

CENTRO NACIONAL DE METROLOGÍA

ÁREA DE METROLOGÍA FÍSICA

Dirección de óptica y radiometría



DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE APLICACIÓN CON RECONOCIMIENTO ÓPTICO DE CARACTERES EN IMÁGENES DIGITALES PARA LA TOMA DE LA LECTURA DE LAS MEDICIONES Y REGISTRO DE LOS VALORES DE CALIBRACIÓN DE LA ESCALA DE ATENUACIÓN EN DECIBELES DE LOS REFLECTÓMETROS ÓPTICOS DE DOMINIO EN EL TIEMPO.

juan miguel ORTIZ MELÉNDEZ.
mortiz@cenam.mx

El Marqués, Querétaro, México.
Agosto, 2023.

DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE APLICACIÓN CON RECONOCIMIENTO ÓPTICO DE CARACTERES EN IMÁGENES DIGITALES PARA LA TOMA DE LA LECTURA DE LAS MEDICIONES Y REGISTRO DE LOS VALORES DE CALIBRACIÓN DE LA ESCALA DE ATENUACIÓN EN DECIBELES DE LOS REFLECTÓMETROS ÓPTICOS DE DOMINIO EN EL TIEMPO.

SÍNTESIS

Actualmente, en el laboratorio de fibras ópticas del Centro Nacional de Metrología se cuenta con el servicio de calibración de reflectómetros ópticos de dominio en el tiempo – OTDR *por las siglas en inglés de: optical time domain reflectometer* [1], en este servicio de calibración se caracteriza la escala de atenuación del instrumento, esta caracterización consta de un conjunto de mediciones en atenuación que se llevan a cabo en el laboratorio; estas mediciones son realizadas desde luego en el instrumento del cliente de acuerdo a como se muestra a continuación, ver figura 1.

1 EL REFLECTÓMETRO ÓPTICO DE DOMINIO EN EL TIEMPO DEL CLIENTE REALIZA LAS MEDICIONES DE CALIBRACIÓN

2 SE TOMAN LAS LECTURAS DE ESTAS DESDE EL OTDR EN EL FORMATO DE DATOS PARA EL OTDR

Datos del OTDR					
Marca / Modelo:	LADO: A 1 B 2	L: 1 310 mm 1 820 mm 1 820 mm	Distancia A-B: 100 m 50 m 50 m		
No. Serie:	Clipping Level:	26.95 dB	Fecha: 17/04/2019		
Hora de arranque del equipo:	01:45	Hora de inicio de la calibración:	10:15	Hora de fin de la calibración:	11:19
Hora de inicio de la estabilización:	01:45	Temperatura ambiental:	23.49°C	Temperatura Fibra:	22.59°C
Hora de fin de la Estabilización:	10:15	Humedad inicial:	57.54%	Humedad Final:	56.83%
Nivel Atenuación (dB)		Pot. Potencia		Atenuación total	
0	22.72	21.24	1.48	0.318	
0.5	21.63	20.15	1.48	0.335	
1.0	21.19	19.71	1.48	0.348	
1.5	20.68	19.27	1.41	0.334	
2.0	20.17	18.76	1.41	0.344	
2.5	19.67	18.26	1.41	0.335	
3.0	19.18	17.76	1.42	0.342	
3.5	18.68	17.26	1.42	0.318	
4.0	18.19	16.77	1.42	0.342	
4.5	17.69	16.27	1.42	0.334	
5.0	17.19	15.77	1.42	0.342	
5.5	16.69	15.27	1.42	0.334	
6.0	16.19	14.77	1.42	0.343	
6.5	15.69	14.27	1.42	0.343	
7.0	15.19	13.77	1.42	0.343	
7.5	14.69	13.27	1.42	0.343	
8.0	14.19	12.77	1.42	0.343	
8.5	13.69	12.27	1.42	0.343	
9.0	13.19	11.77	1.42	0.343	
9.5	12.69	11.27	1.42	0.343	
10.0	12.19	10.77	1.42	0.343	

3 EL USUARIO REALIZA EL REGISTRO DE LAS MEDICIONES EN LA HOJA DE CÁLCULO EN LA COMPUTADORA PARA ESTE FIN

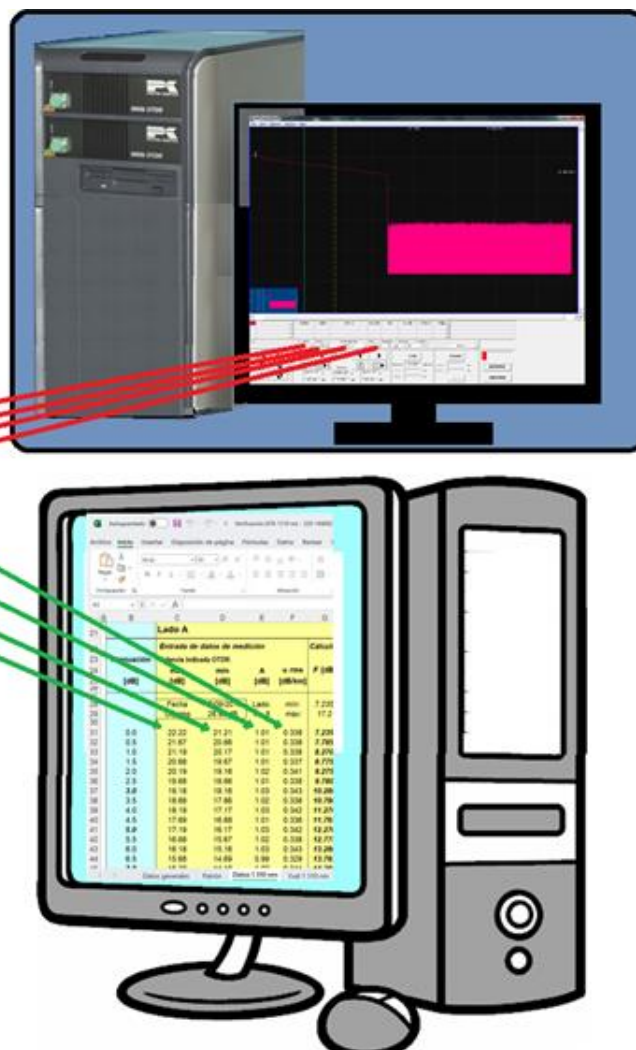


Figura 1. Calibración en atenuación del OTDR, toma de lecturas y registro de los resultados de las mediciones.

Valga la pena mencionar que este proceso es llevado a cabo en secuencias repetidas hasta el punto de completar el llenado de al menos 16 formatos de datos de registro del OTDR como el que se muestra en la parte izquierda de la figura anterior.

Se trata entonces, de una necesidad planteada con el fin de automatizar el proceso *manual* de la toma de las lecturas de los resultados, así como también el registro de los valores de la calibración en sus hojas de cálculo como acabamos de ver.

Introducimos aquí una herramienta que nos ayude a resolver esta situación:

El reconocimiento óptico de caracteres, el cual es un proceso de digitalización de textos a partir de imágenes de símbolos o caracteres que pertenecen a un determinado alfabeto. Gracias a ello se pueden identificar y almacenar los datos reconocidos a partir de las imágenes y así poder interactuar con estos caracteres. El presente proyecto está enfocado a este fin: obtener, a partir de las imágenes en pantalla de una computadora, las lecturas de los valores de los resultados de las mediciones de la calibración de la escala de atenuación en decibeles de los *reflectómetros ópticos de dominio en el tiempo* y elaborar un compilado con estos que nos permitan también realizar su registro dentro de las hojas de cálculo correspondientes.

Se presenta el diseño, desarrollo e implementación de una aplicación en software y bibliotecas orientadas al reconocimiento óptico de caracteres – OCR *por las siglas en inglés de: Optical Character Recognition* – [2] con el fin mencionado.

ÍNDICE

SÍNTESIS	2
1. INTRODUCCIÓN	5
1.1 Motivación del proyecto.....	5
1.2 Objetivo del proyecto.....	6
1.3 Organización del documento.....	6
2. MARCO TEÓRICO	6
2.1 Conceptos teóricos.....	6
2.2 El reflectómetro óptico de dominio en el tiempo – OTDR.....	6
2.3 Imágenes digitales.....	8
2.4 El histograma.....	10
2.5 Binarización de las imágenes.....	12
2.5.1 Binarización mediante el método de Otsu.....	12
2.5.2 Binarización mediante la media.....	13
2.6 Fase de análisis.....	13
2.7 Fase de diseño.....	16
2.7.1 Vista Conceptual.....	16
2.7.2 Vista Lógica.....	17
2.7.3 Arquitectura Modelo Vista Controlador.....	18
2.7.4 Diseño del módulo.....	20
2.8 Fase de Implementación.....	20
2.8.1 Descripción.....	20
2.8.2 Lenguaje de Programación: Labview 2010.....	20
2.9 Fase de pruebas.....	26
2.10 Aplicación desarrollada.....	26
3. CONCLUSIONES	31
4. REFERENCIAS BIBLIOGRÁFICAS, JURÍDICAS Y/O HEMEROGRÁFICAS	31

DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE APLICACIÓN CON RECONOCIMIENTO ÓPTICO DE CARACTERES EN IMÁGENES DIGITALES PARA LA TOMA DE LA LECTURA DE LAS MEDICIONES Y REGISTRO DE LOS VALORES DE CALIBRACIÓN DE LA ESCALA DE ATENUACIÓN EN DECIBELES DE LOS REFLECTÓMETROS ÓPTICOS DE DOMINIO EN EL TIEMPO.

1. INTRODUCCIÓN.

1.1 Motivación del proyecto.

El reconocimiento óptico de caracteres es esencial en el tratamiento de imágenes para la identificación de secuencias de caracteres con un significado específico. Este es el caso que se plantea en el marco del presente proyecto para dar solución a un problema concreto presente en la realización de los servicios de calibración de los reflectómetros ópticos de dominio en el tiempo que se dan en el laboratorio de fibras ópticas del Centro Nacional de Metrología. Se pretende automatizar el proceso de lectura de las mediciones y registro de sus valores de calibración mediante la captura de las imágenes de la pantalla del reflectómetro con el fin de identificar los caracteres correspondientes a las lecturas de las mediciones y demás información relativa a la configuración de este durante el proceso de su calibración.

Una imagen es capaz de transmitir de manera precisa lo ocurrido en un instante determinado. Es tan importante que se puede aplicar tanto a aspectos científicos como personales. Desde la aparición de los algoritmos de OCR, han sido muchos los servicios que han introducido estos procesos para aumentar su rendimiento y otros que se basan completamente en estas tecnologías. Por ejemplo, el reconocimiento de texto manuscrito, reconocimiento de matrículas, indexación en bases de datos, reconocimiento de datos estructurados con OCR Zonal, son sólo algunos ejemplos de aplicación de estas tecnologías.

Además, de la motivación explícita del trabajo expresada previamente, cabe mencionar que el trabajo desarrollado ha contribuido, a través del procesamiento digital de imágenes, a conocer las técnicas de tratamiento de imágenes con el objetivo de mejorar la calidad y facilitar así la búsqueda de información en ellas, ampliando así el campo para futuras aplicaciones del software informáticas. En definitiva, este proyecto está enfocado con el fin de aplicar algoritmos y técnicas de tratamiento de imágenes, para ser capaces de reconocer caracteres de interés en aplicaciones de software de control de reflectómetros ópticos de dominio en el tiempo. Para ello, se ha desarrollado la aplicación que aquí se presenta bajo el lenguaje de programación de Labview.

El presente trabajo es de carácter práctico, y el tema en el que se encuentra abarca principalmente los dos siguientes:

Objetivos prioritarios del Programa Nacional de Combate a la Corrupción y a la Impunidad, y de Mejora de la Gestión Pública 2019-2024:

- Promover la eficiencia y eficacia de la gestión pública.
- Promover la profesionalización y la gestión eficiente de los recursos humanos de la Administración Pública Federal.
- Promover el uso eficiente y responsable de los bienes del Estado Mexicano.

Programa Institucional del Centro Nacional de Metrología 2020-2024:

- Productividad y competitividad industrial e innovación.

Actualmente se encuentra implementado y en completo funcionamiento en el laboratorio de Fibras ópticas del Centro Nacional de Metrología desde junio del 2022. Cabe mencionar que este no corresponde ni es parte de alguna tesis para obtener algún grado académico, de instructivos y/o manuales.

1.2 Objetivo del proyecto.

Como se ha indicado, el objetivo del proyecto consiste en el desarrollo de una aplicación o software informático para el reconocimiento automático de secuencias de caracteres a partir de imágenes digitales de la pantalla de la computadora de aplicaciones de software de control de reflectómetros ópticos de dominio en el tiempo. Se trata de capturar una imagen e identificar los caracteres o cadenas de caracteres de interés en la misma, las cuales son las lecturas de las mediciones y demás información relativa a la configuración del reflectómetro óptico durante su proceso de calibración.

1.3 Organización del documento.

Tras el resumen previo, junto con la motivación y los objetivos del proyecto expuestos, se describe a continuación la organización del documento.

En el capítulo 2 se analizan los conceptos teóricos en los que está basada la aplicación construida. Se detalla el diseño y la arquitectura general de esta. Se detallan las tecnologías utilizadas para desarrollar el proyecto, así como una explicación del por qué se han elegido. Se presenta un resumen paso a paso del uso de la aplicación. Se explica el proceso seguido para desarrollar el proyecto. Y en el capítulo 3 se detallan los resultados obtenidos para diferentes imágenes, así como las conclusiones extraídas.

2. MARCO TEÓRICO.

2.1 Conceptos teóricos.

En este capítulo se introducen algunos de los conceptos teóricos y términos frecuentemente utilizados en el ámbito de la aplicación. Al ser un proyecto en el que se están tratando imágenes digitales, se explica dicho concepto, cómo se representan las imágenes y cuál es su contenido, así como la descripción de histograma, que constituye un elemento parte de los métodos de procesamiento que se incluyen en la aplicación. Además, se ilustran los distintos tipos de algoritmos para la extracción de caracteres, finalizando con el concepto de binarización de imágenes entre otros.

2.2 El reflectómetro óptico de dominio en el tiempo – OTDR.

En telecomunicaciones, un reflectómetro óptico de dominio en el tiempo – OTDR *por las siglas en inglés de: optical time domain reflectometer*, es un instrumento óptico-electrónico usado para caracterizar y diagnosticar una red de fibra óptica [1]. Un OTDR puede ser utilizado para estimar la longitud de la fibra y su atenuación, lo que incluye pérdidas por empalmes y conectores, es el instrumento de campo más importante para el control y supervisión de enlaces de fibra óptica. También puede ser utilizado para detectar fallos, tales como roturas de la fibra. Para realizar su función, el OTDR inyecta en un extremo de la fibra bajo análisis una serie de pulsos ópticos y observa la luz que ha sido dispersada y reflejada de vuelta desde puntos de la fibra con un cambio en el índice de refracción. La intensidad del pulso devuelto es integrada como una función del tiempo, y representada en función de la longitud de la fibra. En la figura 2, se muestra el esquema de la tecnología interna de funcionamiento del OTDR. Y en la figura 3, podemos observar el uso del OTDR dentro de una red de distribución de fibra óptica en donde se presentan los eventos de localización de los diferentes componentes de esta, desde el inicio del enlace en el lanzamiento de la señal desde la central de distribución hasta la recepción en un hogar.

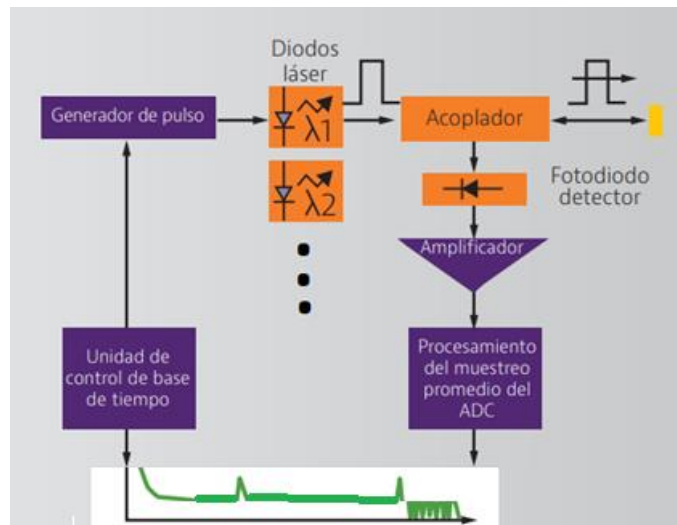


Figura 2. Estructura de la tecnología del reflectómetro óptico de dominio en el tiempo.

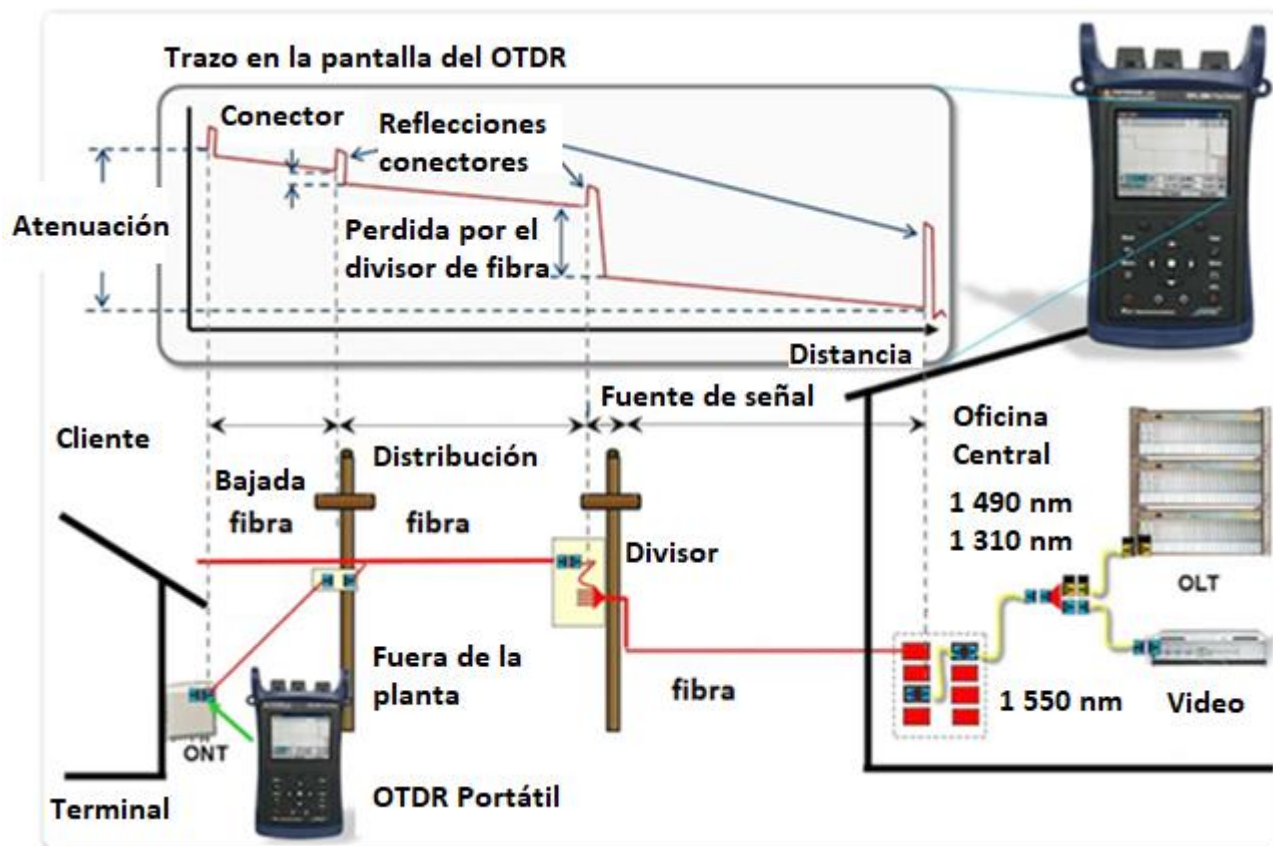


Figura 3. El OTDR en una red de distribución de fibra óptica.

2.3 Imágenes digitales.

Las imágenes digitales son capturadas mediante los dispositivos apropiados para su almacenamiento en una computadora. La figura 4 representa un proceso de captura de una escena tridimensional. La escena del mundo 3D se proyecta, a través del sistema óptico, sobre el elemento sensor en la figura un CCD – Del inglés: *Charge Coupled Device*, que bien podría ser otro de distinta naturaleza tal como los CMOS – Del inglés: *Complementary metal-oxide-semiconductor*. En cualquier caso, de forma general, estos dispositivos poseen una distribución en forma de matriz de elementos denominados píxeles – Del inglés: *picture elements*. Cada elemento se activa con la incidencia de la radiación espectral, generando un valor de tensión eléctrica a la salida, de forma que, mediante la circuitería apropiada, dicho valor se convierte a un valor numérico. Todos estos valores, organizados igualmente en forma de matriz se transfieren a la computadora, constituyendo lo que se conoce como una imagen digital, quedando en este momento disponible para su posterior procesamiento [3].

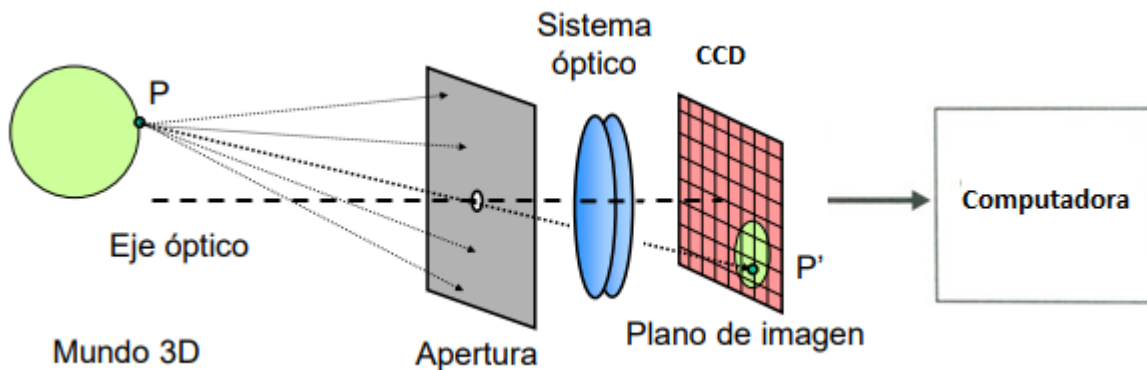


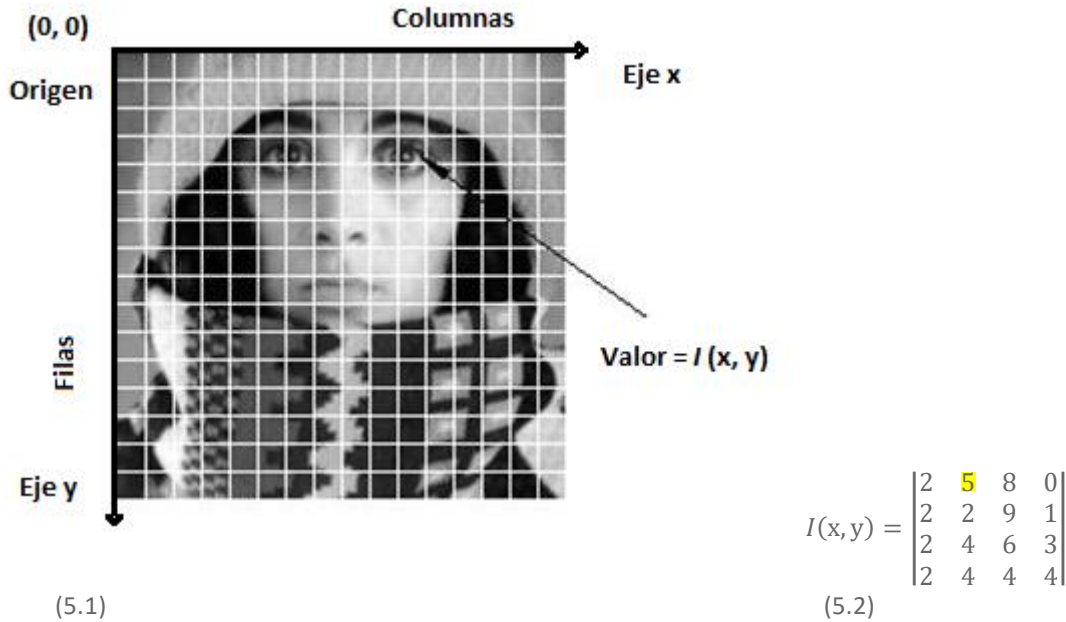
Figura 4. Captura de una imagen 3D por un dispositivo CCD.

El proceso finaliza con el almacenamiento de la imagen en la computadora. Este almacenamiento se lleva a cabo a través de algún formato característico de imágenes – TIFF, BMP, JPEG, etc.

Vemos entonces que la imagen se representa desde el punto de vista de su tratamiento computacional como una matriz numérica de dimensión $M \times N$, es decir, con M filas y N columnas. El contenido de esa matriz se refiere a valores enteros situados en las localizaciones espaciales (x, y) o píxeles, dicho valor es el resultado de la cuantificación de intensidad o nivel de gris.

Si la imagen es en blanco y negro, se almacena un valor por cada píxel. Este valor es el nivel de intensidad o nivel de gris ya mencionado. Se suele utilizar un rango de valores para su representación, que generalmente es de 0 a $2^n - 1$. Uno de los valores más utilizados es con n igual a 8 ; esto significa que el rango de valores para este caso varía entre 0 y 255 . En este caso, el 0 representa el negro absoluto y el 255 , el blanco absoluto. Esto indica que podemos tener una resolución o precisión en la escala de los grises de 256 . El hecho de utilizar 256 niveles es porque con 8 bits de la computadora se pueden codificar 256 valores distintos desde la combinación 00000000 , que representa el nivel 0 , hasta la combinación 11111111 , que representa el nivel 255 .

Sobre cada matriz se establece un sistema de coordenadas con origen normalmente en la esquina superior izquierda y con los ejes “ xy ” tales que su orientación es positiva hacia la derecha en el caso del eje x , y hacia abajo en el caso del eje y , figura 5.



(5.1)

Figura 5. Representación de una imagen digital en grises.

Con esta disposición de imágenes se tiene acceso a los valores de los píxeles en sus diferentes localizaciones, siendo el convenio de acceso el que se expresa a continuación, con relación a la figura 5.2, ecuación 1:

$$I(x, y) = I(2, 1) = 5 \quad \text{Ecuación 1.}$$

Cuando la imagen es en color, para cada localización espacial existen tres valores de intensidad asociados, es decir, los elementos de la matriz vienen dados por tres valores que representan cada uno de los componentes básicos del color en cuestión. Estos componentes son el rojo (R), verde (G), y azul (B) y es lo que se conoce como código RGB. En este caso el conjunto de valores $(0, 0, 0)$ representa al negro, mientras que los valores $(255, 255, 255)$ son el color blanco absoluto. La combinación de distintos valores representa otros colores. Debido a lo anterior, una imagen en color posee tres bandas espectrales: rojo, verde, azul; cada una de ellas viene representada por una matriz de números con valores en el rango 0 a 255 para imágenes de 8 bits. Una imagen de color vendría dada por las tres imágenes componentes o canales como en la figura 6.

$$R(x,y) = \begin{pmatrix} 2 & 5 & 8 & 0 \\ 2 & 2 & 9 & 1 \\ 2 & 4 & 6 & 3 \\ 2 & 4 & 4 & 4 \end{pmatrix}; \quad G(x,y) = \begin{pmatrix} 8 & 4 & 4 & 3 \\ 8 & 3 & 4 & 0 \\ 6 & 9 & 1 & 6 \\ 6 & 9 & 3 & 5 \end{pmatrix}; \quad B(x,y) = \begin{pmatrix} 7 & 6 & 7 & 3 \\ 7 & 6 & 8 & 4 \\ 1 & 9 & 7 & 2 \\ 3 & 9 & 9 & 6 \end{pmatrix}$$

Figura 6. Representación de una imagen digital en color: Rojo(R), Verde (G) y Azul (B).

El píxel que se encuentra en la localización espacial (x, y) posee componentes (R, G, B) . Por ejemplo, dada la imagen de la figura 6 los valores del píxel en $(x, y) = (2, 3)$ serían los siguientes: $(4, 9, 9)$. Los tres valores de cada píxel para las imágenes en color corresponden al modelo de color RGB.

2.4 El histograma.

El histograma de una imagen es una función discreta que representa el número de píxeles en la imagen en función de los niveles de intensidad, g . La probabilidad de ocurrencia de un determinado nivel se define como en la ecuación 2:

$$P(g) = \frac{N(g)}{M} \quad \text{Ecuación 2.}$$

Donde M es el número de píxeles en la imagen y $N(g)$ es el número de píxeles en el nivel de intensidad g . Como con cualquier distribución de probabilidad todos los valores de $P(g)$ son menores o iguales que 1 y la suma de todos los valores de $P(g)$ es 1.

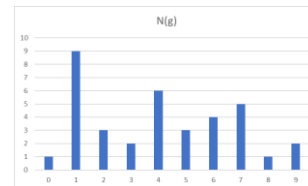
El histograma de una imagen y su representación constituye una herramienta visual de gran utilidad para el estudio de imágenes digitales, que puede proporcionar una idea muy aproximada de la distribución de niveles de gris.

Las intensidades o niveles de gris están representadas a lo largo del eje x , y el número de ocurrencias para cada intensidad se representa en el eje y .

Por ejemplo, dada la imagen de la figura 7 (a) con 10 niveles de gris, del 0 al 9; su histograma es el que se muestra en la figura 7 (b).

0	3	3	4	4	4
1	1	1	4	4	5
1	1	1	4	5	5
1	1	1	7	6	6
2	2	7	7	6	6
8	2	7	7	9	9

(a)



(b)

Figura 7. (a) Imagen con 10 niveles de gris del 0 al 9; (b) Su histograma, niveles de gris.

En la figura 8 se muestra una imagen de grises relativa a un reflectómetro óptico de los analizados en el proyecto y en la figura 9 su histograma de los niveles de gris en cada uno de los tres canales con valores de intensidad variando en el rango del 0 al 255. La misma imagen se triplica de forma que cada canal R , G y B representa exactamente la misma imagen.

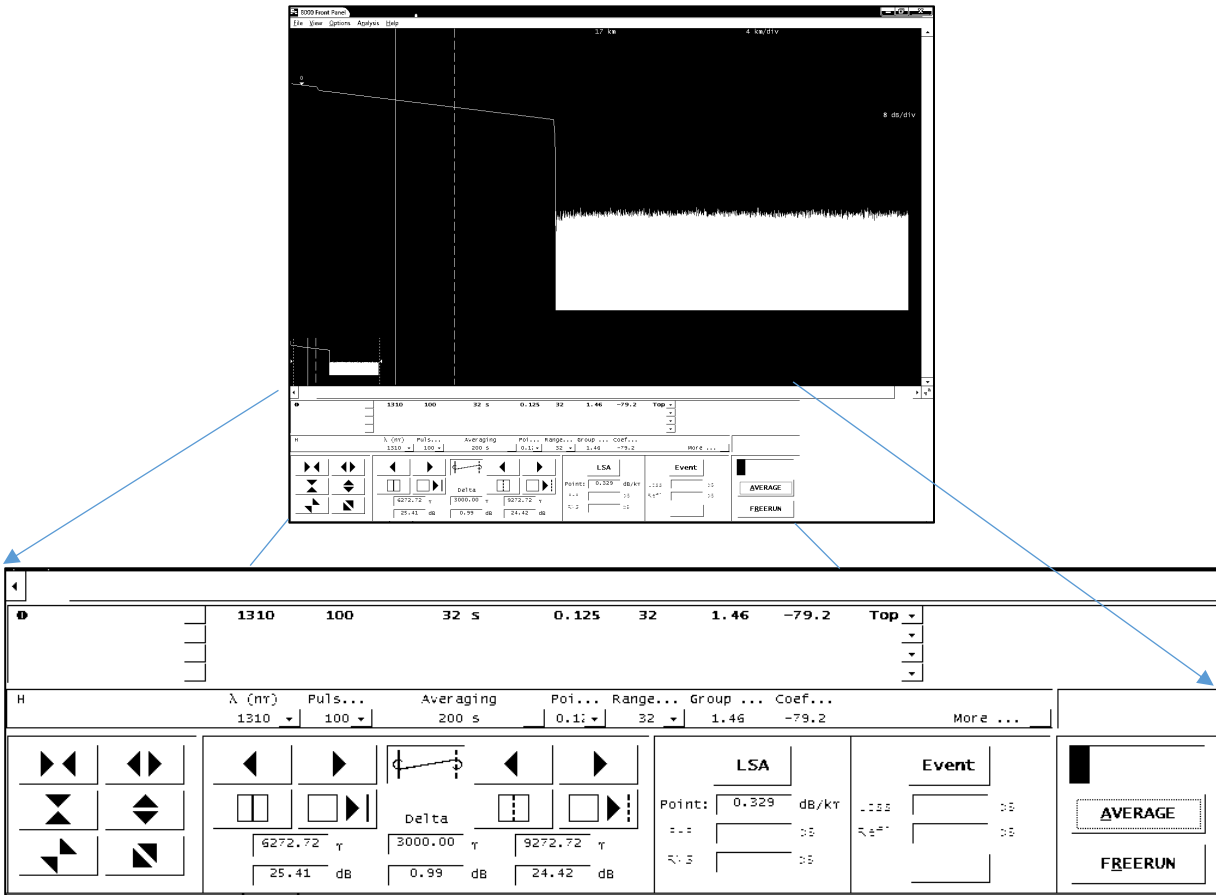


Figura 8. Imagen original de grises de la pantalla de un reflectómetro óptico.

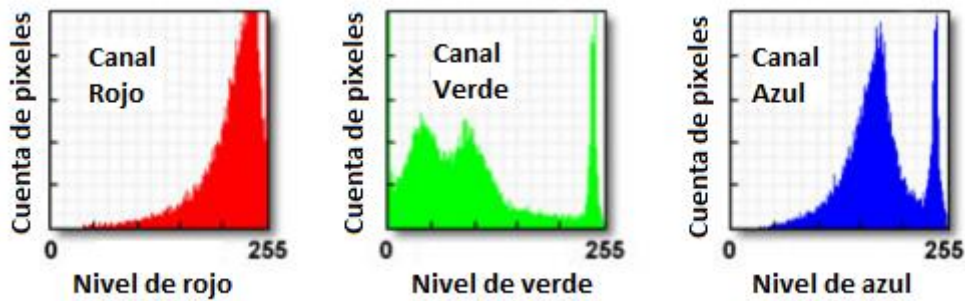


Figura 9. Histogramas de los tres canales espectrales en el modelo de color RGB.

La parte numérica de los reflectómetros ópticos que han servido de base para el desarrollo del presente proyecto se muestra a continuación, en la figura 10 para su posterior análisis.

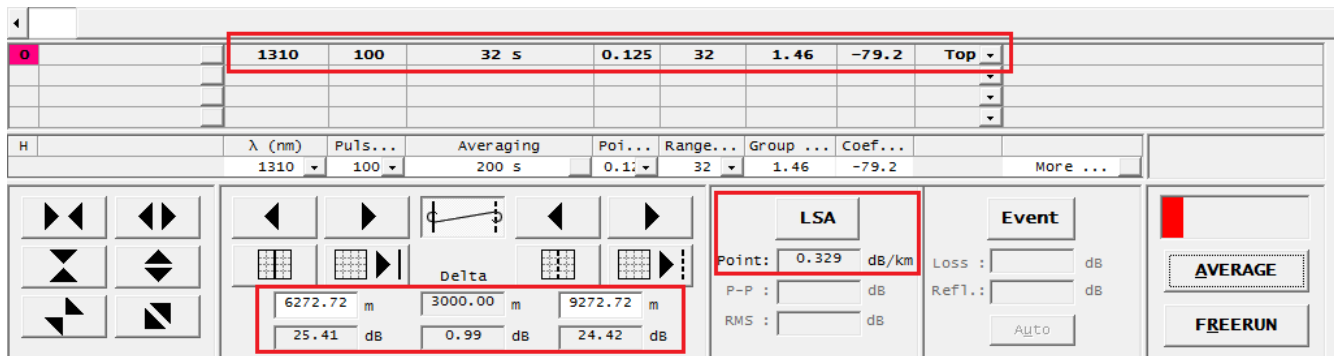


Figura 10. Localización de la parte numérica de interés de los reflectómetros ópticos.

2.5 Binarización de las imágenes.

La binarización de una imagen digital consiste en transformar la imagen en escala de grises en una imagen en blanco y negro. Para realizar la operación de binarización, se deberá elegir un valor adecuado de umbral dentro de los niveles de grises. Una vez elegido el umbral, todos los niveles de grises menores que el valor de umbral fijado se convertirán en negro y todos los que resulten ser mayores en blanco, ver ecuación 3.

$$g(x, y) = \begin{cases} 0, & \text{si } f(x, y) < T \\ 1, & \text{si } f(x, y) \geq T \end{cases} \quad \text{Ecuación 3.}$$

El cual es un punto clave el seleccionar un valor de umbral para T óptimo.

Para realizar la binarización nos basaremos en el histograma de la imagen, que constituye el punto de partida del proceso.

Existen diferentes métodos de binarización de imágenes. De entre ellos, el método de Otsu [4] es uno de los clásicos y más populares. Otro método posible es la binarización mediante la media estadística del histograma como valor de umbral.

2.5.1 Binarización mediante el método de Otsu.

El método de Otsu resulta ser un método de binarización de imágenes digitales ampliamente utilizado en la literatura. Se basa en conceptos estadísticos, particularmente en la dispersión de valores – en este caso, niveles de gris – a través de la varianza, Bostjan [5]. Si partimos de una imagen con L niveles de intensidad, en nuestro caso 256, y asumiendo que el umbral buscado es T , las probabilidades hasta T y desde T hasta L resultan ser como en la ecuación 4:

$$w1(t) = \sum_{z=1}^T P(z), \quad w2(t) = \sum_{z=T+1}^L P(z) \quad \text{Ecuación 4.}$$

Donde $P(z)$ es el total de apariciones del nivel de intensidad z . A continuación, se obtienen las medias y las varianzas asociadas, ecuación 5:

$$\sigma_1(t)^2 = \sum_{z=1}^T (z - \mu_1(t))^2 \frac{P(z)}{w_1(t)}; \quad \sigma_2(t)^2 = \sum_{z=T+1}^L (z - \mu_2(t))^2 \frac{P(z)}{w_2(t)}$$

Ecuación 5.

Finalmente, se obtiene la varianza ponderada, ecuación 6:

$$\sigma_w^2(t) = w_1(t)\sigma_1(t)^2 + w_2(t)\sigma_2(t)^2$$

Ecuación 6.

Se elige el umbral T correspondiente al nivel de intensidad que proporcione la mínima varianza ponderada.

2.5.2 Binarización mediante la media.

Por media se entiende el valor medio de los niveles de intensidad en la imagen. Se obtiene directamente a partir del histograma de la imagen, mediante la ecuación 7:

$$m = \sum_{i=0}^{L-1} z_i * p(z_i),$$

Ecuación 7.

Donde L es el número total de niveles de intensidad, en nuestro caso 256. Las imágenes mostradas en la figura 11 son el resultado de aplicar los mencionados métodos de binarización.



Figura 11. Imágenes binarizadas con la media – izquierda – y Otsu – derecha.

2.6 Fase de análisis.

A continuación, se documenta y se elabora respecto las fases típicas de todo proyecto de desarrollo de software, que son las aplicadas en este caso:

- Fase de requisitos, en la cual se identificarán los requisitos del sistema.
- Fase de diseño, en esta fase se detallarán los aspectos más importantes de la arquitectura del sistema.
- Fase de implementación, es en la que se traduce el diseño a un lenguaje de programación – en nuestro caso Labview, generación de código e implementación de funciones como tareas principales.
- Fase de pruebas, como en todo proyecto, se realizan diversas pruebas del software para comprobar si se obtiene el resultado esperado.
- Fase de mantenimiento, está orientado a mantener la operatividad del software, ya que al utilizarlo como usuario final puede ser que no cumpla con todas las expectativas previstas.

Todas las acciones y tareas que se incluyen y se realizan sobre cada una de estas fases, es lo que se denomina el ciclo de vida de un proyecto software.

Existen varios modelos para explicar el ciclo de vida de un proyecto de estas características. Uno de los más comunes es sin duda el relativo al desarrollo en cascada, ver figura 12. Es una visión del proceso de desarrollo de software como una sucesión de fases que producen productos intermedios y al final de cada una de estas fases se deben revisar las tareas y productos que se han generado en la fase correspondiente. Es decir, cada etapa deja el camino preparado para la siguiente, de forma que esta última no puede comenzar hasta que no haya terminado la anterior.

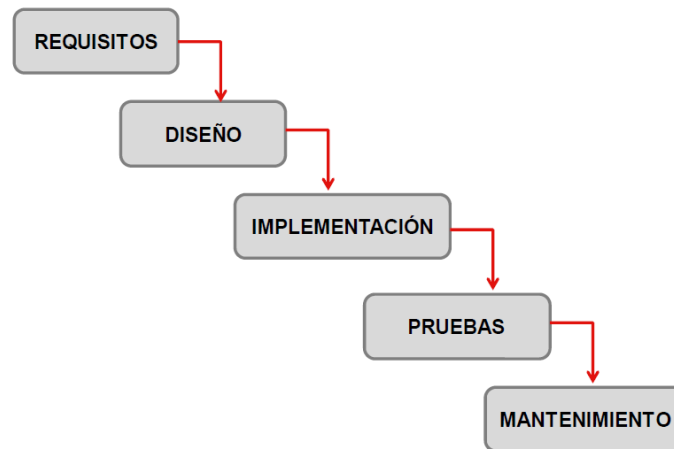


Figura 12. Modelo en cascada del desarrollo software.

La principal limitación es que no se permiten hacer iteraciones entre las fases y por tanto resulta poco realista desde un punto de vista práctico. De estos problemas surgen las principales variantes a este modelo, como por ejemplo el modelo en cascada con retroalimentación entre fases, ver figura 13. Si al terminar una etapa los resultados no son los esperados, podemos volver a la fase anterior. Este modelo es mucho más flexible y realista en la práctica que el anterior. Además, promueve una metodología de trabajo efectiva basada en los siguientes principios: definir antes de diseñar, diseñar antes de implementar.

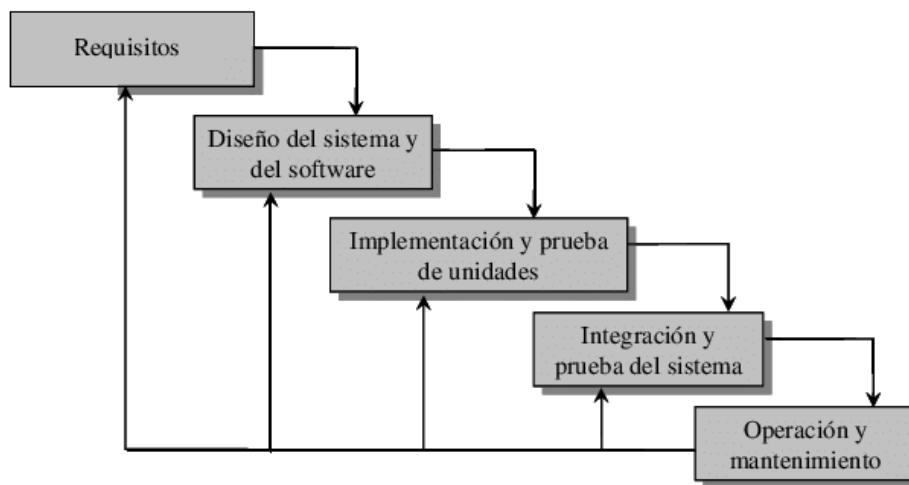


Figura 13. Modelo en cascada del desarrollo software con retroalimentación.

Por lo tanto, y en vista de las ventajas y desventajas de ambos modelos, se eligió el modelo en cascada con retroalimentación a la hora de desarrollar el proyecto de la aplicación de software.

Los requisitos que se explican a continuación están relacionados directamente con las propiedades que debe tener el funcionamiento de la aplicación del software.

Facilidad para la ejecución paso a paso.

Para lograr que la aplicación sea lo más sencilla posible e incluso didáctica trataremos de que el algoritmo del programa se pueda ir ejecutando paso por paso, identificándose claramente en cada uno de estos pasos que se está haciendo y por qué.

Facilidad de uso de la aplicación.

La intención es que la aplicación sea lo más fácil posible de utilizar prescindiendo de interfaces complicadas y recargadas de botones. Mientras menos botones más facilidad de uso. Además, éstos serán muy intuitivos desde un punto de vista práctico.

Cálculos complejos transparentes al usuario.

Se pretende que la aplicación sea lo más visual posible, por lo que los cálculos o tratamientos de la imagen junto con sus algoritmos deben quedar lo más posible ocultos al usuario. Al usuario final del sistema sólo le interesa que el sistema reconozca el conjunto de datos con los resultados de las mediciones y la configuración utilizada para ello dentro de la calibración del reflectómetro óptico. El cómo se llega al reconocimiento de los caracteres debe quedar oculto porque complicaría la interfaz y el uso del programa.

Rapidez de reconocimiento.

Los algoritmos para utilizar deben ser lo menos complejos temporalmente posible, lograr una aplicación veloz en el reconocimiento, para que pueda usarse en la medida de lo posible en el sistema en tiempo real con una alta carga de trabajo.

Robustez del algoritmo.

Este es el gran caballo de batalla de todos los reconocedores de imágenes, el algoritmo debe ser robusto ante posibles eventualidades en la imagen original, debidas en gran parte a los problemas derivados de la captura de estas. Sin embargo, hay que mencionar que no todas las posibles dificultades son evitables.

El resto de los requisitos que se explican a continuación se refieren a propiedades sobre la calidad que debe tener el producto o aplicación final del software.

Escalabilidad.

El sistema podría ser ampliado con facilidad ya que el tratamiento de estas imágenes está dominado.

Modularidad.

Es recomendable que el programa esté dividido en módulos independientes de manera que cada uno de ellos realice tareas específicas y concretas. De esta forma si se necesita mejorar o corregir un aspecto del programa sólo tendríamos que modificar aquellos módulos implicados sin necesidad de intervenir en todo el programa. Por lo tanto, se logra también un mejor mantenimiento del sistema.

Usabilidad.

Se trata de que la aplicación pueda ser utilizada por todo tipo de usuarios, desde el usuario experto en la tecnología como el que no.

Fiabilidad.

El sistema debe dar resultados correctos.

2.7 Fase de diseño.

Después de la fase de análisis y de determinar las distintas propiedades o requisitos que debería tener el sistema a desarrollar, pasamos a la fase de diseño. En esta fase se realiza una propuesta de cómo deben ser las lecturas del reflectómetro óptico a nivel de diseño, que lógicamente vendrá guiado por la fase de análisis anteriormente expuesta. La estrategia de la solución de diseño se centra en **la creación de un programa capaz de reconocer los caracteres del reflectómetro óptico a partir de las imágenes de las capturas de pantalla de la computadora del cliente, que es en donde se encuentra la aplicación de control del reflectómetro óptico**. Se pretende que la solución sea lo más fácil posible de usar, por lo que se intenta automatizar y hacer lo más simple posible las operaciones a realizar por la aplicación, para alcanzar la funcionalidad y solución deseadas.

En la figura 14 podemos ver la arquitectura del sistema de medición requerido por la norma IEC/CEI-61746:2005 [6] que se implementa para poder llevar a cabo la calibración del OTDR en su escala de atenuación. Como veremos, es necesario tener presente esta arquitectura para poder definir adecuadamente la vista conceptual del sistema más adelante.



Figura 14. Sistema para la calibración de la escala de atenuación del OTDR.

2.7.1 Vista Conceptual.

La arquitectura para el diseño de la aplicación debe estar cimentada en los requisitos funcionales y no funcionales descritos en la fase de análisis, es decir la arquitectura debe diseñarse para lograr el objetivo final, esto es, reconocer

los números de interés del reflectómetro óptico. En la figura 15 podemos distinguir la siguiente vista conceptual teniendo en cuenta la visión que el usuario tendría de la aplicación.

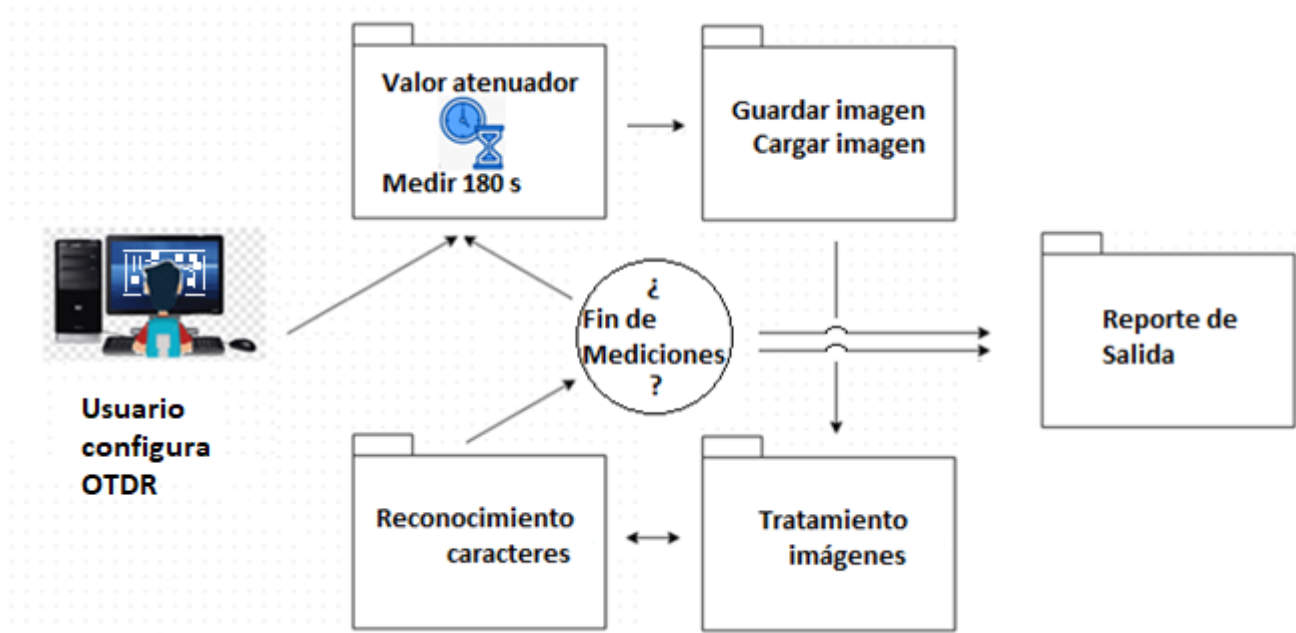


Figura 15. Vista conceptual.

2.7.2 Vista Lógica.

En lo que se refiere a la vista lógica, la aplicación, como puede verse en la vista conceptual, presenta cuatro subsistemas recogidos en un único sistema que proporciona la funcionalidad de los algoritmos para el tratamiento de las imágenes digitales. Se tiene por un lado el **valor** de referencia de un **atenuador** de señal y un tiempo de promediado inicial de **180 segundos**, **guardar y cargar imagen**, que será una captura de pantalla que se cargará directamente al sistema y de manera automática por la aplicación misma. Por otro lado, aparece el **tratamiento de imágenes**, el cual se encargará de localizar las zonas de interés del reflectómetro óptico. Y, por último, aparece el **reconocimiento de caracteres** del reflectómetro óptico; procediéndose a la lectura de caracteres en dichas zonas. En la figura 16 se muestra el esquema general de la vista lógica.

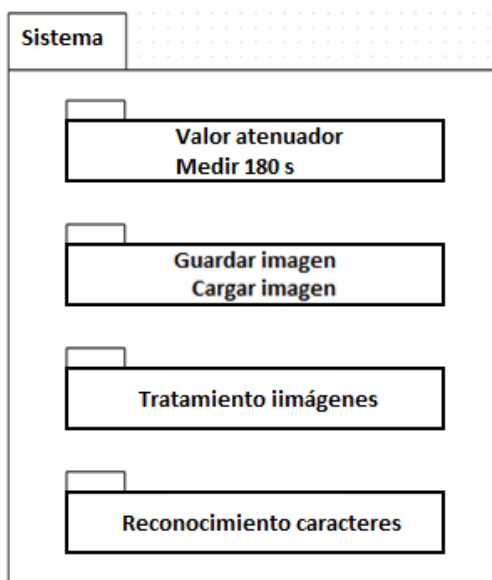


Figura 16. Vista lógica.

2.7.3 Arquitectura Modelo Vista Controlador.

La arquitectura Modelo Vista Controlador – MVC – es un patrón de diseño que separa la lógica de funcionamiento, la interfaz de usuario y los datos con los que se trabaja. Divide la aplicación en tres partes:

- Un modelo que contiene la información con la que se trabaja, es decir, los datos.
- Vistas que gestionan cómo se muestra la información al usuario de una manera accesible.
- Uno o varios controladores que realizan las funciones correspondientes en el modelo como respuesta a eventos procedentes generalmente de la vista.

Una ventaja de usar el MVC es que se incrementa el nivel de reutilización de los diferentes módulos software y de la aplicación en general. Para ello el modelo no puede tener acceso a ninguna clase del Controlador ni de la Vista. Además, el cambio de modelo no debe afectar a las vistas, y el controlador debe ver las clases del modelo, pero no de la vista. En la figura 17 pueden observarse las principales características de este patrón, así como la manera de interactuar entre los distintos componentes que forman parte de la arquitectura MVC.

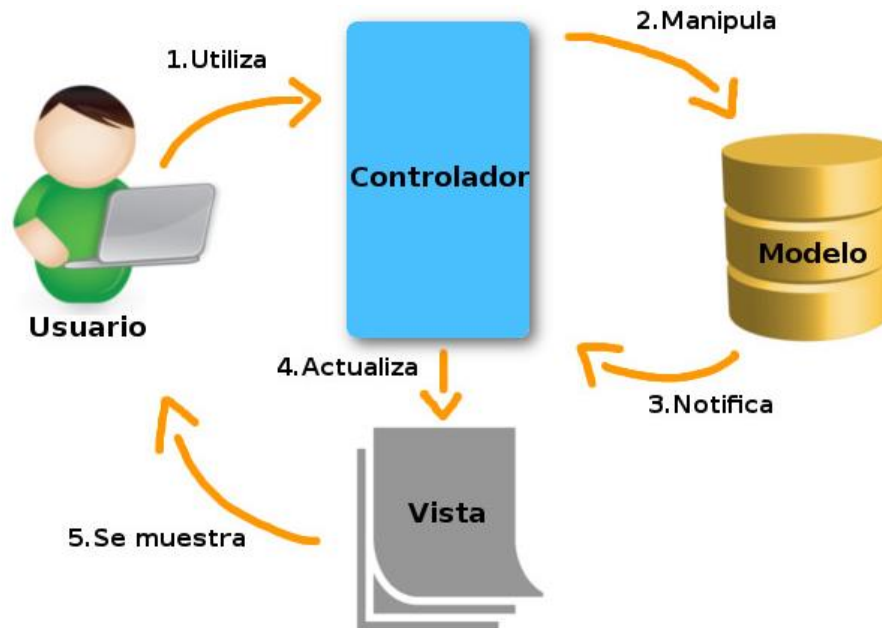


Figura 17. Arquitectura modelo-vista-controlador.

El uso del patrón MVC permite la evolución de forma separada de cada una de las partes, lo que proporciona una mayor flexibilidad a la hora de desarrollar los componentes del sistema.

Aunque existen diferentes implementaciones del patrón MVC, normalmente el flujo que sigue es el siguiente:

- Usuario interactúa con la interfaz de usuario.
- El controlador gestiona el evento de entrada.
- El controlador informa al modelo sobre la acción elegida por el usuario, pudiendo acceder al modelo y modificarlo si fuera necesario de acuerdo con dicha acción.
- Se modifica la vista, tomando los datos del modelo.
- La interfaz de usuario espera nuevas interacciones del usuario.

Las ventajas sobre el uso del patrón MVC son las que se expresan a continuación:

- Teniendo un mismo modelo, es posible conservar diferentes vistas.
- Es posible añadir nuevas vistas sin la necesidad de modificar el modelo
- Menor acoplamiento entre vistas y modelo.
- Diseño más claro.

Evidentemente, este modelo no es único, existen muchas variaciones e interpretaciones, si bien, lo importante es separar los tres elementos de que consta, como el utilizado en este trabajo.

2.7.4 Diseño del módulo.

A continuación, se muestra en la figura 18, el módulo del sistema del reconocimiento óptico de caracteres que se encarga de extraer los caracteres de interés. Su entrada es la imagen digital del reflectómetro óptico y su salida son los números que hemos sido capaces de reconocer. En los pasos intermedios se indican las operaciones a realizar, exactamente con el nombre con el que aparecen en la figura,

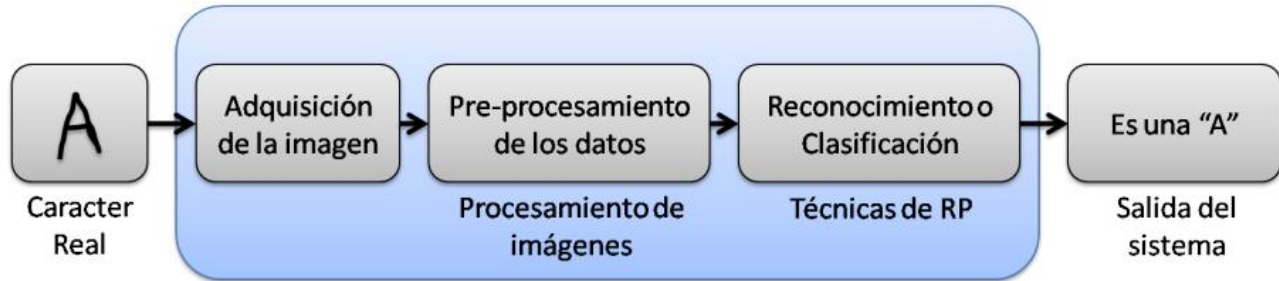


Figura 18. Diseño del módulo.

2.8 Fase de Implementación.

2.8.1 Descripción.

En esta fase se trata todo lo relacionado con la implementación de la aplicación, se verán las técnicas utilizadas para la extracción de los caracteres y las técnicas de uso para el reconocimiento de estos. También veremos cuál es la estructura de la aplicación y las distintas fases que la componen. Todo este proceso de implementación se adecua a todo lo desarrollado en la fase de análisis y en la fase de diseño.

2.8.2 Lenguaje de Programación: Labview 2010.

El hecho de que sea un lenguaje multiplataforma y el tener experiencia con este lenguaje a lo largo de los años fue decisivo a la hora de elegirlo. LabVIEW de National Instrument es una herramienta de programación gráfica, altamente productiva, para la construcción de sistemas de adquisición de datos, instrumentación y control [8], ver figura 19.



Figura 19. Software de LabVIEW 2010.

LabVIEW nos da la capacidad de crear rápidamente una interfaz de usuario que nos proporciona la interactividad con el sistema. La programación G es el corazón de LabVIEW, y difiere de otros lenguajes de programación como C o Basic, en que éstos están basados en texto, mientras que G es una programación gráfica. Los programas en G, o instrumentos virtuales – *VIs por las siglas en inglés de: Virtual Instruments* – constan de una interfaz interactiva de usuario y un diagrama de flujo de datos que hace las funciones de código fuente.

Componentes de un instrumento virtual o VI.

El panel frontal: en él se encuentran los controles o entradas de datos y los indicadores o salidas de información de nuestro programa de software, ver figura 20.



Figura 20. Panel frontal de un instrumento virtual.

El diagrama de bloque o programa de acompañamiento para el panel frontal, aquí se localizan los componentes del programa “cableados” entre sí, es en donde se desarrolla el flujo y procesamiento de la información que conforma a nuestro programa de software, ver figura 21.

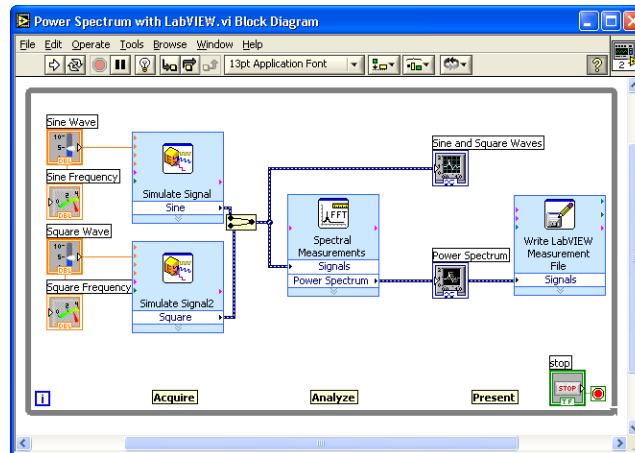


Figura 21. Diagrama de bloque de un instrumento virtual.

Como ya se mencionó, los programas en LabVIEW son llamados instrumentos virtuales o VIs. Cada VI contiene tres partes principales a saber: El Panel frontal – Cómo el usuario interactúa con el VI, ver figura 22. El diagrama de bloque – El código que controla el programa, ver figura 23. Y los Iconos o conectores – medios para conectar un VI con otros.

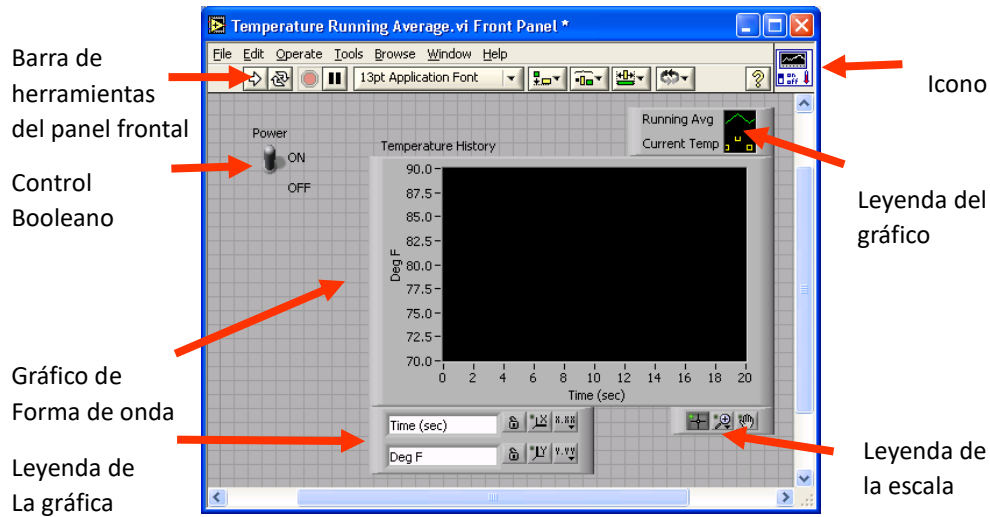


Figura 22. Panel frontal de un instrumento virtual.

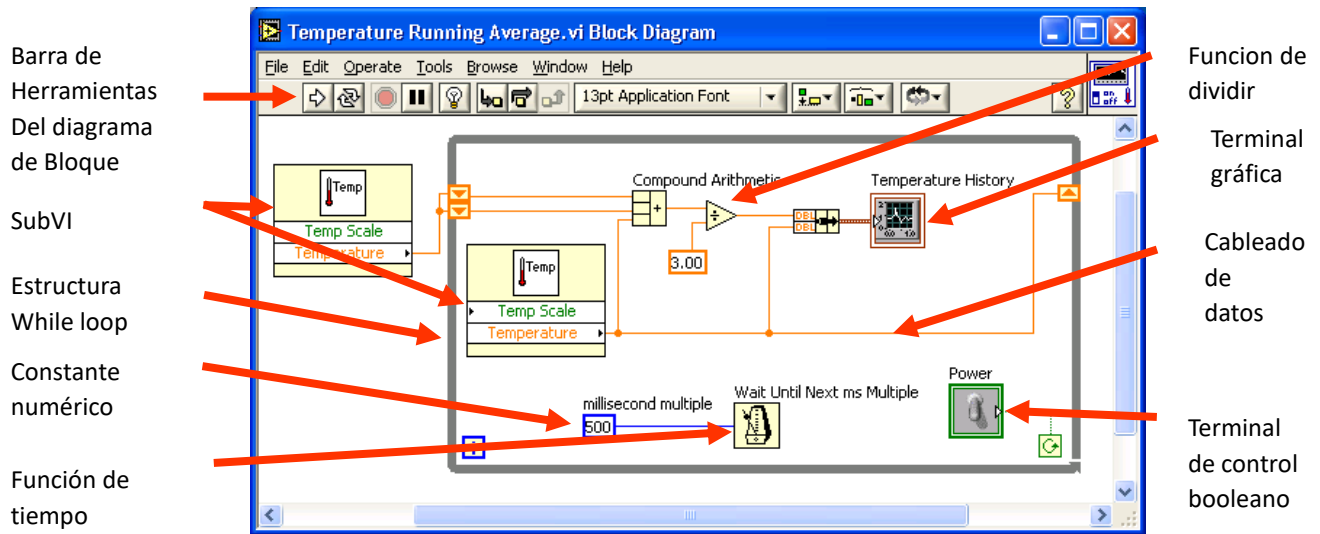


Figura 23. Diagrama de bloque de un instrumento virtual.

El panel frontal es utilizado para interactuar con el usuario cuando el programa está corriendo. Los usuarios pueden controlar el programa, cambiar entradas, y ver datos actualizados en tiempo real. Ajustando controles de deslizamiento para colocar un valor de alarma, encendiendo o apagando un switch, o parando un programa. Los indicadores son usados como salidas. Termómetros, luces, y otros indicadores indican valores del programa. Esto puede incluir datos,

estados de programa y otra información. Cada control o indicador del panel frontal tiene una terminal correspondiente en el diagrama de bloques. Adicionalmente, el diagrama de bloque contiene funciones y estructuras incorporadas en las bibliotecas de LabVIEW VI. Los cables conectan cada uno de los nodos en el diagrama de bloques, incluyendo controles e indicadores de terminal, funciones y estructuras. Cuando un VI se ejecuta, los valores de los controles fluyen a través del diagrama de bloques, estos son usados en las funciones del diagrama, y los resultados son pasados a otras funciones o indicadores.

Paleta de controles y funciones.

Se utiliza la paleta de controles – Controls - para colocar los controles e indicadores en el panel frontal. La paleta de controles está disponible solamente en el panel frontal. Se selecciona a través de Window»Show Controls palette o se hace clic derecho en el espacio de trabajo en el panel frontal para desplegar la paleta de controles, ver figura 24. También se puede desplegar la paleta de controles haciendo un clic derecho en un área abierta del panel frontal. Para desaparecer la paleta de controles se presiona el botón en la parte superior izquierda de la paleta. Se utiliza la paleta de funciones – Functions - para construir un diagrama de bloque. La paleta de funciones está disponible solamente en el diagrama de bloque. Se selecciona a través de Window»Show Functions Palette o se hace un click derecho en el espacio de trabajo del diagrama de bloque para desplegar la paleta de funciones, ver figura 25. También se puede desplegar la paleta de funciones dando un click derecho en un área abierta del diagrama de bloques. Para desaparecer la paleta de funciones se presionar el botón en la parte superior izquierda de la paleta.

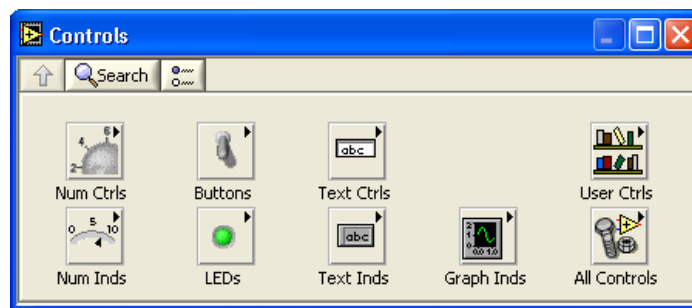


Figura 24. Paleta de controles en el Panel Frontal de un instrumento virtual.

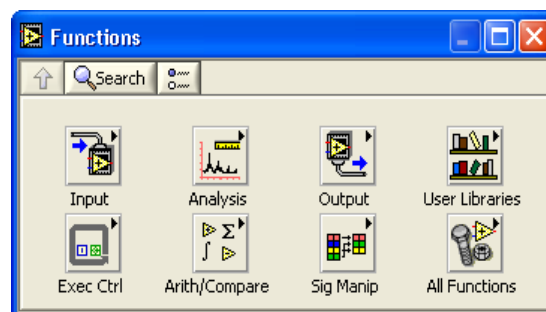


Figura 25. Paleta de funciones en el Diagrama de bloque de un instrumento virtual.

Paleta de Herramientas.

Es una Paleta flotante utilizada para operar y modificar objetos en el panel frontal y en el diagrama de bloques. Si la herramienta de selección automática está habilitada y uno mueve el cursor sobre un objeto en el panel frontal o en el diagrama de bloque, LabVIEW automáticamente selecciona la herramienta correspondiente de la paleta de controles. Para cambiar a la herramienta de selección automática, solo hay que teclear el botón de selección de Herramienta Automática en la paleta de Herramientas. Utilice la herramienta de operación para cambiar valores o para seleccionar texto de un control. Utilice la herramienta de posicionamiento para seleccionar, mover o redimensionar objetos. La herramienta de posicionamiento cambia las formas cuando se mueve sobre una esquina de un objeto reajutable. Se utiliza la herramienta de etiquetado para editar textos y crear etiquetas libres. La herramienta de etiquetado se convierte en cursor cuando se crean etiquetas libres. Se utiliza la herramienta de cableado para cablear objetos juntos en el diagrama de bloque, ver figura 26.



- **Paleta flotante**
- Utilizado para operar y modificar objetos en el panel frontal y en el diagrama de bloques.



Herramienta de selección automática



Herramienta de operación



Herramienta de desplazamiento



Herramienta de posicionamiento y redimensión



Herramienta de punto de paro



Herramienta de etiquetado



Herramienta de prueba



Herramienta de cableado



Herramienta para copia de color



Herramienta de menú (atajo)



Herramienta para colorear

Figura 26. Paleta de herramientas en el diagrama de bloque de un instrumento virtual.

Barra de herramientas de Estado.

Y por último la barra de herramientas de estado. Pulse el botón de ejecución – Run – para ejecutar el VI. Mientras el VI se está ejecutando, el botón de ejecución aparecerá con una flecha negra si es que el VI es un VI principal, lo que significa que no ha sido llamado por otro VI y por lo tanto este no es un subVI. Se pulsa el botón de ejecución continua – Continuous Run – para ejecutar el VI hasta que el botón de cancelación de ejecución o de pausa sea presionado. También se puede pulsar este botón nuevamente para deshabilitar la ejecución continua. Mientras el VI se está

ejecutando, el botón de cancelación de ejecución – Abort Execution – aparece. Presionando este botón se puede detener el VI inmediatamente. Nota: Hay que evitar en lo posible usar el botón de cancelación de ejecución – Abort Execution – para detener el VI. Es mejor dejar que el VI complete su flujo de datos o diseñar un método para detener el VI programáticamente. Al hacer esto el VI se encuentra en un estado conocido, lo cual es mejor en todos los casos.

Por ejemplo, colocando un botón en el panel frontal que detenga el VI cuando sea presionado. Se pulsa el botón de pausa – Pause – para detener momentáneamente la ejecución de un VI. Cuando se presiona el botón de Pausa, LabVIEW señala la posición donde uno detuvo la ejecución en el diagrama de bloque. Pulsando el botón de Pausa nuevamente el VI continúa ejecutándose nuevamente. Al seleccionar el menú de Configuración de Textos – Text Settings – se puede cambiar el tipo de fuente – letra – del VI, incluyendo el tamaño, estilo y color. Seleccionando el menú Alineamiento de Objetos – Align Objects – se pueden alinear objetos con respecto a los ejes, incluyendo eje vertical, superior, izquierdo, etc. Seleccionando el menú de Distribución de Objetos – Distribute Objects – se pueden espaciar objetos uniformemente, incluyendo espacios vacíos, compresiones, etc. Al seleccionar el botón de Redimensionamiento de Objetos – Resize Objects – se pueden cambiar el ancho y alto de objetos del panel frontal, ver figura 27.

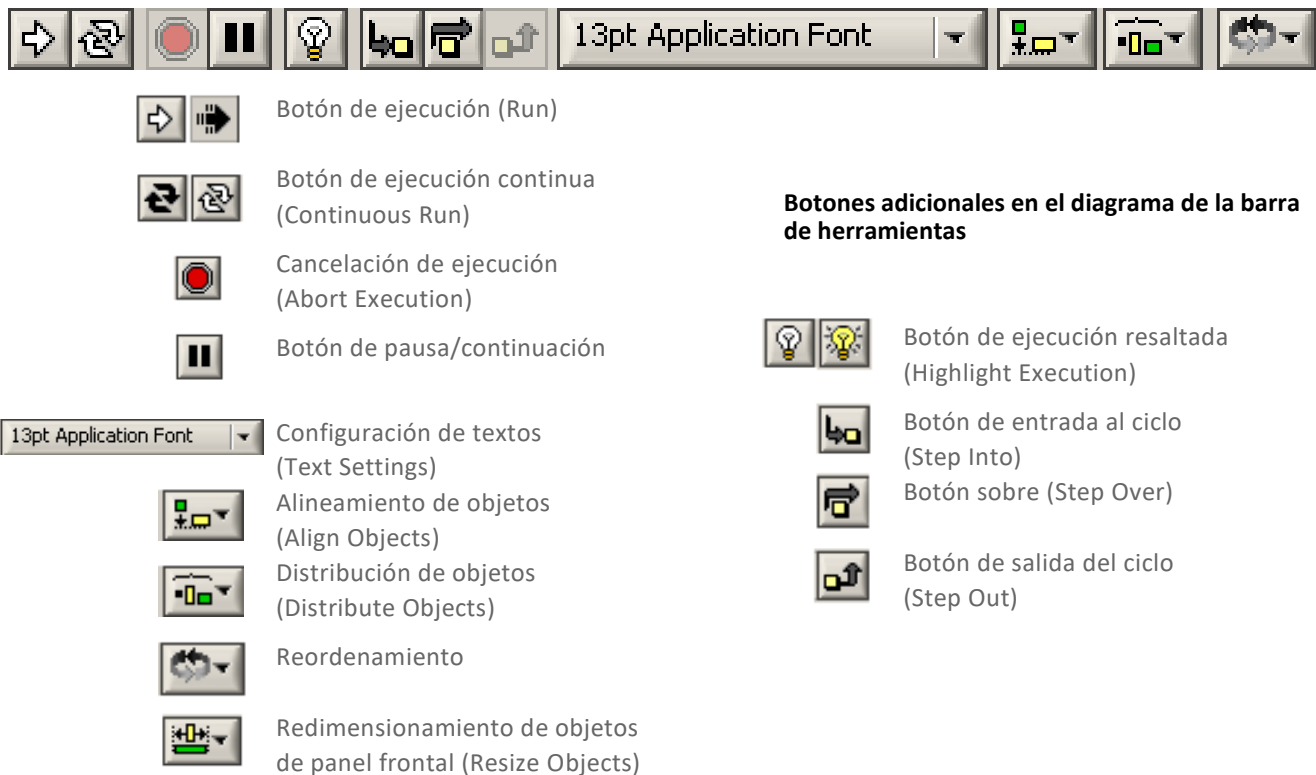


Figura 27. Barra de herramientas de estado en el Panel Frontal de un instrumento virtual.

EL Módulo de Desarrollo de Visión de LabVIEW.

El Módulo de Desarrollo de Visión de LabVIEW de National Instruments – VDM de NI por las siglas en inglés de: Vision Development Module; ofrece cientos de funciones para desarrollar e implementar aplicaciones de visión artificial.

El Vision Development Module – VDM de NI – está diseñado para ayudarnos a desarrollar aplicaciones de visión artificial usando LabVIEW o C++ e implementar esas aplicaciones en hardware Windows o NI Linux Real-Time. Con su extensa biblioteca de funciones, se puede tener acceso a cientos de algoritmos de procesamiento de imágenes y funciones de visión artificial para mejorar imágenes, verificar presencia, ubicar características, identificar objetos, medir partes y más. El VDM también incluye el Vision Assistant, el cual es una herramienta de diseño de algoritmos que simplifica el diseño del sistema de visión, ayudándonos a desarrollar algoritmos para implementar en CPUs o FPGAs.

Y con este modulo completamos la lista de nuestras necesidades de software base de programación para poder llevar a cabo el diseño, desarrollo y puesta en marcha de nuestro presente proyecto de ocr.

Problemas y resolución.

El proyecto es un campo nuevo para su servidor, hablando en términos del tratamiento de imágenes, así como el del reconocimiento de caracteres. El desarrollo de una aplicación para tal fin ha requerido y formulado un importante trabajo de investigación, líneas y posibilidades. El hecho de que, los otdrs no cuenten con algún protocolo de comunicación de acceso a la información de interés, nos ha dirigido a buscar soluciones alternativas al respecto. Estas circunstancias han hecho que finalmente el trabajo haya sido complicado, no obstante, habiéndose logrado totalmente la resolución del problema.

2.9 Fase de pruebas.

El objetivo de la realización de pruebas respecto al desarrollo de la aplicación es cerciorarse y comprobar el correcto funcionamiento del sistema, así como garantizar que cumple con todos los requisitos propuestos en la fase de análisis.

Se realizaron una serie de pruebas que consistieron en poner a trabajar la aplicación, monitoreando las conversiones y/o extracciones de la información de interés, obteniéndose resultados satisfactorios de las mismas, se obtuvo un 100% de fiabilidad y confianza con respecto al uso y funcionamiento de la aplicación de ocr en los otdrs.

2.10 Aplicación desarrollada.

A continuación, se presenta la aplicación implementada como solución al proyecto. Su interfaz gráfica y la secuencia visual de su funcionalidad en forma de ayuda visual, las ilustraciones en la mayoría de ellas hablan por sí mismas.

Primero hay que instalar la aplicación en la computadora de control del otdr, enseguida realizar la configuración y estabilización previa del otdr, y configurar un tiempo de promediado en el otdr de 200 segundos, ver figura A.1.

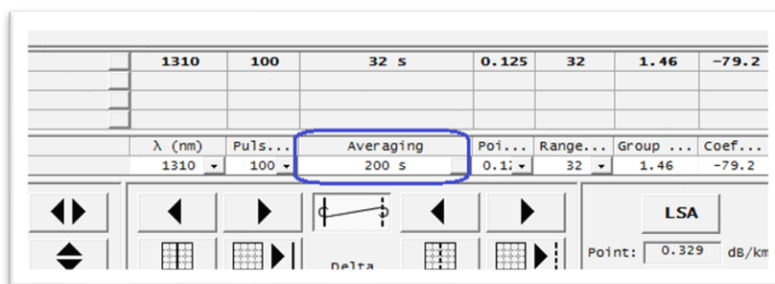


Figura A.1. configurar un tiempo de promediado en el otdr de 200 segundos.

Abrir la aplicación de calibración de atenuación en otdrs que es la que hace el ocr en los otdrs, ver figura A.2.



Figura A.2. Calibración de atenuación en otdrs.exe.

La aplicación y su interfaz de usuario principal se abrirá como se muestra en la figura A.3.

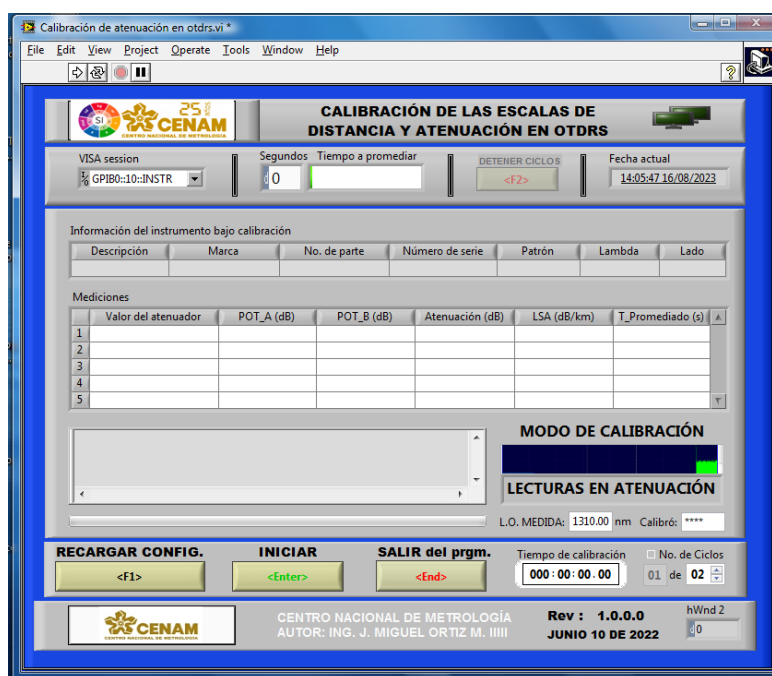


Figura A.3. Ciclo de espera. Interfaz de usuario de la aplicación del ocr en los otdrs.

Empezamos ingresando la longitud de onda medida, las iniciales del usuario y el número de ciclos a realizar, ver figura A.4.

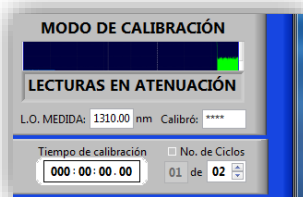


Figura A.4. Ingresar la longitud de onda medida, las iniciales del usuario y el número de ciclos a realizar.

Cada ciclo se compone de 21 mediciones de promedios de 180 segundos cada uno. Esto último se puede configurar desde los archivos externos de configuración de la aplicación.

Ahora si estamos listos para poder dar inicio al núcleo de la funcionalidad de la aplicación, ver figura A.5.



Figura A.5. Presionar “INICIAR” para comenzar con la rutina de la aplicación.

Ahora la aplicación nos va a requerir ingresar en el orden indicado en la figura A.6 la siguiente información:

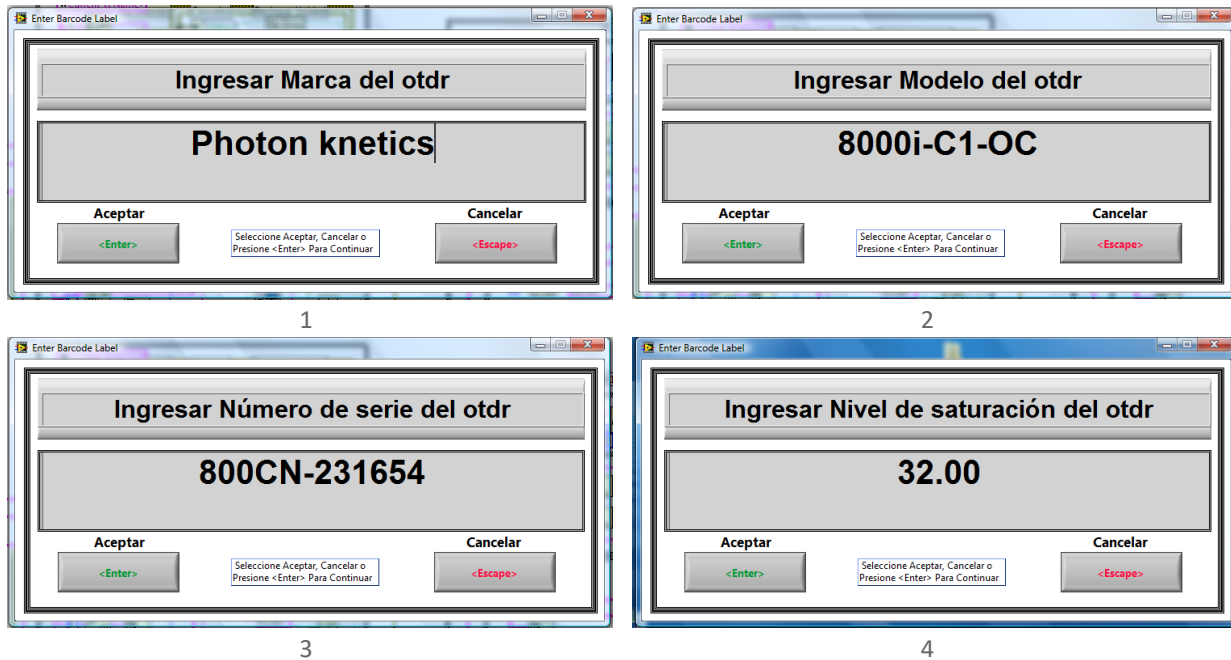
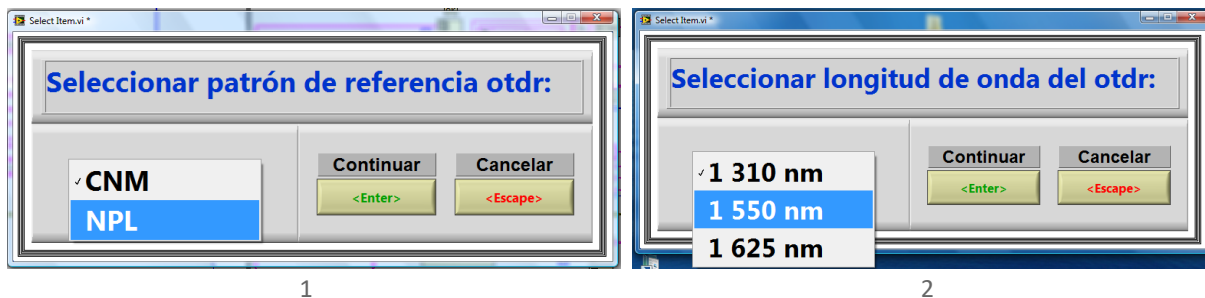
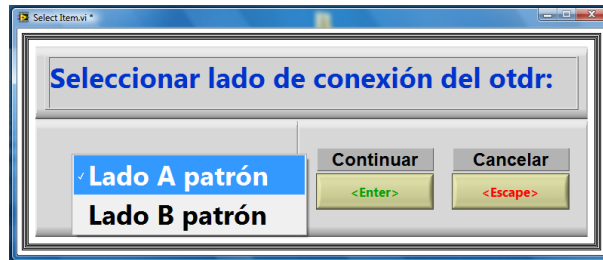


Figura A.6. Ingresar la información del otdr por el usuario de la aplicación según se indica.

Ahora la aplicación nos va a requerir seleccionar en el orden indicado en la figura A.7 la siguiente información, de acuerdo con las opciones disponibles mostradas:





3

Figura A.7. Seleccionar la información del otdr por el usuario de la aplicación según se indica.

Conforme se vaya avanzando en la aplicación, se irán actualizando los indicadores en la interfaz de usuario como corresponda, ver figura A.8.

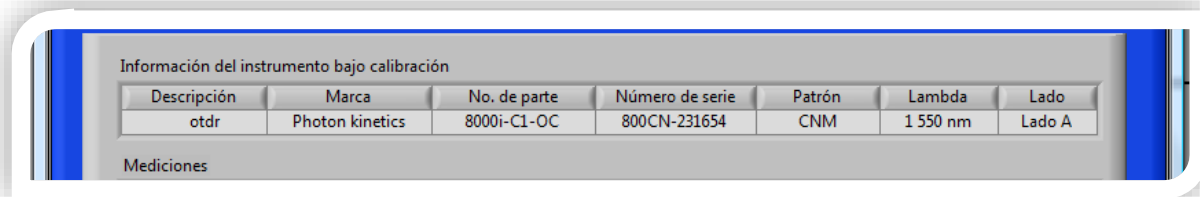


Figura A.8. La interfaz de usuario se va actualizando conforme se va ingresando esta información.

De aquí en adelante, la aplicación de software de labVIEW desarrollada, realiza de manera automática y sin la intervención necesaria del usuario todo el proceso siguiente que corresponde, ver figura A.9. La aplicación controla al software del otdr, le extrae una imagen de su pantalla, la procesa y le realiza el ocr.



Figura A.9. Información del ocr de la aplicación en tiempo real – indicador de “Mediciones”.

Se realiza al final un reporte consolidado y detallado de las mediciones con los resultados obtenidos y se informa de la finalización con éxito del proceso, ver figura A.10. En su caso y si es que existiera alguna eventualidad también será informada, describiendo los detalles en su caso.



Patrón: CNM
 Lnominal: 1 310 nm
 Lmedida: 1 310.00 nm
 Lado: Ai
 D.A-B: 3000.000 m

Fecha de cal.: 27/02/2023
 Hora de Inicio: 12:05 p.m.
 Hora de fin: 12:10 p.m.
 Calibró: JMOM

Fecha: 27/02/2023
 Clipping level (dB): 32.00

22.930	21.922	1.008	0.338
22.365	21.352	1.013	0.340
21.835	20.821	1.014	0.340
21.312	20.312	1.000	0.335
20.790	19.799	0.991	0.332
20.273	19.282	0.991	0.332
19.766	18.786	0.981	0.329
19.253	18.284	0.969	0.325
18.748	17.787	0.961	0.322
18.249	17.290	0.959	0.321
17.764	16.817	0.947	0.317
17.263	16.297	0.966	0.324
16.755	15.800	0.955	0.320
16.264	15.304	0.959	0.321
15.760	14.815	0.945	0.317
15.283	14.326	0.956	0.320
14.780	13.840	0.940	0.315
14.240	13.261	1.009	0.338
13.756	12.795	0.961	0.322
13.299	12.370	0.929	0.311
12.839	11.892	0.946	0.317

Pot. (dB) Pot. (dB) Atn. (dB) Coef. Atn (dB/km)

Atenuador (dB) **Lambda(nm)** Pulse(m) Avrg(s) Point(m) Range(km) G.I. Coeff.(dB) **A(m) B(m) D(m)**

0.0	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
0.5	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
1.0	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
1.5	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
2.0	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
2.5	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
3.0	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
3.5	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
4.0	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
4.5	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
5.0	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
5.5	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
6.0	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
6.5	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
7.0	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
7.5	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
8.0	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
8.5	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
9.0	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
9.5	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000
10.0	1310	100	180	10	32	1.46	-79.2	2.000	5.000	3.000

Check de specs: OK

Instrumento: OTDR
 Marca: EXFO
 Modelo: FTB-730-23B-EI
 No. de Serie: 548991

Figura A.10. Reporte de Información del ocr de la aplicación.

3. CONCLUSIONES.

El proyecto desarrollado ha tenido su motivación en un interés y/o problema real, consistente en la automatización del proceso de lectura de los valores de los resultados de las mediciones de la calibración de la escala de atenuación en decibeles de los reflectómetros ópticos de dominio en el tiempo, con el fin de automatizar un proceso tedioso y complicado, que ha de redundar necesariamente en una mayor efectividad laboral a la vez que se reducen los costos derivados del proceso de errores de lectura mediante el procedimiento clásico y, por supuesto no más papel de formatos a llenar a mano.

Nuevos conceptos han sido aplicados en este proyecto y se ha llegado a ampliar desarrollos relativos al tratamiento y manipulación de imágenes, abriendo todo un campo nuevo de posibilidades en el que todavía queda mucho por hacer e investigar. En cuanto al diseño de la interfaz gráfica se ha conseguido una estructura eficaz, capaz de proporcionar al usuario un punto de vista amigable y eficiente mediante el uso fácil de la misma.

Trabajo futuro.

Mediante el desarrollo realizado se ha conseguido una aplicación con los resultados indicados. Hay que continuar en la búsqueda de oportunidades para poder agregar el uso de estas herramientas de apoyo a situaciones de necesidad como la descrita en el presente trabajo.

4. REFERENCIAS BIBLIOGRÁFICAS, JURÍDICAS Y/O HEMEROGRÁFICAS.

- [1] [Reflectómetro óptico de dominio en el tiempo - OTDR - Wikipedia, la enciclopedia libre](#)
- [2] [Reconocimiento óptico de caracteres - OCR - Wikipedia, la enciclopedia libre](#)
- [3] Pajares, G. y de la Cruz, J.M (2007). Visión por computador. Imágenes digitales y aplicaciones.
- [4] Otsu, N. (1979). A threshold selection method from gray-level histogram. IEEE Transactions on System Man and Cybernetics 9, 62-66, 1979.
- [5] Bostjan (2015). Java Image Binarization using Otsu's algorithm <http://developer.bostjan-cigan.com/java-image-binarization/>.
- [6] Subcommittee 86A, "Optical Fibres: Calibration of optical time domain reflectometers (OTDR)", IEC/CEI-61746:2005, International Electrotechnical Commission, Geneva, Switzerland, (2005).
- [7] Bunke, H. y Wang, P.S.P. (Eds.) (1997). Handbook of Character Recognition and Document Image Analysis. World Scientific.
- [8] [National Instruments, www.ni.com](http://www.ni.com).