



UNIVERSIDAD JOSÉ ANTONIO PÁEZ

**COMPARACIÓN DE RESULTADOS DE
SIMULACIONES DE TRATAMIENTO
TÉRMICO MEDIANTE TEMPLE EN AGUA
EN ACEROS DE BAJA ALEACIÓN AISI
4XXX Y 5XXX ENTRE EL MÉTODO DE
ELEMENTOS FINITOS Y PYTHON**

Autores:

Campillo L. Cristina

C.i.: 20.891.359

Sequera O. Jesus A.

C.i.: 23.649.768

Tutor:

Ing. Romanello Donato.

C.i.: 4.131.877

Urb. Yuma ii, calle n° 3, municipio San Diego

Teléfono: (0241) 8714240 (master) - fax: (0241) 8712394



**REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA MECÁNICA**

**COMPARACIÓN DE RESULTADOS DE SIMULACIONES DE
TRATAMIENTO TÉRMICO MEDIANTE TEMPLE EN AGUA EN ACEROS
DE BAJA ALEACIÓN AISI 4XXX Y 5XXX ENTRE EL MÉTODO DE
ELEMENTOS FINITOS Y PYTHON**

**Trabajo especial de grado presentado como requisito para optar
al título de**

INGENIERO MECÁNICO

Autores:

Campillo L. Cristina

C.I.: 20.891.359

Sequera O. Jesus A.

C.I.: 23.649.768

Tutor:

Ing. Romanello Donato.

C.I.: 4.131.877

San Diego, Marzo de 2018



REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA MECÁNICA

ACEPTACIÓN DEL TUTOR

Quien suscribe, Ingeniero Donato Romanello, portador de la cédula de identidad N° 4.131.877, en mi carácter de tutor del trabajo de grado presentado por los ciudadanos Cristina Campillo, portadora de la cedula de identidad N° 20.891.359 y Jesus Sequera, portador de la cedula de identidad N° 23.649.768 titulado **COMPARACIÓN DE RESULTADOS DE SIMULACIONES DE TRATAMIENTO TÉRMICO MEDIANTE TEMPLE EN AGUA EN ACEROS DE BAJA ALEACIÓN AISI 4XXX Y 5XXX ENTRE EL MÉTODO DE ELEMENTOS FINITOS Y PYTHON**, presentado como requisito parcial para optar al título de Ingeniero Mecánico, considero que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del jurado examinador que se designe.

En San Diego, a los 15 Marzo del año dos mil dieciocho.

Ing. Donato Romanello
C.I.: 4.131.877

ÍNDICE

ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS	x
RESUMEN.....	xi
INTRODUCCIÓN	1

CAPÍTULO

I EL PROBLEMA.....	3
1.1 Planteamiento del problema	3
1.2 Formulación del problema.....	5
1.3 Objetivos de la investigación	5
1.3.1 Objetivo general	5
1.3.2 Objetivos específicos	5
1.4 Justificación del problema.....	5
1.5 Alcance.....	6
II MARCO TEÓRICO	7
2.1 Antecedentes	7
2.2 Bases teóricas	8
2.2.1 Generalidades de las aleaciones ferrosas.....	8
2.2.2 Generalidades de los aceros de baja aleación.....	9
2.2.3 Genominaciones de los aceros.....	10

2.2.4 Efecto de diversos elementos en los aceros	13
2.2.5 Diagrama hierro-carbono (fe-c).....	15
2.2.6 Tratamientos térmicos	19
2.2.7 Etapas del tratamiento térmico	20
2.2.8 Tipos de tratamientos térmicos.....	20
2.2.8.1 Recocido	20
2.2.8.2 Temple	21
2.2.8.3 Normalizado	22
2.2.9 Diagrama TTT y CCT	23
2.2.10 Influencia de los elementos de aleación sobre los diagramas TTT y CCT	25
2.2.11 Generalidades de la simulación	26
2.2.11.1 Definición de simulación	28
2.2.11.2 Etapas para realizar un estudio de simulación	29
2.2.11.3 Software a evaluar para la simulación	30
2.2.12 Definición de modelo matemático.....	32
2.2.12.1 Modelos matemáticos a evaluar.....	32
III MARCO METODOLÓGICO	39
3.1 Tipo de investigación	39
3.2 Diseño de la investigación	39
3.3 Nivel de la investigación	40
3.4 Población y muestra	40

3.5 Técnicas e instrumentos de recolección de datos	41
3.6 Fases de la investigación	41
IV RESULTADOS	43
4.1 Seleccionar programas libres capaces de realizar simulaciones de tratamientos térmicos en los aceros en estudio.	43
4.2 Aplicación del modelo matemático, obtención de los diagramas TTT de los aceros seleccionados mediante el uso del software Python y comparacion con la USS	43
4.3 Aplicación del modelo matemático y obtener los diagramas CCT de los aceros seleccionados con el software Python.....	50
4.4 Comparar los resultados obtenidos por Python con Matlab.....	53
4.4.1Requerimiento Computacional	54
4.4.2Comparación entre el Diagrama TTT generado por Matlab y Python	55
4.4.3Comparación entre el Diagrama CCT generado por Matlab y Python.....	57
CONCLUSIONES	63
RECOMENDACIONES.....	65
BIBLIOGRAFÍA.....	66
ANEXOS	67

ÍNDICE DE FIGURAS

CONTENIDO	PÁG.
FIGURA	
1 Clasificación de las aleaciones ferrosas.....	9
2 Designación según normas AISI-SAE.....	10
3 Designaciones de los aceros según AISI-SAE.....	11
4 Convención del primer dígito según normas AISI-SAE.....	11
5 Designaciones y composiciones de cinco aceros bajos encarbono y de tres aceros de baja aleación y alta resistencia.....	13
6 Diagrama de fases hierro-carbono.....	18
7 Influencia de los elementos aleantes en la temperatura eutectoide A_1 Y el contenido de carbono del punto eutectoide....	18
8 Método de obtención de los diferentes tipos de recocido.....	21
9 Esquema de los tratamientos térmicos más empleados y del desplazamiento de las zonas de transformación.	23
10 Diagrama TTT para un acero.....	24
11 Diagrama TTT con la influencia de elementos aleantes.....	24
12 Representación de la ecuación JMAK ($V \{t\}$) en función del tiempo para dos (2) temperaturas diferentes (T_1 y T_2) y los tiempos de incubación para las dos (2) temperaturas (t_1 y t_2)....	35
13 Curva TTT de un Acero. Se indican los tiempos t_i y t_f a una cierta temperatura T_n	35
14 Aplicación de la Regla de Adición. Izquierda: tiempo de incubación de la transformación no isotérmica (

21	Comparación curva de transformación isotérmica del acero AISI 5140 de las USS con el software Python.....	49
22	Composición de los aceros estudiados.....	50
23	Declaración de las variables en el programa para el acero AISI 4027.....	51
24	Formulación de las gráficas en el programa para el acero AISI 4027.....	52
25	Diagrama CCT del acero AISI 4027 según el modelo matemático.....	52
26	Diagrama CCT del acero AISI 4340 según el modelo matemático.....	53
27	Diagrama CCT del acero AISI 5140 según el modelo matemático.....	53
28	Comparación curva de transformación isotérmica del acero AISI 4027 de las del software Matlab entre el software Python..	55
29	Comparación curva de transformación isotérmica del acero AISI 4340 de las del software Matlab entre el software Python..	56
30	Comparación curva de transformación isotérmica del acero AISI 5140 de las del software Matlab entre el software Python..	57
31	Comparación diagrama CCT del acero AISI 4340 entre las de Matlab y la de Python.....	58
32	Comparación diagrama CCT del acero AISI 4027 entre las de Matlab y la de Python.....	59
33	Comparación diagrama CCT del acero AISI 5140 entre las de Matlab y la de Python.....	60

ÍNDICE DE TABLA

CONTENIDO	PÁG
TABLA	
1 Requerimientos Computacionales de cada programa.....	44
2 Propiedades de los aceros en estudio para los diagrama TTT.	50
3 Propiedades de los aceros en estudio para los diagrama CCT.....	54
4 Ventajas y desventajas de utilizar Python como Matlab.	61

REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA MECÁNICA

**COMPARACIÓN DE RESULTADOS DE SIMULACIONES DE
TRATAMIENTO TÉRMICO MEDIANTE TEMPLE EN AGUA EN ACEROS
DE BAJA ALEACIÓN AISI 4XXX Y 5XXX ENTRE EL MÉTODO DE
ELEMENTOS FINITOS Y PYTHON**

Autores:Campillo Cristina.

Sequera Jesus.

Tutor: Romanello Donato.

Fecha:Marzo 2018

RESUMEN

Debido al extenso y arduo trabajo que representa realizar las curvas Temperatura-Tiempo-Transformación(TTT) y Transformación de enfriamiento continuo (CCT) de forma experimental en aceros, la ciencia e ingeniería se ha apoyado en el avance de las herramientas computacionales para simular ensayos mediante modelos matemáticos. Para ello se realizó una investigación comparativa, en dicha investigación se escogió como software Python donde se escribieron códigos que simulan las curvas TTT y CCT basados en los modelos matemáticos de Kirkaldy y JMAK. Seguidamente se compararon con los resultados de las curvas de la “Corporación de Acero de los Estados Unidos” (USS) realizadas con métodos experimentales y con otra herramienta computacional denominada Matlab, de dichas comparaciones se obtuvo variaciones en: tiempos de las transformaciones, temperaturas, amplitudes de las zonas microconstituyentes, entre otro. Se establecieron las similitudes, ventajas y desventajas de realizar los diagramas entre Matlab y Python.

Descriptores: TTT, CCT, herramienta computacional, simulación, comparación, modelos matemáticos, USS, Matlab, Python.

INTRODUCCIÓN

Este estudio centró su atención en el ámbito de nuevas tecnologías gratuitas, más específicamente en el apartado de software y programación en ingeniería, donde se exploró en un programa libre, sin mucha presencia de antecedentes o estudios realizados para el uso; en el cual, se aprovechó dicho software. Los resultados obtenidos se compararon con otro software ya consolidado y con una historia marcada en la ingeniería como simuladores, donde se buscó obtener de la comparación, cual método de predicción es más eficiente y eficaz.

Estas simulaciones fueron comprendidas en el ámbito de la ciencia de los materiales como lo son los tratamientos térmicos, más específicamente el temple en agua; que no ha dejado de ser importante para la industria actual. Minimizar costos es de las metas más importantes en las industrias, este trabajo de investigación y el código que se escribió en el software gratuito, podrán ser utilizados por ingenieros o estudiantes de ingeniería con conocimientos básico de física térmica, ciencia de los materiales, programación y otras desviaciones. Esta investigación estuvo constituida por cuatro capítulos, la cual se estructuró de la siguiente manera:

El Capítulo I, que comprende toda la problemática del tema; conocimientos sobre los tratamientos térmicos, su aporte hasta la actualidad y perspectiva futurista; a nivel industrial en la sustitución de otros software debido a su simple uso y su estado de libre distribución. Una vez que se identificó el problema, se procedió a la formulación del problema, para establecer el objetivo general y sus objetivos específicos, las cuales fueron el sustento de esta investigación. Posteriormente se justificó el problema donde se reflejó la importancia de esta investigación sujeta a los posibles resultados a obtenerse. Del mismo modo, el alcance que tuvo.

El capítulo II, comprende los aspectos del marco teórico, como son antecedentes, referidos a trabajos realizados a nivel nacional e internacional. Así como también bases teóricas donde se sustentara la científicidad documental del

trabajo, el cual contempla generalidades de los tratamientos térmicos, programas, simulaciones y características de las aleaciones de los aceros.

El capítulo III, viene referido al marco metodológico el cual explica los procedimientos propios del estudio, como concepto de investigación, tipo de investigación, diseño, población y muestra. Así como las fases y procedimientos técnicos e instrumental del estudio.

En el capítulo IV, se aborda brevemente el cierre de cada uno de los procesos llevados a cabo y se presentanal lector los resultados obtenidos de cada una de las fases planteadas en el capítulo III para el cumplimiento de los objetivos del trabajo de grado, y finalizando con las pertinentes conclusiones y recomendaciones del proceso investigativo. Finalmente, cierra la estructura el listado de las bibliografías consultadas y los anexos que se manejaron en la investigación.

CAPÍTULO I

EL PROBLEMA

1.1 Planteamiento del problema

La ingeniería desde sus inicios combina conocimientos de las ciencias naturales, la matemática y la experiencia, para desarrollar nuevas formas de utilización de materiales y recursos en beneficio del hombre.

El ingeniero a lo largo de la historia ha buscado como utilizar los materiales de su entorno a su favor, es allí donde en su intensa búsqueda para facilitar las condiciones de vida encuentra, en el acero un material indispensable en la ingeniería moderna por todas sus aplicaciones posibles. La necesidad del acero en el progreso de la civilización humana ha sido tal que, en construcciones, maquinarias, dispositivos y otros, tienden a tener un gran porcentaje de utilización en su estructura debido a la versatilidad del material, su flexibilidad y la conveniente relación entre resistencia y peso manteniendo un costo adecuado.

En el instante en que se empieza a realizar un proyecto de diseño es necesario conocer las propiedades internas de los materiales para poder predecir su comportamiento en el ámbito laboral cuando estos se encuentren sometidos a esfuerzos y fatiga, para ello anteriormente se utilizaban datos estadísticos tabulados en donde los ingenieros se guiaban por simples promedios, dichos promedios no están lo más acercados a la realidad debido a que muy pocos comportamientos en la naturaleza son lineales, estas formas de querer predecir el comportamiento de un material puede ser confiable hasta cierto límite, por lo cual los métodos modernos son preferidos en las industrias.

En las industrias es necesario alta precisión a la hora de crear o reemplazar piezas metálicas en la construcción, procesos, entre otros, después de haberle realizado a estas piezas procesos como recubrimientos o tratamientos térmicos, que comúnmente son las fases finales en los procesos de manufactura de las piezas

mecánicas para modificar su estructura interna y por ende sus propiedades mecánicas. Los ensayos son procesos utilizados en la industria para observar la estructura interna del material y con esos resultados predecir cómo se comportara el material en funcionamiento, dichos ensayos componen un gasto de tiempo y dinero a la empresa sin tener mucha referencia cuales resultados obtendrán, abriendo la posibilidad de que se consigan piezas no aptas para la ejecución de un trabajo.

En los tratamientos térmicos podemos conseguir diversos tipos, uno de ellos es el temple en agua cuyo objetivo es conseguir la estructura martensítica, la cual posee mayor dureza y resistencia mecánica. Para obtener dicha estructura intervienen diversos factores como lo son la templabilidad del acero, la cantidad de tiempo en la que es sometido el temple, la temperatura en que se realiza el temple y el tamaño de la pieza templada. El temple es un proceso delicado, cualquier variable que no esté entre los rangos aceptables modificará de forma negativa su microestructura produciendo que la pieza quede inutilizable para el trabajo que se le quiere asignar.

Se tiene que considerar que actualmente la variedad de los aceros es muy alta y cada proceso de temple es diferente para cada tipo de acero, es por ello que los resultados de un temple beneficioso para un acero no son con exactitud beneficioso para otro. Por ende se utiliza el diagrama temperatura, tiempo y transformación (TTT) que resume las posibles transformaciones de la austenita para cada acero, imprescindibles tanto para el diseño de tratamientos térmicos como para la interpretación de las microestructuras resultantes después de los mismos.

El avance de la tecnología ha permitido que los ensayos de tratamientos térmicos se puedan ejecutar desde un computador, de esta forma reducir pérdidas de material, dinero y tiempo. Sin embargo los simuladores convencionales que utilizan el método de elementos finitos a pesar de no representar tanta pérdida no están absueltos de estas, es por ello que la evolución del software libre ha acaparado la atención en las grandes industrias.

Las empresas invierten dinero en los simuladores convencionales que se basan en el método de elementos finitos, originado por la antigua inexistencia de programas

computacionales económicos y precisos. Si esto sigue de esa forma seguirán aumentando pérdidas que anualmente se van incrementado y que son evitables. Por ello en este trabajo de grado se planteó la utilización de un software libre denominado Python que cuenta con miles de librerías en línea sin gastar un dólar además ha sido modificado para poder funcionar en diferentes plataformas como Linux, Windows, Macintosh, Solaris y muchas más.

Los resultados que se obtuvieron utilizando Python se compararon con resultados ya previamente obtenidos con el software Matlab en aceros de baja aleación 4xxx y 5xxx.

1.2 Formulación del problema

¿De qué manera se puede pronosticar cuál de las curvas TTT de los aceros de baja aleación AISI 4xxx y 5xxx al aplicarles un temple en agua, se asemeja más a la realidad?

1.3 Objetivos de la investigación

1.3.1 Objetivo general

Comparación de resultados de simulaciones de tratamiento térmico mediante temple en agua en aceros de baja aleación AISI 4xxx y 5xxx entre el método de elementos finitos y Python.

1.3.2 Objetivos específicos

Seleccionar programas libres capaces de realizar simulaciones de tratamientos térmicos en los aceros en estudio.

Describir el modelo matemático y obtener los diagramas TTT de los aceros seleccionados mediante el uso del software Python.

Describir el modelo matemático y obtener los diagramas CCT de los aceros seleccionados con el software Python.

Comparar los resultados obtenidos por Python con Matlab.

1.4 Justificación del problema

Esta investigación se realizó con la finalidad de elaborar una herramienta computacional funcional para graficar las curvas TTT de los aceros de baja aleación al aplicarle un temple en agua, observando así el comportamiento de estos mediante un simulador, de tal manera que esta investigación podrá ser de gran ayuda desde el punto de vista académico e industrial, puesto que se evita el uso de ensayos, materiales, laboratorios entre otros; que involucran dinero, espacio y tiempo; que son factores importantes a considerar una empresa para tener una producción más eficiente.

1.5 Alcance

Se elaboró una simulación computacional con el software Python el cual admitió representar las curvas TTT y CCT de los aceros de baja aleación AISI 4xxx y 5xxx al aplicarle un temple en agua y posteriormente la comparación con los resultados obtenidos por este simulador y el software utilizado en elementos finitos. Estas comparaciones no fueron realizadas con ensayos de laboratorio.

CAPÍTULO II

MARCO TEÓRICO.

Este aspecto viene referido al soporte principal del estudio. Ampliándose la descripción o planteamiento del problema, permitiendo integrar la teoría con la investigación y establecer sus interrelaciones, de manera coordinada y coherente de conceptos que abordan el problema.

2.1 Antecedentes:

Hoy en día se conoce varios estudios afines con la predicción de las propiedades de los aceros y la simulación de la misma. Por ende desde la perspectiva metodológica, estos trabajos sirven de base para la ejecución y estudio del tema en particular. Es conveniente destacar el trabajo realizado por Salas M (2017), donde efectuó un trabajo de investigación titulado "**Simulación de tratamiento térmico mediante temple en agua en aceros de baja aleación AISI 4xxx y 5xxx empleando el método de elementos finitos**". Universidad José Antonio Páez, San Diego, Venezuela. Se ha realizado una simulación de temple aplicado a los aceros 4027, 4340 y 5140, así como la obtención de las curvas de transformación isotérmica TTT y continua CCT mediante la implementación de los modelos matemáticos y elementos finitos en los programas de simulación.

El aporte de este trabajo fue significativo debido a la realización de la simulación del tratamientos térmico mediante temple, además de la utilización de aceros 4027,4340 y 5140.

Por otra parte, Anders Malthe-Sorensen (2016), titulado "**Elementary thermal physics using Python**". Departamento de física, Universidad de Oslo, Noruega. Se ha llevado a cabo un estudio sobre el uso de cálculos simbólicos en Python tanto en la exposición como en los ejemplos de problemas físicos. Esto es para presentar una herramienta útil en su contexto natural. Utilizan la caja de

herramientas simbólica en Python para conservar todo dentro de un solo marco de programación.

El estudio citado tiene relación por cuanto al uso del programa Python como herramienta para resolver problemas físicos, en donde se encuentra la física térmica, que puede servir como base para los cálculos próximos a realizar con Python.

Del mismo modo, Petr Krysl (2017), realizó un trabajo de investigación, titulado “**Finite element modeling with abaqus and Python for thermal and stress analysis**”. Universidad de California, San Diego, Estados Unidos De América. El software utilizado en dicho manual consta de un programa de elementos finitos con una interfaz gráfica de usuario, Abaqus / cae., y el lenguaje de código abierto orientado a objetos Python.

La pertinencia de esta investigación citada es, la realización de un manual para instruir a los usuarios el uso de lenguajes de programación para el análisis térmico y de estrés en los materiales metálicos.

Este texto forma parte de una serie de textos en los que cada texto introduce conceptos computacionales clave relevantes para el tipo particular de física introducido en el texto. Para la física térmica y estadística, introduciremos nuevos conceptos en modelos algorítmicos y estocásticos, modelos de dinámica molecular, descripciones estadísticas de grandes conjuntos de datos, necesidades de métodos de optimización para la minimización de energías libres y métodos de modelado continuo necesarios para la teoría cinética y el transporte.

2.2 Bases teóricas.

2.2.1 Generalidades de las aleaciones ferrosas.

Según Callister (2002) las aleaciones ferrosas, es la unión homogénea del elemento hierro como su principal metal de aleación, y con otro material con el destino de mejorar sus propiedades. Según la cantidad de carbono, las aleaciones ferrosas comprenden tres grupos: hierro, acero y fundición. El hierro puro posee aceros contienen entre 0,008% C y 2,11% C, aunque los aceros pueden llegar a tener

hasta 2,11%C, rara vez se excede del 1% en la práctica. Las fundiciones son las aleaciones ferrosas con contenido de carbono entre 2,11% y 6,7%. Las aleaciones ferrosas, basadas en aleaciones hierro-carbono, incluyen los aceros al bajo carbono, los aceros aleados y para herramientas, los aceros inoxidable y los hierros fundidos. Éstos son los materiales más ampliamente utilizados en el mundo. Los aceros que son aleaciones ferrosas son los más importantes debido a que su costo es relativamente bajo y la variedad de aplicaciones que posee por sus propiedades mecánicas. (Ver figura 1).

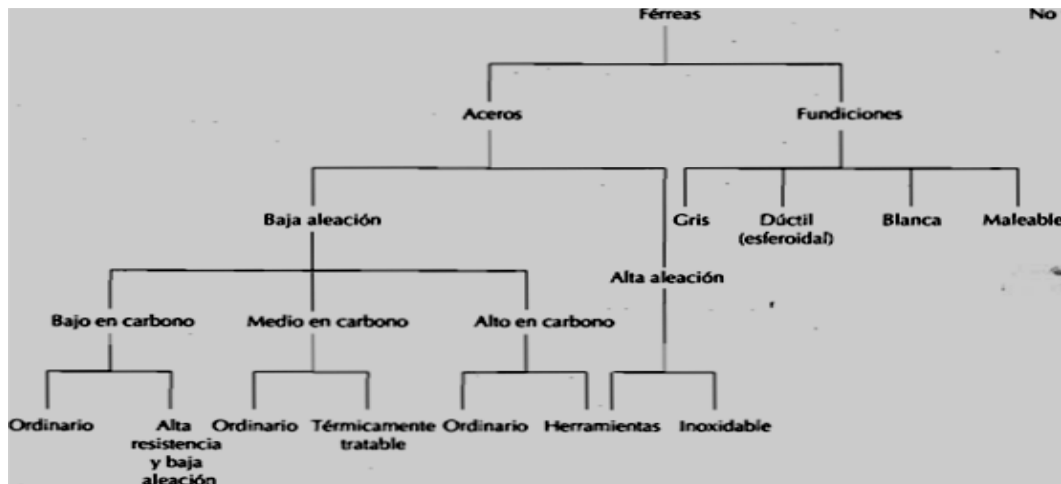


Figura 1. Clasificación de las aleaciones ferrosas.

Fuente: William D. Callister (2002).

2.2.2 Generalidades de los aceros de baja aleación.

Fundinox Chile (2010) menciona que a principios de la década de 1930, fueron desarrollados los aceros de baja aleación (HSLA, por sus siglas en inglés), tienen un bajo contenido de carbono (en general menos de 0.30%) y la sumatoria de elementos aleantes menor al 5%. Se caracterizan por una microestructura consistente en una fase de ferrita de grano fino y una segunda fase de martensita y austenita.

Este tipo de materiales han sido desarrollados y utilizados extensivamente para requerimientos especiales donde no es posible utilizar aceros al carbono corrientes con poca capacidad de endurecimiento. Al aumentar el contenido de elementos aleantes se consiguen varias mejoras tales como: aumentar el grado de

endurecimiento, aumentar su resistencia al desgaste, proporcionar resistencia al impacto, lograr buena maquinabilidad incluso con alta durezas, mejorar las propiedades mecánicas en temperaturas elevadas o muy bajas y lograr una resistencia a la corrosión superior a la de un acero al carbono ordinario. Estos materiales son producidos para encontrar cifras de resistencia a la tracción del orden de 485 a 1380mpa junto con algunas características especiales.

Escobar(2012) enuncia que los aceros de baja aleación se clasifican en función del porcentaje en C:

Bajo en C, si tiene menos de un 0,25% de C

Medio en C, si tiene entre un 0,25 y un 0,6% de C

Alto en C, si tiene entre un 0,6 y un 1,4% de C.

2.2.3 Denominaciones de los aceros.

Orozco (2015) indica que tradicionalmente, el “American Iron and Steel Institute (AISI)” y la “Society of Automotive Engineers (SAE)” han denominado los aceros al carbono y aleados utilizando cuatro dígitos. Los dos primeros dígitos indican los elementos aleantes y sus porcentajes; y los dos últimos indican el contenido de carbono en peso (Ver figura de la 2 a la 4).

DESIGNACIÓN	TIPO
10XX	Aceros ordinarios al carbón
11XX	Aceros al carbono re sulfurados de fácil maquinado
13XX	Aceros con 1.75% de Mn (1.5-2%)
15XX	Aceros al manganeso (1.0-1.65%)
23XX	Aceros al níquel, 3.5% de Ni (3.25-3.75%)
25XX	Aceros al níquel, 5% de Ni (4.75-5.25%)
31XX	Aceros al níquel-Cromo, 1.25% Ni y 0.65% Cr
33XX	Aceros al níquel-Cromo, 3.5% Ni y 1.60% Cr
40XX	Aceros al molibdeno, 0.25% Mo.
41XX	Aceros con Cr (0.4-1.2%), Mo (0.08-0.25%)
43XX	Aceros al Ni-Cr-Mo (1.8% Ni, 0.65% Cr, 0.25% Mo)
44XX	Molibdeno, (0.4-0.53%)

Continuación.

DESIGNACIÓN	TIPO
45XX	Molibdeno, (0.55%)
46XX	Níquel- Molibdeno, (1.8%Ni, 0.2% Mo)
47XX	Níquel- Cromo- Molibdeno, (1.05%Ni, 0.45%Cr, 0.2% Mo)
48XX	Níquel- Molibdeno, (3.5%Ni, 0.25% Mo)
50XX	Aceros al Cromo (bajo cromo, 0.28-0.40%)
51XX	Medio Cromo, (0.8-1.05%)
50XXX	Acero resistente al desgaste, 0.5%Cr
51XXX	Acero resistente al desgaste, medio Cr 1%
52XXX	Acero resistente al desgaste, alto Cr 1.45%
61XX	Aceros al Cromo-Vanadio, (0.75% Cr, 0.15%V)
8XXX	Aceros de triple aleación
81XX	0.3%Ni, 0.4%Cr, 0.12%Mo
86XX	0.55%Ni, 0.50%Cr, 0.20%Mo
87XX	0.55%Ni, 0.50%Cr, 0.25%Mo
88XX	0.55%Ni, 0.50%Cr, 0.35%Mo
92XX	Acero al Silicio-Manganeso, (2%Si y 0.8%Mn)
93XX	Aceros de triple aleación, 3.25%Ni, 1.2%Cr, 0.12%Mo
98XX	Aceros de triple aleación, 1%Ni, 0.8%Cr, 0.25%Mo
XXBXX	Aceros con Boro, (mínimo 0.0005% B)
50BXX	0.5%Cr
51BXX	0.8%Cr
81BXX	0.3%Ni, 0.45%Cr, 0.12%Mo
XXBVXX	Acero al Boro-Vanadio
XXLXX	Acero con plomo
XXXH	Acero con banda de templabilidad
EX	Nuevos tipos de acero con designación temporal

Figura 2. Designaciones de los aceros según AISI-SAE

Fuente: "Orozco A. (2015)"

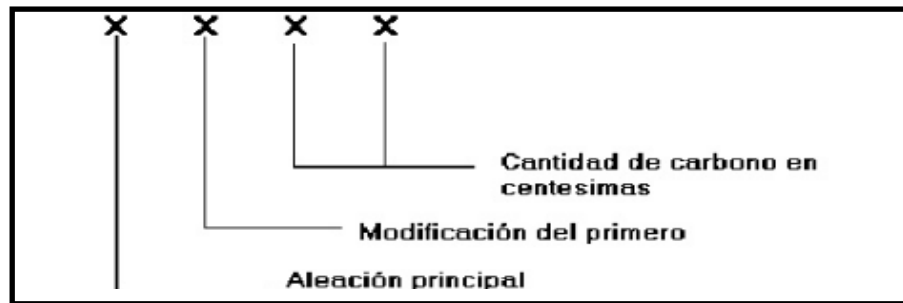


Figura 3. Designación según normas AISI-SAE

Fuente: "Orozco A. (2015)"

IDENTIFICADOR	ALEANTE
1.	MANGANESO
2.	NIQUEL
3.	NIQUEL-CROMO Principal aleante el cromo.

Continuación

IDENTIFICADOR	ALEANTE
4.	MOLIBDENO.
5.	CROMO.
6.	CROMO-VANADIO Principal aleante el cromo.
7.	NIQUEL-CROMO-MOLIBDENO Principal aleante el molibdeno.
8.	NIQUEL-CROMO-MOLIBDENO Principal aleante el níquel.

Figura 4. Convención del primer dígito según normas AISI-SAE.

Fuente: “Orozco A. (2015)”

Se observa entonces que si el primer número es 1 se sabe que es un acero al carbono; si el dígito siguiente es el 0, o sea que la designación es 10xx, se trata de un acero ordinario al carbono; así 1030 significa un acero ordinario al carbono con 0,30% c. Si el segundo dígito es uno, la designación es 11xx y significa que se trata de un acero re sulfurado, es decir, se le añadió azufre para hacerlo más maquinable. Si el segundo dígito es 3, la designación es 13xx y se trata de un acero con contenido de manganeso ente 1,5 y 2%. Si el primer dígito es 2, se trata de acero al níquel, por ejemplo el acero SAE 23xx.

Schmid Kalpakjian (2008) señala que las denominaciones de la “American Society for Testing and Materials”(ASTM) son otro sistema de numeración que incorpora las denominaciones de AISI y SAE e incluye especificaciones estándar para los productos de acero. Para los metales ferrosos, la denominación consiste en la letra “a” seguida de números arbitrarios (en general tres).

Por otro parte, al sistema de numeración actual se le conoce como “sistema unificado de numeración (ver figura 5) (UNS, por sus siglas en inglés)” y ha sido ampliamente adoptado por las industrias de metales ferrosos y no ferrosos. Consta de

una letra que indica la clase general de la aleación, seguida de cinco dígitos que denominan su composición química. Las letras utilizadas para denominación son:

G: para aceros AISI y SAE al carbono y aleados.

J: para aceros fundidos.

K: para aceros diversos y aleaciones ferrosas.

S: para aceros inoxidable y superaleaciones.

T: para aceros para herramientas.

Designación ^a		Composición (% en peso) ^b		
AISI/SAE o número ASTM	Número UNS	C	Mn	Otros
Aceros bajos en carbono				
1010	G10100	0,10	0,45	
1020	G10200	0,20	0,45	
A36	K02600	0,29	1,00	0,20 Cu (mín)
A516 Grado 70	K02700	0,31	1,00	0,25 Si
Aceros de baja aleación y alta resistencia				
A440	K12810	0,28	1,35	0,30 Si (máx), 0,20 Cu (mín)
A633 Grado E	K12002	0,22	1,35	0,30 Si, 0,08 V, 0,20 N, 0,03 Nb
A656 Grado 1	K11804	0,18	1,60	0,60 Si, 0,1 V, 0,20 Al, 0,015 N

Figura 5. Designaciones y composiciones de cinco aceros bajos en carbono y de tres aceros de baja aleación y alta resistencia.

Fuente: adaptada de Metals Handbook: properties and selección, iron and steels, vol 1, 9 edición b.

Bardes, American Society for Metals, 1978, pp. 185,407

2.2.4 Efectos de diversos elementos en los aceros.

Schmid Kalpakjian (2008) enuncia que a los aceros se agregan diversos elementos para impartirles propiedades como templabilidad, resistencia, dureza, tenacidad, resistencia al desgaste y capacidad para trabajarlos, soldarlos y maquirarlos. En general, cuanto mayor porcentaje de estos elementos contengan los aceros, mayores serán sus propiedades particulares. Por ejemplo, a mayor contenido de carbono, mayor será la templabilidad del acero y tendrá más resistencia, dureza y resistencia al desgaste. En cambio, la ductilidad, soldabilidad y tenacidad se reducen con el aumento del carbono.

Azufre: mejora la maquinabilidad cuando se combina con manganeso; reduce la resistencia al impacto y la ductilidad, además de dañar la calidad de la superficie y la soldabilidad.

Boro: mejora la templabilidad con pérdida (o incluso con alguna mejora) en la maquinabilidad y la formabilidad.

Calcio: desoxida los aceros, mejora la tenacidad y puede mejorar la formabilidad y la maquinabilidad.

Carbono: mejora la templabilidad, resistencia, dureza y resistencia al desgaste; reduce la ductilidad, la soldabilidad y la tenacidad.

Cerio: controla la forma de las inclusiones y mejora la tenacidad en los aceros de baja aleación de alta resistencia; desoxida los aceros.

Cobalto: mejora la resistencia y la dureza a temperaturas elevadas.

Cobre: mejora la resistencia a la corrosión atmosférica y, en menor medida, incrementa la resistencia con una pequeña pérdida de ductilidad; afecta el trabajo en caliente y la calidad de la superficie.

Cromo: mejora la tenacidad, templabilidad y resistencia al desgaste, a la corrosión y a la alta temperatura; incrementa la profundidad de penetración de dureza resultante del tratamiento térmico, al promover la carburización.

Fósforo: mejora la resistencia, templabilidad, resistencia a la corrosión y maquinabilidad; reduce severamente la ductilidad y la tenacidad.

Magnesio: tiene los mismos efectos del cerio.

Manganeso: mejora la templabilidad, resistencia, resistencia a la abrasión y maquinabilidad; desoxida el acero fundido, reduce la fragilización en caliente y disminuye la soldabilidad.

Molibdeno: mejora la templabilidad, resistencia al desgaste, tenacidad, resistencia a la temperatura elevada, resistencia a la termofluencia y dureza; minimiza la fragilización por revenido.

Niobio (): refina el tamaño del grano y mejora la resistencia y tenacidad al impacto; reduce la temperatura de transición y puede disminuir la templabilidad.

Níquel: mejora la resistencia, tenacidad y resistencia a la corrosión; mejora la templabilidad.

Plomo: mejora la maquinabilidad; provoca la fragilización por metal líquido.

Selenio: mejora la maquinabilidad.

Silicio: mejora la resistencia, dureza, resistencia a la corrosión y conductividad eléctrica; disminuye la pérdida por histéresis magnética, maquinabilidad y formabilidad en frío.

Tantalio: tiene efectos similares a los del niobio.

Telurio: mejora la maquinabilidad, formabilidad y tenacidad.

Titanio: mejora la templabilidad; desoxida los aceros.

Tungsteno: tiene los mismos efectos que el cobalto.

Vanadio: mejora la resistencia, tenacidad, resistencia a la abrasión y dureza a temperaturas elevadas; inhibe el crecimiento del grano durante el tratamiento térmico.

Zirconio: tiene los mismos efectos que el cerio.

2.2.5 Diagrama hierro-carbono (fe-c)

Anibal Trujillo (2017) señala que para entender los tratamientos térmicos de los aceros es necesario del conocido diagrama de fases fe-c; tanto las propiedades de los aceros y fundiciones de las aleaciones al carbono, como la clase de microestructura que estas posean, pueden ser entenderse y explicar en dicho diagrama, así pues, la relación que existe entre una fase y otra, se puede expresar en términos del contenido de carbono existente en esta, por lo que nos lleva a la consideración de que existe un sistema estable (hierro-grafito) y otro sistema metaestable (hierro-carbono).

Debido a que el primero requiere una gran cantidad de tiempo para desarrollarse, sobre todo en el lugar de la baja temperatura y del bajo porcentaje de

carbono la industria utiliza el segundo sistema. Tomando como referencia el diagrama hierro-carbono, los aceros son aquellos comprendidos con un contenido de carbono menor o igual al 2,1% de este; sin embargo, éste varía dependiendo del autor, por otro lado las fundiciones de hierro poseen contenidos de carbono superiores al 2,1% hasta aproximadamente 6,67% del mismo. De lo anterior expuesto, se puede entonces dividir hasta en 2,1% de carbono en tres (3) clasificaciones, las cuales son: aceros de bajo contenido de carbono o aceros hipoeutectoides (con un contenido de carbono entre 0.08% y 0.25%), aceros de medio contenido de carbono o aceros eutectoides (con un contenido de C entre 0.3% y 0.77%) y los aceros de alto contenido de carbono o hipereutectoides (con un rango de %C de 0.77% a 2.1%).

En dicho día

conocida como ferrita, la cual posee un máximo de solubilidad de 0.03% de carbono

conoce como austenita, cuya solubilidad máxima es de 2.1% a aproximadamente 1148°C. La solubilidad en el diagrama hierro-carbono se ve afectada por la temperatura, esto sucede tanto en la austenita como en la ferrita, la última fase existente en éste que es la cementita, la cual es un compuesto intermetálico cuya composición se representa como Fe_3C , el cual se encuentra al 6.67% de carbono. Existen también entre los diferentes campos zonas en las cuales se mezclan algunas fases como lo son la combinación ferrita-cementita, ferrita-austenita y austenita-cementita, a medida que se aumenta la temperatura, se puede encontrar la zona en la cual existe la fase líquida y por debajo de esta la mezcla de algunas otras fases más líquida, líquido-cementita y líquido-austenita, siendo la fase líquida una fase no deseada a alcanzar en los tratamientos térmicos.

También se hallan algunos límites importantes en los campos de una sola fase y a estos se les han asignado nombres los cuales facilitan en gran medida la forma de hablar de ellos, entre ellos están:

A1: también llamada línea de temperatura eutectoide, es la temperatura mínima para que exista austenita.

A3: es el límite inferior existente de la temperatura de austenización de los contenidos de bajo peso de carbono.

Acm: es el límite existente para los aceros de alto contenido de carbono en peso, esta línea se encuentra justo frente la línea a3.

Mf: es la temperatura final que existe para la transformación de la martensita mientras esta se enfría.

Ms: temperatura inicial para que la austenita se convierta en martensita.

En caso de agregar elementos aleados al acero, las posiciones de los límites A_{cm} , A_1 , A_3 Y la composición del punto eutectoide varían. Es importante el denotar que todas las aleaciones importantes disminuyen el contenido de carbono necesario para el punto eutectoide, también que, elementos que estabilizan la austenita como el manganeso y el níquel disminuyen la temperatura A_1 Y los elementos estabilizantes de la ferrita como el cromo, silicio, molibdeno y el tungsteno logran el efecto contrario, aumentan la temperatura A_1 . Mediante cálculos termodinámicos modernos se permite la determinación de los cambios que afectan la fuerza motriz de la transformación de fase. Estos métodos también permiten los cálculos de diagramas ternarios y de diagramas de fase de orden superior, incluyendo carburos aleados (Ver figura 6)

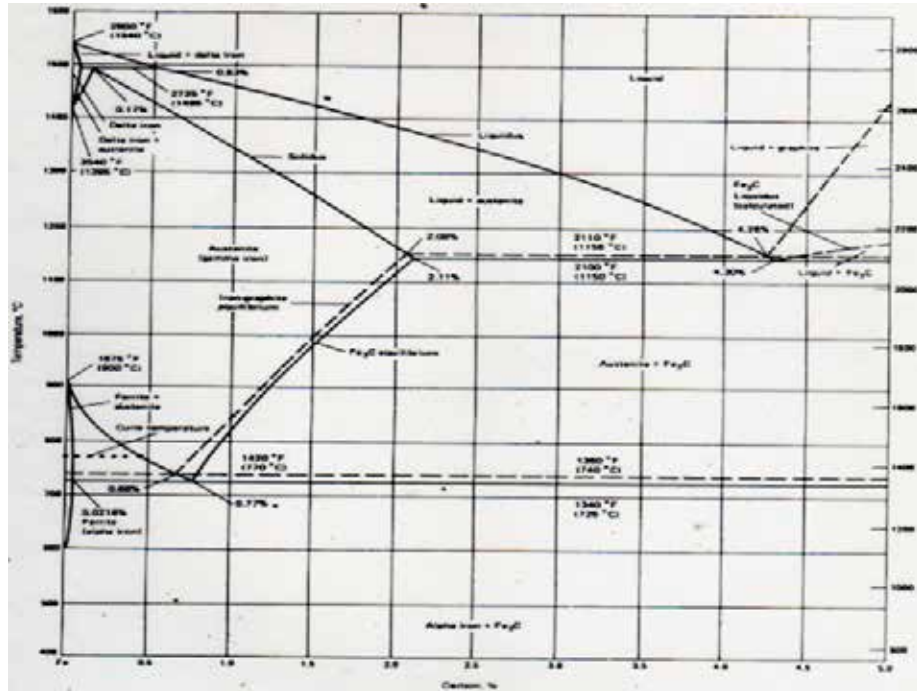


Figura 6. Diagrama de fases hierro-carbono.

Fuente: Universidad tecnológica de Pereira, 2013

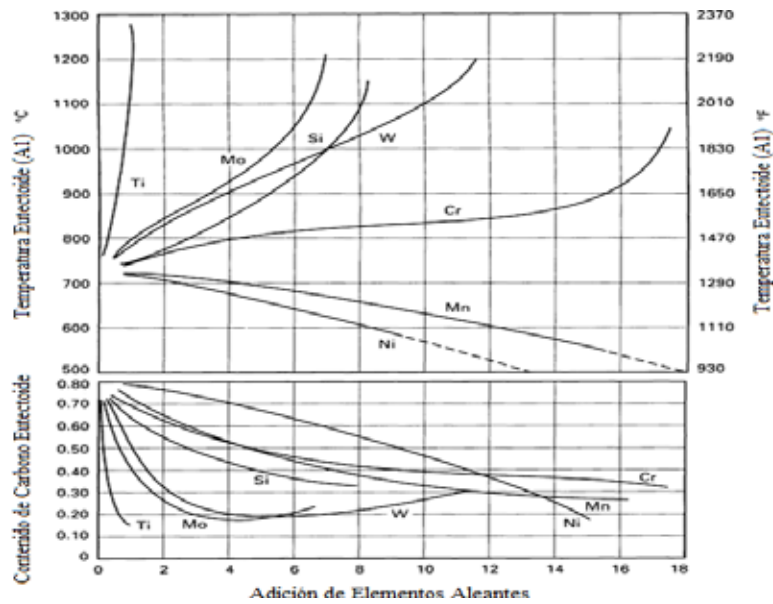


Figura 7. Influencia de los elementos aleantes en la temperatura eutectoide A_1 Y el contenido de carbono del punto eutectoide.

Fuente: ASM International, 1991.

En la figura 7 se aprecian otros puntos de importancia, los cuales son:

Reacción eutectoide: la cual ocurre estando en el estado sólido aproximadamente a los 723°C y a 0.77% de C, pasando de tener austenita pura a una composición bifásica de ferrita alfa y cementita; la composición de estas dos fases se le conoce como perlita.

Reacción peritectica:

líquido, obteniendo de esta austenita.

Reacción eutéctica: se produce a los 1148°C y a 4.3% de C, llevándose a cabo en estado líquido y transformándose así en una composición bifásica de austenita y cementita.

2.2.6 Tratamientos térmicos.

Callister (2002) Apunta que los tratamientos térmicos tienen por objeto mejorar las propiedades y características de los aceros, y consisten en calentar y mantener las piezas o herramientas de acero a temperaturas adecuadas, durante un cierto tiempo y enfriarlas luego en condiciones convenientes. De esta forma, se modifica la estructura microscópica de los aceros, se verifican transformaciones físicas y a veces hay también cambios en la composición del metal.

El tiempo y la temperatura son los factores principales y hay que fijarlos siempre de antemano, de acuerdo con la composición del acero, la forma y el tamaño de las piezas y las características que se desean obtener.

Las herramientas y las piezas de máquina tienen que tener una dureza, una permanencia de poder cortante y una resistencia adecuada a la utilización que han de tener. El filo de un cincel, por ejemplo, ha de estar enteramente templado. Estas diferentes propiedades pueden obtenerse mediante la elección de un material apropiado y de un conveniente tratamiento térmico. Estos tratamientos inducen las transformaciones de fase que influyen en gran medida en propiedades mecánicas como la resistencia, dureza, ductilidad, tenacidad y resistencia al desgaste de las aleaciones. Los efectos del tratamiento térmico dependen de la aleación en particular,

su composición y su microestructura, el grado de trabajo previo en frío, así como las velocidades de calentamiento y de enfriamiento durante el tratamiento térmico.

Los tratamientos térmicos más usados son: el recocido, temple, normalizado, revenido, etc.

2.2.7 Etapas del tratamiento térmico.

Antonio(2009) plantea que un tratamiento térmico consta de tres etapas que se presentan a continuación:

Calentamiento hasta la temperatura fijada: la elevación de temperatura debe ser uniforme en la pieza.

Permanencia a la temperatura fijada: su fin es la completa transformación del constituyente estructural de partida. Puede considerarse suficiente una permanencia de unos 2 minutos por milímetro de espesor.

Enfriamiento: este enfriamiento tiene que ser rigurosamente controlado en función del tipo de tratamiento que se realice.

2.2.8 Tipos de tratamientos térmicos.

Existen varios tipos de tratamientos térmicos y estos cambian primordialmente en el método de enfriamiento de la pieza. En la mayoría de los casos el calentamiento y el mantenimiento es el mismo. A continuación se explican los tratamientos más utilizados en las industrias.

2.2.8.1 Recocido.

Schmid Kalpakjian (2008) expresa que el recocido es un término general utilizado para describir la restauración de una aleación trabajada en frío o tratada térmicamente en sus propiedades originales. El objetivo principal de este tratamiento es ablandar el acero; otras veces también se desea además regenerar su estructura o eliminar tensiones internas. Por ejemplo, el recocido se puede utilizar para aumentar la ductilidad (y de ahí su formabilidad) y reducir la dureza y la resistencia, o para modificar su microestructura. El proceso de recocido también se utiliza para relevar esfuerzos residuales en una parte manufacturada, así como para mejorar la

maquinabilidad y la estabilidad dimensional. Consisten en calentamientos a temperaturas adecuadas, seguidos generalmente de enfriamientos lentos. Las diferentes clases de recocidos que se emplean en la industria se pueden clasificar en tres grupos: recocidos con austenización completa, recocidos suscritos y recocidos con austenización incompleta (ver figura 8).

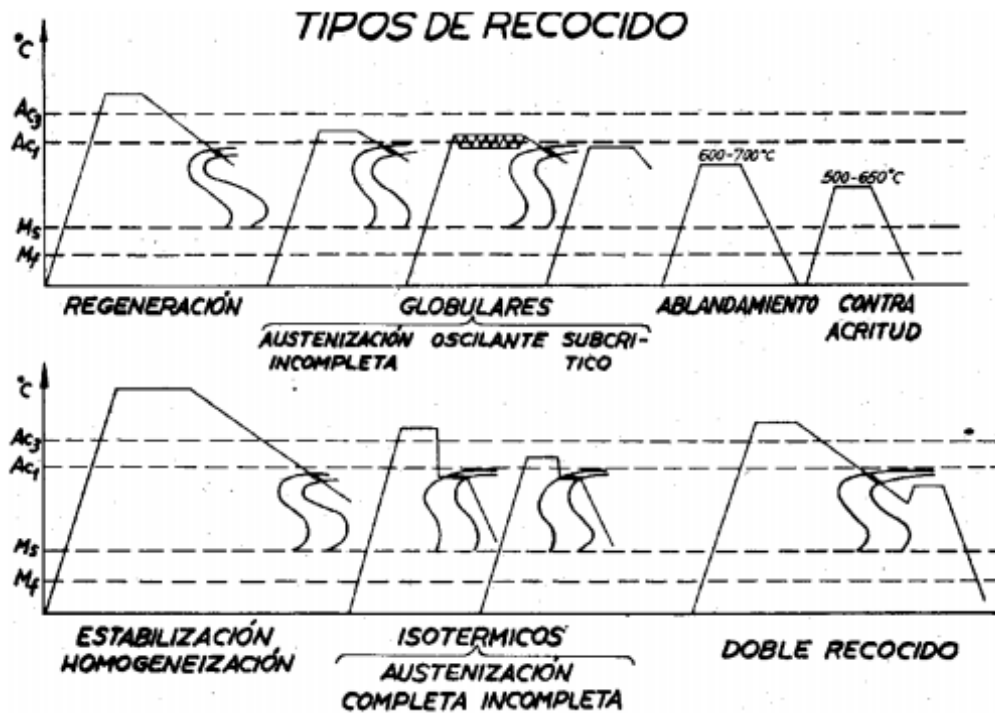


Figura 8. Tipos de recocido.

Fuente. Francisco Blanco Alvarez (2017)

2.2.8.2 Temple.

La Escuela Colombiana de Ingeniería (2008) señala que el temple es un tratamiento térmico que tiene por objetivo aumentar la dureza y resistencia mecánica del material, transformando toda la masa en austenita con el calentamiento y después, por medio de un enfriamiento brusco (con aceites, agua o salmuera), se convierte en martensita, que es el constituyente duro típico de los aceros templados. En el temple, es muy importante la fase de enfriamiento y la velocidad alta del mismo, además, la temperatura para el calentamiento óptimo debe ser siempre superior a la crítica para

poder obtener de esta forma la martensita. Existen varios tipos de temple, clasificados en función del resultado que se quiera obtener y en función de la propiedad que presentan casi todos los aceros, llamada templabilidad (capacidad a la penetración del temple), que a su vez depende, fundamentalmente, del diámetro o espesor de la pieza y de la calidad del acero.

Factores que influyen en el temple.

Mandujano (2009) indica que la velocidad de enfriamiento de una muestra depende de la rapidez de eliminación de energía térmica, que es función a su vez de:

1. Composición del acero.
2. Temperatura a la que hay que calentar.
3. Tiempo de calentamiento.
4. Velocidad de enfriamiento.
5. Características del medio donde se realiza temple.
6. Tamaño y geometría de la muestra.

Medios de temple.

Mandujano (2009) dice que existen diversos medios entre los cuales los más comunes son el agua, aceite y aire.

Para el caso de este trabajo de grado nos interesa el medio de temple en agua. Es un medio rápido de enfriamiento, se consiguen temples muy fuertes; es necesaria una cantidad de agua suficiente para que la temperatura de la misma no suba de 30°C. Las piezas se han de agitar dentro del medio acuoso, para evitar la formación de burbuja de vapor alrededor de la pieza, que actuaría de aislamiento térmico y retardaría el enfriamiento.

2.2.8.3 Normalizado.

Mandujano (2009) explica que este tratamiento consiste en un calentamiento a temperatura ligeramente más elevada que la crítica superior, seguido de un enfriamiento en aire tranquilo. De esta forma, se deja el acero con una estructura y propiedades que arbitrariamente se consideran como normales y características de su

composición. Se suele utilizar para piezas que han sufrido trabajos en caliente, trabajos en frío, enfriamientos irregulares o sobrecalentamientos, y también sirve para destruir los efectos de un tratamiento anterior defectuoso. Por medio del normalizado se eliminan las tensiones internas y se uniformiza el tamaño del grano del acero. Se emplea casi exclusivamente para los aceros de construcción al carbono o de baja aleación.

Esquema de los tratamientos térmicos más empleados y del desplazamiento de las zonas de transformación. Los puntos y zonas negras señalan los puntos de transformación del acero (ver figura 9).

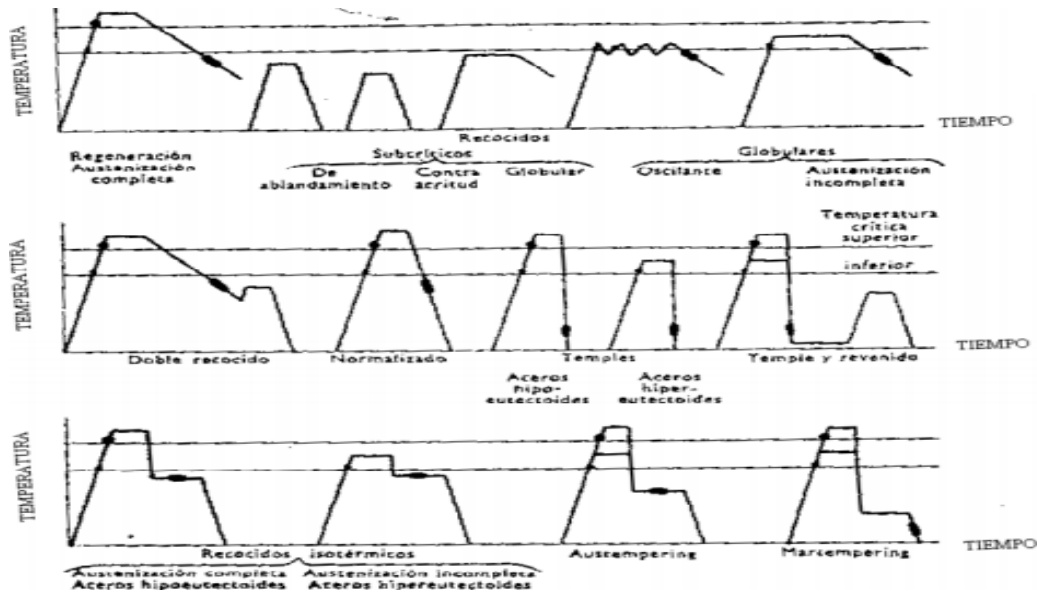


Figura 9. Esquema de los tratamientos térmicos más empleados y del desplazamiento de las zonas de transformación.

Fuente: Mandujano (2009)

2.2.9 Diagrama TTT y CCT

Diagrama TTT (tiempo-temperatura-transformación).

La Escuela Colombiana de Ingeniería (2008) manifiesta que este tipo de diagrama muestra cómo la velocidad de enfriamiento afecta la transformación de austenita en varias fases posibles, las cuales pueden ser: (1) formas alternativas de ferrita y cementita, (2) martensita. El tiempo se presenta logarítmicamente a lo largo

del eje horizontal y la temperatura en el eje vertical. Esta curva se interpreta partiendo del tiempo t_0 (pocos segundos transcurridos) en la región austenita y continúa hacia abajo y a la derecha a lo largo de la trayectoria que muestra cómo se enfría el metal en función del tiempo (ver figura 10 y 11). Los tiempos de transformación dependen de la aleación del material.

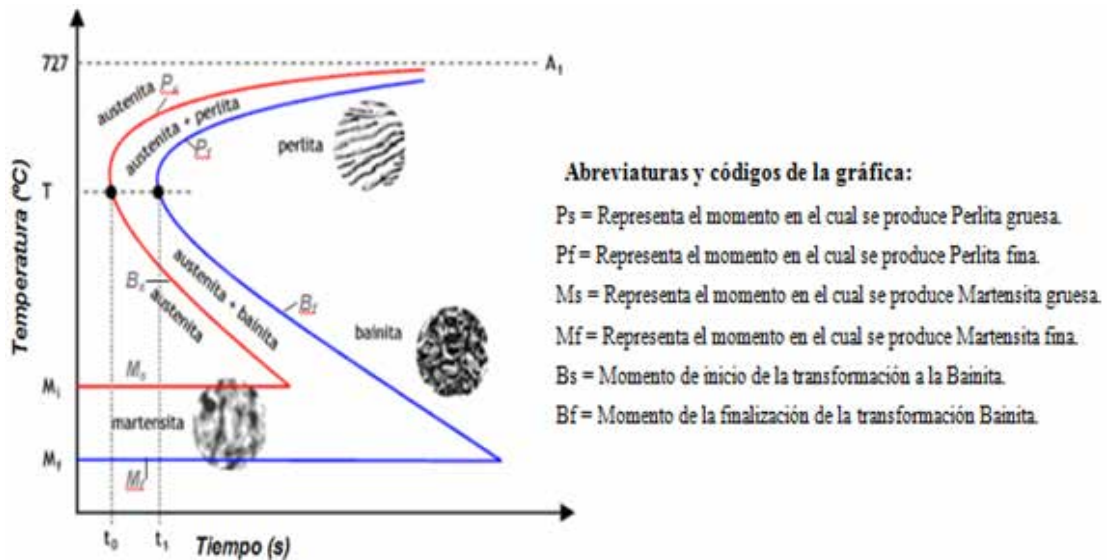


Figura 10. Diagrama TTT para un acero

Fuente: ASM International (1991)

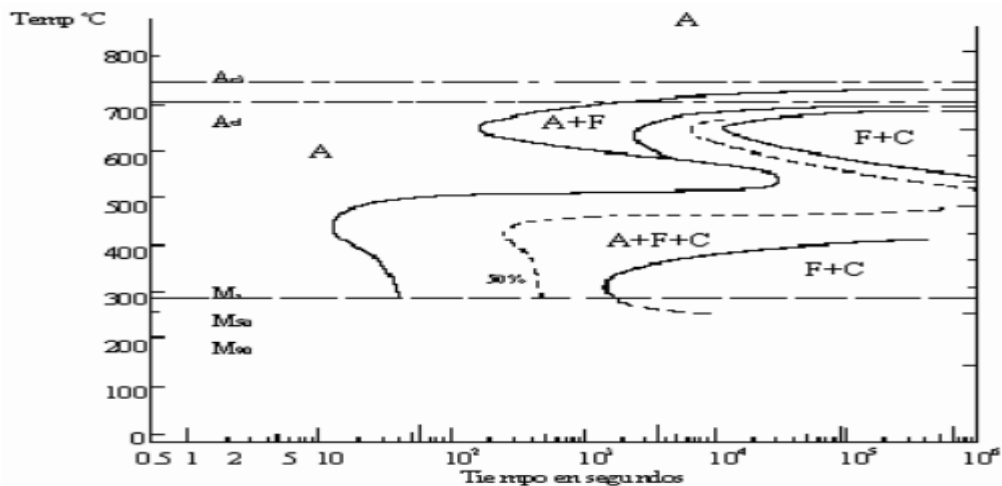


Figura 11. Diagrama TTT con la influencia de elementos aleantes.

Fuente: ASM International (1991)

Diagrama CCT

Felipe (2012) establece que los tratamientos térmicos en su gran mayoría implican un enfriamiento continuo del material, lo que lleva a una limitación en el uso de los diagramas TTT, ya que no proporcionan información correcta en el caso de enfriamientos continuos. De aquí surge la necesidad de tener diagramas de transformación con enfriamiento continuo o curvas CCT (continuous cooling transformation). El método más usual y más directo para obtener las curvas CCT es por dilatometría. La representación gráfica de las transformaciones en condiciones de enfriamiento continuo, se realiza en un diagrama temperatura contra el logaritmo del tiempo, en el cual se muestra además: las curvas que representan gráficamente las velocidades de enfriamiento, la indicación sobre las curvas del porcentaje aproximado de transformación, la indicación de las distintas zonas de fases de acuerdo con lo siguiente: a = austenita, f = ferrita, c = carburo y m = martensita, la línea punteada, de trazos cortos, que indica un 50 % de fase transformada y una indicación en la parte inferior del diagrama de la dureza que adquiere el metal de acuerdo a cada curva de enfriamiento.

2.2.10 Influencia de los elementos de aleación sobre los diagramas TTT y CCT.

Influencia sobre los diagramas TTT.

Felipe (2012) señala que el principal papel de los elementos de aleación sobre los aceros es el de aumentar su templabilidad, lo que se refleja en los diagramas TTT en un desplazamiento hacia la derecha, de las curvas de inicio y fin de la transformación austenítica. Este efecto se observa para todos los elementos de aleación, excepto para el cobalto y el carbono (para porcentajes mayores de 0.8 %), los cuales desplazan la curva hacia la derecha. Otro efecto que se observa, es la separación de las rodillas perlítica y bainítica.

Influencia sobre los diagramas CCT.

Felipe (2012) apunta que lo mencionado en la sección anterior, referente a la influencia de los elementos de aleación sobre la transformación isotérmica sigue

siendo válido para las transformaciones en enfriamiento continuo. En este punto es importante mencionar que respecto al metal Fe en su “forma pura” (sin adición de

- genos).

- genos tenemos: Ni, Mn, Co, Pr, Pd, Ru, Rh, Os, Ir, C, N, Cu,

- genos tenemos: Cr, W, Mo, V, Ti, Si, Al, P, Be,

As, Sn Y Sb. De manera general, los elementos gamágenos como el níquel o el manganeso (con excepción del cobalto) desplazan la curva CCT hacia la derecha y hacia abajo. Los elementos alfégenos como el cromo, el molibdeno o el vanadio, tienen una influencia adicional específica sobre las dos zonas de transformación, perlítica y bainítica.

2.2.11 Generalidades de la simulación.

Jose Zetino (2014) indica que con el advenimiento de la computadora, una de las más importantes herramientas para analizar el diseño y operación de sistemas o procesos complejos es la simulación. Aunque la construcción de modelos arranca desde el renacimiento el uso moderno de la palabra simulación data de 1940, cuando los científicos Von Neuman y Ulan que trabajaban en el proyecto “monte-carlo”, durante la segunda guerra mundial, resolvieron problemas de reacciones nucleares cuya solución experimental sería muy cara y el análisis matemático muy complicado.

La experimentación con las simulaciones se realiza dando entradas al modelo y analizando sus salidas. Para describir las simulaciones deben tenerse en cuenta: el tipo de sistema real que es representado en el modelo, así como la fidelidad del modelo y sus características internas:

El sistema real puede ser físico, artificial o hipotético.

La fidelidad puede ser física y de percepción (cómo se ve, oye y siente), o fidelidad de manipulación (lo que puede hacerse en la simulación).

Las características internas se refieren a la estructura y relaciones de las variables en el modelo del sistema simulado.

Para programar las simulaciones existen diferentes herramientas de software, el uso de cualquiera de ellos supone ventajas y limitaciones con respecto al resto:

Lenguajes de programación de uso general: entre sus ventajas se encuentra que son conocidos por los programadores, lo que supone menor esfuerzo de aprendizaje. Están disponibles en cualquier sistema operativo, a diferencia de los lenguajes de simulación. Algunos tienen extensiones con bibliotecas para manejar tareas que se necesitan de manera general en la programación de simulaciones. Son más eficientes desde el punto de vista de la velocidad, portabilidad y flexibilidad, pero el tiempo de desarrollo de las aplicaciones es más largo que en lenguajes y herramientas específicas.

Lenguajes de simulación: el entorno de desarrollo es específico para simulaciones; están diseñados para facilitar la tarea de programar este tipo de aplicaciones, proporcionando automáticamente la mayoría de los elementos necesarios en la programación de modelos de simulación. En general dan al usuario un conjunto de conceptos de modelado para describir el sistema y un sistema de programación para convertir la descripción en un programa que ejecuta la simulación, además de facilidades para la detección automática de errores potenciales que ya han sido identificados. Ahorran tiempo en el desarrollo. Producen un código más legible, modificable y menos largo, lo que minimiza el número de errores.

Paquetes de simulación: proporcionan bloques de construcción ya programados que el usuario puede seleccionar y ensamblar en el orden que necesita para formar el modelo del sistema particular que desea construir, lo que facilita mucho esa tarea con respecto a los lenguajes, ya que no es necesario programar. Su principal ventaja es el ahorro de tiempo, y la desventaja es su poca flexibilidad ya que solo permiten simular los sistemas para los que fueron definidos.

La simulación por computadora se ha convertido en una parte útil del modelado de muchos sistemas naturales en física, química y biología, y de sistemas humanos como el económico, entre otros. Pueden sustituir por completo, o ser utilizada como un accesorio, en sistemas de modelado para los cuales no es posible tener soluciones analíticas de forma cerrada simple.

En la enseñanza, esta posibilidad de ensayar, probar teorías y entrenarse en un ambiente controlable y sin riesgos es una gran oportunidad.

2.2.11.1 Definición de simulación

La palabra “simulación” posee un contexto bastante amplio, pero entre las definiciones que la rodean, una muy aceptable es la formulada por Thomas H. Naylor, (1982) la define como “una técnica para conducir experimentos en una computadora digital”. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos de mundo real a través de largos periodos de tiempo.

Maisel, H. y Gnugnoli, G., (1972) definen la simulación como “una técnica numérica para realizar experimentos en una computadora digital”. Estos experimentos involucran ciertos tipos de modelos matemáticos y lógicos que describen el comportamiento de sistemas de negocios, económicos, sociales, biológicos, físicos o químicos a través de largos períodos de tiempo.

Otro estudioso del tema como Shannon (2012) define la simulación como “el proceso de diseñar y desarrollar un modelo computarizado de un sistema o proceso y conducir experimentos con este modelo con el propósito de entender el comportamiento del sistema o evaluar varias estrategias con las cuales se puede operar el sistema”.

2.2.11.2 Etapas para realizar un estudio de simulación

Jose Zetino (2014) menciona que los pasos necesarios para llevar a cabo un experimento de simulación son los siguientes:

Definición del sistema: para tener una definición exacta del sistema que se desea simular, es necesario hacer primeramente un análisis preliminar del mismo, con el fin de determinar la interacción del sistema con otros sistemas, las restricciones del sistema, las variables que interactúan con otros sistemas, las restricciones del sistema, las variables que interactúan dentro del sistema y sus interrelaciones, las medidas de efectividad que se van a utilizar para definir y estudiar el sistema y los resultados que se esperan obtener del estudio.

Formulación del modelo: una vez que están definidos con exactitud los resultados que se esperan obtener del estudio, el siguiente paso es definir y construir el modelo con el cual se obtendrán los resultados deseados. En la formulación del modelo es necesario definir todas las variables que forman parte de él, sus relaciones lógicas y los diagramas de flujo que describan en forma completa al modelo.

Colección de datos: es posible que la facilidad de obtención de algunos datos o la dificultad de conseguir otros pueda influenciar el desarrollo y formulación del modelo. Por consiguiente, es muy importante que se defina con claridad y exactitud o datos que el modelo va a requerir para producir los resultados deseados. Normalmente, la información requerida por un modelo va a requerir para producir los resultados deseados.

Implementación del modelo en la computadora: con el modelo definido, el siguiente paso es decidir si se utiliza algún lenguaje y obtener los resultados deseados.

Validación: una de las principales etapas de un estudio de simulación es la validación. A través de esta etapa es posible detallar las deficiencias en la formulación del modelo o de los datos alimentados al modelo. Las formas más comunes de validar un modelo son:

1. La opinión de expertos sobre los resultados de la simulación.
2. La exactitud con que se predicen datos históricos.

3. La exactitud en la predicción del futuro.
4. La comprobación de falla del modelo de simulación al utilizar datos que hacen fallar al sistema real.
5. La aceptación y confianza en el modelo de la persona que hará uso de los resultados que arroje el experimento de simulación.

Experimentación: la experimentación con el modelo se realiza después de que éste ha sido validado. La experimentación consiste en generar datos deseados y en realizar análisis sensibles de los índices requeridos.

Interpretación: en esta etapa del estudio, se interpretan los resultados que arroja la simulación y en base a esto se toma una decisión.

Documentación: dos tipos de documentación son requeridos para hacer mejor uso del modelo de simulación. La primera se refiere a la documentación del tipo técnico, es decir a la documentación que el departamento de procesamiento de datos debe tener del modelo. La segunda se refiere al manual del usuario, con el cual se facilita la interacción y el uso del modelo desarrollado, a través de una terminal de computadora.

2.2.11.3 Software a evaluar para la simulación.

Extenso número de herramientas computacionales existe hoy en día para fines de estudio del análisis térmico; como lo es Mono, Scala, wxWidgets y Python aunque la mayoría de esos programas funcionan y cumplen con el propósito que se busca realizar; su comprensión, su desarrollo en solo algunas plataformas, manuales en otros idiomas diferente al español o información acerca de estos es muy limitada; por ende de los programas mencionados se reduce drásticamente.

Mono: versión libre de la plataforma de desarrollo de Microsoft, basada en los estándares de ECMA (European Computer Manufacturers Association) y el CLR (Common Language Runtime), el entorno en tiempo de ejecución de lenguaje común. Es preciso tener antes el MS Visual C++ para poder instalarla.

Scala: que posee características de la programación funcional y la orientada y requiere Java; CLISP (para Common Lisp), que contiene un intérprete, compilador, debugger y otros instrumentos.

WxWidgets: una biblioteca C++ para elaborar interfaces de usuario gráficas(GUI) que es posible compilar en multitud de plataformas. Cuenta también con wxFormBuilder, su generador visual de interfaces gráficas.

Guido van Rossum (2009) Señala que Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

El intérprete de Python y la extensa biblioteca estándar están a libre disposición en forma binaria y de código fuente para las principales plataformas desde el sitio web de Python, y puede distribuirse libremente.

Python también puede usarse como un lenguaje de extensiones para aplicaciones personalizables.

2.2.12 Definición de modelo matemático.

Duran(2006) define que un modelo matemático es una descripción, en lenguaje matemático, de un objeto que existe en un universo no-matemático. Estamos familiarizados con las previsiones del tiempo, las cuales se basan en un modelo matemático meteorológico; así como con los pronósticos económicos, basados éstos en un modelo matemático referente a economía.

La mayoría de las aplicaciones de cálculo (por ejemplo, problemas de máximos y mínimos) implican modelos matemáticos. En la industria abundan procesos térmicos o termo – mecánicos durante los cuales se producen distorsiones difíciles de predecir y controlar. Las diferentes velocidades de enfriamiento ocasionadas a lo largo de la pieza durante el proceso producen una distribución de fases con volumen específico y propiedades mecánicas diferentes, que generan tensiones internas y

dichas distorsiones. Incorporar la evolución microestructural a la simulación permitiría definir procesos industriales optimizados

2.2.12.1 Modelos matemáticos a evaluar.

Para la realización de la simulación aplicada en el trabajo de grado se han implementado diferentes modelos como: el modelo de Johnson -Mehl -Avrami – Kolmogorov (JMAK) para determinar la evolución de las transformaciones de fase gobernadas por la difusión y el modelo de Koistinen - Marburger para la transformación martensítica. Para obtener el tiempo de incubación de las transformaciones no isotérmicas y el crecimiento de las propias fases se ha utilizado la Regla de Adición. Los resultados que se han obtenidos hacen concluir que los modelos no siempre describen fielmente las transformaciones de fase. Generalmente, la ferrita y la perlita quedan bastante bien modeladas, sin embargo, para la bainita, tanto el tiempo de incubación como la fracción transformada no quedan bien descritos.

La descomposición de la austenita en las diferentes fases no se inicia inmediatamente al bajar la temperatura por debajo de su zona de estabilidad, es necesario un tiempo para que empiecen a formarse los núcleos de la nueva fase y empiecen a crecer; este periodo es la llamada etapa de incubación. En el momento en que éstos núcleos se han formado y han crecido lo suficiente para ser observados, que por convenio se ha tomado como el momento en que el volumen de la nueva fase alcanza el valor del 1%, termina la etapa de incubación y empieza la transformación,

viene descrito por las curvas TTT en los procesos isotérmicos, y por las curvas CCT en los no isotérmicos. En aceros hipoeutectoides, la Austenita puede transformar mediante difusión en ferrita, perlita y bainita, y por transformación no difusiva en Martensita.

Estas ecuaciones de JMAK precisan conocimiento previo de las curvas TTT. Para poder ser aplicadas a la simulación se deben utilizar los modelos matemáticos de estas. El modelo de Kirkaldy es uno de los diferentes modelos matemáticos existentes

para determinar las curvas TTT mediante cálculos y se escogió este por tener claramente definidos las bases para los cálculos requeridos, además de que los parámetros empíricos pueden ser ajustados y controlados fácilmente.

Modelo de transformación de Kirkaldy: Un modelo de cálculo para la ferrita y perlita fue presentado inicialmente por Kirkaldy, siguiendo ecuaciones desarrollados por Zener y Hiller. Ese modelo inicial representaba la forma global de las curvas de enfriamiento sin diferenciar las regiones difusivas y no difusivas.

una fracción (x) de la austenita a una temperatura dada (T). El modelo se extendió para incluir las curvas de enfriamiento de la perlita y de la bainita y también permitir un cálculo general de la cantidad de transformación como función de tiempo a cierta temperatura. Esto proporciono tres ecuaciones para la cantidad de transformación de ferrita (f), perlita (p) y bainita (B).

$$f = \frac{60.\%Mn+2.\%Ni+68.\%Cr+244.\%Mo}{6 \cdot 2^{\frac{N}{8}} \cdot T^3 D_f} I \quad (1)$$

$$p = \frac{1.8+5.4(\%Cr+\%Mo+4.\%Mo.\%Ni)}{6 \cdot 2^{\frac{N}{8}} \cdot T^3 D_p} I \quad (2)$$

$$B = \frac{(2.3+10 \ \%C+4 \ \%Cr+19 \ \%Mo) \ 10^{-4}}{6 \cdot 2^{\frac{N}{8}} \cdot T^2 e^{\left(-\frac{27500}{R T}\right)}} I \quad (3)$$

Donde la cantidad de cada elemento está provisto en %peso. El término N asume la incubación transitoria, por tanto, para la superficie del grano, el tiempo del volumen transformado varía desde el tamaño hasta $\frac{1}{4}$ de este (debido a ello el término $2^{\frac{N}{8}}$). Entonces los parámetros críticos son el tamaño de grano, que debe ser conocido para cada caso, la temperatura de estabilización de las fases y un parámetro Df y Dp que se dan a continuación.

$$D_f = e^{\frac{-23500}{R T}} \quad (4)$$

$$\frac{1}{D_p} = \frac{1}{e^{\frac{-27500}{R T}}} + \frac{0.5 \ Mo}{e^{\frac{-37000}{R T}}} \quad (5)$$

El parámetro I al final de las ecuaciones es la fracción volumétrica integral.

$$I = \int_0^X \frac{dx}{x^{\frac{2(1-x)}{3}} (1-x)^{2x/3}} \quad (6)$$

Transformaciones difusivas: la ecuación JMAK es aplicable a las transformaciones difusivas isotermales, describe la evolución del volumen de fase transformada ($V\{t\}$) en función del tiempo mediante una función sigmoideal:

$$V\{t\} = 1 - e^{-b(T)t^{n(T)}} \quad (7)$$

Donde $b(T)$ y $n(T)$ son parámetros que deben determinarse empíricamente a través de la curva TTT del material, este modelo funciona para la transformación de austenita para ferrita, perlita y bainita. Aunque para la Ferrita se toma el valor de $n=1$. De forma generalizada se puede establecer que:

$$n = \frac{\log\left[\frac{\ln(1-V_s)}{\ln(1-V_f)}\right]}{\log\left[\frac{t_s(T)}{t_f(T)}\right]} \quad (8)$$

$$b = \frac{-\ln(1-V_s)}{t_s(T)} \quad (9)$$

Los sufijos S y f indican inicio y final respectivamente. En los diagramas TTT, V_s se toma como 1% y V_f 99%, pero cualquier otro porcentaje puede ser tomado. Al simular las curvas en forma de C la computadora, guarda el diagrama como una función spline. La curva de enfriamiento es aproximada a una escalera y cada escalón es igual a un paso de tiempo. La transformación de la fase es entonces calculada de manera isotérmica en cada escalón (Ver figura 12).

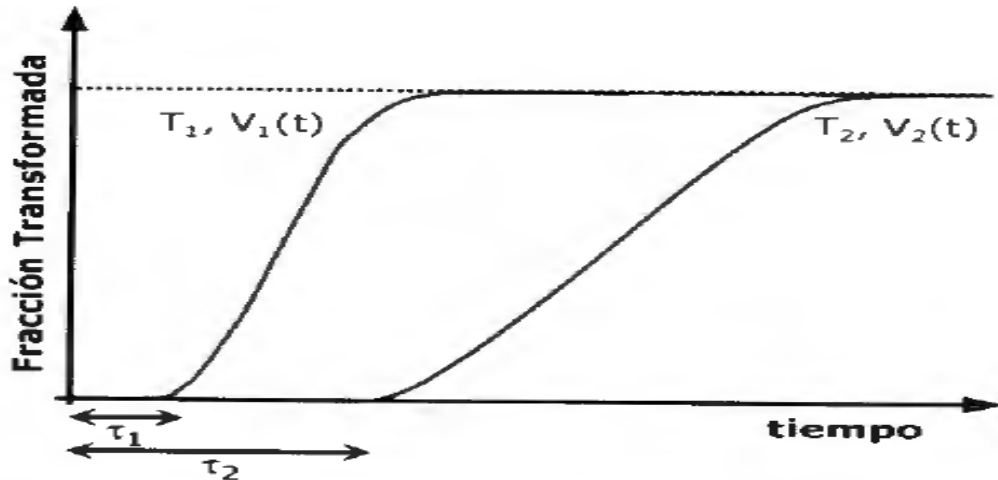


Figura 12. Representación de la ecuación JMAK ($V \{t\}$) en función del tiempo para dos (2) temperaturas diferentes (T_1 y T_2) y los tiempos de incubación para las dos (2) temperaturas (t_1 y t_2).

Fuente: Simulación de un tratamiento térmico mediante elementos finitos, Giménez M., Liobet A., Riera M. y Prado J., 2013.

La curva TTT de un acero, muestra precisamente el tiempo de inicio (tiempo de incubación, tiempo para el cual la transformación ha alcanzado el 1%) y el fin de la transformación (momento en que la transformación ha alcanzado el 99%) en los procesos isotérmicos; (Ver figura 13). Un vez conocida la TTT del material y operando con la misma ecuación JMAK se obtienen los parámetros $b(T)$ y $n(T)$ para cada temperatura.

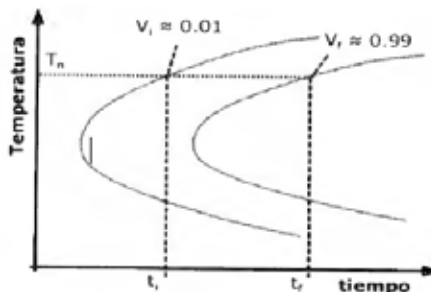


Figura 13. Curva TTT de un Acero. Se indican los tiempos t_i y t_f a una cierta temperatura T_n .

Fuente: Simulación de un tratamiento térmico mediante elementos finitos, Giménez M., Liobet A., Riera M. y Prado J., 2013.

Transformación Martensítica: en el caso de la transformación martensítica, se ha utilizado el modelo de Koistinen - Marburger. En este modelo el volumen de austenita transformado a martensita depende únicamente del grado de n subenfriamiento a partir de la temperatura de inicio de la transformación martensítica (M_s).

$$V_m = 1 - e^{[-0,011(M_s - T)]} \quad (10)$$

Procesos no isotérmicos: en el caso de las transformaciones difusivas no isotérmicas se ha aplicado la Regla de Adición (o ley de Scheil). Esta consiste en una discretización de los procesos convirtiendo los no isotérmicos en múltiples procesos isotérmicos infinitesimales. De igual forma que para el proceso real, el cálculo de las transformaciones de fase también viene dividido en dos etapas: etapa de nucleación y etapa de crecimiento. Durante la etapa de nucleación se calcula el tiempo de incubación de cada una de las fases y en la etapa de crecimiento, el volumen transformado. La etapa de nucleación se rige por la ecuación (8); Regla de Adición para la nucleación. Si el periodo de incubación de una fase a temperatura T_i durante un proceso isotérmico es i^{TTT} , una muestra mantenida a esa temperatura durante un tiempo de t_i segundos ($t_i < i^{TTT}$), completará una fracción del tiempo de incubación igual a t_i / i^{TTT} .

En el momento en que la suma de todas esas fracciones sea igual a la unidad (1), la fase empezará a crecer, y se habrá cumplido el tiempo de incubación de la transformación no isotérmica (i^{CCT}). Por ejemplo, si se inicia la descomposición de la Austenita. A partir de la TTT del material se obtendrán los valores de las i^{TTT} para las temperaturas de cada uno de los i incrementos (Ver figura 14).

Aplicando la Regla de Adición al tiempo de incubación se obtiene un diagrama CCT teórico a partir de la TTT empírica. Durante la etapa de crecimiento la Regla de Adición se aplica de forma diferente. La transformación no isotérmica se describe como la suma de una serie de transformaciones isotérmicas y a cada transformación isotérmica le corresponde una curva JMAK. La fracción transformada para cada

incremento de tiempo se calcula con la ecuación JMAK correspondiente a la temperatura en ese incremento de tiempo.

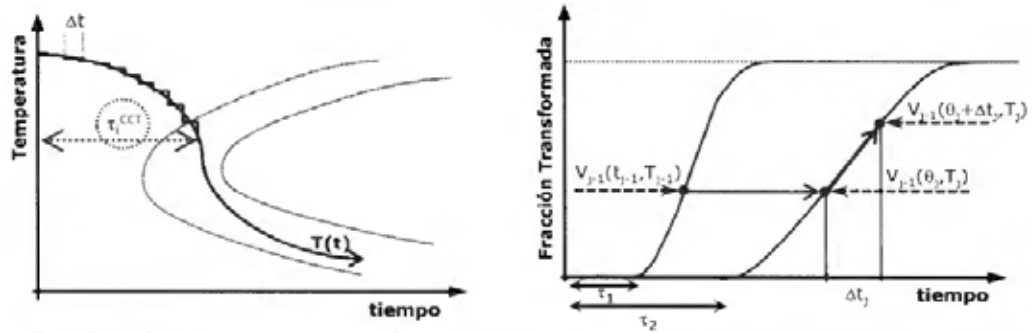


Figura 14. Aplicación de la Regla de Adición. Izquierda: tiempo de incubación de la transformación no isotérmica (Ti^{CCT}). Derecha: Crecimiento de las fases.

Fuente: Simulación de un tratamiento térmico mediante elementos finitos, Giménez M., Liobet A., Riera M. y Prado J., 2013.

$$\int_0^{CCT} \frac{dt}{TTT} = 1 \quad \text{o} \quad \sum_{i=1}^n \frac{\Delta t}{iTTT} = 1 \quad (11)$$

Al inicio de cada incremento, j , se tiene una cierta fracción de volumen V_{j-1} igual a la correspondiente al final del incremento anterior, $j-1$. Asimismo, durante este incremento, la temperatura (T_j) es constante y le corresponde una curva de crecimiento isoterma de la fase $V_j(t, T_j)$. Se introduce entonces un tiempo virtual, θ_j (11), que corresponde al instante para el que, durante un proceso isoterma a temperatura T_j se tendría la fracción de volumen V_{j-1} , correspondiente al final del incremento anterior con temperatura T_{j-1} . Es decir $V_j(\theta_j, T_j) = V_{j-1}$. Se calcula a continuación la fracción de volumen, V_j correspondiente al instante $\theta_j + \Delta t_j$ según la ecuación $V_j(t, T_j)$. A saber, $V_j = V_j(\theta_j + \Delta t_j, T_j)$ (Ecuación 11); donde Δt_j es la duración del incremento j . Se partirá entonces de esta V_j para calcular el crecimiento en el siguiente incremento a T_{j+1} así sucesivamente hasta el final del proceso.

$$\theta_j = \left[\frac{\ln(1-V_{i-1}^{j-1})}{b} \right]^{\frac{1}{n}} \quad (12)$$

$$V_{i+1}^j = 1 - e^{-b (\theta_{j+1} + \Delta t_j)^n} \quad (13)$$

El Volumen de fracción transformada está normalizado a uno (1) de forma que la suma del volumen de todas las fases siempre dará uno (1). Ecuación (14).

$$V_{Austenita} + V_{Ferrita} + V_{Perlita} + V_{Bainita} + V_{Martenita} = 1 \quad (14)$$

CAPÍTULO III

MARCO METODOLÓGICO

En este capítulo se presentan los aspectos formales que definen la naturaleza y factores que precisan el método a ser utilizado en este proyecto. Para ello se definirá el tipo, diseño y nivel de la investigación, así como aquellos elementos que fundamentan la recolección y procesamientos de los datos que permitieron el desarrollo de conclusiones.

3.1 Tipo de investigación.

Sierra Bravo (1991) dice que genéricamente, la investigación es una actividad del hombre orientada a descubrir algo desconocido. Baray (2006) dice que la investigación comparativa se utiliza para evaluar las semejanzas y diferencias de corrientes, pensamientos, autores y teorías. En este sentido se comprende, que es una investigación comparativa por ser el estudio de un problema con el propósito de ampliar y profundizar el conocimiento de su naturaleza, comparando resultados del mismo tema con trabajos previos que se diferencian por distintos medios de obtención de dichos resultados.

3.2 Diseño de la investigación.

López (2016) dice que el diseño de investigación constituye el plan general del investigador para obtener respuestas a sus interrogantes o comprobar la hipótesis de investigación. El diseño de investigación desglosa las estrategias básicas que el investigador adopta para generar información exacta e interpretable. La Universidad de Jaen (2015) define a la investigación documental como aquella que se realiza a través de la consulta de documentos (libros, revistas, periódicos, memorias, anuarios, registros, códigos, constituciones, etc.) La investigación de la misma se puede definir como documental, pues se desarrollara mediante modelos matemáticos, simulaciones en Python de información extraída de documentos externos.

3.3 Nivel de la investigación.

Según Arias (2006) el nivel de investigación se refiere al grado de profundidad con que se aborda un fenómeno u objeto de estudio:

1. **Investigación exploratoria:** La investigación exploratoria es aquella que se efectúa sobre un tema u objeto desconocido o poco estudiado, por lo que sus resultados constituyen una visión aproximada de dicho objeto, es decir, un nivel superficial de conocimientos
2. **Investigación descriptiva:** La investigación descriptiva consiste en la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura o comportamiento. Los resultados de este tipo de investigación se ubican en un nivel intermedio en cuanto a la profundidad de los conocimientos se refiere.
3. **Investigación explicativa:** La investigación explicativa se encarga de buscar el porqué de los hechos mediante el establecimiento de relaciones causa-efecto. En este sentido, los estudios explicativos pueden ocuparse tanto de la determinación de las causas (investigación post facto), como de los efectos (investigación experimental), mediante la prueba de hipótesis. Sus resultados y conclusiones constituyen el nivel más profundo de conocimientos.

Se puede definir que el nivel de la investigación es de tipo descriptiva, se caracterizaran hechos simulados para estudiar sus resultados.

3.4 Población y muestra

Todo estudio implica, en la fase de diseño, la determinación del tamaño poblacional y muestral necesario para su ejecución. Así mismo, Stracuzzi (2003), define a la población de una investigación como el conjunto de unidades de las que se desea obtener información y sobre las que se van a generar conclusiones. La población puede ser definida como el conjunto finito o infinito de elementos, personas o cosas pertinentes a una investigación En esta investigación la población y muestra se limita a todo aquel que tenga acceso al trabajo de investigación y el programa de simulación Python.

3.5 Técnicas e instrumentos de recolección de datos

Para llevar a cabo la investigación se utilizarán tres instrumentos fundamentales, uno de ellos es el análisis documental, la cual “se trata del acopio de los antecedentes relacionados con la investigación, se realiza por la consulta de documentos escritos, sean formales o no, en los que se plasmará un conocimiento que fue avalado por autores que realizaron una investigación previa”.

Por otra parte la observación experimental “elabora datos en condiciones relativamente controladas por el investigador, particularmente porque éste puede manipular la o las variables. Tamayo (2008) dice que es una poderosa técnica de investigación científica. Puede utilizar como instrumento la hoja o ficha de registro de datos.”.

El tercer instrumento es el programa Python “Guido Van Rossum. (2009), dice que Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.”.

3.6 Fases de la investigación

En este capítulo se describirá el procedimiento ejecutado durante el desarrollo de la investigación el cual fue desplegado en cuatro fases las cuales proporcionarán el cumplimiento a los objetivos específicos ya mencionados.

Fase I: Seleccionar programas libres capaces de realizar simulaciones de tratamientos térmicos.

Para la realización de esta fase fue necesario realizar una revisión de software libres para el estudio de tratamientos térmicos, para así escoger entre ellos el que proporcione la misma salida tomando en cuenta la fácil comprensión, manejo del programa e información acerca de este. Dando como resultado que Python fue la

mejor opción pues utilizan las ecuaciones de gobierno mencionadas en el marco teórico para la obtención de los diagramas TTT y CTT de cada uno de los aceros en estudio.

Fase II: Describir el modelo matemático y obtener los diagramas TTT de los aceros seleccionados mediante el uso del software Python.

En esta segunda fase se procedió a obtener el diagrama de enfriamiento implementando los modelos matemáticos Kirkaldy para el TTT el cual se comparó con las establecidas por la USS.

Fase III: Describir el modelo matemático y obtener los diagramas CCT de los aceros seleccionados con el software Python.

En esta fase se describió el modelo matemático a utilizar para la CCT que viene siendo el de JMAK con la regla de la adición. Posteriormente se comenzó a realizar la simulación del temple de los aceros escogidos a través del programa seleccionado.

Fase IV: Comparar los resultados obtenidos por Python con Matlab.

En la cuarta fase se estableció las semejanzas y diferencias tanto cualitativa como cuantitativa entre el programa estudio y la herramienta computacional utilizada para la aplicación del método de elemento finito.

CAPÍTULO IV

RESULTADOS

Cumpliendo con los procesos metodológicos de esta investigación, se realizaron simulaciones de temple en agua de aceros 4XXX y 5XXX., mediante la obtención de las curvas de transformación isotérmica TTT y continua CCT a través del uso de modelos matemáticos y el software Python. Para estudiar estos resultados se escogieron tres tipos de aceros (AISI 4027, el AISI 4340 y el AISI 5140) pertenecientes a la familia nombrada anteriormente.

4.1 Seleccionar programas libres capaces de realizar simulaciones de tratamientos térmicos en los aceros en estudio.

Para la selección de los software libres se buscó aquellos que permitieran el análisis de los tratamientos térmicos y dando como solución los diagramas de TTT y CCT. Para ello se investigaron varios de los software comerciales más utilizados como Mono, Scala, wxWidgets y Python, se estudiaron sus características, donde se seleccionó el que tuviese un lenguaje de programación sencillo y fuesen lo bastante popular para ser considerado un buen aporte investigativo a cualquier persona que decida hacer uso del mismo. El resultado de esto fue la selección de Python por que presentan una interfaz fácil de entender y un sistema de ayuda que rectifica cualquier duda que se haya tenido; además existen gran información del mismo en inglés.

4.2 Aplicación del modelo matemático, obtención de los diagramas TTT de los aceros seleccionados mediante el uso del software Python y comparación con la USS.

Luego de haber seleccionado el software Python, se le aplicó el modelo matemático de Kirkaldy a cada uno de los aceros en estudio, el cual permitió tener como resultado la obtención de las curvas de transformaciones isotérmicas (TTT) de los diferentes microconstituyentes, como está descrito en el capítulo 2 de la presente

investigación (Ecuaciones 1, 2 y 3). Para lograrlo se inició estableciendo las constantes necesarias las cuales se muestran (Ver tabla 1).

Tabla 1. Propiedades de los aceros en estudio para los diagrama TTT.

PROPIEDADES	ACERO 4027	ACERO 4340	ACERO 5140
R (ctte de gases)	8,31447	8,31447	8,31447
N (tamaño grano)	7	7,5	5,5
%C	0,26	0,42	0,41
%Mn	0,087	0,78	0,75
%Cr	0	0,8	1,05
%Si	0,15	0	0,1
%Mo	0,26	0,33	0
%Ni	0	1,79	0
Tp (°C)	670	655	655
Tf (°C)	760	710	725

Fuente:(CAMPILLO & SEQUERA, 2018).

Posteriormente se procede a establecer las integrales (Figura 16) que determinan las posiciones de las curvas en el eje x; y luego determinar el resultado de las ecuaciones de Kirkaldy permitiendo obtener los tiempos de transformación a las diferentes temperaturas (Figura 17).

```

#Se resuelve la integral de Kirkaldy
x=Symbol('x')
integrate(1/(x**(2*(1-x)/3)*(1-x)**(2*x/3)),x)
f=lambda x:1/(x**(2*(1-x)/3)*(1-x)**(2*x/3))
I=quad(f,0,0.99)
I2=quad(f,0.,0.1)
I3=quad(f,0.,0.5)

```

Figura 16. Integral de la ecuación de Kirkaldy aplicada al programa del acero AISI 4027.

Fuente: (CAMPILLO & SEQUERA, 2018)

Las ecuaciones utilizadas en el programa tienen una diferencia clara con respecto a las originales de Kirkaldy, y esta es, que se multiplican por un factor elevado ya sea a la 6 o a la 5, la razón de este factor es que al utilizar la ecuación normal resultaba una falla de precisión del modelo, puesto que daba como resultado tiempos increíblemente bajos. Al emplear el factor se lograron adecuar los tiempos de transformación con los valores de las curvas USS.

```

while Ti<860:

#Ecuacion para transformacion de austenita a Ferrita
Df=math.e**(-23500/(R*Ti))
taf.append(((60*Mn+2*Ni+68*Cr+244*Mo)/(6**2*(N/8)*(Tf-(Ti))**3*Df)*I*10**5)
taf2.append(((60*Mn+2*Ni+68*Cr+244*Mo)/(6**2*(N/8)*(Tf-(Ti))**3*Df)*I2*10**4)

#Ecuacion para la transformacion de Austenita a Perlita
sp=1/(math.e**(-27500/(R*Ti)))+(0.5*Mo)/(math.e**(-37000/(R*Ti)))
Dp=sp**-1
tap.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2*(N/8)*(Tp-Ti)**3*Dp)*I*10**6)
tap2.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2*(N/8)*(Tp-Ti)**3*Dp)*I2*10**5)
tap3.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2*(N/8)*(Tp-Ti)**3*Dp)*I3*10**5)

#Ecuacion para la transformacion de Austenita a Bainita
if Ti < Tb:
    tab.append(((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2*(N/8)*(Tb-Ti)**2*math.e**(-27500/(R*Ti)))*I*10**6)
    tab2.append(((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2*(N/8)*(Tb-Ti)**2*math.e**(-27500/(R*Ti)))*I2*10**5)
    tab3.append(((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2*(N/8)*(Tb-Ti)**2*math.e**(-27500/(R*Ti)))*I3*10**5)
    p.append(Ti)
Ti2.append(Ti)
Ti=Ti+2

```

Figura 17. Ecuación de Kirkaldy para la ferrita, perlita y bainita para el acero AISI 4027.

Fuente: (CAMPILLO & SEQUERA, 2018).

Ya establecidas las ecuaciones en el programa, se procedió finalmente a elaborar las curvas con los resultados de cada uno de las ecuaciones establecidas (Ver figura 18).

```
#Procedimiento para graficar linea Ms
x=1
y=0
n=1
q=list()
z=list()

while y<10**4:
    x=n+1
    n=n+1
    y=y+1
    q.append(x)
    z.append(Ms)

#Grafica
plt.xscale('log')
plt.ylim(0, 1000)
plt.title('(Diagrama TTT Acero AISI 4027)')
plt.plot(taf2, Ti2, 'y-', label = "Ferrita") # Dibuja el gráfico
plt.xlabel("Tiempo (s)") # Inserta el título del eje X
plt.ylabel("Temperatura (°C)") # Inserta el título del eje Y
plt.ioff() # Desactiva modo interactivo de dibujo
plt.plot(tap, Ti2, 'b-', label = "Perlita") # No dibuja datos de lista2
plt.ion() # Activa modo interactivo de dibujo
plt.plot(tap, Ti2, 'b-') # Dibuja datos de lista2 sin borrar datos de lista1
plt.ioff()
plt.plot(tap2, Ti2, 'b-')
plt.ion()
plt.plot(tap2, Ti2, 'b-')
plt.ioff()
plt.plot(tab, p, 'k-')
plt.ion()
plt.plot(tab, p, 'k-', label = "Bainita")
plt.ioff()
plt.plot(tab2, p, 'k-')
plt.ion()
plt.plot(tab2, p, 'k-')
plt.ioff()
plt.plot(tab3, p, 'k--')
plt.ion()
plt.plot(tab3, p, 'k--')
plt.ioff()
plt.plot(q, z)
plt.ion()
plt.plot(q, z)
plt.legend()
```

Figura 18. Formulación de las gráficas en el programa para el acero AISI 4027.

Fuente: (CAMPILLO & SEQUERA, 2018).

Con la función `plt.xscale(log)` se logró obtener la escala logarítmica en el eje x. `plt.title`, `plt.xlabel` y `plt.ylabel` permiten identificar las curvas y los ejes al observarse

en el diagrama. Por otro lado el plt.ylim limita el diagrama. Después de establecidos esos parámetros se obtuvieron las siguientes gráficas para cada uno de los aceros de estudio, con sus respectivas comparaciones con las curvas establecidas en la USS.

Observando el diagrama generada del acero 4027 con el software Python y la establecida por la USS (Ver figura 19), se pudo distinguir una discrepancia en el área de transformación perlítica. A pesar de ello, el modelo matemático para esta curva presentó en mayor parte las cualidades presentes en la curva establecida por la USS. El punto crítico de la formación bainítica se encuentra al 0,7s en el modelo matemático mientras que en la USS se encuentra cerca de los 0,5s. Asimismo las temperaturas en las que inician todas las transformaciones se encuentran entre 20 y 50°C de diferencia con respecto a la curva USS.

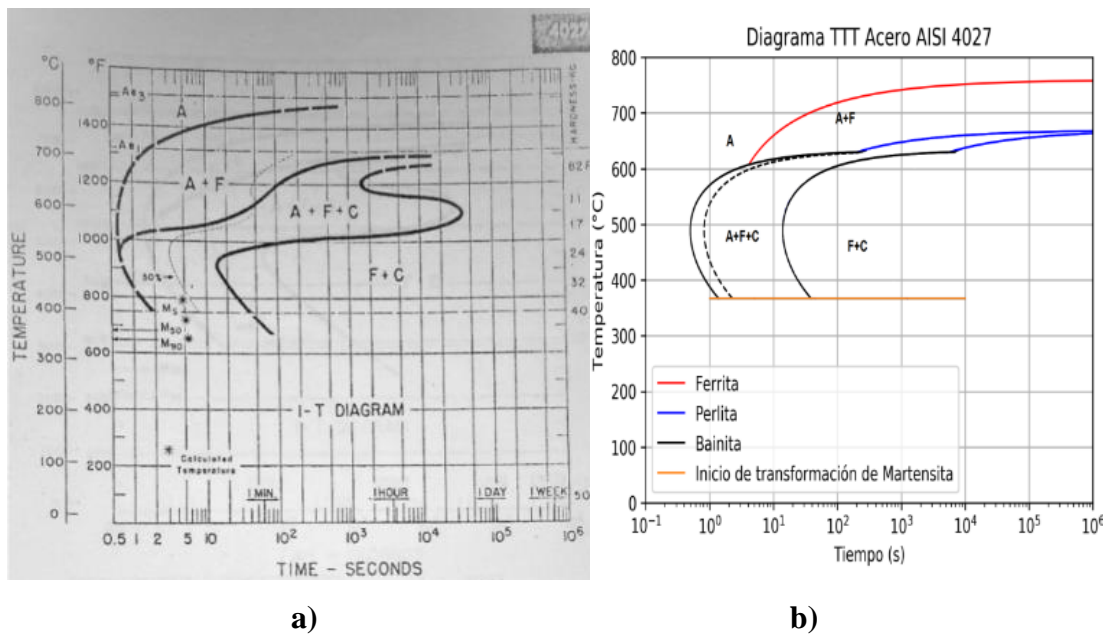


Figura 19. Comparación curva de transformación isotérmica del acero AISI 4027 de las USS con el software Python. a) USS. Atlas of Isothermal Transformation Diagrams United States Steel (1961). b) Software Python.

Fuente: (CAMPILLO & SEQUERA, 2018)

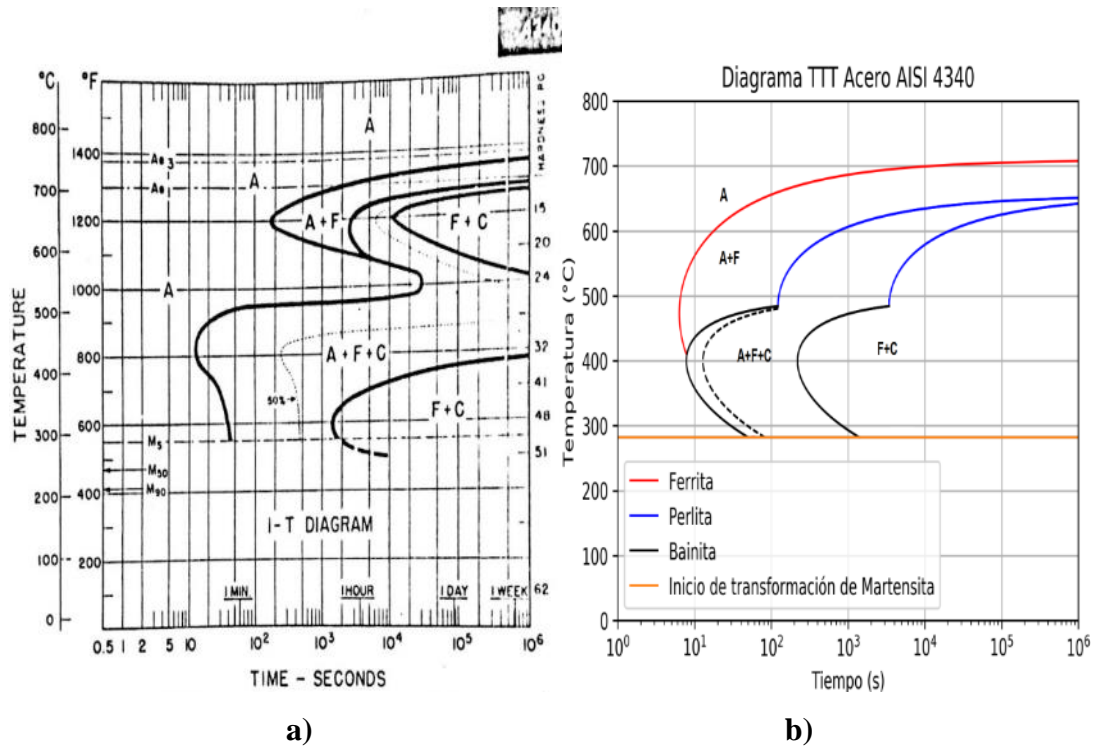


Figura 20. Comparación curva de transformación isotérmica del acero AISI 4340 de las USS con el software Python. a) USS. Atlas of Isothermal Transformation Diagrams United States Steel (1961). b) Software Python.

Fuente: (CAMPILLO & SEQUERA, 2018)

Observando el diagrama generada del acero 4340 con el software Python y la establecida por la USS (Ver figura 20), se nota diferencia considerable en la curva que representa el inicio de transformación de la austenita en ferrita, es notable con una diferencia de aproximadamente 200 °C, además se puede apreciar la gran diferencia en los tiempos de transformación, con una diferencia considerable de más de 990000 segundos a aproximadamente 500 °C.

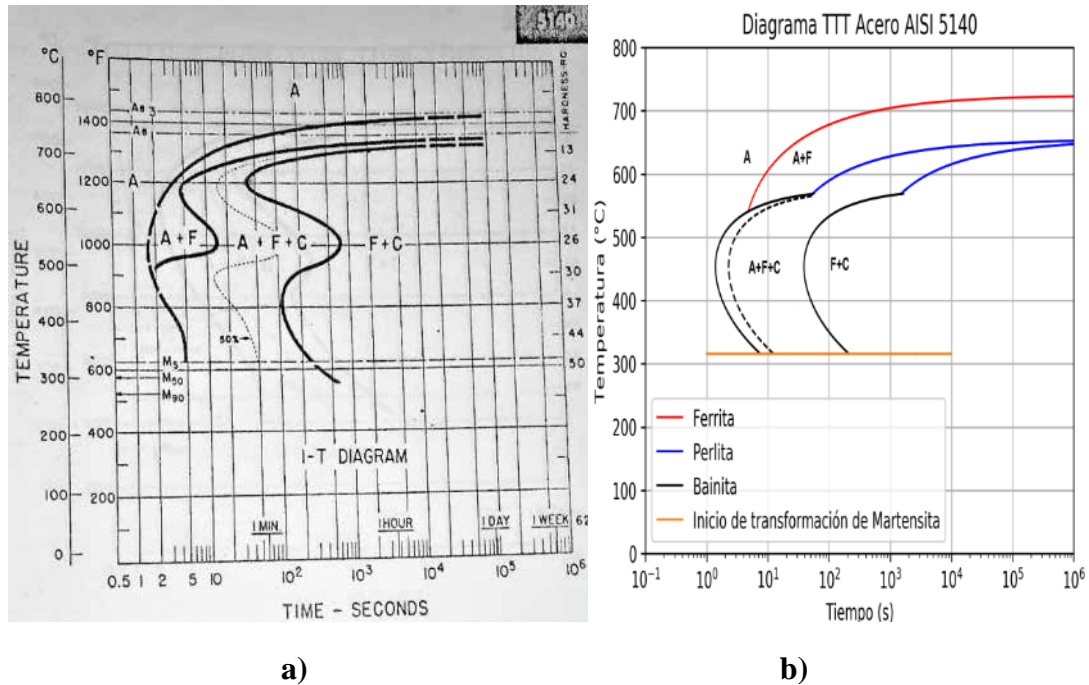


Figura 21. Comparación curva de transformación isotérmica del acero AISI 5140 de las USS con el software Python. a) USS. Atlas of Isothermal Transformation Diagrams United States Steel (1961). b) Software Python.

Fuente: (CAMPILLO & SEQUERA, 2018)

Observando el diagrama generada del acero 5140 con el software Python y la establecida por la USS (Ver figura 21), se puede apreciar cierta similitud en los tiempos de transformación, el punto crítico de la transformación de ferrita tiene una diferencia no mayor a 50°C mientras que los tiempos de transformación no se separan mucho de la realidad. La zona que encierra la A+F+C (austenita mas ferrita mas cementita) es bastante aceptable en comparación con la del USS.

Hay que resaltar que en los diagramas anteriores la cantidad de porcentaje de los aleantes en los aceros en estudio utilizados por la USS no se tiene información exacta del valor utilizado en sus ensayos a diferencia del código de programación utilizado el cual se tiene un porcentaje conocido, he aquí una de las razones por las cuales las curvas diferencian unas a otras (Ver figura 22) además de que se realizaron

hace más de medio siglo y bajo la metodología de promedios extraídos de sus resultados.

Otra diferencia que se puede notar en todas las curvas es el 50% de transformación en el área A+F+C (austenita mas ferrita mas cementita), esto puede abre la posibilidad a que el modelo matemático utilizado pueda tener oportunidades de mejora.

ACEROS ALEADOS



AISI Número	C	Mn	P Max	S Max	Si	Ni	Cr	Mo	SAE Número
4027	0.25-0.30	0.70-0.90	0.040	0.040	0.20-0.35	0.20-0.30	4027
4340	0.38-0.43	0.60-0.80	0.040	0.040	0.20-0.35	1.65-2.00	0.70-0.90	0.20-0.30	4340
5150	0.48-0.53	0.70-0.90	0.040	0.040	0.20-0.35	0.70-0.90	5150

Figura 22. Composición de los aceros estudiados.

Fuente: (Servicio Industrial, S.A. de C.V, 2010).

4.3 Aplicación del modelo matemático y obtener los diagramas CCT de los aceros seleccionados con el software Python.

Este proceso consistió en utilizar el software Python al igual que en la obtención de los diagramas TTT pero, con la diferencia que el modelo matemático a manejar aquí es el de JMAK, el cual permitió tener como resultado la obtención de las curvas CCT. Para lograrlo se inició estableciendo las constantes necesarias, las cuales se muestran (Ver tabla 2).

Tabla 2. Propiedades de los aceros en estudio para los diagrama CCT.

Propiedades	Acero 4027	Acero 4340	Acero 5140
T₀ (°C)	850	980	850
N (tamaño del grano)	7	7,5	5.5
C	0,26	0,42	0,41
Mn	0,87	0,78	0,75
Cr	0	0,8	1,05
Si	0,15	0	0,1
Mo	0,26	0,33	0
Ni	0	1,79	0

Fuente: (CAMPILLO & SEQUERA, 2018).

Luego de definir las constantes se realizaron todos los cálculos bases correspondientes tales como la temperatura de inicialización de la martensita, temperatura de estabilización de la ferrita, temperatura de estabilización de la perlita y la temperatura de estabilización de la bainita, todos estos cálculos (Ver figura 23) base fueron pasados a números enteros para su mejor manejo dentro del código debido a los múltiples bucles y condicionales, además del ahorro de memoria registrada por Python el cual fue considerable por la alta cantidad de datos registrados.

```
# -*- coding: utf-8 -*-
"""
Created on Wed Apr  4 22:09:56 2018

@author: Familia
"""

#Librerias que se van a utilizar
import matplotlib.pyplot as plt
from sympy import Symbol
from sympy import integrate
from scipy.integrate import quad
import math

#Se declaran las variables
To=850
R=8.31447 #Constante internacional de los gases
N=7
C=0.26
Mn=0.87
Cr=0
Si=0.15
Mo=0.26
Ni=0
Tp=670 #Temperatura de estabilización de Perlita
Tf=760 #Temperatura de estabilización de Ferrita
Tb=844-597*C-63*Mn-16*Ni-78*Cr #Temperatura de estabilización de Bainita
Ms=521-353*C-24.3*Mn-225*Si-27.4*Ni-17.7*Cr-25.8*Mo #Temperatura de inicio de transformación de Martensita
Ms = int(Ms)
Ti=Ms #Temperaturas de estudio
Tb = int(Tb)
```

Figura 23. Declaración de las variables en el programa para el acero AISI 4027.

Fuente: (CAMPILLO & SEQUERA, 2018).

Con los datos ya registrados en el sistema se realizó nuevamente el cálculo necesario para las curvas TTT debido a que los tiempos de transformación son datos claves para el desarrollo de las ecuaciones de la JMAK (Figura 24) con la cual se consiguieron los siguientes resultados (Figura 25, 26, 27).

```

Xb=10
while i<=700:
  if Ti<Ms:
    Fm=1-math.e**(-0.011*(Ms-Ti))
  if Xb<=1:
    FunXb=Dt/tab2;
    Xb=Xb+FunXb;
    Xib=Xb;
  else:
    if Xb<100 and Xb>=1:
      tb=Dt(i-1)
      Tbt(j)=T(i-1)
      Xb=10000000000000000;
      h=i-1;
      Fib=0;
      Titajb=Dt(i);
      nb=0;
      bb=0;
      while h<=length(tab2):
        if Fib<1:
          nb(h)=(log((log(1-Vo1i))./(log(1-Vo1f))))./(log(TauB01(h)./TauB(h)));
          bb(h)=-((log(1-Vo1i))./(TauB01(h).^nb(h)));
          Fib=1-exp(-bb(h).*(Titajb+0.5).^nb(h));
          Titajb=(-log(1-Fib))/bb(h).^(1./nb(h));
    if Ti<Tp:
      if Xp<=1:
        FunXp=Dt/tap2
        Xp=Xp+FunXp;
        Xip=Xp;
      else:
        if Xp<10 && Xp>=1:
          tp(j)=Dt(i-1);
          Tpt(j)=T(i-1);
          Xp=1000000;
          h=i-1;
          Fip=0;
          Titajp=Dt(i);
          np=0;%Parametro n para la Perlita
          bp=0;%Parametro b para la Perlita
          Fipx=0;
        while h<=length(tap2):
          if Fip<1:
            np(h)=(log((log(1-vo1i))./(log(1-vo1f))))./(log(TauP01(h)./TauP(h)));
            bp(h)=-((log(1-Vo1i))./(TauP01(h).^np(h)));
            Fip=1-exp(-bp(h).*(Titajp+0.5).^np(h));
            Titajp=(-log(1-Fip))/bp(h).^(1./np(h));

```

Figura 24. Formulación de las gráficas en el programa para el acero AISI 4027.

Fuente: (CAMPILLO & SEQUERA, 2018).

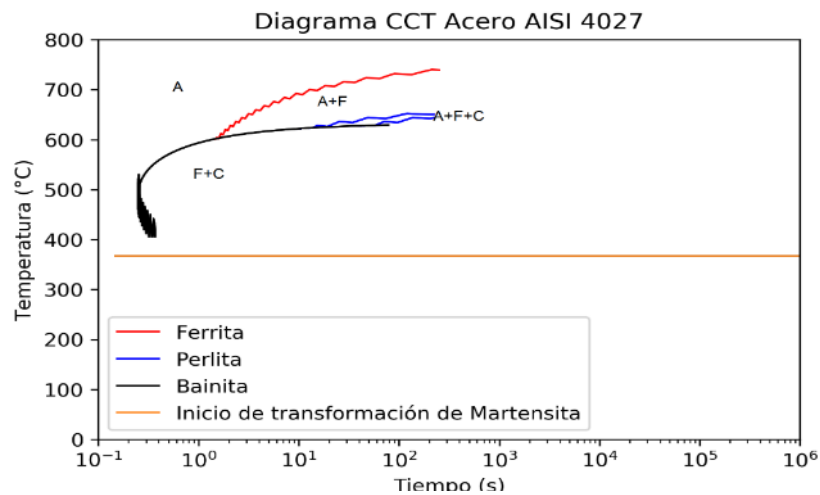


Figura 25. Diagrama CCT del acero AISI 4027 según el modelo matemático.

Fuente: (CAMPILLO & SEQUERA, 2018)

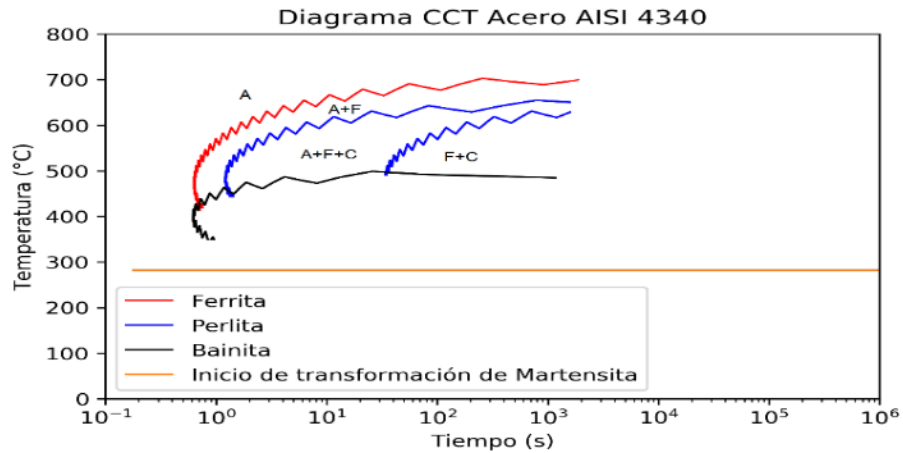


Figura 26. Diagrama CCT del acero AISI 4340 según el modelo matemático.

Fuente: (CAMPILLO & SEQUERA, 2018)

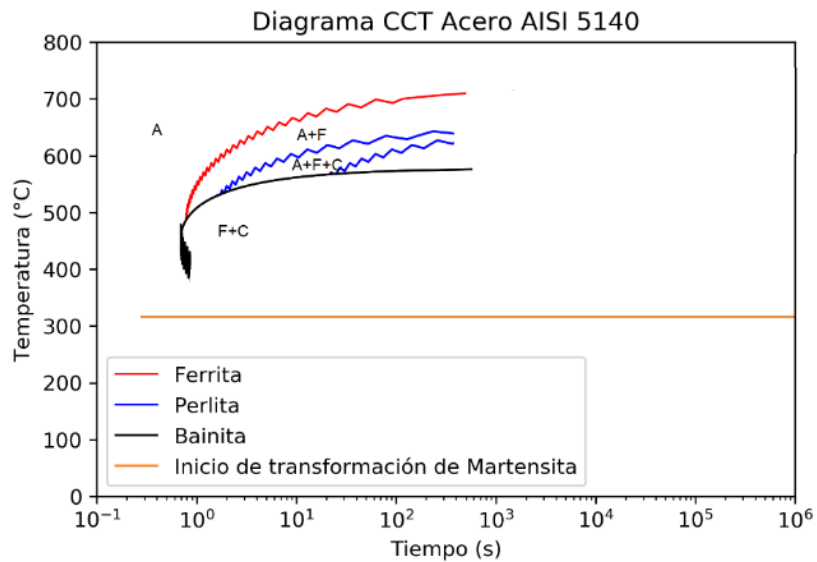


Figura 27. Diagrama CCT del acero AISI 5140 según el modelo matemático.

Fuente: (CAMPILLO & SEQUERA, 2018)

4.4 Comparar los resultados obtenidos por Python con Matlab.



Este apartado presentó el cumplimiento del último objetivo específico y por tanto el general, donde se procede a identificar las diferencias y semejanzas de las dos software con el fin de, por medio de ellas obtener el diagrama más parecido a la

realidad. Para ello se validarán los resultados obtenidos y finalmente se compararán entre ellos.

4.4.1 Requerimiento Computacional.

Una de las variables que se tomó en cuenta al realizar la comparación de los programas fueron los parámetros que especifica la casa matriz como indispensables a poseer por el equipo computacional o hardware en el que pretenda desarrollar el usuario las simulaciones de un sistema, estos requerimientos computacionales se listan en la siguiente tabla.

Tabla 3. Requerimientos Computacionales de cada programa.

Requerimiento	Matlab 	Python (Para el uso de la investigación) 
Sistema operativo	Windows, Linux, Mac-OS.	Windows, Linux, Mac-OS
Procesador	Mínimo Intel o AMD x86-64. Recomendado Intel o AMD x86-64 con 4 núcleos. AVX	Intel Pentium Dual Core 2.60 GHz.
Memoria RAM	Mínimo 4GB. Recomendado 8GB.	2 GB.
Disco Duro	Mínimo 4-6 GB HDD. Recomendado 22 GB SSD.	1 GB.
Tarjeta gráfica	Open GL con 1GB mínimo.	256 MB.
Acceso	Limitado.	Libre.
Adquisición	5000-6000 \$.	Gratis.
Obtención	Descarga desde el sitio Web oficial. Mediante representantes comerciales.	Descarga desde el sitio Web oficial. Mediante IDE.
Software educacional	Versión educacional disponible por 50\$ anuales.	No aplica.

Fuente: (CAMPILLO & SEQUERA, 2018).

4.4.2 Comparación entre el Diagrama TTT generado por Matlab y Python.

A continuación se muestran los diagramas TTT generados por ambos software.

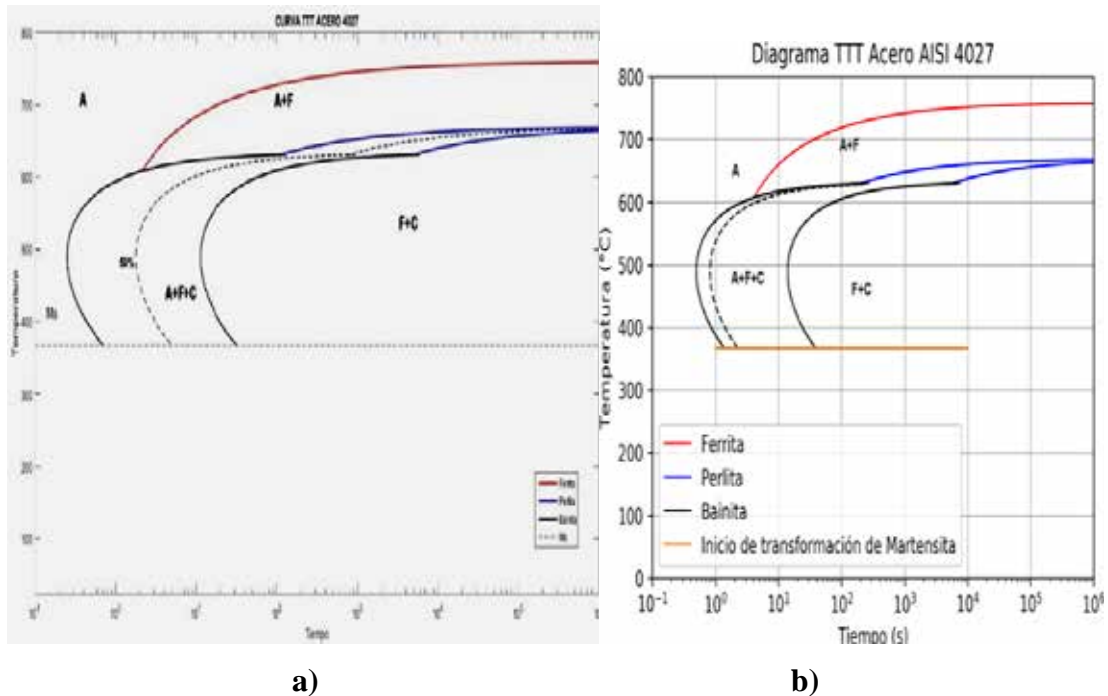


Figura 28. Comparación curva de transformación isotérmica del acero AISI 4027 de las del software Matlab entre el software Python. a) Software Matlab. (Salas, 2017). b) Software Python.

Fuente: (CAMPILLO & SEQUERA, 2018)

Para el acero AISI 4027 (Ver figura 28) no hay mucha diferencia salvo la del 50% de transformación, mostrando así la similitud en cuanto este resultado entre las dos herramientas computacionales.

Por parte de python se pudo seleccionar la anchura de la gráfica, tales como el tamaño de la imagen en pixeles, grueso y tipo de las líneas cosas que en el código de Matlab no se apreció.

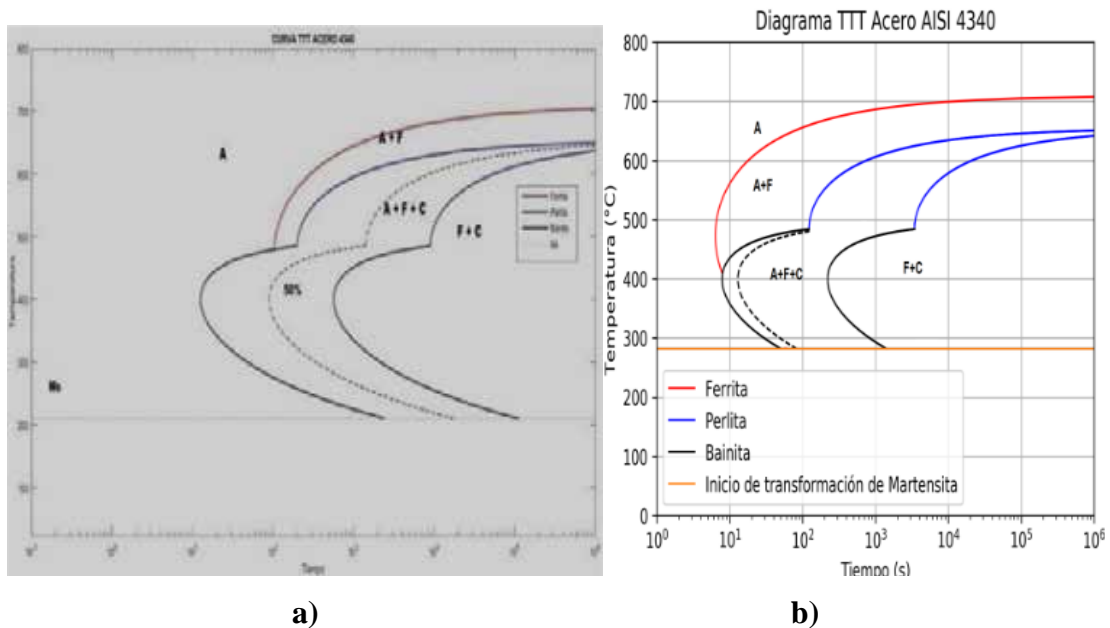


Figura 29. Comparación curva de transformación isotérmica del acero AISI 4340 de las del software Matlab entre el software Python. a) Software Matlab. (Salas, 2017). b) Software Python.

Fuente: (CAMPILLO & SEQUERA, 2018).

En el acero AISI 4340 (ver figura 29) se puede apreciar con más detalle que la línea del 50% de transformación A+F+C (austenita más ferrita más cementita) aparece a partir de la temperatura de estabilización de la bainita además de estar mucho más cerca de la línea donde inicia la transformación.

Se aprecia que la zona que representa la transformación “A+F” (austenita más ferrita) es mucho más amplia en Python, pudiendo dar resultados muy diferentes en caso de realizar un ensayo experimental confiando en la predicción de una de las dos herramientas y obteniendo resultados de la otra.

El resto de la gráfica goza de la misma similitud que la gráfica anterior (ver figura 28), dando resultados muy similares.

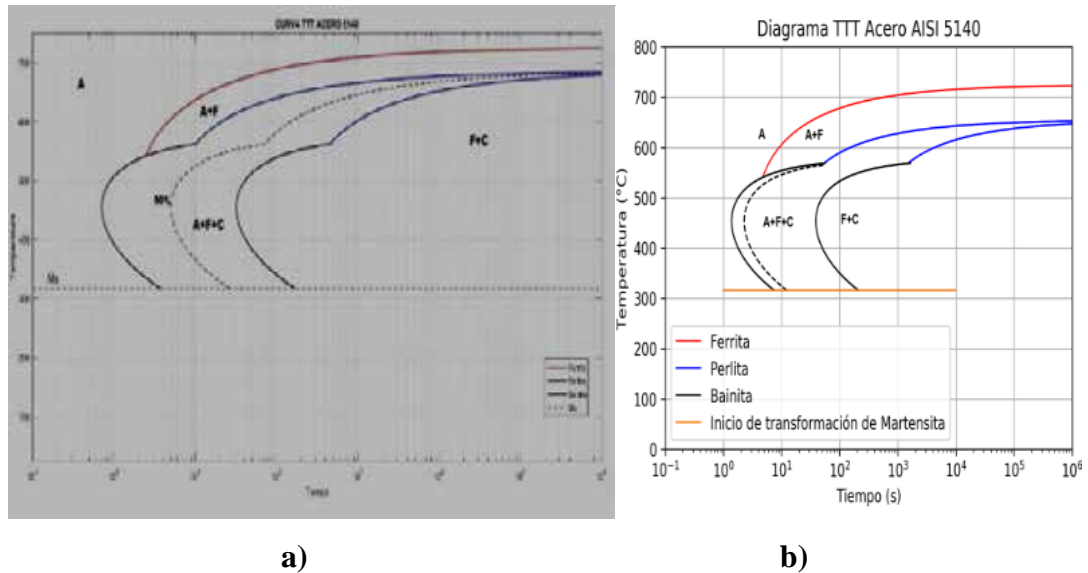


Figura 30. Comparación curva de transformación isotérmica del acero AISI 5140 de las del software Matlab entre el software Python. a) Software Matlab. (Salas, 2017). b) Software Python.

Fuente: (CAMPILLO & SEQUERA, 2018).

Para el caso de acero AISI 5140 (ver figura 30) ocurre el mismo caso con respecto a la zona de transformación “A+F”(austenita mas ferrita) y la línea del 50% que las gráficas anteriores (ver figura 28, 29).

Se puede anexar diferencias en los tiempos de transformación entre las dos graficas no solo en el área de la transformación “A+F”(austenita mas ferrita) también en la zona “A+F+C” (austenita mas ferrita mas cementita) donde en Matlab esta zona se consigue con menor tiempo.

4.4.3 Comparación entre el Diagrama CCT generado por Matlab y Python.

Posteriormente de haber realizado el análisis comparativo del diagrama TTT se procedió a realizar el del diagrama CCT.

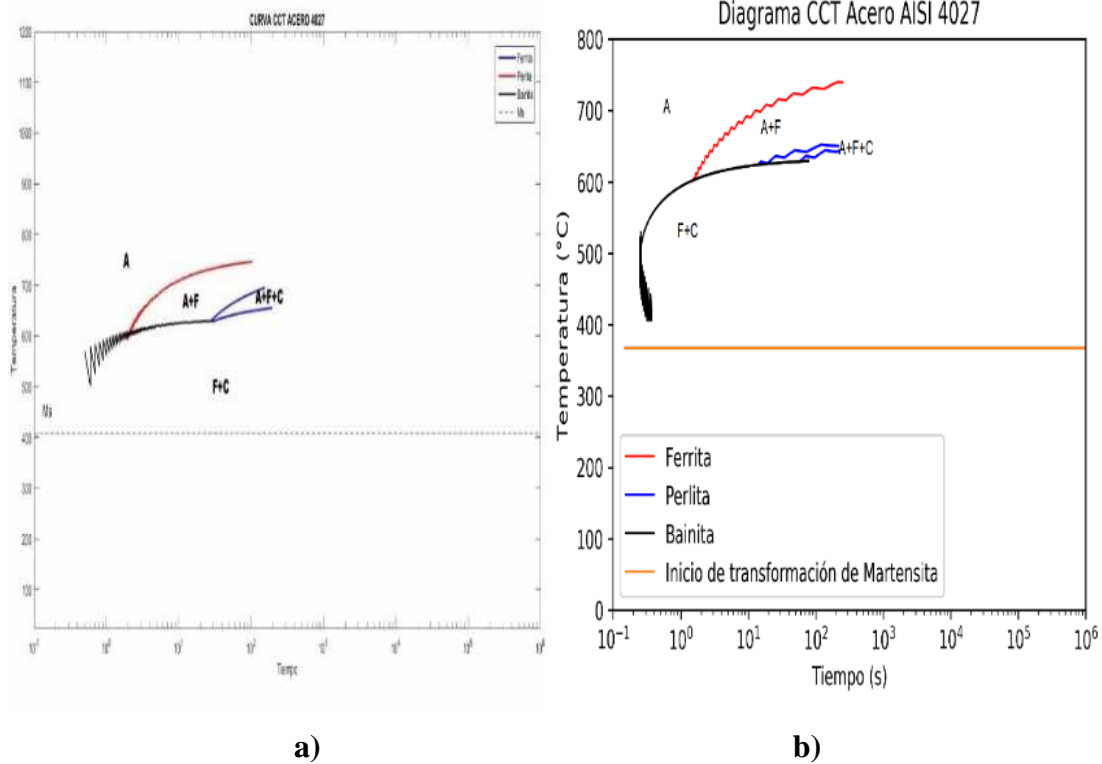


Figura 31. Comparación diagrama CCT del acero AISI 4027 entre las de Matlab y la de Python. a) Matlab. (Salas, 2017). b) Python.

Fuente: (CAMPILLO & SEQUERA, 2018).

En el acero AISI 4027 (Ver figura 31) se puede apreciar una gran diferencia con respecto a la anchura del diagrama, se puede visualizar detalladamente el comportamiento de los diagramas hechos en Python que con la ejecutada por Matlab.

Se puede apreciar que el inicio de transformación es casi inmediato en el diagrama hecho en Python, cerca de 0,3s y en el resultado obtenido por Matlab se observa aproximadamente a 1s. Además se observa en la zona "A+F+C" (austenita mas ferrita mas cementita) de Python tiene una amplitud menor que Matlab.

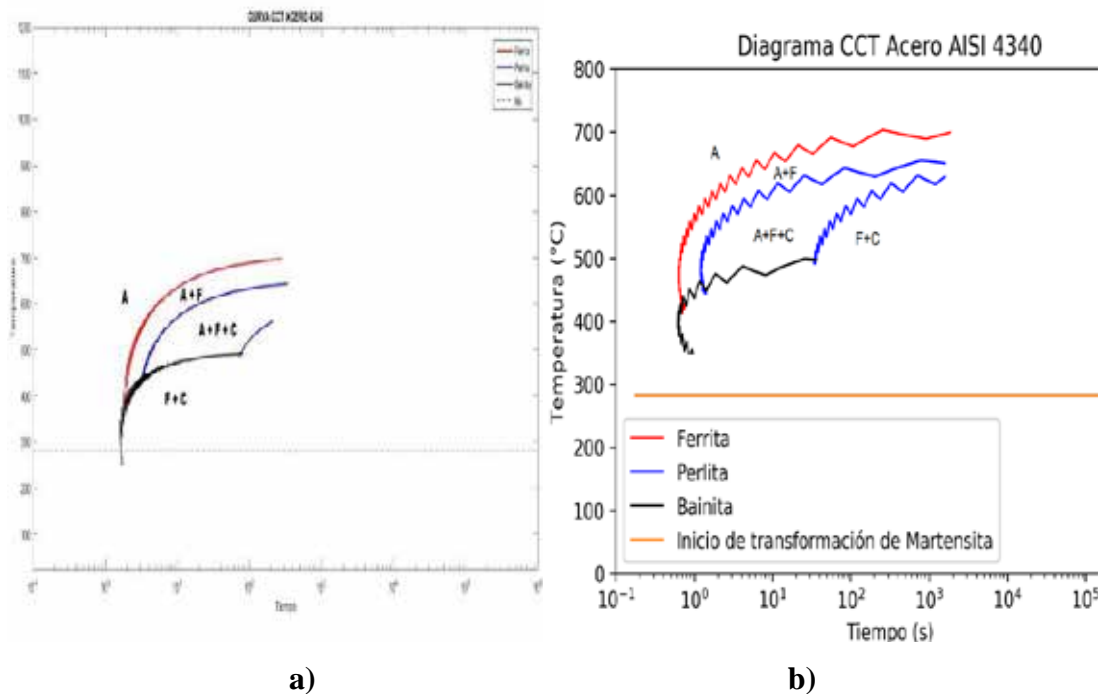


Figura 32. Comparación diagrama CCT del acero AISI 4340 entre las de Matlab y la de Python. a) Matlab.(Salas, 2017). b) Python.

Fuente: (CAMPILLO & SEQUERA, 2018).

En el caso del acero AISI 4340 (ver figura 32) se puede observar mucha similitud entre los dos diagramas salvo por la forma de las curvas, en Python se detalla variaciones (mostrando picos) y en Matlab se aprecian curvas finas sin desviaciones.

Se puede apreciar como en la figura anterior (ver figura 31) se percibe una gran diferencia con respecto a la anchura del diagrama, puesto que se puede visualizar de una forma más detallada lo que ocurre con la gráfica realizada en Python comparado con la elaborada por Matlab.

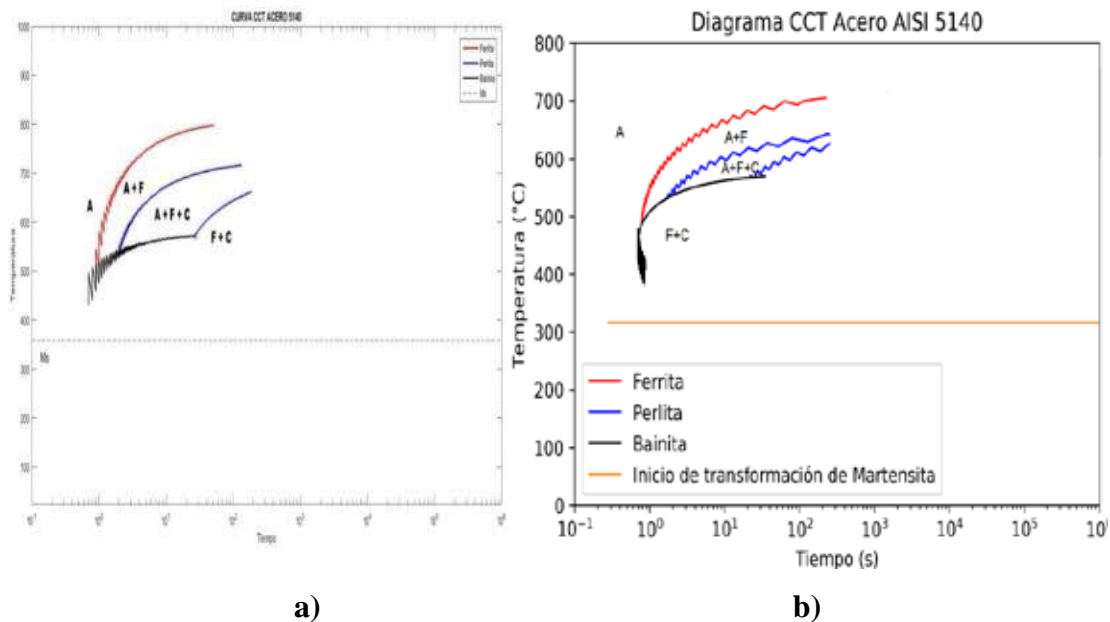


Figura 33. Comparación diagrama CCT del acero AISI 5140 entre las de Matlab y la de Python. a) Matlab.(Salas, 2017). b)Python.

Fuente: (CAMPILLO & SEQUERA, 2018).

Con respecto al acero AISI 5140 (ver figura 33) se puede ver que el inicio de transformación es muy similar en ambos diagramas, aproximadamente a 1s. Además se observa en la zona “A+F+C” (austenita mas ferrita mas cementita) de Matlab para un tiempo de 100 segundos cuenta con un intervalo entre 645 a 710°C de amplitud mientras que en Python se encuentra comprendida entre 605 a 630°C. Con esto se concluye que la amplitud en los diagramas de Matlab es mayor.

De manera general al igual que en caso de las TTT las CCT presentan la misma diferencia en cuanto a la visibilidad de los diagramas de Matlab, siendo Python más agradable para el lector.

En cuanto a las curvas todos los diagramas en Python presentaron fluctuaciones (picos), siendo en Matlab estas curvas más finas y sin variaciones. Se debe destacar que Python está más adaptado a la realidad debido a que en los ensayos se obtienen resultados que se promedian, con el objetivo de que se entienda mejor los diagramas.

Tabla 4. Ventajas y desventajas de utilizar Python como Matlab.

Ventajas	Desventajas
+Herramienta gratis.	-Tutoría escasa en Venezuela
+Rápida adaptación.	-Soporte técnico nulo.
+Múltiples aplicaciones	-Baja información en español.
+Variedades de IDE (Entorno de desarrollo integrado)	-Puede generar fallas en el software por mal uso
+Grandes cantidades de librerías	-No posee la reputación de Matlab en el ámbito de la ciencia.
+No necesita alto recurso computacional	-Requirió más líneas de programación para el mismo fin.
+Rápida y fácil adquisición	

Fuente: (CAMPILLO & SEQUERA, 2018).

CONCLUSIONES

Hay que resaltar que la cantidad de porcentaje de los aleantes en los aceros en estudio utilizados por la USS no se tiene información exacta del valor utilizado en sus ensayos a diferencia del código de programación utilizado el cual se tiene un porcentaje conocido, he aquí una de las razones por las cuales las curvas diferencian unas a otras.

Python se adaptará a la realidad debido a que en los ensayos realizados por la USS obtuvieron resultados que son promediados, con el objetivo de que se entienda mejor los diagramas.

Se perciben diferencias entre Python y la USS que se notan en todas las curvas del 50% de transformación en el área A+F+C (austenita mas ferrita mas cementita), esto puede abrir la posibilidad a que el modelo matemático utilizado pueda tener oportunidades de mejora.

En 2 de 3 diagramas TTT se apreció que la zona que representa la transformación “A+F” (austenita mas ferrita) es mucho más amplia en Python, pudiendo dar resultados muy diferentes en caso de realizar un ensayo experimental confiando en la predicción de una de las dos herramientas y obteniendo resultados de la otra.

Por parte de python se puede seleccionar las especificaciones de la gráfica, tales como el tamaño de la imagen en pixeles, grueso y tipo de las líneas que en el código de Matlab no se apreció, esto hizo que la lectura de los diagramas fueran más agradables para el lector.

Existen diferencias y similitudes en los tiempos de transformación entre algunos diagramas de Python y Matlab.

Todos los diagramas CCT en Python presentaron fluctuaciones (picos), siendo en Matlab curvas más finas y sin variaciones.

Nuevamente Python se destaca por estar más amoldado a la realidad en comparación con Matlab.

Las herramientas computacionales comparadas no son polos opuestos, más allá de utilizar lenguajes de programación diferentes son parecidos entre sí en el ámbito de la ciencia e ingeniería, además se pueden entrelazar de ser necesario mediante funciones de llamadas, pudiendo utilizar las dos herramientas para un mismo fin de acuerdo a la necesidad y/o habilidad del programador.

Hay múltiples entornos de desarrollo interactivo IDE que fueron desarrollados con la idea de convertir Python y Matlab, esto es muy conveniente porque sin importar que dos programadores utilizaron lenguajes diferentes la comunicación entre ambos a la hora de debatir sobre algún proyecto no es un inconveniente para entender de lo que se está discutiendo en el debate.

Se concluye que los objetivos establecidos fueron cumplidos satisfactoriamente, así como la pregunta planteada en la formulación del problema fue respondida en los resultados del capítulo IV.

RECOMENDACIONES

Utilizar otros modelos matemáticos para el estudio y observar sus comportamientos, de tal manera que se puedan descartar aquellos que presenten desviaciones significativas a los mostrados en esta investigación.

Buscar diagramas TTT de ensayos experimentales actualizados por instituciones acreditadas.

La utilización de Python en la Universidad “José Antonio Páez” sería una buena alternativa para suplir a los programas que requieren licencias costosas.

Python funciona para muchas aplicaciones que fácilmente se adaptarían a cursos impartidos en ingeniería.

BIBLIOGRAFIA

- Alvarez, F. B. (2017). **Tratamientos termicos de los aceros.**Oviedo.
- Arias, F. (2006). **El Proyecto de Investigación.**
- ASM International. (1991). **ASM Handbook.**
- Atlas of Isothermal Transformation Diagrams United States Steel, A. (1961). **Atlas of Isothermal Transformatio.**
- Baray, H. L. (2006). **Introduccion a la metodologia de la investigacion.**Chihuahua.
- Callister, W. (2002). **Introduccion a la ciencia e ingenieria de los materiales.**Mexico: Reverte S.A.
- Duran, G. (2006). **Investigacion de operaciones. modelos.**Santiago.
- Escuela Colombiana de Ingenieria. (2008). **Tratamientos termicos protocolo, curso de materiales.**Bogota.
- Felipe, A. D. (2012). **Aceros, estructuras y tratamientos termicos.** Cuatitlan.
- Fundinox Chile. (2010). **Fundinox.** Recuperado el 01 de 10 de 2017, de http://www.fundinox.cl/ASTM_A958.pdf
- López, N. M. (2016). **Diseño de la investigacion.** Alicante.
- Maisel, H., & Gnugnoli, G. (1972). **Simulation of discrete stochastic systems.**Science Research Associates, Inc.
- Mandujano, A. (2009). **Propuesta de fabricacion de piezas mecanicas de un acero inoxidable aisi 416 cementadas para ser sometidas a una prueba de degaste abrasivo.** Mexico DF.
- Naylor, T. H. (1982). **Técnicas de simulación en computadoras.** Limusa.
- Orozco, A. (2015). **Clasificacion AISE-SAE de los aceros.**San Marcos.
- Rossum, G. v. (2009). **El tutorial de Python.** Haarlem: Python Software Foundation.

Salas, M. (2017). **Simulacion de tratamiento termico mediante temple en agua en aceros de baja aleación AISI 4xxx y 5xxx empleando el metodo de elementos finitos.**Carabobo.

Servicio Industrial, (2010) **Tabla aceros SAE – AISI.** México

Schmid Kalpakjian. (2008). **Manufactura, ingenieria y tecnologia.** Mexico DF: Pearson.

Shannon, R. E. (1988). **Simulacion de sistemas.**

Sierra Bravo, I. (1991). **Metodologia de la investigacion.**

Stracuzzi, S. P. (2003). **Metodología de la investigación cuantitativa.**

Tamayo, C. (2008). **Tecnicas e instrumentos para la recoleccion de datos.**Lima.

Trujillo, A. (2017). **Obtencion de las curvas TTT en aceros hipereutectoides mediante un proceso de simulacion por matlab** Carabobo.

Universidad de Jaen. (2015). **Diseño documental** Andalucia.

Universidad tecnologica de Pereira. (2013). **Diagrama hierro-carbono.**

Zetino, J. A. (2014). **Simulacion de los esfuerzos residuales resultantes de un tratamiento termico de temple en piezas cilindricas de acero.** San Salvador.

ANEXOS

ANEXO A: Codificación de los programas para elaborar los diagramas TTT.

```
#Librerías que se van a utilizar
import matplotlib.pyplot as plt
from sympy import Symbol
from sympy import integrate
from scipy.integrate import quad
import math

#Se declaran las variables
R=8.31447 #Constante internacional de los gases
N=7
C=0.26
Mn=0.87
Cr=0
Si=0.15
Mo=0.26
Ni=0
Tp=670 #Temperatura de estabilización de Perlita
Tf=760 #Temperatura de estabilización de Ferrita
Tb=844-597*C-63*Mn-16*Ni-78*Cr #Temperatura de estabilización de bainita
Ms=521-353*C-24.3*Mn-225*Si-27.4*Ni-17.7*Cr-25.8*Mo #Temperatura de inicio
de transformación de Martensita
Ms = int(Ms)
Ti=Ms #Temperaturas de estudio
```

```
Tb = int(Tb)
```

```
#Se resuelve la integral de Kirkaldy
```

```
x=Symbol('x')
```

```
integrate(1/(x**(2*(1-x)/3)*(1-x)**(2*x/3)),x)
```

```
f=lambda x:1/(x**(2*(1-x)/3)*(1-x)**(2*x/3))
```

```
I=quad(f,0,0.99)
```

```
I2=quad(f,0.,0.1)
```

```
I3=quad(f,0.,0.5)
```

```
#Se definen variables como listas
```

```
#Transformacion de austenita a Ferrita
```

```
taf=list()
```

```
taf2=list()
```

```
#Transformacion de austenita a Perlita
```

```
tap=list()
```

```
tap2=list()
```

```
tap3=list()
```

```
#Transformacion de austenita a Bainita
```

```
tab=list()
```

```
tab2=list()
```

```
tab3=list()
```

```
#Uso practico
```

```
Ti2=list()
```

```
p=list()
```

```
while Ti<860:
```

```

#Ecuacion para transformacion de austenita a Ferrita
Df=math.e**(-23500/(R*Ti))
taf.append(((60*Mn+2*Ni+68*Cr+244*Mo)/(6**2**(N/8)*(Tf-
(Ti))**3*Df)*I*10**5)
taf2.append(((60*Mn+2*Ni+68*Cr+244*Mo)/(6**2**(N/8)*(Tf-
(Ti))**3*Df)*I2*10**4)

#Ecuacion para la transformacion de Austenita a Perlita
sp=1/(math.e**(-27500/(R*Ti)))+(0.5*Mo)/(math.e**(-37000/(R*Ti)))
Dp=sp**-1
tap.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-Ti)**3*Dp)*I*10**6)
tap2.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti)**3*Dp)*I2*10**5)
tap3.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti)**3*Dp)*I3*10**5)

#Ecuacion para la transformacion de Austenita a Bainita
if Ti < Tb:
    tab.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti)**2*math.e**(-27500/(R*Ti)))*I*10**6)
    tab2.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti)**2*math.e**(-27500/(R*Ti)))*I2*10**5)
    tab3.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti)**2*math.e**(-27500/(R*Ti)))*I3*10**5)
p.append(Ti)
Ti2.append(Ti)
Ti=Ti+2

```

```

#Procedimiento para graficar linea Ms
x=1
y=0
n=1
q=list()
z=list()

while y<10**4:
    x=n+1
    n=n+1
    y=y+1
    q.append(x)
z.append(Ms)

#Grafica
plt.xscale('log')
plt.ylim(0, 1000)
plt.title('(Diagrama TTT Acero AISI 4027)')
plt.plot(taf2, Ti2, 'y-', label = "Ferrita") # Dibuja el gráfico
plt.xlabel("Tiempo (s)") # Inserta el título del eje X
plt.ylabel("Temperatura (°C)") # Inserta el título del eje Y
plt.ioff() # Desactiva modo interactivo de dibujo
plt.plot(tap, Ti2, 'b-', label = "Perlita") # No dibuja datos de lista2
plt.ion() # Activa modo interactivo de dibujo
plt.plot(tap, Ti2, 'b-') # Dibuja datos de lista2 sin borrar datos de lista1
plt.ioff()
plt.plot(tap2, Ti2, 'b-')
plt.ion()
plt.plot(tap2, Ti2, 'b-')

```

```

plt.ioff()
plt.plot(tab, p, 'k-')
plt.ion()
plt.plot(tab, p, 'k-', label = "Bainita")
plt.ioff()
plt.plot(tab2, p, 'k-')
plt.ion()
plt.plot(tab2, p, 'k-')
plt.ioff()
plt.plot(tab3, p, 'k--')
plt.ion()
plt.plot(tab3, p, 'k--')
plt.ioff()
plt.plot(q, z)
plt.ion()
plt.plot(q, z)
plt.legend()

```

Figura A-1. Código de Python completo para obtención del diagrama TTT del Acero 4027.

Fuente: (CAMPILLO & SEQUERA, 2018)

```

#Librerías que se van a utilizar
import matplotlib.pyplot as plt
from sympy import Symbol
from sympy import integrate
from scipy.integrate import quad
import math

```

#Se declaran las variables

R=8.31447 #Constante internacional de los gases

N=7.5

C=0.42

Mn=0.78

Cr=0.8

Si=0

Mo=0.33

Ni=1.79

Tp=655 #Temperatura de estabilización de Perlita

Tf=710 #Temperatura de estabilización de Ferrita

Tb=830-270*C-90*Mn-37*Ni-70*Cr-83*Mo #Temperatura de estabilización

de Bainita

Ms=521-353*C-24.3*Mn-225*Si-27.4*Ni-17.7*Cr-25.8*Mo

Ms = int(Ms) #Temperatura de inicio de transformación de Martensita

Ti=Ms

Tb = int(Tb)

#Se resuelve la integral de Kirkaldy

x=Symbol('x')

integrate(1/(x**(2*(1-x)/3)*(1-x)**(2*x/3)),x)

f=lambda x:1/(x**(2*(1-x)/3)*(1-x)**(2*x/3))

I=quad(f,0,0.99)

I2=quad(f,0,0.1)

I3=quad(f,0,0.5)

#Se definen variables como listas

#Transformacion de austenita a Ferrita

```

taf=list()
taf2=list()

#Transformacion de austenita a Perlita
tap=list()
tap2=list()
tap3=list()

#Transformacion de austenita a Bainita
tab=list()
tab2=list()
tab3=list()

#Uso practico
Ti2=list()
p=list()

while Ti<860:

    #Ecuación para transformación de austenita a Ferrita
    Df=math.e**(-23500/(R*Ti))
    taf.append(((60*Mn+2*Ni+68*Cr+244*Mo)/(6*2**(N/8)*(Tf-
(Ti))**3*Df)*I*10**5)
    taf2.append(((60*Mn+2*Ni+68*Cr+244*Mo)/(6*2**(N/8)*(Tf-
(Ti))**3*Df)*I2*10**4)

    #Ecuación para la transformación de Austenita a Perlita
    sp=1/(math.e**(-27500/(R*Ti)))+(0.5*Mo)/(math.e**(-37000/(R*Ti)))
    Dp=sp**-1

```

```

        tap.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti)**3*Dp)*I*10**6)
        tap2.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti)**3*Dp)*I2*10**5)
        tap3.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti)**3*Dp)*I3*10**5)

    if Ti < Tb:
        tab.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti)**2*math.e**(-27500/(R*Ti)))*I*10**6)
        tab2.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti)**2*math.e**(-27500/(R*Ti)))*I2*10**5)
        tab3.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti)**2*math.e**(-27500/(R*Ti)))*I3*10**5)

    p.append(Ti)
    Ti2.append(Ti)
    Ti=Ti+3

#Procedimiento para graficar línea Ms
x=1
y=0
n=1
q=list()
z=list()

while y<10**6:
    x=n+1
    n=n+1
    y=y+1

```

```

    q.append(x)
z.append(Ms)

#Grafica
plt.xscale('log')
plt.ylim(0, 1000)
plt.grid(True)
plt.title('(TTT)')
plt.plot(taf2, Ti2, 'y-', label = "Ferrita") # Dibuja el gráfico
plt.xlabel("tiempo") # Inserta el título del eje X
plt.ylabel("temperatura") # Inserta el título del eje Y
plt.ioff() # Desactiva modo interactivo de dibujo
plt.plot(tap, Ti2, 'b-', label = "Perlita") # No dibuja datos de lista2
plt.ion() # Activa modo interactivo de dibujo
plt.plot(tap, Ti2, 'b-') # Dibuja datos de lista2 sin borrar datos de lista1
plt.ioff()
plt.plot(tap2, Ti2, 'b-')
plt.ion()
plt.plot(tap2, Ti2, 'b-')
plt.ioff()
plt.plot(tab, p, 'k-')
plt.ion()
plt.plot(tab, p, 'k-', label = "Bainita")
plt.ioff()
plt.plot(tab2, p, 'k-')
plt.ion()
plt.plot(tab2, p, 'k-')
plt.ioff()
plt.plot(tab3, p, 'k--')

```

```
plt.ion()
plt.plot(tab3, p, 'k--')
plt.ioff()
plt.plot(q, z)
plt.ion()
plt.plot(q, z)
plt.legend()
```

Figura A-2. Código de Python completo para obtención del diagrama TTT del Acero 4340.

Fuente: (CAMPILLO & SEQUERA, 2018)

```
import matplotlib.pyplot as plt
from sympy import Symbol
from sympy import integrate
from scipy.integrate import quad
import math

R=8.31447
N=5.5
C=0.41
Mn=0.75
Cr=1.05
Si=0.1
Mo=0
Ni=0

Tp=655
Tf=725
Tb=830-270*C-90*Mn-37*Ni-70*Cr-83*Mo
```

```
Ms=521-353*C-24.3*Mn-225*Si-27.4*Ni-17.7*Cr-25.8*Mo
```

```
Ms = int(Ms)
```

```
Ti=Ms
```

```
Tb = int(Tb)
```

```
x=Symbol('x')
```

```
integrate(1/(x**(2*(1-x)/3)*(1-x)**(2*x/3)),x)
```

```
f=lambda x:1/(x**(2*(1-x)/3)*(1-x)**(2*x/3))
```

```
I=quad(f,0,0.99)
```

```
I2=quad(f,0,0.01)
```

```
I3=quad(f,0,0.5)
```

```
taf=list()
```

```
taf2=list()
```

```
tap=list()
```

```
tap2=list()
```

```
tap3=list()
```

```
tab=list()
```

```
tab2=list()
```

```
tab3=list()
```

```
Ti2=list()
```

```
p=list()
```

```
while Ti<860:
```

```
    #Ecuacion para transformacion de austenita a Ferrita
```

```
    Df=math.e**(-23500/(R*Ti))
```

```

    taf.append(((60*Mn+2*Ni+68*Cr+244*Mo)/(6**2**(N/8)*(Tf-
(Ti))**3*Df)*I*10**5)

```

```

    taf2.append(((60*Mn+2*Ni+68*Cr+244*Mo)/(6**2**(N/8)*(Tf-
(Ti))**3*Df)*I2*10**4)

```

#Ecuacion para la transformacion de Austenita a Perlita

```

sp=1/(math.e**(-27500/(R*Ti)))+(0.5*Mo)/(math.e**(-37000/(R*Ti)))

```

```

Dp=sp**-1

```

```

    tap.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti))**3*Dp)*I*10**6)

```

```

    tap2.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti))**3*Dp)*I2*10**5)

```

```

    tap3.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti))**3*Dp)*I3*10**5)

```

```

if Ti < Tb:

```

```

    tab.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti))**2*math.e**(-27500/(R*Ti)))*I*10**6)

```

```

    tab2.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti))**2*math.e**(-27500/(R*Ti)))*I2*10**5)

```

```

    tab3.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti))**2*math.e**(-27500/(R*Ti)))*I3*10**5)

```

```

p.append(Ti)

```

```

Ti2.append(Ti)

```

```

Ti=Ti+2

```

```

x=1

```

```

y=0

```

```

n=1

```

```

q=list()
z=list()

while y<10**4:
    x=n+1
    n=n+1
    y=y+1
    q.append(x)
z.append(Ms)

plt.xscale('log')
plt.ylim(0, 1000)
plt.grid(True)
plt.title('(TTT)')
plt.plot(taf2, Ti2, 'y-', label = "Ferrita") # Dibuja el gráfico
plt.xlabel("tiempo") # Inserta el título del eje X
plt.ylabel("temperatura") # Inserta el título del eje Y
plt.ioff() # Desactiva modo interactivo de dibujo
plt.plot(tap, Ti2, 'b-', label = "Perlita") # No dibuja datos de lista2
plt.ion() # Activa modo interactivo de dibujo
plt.plot(tap, Ti2, 'b-') # Dibuja datos de lista2 sin borrar datos de lista1
plt.ioff()
plt.plot(tap2, Ti2, 'b-')
plt.ion()
plt.plot(tap2, Ti2, 'b-')
plt.ioff()
plt.plot(tab, p, 'k-')
plt.ion()
plt.plot(tab, p, 'k-', label = "Bainita")

```

```
plt.ioff()
plt.plot(tab2, p, 'k-')
plt.ion()
plt.plot(tab2, p, 'k-')
plt.ioff()
plt.plot(tab3, p, 'k--')
plt.ion()
plt.plot(tab3, p, 'k--')
plt.ioff()
plt.plot(q, z)
plt.ion()
plt.plot(q, z)
plt.legend()
```

Figura A-3. Código de Python completo para obtención del diagrama TTT del Acero 5140.

Fuente: (CAMPILLO & SEQUERA, 2018)

ANEXO B: Codificación de los programas para elaborar los diagramas CCT.

```
#Librerías que se van a utilizar

import matplotlib.pyplot as plt

from sympy import Symbol

from sympy import integrate

from scipy.integrate import quad

import math

#Se declaran las variables

To=850

R=8.31447 #Constante internacional de los gases

N=7

C=0.26

Mn=0.87

Cr=0

Si=0.15

Mo=0.26

Ni=0

Tp=670 #Temperatura de estabilización de Perlita

Tf=760 #Temperatura de estabilización de Ferrita
```

Tb=844-597*C-63*Mn-16*Ni-78*Cr #Temperatura de estabilización de
Bainita

Ms=521-353*C-24.3*Mn-225*Si-27.4*Ni-17.7*Cr-25.8*Mo #Temperatura de
inicio de transformación de Martensita

Ms = int(Ms)

Ti=Ms #Temperaturas de estudio

Tb = int(Tb)

#Se resuelve la integral de Kirkaldy

#Se resuelve la integral de Kirkaldy

x=Symbol('x')

integrate(1/(x**(2*(1-x)/3)*(1-x)**(2*x/3)),x)

f=lambda x:1/(x**(2*(1-x)/3)*(1-x)**(2*x/3))

I=quad(f,0,0.99)

I2=quad(f,0,0.1)

I3=quad(f,0,0.5)

#Se definen variables como listas

#Transformacion de austenita a Ferrita

taf=list()

```
taf2=list()
```

```
#Transformacion de austenita a Perlita
```

```
tap=list()
```

```
tap2=list()
```

```
tap3=list()
```

```
#Transformacion de austenita a Bainita
```

```
tab=list()
```

```
tab2=list()
```

```
tab3=list()
```

```
#Uso practico
```

```
Ti2=list()
```

```
p=list()
```

```
while Ti<860:
```

```
#Ecuacion para transformacion de austenita a Ferrita
```

```
Df=math.e**(-23500/(R*Ti))
```

```

    taf.append(((60*Mn+2*Ni+68*Cr+244*Mo)/(6**2**(N/8)*(Tf-
(Ti)**3*Df)*I*10**5)

    taf2.append(((60*Mn+2*Ni+68*Cr+244*Mo)/(6**2**(N/8)*(Tf-
(Ti)**3*Df)*I2*10**4)

#Ecuacion para la transformacion de Austenita a Perlita

sp=1/(math.e**(-27500/(R*Ti)))+(0.5*Mo)/(math.e**(-37000/(R*Ti)))

Dp=sp**-1

    tap.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti)**3*Dp)*I*10**6)

    tap2.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti)**3*Dp)*I2*10**5)

    tap3.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti)**3*Dp)*I3*10**5)

if Ti < Tb:

    tab.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti)**2*math.e**(-27500/(R*Ti)))*I*10**6)

    tab2.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti)**2*math.e**(-27500/(R*Ti)))*I2*10**5)

    tab3.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti)**2*math.e**(-27500/(R*Ti)))*I3*10**5)

p.append(Ti)

```

```
Ti2.append(Ti)
```

```
Ti=Ti+2
```

```
#Procedimiento para graficar linea Ms
```

```
x=-1000
```

```
y=-1000
```

```
n=-1000
```

```
q=list()
```

```
z=list()
```

```
while y<10**6:
```

```
    x=n+1
```

```
    n=n+1
```

```
    y=y+1
```

```
    q.append(x)
```

```
z.append(Ms)
```

```
Xb=10
```

```

while i<=700:

if Ti<Ms:

    Fm=1-math.e**(-0.011*(Ms-Ti))

if Xb<=1:

    FunXb=Dt/tab2;

    Xb=Xb+FunXb;

    Xib=Xb;

else:

if Xb<100 and Xb>=1:

tb=Dt(i-1)

Tbt(j)=T(i-1)

    Xb=10000000000000000;

    h=i-1;

    Fib=0;

    Titajb=Dt(i);

nb=0;

bb=0;

while h<=length(tab2):

if Fib<1:

```

```

        nb(h)=(log((log(1-Voli))./(log(1-
Volf))))/(log(TauB01(h)./TauB(h)));

        bb(h)= -(log(1-Voli))./(TauB01(h).^nb(h));

        Fib=1-exp(-bb(h).*(Titajb+0.5).^nb(h));

        Titajb=(-log(1-Fib)/bb(h)).^(1./nb(h));

if Ti<Tp:

if Xp<=1:

        FunXp=Dt/tap2

        Xp=Xp+FunXp;

        Xip=Xp;

else:

if Xp<10 && Xp>=1

        tp(j)=Dt(i-1);

        Tpt(j)=T(i-1);

        Xp=1000000;

        h=i-1;

        Fip=0;

        Titajp=Dt(i);

np=0;%Parametro n para la Perlita

bp=0;%Parametro b para la Perlita

```

```

Fipx=0;

while h<=length(tap2):

if Fip<1:

    np(h)=(log((log(1-Voli))./(log(1-Volf))))./log(TauP01(h)./TauP(h));

    bp(h)= -(log(1-Voli))./(TauP01(h).^np(h));

    Fip=1-exp(-bp(h).*(Titajp+0.5).^np(h));

    Titajp=((-log(1-Fip))/bp(h)).^(1./np(h));

for i in range(1, 193, 4):

Ti2[i]+=3

Ti2.insert(i, Ti2[i])

del Ti2[i+1]

for i in range(3, 195, 4):

Ti2[i]-=3

Ti2.insert(i, Ti2[i])

```

```

del Ti2[i+1]

for j in range(1, 73, 4):

    p[j]+=25

p.insert(j, p[j])

del p[j+1]

for j in range(3, 75, 4):

    p[j]-=25

p.insert(j, p[j])

del p[j+1]

plt.figure(dpi = 300)

plt.xscale('log')

plt.xlim(0.1, 10**6)

plt.ylim(0, 800)

plt.title('Diagrama CCT Acero AISI 4027')

plt.plot(taf2, Ti2, 'r-', linewidth = 1, label = "Ferrita") # Dibuja el gráfico

plt.xlabel("Tiempo (s)") # Inserta el título del eje X

plt.ylabel("Temperatura (°C)") # Inserta el título del eje Y

plt.ioff() # Desactiva modo interactivo de dibujo

plt.plot(tap, Ti2, 'b-', linewidth = 1, label = "Perlita") # No dibuja datos de
lista2

```

```

plt.ion() # Activa modo interactivo de dibujo

plt.plot(tap, Ti2, 'b-', linewidth = 1) # Dibuja datos de lista2 sin borrar datos de
lista1

plt.ioff()

plt.plot(tap2, Ti2, 'b-', linewidth = 1)

plt.ion()

plt.plot(tap2, Ti2, 'b-', linewidth = 1)

plt.ioff()

plt.plot(tab2, p, 'k-', linewidth = 1, label = "Bainita")

plt.ion()

plt.plot(tab2, p, 'k-', linewidth = 1)

plt.ioff()

plt.plot(q, z, linewidth = 1)

plt.ion()

plt.plot(q, z, linewidth = 1, label = "Inicio de transformación de Martensita")

plt.legend()

```

Figura B-1. Código de Python completo para obtención del diagrama CCT del Acero 4027.

Fuente: (CAMPILLO & SEQUERA, 2018)

```
#Librerias que se van a utilizar
import matplotlib.pyplot as plt
from sympy import Symbol
from sympy import integrate
from scipy.integrate import quad
import math
```

```
#Se declaran las variables
```

```
To=850
```

```
R=8.31447 #Constante internacional de los gases
```

```
N=7.5
```

```
C=0.42
```

```
Mn=0.78
```

```
Cr=0.8
```

```
Si=0
```

```
Mo=0.33
```

```
Ni=1.79
```

```
Tp=655 #Temperatura de estabilización de Perlita
```

```
Tf=710 #Temperatura de estabilización de Ferrita
```

```
Tb=830-270*C-90*Mn-37*Ni-70*Cr-83*Mo #Temperatura de estabilización
```

```
de Bainita
```

```
Ms=521-353*C-24.3*Mn-225*Si-27.4*Ni-17.7*Cr-25.8*Mo
```

```
Ms = int(Ms) #Temperatura de inicio de transformación de Martensita
```

```
Ti=Ms
```

```
Tb = int(Tb)
```

```
#Se resuelve la integral de Kirkaldy
```

```
#Se resuelve la integral de Kirkaldy
x=Symbol('x')
integrate(1/(x**(2*(1-x)/3)*(1-x)**(2*x/3)),x)
f=lambda x:1/(x**(2*(1-x)/3)*(1-x)**(2*x/3))
I=quad(f,0,0.99)
I2=quad(f,0,0.1)
I3=quad(f,0,0.5)
```

```
#Se definen variables como listas
#Transformacion de austenita a Ferrita
taf=list()
taf2=list()
```

```
#Transformacion de austenita a Perlita
tap=list()
tap2=list()
tap3=list()
```

```
#Transformacion de austenita a Bainita
tab=list()
tab2=list()
tab3=list()
```

```
#Uso practico
Ti2=list()
p=list()
```

```
while Ti<860:
```

```

#Ecuacion para transformacion de austenita a Ferrita
Df=math.e**(-23500/(R*Ti))
taf.append(((60*Mn+2*Ni+68*Cr+244*Mo)/(6**2**(N/8)*(Tf-
(Ti))**3*Df)*I*10**5)
taf2.append(((60*Mn+2*Ni+68*Cr+244*Mo)/(6**2**(N/8)*(Tf-
(Ti))**3*Df)*I2*10**4)

#Ecuacion para la tranformacion de Austenita a Perlita
sp=1/(math.e**(-27500/(R*Ti)))+(0.5*Mo)/(math.e**(-37000/(R*Ti)))
Dp=sp**-1
tap.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti))**3*Dp)*I*10**6)
tap2.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti))**3*Dp)*I2*10**5)
tap3.append(((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti))**3*Dp)*I3*10**5)

if Ti < Tb:
    tab.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti))**2*math.e**(-27500/(R*Ti)))*I*10**6)
    tab2.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti))**2*math.e**(-27500/(R*Ti)))*I2*10**5)
    tab3.append((((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti))**2*math.e**(-27500/(R*Ti)))*I3*10**5)

p.append(Ti)
Ti2.append(Ti)
Ti=Ti+3

```

```
#Procedimiento para graficar linea Ms
```

```
x=-1000
```

```
y=-1000
```

```
n=-1000
```

```
q=list()
```

```
z=list()
```

```
while y<10**6:
```

```
    x=n+1
```

```
    n=n+1
```

```
    y=y+1
```

```
    q.append(x)
```

```
z.append(Ms)
```

```
Xb=10
```

```
while i<=700:
```

```
if Ti<Ms:
```

```
    Fm=1-math.e**(-0.011*(Ms-Ti))
```

```
if Xb<=1:
```

```
    FunXb=Dt/tab2;
```

```
    Xb=Xb+FunXb;
```

```
    Xib=Xb;
```

```
else:
```

```
if Xb<100 and Xb>=1:
```

```
tb=Dt(i-1)
```

```
Tbt(j)=T(i-1)
```

```
    Xb=10000000000000000;
```

```
    h=i-1;
```

```
    Fib=0;
```

```

        Titajb=Dt(i);
nb=0;
bb=0;
while h<=length(tab2):
if Fib<1:
        nb(h)=(log((log(1-Voli))./(log(1-
Volf))))/(log(TauB01(h)./TauB(h)));
        bb(h)= -(log(1-Voli))./(TauB01(h).^nb(h));
        Fib=1-exp(-bb(h).*(Titajb+0.5).^nb(h));
        Titajb=((-log(1-Fib))/bb(h)).^(1./nb(h));
if Ti<Tp:
if Xp<=1:
        FunXp=Dt/tap2
        Xp=Xp+FunXp;
        Xip=Xp;
else:
if Xp<10 && Xp>=1
tp(j)=Dt(i-1);
Tpt(j)=T(i-1);
        Xp=1000000;
        h=i-1;
        Fip=0;
        Titajp=Dt(i);
np=0;%Parametro n para la Perlita
bp=0;%Parametro b para la Perlita
        Fipx=0;

while h<=length(tap2):
if Fip<1:

```

```
np(h)=(log((log(1-Voli))./(log(1-Volf))))./log(TauP01(h)./TauP(h));
bp(h)= -((log(1-Voli))./(TauP01(h).^np(h)));
```

```
Fip=1-exp(-bp(h).*(Titajp+0.5).^np(h));
Titajp=((-log(1-Fip))/bp(h)).^(1./np(h));
```

```
for i in range(1, 193, 4):
    Ti2[i]+=10
    Ti2.insert(i, Ti2[i])
    del Ti2[i+1]
```

```
for i in range(3, 195, 4):
    Ti2[i]-=10
    Ti2.insert(i, Ti2[i])
    del Ti2[i+1]
```

```
for j in range(1, 73, 4):
    p[j]+=10
    p.insert(j, p[j])
    del p[j+1]
```

```
for j in range(3, 75, 4):
    p[j]-=10
    p.insert(j, p[j])
    del p[j+1]
```

```
plt.figure(dpi = 300)
plt.xscale('log')
```

```

plt.xlim(0.1, 10**6)
plt.ylim(0, 800)
plt.title('Diagrama CCT Acero AISI 4340')
plt.plot(taf2, Ti2, 'r-', linewidth = 1, label = "Ferrita") # Dibuja el gráfico
plt.xlabel("Tiempo (s)") # Inserta el título del eje X
plt.ylabel("Temperatura (°C)") # Inserta el título del eje Y
plt.ioff() # Desactiva modo interactivo de dibujo
plt.plot(tap, Ti2, 'b-', linewidth = 1, label = "Perlita") # No dibuja datos de
lista2
plt.ion() # Activa modo interactivo de dibujo
plt.plot(tap, Ti2, 'b-', linewidth = 1) # Dibuja datos de lista2 sin borrar datos de
lista1
plt.ioff()
plt.plot(tap2, Ti2, 'b-', linewidth = 1)
plt.ion()
plt.plot(tap2, Ti2, 'b-', linewidth = 1)
plt.ioff()
plt.plot(tab2, p, 'k-', linewidth = 1, label = "Bainita")
plt.ion()
plt.plot(tab2, p, 'k-', linewidth = 1)
plt.ioff()
plt.plot(q, z, linewidth = 1)
plt.ion()
plt.plot(q, z, linewidth = 1, label = "Inicio de transformación de Martensita")
plt.legend()

```

Figura B-2. Código de Python completo para obtención del diagrama CCT del Acero 4340.

Fuente: (CAMPILLO & SEQUERA, 2018)

```
#Librerias que se van a utilizar

import matplotlib.pyplot as plt

from sympy import Symbol

from sympy import integrate

from scipy.integrate import quad

import math

To=850

R=8.31447

N=5.5

C=0.41

Mn=0.75

Cr=1.05

Si=0.1

Mo=0

Ni=0

Tp=655

Tf=725

Tb=830-270*C-90*Mn-37*Ni-70*Cr-83*Mo
```

Ms=521-353*C-24.3*Mn-225*Si-27.4*Ni-17.7*Cr-25.8*Mo

Ms = int(Ms)

Ti=Ms

Tb = int(Tb)

#Se resuelve la integral de Kirkaldy

x=Symbol('x')

integrate(1/(x**(2*(1-x)/3)*(1-x)**(2*x/3)),x)

f=lambdax:1/(x**(2*(1-x)/3)*(1-x)**(2*x/3))

I=quad(f,0,0.99)

I2=quad(f,0,0.1)

I3=quad(f,0,0.5)

#Se definen variables como listas

#Transformacion de austenita a Ferrita

taf=list()

taf2=list()

#Transformacion de austenita a Perlita

tap=list()

```

tap2=list()

tap3=list()

#Transformacion de austenita a Bainita

tab=list()

tab2=list()

tab3=list()

#Uso practico

Ti2=list()

p=list()

while Ti<860:

    #Ecuacion para transformacion de austenita a Ferrita

    Df=math.e**(-23500/(R*Ti))

    taf.append(((60*Mn+2*Ni+68*Cr+244*Mo)/(6**2**(N/8)*(Tf-
(Ti))**3*Df)*I*10**5)

    taf2.append(((60*Mn+2*Ni+68*Cr+244*Mo)/(6**2**(N/8)*(Tf-
(Ti))**3*Df)*I2*10**4)

    #Ecuacion para la transformacion de Austenita a Perlita

    sp=1/(math.e**(-27500/(R*Ti)))+(0.5*Mo)/(math.e**(-37000/(R*Ti)))

    Dp=sp**-1

```

```

tap.append((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti)**3*Dp)*I*10**6)

tap2.append((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti)**3*Dp)*I2*10**5)

tap3.append((1.8+5.4*(Cr+Mo+4*Mo*Ni))/(6**2**(N/8)*(Tp-
Ti)**3*Dp)*I3*10**5)

if Ti < Tb:

    tab.append(((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti)**2*math.e**(-27500/(R*Ti)))*I*10**6)

    tab2.append(((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti)**2*math.e**(-27500/(R*Ti)))*I2*10**5)

    tab3.append(((2.33+10*C+4*Cr+19*Mo)*10**-4)/(6**2**(N/8)*(Tb-
Ti)**2*math.e**(-27500/(R*Ti)))*I3*10**5)

p.append(Ti)

Ti2.append(Ti)

Ti=Ti+2

#Procedimiento para graficar linea Ms

x=-1000

y=-1000

n=-1000

q=list()

```

```

z=list()

while y<10**6:

    x=n+1

    n=n+1

    y=y+1

    q.append(x)

z.append(Ms)

Xb=10

while i<=700:

    if Ti<Ms:

        Fm=1-math.e**(-0.011*(Ms-Ti))

    if Xb<=1:

        FunXb=Dt/tab2;

        Xb=Xb+FunXb;

        Xib=Xb;

    else:

        if Xb<100 and Xb>=1:

            tb=Dt(i-1)

            Tbt(j)=T(i-1)

```

```

Xb=10000000000000000;

h=i-1;

Fib=0;

Titajb=Dt(i);

nb=0;

bb=0;

while h<=length(tab2):

if Fib<1:

        nb(h)=(log((log(1-Voli))./(log(1-
Volf))))/(log(TauB01(h)./TauB(h)));

        bb(h)=-((log(1-Voli))./(TauB01(h).^nb(h)));

        Fib=1-exp(-bb(h).*(Titajb+0.5).^nb(h));

        Titajb=((-log(1-Fib))/bb(h)).^(1./nb(h));

if Ti<Tp:

if Xp<=1:

        FunXp=Dt/tap2

        Xp=Xp+FunXp;

        Xip=Xp;

else:

if Xp<10 && Xp>=1

```

```

tp(j)=Dt(i-1);

Tpt(j)=T(i-1);

        Xp=1000000;

        h=i-1;

        Fip=0;

        Titajp=Dt(i);

np=0;%Parametro n para la Perlita

bp=0;%Parametro b para la Perlita

        Fipx=0;

while h<=length(tap2):

if Fip<1:

        np(h)=(log((log(1-Voli))./(log(1-Volf))))./log(TauP01(h)./TauP(h));

bp(h)= -(log(1-Voli))./(TauP01(h).^np(h));

        Fip=1-exp(-bp(h).*(Titajp+0.5).^np(h));

        Titajp=((-log(1-Fip))/bp(h)).^(1./np(h));

```

```
for i in range(1, 193, 4):
```

```
    Ti2[i]+=5
```

```
    Ti2.insert(i, Ti2[i])
```

```
    del Ti2[i+1]
```

```
for i in range(3, 195, 4):
```

```
    Ti2[i]-=5
```

```
    Ti2.insert(i, Ti2[i])
```

```
    del Ti2[i+1]
```

```
for j in range(1, 73, 4):
```

```
    p[j]+=25
```

```
    p.insert(j, p[j])
```

```
    del p[j+1]
```

```
for j in range(3, 75, 4):
```

```
    p[j]-=25
```

```
    p.insert(j, p[j])
```

```
    del p[j+1]
```

```

plt.figure(dpi = 300)

plt.xscale('log')

plt.xlim(0.1, 10**6)

plt.ylim(0, 800)

plt.title('Diagrama CCT Acero AISI 5140')

plt.plot(taf2, Ti2, 'r-', linewidth = 1, label = "Ferrita") # Dibuja el gráfico

plt.xlabel("Tiempo (s)") # Inserta el título del eje X

plt.ylabel("Temperatura (°C)") # Inserta el título del eje Y

plt.ioff() # Desactiva modo interactivo de dibujo

plt.plot(tap, Ti2, 'b-', linewidth = 1, label = "Perlita") # No dibuja datos de
lista2

plt.ion() # Activa modo interactivo de dibujo

plt.plot(tap, Ti2, 'b-', linewidth = 1) # Dibuja datos de lista2 sin borrar datos de
lista1

plt.ioff()

plt.plot(tap2, Ti2, 'b-', linewidth = 1)

plt.ion()

plt.plot(tap2, Ti2, 'b-', linewidth = 1)

plt.ioff()

```

```
plt.plot(tab2, p, 'k-', linewidth = 1, label = "Bainita")

plt.ion()

plt.plot(tab2, p, 'k-', linewidth = 1)

plt.ioff()

plt.plot(q, z, linewidth = 1)

plt.ion()

plt.plot(q, z, linewidth = 1, label = "Inicio de transformación de Martensita")

plt.legend()
```

Figura B-3. Código de Python completo para obtención del diagrama CCT del Acero 5140.

Fuente: (CAMPILLO & SEQUERA, 2018)