

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS EN INFORMÁTICA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Implementación de una Biblioteca de Tests Psicotécnicos para realizarlos en Internet

Realizado por

ANTONIO JOSÉ CAMACHO FUENTES

Dirigido por

RICARDO CONEJO MUÑOZ

MARÍA VICTORIA BELMONTE MARTÍNEZ

Departamento

LENGUAJE Y CIENCIAS DE LA COMPUTACIÓN

UNIVERSIDAD DE MÁLAGA

MÁLAGA, DICIEMBRE DE 2002

UNIVERSIDAD DE MÁLAGA

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS EN
INFORMÁTICA**

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

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

Presidente Dº/Dª: _____

Secretario Dº/Dª: _____

Vocal Dº/Dª: _____

para juzgar el proyecto Fin de Carrera titulado:

**Implementación de una Biblioteca de Tests
Psicotécnicos para realizarlos en Internet**

Del alumno Dº: ANTONIO JOSÉ CAMACHO FUENTES

Dirigido por Dº: RICARDO CONEJO MUÑOZ Y MARÍA VICTORIA BELMONTE
MARTÍNEZ

ACORDÓ POR _____ OTORGAR

LA CALIFICACIÓN DE _____

Y PARA QUE CONSTE SE EXTIENDE FIRMADA POR LOS COMPONENTES DEL
TRIBUNAL, LA PRESENTE ACTA.

Málaga, a de Diciembre de 2002

EL PRESIDENTE

EL SECRETARIO

EL VOCAL

Fdo:

Fdo:

Fdo:

Agradecimientos.

Desearía dedicar el proyecto a esas personas que siempre han estado a mi lado durante mi formación, tanto intelectual como personal ya que con su entusiasmo y apoyo desmedido han hecho que este proyecto vea la luz y haciendo que se convierta en realidad:

A mi abuelo, él quería que fuera un hombre de provecho, espero que allí en el cielo donde seguro que estás te sientas orgulloso de mi.

A mis padres, María y Antonio, y a mi hermano Rafael, por apoyarme en los malos momentos y disfrutar conmigo de los buenos y porque siempre están ahí cuando se les necesita. Pero en especial te lo quiero dedicar a ti, hermano.

A Sandra, por ser como es y llenar mi vida a todos los niveles. También quisiera agradecerle por entender todo el tiempo que no he podido estar a su lado por la realización de este proyecto. Gracias por desear compartir la vida conmigo.

A dos familias que me brindaron y abrieron las puertas de su casa para terminar este proyecto. A mis suegros, Maruja y Manolo, y los padres de un gran amigo, Ana y Benito. Gracias por vuestra hospitalidad, nunca sabré como compensarles.

A mis amigos en general. Por permitirme compartir esos buenos momentos con ellos. No quisiera nombrar a nadie ya que seguro me olvidaría de alguien, sería injusto. Ellos saben quienes son y decirles también que los llevo siempre en el corazón. Gracias por todo.

A Ricardo Conejo Muñoz y María Victoria Belmonte Martínez, que me dieron la posibilidad de llevar a cabo la realización de este proyecto. En todo este tiempo me ha demostrado su constancia y buen hacer.

Y por último, pero no menos importante a mi familia en general, destacando a los chiquitines de la familia, a mi primo y ahijado, José Abel, y a mi sobrina, Lydia María, por alegrarme la vida.

ÍNDICE

1. INTRODUCCIÓN.....	1
1.1. OBJETIVOS.....	3
2. LENGUAJES DE PROGRAMACIÓN.....	5
2.1. ESTUDIO SOBRE EL LENGUAJE DE PROGRAMACIÓN UTILIZADO.....	5
2.2. EL LENGUAJE JAVA.....	6
2.2.1. Orígenes del lenguaje Java.....	6
2.2.2. Características de Java.....	7
2.3. El lenguaje HTML.....	10
2.4. Aplicaciones y Applets.....	11
3. LOS TESTS PSICOTÉCNICOS.....	19
3.1. INTRODUCCIÓN.....	19
3.1.1. Desmitificación de los test.....	20
3.1.2. Las criticas dirigidas a los tests.....	21
3.1.3. Los tests y la selección.....	22
3.1.4. Concepto de <i>Aptitud</i>	24
3.2. INTRODUCCIÓN A LOS DIFERENTES TIPO DE TESTS PSICOTÉCNICOS.....	26
3.2.1. Tests de aptitud al razonamiento para el soporte verbal.....	26
3.2.2. Tests de aptitud al razonamiento para el soporte no verbal.....	27
3.2.3. Tests de memoria.....	28
3.2.4. Tests de motricidad, destreza y habilidad.....	29
3.2.5. Tests de reacciones psicomotrices.....	30
3.2.6. Tests de aptitudes sensoriales.....	30
3.2.7. Tests de creatividad.....	31

3.2.8. Tests de aptitudes artísticas.....	31
3.2.9. Tests de conocimientos o de aptitudes particulares.....	32
4. EL SISTEMA SIETTE.....	33
4.1. DESCRIPCIÓN GENERAL.....	33
4.2. ARQUITECTURA DEL SISTEMA.....	36
4.2.1. La Base de Conocimientos.....	37
4.2.2. El generador de Tests.....	38
4.2.3. Editores de tests.....	40
4.3 INTEGRACIÓN DE LOS TESTS EN SIETTE.....	43
4.3.1 Pasos básicos para acceder al editor de tests.....	44
5. IMPLEMENTACIÓN DE LA BIBLIOTECA DE TEST.....	53
5.1. PROGRAMACIÓN JAVA EN LA IMPLEMENTACIÓN DE LOS TESTS.....	53
5.1.1. Clases que componen nuestra biblioteca.....	56
6. MANUAL PARA EL USO DE LA BIBLIOTECA DE TEST.....	69
6.1 INTRODUCCIÓN.....	69
6.2. FUNCIONAMIENTO GENERAL DE LA BIBLIOTECA.....	72
6.2.1. El Test de Dominós.....	72
6.2.2. Test Cuadrado de Letras.....	75
6.2.3. Test de las Figuras Iguales o B.G.3.....	76
6.2.4. Test de Razonamiento y Estructuración Espacial (B.43 de Bonnardel).....	77
6.2.5. Test Espacio-Perceptivo o Situación-1.....	78
6.2.6. Test de Aptitudes Mentales Primarias (PMA).....	79
6.2.7. Test de Resistencia a la Fatiga.....	82
6.2.8. Test de Atención y Percepción.....	83

7. MANUAL DE REFERENCIA.....	85
7.1. INSTALACIÓN.....	85
7.2. CONFIGURACIÓN.....	87
7.2.1. Configuración del test de Dominós.....	87
7.2.2. Configuración del test Cuadrado de Letras.....	92
7.2.3. Configuración del test de las Figuras Iguales o B.G.3.....	93
7.2.4. Configuración del test de Razonamiento y Estructuración Espacial o B.43.....	94
7.2.5. Configuración del test Espacio-Perceptivo o Situación-1.....	98
7.2.6. Configuración del test de Aptitudes Mentales Primarias o PMA.....	99
7.2.7. Configuración del test de Resistencia a la Fatiga.....	100
7.2.8. Configuración del test de Atención y Percepción.....	101
8. CONCLUSIÓN Y TRABAJOS FUTUROS.....	103
8.1. CONCLUSIONES.....	103
8.2. TRABAJOS FUTUROS.....	104
9. BIBLIOGRAFÍA.....	105

Capítulo 1.

Introducción.

En la última década, los sistemas de educación asistido por ordenador han hallado en Internet, y específicamente en la World Wide Web, un medio ideal para su divulgación. Como consecuencia de esto, actualmente son muchos los centros de enseñanza y universidades que introducen en sus páginas web al menos el soporte para la divulgación de sus contenidos y material educativo. La WWW ofrece muchas ventajas en comparación con otros medios más habituales de divulgación de programas: no precisa de instalaciones adicionales al usuario; se puede acceder al sistema desde cualquier parte sin obligación de transportar el software; los contenidos se distribuyen y mantienen actualizados de forma inmediata; proporciona la fabricación de sistemas distribuidos; existe la posibilidad de adquirir datos sobre la eficacia de la instrucción y por lo tanto optimizarla, etc.

Es incuestionable que una máquina nunca podrá alcanzar el nivel profesional de un tutor humano en muchos aspectos, pero no pueden pasarse por alto sus importantes cualidades (disponibilidad ilimitada en el tiempo, manejo personalizado por parte del alumno, interactividad constante en ambos sentidos...).

Por todo ello la creación de un sistema Web en el que podamos encontrar el material docente, prácticas, y demás contenido de una asignatura, es de gran provecho tanto para el alumno como para el profesor. Para su desarrollo es necesario buscar las características empleadas en todo ambiente telemático: estructuras de hipertexto, recursos de multimedia, posibilidad de interacción, facilidad de actualización y propagación de contenidos. Con esa tentativa se intenta facilitar un ambiente favorable a las interacciones más ricas en la fabricación de un conocimiento concreto.

En todo sistema Web educativo no hay que prescindir de uno de los campos más significativos en el tema de la educación y el aprendizaje, la evaluación. Sin duda alguna uno de los dispositivos de evaluación de mayor difusión, por su sencillez de corrección es la fabricación de tests. Por eso sería de gran utilidad la incorporación de un mecanismo de autoevaluación en el sistema Web, donde el alumno pueda autoevaluar los conocimientos que va alcanzando.

SIETTE es un sistema de evaluación mediante tests adaptativos, que ha sido diseñado para ser usado sobre la World Wide Web (WWW). Dispone tanto a profesores como a alumnos un marco flexible, permitiendo a los profesores sumar nuevos tests e ítems, y proporcionado a los alumnos un aula virtual en el que realizar dichos tests.

Éstos tests no sólo destacaran por ser tests adaptativos sobre WWW, sino que admitirán como cuestiones o ítems ciertas plantillas que el usuario podrá detallar y que aportarán la creación a final del test. A su vez, estos ítems y plantillas podrán incluir componentes multimedia, tales como: videos, ficheros de audio, animación y gráficos; ampliando y haciendo mucho más divertido el aprendizaje del alumno.

Por otro lado, la materia que va hacer tratada en este proyecto tiene una finalidad práctica habitualmente aplicada a la vida diaria y laboral. Cada día son más las exigencias de las empresas a la hora de seleccionar a las personas para ocupar cualquier puesto de trabajo, y cada vez esta labor de selección se efectúa con más precisión y minuciosidad. Hoy ya no basta con dar buena imagen, es necesario probar de una manera clara nuestras aptitudes y nuestra capacidad, independientemente de cuál sea el nivel, la dificultad o el grado de responsabilidad que implique al que aspiremos.

Uno de los sistemas más generalizados hoy día y que se extiende cada vez más, es el test psicológico. Por lo tanto, familiarizarse con este tipo de prueba y aprender el método para superarlas, es una condición necesaria para aspirar a cualquier puesto de trabajo.

1.1 Objetivos.

El objetivo del Proyecto es conseguir un Entorno de Aprendizaje, accesible desde la World Wide Web, para que los alumnos puedan aprender todo lo relacionado con los tests psicotécnicos, independientemente de su conocimiento inicial en la materia.

Para ello, se realizará una serie de aplicaciones que irán destinadas a la creación de diversos tests para realizarlos en la Web de forma educativa. En él se pretende que el usuario aprenda de una forma cómoda e intuitiva todo lo relacionado con los tests psicotécnicos. Estas aplicaciones estarán implementadas en applets de Java favoreciendo así la interactividad con el alumno. El objetivo será hacer que el aprendizaje de la materia sea lo más divertido posible.

Para que el usuario pueda autoevaluar los conocimientos que va adquiriendo, haremos uso del sistema SIETTE (Sistema Inteligente de Evaluación mediante Test para TeleEducación). SIETTE (<http://www.lcc.uma.es/SIETTE>) es una herramienta para la creación y realización de tests adaptativos informatizados (TAIs) a través de Internet. En SIETTE, tanto la decisión de finalizar el test como el criterio de selección de la siguiente pregunta que hay que mostrar al alumno en cada momento, están basados en el nivel de conocimiento del alumno estimado por el sistema en cada momento.

El sistema ofrecerá al usuario una herramienta de ayuda en el aprendizaje y entrenamiento en el desarrollo de los tests psicotécnicos. Para ello, el sistema mostrará al usuario una lista acerca de los tipos de tests psicotécnicos más representativos (de atención, percepción, estructuración espacial, ...) posteriormente, y basándose en la elección que haga el usuario del tipo de test, el sistema mostrará una batería de cuestiones del tipo seleccionado, así como una evaluación real, de acuerdo a las mediciones estándares de la psicometría, de los resultados obtenidos.

Debido a las anteriores características, y al hecho de que el sistema está desarrollado en Internet, esta herramienta puede resultar muy útil a aquellos usuarios que necesiten adquirir un entrenamiento en el desarrollo de este tipo de test.

Una vez definido el sistema instructor y el lenguaje utilizado, el siguiente paso es elegir la información que ofreceremos al usuario. Para ello se han implementados tests tan conocidos y

utilizados como el test de Dominós, Cuadrado de Letras, etc. Estos tipos de tests miden capacidades tales como la rapidez al hacer una tarea de forma precisa, la resistencia al desgaste o capacidad para realizar tareas monótonas, frecuentemente son utilizados en pruebas de aptitud al razonamiento tanto verbal como no verbal y también para la inteligencia general.

Con todo esto, conseguimos un conjunto de herramientas muy útil para las partes implicadas en todo proceso de enseñanza: el alumno y el profesor.

Capítulo 2.

Lenguajes de programación.

2.1 Estudio sobre el lenguaje de programación utilizado.

El proyecto se ha desarrollado en JAVA, ya que se basa en la creación de aplicaciones para ejecutarlas en Internet, es una tecnología apta o muy apropiada para el desarrollo de aplicaciones bajo este entorno y tiene una característica muy importante que es su independencia de la plataforma.

En este momento es importante solventar cual sería el lenguaje de programación más apropiado para la implementación de nuestra aplicación. Una mala elección del lenguaje puede hacernos perder muchas horas de trabajo, por no habernos planteado convenientemente cuales eran las necesidades de nuestra aplicación.

Hay varios aspectos que hay que tener en cuenta a la hora de la elección, en este caso estamos buscando la creación de una aplicación con una interfaz de usuario intuitiva y cómoda, para optimizar el rendimiento del usuario, así pues habrá que encontrar un lenguaje que contenga un completo soporte para la fabricación de interfaces gráficos.

Uno de los aspectos más importantes a tener en cuenta es en que plataforma se va a utilizar la aplicación. Debido a la naturaleza heterogénea de Internet, el generar un programa ejecutable en cualquier lenguaje de alto nivel, se convierte en una tarea problemática: no se puede saber a priori qué equipo, o, qué sistema operativo estará ejecutando el usuario que pretenda utilizar ese software, en definitiva, bajo qué plataforma se quiere ejecutar la aplicación. Bajo esta condición, habría que crear y compilar esa aplicación en todas las combinaciones de

plataforma y Sistemas Operativos que existan en cada momento y darle a los usuarios la opción de elegir su versión antes de ejecutar la aplicación. Como se puede comprobar esta opción no es muy viable en la realidad.

Por todo esto hemos usado un lenguaje innovador, dinámico y con una amplia librería, y sin duda una de las mejores características del lenguaje, independiente de la plataforma. Hemos seleccionado a JAVA como nuestro lenguaje de programación para la implementación de nuestra biblioteca de tests psicotécnicos, siendo la que mejor se adapta a todas nuestras necesidades.

2.2 El lenguaje JAVA

2.2.1 Orígenes del lenguaje Java.

Los orígenes de Java se remontan al año 1990, cuando un equipo de la compañía Sun Microsystems investigaba, bajo la dirección del ingeniero James Gosling, en el diseño y elaboración de software para pequeños dispositivos electrónicos de consumo.

En un primer momento se pensó en la utilización de lenguajes de programación como C o C++, pero para poder compilar un programa en estos lenguajes es preciso adaptarlo a las características de la plataforma en la que debe funcionar, esta situación constituía un gran inconveniente para las compañías dedicadas a la construcción de dispositivos electrónicos, pues cada pocas semanas aparecen en el mercado versiones más potentes y baratas de los chips utilizados, y por lo tanto, el software que se había diseñado para un chip determinado debía modificarse y adaptarse para explotar las características de los chips de reciente aparición.

Se hace patente la necesidad de introducir un nuevo lenguaje de programación, que permita desarrollar programas independientes del tipo de plataforma. Los dispositivos que se pretenden fabricar son calculadoras, relojes, equipos de música, cafeteras, etc..., que no tienen una gran capacidad computacional, por lo que el nuevo lenguaje debe ser capaz de generar programas pequeños y rápidos, además de ser fiables y robustos.

La primera versión de este nuevo lenguaje se denominó Oak (roble), pero más tarde Sun descubrió que este nombre estaba ya registrado, y lo tuvieron que cambiar, el nuevo nombre

fue Java (una de las versiones sobre el significado del nombre es que Java es un término popularmente empleado en California para designar café de buena calidad).

A comienzos de 1993 aparecieron nuevas herramientas gráficas para facilitar la comunicación y navegación por Internet, se concibió la idea de aplicar técnicas de documentos con enlaces de tipo hipertextual para facilitar la navegación por Internet, y se desarrolló el primer browser (navegador Web o visualizador) de lo que se comenzó a denominar World Wide Web. Esta herramienta era denominada Mosaic.

El equipo de James Gosling se planteó como objetivo la utilización de Java como lenguaje en el que escribir aplicaciones que pudiesen funcionar a través de Internet. Como resultado de su trabajo se desarrolló un nuevo navegador completamente escrito en Java, llamado HotJava. Este navegador permitía la integración de pequeñas aplicaciones en el interior de las páginas Web. El desarrollo de HotJava hizo patente que las características de Java se adaptan perfectamente a las peculiaridades de Internet.

A partir de su primera y sencilla versión Java ha ido creciendo progresiva y espectacularmente para pasar a ofrecer un potente y complejo lenguaje con el que se pueden abarcar una gran cantidad de campos.

2.2.2 Características de Java.

El entorno de programación Java es un proyecto que presenta una posible aproximación a lo que serán los lenguajes de programación de los próximos años. Sus objetivos se resumen en dotar de herramientas de programación adecuadas para desarrollar aplicaciones que se ejecuten en entornos heterogéneos, conectados a través de redes de comunicaciones. Estas aplicaciones deben ser eficientes, podrán crecer dinámicamente, en función de las necesidades, y serán operativas en una gran variedad de ordenadores.

Los principales resultados del lenguaje de programación Java son:

- Se ha desarrollado un nuevo lenguaje de programación, denominado **Java**, que utiliza desarrollo orientado a objetos. Su sintaxis está basada en C++ (para simplificar el aprendizaje), pero elimina la mayoría de los aspectos oscuros de este lenguaje: se elimina el preprocesador de C, los typedefs, las funciones, y los punteros (casi todo lo heredado de C).

- El programador de Java exclusivamente puede utilizar desarrollo orientado a objetos; no existen las funciones, salvo como métodos de acceso a una clase. Las variables siempre están incluidas dentro de clases, para favorecer la encapsulación del código.
- Sus librerías de objetos predefinidas (el equivalente a las librerías de funciones) proporcionan herramientas para las comunicaciones a través de la red, diseño de interfaces de usuario, ...
- El programador queda liberado del manejo de memoria, que tantos problemas producen en lenguajes como C. Los objetos obtienen un espacio de almacenamiento durante su creación, que es recuperado automáticamente cuando se destruyen.
- Las aplicaciones Java pueden tener varias líneas de ejecución (*threads*) concurrente. La multitarea por *threads* está implementada en el propio lenguaje.
- Se ha modificado el proceso tradicional de compilación. Los programas Java se compilan y traducen a un formato denominado *bytecodes*, un formato intermedio de representación de las aplicaciones, que es independiente de la arquitectura del ordenador y de su sistema operativo. Para ejecutar aplicaciones Java en cualquier sistema, es necesario disponer de un software que interprete los *bytecodes* (un *runtime* de Java); sin embargo, el código de las aplicaciones es siempre el mismo.
- Con Java se pueden desarrollar aplicaciones completas independientes (al estilo de los entornos de programación tradicionales), o bien *applets*, módulos que pueden ser insertados en otra aplicación (por ejemplo, en un cliente Web), para realizar una función concreta. En cualquier caso, se dispone de la capacidad de incorporar módulos (objetos) dinámicamente, a través de una red de comunicaciones, sin necesidad de repetir el proceso de enlazado de la aplicación global.

Los dos últimos puntos constituyen uno de los aspectos más importantes de Java. Cualquier sistema que disponga de un *runtime* de Java es capaz de ejecutar sus aplicaciones. Por tanto, la única parte dependiente de la máquina es este módulo, que se desarrolla y optimiza en

función de sus características. La posibilidad de incorporar dinámicamente objetos a una aplicación activa elimina muchos de los problemas derivados de actualizar software, ya que se puede disponer de un servidor central de aplicaciones, del que se recogen módulos a medida que se necesitan.

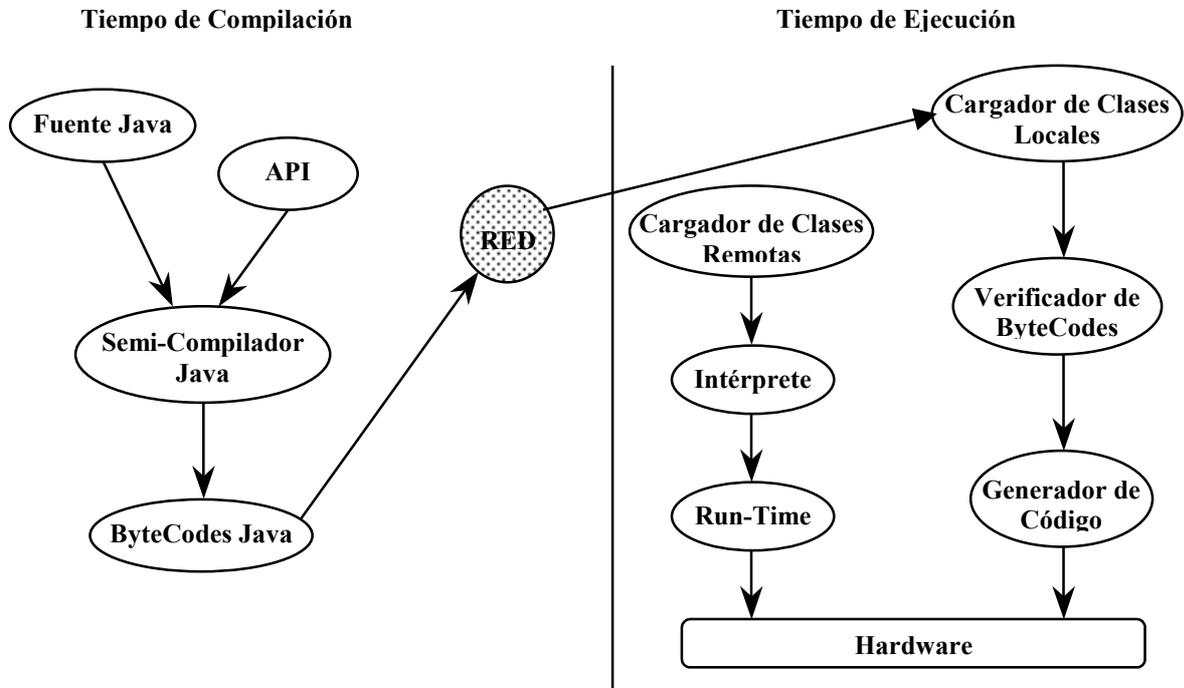


Figura 2.1 Esquema de compilación y ejecución en Java

2.3 El lenguaje HTML

HTML (*HyperText Markup Language*) es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con *enlaces* (*hyperlinks*) que conducen a otros documentos o fuentes de información relacionadas, y con *inserciones* multimedia (gráficos, sonido...).

La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Mosaic, o Netscape).

La mayoría de los efectos se especifican de la misma forma: rodeando el texto que se quiere marcar entre dos *etiquetas* o *directivas* (*tags*, en inglés), que definen el efecto o unidad lógica que se desea. Las etiquetas están formadas por determinados códigos metidos entre los signos < y >, y con la barra / cuando se trata de la segunda etiqueta de un efecto (la de cierre). Por ejemplo: <efecto> para abrir y </efecto> para cerrar.

A veces es necesario ofrecer datos adicionales en una directiva. Por ejemplo, cuando se define un *hiperenlace* hay que especificar su destino. Para ello se incluyen parámetros en la directiva inicial (la de apertura), de la siguiente forma: <efecto parametro1 parametro2...>. La directiva de cierre, caso de ser necesaria, queda como antes: </efecto>.

En la estructura básica de un documento HTML comienza con la etiqueta <html>, y termina con </html>. Dentro del documento (entre las etiquetas de principio y fin de html), hay dos zonas bien diferenciadas: el *encabezamiento*, delimitado por <head> y </head>, que sirve para definir diversos valores válidos en todo el documento; y el *cuerpo*, delimitado por <body> y </body>, donde reside la información del documento.

La única utilidad del encabezamiento en la que nos detendremos es la directiva <title>, que permite especificar el título de un documento HTML. Este título no forma parte del documento en sí: no aparece, por ejemplo, al principio del documento una vez que este se presenta con un programa adecuado, sino que suele servir como título de la ventana del programa que nos la muestra.

El cuerpo de un documento HTML contiene el texto que, con la presentación y los efectos que se decidan, se presentará al navegador. Dentro del cuerpo son aplicables el conjunto de los distintos efectos los cuales se especifican exclusivamente a través de directivas. Esto quiere decir que los espacios, tabulaciones y retornos de carro que se introduzcan en el fichero fuente no tienen ningún efecto a la hora de la presentación final del documento.

En resumen, la estructura básica de un documento HTML queda de la siguiente forma:

<code><HTML></code>	Indica el inicio del documento.
<code><HEAD></code>	Inicio de la cabecera.
<code><TITLE></code>	Inicio del título del documento.
<code></TITLE></code>	Final del título del documento.
<code></HEAD></code>	Final de la cabecera del documento.
<code><BODY></code>	Inicio del cuerpo del documento.
<code></BODY></code>	Final del cuerpo del documento.
<code></HTML></code>	Final del documento.

Figura 2.2 Estructura de un documento HTML

2.4 Aplicaciones y Applets.

El lugar más probable en el que puede encontrarse con JAVA, es sin lugar a dudas en la WWW, aunque en contra de la creencia (muy extendida) de que JAVA sólo sirve para esta función, sus uso no está limitado a la red. Como lenguaje de propósito general, es posible desarrollar grandes aplicaciones con él, como se puede hacer con cualquier otro lenguaje de alto nivel. Pero aparte de este uso general, también es posible desarrollar unas ‘mini’ aplicaciones interactivas que se ejecutan en el interior de las páginas Web: los *applets*.

Un applet es una aplicación escrita en lenguaje JAVA que se ejecuta dentro de un documento Web, es decir, la llamada que inicia la ejecución de ese programa, aparece en un fichero HTML dentro de unas determinadas etiquetas y es transportado desde el servidor al cliente vía HTTP igual que los demás elementos de la página en que está incluido.

Naturalmente, y como consecuencia de los diferentes entornos en que se ejecutan los applets y cualquier otra aplicación normal escrita en JAVA, existen marcadas diferencias entre unos y otros, entre las que cabe destacar las siguientes:

- El hecho de que un applet se ejecute dentro de una hoja o página HTML en un visor Web, indica claramente que serán programas con un entorno inherentemente gráfico, mientras que para las aplicaciones generales, el programador decidirá si se utiliza o no un entorno gráfico.
- También existen grandes diferencias entre los requisitos del sistema (recursos) que necesita un applet y los de una aplicación. Un applet necesitará además de los recursos que pueda requerir como aplicación que es, los que requiere el propio visor de páginas Web. Un applet es una ‘mini’ aplicación, y como tal requerirá pocos recursos.
- Tipo de entrada del entorno. La única entrada exterior a una aplicación, es una lista de parámetros en la lista de comandos, mientras que una applet va acompañado de mucha más información transmitida desde el visor de páginas: localización en la pantalla, tamaño dentro de la página en que se encuentra, parámetros propios que pueda necesitar el applet como aplicación, y sobre todo, información acerca de cuando tiene que activar o desactivar su ejecución.

Como se ha mencionado en la última diferencia, un applet necesita saber cuando debe iniciar su ejecución y, cuando debe pararse. Una aplicación simplemente necesita una llamada a su rutina *main* para inicial su ejecución y su detención vendrá dada por el propio programa. El applet, por lo contrario, dispone de todo un ciclo de vida desde que es cargado por el navegador, hasta que es destruido o liberado de la memoria.

En todo applet existe una serie de rutinas o procedimientos que deben ser llamados para controlar su ejecución y, que conforman ese ciclo de vida tan particular:

- Inicialmente, cuando el navegador carga la página Web en donde está el applet, también carga el código de éste mediante peticiones HTTP al servidor.

- Una vez que obtiene por primera vez este código, el *browser* llama a la rutina *init* del applet (inicialización).
- Es un buen lugar el procedimiento *init* para que el applet cargue todos los elementos (ficheros gráficos, sonidos, parámetros, etc.) que va a necesitar en su ejecución, ya que sólo se va a ejecutar una sola vez en todo el ciclo de vida.
- Una vez que el applet ha sido inicializado, necesita iniciar su ejecución como tal. El navegador va a realizar una llamada al procedimiento *start* del applet cada vez que vaya a presentarlo al usuario la página que se encuentre éste.
- Cuando el usuario decide cambiar la página que está mostrándole el navegador, éste realizará una llamada al procedimiento *stop* del applet para suspender temporalmente su ejecución.
- Por último, para completar el ciclo de vida de una applet, existe el procedimiento *destroy*, que va a ser llamado por el navegador cuando decida descargar el applet de la memoria (decide no utilizarlo más).
Aunque el método *destroy* no es obligatorio definirlo en cualquier implementación del applet (éste funcionará aunque no exista), es el procedimiento indicado para liberar los posibles recursos que esté utilizando el applet (contextos gráficos, cerrar conexiones o ficheros abiertos, etc.).

Además de todas estas rutinas que conforman el ciclo de vida normal de un applet existe otro procedimiento que siempre va a ser definido en el cuerpo de la aplicación:

Método *paint*. Este procedimiento va a ser llamado por el navegador cada vez que necesite redibujar el applet en la pantalla, por lo que va a ser uno de los métodos más importantes dentro del programa.

Aparte de ese ciclo de vida tan particular de los applets, también hay que tener en cuenta una serie de características o limitaciones en el momento de diseñar o implementar cada applet. El

hecho de ser una aplicación cargada y ejecutada a través del entorno de Internet siempre habrá que cumplir una serie de requisitos y recomendaciones:

- * En primer lugar, una recomendación en cuanto al tamaño de cada applet.

Desde el punto de vista del usuario final al que va destinado el código del applet, siempre es deseable o recomendable que los retrasos en el tiempo de carga a través de la red de los documentos HTML y los elementos que lo componen sea el mínimo posible.

Lógicamente, esta recomendación también es aplicable al código de un applet, y, es por esta razón, que casi siempre suelen ser aplicaciones muy pequeñas en lugar de programas complejos (de mayor tamaño) cuya funcionalidad está muy limitada. El uso más extendido de los applet que se encuentran en Internet, es el de presentar animaciones al usuario final o también, formularios para que introduzcan datos que le sirvan a otra aplicación mayor (bases de datos, aplicaciones en un servidor, etc.).

En lado contrario, el diseñador y programador de applet, debe tener siempre en consideración estas limitaciones recomendadas en cuanto al tamaño de sus creaciones, y deberá intentar por todos los medios posibles el depurar al máximo el código o utilizar algunos de los métodos disponibles para reducir el tamaño final del applet.

Algunas posibilidades que existen para conseguir esta disminución de tamaño son: utilizar las menos clases posibles (cada clase se carga en una conexión separada), o bien utilizar cualquiera de los métodos para unir todas las clases que componen un applet en un único archivo, con lo cual todo el código (clases) de un applet es descargado a través de una sola conexión HTTP desde el servidor, con la gran ventaja que supone en cuanto a velocidad de descarga.

También, la última generación de navegadores Web permiten la utilización de ficheros comprimidos (JAR, ZIP, CAB, ...) en los que, además de poder guardar todas las clases que componen el applet, se pueden almacenar todos los recursos que éstos necesitan para su correcta ejecución (imágenes, archivos de sonido, etc.).

- * Otra limitación impuesta a los applets, que quizás sea la más importante en cuanto a disminución de funcionalidad, es la referente a temas de seguridad.

Cuando se habla de Internet, todo lo concerniente a seguridad es de vital importancia para cualquier usuario, ya que siempre hay que evitar que cualquier descarga de software a través de la Red cause algún efecto no deseado en la máquina que lo recibe, y éstas consideraciones también son aplicables a los applets de JAVA como software que se obtiene de la Red. En principio, las restricciones impuestas a los applets de JAVA son bastantes importantes:

No se le permite a ningún applet, el manejar ficheros de la máquina del usuario que los ejecuta, de lo contrario, un applet 'malicioso' podría acceder a los archivos de configuración del sistema y modificarlos, lo cual, supondría un serio problema para ese usuario. Una vez que el código del applet es obtenido de la Red, es analizado por un Gestor de Seguridad para detectar posibles intentos de acceso a datos de los discos locales, bien para lectura, bien para escritura.

A pesar de ser un lenguaje orientado principalmente a la red, también a los applets se les prohíbe abrir conexiones con el fin de evitar que puedan enviar o leer datos importantes. En los esquemas más importantes de seguridad, sólo se deja abrir conexiones con el servidor de donde proviene el código del applet, para poder acceder a los recursos que necesita (ficheros de clases de applet, archivos gráficos y de sonido, etc.).

Aunque estas limitaciones a los applets son muy importantes, con el paso del tiempo se están creando nuevas políticas de seguridad basadas en sistemas de permisos (applets cifrados), por los cuales se pueden dar ciertos privilegios a los applets que implementan esta política para, por ejemplo, abrir conexiones con sistemas exteriores, o permitirles que accedan a ficheros locales.

Dado que los applets están mayormente destinados a ejecutarse en navegadores Web, había que preparar el lenguaje HTML para soportar Java, o mejor, los applets. El esquema de marcas de HTML, y la evolución del estándar marcado por Netscape hicieron fácil la adición

de una nueva marca que permitiera, una vez añadido el correspondiente código gestor en los navegadores, la ejecución de programas Java en ellos.

La sintaxis de las etiquetas <APPLET> y <PARAM> es la que se muestra a continuación:

```
<APPLET CODE=      WIDTH=      HEIGHT=      [CODEBASE=]      [ALT=]
[NAME=]  [ALIGN=]   [VSPACE=]   [HSPACE=]   [ARCHIVE=]>
<PARAM NAME=      VALUE= >
</APPLET>
```

Atributos obligatorios:

- CODE: Nombre de la clase principal (.class)
- WIDTH: Anchura inicial
- HEIGHT: Altura inicial

Atributos opcionales:

- CODEBASE: URL base del applet
- ALT: Texto alternativo
- NAME: Nombre de la instancia
- ALIGN: Justificación del applet
- VSPACE: Espaciado vertical
- HSPACE: Espaciado horizontal
- ARCHIVE: Archivo JAR y ZIP que contiene los .class

Figura 2.3 Estructura básica de la colocación de un Applet en una página HTML

Las marcas PARAM permiten pasar diversos parámetros desde el fichero HTML al programa JAVA del applet, de una forma análoga a la que se utiliza para pasar argumentos a *main()*.

Cada parámetro tiene un *nombre* y un *valor*. Ambos se dan en forma de *String*, aunque el valor sea numérico. El applet recupera estos parámetros y, si es necesario, convierte los *Strings* en valores numéricos. El valor de los parámetros se obtienen con el siguiente método de la clase *Applet*:

String *getParameter* (String name)

La conversión de **String** a los tipos primitivos se puede hacer con los métodos asociados a los wrappers que JAVA proporciona para dichos tipos fundamentales (Integer.parseInt(String), Double.valueOf(String), ...).

En los **nombres** de los parámetros no se distingue entre mayúsculas y minúsculas, pero sí en los **valores**, ya que serán interpretados por un programa JAVA, que sí distingue.

El programador del applet debería prever siempre unos **valores por defecto** para los parámetros del applet, para el caso de que en la página HTML que llama al applet no se definan.

El método *getParameterInfo()* devuelve una matriz de **Strings** (String[][]) con información sobre cada uno de los parámetros soportados por el applet: nombre, tipo, descripción, cada uno de ellos en un **String**. Este método debe ser redefinido por el programador del applet y utilizado por la persona que prepara la página HTML que llama al applet. En muchas ocasiones serán personas distintas, y ésta es una forma de que el programador del applet dé información al usuario.

Capítulo 3.

Los tests psicotécnicos.

La palabra *tests* es un término del inglés que significa prueba y proviene de raíz latina *testis* (testigo). Podemos definir un test psicológico, psicométrico, como un instrumento científico, que permite obtener una medida puntual, objetiva, fiable, válida y comparable, de las variables psicológicas personales, mediante el registro y análisis de las respuestas o manifestaciones conductuales individuales, a los requerimientos planteados por sus preguntas o items en condiciones estandarizadas.

3.1 Introducción.

Existe tal cantidad y variedad de test psicológicos, dependiendo de sus características, que su clasificación exhaustiva resultaría prolija y fuera de lugar. Nosotros particularmente nos centraremos en los tests utilizados en el mundo del trabajo. Estos a su vez se pueden clasificar en dos grupos: los de personalidad y los de aptitud o inteligencia o eficiencia. En un primer momento, la cuestión será la utilización de estos tests: sólo a través de la desmitificación será posible devolver a esta técnica sus justos valores. Tanto unos como otros requieren saber sobre qué criterios evaluar la inteligencia y la personalidad (psicometría).

En ambas categorías existe una gran diversidad de test, debido a que no existe un concepto único de inteligencia y de personalidad: todos los tests son incompletos, cuando no parciales. Para paliar esta falta de objetividad, generalmente se utilizan varios para valorar la inteligencia y la personalidad.

De los dos grandes grupos de tests psicotécnicos, nosotros trataremos mediante un sistema instructor llamado SIETTE, enseñar al alumno cómo realizar un test de inteligencia o de aptitudes dedicado para la inserción en empresa, así como dar la posibilidad de realizarlos y

conocer la evaluación, de tal manera que se podrá utilizar este proyecto para cualquier prueba de acceso a un puesto de trabajo, en la que requiera la realización de un test de este tipo.

Nuestro objetivo es suscitar e intentar una síntesis de los aspectos positivos y negativos de los tests psicotécnicos, dando la posibilidad de conocer estas técnicas a través de ejemplos concretos. Esto permitirá, no adiestrarse, sino familiarizarse con estos métodos. La presentación será lo menos técnica posible con el fin de resultar clara y fácilmente comprensible.

Los asesores de orientación profesional carecen con frecuencia de una formación psicológica, lo que les hace buscar la ayuda de los psicólogos que, a su vez, no conocen a fondo las exigencias técnicas y económicas de la gestión de los recursos humanos.

La investigación de la personalidad enfocada a la elección de una profesión o al éxito de una carrera no es en sí misma reprochable; sin embargo, se conocen muchos casos en los que por desgracia los tests psicológicos han servido más para eliminar que para seleccionar.

Por todos estos motivos, la forma en que se desarrolla la práctica de los tests psicológicos en el seno de la empresa debe ser mucho mejor comprendida por todos los que en ellos intervienen, y en especial, por las personas que deben someterse a ellos.

3.1.1 Desmitificación de los tests.

Es importante, antes de abordar la presentación de los diversos tests utilizados en el mundo laboral (implementados en este proyecto), señalar ciertas ideas adquiridas, desprovistas de todo fundamento.

Aunque el método de los tests está lejos de ser perfecto, y ciertas críticas son totalmente justificadas, vamos a intentar definir mejor aquellos puntos que resulten oscuros, así como los medios puestos en funcionamiento para satisfacer las carencias que padece esta técnica.

Lo que se intenta y desea con este proyecto es dar a conocer estos tests de la forma más objetiva posible, situándolos en su justo lugar.

3.1.2 Las críticas dirigidas a los tests.

A continuación abordaremos los problemas que surgen en la clasificación de los individuos así como las consecuencias de la vulgarización de estos métodos.

· *La clasificación de los individuos*

A muchas personas les atemoriza el aspecto selectivo de los tests. A nadie le gusta verse clasificado, incluido en una categoría, y por tanto, comparado a otros. Sin embargo es preciso reconocer que actualmente, aún más que hace unos años, la selección se ha convertido en una necesidad “económica” y “política”.

La utilización de los tests permite paliar los errores que se cometerían con los métodos en los que la elección del candidato es al azar; tomando solamente uno, si elige al primero que acude, o incluso si se elige a aquel que para tenga la mayor aptitud, sabiendo que éste no será más que uno entre los muchos métodos que se pueden utilizar para la selección. Es la combinación de todas las técnicas las que permiten más o menos fiable. El objetivo de la selección no es clasificar colocando a cada individuo la etiqueta de “bueno” o “malo”, sino intentar encontrar objetivamente el sujeto cuyas capacidades evaluadas por medio de los tests correspondan a las exigencias impuestas por el puesto.

· *Las consecuencias de la vulgarización de estos métodos*

Los tests son acusados a menudo de falta de seriedad y de un carácter pseudocientífico. La psicometría no puede recibir el apelativo de ciencia exacta como la física, la biología, etc. No es una ciencia, en el sentido de que no pretende descubrir leyes objetivas de ciertos fenómenos para difundir sus explicaciones. La psicometría es, sobre todo, una disciplina que se añade al conjunto de conocimiento del hombre, e intenta ponerlos en evidencia mediante el uso de técnicas apropiadas.

Pero la inteligencia y la personalidad son dos aspectos muy difíciles de definir y de estudiar; las técnicas desarrolladas no consiguen más que explorar una ínfima parte del individuo. Esta noción de conocimiento parcial queda por otra parte bien clara en el espíritu de los profesionales. La utilización del ordenador para la

interpretación de los datos no ha hecho más que reforzar para el gran público el carácter pseudocientífico de estos métodos, ayudando a su evaluación.

3.1.3 Los tests y la selección.

En este apartado trataremos de ver el papel que juegan los tests en el proceso de selección, las herramientas de las que disponen los psicólogos así como los diferentes métodos de elección.

El papel de los tests

El test es una prueba destinada a evaluar las *aptitudes* (es decir, las capacidades), y las *actitudes* (la manera de ser y reaccionar ante una situación o un trabajo) a fin de poder compararlas con las exigencias del empleo solicitado. Esta prueba deberá ser reproducida de forma estrictamente idéntica, con el fin de situar a los individuos en situaciones similares y poder así compararlos entre ellos.

Existen cuatro campos de utilización de los tests: clínico, militar, escolar y profesional.

Las razones que le darán de para qué sirven los tests serán siempre incompletas, y a veces falsas; pero nos guste o no, en alguna ocasión seremos sometidos a unas pruebas impuestas sin saber sus consecuencias. Por ello es lógico un cierta aprehensión a pasar esos exámenes, a ser juzgados; cuando no se conocen los resultados ni las condiciones. Con este proyecto nos proponemos ayudarle a superar todos estos temores; en él expondremos los tests más utilizados para la selección de personal, explicando el objetivo que se persigue con cada uno de ellos. De esta manera, resulta más fácil prepararse para la prueba, tanto moral y como psicológicamente. En la práctica siempre es más fácil superar una prueba cuando se ha entrenado y ejercitado para ella.

Aunque los tests psicológicos representan uno de los medios más fáciles para juzgar objetivamente la capacidad y competencia personal de un candidato no resulta tan sencilla las motivaciones del mismo.

Herramientas en manos de los psicólogos

Los tests psicológicos son unos instrumentos muy útiles para valorar los factores personales del éxito: inteligencia y carácter, a pesar de la carencia de objetividad científica. Para ser buenos instrumentos deben utilizarse cumpliendo las siguientes condiciones:

- No ser utilizados de forma aislada en la valoración de los factores personales.
- Ser empleados de forma adecuada por psicólogos competentes y experimentados.
- Matizar al máximo la interpretación y comunicación de sus resultados.

Los diferentes métodos de elección

El método de los tests es, por tanto, uno de los medios que permite solucionar la selección de un individuo. La combinación y la confrontación de diferentes datos que hacen posible llegar a una conclusión sobre tal o cual sujeto. Los distintos métodos tomados por separado no dan lugar a ninguna decisión definitiva, aportando cada uno de ellos elementos complementarios, o bien incidiendo sobre ciertos puntos específicos.

Además de los tests es necesario disponer de otros criterios para la evaluación de un individuo. Entre ellos podemos destacar:

- El dossier de candidatura (*currículum vitae*, una carta manuscrita y una foto).
- La entrevista.

3.1.4 Concepto de Aptitud.

El objetivo de este proyecto es presentar al usuario un amplio abanico de tests de aptitudes utilizados en situaciones de selección.

Actualmente no existe una definición universal de inteligencia. Se habla de eficiencia, de coeficientes intelectuales y también de aptitudes. Nosotros definiremos por “aptitud intelectual” como el producto de capacidades innatas y la influencia del medio. En otros términos, todos poseemos un potencial de base (una memoria, un modo de razonamiento, etc.), que puede ser desarrollado de forma diferente según el medio en el cual nos desenvuelve.

Distinguimos, pues, diversas formas de aptitudes intelectuales (entre las reconocidas), que permiten apreciar las capacidades de un individuo para satisfacer las exigencias impuestas ante un cierto puesto. Se pueden clasificarse de la manera siguiente:

- la memoria
- la atención
- la comprensión verbal
- la soltura verbal
- el razonamiento por inducción
- la soltura numérica
- las reacciones psicomotrices
- la destreza y la habilidad
- la organización del espacio

Algunas de las aptitudes específicas que siguen y que se miden por los tests, permiten comprender hasta qué punto la inteligencia es difícilmente reducible a un solo factor, aunque éste sea lo más general posible:

- ◆ Capacidad de deducción, o aptitud para razonar de lo general a lo particular.
- ◆ Capacidad de inducción, o aptitud para razonar de lo particular a lo general.
- ◆ Criterio, o aptitud para encontrar la solución de un problema entre otras soluciones propuestas.

- Articulación, o aptitud por la articulación rápida y distinta.
- Memoria, o capacidad de un sujeto para memorizar una serie de pares de palabras sin ninguna lógica entre ellas, y para reconocer rápidamente la segunda palabra de la pareja ante la simple presentación de la primera.
- La fluidez ideacional, o aptitud para producir un gran número de ideas sobre un sujeto impuesto.
- La fluidez verbal, o la aptitud para evocar palabras lo más rápidamente posible sin restricción de sentidos, pero imponiendo una contradicción llamada mecánica, como no dar más que palabras de tantas letras o empezando por tal prefijo, etc.
- La fluidez de expresión, o la aptitud para desarrollar una idea que se impone.
- La concepción verbal, o la aptitud que abarca los diferentes aspectos de la comunicación por la utilización del lenguaje.
- La rapidez de asociación, o aptitud para asociar lo más rápidamente posible un objeto y el nombre que lo simboliza.
- Aplicación, o aptitud para hacer una tarea sin cometer errores a pesar de la rapidez.

La lista no es exhaustiva, pero permite dar una idea de las diferentes aptitudes intelectuales que se pueden medir por los tests contruidos a este efecto.

3.2 Introducción a los diferentes tipos de tests psicotécnicos.

A continuación iremos detallando los distintos test que se utilizan para cada forma de aptitud, así como los criterios que se han tomado como referencia y los objetivos perseguidos.

3.2.1 Tests de aptitud al razonamiento para el soporte verbal.

Esta categoría de tests tiene por objeto medir la capacidad de un individuo para razonar sobre problemas que ponen en juego la utilización del lenguaje. Esta aptitud puede ser evaluada a partir de diversos criterios:

- la comprensión verbal;
- la fluidez verbal;
- el vocabulario;
- la asociación de ideas;
- la utilización de conocimientos lingüísticos (sintaxis, gramática).

La presentación de las pruebas permite apreciar estas aptitudes nos lleva a comprender mejor lo que se busca medir. La batería de tests que componen estas pruebas de aptitud al razonamiento para el soporte verbal es:

- Test de inteligencia general de R. Bonnardel (contenido verbal)
- El BV o Test de inteligencia de R. Bonnardel
- El BLS4 o Test de inteligencia general de R. Bonnardel
- El Test BV 16 de Bonnardel
- El BV 50
- El BV 51
- El Test BV 8 de Bonnardel
- BOR 18, Test de ortografía de R. Bonnardel
- BOR 19, Test de ortografía de R. Bonnardel
- Batería Factorial de Aptitud de J. Manzione

3.2.2 Tests de aptitud en el razonamiento para el soporte no verbal.

Esta categoría de tests tiene como función medir la capacidad de razonar sobre problemas de lógica. Podemos distinguir dos formas: los tests de papel-lápiz y los tests de resultado práctico. Llamamos aquí papel-lápiz a todas las pruebas por las cuales un individuo debe responder por escrito a preguntas que le son planteadas. Por otro lado los de resultado práctico son toda aquella prueba que necesita la utilización de un soporte material (cubos, puzzle, etc.) para llegar al resultado. Contrariamente al test lápiz-papel, éstos últimos nos permiten observar, por la manipulación, el comportamiento del individuo: habilidad manual, interés, perseverancia, tranquilidad, ingenio. Los criterios de referencia que nos permiten definir estas aptitudes son:

- la reflexión
- la inducción, la deducción
- la utilización de diversas operaciones (sumas, división, etc.)
- la utilización de ciertas leyes
- la organización y la representación espacial
- el método de trabajo (ordenado, estructurado, etc.)
- la comprensión de la tarea a efectuar
- la facilidad en el manejo de las formas geométricas
- el sentido de las proporciones, las dimensiones, las superficies
- el sentido de la observación
- la recopilación de informaciones y su tratamiento.

La batería de tests que componen esta prueba de aptitud en el razonamiento para el soporte no verbal es:

- El Test BV 53 de Bonnardel
- D 48, el test del dominó
- Los naipes de Pire
- Test de Razonamiento
- Test de las Series
- Test de Formas Idénticas de Thurstone
- Test de Resultados prácticos
- El B 43 de Bonnardel, o Test de Razonamiento y Estructuración Espacial

- El B 101 de Bonnardel o Test de Inteligencia Concreta
- El B 20 de Bonnardel o Test de Inteligencia Concreta
- El PM 38 o Matrices de Raven
- El PMA o Matriz Progresiva de Raven (nivel superior)
- Escala de Inteligencia Práctica de Goguelin
- Bloc Puzzle de P. Laborde
- Cuadrados de Letras
- Test de las Figuras Iguales o B.G.3
- Test espacio-perceptivo o Situación – 1

3.2.3 Tests de Memoria.

Existen cuantiosas clases de aptitudes intelectuales (atención, comprensión, verbal, razonamiento . . .) que, agrupadas, permiten estimar “la inteligencia”. Entre ellas forma parte la memoria, que podemos definir como la capacidad de recordar, de conservar las ideas adquiridas anteriormente.

El estudio de la memoria permite apreciar un cierto modo de funcionamiento del pensamiento. Este puede ser razonado o organizado, o por el contrario, sin orden ni estructura. La memoria puede ser evaluada a partir de diferentes puntos de referencia:

- concentración;
- observación;
- comprensión;
- exactitud del trabajo;
- método de trabajo.

La memoria está en relación directa con la atención. Por ello esta aptitud es frecuentemente explorada y demandada para los puestos que exigen gran vigilancia y responsabilidades de seguridad. La batería de tests que componen esta prueba de memoria es:

- Test de Retención Visual
- Test de Memoria de la Narración de Goguelin
- Test de Barrado de Signos de J. M. Lahy
- Test de Barrado de Cifras de S. Pacaud

- Test de Memoria Topográfica de S. Pacaud

3.2.4 Tests de motricidad, destreza y habilidad.

La motricidad, la destreza y la habilidad son formas de aptitud, ya que permiten un desarrollo profesional en ciertos empleos y, en particular, en los que requieren trabajos manuales. Estos factores no son tan evidentes de precisar como se podría creer. De una manera general, podemos asociarlos a la actividad gestual del individuo. El gesto, y en particular el efectuado por la mano, reviste una importancia esencial en la ejecución de ciertas tareas. Será entonces, tomado en cuenta como aptitud profesional. La cualidad de un trabajo podrá ser evaluada en función de la precisión, la finura, el cuidado y la rapidez con las que es llevado a cabo.

Comprendemos el interés que esto puede tener al nivel de productividad: un trabajo mal hecho lleva consigo inevitablemente una pérdida de tiempo, de dinero y de rentabilidad inesperada de consecuencias tediosas. Por eso, es de interés para la empresa escoger a los empleados más aptos para efectuar convenientemente su trabajo.

La batería de tests que compone esta prueba de motricidad, destreza y habilidad son:

- Tremómetro de R. Bize
- Test de Rapidez-Ojeada-Precisión de P. Goguelin y D. Guyot
- Test del Alambre de P. Goguelin
- Test de Destreza de R. Bonnardel
- Test de Señalización a Dos Velocidades
- Test de Disociación de los Movimientos
- Test Aptitud mecánica de Macquarrie
- Test de la Greca de R. Bonnardel
- Trazado
- El Tapping
- Señalización
- Test de copia
- Test de localización
- Ejercicio del Test de los Ladrillos
- Test de persecución

3.2.5 Tests de reacciones psicomotrices.

Esta categoría de tests tiene por objetivo medir el grado de atención de los individuos. La atención es la aptitud de concentrar el pensamiento en cualquier cosa. Esta capacidad, muy buscada en cuantiosos empleos, es una parte del potencial intelectual del hombre.

Estas pruebas son normalmente utilizadas para la selección de candidatos a ocupar puestos en los cuales la atención es la primera aptitud profesional. La batería de tests que componen esta prueba de reacciones psicomotrices es:

- Test de Reacciones Complejas de R. Bonnardel
- Test de atención difundida de J. M. Lahy y Guyot
- Test de Atención concentrada en reacciones Manuales de J. M. Lahy
- Tests de colación

3.2.6 Test de aptitudes sensoriales.

Son tests para medir la visión y la audición. No entran directamente en las situaciones de selección clásica, pero parece importante tener conocimiento de ellos, sin embargo, son utilizados para ciertas profesiones.

Se utilizan frecuentemente en la medicina para descubrir las deficiencias sensoriales. Éstas pueden ser la causa de un impedimento intelectual y de un comportamiento poco armonioso. También se utilizan para la selección de candidatos a puestos muy específicos, para los cuales una deficiencia, tanto visual como auditiva, puede poner en peligro la seguridad de las personas, o conllevar la mala realización de la tarea a ejecutar.

La batería de tests que componen esta prueba de aptitudes sensoriales es:

- Tacoscopio de proyección
- Audímetro
- Test de visión

3.2.7 Test de creatividad.

Estos tests procuran evaluar el proceso de creatividad de un individuo. Además, se ponen todos los medios para estimular la creatividad por técnicas específicas que buscan liberar la imaginación con el fin de ponerla al servicio de la eficacia técnica.

Una definición de Torrance permite resumir ciertos objetivos de estos tests cuando procuran evaluar el proceso de creatividad:

“ Proceso por el cual nos volvemos sensibles a los problemas, a las faltas, a las lagunas de conocimientos, a la ausencia de ciertos elementos, a las faltas de armonía . . . , por el cual identificamos la dificultad, por el cual buscamos soluciones, hacemos conjeturas o formulamos hipótesis a propósito de los déficit, por el cual probamos y volvemos a probar estas hipótesis y eventualmente las modificamos y volvemos a probar las modificaciones, y, finalmente, por el cual comunicamos los resultados.”

La batería de tests que componen esta prueba de creatividad es:

- Batería de test de Guilford
- Test de creatividad de Meunier

3.2.8 Test de aptitudes artísticas.

Ciertos psicólogos han estudiado el ámbito artístico con técnicas psicométricas. Los trabajos resultantes han permitido elaborar tanto tests de producción artística como de juicio y de cultura artística.

Por el contrario, los tests que siguen a continuación van a permitirles establecer una aproximación de los tests llamados de producción artística, que procuran, en su inmensa mayoría, dar una evaluación de las aptitudes musicales u otras:

- Medidas de talentos musicales de Carl E. Seashore
- Knauber Art Tests
- Tests de Aptitudes Fundamentales a las artes visuales de Lewerenz

3.2.9 Test de conocimientos o de aptitudes particulares.

Esta categoría de tests es un poco diferente a las otras, ya que profundiza sobre un conocimiento específico. Podemos constatar que todos los tests descritos anteriormente pueden ser realizados por cualquiera que sea el candidato, cualesquiera que sea su nivel de estudios o su especialidad. Según sus aptitudes, saldrá adelante más o menos bien, siendo capaz de comprender lo que se le pide y de contestar.

Ahora bien, los tests de conocimiento que vamos a presentar no pueden dirigirse a cualquier candidato. El conocimiento será aquí evaluado como la aptitud profesional para ejercer una cierta función. La batería de tests que componen esta prueba de conocimiento o de aptitudes particulares son:

- ✦ Test de Conocimientos Mecánicos de M. Alquier
- ✦ Test de Conocimiento de Electricidad
- ✦ Test de Conocimiento de Hidráulica
- ✦ Test de Conocimiento de Dibujo Industrial
- ✦ Test de Conocimiento de Matemáticas de Xydias
- ✦ Test de Conocimiento jurídico
- ✦ Test Técnico Práctico de P. Goguelin
- ✦ Test de Apreciación de Aptitud para trabajo de Taller de P. Goguelin
- ✦ Batería de Tests para Agente Administrativo
- ✦ Test para Empleos de Oficinas de H. Duchapt

Capítulo 4.

El Sistema SIETTE.

4.1 Descripción general.

SIETTE es un sistema de evaluación mediante tests adaptativos de contenido equilibrado basado en la teoría de respuesta al ítem, que ha sido diseñado para ser usado sobre la World Wide Web.

Utilizando un navegador estándar como interfaz gráfica de usuario, ofrece a los alumnos un aula virtual en el que realizar tests y permite a los profesores crear nuevos tests o añadir nuevas cuestiones a los tests ya creados. Además, dichos tests no sólo destacarán por ser tests adaptativos sobre WWW, sino que aceptarán como cuestiones ciertas plantillas que el usuario podrá definir y que facilitarán la creación final del test. A su vez, estas cuestiones y plantillas podrán incluir objetos multimedia, tales como: video-clips, ficheros de audio, animación y gráficos; mejorando así el aprendizaje del alumno y el modo de evaluar su conocimiento.

Para comprender mejor el sistema SIETTE, se exponen a continuación sus principales características:

- Reduce el coste de evaluación de los alumnos.
- Suministra a los usuarios dos tipos de herramientas. Por un lado, un conjunto de herramientas de edición de tests con las que se construye la base de conocimiento (especificaciones de tests, cuestiones, plantillas de cuestiones y objetos multimedia) del sistema. Por otro lado, el sistema de generación de tests, que construye el test más apropiado dependiendo del examinado. Además, se va a construir una nueva

herramienta gráfica que va a permitir a los profesores analizar los resultados de los alumnos que hayan realizado tests de sus asignaturas. Será posible acceder a estadísticas sobre el número de preguntas realizadas por los alumnos, número de preguntas que no han sido respondidas, etc.

- Confidencialidad de la información existente. El acceso tanto al editor de tests como al generador de tests, está restringido mediante palabras claves.
- Arquitectura distribuida: el sistema permite que haya múltiples alumnos realizando el mismo test, y a su vez permite que haya profesores editando las preguntas del test o insertando nuevas preguntas. La coherencia de la base de conocimiento (banco de ítems) se mantiene gracias a un mecanismo automático de validación de datos, que determina el momento más apropiado para llevar a cabo las actualizaciones en la base de datos.
- Adaptable al gusto del usuario. El formato y apariencia de las cuestiones son muy flexibles. Basta con conocer la sintaxis del lenguaje HTML para que éstas obtengan la apariencia que el creador del test considere oportuna.
- Uso de contenido multimedia. Las cuestiones del test pueden contener objetos multimedia (imágenes, vídeos, sonido, etc.) si el diseñador del test así lo desea. Además a través de la biblioteca de esquemas de preguntas, es posible proponer a los alumnos casi cualquier tipo de pregunta.
- Accesible a través de Internet. El acceso al sistema y sus resultados están disponibles a cualquier hora y desde cualquier lugar.
- Administración automática e inteligente del test. Se construyen tests adaptativos, en los cuales el número de preguntas necesarias para que el alumno obtenga su calificación es mucho menor que en los clásicos tests de papel y lápiz, en los que todos los alumnos, independientemente de su nivel de conocimiento, tienen que responder un número fijo de cuestiones. En SIETTE, a un alumno sólo se le plantearán las preguntas adecuadas para su nivel de conocimiento, evitando así la frustración o el aburrimiento de dicho alumno. De este modo, en el sistema se han

tenido en cuenta factores como: el grado de adivinanza ligado a cada pregunta del test, la dificultad de las preguntas, así como el factor de distracción. La precisión del sistema se incrementa al aumentar el número de cuestiones que el creador del test introduce en la base de conocimiento de la que hace uso el sistema. Se favorece así, la no repetición de preguntas en la misma sesión de test, entre alumnos del mismo nivel de conocimiento, etc.

- Aprendizaje de la dificultad de las preguntas. A través del procedimiento de aprendizaje nos aseguramos que las preguntas están correctamente calibradas, ya que el profesor cuando elabora el test sólo da una estimación personal de la dificultad real de la pregunta, y aunque en la mayoría de los casos estará cercana a la dificultad real de cada pregunta, no siempre sucederá así. A través de los datos empíricos obtenidos a partir de la ejecución de los tests por parte de los alumnos, el sistema de aprendizaje será capaz de asignar cuál es la verdadera dificultad de cada pregunta y a su vez de cada una de las respuestas a ésta.
- Calificación generada por SIETTE. El objetivo final del sistema SIETTE es estimar el nivel de conocimiento del alumno proponiendo a éste las preguntas más adecuadas para su nivel de conocimiento estimado en cada momento, y con el menor número de preguntas posibles. Por tanto, la calificación generada por el sistema está formada por: una distribución de un número establecido a priori de probabilidades y una estimación del nivel de conocimiento del alumno según esa distribución.
- SIETTE crea un historial del alumno de tal modo que si un alumno abandona un test de autoevaluación antes de que este finalice y posteriormente vuelve a realizar dicho test, el sistema le planteará preguntas según el nivel de conocimiento previamente estimado, sin necesidad de partir de nuevo del nivel de conocimiento inicial. En cambio, si el test es de evaluación, el alumno sólo podrá realizar el test una sola vez impidiéndose así que pueda modificar los resultados estimados en la sesión anterior.
- Refuerzo inmediato. Los resultados del test son calculados inmediatamente y están disponibles en formato gráfico. Dichos resultados junto con otros datos sobre el alumno son almacenados para posteriores análisis sobre el perfil de aprendizaje de dicho alumno. El alumno puede ver las respuestas correctas a las preguntas que se

les hace (ya que éstas no serán repetidas en la sesión actual de test) bien después de cada pregunta, o bien al finalizar el test.

4.2 Arquitectura del sistema.

La arquitectura del sistema SIETTE, contiene los principales componentes de un test adaptativo agrupados en seis módulos principales, su representación gráfica se muestra en la siguiente figura.

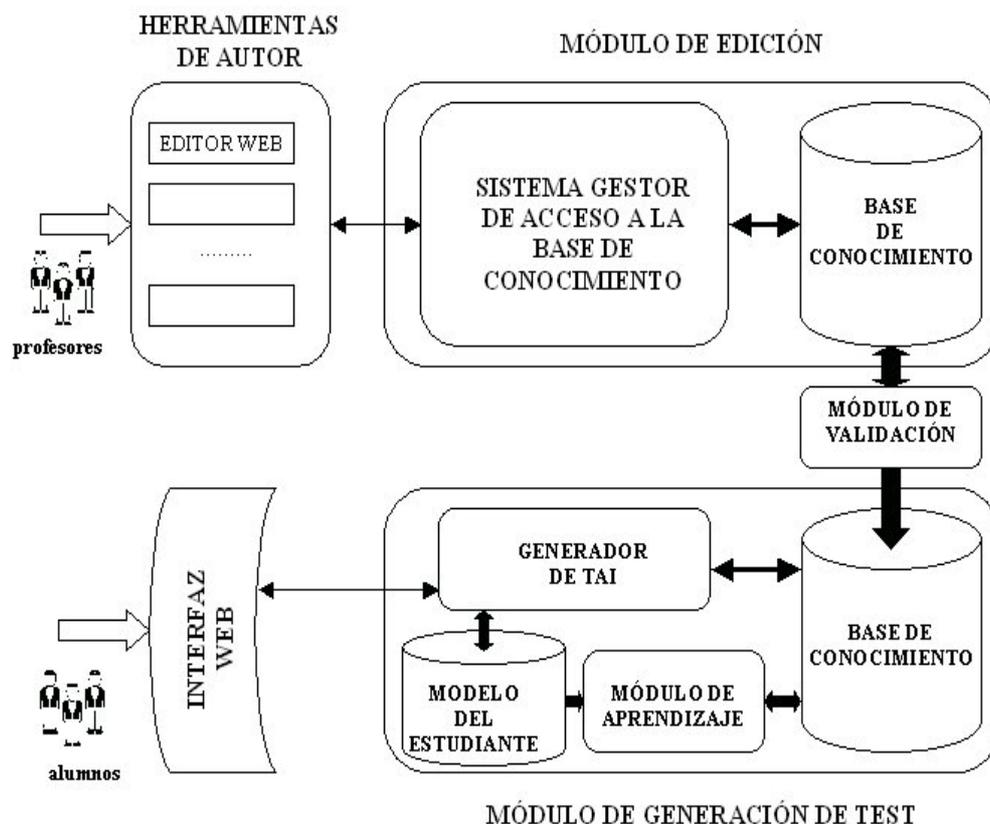


Figura 4.1. Arquitectura del sistema SIETTE.

La *base de conocimientos*: Está compuesta por el currículum o estructura del temario, las especificaciones de tests y el banco de ítems, que constituye la colección de posibles cuestiones a presentar en un test, todas ellas calibradas con una serie de parámetros.

El *generador de tests*: Es el módulo principal del sistema SIETTE. Es el encargado de seleccionar las preguntas a plantear al alumno, según las especificaciones del test y del modelo temporal del alumno.

El *módulo de edición*: Permite a los profesores acceder a la base de conocimientos, donde podrán añadir nuevas preguntas, respuestas, conceptos, tests, etc. Actualmente hay dos editores: un editor Web que la modifica de forma on-line.

Modelos temporales de los alumnos: Es creado y actualizado por SIETTE por cada estudiante que hace un test. Básicamente, el modelo de un estudiante consiste en una estimación de su nivel de conocimiento.

El *módulo de validación y activación de tests*: Las modificaciones realizadas en la base de conocimiento a través del editor de tests, no son inmediatamente actualizadas. Si el proceso de actualización fuera realizado en tiempo real, el sistema SIETTE podría tener problemas de inconsistencia. Por ello, necesitamos de un mecanismo de validación y activación de tests. De esta forma, los profesores hacen sus modificaciones en una base de conocimiento distinta de la base de conocimiento del generador de tests. Un proceso interno bloquea los elementos modificados y los actualiza en la base de conocimiento del generador de tests.

El *módulo de aprendizaje*: Se encarga de realizar una calibración de los parámetros de los ítems a partir de la información obtenida de forma empírica tras las sucesivas ejecuciones de los tests por parte de los estudiantes. Esta calibración es necesaria ya que los valores asociados a los ítems que introducen los profesores, sólo son aproximaciones iniciales.

Además de los componentes que han sido descritos anteriormente, SIETTE dispone de un simulador que permite el estudio empírico de las variables estadísticas que se manejan, así como de la influencia de los parámetros de los ítems, e incluye mecanismos para el aprendizaje automático de estos parámetros.

4.2.1 La Base de Conocimientos.

SIETTE es un sistema genérico para evaluación. Está estructurado por materias o asignaturas independientes. Cada una de estas asignaturas tiene una base de conocimientos distinta. La base de conocimientos de SIETTE está formada por tres tipos de objetos: Los *conceptos*, que son los temas o elementos en los que se descompone la asignatura. Están estructurados jerárquicamente, formando un *currículum*. Los *tests*, que representan las sesiones de evaluación. Cada *test* está formado por un conjunto de *ítems* o cuestiones. Los *ítems* están

asociados a uno o varios conceptos o temas. La definición de los tests se hace en función de los temas sobre los que se desea evaluar.

4.2.2 El generador de Tests.

Una vez que el profesor ha elaborado un test, los alumnos pueden acceder a él a través de una interfaz Web denominada *aula de tests*. Las preguntas se generan dinámicamente y de forma individualizada para cada alumno según la materia y el test que haya elegido.

El primer paso que tiene que dar el alumno es identificarse dentro del sistema. SIETTE permite el acceso tanto de forma identificada como anónima. Para la primera opción, el usuario deberá introducir un identificador personal y una contraseña. En caso de que sea la primera vez que el usuario accede al sistema, deberá dar sus datos personales: nombre, apellidos, correo electrónico, así como seleccionar un identificador personal y una contraseña. Una vez terminado el proceso de identificación, se le muestran los tests disponibles para él. Cuando el usuario selecciona el test que quiere realizar, SIETTE le mostrará información sobre este test antes de empezar a formularle cuestiones. Si posteriormente quisiera volver a realizar otro o el mismo test que resolvió anteriormente, únicamente tendrá que introducir en el sistema su identificador y palabra de paso, puesto que ya consta como usuario registrado.

Para los usuarios registrados el sistema tiene en cuenta los tests realizados anteriormente, y es capaz de mantener el modelo temporal como punto de partida para una nueva evaluación. Durante el proceso de realización del test, el sistema mantiene un registro temporal de la evolución del alumno, que se tiene en cuenta en el proceso de selección del siguiente ítem; y un registro histórico de las respuestas a cada cuestión, que servirá como fuente de información para el aprendizaje automático de los parámetros de las preguntas.

Las dos figuras que se exponen a continuación, muestran un estado intermedio durante la realización de un test. Ambas corresponden a páginas generadas dinámicamente por el generador de tests de SIETTE para un test de razonamiento y estructuración espacial o B.43. Una vez que el alumno resuelva la pregunta, el generador indicará cuál es la solución correcta. En este caso, como se puede apreciar, el alumno ha respondido correctamente, el generador muestra la respuesta correcta con la marca en verde.

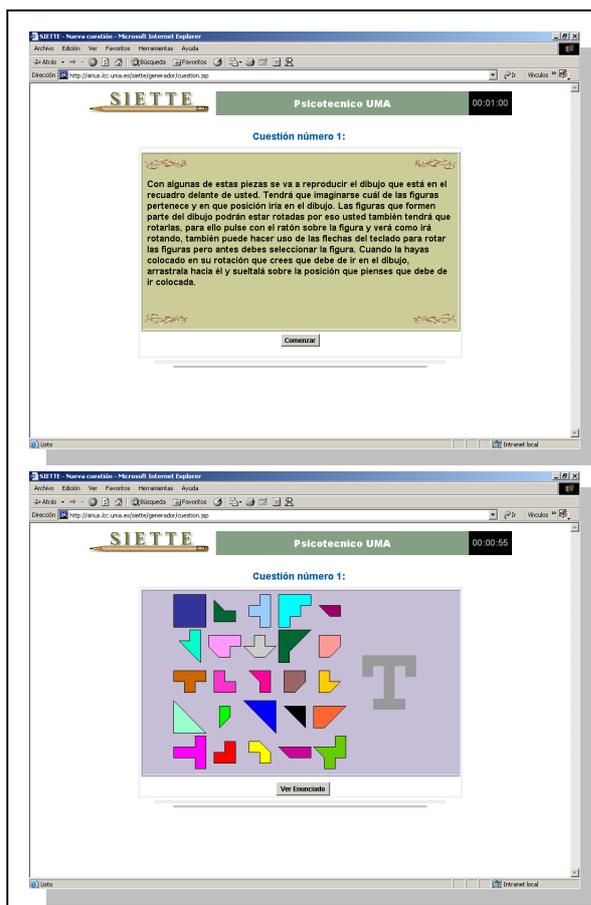


Figura 4.2. Ítem del generador de tests.

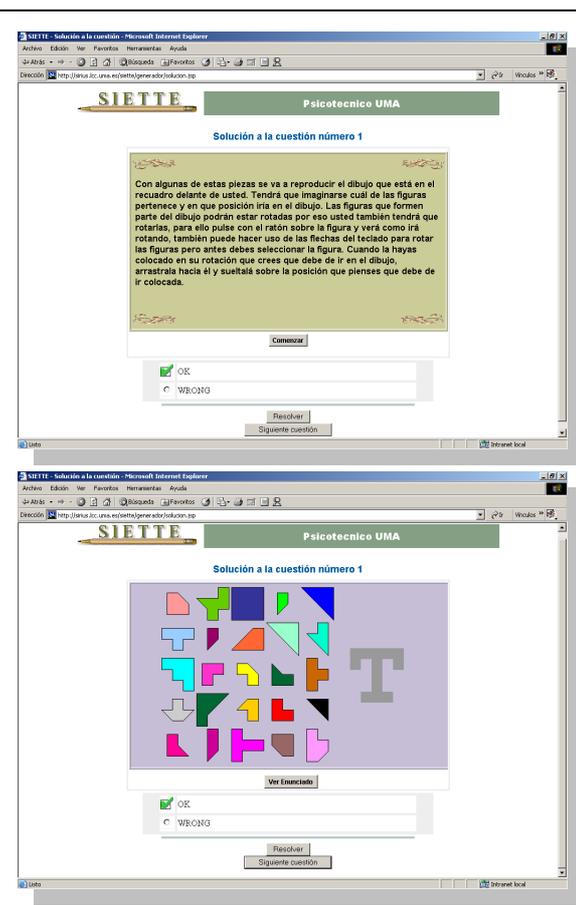


Figura 4.3. Corrección de un ítem.

Una vez finalizado el test, el sistema devolverá el nivel estimado de conocimiento del alumno, el número de preguntas que ha realizado, así como el número de preguntas que han sido respondidas correctamente. Alternativamente, si SIETTE se usa integrado en un sistema instructor inteligente, al terminar la evaluación esta información pasa al sistema tutor mediante la llamada a una URL distinta para cada nivel de conocimientos, o mediante una misma URL a la que se pasan estos datos mediante parámetros.

Además se mantiene un registro temporal de la evolución del alumno durante la sesión que se tiene en cuenta en el proceso de selección de preguntas y un registro histórico de las respuestas a cada pregunta que servirá como fuente de información para el aprendizaje automático de los parámetros de las preguntas.

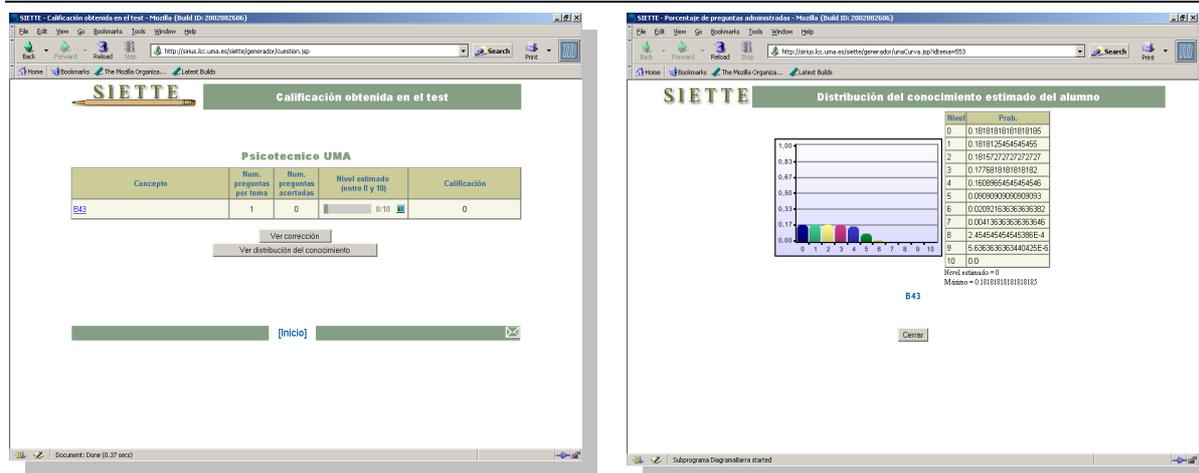


Figura 4.4. Evaluación final de un test.

4.2.3 Editor de tests.

El editor de tests es una herramienta de autor que permite a los profesores insertar cuestiones, y definir los tests que pueden realizar los alumnos. La información generada por el editor se almacena en una base de conocimiento temporal hasta que el módulo de validación y activación las transfiera a la base de conocimiento del generador de tests, estando a partir de este momento a disposición de los estudiantes.

Actualmente coexisten dos editores. Un editor a través de la web, que fue desarrollado en la primera versión de SIETTE y que ha ido adoptando varias transformaciones en las sucesivas versiones que se ha ido actualizando, y una nueva herramienta para edición y gestión de tests desarrollada en Java, que no requiere el uso del navegador.

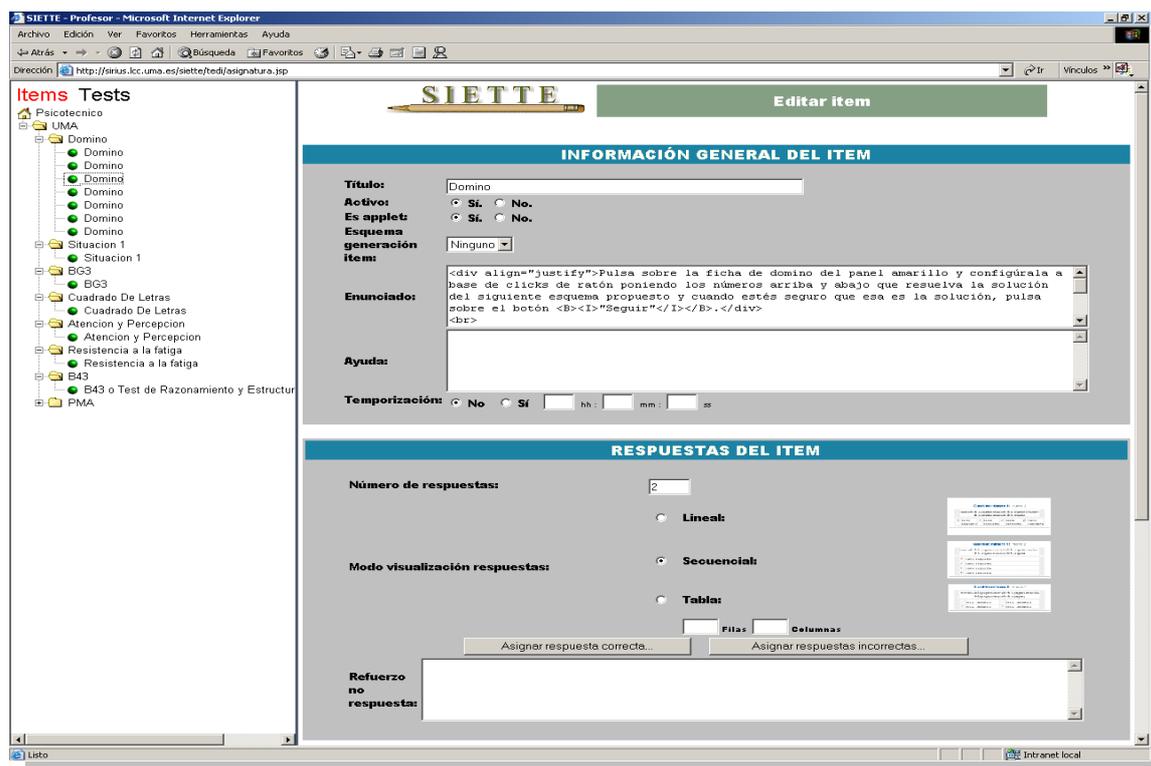


Figura 4.5. Interfaz Web para la edición de tests, temas y cuestiones.

La figura 4.5 nos muestra la interfaz del editor web. Para acceder a ella el profesor debe disponer de un identificador y una contraseña válida para la asignatura que desee modificar. Cada asignatura o materia se subdivide en temas en la forma que el profesor crea mas adecuada, usando para ello las secciones para añadir, modificar o eliminar temas. Los temas representan grandes bloques de la asignatura sobre los que se desea evaluar y no necesariamente conceptos concretos. Una vez definidos los temas, el profesor definirá los ítems o cuestiones que compondrán los tests. Para ello rellena una ficha en la que se incluye el enunciado, la respuesta correcta y una o varias alternativas de respuestas incorrectas. El sistema ha sido diseñado de forma que también almacene una posible ayuda, asociada al enunciado del ítem, en caso de que el alumno solicite más información para resolver la pregunta. También es posible añadir un refuerzo por cada una de las respuestas, de forma que si el alumno no ha acertado al resolver la pregunta, el sistema puede indicarle por qué su respuesta no es válida. Esta característica convierte a SIETTE en un sistema no sólo evaluador sino también en cierto modo instructor.

Los textos de las preguntas y las respuestas son trozos de código HTML, a los que se puede añadir JavaScript, Applets de JAVA, e incluso metacódigo en PHP, PERL, JSP, etc. El formato de las cuestiones admite cualquier objeto multimedia. Además del enunciado, las respuestas, la ayuda, y los refuerzos, el profesor debe proporcionar algunos datos sobre la cuestión, como por ejemplo, el número de alternativas incorrectas a mostrar, el factor de dificultad y debe asociar cada pregunta a uno o varios temas de entre los definidos anteriormente. Opcionalmente, puede proporcionar otros parámetros como el factor de adivinanza, el factor de discriminación, la distribución en pantalla, etc. Dado que el enunciado y las respuestas permiten el uso de metacódigo y Applets, cada cuestión introducida puede ser en realidad un esquema generador de cuestiones, lo que permite una gran variedad de preguntas. También es posible simular como quedará la pregunta al presentarla. Esto es especialmente útil en el caso de preguntas generativas.

Una vez definidas las preguntas de una materia, asociadas cada una de ellas a uno o varios temas, el profesor puede definir los tests que se van a realizar. Un test se compone de un conjunto de preguntas seleccionadas según diversos criterios tanto para la selección de preguntas como para la finalización del test. Por cuestiones prácticas se fija un número mínimo y máximo de preguntas para garantizar que en cualquier caso el test tiene un final, se alcance o no el criterio de finalización estadístico.

La nueva herramienta de edición mantiene las características ofrecidas por el anterior editor y añade otras nuevas, una de las características principales es que para llevar a cabo modificaciones en la estructura de las asignaturas, únicamente será necesario conectarse al servidor de la BC al inicio de la sesión de modificación, para recuperar toda la información sobre la asignatura; y al final de la sesión, para transferir las modificaciones realizadas, evitando así tener que estar usando una conexión permanente a Internet durante el proceso de modificación del test, como ocurría en el caso de editor Web.

La nueva herramienta de gestión de tests muestra la estructura del currículum de la asignatura en forma de árbol. Para realizar una modificación o nueva inserción de un tema o de una cuestión, el profesor sólo tiene que seleccionar el elemento con el botón de la derecha del ratón y elegir la opción que desee.

Tras especificar los temas, el profesor debe añadir las cuestiones que compondrán el test. En el editor a través de la web tanto el enunciado de las cuestiones como sus respuestas tenían que ser directamente insertados en código HTML. Este nuevo editor ofrece un conjunto de facilidades adicionales de edición del texto, ofreciendo ciertas ventajas al insertar código HTML y objetos multimedia, ya que dispone de un entorno mas elaborado.

Una vez que el profesor termina de modificar un test o cualquiera de los componentes de una asignatura, se generará un archivo en formato XML para mantener toda la información. Posteriormente, cuando desee transferir los datos a SIETTE, el editor enviará este archivo en formato XML que se traducirá en modificaciones en la base de conocimientos, mediante el módulo de validación.

En un futuro pueden desarrollarse herramientas de edición específicas para dominios concretos, que sean capaces de gestionar sus propias bibliotecas de ítems, a fin de facilitar la creación del banco de ítems complejos por parte de profesores no programadores.

4.3 Integración de los tests en SIETTE.

Como ya hemos comentado anteriormente, el sistema SIETTE está formado por dos grandes módulos: el sistema de edición de la base de conocimientos del sistema SIETTE (tests, temas y cuestiones) y el sistema de generación de tests. Esta distinción refleja de algún modo, quienes serán los usuarios de cada una de ellas. En esta sección, explicaremos cómo utilizar la herramienta de edición para insertar los tests que hemos implementados en este proyecto.

El editor nos permite crear un banco de preguntas, de temas y de tests relativos a la asignatura que vamos a tratar que es la de los psicotécnicos; de manera cómoda y flexible. Constituye así una herramienta necesaria para la creación de la base de conocimiento del sistema de tests. Su finalidad es por una parte la de guardar, vigilar y avalar la reutilización de los datos necesarios para la evaluación de cualquier alumno en la asignatura, y por otra parte, la de permitir al sistema de generación de tests disponer de gran número de cuestiones o ítems y de especificaciones de tests, para llevar a cabo una evaluación precisa del conocimiento del alumno.

El editor de tests es una herramienta que almacena los datos proporcionados por el profesor y los guarda en el servidor. Sólo si dichos datos cumplen los requisitos mínimos para la creación de tests válidos, serán guardados de la base de conocimiento del editor a la base de conocimiento del sistema de generación de tests. Por un test válido se entiende, que es aquel para el que existen un número mínimo de preguntas válidas para cada tema que compone el test, y que fijó el profesor. A su vez se entenderá por preguntas validas aquellas para las que existen una respuesta correcta y el número de respuesta incorrectas, establecido por el diseñador del test (profesor). Los tests válidos que cumplan los requisitos establecidos serán los ofrecidos por el sistema de generación de tests al resto de usuarios.

A continuación detallamos los pasos básicos para acceder al editor de test y que opciones de las que ofrece dicha herramienta deberemos rellenar para el funcionamiento de nuestros tests (applets).

4.3.1 Pasos básicos para acceder al editor de tests.

Para acceder al editor de tests, desde el navegador de internet nos conectamos a la URL dónde está SIETTE instalada (<http://www.lcc.uma.es/SIETTE>) y se podrá ver en el navegador, la página principal (véase figura 4.6). En dicha página pueden observarse dos enlaces, cada uno de ellos va dirigido a las dos aplicaciones con las que pueden interactuar los usuarios.

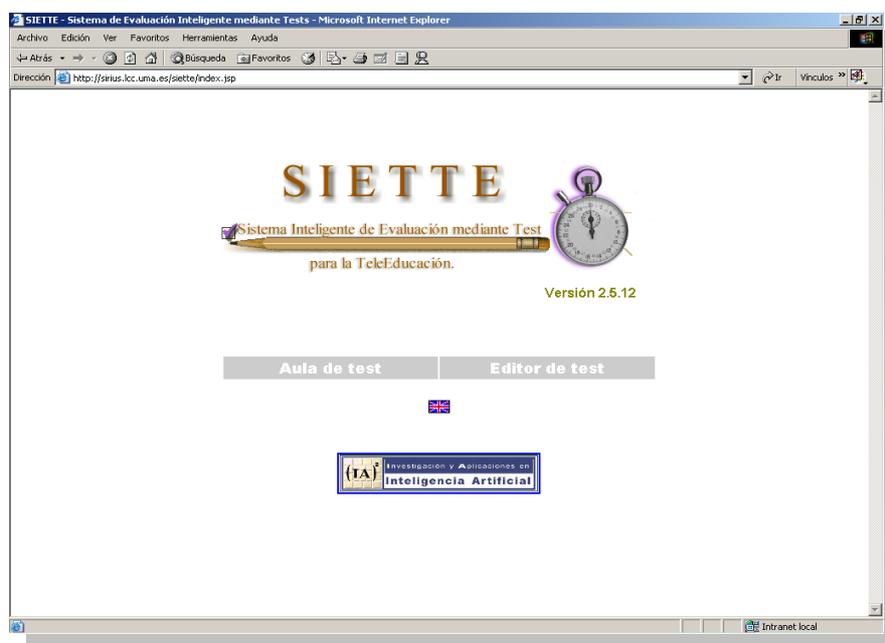


Figura 4.6. Página principal de SIETTE.

Como lo que se quiere es crear tests, tendremos que hacer clic con el ratón sobre el enlace “*Editor de test*”, y seguidamente aparecerá una nueva página que nos solicitará el código de identificación del profesor y una clave. Ambos datos son necesarios para poder acceder al editor de tests, y le serán suministrados por el administrador de SIETTE (véase figura 4.7).



Figura 4.7. Identificación del profesor.

Esta página asegura la confidencialidad de los datos suministrados y la seguridad de los mismos. Si como hemos dicho, el usuario dispone del código de identificación y de la clave de paso, el sistema mostrará una nueva página en la que se podrá crear o eliminar asignaturas y también se visualizarán las disponibles para ese profesor, si es que hay alguna (obsérvese figura 4.8).

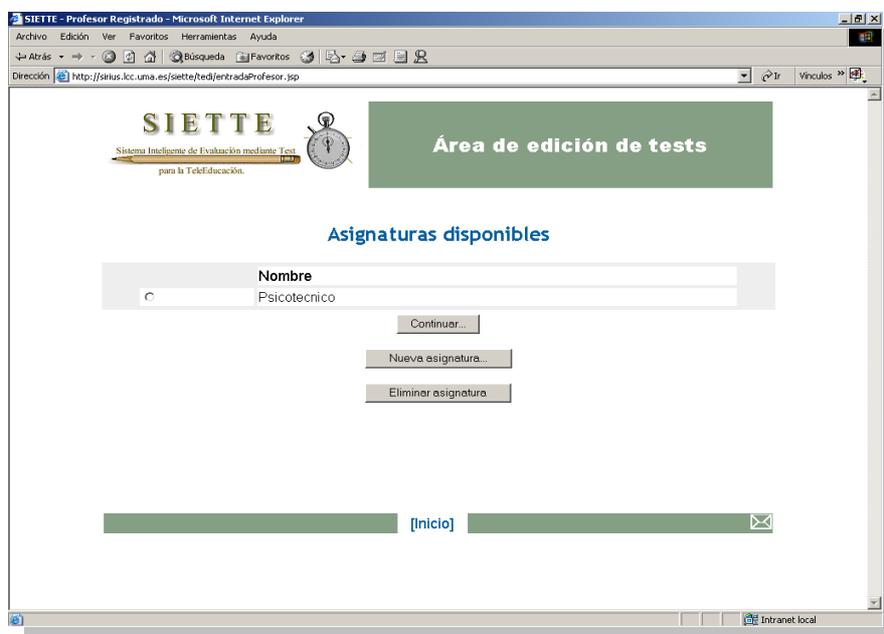


Figura 4.8. Creación, eliminación o selección de asignaturas.

Una vez que hayamos creado o seleccionado una asignatura, el sistema mostrará el menú que ofrece la herramienta edición para la creación, eliminación o modificación de temas, ítems o tests (véase figura 4.9).

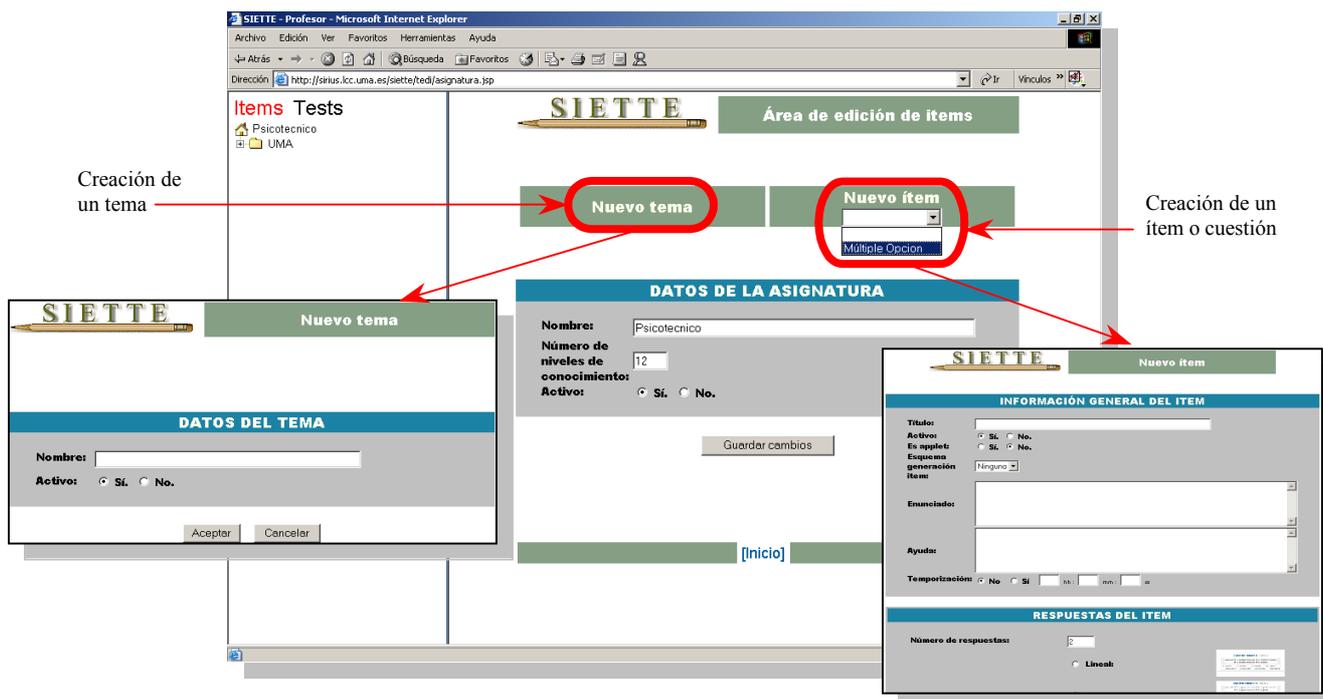


Figura 4.9. Editor de test.

Para la inserción de nuestros psicotécnicos hemos creado un tema por cada uno de ellos, y dentro hemos colocado las cuestiones o ítems para la creación de nuestros tests. A continuación vamos a mostrar un ejemplo de cómo hemos insertado un psicotécnico en el editor de test, todos se realizarán de manera análoga. Para este ejemplo hemos escogido el test de razonamiento y estructuración espacial o B.43 de Bonnardel.

Como dijimos antes, primero creamos un tema para este test que lo llamaremos “**B43**”, este tema se insertará en el árbol curricular y después incorporaremos el ítem dentro de él (véase figura 4.10).

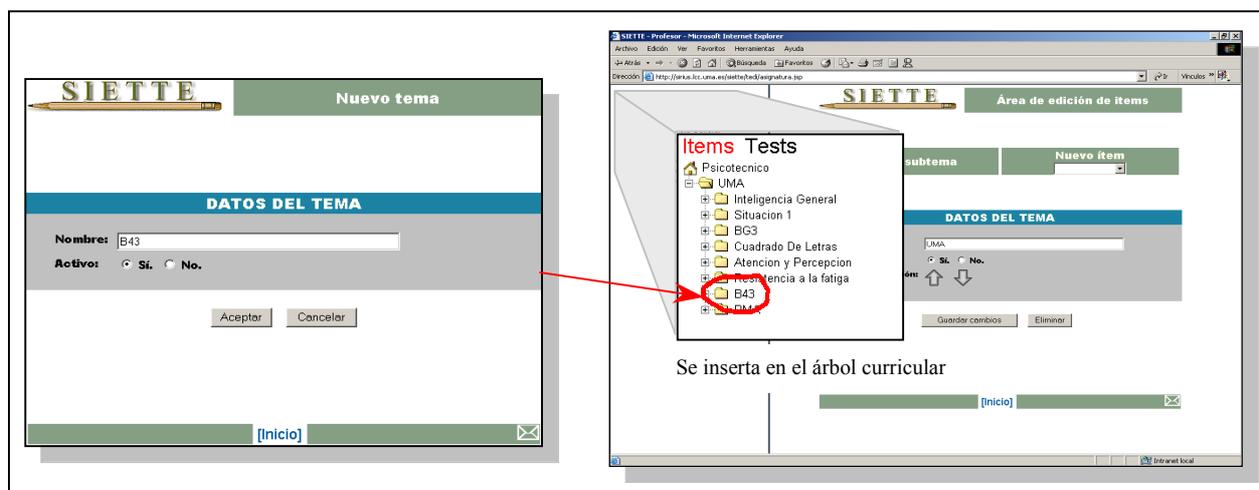


Figura 4.10. Creación de un nuevo tema.

Cuando se haya creado el tema, insertaremos la cuestión. Una cuestión en SIETTE estará definida por una serie de propiedades generales a partir de las cuales se podrá identificar y valorar dicha cuestión entre otras cuestiones existentes en la asignatura. Los datos aquí suministrados contribuyen a la elección de una pregunta frente a otras a la hora de plantear un test a un alumno. Una cuestión estará formada básicamente por el enunciado de la pregunta, la respuesta correcta y por las posibles respuestas incorrectas.

Para añadir una cuestión al banco de cuestiones de la asignatura que hemos creado sólo tiene que pulsar sobre el tema que estamos tratando que es “**B43**” en el árbol curricular y en el área de edición de ítems haremos clic sobre la lista desplegable de “**Nuevo ítem**” y seleccionamos la opción “**Múltiple opción**” (ver figura 4.9) ya que nuestras cuestiones podrán dar varias respuestas pero sólo una es la correcta. A continuación, se mostrará en su navegador una página solicitando los datos que definen a una cuestión para poderla insertar en la base de

datos. Estos campos son los datos de información general del ítem, los datos de las respuestas del ítem y los datos de los temas asociados al ítem.

- **Datos de información general al ítem.** Son todos aquellos datos que permiten identificar y valorar las preguntas del test (ver figura4.11).

Título: Nombre que identifica la cuestión entre aquellas que forman la misma asignatura. No es obligatorio dar nombre a la cuestión ya que en este caso es el sistema el que le asigna un nombre único. A nuestro ejemplo le pondremos el título del nombre del test psicotécnico al que hará referencia, “*B43 o Test de Razonamiento y Estructuración Espacial*”.

Activo: Si está disponible o no este ítem para utilizarlo en el test que esté insertado. Nosotros lo pondremos a “*Si*” para nuestro ejemplo.

Es applet: Este campo indica al sistema que la cuestión que estamos creando es o no una cuestión controlada por un applet. Como en nuestro caso lo que estamos creando son applets que generan plantillas del psicotécnico asociado, podremos esta casilla también a “*Si*”.

Esquema generación ítem: Este apartado nos da la posibilidad de que el esquema del enunciado sea o no estático. Para ello nos da a elegir en que lenguaje lo queremos escribir: en HTML o texto plano para los estáticos y PHP, JSP, PERL para los enunciados dinámicos. En esta casilla se elegirá la opción dependiendo del applet que vayamos a insertar, para el ítem que estamos insertando unos de los parámetros que podemos introducirle al applet es el número de la plantilla o puzzle que queremos que nos presente y ya que existen varias, decidimos que el sistema nos elija una al azar y nos la muestre. Por ello utilizaremos como esquema de generación del ítem, el lenguaje JSP.

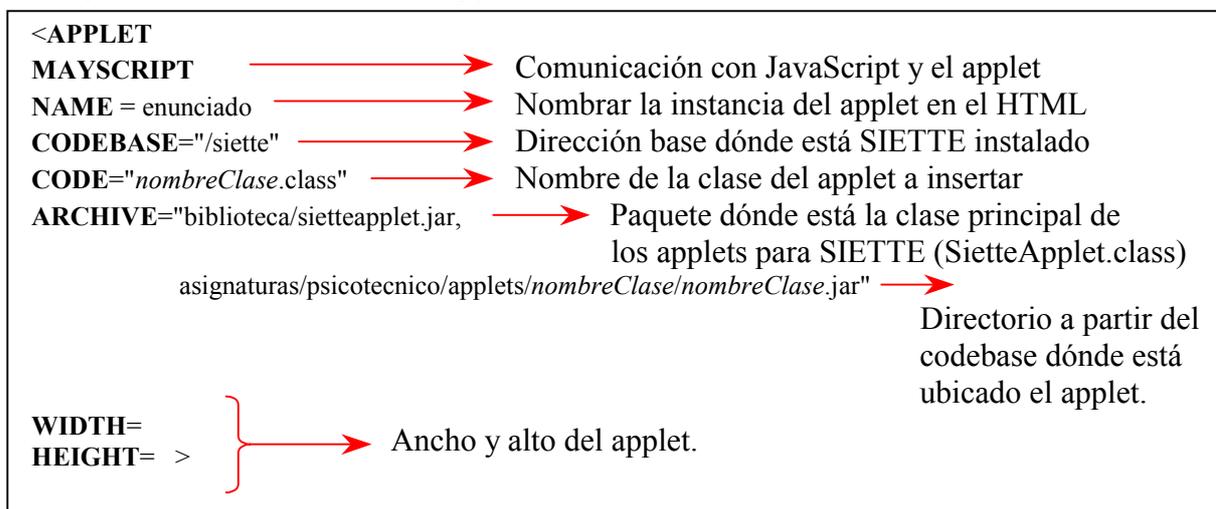
Enunciado: En este campo debe escribir el texto que aparecerá como enunciado de la pregunta en el test. El texto que se puede redactar es texto normal o bien texto HTML para dar el formato al enunciado que se mostrará al alumno. Por lo tanto, el formato y la visualización de la pregunta serán totalmente flexibles. Como nuestras cuestiones son controladas por applets, lo que deberemos de escribir es el código para insertar un applet

en una página HTML usando las etiquetas `<APPLET></APPLET>` para definir el applet y las etiquetas `<PARAM name= value= >` para pasar los parámetros al applet. Por implementación del sistema de evaluación SIETTE, la instancia del applet se deberá llamar “*enunciado*”, es decir, en el atributo de *name* de la etiqueta applet pondremos el string “*enunciado*” y además el atributo “*MAYSCRIPT*” debe estar presente en la etiqueta applet, de esta manera podremos comunicar el applet con el código JavaScript de la página que contenga a éste y llamar a algunos de sus métodos, como podría ser el método *Evaluación()* o *Resolver()*. En conclusión este atributo autoriza la utilización de código o información JavaScript por parte de un applet. Para el ejemplo que estamos tratando el enunciado que redactaríamos sería el siguiente:

```

<%@ page import="java.util.*" %>
<%
    Random random = new Random();
    int numPuzzle;
    numPuzzle = Math.abs( random.nextInt()%19 ) + 1;
%>
<CENTER>
<APPLET    MAYSCRIPT    NAME = enunciado    CODEBASE="/siette"    CODE="B43.B43.class"
    ARCHIVE="biblioteca/sietteapplet.jar,asignaturas/psicotecnico/applets/B43/B43.jar"    WIDTH=594    HEIGHT=380>
<PARAM NAME="Enunciado" VALUE="Con algunas de estas piezas se va a reproducir el dibujo que está en el recuadro delante
de usted. Tendrá que imaginarse cuál de las figuras pertenece y en que posición iría en el dibujo. Las figuras que formen parte del
dibujo podrán estar rotadas por eso usted también tendrá que rotarlas, para ello pulse con el ratón sobre la figura y verá como irá
rotando. Cuando la hayas colocado en su rotación que crees que debe de ir en el dibujo, arrástrala hacia él y suéltala sobre la
posición que pienses que debe de ir colocada.">
<PARAM NAME="AnchoEnunciado" VALUE="584">
<PARAM NAME="AltoEnunciado" VALUE="320">
<PARAM NAME="TamañoFuenteEnunciado" VALUE="17">
<PARAM NAME="NumeroDelPuzzle" VALUE="<%= numPuzzle %>" >
<PARAM NAME="FicheroFiguras"
    VALUE="http://sirius.lcc.uma.es/siette/asignaturas/psicotecnico/applets/B43/config/DefFiguras.b43">
<PARAM NAME="FicheroPlantillas"
    VALUE="http://sirius.lcc.uma.es/siette/asignaturas/psicotecnico/applets/B43/config/DefPlant.b43">
<PARAM NAME="GruposPiezas" VALUE="5">
<PARAM NAME="ColorEnunciado" VALUE="CCCC99">
<PARAM NAME="ColorFondo" VALUE="C6BED6">
<PARAM NAME="ColorPuzzle" VALUE="E7E3EE">
<PARAM NAME="MostrarBordePuzzle" VALUE="0">
</APPLET></CENTER>
    
```

Hay parte del enunciado y en concreto atributos, que es común a todos los ítems que componen nuestra biblioteca de applets.



Se puede observar que hemos incorporado código JSP en el enunciado, como se seleccionó en el campo de esquemas de generación, para generar un número aleatorio de entre las plantillas existentes y comunicárselo al applet para que cuando SIETTE muestre la solución genere la misma plantilla que el alumno intentó resolver.

Ayuda: Sirve para colocar aquí cualquier aclaración o explicación de la pregunta que facilite su solución por parte del alumno. Si se aporta información en este campo, el alumno verá debajo de la pregunta del test un botón de ayuda. Al pulsar dicho botón se mostrará el texto tal y como aquí se redacte.

Temporización: En este campo permite restringir el tiempo de respuestas en cada ítem. En caso de que sea temporizado habrá que introducir la duración de la presentación del ítem.

INFORMACIÓN GENERAL DEL ÍTEM

Título: B43 o Test de Razonamiento y Estructuración Espacial

Activo: Sí No.

Es applet: Sí No.

Esquema generación ítem: JSP

Enunciado:

```

NAME = enunciado
CODEBASE="/siette"
CODE="B43.class"
ARCHIVE="biblioteca/sietteapplet.jar, asignaturas/psicotecnico/applets/B43/B43.jar"
WIDTH=594
HEIGHT=380
    
```

Ayuda:

Temporización: No Sí 0 hh : 1 mm : 0 ss

Figura 4.11. Información general del ítem.

- **Datos de las respuestas del ítem.** Los datos relativos a las respuestas de las preguntas son: el número de respuestas, correctas e incorrectas, el modo en que se desean visualizar las preguntas, la respuesta correcta, las respuestas incorrectas y el refuerzo asociado a la no selección de respuesta (ver figura 4.12).

RESPUESTAS DEL ÍTEM

Número de respuestas: 2

Modo visualización respuestas: Lineal Secuencial Tabla

Refuerzo no respuesta:

Asignar respuesta correcta... Asignar respuestas incorrectas...

Figura 4.12. Modo de visualización de las respuestas de la pregunta dentro del test.

El número de respuestas por defecto es 2, pero se puede modificar dependiendo de las respuestas que devuelva el applet.

El modo de visualización de las respuestas de la pregunta dentro de un test puede ser los siguientes:

- **Lineal:** Se exponen las respuestas en una misma fila.

Este diagrama muestra un formulario con un campo de texto superior etiquetado como 'Enunciado de la pregunta'. Debajo de este campo hay un botón etiquetado como 'Ayuda'. En la parte inferior del formulario, se muestran dos opciones de respuesta, 'Respuesta1' y 'Respuesta2', cada una precedida por un botón de radio.

Figura 4.13. Modo de visualización de respuesta lineal.

- **Secuencial:** Se visualizan las respuestas cada una en una fila distinta.

Este diagrama muestra un formulario con un campo de texto superior etiquetado como 'Enunciado de la pregunta'. Debajo de este campo hay un botón etiquetado como 'Ayuda'. En la parte inferior del formulario, se muestran tres opciones de respuesta, 'Respuesta1', 'Respuesta2' y 'Respuesta3', cada una precedida por un botón de radio y dispuestas en líneas separadas.

Figura 4.14. Modo de visualización de respuesta secuencial.

- **Tabla:** Se muestran las respuestas en una tabla de n filas y m columnas, esta dimensión es propuesta por el profesor.

Este diagrama muestra un formulario con un campo de texto superior etiquetado como 'Enunciado de la pregunta'. Debajo de este campo hay un botón etiquetado como 'Ayuda'. En la parte inferior del formulario, se muestran cuatro opciones de respuesta, 'Respuesta1', 'Respuesta2', 'Respuesta3' y 'Respuesta4', dispuestas en una tabla de 2x2 con un botón de radio a la izquierda de cada celda.

Figura 4.15. Modo de visualización de respuesta en tabla.

Para los ítems de nuestros applets nos es indiferente como queremos que nos muestre las respuestas, se deja a elección del profesor el modo de visualización.

Cuando se haya decidido el modo de visualización pasamos a introducir la respuesta correcta y las incorrectas. En este apartado de deben suministrar el texto y alguna explicación sobre las respuestas a la pregunta (ver figura 4.16), esto último es opcional.

Al igual que en el campo de texto del enunciado de la pregunta, en el campo de la respuesta puede colocar texto plano, texto HTML para dar formato y texto con código

PHP, JSP o PERL si quiere crear plantillas de respuestas en lugar de dar las respuestas directamente. El texto que coloque en el refuerzo (aclaraciones, explicaciones, etc.) se visualizará cuando se muestre la solución del ítem y presentará el refuerzo de la respuesta correcta si el resultado de la pregunta ha sido acertado o por el contrario presentará el refuerzo correspondiente de la respuesta incorrecta.

El texto que se debe suministrar es el mismo que el que devuelve el applet del ítem que estemos creando, así pues si el applet para una respuesta correcta devuelve el texto “OK” en el campo de texto de la respuesta correcta deberemos poner “OK” tal y como lo devolvería el applet, y lo mismo ocurre con la respuesta incorrecta, si devuelve “WRONG” en el campo de texto correspondiente a la respuesta incorrecta colocaremos este texto y si existe más de una respuesta incorrecta se deberán poner cada una de ellas en el campo correspondiente a las respuestas incorrectas.

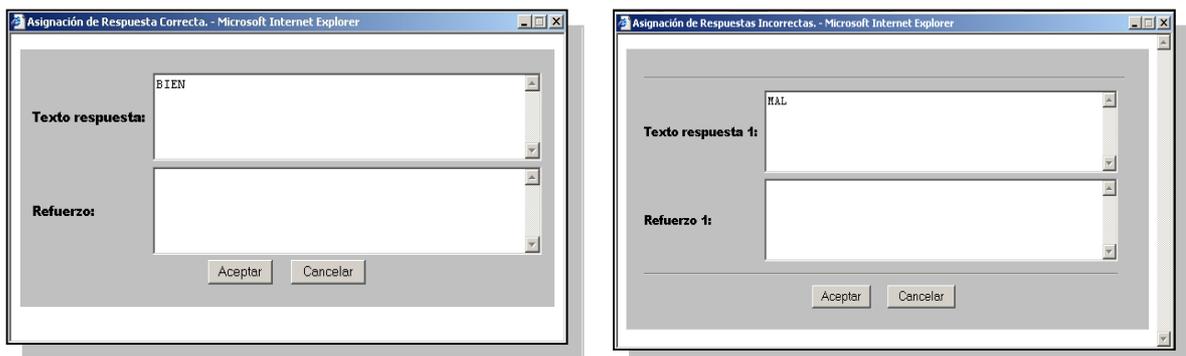


Figura 4.16. Respuesta correcta y respuestas incorrectas.

Para el ítem del test que estamos creando, el applet sólo devuelve dos repuestas, “BIEN” para una respuesta correcta y “MAL” para una respuesta incorrecta.

- **Datos de los temas asociados al ítem.** En este apartado se comunica al sistema los temas a los que pertenece la cuestión que se está creando. Basta con seleccionar aquellas que se deseen, para establecer la relación: “cuestión-temas” (véase figura 4.17).

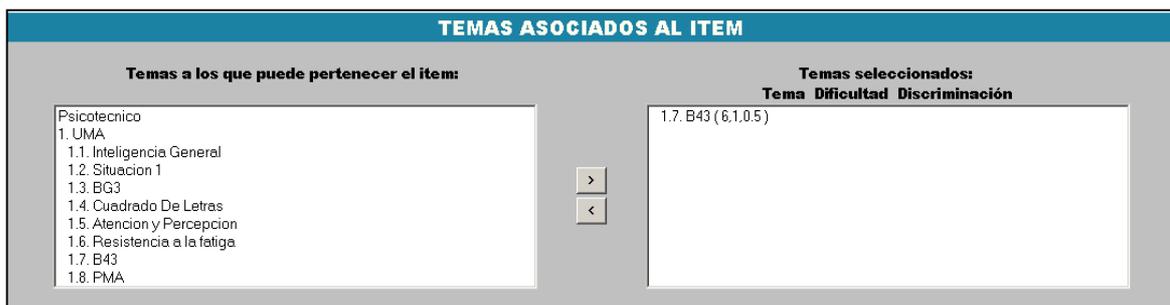


Figura 4.17. Datos de los temas asociados al ítem.

Antes de conectar dicha relación, estableceremos algunos parámetros que son los siguientes:

Grado de dificultad del ítem: Indicador que dice cómo de difícil es la cuestión dentro del tema del que forma parte. Es uno de los tres parámetros que sirven para calibrar las preguntas y permitir identificar la más apropiada para cada alumno.

Factor de discriminación: Es proporcional a la exactitud que tiene la pregunta para clasificar adecuadamente el conocimiento del alumno según la respuesta dada a esta pregunta. Un índice de discriminación alto quiere decir que es muy probable que un alumno cuyo conocimiento supere la dificultad fijada acierte esta pregunta, y que uno que no la supere no la acierte. Por el contrario un factor de discriminación bajo indica que aunque el conocimiento del alumno sea mayor que la dificultad de la pregunta, no es seguro que acierte. Para preguntas cuya dificultad no está muy clara deben emplearse factores de discriminación bajos.

Factor de adivinanza: Es la probabilidad de acertar un alumno la pregunta sin tener conocimientos previos de la materia que se le está mostrando.



The image shows a Windows-style dialog box with the title "Asignación de Dificultad y Discriminaci...". It contains three labeled input fields: "Dificultad:" with a dropdown menu showing the value "6", "Discriminación:" with a text box containing "1.0", and "Factor de adivinanza:" with a text box containing "0.5". At the bottom of the dialog are two buttons: "Aceptar" and "Cancelar".

Figura 4.18. Asignación de la dificultad, discriminación y del factor de adivinanza.

En nuestro ejemplo pondremos una dificultad de 6, una discriminación de 1.0 y un factor de adivinanza de 0.5 .

Cuando ya se haya completado los datos del formulario para la inserción del ítem se pulsa el botón de insertar, o si se desea se puede primero previsualizar el ítem antes de insertarlo. Y después de insertarlo, éste aparecerá en el árbol curricular colgado del tema donde se ha creado.

Capítulo 5.

Implementación de la biblioteca de test.

5.1 Programación JAVA en la implementación de los tests.

Todo el contenido de la biblioteca ha sido implementado en JAVA. Cada test psicotécnico es un applet que se incluye en los documentos que proceda. Toda las clases desarrolladas para la biblioteca de test es una extensión de la clase *SietteApplet*. Esta clase es la encargada de conectar nuestros applets con el sistema de evaluación. En ella se definen varios métodos para la comunicación de nuestros applets con SIETTE.

- Clase SietteApplet

La clase más importante de toda la biblioteca y la que proporciona el sistema de evaluación para comunicase con nuestros applets. Esta clase es una extensión de la clase JAVA *Applet* a la que se le ha añadido varios métodos que más tarde detallaremos su funcionamiento. Se ha definido como abstracta (*abstract*) para que se pueda utilizar como clase base para la herencia de applets para SIETTE, es equivalente al prototipo de una función en C++.

Como su nombre indica, esta clase va a definir los métodos de los applets que SIETTE manejará en la presentación de los mismos. Estos métodos se pueden dividir en dos tipos:

➤ Métodos de evaluación:

Estos métodos son definidos como abstractos (*abstract*), es decir, indica que el método no está definido en la clase, sino que ha sido declarado ahí para ser

definido en una subclase. Por lo tanto es el propio test psicotécnico (subclase de *SietteApplet*) el que debe de definir estos métodos.

* *Evaluacion()*.

Evalúa el test (applet) devolviendo una cadena de caracteres. Nosotros devolveremos “*BIEN*” si se ha realizado correctamente y “*MAL*” si no se resolvió como debería. En algún test además de poder devolver la cadena “*MAL*”, también se podrá retornar otros valores dependiendo del porcentaje de error de la pregunta devolverá “*REGULAR*” o “*MUYMAL*”.

* *Resolver()*.

Este método no devuelve nada, sólo es el encargado de resolver el test (applet) mostrando la solución cuando el sistema de evaluación lo requiera, o el usuario lo desee.

➤ Métodos de comunicación con el sistema.

* *RespuestaActiva(String resultado)*.

Este método recibe como parámetros una cadena de caracteres que es el resultado de evaluar el applet (test). Su función es enviar al sistema, el resultado de la prueba. Este método se utiliza cuando activamente el usuario quiere continuar realizando otra cuestión o el sistema decida pasar a otra cuestión por el hecho de ser un applet temporizado.

Este método ha sido declarado como protegido (*protect*), es decir, sólo las subclases de la clase donde esté descritos y nadie más puede acceder a este método de instancia.

* *RespuestaPasiva()*.

Este método no recibe ningún parámetros y devuelve una cadena de caracteres. Su función es realizar una llamada al método *Evaluación()* (anteriormente descrito) y retornar el resultado de esa llamada. Este método se ha declarado como público (*public*), cualquier clase desde cualquier lugar puede acceder al método de instancia.

* paraReloj().

Este método ni recibe ni devuelve nada. Su función es parar el cronómetro que SIETTE usa para aquellos applets que son temporizados. Este método se ha declarado como público (public), cualquier clase desde cualquier lugar puede acceder al método de instancia.

* reiniciarReloj().

Este método ni recibe ni devuelve nada. Su función es reiniciar a cero el cronómetro que SIETTE utilizar para aquellos applets que son temporizados. Este método se ha declarado como público (public), cualquier clase desde cualquier lugar puede acceder al método de instancia.

* continuarReloj().

Este método ni recibe ni devuelve nada. Su función es poner en marcha el cronómetro de SIETTE con el valor que posea en ese momento (applets temporizados). Este método se ha declarado como público (public), cualquier clase desde cualquier lugar puede acceder al método de instancia.

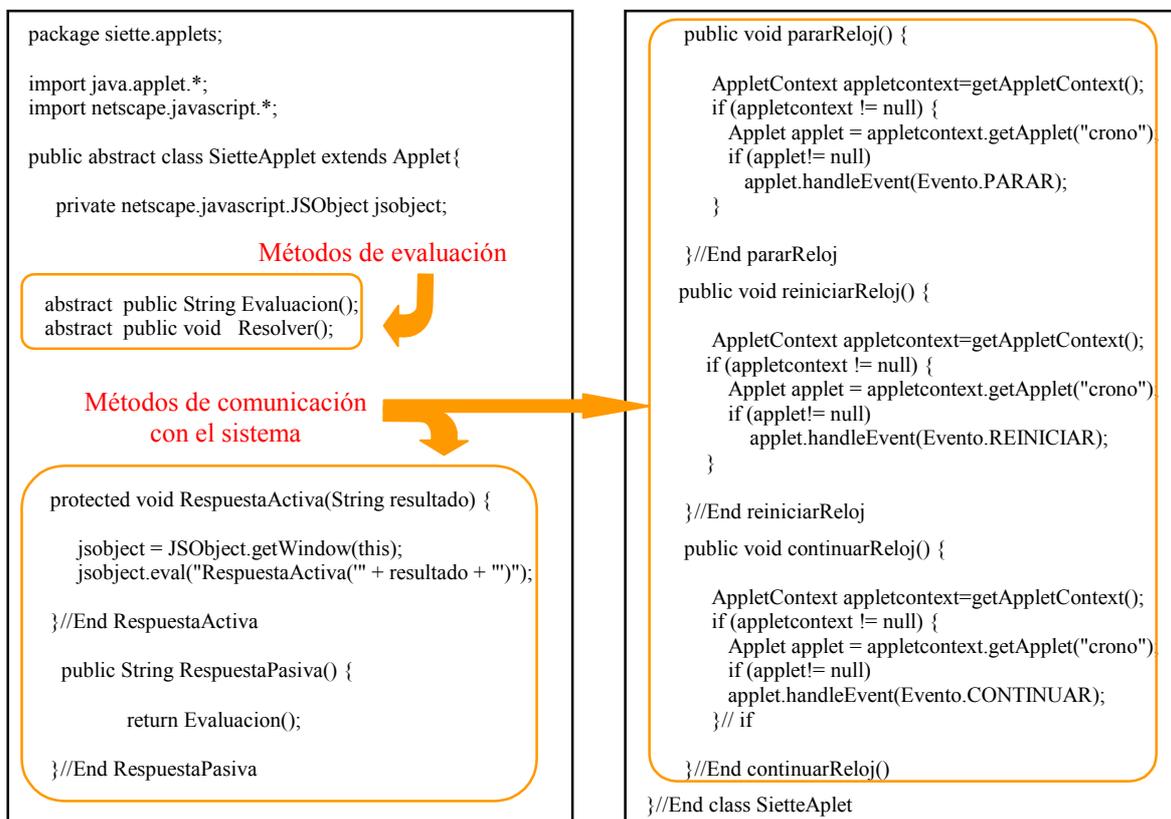


Figura 5.1 Código de SietteApplet.java

5.1.1 Clases que componen nuestra biblioteca.

En este punto comentaremos resumidamente alguna de las clases más importantes de Java, las cuales han sido creadas para la construcción de nuestra biblioteca de tests psicotécnicos. Durante el desarrollo de este epígrafe no expondremos el código en sí de la clase, sino una pequeña descripción de su función con su correspondiente nombre. Para ello iremos detallando, test a test, las clases que lo definen.

➤ Test de Dominós

Este test lo forman, por defecto, 44 plantillas que son la base para construir los ejercicios que se van a mostrar al usuario. Éstas están almacenadas en un fichero de configuración llamado DefPlan.ig con una estructura especial, pero el usuario si quiere puede crearse su propio fichero con su base de plantillas y respetando la estructura de definición y pasando este fichero al applet. Las plantillas están clasificadas según el *modelo* o representación espacial en la que estén almacenadas. A partir de estas plantillas se generarán otras de una forma aleatoria que serán las usadas para la creación de nuestro test. Existe otro fichero llamado DefMod.ig en el que se define los parámetros para la representación gráfica o el modelo de representación espacial sobre la pantalla. Entre otros parámetros se definirán la distancia entre fichas, número de valores que puede tomar una figura, etc. Al igual que en el anterior fichero, el usuario puede crearse su propio fichero de modelos y pasárselo por parámetros al applet (véase el capítulo 7 apartado 7.2.1. para ver la estructura de los ficheros para el buen funcionamiento del applet).

A continuación vamos a comentar las clases más importante dentro de este test, para su implementación:

*** IG.java**

La clase **IG** desciende directamente de la clase *SietteApplet* descrita antes, y por lo tanto, hereda su representación interna (variables y funciones miembro). Este nombre deriva de las siglas de Inteligencia General ya que este test mide esta aptitud.

Esta clase reside todo el cuerpo para la construcción del ítem o ejercicio del test de dominós. Aquí se encuentra muchas de las variables de la aplicación que nos describen información general sobre el mismo. Como por ejemplo:

- *numPlantilla* que nos determina la plantilla dentro del fichero de plantillas que queremos mostrar.
- *pasoAleatorio* nos fija el incremento que debemos sumar a cada componente para generar otra plantilla distinta.
- *tipo* es la variable que guarda el modo de representación gráfica de las fichas sobre la pantalla.
- *test* es un array que contiene la configuración de las fichas que se van a utilizar para la presentación del ejercicio a realizar por el alumno.
- *figuraSolucionTest* es la variable que alberga la representación de la ficha solución que el sistema ha configurado para la solución del ejercicio o ítem presentado.
- *figuraSolucionValor* es la variable que alberga la representación de la ficha solución que el usuario ha configurado para la solución del ejercicio o ítem presentado.

La clase ***PlantillaRaton*** es una clase anidada en la clase IG y es la encargada de manejar los eventos que se producen con el ratón, tales como la pulsación del ratón sobre la ficha solución para configurarla (*mouseClicked*), el cambio del cursor del ratón cuando se entra o sale de la superficie definida por la figura solución (*mouseEntered* y *mouseExited* respectivamente).

Esta clase implementa la interfaz JAVA *MouseListener*, que es la destinada a la gestión de los eventos de ratón que se produzcan, y la que notifica al objeto asociado a ella la acción que ha sido registrada, tal como un click, pulsación o soltado de un botón del ratón o cualquier otra acción que se pueda realizar con un ratón. Se produce un evento *MouseEvent* cada vez que el cursor movido por el ratón o sale de un componente visible en la pantalla, al clicar, o cuando se pulsa o se suelta un botón del ratón. Los métodos de la interfaz *MouseListener* se relacionan con estas

acciones, y son los siguientes métodos: *mouseClicked()*, *mouseEntered()*, *mouseExited()*, *mousePressed()* y *mouseReleased()*. Todos son void y reciben como argumento un objeto *MouseEvent*.

* **Figura.java**

Es la clase de la cual deben de heredar todas aquellas que quieran implementar la representación gráfica de las figuras que formen parte del test. Esta es una clase abstracta y por definición contienen métodos abstractos los cuales son necesarios para la creación y manipulación de la figuras.

Esta una clase descendiente de la clase JAVA *Canvas*. Un *Canvas* es una zona rectangular de pantalla en la que se puede dibujar y en la que se pueden generar eventos tales como eventos de ratón, por ello su elección. Las *Canvas* permiten realizar dibujos, mostrar imágenes y crear componentes a medida, de modo que muestren un aspecto similar en todas las plataformas.

Las clases que hereden de ésta deberán de definir dos constructores con esta cabecera:

- *nombreConstructor* (String *orientación*, int [] *valor*, Color *background*).

El parámetro *orientación* guardará información sobre la disposición de la ficha que se vaya a crear, y puede tomar solamente estos dos valores: “*vertical*” y “*horizontal*”. El parámetro *valor* es un array de enteros de longitud variable que depende del número de componentes que se defina para el tipo de figura que se presentará al alumno, este número se fijará en el archivo de configuración de modelos que comentamos antes bajo el parámetro [NumeroDeComponentesDeLaFigura](#). Cada componente determinará una zona en la figura y en esas zonas se visualizarán las correspondientes representaciones gráficas asociadas a los números almacenados en el array *valor*. Y por último el parámetro *background* tomará el valor del color fondo de la ficha.

Los posibles valores que tomará cada componente del array *valor* oscila entre 0 y el máximo valor que pueda alcanzar esa componente menos 1. Estos máximos

se lo comunicaremos al applet mediante el método *rangoValores* que más tarde definiremos.

Expondremos a continuación un ejemplo de fichas en la que se puede observar una distinta representación gráfica (figura 5.2).

La figura (a) representa una naipa de la baraja francesa, en ella hemos definido dos zonas, una para poner el número y otra para dibujar palo del naipes. Previamente se ha tenido que precisar el parámetro [NumeroDeComponentesDeLaFigura](#) a 2. Por lo tanto el array *valor* sólo tendrá dos elementos, el primero será el número de la carta y el segundo será el palo. Aquí vemos que el rango de valores que puede tomar cada componente del array es distinto, luego debemos comunicarle al applet cuál es el máximo valor que puede tomar cada componente, como dijimos antes, esto se realizará a través del método *rangoValores* (p.e. `public int[] rangoValores(){ return new int[]{12,4}; }`). Al primer elemento del array habrá que sumarle 1 ya que éste oscila entre 0 y 11, al segundo elemento cada palo tiene asociado un número, así pues el valor 0 será para el palo de corazones, el 1 para el de picas, el 2 para el de trébol y 3 para el de diamantes. De esta manera el valor del array *valor* de la figura (a) será {3,0}.

La figura (b) representa una ficha de dominó y por lo tanto tiene dos zonas, los valores que pueden contener estas regiones estarán comprendidos en el rango (0,6) ambos inclusive, éstos representarán el número de puntos que se dibujarán, así pues en la figura (a) el valor del array *valor* será {5,6}. Previamente debe haberse definido el parámetro [NumeroDeComponentesDeLaFigura](#) a 2.

La figura (c) es un pentágono en el que se han determinado 5 zonas que son las que se muestran en la figura 5.2, por lo tanto tendremos que establecer el parámetro [NumeroDeComponentesDeLaFigura](#) a 5. Éstas pueden rellenarse de varios colores preestablecidos por el programador, cada color tiene asociado un número que es el que se almacenará en el array *valor*, al haber definido el número de componentes a 5, el rango de los posibles valores que puede tomar

los elementos del array oscila entre 0 y 4. Así pues el programador puede definir que el valor 0 sea el color rojo, el 1 sea el color verde, ...y el 4 sea el color azul (ver figura 5.2 (c)). Para el ejemplo el array *valor* será {0,1,2,3,4}.

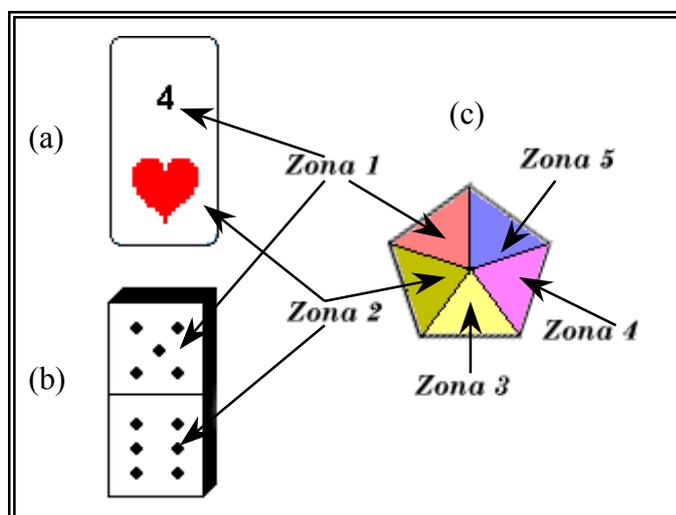


Figura 5.2. Zonas de una figura.

Este constructor creará una figura con la orientación, valor y color de fondo que se le introduzca como parámetro, lo que nosotros llamaremos como figura rellena. A veces puede que la orientación de la figura nos de igual como ocurre con el pentágono de la figura 5.2.

- *nombreConstructor* (Color *background*).

El parámetro *background* al igual que en el anterior constructor, guardará información sobre el color del fondo de la figura.

Este constructor sólo creará una figura con orientación “*vertical*”, valores no validos y con el color de fondo que se le pase como parámetro, lo que nosotros llamaremos una figura vacía. Esta figura hará el papel de figura solución, es decir, figura que el alumno tendrá que configurar para crear la solución.

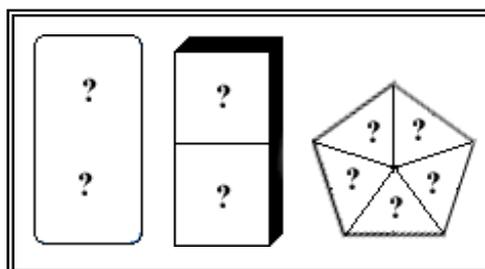


Figura 5.3. Figuras vacías.

Además de implementar estos dos constructores tendrán que definir aquellos métodos que son abstractos, y que explicaremos su función a continuación:

- void *modificar* (int *pos*, int *value*).
Este método modifica el valor que ocupa la posición *pos* en el array *valor* (ver cabecera constructor) por el valor *value*.
- void *actualizar* (int *x*, int *y*, int *incremento*).
Su función es actualizar el valor asociado a la zona que pertenezca el punto (x,y) dentro de la figura sumándole el incremento que se le pasa como parámetro. Cuando se habla se actualizar se entiende por reemplazar la visualización anterior por la nueva configuración de la figura, es decir, se tendrá que volver a repintar.
- int *valor* (int *pos*).
Devuelve el valor que contenga el array *valor* en la posición *pos*.
- int *tipoFigura* ().
Devuelve el tipo de la figura, es decir, si la figura es una figura rellena devuelve el identificador FICHA_RELLENA y si por el contrario la figura es una figura vacía devolverá el identificador FICHA_VACÍA.
- int [] *rangoValores* () .
Devuelve un array de enteros en el que la posición i-ésima del mismo establece el número de valores distintos que puede tomar el valor de la posición i-ésima del array *valor*.
- int *getWidth*() y int *getHeight*() .
Devuelven el ancho y el alto respectivamente de la figura.

Al heredar de la clase JAVA Canvas también se deberá de redefinir el método `paint` de esta clase para poder pintar la representación gráfica de la figura sobre el área de dibujo.

➤ **Test Cuadrado de letras**

Este test lo conforman una serie de matrices de letras que van a dar lugar al objeto de estudio de nuestra prueba. A partir de estas matrices se generarán otras de una forma aleatoria que serán las usadas para la creación de nuestro test.

* **MatrizDeLetras.java**

Para la implementación en JAVA se ha construido esta interfaz en la que se incluyen definiciones muy generales de variables que van a servir al resto de clases como identificadores: se ha asignado a variables varios modelos de matrices de letras que serán el molde para crear otras nuevas de forma aleatorio pero manteniendo la misma base que la original.

* **CuadradoDeLetras.java**

La clase CuadradoDeLetras desciende directamente de la clase *SietteApplet* descrita antes, y por lo tanto, hereda su representación interna (variables y funciones miembro).

Como su propio nombre indica, esta clase va a construir y mostrar un ítem o ejercicio del test cuadrado de letras, en ella se definen todos los parámetros, variables y métodos que necesitemos para crearlo.

Para la creación de la matriz de letras se ha implementado una clase que se encarga de ello esta clase es la más importante y se llama **Cuadrado**. Esta clase es una subclase de CuadradoDeLetras que está definida como *protected*, desciende de la clase JAVA Panel, esta clase define los *Container* (contenedores) más sencillos, capaces de contener otros elementos del **AWT** (Abstract Window Toolkit). Es la responsable final de que las casillas de la matriz de letras aparezcan dibujados en la pantalla. Esta clase representará el contenido de la matriz mediante un cuadrado que almacenará los caracteres. La implementación de este cuadrado se realiza con una matriz de 4x4 objetos de tipo **Casilla**.

La clase **MouseBoton** es la encomendada para el manejo de los eventos del ratón que se produzca en la selección de las casillas para formar la solución de la prueba.

Es una clase que implementa la interfaz *JAVA MouseListener*, que es la destinada a la gestión de los eventos de ratón que se produzcan. El funcionamiento de esta clase es seleccionar y dar nota de ello, para la formación de la solución, las casillas que el usuario elija y dotándola de un color distinto aquellas que sean seleccionadas, actualizando la solución de la aplicación.

✱ **Casilla.java**

La clase *Casilla* es la destinada a dar una visión gráfica de los componentes que forman nuestra matriz de letras. Es una clase que desciende de la clase *JAVA Canvas*.

Esta clase se ha implementado para que tenga un funcionamiento parecido al que pudiera tener un botón de comando.

➤ **Test de las Figuras Iguales o B.G.3**

Este test lo forman un conjunto de figuras, todas de ellas distintas. Se presentarán al usuario un conjunto de 10 figuras, dándose la característica que una de ellas esta repetida. El cometido de esta prueba es encontrar y seleccionar (pulsando sobre las casillas) aquellas figuras que sean iguales.

✱ **BG3.java**

La clase *BG3* desciende directamente de la clase *SietteApplet*, y por lo tanto, hereda su representación interna (variables y funciones miembro). En esta clase reside todo el cuerpo de la implementación de la cuestión; en ella se eligen de forma aleatoria las figuras que se van a mostrar distinguiendo una de ellas para crear otra homónima para construir la pregunta.

La clase *Tests* es una subclase de *BG3* y, en realidad, es la responsable de elegir y producir las figuras que compone el test de figuras iguales. Se ha realizado un estudio sobre las posibles combinaciones de las posiciones que los dos puntos pueden tomar (sin pisar uno al otro) y obteniendo así el total de imágenes distintas que utilizaremos, este número es de 28 figuras diferentes. De estas figuras se

elegirán 9 aleatoriamente para la exposición y luego de entre estas 9 se escogerá 1 al azar y se duplicará. Estas figuras serán mostradas al usuario desordenadas y será él quién tratará de encontrar las imágenes iguales.

La clase Figuras es una subclase de BG3 y descende de la clase JAVA Canvas. Por su propia definición y al descender de Canvas, esta clase tratará el tema de la representación gráfica de las figuras. Con esta clase se construirá todas las imágenes que mostraremos al usuario.

Esta clase elaborará figuras en la que se representará un cuadrado con dos puntos, estos puntos estarán situados en el perímetro de este rectángulo, pero no en cualquier posición, sólo podrá situarse a los extremos de los lados del cuadrado o en el centro de estos lados (véase la figura 5.4).

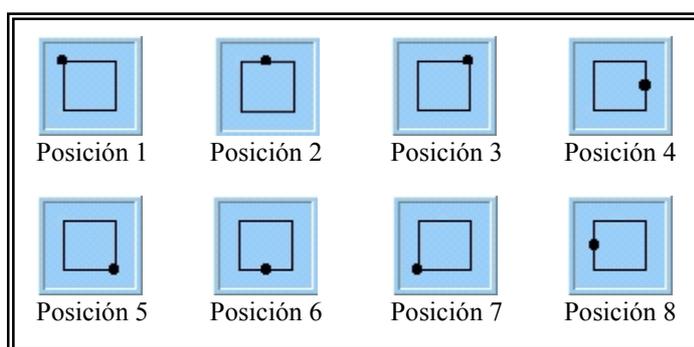


Figura 5.4 Posibles posiciones de los puntos.

➤ Test de Razonamiento y Estructuración Espacial (B43 de Bonnardel)

El test de razonamiento y estructuración espacial lo forman, por defecto, un conjunto de 25 piezas y 19 plantillas o puzzles. Estas piezas y plantillas están definidas en los ficheros DefFiguras.b43 y DefPlant.b43 respectivamente, estos ficheros están definidos de una forma especial (ver Capítulo 7 epígrafe 7.2.4.).

Cada pieza tiene asociada cuatro imágenes que corresponde con las posibles rotaciones de 90 grados que pueden experimentar éstas en el transcurso de la ejecución de la aplicación, nombrándose de la siguiente manera: “[figura](#)” + número_de_la_figura + “ - ” + rotación + “.gif”, donde rotación puede tomar los valores 0 (0°), 1 (90°), 2 (180°), 3 (270°) y la rotación será en el sentido de las agujas del reloj.

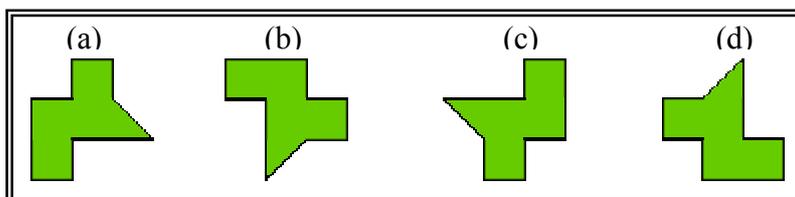


Figura 5.5. (a) [figura12-0.gif](#) (b) [figura12-1.gif](#) (c) [figura12-2.gif](#) (d) [figura12-3.gif](#)

Se muestran por pantalla todas las piezas que se definan en el fichero de figuras y un puzzle, que se introduce al applet como parámetro y muestra un error si no existe, este parámetro será el número o identificador del puzzle definido en el fichero de plantillas.

El puzzle tiene dos modos de presentación que son: mostrando el limite de las piezas que lo compone o sin mostrarle el borde. Atendiendo a esta característica, cada puzzle tiene asociado a él dos imágenes, una con el borde pintado y otra sin él, de esta manera el nombre del puzzle será determinado por esta característica y será nombrado así: “Plantilla” + número_de_la_plantilla + “-” + borde + “.gif”, donde borde tomará el valor 0 si no se tiene el borde de las piezas y 1 si lo tiene. Esta característica será definida como un parámetro de entrada del test (applet). Todas la imágenes deben ser gifs, tanto las piezas como las plantillas.

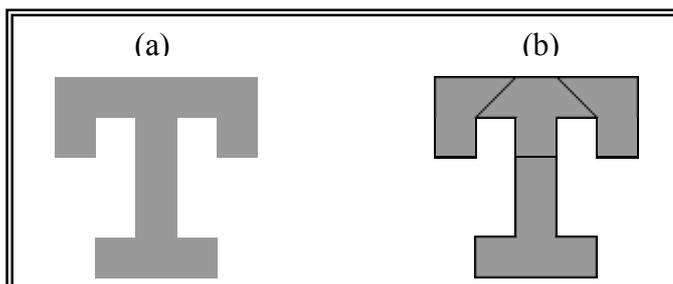


Figura 5.6. (a) [Plantila3-0.gif](#), (b) [Plantilla3-1.gif](#)

El objetivo de esta prueba consiste en elegir y situar, en una posición correcta, las figuras en el puzzle, para ello se podrá rotar las figuras, desplazarlas sobre la pantalla para colocarlas sobre la plantilla.

* **B43.java**

La clase B43 desciende directamente de la clase *SietteApplet*. Esta clase será la encargada de crear las figuras o piezas que se especifiquen en el fichero de configuración de las figuras para presentárselas al alumno que realizará la prueba, así como también de obtener del fichero de plantillas, el puzzle que se introduce por parámetro a la aplicación y colocarla en la pantalla para descubrir las piezas que la compone.

En esta clase se tratará de resolver el puzzle que se presenta para ello el alumno deberá de rotar y mover las piezas hasta colocar sobre el puzzle y luego soltarlas. Si la pieza que ha elegido está en su rotación correcta y pertenece al puzzle, se situará sobre él en la posición que le corresponda.

➤ **Test de Atención y Percepción**

El código que define el test de atención y percepción implementará las partes que se describió en el capítulo 6 epígrafe 6.2.8. Esta será su misión y para ello se implementará en el siguiente fichero el código necesario para el desarrollo y funcionamiento del test.

* **AtencionPercepcion.java**

La clase *AtencionPercepcion* hereda de la clase *SietteApplet*. Esta clase generará aleatoriamente las cadenas de caracteres que se mostrará al alumno para éste determine si son o no iguales y anotar la solución en la columna central de la aplicación que se presenta al usuario. La clase a su vez determinará internamente las soluciones para luego poderlas comprobar con las que el alumno obtenga, teniendo en cuenta o no los fallos cometidos por éste. Se podrá programar la aplicación en varios niveles.

➤ **Test de Resistencia a la Fatiga**

Este test, como dijimos en el capítulo anterior, mide nuestra facultad para realizar tareas periódicas pero que no son totalmente mecánicas y que requieren de todo nuestro interés. El código que implementará la actuación de este psicotécnico está en el siguiente fichero:

* **ResistenciaFatiga.java**

La clase *ResistenciaFatiga* extiende de la clase *SietteApplet* heredando todos sus métodos y variables. Esta clase construirá aleatoriamente filas de 25 números comprendido entre 0 y 9. El número de filas que genere dependerá del nivel en el que esté configurada la aplicación. Cada vez que se cree una de estas filas, se calculará la solución asociada a ella, es decir, calcular los números de la columna que ocupa el primer número de las parejas que sume 10. Estas soluciones se

compararán con las introducidas por el alumno, para devolver un resultado en el que se ha tenido en cuenta o no los fallos cometidos por éste.

➤ **Test de Aptitudes Mentales Primarias o PMA**

Este test está diseñado para calcular varios factores, dependiendo del valor del parámetro *Tipo* que se le pasa a la aplicación. Este parámetro puede recibir varios valores que son los siguientes: “FactorR”, “FactorV”, “FactorN”, “FactorE”. La implementación de cada uno de los factores se ha agrupado en un único fichero que a continuación detallaremos.

✱ **PMA.java**

La clase PMA, como todas las anteriores, descende de la clase *SietteApplet*, necesaria para acoplarla al sistema de evaluación SIETTE. En esta clase como se ha explicado antes, dependiendo del valor que tome el parámetro *Tipo* se generará un factor u otro. Una vez decidido que factor se implementará, se creará una instancia de la clase que defina ese factor y se mostrará al alumno para su resolución.

➤ **Test Espacio-Perceptivo o Situación-1**

La aplicación que presenta una cuestión del test espacio-perceptivo o situación-1 está diseñada para construir fielmente la estructura de este psicotécnico. Ésta mostrará un eje de coordenadas en el que el alumno deberá pulsar sobre él para ubicar la flecha en la orientación que se le pida, atendiendo al código de orientación de flechas (ver figura 6.15).

✱ **Situacion1.java**

La clase Situacion1 extiende de *SietteApplet*. En ella se define la clase Flecha que es una subclase de Situacion1 y que es la más importante ya que es ella la encargada de dibujar el eje y la flecha en la orientación que el usuario elija durante la ejecución de la aplicación.

Capítulo 6.

Manual para el uso de la biblioteca de test.

6.1 Introducción.

El sistema SIETTE como ya hemos comentado en muchas ocasiones, es un Sistema Inteligente de Evaluación mediante Tests para la TeleEducación. En SIETTE, tanto la decisión de finalizar el test como el criterio de selección de la siguiente pregunta que hay que mostrar al alumno en cada momento, están basados en el nivel de conocimiento del alumno estimado por el sistema en cada momento. La elección de SIETTE como sistema de autoevaluación tiene muchas ventajas, por eso se ha optado por su uso para la autoevaluación de la materia de tests psicotécnicos.

Para acceder a la autoevaluación de los tests psicotécnicos, el alumno deberá cargar en el navegador de internet, la pagina de inicio del sistema de evaluación SIETTE, ubicado en la URL <http://www.lcc.uma.es/SIETTE>, y hacer clic sobre el enlace “*Aula de Tests*”. A continuación aparecerá una nueva página para crear un alumno nuevo o identificar otro existente. Cuando se haya creado o identificado al alumno se presentará las asignaturas disponibles, de ellas se seleccionará la asignatura “*Psicotécnico*” y seguidamente se mostrará los tests disponibles. El alumno deberá elegir aquel test que desee en ese momento realizar, y el sistema le irá mostrando las cuestiones relacionadas con el test psicotécnico escogido.

Para la realización de un test el mecanismo es muy simple, se le mostrará por pantalla un enunciado que deberá de leer para saber como se debe realizar la prueba y una vez leído se pulsará sobre el botón “*Comenzar*”, seguidamente se mostrará la plantilla del test que haya elegido para realizar. El alumno deberá resolver el ejercicio intentando lograr una respuesta

correcta de la cuestión que se le presenta. En algunos tests aparecerá un botón “*Seguir*”, éste se pulsará cuando se haya obtenido la solución o quiera pasar a la siguiente cuestión, y en otros no aparecerá. En éstos últimos, el modo de pasar a otro ítem es el de resolver obligatoriamente la cuestión o esperar que se termine el tiempo de presentación del mismo.

En la figura 6.1 exponemos un ejemplo de una cuestión dentro del test de dominós para la inteligencia general. En la cuestión se muestra un enunciado en la que si pulsamos el botón de “*Comenzar*” se presentará la plantilla asociada a éste, en la cual se le ha configurado el modo de visualización en *Lineal Doble*.

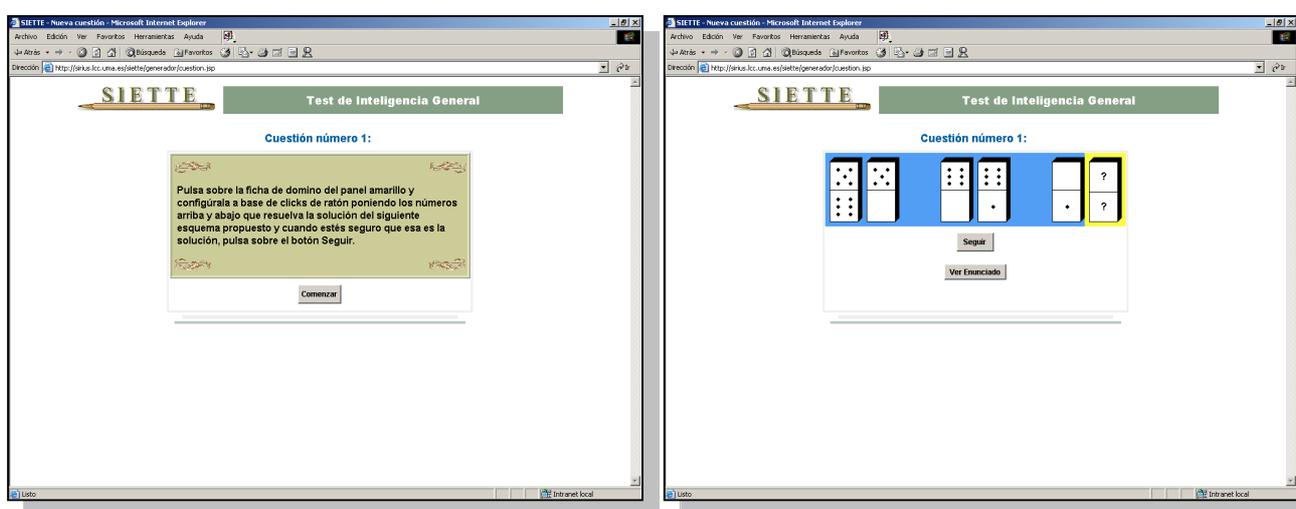


Figura 6.1. Elección de una cuestión dentro del test de Dominós.

Se puede observar como en la figura 6.2 que la solución se realizará sobre la ficha que tiene el fondo en amarillo, eligiendo el alumno la de ficha dominó 1/1 para componer la solución. En la figura 6.3 se aprecia como la elección correcta era la ficha de dominó 0/2 fallando así en la elección de la respuesta.

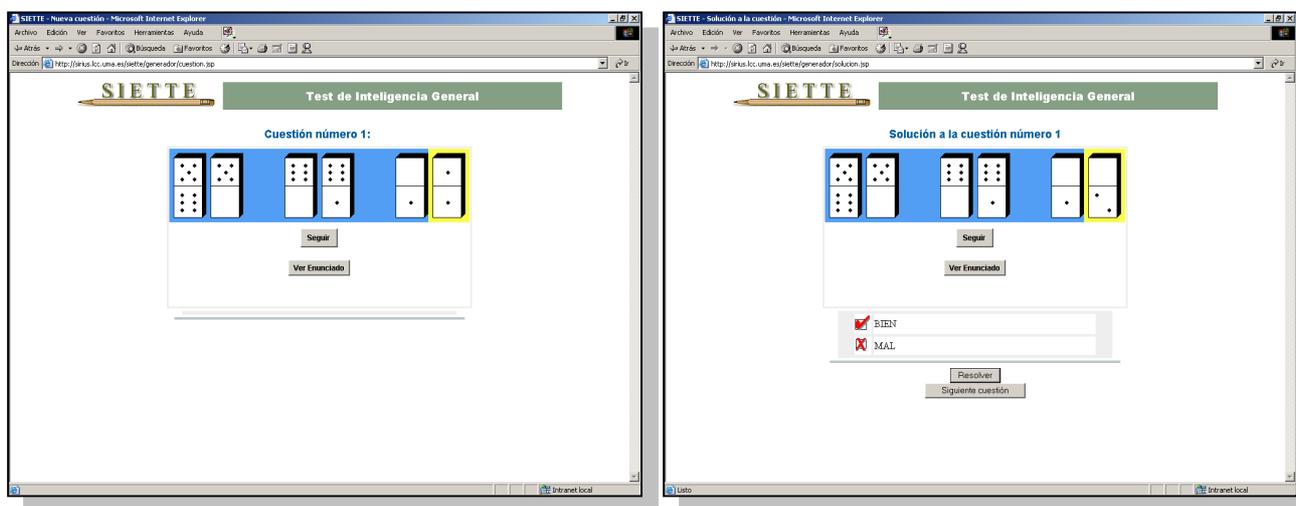


Figura 6.2. Elección de la solución por el alumno para componer la solución.

Figura 6.3. Solución a la elección de la cuestión anterior.

Este proceso se repite a lo largo del test, hasta que se realice un número establecido de preguntas. A la finalización del mismo se le muestra al alumno por pantalla la calificación obtenida en el test como se puede apreciar en la siguiente figura:

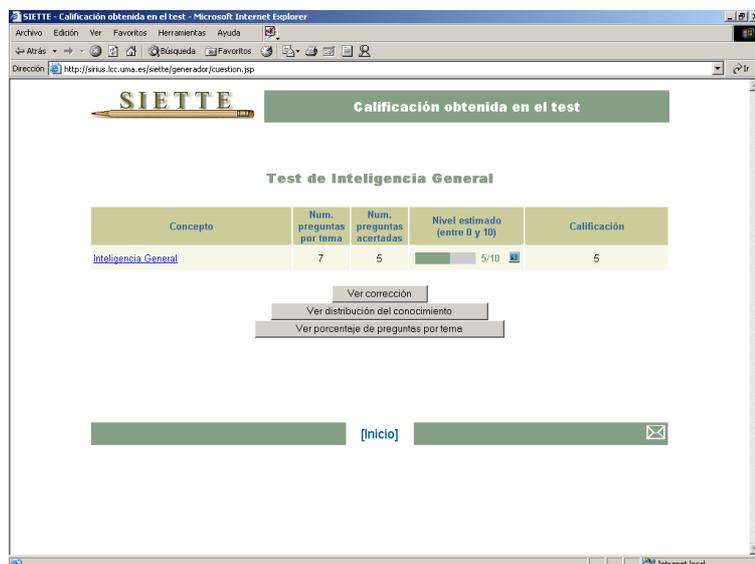


Figura 6.4. Calificación obtenida por el alumno.

La calificación viene distribuida de dos formas una más específica que clasifica cada una de las secciones o temas que comprenden el test, y otra más general que clasifica el test completo abarcando todas sus secciones. Además, en la pantalla de calificación podemos observar los distintos campos que nos ofrece como el número de preguntas acertadas, número de preguntas por tema, y nivel estimado del alumno.

Finalmente se le da la opción al alumno de ver todas las preguntas que ha realizado, a través del botón “*Ver corrección*”, pudiendo comprobar la que hizo y su correspondiente fallo o acierto del mismo. A su vez también tiene la opción de “*Ver la distribución del conocimiento*” y de “*Ver el porcentaje de preguntas por tema*”.

Con todo esto el alumno habrá puesto en práctica sus conocimientos sobre la materia pudiendo autoevaluarse y comprobando así el grado de conocimiento sobre la misma.

6.2 Funcionamiento general de la biblioteca.

La biblioteca consta básicamente de unos applets implementados en lenguaje JAVA, cada applet contiene un test determinado para evaluar un tipo de aptitud. Estas aplicaciones se insertan en el sistema de evaluación SIETTE para su puesta en funcionamiento, éste es el encargado de visualizarlo, recoger los resultados de la prueba y mostrar su calificación teniendo en cuenta el tiempo en que se realiza. Este sistema de evaluación del que hablamos se ha detallado en el capítulo 4.

A continuación vamos a describir cuál de estos tests se han implementados para su uso en ordenador por su fácil diseño y manejo. Éstos servirán para cualquier persona que tengan que realizar una prueba para un puesto de trabajo determinado y se le exija hacer alguno de ellos.

6.2.1 El test de Dominós.

En 1948, Anstey tuvo la idea de utilizar un material homogéneo, constituido únicamente por fichas de dominó, para construir un test cuyo objetivo es medir la inteligencia general.

El test del dominó es uno de los mejores test colectivos de inteligencia general; está destinado a valorar la capacidad para comprender, conceptualizar y aplicar el razonamiento sistemático a nuevos problemas y apreciar las funciones centrales de la inteligencia: abstracción y comprensión de relaciones. Se usa mucho para conseguir el llamado **FACTOR "G"** (general) de la inteligencia.

Este test consta de una serie de plantillas que muestran distintos ejercicios a realizar. El problema es encontrar una ley que rijan una serie de fichas de dominó mostradas, con el fin de completar una ficha virgen que forma parte de este mismo grupo. Para nuestro caso obtuvimos la ley que rigen estas plantillas y hemos implementado ejercicios nuevos, que de una forma aleatoria construyan los ejemplos iniciales, dotando a los ejercicios de una interacción multimedia con el usuario para su fácil manejo y resolución.

Hay diferentes modos de representar las fichas en la pantalla, y dependiendo de esa representación espacial será más fácil de obtener o no esa ley. Se pueden clasificar en 7 modelos:

- **Lineal:** Se muestran las fichas en una misma fila, una al lado de la otra.

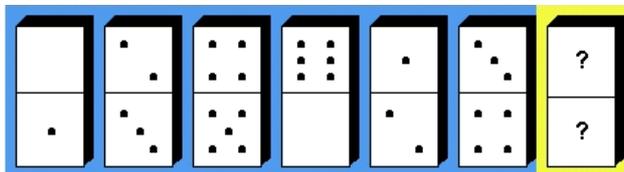


Figura 6.5. Modo Lineal.

- **Lineal Doble:** Se presentan las fichas una al lado de la otra pero en grupos de dos.

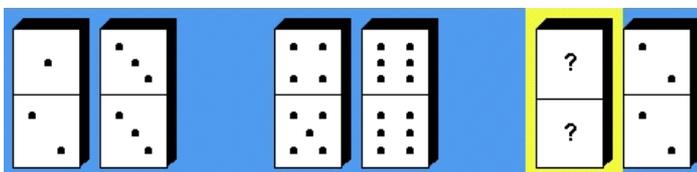


Figura 6.6. Modo Lineal Doble.

- **Lineal Triple:** Se visualizan las figuras al igual que las anteriores, una al lado de la otra pero en grupos de tres.

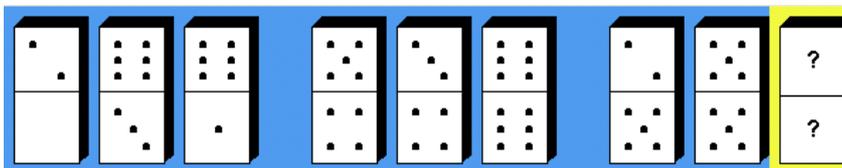


Figura 6.7. Modo Lineal Triple.

- **Estrella:** Se muestran la fichas como se presenta en la figura 6.8a y sigue un orden de inserción que es el que se muestra en la figura 6.8b.

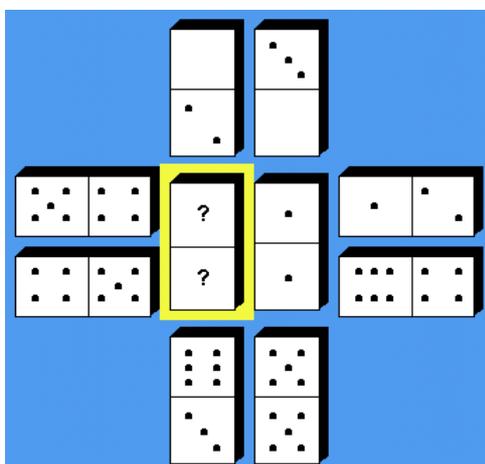


Figura 6.8a. Modo Estrella.

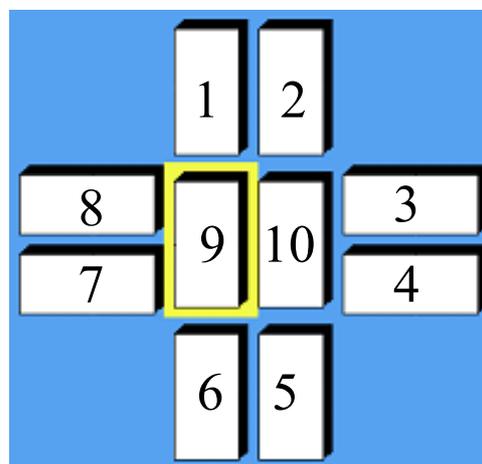


Figura 6.8b. Orden de inserción de fichas

- **Cruz:** Se exponen las figuras como se presenta en la figura 6.9a y responde a un orden de inclusión, el cual se muestra en la figura 6.9b.

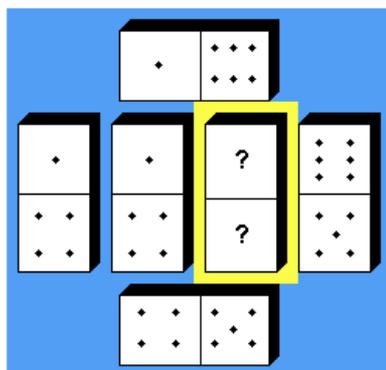


Figura 6.9a. Modo Cruz.

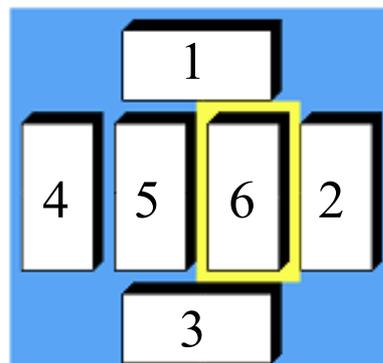


Figura 6.9b. Orden de inserción de fichas en Cruz.

- **Alternada Horizontal:** Se presentan las fichas de dominó en una misma fila, una detrás de la otra pero alternando una ficha horizontal y la siguiente vertical. Comienza con una ficha en posición horizontal, de ahí el nombre del modo.

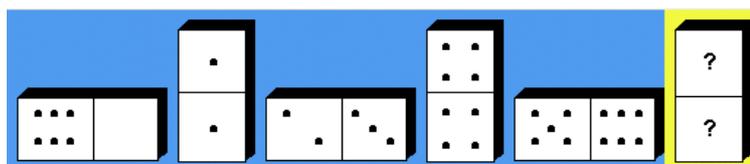


Figura 6.10. Modo Alternada Horizontal.

- **Alternada Vertical:** Se visualizan las fichas de dominó en una misma fila, una detrás de la otra pero alternando una ficha vertical y la siguiente horizontal. Comienza con una ficha en posición vertical, de ahí el nombre del modo.

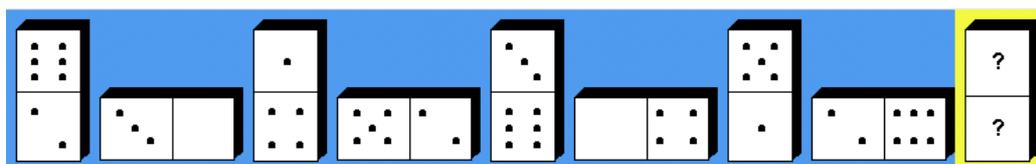


Figura 6.11. Modo Alternada Vertical.

La dinámica de este test es, como dijimos antes, la de encontrar una ley que rijan la serie de fichas de dominó que se muestra al alumno. Una vez encontrada, el alumno deberá configurar la ficha “virgen”, que es la que se encuentra en el fondo amarillo con los signos de interrogación. Esto se llevará a cabo haciendo uso del ratón, clicando sobre ella para establecer la solución a este ejercicio.

6.2.2 Test de Cuadrados de Letras.

La prueba *Cuadrados de Letras*, tal como se presenta, es un buen instrumento para medir la percepción y atención en niveles elevados. Esta prueba se incluye normalmente entre las que miden *Aptitudes perceptivas y de atención*. La aplicación de la prueba esta destinada a personas de 11 años en adelante.

La prueba consta de una serie de cuadrados en los que hay 16 letras distribuidas en 4 filas y 4 columnas. Solamente en una fila o en una columna hay una letra repetida. El test consiste en buscar en el cuadrado de letras en qué fila o columna hay una letra repetida. Cuando la encuentre, tache con una línea toda la fila o columna.

Para que el alumno realice de una forma cómoda la prueba, se ha diseñado el cuadrado de letras como una matriz de botones en la que éste deberá pulsar sobre aquellos botones que contenga letras repetidas teniendo en cuenta la restricción de que sólo puede aparecer dos letras en una misma fila o columna.

Estos cuadrados de letras se generan aleatoriamente, por tanto cuando se cree una nueva ejecución de la aplicación aparecerá una configuración de letras distinta, produciendo a su vez una nueva solución para esa configuración.

A continuación se muestra la apariencia de la prueba (figuras 5.8a-b) que se presenta al alumno una vez que es llamada desde SIETTE:

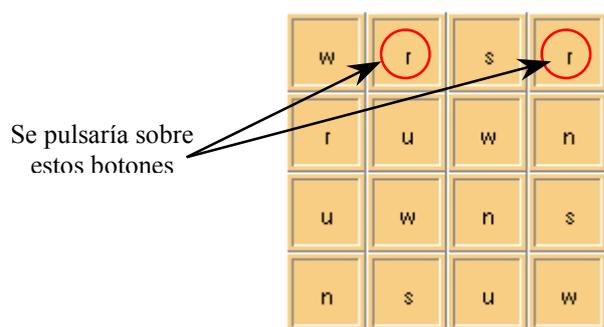


Figura 6.12a. Cuadrado de letras.

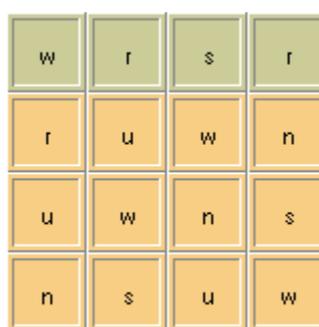


Figura 6.12b. Solución del cuadrado de letras.

6.2.3 Test de las Figuras Iguales o B.G.3.

Test de las figuras iguales, forma parte de la Batería del Profesor Bonnardel en su modalidad: *II Examen a nivel directivo y mandos medios*.

Sirve para obtener la nota CPM: *Rapidez perceptiva y mental*, nota especialmente relacionada con los trabajos rutinarios de oficina. Generalmente las fluctuaciones del rendimiento de los sujetos en este tipo de pruebas son bastante grandes. En virtud de esto, es conveniente utilizar varias pruebas, en vez de una sola y calcular un rendimiento medio.

En esta prueba se le presentan al sujeto filas de 10 dibujos cada una. En cada fila debe de buscar los dibujos que son completamente idénticos y pulsando con el ratón sobre ellos. La aplicación de la prueba puede ser individual o colectiva y la duración de la misma es de 6 minutos.

Esta sería el aspecto que ofrecería el sistema de evaluación (SIETTE) al alumno para la realización de la prueba:

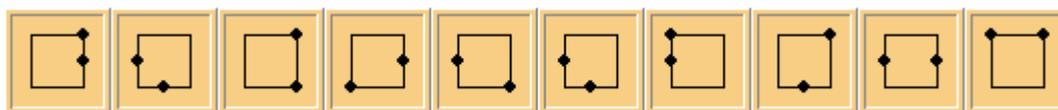


Figura 6.13a. Test de las figuras iguales.

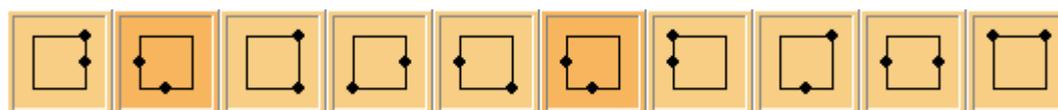


Figura 6.13b. Solución al test de las figuras iguales.

6.2.4 Test de Razonamiento y Estructuración Espacial (B 43 de Bonnardel).

Este test de razonamiento fue propuesto por Bonnardel. El objetivo de este test es apreciar la manera en la que se organiza el pensamiento frente a una tarea que pide una buena representación y estructuración espacial. Este test es frecuentemente utilizado para la selección de candidatos a un empleo en la industria y, en particular, para los obreros profesionales y los aprendices.

En el test original se le presentan al individuo cuarenta piezas de madera y un cuaderno de 10 figuras. El alumno deberá reproducir las figuras geométricas que aparecen con las piezas de madera. Para nuestra prueba se ha elaborado una batería de ejercicios, que consta de 19 figuras y veinticinco piezas. Pero se da la posibilidad al profesor de crear más figuras y piezas.

Esta prueba debe reproducir, con la ayuda de las piezas, las figuras geométricas de la batería. Los ejercicios se realizan completando la figura que aparece a la derecha mediante las piezas de la izquierda. Éstos a su vez podrán ser rotados para que encajen correctamente. Cada vez que se ejecuta la aplicación, tanto las piezas como la figura geométrica cambian. Los primeros aparecen en una disposición y rotación distinta y la segunda suele ser diferente.

Una posible apariencia de este test viene dada por la figura 6.14 en la que han sido colocada dos piezas de la figura geométrica que se pretende completar.

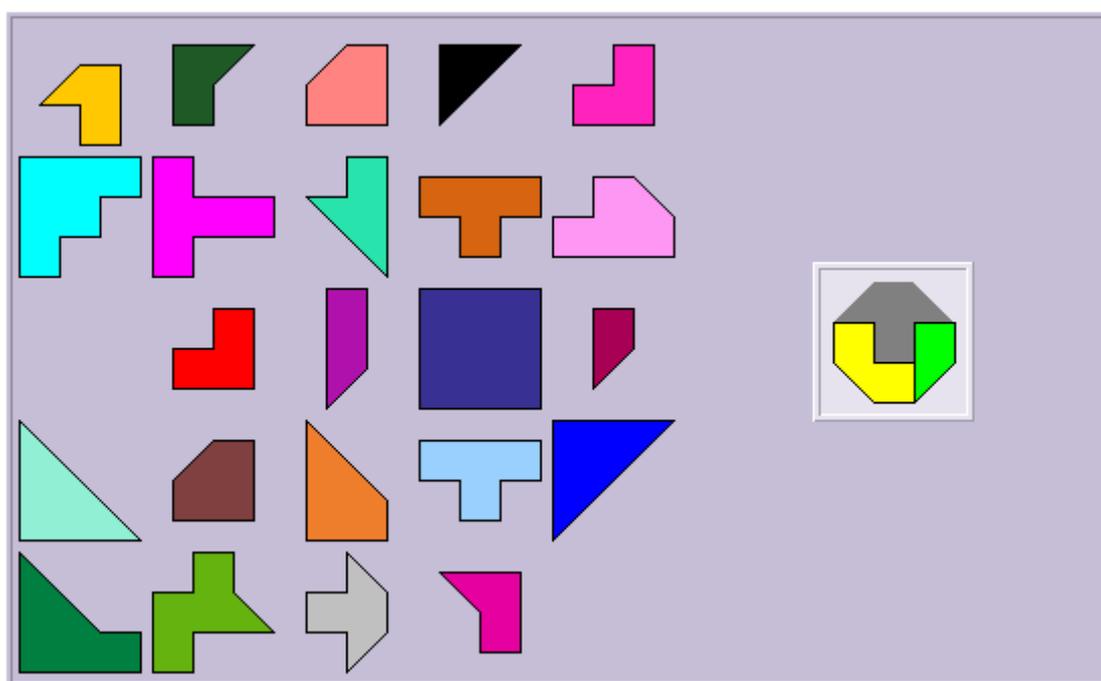


Figura 6.14. Test de Razonamiento y Estructuración Espacial.

6.2.5 Test espacio - perceptivo o Situación-1.

Esta prueba fue una idea original de Nicolás Seisdedos Cubero, Técnico de la Sección de Estudios de TEA Ediciones S.A. y publicado en 1990.

El test espacio-perceptivo o Situación-1 forma parte de un conjunto de herramientas de reciente diseño destinados a medir las dotes perceptivas y de atención empleando diferente material a percibir.

Así pues, la intencionalidad de la medida es, la apreciación de la rapidez perceptivo-espacial, es decir, de las dotes de atención y observación en una tarea con implicación espacial. Esta función mental es básica e instrumental en cualquier actividad humana, sea escolar, profesional o lúdica; todas ellas tienen en su base la captación y localización de una realidad.

Finalmente, hay que señalar que la prueba puede resultar útil tanto en situaciones de medida escolar como de evaluación profesional, pues tanto en una como en otra el componente perceptivo está presente. Asimismo, sus resultados podrían aportar también alguna luz a determinados comportamientos de bajo rendimiento.

En esta prueba se han empleado elementos que están formados por flechas que parten de un punto central y se orientan hacia ocho posiciones distintas del espacio bidimensional de un plano (propio de una prueba de papel y lápiz). La tarea del usuario es determinar dicha dirección de acuerdo con un código simple de seis letras:

A/B = Alto/Bajo D/I = Derecha/Izquierda V/H = Vertical/Horizontal

Veamos un ejemplo de esta prueba, pero antes de nada observemos cuales son todas las situaciones posibles y sus letras correspondientes:

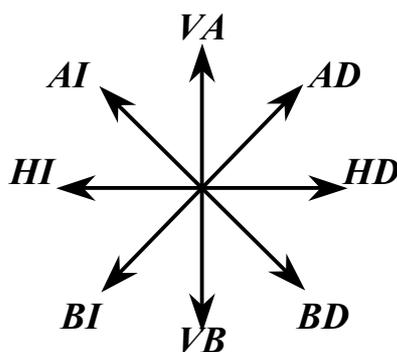


Figura 6.15. Código de orientación de las flechas

En cada ejecución habrá que determinar la orientación de la flecha que se pide en el enunciado de la derecha atendiendo al código antes establecido. A continuación se muestra una posible visualización de la prueba:

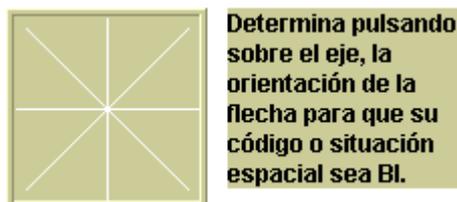


Figura 6.16. Situación-1.

6.2.6 PMA o Test de Aptitudes Mentales Primarias.

La batería PMA (Aptitudes Mentales Primarias), fruto de varios años de estudio de L. L. Thurstone y colaboradores (Chicago, USA, 1938-41), ha sido un instrumento construido sobre las bases de los resultados obtenidos con la técnica del análisis factorial. Tiene, a pesar de sus limitaciones, una gran aplicabilidad para evaluar varios aspectos de la inteligencia en sujetos de diversos niveles de edad y diferentes grados de formación.

La batería PMA permite una evaluación muy importante, aunque no única, de la inteligencia en la labor de psicodiagnóstico, ya que cualquier actividad de la conducta implica todos los rasgos de la personalidad; la inteligencia es una excelente y principal función del individuo, pero el hombre no es sólo inteligencia. Una de las ventajas de la batería PMA; como cualquier otro instrumento psicométrico tiene limitaciones en cuanto a la universalidad, fiabilidad y validez de sus aplicaciones, pero es muy útil en cuanto puede presentar un perfil de las principales dimensiones o aptitudes mentales primarias de la conducta cognoscitiva, para orientar o encauzar a los individuos a las actividades o profesiones en las que pueden destacar.

La batería consta de cinco pruebas que detectan aisladamente cinco factores básicos de la inteligencia; factores que llamó Aptitudes Mentales Primarias y que se identifica con los siguientes nombres:

➤ **Factor V: Comprensión verbal**

Es la capacidad para comprender ideas expresadas en palabras. Se necesita en actividades en las cuales haya que captar los problemas por medio de la palabra escrita o hablada.

La prueba, por defecto, consta de 10 elementos o problemas de elección múltiple que corresponde a un nivel 1 pero se puede configurar hasta un máximo de 5 niveles. Cada nivel tiene $10 * n^{\circ}$ nivel elementos.

El sujeto debe hallar los sinónimos de las palabras propuestas.

RAPIDO	<input type="radio"/> Ligero	<input type="radio"/> Corto	<input type="radio"/> Difamatorio	<input type="radio"/> Delgado
PRODIGO	<input type="radio"/> Despilfarrador	<input type="radio"/> Deplorable	<input type="radio"/> Aparente	<input type="radio"/> Forastero
ETERNO	<input type="radio"/> Perpetuo	<input type="radio"/> Estudioso	<input type="radio"/> Peculiar	<input type="radio"/> Modesto
ESENCIAL	<input type="radio"/> Lastimoso	<input type="radio"/> Indispensable	<input type="radio"/> Valeroso	<input type="radio"/> Variable
APOCADO	<input type="radio"/> Hosco	<input type="radio"/> Exótico	<input type="radio"/> Tímido	<input type="radio"/> Acerbo
SISTEMATICO	<input type="radio"/> Oculto	<input type="radio"/> Grave	<input type="radio"/> Principal	<input type="radio"/> Ordenado
BENEVOLO	<input type="radio"/> Reciente	<input type="radio"/> Diminuto	<input type="radio"/> Santo	<input type="radio"/> Amable
DISCRETO	<input type="radio"/> Prudente	<input type="radio"/> Amorfo	<input type="radio"/> Sombrio	<input type="radio"/> Femenino
FLUIDO	<input type="radio"/> Doméstico	<input type="radio"/> Importante	<input type="radio"/> Fidedigno	<input type="radio"/> Líquido
FAMOSO	<input type="radio"/> Ciudadano	<input type="radio"/> Accidental	<input type="radio"/> Célebre	<input type="radio"/> Aleveso

Figura 6.17. PMA (Factor V).

➤ Factor E: Concepción espacial

Es la capacidad para imaginar y concebir objetos en dos o tres dimensiones.

La prueba, por defecto, consta de 5 elementos que concierne al nivel 1 pero al igual que el factor anterior, se puede configurar en varios niveles, hasta un máximo de 4. Cada nivel tiene $5 * n^{\circ}$ nivel elementos. Cada elemento presenta un modelo geométrico plano y seis figuras similares.

El sujeto debe determinar cuáles de estas últimas, presentadas en distintas posiciones, coinciden con el modelo aunque hayan sufrido alguna rotación sobre el mismo plano. Para ello, el alumno deberá clicar sobre todas las figuras que cumpla la condición para pertenecer a la solución.

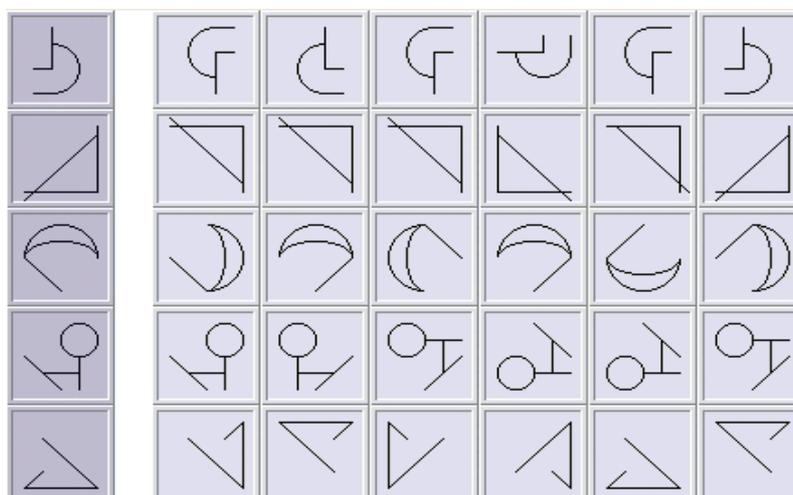


Figura 6.18. PMA (Factor E).

➤ Factor R: Razonamiento

Es la capacidad para resolver problemas lógicos, comprender y planificar. Investigaciones recientes muestran que el razonamiento implica dos capacidades diferentes: una, inductiva, la aptitud para inferir de los casos particulares la norma general, y otra, deductiva, la capacidad de extraer de las premisas la conclusión lógica. La prueba presente es una exploración de ambas aptitudes.

Esta prueba, al igual que los anteriores factores, está configurada por defecto para el nivel 1 y consta de 10 elementos. El máximo nivel para este factor es de 10; cada nivel se compone de $10 \cdot n^\circ$ nivel elementos.

El sujeto debe determinar la letra que prolongaría la serie que se le muestra, y una vez averiguada la relación lógica que las vincula, pulsar sobre ella.



Figura 6.19. PMA (Factor R).

➤ Factor N: Cálculo numérico

Es la capacidad de manejar números, de resolver rápidamente y con acierto problemas simplemente cuantitativos.

Esta prueba consta de 6 elementos o problemas, estos elementos son sumas de cuatro cifras de dos dígitos cada una.

El sujeto debe determinar si la suma de cuatro números de dos dígitos cada uno está bien o mal hecha, y cuando lo resuelva pulsar sobre el panel de la solución, que es el que está a la derecha de las sumas, la opción **B**(ien) o **M**(al) de la suma que corresponda.

(1)	(2)	(3)	(4)	(5)	(6)	
16	22	76	11	37	56	(1) <input type="radio"/> B <input type="radio"/> M
48	74	20	30	57	22	(2) <input type="radio"/> B <input type="radio"/> M
83	36	86	78	73	49	(3) <input type="radio"/> B <input type="radio"/> M
60	62	22	72	97	90	(4) <input type="radio"/> B <input type="radio"/> M
<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	(5) <input type="radio"/> B <input type="radio"/> M
207	194	164	169	264	217	(6) <input type="radio"/> B <input type="radio"/> M

Figura 6.20. PMA (Factor N).

➤ Factor F: Fluidez verbal

Es la capacidad para hablar y escribir con facilidad. Los sujetos a quienes les fluyen las palabras a la mente con prontitud y de corrido poseen el factor F en alto grado.

Para la exploración de este factor, la prueba pide a los sujetos que escriban palabras que empiecen por una determinada letra.

La aplicación de la prueba puede ser tanto colectiva como individual y la edad de aplicación del test es de 10/11 años en adelante.

6.2.7 Tests de resistencia a la fatiga.

Este tipo de tests, ponen a prueba nuestra capacidad para realizar tareas monótonas pero que no son totalmente mecánicas y que requieren de toda nuestra atención. Es importante estar relajados y concentrados mientras la realizamos ya que si nos ponemos nerviosos podemos echar al traste toda la prueba.

Esta prueba está formada por filas de 25 números del 0 al 9. En cada ejecución de esta aplicación se construye estas líneas, con números aleatorios comprendidos entre 0 y 9, obteniéndose en cada una de ellas sus correspondientes soluciones. A su vez estas filas de números se agrupan de 10 en 10 dependiendo del nivel en el que se configure la aplicación, así pues para un nivel 2 el número de filas sería de 20 sumándole dos filas más que son las *filas guía* para la numeración de las columnas (ver figura 6.21). Como hemos dicho esta aplicación está diseñada en varios niveles hasta un máximo de 10.

Numeración de las columnas para un nivel 2.

Casilla de la solución.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
0	7	1	2	6	7	0	1	5	8	1	3	1	8	5	9	0	1	2	0	2	5	9	8	3	
7	7	7	9	2	0	5	4	9	2	8	5	8	5	3	5	2	2	3	1	1	1	1	2	4	
9	8	3	7	9	1	4	1	1	6	4	5	0	6	0	7	4	4	8	7	7	2	8	1	5	
8	6	6	2	7	7	3	7	1	0	0	5	4	7	7	6	1	1	0	7	1	1	8	8	7	
4	1	5	2	8	4	0	5	6	2	4	4	3	2	7	7	0	0	3	5	0	5	0	8	8	
9	7	8	5	8	4	8	3	6	7	1	1	8	8	8	8	2	6	7	3	8	2	3	5	3	
3	2	6	4	4	5	1	0	0	6	5	3	8	5	9	7	7	6	6	9	2	7	0	6	2	
4	7	2	7	3	5	7	2	5	1	3	7	2	0	7	3	7	6	9	5	2	8	7	5	4	
1	3	8	9	9	8	8	7	5	3	4	3	6	6	4	4	1	3	4	3	3	7	4	6	1	
7	9	5	4	8	8	9	7	6	3	4	3	3	2	4	2	6	8	7	2	8	5	4	6	6	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
5	2	2	3	3	9	4	3	5	1	4	2	1	7	3	3	1	3	4	3	5	8	7	7	8	
2	5	0	1	2	3	7	3	2	8	6	2	8	9	5	2	8	4	4	0	2	1	2	3	0	6, 7, 9, 12, 16
1	3	2	7	3	4	8	5	1	1	3	3	8	7	3	8	7	8	5	8	1	2	7	7	0	
7	6	1	2	1	5	4	3	7	6	7	4	4	8	5	2	8	7	8	6	5	6	8	1	0	
2	3	4	7	4	8	3	1	1	7	0	6	1	6	7	1	4	2	2	3	7	5	5	7	1	
2	7	2	8	7	9	6	0	6	6	8	6	8	6	6	3	5	7	0	3	7	1	4	5	3	
4	4	5	0	2	8	7	0	2	4	1	0	4	4	5	9	6	9	1	8	0	2	8	2	2	
3	3	9	6	5	8	2	1	6	3	2	8	4	7	6	6	7	9	6	1	5	2	7	3	2	
1	2	8	3	4	5	0	0	8	8	1	0	3	8	7	7	7	0	9	8	8	0	5	1	3	
5	1	2	4	8	2	3	2	5	1	4	9	7	6	3	4	1	8	5	2	4	7	4	5	2	

Solución a esta fila

Figura 6.21. Test de resistencia a la fatiga de un nivel 2.

El objetivo es encontrar las parejas de números consecutivos que sumen 10. La solución se anotará en la casilla que aparece en la derecha de cada fila (ver figura 6.21), en ella escribiremos el número de la columna que ocupa el primer número de la pareja que sumen 10, y si encontramos varias parejas, separaremos los números con comas.

6.2.8 Test de Atención y Percepción.

Estos tests miden capacidades tales como la rapidez al hacer una tarea de forma precisa y la resistencia al desgaste, ya que nos obliga a realizar tareas rutinarias pero que requieren gran nivel de atención y concentración con la dificultad añadida de la limitación temporal. Es importante estar relajados y concentrados mientras la realizamos para que la prueba dé buenos resultados.

En esta prueba distinguiremos tres partes :

- una primera columna donde se ubica una serie de casillas que contiene cada una de ellas una serie de letras de longitud variable,
- una segunda columna donde estará la solución que el usuario obtenga,

- y por último otra columna semejante a la primera pero a la que le cambiaremos, o no una letra de esa serie de letras.

Una vez definidas las partes que componen nuestra prueba vamos a explicar en que consiste el test: tendremos que comprobar si las palabras que aparecen a derecha e izquierda de cada fila son iguales o distintas y anotaremos la solución en la columna central.

Al igual que el test anteriormente descrito esta prueba puede presentar varios niveles, dependiendo del profesor en qué nivel la configure. Para un nivel i , el número de filas de casillas que se presenta es $10 * i$. A continuación mostraremos una posible visualización que nos proporcionaría el sistema:

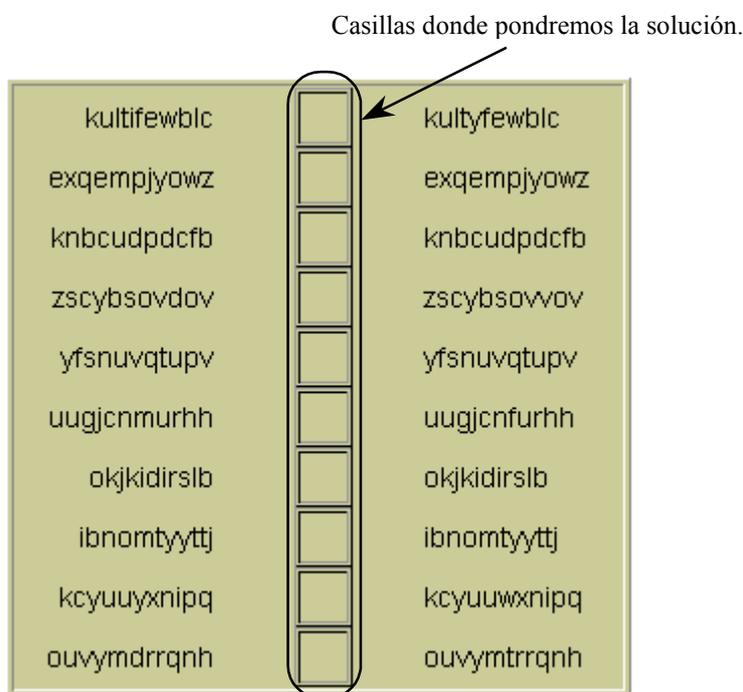


Figura 6.22 Test de Atención y Percepción (Nivel 1).

El objetivo de esta prueba es la de averiguar si la cadena de letras que aparece a la izquierda es la misma que la de la derecha, y dependiendo de que la solución sea o no igual, se establecerá en la botonera central pulsando sobre ella, la figura 6.23 para una solución correcta o la figura 6.24 para una solución incorrecta.



Figura 6.23. Opción para una respuesta correcta.



Figura 6.24. Opción para una respuesta incorrecta.

Capítulo 7.

Manual de referencia.

En el capítulo siguiente explicaremos la instalación de los applets (directorios, path), la configuración (parámetros al applet, archivos de configuración, ...) y por últimos las modificaciones del código fuente para futuras creaciones de nuevos applets.

7.1 Instalación.

Para la correcta integración de los tests psicotécnicos en SIETTE y su puesta en marcha son necesarios que los applets que implementan a éstos, estén ubicados en unos determinados directorios. Del mismo modo ocurre con las imágenes que se utilizan en los diferentes applets para su presentación. Vamos a detallar estos requisitos con más detalle.

La URL básica de donde cuelgan los directorios de los applets es la siguiente: <http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/applets/>. Estos directorios se llaman igual que el nombre del applet que contiene, por ejemplo el directorio que contiene al applet IG tendrá esta URL <http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/applets/IG>.

Como dijimos antes, las imágenes que usan los diferentes test están también clasificadas en distintos directorios dependiendo en que test se utilice. La URL base desde dónde cuelga todas las imágenes es la que a continuación se detalla <http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/images/>, a partir de ella se han creado directorios con los nombre de los applets con cuyo contenido serán esas imágenes que use ese applet en concreto. Existe otro directorio en el que contiene imágenes que es de uso general

para todos y cada uno de los applets, en él están almacenados los iconos que se muestran en el panel del enunciado al inicio de cada prueba. Este directorio es el siguiente <http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/images/general>.

La tabla 7.1 refleja la ubicación de los directorios donde está almacenado el applet de cada test psicotécnico y de sus imágenes.

Test de Dominós	
Directorio del applet	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/applets/IG
Directorio de las imágenes	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/images/IG
Test Cuadrado de letras	
Directorio del applet	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/applets/CuadradoDeLetras
Directorio de las imágenes	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/images/CuadradoDeLetras
Test de las Figuras Iguales o B.G.3	
Directorio del applet	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/applets/BG3
Directorio de las imágenes	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/images/BG3
Test de Razonamiento y Estructuración Espacial o B.43	
Directorio del applet	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/applets/B43
Directorio de las imágenes	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/images/B43
Test Espacio-Perceptivo o Situacion1	
Directorio del applet	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/applets/Situacion1
Directorio de las imágenes	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/images/Situacion1

Test de Aptitudes Mentales Primaria	
Directorio del applet	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/applets/PMA
Directorio de las imágenes	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/images/PMA
Test de Resistencia a la Fatiga	
Directorio del applet	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/applets/ResistenciaFatiga
Directorio de las imágenes	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/images/ResistenciaFatiga
Test de Atención y Percepción	
Directorio del applet	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/applets/AtencionPercepcion
Directorio de las imágenes	http://sirius.lcc.uma.es/siette/asiganturas/psicotecnico/images/AtencionPercepcion

Tabla 7.1. Directorios de instalación.

7.2 Configuración.

En esta apartado puntualizaremos los parámetros que necesitamos introducirle a los applets para el correcto funcionamiento de los mismos. Del mismo modo, también detallaremos si el applet que estemos tratando necesita o de algún fichero de configuración y cual sería su estructura.

7.2.1 Configuración del test de Dominós.

Para el test de Dominós estos son los parámetros que debemos pasarle al applet para una ejecución correcta de la aplicación.

Parámetros	Definición
ClaseDeFigura	Nombre de la clase Java que extiende de la clase abstracta Figura que se ha implementado para el manejo y presentación de las fichas. Por defecto la nombre de la clase es Ficha (Ficha.class) que se adjunta con el applet.
FicheroPlantillas	Path completo donde está ubicado el fichero de plantillas incluyendo el nombre del fichero.
FicheroModelos	Path completo donde esta archivado el fichero de modelos.
Plantillas	Número de la plantilla a presentar.
PasoAleatorio	Valor que se suma a cada componente de las fichas de una plantilla (incluida la solución) para generar otra nueva.
ColorFondoFiguraVacía	Color del fondo de la figura vacía o solución (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
ColorFondoFiguraRellena	Color del fondo de la figura rellena (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
AnchoMaximoPlantillas	Ancho máximo del modelo de plantilla que se va a presentar.
Enunciado	Enunciado de la pregunta.
TamañoFuenteEnunciado	Tamaño de la letra del enunciado, por defecto es 12.
ColorEnunciado	Color del fondo del enunciado (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
AnchoEnunciado	Ancho del enunciado.
AltoEnunciado	Alto del enunciado.
ColorPanelBotones	Color del fondo del panel que contiene a los botones.

Tabla 7.2. Definición de parámetros para el applet IG.

Además de estos parámetros, esta aplicación maneja un fichero que es la base para la creación de nuevas plantillas (parámetro [FicheroPlantillas](#)) y otro en el que se especifica la configuración de los modelos de representación visual de las plantillas sobre la pantalla (parámetro [FicheroModelos](#)). Estos ficheros atiende a una estructura de definición específica y que se debe seguir estrictamente.

La aplicación incorpora unos ficheros de configuración por defecto, éstos están almacenados en el directorio “*config*” que está en el directorio base de esta aplicación. Estos ficheros de llaman *DefPlan.ig* y *DefMod.ig*, fichero de plantillas y modelos respectivamente.

El fichero de plantilla debe obedecer a la siguiente estructura que se detalla en la siguiente tabla, pero antes de nada debemos decir que las palabras en azul son palabras reservadas, las que están en negro es el texto que hay que sustituir por el adecuado en cada caso y el signo de separación de elementos es “:” y está en rojo:

Para poner un comentario:	Ejemplo:
Se definen comentarios orientado a línea o a más de una línea.	//Comentario para una sola línea /* Comentario de más de una línea */
Para definir una plantilla:	Ejemplo:
<p>PlantillaNumeroDeLaPlantilla:a:b a = número de fichas que compone la plantilla. b = modelo de representación. Modo de representación: Lineal, LinealDoble, LinealTriple, Estrella, Cruz, AlternadaVertical, AlternadaHorizontal.</p> <p>Una vez definida una plantilla se puede que definir los siguientes elementos:</p>	<p>Plantilla1:6:Lineal</p> <p>Luego debajo se definen las 6 fichas y la solución.</p>
Para definir una ficha rellena	Ejemplo:
FiguraRellena:valor1:valor2: ... :valorN	FiguraRellena:5:2 Ficha con dos componentes.
Para definir una figura vacía	Ejemplo:
FiguraVacía Simboliza la representación visual de la ficha solución sobre la pantalla.	FiguraVacía
Para definir la solución de la plantilla:	Ejemplo:
Solución:valor1:valor2: ... :valorN	Solución:5:5

Tabla 7.3. Definición de plantillas.

Para que sirva de ejemplo, a continuación exponemos un fragmento del contenido del fichero de plantillas *DefPlan.ig*.

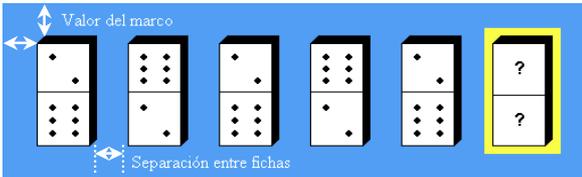
```

...
...
/*Definición de la plantilla número 15 con
modelo de representación en Estrella*/
//Plantilla 15
Plantilla15:8:Estrella
FiguraRellena:0:3
FiguraRellena:4:5
FiguraRellena:2:6
FiguraRellena:1:4
FiguraRellena:1:4
FiguraVacía
FiguraRellena:4:5
FiguraRellena:0:3
Solucion:2:6
...
...

```

Figura 7.1. Definición de la plantilla número 15 en el fichero *DefPlan.ig*

Al igual que el fichero de plantillas, el fichero de los modelos también tiene que obedecer una estructura especial para definir los siguientes parámetros:

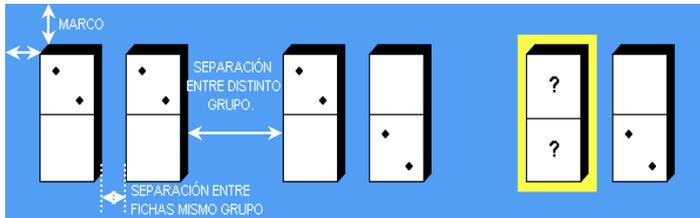
Parámetros	Ejemplo:
<code>NumeroDeComponentesDeLaFigura:n°_componentes</code>	<code>NumeroDeComponentesDeLaFigura:2</code>
<p><code>Lineal:a:b</code></p> <p>a = separación de las fichas respecto al marco.</p> <p>b = separación entre fichas.</p> 	<code>Lineal:30:30</code>

LinealDoble:a:b:c

a = separación de las fichas respecto al marco.

b = separación entre fichas del mismo grupo.

c = separación entre fichas de distinto grupo.



LinealDoble:30:20:80

LinealTriple:a:b:c

Igual que en el parámetro lineal doble.

LinealTriple:7:60:90

Estrella:(x1,y1):(x2,y2): ... :(xi, yi): ... :(x10,y10)

Dimension:anchoPlantilla:altoPlantilla

El eje de coordenadas como es habitual en informática están situados en la esquina superior izquierda del applet. Estas coordenadas son absolutas y cada una de ellas corresponde a la coordenada de la esquina superior izquierda de la figura i-ésima de la plantilla definida como modo de representación estrella.

Estrella:(117,7):(177,7):(237,117):
(237,177):(177,237):(117,237):
(7,177):(7,117):(117,120):
(177,120)

Dimension:347:347

Cruz:(x1,y1):(x2,y2): ... :(xi, yi): ... :(x6,y6)

Dimension:anchoPlantilla:altoPlantilla

Ídem que el anterior pero para una plantilla definida como modo de representación cruz.

Cruz:(72,7):(187,67):(72,177):
(7,67):(67,67):(127,67)

Dimension:247:237

AlternadaHorizontal:a:b:c

a = separación de las fichas respecto al marco.

b = coordenada y respecto al origen (0,0) de la figura horizontal.

c = separación entre fichas.

AlternadaHorizontal:7:56:7

<p>AlternadaVertical:a:b:c</p> <p>a = separación de las fichas respecto al marco.</p> <p>b = coordenada y respecto al origen (0,0) de la figura horizontal.</p> <p>c = separación entre fichas</p>	<p>AlternadaVertical:7:56:7</p>
---	--

Tabla 7.4. Definición de la estructura del fichero de configuración de modelos.

A continuación mostramos el contenido del fichero de modelos *DefMod.ig*. En el archivo también se puede realizar comentarios.

```

/*Fichero de modelos y definición del número de componentes de la figura.*/
NumeroDeComponentesDeLaFigura:2
Lineal:7:7
LinealDoble:7:7:67
LinealTriple:7:7:33
Estrella:(117,7):(177,7):(237,117):(237,177):(177,237):(117,237):(7,177):(7,117):(117,120):(177,120)
Dimension:347:347
Cruz:(72,7):(187,67):(72,177):(7,67):(67,67):(127,67)
Dimension:247:237
AlternadaHorizontal:7:56:7
AlternadaVertical:7:56:7
    
```

Figura 7.2. Fichero de modelos *DefMod.ig*.

7.2.2 Configuración del test Cuadrados de letras.

Para el test de Cuadrados de letras estos son los parámetros que debemos pasarle al applet para una actuación correcta de la aplicación.

Parámetros	Definición
ColorHabilitado	Color del fondo de la figura cuando no está seleccionada (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
ColorDeshabilitado	Color del fondo de la figura cuando está seleccionada (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
Enunciado	Enunciado de la pregunta.
TamañoFuenteEnunciado	Tamaño de la letra del enunciado, por defecto es 12.
ColorEnunciado	Color del fondo del enunciado (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
AnchoEnunciado	Ancho del enunciado.
AltoEnunciado	Alto del enunciado.

ColorPanelBotones	Color del fondo del panel que contiene a los botones.
-----------------------------------	---

Tabla 7.5. Definición de parámetros para el applet Cuadrado de Letras.

7.2.3 Configuración del test de las Figuras iguales o B.G.3.

Para el test de las figuras iguales o B.G.3 estos son los parámetros que debemos comunicar al applet para su ejecución.

Parámetros	Definición
ColorHabilitado	Color del fondo de la figura cuando no está seleccionada (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
ColorDeshabilitado	Color del fondo de la figura cuando está seleccionada (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
Enunciado	Enunciado de la pregunta.
TamañoFuenteEnunciado	Tamaño de la letra del enunciado, por defecto es 12.
ColorEnunciado	Color del fondo del enunciado (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
AnchoEnunciado	Ancho del enunciado.
AltoEnunciado	Alto del enunciado.
ColorPanelBotones	Color del fondo del panel que contiene a los botones.

Tabla 7.6. Definición de parámetros para el applet B.G.3.

7.2.4 Configuración del test de Razonamiento y Estructuración Espacial o B.43.

Para el test de razonamiento y estructuración espacial o B.43 estos son los parámetros que debemos notificar al applet para la perfecta ejecución de la aplicación.

Parámetros	Definición
NumeroDelPuzzle	Número del puzzle a presentary definido en el fichero de plantillas.
MostrarBordePuzzle	Parámetro que decide si mostrar el borde de las figuras que componen el puzzle (0: No y 1:Si).
GrupoPiezas	Parámetro que establece el numero de piezas que compone una columna. Por defecto es 5.
FicheroFiguras	Url completa donde está ubicado el fichero de figuras.
FicheroPlantillas	Url completa donde está ubicado el fichero de plantillas incluyendo el nombre del fichero.
ColorFondo	Color del fondo de la prueba (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
Enunciado	Enunciado de la pregunta.
TamañoFuenteEnunciado	Tamaño de la letra del enunciado, por defecto es 12.
ColorEnunciado	Color del fondo del enunciado (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
AnchoEnunciado	Ancho del enunciado.
AltoEnunciado	Alto del enunciado.
ColorPanelBotones	Color del fondo del panel que contiene a los botones.

Tabla 7.7. Definición de parámetros para el applet B.43.

Además de estos parámetros, esta aplicación maneja un fichero que es el que tiene almacenado las características de las figuras a mostrar (parámetro [FicheroFiguras](#)) y otro en el que se especifica la definición y composición de las plantillas o puzzles que se van a presentar

sobre la pantalla (parámetro [FicheroPlantillas](#)). Estos ficheros atiende a una estructura de definición específica y que se debe seguir estrictamente.

La aplicación incorporada unos ficheros de configuración por defecto, éstos están almacenados en el directorio “*config*” que está en el directorio base de esta aplicación. Estos ficheros de llaman *DefPlant.b43* y *DefFiguras.b43*, fichero de plantillas y figuras respectivamente.

El fichero de figuras debe obedecer a la siguiente estructura que se detalla en la siguiente tabla, pero antes de nada debemos decir que las palabras en azul son palabras reservadas, las que están en negro es el texto que hay que sustituir por el adecuado en cada caso y el signo de separación de elementos es “:” y está en rojo:

Para poner un comentario:	Ejemplo:
Se definen comentarios orientado a línea o a más de una línea.	//Comentario para una sola línea /* Comentario de más de una línea */
Para definir el número de figuras a mostrar:	Ejemplo:
NumeroDeFiguras : <i>NºdeFigurasADefinir</i>	NumeroDeFiguras :25
Para definir una figura:	Ejemplo:
Figura <i>NºFigura</i> Cabecera para definir la figura número <i>NºFigura</i> y a continuación se meten su parámetros que son los que se detallan ahora.	Figura 1

<p>Una vez definida una figura se puede que definir los siguientes elementos:</p>	
<p>Para definir un alias a la figura</p>	<p>Ejemplo:</p>
<p><code>alias:no nombreAlias</code></p> <p>Se define un alias para una figura, cuando hay un conjunto de ellas que son iguales y se pueden usar indistintamente. Todas ellas tendrían el mismo alias. Valores que puede tomar:</p> <ul style="list-style-type: none"> * <code>no</code> * nombre que se elija para definir a ese grupo de figuras iguales. 	<p><code>alias:no</code></p> <p>o</p> <p><code>alias:triangulo</code></p>
<p>Para definir si rotar o no las figuras</p>	<p>Ejemplo:</p>
<p><code>rotacionesIguales:si no</code></p> <p>Definir si en una figura sus rotaciones son iguales, p.e.: el cuadrado al rotarlo es igual si no se hubiese rotado. Valores que puede tomar:</p> <ul style="list-style-type: none"> * <code>si</code> * <code>no</code> 	<p><code>rotacionesIguales:si</code></p> <p>o</p> <p><code>rotacionesIguales: no</code></p>
<p>Para definir las dimensiones de la figura en cada rotación</p>	<p>Ejemplo:</p>
<p><code>0:anchoRotacion0grados:altoRotacion0grados</code></p> <p><code>1:anchoRotacion90grados:altoRotacion90grados</code></p> <p><code>2:anchoRotacion180grados:altoRotacion180grados</code></p> <p><code>3:anchoRotacion270grados:altoRotacion270grados</code></p> <p>Define el ancho y alto de la figura con una de rotación 0, 90, 180 y 270 grados.</p>	<p><code>0:41:61</code></p> <p><code>1:61:41</code></p> <p><code>2:41:61</code></p> <p><code>3:61:41</code></p>

Tabla 7.8. Definición de figuras.

A continuación y para que sirva de ejemplo, mostraremos un fragmento del contenido del fichero *DefFiguras.b43* para la definición de algunas figuras.

```

... ..
/*Definición del cuadrado*/
Figura15
alias:no
rotacionesIguales:si
0:61:61
1:61:61
2:61:61
3:61:61
... ..
Figura18
alias:tetris
rotacionesIguales:no
0:41:61
1:61:41
2:41:61
3:61:41
... ..
Figura19
alias:tetris
rotacionesIguales:no
0:41:61
1:61:41
2:41:61
3:41:41
... ..

```

Figura 7.3. Fichero de definición de figuras *DefFigura.b43*

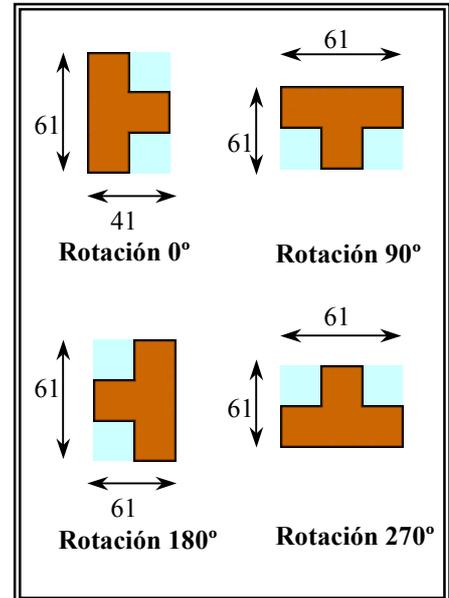


Figura 7.4. Anchura y altura de la figura 19 en sus distintas rotaciones.

En el fichero de definición de las plantillas también se ha establecido una manera muy similar de especificar los puzzles que se van a mostrar al alumno. La leyenda de colores es la misma que la del fichero de configuración de figuras, y la estructura de definición es la detallada en la siguiente tabla.

Para poner un comentario:	Ejemplo:
Se definen comentarios orientado a línea o a más de una línea.	//Comentario para una sola línea /* Comentario de más de una línea */
Para definir una plantilla:	Ejemplo:
PlantillaNumeroDeLaPlantilla:a:b:c a = número de fichas que compone la plantilla. b = ancho del puzzle. c = alto del puzzle.	Plantilla1:6:101:61 Luego debajo se definen las 6 fichas que componen el puzzle.
Para definir una componente del puzzle:	Ejemplo:
NombreFiguraOalias:rotacion:(x,y) donde NombreFiguraOalias = es el nombre de la figura o el alias que se ha especificado en el fichero de figuras.	figura3:3:(20,0) o cuadradoRoto:2:(0,40)

rotacion = la rotación en la que quedaría la figura en el puzzle. Valores: 0 (0°), 1 (90°), 2 (180°), 3 (270°).

(x,y) = coordenada dentro del puzzle donde se ubicaría la esquina superior izquierda del mínimo rectángulo que engloba a la figura que estamos incluyendo.

Tabla 7.9. Definición de las plantillas o puzzles.

Ejemplo de definición de la plantilla número 18 en el fichero *DefPlant.b43*:

```
/*Definición de la plantilla
número 8*/
//Plantilla 8
Plantilla8:4:81:101
figura21:0:(0,0)
figura23:2:(20,0)
figura25:3:(40,20)
figura11:3:(20,40)
```

- 1 → Define la figura21 con rotación de 0° (0) y la ubicamos en el punto (0,0).
- 2 → Define la figura23 con rotación de 180° (2) y la ubicamos en el punto (20,0).
- 3 → Define la figura25 con rotación de 270° (3) y la ubicamos en el punto (40,20).
- 4 → Define la figura11 con rotación de 270° (3) y la ubicamos en el punto (20,40).

Figura 7.5. Definición de una plantilla.

7.2.5 Configuración del test Espacio-Perceptivo o Situacion-1.

Para la creación de una instancia de la aplicación que muestra un ejercicio del test Situacion-1 hay que pasar unos parámetros al applet para el perfecto funcionamiento, y son los que se especifican en la siguiente tabla.

Parámetros	Definición
OrientacionFlecha	Número entre 1 y 8 ambos inclusive que especifica la orientación de la flecha, donde 1: VA, 2:AD, 3:HD, 4:BD, 5:VB, 6:BI, 7:HI, 8:AI.
AnchoCuadro	Ancho del marco que contiene al eje donde se establecerá la orientación la flecha.
AltoCuadro	Ancho del marco que contiene al eje donde se establecerá la orientación la flecha.

ColorCuadro	Color del fondo del cuadro que contiene al eje y a la flecha (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
ColorFlecha	Color de la flecha en la solución (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
ColorFlechaUsuario	Color del flecha del alumno (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
ColorEje	Color del eje de la solución (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
ColorEjeUsuario	Color del eje del alumno (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
Enunciado	Enunciado de la pregunta.
TamañoFuenteEnunciado	Tamaño de la letra del enunciado, por defecto es 12.
ColorEnunciado	Color del fondo del enunciado (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
AnchoEnunciado	Ancho del enunciado.
AltoEnunciado	Alto del enunciado.
ColorPanelBotones	Color del fondo del panel que contiene a los botones.

Tabla 7.10. Definición de parámetros para el applet Situacion-1.

7.2.6 Configuración del test de Aptitudes Mentales Primarias o PMA.

Parámetros que hay que pasarle a la aplicación de muestra una pregunta del test de aptitudes mentales primarias para una correcta ejecución.

Parámetros	Definición
Tipo	Tipo de factor a examinar (FactorR, FactorN, FactorE, FactorV).
Nivel	Nivel de la pregunta que estemos creando, dependerá del factor (ver en el capítulo 6 el apartado 6.2.6).
PorcentajeAcierto	Establecerá el porcentaje de preguntas acertadas.
TenerEnCuentaFallos	Determinará si a la hora de dar la solución, tendrá en cuenta los fallos cometidos por el alumno. Valores posibles “Si” o “No”.

ColorFondo1	Color del fondo de la pregunta (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
ColorFondo2	Color del fondo de las posibles soluciones (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
Enunciado	Enunciado de la pregunta.
TamañoFuenteEnunciado	Tamaño de la letra del enunciado, por defecto es 12.
ColorEnunciado	Color del fondo del enunciado (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
AnchoEnunciado	Ancho del enunciado.
AltoEnunciado	Alto del enunciado.
ColorPanelBotones	Color del fondo del panel que contiene a los botones.

Tabla 7.11. Definición de parámetros para el applet PMA.

7.2.7 Configuración del test de Resistencia a la fatiga.

Parámetros de entrada al applet para la visualización de una cuestión del test de resistencia a la fatiga.

Parámetros	Definición
Nivel	Nivel de la pregunta.
TenerEnCuentaFallos	Determinará si a la hora de dar la solución, tendrá en cuenta los fallos cometidos por el alumno. Valores posibles “ <i>Si</i> ” o “ <i>No</i> ”.
ColorFondo	Color del fondo de la pregunta (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
ColorNumeros	Color del fondo de la línea de números (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
Enunciado	Enunciado de la pregunta.
TamañoFuenteEnunciado	Tamaño de la letra del enunciado, por defecto es 12.
ColorEnunciado	Color del fondo del enunciado (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
AnchoEnunciado	Ancho del enunciado.

AltoEnunciado	Alto del enunciado.
ColorPanelBotones	Color del fondo del panel que contiene a los botones.

Tabla 7.12. Definición de parámetros para el applet ResistenciaFatiga.

7.2.8 Configuración del test de Atención y Percepción.

Para el test de atención y percepción estos son los parámetros que debemos comunicar al applet para su ejecución.

Parámetros	Definición
Nivel	Nivel de la pregunta.
TenerEnCuentaFallos	Determinará si a la hora de dar la solución, tendrá en cuenta los fallos cometidos por el alumno. Valores posibles “ Si ” o “ No ”.
ColorFondo	Color del fondo de la pregunta (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
Enunciado	Enunciado de la pregunta.
TamañoFuenteEnunciado	Tamaño de la letra del enunciado, por defecto es 12.
ColorEnunciado	Color del fondo del enunciado (6 dígitos hexadecimales: 2 para rojo, 2 para verde, 2 para azul).
AnchoEnunciado	Ancho del enunciado.
AltoEnunciado	Alto del enunciado.
ColorPanelBotones	Color del fondo del panel que contiene a los botones.

Tabla 7.13. Definición de parámetros para el applet AtencionPercepcion.

Capítulo 8.

Conclusiones y trabajos futuros.

En el siguiente capítulo se va a hacer un análisis de lo que ha sido la realización del proyecto. Una vez finalizado, es necesario realizar un estudio de éste para poder obtener conclusiones o resultados que hagan entender mejor lo que se ha conseguido, y a la vez, obtener una posible proyección del mismo.

8.1 Conclusiones.

Este proyecto ha sido realizado con el objeto de ofrecer una biblioteca de test psicotécnicos, cuyo funcionamiento resulte cómodo y sencillo. En líneas generales se han cumplido los objetivos que se marcaron inicialmente e incluso se han incorporado algunas mejoras. Los capítulos precedentes son el resultado de un detallado trabajo sobre el diseño de la biblioteca.

Para llevar a cabo este proyecto ha sido necesario realizar una amplia búsqueda bibliográfica sobre diversos libros específicos dedicados a este tema, así como una minuciosa búsqueda en la red Internet, con la intención de que sirva de ayuda en la realización de tests psicotécnicos.

El objetivo que nos habíamos marcado con el desarrollo de este proyecto era la realización de un entorno de aprendizaje, accesible desde la World Wide Web, para que los alumnos consiguieran realizar tests psicotécnicos, pudiendo obtener una autoevaluación de los tests realizados. Para dicha autoevaluación se pensó en la de crear una biblioteca de applets en la que el profesor pudiera introducir tests específicos dentro del sistema de TeleEducación SIETTE.

Para poder lograr estos objetivos planteados, hemos utilizados las tecnologías más avanzadas punteras que existen actualmente para el desarrollo de aplicaciones. El lenguaje Java nos proporciona todo lo necesario para desarrollar la aplicación deseada, destacando la característica principal de portabilidad de la aplicación a cualquier plataforma.

Para concluir podemos decir que serán los usuarios los que juzguen si verdaderamente la biblioteca resulta una herramienta eficaz para realizar tests psicotécnicos para la evaluación de la atención y percepción y de la inteligencia en general.

8.2 Trabajos futuros.

En el campo de los tests psicotécnicos hay una infinidad de pruebas y continuamente están desarrollándose otras nuevas, que pueden ser implementadas en algún lenguaje. Por tanto el número de tests que compone la biblioteca se puede ampliar con las pruebas que vayan surgiendo o de las que ya existen y todavía no hayan sido implementadas.

Por otro lado cada vez surgen nuevas herramientas en las aplicaciones gráficas, haciendo más fácil su uso o bien con nuevas características. La interfaz aunque es bastante completa siempre se le puede añadir nuevos objetos.

Bibliografía

- [1] Laurence Genain, Martine Lerond. “*Los tests psicológicos*”, Editorial De Vecchi, S.A. Barcelona 1991. Colección psicología.
- [2] Gilles Azzopardi. “*Los test de selección de personal: sus secretos y sus trampas*”, Ediciones Deusto S.A. Madrid 1993. Colección biblioteca Deusto de desarrollo personal.
- [3] A.Rios, E. Millán, M. Trella, J. Perez-de-la-Cruz, R. Conejo, Internet Based Evaluation System, en: ARTIFICIAL INTELLIGENCE IN EDUCATION, Open Learning Environments: New Computational Technologies to Support Learning, Exploration and Collaboration, Le Mans, France (1999) 387-394.
- [4] I. Arroyo, R. Conejo, E. Guzmán, B. P .Woolf, An Adaptive Web-based Component for Cognitive Ability Estimation, en: J.D. Moore, C. Luckhardt-Redfield, W. Lewis Johnson (Eds.), Artificial Intelligent in Education: AI-ED in the Wired and Wireless Future (IOS Press, Amsterdam , 2001) 456-466.
- [5] García Álvarez, A. “*HTML: Creación de páginas Web, 1ª edición*”. Ed. Anaya Multimedia,1997.
- [6] García, G. “*HTML: Iniciación y referencia*”. Ed. McGraw-Hill, 1997
- [7] Powell, T.: “*HTML: Manual de referencia*”. Ed. McGraw-Hill, 1998
- [8] Ian S. Graham. “*The HTML sourcebook: a complete guide to HTML 3.0 2ª Edition* ”. Ed. John Wiley & Sons, cop. New York 1996

- [9] Agustín Froufe. “*Java™ 2 Manual de usuario y tutorial 2ª Edición*”, Editorial Ra-Ma.
- [10] Javier García de Jalón, José Ignacio Rodríguez, Íñigo Mingo, Aitor Imaz, Alfonso Brazales, Alberto Larzabal, Jesús Calleja, Jon García. “*Aprenda Java como si estuviera en primero*”, Escuela Superior de Ingenieros Industriales de San Sebastián, Universidad de Navarra (Febrero 2000).
- [11] Eckel, B.: “*Thinking in Java*”. Ed. Prentice-Hall. 1998.
- [12] Lemay, L. “*Aprendiendo Java 2 en 21 días*”. Ed. Prentice-Hall. 1999
- [13] Nauhton, P y Schildt, H. “*Java: manual de referencia*”. Ed. McGraw-Hill 2000
- [14] Niemeyer, P y Knudsen, J. “*Curso de Java*”. Ed. Anaya O’reilly, 2000.
- [15] Patrick Naughton. “*The Java handbook*”. Ed. Osborne MacGraw-Hill. Berkeley 1996.