



ESCUELA POLITÉCNICA NACIONAL



FACULTAD DE INGENIERÍA MECÁNICA

“DESARROLLO DEL CÓDIGO DE PROGRAMACIÓN PARA PROCESAMIENTO DE IMÁGENES APLICADA EN FUNDICIONES NODULARES”

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO MECÁNICO

ANÍBAL SEBASTIÁN SILVA SÁNCHEZ

anibal.silva@epn.edu.ec

DIRECTOR: ING. VICTOR HUGO HIDALGO, D.Sc.

victor.hidalgo@epn.edu.ec

CODIRECTOR: ING. CARLOS WIME DÍAZ, M.Sc.

carlos.diaz@epn.edu.ec

Junio, 2017

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por el señor **Aníbal Sebastián Silva Sánchez**, bajo mi supervisión.

Ing. Víctor Hugo Hidalgo, DSc.

DIRECTOR DE PROYECTO

Ing. Carlos Wime Díaz, MSc.

CODIRECTOR DE PROYECTO

DECLARACIÓN

Yo, **Aníbal Sebastián Silva Sánchez**, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

Aníbal Sebastián Silva Sánchez

DEDICATORIA

Este trabajo está dedicado a mis padres y mis abuelos a quienes les dedico todo el esfuerzo realizado durante toda mi carrera

AGRADECIMIENTO

Llegado a este punto de la vida uno mira hacia atrás y piensa en tantas cosas. Por fortuna, llegado el final de mi vida estudiantil, al voltear la vista solo puedo sonreír por tantos gratos momentos, lo que me hace pensar que todo este tiempo ha valido toda la pena del mundo.

En primer lugar quiero agradecer a la vida, por mostrarme este camino, a mis padres por darme la entereza, la decisión y las ganas de salir adelante. A mi hermana por su apoyo, su tiempo para escucharme con su cariño y la por la paciencia que me ha tenido.

Agradezco a toda mi familia, a mis primos y mis tíos, por el apoyo y las palmadas en la espalda deseándome que todo me vaya bien. Principalmente agradezco a Richard, Mauricio y Ramiro; gracias por sus consejos y su cariño desde que soy un niño, pues han sido para mí como los hermanos mayores que nunca tuve.

Gracias especialmente a Richard por las largas noches revisando el código de este proyecto conmigo, pues soy mecánico y la programación fue un reto que felizmente fue cumplido. Muchas veces el código no funcionaba por errores pequeños que me ayudaste a identificar. Muchas gracias tío por todo.

No puedo continuar con este agradecimiento sin mencionar a mis abuelos José, Mercedes y María. Solo puedo decir un inmenso, gracias por todo, desde cambiarme cuando era un bebé, recogerme de la escuela, hasta al final de este camino estudiantil, estar siempre pendientes de mí y desearme lo mejor.

Estaré eternamente agradecido por tantas cosas vividas en mi querida facultad, la Mejor Facultad del Mundo !!!!. Todas las malas noches, los largos informes, las tardes en la biblioteca estudiando en grupo; todo se ve tan distante y finalmente solo queda un sentimiento de paz y gratitud.

Agradezco profundamente a mi director Víctor Hugo Hidalgo por los consejos, las charlas y la guía en este proyecto, de igual manera agradezco a mi codirector Carlos Díaz por el asesoramiento durante todo este proyecto.

Gracias nuevamente a la Facultad de Ingeniería Mecánica, por permitirme conocer tantas personas fuera de serie, fue un honor haber compartido con todos estos ya casi seis años de mi vida; especialmente a los panas del 206, espero contar siempre con su amistad.

Sin más que decir y con el corazón en la mano, pues siempre es difícil decir adiós, me despido de mi querida facultad con un estruendoso: ¡¡¡ TRES RASES !!! ¡¡¡ UN CHISPUN Y UN CARAJOS POR MECANICA!!!!

ÍNDICE

Resumen.....	viii
Abstract.....	xi
INTRODUCCIÓN	1
Objetivo General	2
Objetivos Específicos	2
1. MARCO TEÓRICO	3
1.1. Aleaciones Ferrosas.....	3
1.1.1. Estructuras presentes en el Diagrama de Equilibrio Hierro-Fe ₃ C.....	3
1.2. Hierros Fundidos.....	4
1.2.1. Principios de la metalurgia en las aleaciones de hierro Fundido.....	5
1.3. Fundiciones Nodulares.....	10
2. METODOLOGÍA	19
2.1. Software Libre.....	19
2.1.1. Python y Open CV.....	19
2.2. Procesamiento de Imágenes.....	20
2.2.1. Escala de Grises.....	20
2.2.2. Discriminación matricial.....	20
2.2.3. Análisis de Perlita, Ferrita y Grafito.....	21
2.2.4. Análisis de Nodularidad.....	24
2.2.5. Tamaño y Conteo de Nódulos.....	28
2.2.6. Tipo de Grafito.....	29
2.3. Adición de resultados a una base de datos.....	29
2.4. Experimentación.....	30
2.4.1. Obtención de muestras.....	30
2.4.2. Análisis de muestras proporcionado por el laboratorio.....	32
2.5. Interfaz Gráfica.....	34
3. RESULTADOS Y DISCUSIÓN	39
3.1. Resultados.....	39
3.2. Discusión.....	41
3.2.1. Valor de la escala de grises.....	42
3.2.2. Comparación de porcentajes de grafito ferrita y perlita experimentales.....	43
3.2.3. Tipo de Grafito.....	45
3.2.4. Análisis de Nodularidad.....	46
4. CONCLUSIONES	47
Referencias Bibliográficas.....	49

ANEXOS.....	50
ANEXO I. DIAGRAMA DE HIERRO CARBONO.....	48
ANEXO II. RELACIÓN ENTRE EL CONTENIDO DE CARBONO Y SILICIO.....	49
ANEXO III. CODIGO DEL SOFTWARE PLANTEADO.....	50
ANEXO IV. CODIGO DE LA INTERFAZ GRÁFICA.....	80

RESUMEN

En este estudio se plantea la elaboración de un software capaz de realizar un análisis apropiado de metalografías de fundiciones nodulares, aleaciones conocidas también como hierro fundido dúctil. El código fue desarrollado en el lenguaje de programación Python, en conjunto con la librería OpenCV, la cual está destinada para el procesamiento de imágenes. A través del análisis de color de los píxeles de una metalografía en una escala de grises, se diferencia y se obtiene el porcentaje de las fases microestructurales que presenta la aleación. A través de operaciones entre imágenes se aísla y analiza los nódulos de las aleaciones, obteniéndose la cantidad de los mismos en función del área, así como el tamaño promedio, y la relación de cuanto grafito presente en la aleación se concentró en nódulos, propiedad que es conocida como nodularidad. Para completar el análisis el tipo de grafito presente en la aleación tiene que ser identificado. Finalmente los resultados del programa son añadidos a una tabla en la base de datos MySQL. Para la fácil operación del código se desarrolló una interfaz gráfica elaborada en Qt designer, de manera que se puede utilizar como herramienta para el estudio metalografías de las aleaciones de hierro fundido dúctil. A través de la experimentación y la comparación de resultados, se ha obtenido valores acordes con las referencias, lo que demuestra aplicabilidad del software desarrollado.

Palabras Clave: Ferrita, Imágenes, Nódulos, Perlita, Píxeles, Procesamiento

ABSTRACT

In this study it is proposed the development of a software capable of performing an appropriate analysis of nodular cast iron metallographys; that alloys known as ductile cast iron too. The code was developed in the Python programming language, in addition with the OpenCV library, which is using for image processing. Through the color analysis of the pixels at the metallography on a gray scale, the percentage of the microstructural phases of the alloy is differentiated and obtained. Using operations between images, the alloy nodules are isolated and analyzed, obtaining the quantity of them as a function of the area, as well as the average size, and the ratio of the graphite present in the alloy was concentrated in nodules, it is known as nodularity. To complete the analysis the type of graphite present in the alloy has to be identified. Finally the results of the program are added to a table in the MySQL database. For the easy operation of the code a graphical interface was developed in Qt designer, so it can be used as a tool for the study of metallography of ductile cast iron´s alloys. Through experimentation and comparison of results, values have been obtained quite close to the references, demonstrating applicability of the developed software

Keywords: Ferrite, Image, Nodules, Perlite, Pixels, Process

“DESARROLLO DEL CÓDIGO DE PROGRAMACIÓN PARA PROCESAMIENTO DE IMÁGENES APLICADA EN FUNDICIONES NODULARES”

INTRODUCCIÓN

La metalurgia es un campo importante dentro del país, aplicable en varios campos como: líneas petroleras, estructuras metálicas, centrales térmicas, centrales hidráulicas y otras aplicaciones. Lo cual hace creciente la necesidad del desarrollo de herramientas para el análisis rápido y preciso de muestras metalográficas, una respuesta a esta problemática surgió con la primera versión de “procesamiento de imágenes mediante software libre python para el análisis metalográfico en aceros de bajo contenido de carbono”, implementado en el laboratorio de metalografía; esta herramienta tiene la limitante, fue direccionada a aceros de bajo carbono. En el presente proyecto se pretende realizar una herramienta que permita realizar un análisis rápido y preciso a metalografías de fundiciones nodulares, ya que este tipo de aleaciones tienen una gran resistencia al impacto, soportan altas temperaturas, absorben las vibraciones y además presentan una buena resistencia al desgaste; por tal motivo se emplea este tipo de aleaciones en una gran variedad de aplicaciones como en: bombas hidráulicas, bloques de motores, y en general soportes para maquinaria, lo cual hace necesario el análisis de estas aleaciones, en la manufactura.

El análisis metalográfico es una herramienta muy útil en los procesos de manufactura, para poder garantizar la calidad de los materiales empleados en un producto, el software de procesamiento rápido de imágenes para fundiciones nodulares planteado, puede ser implementado en el laboratorio de metalografía, brindando oportunidades de mejorar del aprendizaje de los estudiantes en la facultad, convirtiéndose en un aporte para la docencia, y la investigación al alcance de estudiantes y profesores. Puesto que al emplear software libre es posible distribuir el programa desarrollado con gran facilidad, este puede ser empleado libremente en aplicaciones complementarias en futuros proyectos, relacionados con el procesamiento de imágenes.

Las fundiciones nodulares son aleaciones de hierro y carbono como sus principales aleantes, estas presentan un porcentaje entre el 3 y 4% de contenido de carbono, al agregar una pequeña cantidad de magnesio en su estado líquido el grafito se agrupa en esferoides llamados nódulos, al realizar una metalografía es necesario identificar la microestructuras ferrítica y perlítica, presentes en la aleación, adicionalmente la cantidad

de grafito que se concentró en nódulos, propiedad conocida como nodularidad, así como un análisis de los nódulos observados en la metalografía. En concordancia con la norma ASTM A 427-16 el programa planteado obtiene porcentajes de las microestructuras y de las propiedades descritas.

Objetivo General

Desarrollar un código de programación para el análisis de metalografías de fundiciones, principalmente nodulares.

Objetivos Específicos

- Realizar una metodología de análisis de muestras de fundiciones.
- Dar una herramienta de apoyo para futuras investigaciones en este campo, y además también para la docencia.
- Establecer estadísticas de apoyo que garantice la precisión del software desarrollado.

1. MARCO TEÓRICO

1.1. Aleaciones Ferrosas

El hierro es uno de los metales más altamente empleado desde edades muy antiguas, ventajosamente a través del descubrimiento del acero, al combinar el hierro con carbono, se ha generado un sin número de aleaciones, con diferentes propiedades y aplicaciones a nivel industrial. En el espectro general de las aleaciones ferrosas se incluye: aceros de bajo, medio y alto contenido de carbono, aceros aleados, aceros inoxidable y hierros fundidos. (Smith & Hashemi, 2006)

1.1.1. Principales constituyentes microscópicos de aleaciones ferrosas

Con el conocimiento del comportamiento alotrópico del hierro es necesario analizar el diagrama de equilibrio hierro- carbono, donde se muestra las distintas fases presentes en las aleaciones. En el diagrama se muestra a la mezcla hierro- carbono enfriada muy lentamente a diferentes temperaturas y con diferentes composiciones de carbono hasta llegar a 6,67 por ciento de carbono, como se muestra en el Anexo I. Es necesario recalcar que este diagrama es un diagrama idealizado, debido a que el enfriamiento en la generación de estas aleaciones se realiza en una manera mucho más rápida que la idealizada en el diagrama, pero en general se obtiene constituyentes microscópicos bastante aproximados por lo cual es una buena referencia de trabajo. Dentro del diagrama se tiene los siguientes compuestos: (Smith & Hashemi, 2006)

Ferrita (α): Esta fase es prácticamente hierro en su estado puro, presenta una estructura BCC, el carbono es ligeramente soluble en la ferrita, y alcanza una solubilidad de sólidos máxima de 0.02 % a 723°C y de 0,008 % de Carbono a temperatura ambiente. Es la estructura más suave que aparece en el diagrama. (Smith & Hashemi, 2006)

Austenita (γ): Es una solución sólida intersticial de carbono en hierro (γ). La solución presenta una estructura FCC, a una temperatura entre 723°C y 1300°C. La estructura FCC proporciona una capacidad mayor de albergar al carbono en los intersticios de la estructura cristalina en comparación con la solubilidad del carbono en la estructura BCC. No suele cristalizar a temperatura ambiente, es necesario tener procesos especiales para su obtención a temperatura ambiente. (Smith & Hashemi, 2006); (Avner, 1988)

Cementita (Fe_3C): Es un compuesto intermetálico, el cual presenta una alta dureza, siendo el compuesto más duro formado en este tipo de aleaciones, tiene una estructura

ortorrómbica, por lo cual es muy frágil; la cementita tiene una composición de 6,67% C y 93,3% Fe. (Smith & Hashemi, 2006)

Ferrita (δ): Solución sólida similar a la ferrita con una estructura cúbica de cuerpo centrado, con la diferencia que la red presenta una constante de red mayor, presentando una solubilidad de carbono, más alta que la ferrita, de 0,09 % de carbono. (Smith & Hashemi, 2006)

1.2. Hierros Fundidos

Dentro de las aleaciones Ferrosas se conocen como aceros a las aleaciones que tengan hasta el 2.1% de Carbono en la aleación, mientras que se habla de fundiciones de hierro a las aleaciones que tienen un porcentaje de carbono mayor al 2.1%. En las fundiciones además del carbono se puede apreciar una cantidad considerable de Silicio, de 0.5 a 3 %; este elemento cumple con la función principal de promover la grafitización en la aleación, es decir fomenta un enfriamiento más lento en la aleación colada generándose así la descomposición del carburo de hierro, en grafito y ferrita. (American Society for Metals, 2005); (Donald R. Askeland, 2003)

En general al observar metalografías de aleaciones de hierro fundido se puede identificar una fase rica en grafito, esto se da por la pobre solubilidad del carbono, que se presenta en la fase de hierro alfa que es de 0,02 % a temperatura de su reacción eutectoide. Al tener un porcentaje considerable de carbono y una cantidad adecuada de silicio, se observa claramente al grafito en la aleación, el cual se agrupa en diferentes formas como se observa en la Figura 1.1. (Callister, 1990); (Avner, 1988); (American Society for Metals, 2005)

Tradicionalmente se las clasificó a las fundiciones de hierro por el color presentado por la superficie fracturada, de tal manera que se tiene fundiciones blancas, atruchadas y grises; es el caso de fundiciones que presentan una superficie clara, por la presencia de carburo de hierro, se denominaron fundiciones blancas; por otro lado al presentarse una superficie oscura grisácea, por la presencia de grafito en la aleación, se denominó a esta aleación fundición gris; las fundiciones atruchadas conocidas también como hierro moteado se identifica a estas aleaciones como una mezcla entre los dos tipos mencionados anteriormente. Con el desarrollo de la metalurgia, se han generado distintos tipos de aleaciones a partir del conocimiento y tratamiento de las aleaciones tradicionales, de tal manera en la actualidad se tiene diferentes fundiciones como: fundiciones compactas, maleables, nodulares también conocidas como fundiciones dúctiles. Las aleaciones de

hierro fundido empleadas en la actualidad designadas por American Society for Metals se encuentran resumidas en la Tabla 1.1. (Donald R. Askeland, 2003); (American Society for Metals, 2005)

Tabla 1.1. Clasificación de fundiciones por la designación comercial, microestructura y proceso de producción.

Designación Comercial	Estructura de la fase de Carbono	Matriz	Fractura (a)	Proceso de producción
Hierro Gris	Hojuelas de grafito	P	Gris	Solidificación
Hierro Dúctil	Grafito esferoidal	F, P, A	Plateada-Gris	Solidificación o tratamiento térmico
Hierro Compacto	Grafito vermicular compactado	F, P	Gris	Solidificación
Hierro Blanco	Fe ₃ C	P, M	Blanca	Solidificación y tratamiento térmico (b)
Hierro Moteado	Grafito Laminar + Fe ₃ C	P	Moteada	Solidificación
Hierro Maleable	Grafito templado	F, P	Plateada-Gris	Tratamiento térmico
Hierro Dúctil Austemperado	Grafito esferoidal	At	Plateada-Gris	Tratamiento térmico

(a) F, ferrita; P, perlita, A austenita; M, martensita; At, austemperado (bainita). (b) Las aleaciones de Hierro blanco no suele necesitar tratamiento térmico, excepto para alivio de esfuerzos y para la transformación austenítica. (American Society for Metals, 2005)

1.2.1. Principios metalúrgicos en las aleaciones de hierro fundido

Desde el punto vista de metalúrgico, se busca generar una aleación que cumpla con los requerimientos de las aplicaciones en las cuales va a ser empleada. En el caso de las fundiciones, que se expone en la Tabla 1.1, se tiene estructuras de grafito correspondientes a diferentes tipos de hierro fundido, los cuales tiene propiedades diferentes, es necesario un proceso diferente de producción para poder obtener cada uno de estos, para lo cual se debe tener en cuenta los parámetros enumerados a continuación,

conocidos como principios metalúrgicos. (American Society for Metals, 2005); (Donald R. Askeland, 2003)

1. Composición química
2. Velocidad de enfriamiento
3. Tratamiento del metal líquido
4. Tratamiento térmico

En el caso de la composición química, dependiendo de los materiales que se añada, se fomenta el potencial de grafitización, es decir se genera la descomposición de la cementita (Fe_3C) en hierro (Fe) y carbono (Grafito), los materiales que fomentan la grafitización son: Silicio, Titanio, Fósforo, Aluminio, Cobre y Níquel. También se tiene materiales los cuales estabilizan y fomentan la formación de cementita, estos son: Manganeso Cromo, Molibdeno y Vanadio. Por lo cual en la formación de la mezcla líquida se debe añadir, o depurar estos elementos cuidadosamente. En la Tabla 1.2 se puede apreciar las composiciones de diferentes hierros fundidos. (Avner, 1988); (American Society for Metals, 2005)

Tabla 1.2. Composición de los diferentes tipos de hierros fundidos.

Tipo de Fundición	Composición %				
	C	Si	Mn	P	S
Gris	2,5- 4	1 - 3	0,2 - 1	0,002 - 1	0,02 - 0,25
Compacta	2,5- 5	1 - 3	0,2 - 1	0,01 - 0,1	0,01 - 0,03
Dúctil (Nodular)	3 - 4	1,8 - 2,8	0,1 - 1	0,01 - 0,1	0,01 - 0,03
Blanca	1,8 - 3,6	0,5 - 1,9	0,25 - 0,8	0,06 - 0,2	0,06 - 0,2
Maleable	2,2 - 2,9	0,9 - 1,9	0,15 - 1,2	0,02 - 0,2	0,02 - 0,2

(American Society for Metals, 2005)

El elemento más empleado para promover la grafitización, es el Silicio, el cual es uno de los elementos en las aleaciones de hierro fundido que más influencia en las propiedades. Al tener un porcentaje bajo de silicio no se facilita el proceso de grafitización y se obtienen aleaciones frágiles por el alto contenido de cementita. Por otro lado al tener una cantidad muy elevada de silicio obtenemos una grafitización excesiva, se tiene una cantidad muy elevada grafito esparcido en la matriz lo cual tampoco genera buenas propiedades en la

aleación, porque se tiene concentradores de esfuerzos esparcidos por toda la aleación. (American Society for Metals, 2005)

A una composición de 4,3% de Carbono en el diagrama de equilibrio, se tiene una reacción eutéctica, es decir que al tener esa composición, desde una mezcla líquida de hierro y carbono, se genera una solución sólida homogénea de dos fases, austenita y cementita. A la solución sólida obtenida se la conoce con el nombre de ledeburita. En general los compuestos eutécticos generan una mezcla muy fina de los elementos presentes en la aleación. Al solidificar los compuestos eutécticos tienen un comportamiento similar a un material puro, lo que genera una mejor distribución de los aleantes. La austenita no es estable a temperatura ambiente; por lo cual al enfriarse la solución por debajo de la temperatura eutectoide, la austenita se descompone en ferrita y perlita distribuidas en la estructura de la aleación en los lugares donde se encontraban los granos formados. La matriz microestructural obtenida puede ser predominantemente ferrítica, perlítica o una combinación de ambas. La matriz es uno de los factores que determina las propiedades mecánicas de la aleación. Al considerar los efectos de otros aleantes, se observa que se tiene diversas consecuencias en la aleación, como es el caso de la grafitización, mencionada con anterioridad, o la constitución de estructuras diferentes en la fase de grafito, característica de las fundiciones. (American Society for Metals, 2005); (Donald R. Askeland, 2003); (Avner, 1988)

Se debe tener en cuenta que la adición de los diferentes aleantes, genera que el punto eutéctico del diagrama de equilibrio se desplace hacia la izquierda, por lo cual se usa el término de carbono equivalente, como se observa en la Ecuación 1.1, para comparar el comportamiento de las diferentes aleaciones de hierro fundido mostradas en la Tabla 1.2, con el comportamiento de una aleación de hierro carbono descrita en el diagrama de equilibrio. (American Society for Metals, 2005)

$$CE = \%C + 0.3(\%Si) + 0.33(\%P) - 0.027(\%Mn) + 0.4(\%S)$$

Ecuación 1.1. Carbono equivalente.

(American Society for Metals, 2005)

La matriz generada depende de la composición, de la razón de enfriamiento, y del tratamiento térmico aplicado, pues de esta manera se puede obtener matrices ferríticas,

perlíticas, austeníticas o martensíticas, dependiendo de la estructura y las propiedades buscadas. (Donald R. Askeland, 2003)

Dependiendo del proceso de manufactura empleado para la obtención de las aleaciones de hierro fundido es posible obtener diferentes estructuras de grafito, las cuales se indican en la Figuras 1.1. Las diferentes fundiciones se caracterizan por el tipo de grafito que posean, esto se indica en la Figura 1.2. (ASTM, A247-16, 2016); (Donald R. Askeland, 2003)

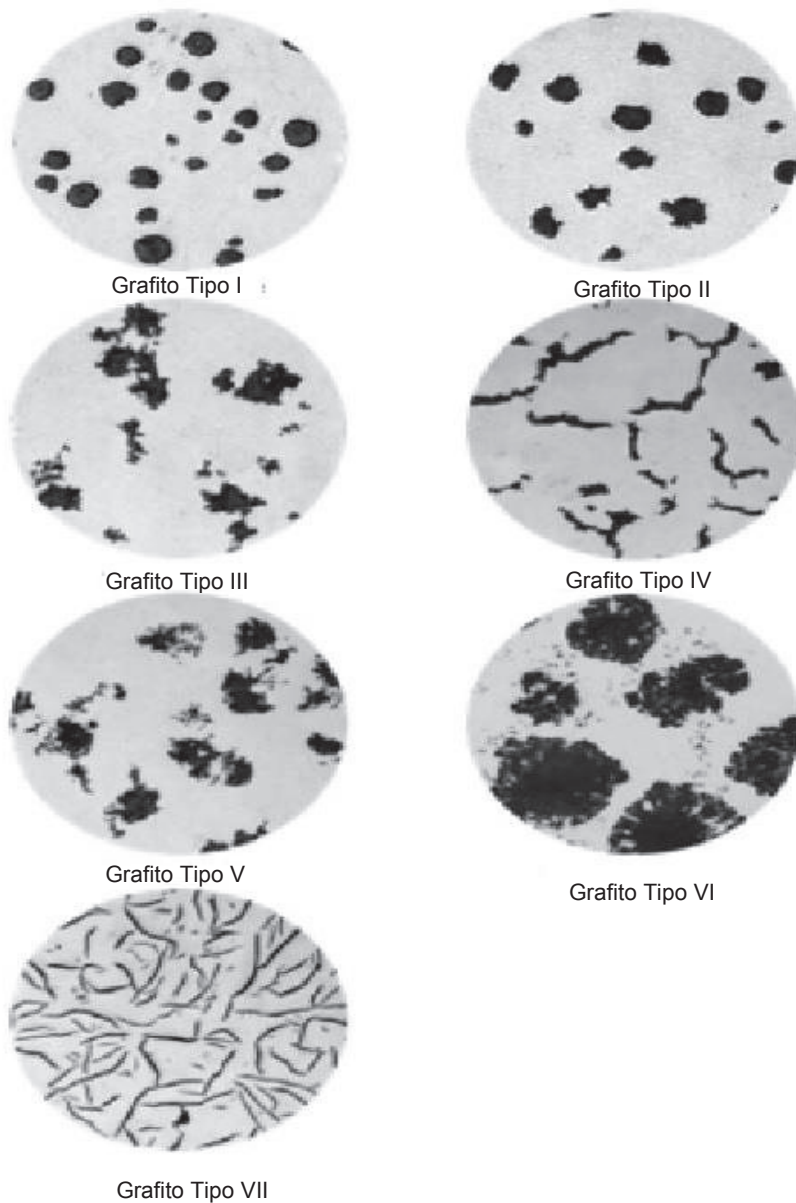


Figura 1.1. Estructuras de grafito en hierros fundidos.

Norma: (ASTM, A247-16, 2016)

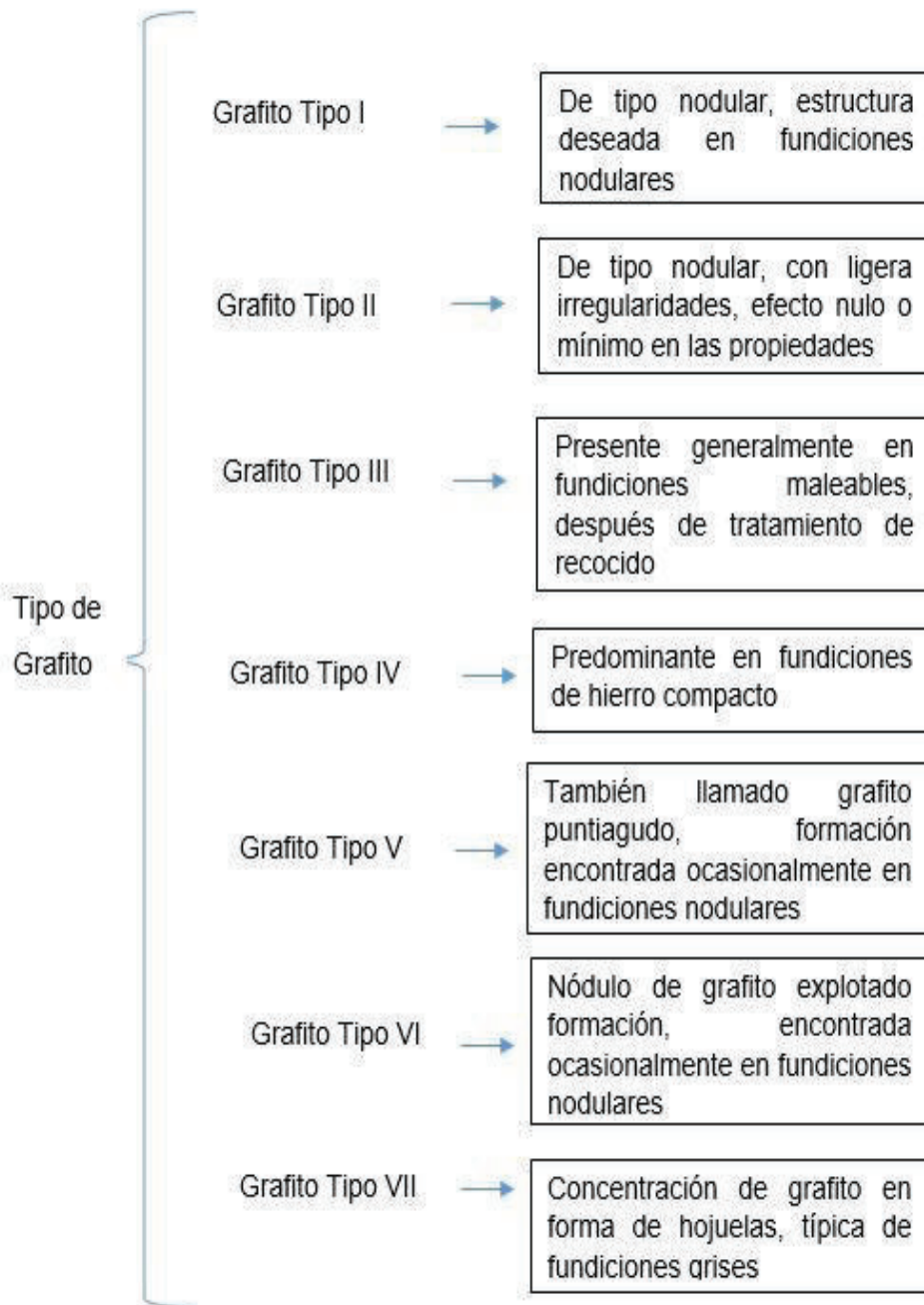


Figura 1.2. Resumen tipo de grafito en fundiciones.

Norma: (ASTM, A247-16, 2016)

1.3. Fundiciones Nodulares

Como se expuso anteriormente, las fundiciones nodulares también conocidas como fundiciones dúctiles o hierro dúctil, son aleaciones ferrosas, con una cantidad entre 3 y 4% de Carbono, donde se generan esferoides bien definidos de grafito. Los esferoides son concentradores de esfuerzos que interrumpen la estructura cristalina del hierro en menor medida, al compararlos con las hojuelas y las agrupaciones amorfas de grafito; por lo tanto se tiene un menor impacto en las propiedades de resistencia y ductilidad a diferencia de otras aleaciones de hierro fundido. (Avner, 1988); (Smith & Hashemi, 2006)

Pero para prevenir factores como: grafito flotante es decir grafito esparcido en la aleación, la contracción excesiva, la alta tendencia de enfriamiento, así como el impacto de un alto gradiente de temperatura, se ha establecido las cantidades óptimas de grafito y silicio para este tipo de aleaciones, indicadas en la curva mostrada en el Anexo II. (American Society for Metals, 2005)

Para que se produzca la esferización del grafito y se obtengan estructuras Tipo I y Tipo II principalmente como se observa en la Figura 1.1 es necesario añadir a la mezcla elementos que faciliten que se de este resultado, dentro de los cuales tenemos: magnesio, calcio, cerio, lantano, entre otros. Por otro lado se tiene elementos que impiden la formación de esferoides, siendo estos los siguientes: aluminio, arsénico, bismuto titanio, azufre, antimonio, entre otros. (American Society for Metals, 2005); (Donald R. Askeland, 2003)

El elemento de esferización más ampliamente utilizado es el magnesio, este es añadido a la mezcla líquida mediante aleaciones de níquel – magnesio con un 15% de magnesio aproximadamente, y a veces cobre- magnesio con 15 a 25 % de magnesio. (Barreiro, 1963)

El magnesio por si solo actúa como desulfurizante, por lo que en acción combinada con otros elementos presentes en la aleación genera la formación de los esferoides. La cantidad residual de magnesio esta generalmente entre el rango de 0.03% a 0.05%. (American Society for Metals, 2005; Donald R. Askeland, 2003)

La relación entre el porcentaje de esferización o nodulización del grafito, con el porcentaje de magnesio residual añadido a la mezcla se expone en la Figura 1.3, se puede observar la influencia del magnesio en la generación de los esferoides. (American Society for Metals, 2005)

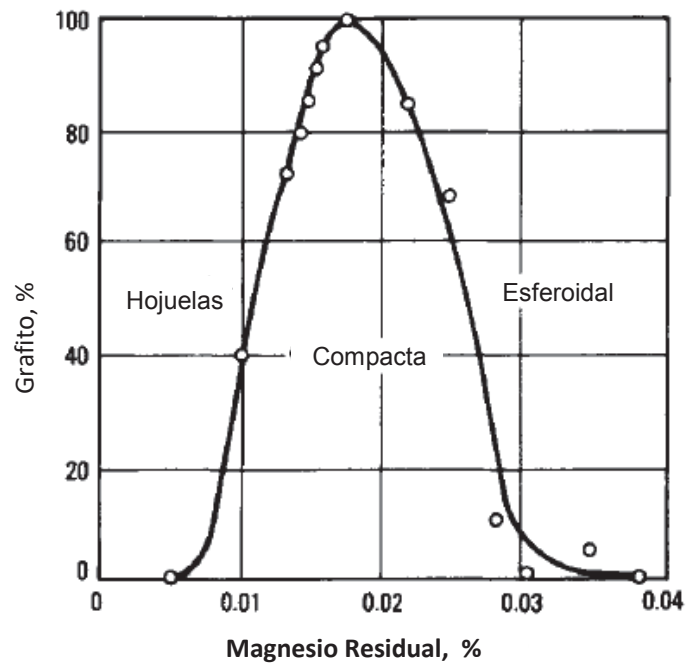


Figura 1.3. Influencia del magnesio residual en el forma de grafito.

(American Society for Metals, 2005)

A la relación entre la cantidad de grafito existente en la aleación con la cantidad de grafito que ha formado nódulos se conoce como nodularidad, si esta se expresa en porcentaje, como se puede ver en la Figura 1.4. (ASTM, A247-16, 2016)

Las propiedades de las fundiciones de hierro dúctil están muy ligadas a la formación de los nódulos de grafito en la aleación de tal manera que la nodularidad obtenida es una propiedad crítica, y podemos ver su influencia en la Figura 1.5, en donde se puede observar, la variación de las propiedades mecánicas en función de la nodularidad. (American Society for Metals, 2005)

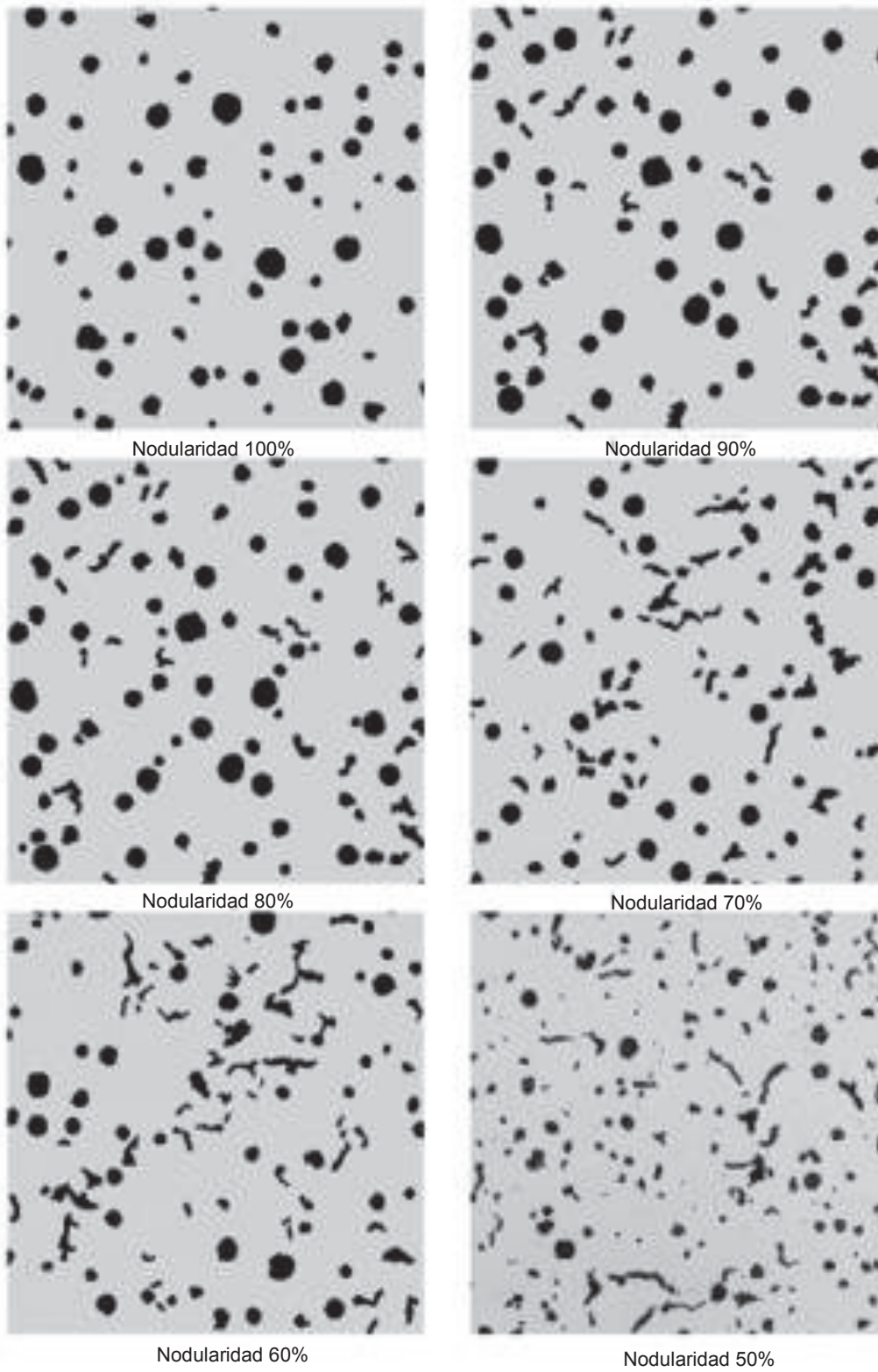


Figura 1.4. Comparación de nodularidad.

(ASTM, A247-16, 2016)

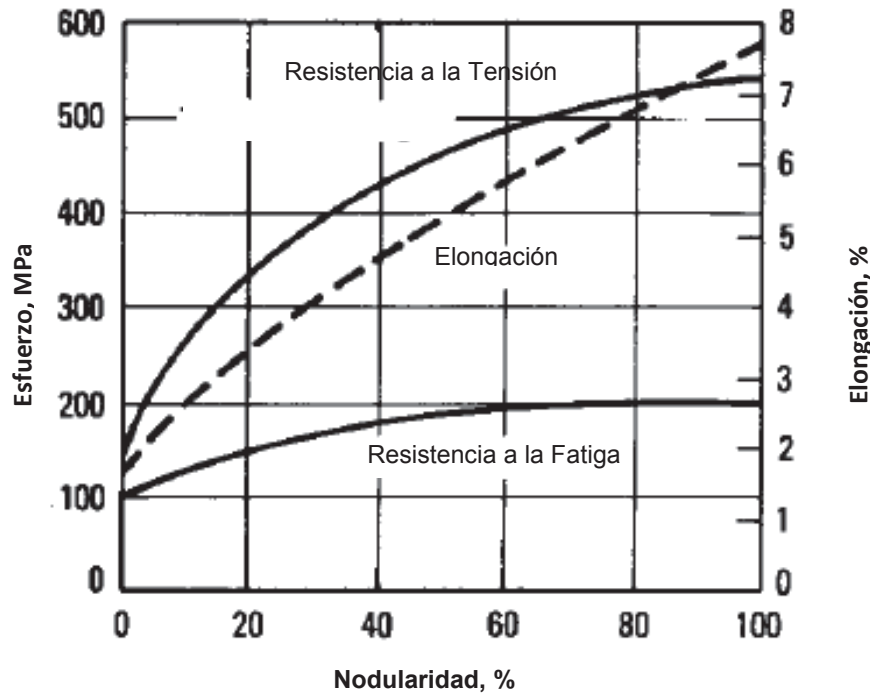


Figura 1.5. Propiedades mecánicas de fundiciones de hierro dúctil comparadas con el porcentaje de nodularidad.

(American Society for Metals, 2005)

La matriz microestructural es otro parámetro importante, propiedad que está ligada a la composición química del material, a la velocidad de enfriamiento, y al tratamiento térmico que se aplique al material, en caso de ser aplicado. (Donald R. Askeland, 2003); (Avner, 1988)

Se puede tener una matriz ferrítica como se observa en la Figura 1.6, si tiene un enfriamiento lento, esta matriz proporciona una ductilidad, tenacidad y maquinabilidad, máximas dentro de este tipo de aleaciones. Se considera una matriz ferrítica si se tiene hasta un máximo de 10% de perlita en la aleación. (Avner, 1988)

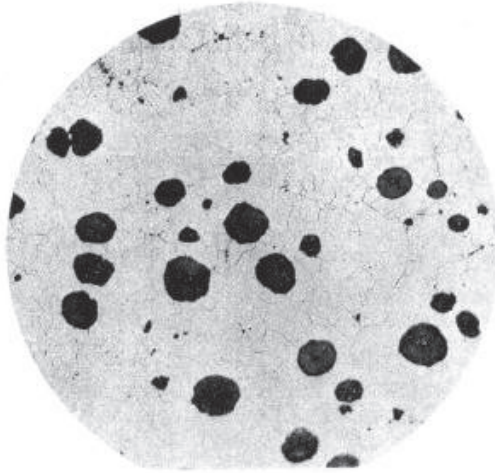


Figura 1.6. Hierro dúctil con matriz ferrítica 100X, atacado químicamente con: nital al 2%.

(Avner, 1988)

Una matriz perlítica, como se indica en la Figura 1.7, se puede obtener mediante un enfriamiento de una aleación de un contenido de carbono elevado, o mediante un tratamiento térmico de normalización, a una temperatura mayor a la temperatura del punto eutectoide y un posterior enfriamiento en aire. La matriz perlítica proporciona una mayor resistencia que la matriz ferrítica, pero con una ductilidad menor. Se puede obtener una matriz martensítica, al enfriar en aceite a la mezcla de la aleación desde temperaturas alrededor de 930 °C, para realizar el posterior revenido de alivio de esfuerzos, de tal manera que finalmente se tendrá una matriz bainítica en los hierros dúctiles austemperados como se expuso en la Tabla 1.1. (Donald R. Askeland, 2003); (Callister, 1990)

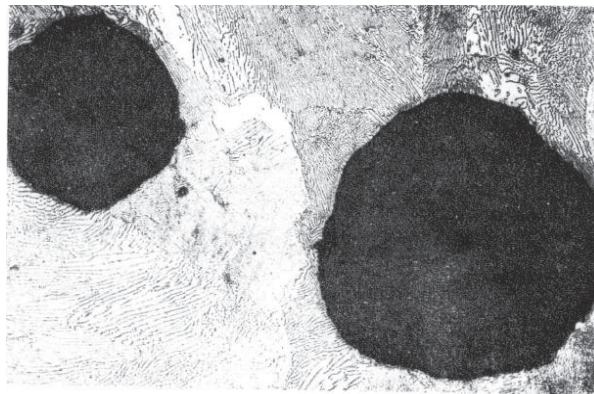


Figura 1.7. Hierro dúctil perlítico 500X, atacado químicamente con nital 2%.

(Avner, 1988)

Las propiedades mecánicas de hierros dúctiles con matrices ferríticas y perlíticas se exponen en la Tabla 1.3.

Tabla 1.3. Propiedades mecánicas de tipos básicos de hierro dúctil.

Tipo	Contenido de la Aleación	Resistencia a la Tension		Resistencia a la Fluencia		Elongación	Dureza (BHN)
		ksi	Mpa	ksi	Mpa		
Ferrítico	Bajo	55	379	35	241	25	130
	Alto	90	621	70	483	12	210
Perlítico	Bajo	80	552	60	414	10	200
	Bajo*	130	896	90	621	7	275
	Alto	130	896	110	758	2	275

* Normalización. (Avner, 1988)

Como se indica en la Tabla 1.3 se puede obtener matrices tanto ferríticas o perlíticas con una cantidad de aleantes diferentes, lo cual muestra la importancia de la velocidad de enfriamiento, de igual manera que la composición química, puesto que hablar de aleaciones altamente aleadas, se expresa que la cantidad de elementos aleantes tienen una influencia directa en las propiedades del hierro dúctil. El tratamiento térmico empleado en la aleación es de igual importancia en las propiedades mecánicas, a pesar de que los hierros dúctiles pueden ser obtenidos por fundición al someter a la aleación a un tratamiento térmico, se obtiene una variación en las propiedades mecánicas.

Se puede obtener matrices austeníticas, en hierros dúctiles, pero para obtener dicha estructura, estas aleaciones tienen que tener una cantidad muy alta de aleantes. Se ejemplifica la composición química de dos fundiciones dúctiles con matriz austenítica: Composición uno: 3% C, 2.5% Si, 20% Ni, 2% Mn. Composición dos: 3% C, 2% Si, 20% Ni, 1% Mn, 1.5% Cr (Donald R. Askeland, 2003)

Se puede obtener una combinación de las matrices ferríticas y perlíticas, como se indica en la Figura 1.8. Al tener la combinación en la estructura ferríticas y perlíticas también se obtiene una combinación de propiedades. (Avner, 1988); (Callister, 1990); (Donald R. Askeland, 2003)

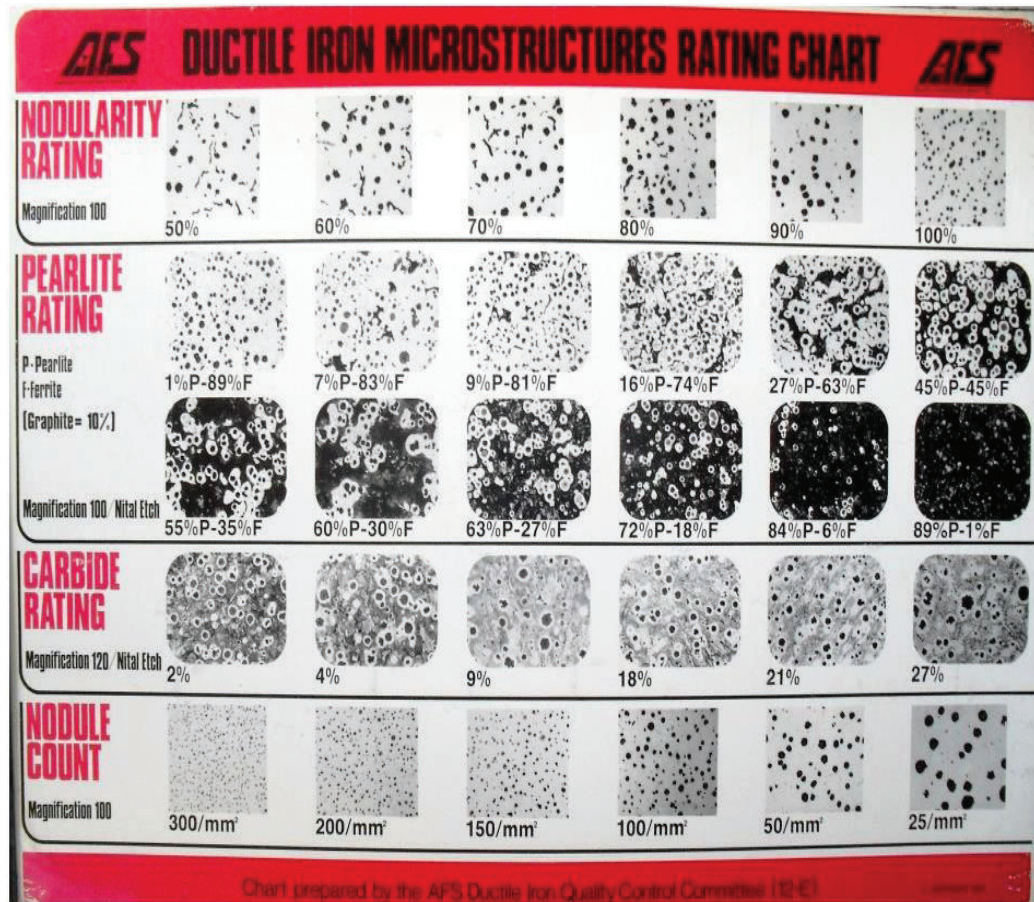


Figura 1.8. Carta de clasificación de microestructuras presentes en fundiciones nodulares.

(ASF Ductile Iron Quality control Committee, 2010)

Con la combinación de diferentes matrices se obtendrán propiedades mecánicas diferentes, y por lo tanto aplicaciones diferentes, las cuales están especificadas en el la tabla 1.4, además del proceso para manufacturar la aleación. Cabe recalcar que se empleó dos especificaciones diferentes para hierros dúctiles, por lo cual esto se indica también en la Tabla 1.4. (American Society for Metals, 2005)

Las propiedades mecánicas de las aleaciones mostradas en la Tabla 1.4 están indicadas en la Tabla 1.5 donde se muestra el límite a la tensión, el límite a la fluencia, la elongación y las aplicaciones en concreto de cada aleación tratada.

Tabla 1.4. Composiciones y usos generales para grados estándar de hierro dúctil.

Especificación	Nombre	Matriz	Proceso de manufactura	Uso General
ASTM A 536	60-40-18	Ferrítica	Puede ser recocido	Partes resistentes a golpes; servicios de baja temperatura
ASTM A 537	65-45-12	Mayormente Ferrítica	De fundición o recocido	Servicio General
ASTM A 538	80- 55-06	Ferrítica Perlítica	De fundición	Servicio General
ASTM A 539	100-70-03	Mayormente Perlítica	Puede ser normalizado	Mejor combinación de resistencia y resistencia al desgaste y mejor respuesta al endurecimiento superficial
SAE J434	D7003	Perlítica	De fundición	Piezas de alta resistencia que requieren muy buena resistencia al desgaste y buena respuesta al endurecimiento selectivo
ASTM A 541	120-90-02	Martensítica	Templado y enfriado en aceite	Alta resistencia y resistencia al desgaste

(American Society for Metals, 2005)

Tabla 1.5. Propiedades mecánicas y aplicaciones típicas para calidades estándar de hierro dúctil.

Nombre	Resistencia a la tracción		Resistencia a la fluencia		Elongación (%)	Aplicación típica
	Mpa	ksi	Mpa	ksi		
60-40-18	414	60	414	40	18	Partes de contención de presión como válvulas
65-45-12	448	65	310	45	12	Partes mecánicas, como elementos de choque o a cargas cíclicas
80- 55-06	552	80	379	55	6	Cigüeñales, engranajes y rodillos

100-70-03	689	100	483	70	3	Engranajes de gran carga, y componentes de maquina
D7003	689	100	483	70	3	Engranajes
120-90-02	827	120	621	90	2	Piñones, engranajes

(American Society for Metals, 2005)

Para el análisis metalográfico es necesario emplear un reactivo químico, con el cual reaccione la microestructura de la aleación y de esta manera se revele la estructura buscada. Hay que tener muy en cuenta el reactivo a utilizarse, para lo cual se emplea Norma ASTM E407- 07. De acuerdo a la norma, los reactivos para visualizar la microestructura de las aleaciones de hierro carbono, son: Nital, Prical, Super Prical, Fry's. Las más empleadas en el medio son Nital y Prical, con ambas soluciones el ataca solo dura unos pocos segundos, para más información consultar en la norma. (ASTM- E407-07, 2015)

Es importante la selección del reactivo para el ataque químico, pues depende de te tipo de micro estructura se quiere revelar; al utilizar el reactivo incorrecto, se puede realizar un análisis erróneo. (ASTM- E407-07, 2015)

2. METODOLOGÍA

2.1. Software Libre

Existen varios de lenguajes de programación, como Matlab, en donde se puede realizar un código de programación, pero es necesario adquirir la licencia oficial del programa, lo cual incrementa los costos de investigación y uso. Es posible acceder a licencias estudiantiles para poder hacer proyectos como es el caso del presente trabajo; pero bajo estas condiciones ciertas librerías no están hábiles para el usuario. Además para correr el programa en otra computadora es necesario adquirir otra licencia, por lo que su difusión es muy complicada. (Hidalgo, 2016)

El software libre se ha desarrollado con la filosofía de que el código principal debe estar abierto, para que las personas que están ligadas con la investigación y el desarrollo puedan aprovechar sus ventajas. (Hidalgo, 2016) ” En 1985, Richard Stallman definió el concepto de Software Libre en cuatro libertades”: (Hidalgo, 2016)

1. Libertad de correr el programa, por cualquier motivo.
2. Libertad de estudiar las fuentes del programa y cambiarlo.
3. Libertad de copiar y distribuir copias exactas, en cualquier momento sin problema.
4. Libertad de hacer modificaciones y distribuirlas, para beneficio de la comunidad.

2.1.1. Python y Open CV

Python es un lenguaje de programación, fácil de aprender y de gran capacidad. El cual presenta estructuras de alto nivel con un enfoque simple pero efectivo, en la programación orientada a objetos, lo que lo convierte en un lenguaje muy empleado en ingeniería. (Python , 2017)

Para la utilización del programa es necesario incluir diferentes librerías, como numpy, que es empleada en el análisis, y estructuración de matrices; otra librería es Matplotlib que genera gráficas en 2D y 3D; la librería que proporciona las herramientas para leer y manipular imágenes es OpenCV; y finalmente para realizar una interfaz gráfica, se emplea PyQt4 que es una librería que conecta a Python con el desarrollador de programas Qt. (Willems, 2017); (The Matplotlib development team., 2017); (OpenCV-Python, 2016).

A través del desarrollo de código en Python se plantea, un software que emplee el procesamiento en imágenes de metalografías de fundiciones nodulares.

2.2. Procesamiento de Imágenes

2.2.1. Escala de Grises

La librería OpenCV carga imágenes en formato BGR, es decir azul (Blue), verde (Green) y rojo (Red). Cada pixel de la imagen tiene un vector que expresa el color que tiene cada pixel. El vector presenta tres componentes (B, G, R), las cuales tienen un valor que va de 0 a 255, cuyas combinaciones definen los colores del vector. (OpenCV-Python, 2016)

Dentro de esta librería es posible hacer una conversión de formato y cambiar el mapa de colores de una imagen, para el caso de estudio se transforma del formato BGR a una escala de grises. En escala de grises, cada pixel está definido por un solo valor, entre 0 y 255; el valor de cero 0 representa al color negro, mientras que el valor de 255 representa al color blanco. Cualquier valor intermedio define una mezcla entre los dos colores con un porcentaje mayor de un color u otro dependiendo su ascenso en la escala. (OpenCV-Python, 2016)

2.2.2. Discriminación matricial

Al pasar a una escala de grises la imagen cargada, a cada pixel tiene un valor dentro de 0 y 255, se considera a una imagen como una matriz, tal que se puede extraer el tamaño de dicha matriz; se genera una matriz con el mismo tamaño de la imagen, con valores de cero.

Para el análisis de propiedades se analiza pixel a pixel, a través de un bucle " for " para cada posición dentro de la matriz, se compara el valor de la escala de grises del elemento con un rango de valores dados por el usuario, rango que depende de la propiedad que se vaya a analizar. Si el valor del elemento está dentro del rango propuesto, el pixel se imprimirá en la segunda matriz generada, mientras que los pixeles que estén fuera de los valores seteados por el usuario asumirán un valor predeterminado. Este proceso se ilustra en la Figura 2.1.

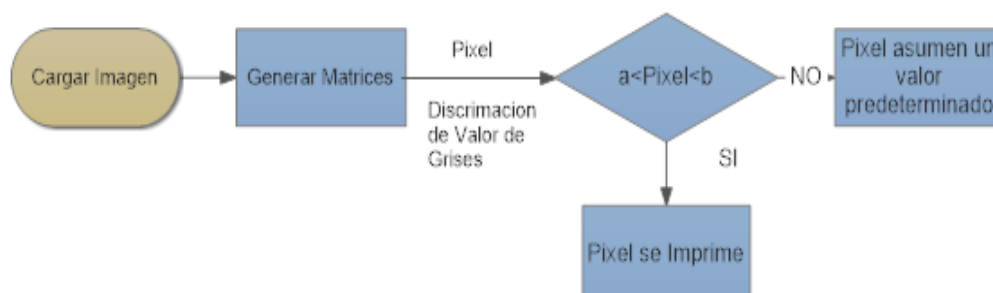


Figura 2.1. Diagrama de flujo de la discriminación de matrices.

El código de programación se encuentra expuesto en el Anexo III, al final del presente trabajo.

2.2.3. Análisis de Perlita, Ferrita y Grafito

El análisis de los porcentajes de microestructura revelada en la metalografía se obtiene usando la discriminación matricial planteada, añadiendo un vector contador. A este vector se añadirán los pixel que cumplen con la condición planteada en la discriminación matricial. Se realizara un conteo de los elementos de este vector, y se compara con la cantidad de pixeles presentes en la imagen, lo que se resume en la Figura 2.2.

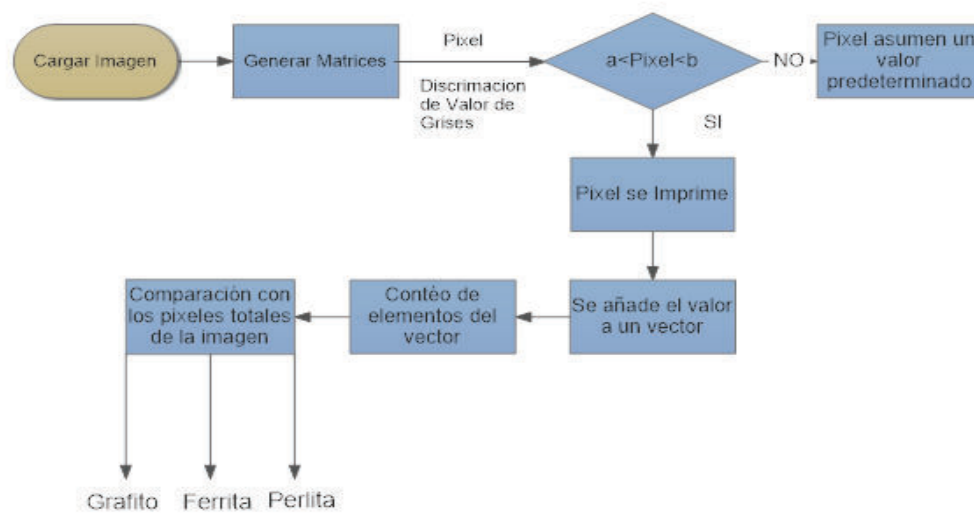


Figura 2.2. Diagrama de flujo de obtención de porcentajes de ferrita perlita y grafito.

Para realizar un análisis acertado, es necesario plantear dos valores umbrales en el algoritmo de la discriminación matricial. En primer lugar es necesario identificar el valor umbral para la identificar del grafito y posteriormente es necesario encontrar el valor umbral para que de manera similar la ferrita pueda ser identificada en la metalografía. Se plantea una selección del valor de escala de grises empleando scrolls como se muestra en le Figura 2.3. Se realizará un ejemplo del análisis empleando la metalografía que se ilustra en la Figura 2.4 empleando la interfaz gráfica del código del presente estudio, en donde se variará los valore umbrales expuestos

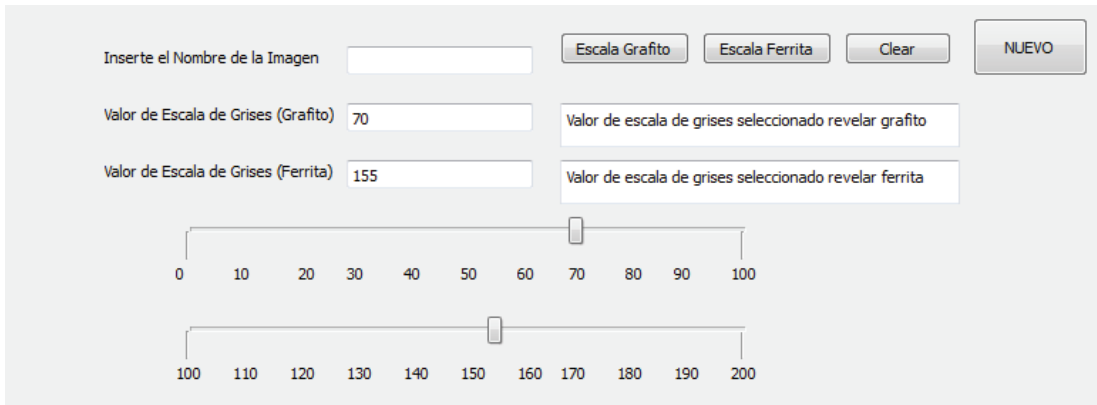


Figura 2.3. Sección del programa generado; cargado de imagen y selección de valor de escala de grises.

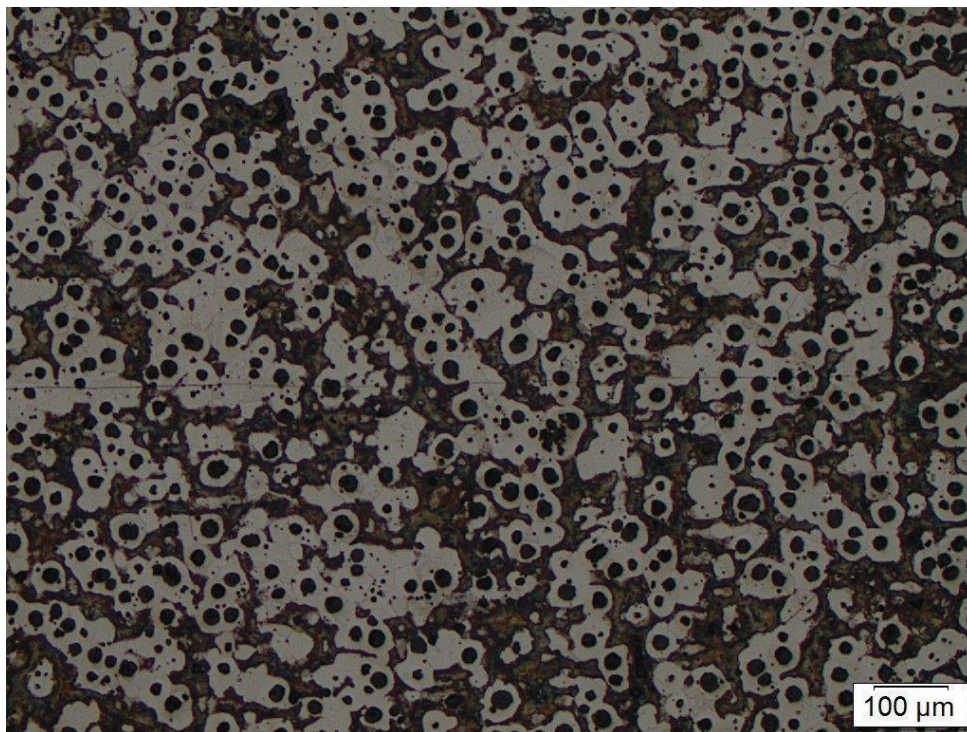


Figura 2.4. Metalografía de hierro fundido nodular.

(Metalografía Obtenida: Laboratorio de Metalografía de la Facultad de Ingeniería Mecánica EPN)

Para la identificación del grafito se varía el valor de la escala de grises, como se visualiza en la Figura 2.4, hasta que se pueda visualizar claramente los nódulos de grafito. Al obtener el valor de escala de grises adecuado, se da un orden de ejecución, los valores desde cero hasta el valor obtenido, serán los límites en el algoritmo mostrado en la Figura 2.2, obteniéndose así el porcentaje de grafito en la metalografía.

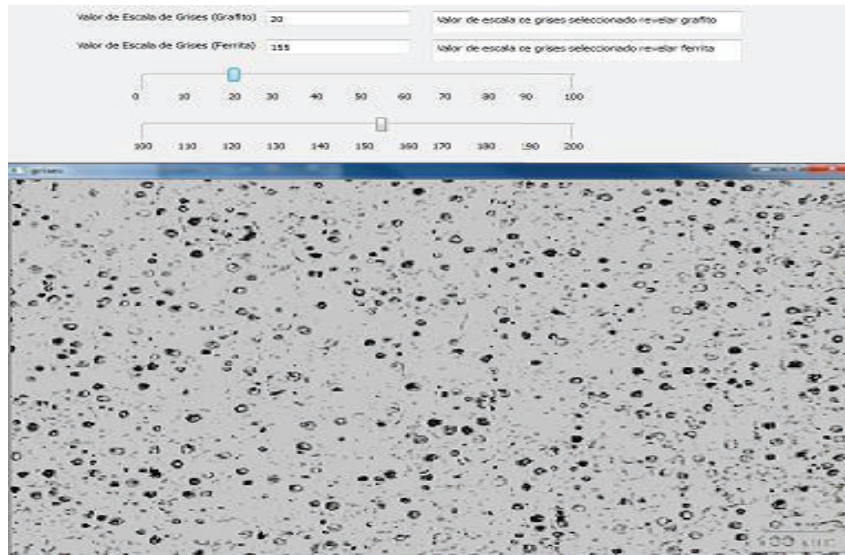


Figura 2.5. Variación de la escala de Grises de la Metalografía para identificar el grafito.

De forma similar que la identificación del grafito se varía el valor de la escala de grises en forma dinámica, hasta comprobar que se ha revelado toda la ferrita presente en la metalografía, como se ve en la Figura 2.5, obteniéndose así el segundo valor umbral necesario para realizar la discriminación matricial, y obtener los valores porcentuales de la microestructura perlítica y ferrítica.

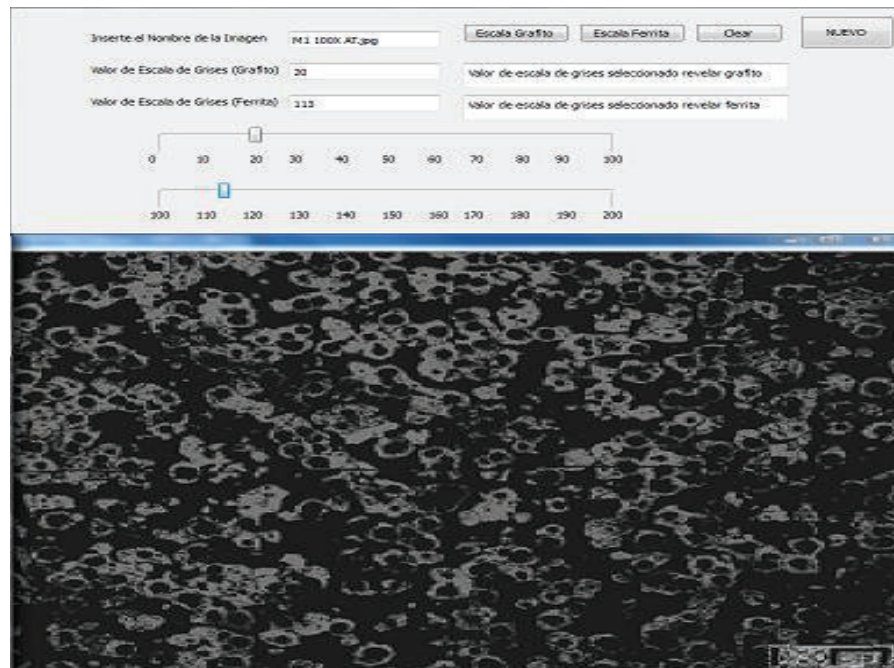


Figura 2.6. Variación de la escala de grises para identificar la ferrita.

La ferrita es la micro estructura más clara en la metalografía, por lo se asignará el segundo valor umbral y el valor de 255 en el algoritmo mostrado en la Figura 2.2. Por otro lado los valores asignados para la obtención porcentual de perlita son los dos valores umbrales. Para obtener los valores porcentuales de ferrita y perlita en la metalografía, es necesario obviar los pixeles de la metalografía correspondientes al grafito.

2.2.4. Análisis de Nodularidad

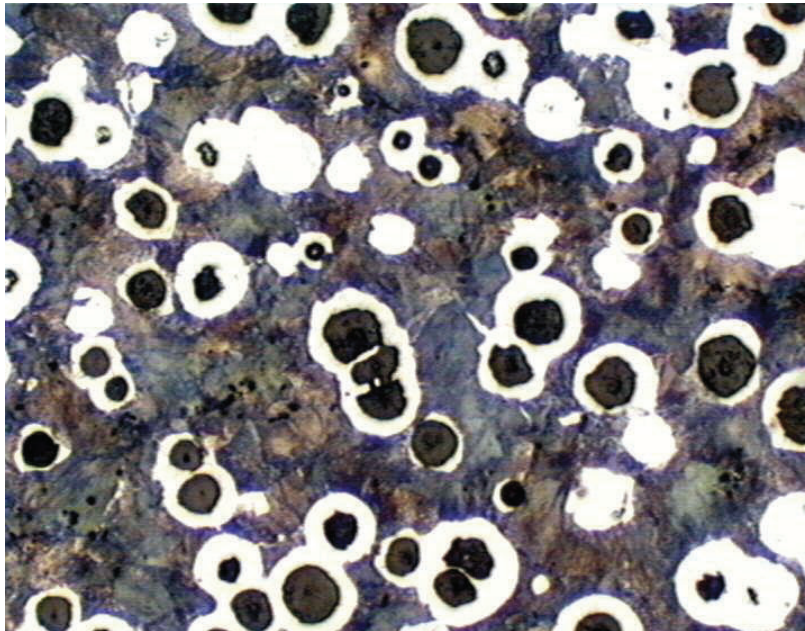


Figura 2.7. Metalografía de una fundición nodular.

(Metalografía de la Fundición , s.f.)

Como se indicó en el capítulo anterior, la nodularidad expresa el porcentaje de cuanto grafito de la aleación se ha concentrado en nódulos. Se tomará la metalografía mostrada en la Figura 2.7 como ejemplo del análisis, el cual se ha seguido el siguiente procedimiento:

Obtención de Datos: Es necesario aislar los nódulos de la metalografía para poder analizarlos y posteriormente comparar con el grafito presente en la metalografía. Al realizar la discretización manual podemos obtener una imagen como se ve en la Figura 2.8. No es posible eliminar el grafito libre presente en la metalografía debido a que tiene el mismo valor de escala de grises que los nódulos. Los métodos automáticos, de detección de círculos son ineficientes para este caso.

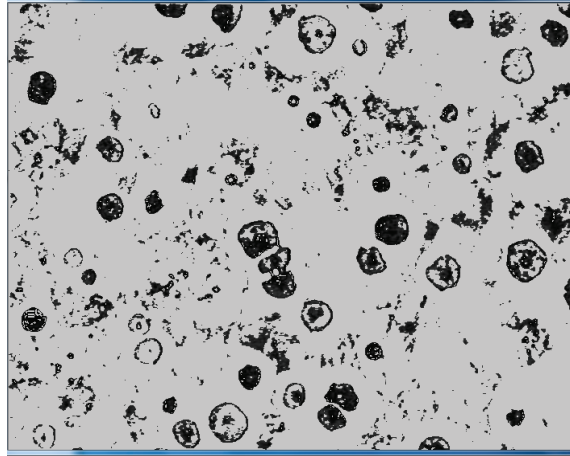


Figura 2.8. Discretización de grafito de una metalografía de fundición nodular.

Se plantea la selección de los nódulos empleando el mouse de la imagen analizada, donde el usuario diferencia entre nódulos, inclusiones y concentraciones de grafito amorfas. Para seleccionar un nódulo, hay que seleccionar puntos opuestos, que describan el diámetro de un nódulo, como se indica en la Figura 2.9.

Al instante que se selecciona un nódulo se ingresa la información del tamaño del mismo en una matriz en blanco del mismo tamaño de la imagen, como se indica en la Figura 2.9.

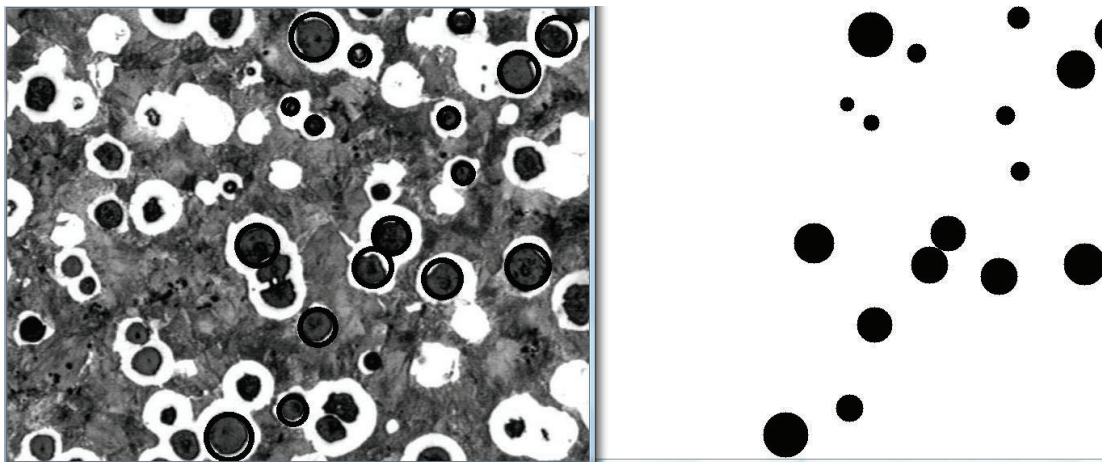


Figura 2.9. Selecciones de nódulos y generación de círculos en matriz en blanco.

Aislamiento de Nódulos y Análisis de Nodularidad: Una vez seleccionados los nódulos, se realiza una operación de imágenes. Se invierte los colores de la matriz donde se generaron los círculos generando una máscara, se añade una instrucción, en donde se corte todo lo que sea blanco en esta matriz, y posteriormente se sobrepone a la imagen de la metalografía original. Como resultado de eso los nódulos quedan aislados, finalmente se añade los nódulos aislados a un fondo flanco donde se puede visualizar con claridad, como se ve en la Figura 2.10, 2.11 y 2.12.

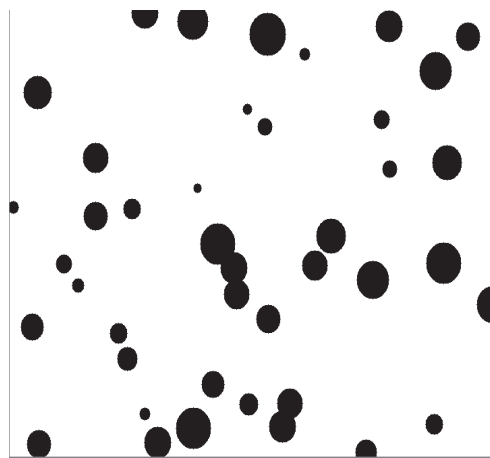


Figura 2.10. Máscara.

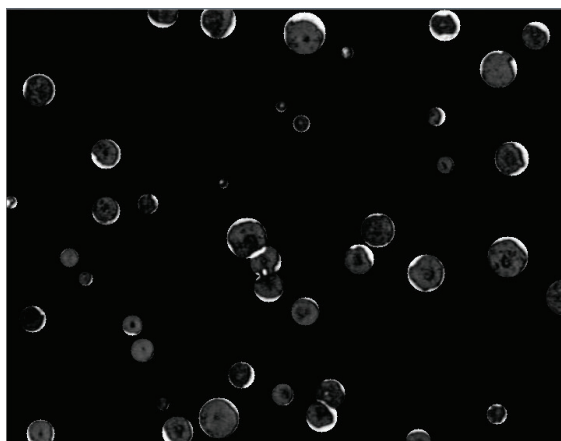


Figura 2.11. Imágenes superpuestas.

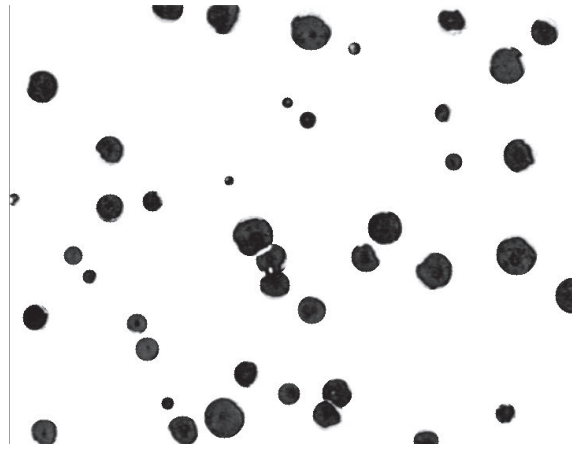


Figura 2.12. Nódulos aislados.

Al obtener finalmente una imagen con los nódulos aislados, se contabiliza los pixeles que sean diferentes de blanco, el proceso es el mismo que la discriminación de matrices. Posteriormente se compara, con el número de pixeles que obtenidos al calcular el porcentaje de grafito. Finalmente se devuelve un porcentaje el porcentaje de nodularidad. Este proceso se ilustra en el diagrama de flujo que se muestra en la Figura 2.13

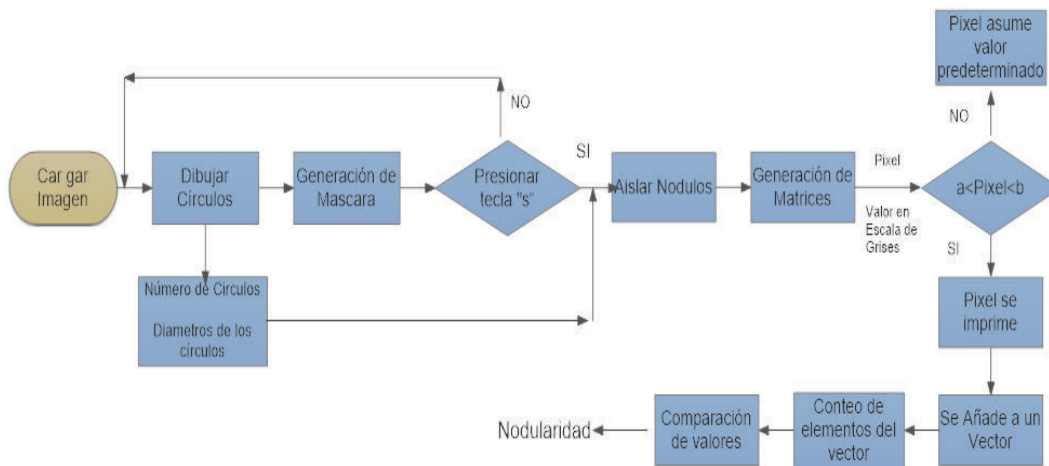


Figura 2.13. Diagrama de flujo del proceso de obtención de la modularidad.

2.2.5. Tamaño y Conteo de Nódulos

Mientras se realiza la selección de los nódulos, se añade a un vector el valor del diámetro de cada círculo generado, el cual es lo más aproximado al nódulo. Finalmente al recibir la señal de la tecla “S”, como se indica en la Figura 2.13, se realiza una sumatoria de los elementos que tiene el vector; y se divide para el número de elementos en el vector. Para obtener el valor promedio del diámetro de nódulos. El número de nódulos presentes en la metalografía se obtienen del número de elementos en el vector de diámetros de los círculos. Se puede revisar el código de programación en el Anexo III.

Finalmente es necesario adjuntar a los resultados adquiridos una magnitud. En el caso del conteo de nódulos debe ser expresado en nódulos por área, la cual es obtenida, de los datos del software “Stream Essentials”, empleado en el laboratorio de metalografía de la Facultad de Ingeniería Mecánica. En el cual una metalografía tomada a un aumento de 100X presenta un área de 1.29 mm². Al realizar una comparación entre los pixeles de la imagen, la medición manual realizada en el programa, que se ilustra en la Figura 2.14, y el área total expresada en los resultados del programa se obtiene la relación pixel - milímetro mostrada en la Ecuación 2.1. De manera que el tamaño promedio del diámetro de los nódulos es expresado en milímetros. (ASTM, A247-16, 2016)

$$1 [pixel] = 365 \times 10^{-4} [mm]$$

Ecuación 2.1. Relación entre pixel y milímetro.

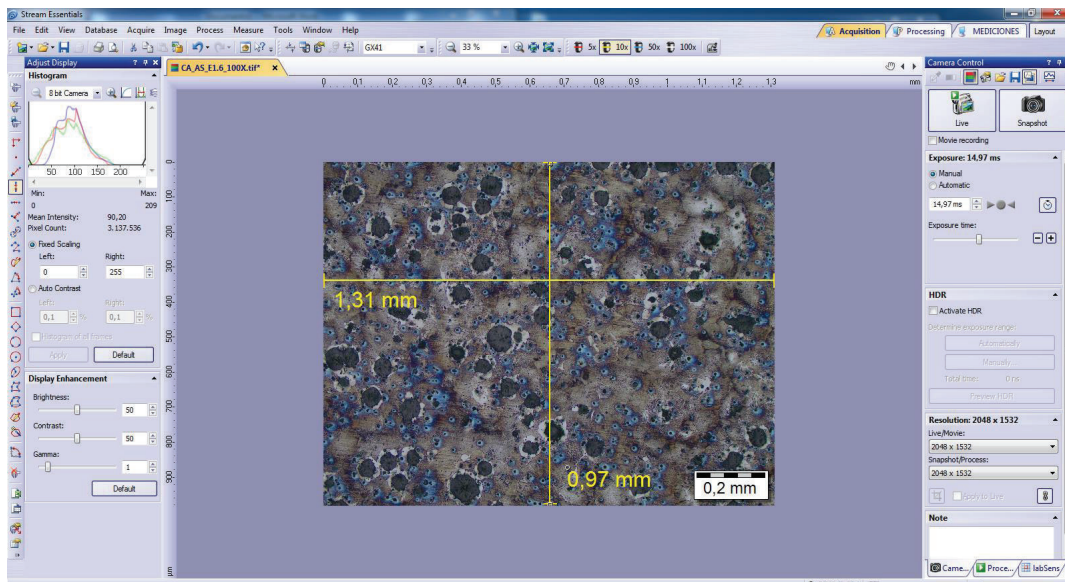


Figura 2.14. Medición en mm en el software “Stream Essentials”.

2.2.6. Tipo de Grafito

Se selecciona el tipo de grafito presente en la metalografía, se selecciona entre el tipo de grafito correspondiente a fundiciones de hierro dúctil, es decir entre el Tipo I, Tipo II, Tipo V y Tipo VI. En caso de existir la presencia de un segundo tipo se realizará una selección del tipo secundario de grafito, en la imagen de los nódulos aislado obtenida, generándose una máscara y la operación entre imágenes similar a la realizada para aislar los nódulos pero en este caso se empleará rectángulos, como se ve en la Figura 2.15. Finalmente se realiza una discriminación matricial entre la imagen de los nódulos aislados y la imagen de los nódulos aislados en segunda instancia comparando porcentualmente la cantidad del tipo de grafito principal y secundario.

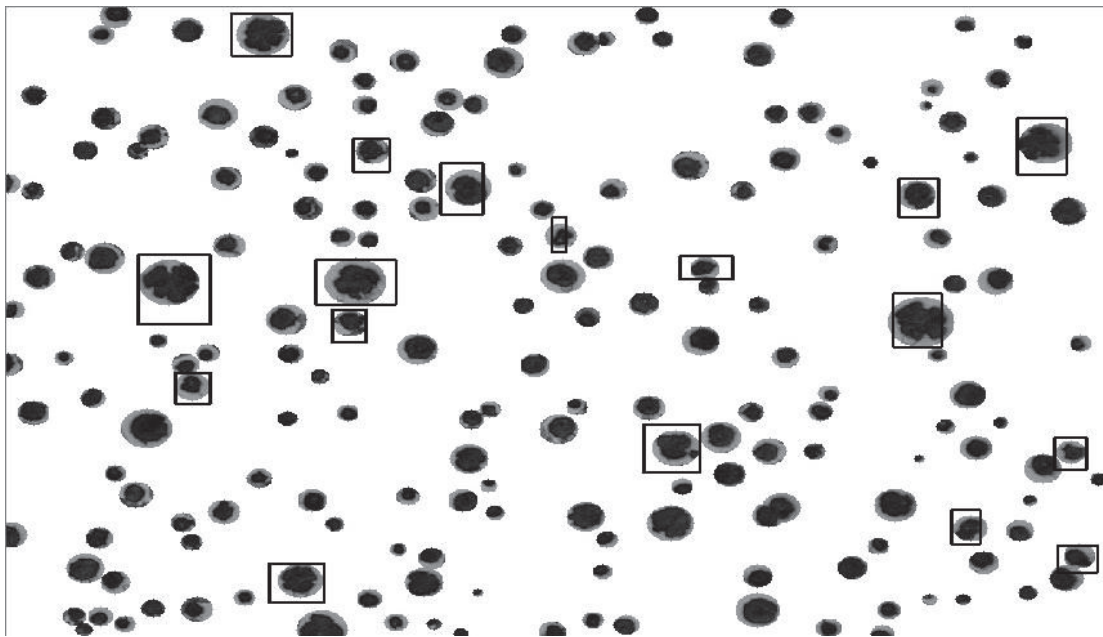


Figura 2.15. Selección de tipo secundario de grafito.

2.3. Adición de resultados a una base de datos

Finalmente para llevar constancia del análisis realizado, se añaden los resultados obtenidos en una base de datos de MySQL, como se indica en la Figura 2.16. En la tabla está incluido el nombre del archivo analizado, el porcentaje de ferrita, perlita y grafito, así como el valor de escala de grises empleado para identificar el grafito y la ferrita, además se añade el porcentaje de modularidad, el tamaño promedio, el número de nódulos, y el

tipo de concentración de grafito presente en la metalografía; finalmente se añadirá el nombre de la persona que realiza el análisis, la fecha, así como el reactivo empleado para revelar la microestructura, y cualquier observación realizada.

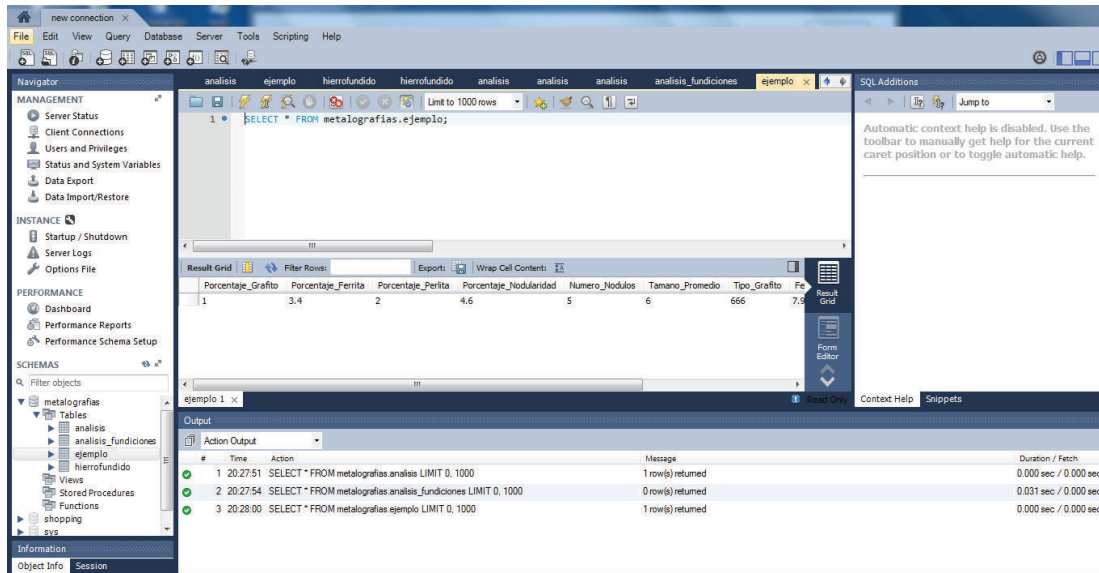


Figura 2.16. Base de datos en MySQL.

2.4. Experimentación

En la búsqueda de corroborar los resultados del software planteado es necesario obtener muestras, las cuales se someterán a un análisis por el software realizado en el presente estudio y el empleado en el laboratorio de metalurgia de la Facultad de Ingeniería Mecánica. Se extrajo una muestra del actuador de una válvula mariposa, empleada en tuberías del servicio de agua potable, la cual estaba presente el laboratorio

2.4.1 Obtención de muestras

A partir de la probeta obtenida se realizó el proceso de preparación del elemento para ser observado en el microscopio, el desbastado, lijado y pulido, indicado en la Figura 2.17, fue realizado según indicaciones la norma ASTM E3-11. (ASTM INTERNATIONAL, 2011).



Figura 2.17. Muestra obtenida y la máquina de pulido de fondo.

Con la probeta lista, se procedió a realizar el ataque químico para revelar la microestructura de la aleación con nital a 2%, es decir ácido nítrico diluido en agua, en una solución al dos por ciento. El equipo empleado en el laboratorio es microscopio OLYMPUS SC30 de la serie U-TV0.5XC-3, número 4G02894, elaborado en Toykio Japon, que se indica en la Figura 2.18. El microscopio esta enlazado a una computadora presente y emplea el software Stream Essentials



Figura 2.18. Microscopio del Laboratorio de Metalografía de la Facultad de Ingeniería Mecánica.

Con la probeta lista, se procedió a realizar el ataque químico para revelar la microestructura de la aleación con nital a 2%, es decir ácido nítrico diluido en agua, en una solución al dos por ciento. El equipo empleado en el laboratorio es microscopio OLYMPUS SC30 de la serie U-TV0.5XC-3, número 4G02894, elaborado en Toykio Japon. El microscopio esta enlazado a una computadora presente y emplea el software Stream Essentials. Software con el cual se realizó el análisis metalográfico en el laboratorio.

2.4.2. Análisis de muestras proporcionado por el laboratorio

El software “Stream Essentials”, proporciona porcentajes de las microestructuras presentes en diferentes aleaciones analizadas en el laboratorio, el estudio se concentró en el análisis de metalografías de hierro fundido dúctil, por lo que la norma usada por el programa para realizar la comparación de las imágenes es la norma ASTM A247-10.

Para el análisis correspondiente, de manera análoga que en el software planteado en el presente estudio, se identifica al grafito presente en la metalografía como podemos observar en la Figura 2.19, en donde el grafito es marcado con el color rojo.

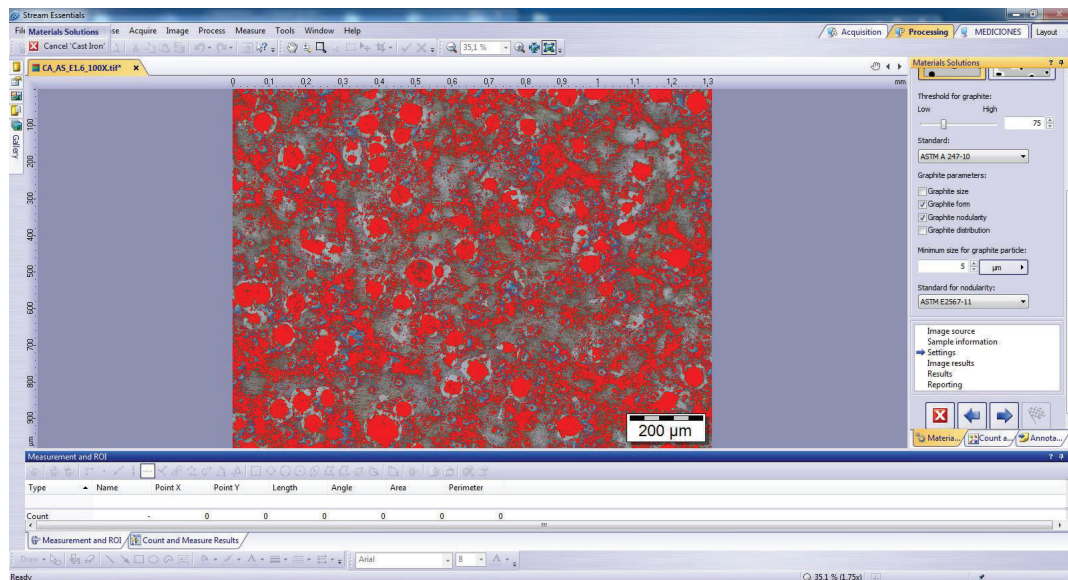


Figura 2.19. Identificación de grafito en el software Stream Essentials.

Por otro lado es necesario identificar las áreas en donde se tenga ferrita como se indica en la Figura 2.20, se necesita ajustar el valor límite en la escala de colores, de forma que toda la ferrita del material sea identificada por el programa.

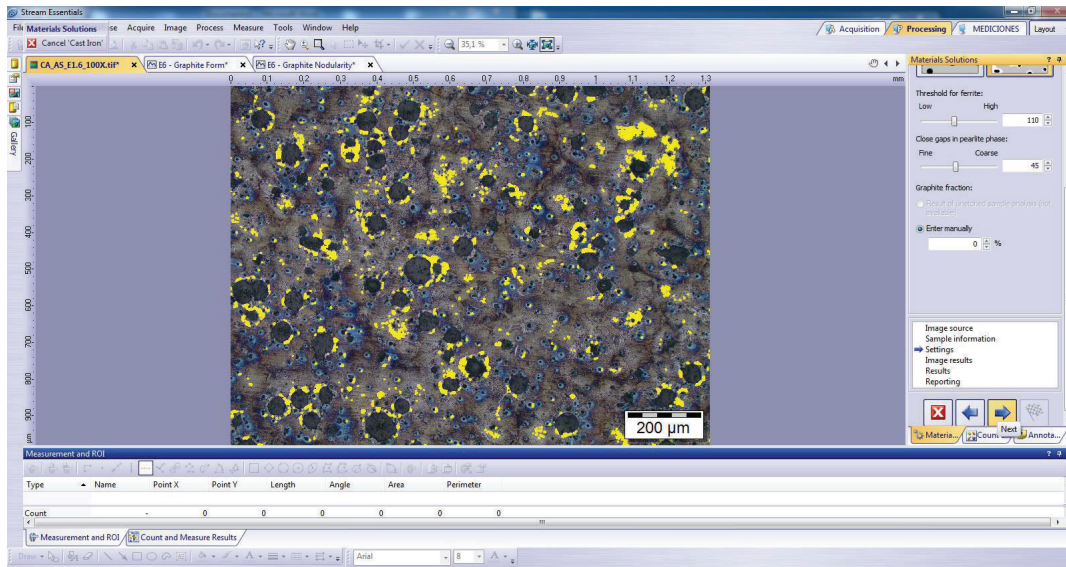


Figura 2.20. Identificación de ferrita en el software Stream Essentials.

El programa del laboratorio realiza dos análisis diferentes, el primero en el cual se analiza el porcentaje de las microestructuras, en el caso de estudio se identifica porcentualmente la ferrita y la perlita presentes en las aleaciones de hierro fundido, finalmente se presenta los resultados como se observa en la Tabla 2.1.

Tabla 2.1. Resultados del análisis de microestructuras proporcionado por el software Stream Essentials.

Cast Iron analysis (Summary)	
Standar	ASTM A247-10
Standar2	ASTM E2567-11
Total Sample Area	1,29 mm ²
Date	17-04-20176
Ferrite Ratio	4,10%
Perlite Ratio	95,90%

El segundo análisis que el software realiza, es un análisis del grafito presente en la aleación, añadiendo los resultados obtenidos en el análisis de microestructura realizado previamente, los resultados se muestran como se indica en la Tabla 2.2.

Tabla 2.2. Resultados del análisis de grafito proporcionado por el software Stream Essentials.

Cast Iron analysis (Summary)	
Standar	ASTM A247-10
Standar2	ASTM E2567-11
Total Sample Area	1,29 mm ²
Date	17-04-20176
Graphite Form	III(42%) IV(32%)V(23%)
Ferrite Radio	4,10%
Perlite Ratio	95,90%
Graphite Fraction	27,30%
Nodularity by Area (ASTM E2567)	16,80%
Nodularity by Count (ASTM E2567)	13,50%

2.5. Interfaz Gráfica

Para que el usuario del programa, pueda operar el código sin inconveniente, se plantea una interfaz gráfica ,la cual se indica en la Figura 2.21, la cual se divide en cuatro zonas la primera en se plantea los parámetros de análisis de la imagen, la zona de análisis de la microestructura y contenido de grafito, la tercera zona dedicada al análisis de los nódulos presentes en la metalografía, y finalmente la zona en donde se adjuntará datos como la persona a cargo, así como la fecha, reactivo empleado y observaciones.

El código de programación de la interfaz gráfica se encuentra añadido en el Anexo IV.

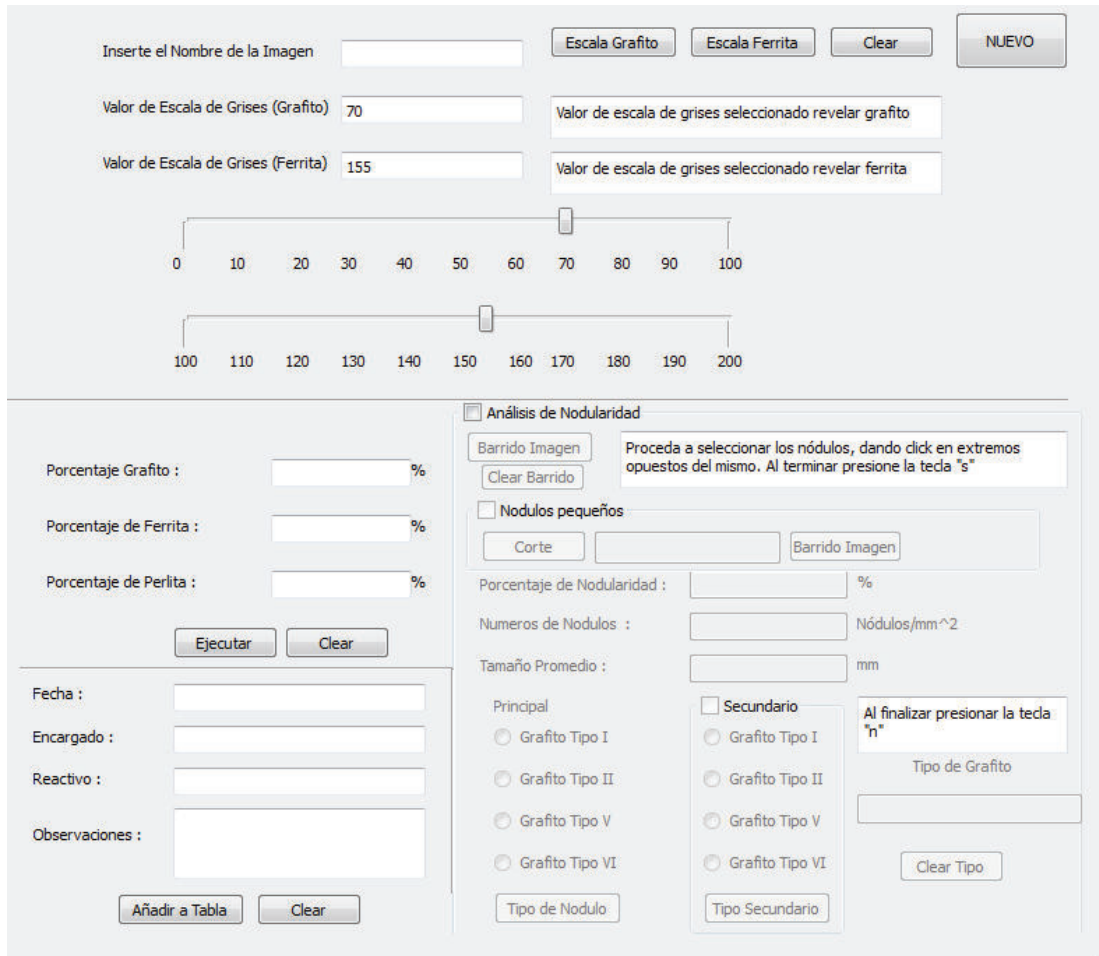


Figura 2.21. Interfaz gráfica del programa realizado.

En la primera zona de la interfaz es donde se añadirá el nombre de la imagen a analizar con su extensión de archivo, en este lugar es donde se planteará las condiciones de análisis de una metalografía, en donde a través de los scrolls mostrados en la Figura 2.21, se cambiará de manera dinámica el valor de la escala de grises hasta que el grafito y la ferrita sean claramente identificados en la metalografía de tal manera que se realice un análisis adecuado de la muestra metalográfica. Para lo cual, una vez ingresado el nombre de la imagen con la extensión correspondiente, se presionará el botón “Escala de grafito”, y se moverá el scroll ubicado superior, y se observará la variación dinámica de la metalografía en una ventana generada por la librería OpenCV. Por otro lado para se realiza el mismo procedimiento, para obtener identificar el valor en escala de grises para la ferrita, empleando el botón “Escala Ferrita” y el scroll ubicado en la parte inferior de la Figura 2.22.

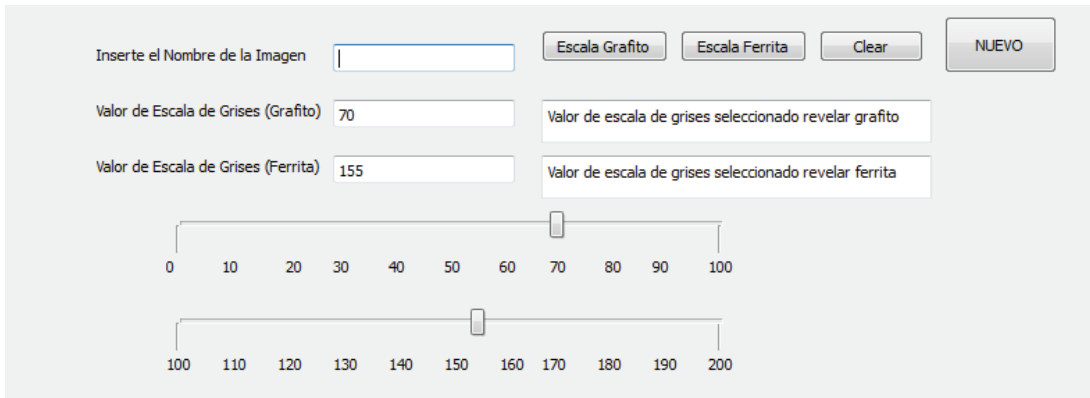


Figura 2.22. Primera parte de la interfaz gráfica donde se definen las condiciones de análisis.

Utilizando el botón “Clear” se borrarán el nombre de la imagen así como los valores de grafito y ferrita. Finalmente el botón “NUEVO” limpia todos los valores en la interfaz para, de tal manera que se puede realizar un nuevo análisis.

Una vez seleccionado los valores en escala de grises del grafito y la perlita, en la segunda zona, indicada en la Figura 2.23, al presionar “Ejecutar”, se realizará el cálculo de los porcentajes de grafito, ferrita y perlita, en la aleación. Si se presiona el botón “Clear” se borrarán los elementos de esta sección.

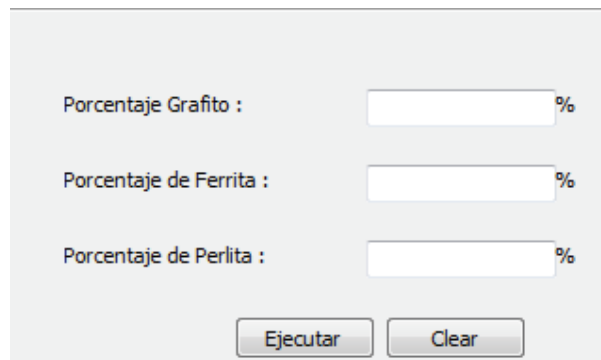


Figura 2.23. Sección de la interfaz gráfica de cálculo de porcentajes de microestructuras.

La sección mostrada en la Figura 2.24, está destinada al análisis de los nódulos de la aleación, después de activarla, es necesario presionar el botón “Barrido Imagen” en donde se realizará la selección de nódulos, una vez terminada la selección los resultados obtenidos se mostrarán en las líneas de texto correspondientes a: Porcentaje de Nodularidad, Número de Nódulos , Tamaño Promedio. Al finalizar el proceso el usuario tiene que seleccionar el tipo de nódulo presente en la metalografía y al presionar en el botón “Tipo de Nódulo” la selección se imprimirá en la línea de texto “Tipo de Grafito”; si se

tiene la presencia de otro tipo de grafito se activara la subsección “Secundario” se seleccionará el tipo secundario de grafito en la metalografía como se muestra en la Figura 2.15, y finalmente se imprimirá la relación porcentual entre las dos tipos de grafito en la línea de texto “Tipo de Grafito”.



Figura 2.24. Sección de la interfaz gráfica de análisis de nódulos.

En el caso que se tenga metalografías con nódulos muy pequeños y en gran cantidad, se activa la subsección “Nódulos Pequeños” en donde se realizará un corte a la imagen de la metalografía original y presionar el botón “Barrido de Imagen” dentro esta subsección se realizará el mismo proceso que el descrito anteriormente, pero con la imagen cortada.

Finalmente se tiene que añadir los campos en la última sección que se indica en la Figura 2.25. En esta sección se añaden campos para el control y constancia de la realización de la metalografía.

Fecha :

Encargado :

Reactivo :

Observaciones :

Figura 2.25. Sección de datos de constancia de la Metalografía.

Al llenar los campos se debe presionar el botón “Añadir a Tabla”, lo que añadirá, valga la redundancia, los valores de los campos de la interfaz a una base de datos en MySQL, en donde los llevará una constancia de los análisis realizados por el programa, para registro y documentación.

3. RESULTADOS Y DISCUSIÓN

3.1. Resultados

Se realizó un análisis empleando el software propuesto, a metalografías presentes en la carta de clasificación de microestructuras de hierro dúctil, la cual podemos observar en la Figura 1.9. (ASF Ductile Iron Quality control Committee, 2010)

Tabla 3.1. Resultado de porcentaje de grafito, ferrita y perlita de las metalografías de la carta de clasificación de microestructuras de hierro dúctil.

Nombre	Porcentaje de Grafito (%)	Porcentaje de Ferrita (%)	Porcentaje de Perlita (%)
caso1	12,63	45,75	45,25
caso2	9,79	32,61	61,32
caso3	11,21	31,43	63,27
caso4	13,54	21,26	71,7

Tabla 3.2 Resultado del análisis de nodularidad a metalografías presentes en la norma: ASTM A247-16.

Nodularidad	Software
nod40.png	43,25
nod50.png	92,63
nod70.jpg	70,95
nod50.jpg	51,65
nod10.png	12,35

En la tabla 3.2 se presenta los resultados del análisis de nodularidad realizado a metalografías presentes en la norma ASTM A247-16. Por otro lado en la Tabla 3.3, se muestran los resultados obtenidos por el programa de metalografías obtenidas mediante experimentación.

Tabla 3.3. Resumen de datos obtenidos de la experimentación.

	Nombre	Valor Grafito	Valor Ferrita	Porcentaje de Grafito %	Porcentaje de Ferrita %	Porcentaje de Perlitita %	Porcentaje de Nodularidad %	Numero de Nódulos (/mm ²)	Tamaño Promedio (mm)
1	CA_AS_E1.5_100X	70	155	35,81	3,45	94,33	24,81	29,46	0,0604
2	CA_AS_E1.6_100X	70	155	31,86	4,76	93,16	30,34	38,76	0,0551
3	CA_AS_E1.8_100X	70	155	30,46	3,61	94,63	47,3	55,04	0,057
4	CA_AS_E1.9_100X	70	155	27,79	7,36	90,9	54,61	55,04	0,0599
5	CA_AS_E1.10_100X	70	155	29,2	3,63	94,66	33,17	41,09	0,0555
6	M1 100X AT	28	109	20,08	40,94	54,43	81,89	273,64	0,0143
7	100X M1.2	58	127	18,79	35,59	63,04	74,31	118,6	0,0196

Tipo de Grafito

1	Tipo II
2	Tipo II
3	Tipo II (84.39%) Tipo I (15.61%)
4	Tipo II (95.44%) Tipo II (4.56%)
5	Tipo II (96.92%) Tipo II (3.08%)
6	Tipo II (89.88%) Tipo VI (10.12%)
7	Tipo II (78.81%) Tipo VI (21.19%)

3.2. Discusión

Para la validación de los resultados obtenidos por el software planteado, se analizó, varias metalografías presentes en la “Carta de Clasificación de Microestructuras de Hierro Dúctil” indicada en la Figura 1.9. En la Tabla 3.4, 3.5 y 3.6 se observa una comparación entre los valores de la carta y lo valores obtenidos por el software.

Para el cálculo del error se emplea la Ecuación 3.1.

$$diferencia = \frac{|v_{referencia} - v_{obtenido}|}{v_{referencia}} \times 100\%$$

Ecuación 3.1. Calculo del error.

Tabla 3.4. Valores empleados para identificar el grafito y la ferrita en la metalografía.

Nombre	Valor escala de Grises 1*	Valor escala de Grises 2**
caso1	38	177
caso2	30	175
caso3	30	175
caso4	31	176

*Valor empleado para identificar el grafito **Valor empleado para identificar la ferrita

Tabla 3.5. Comparación entre los valores de la “Carta de Clasificación de Microestructuras de Hierro Dúctil” y lo valores obtenidos por el software de porcentaje de grafito en las metalografías analizadas.

Nombre	Porcentaje de Grafito (%)		Diferencia	Error (%)
	Programa	Referencia		
caso1	12.63	10	2.63	26.3
caso2	9.79	10	0.21	2.1
caso3	11.21	10	1.21	12.1
caso4	13.54	10	3.54	35.4

(ASF Ductile Iron Quality control Committee, 2010)

Tabla 3.6. . Comparación entre los valores de la “Carta de Clasificación de Microestructuras de Hierro Dúctil” y lo valores obtenidos por el software de porcentaje de ferrita en las metalografías analizadas.

Nombre	Porcentaje de Ferrita (%)		Diferencia	Error (%)
	Programa	Referencia		
caso1	45.75	45	0.75	1.67
caso2	32.61	30	2.61	8.7
caso3	31.43	27	4.43	16.41
caso4	21.26	18	3.26	18.11

(ASF Ductile Iron Quality control Committee, 2010)

Tabla 3.7. Comparación entre los valores de la “Carta de Clasificación de Microestructuras de Hierro Dúctil” y lo valores obtenidos por el software de porcentaje de ferrita en las metalografías analizadas.

Nombre	Porcentaje Perlita (%)		Diferencia	Error (%)
	Programa	Referencia		
caso1	45.25	45	0.25	0.56
caso2	61.32	60	1.32	2.2
caso3	63.27	63	0.27	0.43
caso4	71.7	72	0.3	0.417

(ASF Ductile Iron Quality control Committee, 2010)

3.2.1. Valor de la escala de grises

Es necesario tener muy en cuenta el valor en escala de grises para la identificación de grafito y ferrita, como se muestra en las Tablas 3.8 y 3.9, se tiene valores diferentes de la escala empleada en la identificación de la ferrita, finalmente esto da como resultado porcentajes diferentes de la microestructura perlítica y ferrítica.

Tabla 3.8. Comparación de resultados obtenidos por el programa desarrollado y el empleado por el laboratorio con un valor umbral de ferrita de 200.

	Programa	Laboratorio
Nombre	CA_AS_E1.5_100X	
Valor de escala de grises1*	70	-
Valor de escala de grises2**	200	-
Porcentaje de Grafito	35.81	27.3
Porcentaje de Ferrita	0.06	4.1
Porcentaje de Perlita	97.72	95.9

*Valor empleado para identificar el grafito **Valor empleado para identificar la ferrita

Tabla 3.9. Comparación de resultados obtenidos por el programa desarrollado y el empleado por el laboratorio con un valor umbral de ferrita de 155.

	Programa	Laboratorio
Nombre	CA_AS_E1.5_100X	
Valor de escala de grises1*	70	-
Valor de escala de grises2**	155	-
Porcentaje de Grafito	35.81	27.3
Porcentaje de Ferrita	3.45	4.1
Porcentaje de Perlita	94.33	95.9

*Valor empleado para identificar el grafito **Valor empleado para identificar la ferrita

Al comparar con software propuesto y el empleado en el laboratorio de la facultad, se tiene una diferencia de 4,04 puntos en el porcentaje de ferrita al tener un valor de la escala de grises de 200 en la identificación de la ferrita, mientras que por otro lado si se varía este parámetro hasta un valor de 155, se tiene una diferencia de 0,65 puntos porcentuales.

Del mismo modo si comparamos los porcentajes de perlita en la metalografía cuando se emplea el valor de ferrita de 200 se tiene un valor de 97.72% generándose una diferencia de 1.82 puntos porcentuales con el resultado del programa empleado en el laboratorio y al realizar el análisis con un valor de ferrita de 155 es de 94.33 % produciendo una diferencia de 1.57 puntos porcentuales.

3.2.2. Comparación de porcentajes de grafito ferrita y perlita experimentales

A continuación se presentará varias tablas en las que se muestra los resultados de los porcentajes de perlita, ferrita y grafito del análisis realizado por el software desarrollado y los obtenidos en el laboratorio Stream Essentials.

Tabla 3.10. Valores empleados para identificar el grafito y la ferrita en la metalografía.

Nombre	Valor escala de Grises 1*	Valor escala de Grises 2**
CA_AS_E1.5_100X	70	155
CA_AS_E1.6_100X	70	155
CA_AS_E1.8_100X	70	155
CA_AS_E1.9_100X	70	155
CA_AS_E1.10_100X	70	155
M1 100X AT	28	109
100X M1.2	58	127

*Valor empleado para identificar el grafito **Valor empleado para identificar la ferrita

Tabla 3.11. Comparación entre los valores del porcentaje de grafito obtenidos con el software Stream Essentials y los obtenidos por el software propuesto.

Nombre	Porcentaje de Grafito		Diferencia
	Programa	Referencia	
CA_AS_E1.5_100X	35.81	27.3	8.51
CA_AS_E1.6_100X	31.86	30.8	1.06
CA_AS_E1.8_100X	30.46	22.7	7.76
CA_AS_E1.9_100X	27.79	28.1	0.31
CA_AS_E1.10_100X	29.2	30	0.8
M1 100X AT	20.08	14.6	5.48
100X M1.2	18.79	26	7.21

Tabla 3.12. Comparación entre los valores del porcentaje de ferrita obtenidos con el software Stream Essentials y los obtenidos por el software propuesto.

Nombre	Porcentaje de Ferrita (%)		Diferencia
	Programa	Referencia	
CA_AS_E1.5_100X	3.45	4.1	0.65
CA_AS_E1.6_100X	4.76	6	1.24
CA_AS_E1.8_100X	3.61	9.1	5.49
CA_AS_E1.9_100X	7.36	10.9	3.54
CA_AS_E1.10_100X	3.63	7.7	4.07
M1 100X AT	40.94	47.2	6.26
100X M1.2	35.59	34.6	0.99

Tabla 3.13. Comparación entre los valores del porcentaje de perlita obtenidos con el software Stream Essentials y los obtenidos por el software propuesto.

Nombre	Porcentaje Perlita (%)		Diferencia
	Programa	Referencia	
CA_AS_E1.5_100X	94.33	95.9	1.57
CA_AS_E1.6_100X	93.16	94	0.84
CA_AS_E1.8_100X	94.63	90.9	3.73
CA_AS_E1.9_100X	90.9	89.1	1.8
CA_AS_E1.10_100X	94.66	92.3	2.36
M1 100X AT	54.43	52.8	1.63
100X M1.2	63.04	65.4	2.36

Se tiene una diferencia de resultados pequeña en los porcentajes de perlita con una diferencia máxima de 3.73 puntos porcentuales. Por otro lado diferencias un poco más

amplias en los porcentajes de grafito y ferrita. Esta diferencia se da por la diferencia entre la apreciación del operador al plantear los valores de escala de grises necesarios para identificar el grafito y la ferrita para poder realizar el análisis en ambos softwares, pues trabajan de manera similar. Por otro lado el error presentado en el porcentaje de ferrita, es una acumulación de errores, la apreciación del usuario, la sumatoria de errores matemáticos presentados por el programa, pues al realizar operaciones de división y multiplicación se omiten decimales que finalmente resulta en una falta de precisión en el resultado final; lo que se puede comprobar al sumar los porcentajes de ferrita y perlita tenemos un resultado valores desde 98,63 hasta 95,37 lo que en teoría debería dar 100.

Como se puede ver en la metalografía CA_AS_E1.8_100X es la que presenta un diferencia mayor a las otras, esto se da por una mala apreciación del usuario del programa al definir los valores en escala de grises como se explicó con anterioridad.

3.2.3. Tipo de Grafito

El tipo de grafito que el programa Stream Essentials entrega como resultado del análisis es erróneo debido al grafito libre presente en las metalografías, el programa relaciona a los nódulos y al grafito como un solo cuerpo, por lo que lo asocia con el tipo de concentraciones de grafito presentes en aleaciones como fundiciones grises, como se indica en la Tabla 3.12.

Tabla 3.14. Resumen del tipo de grafito presente en la metalografías.

Nombre	Programa	Laboratorio
CA_AS_E1.5_100X	Tipo II	Tipo III (45%) Tipo IV (32%) Tipo V (23%)
CA_AS_E1.6_100X	Tipo II	Tipo III (42%) Tipo IV (32%) Tipo V (26%)
CA_AS_E1.8_100X	Tipo II (84.39%) Tipo I (15.61%)	Tipo III (37%) Tipo IV (37%) Tipo V (26%)
CA_AS_E1.9_100X	Tipo II (95.44%) Tipo I (4.56%)	Tipo III (40%) Tipo IV (45%) Tipo V (25%)
CA_AS_E1.10_100X	Tipo II (96.92%) Tipo I (3.08%)	Tipo III (32%) Tipo IV (34%) Tipo V (25%)
M1 100X AT	Tipo II (89.88%) Tipo VI (10.12%)	Tipo II (41%) Tipo III (33%) Tipo I (27%)
100X M1.2	Tipo II (78.81%) Tipo VI (21.19%)	Tipo III (66%) Tipo IV (21%) Tipo II (14%)

3.2.4. Análisis de Nodularidad

Para corroborar el análisis de nodularidad, se empleó las metalografías expuestas en la norma ASTM A247-16, en las cuales se muestra como varía la nodularidad porcentualmente, como se indica en la Figura 1.5. , obteniéndose como resultado los valores de la Tabla 3.15

Tabla 3.15. Comparación de porcentajes de nodularidad de metalografías procedentes de la norma ASTM 247-16 y el análisis realizado por el software planteado

Nodularidad	Norma	Software	Diferencia	Error
nod40.png	40	43.25	3.25	8.13
nod50.png	90	92.63	2.63	2.92
nod70.jpg	70	70.95	0.95	1.36
nod50.jpg	50	51.65	1.65	3.30
nod10.png	10	12.35	2.35	23.50

4. CONCLUSIONES

Se desarrolló exitosamente un código de programación que bien aplicado, satisface el análisis metalográfico de una fundición nodular, y además de eso puede ser empleado para obtener los porcentajes de microestructuras en fundiciones de tipos diferentes.

La metodología de análisis se fundamenta en los conceptos y definiciones existentes en la norma ASTM A 247-16 lo cual deja la puerta abierta a la reproducción y futura innovación del código de programación, para futuros proyectos.

Al presentar los resultados en magnitudes más manejables, en comparación con los resultados obtenidos por el software Stream Essentials, se proporciona una herramienta didáctica y práctica. Lo que podría conllevar al uso del código para el aprendizaje relacionado con área del estudio de metalografías destinado a los estudiantes de la Facultad de Ingeniería Mecánica.

Al comparar el resultado obtenidos por el software propuesto y los valores presentes en la *Carta de Clasificación de Microestructuras de Hierro Dúctil* podemos observar que se tiene unas diferencias pequeñas; lo que comprueba la validez del software planteado con una exactitud aceptable.

Al ser un software libre se puede realizar su distribución a estudiantes como docentes involucrados en esta línea de investigación proporcionando una herramienta o un punto de partida para la realización de trabajos futuros en los cuales sea necesario la aplicación de procesamiento de imágenes.

El código python puede ser convertido a un formato .web de tal manera que puede convertirse en una aplicación para futuros proyectos y ser trabajado en línea

Las diferencias presentadas con el software Stream Essentials, están relacionadas con la diferencia de apreciación del operario del software. Además el software Stream Essentials, al parecer encuentra un porcentaje y realiza la resta del 100%.

La nodularidad obtenida por el software propuesto, es muy acorde con la nodularidad indicada en la norma ASTM A247-16. Pero el análisis de la nodularidad, se ve afectada al analizar una aleación atacada químicamente, en la búsqueda de observar la microestructura. En una metalografía atacada se visualizan zonas oscuras

correspondientes a la perlita y otras correspondientes al grafito libre presente en la aleación. Por tal motivo es necesario identificar correctamente el grafito al plantear el valor umbral de la escala de grises.

Al considerar al grafito libre dentro del análisis de la nodularidad se realiza un análisis más acorde con la definición de la norma ASTM A247-16, en la cual indica que la nodularidad es la medida en porcentaje de cuantas partículas de grafito se concentraron en los nódulos de la aleación respecto al grafito presente en la aleación. (ASTM, A247-16, 2016)

Referencias Bibliográficas

- American Society for Metals. (1 de Septiembre de 2005). ASM Handbook, Volume 1. *Properties and Selection: Irons, Steels, and High Performance Alloys*. Estados Unidos.
- ASF Ductile Iron Quality control Committee. (2010). *Ductile Iron Microstructures Rating Chart*.
- ASTM- E407-07. (2015). *Standard Practice for Microetching Metals and Alloys*1. Estados Unidos: ASTM.
- ASTM INTERNATIONAL. (1 de Mayo de 2011). E3-11. *Standar Guide for Preparation of Metallographic Specimens*. United States.
- ASTM, A247-16. (Abril de 2016). *Standard Test Method for Evaluating the Microstructure of Graphite in Iron Castings*. Estados Unidos.
- Avner, S. H. (1988). *Introduccion a la Metalurgia Física*. Mexico: McGraw- Hill.
- Barreiro, J. A. (1963). *Fundiciones* (2da. Edición ed.). Madrid: Dossat.
- Callister, W. D. (1990). *Introduccion a la Ciencia e Ingeniería de los Materiales*. Barcelona: Reverté.
- Donald R. Askeland, P. P. (2003). *Ciencia e Ingeniería de los Materiales* (4ta ed.). Mexico: Thomson.
- Gomez, C. S. (14 de Diciembre de 2012). *EAF*. Obtenido de Diagrama Hierro-Carbono: <http://carlossaiz.blogspot.com/2012/12/diagrama-hierro-carbono.html>
- Hidalgo, V. H. (Junio de 2016). Numerical study on unsteady cavitating flow and erosion based on homogeneous mixture assumption. Pekin, China: Tsinghua University.
- Metalografía de la Fundición* . (s.f.). Obtenido de Fundición Nodular: <http://matensayos.webcindario.com/metalofundicion/fundinodular/nodular.htm>
- OpenCV-Python. (Marzo de 2016). *OpenCV-Python Tutorials Documentaon*. Obtenido de https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html
- Python . (27 de Marzo de 2017). *The Python Tutorial*. Obtenido de Python Documentation: <https://docs.python.org/2/tutorial/>
- Richard G. Budynas, J. N. (2008). *Diseño en Ingeniería Mecánica de Shigley* (8va ed.). Mexico: McGrawHill.
- Ruiz Ortiz, R. (s.f.). *Ciencia de Materiales I*. Quito: Escuela Politécnica Nacional.
- Smith, W. F., & Hashemi, J. (2006). *Fundamentos de la Ciencia e Ingeniería de Materiales* (4th Edición ed.). Mexico D.F.: McGrawHill.
- The Matplotlib development team. (02 de Febrero de 2017). *Matplotlib*. Obtenido de <https://matplotlib.org/>
- Willems, K. (18 de Enero de 2017). *Learn to be a Data Scientist. Start your free account with DataCamp*. Obtenido de Python Numpy Array Tutorial: <https://www.datacamp.com/community/tutorials/python-numpy-tutorial#gs.R7kkNE4>

ANEXOS

ANEXO I. DIAGRAMA DE HIERRO CARBONO.

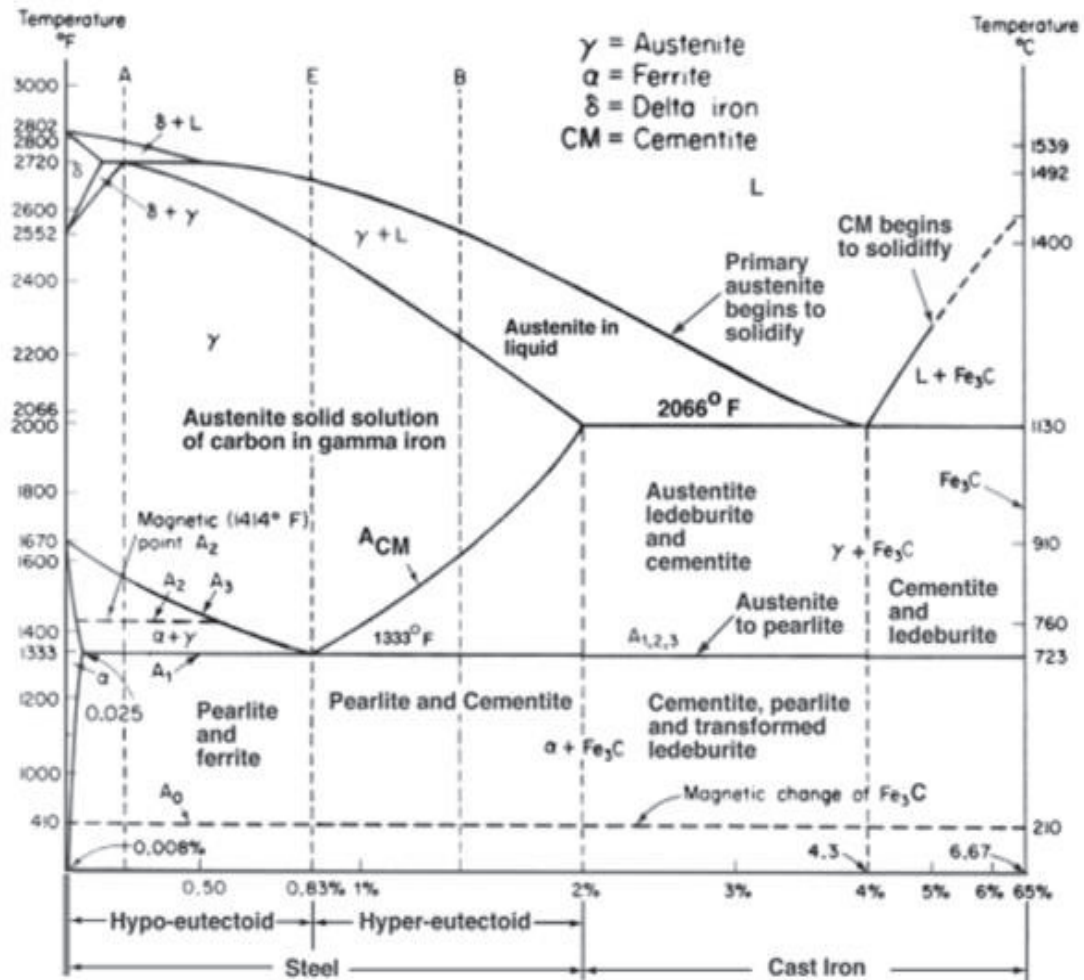


Figura I.1. Diagrama de Equilibrio Hierro Carbono.

(Gomez, 2012)

ANEXO II. RELACIÓN ENTRE EL CONTENIDO DE CARBONO Y SILICIO.

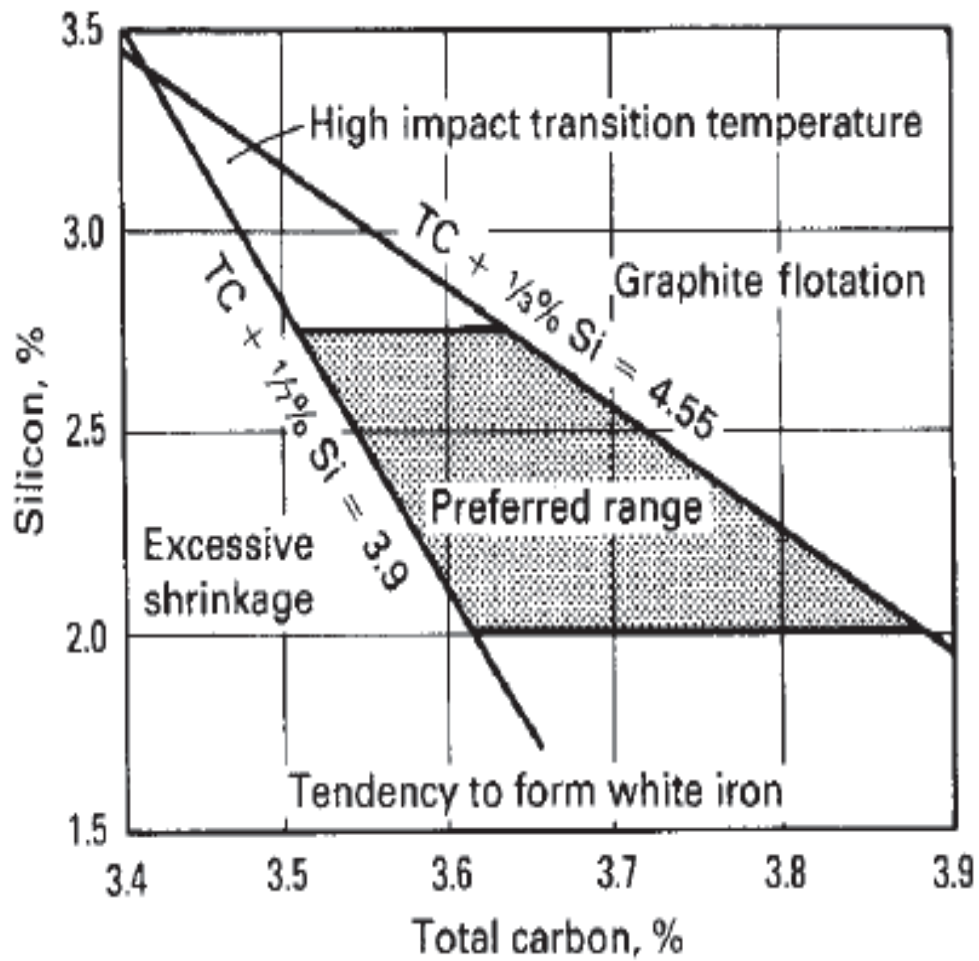


Figura I.2. Composición de Silicio y Carbono ideales en un Hierro Dúctil

(American Society for Metals, 2005)

ANEXO III. CODIGO DEL SOFTWARE PLANTEADO.

```
import sys

import matplotlib.pyplot as plt

from NodularProcesamiento13 import *

import cv2

import numpy as np

import math

import MySQLdb

from PIL import Image

class MiFormulario (QtGui.QMainWindow):

    grises = False

    def __init__(self, parent=None):

        QtGui.QWidget.__init__(self, parent)

        self.ui =Ui_MainWindow()

        self.ui.setupUi(self)

        self.ui.horizontalSlider.valueChanged.connect (self.cambio)

        self.ui.horizontalSlider_2.valueChanged.connect (self.cambio2)

        QtCore.QObject.connect(self.ui.btnEjecutar,QtCore.SIGNAL('clicked()'),self.Ejecutar)

        QtCore.QObject.connect(self.ui.btnCargarImg,QtCore.SIGNAL('clicked()'),self.CargarImagen)

        QtCore.QObject.connect(self.ui.btnFerr,QtCore.SIGNAL('clicked()'),self.CargarFer)

        QtCore.QObject.connect(self.ui.btnBarrido,QtCore.SIGNAL('clicked()'),self.Barrido)

        QtCore.QObject.connect(self.ui.btnCorte,QtCore.SIGNAL('clicked()'),self.Corte)

        QtCore.QObject.connect(self.ui.btnBarrido2,QtCore.SIGNAL('clicked()'),self.Barrido2)

        QtCore.QObject.connect(self.ui.btnTipo,QtCore.SIGNAL('clicked()'),self.NoduloTipo)
```

```
QtCore.QObject.connect(self.ui.btnSecundario,QtCore.SIGNAL('clicked()'),self.TipoSecundario)
QtCore.QObject.connect(self.ui.btnAnadirTabla,QtCore.SIGNAL('clicked()'),self.AnadirTabla)
```

```
def cambio (self, value):
```

```
    global grises
```

```
    self.ui.lineGrises.setText(str(value))
```

```
    if grises == True:
```

```
        var = int (self.ui.lineGrises.text())
```

```
        img= str(self.ui.lineInserte.text())
```

```
        i1 = cv2.imread(img, 0)
```

```
        p, q = i1.shape
```

```
        i2 = np.ones((p,q), np.uint8)
```

```
        i1_clean = 255*i2
```

```
        for i in xrange(p):
```

```
            for j in xrange(q):
```

```
                if 0 < i1[i,j] < var:
```

```
                    i1_clean[i,j] = i1[i,j]
```

```
                else:
```

```
                    i1_clean[i,j] = 200
```

```
        while(1):
```

```
            #cv2.imshow('imagen',i1)
```

```
            cv2.namedWindow('grises', cv2.WINDOW_NORMAL)
```

```
cv2.imshow('grises', i1_clean)
k = cv2.waitKey(1) & 0xFF
if k == 27:
    grises = False
    break

cv2.destroyAllWindows()
```

```
def cambio2 (self,value):
    global ferrita
    self.ui.lineFe.setText(str(value))

if ferrita == True:
    var2 = int (self.ui.lineFe.text())

    img2 = str(self.ui.lineInserte.text())

    ig1 = cv2.imread(img2, 0)
    r , s = ig1.shape
    ig2 = np.ones((r,s), np.uint8)
    ig1_clean = 255*ig2

    for i in xrange(r):
        for j in xrange(s):
            if var2 < ig1[i,j] < 255:
                ig1_clean[i,j] = ig1[i,j]
            else:
```

```
ig1_clean[i,j] = 30
```

```
while(1):
```

```
    #cv2.imshow('imagen',i1)
```

```
    cv2.namedWindow('ferrita', cv2.WINDOW_NORMAL)
```

```
    cv2.imshow('ferrita', ig1_clean)
```

```
    k = cv2.waitKey(1) & 0xFF
```

```
    if k == 27:
```

```
        ferrita = False
```

```
        break
```

```
cv2.destroyAllWindows()
```

```
def CargarImagen (self):
```

```
    global grises
```

```
    grises = True
```

```
    return grises
```

```
def CargarFer (self):
```

```
    global ferrita
```

```
    ferrita = True
```

```
    return ferrita
```

```
def Ejecutar (self):
```

```

img = str(self.ui.lineInserte.text())
a = int(self.ui.lineGrises.text())
b = int (self.ui.lineFe.text())

print a
print b
print img
print type (a)
print type (img)

i1 = cv2.imread(img, 0)

li, co = i1.shape

TotalPix = li*co

i1_clean = np.zeros((li,co))
i2_clean = np.zeros((li,co))
i3_clean = np.zeros((li,co))

i1_pix = []
i2_pix = []
i3_pix = []

for i in xrange(li):
    for j in xrange(co):
        if 0 < i1[i,j] < a:
            i1_clean[i,j] = i1[i,j]
            i1_pix.append(i1[i,j]) #To define an array with elements, which contained nodulars
forms.
        else:

```

```
i1_clean[i,j] = 200
```

```
for i in xrange(li):  
    for j in xrange(co):  
        if b < i1[i,j] < 255:  
            i2_clean[i,i] = i1[i,j]  
            i2_pix.append(i1[i,j]) #To define an array with ferrita.  
        else:  
            i2_clean[i,j] = 10
```

```
for i in xrange(li):  
    for j in xrange(co):  
        if a < i1[i,j] < b:  
            i3_clean[i,i] = i1[i,j]  
            i3_pix.append(i1[i,j]) #To define an array with perlita.  
        else:  
            i3_clean[i,j] = 200
```

```
def percT(ne, nT):  
    nT = nT + 0.0  
    pe = ne/nT  
    return pe
```

```
#Calculo de porcentaje de grafito, ferrita y perlita
```

```

TotalPix2 = (TotalPix - len(i1_pix))
print 'graffito' + str(len(i1_pix))
print 'ferrita' + str(len(i2_pix))
print 'perlita' + str(len(i3_pix))

graff, fer, per = percT(len(i1_pix), TotalPix), percT(len(i2_pix), TotalPix2), percT(len(i3_pix),
TotalPix2)

GRAFF1 = graff*100
FER1 = fer*100
PER1 = per*100

GRAFF = '{0:.2f}'.format(GRAFF1)
FER = '{0:.2f}'.format(FER1)
PER = '{0:.2f}'.format(PER1)

self.ui.lineGraff.setText(str(GRAFF))
self.ui.lineFer.setText(str(FER))
self.ui.linePer.setText(str(PER))

print (GRAFF, FER, PER)

plt.subplot(1,2,1)
plt.imshow(i1, cmap=plt.cm.gray)
plt.xticks([], plt.yticks([]))

plt.subplot(1,2,2)

```

```

plt.imshow(i1_clean, cmap=plt.cm.gray)
plt.xticks([], plt.yticks([]))

print ("imageSize = ", i1.size)
print ("ImageShape = ", i1.shape, i1_clean.shape, i2_clean.shape, i3_clean.shape)
plt.show()

def Barrido (self):
    global cont, img_points, SumaAreas, img, img2, img1, i1_pix, inod_pix, inod_clean, va,
    i1_clean, SumatoriaDiametros, tipo

    imagen = str(self.ui.lineInserte.text())
    va = int (self.ui.lineGrises.text())
    tipo = 0

    drawing = False # true if mouse is pressed
    mode = True # if True, draw rectangle. Press 'c' to toggle to curve

    img_points = []
    SumaAreas = []
    SumatoriaDiametros = []
    cont = 0
    ix, iy = -1, -1

    img= cv2.imread(imagen, 0)
    a, b = img.shape

```



```

i2 = np.ones((a,b), np.uint8)
img2 = 255* i2

i1_pix = []
inod_pix = []

inod_clean = np.zeros((a,b))
i1_clean = np.zeros((a,b))

def dibujar_circulos(event,x,y,flags,param):
    global img_points,drawing,mode, cont, dst

    if event == cv2.EVENT_LBUTTONDOWN:
        drawing = True
        img_points.append((x,y))
        cont = cont + 1

    print img_points

    p1 = img_points[0]
    px = p1[0]
    py = p1[1]
    print p1
    print px, py
    print cont

```

```

if cont == 2:

    q1 = img_points[1]
    qx = q1[0]
    qy = q1[1]
    cont = 0

    print q1
    print qx, qy

    cx = (qx + px)/2
    cy = (qy + py)/2

    diametro = math.sqrt(((qx - px)**2) + ((qy - py)**2))
    SumatoriaDiametros.append (diametro)

    radio = int (diametro/2)

    area = (math.pi)*diametro
    SumaAreas.append(area)

    print cx, cy, radio

    cv2.circle(img,(cx,cy),radio,(0,255,0),5)
    cv2.circle(img2,(cx,cy),radio,0,-1)

    img_points = []

```

```
elif event == cv2.EVENT_LBUTTONDOWN:
```

```
    drawing = False
```

```
cv2.namedWindow('imagen',cv2.WINDOW_NORMAL)
```

```
cv2.setMouseCallback('imagen',dibujar_circulos)
```

```
while(1):
```

```
    cv2.imshow('imagen',img)
```

```
    k = cv2.waitKey(1) & 0xFF
```

```
    if k == ord('s'):
```

```
        imag = str(self.ui.lineEdit.text())
```

```
        img1 = cv2.imread(imag,0)
```

```
        c, d = img1.shape
```

```
        i3 = np.ones((c,d), np.uint8)
```

```
        img3 = 255*i3
```

```
        ret, mask = cv2.threshold (img2,80,255,cv2.THRESH_BINARY_INV)
```

```
        mask_inv = cv2.bitwise_not (mask)
```

```
        img1_atras = cv2.bitwise_or(img1, img1,mask = mask)
```

```
        img2_detras = cv2.bitwise_or(img3,img3,mask = mask_inv)
```

```

dst = cv2.add(img1_atras, img2_detras)
cv2.namedWindow('dst', cv2.WINDOW_NORMAL)
cv2.imshow('dst',dst)
cv2.imwrite(str(self.ui.lineEdit.text())+'_Nodulos.jpg',dst)

```

```

for i in xrange(c):
    for j in xrange(d):
        if 0 < img1[i,j] < va:
            i1_clean[i,j] = img1[i,j]
            i1_pix.append(img1[i,j]) #To define an array with elements, which contained
nodulars forms.
        else:
            i1_clean[i,j] = 200

```

```

for i in xrange(c):
    for j in xrange (d):
        if 0 < dst[i,j] < 200:
            inod_clean[i,j] = dst[i,j]
            inod_pix.append(dst[i,j])
        else:
            inod_clean[i,j] = 255

```

```

sum=0
for i in range(0,len(SumatoriaDiametros)):
    sum=sum+SumatoriaDiametros[i]

```

```

r = len (SumatoriaDiametros)

```

```

promedio1 = sum/r
promedio2 = promedio1 * 6.4e-4
promedio = '{0:.4f}'.format (promedio2)

print 'El promedio es el siguiente'
print promedio

particulas = len (inod_pix)

print 'El numero de particulas es ..... '
print particulas

self.ui.lineTamano.setText(str(promedio))

def Porcentaje(ne,nT):
    nT = nT + 0.0
    pe = ne/nT
    return pe

nodulos = len (inod_pix)
grafito = len (i1_pix)

nod = Porcentaje(nodulos,grafito)
NOD1 = nod*100

NOD = '{0:.2f}'.format (NOD1)

```

```
print 'pilas!!!!'  
print nodulos, grafito  
print nod, NOD  
  
numero = len (SumaAreas)  
  
self.ui.lineNod.setText(str(NOD))  
  
numero_nod1 = numero/1.29  
numero_nod = '{0:.2f}'.format (numero_nod1)  
  
self.ui.lineNumero.setText(str(numero_nod))
```

```
elif k == 27:  
    break
```

```
cv2.destroyAllWindows()
```

```
def Corte (self):  
    global draw1, mod1, x1, y1, x2, y2, contador, corte_points  
    picture = str(self.ui.lineInserte.text())  
    pic1 = cv2.imread (picture,1)  
    corte_points = []  
  
    draw1 =False  
    mod1 = True  
    xi, yi = -1, -1  
    contador = 0
```

```

def seleccionar_area (event,x,y,flags,param):
    global x1, y1,x2,y2, draw1, contador

    if event == cv2.EVENT_LBUTTONDOWN:
        draw1 == True
        xi, yi = x,y
        corte_points.append ((x,y))
        contador = contador + 1
        print corte_points

        p1 = corte_points [0]
        x1 = p1[0]
        y1 = p1[1]
        print p1
        print x, y
        print contador

    if contador == 2:
        p2 = corte_points [1]
        x2 = p2 [0]
        y2 = p2 [1]

        cv2.rectangle(pic1,(x1,y1),(x2,y2),(0,0,0),2)

        imagen = str(self.ui.lineInserte.text())
        recorte = Image.open(imagen)

        box = (x1, y1, x2, y2)

```

```

        region = recorte.crop(box)

        region.show()

        region.size

        region.save(str(self.ui.lineInserte.text())+'_recorte.jpg')

        self.ui.lineCorte.setText(str(self.ui.lineInserte.text())+'_recorte.jpg')

elif event == cv2.EVENT_LBUTTONDOWN:
    draw1 == False

cv2.namedWindow('Recorte', cv2.WINDOW_NORMAL)
cv2.setMouseCallback('Recorte',seleccionar_area)

while (1):
    cv2.imshow('Recorte',pic1)
    k = cv2.waitKey(1) & 0xFF
    if k == 27:

        break

cv2.destroyAllWindows()

##     if k == ord('c'):
##
##         imagen = str(self.ui.lineInserte.text())

```



```

def Barrido2 (self):

    global cont,img_points, SumaAreas, img, img2, img1, i1_pix, inod_pix, inod_clean, va,
    i1_clean, SumatoriaDiametros, tipo

    imagen = str(self.ui.lineCorte.text())

    va = int (self.ui.lineGrises.text())

    tipo = 0

    drawing = False # true if mouse is pressed
    mode = True # if True, draw rectangle. Press 'c' to toggle to curve

    img_points = []
    SumaAreas = []
    SumatoriaDiametros = []
    cont = 0
    ix, iy = -1, -1

    img= cv2.imread(imagen, 0)
    a, b = img.shape
    i2 = np.ones((a,b), np.uint8)
    img2 = 255* i2

    i1_pix = []
    inod_pix = []

    inod_clean = np.zeros((a,b))
    i1_clean = np.zeros((a,b))

```

```
def dibujar_circulos(event,x,y,flags,param):  
    global img_points,drawing,mode, cont, dst  
  
    if event == cv2.EVENT_LBUTTONDOWN:  
        drawing = True  
        img_points.append((x,y))  
        cont = cont + 1  
  
        print img_points  
  
        p1 = img_points[0]  
        px = p1[0]  
        py = p1[1]  
        print p1  
        print px, py  
        print cont  
  
        if cont == 2:  
            q1 = img_points[1]  
            qx = q1[0]  
            qy = q1[1]  
            cont = 0
```

```

print q1
print qx, qy

cx = (qx + px)/2
cy = (qy + py)/2

diametro = math.sqrt(((qx - px)**2) + ((qy - py)**2))
SumatoriaDiametros.append (diametro)

radio = int (diametro/2)

area = (math.pi)*diametro
SumaAreas.append(area)

print cx, cy, radio

cv2.circle(img,(cx,cy),radio,(0,255,0),5)
cv2.circle(img2,(cx,cy),radio,0,-1)

img_points = []

elif event == cv2.EVENT_LBUTTONDOWN:
    drawing = False

cv2.namedWindow('imagen',cv2.WINDOW_NORMAL)

cv2.setMouseCallback('imagen',dibujar_circulos)

```

```

while(1):

    cv2.imshow('imagen',img)

    k = cv2.waitKey(1) & 0xFF

    if k == ord('s'):

        imag = str(self.ui.lineCorte.text())

        img1 = cv2.imread(imag,0)
        c, d = img1.shape
        i3 = np.ones ((c,d), np.uint8)
        img3 = 255*i3

        ret, mask = cv2.threshold (img2,80,255,cv2.THRESH_BINARY_INV)

        mask_inv = cv2.bitwise_not (mask)
        img1_atras = cv2.bitwise_or(img1, img1,mask = mask)
        img2_detras = cv2.bitwise_or(img3,img3,mask = mask_inv)

        dst = cv2.add(img1_atras, img2_detras)
        cv2.namedWindow('dst', cv2.WINDOW_NORMAL)
        cv2.imshow('dst',dst)
        cv2.imwrite(str(self.ui.lineInserte.text())+'_Nodulos.jpg',dst)

    for i in xrange(c):
        for j in xrange(d):

```

```

    if 0 < img1[i,j] < va:
        i1_clean[i,j] = img1[i,j]
        i1_pix.append(img1[i,j]) #To define an array with elements, which contained
nodulars forms.
    else:
        i1_clean[i,j] = 200

for i in xrange(c):
    for j in xrange (d):
        if 0 < dst[i,j] < 200:
            inod_clean[i,j] = dst[i,j]
            inod_pix.append(dst[i,j])
        else:
            inod_clean[i,j] = 255

sum=0
for i in range(0,len(SumatoriaDiametros)):
    sum=sum+SumatoriaDiametros[i]

r = len (SumatoriaDiametros)

promedio1 = sum/r
promedio2 = promedio1 * 6.4e-4
promedio = '{0:.4f}'.format (promedio2)

print 'El promedio es el siguiente'
print promedio

particulas = len (inod_pix)

```

```
print 'El numero de particulas es .....'  
print particulas  
  
self.ui.lineTamano.setText(str(promedio))
```

```
def Porcentaje(ne,nT):  
    nT = nT + 0.0  
    pe = ne/nT  
    return pe  
  
nodulos = len (inod_pix)  
grafito = len (i1_pix)  
  
nod = Porcentaje(nodulos,grafito)  
NOD1 = nod*100  
  
NOD = '{0:.2f}'.format (NOD1)  
  
print 'pilas!!!!'  
print nodulos, grafito  
print nod, NOD  
  
numero = len (SumaAreas)  
  
self.ui.lineNod.setText(str(NOD))
```

```
numero_nod1 = numero/1.29

numero_nod = '{0:.2f}'.format (numero_nod1)

self.ui.lineNumero.setText(str(numero_nod))
```

```
elif k == 27:
    break
```

```
cv2.destroyAllWindows()
```

```
def NoduloTipo (self):
    global dst, tipo
    tipo = 0
    #nodulos = cv2.imread('nodulos.jpg',0)

    if self.ui.radioBtnTipo1.isChecked() == True:
        tipo = 'Tipo I'
    if self.ui.radioBtnTipo2.isChecked() == True:
        tipo = 'Tipo II'
    if self.ui.radioBtnTipo5.isChecked() == True:
        tipo = 'Tipo V'
```

```

if self.ui.radioBtnTipo6.isChecked() == True:
    tipo = 'Tipo VI'

self.ui.lineTipo.setText(str(tipo))

def TipoSecundario(self):
    global dst, draw, mod, tipo, tipo2, nodA, nodA_points1, nodA_points2, inodA, nod1_pix,
    nod2_pix, a, b, SecdTipo, PrinTipo

    draw = False
    mod = True
    self.ui.lineTipo.setText(str( ))

    PrinTipo = 0
    SecdTipo = 0

    nodA= cv2.imread(str(self.ui.lineInserte.text())+'_Nodulos.jpg',0)

    nodA_points1 = []
    nodA_points2 = []

    iix, iiy = -1, -1
    a, b = nodA.shape
    inod = np.ones((a,b),np.uint8)
    inodA= 255*inod

    nod1_pix = []
    nod2_pix = []

```



```

inod1_clean = np.zeros((a,b))
inod2_clean = np.zeros((a,b))

def capturar_nodulos (event,x,y,flags,param):
    global iix, iiy, draw, mod

    if event == cv2.EVENT_LBUTTONDOWN:
        draw == True
        iix, iiy = x,y
        print x, y
        print iix, iiy

    elif event == cv2.EVENT_MOUSEMOVE:
        if draw == True:
            if mod ==True:
                cv2.rectangle(nodA,(iix,iiy),(x,y),(0,0,0),2)
                cv2.rectangle(inodA,(iix,iiy),(x,y),(0,0,0),-1)

    elif event == cv2.EVENT_LBUTTONUP:
        draw == False
        if mod == True:
            cv2.rectangle(nodA,(iix,iiy),(x,y),(0,0,0),2)
            cv2.rectangle(inodA,(iix,iiy),(x,y),(0,0,0),-1)

cv2.namedWindow('Tipo', cv2.WINDOW_NORMAL)
cv2.setMouseCallback('Tipo',capturar_nodulos)

while(1):
    cv2.imshow('Tipo',nodA)

```

```

k = cv2.waitKey(1) & 0xFF
if k == ord('n'):

    imNod= cv2.imread(str(self.ui.lineEdit.text())+'_Nodulos.jpg',0)
    c, d = imNod.shape

    i3 = np.ones ((a,b), np.uint8)
    img3 = 255*i3

    ret, mask = cv2.threshold (inodA,80,255,cv2.THRESH_BINARY_INV)

    mask_inv = cv2.bitwise_not (mask)
    img1_atras = cv2.bitwise_or(imNod, imNod,mask = mask)
    img2_detras = cv2.bitwise_or(img3,img3,mask= mask_inv)

    dt = cv2.add(img1_atras, img2_detras)

    cv2.namedWindow('tipoS', cv2.WINDOW_NORMAL)
    cv2.imshow('tipoS',dt)

    #cv2.imwrite('nodulos.jpg',dst)

    for i in xrange(a):
        for j in xrange (b):
            if 0 < nodA[i,j] < 200:
                inod1_clean[i,j] = nodA[i,j]
                nod1_pix.append(dt[i,j])
            else:

```

```
inod1_clean[i,j] = 255
```

```
for i in xrange(a):  
    for j in xrange (b):  
        if 0 < dt[i,j] < 200:  
            inod2_clean[i,j] = dt[i,j]  
            nod2_pix.append(dt[i,j])  
        else:  
            inod2_clean[i,j] = 255
```

```
def Porcentajes (ne,nT):  
    nT = nT + 0.0  
    pe = ne/nT  
    return pe
```

```
total = len (nod1_pix)
```

```
tipo2 = len (nod2_pix)
```

```
tipo1 = total - tipo2
```

```
Tipo1 = Porcentajes (tipo1,total)
```

```
T1 = Tipo1 * 100
```

```
t1 = '{0:.2f}'.format (T1)
```

```
Tipo2 = Porcentajes (tipo2,total)
```

```
T2 = Tipo2 * 100
```

```
t2 = '{0:.2f}'.format (T2)
```

```
print T1, T2
```

```
if self.ui.radioBtnTipo1.isChecked() == True:
```

```
    PrinTip = 'Tipo I'
```

```
elif self.ui.radioBtnTipo2.isChecked() == True:
```

```
    PrinTip = 'Tipo II'
```

```
elif self.ui.radioBtnTipo5.isChecked() == True:
```

```
    PrinTip = 'Tipo V'
```

```
elif self.ui.radioBtnTipo6.isChecked() == True:
```

```
    PrinTip = 'Tipo VI'
```

```
if self.ui.radioBtnTipo1s.isChecked() == True:
```

```
    SecdTipo = 'Tipo I'
```

```
elif self.ui.radioBtnTipo2s.isChecked() == True:
```

```
    SecdTipo = 'Tipo II'
```

```
elif self.ui.radioBtnTipo5s.isChecked() == True:
```

```
    SecdTipo = 'Tipo V'
```

```
elif self.ui.radioBtnTipo6s.isChecked() == True:
```

```
    SecdTipo = 'Tipo VI'
```

```
self.ui.lineTipo.setText(str(PrinTip) + str(t1) + str(SecdTipo) + str(t2))
```

```
elif k==27:
```

```
    break
```

```
cv2.destroyAllWindows()
```

```
def AnadirTabla(self):
```

```
    img = str (self.ui.lineInserte.text())
```

```
    gr = int (self.ui.lineGrises.text())
```

```
    a= float(self.ui.lineGraff.text())
```

```
    b= float(self.ui.lineFer.text())
```

```
    c= float(self.ui.linePer.text())
```

```
    d= float(self.ui.lineNod.text())
```

```
    e= float(self.ui.lineNumero.text())
```

```
    f= float(self.ui.lineTamano.text())
```

```
    g= str(self.ui.lineTipo.text())
```

```
    h= str(self.ui.lineFecha.text())
```

```
    i= str(self.ui.lineEncargado.text())
```

```
    j= str(self.ui.lineReactivo.text())
```

```
    k= str(self.ui.lineObservaciones.text())
```

```
    conn = MySQLdb.connect(host="localhost", user = "root", passwd = "password", db=
"metalografias")
```

```
    cursor = conn.cursor ()
```

```
    cursor.execute("""
```

```
INSERT INTO analisis_fundiciones (Nombre, Valor_Grises, Porcentaje_Grafito,  
Porcentaje_Ferrita, Porcentaje_Perlita, Porcentaje_Nodularidad, Numero_Nodulos,  
Tamano_Promedio, Tipo_Grafito, Fecha, Encargado, Reactivo, Observaciones)
```

```
VALUES (%s, %d, %f, %f, %f, %f, %f, %f, %s, %s, %s, %s, %s)
```

```
"" %(img,gr,a,b,c,d,e,f,g,h,i,j,k)
```

```
print ('Una fila insertada en la tabla analisis_fundiciones')
```

```
cursor.close()
```

```
conn.commit()
```

```
conn.close()
```

```
if __name__=="__main__":
```

```
    app=QtGui.QApplication(sys.argv)
```

```
    myapp=Miformulario()
```

```
    myapp.show()
```

```
    sys.exit(app.exec_())
```

ANEXO IV. CODIGO DE LA INTERFAZ GRÁFICA.

```
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'NodularProcesamiento13.ui'
#
# Created by: PyQt4 UI code generator 4.11.4
#
# WARNING! All changes made in this file will be lost!

from PyQt4 import QtCore, QtGui

try:
    _fromUtf8 = QtCore.QString.fromUtf8
except AttributeError:
    def _fromUtf8(s):
        return s

try:
    _encoding = QtGui.QApplication.UnicodeUTF8
    def _translate(context, text, disambig):
        return QtGui.QApplication.translate(context, text, disambig, _encoding)
except AttributeError:
    def _translate(context, text, disambig):
        return QtGui.QApplication.translate(context, text, disambig)

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
```

```
MainWindow.setObjectName(_fromUtf8("MainWindow"))
MainWindow.resize(778, 709)
self.centralwidget = QtGui.QWidget(MainWindow)
self.centralwidget.setObjectName(_fromUtf8("centralwidget"))
self.label = QtGui.QLabel(self.centralwidget)
self.label.setGeometry(QtCore.QRect(70, 30, 154, 16))
self.label.setObjectName(_fromUtf8("label"))
self.label_2 = QtGui.QLabel(self.centralwidget)
self.label_2.setGeometry(QtCore.QRect(70, 70, 161, 16))
self.label_2.setObjectName(_fromUtf8("label_2"))
self.label_3 = QtGui.QLabel(self.centralwidget)
self.label_3.setGeometry(QtCore.QRect(30, 330, 161, 16))
self.label_3.setObjectName(_fromUtf8("label_3"))
self.label_4 = QtGui.QLabel(self.centralwidget)
self.label_4.setGeometry(QtCore.QRect(30, 370, 161, 16))
self.label_4.setObjectName(_fromUtf8("label_4"))
self.textBrowser = QtGui.QTextBrowser(self.centralwidget)
self.textBrowser.setEnabled(True)
self.textBrowser.setGeometry(QtCore.QRect(390, 70, 281, 31))
self.textBrowser.setObjectName(_fromUtf8("textBrowser"))
self.lineGrises = QtGui.QLineEdit(self.centralwidget)
self.lineGrises.setGeometry(QtCore.QRect(240, 70, 131, 20))
self.lineGrises.setObjectName(_fromUtf8("lineGrises"))
self.lineGraff = QtGui.QLineEdit(self.centralwidget)
self.lineGraff.setGeometry(QtCore.QRect(190, 330, 101, 20))
self.lineGraff.setObjectName(_fromUtf8("lineGraff"))
self.lineFer = QtGui.QLineEdit(self.centralwidget)
self.lineFer.setGeometry(QtCore.QRect(190, 370, 101, 20))
self.lineFer.setObjectName(_fromUtf8("lineFer"))
self.btnEjecutar = QtGui.QPushButton(self.centralwidget)
```



```
self.btnEjecutar.setGeometry(QtCore.QRect(120, 450, 75, 23))
self.btnEjecutar.setObjectName(_fromUtf8("btnEjecutar"))
self.linePer = QtGui.QLineEdit(self.centralwidget)
self.linePer.setGeometry(QtCore.QRect(190, 410, 101, 20))
self.linePer.setObjectName(_fromUtf8("linePer"))
self.label_5 = QtGui.QLabel(self.centralwidget)
self.label_5.setGeometry(QtCore.QRect(30, 410, 161, 16))
self.label_5.setObjectName(_fromUtf8("label_5"))
self.lineInserte = QtGui.QLineEdit(self.centralwidget)
self.lineInserte.setGeometry(QtCore.QRect(240, 30, 131, 20))
self.lineInserte.setObjectName(_fromUtf8("lineInserte"))
self.btnCargarImg = QtGui.QPushButton(self.centralwidget)
self.btnCargarImg.setGeometry(QtCore.QRect(390, 20, 91, 23))
self.btnCargarImg.setCursor(QtGui.QCursor(QtCore.Qt.ArrowCursor))
self.btnCargarImg.setObjectName(_fromUtf8("btnCargarImg"))
self.line = QtGui.QFrame(self.centralwidget)
self.line.setGeometry(QtCore.QRect(0, 280, 761, 16))
self.line setFrameShape(QtGui.QFrame.HLine)
self.line setFrameShadow(QtGui.QFrame.Sunken)
self.line.setObjectName(_fromUtf8("line"))
self.btnAnadirTabla = QtGui.QPushButton(self.centralwidget)
self.btnAnadirTabla.setGeometry(QtCore.QRect(80, 641, 91, 23))
self.btnAnadirTabla.setObjectName(_fromUtf8("btnAnadirTabla"))
self.lineFecha = QtGui.QLineEdit(self.centralwidget)
self.lineFecha.setGeometry(QtCore.QRect(120, 491, 181, 20))
self.lineFecha.setObjectName(_fromUtf8("lineFecha"))
self.lineEncargado = QtGui.QLineEdit(self.centralwidget)
self.lineEncargado.setGeometry(QtCore.QRect(120, 521, 181, 20))
self.lineEncargado.setObjectName(_fromUtf8("lineEncargado"))
self.line_2 = QtGui.QFrame(self.centralwidget)
```

```
self.line_2.setGeometry(QRect(10, 470, 311, 21))
self.line_2.setFrameShape(QtGui.QFrame.HLine)
self.line_2.setFrameShadow(QtGui.QFrame.Sunken)
self.line_2.setObjectName(_fromUtf8("line_2"))
self.label_6 = QtGui.QLabel(self.centralwidget)
self.label_6.setGeometry(QRect(20, 491, 46, 13))
self.label_6.setObjectName(_fromUtf8("label_6"))
self.label_7 = QtGui.QLabel(self.centralwidget)
self.label_7.setGeometry(QRect(20, 521, 71, 16))
self.label_7.setObjectName(_fromUtf8("label_7"))
self.groupBox = QtGui.QGroupBox(self.centralwidget)
self.groupBox.setGeometry(QRect(320, 290, 451, 411))
self.groupBox.setCheckable(True)
self.groupBox.setChecked(False)
self.groupBox.setObjectName(_fromUtf8("groupBox"))
self.lineNod = QtGui.QLineEdit(self.groupBox)
self.lineNod.setGeometry(QRect(170, 120, 113, 20))
self.lineNod.setObjectName(_fromUtf8("lineNod"))
self.label_8 = QtGui.QLabel(self.groupBox)
self.label_8.setGeometry(QRect(20, 120, 141, 20))
self.label_8.setObjectName(_fromUtf8("label_8"))
self.btnBarrido = QtGui.QPushButton(self.groupBox)
self.btnBarrido.setGeometry(QRect(10, 20, 91, 23))
self.btnBarrido.setObjectName(_fromUtf8("btnBarrido"))
self.textBrowser_2 = QtGui.QTextBrowser(self.groupBox)
self.textBrowser_2.setEnabled(True)
self.textBrowser_2.setGeometry(QRect(120, 20, 321, 41))
self.textBrowser_2.setObjectName(_fromUtf8("textBrowser_2"))
self.label_9 = QtGui.QLabel(self.groupBox)
self.label_9.setGeometry(QRect(20, 150, 141, 16))
```

```
self.label_9.setObjectName(_fromUtf8("label_9"))
self.lineNumero = QtGui.QLineEdit(self.groupBox)
self.lineNumero.setGeometry(QtCore.QRect(170, 150, 113, 20))
self.lineNumero.setObjectName(_fromUtf8("lineNumero"))
self.label_10 = QtGui.QLabel(self.groupBox)
self.label_10.setGeometry(QtCore.QRect(20, 180, 101, 16))
self.label_10.setObjectName(_fromUtf8("label_10"))
self.lineTamano = QtGui.QLineEdit(self.groupBox)
self.lineTamano.setGeometry(QtCore.QRect(170, 180, 113, 20))
self.lineTamano.setObjectName(_fromUtf8("lineTamano"))
self.radioBtnTipo1 = QtGui.QRadioButton(self.groupBox)
self.radioBtnTipo1.setGeometry(QtCore.QRect(30, 230, 91, 17))
self.radioBtnTipo1.setObjectName(_fromUtf8("radioBtnTipo1"))
self.radioBtnTipo2 = QtGui.QRadioButton(self.groupBox)
self.radioBtnTipo2.setGeometry(QtCore.QRect(30, 260, 91, 17))
self.radioBtnTipo2.setObjectName(_fromUtf8("radioBtnTipo2"))
self.radioBtnTipo5 = QtGui.QRadioButton(self.groupBox)
self.radioBtnTipo5.setGeometry(QtCore.QRect(30, 290, 91, 17))
self.radioBtnTipo5.setObjectName(_fromUtf8("radioBtnTipo5"))
self.radioBtnTipo6 = QtGui.QRadioButton(self.groupBox)
self.radioBtnTipo6.setGeometry(QtCore.QRect(30, 320, 101, 17))
self.radioBtnTipo6.setObjectName(_fromUtf8("radioBtnTipo6"))
self.lineTipo = QtGui.QLineEdit(self.groupBox)
self.lineTipo.setGeometry(QtCore.QRect(290, 280, 161, 21))
self.lineTipo.setObjectName(_fromUtf8("lineTipo"))
self.btnTipo = QtGui.QPushButton(self.groupBox)
self.btnTipo.setGeometry(QtCore.QRect(30, 350, 91, 23))
self.btnTipo.setObjectName(_fromUtf8("btnTipo"))
self.label_23 = QtGui.QLabel(self.groupBox)
self.label_23.setGeometry(QtCore.QRect(30, 210, 46, 13))
```

```
self.label_23.setObjectName(_fromUtf8("label_23"))
self.label_25 = QtGui.QLabel(self.groupBox)
self.label_25.setGeometry(QtCore.QRect(330, 250, 81, 20))
self.label_25.setObjectName(_fromUtf8("label_25"))
self.groupBox_2 = QtGui.QGroupBox(self.groupBox)
self.groupBox_2.setGeometry(QtCore.QRect(170, 210, 111, 171))
self.groupBox_2.setCheckable(True)
self.groupBox_2.setChecked(False)
self.groupBox_2.setObjectName(_fromUtf8("groupBox_2"))
self.radioBtnTipo6s = QtGui.QRadioButton(self.groupBox_2)
self.radioBtnTipo6s.setGeometry(QtCore.QRect(10, 110, 101, 17))
self.radioBtnTipo6s.setObjectName(_fromUtf8("radioBtnTipo6s"))
self.radioBtnTipo1s = QtGui.QRadioButton(self.groupBox_2)
self.radioBtnTipo1s.setGeometry(QtCore.QRect(10, 20, 91, 17))
self.radioBtnTipo1s.setObjectName(_fromUtf8("radioBtnTipo1s"))
self.radioBtnTipo5s = QtGui.QRadioButton(self.groupBox_2)
self.radioBtnTipo5s.setGeometry(QtCore.QRect(10, 80, 91, 17))
self.radioBtnTipo5s.setObjectName(_fromUtf8("radioBtnTipo5s"))
self.radioBtnTipo2s = QtGui.QRadioButton(self.groupBox_2)
self.radioBtnTipo2s.setGeometry(QtCore.QRect(10, 50, 91, 17))
self.radioBtnTipo2s.setObjectName(_fromUtf8("radioBtnTipo2s"))
self.btnSecundario = QtGui.QPushButton(self.groupBox_2)
self.btnSecundario.setGeometry(QtCore.QRect(10, 140, 91, 23))
self.btnSecundario.setObjectName(_fromUtf8("btnSecundario"))
self.label_24 = QtGui.QLabel(self.groupBox)
self.label_24.setGeometry(QtCore.QRect(290, 150, 81, 16))
self.label_24.setObjectName(_fromUtf8("label_24"))
self.label_26 = QtGui.QLabel(self.groupBox)
self.label_26.setGeometry(QtCore.QRect(290, 180, 46, 13))
self.label_26.setObjectName(_fromUtf8("label_26"))
```

```

self.label_31 = QtGui.QLabel(self.groupBox)
self.label_31.setGeometry(QtCore.QRect(290, 120, 21, 16))
self.label_31.setObjectName(_fromUtf8("label_31"))
self.groupBox_3 = QtGui.QGroupBox(self.groupBox)
self.groupBox_3.setGeometry(QtCore.QRect(10, 70, 411, 51))
self.groupBox_3.setCheckable(True)
self.groupBox_3.setChecked(False)
self.groupBox_3.setObjectName(_fromUtf8("groupBox_3"))
self.btnCorte = QtGui.QPushButton(self.groupBox_3)
self.btnCorte.setGeometry(QtCore.QRect(11, 21, 75, 23))
self.btnCorte.setObjectName(_fromUtf8("btnCorte"))
self.btnBarrido2 = QtGui.QPushButton(self.groupBox_3)
self.btnBarrido2.setGeometry(QtCore.QRect(231, 21, 81, 23))
self.btnBarrido2.setObjectName(_fromUtf8("btnBarrido2"))
self.lineCorte = QtGui.QLineEdit(self.groupBox_3)
self.lineCorte.setGeometry(QtCore.QRect(92, 22, 133, 23))
self.lineCorte.setObjectName(_fromUtf8("lineCorte"))
self.pushButton = QtGui.QPushButton(self.groupBox)
self.pushButton.setGeometry(QtCore.QRect(20, 43, 75, 20))
self.pushButton.setObjectName(_fromUtf8("pushButton"))
self.pushButton_2 = QtGui.QPushButton(self.groupBox)
self.pushButton_2.setGeometry(QtCore.QRect(320, 320, 75, 23))
self.pushButton_2.setObjectName(_fromUtf8("pushButton_2"))
self.textBrowser_4 = QtGui.QTextBrowser(self.groupBox)
self.textBrowser_4.setEnabled(True)
self.textBrowser_4.setGeometry(QtCore.QRect(290, 210, 151, 41))
self.textBrowser_4.setObjectName(_fromUtf8("textBrowser_4"))
self.btnNuevo = QtGui.QPushButton(self.centralwidget)
self.btnNuevo.setGeometry(QtCore.QRect(680, 10, 81, 41))
self.btnNuevo.setObjectName(_fromUtf8("btnNuevo"))

```

```
self.horizontalSlider = QtGui.QSlider(self.centralwidget)
self.horizontalSlider.setGeometry(QtCore.QRect(130, 150, 391, 20))
self.horizontalSlider.setMaximum(100)
self.horizontalSlider.setProperty("value", 70)
self.horizontalSlider.setOrientation(QtCore.Qt.Horizontal)
self.horizontalSlider.setObjectName(_fromUtf8("horizontalSlider"))
self.btnClearData = QtGui.QPushButton(self.centralwidget)
self.btnClearData.setGeometry(QtCore.QRect(590, 20, 75, 23))
self.btnClearData.setObjectName(_fromUtf8("btnClearData"))
self.btnClear2 = QtGui.QPushButton(self.centralwidget)
self.btnClear2.setGeometry(QtCore.QRect(180, 641, 75, 23))
self.btnClear2.setObjectName(_fromUtf8("btnClear2"))
self.lineReactivo = QtGui.QLineEdit(self.centralwidget)
self.lineReactivo.setGeometry(QtCore.QRect(120, 551, 181, 20))
self.lineReactivo.setObjectName(_fromUtf8("lineReactivo"))
self.label_11 = QtGui.QLabel(self.centralwidget)
self.label_11.setGeometry(QtCore.QRect(20, 551, 61, 16))
self.label_11.setObjectName(_fromUtf8("label_11"))
self.label_12 = QtGui.QLabel(self.centralwidget)
self.label_12.setGeometry(QtCore.QRect(396, 180, 20, 20))
self.label_12.setObjectName(_fromUtf8("label_12"))
self.label_13 = QtGui.QLabel(self.centralwidget)
self.label_13.setGeometry(QtCore.QRect(510, 180, 21, 20))
self.label_13.setObjectName(_fromUtf8("label_13"))
self.label_14 = QtGui.QLabel(self.centralwidget)
self.label_14.setGeometry(QtCore.QRect(120, 180, 20, 20))
self.label_14.setObjectName(_fromUtf8("label_14"))
self.label_15 = QtGui.QLabel(self.centralwidget)
self.label_15.setGeometry(QtCore.QRect(360, 180, 20, 20))
self.label_15.setObjectName(_fromUtf8("label_15"))
```

```
self.line_3 = QtGui.QFrame(self.centralwidget)
self.line_3.setGeometry(QtCore.QRect(120, 160, 16, 20))
self.line_3.setMaximumSize(QtCore.QSize(1400000, 16777215))
self.line_3 setFrameShape(QtGui.QFrame.VLine)
self.line_3 setFrameShadow(QtGui.QFrame.Sunken)
self.line_3.setObjectName(_fromUtf8("line_3"))
self.line_4 = QtGui.QFrame(self.centralwidget)
self.line_4.setGeometry(QtCore.QRect(510, 160, 16, 20))
self.line_4 setFrameShape(QtGui.QFrame.VLine)
self.line_4 setFrameShadow(QtGui.QFrame.Sunken)
self.line_4.setObjectName(_fromUtf8("line_4"))
self.label_16 = QtGui.QLabel(self.centralwidget)
self.label_16.setGeometry(QtCore.QRect(320, 180, 20, 20))
self.label_16.setObjectName(_fromUtf8("label_16"))
self.label_17 = QtGui.QLabel(self.centralwidget)
self.label_17.setGeometry(QtCore.QRect(436, 180, 20, 20))
self.label_17.setObjectName(_fromUtf8("label_17"))
self.label_18 = QtGui.QLabel(self.centralwidget)
self.label_18.setGeometry(QtCore.QRect(280, 180, 20, 20))
self.label_18.setObjectName(_fromUtf8("label_18"))
self.label_19 = QtGui.QLabel(self.centralwidget)
self.label_19.setGeometry(QtCore.QRect(470, 180, 20, 20))
self.label_19.setObjectName(_fromUtf8("label_19"))
self.label_20 = QtGui.QLabel(self.centralwidget)
self.label_20.setGeometry(QtCore.QRect(240, 180, 20, 20))
self.label_20.setObjectName(_fromUtf8("label_20"))
self.label_21 = QtGui.QLabel(self.centralwidget)
self.label_21.setGeometry(QtCore.QRect(206, 180, 20, 20))
self.label_21.setObjectName(_fromUtf8("label_21"))
self.label_22 = QtGui.QLabel(self.centralwidget)
```

```
self.label_22.setGeometry(QCore.QRect(160, 180, 20, 20))
self.label_22.setObjectName(_fromUtf8("label_22"))
self.line_7 = QtGui.QFrame(self.centralwidget)
self.line_7.setGeometry(QCore.QRect(310, 480, 20, 161))
self.line_7 setFrameShape(QtGui.QFrame.VLine)
self.line_7 setFrameShadow(QtGui.QFrame.Sunken)
self.line_7.setObjectName(_fromUtf8("line_7"))
self.label_27 = QtGui.QLabel(self.centralwidget)
self.label_27.setGeometry(QCore.QRect(20, 591, 81, 16))
self.label_27.setObjectName(_fromUtf8("label_27"))
self.lineObservaciones = QtGui.QLineEdit(self.centralwidget)
self.lineObservaciones.setGeometry(QCore.QRect(120, 580, 181, 51))
self.lineObservaciones.setObjectName(_fromUtf8("lineObservaciones"))
self.label_28 = QtGui.QLabel(self.centralwidget)
self.label_28.setGeometry(QCore.QRect(290, 330, 21, 16))
self.label_28.setObjectName(_fromUtf8("label_28"))
self.label_29 = QtGui.QLabel(self.centralwidget)
self.label_29.setGeometry(QCore.QRect(290, 370, 21, 16))
self.label_29.setObjectName(_fromUtf8("label_29"))
self.label_30 = QtGui.QLabel(self.centralwidget)
self.label_30.setGeometry(QCore.QRect(290, 410, 16, 16))
self.label_30.setObjectName(_fromUtf8("label_30"))
self.label_32 = QtGui.QLabel(self.centralwidget)
self.label_32.setGeometry(QCore.QRect(240, 250, 20, 20))
self.label_32.setObjectName(_fromUtf8("label_32"))
self.line_5 = QtGui.QFrame(self.centralwidget)
self.line_5.setGeometry(QCore.QRect(510, 230, 16, 20))
self.line_5 setFrameShape(QtGui.QFrame.VLine)
self.line_5 setFrameShadow(QtGui.QFrame.Sunken)
self.line_5.setObjectName(_fromUtf8("line_5"))
```



```
self.label_33 = QtGui.QLabel(self.centralwidget)
self.label_33.setGeometry(QtCore.QRect(320, 250, 41, 20))
self.label_33.setObjectName(_fromUtf8("label_33"))
self.label_34 = QtGui.QLabel(self.centralwidget)
self.label_34.setGeometry(QtCore.QRect(430, 250, 20, 20))
self.label_34.setObjectName(_fromUtf8("label_34"))
self.label_35 = QtGui.QLabel(self.centralwidget)
self.label_35.setGeometry(QtCore.QRect(120, 250, 20, 20))
self.label_35.setObjectName(_fromUtf8("label_35"))
self.label_36 = QtGui.QLabel(self.centralwidget)
self.label_36.setGeometry(QtCore.QRect(280, 250, 20, 20))
self.label_36.setObjectName(_fromUtf8("label_36"))
self.label_37 = QtGui.QLabel(self.centralwidget)
self.label_37.setGeometry(QtCore.QRect(200, 250, 20, 20))
self.label_37.setObjectName(_fromUtf8("label_37"))
self.horizontalSlider_2 = QtGui.QSlider(self.centralwidget)
self.horizontalSlider_2.setGeometry(QtCore.QRect(130, 220, 391, 20))
self.horizontalSlider_2.setMinimum(100)
self.horizontalSlider_2.setMaximum(200)
self.horizontalSlider_2.setProperty("value", 155)
self.horizontalSlider_2.setSliderPosition(155)
self.horizontalSlider_2.setOrientation(QtCore.Qt.Horizontal)
self.horizontalSlider_2.setObjectName(_fromUtf8("horizontalSlider_2"))
self.label_38 = QtGui.QLabel(self.centralwidget)
self.label_38.setGeometry(QtCore.QRect(390, 250, 20, 20))
self.label_38.setObjectName(_fromUtf8("label_38"))
self.label_39 = QtGui.QLabel(self.centralwidget)
self.label_39.setGeometry(QtCore.QRect(360, 250, 51, 20))
self.label_39.setObjectName(_fromUtf8("label_39"))
self.label_40 = QtGui.QLabel(self.centralwidget)
```

```
self.label_40.setGeometry(QRect(470, 250, 20, 20))
self.label_40.setObjectName(_fromUtf8("label_40"))
self.label_41 = QtGui.QLabel(self.centralwidget)
self.label_41.setGeometry(QRect(160, 250, 31, 20))
self.label_41.setObjectName(_fromUtf8("label_41"))
self.label_42 = QtGui.QLabel(self.centralwidget)
self.label_42.setGeometry(QRect(510, 250, 21, 20))
self.label_42.setObjectName(_fromUtf8("label_42"))
self.line_8 = QtGui.QFrame(self.centralwidget)
self.line_8.setGeometry(QRect(120, 230, 16, 20))
self.line_8.setMaximumSize(QSize(1400000, 16777215))
self.line_8 setFrameShape(QtGui.QFrame.VLine)
self.line_8 setFrameShadow(QtGui.QFrame.Sunken)
self.line_8.setObjectName(_fromUtf8("line_8"))
self.label_43 = QtGui.QLabel(self.centralwidget)
self.label_43.setGeometry(QRect(70, 110, 161, 16))
self.label_43.setObjectName(_fromUtf8("label_43"))
self.lineFe = QtGui.QLineEdit(self.centralwidget)
self.lineFe.setGeometry(QRect(240, 110, 131, 20))
self.lineFe.setObjectName(_fromUtf8("lineFe"))
self.textBrowser_3 = QtGui.QTextBrowser(self.centralwidget)
self.textBrowser_3.setEnabled(True)
self.textBrowser_3.setGeometry(QRect(390, 110, 281, 31))
self.textBrowser_3.setObjectName(_fromUtf8("textBrowser_3"))
self.btnFerr = QtGui.QPushButton(self.centralwidget)
self.btnFerr.setGeometry(QRect(490, 20, 91, 23))
self.btnFerr.setObjectName(_fromUtf8("btnFerr"))
self.btnClearResult = QtGui.QPushButton(self.centralwidget)
self.btnClearResult.setGeometry(QRect(200, 450, 75, 23))
self.btnClearResult.setObjectName(_fromUtf8("btnClearResult"))
```

```

MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QtGui.QMenuBar(MainWindow)
self.menubar.setGeometry(QtCore.QRect(0, 0, 778, 21))
self.menubar.setObjectName(_fromUtf8("menubar"))
MainWindow.setMenuBar(self.menubar)
self.statusbar = QtGui.QStatusBar(MainWindow)
self.statusbar.setObjectName(_fromUtf8("statusbar"))
MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)

QtCore.QObject.connect(self.btnClearData, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineInserte.clear)

QtCore.QObject.connect(self.btnNuevo, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.linePer.clear)

QtCore.QObject.connect(self.btnNuevo, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineFer.clear)

QtCore.QObject.connect(self.btnNuevo, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineGraff.clear)

QtCore.QObject.connect(self.btnNuevo, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineTamano.clear)

QtCore.QObject.connect(self.btnNuevo, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineNumero.clear)

QtCore.QObject.connect(self.btnNuevo, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineNod.clear)

QtCore.QObject.connect(self.btnNuevo, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineInserte.clear)

QtCore.QObject.connect(self.btnClear2, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineEncargado.clear)

QtCore.QObject.connect(self.btnClear2, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineFecha.clear)

QtCore.QObject.connect(self.btnClearResult, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.linePer.clear)

QtCore.QObject.connect(self.btnClearResult, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineFer.clear)

```

```
QtCore.QObject.connect(self.btnClearResult, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineGraff.clear)
```

```
QtCore.QObject.connect(self.btnNuevo, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineTipo.clear)
```

```
QtCore.QObject.connect(self.btnNuevo, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineReactivo.clear)
```

```
QtCore.QObject.connect(self.btnNuevo, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineObservaciones.clear)
```

```
QtCore.QObject.connect(self.btnClear2, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineReactivo.clear)
```

```
QtCore.QObject.connect(self.btnClear2, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineObservaciones.clear)
```

```
QtCore.QObject.connect(self.pushButton_2, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineTipo.clear)
```

```
QtCore.QObject.connect(self.pushButton, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineCorte.clear)
```

```
QtCore.QObject.connect(self.pushButton, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineNod.clear)
```

```
QtCore.QObject.connect(self.pushButton, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineNumero.clear)
```

```
QtCore.QObject.connect(self.pushButton, QtCore.SIGNAL(_fromUtf8("clicked()")),
self.lineTamano.clear)
```

```
QtCore.QMetaObject.connectSlotsByName(MainWindow)
```

```
MainWindow.setTabOrder(self.lineInserte, self.lineGrises)
```

```
MainWindow.setTabOrder(self.lineGrises, self.btnCargarImg)
```

```
MainWindow.setTabOrder(self.btnCargarImg, self.btnEjecutar)
```

```
MainWindow.setTabOrder(self.btnEjecutar, self.btnBarrido)
```

```
MainWindow.setTabOrder(self.btnBarrido, self.btnTipo)
```

```
MainWindow.setTabOrder(self.btnTipo, self.btnSecundario)
```

```
MainWindow.setTabOrder(self.btnSecundario, self.lineFecha)
```

```
MainWindow.setTabOrder(self.lineFecha, self.lineEncargado)
```

```
MainWindow.setTabOrder(self.lineEncargado, self.lineReactivo)
```

```
MainWindow.setTabOrder(self.lineReactivo, self.lineObservaciones)
```

```
MainWindow.setTabOrder(self.lineObservaciones, self.btnAnadirTabla)
```

```
MainWindow.setTabOrder(self.btnAnadirTabla, self.btnNuevo)
MainWindow.setTabOrder(self.btnNuevo, self.lineGraff)
MainWindow.setTabOrder(self.lineGraff, self.lineFer)
MainWindow.setTabOrder(self.lineFer, self.linePer)
MainWindow.setTabOrder(self.linePer, self.lineNod)
MainWindow.setTabOrder(self.lineNod, self.textBrowser)
MainWindow.setTabOrder(self.textBrowser, self.groupBox)
MainWindow.setTabOrder(self.groupBox, self.textBrowser_2)
MainWindow.setTabOrder(self.textBrowser_2, self.lineNumero)
MainWindow.setTabOrder(self.lineNumero, self.lineTamano)
MainWindow.setTabOrder(self.lineTamano, self.radioBtnTipo1)
MainWindow.setTabOrder(self.radioBtnTipo1, self.radioBtnTipo2)
MainWindow.setTabOrder(self.radioBtnTipo2, self.radioBtnTipo5)
MainWindow.setTabOrder(self.radioBtnTipo5, self.radioBtnTipo6)
MainWindow.setTabOrder(self.radioBtnTipo6, self.lineTipo)
MainWindow.setTabOrder(self.lineTipo, self.horizontalSlider)
MainWindow.setTabOrder(self.horizontalSlider, self.btnClearData)
MainWindow.setTabOrder(self.btnClearData, self.btnClear2)
MainWindow.setTabOrder(self.btnClear2, self.radioBtnTipo6s)
MainWindow.setTabOrder(self.radioBtnTipo6s, self.radioBtnTipo1s)
MainWindow.setTabOrder(self.radioBtnTipo1s, self.radioBtnTipo5s)
MainWindow.setTabOrder(self.radioBtnTipo5s, self.radioBtnTipo2s)
```

```
def retranslateUi(self, MainWindow):
```

```
    MainWindow.setWindowTitle(_translate("MainWindow", "Procesamiento de Imagenes",
None))
    self.label.setText(_translate("MainWindow", "Inserte el Nombre de la Imagen ", None))
    self.label_2.setText(_translate("MainWindow", "Valor de Escala de Grises (Grafito)", None))
    self.label_3.setText(_translate("MainWindow", "Porcentaje Grafito :", None))
    self.label_4.setText(_translate("MainWindow", "Porcentaje de Ferrita :", None))
```

```

        self.textBrowser.setHtml(_translate("MainWindow", "<!DOCTYPE HTML PUBLIC \"/>
//W3C//DTD HTML 4.0//EN\" \"http://www.w3.org/TR/REC-html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:'MS Shell Dlg 2'; font-size:8.25pt; font-weight:400;
font-style:normal;\">\n"
"<p align=\"justify\" style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\"><span style=\" font-size:8pt;\">Valor de escala
de grises seleccionado revelar grafito</span></p></body></html>", None))

        self.lineGrises.setText(_translate("MainWindow", "70", None))

        self.btnEjecutar.setText(_translate("MainWindow", "Ejecutar", None))

        self.label_5.setText(_translate("MainWindow", "Porcentaje de Perlita :", None))

        self.btnCargarImg.setText(_translate("MainWindow", "Escala Grafito", None))

        self.btnAddirTabla.setText(_translate("MainWindow", "Añadir a Tabla", None))

        self.label_6.setText(_translate("MainWindow", "Fecha :", None))

        self.label_7.setText(_translate("MainWindow", "Encargado :", None))

        self.groupBox.setTitle(_translate("MainWindow", "Análisis de Nodularidad", None))

        self.label_8.setText(_translate("MainWindow", "Porcentaje de Nodularidad :", None))

        self.btnBarrido.setText(_translate("MainWindow", "Barrido Imagen", None))

        self.textBrowser_2.setHtml(_translate("MainWindow", "<!DOCTYPE HTML PUBLIC \"/>
//W3C//DTD HTML 4.0//EN\" \"http://www.w3.org/TR/REC-html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:'MS Shell Dlg 2'; font-size:8.25pt; font-weight:400;
font-style:normal;\">\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-
indent:0; text-indent:0px;\"><span style=\" font-size:8pt;\">Proceda a seleccionar los nódulos,
dando click en extremos opuestos del mismo. Al terminar presione la tecla &quot;s&quot;
</span></p></body></html>", None))

        self.label_9.setText(_translate("MainWindow", "Numeros de Nodulos :", None))

        self.label_10.setText(_translate("MainWindow", "Tamaño Promedio : ", None))

        self.radioBtnTipo1.setText(_translate("MainWindow", "Grafito Tipo I", None))

        self.radioBtnTipo2.setText(_translate("MainWindow", "Grafito Tipo II", None))

        self.radioBtnTipo5.setText(_translate("MainWindow", "Grafito Tipo V", None))

```

```

self.radioBtnTipo6.setText(_translate("MainWindow", "Grafito Tipo VI", None))
self.btnTipo.setText(_translate("MainWindow", "Tipo de Nodulo", None))
self.label_23.setText(_translate("MainWindow", "Principal", None))
self.label_25.setText(_translate("MainWindow", "Tipo de Grafito", None))
self.groupBox_2.setTitle(_translate("MainWindow", "Secundario", None))
self.radioBtnTipo6s.setText(_translate("MainWindow", "Grafito Tipo VI", None))
self.radioBtnTipo1s.setText(_translate("MainWindow", "Grafito Tipo I", None))
self.radioBtnTipo5s.setText(_translate("MainWindow", "Grafito Tipo V", None))
self.radioBtnTipo2s.setText(_translate("MainWindow", "Grafito Tipo II", None))
self.btnSecundario.setText(_translate("MainWindow", "Tipo Secundario", None))
self.label_24.setText(_translate("MainWindow", "Nódulos/mm^2", None))
self.label_26.setText(_translate("MainWindow", "mm", None))
self.label_31.setText(_translate("MainWindow", "%", None))
self.groupBox_3.setTitle(_translate("MainWindow", "Nodulos pequeños", None))
self.btnCorte.setText(_translate("MainWindow", "Corte", None))
self.btnBarrido2.setText(_translate("MainWindow", "Barrido Imagen", None))
self.pushButton.setText(_translate("MainWindow", "Clear Barrido", None))
self.pushButton_2.setText(_translate("MainWindow", "Clear Tipo", None))

self.textBrowser_4.setHtml(_translate("MainWindow", "<!DOCTYPE HTML PUBLIC \
//W3C//DTD HTML 4.0//EN\ "http://www.w3.org/TR/REC-html40/strict.dtd">\n"
"<html><head><meta name=\\"qrichtext\" content=\\"1\" /><style type=\\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\\" font-family:\'MS Shell Dlg 2\'; font-size:8.25pt; font-weight:400;
font-style:normal;\">\n"
"<p style=\\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-
indent:0; text-indent:0px;\"><span style=\\" font-size:8pt;\">Al finalizar presionar la tecla
"n"</span></p></body></html>", None))

self.btnNuevo.setText(_translate("MainWindow", "NUEVO", None))
self.btnClearData.setText(_translate("MainWindow", "Clear", None))
self.btnClear2.setText(_translate("MainWindow", "Clear", None))
self.label_11.setText(_translate("MainWindow", "Reactivo :", None))
self.label_12.setText(_translate("MainWindow", "70", None))

```

```

self.label_13.setText(_translate("MainWindow", "100", None))
self.label_14.setText(_translate("MainWindow", "0", None))
self.label_15.setText(_translate("MainWindow", "60", None))
self.label_16.setText(_translate("MainWindow", "50", None))
self.label_17.setText(_translate("MainWindow", "80", None))
self.label_18.setText(_translate("MainWindow", "40", None))
self.label_19.setText(_translate("MainWindow", "90", None))
self.label_20.setText(_translate("MainWindow", "30", None))
self.label_21.setText(_translate("MainWindow", "20", None))
self.label_22.setText(_translate("MainWindow", "10", None))
self.label_27.setText(_translate("MainWindow", "Observaciones :", None))
self.label_28.setText(_translate("MainWindow", "%", None))
self.label_29.setText(_translate("MainWindow", "%", None))
self.label_30.setText(_translate("MainWindow", "%", None))
self.label_32.setText(_translate("MainWindow", "130", None))
self.label_33.setText(_translate("MainWindow", "150", None))
self.label_34.setText(_translate("MainWindow", "180", None))
self.label_35.setText(_translate("MainWindow", "100", None))
self.label_36.setText(_translate("MainWindow", "140", None))
self.label_37.setText(_translate("MainWindow", "120", None))
self.label_38.setText(_translate("MainWindow", "170", None))
self.label_39.setText(_translate("MainWindow", "160", None))
self.label_40.setText(_translate("MainWindow", "190", None))
self.label_41.setText(_translate("MainWindow", "110", None))
self.label_42.setText(_translate("MainWindow", "200", None))
self.label_43.setText(_translate("MainWindow", "Valor de Escala de Grises (Ferrita)", None))
self.lineFe.setText(_translate("MainWindow", "155", None))

self.textBrowser_3.setHtml(_translate("MainWindow", "<!DOCTYPE HTML PUBLIC \"/>
//W3C//DTD HTML 4.0//EN\" \"http://www.w3.org/TR/REC-html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }")

```



```
"</style></head><body style=\" font-family:\'MS Shell Dlg 2\'; font-size:8.25pt; font-weight:400; font-style:normal;\">>\n"
```

```
"<p align=\"justify\" style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-indent:0; text-indent:0px;\"><span style=\" font-size:8pt;\">>Valor de escala de grises seleccionado revelar ferrita</span></p></body></html>", None))
```

```
self.btnFerr.setText(_translate("MainWindow", "Escala Ferrita", None))
```

```
self.btnClearResult.setText(_translate("MainWindow", "Clear", None))
```