tiene en su parte superior. La carta inferior es la siguiente carta a emparejar con los estímulos, es decir la carta superior del mazo y finalmente a la derecha de esta aparece la animación que indica si es correcto o no el emparejamiento.



*Figura 7.2. Aproximación a la interface del Test.*

### **7.1.2 Diseño de la interface en el IDE.**

En este punto del capítulo, se describen los pasos dados para el diseño de la interface, esto se hace en el IDE de NetBeans, ya que este nos permite un diseño rápido, atractivo y adecuado sin necesidad de tener altos conocimientos de java.

La interface no debe ser nada más y nada menos como la aproximación o boceto que aparece en la Figura 7.2, lo primero es diseñarla mediante el editor y a continuación se interrelacionarán los elementos que vamos a incluir en la misma.

La componente básica que requerimos para implementar una interfaz visual con la librería Swing es la clase JFrame. Esta clase encapsula una Ventana clásica de cualquier sistema operativo con entorno gráfico (Windows, OS X, Linux etc.)

Posteriormente se programarán los elementos para captar las respuestas o estímulos a los que responde el sujeto. Utilizando esta clase a diferencia de las “JInternalFrames”, cuando se llame a la ejecución del test, este no aparecerá sobre la interface de la ventana del menú principal, sino que se superpondrá a esta.

Esta clase se encuentra en el paquete javax.swing, desde el IDE al crear una clase tipo JFrame Form lo que realmente se está haciendo es crear una clase que herede de la clase JFrame y extienda sus responsabilidades, a esta se podrán agregar botones, etiquetas, editores de líneas etc, esta será llamada “EjecucionTest”.

La importación del paquete donde se encuentra la clase JFrame la incorpora directamente el IDE (`import javax.swing.*;`), y en la Main crea directamente un objeto de dicha clase, que inicialmente se dejará para probar el funcionamiento de forma independiente a las demás clases y al finalizar el desarrollo de la aplicación se eliminará, ya que esta será llamada desde la clase que implementa el menú principal.

Pasando en el IDE de la pizarra source a desing tendremos inicialmente lo que aparece en la figura 7.3, que es simplemente la base o fondo donde se van a colocar el resto de elementos.



**Figura7.3.** Inicio interface

Una vez hecho esto, debemos insertar los elementos necesarios para simular el test, estos serán los siguientes:

**1-. Cuatro botones** con la imagen de las cuatro tarjetas estímulo. Cada uno de estos botones estará programado para captar la respuesta del paciente en el momento en el que este seleccione una tarjeta estímulo, o mejor dicho empareje la tarjeta respuesta del mazo al estímulo.

Para incluir los botones, se deben arrastrar desde la paleta de controles cuatro elementos tipo `JButton` o “botones” de la librería Swing, estos permiten una captación de la respuesta del sujeto mediante eventos y serán programados con el código adecuado descrito más avanzado el capítulo. En la figura 7.4. Aparece la inclusión de dichos botones sobre el frame.



**Figura 7.4.** Botones en la interface.

Para incluir las imágenes en cada uno de los botones, debemos acceder a la ventana de herramientas, posteriormente en propiedades del elemento localizar la opción “icon” y en su ventana desplegable buscar donde tenemos la imagen albergada. Anteriormente se habrá creado un package llamado estímulos donde guardaremos las imágenes de dichos estímulos.

El formato de las imágenes será del tipo “jpg” y las dimensiones 200x175, estas han sido creadas en un editor de imágenes y para referirnos a ellas se les ha nombrado según están colocadas de izquierda a derecha en la interface como: estímulo1.jpg correspondiente a la imagen de un triángulo rojo, estímulo2.jpg correspondiente a dos estrellas verdes, estímulo3.jpg tres cruces amarillas y estímulo4.jpg que contiene cuatro círculos azules.

**2-. Cuatro contenedores** o paneles (tipo JPanel) en los que aparecerá la última tarjeta emparejada a la que hay encima de la misma, es decir al estímulo superior.

**3-. Un contenedor que simule el mazo de cartas**, que cada vez que se empareje una carta estímulo con la carta respuesta del mazo, en este aparezca la siguiente tarjeta respuesta. Estos cinco paneles estarán implementados mediante objetos o instancias de la clase JPanelConFondo, esta es una clase implementada independientemente y que será instanciada cinco veces en la clase de la interface “EjecucionTest”, habrán cinco objetos de la misma, cuyas variables serán diferentes.

La clase JPanelConFondo hereda de la clase del paquete Swing JPanel que viene por defecto en el IDE para la programación gráfica en Java. A esta se le incluyen las líneas de código descritas a continuación, en la (tabla 7.1)

Esta clase tiene como finalidad incorporar las imágenes a los contenedores o paneles de la interface del test y actualizarlos, para que un **JPanel** tenga una imagen de fondo lo que se hace es sobrescribir el método **paint**, dibujando la imagen del tamaño del panel.

```
1. package Test;
2.
3. import java.awt.Graphics;
4. import java.awt.Image;
5. import javax.swing.ImageIcon;
6. import javax.swing.JPanel;
7.
8. public class JPanelConFondo extends JPanel {
9.
10.     private Image imagen;
11.
12.     public JPanelConFondo() {
13.     }
14.     public void setImagen(String nombreImagen) {
15.         if (nombreImagen != null) {
16.             Imagen= new ImageIcon(getClass().getResource(nombreImagen)).getImage();
17.         } else {
18.             imagen = null;
19.         }
20.         repaint();
21.     }
22.     public void paint(Graphics g) {
23.         if (imagen != null) {
24.             g.drawImage(imagen, 0, 0, getWidth(), getHeight(), this);
25.             setOpaque(false);
26.         } else {
27.             setOpaque(true);
28.         }
29.         super.paint(g);
30.     }
31. }
```

**Tabla 7.1** Código Clase JPanelConFondo

En este caso de la clase “paint” incluida en la librería awt.graphics se han utilizado los siguientes métodos:

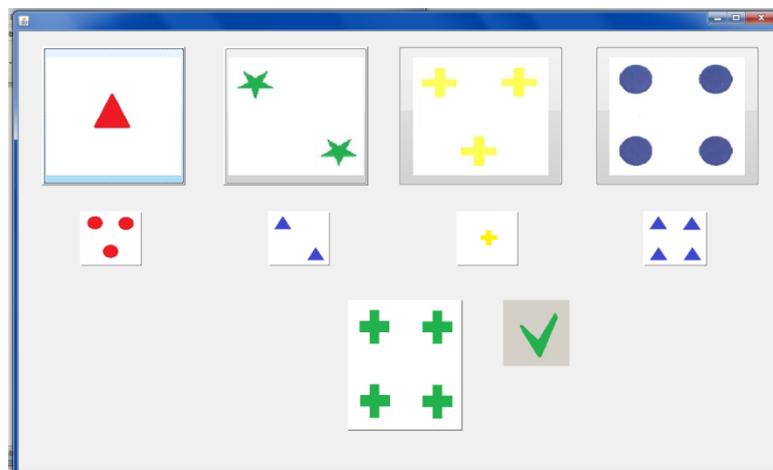
- Framework g.drawImage: dibujamos la imagen (guardada en el atributo “imagen”) en toda la extensión del panel gracias a getWidth() y getHeight().
- g.drawImage: dibujamos la imagen (guardada en el atributo “imagen”) en toda la extensión del panel gracias a getWidth() y getHeight().
- setOpaque(false): le indicamos al panel que no dibuje su fondo por defecto (sino este taparía la imagen)
- super.paint(g): le indicamos al panel que continúe dibujando el resto de los componentes

Para que el panel dibuje la imagen solo si está asignada a la variable “imagen”, en caso contrario (imagen == null) se dibujara el fondo por defecto.

- repaint(): Lo llamamos explícitamente para que cuando cambiemos la imagen esta se muestre inmediatamente. Si el parámetro nuevaImagen es null, veremos el fondo por defecto.

Este código se podría haber implementado en la propia clase EjecucionTest, pero de esta forma adquiere una mayor modularidad y sencillez de estructura.

Finalmente la interface quedará según la figura 7.5.



**Figura7.5** .Interface Completa del test.

### 7.1.3 Programación de los botones de la interface.

Una vez diseñada la interface gráfica, se procede a la programación de cada uno de los botones para que realicen sus funciones, así como la interconexión con los paneles que deberán ir mostrando los estímulos o imágenes durante la prueba. Además a continuación podremos ver que a los botones se les ha añadido alguna función extra aparte de la captación simple de la respuesta.

En la Tabla 7.2 se muestra el inicio de la clase “EjecuciónTest” que como comentamos anteriormente mediante “extends” recibe las características de la clase JFrame incorporada en la biblioteca gráfica Swing. Además se pueden observar las variables creadas para el desarrollo de la clase y del programa en sí. Algunas variables van acompañadas del modificador public, para que todos puedan acceder a esa variable, y del modificador static que indica que será la misma para todas las instancias de una clase.

Serán necesarias variables para indicar la evolución o avance del programa como i ,i2, cat, cat2 que contendrán los valores del número de estímulo actual, y de las categorías presentes en cada momento. Además se incluyen las variables que almacenaran el número de intentos así como el número de aciertos.

Cabe destacar la creación de dos variables, una de ellas de tipo string inicializada con la dirección de la primera carta estímulo y la segunda del tipo cadena de enteros, que reserva tres espacios en memoria destinada a albergar la selección del estímulo, para poder compararla posteriormente con la respuesta adecuada.

```
1. package Test;
2.
3. public class EjecucionTest extends javax.swing.JFrame {
4.
5.     public int i= 1;
6.     static int i2=0;
7.     public int cat,cat2=0;
8.     static int catAcertadas=0;
9.     public String c="/figuras/figural.jpg";
10.    public String cultima;
11.    public int contacierto =0;
12.    static int numerointentos =0;
13.
14.    public int[] select=new int[3];
```

**Tabla 7.2** Declaración de variables en la clase EjecucionTest.

Para finalizar con las variables se crean las dos más importantes

- Int [i][j]valorfigura
- Int [i][j]salida

La primera se inicializa con el código correspondiente a cada figura. La segunda será la matriz que almacene cada una de las respuestas, para ello tendrá una dimensión 128 filas y 9 columnas, cuyo contenido se describirá posteriormente.

Antes de comenzar con la programación de cada uno de los botones y las interrelaciones de todos los elementos, se describe el sistema adoptado para la captación de las respuestas y del funcionamiento de la interface.

Para la captación y almacenamiento de las respuestas se usa un sistema simple de codificación, que posteriormente servirá, para el tratamiento y procesado de resultados.

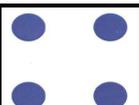
Esta codificación consiste en asignar un código a cada figura según categorías de clasificación, de esta forma cuando se empareje la tarjeta respuesta a la estímulo lo que se hace es captar el código de la figura del estímulo (seleccionado mediante la pulsación del botón) almacenarla en una variable y compararla con el código de la tarjeta respuesta, el resultado de esta comparación será la salida. Como en la hoja de anotación tradicional lo que se apunta a cada respuesta son las coincidencias en cada una de las categorías (color, forma, número y otros) se asignará un valor “1” en caso de coincidencia en categoría y “0” en caso de no coincidencia y esto será así para cada una de ellas.

La codificación utilizada para las figuras, se muestra a continuación.

CÓDIGO	COLOR	FORMA	NÚMERO
1	ROJO	TRIANGULO	1
2	VERDE	ESTRELLA	2
3	AMARILLO	CRUZ	3
4	AZUL	CÍRCULO	4

**Figura7. 6** Sistema de codificación seleccionado

Esto fija los códigos de las figuras estímulo (figura 7.7), del mismo modo serán codificadas las tarjetas respuesta.

	COLOR	FORMA	NUMERO
	1	1	1
	2	2	2
	3	3	3
	4	4	4

**Figura 7.7:** Códigos de las tarjetas estímulo.

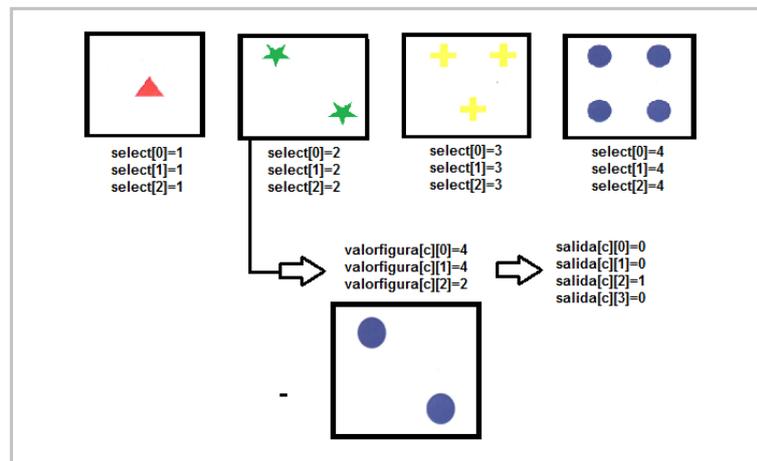
El funcionamiento será el siguiente: habrá una matriz que contendrá el valor del código de cada una de las tarjetas respuesta; como estas son 128 y están definidas por su color, forma, número y la categoría otros, la dimensión de la matriz necesaria para la captación de respuestas será 128x4. Esta seguirá el orden establecido por el test, de

forma que la primera carta que deberá emparejar el sujeto tendrá la primera posición en la matriz, es decir será la primera fila y deberemos dirigirnos a ella con el subíndice 0.

Al iniciar la aplicación se inicia un contador, que corresponde a cada posición de las tarjetas respuesta en el mazo de cartas, de forma que cada vez que se haga un emparejamiento se aumentará el contador.

Este nos servirá para mostrar la secuencia de las tarjetas-respuesta en el orden adecuado, así como para acceder a la posición en la que está guardado el código de dicha tarjeta.

Para explicar mejor el funcionamiento, se muestra en la figura 7.8 un ejemplo gráfico de lo expuesto anteriormente.



**Figura7. 8:** Ejemplo de funcionamiento en la captación de respuestas.

En este caso la tarjeta respuesta que corresponde a la de los dos círculos azules, se asocia a la tarjeta estímulo con dos estrellas verdes, por tanto no habrá coincidencia en las categorías color y forma pero si la habrá para número.

Se debe añadir que la cuarta posición del vector salida corresponde a la categoría otras y a esta se le asignará un valor "1" en el caso de no haber acertado en ninguna de las categorías anteriores y un valor cero "0" si por el contrario se ha acertado al menos una de las categorías color, forma o número.

Este proceso se debe realizar cada vez que se pulse uno de los botones estímulo en el emparejamiento de tarjetas. Por tanto será la programación del propio botón la que implemente este proceso.

Para entender esto se debe hacer una breve introducción sobre el manejo de eventos en java, un evento es un objeto que representa un cambio en un componente, este es generalmente producido por el usuario al realizar alguna operación, por ejemplo al hacer click sobre un botón. El método que se sigue para atender un evento es el siguiente:

1. Se relaciona a un oyente (listener) con un componente para que el componente le notifique al listener cuando ocurra un evento específico.

2. Cuando se produce el evento, el listener atiende el evento.

En este método la implementación de la interface de usuario donde se produce el evento y la implementación del oyente están en clases separadas. El manejo de eventos se puede implementar a través de las clases abstractas e interfaces que java proporciona en el paquete “java.awt.event”

En nuestro caso estamos ante eventos generados por botones, para atender los eventos generados por componentes Button, se usa una clase anidada que implementa a la interface ActionListener y se define el método acciónPerformed. Es posible que dos o más botones sean atendidos por el mismo método si a cada uno se le añade el ActionListener.

NetBeans nos permite realizar esta operación de forma sencilla, para ello en la pizarra de diseño colocando el puntero del ratón encima del botón y pulsando click derecho, se está definiendo el método actionPerformed, que espera un evento, en este caso un click sobre él. En la figura 7.9 se presenta una captación de pantalla de la asignación del evento

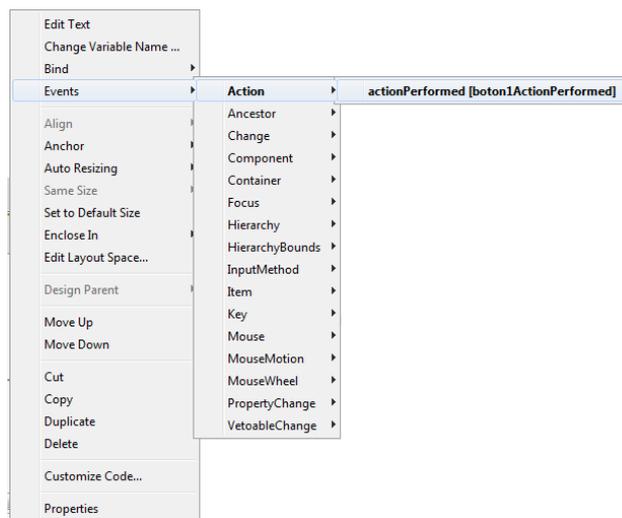


Figura7. 9. Selección de menú: Asignación de evento.

Una vez asignado el evento, cada vez que se pulse el botón el código ejecutado será el que contenga el objeto del evento, la programación de cada uno de los botones es similar, la diferencias están en que cada uno tiene asignado un código de selección diferente según la figura que contenga, por lo tanto a continuación se muestra y explica el código correspondiente a uno de los botones y el resto será similar con el cambio comentado.

La ejecución del código será secuencial y basada en estructuras if-else, para comparar cada uno de los bits que intervienen en el proceso. En la siguiente tabla se muestra dicho código y a continuación se hace una descripción del mismo.

```
1. private void botonActionPerformed(java.awt.event.ActionEvent evt) {
2.
3.     i++;
4.
5.     c= "figuras/figura"+i+".jpg";
6.     cultima="figuras/figura"+i2+".jpg";
7.
8.     ((JPanelConFondo) jPanelConFondo5).setImagen(c);
9.     ((JPanelConFondo) jPanelConFondo1).setImagen(cultima);
10.
11.    select[0]=1;
12.    select[1]=1;
13.    select[2]=1;
14.
15.    if(select[0]==valorfigura[i-2][0]){
16.        salida[i-2][0]=1;}
17.    else{
18.        salida[i-2][0]=0;}
19.
20.    if(select[1]==valorfigura[i-2][1]){
21.        salida[i-2][1]=1;}
22.    else{
23.        salida[i-2][1]=0;}
24.
25.    if(select[2]==valorfigura[i-2][2]){
26.        salida[i-2][2]=1;}
27.    else{
28.        salida[i-2][2]=0;}
29.
30.    if((salida[i-2][0]==0)&&(salida[i-2][1]==0)&&(salida[i-2][2]==0)){
31.        salida[i-2][3]=1;}
32.    else{
33.        salida[i-2][3]=0;}
```

**Tabla 7.3** Programación de botones que simulan las cartas estímulo.

El proceso sigue un orden secuencial, es decir al pulsar el botón:

- 1- Se incrementa el contador que actualiza la posición u orden de la tarjeta-respuesta. (línea 3)
- 2- El valor de dicho contador se utiliza para mostrar la siguiente tarjeta-respuesta y colocar la tarjeta estímulo seleccionada en el panel destinado a contener la última tarjeta emparejada.
- 3- Mediante los objetos de la clase JPanelConFondo se muestran:
  - a- En el mazo(jPanelConFondo1) la siguiente carta a emparejar. Pasándole a este el contenido de la variable string “c” que corresponde a la dirección en la que esta almacenada dicha figura. (línea 8)
  - b- En el panel que hay justo debajo del botón que ha captado el evento (ha sido seleccionado) se muestra la imagen de la última carta emparejada. Para esto se le pasa al objeto jPanelConFondo5, como argumento la dirección de la última carta emparejada (variable “cultima”) (línea 9)

- 4- Se almacena el código de la tarjeta estímulo seleccionada en la variable “select”. (líneas 11,12 y 13)
- 5- Se compara si hay coincidencia en categoría color mediante un if-else., después se hace lo mismo para forma y número. (filas 15 a 28)
- 6- Se comprueba si no hay ninguna coincidencia en color, forma y número, al no poder saber en función a que criterio a emparejado el sujeto las tarjetas, la categoría correspondiente será “otras”. A la variable salida correspondiente a “otras” se le asignará un “1”. (filas 30 a 33)

De **las dimensiones valoradas, correcto-incorrecto, ambiguo-no ambiguo y perseverativo-no perseverativo** a continuación se comprueban las dos primeras en el propio código del botón. En cuanto a la perseveración se verá más adelante y estará implementada individualmente en una sola clase debido a su mayor complejidad.

7- Comprobación ambigüedad

Recordamos que cuando una tarjeta-respuesta se empareja con una tarjeta-estímulo atendiendo a una, y sólo una de sus características (por ejemplo, Color resulta obvia, **no ambigua**, la categoría utilizada por el sujeto para realizar el emparejamiento.

Esto sucede si, por ejemplo, una tarjeta-respuesta con tres cruces azules se ha emparejado con la tarjeta-estímulo que tiene una cruz azul.

Sin embargo las tarjetas pueden emparejarse en más de una dimensión. Un ejemplo de esto ocurre cuando una tarjeta-respuesta con tres cruces azules se empareja con el estímulo que sólo tiene una cruz azul. En esta situación, la respuesta es ambigua pues no se puede estar seguro de que el sujeto haya clasificado esa tarjeta considerando la categoría Color o Forma.

Por tanto, cuando la tarjeta-respuesta se empareja con la tarjeta estímulo en una, y sólo una característica se considera una respuesta no ambigua y cuando la tarjeta respuesta se empareja con la tarjeta-estímulo en dos o más características se considerará ambigua. Por definición la categoría otros contienen siempre respuestas ambiguas. La dimensión **ambigua-no ambigua** es independiente de que la respuesta sea correcta o errónea.

Para comprobar la ambigüedad en código (Tabla 7.4), se utiliza un if-else en el que si se tiene una salida “1” en la posición de otros o bien hay dos o más categorías emparejadas será ambigua, en caso contrario, si solo hay una categoría emparejada será no ambigua. Esta dimensión se almacenara en la posición 6 de la variable salida, correspondiente al número de respuesta en cuestión.(Tabla 7.4)

```

34. if((salida[i-2][3]==1) || ((salida[i-2][0]+salida[i-2][1]+salida[i-2][2]) >1)) {
35.     salida[i-2][6]=1;}
36. else{

```

```
37. salida[i-2][6]=0;}
```

**Tabla 7.4.** Código correspondiente a la comprobación de ambigüedad en la respuesta.

### 8- Comprobación dimensión correcta-incorreta.

En este caso se debe proporcionar al sujeto “feedback” en cada respuesta, tanto si es correcta como si es incorrecta, esto depende de que empareje las tarjetas considerando la categoría establecida. Cada vez que da una respuesta si esta corresponde a la categoría de clasificación establecida se puntúan como correctas y las que no corresponden como erróneas.

El “feedback” en la aplicación es implementado mediante una animación y un sonido, distintos en los casos de respuesta correcta o incorrecta.

Se deberá numerar la posición de la respuesta correcta dada a ítems consecutivos durante la aplicación. Si el sujeto da una o varias respuestas erróneas, se comienza de nuevo la numeración, y se reiniciará cuando vuelva a dar respuestas correctas.

A continuación se muestra el código (tabla 7.5) que implementa esta categoría, en este se crean instancia de la clase correspondiente al reproductor de sonido y se inicia con el sonido correspondiente a la señal asociada, ya sea esta correcta o incorrecta. De la misma forma se ejecuta la animación en función de la respuesta correcta o incorrecta, se almacena en la posición 4 del vector de salida un “1” en caso de respuesta correcta y un “0” en caso de incorrecta. Y finalmente se inicia el contador acierto.

```
38. if(salida[i-2][cat]==1){
39. new AePlayWave("c:/sonidos/bien.wav").start();
40. a.startAnimation();
41. contacierto++;
42. salida[i-2][4]=contacierto;}
43. else{
44. contacierto=0;
45. salida[i-2][4]=0;
46. new AePlayWave("c:/sonidos/mal.wav").start();
47. a.startAnimation2(); }
```

**Tabla 7.5.** Variables de la clase EjecucionTest.

### 9- Comprobación de categoría completa.

Para cada respuesta se comprueba si es la décima y por tanto última acertada por categoría, en cuyo caso se inicia el contador de categorías acertadas y se reinicia el contador de aciertos. En la posición 5 del vector “salida”, correspondiente a la respuesta actual, se almacena el número de categoría que acaba de ser acertada.

Cada botón también está configurado para salir del test cuando este ha sido finalizado, ya sea por acertar 6 categorías o bien por llegar al final de intentos posibles. Para este procedimiento el código usado es el mostrado en la siguiente tabla (5.5).

```
1. if( contacierto==10){
2. cat++;
3. catAcertadas++;
4. contacierto=0;
5. salida[i-2][5]=cat;}
6.
7. if(i!=129){
8. switch(cat2){
9. case 0:
10. if(cat==3){
11. cat=0;
12. cat2=1;
13. }
14. break;
15. case 1:
16. if(cat==3){
17. numerointentos=i2;
18. cat2=0;
19. cat=0;
20. this.dispose();
21. }
22. break;
23. }}
24. else{
25. numerointentos=128;
26. this.dispose();
27. }
```

**Tabla 7.5.** Código que implementa la comprobación de categoría completa.

## 7.2 DESCRIPCIÓN DE LA MATRIZ SALIDA[][]

La matriz “**salida[128][9]**” posee 128 filas correspondientes a cada una de las respuestas dadas al test. El número de columnas es 9, cada uno de estos nueve espacios está reservado para las diferentes categorías de puntuación e información relevante para el tratamiento de los resultados.

Tras finalizar el test, los espacios de memoria que tendrán contenidos valores, serán las 7 primeras columnas de cada una de las filas a las que se le ha dado respuesta. Al mismo tiempo que se va ejecutando el test, se van rellenando dichos espacios de memoria.

Los dos últimos espacios de cada una de las filas, están reservados para la puntuación referida a la perseveración en las respuestas. Esta puntuación no es directa,

por tanto será completada tras finalizar el test y pulsar el botón para mostrarlo. En este momento se realizará una llamada a la clase “Perseveracion” que se encarga de dicha puntuación, terminando así de rellenar la matriz de salida que contiene las respuestas.

Dicha matriz salida contiene solamente enteros, estos serán transformados posteriormente para facilitar el visionado de resultados. Es decir, los enteros serán transformados en letras según la categoría de puntuación y así ser mostrados en la hoja de resultados.

A continuación se muestran dos figuras equivalentes con 16 filas, correspondientes a los 16 primeros intentos de una prueba hipotética, la de la izquierda corresponde a la matriz salida (figura 7.10) y la de la derecha es la misma pero transformada para mostrarla en la hoja de respuestas.(figura 7.11)

Esto es así, porque el programa trabaja internamente con enteros y a la hora de trasladar dichos resultados a la hoja de respuestas, estos son transformados.

C	F	N	O	n	c	a	Cp	P
	1	1					1	
1				1				
1				2				
1		1		3		1		
1	1			4		1		
1	1			5		1		
1		1		6		1		
1				7				
1				8				
1				9				
1				10	1			
1	1			1				
1							1	5
1							1	5
	1			1			1	
1	1			2	1	1		

Figura 7.10: Matriz de Resultados.

N	C	F	N	O	n	c	a	cp	p
1		X	X				Am		
2	X				1				
3	X				2				
4	X		X		3		Am		
5	X	X			4		Am		
6	X	X			5		Am		
7	X		X		6		Am		
8	X				7				
9	X				8				
10	X				9				
11	X				10	C			
12	X	X			1		Am		
13	X							C	p
14	X							C	p
15		X			1			C	
16	X	X	X		2		Am	C	

Figura 7.11: Matriz de Resultados transformada.

- **Columna 1.** N: Número de intento, en la figura aparece rango de intentos desde el 1 hasta el 16. Se debe tener en cuenta, que aunque aparezca esta columna en ambas figuras, la matriz salida de la izquierda no posee dicha columna. En este caso solo se ha puesto para la comparación.
- **Columnas 2, 3, 4, 5.** correspondientes a las categorías “C”: Color, “F”: Forma, “N”: Número y “O”: Otros.

Si la tarjeta-respuesta y la tarjeta-estímulo se emparejan solamente en una dimensión, la hoja de anotación se marcará con una X solamente en el hueco que corresponde a esa dimensión. Por ejemplo si la tarjeta-respuesta se empareja con la tarjeta-estímulo solamente en cuanto a Color en la hoja de anotación se marcará este ítem poniendo sobre el intento en cuestión una X en la categoría color. Si la tarjeta-respuesta se empareja con la tarjeta estímulo en dos dimensiones, se marcarán con una X los dos huecos correspondientes a estas dimensiones en el intento correspondiente. Finalmente, si la respuesta no se empareja con la tarjeta estímulo

en ninguna dimensión se marcará en la hoja de anotación la opción Otros (C,F,N,Ø). Las respuestas se anotarán de esta forma en cada ítem, con independencia de que sean correctas o incorrectas respecto a la categoría de clasificación que esté considerando.

➤ **Fila 6** ó “n”: esta columna pertenece a la dimensión correcto-incorrecto, en esta solamente se numeran las respuestas correctas, si el sujeto interrumpe la secuencia con un error, cuando vuelva a dar una respuesta correcta se volverá a comenzar la numeración con el número 1, cuando se alcance el criterio de diez respuestas consecutivas, la categoría que acaba de completarse será indicada en la siguiente columna, entonces se vuelve a iniciar la numeración.

Que la respuesta sea correcta o incorrecta depende de que el sujeto empareje las tarjetas considerando la categoría establecida, cada vez que da una respuesta. En la categoría de clasificación establecida se puntúan como correctas y las que no corresponden, como erróneas, en estas aparecerá el hueco correspondiente en blanco.

➤ **Fila 7**, “c” categoría acertada, cada vez que se completa una categoría, el hueco reservado a este, indicará la dimensión de dicha categoría. Es decir, este espacio está reservado para marcar el final de categoría acertada, si la categoría presente es forma y el sujeto acaba de dar diez respuesta en forma, en el ítem correspondiente aparecerá “F”.

➤ **Fila 8**, “Am”, ambigüedad, este espacio es reservado para indicar la ambigüedad o no de la respuesta correspondiente.

Cabe destacar que una respuesta es ambigua cuando una tarjeta-respuesta se empareja con una tarjeta-estímulo atendiendo a una, y sólo una de sus características (por ejemplo, Color), resulta obvia, no ambigua, la categoría utilizada por el sujeto para realizar el emparejamiento. Esto sucede si, por ejemplo, una tarjeta-respuesta con tres círculos rojo se ha emparejado con la tarjeta-estímulo que tiene un triángulo rojo. Sin embargo las tarjetas pueden emparejarse en más de una dimensión. Un ejemplo de esto ocurre cuando una tarjeta-respuesta con tres triángulos rojos se empareja con el estímulo que sólo tiene un triángulo rojo. En esta situación, la respuesta es ambigua, pues no puede estar seguro de que el sujeto haya clasificado esa tarjeta considerando la categoría Color o la categoría Forma.

Por tanto, cuando la tarjeta-respuesta se empareja con la tarjeta estímulo en una, y sólo una, característica se considera una respuesta no ambigua y cuando la tarjeta respuesta se empareja con la tarjeta-estímulo en dos o más características se considerará ambigua. Por definición la categoría otros contienen siempre respuestas ambiguas. La dimensión ambigua-no ambigua es independiente de que la respuesta sea correcta o errónea.

➤ **Fila 9**. “cp” categoría del principio de perseveración. Las posibles categorías de perseveración serán Color, Forma o Número.

➤ **Fila 10**. “P” Respuesta perseverativa. En el espacio reservado a esta dimensión aparecerá una “p” en caso de que la respuesta sea perseverativa y quedará en blanco en caso de que no lo sea.

Las dos últimas filas están reservadas para la dimensión **Perseverativa-no perseverativa**: Cuando el sujeto persiste en responder una característica del estímulo que no es correcta, la respuesta se considera una perseveración hacia ese criterio y se puntúa como “perseverativa”. El sujeto puede perseverar en Color, Forma o Número. Sin embargo, no es posible que esto suceda en la categoría Otras porque el examinador no puede estar seguro del criterio de clasificación seguido por el sujeto. El principio de perseveración es operativo una vez establecido. De hecho las respuestas que se emparejen con el principio de perseveración establecido se consideran perseverativas, prescindiendo que sean correctas o incorrectas. Las respuestas que no se emparejen siguiendo el principio de perseveración se consideran no perseverativas.

### 7.3 CONVERSIÓN Y ALMACENAMIENTO DE LA MATRIZ DE RESPUESTAS EN LA BASE DE DATOS.

Para almacenar la matriz de respuestas en la base de datos, es necesario convertir esta matriz en un string y salvarla como un varchar. Posteriormente para rescatarla se lee de la base de datos y se convierte en un arreglo de nuevo, esto se hace para cada una de las respuestas almacenadas de forma que se llega a la reconstrucción de la matriz de respuestas.

Para realizar estas operaciones se construye la clase M (Matriz), esta va a contener los métodos necesarios para lograr las diversas transformaciones que se van a dar tanto en el almacenamiento como en la recuperación de datos relativos a las respuestas dadas al test.

```
1. public static String toString( int [][] m ) {
2.     StringBuilder sb = new StringBuilder();
3.     // recorre la matriz y ponerles “,”
4.     for( int I = 0 ; I < m.length ; i++ ) {
5.         for( int j = 0 ; j < m[i].length ; j++ ) {
6.             sb.append( m[i][j] );
7.             sb.append( “,” );
8.         }
9.     }
10.    // ni siquiera hace falta quitarle la ultima “,”
11.    return sb.toString();
12. }
13. }
```

A continuación se van a describir los dos métodos encargados de realizar estas operaciones, El primero de ellos es toString(): Tabla 7.6, como su nombre indica es el encargado de recibir como argumento la matriz de respuestas de tipo entero y devuelve una cadena que contiene la secuencia de caracteres en el constructor.



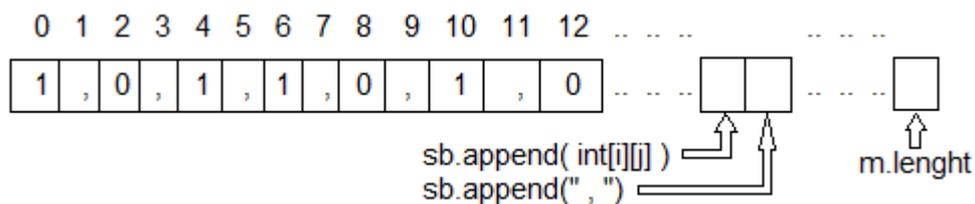
**Tabla 7.6.** Metodo “toString”, para convertir una matriz en cadena.

Los objetos StringBuilder son como los objetos String, excepto que estos pueden ser modificados. Internamente estos objetos se tratan como arreglos de longitud variable que contienen una secuencia de caracteres. En cualquier punto, la longitud y el contenido de la secuencia pueden cambiarse a través de invocaciones a estos.

Mediante la secuencia `StringBuilder sb = new StringBuilder();` se crea un constructor de cadena vacío, en el que inicialmente la capacidad es de 16. A continuación mediante un bucle for se recorre la cadena, en los argumentos de este se utiliza el método de longitud `length()` para obtener la longitud e `n` la secuencia de caracteres de la matriz en el constructor.

Conforme se recorre la matriz, se va invocando el método `append()` (que es de las operaciones principales sobre un `StringBuilder()`), este nos permite insertar los caracteres de esa cadena a la secuencia de caracteres en el constructor de cadenas. El método “`append()`” siempre agrega estos caracteres al final de la secuencia existente de caracteres.

Finalmente el objeto “sb” contendrá la cadena a almacenar en la base de datos, gráficamente el proceso se puede ver en la figura 7.12.



**Figura 7.12:** Funcionamiento del método `toString()`.

El segundo método implementado en esta clase, se denomina “`fromString`” realiza la operación contraria al anterior, es decir recibe una cadena “`matrizComoString`” y la convierte en una matriz de enteros denominada “`restored`” correspondiente a las respuestas del test.

Las operaciones que hace este método consisten en recorrer la cadena separada por comas que hasido almacenada en la base de datos y mediante el uso del método `Split`, devolver un array de cadenas in la coma. El for continuará hasta la última coma del array “`matrizComoString`” en cada iteración del for se irán almacenando en la matriz `restored` según la posición indicada, en este caso se hace necesaria la conversión de la

cadena resultante a enteros que es posible gracias al empleo del método estático de la

```
1. // Separar por "," y ponerlos en la matriz.
2. public static void fromString( String matrizComoString, int [][] restored ) {
3.     int i = 0;
4.     int j = 0;
5.     for( String numero : matrizComoString.split(",")) {
6.         out.printf( "(%d,%d) = %s%n",i,j,numero );
7.         restored[i][j++] = Integer.parseInt( numero );
8.         if( j > 2 ) {
9.             i++;
10.            j = 0;
11.        }
12.    }
13. }
```

case Integer, parseInt. En este caso para finalizar el bucle se utiliza el indicador incremental j, que determina la posición de la columna de la matriz. Como la salida del test está formada inicialmente por las posiciones Color, Forma, Número y Otros, es decir 4 posiciones y el indicador empieza por =, cuando j es mayor de 2 (0,1,2,3), pasa a la siguiente fila.

**Tabla 7.7.** Metodo "fromString", para convertir una cadena en matriz.

## CAPÍTULO 8

### HISTORIAL

---

---

En este punto del desarrollo del proyecto, ya es posible almacenar tanto datos de pacientes como de pruebas, ejecutar el test y mostrar los resultados sin llegar a cerrar la aplicación.

En los requisitos iniciales está el de poder mostrar historiales de pruebas. Es necesario un formulario-interface que permita recuperar la información almacenada de forma sencilla y rápida. También es requisito, la posibilidad de borrar e imprimir resultados de pruebas que hayan sido almacenados con anterioridad.

A continuación se describen cada una de las partes que ocupan esta función, como en capítulos anteriores, se comenzarán describiendo los controles seleccionados para el diseño de la interface y el montaje completo de esta.

Posteriormente se describirán los métodos más importantes usados en la programación de la aplicación. El código completo no se muestra en este capítulo, pero puede ser revisado en el anexo correspondiente a Códigos de programa.

#### 8.1 DISEÑO DE LA INTERFACE DE HISTORIALES

##### 8.1.1 Interface: Búsqueda de test (Clase Historial())

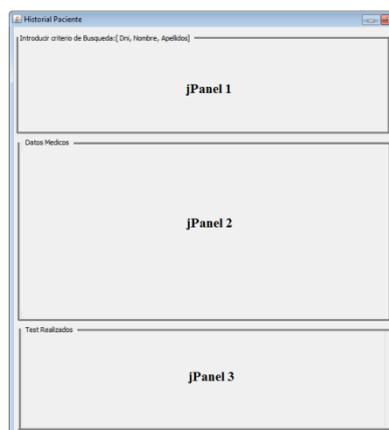
Esta implementa el formulario de localización de sujetos y tests.

Al igual que para todo el diseño, se trabaja en el IDE, se crea la clase Historial() en el Package Test; Esta heredará del tipo JFrame que implemente nuestra

interface, por tanto podrá ser llamada durante la ejecución del programa desde el menú principal, es decir desde una JFrame.

Para poder colocar los elementos necesarios sobre dicha JInternalFrame en Java son necesario contenedores, en este caso se van a utilizar los elementos JPanel, a los que será necesario modificarle las propiedades con las que viene por defecto y darle la apariencia que corresponda.

Esta interface esta dividida en tres formularios que a su vez corresponden con los tres elementos JPanel que se incluyen en la misma. En la figura se ve la organización de los mismos. Para lograr ese aspecto: Accediendo a propiedades con botón derecho sobre el JPanel y seleccionando “Border” se procede a la inclusión de un borde de texto, esto se hace para los tres contenedores del frame.



**Figura 8.1:** Matriz de Resultados.

Una vez hecho esto, se procede a la colocación de los controles: Botones, campos de texto, tablas, títulos etc.

Cada una de las tres zonas descritas en los párrafos anteriores, han sido programadas de forma independiente, utilizando métodos diferentes para el funcionamiento de cada una de ellas. En las siguientes líneas se describe tanto los controles utilizados, como el código que los hace funcionar en la aplicación e interrelacionarse entre sí.

**Zona 1.** Corresponde con la zona de búsqueda de pacientes. (jPanel1)

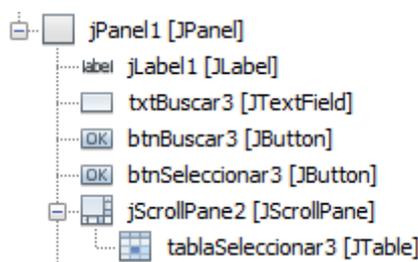
id_Paciente	DNI	Nombre	P.Apellido	S.Apellido
7	23232323M	Sergio	Almoradi	Fernandez
8	6789788C	Fuensanta	Carmoli	Alcantara
9	999999997	Julia	Zamora	Villar

**Figura 8.2:** Zona búsqueda de pacientes, en el formulario: Historial.

Esta zona es igual que la zona de búsqueda de pacientes descrita en el capítulo de “Registro de pacientes”, por tanto no procede volver a describir el código implementado. El funcionamiento consiste en localizar coincidencias de búsqueda con los criterios ingresados, que en este caso pueden ser dni, nombre, apellidos o combinación de ellos.

Tras pulsar el botón buscar aparecerá una lista con los pacientes cuyos datos coincidan con los criterios ingresados. Una vez localizado el sujeto en cuestión, señalándolo con el ratón en la tabla y pulsando el botón seleccionar, se rellenaran todos los datos personales y clínicos relevantes a estos en la siguiente zona. Al mismo tiempo aparecerán los test realizados por el sujeto seleccionado en la tabla de la zona baja del formulario.

La figura 8.3 muestran una imagen de la tabla de navegación del IDE, con los elementos añadidos al JPanel1 en el formulario Historial, ya que en apartados anteriores no se hizo.

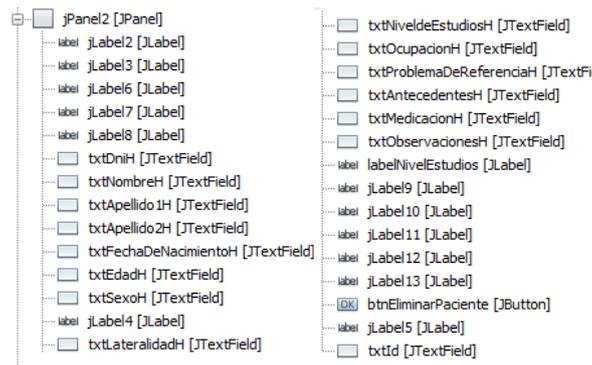


**Figura 8.3:** Componentes de la Zona “búsqueda de pacientes” en el formulario: Historial.

**Zona 2:** Zona de muestra de datos personales y clínicos del paciente. Está también da la posibilidad de borrar un sujeto, conllevando esto a la eliminación total de los registros asociados a este (datos personales, datos clínicos, y resultados de pruebas realizadas.

**Figura 8.3:** Zona de eliminación de pacientes y selección para visualización de resultados.

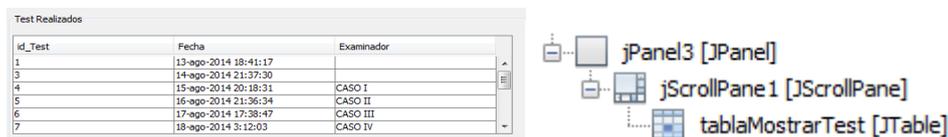
Para este formulario en la siguiente figura se muestran sus componentes, para poder ver cómo es su estructura. En cuanto al funcionamiento es el mismo al descrito en la selección de paciente del formulario de inicio. El código completo aparece en el anexo de códigos, concretamente en la clase “Historiales”.



**Figura 8.4:** Estructura del contenido del jPanel2, correspondiente al formulario de la clase “Historial”.

**Zona 3:** Zona de selección de Test. Esta muestra los registros de pruebas realizadas por el sujeto seleccionado en la parte superior de la interface.

Para implementar el funcionamiento de esta, se utiliza el método CargarTablaTest(), cuyo código se incluye en el anexo correspondiente, en la clase Historia().



**Figura 8.5:** Izda:Tabla que muestra los test realizados. Drcha:Estructura del contenido del jPanel3.

El resultado final del formulario de “selección historiales”, se muestra en la figura 8.6., este es implementado mediante la clase Historial().

Historial Paciente

Introducir criterio de Búsqueda [Dni, Nombre, Apellidos]

id_Paciente	DNI	Nombre	P.Apellido	S.Apellido
7	23232323M	Sergio	Almaraz	Fernandez
8	6200788C	Puenteana	Carroll	Acostera
9	50000000Z	Juan	Zamora	Villar

Datos Medicos

DNI:  Nombre:

Id:  Fecha de nacimiento:  Edad:  Sexo:  Lateralidad:

Nivel de estudios:

Ocupación:

Problema de referencia:

Antecedentes, enfermedades:

Medicación, dosis:

Observaciones de conducta:

Test Realizados

id_Test	Fecha	Examinador
1	13-ago-2014 18:41:17	
3	14-ago-2014 21:37:30	
4	15-ago-2014 20:18:31	CASO I
5	16-ago-2014 21:36:34	CASO II
6	17-ago-2014 17:38:47	CASO III
7	18-ago-2014 3:12:03	CASO IV

**Figura 8.6:** Formulario de selección de Historiales.

### 8.1.2 Interface-Ventana de muestra de resultados, (Clase: MostrarResultadosTest())

Se implementa la interface que recupera la hoja de respuestas y a su vez obtiene los resultados asociados a dicha prueba, todo esto mostrado de una forma organizada en pantalla. A continuación se muestra una figura 8.6 que es la captación de pantalla del resultado final de este punto.

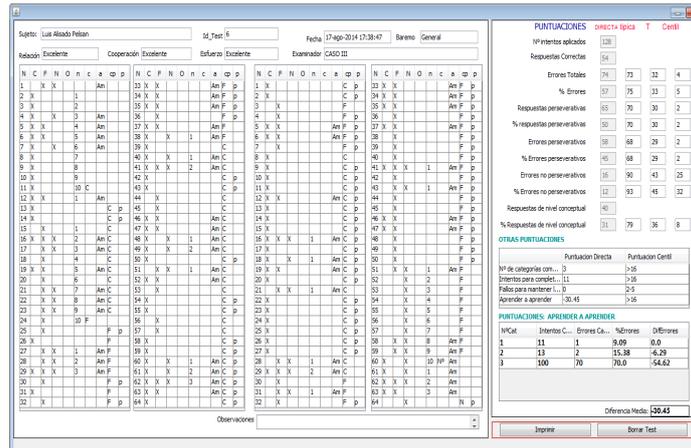


Figura 8.6: Ventana de respuestas y resultados.

Como se observa en la figura anterior, la hoja de respuesta aparece en la parte izquierda de la pantalla y la de resultados en la derecha. La interface muestra simultáneamente toda la información almacenada relativa a un test.

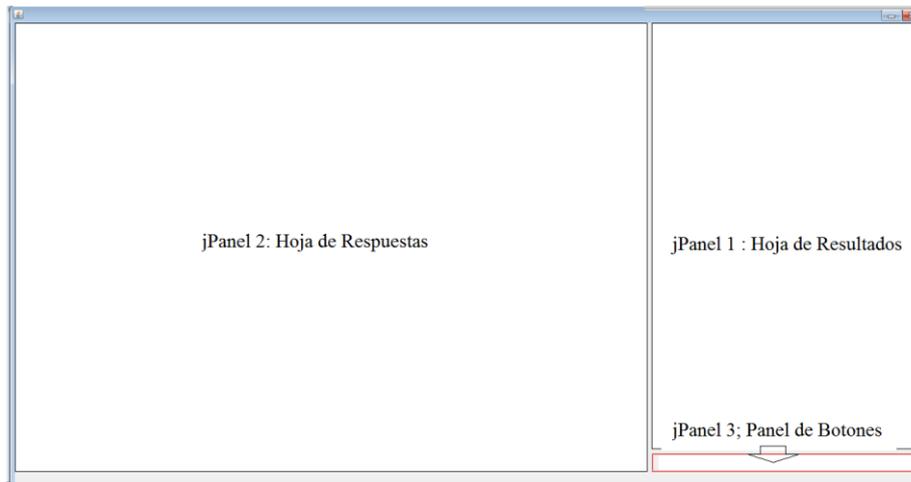
Desde esta, es posible mediante dos botones colocados en la zona inferior derecha, la impresión de esta información en dos hojas separadas. Se debe destacar que en esta hoja de resultados la información es más amplia que la aportada por el método tradicional, ya que informa de variables que no lo hace el método tradicional de captación.

En el diseño se ha optado, por darle un fondo blanco, con la finalidad de que tenga la misma apariencia que si estuviese en formato de papel.

Lo primero es crear la clase ResultadosHistorial() en el Package Test; Esta como el resto de interfaces del menú hereda del tipo JInternalFrame, por tanto podrá ser llamada durante la ejecución del programa desde otra JFrame.

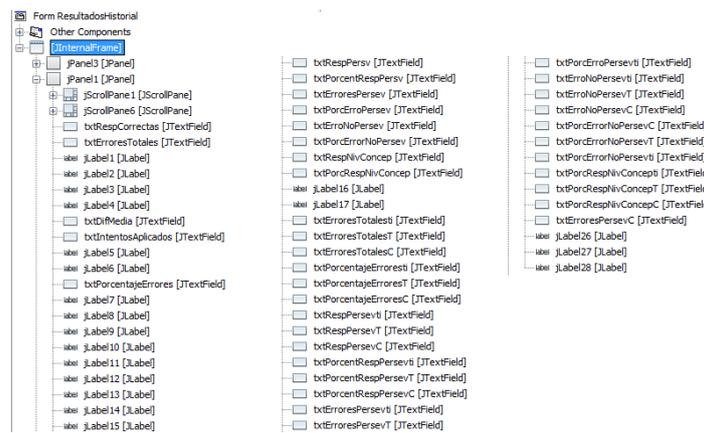
Como se ha hecho en los anteriores apartados, se procede a dividir el JInternalFrame con tres contenedores o JPanel, a los que será necesario modificarle las propiedades con las que viene por defecto y darle la apariencia que corresponda.

En la figura8.7 se ve la organización de los mismos. Para lograr ese aspecto: Se debe acceder a propiedades con botón derecho sobre cada uno de los JPanel, entre estas se les pone fondo blanco y se configuran los bordes.



**Figura 8.7:** Estructura de la Ventana de respuestas y resultados.

Dicha interface está dividida en tres zonas según paneles, esto paneles van a contener los siguientes elementos( Estos elementos se muestran en la 3 figuras siguientes: (8.8, 8.9, 8.10)



**Figura 8.7:** Estructura de la Ventana de respuestas y resultados.



**Figura 8.7:** Estructura de la hoja de respuestas. **Figura 8.7:**Estructura del panel de botones.

## 8.2 FORMA DE EJECUCIÓN EN EL CÁLCULO Y MUESTRA DE RESULTADOS.

### Clase: *MostrarResultadosHistorial()*:

La descripción del funcionamiento de esta clase se hace a continuación, como se ha comentado esta recibe el array respuestas correspondiente al sujeto seleccionado y a partir de ese momento se realiza una serie de acciones de forma secuencial, que corresponden en el código con las llamadas a los diferentes métodos. Como resultado final se obtendrá en pantalla toda la información asociada a un test.

Como en otros apartados no se describe el código al completo, este puede ser revisado en el apéndice. Lo que se pretende es mostrar la forma de trabajar que tienen las clases y métodos que componen la aplicación y la forma de interactuar.

El funcionamiento de esta clase, se base en llamadas a métodos que trabajan y modifican la matriz de respuestas con la finalidad de mostrar toda la información.

```

➤ public ResultadosHistorial() {
➤     initComponents();
➤
➤     RecuperarDatosUnTest(Historial.idt);
➤     fromString( Sal, restored );
➤     MostrarTest();
➤     HacerCalculos();
➤     MostrarResultados();
➤     Aprender();
➤     MostrarResultadosAprender();
➤     MRS();
➤     OtrosRAaA();
➤     MostrarResultadosOtrasPuntuaciones();
➤ }
    
```

Como se puede ver en la figura el proceso seguido para el montaje de la interface y mostrar toda la información es el siguiente:

1-. RecuperarDatosUnTest(Historial.idt): Mediante este método lo que se hace es recuperar los datos correspondiente a un test, dicho test es obtenido pasándole a este método como argumento el número que lo identifica.

2-. fromString(Sal,restored) .Como se comento en el capítulo del desarrollo del test, para almacenar este, se tuvo que convertir la matriz de respuestas en un array separado por comas(Sal), ahora para mostrar los resultados es necesario realizar la acción opuesta, tranformar el array en matriz(restored), para ello se utiliza este método.

3-. `MostrarTest()`: método encargado de rellenar las cuatro columnas de respuestas.

4-. `HacerCalculos()`: Se encarga de realizar los cálculos necesarios para mostrar las puntuaciones Directas en la hoja de resultados

5-. `MostrarResultados()`: Muestra los resultados obtenidos con el método anterior.

6-. `Aprender()`: Se encarga de realizar los cálculos necesarios para obtener las puntuaciones correspondientes a la categoría Aprender a aprender.

7-. `MostrarResultadosAprender()`: Muestra los resultados obtenidos con el método `Aprender()`.

8-. `MRS()`: Esta clase es encargada de obtener los resultados estadísticos correspondientes a las puntuaciones directas. Será descrita en profundidad en el capítulo siguiente, que abarca todo lo relacionado a puntuaciones estadísticas.

9-. `OtrosRAaA()`:

10-. `MostrarResultadosOtrasPuntuaciones()`: como su nombre indica es encargada de mostrar los resultados correspondientes a otras puntuaciones.

A continuación se van a describir algunos de los métodos descritos anteriormente, otros han sido descritos y usados anteriormente en el desarrollo de la aplicación, en cuanto al método `MRS()`, será descrito junto a toda la información correspondiente a los resultados estadísticos.

No se trata de hacer una descripción minuciosa de código, sino describir la forma de operación que tiene esta aplicación. Cabe destacar que en las tablas de código mostradas a continuación se omiten líneas en las que se realizan las mismas operaciones varias veces para no ser repetitivo, en los anexos y el cd del proyecto se muestra el código completo con anotaciones.

## ***CAPÍTULO 9***

### **RESULTADOS ESTADÍSTICOS**

---

---

En el presente capítulo se describen diferentes puntos desarrollados con la finalidad de poder obtener resultados estadísticos y de esta forma alcanzar la validez de la prueba.

Los resultados estadísticos están basados en muestras de tipificación y muestras normativas, en el capítulo se muestra el proceso en el que una vez que la aplicación dispone de las puntuaciones directas, accediendo a una nueva base de datos, desarrollada también en este capítulo y que contiene las tablas de baremos, accede a estas para obtener variaciones respecto de los valores de referencia. De esta forma poder ayudar al clínico en la búsqueda del diagnóstico, así como ahorrar tiempo y errores en la corrección del test.

Además de obtener resultados estadísticos, el programa se ha completado con un visor de tablas, el cual contienen métodos y otra información que puede ser útil para el clínico, ayudando al mismo a valorar el nivel de importancia que se puede dar a los resultados obtenidos en la prueba.

#### **9.1 TRANSFORMACIÓN DE LAS PUNTUACIONES DIRECTAS EN TÍPICAS**

Una vez obtenidas las puntuaciones directas del WCST, el programa como haría el clínico o examinador, como primer paso debe localizar y seleccionar la tabla de baremos adecuada para el sujeto, se deberá tener en cuenta su edad (si se trata de niños y adolescentes de entre 6 años y medio a 19 años), o la combinación de edad y años de estudios (en el caso de adultos de 20 años ó más). También será posible utilizar una tabla denominada “General” que sirve para hacer comparaciones con la población general de adultos de entre 20 y 89 años de edad.

El programa empieza calculando la puntuación típica,  $T$  y centil, correspondientes a “Número total de errores”. Para ello se localiza la puntuación directa obtenida por el sujeto, en la variable que ha sido guardada, estos valores aparecerán en la columna de tabla de resultados correspondiente; en esa misma fila y tres columnas situadas a la derecha de la misma.

Estos valores quedarán registrados en la Hoja de resultados. Se sigue el procedimiento similar para localizar y almacenar las puntuaciones típicas,  $T$  y centil

correspondientes a “Porcentaje de errores”, “Respuestas perseverativas”, “Porcentaje de respuestas perseverativas”, “Errores no perseverativos”, “Porcentajes de errores no perseverativos” y “Porcentaje de respuestas de nivel conceptual”.

Según lo comentado en los dos párrafos anteriores, el procedimiento para llegar a los resultados estadísticos es similar en caso de realizarlo un profesional o la aplicación informática. Resumiendo, en ambos casos, teniendo la puntuación directa, y con esta como criterio de entrada a la tabla correspondiente de baremos, obtenemos las puntuaciones típicas, T y Centil para cada una de las categorías de dichas puntuaciones.

En la siguiente figura se ve la parte de la hoja de resultados que muestra dichas puntuaciones, más avanzado el capítulo se describe el proceso detallado seguido por la aplicación del WCST para a mostrar los resultados estadísticos.

PUNTUACIONES	DIRECTA	típica	T	Centil
Nº intentos aplicados	128			
Respuestas Correctas	54			
Errores Totales	74	73	32	4
% Errores	57	75	33	5
Respuestas perseverativas	65	70	30	2
% respuestas perseverativas	50	70	30	2
Errores perseverativos	58	68	29	2
% Errores perseverativos	45	68	29	2
Errores no perseverativos	16	90	43	25
% Errores no perseverativos	12	93	45	32
Respuestas de nivel conceptual	40			
% Respuestas de nivel conceptual	31	79	36	8

**Figura 9.1:** Hoja de resultados. Puntuaciones típica, T y centil.

En la hoja de resultados también existe una zona para mostrar otros aspectos estadísticos, que para completarlos se debe hacer uso de otra tabla asociada a la anterior y cuyo nombre es el mismo que la primera pero con el sufijo –bis. Es decir en caso de haber seleccionado la tabla de baremos “nueve”, esta segunda tabla asociada a la primera se denominará “nuevebis”. Aquí se localiza la columna “Numero de categorías completas” y dentro de ella, el número correspondiente a las realizadas por el sujeto, es decir su puntuación directa en este apartado, en la misma línea, en los extremos de la tabla aparece la puntuación centil correspondiente; esta puntuación será trasladada a la hoja de resultados. Se sigue un procedimiento similar para la transformación y anotación de las puntuaciones correspondientes a Intentos para completar la primera categoría, Fallos en Mantenimiento de la actitud y Aprender a Aprender. En la figura xx se puede ver como se muestra en la aplicación informática los resultados obtenidos en un caso supuesto para estas puntuaciones.

#### OTRAS PUNTUACIONES

	Puntuacion Directa	Puntuacion Centil
Nº de categorías com...	3	>16
Intentos para complet...	11	>16
Fallos para mantener l...	0	2-5
Aprender a aprender	-30.45	>16

**Figura 9.2:** Hoja de resultados. Zona de otras puntuaciones.

En los siguientes apartados del capítulo se describe el proceso que se ha llevado a cabo tanto para el correcto ingreso de las tablas de baremos en la base de datos, como la programación desarrollada para la interacción con esta y obtener unos correctos resultados a la hora de su aplicación.

**9.1.1 Ingreso de ábacos y tablas tabuladas en la base de datos**

Para el cálculo de resultados estadísticos es necesario disponer de una serie de tablas, las cuales deben ser almacenadas en una base de datos para posteriormente poder ser accesibles por la aplicación informática.

El proceso para ingresar las tablas en la base de datos no es directo mediante el administrador de bases de datos de MySQL, es decir, antes de tener a disposición el contenido de las tablas en la base de datos es necesaria una transformación de las mismas, ya que como se verá en párrafos siguientes servirá para evitar demasiadas comparaciones.

Se muestra como es una de las tablas originales de los baremos, en particular la que hace referencia al estudio de niños con edad comprendida entre 9 años, 0 meses y 0 días hasta 9 años, 11 meses y 30 días.(10 años). En esta no se tiene en cuenta en nivel de educación, ya que los años de escolarización posibles no son determinantes. En el manual de referencia la tabla correspondiente está denominada como C4.

<b>Tabla C4</b> Edad: 9 años, 0 meses y 0 días hasta 9 años, 11 meses y 30 días Nivel educación: —												
Puntuaciones directas												
Centil	Punt. T	Puntuación típica	Nº de errores	% de errores	Hospuest. perseverati-vas	% de respuest. perseverati-vas	Errores perseverati-ivos	% de errores perseverati-ivos	Errores no perseverati-ivos	% de errores no perseverati-ivos	% de respuest. nivel conceptual	
>99	>80	>145	0-1	0-9	0-2	0-3	0-2	0-3	0-1	0-2	92-100	
>99	80	145									91	
>99	79	144									90	
>99	78	143									89	
>99	77	142	5	10							88	
>99	76	141				4	3		2		87	
>99	75	140	6								86	
>99	74	139									85	
>99	73	138									84	
>99	72	137	7	11	3						83	
>99	71	136									82	
>99	70	135									81	
1	26	64	98-99	73-74	76-77	57-58	62-63	47	55	42	4-5	
1	25	63	100-101	75	78-79	59	64	48	56-57	43	2-3	
1	24	62	102-103	76-77	80-81	60-61	65-66	49	58	44	0-1	
<1	23	61	104-105	78-79	82-83	62	67-68	50	59-60	45	16	
<1	22	60	107-108	80	84-85	63-64	69	51-52	61	47	15	
<1	21	59	109-110	81-82	86-87	65	70-71	53	62-63	47	14	
<1	20	58	111-112	83	88-90	66-67	72-73	54	64	48-49	13	
<1	19	57	113-114	84-85	91-92	68	74	55	65-66	50	12	
<1	18	56	115-116	86	93-94	69-70	75-76	56	67	51	11	
<1	17	55	117-118	87-88	95-96	71-72	77	57-58	68-69	52	10	
<1	16	54	119-120	89-90	97-100	73-100	78-94	59-100	70-128	53-100	9	

Figura 9.3: Ejemplo de una tabla original de baremación. (1/2)

Puntuaciones directas					
Centil	Nº de categorías completas	Intentos para completar la 1ª categoría	Fallos para mantener la actividad	Aprender a aprender	Centil
>16	4-6	30-38	0-3	≥-8.60	>16
11-16	3	39-34		-8.61 to -12.29	11-16
6-10	2	25-34		-12.30 to -23.79	6-10
2-5	1	35-33	4	-23.80 to -29.60	2-5
≤1	0-1	54-129	5-21	≤-29.61	≤1

Figura 9.4: Ejemplo de una tabla original de baremación. (2/2), “otras puntuaciones”.

A continuación se describen los pasos seguidos para convertir dichas tablas estadísticas del manual de referencia, en tablas cuyo contenido está disponible para su tratamiento.

1- El primer paso consiste en trasladar la tabla de la figura anterior a una hoja de Excel. Se debe tener en cuenta que al ingresar los datos en Excel, se deben eliminar los rangos de valores que se pueden observar en la tabla original, o lo que es lo mismo, en cada casilla de la hoja de cálculo solo debe haber un elemento o valor. Para ver esto de forma más clara, a continuación se presenta un ejemplo.

Para la siguiente figura que corresponde a una parte de tabla original, la línea resaltada con un rectángulo rojo, contiene la misma puntuación Centil, T y típica para diferentes puntuaciones directas para las distintas categorías, como se ha comentado anteriormente para poder trabajar con tablas sin la necesidad de hacer comparaciones y comprobaciones, debemos individualizar cada una de las puntuaciones directas de las distintas categorías.

Tabla C4		Edad: 9 años, 0 meses y 0 días hasta 9 años, 11 meses y 30 días Nivel educación: —									
Centil	Punt. T	Puntuación típica	Puntuaciones directas								% de respuestas nivel concept
			Nº de errores	% de errores	Respuest. perseverati-vas	% de respuest. perseverati-vas	Errores perseve-ralivos	% de errores perseve-ralivos	Errores no persevera-tilivos	% de errores no persevera-tilivos	
>99	>80	>145	0-1	0-9	0-2	0-3	0-2	0-3	0-1	0-2	92-100
>99	80	145									
>99	79	144									
>99	79	143									
>99	78	142									
>99	77	141									
>99	77	140				4	3		2		

Figura 9.5: Selección para el traslado de una fila a la base de datos.

A continuación se puede observar la tabla de Excel (figura 9.6) cuyo contenido para la línea remarcada es el mismo que el de la figura anterior o tabla original, este procedimiento se debe hacer hasta el final de la tabla para cada una de las líneas.

>99	>80	>145	0	0	0	0	0	0	0	0	100
>99	>80	>145	1	1	1	1	1	1	1	1	99
>99	>80	>145	2	2	2	2	2	2	2	2	98
>99	>80	>145	3	3			3		3		97
>99	>80	>145	4	4							96
>99	>80	>145									95
>99	>80	>145									94
>99	>80	>145									93
>99	>80	>145									92
>99	>80	>145									
>99		80	145						4		91
>99		78	142	5	10						90
>99		77	141							3	
>99		77	140				4	3		2	

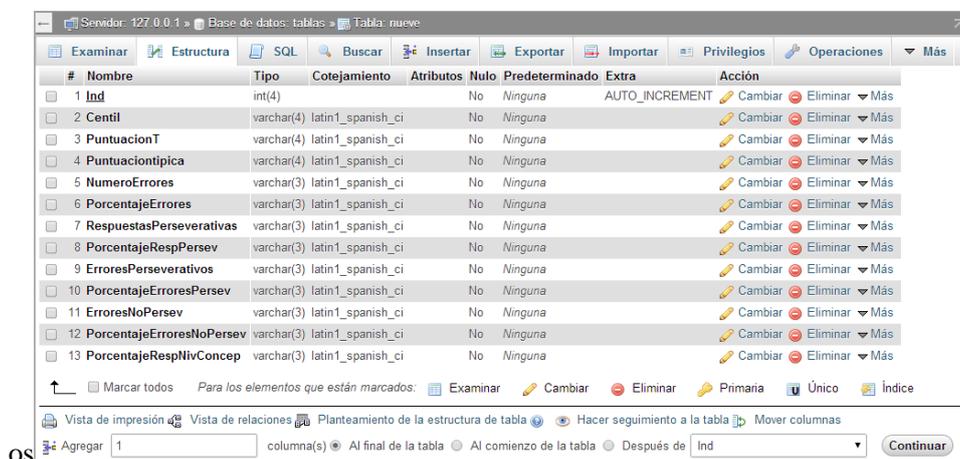
Figura 9.6: Implementación una fila original a la tabla de Excel.

Se puede observar que cada fila de la tabla original se ha desdoblado en varias filas en la nueva tabla de Excel, de esta forma solo corresponde un valor a cada celda. Una vez completada toda la translación de datos queda una tabla con más filas que la inicial. Las tablas completas se pueden ver en el anexo xxx.

2-. Se guarda la hoja de datos de Excel con formato: CSV (“Delimitado por comas”) Esto es necesario para posteriormente poder importar los datos a una tabla en MySQL.

**3-.** En el administrador de bases de datos de MySQL se debe crear una tabla cuyo futuro contenido será el de la hoja de cálculo, ya que inicialmente lo que se hace es solo reservar memoria. Esta tabla estará dentro de la base de datos “tablas” creada con anterioridad para contener a estas.

Esta tabla debe contener 12 columnas, correspondientes a todas las categorías de las distintas puntuaciones directas, además de las tres puntuaciones estadísticas. Como es posible que contengan símbolos debemos indicarle que el contenido va a ser de tipo varchar. Lo que estamos haciendo es reservar memoria para el ingreso de los datos en la tabla, ya que en principio esta está vacía. En la siguiente figura se puede ver el orden y los nombres de cada una de las columnas destinadas a almacenar todos los componentes de la tabla. También se puede observar que menos la primera variable que es de tipo entero y autoincremental, (requisito necesario para posibles futuras modificaciones), todas la variables son de tipo entero, en cuanto al espacio reservado para las variables correspondientes a las puntuaciones estadísticas será 4 y para el resto de variables 3.

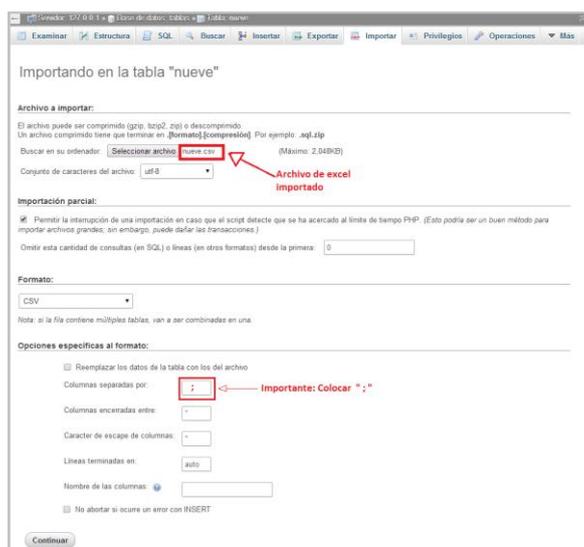


**Figura 9.7:** Estructura de una tabla de baremación en la base de datos.

Este orden no puede ser alterado, ya que corresponde con el que se le ha dado a las columnas en la hoja de cálculo.

**4-.** En el administrador de bases de datos, se importan los datos contenidos en el archivo .CSV (nueve.csv) a la tabla que ha sido creada en el apartado anterior. Para ello se accede a la pestaña superior “Importar” se busca la dirección del archivo en la que ha sido guardada.

En este caso como el formato en el que viene el archivo de origen es .csv, corresponde a un formato separado por comas, por tanto accediendo a opciones específicas de formato se modifica el cuadro en el que inicialmente aparece una “,” por un “;”. Esto indica que la tabla ha sido guardada separando las filas por “;”. Una vez configurada la importación se procede a ejecutarla mediante el botón “Continuar”. Esta configuración se puede observar a continuación:



**Figura 9.8:** Ventana de configura en la importación de una tabla de baremos.

Cabe destacar que en este punto ya estará incluida la tabla de baremación correspondiente a una edad de 9 años, este procedimiento hay que hacerlo para todas las tablas que bareman las puntuaciones, en función de la edad o edad y nivel de estudios del paciente.

Antes de continuar, se debe comprobar que la importación ha sido correcta, es decir que todos los datos están contenidos en sus correspondientes columnas y que no se ha cometido ningún fallo en el orden. Para ello accediendo en el administrador de bases de datos a la base de datos “tablas” deberá aparecer la nueva tabla denominada “nueve”, selección dicha tabla se pulsa la opción “examinar” y se comparan los datos con la original creada en Excel.

### **9.1.2 Funcionamiento del programa en la muestra de puntuaciones estadísticas.**

El código siguiente (Tabla 9.1) implementa la muestra de puntuaciones estadísticas (T, típica y centil, correspondientes a la primera puntuación directa de la hoja de resultados (Figura 9.9). En el código completo, mostrado en el anexo correspondiente se podrá ver que el proceso es el mismo para cada una de las puntuaciones directas.

En funcionamiento de dicha clase consiste básicamente en: Recupera el contenido de la variable Baremo, esta variable contiene el nombre de la tabla a acceder, posteriormente mediante métodos comentados en apartados anteriores, se procede a realizar una consulta sobre esta tabla en la base de datos “tablas”, previamente cargada.

El criterio de entrada es la puntuación directa obtenida en la categoría correspondiente, y esto nos dará como salida en 3 campos de texto situados a la derecha del campo de texto con dicha puntuación directa.

```

1.         void MRS() {
2.
3.         String i = Baremo;
4.
5.         String sSQL1;
6.         String Eti="","ET="","EC="";
7.
8.         ConexionMySQLTablas mysql1 = new ConexionMySQLTablas();
9.         Connection cn1 = mysql1.Conectar();
10.
11.        sSQL1 = "SELECT Centil, PuntuacionT, Puntuaciontipica,
NumeroErrores FROM "+i+" "
12.
13.        try
14.        {
15.
16.        Statement st1 = cn1.createStatement();
17.        ResultSet rs1 = st1.executeQuery(sSQL1);
18.
19.        while(rs1.next())
20.        {
21.        ET = rs1.getString("PuntuacionT");
22.        Eti = rs1.getString("Puntuaciontipica");
23.        EC = rs1.getString("Centil");
24.        }
25.
26.        txtErroresTotalesti.setText(Eti);
27.        txtErroresTotalesT.setText(ET);
28.        txtErroresTotalesC.setText(EC);
29.
30.        }
31.        catch (SQLException ex)
32.        {
33.        JOptionPane.showMessageDialog(null, ex);
34.        }
35.
36.    }
    
```

**Tabla 9.1.** Método MRS(): encargado de mostrar los resultados estadísticos.

PUNTUACIONES	DIRECTA	tipica	T	Centil
Nº intentos aplicados	128			
Respuestas Correctas	54			
Errores Totales	74	73	32	4
% Errores	57			
Respuestas perseverativas	65			
% respuestas perseverativas	50			
Errores perseverativos	58			
% Errores perseverativos	45			
Errores no perseverativos	16			
% Errores no perseverativos	12			
Respuestas de nivel conceptual	40			
% Respuestas de nivel conceptual	31			

**Figura 9.9:** Ventana de configura en la importación de una tabla de baremos.

## 9.2 MUESTRA NORMATIVA

### 9.2.1 Muestra de tipificación

Los datos normativos del WCST contenidos en la bibliografía y utilizados para el desarrollo de la aplicación se han obtenido a partir de una muestra de 899 sujetos normales que proceden de seis muestra distintas. En la siguiente figura se muestra el número de sujetos, edad, nivel cultura, finalidad y sexo de los sujetos que han formado parte de la muestra.

Nº	Nº sujetos	Edades	Nivel cultural	Finalidad	Sexo
1	453	Niños y adolescentes	Normal, escuelas públicas.	Disfunciones, problemas aprendizaje	52% niños, 48% niñas.
2	49	18 años	Estudiantes, 12 a 13 años escolarización.	Proyecto investigación	49% Varones, 51% Mujeres.
3	150	Entre 15 y 77 años.		Estudio relativo envenenamiento por pesticidas	17% Mujeres y 83% Varones.
4	50	58 a 84 años.		Investigación, tesis	66% Mujeres 34% Hombres.
5	124	24 a 65 años.	Pilotos líneas aéreas. 14-20 años	Evaluación neuropsicológica computerizada.	9% Mujeres y 91% Hombres.
6	73	51 a 89 años.	6 a 20 años de escolarización.	Proyecto de promoción de salud	55% Mujeres, 45% Hombres

**Figura 1.1** Datos Normativos. Muestra 899 sujetos.

### 9.2.2 Variables Normativas

Las variables para establecer la baremación del WCST son:

- Número total de errores
- Porcentaje de errores.
- Respuestas perseverativas
- Porcentaje de respuestas perseverativas.
- Errores perseverativos
- Porcentaje de errores perseverativos
- Errores no perseverativos
- Porcentaje de errores no perseverativos

- Porcentaje de respuestas de nivel conceptual.
- Número de categorías completas
- Intentos para completar la primera categoría
- Fallos para mantener la actitud
- Aprender a aprender

### **9.2.3 Normas por categorías**

Los baremos correspondientes se elaboraron utilizando solamente la edad a causa del inadecuado tamaño de la muestra en algunos grupos de edad y nivel de educación. Dentro de cada grupo de edad, las puntuaciones directas se ordenaron y se asociaron a los centiles calculados. Se determinaron los niveles de puntuación directa para cinco categorías.

- a) Puntuaciones superiores al centil 16.
- b) Puntuaciones comprendidas entre los centiles 11 y 16
- c) Puntuaciones comprendidas entre los centiles 10 y 6.
- d) Puntuaciones comprendidas entre los centiles 2 y 5.
- e) Puntuaciones comprendidas iguales o menores al centil 1.

### **9.2.4 Estadísticos de la muestra general**

Durante los estudios se elaboró también una tabla con los datos normativos correspondientes a la muestra de adultos de más de 20. Se calcularon las medias y desviaciones típicas. Para las variables normativas antes mencionadas. Las puntuaciones típicas normalizadas y las puntuaciones T de cada uno de los aspectos del WCST se obtuvieron siguiendo el método de la curva de Johnson para obtener los centiles, utilizando los valores de las respectivas medias, desviación típica, sesgo y curtosis.

## **9.3 INTERPRETACIÓN**

Aunque no es fin de este proyecto, llegar a la interpretación de resultados el software ofrece una opción para visualizar herramientas necesarias de las que el profesional puede apoyarse para obtener una mejor interpretación de los resultados.

Para la interpretación se presentan pautas. Hay que tener presente en cualquier caso que la prueba es solamente una de las que proporcionara información sobre importantes áreas del funcionamiento neuropsicológico. Los resultados del WCST deben ser interpretados en el contexto de una gran cantidad de pruebas de evaluación y la información clínica e historia del sujeto.

### **9.3.1 Comparación con los baremos**

Los dos objetivos neuropsicológicos principales de la interpretación de test están relacionados directamente con diagnósticos de Trastorno cerebral y con el nivel de adecuación de la habilidad examinada en términos absolutos o lo que es lo mismo; Comparar los resultados de esta persona con los de la población general y comprobar si son buenos o malos.

Si se aplica la prueba con propósitos de diagnóstico, se recomienda utilizar los baremos que consideran edad y nivel cultural. Estas tablas darán información sobre la realización de la prueba del sujeto comparándola con el grupo normativo de edad o de nivel cultural.

Las tablas de baremos permiten la transformación de las puntuaciones directas en típicas normalizadas y en puntuaciones T (igual que en los centiles correspondientes a éstas). Las puntuaciones típicas normalizadas han sido calculadas igualando la media a 100 y la desviación típica a 15, las puntuaciones T tienen una media igual a 50 y una desviación típica igual a 10. Las puntuaciones incluidas proporcionan información sobre la realización del sujeto comparándola con el grupo normativo de edad o nivel cultural que tengas sus mismas características. Ejemplo: una puntuación típica da 115 indica que los resultados del sujeto superan a los obtenidos por el 85% de su grupo normativo.

Para comprobar si las habilidades de un sujeto se adecuan al funcionamiento normal del mismo o si pueden ser más complejas, ya que se requiere que el profesional o clínico estime la importancia que debe tener cada habilidad en particular en el funcionamiento de dicho sujeto. Los datos normativos contenidos en la tabla “General” de la base de datos tablas pueden ser muy útiles para responder a preguntas sobre la capacidad de adultos, pero no para niños y adolescentes hasta 20 años.

Por ejemplo en el caso de un varón de 65 años, con nivel educativo correspondiente a 8 años de escolarización, que ha obtenido en el WCST, en Respuestas perseverativas, la puntuación directa de 46. Los datos normativos que se presentan en la tabla correspondiente a edad 65-69 y nivel de escolarización 9-11 años, indican que a esta puntuación le corresponde la puntuación centil 21, y ello significa que los resultados del sujeto se sitúan por debajo de la media, en el nivel medio-bajo cuando se comparan con los de un grupo de sujetos con sus mismas características, sin embargo esta puntuación no sugiere la presencia de ningún desorden. Por otro lado los datos normativos que aparecen en el Apéndice B revelan que esta puntuación difercta se sitúa significativamente por debajo de la media (centil 5) si se compara con el baremo de la población general de adultos. Por tanto se puede esperar que, cuando el suejto intente realizar eficazmente un trabajo o una función en situaciones que habitualmente requieran la flexibilidad cognitiva media de los adultos, encontrará algunas dificultades para ello.

Sistema de Heaton de clasificación para interpretar las puntuaciones normativas, basándose en su experiencia clínica. Estas clasificaciones se pueden usar con los baremos antes introducidos en la base de datos de la aplicación. Las siguientes agrupaciones de puntuaciones típicas (media=100 y desviación típica =15) y

puntuaciones T (media=50 y desviación típica= 10), definen estos niveles clínicamente relevantes.

Puntuaciones Típicas	Puntuaciones T	Nivel de Deterioro
$\mu \leq 54$	$\leq 19$	Grave
$55 \leq \mu \leq 61$	$20 \leq t \leq 24$	Grave a Moderado
$62 \leq \mu \leq 69$	$25 \leq t \leq 29$	Moderado
$70 \leq \mu \leq 76$	$30 \leq t \leq 34$	Moderado a Intermedio
$77 \leq \mu \leq 84$	$35 \leq t \leq 39$	Intermedio
$85 \leq \mu \leq 91$	$40 \leq t \leq 44$	Superior a la media
$92 \leq \mu \leq 106$	$45 \leq t \leq 54$	Medio
$\mu \geq 107$	$t > 55$	Inferior a la media

Si se establecen categorías en relación con los niveles de puntuación típica, se deberá tener en cuenta que el enfoque de la interpretación se basa en la realización del sujeto y por tanto, es la realización en el WCST la que puede considerarse deficiente o deteriorada, más que le propio sujeto. Esta distinción resulta especialmente importante, cuando se evalúa a niños o adultos en edad avanzada, dada la gran variabilidad que presentan los resultados del WCST correspondientes a sujetos neurológicamente sanos de estos niveles de edad.

### 9.3.2 Comparación con valores de referencia

Los datos normativos obtenidos de la bibliografía (Manual del Test Wisconsin) permiten la comparación de los resultados de un sujeto con los de muestras de individuos de la población normal. Algunos individuos normales, neurológicamente intactos obtienen puntuaciones bajas en este test, algunos sujetos con problemas neurológicos consiguen en esta prueba puntuaciones medias e incluso altas. Por esto se debe tener en cuenta si la realización del test del sujeto se parece a la de otros sujetos que no tienen problemas o a la de los que los tienen.

Como ayuda para la interpretación, se incluyen en la aplicación, los valores de referencia de la muestra de tipificación y algunos grupos clínicos. Debemos recordar que estos datos son obtenidos de estudios desarrollados en el pasado e incluidos evn la aplicación y obtenidos en este caso del Manual original de desarrollo del WCST.

Las puntuaciones en el WCST se han clasificado en varios grupos, según datos demográficos de la población a que corresponden y a las características clínicas más importantes. Los valores de referencia se presentan, por separado, para adultos y adolescentes.

Estos valores de referencia estarán contenidos en la aplicación, y es posible acceder a ellos para su comparación con los resultados del test. En la última pestaña del menú principal denominada “Info” existen dos opciones, la primera de ellas “Abacos” nos lleva a un visualizador de imágenes o en este caso tablas, el cual nos permite movernos por las diferentes tablas de referencia, al ser una ventana, también es posible simultáneamente comparar con resultados del historial de cualquier sujeto.

Dichas tablas presentan los valores de referencia correspondientes a algunos datos en los que el carácter específico del WCST se mantenía constante en un 88%, aproximadamente. Para ello se tomó una desviación típica como punto de corte y empleó esta magnitud para definir una realización deficiente. Considerando las respuestas perseverativas como una puntuación representativa del WCST, en subgrupos de sujetos con daños cerebrales, en los que estaba implicado el lóbulo frontal, la frecuencia de realizaciones deficientes alcanza un nivel comprendido aproximadamente entre el 53 y el 59%. Por contraste, la frecuencia de este tipo de respuestas dentro del grupo focal no frontal fue inferior al 38%. En general, estos datos muestran una modesta sensibilidad de las respuestas perseverativas del WCST a los problemas cerebrales. Por otro lado, solo el 20% de los pacientes con lesiones cerebrales en el lóbulo frontal que mostraban deterioro tenían lesiones focales no frontales. Esto aparece de forma muy similar en los grupos de lesiones de tipo focal frontal y en daño cerebral difuso.

Para mostrar la utilidad de estos valores de referencia, consideremos el caso de una mujer de 34 años, con nivel educativo correspondiente a 16 años de estudios. En el WCST, obtuvo puntuación directa de 24 Respuestas perseverativas. Esta puntuación equivale a la típica 74 o la T 33 (según su tabla de baremo). Con estas puntuaciones dirigiéndose a la tabla: “d1 “ se puede ver que esa puntuación está comprendida en el rango correspondiente a “deterioro moderado a intermedio”. Localizando los valores de referencia para respuestas perseverativas, el grupo de adultos normales, se observa que el 94.5% de los sujetos de este obtienen puntuaciones más altas que la que ella ha logrado (7%, +10.4%, +50.8%, +25.3%) (grupos comprendidos entre el inmediatamente superior al nivel conseguido por la mujer del ejemplo y que representan el nivel superior de puntuación) mientras que el grupo formado por la muestra clínica, solamente el 64.4% obtienen puntuaciones más altas que las logradas por ella en este aspecto. Por tanto se puede considerar que la realización de la persona que estamos valorando parece ser más similar a la de los adultos con problemas neurológicos que a la de los neurológicamente intactos.

Los valores de los datos presentados en estas tablas, se pueden utilizar también para establecer otros niveles y categorías interpretativas que se sugieren en este Manual. Basándose en su experiencia y en sus conocimientos, los profesionales pueden establecer categorías interpretativas más o menos rigurosas combinando o reclasificando grupos y categorías. Por ejemplo, utilizando los datos presentados en la tabla D-1, se puede crear una categorías de realización “borderline” combinando las categorías “intermedio” y “moderado a intermedio” (puntuaciones típicas entre 70 y 80 o puntuaciones T entre 30 y 39). También se puede crear una categoría de “realización deficiente” reclasificando la categoría “moderado” (puntuaciones típicas de 62 a 69 y puntuaciones T de 25 a 29). Finalmente las categorías “grave” y “grave a moderada” se pueden combinar para formar una categoría de realización “significativamente deficiente” (puntuaciones típicas de 0 a 61 o puntuaciones T de 0 a 24). Comparando los porcentajes de sujetos normales y clínicos incluidos en estos nuevos grupos, para Respuestas perseverativas, resulta.

Los profesionales también pueden utilizar los porcentajes de las muestras normales presentados en el visor de tablas, para establecer los criterios que puedan proporcionarles los mejores ratios de clasificación (por ejemplo verdaderos positivos y verdaderos negativos) en comparación con los valores básicos de disfunción cerebral reflejados en situaciones clínicas individuales.

En este trabajo solo se muestran aspectos funcionales del test, si se quiere obtener más información a cerca de casos ilustrativos, desarrollos y justificaciones estadísticas, fiabilidad de la prueba, errores típicos, así como información relativa a pruebas realizadas teniendo como referencia determinadas enfermedades cerebrales habrá que acudir al manual original del WCST.

## 9.4 PROGRAMACIÓN DEL VISOR DE TABLAS Y MÉTODOS

### 9.4.1 Descripción del visor.

EL software ofrece una opción para visualizar herramientas (tablas y ábacos), de las que el profesional puede ayudarse para obtener una mejor interpretación de los resultados.

Estas tablas corresponden a métodos, valores típicos y de referencia. De esta forma el examinador no tiene que buscar dicha información en la documentación, simplemente abriendo el visor dispondrá de ellos. Este está disponible en el menú inicio, en la opción: **Info>>Muestra de tipificación(Valores de referencia)**

Tras acceder a dicha opción se abre en una ventana el visor, según la figura 29, este se puede mantener abierto simultáneamente con la ventana de resultados y así poder realizar una comprobación sobre los datos.

Porcentaje de adultos de las muestras normal y clínica, según la localización de la lesión, en los distintos niveles de puntuaciones típicas.									
DETERIORO	Nivel de realización								
	Muestra Clínica						Muestra Normal		
	Grave	Severo	Moderada-severo	Moderado	Moderado-medio	medio	Debajo de la media	Media	Encima de la media
Puntuación típica	0-84	0-54	55-61	62-69	70-76	77-84	85-91	92-106	107+
Puntuación T	0-39	0-19	20-24	25-29	30-34	35-39	40-44	45-54	55+
% predicho en la distrib normal.	14,6	0,1	0,4	1,5	4,0	8,6	14,4	38,2	32,8
<b>WCST puntuación</b>									
<b>Nº total errores</b>									
Grupo normal	14.2	0.3	0.8	1.6	4.2	7.3	9.4	47.9	28.6
Grupo clínico total	57.8	4.4	9.9	11.1	15.2	17.2	8.7	22.2	11.4
Frontal	59.4	10.2	11.9	8.5	16.9	11.9	11.9	20.3	8.5
Frontal más	62.2	1.9	11.3	11.3	15.1	22.6	9.4	15.1	13.2
Difuso	61.1	4.5	10.2	11.9	14.7	19.8	9.0	20.9	9.0
No frontal	40.8	0.0	5.6	11.1	14.8	9.3	3.7	35.2	20.4
<b>% de errores</b>									
Grupo normal	12.4	0.5	1.0	1.0	3.1	6.08	11.2	48.4	27.9
Grupo clínico total	54.2	2.9	10.5	9.3	15.8	16.0	12.0	22.4	11.4
Frontal	57.8	6.8	15.3	5.1	15.3	15.3	8.5	20.3	13.6
Frontal más	56.6	1.9	11.3	9.4	17.0	17.0	17.0	17.0	9.4

Figura 29: Visor de valores de referencia.

El visor nos muestra información, como por ejemplo el método Heaton de clasificación, que sirve para interpretar las puntuaciones normativas basándose en experiencias clínicas anteriores.

Sistema de Heaton de clasificación para interpretar las puntuaciones normativas basándose en su experiencia clínica		
Las siguientes agrupaciones de puntuaciones típicas (media=100 y desviación típica =15) y puntuaciones T (media=50 y desviación típica= 10), definen estos niveles clínicamente relevantes.		
Puntuaciones Típicas	Puntuaciones T	Nivel de Deterioro
$\mu \leq 54$	$\leq 19$	Grave
$55 \leq \mu \leq 61$	$20 \leq t \leq 24$	Grave a Moderado
$62 \leq \mu \leq 69$	$25 \leq t \leq 29$	Moderado
$70 \leq \mu \leq 76$	$30 \leq t \leq 34$	Moderado a Intermedio
$77 \leq \mu \leq 84$	$35 \leq t \leq 39$	Intermedio
$85 \leq \mu \leq 91$	$40 \leq t \leq 44$	Superior a la media
$92 \leq \mu \leq 106$	$45 \leq t \leq 54$	Medio
$\mu \geq 107$	$t > 55$	Inferior a la media

**Figura 30: Tabla método Heaton.**

En el visor se pueden encontrar tablas de tipificación de muestras normal y clínica, haciendo la distinción entre adultos y niños o adolescentes. También estarán clasificadas según la localización de la lesión o categorías del diagnóstico. Los niveles de puntuación utilizados estarán referenciados a puntuaciones típicas o centiles. A continuación aparece la descripción de las tablas incorporadas.

- Porcentaje de adultos de las muestras normal y clínica, según la localización de la lesión, en los distintos niveles de puntuaciones típicas.
- Porcentaje de adultos de las muestras normal y clínica, según localización de la lesión, en los distintos centiles.

Además se mostrarán las cuatro siguientes:

- Porcentaje de niños y adolescentes de las muestras normal y clínica, según la localización de la lesión, en los distintos niveles de puntuaciones típicas.
- Porcentaje de niños y adolescentes de las muestras normal y clínica, según localización de la lesión, en los distintos centiles.
- Porcentaje de niños y adolescentes de las muestras normal y clínica, según la categoría del diagnóstico, en los distintos niveles de puntuaciones típicas.
- Porcentaje de niños y adolescentes de las muestras normal y clínica, según la categorías del diagnósticos, en los distintos centiles.

Porcentaje de <b>adultos</b> de las muestras normal y clínica, según <b>localización de la lesión</b> , en los <b>distintos centiles</b> .							
WCST PUNTUACIÓN		Centiles					
		Total 0-16	0-1	2-5	6-10	11-6	>16
<b>Nº de categorías completas</b>	Grupo normal	13.5	0.8	2.1	7.0	3.6	86.5
	Grupo clínico total	58.1	14.0	16.9	15.5	11.7	42.0
	Frontal	64.5	22.0	13.6	15.3	13.6	35.6
	Frontal más	60.4	5.7	30.2	9.4	15.1	39.6
	Difuso	59.4	14.7	14.7	18.1	11.9	40.7
	No frontal	44.5	11.1	14.8	13.0	5.6	55.6
<b>Intentos para completar la 1ª categoría.</b>	Grupo normal	11.7	0.5	3.4	4.4	3.4	88.3
	Grupo clínico total	40.9	11.1	16.9	7.9	5.0	59.2
	Frontal	34.0	15.3	10.2	6.8	1.7	66.1
	Frontal más	49.0	7.5	24.5	11.3	5.7	50.9
	Difuso	42.9	11.9	17.5	7.9	5.6	57.1
	No frontal	33.4	7.4	14.8	5.6	5.6	66.7
<b>Fallos para mantener la actitud.</b>	Grupo normal	10.7	0.8	1.0	4.7	4.2	89.3
	Grupo clínico total	21.0	2.6	4.4	6.4	7.6	79.0
	Frontal	22.1	3.4	5.1	5.1	8.5	78.0
	Frontal más	18.8	1.9	7.5	1.9	7.5	81.1
	Difuso	22.1	2.3	3.4	8.5	7.9	78.0
	No frontal	18.6	3.7	3.7	5.6	5.6	81.5
<b>Aprender a aprender</b>	Grupo normal	14.9	1.2	3.2	4.4	6.1	85.1
	Grupo clínico total	40.5	5.3	13.6	10.2	11.4	59.5
	Frontal	46.5	9.3	18.6	9.3	9.3	53.5
	Frontal más	46.6	4.7	16.3	7.0	18.6	53.5
	Difuso	39.8	6.0	12.0	12.8	9.0	60.2
	No frontal	31.1	0.0	11.1	6.7	13.3	68.9

Muestras utilizadas: Normal (N=356); Frontal (N=59); Frontal más (N=53); Difusa (N=177); No frontal (N=55). Los porcentajes correspondientes a Aprender a Aprender se han calculado sobre una muestra de 342 sujetos normales, 43 del grupo frontal, 43 del Frontal más, 133 del Difuso y 45 de No frontal, para los que ha sido posible calcular puntuación

Figura 31: Tabla ejemplo de valores de referencia.

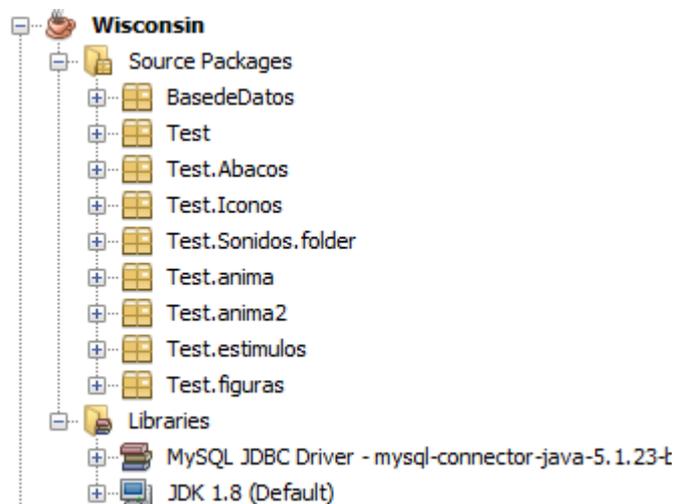
## REVISIÓN FINAL

---

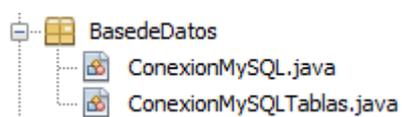
---

### 10.1 ESQUEMA GENERAL DE LA ESTRUCTURA FINAL DE LA APLICACIÓN :

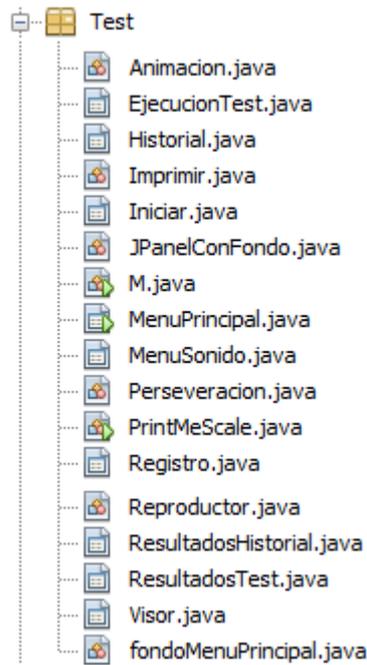
Esquema General de los paquetes y librerías utilizados:



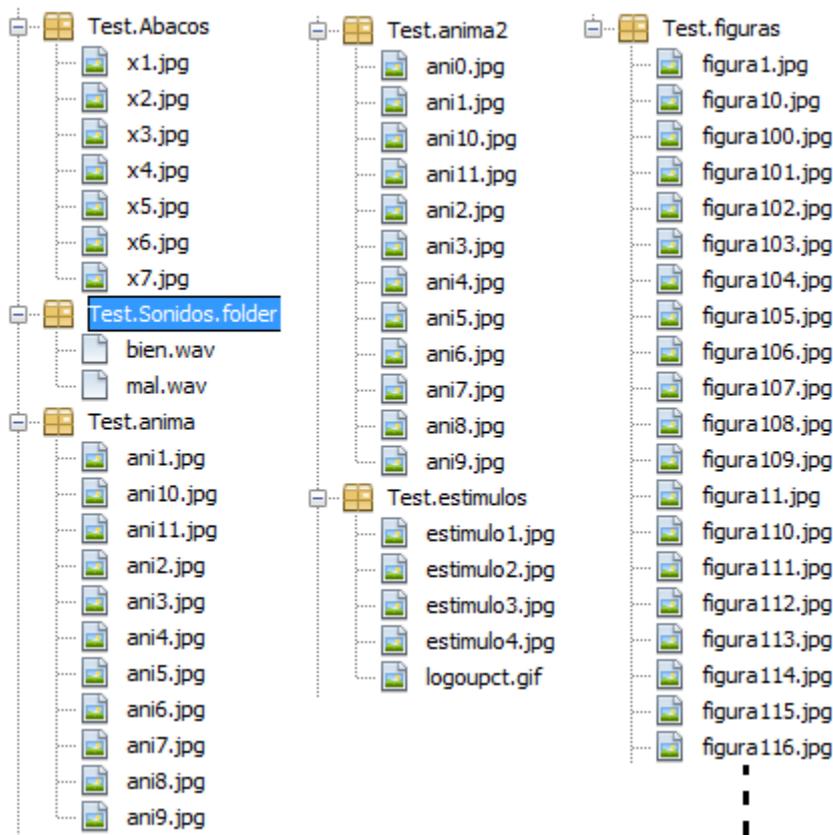
- Esquema de las clases que componen el paquete Base de datos. En el aparecen las clases que nos van a permitir la conexión a las mismas.



- Esquema de las clases que componen el paquete principal de la aplicación: Test.



- Esquema de los paquetes que incluyen las figuras y sonidos utilizados en el desarrollo de aplicación.



## 10.2 BIBLIOGRAFÍA:

- Introducción a la programación con Java. John Dean. Ed. Mc Graw Hill
- NetBeans IDE 7.0 CoolBook. Rhawl Dantas. PACKT Publishing.
- Mysql. Gilfillan, Ian. Anaya Multimedia.
- Pensa en Java . Bruce Ecket 2002
- Fundamentos de Informática Universidad complutense de Madrid. Edit:EME
- Tutoriales I y II . Introducción a la programación y programando en NetBeasn, página SEFCARM.es
- Wisconsin Card Sorting Test. Robert K. Heaton, Gordon J.Chelune. Adaptación española. Maria Victoria de la Cruz. Ed. TED Ediciones.
- Wisconsin Test Administration Manual. Smarter Balanced Ed.2014
- Web: [www.MySQL.com](http://www.MySQL.com)

## 10.3 FUTURAS MEJORAS

- Incluir Auto-instalador de bases de datos.
- Preparar la aplicación para el control mediante conexión LAN. (Red Local)
- Adaptación de sensores, para medir pulsaciones y sudoración, así como los protocolos de comunicación de estos con la aplicación. Para captar el estado del paciente durante la ejecución de la prueba.
- Compactación del código de programa.

## 10.4 RESUMEN

El objetivo fundamental del proyecto consistía en realizar una aplicación desarrollada en java, que mantenga una conexión a base de datos mediante MySQL.

Este ha sido superado en cuanto a las posibilidades que brinda el software respecto a los objetivos particulares de diseño. La aplicación permite guardar de forma separada: datos de sujetos y datos de pruebas realizadas por estos.

Se ha ampliado la información dada en la hoja de respuestas, antes debía seguirse un largo proceso para entender los resultados una persona que no había corregido el test, la aplicación al ampliar la información permite que con un vistazo a la hoja de respuestas sacar más conclusiones que de forma tradicional.

Para completar la aplicación, se ha incluido aunque no era requisito inicial del proyecto, un visor de tablas y métodos que permiten, la comparación de los resultados obtenidos en la prueba con muestras típicas y normalizadas así tener que acceder a la bibliografía para realizar dichas comparaciones.

Esta es la versión inicial, creada con perspectiva a futuras modificaciones y mejoras.

En el anexo se incluyen 4 ejemplos reales que han sido ejecutados y evaluados por la aplicación, arrojando los resultados correctos y demostrando el correcto funcionamiento de la aplicación.

el programa permite la impresión de los test en dos páginas separadas, la primera corresponde con la hoja de respuestas y la segunda con la de resultados. Siendo esté el último requisito esencial de la aplicación que pone punto final al desarrollo del proyecto.

## **10.5 CONTENIDO DEL CD DE PROYECTO.**

---

- Aplicación del WCST (contenida en la carpeta “Wisconsin”), archivo ejecutable: Wisconsin>>dist>>Wisconsin.Jar
- Aplicación Wisconsinlight (Igual que la anterior, pero sin las opciones de acceso a bases de datos, solo es para ejecutar el test y obtener inmediatamente los resultados, no es posible almacenarlos)
- Memoria del proyecto
- Video-tutorial
- Tablas y ábacos utilizados en el desarrollo de la aplicación
- Bases de datos a importar en el servidor de MySql
-

## TABLAS Y ÁBACOS.

<b>Porcentaje de adultos de las muestras normal y clínica, según localización de la lesión, en los distintos centiles.</b>							
WCST PUNTUACIÓN		Centiles					
		Total 0-16	0-1	2-5	6-10	11-6	>16
<b>Nº de categorías completas</b>	Grupo normal	13.5	0.8	2.1	7.0	3.6	86.5
	Grupo clínico total	58.1	14.0	16.9	15.5	11.7	42.0
	Frontal	64.5	22.0	13.6	15.3	13.6	35.6
	Frontal más	60.4	5.7	30.2	9.4	15.1	39.6
	Difuso	59.4	14.7	14.7	18.1	11.9	40.7
	No frontal	44.5	11.1	14.8	13.0	5.6	55.6
<b>Intentos para completar la 1ª categoría.</b>	Grupo normal	11.7	0.5	3.4	4.4	3.4	88.3
	Grupo clínico total	40.9	11.1	16.9	7.9	5.0	59.2
	Frontal	34.0	15.3	10.2	6.8	1.7	66.1
	Frontal más	49.0	7.5	24.5	11.3	5.7	50.9
	Difuso	42.9	11.9	17.5	7.9	5.6	57.1
	No frontal	33.4	7.4	14.8	5.6	5.6	66.7
<b>Fallos para mantener la actitud.</b>	Grupo normal	10.7	0.8	1.0	4.7	4.2	89.3
	Grupo clínico total	21.0	2.6	4.4	6.4	7.6	79.0
	Frontal	22.1	3.4	5.1	5.1	8.5	78.0
	Frontal más	18.8	1.9	7.5	1.9	7.5	81.1
	Difuso	22.1	2.3	3.4	8.5	7.9	78.0
	No frontal	18.6	3.7	3.7	5.6	5.6	81.5
<b>Aprender a aprender</b>	Grupo normal	14.9	1.2	3.2	4.4	6.1	85.1
	Grupo clínico total	40.5	5.3	13.6	10.2	11.4	59.5
	Frontal	46.5	9.3	18.6	9.3	9.3	53.5
	Frontal más	46.6	4.7	16.3	7.0	18.6	53.5
	Difuso	39.8	6.0	12.0	12.8	9.0	60.2
	No frontal	31.1	0.0	11.1	6.7	13.3	68.9

Muestras utilizadas: Normal (N=356); Frontal (N=59); Frontal más (N=53); Difusa (N=177); No frontal (N=55). Los porcentajes correspondientes a Aprender a Aprender se han calculado sobre una muestra de 342 sujetos normales, 43 del grupo frontal, 43 del Frontal más, 133 del Difuso y 45 de No frontal, para los que ha sido posible calcular puntuación

Porcentaje de <b>adultos</b> de las muestras normal y clínica, según la <b>localización de la lesión</b> , en los <b>distintos niveles de puntuaciones típicas.</b>									
	Nivel de realización								
	Muestra Clínica						Muestra Normal		
<i>DETERIORO</i>	<i>Grave</i>	<i>Severo</i>	<i>Moderado-severo</i>	<i>Moderado</i>	<i>Moderado-medio</i>	<i>medio</i>	<i>Debajo de la media</i>	<i>Media</i>	<i>Encima de la media</i>
<b>Puntuación típica</b>	0-84	0-54	55-61	62-69	70-76	77-84	85-91	92-106	107+
<b>Puntuación T</b>	0-39	0-19	20-24	25-29	30-34	35-39	40-44	45-54	55+
<b>% predicho en la distrib normal.</b>	14,6	0,1	0,4	1,5	4,0	8,6	14,4	38,2	32,8
<b>WCST puntuación</b>									
<b>Nº total errores</b>									
Grupo normal	14.2	0.3	0.8	1.6	4.2	7.3	9.4	47.9	28.6
Grupo clínico total	57.8	4.4	9.9	11.1	15.2	17.2	8.7	22.2	11.4
Frontal	59.4	10.2	11.9	8.5	16.9	11.9	11.9	20.3	8.5
Frontal más	62.2	1.9	11.3	11.3	15.1	22.6	9.4	15.1	13.2
Difuso	61.1	4.5	10.2	11.9	14.7	19.8	9.0	20.9	9.0
No frontal	40.8	0.0	5.6	11.1	14.8	9.3	3.7	35.2	20.4
<b>% de errores</b>									
Grupo normal	12.4	0.5	1.0	1.0	3.1	6.08	11.2	48.4	27.9
Grupo clínico total	54.2	2.9	10.5	9.3	15.5	16.0	12.0	22.4	11.4
Frontal	57.8	6.8	15.3	5.1	15.3	15.3	8.5	20.3	13.6
Frontal más	56.6	1.9	11.3	9.4	17.0	17.0	17.0	17.0	9.4
Difuso	56.0	2.8	10.2	11.3	15.3	16.4	14.7	20.3	9.0
No frontal	42.6	0.0	5.6	7.4	14.8	14.8	1.9	37.0	18.5
<b>Respuestas perseverativas</b>									
Grupo normal	12.5	0.3	0.8	0.8	3.6	7.0	10.4	50.8	26.3
Grupo clínico total	52.5	13.4	4.7	8.2	9.3	16.9	14.0	23.3	10.2
Frontal	57.7	22.0	6.8	3.4	10.2	15.3	11.9	20.3	10.2
Frontal más	58.6	11.3	1.9	15.1	13.2	17.0	17.0	13.2	11.3
Difuso	53.7	13.6	5.6	6.8	6.8	20.3	15.8	22.0	8.5
No frontal	37.2	5.6	1.9	11.1	13.0	5.6	7.4	40.7	14.8
<b>% Respuestas perseverativas</b>									
Grupo normal	12.1	0.3	0.8	0.8	2.9	7.3	10.9	49.2	27.9
Grupo clínico total	52.3	14.6	4.1	8.2	8.5	16.9	15.5	22.2	10.2
Frontal	57.6	23.7	5.1	3.4	8.5	16.9	13.6	16.9	11.9
Frontal más	62.3	11.3	1.9	17.0	11.3	20.8	13.2	15.1	9.4
Difuso	52.5	15.3	5.1	5.6	7.3	19.2	16.4	23.7	7.3
No frontal	35.4	5.6	1.9	13.0	9.3	5.6	16.7	29.6	18.5
<b>Errores perseverativos</b>									
Grupo normal	12.2	0.5	0.5	1.0	3.4	6.8	11.5	49.5	26.8
Grupo clínico total	52.9	12.8	5.2	8.7	9.6	16.6	14.9	22.2	9.9
Frontal	57.7	20.3	6.8	5.1	11.9	13.6	15.3	16.9	10.2
Frontal más	60.4	11.3	3.8	17.0	11.3	17.0	13.2	17.0	9.4
Difuso	53.7	13.0	6.2	6.8	8.5	19.2	18.1	20.3	7.9
No frontal	39.0	5.6	1.9	11.1	9.3	11.1	5.6	38.9	16.7
<b>% de errores perseverativos</b>									
Grupo normal	11.9	0.5	0.5	1.6	2.3	7.0	11.7	47.1	29.2
Grupo clínico total	52.8	14.0	5.0	8.2	9.0	16.6	14.3	22.2	10.8
Frontal	56.0	23.7	3.4	5.1	11.9	11.9	13.6	18.6	11.9
Frontal más	62.3	13.2	1.9	17.0	9.4	20.8	7.5	22.6	7.5
Difuso	53.6	13.6	6.2	6.2	9.0	18.6	16.4	20.9	9.0
No frontal	37.2	5.6	5.6	9.3	5.6	11.1	14.8	29.6	18.5
<b>Errores no perseverativos</b>									
Grupo normal	15.8	0.5	1.0	0.5	4.4	9.4	10.2	47.9	26.0
Grupo clínico total	38.1	2.6	2.0	6.4	9.3	17.8	14.3	30.3	17.2
Frontal	37.3	3.4	0.0	8.5	5.1	20.3	13.6	27.1	22.0
Frontal más	41.6	0.0	3.8	11.3	5.7	20.8	22.6	18.9	17.0
Difuso	41.6	3.4	1.7	8.2	11.3	18.6	11.9	29.9	16.9
No frontal	26.0	1.9	3.7	0.0	11.1	9.3	14.8	46.3	13.0

**Porcentaje de niños y adolescentes de las muestras normal y clínica, según localización de la lesión, en los distintos centiles.**

WCST PUNTUACIÓN		Centiles					
		Total 0-16	0-1	2-5	6-10	11-6	>16
<b>Nº de categorías completas</b>	Grupo normal	11.3	0.0	107	3.9	5.7	88.7
	Grupo clínico total	32.5	13.3	1.2	10.8	7.2	67.5
	Frontal	44.4	33.3	0.0	0.0	11.1	55.6
	Frontal más	40.0	0.0	20.0	20.0	0.0	60.0
	Difuso	40.6	18.8	0.0	15.6	6.2	59.4
	No frontal	21.6	5.4	0.0	8.1	8.1	78.4
<b>Intentos para completar la 1ª categoría.</b>	Grupo normal	15.5	0.9	3.7	5.7	5.2	84.5
	Grupo clínico total	24.0	12.0	1.2	6.0	4.8	75.9
	Frontal	55.5	44.4	11.1	0.0	0.0	44.4
	Frontal más	40.0	0.0	0.0	20.0	20.0	60.0
	Difuso	28.1	12.5	0.0	9.4	6.2	71.9
	No frontal	10.8	5.4	0.0	2.7	2.7	89.2
<b>Fallos para mantener la actitud.</b>	Grupo normal	8.0	0.2	2.8	1.3	3.7	91.9
	Grupo clínico total	19.2	1.2	4.8	7.2	8.0	80.7
	Frontal	0.0	0.0	0.0	0.0	0.0	100.0
	Frontal más	0.0	0.0	0.0	0.0	0.0	100.0
	Difuso	43.7	3.1	12.5	12.5	15.6	56.2
	No frontal	5.4	0.0	0.0	5.4	0.0	94.6
<b>Aprender a aprender</b>	Grupo normal	14.1	0.5	3.1	5.0	5.5	86.0
	Grupo clínico total	24.1	8.9	5.1	2.5	7.6	75.9
	Frontal	11.1	0.0	0.0	0.0	11.1	89.9
	Frontal más	40.0	0.0	40.0	0.0	0.0	60.0
	Difuso	39.2	10.7	7.1	7.1	14.3	60.7
	No frontal	13.5	10.8	0.0	0.0	2.7	86.5

Muestras utilizadas: Normal (N=459); Frontal (N=9); Frontal más (N=5); Difusa (N=33); No frontal (N=37). Los porcentajes correspondientes a Aprender a Aprender se han calculado sobre una muestra de 421 sujetos normales, 9 del grupo frontal, 5 del Frontal más, 28 del Difuso y 37 de No frontal, para los que ha sido posible calcular esta puntuación

Porcentaje de niños y adolescentes de las muestras normal y clínica, según la categorías del diagnóstico, en los distintos niveles de puntuaciones típicas.									
	Nivel de realización								
	Muestra Clínica						Muestra Normal		
<i>DETERIORO</i>	<i>Grave</i>	<i>Severo</i>	<i>Moderado-severo</i>	<i>Moderado</i>	<i>Moderado-medio</i>	<i>medio</i>	<i>Debajo de la media</i>	<i>Media</i>	<i>Encima de la media</i>
Puntuación típica	0-84	0-54	55-61	62-69	70-76	77-84	85-91	92-106	107+
Puntuación T	0-39	0-19	20-24	25-29	30-34	35-39	40-44	45-54	55+
% predicho en la distrib normal.	14,6	0,1	0,4	1,5	4,0	8,6	14,4	38,2	32,8
<b>WCST puntuación</b>									
<b>Nº total errores</b>									
Grupo normal	22.4	0.0	0.4	0.4	5.0	16.3	13.5	25.7	38.3
Grupo clínico total	32.8	0.0	1.3	3.8	12.8	14.7	15.4	25.6	26.3
Frontal	39.9	0.0	0.0	3.3	13.3	23.3	16.7	28.7	16.7
Frontal más	35.9	0.0	3.8	1.9	11.3	18.9	15.1	26.4	22.6
Difuso	30.7	0.0	0.0	8.2	14.3	8.2	12.2	24.5	32.7
No frontal	20.8	0.0	0.0	0.0	12.5	8.3	20.8	25.0	33.3
<b>% de errores</b>									
Grupo normal	20.5	0.2	0.2	1.1	5.7	13.3	13.9	30.9	34.6
Grupo clínico total	30.8	0.0	1.9	5.8	10.3	12.8	17.9	28.2	23.1
Frontal	33.3	0.0	0.0	3.3	10.0	20.0	30.0	23.3	13.3
Frontal más	34.0	0.0	3.8	3.8	9.4	17.0	15.1	34.0	17.0
Difuso	30.6	0.0	2.0	10.2	10.2	8.2	14.3	24.5	30.6
No frontal	20.9	0.0	0.0	4.2	12.5	4.2	16.7	29.2	33.3
<b>Respuestas perseverativas</b>									
Grupo normal	17.2	0.4	0.4	2.0	4.4	10.0	12.9	33.6	36.4
Grupo clínico total	28.8	3.2	3.2	5.1	5.1	12.2	15.4	30.8	25.0
Frontal	40.1	0.0	6.7	6.7	6.7	20.0	10.0	30.0	20.0
Frontal más	35.8	3.8	0.0	9.4	7.5	15.1	13.2	22.6	28.3
Difuso	22.5	6.1	4.1	0.0	4.1	8.2	18.4	30.6	28.6
No frontal	12.6	0.0	4.2	4.2	0.0	4.2	20.8	50.0	16.7
<b>% Respuestas perseverativas</b>									
Grupo normal	16.4	0.4	0.9	1.5	4.4	9.2	13.1	37.7	32.9
Grupo clínico total	29.5	3.2	3.2	5.1	5.8	12.2	12.2	34.6	23.7
Frontal	40.1	0.0	6.7	6.7	20.0	10.0	30.0	20.0	0.0
Frontal más	35.8	3.8	0.0	9.4	9.4	13.2	11.3	30.2	22.6
Difuso	24.5	6.1	4.1	0.0	4.1	10.2	12.2	34.7	28.6
No frontal	12.6	0.0	4.2	4.2	0.0	4.2	16.7	50.0	20.8
<b>Errores perseverativos</b>									
Grupo normal	19.0	0.2	0.7	2.0	3.5	12.6	10.9	31.6	38.6
Grupo clínico total	30.1	3.2	3.2	5.1	5.1	13.5	14.7	30.1	25.0
Frontal	40.0	0.0	3.3	10.0	6.7	20.0	16.7	26.7	16.7
Frontal más	35.8	3.8	1.9	7.5	9.4	13.2	11.3	24.5	28.3
Difuso	24.4	6.1	4.1	0.0	2.0	12.2	12.2	32.7	30.6
No frontal	16.7	0.0	4.2	4.2	0.0	8.3	25.0	41.7	16.7
<b>% de errores perseverativos</b>									
Grupo normal	16.6	0.2	0.9	2.2	3.1	10.2	12.0	36.6	34.6
Grupo clínico total	28.8	3.2	3.8	4.5	5.1	12.2	14.1	35.3	21.8
Frontal	36.8	0.0	6.7	6.7	6.7	16.7	13.3	33.3	16.7
Frontal más	33.9	3.8	1.9	7.5	9.4	11.3	13.2	28.3	24.5
Difuso	26.5	6.1	4.1	0.0	2.0	14.3	10.2	40.8	22.4
No frontal	12.6	0.0	4.2	4.2	0.0	4.2	25.0	41.7	20.8
<b>Errores no perseverativos</b>									
Grupo normal	15.0	0.7	0.4	1.7	4.1	8.1	15.7	34.9	34.4
Grupo clínico total	21.2	0.0	1.9	5.1	4.5	9.6	11.5	38.5	28.8
Frontal	26.7	0.0	0.0	6.7	3.3	16.7	23.3	36.7	13.3
Frontal más	20.8	0.0	1.9	5.7	3.8	9.4	9.4	45.3	24.5

<b>Porcentaje de niños y adolescentes de las muestras normal y clínica, según la categoría del diagnóstico, en los distintos centiles.</b>							
WCST Puntuación		Centiles					
		Total 0-16	0-1	2-5	6-10	11-6	>16
<b>Nº de categorías completas</b>	Grupo normal	11.3	0.0	1.7	3.9	5.7	88.7
	Grupo clínico total	30.2	9.0	1.3	10.9	9.0	69.9
	Frontal	43.4	16.7	0.0	20.0	6.7	56.7
	Frontal más	24.3	11.3	1.9	5.7	5.7	75.5
	Difuso	28.5	6.1	2.0	10.2	10.2	71.4
	No frontal	29.2	0.0	0.0	12.5	16.7	70.8
<b>Intentos para completar la 1ª categoría.</b>	Grupo normal	15.5	0.9	3.7	5.7	5.2	84.5
	Grupo clínico total	22.4	7.7	4.5	5.1	5.1	77.6
	Frontal	30.0	10.0	0.0	13.3	6.7	70.0
	Frontal más	23.3	11.3	5.7	3.8	7.5	71.7
	Difuso	14.3	6.1	4.1	0.0	4.1	85.7
	No frontal	16.6	0.0	8.3	8.3	0.0	83.3
<b>Fallos para mantener la actitud.</b>	Grupo normal	8.0	0.2	2.8	1.3	3.7	91.9
	Grupo clínico total	14.1	1.3	3.2	3.8	5.8	85.9
	Frontal	46.6	3.3	13.3	10.0	20.0	53.3
	Frontal más	5.7	0.0	0.0	3.8	1.9	94.3
	Difuso	6.1	0.0	2.0	0.0	4.1	93.9
	No frontal	8.4	4.2	0.0	4.2	0.0	91.7
<b>Aprender a aprender</b>	Grupo normal	14.1	0.5	3.1	5.0	5.5	86.0
	Grupo clínico total	27.7	7.1	5.7	2.8	12.1	72.3
	Frontal	42.3	11.5	7.7	7.7	15.4	57.7
	Frontal más	22.9	6.2	2.1	2.1	12.5	77.1
	Difuso	26.7	6.7	8.9	2.2	8.9	73.3
	No frontal	22.6	4.5	4.5	0.0	13.6	77.3

Muestras utilizadas: Normal (N=459); Frontal (N=59); Frontal más (N=53); Difusa (N=177); No frontal (N=55).  
 Los porcentajes correspondientes a Aprender a Aprender se han calculado sobre una muestra de 421 sujetos normales, 26 del grupo Traumatismo cerebral, 48 del Epilepsia , 45 de Déficit de atención y 24 del incapaz de aprender a leer, , para los que ha sido posible calcular esta puntuación

**Sistema de Heaton de clasificación para interpretar las puntuaciones normativas basándose en su experiencia clínica**

Las siguientes agrupaciones de puntuaciones típicas (media=100 y desviación típica =15) y puntuaciones T (media=50 y desviación típica= 10), definen estos niveles clínicamente relevantes.

Puntuaciones Típicas	Puntuaciones T	Nivel de Deterioro
$\mu \leq 54$	$\leq 19$	Grave
$55 \leq \mu \leq 61$	$20 \leq t \leq 24$	Grave a Moderado
$62 \leq \mu \leq 69$	$25 \leq t \leq 29$	Moderado
$70 \leq \mu \leq 76$	$30 \leq t \leq 34$	Moderado a Intermedio
$77 \leq \mu \leq 84$	$35 \leq t \leq 39$	Intermedio
$85 \leq \mu \leq 91$	$40 \leq t \leq 44$	Superior a la media
$92 \leq \mu \leq 106$	$45 \leq t \leq 54$	Medio
$\mu \geq 107$	$t > 55$	Inferior a la media

**ANEXO II****MANUAL DE USUARIO**

---

---

Este manual ha sido redactado con la finalidad de que sirva de referencia para el uso adecuado de la aplicación, en él se muestran de forma sencilla y detalla todas las posibles opciones que nos permite el software. Antes de entrar en estas posibilidades así como detallar el funcionamiento y herramientas de las que consta, se hace una breve descripción del test en el que está basado.

Los siguientes puntos se deben tratar y comprender para una correcta adaptación a la simulación informática. Dicha adaptación es un software que permite tanto el desarrollo del test de forma interactiva como las interfaces y formularios necesarios para el almacenamiento y posterior tratamiento de los resultados. Algunas partes de este manual están obtenidos directamente de la documentación o Memoria general del proyecto.

**1- ASPECTOS GENERALES DEL TEST**

A continuación se tratan una serie de puntos relacionados directamente con la aplicación del test original, así como algunos aspectos a tener en cuenta por el profesional a la hora de su aplicación, como la funcionalidad, los elementos que contiene, la población a la que se destina, los materiales que se usan, los fines perseguidos, así como los requisitos profesionales necesarios para su correcta aplicación.

**12.1.1 Antecedentes y campo de aplicación**

Test desarrollado originalmente para evaluar el razonamiento abstracto y la habilidad para cambiar las estrategias cognitivas como respuesta a eventuales modificaciones ambientales. Puede ser considerado como una medida de la función ejecutiva que requiere habilidad para desarrollar y mantener las estrategias de solución de problemas que resultan adecuadas para conseguir unos objetivos, a través de condiciones que implican cambios de estímulos

El WCST requiere estrategias de planificación, indagaciones organizadas, utilizando el “feedback ambiental” para cambiar esquemas, orientación de la conducta hacia el logro de un objetivo y modulación de las respuestas impulsivas. Este a

diferencia de otras medidas de razonamiento abstracto, proporciona puntuaciones objetivas del total de aciertos y además las correspondientes a otras fuentes específicas de dificultad en la tarea. (por ejemplo, conceptualización inicial deficiente, fallos para mantener la actitud o el curso cognitivo, perseveración y deficiencia de aprendizaje a lo largo de las etapas del test).

Durante su desarrollo se utilizó como medida del razonamiento abstracto en poblaciones de adultos normales, El WCST se ha empleado cada vez en mayor escala como instrumento clínico neurológico. El test tiene sus raíces en anteriores investigaciones sobre la abstracción y los efectos de las lesiones en el lóbulo frontal, en el mantenimiento de la actitud cognitiva, realizadas con personas y animales.

Gran parte de la popularidad alcanzada entre los clínicos radica en su sensibilidad para proporcionar información sobre disfunciones cerebrales que afecten a los lóbulos frontales. El interés respecto a los efectos que produce el daño temprano del lóbulo frontal, en aspectos cognitivos y de desarrollo, en los niños, hizo surgir también un considerable interés hacia la utilización de WCST como medida potencial de la función ejecutiva en la edad escolar.

A causa de su aparente sensibilidad a los efectos de las lesiones en el lóbulo frontal, El WCST se menciona frecuentemente como una medida del funcionamiento frontal o pre-frontal. Pero esto representa una excesiva simplificación. Si bien se considera que algunas de las dimensiones cognitivas que aprecia el test son particularmente vulnerables las condiciones neurológicas que afectan a las zonas frontales del cerebro, cualquier irregularidad médica o psicológica que desorganice las funciones ejecutivas, en todo o en parte, puede dar lugar como resultado un deterioro en la realización del test. Por esta razón, la interpretación clínica de la realización de un sujeto en el test debe ser considerada en el contexto de una evaluación neuropsicológica integrada por datos neuropsicológicos, datos médicos, información psicosocial e historia del sujeto.

### **12.1.2 Descripción del test.**

Básicamente, el test consiste en 4 tarjetas modelo (Figura 1) y 128 tarjetas de prueba (divididas en dos grupos de 64 tarjetas), que contienen dibujos que varían en la forma (cruces, círculos, triángulos o estrellas), color (rojo, azul, amarillo o verde), y número de figuras (uno, dos, tres o cuatro). El examinador coloca las 4 tarjetas modelo delante del paciente y le entrega un grupo de 64 tarjetas. A continuación, se le pide que clasifique cada una de las tarjetas según el criterio que quiera el propio paciente. Se le indicará si la respuesta es correcta o no. El orden de clasificación de las categorías es invariable, color, forma y número. De forma tradicional el examinador debe marcar en la hoja de respuestas las categorías (color, forma y número) correctas de clasificación. Así por ejemplo, teniendo en cuenta que el criterio inicial es el color, si el paciente coloca su primera tarjeta (. una estrella de color azul) sobre la tarjeta modelo “una estrella de color amarillo”, el examinador tachará la categoría “forma” y “número”, sin embargo la respuesta es “incorrecto” para el paciente. En el siguiente ensayo, el paciente puede optar por seguir clasificando las tarjetas en relación con algunas de las categorías anteriores o cambiar de criterio y seleccionar el “color”. En este caso la

respuesta sería “correcto”. Cuando el sujeto consigue 10 respuestas correctas se cambia de criterio sin previo aviso.

Una ejecución adecuada implica que el sujeto determine en primer lugar el principio de clasificación correcto en función de la retroalimentación que se le es proporcionada, y mantener después este principio de clasificación o “set” (color), a través de los estímulos que cambian e ignorar las demás dimensiones irrelevantes (forma y número). Por el contrario, se considera un fracaso o una incapacidad para mantener el criterio cuando el paciente comete 5 o más respuestas correctas consecutivas y después comete 1 error antes de conseguir con éxito el criterio (10 respuestas correctas consecutivas). Ahora bien, cuando el paciente clasifica 10 respuestas correctas dentro de una categoría, y el examinador cambia el criterio, el sujeto debe inhibir la tendencia a persistir o perseverar el criterio anterior. Se registra el número de respuestas correctas, el número de categorías completadas, el número de errores, y el de perseveraciones en el error.

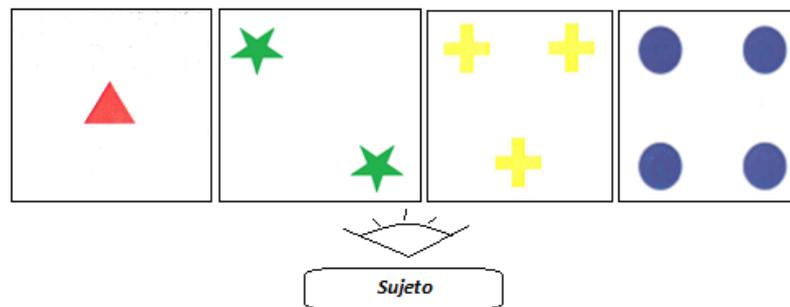


Figura 1. *Tarjetas Modelo.*

### **12.1.3 Datos del test**

- 1- Administración: Individual
- 2- Duración: Variable
- 3- Aplicación: Niños, adolescentes y adultos con edades comprendidas entre 6.5 años y 89 años.
- 4- Significación: Medida de la función ejecutiva que requiere estrategias de planificación, indagaciones organizadas y utilización del feedback ambiental para cambiar esquemas. Dada su posible sensibilidad a los efectos de las lesiones en el lóbulo frontal, se menciona frecuentemente como una medida del funcionamiento frontal y prefrontal.
- 5- Baremación: Puntuaciones directas y típicas por edades para cada una de las variables de la prueba.

### **12.1.4 Modo tradicional de aplicar el test**

- 1- La aplicación consiste en colocar las cuatro tarjetas-estímulo ante el sujeto ordenándolas de izquierda a derecha de la forma: 1º Un triángulo rojo, 2º Dos estrellas verdes, 3º Tres cruces amarillas, 4º cuatro círculos azules.
- 2- Se entrega al sujeto un bloque tarjetas y se le indica que tiene que emparejar cada una de las tarjetas de ese bloque con una de las tarjetas-estímulo que se han colocado ante él y que puede emparejarlas de cualquier modo que le parezca posible. Cada vez que el sujeto coloque una tarjeta se le indicará si lo ha hecho de forma correcta o errónea, pero no se le dice nunca cuál es la categoría que se tiene en cuenta para clasificar.
- 3- Cuando el sujeto ha logrado un número determinado de respuestas correctas consecutivas emparejando las tarjetas con arreglo a la categoría de clasificación establecida inicialmente (Color), se cambia dicha categoría de clasificación pasando a Forma o Número, sin avisar, esto requiere que el sujeto utilice el “feedback” que recibe del examinador para establecer una nueva estrategia de clasificación.
- 4- La aplicación avanza de esta forma cambiando las categorías establecidas como criterios de clasificación, seleccionando un nuevo criterio de clasificación entre los posibles.

### **12.1.5 Poblaciones a las que se destina**

Niños, adolescentes y adultos, con edades comprendidas entre 6.5 años y 89 años. Se debe poner gran cautela al interpretar los resultados correspondientes a sujetos de las edades comprendidas entre 85 y 89 años, ya que el número de sujetos de estas edades incluidos en la muestra es escaso.

Los sujetos a los que se le aplique la prueba deben poseer una visión normal, o corregida con gafas, y la capacidad auditiva suficiente para comprender las instrucciones; es preciso también que sean capaces de discriminar visualmente los estímulos de color, forma y número. Entre la población clínica es posible que en algunos casos no se cumplan estas condiciones; esto sucede, por ejemplo, cuando se trata de sujetos de quienes se tenga conocimiento o sospecha de disfunción cerebral. En tales casos, la fiabilidad y validez de las puntuaciones del WCST dependen en gran manera de la habilidad clínica y la experiencia del examinador.

### **12.1.6 Requisitos profesionales**

Aunque cualquier persona con experiencia en la aplicación de pruebas puede llevar a cabo la aplicación del WCST, los procedimientos de administración, registro de respuestas y puntuación que se detallan en apartados siguientes deben ser cuidadosamente estudiados y ensayados antes de utilizar la prueba en una situación

clínica. El entrenamiento y supervisión en estos procedimientos debe estar a cargo de un psicólogo cualificado para ello.

La interpretación clínica del WCST debe ser realizada por un psicólogo especializado en psicología clínica o neuropsicología. La utilidad y la validez de este test como medida clínica de la habilidad cognitiva están relacionadas directamente con la formación y los conocimientos del profesional que lo utilice, así como con el conocimiento de la información que contiene este manual. La interpretación de las puntuaciones del WCST con fines clínicos o de diagnóstico no deben abordarse si no se posee una clara comprensión de la relación conducta-cerebro y de los factores médicos y psicológicos que la afectan.

Dicho esto cabe destacar que la aplicación no trata de sustituir al profesional en el desarrollo de un diagnóstico acertado, sino ayudar a y simplificar la labor del mismo, evitando fallos de interacción y cálculo de resultados, además de permitir llevar una evolución de las pruebas gracias al sistema de almacenamiento del software. Por otra parte se ha desarrollado con el objetivo de que al sujeto que lo realiza le parezca atractivo y no se sienta cohibido por la presencia del profesional. Para un niño que realiza el test le puede parecer un juego, de forma que ponga interés por sacar buena puntuación, aunque en realidad eso no es lo que se persigue.

### **12.1.7 Aplicación del test**

El test se utiliza en variedad de situaciones, tanto con niños como con adultos, dado que se trata de una prueba de aplicación individual, el WCST se puede utilizar como instrumento clínico o de investigación combinándolo con pruebas que evalúen otros aspectos del funcionamiento cognitivo y neuropsicológico.

Las normas de aplicación y puntuación se presentan en el proyecto, ya que no son objetivo de este manual, estos serán sistematizados mediante algoritmos en la aplicación informática ahorrando tiempo al profesional que lo desarrolla, así como evitando errores de valoración.

Es necesario que la aplicación se realice en un lugar tranquilo, con iluminación adecuada para que el sujeto pueda percibir correctamente los estímulos del test. En la aplicación tradicional se necesitarían una mesa y dos sillas, una para el sujeto y otra para el examinador, en la aplicación informática es solo el paciente el que se sitúa delante del PC.

En el proceso tradicional de aplicación el examinador deberá anotar las respuestas en la hoja de anotación y posteriormente tratarlas para llegar al resultado o diagnóstico en este caso. La aplicación informatizada recibe mediante estímulos captados por los periféricos (ratón, teclado del ordenador) la información a anotar. Con los datos recibidos realiza de forma instantánea y sin ningún tipo de error los resultados requeridos por el profesional para realizar y evaluar un diagnóstico.

En la ejecución de la aplicación desaparecen las comprobaciones iniciales por parte del examinador, como es la correcta colocación de las cartas estímulo, así como su orientación y la correcta secuencia de las figuras del mazo.

## 2- ENTORNO

Este manual ha sido redactado con la finalidad de que sirva de referencia para el uso adecuado de la aplicación, en él se muestran de forma sencilla y detalla todas las posibles opciones que nos permite el software.

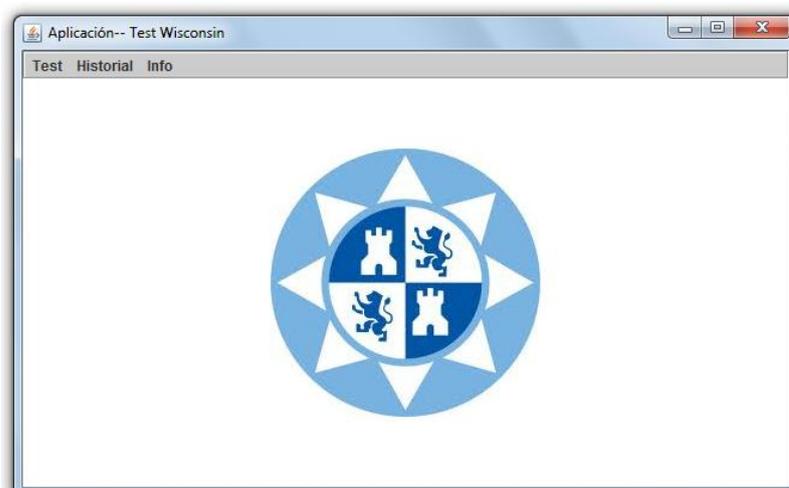
Las imágenes que aparecen en el desarrollo del mismo son capturas directas sobre los formularios, menús y ventanas de las que consta la aplicación, acompañadas de breves explicaciones sobre el funcionamiento de las diferentes opciones de registro, realización de test y revisión de historiales. Estos han sido elaborados a partir de los resultados obtenidos en los test ejecutados, por diferentes sujetos en evaluación clínica.

La aplicación consta de una interface sencilla que por medio de sus menús, nos permite ir paso a paso de una forma estructurada en un proceso de ayuda al diagnóstico. Es un software que permite tanto el desarrollo del test de forma interactiva como los formularios necesarios para el almacenamiento y posterior tratamiento específico de los resultados.

Para abrir la aplicación se debe acceder y ejecutar al archivo que lleva el mismo nombre que el propio test “Wisconsin.jar”, este es un ejecutable programado en java, esto quiere decir que se hace necesario disponer de una versión actualizada de Java en el ordenador.

Se puede descargar la versión más actualizada de la pagina web [www.java.com](http://www.java.com) accediendo a la zona de descargas y seleccionando el sistema operativo instalado en el ordenador.

Una vez ejecutado el archivo “Wisconsin.jar” aparece la pantalla Principal de la aplicación (Figura 2), sobre este se van a mostrar los formularios y la propia interface durante la ejecución del test. En la parte superior hay una barra de menús o Menú principal que contiene las opciones: Test, Historial e Info.



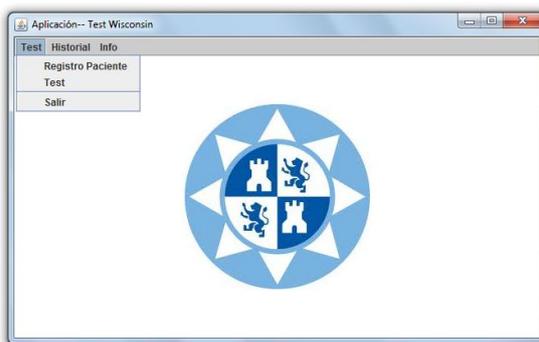
**Figura 2: Menú Principal.**

Los menús están diseñados para seguir un orden en la aplicación del test. Además de poder acceder de forma rápida a los historiales de pruebas realizadas con anterioridad y a documentación de apoyo al diagnóstico.

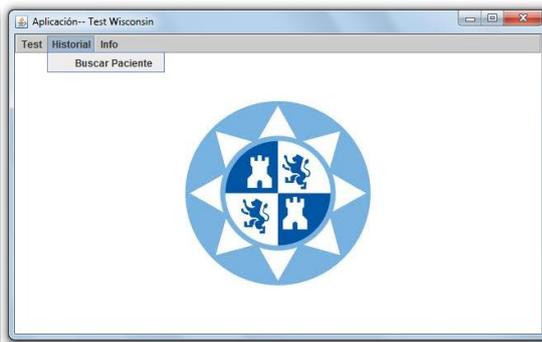
La aplicación consta básicamente de tres partes que corresponden a las opciones del menú principal y son las siguientes:

1-. Test: Esta opción de la barra de menús (figura 3.a) contiene a su vez dos opciones, Registro de Pacientes y Test. Es la zona donde se procederá al registro de los datos personales y médicos así como y la ejecución de test.

2-. Historial: En este caso solo aparece la opción Buscar paciente (Figura 3.b), y como su nombre indica, su selección lleva la pantalla de Revisión de Historiales.

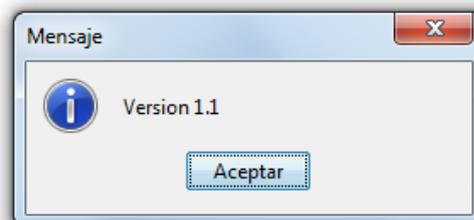


**Figura 3a: Sub-menú Test.**



**Figura 3b: Sub-menú Historial.**

3-. Info: Contiene dos opciones, la primera corresponde a un visor de imágenes que contiene información adicional para el diagnóstico denominada Muestra de tipificación (Valores de referencia) según figura 4a, en cuanto a la segunda muestra un mensaje con la versión actual de la aplicación, según figura 4b.



**Figura 4a:** Sub-menú Info**Figura 4b:** Mensaje de la versión actual.

## A. REGISTRO DE PACIENTES

Inicialmente se ejecuta la aplicación y se selecciona dentro de la barra de menús la opción: Test>>Registro Paciente, esto nos lleva al formulario de registro está dividido en dos partes diferenciadas (Figura 5), una superior denominada Captura Datos Sujeto y otra inferior denominada Buscar Paciente. Por lo tanto el programa nos permite tanto captar un nuevo paciente, comprobar si ya ha sido ingresado en la base de datos y realizar modificaciones en los datos guardados.

Los datos registrados en el formulario son tanto Personales como clínicos. Los personales son: dni, nombre, apellidos, fecha de nacimiento, sexo, lateralidad, nivel de estudios y ocupación. Los clínicos corresponden a: Problema de referencia, antecedentes, medicación y observaciones sobre la conducta.

id_paciente	DNI	Nombre	P.Apellido	S.Apellido	Fecha de nac.	Sexo
4	12341234T	Victor	Almiano	Contreras	06-06-2005	H
7	23212323R	Sergio	Almoradi	Fernandez	08-03-1960	H
8	6789789C	Puencanta	Carroll	Alcantara	12-03-1991	M
9	9999999Z	Julia	Zamora	Villar	13-12-1970	M

**Figura 5.** Formulario Registro de Pacientes.

### i. Ejemplo de Registro:

Al iniciar el formulario, todos los campos aparecen desactivados, incluso los botones de Guardar y Cancelar, pulsando el botón Nuevo quedan habilitados para poder ser ingresados. El examinador entrevista al sujeto e introduce en cada campo los datos correspondientes.

Tras ser ingresados, pulsando el botón Guardar quedarán almacenados en la base de datos, simultáneamente se sumarán a la tabla inferior de búsqueda de paciente. En la siguiente figura se muestra una captura de pantalla justo después de dar al botón Guardar, como se puede observar los datos del formulario de Captura han aparecido en la tabla inferior.

**Captura Datos Sujeto**

Dni: 23232323F Nombre: Francisco Primer Apellido: Pardo Segundo Apellido: Alsan

Fecha de nacimiento: 09-06-1982 Edad Cronologica: 32 Sexo: H Lateralidad: D

Nivel de Estudios: Fp Superior.

Ocupación: Auxiliar Enfermería.

Problema de referencia: Falta de atención y concentración.

Antecedentes, enfermedades actuales: Ninguna enfermedad actual conocida.

Medicación, dosis: N/P.

Observaciones sobre la conducta: Ninguna alteración en la conducta durante la entrevista.

Nuevo Cancelar Guardar Salir

**Buscar Paciente**

Buscar: [ ] Buscar

id_paciente	DNI	Nombre	P.Apellido	S.Apellido	Fecha de nac.	Sexo
4	12341234T	Victor	Almiñano	Contreras	06-06-2005	H
7	23232323M	Sergio	Almoradi	Fernandez	08-03-1960	H
8	6789788C	Fuensanta	Carmoli	Alcantara	23-01-2001	M
9	99999999Z	Julia	Zamora	Villar	13-12-1970	M
12	23232323F	Francisco	Pardo	Alsan	09-06-1982	H

Figura 6. Ejemplo de registro de paciente

Si todo ha sido correcto deberá aparecer el mensaje según figura 7.



Figura 7. Mensaje ingreso de datos

**ii. Ejemplo de Modificación de registro:**

Como se ha descrito con anterioridad el formulario actual dispone en la parte inferior de una zona de búsqueda, que nos permite ingresar como criterio de búsqueda, el dni, el nombre o los apellidos del sujeto, tras hacer esto y pulsar el botón "buscar" aparecerán las coincidencias respecto del criterio de búsqueda en la tabla (Resultados de la búsqueda).

A continuación para mostrar el funcionamiento de una forma más detallada se presenta gráfico, de un caso hipotético en el que se van a modificar los datos personales.

El nombre del paciente a modificar es Francisco Pardo, tras ingresar este en el campo buscar y pulsar el botón, aparece en la tabla la única coincidencia, es decir solo hay un Francisco Pardo en la Base de datos. Una vez localizado el sujeto se procede a realizar las modificaciones pertinentes, para ello debe ser seleccionada la fila de la tabla correspondiente a dicho sujeto, en este caso solo hay una.

A continuación con el botón derecho sobre la fila aparece la opción “Seleccionar-Modificar”, pulsando sobre esta, los datos relativos a dicho sujeto, son cargados en la parte superior del formulario. Figura 8.

The screenshot shows a web application window titled "Registro paciente". It is divided into two main sections: "Captura Datos Sujeto" and "Buscar Paciente".

**Captura Datos Sujeto:**

- Dni: 23232323F
- Nombre: Francisco
- Primer Apellido: Pardo
- Segundo Apellido: Alisan
- Fecha de nacimiento: 09-06-1982
- Edad Cronologica: 32
- Sexo: H
- Lateralidad: D
- Nivel de Estudios: Fp Superior.
- Ocupación: Auxiliar Enfermería.
- Problema de referencia: Falta de atención y concentración.
- Antecedentes, enfermedades actuales: Ninguna enfermedad actual conocida.
- Medicación, dosis: N/P.
- Observaciones sobre la conducta: guna alteración en la conducta durante la entrevista.

**Buscar Paciente:**

Buscar: Francisco Pardo

id_paciente	DNI	Nombre	P.Apellido	S.Apellido	Fecha de nac.	Sexo
12		Francisco	Pardo	Alisan	09-06-1982	H

A red box highlights the "Seleccionar-Modificar" button in the first row of the search results table.

**Figura 8:** Formulario de Registro. Ejemplo modificación de Registro 1.

Para proceder a modificarlos, simplemente situados sobre ellos se borra el contenido y se completa con el deseado. En el ejemplo actual se han modificado hipotéticamente los apellidos Pardo Alisan por Alegría Bosman, a continuación pulsando el botón Guardar, los datos quedarán modificados. Si todo ha sido correcto aparecerá el mensaje, ” Los datos han sido modificados correctamente. El resultado de las modificaciones del ejemplo se puede observar en la figura 9.

The screenshot shows the same "Registro paciente" application window. The "Captura Datos Sujeto" form is now populated with modified data:

- Dni: 23232323F
- Nombre: Francisco
- Primer Apellido: Alegría
- Segundo Apellido: Bosman
- Fecha de nacimiento: 09-06-1982
- Edad Cronologica: 32
- Sexo: H
- Lateralidad: D
- Nivel de Estudios: Fp Superior.
- Ocupación: Auxiliar Enfermería.
- Problema de referencia: Falta de atención y concentración.
- Antecedentes, enfermedades actuales: Ninguna enfermedad actual conocida.
- Medicación, dosis: N/P.
- Observaciones sobre la conducta: guna alteración en la conducta durante la entrevista.

The "Buscar Paciente" section shows the search term "Francisco".

id_paciente	DNI	Nombre	P.Apellido	S.Apellido	Fecha de nac.	Sexo
12	23232323F	Francisco	Alegría	Bosman	9-06-1982	H

A message dialog box is overlaid on the form, containing the text: "Mensaje: Los datos se han Modificado correctamente" and an "Aceptar" button.

**Figura 9:** Formulario de Registro. Ejemplo modificación de Registro 2.

## 2.2 INICIO Y EJECUCIÓN DEL TEST.

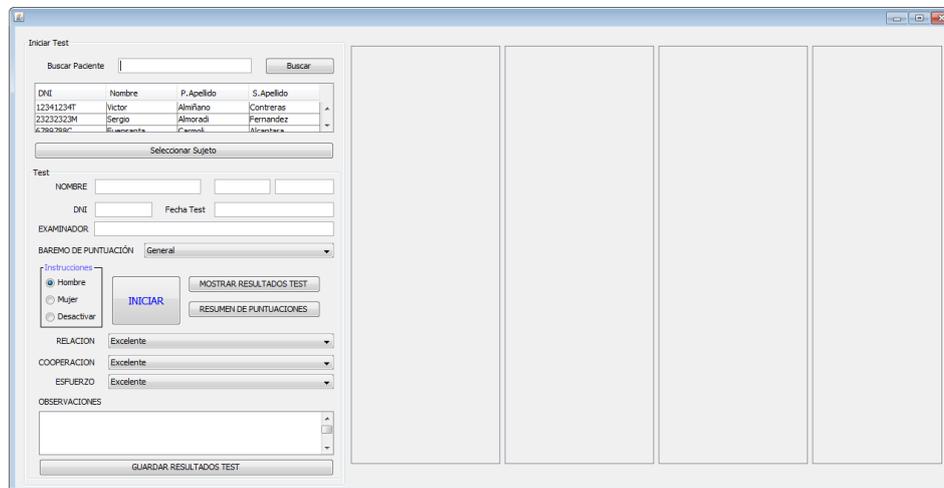
Una vez registrado el paciente en la base de datos, se procede a la ejecución del test. Para ello, el examinador debe dirigirse al Formulario de Inicio: Test→Iniciar.

Dicho formulario contiene, campos a rellenar antes de la ejecución del test y campos a rellenar tras la ejecución del mismo. Por tanto se pueden considerar 3 etapas:

- 4- *Etapas Pre-Test:* Búsqueda del sujeto a evaluar e ingreso de datos iniciales.
- 5- *Ejecución del test* por el sujeto.
- 6- *Etapas Post-Test:* El examinador completa datos relativos al comportamiento del sujeto durante la ejecución de la prueba y finaliza el proceso guardando los resultados.

La pantalla de inicio está dividida en dos zonas, la izquierda o formulario de inicio del test, que es donde se localiza al paciente que va a realizar la prueba y se introducen los parámetros necesarios para obtener unos resultados adecuados, y la parte derecha o zona de resultados formada por cuatro columnas, estas serán auto-rellenadas con las respuestas del sujeto al finalizar el test.

El contenido de las columnas, es el mismo que va a aparecer en el historial del paciente y será descrito en el correspondiente apartado. A continuación se puede observar la pantalla de inicio del test en la figura 10.



**Figura 10.** Formulario de inicio del test y zona de resultados

Como se puede ver en la siguiente imagen (figura 11a) para iniciar un test y asignarlo a un paciente, es necesario buscar dicho paciente en la zona superior del formulario, una vez localizado, debe ser marcado sobre la tabla y a continuación pulsando el botón seleccionar paciente, los datos son trasladados a los campos inferiores.

Iniciar Test

Buscar Paciente:

DNI	Nombre	P.Apellido	S.Apellido
23232323F	Francisco	Alegria	Bosman
56756756Y	Francisco	Perez	Hernan

Test

NOMBRE:

DNI:  Fecha Test:

EXAMINADOR:

BAREMO DE PUNTUACIÓN:

Instrucciones:

Hombre  Mujer  Desactivar

RELACION:

COOPERACION:

ESFUERZO:

OBSERVACIONES:

**Figura 11a.** Formulario de inicio: Selección

Iniciar Test

Buscar Paciente:

DNI	Nombre	P.Apellido	S.Apellido
23232323F	Francisco	Alegria	Bosman
56756756Y	Francisco	Perez	Hernan

Test

NOMBRE:

DNI:  Fecha Test:

EXAMINADOR:

BAREMO DE PUNTUACIÓN:

Instrucciones:

Hombre  Mujer  Desactivar

RELACION:

COOPERACION:

ESFUERZO:

OBSERVACIONES:

**Figura 11b.** Formulario de inicio: Baremo

Una vez seleccionado el paciente, es posible ingresar el nombre del examinador que será almacenado junto a los demás datos y los resultados del test.

Es importante seleccionar el BAREMO DE PUNTUACION correspondiente, ya que según sea seleccionado uno u otro, los resultados pueden variar, este baremo es un dato necesario para obtener los resultados estadísticos de la prueba, ya que está creado a partir de una base de datos de experiencias anteriores.

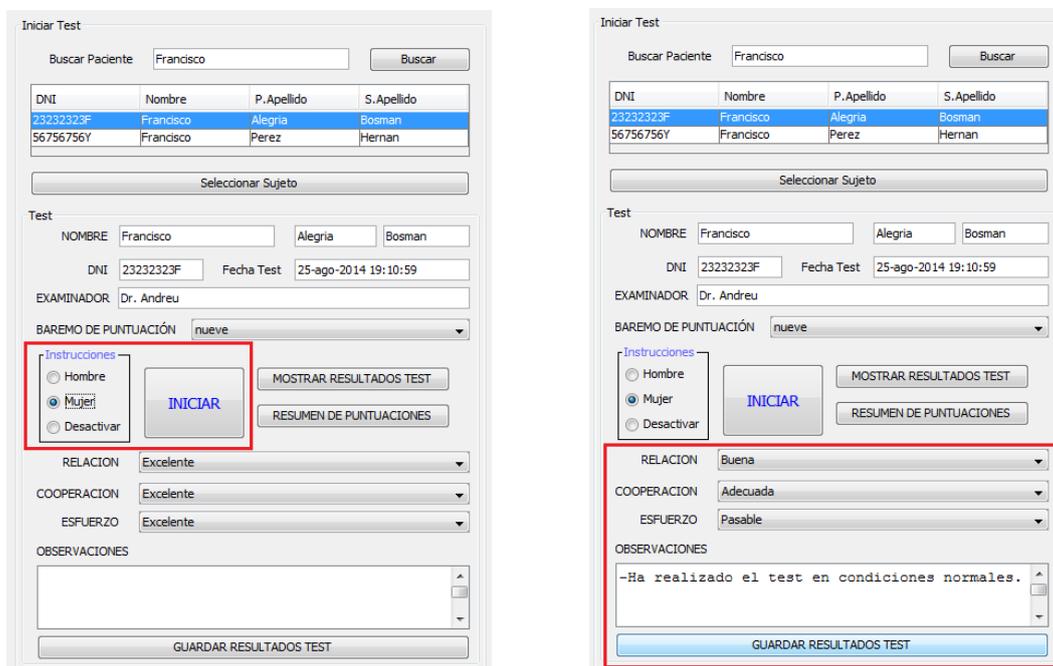
Lo único que queda antes de empezar el test, es seleccionar el interlocutor o narrador virtual de las “instrucciones”, este puede ser la voz de un hombre o de una mujer, y para los casos que el test no es realizado por primera vez es posible marcar la opción “Desactivar”, de esta forma no habrá narración inicial. Las opciones de interlocución se pueden observar remarcadas en la figura 12a. Las interlocuciones son mostradas en el siguiente apartado del manual.

En este momento se orienta la pantalla hacia el sujeto que va realizar la prueba, cuando esté preparado se le da al Botón “Iniciar”. Desde este momento el examinador puede observar, pero no es necesaria su intervención, ya que el test está diseñado para auto-finalizarse cuando se completen los intentos correspondientes.

Una vez finalice el test, las observaciones del examinador sobre: La Relación, Cooperación y Esfuerzo, serán seleccionadas en los desplegados correspondientes, eligiendo entre las posibles opciones para completar el formulario. (Figura 12b).

Una vez hecho esto es posible visualizar los resultados antes de ser guardados, para ello a la derecha del formulario de inicio hay dispuestas 4 columnas que contienen los intentos aplicados por el sujeto durante el test, así como las dimensiones valoradas. Estas son autocompletadas pulsando el botón “Mostrar Resultados Test”.

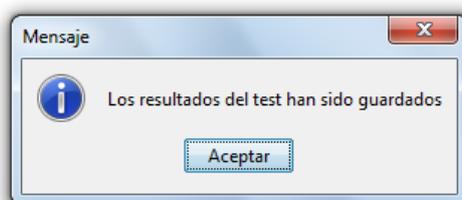
Finalmente el test será almacenado y asignado al paciente, mediante el botón “GUARDAR RESULTADOS TEST”. De esta forma podrá ser visionado posteriormente en la opción Historial.



**Figura 12a.** Formulario de inicio: Instrucciones **Figura 12b.** Formulario de inicio: Observaciones

Si todo el proceso se ha realizado de forma correcta tras pulsar el botón de guardado del test se mostrará el mensaje mostrado en la figura 13.

Una vez descritas las etapas pre y post test, se procede a hacer una descripción de la ejecución del test en los siguientes apartados.



**Figura 13.** Mensaje, resultados guardados de forma correcta.

**2.2.1. Instrucciones y forma de aplicación.**

Antes de empezar con la aplicación del test al sujeto, el examinador debe haber registrado los datos o informaciones de carácter médico o emocional que puedan afectar a la realización del test. A continuación se pedirá al sujeto que se siente frente al PC. Durante la ejecución del test el examinador puede estar o no presente. En caso que esté presente la aplicación servirá para evitar fallos de anotación y que este se pueda centrar de una forma más específica en observar la conducta del sujeto, así como sus emociones. Aunque el test está diseñado para que durante esta parte no sea necesaria la presencia del examinador.

Al inicio de la aplicación de forma tradicional el examinador debe trasladar al sujeto las indicaciones sobre el desarrollo del test, es decir el examinador narrará las instrucciones expuestas en la Figura 14.

*“ Este test es poco usual porque yo no voy a dar mucha información sobre lo que hay que hacer: la tarea consiste emparejar estas tarjetas (señalando los bloque de tarjetas-respuestas) con estas cuatro que he colocado aquí y que sirven de clave (señalar las tarjetas-estímulo sucesivamente, empezando por el triángulo rojo) hay que tomar siempre una tarjeta de este bloque, la de arriba, (señalar) y colocarla debajo de la tarjeta clave con la que se crea que hace pareja (señalar la fila de tarjetas estímulo). Yo no puedo decirle cómo se deben emparejar las tarjetas, pero cada vez que VD, coloque una; le diré si lo ha hecho correctamente o si se ha equivocado. En los casos en que la haya colocado mal, simplemente dejará la tarjeta donde la haya puesto e intentará colocar la siguiente correctamente. Para esta prueba no hay tiempo límite, ¿está Vd. preparado? Puede comenzar”*

Figura 14: Texto narrado por el examinador.

En la aplicación informática, el texto anterior debe ser modificado, ya que ahora la interacción es directamente con el PC. Por tanto es este el que debe transmitir al sujeto las indicaciones necesarias sobre la realización del mismo.

Se han realizado determinadas modificaciones, centradas en las indicaciones que se le deben dar al paciente durante la ejecución de la prueba. Es decir, se le indican las señales tanto visuales como sonoras que van a aparecer durante el test, relacionadas con si las respuestas son correctas o erróneas.

*“Este test es poco usual porque yo no voy a dar mucha información sobre lo que hay que hacer: La tarea consiste en emparejar la tarjeta de abajo con una de las cuatro que hay colocada arriba y que sirven de clave, cada vez que se seleccione una de las tarjetas de arriba que se empareja directamente con la de abajo y a continuación aparece una nueva tarjeta en la parte inferior para emparejarla de nuevo con una de la parte superior. No puedo decir cómo se deben emparejar las tarjetas, pero cada vez que Vd, coloque una; se indicará en pantalla y mediante un sonido si lo ha hecho correctamente o si se ha equivocado. Si es correcta aparecerá una V verde acompañada del siguiente sonido (“sonido correcta”), y si es incorrecta aparecerá una X en rojo y el sonido (“sonido incorrecta”). Para esta prueba no hay tiempo límite, ¿está Vd. preparado? Puede comenzar “*

Figura 15: Texto narrado por el PC.

A causa de la naturaleza ambigua del test los sujetos pueden formular diversas preguntas al examinador, pero es adecuada la aplicación informática ya que no habría que hacer ninguna aclaración debido a que no se puede vulnerar la integridad del WCST dando indicaciones sobre las categorías de clasificación o sobre la naturaleza de los

cambios entre una categoría y la siguiente. Generalmente, la repetición de las instrucciones es suficiente para la mayoría de los sujetos.

En las aplicaciones realizadas por los autores del test se probó que, incluso los niños más pequeños tenían pocas dificultades para seguir las instrucciones normales del WCST. Sin embargo, con algunos de los más pequeños, se observó que la prueba resultaba menos amenazante si se presenta como un juego y no como un test. En este caso de aplicación informática por su propia naturaleza no se presenta este problema, incluso es posible que el niño se sienta en mejor disposición a su realización al ser de forma informatizada y visualmente más atractiva. A continuación aparece la imagen de la pantalla de simulación del test (figura 16), como se puede comprobar en la parte superior aparecen las cuatro tarjetas-estímulo, justo debajo los huecos para la última tarjeta asociada a la directamente superior y finalmente la tarjeta que se ve muestra sola será el mazo, e irá cambiando de figura con cada intento durante la ejecución del test, este siempre seguirá mostrando las cartas de la forma preestablecida.

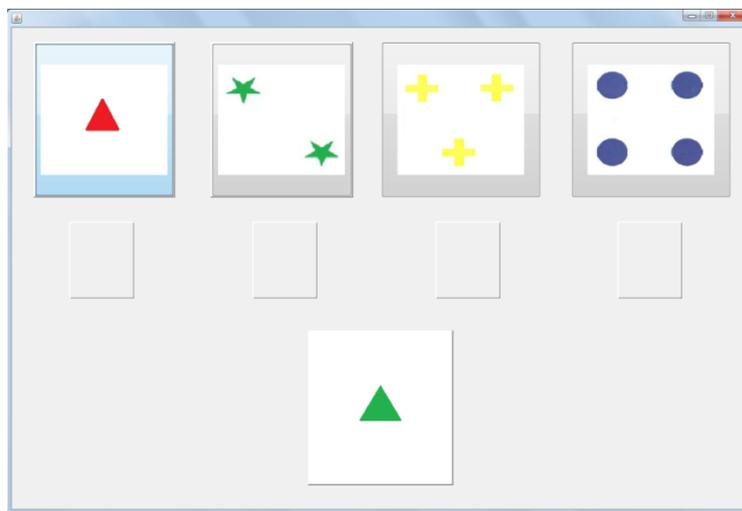


Figura 16: Inicio simulación del test.

**2.2.2 Funcionamiento de la aplicación durante la ejecución.**

Al iniciar el test, la primera categoría correcta de clasificación es color. Cuando el sujeto empiece a colocar las tarjetas respuesta, se le indicará mediante un sonido y una animación si la tarjeta ha sido emparejada de forma correcta en color o de forma incorrecta en cualquier otra categoría. Dicho sonido y animación será diferente en cada caso. En la siguiente figura se puede observar que la animación será una “V” si la respuesta ha sido correcta según la figura 17.a y una “X” si de lo contrario ha sido incorrecta, figura 17.b

La aplicación continúa de este modo hasta que el sujeto haya colocado correctamente diez tarjetas consecutivas, atendiendo a la característica Color.

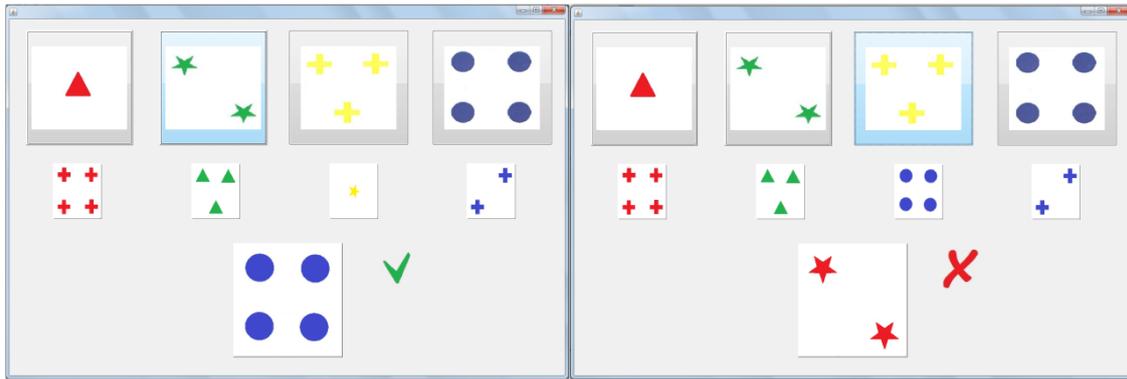


Figura 17.a: Animación: V → "Correcto"

Figura 17.b Animación X → "Incorrecto"

- Sin dar ninguna otra indicación, se cambiará en ese momento la categoría correcta que a partir de entonces será **Forma**; es decir, si el sujeto coloca la tarjeta siguiente atendiendo a Color, que hasta entonces era lo correcto, se indicará que no es correcto.
- **Forma** se mantiene como categoría correcta has que el sujeto haya alcanzado diez respuestas correctas consecutivas; cuando esto sucede, sin aviso ni indicación de los que está sucediendo, se cambiará la categoría de clasificación a Número.
- Cuando el sujeto haya dado diez respuestas correctas consecutivas en la categoría Número se volverá a establecer de nuevo Color como categoría correcta de clasificación, después Forma y después Número, actuando siempre del modo descrito anteriormente.
- La prueba continúa hasta que el sujeto haya completado con éxito seis categorías o hasta que haya utilizado todas las tarjetas , en el caso de que esto ocurra antes de completar las categorías indicadas.

*“No se proporcionará al sujeto ninguna información distinta de la incluida anteriormente en ningún momento, ni durante el desarrollo de la misma, ni después de finalizarla.”*

El WCST no es una prueba con tiempo limitado y en las instrucciones iniciales se informa de ello al sujeto. Aunque el tiempo empleado en las aplicaciones realizadas es bastante variable, la mayoría de las veces se ha completado en un período de 20 o 30 minutos. Si la realización fuese mediante cartas podría suceder que el sujeto comience a colocar las tarjetas muy deprisa dificultando la anotación de las respuestas; en estos casos el examinador pedirá que lo haga más despacio, y deberá dejar constancia de ello en la Hoja de anotación. También puede hacer más lento el “feedback” con el sujeto (por ejemplo no diciendo si la respuesta es correcta o errónea hasta su anotación en la Hoja.

Los problemas anteriores que le surgen al examinador, mediante la aplicación informática se eliminan, y ahora el test puede ser realizado de una forma más precisa en cuanto a la captación de los datos o respuestas

### 3- HISTORIAL.

El historial nos permite visualizar las pruebas realizadas por un sujeto específico, así como su cronología y otros datos médicos de interés. A continuación se muestra un ejemplo, en el que se imprime un test.

Como sujeto se ha supuesto un paciente llamado “Luis Alisado Pelsan”, accediendo a la opción **Historial** → **Buscar Paciente**, se abre el formulario de historiales (Figura 18) , a continuación se ingresa el nombre del paciente y el primer apellido “Francisco Alisado” como criterio de búsqueda. En la siguiente figura se puede observar que aparecen dos sujetos llamados Luis Alisado, de los cuales seleccionamos el que estamos buscando, que tiene como segundo apellido Pelsan (en caso de haber ingresado como criterio de búsqueda el nombre y los dos apellidos, solo habría aparecido en la tabla el sujeto buscado).

Una vez localizado se marca en la tabla y se pulsa el botón “Seleccionar”, los datos relativos al sujeto serán cargados en los correspondientes campos situados debajo de dicha tabla.

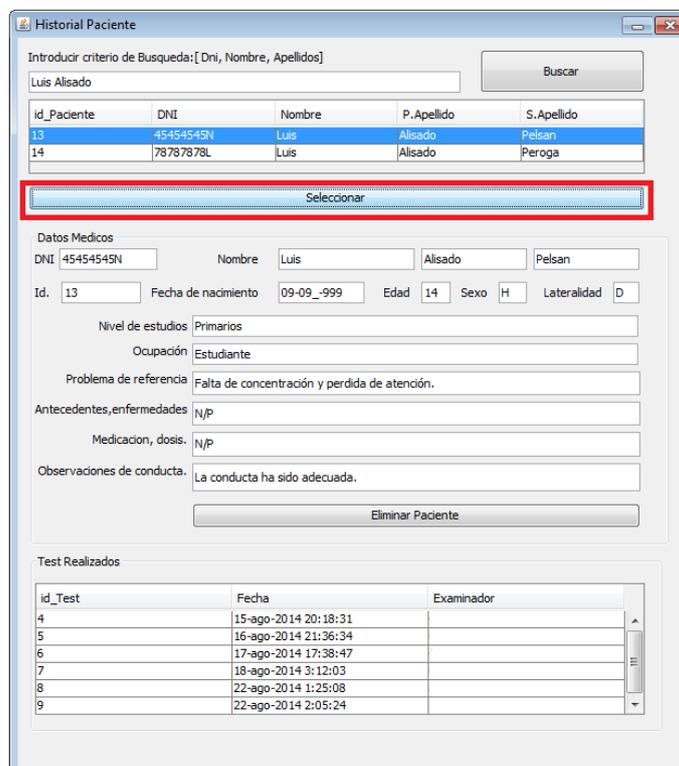
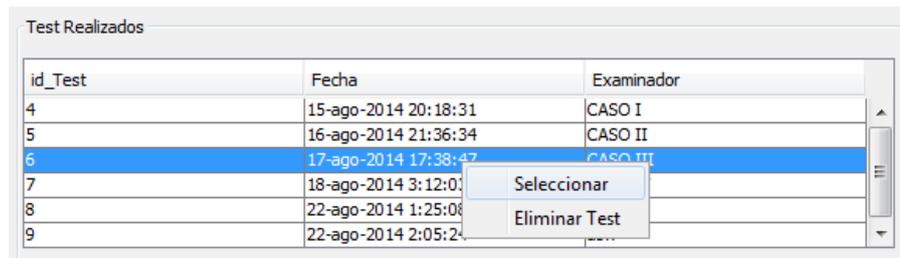


Figura 18: Ventana de Historiales.

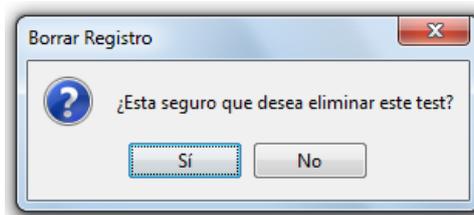
En la última tabla del formulario aparecen los test realizados por el sujeto seleccionado, estos están identificados por un número, además de la fecha de ejecución y del examinador u otra información que este hubiese creído conveniente. Para seleccionar uno de ellos y visualizarlo debe ser marcado y a continuación con botón derecho del ratón se abrirá una cortina (Figura 19) con dos opciones: Seleccionar Test y Eliminar Test, como sus propios nombres indican, la primera opción nos abre un test guardado y la segunda lo elimina.



id_Test	Fecha	Examinador
4	15-ago-2014 20:18:31	CASO I
5	16-ago-2014 21:36:34	CASO II
6	17-ago-2014 17:38:47	CASO III
7	18-ago-2014 3:12:01	
8	22-ago-2014 1:25:08	
9	22-ago-2014 2:05:24	

**Figura 19:** Tabla con los test almacenados en base de datos.

En el caso de pulsar la opción: Eliminar Test, antes de proceder a ser borrado, se abre una ventana de verificación como la mostrada en la figura 20



**Figura 20:** Mensaje: Verificación de borrado.

La opción seleccionar de la cortina nos abre todos los datos relativos a ese test, es decir tanto la hoja de anotación de test como la hoja de resultados y todo en la misma pantalla.

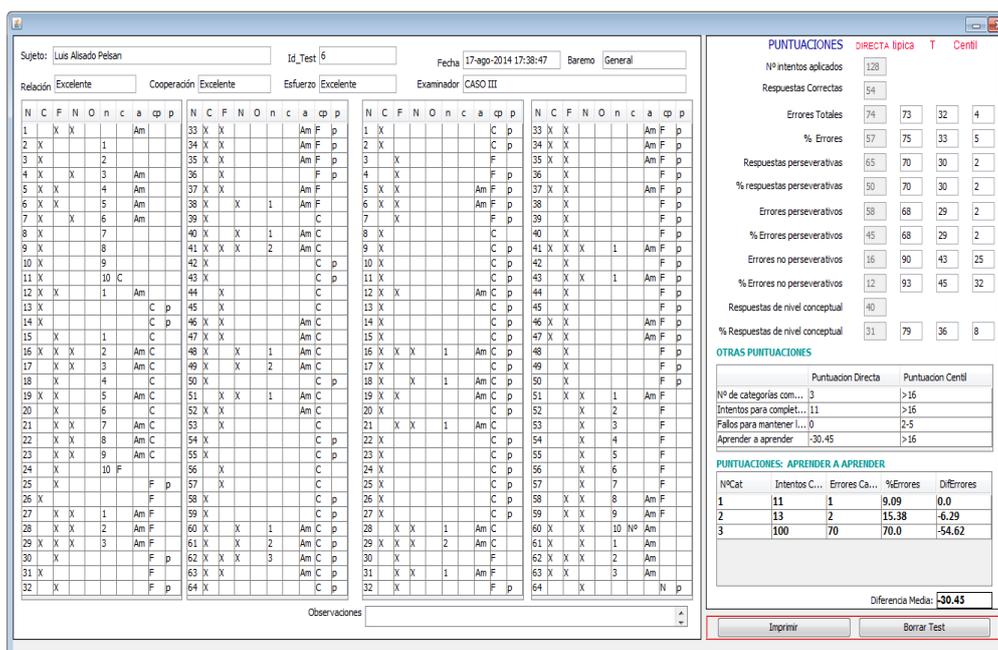


Figura 21: Visionado de un test guardado.

Como se ve en la figura anterior, en esta ventana se dispone de toda la información correspondiente a un test. En la parte de derecha aparece lo que de forma tradicional se llamaba hoja de anotación de respuestas, formada por las 128 filas correspondientes a los posibles intentos que permite el test. Cada fila contiene la información relativa a cada respuesta, de forma más detallada que tradicionalmente. En la zona derecha aparecen los resultados tanto directos como estadísticos y otros cálculos de importancia.

El programa nos permite **imprimir el test**, para ello solamente hay que pulsar en el botón situado en zona baja derecha de la pantalla como se ve en la figura 21. Además también existe la posibilidad de borrar el test desde el mismo sitio.

Si se pulsa el botón imprimir, la aplicación nos lleva directamente al cuadro de diálogo de impresión, figura 22.

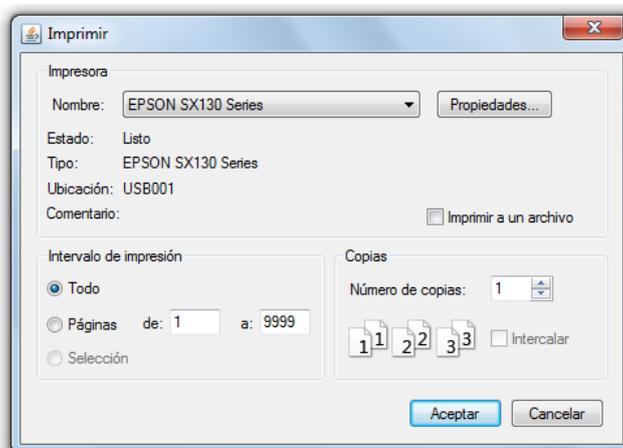


Figura 22: Cuadro de diálogo de impresión.

La impresión va a constar de dos páginas, la primera corresponde a la hoja de anotación de respuestas y la segunda con la hoja de resultados. Figuras 23 y 24. Estas se muestran en las dos páginas siguiente de forma similar a como saldrían tras ser imprimidas.

Sujeto:  Id\_Test:  Fecha:  Baremo:

Relación:  Cooperación:  Esfuerzo:  Examinador:

N	C	F	N	O	n	c	a	cp	p	N	C	F	N	O	n	c	a	cp	p	N	C	F	N	O	n	c	a	cp	p	N	C	F	N	O	n	c	a	cp	p									
1		X	X					Am		33	X	X						Am	F	p	1	X							C	p	33	X	X						Am	F	p							
2	X				1					34	X	X							Am	F	p	2	X							C	p	34	X	X						Am	F	p						
3	X				2					35	X	X							Am	F	p	3	X							F		35	X	X						Am	F	p						
4	X		X		3			Am		36	X									F	p	4	X							F	p	36	X								F	p						
5	X	X			4			Am		37	X	X								Am	F	5	X	X					Am	F	p	37	X	X							Am	F	p					
6	X	X			5			Am		38	X	X		1						Am	F	6	X	X					Am	F	p	38	X									F	p					
7	X		X		6			Am		39	X									C		7	X						F	p	39	X										F	p					
8	X				7					40	X	X		1						Am	C	8	X						C		40	X										F	p					
9	X				8					41	X	X	X	2						Am	C	9	X						C	p	41	X	X	X	1					Am	F	p						
10	X				9					42	X									C	p	10	X						C	p	42	X											F	p				
11	X				10	C				43	X									C	p	11	X						C	p	43	X	X		1					Am	F	p						
12	X	X			1			Am		44	X									C		12	X	X					Am	C	p	44	X										F	p				
13	X								C	p	45	X								C		13	X						C	p	45	X											F	p				
14	X								C	p	46	X	X							Am	C	14	X						C	p	46	X	X								Am	F	p					
15	X				1				C		47	X	X							Am	C	15	X						C	p	47	X	X								Am	F	p					
16	X	X	X		2			Am	C	48	X	X		1						Am	C	16	X	X	X	1			Am	C	p	48	X											F	p			
17	X	X			3			Am	C	49	X	X		2						Am	C	17	X						C	p	49	X											F	p				
18	X				4				C	50	X									C	p	18	X	X		1			Am	C	p	50	X											F	p			
19	X	X			5			Am	C	51	X	X		1						Am	C	19	X	X					Am	C	p	51	X	X		1						Am	F					
20	X				6				C	52	X	X								Am	C	20	X						C	p	52		X		2								F					
21	X	X			7			Am	C	53	X									C		21	X	X		1			Am	C	53		X		3								F					
22	X	X			8			Am	C	54	X									C	p	22	X						C	p	54		X		4								F					
23	X	X			9			Am	C	55	X									C	p	23	X						C	p	55		X		5								F					
24	X				10	F				56	X									C		24	X						C	p	56	X			6								F					
25	X								F	p	57	X								C		25	X						C	p	57	X			7								F					
26	X								F		58	X								C	p	26	X						C	p	58	X	X		8						Am	F						
27	X	X			1			Am	F	59	X									C	p	27	X						C	p	59	X	X		9						Am	F						
28	X	X			2			Am	F	60	X	X		1						Am	C	p	28	X	X		1			Am	C	60	X	X		10	Nº				Am							
29	X	X	X		3			Am	F	61	X	X		2						Am	C	p	29	X	X	X	2			Am	C	61	X	X		1					Am							
30	X								F	p	62	X	X	X	3					Am	C	p	30	X						F		62	X	X	X	2					Am							
31	X								F		63	X	X							Am	C	p	31	X	X		1			Am	F	63	X	X		3					Am							
32	X								F	p	64	X								C	p	32	X						F	p	64		X								N	p						

Observaciones:

Figura 23: Hoja de Anotación de respuestas.

PUNTUACIONES		DIRECTA típica	T	Centil
Nº intentos aplicados	128			
Respuestas Correctas	54			
Errores Totales	74	73	32	4
% Errores	57	75	33	5
Respuestas perseverativas	65	70	30	2
% respuestas perseverativas	50	70	30	2
Errores perseverativos	58	68	29	2
% Errores perseverativos	45	68	29	2
Errores no perseverativos	16	90	43	25
% Errores no perseverativos	12	93	45	32
Respuestas de nivel conceptual	40			
% Respuestas de nivel conceptual	31	79	36	8
<b>OTRAS PUNTUACIONES</b>				
	Puntuacion Directa	Puntuacion Centil		
Nº de categorías com...	3	>16		
Intentos para complet...	11	>16		
Fallos para mantener l...	0	2-5		
Aprender a aprender	-30.45	>16		
<b>PUNTUACIONES: APRENDER A APRENDER</b>				
NºCat	Intentos C...	Errores Ca...	%Errores	DifErrores
<b>1</b>	<b>11</b>	<b>1</b>	<b>9.09</b>	<b>0.0</b>
<b>2</b>	<b>13</b>	<b>2</b>	<b>15.38</b>	<b>-6.29</b>
<b>3</b>	<b>100</b>	<b>70</b>	<b>70.0</b>	<b>-54.62</b>
Diferencia Media: <b>-30.45</b>				

Figura 24: Hoja de resultados.

### 2.3.1 Descripción de la hoja de anotación

Como se ha descrito anteriormente la hoja de anotación recoge todas las respuestas y dimensiones utilizadas. Aparecen cuatro columnas que son donde se quedan registradas las respuestas dadas por el sujeto a los 128 ítems (cuatro columnas de 32 ítems).

Las dos primeras columnas contienen los espacios numerados del 1 al 64 en los que se anotarán las respuestas a los elementos correspondientes. Lo mismo sucede con las dos últimas en las que se anotarán las respuestas correspondientes al segundo conjunto de tarjetas. En la siguiente figura se muestra un trozo de columna con los intentos 1 al 16, a continuación se describe el contenido de cada uno de los ítems o respuestas.

	N	C	F	N	O	n	c	a	cp	p
1			X	X				Am		
2	X					1				
3	X					2				
4	X		X			3		Am		
5	X	X				4		Am		
6	X	X				5		Am		
7	X		X			6		Am		
8	X					7				
9	X					8				
10	X					9				
11	X					10	C			
12	X	X				1		Am		
13	X								C	p
14	X								C	p
15		X				1			C	
16	X	X	X			2		Am	C	

Figura 25: Intentos del 1 al 16, hoja de anotación.

- Fila 1. N: Número de intento, en la figura aparece rango de intentos desde el 1 hasta el 16.
- Filas 2, 3, 4, 5 correspondientes a las categorías “C”: Color, “F”: Forma, ”N” : Número y “O” : Otros.

Si la tarjeta-respuesta y la tarjeta-estímulo se emparejan solamente en una dimensión, la hoja de anotación se marcará con una X solamente en el hueco que corresponde a esa dimensión. Por ejemplo si la tarjeta-respuesta se empareja con la tarjeta-estímulo solamente en cuanto a Color en la hoja de anotación se marcará este ítem poniendo sobre el intento en cuestión una X en la categoría color. Si la tarjeta-respuesta se empareja con la tarjeta estímulo en dos dimensiones, se marcarán con una X los dos huecos correspondientes a estas dimensiones en el intento correspondiente. Finalmente, si la respuesta no se empareja con la tarjeta estímulo en ninguna dimensión se marcará en la hoja de anotación la opción Otros (C,F,N,Ø). Las respuestas se

anotarán de esta forma en cada ítem, con independencia de que sean correctas o incorrectas respecto a la categoría de clasificación que esté considerando.

- Fila 6 ó “n”: esta columna pertenece a la dimensión correcto-incorrecto, en esta solamente se numeran las respuestas correctas, si el sujeto interrumpe la secuencia con un error, cuando vuelva a dar una respuesta correcta se volverá a comenzar la numeración con el número 1, cuando se alcance el criterio de diez respuestas consecutivas, la categoría que acaba de completarse será indicada en la siguiente columna, entonces se vuelve a iniciar la numeración.

Que la respuesta sea correcta o incorrecta depende de que el sujeto empareje las tarjetas considerando la categoría establecida, cada vez que da una respuesta. En la categoría de clasificación establecida se puntúan como correctas y las que no corresponden, como erróneas, en estas aparecerá el hueco correspondiente en blanco.

- Fila 7, “c” categoría acertada, cada vez que se completa una categoría, el hueco reservado a este, indicará la dimensión de dicha categoría. Es decir, este espacio está reservado para marcar el final de categoría acertada, si la categoría presente es forma y el sujeto acaba de dar diez respuesta en forma, en el ítem correspondiente aparecerá “F”.
- Fila 8, “Am”, ambigüedad, este espacio es reservado para indicar la ambigüedad o no de la respuesta correspondiente.

Cabe destacar que una respuesta es ambigua cuando una tarjeta-respuesta se empareja con una tarjeta-estímulo atendiendo a una, y sólo una de sus características (por ejemplo, Color), resulta obvia, no ambigua, la categoría utilizada por el sujeto para realizar el emparejamiento. Esto sucede si, por ejemplo, una tarjeta-respuesta con tres círculos rojo se ha emparejado con la tarjeta-estímulo que tiene un triángulo rojo. Sin embargo las tarjetas pueden emparejarse en más de una dimensión. Un ejemplo de esto ocurre cuando una tarjeta-respuesta con tres triángulos rojos se empareja con el estímulo que sólo tiene un triángulo rojo. En esta situación, la respuesta es ambigua, pues no puede estar seguro de que el sujeto haya clasificado esa tarjeta considerando la categoría Color o la categoría Forma.

Por tanto, cuando la tarjeta-respuesta se empareja con la tarjeta estímulo en una, y sólo una, característica se considera una respuesta no ambigua y cuando la tarjeta respuesta se empareja con la tarjeta-estímulo en dos o más características se considerará ambigua. Por definición la categoría otros contienen siempre respuestas ambiguas. La dimensión ambigua-no ambigua es independiente de que la respuesta sea correcta o errónea.

- Fila 9. “cp” categoría del principio de perseveración. Las posibles categorías de perseveración serán Color, Forma o Número.

- Fila 10. “P” Respuesta perseverativa. En el espacio reservado a esta dimensión aparecerá una “p” en caso de que la respuesta sea perseverativa y quedará en blanco en caso de que no lo sea.

Las dos últimas filas están reservadas para la dimensión **Perseverativa-no perseverativa**: Cuando el sujeto persiste en responder una característica del estímulo que no es correcta, la respuesta se considera una perseveración hacia ese criterio y se puntúa como “perseverativa”. El sujeto puede perseverar en Color, Forma o Número. Sin embargo, no es posible que esto suceda en la categoría Otras porque el examinador no puede estar seguro del criterio de clasificación seguido por el sujeto. El principio de perseveración es operativo una vez establecido. De hecho las respuestas que se emparejen con el principio de perseveración establecido se consideran perseverativas, prescindiendo que sean correctas o incorrectas. Las respuestas que no se emparejen siguiendo el principio de perseveración se consideran no perseverativas.

### 2.3.2 Descripción de la hoja de resultados.

La hoja de resultados está dividida en tres zonas diferenciadas. La zona superior corresponde a las puntuaciones Directa y Estadísticas. A su vez esta última se divide en tres tipos de puntuaciones, puntuación Típica, T y Centil.

PUNTUACIONES	DIRECTA	típica	T	Centil
Nº intentos aplicados	128			
Respuestas Correctas	54			
Errores Totales	74	73	32	4
% Errores	57	75	33	5
Respuestas perseverativas	65	70	30	2
% respuestas perseverativas	50	70	30	2
Errores perseverativos	58	68	29	2
% Errores perseverativos	45	68	29	2
Errores no perseverativos	16	90	43	25
% Errores no perseverativos	12	93	45	32
Respuestas de nivel conceptual	40			
% Respuestas de nivel conceptual	31	79	36	8

Figura 26: Hoja de resultados 1/3

Las puntuaciones que aparecen en esta hoja son las siguientes:

- **Número de intentos aplicados:** corresponde al total de intentos aplicados durante la prueba, será 128 en caso de que el sujeto no consiga llegar a seis categorías correctas.

- **Respuestas correctas:** como su nombre indica corresponde al número de aciertos conseguidos por el sujeto.

Para las siguientes puntuaciones además de la puntuación directa obtenida en la prueba, el software nos da *los resultados estadísticos* como **puntuación típica, puntuación T y puntuación Centil**. Para ello el software utiliza tablas y ábacos estadísticos.

- **Respuestas perseverativas.** indica el número de respuestas consideradas como perseverativas o lo que es lo mismo el número de “p” que aparecen en la última columna de la hoja de respuestas
- **Errores perseverativos:** número de respuestas perseverativas que además son incorrectas.
- **Errores no perseverativos:** número de respuestas que son incorrectas y además no son perseverativas.
- **Respuestas de nivel conceptual y porcentaje de respuestas de nivel conceptual:** refleja la comprensión de los principios de clasificación. El nivel conceptual de respuestas está definido por series de tres o más respuestas correctas consecutivas. Estas repuestas son llamadas respuestas de nivel conceptual porque se presume que requieren alguna comprensión de la estrategia necesaria para hacer colocaciones correctas y que es poco probable conseguir una serie correcta de esta longitud por azar. El porcentaje de respuestas de nivel conceptual se calcula dividiendo el número total de respuestas de este tipo entre el número total de intentos, multiplicando el resultado por 100 y redondeando al número obtenido al entero más próximo.

Por ejemplo si en los 128 elementos del test, el sujeto dio 70 respuestas de Nivel Conceptual en diez series de tres o más respuestas consecutivas correctas (10, 10,10,3,10,3,4,4,10 y 6) Dividiendo 70 (número de respuestas de nivel conceptual) entre 128, multiplicando el resultado por 100 y redondeándolo al entero más próximo se obtiene el porcentaje de respuestas de Nivel Conceptual, que en este caso es 55%.

- **Porcentaje de errores, de respuestas perseverativas y de errores no perseverativos:** estas puntuaciones se presentan principalmente como ayuda para posibles investigaciones. Se calculan dividiendo la puntuación directa de cada uno de esos aspectos entre el número de intentos realizados, multiplicando el resultado por 100 y redondeando la cifra resultante al entero más próximo. . Las puntuaciones de porcentaje de errores pueden resultar muy útiles en estudios de investigación en los que sea necesario controlar las diferencias en el número de intentos administrados. Sin embargo, no se recomienda su uso para la interpretación clínica del WCST porque implican valores corregidos para una medida del éxito en todo el test (el número de intentos empleados para completar la prueba), y porque los índices de

fiabilidad de estos porcentajes son más bajos que los de las puntuaciones que los generan.

La segunda zona corresponde a otras puntuaciones, como se puede observar en la figura xx, proporciona información de la puntuación directa y de la puntuación centil. A continuación se describen estas puntuaciones.

OTRAS PUNTUACIONES		
	Puntuacion Directa	Puntuacion Centil
Nº de categorías com...	3	>16
Intentos para complet...	11	>16
Fallos para mantener l...	0	2-5
Aprender a aprender	-30.45	>16

Figura 27: Hoja de resultados 2/3

- **Categorías completas:** El número de categorías completa es, simplemente, el número de secuencia de diez respuestas correctas consecutivas, que siguen el criterio de clasificación de una categoría, dadas por el sujeto en la prueba. En este apartado, las puntuaciones van desde un mínimo de 0 a un máximo de 6.
- **Intentos para completar la primera categoría:** El número total de intentos para completar correctamente la primera categoría proporciona una indicación de la conceptualización inicial antes de que se produzca un cambio de ordenación de categoría. En los casos en que esto sucede, se concede al sujeto la puntuación directa 128 intentos para completar la primera categoría.
- **Porcentaje de errores perseverativos:** el porcentaje de errores perseverativos refleja la densidad o concentración de los errores perseverativos en la realización del test. Se calcula dividiendo el número de errores perseverativos por el número de intentos realizados. El resultado de esta operación se multiplica por 100 y se redondea al entero más próximo. Por ejemplo, para un sujeto que haya completado el WCST en 106 intentos y con 15 errores perseverativos, el porcentaje de errores perseverativos será el 14%.
- **Fallos para mantener la actitud:** Otra puntuación básica del WCST es la correspondiente a los fallos en el mantenimiento de la actitud. Este tipo de fallo se produce cuando el sujeto da cinco o más repuestas correctas consecutivas y luego comete un error antes de completar la categoría con éxito. El fallo en mantenimiento de la actitud no se puntúa si, antes de finalizar el test, el sujeto coloca correctamente cinco tarjetas o más y luego simplemente pasa las tarjetas hasta llegar a la 128.

La última zona de la hoja de resultados corresponde a la puntuación Aprender a Aprender. esta refleja el promedio de los cambios en eficacia conceptual del sujeto a través de las diversas etapas de su realización en el WCST

PUNTUACIONES: APRENDER A APRENDER				
NºCat	Intentos C...	Errores Ca...	%Errores	DifErrores
1	11	1	9.09	0.0
2	13	2	15.38	-6.29
3	100	70	70.0	-54.62
				Diferencia Media: -30.45

Figura 28: Hoja de resultados 3/3

. El cálculo de esta puntuación requiere de cuatro pasos.

- 1- Determinar si es posible o no calcular la puntuación Aprender a aprender. Para ello es preciso contar el número de categorías que el sujeto ha completado o intentado. Se considera que se ha “intentado” una categoría cuando en ella se han realizados por lo menos 10 intentos, incluso si dicha categoría no se ha completado correctamente (por ejemplo, al final del test). La puntuación Aprender a aprender solamente se puede calcular en los casos en que el sujeto ha completado tres categorías o más o si ha completado dos e intentado la tercera.
  
- 2- El segundo paso es calcular el porcentaje de errores en cada categoría que se ha completado o intentado (total de errores cometidos en esa categoría dividido por el total de intentos realizados en ella y multiplicado por 100).
  
- 3- Calcular las diferencias entre el porcentaje de errores de dos categorías consecutivas o adyacentes. La diferencia entre porcentajes de errores se obtiene restando el correspondiente a la categoría 2 (Forma) del correspondiente a la categoría 1 (Color); el correspondiente a la categoría 3 (Número) del correspondiente a la categoría 2 (Forma) y así sucesivamente. Para finalizar, los valores de las diferencias en los porcentajes de errores se suman y se calcula la media que constituye la puntuación directa en Aprender a aprender. Una puntuación positiva en este factor indica una mejora en eficacia a través de la realización de categorías consecutivas, presumiblemente a causa del aprendizaje. Sin embargo es, probable que la mayoría de los sujetos, obtengan una puntuación negativa en ese aspecto porque se incluye en el cálculo la diferencia del porcentaje de errores entre la categoría 1 y la categoría 2 y esta inclusión hace más difícil obtener una puntuación positiva en Aprender a aprender dada la mayor dificultad que representa realizar el primer cambio de la categoría de clasificación. Sin embargo, en los datos normativos se ha tenido en cuenta este hecho; es decir, los baremos permiten comparar la puntuación de un sujeto dado con la obtenida por su grupo de referencia (que tiene sus mismas características) y

así es posible identificar una realización relativamente buena incluso si la puntuación total no es positiva.

#### 4- INTERPRETACIÓN.

EL software ofrece una opción para visualizar herramientas (tablas y ábacos), de las que el profesional puede ayudarse para obtener una mejor interpretación de los resultados.

Estas tablas corresponden a métodos, valores típicos y de referencia. De esta forma el examinador no tiene que buscar dicha información en la documentación, simplemente abriendo el visor dispondrá de ellos. Este está disponible en el menú inicio, en la opción: **Info>>Muestra de tipificación(Valores de referencia)**

Tras acceder a dicha opción se abre en una ventana el visor, según la figura 29, este se puede mantener abierto simultáneamente con la ventana de resultados y así poder realizar una comprobación sobre los datos.

Valores de Referencia

Porcentaje de adultos de las muestras normal y clínica, según la localización de la lesión, en los distintos niveles de puntuaciones típicas.

DETERIORO	Nivel de realización								
	Muestra Clínica						Muestra Normal		
	Grave	Severo	Moderada-severo	Moderado	Moderado-medio	medio	Debajo de la media	Media	Encima de la media
Puntuación típica	0-84	0-54	55-61	62-69	70-76	77-84	85-91	92-106	107+
Puntuación T	0-39	0-19	20-24	25-29	30-34	35-39	40-44	45-54	55+
% predicho en la distrib normal.	14,6	0,1	0,4	1,5	4,0	6,6	14,4	38,2	32,8
WCST puntuación									
Nº total errores									
Grupo normal	14.2	0.3	0.8	1.6	4.2	7.3	9.4	47.9	28.6
Grupo clínico	57.8	4.4	9.9	11.1	15.2	17.2	8.7	22.2	11.4
total	59.4	10.2	11.9	8.5	16.9	11.9	11.9	20.3	8.5
Frontal	62.2	1.9	11.3	11.3	15.1	22.6	9.4	15.1	13.2
Frontal más	61.1	4.5	10.2	11.9	14.7	19.8	9.0	20.9	9.0
Difuso	40.8	0.0	5.6	11.1	14.8	9.3	3.7	35.2	20.4
No frontal									
% de errores									
Grupo normal	12.4	0.5	1.0	1.0	3.1	6.08	11.2	48.4	27.9
Grupo clínico	54.2	2.9	10.5	9.3	15.5	16.0	12.0	22.4	11.4
total	57.8	6.8	15.3	5.1	15.3	15.3	8.5	20.3	13.6
Frontal	56.6	1.9	11.3	9.4	17.0	17.0	17.0	17.0	9.4
Frontal más									

Anterior      Siguiente

Figura 29: Visor de valores de referencia.

El visor nos muestra información, como por ejemplo el método Heaton de clasificación, que sirve para interpretar las puntuaciones normativas basándose en experiencias clínicas anteriores.

Sistema de Heaton de clasificación para interpretar las puntuaciones normativas basándose en su experiencia clínica		
Las siguientes agrupaciones de puntuaciones típicas (media=100 y desviación típica =15) y puntuaciones T (media=50 y desviación típica= 10), definen estos niveles clínicamente relevantes.		
Puntuaciones Típicas	Puntuaciones T	Nivel de Deterioro
$\mu \leq 54$	$\leq 19$	Grave
$55 \leq \mu \leq 61$	$20 \leq t \leq 24$	Grave a Moderado
$62 \leq \mu \leq 69$	$25 \leq t \leq 29$	Moderado
$70 \leq \mu \leq 76$	$30 \leq t \leq 34$	Moderado a Intermedio
$77 \leq \mu \leq 84$	$35 \leq t \leq 39$	Intermedio
$85 \leq \mu \leq 91$	$40 \leq t \leq 44$	Superior a la media
$92 \leq \mu \leq 106$	$45 \leq t \leq 54$	Medio
$\mu \geq 107$	$t > 55$	Inferior a la media

Figura 30: Tabla método Heaton.

En el visor se pueden encontrar tablas de tipificación de muestras normal y clínica, haciendo la distinción entre adultos y niños o adolescentes. También estarán clasificadas según la localización de la lesión o categorías del diagnóstico. Los niveles de puntuación utilizados estarán referenciados a puntuaciones típicas o centiles. A continuación aparece la descripción de las tablas incorporadas.

- Porcentaje de adultos de las muestras normal y clínica, según la localización de la lesión, en los distintos niveles de puntuaciones típicas.
- Porcentaje de adultos de las muestras normal y clínica, según localización de la lesión, en los distintos centiles.

Además se mostrarán las cuatro siguientes:

- Porcentaje de niños y adolescentes de las muestras normal y clínica, según la localización de la lesión, en los distintos niveles de puntuaciones típicas.
- Porcentaje de niños y adolescentes de las muestras normal y clínica, según localización de la lesión, en los distintos centiles.
- Porcentaje de niños y adolescentes de las muestras normal y clínica, según la categoría del diagnóstico, en los distintos niveles de puntuaciones típicas.
- Porcentaje de niños y adolescentes de las muestras normal y clínica, según la categorías del diagnósticos, en los distintos centiles.

Porcentaje de <b>adultos</b> de las muestras normal y clínica, según <b>localización de la lesión</b> , en los <b>distintos centiles</b> .							
WCST PUNTUACIÓN		Centiles					
		Total 0-16	0-1	2-5	6-10	11-6	>16
<b>Nº de categorías completas</b>	Grupo normal	13.5	0.8	2.1	7.0	3.6	86.5
	Grupo clínico total	58.1	14.0	16.9	15.5	11.7	42.0
	Frontal	64.5	22.0	13.6	15.3	13.6	35.6
	Frontal más	60.4	5.7	30.2	9.4	15.1	39.6
	Difuso	59.4	14.7	14.7	18.1	11.9	40.7
	No frontal	44.5	11.1	14.8	13.0	5.6	55.6
<b>Intentos para completar la 1ª categoría.</b>	Grupo normal	11.7	0.5	3.4	4.4	3.4	88.3
	Grupo clínico total	40.9	11.1	16.9	7.9	5.0	59.2
	Frontal	34.0	15.3	10.2	6.8	1.7	66.1
	Frontal más	49.0	7.5	24.5	11.3	5.7	50.9
	Difuso	42.9	11.9	17.5	7.9	5.6	57.1
	No frontal	33.4	7.4	14.8	5.6	5.6	66.7
<b>Fallos para mantener la actitud.</b>	Grupo normal	10.7	0.8	1.0	4.7	4.2	89.3
	Grupo clínico total	21.0	2.6	4.4	6.4	7.6	79.0
	Frontal	22.1	3.4	5.1	5.1	8.5	78.0
	Frontal más	18.8	1.9	7.5	1.9	7.5	81.1
	Difuso	22.1	2.3	3.4	8.5	7.9	78.0
	No frontal	18.6	3.7	3.7	5.6	5.6	81.5
<b>Aprender a aprender</b>	Grupo normal	14.9	1.2	3.2	4.4	6.1	85.1
	Grupo clínico total	40.5	5.3	13.6	10.2	11.4	59.5
	Frontal	46.5	9.3	18.6	9.3	9.3	53.5
	Frontal más	46.6	4.7	16.3	7.0	18.6	53.5
	Difuso	39.8	6.0	12.0	12.8	9.0	60.2
	No frontal	31.1	0.0	11.1	6.7	13.3	68.9

Muestras utilizadas: Normal (N=356); Frontal (N=59); Frontal más (N=53); Difusa (N=177); No frontal (N=55). Los porcentajes correspondientes a Aprender a Aprender se han calculado sobre una muestra de 342 sujetos normales, 43 del grupo frontal, 43 del Frontal más, 133 del Difuso y 45 de No frontal, para los que ha sido posible calcular puntuación

Figura 31: Tabla ejemplo de valores de referencia.

**ANEXO III**

**EJEMPLOS DE APLICACIÓN**

Los casos que se presentan a continuación han sido simulados con la aplicación para comprobar el correcto funcionamiento de la misma. Obteniendo para todos los mismos resultados que en los casos bibliográficos.

Son cuatro ejemplos de test realizados a sujetos reales. Para los dos últimos casos la justificación incluye el diagnóstico de los pacientes examinados contrastando los resultados de otras pruebas con los obtenidos por el WCST.

En el CD del proyecto, se incluye la documentación de los cuatro casos, pudiéndose comprobar la adecuación de la aplicación en el cálculo de los resultados.

**CASO 1: Varón de 45 años**

**1.1 Hoja de Respuestas**

N	C	F	N	O	n	c	a	cp	p
1		X	X					Am	
2			X					N	
3	X				1			N	
4	X		X		2		Am	N	
5	X	X			3		Am	N	
6	X	X			4		Am	N	
7	X		X		5		Am	N	
8	X				6			N	
9	X				7			N	
10	X				8			N	
11	X				9			N	
12	X	X			10	C	Am		
13	X							C	p
14	X							C	p
15	X							C	p
16	X	X	X		1		Am	C	p
17	X							C	p
18	X		X				Am	C	p
19	X	X			1		Am	C	p
20	X							C	p
21	X							C	p
22		X	X		1		Am	C	
23	X	X	X		2		Am	C	
24	X				3			C	
25	X				4			C	
26	X				5			C	
27	X	X	X		6		Am	C	
28	X	X	X		7		Am	C	
29	X	X	X		8		Am	C	
30	X				9			C	
31	X	X	X		10	F	Am		
32	X							F	p

N	C	F	N	O	n	c	a	cp	p
33	X	X						Am	F p
34	X	X						Am	F p
35	X	X						Am	F p
36	X								F p
37	X	X							Am F
38	X		X		1			Am	F
39	X								F
40	X		X		1			Am	F
41	X	X	X		2			Am	F
42	X								F
43	X	X			1			Am	F
44	X	X	X		2			Am	F
45		X			3				F
46		X			4				F
47		X			5				F
48	X		X		6			Am	F
49	X		X		7			Am	F
50			X		8				F
51	X	X			9			Am	F
52	X		X		10	Nº			
53		X							N p
54		X							N p
55		X							N p
56		X							N p
57		X							N p
58	X	X						Am	N p
59	X	X						Am	N p
60	X	X	X		1			Am	N p
61	X	X	X		2			Am	N p
62	X	X	X		3			Am	N p
63		X							N p
64		X							N p

N	C	F	N	O	n	c	a	cp	p
1		X	X					Am	N p
2			X						N p
3	X				1				N
4	X		X		2			Am	N
5	X	X			3			Am	N
6	X	X			4			Am	N
7	X		X		5			Am	N
8	X				6				N
9	X				7				N
10	X				8				N
11	X				9				N
12	X	X			10	C	Am		
13	X								C p
14	X								C p
15	X				1				C
16	X	X	X		2			Am	C
17	X	X	X		3			Am	C
18	X		X					Am	C
19	X	X			1			Am	C
20	X								C p
21	X	X	X		1			Am	C
22	X	X	X		2			Am	C
23	X	X	X		3			Am	C
24	X				4				C
25	X								C
26	X				1				C
27	X	X	X		2			Am	C
28	X	X	X		3			Am	C
29	X	X	X		4			Am	C
30	X	X	X					Am	C
31	X	X	X		1			Am	C
32	X	X	X					Am	C

N	C	F	N	O	n	c	a	cp	p
33			X						C
34			X						C
35	X	X			1			Am	C
36			X		2			Am	N
37				X					Am C
38	X				1				C
39	X				2				C
40	X				3				C
41	X	X	X		4			Am	C
42	X				5				C
43	X	X			6			Am	C
44	X				7				C
45	X				8				C
46	X	X			9			Am	C
47	X	X			10	F	Am		
48	X								F p
49	X								F p
50	X								F p
51	X	X			1			Am	F p
52	X	X						Am	F p
53	X								F p
54	X								F p
55	X								F p
56	X								F p
57	X								F p
58	X	X	X		1			Am	F
59	X	X	X		2			Am	F
60	X	X	X		3			Am	F
61	X	X	X		4			Am	F
62	X	X	X		5			Am	F
63	X	X	X		6				F
64	X								F

## 1.2 Hoja de Resultados

PUNTUACIONES		DIRECTA	típica	T	Centil
Nº intentos aplicados	128				
Respuestas Correctas	79				
Errores Totales	49	89	43	23	
% Errores	38	91	44	27	
Respuestas perseverativas	40	83	39	13	
% respuestas perseverativas	31	83	39	13	
Errores perseverativos	34	83	39	13	
% Errores perseverativos	26	83	39	13	
Errores no perseverativos	15	100	50	50	
% Errores no perseverativos	11	106	54	66	
Respuestas de nivel conceptual	70				
% Respuestas de nivel conceptual	54	93	45	32	

OTRAS PUNTUACIONES		
	Puntuacion Directa	Puntuacion Centil
Nº de categorías com...	5	>16
Intentos para complet...	12	>16
Fallos para mantener l...	1	>16
Aprender a aprender	-8.43	>16

PUNTUACIONES: APRENDER A APRENDER				
NºCat	Intentos C...	Errores Ca...	%Errores	DifErrores
1	12	2	16.67	0.0
2	19	7	36.84	-20.18
3	21	8	38.1	-1.25
4	24	11	45.83	-7.74
5	35	11	31.43	14.4
6	17	10	58.82	-27.39

Diferencia Media: **-8.43**

## 1.3 Interpretación

En este caso el sujeto ha completado correctamente cinco categorías e intentando una sexta (realizando en ella 17 intentos). Por tanto se puede calcular la puntuación Aprender a aprender. Las puntuaciones correspondientes a diferencias entre los porcentajes de errores en pares de categorías adyacentes son muy significativos, al igual que la suma de los valores de las puntuaciones diferencia de porcentaje de errores con un valor de -42.15 correspondiendo esta cifra a una diferencia media, o una puntuación directa en Aprender a aprender de -8.43 (-42.5 dividido entre 5). Esta puntuación indica claramente que el sujeto se hace menos eficaz, en lugar de hacerse más, durante las sucesivas etapas del WCST, sin embargo, al compararlo con el baremo la puntuación centil revela que en este aspecto la realización del sujeto no resulta inusual en personas neurológicamente normales.

La realización del sujeto mencionado contrasta con la de otro sujeto cuyos datos aparecen en el siguiente caso.

**CASO 2: Mujer 57 años**

**2.1 Hoja de Respuestas**

N	C	F	N	O	n	c	a	cp	p	N	C	F	N	O	n	c	a	cp	p	N	C	F	N	O	n	c	a	cp	p	N	C	F	N	O	n	c	a	cp	p
1		X	X					Am		33	X	X			1		Am			33	X	X			3		Am	C											
2	X				1					34	X	X			2		Am			34	X	X			4		Am	C											
3	X				2					35	X	X			3		Am			35	X	X			5		Am	C											
4	X		X		3			Am		36	X							C p		36	X				6			C											
5	X	X			4			Am		37	X	X			1		Am	C p		37	X	X			7		Am	C											
6	X	X			5			Am		38	X		X				Am	C p		38	X				8			C											
7	X		X		6			Am		39	X							C p		39	X				9			C											
8	X				7					40	X		X				Am	C		40	X				10	F													
9	X				8					41	X	X	X		1		Am	C		41	X	X	X		1		Am												
10			X						N	42		X						C		42	X						F	p											
11			X						N p	43	X	X			1		Am	C		43	X	X			1		Am	F											
12	X	X			1			Am	N	44	X	X					Am	C		44	X	X	X		2		Am	F											
13	X				2				N	45	X							C p		45	X							F											
14	X				3				N	46		X						C		46		X						F											
15		X							N	47		X						C		47		X						F											
16	X	X	X		1			Am	N	48	X		X				Am	C		48	X		X				Am	F											
17		X	X					Am	N	49		X			1			C		49	X		X				Am	F											
18	X		X		1			Am	N	50	X				2			C		50	X		X					F											
19			X						N p	51	X	X			3		Am	C		51	X	X					Am	F											
20	X				1				N	52	X	X			4		Am	C		52	X	X						F											
21	X				2				N	53	X				5			C		53	X							F											
22		X	X					Am	N	54	X				6			C		54	X							F											
23	X				1				N	55	X				7			C		55	X							F											
24	X				2				N	56	X				8			C		56	X							F											
25	X				3				N	57	X				9			C		57	X							F											
26	X				4				N	58	X	X			10	F	Am		58	X	X							F											
27	X				5				N	59	X	X			1		Am		59	X	X							F											
28	X				6				N	60	X							F p		60	X							F											
29	X	X	X		7			Am	N	61	X							F p		61	X	X	X					F											
30	X		X		8			Am	N	62	X	X	X		1		Am	F		62	X	X	X					F											
31	X				9				N	63	X	X					Am	F		63	X	X						F											
32	X		X		10	C	Am			64			X		1			F		64			X					F											

**2.2 Hoja de Resultados**

<b>PUNTUACIONES</b>				
	<b>DIRECTA</b>	<b>típica</b>	<b>T</b>	<b>Centil</b>
Nº intentos aplicados	116			
Respuestas Correctas	89			
Errores Totales	27	94	46	34
% Errores	23	96	47	39
Respuestas perseverativas	14	96	47	39
% respuestas perseverativas	12	98	49	45
Errores perseverativos	13	96	47	39
% Errores perseverativos	11	98	49	45
Errores no perseverativos	14	92	45	30
% Errores no perseverativos	12	93	45	32
Respuestas de nivel conceptual	79			
% Respuestas de nivel conceptual	68	94	46	34

<b>OTRAS PUNTUACIONES</b>		
	Puntuacion Directa	Puntuacion Centil
Nº de categorías com...	6	> 16
Intentos para complet...	32	2-5
Fallos para mantener l...	1	> 16
Aprender a aprender	2.71	> 16

<b>PUNTUACIONES: APRENDER A APRENDER</b>				
NºCat	Intentos C...	Errores Ca...	%Errores	DifErrores
1	32	7	21.88	0.0
2	26	10	38.46	-16.59
3	19	5	26.32	12.15
4	12	2	16.67	9.65
5	15	2	13.33	3.33
6	12	1	8.33	5.0

Diferencia Media:	<b>2.71</b>
-------------------	-------------

**2.3 Interpretación**

Esta mujer completó las seis categorías en 116 intentos. Como se puede ver en las hojas anteriores, su puntuación en diferencia de porcentaje de errores era: 21,88%, 38,46%, 26,32%, 16,67%, 13,33%, 8,33% respectivamente. Estos datos proporcionan las siguientes puntuaciones de diferencia: -15,58; +12,14; +9,65; +3,34 y +5. La estabilidad de las puntuaciones de diferencia positivas, a partir de la primera calculada, sugiere que la persona examinada se hizo progresivamente más eficaz en la tarea de clasificación a medida que iba realizando las diversas etapas del test, este incremento de su eficacia se refleja en la puntuación directa de aprender a aprender, que es +2,71.

**CASO 3: Varón, Médico de 65 años, diestro.**

**3.1 Hoja de Respuestas**

N	C	F	N	O	n	c	a	cp	p
1		X	X					Am	
2	X				1				
3	X				2				
4	X	X			3		Am		
5	X	X			4		Am		
6	X	X			5		Am		
7	X		X		6		Am		
8	X				7				
9	X				8				
10	X				9				
11	X				10	C			
12	X	X			1		Am		
13	X						C	p	
14	X						C	p	
15		X			1		C	p	
16	X	X	X		2		Am	C	
17	X	X	X		3		Am	C	
18	X				4		C		
19	X	X			5		Am	C	
20	X				6		C		
21	X	X			7		Am	C	
22	X	X			8		Am	C	
23	X	X			9		Am	C	
24	X				10	F			
25	X						F	p	
26	X						F		
27	X	X	X		1		Am	F	
28	X	X	X		2		Am	F	
29	X	X	X		3		Am	F	
30	X						F	p	
31	X						F		
32	X						F	p	

N	C	F	N	O	n	c	a	cp	p
33	X	X					Am	F	p
34	X	X					Am	F	p
35	X	X					Am	F	p
36	X						F	p	
37	X	X					Am	F	
38	X		X		1		Am	F	
39	X						C		
40	X		X		1		Am	C	
41	X	X	X		2		Am	C	
42	X						C	p	
43	X						C	p	
44	X						C		
45	X						C		
46	X	X					Am	C	
47	X	X					Am	C	
48	X		X		1		Am	C	
49	X		X		2		Am	C	
50	X						C	p	
51	X	X	X		1		Am	C	
52	X	X					Am	C	
53	X						C		
54	X						C	p	
55	X						C	p	
56	X						C		
57	X						C		
58	X						C	p	
59	X						C	p	
60	X		X		1		Am	C	p
61	X		X		2		Am	C	p
62	X	X	X		3		Am	C	p
63	X	X					Am	C	p
64	X						C	p	

N	C	F	N	O	n	c	a	cp	p
1	X							C	p
2	X							C	p
3	X							F	
4	X							F	p
5	X	X					Am	F	p
6	X	X					Am	F	p
7	X							F	p
8	X							C	
9	X							C	p
10	X							C	p
11	X							C	p
12	X	X					Am	C	p
13	X							C	p
14	X							C	p
15	X							C	p
16	X	X	X		1		Am	C	p
17	X							C	p
18	X		X		1		Am	C	p
19	X	X					Am	C	p
20	X							C	p
21	X	X			1		Am	C	p
22	X							C	p
23	X							C	p
24	X							C	p
25	X							C	p
26	X							C	p
27	X							C	p
28	X	X			1		Am	C	
29	X	X	X		2		Am	C	
30	X							F	
31	X	X			1		Am	F	
32	X							F	p

N	C	F	N	O	n	c	a	cp	p
33	X	X					Am	F	p
34	X	X					Am	F	p
35	X	X					Am	F	p
36	X							F	p
37	X	X					Am	F	p
38	X		X					F	p
39	X							F	p
40	X							F	p
41	X	X	X		1		Am	F	p
42	X							F	p
43	X	X	X		1		Am	F	p
44	X							F	p
45	X							F	p
46	X	X					Am	F	p
47	X	X					Am	F	p
48	X							F	p
49	X							F	p
50	X							F	p
51	X	X	X		1		Am	F	p
52	X		X		2			F	
53	X		X		3			F	
54	X		X		4			F	
55	X		X		5			F	
56	X		X		6			F	
57	X		X		7			F	
58	X	X	X		8		Am	F	
59	X	X	X		9		Am	F	
60	X	X	X		10	Nº	Am		
61	X	X	X		1		Am		
62	X	X	X		2		Am		
63	X		X					N	p
64	X		X					N	p

**3.2 Hoja de Resultados**

PUNTUACIONES	DIRECTA típica	T	Centil
Nº intentos aplicados	128		
Respuestas Correctas	53		
Errores Totales	75	70	30 2
% Errores	58	71	31 3
Respuestas perseverativas	66	71	31 3
% respuestas perseverativas	51	71	31 3
Errores perseverativos	59	68	29 2
% Errores perseverativos	46	68	29 2
Errores no perseverativos	16	89	43 25
% Errores no perseverativos	12	91	44 27
Respuestas de nivel conceptual	39		
% Respuestas de nivel conceptual	30	75	33 5

**OTRAS PUNTUACIONES**

	Puntuacion Directa	Puntuacion Centil
Nº de categorías com...	3	>16
Intentos para complet...	11	>16
Fallos para mantener l...	0	>16
Aprender a aprender	-30.45	≤1

**PUNTUACIONES: APRENDER A APRENDER**

NºCat	Intentos C...	Errores Ca...	%Errores	DifErrores
1	11	1	9.09	0.0
2	13	2	15.38	-6.29
3	100	70	70.0	-54.62

Diferencia Media: **-30.45**

**3.3 Interpretación**

Inicialmente había sido examinado en un departamento de medicina interna para pacientes externos, donde se evaluó durante una semana. Su historia indicaba un cambio brusco de personalidad caracterizado por indiferencia, retraimiento social y excesiva fatiga. Durante el examen, el paciente aparecía medianamente deprimido y manifestaba que sentía una ligera debilidad en el lado derecho y una disminución de los reflejos en la

extremidad superior derecha. Por otro lado, el examen clínico neurológico había sido normal. A causa de la brusca del cambio de personalidad y de algunos ligeros signos neurológicos, el sujeto fue ingresado en el hospital para una posterior evaluación. La tomografía computerizada (TC) de la cabeza reveló un área bien definida de atenuación decreciente que implicaba tanto a la materia blanca como a la materia gris del lóbulo frontal derecho. Aunque la localización del daño era algo sorprendente en relación con los síntomas motores, los datos neurológicos resultaban coherentes con un infarto en la zona frontal derecha. El paciente fue sometido a una evaluación neuropsicológica para valorar si presentaba algún déficit cognitivo significativo, asociado con el accidente cerebro vascular, que impidiera su retorno al trabajo. Se le realizó también un amplio examen en el que se incluyó el WCST.

En general, el paciente respondió relativamente bien a la batería de pruebas neuropsicológicas. Sus CI Verbal y Manipulativo en el WAIS-R fueron superiores a la media; en memoria a corto plazo, con estímulos verbales y visuales, alcanzó los niveles superior y muy superior en la Escala de Memoria de Wechsler. Tuvo algunas dificultades en la categoría “aplazada” en el WMS-R, logrando una puntuación media general. Una evaluación posterior con el Test auditivo-verbal de Rey mostró un relativo aplastamiento en la curva en reconocimiento verbal (aunque normal). El paciente tuvo algunas dificultades respecto a reconocimiento de caras, aprendizaje no verbal y recuerdo. A pesar de todo, sus habilidades visoespaciales estaban intactas y no presentaba evidencia de dispraxia constructiva. En el aspecto motor, mostraba una rapidez motora normal en su lado derecho (dominante) y mínimamente deprimida en el lado izquierdo. Las aptitudes básicas de lenguaje y comunicación del paciente se situaban dentro de los límites normales, aunque su fluencia verbal e iniciación se situaban en la parte inferior de la zona media (centiles 25 a 29).

En contraste con los resultados expuestos, el paciente manifestaba dificultades significativas en aspectos de capacidad ejecutiva superior. Su resultado de 119 errores en el Halstead Category Test, le situaba en un nivel moderado de deterioro al compararlo con los sujetos de su misma edad, formación y sexo. Aunque realizó.

Esto se puede comprobar en la tabla de Heaton. Aunque realizó adecuadamente la Parte A del Trail Making Test, tuvo algunas dificultades para alternar dos series cognitivas en un orden específico, en la parte B del Test.

Estas dificultades en la capacidad cognitiva superior son bastante coherentes con su realización en el WCST. Como se puede ver en las hojas superiores, se le aplicaron 128 intentos del WCST y solamente pudo completar con éxito tres categorías. Si bien el número de categorías completa se sitúa en la parte más baja del límite de lo normal para un individuo con su edad y educación, (> centil 16), es un tanto inferior a lo que se puede esperar en el caso de un médico que ha sido un buen profesional de la medicina. Aunque necesitó solamente 11 intentos para completar la primera categoría y completó la segunda en los trece intentos siguientes, su puntuación Aprender a aprender (-36,45) es muy pobre (centil 1). El sujeto comenzó a perseverar respecto a

Forma (ver intento 25) y tuvo grandes dificultades para generar alternativas y propuestas para solucionar problemas. Esta dificultad se refleja en su alta puntuación (66) en Respuestas perseverativas, que se sitúa en el centil 3 al compararla con los resultados del grupo correspondiente a su edad y nivel educativo y el centil 2 al

compararla con la población general de adultos, según la tabla general. Coherentemente con su moderada deficiencia en la realización del HCT,

El nivel conceptual general del paciente en el WCST, con un puntuación típica igual a 75 en porcentaje de respuestas de nivel conceptual, se sitúa en un ratio de deficiencia de medio a moderado, considerando la tabla incluida en el visor correspondiente a: Porcentaje de adultos de la muestra normal y clínica, según la localización de la lesión en los distintos niveles de puntuaciones típicas.

Considerando estos datos, resulta claro que la capacidad ejecutiva superior del paciente estaba marcadamente deteriorada, en relación con sus otras capacidades. El sujeto mantenía un nivel medio-alto en habilidades intelectuales y era capaz de aprender y retener informaciones nuevas con un nivel medio, lo cual le situaba por encima de la media del grupo normativo en eficacia. Su lenguaje básico y sus aptitudes de comunicación parecían intactos y no tenía dificultades en tareas que requieren aspectos complejos de habilidad visoespacial. Sin embargo, su capacidad para el análisis lógico y para formar nuevos conceptos estaba deteriorada, y parecía tener dificultades en flexibilidad cognitiva. Su pensamiento parecía ser extremadamente perseverativo y al comenzar sin éxito un intento de aproximación a un nuevo problema o situación, podía continuar intentándolo de la misma forma, sin éxito, más allá del límite en que otros sujetos buscan soluciones alternativas. A causa de los altos niveles de capacidad, respecto a toma de decisiones y resolución de problemas, que se requieren en el ejercicio de la medicina, se recomendó que no volviese a la práctica médica en el futuro próximo. Sin embargo, como su ataque era relativamente reciente, podría producirse alguna recuperación espontánea de las habilidades afectadas en los 6 a 9 meses siguientes. Por ello se informó al paciente de que se le practicaría una nueva evaluación neuropsicológica nueve meses después para valorar la naturaleza y extensión de su recuperación y para ayudar a determinar si podría o no retornar a la práctica clínica en un periodo determinado.

**CASO 4:** varón de 33 años, diestro, con un nivel educativo correspondiente a 14 años de estudios.

**12.3 Hoja de Respuestas**

1	N	C	F	N	O	n	c	a	cp	p	33	N	C	F	N	O	n	c	a	cp	p	1	N	C	F	N	O	n	c	a	cp	p	33	N	C	F	N	O	n	c	a	cp	p		
2		X	X					Am			34			X							C		2	X							6				X								Am	F	
3		X						N			35	X	X			1	Am				C		3	X							8				X								Am	F	
4	X	X	X	1	Am						36	X				2					C		4	X	X	X	9	Am							X									F	
5			X					N	p		37	X	X			3	Am				C		5	X	X			10	C	Am					X	X							Am	F	
6			X					N	p		38	X				4					C		6	X	X			1	Am						X	X							F	p	
7	X	X	X	1	Am						39	X				5					C		7	X	X	X	1	Am							X	X							F		
8	X			2	N						40	X				6					C		8	X											X							C	p		
9	X			3	N						41	X	X	X		7	Am				C		9		X										X	X	X	1	Am				Am	F	
10	X			4	N						42	X				8					C		10		X										X								C		
11	X			5	N						43	X	X			9	Am				C		11	X											X								C	p	
12	X	X		6	Am	N					44	X				10	F						12	X	X			1	Am						X	X							Am	C	
13	X			7	N						45	X									F	p	13		X										X								C		
14	X			8	N						46	X	X				Am				F		14		X										X								C		
15	X			9	N						47	X	X				Am				F		15	X											X								C	p	
16	X	X	X	10	C	Am					48	X	X			1	Am				F		16	X	X	X	1	Am							X	X	X	1	Am				Am	C	
17	X							C	p		49	X	X			2	Am				F		17	X	X	X	2	Am							X	X	X	2	Am				Am	C	
18	X	X	X		Am	C	p				50	X	X			3					F		18	X	X	X		Am	C						X	X	X		Am				Am	C	
19	X	X		1	Am	C	p				51	X	X			4	Am				F		19	X	X										X								C		
20	X							C	p		52	X				5					F		20	X				1							X								C		
21	X							C	p		53	X				6					F		21	X	X			2	Am						X	X			Am				Am	C	
22	X							C	p		54	X				7					F		22	X	X			3	Am						X	X			Am				Am	C	
23	X							C	p		55	X				8					F		23	X	X			4	Am						X	X			Am				Am	C	
24	X							C	p		56	X				9					F		24	X				5							X								C		
25	X							C	p		57	X				10	Nº						25	X				6							X								C		
26	X							C	p		58	X	X				Am						26	X				7							X								C		
27	X							C	p		59	X	X				Am						27	X	X			8	Am						X	X			Am				Am	C	
28	X							C	p		60	X	X			1	Am						28	X	X			9	Am						X	X			Am				Am	C	
29	X	X	X	1	Am	C					61	X	X			2	Am						29	X	X	X	10	F	Am						X	X	X	10	F	Am				Am	C
30	X	X	X		Am	C					62	X	X	X		3	Am						30	X											X								F	p	
31	X	X	X	1	Am	C					63	X	X			4	Am						31	X	X	X	1	Am	F	p					X	X	X	1	Am	F	p			Am	F
32	X	X	X		Am	C					64	X				5							32	X											X								F	p	

**12.4 Hoja de Resultados**

<b>PUNTUACIONES</b>	<b>DIRECTA</b>	<b>típica</b>	<b>T</b>	<b>Centil</b>
Nº intentos aplicados	113			
Respuestas Correctas	69			
Errores Totales	44	87	41	19
% Errores	38	87	41	19
Respuestas perseverativas	22	89	43	23
% respuestas perseverativas	19	89	43	23
Errores perseverativos	20	89	43	23
% Errores perseverativos	17	89	43	23
Errores no perseverativos	24	82	38	12
% Errores no perseverativos	21	81	37	10
Respuestas de nivel conceptual	61			
% Respuestas de nivel conceptual	53	88	42	21

**OTRAS PUNTUACIONES**

	Puntuacion Directa	Puntuacion Centil
Nº de categorías com...	6	>16
Intentos para complet...	16	11-16
Fallos para mantener l...	0	2-5
Aprender a aprender	-2.75	>16

**PUNTUACIONES: APRENDER A APRENDER**

NºCat	Intentos C...	Errores Ca...	%Errores	DifErrores
1	16	5	31.25	0.0
2	28	15	53.57	-22.32
3	13	3	23.08	30.49
4	12	2	16.67	6.41
5	24	10	41.67	-25.0
6	20	9	45.0	-3.33

Diferencia Media: **-2.75**

**12.5 Interpretación**

Se trata de un varón de 33 años, diestro, con un nivel educativo correspondiente a 14 años de estudios. Este sujeto trabajaba como operador técnico informático en una importante empresa, donde había ingresado tres años antes. Su historial médico no reflejaba ninguna patología hasta seis meses antes, cuando empezó a experimentar ataques recurrentes. Se recomendó una exploración detallada en la que se incluía una evaluación neurológica. El EEG y la RM revelaron la presencia de una amplia lesión en la masa frontal. Por ello fue necesario practicarle una craneotomía para la resección parcial del tumor. La intervención fue seguida de una serie de radiaciones y el sujeto pudo volver a su trabajo. Algunos estudios neurológicos posteriores revelaron la presencia de un resto del tumor y una necrosis por radiación. Finalmente, perdió su trabajo y tuvo que someterse a una segunda intervención para eliminar el tumor residual. Comenzó a sufrir ataques aproximadamente dos años antes de la presente evaluación neuropsicológica, aunque la serie de estudios con RM que se le practicaron después de la segunda intervención indicaban que la situación de su tumor era, en ese momento, estable. Hasta que se le practicó la segunda intervención, el sujeto informaba de que experimentaba dificultades de coordinación en lado izquierdo de su cuerpo y también lagunas de memoria, visión borrosa y problemas de comprensión lectora. Teniendo en cuenta estas quejas neurocognitivas, se recomendó una evaluación neuropsicológica para establecer el nivel básico, sus puntos fuertes y sus limitaciones para futuras comparaciones. Se le aplicó el WCST como parte de la batería de evaluación neuropsicológica.

Los resultados de este paciente en el WAIS-R presentaban S-R, aparecía una discrepancia similar (27 puntos) entre los resultados conseguidos con material visual y con material verbal (puntuaciones típicas de 115 y 88 respectivamente). Su habilidad para retener ambos tipos de estímulos después de un tiempo establecido se situaba en un nivel medio (puntuación típica 97). Su capacidad para reconocer materiales verbales previamente aprendidos se situaba en el nivel medio-bajo y su habilidad para reconocer rostros, medida con el Warrington Recognition Memory Test se situaba en un nivel inferior. Su lenguaje básico y sus aptitudes de comunicación se mantenían intactos, aunque, en el campo verbal, la fluencia y la iniciación parecerían algo deterioradas (centil 5 a 9). No tuvo dificultades con las tareas visoconstructivas y se situó por encima de la media en habilidad viso-organizativa compleja. Respecto a los aspectos motores, su motricidad fina y su coordinación eran esencialmente normales en su lado derecho (dominante), pero en su lado izquierdo se situaban en un nivel de deterioro de moderado a intermedio.

En una prueba de búsqueda visual y habilidad para la secuenciación, el paciente logró un resultado normal. Del mismo modo, realizó bien una tarea más compleja de atención distribuida, que requería alternar dos elementos cognitivos de una forma específica. El sujeto alcanzó un nivel algo bajo en cuanto a establecimiento de la primera categoría en el WCST (Número de intentos para completar la primera categoría, puntuación centil 6-10) y mostraba un pensamiento medianamente perseverativo comparado con su grupo de referencia (respuestas perseverativas, centil 9). Por otro lado, este grado de perseveración no es particularmente raro en la población general de adultos (Respuestas perseverativas, centil 23, en tabla General). Así, al mismo tiempo que sus tendencias perseverativas podrían situarle en desventaja respecto a algunas demandas del trabajo, normales en su nivel educativo, este déficit se pondría de manifiesto en algunas rutinas específicas del trabajo y en su vida personal. Aunque la puntuación alcanzada en porcentaje de respuestas de nivel conceptual se situaba por debajo del centil 10, logró completar seis categorías en 113 intentos. Del mismo modo,

mantuvo una estrategia de resolución de problemas adecuada dentro de las diversas categorías ( Fallos den mantenimiento de la actitud, 0) y se hizo ligeramente más eficaz en la tarea de clasificación a medida que avanzaba la aplicación (Aprender a aprender, centil 16).

En resumen, los resultados de la evaluación neuropsicológica de este paciente revelaron un modelo significativo de la lateralización cognitiva deficiente, como se pone de manifiesto en las discrepancias entre los CI Verbal y Manipulativo, sus índices de memoria en el WMS-R y RMT y un déficit motor lateral que implicaba la mano izquierda. Aunque el paciente ha mostrado algunas dificultades en pensamiento perseverativo y su percepción cognitiva se ha reducido, en relación con otras habilidades neuropsicológicas, le resultó posible utilizar el “feedback” del examinador en cada intento para hacer cambios conceptuales y completar el número total de categorías en menos de 128 intentos, límite del WCST. A pesar de que el paciente mostraba claramente algún déficit cognitivo en la batería de pruebas neuropsicológicas, sus funciones ejecutivas de nivel superior estaban relativamente bien conservadas, teniendo en cuenta la naturaleza y gravedad de su lesión cerebral y los tratamientos seguidos.

## ***ANEXO IV***

### **CODIGO PROGRAMA**

---

---

**El código de programa completo se encuentra disponible en el archivo.  
AnexoIV.pdf incluido en el CD de proyecto.**

**No se ha incluido aquí porque es más extenso que el propio proyecto.**