



**UNIVERSIDAD NACIONAL
“SANTIAGO ANTUNEZ DE MAYOLO”**

ESCUELA DE POSTGRADO

**SISTEMA OMR PARA CALIFICACIÓN DE TARJETAS DE
RESPUESTAS DE EXÁMENES, MEDIANTE EL USO DE UN
ESCÁNER CONVENCIONAL Y UNA COMPUTADORA
PERSONAL**

Tesis para optar el grado de maestro
en Ciencias e Ingeniería
Mención en Computación e Informática

LUIS AUGUSTO TORRES ALEGRE

Asesor: **Dr. JESÚS EDILBERTO ESPINOLA GONZÁLES**

Huaraz – Perú
2010

Nº. Registro: _ _ _ _ _



UNIVERSIDAD NACIONAL
"SANTIAGO ANTÚNEZ DE MAYOLO"
ESCUELA DE POSTGRADO

ACTA DE SUSTENTACION DE TESIS

Los miembros del Jurado de Sustentación de Tesis, que suscriben, reunidos en acto público en el local de la Escuela de Postgrado de la Universidad Nacional "Santiago Antúnez de Mayolo" para calificar la Tesis presentada por el:

Bachiller : **LUIS AUGUSTO TORRES ALEGRE**

Título : Sistema de reconocimiento de Marcas Ópticas (OMR) para calificación de Tarjetas de Respuestas de Exámenes, mediante el uso de un escáner convencional y una computadora personal.

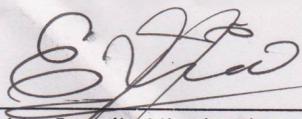


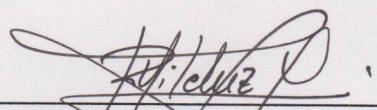
Después de haber escuchado la sustentación y las respuestas a las preguntas y observaciones finales, lo declaramos:

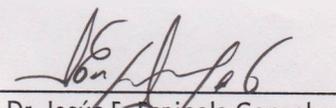
APROBADO CON MENCIÓN, con el calificativo de Dieciocho (18)

De conformidad al Reglamento General a la Escuela de Postgrado y al Reglamento de Normas y Procedimientos para optar los Grados Académicos de Maestro y Doctor, queda en condición de ser aprobado por el Consejo de la Escuela de Postgrado y recibir el Grado Académico de Maestro en Ciencias e Ingeniería con Mención en Computación e Informática a otorgarse por el Honorable Consejo Universitario de la UNASAM.

Huaraz, 22 de ABRIL del 2010.


Mg. Esmelin Niquín Alayo
PRESIDENTE


Mg. Rosa Vilchez Vásquez
SECRETARIO


Dr. Jesús E. Espinola Gonzales
VOCAL

MIEMBROS DEL JURADO

Magíster

Presidente

Magíster

Secretario

Doctor Jesús Edilberto Espinola Gonzáles

Vocal

ASESOR

Doctor Jesús Edilberto Espinola Gonzáles

A la memoria de mi padre Luis Alejandro

A mi querida madre María Alina

A mi amada esposa Norma

A mis queridos hijos Luis y Rafael

A mis hermanos Hugo, Doris y Roger.

ÍNDICE

	Página
Resumen	vii
Abstract	viii
I. INTRODUCCIÓN	1 – 5
Objetivo general	5
Objetivos específicos	5
II. MARCO TEÓRICO	6 - 34
2.1 Antecedentes	6
2.2 Base teórica	6
Píxel	6
Imagen digital RGB	8
Formato de los archivos de imagen de mapa de bits	10
Sistema de coordenadas de la imagen	30
Traslación de los ejes de coordenadas	31
Rotación de los ejes de coordenadas	32
Escalamiento de los ejes de coordenadas	33
III. MATERIALES Y MÉTODOS	35 - 63
3.1 Tipo y diseño de investigación	35
3.2 Instrumentos de recolección de la información	36
3.3 Plan de procesamiento y análisis de la información	36
3.4 Organización del formulario de respuestas	36

3.5	Escaneo del formulario de respuestas	41
3.6	Características básicas del software	41
3.7	Descripción del proceso de examen usando el sistema Universal OMR	42
3.8	Descripción del procesamiento de las imágenes mediante el sistema Universal OMR	43
3.9	Descripción de los modos de presentación de los resultados	48
3.10	Herramientas computacionales	49
3.11	Manual de uso del sistema Universal OMR	49
IV.	RESULTADOS	64 – 71
V.	DISCUSIÓN	72
VI.	CONCLUSIONES	74
VII.	RECOMENDACIONES	75
VIII.	REFERENCIAS BIBLIOGRÁFICAS	77
	ANEXO	79 - 92

RESUMEN

La automatización de la calificación de exámenes de admisión en los centros superiores de estudios del país, está restringida a algunas universidades; que tienen la capacidad económica para adquirir los equipos OMR necesarios para tal fin. Sin embargo, existen miles de institutos superiores que no pueden acceder a estos equipos debido a su alto costo, por lo que el presente trabajo tiene como finalidad desarrollar un sistema OMR para cubrir este vacío.

El objetivo es desarrollar un sistema OMR utilizando un escáner de mesa y una computadora personal, para la calificación de tarjetas de respuestas de exámenes, equipos que en la actualidad, están al alcance de cualquier institución educativa.

El software desarrollado presenta una solución para la calificación de formularios de respuestas de exámenes, mediante la técnica de reconocimiento de marcas ópticas (OMR), contenidas en imágenes de formularios de respuestas almacenadas en archivos digitales con formato de imágenes de mapa de bits, obtenidas mediante un proceso de escaneo.

Palabras clave: OMR, reconocimiento de marcas ópticas, formatos de imágenes digitales, mapa de bits, escáner.

ABSTRACT

The automation of the qualification examination for admission to higher studies in the country, is restricted to some universities that have the economic capacity to purchase the OMR equipment needed for this purpose. However, there are thousands of colleges that can not access these devices due to its high cost, so this paper aims to develop an OMR system to cover this gap.

The aim is to develop an OMR system using a table scanner and a personal computer, for the qualification of cards of exams responses, teams than at the present time, they are within reach of any educational institution.

The developed software provides a solution for the rating of examination answers forms using the technique of optical marks recognition (OMR), in response to images stored on digital files formatted as bitmap images, obtained by a scanning process.

Key words: OMR, optical mark recognition, formats of digital images, map of bits, scanner.

I. INTRODUCCIÓN

Las instituciones educativas superiores públicas y privadas, tales como institutos superiores y universidades; efectúan uno o dos exámenes de admisión por año, con la finalidad de seleccionar a los postulantes que ingresarán a la institución en calidad de alumnos.

El uso de sistemas OMR para la calificación de respuestas de exámenes mediante una tarjeta que permite introducir marcas ópticas como respuesta a cada pregunta, está generalizado en las grandes instituciones educativas, utilizándose para ello sistemas OMR diseñados específicamente para tal fin. Éstos son sistemas especializados y tienen un alto costo.

En la actualidad, han empezado a ser comercializados escáneres OMR de sobremesa que pueden procesar formularios de respuestas impresas en papel A4. Estos escáneres están diseñados especialmente para esta función, proporcionando además el software correspondiente para el procesamiento. El costo de estos equipos es menor al de los sistemas OMR, pero aún sigue siendo alto.

Debido a que las instituciones educativas superiores públicas y algunas universidades no cuentan con recursos económicos suficientes para la adquisición de un sistema OMR comercial para la calificación de tarjetas o formularios de respuestas de exámenes, cada institución diseña los formularios de respuestas para que los alumnos seleccionen las respuestas a las preguntas de los exámenes, mediante la realización de marcas ópticas con un lápiz en estos formularios, que son comúnmente impresos en papel A4.

El procesamiento de estos formularios se realiza en forma manual, lo que demanda un gran trabajo y un lapso de tiempo prolongado, y está sujeto a errores que pueden cometer las personas que se encargan de la calificación de los formularios de respuestas.

El costo de los sistemas de reconocimiento de marcas ópticas es elevado y no está al alcance de la mayoría de instituciones educativas peruanas, por lo que en el presente trabajo se desarrolla un sistema OMR para la calificación de formularios de respuestas de exámenes, utilizando medios convencionales como un escáner de mesa y una computadora personal, equipos al alcance de cualquier institución educativa.

Con el objetivo de automatizar la calificación de los exámenes de admisión, y en general de cualquier examen que emplee la misma metodología de calificación; se ha diseñado un formulario de respuestas estándar, que permite realizar marcas ópticas como respuestas de hasta 100 preguntas.

El formulario de respuestas está diseñado para exámenes de tipo objetivo, en el cual el estudiante puede seleccionar una opción de cinco respuestas posibles por pregunta, sombreando mediante un lápiz uno de los casilleros rectangulares disponibles por pregunta en el formulario. Este formulario se puede imprimir en una hoja de papel tamaño A4.

Para el procesamiento de estos formularios, se ha desarrollado un sistema OMR, denominado “Universal OMR”; para la calificación de los formularios de respuestas, el cual explora los píxeles de la imagen digital del formulario,

determinando la posición de cada marca, en base a lo cual obtiene la calificación alcanzada por el postulante o alumno.

Con esto, se pretende minimizar los costos de la automatización de la calificación de las respuestas en los exámenes de admisión u otro tipo de examen similar, ya que el software desarrollado será distribuido en forma gratuita, a las instituciones educativas nacionales que lo soliciten.

Para procesar los formularios de respuestas, éstos deberán ser primeramente escaneados, y los archivos digitales de imagen obtenidos deben ser almacenados en una carpeta, desde donde serán procesados. Los archivos digitales de imagen deben ser de mapa de bits, y pueden tener extensión BMP, JPG, GIF, etc.

El software desarrollado, examinará cada imagen de formulario en busca de las marcas ópticas realizadas por el estudiante, determinando el código del alumno y las respuestas correctas, calculando el puntaje obtenido.

A continuación, se realiza una breve descripción del contenido del resto de los capítulos del presente documento:

En el **capítulo II**: Marco teórico, se mencionan los antecedentes que existen sobre temas similares al presente estudio. A continuación, se desarrolla el marco teórico, con la definición de los conceptos y los conocimientos que se utilizaron para el desarrollo del trabajo. Se aborda la definición de píxel, el estudio de los formatos de los archivos de imágenes digitales, la organización del sistema de coordenadas que se utiliza para mostrar una imagen en la pantalla y la rotación de los ejes de la pantalla en función de la pendiente de una recta de referencia.

En el **capítulo III**: Materiales y métodos, se muestra el diseño del formulario de respuestas, en el cual contiene 4 bloques de 25 filas, correspondiendo cada fila a una pregunta. Cada fila está compuesta a su vez de 5 casilleros como alternativas de respuesta. Asimismo, se dispuso de 4 columnas de casilleros con alternativas para los números del 0 al 9, para el registro del código del postulante o alumno.

A continuación, se describe las acciones que deberán realizarse para llevar a cabo un examen utilizando el sistema desarrollado. Luego, se describe en forma genérica las acciones que realiza el sistema OMR para el procesamiento de las imágenes de los formularios y la obtención de los resultados.

Luego, se describen los modos de presentación de los resultados, la exportación de éstos a una hoja de cálculo, y la impresión de los mismos.

Finalmente, se describen las tablas que componen la base de datos que utiliza el sistema, y la definición y función de los campos de datos.

En el **capítulo IV**: Resultados y discusión, se presentan los resultados obtenidos en las pruebas realizadas al software desarrollado, utilizando diversos archivos digitales de imagen. Asimismo, se realiza una discusión sobre las dificultades encontradas durante el desarrollo del trabajo y las soluciones planteadas.

En el **capítulo V**: Recomendaciones, se realizan algunas recomendaciones en relación al uso del sistema desarrollado y sobre las mejoras que podrían ser introducidas en trabajos futuros.

En el **capítulo VI**: Referencia bibliográfica, se listan los libros que se consultaron así como las páginas Web de las que se obtuvo información.

En el **capítulo VII**: Anexos, se muestra el código que se escribió para el software desarrollado.

OBJETIVOS:

Objetivo General:

Desarrollar un sistema OMR utilizando un escáner de mesa y una computadora personal.

Objetivos específicos:

- a) Diseñar el prototipo de tarjeta de respuestas.
- b) Desarrollar el algoritmo para el procesamiento de las imágenes de los formularios de respuestas obtenidas mediante el escáner.
- c) Desarrollar el software, la base de datos y la interfase gráfica adecuada para el sistema.

II. MARCO TEÓRICO

2.1 Antecedentes

El reconocimiento de marcas ópticas (OMR), es una tecnología que permite leer e interpretar ciertas clases particulares de marcas ubicadas en determinadas posiciones dentro de un papel impreso. La tecnología OMR ha sido usada desde los años 70 para varios propósitos, como la calificación de tarjetas de respuestas de exámenes, sistemas de votación y lectura de código de barras. Para la calificación de tarjetas de respuestas de exámenes, se utilizan tarjetas especiales que contienen las marcas y son leídas por escáneres diseñados específicamente para cada tipo de tarjeta.

No se han encontrado publicaciones de estudios anteriores sobre el tema del reconocimiento de marcas ópticas, aplicados a la calificación de tarjetas o formularios de respuestas de exámenes. Solo existen estudios sobre la detección de bordes y contornos, y la lectura de código de barras.

2.2 Base teórica

Píxel

Una imagen digital está formada por un conjunto definido de puntos de forma cuadrada llamados píxeles. La palabra píxel proviene de la combinación de dos palabras inglesas : picture (imagen) y element (elemento).

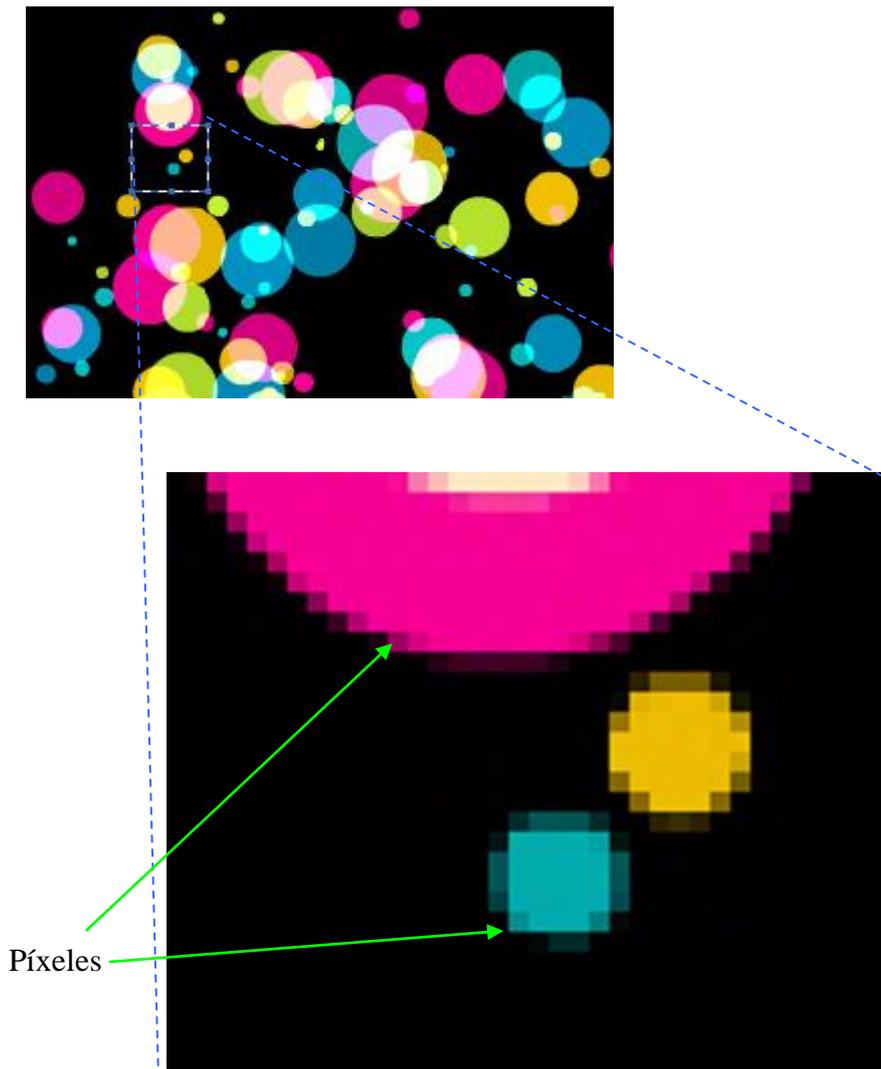


Figura 1.- Píxeles que componen una imagen digital

El color específico de un píxel es una combinación de tres componentes del espectro de colores: rojo (R), verde (G) y azul (B).

Los píxeles se reproducen en la pantalla del monitor, mediante diminutos puntos que poseen los tres componentes de color R, G y B. Cada componente de color del punto se ilumina según el valor de cada componente de color del píxel, lo que determina el tono de color y el brillo del punto. El tamaño físico

de un punto individual en la pantalla, lo determina el fabricante del monitor y se mide en milímetros (mm).

Imagen digital RGB

Una imagen digital está formada por un conjunto de píxeles. Los píxeles de la imagen están dispuestos en una matriz bidimensional, organizada en filas y columnas. Por ejemplo, si una imagen está formada por una matriz de 800 columnas por 500 filas, tiene un tamaño de 800 x 500 píxeles. Cuanto mayor sea el número de filas y columnas, mayor será el detalle de la imagen y mayor será el tamaño del archivo digital.

La profundidad de color de una imagen digital, está determinada por el número de bits que se utilizan para definir el color de un píxel. El número de colores que se pueden representar con n bits, es igual a 2^n . Esto determina el tipo de imagen digital. Existen imágenes digitales en blanco y negro, en escala de grises y a color.

En una imagen digital en blanco y negro, existen solo dos colores (blanco y negro) que se representan mediante un bit por píxel. El valor 0 representa el negro y el 1 el blanco.

En una imagen digital en escala de grises de 8 bits, cada píxel de la imagen se almacena en un byte (8 bits), donde su valor numérico representa su tono de gris, que puede oscilar entre el negro (0) y el blanco (255). Esto quiere decir, que es una imagen donde existen 256 tonos de gris ($2^8 = 256$).

En una imagen a color de 8 bits (color indexado), se utilizan 8 bits por píxel, lo que proporciona hasta 256 colores distintos (0 a 255). Cada valor numérico del píxel, se corresponde con uno de los 256 colores definidos en una paleta de colores, que utiliza a su vez 4 bytes para definir el color: uno para el rojo, uno para el verde, uno para el azul y un byte de relleno con valor cero.

En una imagen a color de 16 bits (color de alta resolución), se utilizan 16 bits por píxel, lo que permite representar hasta 65,536 colores distintos. Cada valor numérico del píxel, se corresponde con uno de los 65,536 colores definidos en una paleta de colores, de características similares a la descrita anteriormente.

En una imagen a color de 24 bits (color verdadero), se asignan tres bytes para especificar el color de un píxel individual, utilizando un byte para cada color (RGB). Los tres bytes (24 bits por píxel), permiten la representación de 16'777,216 colores diferentes (16.7 millones de colores). En este tipo de imagen, no existe ninguna entrada en la paleta de colores, ya que cada color RGB está definido directamente por un byte en el valor de color del píxel.

Existen imágenes a color de 32 bits, cuyo formato es similar al de 24 bits, pero se le ha adicionado un byte para definir la transparencia de la imagen (componente Alfa).

La resolución de una imagen digital, es la medida de la cantidad de píxeles por unidad de longitud, comúnmente píxeles por pulgada, que se suele

abreviar como ppp o dpi (dots per inch). A mayor resolución, mayor número de píxeles en el mismo espacio, y por lo tanto; mayor definición de la imagen.

La calidad de la presentación de imágenes digitales en un monitor, depende de su resolución o de cuantos píxeles pueda mostrar en la pantalla, y cuántos bits se utilizan para representar cada píxel. Los equipos basados en el adaptador de gráficos de video (VGA), pueden mostrar hasta 640 x 480 píxeles, es decir hasta 307,200 píxeles. Los equipos basados en el adaptador de gráficos de video superior (SVGA), pueden mostrar hasta 1024 x 768 píxeles, o un total de 786,432 píxeles.

Formato de los archivos de imagen de mapa de bits

Formato BitMaP

Creado para Microsoft Windows e IBM OS/2, el archivo de imagen BitMaP es un archivo que guarda una imagen digital. Este archivo tiene extensión bmp.

Está compuesto de los siguientes elementos: una estructura de datos llamada BMPInfoHeader (encabezado de información), otra estructura de datos llamada BMPCoreInfo (información del núcleo), un arreglo de bytes llamado RGBQUAD con información de la paleta de colores, y un arreglo de bytes con la información correspondiente a los píxeles que componen la imagen.

La información almacenada en la estructura BMPInfoHeader, es la siguiente:

Offset	Campo	Tamaño	Contenido
0000h	Identificador	2 bytes	<p>Caracteres que identifican el BitMaP. Las entradas pueden ser las siguientes:</p> <p>‘BM’ - Windows 3.1x, 95, 98, NT, XP, ...</p> <p>‘BA’ - OS/2 Arreglo Bitmap</p> <p>‘CI’ - OS/2 Icono a colores</p> <p>‘CP’ - OS/2 Apuntador a color</p> <p>‘IC’ - OS/2 Icono</p> <p>‘PT’ - OS/2 Apuntador</p>
0002h	Tamaño del archivo	1 dword	Tamaño total del archivo en bytes.
0006h	Reservado	1 dword	Reservado para uso futuro.
000Ah	Desplazamiento de los datos	1 dword	Desplazamiento desde el inicio del archivo hasta el comienzo de los bits de datos de la imagen, en bytes.

Nota: Offset se refiere al desplazamiento de la posición del dato, respecto al origen del archivo, y los valores están dados en hexadecimal. Un word es igual a 2 bytes y un dword es igual a 4 bytes.

La información almacenada en la estructura BMPCoreInfo, es la siguiente:

Offset	Campo	Tamaño	Contenido
000Eh	Tamaño de la estructura de información básica del BitMaP	1 dword	Tamaño de la estructura BMPCoreInfo del BitMaP, usado para describir los colores, compresión, resolución, etc. No incluye el tamaño de la paleta de colores ni la de los datos de la imagen. Puede tener los tamaños siguientes: 28h - Windows 3.1x, 95, 98, NT, XP, ... 0Ch - OS/2 1.x F0h - OS/2 2.x
0012h	Ancho	1 dword	Ancho horizontal de la imagen en píxeles.

0016h	Alto	1 dword	Alto vertical de la imagen en píxeles.
001Ah	Planos	1 word	Número de planos en el BitMaP.
001Ch	Bits por Píxel	1 word	<p>Bits por píxel usados para guardar la información de las entradas en la paleta de colores. También identifica indirectamente el número de colores posibles. Los valores pueden ser:</p> <p>1 bit - Monocromo (Blanco y negro)</p> <p>4 bits - 16 colores</p> <p>8 bits - 256 colores</p> <p>16 bits – 65,536 colores (Alta resolución)</p> <p>24 bits – Color verdadero (16.7 millones de colores)</p> <p>32 bits – Color verdadero (16.7 millones de colores), mas componente Alfa</p>
001Eh	Compresión	1 dword	Especificaciones de compresión. Los

			<p>valores pueden ser los siguientes:</p> <p>0 - Ninguno (Identificado también por BI_RGB)</p> <p>1 - RLE 8-bit / píxel (Identificado también por BI_RLE4)</p> <p>2 - RLE 4-bit / píxel (Identificado también por BI_RLE8)</p> <p>3 - Bitfields (Identificado también por BI_BITFIELDS)</p>
0022h	Tamaño de los datos de la imagen	1 dword	Tamaño de los datos de la imagen en bytes. Este número debe ser redondeado a la próxima frontera de 4 bytes.
0026h	Resolución Horizontal	1 dword	Resolución horizontal expresada en píxeles por metro.
002Ah	Resolución Vertical	1 dword	Resolución vertical expresada en píxeles por metro.
002Eh	Colores	1 dword	Número de colores usados por el bitmap. Para un bitmap de 8-bit /

			píxel debe ser 100h o 256.
0032h	Colores Importantes	1 dword	Número de colores importantes. Será igual al número de colores cuando todos los colores son importantes.

El arreglo RGBQUAD, con información de la paleta de colores, está compuesta de la siguiente información:

Offset	Campo	Tamaño	Contenido
0036h	Paleta	N * 4 bytes	Especificación de la paleta de colores. N = número de colores de la paleta. Para cada entrada de la paleta, se usarán cuatro bytes para los valores RGB que describirán cada color, de la siguiente manera: 1 byte para el componente azul 1 byte para el componente verde

			1 byte para el componente rojo
			1 byte de relleno con valor cero

La paleta de colores para imágenes monocromáticas o en blanco y negro (1 bit por píxel) tiene un tamaño de 8 bytes. Se inicia en el offset 0036h (54) y termina en el offset 003Dh (61).

La paleta de colores para imágenes de 16 colores (4 bits por píxel) tiene un tamaño de 64 bytes. Se inicia en el offset 0036h (54) y termina en el offset 0075h (117).

La paleta de colores para escala de grises (8 bit por píxel) tiene un tamaño de 1024 bytes. Se inicia en el offset 0036h (54) y termina en el offset 0435h (1077). Los valores para el rojo, verde y azul en la paleta, son los valores de las entradas para cada color de la paleta.

La paleta de colores para los archivos de imagen de 256 colores (8 bit por píxel) también tiene un tamaño de 1024 bytes, aunque el valor de los colores de la paleta es diferente a los de la escala de grises.

La paleta de colores para los archivos de imagen de 65,536 colores (16 bit por píxel) tiene un tamaño de 262,144 bytes. Se inicia en el offset 0036h (54) y termina en el offset 10036h (65,590).

Los archivos de imagen digital de color verdadero (16.7 millones de colores), no tienen paleta de colores, debido a que el valor de color está indicado

directamente en los tres bytes que definen el valor de los píxeles de la imagen.

El arreglo con la información correspondiente a los píxeles que componen la imagen, está definida de la siguiente manera:

Offset	Campo	Tamaño	Contenido
0XXXh	Datos de los píxeles de la imagen	n bytes	Dependiendo de la especificación de compresión, este campo contiene los bytes de datos que representan los índices de color en la paleta (4, 8, 16 bits) o directamente los datos de la imagen (24 y 32 bits).

El desplazamiento exacto del inicio de los datos de los píxeles de la imagen, depende del tamaño de la paleta de colores.

Formato JPEG

JPEG es un estándar de compresión de imágenes digitales desarrollado por el Joint Photographic Experts Group, de cuyas siglas viene su nombre, y cuya traducción al castellano es grupo de expertos en fotografía. El grupo fue organizado en 1986, expidiendo la norma del estándar en 1992, que fue aprobado como ISO/IEC IS 10918-1 | ITU-T Recommendation T.81. Los

archivos de imagen digital que utilizan este estándar tienen extensión jpg, jif, jfif o jpeg.

El estándar JPEG especifica los procesos para convertir los datos originales de una imagen, en datos de una imagen comprimida, y como realizar el proceso inverso. También, da orientación sobre la manera de aplicar estos procesos en la práctica y especifica las representaciones codificadas de los datos de la imagen comprimida.

Este sistema permite la compresión de imágenes a color y escala de grises, modificando la imagen mediante la eliminación de datos redundantes que no son importantes y suavizando los bordes y áreas que tienen un color similar.

Hay cuatro modos independientes de codificación de la imagen: basado en DCT secuencial, basado en DCT progresiva, sin pérdidas, y jerárquica. Los dos primeros modos están basados en la transformada discreta del coseno (DCT).

El modo DCT secuencial, muestra la imagen en forma secuencial de arriba hacia abajo y de izquierda a derecha, línea por línea.

El modo DCT progresivo, muestra inicialmente solo una parte de la información de cada bloque a la misma resolución que la de la imagen, y va incrementando progresivamente la información mostrada, hasta completar la imagen.

La compresión sin pérdidas, utiliza algoritmos de compresión sin pérdidas como la codificación diferencial y estructuras de predicción simples. La tasa máxima de compresión que se puede lograr es de aproximadamente 2:1. La imagen se muestra de forma similar a la del modo secuencial.

La compresión jerárquica representa la imagen en capas de diferente resolución. Se muestra inicialmente la capa de menor resolución, y luego se van mostrando las demás capas, hasta que finalmente se muestra la imagen completa.

Los pasos definidos por este estándar, para la codificación de una imagen digital basada en DCT secuencial, son los siguientes:

- **Transformación del espacio de color.**- Transforma una imagen en RGB a otro espacio de color llamado YCbCr ó YUV. Este espacio tiene tres componentes: luminancia (Y), saturación (Cb) y tono (Cr). La luminancia es la información del brillo de la imagen (imagen en escala de grises), la saturación es una medida de la pureza del color (cantidad de blanco de la imagen) y el tono es la tonalidad del color (longitud de onda asociada al color). Los últimos dos componentes contienen la información del color, y componen lo que se denomina crominancia.

El cambio de espacio de color, se realiza según las siguientes ecuaciones:

$$Y = 0.257 * R + 0.504 * G + 0.098 * B + 16$$

$$Cb = -0.148 * R - 0.291 * G + 0.439 * B + 128$$

$$Cr = 0.439 * R - 0.368 * G - 0.071 * B + 128$$

- **Submuestreo.**- Consiste en reducir la información del color respecto al brillo. Es un método que introduce pérdidas en la información de la imagen, pero que en la práctica no se percibe, debido a que el ojo humano es mucho más sensible al brillo que al color, es decir; es mucho más sensible al cambio en la luminancia que en la crominancia. El uso del submuestreo no es obligatorio.

Si no se lleva a cabo, se denomina submuestreo 4:4:4 y la imagen no sufre pérdidas. Si se reduce la información de la crominancia a la mitad (reducir a la mitad la información del color en forma horizontal), se denomina submuestreo 4:2:2. Si se reduce la información de la crominancia a la cuarta parte (reducir a la mitad la información del color en forma horizontal y vertical), se denomina submuestreo 4:2:0. Si la imagen inicial estaba en escala de grises, se puede eliminar toda la información de la crominancia.

- **Codificación basada en DCT.**- En el proceso de codificación, la imagen es descompuesta en bloques de 8x8 píxeles, a los cuales se le aplica la DCT, que los transforma en un conjunto de 64 valores, conocidos como coeficientes DCT. Uno de los valores, es llamado coeficiente DC y los 63 restantes son llamados coeficientes AC.

Los 64 coeficientes son sometidos a un proceso de cuantización, utilizando los elementos correspondientes de una tabla de cuantización. Este es el proceso en el que se pierde la mayor parte de la información. Sin embargo, esta pérdida de información no es percibida en la imagen,

debido a que el ojo humano detecta pequeños cambios de brillo en áreas relativamente grandes, pero no cuando el brillo cambia rápidamente en pequeñas áreas, esto permite eliminar las altas frecuencias, sin perder excesiva calidad visual. Las especificaciones de la tabla de cuantización utilizada, son almacenadas junto a la información de la imagen.

Después de la cuantización, el coeficiente DC y los 63 coeficientes AC son preparados para un proceso de codificación entrópica. Para esto, el coeficiente DC se somete a una codificación diferencial, y junto con los 63 coeficientes AC; son convertidos en una secuencia unidimensional en zig-zag. Los coeficientes cuantizados, pasan a un proceso de codificación entrópica, que comprime aún más los datos. Se puede utilizar codificación Huffman o codificación aritmética. Generalmente se utiliza la codificación Huffman, siendo raros los casos en que se utiliza la codificación aritmética. En cualquier caso, las especificaciones de la tabla de codificación, deben ser provistas al codificador y almacenadas junto a la información de la imagen.

Luego de concluido el proceso de codificación antes descrito, se obtiene el archivo de imagen codificada según el estándar JPEG.

El proceso de decodificación, consiste en realizar el proceso inverso al descrito, aplicando la transformada inversa del coseno. La imagen obtenida no es igual a la original, pero generalmente tiene una buena calidad, dependiendo del nivel de compresión utilizado.

Los sistemas de compresión de imágenes JPEG con pérdidas (DCT secuencial, DCT progresiva y Jerárquica con pérdidas), consiguen un nivel de compresión muy alto. Como son sistemas de compresión con pérdidas, hacen que la calidad de la imagen se degrade levemente, a cambio de proporcionar un alto índice de compresión.

El formato de archivo original, tal como se especifica en el anexo B de la norma; es conocido como “Formato de intercambio JPEG” (JIF). Sin embargo, este formato de archivo rara vez se utiliza, principalmente debido a la dificultad de programación de codificadores y decodificadores que apliquen plenamente todos los aspectos de la norma y debido a ciertas deficiencias de la norma.

Otras normas han evolucionado para hacer frente a esta situación. Una de ellas, publicada en 1992; define el “Formato de intercambio de archivos JPEG” (JFIF), que es normalmente utilizado para el almacenamiento y la transmisión de imágenes fotográficas en Internet. Otra norma más reciente, define el “Formato de archivo de imagen intercambiable” (EXIF), que es el formato de imagen usado por las cámaras digitales y otros dispositivos de captura de imágenes fotográficas. Últimamente, ha aparecido una nueva norma denominada JPEG 2000. Sin embargo, en la práctica no se hace distinción de estas diferencias en los formatos, y son llamados simplemente imágenes JPEG.

Los archivos de imagen que utilizan el **formato de intercambio de archivos JPEG** (JFIF), consisten en una secuencia de segmentos que se inician con un

marcador, los cuales comienzan con el byte FF (en hexadecimal), seguido por un byte que define el tipo de marca.

La mayoría de marcadores tienen información adicional a continuación. En este caso, el marcador y su información asociada son conocidos como “cabecera”. En la cabecera, el marcador es seguido por 2 bytes que indican el tamaño de la información en bytes, que contiene el encabezado. Los 2 bytes que indican el tamaño, están incluidos en el conteo, pero no incluyen los 4 bytes de la marca.

Según el estándar, el archivo de imagen JPEG se inicia con el marcador FFD8, denominado SOI (Start Of Image), y termina con el marcador FFD9, denominado EOI (End Of Image).

El archivo JPEG/ JFIF es totalmente compatible con el formato de intercambio JPEG estándar, el único requisito adicional es la presencia obligatoria del marcador FFE0, denominado marcador APP₀ (marcador JFIF); inmediatamente después del marcador SOI. Puede estar presente también el marcador FFE1, denominado APP₁ (marcador EXIF), el cual será interpretado por las aplicaciones que tengan capacidad para ello, caso contrario será ignorado. El formato de intercambio JPEG requiere (al igual que JFIF), que todas las tablas de especificaciones utilizadas en el proceso de codificación, sean codificadas dentro del flujo de bits correspondiente a la imagen JPEG, antes de su uso.

La estructura de los archivos JPEG de las imágenes utilizadas en el presente trabajo, obtenidas mediante el escaneo de los formularios de respuestas, es la siguiente:

Nombre corto	Bytes de la marca	Información útil	Nombre	Contenido
SOI (Start Of Image)	FFD8h	No	Inicio de la imagen	Indica el inicio de una imagen JPEG.
APP₀ (Application)	FFE0h	Tamaño variable	Segmento de la aplicación	<p>Longitud: 2 bytes.</p> <p>Identificador: 5 bytes. Cadena terminada en cero que contiene los caracteres JFIF.</p> <p>Versión: 2 bytes. Usualmente 01 o 02.</p> <p>Unidades: 1 byte. Unidades para las resoluciones de X e Y.</p> <p>0 = Sin unidades.</p> <p>1 = Puntos por pulgada (ppp).</p> <p>2 = Puntos por</p>

Nombre corto	Bytes de la marca	Información útil	Nombre	Contenido
				<p>centímetro.</p> <p>Resolución X: 2 bytes.</p> <p>Resolución Y: 2 bytes.</p> <p>Xthumbnail: (Miniatura X) 0 = Sin miniatura.</p> <p>Ythumbnail: (Miniatura Y) 0 = Sin miniatura.</p> <p>(RGB)n: 3n bytes. Valores RGB de los píxeles de la imagen en miniatura.</p> <p>$n = Xthumbnail * Ythumbnail$</p>
APP1	FFE1h	Tamaño variable	Segmento de la aplicación	<p>Longitud: 2 bytes.</p> <p>Identificador: 5 bytes. Cadena terminada en cero que contiene los caracteres EXIF.</p> <p>Contiene además,</p>

Nombre corto	Bytes de la marca	Información útil	Nombre	Contenido
				información de la cabecera EXIF, como referencias, datos de la miniatura, datos del dispositivo que produjo la imagen, etc.
DQT (Define Quantization table)	FFDBh	Tamaño variable	Define tabla de cuantización.	<p>Longitud: 2 bytes.</p> <p>Primera tabla de cuantización para baseline JPEG:</p> <ul style="list-style-type: none"> • Precisión e índice de la tabla de cuantización: 1 byte. Precisión: 4 bits más significativos. Índice: 4 bits menos significativos. • Valores de cuantización: 64 bytes. Las tablas de cuantización son almacenadas

Nombre corto	Bytes de la marca	Información útil	Nombre	Contenido
				en formato zig-zag.
DQT (Define Quantization table)	FFDBh	Tamaño variable	Define tabla de cuantización.	Longitud: 2 bytes. Segunda tabla de cuantización para baseline JPEG (similar a la marca anterior).
SOF₀ (Start Of Frame)	FFC0h	Tamaño variable	Inicio del cuadro del proceso DCT básico (baseline DCT).	Longitud: 2 bytes. P: Muestra la precisión en bits (usualmente 8 en baseline DCT). X: 2 bytes Y: 2 bytes. Nf: 1 byte. Número de componentes de la imagen. 3 = Imagen a color (RGB). 1 = Imagen en escala de grises.

Nombre corto	Bytes de la marca	Información útil	Nombre	Contenido
				<p>Nf veces: compuesto de:</p> <ul style="list-style-type: none"> • Component ID: 1 byte. • Factores de muestreo X e Y: 1 byte. • Los primeros 4 bits son el componente X. Los últimos 4 bits son el componente Y. • Número de tabla de cuantización: 1 byte.
<p>DHT (Define Huffman Table)</p>	<p>FFC4h</p>	<p>Tamaño variable</p>	<p>Define una tabla Huffman.</p>	<p>Longitud: 2 bytes.</p> <p>Índice: 1byte. Si es > 15 es una tabla AC. Sino es una tabla DC.</p> <p>Bits: 16 bytes.</p>

Nombre corto	Bytes de la marca	Información útil	Nombre	Contenido
				Valores Huffman: Número de bytes = suma de los 16 bytes anteriores.
DHT (Define Huffman Table)	FFC4h	Tamaño variable	Define una tabla Huffman.	Igual que el cuadro anterior.
DHT (Define Huffman Table)	FFC4h	Tamaño variable	Define una tabla Huffman.	Igual que el cuadro anterior.
DHT (Define Huffman Table)	FFC4h	Tamaño variable	Define una tabla Huffman.	Igual que el cuadro anterior.
SOS (Start of scan)	FFDAh	Tamaño variable	Inicio de la exploración	Longitud: 2 bytes. Número de componentes: 1 byte. N veces:

Nombre corto	Bytes de la marca	Información útil	Nombre	Contenido
				<p>Component ID: 1 byte.</p> <p>Números de las tablas AC y DC: 1 byte. Los 4 primeros bits para el número de la tabla DC, y los 4 últimos para la tabla AC.</p> <p>S_s: 1 byte.</p> <p>S_e: 1 byte.</p> <p>A_h y A_l: 1 byte.</p>
				<p>Datos de la imagen comprimida.</p>
<p>EOI</p> <p>(End Of Image)</p>	<p>FFD9h</p>	<p>No</p>	<p>Fin del archivo</p>	<p>Indica la finalización del archivo de imagen.</p>

Sistema de coordenadas de la imagen

La posición que ocupa cada píxel de la imagen en la pantalla del monitor, está determinada por un sistema de coordenadas donde el eje horizontal crece hacia la derecha y el eje vertical crece hacia abajo.

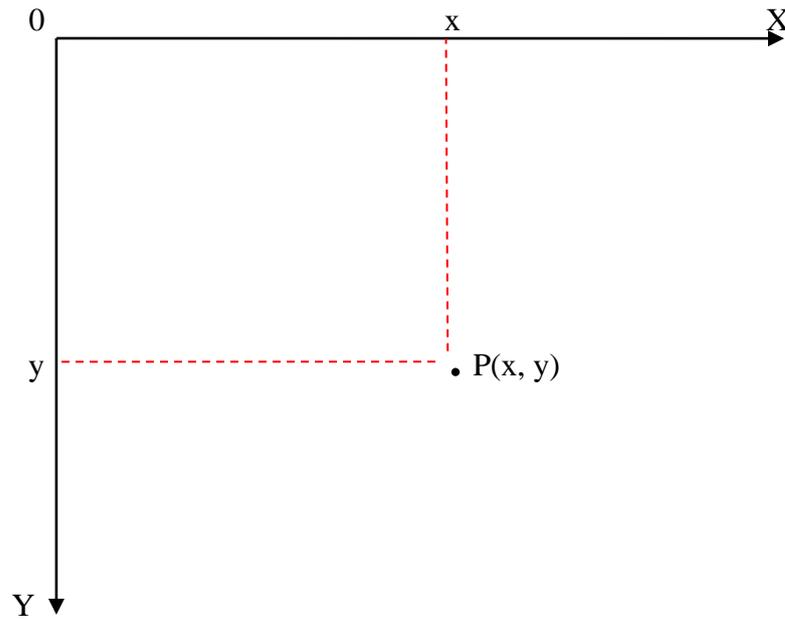


Fig. 2.- Sistema de coordenadas de la imagen

Una imagen digital, está compuesta por un arreglo bidimensional de píxeles, en donde la posición de cada píxel dentro del arreglo, se corresponde con una posición dentro del sistema de coordenadas de la imagen reproducida en el monitor.

Traslación de los ejes de coordenadas

Es el cambio de posición de los ejes de coordenadas, sin rotarlos. Cada uno de los ejes permanece paralelo a su posición original.

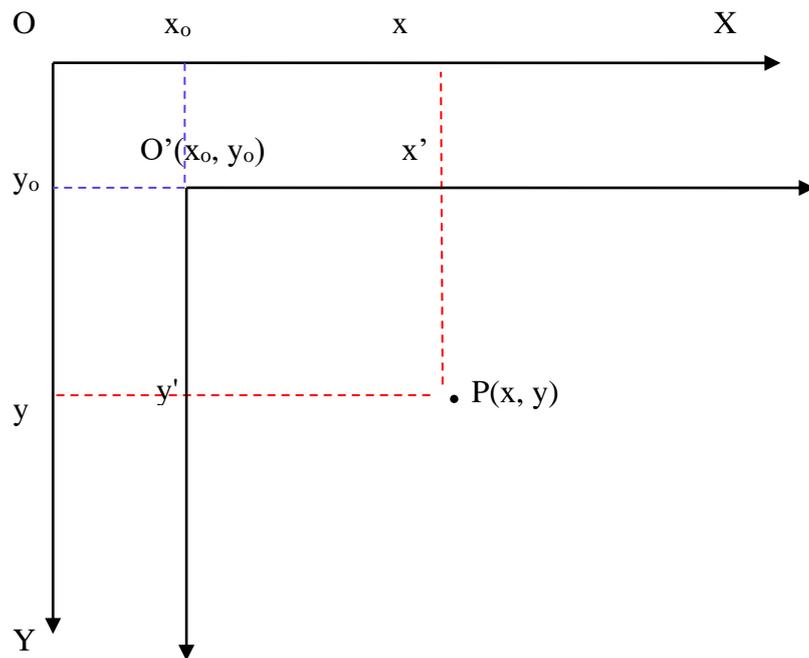


Fig. 3.- Translación del sistema de coordenadas de la imagen

Una vez que el origen de un sistema de coordenadas se cambia a un punto $O'(x_0, y_0)$, cada punto $P(x, y)$ en el sistema original, tendrá un nuevo conjunto de coordenadas $P'(x', y')$, de acuerdo a las siguientes relaciones:

$$x' = x - x_0$$

$$y' = y - y_0$$

Rotación de los ejes de coordenadas

Es el cambio de orientación de los ejes de coordenadas, conservando el punto de origen.

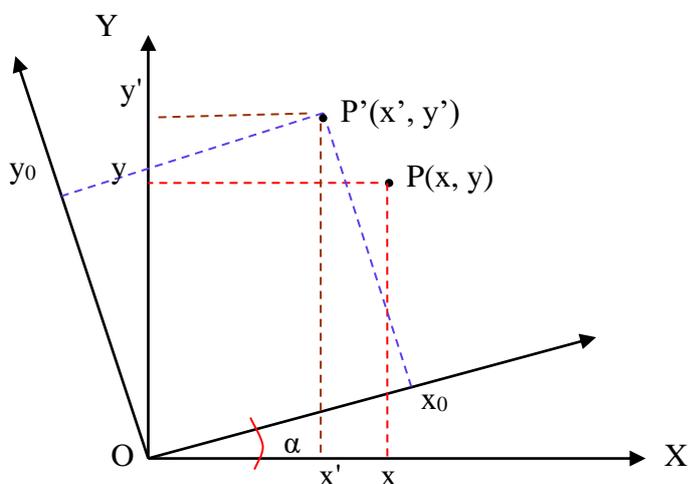


Fig. 4.- Rotación de ejes de coordenadas

Sea un punto $P(x, y)$ en un sistema de coordenadas cartesianas. Si se rotan los ejes de coordenadas un ángulo α , el punto P se traslada de posición a las nuevas coordenadas $P'(x', y')$. Las nuevas coordenadas del punto, están relacionadas con las coordenadas originales, según las siguientes expresiones:

$$x' = x \cos \alpha + y \sin \alpha$$

$$y' = -x \sin \alpha + y \cos \alpha$$

Escalamiento de los ejes de coordenadas

Consiste en modificar la escala original de los ejes de coordenadas, manteniendo el mismo punto de origen. El escalamiento de un punto, se realiza multiplicando las coordenadas x e y por un factor de escalamiento, de modo que el punto tiene una nueva ubicación $P'(x', y')$. Se puede aplicar la misma escala a ambos ejes, o diferentes escalas a cada uno. Como resultado de esta operación, un punto se traslada a una nueva posición, proporcionalmente a la escala aplicada.

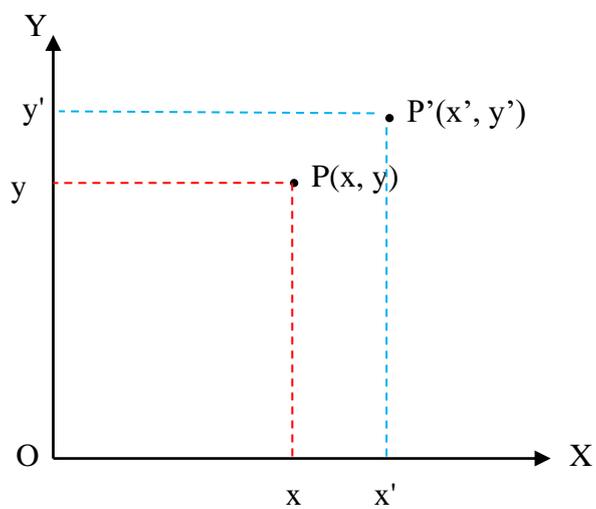


Fig. 5.- Escalamiento de ejes

La relación entre las nuevas coordenadas del punto, en función de las coordenadas originales, es la siguiente:

$$x' = x \cdot f_{ch}$$

$$y' = y \cdot f_{cv}$$

Donde:

f_{ch} = Factor de escalamiento horizontal.

f_{cv} = Factor de escalamiento vertical.

III. MATERIALES Y MÉTODOS

El presente trabajo desarrolla un sistema OMR (reconocimiento de marcas ópticas) para la calificación de formularios de respuestas de exámenes, en los cuales se han efectuado marcas ópticas como respuestas a las preguntas de un examen.

Las imágenes digitales de los formularios con las marcas realizadas por los alumnos, son obtenidas mediante el escaneo de los formularios y almacenadas en una carpeta, desde donde el sistema OMR las procesará para obtener la calificación de todos los alumnos que rindieron el examen.

El software desarrollado detecta las alternativas de respuestas marcadas en la imagen digital del formulario, y según la ubicación de cada respuesta calcula la calificación obtenida por el alumno.

3.1 Tipo y diseño de investigación

De acuerdo a la orientación, el tipo de investigación es aplicada. Se desarrolla un software para la calificación de formularios de respuestas de exámenes.

De acuerdo a la técnica de contrastación, el tipo de investigación es descriptiva. Se obtienen un conjunto de datos de los formularios de exámenes, para luego ser procesados y obtener los resultados, sin alterar ni modificar los datos, preservando la forma original de los formularios.

El diseño de investigación es descriptivo.

3.2 Instrumentos de recolección de la información

Como instrumento para recolectar información, se utilizaron formularios de respuestas para exámenes, en los cuales se realizaron marcas ópticas a modo de respuestas.

3.3 Plan de procesamiento y análisis de la información

Los formularios marcados fueron procesados por el sistema y el resultado de la calificación fue comparado con el obtenido en forma manual, lo que permitió determinar el porcentaje de aciertos en la calificación, utilizando el sistema OMR.

3.4 Organización del formulario de respuestas

El formulario diseñado permite seleccionar las respuestas para 100 preguntas de examen, para lo cual presenta un conjunto de 5 casilleros de alternativas de respuesta por pregunta, debiendo el alumno marcar un casillero mediante el sombreado del mismo, como respuesta a una determinada pregunta.

Nº	a	b	c	d	e
1					
2					
3					
4					
5					
-					

Figura 6.- Casilleros de alternativas de respuestas

Asimismo, el formulario presenta un conjunto de 4 columnas de casilleros numerados del cero al nueve, en donde el alumno deberá registrar su código de inscripción o matrícula, marcando los casilleros correspondientes al código que se le ha asignado.

Código			
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			

Figura 7.- Casilleros para registrar el código

La posición del área de búsqueda de las marcas del alumno en cada casillero, se calcula en base a una marca de posición que se describirá más adelante. La búsqueda de las marcas, se realiza examinando el nivel de gris de los píxeles en el área seleccionada.

Para una resolución de imagen de 96 ppp, el área a examinarse en cada casillero es de 64 píxeles (8 píxeles horizontales por 8 píxeles verticales).

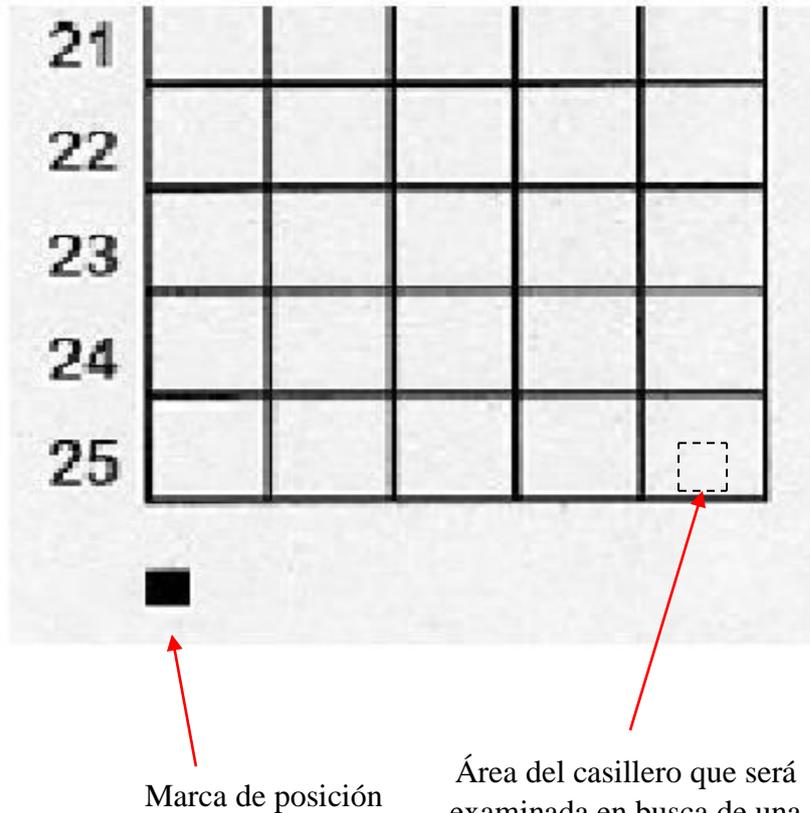


Figura 8.- Área del casillero que será examinada en busca de una marca

Se han dispuesto dos marcas de referencia de posición: una en la esquina inferior izquierda y otra en la esquina inferior derecha, marcas que permiten calcular la pendiente de la recta que une los píxeles inferiores de las marcas de posición de la imagen del formulario, y el ángulo de inclinación de la imagen respecto al eje horizontal.

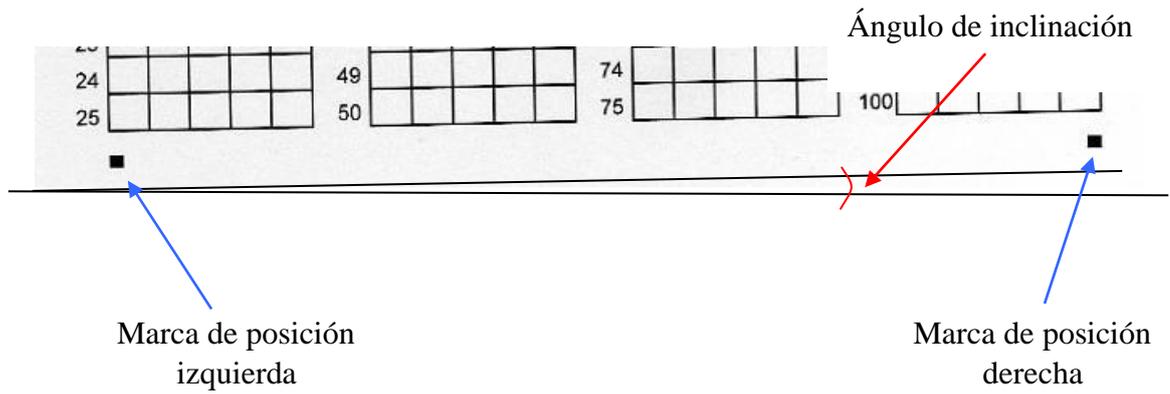


Figura 9.- Marcas de posición y ángulo de inclinación de la imagen

Se utiliza la posición del píxel inicial inferior de la marca de posición izquierda y la posición del píxel final inferior de la marca de posición derecha, para determinar la inclinación de la imagen del formulario, por lo que se deberá detectar primero la posición de éstos píxeles, y a partir de esta información; se puede determinar la pendiente de la recta que une los píxeles antes mencionados y el ángulo de inclinación de la imagen.

En la siguiente figura, se muestra la imagen completa del formulario diseñado, en el que se indican los elementos principales del mismo.

Código					Datos Personales																							
0					Apellido Paterno			Apellido Materno																				
1					Nombres																							
2																												
3					Especialidad																							
4					Instrucciones - Rellene el cuadrado completamente y solo uno por pregunta. - Use solo lápiz 2B. - En caso de error, borre cuidadosamente y rellene el que considere correcto. - No arrugue ni maltrate este formulario.																							
5																												
6																												
7																												
8					SECCIÓN DE RESPUESTAS																							
9					Nº	a	b	c	d	e	Nº	a	b	c	d	e	Nº	a	b	c	d	e	Nº	a	b	c	d	e
	1					26					51					76												
	2					27					52					77												
	3					28					53					78												
	4					29					54					79												
	5					30					55					80												
	6					31					56					81												
	7					32					57					82												
	8					33					58					83												
	9					34					59					84												
	10					35					60					85												
	11					36					61					86												
	12					37					62					87												
	13					38					63					88												
	14					39					64					89												
	15					40					65					90												
	16					41					66					91												
	17					42					67					92												
	18					43					68					93												
	19					44					69					94												
	20					45					70					95												
	21					46					71					96												
	22					47					72					97												
	23					48					73					98												
	24					49					74					99												
	25					50					75					100												

Casillero de alternativa para registrar el código

Casillero de alternativa de respuesta

Marca de posición izquierda

Marca de posición derecha

Figura 10.- Formulario de respuestas

3.5 Escaneo del formulario de respuestas

El escaneo del formulario se realizó a una resolución de 96 ppp (puntos por pulgada), en escala de grises de 8 bits, y en formato JPG. El escaneo en esta resolución, genera una imagen que tiene las mismas dimensiones en píxeles que la imagen del formulario de respuestas diseñado. El escaneo con un tipo de imagen en escala de grises de 8 bits, proporciona una imagen con 256 niveles de gris, niveles suficientes para poder determinar las marcas realizadas en el formulario de respuestas. Se puede utilizar cualquier formato de imagen, con o sin compresión de datos, según las necesidades del usuario.

3.6 Características básicas del software

El sistema desarrollado, se ha denominado “Universal OMR” y abre archivos con extensión **omr**, que corresponden a un proyecto OMR para la calificación de formularios de respuestas para exámenes.

El software está desarrollado para procesar imágenes escaneadas con una resolución de 96 ppp, aunque se pueden utilizar resoluciones superiores. El tipo de imagen no es determinante en la obtención del resultado. Se pueden utilizar imágenes en cualquier formato y en escala de grises o a color.

Para el desarrollo del sistema OMR, se utilizó el IDE Microsoft Visual Basic 2005. Asimismo, se utilizó Microsoft Access para la implementación de la base de datos, y Microsoft Excel para el diseño e impresión del formulario, y la exportación de los resultados obtenidos en el proceso de calificación de los alumnos.

El sistema operativo requerido para el software desarrollado, es Microsoft Windows 2000/2003/XP/Vista, con NET Framework 2.0 o superior.

3.7 Descripción del proceso de examen usando el sistema Universal OMR

El desarrollo de los eventos, para llevar a cabo el examen de admisión o cualquier otro examen que tenga características similares, utilizando el sistema OMR desarrollado; es el siguiente:

1. Se registran los datos de los alumnos en el sistema, así como el nombre de la especialidad, los cuales se guardarán en la base de datos.
2. Se ingresan al sistema las claves de respuestas para cada pregunta, indicando por cada número de pregunta, la opción que es la respuesta correcta, así como el puntaje que le corresponde (el valor por defecto es 1).
3. Se editan e imprimen los formularios que serán utilizados en el examen de admisión o cualquier otro tipo de prueba similar.
4. Los alumnos ingresan a las aulas en las que rendirán su examen. Reciben un cuestionario de preguntas y un formulario de respuestas, en el que registrarán su código o número de inscripción, marcando los casilleros correspondientes al código.
5. Los alumnos rinden su examen y registran sus respuestas mediante marcas en los casilleros del formulario de respuestas.

6. Al culminar la prueba, entregan el formulario de respuestas a las personas a cargo del control del examen.
7. Se reúnen todos los formularios y se procede a escanearlos, preferentemente a una resolución de 96 o 100 ppp y en formato JPEG.
8. Las imágenes digitales obtenidas, son almacenadas en una carpeta de archivos del sistema OMR, destinada específicamente para almacenar los archivos de imagen, o de considerarlo pertinente; en cualquier otra carpeta.
9. De ser necesario, se selecciona la carpeta en la que se encuentran las imágenes. Luego, se indica al sistema OMR que inicie el procesamiento de las imágenes y el reconocimiento de marcas, obteniendo los resultados correspondientes a cada alumno, los cuales serán almacenados en la base de datos.
10. Se procede a imprimir los resultados, ya sea en forma general o por especialidades, o exportarlos a una hoja de cálculo Excel.

3.8 Descripción del procesamiento de las imágenes mediante el sistema Universal OMR

Las acciones que realiza el software Universal OMR para procesar las imágenes, son las siguientes:

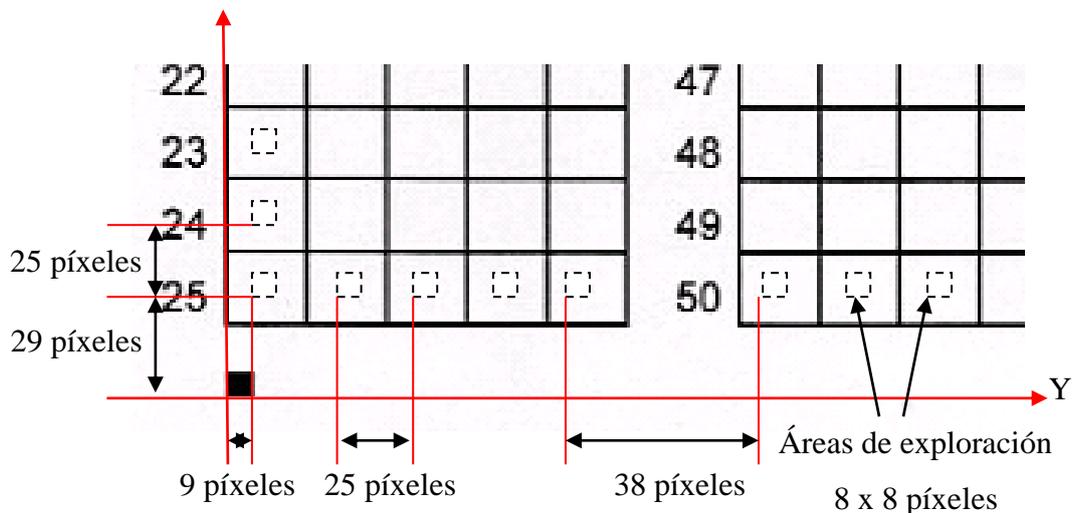
1. Abre el archivo de imagen correspondiente a uno de los formularios de respuestas. Abrirá archivos de imagen de mapa de bits, con extensión jpg, bmp o cualquier otra extensión conocida. Si la resolución de la imagen es

de 96 dpi, asigna un factor de escalamiento igual a 1 para el procesamiento de las marcas. Si la resolución de la imagen es diferente, calcula el factor de escalamiento adecuado y procesa las marcas. Luego, si no ha sido el último archivo de imagen de la carpeta, pasa a abrir el siguiente archivo de imagen, para su procesamiento.

2. Procede a ubicar las marcas de posición izquierda y derecha, explorando los píxeles en la vecindad de las esquinas inferiores de la imagen, hasta ubicar el primer píxel inferior de la marca de posición izquierda, y el último píxel inferior de la marca de posición derecha. Almacena las coordenadas de los puntos de ubicación de las marcas, en objetos de tipo punto (Point en Visual Basic 2005). Si no ubica estas marcas, inserta un mensaje de error en la lista de mensajes de error y considera al archivo como erróneo, y si no ha sido el último archivo de imagen de la carpeta; pasa a abrir el siguiente archivo de imagen, para su procesamiento.
3. Si ubica las marcas de posición, procede a una traslación de los ejes de coordenadas, siendo la posición del píxel de la marca de posición izquierda el nuevo origen de los ejes de coordenadas, que utilizará para el procesamiento de la imagen. A continuación, procede a realizar una rotación del eje Y de 180°, con lo que se tiene un sistema de coordenadas con origen en la marca de posición izquierda y un eje Y que crece hacia arriba.
4. En el nuevo sistema de coordenadas de la imagen, y con la información de la posición de los píxeles inferiores de las marcas de posición, determina la

pendiente de la recta que une los píxeles de las marcas, y calcula el ángulo de inclinación respecto al eje X. El ángulo de inclinación, es utilizado como ángulo de rotación para el sistema de ejes coordenados, de manera que queden orientados con la imagen.

- Una vez alineados los ejes coordenados con la imagen, inicia el cálculo de la posición del píxel inferior izquierdo de las áreas de exploración de los casilleros correspondientes al código del alumno (10 filas de 4 casilleros), y de los casilleros de respuestas (4 bloques de 25 preguntas y 5 casilleros de respuesta por cada pregunta), que utilizará para la búsqueda de las marcas realizadas por el alumno. El área de exploración es un cuadrado de 8x8 píxeles en el centro del casillero. Esta área de exploración es para una imagen con una resolución de 96 ppp. Si la resolución es diferente, calculará el factor de escalamiento adecuado para calcular el área de exploración, y la posición de los píxeles de referencia de las áreas de exploración en cada casillero.



6. Una vez determinadas las posiciones de los píxeles de referencia de las áreas de exploración, procede a determinar el código del alumno, utilizando las áreas de exploración correspondientes al código. Recupera el valor de los píxeles dentro del área de exploración, comprobando si los valores corresponden a un nivel de gris menor a 150. Considera que el casillero está marcado, si encuentra por lo menos 5 píxeles con el nivel de gris antes mencionado, o el número adecuado según el factor de escalamiento.
7. Verifica que el código tenga 4 dígitos, que exista en la base de datos, y que no sea duplicado. Si detecta que no se cumple alguna de las condiciones anteriores, considera que ha existido un error al llenar el código. Muestra el mensaje de error correspondiente en la lista de mensajes de error, y si no ha sido el último archivo de imagen de la carpeta; pasa a abrir el siguiente archivo de imagen, para su procesamiento.
8. Si el código es correcto, procede a explorar los casilleros correspondientes a las respuestas de las 100 preguntas del examen. Explora los 5 casilleros que corresponden a cada pregunta, comenzando en los casilleros de la pregunta 1 hasta la pregunta 100. La exploración las realiza en busca de marcas efectuadas por el estudiante. Considera que el casillero está marcado, si encuentra por lo menos 5 píxeles con un nivel de gris menor a 150.
9. El sistema define un arreglo de 100 elementos de tipo numérico, para almacenar los resultados de la exploración de las respuestas. El valor

inicial de cada elemento del arreglo es cero. Si determina que existe una marca, asigna al elemento del arreglo correspondiente al número de pregunta, un valor igual al de su posición dentro del conjunto de casilleros, más un peso adicional de 5; lo cual permitirá determinar posteriormente al procesar los resultados, si el alumno marcó 2 o más casilleros correspondientes a una misma pregunta.

10. Una vez culminada la exploración de los 500 casilleros (5 casilleros por 100 preguntas), procede a calcular el puntaje obtenido por el alumno. En la base de datos del sistema, existe una tabla que almacena los datos correspondientes a la respuesta por cada pregunta, incluyendo la posición relativa del casillero que le corresponde a cada respuesta correcta. Para determinar si la respuesta es correcta, recupera el valor del elemento del arreglo correspondiente al número de pregunta y le resta 5, obteniendo un valor que representa la posición relativa de la marca. El resultado de la resta se clasifica en tres rangos: valores entre 1 y 5, valores mayores a 5 y valor igual a -5. Si el resultado se encuentra en el rango de 1 a 5, ha realizado una sola marca. Si al compararlo con el valor de la posición almacenada en la tabla de repuestas del sistema, éstos coinciden; la respuesta es correcta y el puntaje total es incrementado en el puntaje correspondiente a la pregunta. Si al compararlo con el valor de la posición almacenada en la tabla de repuestas del sistema, éstos no coinciden; la respuesta es incorrecta. Si el resultado obtenido es mayor a 5, significa que el alumno marcó 2 o más casilleros en una misma fila, y si el resultado es -5; significa que no marcó ningún casillero.

$$\text{Pos} = X(i) - 5$$

11. Obtenido el puntaje, procede a actualizar la lista de imágenes procesadas correctamente. Luego, almacena el puntaje en la tabla de datos que forma parte de la base de datos del sistema.
12. Si no ha sido el último archivo de imagen de la carpeta, abre el siguiente archivo de imagen, para su procesamiento.

3.9 Descripción de los modos de presentación de los resultados

El sistema Universal OMR, brinda tres modos de presentación de los resultados obtenidos: impresión de resultados general (sin considerar la especialidad), impresión de resultados por especialidad y exportación de los resultados a una hoja de cálculo de Excel.

En el modo de impresión de resultados general, el sistema imprime los resultados obtenidos por todos los estudiantes que rindieron el examen, ordenados por el puntaje alcanzado, en orden descendente.

En el modo de impresión de resultados por especialidad, el sistema imprime los resultados obtenidos por los estudiantes que rindieron el examen para una determinada especialidad, ordenados por el puntaje alcanzado, en sentido descendente.

En el modo de exportar los resultados a Excel, el sistema crea un libro de Excel y almacena los resultados obtenidos por todos los estudiantes que rindieron el examen, o de los estudiantes que rindieron el examen para una

determinada especialidad, en una hoja de cálculo; ordenados por el puntaje alcanzado y en sentido descendente. El sistema abre el aplicativo Excel y muestra la hoja de cálculo con los resultados.

3.10 Herramientas computacionales

Para el desarrollo del presente estudio, se utilizó una computadora con microprocesador Intel Celeron de 2.4 GHz con 512 MB de RAM, el escáner de un equipo multifuncional Epson Stylus CX 4100, y una impresora láser HP LaserJet P2014 para la impresión de los formularios de respuestas.

3.11 Manual de uso del sistema Universal OMR

La carpeta que contiene el software del sistema Universal OMR, se debe copiar a una unidad de disco duro de la computadora.

Para iniciar la aplicación, abrir la carpeta y hacer doble clic sobre el archivo Universal OMR. Se mostrará la siguiente ventana:

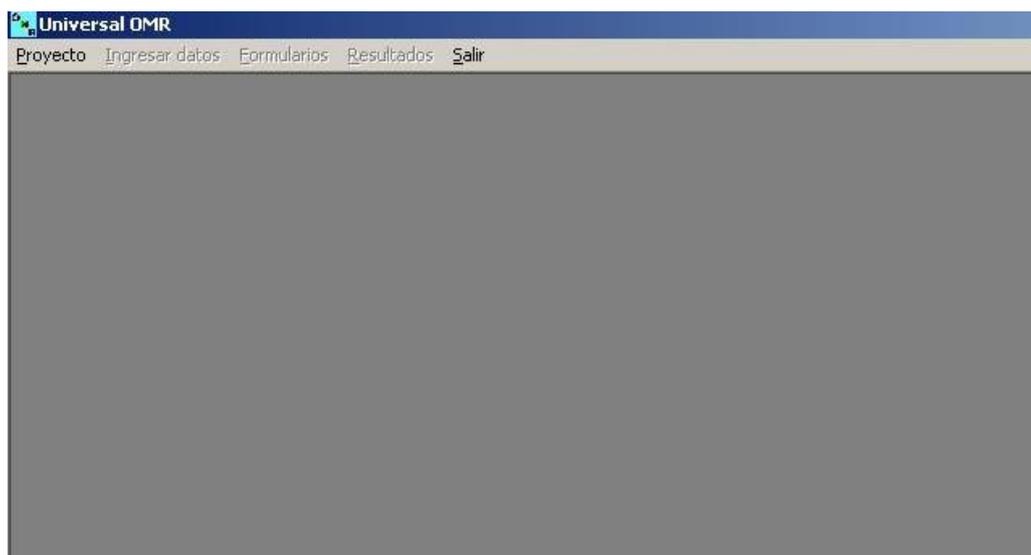


Figura 11.- Ventana inicial del sistema Universal OMR

La ventana inicial muestra un menú con las opciones **Proyecto** y **Salir**. Si se selecciona la opción **Salir**, se sale del sistema, y si se selecciona la opción **Proyecto**, se muestra un submenú con tres opciones: **Nuevo**, **Abrir** y **Guardar**. La opción **Guardar** aparece inicialmente deshabilitada.

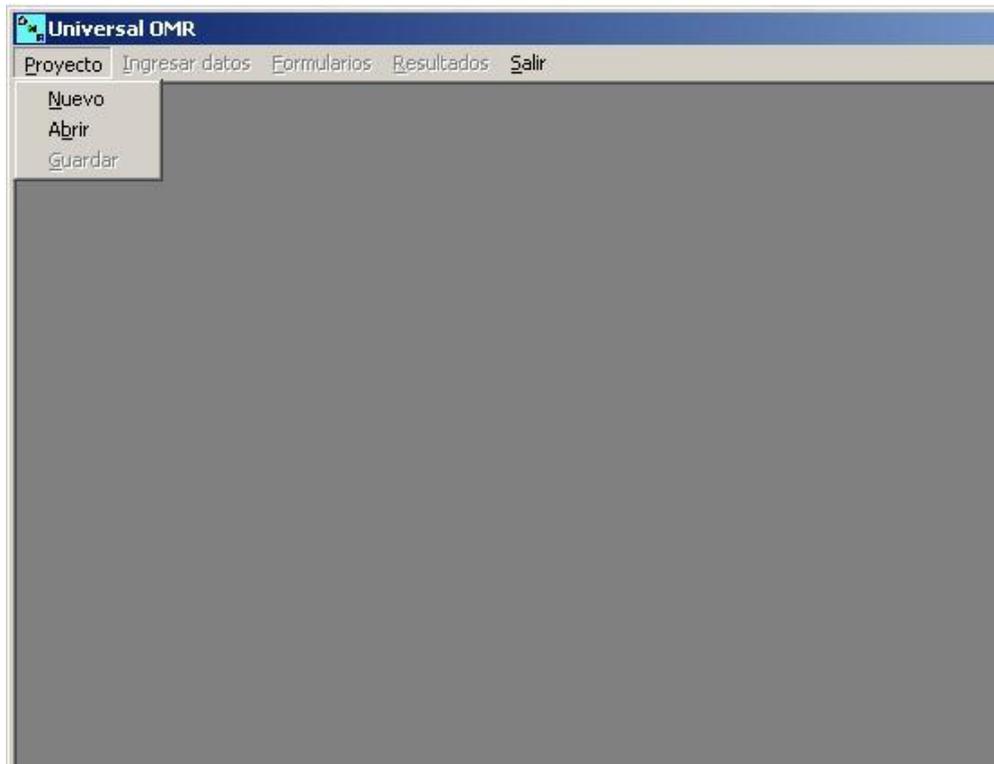
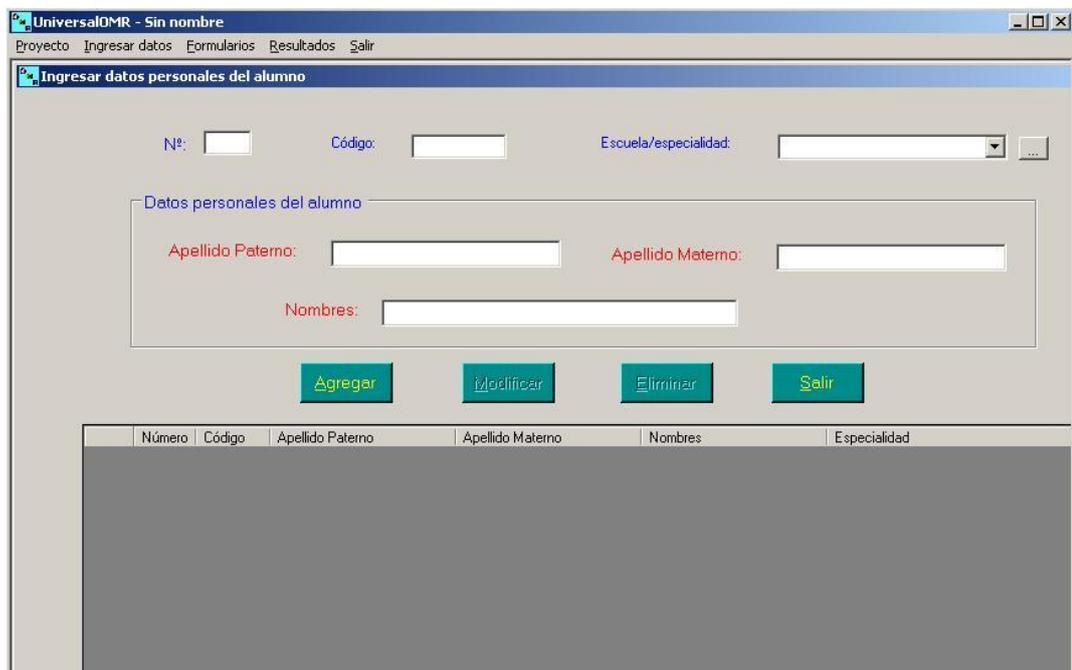


Figura 12.- Opciones del menú Proyecto

Si se selecciona la opción **Nuevo**, se estará creando un nuevo proyecto OMR. Si se selecciona la opción **Abrir**, se mostrará una ventana que permitirá abrir un archivo OMR guardado, permitiendo continuar el ingreso de datos y luego procesar las imágenes de los formularios.

3.11.1 Nuevo proyecto

Cuando se selecciona la opción **Nuevo**, se crea una base de datos para el nuevo proyecto y se muestra inicialmente la ventana para ingresar datos al nuevo proyecto. El nuevo proyecto tiene como título “Universal OMR - Sin nombre”.



The screenshot shows a software window titled "UniversalOMR - Sin nombre". The menu bar includes "Proyecto", "Ingresar datos", "Formularios", "Resultados", and "Salir". The main area is titled "Ingresar datos personales del alumno" and contains the following fields and controls:

- Fields for "Nº:", "Código:", and "Escuela/especialidad:" (a dropdown menu).
- A section titled "Datos personales del alumno" containing fields for "Apellido Paterno:", "Apellido Materno:", and "Nombres:".
- Four buttons: "Agregar", "Modificar", "Eliminar", and "Salir".
- A table at the bottom with columns: "Número", "Código", "Apellido Paterno", "Apellido Materno", "Nombres", and "Especialidad".

Figura 13.- Ventana para ingresar los datos del alumno

Si se pulsa con el ratón sobre el botón **Salir**, se cierra la ventana. Si se pulsa sobre el botón **Agregar**, se habilitan las cajas de texto y automáticamente comienza la numeración de los estudiantes registrados. También, los textos de los botones **Agregar** y **Salir** cambian a los textos **Aceptar** y **Cancelar**, respectivamente.

Nº: Código: Escuela/especialidad:

Datos personales del alumno

Apellido Paterno: Apellido Materno:

Nombres:

Número	Código	Apellido Paterno	Apellido Materno	Nombres	Especialidad

Figura 14.- Ventana con los datos ingresados del alumno

Se debe ingresar el código del alumno, en cual deberá tener 4 caracteres. Asimismo, se ingresa el nombre de la escuela/especialidad, y el apellido paterno, apellido materno y nombres del alumno.

Una vez ingresados los datos, si se pulsa sobre el botón **Aceptar**, se almacenarán los datos en la base de datos del proyecto y se regresa a la pantalla inicial. Se habilitan los botones **Modificar** y **Eliminar**.

Si se pulsa sobre el botón Cancelar, se regresa a la pantalla inicial sin guardar los datos.

Para ingresar el nombre de la escuela/especialidad, existen dos opciones: la primera es ingresar directamente el nombre en el cuadro de lista desplegable, y la segunda es pulsar sobre el botón con tres puntos

ubicado al costado derecho del control antes mencionado, abriéndose la ventana que permite ingresar todos los nombres, que luego estarán disponibles para el proyecto en el cuadro de lista desplegable.



Figura 15.- Ventana para ingresar los nombres de la escuela/especialidad

El número y el nombre de la escuela/especialidad, se ingresan en los casilleros correspondientes de la tabla de datos que muestra la ventana. Luego de ingresar los nombres, se debe pulsar sobre el botón **Guardar** para que los datos ingresados sean almacenados en la base de datos del proyecto, cerrándose la ventana. Si se pulsa sobre el botón **Salir**, se saldrá de la ventana sin guardar los datos ingresados.

3.11.2 Abrir proyecto

Cuando se selecciona la opción **Abrir**, se muestra una ventana que permite seleccionar el proyecto OMR que desea abrirse.

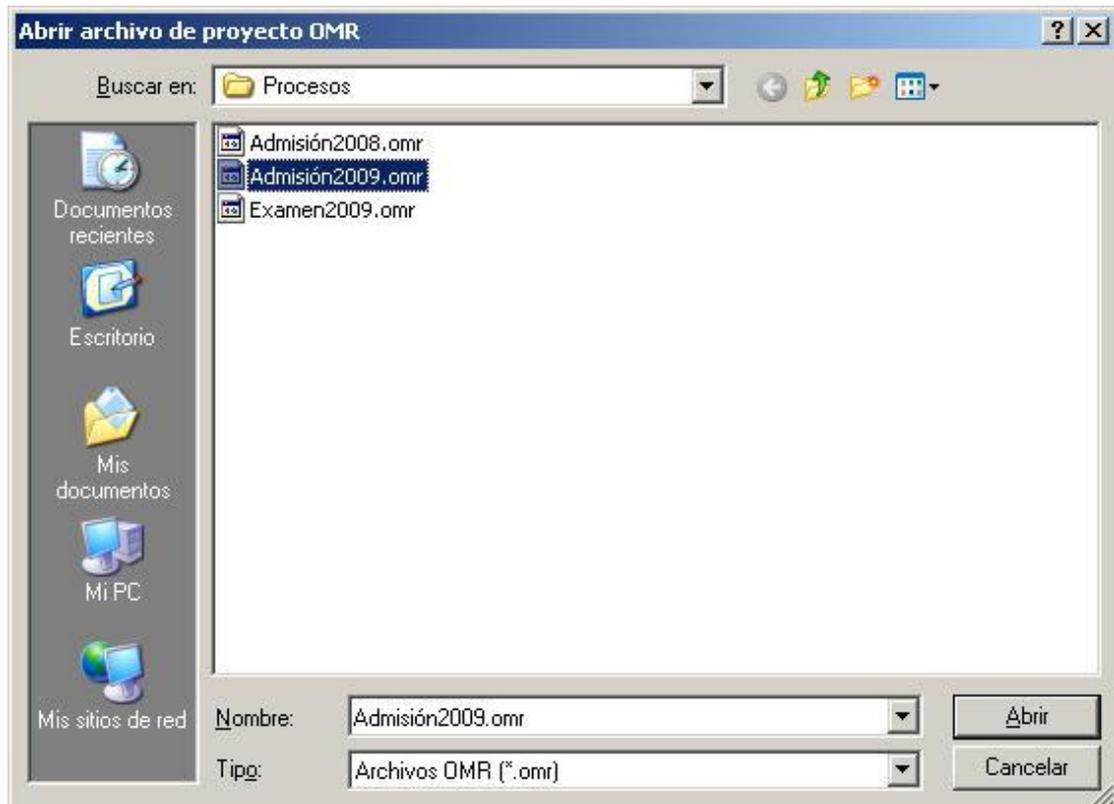


Figura 16.- Ventana para abrir proyecto

Si se pulsa el botón **Aceptar**, se cargarán los datos del proyecto seleccionado, para que el usuario pueda continuar ingresando datos y procesar los formularios.

Ingresar datos personales del alumno

Nº: Código: Escuela/especialidad: ...

Datos personales del alumno

Apellido Paterno: Apellido Materno:

Nombres:

Número	Código	Apellido Paterno	Apellido Materno	Nombres	Especialidad
1	0001	Pérez	Rodríguez	Juan	Ingeniería informática y de Sistemas
2	0002	Morales	Sarmiento	David	Arquitectura
3	0003	Paredes	Flores	María	Ingeniería informática y de Sistemas
4	0004	Sánchez	García	Ana	Contabilidad
5	0005	Coral	Vega	Armando Javier	Administración
6	0006	López	Tranca	Miguel	Arquitectura
7	0007	Mejía	Morales	Camila	Contabilidad
8	0008	Lugo	Oro	Yesenia	Arquitectura
9	0009	Pérez	Alegría	Armando	Ingeniería informática y de Sistemas

Figura 17.- Ventana con los datos del proyecto abierto

3.11.3 Menú Ingresar datos

Este menú presenta 2 opciones: **Datos del alumno** y **Claves**.

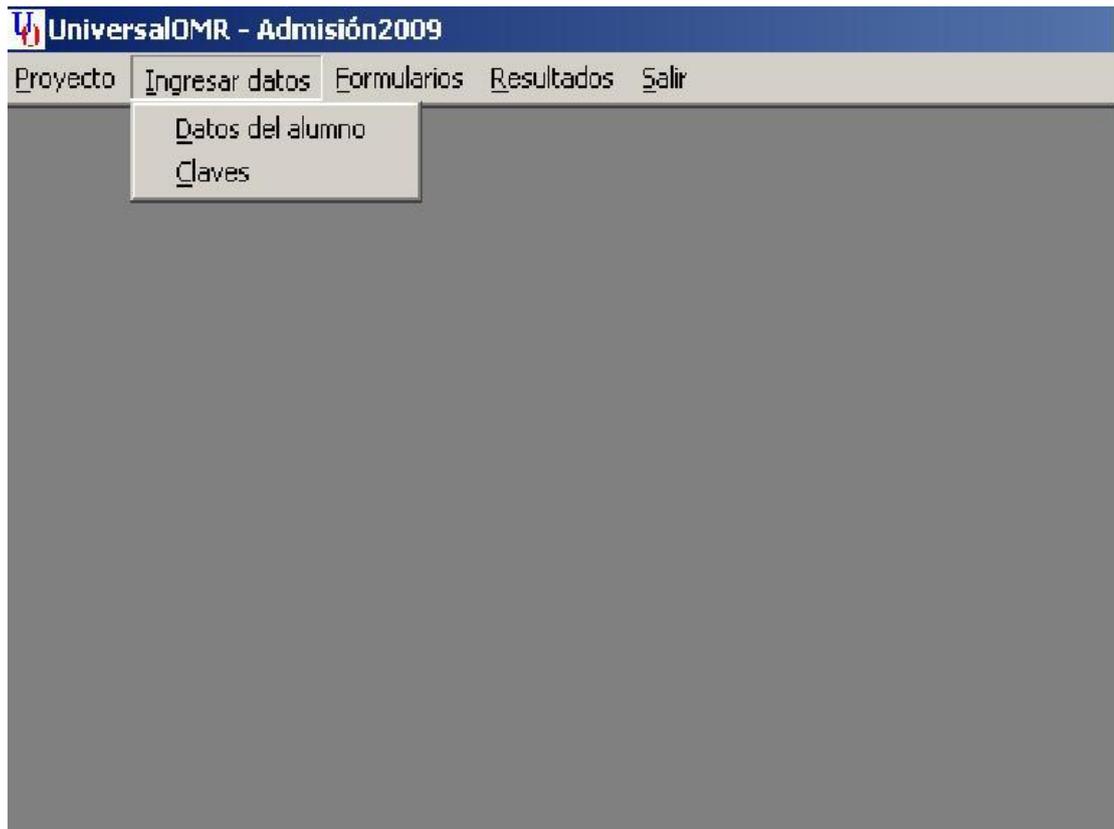


Figura 18.- Ventana con las opciones del menú Ingresar datos

Si se selecciona la opción **Datos del alumno**, se abre la ventana de la figura 17.

Si se selecciona la opción Claves, se abre la siguiente ventana:

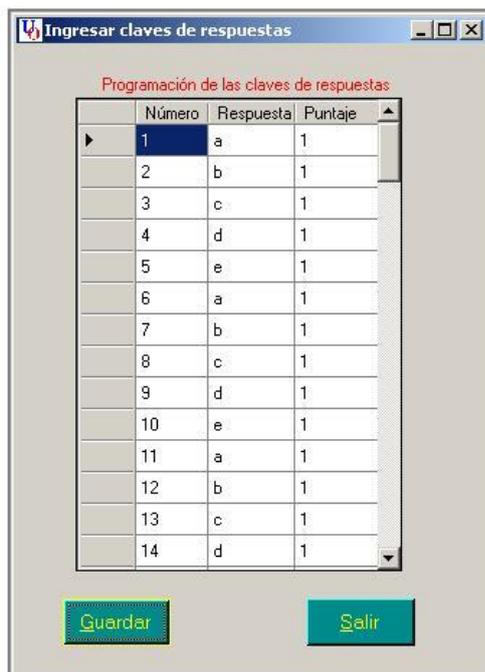


Figura 19.- Ventana para ingresar las claves de respuestas y el puntaje

Los datos se ingresan en las casillas de la tabla, teniendo cuidado de que no se ingresen valores duplicados en la columna del número de pregunta.

Si se ingresan los datos y se pulsa sobre el botón **Guardar**, se guardan los datos en una tabla de la base de datos del sistema. Si se pulsa en el botón **Salir**, se cerrará la ventana sin guardar los datos.

3.11.4 Menú Formularios

Este menú presenta dos opciones: **Imprimir formularios** y **Procesar formularios**.

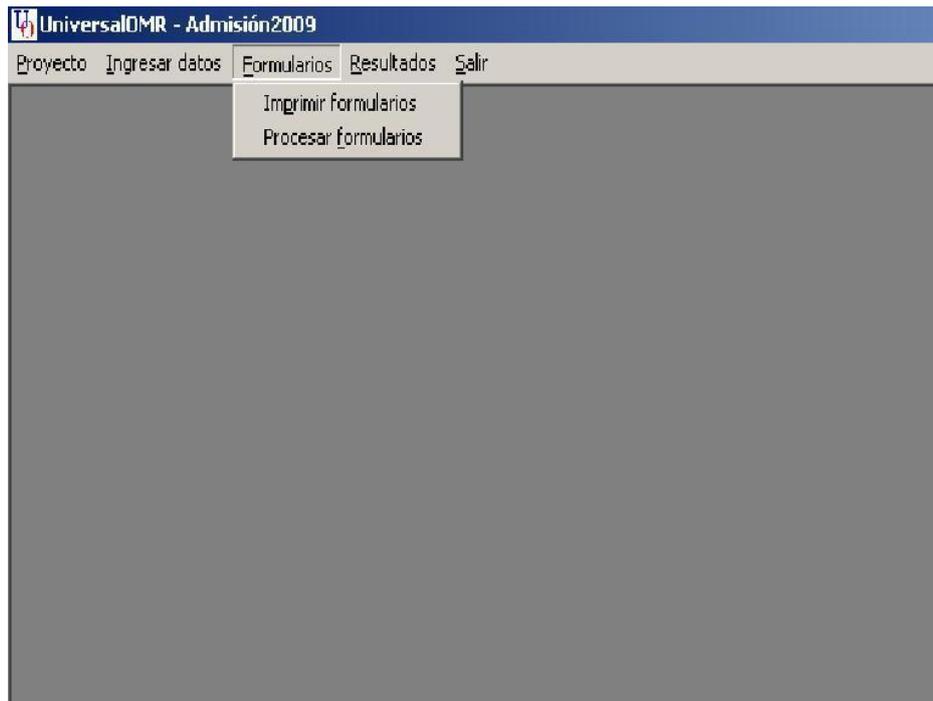


Figura 19.- Ventana con el menú formularios

Si se selecciona la opción **Imprimir formularios**, se abre una hoja de cálculo de Excel conteniendo el formulario a imprimir. Se puede editar el título del formulario, pero no se puede modificar la parte que contiene los casilleros de respuestas y las marcas de posición.

Si se selecciona la opción Procesar formularios, se abre la siguiente ventana:

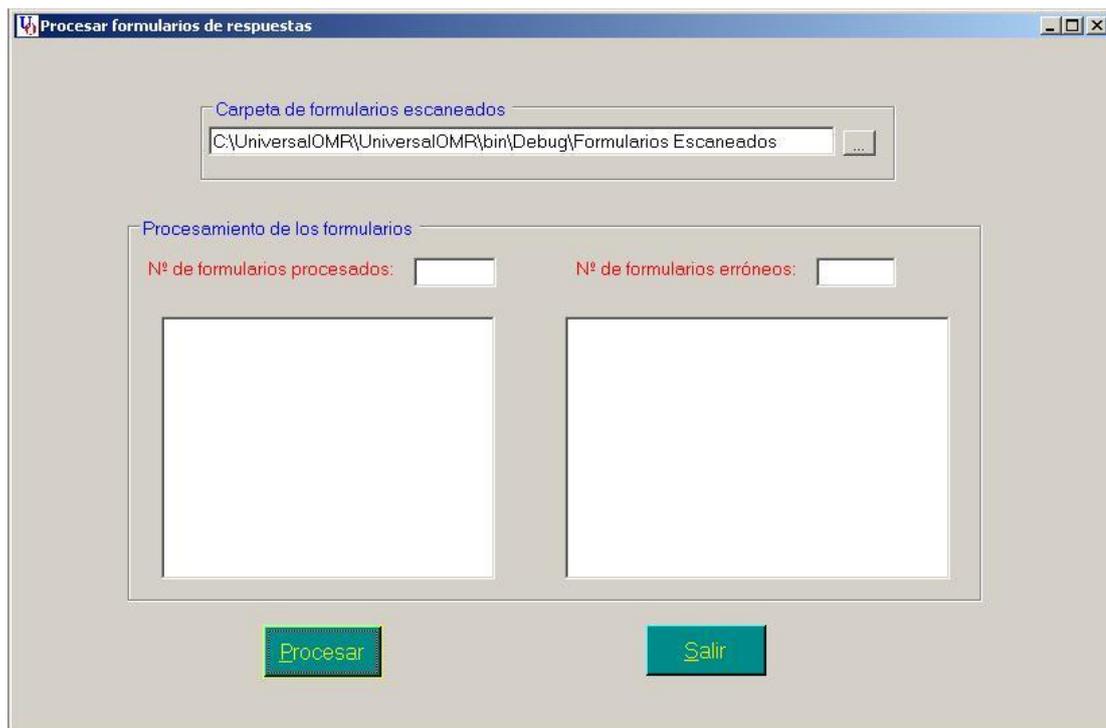


Figura 20.- Ventana para procesar los formularios

En la parte superior de la ventana, se encuentra una caja de texto mostrando la ruta por defecto de la carpeta que contiene las imágenes de los formularios escaneados que se procesarán. El botón con 3 puntos que se encuentra a la derecha de la caja de texto, permite seleccionar la carpeta que contiene las imágenes. Si se pulsa este botón, se abre la siguiente ventana:



Figura 21.- Ventana para seleccionar la carpeta que contiene las imágenes

Esta ventana es similar a las ventanas de selección de carpetas usada por las aplicaciones de Windows. Luego de seleccionar la carpeta, se pulsa sobre el botón aceptar, con lo que la ruta de la nueva carpeta aparece en la caja de texto.

Si se pulsa sobre el botón **Procesar**, se iniciará el procesamiento de todos los archivos de imagen de los formularios escaneados que se encuentren en la carpeta seleccionada, almacenándose los resultados en una tabla de la base de datos del sistema.

En la parte inferior de la ventana de la figura 20, aparecen 2 cajas de lista. En la lista de la izquierda, se registrarán los nombres de los archivos

de imagen procesados correctamente, y en la lista de la derecha, se registrarán los nombres de los archivos de imagen erróneos.

En la parte superior de estas cajas de lista, se encuentran cajas de texto en las que se indicarán el número de formularios procesados correctamente y erróneos, respectivamente.

Luego de haber terminado el procesamiento de las imágenes, si es necesario procesar adicionalmente alguna imagen por haberse cometido algún error al escanear un formulario o cuando existen códigos duplicados, se deberán almacenar las imágenes adicionales en la carpeta llamada “Formularios Adicionales” o cualquier carpeta diferente a la que contiene las imágenes procesadas, y procesarlas desde esa ubicación. Los resultados adicionales serán almacenados en la base de datos del sistema.

3.11.5 Menú Resultados

Si se selecciona este menú, se abre la siguiente ventana:

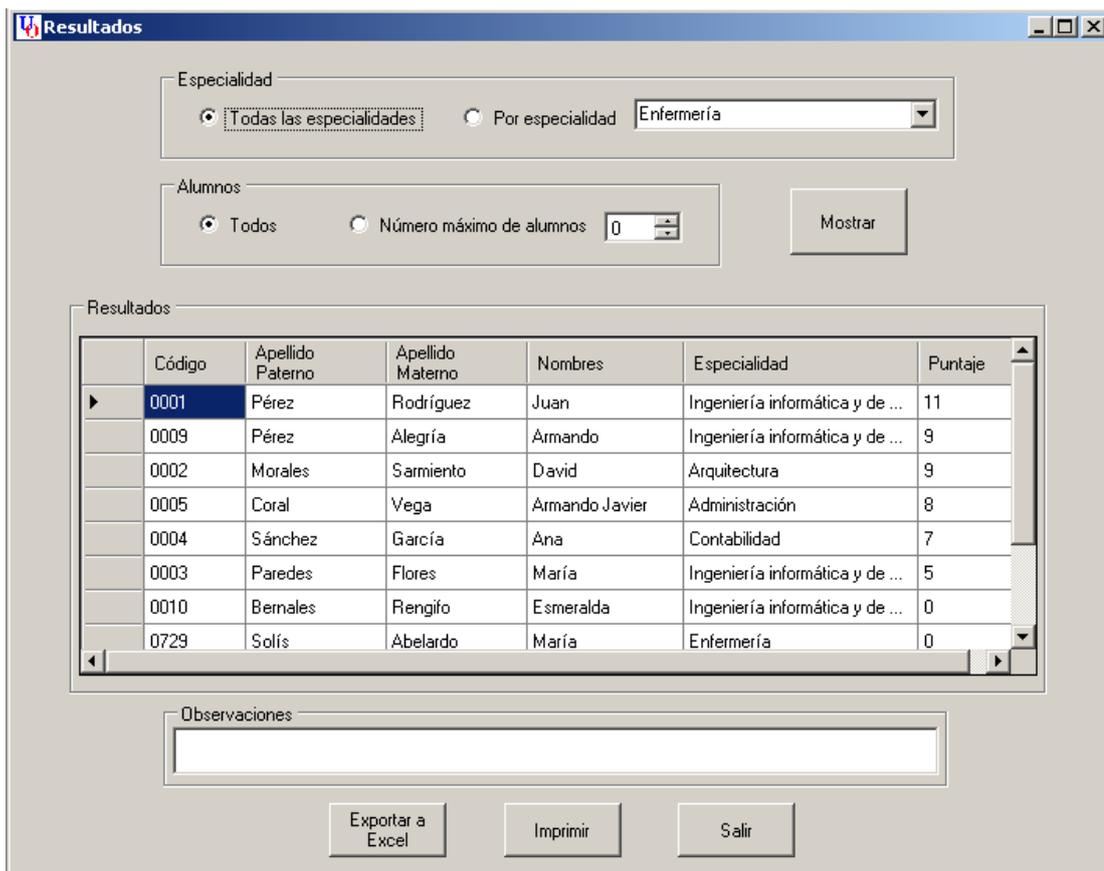


Figura 22.- Ventana que muestra los resultados

En la parte inferior de la ventana, aparecen tres botones: **Exportar a Excel**, **Imprimir** y **Salir**. Si se pulsa sobre el botón **Exportar a Excel**, se abre una hoja de cálculo de Excel al que se han exportado los resultados.

Microsoft Excel - xls1B4.xls

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

Seguridad...

	A	B	C	D	E	F	G
1	Número	Código	Apellido Paterno	Apellido Materno	Nombres	Especialidad	Puntaje
2	1	1	Pérez	Rodríguez	Juan	Ingeniería informática y de Sistemas	68
3	9	9	Pérez	Alegría	Armando	Ingeniería informática y de Sistemas	61
4	2	2	Morales	Sarmiento	David	Arquitectura	57
5	5	5	Coral	Vega	Armando Javier	Administración	53
6	4	4	Sánchez	García	Ana	Contabilidad	50
7	3	3	Paredes	Flores	María	Ingeniería informática y de Sistemas	49
8	11	10	Bernales	Rengifo	Esmeralda	Ingeniería informática y de Sistemas	49
9	10	729	Solís	Abelardo	María	Enfermería	48
10	8	8	Lugo	Oro	Yesenia	Arquitectura	46
11	7	7	Mejía	Morales	Camila	Contabilidad	45
12	6	6	López	Tranca	Miquel	Arquitectura	44

Figura 23.- Resultados exportados a Excel

Si se pulsa sobre el botón **Imprimir**, se abre una ventana que permite imprimir los resultados.

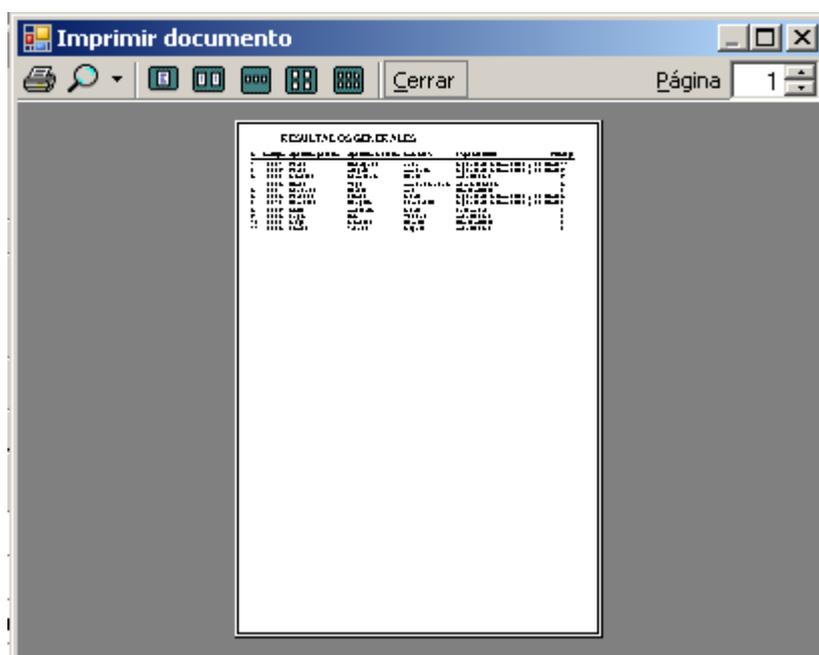


Figura 24.- Ventana para imprimir los resultados

IV. RESULTADOS

4.1 Métodos de procesamiento del archivo de imagen del formulario

Para el procesamiento de las imágenes de los formularios, se ensayaron dos métodos de procesamiento:

- a) Procesar directamente el archivo de la imagen, abriéndolo como un archivo secuencial.
- b) Crear un objeto BitMap, y luego almacenar la imagen del formulario en el objeto, procesándolo como una imagen.

En el método del caso a, se abre el archivo de imagen como un archivo secuencial, recuperándose los valores de los píxeles mediante un objeto BinaryReader, que lee un flujo de bytes. La desventaja es que solo se pueden procesar archivos de imagen sin codificar, como es el caso de los archivos de imagen en el formato BitMap de Windows (extensión bmp). Esto se debe a que en los archivos codificados, la información de los píxeles de la imagen está codificada y tendría primero que decodificarse, como es el caso del formato JPEG con compresión de datos. Es decir, los archivos deberían ser primero transformados al formato BitMap para ser procesados.

Igualmente, solo es aplicable a archivos de imagen que no estén inclinados, debido a que el valor de los píxeles se recupera secuencialmente en un solo flujo, y una inclinación en la imagen, hará que los píxeles de las marcas de

posición no aparezcan en una misma secuencia (como en el caso que estuvieran en una misma línea), imposibilitando su detección.

En el método del caso b, se crea un objeto Bitmap, en el cual se almacena la imagen digital. Se puede utilizar el sistema de coordenadas de la imagen para recuperar el valor de los píxeles, mediante el método GetPixel del objeto Bitmap; accediéndose a los datos de la imagen en forma aleatoria.

Otra ventaja de este método, es que se pueden abrir imágenes almacenadas en archivos de imagen con diferentes formatos.

La desventaja, es que la velocidad de proceso disminuye al utilizar el método GetPixel y bucles para la recuperación del valor de los píxeles, siendo el tiempo empleado para el procesamiento de una imagen de alrededor de 0.5 segundos, en la computadora utilizada para las pruebas.

Se decidió por la utilización del segundo método, ya que aporta más ventajas, como la capacidad de procesar imágenes en varios formatos de imagen y la simplificación de la elaboración del código, al utilizarse un sistema de coordenadas.

4.2 Formato de las imágenes

Se pueden utilizar imágenes de los formularios de respuestas en formato BMP, JPEG, TIFF, GIF o PNG. Sin embargo, se observó que algunos formatos de imagen como GIF y PNG, no están soportados por algunos escáneres como formatos para los archivos de destino de las imágenes.

Asimismo, el tamaño de los archivos de imagen obtenidos al escanear un formulario, varía según el formato del archivo. Para una imagen de 208 mm. de ancho por 250 mm. de alto, escaneada en escala de grises de 8 bits y 96 ppp de resolución, se obtuvieron los siguientes tamaños de archivo:

- Formato BMP: 727 KB.
- Formato JPEG (nivel de compresión 10) 226 KB.
- Formato JPEG (nivel de compresión 60) 93.6 KB.
- Formato TIFF 725 KB.

Se transformó el archivo de imagen BMP a los formatos GIF y PNG, obteniéndose los siguientes tamaños:

- Formato GIF 519 KB.
- Formato PNG 457 KB.

Se observa que el formato de imagen que produce archivos más pequeños, es el formato JPEG. En este formato, el nivel de compresión predeterminado del escáner utilizado fue de 10 : 1.

Se realizaron pruebas del sistema con imágenes de un formulario escaneado en formato JPEG, con diferentes niveles de compresión. Se fue incrementando en 5 puntos el nivel de compresión, para determinar el nivel máximo que admite el sistema sin fallar. Se halló que el nivel máximo de compresión soportado por el sistema es de aproximadamente 60 : 1.

El motivo por el que el sistema finalmente falló, es porque el aumento del nivel de compresión genera más pérdidas en los datos de la imagen, y en un nivel de compresión alto, las pérdidas de datos son tan severas que el sistema no puede determinar las marcas de posición, considerando que la imagen no corresponde a un formulario.

4.3 Nivel de gris de los píxeles

Se realizaron pruebas sobre el valor que toman los píxeles en la escala de grises en los archivos de imagen obtenidos con el escáner, correspondientes a las marcas negras de posición impresas en el formulario por la impresora láser, las marcas borradas y las marcas de respuestas; para los formatos de imagen BitMap y JPEG. El escaneo de los formularios se realizó en escala de grises de 8 bits (256 niveles de gris), y a una resolución de 96 ppp.

En la siguiente tabla, se muestran los rangos de valores en escala de grises de los píxeles, obtenidos para las marcas señaladas en los formatos BitMap y JPEG:

Tipo de marca	Formato BitMap	Formato JPEG
Marca de inicio impresa por la impresora láser	0	0 a 6
Marca de respuesta borrada	184 – 213	188 – 212
Marca de respuesta realizada con lápiz 2B	14 – 125	18 – 123

Marca de respuesta realizada con lápiz HB	21 – 130	23 – 125
---	----------	----------

La diferencia entre los valores hallados para los píxeles correspondientes al color negro, para los formatos BitMap y JPEG; se debe a la pérdida de información introducida por el sistema de compresión que utiliza el formato JPEG.

En base a los valores antes indicados, se seleccionó 150 como umbral para distinguir entre un píxel correspondiente a una marca (0 a 149) y no marcado (150 a 255).

4.4 Inclinación de la imagen

Se determinó que utilizando un escáner para papel A4, la máxima inclinación que puede esperarse en la imagen de un formulario que se encuentre dentro del panel de cristal del escáner, es de aproximadamente 1.73° . El sistema está diseñado para procesar imágenes con esta inclinación y aún mayores, rotando los ejes de coordenadas de la imagen.

4.5 Tiempo de procesamiento de las imágenes

El tiempo necesario para el procesamiento de la imagen de un formulario de respuestas escaneado, no depende del formato del archivo en que se almacenó la imagen, esto debido a que al asignar el archivo de imagen a un objeto Bitmap, la imagen se convierte automáticamente al formato BitMap, y la

función GetPixel recupera el píxel en la posición que se le indica, desde el objeto Bitmap.

El tiempo de proceso de las imágenes de los formularios, depende directamente de la velocidad del microprocesador de la computadora, y de la cantidad de memoria RAM instalada. Esto se debe a que el objeto Bitmap al que se asignó la imagen, se encuentra almacenado en la memoria, y al accederse a mayor velocidad a la memoria; se pueden recuperar los píxeles más rápidamente. También, la ejecución del código es más rápida, lo que incrementa la velocidad de procesamiento. La cantidad de memoria instalada influye directamente en la frecuencia con que se accede al disco duro en busca del código e información del proceso en ejecución. El acceso al disco duro es un proceso que demanda tiempo, debido a la lentitud de este periférico, por lo que una mayor cantidad de memoria permite almacenar más código y datos, reduciendo la frecuencia de acceso al disco duro.

4.6 Pruebas de simulación

Se realizaron pruebas de procesamiento de las imágenes de los formularios de respuestas, obtenidos mediante el escaneo de los mismos. Los tipos de pruebas realizados con el sistema Universal OMR, son los siguientes:

- Prueba utilizando 25 imágenes de formularios correctamente escaneados, que contienen respuestas correctas, incorrectas, en blanco y con dos o más marcas (una de las marcas es correcta).

- Prueba utilizando imágenes de los 25 formularios anteriores, en los cuales se han realizado modificaciones borrando algunas marcas y marcado otros casilleros de respuestas.
- Prueba con 25 imágenes de formularios, 15 escaneados en forma correcta y 10 escaneados en forma incorrecta (invertidos).
- Prueba con imágenes de 10 formularios escaneados con una resolución mayor a 96 ppp.
- Prueba con imágenes de 10 formularios escaneados con una inclinación máxima, 5 en sentido horario y 5 en sentido antihorario.
- Prueba con imágenes de 25 formularios escaneados, de los cuales 10 contienen códigos duplicados.

Resultados:

- En la prueba de calificación de los 25 formularios de respuestas, conteniendo repuestas correctas, incorrectas, en blanco y con dos o más marcas, se obtuvo un porcentaje de aciertos del 100%. El sistema calificó solamente las marcas correctas, desechando las respuestas con dos o más marcas, aún si una de las marcas era correcta.
- En la prueba de calificación de los 25 formularios de respuestas conteniendo marcas borradas, se obtuvo un porcentaje de aciertos del 100%. El sistema discriminó correctamente las marcas borradas.

- En la prueba de formularios escaneados en forma incorrecta, se verificó que los 15 formularios escaneados correctamente fueron calificados, mientras que los 10 formularios escaneados incorrectamente, fueron detectados y el nombre de los archivos aparecieron en la lista de imágenes erróneas.
- En la prueba de formularios escaneados a una resolución mayor a 96 ppp, se utilizaron 5 imágenes de formularios escaneados a 200 ppp y 5 escaneados a 300 ppp. En ambos casos fueron correctamente calificados, aunque el tiempo de proceso casi se duplicó para la resolución de 200 ppp y prácticamente se triplicó para la resolución de 300 ppp. Esto se debe a que se incrementa el área (cantidad de píxeles) que se explorará en busca de una marca.
- En la prueba de los formularios inclinados, se hizo la prueba escaneando los formularios con la máxima inclinación que puede alcanzarse, manteniendo el formulario dentro del panel de cristal del escáner. Se escanearon 5 formularios inclinados en sentido horario y 5 en sentido antihorario. Las imágenes de los 10 formularios fueron procesados correctamente.
- En la prueba de códigos duplicados, el sistema calificó correctamente las imágenes de los primeros 15 formularios cuyos códigos no eran duplicados. Los nombres de los formularios que se procesaron y que tenían códigos duplicados, fueron mostrados en la lista de formularios erróneos, indicando que se trata de formularios con código duplicado.

V. DISCUSIÓN

- 5.1** En los métodos de procesamiento de la información de las imágenes, obtenida mediante escaneo de las mismas, se ensayaron dos modos de procesamiento del archivo de imagen: como un archivo secuencial y como un archivo de imagen. Se decidió utilizar el último de los modos, porque permite procesar un archivo de imagen codificado en cualquier formato, aunque la velocidad de proceso es menor.
- 5.2** En lo que corresponde al formato de las imágenes, el que aporta más ventajas respecto al tamaño de la imagen comprimida, es el formato JPEG; el cual aunque utiliza un algoritmo de compresión de información con pérdidas, con niveles de compresión bajos, la confiabilidad de la información es óptima y se obtiene un ahorro importante de espacio en disco.
- 5.3** Por definición, un punto blanco tiene un valor de píxel de 255. Las marcas negras realizadas con lápiz tienen un valor de píxel entre 14 a 130, lo cual permite diferenciar fácilmente las marcas realizadas en los casilleros de respuestas. Sin embargo, una situación que se debe tener en cuenta, es que se pueden haber marcado y luego borrado algunos casilleros de respuestas, lo que produce píxeles cuyos valores se encuentran entre 184 a 212. Lo mencionado, obliga a establecer valores de umbral para los valores que representan una marca válida. El tipo de lápiz utilizado no es determinante en el valor de píxel de una marca.

5.4 Para el procesamiento de formularios que accidentalmente hayan sido escaneados en forma inclinada, es obligatorio que aparezcan dentro de la imagen las dos marcas de posición, puesto que son referencias necesarias para calcular el ángulo en que se rotarán los ejes de coordenadas de la imagen. Se optó por rotar los ejes de coordenadas en vez de rotar la imagen, porque el volumen de información a procesar es menor e incide de manera decisiva en el tiempo de proceso.

5.5 El tiempo de procesamiento de la imagen de un formulario escaneado, depende básicamente de las características técnicas de la computadora a utilizarse. Al tratarse del procesamiento de imágenes mediante funciones de la plataforma NET, es aconsejable la utilización de una computadora actualizada con tarjeta de video AGP con 128 MB de RAM o superior. Otro factor que influye en la velocidad de proceso, es la resolución con que se realiza el escaneo, puesto que aumenta el número de píxeles que se deberán revisar para detectar una marca. La resolución nominal de escaneo es de 96 ppp. en escala de grises, y si se utiliza una resolución de escaneo superior; el tiempo se incrementará proporcionalmente al tamaño del archivo obtenido. Igualmente, el tamaño del archivo se incrementa si el escaneo se realiza en colores, con el consiguiente aumento en el tiempo de proceso.

VI. CONCLUSIONES

1. Se ha desarrollado un eficiente sistema OMR para la calificación de formularios de respuestas de exámenes, utilizando un escáner convencional y una computadora personal.
2. Para la impresión de los formularios, es suficiente con imprimirlo en una impresora láser.
3. Es posible la automatización de la calificación de formularios de respuestas de exámenes, reduciendo el tiempo necesario para ello, en comparación al tiempo que demanda la calificación manual.
4. Con el uso del sistema universal OMR, se eliminan los errores humanos en la calificación de los formularios de respuestas.
5. El funcionamiento del sistema desarrollado, es independiente del formato del archivo de imagen del formulario escaneado.
6. Haciendo uso del sistema creado, se reducen los elevados costos de adquisición de sofisticados equipos OMR, especializados en la calificación de tarjetas de respuestas para exámenes.

VII. RECOMENDACIONES

1. Se debe utilizar preferentemente archivos de imagen con formato JPEG, debido a que son archivos de tamaño pequeño, lo cual es una ventaja en computadoras con limitaciones de espacio en disco.
2. Si se utiliza archivos con formato JPEG, no se debe utilizar tasas de compresión muy altas (menores a 30:1) al escanear los formularios, porque podrían producirse pérdidas de información en el valor de los píxeles, que afectarían el resultado de la calificación.
3. La resolución a la que se recomienda escanear los formularios, es de 96 ppp; para lograr la máxima velocidad en el procesamiento de las imágenes y la obtención de los resultados.
4. En el caso que se produzca un error de código duplicado en el procesamiento de las imágenes de los formularios, se deberá proceder a identificar al alumno a quien realmente corresponde el código, volviendo a procesar el formulario correcto de ser el caso, para evitar perjudicarlo.
5. Las personas encargadas de manejar el sistema OMR, deben tener conocimientos de informática, por lo menos de nivel medio, debido a que la terminología utilizada, el escaneo de los formularios, y el manejo de archivos y carpetas así lo requiere.
6. El sistema creado, puede ser adaptado a otros ámbitos, como aplicación para cuestionarios de encuesta, cuestionarios diversos de adquisición de

datos, etc.

7. En este trabajo se sientan las bases para investigaciones en reconocimiento de imágenes, como imágenes médicas, imágenes satelitales, etc.

VII. REFERENCIAS BIBLIOGRÁFICAS

1. GONZÁLES, Rafael C.; Richard E. WOODS. (1996) Tratamiento digital de imágenes. 1ª. ed. Editorial Díaz de Santos S.A. 800 p.
2. MUNÁRRIZ ORTIZ, Rafael. (2006) Imagen Digital. 1ª. ed. Tursen S.A. – H. Blume. 224 p.
3. HORVAT, Les.; Mark GALER. (2005) La Imagen Digital. 1ª. ed. Ediciones Anaya Multimedia. 368 p.
4. RODRÍGUEZ ALONSO, Hugo. (2005) Imagen Digital: Conceptos Básicos. 1ª. ed. Editorial Marcombo S.A. 110 p.
5. PARKER, James. (2006) Algorithms For Image Processing And Computer Vision. 1ª. ed. John Wiley & Sons 432 p.
6. BHABATOSH, Chanda. (2007) Digital Image Processing And Analysis. 1ª. ed. Editorial Prentice-hall 384 P.
7. KASTLEMAN, Kenneth. (1995) Digital Image Processing. 2º ed. Editorial Prentice-hall 667 P.
8. GÓMEZ JIMÉNEZ, Enrique. (2002) Visual Basic .NET 1º ed. Editorial Anaya Multimedia 384 p.
9. EXJEN, Bill; Jason BERES. (2002) El libro de VISUAL BASIC .NET. 1ª. ed. Ediciones Anaya Multimedia 1181 p.

10. *Encyclopedia of Graphics File Formats*. [en línea]. FileFormat.Info.

Disponible en: <http://www.fileformat.info/mirror/egff/index.htm>

[consulta: 20 noviembre 2008]

11. MEJÍA VILET, José. *Procesamiento Digital De Imágenes*. [en línea].

Disponible en: <http://66.240.239.19/4/2/4/42480.ZIP> [consulta: 10 enero

2009]

12. CCITT. Terminal Equipment and Protocols for Telematic Services.

Recommendation T.81. (1992). ISO/IEC International Standard 10918-1

ANEXO

Código del formulario MDI padre

```
Imports System
Imports System.Drawing
Imports System.Collections
Imports System.ComponentModel
Imports System.Windows.Forms
Imports System.Data

Public Class Principal

    Public Cerrar As Boolean = True

    Private Sub Principal_FormClosing(ByVal sender As Object, ByVal e As
        System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

        If Me.Text.Length > 15 Then
            If Guardar.Enabled Then
                Dim resp As System.Windows.Forms.DialogResult
                resp = MessageBox.Show( _
                    "Los datos del proceso actual se perderán. ¿Desea" & _
                    "guardar el proyecto?", "Salir del programa", _
                    MessageBoxButtons.YesNoCancel, _
                    MessageBoxIcon.Information, _
                    MessageBoxDefaultButton.Button1)

                If resp = Windows.Forms.DialogResult.Yes Then
                    If GuardarDatos() = False Then
                        e.Cancel = True
                    End If
                ElseIf resp = Windows.Forms.DialogResult.Cancel Then
                    e.Cancel = True
                End If
            End If
        End If

    End Sub

    Private Sub Principal_Load(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles MyBase.Load
        Ingresar.Enabled = False
        Formularios.Enabled = False
        Resultados.Enabled = False
        Guardar.Enabled = False
        nResp = 100
    End Sub

    Private Sub Claves_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles Claves.Click
        Dim f As New Ingresar_datos
        f.MdiParent = Me
        f.Show()
    End Sub

    Private Sub Datos_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles Datos.Click
        Dim f As New Datos_Personales
        f.MdiParent = Me
        f.Show()
    End Sub

    Private Sub Imprimir_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles Imprimir.Click
```

```

        Dim f As New Imprimir
        f.MdiParent = Me
        f.Show()
    End Sub

    Private Sub Procesar_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles Procesar.Click
        Dim f As New Procesar
        f.MdiParent = Me
        f.Show()
    End Sub

    Private Sub Salir_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles Salir.Click
        If Me.Text.Length > 15 Then
            If Guardar.Enabled Then
                Dim resp As System.Windows.Forms.DialogResult
                resp = MessageBox.Show( _
                    "Los datos del proceso actual se perderán. ¿Desea
                    guardar el proyecto?", "Salir del programa", _
                    MessageBoxButtons.YesNoCancel, _
                    MessageBoxIcon.Information, _
                    MessageBoxDefaultButton.Button1)
                If resp = Windows.Forms.DialogResult.Yes Then
                    If GuardarDatos() = False Then
                        Exit Sub
                    End If
                ElseIf resp = Windows.Forms.DialogResult.Cancel Then
                    Exit Sub
                End If
            End If
            Guardar.Enabled = False
            Me.Close()
        End Sub

    Private Sub Nuevo_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles Nuevo.Click

        If Guardar.Enabled Or Me.Text.Length > 13 Then
            Dim resp As System.Windows.Forms.DialogResult
            resp = MessageBox.Show("El proceso actual se cerrará.
                ¿Desea continuar?", "Cerrar proyecto", _
                MessageBoxButtons.YesNo, _
                MessageBoxIcon.Information, _
                MessageBoxDefaultButton.Button1)

            If resp = Windows.Forms.DialogResult.No Then
                Exit Sub
            Else
                For Each formulario As Form In Me.MdiChildren
                    formulario.Close()
                Next
            End If

        End If

        NombreProceso = "Sin nombre"
        RutaProc = Application.StartupPath & "\Procesos\"
        RutaBD = Application.StartupPath & "\Procesos\Sin nombre.mdb"

        Me.Text = "UniversalOMR - " & NombreProceso
        My.Computer.FileSystem.CopyFile(Application.StartupPath & _
            "\Recursos\DatosOMR.mdb", RutaBD, True)

        Ingresar.Enabled = True
    
```

```

Formularios.Enabled = True
Resultados.Enabled = True

Dim f As New Datos_Personales
f.MdiParent = Me
f.Show()

RutaForm = Application.StartupPath & "\Formularios Escaneados"

End Sub

Private Sub Abrir_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Abrir.Click

If Guardar.Enabled Or Me.Text.Length > 13 Then
Dim resp As System.Windows.Forms.DialogResult
resp = MessageBox.Show("El proceso actual se cerrará.
¿Desea continuar?", "Cerrar proyecto", _
MessageBoxButtons.YesNo, _
MessageBoxIcon.Information, _
MessageBoxDefaultButton.Button1)
If resp = Windows.Forms.DialogResult.No Then
Exit Sub
Else
For Each formulario As Form In Me.MdiChildren
formulario.Close()
Next
End If
End If

Dim i As Integer
Dim explorador As New OpenFileDialog
explorador.InitialDirectory = Application.StartupPath & _
"\Procesos"
explorador.Filter = "Archivos OMR (*.omr)|*.omr"
explorador.Title = "Abrir archivo de proyecto OMR"

If explorador.ShowDialog() = Windows.Forms.DialogResult.OK Then
Ingresar.Enabled = True
Formularios.Enabled = True
Resultados.Enabled = True
Guardar.Enabled = False

Try
RutaProc = ""
Dim a() As String = explorador.FileName.Split("\")
For i = 0 To a.Length - 2
RutaProc += a(i) & "\"
Next

Dim nom() As String = a(a.Length - 1).Split(".")
NombreBD = nom(0) & ".mdb"
RutaBD = RutaProc & NombreBD
Me.Text = "UniversalOMR - " & nom(0)

Dim f As New Datos_Personales
f.MdiParent = Me
f.Show()

RutaForm = Application.StartupPath & _
"\Formularios Escaneados"

Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical, "Error")
End Try

```

```

        End If

    End Sub

    Private Sub Guardar_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles Guardar.Click

        If Microsoft.VisualBasic.Right(Me.Text, Me.Text.Length - 15)
            = "Sin nombre" Then
            Dim explorador As New SaveFileDialog
            explorador.DefaultExt = ".omr"
            explorador.Filter = "Archivos OMR (*.omr)|*.omr"
            explorador.FileName = NombreProceso
            explorador.InitialDirectory = RutaProc
            explorador.OverwritePrompt = False

            If explorador.ShowDialog() = _
                Windows.Forms.DialogResult.OK Then

                Dim a() As String = explorador.FileName.Split("\")
                NombreProceso = a(a.Length - 1)
                Dim nom() As String = a(a.Length - 1).Split(".")
                NombreBD = nom(0) & ".mdb"
                Me.Text = "UniversalOMR - " & NombreProceso

                Try
                    If Not My.Computer.FileSystem.FileExists(NombreBD) Then
                        My.Computer.FileSystem.RenameFile(RutaBD, NombreBD)
                        Dim sw As New System.IO.StreamWriter(NombreProceso)
                        sw.WriteLine("Nombre: " & NombreProceso)
                        sw.Close()
                        Cerrar = True
                        Guardar.Enabled = False

                    Else
                        MessageBox.Show("Error: el archivo ya existe", _
                            "Guardar registro", _
                            MessageBoxButtons.OK, _
                            MessageBoxIcon.Warning)

                        Cerrar = False
                    End If
                Catch ex As Exception
                    MsgBox(ex.Message, MsgBoxStyle.Critical, "Error")
                End Try

            Else
                Cerrar = False
            End If
        End If

    End Sub

    Private Sub Resultados_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles Resultados.Click
        Dim f As New Resultados
        f.MdiParent = Me
        f.Show()

    End Sub

End Class

```

Código del módulo de funciones para el procesamiento de las imágenes

```
Imports System
Imports System.Drawing
Imports System.Drawing.Imaging
Imports System.Drawing.Printing
Imports System.Data
Imports System.Data.OleDb
Imports System.IO
Imports Microsoft.Office.Interop

Module Funciones
    Public RutaForm, Título As String
    Public grid As New DataGridView
    Public mensaje, códigos(), NombreBD, RutaBD, RutaProc, _
        NombreProceso As String
    Public tablas() As DataRow
    Public nResp As Integer      'Número de respuestas
    Public fEsc As Single       'Factor de escalamiento

    'Función que determina si el píxel es un punto de la marca
    Public Function Marcado(ByVal PtoRefY As Integer, _
        ByVal X As Integer, ByVal Y As Integer, _
        ByVal ángulo As Single, ByRef figura As Bitmap) As Boolean

        Dim i, j, n, b As Integer

        'Rotación de ejes
        X = Math.Round((X * Math.Cos(ángulo) - (PtoRefY - Y) * _
            Math.Sin(ángulo)))

        Y = Math.Round((Y * Math.Cos(ángulo) - X * _
            Math.Sin(ángulo)))

        n = 0      'Píxeles negros
        b = 0      'Píxeles blancos
        i = 0      'Columna que se explorará
        j = 0      'Fila que se explorará

        Do While i < 5 * fEsc And n < 5 * fEsc And b < 5 * fEsc
            For j = 0 To 5 * fEsc
                If figura.GetPixel(X + i, Y - j).R < 150 Then
                    n += 1
                Else
                    b += 1
                End If
            Next
            i += 1
        Loop

        If n < 5 * fEsc Then
            Return False
        Else
            Return True
        End If

    End Function

    'Función que procesa el formulario
    Public Function ProcesarForm(ByVal fig As Bitmap, _
        ByRef Cod As String, ByRef Resp() As Integer) As Boolean

        Dim i, j, k As Integer
```

```

Dim puntoizq, puntoder, PuntosV(35), PuntosH(20) As Point
Dim figura As New Bitmap(fig)
Dim ang, pendiente As Single
Dim r As Boolean = True

'Halla los puntos de referencia para calcular el ángulo
puntoizq = PtoRefIzq(figura, figura.Width \ 5, figura.Height \ 10)
puntoder = PtoRefDer(figura, figura.Width \ 10, figura.Height \ 10)

'Calcula la pendiente y el ángulo
pendiente = (puntoizq.Y - puntoder.Y) / (puntoder.X - puntoizq.X)
ang = Math.Atan(pendiente)

'Hallar la marca de inicio izquierda

If ang < 0.0015 Then
    puntoizq = PtoRefIzq(figura, figura.Width \ 5, figura.Height \ 10)
End If

If ang >= 0.0015 Then
    puntoizq = PtoRefIzq(figura, figura.Width \ 5, figura.Height \ 10)
End If

mensaje = ""

Try

    'Se aplica translación de ejes, escalamiento y rotación
    'de 180° del eje Y, en los cálculos siguientes

    'Calcular la posición de los puntos de referencia horizontales

    k = puntoizq.X + Math.Round(9 * fEsc)
    For i = 0 To 3
        For j = 0 To 5
            PuntosH(i * 5 + j).X = k
            PuntosH(i * 5 + j).Y = puntoizq.Y
            k += Math.Round(25 * fEsc)
        Next
        k += Math.Round(13 * fEsc)
    Next

    'Calcular la posición de los puntos de referencia verticales

    k = puntoizq.Y - Math.Round(6 * fEsc)

    For i = 0 To 24
        k -= Math.Round(23 * fEsc)
        PuntosV(i).X = puntoizq.X
        PuntosV(i).Y = k
    Next

    'Calcular los puntos de referencia del código

    k -= Math.Round(48 * fEsc)

    For i = 25 To 34
        k -= Math.Round(23 * fEsc)
        PuntosV(i).X = puntoizq.X
        PuntosV(i).Y = k
    Next

    '***** Hallar el código del alumno *****

```

```

For i = 0 To 3
    For j = 25 To 34

        If Marcado(puntoizq.Y, PuntosH(i).X, _
            PuntosV(j).Y, ang, figura) Then

            Cod = Cod & (34 - j).ToString

        End If

    Next
Next

'Verificar errores en el código
If Cod.Length <> 4 Then
    r = False
    mensaje = "Error en el código " & Cod & "- Archivo "
ElseIf Not existe(Cod) Then
    r = False
    mensaje = "Error: el código no existe " & Cod _
        & "- Archivo "
ElseIf duplicado(Cod) Then
    r = False
    mensaje = "Código duplicado " & Cod & "- Archivo "
Else
    ReDim Preserve códigos(códigos.Length - 1)
    códigos(códigos.Length - 1) = Cod

    '***** Hallar las respuestas *****
    'Para los 4 bloques de respuestas
    For i = 0 To 3
        'Para las 25 filas de respuestas
        For j = 24 To 0 Step -1
            'Para las 5 columnas de respuestas por bloque
            For k = 0 To 4
                If Marcado(puntoizq.Y, _
                    PuntosH(k + 5 * i).X, _
                    PuntosV(j).Y, ang, figura) Then
                    Resp((24 - j) + 25 * i) += 5 + k + 1
                End If
            Next
        Next
    Next
End If

Catch ex As Exception
    r = False
    mensaje = "Error al procesar archivo "
End Try

Return r

End Function

'Función que verifica si el código es duplicado
Public Function duplicado(ByVal cod As String) As Boolean

    Dim i As Integer
    Dim r As Boolean = False
    For i = 0 To códigos.Length - 1
        If códigos(i) = cod Then
            r = True
            Exit For
        End If
    Next
Next

```

```

        Return r
    End Function

'Función que determina el puntaje
Public Function Puntaje(ByVal resp As Integer(), _
    ByVal datos As DataSet) As Integer

    Dim i As Integer

    'Contador de aciertos
    Dim p As Integer = 0

    For i = 0 To 99
        If resp(i) - 5 = Cint(datos.Tables(0).Rows(i).Item(3)) Then
            p += 1
        End If
    Next

    Return p
End Function

'Función que verifica si el código existe
Public Function existe(ByVal cod As String) As Boolean
    Dim i As Integer
    Dim r As Boolean = False

    Dim cn As New OleDbConnection( _
        "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & RutaBD)
    Dim adaptadorDatos As New OleDbDataAdapter( _
        "SELECT * FROM Datos", cn)
    Dim Datos As New DataSet

    Datos.Clear()
    adaptadorDatos.Fill(Datos)

    For i = 0 To Datos.Tables(0).Rows.Count - 1
        If Datos.Tables(0).Rows.Item(i).Item(1) = cod Then
            r = True
            Exit For
        End If
    Next

    Return r
End Function

'Función que determina la marca de posición izquierda
Public Function PtoRefIzq(ByVal imagen As Bitmap, _
    ByVal ancho As Integer, ByVal alto As Integer) As Point

    'ancho = ancho del rectángulo explorado para hallar
    'la marca de referencia
    'alto = alto del rectángulo explorado para hallar
    'la marca de referencia
    Dim i, j As Integer
    Dim punto As Point = New Point()
    Dim figura As New Bitmap(imagen)

    j = imagen.Height - 1

    'Mientras j en el intervalo del alto
    Do While j >= imagen.Height - 1 - alto

```

```

        'Para i en el intervalo del ancho
        For i = 0 To ancho
            'Si el pixel es negro
            If imagen.GetPixel(i, j).R < 5 Then
                Exit Do 'Salir del bucle While
            End If
        Next
        j -= 1
    Loop

    'Asignar las coordenadas del punto
    punto.X = i
    punto.Y = j

    Return punto
End Function

'Función que determina la marca de posición derecha
Public Function PtoRefDer(ByVal imagen As Bitmap, _
    ByVal ancho As Integer, ByVal alto As Integer) As Point

    Dim i, j As Integer
    Dim punto As Point = New Point()

    j = imagen.Height - 1

    'Mientras j en el intervalo del alto
    Do While j >= imagen.Height - 1 - alto
        'Para i en el intervalo del ancho
        For i = imagen.Width - 1 To imagen.Width - ancho Step -1
            'Si el pixel es negro
            If imagen.GetPixel(i, j).R < 5 Then
                Exit Do 'Salir del bucle While
            End If
        Next
        j -= 1
    Loop

    'Asignar las coordenadas del punto
    punto.X = i
    punto.Y = j

    Return punto
End Function

'Procedimiento que actualiza el ListBox
Public Sub Actualizar(ByVal lista As ListBox, _
    ByVal txt As TextBox, ByVal nombre As String, _
    ByVal numero As Integer)
    lista.Items.Add(nombre)
    txt.Text = numero.ToString
End Sub

'Procedimiento que muestra los resultados del proceso
Public Sub MostrarResultados(ByVal esp As String, _
    ByVal n As Integer, ByVal lista As DataGridView)

    Dim cn As New _
        OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;" & _
            "Data Source=" & RutaBD)

    Dim adaptadorDatos As New _
        OleDbDataAdapter("SELECT * FROM Datos " & _
            "ORDER BY Puntaje DESC", cn)

```

```

Dim Datos As New DataSet

If esp = "Todas" And n = 0 Then
    Datos.Clear()
    adaptadorDatos.Fill(Datos)
    lista.DataSource = Datos
    lista.DataMember = "Table"

ElseIf esp = "Todas las especialidades" And n > 0 Then
    adaptadorDatos.SelectCommand.CommandText = "SELECT TOP " & _
        & n & " * FROM datos ORDER BY Puntaje DESC"
    Datos.Clear()
    adaptadorDatos.Fill(Datos)
    lista.DataSource = Datos
    lista.DataMember = "Table"

ElseIf esp <> "Todas" And n = 0 Then
    adaptadorDatos.SelectCommand.CommandText = _
        "SELECT * FROM datos WHERE Especialidad='" & _
        esp & "' ORDER BY Puntaje DESC"
    Datos.Clear()
    adaptadorDatos.Fill(Datos)
    lista.DataSource = Datos
    lista.DataMember = "Table"

ElseIf esp <> "Todas las especialidades" And n > 0 Then
    adaptadorDatos.SelectCommand.CommandText = _
        "SELECT TOP " & n & _
        " * FROM datos WHERE Especialidad='" & _
        & esp & "' & "ORDER BY Puntaje DESC"
    Datos.Clear()
    adaptadorDatos.Fill(Datos)
    lista.DataSource = Datos
    lista.DataMember = "Table"
End If

End Sub

'Procedimiento que imprime los resultados
Public Sub Imprime(ByVal previsualizar As Boolean)

    Dim ImpDoc As New PrintDocument

    AddHandler ImpDoc.PrintPage, AddressOf PáginaImprimir

    If previsualizar Then
        Dim PrevDoc As New PrintPreviewDialog
        PrevDoc.Text = "Imprimir documento"
        PrevDoc.Document = ImpDoc
        PrevDoc.ShowDialog()
    Else
        Dim DialogImp As New PrintDialog
        If DialogImp.ShowDialog = Windows.Forms.DialogResult.Yes Then
            ImpDoc.Print()
        End If
    End If
End Sub

'Procedimiento que dá formato a la página a imprimirse
Private Sub PáginaImprimir(ByVal sender As Object, _
    ByVal e As PrintPageEventArgs)
    Dim xPos As Single = e.MarginBounds.Left
    Dim titFont As New Font("Arial", 16, FontStyle.Bold)
    Dim yPos As Single = titFont.GetHeight(e.Graphics)

```

```

e.Graphics.DrawString(Título, titFont, Brushes.Black, _
                        xPos, yPos)
yPos += 40
Dim numPos, codPos, apPos, amPos, nomPos, espPos, punPos As Integer
numPos = 30
codPos = 60
apPos = 120
amPos = 250
nomPos = 380
espPos = 500
punPos = 720

Dim EncFont As New Font("Arial", 10, FontStyle.Bold)
e.Graphics.DrawString("N°", EncFont, Brushes.Black, _
                        numPos, yPos)
e.Graphics.DrawString("Código", EncFont, Brushes.Black, _
                        codPos, yPos)
e.Graphics.DrawString("Apellido paterno", EncFont, _
                        Brushes.Black, apPos, yPos)
e.Graphics.DrawString("Apellido materno".ToString, EncFont, _
                        Brushes.Black, amPos, yPos)
e.Graphics.DrawString("Nombres", EncFont, Brushes.Black, _
                        nomPos, yPos)
e.Graphics.DrawString("Especialidad", EncFont, _
                        Brushes.Black, espPos, yPos)
e.Graphics.DrawString("Puntaje", EncFont, Brushes.Black, _
                        punPos, yPos)
yPos += 15

e.Graphics.DrawLine(Pens.Black, 30, yPos, 775, yPos)

yPos += 5

Dim txtFont As New Font("Arial", 10, FontStyle.Regular)
Dim fila As Integer = 0
While fila < grid.Rows.Count
    e.Graphics.DrawString((fila + 1).ToString, txtFont, _
                           Brushes.Black, numPos, yPos)
    e.Graphics.DrawString(
        grid.Rows(fila).Cells(1).Value.ToString, _
        txtFont, Brushes.Black, codPos + 5, yPos)
    e.Graphics.DrawString(
        grid.Rows(fila).Cells(2).Value.ToString, _
        txtFont, Brushes.Black, apPos, yPos)
    e.Graphics.DrawString(
        grid.Rows(fila).Cells(3).Value.ToString, _
        txtFont, Brushes.Black, amPos, yPos)
    e.Graphics.DrawString(
        grid.Rows(fila).Cells(4).Value.ToString, _
        txtFont, Brushes.Black, nomPos, yPos)
    e.Graphics.DrawString(
        grid.Rows(fila).Cells(5).Value.ToString, _
        txtFont, Brushes.Black, espPos, yPos)
    e.Graphics.DrawString(
        grid.Rows(fila).Cells(6).Value.ToString, _
        txtFont, Brushes.Black, punPos + 20, yPos)
    fila += 1
    yPos += 15
End While

e.HasMorePages = False

End Sub

'Definición de la clase para exportar a Excel

```

```

Public Class ExportExcel
    Public Sub DGVaExcel(ByVal dgvResultados As DataGridView)
        Dim ArchTemp As String = Path.GetTempFileName()
        FileOpen(1, ArchTemp, OpenMode.Output)
        Dim TextoLínea As String = ""
        Dim columna As DataGridViewTextBoxColumn
        For Each columna In dgvResultados.Columns
            TextoLínea &= columna.HeaderText & _
                Microsoft.VisualBasic.ControlChars.Tab
        Next
        PrintLine(1, TextoLínea)
        Dim i As Integer = 0
        Dim filas As DataGridViewRow
        For Each filas In dgvResultados.Rows
            i = 0
            TextoLínea = ""
            For Each columna In dgvResultados.Columns
                If Not IsDBNull(filas.Cells(i).Value) Then
                    TextoLínea &= CStr(filas.Cells(i).Value) & _
                        Microsoft.VisualBasic.ControlChars.Tab
                Else
                    TextoLínea &= _
                        Microsoft.VisualBasic.ControlChars.Tab
                End If
                i += 1
            Next
            PrintLine(1, TextoLínea)
        Next
        FileClose(1)
        ExpTextExcel(ArchTemp)
    End Sub

    Public Sub ExpTextExcel(ByVal NombreArchivo As String)
        Dim vFormato As Excel.XlRangeAutoFormat
        Dim vCultura As System.Globalization.CultureInfo = _
            System.Threading.Thread.CurrentThread.CurrentCulture
        System.Threading.Thread.CurrentThread.CurrentCulture = _
            System.Globalization.CultureInfo. _
                CreateSpecificCulture("en-US")
        Dim Exc As Excel.Application = New Excel.Application
        Exc.Workbooks.OpenText(NombreArchivo, , , _
            Excel.XlTextQualifier.xlTextQualifierNone, , True)
        Dim Wb As Excel.Workbook = Exc.ActiveWorkbook
        Dim Ws As Excel.Worksheet = Wb.ActiveSheet
        'Se le indica el formato al que queremos exportarlo
        Dim valor As Integer = 10
        If valor > -1 Then
            Select Case valor
                Case 10 : vFormato = _
                    Excel.XlRangeAutoFormat.xlRangeAutoFormatClassic1
            End Select
            Ws.Range(Ws.Cells(1, 1), Ws.Cells(Ws.UsedRange.Rows.Count, _
                Ws.UsedRange.Columns.Count)).AutoFormat(vFormato)
            NombreArchivo = Path.GetTempFileName().Replace("tmp", "xls")
            File.Delete(NombreArchivo)
            Exc.ActiveWorkbook.SaveAs(NombreArchivo, _
                Excel.XlTextQualifier.xlTextQualifierNone - 1)
        End If
        Exc.Quit()
        Ws = Nothing
        Wb = Nothing
        Exc = Nothing
        GC.Collect()
        If valor > -1 Then
            Dim p As System.Diagnostics.Process = _

```

```

        New System.Diagnostics.Process
        p.EnableRaisingEvents = False
        System.Diagnostics.Process.Start(NombreArchivo)
    End If
    System.Threading.Thread.CurrentThread.CurrentCulture = vCultura
End Sub

End Class

'Procedimiento que habilita el botón guardar
Public Sub HabilitarBotón(ByVal datos As DataSet)
    If Principal.Text.Length > 15 And datos.Tables(0).Rows.Count > 0 _
        And Trim(Microsoft.VisualBasic.Right(Principal.Text, _
        Principal.Text.Length - 15)) = "Sin nombre" Then
        Principal.Guardar.Enabled = True
    Else
        Principal.Guardar.Enabled = False
    End If
End Sub

'Función que almacena los cambios realizados en los datos
Public Function GuardarDatos() As Boolean
    Dim explorador As New SaveFileDialog
    explorador.DefaultExt = ".omr"
    explorador.Filter = "Archivos OMR (*.omr)|*.omr"
    explorador.FileName = NombreProceso
    explorador.InitialDirectory = RutaProc
    explorador.OverwritePrompt = False

    If explorador.ShowDialog() = Windows.Forms.DialogResult.OK Then
        Dim a() As String = explorador.FileName.Split("\")
        NombreProceso = a(a.Length - 1)
        Dim nom() As String = a(a.Length - 1).Split(".")
        NombreBD = nom(0) & ".mdb"
        Principal.Text = "UniversalOMR - " & NombreProceso
        Try
            If Not My.Computer.FileSystem.FileExists(NombreBD) Then
                My.Computer.FileSystem.RenameFile(RutaBD, NombreBD)
                Dim sw As New System.IO.StreamWriter(NombreProceso)
                sw.WriteLine("Nombre: " & NombreProceso)
                sw.Close()
                Return True
            Else
                MessageBox.Show("Error: el archivo ya existe", _
                    "Guardar registro", _
                    MessageBoxButtons.OK, _
                    MessageBoxIcon.Warning)
                Return False
            End If
        Catch ex As Exception
            MsgBox(ex.Message, MsgBoxStyle.Critical, "Error")
        End Try
    Else
        Return False
    End If
End Function

End Module

```