



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

**“DESARROLLO DE UN INSTRUMENTO VIRTUAL
EXPERIMENTAL QUE DETERMINE GRÁFICAMENTE
ESTRUCTURAS GRANULARES GENERADAS POR
INESTABILIDADES HIDRODINÁMICAS”**

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTORES: LUIS ISRAEL GRANIZO MORA

JOEL ALEXANDER ARGOS QUISHPE

DIRECTOR: Ing. JORGE LUIS PAUCAR SAMANIEGO

Riobamba – Ecuador

2023

© 2023, Luis Israel Granizo Mora; Joel Alexander Argos Quishpe

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Nosotros, Luis Israel Granizo Mora y Joel Alexander Argos Quishpe, declaramos que el presente Trabajo de Integración Curricular es de nuestra autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autores asumimos la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 14 de noviembre de 2023.



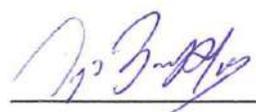
Luis Israel Granizo Mora
0605065291



Joel Alexander Argos Quishpe
0604818617

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; Tipo: Proyecto Técnico, “**DESARROLLO DE UN INSTRUMENTO VIRTUAL EXPERIMENTAL QUE DETERMINE GRÁFICAMENTE ESTRUCTURAS GRANULARES GENERADAS POR INESTABILIDADES HIDRODINÁMICAS**”, realizado por los señores: **LUIS ISRAEL GRANIZO MORA** y **JOEL ALEXANDER ARGOS QUSHPE**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Ing. Jorge Luis Hernandez Ambato PRESIDENTE DEL TRIBUNAL		2023-11-14
Ing. Jorge Luis Paucar Samaniego DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2023-11-14
Ing. Diego Guillermo Barba Maggi ASESOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2023-11-14

DEDICATORIA

A mi madre que me enseñó a pensar y a mi padre que me enseñó a soñar.

Israel

A mi hermana, cuya sonrisa es motivación suficiente para levantarme cada mañana y abrir las persianas de mi ventana, cuya mirada penetra en el fondo de mi alma, con la frescura de su voz describiendo este maravilloso obsequio llamado presente, convirtiendo hasta la más dura herida en una hermosa cicatriz, dándole colores incluso al cielo más gris. A mis padres, cuyo acerado esfuerzo ha hecho de mi ser algo extraordinario, llevando siempre con orgullo el hilo de sangre que me ha sido otorgado.

Joel

AGRADECIMIENTO

He de agradecer profundamente a mis padres, por enseñarme a que la familia es lo más importante, encontrando mi mayor inspiración en su sacrificio diario, en su despertar demasiado temprano para traer el pan a la mesa del hogar, gracias por su apoyo infinito, y perdón por no ser el mejor hijo. A mis hermanos, cuyo vivo reflejo demuestra todo lo que hemos superado en las inmensas batallas de la vida, gracias por su presencia y su alegría, gracias por compartir conmigo este pedacito de mi vida, que el éxito abrigue sus corazones hasta la eternidad y que lleven siempre con orgullo el semblante de nuestra familia. A mis amigos, que han sido testigos del camino recorrido, gracias por tantos momentos vividos y por tantas sonrisas, que mantendré guardados en lo más profundo de mi ser. A mi tutor y asesor de tesis, gracias por sus enseñanzas y su tutela en todo este trayecto. A mi amigo Israel, gracias por acompañarme en este camino y añadir un escalón más para alcanzar el podio del éxito y la buenaventura. A todas aquellas personas que han sabido poner sus valores por encima de sus penumbras, aguantando con dignidad todos los golpes que da la vida, gracias por ser fruto de mi inspiración.

Joel

Queridos padres y hermano, al culminar este emocionante capítulo de mi vida académica, no encuentro palabras suficientes para expresar mi profunda gratitud hacia ustedes. Su amor incondicional, sacrificio y apoyo constante han sido los pilares que han sostenido cada paso de mi camino hacia la culminación de este proyecto académico. A través de las incontables horas de estudio, los desafíos superados y las victorias celebradas, su respaldo ha sido mi fuente inagotable de inspiración. A mi tutor y asesor de tesis gracias por su guía experta, paciencia y dedicación inquebrantable. A mi compañero de tesis, Joel, le agradezco sinceramente su colaboración, entusiasmo y compromiso, elementos clave en el desarrollo de nuestro proyecto. Este logro no solo representa mi esfuerzo individual, sino también el resultado de personas excepcionales que han estado a mi lado en cada etapa. A todos ustedes, mi más sincero agradecimiento. Su presencia ha sido fundamental en este viaje, convirtiéndolo en una experiencia rica en aprendizaje, crecimiento y éxito. Con gratitud,

Israel

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS.....	xiv
ÍNDICE DE ILUSTRACIONES.....	xvi
ÍNDICE DE ECUACIONES	xxi
ÍNDICE DE SIMBOLOGÍA	xxii
ÍNDICE DE ANEXOS	xxiii
RESUMEN.....	xxiv
ABSTRACT	xxv
INTRODUCCIÓN	1

CAPÍTULO I

1.	DIAGNÓSTICO DEL PROBLEMA	2
1.1	Antecedentes	2
1.2	Planteamiento del problema.....	3
1.2.1	<i>Sistematización del problema</i>	<i>4</i>
1.3	Justificación Teórica	4
1.4	Justificación Aplicativa.....	6
1.5	Objetivos	7
1.5.1	<i>Objetivo general.....</i>	<i>7</i>
1.5.2	<i>Objetivos específicos</i>	<i>7</i>
1.6	Metodología	8

CAPÍTULO II

2.	MARCO TEÓRICO	9
2.1	Estructuras granulares	9
2.1.1	<i>Propiedades físicas de las estructuras granulares</i>	<i>9</i>

2.1.1.1	<i>Tamaño y forma de las partículas</i>	9
2.1.1.2	<i>Empaquetamiento de partículas</i>	9
2.1.1.3	<i>Porosidad y densidad</i>	10
2.1.1.4	<i>Propiedades mecánicas</i>	10
2.1.2	<i>Tipos de estructuras granulares</i>	10
2.1.2.1	<i>Estructuras granulares amorfas</i>	10
2.1.2.2	<i>Estructuras granulares cúbicas</i>	11
2.1.2.3	<i>Estructuras granulares en capas</i>	11
2.1.2.4	<i>Estructuras granulares en cadena</i>	11
2.1.2.5	<i>Estructuras granulares con fuerzas activas</i>	12
2.2	<i>Suspensión de partículas</i>	12
2.2.1	<i>Propiedades de las partículas</i>	12
2.3	<i>Inestabilidades</i>	13
2.3.1	<i>Tipos de inestabilidades</i>	13
2.4	<i>Inestabilidades hidrodinámicas</i>	14
2.4.1	<i>Inestabilidad de Faraday</i>	14
2.5	<i>Instrumentación tradicional</i>	15
2.6	<i>Instrumentación virtual</i>	16
2.6.1	<i>Instrumentación virtual vs instrumentación tradicional</i>	17
2.6.2	<i>Herramientas de programación y desarrollo de instrumentación virtual</i>	18
2.6.2.1	<i>MATLAB (Matrix Laboratory)</i>	19
2.6.2.2	<i>LabWindows/CVI (C for Virtual Instrument)</i>	20
2.6.2.3	<i>LabVIEW (Laboratory Virtual Engineering Workbench)</i>	20
2.6.2.4	<i>Diferencias de programación</i>	22
2.6.3	<i>Diseño de instrumentación virtual</i>	22
2.6.3.1	<i>Adquisición de datos</i>	23
2.6.3.2	<i>Acondicionamiento de señales</i>	23
2.6.3.3	<i>Digitalización de señales</i>	24
2.6.3.4	<i>Diseño de la interfaz de usuario y adquisición de señales</i>	25

2.6.3.5	<i>Validación y verificación de la instrumentación virtual</i>	25
2.7	Procesamiento digital de imágenes	26
2.7.1	Adquisición de imágenes	27
2.7.1.1	<i>Detección y adquisición de imágenes</i>	28
2.7.1.2	<i>Muestreo y cuantificación de las imágenes</i>	28
2.7.1.3	<i>Resolución espacial y de intensidad</i>	29
2.7.1.4	<i>Interpolación de imágenes</i>	29
2.7.2	Filtrado y mejora de las imágenes	29
2.7.2.1	<i>Transformaciones de intensidad y filtrado espacial</i>	30
2.7.2.2	<i>Correlación espacial y convolución</i>	30
2.7.2.3	<i>Filtros espaciales de suavizado</i>	31
2.7.2.4	<i>Filtros espaciales de realce o enfoque</i>	32
2.7.3	Compresión de imágenes	32
2.7.3.1	<i>Tipos de compresión</i>	32
2.7.3.2	<i>Formatos de imagen</i>	33
2.7.4	Segmentación de imágenes	34
2.7.4.1	<i>Segmentación basada en bordes</i>	35
2.7.4.2	<i>Segmentación basada en umbralización (Thresholding)</i>	36
2.7.5	Representación y descripción	36
2.7.5.1	<i>Representación</i>	36
2.7.5.2	<i>Descriptores de contorno</i>	37
2.7.5.3	<i>Descriptores regionales</i>	38
2.7.6	Reconocimiento de objetos usando redes neuronales artificiales	40
2.7.6.1	<i>Patrones y clases de patrones</i>	40
2.7.6.2	<i>Tipo de Datasets</i>	40
2.7.6.3	<i>Deep Learning</i>	41

CAPÍTULO III

3.	MARCO METODOLÓGICO	44
3.1	Requerimientos del sistema	44
3.1.1	<i>Requerimientos generales</i>	44
3.2	Concepción de la arquitectura	45
3.3	Diseño de la arquitectura de los módulos del instrumento virtual	46
3.3.1	<i>Módulo de generación de vibraciones</i>	46
3.3.2	<i>Módulo de generación de muestras</i>	47
3.3.3	<i>Módulo de adquisición de datos</i>	48
3.3.4	<i>Módulo de procesamiento y visualización</i>	49
3.4	Dispositivos estructurales y de hardware	49
3.4.1	<i>Selección de los dispositivos estructurales y de hardware</i>	50
3.4.1.1	<i>Estructura antivibratoria</i>	50
3.4.1.2	<i>Generador de funciones</i>	51
3.4.1.3	<i>Amplificador de potencia lineal</i>	52
3.4.1.4	<i>Vibrador de imán permanente</i>	53
3.4.1.5	<i>Balanza de precisión</i>	54
3.4.1.6	<i>Agitador magnético</i>	54
3.4.1.7	<i>Cámara de alta resolución espacial</i>	55
3.4.1.8	<i>Cámara de alta resolución temporal</i>	56
3.4.2	<i>Dispositivos de iluminación</i>	57
3.4.3	<i>Celdas contenedoras</i>	58
3.5	Selección de las herramientas de software	59
3.5.1	<i>LabVIEW</i>	59
3.5.2	<i>Google Colab</i>	60
3.5.3	<i>Visual Studio Code</i>	61
3.6	Preparación de muestras	62
3.6.1	<i>Celda cilíndrica</i>	62

3.6.1.1	<i>Proporción de los componentes</i>	62
3.6.1.2	<i>Análisis de las muestras</i>	64
3.6.2	<i>Celda cuasi 2D</i>	66
3.6.2.1	<i>Proporción de los componentes</i>	66
3.6.2.2	<i>Ajuste de proporciones</i>	66
3.7	Diseño y desarrollo del instrumento virtual	67
3.7.1	<i>Creación, entrenamiento y validación de la red neuronal convolucional</i>	68
3.7.1.1	<i>Base de datos</i>	68
3.7.1.2	<i>Librerías necesarias</i>	69
3.7.1.3	<i>Importación de base de datos a Google Colab</i>	70
3.7.1.4	<i>Creación de la red neuronal convolucional</i>	71
3.7.1.5	<i>Entrenamiento de la red neuronal convolucional</i>	74
3.7.1.6	<i>Evaluación del modelo</i>	74
3.7.1.7	<i>Exportación del modelo entrenado</i>	75
3.7.1.8	<i>Función de importación del modelo y predicción en Python</i>	76
3.7.2	<i>Desarrollo del entorno virtual</i>	76
3.7.2.1	<i>Llamado de la función en Python</i>	77
3.7.2.2	<i>Comunicación con Hardware</i>	78
3.7.3	<i>Funcionamiento del instrumento virtual mediante diagramas de flujo</i>	79
3.7.4	<i>Validación del instrumento virtual bajo estándares internacionales</i>	85

CAPÍTULO IV

4.	PRUEBAS	87
4.1	Equipos para la validación experimental	87
4.1.1	<i>Acelerómetro ADXL326</i>	87
4.1.2	<i>myDAQ</i>	88
4.1.3	<i>Programación en LabVIEW para la lectura de aceleración y frecuencia</i>	89
4.2	Implementación del equipo	90

4.3	Análisis de patrones en la celda cilíndrica	90
4.3.1	<i>Nivelación de la celda cilíndrica</i>	91
4.3.2	<i>Captura de imágenes de las estructuras granulares</i>	91
4.3.3	<i>Clasificación de las imágenes y comparación de los patrones obtenidos</i>	93
4.3.3.1	<i>Comparación a distintos espesores de la capa de líquido Δh</i>	94
4.3.4	<i>Momentos invariantes de Zernike y creación de la base de datos</i>	95
4.4	Análisis de patrones en la celda cuasi 2D	98
4.4.1	<i>Formación y comparación de ondas de Faraday</i>	98
4.4.2	<i>Inestabilidad de Faraday</i>	100
4.5	Precisión del instrumento virtual	101
4.6	Obtención de datos de las estructuras granulares	102

CAPÍTULO V

5.	RESULTADOS	104
5.1	Evaluación de pérdidas en el entrenamiento de la red neuronal convolucional	104
5.2	Exactitud de la red neuronal convolucional	105
5.3	Detección de patrones mediante el instrumento virtual	105
5.3.1	<i>Análisis de patrones de anillos</i>	106
5.3.2	<i>Análisis de patrones diamantes</i>	108
5.4	Análisis de datos con el instrumento virtual	111
5.4.1	<i>Análisis de los datos en patrones de anillos</i>	111
5.4.2	<i>Análisis de los datos en estructuras de diamantes</i>	113
5.4.3	<i>Análisis de patrones en la celda cuasi 2D</i>	115

CAPÍTULO VI

6.	CONCLUSIONES Y RECOMENDACIONES	117
6.1	Conclusiones	117
6.2	Recomendaciones	118

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla 2-1:	Instrumentos tradicionales vs Instrumentos virtuales.	17
Tabla 2-2:	Tabla comparativa entre los distintos tipos de software de instrumentación.	22
Tabla 2-3:	Formatos de imagen más utilizados.	33
Tabla 2-4:	Momentos invariantes para las imágenes anteriores.	39
Tabla 3-1:	Dispositivos estructurales y de hardware utilizados para la obtención de estructuras granulares y procesamiento de imágenes.	49
Tabla 3-2:	Especificaciones técnicas del generador de señales SFG-2104.	52
Tabla 3-3:	Especificaciones técnicas del amplificador de potencia lineal LDS LPA 600....	52
Tabla 3-4:	Características específicas y técnicas principales del vibrador de imán permanente LDS V406.	53
Tabla 3-5:	Características específicas y técnicas de la balanza de precisión PS 4500.X2.M.	54
Tabla 3-6:	Características específicas y técnicas del agitador magnético MS7-H550-Pro...	55
Tabla 3-7:	Especificaciones técnicas de la cámara NIKON D7500.	56
Tabla 3-8:	Especificaciones técnicas de la cámara BASLER acA640.750um.	57
Tabla 3-9:	Datos obtenidos de las pruebas realizadas con distintos tipos de muestras.	65
Tabla 4-1:	Especificaciones técnicas del acelerómetro ADXL326.	88
Tabla 4-2:	Especificaciones técnicas de la tarjeta de adquisición <i>myDAQ</i>	89
Tabla 4-3:	Patrones en el lecho de partículas al ser sometidas a inestabilidades hidrodinámicas con diferentes valores de frecuencia y aceleración, con un Δh constante de 5 ml.	93
Tabla 4-4:	Comparación de los anillos de Faraday entre distintos espesores de la capa de líquido, sometidos a distintas frecuencias con una aceleración constante de 1g.	94
Tabla 4-5:	Comparación de patrones generados entre distintos espesores de la capa de líquido Δh , sometidos a distintos valores de frecuencia y aceleración.	97
Tabla 4-6:	Comparación de patrones inestables generados entre distintos espesores de la capa de líquido Δh , sometidos a distintos valores de frecuencia y aceleración.	97
Tabla 4-7:	Comparación del umbral de aparición de las ondas de Faraday a distintos niveles de espesor de la capa de líquido Δh	100

Tabla 4-8:	Análisis de precisión del instrumento virtual con pruebas repetidas.	101
Tabla 4-9:	Longitud de onda (λ), en píxeles, de anillos con estructuras de 5 ml a 1 g con distancias filtradas.....	102
Tabla 4-10:	Frecuencia f vs número de anillos N_a a distintos espesores de líquido Δh , en 5 ml y 6 ml a 1 g y en 7 ml a 0.9 g.....	102
Tabla 4-11:	Análisis de las estructuras de diamantes a varias frecuencias con 5ml de espesor de líquido.	103
Tabla 4-12:	Análisis de las estructuras de diamantes a varias frecuencias con 6ml de espesor de líquido.	103
Tabla 4-13:	Análisis de las estructuras de diamantes a varias frecuencias con 7ml de espesor de líquido.	103
Tabla 5-1:	Resumen de los resultados obtenidos de la red neuronal durante 70 épocas. ...	105

ÍNDICE DE ILUSTRACIONES

Ilustración 1-1:	Proceso dinámico del montaje de partículas flotantes en plantillas líquidas al variar parámetros de frecuencia y aceleración. El montaje empieza en cero hasta adquirir una forma de anillo, pasando por una estructura en forma de “H” para volver a su forma de estructura de anillo.	5
Ilustración 2-1:	Ejemplos de estructuras amorfas: Estructura de silicio y estructura de cuarzo.	11
Ilustración 2-2:	Estructura cristalina.....	11
Ilustración 2-3:	Fibras de ADN.	12
Ilustración 2-4:	Patrones generados en la inestabilidad de Faraday utilizando agua a una frecuencia de 70 Hz.....	14
Ilustración 2-5:	Ondas de Faraday usando la técnica de luz difusa.	15
Ilustración 2-6:	Izquierda: Instrumentos tradicionales. Derecha: Instrumentos virtuales. ...	17
Ilustración 2-7:	Instrumento virtual en MATLAB utilizando App Designer.	19
Ilustración 2-8:	Ejemplo de interfaz gráfica con LabWindows/CVI.	20
Ilustración 2-9:	Ejemplo de un panel frontal de un VI.	21
Ilustración 2-10:	Ejemplo de un diagrama de bloques de un VI.	21
Ilustración 2-11:	Etapas para el desarrollo de un instrumento virtual.	22
Ilustración 2-12:	Proceso de un conversor AD.....	24
Ilustración 2-13:	Etapas del procesamiento digital de imágenes.....	27
Ilustración 2-14:	Proceso de adquisición de imágenes digitales.	27
Ilustración 2-15:	Generación de una imagen digital.....	29
Ilustración 2-16:	Correlación y convolución de un filtro con un impulso unitario discreto... ..	31
Ilustración 2-17:	Detección de líneas en un circuito electrónico. Arriba: Imagen original e imagen Laplaciana. Abajo: Valor absoluto y valores positivos del Laplaciano.....	35
Ilustración 2-18:	Detección de bordes utilizando la gradiente de intensidad.	36
Ilustración 2-19:	Código de cadena de Freeman.	37
Ilustración 2-20:	Región con dos agujeros.	38
Ilustración 2-21:	Ejemplos de textura: Suavidad, grosor y regularidad.	38

Ilustración 2-22:	Imágenes utilizadas para determinar los momentos invariantes.	39
Ilustración 2-23:	Creación de una red neuronal convolucional.	42
Ilustración 2-24:	Creación de una red neuronal recurrente.	43
Ilustración 3-1:	Concepción de la arquitectura del dispositivo.	45
Ilustración 3-2:	Diagrama de bloques del módulo de generación de vibraciones.	47
Ilustración 3-3:	Diagrama de bloques del módulo de generación de muestras.	48
Ilustración 3-4:	Diagrama de bloques del módulo de adquisición de datos.	48
Ilustración 3-5:	Diagrama de bloques del módulo de procesamiento y visualización.	49
Ilustración 3-6:	Estructura antivibratoria.	51
Ilustración 3-7:	Generador de señales SFG-2104.	51
Ilustración 3-8:	Amplificador de potencia lineal LDS LPA600.	52
Ilustración 3-9:	Vibrador de imán permanente LDS V406.	53
Ilustración 3-10:	Balanza de precisión PS 4500.X2.M.	54
Ilustración 3-11:	Agitador magnético MS7-H550-Pro.	55
Ilustración 3-12:	Cámara NIKON D7500.	56
Ilustración 3-13:	Cámara BASLER acA640.750um.	57
Ilustración 3-14:	Izquierda: Cono de luz led fría. Derecha: Aro de luz led fría.	58
Ilustración 3-15:	Panel de luz led.	58
Ilustración 3-16:	Izquierda: Celda cilíndrica. Derecha: Celda <i>cuasi</i> 2D.	59
Ilustración 3-17:	Entorno de trabajo y visualización de LabVIEW.	60
Ilustración 3-18:	Entorno de trabajo de Google Colab.	61
Ilustración 3-19:	Entorno de trabajo de Visual Studio Code.	62
Ilustración 3-20:	Izquierda: Patrones con la muestra M15-6-80. Derecha: Patrones con la muestra M15-6-140.	65
Ilustración 3-21:	Depósito de la muestra en la celda <i>cuasi</i> 2D.	67
Ilustración 3-22:	Ejemplo de base de datos utilizadas para el entrenamiento de la red neuronal convolucional. Superficie libre (superior) y patrones en el lecho de partículas (inferior).	69
Ilustración 3-23:	Librerías utilizadas para el entrenamiento de la red neuronal convolucional en el entorno de Google Colab.	70

Ilustración 3-24:	Código en Google Colab para cargar los datos de entrenamiento y validación en la red neuronal convolucional.	71
Ilustración 3-25:	Código del modelo de la red neuronal convolucional.....	73
Ilustración 3-26:	Modelo compilado de la ilustración anterior	73
Ilustración 3-27:	Ejemplo del entrenamiento del modelo.	74
Ilustración 3-28:	Código para la evaluación del modelo entrenado.	74
Ilustración 3-29:	Ejemplo de la evolución del modelo a lo largo de las épocas. Izquierda: promedio de validación y entrenamiento. Derecha: pérdidas en el entrenamiento y validación.	75
Ilustración 3-30:	Código para la exportación del modelo ya entrenado.....	75
Ilustración 3-31:	Código para la importación en Visual Studio Code del modelo exportado.	76
Ilustración 3-32:	Código para la predicción de la imagen y retorno de valor hacia Labview	76
Ilustración 3-33:	Panel Frontal del instrumento virtual.....	77
Ilustración 3-34:	Código para el llamado de la función en <i>Python</i> y presentación del resultado de la clasificación.	77
Ilustración 3-35:	Código para la comunicación con los dispositivos hardware.	78
Ilustración 3-36:	Diagrama de flujo del funcionamiento general.....	79
Ilustración 3-37:	Diagrama de flujo de la función “carga de imagen”.....	80
Ilustración 3-38:	Diagrama de flujo de la función “predicción de red neuronal convolucional”.	81
Ilustración 3-39:	Diagrama de flujo de la función “recorte de área de interés”.	82
Ilustración 3-40:	Diagrama de flujo de la función “anillos”.	83
Ilustración 3-41:	Izquierda: Diagrama de flujo de la función “diamantes”. Derecha: Diagrama de flujo de la función “diamantes Python”	84
Ilustración 3-42:	Diagrama de flujo de la función “2D”.	85
Ilustración 4-1:	Acelerómetro ADXL326EB.....	87
Ilustración 4-2:	Tarjeta de adquisición <i>myDAQ</i>	88
Ilustración 4-3:	Programación en LabVIEW para la lectura de datos de frecuencia y aceleración, enviados desde el acelerómetro ADXL326, por medio de la tarjeta de adquisición de datos <i>MyDAQ</i>	89

Ilustración 4-4:	Izquierda: Implementación general del equipo. Derecha: Celda <i>cuasi</i> 2D removible y reemplazable por la celda cilíndrica según sea necesario.	90
Ilustración 4-5:	Nivelación de la celda cilíndrica utilizando un nivel tipo “ojo de buey”....	91
Ilustración 4-6:	Ejemplo de una imagen original tomada con la cámara Nikon D7500 con una resolución de 4K, con un ángulo tipo vista de pájaro o espacial.....	92
Ilustración 4-7:	Los primeros 25 momentos invariantes de Zernike calculados para los patrones generados a 100Hz_1 g con 5, 6 y 7 ml de espesores de la capa de líquido.....	96
Ilustración 4-8:	Base de datos creada para el estudio de patrones formados en la celda cilíndrica.	98
Ilustración 4-9:	Patrones de las ondas de Faraday en un medio ciclo, a 24 Hz con 1.25 g, con un Δh de 6 mm y un hp de 10 mm.....	99
Ilustración 4-10:	Inestabilidad de Faraday con un Δh de 8 ml, a 14 Hz y 1.4 g.....	100
Ilustración 5-1:	Pérdidas de entramiento y validación de la red neuronal convolucional. .	104
Ilustración 5-2:	Graficas de la evolución del entramiento del algoritmo detector de patrones... ..	106
Ilustración 5-3:	Resultados de la detección de patrones sin importar el lugar donde se encuentren.	106
Ilustración 5-4:	Conversión de la imagen original a una imagen en coordenadas polares de un patrón de anillos a una frecuencia f de 40 Hz, con una aceleración ag de 1 g y un nivel espesor de la capa de líquido Δh de 5 ml.....	107
Ilustración 5-5:	Izquierda: Intensidad de píxeles vs distancia en píxeles de la imagen original. Derecha: Intensidad de píxeles vs distancia en píxeles del filtrado de la imagen original utilizando la media móvil.	107
Ilustración 5-6:	Localización de los picos y valles.....	108
Ilustración 5-7:	Obtención de la distancia entre anillos y su valor promedio dpa.	108
Ilustración 5-8:	Ejemplo de imagen utilizada para el análisis de patrones de diamantes. La línea verde es creada manualmente para la obtención de la distancia entre diamantes.	109
Ilustración 5-9:	Imagen de entrada procesada para el conteo de aglomeraciones de partículas, a fin de obtener el número de diamantes con relación a su frecuencia.	110
Ilustración 5-10:	Imagen de salida tras el procesamiento del patrón de la Ilustración 8-5, para la obtención del porcentaje que ocupan las estructuras respecto al área del fondo del recipiente.....	110

Ilustración 5-11: Intensidad de píxeles vs distancia en píxeles, correspondientes a la línea recta creada en la Ilustración 5-8 con el fin de obtener la distancia entre diamantes.	111
Ilustración 5-12: λ (cm) vs f (Hz). Círculos verdes: valores obtenidos a través del instrumento virtual. Barras verticales: desviación típica. Cuadros azules, círculos rojos y línea discontinua: valores tomados de la tesis doctoral del Dr. Diego Barba (Barba 2020).	112
Ilustración 5-13: Relación N_a vs f (Hz) con distintos niveles de espesor de capa de líquido.....	112
Ilustración 5-14: Relación N_d vs f (Hz).	113
Ilustración 5-15: AT_d (%) vs f (Hz) con distintos niveles de espesor de capa de líquido. ...	114
Ilustración 5-16: A_d (mm ²) vs f (Hz) con distintos niveles de espesor de capa de líquido.	114
Ilustración 5-17: d_{pd} (mm) vs f (Hz) con distintos niveles de espesor de capa de líquido..	115
Ilustración 5-18: Longitud de onda λ (mm) vs aceleración a_g (g)	115
Ilustración 5-19: Amplitud de onda A (mm) vs aceleración a_g (g).	116
Ilustración 5-20: Semiciclo de onda formado a 14 Hz y a 1.1 g, con un Δh de 6 mm.	116

ÍNDICE DE ECUACIONES

Ecuación 1: Concentración de glicerina.	63
Ecuación 2: Fracción volumétrica.	63
Ecuación 3: Cálculo de masa de glicerina.	64
Ecuación 4: Masa de glicerina.	64
Ecuación 5: Cálculo de masa de partículas.	64
Ecuación 6: Masa de partículas.	65

ÍNDICE DE SIMBOLOGÍA

C_G	Concentración de glicerina
ϕ	Fracción volumétrica
D_P	Diámetro de las partículas
M_G	Masa de glicerina
M_{H_2O}	Masa de agua destilada
$M_{partículas}$	Masa de partículas
M_{total}	Masa total de la muestra
f	Frecuencia
a_g	Aceleración
Δh	Espesor de la capa de líquido sobrenadante
h_p	Espesor de la capa de partículas
λ	Longitud de onda
N_a	Número de anillos
d_{pa}	Distancia promedio entre anillos
N_d	Número de diamantes
A_d	Área de cada diamante
A_{Td}	Área total de los diamantes
d_{pd}	Distancia promedio entre diamantes
A	Amplitud de onda
cd/m^2	Candelas por metro cuadrado
mV/g	Milivoltios por gravedad
Hz	Hertz
g	Gravedad
gr	Gramos
ml	Mililitros
ms	Milisegundos
mm	Milímetros
μm	Micrómetros

ÍNDICE DE ANEXOS

Anexo A: Dimensiones de la estructura antivibratoria en mm.

Anexo B: Ficha de datos del generador de señales SFG-2104.

Anexo C: Ficha de datos del amplificador de potencia lineal LDS LPA600.

Anexo D: Ficha de datos del vibrador de imán permanente LDS V406.

Anexo E: Ficha de datos de la balanza de precisión PS 4500.X2.M.

Anexo F: Ficha de datos del agitador magnético MS7-H550-Pro.

Anexo G: Ficha de datos de la cámara NIKON D7500.

Anexo H: Ficha de datos de la cámara BASLER acA640.750um.

Anexo I: Ficha de datos del acelerómetro ADXL326EB.

Anexo J: Ficha de datos de la tarjeta de adquisición *myDAQ*.

Anexo K: Código del procesamiento en Python.

RESUMEN

El objetivo del presente trabajo de integración curricular fue desarrollar un instrumento virtual experimental basado en procesamiento de imágenes que permita determinar gráficamente estructuras granulares generadas por inestabilidades hidrodinámicas formadas al modificar las variables frecuencia de excitación, aceleración y espesor de la capa de líquido sobrenadante. Por esta razón, mediante técnicas de revisión bibliográfica, se examinaron los fundamentos teóricos y conceptuales de todo lo relacionado a la influencia de inestabilidades hidrodinámicas en la formación de estructuras granulares, incluyendo temas relacionados al procesamiento de imágenes. El proyecto consta de 4 módulos, cada uno con una función diferente y complementaria. El módulo de generación de vibraciones se encarga de enviar, por medio de un generador de funciones y un amplificador de potencia, una onda senoidal amplificada hacia el *shaker*, para producir vibraciones verticales que afecten a las celdas contenedoras. El módulo de generación de muestras tiene como objetivo medir porcentajes y pesos del líquido portador y las partículas, hasta obtener una muestra que se adapte de mejor manera a las vibraciones verticales. En el módulo de adquisición de datos, se encuentran las cámaras fotográficas y la iluminación necesaria para la captura de imágenes de los patrones formados en la superficie libre de líquido y en el lecho de partículas. El módulo de procesamiento y visualización finiquita con el desarrollo del instrumento virtual, donde se explica paso a paso el proceso para el diseño y desarrollo del mismo, utilizando varias herramientas de programación como *Python* y de aprendizaje automático como las redes neuronales convolucionales. Finalmente, se muestran las pruebas realizadas del instrumento, con sus respectivos resultados, para obtener datos de alta calidad que sean relevantes y precisos en el estudio de las estructuras granulares generadas a partir de inestabilidades hidrodinámicas, así como resultados que muestren la fiabilidad y validación del instrumento virtual.

Palabras clave: <ESTRUCTURA GRANULAR >, <INESTABILIDAD HIDRODINÁMICA >, <APRENDIZAJE AUTOMÁTICO >, <INSTRUMENTO VIRTUAL >, <ACELERACIÓN >, <FRECUENCIA >, < ESPESOR DE LA CAPA DE LÍQUIDO >.



ABSTRACT

This curricular integration research project aimed to develop an experimental virtual instrument based on image processing that allows graphically determining granular structures generated by hydrodynamic instabilities formed by modifying the variables excitation frequency, acceleration, and thickness of the overlying liquid layer. For this reason, through literature review techniques, the theoretical and conceptual foundations were examined of everything related to the influence of hydrodynamic instabilities in the formation of granular structures, including topics related to image processing. The project consists of 4 modules, each with a different and complementary function. The vibration generation module is responsible for sending an amplified sine wave toward the shaker through a function generator and a power amplifier to produce vertical vibrations that affect the containing cells. The sample generation module aims to measure the percentages and weights of the carrier liquid and the particles to obtain a sample that best fits the vertical vibrations. In the data acquisition module are the photographic cameras and the necessary lighting for capturing images of the patterns formed on the free liquid surface of and on the particle bed. The processing and visualization module conclude with the development of the virtual instrument, where the step-by-step process for the design and development of the instrument is explained, using various programming tools such as Python and machine learning like convolutional neural networks. Finally, the tests are conducted on the instrument, along with their respective results, to obtain high-quality data that is relevant and accurate in the study of granular structures generated from hydrodynamic instabilities, as well as results that show the reliability and validation of the virtual instrument.

Keywords: <GRANULAR STRUCTURE>, <HYDRODYNAMIC INSTABILITY>, <MACHINE LEARNING>, <VIRTUAL INSTRUMENT>, <ACCELERATION>, <FREQUENCY>, <LIQUID LAYER THICKNESS>.



INTRODUCCIÓN

En el campo de la física y la mecánica de fluidos, la interacción entre materiales granulares y flujos hidrodinámicos ha capturado un interés creciente debido a su profunda relevancia en distintos campos de la ciencia y la ingeniería. El fenómeno complejo y fascinante de la formación y evolución de estructuras granulares a raíz de inestabilidades hidrodinámicas posee implicaciones que abarcan diversas aplicaciones, desde la geofísica hasta la industria energética y el diseño de sistemas de transporte.

El presente trabajo de integración curricular hace parte del grupo de investigación GIICYT, perteneciente a la Escuela Superior Politécnica de Chimborazo, y se adentra en la creación de un instrumento virtual experimental, cuyo propósito central radica en la representación gráfica y el análisis de las estructuras granulares que surgen debido a inestabilidades hidrodinámicas. Dichas inestabilidades, originadas por la fuerza de vibraciones verticales, tienen la capacidad de desencadenar patrones de movimiento que son complejos y enigmáticos, brindando información crucial sobre los patrones formados y las características de aglomeraciones de partículas, como valores de frecuencia, aceleración y espesor de la capa de líquido de un patrón de partículas en específico.

El documento se encuentra dividido en varios capítulos. El primer capítulo abarca los fundamentos teóricos y aplicativos como punto de partida en el desarrollo del proyecto, así como los antecedentes, planteamiento del problema, justificación teórica, justificación aplicada y objetivos a alcanzar en el desarrollo del trabajo de integración curricular.

En el capítulo dos se presenta el Marco Teórico, donde se redactan conceptos y definiciones generales relacionados al tema del presente proyecto, por medio de una revisión bibliográfica. El tercer capítulo abarca el Marco Metodológico, donde se analizan los requerimientos del sistema y se desarrolla el proyecto mediante módulos representados por diagramas de bloques, además, se explica paso a paso el proceso para el diseño y el desarrollo del instrumento virtual que permita determinar gráficamente estructuras granulares generadas por inestabilidades hidrodinámicas.

Finalmente, en los capítulos cuatro y cinco, se presentan las pruebas y los resultados obtenidos del desarrollo del proyecto, junto con el análisis de las imágenes de patrones tanto en el lecho de partículas como en la superficie libre de líquido, presentando los parámetros fundamentales que caracterizan a cada patrón, como la frecuencia, aceleración y espesor de la capa de líquido.

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA

En este capítulo, se examinan diversos estudios con el propósito de identificar el origen del problema. Se proporcionan las justificaciones pertinentes que fundamentan la formulación del problema, además de esbozar los objetivos que guiarán el desarrollo del trabajo de titulación.

1.1 Antecedentes

Los patrones de partículas producidos por inestabilidades hidrodinámicas se conocen como estructuras granulares y pueden tener diferentes formas y tamaños, y su evolución y comportamiento dependen de varios factores, como la geometría del sistema, la concentración de partículas y las propiedades físicas del líquido. Estudiar estas estructuras puede ser útil para entender el comportamiento de sistemas granulares en diferentes contextos, como la ingeniería de fluidos, la geología o la biología (Chen et al. 2014).

El movimiento descendente de partículas sólidas que se encuentran presentes en un líquido por acción de la gravedad se conoce como sedimentación, ya que las partículas presentan mayor densidad que el líquido portador (Barba 2020).

Entre los estudios de estructuras granulares se encuentran estudios enfocados al área de la ingeniería civil, como el presentado por Mikheil Kharbedia en su tesis doctoral sobre la hidrodinámica superficial en materia blanda: turbulencia de ondas y organización emergente. Publicada en el año de 2020, pero estos tipos de estudios son enfocados en un ambiente seco no húmedo o líquido (Kharbedia 2020).

La inestabilidad de una capa de fluido en un sistema vibratorio, fue estudiado por primera vez por Faraday en el año de 1831, que presentó un estudio sobre inestabilidad hidrodinámica enfocado a líquidos con una vibración vertical (Faraday 1831). Posteriormente, en los años de 2003 los científicos Wright y Saylor, publicaron un estudio experimental de patrones que habían sido provocados por la inestabilidad de Faraday, utilizando partículas de talco y agua desionizada. Para el año 2005, se realizó un estudio sobre dichos experimentos, pero con un enfoque matemático (Saylor y Kinard 2005).

En el año 2015, se llevó a cabo un estudio de compactación de suspensiones granulares inducidas por vibraciones. Donde se observó el comportamiento dinámico de las suspensiones granulares

mediante procesamiento de imágenes digitales y la evolución de la compactación en distintos estados de movimiento o de relajación (Richter et al. 2015).

Recientemente, una pareja de ingenieros realizó un proyecto ambicioso enfocado a la generación de un sistema vibratorio automatizado mediante una interfaz HMI, donde se observa la creación de una estructura antivibratoria capaz de soportar movimientos bruscos, permitiendo que las vibraciones producidas por el equipo vibratorio no afecten el entorno externo de las celdas contenedoras, mostrando patrones en la superficie libre de líquido más compactos (Jaramillo y Marcatoma 2023).

Un estudio más enfocado a la formación de patrones de partículas debido a inestabilidades hidrodinámicas (vibraciones verticales) y al procesamiento de dichos patrones, es el trabajo realizado por el Dr. Diego Barba, en su tesis doctoral: Dinámica de suspensiones concentradas sometidas a vibración mecánica; donde se extraen diversos patrones formados en el lecho de partículas contenidas en recipientes de distintas formas, para observar el comportamiento de las mismas al ser sometidas a vibraciones verticales. Además, se llevó a cabo el procesamiento de las imágenes obtenidas para visualizar el progreso de los patrones a medida que pasa un determinado tiempo (Barba 2020).

En la década de 1960, se desarrolló el primer algoritmo de procesamiento de imágenes, que se utilizaba para mejorar la calidad de las imágenes mediante el uso de filtros. A medida que la tecnología avanzaba, se desarrollaron nuevos algoritmos y técnicas de procesamiento de imágenes, como la segmentación y la reconstrucción de imágenes (Gonzalez y Woods 2008).

Hoy en día, el procesamiento de imágenes es un campo muy importante en la informática, y se utiliza en una amplia variedad de aplicaciones, como la medicina, la vigilancia, la industria y el entretenimiento. Se están desarrollando constantemente nuevas técnicas y algoritmos de procesamiento de imágenes, lo que permite mejorar aún más la calidad y la precisión de las imágenes procesadas (Gonzalez y Woods 2008).

1.2 Planteamiento del problema

¿Existe algún dispositivo que permita determinar gráficamente las estructuras granulares generadas por inestabilidades hidrodinámicas?

1.2.1 Sistematización del problema

¿Cómo se piensa obtener la información pertinente para el desarrollo del instrumento virtual experimental?

¿Qué alcance tendrá el instrumento virtual que se va a desarrollar?

¿Cómo se puede obtener los patrones de partículas generados por inestabilidades hidrodinámicas?

¿Como se puede saber que el dispositivo funciona correctamente y es capaz de ayudar a futuras investigaciones?

1.3 Justificación Teórica

Las inestabilidades hidrodinámicas que generan estructuras granulares diluidas en cualquier líquido o fluido, serán determinadas gráficamente mediante un instrumento virtual experimental.

Las estructuras granulares son patrones de formación que se producen en sistemas granulares, como el flujo de partículas en un lecho fluidizado o el movimiento de arena en una playa. Pueden ser generadas por inestabilidades hidrodinámicas, que son desequilibrios en el flujo de un líquido, cuyo estudio es muy necesario debido a su gran importancia en determinados fenómenos físicos en todas las escalas de la materia (Gómez 2000).

Algunos ejemplos donde dichas estructuras juegan un papel muy importante son la catálisis en ingeniería química, el almacenamiento de cereales en silos de agricultura, el transporte de carbón en la industria mineral y el de pastillas en la industria farmacológica (Gómez 2000).

En la hidrodinámica de los medios granulares diluidos, se determinó que existe un régimen de flujo rápido, en el cual las partículas que constituyen la estructura granular se mueven libremente excepto cuando colisionan entre sí, es decir, se comportan de manera parecida a las partículas de un fluido granular (Richter et al. 2015).

Una manera de estudiar las inestabilidades hidrodinámicas es mediante las ondas de Faraday, que muestran el nivel de aceleración que se necesita para llevar un patrón en forma de anillos de Faraday a su punto de inestabilidad. La generación de patrones inestables depende principalmente de los valores de frecuencia y aceleración provenientes de las vibraciones, así como el nivel de líquido y de partículas presentes en el recipiente contenedor (Barba et al. 2019).

En cuanto a la manipulación de estructuras granulares, existen trabajos dedicados al ensamblaje de materiales a microescala utilizando partículas macroscópicas, debido a la gran exigencia de diversos campos tecnológicos, como la ingeniería de tejido, donde la organización de las células y microportadores celulares en arquitecturas 3D imitan funciones importantes y específicas de tejido. Utilizando patrones estacionarios en la superficie libre de líquido sometidos a vibraciones, se localizan partículas flotantes, evidenciando que puede existir una reconfiguración rápida de los patrones en intervalos de tiempo muy cortos, para obtener una amplia gama de patrones flotantes (Chen et al. 2014).

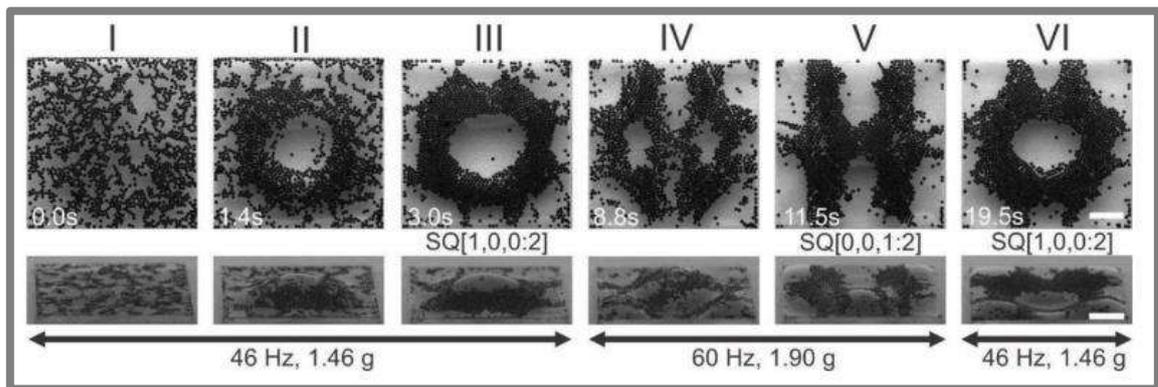


Ilustración 1-1: Proceso dinámico del montaje de partículas flotantes en plantillas líquidas al variar parámetros de frecuencia y aceleración. El montaje empieza en cero hasta adquirir una forma de anillo, pasando por una estructura en forma de “H” para volver a su forma de estructura de anillo.

Fuente: (Chen et al. 2014).

El estudio de dichas estructuras se obtiene mediante un procesamiento exhaustivo de imágenes, de forma que se logre llegar a un estudio profundo de las estructuras. El procesamiento de imágenes muestra el comportamiento de las estructuras granulares con el paso del tiempo, resultando en diversas comparaciones realizadas entre distintos parámetros, que pueden ser valores de aceleración vs frecuencia, o frecuencia vs longitud de onda (Barba 2020).

El procesamiento digital de imágenes también conocido como manejo de imágenes por computadora, es un campo de la informática que se ocupa de la adquisición, manipulación y análisis de imágenes digitales. Esto incluye técnicas como el filtrado, la segmentación y la reconstrucción de imágenes (Gonzalez y Woods 2008).

Desde una perspectiva de ingeniería, el determinar gráficamente estructuras granulares que se generan debido a inestabilidades hidrodinámicas es de gran importancia, ya que surge la necesidad de utilizar imágenes con la mayor calidad posible, teniendo en cuenta que las

estructuras granulares están diluidas en líquido, lo que dificulta su manipulación y la obtención de dichas estructuras de manera ordenada dentro del líquido portador, es así que el objetivo de llevar a cabo un instrumento virtual para determinar gráficamente patrones que se forman en la superficie libre de líquido y en el lecho de partículas, ayudará al avance tecnológico no solo en Ecuador, sino también en América Latina.

El objetivo de la instrumentación virtual es generar un nuevo enfoque de la instrumentación tradicional que incorpora nuevos elementos y mejores características de sistema agregando mejoras tecnológicas en innovaciones científicas, para reducir los costos de producción de los distintos sistemas y aprovechar el uso correcto de nuevas tecnologías, con el fin de obtener una mayor acogida en el mercado y nuevas plataformas de diseño extremadamente eficientes y flexibles (Denning 2001).

1.4 Justificación Aplicativa

Para consulta de textos literarios sobre estructuras granulares generadas por inestabilidades hidrodinámicas, procesamiento de imágenes y desarrollo de software, se utilizan herramientas de internet, como páginas web, libros electrónicos, artículos científicos, etc.

Una vez obtenida la información necesaria, se comienza el estudio que delimite el alcance del instrumento virtual con base en comportamientos físicos de las estructuras granulares, en limitaciones ópticas y en obtención de información a través del procesamiento de imágenes.

Las estructuras granulares se obtienen de un sistema vibratorio que emplea un vibrador electromagnético, encargado de producir vibraciones en el entorno de una celda contenedora, donde están contenidas las partículas de poliestireno insoluble junto con el líquido portador. Además, el sistema cuenta con una corona de luces led alrededor del recipiente y una superficie semireflectante en la parte superior que proyecta la luz recibida desde un cañón de luz led fría, para iluminar el interior del recipiente.

Como primer paso, se debe comprobar y validar que la mesa anti vibratoria soporte los movimientos resultantes de las vibraciones producidas por el dispositivo vibrador, que debe generar vibraciones y aceleraciones correctas. Además, los ángulos de inclinación de las celdas deben ser cercanos a 0° , la iluminación debe ser adecuada para la toma de fotografías y se debe tomar en cuenta otros parámetros externos que influyan en los experimentos para que la toma de datos, fotografías y otros, sean lo más acertadas posibles.

Para el desarrollo del instrumento virtual se cuenta con dos cámaras digitales, conectadas mediante comunicación USB hacia una computadora que contendrá el instrumento virtual, con las cámaras se realizan las fotografías, que serán enviadas al software para su posterior análisis, también existe la posibilidad de cargar imágenes almacenadas en el ordenador, siempre y cuando tengan el formato adecuado y las características esenciales determinadas en el estudio exploratorio que busca el alcance del instrumento virtual. Una vez realizado el análisis de la imagen, se entrega información útil sobre las estructuras granulares generadas, como la frecuencia, aceleración y espesor de la capa de líquido.

El instrumento virtual será desarrollado mediante procesamiento digital de imágenes, con altas probabilidades de usar software libre como MATLAB, LabVIEW, Python, entre otros.

Una vez desarrollado el instrumento virtual, se realizará una validación que verifique el cumplimiento de estándares internacionales en cuanto a errores, tipo de formato, unidades, texto, etc. Con el fin de permitir su utilización en investigaciones futuras.

1.5 Objetivos

1.5.1 Objetivo general

Desarrollar un instrumento virtual experimental basado en procesamiento de imágenes que permita determinar gráficamente estructuras granulares generadas por inestabilidades hidrodinámicas formadas al modificar las variables frecuencia de excitación, aceleración y espesor de la capa de líquido sobrenadante.

1.5.2 Objetivos específicos

- Ejecutar actividades de detección, obtención y consulta de literatura pertinente que permita desarrollar un instrumento virtual experimental, para detectar estructuras granulares generadas por inestabilidades hidrodinámicas, con el fin de extraer y recopilar información de interés.
- Establecer un estudio exploratorio que permita establecer el alcance del trabajo de integración curricular.
- Diseñar el instrumento virtual experimental que permita obtener patrones de partículas asociados a inestabilidades hidrodinámicas generadas en un líquido portador, tomando

como referencia la frecuencia de excitación, la aceleración y el espesor de la capa de líquido sobrenadante.

- Validar el instrumento virtual experimental con estándares internacionales para que brinde robustez a futuras investigaciones derivadas del tema de investigación.

1.6 Metodología

Para el desarrollo del proyecto es necesario utilizar de manera combinada los siguientes métodos de investigación:

Los métodos teóricos, como la revisión bibliográfica de documentos y estudios relacionados con inestabilidades hidrodinámicas y la formación de estructuras granulares, para definir los requerimientos específicos en el desarrollo del instrumento virtual, y establecer los parámetros clave que se desean visualizar y analizar. Además, mediante pruebas y validaciones, se requiere comparar los resultados obtenidos del instrumento virtual con datos experimentales obtenidos del acelerómetro.

La experimentación empírica, para realizar las pruebas necesarias y observar de manera directa la generación de estructuras granulares dentro de las celdas contenedoras, y de esta manera recopilar datos en tiempo real.

CAPÍTULO II

2. MARCO TEÓRICO

2.1 Estructuras granulares

Las estructuras granulares son sistemas o grupos compuestos de partículas sólidas de diferentes tamaños que interactúan entre sí por medio de fuerzas mecánicas. Dichas estructuras se pueden observar tanto en la naturaleza, como en la creación de estructuras humanas como carreteras o edificios (Feng y Miao 2009).

Las estructuras granulares pueden ser vistas desde perspectivas de multivista y multinivel, aunque la noción se centra en la estructura granular de multinivel, la cual consiste en gránulos interconectados que interactúan entre sí, donde existen familias granulares definidas como niveles de granularidad diferente y estructuras múltiples parcialmente ordenadas conocidas como estructuras jerárquicas. Estos aspectos son el resultado de una estructura, interpretación, representación y descripción de un sistema o mundo real (Feng y Miao 2009).

2.1.1 *Propiedades físicas de las estructuras granulares*

El estudio de las propiedades físicas de las estructuras granulares es de gran relevancia, ya que permite conocer y entender su comportamiento mecánico y su respuesta a estímulos externos.

2.1.1.1 *Tamaño y forma de las partículas*

El tamaño y la forma de las partículas que componen una estructura granular influyen directamente en su porosidad, densidad y resistencia mecánica. Por ende, partículas más grandes tienen la tendencia a formar estructuras más porosas y menos densas, mientras que las más pequeñas generan estructuras más densas y compactas. Asimismo, la forma de las partículas afecta su capacidad de apilamiento y comprensión (Ma et al. 2019).

2.1.1.2 *Empaquetamiento de partículas*

La importancia del empaquetamiento de partículas (lineal o no lineal) deriva en la determinación de la porosidad y densidad de una estructura granular. El tipo de empaquetamiento que se realice o la forma en la que las partículas se ordenan, influye en su capacidad de resistir fuerzas externas y en su estabilidad (Liu et al. 2019).

2.1.1.3 Porosidad y densidad

La influencia de la porosidad y la densidad de una estructura granular se presenta en su capacidad de absorción de líquidos, gases, y su respuesta a las fuerzas externas. La densidad hace referencia a la masa de las partículas por unidad de volumen, mientras que la porosidad es la fracción de espacio vacío en una estructura granular. Existen cinco tipos de interacciones que rigen la densidad para un buen empaquetamiento granular mencionado en el apartado anterior, que son los efectos de ocupación y de pared de partículas gruesas, así como los efectos de aflojamiento, acuñamiento y relleno de partículas finas (Liu et al. 2019).

2.1.1.4 Propiedades mecánicas

Se consideran propiedades mecánicas a la capacidad de una estructura granular para soportar fuerzas externas, a su habilidad para deformarse y su resistencia a fracturarse. Estas propiedades están influenciadas por su grado de compactación o empaquetamiento y la cantidad de porosidad presente, así como por las propiedades físicas de las partículas que componen la estructura granular (Zhang y Yin 2021).

2.1.2 Tipos de estructuras granulares

Según Feng y Miao (2009) existen distintos tipos de estructuras granulares que varían dependiendo de la forma en que las partículas interactúan y se organizan entre sí. Los tipos de estructuras granulares más comunes son:

2.1.2.1 Estructuras granulares amorfas

Se refieren a estructuras granulares que no presentan una ordenación espacial regular. Las partículas se organizan aleatoriamente, dando como resultado una estructura porosa y sin forma definida, tal y como se observa en la ilustración 2-1. El ejemplo más común de este tipo de estructura granular, son las arenas de sílice altamente porosas (Feng, Q y Miao 2009).

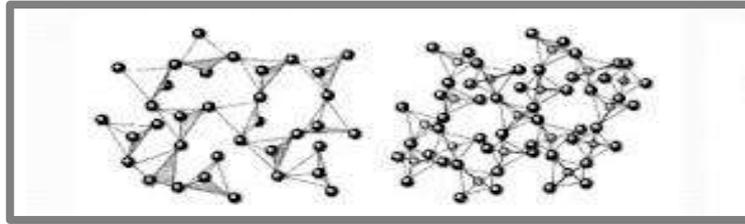


Ilustración 2-1: Ejemplos de estructuras amorfas: Estructura de silicio y estructura de cuarzo.

Fuente: (Zapata 2022).

2.1.2.2 Estructuras granulares cúbicas

En este caso, las partículas se organizan con una disposición cúbica. Los granos se organizan en filas y columnas, lo que genera una estructura altamente ordenada. Un ejemplo de estructuras granulares cúbicas son las estructuras cristalinas que poseen algunos metales (Ilustración 2-2) (Feng, Q y Miao 2009).

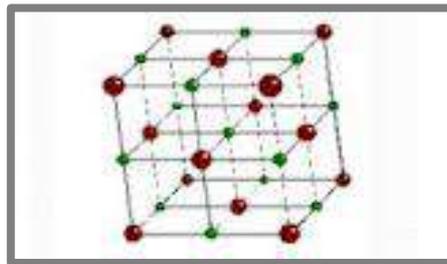


Ilustración 2-2: Estructura cristalina.

Fuente: (González 2021).

2.1.2.3 Estructuras granulares en capas

Como su nombre lo dice, son estructuras granulares que se agrupan en capas de partículas ordenadas en distintas direcciones. Este tipo de estructuras se encuentran comúnmente en los suelos y en depósitos sedimentarios (Feng, Q y Miao 2009).

2.1.2.4 Estructuras granulares en cadena

Las partículas se organizan a manera de cadena, es decir, en este tipo de estructuras granulares las partículas se unen mediante enlaces covalentes para formar estructuras filamentosas. El ejemplo más notorio de estructura granular en cadena, son las fibras de ADN (Ilustración 2-3) (Feng, Q y Miao 2009).

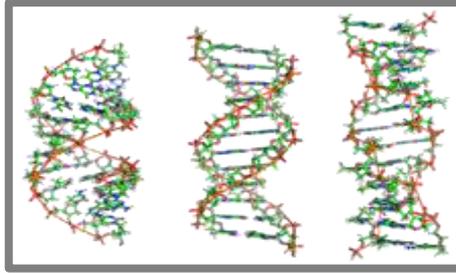


Ilustración 2-3: Fibras de ADN.

Fuente: (Reyes y Hinojosa 2000).

2.1.2.5 *Estructuras granulares con fuerzas activas*

Estas estructuras presentan un comportamiento dinámico en presencia de fuerzas activas que interactúan entre las partículas, como la compresión o la fricción. Ejemplos de estas estructuras son los agregados (Feng, Q y Miao 2009).

2.2 **Suspensión de partículas**

La suspensión de partículas abarca muchas áreas como la química, la física, la biología y la ingeniería. Se dice que la mezcla heterogénea de partículas sólidas o líquidas no solubles dispersas en algún medio líquido o gaseoso es una suspensión de partículas (Barba 2020).

Las suspensiones de partículas pueden clasificarse en suspensiones coloidales, donde las partículas tienen un tamaño menor a $1\mu\text{m}$, y en suspensiones macroscópicas, en las que el tamaño característico de las partículas es mayor a $1\mu\text{m}$ (Barba 2020).

2.2.1 *Propiedades de las partículas*

Tamaño

Esta propiedad que tienen las partículas es la más fácil de medir, por lo que es una de las propiedades físicas más importantes en diversos casos prácticos. En las partículas que conforman suspensiones macroscópicas, su rango de tamaño está comprendido desde $1\mu\text{m}$ hasta $100\mu\text{m}$. Los efectos de inercia se desprecian frente a efectos viscosos que presente el líquido, debido a que el número de Reynolds para el flujo de las partículas es demasiado pequeño en comparación a la unidad (Barba 2020).

Forma

La forma de las partículas se relaciona con la configuración espacial de las mismas, independientemente del tamaño o la composición. Desde el punto de vista de simplicidad, la manera más efectiva de encontrar el factor forma es: $S.F = \frac{k_1}{\sqrt{k_2 k_3}}$ (Barba 2020).

Donde $S.F$ es el factor forma y es un valor adimensional que equivale a la unidad cuando se trata de una esfera, mientras que k_1 , k_2 y k_3 son las dimensiones del mayor, intermedio y menor eje perpendiculares entre sí de la partícula (Barba 2020).

Densidad

Esta propiedad depende fundamentalmente de la composición de las partículas, es decir, depende del tamaño y de la forma de las mismas. La densidad se mide en g/cm^3 y en muchos casos se usa la densidad relativa, que es la razón entre la densidad del agua y la densidad de las partículas (Barba 2020).

2.3 Inestabilidades

Las inestabilidades son fenómenos que ocurren cuando un sistema se encuentra en un estado inestable y experimenta cambios impredecibles y no lineales en su comportamiento. Las inestabilidades pueden ocurrir en diferentes tipos de sistemas, como mecánicos, eléctricos, químicos y de fluidos (Barba 2020).

2.3.1 Tipos de inestabilidades

En la física, hay varios tipos de inestabilidades, incluyendo:

Inestabilidades dinámicas: ocurren cuando un sistema es propenso a oscilar o vibrar. Por ejemplo, un péndulo puede ser inestable si se le aplica una fuerza desbalanceada, lo que puede hacer que se vuelque o se desplace de su posición de equilibrio (Zhang y Yin 2021).

Inestabilidades térmicas: ocurren cuando hay una diferencia de temperatura en un sistema. Por ejemplo, una taza de café caliente en una habitación fría puede ser inestable debido a la diferencia de temperatura. Las ondas de choque son ondas transversales que se propagan de temperatura entre la taza y el aire (Zhang y Yin 2021).

Inestabilidades gravitacionales: ocurren cuando un cuerpo celeste, como un planeta o una estrella, se encuentra en una posición inestable debido a la gravedad de otro cuerpo celeste. Por ejemplo, la Luna es inestable debido a la gravedad de la Tierra (Zhang y Yin 2021).

2.4 Inestabilidades hidrodinámicas

Las inestabilidades hidrodinámicas son aquellas que involucran el movimiento y el comportamiento de los fluidos. Estos pueden incluir líquidos o gases, y pueden ser causados por factores como la fricción, la viscosidad y la elasticidad de los fluidos (Zhang y Yin 2021).

2.4.1 Inestabilidad de Faraday

Un ejemplo de inestabilidades hidrodinámicas son las ondas de Faraday, que no son más que perturbaciones excitadas paraméricamente que aparecen en una superficie líquida, cuando dicha superficie se encuentra vibrando verticalmente. Además, estas ondas generan una gran variedad de patrones bidimensionales (Francois et al. 2015).

Faraday realizó un estudio profundo de dichas perturbaciones en líquidos sometidos a vibración vertical, adquiriendo gran influencia en sistemas no lineales en desequilibrio. La Ilustración 2-4 muestra que esta inestabilidad es de tipo subarmónico (mitad de la frecuencia de excitación) y se genera cuando la aceleración de vibración supera cierto valor de umbral. Esto resulta en que los efectos inerciales de vibración superan a los efectos restauradores provistos por la tensión superficial y la gravedad, dando como resultado un patrón de ondas estacionarias en la superficie libre del líquido (Douady 1990).

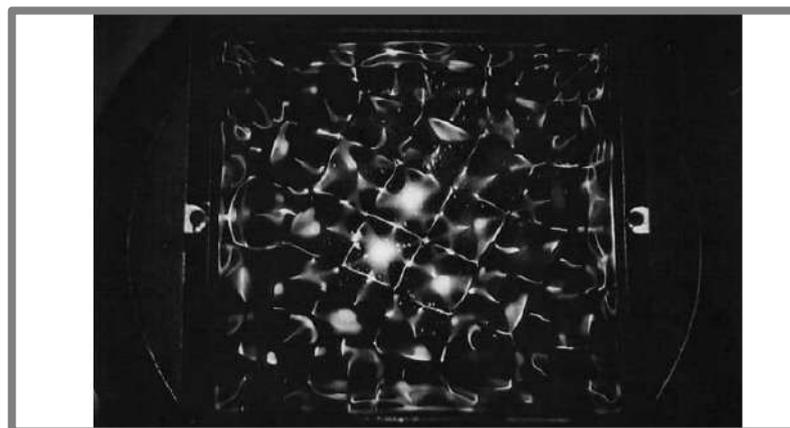


Ilustración 2-4: Patrones generados en la inestabilidad de Faraday utilizando agua a una frecuencia de 70 Hz.

Fuente: (Douady 1990).

La Ilustración 2-5 utiliza una técnica de imágenes de luz difusiva, donde los picos y valles son manchas blancas u oscuras. Se detectan los máximos de ondas locales, es decir, los puntos blancos dentro de manchas oscuras, y continúa su movimiento mediante técnicas PTV. Se observa también el desplazamiento horizontal cuadrático medio de los oscilones alejándose de la posición inicial en función del tiempo (Francois et al. 2015).

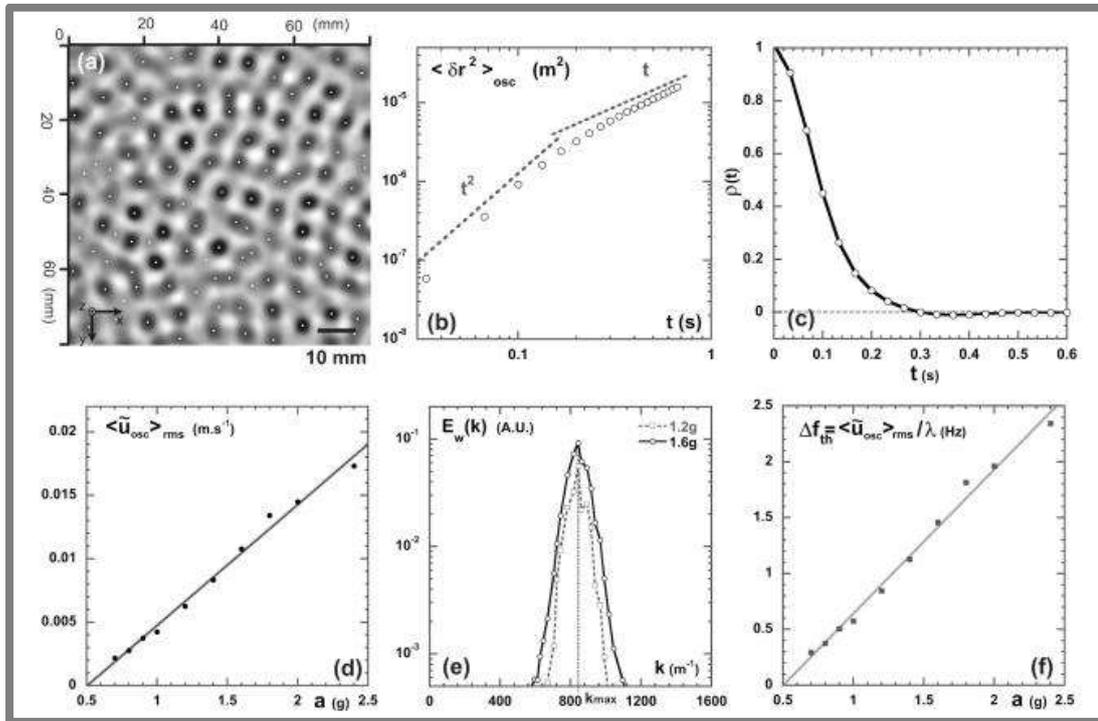


Ilustración 2-5: Ondas de Faraday usando la técnica de luz difusa.

Fuente: (Francois et al. 2015).

2.5 Instrumentación tradicional

La instrumentación se define como el conjunto de sistemas que se utilizan para el sentido y procesamiento de información o datos provenientes de variables físicas y químicas, mediante las cuales se lleva a cabo el monitoreo y control de procesos, empleando dispositivos electrónicos (Juárez 2013).

Los instrumentos convencionales pueden ser elementos eléctricos, mecánicos, neumáticos, hidráulicos, electrónicos, entre otros, capaces de medir variables físicas y ejercer acciones de control mediante actuadores, para transmitir información desde un punto hacia otro e interpretar señales que mediante algoritmos deciden el comportamiento del sistema (Mendiburu 2006).

Mendiburu (2006, pp. 7-8) señala que algunas características de los instrumentos de medición, son las siguientes:

- **Campo de medida o Rango:** Conjunto de valores en la escala de medición comprendido entre los límites superior e inferior que denota la capacidad del equipo (Mendiburu 2006).
- **Alcance (Span):** Diferencia algebraica entre el valor superior e inferior del rango del equipo (Mendiburu 2006).
- **Exactitud:** Capacidad del instrumento para medir el valor real (Mendiburu 2006).
- **Precisión:** El instrumento muestra el valor real según su capacidad para hacerlo (Mendiburu 2006).
- **Zona muerta:** Intervalo de valores que no producen respuesta alguna (Mendiburu 2006).
- **Umbral:** Nivel mínimo necesario para que el instrumento indique la medida (Mendiburu 2006).

2.6 Instrumentación virtual

El concepto de Instrumento virtual (IV) es un instrumento no real, capaz de ejecutarse en una computadora con funciones definidas por software. De esta manera, se define la instrumentación virtual como un sistema de medición, análisis y control de señales físicas desde una computadora personal o PC utilizando instrumentos virtuales (Chacón 2022, p.80).

Un instrumento virtual puede mostrar en pantalla dos tipos de elementos gráficos:

- **Elementos indicadores** como salidas que muestran información al usuario, por ejemplo, gráficas, LED, indicadores de aguja, etc. (Mendiburu 2006).
- **Elementos de control**, los cuales aceptan la entrada del usuario, actuando sobre el comportamiento del programa o sistema, algunos ejemplos son los interruptores, cajas de entrada de texto, pulsadores, etc. (Mendiburu 2006).

2.6.1 Instrumentación virtual vs instrumentación tradicional

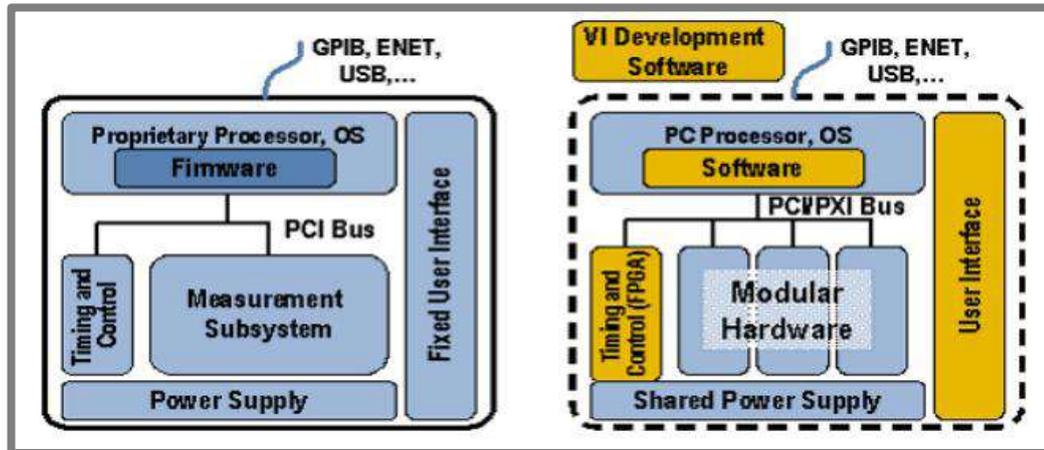


Ilustración 2-6: Izquierda: Instrumentos tradicionales. Derecha: Instrumentos virtuales.

Fuente: (Juárez 2013, p. 4).

La instrumentación virtual y la instrumentación convencional se comparan en el proceso de sensado, ya que en la instrumentación tradicional un instrumento de sensado recoge señales y proporciona medidas a un dispositivo controlador, mientras que en la instrumentación virtual la computadora recoge dichas señales y las procesa mediante el hardware necesario. Los dos tipos de instrumentación son comparados en la Tabla 2-1, y la Ilustración 2-6 muestra las diferencias entre ambos (Mendiburu 2006).

Tabla 2-1: Instrumentos tradicionales vs Instrumentos virtuales.

Instrumento convencional	Instrumento virtual
Se define por el fabricante.	Se define por el usuario.
La clave es el Hardware.	La clave es el software.
Alto costo de mantenimiento y bajas economías de escala.	Bajos costos de mantenimiento y altas economías de escala.
Incorporación lenta de nuevas tecnologías.	Incorporación rápida de nuevas tecnologías gracias a la PC.
Su arquitectura es “cerrada”.	Su arquitectura es “abierta”.
Tiene un alto costo de funcionalidad.	Tiene un bajo costo de funcionalidad y variedad de funciones.
Conectividad limitada y funcionalidad específica.	Conectividad amplia y funcionalidad ilimitada.

Fuente: (Mendiburu 2006).

Realizado por: Argos J; Granizo L, 2023.

Las ventajas que tiene la instrumentación virtual respecto a la instrumentación tradicional son varias, como la flexibilidad, la reducción de costos, el software de instrumentación virtual, entre otras (Serrano 2017, p. 12).

2.6.2 Herramientas de programación y desarrollo de instrumentación virtual

Para el desarrollo de un instrumento virtual es necesario el uso de un entorno adecuado para gestionar el proceso requerido en el sistema de instrumentación, así como un software adecuado para llevar a cabo la programación (Mánuel et al. 2002).

Los lenguajes de programación dirigidos a la instrumentación pueden estar basados en conjuntos de comandos creando líneas de código como, por ejemplo, C/C++, Java, Python y Matlab, por mencionar los más populares. Sin embargo, los lenguajes de tipo gráfico ofrecen mayores ventajas en comparación con los lenguajes tradicionales basados en texto, donde el lenguaje gráfico (lenguaje G) más utilizado es LabView (Pineda 2013).

Según Mánuel, Biel, Olivé y Sánchez (2002, p. 227), se exigen dos características fundamentales a cualquier software de instrumentación virtual, que son:

- Debe ser capaz de controlar todo el sistema, desde instrumentos GPIB, VXI, hasta comunicación serie (Mánuel et al. 2002).
- Debe permitir realizar un diseño personalizado del instrumento, obteniendo así un instrumento resultante que adquirirá el dominio de instrumento virtual (Mánuel et al. 2002).

Según Mánuel, Biel, Olivé y Sánchez (2002, p. 227), la estructura típica de un programa de instrumentación está compuesta de los siguientes módulos:

Interfaz gráfica: Mediante el editor de la interfaz de usuario pueden construirse paneles interactivos para un programa, de forma que la interfaz gráfica sea capaz de satisfacer las necesidades del usuario (Mánuel et al. 2002).

Programa de control: Este programa permite el control de toda la aplicación: la adquisición y análisis de los datos, y la manipulación de la interfaz gráfica, para esto, existen los lenguajes de alto nivel, como los lenguajes gráficos o el lenguaje C (Mánuel et al. 2002).

Adquisición de datos: Se puede adquirir datos de un instrumento o de una tarjeta, para lo cual es necesario realizar un control del proceso de adquisición y de la tarea que ejecuta cada dispositivo en cada momento (Mánuel et al. 2002).

Análisis de datos: El software debe ser capaz de procesar la señal, realizar operaciones estadísticas sobre los datos, la adecuación de la señal a un tipo de formato capaz de ser escrito en un fichero, etc. (Mánuel et al. 2002).

Según Mánuel, Biel, Olivé y Sánchez (2002, p.230), las herramientas de programación más utilizadas para el desarrollo de un instrumento virtual son las siguientes:

2.6.2.1 MATLAB (Matrix Laboratory)

MATLAB viene siendo una plataforma de programación y cálculo numérico utilizada por el mundo científico y de la ingeniería, para realizar diversas actividades de programación como son el análisis de datos, el desarrollo de algoritmos y la creación de modelos. Además, combina los procesos de diseño con un lenguaje de programación que usa las matemáticas de matrices y arrays directamente (MathWorks 2016)

La interfaz gráfica, también conocida como GUI se muestra en la Ilustración 2-7 y se desarrolla mediante la distribución de los componentes visuales (mediante líneas de código o utilizando un toolbox) y la programación del comportamiento de los mismos desarrollada en un script de MATLAB (MathWorks 2016).

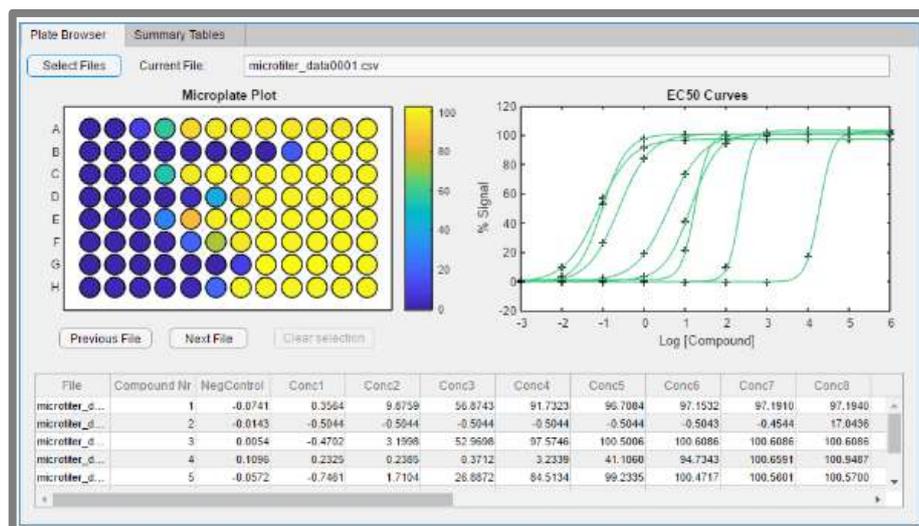


Ilustración 2-7: Instrumento virtual en MATLAB utilizando App Designer.

Fuente: (MathWorks 2016).

2.6.2.2 LabWindows/CVI (C for Virtual Instrument)

LabWindows/CVI es un conjunto de herramientas de programación de instrumentación virtual desarrollado por NI, que puede ser utilizado para desarrollar aplicaciones de pruebas automáticas, de medición y control, de adquisición de datos, de monitoreo de procesos, entre otras. Provee, además una ventana para el rastreo de variables, y es compatible con ejecuciones del tipo *single-step*, ejecuciones de interrupciones, seguimiento de procesos, verificación de parámetros, etc. (Wang y Tan 2006).

Un programa en este software consta de dos partes: una interfaz gráfica (denominada GUI, *Graphical User Interface*) capaz de programarse en lenguaje gráfico que permita visualizar y controlar operaciones y un programa escrito en C encargado de gestionar los mensajes de control. La interfaz o GUI consta principalmente de paneles, menús, controles e indicadores, tal y como se observa en la Ilustración 2-8 (Mánuel et al. 2002).

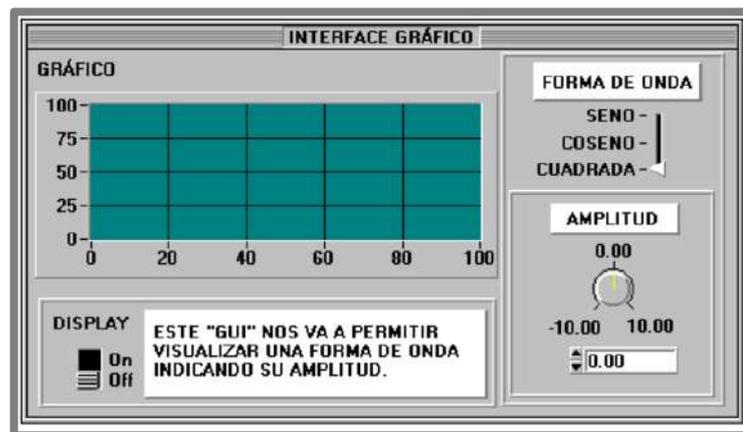


Ilustración 2-8: Ejemplo de interfaz gráfica con LabWindows/CVI.

Fuente: (Mánuel et al. 2002, p. 230).

2.6.2.3 LabVIEW (Laboratory Virtual Engineering Workbench)

LabVIEW es un entorno basado en programación gráfica, muy usado en el campo de la ingeniería para el desarrollo de sistemas y pruebas automatizadas de investigación, validación y producción (National Instruments 2020).

LabVIEW es un conjunto de entornos gráficos revolucionarios, diseñado especialmente para la adquisición de datos, control de dispositivos, análisis de datos y representación de los mismos. Es compatible con diversas plataformas como Windows, Mac OS, Sun, HP-UX, MAC, etc.,

utilizando diferentes buses de comunicación para la adquisición de datos como GPIB, VXI, PXI, dispositivos seriales, PLC, entre otras tarjetas de adquisición de datos (Wang y Tan 2006).

Los programas que se realizan en este software se denominan Instrumentos Virtuales (VI), puesto que imitan el aspecto y uso de instrumentos reales. El *Front Panel* (Ilustración 2-9) viene a ser la interfaz interactiva con el usuario de un VI, ya que simula ser un panel frontal de un instrumento físico, este contiene interruptores, botones, gráficos, controles e indicadores. Mientras que el panel denominado *Block Diagram* (Ilustración 2-10) es el encargado de enviar instrucciones al VI, donde dicho panel es construido en G (*graphical programming language*) (Mánuel et al. 2002).

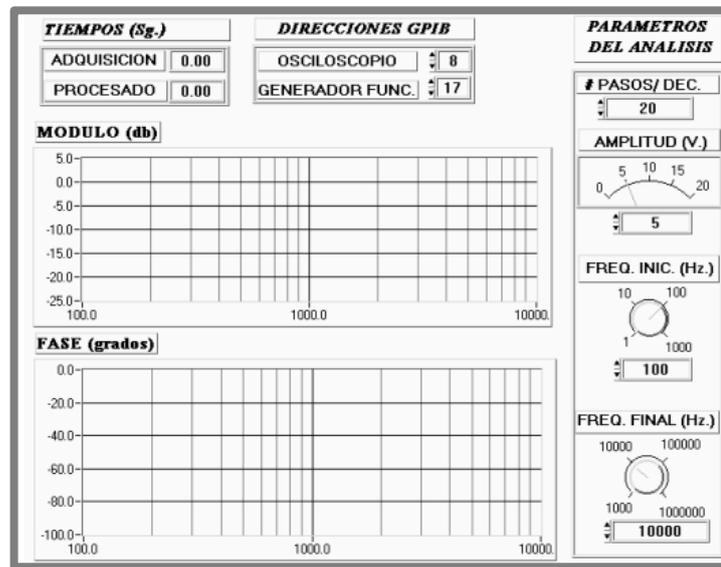


Ilustración 2-9: Ejemplo de un panel frontal de un VI.

Fuente: (Mánuel et al. 2002, p. 263).

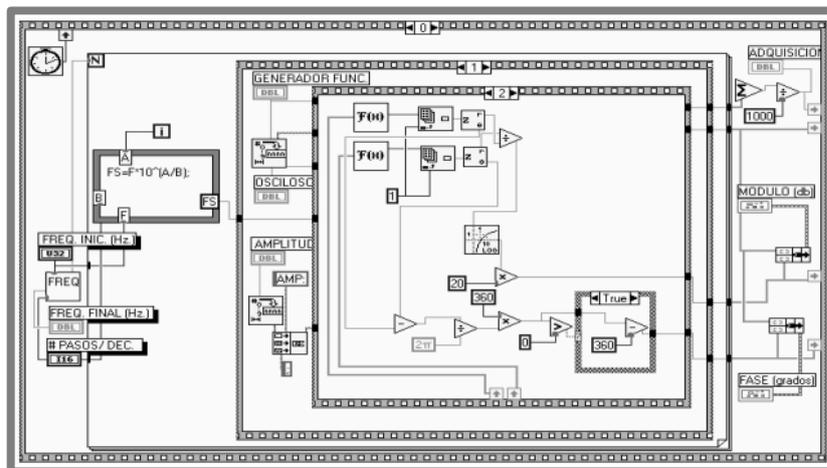


Ilustración 2-10: Ejemplo de un diagrama de bloques de un VI.

Fuente: (Mánuel et al. 2002, p. 263).

2.6.2.4 Diferencias de programación

A continuación, se muestran las principales diferencias entre los softwares mencionados anteriormente, haciendo énfasis en su entorno de programación con respecto a la utilidad que poseen al momento de generar un instrumento virtual.

Tabla 2-2: Tabla comparativa entre los distintos tipos de software de instrumentación.

Software	Labview	Matlab	LabWindows/CVI
Desarrollador	NI	Mathworks	NI
Programación	Gráfica	Informática	Comandos C
Compatibilidad	Windows, Linux, macOS y multiplataforma	Windows, Linux y macOS	Windows
Tipo de entorno	Plataforma de diseño	Informático/numérico	Gráfico/numérico

Fuente: (Mánuel et al. 2002).

Realizado por: Argos J; Granizo L, 2023.

Según la Tabla 2-2, el mejor software para el desarrollo de instrumentos virtuales es LabVIEW, no solo por el tipo de programación, sino también por la compatibilidad con los diversos sistemas operativos y por la plataforma de diseño que brinda al usuario (Mánuel et al. 2002).

2.6.3 Diseño de instrumentación virtual

Según Mendiburu (2006, p. 69), las etapas para el diseño de un instrumento virtual son las siguientes:

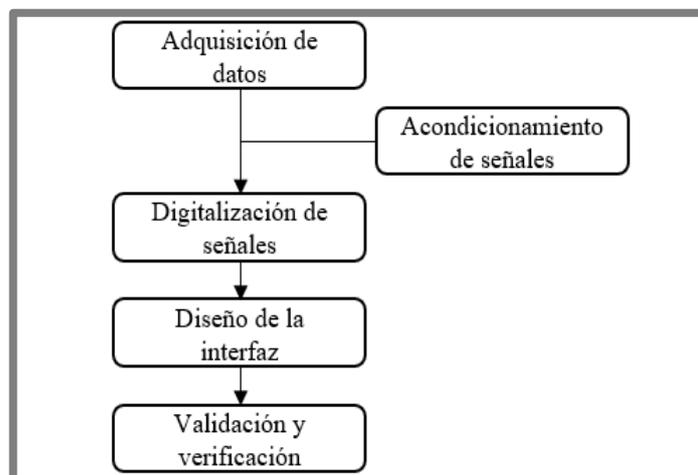


Ilustración 2-11: Etapas para el desarrollo de un instrumento virtual.

Fuente: (Mendiburu 2006, p. 69).

2.6.3.1 Adquisición de datos

La adquisición de datos consiste en captar una señal física y transportarla a una computadora, esto quiere decir que el conjunto de variables que se adquieren en forma física, debe ser convertido en un conjunto de voltajes eléctricos, para que puedan ser usadas o leídas desde la PC (Mendiburu 2006).

Las señales eléctricas se transforman en señales digitales dentro de la memoria de la PC, para que puedan ser procesadas mediante un programa adecuado al uso requerido por el usuario, así mismo, una señal digital o binaria puede convertirse en una eléctrica, para que la PC pueda enviar señales hacia los actuadores (Mendiburu 2006).

Para una correcta adquisición de datos, se debe realizar una selección adecuada de los dispositivos denominados transductores, encargados de convertir una magnitud física (presión, temperatura, luz, etc.) en señales eléctricas de voltaje o corriente (Mendiburu 2006).

2.6.3.2 Acondicionamiento de señales

Según Mendiburu (2006, p. 70) el acondicionamiento de señales es opcional, ya que depende mucho de cada señal y aplicación, donde se puede o no requerir amplificación, atenuación, filtraje, etc. Este proceso requiere de varias subetapas, que son:

- **Amplificación:** la señal es amplificada para obtener la mayor precisión posible, de modo que su nivel máximo coincida con el nivel máximo del convertidor o tarjeta (Mendiburu 2006).
- **Aislamiento:** se realiza un aislamiento eléctrico entre el transductor y el ordenador, para proteger ambos dispositivos de señales transitorias de alta tensión que puedan dañar el proceso (Mendiburu 2006).
- **Multiplexado:** Es la conmutación de las entradas del convertidor, permitiendo la medición de los datos de diferentes canales de entrada desde un solo convertidor (Mendiburu 2006).
- **Filtrado:** Consiste en eliminar las señales no deseadas como el ruido. Para señales alternas como la vibración, se usa un filtro *antialiasing*, que es un filtro pasa bajo analógico que elimina totalmente las señales de mayor frecuencia (Mendiburu 2006).

- **Linealización:** La linealización se la realiza en diversos transductores que presentan una respuesta no lineal. Lo recomendable es hacer la linealización mediante circuitos externos, aunque también puede realizarse con métodos numéricos en el sistema de adquisición de datos (Mendiburu 2006).

2.6.3.3 Digitalización de señales

Las señales son ondas que comunican un punto de referencia con un punto receptor. Las señales eléctricas pueden ser continuas (analógicas) o discretas (digitales), donde generalmente las señales analógicas permiten la comunicación de los instrumentos de campo, mientras que el ordenador y demás dispositivos electrónicos se conectan por medio de señales digitales. Por lo que se requiere el paso de un tipo de señal hacia otro (Mendiburu 2006).

Según Mendiburu (2006, p. 72), un convertidor analógico-digital es un dispositivo que convierte una señal analógica (tensión o corriente) de entrada en una señal digital (binaria) de salida. Realiza 3 funciones en específico, empezando por las etapas de muestreo y cuantificación, y terminando con la codificación de las señales. Tal y como se observa en la Ilustración 2-12.

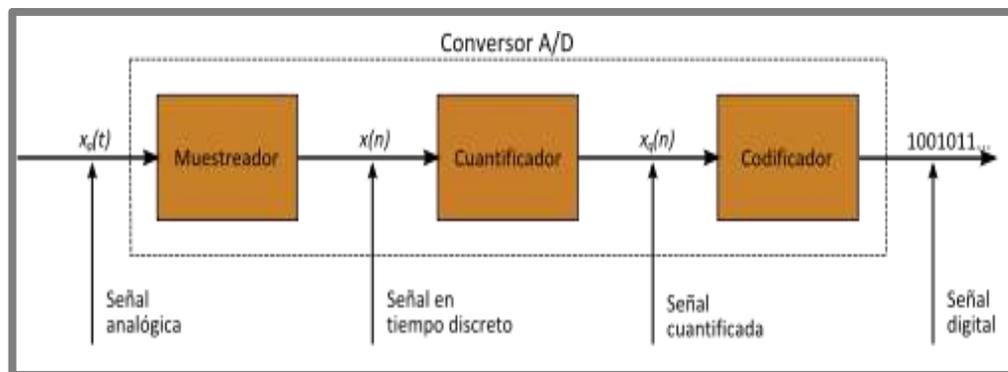


Ilustración 2-12: Proceso de un convertidor AD.

Fuente: (Mendiburu 2006, p. 72).

El convertidor DA es un dispositivo que realiza el proceso inverso al convertidor AD, emitiendo una señal analógica a partir de una señal de entrada digital (Mendiburu 2006).

En la conversión de señales, pueden ocurrir diversos errores, entre los cuales se encuentran el error de ganancia, error diferencial no lineal, error integral no lineal, error de apertura, error de *Offset* y el error de efecto *Aliasing* (Mendiburu 2006).

2.6.3.4 *Diseño de la interfaz de usuario y adquisición de señales*

Una vez adquiridas las señales desde los sensores o cualquier tipo de dispositivo de adquisición de datos, deben ser procesadas utilizando las herramientas de programación tratadas en el apartado anterior, cuya aplicación es la de adquirir, procesar, analizar, almacenar y visualizar datos provenientes de campo tomados por hardware de instrumentación virtual (Mendiburu 2006).

La interfaz gráfica se diseña mediante el software con el editor de la interfaz de usuario, donde existirá la presencia de paneles interactivos para el programa, adecuando la interfaz a las necesidades del usuario. En dicha interfaz, se visualiza cualquier tipo de señales y datos que se necesiten para la interpretación de la aplicación en curso. Es por eso que, dentro de la interfaz debe existir una serie explícita de controles, cajas de diálogos y menús (Mánuel et al. 2002).

2.6.3.5 *Validación y verificación de la instrumentación virtual*

La verificación y la validación son un conjunto de procesos de análisis y comprobación que aseguran que el software desarrollado cumpla con su especificación y satisfaga las necesidades de los usuarios. El proceso de verificación comprueba que el software cumpla con los requisitos funcionales y no funcionales de su especificación, mientras que la validación asegura que el software cumpla con las expectativas esperadas por el usuario (Drake y López 2009).

Las técnicas de verificación y validación son las siguientes:

- **Inspecciones del software:** Donde se analizan las diferentes representaciones del sistema (diagramas y código) en busca de defectos o errores, y el código no necesita ser ejecutado ya que son pruebas de validación estáticas, y estas deben realizarse durante todo el ciclo de desarrollo (Drake y López 2009).
- **Pruebas del software:** Las respuestas de una implementación del software son sometidas a diversas series de datos de prueba, para examinar dichas respuestas y su comportamiento operacional, comprobando el desempeño del software conforme a lo requerido. Ejecutar estas pruebas requiere una técnica dinámica de la verificación y la validación ya que es necesario disponer de un prototipo ejecutable del sistema (Drake y López 2009).

En resumen, para construir un instrumento virtual se requiere de una PC, una tarjeta de adquisición de datos con acondicionamiento de señales (PCMCIA, ISA, XT, PCI) y el software apropiado para llevar a cabo la programación, estos tres elementos son clave para la creación de cualquier instrumento virtual, teniendo al acondicionamiento de señales como elemento opcional (Mendiburu 2006).

2.7 Procesamiento digital de imágenes

Matemáticamente, una imagen se define como una función bidimensional $f(x,y)$, siendo x y y coordenadas espaciales en un plano, y f es la intensidad o nivel de gris en cualquier par de coordenadas (Gonzalez y Woods 2008).

Una imagen digital es una imagen en la que los valores x , y y f son todos finitos (cantidades discretas). Esta se compone de un número finito de elementos, que tienen un lugar y valor específicos, conocidos como *pels* o *pixels* (Gonzalez y Woods 2008).

Entonces, el procesamiento digital de imágenes es el conjunto de técnicas que se aplican a imágenes digitales con el fin de mejorar la calidad o facilitar la búsqueda de información. Un paradigma útil es considerar tres tipos de procesos computarizados que según Gonzales y Woods (2008), son:

- **Procesos de bajo nivel:** Utilizan el preprocesamiento de imagen, mejora y filtros de enfoque. Sus entradas son imágenes y sus salidas también (Gonzalez y Woods 2008).
- **Procesos de nivel medio:** Utilizan la segmentación y clasificación de objetos individuales. Sus entradas son imágenes, mientras que sus salidas son atributos de las imágenes (contornos, bordes, etc.) (Gonzalez y Woods 2008).
- **Procesos de alto nivel:** Operan con el análisis de imágenes y realizan funciones cognitivas asociadas con la vista (Gonzalez y Woods 2008).

Ejemplos de campos que usan el procesamiento digital de imágenes son el reconocimiento militar, el procesamiento automático de huellas dactilares, la detección de rayos X y muestras de sangre (Gonzalez y Woods 2008).

La Ilustración 2-13 muestra las etapas que componen el procesamiento digital de imágenes, que según Gonzales y Woods (2008, p.26) son:

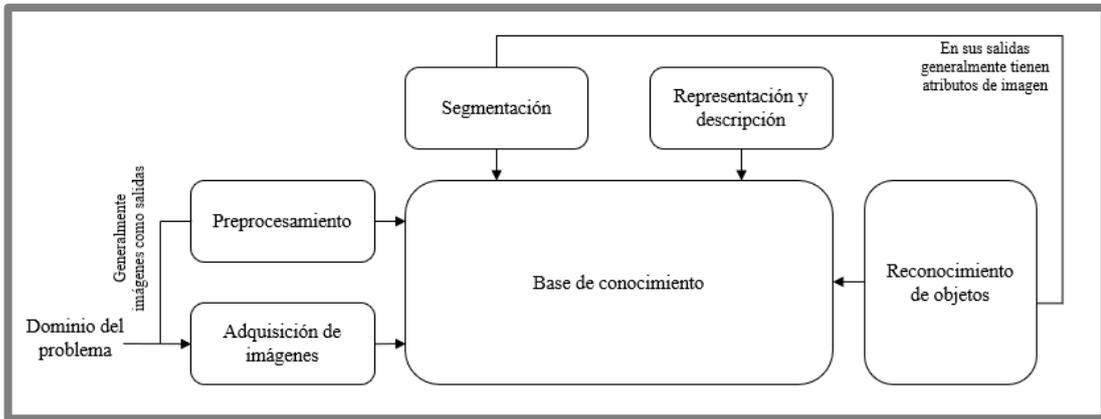


Ilustración 2-13: Etapas del procesamiento digital de imágenes.

Fuente: (Gonzalez y Woods 2008, p. 26).

2.7.1 Adquisición de imágenes

Para la adquisición se necesitan dos elementos: un dispositivo físico sensible a una banda del espectro electromagnético (rayos X, ultravioleta, infrarrojo o visible) que sea capaz de producir una señal eléctrica de salida proporcional al nivel de energía requerido y un digitalizador, cuya finalidad es convertir la señal de salida del sistema a su forma digital (Gonzalez y Woods 2008).

La Ilustración 2-14 muestra el proceso de adquisición de imágenes, donde la energía de iluminación se refleja en un elemento de escena, para que el sistema de imágenes recolecte la energía de entrada y la enfoque en un plano de imagen. La imagen de salida es proporcional a la integral de la luz recibida en cada sensor y debe ser digitalizada (Gonzalez y Woods 2008).

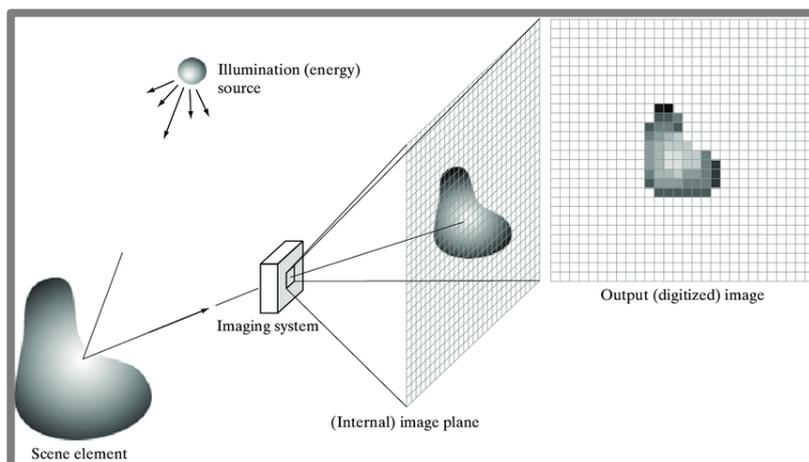


Ilustración 2-14: Proceso de adquisición de imágenes digitales.

Fuente: (Gonzalez y Woods 2008, p. 51).

2.7.1.1 *Detección y adquisición de imágenes*

Existen 3 disposiciones principales de sensores usadas para transformar energía de iluminación en imágenes digitales, el proceso que se lleva a cabo es el de transformar la energía entrante en voltaje mediante la combinación de energía eléctrica de entrada y un sensor capaz de responder al tipo de energía que se detecta (Gonzalez y Woods 2008).

Adquisición con un solo sensor: El sensor más conocido es el fotodiodo de silicio, cuyo voltaje de salida es proporcional a la luz, aunque a veces se usa un filtro adelante del sensor para mejorar la selección de varias longitudes de onda (Gonzalez y Woods 2008).

Adquisición con bandas o líneas de sensores: Se forma una banda mediante un arreglo de sensores en línea, cuya función es la de proveer elementos de la imagen en una dirección perpendicular. Otra configuración es la banda circular o anillo, donde la fuente rota mientras los sensores en el lado opuesto recolectan energía que pasa a través del objeto (Gonzalez y Woods 2008).

Adquisición por arreglos de sensores: Numerosos sensores electromagnéticos y ultrasónicos se ordenan en forma de arreglos, uno de los más típicos es el dispositivo de carga acoplada (CCD) utilizado para cámaras. La mayor ventaja de los arreglos es que con sólo enfocar el patrón de energía en la superficie del arreglo se puede obtener una imagen completa (Gonzalez y Woods 2008).

2.7.1.2 *Muestreo y cuantificación de las imágenes*

La Ilustración 2-15 muestra la idea básica de muestreo y cuantificación, donde una imagen continua f se convierte en digital. Este proceso de digitalización se lleva a cabo muestreando los valores de la función en ambas coordenadas de la función (muestreo) y en su amplitud (cuantificación). La representación digital de una imagen se identifica como un plano que contiene los píxeles de la misma (Gonzalez y Woods 2008).

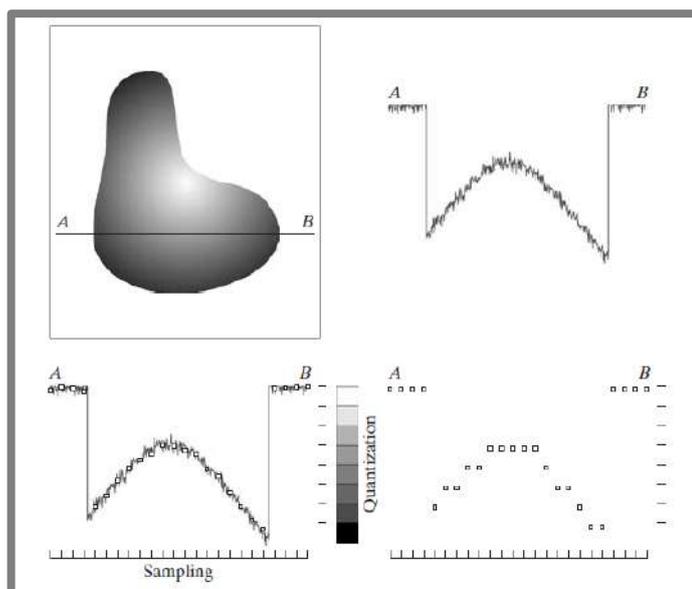


Ilustración 2-15: Generación de una imagen digital.

Fuente: (Gonzalez y Woods 2008, p. 53).

2.7.1.3 Resolución espacial y de intensidad

La resolución espacial se refiere a la medida del más mínimo detalle perceptible en una imagen, la cual puede establecerse como pares de líneas por unidad de distancia y puntos (píxeles) por unidad de distancia, entre las medidas más comunes. La resolución de intensidad es el cambio perceptible más pequeño en el nivel de intensidad, cuyo número se presenta como una potencia entera, siendo 8 bits el más común mientras que 16 y 32 bits se utilizan para aplicaciones específicas (Gonzalez y Woods 2008).

2.7.1.4 Interpolación de imágenes

Se refiere al proceso de utilizar datos conocidos para estimar valores en lugares desconocidos. Esta herramienta es ampliamente utilizada en tareas como aumentar (zoom) y reducir el tamaño de una imagen, realizar correcciones rotatorias y geométricas (Gonzalez y Woods 2008).

2.7.2 Filtrado y mejora de las imágenes

Una imagen filtrada o procesada se genera a medida que el centro del filtro se sobrepone en cada pixel de la imagen de entrada. Un filtro puede ser lineal o no lineal y puede trabajar tanto en el dominio espacial como en el dominio de la frecuencia (Gonzalez y Woods 2008).

2.7.2.1 *Transformaciones de intensidad y filtrado espacial*

El dominio espacial es la sección del plano real atravesada por las coordenadas de una imagen, siendo x e y variables o coordenadas espaciales. Al hablar de dominio espacial, entra en juego el propio plano de la imagen donde se realiza el filtrado y procesamiento de la misma. Dos categorías principales de procesamiento espacial son las transformaciones de intensidad y el filtrado espacial (Gonzalez y Woods 2008).

Las transformaciones de intensidad (llamadas también niveles de gris o mapeo) operan en píxeles individuales de una imagen, con el fin de manipular el contraste, umbralización y binarización de la misma. El filtrado espacial (máscara espacial, kernel, plantilla o ventana) realiza operaciones en la nitidez de la imagen, trabajando en un vecindario que rodea a cada píxel. Generalmente, las técnicas de dominio espacial son más eficientes a nivel computacional y requieren menos procesamiento y recursos para su implementación (Gonzalez y Woods 2008).

2.7.2.2 *Correlación espacial y convolución*

La correlación es el movimiento de una máscara de filtro (filtro kernel) sobre una imagen para hallar la suma de los productos de cada píxel en cada ubicación. La convolución realiza el mismo proceso que la correlación, con la excepción de que el filtro debe rotarse 180° antes de hacer la operación (Gonzalez y Woods 2008).

La Ilustración 2-16 muestra de mejor manera ambos procesos, donde se aplica un filtro de tamaño $m \times n$ (a) y se rellena una función, en este caso un impulso unitario discreto, con $m-1$ filas de ceros en la parte superior e inferior y $n-1$ columnas en la parte izquierda y derecha (b). El proceso de correlación se muestra en (c) y (d), mientras que (e) presenta el resultado recortado. Para la convolución se rota la máscara de filtro 180° y se realiza la suma de productos, cuyo resultado se muestra desde (f) hasta (h). Si la máscara de filtro es simétrica, la correlación y la convolución producen el mismo resultado (Gonzalez y Woods 2008).

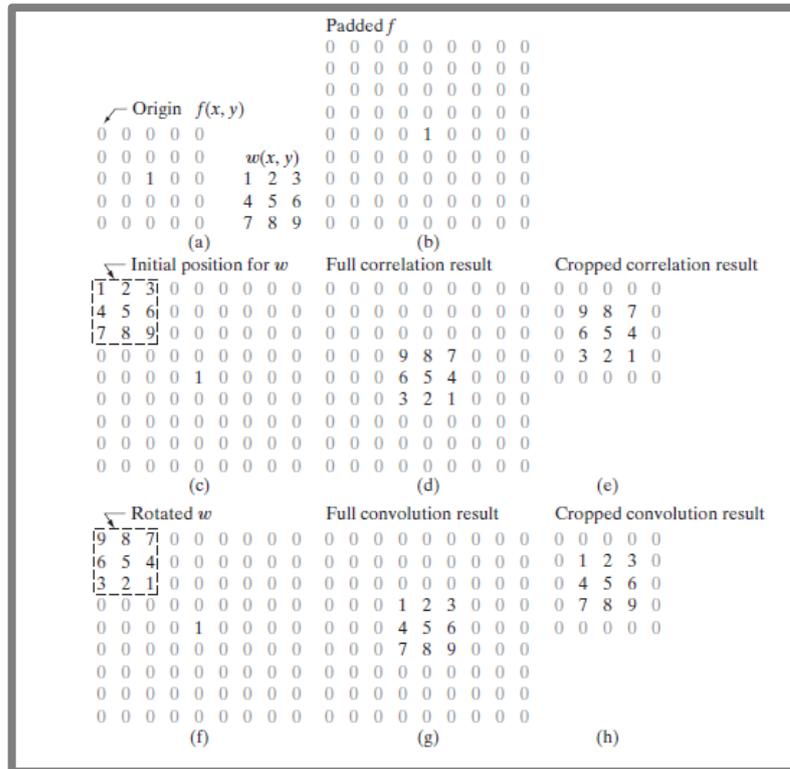


Ilustración 2-16: Correlación y convolución de un filtro con un impulso unitario discreto.

Fuente: (Gonzalez y Woods 2008, p. 149).

2.7.2.3 Filtros espaciales de suavizado

Los filtros de suavizado sirven para difuminar y reducir el ruido de una imagen. En tareas de preprocesamiento se utiliza el difuminado para la eliminación de detalles pequeños de una imagen antes de la extracción de objetos grandes y para la conexión de pequeñas brechas en curvas o líneas. La reducción de ruido se obtiene utilizando el difuminado con un filtro lineal o no lineal (Gonzalez y Woods 2008).

Filtro de suavizado lineal: Llamados también filtros de promedio o filtros pasa bajo, la salida o respuesta que obtienen es el promedio de los píxeles contenidos en el vecindario de la máscara de filtro. Al reemplazar el valor de cada píxel de una imagen por el promedio de los niveles de intensidad en un vecindario definido por la máscara de filtro, se genera una imagen con transiciones de intensidad reducidas. Este tipo de filtros son usados para la reducción de ruido, el suavizado de contornos falsos y la reducción de detalles irrelevantes en una imagen (Gonzalez y Woods 2008).

Filtros estadísticos de orden (No lineales): Su respuesta se basa en clasificar y ordenar los píxeles contenidos en el área contenida por el filtro, para luego reemplazar el valor del píxel central por el valor resultante de la clasificación. El más usado es el filtro de la mediana, que reemplaza el valor de un píxel por la mediana de los valores de intensidad del vecindario en el que se encuentre, incluyendo el valor original del píxel. Son eficientes en presencia de ruido impulsivo, llamado también ruido sal y pimienta por su la presencia de puntos blancos y negros superpuestos en la imagen (Gonzalez y Woods 2008).

2.7.2.4 *Filtros espaciales de realce o enfoque*

Tienen como objetivo principal resaltar las transiciones en la intensidad utilizando diferenciación espacial, es decir, la fuerza de la respuesta de un operador derivativo es proporcional al grado de discontinuidad de intensidad de la imagen en el punto donde se aplica el operador (Gonzalez y Woods 2008).

Dentro de este tipo de filtros se encuentra el Laplaciano, que usa la segunda derivada para el realce de la imagen. También se encuentra el Gradiente, cuyo proceso para llevar a cabo el realce de la imagen (no lineal) incluye el uso derivadas de primer orden (Gonzalez y Woods 2008).

2.7.3 *Compresión de imágenes*

La compresión de datos es el proceso mediante el cual se reduce la cantidad de datos necesarios para representar cierta cantidad de información, donde los datos son el medio para transmitir dicha información. En este caso, la compresión de imágenes es una de las tecnologías más utilizadas para representar una imagen en una cantidad reducida de datos (Gonzalez y Woods 2008).

Para llevar a cabo la compresión de imágenes se necesitan de dos únicos procesos: la codificación y la decodificación. El codificador se encarga de realizar la compresión, eliminando la redundancia espacial y temporal mediante un mapeo de la imagen, cuantificándola para eliminar información irrelevante y finalmente, mediante un codificador de símbolos reducir la redundancia de codificación. La decodificación o descompresión de una imagen se realiza mediante un decodificador de símbolos y un mapeo inverso (Gonzalez y Woods 2008).

2.7.3.1 *Tipos de compresión*

Compresión con pérdidas: Los datos se reducen eliminando cierta información menos relevante de los archivos, este tipo de compresión suele aplicarse a formatos de imagen y video, donde pese

a la reducción significativa del tamaño del archivo, la cantidad de datos que se pierde es mínima (Gonzalez y Woods 2008).

Compresión sin pérdidas: Tiene como objetivo reducir el tamaño del archivo sin perder información, esto se logra mediante algoritmos que reorganizan los datos para eliminar redundancias y patrones innecesarios, donde la información original se recupera completamente al descomprimir el archivo. Por lo general se usa en archivos de texto y de audio (Gonzalez y Woods 2008).

2.7.3.2 Formatos de imagen

Un formato de imagen es una manera estándar de organizar y almacenar datos de imágenes, definiendo la organización de dichos datos y el tipo de compresión que se utiliza (Gonzalez y Woods 2008).

La Tabla 2-3 muestra los formatos más utilizados según los estándares internacionales establecidos por diversas organizaciones (Gonzalez y Woods 2008).

Tabla 2-3: Formatos de imagen más utilizados.

Formato de imagen	Método de compresión	Organización	Característica principal
PNG	-Codificación de Huffman. -Codificación Run-Length	World Wide Web Consortium	Comprime sin pérdida imágenes a color completo (hasta 48 bits/píxel).
JPEG	-Codificación Run-Length. -Codificación predictiva	ISO/IEC/ITU-T	Su sistema de codificación básica con pérdidas utiliza transformadas de coseno discreto cuantizadas (DCT) en bloques de imágenes de 8x8.
GIF	Codificación LZW	CompuServe	Utiliza codificación sin pérdida para imágenes de 1 a 8 bits, siendo su mayor uso el de crear pequeñas animaciones.

Formato de imagen	Método de compresión	Organización	Característica principal
TIFF	Codificación LZW	Aldus	Es un formato de archivo de imagen etiquetada que admite una variedad de estándares de compresión de imágenes.
DV	Codificación de transformación	Microsoft	Es un formato de video digital donde los fotogramas se comprimen independientemente para facilitar la edición.
BMP	Codificación Run-Length	Microsoft	Se utiliza principalmente para imágenes simples sin compresión.
VC-1	Codificación de transformación de bloque	SMPTE	Es el formato de video más usado en Internet.
AVS	Codificación Golomb	MII	Es un estándar de audio y video desarrollado en China que utiliza la codificación Golomb exponencial.
HDV	Codificación de transformación de bloque	IEC	Sirve para videos de alta definición, donde se incluye la eliminación de redundancia temporal mediante diferenciación de predicción.

Fuente: (Gonzalez y Woods 2008).

Realizado por: Argos J; Granizo L, 2023.

En la actualidad, el formato más utilizado es el JPEG (Joint Photographic Experts Group) debido a su capacidad para comprimir imágenes con pérdida de manera eficiente, lo que reduce el tamaño del archivo y facilita su compartición en línea (Gonzalez y Woods 2008).

2.7.4 Segmentación de imágenes

La segmentación divide una imagen en regiones u objetos constituyentes que sean de interés y el nivel de detalle de cada subdivisión depende del problema o aplicación. Los algoritmos de segmentación se basan en dos propiedades básicas de los valores de intensidad: discontinuidad y similitud. En la discontinuidad, la imagen se divide en función de cambios abruptos en la

intensidad, como los bordes. La similitud divide una imagen en regiones similares según varios criterios predefinidos (Gonzalez y Woods 2008).

2.7.4.1 Segmentación basada en bordes

Este tipo de segmentación se basa en la detección de cambios locales y abruptos en la intensidad, donde los píxeles de borde son aquellos en los que la intensidad de una función de imagen cambia bruscamente, y su agrupación se denomina borde o segmentos de borde (Gonzalez y Woods 2008).

Detección de puntos aislados: Se basa en identificar píxeles individuales con características distintivas en comparación a su entorno. Dichos puntos pueden ser regiones que sobresalen debido a un cambio abrupto o inusual en la intensidad en comparación con sus alrededores (Gonzalez y Woods 2008).

Detección de líneas: Una línea es un segmento de borde en el que la intensidad de sus alrededores es mucho más alta o mucho más baja que la intensidad de los píxeles de la línea. Para la detección de líneas se usa el operador convolucional conocido como máscara Laplaciana, que consiste en un filtro para resaltar regiones donde ocurren cambios abruptos de la intensidad, tal y como se muestra en la Ilustración 2-17 (Gonzalez y Woods 2008).

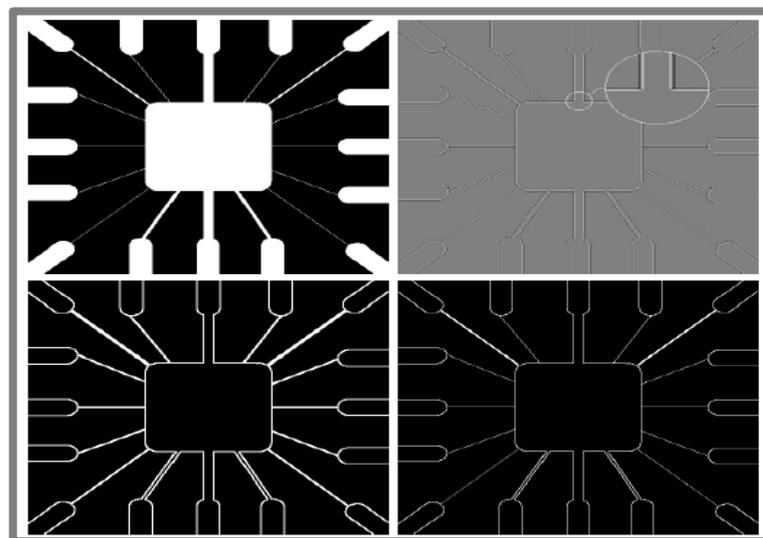


Ilustración 2-17: Detección de líneas en un circuito electrónico.

Arriba: Imagen original e imagen Laplaciana.

Abajo: Valor absoluto y valores positivos del Laplaciano.

Fuente: (Gonzalez y Woods 2008, p. 698).

Detección de bordes: Para identificar cambios bruscos en la intensidad en diferentes áreas de una imagen se pueden utilizar métodos como el gradiente de intensidad, operadores de derivadas, entre otros; en este caso, el más utilizado es la máscara Laplaciana, cuya ejemplificación se muestra en la Ilustración 2-18 (Gonzalez y Woods 2008).



Ilustración 2-18: Detección de bordes utilizando la gradiente de intensidad.

Fuente: (Gonzalez y Woods 2008, p. 711).

2.7.4.2 Segmentación basada en umbralización (*Thresholding*)

La umbralización también conocida como binarización de imágenes es un método de segmentación en el que se separan objetos de interés del fondo en una imagen. En otras palabras, se asigna un valor de umbral y se clasifican los píxeles de la imagen en dos categorías: objetos (valores de intensidad superiores al umbral) y fondo (valores de intensidad iguales o inferiores al umbral) (Gonzalez y Woods 2008).

2.7.5 Representación y descripción

2.7.5.1 Representación

Representar una región que sea útil para una computadora implica dos opciones: 1) representar la región en términos de sus características externas (contorno), o 2) representarla en términos de sus características internas (píxeles de la región). La representación externa se da cuando se trabaja con las características de forma, y la representación interna se enfoca en las propiedades regionales, como el color y la textura (Gonzalez y Woods 2008).

Existen varios enfoques de representación, entre los más utilizados se encuentran la representación mediante seguimiento de límites (bordes), códigos de cadena, aproximaciones poligonales utilizando polígonos de mínimo perímetro y segmentos de contorno (Gonzalez y Woods 2008).

La Ilustración 2-19 muestra un ejemplo de representación utilizando códigos de cadena, donde se representa el contorno de la imagen mediante una secuencia conectada de segmentos de línea recta con dirección y longitud especificadas. La dirección de cada segmento se codifica mediante un esquema de numeración para formar un contorno representado por una secuencia de números direccionales. Dicho esquema recibe el nombre de código de cadena Freeman (Gonzalez y Woods 2008).

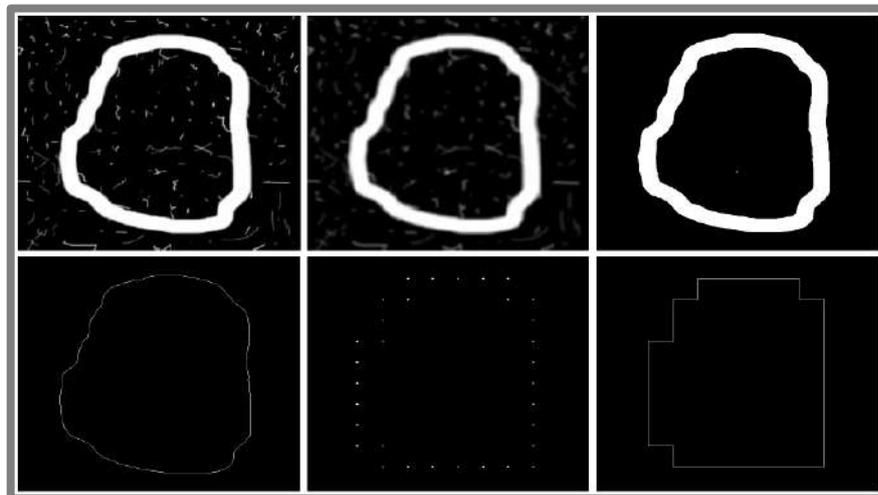


Ilustración 2-19: Código de cadena de Freeman.

Fuente: (Gonzalez y Woods 2008, p. 800).

2.7.5.2 Descriptores de contorno

Los descriptores de contorno son características numéricas o medidas que se utilizan para describir propiedades topológicas y geométricas del contorno de una imagen digital. Entre los descriptores más comunes se encuentran la longitud, el diámetro, el eje mayor y menor y la curvatura del contorno (Gonzalez y Woods 2008).

Otros descriptores más complejos se basan en los números de forma, así como también en las series de Fourier y momentos estadísticos. En el caso de usar momentos estadísticos, entra en juego medidas como la media, la varianza y los momentos de orden superior (Gonzalez y Woods 2008).

2.7.5.3 Descriptores regionales

Se utilizan para describir las propiedades de una región en una imagen digital. Los descriptores de región más simples son el área, el perímetro y la relación de circularidad, todos representados por píxeles (Gonzalez y Woods 2008).

Otros descriptores regionales con mayor complejidad son:

Descriptores topológicos: La topología estudia las propiedades de una figura que no se ve afectada por ninguna deformación, siempre y cuando no exista algún desgarro ni unión. La Ilustración 2-20 muestra una región con dos agujeros, donde un descriptor topológico puede definirse en función del número de agujeros que existan, en este caso las propiedades topológicas no dependen de la noción de la distancia (Gonzalez y Woods 2008).

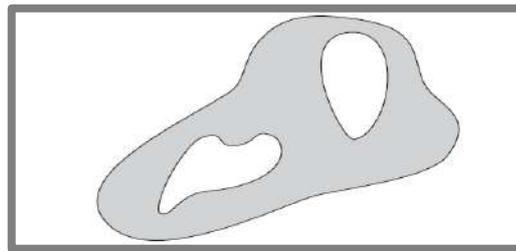


Ilustración 2-20: Región con dos agujeros.

Fuente: (Gonzalez y Woods 2008, p. 825).

Textura: Este descriptor proporciona medidas de propiedades como grosor, regularidad y suavidad (algunos ejemplos se muestran en la ilustración 2-21). Para describir la textura de una región se utilizan tres enfoques principales que son estadísticos (texturas suaves, gruesas, granulosas, etc.), espectrales (periodicidad global) y estructurales (textura basada en líneas paralelas) (Gonzalez y Woods 2008).

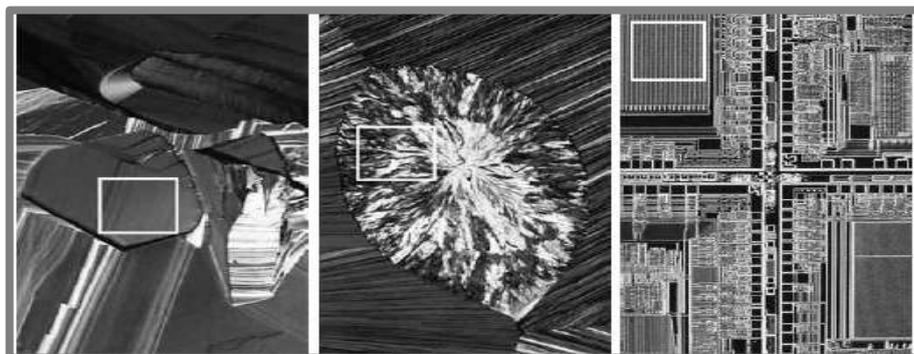


Ilustración 2-21: Ejemplos de textura: Suavidad, grosor y regularidad.

Fuente: (Gonzalez y Woods 2008, p. 828).

Momentos invariantes de Zernike: Reciben este nombre debido a que son momentos invariantes ante transformaciones rígidas y proporcionan información sobre la forma y el contenido de la región. Utilizan funciones matemáticas para describir características geométricas y texturales de objetos en una imagen. Se utilizan en diversas aplicaciones como el análisis de estructuras, el reconocimiento de patrones, la detección de objetos, entre otras (Gonzalez y Woods 2008).

La Ilustración 2-22 muestra un ejemplo de los momentos invariantes, donde la imagen pasa por diversos procesos de traslación, escalado, reflejado y giro de 45° y 90° respectivamente. La Tabla 2-4 muestra los valores que adquieren las imágenes para siete momentos invariantes de Zernike, donde dichos valores son muy similares con la excepción del séptimo polinomio para la imagen reflejada, que adquiere un valor negativo (Gonzalez y Woods 2008).

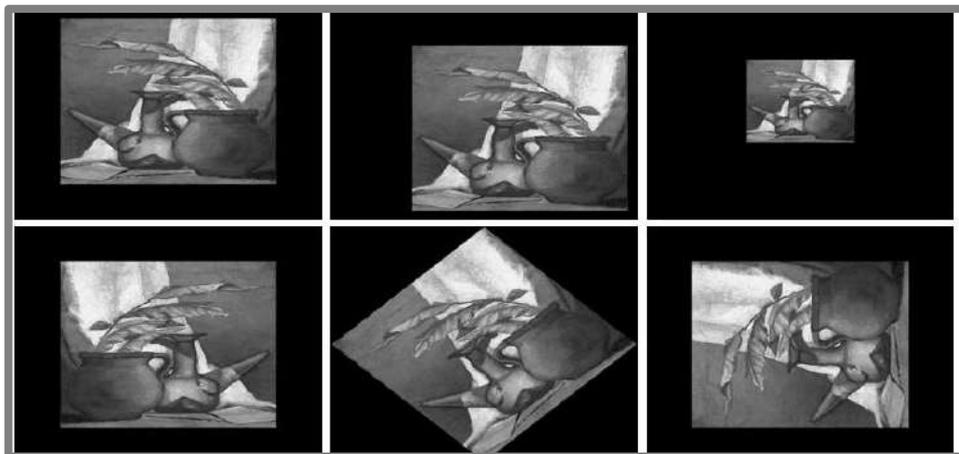


Ilustración 2-22: Imágenes utilizadas para determinar los momentos invariantes.

Fuente: (Gonzalez y Woods 2008, p. 842).

Tabla 2-4: Momentos invariantes para las imágenes anteriores.

Momentos invariantes	Imagen original	Imagen trasladada	Imagen escalada	Imagen reflejada	Imagen rotada 45°	Imagen rotada 90°
\emptyset_1	2.8662	2.8662	2.8664	2.8662	2.8661	2.8662
\emptyset_2	7.1265	7.1265	7.1257	7.1265	7.1266	7.1265
\emptyset_3	10.4109	10.4109	10.4047	10.4109	10.4115	10.4109
\emptyset_4	10.3742	10.3742	10.3719	10.3742	10.3742	10.3742
\emptyset_5	21.3674	21.3674	21.3924	21.3674	21.3363	21.3674
\emptyset_6	13.9417	13.9417	13.9383	13.9417	13.9417	13.9417
\emptyset_7	-20.7809	-20.7809	-20.7724	20.7809	-20.7813	-20.7809

Fuente: (Gonzalez y Woods 2008).

Realizado por: Argos J; Granizo L, 2023.

2.7.6 Reconocimiento de objetos usando redes neuronales artificiales

El reconocimiento de objetos tiene como objetivo identificar y clasificar objetos o patrones dentro de una imagen. Los enfoques del reconocimiento se dividen en dos áreas: decisional (mediante descriptores cuantitativos como longitud, área y textura) y estructural (mediante descriptores cualitativos) (Gonzalez y Woods 2008).

Las redes neuronales son modelos de aprendizaje automático que imitan la forma en que las neuronas en el cerebro humano procesan la información. Cada neurona realiza una transformación no lineal de sus entradas ponderadas y genera una salida. Las neuronas se organizan en capas, que incluyen la capa de entrada, las capas ocultas y la capa de salida. La unidad básica de las redes neuronales es el perceptrón, cuya función es aprender a separar patrones en distintas clases mediante ajustes iterativos de sus pesos y umbrales, aunque tiene limitaciones en problemas no lineales y clases no separables (Gonzalez y Woods 2008).

2.7.6.1 Patrones y clases de patrones

Un patrón no es más que un arreglo de descriptores, mientras que una clase es una familia de patrones que comparten propiedades en común. Los arreglos más comunes de patrones utilizados en la práctica son los vectores (para descripciones cuantitativas), cadenas y árboles (para descripciones estructurales) (Gonzalez y Woods 2008).

Vectores de patrones: Se representan como columnas (matrices) y la naturaleza de sus componentes depende del enfoque utilizado para describir el patrón físico en sí (Gonzalez y Woods 2008).

Cadenas de patrones: Generan patrones de objetos con una estructura basada en una conectividad simple de cadenas primitivas, asociada a la forma del límite. El uso de descripciones en forma de árbol es un enfoque más práctico ya que la mayoría de esquemas jerárquicos de ordenación conducen a dichas estructuras (Gonzalez y Woods 2008).

2.7.6.2 Tipo de Datasets

Un conjunto o base de datos es una colección organizada de imágenes que se utiliza para entrenar o validar modelos y técnicas en tareas de reconocimiento de objetos (Hastie, Tibshirani y Friedman 2008).

Continuos: involucra datos numéricos que representan valores asociados a los píxeles de una imagen. Estos valores pueden representar intensidades de color, niveles de gris o cualquier característica cuantitativa que describa la información visual de la imagen (Hastie, Tibshirani y Friedman 2008).

Categoricos: los datos pertenecen a un subconjunto de valores prescritos, pueden ser representados como etiquetas que clasifican objetos dentro de una imagen (Hastie, Tibshirani y Friedman 2008).

2.7.6.3 *Deep Learning*

El aprendizaje profundo (*deep learning*), es una rama del aprendizaje automático (*machine learning*) y se centra en el uso de redes neuronales artificiales profundas para modelar y resolver problemas complejos. Se puede entender como un conjunto de algoritmos desarrollados para entrenar redes neuronales artificiales con muchas capas de manera más eficiente, lo que permite a la red aprender representaciones jerárquicas de datos (Raschka 2015).

Según Raschka (2015, p. 381), las tres arquitecturas más importantes de redes neuronales profundas, son las siguientes:

Redes neuronales de perceptrones multicapa (MLP): Es una arquitectura básica y fundamental con capas de nodos organizados en al menos tres capas: capa de entrada, una o varias capas ocultas y capa de salida, la información se propaga hacia adelante (*feedforward*) a través de la red, desde la capa de entrada hasta la capa de salida, sin ciclos ni retroalimentación. Se utilizan en una variedad de aplicaciones, desde clasificación y regresión hasta procesamiento de lenguaje natural (Raschka 2015).

El entrenamiento de este tipo de redes se realiza mediante aprendizaje supervisado, utilizando un conjunto de datos etiquetados, donde las entradas están asociadas a salidas conocidas. Para calcular y ajustar los gradientes en relación a la función de pérdida, se utiliza el algoritmo de retropropagación, esto permite actualizar los pesos de la red de manera eficiente (Raschka 2015).

Redes neuronales convolucionales (CNN): Son diseñadas para procesar datos estructurados en cuadrículas (imágenes y datos de video) y tienen como objetivo capturar patrones y características visuales independientemente de la posición que tengan en la imagen. Consisten en capas convolucionales que aplican filtros en paralelo a diferentes regiones de la imagen para la detección automática de características visuales en diferentes escalas y orientaciones. Son

cruciales en tareas de visión por computadora, como clasificación de imágenes, detección de objetos y segmentación semántica (Gonzalez y Woods 2008).

En la Ilustración 2-23, una capa convolucional conecta los píxeles de entrada a cada unidad en el mapa de características, donde cada unidad representa la activación de un filtro específico en respuesta a patrones detectados en la entrada, para identificar características relevantes en las imágenes. La capa de agrupación se encarga de extraer características tomando el valor promedio o máximo de un conjunto de características vecinas para enviarlas a la siguiente capa. Para crear una red neuronal convolucional profunda, se apila múltiples capas, alternando entre capas convolucionales y de agrupación, antes de conectarlas a un perceptrón multicapa para realizar la clasificación (Raschka 2015).

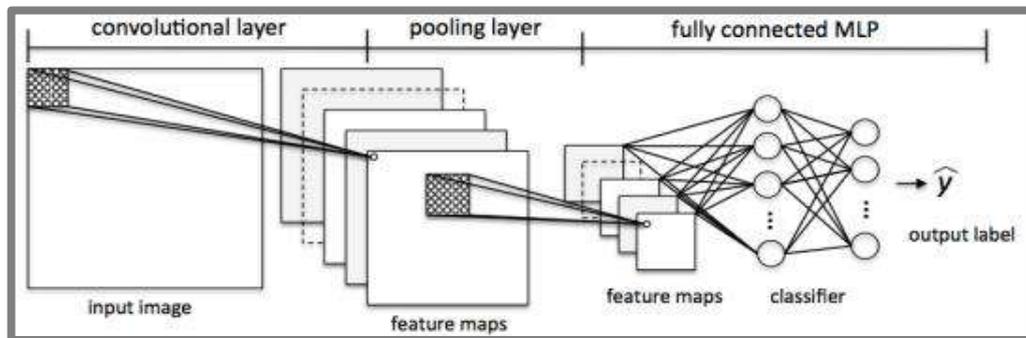


Ilustración 2-23: Creación de una red neuronal convolucional.

Fuente: (Raschka 2015, p. 383).

Redes neuronales recurrentes (RNN): Se consideran como redes neuronales de avance o propagación hacia adelante (*feedforward*) con bucles de retroalimentación (*feedback*) a través del tiempo. En las RNN, las neuronas se activan durante un tiempo limitado antes de desactivarse temporalmente. A su vez, estas neuronas activan otras neuronas que se activan en un instante posterior en el tiempo. Básicamente, las redes neuronales recurrentes pueden considerarse como perceptrones multicapa con una variable de tiempo adicional. La estructura dinámica y el componente temporal permiten que la red utilice no solo las entradas actuales, sino también las anteriores. Diseñadas para manejar datos secuenciales, como texto o series temporales. Son efectivas en tareas como procesamiento de lenguaje natural, traducción automática y generación de texto (Raschka 2015).

La Ilustración 2-24 muestra el proceso que conlleva a la creación de una red neuronal recurrente. Estas arquitecturas de red son más difíciles de entrenar, debido a que no se puede realizar la retropropagación (*backpropagate*) del error capa por capa, ya que existe el componente adicional

de tiempo, que impide calcular los gradientes necesarios durante el entrenamiento de manera correcta (Raschka 2015).

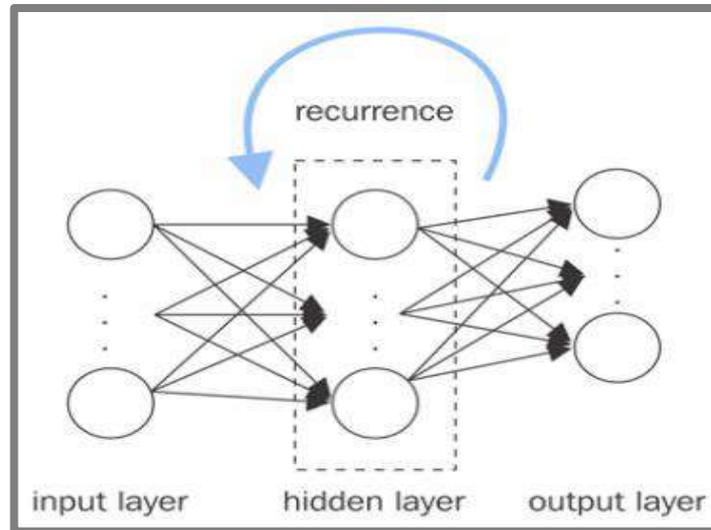


Ilustración 2-24: Creación de una red neuronal recurrente.

Fuente: (Raschka 2015, p. 383).

CAPÍTULO III

3. MARCO METODOLÓGICO

Este trabajo busca diseñar un instrumento virtual que permita controlar un sistema automatizado para la determinación gráfica de estructuras granulares, donde se obtendrán parámetros como la aceleración de umbral y la frecuencia. En este capítulo se detallan los requerimientos necesarios para llevar a cabo el sistema solicitado, haciendo énfasis en el funcionamiento del mismo, tomando como base los requerimientos planteados para cada módulo.

3.1 Requerimientos del sistema

Tomando en cuenta el estudio bibliográfico realizado en el capítulo anterior, en cuanto al desarrollo de un instrumento virtual para identificar estructuras granulares generadas por inestabilidades hidrodinámicas, se distinguieron varios requerimientos necesarios y fundamentales para llevar a cabo el diseño de dicho instrumento. Dichos requerimientos se detallan a continuación.

3.1.1 *Requerimientos generales*

- Se requiere un conocimiento claro sobre el comportamiento de las estructuras granulares generadas en el líquido portador.
- El sistema vibratorio debe contar con dos recipientes contenedores (recipiente cilíndrico y celda cuasi 2D) para la realización de pruebas, que consisten en generar distintos patrones a diferentes frecuencias y amplitudes de vibración.
- Se requiere de la iluminación necesaria para obtener una mejor toma de imágenes, tanto para el recipiente cilíndrico como para la celda cuasi 2D.
- Se requiere diseñar un sistema de adquisición de imágenes que cuente con una cámara de alta resolución espacial y otra con una excelente resolución temporal.
- Desarrollar un instrumento virtual con algoritmos y códigos de programación eficientes para el procesamiento de imágenes, que identifiquen y analicen los patrones generados en la superficie libre de líquido y en el lecho de partículas.

- El sistema debe poseer una interfaz gráfica que permita el monitoreo, visualización y exportación de los resultados por parte del usuario.
- El instrumento virtual debe ser sometido a diversas pruebas para garantizar la precisión y confiabilidad de los resultados, permitiendo la calibración y validación del mismo.

3.2 Concepción de la arquitectura

El sistema de desarrollo virtual consta de diferentes partes, entre las cuales se encuentran cuatro módulos que abarcan el desarrollo del instrumento virtual, que son: *generación de vibraciones*, *generación de muestras*, *adquisición de datos*, *procesamiento y visualización*. La Ilustración 3-1 muestra una concepción clara de la arquitectura a desarrollar.

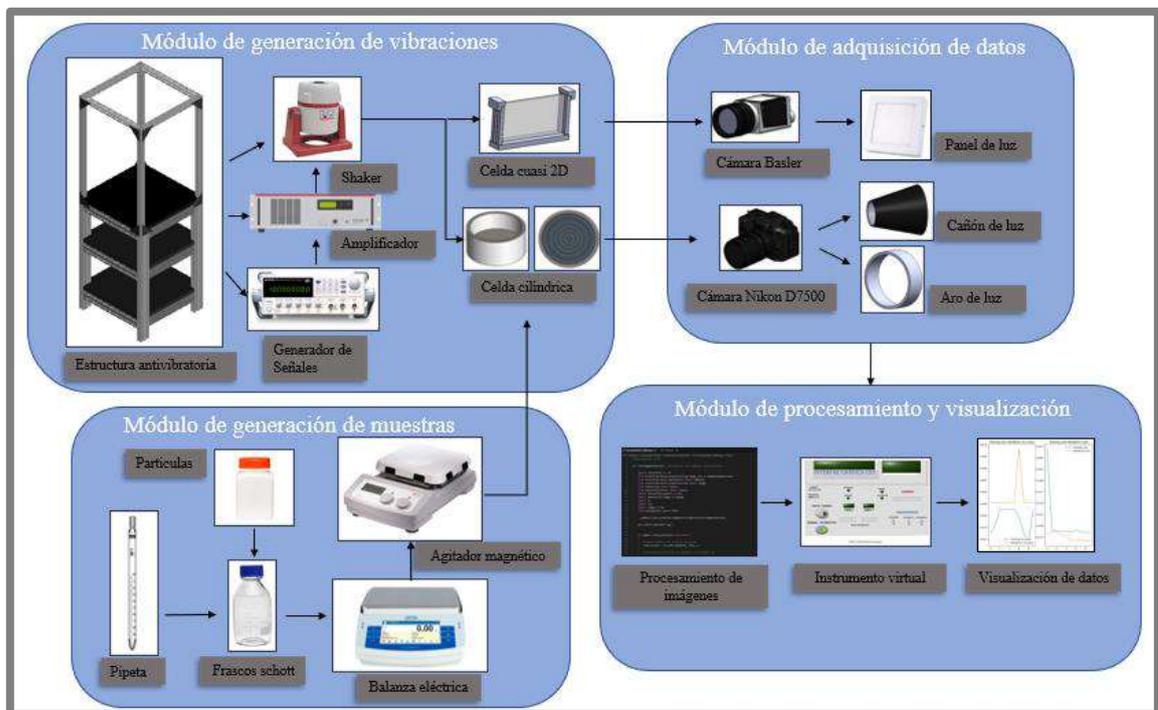


Ilustración 3-1: Concepción de la arquitectura del dispositivo.

Realizado por: Argos, J.; Granizo, I., 2023.

El módulo de generación de vibraciones es el encargado de generar las inestabilidades hidrodinámicas, en este caso vibraciones verticales, para obtener patrones a diferentes frecuencias y aceleraciones dentro de las celdas contenedoras. Está compuesto de una estructura antivibratoria en la que se alojan los distintos dispositivos a utilizar, que son un agitador magnético permanente para generar vibraciones, un amplificador de señales para aumentar el nivel de la onda enviada por un generador de señales, y las celdas contenedoras.

El módulo de generación de muestras es el encargado de producir la muestra adecuada para la sedimentación de partículas en las celdas contenedoras. Para la celda cilíndrica, a base de mezclar distintos porcentajes de agua destilada y glicerina junto con las partículas de poliestireno insoluble, se obtiene la formación de estructuras granulares al someter la mezcla a diversas frecuencias y aceleraciones de vibración. En la celda cuasi-2D, la muestra se realiza con etanol al 100% junto con las partículas de poliestireno para la visualización de las ondas de Faraday con vista lateral y el efecto que estas producen sobre las partículas.

El módulo de adquisición de datos se encarga de obtener las imágenes necesarias para llevar a cabo el procesamiento de las mismas, el módulo cuenta con dos cámaras, cada una con una función diferente. Además, para que las imágenes tengan la mayor resolución posible, el dispositivo cuenta con dos fuentes de luz de led fría para la celda cilíndrica, un cañón de luz que funciona a 12Vcc para la superficie libre de líquido y un anillo de luz que se alimenta con 8 Vcc para la visualización de patrones en el lecho de partículas. Para la iluminación de la celda cuasi 2D, el dispositivo cuenta con un panel de luz led que funciona con 110 Vca.

El módulo de procesamiento y visualización está realizado netamente bajo programación de lenguaje tipo gráfico y programación basada en redes neuronales convolucionales, en el que consta toda la programación para procesar las imágenes obtenidas mediante el diseño del instrumento virtual, reemplazando dispositivos físicos que tienen la misma utilidad. Este módulo es capaz de adquirir las imágenes desde una computadora y obtener datos de aceleración y frecuencia mediante el procesamiento de las mismas. Por otro lado, también se enfoca en la visualización y participación por parte del usuario, donde una interfaz gráfica se encarga de presentar los datos requeridos (aceleración, frecuencia, espesor de líquido, etc.), completando de esta manera el desarrollo de un instrumento virtual capaz de identificar gráficamente estructuras granulares que se generan a partir de inestabilidades hidrodinámicas (vibraciones verticales).

3.3 Diseño de la arquitectura de los módulos del instrumento virtual

Una vez descritos los módulos que constituyen la arquitectura del instrumento virtual, se muestra a detalle el funcionamiento y composición de cada uno mediante el uso de diagramas de bloques.

3.3.1 Módulo de generación de vibraciones

La Ilustración 3-2 muestra el diagrama de bloques que compone el módulo de generación de vibraciones, formado a partir de 5 partes o bloques fundamentales. El primer bloque consta de un generador de señales SFG-2104 que envía una señal sinusoidal a distintas frecuencias hacia el

bloque amplificador, el cual se compone de un amplificador de potencia lineal LDS LPA600 de la marca Brüel & Kjær encargado de amplificar la amplitud de onda proveniente desde el generador. Las ondas que se obtienen ponen en funcionamiento al bloque de actuación, el cual dispone de un vibrador de imán permanente LDS V406 de la marca Brüel & Kjær que genera vibraciones verticales, además cuenta con un bloque de enfriamiento para evitar un sobrecalentamiento del vibrador. Como parte final, se encuentra el bloque contenedor, donde las celdas contenedoras (cilíndrica y cuasi-2D) muestran los patrones formados en la superficie libre de líquido y en el lecho de partículas, para la obtención de estructuras granulares.

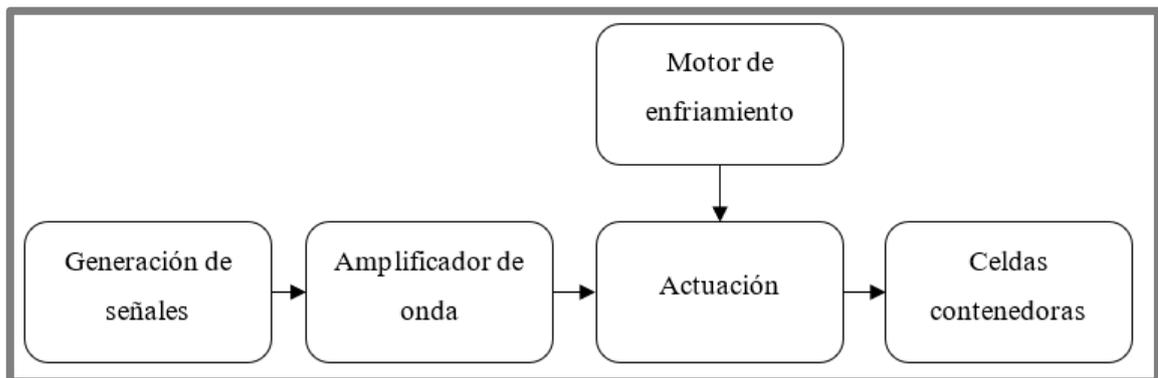


Ilustración 3-2: Diagrama de bloques del módulo de generación de vibraciones.

Realizado por: Argos, J.; Granizo, I., 2023.

3.3.2 Módulo de generación de muestras

El diagrama de bloques de la Ilustración 3-3 muestra el proceso que rige al módulo de generación de muestras. Para las muestras en la celda cilíndrica, el primer bloque es el de preparación, en el que se encuentran distintas concentraciones de agua destilada, glicerina y partículas de poliestireno insoluble, depositadas en un frasco *schott* con la ayuda de una pipeta o pera de succión. El frasco pasa por un bloque de pesaje, el cual contiene una balanza de precisión PS 4500.X2.M y se encarga de tomar el peso requerido de las muestras. Finalmente, el bloque de mezclado y depósito contiene un agitador magnético MS7-H550-Pro y se encarga de mezclar las diferentes concentraciones de glicerina, agua destilada y partículas, para obtener como resultado una muestra final, que se deposita en la celda cilíndrica. Para la celda cuasi-2D, se realiza solamente la preparación de la muestra junto con un mezclado manual en un frasco *schott*. Luego de depositar la muestra final, se ajusta manualmente mediante una jeringa la cantidad de partículas y etanol puro a la altura deseada.

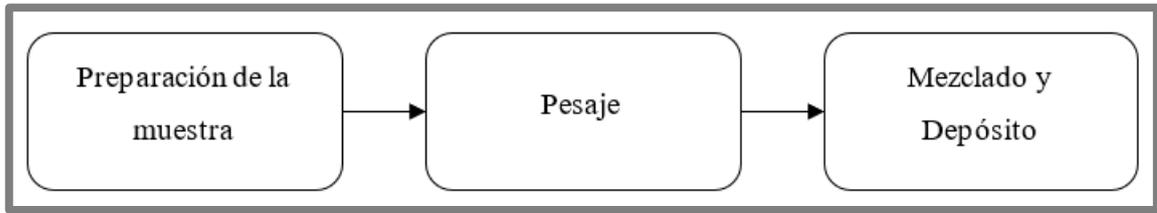


Ilustración 3-3: Diagrama de bloques del módulo de generación de muestras.

Realizado por: Argos, J.; Granizo, I., 2023

3.3.3 Módulo de adquisición de datos

La Ilustración 3-4 muestra el diagrama de bloques que dicta el proceso que se lleva a cabo en el módulo de adquisición de datos. Como bloque inicial está el denominado generación de estructuras granulares, donde las muestras depositadas en las celdas (cilíndrica o *cuasi-2D*) forman patrones al ser sometidas a vibraciones verticales, tanto en la superficie libre de líquido como en el lecho de partículas (estructuras granulares). El bloque de adquisición se encarga de tomar imágenes de las estructuras granulares, utilizando dos dispositivos fotográficos que son una cámara NIKON D7500 de alta resolución espacial para la toma de imágenes de los patrones formados en la celda cilíndrica (con vista espacial) y otra cámara BASLER acA640.750um de alta resolución temporal para la toma de imágenes con vista lateral de los patrones formados en la celda *cuasi-2D*. Para una mejor toma de imágenes, se utiliza el bloque de iluminación, el cual contiene una superficie reflectante y dos fuentes de luz led fría para la celda cilíndrica, que consisten en un cañón de luz para la superficie libre de líquido y un aro de luz led para el lecho de partículas; para la celda *cuasi-2D* se utiliza un panel de luz led fría. Los datos adquiridos por el bloque de adquisición, son enviados por los dispositivos fotográficos hacia un dispositivo de almacenamiento a través de un cable de conexión tipo USB, este proceso se lleva a cabo bajo las exigencias del bloque de transmisión de información.

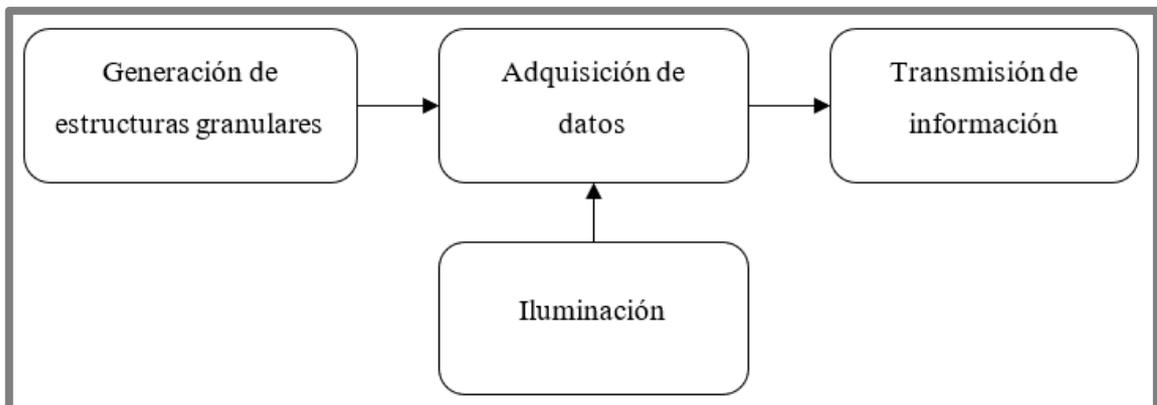


Ilustración 3-4: Diagrama de bloques del módulo de adquisición de datos.

Realizado por: Argos, J.; Granizo, I., 2023

3.3.4 Módulo de procesamiento y visualización

La Ilustración 3-5 representa el diagrama de bloques del módulo de procesamiento y visualización, el cual cuenta con 3 bloques para su correcto funcionamiento y desarrollo. El bloque de adquisición y almacenamiento de imágenes se encarga de recibir las imágenes enviadas desde el módulo de adquisición de datos y almacenarlas, este proceso se lo realiza mediante el software LabVIEW. Una vez adquiridas y/o almacenadas las imágenes, estas deben ser procesadas digitalmente, para lo cual se tiene el bloque de procesamiento que cuenta con un modelo creado de red neuronal convolucional en el lenguaje de programación Python, para llevar a cabo la clasificación, análisis y predicción de las imágenes mediante el entrenamiento de la red. Como bloque final está el bloque de visualización, el cual se encarga del diseño de una interfaz gráfica que muestra los resultados requeridos, acordes a la magnitud del procesamiento realizado, dicho diseño se realiza en el software de LabVIEW.

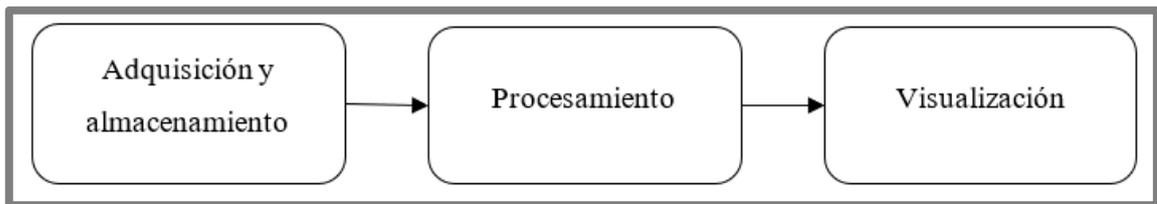


Ilustración 3-5: Diagrama de bloques del módulo de procesamiento y visualización.

Realizado por: Argos, J.; Granizo, I., 2023

3.4 Dispositivos estructurales y de hardware

La Tabla 3-1 muestra los elementos que conforman todo el sistema, tanto para la generación de las estructuras granulares, como para la adquisición y procesamiento de imágenes.

Tabla 3-1: Dispositivos estructurales y de hardware utilizados para la obtención de estructuras granulares y procesamiento de imágenes.

Cantidad	Dispositivo	Modelo	Descripción
1	Estructura antivibratoria		Evita vibraciones externas a las producidas por el Shaker.
1	Generador de señales	SFG-2104	Genera ondas sinusoidales a distintas frecuencias.
1	Vibrador de imán permanente (Shaker)	LDS V406	Produce vibraciones verticales.

Cantidad	Dispositivo	Modelo	Descripción
1	Amplificador de potencia lineal	LDS LPA600	Amplifica la señal de onda recibida por el generador de señales.
1	Balanza de precisión	PS 4500.X2.M	Muestra el peso de las muestras con la mayor precisión posible.
1	Agitador magnético	MS7-H550-Pro	Produce agitaciones magnéticas para mezclar distintas muestras.
1	Cámara de alta resolución espacial	NIKON D7500	Cámara con gran nitidez y rapidez de procesamiento de imágenes.
1	Cámara de alta resolución temporal	BASLER acA640.750um	Cámara que ofrece 751 fotogramas por segundo con resolución VGA (640x480 píxeles).

Realizado por: Argos, J.; Granizo, I., 2023

3.4.1 Selección de los dispositivos estructurales y de hardware

En esta sección se describen los distintos componentes estructurales y de hardware a ser utilizados en el desarrollo y obtención tanto de las estructuras granulares como de los parámetros a ser determinados gráficamente por el instrumento virtual.

3.4.1.1 Estructura antivibratoria

Según Jaramillo y Marcatoma (2023), la estructura antivibratoria está diseñada a base de perfiles de aluminio estructurado, que sirven como soporte para superficies de madera de laurel de 5 cm de grosor, según se observa en la Ilustración 3-6. Para mayor información, revise el anexo A.

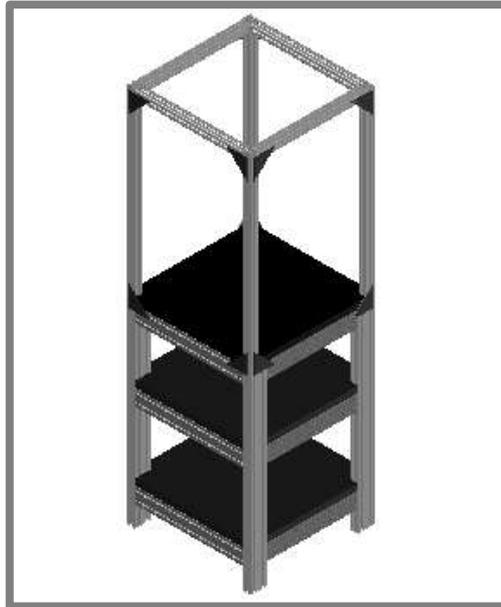


Ilustración 3-6: Estructura antivibratoria.

Fuente: (Jaramillo y Marcatoma 2023).

3.4.1.2 *Generador de funciones*

La Ilustración 3-7 muestra un generador de señales de la marca GW Instek modelo SFG-2104, construido con un rendimiento alto para la generación de ondas con frecuencias de salida estables, baja distorsión y una resolución de frecuencia fina (TEquipment 2017).



Ilustración 3-7: Generador de señales SFG-2104.

Fuente: (TEquipment 2017).

Las características principales del dispositivo se muestran en la tabla 3-2. Para mayor información, consulte el anexo B.

Tabla 3-2: Especificaciones técnicas del generador de señales SFG-2104.

Parámetros	Especificaciones
Rango de frecuencia	0.1 Hz – 4 MHz
Precisión de alta frecuencia	±20 ppm
Estabilidad de alta frecuencia	±20 ppm
Resolución de frecuencia	100 MHz
Onda sinusoidal de baja distorsión	-55 dBc, 0.1 Hz – 200 kHz

Realizado por: Argos, J.; Granizo, I., 2023

3.4.1.3 Amplificador de potencia lineal

El amplificador de potencia lineal LDS LPA600 de la marca Brüel & Kjær que aparece en la Ilustración 3-8, es una unidad independiente, compacta y eficiente, diseñada para impulsar el agitador de imán permanente, ofreciendo un rendimiento de bajo ruido y baja distorsión (Brüel & Kjaer 2021).



Ilustración 3-8: Amplificador de potencia lineal LDS LPA600.

Fuente: (Brüel & Kjaer 2021).

La Tabla 3-3 muestra las características técnicas del amplificador de potencia lineal LDS LPA 600. Para mayor información, consulte el anexo C.

Tabla 3-3: Especificaciones técnicas del amplificador de potencia lineal LDS LPA 600.

Parámetros	Especificaciones
Rango de potencia	154 VA – 1296 VA
Rangos de frecuencia de DC	15 kHz
Distorsión armónica total	< 0.1 %
Relación señal-ruido	> 95 dB
Parámetros	Especificaciones
Tipo de clasificación para eficiencia	Clase B

Realizado por: Argos, J.; Granizo, I., 2023

3.4.1.4 Vibrador de imán permanente

La Ilustración 3-9 muestra un vibrador electrodinámico de imán permanente de la marca Brüel & Kjær modelo LDS V406, usado para realizar ensayos de vibración de pequeños componentes y sistemas. Además, tiene varias aplicaciones en experimentos de laboratorio, estudios modales y análisis estructural (BRÜEL & KJÆR 2021).



Ilustración 3-9: Vibrador de imán permanente LDS V406.

Fuente: (BRÜEL & KJÆR 2021).

La Tabla 3-4 muestra las características técnicas del dispositivo en cuestión. Para mayor información, consulte el anexo D.

Tabla 3-4: Características específicas y técnicas principales del vibrador de imán permanente LDS V406.

Parámetros	Especificaciones
Pico de fuerza sinusoidal	98 N (22 lbf)
Pico de fuerza sinusoidal (con el amplificador LDS PA600)	196 N (44 lbf)
Máxima fuerza RMS aleatoria	89 N
Pico sinusoidal de aceleración máxima	100 gr
Pico sinusoidal de velocidad	1,78 m/s
Dimensiones (ancho x alto x profundidad)	135.5 mm x 104 mm x 72.5 mm
Masa del elemento en movimiento	0,20 kg
Rango de frecuencia utilizable	5 Hz – 9 kHz

Realizado por: Argos, J.; Granizo, I., 2023

3.4.1.5 Balanza de precisión

La Ilustración 3-10 muestra la balanza de precisión marca RADWAG modelo PS 4500.X2.M, misma que permite pesar cargas con una precisión de 0.1 gramos con una capacidad de hasta 10 kg y se caracteriza por un método innovador de sujeción del platillo en un solo punto, garantizando muy buena geometría y minimizando el error de excentricidad, además de que su diseño garantiza un alto nivel de resistencia a la contaminación (RADWAG 2018).



Ilustración 3-10: Balanza de precisión PS 4500.X2.M.

Fuente: (RADWAG 2018).

La Tabla 3-5 muestra las principales características de la balanza de precisión. Para mayor información, consulte el anexo E.

Tabla 3-5: Características específicas y técnicas de la balanza de precisión PS 4500.X2.M.

Parámetros	Especificaciones
Precisión d	0.01 g
Nivel de resistencia a factores externos	Alto
Tipo de tecnología	MONOBLOCK
Tipo de memoria	ALIBI
Capacidad de pesaje	10 kg
Rango máximo de pesaje	4500 kg
Rango mínimo de pesaje	0.5 g

Realizado por: Argos, J.; Granizo, I., 2023

3.4.1.6 Agitador magnético

El agitador magnético de la marca DLAB con placa de calentamiento MS7-H550-Pro que se muestra en la Ilustración 3-11, combina una placa vitrocerámica, un motor sin escobillas y un

sensor de temperatura externo, para el desarrollo de aplicaciones que requieran de síntesis química (Labomersa 2018).



Ilustración 3-11: Agitador magnético MS7-H550-Pro.

Fuente:(Labomersa 2018).

La Tabla 3-6 muestra las principales características del agitador magnético. Para mayor información, consulte el anexo F.

Tabla 3-6: Características específicas y técnicas del agitador magnético MS7-H550-Pro.

Parámetros	Especificaciones
Dimensión de la placa de trabajo	184x184 mm (7 plg)
Rango de temperatura de calentamiento	Temperatura ambiente-550 °C
Poder de calefacción	1000 W
Voltaje	100-120/200-240 V, 50/60 Hz
Rango de velocidad	100-1500 rpm
Barra magnética	80 mm

Realizado por: Argos, J.; Granizo, I., 2023

3.4.1.7 Cámara de alta resolución espacial

Para la toma de imágenes de las estructuras granulares contenidas en la celda cilíndrica, se utiliza una cámara fotográfica NIKON D7500, la cual tiene un enfoque muy preciso, con un rango dinámico de gran tamaño y una calidad de imagen consistente (Nikon 2018). La Ilustración 3-12 muestra el modelo de la cámara.



Ilustración 3-12: Cámara NIKON D7500.

Fuente: (Nikon 2018).

La Tabla 3-7 muestra las características técnicas más importantes de la cámara NIKON D7500. Para mayor información, consulte el anexo G.

Tabla 3-7: Especificaciones técnicas de la cámara NIKON D7500.

Parámetros	Especificaciones
Tipo de sensor	CMOS APS-C (23.5x15.6 mm)
Píxeles efectivos (megapíxeles)	20.9 millones
Tamaño del sensor	23.5mm x 15.6mm
Tamaño de pantalla	3.2 pulgadas en diagonal
Medios de almacenamiento	SD, SDHC, SDXC
Dimensiones (ancho x alto x profundidad)	135.5 mm x 104 mm x 72.5 mm

Realizado por: Argos, J.; Granizo, I., 2023

3.4.1.8 Cámara de alta resolución temporal

Para la toma de imágenes con vista lateral en el recipiente cuasi-2D, se utiliza una cámara BASLER acA640.750um que cuenta con un control automático de exposición, agrupamiento de píxeles, duplicación de imágenes horizontales, entre otras características. Además, los datos de píxeles se generan con una profundidad de 8 o 12 bits, permitiendo ajustar el nivel de negro de la cámara (Basler 2015). La Ilustración 3-13 muestra el modelo de la cámara nombrada anteriormente.

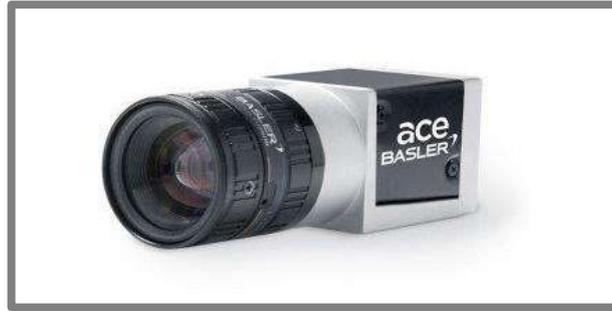


Ilustración 3-13: Cámara BASLER acA640.750um.

Fuente: (Basler 2015).

La Tabla 3-8 describe las principales características que posee esta cámara. Para mayor información, consulte el anexo H.

Tabla 3-8: Especificaciones técnicas de la cámara BASLER acA640.750um.

Parámetros	Especificaciones
Tipo de sensor	EN SemiPÝTHON 300 CMOS
Resolución de los sensores (píxeles)	640 x 480
Velocidad de fotogramas	750.0 frames/segundo
Rango dinámico	56.6 dB
Tipo de obturador	Global
Velocidad de datos	USB 3.0 (5 Gbps)

Realizado por: Argos, J.; Granizo, I., 2023

3.4.2 Dispositivos de iluminación

Las diferentes fuentes de luz led fría se realizaron acorde a la necesidad de obtener una buena visualización y claridad, tanto de la superficie libre de líquido como de los patrones formados con partículas. La Ilustración 3-14 muestra dos fuentes de iluminación: el cañón de luz que funciona con 12 Vcc de entrada usado para la iluminación de la superficie libre de líquido con ayuda de una superficie reflectante y el aro de luz que se encarga de revelar de los patrones formados dentro del líquido portador y funciona con 8 Vcc de entrada; ambas fuentes de iluminación, tanto el cañón como el aro de luz, asisten a las necesidades del recipiente cilíndrico.

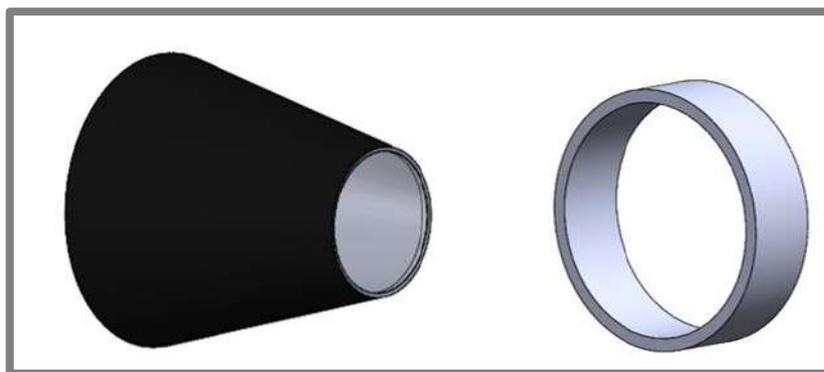


Ilustración 3-14: Izquierda: Cono de luz led fría. Derecha: Aro de luz led fría

Realizado por: Argos, J.; Granizo, I., 2023.

Para la celda *cuasi* 2D, la iluminación se realiza con un panel de luz led fría que funciona a 110 Vac, tal y como se muestra en la Ilustración 3-15.



Ilustración 3-15: Panel de luz led.

Realizado por: Argos, J.; Granizo, I., 2023.

3.4.3 Celdas contenedoras

Tanto la celda cilíndrica como la celda *cuasi*-2D fueron diseñadas a base de acrílico, en el caso de la celda cilíndrica, posee unas dimensiones de 90 mm de diámetro externo, 78 mm de diámetro interno y 40 mm de altura (30 mm de altura interna y 10 mm de altura de la base), y tiene como función principal contener la muestra de partículas, permitiendo visualizar de manera espacial e independiente los patrones formados en la superficie libre de líquido y en el lecho de partículas. La celda *cuasi*-2D posee dimensiones de 75 mm de altura, 55 mm de largo, 3 mm de ancho de cada placa de acrílico y un espacio interno de 2 mm, y permite una vista lateral de la interfaz de las partículas sedimentadas y la capa de líquido sobrenadante a ellas. La Ilustración 3-16 muestra ambas celdas contenedoras.

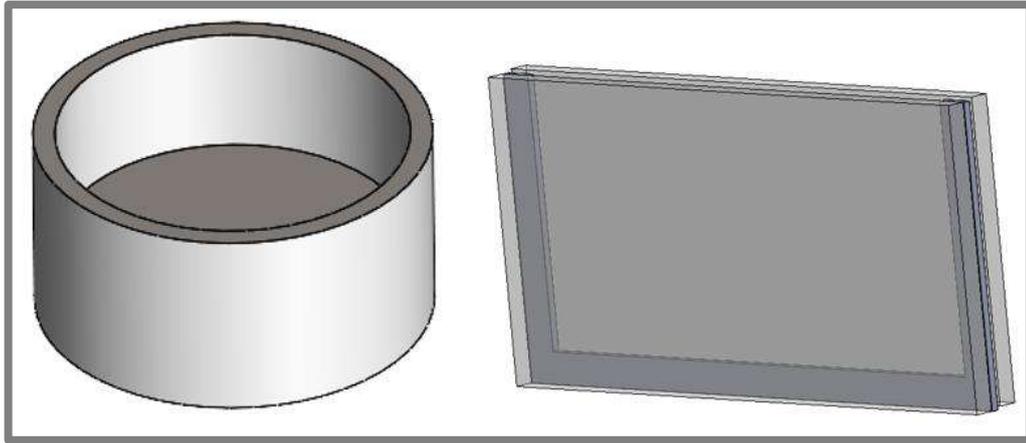


Ilustración 3-16: Izquierda: Celda cilíndrica. Derecha: Celda *cuasi* 2D.

Realizado por: Argos, J.; Granizo, I., 2023.

3.5 Selección de las herramientas de software

Para el desarrollo del instrumento virtual, se identificaron los distintos softwares a ser utilizados para cada etapa. Dichos softwares se detallan a continuación.

3.5.1 *LabVIEW*

El entorno de programación *LabVIEW* simplifica en gran manera la integración de hardware para diversas aplicaciones de ingeniería, ya que permite adquirir datos desde hardware propio de NI y de terceros (National Instruments 2020).

La Ilustración 3-17 muestra un entorno gráfico desarrollado en *LabVIEW* que permite el control y visualización de datos. Este software permite desarrollar un instrumento virtual de gran robustez gracias a su capacidad de interacción con los distintos elementos de hardware y software para la toma de imágenes claras y precisas y para llevar a cabo el procesamiento digital de las mismas. Además, en caso de ser necesario *LabVIEW* puede interactuar con distintos lenguajes de programación.

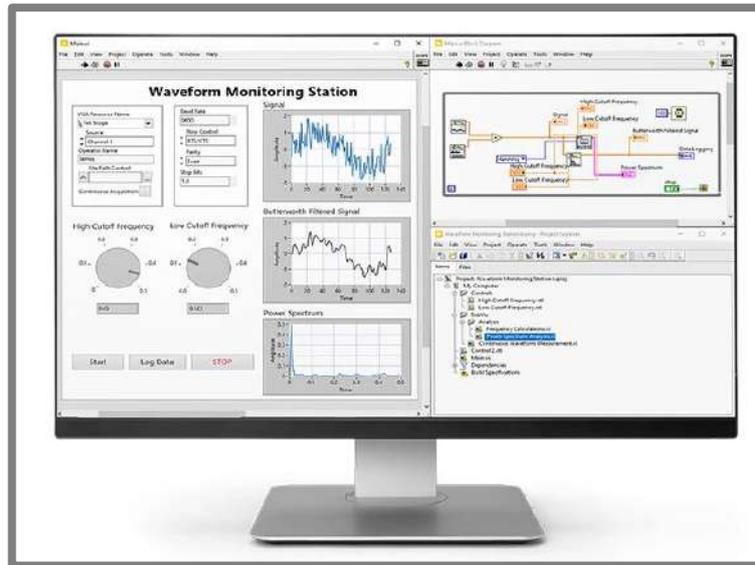


Ilustración 3-17: Entorno de trabajo y visualización de LabVIEW.

Fuente: (National Instruments 2020).

3.5.2 Google Colab

Google Colab (Colaboratory) es un servicio en línea alojado de Jupyter Notebook que no necesita configuración alguna para su uso, ya que brinda acceso gratuito a recursos informáticos, incluyendo GPU y TPU (Google 2023).

La Ilustración 3-18 muestra la ventana principal de la plataforma de *Google Colab*, cuyo lenguaje de programación es *Python*. Permite la utilización de otras herramientas como *Google Drive*, *BigQuery*, *Google Sheets* y una herramienta denominada *TensorFlow* que se mostrará más adelante. Debido al fácil acceso que proporciona a recursos de cómputo de alta calidad como GPUs (Unidades de procesamiento gráfico), se puede acelerar significativamente el entrenamiento de modelos de aprendizaje profundo, a diferencia de una computadora convencional que no posee grandes características de hardware, donde el entrenamiento efectivo de una red neuronal puede superar las capacidades de la misma y presentar errores de entrenamiento.

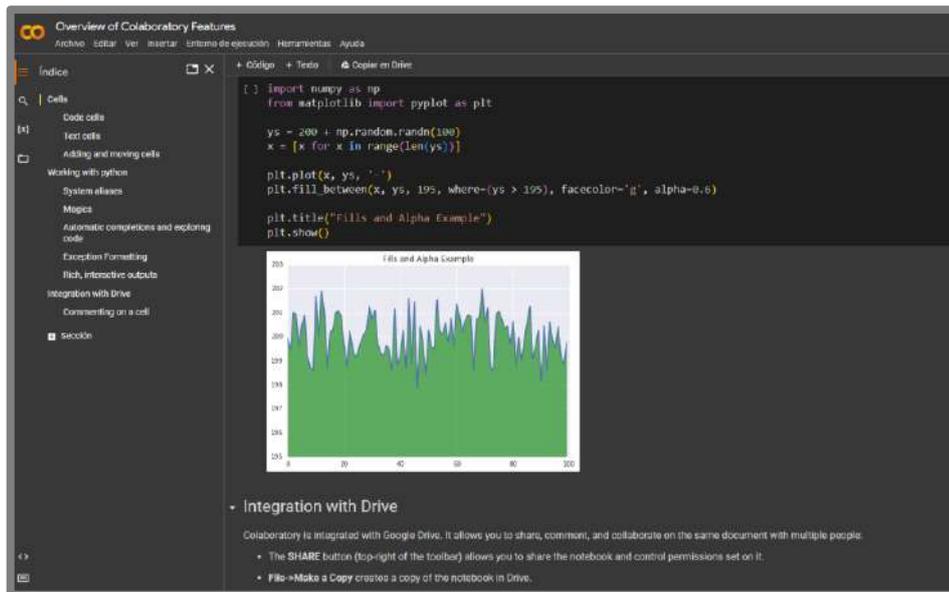


Ilustración 3-18: Entorno de trabajo de Google Colab

Realizado por: Argos, J.; Granizo, I., 2023.

3.5.3 *Visual Studio Code*

Visual Studio Code es un editor de código fuente, ligero y a la vez potente, que puede ejecutarse desde el escritorio de una computadora y está disponible para Windows, macOS y Linux. Incluye soporte para la depuración, finalización inteligente de código, fragmentos y refactorización de código, control integrado de Git y resaltado de sintaxis (Visual Studio 2015).

Visual Studio Code es un entorno de desarrollo integrado (IDE) para varios lenguajes de programación, incluyendo Python. Debido a que Google Colab no puede interactuar directamente con LabVIEW se necesita de un intérprete, que en este caso es Visual Studio Code, no solo porque permite una rápida y fácil gestión de las librerías de Python, sino también por su facilidad de uso y rápida comprensión de código. En la Ilustración 3-19 se observa este entorno de programación, donde estará contenido el código necesario para invocar a LabVIEW y cargar el modelo entrenado en Google Colab, para posteriormente enviar la predicción de imágenes y los resultados (aceleración, frecuencia y porcentaje de exactitud) hacia LabVIEW junto con la base de datos empleada en el entrenamiento de la red neuronal convolucional.

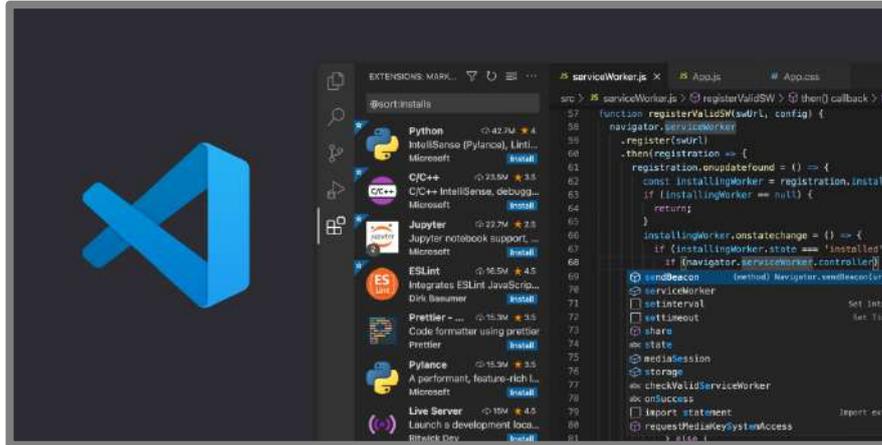


Ilustración 3-19: Entorno de trabajo de Visual Studio Code

Realizado por: (Visual Studio 2015).

3.6 Preparación de muestras

3.6.1 Celda cilíndrica

Partículas: Se utilizaron 2 tamaños diferentes de esferas de poliestireno insoluble que son $80\ \mu\text{m}$ y $140\ \mu\text{m}$ de diámetro.

Líquidos: Las sustancias utilizadas para la preparación de las muestras son la glicerina ($C_3H_8O_3$) y agua destilada (H_2O), ya que son solubles entre sí y se puede crear una mezcla homogénea sin mayor preocupación. Además, debido a que la glicerina es un líquido altamente viscoso en comparación con el agua, se puede utilizar esta característica para controlar la interacción de las partículas y ajustar la velocidad a la que se mueven en la mezcla para organizarlas en patrones específicos. Además, para que se produzca la sedimentación total de las partículas dentro de la mezcla, se necesita romper la tensión superficial de la misma, para lo cual se utiliza un surfactante. En este caso, se utilizó un detergente líquido comercial.

3.6.1.1 Proporción de los componentes

La nomenclatura que define el tipo de muestra es la siguiente:

$$M_{C_G-\phi-D_P}$$

Donde C_G representa la concentración de glicerina (en %) y ϕ equivale a la fracción volumétrica o cantidad de partículas (en %), ambos con respecto al agua destilada. D_P es el diámetro de las partículas en μm .

Para hallar C_G se utiliza la Ecuación 1:

Ecuación 1: Concentración de glicerina.

$$C_G \% = \frac{M_G}{M_G + M_{H_2O}} * 100 \quad (1)$$

Donde M_G representa la masa de la glicerina (en gr) y M_{H_2O} la masa del agua (en gr).

La fracción volumétrica se determina mediante la Ecuación 2:

Ecuación 2: Fracción volumétrica.

$$\phi \% = \frac{M_{particulas}}{M_{total}} * 100 \quad (2)$$

Donde $M_{particulas}$ representa la masa de las partículas y M_{total} la masa total de la muestra ($M_G + M_{H_2O}$) en gramos.

El proceso que rige la preparación de las muestras se describe a continuación:

1. Colocar el frasco *shott* sobre la balanza de precisión (antes de colocar el líquido se debe tarar la balanza) y con la ayuda de una pipeta vaciar 50 gramos de agua destilada en el mismo.
2. Realizar una nueva calibración de la balanza y, utilizando una pipeta, transferir con precisión la cantidad calculada de glicerina.
3. Colocar el frasco cerrado en el agitador magnético y ajustar la velocidad a 200 rpm y la temperatura a 25°C para mezclar ambos líquidos con la ayuda de un buzo magnético.
4. Con la ayuda de una espátula pequeña se sobreponen las partículas sobre un trozo de papel aluminio y se mide la cantidad de partículas calculada con anterioridad con la ayuda de la balanza de precisión.

5. Colocar las partículas en el frasco que contiene la mezcla de agua y glicerina, junto con unas gotas (máximo 4) de surfactante y mezclar todo el contenido durante 5 minutos aproximadamente.
6. Finalmente, con la ayuda de una pipeta colocar la cantidad requerida de la muestra en el recipiente cilíndrico.

3.6.1.2 Análisis de las muestras

La manera de obtener un análisis concreto de las muestras, fue sometiéndolas dentro de la celda cilíndrica a vibraciones verticales a frecuencias de entre 40 a 60 Hz con un espesor de líquido de 5ml, con aceleraciones variantes hasta encontrar un patrón adecuado en el lecho de partículas que concuerde lo mejor posible con su homónimo en la superficie libre de líquido.

Muestra $M_{15-6-80}$

Se obtuvieron mejores resultados en la formación de estructuras granulares al trabajar con la muestra de nombre $M_{15-6-80}$ (15% de C_G , 6% de ϕ , $D_p = 80 \mu m$).

Teniendo como base 50 gramos de agua destilada, se obtuvo la masa de glicerina con una concentración del 15 % en base a los siguientes cálculos:

Ecuación 3: Cálculo de masa de glicerina.

$$15 = \frac{M_G}{M_G + 50} * 100 \quad (3)$$

Ecuación 4: Masa de glicerina.

$$M_G = 8.8235 \text{ gr} \quad (4)$$

Y el 6 % de cantidad de partículas equivale a

Ecuación 5: Cálculo de masa de partículas.

$$\frac{6}{100} = \frac{M_{particulas}}{50 + 8.8235} \quad (5)$$

Ecuación 6: Masa de partículas.

$$M_{particulas} = 3.5294 \text{ gr} \quad (6)$$

La Tabla 3-9 expone los tipos de muestras en los que se obtuvieron mejores resultados a la hora de obtener estructuras granulares.

Tabla 3-9: Datos obtenidos de las pruebas realizadas con distintos tipos de muestras.

Tipo de muestra	D_p (μm)	Masa H_2O (gr)	Masa de Glicerina (gr)	Masa partículas (gr)	Resultados
$M_{9-4-140}$	140	50	4.9	2.359	Patrones inestables
$M_{15-6-140}$	140	50	8.824	3.529	Patrones porosos
$M_{15-6-80}$	80	50	8.824	3.529	Patrones compactos

Realizado por: Argos, J.; Granizo, I., 2023.

La Ilustración 3-20 muestra una comparación entre las muestras $M_{15-6-80}$ y $M_{15-6-140}$, donde la única diferencia es el diámetro de las partículas. Ambas muestras fueron sometidas a vibraciones con una frecuencia de 40 Hz y una aceleración de 1 g en 5 ml de espesor de líquido. La diferencia se hace notable en los patrones de anillos, donde las partículas de 80 μm forman una estructura más densa, es decir que las partículas están unidas de forma compacta, en cambio las partículas de 140 μm forman estructuras porosas y dispersas, generando pequeños espacios entre la aglomeración de las mismas, lo cual es inadecuado para el estudio de los patrones.



Ilustración 3-20: Izquierda: Patrones con la muestra $M_{15-6-80}$.

Derecha: Patrones con la muestra $M_{15-6-140}$.

Realizado por: Argos, J.; Granizo, I., 2023.

Por lo tanto, la muestra utilizada para la creación de estructuras granulares en la celda cilíndrica es la que lleva de nombre $M_{15-6-80}$.

3.6.2 Celda cuasi 2D

Líquido y partículas: La sustancia utilizada en este caso es el etanol, ya que puede ser afectado por las vibraciones de manera efectiva debido a su baja viscosidad y su compatibilidad con las partículas ($80 \mu m$) permite observar de mejor manera la interacción entre el lecho de partículas y el líquido.

3.6.2.1 Proporción de los componentes

Para la preparación de la muestra que irá contenida en la celda cuasi 2D se optó por no emplear porcentajes, sino que se llevó a cabo un ajuste manual de la cantidad de etanol y de partículas. Para esto, es necesario mezclar una cantidad aleatoria de etanol y de partículas en un frasco *shott* sin la ayuda de instrumentos de medición, posteriormente, se agita manualmente el recipiente para lograr una mejor homogeneización. Con la ayuda de una jeringa, se introduce la cantidad requerida de la muestra en la celda cuasi 2D y esperando el tiempo adecuado para que las partículas sedimenten, se controla el nivel de líquido con la ayuda de otra jeringa.

El espesor de la capa de partículas h_p es constante, manteniéndose siempre en 10 mm, para la corrección del mismo se utiliza una jeringa para succionar la cantidad excedente.

3.6.2.2 Ajuste de proporciones

Una vez completa la operación de ajustar el nivel de espesor de la capa de partículas h_p en 10mm dentro de la celda *cuasi 2D*, se debe nivelar con el mayor cuidado posible el espesor de la capa de líquido Δh . Para el presente trabajo, el nivel de Δh empieza en 20 mm, que cambia a 10 mm una vez acabada la sesión de pruebas, luego el nivel disminuye en un milímetro hasta llegar a 1. La Ilustración 3-21 muestra los valores de Δh y h_p en un estado estable o de equilibrio, momentos antes de realizar las pruebas.

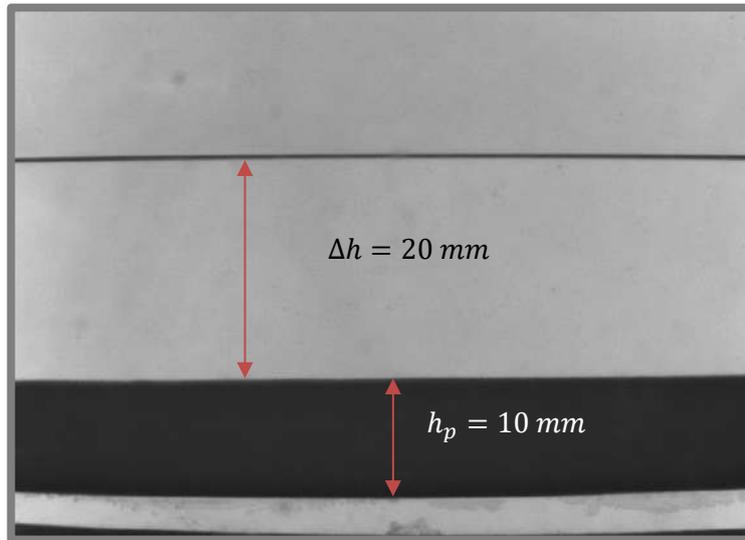


Ilustración 3-21: Depósito de la muestra en la celda *cuasi* 2D.

Realizado por: Argos, J.; Granizo, I., 2023.

3.7 Diseño y desarrollo del instrumento virtual

Durante la conversación con el autor de la tesis doctoral titulada “Dinámica de suspensiones concentradas sometidas a vibración mecánica”, se identificaron diversos elementos que podrían enriquecer futuras investigaciones sobre estructuras granulares. Se considera pertinente que el instrumento virtual que se está desarrollando incorpore los siguientes aspectos:

- La capacidad de recortar regiones de interés donde se ubiquen las estructuras granulares.
- La medición de la longitud de onda en las celdas cilíndricas y en las celdas cuasi 2D.
- El conteo del número de patrones en las estructuras, así como la identificación de anillos en formación de anillos y diamantes en formación de diamantes.
- La determinación del porcentaje de ocupación de las estructuras de diamantes.
- La medición del tamaño de las estructuras de diamantes.
- La representación de imágenes de estructuras de anillos en coordenadas polares.
- La capacidad de convertir unidades de píxeles a milímetros.

- La posibilidad de visualizar el ciclo de onda en la celda 2D a través de una compilación de imágenes.

Estos son solo algunos de los elementos que se han identificado y que se buscan implementar en el instrumento virtual. Además, el objetivo principal es que tenga la capacidad de determinar la frecuencia, aceleración y espesor del líquido con el que se ha generado la estructura granular.

3.7.1 Creación, entrenamiento y validación de la red neuronal convolucional

La elección de utilizar redes neuronales convolucionales (CNN) se fundamenta en su gran capacidad para clasificar imágenes y procesar datos visuales. Además, las CNN pueden identificar patrones de imágenes de manera precisa, presentando una gran ventaja para lograr el objetivo de analizar las estructuras granulares generadas por inestabilidades hidrodinámicas.

En esta etapa, se muestra el proceso que se lleva a cabo para la creación de la red neuronal convolucional, incluyendo las librerías necesarias, el entrenamiento de la red y su evolución a lo largo de distintas épocas. La red neuronal se encarga de reconocer, clasificar y predecir los patrones contenidos en cada imagen.

3.7.1.1 Base de datos

Para la creación de cualquier tipo de red neuronal (clásica, profunda, convolucional, etc.), inclusive para aplicaciones que requieran de un aprendizaje automático (*Machine Learning*), la mayor parte de los recursos (económicos, tiempo y espacio) son destinados a la obtención de imágenes para establecer una base de datos que se emplea en el proceso de entrenamiento de estos modelos computacionales.

Los datos son de tipo numérico ya que las imágenes se representan con datos de píxeles, además, la base de datos debe contener etiquetas (*targets*) para diferenciar una clase de otra; estos parámetros son importantes ya que la red neuronal es entrenada para realizar una tarea de aprendizaje profundo. La Ilustración 3-22 muestra un conjunto de imágenes utilizadas para la creación de la base de datos, cada una con su respectiva clase o categoría para llevar a cabo la predicción mediante la red neuronal, en este caso, de tipo convolucional.

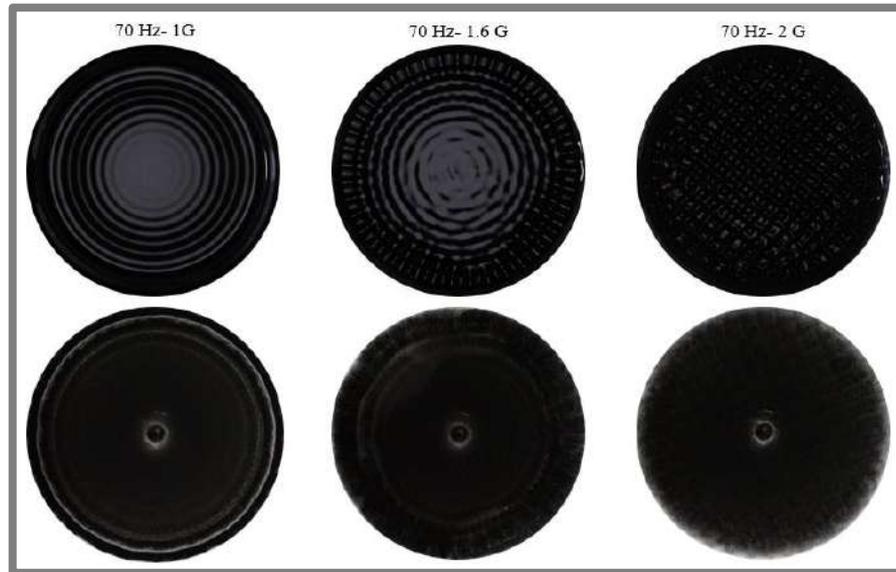


Ilustración 3-22: Ejemplo de base de datos utilizadas para el entrenamiento de la red neuronal convolucional. Superficie libre (superior) y patrones en el lecho de partículas (inferior).

Realizado por: Argos, J.; Granizo, I., 2023.

Al momento de generar patrones en el lecho de partículas (estructuras granulares) sometidas a una inestabilidad hidrodinámica (vibración vertical) en específico, existe una gran posibilidad de obtener el mismo patrón en todas las pruebas realizadas con una sola muestra; incluso en ocasiones donde la aglomeración de partículas presente algún cambio (por ejemplo, una rotación del patrón), por mínimo que sea, las partículas se moverán hasta llegar al mismo patrón. Esto genera un gran problema al momento de entrenar la red neuronal, ya que se llegaría a un término conocido como sobreentrenamiento, donde la red neuronal solo podría clasificar una imagen y descartaría cualquier otra. Para solucionar este problema, se modificaron algunas características de las imágenes como el brillo o contraste (o ambas simultáneamente), con el fin de obtener más imágenes para el entrenamiento. Cambios más notables, como la rotación, se verán más adelante junto con la librería *Tensorflow*, que ofrece la posibilidad de realizar estos cambios en la imagen de manera automática, con la misma finalidad de obtener más imágenes para entrenar la red neuronal convolucional de forma correcta.

3.7.1.2 Librerías necesarias

En la Ilustración 3-23 se observa el código utilizado para importar las librerías necesarias en el desarrollo y evaluación de la red neuronal convolucional, así como para la generación de representaciones gráficas.

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.preprocessing import image
from tensorflow import keras
from tensorflow.keras import layers
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import os
import random
import numpy as np
```

Ilustración 3-23: Librerías utilizadas para el entrenamiento de la red neuronal convolucional en el entorno de Google Colab.

Realizado por: Argos, J.; Granizo, I., 2023.

Tensorflow: En esta sección, se destaca la librería *tensorflow* desarrollada por Google, que se utiliza para la creación, entrenamiento y optimización de modelos de aprendizaje automático y redes neuronales, abarcando no solo las convolucionales, sino también las redes neuronales clásicas, profundas, recurrentes, etc.

Os: Otra librería a destacar es la denominada *os*, que sirve para gestionar las rutas de archivos y directorios, y así guardar el modelo entrenado.

Numpy: La librería *numpy* permite al modelo trabajar con valores numéricos, ya que las imágenes se representan mediante matrices y es necesario realizar operaciones numéricas con los datos de dichas matrices.

3.7.1.3 Importación de base de datos a Google Colab

Como se mencionó en el apartado 3.6.2, Google Colab puede interactuar con otras herramientas como Google Drive, que proporciona almacenamiento en la nube de forma gratuita. En este espacio se almacenan todas las imágenes destinadas al entrenamiento de la red, ya que la memoria disponible en Google Colab no es lo suficientemente extensa para almacenar un gran volumen de imágenes.

```
[ ] train_data_dir = '/content/drive/MyDrive/Base de datos final/entrenamiento'
validation_data_dir = '/content/drive/MyDrive/Base de datos final/test'

train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(width_shape, height_shape),
    batch_size=batch_size,
    #save_to_dir='',
    class_mode='categorical')

validation_generator = valid_datagen.flow_from_directory(
    validation_data_dir,
    target_size=(width_shape, height_shape),
    batch_size=batch_size,
    #save_to_dir='',
    class_mode='categorical')
```

Ilustración 3-24: Código en Google Colab para cargar los datos de entrenamiento y validación en la red neuronal convolucional.

Realizado por: Argos, J.; Granizo, I., 2023.

La Ilustración 3-24 muestra el código responsable de cargar las imágenes desde Google Drive y guardarlas en una variable. Además, una parte de la base de datos se destina al entrenamiento de la red y la parte restante se utiliza para la validación del entrenamiento. Un parámetro muy importante es el tamaño de las imágenes (*target_size*), donde se especifica la dimensión de las mismas para que no existan complicaciones en la velocidad o capacidad de entrenamiento.

3.7.1.4 Creación de la red neuronal convolucional

Una práctica muy común en el mundo de la programación es la reutilización de modelos creados por otros expertos con el fin de agilizar tareas, un ejemplo notable es la librería *TensorFlow*, creada con el fin de que pueda ser utilizada por cualquier usuario. De no poder reutilizar modelos ya existentes, la mayoría de nuevos usuarios dispuestos a innovar, se verían en la obligación de recurrir a fundamentación matemática para establecer un único parámetro en una red neuronal.

Dentro de la creación de una red neuronal, existe una técnica denominada *Transfer Learning* en la que un modelo ya estudiado, preentrenado y validado para una tarea, puede ser adaptado y reutilizado como base para realizar una tarea diferente pero relacionada y así crear un nuevo proceso computacional en el ámbito del aprendizaje automático y la inteligencia artificial.

Para la creación del modelo, se tomaron como base las capas de una red neuronal convolucional ya entrenada y aprobada por Google, conocida como *MobileNet*, que permite clasificar hasta 1001 clases diferentes. Para adecuar esta red neuronal al modelo a crear, se elimina la última capa de la misma ya que tiene 1001 salidas, y el número de clases del modelo no es tan extenso, por lo que no necesita un entrenamiento tan exhaustivo, lo que reduce drásticamente el tiempo de entrenamiento. Además, se requiere adaptar la red neuronal de manera que su última capa tenga un número de salidas equivalentes a las clases del modelo previamente establecido.

El modelo a crear consta de 25 clases y emplea varias capas con diversas características. Algunas de las capas se presentan en la Ilustración 3-25. Del modelo se entrenan solo las capas añadidas, mientras que las capas ya entrenadas con los pesos de *MobileNet* permanecen congeladas.

Las capas más importantes del modelo creado son las siguientes:

Rescaling: Se encarga de escalar o reescalar las imágenes de entrada para normalizar los valores de los píxeles antes de enviarlas a las capas de convolución.

Conv2D (Convolución 2D): Realiza operaciones de convolución en las imágenes de entrada, detectando patrones locales mediante el deslizamiento de filtros sobre las mismas.

MaxPooling 2D (Agrupación máxima 2D): Reduce la resolución espacial de las representaciones, manteniendo las características más importantes, en otras palabras, toma el valor máximo de un grupo de píxeles.

Flatten: Transforma los datos bidimensionales a un formato unidimensional, para convertir las características extraídas en un formato adecuado.

Dense (Capa densa): Se encarga de realizar operaciones de redes neuronales (Producto punto, función de activación y propagación hacia adelante) totalmente conectadas para aprender patrones más complejos y así obtener la clasificación final.

Dropout: Se encarga de apagar aleatoriamente el 30% de las neuronas durante el entrenamiento, para prevenir el sobreentrenamiento (*overfitting*) de la red.

```

transferlearning entrenamiento en pc.py
transferlearning entrenamiento en pc.py

m_MobileNet = MobileNet(input_tensor=image_input, include_top=False, weights='imagenet')
m_MobileNet.summary()

last_layer = m_MobileNet.layers[-1].output
x= Flatten(name='flatten')(last_layer)
x = Dense(64, activation='relu', name='fc1')(x)
x=Dropout(0.3)(x)
out = Dense(num_classes, activation='softmax', name='output')(x)
custom_model = Model(image_input, out)
custom_model.summary()

# freeze all the layers except the dense layers
for layer in custom_model.layers[:4]:
    layer.trainable = False
custom_model.summary()

custom_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

```

Ilustración 3-25: Código del modelo de la red neuronal convolucional

Realizado por: Argos, J.; Granizo, I., 2023.

El siguiente paso es la compilación del modelo creado, en la Ilustración 3-26 se pueden observar todos los parámetros con los que va a trabajar cada capa de la red neuronal convolucional, obteniendo los valores más importantes del modelo, que son los parámetros totales junto con los 5 millones de parámetros que sirven para el entrenamiento.

PROBLEMAS	SALIDA	CONSOLA DE DEPURACIÓN	TERMINAL	PUERTOS
	conv_pw_12_bn (BatchNormalization)	(None, 9, 9, 1024)		4096
	conv_pw_12_relu (ReLU)	(None, 9, 9, 1024)		0
	conv_dw_13 (DepthwiseConv2D)	(None, 9, 9, 1024)		9216
	conv_dw_13_bn (BatchNormalization)	(None, 9, 9, 1024)		4096
	conv_dw_13_relu (ReLU)	(None, 9, 9, 1024)		0
	conv_pw_13 (Conv2D)	(None, 9, 9, 1024)		1048576
	conv_pw_13_bn (BatchNormalization)	(None, 9, 9, 1024)		4096
	conv_pw_13_relu (ReLU)	(None, 9, 9, 1024)		0
	flatten (Flatten)	(None, 82944)		0
	fc1 (Dense)	(None, 64)		5308480
	dropout (Dropout)	(None, 64)		0
	output (Dense)	(None, 25)		1625
=====				
Total params: 8,538,969				
Trainable params: 5,310,105				
Non-trainable params: 3,228,864				

Ilustración 3-26: Modelo compilado de la ilustración anterior

Realizado por: Argos, J.; Granizo, I., 2023.

3.7.1.5 Entrenamiento de la red neuronal convolucional

En la ilustración 3-27 se puede observar el código utilizado para entrenar la red neuronal, donde las épocas (*epochs*) representan el número de iteraciones del entrenamiento del modelo y cuyo valor varía dependiendo de cuan eficiente sea cada entrenamiento. Google Colab, junto con su gran capacidad de hardware proveniente de Google, ayuda a reducir significativamente los tiempos de entrenamiento.

```
model_history = custom_model.fit_generator(  
    train_generator,  
    epochs=epochs,  
    validation_data=validation_generator,  
    steps_per_epoch=nb_train_samples//batch_size,  
    validation_steps=nb_validation_samples//batch_size)
```

Ilustración 3-27: Ejemplo del entrenamiento del modelo.

Realizado por: Argos, J.; Granizo, I., 2023.

3.7.1.6 Evaluación del modelo

La evaluación del modelo consiste en medir el rendimiento y capacidad que tiene para generalizar nuevos datos a lo largo del tiempo, mediante gráficas que muestran el avance del entrenamiento a lo largo de las épocas. El código necesario para realizar esta operación se muestra en la Ilustración 3-28.

```
acc = history.history['accuracy']  
val_acc = history.history['val_accuracy']  
  
loss = history.history['loss']  
val_loss = history.history['val_loss']  
  
epochs_range = range(epochs)  
  
plt.figure(figsize=(8, 8))  
plt.subplot(1, 2, 1)  
plt.plot(epochs_range, acc, label='Training Accuracy')  
plt.plot(epochs_range, val_acc, label='Validation Accuracy')  
plt.legend(loc='lower right')  
plt.title('Training and Validation Accuracy')  
  
plt.subplot(1, 2, 2)  
plt.plot(epochs_range, loss, label='Training Loss')  
plt.plot(epochs_range, val_loss, label='Validation Loss')  
plt.legend(loc='upper right')  
plt.title('Training and Validation Loss')  
plt.show()
```

Ilustración 3-28: Código para la evaluación del modelo entrenado.

Realizado por: Argos, J.; Granizo, I., 2023.

La Ilustración 3-29 muestra un ejemplo de la evaluación del modelo entrenado. En la gráfica de la izquierda, se observa que el entrenamiento del modelo tiene una excelente precisión (ascendente) de acuerdo a la validación de las imágenes. La gráfica de la derecha muestra que las pérdidas que ha tenido el entrenamiento son bajas ya que tienden a cero, obteniendo una validación precisa del modelo.

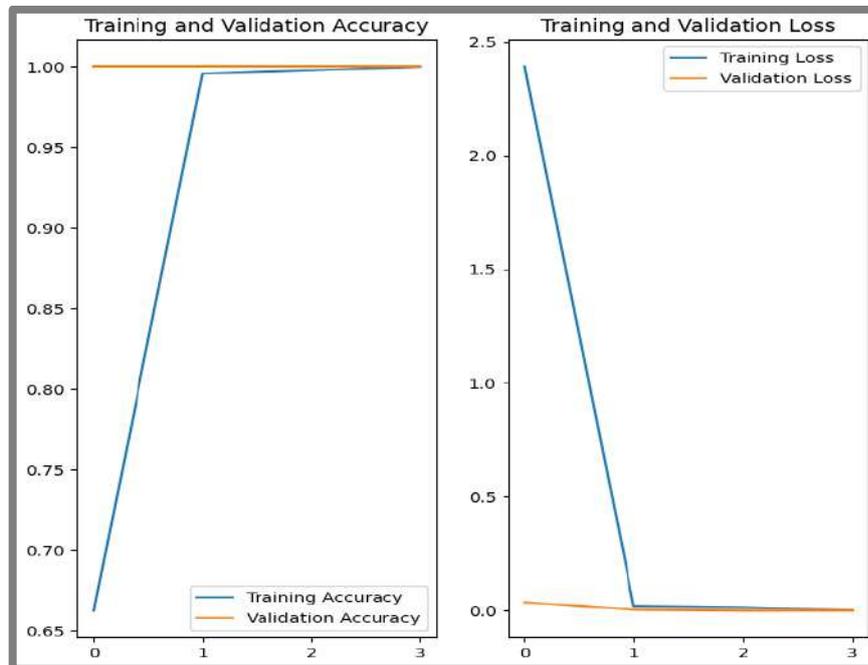


Ilustración 3-29: Ejemplo de la evolución del modelo a lo largo de las épocas. Izquierda: promedio de validación y entrenamiento. Derecha: pérdidas en el entrenamiento y validación.

Realizado por: Argos, J.; Granizo, I., 2023.

3.7.1.7 Exportación del modelo entrenado

Luego completar el entrenamiento del modelo con una evolución satisfactoria, se lleva a cabo su exportación (Ilustración 3-30). Este paso es necesario para prevenir la necesidad de repetir el entrenamiento cada vez que se realice una predicción, obteniendo una práctica más eficiente y viable en términos de gestión de recursos.

```
model.save('/content/drive/MyDrive/modelorc')
```

Ilustración 3-30: Código para la exportación del modelo ya entrenado

Realizado por: Argos, J.; Granizo, I., 2023.

Al compilar el código, se genera una carpeta en Google Drive para su posterior descarga e importación hacia la memoria del ordenador. De este modo, las predicciones se realizan de manera rápida sin la necesidad de reentrenar el modelo.

3.7.1.8 Función de importación del modelo y predicción en Python

La Ilustración 3-31 muestra el código de *Python* en *Visual Studio Code* para la importación del modelo ya entrenado, utilizando las librerías de *TensorFlow*. La función recibe la ruta de una imagen enviada por LabVIEW y realiza la predicción correspondiente.

```
ruta_descargas = os.path.expanduser('D:\Descargas\TESIS')
nombre_archivo = 'modelorc'
ruta_modelo = os.path.join(ruta_descargas, nombre_archivo)

model = tf.keras.models.load_model('D:\Descargas\TESIS\modelorc')
```

Ilustración 3-31: Código para la importación en Visual Studio Code del modelo exportado.

Realizado por: Argos, J.; Granizo, I., 2023.

El resultado de la predicción incluye información crucial como la frecuencia y la aceleración del patrón generado en la superficie libre del líquido o en el lecho de partículas, junto con un porcentaje de exactitud en la predicción. Estos valores son enviados a LabVIEW para su posterior presentación. El código para realizar este procedimiento se encuentra en la Ilustración 3-32.

```
pathlabview = imagenlabview
image = tf.keras.preprocessing.image.load_img(imagenlabview, target_size=(485, 475))
input_arr = tf.keras.preprocessing.image.img_to_array(image)
input_arr = np.expand_dims(input_arr, axis=0) # Convert single image to a batch.
predictions = model.predict(input_arr)

score = tf.nn.softmax(predictions[0])
salidalabview= "Esta imagen parece ser {} con un {:.2f} % de exactitud.".format(class_names[np.argmax(score)], 100 * np.max(score))
print(salidalabview)
return salidalabview
```

Ilustración 3-32: Código para la predicción de la imagen y retorno de valor hacia Labview

Realizado por: Argos, J.; Granizo, I., 2023.

3.7.2 Desarrollo del entorno virtual

La interfaz usuario-máquina desarrollada en LabVIEW permite realizar el llamado a la red neuronal convolucional desde un archivo de *Python* generado en la sección anterior. Además, LabVIEW contiene el código necesario para realizar la comunicación del instrumento virtual con los dispositivos hardware. La Ilustración 3-33 muestra el panel frontal del instrumento virtual.



Ilustración 3-33: Panel Frontal del instrumento virtual

Realizado por: Argos, J.; Granizo, I., 2023.

3.7.2.1 Llamado de la función en Python

Para invocar la función, es necesario abrir una sesión de Python donde se especifique la versión de uso. A continuación, se selecciona la ruta del archivo *.py*, se especifica el nombre de la función que contiene la red neuronal convolucional (en este caso, “*rnc*”) y se establece el formato de respuesta, que en este caso es de tipo *String*. Una vez que se haya obtenido la respuesta deseada, se procede a cerrar la sesión de Python.

La información que se envía a la función de Python corresponde a la imagen cargada desde el ordenador. Esta secuencia de bloques se encuentra representada a detalle en la Ilustración 3-34.

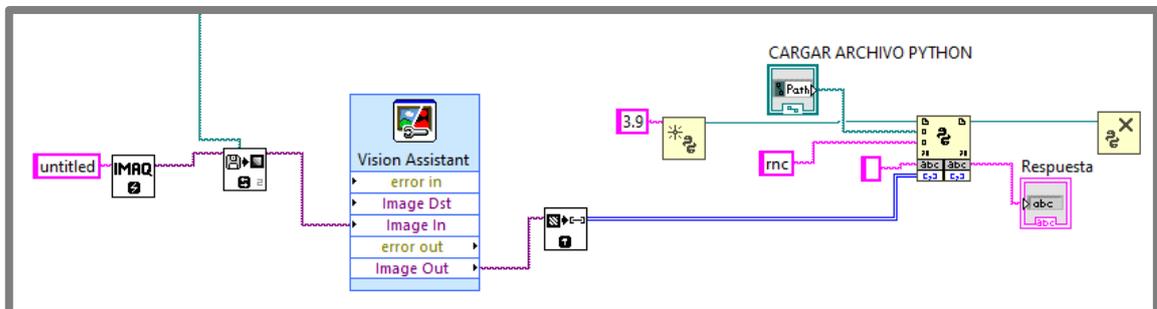


Ilustración 3-34: Código para el llamado de la función en *Python* y presentación del resultado de la clasificación.

Realizado por: Argos, J.; Granizo, I., 2023.

3.7.2.2 Comunicación con Hardware

Para la comunicación del instrumento virtual con los dispositivos hardware, en este caso las cámaras NIKON D7500 y BASLER acA640.750um, se requieren las extensiones de NI-IMAQ de LabVIEW, las cuales brindan el soporte necesario para el procesamiento de imágenes y la captura de imágenes y videos.

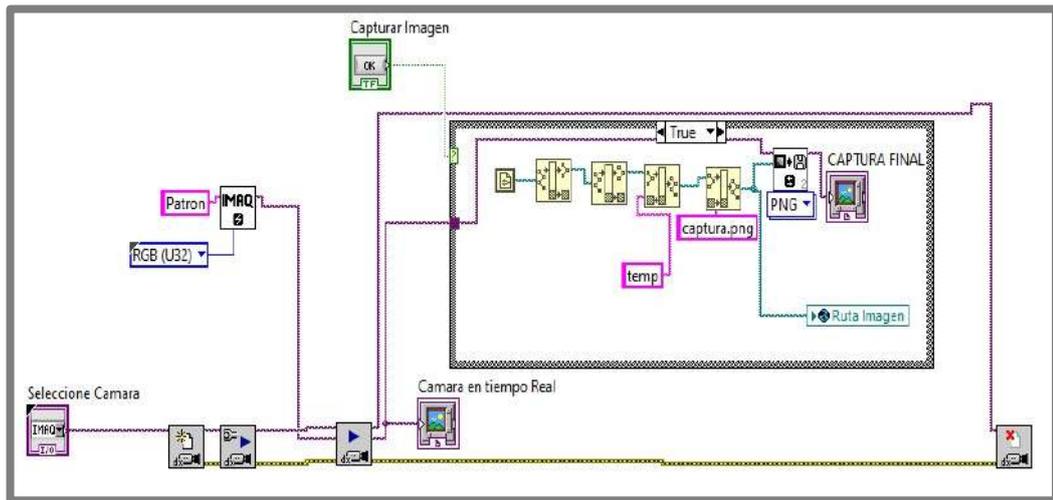


Ilustración 3-35: Código para la comunicación con los dispositivos hardware.

Realizado por: Argos, J.; Granizo, I., 2023.

En la Ilustración 3-35 se observa la configuración de bloques necesaria para la lectura, visualización en tiempo real y almacenamiento de una imagen en el ordenador. Como primer paso, se inicia una sesión, configurada para seleccionar la cámara dependiendo del número de cámaras conectadas y puertos existentes (en este caso, 2). Luego, se conecta el bloque “*IMAQX configure Grab*”, que permite iniciar la grabación de la cámara, la operación se realiza dentro de un ciclo *While Loop* para mostrar la grabación en tiempo real, y así mostrar el video en el panel frontal mediante el bloque denominado “Cámara en tiempo real”. Dentro de la estructura *Case*, al presionar el botón “capturar imagen”, se toma una foto y se almacena en el ordenador gracias al bloque “*IMAQ Write File*”, que además permite, mediante un control en el panel frontal, seleccionar la ruta de almacenamiento de la imagen y especificar el formato de la misma, mostrando la captura realizada en el panel frontal. La lectura de la cámara se finaliza al cerrar sesión fuera del ciclo *Case* con la ayuda del bloque “*IMAQ Close Camara*”.

En el anexo K se encuentra el código completo del instrumento virtual desarrollado en Python.

3.7.3 Funcionamiento del instrumento virtual mediante diagramas de flujo

La Ilustración 3-36 muestra el diagrama de flujo que representa el funcionamiento general del entorno virtual, donde luego de cargar una imagen, el instrumento identifica el tipo de estructura contenida en dicha imagen para realizar el procesamiento respectivo de acuerdo con el tipo de patrón adquirido, y visualizar los resultados del análisis a manera de valores numéricos y gráficas. Cada bloque representa una función diferente, cuyos procesos se exponen por separado.

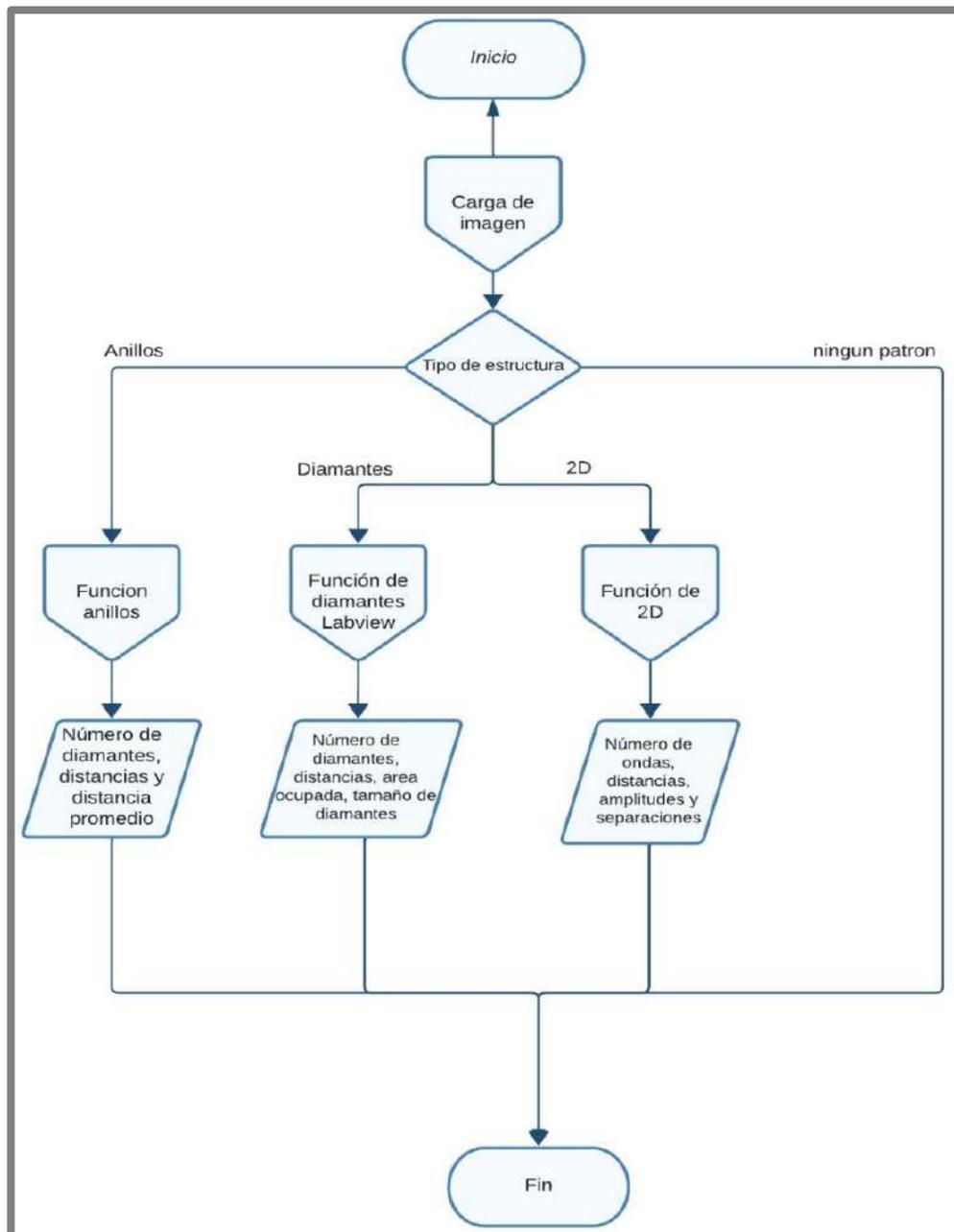


Ilustración 3-36: Diagrama de flujo del funcionamiento general.

Realizado por: Argos, J.; Granizo, I., 2023.

El desarrollo del entorno virtual consiste en realizar el llamado de varias funciones creadas en LabVIEW, tal es el caso de la función denominada “carga de imagen” que tiene un proceso exclusivo, cuyo objetivo es brindar la opción de elegir entre dos formas distintas de cargar una imagen. La primera opción consiste en importar una imagen desde el ordenador mediante un panel de selección. La segunda opción se basa en la selección de la cámara (Nikon D7500 o Basler) para capturar una imagen en tiempo real que se almacena en una carpeta temporal. La imagen cargada se envía al modelo en *Python* para su respectiva predicción y finalmente, el entorno virtual muestra una respuesta de clasificación. La Ilustración 3-37 describe este proceso.

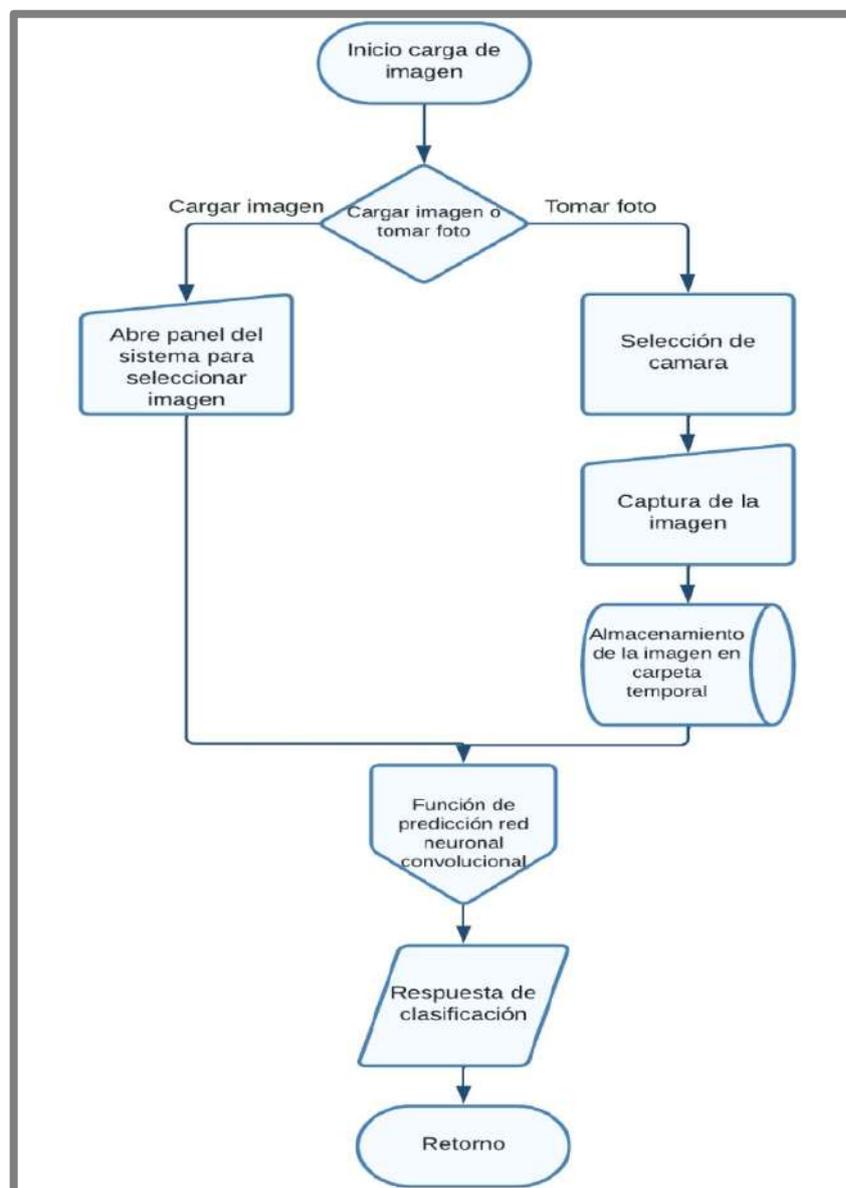


Ilustración 3-37: Diagrama de flujo de la función “carga de imagen”.

Realizado por: Argos, J.; Granizo, I., 2023.

La función llamada “predicción de red neuronal convolucional” que se muestra en la Ilustración 3-38, se encarga de recortar el área de interés de la imagen adquirida, para su posterior predicción, donde se evalúa el análisis a realizar dependiendo del tipo de celda contenedora.

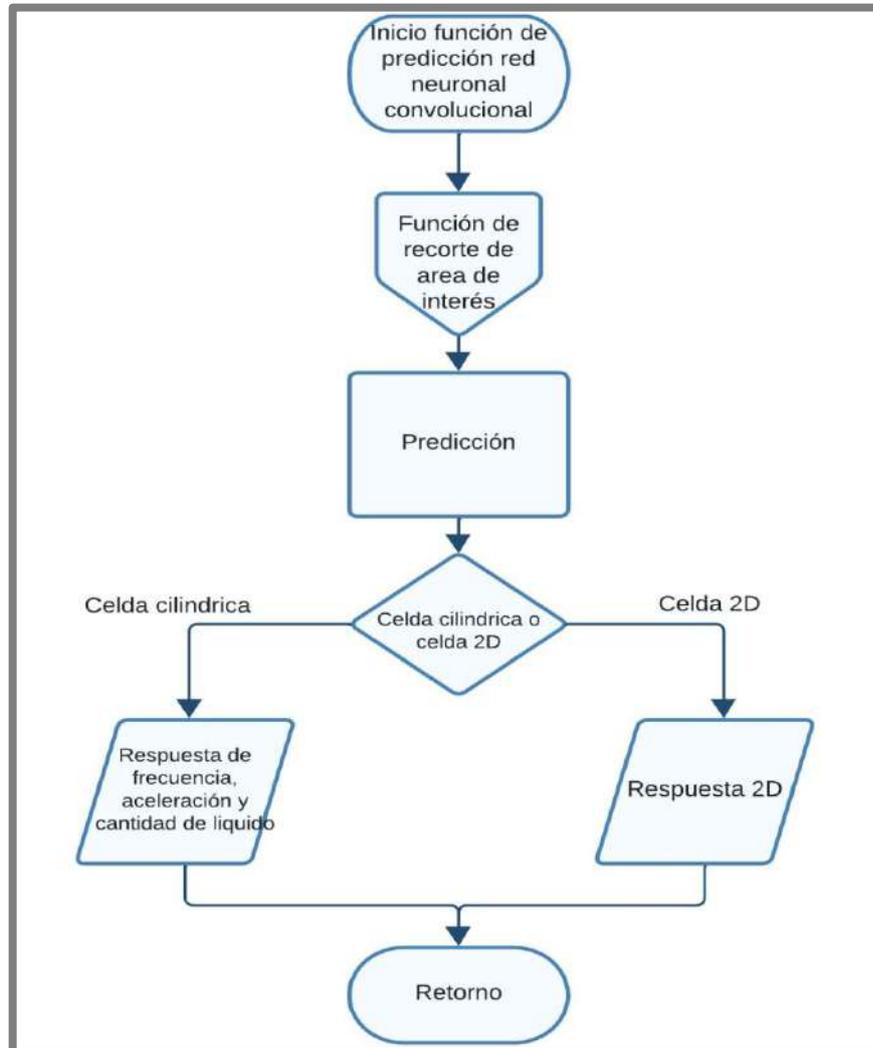


Ilustración 3-38: Diagrama de flujo de la función “predicción de red neuronal convolucional”.

Realizado por: Argos, J.; Granizo, I., 2023.

El recorte del área de interés es fundamental para realizar un análisis profundo de la imagen, ya que permite identificar el tipo de celda contenedora como punto de partida para realizar la predicción y el análisis respectivo. La función de “recorte de área de interés” realiza este proceso, con la ayuda del modelo de red neuronal convolucional denominado *YOLOv8*, popular en la detección de objetos en imágenes y videos. Este modelo de red está entrenado para clasificar y detectar los patrones contenidos en las imágenes, ya sea en la celda cilíndrica o en la celda *cuasi* 2D, donde luego de identificar el tipo de patrón, realiza una detección de formas, encerrando y recortando el área de interés. En la celda cilíndrica se recorta el borde del recipiente, mostrando

solamente el interior del mismo con el patrón respectivo, mientras que en la celda *cuasi* 2D se recorta el área que contenga una vista clara de las ondas. El diagrama de flujo representado en la Ilustración 3-39 muestra este proceso.

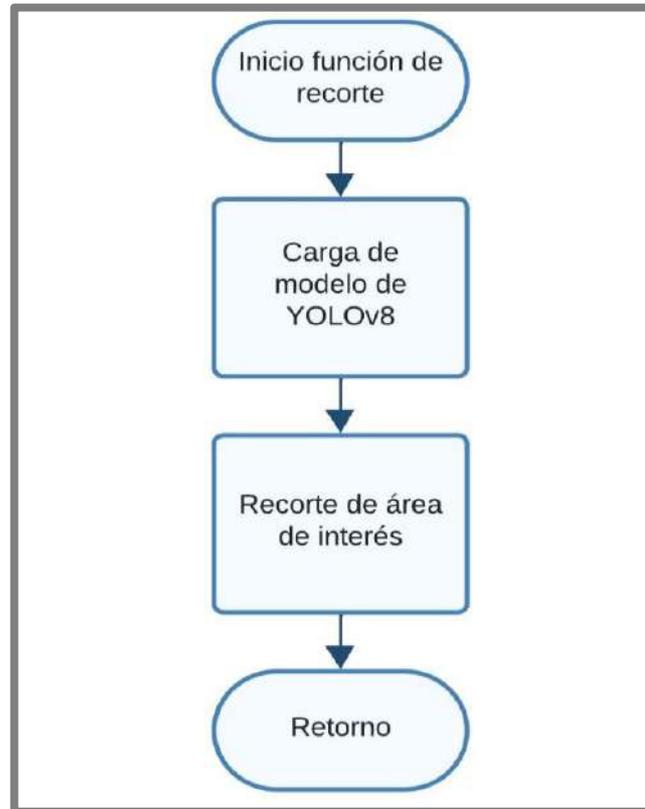


Ilustración 3-39: Diagrama de flujo de la función “recorte de área de interés”.

Realizado por: Argos, J.; Granizo, I., 2023

Una vez realizada la predicción del área de interés, se obtiene el tipo de celda contendor para realizar el análisis respectivo, en caso de tener como punto de partida la celda cilíndrica, se debe identificar el tipo de patrón contenido en la celda, tal es el caso de la función de “anillos”, que hace referencia a los anillos de Faraday y cuyo estudio consiste realizar una serie de pasos para la obtención de resultados, tal y como se observa en la Ilustración 3-40.

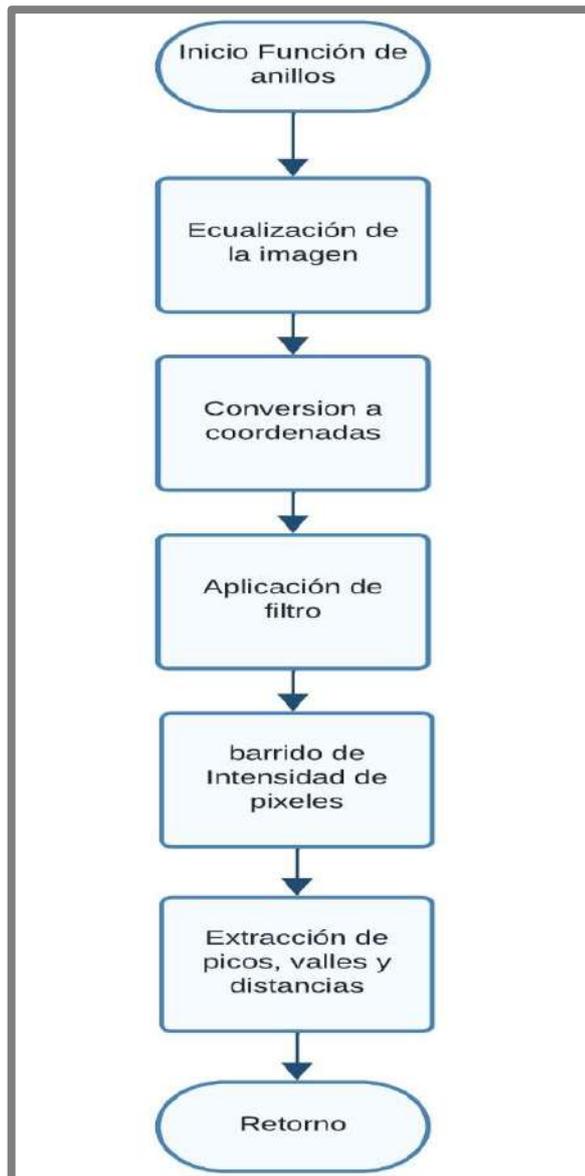


Ilustración 3-40: Diagrama de flujo de la función “anillos”.

Realizado por: Argos, J.; Granizo, I., 2023.

En el caso de tener patrones de diamantes, el procesamiento de la imagen es diferente, ya que se requiere de un análisis exclusivo para obtener resultados concretos que conlleven a una comprensión clara y concisa del comportamiento de este tipo de patrón. La función “diamantes” que se muestra en la Ilustración 3-41 hace referencia a lo establecido por dichos patrones, donde se realizan varias etapas para desembocar en los resultados requeridos, destacando también la función “diamantes Python”, cuyo proceso se lleva a cabo en la plataforma de *Python* para retornar hacia la función original de “diamantes”.

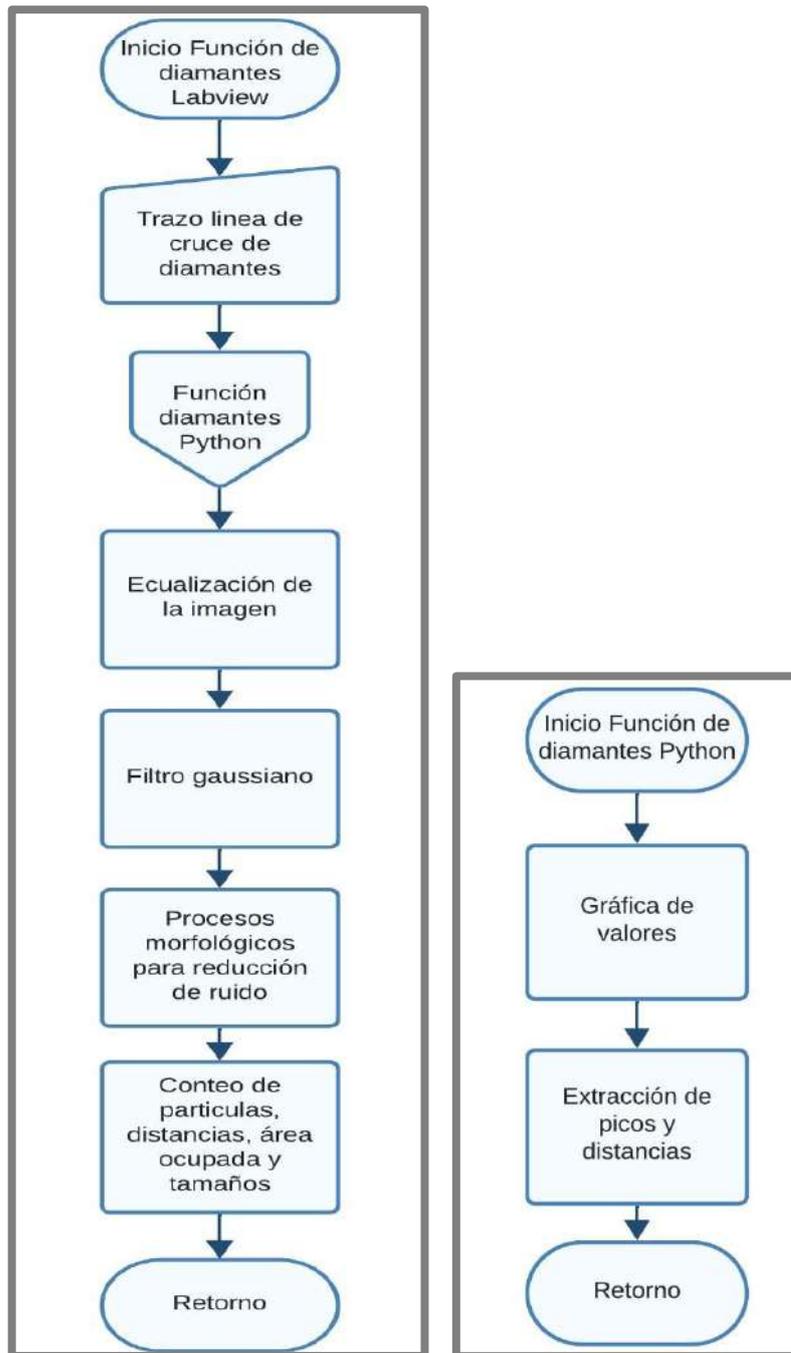


Ilustración 3-41: Izquierda: Diagrama de flujo de la función “diamantes”. Derecha: Diagrama de flujo de la función “diamantes Python”

Realizado por: Argos, J.; Granizo, I., 2023.

En caso de tener una imagen con la celda *cuasi* 2D, el proceso para su respectivo estudio se muestra en la Ilustración 3-42, donde se llevan a cabo distintas etapas de procesamiento, que sirven para visualizar resultados con la mayor claridad posible y de fácil comprensión para el usuario.

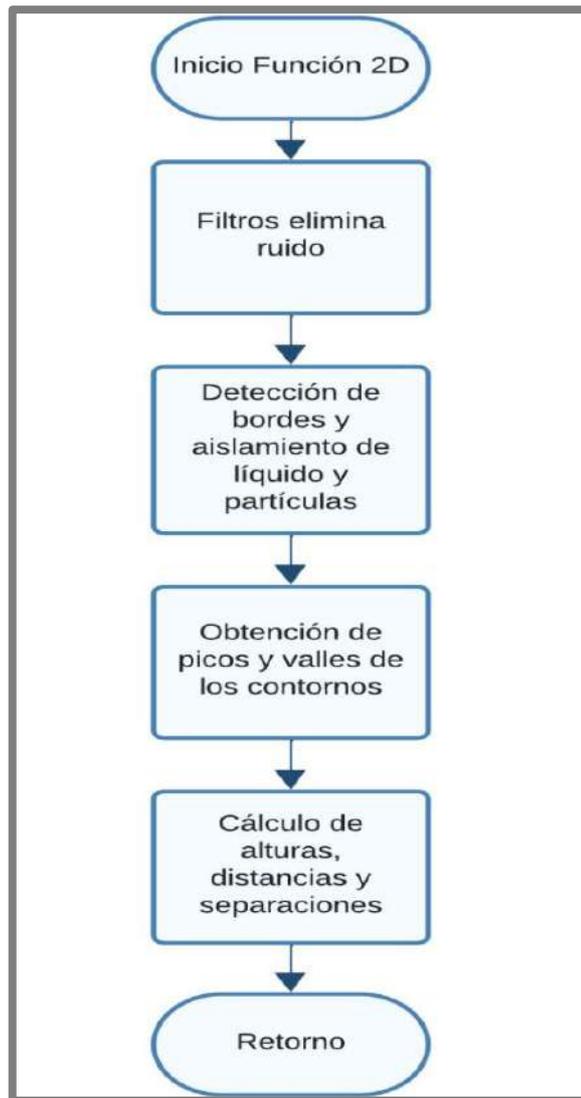


Ilustración 3-42: Diagrama de flujo de la función “2D”.

Realizado por: Argos, J.; Granizo, I., 2023.

3.7.4 Validación del instrumento virtual bajo estándares internacionales

Debido a que la mayoría de medidas cuenta con prefijos para un entendimiento más claro de los valores, uno de los principales estándares que sigue el instrumento virtual es la utilización del Sistema Internacional de Unidades creado en 1960 por la 11^{va} Conferencia General de Pesas y Medidas, ya que el instrumento virtual hace uso de distintas unidades de medida como el metro, Hertz, litros, etc., junto con sus respectivos múltiplos y submúltiplos. Otro estándar empleado es el idioma, si bien no se ha encontrado ninguna norma o estándar que determine el lenguaje a utilizar en un instrumento virtual más que el enfocado al público, según la página de estadística “*Statista*” en su publicación del 1 de junio de 2023, el idioma más comprendido a nivel mundial

es el inglés, por esta razón se ha optado por utilizar este lenguaje en el desarrollo del instrumento virtual.

ISO/IEC 25010, esta norma detalla varias características para evaluar la calidad de un software, entre las cuales destaca la funcionalidad, fiabilidad, compatibilidad y seguridad, donde los dos primeros aspectos se abordan en la sección de resultados y diseño del instrumento virtual.

- La compatibilidad, al ser el software de ejecución LabVIEW, queda a cargo de *National Instrument*, hasta su publicación actualizada al día 15 de septiembre del 2023, el instrumento virtual puede ser ejecutado en *Windows 11* y *Windows 10*, *Ubuntu 20*, *Ubuntu 18*, *openSUSE Leap 15.2*, *openSUSE Leap 15.1*, *Red Hat Enterprise Linux 8* y *Red Hat Enterprise Linux 7*.
- Dada la naturaleza del instrumento virtual, el cual opera de manera autónoma sin requerir una conexión a internet ni depender de una base de datos, se prescinde de la necesidad de implementar medidas de seguridad específicas en este contexto.

Aunque la norma **ISO/IEC 1158** no se centra explícitamente en instrumentos virtuales, aborda con profundidad las interfaces de usuario, símbolos e iconografía. En el caso del instrumento virtual, se observa una conformidad con los preceptos establecidos. Por ejemplo, se hace uso del ícono de puntero, el cual, alineado con el estándar del sistema operativo, proporciona una experiencia continuista. De la misma manera, los íconos de acciones presentan una representación tridimensional, destacando la función a ejecutar mediante la combinación de texto y figura. Por último, los íconos de objetos resaltan de manera perceptible en el entorno del software, asegurando una identificación clara y eficiente.

Las normas técnicas **ISO 9241** y **EN 29241** detallan los requisitos fundamentales para la presentación de datos en una pantalla. Dentro de los criterios empleados en el instrumento virtual, destacan la exigencia de una matriz mínima de representación de caracteres de al menos 5x7 píxeles, garantizando una legibilidad óptima. Asimismo, se subraya la importancia de un espacio entre caracteres equivalente a la anchura del trazo, promoviendo una presentación clara y definida. En adición, se estipula un contraste mínimo de 35 cd/m^2 para los caracteres, o en su defecto, una polaridad positiva del fondo y caracteres en polaridad negativa, o viceversa, asegurando una óptima visibilidad y claridad en la visualización de la información.

CAPÍTULO IV

4. PRUEBAS

En este capítulo se realizan las pruebas necesarias para la generación de estructuras granulares y la obtención de la base de datos definitiva para el desarrollo del instrumento virtual, donde se realizan diversas comparaciones entre patrones formados a diferentes valores de frecuencia f , aceleración a_g , espesor de la capa de líquido Δh y en el caso de la celda *cuasi* 2D, espesor de la capa de partículas h_p .

4.1 Equipos para la validación experimental

Para la obtención de datos confiables y verídicos en aceleración y frecuencia, se utiliza un acelerómetro ADXL326 de 3 ejes, perteneciente a la misma familia del acelerómetro utilizado en la tesis doctoral del Dr. Diego Barba en el año 2020 (Barba 2020). Para la adquisición de los datos enviados por el acelerómetro, es necesario el uso de una tarjeta de adquisición denominada *myDAQ* de *National Instruments*, cuya función es la de transformar los datos enviados desde el acelerómetro a valores fundamentales de frecuencia y aceleración.

4.1.1 Acelerómetro ADXL326

La Ilustración 4-1 muestra el acelerómetro utilizado para la toma de datos de frecuencia y aceleración obtenidos de las vibraciones. Este sensor puede medir la aceleración dinámica de la gravedad, proveniente de las vibraciones producidas por el *shaker*.

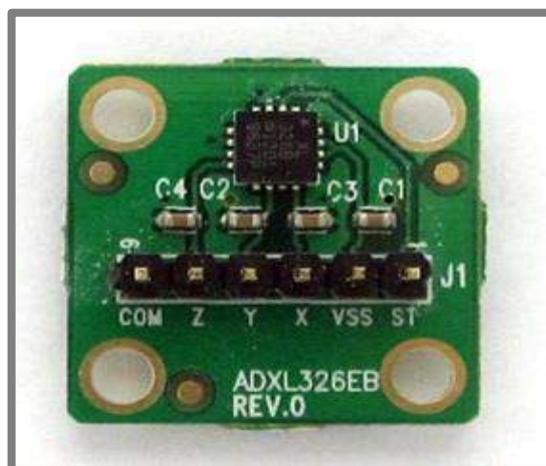


Ilustración 4-1: Acelerómetro ADXL326EB.

Fuente: (Analog Devices 2009).

La Tabla 4-1 muestra las principales características del sensor. Para más información, revise el anexo I.

Tabla 4-1: Especificaciones técnicas del acelerómetro ADXL326.

Parámetros	Especificaciones
Número de ejes	3
Rango de medición	± 16 g
Dimensión	4 mm \times 4 mm \times 1.45 mm
Baja potencia	350 μ A (valor típico)
Voltaje de alimentación	1.8 V a 3.6 V
Rango de frecuencia	0.5 Hz a 1600 Hz

Realizado por: Argos, J.; Granizo, I., 2023.

4.1.2 myDAQ

En la Ilustración 4-2 se aprecia la tarjeta de adquisición *myDAQ* de *National Instruments*, que cuenta con entradas y salidas analógicas que sirven para establecer un voltaje de salida óptimo hacia el acelerómetro ADXL326 (según el *datasheet*) y realizar la lectura de datos enviados por el mismo.



Ilustración 4-2: Tarjeta de adquisición *myDAQ*.

Fuente: (National Instruments 2012).

La Tabla 4-2 muestra las principales características de la tarjeta de adquisición. Para más información, revise el anexo J.

Tabla 4-2: Especificaciones técnicas de la tarjeta de adquisición *myDAQ*.

Parámetros	Especificaciones
Entradas analógicas	8 canales
Salidas analógicas	2 canales
Muestreo máximo	200 000 muestras por segundo
Entradas digitales	8 canales
Salidas digitales	4 canales
Alimentación	USB
Software	LabVIEW incluido

Realizado por: Argos, J.; Granizo, I., 2023.

4.1.3 Programación en LabVIEW para la lectura de aceleración y frecuencia

La tarjeta de adquisición *myDAQ* es controlada desde el software LabVIEW (ambos con el mismo fabricante, *National Instruments*), donde se encuentra la programación necesaria para mostrar los datos ya convertidos de aceleración en *g* y frecuencia en Hz.

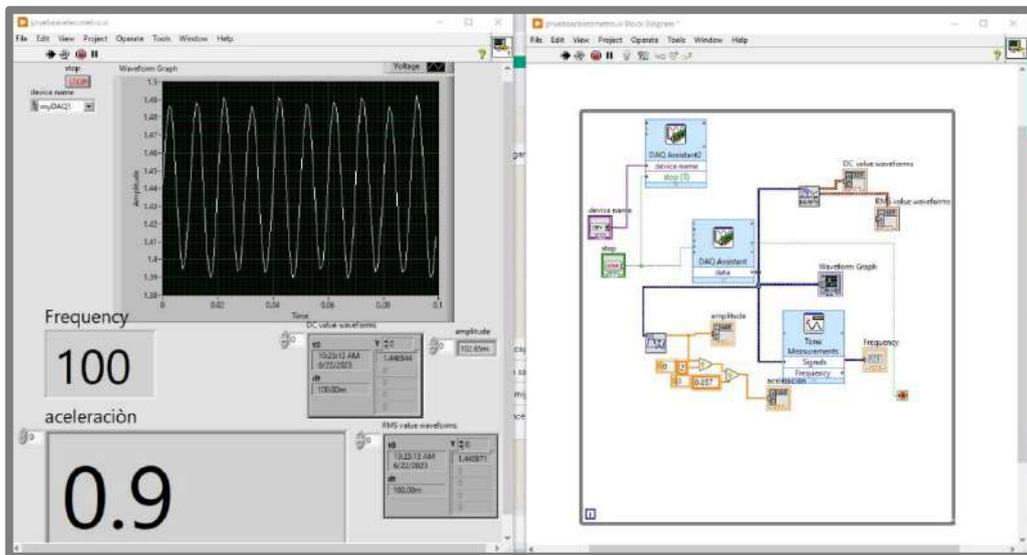


Ilustración 4-3: Programación en LabVIEW para la lectura de datos de frecuencia y aceleración, enviados desde el acelerómetro ADXL326, por medio de la tarjeta de adquisición de datos *MyDAQ*.

Realizado por: Argos, J.; Granizo, I., 2023.

La Ilustración 4-3 muestra el panel frontal y el diagrama de bloques de la interfaz gráfica que proporciona la programación necesaria para la lectura y adecuación de datos, y así obtener la visualización de los mismos en las unidades de interés. Para obtener la aceleración, el valor inicial pasa por un bloque que entrega la amplitud de la señal senoidal, luego se divide por 2 para adquirir

solo la parte negativa o positiva de la onda y obtener la aceleración en un solo eje (positivo o negativo), finalmente se realiza una conversión dividiendo el valor obtenido para el factor de conversión del acelerómetro, que es el mismo que se encuentra detallado en el *datasheet* del sensor, y es 57 mV/g .

4.2 Implementación del equipo

Para realizar las pruebas necesarias, se llevó a cabo la implementación del equipo vibratorio junto con los componentes necesarios para la toma de imágenes y la generación de estructuras granulares, tal y como se muestra en la Ilustración 4-4. Dicho equipo de encuentra ubicado en las instalaciones del laboratorio perteneciente al grupo de investigación GIICYT.

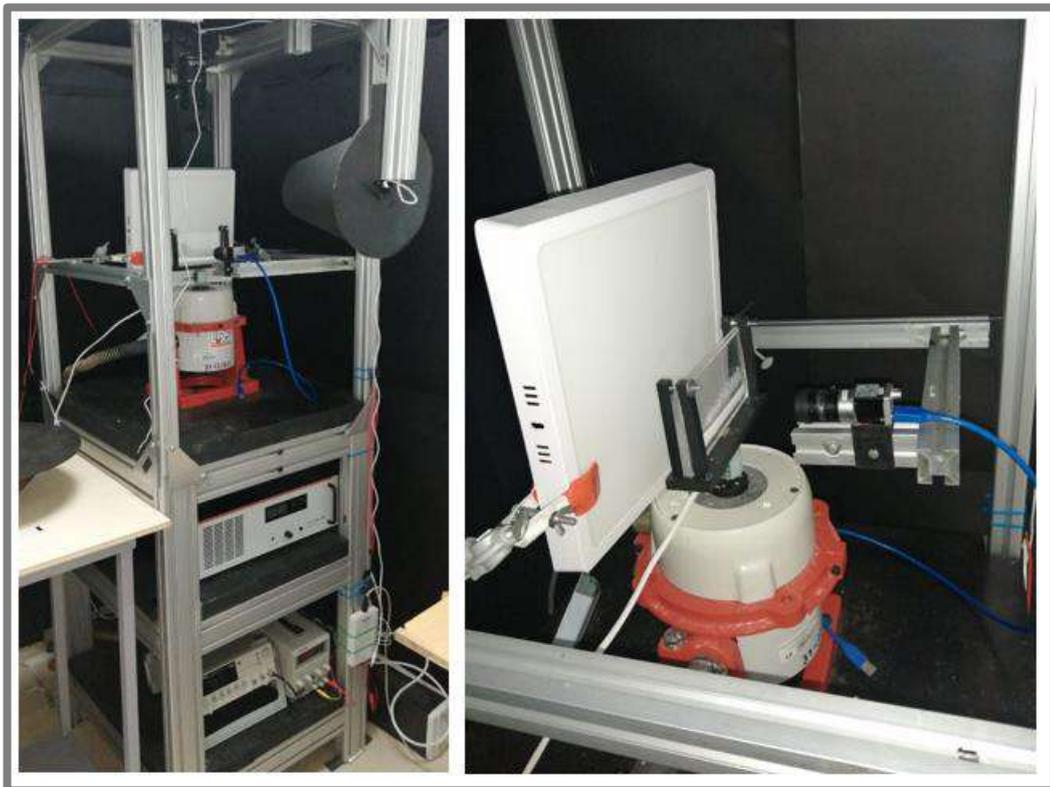


Ilustración 4-4: Izquierda: Implementación general del equipo. Derecha: Celda *cuasi* 2D removible y reemplazable por la celda cilíndrica según sea necesario.

Realizado por: Argos, J.; Granizo, I., 2023.

4.3 Análisis de patrones en la celda cilíndrica

A continuación, se muestran las pruebas realizadas con el uso de la celda cilíndrica, cuyo fin es llevar a cabo la creación de la base de datos, que sirve para el entrenamiento, validación y predicción de la red neuronal convolucional, así como para la obtención de resultados por parte

del entorno virtual. Los datos equivalen a las imágenes obtenidas a partir de los experimentos realizados en la generación de estructuras granulares en la celda cilíndrica.

4.3.1 Nivelación de la celda cilíndrica

Para la generación de patrones en la celda cilíndrica es fundamental realizar un ajuste de nivelación de la misma, ya que los efectos que un desnivel pueden producir en las partículas son negativos, obligando a las partículas desplazarse hacia un solo lado en vez de obtener una acumulación homogénea en toda base de la celda. Realizar una nivelación perfecta resulta hasta cierto punto improbable, ya que el trabajo es de tipo manual, con la única ayuda de un nivel tipo “ojo de buey” para conocer el grado de inclinación de la celda. La Ilustración 4-5 muestra el trabajo de nivelar la celda cilíndrica, donde a pesar de los esfuerzos realizados, se mantiene un pequeño desnivel, cuyos efectos generan movimientos inesperados en las partículas.

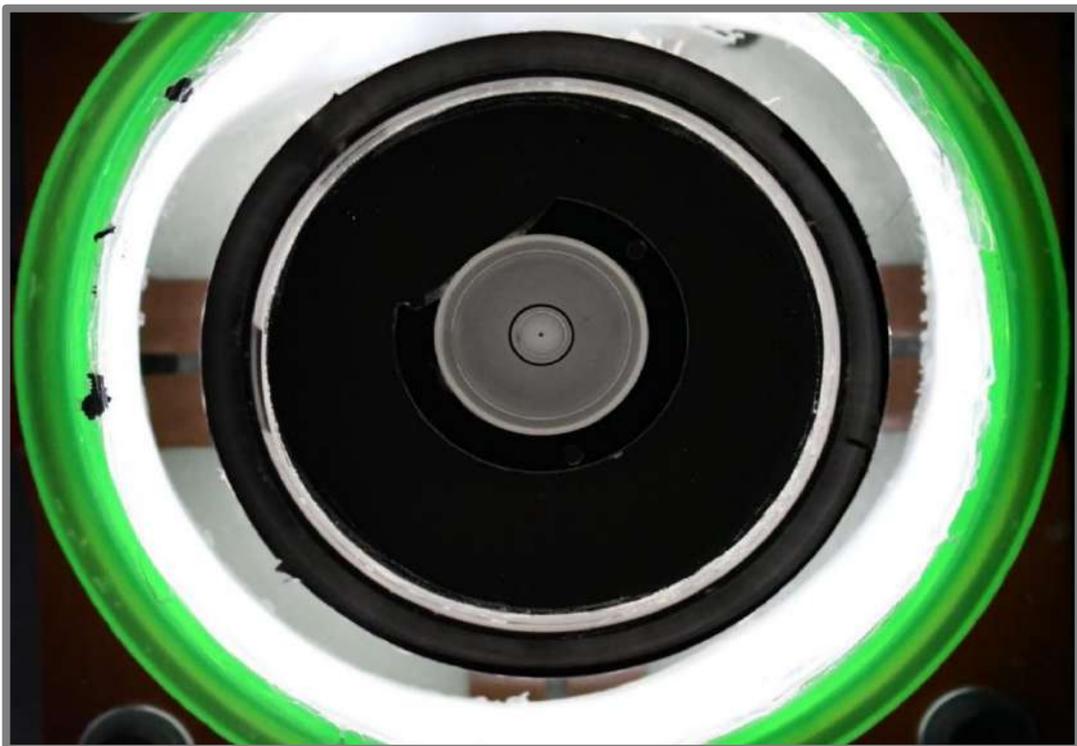


Ilustración 4-5: Nivelación de la celda cilíndrica utilizando un nivel tipo “ojo de buey”.

Realizado por: Argos, J.; Granizo, I., 2023.

4.3.2 Captura de imágenes de las estructuras granulares

Las fotografías obtenidas constituyen el pilar fundamental en el desarrollo de la base de datos, realizando varias etapas para su creación, que incluyen el preprocesamiento de las imágenes para asegurar una buena calidad, etiquetar de forma manual las imágenes en grupos distintos

dependiendo de la frecuencia y aceleración a la que pertenezcan, y permitir la participación del conjunto de datos en el entrenamiento, validación y prueba del modelo de red neuronal convolucional.

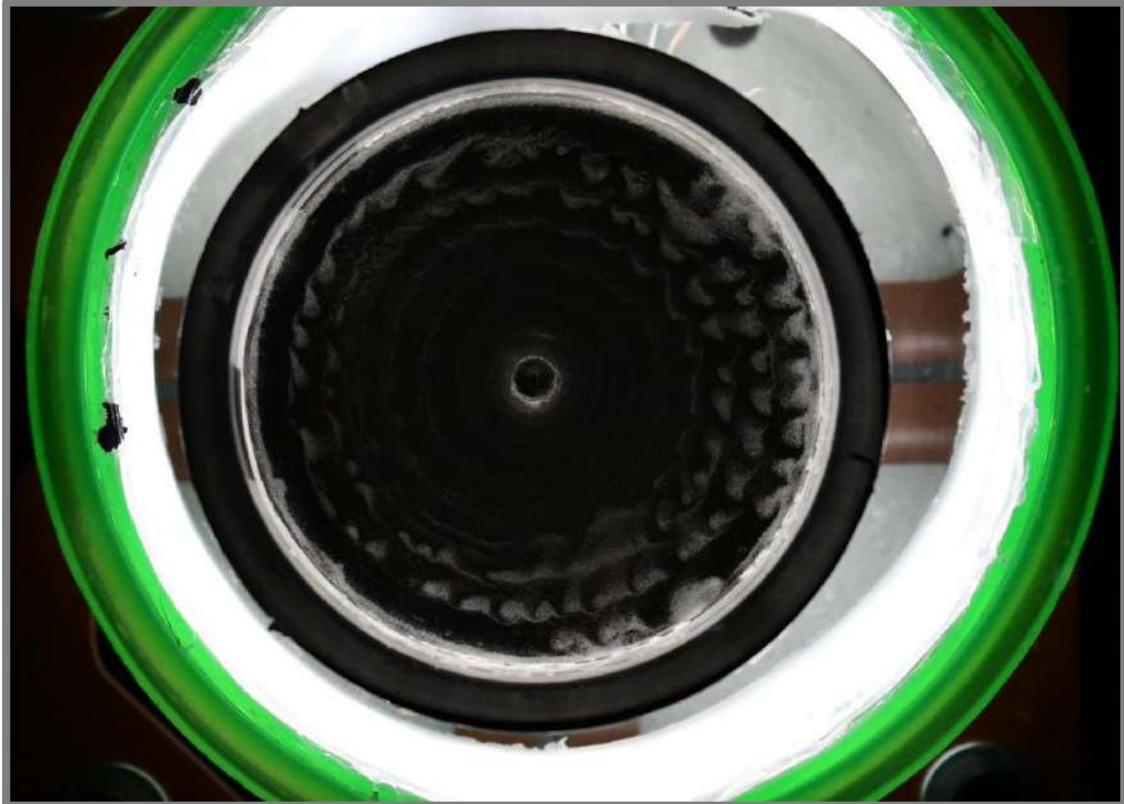


Ilustración 4-6: Ejemplo de una imagen original tomada con la cámara Nikon D7500 con una resolución de 4K, con un ángulo tipo vista de pájaro o espacial.

Realizado por: Argos, J.; Granizo, I., 2023.

La Ilustración 4-6 es un ejemplo de las imágenes “crudas” tomadas con la cámara Nikon D7500, que deben ser acondicionadas para el entrenamiento de la red, modificando su tamaño ya que originalmente las imágenes tienen una resolución de 3840×2160 píxeles, y para el entrenamiento de la red solo se necesita del espacio que contiene los patrones de partículas, siendo el resto de la imagen información inútil que se debe desechar, para evitar que el tiempo de entrenamiento se intensifique debido a la existencia de píxeles innecesarios que resultan similares en todas las imágenes, sin importar la clase a la que pertenezcan.

Para de toma de imágenes de cada patrón generado en la celda cilíndrica, tanto para la superficie libre de líquido como para el lecho de partículas, se manipula la cámara Nikon D7500 para capturar fotos durante 3 minutos, con un intervalo de tiempo entre imagen de 4 segundos.

4.3.3 Clasificación de las imágenes y comparación de los patrones obtenidos

Las imágenes deben ser clasificadas en función de la frecuencia f (Hz), aceleración a_g (g) y espesor de la capa de líquido sobrenadante Δh (ml) asociados con cada patrón generado en la superficie libre de líquido y en el lecho de partículas, con la finalidad de abastecer la red neuronal convolucional con un conjunto de datos numéricos separados por etiquetas.

Tabla 4-3: Patrones en el lecho de partículas al ser sometidas a inestabilidades hidrodinámicas con diferentes valores de frecuencia y aceleración, con un Δh constante de 5 ml.

Frecuencia f / Aceleración a_g	40 Hz	60 Hz	70 Hz	80 Hz	100 Hz
1g					
1.5g a 2g					
Inestabilidad(Chaos)					

Realizado por: Argos, J.; Granizo, I., 2023.

La Tabla 4-3 muestra un conjunto de imágenes tomadas de distintos patrones formados en el lecho de partículas al variar la frecuencia y aceleración, con un espesor de la capa de líquido Δh de 5 ml. La descripción de cada conjunto de patrones se muestra a continuación:

- **Aceleración de 1g:** se visualiza la formación de anillos de Faraday, a medida que se aumenta el valor de la frecuencia, los anillos reducen su número empezando desde el centro del recipiente hacia el borde del mismo.
- **Aceleración de 1.5g a 2g:** se forman los patrones bautizados como diamantes, con excepción del patrón formado a 60 Hz, que presenta una aglomeración irregular de partículas. A medida que aumenta la frecuencia, la cantidad de diamantes se incrementa ya que la longitud de onda es menor y, por ende, existe menos distancia entre cada aglomeración de partículas.

- **Inestabilidad (Aceleración mayor a 2.5g):** no existe la formación de un patrón estable en ninguna frecuencia ya que las partículas no se aglomeran adecuadamente, este suceso es conocido como *chaos*.

4.3.3.1 Comparación a distintos espesores de la capa de líquido Δh

Para obtener un estudio global del comportamiento de las estructuras granulares, se lleva a cabo una comparación entre distintos niveles de espesor de líquido, con una misma aceleración a distintas frecuencias. La Tabla 4-4 muestra la comparación realizada entre 5,6 y 7 ml de espesor del líquido portador Δh , con una aceleración constante de 1g.

Tabla 4-4: Comparación de los anillos de Faraday entre distintos espesores de la capa de líquido, sometidos a distintas frecuencias con una aceleración constante de 1g.

f_{-a_g} Δh	40Hz 1g	60Hz 1g	70Hz 1g	80Hz 1g	100Hz 1g
5ml					
6ml					
7ml					

Realizado por: Argos, J.; Granizo, I., 2023.

Los patrones formados en el lecho de partículas son una representación cercana a los anillos de Faraday, con varias diferencias entre cada nivel de espesor de líquido, que se detallan a continuación:

40Hz_1g: Los anillos formados con 5 ml de Δh presentan una dispersión de las partículas mucho menor en comparación a los patrones formados con 6 y 7 ml. Aunque con 7ml de Δh se presenta una mayor cantidad de anillos en el lecho de partículas, son menos compactos y las partículas se encuentran más dispersas en los bordes del recipiente, adquiriendo una forma distinta a los anillos

de Faraday. Esta característica también ocurre con 5 y 6 ml de Δh , aunque con menor intensidad y manteniendo una forma de anillos dispersos en los bordes del recipiente.

60Hz_1g: El lecho de partículas adquiere una forma más cercana a los anillos de Faraday, con mayor presencia en los extremos del recipiente, manteniendo cierta distorsión en los espesores de la capa de líquido de 5 y 6 ml.

70Hz-80Hz_1g: Los cambios no son tan significativos ya que los anillos generados son más compactos en el extremo del recipiente (sin mayor presencia en el centro), siguiendo un orden establecido por los anillos formados en la superficie libre de líquido.

100Hz_1g: Los patrones presentan una mayor aglomeración de partículas en los bordes del recipiente, con pocas o nulas diferencias entre los distintos niveles de Δh . En el centro del recipiente, los patrones de anillos son casi nulos.

4.3.4 Momentos invariantes de Zernike y creación de la base de datos

En la Tabla 4-4 se aprecia que, al aumentar la frecuencia, los patrones de anillos muestran una progresiva disminución en su apariencia, esto es bastante notable al pasar de 40 a 60 Hz y de 60 a 70 Hz. De 70 a 100 Hz, los patrones son muy similares sin importar el nivel de espesor de la capa de líquido Δh , esto representa una gran dificultad para el procesamiento de imágenes, ya que no existen características suficientes para diferenciar los parámetros de frecuencia y espesor de líquido, y el procesamiento no funcionaría de manera óptima, presentando una gran confusión al no poder diferenciar los patrones obtenidos con distintos valores de frecuencia, aceleración y espesor de la capa de líquido.

Para encontrar diferencias más concretas entre las imágenes, se utilizan los momentos invariantes de Zernike, que describen la distribución espacial de una imagen en función de sus coeficientes de momentos. Cada coeficiente de los momentos de Zernike captura información específica sobre la forma, simetría y contenido de una imagen, siendo invariantes a ciertas transformaciones geométricas como rotaciones, escalas y traslaciones. Esto implica que los momentos de Zernike, pueden capturar características distintivas de una imagen, independientemente de su orientación o tamaño.

En la Ilustración 4-7 se observan los momentos invariantes de Zernike de la última columna de la Tabla 4-4, con el fin de encontrar diferencias significativas en las imágenes.

```
[3.18309886e-01 4.75012329e-04 2.30226392e-03 9.74937729e-08
9.50627629e-04 3.29307265e-04 3.79882135e-03 3.80661086e-07
3.00201228e-03 1.42743375e-03 4.93010239e-04 3.69440993e-04
5.23486725e-03 9.91937124e-07 4.06315710e-03 1.69266980e-07
1.90598669e-03 6.55633380e-04 4.94344018e-04 3.26422549e-04
6.58020223e-03 2.06368931e-06 4.98032640e-03 3.54482787e-07
5.41403880e-03]
[3.18309886e-01 4.78469341e-04 2.30224838e-03 9.00279832e-08
9.57491500e-04 3.29307267e-04 3.79874365e-03 3.53900014e-07
3.00201643e-03 1.43760543e-03 4.93055448e-04 3.69332085e-04
5.23464957e-03 9.24835407e-07 4.06315593e-03 1.55739800e-07
1.91932086e-03 6.55774699e-04 4.94002186e-04 3.26422557e-04
6.57973543e-03 1.92660524e-06 4.98031274e-03 3.39279834e-07
5.41402971e-03]
[3.18309886e-01 5.02706406e-04 2.30216141e-03 4.61997524e-08
1.00568936e-03 3.79000419e-04 3.79830865e-03 1.74686367e-07
3.00211702e-03 1.50921844e-03 5.68670026e-04 3.78170057e-04
5.23343103e-03 4.49963784e-07 4.06337263e-03 7.95889775e-08
2.01354852e-03 7.58533373e-04 5.04086850e-04 5.67797855e-04
6.57712279e-03 9.30869160e-07 4.98072063e-03 8.53157437e-08
5.41417990e-03]
```

Ilustración 4-7: Los primeros 25 momentos invariantes de Zernike calculados para los patrones generados a 100Hz_1g con 5, 6 y 7 ml de espesores de la capa de líquido.

Realizado por: Argos, J.; Granizo, I., 2023.

Resulta evidente que los polinomios presentan una notable similitud entre sí. Las pequeñas variaciones observadas en los polinomios dependen de la forma que tenga cada patrón, o incluso de acumulaciones de partículas adicionales, que en diferentes experimentos pueden influir en los momentos de Zernike, aunque de manera tan insignificante cuando se altera el nivel de espesor de la capa de líquido. Por esta razón, los patrones que son muy parecidos (con pocas o nulas variaciones) se agrupan en la misma clase, independientemente de las diferencias en el nivel de espesor de líquido.

De igual manera sucede con los patrones denominados diamantes de la Tabla 4-5, donde los patrones son prácticamente idénticos en 100 Hz a 1.85, 1.75 y 1.55 g respectivamente, sin importar el cambio de espesor de la capa de líquido, por lo que dichos patrones se agrupan en una sola clase.

Tabla 4-5: Comparación de patrones generados entre distintos espesores de la capa de líquido Δh , sometidos a distintos valores de frecuencia y aceleración.

$f \cdot a_g$ \ Δh	40Hz 1.5g ; 1.2g ; 1.2g	60Hz 1.5g ; 1.6g ; 1.3g	70Hz 1.6g ; 1.7g ; 1.5g	80Hz 1.65g ; 1.4g ; 1.3g	100Hz 1.85g ; 1.75g ; 1.55g
5ml					
6ml					
7ml					

Realizado por: Argos, J.; Granizo, I., 2023.

Para clasificar patrones inestables (*chaos*), se opta por agruparlos en una sola clase, sin importar el nivel de espesor de la capa de líquido, debido a que para cada frecuencia los patrones no muestran diferencias significativas, tal y como se observa en la Tabla 4-6.

Tabla 4-6: Comparación de patrones inestables generados entre distintos espesores de la capa de líquido Δh , sometidos a distintos valores de frecuencia y aceleración.

$f \cdot a_g$ \ Δh	40Hz > 2g	60Hz > 2g	70Hz > 2g	80Hz > 2g	100Hz > 2g
5ml					
6ml					
7ml					

Realizado por: Argos, J.; Granizo, I., 2023.

La Ilustración 4-8 muestra las 25 clases formadas para el conjunto de datos obtenidos de la celda cilíndrica, agrupadas en distintos valores de frecuencia, aceleración y espesor de la capa de líquido. La base de datos se utiliza para el entrenamiento, validación y predicción de la red neuronal convolucional, así como para la obtención de resultados por parte del entorno virtual.



Ilustración 4-8: Base de datos creada para el estudio de patrones formados en la celda cilíndrica.

Realizado por: Argos, J.; Granizo, I., 2023.

4.4 Análisis de patrones en la celda *cuasi* 2D

En el marco de todas las pruebas llevadas a cabo en la celda *cuasi* 2D, se procedió a la captura de imágenes con una frecuencia de disparo de 350 ms (una fotografía cada 350 milisegundos) durante un período de 2 minutos. Para este propósito, se utilizó la cámara de alta velocidad BASLER acA640.750um.

4.4.1 Formación y comparación de ondas de Faraday

El espesor de la capa de líquido sobrenadante Δh varía en mm, solamente para el análisis con la celda *cuasi* 2D, y dicha variación se realiza de forma manual. Además, el espesor de la capa de partículas h_p , se mantiene siempre constante con un valor de 10 mm.

En lo que concierne a las pruebas efectuadas en la celda *cuasi* 2D, se implementaron múltiples niveles de líquido, variando desde 1 mm hasta 10 mm y 20 mm. Cabe resaltar que, para los resultados presentados, se procedió a procesar únicamente las capas de líquido de 10, 9 y 8 mm de altura. Esta selección se basó en el entendimiento de que una mayor profundidad de líquido no

incide de manera significativa en el lecho de partículas. Por otro lado, las alturas de 9 y 8 mm se revelaron como las más idóneas para el estudio del lecho de partículas, dado que, a estas alturas, el líquido empieza a ejercer una influencia discernible sin afectar de manera drástica la formación de ondas en la superficie. Adicionalmente, se optó por utilizar imágenes en las que la onda había alcanzado su punto máximo, es decir, en mitad de un semiciclo.

La Ilustración 4-9 muestra la generación de las ondas de Faraday en un medio ciclo, se observa como las ondas en la superficie libre de líquido no afectan demasiado a las partículas, ya que dichas partículas se elevan una altura casi despreciable. A medida que el nivel de espesor de la capa de líquido disminuye, mayor es la aceleración necesaria para que se formen las ondas de Faraday, ya que la cantidad de líquido no es lo suficiente densa para ejercer una formación adecuada.

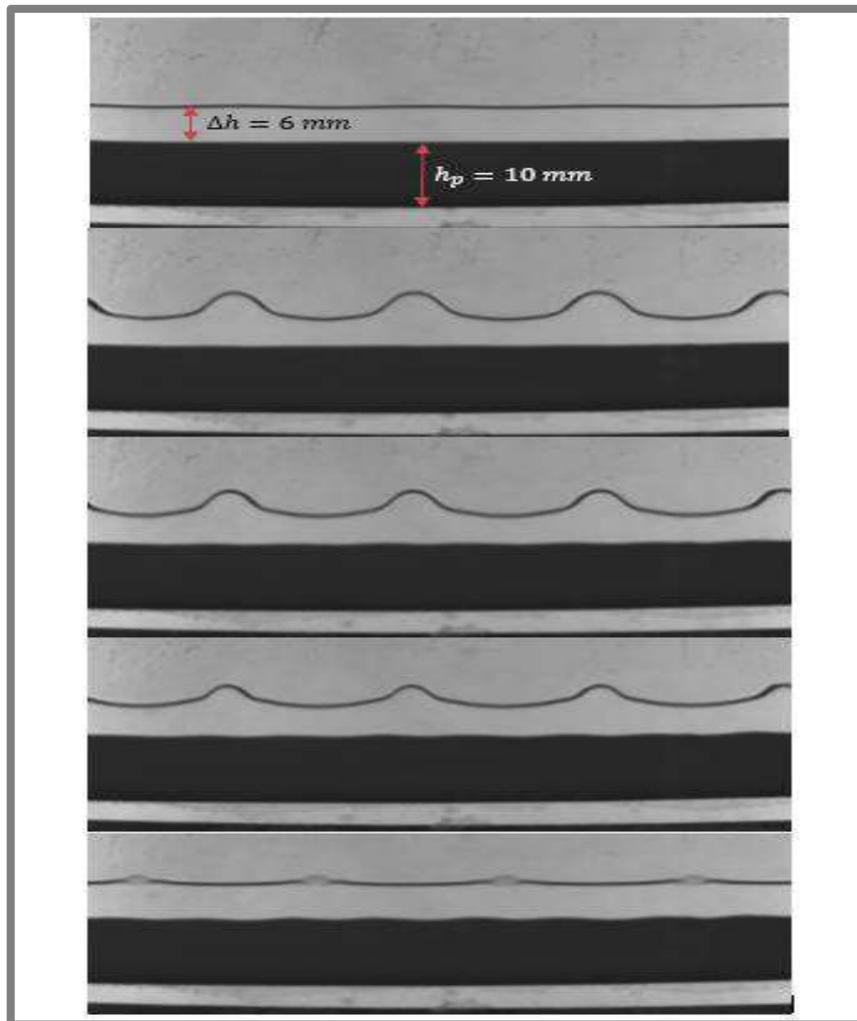
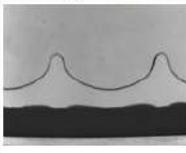
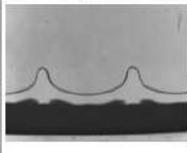
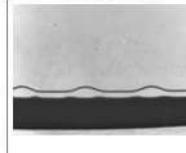
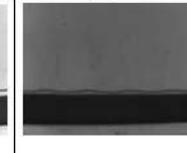


Ilustración 4-9: Patrones de las ondas de Faraday en un medio ciclo, a 24 Hz con 1.25 g, con un Δh de 6 mm y un h_p de 10 mm.

Realizado por: Argos, J.; Granizo, I., 2023.

La Tabla 4-7 muestra el umbral de aparición de las ondas de Faraday a distintos niveles de espesor de la capa de líquido Δh con una frecuencia de 14 Hz para todos los casos, donde a medida que se reduce el nivel, se necesita de una aceleración mayor para la aparición de dichas ondas. Las partículas muestran una mejor alteración a 10 y 5 ml, presentando aglomeraciones homogéneas, mientras que a 1 ml se necesita de 2 g para generar patrones en la superficie libre de líquido, con un efecto menor sobre las partículas.

Tabla 4-7: Comparación del umbral de aparición de las ondas de Faraday a distintos niveles de espesor de la capa de líquido Δh .

$\alpha_g; \Delta h$					
f	1.1 g; 20 ml	1.1 g; 10 ml	1.1 g; 5 ml	1.25 g; 3 ml	2 g; 1 ml
14 Hz					

Realizado por: Argos, J.; Granizo, I., 2023.

4.4.2 Inestabilidad de Faraday

Mediante la realización de diferentes pruebas se obtuvo la inestabilidad de Faraday, donde la superficie libre de líquido desarrolla patrones inestables sin una formación en específico, en este caso el lecho de partículas se ve afectado de manera irregular, presentando alteraciones en varios sectores. La Ilustración 4-10 ejemplifica este fenómeno.

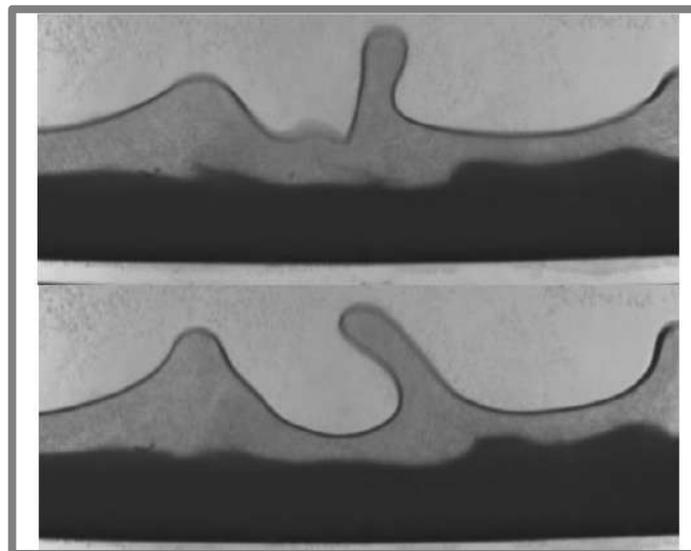


Ilustración 4-10: Inestabilidad de Faraday con un Δh de 8 ml, a 14 Hz y 1.4 g.

Realizado por: Argos, J.; Granizo, I., 2023.

4.5 Precisión del instrumento virtual

Con el propósito de evaluar la precisión del instrumento virtual, se ha desarrollado una prueba que consiste en procesar diez veces una misma imagen que contenga un patrón en la celda cilíndrica, para de esta manera poder determinar valores como la desviación estándar y el coeficiente de variación. Para asegurar unos resultados confiables, en cada prueba se ha cerrado por completo el instrumento virtual y se lo ha vuelto a ejecutar, dando los resultados presentes en la Tabla 4-8. Es importante destacar que la imagen empleada en estas pruebas corresponde al patrón de 40 Hz, con 1 g, en 6 ml, proporcionando así una muestra representativa para este análisis de precisión.

Tabla 4-8: Análisis de precisión del instrumento virtual con pruebas repetidas.

Número de prueba	Distancia promedio de anillos d_{pa}	Número de anillos N_a	Distancias entre anillos (6 primeros valores) (píxeles)					
1	21,1	11	19	19	14	20	12	32
2	21,1	11	19	19	14	20	12	32
3	21,1	11	19	19	14	20	12	32
4	21,1	11	19	19	14	20	12	32
5	21,1	11	19	19	14	20	12	32
6	21,1	11	19	19	14	20	12	32
7	21,1	11	19	19	14	20	12	32
8	21,1	11	19	19	14	20	12	32
9	21,1	11	19	19	14	20	12	32
10	21,1	11	19	19	14	20	12	32

Realizado por: Argos, J.; Granizo, I., 2023.

Los resultados obtenidos revelan una desviación estándar y un coeficiente de variación de 0 para todos los casos, lo cual corrobora una precisión excepcional en el funcionamiento del instrumento virtual. Esta elevada precisión se atribuye a su enfoque basado en procesamiento de imágenes, apoyado por modelos y funciones matemáticas, como se ha destacado anteriormente. Es esta característica la que garantiza que, para una misma entrada, el instrumento virtual produzca de manera consistente resultados idénticos, lo que refuerza su confiabilidad y exactitud.

4.6 Obtención de datos de las estructuras granulares

Con el fin de recabar información relativa a las estructuras granulares, se procedió a someter todas las imágenes correspondientes a la creación de la base de datos de la red neuronal, al procesamiento a través del instrumento virtual. Esto implicó realizar manualmente el ajuste espacial para convertir de píxeles a milímetros y, específicamente para las estructuras de diamantes, se llevó a cabo la trazabilidad manual de líneas rectas con el propósito de determinar las distancias entre dichos diamantes. Estas pruebas han proporcionado los siguientes resultados.

Tabla 4-9: Longitud de onda (λ), en píxeles, de anillos con estructuras de 5 ml a 1 g con distancias filtradas.

40 Hz	60 Hz	70 Hz	80 Hz	100Hz
34	24	22	21	19
30	23	22	20	18
32	24	22	21	19
31	23	23	20	18
32	23	22	20	18
29	20	23	20	18
		23	20	17
			21	19
			20	18
			20	20

Realizado por: Argos, J.; Granizo, I., 2023.

Tabla 4-10: Frecuencia f vs número de anillos N_a a distintos espesores de líquido Δh , en 5 ml y 6 ml a 1 g y en 7 ml a 0.9 g.

	Frecuencias	5ml	6ml	7ml
Número de anillos	40	9	9	9
	60	10	10	10
	70	10	11	11
	80	12	12	12
	100	13	12	13

Realizado por: Argos, J.; Granizo, I., 2023.

Tabla 4-11: Análisis de las estructuras de diamantes a varias frecuencias con 5ml de espesor de líquido.

Frecuencia (Hz)	Número de diamantes N_d	Porcentaje de área total (%)	Tamaño promedio (mm^2)	Distancia promedio (mm)
40	84	29,0453	24,8766	1,9232
60	111	26,237	25,3249	1,7669
70	135	26,8341	25,2619	1,66667
80	138	25,0598	19,8058	1,5416
100	184	26,1541	15,6182	1,52357

Realizado por: Argos, J.; Granizo, I., 2023.

Tabla 4-12: Análisis de las estructuras de diamantes a varias frecuencias con 6ml de espesor de líquido.

Frecuencia (Hz)	Número de diamantes N_d	Porcentaje de área total (%)	Tamaño promedio (mm^2)	Distancia promedio (mm)
40	84	33,747	45,2878	1,69529
60	102	33,2672	49,4237	1,79289
70	138	29,7008	29,8142	1,6999
80	141	28,38	25,556	1,56635
100	158	28,02	14,3707	1,5437

Realizado por: Argos, J.; Granizo, I., 2023.

Tabla 4-13: Análisis de las estructuras de diamantes a varias frecuencias con 7ml de espesor de líquido.

Frecuencia (Hz)	Número de diamantes N_d	Porcentaje de área total (%)	Tamaño promedio (mm^2)	Distancia promedio (mm)
40	89	26,7823	34,2049	1,53981
60	119	27,612	27,9241	1,66843
70	126	33,8675	32,8352	1,45355
80	135	29,6071	24,7199	1,78651
100	163	28,1492	24,4696	1,26148

Realizado por: Argos, J.; Granizo, I., 2023

CAPÍTULO V

5. RESULTADOS

5.1 Evaluación de pérdidas en el entrenamiento de la red neuronal convolucional

La red fue entrenada con 25 clases diferentes, en la Ilustración 5-1 se puede apreciar la evolución de la red neuronal convolucional durante 70 épocas (*epochs*), teniendo pérdidas de entrenamiento y validación cercanas a cero.

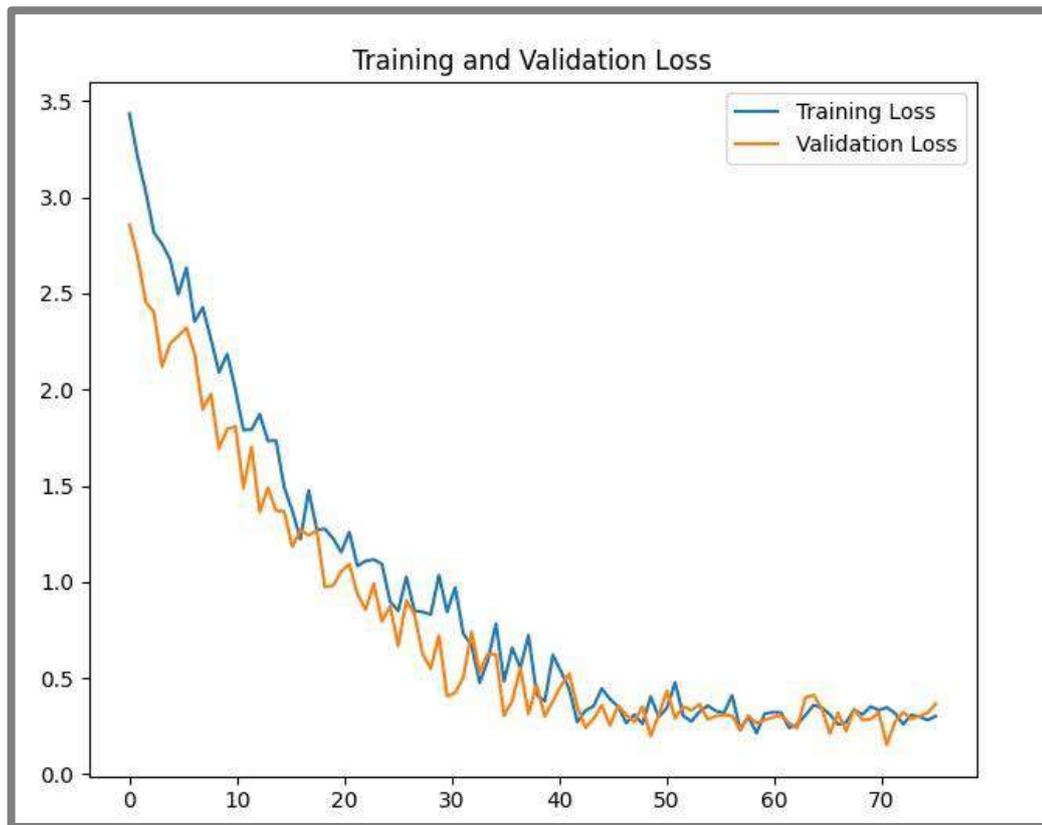


Ilustración 5-1: Pérdidas de entrenamiento y validación de la red neuronal convolucional.

Realizado por: Argos, J.; Granizo, I., 2023.

Las pérdidas presentan una trayectoria descendente, acercando su valor a 0, lo que es de gran beneficio para saber la cantidad de información perdida o desechada en el entrenamiento de la red. El modelo fue entrenado de manera correcta y eficiente, presentando márgenes de error o pérdidas en un rango aceptable. En este caso, los porcentajes de exactitud y pérdida dependen mucho de la cantidad y del tipo de datos de entrenamiento.

La Tabla 5-1 muestra la comparación entre los resultados de pérdida y exactitud, donde la exactitud con datos de entrenamiento es del 100% y con datos de validación es del 97.22%. Así mismo, la pérdida, tanto para datos de entrenamiento como para datos de validación es del 0.3%.

Tabla 5-1: Resumen de los resultados obtenidos de la red neuronal durante 70 épocas.

	Entrenamiento	Validación
Exactitud	100%	97.22%
Pérdida	0.30%	0.30%

Realizado por: Argos, J.; Granizo, I., 2023.

5.2 Exactitud de la red neuronal convolucional

La evaluación de la precisión del sistema implicó la realización de 150 iteraciones para predecir estructuras granulares seleccionadas al azar. Los resultados obtenidos demostraron un índice de aciertos cercano al 70%. Esta cifra coincide notablemente con el margen de error señalado en la Ilustración 5-1, el cual indica un 30% de variabilidad en las predicciones. Es importante destacar que las predicciones incorrectas se asociaron consistentemente con una misma estructura granular, lo que sugiere una posible limitación en la capacidad de clasificación de la red neuronal convolucional en dicho contexto.

5.3 Detección de patrones mediante el instrumento virtual

Con el fin de mejorar los resultados obtenidos al momento de realizar la clasificación de los patrones, se ha realizado el desarrollo de un algoritmo que permite la detección y la localización espacial de los patrones, además se ha añadido un recorte automático para eliminar las secciones de desinterés al momento de la clasificación, consideradas como ruido.

En la Ilustración 5-2 se puede apreciar la evolución durante el entrenamiento del algoritmo para la detección de objetos, donde las gráficas representan los porcentajes de precisión y pérdidas que existen, tanto para los datos de entrenamiento como para los datos de validación, siendo más importantes los valores relacionados a los datos de validación, teniendo como resultado que la precisión tiende al 100% mientras que la pérdida ronda el 20%.

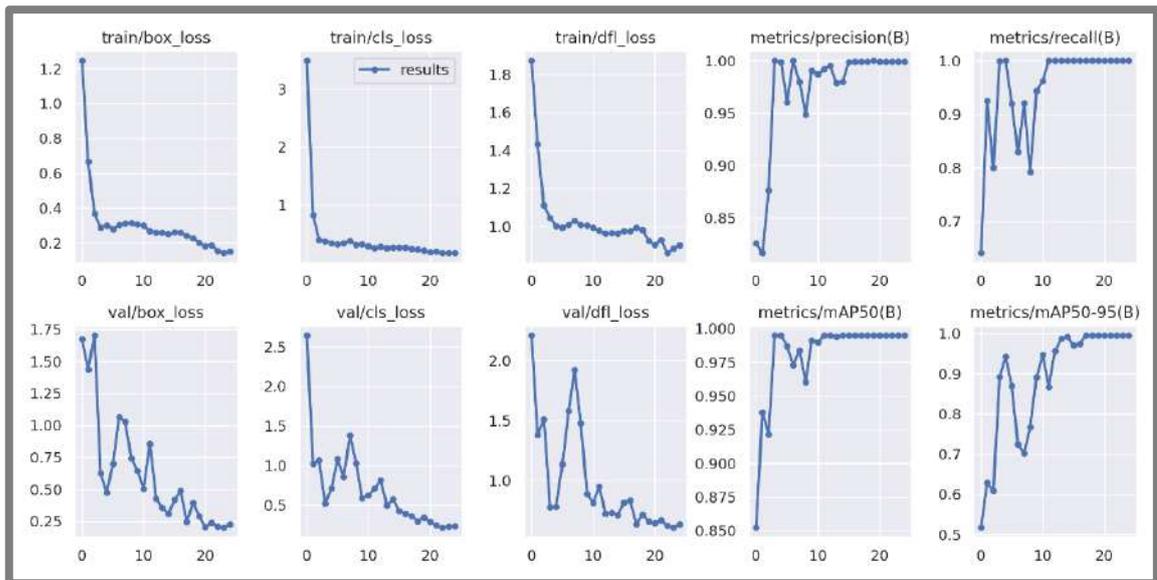


Ilustración 5-2: Graficas de la evolución del entramiento del algoritmo detector de patrones

Realizado por: Argos, J.; Granizo, I., 2023.

En la Ilustración 5-3 se muestran algunos ejemplos de la detección de patrones, donde sin importar la ubicación original de la celda cilíndrica, se selecciona solamente la región de interés, que es recortada y procesada para la clasificación de los patrones, aumentando así su efectividad debido a que se elimina el ruido de los alrededores.

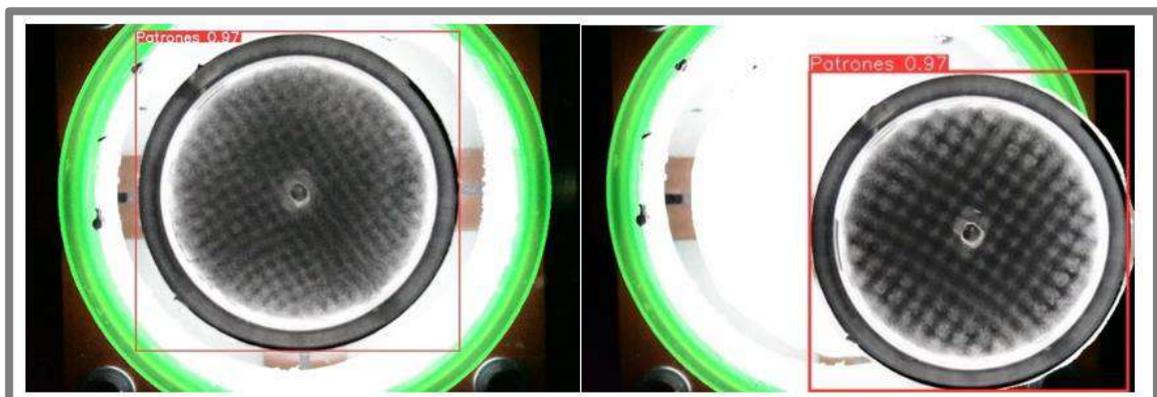


Ilustración 5-3: Resultados de la detección de patrones sin importar el lugar donde se encuentren.

Realizado por: Argos, J.; Granizo, I., 2023.

5.3.1 *Análisis de patrones de anillos*

Para obtener información clara y concisa sobre los patrones de anillos, se ha desarrollado un código para la obtención de información pertinente, como la intensidad de píxeles de forma radial, el número de anillos presentes en el patrón y la distancia promedio de dichos anillos. En la

Ilustración 5-4 se observa la conversión de la imagen a coordenadas polares, donde al hacer un barrido de píxeles se obtiene el valor medio de su intensidad.

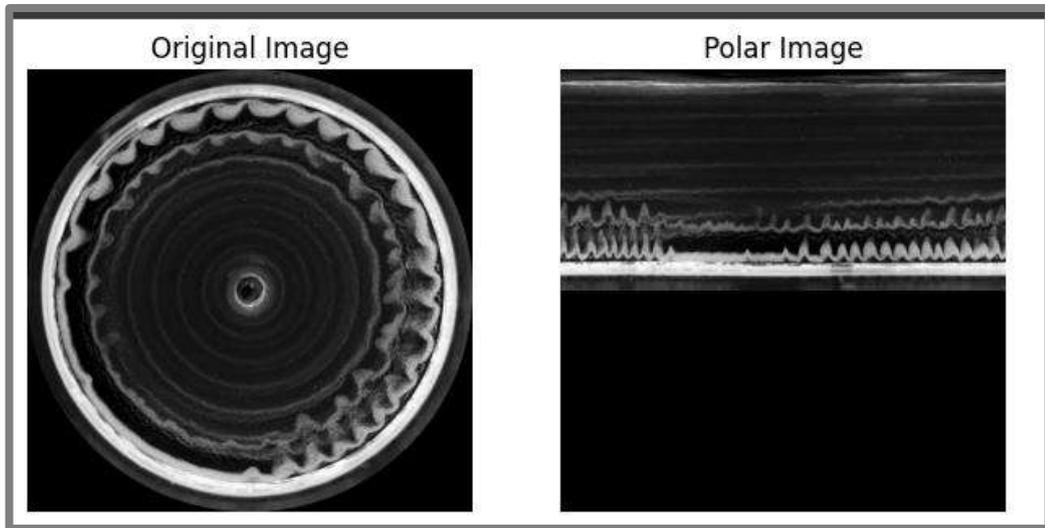


Ilustración 5-4: Conversión de la imagen original a una imagen en coordenadas polares de un patrón de anillos a una frecuencia f de 40 Hz, con una aceleración a_g de 1 g y un nivel espesor de la capa de líquido Δh de 5 ml.

Realizado por: Argos, J.; Granizo, I., 2023.

La información resultante de dicha operación se aprecia en la Ilustración 5-5, donde la imagen original pasa por un filtro de media móvil, con el fin de eliminar el ruido, similar a un filtro pasa bajos.

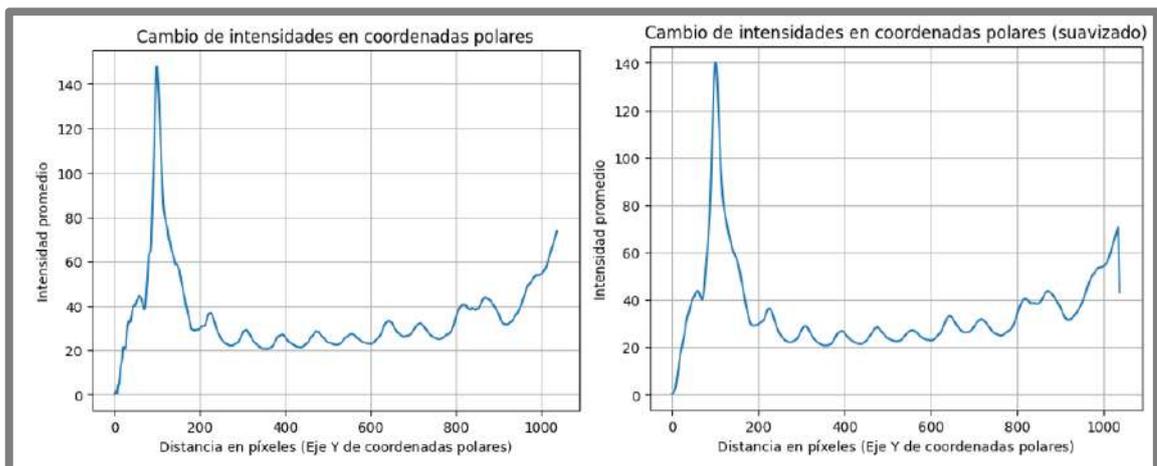


Ilustración 5-5: Izquierda: Intensidad de píxeles vs distancia en píxeles de la imagen original. Derecha: Intensidad de píxeles vs distancia en píxeles del filtrado de la imagen original utilizando la media móvil.

Realizado por: Argos, J.; Granizo, I., 2023.

En la Ilustración 5-6, los puntos rojos representan los picos de la imagen, donde existe la mayor intensidad de un anillo, mientras que los puntos verdes representan su menor intensidad, es decir, los valles de la imagen. Por consiguiente, el número de picos existentes en la imagen indica el número de anillos que contiene la estructura granular.

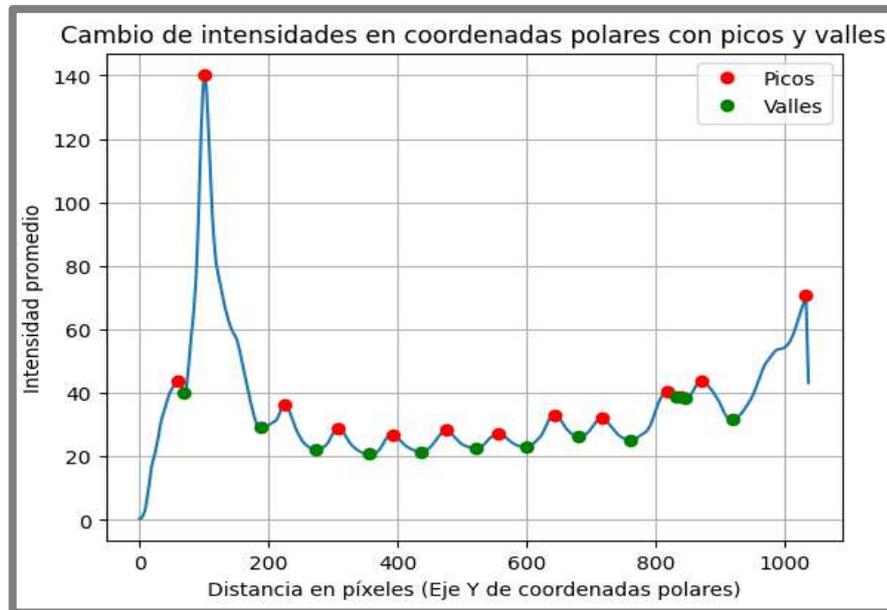


Ilustración 5-6: Localización de los picos y valles.

Realizado por: Argos, J.; Granizo, I., 2023.

La Ilustración 5-7 indica la distancia existente, en píxeles, entre cada pico de la ilustración anterior, estos valores representan el ancho de la onda de los anillos formados en la estructura granular, el cual puede cambiar dependiendo de la aceleración y el espesor de líquido al trabajar con un valor constante de frecuencia.

```

Distancia entre anillos en píxeles:
[ 42 124 84 85 82 81 87 73 103 20 32 162 ]
Distancia promedio entre anillos: 81.25
    
```

Ilustración 5-7: Obtención de la distancia entre anillos y su valor promedio d_{pa} .

Realizado por: Argos, J.; Granizo, I., 2023.

5.3.2 Análisis de patrones diamantes

Mediante un código en LabVIEW se obtiene información relevante acerca de los patrones de diamantes, incluyendo el tamaño de cada uno, el número total de diamantes, el área individual y el porcentaje de área total que ocupan los diamantes respecto al recipiente. Adicionalmente, al

trazar manualmente una línea que atraviese los diamantes, se genera la curva de intensidad al igual que en la Ilustración 5-6, con la particularidad de que, en este caso, los picos corresponden a cada estructura de diamante, indicando la distancia en píxeles entre cada uno.

La Ilustración 5-8 muestra la imagen utilizada para la obtención de datos en patrones de diamantes, donde se requiere el uso manual de la herramienta “línea” del apartado ROI de LabVIEW, ya que los diamantes no siempre siguen una misma recta. Por esta razón, la intervención humana es esencial para determinar la distancia entre los diamantes.

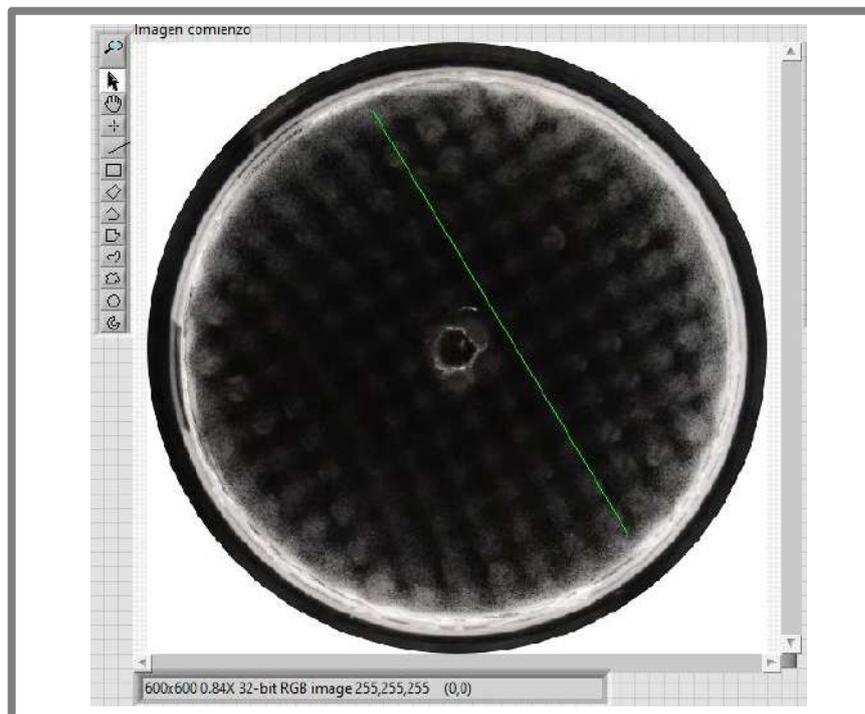


Ilustración 5-8: Ejemplo de imagen utilizada para el análisis de patrones de diamantes. La línea verde es creada manualmente para la obtención de la distancia entre diamantes.

Realizado por: Argos, J.; Granizo, I., 2023.

En la Ilustración 5-9 se observa el procesamiento realizado del ejemplo de la ilustración anterior, con el propósito de obtener la cantidad de diamantes en la imagen y establecer una relación entre el número de diamantes y la frecuencia.

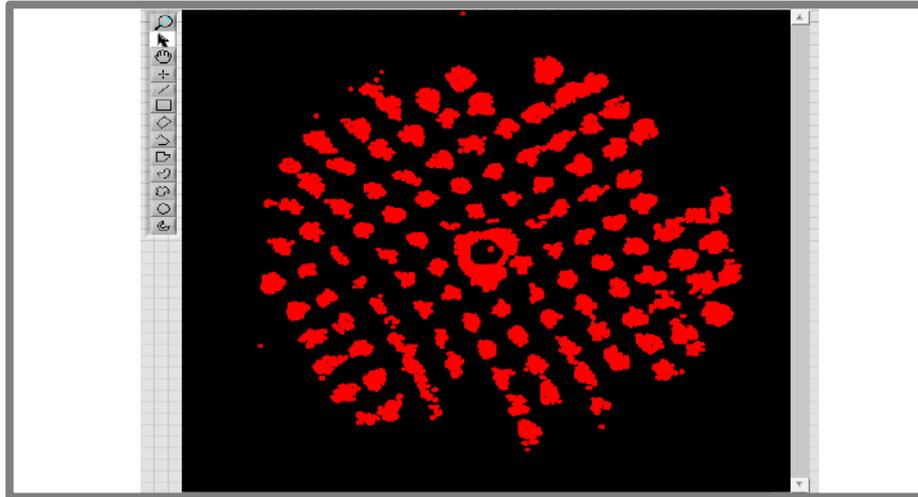


Ilustración 5-9: Imagen de entrada procesada para el conteo de aglomeraciones de partículas, a fin de obtener el número de diamantes con relación a su frecuencia.

Realizado por: Argos, J.; Granizo, I., 2023.

La Ilustración 5-10 representa el estudio de los patrones de diamantes manteniendo todas las partículas posibles, la diferencia con el procesamiento de la ilustración anterior radica en la presencia de partículas muy pequeñas en los alrededores del patrón, cuya presencia aumentaría de forma realmente significativa el conteo de diamantes. De igual manera, las pequeñas uniones entre cada diamante equivaldrían a 1, convirtiéndose en un error, por esta razón se realiza un procesamiento adicional para la eliminación de pequeñas partículas y la separación de las estructuras de diamantes.

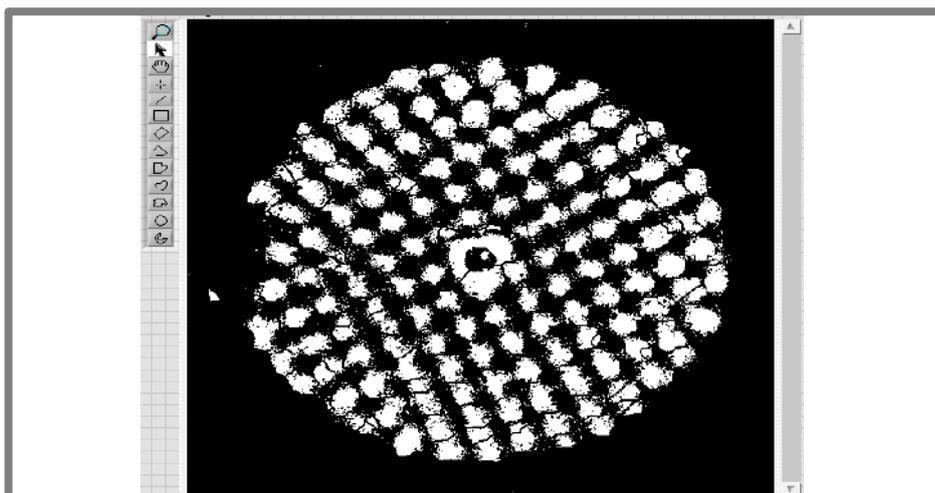


Ilustración 5-10: Imagen de salida tras el procesamiento del patrón de la Ilustración 8-5, para la obtención del porcentaje que ocupan las estructuras respecto al área del fondo del recipiente

Realizado por: Argos, J.; Granizo, I., 2023.

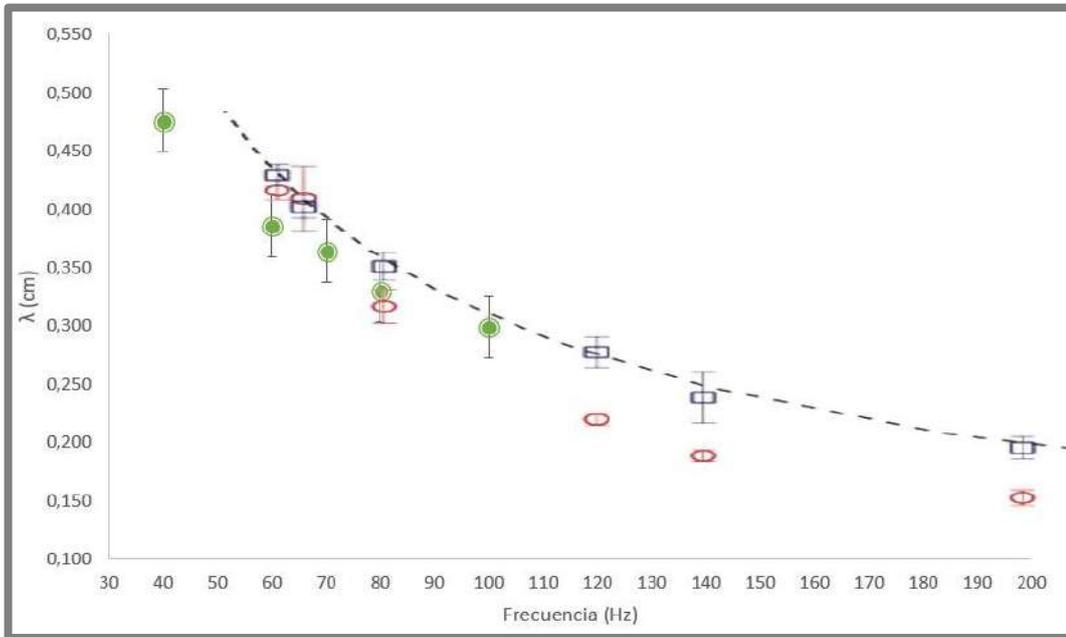


Ilustración 5-12: λ (cm) vs f (Hz). Círculos verdes: valores obtenidos a través del instrumento virtual. Barras verticales: desviación típica. Cuadros azules, círculos rojos y línea discontinua: valores tomados de la tesis doctoral del Dr. Diego Barba (Barba 2020).

Realizado por: Argos, J.; Granizo, I., 2023.

Otro valor que se obtiene a través del instrumento virtual es el número de anillos N_a a distintas frecuencias en la superficie libre de líquido, los resultados se muestran en la Ilustración 5-13.

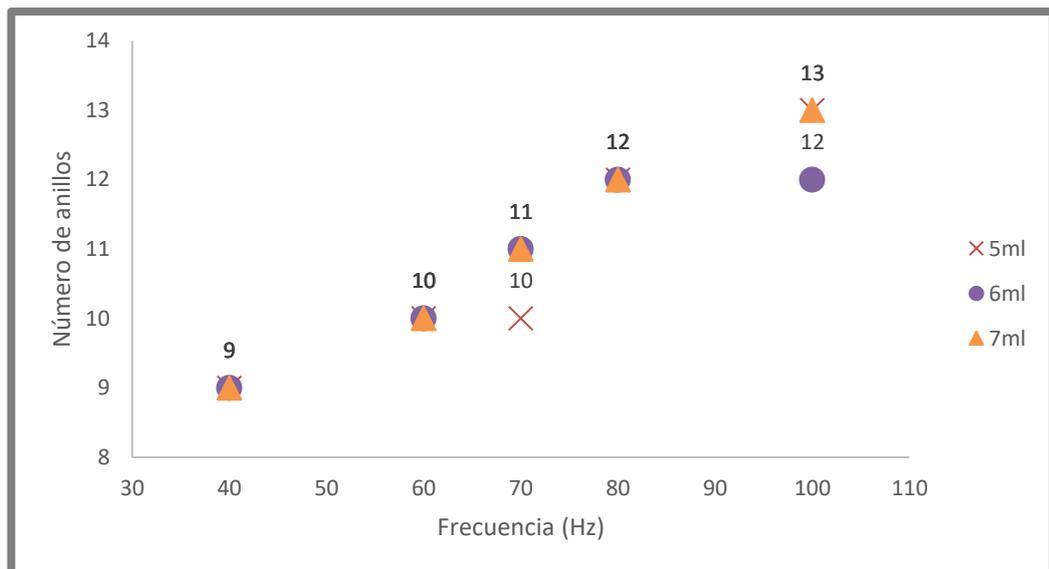


Ilustración 5-13: Relación N_a vs f (Hz) con distintos niveles de espesor de capa de líquido.

Realizado por: Argos, J.; Granizo, I., 2023.

5.4.2 Análisis de los datos en estructuras de diamantes

El procesamiento de las estructuras de diamantes otorga información extra como el número de diamantes N_d , el tamaño promedio de los mismos A_d en mm^2 , el área que ocupan en relación con el fondo del recipiente A_{Td} en %, y mediante el trazo de una línea, se encuentra la distancia promedio d_{pd} en mm.

En la Ilustración 5-14 se evidencia el aumento en la cantidad de estructuras de diamantes en relación con la frecuencia. El conteo de los diamantes se lleva a cabo a partir de la información proporcionada en la Ilustración 5-9.

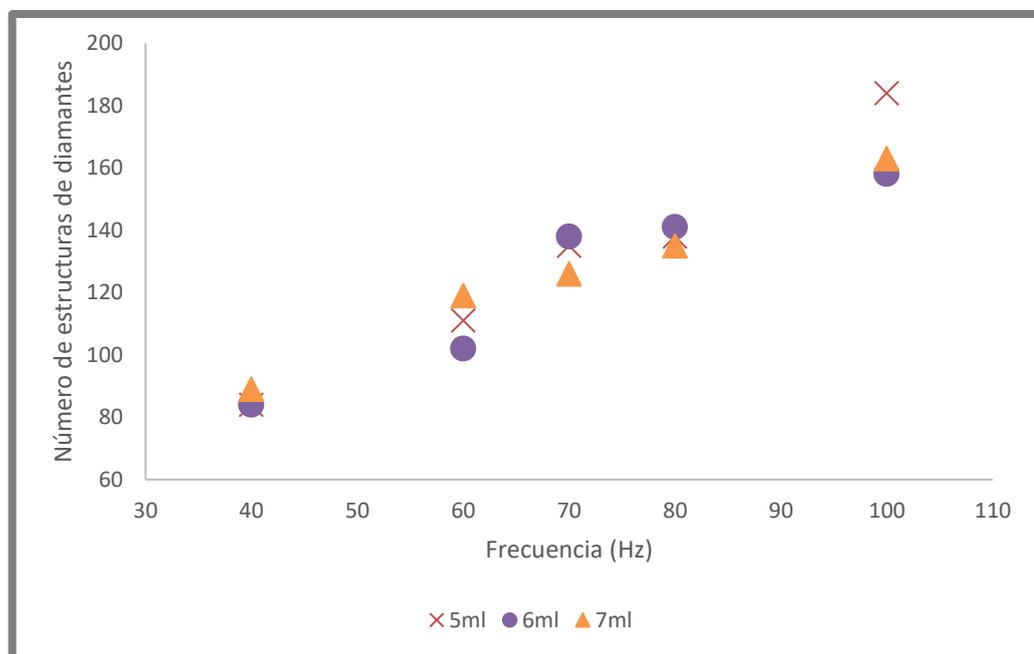


Ilustración 5-14: Relación N_d vs f (Hz).

Realizado por: Argos, J.; Granizo, I., 2023.

La Ilustración 5-15 muestra el porcentaje de área total que ocupan los diamantes en relación con el fondo del recipiente, se mantiene en un rango de entre el 25 al 32 % con cambios irregulares a medida que varían los valores de frecuencia y espesor de la capa de líquido, presentando el porcentaje más bajo en un valor de frecuencia de 100 Hz, aunque sin mayor diferencia en comparación con el resto de porcentajes.

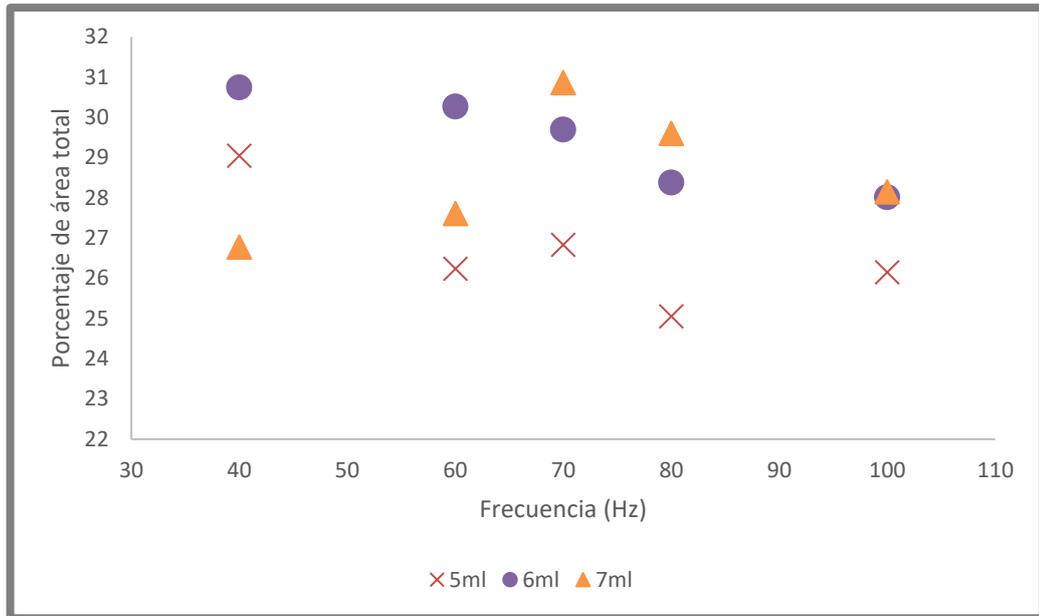


Ilustración 5-15: $A_{Td} (%)$ vs $f (Hz)$ con distintos niveles de espesor de capa de líquido.

Realizado por: Argos, J.; Granizo, I., 2023.

Frente al aumento de valores de frecuencia, se espera que el tamaño o área individual de cada diamante disminuya, ya que se genera una mayor cantidad de ondas que producen un mayor acercamiento de los diamantes, reduciendo de esta manera su tamaño. La Ilustración 5-16 muestra el tamaño o área promedio de los diamantes con relación a la frecuencia de excitación.

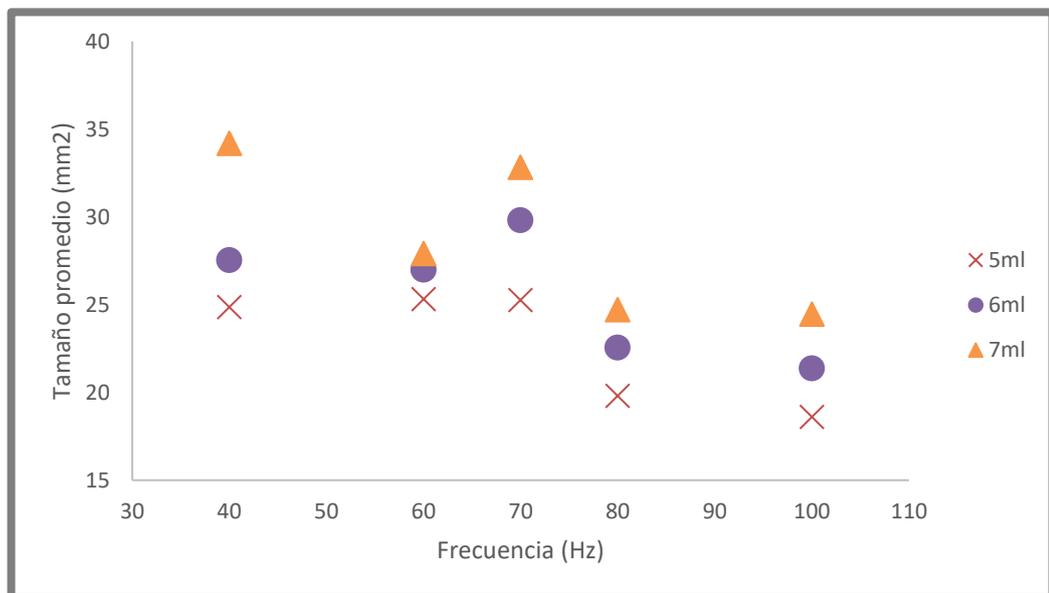


Ilustración 5-16: $A_d (mm^2)$ vs $f (Hz)$ con distintos niveles de espesor de capa de líquido.

Realizado por: Argos, J.; Granizo, I., 2023.

Finalmente, en la Ilustración 5-17 se puede apreciar la distancia que existe entre los diamantes en relación a la frecuencia.

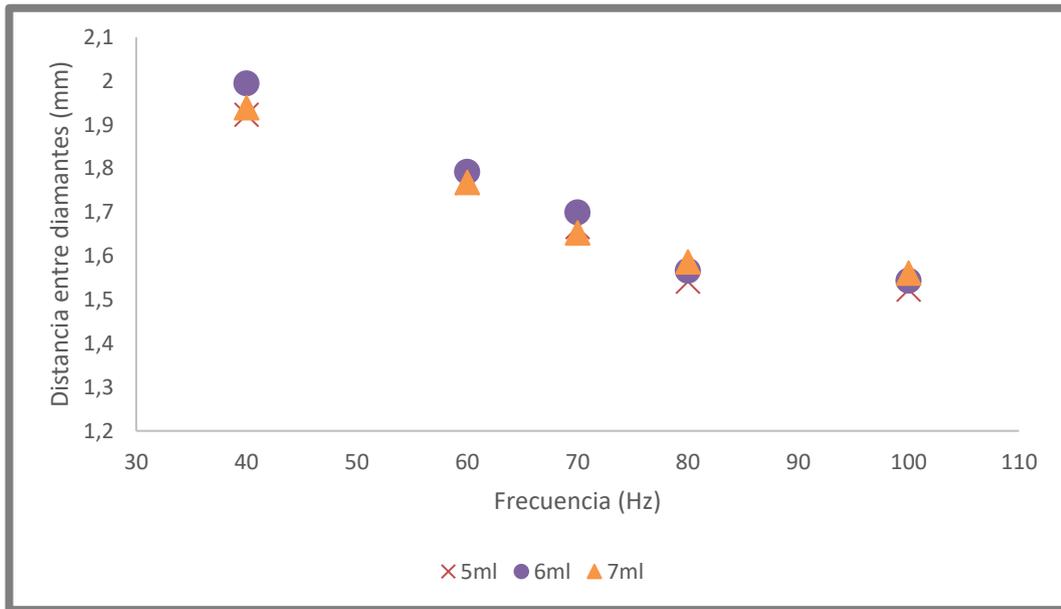


Ilustración 5-17: d_{pd} (mm) vs f (Hz) con distintos niveles de espesor de capa de líquido.

Realizado por: Argos, J.; Granizo, I., 2023.

5.4.3 Análisis de patrones en la celda cuasi 2D

Las pruebas se realizaron a base de etanol puro con un espesor de la capa de partículas h_p de 10 mm a 24 Hz constantes, con una calibración espacial de 0.95 píxel/mm. La Ilustración 5-18 muestra la longitud de onda λ vs la aceleración.

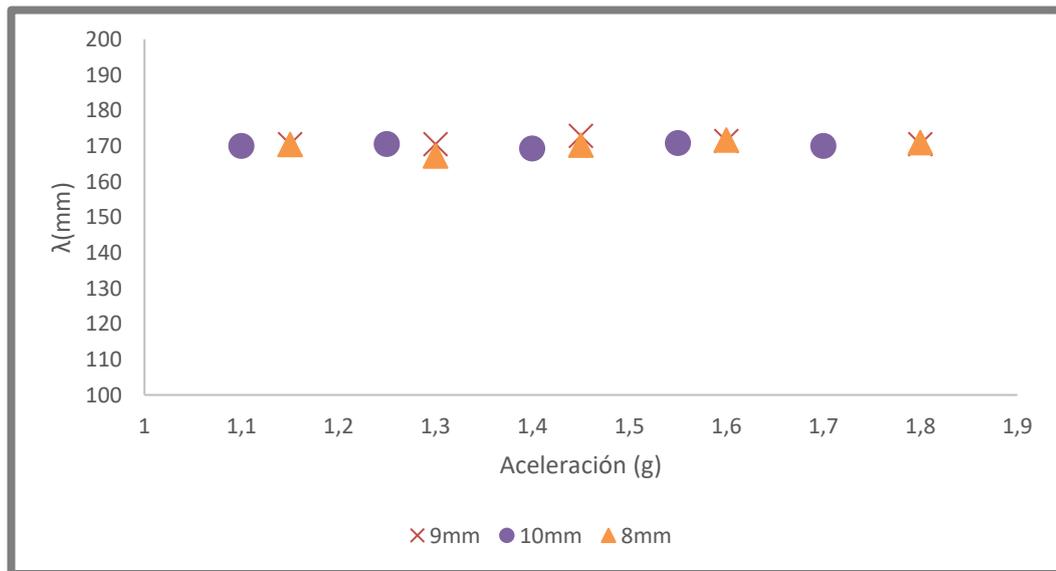


Ilustración 5-18: Longitud de onda λ (mm) vs aceleración a_g (g)

Realizado por: Argos, J.; Granizo, I., 2023

La Ilustración 5-19 muestra la amplitud de onda A en mm vs la aceleración a_g en g a distintos niveles de espesor de la capa de líquido Δh .

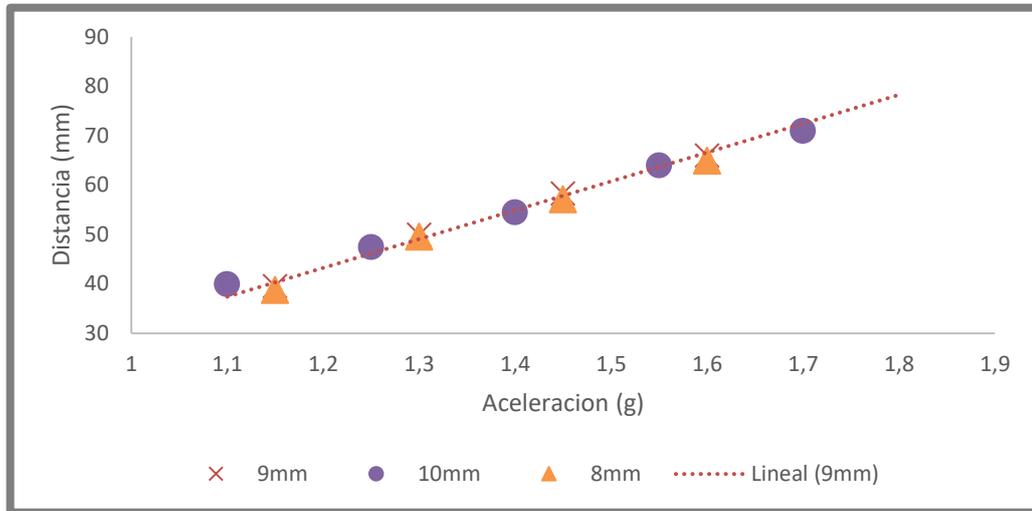


Ilustración 5-19: Amplitud de onda A (mm) vs aceleración a_g (g).

Realizado por: Argos, J.; Granizo, I., 2023

En la Ilustración 5-20 se observa el resultado de cargar un conjunto de imágenes en el instrumento virtual, donde se aprecia el análisis de los picos y valles, armonizados en el mismo valor del eje X, al igual que la altura máxima que alcanza la onda al estar completamente formada, presentando de esta manera, el impacto que se produce sobre el lecho de partículas.

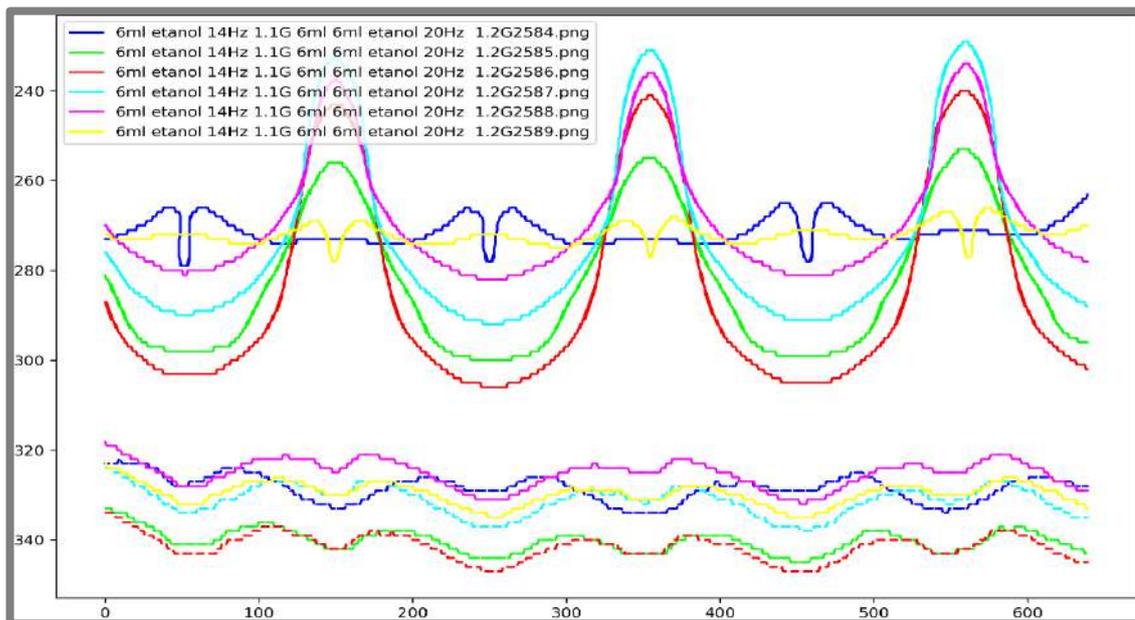


Ilustración 5-20: Semiciclo de onda formado a 14 Hz y a 1.1 g, con un Δh de 6 mm.

Realizado por: Argos, J.; Granizo, I., 2023.

CAPÍTULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

- El porcentaje correcto de los componentes utilizados en la generación estructuras granulares es de vital importancia, ya que los patrones formados en el lecho de partículas dependen del tipo de muestra creada, en este caso en particular, las partículas de 80 μm demostraron adaptarse de mejor manera a la mezcla de glicerina y agua destilada, siendo el diámetro más apto para la realización de experimentos. En caso de usar mayor cantidad de glicerina o menor cantidad de agua, las partículas no sedimentarán correctamente sobre la superficie del recipiente cilíndrico, debido al aumento de viscosidad y densidad de la muestra.
- Al realizar la toma de imágenes de los patrones formados en el recipiente cilíndrico, se utilizó un intervalo de tiempo para cada patrón, demostrando su estabilidad y continuidad con el paso de tiempo. En varios casos se encontraron patrones desconocidos en la superficie libre de líquido, cuyas formas irradiaban estructuras peculiares y de gran ventaja para realizar un estudio profundo de los mismos; dicho estudio no se realizó debido a la ruptura de los patrones en un intervalo de tiempo aleatorio, siendo patrones rechazados debido a su falta de continuidad.
- El diseño del instrumento virtual plantea la idea básica de reemplazar un acelerómetro tradicional en el ámbito de estructuras granulares ya formadas, dando como respuesta la frecuencia, aceleración y espesor de líquido. Además, a través de procesos adicionales, el instrumento extiende su capacidad al extraer detalles cruciales como la longitud de onda, distancias entre estructuras, tamaño promedio, número de anillos, etc. proporcionando así una comprensión más profunda y detallada de las características de las estructuras.
- La validación del instrumento virtual es llevada a cabo conforme a los estándares internacionales, empleando el sistema de medidas internacional, adoptando el inglés como idioma de referencia y garantizando su compatibilidad con diversos sistemas operativos, partiendo de computadores de baja gama. Este logro no solo amplía significativamente el potencial de aplicaciones del instrumento, sino que también lo posiciona como una herramienta de gran valor y versatilidad para investigadores y

profesionales en el campo de la hidrodinámica y disciplinas afines, promoviendo la generación de conocimiento de alta calidad y su adopción a nivel global.

6.2 Recomendaciones

- Para la correcta generación y obtención de estructuras granulares, se debe tomar en cuenta varios factores aparte del tipo de muestra a utilizar, siendo elementos claves los equipos de iluminación, donde un posicionamiento inadecuado o la falta de dicho equipamiento impide la visualización de patrones, tal es el caso del cono de luz led fría, que sin una superficie reflectante presente, no proyectará la luz necesaria sobre el recipiente cilíndrico para visualizar los patrones formados en la superficie libre de líquido.
- En el proceso de sedimentación de las partículas, es fundamental romper la tensión superficial del líquido, debido a que las partículas son menos densas, impidiendo el sumergimiento de las mismas hacia el fondo del recipiente. Sin la ayuda de un surfactante es improbable romper la tensión superficial de la muestra, por lo que la presencia del mismo en la preparación de cualquier muestra en la que las partículas posean una menor densidad que el líquido, es obligatoria.
- Es crucial contar con un conocimiento detallado del hardware que se empleará para garantizar la compatibilidad con el instrumento virtual. Además, contar con el lenguaje de programación adecuado ayuda a un mejor procesamiento de las imágenes obtenidas, con una mayor inclinación al uso de *Python* debido a su amplia variedad de bibliotecas especializadas en el desarrollo y entrenamiento de redes neuronales.
- Para futuros desarrollos de software o versiones mejoradas del instrumento virtual, se recomienda la creación de una red neuronal de autoría totalmente propia en caso de contar con la tecnológica adecuada, utilizando distintas configuraciones de la red (en serie o en paralelo) para enfocar el procesamiento de la red a un solo tipo de estructura, en este caso estructuras granulares. Esto ayudará a reducir aún más el porcentaje de pérdida en cada entrenamiento realizado.

BIBLIOGRAFÍA

ANALOG DEVICES, 2009. Analog. *ADXL326* [en línea]. 2021. Recuperado a partir de: <https://www.analog.com/en/products/adxl326.html#product-overview> [accedido 1 octubre 2023].

BARBA MAGGI, Diego Guillermo, 2020. *DINÁMICA DE SUSPENSIONES CONCENTRADAS SOMETIDAS A VIBRACIÓN MECÁNICA*. Tesis Doctoral. Mecánica de fluidos. Buenos Aires: UNIVERSIDAD DE BUENOS AIRES.

BARBA MAGGI, Diego; et al., 2019. Faraday waves over a permeable rough substrate. *PHYSICAL REVIEW E* [en línea]. Vol. 99, 053110, págs. 2-3, ISSN 24700053. DOI <https://doi.org/10.1103/PhysRevE.99.053110>

BASLER, 2015. Baslerweb. *Basler ace - acA640-750um* [en línea]. 2015. Recuperado a partir de: <https://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace/aca640-750um/> [accedido 28 junio 2023].

BRÜEL & KJAER, 2021. LDS LPA - LINEAR POWER AMPLIFIERS. *LDS LPA - LINEAR POWER AMPLIFIERS* [en línea]. 2021. Recuperado a partir de: <https://www.bksv.com/es/instruments/vibration-testing-equipment/vibration-amplifiers/lds-shaker-amplifier/lds-lpa-amplifiers> [accedido 28 junio 2023].

BRÜEL & KJAER, 2021. LDS V406 VIBRADOR DE IMÁN PERMANENTE. *LDS V406 VIBRADOR DE IMÁN PERMANENTE* [en línea]. 2021. Recuperado a partir de: <https://www.bksv.com/es/instruments/vibration-testing-equipment/lds-shakers/permanent-magnet-shakers/lds-v406> [accedido 28 junio 2023].

CHACÓN RUGELES, Rafael, 2002. La instrumentación virtual en la enseñanza de la Ingeniería Electrónica. *ACCIÓN PEDAGÓGICA*. 2002. Vol. 11, 1, págs. 74-84.

CHEN, Pu; et al., 2014. Microscale assembly directed by liquid-based template. *ADVANCED MATERIALS* [en línea]. Vol. 26, 34, págs. 5936-5940, ISSN 15214095. DOI <https://doi.org/10.1002/adma.201402079>

DENNING, Peter, 2001. Origin of virtual machines and other virtualities. *IEEE Annals of the history of computing* [en línea]. Vol. 23, 3, págs. 73-75. DOI <https://doi.org/10.1109/85.948908>

DOUADY, S., 1990. Experimental study of the Faraday instability. *J. Fluid Mech* [en línea]. Vol. 221, 0, págs. 383-409. DOI <https://doi.org/10.1017/S0022112090003603>

DRAKE, José y LÓPEZ, Patricia, 2009. *VERIFICACIÓN Y VALIDACIÓN SITUACIÓN DENTRO DEL PROCESO DE DESARROLLO*. . Ingeniería de Programación.

FARADAY, Michael, 1837. On a peculiar class of acoustical figures; and on certain forms assumed by groups of particles upon vibrating elastic surfaces. *Philosophical transactions of the*

Royal Society of London [en línea]. Vol. 3, 3, págs. 49-51, ISSN 0365-5695. DOI <https://doi.org/10.1098/rspl.1830.0024>

FENG, Qinrong y MIAO, Duoqian, 2009. Structured Prior Knowledge and Granular Structures. *Springer, Berlin, Heidelberg* [en línea]. Vol. 5819, págs 115-126. DOI https://doi.org/10.1007/978-3-642-04954-5_22

FRANCOIS, N. et al., 2015. Wave-particle interaction in the Faraday waves. *European Physical Journal E* [en línea]. Vol. 38, 106, págs. 1-5, ISSN 1292895X. DOI <https://doi.org/10.1140/epje/i2015-15106-4>

GÓMEZ CUBERO, David, 2000. *HIDRODINÁMICA DE LOS MEDIOS GRANULARES DILUIDOS*. Tesis Doctoral. Ciencias físicas. Sevilla: UNIVERSIDAD DE SEVILLA.

GONZÁLEZ, Adriana, 2021. Liferder. *Estructura Cristalina* [en línea]. 2021. Recuperado a partir de: <https://www.liferder.com/estructura-cristalina/> [accedido 13 octubre 2023].

GONZALEZ, Rafael C. y WOODS, Richard E., 2008. *Digital image processing*. Third Edition. United States of America: Pearson Education. ISBN 9780131687288.

GOOGLE, 2023. Google Colab. *Google Colab* [en línea]. 2023. Recuperado a partir de: <https://colab.google/> [accedido 3 junio 2023].

HASTIE, Trevor, TIBSHIRANI, Robert y FRIEDMAN, Jerome, 2008. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Second Edition. Stanford, California: Springer. ISBN 1282126741.

JARAMILLO VÁSQUEZ, Mildred Korina y MARCATOMA LÓPEZ, Israel Alexis, 2023. *IMPLEMENTACIÓN DE UN SISTEMA VIBRATORIO AUTOMATIZADO MEDIANTE UNA INTERFAZ HMI PARA EL ESTUDIO DE LA ESTRUCTURA GRANULAR A TRAVÉS DE INESTABILIDADES HIDRODINÁMICAS*. Electrónica y Automatización. Riobamba: ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

JUÁREZ SANDOVAL, Oswaldo Ulises, 2013. *Diseño de un practinario para instrumentación virtual a través del protocolo USB 2.0*. Comunicaciones y Electrónica. México D.F.: Instituto Politécnico Nacional.

KHARBEDIA, Mikheil, 2020. *Hidrodinámica superficial en materia blanda: turbulencia de ondas y organización emergente* [en línea]. Tesis Doctoral. Madrid: Universidad Complutense de Madrid. Recuperado a partir de: <https://hdl.handle.net/20.500.14352/11288> [accedido 4 octubre 2023].

LABOMERSA, 2018. Labomersa. *Agitador Magnético con Placa de Calentamiento MS7-H550-Pro* [en línea]. 2018. Recuperado a partir de: <https://labomersa.com/producto/agitador-magnetico-con-placa-de-calentamiento-ms7-h550-pro-2/> [accedido 28 junio 2023].

- LIU, Zhang-Rong; et al.**, 2019. A nonlinear particle packing model for multi-sized granular soils. *Construction and Building Materials* [en línea]. Vol. 221, págs. 274-282, ISSN 09500618. DOI <https://doi.org/10.1016/j.conbuildmat.2019.06.075>
- MA, Gang; et al.**, 2019. Evolution of particle size and shape towards a steady state: Insights from FDEM simulations of crushable granular materials. *Computers and Geotechnics* [en línea]. Vol. 112, págs. 147-158, ISSN 18737633. DOI <https://doi.org/10.1016/j.compgeo.2019.04.022>
- MÁNUEL, Antoni; et al.**, 2001. *Instrumentación virtual: Adquisición, procesado y análisis de señales*. Primera edición. Barcelona: Barcelona Digital, ISBN 84-8301-473-4.
- MATHWORKS**, 2016. Mathworks. *MATLAB* [en línea]. 2016. Recuperado a partir de: <https://la.mathworks.com/products/matlab.html> [accedido 21 agosto 2023].
- MENDIBURU DÍAZ, Henry Antonio**, 2006. *Instrumentación virtual industrial*. Primera edición. Lima: Indecopi. Vol. 1.
- NATIONAL INSTRUMENTS**, 2012. NI myDAQ. *NI myDAQ* [en línea]. 2012. Recuperado a partir de: <https://www.ni.com/docs/en-US/bundle/mydaq-specs/page/specs.html> [accedido 1 octubre 2023].
- NATIONAL INSTRUMENTS**, 2020. NI. *LabVIEW* [en línea]. 2020. Recuperado a partir de: <https://www.ni.com/es-cr/shop/software/products/labview.html> [accedido 3 junio 2023].
- NIKON**, 2018. NIKON. *DSLR Nikon D7500* [en línea]. 2018. Recuperado a partir de: <https://www.nikon.com.mx/nikon-products/product/dslr-cameras/d7500.html> [accedido 28 junio 2023].
- PINEDA OLIVARES, Alejandro**, 2013. *Instrumentación virtual. Fundamentos de programación gráfica con LabVIEW*. Primera edición. Monterrey: Editorial Digital Tecnológico de Monterrey. Vol. 1.
- RADWAG**, 2018. Balanza de precisión - PS 4500.X2.M. *Balanza de precisión - PS 4500.X2.M* [en línea]. 2018. Recuperado a partir de: <https://radwag.com/es/balanza-de-precision-ps-4500-x2-m,w1,M27,101-102-114> [accedido 28 junio 2023].
- RASCHKA, Sebastian**, 2015. *Python machine learning: Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics*. Birmingham: Packt Publishing. Vol. 1, ISBN 9781783555130.
- REYES MELO, Martín E. y HINOJOSA RIVERA, Moisés**, 2000. Estructura de sólidos amorfos. *Ingenierías*. Vol. 3, 9, págs. 7-14.
- KIESGEN de RICHTER, S.; et al.**, 2015. Vibration-induced compaction of granular suspensions. *European Physical Journal E* [en línea]. Vol. 38, 74, págs. 1-9, ISSN 1292895X. DOI <https://doi.org/10.1140/epje/i2015-15074-7>

SAYLOR, J. R. y KINARD, A. L., 2005. Simulation of particle deposition beneath Faraday waves in thin liquid films. *Physics of Fluids* [en línea]. Vol. 17, 4, ISSN 10706631. DOI <https://doi.org/10.1063/1.1884111>

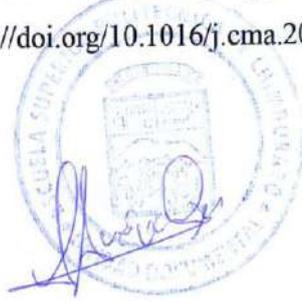
TEQUIPMENT, 2017. Tequipment. *Instek SFG-2104 4 Mhz DDS Function Generator w/ Counter, Sweep & AM/FM* [en línea]. 2017. Recuperado a partir de: <https://www.tequipment.net/InstekSFG-2104.html> [accedido 28 junio 2023].

VISUAL STUDIO, 2015. Visual Studio Code. *Visual Studio Code* [en línea]. 2015. Recuperado a partir de: <https://code.visualstudio.com/docs> [accedido 6 junio 2023].

WANG, Lingfeng y TAN, Kay Chen, 2006. *Modern Industrial Automation Software Design Principles and Real-World Applications*. First edition. Hoboken, New Jersey: John Wiley & Sons, Inc. Vol. 1, ISBN 0471683736.

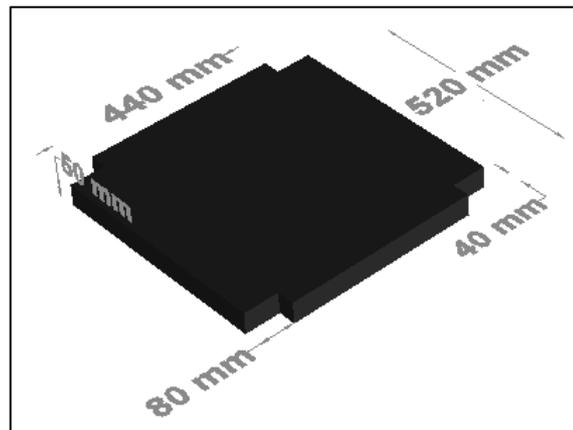
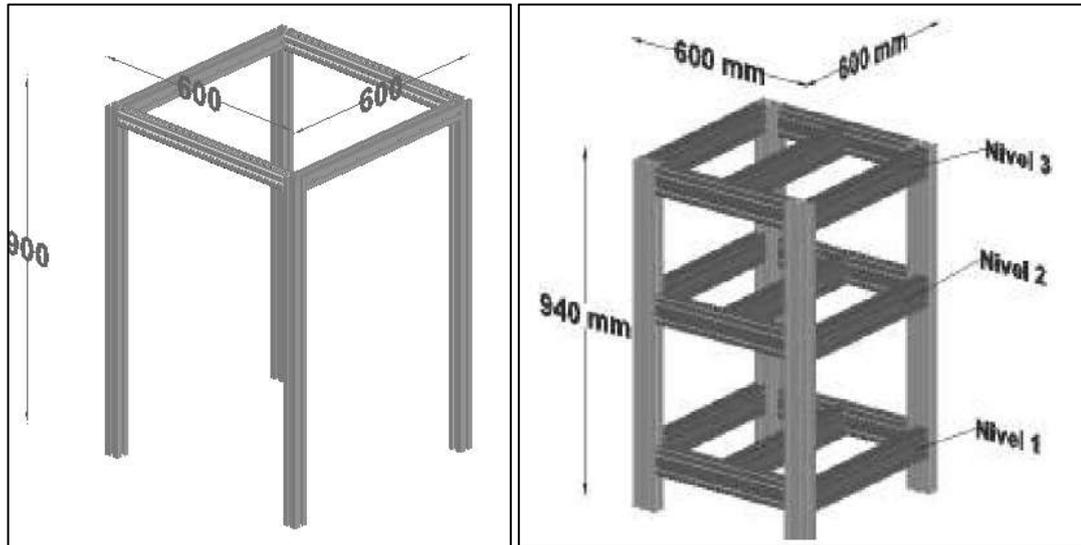
ZAPATA, M., 2022. Guía de estudios. *CLASIFICACIÓN DE LOS SÓLIDOS: AMORFOS Y CRISTALINOS* [en línea]. 2022. Recuperado a partir de: <https://quimicaencasa.com/clasificacion-de-los-solidos-amorfos-y-cristalinos/> [accedido 13 octubre 2023].

ZHANG, Pin y YIN, Zhen Yu, 2021. A novel deep learning-based modelling strategy from image of particles to mechanical properties for granular materials with CNN and BiLSTM. *Computer Methods in Applied Mechanics and Engineering* [en línea]. Vol. 382. DOI <https://doi.org/10.1016/j.cma.2021.113858>



ANEXOS

Anexo A: Dimensiones de la estructura antivibratoria en mm.



Anexo B: Ficha de datos del generador de señales SFG-2104.

SPECIFICATIONS								
	SFG-2000 Series				SFG-2100 Series			
MAIN	SFG-2004	SFG-2007	SFG-2010	SFG-2020	SFG-2104	SFG-2107	SFG-2110	SFG-2120
Frequency Range(For Sine, Square) Range(For Triangle) Resolution Stability Accuracy Aging Output Function Amplitude Range Impedance Attenuator DC Offset Duty Control Range Resolution Display	0.1Hz~ 4MHz	0.1Hz~ 7MHz	0.1Hz~ 10MHz	1Hz~ 20MHz	0.1Hz~ 4MHz	0.1Hz~ 7MHz	0.1Hz~ 10MHz	1Hz~ 20MHz
	0.1Hz~1MHz (1Hz ~ 1MHz for SFG-2020/2120) 0.1Hz (1Hz for SFG-2020/2120) ± 20 ppm ± 20 ppm ± 5 ppm / year Sine, Square, Triangle 2mV ~ 10Vpp(into 50Ω load) 50Ω ±10% -20dB±1dB×2 <-5V ~ >+5V(into 50Ω load) 20% to 80% , 2Hz ~1MHz (Square wave only) 1% 9 digits LED display							
SINE WAVE								
Harmonics Distortion	-55dBc,0.1Hz~200kHz; -40dBc,0.2MHz~4MHz; -30dBc,4MHz~10MHz (Specification applied to both TTL/CMOS OFF and from MAX. to 1/10 level)							
Flatness(Relative to 1kHz)	<±0.3dB, 0.1Hz~1MHz; <±0.5dB, 1MHz~4MHz; <±2dB, 4MHz~10MHz							
TRIANGLE WAVE								
Linearity	≥98%,0.1Hz~100kHz; ≥95%,100kHz~1MHz							
SQUARE WAVE								
Symmetry Rise or Fall Time	±1% of period + 4ns, 0.1Hz~100kHz ≤25ns at maximum output. (into 50Ω load)							
CMOS OUTPUT								
Level	4Vpp±1Vpp~15Vpp±1Vpp adjustable; Rise or Fall Time ≤120ns							
TTL OUTPUT								
Level Fan Out Rise and Fall Time	≥3Vpp 20 TTL load ≤25ns							
SWEEP OPERATION								
Rate Time Mode	—				100:1 ratio max. and adjustable(*) 1Sec~30Sec adjustable(**) Lin./Log. switch selector			
AMPLITUDE MODULATION								
Depth & Modulation Frequency Carrier BW EXT Modulation Sensitivity	—				0~100% ; 400Hz(INT), DC~1MHz(EXT) 100Hz~5MHz(-3dB) ≤10Vpp for 100% modulation			
FREQUENCY MODULATION								
Deviation & Modulation Frequency EXT Modulation Sensitivity	—				≥±50kHz, center at 1MHz, 400Hz fixed(INT),1kHz fixed(EXT) ≤10Vpp for 10% modulation(center at 1kHz)			
FREQUENCY COUNTER								
Range Accuracy Time base Resolution Input Impedance Sensitivity	—				5Hz~150MHz Time base accuracy±1 count ±20ppm(23℃±5℃) after 30 minutes warm up 100nHz for 1Hz ; 0.1Hz for 100MHz 1MΩ /150pf ≤35mVrms (5Hz~100MHz) ≤45mVrms (100MHz~150MHz)			

NOTE : 1.(*) In order to get the maximum sweep span, the sweep time needs to be tuned on when adjusting the sweep span.

2.(**) When the sweep time is too long, the stop frequency will reach and stay at the maximum frequency of the instrument until the end of the sweep cycle.

SPECIFICATIONS								
	SFG-2000 Series				SFG-2100 Series			
	SFG-2004	SFG-2007	SFG-2010	SFG-2020	SFG-2104	SFG-2107	SFG-2110	SFG-2120
STORE/RECALL FUNCTION								
	10 groups of panel settings							
POWER SOURCE								
	AC115V±10%, AC230V+10%/-15%, 50/60Hz							
DIMENSION & WEIGHT								
	266(W)x107(H)x293(D) mm; Approx. 3.1kg				266(W)x107(H)x293(D) mm; Approx. 3.2kg			

ORDERING INFORMATION	
SFG-2004	4MHz DDS Function Generator
SFG-2007	7MHz DDS Function Generator
SFG-2010	10MHz DDS Function Generator
SFG-2020	20MHz DDS Function Generator
SFG-2104	4MHz DDS Function Generator with Counter, Sweep & AM, FM Modulation
SFG-2107	7MHz DDS Function Generator with Counter, Sweep & AM, FM Modulation
SFG-2110	10MHz DDS Function Generator with Counter, Sweep & AM, FM Modulation
SFG-2120	20MHz DDS Function Generator with Counter, Sweep & AM, FM Modulation
ACCESSORIES :	
User manual x 1, Power Cord x 1	
GTL-101 test lead x1 (SFG-2000 Series)	
GTL-101 test lead x2 (SFG-2100 Series)	

SELECTION GUIDE								
FREQUENCY RANGE	4MHz		7MHz		10MHz		20MHz	
MODEL	SFG-2004	SFG-2104	SFG-2007	SFG-2107	SFG-2010	SFG-2110	SFG-2020	SFG-2120
DUTY	✓	✓	✓	✓	✓	✓	✓	✓
TTL/CMOS	✓	✓	✓	✓	✓	✓	✓	✓
DC OFFSET	✓	✓	✓	✓	✓	✓	✓	✓
LIN/LOG SWEEP		✓		✓		✓		✓
AM/FM MODULATION		✓		✓		✓		✓
EXT COUNTER		✓		✓		✓		✓

Anexo C: Ficha de datos del amplificador de potencia lineal LDS LPA600.

	LPA100	LPA600	LPA1000
Classification	Class B linear amplifier, air-cooled		
Input Supply ($\pm 10\%$)	100, 120 and 230 V at 50/60 Hz		
Input kVA	<0.44 kVA	<1.85 kVA	<2.70 kVA
Rated Power Output	94 VA in 3.15 Ω	656 VA in 2.5 Ω	961 VA in 4.0 Ω
Maximum Power Output Capacity	154 VA in 3.15 Ω	810 VA in 2.5 Ω	1296 VA in 4.0 Ω
Gain	22 V/V ± 2 dB max.	45 V/V ± 2 dB max.	72 V/V ± 2 dB max.
Monitoring Output – voltage	0.1 V/V $\pm 3\%$, 5 Hz to 15 kHz	0.05 V/V $\pm 3\%$, 5 Hz to 10 kHz	
Monitoring Output – current	0.1 V/A $\pm 3\%$, 5 Hz to 15 kHz	0.1 V/A $\pm 3\%$, 5 Hz to 10 kHz	
Frequency Range at Maximum Power	dc to 15 kHz for 30 min at maximum VA	40 Hz to 10 kHz for 30 min at maximum VA	
Total Harmonic Distortion at Rated Output	<0.1%, 15 Hz to 5 kHz <0.2%, 5 kHz to 15 kHz	<0.2%, 40 Hz to 5 kHz <0.3%, 5 kHz to 10 kHz	
Maximum Output Voltage – no load	22 Vrms, dc to 15 kHz	45 Vrms, dc to 10 kHz	72 Vrms, dc to 10 kHz
Output Current at Rated VA	5.5 A rms	16.2 A rms	15.5 A rms
Maximum Output Current	7.0 A rms, dc to 15 kHz for 30 min	17.75 A rms, 40 Hz to 10 kHz for 30 min	
Signal-to-noise Ratio	>95 dB		
Amplifier Efficiency	52%	57%	64%
Sound Power Level at 2 m (6.6 ft)	41 dBA	45 dBA	49 dBA
Maximum Ambient Working Temperature	35 °C (95 °F)		
Drive Input Connection	BNC		
Weight	14.0 kg (31 lb)	27 kg (60 lb)	33 kg (73 lb)
Height – excluding feet	88 mm (3.5 in)	132 mm (5.2 in)	
Width	482.6 mm (19.0 in), with flanges for standard 19-in rack mounting		
Depth	450 mm (17.7 in)		550 mm (21.6 in)

Armature interface

	V100 SHAKER SERIES	V200 SHAKER SERIES	V400 SHAKER SERIES	V450 SHAKER SERIES	V455 SHAKER SERIES
Armature Diameter	–	–	38.0 mm (1.5 in)	63.5 mm (2.5 in)	63.5 mm (2.5 in)
Insert Options	M4, 6/32 UNF	M4, 10/32 UNF	M4, 10/32 UNF	M5, 10/32 UNF	M5, 10/32 UNF
Insert Pattern	1 insert on central spigot	1 insert on central spigot	1 insert at centre, 6 inserts equispaced on 25.4 mm (1.0 in) PCD*	1 insert at centre, 5 inserts equispaced on 50.0 mm (1.97 in) PCD*	1 insert at centre, 5 inserts equispaced on 50.0 mm (1.97 in) PCD*

* PCD = pitch circle diameter

Cooling fans

	V100 SHAKER SERIES	V200 SHAKER SERIES	V400 SHAKER SERIES	V450 SHAKER SERIES	V455 SHAKER SERIES
Dimensions	–	–	249 × 241 × 389 mm (9.8 × 9.5 × 15.3 in)	249 × 241 × 186 mm (9.8 × 9.5 × 7.5 in)	249 × 241 × 186 mm (9.8 × 9.5 × 7.5 in)
Weight	–	–	17 kg (37 lb)	15 kg (33 lb)	15 kg (33 lb)
Electrical Requirement	–	–	Single phase: 230 V (50 Hz) or 120 V/230 V (60 Hz)		

Safety

Complies with the following EU directives:

- Machinery: 2006/42/EC
- Low Voltage: 2014/35/EU
- EMC: 2014/30/EU
- Designed in accordance with EN 61010-1:2010

Further information

For further details on LDS products and systems, including outline drawings, please contact your local HBK representative.

Anexo D: Ficha de datos del vibrador de imán permanente LDS V406.

	V100 SERIES	V200 SERIES	V400 SERIES	V450 SERIES	V455 SERIES
Recommended LDS Amplifier	LPA100	LPA100	LPA100 – no fan LPA600 – with fan	LPA600	LPA1000
Natural Air Cooling – no fan					
Sine Peak Force	8.9 N (2.0 lbf)	17.8 N (4.0 lbf)	98.0 N (22.0 lbf)	177.0 N (39.8 lbf)	–
Random Force rms	–	–	38.0 N (8.5 lbf)	–	–
Half-sine Shock Force*	–	–	90.0 N (21.0 lbf)	–	–
Velocity Sine Peak	1.31 m/s (51.6 in/s)	1.49 m/s (58.7 in/s)	1.52 m/s (59.8 in/s)	1.78 m/s (70.1 in/s)	–
Acceleration Sine Peak	1373 m/s ² (140.0 g _n)	892 m/s ² (91.0 g _n)	490 m/s ² (50.0 g _n)	415 m/s ² (42.3 g _n)	–
Acceleration Random rms	–	–	190 m/s ² (19.4 g _n)	–	–
Displacement Peak-Peak [†]	2.5 mm (0.1 in)	5.0 mm (0.2 in)	14.0 mm (0.55 in)	19.0 mm (0.75 in)	–
Forced Air Cooling – with fan					
Sine Peak Force	–	–	196 N (44.0 lbf)	311 N (70.0 lbf)	489 N (110 lbf)
Random Force rms	–	–	89 N (20.0 lbf)	214 N (48.0 lbf)	290 N (66.0 lbf)
Half-sine Shock Force*	–	–	200 N (44.0 lbf)	460 N (103 lbf)	730 N (163 lbf)
Velocity Sine Peak	–	–	1.78 m/s (70.1 in/s)	1.78 m/s (70.1 in/s)	2.50 m/s (98.4 in/s)
Acceleration Sine Peak	–	–	980 m/s ² (100.0 g _n)	730 m/s ² (74.4 g _n)	1147 m/s ² (117.0 g _n)
Acceleration Random rms	–	–	446 m/s ² (45.5 g _n)	501 m/s ² (51.1 g _n)	686 m/s ² (70.0 g _n)
Displacement Peak-Peak [†]	–	–	17.6 mm (0.69 in)	19.0 mm (0.75 in)	–
Mass of Moving Elements	0.0065 kg (0.014 lb)	0.02 kg (0.044 lb)	0.2 kg (0.44 lb)	–	0.426 kg (0.94 lb)
Body Mass					
Base-mounted Shaker	0.91 kg (2.0 lb)	1.81 kg (4.0 lb)	14.1 kg (31.0 lb)	–	64.0 kg (141.1 lb)
Trunnion-mounted Shaker	–	3.17 kg (7.0 lb)	22.7 kg (50.0 lb)	–	81.7 kg (180.1 lb)
Armature Resonance (f_a)	12000 Hz	13000 Hz	9000 Hz	–	5600 Hz
Usable Frequency Range	dc to 12000 Hz	dc to 13000 Hz	dc to 9000 Hz	–	dc to 7500 Hz
Acoustic Noise[‡]					
Shaker – naturally cooled	<70 dBA	75 dBA	82 dBA	–	105 dBA
Shaker – forced-air cooled	–	75 dBA	105 dBA	–	105 dBA
Total Heat Dissipation	9.5 W	48.0 W	340 W	820 W	1.6 kW
Maximum Ambient Working Temperature	30 °C (86 °F)				
Maximum Dimensions					
Base-mounted Shaker	H 89.0 × ∅ 65.0 mm H 3.25 × ∅ 2.6 in	H 96.0 × ∅ 78.0 mm H 3.8 × ∅ 3.1 in	H 198.4 × ∅ 165 mm H 7.8 × ∅ 6.5 in	–	H 290 × ∅ 265 mm H 11.4 × ∅ 10.4 in
Trunnion-mounted Shaker	–	120 × 117 × 100 mm 4.7 × 4.6 × 3.9 in	274 × 259 × 165 mm 10.8 × 10.2 × 6.5 in	–	395 × 375 × 275 mm 15.6 × 14.8 × 10.8 in

* Half-sine shock force is calculated with the standard payload, 2 ms pulsewidth, 10% pre/post pulse.

† Displacement can vary with payload and shaker orientation. Please contact HSK for advice on specific test requirements.

‡ Acoustic noise is measured at a distance of 1 m (3.3 ft) and at a height of 1.6 m (5.2 ft) above floor level in an enclosed cell.

Anexo E: Ficha de datos de la balanza de precisión PS 4500.X2.M.

Datos técnicos

Metrological parameters	
Maxima capacidad [Max]	4500 g
Minima capacidad	0,5 g
Legibilidad [d]	0,01 g
División de legalización [e]	0,1 g

Metrological parameters	
Rango de tara	-4500 g
Pesada mínima usp	10 g
Pesada mínima	1 g
Repetibilidad (Max)	0,008 g
Repetibilidad (5% Max)	0,005 g
Linealidad	±0,02 g
Tiempo de estabilización	1,5 s
Calibración	interna (automatica)
Clase OIML	II
Deriva de temperatura de sensibilidad	$2 \times 10^{-6} / ^\circ\text{C} \times \text{Rt}$
Grado de protección	IP 43
Humedad relativa de aire	40% ÷ 80%

Physical parameters	
Sistema de nivelación	manual
Pantalla	5" graphic color touchscreen
Elementos del set	Balanza, platillo, protección del platillo, adaptador de CA
Dimensión de platillo	195×195 mm
Dimensiones de aparato	333×206×107 mm
Dimensiones de embalaje	465×370×290 mm
Masa neta	4,5 kg
Masa bruta	5,5 kg

Features of use	
Tamaño de base de datos	7
Touchless operation	2 sensores infrarrojos
Communication interface	
Conectividad	2×RS232 ¹ , USB-A, USB-B, Ethernet, Wi-Fi
Electrical parameters	
Alimentación	Adapter: 100 – 240V AC 50/60Hz 0,6A; 12V DC 1,2A Balanza: 12 – 15V DC 0,8A max
Potencia consumida por el dispositivo	4 W
Environmental conditions	
Temperatura de trabajo	+10 ÷ +40 °C
Ambient conditions monitoring	THBR 2.0 System, THBR BOX, THB P, THB W, THB S
Temperatura de almacenamiento	-20 ÷ +50 °C

La repetibilidad se expresa como una desviación estándar de 10 posiciones de carga.

El tiempo de estabilización depende de las condiciones externas y la dinámica de colocar los pesos en el platillo; especificado para el perfil FAST.

¹ Los lectores de códigos de barras disponibles como accesorio cooperan con la balanza utilizando solo la interfaz RS232.

Anexo F: Ficha de datos del agitador magnético MS7-H550-Pro.

Specifications	MS7-H550-Pro	MS7-H550-S
Work plate Dimension [W x D]	184x184mm (7 inch)	184x184mm (7 inch)
Work plate material	Glass ceramic	Glass ceramic
Motor type	Brushless DC motor	Shaded pole motor
Stirring positions	1	1
Max. stirring quantity, [H ₂ O]	20L	10L
Max. magnetic bar[length]	80mm	80mm
Speed range	100-1500rpm resolution ±1rpm	0-1500rpm
Speed display	LCD	Scale
Temperature display	LCD	LED
Heating temperature range	Room Temp.-550°C, increment 1°C	Room temp.-550°C, increment 5°C
Temperature control accuracy	±1°C(<100°C) ±1%(>100°C)	±10°C
Overheating protection	580°C	580°C
Temperature display accuracy	±0.1°C	±1°C
External temperature sensor	PT1000 (accuracy ±0.2)	PT1000 (accuracy ±0.2)
"Hot" warning	50°C	50°C
Data connector	RS232	-
Protection class	IP21	IP21
Motor rating input	18W	15W
Motor rating output	10W	1.5W
Power	1050W	1030W
Heating output	1000W	1000W
Voltage	100-120/200-240V 50/60Hz	100-120/200-240V 50/60Hz
Dimension [WxDxH]	215x360x112mm	215x360x112mm
Weight	5.3kg	4.5kg
Permissible ambient temperature and humidity	5-40°C, 80%RH	5-40°C, 80%RH

Specifications

■ Nikon D7500 Digital Camera

Type	
Type	Single-lens reflex digital camera
Lens mount	Nikon F mount (with AF coupling and AF contacts)
Effective angle of view	Nikon DX format; focal length in 35 mm [135] format equivalent to approx. 1.5× that of lenses with FX format angle of view
Effective pixels	
Effective pixels	20.9 million
Image sensor	
Image sensor	23.5 × 15.7 mm CMOS sensor
Total pixels	21.51 million
Dust-reduction System	Image sensor cleaning, Image Dust Off reference data (requires NX Studio)
Storage	
Image size (pixels)	<ul style="list-style-type: none"> • DX (24×16) image area 5568 × 3712 (Large: 20.7 M) 4176 × 2784 (Medium: 11.6 M) 2784 × 1856 (Small: 5.2 M) • 1.3× (18×12) image area 4272 × 2848 (Large: 12.2 M) 3200 × 2136 (Medium: 6.8 M) 2128 × 1424 (Small: 3.0 M) • Photographs with image area of DX taken during movie recording 5568 × 3128 (Large) 4176 × 2344 (Medium) 2784 × 1560 (Small)

Storage	
Image size (pixels)	<ul style="list-style-type: none"> • Photographs with image area of 1.3× taken during movie recording 4272 × 2400 (Large) 3200 × 1800 (Medium) 2128 × 1192 (Small) • Photographs taken during movie recording at a frame size of 3840 × 2160: 3840 × 2160
File format	<ul style="list-style-type: none"> • NEF (RAW): 12 or 14 bit, lossless compressed or compressed • JPEG: JPEG-Baseline compliant with fine (approx. 1 : 4), normal (approx. 1 : 8), or basic (approx. 1 : 16) compression; optimal quality compression available • NEF (RAW) + JPEG: Single photograph recorded in both NEF (RAW) and JPEG formats
Picture Control System	Auto, Standard, Neutral, Vivid, Monochrome, Portrait, Landscape, Flat; selected Picture Control can be modified; storage for custom Picture Controls
Media	SD (Secure Digital) and UHS-I compliant SDHC and SDXC memory cards
File system	DCF 2.0, Exif 2.31, PictBridge

Movie	
Metering system	TTL exposure metering using main image sensor
Metering mode	Matrix, center-weighted, or highlight-weighted
Frame size (pixels) and frame rate	<ul style="list-style-type: none"> • 3840 × 2160 (4K UHD); 30p (progressive), 25p, 24p • 1920 × 1080; 60p, 50p, 30p, 25p, 24p • 1280 × 720; 60p, 50p <p>Actual frame rates for 60p, 50p, 30p, 25p, and 24p are 59.94, 50, 29.97, 25, and 23.976 fps respectively; ★ high quality available at all frame sizes, normal quality available at all sizes except 3840 × 2160</p>
File format	MOV, MP4
Video compression	H.264/MPEG-4 Advanced Video Coding

Anexo H: Ficha de datos de la cámara BASLER acA640.750um.

Basic Information	
Vendor	Basler
Model	acA640-750um
Type of data presented	Typical
Number of samples	100
Sensor	Python300
Sensor type	CMOS
Sensor diagonal	Diagonal 3.8 mm , Optical Size 1/4 inch
Indication of lens category to be used	C-Mount
Resolution	640 x 480 pixel
Pixel width	4.80 μm
Pixel height	4.80 μm
Readout type	-
Transfer type	-
Shutter type	Global shutter
Overlap capabilities	-
Maximum readout rate	750 frames/second
General conventions	-
Interface type	USB 3

Operating Point		
Item	Symbol	Remarks
Video output format		12 bits/pixel
Gain	Register raw	136
Offset	Register raw	30
Exposure time	T_{exp}	59.0 μs to 4.5 ms

Anexo I: Ficha de datos del acelerómetro ADXL326EB.

ADXL326

SPECIFICATIONS

$T_A = 25^\circ\text{C}$, $V_S = 3\text{ V}$, $C_{L1} = C_{L2} = C_{L3} = 0.1\ \mu\text{F}$, acceleration = 0 g, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT					
Measurement Range	Each axis	± 16	± 19		g
Nonlinearity	Percent of full scale		± 0.3		%
Package Alignment Error			± 1		Degrees
Interaxis Alignment Error			± 0.1		Degrees
Cross Axis Sensitivity ¹			± 1		%
SENSITIVITY (RATIOMETRIC)²					
Sensitivity at X_{OUT} , Y_{OUT} , Z_{OUT}	Each axis $V_S = 3\text{ V}$	51	57	63	mV/g
Sensitivity Change Due to Temperature ³	$V_S = 3\text{ V}$		± 0.01		%/ $^\circ\text{C}$
ZERO g BIAS LEVEL (RATIOMETRIC)					
0 g Voltage at X_{OUT} , Y_{OUT}	$V_S = 3\text{ V}$	1.35	1.5	1.65	V
0 g Voltage at Z_{OUT}	$V_S = 3\text{ V}$	1.2	1.5	1.8	V
0 g Offset vs. Temperature			± 1		mg/ $^\circ\text{C}$
NOISE PERFORMANCE					
Noise Density X_{OUT} , Y_{OUT} , Z_{OUT}			300		$\mu\text{g}/\sqrt{\text{Hz}}$ rms
FREQUENCY RESPONSE⁴					
Bandwidth X_{OUT} , Y_{OUT} ⁵	No external filter		1600		Hz
Bandwidth Z_{OUT} ⁵	No external filter		550		Hz
R_{TOL} Tolerance			$32 \pm 15\%$		k Ω
Sensor Resonant Frequency			5.5		kHz
SELF TEST⁶					
Logic Input Low			+0.6		V
Logic Input High			+2.4		V
ST Actuation Current			+60		μA
Output Change at X_{OUT}	Self test 0 to 1	-29	-62	-114	mV
Output Change at Y_{OUT}	Self test 0 to 1	+29	+62	+114	mV
Output Change at Z_{OUT}	Self test 0 to 1	+29	+105	+190	mV
OUTPUT AMPLIFIER					
Output Swing Low	No load		0.1		V
Output Swing High	No load		2.8		V
POWER SUPPLY					
Operating Voltage Range		1.8		3.6	V
Supply Current	$V_S = 3\text{ V}$		350		μA
Turn-On Time ⁷	No external filter		1		ms
TEMPERATURE					
Operating Temperature Range		-40		+85	$^\circ\text{C}$

ADXL326

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
Acceleration (Any Axis, Unpowered)	10,000 g
Acceleration (Any Axis, Powered)	10,000 g
V_S	-0.3 V to +3.6 V
All Other Pins	(COM - 0.3 V) to ($V_S + 0.3\text{ V}$)
Output Short-Circuit Duration (Any Pin to Common)	Indefinite
Temperature Range (Powered)	-55 $^\circ\text{C}$ to +125 $^\circ\text{C}$
Temperature Range (Storage)	-65 $^\circ\text{C}$ to +150 $^\circ\text{C}$

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ESD CAUTION

NI myDAQ Specifications

Definitions

Warranted specifications describe the performance of a model under stated operating conditions and are covered by the model warranty.

Characteristics describe values that are relevant to the use of the model under stated operating conditions but are not covered by the model warranty.

- **Typical** specifications describe the performance met by a majority of models.
- **Nominal** specifications describe an attribute that is based on design, conformance testing, or supplemental testing.

Specifications are **Typical** unless otherwise noted.

Conditions

Specifications are valid at 25 °C unless otherwise noted.

Analog Input

Number of channels	2 differential or 1 stereo audio input
ADC resolution	16 bits
Maximum sampling rate	200 kS/s
Timing accuracy	100 ppm of sample rate
Timing resolution	10 ns

Range	
Analog input	± 10 V, ± 2 V, DC-coupled
Audio input	± 2 V, AC-coupled
Passband (-3 dB)	
Analog input	DC to 400 kHz
Audio input	1.5 Hz to 400 kHz
Connector type	
Analog input	Screw terminals
Audio input	3.5 mm stereo jack
Input type (audio input)	Line-in or microphone
Microphone excitation (audio input)	5.25 V through 10 k Ω

Analog Output

Number of channels	2 ground-referenced or 1 stereo audio output
DAC resolution	16 bits
Maximum update rate	200 kS/s
Range	
Analog output	± 10 V, ± 2 V, DC-coupled

Maximum output current (analog output)	2 mA
Output impedance	
Analog output	1 Ω
Audio output	120 Ω
Minimum load impedance (audio output)	8 Ω
Connector type	
Analog output	Screw terminals
Audio output	3.5 mm stereo jack
AC-coupling high-pass frequency (audio output with 32 Ω load)	48 Hz

Digital I/O

Number of lines	8; DIO <0..7>
Direction control	Each line individually programmable as input or output
Update mode	Software-timed
Pull-down resistor	75 k Ω
Logic level	5 V compatible LVTTTL input; 3.3 V LVTTTL output
V _{IH} min	2.0 V
V _{IL} max	0.8 V
Maximum output current per line	4 mA

Anexo K: Código del procesamiento en Python.

#Procesamiento final

```
def rnc(imagenlabview):

    import tensorflow as tf
    from tensorflow.keras.preprocessing.image import ImageDataGenerator
    from tensorflow.keras.optimizers import RMSprop
    from tensorflow.keras.preprocessing import image
    from tensorflow import keras
    from tensorflow.keras import layers
    import matplotlib.pyplot as plt
    import matplotlib.image as mpimg
    import os
    import cv2
    import numpy as np
    from ultralytics import YOLO

    _, nombre_clase_predicha, imagenrecortada=recorte(imagenlabview)
    plt.switch_backend('Agg')
    if nombre_clase_predicha=="patrones":
        ruta_actual = os.path.abspath(__file__)
        directorio_actual = os.path.dirname(ruta_actual)
        ruta_modelo_clasificador = os.path.join(directorio_actual,
'modelfinal93validacion038error')
        modelo = tf.keras.models.load_model(ruta_modelo_clasificador)
        ruta_actual = os.path.abspath(__file__)
        directorio_actual = os.path.dirname(ruta_actual)
        ruta_clases = os.path.join(directorio_actual, 'Clases')
        val_ds = tf.keras.utils.image_dataset_from_directory(ruta_clases)
        class_names = val_ds.class_names
        input_arr = tf.keras.preprocessing.image.img_to_array(imagenrecortada)
        input_arr = np.expand_dims(input_arr, axis=0)
        predictions = modelo.predict(input_arr)
        score = tf.nn.softmax(predictions[0])
        salidalabview= "Esta imagen parece ser {} con un {:.2f} % de
exactitud.".format(class_names[np.argmax(score)], 100 * np.max(score))
        print(salidalabview)
        clase_patrones= "{}".format(class_names[np.argmax(score)])
        respuestas_anillos = [
"40Hz a 0.9G en 7ml", "40Hz a 0.97 G en 7ml", "40Hz a 1G en 5ml", "40Hz a 1G en 6ml",
"60Hz a 0.8G en 7ml", "60Hz a 0.95 G en 7ml", "60Hz a 1G en 5ml", "60Hz a 1G en 6ml",
"70Hz a 0.9-1 G entre 5 a 7 ml", "70Hz a 0.95 G en 7ml", "80Hz a 0.99 G en 7ml",
"100Hz a 0.95 G en 7ml", "Entre 80 a 100Hz a 0.9-1 G entre 5 a 7ml"]
        if clase_patrones in respuestas_anillos:
            respuesta_clasificacion = "Anillos"
        else:
            respuesta_clasificacion = "Diamantes o inestabilidad"
    elif nombre_clase_predicha=="2D":
```

```

        salidalabview="Esta imagen parece ser una estrucutra en 2D"
        respuesta_clasificacion = nombre_clase_predicha
    else:
        salidalabview="Esta imagen no parece corresponder a ninguna estructura"
        respuesta_clasificacion = "Desconocida"
    return salidalabview,respuesta_clasificacion
def recorte(imagenlabview): #funcion de recorte solo patron
    from ultralytics import YOLO
    import cv2
    import numpy as np
    import os
    ruta_actual = os.path.abspath(__file__)
    directorio_actual = os.path.dirname(ruta_actual)
    ruta_modelo_yolov8 = os.path.join(directorio_actual, 'bestfinalcon2D.pt')
    model= YOLO(ruta_modelo_yolov8)
    resultado= model.predict(imagenlabview,show=False,save=False, conf=0.5)
    cuadrado=resultado[0].boxes.xyxy
    print(cuadrado)
    try:
        clase_predicha = int(resultado[0].boxes.cls[0])
    except IndexError:
        clase_predicha = -1
    nombres_clases = {
        0: "patrones",
        1: "2D"
    }
    nombre_clase_predicha = nombres_clases.get(clase_predicha, "Desconocida")
    if nombre_clase_predicha == "Desconocida":
        imagen = cv2.imread(imagenlabview)
        imagen_redimensionada = cv2.resize(imagen, (400, 400))
        imagen_redimensionada2 = imagen_redimensionada
    else:
        coordenadas = resultado[0].boxes.xyxy[0]
        x_min, y_min, x_max, y_max = coordenadas
        imagen = cv2.imread(imagenlabview)
        imagen_recortada = imagen[int(y_min):int(y_max), int(x_min):int(x_max)]
        imagen_redimensionada = cv2.resize(imagen_recortada, (300, 300))
        imagen_envio = cv2.resize(imagen_recortada, (600, 600))
    height, width, _ = imagen_envio.shape
    if nombre_clase_predicha == "patrones":
        mascara = np.zeros_like(imagen_envio)
        radio = min(height, width) // 2
        centro = (height // 2, width // 2)
        cv2.circle(mascara, centro, radio, (255, 255, 255), -1)
        imagen_envio = cv2.bitwise_and(imagen_envio, mascara)
    u32_array = np.zeros((height, width), dtype=np.uint32)
    for y in range(height):
        for x in range(width):

```

```

        b, g, r = imagen_envio[y, x]
        pixel_value = (r << 16) | (g << 8) | b
        u32_array[y, x] = pixel_value
    return u32_array,nombre_clase_predicha,imagen_redimensionada
def cartesian_to_polar(image):
    import cv2
    import numpy as np
    height, width = image.shape[:2]
    theta, rho = np.meshgrid(np.linspace(0, 2*np.pi, width), np.arange(height))
    x = rho * np.cos(theta) + width//2
    y = rho * np.sin(theta) + height//2
    polar_image = cv2.remap(image, x.astype(np.float32), y.astype(np.float32),
interpolation=cv2.INTER_LINEAR)
    return polar_image
def procesamintoanillos(imagen):
    import cv2
    import numpy as np
    from scipy.signal import convolve
    from scipy.signal import find_peaks
    import matplotlib.pyplot as plt
    import os
    imagen = cv2.imread(imagen)
    imagen = cv2.resize(imagen, (600, 600))
    imagengris = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)
    clip = 4.0
    tile = 8
    clahe = cv2.createCLAHE(clipLimit=clip, tileGridSize=(tile, tile))
    image = clahe.apply(imagengris)
    polar_image = cartesian_to_polar(image)
    ruta_actual = os.path.abspath(__file__)
    directorio_actual = os.path.dirname(ruta_actual)
    ruta_carpeta = os.path.join(directorio_actual, "temp")
    ruta_guardado = os.path.join(ruta_carpeta, 'polarimage.png')
    cv2.imwrite(ruta_guardado, polar_image)
    limite_superior = int(polar_image.shape[0] * 0.4)
    intensidades_eje_y = np.mean(polar_image[:limite_superior, :], axis=1)
    distancia_pixeles_y = np.arange(limite_superior)
    tamaño_ventana = 10 # Tamaño de la ventana del filtro
    filtro = np.ones(tamaño_ventana) / tamaño_ventana
    intensidades_eje_y_suavizado = convolve(intensidades_eje_y, filtro, mode='same')
    picos, _ = find_peaks(intensidades_eje_y_suavizado, distance=5)
    valles, _ = find_peaks(-intensidades_eje_y_suavizado, distance=5)
    numero_picos= len(picos)
    distancia_anillos = np.diff(picos)
    filtered_values = distancia_anillos[~mad_based_outlier(distancia_anillos)]
    distancia_promedio_anillos = np.mean(filtered_values)
    ruta_actual = os.path.abspath(__file__)
    directorio_actual = os.path.dirname(ruta_actual)

```

```

nombre_carpeta = "temp"
ruta_carpeta = os.path.join(directorio_actual, nombre_carpeta)
if not os.path.exists(ruta_carpeta):
    os.makedirs(ruta_carpeta)
plt.plot(distancia_pixeles_y, intensidades_eje_y_suavizado, '-')
    plt.plot(distancia_pixeles_y[picos], intensidades_eje_y_suavizado[picos], 'ro',
label='Picos')
    plt.plot(distancia_pixeles_y[valles], intensidades_eje_y_suavizado[valles], 'go',
label='Valles')
plt.xlabel('Distancia en píxeles')
plt.ylabel('Intensidad promedio')
plt.legend()
plt.grid(True)
ruta_guardar_imagen = os.path.join(ruta_carpeta, 'graficadeanillos.png')
plt.savefig(ruta_guardar_imagen, dpi=300, bbox_inches='tight', pad_inches=0)
image = cv2.imread(ruta_guardar_imagen)
height, width, _ = image.shape
u32_array = np.zeros((height, width), dtype=np.uint32)

for y in range(height):
    for x in range(width):
        b, g, r = image[y, x]
        pixel_value = (r << 16) | (g << 8) | b
        u32_array[y, x] = pixel_value
return u32_array, distancia_promedio_anillos, filtered_values, numero_picos
def mad_based_outlier(points, thresh=3):
import numpy as np
median = np.median(points)
mad = np.median(np.abs(points - median))
modified_z_score = 0.6745 * (points - median) / mad
return np.abs(modified_z_score) > thresh
def procesamientodiamantes(vector):
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks
import os
import cv2
eje_x = np.arange(len(vector))
picos, _ = find_peaks(vector, distance=5)
valles, _ = find_peaks(-1*vector, distance=5)
distancia_diamantes = np.diff(picos)
filtered_values = distancia_diamantes[~mad_based_outlier(distancia_diamantes)]
distancia_promedio_diamantes = np.mean(filtered_values)
directorio_actual = os.path.dirname(ruta_actual)
nombre_carpeta = "temp"
ruta_carpeta = os.path.join(directorio_actual, nombre_carpeta)
if not os.path.exists(ruta_carpeta):
    os.makedirs(ruta_carpeta)

```

```

plt.plot(eje_x, vector, label='Datos')
plt.plot(eje_x[picos], np.array(vector)[picos], 'ro', label='Picos')
plt.plot(eje_x[valles], np.array(vector)[valles], 'go', label='Valles')
plt.legend()
plt.xlabel('Posición en el Vector')
plt.ylabel('Intensidad')
plt.title('Intensidad de pixeles')
ruta_guardar_imagen = os.path.join(ruta_carpeta, 'grafidiamantes.png')
plt.savefig(ruta_guardar_imagen, bbox_inches='tight', pad_inches=0)
image = cv2.imread(ruta_guardar_imagen)
height, width, _ = image.shape
u32_array = np.zeros((height, width), dtype=np.uint32)
for y in range(height):
    for x in range(width):
        b, g, r = image[y, x]
        pixel_value = (r << 16) | (g << 8) | b
        u32_array[y, x] = pixel_value
return u32_array, distancia_promedio_diamantes
def procesamiento2D(rutaimagen, tipoestructura):
    import cv2
    import numpy as np
    from matplotlib import pyplot as plt
    from scipy.signal import find_peaks
    import os
    if tipoestructura== "2D":
        imagen = cv2.imread(rutaimagen, cv2.IMREAD_GRAYSCALE)
        th3 = cv2.adaptiveThreshold(imagen, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY, 55, 4)
        filtered_image = cv2.medianBlur(th3, 7)
        edges = cv2.Canny(filtered_image, threshold1=30, threshold2=100)
        contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        min_contour_length = int(0.9 * edges.shape[1])
        filtered_contours = [contour for contour in contours if cv2.arcLength(contour, True) >
min_contour_length]
        sorted_contours = sorted(filtered_contours, key=lambda x: cv2.minEnclosingCircle(x)[0][1])
        contour1 = sorted_contours[0]
        contour3 = sorted_contours[2]
        contour1_array = np.array([contour1[:, 0, 0], contour1[:, 0, 1]]).T
        contour3_array = np.array([contour3[:, 0, 0], contour3[:, 0, 1]]).T
        peaks1, _ = find_peaks(contour1_array[:, 1])
        valleys1, _ = find_peaks(-contour1_array[:, 1])
        peaks3, _ = find_peaks(contour3_array[:, 1])
        valleys3, _ = find_peaks(-contour3_array[:, 1])
        contour1_peaks_x = contour1_array[peaks1, 0]
        contour1_peaks_y = contour1_array[peaks1, 1]
        contour1_valleys_x = contour1_array[valleys1, 0]
        contour1_valleys_y = contour1_array[valleys1, 1]
        contour3_peaks_x = contour3_array[peaks3, 0]

```

```

contour3_peaks_y = contour3_array[peaks3, 1]
contour3_valleys_x = contour3_array[valleys3, 0]
contour3_valleys_y = contour3_array[valleys3, 1]
contour1_peaks_sorted_idx = np.argsort(contour1_peaks_x)
contour1_valleys_sorted_idx = np.argsort(contour1_valleys_x)
contour3_peaks_sorted_idx = np.argsort(contour3_peaks_x)
contour3_valleys_sorted_idx = np.argsort(contour3_valleys_x)
contour1_peaks_x = contour1_peaks_x[contour1_peaks_sorted_idx]
contour1_peaks_y = contour1_peaks_y[contour1_peaks_sorted_idx]
contour1_valleys_x = contour1_valleys_x[contour1_valleys_sorted_idx]
contour1_valleys_y = contour1_valleys_y[contour1_valleys_sorted_idx]
contour3_peaks_x = contour3_peaks_x[contour3_peaks_sorted_idx]
contour3_peaks_y = contour3_peaks_y[contour3_peaks_sorted_idx]
contour3_valleys_x = contour3_valleys_x[contour3_valleys_sorted_idx]
contour3_valleys_y = contour3_valleys_y[contour3_valleys_sorted_idx]
def process_peaks_valleys(x_values, y_values, threshold_distance):
    new_x = []
    new_y = []
    i = 0
    while i < len(x_values):
        current_x = x_values[i]
        current_y = y_values[i]
        sum_x = current_x
        count = 1
        j = i + 1
        while j < len(x_values) and abs(x_values[j] - current_x) < threshold_distance and
y_values[j] == current_y:
            sum_x += x_values[j]
            count += 1
            j += 1
        new_x.append(sum_x / count)
        new_y.append(current_y)
        i = j
    return new_x, new_y
threshold_distance = 50
contour1_peaks_x, contour1_peaks_y = process_peaks_valleys(contour1_peaks_x,
contour1_peaks_y, threshold_distance)
contour1_valleys_x, contour1_valleys_y = process_peaks_valleys(contour1_valleys_x,
contour1_valleys_y, threshold_distance)
contour3_peaks_x, contour3_peaks_y = process_peaks_valleys(contour3_peaks_x,
contour3_peaks_y, threshold_distance)
contour3_valleys_x, contour3_valleys_y = process_peaks_valleys(contour3_valleys_x,
contour3_valleys_y, threshold_distance)
def filter_values(x_values, y_values, y_threshold):
    filtered_x = []
    filtered_y = []
    median_y = np.median(y_values)
    y_deviation = np.std(y_values)

```

```

        for x, y in zip(x_values, y_values):
            if abs(y - median_y) <= y_deviation * y_threshold:
                filtered_x.append(x)
                filtered_y.append(y)
        return filtered_x, filtered_y
y_threshold = 2
        contour1_valleys_x, contour1_valleys_y = filter_values(contour1_valleys_x,
contour1_valleys_y, y_threshold)
        contour1_peaks_array = np.column_stack((contour1_peaks_x, contour1_peaks_y))
        contour1_valleys_array = np.column_stack((contour1_valleys_x, contour1_valleys_y))
        contour3_peaks_array = np.column_stack((contour3_peaks_x, contour3_peaks_y))
        contour3_valleys_array = np.column_stack((contour3_valleys_x, contour3_valleys_y))
def calculate_average_height(y_values):
    average_height = np.mean(y_values)
    return average_height
contour1_peaks_average_height = calculate_average_height(contour1_peaks_y)
contour1_valleys_average_height = calculate_average_height(contour1_valleys_y)
contour3_peaks_average_height = calculate_average_height(contour3_peaks_y)
contour3_valleys_average_height = calculate_average_height(contour3_valleys_y)
        contour1_average_distance_y = contour1_peaks_average_height -
contour1_valleys_average_height
        contour3_average_distance_y = contour3_peaks_average_height -
contour3_valleys_average_height
        contour1_average_valleys_distance_x = np.mean(np.diff(contour1_valleys_x))
        contour1_average_peaks_distance_x = np.mean(np.diff(contour1_peaks_x))
        contour3_average_valleys_distance_x = np.mean(np.diff(contour3_valleys_x))
        contour3_average_peaks_distance_x = np.mean(np.diff(contour3_peaks_x))
        contour1_average_distance_x= (contour1_average_valleys_distance_x+
contour1_average_peaks_distance_x)/2
        contour3_average_distance_x= (contour3_average_valleys_distance_x+
contour3_average_peaks_distance_x)/2
        ruta_actual = os.path.abspath(__file__)
        directorio_actual = os.path.dirname(ruta_actual)
        nombre_carpeta = "temp"
        ruta_carpeta = os.path.join(directorio_actual, nombre_carpeta)
        if not os.path.exists(ruta_carpeta):
            os.makedirs(ruta_carpeta)
        plt.figure()
        plt.figure(figsize=(10, 6))
        plt.plot(contour1_array[:, 0], contour1_array[:, 1], label='Superficie de líquido')
        plt.plot(contour3_array[:, 0], contour3_array[:, 1], label='Superficie de partículas')
        plt.plot(contour1_peaks_x, contour1_peaks_y, 'o', label='Valles liquido',markersize=8,
linewidth=2)
        plt.plot(contour1_valleys_x, contour1_valleys_y, 'x', label='Picos liquido',markersize=8,
linewidth=2)
        plt.plot(contour3_peaks_x, contour3_peaks_y, 'o', label='Valles particulas',markersize=8,
linewidth=2)

```

```

        plt.plot(contour3_valleys_x, contour3_valleys_y, 'x', label='Picos
particulas',markersize=8, linewidth=2)
    plt.gca().invert_yaxis()
    plt.xlabel('Coordenada X')
    plt.ylabel('Coordenada Y')
    plt.legend()
    ruta_guardar_imagen = os.path.join(ruta_carpeta, 'grafico2d.png')
    plt.savefig(ruta_guardar_imagen, bbox_inches='tight', pad_inches=0)
RGB
    image = cv2.imread(ruta_guardar_imagen)
    height, width, _ = image.shape
    u32_array = np.zeros((height, width), dtype=np.uint32)
    for y in range(height):
        for x in range(width):
            b, g, r = image[y, x]
            pixel_value = (r << 16) | (g << 8) | b
            u32_array[y, x] = pixel_value
    return
(u32_array, contour1_peaks_array, contour1_valleys_array, contour3_peaks_array, contour3_valleys_arr
ay,
        contour1_average_distance_y, contour3_average_distance_y, contour1_average_distance_x,
contour3_average_distance_x)
def cicloonda(ruta_carpeta):
    import cv2
    import numpy as np
    from matplotlib import pyplot as plt
    from scipy.signal import find_peaks
    import os
    if not ruta_carpeta == '':
        nombres_archivos = os.listdir(ruta_carpeta)
        colores = ['b', 'g', 'r', 'c', 'm', 'y', 'k', 'orange', 'purple', 'brown', 'pink']
        plt.figure(figsize=(12, 8))

        for i, nombre_archivo in enumerate(nombres_archivos):
            rutaimagen = os.path.join(ruta_carpeta, nombre_archivo)
            imagen = cv2.imread(rutaimagen, cv2.IMREAD_GRAYSCALE)
            th3 = cv2.adaptiveThreshold(imagen, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY, 55, 4)
            filtered_image = cv2.medianBlur(th3, 7)
            edges = cv2.Canny(filtered_image, threshold1=30, threshold2=100)
            contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
            min_contour_length = int(0.9 * edges.shape[1])
            filtered_contours = [contour for contour in contours if cv2.arcLength(contour, True)
> min_contour_length]
                sorted_contours = sorted(filtered_contours, key=lambda x:
cv2.minEnclosingCircle(x)[0][1])
            contour1 = sorted_contours[0]
            contour3 = sorted_contours[2]

```

```
        color = colores[i % len(colores)]
        plt.plot(contour1[:, :, 0], contour1[:, :, 1], color + '--', label=f'Onda -
{nombre_archivo}')
        plt.plot(contour3[:, :, 0], contour3[:, :, 1], color + '--')
plt.gca().invert_yaxis()
plt.title('Contornos 1 y 3 con Nuevos Picos y Valles')
plt.xlabel('Coordenada X')
plt.ylabel('Coordenada Y')
plt.legend()
ruta_actual = os.path.abspath(__file__)
directorio_actual = os.path.dirname(ruta_actual)
nombre_carpeta = "temp"
ruta_carpeta = os.path.join(directorio_actual, nombre_carpeta)
if not os.path.exists(ruta_carpeta):
    os.makedirs(ruta_carpeta)
ruta_guardar_imagen = os.path.join(ruta_carpeta, 'cicloondas.png')
plt.savefig(ruta_guardar_imagen, dpi=300, bbox_inches='tight', pad_inches=0)
return
```



ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO

DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL
APRENDIZAJE



UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 22/ 11 / 2023

INFORMACIÓN DE LOS AUTORES	
Nombres – Apellidos: Joel Alexander Argos Quishpe - Luis Israel Granizo Mora	
INFORMACIÓN INSTITUCIONAL	
Facultad: Informática y Electrónica	
Carrera: Electrónica y Automatización	
Título a optar: Ingeniero en electrónica y automatización	
f. Analista de Biblioteca responsable:	 Ing. Fernanda Arévalo M.



1837-DBRA-UPT-2023