

**UNIVERSIDAD DE SAN CARLOS
FACULTAD DE INGENIERIA**



**IMPLEMENTACION DE UN MANEJADOR DE PRUEBAS
PARA LA EVALUACION Y MEDICION DEL
COEFICIENTE INTELECTUAL**

TESIS

PRESENTADA A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERIA

POR

Freidy Javier Gramajo López

AL CONFERIRLE EL TITULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

Guatemala de la Asunción, marzo de 1,995

PROPIEDAD DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
Biblioteca Central

08
T(3495)
C. 4

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

Escuelas de Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, Escuela Técnica, Ingeniería en Sistemas Ingeniería Electrónica y Escuela Regional de Ingeniería Sanitaria y Recursos Hidráulicos.
Apartado Postal 217-I-01-907, Guatemala
Ciudad Universitaria, Zona 12
Guatemala, Centroamérica

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de tesis titulado :

**IMPLEMENTACION DE UN MANEJADOR DE PRUEBAS
PARA LA EVALUACION Y MEDICION DEL
COEFICIENTE INTELECTUAL**

tema que me fuera asignado por la Coordinación de la Carrera de Ingeniería en Ciencias y Sistemas con fecha 12 de julio de 1,994.

[Handwritten signature]
Francisco Javier Guzmán López



FACULTAD DE INGENIERIA

Escuelas de Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, Escuela Técnica, Ingeniería en Sistemas Ingeniería Electrónica y Escuela Regional de Ingeniería Sanitaria y Recursos Hidráulicos.
Apartado Postal 217-I-01-907, Guatemala
Ciudad Universitaria, Zona 12
Guatemala, Centroamérica

Miembros de Junta Directiva

DECANO	Ing. Julio Ismael González Podszueck
VOCAL 1ero.	Ing. Miguel Angel Sánchez Guerra
VOCAL 2do.	Ing. Jack Douglas Ibarra Solórzano
VOCAL 3ero.	Ing. Juan Adolfo Echeverria Méndez
VOCAL 4to.	Freddy Rodríguez Quezada
VOCAL 5to.	Mario Nephtali Morales Solís
SECRETARIO	Ing. Francisco Javier Gonzalez López

Tribunal que práctico el EXAMEN GENERAL PRIVADO

DECANO	Ing. Julio Ismael González Podszueck
EXAMINADOR	Ing. Christian Bradna Villanueva
EXAMINADOR	Ing. Mario Enrique Sosa Castillo
EXAMINADOR	Ing. Carlos Alfredo Azurdia Morales
SECRETARIO	Ing. Francisco Javier Gonzalez López

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

Escuelas de Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, Escuela Técnica, Ingeniería en Sistemas Ingeniería Electrónica y Escuela Regional de Ingeniería Sanitaria y Recursos Hidráulicos.

Apartado Postal 217-I-01-907, Guatemala
Ciudad Universitaria, Zona 12
Guatemala, Centroamérica

Guatemala 20 de enero de 1,995

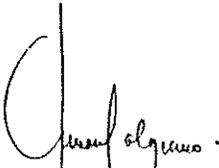
Ing. Calixto Raul Monzón
Coordinador de Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Estimado Ingeniero:

Por este medio me permito informarle que he procedido a revisar el trabajo de tesis titulado **IMPLEMENTACION DE UN MANEJADOR DE PRUEBAS PARA LA EVALUACION DEL COEFICIENTE INTELECTUAL**, elaborado por el estudiante Freiry Javier Gramajo López y a mi juicio, la misma cumple con los objetivos propuestos para su desarrollo.

Sin otro particular me suscribo de usted,

Atentamente,


Licda. Loris de Salguero
Asesora
Loris de Salguero
LICDA. DE PSICOLOGIA
CUL 1261

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

Escuelas de Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, Escuela Técnica, Ingeniería en Sistemas Ingeniería Electrónica y Escuela Regional de Ingeniería Sanitaria y Recursos Hidráulicos.
Apartado Postal 217-I-01-907, Guatemala
Ciudad Universitaria, Zona 12
Guatemala, Centroamérica

Guatemala 23 de enero de 1,995

Ing. Calixto Raul Monzón
Coordinador de Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Estimado Ing. Monzón:

Atentamente me dirijo a usted para informarle que he revisado la tesis titulada **IMPLEMENTACION DE UN MANEJADOR DE PRUEBAS PARA LA EVALUACION DEL COEFICIENTE INTELECTUAL**, del estudiante Freiry Javier Gramajo López, y la misma cumple con los objetivos propuestos.

Atentamente,

A handwritten signature in black ink, appearing to read 'C. Bradna'.

Ing. Christian Bradna Villanueva
Director de area de Privados y Públicos

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

Escuelas de Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, Escuela Técnica, Ingeniería en Sistemas Ingeniería Electrónica y Escuela Regional de Ingeniería Sanitaria y Recursos Hidráulicos.
Apartado Postal 217-1-01-907, Guatemala
Ciudad Universitaria, Zona 12
Guatemala, Centroamérica

Guatemala 28 de febrero de 1,995

Ingeniero
Julio Ismael Gonzáles Podszuek
Decano
Facultad de Ingeniería

Señor Decano:

Atentamente me dirijo a usted, para informarle que después de conocer el dictamen de la Asesora Licda. Loris de Salguero y del Director de área de Privados y Públicos Ing. Christian Bradna Villanueva, sobre el trabajo de tesis del estudiante Freiry Javier Gramajo López, titulado **IMPLEMENTACION DE UN MANEJADOR DE PRUEBAS PARA LA EVALUACION DEL COEFICIENTE INTELECTUAL**, procedo a la aprobación de dicha tesis.



Calixto Raúl Monzón P.
Ing. Calixto Raúl Monzón P.
Coordinador
Ingeniería en Ciencias y Sistemas

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

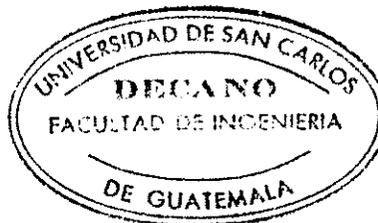
Escuelas de Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, Escuela Técnica, Ingeniería en Sistemas Ingeniería Electrónica y Escuela Regional de Ingeniería Sanitaria y Recursos Hidráulicos.
Apartado Postal 217-I-01-907, Guatemala
Ciudad Universitaria, Zona 12
Guatemala, Centroamérica

El Decano de la Facultad de Ingeniería, luego de conocer la autorización por parte del Coordinador de Ingeniería en Ciencias y Sistemas, al trabajo de tesis titulado: **IMPLEMENTACION DE UN MANEJADOR DE PRUEBAS PARA LA EVALUACION DEL COEFICIENTE INTELECTUAL**, del estudiante universitario Freiry Javier Gramajo López, procede a la autorización para la impresión de la misma.

IMPRIMASE

Ing. Julio Ismael González Podszueck
Decano de la Facultad de Ingeniería

Guatemala, marzo de 1,995.



ACTO QUE DEDICO A

MIS PADRES

Lic. Feliciano Isaiás Gramajo Calderón
Licda. Benilda López Pómes de Gramajo

Como reconocimiento a su lucha, profesionalismo y voluntad ante la vida.

MIS HERMANOS

Melvin Ruben Gramajo López, Elizabeth Recinos Estrada
Maidy Lisbeth Gramajo López de Martínéz
Ludwing Gramajo López
Frandy Gramajo López

Por su ayuda y comprensión.

FAMILIA OZAETA HALL

Berta Lidia Hall
Eddy, Norman, Aaron y Heydi

Parte importante de mi existencia.

MIS AMIGOS

Elvira Alvarado Figueroa sin palabras.
Aaron Ozaeta Hall mi hermano.

A MIS FAMILIARES EN GENERAL

Julia Estela Pómes Alvarado Vda. de López
Mima López Pómes de Gramajo
Elfego Gramajo Calderón
Rafael Montúfar López

Muchas gracias.

A MIS COMPAÑEROS

Con quienes compartí algunos años de trabajo, esfuerzo, dedicación, voluntad, carácter y sobre todo amistad.

I. Introducción

El desarrollo de un manejador de pruebas para evaluar el Coeficiente Intelectual (C.I.) implica en primer lugar, la búsqueda de un tipo de prueba que se adapte al manejo por medio de una computadora, por lo que se determinó con la ayuda del psicometrista que las pruebas de orden menor eran las más adecuadas por su adaptabilidad y manejo; dichas pruebas consisten en una pregunta con una o más respuestas. Luego se procedió a realizar un estudio de las posibles reacciones o resistencia que esta prueba podría causar en un usuario que no estuviera relacionado con equipo de computación; también se evaluaron los requerimientos de información para el manejo de la prueba y la generación de resultados como reportes.

La contribución del presente trabajo al área de la psicometría en nuestro país es el de proporcionar una nueva herramienta de trabajo que haga más tecnificado el accionar de los psicometristas, para obtener resultados inmediatos del cualquier prueba realizada a un individuo o grupo de ellos, lo cual permite hacer inferencias sobre el resultado de la evaluación de una persona respecto a un grupo que hubiese tomado la misma prueba con anterioridad o hacer inferencias sobre todo el grupo; al mismo tiempo, permite almacenar la información el tiempo que sea necesaria y poder acumular los resultados con el fin de utilizar esta herramienta como medio de investigación a través de los resultados que se obtengan de su aplicación en el campo.

Se trabajo bajo los siguientes lineamientos: el usuario hace uso del menor número de teclas de la computadora para un rápido aprendizaje, para no tener dificultad a la hora de tomar la prueba; en el caso del psicometrista, se le proporciona un menú y dependiendo de su selección, sólo deberá utilizar las teclas de movimiento; para el caso de ingreso de la prueba, hará uso del teclado como el de una máquina de escribir común al ingresar las preguntas con sus respectivas respuestas.

Cabe destacar que el presente trabajo no utiliza una sola prueba, sino que permite el ingreso de varias pruebas, y las almacena para utilizations futuras, por lo que es competencia del psicometrista elegir la prueba adecuada al momento de evaluar a un individuo, con lo cual se mantiene la necesidad de que esta herramienta sea utilizada por un profesional de la psicometría.

II. Objetivos

II.1 Generales

- II.1.a Desarrollo de un manejador de pruebas para la evaluación del coeficiente intelectual u otro tipo de pruebas de igual similitud.
- II.1.b Implementación de un manejador de Pruebas del C.I. en un lenguaje de Programación de tercera generación.
- II.1.c Desarrollo del Software necesario para la implementación del interfase, que facilite la comunicación entre el usuario y la computadora.
- II.1.d Explotación de los recursos que nos ofrece la computadora, en la aplicación de las pruebas.

II.2 Específicos

- II.2.a Aplicación de las pruebas a grupos de personas para obtener datos que puedan ser tratados estadísticamente y puedan llegar a formar parte de información de estudio, por parte de instituciones interesadas, en el área de análisis de C.I. de la nación.
- II.2.b Desarrollo de Menús que permitan trabajar con distintos tipos de pruebas, que depende del usuario que realice la prueba.
- II.2.c Distribución de la implementación del manejador de pruebas a instituciones relacionadas con el área, para lograr el alcance esperado.
- II.2.d Que el manejador pueda ser utilizado para el reclutamiento o evaluación de personal en la industria, por parte de los encargados de recursos humanos.

III. Antecedentes

Se investigó en el tesario de las universidades y no se encontró material que estuviera relacionado con el punto de implementación en computadora, por lo que sólo se hace referencia a las tesis de las áreas de psicología que de alguna manera investigan el tema, las cuales, sin embargo, sólo son áreas relacionadas al estudio de pruebas y su veracidad en la aplicación.

Estudio de validación de una batería de tests empleada en INTECAP.
Aldana Rodas de Samayoa, Sandra
USAC

Causas fundamentales que provocan las deficiencias del cociente intelectual en los niños.
Celis Aragón, Sonia Susana.
USAC

Validez predictiva del test de habilidad mental OTIS-LENNON
Buitrago, José Gabriel Jaime
UVG

Predicción del rendimiento académico
UVG.

Relación entre los resultados obtenidos en los tests
UVG.

Validez predictiva de los test de aptitud diferencial
UVG.

Estudio correlativo de las baterías de aptitud TAD y AMP
UFM.

Estudio correlativo del cociente intelectual del test AMP y el test OTIS
UFM.

Estudio normativo de las pruebas de destreza manual de la batería general.
URL.

Etapas de la inteligencia sensomotriz en menos de 5 a 9 años
URL.

Correlación entre el test de velocidad y exactitud y el rendimiento académico
URL.

Baremación del test batería factorial
URL.

Las pruebas miden la capacidad para tratar con conceptos abstractos presentados en forma numérica o verbal, figuras simbólicas, que es lo que se presentan en varias de las tesis antes mencionadas.

IV. Justificación

El presente trabajo es un aporte a las áreas relacionadas con la psicología y la educación en nuestro país, que nos permite enfocar dicha área desde un punto de vista diferente, lo que nos permite saber si los trabajos realizados allí, están produciendo los resultados esperados y si debemos reorientarlos, para llegar a tener una sociedad con desarrollo intelectual.

Por esa razón la implementación de pruebas para la medición y evaluación del C. I. ofrece una herramienta para los profesionales que se desenvuelven en esta área, y contribuyen a su desarrollo en el campo de la investigación de pruebas del Coeficiente Intelectual, sin pretender ingresar en un campo que pertenece a profesionales de la psicología, quines tienen la responsabilidad de desarrollar el campo de la psicometría y exigir nuevas herramientas para su uso a profesionales a los que puedan proporcionárselas.

Esto se debe a que el desarrollo es un proceso particular de cada pueblo que involucra todas las áreas que lo componen y cada uno debe establecer sus propios requerimientos, métodos, así como medir sus pasos; sin embargo, la estrategia de los pueblos deben también ser sensibles a su ambiente externo. La mayor parte convendrá en que el desarrollo tendrá que continuar en el manejo del poder de una buena política económica y el avance de la tecnología.

V. Listado de Tablas

Tabla No.	Página
-----------	--------

1	
---	--

VI. Listado de Figuras

Figura No.	Página
------------	--------

1	30
---	----

2	31
---	----

3	31
---	----

4	32
---	----

VII. Contenido

VII.1 Presentación estructurada del contenido

Sistema de computación
Orígenes y desarrollo
Componentes

La aparición y desarrollo de la psicometría
Generalidades
Los tests
Clasificación
Aplicación
Cualidades
Tipificación
Elaboración
Uso

Desarrollo
Introducción
Definición del sistema
Diseño
Implementación

VII.2 Descripción del contenido por capítulos

CAPITULO 1: Sistema de Computación

Aquí se darán las definiciones necesarias para el manejo de: conceptos de sistemas de computación y las aplicaciones, así como del equipo con que se debe contar para el manejo de la aplicación; esto se hace con el objeto de que una persona no relacionada con el campo pueda comprenderlo.

CAPITULO 2: Psicometría

Se hará una reseña histórica de las psicometría, los tests, concepto, clasificación y aplicación.

CAPITULO 3: Desarrollo

Se hará una introducción acerca de los aspectos técnicos, se dará una definición del sistema, diseño, programación, pruebas, manejo de los resultados, y se definirá el manual del usuario, lo cual implica el manejo de la aplicación por medio del interfase con el usuario; también incluye listados de programa fuente.

VII.3	Indice	
I.	Introducción	i
II.	Objetivos	ii
	II.1 Generales	ii
	II.2 Específicos	ii
III.	Antecedentes	iii
IV.	Justificación	iv
V.	Listado de tablas	v
VI.	Listado de figuras	v
VII.	Contenido	vi
	VII.1 Presentación estructurada del contenido	vi
	VII.2 Descripción del contenido por capítulos	vi
VII.3	Indice	vii
VII.1	Sistema de computación	1
VII.1.1	Orígenes y desarrollo	1
VII.1.2	Componentes	4
	VII.1.2.a Hardware	6
	VII.1.2.b Software	7
VII.2	La aparición y desarrollo de la psicometría	11
VII.2.1	Bosquejo histórico y generalidades	11
VII.2.2	Los Tests: Concepto, Historia, Clasificación y Aplicaciones	12
	VII.2.2.a Concepto	12
	VII.2.2.b Historia	14
	VII.2.2.b.1 Orígenes	14
	VII.2.2.b.2 La influencia de la psicología experimental	15
	VII.2.2.b.3 Los test de procesos mentales superiores	16
	VII.2.2.b.4 Los test colectivos	17
	VII.2.2.b.5 Los tests de personalidad	17
VII.2.3	Clasificación de los tests	18
	VII.2.3.a Según el modo de administrarlos	18
	VII.2.3.b Según el modo de expresión	19
	VII.2.3.c Según el sector que explore el test	19
	VII.2.3.c.1 Tests de eficiencia	19
	VII.2.3.c.2 Tests de personalidad	19
VII.2.4	Aplicaciones de los tests	20
VII.2.5	Cualidades de los tests : fiabilidad y validez	21
VII.2.6	Tipificación, sistemas de medida, normas	22
VII.2.7	La elaboración de un test	23
VII.2.8	La aplicación y uso de los tests	24
VII.3	Desarrollo	26
VII.3.1	Introducción	26
VII.3.2	Definición del sistema	27
	VII.3.2.a Elaboración de la prueba	27
	VII.3.2.b Toma de la prueba	28
	VII.3.2.c Reportes	29
	VII.3.2.d Supervisores	29
VII.3.3	Diseño	29
	VII.3.3.a Archivos	29
	VII.3.3.b Pantallas	29
	VII.3.3.c Menús	32
	VII.3.3.d Ingresos	32
	VII.3.3.e Salida	33
	VII.4.3.e.1 Pantalla	33
	VII.4.3.e.2 Impresora	33
VII.3.4	Implementación	33
VIII.	Conclusiones	viii
IX.	Recomendaciones	viii
X.	Bibliografía	ix
XI.	Glosario	x
XII.	Indice invertido	xv

VII.1 Sistema de computación

Cada día más naciones han empezado a percibir a la información como un nuevo recurso, y a la informática como la forma racional, sistemática y necesaria para el empleo de la información como apoyo a la solución de los problemas políticos, económicos y sociales.

El reconocimiento del potencial de esta tecnología ha generado el interés, tanto de las naciones avanzadas como en las en vías de desarrollo, que están por impulsar políticas orientadas al desarrollo de la tecnología informática y compatibilizar sus metas e instrumentos con la política nacional global.

Los sistemas modernos de computación consisten en una gran conjunción de elementos de circuitos y de programación que han sido diseñados para proporcionar a la computación un ambiente productivo y agradable.

Muchos aspectos de la sociedad moderna, que han llegado a ser considerados lugares comunes, no serían posibles sin la presencia de la computadora. Estas máquinas se utilizan ampliamente en muchas áreas de los negocios, la industria, la ciencia y la educación. ¿Qué es una Computadora? Para muchos representa el encuentro violento con el futuro. En el cine y la televisión, se presentan enormes y complejas máquinas que encienden luces y hacen girar carretes de cinta. Con mucha frecuencia, la computadora tiene una personalidad fría e impersonal, siniestra e intrigante. EL operador de la computadora es un técnico brillante, que se encuentra al borde la locura.

VII.1.1 Orígenes y desarrollo

Las necesidades del cálculo por parte del hombre datan desde hace miles de años. Todos han oído historias de pueblos primitivos contando sus ovejas por medio de piedrecillas. Nuestro sistema numérico de base 10 proviene indudablemente del uso de los 10 dedos de las manos como elementos de cálculo.

Las habilidades primitivas en el campo de la computación se desarrollaron en forma sorprendente. Los arqueólogos han desenterrado tablillas de arcilla que contienen cálculos matemáticos elaborados en la Edad Media. Tablillas que contienen tablas de multiplicación y de recíprocos han sido mil setecientos años A. de J. C. Los babilonios utilizaban un sistema numérico sexagesimal (de base 60), del cual provienen nuestras actuales unidades de horas, minutos, segundos. Hay suficiente evidencia para demostrar que los babilonios resolvieron muchas clases de ecuaciones algebraicas; las fórmulas se representaban por listas de reglas que se debían seguir paso a paso para la evaluación con ejemplos numéricos reales. Las reglas que se proporcionan son suficientemente generales como para permitir la sustitución de números diferentes, es decir, la solución de la ecuación con distintos argumentos. De esta forma, las reglas hacen suponer que los babilonios estaban capacitados para predecir los eclipses con precisión, unos quinientos años A. de J. C.

Desde los más antiguos tiempos, el hombre ha tenido conciencia de sus limitaciones respecto a su capacidad mental de cálculo, por lo que ha ido desarrollando las más diversas herramientas de apoyo, que van desde las muy simples hasta las más complejas. A lo largo de cientos de años, los viajeros que visitan Salisbury Plain, en el sur de Inglaterra, quedaban fascinados con un misterioso agrupamiento de enormes piedras conocido como Stonehenge, como un mercado, tal vez un templo, e incluso un lugar de sacrificios humanos.

En 1,965 en un libro Stonechenge Decoded, Gerald S. Hawkins presentó los resultados de un profundo estudio acerca de los propósitos de Stonechenge. La conclusión es que el lugar fue construido entre el período de mil novecientos a mil seiscientos años A. de J. C. y se utilizó una especie de combinación de observatorio computadora. En esta investigación (realizada de hecho con ayuda de una computadora), Hawkins descubrió un sorprendente número de alineaciones entre grupos de varias piedras y las posiciones importantes del Sol y la Luna, que son demasiadas en la práctica para que sean accidentales. Se supone posteriormente que los operadores de las computadora de Stonechenge lo utilizaban no sólo como registro de la llegada y del paso de las estaciones, sino también como un predictor de eclipses. La operación de la computadora de Stonechenge, aunque muy simple, revela un diseño cuidadoso y elaborado.

El sitio exacto de Stonechenge está particularmente bien elegido; existe sólo una latitud en el hemisferio norte donde, en sus posiciones extremas, los azimuts del Sol y la Luna están separados por 90 grados; se encontró que Stonechenge está a sólo unas millas de esa latitud. Parece como si los constructores de Stonechenge eran suficientemente capaces de construir una "computadora"; también podrían escoger el sitio exacto por medio de su geometría astronómica.

Estructuras que parecen efectuar similares a Stonechenge se encuentran en varios lugares, incluyendo Norteamérica y América del Sur. Una serie de "ruedas medicinales" en el oeste de Estados Unidos y Canadá, algunas de las cuales parecen ser tan antiguas como Stonechenge, muestran características marcadamente similares.

Por desgracia, algunas de las habilidades desarrolladas en el diseño de estas estructuras fueron desapareciendo con el tiempo, que gracias a ellas habría sido posible disfrutar mucho antes de las ventajas de la tecnología de la computación. Las necesidades de medios de computación han permanecido, a pesar de todo, y el ser humano parece estar mal preparado para afrontar la precisión. En la siguiente sección se analizará el desarrollo de las herramientas que sirven de ayuda en la tarea de la computación; todo ello culmina en el siglo XX con la computadora electrónica digital.

Todas la grandes civilizaciones de la antigüedad inventaron métodos e instrumentos de cálculo. Los Sumerios, Caldeos, Babilonios y Egipcios utilizaron conceptos matemáticos y operaciones elementales. El instrumento de cálculo más antiguo que se ha encontrado es el ábaco que data de dos mil quinientos A. de J. C. se supone que fue inventado primero por la civilización China. La gran civilización Maya proporcionó un gran aporte a la ciencia matemática al inventar el cero. Los Árabes nos legaron el sistema decimal, elementos de álgebra, trigonometría, etc. Los griegos sobresalieron en lógica y matemática.

Es hasta el siglo XVII, cuando nace el cálculo automático. El hombre inventa las primeras máquinas capaces de hacer cálculos por medios mecánicos; entre este tipo de máquinas sobresalen :

- 1,642 Blaise Pascal inventa la "Pascalina", máquinas sumadora.
- 1,671 Leibnitz inventa una máquina calculadora para sus tablas trigonométricas y astronómicas.
- 1,804 Jacquard inventa una tarjeta perforadora para usar los telares.
- 1,822 Babbage y Lady Lovelace inventan la máquina analítica diferencial. El valor de estas invenciones es que, aunque en su tiempo nunca funcionaron, dejaron sentados los fundamentos básicos de las ciencias de la computación.
- 1,889 Burroghs inventa la máquina calculadora de multiplicación directa.

- 1,890 Hollerith inventa la tarjeta perforada usando la idea de la tarjeta perforada de los telares de Jacquard. Lo hace para facilitar el conteo del censo de los EUA.
- 1,930 Atanasoff inventa la primera computadora digital con sus alumnos del curso de Física de la Universidad del estado de Iowa. No fue una computadora en gran escala, sólo fue un proyecto de una clase de física.
- 1,940 Zuse inventa varias calculadoras automáticas en Berlín.
- 1,944 Aiken inventa la Mark I, hecha de relés. Aquí puede decirse que termina la prehistoria de la computadora.
- 1,943 Eckert, Mauchly y Goldstine inventa la primera computadora en gran escala que usó el Gobierno de los Estados Unidos. Era completamente electrónica. Funcionaba con 19,000 tubos al vacío. El nombre de esta computadora fue Eniac.
- 1,952 Von Neumann y colaboradores inventan la EDSAC y la EVAC, primeras computadoras con capacidad de almacenar un programa.
- 1,944 Se inventa una serie de computadoras entre ellas : ILLIAC,
1,952 JHONIAc, MANIAC, UNIVAC.
- 1,953 IBM inventa la 701, primera de una serie de computadoras científicas.
- 1,959 Se inventan las computadoras transistorizadas, es decir, que ya no usaban tubos al vacío, sino transistores.
- 1,960 Inventan una serie de computadoras que ya usan circuitos integrados en vez de transistores. Las marcas que más sobresalen son: IBM, NCR, BURROUGHS, HEWLET PACKARD, etc. En estos años, las computadoras se usaron principalmente para negocios y cálculos científicos.

En esta breve historia hasta antes de 1,975, las computadoras se consideraban para uso de científicos, hombres de negocios y estudiantes avanzados de las universidades. Fue en 1,976 cuando se inicia la era de las microcomputadoras en EUA; antes de inventarse la APPLE, que fue la primera en llegar a popularizarse, ya se habían inventado otras micros que no tuvieron tanto éxito en su época pero que deben ser consideradas como pioneras de la computadora personal. Entre ellas están : SPHERE COMPUTER, PROCESSOR TECHNOLOGY SOL, ALTAIR 8800, IMSAI, SCALBI 8B, etc.

La APPLE la inventaron dos jóvenes llamados Wozniak y Jobs en 1,976 en el garage de la casa de Wozniak. Se considera como la primera computadora personal que fue diseñada para todo uso.

Lo verdaderamente importante de las computadoras personales es que han permitido que esta maravilla de la tecnología moderna esté al alcance del público en general, incluso de los niños y jóvenes. Ya no es necesario ser científico para poder aprender, usar y disfrutar de una computadora. Los usos que pueden dársele son muy variados; desde procesadores de palabras, archivos, programas de finanzas, hasta juegos, programas educativos, programas artísticos que ayudan a componer música, a dibujar, a crear nuevas formas y diseños. También pueden usarse para ayudar a personas lisiadas, a personas con problemas de aprendizaje, para enseñar a leer los niños pequeños, etc. Los campos en que la computadora personal puede utilizarse son verdaderamente ilimitados.

VII.1.2 Componentes

¿ Qué es realmente una computadora ?

La pregunta no puede responderse con la misma sencillez que, como ¿Que es un televisor? La razón es que la computadora no posee una única función, sino que constituye una de las nuevas máquinas capaces de ejecutar una serie enorme de funciones, según el programa que su usuario le suministre. Veamos así que la computadora no es solamente maquinaria o un conjunto de dispositivos electrónicos (hardware); pues existe otro elemento llamado software, el cual determina la actividad de la computadora, y sin el cual ésta no es más que una maza inerte de piezas electrónicas.

En términos generales, es una máquina electrónica muy veloz, que realiza una gran diversidad de funciones, según el programa que le suministra la persona que la usa. En términos generales, una computadora es una máquina electrónica sumamente veloz, la cual acepta información de entrada que procesa según un programa almacenado en su memoria, y produce información de salida. Una computadora se compone de cinco unidades funcionales independientes :

- entrada
- memoria
- lógica / aritmética
- control
- salida

En una computadora personal (o PC) la unidad de entrada y de salida corresponden al teclado y a la pantalla o monitor, respectivamente. Estas unidades permiten la comunicación entre la computadora y nosotros, formando parte de lo que se denomina interfaz hombre-máquina. La unidad de control y la lógica/aritmética (o ALU por Arithmetic Logic Unit); constituyen la unidad central de procesamiento (o CPU por Central Processing Unit) ésta es el cerebro de la computadora, su centro de procesamiento de información. La unidad de memoria, por su parte, es la encargada de almacenar la información (datos o instrucciones) que la computadora procesará. Está constituida por dos tipos de memorias: ROM y RAM, que se diferencian entre sí por el tipo de acceso a la información que almacenan. El contenido de una RAM puede leerse y modificarse, y un tamaño es una indicación de la capacidad de la computadora para almacenar programas y datos. La memoria tipo ROM solamente puede ser leída, pero no modificada. Se emplea para almacenar programas o datos fijos. Regresando a nuestro ejemplo de la PC, tanto la CPU como la unidad de memoria se encuentran dentro de la misteriosa "caja negra" sobre la cual colocamos la pantalla.

Ahora bien, todos los demás dispositivos que asociamos con una computadora, tales como unidades de disketts, discos duros, unidades de cinta e impresoras, constituyen lo que se llama periféricos. Estos no son más que ampliaciones o extensiones de unidades de entrada, salida o memoria, que conectamos a nuestra computadora con el fin de hacerla más versátil.

Vamos viendo entonces que la computadora no piensa; es simplemente una impresionante máquina electrónica capaz de ejecutar instrucciones a grandes velocidades (es común oír hablar de MIPS o millones de instrucciones por segundo). De nosotros depende aprovechar esta maravilla tecnológica, y alimentarla con los programas adecuados.

El lenguaje propio de la computadoras :

Existen computadoras muy pequeñas y otras que pueden medir hasta dos metros de altura y pesar varias toneladas. La diferencia entre ellas radica en su complejidad de diseño, velocidad de procesamiento, capacidad de memoria y precio. Aunque cada vez es más difícil establecer fronteras entre un tipo y otro de computadora, se mencionarán algunas de las categorías actuales :

microcomputadora,
supermicro o micromini,
minicomputadora,
supermini,
mainframe,
supercomputadora.

Todas esas computadoras tienen algo en común; están formadas por millones de diminutos circuitos electrónicos integrado en pastillas de silicio (uno de los elementos más comunes en la tierra) denominados chips, y todas hablan un lenguaje basado solamente en dos símbolos o estados.

Siempre que se habla de computadoras, se utilizan las palabras bits y bytes. Estos vocablos describen la manera en que las computadoras almacenan y emplean la información, y lo hacen de un modo bastante diferente al nuestro.

Nosotros representamos los números mediante diez símbolos diferentes (0 al 9) y los manejamos en múltiplos de diez. Esto se conoce como un sistema de base 10. Las computadoras, por su magia matemática, se sirven sólo de dos números: el cero y el uno. Toda la información que se manejan la traducen a estos dos símbolos. Los bits y los bytes son formas que representan combinaciones entre dos números.

Un bit es la unidad de información más pequeña que maneja una computadora. Es la forma que ella emplea para representar los ceros y unos. A un grupo de 8 bits ¿qué son y por qué se denominan así?.

Las computadoras son dispositivos electrónicos y en consecuencia, todas sus funciones las realizan en última instancia, con señales eléctricas. Una señal eléctrica puede esta "encendida" o "apagada" : si está apagada representa cero. Si está encendida un uno, éste es el principio por el cual las señales eléctricas pueden representar números.

Dicho de otra forma, las computadoras manejan un sistema binario, o de base dos, compuesto por dos elementos o estados : el cero y el uno. Un bit es, así, un dígito binario (o binary digit) de donde viene el nombre bit. Estamos acostumbrados a oír hablar de lenguajes de programación como Basic, Pascal , C, o bien de paquetes específicos como hojas electrónicas procesadores de textos o bases de datos. A todos ellos los vemos como una serie de instrucciones escritas o presentadas en un lenguaje comprensible para nosotros. Sin embargo, la computadora no las entiende así; antes de que pueda ejecutarlos, debe llevarse a cabo todo un proceso de traducción al lenguaje binario o lenguaje de máquina. Es aquí en donde aparecen en escena programas como los llamados compiladores, los cuales se encargan precisamente de tomar nuestro programa para traducirlo a instrucciones simples expresadas como secuencias de unos y ceros que nuestra computadora ya podrá entender y procesar.

En esta forma, el hombre puede comunicarse con las computadoras por medio de gráficas, texto, imágenes, sonido y mucho más. Mientras la información puede convertirse en una serie de unos y ceros, las computadoras podrán entendernos y obedecernos.

Nuestra comunicación con las computadoras :

Una computadora es una máquina versátil que puede desempeñar diversas funciones. Una misma computadora puede ser utilizada por un ejecutivo, un científico, un estudiante, un técnico, un profesional o un niño. El software es lo que determina su actividad.

Es sus inicios, las computadoras exigían que la persona que pretendiera emplearlas les "hablara" en unos y ceros. Conforme han transcurrido los años, el hombre ha ido establecido formas de comunicarse con las computadoras, cada vez

más parecidas a nuestro lenguaje natural. Contamos ahora con muchos tipos de lenguajes de programación que nos permiten elaborar software o programas en forma sencilla y más cercana a lo "humano", y se alejan de la frenética y compleja actividad binaria que se suscita dentro de la computadora.

Existen dos grupos principales de software : el software de sistema y el software de aplicaciones. El primer grupo incluye todo aquel software de control que realiza funciones comunes para todos los usuarios de la computadora y que permite correr programas de aplicación, interactúa con todos los programas específicos y en algunas ocasiones, con el mismo usuario. Dentro de esta clasificación podemos incluir a los sistemas operativos, manejadores de bases de datos, programas de control de comunicaciones, etc.

El software específico para aplicaciones es el que se desarrolla, empleado algún lenguaje de programación, para una aplicación en particular. Dentro de este grupo se puede hablar de programas "hechos en casa", o bien paquetes de software o "paquetes enlatados". Es decir, que un usuario puede programar su propia aplicación según sus necesidades específicas, en cuyo caso aprenderá algún lenguaje de programación y elaborará una serie de instrucciones. integrando un programa para resolver su problema. Por otra parte, cuando se decide emplear, por ejemplo, una hoja electrónica, un procesador de textos o un administrador de proyectos, se puede recurrir a comprar alguno de los múltiples paquetes ya existentes o programas ya elaborados, leer el manual y aprender a utilizar este software enlatado para resolver nuestro problema.

VII.1.2.a Hardware

Cuando alguien hable de hardware, se refiere a la parte física de nuestro sistema de cómputo. Es decir, a las computadoras en si y a todos sus periféricos o dispositivos de entrada y salida.

Periféricos :

Son los aparatos que están conectados al corazón de la computadora (CPU); se les llama también dispositivos.

Algunos de ellos sirven para ingresar letras, números y caracteres especiales que el CPU procesará. Estos se llaman dispositivos de entrada, por ejemplo el teclado.

Otros sirven para que el CPU nos muestre los datos ya procesados, es decir la información. Se llaman dispositivos de salida; por ejemplo la impresora.

Las aplicaciones del hardware pueden dividirse en tres grandes categorías:

- Entrada
- Salida
- Entrada/Salida

Hay algunos periféricos que cumplen ambas funciones, que son de entrada y de salida, por ejemplo, el Disk Drive.

Pantalla :

Es el dispositivo que nos permite visualizar letras, números, caracteres especiales, colores, dibujos, gráficas y también escuchar todo tipo de sonidos y notas musicales e incluso la voz humana. Se llama pantalla de rayo catódicos. Algunos modelos de computadoras ya traen la pantalla incorporada; otros permiten conectarle un monitor especial de computadora.

Impresora :

Es el periférico que imprime en papel la información que sale del CPU. Puede imprimir: letras, números, caracteres especiales, dibujos, gráficas, etc.

Manejador de Diskette (Disk Drive) :

Es el dispositivo que nos permite manejar el diskette (medio electrónico de almacenamiento), es decir, leer y grabar en él. Físicamente es como una caja que tiene una ranura en la cual se inserta el diskette.

Teclado :

Es el dispositivo que nos permite ingresar los datos, presionando las teclas de letras, números y caracteres especiales.

Las teclas de la letras se encuentran dispuestas de manera similar a las de la máquina de escribir, sin embargo, otras teclas que son específicas de una computadora, entre ellas: la tecla de nueva línea, la de tecla de interrumpir, etc.

Diskette (Floppy Disk) :

El diskette está construido de material electromagnético. Es de forma redonda y viene protegido por un estuche de material sintético cuadrado. El diskette no puede ser sacado de su estuche. Además trae un sobre de papel del cual se saca antes de usarlo. Existen dos tamaños de diskettes: 5 1/4 pulgadas y de 3 pulgadas de diámetro.

Disco Duro (Hard Disk) :

Este es un medio de almacenamiento electrónico secundario al igual que el diskette, pero la diferencia es que esté permanece fijo en la computadora y su capacidad de almacenamiento es mucho mayor que la de un diskette.

Las aplicaciones del hardware pueden dividirse en tres grandes categorías:

- Procesamiento de información
- Control de procesos y aplicaciones en tiempo real
- Inteligencia artificial

La ingeniería de hardware se encarga de proceso para desarrollar hardware, el cual se integra por tres fases:

- Planeamiento y especificación
- Diseño e implementación del prototipo
- Fabricación, distribución y servicio en campo

Aunque son inseparables en un sistema de cómputo, el software y el hardware han tenido un crecimiento y una evolución muy diferentes. EL hardware en el mundo del almacenamiento y la transmisión : el software en el mundo de la lógica y el lenguaje.

VII.1.2.b Software

EL software es más que programas de computadora. Es información que existe en dos formas básicas : componentes ejecutables por la máquina. Entre estos últimos podemos hablar de: software de sistema, de tiempo real, de negocios, científico y combinatorio. Sin estos elementos, nuestra computadora no podría hacer absolutamente nada.

Ahora., ¿qué es entonces lo otro que también se llama Software? todos sabemos que el software tiene un ciclo de vida, el cual comprende todas las actividades que se realizan antes de su desarrollo y después de su implementación. Detrás de todo eso existe un método que forma la base de lo que se conoce como ingeniería de software. Podemos dividir al ciclo de vida del software en tres fases: planeamiento, desarrollo y mantenimiento. En cada una de estas etapas se producen documentos específicos: en el planeamiento, el ingeniero de software parte de un plan de software y emite una serie de especificaciones de requerimientos del software. Con esto empieza el desarrollo en sí, el cual parte de una descripción preliminar de diseño y después de muchas horas de

trabajo, culmina con el software ejecutable, una descripción detallada del diseño y una serie de planes y procedimientos de prueba. Ahora bien, ¿qué más falta si ya tenemos nuestro programa listo para usarlo? La última fase de ciclo de vida del software es la de mantenimiento. Esta etapa dura todo el tiempo que viva el programa y se descompone en cuatro formas: mantenimiento correctivo, preventivo adaptivo y perfectivo. Cada acción de mantenimiento se inicia con un reporte del problema en el software y finaliza con un reporte de cambios en el software y sus respectivas modificaciones en el código y el documento de diseño.

Toda esta información nos permite entonces aclarar lo siguiente: el software no es un sinónimo de programa para computadora. Es mucho más que eso. Incluye tanto el código que muestra computadora ejecutará, así como la documentación asociada a su ciclo de vida.

Es muy importante tener claro este concepto, pues está demostrado que muchas fracasos en el campo de aplicaciones de la computación, se deben precisamente al hecho de considerar sólo una cara del software.

Firmware :

En las primeras épocas de las computadoras, el software se codificaba y almacenaba en cintas de papel perforadas, similares a las que emplean los operadores de télex. Luego las cintas de papel fueron reemplazadas por cassettes y discos magnéticos.

En los años setentas se intentó una nueva técnica para almacenar software en chips de memoria tipo ROM, diseñados para tal fin.

A los programas o rutinas almacenados en una memoria ROM de silicio se les denomina firmware o "silicom software".

Programa :

Un programa es una lista de instrucciones que la computadora obedece; un programa debe estar hecho por un ser humano, y debe ser creado por el hombre para que luego la computadora pueda ejecutarlo.

Cuando la computadora es capaz de desplegar colores y movimientos en la pantalla: imprimir dibujos, letras números, gráficas, en el papel de la impresora; cuando ejecuta música y reproduce todo tipo de sonidos y aún voz humana; cuando realiza complicados cálculos de matemática; cuando nos indica por medio de letras o sonidos que hemos cometido un error al ingresar los datos; esto es posible a través de los programas y del equipo de alta tecnología que los seres humanos han inventado.

Hacer un programa es hacer una lista de instrucciones debidamente ordenadas que la computadora obedecerá al pie de la letra. Para poder escribir un programa, es necesario tener mucha lógica, es decir, pensar de una manera ordenada, correcta y creativa. Programar es un trabajo muy creativo que sólo los seres inteligentes pueden realizar; la inteligencia y la capacidad de crear es lo que distingue al ser humano del resto de los seres vivos, entonces todos podemos aprender a programar si aprendemos las técnicas necesarias de la programación. Existe una profesión en computación llamada programador. El programador es la persona que se dedica a elaborar programas.

Lenguajes de Programación :

Si un programa es una serie de instrucciones que la computadora obedece, entonces debemos saber cómo darle esas instrucciones a la computadora. Así como para comunicarnos con otros seres humanos necesitamos usar un lenguaje adecuado que ellos entiendan, también para comunicarnos con la computadora necesitamos usar un lenguaje adecuado que la computadora entienda.

Un lenguaje de programación es un lenguaje que se usa para comunicarse con las computadoras. Los programas se deben escribir en un lenguaje de programación para que la computadora pueda entender y obedecer las instrucciones que le queremos dar.

Un programa no puede ser escrito en lenguaje común. Por ejemplo no puedo darle instrucciones a la computadora así: "Señorita computadora, por favor deseo ver los estados financieros".

Cuando deseamos que la computadora nos ayude a resolver algún problema debemos seguir los pasos siguientes:

1. Pensar el problema en nuestro lenguaje
2. Traducirlo a un algoritmo
3. Hacer un diagrama de flujo
4. Escribirlo en un lenguaje de programación que la computadora entienda
5. Ingresarlo en la memoria de la computadora para que sea traducido a un lenguaje máquina, es decir, a un lenguaje electromagnético (son señales eléctricas y magnéticas)
6. Usar el programa y revisar si los resultados son los esperados. Si no, corregir los errores empezando otra vez desde el paso 1
7. Escribir todos los pasos que seguimos para resolver el problema (del 1 al 6) y archivarlo de manera ordenada para que nos sirva en problemas futuros.

Programas paquete :

Estos son programas que ya vienen hechos por los fabricantes y que para usarlos ya no se necesita programarlos. Estos programas son de juegos, contabilidad, de música, de matemática educativos, Etc. Para usar estos programas necesitamos leer su manual de instrucciones y aprender el uso de cada una de ellas.

Entre los principales y más útiles de estos programas, están los procesadores de palabras (transforma la computadora en una máquina de escribir) la hoja electrónica (transforma la computadora en una hoja de trabajo de filas y columnas como las que se usan en contabilidad) la base de datos (que transforma la computadora en un archivo electrónico que puede ordenarse de cualquier forma).

Un sistema es una combinación estructurada de módulos que interactúan satisfaciendo un conjunto de objetivos. Un sistema computarizado es un conjunto de programas (instrucciones o comandos que se le dan a la computadora), que funcionan conjuntamente para lograr un objetivo.

Importancia y ventajas :

Son muchas las ventajas de computarizar sistemas que tradicionalmente se trabajan manualmente.

Introducir una computadora es un proceso que lleva tiempo y requiere esfuerzo. No es tan sencillo como comprar una máquina cualquiera y contratar a un programador para que adapte algunos procedimientos manuales a la computadora. La decisión de computarizarse debe ir precedida por un análisis de los

procedimientos manuales en uso.

Un analista de sistemas podrá entonces recomendar cuál es el equipo más adecuado y qué procedimientos deberán automatizarse. Todo esto, después de hacer un cuidadoso análisis de la situación actual.

Los objetivos que este analista tendrá en mente serán :

1. Mejorar la información disponible para la toma de decisiones.
2. Mejorar el control sobre las actividades rutinarias, eliminar trabajo innecesario, mejorar la atención a terceras personas, brindándoles información rápida y actualizada. etc.

Trabajar con computadoras no es difícil. Cada día se crean nuevas herramientas (mejores diseños de terminales, mejores programas para interactuar con los usuarios, etc) que hacen más sencillo y agradable trabajar con ellas, por lo que es recomendable en la medida de las posibilidades de cada entidad, establecer sistemas de información que les permitan hacer más efectivo su trabajo.

VII.2 La Aparición y Desarrollo de la Psicometría

VII.2.1 Bosquejo Histórico y Generalidades

En sentido estricto, medir es determinar la longitud, extensión, volumen o capacidad de alguna cosa. En sentido más amplio, medir es también comparar una cosa no material con otra. Es propio y característico de la medida de descripción de los datos que se haga en términos numéricos, lo que reporta la extraordinaria ventaja de que a éstos se les pueden aplicar procedimientos matemáticos. Como es sabido, no pueden considerarse las matemáticas como una ciencia empírica, sino como un lenguaje de validez universal que dispone de un vocabulario prácticamente ilimitado, cuyos términos se pueden definir con gran rigor. Sus leyes operacionales son lógicas y precisas y se pueden considerar como su propia **sintaxis**. La predicción y seguridad de los datos medidos dependen de la calidad, es decir de la fiabilidad y la validez, de los instrumentos mediante los cuales se realiza el proceso mesurativo. Los instrumentos y métodos de medidas se han utilizado siempre en el terreno de las ciencias de la naturaleza en el sentido diltheyano del término, pero no se han empezado a aplicar en los sectores de la ciencias sociales y psicológicas hasta bien avanzado el siglo XIX.

Se denomina psicometría al conjunto de métodos e instrumentos de medida que se utilizan para la investigación, descripción y comprobación de datos sobre el comportamiento psíquico. Las primeras aplicaciones psicométricas tuvieron lugar en íntima conexión con los comienzos de la psicología experimental y sirvieron para la investigación de ciertos aspectos de la psicología humana. Más tarde, las aplicaciones psicométricas se extendieron también al campo de la psicología diferencial, sirviendo de soporte al desarrollo de los tests mentales, pasando a ser estos instrumentos en sí mismos **métodos psicométricos**.

El estudio de diferencias cuantitativas en diversas áreas sensoriales había sido iniciado ya por los psicólogos y fisiólogos mucho antes del siglo XIX, aunque en forma tanto rudimentaria. Sin embargo, el hecho clave en este campo de la investigación psicología puede situarse en 1840, año en el que Weber publica los resultados de sus trabajos sobre las relaciones entre estímulos y respuestas sensoriales y postula diversas leyes entre las cuales las que más divulgación han tenido, han sido las del **umbral absoluto y umbral diferencial**.

Inmediatamente después, Fechner continúa las investigaciones de Weber, estableciendo una fórmula que expresaba matemáticamente la medida de la intensidad de las sensaciones y según la cual **la intensidad de la respuesta sensorial es proporcional al logaritmo del estímulo**. En un deseo de proporcionar la máxima exactitud a sus investigaciones psicofisiológicas, adoptó métodos experimentales ya conocidos e ideó otros nuevos, entre los que figuran los del **error medio y el de los cambios mínimos**.

Con esta orientación, los esfuerzos de la Psicología Experimental se centran cada vez más en la creación de métodos e instrumentos que permitan obtener medidas del comportamiento humano. en 1,879 Wundt inaugura en la Universidad de Leipzig un laboratorio que había de ser el más importante en el campo de la Psicología Experimental. Realiza numerosos trabajos encaminados a averiguar en qué forma se interrelacionan las sensaciones con las percepciones y las palabras con las ideas, estableciendo así un puente de unión entre la psicología y la psicología introspectiva. Uno de sus más brillantes continuadores fue J. Mc. Cattell, más tarde profesor de la Universidad de Pensilvania, que estudió principalmente los mecanismos de la atención y la asociación de palabras, preparando tablas de frecuencias para las respuestas a ciertas palabras-estímulo y realizando numerosos trabajos de psicología diferencial. Se debe también a Cattell la invención del taquistoscopio y la del término **test mental** que utilizó por primera vez en 1,890.

Así pues, los primeros psicólogos experimentales fueron los pioneros de la psicometría. Sin embargo, la aparición de los tests mentales en los albores del siglo XX influiría de tal forma en el desarrollo de la psicometría, que incluso hoy día es a estos métodos, y a los procedimientos estadísticos en que se fundan, a los que habitualmente denominamos métodos psicométricos.

Hemos dicho que se denominan métodos psicométricos a los procedimientos estadísticos en que se basa tanto la construcción de tests como la elaboración y presentación de los datos de las investigaciones psicológicas. Estos procedimientos se basan fundamentalmente en concepciones matemáticas sobre el cálculo de probabilidades.

Antes de 1,600, no existían tales concepciones; fue a partir del siglo XVII cuando comienzan a estudiarse con rigor científico los problemas del azar. Bernoulli fue el primero en publicar una obra enteramente dedicada a cuestiones. Físicos, matemáticos y astrónomos fueron quienes, principalmente, centraron su interés en estos problemas. En 1,812 Laplace publica un trabajo básico sobre el cálculo de probabilidades. Años más tarde Gauss demuestra la utilidad práctica de la curva normal en lo concerniente a los problemas de distribución de medidas y al estudio de los errores inherentes a la observación científica.

En 1,869 Galton, discípulo de Darwin, publica una obra Hereditary Genius, en la que, mediante la aplicación de métodos genealógicos intentó demostrar que la inteligencia se transmitía por vía hereditaria. Para describir los grados de semejanza entre individuos, no sólo ideó algunas pruebas psicométricas para hacer discriminaciones sensoriales, motrices y de otros procesos psíquicos elementales, sino que ideó una serie de procedimientos estadísticos sencillos, principalmente en relación con problemas de distribución y de correlación, que se aplicó a los resultados de sus trabajos experimentales.

K. Pearson científico inglés, discípulo de Galton, estableció a finales del siglo XIX la fórmula para la obtención del coeficiente de correlación, mediante la cual se puede medir el grado de concordancia o discrepancia entre dos series de variables y desarrollo, asimismo otros tipos de métodos estadísticos.

Otro avance en este dominio fue aportado por el estadístico R. A. Fisher al poner a punto el método de análisis de varianza, que permite averiguar la influencia relativa de diversas variables sobre un dato o fenómeno que se va a investigar.

Uno de los hitos más importantes en el desarrollo de los métodos psicométricos fue la aparición, en 1,904, de un trabajo de Ch. Spearman en que aplicando el método del coeficiente de correlación de Pearson a los resultados de diversos tests, sienta las bases de análisis factorial, método con el que posteriormente harían nuevas e importantes aportaciones a la psicometría C. Burt, G. Thompson, T.L. Kelley, L.L. Thurstone, R.B. Cattell y J.P. Guilford.

En conclusión, entendemos que son métodos psicométricos tanto los procedimientos que sirven para dar base científica a la investigación psicológica, como los instrumentos utilizados con este fin y que previamente han probado que responden a los requisitos exigibles a todo instrumento de medida.

VII.2.2 Los tests: Concepto, Historia, Clasificación y Aplicaciones

VII.2.2.a Concepto :

Test es una palabra inglesa que significa prueba y que se deriva de latín testis. Esta misma raíz figura en palabras españolas tales como testimonio o testigo. La palabra test se utiliza sin traducir, en todos los países del mundo y sirve para designar la modalidad de exploraciones muy extendida hoy en día en diversos campos científicos y técnicos y, entre ellos el psicológico.

H. Pieron define los test de la siguiente forma:

"Test es una prueba definida, fijando una tarea que se va a realizar, idéntica para todos los sujetos examinados. Debe disponer de una técnica precisa, que permita discriminar entre soluciones satisfactorias y erróneas o bien de un sistema numérico que permita puntuar el resultado. La tarea puede consistir en poner de manifiesto conocimientos adquiridos (test pedagógico)"¹.

Esta definición que fue adoptada por la Antigua Asociación Internacional de Psicotécnica tiene, sin embargo, la limitación de que puede aplicarse solamente a los tests de inteligencia, aptitudes o conocimientos. Pichot, teniendo en cuenta el creciente desarrollo que durante los últimos decenios han experimentado los test de personalidad, ha propuesto una definición que tiene sentido mucho más amplio y general, y que es el siguiente :

"Un test es una situación experimental estandarizada, sirviendo de estímulo a un comportamiento. Este comportamiento se evalúa mediante una comparación estadística con el de otros individuos colocados en la misma situación, pudiéndose así clasificar al sujeto examinado desde el punto de vista cuantitativo o bien desde el tipológico"².

Del análisis de la definición de Pichot se desprende :

1. La situación experimental (material del test, instrucciones, medio en que se tenga que aplicar, actitud del examinador, etc.), tiene que estar claramente definida, de tal forma que en cualquier aplicación que hagamos del test no varíen en lo más mínimo sus condiciones de administración. Asimismo debe haberse probado previamente que el test sirve para evaluar aquel aspecto de la conducta humana para el que fue propuesto.
2. Como quiera que sea, el test sirve de estímulo a un comportamiento, se precisa registrar éste con toda la precisión y objetividad posibles. A veces este registro es muy fácil y, por lo tanto, se cumple satisfactoriamente esta condición; tal sucede, por ejemplo, cuando el sujeto ha de limitarse a contestar si o no ante un cuestionario de personalidad o ha de resolver un test de aritmética en el que no cabe más que una solución o ha de realizar un dibujo, que será auténtico registro de lo realizado por el sujeto. Pero otras veces la tarea que se va a realizar (hacer relatos, realizar tareas manuales complejas colocar varios objetos o juguetes en determinada posición, etc) exige la introducción de técnicas de registro a menudo complicadas y que aumentan considerablemente el costo de los tests o bien dificultan la fidelidad del registro perfecto de la conducta del sujeto. Por ejemplo, en ciertos tests en que el examinado tiene que hacer relatos historias sobre unas imágenes, es a menudo difícil transmitir con toda exactitud lo que el sujeto va decidiendo y a la vez registrar sus pausas, indicar sus cambios de tono o inflexión de la voz, los momentos en que asocia con facilidad y de prisa o con lentitud, etc.

3. El comportamiento que hemos registrado de cada sujeto lo tenemos que comparar con el que previamente ha manifestado un grupo de individuos, cuyas características se determinaron con anterioridad y cuyos resultados se han elaborado estadísticamente como tablas con normas que permiten conocer la distribución de datos. Generalmente es preciso transformar las respuestas del sujeto en puntuaciones de la misma naturaleza que las utilizadas al construir las normas. La construcción de estas tablas, denominada **normalización o estandarización del test**, es indispensable para que un determinado examen o método de exploración, por muy bien elaborado que esté, se pueda considerar como un verdadero test.
4. Por último, una vez comparado el comportamiento del sujeto, los valores de la tabla de referencia, lo vamos a clasificar.

Esta clasificación o calificación que, en última instancia es el objetivo final de aquel aspecto de su conducta que hemos estudiado en relación con el dentro de una gama que abarcará desde los niveles muy superiores al promedio, hasta los niveles más bajos, pasando por toda la zona central que es la de la normalidad estadística.

VII.2.2.b Historia

VII.2.2.b.1 Orígenes

En esencia la finalidad de los tests consiste en medir la diferencias existentes en una determinada característica o rasgo entre diversos sujetos, o bien entre el comportamiento del mismo individuo en diferentes momentos u ocasiones. La necesidad de evaluar estas diferencias ya se la plantearon a pensadores de la antigüedad clásica, algunos de los cuales elaboraron sistemas de clasificación de las personas que en última instancia, expresaban este deseo de disponer de patrones que permitieran precisar las diferencias existentes entre los individuos. El español J. Huarte de San Juan publicó en el siglo XVI su Examen de ingenios y daba ya normas para descubrir las diversas habilidades e ingenios del hombre y establecer diferencias cualitativas.

Pero los primeros problemas que estimularon el desarrollo de los tests en el campo de la psicología fueron de origen clínico. A mediados del siglo pasado, surge en la mayor parte de los países de cultura occidental y de modo muy concreto en Francia, la necesidad de estudiar y tratar a los enfermos mentales con criterios científicos y, en consecuencia, se planteó la cuestión de la conveniencia de disponer de normas que sirvieran para identificar y clasificar con unas pautas uniformes las diferentes clases de trastornos. El psiquiatra francés Esquirol publicó en 1,838 Des maladies mentales considerées sous les rapports medical Hygienique et medico-legal y de ella daba pautas bastante precisas para clasificar a los débiles mentales.

Esquirol considera que éstos se pueden clasificar, no en comportamientos estancos, sino a lo largo de un continuo que, comenzando en la zona inmediatamente inferior a la inteligencia normal, va descendiendo hasta llegar a la ausencia más completa de inteligencia que sería la idiocia. Este autor da en su obra varias pautas principios clasificatorios, pero considera que, de todos ellos, la observación del lenguaje es la que mejor permite a los débiles mentales. Distingue en estos dos grupos : la idiocia y la imbecilidad. El primero lo subdivide en tres categorías : la primera corresponde a los sujetos que carecen por completo de cualquier modalidad de expresión verbal, la segunda a los que tan sólo emiten algunas monosílabos la tercera a los que son capaces de pronunciar algunas palabras o frases muy cortas. La imbecilidad la subdivide en dos categorías : sujetos con vocabulario muy limitado y sujetos con vocabulario algo limitado, y da pautas para atenerse en lo concerniente a cada uno de estos criterios.

Es conveniente hacer notar que las normas de Esquirol sirvieron luego como punto de partida para la construcción de las primeras escalas de inteligencia (Binet y Simon) y de desarrollo (Gesell) y que, incluso en numerosas escalas para explorar la inteligencia, de reciente creación, se incluye un test de vocabulario.

También en Francia y en esta misma época surge la figura de Seguin, pionero de la rehabilitación de los débiles mentales. Oponiéndose a la idea, entonces muy generalizada, de que el débil mental era un ser absolutamente incurable, fundo en 1,837 la primera escuela que se conoce de este tipo con la finalidad de mejorar a los débiles mentales mediante la realización de ejercicios sensoriales motores. Uno de los tableros para encaje de piezas que ideó Seguin ha llegado hasta nuestros días formando parte de ciertas baterías de tests ejecutivos (escala de Grace Arthur) y en cierto modo ha contribuido a perpetuar el nombre de su autor.

VII.2.2.b.2 La influencia de la Psicología Experimental

En 1,879 Wundt abre en Leipzig las puertas del primer laboratorio de psicología experimental y este hecho iba también a influir indirectamente en el desarrollo del movimiento psicométrico. El objetivo de los primeros psicólogos experimentales fue la formulación de leyes sobre la conducta humana, particularmente en lo tocante a las esferas de la sensación y la percepción y en lo referente al estudio de los tiempos de reacción. La escuela de Leipzig fue de las primeras en plantearse la necesidad de realizar experimentos psicológicos con rigurosos controles, ya que, de no hacerlo así, era imposible llegar a establecer deducciones correctas. El rigor en los procedimientos utilizados, en el registro y cuantificación de los resultados y en su interpretación, sirvió para que los primeros creadores de tests concibieran sus métodos con idéntico rigor experimental. Asimismo, el impacto de esta escuela es el desarrollo de los tests psicológicos y se percibe que los primeros tests que se pusieron a punto fueron principalmente de tiempo sensorial y motor

James Mc. Keen Cattell, norteamericano y discípulo de Wundt, fue quien utilizó por primera vez el término **test mental** para designar a esta modalidad de examen psicológico, y lo hizo en un artículo publicado en 1,890 en la revista *Mind* y titulado *Mental Test and measurements*. Los test mentales serán para Cattell una serie de pruebas psicológicas que el mismo puso a punto con la finalidad de determinar el nivel intelectual. El término **test mental** se ha venido utilizando hasta nuestros días; sin embargo, durante los últimos años se percibe una tendencia a substituir esta denominación por la de **test o prueba psicométrica**, de acepción más amplia, e incluso simplemente por **test**.

Cattell, después de sus estudios en Leipzig y antes de regresar a Norteamérica, trabajó en Londres durante algún tiempo como colaborador de Galton. Este biólogo inglés fue otro estimulador del movimiento psicométrico. Uno de sus principales objetivos fue el estudio de los grados de semejanza y diferencia entre miembros de las mismas familias y, para ello, ideó una serie de pruebas de discriminación sensorial y motriz, con las cuales pensó que podrían establecer diferencias cualitativas y cuantitativas entre las diversas facetas de los procesos intelectivos. Pero su contribución al desarrollo de los tests no solamente fue metodológica, sino que aún fue más importante su aportación en la esfera de la estadística, ya que ideó numerosos procedimientos para el análisis de datos de diferencias individuales. Galton, en colaboración con su discípulo y futuro gran estadístico Person, fue también el fundador de la revista *Biometrika*. Pues bien, Cattell, al regresar a Norteamérica, construyó unas baterías de tests siguiendo las huellas de Galton y las aplicó en Nueva York a grupos de estudiantes de la Universidad de Columbia, mas los resultados fueron muy poco satisfactorios. Ciertamente las pruebas de discriminación sensorial y de tipo motor discriminaban bien entre sujetos deficientes mentales y normales, pero cuando se pretendía discriminar entre grupos de sujetos de nivel normal o

superior, los tests mentales de Cattell demostraron tener muy poco poder discriminativo. A consecuencia de ello, se planteó la necesidad de usar algún tipo de pruebas diferente que se relacionaran más directamente con los procesos mentales superiores.

VII.2.2.b.3 Los test de procesos mentales superiores

Esta labor es iniciada simultáneamente por Kraepelin Ebbinghaus y Oehrn en Alemania, por Guicciardi y Ferrari en Italia, Binet y Henri en Francia, que empiezan a utilizar el método de los tests, pero dando al contenido de las pruebas un enfoque distinto.

En 1,895, Kraepelin el padre la psiquiatría, va a ser el primero en idear pruebas para explorar funciones más complejas, como la susceptibilidad a la distracción o a la fatiga mental, la elaboración de conceptos, el enjuiciamiento práctico, etc. Ebbinghaus en 1,897 da a conocer pruebas para explorar la memoria de fijación, la facilidad para el cálculo aritmético y para completar frases inconclusas. Cuando se aplicaron estas pruebas a algunos grupos de estudiantes, ya se encontró con que había una relación directa entre los resultado y el grado de madurez escolar. Los italianos Guicciardi y Ferrari siguieron las huellas de su maestro Kraepelin y se interesaron particularmente en la creación de pruebas adecuadas para la exploración de los enfermos mentales.

También Binet y Henri, en su trabajo titulado La Psychologie individuelle publicado en 1,895, critican la mayor parte de los tests hasta entonces utilizados, considerándolos demasiado sensoriales y concentrados casi exclusivamente en el estudio de funciones simples, y proponen una serie de tests para explorar la memoria, la imaginación, la atención, la comprensión, la discriminación de formas estéticas, de sugestibilidad, etc. Estos tests fueron el punto de partida de los que luego el mismo Binet utilizaría para elaborar sus escalas de inteligencia que tanta difusión iban a tener en seguida.

Al comienzo del siglo XX hay que destacar dos fechas de suma importancia en la historia de los tests psicológicos. En 1,904 el psicólogo inglés Charles Spearman publica en el *American Journal of Psychology* un artículo titulado General Intelligence objectively determined and measured, calculando los coeficientes de correlación entre series de resultados diversos tests, y sienta las bases del análisis factorial, método que iba a dar un gran impulso al desarrollo de la psicometría y que posteriormente continuarían, modificarían y perfeccionarían otros investigadores como Burt, Kelley y Thurstone. La influencia de estos métodos no se hizo sentir solamente en el progreso que aportarían al conocimiento de los elementos integrantes de la inteligencia y posteriormente de la personalidad, sino también en la construcción de tests. Sin embargo, la elaboración de tests que permitirán medir independientemente una serie de aptitudes mentales. El primer grupo de trabajos arranca de la obra de Spearman y de la escuela factorialista inglesa, el segundo procede principalmente de las investigaciones llevadas a cabo por Thurstone y otros psicólogos americanos, y se desarrolló un decenio después, dando lugar a la construcción de diversas baterías de aptitudes tal y como se han descrito a partir de los resultados de los análisis factoriales.

La segunda fecha a resaltar es la del año 1,905 en el que apareció en la revista L'année psychologique, el trabajo de A. Binet y Th. Simon titulado: Méthodes nouvelles pour diagnostique du niveau intellectuel des anormaux. Esta escala fue el resultado de la propuesta que formalizó el ministerio francés de Instrucción Pública en 1,904 para que se crearan métodos de diagnostico de la debilidad mental para su aplicación en las escuelas de París.

En 1,908 apareció la primera revisión de esta escala, que en muy pocos años llamó la atención de los psicólogos de todo el mundo, en tal forma, que pronto empezaron a realizarse adaptaciones y revisiones de la misma. Una de las más

importantes fue la llevada a cabo bajo la dirección de Terman en Norteamérica, en la Universidad de Stanford, que vio la luz en 1,916. Esta revisión es conocida mundialmente con el nombre de Escala de Stanford-Binet y fue ella donde se introdujo por primera vez el criterio mensurativo de coeficiente intelectual, o relación entre la edad mental y la edad cronológica que tanta difusión y arraigo iba a alcanzar. En el año 1,937 también Lewis Terman, esta vez en la colaboración con Maud Merrill, llevó a cabo una nueva revisión que ha alcanzado gran difusión y que comúnmente se denomina Terman-Merrill o Nueva Revisión Stanford. Esta revisión fue adaptada en seguida al español por J. Germain y M. Rodrigo.

De la escalas de Binet y Simon arrancaron todo un numeroso grupo de test de inteligencia, tanto para adultos como para niños. Otro avance importante en la historia de los tests de inteligencia es la aparición en 1,939 de la Escala Wechsler Bellevue, en la que por primera vez se llevó a cabo un estandarización de un test para adultos en función de su edad cronológica. Esta escala, así como sus ampliaciones y revisiones, adquirieron una amplia difusión, y procede en parte de las escalas tipo Binet y en parte de las escalas por puntos elaborados en 1,905 por Yerkes-Bridges y Hardwick, que fueron asimismo punto de partida de otro grupo numeroso de tests que, por ser de aplicación colectiva, los vamos a considerar aparte.

VII.2.2.b.4 Los test colectivos

Otro importante acontecimiento en la historia de la psicometría lo constituye la aparición de tests adecuados para examinar colectivamente a grupos de personas. Fue la primera guerra mundial el suceso que plantearía la necesidad de elaborar urgentemente este tipo de tests. En efecto, en 1,917, los Estados Unidos se vieron obligados a reclutar un ejército con gran rapidez y a seleccionar los mandos sin disponer previamente de reservas. Las escalas de Binet y otros test a la sazón existentes eran pruebas que podían ser aplicadas solamente individuo por individuo y que, en consecuencia, exigían una considerable cantidad de tiempo del que no se disponía en tales circunstancias y a la vez un numeroso equipo de personas especialmente preparadas para su correcta administración. En consecuencia, el Ministerio del Ejército nombró un Comité de psicólogos (del que formaban parte Terman, Yerkes, Otis, Yoakum y otros) que elaborarían unos tests adecuados para poderse administrar simultáneamente a muchas personas y en los que iba además a simplificar extraordinariamente su administración y evaluación. Los tests del ejército Army Tests (puesto que así se denominan estas pruebas que todavía se utilizan en la actualidad) comprenden dos formas : la alfa y la beta. La forma alfa se componía de varios tests verbales y la beta de pruebas no verbales elaboradas con diversos dibujos, con la finalidad de poderla aplicar a sujetos no familiarizados con la lengua inglesa e incluso a semialfabetos. Con estos test se examinó a cerca de millón y medio de reclutas.

Una vez terminada la guerra, los test del Ejército se revisaron y perfeccionaron y pronto se empezó a aplicarlos a la población civil. A consecuencia de las ventajas que se derivaban de la posibilidad de aplicación colectiva de tests, esta nueva modalidad adquirió un gran auge durante los años siguientes, concibiéndose numerosos tests para diversas edades y fines.

VII.2.2.b.5 Los tests de Personalidad

Con cierto retraso pero paralelamente al desarrollo de los tests de inteligencia fueron apareciendo tests adecuados para explorar los aspectos no intelectivos de la personalidad, es decir, los aspectos caracterológicos. El primer método que apareció fue el cuestionario Hoja de datos personales, de Woodworth, que se compuso también durante la primera guerra mundial con fines de reclutamiento y con la finalidad de economizar el enorme gasto que exigían las

entrevistas individuales. El método de los cuestionarios experimentó pronto un gran auge y difusión y, hoy día, existe una considerable cantidad de instrumentos de este tipo, concebidos para explorar rasgos de comportamiento social, emocional, perturbaciones patológicas, intereses profesionales, actitudes, etc. Aproximadamente durante los últimos 20 años, la influencia del análisis factorial es perceptible también en el desarrollo de estos métodos, existiendo en la actualidad varios cuestionarios que se han construido atendiéndose a los resultados de investigaciones factoriales.

De otra parte, la difusión del psicoanálisis ha ejercido una considerable influencia en este sector, y favoreció la creación de ciertas pruebas que, en mayor o menor grado, se basan en los postulados de psicoanálisis o de la psicodinámica en la más amplia acepción del término. Tal es el caso de las llamadas técnicas proyectivas, por ejemplo, el T.A.T. de Murray que apareció en 1,942 o el test de frustración de Rosenzweig publicado en 1,944. Sin embargo, es el Psicodiagnóstico elaborado por el psiquiatra suizo H. Rorschach, y publicado en 1,921, el que, sin duda alguna, ha alcanzado más difusión y el que, a la vez, ha provocado más controversias.

VII.2.3 Clasificación de los tests

Existen numerosas posibilidades de clasificar los tests en función de los diversos criterios que al respecto se pueden adoptar:

VII.2.3.a Según el modo de administrarlos

Los tests se clasifican según este criterio, de la siguiente forma:

Individuales
Autoadministrados y colectivos

Se denomina test individual al que exige la presencia de una persona para administrar el test, sin que sea posible su aplicación simultánea a varios sujetos. Este tipo de test en general tiene unas instrucciones más complejas que los tests colectivos y es preciso que el examinador las llegue a dominar bien. Quien administra un test individual tiene la posibilidad de establecer un contacto directo con el sujeto que va a pasar la prueba y puede actuar de tal forma que logre su máxima cooperación. Un contacto tan directo no se puede establecer cuando se aplican tests colectivos. La situación de examen individual permite además al examinador hacer observaciones más finas y precisas sobre la forma de responder del sujeto, su actitud y reacciones emocionales, etc. lo cual puede reportar valiosas informaciones complementarias al resultado estricto del examen psicométrico. Numerosas pruebas de inteligencia, como las Escalas Binet, las de Wechsler, las de Grace Arthur y la que requieren manipulación de aparatos o piezas y numerosas técnicas proyectivas como el T.A.T o el Rorschach, están concebidas para su administración individual.

Los tests autoadministrados generalmente tienen unas instrucciones breves y sencillas, concernientes a la tarea que se va a realizar por los sujetos y a la forma de registrar respuestas. A veces estas instrucciones las da verbalmente el propio examinador, mientras que en otros tests ya figuran impresas en el material de examen, en cuyo caso aquél se limita a comprobar si el sujeto las ha comprendido, para lo cual éste debe realizar previamente unos ejemplos similares a los items que constituyen el test propiamente dicho. Este tipo de tests permite su administración colectiva, lo que supone una economía enorme de tiempo y de aquí que estas modalidades de tests se utilicen siempre que haya que examinar a grupos de varios sujetos, como por ejemplo, sucede cuando se hace un examen psicométrico con fines de clasificación o selección. Muchas baterías factoriales de aptitudes mentales, como los P.M.A. de Thurstone, el Multiple Aptitude Test, otras pruebas de inteligencia general como los Tests del Ejército (Army Tests),

así como los cuestionarios de personalidad, no sólo pueden aplicarse colectivamente, sino que en gran parte se construyen teniendo en cuenta esta necesidad.

VII.2.3.b Según el modo de expresión

Los tests, se pueden clasificar en:

- verbales
- impresos
- gráficos
- manipulados

Test verbales son aquellos en que el sujeto nos da de palabra sus respuestas y el material del test no es de tipo perceptivo-visual y manipulativo (por ejemplo, los test que componen la escala verbal de las Escalas de Inteligencia de Wechsler o la técnica proyectiva denominada Fábulas de Duss. Los tests impresos se denominan en los países sajones Tests papel-lapicero y en ellos el sujeto responde por escrito en un impreso a las cuestiones que el test plantee y que pueden ser de naturaleza muy diversa; resolución de problemas, contestación a las preguntas de un cuestionario de personalidad, contestación a tests de vocabulario o de conocimientos, etc. Los tests gráficos son aquellos en que el sujeto tiene que realizar algún trazado, grafismo o dibujo. Por último, los tests manipulativos, frecuentemente denominados de performance, requieren la manipulación de algún material.

VII.2.3.c Según el sector que explore el test

Con arreglo a este criterio los tests pueden clasificarse en:

- Tests de eficiencia
- Tests de personalidad

VII.2.3.c.1 Tests de eficiencia

Estudian los aspectos intelectivos, y cognoscitivos de la personalidad y, según el aspecto que ellos abordan, se puede subdividir este grupo de las siguiente forma:

- Test de inteligencia
- Test de aptitudes
- Test de conocimientos y aprovechamiento

Los test de inteligencia y los de aptitudes se distinguen de los tests de conocimientos y de aprovechamiento en que aquéllos miden aspectos del funcionamiento del intelecto en su mayor parte independientes (aunque no totalmente) de sus adquisiciones culturales. Los tests de conocimiento y aprovechamiento miden las adquisiciones culturales del sujeto en determinadas áreas o materias.

VII.2.3.c.2 Tests de personalidad

Son aquellos que miden las características de las personalidad propiamente dichas, como por ejemplo, la estabilidad emocional, la sociabilidad, los intereses, las actitudes, etc. Podemos distinguir los siguientes subgrupos:

Métodos de observación normalizados
Entrevistas normalizadas
Escalas de estimación (rating scales).

Cuestionarios de personalidad.
Tests de conducta objetivos.
Métodos expresivos.
Tests proyectivos.

Hay una diferencia esencial entre los tests de eficiencia y los de personalidad. En los primeros existe un criterio completamente objetivo para evaluar como buena o como mala una respuesta. Por ejemplo, a la pregunta : **A veinte pesetas cada una, ¿cuánto cuestan tres entradas de cine?** La respuesta buena no puede ser más que sesenta. En principio no sucede lo mismo con las respuestas a los tests de personalidad. A la pregunta : **¿Cuando usted está en una reunión de personas. tiende a quedar callado y aislado?** Tanto la respuesta si como la respuesta no en sí mismas no son ni buenas ni malas. Solamente poniéndolas en relación con un criterio de introversión o extraversión, adquirirán un verdadero significado y la respuesta puede ser buena o mala en función de los fines para los que se aplique el test, por ejemplo, selección de sujetos con valores altos en una escala de extroversión (supongamos vendedores) o con notas altas en una escala de introversión; supongamos investigaciones para trabajos de laboratorio.

Evidentemente la clasificación anterior no incluye ni agota todos los métodos de exploración de la personalidad. Los hemos restringido exclusivamente a los tests, pero existen otros métodos como los observación directa de la conducta, el psicoanálisis, el estudio biográfico, la entrevista e incluso muchas técnicas proyectivas que, aunque tengan menor rigor científico que los tests, pueden adoptar datos valiosísimos y a menudo imprescindibles en el estudio de la personalidad.

VII.2.4 Aplicaciones de los Tests

Los tests se aplican con la finalidad de formular pronósticos o diagnósticos y también como medios de investigación.

El psicólogo en su trabajo cotidiano tiene que hacer pronósticos en multitud de situaciones. Aplicamos unos tests de inteligencia y aptitudes a unos escolares con la finalidad de predecir si están o no capacitados para cursar unos determinados estudios o para seleccionar, dentro de un grupo de aspirantes a ocupar un puesto laboral, aquellos que por ser más inteligentes, estan mejor preparados o tener una personalidad más adecuada podrán desempeñar las funciones propias de este trabajo con mayores posibilidades de éxito.

A menudo se solicitan los servicios del psicométrista con fines diagnósticos : **¿Cuál es el nivel de inteligencia de este paciente?, ¿es un neurótico?, ¿tiene rasgos psicopáticos?, ¿presenta signos de deterioración mental?, ¿cuáles son sus intereses profesionales?** Si bien las funciones de diagnóstico y pronóstico van entre sí muy entrelazadas cuando aplicamos tests con fines diagnósticos, lo que están directamente relacionadas con ciertos esquemas o perfiles diferenciados y que, por lo tanto, permitan formular diagnósticos diferenciales. Evidentemente, a partir de estos datos podemos también hacer pronósticos y, al diagnosticar un oligofrénico, podremos predecir que tendrá dificultades para realizar determinados estudios o desempeñar ciertas actividades profesionales.

Los tests se pueden utilizar también para realizar investigaciones. Aquí los resultados de los tests pueden servir como elemento de control de las variables experimentales **¿se produce pérdida de la memoria de fijación en pacientes sometidos a electrochoques? ¿Mejoran las aptitudes espaciales después**

de un curso en el que se introducen métodos pedagógicos con esta finalidad?, ¿disminuye la ansiedad y se mejora la adaptación global después de un tratamiento psicoterápico?, etc. También se pueden utilizar los tests como medios de descripción de las características humanas : ¿son más inteligentes los sujetos de tal raza en comparación con los de otra?, ¿con los pícnicos más sociables que los leptosomáticos?, ¿cuáles son los rasgos de carácter más frecuentes en quienes desempeñan la profesión de correctores de pruebas?, ¿qué aptitudes mentales, sensoriales y qué rasgos de personalidad deben de tener los pilotos de aviación? He aquí algunos, de entre los muchos ejemplos, que se podrían aducir en lo concerniente a problemas de investigación psicológica en los que la aplicación de tests es imprescindible. Finalmente, los test de pueden utilizar también para verificar hipótesis acerca de la relación existente entre ciertos tipos de conducta: El autoritarismo del padre ¿favorece la aparición de rasgos de indecisión o de rebeldía en los hijos?; los niños a los que desde su primera infancia se les separa de sus padres y se crían en instituciones ¿son luego más inhibidos o torpes que los niños que viven en familia?; una educación más excesivamente perfeccionista ¿determina la aparición de neurosis obsesivas?, etc.

La difusión que ha alcanzado durante los últimos cuarenta años el método de los tests es uno de los fenómenos más notorios de nuestros tiempos en la esfera de las ciencias humanas. Esta difusión no está, sin embargo, exenta de algunos peligros y ha dado lugar, a menudo, a que ante los tests se tomen actitudes pasionales, a veces de aceptación incondicional, a veces de rechazo u oposición sistemática. Estas actitudes se ven frecuentemente favorecidas por la utilización de tests por personas insuficientemente formadas y a veces notoriamente incompetentes, que pretenden poder describir la perfección la personalidad de los individuos y su comportamiento mediante tests. No es raro que tales personas sean aceptadas, en tanto que seleccionadoras de personal, orientadores profesionales y psicoterapeutas, sin averiguar previamente qué títulos académicos ostentan y hasta qué punto tales títulos les capacitan para realizar este tipo de trabajos; el excesivo crédito que inicialmente se les concede por la llamada al elemento mágico que a menudo utilizan, acaba luego despertando actitudes de hostilidad y de rechazo hacia la psicometría, lo cual no sucedería si dispusiéramos de resortes jurídicos para evitar el intrusismo profesional tan frecuente en cualquiera de los campos de la psicología aplicada.

Se puede afirmar, además, que ha sido en gran parte gracias al método de los tests cómo la psicología ha ido adquiriendo consideración e importancia dentro de los sectores científicos en los que hasta hace poco tiempo se la había considerado más bien como una disciplina meramente especulativa.

VII.2.5 Cualidades de los Tests : fiabilidad y validez

Los tests psicológicos están compuestos por una serie de ítems (cada uno de los que se compone un test). Como instrumentos de medida tienen que tener unas características : fiabilidad y validez. Además un test debe estar normalizado o tipificado.

Un test es fiable cuando al aplicarlo dos o más veces a un mismo individuo o grupo de individuos en circunstancias similares obtenemos resultados análogos. Existen varios métodos para calcular la fiabilidad de un test; los más adecuados son los siguientes: Método test_retest, consiste en pasar dos veces el mismo test. Método de series paralelas consiste en pasar dos o más test paralelos, o sea equivalentes. Método de división en dos mitades que estriba en, una vez administrado el test, dividir los elementos en dos partes equivalentes, puntuarlas por separado y calcular su correlación que pretendía medir. La validez de un test no está en función del mismo, sino del uso a que va a destinarse. Se denomina coeficiente de validez al coeficiente de correlación resultante de un proceso de validación.

Al construir un test hay que tener también en cuenta la validez aparente; se designa con este término no a lo que el test mide sino a lo que, a primera vista, parece que mide y hay que tomarlo en consideración en vistas a la aceptación del test por parte de los examinados.

Se han utilizado diversos procedimientos para averiguar la validez de un test; la elección de uno u otro depende de los fines a que se destina el test, de sus características y de otros factores. Se pueden distinguir cinco tipos de validez.

1. Validez del contenido:

Es la que tiene en cuenta principalmente en tests de conocimientos o de madurez o aprovechamiento escolar. Los ítems de estos tests deben ser representativos y los procedimientos para controlar su validez se basan en el análisis de los mismos.

2. Validez predictiva:

Está en relación con el grado de probabilidad de un test para predecir algún aspecto de la conducta de un sujeto y nos vendrá dada por el coeficiente de correlación entre las predicciones efectuadas partiendo del test y los resultados de la ulterior conducta del sujeto, que han servido para construir el criterio.

3. Validez concurrente:

Es el grado de correlación que existe entre la varianza de un test y la varianza de los notas de un criterio tomando ambos datos al mismo tiempo. Los criterios que generalmente se utilizan para hacer validaciones concurrentes son la utilización de grupos de contraste y utilización de distribución de calificaciones.

4. Hipótesis de trabajo:

El proceso de validación, con arreglo a este criterio, requiere formular una hipótesis, que ésta se pueda generalizar, encontrar algún procedimiento que permita ponerla a prueba y que exista correlación entre los resultados obtenidos con este procedimiento y las manifestaciones de la conducta a que hace referencia la hipótesis.

5. Validación Factorial:

Es la que trata de validar la composición factorial de un test, y se determina mediante un análisis factorial.

Existen problemas especiales en la validación de los tests; uno de ellos es el que se refiere a la validez generalizada, es decir, cuando la validez de un test se puede generalizar a otras situaciones similares a las que sirvieron para validar el test en un principio. Otro es el que se refiere a la validez cruzada.

VII.2.6 Tipificación, sistemas de medida, normas

Los resultados que se obtienen al administrar los tests se expresan mediante puntuaciones que se denominan directas o brutas, pero es preciso convertir estas puntuaciones en algún tipo de puntuación tipificada. Entendemos por puntuación tipificada estándar o normativa a aquélla en la que están determinados los intervalos de un modo fijo y preciso y, en consecuencia, nos

permite expresar la posición de cada sujeto dentro de una curva de distribución en un lenguaje de valor universal.

Los sistemas de medida más importantes son los que toman como punto de partida la edad del sujeto, los percentiles y la puntuaciones típicas.

La puntuaciones en relación con la edad son : La Edad Mental y el Coeficiente de Inteligencia. La edad mental es un sistema de medida introducido por Binet basándose en la hipótesis de que, al menos durante la infancia y la adolescencia, a medida que aumenta la edad cronológica, aumenta también el nivel de inteligencia. A este nivel de inteligencia en función de la edad es a lo que se denomina edad mental. El coeficiente de inteligencia es un sistema de medida ideado por el psicólogo alemán Stern y que consiste en dividir la edad mental por la edad cronológica y multiplicarlo por 100.

Los percentiles es uno de los sistemas de medida que más se utilizan. Se denomina percentil al punto de la distribución por encima y por debajo del cual se sitúa un determinado tanto por ciento del grupo normativo y utilizando como medida de comparación la desviación típica de la distribución o alguna fracción de ella. La modalidades de **puntuaciones típicas** más importantes son :

Notas Z : la media se iguala a 0 y la desviación estándar a 1.

Notas T : la media equivale a 50 y la desviación estándar a 10.

Notas CEEB : la media en 500 y la sigma equivalente a 100.

Notas AGCT : la medida se iguala a 100 y la desviación estándar a 20.

Eneatipos : consiste en dividir la población normal en 9 grupos o enatipos; excepto para; ps eneatipos 9 y 1, cada eneatipo corresponde a 1/2 sigma.

Pentas : en este sistema se toma por unidad la sigma. La penta 3 correspondiente a la media.

Coeficientes de inteligencia de la Escala de Wechsler : con unos coeficientes de inteligencia estándar en los que la media se fija arbitrariamente en 100 y la desviación típica en 15.

Cuando aplicamos un test, se evalúan los resultados comparándolos con una normas; hay que tener en cuenta que en estas normas de un test, el psicólogo debe plantearse una serie de cuestiones, entre ellas, si el grupo normativo del test es análogo al de los sujetos que examina él y si dicho grupo normativo es verdaderamente una muestra representativa de aquella población.

VII.2.7 La elaboración de un Test

Para elaborar un test es necesario :

1. Plantearse con la mayor precisión posible qué objetivo de querer alcanzar; para ello se delimitará claramente el área o rasgo de la conducta que se desea medir.
2. Elegir los elementos o ítems más adecuados al fin que nos hayamos propuesto; estos elementos pueden ser variadísimos en cuanto a su contenido.
3. Hacer un análisis de los elementos. Este se hará para averiguar las características de éstos en cuanto a su dificultad, su consistencia interna y su validez. Averiguaremos la dificultad de cada elemento en los tests de inteligencia, aptitudes y conocimientos (en los de personalidad no tendría sentido) teniendo en cuenta que los ítems de cada test no han de ser tan fáciles que los resuelva todo el mundo ni tan difíciles que no los resuelva nadie o casi nadie; el índice

de dificultad más adecuado es el próximo al 50%. Averiguaremos la consistencia interna de cada ítem calculando su correlación con la totalidad del test; cuanto mayor sea esta correlación, tanto más este elemento medirá lo que mide el test en su conjunto. Averiguaremos su validez buscando la correlación de la puntuación de cada elemento (cuando éstos son heterogéneos) o de un agrupación de elementos (cuando éstos son homogéneos) con un criterio exterior o con una dimensión factorial previamente determinada.

4. Determinar cuál será la longitud más adecuada para el test. Cuanto más reducido es el número de elementos de un test más probabilidades hay de su fiabilidad sea baja, no obstante, no por ser excesivamente largo aumentará su fiabilidad; el criterio que se va seguir debe tener en cuenta que la longitud ideal de su test viene determinada por el número de ítems mediante los cuales la fiabilidad llegue a un máximo. En los tests que se administran con tiempo límite, lo ideal es que éste sea el que necesitan del 80 al 90% de los sujetos para finalizarlo.
5. Escoger un procedimiento de puntuación. Este debe ser rápido y seguro. Su elección la basaremos en criterios de fiabilidad y validez, escogiendo aquellos procedimientos que mejoren al máximo éstas sin menoscabo del tiempo o complejidad de la puntuación y la elaboración de la nota total del test.
6. Validar y normalizar el test. Una vez dado al test su formato definitivo (es decir, una vez organizados los elementos seleccionados, decidido si el test debe administrarse con tiempo límite o no, escogido el sistema de puntuación, redactadas las instrucciones definitivas para su administración, etc) escogemos un sistema de tipificación de la respuestas y, tras probar su fiabilidad en función de determinados criterios, procederemos a su normalización en los grupos de personas que se estimen adecuados.

VII.2.8 La aplicación y uso de los tests

Para que los resultados que se obtengan mediante la aplicación de test sean fiables, es preciso que se cumplan una serie de condiciones:

1. Deben administrarlos e interpretarlos las personas que estén capacitadas y preparadas para ello. Existen tests de muy fácil aplicación, pero otros que requieren una sólida preparación y formación, como por ejemplo, la mayor parte de técnicas proyectivas.
2. Deben aplicarse en un medio ambiente adecuado. Este debe reunir condiciones mínimas de comodidad y tranquilidad (deben evitarse ruidos, conversaciones, interrupciones; la luz y la temperatura deben ser adecuada, etc).
3. Hay que obtener la cooperación del examinado consiguiendo que colabore de buen grado en el examen y rinda satisfactoriamente si se trata de tests de inteligencia, aptitudes o conocimientos, y sea lo más sincero posible si se trata de test de personalidad.
4. Seguir fielmente las instrucciones dadas por el manual del test (aunque no con extrema rigidez) Estas se refieren a su administración y evaluación.
5. Registrar las respuestas con la mayor exactitud posible. Esto es muy sencillo en los test de administración colectiva, pero en los de administración individual es veces más complicado.

6. Puntuar debidamente las respuestas. Para ello deben seguirse las normas que figuran en el manual del test. Las respuestas pueden puntuarse a mano o a máquina.
7. Convertir las puntuaciones directas en una puntuación normalizada mediante las tablas que figuren en el manual del test e interpretar la puntuación.

La aplicación de los test presenta varios problemas. Los principales son los siguientes:

1. **Ansiedad:** muchas personas opinan que una situación de examen provoca gran ansiedad que repercute en los resultados de los test. Se han hecho varios experimentos sobre este aspecto sin que se haya podido probar que la ansiedad repercute tan negativamente en el rendimiento, dado que los resultados probablemente no sean fiables. No obstante, es necesario que el administrador de un test actúe de forma que no incremente la ansiedad latente que puedan experimentar los examinados, sino que, en lo posible, la disminuya.
2. **Incentivos y motivación:** se ha dicho que los motivos de los sujetos para pasar el examen pueden producir modificaciones en sus respuestas. En cuanto a la eficiencia, existen estudios que han probado que el elogio puede influir positivamente en los rendimientos, mientras que la censura, al menos a la larga, influye negativamente. En cuanto a la personalidad debemos distinguir entre los casos en que, el sujeto se somete al examen por propia voluntad y aquellos en que más o menos se ve obligado a aceptarlo. En el segundo aspecto, las actitudes del examinando pueden ser tan variada que pueden repercutir en los resultados si el examinador no motiva adecuadamente a los sujetos.
3. **Falseo de resultados:** en los tests de eficiencia el falseo de los resultados se puede producir debido a que el sujeto copie, lo cual se remedia con una discreta pero eficaz vigilancia; puede también el examinador conocer previamente el test (y esto puede influir sobre todo en aquellos test en que el aprendizaje interviene en su resolución) puede darse también el caso de que el sujeto quiera aparentar una inteligencia deficiente, lo cual se subsanará pasando una amplia batería (con lo cual aparecerán generalmente discrepancias) y sobre todo con la habilidad y experiencia del examinador; en los tests de personalidad el problema se centra principalmente en los cuestionarios, ya que en ellos el sujeto suele percatarse del tipo de mejores respuestas para alcanzar un determinado fin. Para controlar las posibilidades de fraude se han introducido en los cuestionarios ciertos procedimientos, los más importantes de los cuales son los siguientes: Valoración empírica de las escalas, introducción de escalas de mentiras, construcción de cuestionarios de elección forzada e introducción de escala con variables supresivas.

VII.3 Desarrollo

VII.3.1 Introducción

La última revolución tecnológica, comúnmente llamada la era de la información, debe ser distinguida de las tempranas tecnologías agrícola e industrial.

Primero, a la agricultura son las herramientas, y la industria las máquinas; ambas sustituyen la fuerza muscular por la fuerza mecánica. La tecnología de la información eleva el poder intelectual; esto no es nuevo. El abaco ha estado con nosotros desde hace veintiséis siglos, que hace diferencia en esto como un salto infinito en la extensión del poder intelectual, que nos da este lujo con la tecnología de la información.

Segundo, la velocidad a la cual la tecnología es expandida y desarrollada puede ser en término de meses y días comparado a los miles de años, que la revolución industrial necesitó, que respectivamente tomó para nacer y crecer. Verdaderamente dentro de pocas décadas gobiernos y comercio en el mundo de la industria dependerán de los rápidos cambios de la computación y de nuevas formas de telecomunicaciones (transmisiones por satélite, cables de fibra óptica) que comunicaran las computadoras.

La nueva era de la información hace imperativo que los pueblos menos desarrollados se actualicen con los avances de la tecnología por su propio interés. Esto significa la liberación a los regímenes de inversión, protección a los derechos de propiedad intelectual y no restringir los servicios del comercio.

Estos pueblos también necesitan ayuda para avanzar en la nueva era de la información. Mejores políticas públicas a solas no resuelven el problema. Los pueblos menos desarrollados necesitan ayuda para entrenar a su gente, construir las instituciones y la infraestructura, preparándolos para que por sí mismos continúen el avance tecnológico.

El uso de la era de la información es ubicua: el control de inventores, costo, financiamiento o mercadeo, planeamiento, transporte, control computarizado del servicio eléctrico, robots; computadoras que controlan herramientas mecánicas e investigación; de hecho, es difícil imaginar todos los usos que pueden beneficiarse con la tecnología de la información, muchos de los cuales dependen preliminarmente de la capacidad computacional.

En breve, la tecnología de la información evolucionará y esto será crítico para los pueblos menos desarrollados. El desarrollo es un proceso particular de cada pueblo y cada uno debe establecer sus propios requerimientos, métodos, y medir sus pasos. Sin embargo, la estrategia y políticas de los pueblos deben también ser sensibles a su ambiente externo. La mayor parte convendrá que el desarrollo tendrá y continuará en el manejo del poder de una buena política económica y el avance de la tecnología.

Esta relación es reflejada por ambos en la historia y las recientes tendencias. Las políticas pueden constriñir o promover el desarrollo de la economía. Por ejemplo, esto se dirige al reconocimiento del crecimiento del desarrollo de la economía y al valor de las políticas que permiten la productividad, propiedad de la sociedad a ser propietaria por las privatización y el uso de ganancia privada. Se puede ver el impacto de los cambios en China en los recientes años. También, en la antigua Unión Soviética hoy Unión de Estados Independientes, es ahora en el penoso esfuerzo de cambiar las políticas del pasado para resurgir con una gran productividad, por ejemplo arrendamiento de las tierras a finqueros individuales arriba de los cincuenta años.

Ciertamente la política gubernamental ha ayudado y reforzado los progresos económicos y muchos de los pueblos menos desarrollados van a terminar en realidad sus actuales políticas que impiden el actual desarrollo de la era de la información.

Es interesante observar el registro de desarrollo de los pueblos desarrollados en la incorporación de la tecnología en su busca de una economía desarrollada.

Primero, el tener que ser célebre por las circunstancias de la competencia internacional y el avance económico, han cambiado para los pueblos menos desarrollados conforme la tecnología ha avanzado.

La tecnología avanzada ha traído un gradual declive en alguno de los pueblos menos desarrollados, ya tradicionales, en comparativa ventaja.

VII.3.2 Definición del sistema

VII.3.2.a Elaboración de la prueba

El objetivo de aplicación es la implementación de un manejador de pruebas del coeficiente intelectual y/o pruebas similares; las aplicaciones para el manejo de las pruebas será desarrollado en un lenguaje de tercera generación 3GL (lenguaje C++); dichas pruebas podrán ser tomadas por personas que pueden tener poca relación con computadoras, ya que el interfase entre el usuario y la computadora, será desarrollado de la manera más amigable posible usando todos los recursos que nos ofrece una computadora (Monitor, Teclado, Impresora etc).

La aplicación de las pruebas podrá ser ejecutada en computadoras personales, compatibles con IBM y sistema operativo DOS.

Se contará con menús en pantalla que permitirán evaluar, crear, modificar y eliminar pruebas. En cada una de las opciones se tendrán pantallas de ingresos y despliegues de datos; se realizarán las verificaciones en cada opción, como por ejemplo la existencia de pruebas y/o usuarios que ya han sido ingresados, por lo que no serán ingresados nuevamente; esto es con el fin de mantener la información correcta y no repetida de todas las pruebas y/o usuarios; dichos usuarios serán las personas que tomarán las pruebas ingresadas por los psicometristas o expertos en el campo. Todas las pruebas serán identificadas por un código único; al igual que las personas que se evalúen, deberán tener un Código o Carnet para poder ser identificados por el sistema (pudiendo ser el código la Cédula)

Las pruebas ingresadas con las preguntas de las cuales están compuestas, serán elaboradas por personal capacitado que únicamente tendrá que ingresar la pregunta, posibles respuestas, respuesta correcta, tiempo estimado de respuesta en segundos, opcional, así como de una puntuación por cada pregunta, como por ejemplo :

Ejemplo 1 :
pregunta

¿ Cual de las palabras que están abajo corresponde a lo que es una manzana ?

1. flor
2. árbol
3. verdura
4. fruta
5. animal

respuesta 4
tiempo de respuesta 80
puntuación 1

Ejemplo 2 :

pregunta

Un pie es a un hombre y una garra es a un gato lo que un casco es a

1. un perro
2. un caballo
3. un zapato
4. un herrero
5. una silla de montar

respuesta 2

tiempo de respuesta 100

puntuación 1

Ejemplo 3 :

pregunta

¿ Cuánto cuestan 6 lápices a cuarenta centavos cada uno ?

respuesta 2.40

tiempo de respuesta 120

puntuación 2

El proceso de ingreso de la prueba se realizará mientras existan preguntas que se vayan a ingresar. Dicho proceso se realiza accedando el menú con la opción Pruebas, luego de terminar de ingresar las preguntas y sus datos asociados, el usuario podrá llamar a la prueba que será tomada por cualquier persona para quién este orientada; con esto, el supervisor de la prueba podrá escoger las distintas pruebas que hayan sido ingresadas con anterioridad.

Con lo antes descrito, se observa que no estamos forzados a manejar un solo tipo de prueba, lo cual hace más flexible la aplicación del sistema, y así poder utilizar distintas pruebas.

VII.3.2.b Toma de la prueba

Antes de que una persona proceda a tomar una prueba podrá hacer uso de un programa de ayuda, el cual le indicará cuáles son las teclas que se van a usar en su localización dentro del teclado, así como la ayuda general, en cualquier momento del desarrollo de la prueba. Dentro de dicha aplicación de ayuda, se tocaran los tópicos en los que el usuario desee alguna explicación, menús, respuestas, teclas, derechos del usuario, etc.

Sobre los tipos de test que se van a manejar serán test autoadministrados, definición que se dio anteriormente en la parte sobre teoría psicométrica.

La persona que tome la prueba deberá ingresar, sus datos, los cuales serán almacenados para luego poder generar algún reporte sobre los resultados obtenidos; si una persona hubiese tomado una prueba con anterioridad y procediera a tomar otra, no será necesario ingresar todos sus datos ya que el sistema los desplegara con sólo ingresar su Carnet o Código y luego continuará con la prueba que deberá seleccionar.

Las preguntas serán desplegadas una a una, las cuales incluyen número de pregunta, pregunta, posibles respuestas según el caso. La respuesta ha ingresar será el índice o sea el número de las posibilidades que se presenten. Si la

persona no deseara contestar podrá saltar a la siguiente pregunta y continuar; también si la persona que se está evaluando sobrepasa el tiempo límite de la prueba, automáticamente será desplegado un mensaje de tiempo terminado. Si por alguna razón, se desea cancelar la prueba se podrá salir.

VII.3.2.c Reportes

Acercas de los reportes, éstos serán generados de forma individual o sobre un grupo de personas que hayan tomado el mismo tipo de tesa; esto significa que se creará una base de datos que almacenaran los resultados de los tests tomados y se acumularán, para luego obtener los reportes que se deseen.

En vista que la aplicación podrá ser ejecutada en varias computadoras, será necesario juntar todos los datos que se vayan generando y podrán ser gradados a una base de datos general sobre la cual se trabajará.

VII.3.2.d Supervisores

Los supervisores son las personas encargadas de administrar el sistema; para los supervisores se creará un módulo de claves, las cuales tendrán asociados los datos personales de los mismos; esto es con el objeto de mantener privacidad sobre las pruebas, así como de los resultados que se puedan obtener; aquí deberán ingresarse datos generales de las personas que sean designadas como supervisores.

Para que usuarios y supervisores puedan manejar la aplicación, se desarrollaran manuales del usuario, los cuales tendrán las especificaciones de todo lo que se maneje en la aplicación.

VII.3.3. Diseño

En esta parte se hace referencia a las especificaciones necesarias para el diseño de la aplicación, estructura y configuración de archivos, pantallas, menús, ingresos y salidas.

VII.3.3.a Archivos

Este es un método de organización de archivos, e intenta explotar la capacidad proporcionada por las unidades de disco y dispositivos similares de lograr acceso a cualquier bloque de dirección conocida. Transforman la llave (código, carnet, etc) mediante un algoritmo de computación antes de utilizarla como dirección. Este método es rápido, ya que evita las operaciones intermedias, pero obligan a que los datos se localicen de acuerdo con un solo atributo llave. Su uso se puede tener en directorios, tablas de precios, programación de tiempos, listas de nombres, etc.

Los archivos que se manejarán en esta aplicación serán de tipo directo; esto quiere decir que los registros que éstos contienen son de acceso directo; por medio del código o carnet; con esto se logra un acceso rápido a la información almacenada.

VII.3.3.b Pantallas

Las pantallas son todo aquello que se presenta en el área de la pantalla al usuario, entre las cuales podemos mencionar :

1. Pantalla de presentación :

Esta presentará al usuario el nombre de la aplicación la fecha en la que se desarrollo y el logotipo del desarrollador.

2. Pantalla de menús :

Aquí se presentan al usuario los nombres de las teclas para hacer llamadas a las distintas opciones del menú, así como las distintas opciones del mismo.

3. Pantalla de ingresos :

En estas pantallas se harán los despliegues de datos, como las preguntas, así como los ingresos de las preguntas, datos de usuarios y pruebas.

4. Pantalla de reportes :

Según sea el reporte generado, se tendrán generalmente los siguientes datos:

Tipo de reporte
Datos de la institución
Fecha y hora de la generación
Datos de la persona evaluada (dependiendo del reporte)
Resultados generados por el reporte

A continuación se presentan algunas de ellas:

Universidad San Carlos de Guatemala Facultad de Ingeniería	Manejador de Pruebas del C.I. Freiry Javier Gramajo López	MENU
<u>Pruebas</u> Análisis Reportes Migrar Salir Ingreso de pruebas y evaluación		

F1=Ayuda <- Enter=Introduce

Figura 1

Universidad San Carlos de Guatemala Manejador de Pruebas del C.I. Facultad de Ingeniería Freiry Javier Gramajo López	TESTS
Evaluar Crear Modificar Eliminar Permite tomar una prueba	

F1=Ayuda <- Enter=Introduce

Figura 2

Universidad San Carlos de Guatemala Manejador de Pruebas del C.I. Facultad de Ingeniería Freiry Javier Gramajo López	EVALUA
CARNET = NOMBRE = DIRECCION = TELEFONO = EDAD = FECHA = XX/XX/XX HORA = XX:XX:XX	
<- Enter=Introduce Esc=Sale	

Figura 3

Universidad San Carlos de Guatemala Facultad de Ingeniería	Manejador de Pruebas del C.I. Freiry Javier Gramajo López	INGRES
PREGUNTA # = PREGUNTA = POSIBLES RESP. = RESPUESTA = TIEMPO/seg = PUNTUACION =		
<- Enter= Introduce Esc =Sale		

Figura 4

VII.3.3.c Menús

Para el manejo de los menús, sólo será necesario mover el cursor con las teclas de movimientos (arriba, abajo, izquierda y derecha).

Los menús serán de acceso inmediato; esto quiere decir que al posicionar el cursor en alguna opción de algún menú sólo será necesario presionar (ENTER <- o INTRO) y automáticamente nos presentara un sub-menú o la pantalla correspondiente de ingreso o salida según sea el caso. Para seleccionar alguna otra opción sólo se moverá el cursor y repetir el procedimiento anterior, si se desea salir de un sub-menú se presionará ESC (escape) y se retornará al menú anterior, si existiera.

VII.3.3.d Ingresos

Los ingresos no son más que los datos que se escribirán y luego serán utilizados por la aplicación. En el caso de la definición de pruebas será necesario definir los siguientes campos:

Código
 Nombre
 Descripción
 Nivel
 Tiempo Total

Para el ingreso de usuarios (personas que tomarán una prueba) es necesario ingresar

Carnet o código
 Nombre
 Dirección
 Teléfono
 Edad

El sistema generará automáticamente la fecha y hora de la evaluación.

Los datos de los usuarios, definición e ingreso de pruebas y sus respectivas preguntas, respuestas, etc. solamente serán ingresos por teclado.

VII.3.3.d Salida

La salida no es más que el manejo de los resultados que se obtienen al completar una prueba, analizando un individuo o grupo de ellos que hayan tomado una determinada prueba.

VII.3.3.d.1 Pantalla

Gráficas, cuadros y datos representativos tomados de la o las muestras son desplegados en pantalla.

VII.3.3.d.2 Impresora

Todos los resultados son mandados a impresora, si así se desea, con lo cual se obtiene una copia en papel de todos los resultados.

VII.3.5 Implementación

Aquí se procede a programar la aplicación que se va a desarrollar con los requerimientos antes descritos.

El siguiente listado que se presenta consiste en todas las funciones y procedimientos que manejan la pantallas, despliegue de todo tipo de mensajes, pantallas, movimientos, etc.

```

    cursores.h.ah=0x2;
    cursores.h.dh=y;
    cursores.h.dl=x;
    cursores.h.bh=0;
    int86(0x10,&cursores,&cursores);
}

```

```

void borrado()
{ int x,y;
  for (y=0;y<25;++y)
    for (x=0;x<80;++x)
      if (caracter(x,y)==' ' && atributo(x,y)==7)
        { gotoxy(x,y);
          atributo(x,y)=0;
          return;
        }
}

```

```

void linea_car(x,y,c,atr,len)
int x,y;
char c;
int atr,len;
{ int i,tem;
  for (i=0;i<len;++i)
    { tem=x+i;
      caracter(tem,y)=c;
      atributo(tem,y)=atr;
    }
}

```

```

void linea_ver(x,y,c,atr,len)
int x,y;
char c;
int atr,len;
{ int i,tem;
  for (i=0;i<len;++i)
    { tem=y+i;
      caracter(x,tem)=c;
      atributo(x,tem)=atr;
    }
}

```

```

char call(x,y)
int x,y;
{
  return(caracter(x,y));
} /* end of fuction call */

```

```

void beep(void)
{ printf("%c",7);
}

```

```

void input_cam(x,y,hil,len)
int x,y;
char hil[];
int len;
{ int i=-1,t,tem;
  do
    { ++i;
      tem=x+i;
    } while (caracter(tem,y)!=' ');
  do
    { gotoxy(x+i,y);
      t=tecla();
      if (t==17) t=32;
    }
}

```

```

if (t==8 || t==15)
{ if (i==0) beep();
  else
  { --i;
    tem=x+i;
    caracter(tem,y)=' ';
    gotoxy(x+i,y);
  }
}
if (t>31 && t<127)
{ if (i==len) beep();
  else
  { tem=x+i;
    caracter(tem,y)=t;
    ++i;
    gotoxy(x+i,y);
  }
}
} while (t!=13);
for (i=0;i<len;++i)
{ tem=x+i;
  hil[i]=caracter(tem,y);
}
hil[len]='\0';
}

```

```

void cls(x,y)
int x,y;
{ int i,j;
  for (i=x;i<25;++i)
    for (j=y;j<80;++j)
      { caracter(j,i)=' ';
        atributo(j,i)=7;
      }
}

```

```

void init()
{ union REGS regs;
  regs.h.ah=0x0f;
  int86(0x10,&regs,&regs);
  if (regs.h.al==7)
    screen=mono;
  else
    screen=cga;
  cls(0,0);
}

```

```

/* ***** *LIMPIA LA PANTALLA *
***** */

```

```

void clrscr()
{ union REGS p;
  p.h.ah=0x6;
  p.h.al=0;
  p.h.ch=0;
  p.h.cl=0;
  p.h.dh=25;
  p.h.dl=80;
  p.h.bh=7;
  int86(0x10,&p,&p);
  gotoxy(0,0);
}

```

```

void keypress(void)

{
    bdos(1,0,0);
}

void Escr_Linea(c,f,texto)
int c,f;
char *texto;
{
    int i=0;

    while (texto[i]) {
        caracter(c,f) = texto[i];
        c++;
        i++;
    }
}

void Coloc_Atr(c,f,atr,cant)
int c,f,atr,cant;
{
    while (cant) {
        atributo(c,f) = atr;
        c++;
        cant--;
    }
}

int Lee_Tecla()
{
    union REGS v,t;

    v.h.ah = 0x0;
    int86(0x16,&v,&t);
    if (!(t.h.ah))
        return (t.h.ah + 256); /* tecla especial */
    return (t.h.ah); /* caracter normal */
}

void Scroll(n,ci,cd,fs,fi,Pu_Pd,atributo)
int n,ci,cd,fs,fi,Pu_Pd,atributo;
{
    union REGS p;

    p.h.ah = Pu_Pd;
    p.h.al = n;
    p.h.cl = ci;
    p.h.ch = fs;
    p.h.dl = cd;
    p.h.dh = fi;
    p.h.bh = atributo;
    int86(0x10,&p,&p);
}

```

```
/* UNIVERSIDAD SAN CARLOS DE GUATEMALA */
/* FACULTAD DE INGENIERIA */
/* Ingeniería en Ciencias y Sistemas */
/* Trabajo de Tesis */
/* Proyect : Manejador de Pruebas */
/* File : Uno.C/Main */
/* Login : Freiry Gramajo López */
/* Date : 22/VII/94 */
/* End Date : 04/IX /94 */
```

```
#include <stdio.h>
#include <stdlib.h>
```

```
void login(void);
void clrscr(void);
void Menu(void);
```

```
int Machine;
```

```
void login(void )
{
    do {
        printf("Login :!");
        scanf("%d",&Machine); }
    while (Machine>=4 || Machine<=0);
} /* end of procedure login */
```

```
void main(void)
{
    clrscr();
    login();
    Menu();

    exit(0);
} /* end main */
```

```

/* UNIVERSIDAD SAN CARLOS DE GUATEMALA */
/* FACULTAD DE INGENIERIA */
/* Ingeniería en Ciencias y Sistemas */
/* Trabajo de Tesis */
/* Proyect : Manejador de Pruebas */
/* File : DBMS.C */
/* Login : Freiry Gramajo López */
/* Date : 22/VII/94 */
/* End Date : 04/IX /94 */

#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <alloc.h>
#include <stdio.h>
#include <ctype.h>
#include <dos.h>

#define Lapso 800
#define Sonido 700

void parametro(int,char[]);
void imprima(int,int,char[],int);
void gotoxy(int,int);
void cls(int,int);
void keypress(void);
void inicializa(void);
void create_col(char[]);
int ver_ident(char[],char[],int);
void crea(char[],char[],int);
int ver_ident(char[],char[],int);
void openit(char[]);
void fij_reg(void);
int syntax_in(char[]);
void fetch(char[]);
void insert(FILE *,int *);

void camp_writes(void);
void camp_write(void);
void desp_preg(char[]);
void listado(void);
void borra(void);
void sinta(char[]);
void camp_datos(void);
void imprime(void);
void list_col(void);
void delete(void);
void reclas(char[]);
void ingresa(long,FILE *);
/*void ingresa(char[],int); */
void lee_linea (char[],long,FILE *);
void pedazo(char[],int);
void llaves(char[],char[]);
void syntax_up(char[]);
int separa(char[],char[],int *);
int compara(char[],char[]);
int busca(int, int);

/* estructura que mantiene la lista de campos de un
registro */
struct columna {
int id;
char name[35];
char tipo;
int largo;
int util;
struct columna *next;
} info;

struct campos {
int camp_num;
char tipo;
int largo;
char field_a[255];
int ocupa,del;
long apunta;
struct campos *sig;
} camp;

int build (int, char[], char, char [], struct columna **);

void construye(int,struct campos **);
struct columna *Root;
struct campos *R_camp;
FILE *fdirec;

int ONOFF,FGL;
char NAME[15];
long OFFSET,FLAG;
int Reg;
int MM,*N_reg;
int SAL=0;

/* -- FUNCION QUE SE ENCARGA DE INICIALIZAR
VARIABLES GLOBALES -- */

void inicializa(void )
{
ONOFF=0;
Root=NULL;
R_camp=NULL;
FLAG=0L;
N_reg=(unsigned int)0;
} /* end of function inicializa */

/* ---- FUNCION QUE SE ENCARGA DE REVISAR EL
IDENTIFICADOR
DE UN ARCHIVO CON SU EXTENCION CORRECTA
SI LA TRAE ---- */

int ver_ident (LIN,Nom,k)
char LIN[];
char Nom[15];
int k;

{ int j=0;
int re,flag;

while (LIN[k] == ' ' ;LIN[k] == '\t')
{ ++k;}
re = isalpha(LIN[k]);

```

```

if (re!=0)
{
while(j<=8 && re!=0)
{
re=isalnum(LIN[k]);
if (re!=0)
{
Nom[j]=toupper(LIN[k]);
++k;
++j;
Nom[j]='\0';
} /* end of if */
} /* end of while LIN j */
if (j<=8)
{
if (LIN[k]!='.';LIN[k]!='';LIN[k]!='\t';
LIN[k]!='\n';LIN[k]!=';')
{ re=1; }

flag=k;
} /* end of if */
} /* end of if */
if (j>8 || re==0)
{ flag=0; } /* end of if */
return(flag);

} /* end of function ver_atrib */

/* -- FUNCION QUE SE ENCARGA DE INSERTAR EL
REGISTRO EN EL ARCHIVO -- */

void insert(f,n_reg)
FILE *f;
int *n_reg;

{
double dd,val_n=0.0;
int large,i=0;
long Posi;
int rr,re,flag=0;
char carac="";
char linea[255],llave[255];
FGL=1;
large=strlen(R_camp->field_a);

while (i<large)
{ val_n=val_n*26+R_camp->field_a[i];
++i;
} /* end while */

dd=fmod(val_n,MM);
Posi=OFSET+(dd)*Reg;

fseek(f,Posi,SEEK_SET);

carac=getc(f);
if (carac=='&')
{ ingresa(Posi,f);
++(*n_reg);
flag=1;
}
else
{ /* primero hacer un fetch y luego hacer linear
probbing */

```

```

lee_linea(linea,Posi,f);
llaves(linea,llave);
re=compara(llave,R_camp->field_a);
if (re==0)
{ imprima(47,23," Error : Cuenta ya Existente
",120);
sound(Sonido);delay(Lapso);nosound();
FGL=0;
}
else
{
rr=dd;
dd+=1;
if (dd==MM)
{ dd=0; }
while (dd!=rr)
{
Posi=OFSET+(dd)*Reg;
fseek(f,Posi,SEEK_SET);
carac=getc(f);
if (carac=='&')
{
lee_linea(linea,Posi,f);
llaves(linea,llave);
re=compara(llave,R_camp->field_a);
if (re==0)
{
imprima(47,23," Error : Cuenta ya
Existente ",120);

sound(Sonido);delay(Lapso);nosound();
flag=1;
FGL=0;
break;
} /* end if re==0 */
} /* end if carac=='&' */

if (carac=='&')
{ ingresa(Posi,f);
++(*n_reg);
flag=1;
break;
}
else
{ if (dd<MM)
{ ++dd; }
if (dd==MM)
{ dd=0; }

} /* end of else */
} /* end of while */

if (flag==0)
{ imprima(47,23," Tabla Llena no Inserciones
",120);

sound(Sonido);delay(Lapso);nosound();
FGL=0;
}

} /* end of else */
} /* end of else */

} /* end of function */

```

```

/* ----- FUNCION QUE SE ENCARGA DE REVISAR LOS
PARAMETROS ----- */

```

```

void parametro (tipo,LIN)
int tipo;
char LIN[];

```

```

{
  int k,flag=0;
  char Nom[15]="\0";
  FILE *ftemp;

  switch (tipo)

  {
    case 0:
      create_col(LIN);
      break;
    case 1:
      k=ver_ident(LIN,Nom,1);
      if (k!=0)
      {
        crea(Nom,LIN,k);
      }
      else
      { imprima(47,23,"      Error : De
Sintaxis",120); }

      break;

    case 2:
      k=ver_ident(LIN,Nom,1);
      if (k!=0)
      {
        if (ONOFF==0)
        {
          strcat(Nom,".DRT");
          ftemp=fopen(Nom,"r+b");
          if (ftemp!=NULL)
          {
            strcpy(NAME,Nom);
            openit(Nom);
            fil_reg();
            ONOFF=1;
          }
          else
          { imprima(47,23,"      Error : Archivo no
Creado",120);

          sound(Sonido);
          delay(Lapso);
          nosound();

          }
        }
      }
      else
      { imprima(47,23,"Nota : Existe un archivo
",120);

      sound(Sonido);delay(Lapso);nosound();
      } /* end of else */

    } /* end of if */

    break;

```

```

case 3:
  if (ONOFF!=1)
  { imprima(47,23,"      Error : No Hay Arch
abierto",120);

  sound(Sonido);delay(Lapso);nosound();
  }
  else
  {
    flag=sintax_in(LIN);
    if (flag==0)
    {
      insert(fdirec,(int *)&N_reg);
    }
    else
    { imprima(47,23,"      Error : En los
Parametros ",120); }

  } /* end of else */
  break;
case 4 :
  if (ONOFF!=1)
  { imprima(47,23,"      Error : No Hay Arch
abierto",120);

  sound(Sonido);delay(Lapso);nosound();
  }
  else
  {
    fetch(LIN);
  } /* end of else */
  break;

case 5 :
  if (ONOFF!=1)
  { imprima(47,23,"      Error : No Hay Arch
abierto",120);

  sound(Sonido);delay(Lapso);nosound();
  }
  else
  { listado(); }
  break;
case 6 :
  if (ONOFF!=1)
  { imprima(47,23,"      Error : No Hay Arch
abierto",120);

  sound(Sonido);delay(Lapso);nosound();
  }
  else
  {
    if (FLAG!=0)
    {
      sintax_up(LIN);
    }
    else
    { imprima(47,23,"      Error : No Hay
Apuntador ",120);

    sound(Sonido);delay(Lapso);nosound();
    } /* end of if */
  } /* end of else */
  break;

case 7:
  if (ONOFF!=1)
  { imprima(47,23,"      Error : No Hay Arch
Abierto",120);

```

```

        sound(Sonido);delay(Lapso);nosound();
    }
    else
    {
        if (FLAG!=0)
        {
            delete();
            FLAG=0L;
        }
        else
        { imprima(47,23," Error : No Hay
Apuntador ",120);
        sound(Sonido);delay(Lapso);nosound();
        }
    } /* end of else */
    break;

    case 8:
    if (ONOFF!=1)
    { imprima(47,23," Error : No Hay Arch
Abierto",120);
    sound(Sonido);delay(Lapso);nosound();
    }
    else
    {
        reclas(LIN);
        FLAG=0L;
    } /* end of else */
    break;

    case 9:
    if (ONOFF!=1)
    { imprima(47,23," Error : No Hay Arch
Abierto",120);
    sound(Sonido);delay(Lapso);nosound();
    }
    else
    {
        ONOFF=0;
        /* fseek(fdirec,14L,SEEK_SET);

        fprintf(fdirec,"%10u",MM);
        fprintf

(fdirec,"%15.5f,%15.5f,%15.5f,%15.5f)",00.0,00.0,00.0,00.0);
*/

        fclose(fdirec);
        borra();
        FLAG=0L;
    } /* end of else */
    break;

    case 10:
    if (ONOFF!=1)
    { imprima(47,23," Error : No Hay Arch
Abierto",120);
    sound(Sonido);delay(Lapso);nosound();
    }
    else
    { desp_preg("1");

    } /* end of else */
    break;

    case 11:
    list_col();
    break;

```

```

    } /* end of switch */

} /* end of function parametro */

/* ----- FUNCION QUE SE ENCARGA DE ARMAR EL
CATALAGO ----- */

int separa(LIN,parte,k)
char LIN[],parte[];
int *k;

{ int j=0,re;
strcpy(parte,"\0");

++(*k);
while (LIN[*k]== ' ' || LIN[*k]== '\t' || LIN[*k]== '\n')
{ ++(*k);}
re=isdigit(LIN[*k]);
while(LIN[*k]!='.' && LIN[*k]!='\'' && j<32 &&
re!=0)
{
    parte[j]=LIN[*k];
    re=isdigit(LIN[*k]);
    if (LIN[*k]=='.')
    { re=10;
    ++(*k);
    ++j;
    parte[j]='\0';
    } /* end of while LIN!='.' */
    if (parte[j-1]== ' ' || parte[j-1]== '\t' || parte[j-1]== '\n')
    { re=10;
    parte[j-1]='\0';
    }
    if (j>32 || re==0)
    { return(1); }
    else
    { return(0); }
} /* end of parte */

/* ----- */

void create_col(LIN)
char LIN[];

{ int k=0,id;
int bandera=0;
char tipo;
char name[33],largo[30];

strcpy(name,"\0");

bandera=separa(LIN,largo,&k);

if (bandera!=1)
{
    id=atoi(largo);
    ++k;
    while (LIN[k]== ' ' || LIN[k]== '\t' || LIN[k]== '\n')
    { ++k;}
    k=ver_ident(LIN,name,k);

    if (k==0)

```

```

    { bandera=1; }
else
    { /* verifica el valor en el registro */
        ++k;
        while (LIN[k]!=' ' ; LIN[k]!='\t' ; LIN[k]!='\n')
            { ++k; }
        tipo=toupper(LIN[k]);
        if (tipo!='N' && tipo!='A')
            { bandera=1; }
        else
            {
                ++k;
                while ( LIN[k]!=' ' ; LIN[k]!='\t')
                    { ++k; }
                bandera=separa(LIN,largo,&k);

                } /* end of if else */
            } /* end of if else */
        } /* end of if else */
/* se va a llenar la lista dinamica temporal */

    if (bandera!=1)
        { bandera=build(id,name,tipo,largo,&Root); }
    else
        { imprima(47,23," Error : De Sitaxis
",120); }

} /* end of function create col */

```

```

/* ----- FUNCION QUE SE ENCARGA DE ABRIR UN
ARCHIVO ----- */

```

```

void openit (Nom)
char Nom[];

```

```

{
    char *point;
    char Linea[200]="\0";
    char Directo[15]="\0";
    char numero[20];
    int k,re,j;

    fdirec=fopen(Nom,"r+b");
    fseek(fdirec,0L,SEEK_SET);
    point=fgets(Linea,200,fdirec);

    if (point!=NULL)
        {
            strncpy(Directo,Linea,13);
            re=strcmp(Directo,"DifRfEiCfTfO");
            if (re==0)
                {
                    k=13;
                    separa(Linea,numero,&k);

                    MM=atoi(numero);

                    for (j=1;j<=4;++j)
                        { separa(Linea,numero,&k); }

                    do {
                        j=separa(Linea,numero,&k);
                        if (j==0)
                            {

```

```

                                re=atoi(numero);
                                construye(re,&R_camp);
                            }
                        } while (Linea[k]!='&');
                        OFSET=k+1;
                    } /* end of if */
                else
                    { imprima(47,23," Error : Arch. No Directo ",120);
                    } /* end of else */

            } /* end of if point!=NULL */

        } /* end of function openit */

```

```

/* ----- FUNCION QUE SE ENCARGA DE CREAR UN
ARCHIVO ----- */

```

```

void crea(Name,LIN,k)

```

```

char Name[],LIN[];
int k;
{
    FILE *temp;
    struct columna *w1,*w2;
    int id;
    int re=0,key=-1;
    char largo[30];
    char linea[255];
    int i,reg=0;
    long Posi;

```

```

do {
    while(LIN[k]!=' ' ; LIN[k]!='\t')
        { ++k; }
    re=separa(LIN,largo,&k);
    if (re!=1)
        { id=atoi(largo);

            while(LIN[k]!=' ' ; LIN[k]!='\t')
                { ++k; }
            if (LIN[k]!='\n')
                {
                    re=busca(id,key);
                    key=1;
                }
        }
    } while (LIN[k]!='\n') && re!=1);

```

```

if (re!=1)
    { strcat(Name,".DRT");
      if (fopen(Name,"r+b")==NULL)
          {
              temp=fopen(Name,"wr+b");
              fprintf(temp,"%s*","DifRfEiCfTfO");
              fprintf(temp,"%10u",id);

```

```

          fprintf(temp,"%15.5f,%15.5f,%15.5f,%15.5f",00.0,00.0,00.0,0
0.0);

```

```

          w2=Root;
          w1=w2->next;
          while (w2!=0)
              { if (w2->util===-1)
                  { fprintf(temp,"%8d",w2->id);
                    w2->util=0;

```

```

        reg+=w2->largo+1;
        break;
    }
    w2=w1;
    w1=w2->next;
} /* end of while */

w2=Root;
w1=w2->next;
while (w2!=0)
{ if (w2->util==1)
    { fprintf(temp,"%8d",w2->id);
      w2->util=0;
      reg+=w2->largo+1;
    }
    w2=w1;
    w1=w2->next;
} /* end of while */
fprintf(temp,"%c",&);
OFFSET=ftell(temp);
reg+=10;
for(i=0;i<254;i++)
    { linea[i]=' '; }
linea[254]='\0';
i=strlen(linea);

i=0;
while (i<id)
    { fprintf(temp,"%s",linea);
      ++i;
    }

i=0;
while (i<id)
    {
        Posi=OFFSET+(i*reg);
        fseek(temp,Posi,SEEK_SET);
        fprintf(temp,"%c",&);
        ++i;
    }
fclose (temp);
} /* end of if */
else
    { imprima(47,23," Error : Arch. ya Creado
",120); }
else
    { imprima(47,23," Error : De Sintaxis
",120); } /* end of else */

} /* end of function crea */

/* ---- REALIZA UNA BUSQUEDA EN LA LISTA DEL
CATALOGO ---- */

int busca (Num,key)
int Num,key;

{ struct columna *w1,*w2;
int re=1;

w2=Root;
w1=w2->next;

while(w1!=0 && w2->id<Num)
    { w2=w1; w1=w2->next; }

if (w2->id==Num)
    { w2->util=key;
      re=0;
    }
return(re);
} /* end of fuction busca */

/* ----- */

void construye (num,r1)
int num;
struct campos **r1;

{ struct campos *w1,*w2,*w3;

w2=*r1;
w1=w2->sig;

if (w2==NULL)
    { (*r1)=malloc(sizeof(camp));
      if (*r1==0)
          { imprima(47,23," Error : No Hay Memoria
",120);
            exit(0); }
/* encadena el nuevo elemento en la lista */
(*r1)->sig=NULL;
(*r1)->camp_num=num;
(*r1)->tipo="";
(*r1)->largo=0;
strcpy((*r1)->field_a,"\0");
(*r1)->ocupa=0;
}
else
    { while ( w1!=0)
      { w2=w1;
        w1=w2->sig;
      } /* end of while */
      if (w1==0)
          { /* nueva entrada en la lista */
            w3=malloc(sizeof(camp));

            if (w3==0)
                { imprima(47,23," Error : No Hay Memoria
",120);
                  exit(0); }
/* encadena el nuevo elemento */
w2->sig=w3;
w3->camp_num=num;
w3->sig=NULL;
w3->tipo="";
w3->largo=0;
strcpy(w3->field_a,"\0");
w3->ocupa=0;

} /* end of else */
} /* end of else */

} /* end of function construye */

```

```
/* CONSTRUYE UNA LISTA DINAMICA TEMPORAL DE
LOS ATRIBUTOS Y VALORES */
```

```
int build (id,name,tipo,largo,r1)
int id;
char name[];
char tipo;
char largo[];
struct columna **r1;
{
    struct columna *w1,*w2,*w3;
    int flag=0;

    w2=*r1;
    w1=w2->next;

    if (w2==NULL)
    {
        (*r1)=malloc(sizeof(info));
        if (*r1==0)
        { imprima(47,23," Error : No Hay Memoria
",120);
            exit(0);
        }
        /* encadena el nuevo elemento en la lista */
        (*r1)->next=NULL;
        (*r1)->id=id;
        strcpy((*r1)->name,name);
        (*r1)->tipo=tipo;
        (*r1)->largo=atoi(largo);
        (*r1)->util=0;
    }
    else
    {
        if (w2->id>id)
        {
            *r1=malloc(sizeof(info));
            if (*r1==0)
            {imprima(47,23," Error : No Hay Memoria
",120);
                exit(0);
            }
            (*r1)->id=id;
            strcpy((*r1)->name,name);
            (*r1)->tipo=tipo;
            (*r1)->largo=atoi(largo);
            (*r1)->util=0;
            (*r1)->next=w2;
        }
        else
        {
            while ( w1!=0 && w1->id<id)
            { w2=w1;
                w1=w2->next;
            } /* end of while */

            if (w1==0 || (w2->id!=id && w1->id!=id) )
            { /* nueva entrada en la lista */
                w3=malloc(sizeof(info));
                if (w3==0)
                { imprima(47,23," Error : No Hay Memoria
",120);
                    exit(0);
                }
                w2->next=w3; /* encadena el nuevo
```

```
elemento */
```

```
strcpy(w3->name,name);
w3->id=id;
w3->tipo=tipo;
w3->largo=atoi(largo);
w3->util=0;
w3->next=w1;
}
else
{ imprima(47,23," Create_Col Repetido
",120);
    flag=0;
}
} /* end of else */
} /* end of else */

return(flag);
} /* end of function build */

/* --- BORRA LA LISTA DINAMICA TEMPORAL DE
ATRIBUTOS Y VALORES --- */

void dell(raiz)
struct columna *raiz;
{
    struct columna *r1,*r2;

    r2=raiz;
    r1=r2->next;

    while(r2!=NULL)
    { free(r2);
        r2=r1;
        if (r2!=NULL)
        { r1=r2->next;}
    } /* end of while */

    raiz=NULL;

} /* end of function dell */

/* ----- BORRA LA LISTA DINAMICA TEMPORAL DE
CAMPOS ----- */

void borra(void)
{
    struct campos *r1,*r2;

    r2=R_camp;
    r1=r2->sig;

    while(r2!=NULL)
    { free(r2);
        r2=r1;
        if (r2!=NULL)
        { r1=r2->sig;}
    } /* end of while */

    R_camp=NULL;

} /* end of function borra */
```

```

/* --- SE ENCARGA DE REVISAR SINTAXIS PARA
INSERTAR UN REGISTRO --- */

```

```

int syntax_in (linea)
char linea[];

{ struct campos *w1,*w2;
char LN[256];
int k=2,i,j;
int error=1,re,flag;

w2=R_camp;
w1=w2->sig;

while(linea[k] == ' ' || linea[k] == '\t' || linea[k] == "")
{ ++k; }

do {
while(linea[k] == ' ' || linea[k] == '\t')
{ ++k; }
strcpy(LN,"");
i=j=0;
re=1;

switch (w2->tipo)
{ case 'N':
while (re!=0 && linea[k]!='.'
&& linea[k]!=""&&i<20)
{
re=isdigit(linea[k]);
if (linea[k] == '.' || linea[k] == ' ')
{ re=9; }
if (re!=0)
{ LN[i]=linea[k];
++i;
LN[j]='\0';
}
++k;
} /* end of while */

if (linea[k-1] == ' ' || linea[k] == '.')
{ re=9; }
if (re!=0 && i<=w2->largo)
{ strcpy(w2->field_a,LN); }
else
{ error=0; }
break;

case 'A':
flag=0;
while(linea[k]!=""&& j<256)
{
LN[j]=linea[k];
++j;
LN[j]='\0';
if (linea[k] == "\" && linea[k+1] == ','
&& linea[k+2] == "\" )
{ flag=1; }
else
{ flag=0; }
++k;
if (linea[k] == ',' && flag==0)
{ break; }
}
if (j>w2->largo)

```

```

{ error=0; }
else
{ strcpy(w2->field_a,LN); }
break;

} /* end of switch */

w2=w1;
w1=w2->sig;

if (w2==0 && linea[k]!="")
{ error=0; }
++k;
} while(linea[k] != ' ' && error!=0 && w2!=0);

while(linea[k] == ' ' || linea[k] == '\t')
{ ++k; }

if (error!=0 && linea[k] == ' ')
{ return(0); }
else
{ return(1); }

} /* end of function syntax_in */

```

```

/* ----- COMPARA DOS LLAVES PARA QUE PUEDAN SER
INSERTANDAS ----- */

```

```

int compara(llave,name)
char llave[];
char name[];

{ int lon1,lon2;
int re,i;
char Key[21] = "\0";

if (R_camp->tipo == 'N')
{
lon1 = strlen(llave);
lon2 = strlen(name);
if (lon2>lon1)
{
i=0;
strcpy(Key,"");
while(lon2>lon1)
{ Key[i] = ' ';
++i;
++lon1;
Key[i] = '\0';
} /* end of while */

strcat(Key,llave);
re=strcmp(Key,name);
}
else if (lon2<lon1)
{
i=0;
strcpy(Key,"");
while(lon1>lon2)
{ Key[i] = ' ';
++i;

```

```

        ++lon2;
        Key[ij]=\'0\';
    } /* end of while */

    strcat(Key,name);
    re=strcmp(llave,Key);
} /* end of else */
else
    { re=strcmp(llave,name);
    }
}
else
    { re=strcmp(llave,name);
    } /* end of else */

return(re);

} /* end of fuction compara */

```

```

/* ----- IMPRIME LOS NOMBRES DE LOS CAMPOS DEL
ARCHIVO ----- */

```

```
void camp_writes(void)
```

```

{ int px,py;
  struct campos *w1,*w2;

  cls(4,0);
  imprima(5,6," CODIGO : ",10);
  imprima(20,6," ",120);
  px=21; py=6;
  w2=R_camp;
  w1=w2->sig;
  imprima(px,py,w2->field_a,120);
  w2=w1;
  w1=w2->sig;

  imprima(5,9," NOMBRE : ",10);
  imprima(20,9," ",120);
  px=21; py=9;
  imprima(px,py,w2->field_a,120);
  w2=w1;
  w1=w2->sig;

  imprima(5,12," DESCRIPCION : ",10);
  imprima(20,12," ",120);
  px=21; py=12;
  imprima(px,py,w2->field_a,120);
  w2=w1;
  w1=w2->sig;

  imprima(5,16," NIVEL : ",10);
  imprima(20,16," ",120);
  px=21; py=16;
  imprima(px,py,w2->field_a,120);
  w2=w1;

  imprima(40,16," TIEMPO TOTAL/min : ",10);
  imprima(60,16," ",120);
  px=61; py=16;
  imprima(px,py,w2->field_a,120);

} /* end of function */

```

```

/* ----- IMPRIME LOS NOMBRES DE LOS CAMPOS DEL
ARCHIVO ----- */

```

```
void camp_write(void)
```

```

{ struct campos *w1,*w2;
  int i=5;

  w2=R_camp;
  w1=w2->sig;
  do {
    imprima(i,14,w2->field_a,10);
    w2=w1;
    w1=w2->sig;
    i+=10;
  } while (w2!=NULL);

} /* end of function camp_write */

```

```

/* -- LOCALIZA EL REGISTRO QUE CUMPLE CON LA
LAVE DE BUSQUEDA -- */

```

```
void fetch(llave)
char llave[];
```

```

{ char key[256];
  int i=1,j=0,large,re,flag=0;
  double dd,val_n=0.0,rr;
  long Posi;
  char linea[256];

  strcpy(key,\'0\');
  while(llave[i]==\' \' || llave[i]==\'t\' || llave[i]==\'')
    { ++i; }

  while(llave[i]!='&& j<256)
  {
    key[j]=llave[i];
    ++j;
    ++i;
    key[j]=\'0\';

  } /* end of while */

  while(llave[i]==\' \' || llave[i]==\'t\' || llave[i]==\'')
    { ++i; }
  large=strlen(key);
  i=0;
  while (i<large)
  {
    val_n=val_n*26+key[i];
    ++i;
  }

  dd=fmod(val_n,MM);
  Posi=OFSET+(dd)*Reg;
  fseek(fdirec,Posi,SEEK_SET);

  lee_linea(linea,Posi,fdirec);
  i=sintax_in(linea);
  re=compara(key,R_camp->field_a);
  if (re==0)
  { if (SAL==0)
    {
      camp_datos();
    }
  }
}

```

```

",10);          imprima(47,23," <--- Enter=Continuar          } while (dd!=rr);

                gotoxy(75,23);
                keypress();
            }
            else if (SAL==2)
            {
                camp_writes();
                imprima(47,23," <--- Enter=Continuar
",10);
                gotoxy(75,23);
                keypress();
            } /* end else if */
            strset(NAME,'\0');
            strcat(NAME,"(");
            strcat(NAME,R_camp->sig->field_a);
            strcat(NAME,R_camp->field_a);
            strcat(NAME,")");
            FLAG=Posi;
        }
        else
        {
            rr=dd;
            dd+=1;
            if (dd==MM)
            { dd=0;}

            do {
                Posi=OFSET+(dd)*Reg;
                lee_linea(linea,Posi,fdirec);
                i=sintax_in(linea);

                re=compara(key,R_camp->field_a);
                if (re==0)
                {

                    if (SAL==0)
                    {
                        camp_datos();
                        imprima(47,23," <--- Enter=Continuar
",10);

                        gotoxy(75,23);
                        keypress();
                    }
                    else if (SAL==2)
                    {
                        camp_writes();
                        imprima(47,23," <--- Enter=Continuar
",10);

                        gotoxy(75,23);
                        keypress();
                    } /* end else if */
                    strset(NAME,'\0');
                    strcat(NAME,"(");
                    strcat(NAME,R_camp->sig->field_a);
                    strcat(NAME,R_camp->field_a);
                    strcat(NAME,")");
                    FLAG=Posi;
                    flag=1;
                    break;
                }
            }
            if (dd<MM)
            { ++dd; }
            if (dd==MM)
            { dd=0;}

                } /* ----- IMPRIME LOS NOMBRES DE LOS CAMPOS DEL
                ARCHIVO ----- */
                void camp_datos(void)

                { int px,py;
                struct campos *w1,*w2;

                imprima(5,6," CARNET      :",10);
                Imprima(20,6,"          ",120);
                px=20; py=6;
                w2=R_camp;
                w1=w2->sig;
                imprima(px,py,w2->field_a,120);
                w2=w1;
                w1=w2->sig;

                imprima(5,9," NOMBRE      :",10);
                imprima(20,9,"          ",120);
                px=20; py=9;
                imprima(px,py,w2->field_a,120);
                w2=w1;
                w1=w2->sig;

                imprima(5,12," DIRECCION  :",10);
                imprima(20,12,"          ",120);
                px=20; py=12;
                imprima(px,py,w2->field_a,120);
                w2=w1;
                w1=w2->sig;

                imprima(5,15," TELEFONO   :",10);
                Imprima(20,15,"          ",120);
                px=20; py=15;
                imprima(px,py,w2->field_a,120);
                w2=w1;
                w1=w2->sig;

                imprima(35,15," EDAD       :",10);
                imprima(50,15,"          ",120);
                px=50; py=15;
                imprima(px,py,w2->field_a,120);
                w2=w1;
                w1=w2->sig;

                imprima(5,18," FECHA     :",10);
                Imprima(20,18,"          ",120);
                px=21; py=18;
                imprima(px,py,w2->field_a,120);
                w2=w1;
                w1=w2->sig;

                imprima(35,18," HORA      :",10);

```

```

    imprima(50,18,"      ",120);
    px=51; py=18;
    imprima(px,py,w2->field_a,120);

} /* end of function */

/* ----- IMPRIME LOS NOMBRES DE LOS CAMPOS DEL
ARCHIVO ----- */
void camp_preg(void)

{ int px,py;
  struct campos *w1,*w2;

  imprima(5,8," PREGUNTA # : ",10);
  imprima(20,8," ",120);
  px=20; py=8;
  w2=R_camp;
  w1=w2->sig;
  imprima(px,py,w2->field_a,120);
  w2=w1;
  w1=w2->sig;

  imprima(5,10," PREGUNTA : ",10);
  imprima(5,12,"
",120);
  px=5; py=12;
  imprima(px,py,w2->field_a,120);
  w2=w1;
  w1=w2->sig;

  imprima(5,15," POSIBLES RESP. : ",10);
  imprima(28,15," ",120);
  px=28; py=15;
  imprima(px,py,w2->field_a,120);
  w2=w1;
  w1=w2->sig;

  imprima(5,18," RESPUESTA : ",10);
  imprima(20,18," ",120);
  px=20; py=18;
  imprima(px,py,w2->field_a,120);
  w2=w1;
  w1=w2->sig;

  imprima(30,18," TIEMPO/seg : ",10);
  imprima(45,18," ",120);
  px=45; py=18;
  imprima(px,py,w2->field_a,120);
  w2=w1;
  w1=w2->sig;

  imprima(55,18," PUNTUACION : ",10);
  imprima(70,13," ",120);
  px=70; py=18;
  imprima(px,py,w2->field_a,120);

} /* end of function */

/* ----- DESPLIEGA LAS PREGUNTAS Y REALIZA LA
PRUEBA ----- */

void desp_preg(key)
char key[];

{ int i,large,re;
  double val_n=0.0,dd;
  long Posi;
  char linea[256];

  large=strlen(key);
  i=0;
  while (i<large)
  { val_n=val_n*26+key[i];
    ++i;
  } /* end of while */

  val_n=dd=fmod(val_n,MM);

  do {
    Posi=OFFSET+(val_n)*Reg;
    fseek(fdirec,Posi,SEEK_SET);
    lee_linea(linea,Posi,fdirec);
    re=sintax_in(linea);
    if (re==0)
    {
      cls(4,0);
      camp_preg();
      imprima(47,23,"      <---| Enter=Continuar
",10);
      keypress();
      FLAG=Posi;
    }
    val_n++;
    val_n=fmod(val_n,MM);
  } while (val_n!=dd);

} /* end of function desp_preg */

/* ----- LISTADO DE LAS PRUEBAS
----- */
void listado(void)
{
  char key[5]="\0";
  double dd=1.0;
  long Posi;
  char linea[256];
  int flag;

  strcpy(key,"0\0");

  do {
    Posi=OFFSET+(dd)*Reg;
    lee_linea(linea,Posi,fdirec);
    flag=sintax_in(linea);

    if (flag==0)
    {
      if (SAL==0)
      {
        imprime();
        camp_writes();
        imprima(47,23,"      <---| Enter=Continuar
",10);
        gotoxy(75,23);
        keypress();
      }
      FLAG=Posi;
    }
  }
}

```

```

        ++dd;
    } while (dd <= MM);

} /* end of function */

/* ----- IMPRIME LOS ATRIBUTOS
----- */

void imprime(void)

{ struct campos *w1,*w2;
  struct columna *q1,*q2;
  int i=5;

  w2=R_camp;
  w1=w2->sig;

  while (w2!=0)
  {
    q2=Root;
    q1=q2->next;

    while(q1!=0 && q2->id!=w2->camp_num)
      { q2=q1; q1=q2->next; }

    if (q2->id==w2->camp_num)
      { imprima(i,12,q2->name,120); }

    w2=w1;
    w1=w2->sig;
    i+=10;
  } /* end of while */

} /* end of function imprime */

/* --- LISTA TODOS LOS DATOS DEL DICCIONARIO
SECUENCIALMENTE --- */

void list_col(void)

{ struct columna *q1,*q2;

  q2=Root;
  q1=q2->next;

  printf("\n\t\t COLUMNA NOMBRE TIPO
LONGITUD\n");
  while (q2!=0)
  {
    printf("\n\t\t");
    printf("%8d %15s %c",q2->id,q2->name,q2->tipo);
    printf(" %10d",q2->largo);
    q2=q1;
    q1=q2->next;

  } /* end of while */

} /* end of function */

/* ----- Llena ----- */

void fil_reg(void)

```

```

{ struct campos *w1,*w2;
  struct columna *q1,*q2;

  w2=R_camp;
  w1=w2->sig;

  Reg=0;

  while (w2!=0)
  {
    q2=Root;
    q1=q2->next;

    while(q1!=0 && q2->id!=w2->camp_num)
      { q2=q1; q1=q2->next; }

    if (q2->id==w2->camp_num)
      {
        w2->tipo=q2->tipo;
        w2->largo=q2->largo;
        Reg+=w2->largo+1;
      }
    w2=w1;
    w1=w2->sig;
  } /* end of while */
  Reg+=10;

} /* end of function */

/*---- FUNCION QUE SE ENCARGA DE BORRAR UN
REGISTRO EN EL ARCHIVO ---- */

void delete(void)

{
  fseek(fdirec,FLAG,SEEK_SET);

  fprintf(fdirec,"%c",&);

} /* end of function delete */

/* ----- FUNCION QUE SE ENCARGA DE
REORGANIZAR EL ARCHIVO ----- */

void reclas(linea)
char llinea[];

{ int k=1,j=0,re=1;
  char num[15]="\0",letra="";
  FILE *temp;
  long posi;

  while(ilinea[k]!='\t' || ilinea[k]!='\t' || ilinea[k]!='\n')
    { ++k; }

  while(ilinea[k]!='\n' && ilinea[k]!='\n' && j<10 && re!=0)
  {
    num[j]=ilinea[k];
    re=isdigit(ilinea[k]);
    ++k;
    ++j;
    num[j]='\0';
  } /* end of while */

```

```

while(linea[k] == ' ' || linea[k] == '\t' || linea[k] == "")
{ ++k; }

if (re!=0 && linea[k] == '\n')
{
    re=atoi(num);
    if (re<=(unsigned int)N_reg || re==MM)
        { printf("\t\tERROR: '\O\' < '\N' \n"); }
    else
    {
        pedazo(num,re);
        temp=fopen(num,"r+b");
        N_reg=0;

        j=0;
        while (j<MM)
        {
            posi=OFFSET+(j*Reg);
            fseek(fdirec,posi,SEEK_SET);
            letra=getc(fdirec);
            if (letra=='&')
            {
                lee_linea(linea,posi,fdirec);
                sintax_in(linea);
                insert(temp,(int *)&N_reg);
            } /* end of if */
            ++j;
        } /* end of while */
        MM=re;
        remove(NAME);
        rename(num,NAME);

    } /* end of else */
}
else
{ printf("\t\tERROR: DE SINTAXIS\n"); }

} /* end of function reclas */

/* ----- */

void ingresa(posi,f)
long posi;
FILE *f;

{ struct campos *w1,*w2;

fseek(f,posi,SEEK_SET);
w2=R_camp;
w1=w2->sig;

do {
    fprintf(f,"%s",w2->field_a);
    w2->ocupa=0;
    strcpy(w2->field_a,"\0");
    w2=w1;
    w1=w2->sig;
    if (w2!=NULL)
        { fprintf(f,":"); }
    else
        { fprintf(f,":"); }

} while (w2!=NULL);

} /* end of ingresa */

/* ----- */

void pedazo(num,re)
char num[];
int re;

{ FILE *temp;
int i=0;
char linea[256];
struct campos *w2,*w1;
long posi;

strcpy(num," SOS.DRT");
temp=fopen(num,"wr+b");
fseek(temp,0L,SEEK_SET);

fprintf(temp,"%s","Diffricctio");
fprintf(temp,"%i0u",re);

fprintf(temp,"%15.5f,%15.5f,%15.5f,%15.5f",00.0,00.0,00.0,0
0.0);

w2=R_camp;
w1=w2->sig;
while (w2!=0)
{ fprintf(temp,"%8d",w2->camp_num);
w2=w1;
w1=w2->sig;
} /* end of while */
fprintf(temp,"%c",&);

for(i=0;i<254;i++)
{ linea[i]=' '; }
linea[254]='\0';
i=strlen(linea);

i=0;
while (i<re)
{ fprintf(temp,"%s",linea);
++i;
}

i=0;
while (i<re)
{
    posi=OFFSET+(i*Reg);
    fseek(temp,posi,SEEK_SET);
    fprintf(temp,"%c",&);
    ++i;
}
fclose(temp);

} /* end of function */

/* ----- */

void lee_linea (linea,posi,f)
char linea[];
long posi;
FILE *f;

{ char letra="";

```

```

int i=0;
fseek(f, posi, SEEK_SET);
strcpy(linea, "\0");
linea[0] = '\0';
linea[1] = '\0';
i=2;

letra=fgetc(f);
while(letra!='\0' && i<Reg+2)
{
    linea[i]=letra;
    ++i;
    linea[i]='\0';
    letra=fgetc(f);
} /* end of while */
linea[i]='\0';
linea[i+1]='\0';
linea[i+2]='\0';

} /* end of function */

/* --- SE ENCARGA DE REVISAR SINTAXIS PARA
INSERTAR UN REGISTRO --- */

void llaves (linea, LN)
char linea[], LN[];

{ int k=2, i, j;
  int re, flag;

  while(linea[k] == ' ' || linea[k] == '\t' || linea[k] == '\n')
  { ++k; }

  strcpy(LN, "\0");
  i=j=0;
  re=1;

  switch (R_camp->tipo)
  { case 'N':
    while (re!=0 && linea[k] != '\0' && i<20)
    {
        re=isdigit(linea[k]);
        if (linea[k] == '.')
        { re=0; }
        if (re!=0)
        { LN[i]=linea[k];
          ++i;
          LN[i]='\0';
        }
        ++k;
    } /* end of while */

    break;

  case 'A':
    flag=0;
    while(flag==0 && j<254)
    {
        LN[j]=linea[k];
        ++j;
        LN[j]='\0';
        if (linea[k] == '\n' && linea[k+1] == '\0' &&
            linea[k+2] == '\0')
            break;
    }
}

{ flag=1; }
else
{ flag=0; }
++k;
if (linea[k] == '\0' && flag==0)
{ break; }
} /* end of while */
break;

} /* end of function llaves */

/* ----- SE ENCARGA DE ACTUALIZAR UN REGISTRO
EN EL ARCHIVO ----- */

void syntax_up (linea)
char linea[];

{ struct columna *w1, *w2;
  struct campos *q2, *q1;
  int error=1, flag=1;
  int k=1, j, i, re;
  char nombre[32];
  char LN[256], line[256];

  while(linea[k] == ' ' || linea[k] == '\t' || linea[k] == '\n')
  { ++k; }

  while(linea[k] != '\0' && error!=0)
  {
      while(linea[k] == ' ' || linea[k] == '\t' || linea[k] == '\n')
      { ++k; }

      strcpy(nombre, "\0");
      j=0;

      while(linea[k] != '\0' && j<32)
      {
          nombre[j]=toupper(linea[k]);
          ++j;
          ++k;
          nombre[j]='\0';
          if (linea[k] == ' ' || linea[k] == '\t')
          { break; }
      } /* end of while */

      if (j>31)
      { imprima(47,23, " Error : De Sintaxis ", 120);
        error=0;
      }
      else
      {
          if (nombre[j-1] == '\0')
          { nombre[j-1]='\0'; }

          w2=Root;
          w1=w2->next;

          while(w2!=0)
          {
              if ((re=strcmp(w2->name.nombre)) == 0)
              {
                  q2=R_camp;
              }
          }
      }
  }
}

```

```

q1=q2->sig;
while(q1!=0 && q2->camp_num<w2->ld)
{ q2=q1; q1=q2->sig; }

if (w2->ld==q2->camp_num)
{ q2->ocupa=0;
break;}
}
w2=w1;
w1=w2->next;
} /* end of while */

if (w2==0)
{ imprima(47,23," Error : Campo no
Reconocido ",120);
flag=0;
error=0;
}
else
{ re=1;
++k;
i=j=0;
strcpy(LN,"\0");

if (linea[k]===' ')
{ ++k;}

switch (q2->tipo)
{ case 'N':

if (q2->ocupa!=1)
{
while (re!=0 && linea[k]!=''
&& linea[k]!='"&&i<20)
{
re=isdigit(linea[k]);
if (linea[k]=='.')
{ re=9;}
if (re!=0)
{ LN[i]=linea[k];
++i;
LN[i]='\0';
}
++k;
} /* end of while */

if (linea[k-1]===' ' && linea[k]=='.')
{ re=9;}
if (re!=0 && i<=q2->largo)
{ strcpy(q2->field_a, LN);
q2->ocupa=1;}
else
{ error=0;}
}
else
{ error=0; }

break;

case 'A':

if (q2->ocupa!=1)
{
flag=0;

```

```

while(linea[k]!='"&& j<256)
{
LN[j]=linea[k];
++;
LN[j]='\0';
if (linea[k]=='\ ' && linea[k+1]==','
&& linea[k+2]=='\ ')
{ flag=1;}
else
{ flag=0;}
++k;
if (linea[k]==',' && flag==0)
{ break;}
}
if (j>q2->largo)
{ error=0;}
else
{ strcpy(q2->field_a, LN);
q2->ocupa=1;
}
}
else
{ error=0;}
break;

} /* end of switch */
} /* end of else */

} /* end of while */
if (error==0)
{
if (flag!=0)
{ imprima(47,23," Error : En Actualizacion
",120);
sound(Sonido);delay(Lapso);nosound();
}
else
{
lee_linea(line,FLAG,fdirec);
sinta(line);
} /* end of if else */
} /* end of function sintax_up */

/* --- SE ENCARGA DE REVISAR SINTAXIS PARA
INSERTAR UN REGISTRO --- */

void sinta (linea)
char linea[];

{ struct campos *w1,*w2;
char LN[256];
int k=2,i,j;
int re,flag;

w2=R_camp;
w1=w2->sig;

if (R_camp->ocupa!=1)
{

```

```

while(linea[k]== ' ' || linea[k]== '\t' || linea[k]== "")
{ ++k; }

do {
while(linea[k]== ' ' || linea[k]== '\t')
{ ++k; }
strcpy(LN, "\0");
i=j=0;
re=1;

switch (w2->tipo)
{ case 'N':

if (w2->ocupa!=1)
{
while (re!=0 && linea[k]!='.'
&& linea[k]!='" && i<20)
{
re=isdigit(linea[k]);
if (linea[k]== '.')
{ re=9; }
if (re!=0)
{ LN[i]=linea[k];
++i;
LN[i]='\0'; }
++k;
} /* end of while */

if (linea[k-1]== ' ' && linea[k]== '.')
{ re=9; }
strcpy(w2->field_a, LN);
}
else
{ while(linea[k]!='.' && linea[k]!='"')
k++; }
break;

case 'A':

if (w2->ocupa!=1)
{
flag=0;
while(linea[k]!='" && j<256)
{
LN[j]=linea[k];
++j;
LN[j]='\0';
if (linea[k]== '\ ' && linea[k+1]== '.'
&& linea[k+2]== '\ ")
{ flag=1; }
else
{ flag=0; }
++k;
if (linea[k]== '.' && flag==0)
{ break; }
}
strcpy(w2->field_a, LN);
}
else
{ while(linea[k]!='.' && linea[k]!='"')
k++; }
break;

} /* end of switch */

w2=w1;
w1=w2->sig;

++k;
} while(linea[k]!=' ' && w2!=0);
ingresa(FLAG, fdirec);
}
else
{ imprima(47,23, " Error : En Actualizacion ", 120);
sound(Sonido); delay(Lapso); nosound();
}
} /* end of function sinta */

```

```

/* UNIVERSIDAD SAN CARLOS DE GUATEMALA
*/
/* FACULTAD DE INGENIERIA */
/* Ingeniería en Ciencias y Sistemas */
/* Trabajo de Tesis */
/* Project : Manejador de Pruebas */
/* File : Menu.C */
/* Login : Freiry Gramajo López */
/* Start Date : 22/VI/94 */
/* End Date : 10/IX /94 */

```

```

#include <dos.h>
#include <stdio.h>
/* #include <string.h> */
#include <stdlib.h>

#define USAC "Universidad San Carlos de
Guatemala"
#define CEDE "Facultad de Ingeniería"
#define PROP "Manejador de Pruebas del C.I."
#define SERIE "Freiry Javier Gramajo López"

#define Sonido 440
#define Sound1 200
#define Lapso 800
#define Click 50
#define Touch 20
#define True 1
#define False 0

```

```

int tecla(void);
void init(void);
void cls(int,int);
void centre(int,char[],int);
void cuadro(int,int,int,int,char[],int);
void imprima(int,int,char[],int);
void gotoxy(int,int);
void linea_ver(int,int,char,int,int);
void beep(void);
void keypress(void);
void cambiaH(int,int,int,int);
void linea_car(int,int,char,int,int);
char call(int,int);
void inicializa(void);
void parametro(int, char[][]);
int rotulo(char[],int *,int);
int entrada(char[],int *,int);
int trabaja(void);
int pruebas(void);
int analisis(void);
int evalua(void);
int pregunta(char[],int *,char[]);

```

```

extern int Machine;
extern int FGL;
extern long FLAG;
extern int SAL;
extern char NAME[];
int px,py;
FILE *ftemp;

```

```
void Menu(void);
```

```

/* ----- PRIMER PANTALLA DE LA EJECUCION DEL
PROGRAMA ----- */

```

```
void presenta(void)
```

```

{ int i;

  init();
  cls(0,0);
  for (i=0;i<25;+ +i)
  {
    centre(i, " ",20);
  }
  centre(5, " ",20);
  centre(6, " ",20);
  centre(7, " ",20);
  centre(8, " ",20);
  centre(9, " ",20);
  centre(10, " ",20);
  centre(11, " ",20);
  centre(12, " ",20);
  centre(13, " ",20);
  centre(14, " ",20);
  centre(15, " ",20);
  centre(17, " MANEJADOR ",120);
  inicializa();

```

```

parametro(0,"(0,codigo,n,3)");
parametro(0,"(1,carnet,n,10)");
parametro(0,"(2,nombre,a,30)");
parametro(0,"(3,direcc,a,40)");
parametro(0,"(4,telefono,n,10)");
parametro(0,"(5,edad,n,3)");
parametro(0,"(6,fecha,a,11)");
parametro(0,"(7,hora,a,9)");
parametro(0,"(8,desc,a,50)");
parametro(0,"(9,nivel,a,15)");
parametro(0,"(10,ttotal,n,3)");
parametro(0,"(11,pregnum,n,3)");
parametro(0,"(12,pregunta,a,59)");
parametro(0,"(13,posibles,a,50)");
parametro(0,"(14,resp,n,3)");
parametro(0,"(15,tiempo,n,3)");
parametro(0,"(16,punteo,n,3)");
parametro(0,"(17,punteo,n,5)");
parametro(0,"(18,fecha,a,11)");
parametro(0,"(19,hora,a,9)");

```

```
} /* end of function present */
```

```
/* ----- FUNCION PRINCIPAL ----- */
```

```
void Menu(void)
```

```

{ int flag = 1;

  presenta();
  sound(Sonido);
  delay(Lapso);
  nosound();

```

```
init();
```

```

cls(0,0);
cuadro(0,0,69,3,"  || =",20);
cuadro(70,0,79,3,"  | =",10);
cuadro(0,22,79,24,"  | =",10);
imprima(2,1,USAC,2);
imprima(2,2,CEDE,2);
imprima(39,1,PROP,10);
imprima(39,2,SERIE,10);

cls(4,0);

do {
    flag=trabaja();
} while (flag!=False);

cls(0,0);
beep();

} /* end of fuction menu */

/* ---- funcion que se encarga de manejar el menu de
ordenes --- */

int trabaja(void)
{
    int i=0;
    int t1;
    int flag=1;
    char LNC[255]="\0";

    cuadro(0,22,79,24,"  | =",10);
    imprima(72,1," MENU ",120);
    imprima(0,4,"Pruebas Análisis Reportes Migrar
Salir ".10);
    imprima(47,23," <— Enter=Introduce",10);
    cambiaH(i,4,10,120);
    gotoxy(i,4);
    imprima(0,5,"Ingreso de pruebas y Evaluación
.5);

    do {

        t1=tecla();
        sound(Click);
        delay(Click);
        nosound();
        if (t1==0 && t1!=-15)
            { cambiaH(i,4,10,10);
              i = 50;
            }

        if (i==40 && t1!=-17)
            { cambiaH(i,4,10,10);
              i = -10;
            }

        if (t1==-17 && i<40)
            { cambiaH(i,4,10,10);
              i+= 10;
            }

        if (t1== -15 && i>= 10)
            { cambiaH(i,4,10,10);
              i = 10;
            }
    }

```

```

cambiaH(i,4,10,120);
gotoxy(i,4);

switch(i)
{
    case 0:
        imprima(0,5,"Ingreso de pruebas y Evaluación
.5);
        break;
    case 10:
        imprima(0,5,"Análisis estadístico de pruebas
evaluadas ".5);
        break;
    case 20:
        imprima(0,5,"Reportes de las pruebas evaluadas
.5);
        break;
    case 30:
        imprima(0,5,"Envío y recepción de datos
.5);
        break;
    case 40:
        imprima(0,5,"Salir de la aplicación
.5);
        break;
} /* end of switch */

} while(t1!= 13);

if (t1==13)
{
    strset(LNC,'\0');

    switch(i)
    {
        case 0:
            linea_car(0,4,' ',10,80);
            linea_car(0,5,' ',10,80);
            do {
                t1=pruebas();
            } while (t1!=27);
            break;
        case 10:
            linea_car(0,4,' ',10,80);
            linea_car(0,5,' ',10,80);
            do {
                t1=analisis();
            } while (t1!=27);
            break;
        case 20:
            linea_car(0,4,' ',10,80);
            linea_car(0,5,' ',10,80);
            cuadro(0,5,79,20,"  || =",50);
            cuadro(0,22,79,24,"  | =",10);

            cls(4,0);
            break;
        case 30:
            linea_car(0,4,' ',10,80);
            linea_car(0,5,' ',10,80);
            cuadro(0,5,79,20,"  || =",50);
            cuadro(0,22,79,24,"  | =",10);
    }
}

```

```

        cls(4,0);
        break;

    case 40:
        linea_car(0,4,' ',7,80);
        linea_car(0,5,' ',7,80);
        cuadro(0,22,79,24,"  | =",10);
        Programa",10);
        Imprima(72,1,"SALIR ",120);
        Imprima(47,23,"<--- Enter=Introduce
Esc=Sale",10);

        do {
            gotoxy(76,23);
            t1=tecla();
        } while (t1!=13 && t1!=27);

        if (t1 == 13)
            { flag=0;}

        break;
    } /* end of switch */
} /* end of if t1 == 13 */
linea_car(0,4,' ',10,80);
linea_car(0,5,' ',10,80);
linea_car(0,6,' ',10,80);

return(flag);

} /* end of fuction trabaja */

/* ----- MANEJA EL MENU DEL PRUEBAS Y SUS
FUNCIONES ----- */

int pruebas()
{
    int i=0,j=0;
    int t1,t2;
    char line[15]="\0",linea[50]="\0";
    char LNC[255]="\0",LNC1[255]="\0";
    struct date fecha;
    struct time hora;
    int preg;
    long POSICION;
    char Nom[30]="\0";

    Imprima(72,1,"TESTS ",120);
    Imprima(0,4, "Evaluar Crear Modificar Eliminar
",10);
    Imprima(47,23,"<--- Enter=Introduce
Esc=Sale",10);
    cambiaH(i,4,10,120);
    gotoxy(i,4);
    Imprima(0,5,"Permite tomar una prueba
",5);

    do {
        t1=tecla();
        sound(Click);
        delay(Click);
        nosound();
        if (i==0 && t1== -15)
            { cambiaH(i,4,10,10);

```

```

        i = 40;}

        if (i==30 && t1== -17)
            { cambiaH(i,4,10,10);
            i = -10;}

        if (t1 == -17 && i<30)
            { cambiaH(i,4,10,10);
            i+=10;}

        if (t1 == -15 && i>= 10)
            { cambiaH(i,4,10,10);
            i= 10;}

        cambiaH(i,4,10,120);
        gotoxy(i,4);

        switch(i)
        {
            case 0:
                Imprima(0,5,"Permite tomar una prueba
",5);
                break;
            case 10:
                Imprima(0,5,"Opción para Crear nuevas pruebas
",5);
                break;
            case 20:
                Imprima(0,5,"Opción para Modificar pruebas
existentes ",5);
                break;
            case 30:
                Imprima(0,5,"Opción para Eliminar pruebas no
necesarias ",5);
                break;
        } /* end of switch */

    } while(t1!= 13 && t1!=27);

    if(t1 == 13)
        { strset(LNC,"\0");
        switch(i)
        {
            case 0:
                linea_car(0,4,' ',10,80);
                linea_car(0,5,' ',10,80);
                Imprima(72,1,"EVALUA",120);
                cuadro(0,4,79,20,"  | || =",50);
                cuadro(0,22,79,24,"  | | =",10);

                Imprima(5,6," CARNET = ",10);
                Imprima(20,6," ",120);
                px=20; py=6;
                strset(line,"\0");
                strcat(LNC,("\0"));
                rotulo(line,&j,10);
                strcat(LNC,line);
                strcat(LNC,("\0"));

                parametro(2,"(usuarios)");
                parametro(4,LNC);
                POSICION=FLAG;
                parametro(9,"\0");
                if (POSICION==0)
                    {

```



```

strcat(Nom, linea);
strcat(LNC, linea);
strcat(LNC, ",");

j=0;
imprima(5, 12, " DESCRIPCION = ", 10);
imprima(20, 12,
", 120);

px=20; py=12;
entrada(linea, &j, 40);
strcat(LNC, linea);
strcat(LNC, ",");

j=0;
imprima(5, 16, " NIVEL = ", 10);
imprima(20, 16,
", 120);
px=20; py=16;
entrada(linea, &j, 15);
strcat(LNC, linea);
strcat(LNC, ",");

j=0;
imprima(40, 16, " TIEMPO TOTAL/min =
", 10);

imprima(60, 16,
", 120);
px=60; py=16;
rotulo(linea, &j, 3);
strcat(LNC, linea);
strcat(LNC, "\n");

imprima(47, 23, "<— Enter=Introduce
Esc = Sale", 10);

gotoxy(46, 23);
t2=tecla();
if (t2 == 13)
{
    parametro(2, "(pruebas)");
    parametro(3, LNC);
    parametro(9, "\0");
    preg=1;
    if (FGL!=0)
    {
        strcat(Nom, linea);
        strcat(LNC1, "(");
        strcat(LNC1, Nom);

strcat(LNC1, ", 11, 12, 13, 14, 15, 16, \200\");
        parametro(1, LNC1);
        if (FGL!=0)
        {
            do {
                cls(4, 0);
                strcpy(linea, "");
                strcat(linea, Nom);
                strcat(linea, "");
                t1=pregunta(line, &preg, linea);
            } while (t1!=27);
            } /* end if FGL */
            else beep();
        } /* end if FGL */
        else beep();
    } /* end of if t2 */

    cls(4, 0);

break;
case 20:
    j=0;
    linea_car(0, 4, ' ', 10, 80);
    linea_car(0, 5, ' ', 10, 80);
    imprima(72, 1, "MODIF ", 120);
    cuadro(0, 4, 79, 20, " ¶ ¶ || =", 50);
    cuadro(0, 22, 79, 24, " ¶ ¶ | =", 10);
    imprima(5, 6, " CODIGO = ", 10);
    imprima(20, 6,
", 120);
    px=20; py=6;
    strset(line, '\0');
    strcat(LNC, "(");
    rotulo(line, &j, 3);
    strcat(LNC, line);
    strcat(LNC, "\n");

    parametro(2, "(pruebas)");
    SAL=2;
    parametro(4, LNC);
    SAL=0;
    parametro(9, "\0");

    cls(4, 0);
break;
case 30:
    j=0;
    linea_car(0, 4, ' ', 10, 80);
    linea_car(0, 5, ' ', 10, 80);
    imprima(72, 1, "ELIMIN", 120);
    cuadro(0, 4, 79, 20, " ¶ ¶ || =", 50);
    cuadro(0, 22, 79, 24, " ¶ ¶ | =", 10);
    imprima(5, 6, " CODIGO = ", 10);
    imprima(20, 6,
", 120);
    px=20; py=6;
    strset(line, '\0');
    strcat(LNC, "(");
    rotulo(line, &j, 3);
    strcat(LNC, line);
    strcat(LNC, "\n");

    parametro(2, "(pruebas)");
    SAL=2;
    parametro(4, LNC);
    SAL=0;
    parametro(7, LNC);
    parametro(9, "\0");

    cls(4, 0);
break;

} /* end of switch */
} /* end of if t1 == 13 */

return(t1);
} /* end of function pruebas */

/* ----- FUNCION QUE SE ENCARGA DEL INGRESO
DE PREGUNTAS ----- */

int pregunta(codprue, idpreg, Nomarch)
char codprue[];
int *idpreg;

```

```

char Nomarch[];

{
  int t1,t2,i= 0;
  char linea[150]="\0";
  char LNC[255]="\0";
  int idresp=1,cont=0;

  imprima(72,1,"PREGS ",120);
  imprima(0,4,"Ingresar ",10);
  imprima(47,23,"<— Enter=Introduce Esc=Sale",10);
  cambiaH(i,4,10,120);
  gotoxy(i,4);
  imprima(0.5,"Permite ingresar nuevas preguntas
",5);

  do {

    t1=tecla();
    sound(Click);
    delay(Click);
    nosound();

    i=0;

    switch(i)
    {
      case 0:
        imprima(0.5,"Permite ingresar nuevas preguntas
",5);
        break;
      } /* end of switch */

    } while(t1!= 13 && t1!=27);

  if(t1 == 13)
  {
    strset(LNC,\0');
    switch(i)
    {
      case 0:
        linea_car(0,4,' ',10,80);
        linea_car(0,5,' ',10,80);
        imprima(72,1,"INGRES",120);
        cuadro(0,4,79,20,"  || =",50);
        cuadro(0,22,79,24,"  | =",10);

        imprima(5,6," CODIGO = ",10);
        imprima(20,6,codprue,120);

        strset(linea,\0');
        itoa(*idpreg,linea,10);
        imprima(5,8," PREGUNTA # = ",10);
        imprima(20,8,linea,120);

        strcat(LNC,("\n"));
        strcat(LNC,linea);
        strcat(LNC,".");
        ++(*idpreg);

        j=0;
        imprima(5,10," PREGUNTA = ",10);
        imprima(5,12."
",120);
        px=5; py=12;
        entrada(linea,&j,60);

        strcat(LNC,linea);
        strcat(LNC,".");
        imprima(5,15," POSIBLES RESP. = ",10);

        do {
          imprima(47,23,"
",10);

          strset(linea,\0');
          itoa(idresp,linea,10);
          strcat(linea,".");
          imprima(24,15,linea,10);
          imprima(28,15," ",120);
          j=0;
          px=28; py=15;
          entrada(linea,&j,10);
          strcat(LNC,linea);
          strcat(LNC,".");
          imprima(47,23,"<—
Enter=Continua Esc=Sale",10);

          do {
            gotoxy(46,23);
            t2=tecla();
            } while (t2!= 13 && t2!=27);
            ++idresp;
            cont+=j;
            } while (t2!=27 && cont<50);
          strcat(LNC,".");

          j=0;
          imprima(47,23," ",10);
          imprima(5,18," RESPUESTA = ",10);
          imprima(20,18," ",120);
          px=20; py=18;
          rotulo(linea,&j,3);
          strcat(LNC,linea);
          strcat(LNC,".");

          j=0;
          imprima(30,18," TIEMPO/seg = ",10);
          imprima(45,18," ",120);
          px=45; py=18;
          rotulo(linea,&j,3);
          strcat(LNC,linea);
          strcat(LNC,".");

          j=0;
          imprima(55,18," PUNTUACION = ",10);
          imprima(70,18," ",120);
          px=70; py=18;
          rotulo(linea,&j,3);
          strcat(LNC,linea);

          strcat(LNC,("\n"));
          parametro(2,Nomarch);
          parametro(3,LNC);
          parametro(9,\0');

          cls(4,0);
          break;
        } /* end of switch */
      } /* end of if t1 = 13 */

    return(t1);
  }
}

```

```

} /* end of function preguntas */

/* --- SE ENCARGA DE DESPLEGAR LAS PREGUNTAS
PARA EVALUACION --- */

int evalua()

{
    int i=0,j=0;
    char line[15]="\0";
    char LNC[255]="\0";
    int t1;
    long temp=0;

    imprima(72,1,"EVALUA",120);
    imprima(0,4,"Selecciona Listado ",10);
    imprima(47,23,"<--- Enter=Introduce
Esc=Sale",10);
    cambiaH(i,4,10,120);
    gotoxy(i,4);
    imprima(0,5,"Permite seleccionar una prueba y
tomarla ",5);

    do {

        t1=tecla();
        sound(Click);
        delay(Click);
        nosound();
        if (i==0 && t1===-15)
            { cambiaH(i,4,10,10);
              i = 20;
            }

        if (i==10 && t1===-17)
            { cambiaH(i,4,10,10);
              i = -10;
            }

        if (t1===-17 && i<10)
            { cambiaH(i,4,10,10);
              i=i+10;}

        if (t1===-15 && i>=10)
            { cambiaH(i,4,10,10);
              i=i-10;}

        cambiaH(i,4,10,120);
        gotoxy(i,4);

        switch(i)
        {
            case 0:
                imprima(0,5,"Permite seleccionar una prueba y
tomarla ",5);
                break;
            case 10:
                imprima(0,5,"Listado de pruebas almacenadas
",5);
                break;
        } /* end of switch */

    } while(t1!=13 && t1!=27);

    if(t1 == 13)

```

```

{ strset(LNC,'\0');
  switch(i)
  {
      case 0: j=0;
              linea_car(0,4,' ',10,80);
              linea_car(0,5,' ',10,80);
              imprima(72,1,"SELEC ",120);
              cuadro(0,4,79,20," || ==",50);
              cuadro(0,22,79,24," || ==",10);
              imprima(5,6," CODIGO = ",10);
              imprima(20,6," ",120);
              px=20; py=6;
              strset(line,'\0');
              strcat(LNC,("\n"));
              rotulo(line,&j,3);
              strcat(LNC,line);
              strcat(LNC,("\n"));

              parametro(2,"(pruebas)");
              SAL=1;
              parametro(4,LNC);
              SAL=0;
              temp=FLAG;
              parametro(9,"\0");
              if (temp!=0)
                  { parametro(2,NAME);
                    parametro(10,"\0");
                    parametro(9,"\0");
                  }
              else beep();

              cls(4,0);
              break;
          case 10 :
              imprima(72,1,"LISTA ",120);
              parametro(2,"(pruebas)");
              parametro(5,"\0");
              parametro(9,"\0");

              cls(4,0);
              break;
          } /* end of switch */
      } /* end of if t1 == 13 */

    return(t1);

} /* end of function evaluar */

/*----- ESTA ENCARGA DEL MANEJO DE FUNCIONES
PARA ANALISIS ----- */

int analisis()
{
    int i=0;
    int t1;
    char LNC[255]="\0";

    imprima(72,1," ",120);
    imprima(0,4,"Corelac Varianza ",10);
    imprima(47,23,"<--- Enter=Introduce
Esc=Sale",10);
    cambiaH(i,4,10,120);
    gotoxy(i,4);
    imprima(0,5,"Se realiza una corelllll
",5);

```

```

do{
    t1=tecla();
    sound(Click);
    delay(Click);
    nosound();
    if (i==0 && t1==15)
        { cambiaH(i,4,10,10);
          i = 20;
        }

    if (i==10 && t1==17)
        { cambiaH(i,4,10,10);
          i = -10;
        }

    if (t1==17 && i<10)
        { cambiaH(i,4,10,10);
          i=i+10;}

    if (t1==15 && i>=10)
        { cambiaH(i,4,10,10);
          i=i-10;}

    cambiaH(i,4,10,120);
    gotoxy(i,4);

    switch(i)
    {
        case 0:
            imprima(0,5,"Se realiza una cores!!!!!!!
",5);
            break;
        case 10:
            imprima(0,5,"Varianza
",5);
            break;
    } /* end of switch */

} while(tt!= 13 && tt!=27);

if(tt==13)
{ strset(LNC,'\0');
  switch(i)
  {
      case 0:
          linea_car(0,4,' ',10,80);
          linea_car(0,5,' ',10,80);
          imprima(72,1,"CORREL",120);
          cls(4,0);
          cuadro(0,5,79,20,"  || =",50);
          cuadro(0,22,79,24,"  | =",10);

          t1 = 13;
          cls(4,0);
          break;

      case 10:
          linea_car(0,4,' ',10,80);
          linea_car(0,5,' ',10,80);
          imprima(72,1,"VARIAN",120);
          cls(4,0);

          t1 = 13;

```

```

        cls(4,0);
        break;
    } /* end of switch */
} /* end of if t1==13 */

return(t1);

} /* end of function analisis */

/* ----- funcion que verifica la entrada de un Rotulo en la
Hoja ----- */

int rotulo(numero,j,l)
char numero[];
int *,j,l;
{ int tt;

  (*j)=0;
  gotoxy(px,py);
  tt=tecla();
  sound(Touch);
  delay(Touch);
  nosound();
  do {
      if ((*j)<1 && tt!=13 && tt!=27 && tt!=8)
          {
              if (tt==48 && tt<=57)
                  {
                      gotoxy(px,py);
                      printf("%c",tt);
                      numero[*j]=call(px,py);
                      ++px;
                      ++(*j);
                      numero[*j]='\0';
                  } /* end of if */
              }
          if (tt==8)
              {
                  if ((*j)>0)
                      {
                          px-=1;
                          (*j)--;
                      }
                  gotoxy(px,py);
                  printf("%c",' ');
                  gotoxy(px,py);
                  numero[*j]='\0';
              }
          tt=tecla();
          sound(Touch);
          delay(Touch);
          nosound();
      } while(tt!=13 || (*j)<1 || tt!=27);

  return(tt);

} /* end of function rotulo */

/* ----- FUNCION QUE LEE UN LETRERO DESDE LA
CONSOLA ----- */

int letrero(linea,i)
char linea[];
int *i;

```

```

{ int tt;

gotoxy(px,py);
tt=tecla();
sound(Touch);
delay(Touch);
nosound();
do{
  if ((*i)<30 && tt!=13 && tt!=27 && tt!=8)
  {
    if ((tt>=65 && tt<=90) || (tt>=97 && tt<=122) ||
        (tt>=48 && tt<=57) || tt==95)
    {
      gotoxy(px,py);
      printf("%c",tt);
      linea[*i]=call(px,py);
      ++px;
      ++(*i);
      linea[*i]='\0';
    } /* end of if */
  }
  if (tt==8)
  { if ((*i)>0)
    { px-=1;
      (*i)-=1;
    }
    gotoxy(px,py);
    printf("%c",' ');
    gotoxy(px,py);
    linea[*i]='\0';
  } /* end of if */
  if (tt==27)
  { break;}
  tt=tecla();
  sound(Touch);
  delay(Touch);
  nosound();

} while(tt!=13 || (*i)<1);

return(tt);

} /* end of function letrero */

```

```

/* ----- FUNCION QUE LEE UN LETRERO DESDE LA
CONSOLA ----- */

```

```

int entrada (linea,i,j)
char linea[];
int *i,j;

```

```

{ int tt;

gotoxy(px,py);
tt=tecla();
sound(Touch);
delay(Touch);
nosound();

do{
  if ((*i)<j && tt!=13 && tt!=27 && tt!=8)
  {
    if (tt>=33 || tt<=253)

```

```

{
  gotoxy(px,py);
  printf("%c",tt);
  linea[*i]=call(px,py);
  ++px;
  ++(*i);
  linea[*i]='\0';
} /* end of if */
}
if (tt==8)
{ if ((*i)>0)
  { px-=1;
    (*i)-=1;}
  gotoxy(px,py);
  printf("%c",' ');
  gotoxy(px,py);
  linea[*i]='\0';
} /* end of if */
if (tt==27)
{
  strcpy(linea,"\0");
  break;
}
tt=tecla();
sound(Touch);
delay(Touch);
nosound();

} while(tt!=13 || (*i)<1 || tt!=27);

return(tt);

} /* end of function entrada */

```

VIII. Conclusiones

1. El desarrollo de un manejador de pruebas del C.I. de este tipo contribuye a hacer más eficiente el trabajo de los psicometristas y elimina información no actualizada.
2. Contribuye como herramienta de trabajo al evaluar a un individuo y obtener resultados inmediatos.
3. Las ventajas de una aplicación de este tipo son muchas, pero la principal es el acercamiento de la psicometría con la tecnología disponible.
4. La característica principal del manejador de pruebas es poder acumular los resultados obtenidos y utilizarlos en investigaciones psicométricas.
5. Se concluye que el presente trabajo contribuye con el área de la psicometría en nuestro país, sin pretender ingresar a un campo de acción de los profesionales y expertos de la misma.

IX. Recomendaciones

1. Se recomienda que las pruebas sean manejadas por personas capacitadas para su uso y aplicación.
2. Es necesario tener el equipo adecuado para la aplicación de las pruebas, el cual consiste en una computadora con sus componentes básicos.
3. Es conveniente utilizar los resultados de las pruebas obtenidos por medio del manejador, con fines de investigación.
4. Se debe mantener discreción en los resultados de las pruebas.
5. La implementación debe ser utilizada, verdaderamente, por profesionales del área de la psicología y específicamente del campo de la psicometría.
6. Se recomienda que este trabajo sea una herramienta, que sea considerada como punto de partida de otras más.

X. Bibliografía

1. BARRIOS PEÑA, Jaime. Prueba proyectiva nacional.
(tesis: Facultad de Psicología, Universidad de San Carlos) Guatemala, 1,964. 40 pp.
2. CERDÁ, Enrique. Psicometría General.
3era Edición. Barcelona: Edit. Herder, 1,984. 120 pp.
3. GAGE, N. La inteligencia. Guatemala : Edit. Piedra Santa, 1,990. p. 57.
4. GARRY, Ralph. The Psychology of learnig.
Washinton: Center for Applied Research In Educación, 1,963. p.38.
5. HURTADO De MENDOZA, Maria de los Angeles. Prueba de rendimiento académico y objetivos de la instrucción. México: Edit. DIANA, 1,980. 89 pp.
6. MEHRENS, William A. Medición y evaluación en la educación. México: Edit. CECSA, 1,982. 100 pp.
7. PASEH VALDEZ, Grete María. Procedimiento computarizado para analizar los factores que inciden sobre el éxito académico.(tesis: Facultad de Ingeniería, Universidad Francisco Marróquin), Guatemala, 1,984. 67 pp.
8. SOLIER P. El problema de la memoria. Ensayo de psico-mecánica. Madrid: Librería de Fernando Fé, 1,972. 140 pp.
9. THURSTONE, L. La medición de la inteligencia, la aptitud y el interés. Buenos Aires: Edit. Paidos, 1,967. 230 pp.
10. WECHSLER, David. La medición de la inteligencia del adulto. La Habana: Edit. Cultural, 1,955. 110 pp.
11. ZUBIRI, Xavier. Inteligencia y logos. Madrid: Edit. Alianza, 1,982. p. 35.



XI. Glosario

A

Abaco

Dispositivo para realizar cálculos aritméticos mediante cuentas deslizantes a lo largo de hilos o alambres.

Algoritmo

Pertenece al uso de algoritmos para la solución de problemas.

Alto, Nivel

Pertenece a un lenguaje de programación en que cada instrucción o declaración fuente corresponde a varias instrucciones de lenguaje máquina; generalmente permite al usuario codificar en una notación que se relaciona con el problemas más que con el computador.

Almacenamiento

1. Pertenece a cualquier dispositivo en que se pueden entrar datos, en que pueden mantenerse y del que se pueden recuperar en tiempo posterior.
2. Cualquier dispositivo que pueda utilizarse para almacenar datos.

Análisis de sistemas

Examen de una empresa, procedimiento, método o actividad para determinar lo que deba lograrse y de qué manera. En general, el uso se refiere al análisis de un problema para determinar la solución computarizada ideal.

Aplicación

Problema o sistema específico al que se aplican técnicas de procesamiento de datos. A menudo, se refiere a las aplicaciones como computacionales (que requieren un elevado nivel de computación) o de procesamiento de datos (que requieren un elevado nivel de datos).

Archivo

Colección organizada de registros relacionados. Por ejemplo, una línea de una factura puede formar un artículo; una factura completa puede constituir un registro; un conjunto completo de esos registros puede formar un archivo; la colección de archivos de control financiero puede constituir una biblioteca y todas las bibliotecas utilizadas por una organización forman su banco de datos.

Automática, Interrupción

Interrupción de un programa generada por un dispositivo de mecánica o una rutina ejecutiva, como resultado de que haya ocurrido un evento en forma independiente del programa que se está interrumpiendo.

Automática, Prueba

Facilidad, de programación o mecánica, para verificar automáticamente la exactitud de los procesos de máquina.

Automatización

1. Teoría de hacer automáticos los procesos.
2. La implementación automática de un proceso.
3. El control de un proceso de producción por medios automáticos.

B

Bajo Nivel

Pertenece a un lenguaje de programación en que cada enunciado en lenguaje fuente tiene un equivalente en lenguaje máquina correspondiente.

X

Bandera

1. Carácter que se utiliza par indicar la ocurrencia de una condición o evento.
2. Artículo que se utiliza como indicador.

Bit

Uno de los dígitos (0 y 1) que se usan en la notación binaria; abreviatura en inglés de dígito binario. El término se utiliza en una diversidad de contextos, cada uno de los cuales indica la representación real de un dígito binario.

Blanco

1. Miembro de un conjunto de caracteres que representa un espacio impreso.
2. Que no contiene información.

Borrar

1. Quitar o eliminar.
2. En uso general, cualquier operación para quitar un registro o grupo de ellos de un archivo.
3. Comúnmente, quitar un programa de memoria cuando ha terminado de ejecutarse.
4. La parte de un conjunto de caracteres que se utiliza para indicar que ciertos datos anteriores están equivocados.
5. Sustituir los datos en memoria con ceros o blancos.
6. Quitar, reescribir o suprimir información de un soporte de almacenaje.

Búsqueda

Localizar una entidad de dato y cargarla a memoria.

BYTE

Conjunto de bits que representan la unidad direccionable más pequeña dentro de la memoria.

C**Carácter**

Uno de un conjunto de símbolos que representa los numerales, el alfabeto, símbolos de puntuación, etc. que se utilizan en un sistema de procesamiento de datos.

Cargar

1. Entrar datos a memoria.
2. Posicionar un soporte de datos en una unidad de entrada.

Central, Procesador

Unidad de un sistema de computador que contiene las unidades aritméticas, lógica y de control y que ordena y coordina la operación del sistema y sus dispositivos periféricos. El procesador central guarda toda las instrucciones que se deben ejecutar y los datos que van a procesarse.

Circuito

1. Enlace de comunicaciones entre dos o más puntos.
2. Sistema de conductores y componentes a través de los cuales fluye una corriente eléctrica.

Comando

1. Señal utilizada para iniciar, detener o continuar alguna forma de instrucción de computador o periférico.

2. Parte de una instrucción que especifica la operación que se va a hacer.
3. En uso general, igual que instrucción.

Computador(a)

Máquina que puede aceptar datos en una forma predefinida, procesarlos, de acuerdo con un conjunto de forma y suministrar los resultados del procesamiento en una forma prescrita. Los resultados pueden estar en forma de información o de señales para controlar otra máquina de proceso.

Corrida

El funcionamiento generalmente, continuo, de un programa o rutina.

Cursor

Punto móvil de luz en la pantalla de rayos catódicos, que indica donde se entrará el siguiente carácter.

D

Datos

Término general que se utiliza para denotar cualesquiera o todos los hechos, letras símbolos y números que se refieren s o describen un objeto, idea, situación, condición u otro factor. En uso general, son elementos de información que se pueden procesar en un computador.

Datos, Base de

Archivo de datos estructurado de tal manera, que se pueden expresar todas las relaciones lógicas entre los registros, y que es independiente de cualquier aplicación específica. De esa manera los programas que accesan la base datos no imponen ellos mismos restricciones a la organización de los datos que contiene.

E

Editar

Arreglar datos en el formato que se requiere para el procesamiento subsecuente.

Ejecutar

Realizar las operaciones especificadas por una rutina o instrucción determinada.

Error

Acción humana que produce un resultado distinto al propuesto, discrepancia entre una cantidad observada, computada o medida y el valor o condición verdadero o teórico.

G

Generar

Producir un programa seleccionando distintas opciones, de una diversidad de opciones disponibles, utilizando parámetros específicos.

I

Implementación

Proceso de diseñar, preparar, probar e instalar un sistema.

Información

Significado humano que se le da a los datos como resultado de su análisis de acuerdo con convenciones conocidas.

Impresora

Dispositivo que imprime o expresa caracteres codificados como una copia sólida.

Informe

Todo análisis impreso.

Instrucción

Parte de un programa de computador que le indica qué función debe realizar en esa etapa. Una instrucción consiste en una operación y uno o más operandos.

Interactivo(a)

Pertenece a cualquier sistema de computador en que un usuario de terminal en línea recibe respuestas inmediatas a cualquier entrada de datos. En especial, el sistema de asistencia a usuarios mediante diagnósticos de inmediato en caso de errores.

J**Jerarquía**

Estructura de datos que consiste en conjuntos y subconjuntos. Cada subconjunto tiene un rango inferior a los datos de su conjunto.

L**Lenguaje**

Conjunto de representaciones, convenciones y reglas que se utilizan para transmitir información.

Lógica

La ciencia del razonamiento; el estudio de la forma de argumento sin consideración del significado específico de los términos del argumento.

M**Máquina**

En uso general, igual que Computador.

Matriz

Arreglo rectangular de cantidades que tiene dos o más dimensiones y de uso general igual que arreglo.

Memoria

Cualquier dispositivo en que se puede entrar, retener y posteriormente recuperar los datos, en uso general, la memoria principal de un computador y que excluye todo almacenaje de soporte.

Mensaje

Combinación de caracteres y símbolos diseñados para transmitir una unidad de información.

O

Operación

Una acción definida que involucra la producción de un resultado a partir de uno o más operandos de acuerdo con las reglas de una instrucción específica.

Operador(a)

Persona que opera una máquina.

Operativo, Sistema

Programática asociados con la ejecución de programas y la coordinación de las actividades de un sistema de computador. Puede incluir procedimientos como administración de almacenaje y de datos, traducción de lenguajes y servicios relativos.

P**Pantalla**

Superficie de un tubo de rayos catódicos.

Paquete

Programa o conjunto de programas generalizados, por lo común de un propietario, diseñados para resolver una aplicación común.

Procesamiento

La secuencia de pasos o el curso de acción que se toma para resolver un problema dado.

Procesador

Cualquier dispositivo que pueda realizar operaciones con datos.

Proceso

Término general para cualquier secuencia de operaciones diseñadas para producir un resultado específico.

Programa

Secuencia de instrucciones o enunciados en forma aceptable a un computador preparada para lograr un resultado específico.

Programación

Todos los procesos que participan en el diseño, escritura y prueba de programas de computador.

Programación, Lenguaje de

Lenguaje que se utiliza en la preparación de programas de computador.

Prueba, corrida de

Ejecución de un programa o programas, utilizando datos de prueba para la verificación de la función correcta del mismo.

Prueba, datos de

Datos preparados específicamente para probar todas las facilidades de un programa o sistema.

S**Salida**

Pertenece a cualquier dispositivo, proceso, programa o canal asociado con un proceso de salida.

XIII. Índice Invertido

Sistema de Computación	1
Computadora electrónica digital	2
Los Sumerios, Caldeos, Babilonios y Egipcios	2
Blaise Pascal	2
Hollerith	3
APPLE	3
Computadora	4
Dispositivos electrónicos	4
Entrada	4
Memoria	4
Lógica / aritmética	4
Control	4
Salida	4
Computadora personal	4
PC	4
ALU por Arithmetic Logic Unit	4
Unidad central de procesamiento	4
RAM	4
ROM	4
Microcomputadora	5
Supermicro o micromini	5
Minicomputadora	5
Supermini	5
Mainframe	5
Supercomputadora	5
Bits y bytes	5
Dígito binario	5
Basic, Pascal, C	5
Software	6
Sistemas operativos	6
Periféricos	6
Pantalla	7
Impresora	7
Disk Drive	7
Teclado	7
Diskette	7
Disco Duro	7
Software	7
Ciclo de vida	8
Firmware	8
Programa	8
Lenguajes de Programación	8
Programas Paquete	9
Psicometría	11
Tests mentales	11
Métodos psicométricos	11
Umbral absoluto	11
Error medio	11
Test mental	11
Tests mentales	12
Métodos psicométricos	12
Pruebas psicométricas	12
Análisis de varianza	12
Análisis factorial	12
Prueba	12
Test	13
Test pedagógico	13
Estandarización	14
Binet y Simon	15
Test mental	15

Prueba psicométrica	15
Coefficiente de validez	21
Validez aparente	22
Validez del contenido	22
Validez predictiva	22
Validez concurrente	22
Hipótesis de trabajo	22
Validación Factorial	22
Coefficiente de Inteligencia	23
Puntuaciones típicas	23
Ansiedad	25
Falseo de resultados	25
Lenguaje de tercera generación 3GL	27
Cédula	27