

**PREDICCIÓN DE PROPIEDADES TERMOFÍSICAS PROVENIENTES DE  
DATOS P<sub>ρ</sub>T DE SISTEMAS BINARIOS ALCANO-CLOROALCANO, USANDO  
UN MÉTODO DE CONTRIBUCIÓN DE GRUPOS**

**MARIA ALEJANDRA QUINTERO MONROY**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICOQUÍMICAS  
ESCUELA DE INGENIERÍA QUÍMICA  
BUCARAMANGA**

**2018**

**PREDICCIÓN DE PROPIEDADES TERMOFÍSICAS PROVENIENTES DE  
DATOS  $P\rho T$  DE SISTEMAS BINARIOS ALCANO-CLOROALCANO, USANDO  
UN MÉTODO DE CONTRIBUCIÓN DE GRUPOS**

**MARIA ALEJANDRA QUINTERO MONROY**

**Trabajo de grado presentado como requisito para optar el título de:  
Ingeniero químico**

**Director:**

**Prof. Hernando Guerrero Amaya, Ph.D.**

**Codirector:**

**PROF. MARTHA JOSEFINA PARRA RAMÍREZ, PH.D.**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISCOQUÍMICAS  
ESCUELA DE INGENIERÍA QUÍMICA  
BUCARAMANGA**

**2018**

*A Dios, por bendecirme con conocimiento a lo largo de mi carrera y por darme fortaleza y guía en los momentos difíciles.*

*A mi esposo, Héctor David, por enseñarme con paciencia y amor, por creer en mí, por su tiempo, apoyo incondicional y por cada palabra de ánimo y aliento. Gracias por compartir tu vida conmigo, y caminar junto a mí en cada paso.*

*A Ana Lucía, por ser mi inspiración, por las lindas sonrisas, abrazos y besos que alegran mis días, por ser mi compañera de estudio, mi hija y mi pequeño amor.*

*A mis padres, por enseñarme la importancia del estudio, la disciplina y el trabajo, porque este logro no es solo mío, sino de ellos también.*

*A mis suegros, por el tiempo, la dedicación y el amor que dan a Ana Lucía, por su apoyo incondicional y sabios consejos.*

## **AGRADECIMIENTOS**

A la **Universidad Industrial de Santander** y a la **Escuela de Ingeniería Química**, por los conocimientos brindados a lo largo de estos años.

Al **Prof. Hernando Guerrero Amaya, Ph.D.**, por suministrar los datos experimentales utilizados para la realización de esta investigación.

Al **Prof. Hernando Guerrero Amaya, Ph.D.** y a la **Prof. Martha Josefina Parra Ramírez**, por su tiempo, correcciones, dedicación y guía en desarrollo del presente trabajo.

## TABLA DE CONTENIDO

	<b>Pág.</b>
INTRODUCCIÓN .....	17
1. ESTADO DEL ARTE Y MARCO TEÓRICO .....	19
1.1. ESTRUCTURA Y PROPIEDADES DE LOS ALCANOS .....	19
1.2. ESTRUCTURA Y PROPIEDADES DE LOS CLOROALCANOS.....	20
1.3. SISTEMAS BINARIOS ALCANO-CLOROALCANO.....	21
1.4. MODELAMIENTO DE PROPIEDADES TERMOFÍSICAS .....	21
1.4.1. Ecuaciones para las propiedades termofísicas de sustancias puras: .....	22
1.4.2. Ecuaciones para las propiedades termofísicas de sistemas binarios:.....	23
2. OBJETIVOS .....	25
2.1. OBJETIVO GENERAL .....	25
2.2. OBJETIVOS ESPECÍFICOS.....	25
3. METODOLOGÍA.....	26
3.1. IDENTIFICACIÓN DE GRUPOS DE CONTRIBUCIÓN PRESENTES EN LOS ALCANOS Y CLOROALCANOS.....	27
3.2. OBTENCIÓN DE MODELOS DE DENSIDAD A PARTIR DE DATOS EXPERIMENTALES EN PYTHON.....	27
3.2.1. Modelo de densidad de alcanos:.....	28
3.2.2. Modelo de densidad de cloroalcanos: .....	29
3.2.3. Modelo de densidad de mezclas binarias alcano-cloroalcano:.....	29
3.3. PREDICCIÓN DE PROPIEDADES TERMOFÍSICAS .....	30
3.4. COMPARACIÓN DE RESULTADOS.....	30
4. RESULTADOS Y ANÁLISIS.....	31

4.1.	MODELO DE DENSIDAD DE SUSTANCIAS PURAS PARA ALCANOS ....	31
4.2.	MODELO DE DENSIDAD DE SUSTANCIAS PURAS PARA CLOROALCANOS .....	33
4.3.	MODELO DE DENSIDAD DE MEZCLAS PARA SISTEMAS BINARIOS ALCANO-CLOROALCANO.....	34
4.4.	PREDICCIÓN DE PROPIEDADES TERMOFÍSICAS DE ALCANOS Y CLOROALCANOS .....	36
4.5.	PREDICCIÓN DE PROPIEDADES DE EXCESO PARA SISTEMAS BINARIOS ALCANO-CLOROALCANO.....	37
4.6.	VALIDACIÓN DE LOS MODELOS .....	38
4.6.1.	Validación del modelo de densidad de alcanos.....	39
4.6.2.	Validación del modelo de densidad de cloroalcanos .....	41
4.6.3.	Validación del modelo de densidad de mezclas binarias .....	42
5.	CONCLUSIONES.....	44
	RECOMENDACIONES.....	45
	BIBLIOGRAFÍA.....	46
	ANEXOS.....	54

## LISTA DE TABLAS

	<b>Pág.</b>
Tabla 1. Clasificación de átomos de carbono en alcanos. ....	19
Tabla 2. Coeficientes obtenidos para el modelo de densidad de alcanos. ....	32
Tabla 3. Coeficientes obtenidos para el modelo de densidad de cloroalcanos.....	33
Tabla 4. Coeficientes para el modelo de mezclas binarias alcano-cloroalcano .....	35
Tabla 5. Desviación estándar y DRCM de expansibilidad isobárica, compresibilidad isotérmica y presión interna de alcanos y cloroalcanos. ....	37
Tabla 6. Desviación estándar de propiedades de exceso (volumen, expansibilidad isobárica y compresibilidad isotérmica) para mezclas con n-hexano y n-heptano.	38

## LISTA DE FIGURAS

	<b>Pág.</b>
Figura 1. Diagrama metodológico de la creación y validación de un modelo de contribución de grupos usando aprendizaje automático en Python.....	26
Figura 2. Diagrama de dispersión con los datos experimentales (·) y el modelo obtenido (-) para la densidad de alcanos (izda.) y cloroalcanos (dcha.).....	31
Figura 3. Variación de la densidad experimental (·) y modelada (-) con respecto a la presión, a diferentes temperaturas para el 2-metilpentano (izda.) y 3-metilpentano (dcha.).....	32
Figura 4. Variación de la densidad experimental (·) y modelada (-) con respecto a la presión, a diferentes temperaturas para el iso-clorobutano (izda.) y ter-clorobutano (dcha.).....	34
Figura 5. Diagrama de dispersión con los datos experimentales (·) y el modelo obtenido (-) para la densidad de mezclas binarias alcano-cloroalcano.....	35
Figura 6. Variación de la densidad experimental (·) y modelada (-) con respecto a la fracción molar de n-hexano, a diferentes temperaturas y 30 MPa para las mezclas n-hexano y 2-clorobutano (izda.) y n-heptano e iso-clorobutano (dcha.).....	36
Figura 7. Comparación de la variación de la densidad del n-pentano experimental con la estimada por (de izda. a dcha.) el modelo de propuesto, Peng Robinson y COSTALD.....	40

Figura 8. Comparación de las densidades del isopentano (sup. izda.), nonano (sup. dcha.), undecano (inf. izda.) y dodecano (inf. dcha.) con el modelo de alcanos, Peng-Robinson y COSTALD. ....	40
Figura 9. Comparación de la densidad del 1-clorobutano (izda.) y del 1-cloroheptano (dcha.) con el modelo de cloroalcanos, Peng-Robinson y COSTALD. ....	41
Figura 10. Comparación de la densidad experimental del 1-cloropropano con la densidad estimada por (de izda a dcha.) el modelo, Peng-Robinson y COSTALD. ....	42
Figura 11. Comparación de la densidad de las mezclas binarias n-decano y 1-clorobutano (izda.) y n-pentano y 1-clorooctano (dcha.) con el modelo de mezclas, Peng-Robinson y COSTALD.....	43

## LISTA DE ANEXOS

	<b>Pág.</b>
Anexo A. Matrices de los datos utilizados para el entrenamiento de los modelos de densidad. ....	54
Anexo B. Grupos de contribución de algunos alcanos y cloroalcanos.....	65
Anexo C. Gráficas de variación de densidad experimental y modelada. ....	67
Anexo D. Gráficas de comparación de los modelos propuestos con datos provenientes de la literatura y estimaciones de Peng-Robinson y COSTALD.....	73
Anexo E. Códigos de programación en Python. ....	81

## GLOSARIO

<b>Término</b>	<b>Descripción</b>
$a - g$	Coefficientes de los modelos de densidades líquidas
C <sub>1</sub> -C <sub>2</sub>	Enlace entre un carbono primario y un carbono secundario
C <sub>1</sub> -C <sub>3</sub>	Enlace entre un carbono primario y un carbono terciario
C <sub>1</sub> -C <sub>4</sub>	Enlace entre un carbono primario y un carbono cuaternario
C <sub>2</sub> -C <sub>2</sub>	Enlace entre un carbono secundario y un carbono secundario
C <sub>2</sub> -C <sub>3</sub>	Enlace entre un carbono secundario y un carbono terciario
C <sub>2</sub> -C <sub>4</sub>	Enlace entre un carbono secundario y un carbono cuaternario
C <sub>3</sub> -C <sub>3</sub>	Enlace entre un carbono terciario y un carbono terciario
C <sub>1</sub> -Cl	Enlace entre un carbono primario y un cloro
C <sub>2</sub> -Cl	Enlace entre un carbono secundario y un cloro
C <sub>3</sub> -Cl	Enlace entre un carbono terciario y un cloro
$N$	Numero de datos experimentales
$n_j$	Cantidad de ocurrencias del enlace j (grupo de contribución)
$\bar{M}$	Peso molecular promedio
$M_i$	Peso molecular de compuesto i
$P$	Presión
$p_i$	Presión interna
$p_{i,i}$	Presión interna del compuesto i
$p_i^E$	Presión interna de exceso
$T$	Temperatura
$V$	Volumen molar
$V_i$	Volumen molar del compuesto i
$V_M$	Volumen molar de la mezcla M
$V^E$	Volumen de exceso
$x_i$	Fracción molar del compuesto i (i=1 es para alcanos; i=2 es para cloroalcanos)

<b>Letra Griega</b>	<b>Descripción</b>
$\alpha_p$	Expansibilidad isobárica
$\alpha_p^E$	Expansibilidad isobárica de exceso
$\alpha_{p,i}$	Expansibilidad isobárica del compuesto i
$\kappa_T$	Compresibilidad isotérmica
$\kappa_T^E$	Compresibilidad isotérmica de exceso
$\kappa_{T,i}$	Compresibilidad isotérmica del compuesto i
$\phi_i$	Fracción de volumen ideal
$\psi_i$	Fracción de compresibilidad
$\rho_i$	Densidad líquida del compuesto i
$\rho_M$	Densidad líquida de la mezcla M
$\rho_{i,exp}$	Densidad líquida experimental
$\rho_{i,cal}$	Densidad líquida calculada
$\rho_{i,alcanos}$	Densidad líquida de los alcanos puros
$\rho_{i,cloroalcanos}$	Densidad líquida cloroalcanos puros
$\sigma$	Desviación estándar

## RESUMEN

**TITULO:** PREDICCIÓN DE PROPIEDADES TERMOFÍSICAS PROVENIENTES DE DATOS P<sub>p</sub>T DE SISTEMAS BINARIOS ALCANO-CLOROALCANO, USANDO UN MÉTODO DE CONTRIBUCIÓN DE GRUPOS\*

**AUTORA:** MARIA ALEJANDRA QUINTERO MONROY\*\*

**PALABRAS CLAVES:** Propiedades termofísicas, alcanos, cloroalcanos, sistemas alcano-cloroalcano, Python, aprendizaje automático, contribución de grupos.

### DESCRIPCIÓN:

El diseño de procesos químicos es de gran importancia en la industria y se beneficia, en gran medida, de la estimación de propiedades termofísicas con alto grado de precisión. Entre los compuestos químicos altamente usados en la industria química, farmacéutica y petrolera se encuentran los alcanos, cloroalcanos y sus mezclas binarias. Por tal razón, es deseable el proponer un modelo que estime dichas propiedades de manera sencilla. Se realizó el modelamiento de densidades para alcanos, cloroalcanos y sus mezclas binarias en Python, mediante aprendizaje automático por contribución de grupos a partir de datos experimentales P<sub>p</sub>T, realizados por Guerrero Amaya en el 2012. Los modelos propuestos son regresiones lineales multivariantes entrenados con tan solo el 5% de los datos experimentales disponibles y con desviación relativa cuadrática media (DRCM) menor al 0,34. Además de ser modelos sencillos, se evidencia su versatilidad al poder utilizarlos para estimar las densidades de compuestos y mezclas no incluidos y a condiciones fuera del rango de temperaturas y presiones estudiados por Guerrero Amaya. Los modelos superan la ecuación de estado de Peng-Robinson y, en algunos casos, las predicciones obtenidas con correlaciones avanzadas para densidades líquidas (ej. Modelo COSTALD). Los modelos obtenidos también permiten estimar propiedades termofísicas como la expansibilidad isobárica, compresibilidad isotérmica y presión interna, tanto de sustancias puras como de mezclas. Dichas propiedades se obtienen mediante expresiones explícitas sencillas en función de los coeficientes de los modelos y las condiciones de operación.

---

\* Trabajo de Grado

\*\* Facultad de ingenierías fisicoquímicas. Escuela de Ingeniería Química.  
Director: Hernando Guerrero Amaya, Ingeniero Químico, PhD.

## ABSTRACT

**TITLE:** PREDICTION OF THERMOPHYSICAL PROPERTIES FROM P<sub>p</sub>T DATA FOR ALKANE-CHLOROALKANE BINARY SYSTEMS, USING A GROUP CONTRIBUTION METHOD\*

**AUTHOR:** MARIA ALEJANDRA QUINTERO MONROY\*\*

**KEYWORDS:** Thermophysical properties, alkanes, chloroalkanes, alkane-chloroalkane systems, Python, machine learning, group contributions.

### DESCRIPTION:

The design of chemical processes is of great importance in industry and greatly benefits from the accurate estimation of thermophysical properties. Alkanes, chloroalkanes and their binary mixtures are among the chemical species most used in the chemical, pharmaceutical and petroleum industries. For this reason, proposing a model that can estimate their thermophysical properties accurately and readily is of great interest. A group contribution liquid density model for alkanes, chloroalkanes and their binary mixtures was fit to P<sub>p</sub>T experimental data obtained from Guerrero Amaya (2012) using Python's machine learning tools. The proposed models consist of multivariate linear regressions, which were trained with only 5% of the experimental dataset. These models provide high accuracy with normalized root-mean-square errors (NRMSE) of less than 0.34%. In addition to being simple models, these models are very flexible, as they can be used to estimate liquid densities of compounds and mixtures not included or under conditions outside the temperature and pressure ranges studied by Guerrero Amaya. These models provide superior results to Peng-Robinson calculations and, in some cases, surpass the performance of advanced correlations for liquid densities (i.e. COSTALD). Thermophysical properties such as isobaric thermal expansivity, isothermal compressibility and internal pressure of pure substances and mixtures can be calculated from these models via simple explicit expressions that depend on model coefficients and operating conditions.

---

\* Bachelor Thesis

\*\* Facultad de ingenierías fisicoquímicas. Escuela de Ingeniería Química.  
Director: Hernando Guerrero Amaya, Ingeniero Químico, PhD.

## INTRODUCCIÓN

Con la alta competitividad de la industria química, es de gran importancia el diseñar procesos químicos óptimos que permitan ahorros energéticos y económicos [1]. Al existir millones de compuestos químicos e interacciones entre ellos, es de gran valor tener modelos que permitan estimar propiedades termofísicas, tales como densidad, entalpia, y volumen de exceso, entre otras, sin necesidad de obtenerlas experimentalmente. En la actualidad, se han indexado alrededor de 10 millones de compuestos químicos. Sin embargo, los paquetes de simulación cuentan con modelos para tan solo el 0,02% de dichos compuestos [2].

Los alcanos y los cloroalcanos son de gran importancia en la industria química y farmacéutica. Los alcanos se utilizan como combustibles, solventes, aceites térmicos y en la síntesis de compuestos químicos, entre otras [3]. Dado que son buenos receptores de electrones [4], los cloroalcanos también son buenos solventes y útiles como materia prima para síntesis de diversos compuestos [5].

Debido a los altos costos y tiempo requeridos para la realización de experimentos, el modelamiento ofrece una mejor alternativa para entender los sistemas alcano-cloroalcano. Hoy en día, existen diversas ecuaciones de estado [6] [7] y correlaciones [8] [9] [10] para modelar el comportamiento PVT de sustancias puras y mezclas. Sin embargo, estas ecuaciones tienen las desventajas de que son complejas, haciendo de su implementación una tarea no trivial, y sólo se pueden usar con compuestos para los que se tienen los parámetros de modelamiento.

Se propone un nuevo modelo de contribución de grupos para sistemas alcano-cloroalcano, desarrollado con el lenguaje de programación Python, mediante técnicas de aprendizaje automático [11] a partir de datos experimentales del trabajo doctoral “Estudio  $p\rho T$  de alcanos, cloroalcanos y de sus mezclas binarias” [12]. El

modelo propuesto es sencillo y de alta precisión en la predicción de las propiedades termofísicas de los alcanos, cloroalcanos y sus mezclas binarias.

# 1. ESTADO DEL ARTE Y MARCO TEÓRICO

## 1.1. ESTRUCTURA Y PROPIEDADES DE LOS ALCANOS

Los alcanos son compuestos conformados por átomos de hidrógeno y carbono, unidos mediante enlaces sencillos. Se clasifican como primarios, secundarios, terciarios y cuaternarios según el número de carbonos con los que están enlazados (ver **Tabla 1**). [13] Los alcanos son compuestos no polares debido a la electronegatividad similar del hidrógeno y el carbono y también a la elasticidad de sus enlaces. Son insolubles en agua y de menor densidad. Las interacciones intermoleculares prominentes en los alcanos son las fuerzas de London, las cuales aumentan con el número de electrones de valencia. Por tanto, los alcanos con mayor peso molecular son menos volátiles y las cadenas de alcanos más largas y menos ramificadas presentan mayores entalpías de evaporación y puntos de ebullición y fusión, debido a que las moléculas pueden estar más cercanas entre sí, aumentando las fuerzas de London. Por esta razón, las cadenas ramificadas poseen fuerzas intermoleculares más débiles que sus isómeros no ramificados [14].

Tabla 1. Clasificación de átomos de carbono en alcanos.

$\begin{array}{ccccccc} & & \text{H} & & \text{H} & & \\ & &   & &   & & \\ \text{H} & - & \text{C} & - & \text{C} & - & \text{H} \\ & &   & &   & & \\ & & \text{H} & & \text{H} & & \end{array}$ <p>Un carbono primario es aquel que se enlaza con solamente un átomo de carbono.</p>	$\begin{array}{ccccccc} & & \text{H} & & \text{H} & & \text{H} \\ & &   & &   & &   \\ \text{H} & - & \text{C} & - & \text{C} & - & \text{C} & - & \text{H} \\ & &   & &   & &   \\ & & \text{H} & & \text{H} & & \text{H} \end{array}$ <p>Un carbono secundario es aquel que se enlaza con dos átomos de carbono.</p>
$\begin{array}{ccc} & \text{H} & \\ &   & \\ \text{H}_3\text{C} & - & \text{C} & - & \text{CH}_3 \\ &   & \\ & \text{CH}_3 & \end{array}$ <p>Un carbono terciario es aquel que se enlaza a tres átomos de carbono.</p>	$\begin{array}{ccc} & \text{CH}_3 & \\ &   & \\ \text{H}_3\text{C} & - & \text{C} & - & \text{CH}_3 \\ &   & \\ & \text{CH}_3 & \end{array}$ <p>Un carbono cuaternario es aquel que está enlazado a cuatro átomos de carbono.</p>

Los alcanos más livianos (entre 1 y 4 carbonos) se comportan como gases a temperatura ambiente, mientras que, desde el hexano hasta el undecano, se encuentran líquidos moderadamente volátiles, los cuales están presentes en la gasolina. Al aumentar el número de carbonos, los alcanos forman parte de otros combustibles como el jet fuel, Diesel y fuel oil. Para cadenas más largas, aumentan las interacciones moleculares, aumentando su viscosidad, hasta llegar a compuestos sólidos como las ceras parafínicas [15] [16].

Otra propiedad que caracteriza a los alcanos es su baja reactividad. No se ven afectados por altas concentraciones de ácido sulfúrico, ácido nítrico, soluciones de hidróxido de sodio en su punto de ebullición o por fuertes agentes oxidantes como el permanganato de potasio. A pesar de que los alcanos no son compuestos reactivos, experimentan dos reacciones importantes: oxidación y sustitución. A través de estas reacciones, los alcanos se pueden usar como combustible o en la síntesis de compuestos como el clorometano y el diclorometano [14].

Los alcanos son usados como combustibles líquidos y gaseosos, lubricantes, fluidos térmicos, solventes [17] [18], ceras [16] (velas [19], recubrimiento de frutas [20], etc.), asfalto, como materia prima en la síntesis de hidrógeno y otros compuestos químicos [15] y para la purificación y manufactura de medicamentos [21].

## **1.2. ESTRUCTURA Y PROPIEDADES DE LOS CLOROALCANOS**

Cuando por lo menos un hidrógeno de un alcano es remplazado por un halógeno se produce un haloalcano. A pesar de tener diversos usos industriales, muchos haloalcanos son tóxicos, como el 1,2-diclorofluoretano, uno de los agentes responsables del deterioramiento de la capa de ozono [14]. El poder modelar dichos compuestos correctamente puede ayudar en su tratamiento y eliminación para dar solución a los problemas ambientales que generan. Entre los haloalcanos más usados se encuentran los cloroalcanos, sustancias insolubles en agua, no

inflamables y altamente solubles en solventes orgánicos no polares. Entre sus aplicaciones se encuentran la elaboración de pinturas, cauchos, aerosoles y en la industria metalúrgica y textil [22].

Al comparar alcanos y haloalcanos de igual número de carbonos, se evidencia un mayor punto de ebullición en los últimos. La primera razón es que las fuerzas de London aumentan con el área superficial de la molécula, lo cual ocurre al sustituir un hidrógeno con un halógeno. La segunda razón es que los haloalcanos presentan interacciones dipolo-dipolo, generadas por las diferencias en electronegatividad del carbono y el halógeno, lo cual aumenta las interacciones intermoleculares [23].

Los cloroalcanos son ampliamente usados en la industria petrolera, textil, farmacéutica [24], papelera, de pinturas [25] y en la producción de solventes [26], líquidos iónicos [27], polímeros, antiespumantes, espesantes, materiales de construcción y fluidos de perforación [28] [29].

### **1.3. SISTEMAS BINARIOS ALCANO-CLOROALCANO**

Dado que los cloroalcanos se sintetizan mediante la cloración de los alcanos [30], existen mezclas de alcanos y cloroalcanos en la producción de los últimos. Adicionalmente, dado que los cloroalcanos no son solubles en agua, pero sí en sustancias no-polares, los alcanos se pueden usar para disolver los cloroalcanos. En estos casos también se crean mezclas de alcanos y cloroalcanos.

### **1.4. MODELAMIENTO DE PROPIEDADES TERMOFÍSICAS**

Para la predicción de propiedades termofísicas han sido usados métodos como: ecuaciones de estado, simulación Monte Carlo, modelos de coeficientes de actividad [31], redes neuronales y contribución de grupos, entre otros. En estudios anteriores se ha realizado modelamiento por redes neuronales al usar datos

experimentales de densidad e índices de refracción a diferentes temperaturas para la predicción de volumen de exceso de mezclas binarias de ácido propanoico, tolueno y agua [32]. Para estos sistemas también ha sido usadas ecuaciones de ajuste como la ecuación de Redlich-Kister junto con ecuaciones de índices de refracción para la obtención de propiedades de exceso [33]. Métodos como la simulación Monte Carlo y la ecuación Soft-SAFT han modelado sistemas con cadenas Lennard-Jones [34]. Para líquidos iónicos han sido usados métodos de contribución de grupos [35] [36] [37]. Para el caso de estudio: sistemas alcanocloroalcano se han realizado mediciones experimentales y obtenido propiedades de exceso mediante el modelo VTPR, siendo está una ecuación de estado que incluye parámetros basados en UNIFAC, obteniendo predicciones con una desviación relativa cuadrática media de 0,66% [12].

En la actualidad, y cada día con más frecuencia, se ha desarrollado software de sistemas de predicción mediante métodos de aprendizaje automático (“machine learning”) como redes neuronales, regresión multivariable, razonamiento de casos y reglas de inducción, entre otros. Una de las grandes ventajas de estos métodos es la obtención de modelos predictivos de alto grado de precisión [11]. En el 2017, Python fue calificado como el mejor lenguaje de programación [38]. Dicho lenguaje tiene herramientas como: **scikit-learn**, ampliamente utilizada en la minería de datos (“data mining”) y el análisis de datos, en aplicaciones como el reconocimiento de imágenes, predicciones de precios de acciones, optimización y selección de modelos, entre otras [39].

#### **1.4.1. Ecuaciones para las propiedades termofísicas de sustancias puras:**

La **densidad** de los líquidos (ecuación 1), depende en gran medida de la temperatura, y en menor medida de la presión [40]. El conocer esta propiedad es de gran importancia en la industria, como en el caso de los hidrocarburos a altas presiones y temperaturas de los yacimientos petroleros. La densidad a tales condiciones es fundamental para poder describir y caracterizar el flujo de

hidrocarburo. A pesar de su importancia, no se encuentra gran cantidad de datos de densidad a tales condiciones en la literatura [41].

La **expansibilidad isobárica** (ecuación 2) es la variación relativa del volumen de un cuerpo a presión constante con respecto a la temperatura. La expansibilidad isobárica de una sustancia es mayor en el estado gaseoso, seguido por el estado líquido y luego por el estado sólido [12]. La **compresibilidad isotérmica** (ecuación 3), hace referencia a la reducción relativa del volumen de una sustancia a temperatura constante con respecto a la presión. El poder calcular la expansibilidad isobárica y la compresibilidad isotérmica de sustancias permite dimensionar equipos industriales de manera correcta, al tener en cuenta cómo se comportan las mismas ante cambios de temperatura y presión.

La **presión interna** o presión de cohesión (ecuación 4) hace referencia al comportamiento que generan las fuerzas de cohesión entre las moléculas presentes en el líquido, creando una presión dentro del mismo [12]. La presión interna de una sustancia es gran importancia en la industria pues está directamente relacionada con la viscosidad, solubilidad y tensión superficial [42].

$$\rho_i = \frac{M_i}{V_i} \quad (1) \qquad \alpha_p = \frac{1}{V} \left( \frac{\partial V}{\partial T} \right)_p \quad (2)$$

$$\kappa_T = -\frac{1}{V} \left( \frac{\partial V}{\partial P} \right)_T \quad (3) \qquad p_i = T \left( \frac{\partial P}{\partial T} \right)_V - P = \frac{\alpha_p T}{\kappa_T} - P \quad (4)$$

#### 1.4.2. Ecuaciones para las propiedades termofísicas de sistemas binarios:

La **densidad de soluciones** (ecuación 5), se define como la relación entre el peso molecular promedio (ecuación 6) y el volumen molar de la mezcla. Las **propiedades de exceso** hacen referencia a la diferencia entre el valor real de la propiedad de una solución y el valor que tendría dicha propiedad si la solución fuera ideal, bajo

las mismas condiciones de presión, composición y temperatura [43]. Las ecuaciones de la 7 a la 14 hacen referencia a las propiedades de exceso calculadas en el presente trabajo: volumen, compresibilidad isobárica y expansibilidad isotérmica, con la fracción de volumen en su término ideal [44] y la presión interna, con su fracción de compresibilidad [12].

$$\rho_M = \frac{\bar{M}}{V_M} \quad (5) \quad \bar{M} = \sum_i M_i x_i \quad (6) \quad V^E = V_M - \sum x_i V_i \quad (7)$$

$$V_i = \frac{M_i}{\rho_i} \quad (8) \quad V_M = \frac{\bar{M}}{\rho_M} \quad (9) \quad \alpha_p^E = \alpha_p - \sum \phi_i \alpha_{p,i} \quad (10)$$

$$\phi_i = \frac{x_i V_i}{\sum_i x_i V_i} \quad (11) \quad \kappa_T^E = \kappa_T - \sum \phi_i \kappa_{T,i} \quad (12) \quad \psi_i = \frac{x_i \kappa_{T,i} V_i}{\sum_i x_i \kappa_{T,i} V_i} \quad (13)$$

$$p^E = p_i - \sum \psi_i p_{i,i} \quad (14)$$

## 2. OBJETIVOS

### 2.1. OBJETIVO GENERAL

Predecir propiedades termofísicas a partir de datos de presión, densidad y temperatura para sistemas binarios alcano-cloroalcano mediante la formulación de un modelo matemático basado en un método de contribución de grupos.

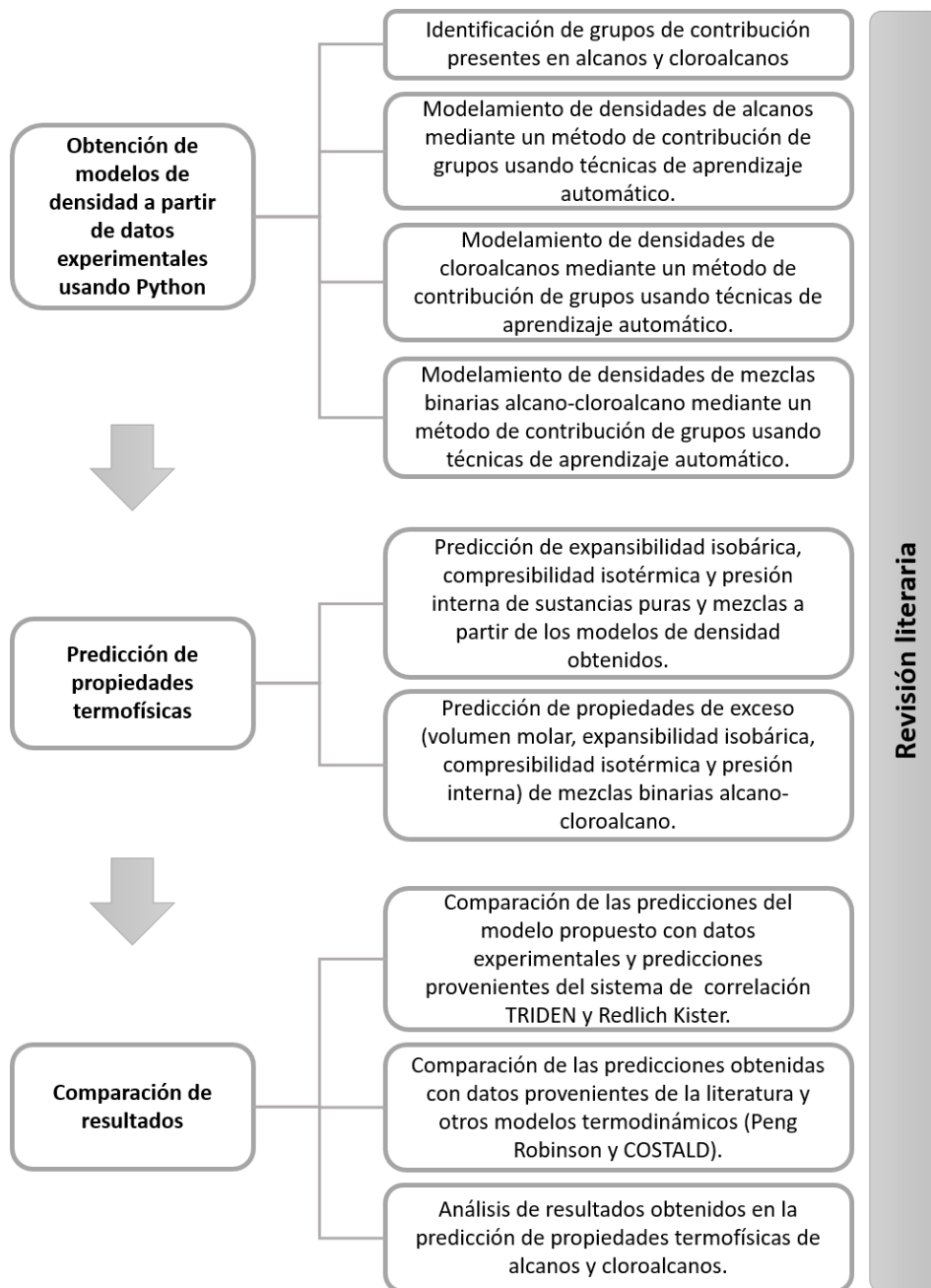
### 2.2. OBJETIVOS ESPECÍFICOS

- Formular ecuaciones que predigan la densidad de sustancias puras y sistemas binarios alcano-cloroalcano a diferentes temperaturas y presiones por un método de contribución de grupos a partir de datos experimentales.
- Estimar propiedades de exceso (volumen, expansibilidad isobárica, compresibilidad isotérmica y presión interna) mediante un modelamiento por un método de contribución de grupos usando datos experimentales de  $P$   $\rho$   $T$  para mezclas binarias alcano-cloroalcano.

### 3. METODOLOGÍA

El desarrollo de la presente investigación se resume en la figura 1.

Figura 1. Diagrama metodológico de la creación y validación de un modelo de contribución de grupos usando aprendizaje automático en Python.



### 3.1. IDENTIFICACIÓN DE GRUPOS DE CONTRIBUCIÓN PRESENTES EN LOS ALCANOS Y CLOROALCANOS

Para los modelos de densidad formulados, se tuvieron en cuenta los enlaces carbono-carbono y carbono-cloro según el tipo de carbono (primario, secundario, terciario y cuaternario) presentes en las moléculas (ver **Anexo B**). Los datos experimentales usados para el modelamiento se tomaron del trabajo doctoral: “Estudio pρT de alcanos, cloroalcanos y de sus mezclas binarias” [12]. Dicho proyecto suministró 7060 datos de densidades experimentales a diferentes condiciones de presión y temperatura para 6 alcanos, 6 cloroalcanos y 8 mezclas binarias.

### 3.2. OBTENCIÓN DE MODELOS DE DENSIDAD A PARTIR DE DATOS EXPERIMENTALES EN PYTHON

Los modelos de densidad para alcanos, cloroalcanos y mezclas binarias se realizaron mediante el lenguaje de programación de código abierto: Python, haciendo uso de la herramienta: Scikit-learn. Se realizó una manipulación y preparación de datos extensa con el fin de alimentarlos al programa y entrenar el modelo con el 5% de los datos experimentales disponibles en el trabajo doctoral de Guerrero Amaya [12]. El 95% de los datos restantes fueron usados para la validación del modelo. Dicha validación se realizó al calcular la desviación relativa cuadrática media, DRCM (ecuación 15), entre los datos experimentales y las densidades estimadas por el modelo, como medida de la exactitud del mismo.

$$DRCM = 100 \cdot \left( \frac{1}{N} \sum_{i=1}^N \left( \frac{\rho_{i,exp} - \rho_{i,cal}}{\rho_{i,exp}} \right)^2 \right)^{1/2} \quad (15)$$

Los modelos de densidades para alcanos, cloroalcanos y mezclas binarias alcano-cloroalcano se obtuvieron mediante regresiones lineales multivariantes, cuyas variables independientes hacen referencia a los grupos de contribución presentes en las moléculas, la presión, el cuadrado de la presión, la temperatura, la interacción entre presión y temperatura, las fracciones molares y el cuadrado de las fracciones molares (ecuaciones 16 a la 20). Los datos usados para el entrenamiento de los modelos fueron seleccionados entre 100 combinaciones aleatorias de datos experimentales (o 200 combinaciones para el modelo de las mezclas), cada una con el 5% de los datos disponibles, eligiendo los modelos con menor DRCM. Los datos usados para los modelamientos se encuentran registrados en el **Anexo A**.

**3.2.1. Modelo de densidad de alcanos:** Se contó con 659 datos experimentales de densidades de alcanos (n-hexano, n-heptano, 2-metilpentano, 3-metilpentano, 2,3-dimetilbutano y 2,2-dimetilbutano), en un rango de temperatura de 283,15-328,15 K y de presiones de 0,1-65 MPa [12]. El modelo (ecuación 16) fue entrenado con el 5% de los datos experimentales, equivalente a 32 datos.

$$\rho_{i,alcanos} = \sum_{j=1}^7 a_j n_j + bP + cP^2 + dT + ePT + f \quad (16)$$

El primer término del modelo representa el efecto que tiene la estructura de la molécula en la densidad mediante los grupos de contribución. El segundo término representa el efecto de la presión dado que un aumento en la presión comprime el fluido, ocasionando un aumento en la densidad. El tercer término se utiliza para crear la concavidad en la relación de densidad con presión, ya que no es una relación lineal. El cuarto término muestra el efecto de la temperatura. Se espera que el coeficiente  $d$  sea negativo, ya que al aumentar la temperatura se produce un aumento en el movimiento de las moléculas y en la separación intermolecular, disminuyendo la densidad. El quinto término representa la interacción entre la presión y la temperatura. Al observar los datos experimentales se evidencia que las

isotermas de densidad tienen diferente inclinación según la temperatura, para el mismo rango de presión, concluyendo que hay una interacción entre presión y temperatura. El último término representa el intercepto del modelo.

**3.2.2. Modelo de densidad de cloroalcanos:** Se contó con 593 datos experimentales de densidades de cloroalcanos (1-cloropropano, 2-cloropropano, 1-clorobutano, 2-clorobutano, iso-clorobutano y ter-clorobutano), en los mismos rangos de presión y temperatura de los alcanos [12]. El modelo fue entrenado con el 5% de los datos experimentales, equivalente a 29 datos. La estructura de este modelo (ecuación 17) es similar a la de los alcanos, pero difiere en los grupos de contribución según los enlaces entre sus átomos.

$$\rho_{i,cloroalcanos} = \sum_{j=1}^6 a_j n_j + bP + cP^2 + dT + ePT + f \quad (17)$$

**3.2.3. Modelo de densidad de mezclas binarias alcano-cloroalcano:** Se contó con 5808 datos experimentales para mezclas de alcanos (n-hexano o n-heptano) y cloroalcanos (1-clorobutano, 2-clorobutano, iso-clorobutano y ter-clorobutano) en los rangos de temperatura anteriormente enunciados a diferentes fracciones molares [12]. El modelo (ecuaciones 18 a la 20) fue entrenado con el 5% de los datos experimentales, equivalente a 290 datos. Los primeros tres términos del modelo representan los efectos de la concentración y las estructuras moleculares de las dos moléculas de la mezcla.

$$\rho_M = a + b(x_1 - x_2) + c(x_1 - x_2)^2 + dP + eP^2 + fT + gPT \quad (18)$$

$$a = a_0 + \frac{1}{2} \sum_{j=1}^6 a_j (n_{j,1} + n_{j,2}) \quad (19)$$

$$b = b_0 + \frac{1}{2} \sum_{j=1}^6 a_j (n_{j,1} - n_{j,2}) \quad (20)$$

### 3.3. PREDICCIÓN DE PROPIEDADES TERMOFÍSICAS

Al reemplazar los modelos de densidad en las ecuaciones 8 y 9 se obtienen expresiones para los volúmenes molares de sustancias puras y mezclas. Al derivar estas expresiones con respecto a temperatura y presión (ecuaciones 2 a la 4), se obtienen la expansibilidad isobárica (ecuación 21), la compresibilidad isotérmica (ecuación 22) y la presión interna (ecuación 23). A partir de las ecuaciones 21 a la 25, se calculan las propiedades de exceso usando las ecuaciones 7, 10, 12 y 14.

$$\alpha_{p,i} = \frac{-d - eP}{\rho} \quad (21) \quad \kappa_{T,i} = \frac{b + 2cP + eT}{\rho_i} \quad (22) \quad p_i = \frac{T(d + eP)}{-(b + 2cP + eT)} - P \quad (23)$$

### 3.4. COMPARACIÓN DE RESULTADOS

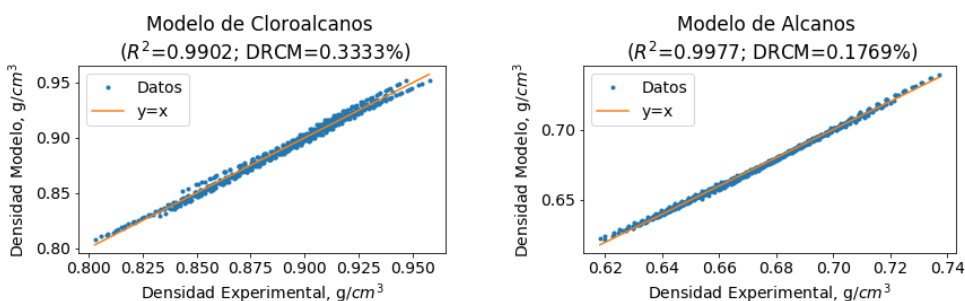
Se compararon las propiedades termofísicas calculadas mediante los modelos obtenidos en el presente trabajo con los valores provenientes de datos experimentales y de otros modelos de estimación. Dichos modelos fueron la ecuación de estado Peng-Robinson, y las correlaciones TRIDEN, Redlich-Kister, y COSTALD (Densidad Líquida de Estados Correspondientes) [10]. La medida de comparación que se utilizó fue la DRCM.

## 4. RESULTADOS Y ANÁLISIS

### 4.1. MODELO DE DENSIDAD DE SUSTANCIAS PURAS PARA ALCANOS

El coeficiente de determinación del modelo de densidad de alcanos obtenido fue de  $R^2 = 0,9977$ . Se validó el modelo con todos los datos experimentales de alcanos, obteniendo una DRCM de 0,1769%. En la Figura 2 se evidencia el buen ajuste entre todos los datos experimentales para alcanos y el modelo obtenido para densidades de alcanos.

Figura 2. Diagrama de dispersión con los datos experimentales (·) y el modelo obtenido (-) para la densidad de cloroalcanos (izda.) y alcanos (dcha.).



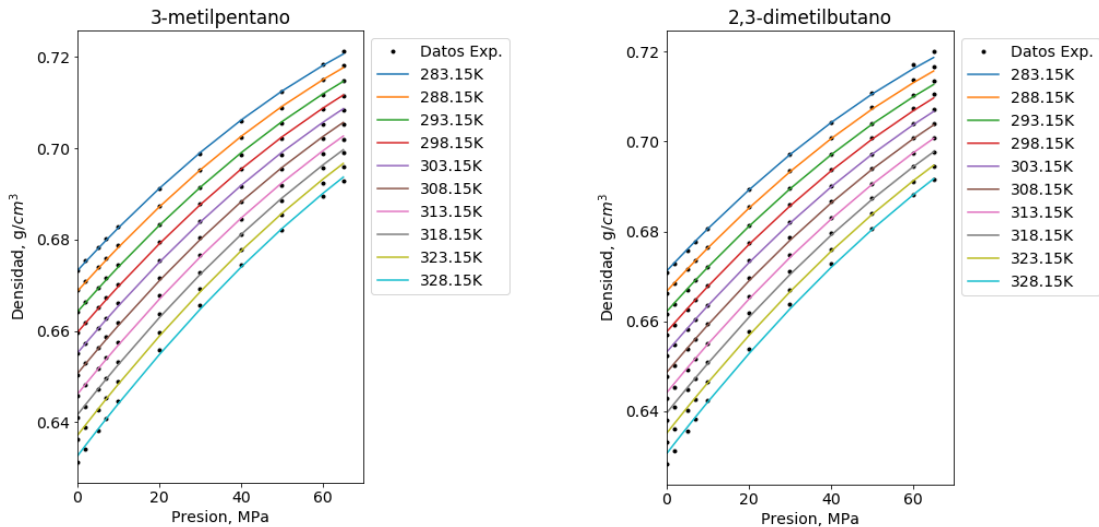
Los coeficientes calculados para el modelo de densidad de alcanos se encuentran registrados en la Tabla 2. El signo positivo de los coeficientes  $a_i$  indica que la densidad de los alcanos aumenta al incrementar la cantidad de enlaces carbono-carbono pues aumentan las fuerzas de London. Los signos negativos de los coeficientes  $b$  y  $c$  verifican que hay una concavidad negativa en la densidad con respecto a la presión. El signo negativo del coeficiente  $d$  concuerda con el hecho de que, a mayor temperatura, disminuye la densidad.

Tabla 2. Coeficientes obtenidos para el modelo de densidad de alcanos.

Coeficiente	Valor	Coeficiente	Valor	Coeficiente	Valor
$a_1(C_1 - C_2)$	0,021907	$a_5(C_2 - C_3)$	0,028580	$c$	-3,7738e-6
$a_2(C_1 - C_3)$	0,016966	$a_6(C_2 - C_4)$	0,008361	$d$	-0,000903
$a_3(C_1 - C_4)$	0,025082	$a_7(C_3 - C_3)$	0,048077	$e$	4,6657e-6
$a_4(C_2 - C_2)$	0,023159	$b$	-0.000345	$f$	0,810966

La Figura 3 muestra la variación de la densidad estimada por el modelo con respecto a la presión, a diferentes temperaturas, y los datos de densidad experimental a las mismas condiciones para el 3-metilpentano y 2,3-dimetilbutano. Se observa un buen ajuste del modelo a los datos experimentales. En el **Anexo C** se encuentran registradas las gráficas de densidad de los demás alcanos estudiados.

Figura 3. Variación de la densidad experimental (·) y modelada (-) con respecto a la presión, a diferentes temperaturas para el 3-metilpentano (izda.) y 2,3-dimetilbutano (dcha.).



## 4.2. MODELO DE DENSIDAD DE SUSTANCIAS PURAS PARA CLOROALCANOS

El coeficiente de determinación del modelo de densidad de cloroalcanos fue de  $R^2 = 0,9902$ . Con todos los datos experimentales de cloroalcanos se realizó la validación respectiva, obteniendo una DRCM de 0,3333%. En la Figura 2 también se evidencia un buen ajuste entre los datos experimentales de cloroalcanos y el modelo obtenido para la densidad de los mismos.

Los coeficientes calculados para el modelo de densidad de cloroalcanos se encuentran registrados en la Tabla 3. Se evidencia que cuando las moléculas tienen el cloro enlazado con un carbono primario, el coeficiente es positivo, es decir, aumenta la densidad. Esto concuerda con el hecho de que el tener el cloro en uno de los extremos de la cadena aumenta la linealidad de la misma, aumentando las fuerzas de London. De la misma manera, el tener el cloro enlazado con un carbono secundario o terciario disminuye la linealidad de la cadena de hidrocarburo. El intercepto de los cloroalcanos es superior al de los alcanos, siguiendo el mismo comportamiento de la densidad, debido a sus propiedades polares que fortalecen las interacciones entre moléculas.

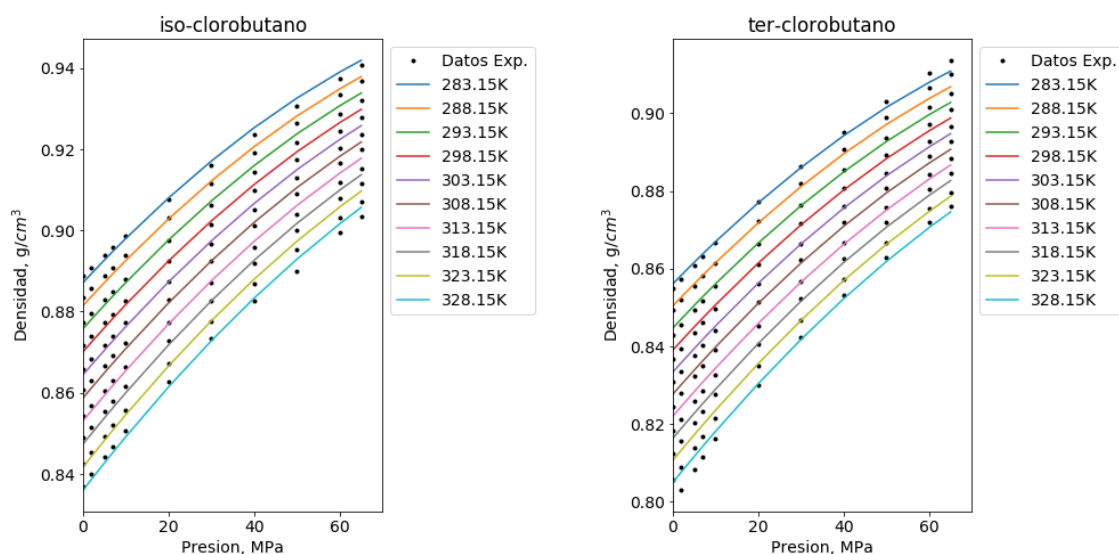
Tabla 3. Coeficientes obtenidos para el modelo de densidad de cloroalcanos.

Coeficiente	Valor	Coeficiente	Valor	Coeficiente	Valor
$a_1(C_1 - C_2)$	0,001570	$a_5(C_2 - Cl)$	-0,001048	$d$	-0,001137
$a_2(C_1 - C_3)$	-0,002354	$a_6(C_3 - Cl)$	-0,014989	$e$	5,0958e-6
$a_3(C_2 - C_2)$	0,000111	$b$	-0,000309	$f$	1,199961
$a_4(C_1 - Cl)$	0,016037	$c$	-4,4746e-6	-	-

La Figura 4 muestra la variación de la densidad estimada por el modelo con respecto a la presión, a diferentes temperaturas, y los datos de densidad experimental a las mismas condiciones para el iso-clorobutano y ter-clorobutano. Se evidencia un

mayor ajuste entre los datos experimentales y el modelo en los rangos de presiones de 20 a 50 MPa para el ter-clorobutano y 0.1 a 30 MPa para el iso-clorobutano. En el **Anexo C** se encuentran registradas las gráficas de densidad de los demás cloroalcanos estudiados.

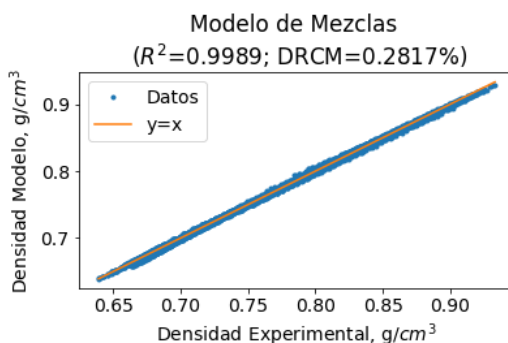
Figura 4. Variación de la densidad experimental (·) y modelada (-) con respecto a la presión, a diferentes temperaturas para el iso-clorobutano (izda.) y ter-clorobutano (dcha.).



#### 4.3. MODELO DE DENSIDAD DE MEZCLAS PARA SISTEMAS BINARIOS ALCANO-CLOROALCANO

El coeficiente de determinación del modelo de mezclas binarias fue de  $R^2 = 0,9989$ . Dicho modelo se validó con todos los datos experimentales de mezclas, obteniendo una DRCM de 0,2817%. En la Figura 5 se evidencia un excelente ajuste entre los datos experimentales y el modelo obtenido para la densidad de las mezclas con los cuales se realizó el modelo.

Figura 5. Diagrama de dispersión con los datos experimentales (·) y el modelo obtenido (-) para la densidad de mezclas binarias alcano-cloroalcano.



Los coeficientes calculados para el modelo de densidad de mezclas binarias alcano-cloroalcano se encuentran en la Tabla 4. Al analizar los signos de los coeficientes del modelo, se observa que todos los enlaces carbono-cloro tienen signo positivo. Esto se debe a que la presencia de cloros en la mezcla aumenta la densidad de la misma al crear dipolos que aumentan las fuerzas intermoleculares. Además, el enlace carbono primario-cloro es el grupo de contribución que más influye en el aumento la densidad, tal como se observa en el modelo de cloroalcanos.

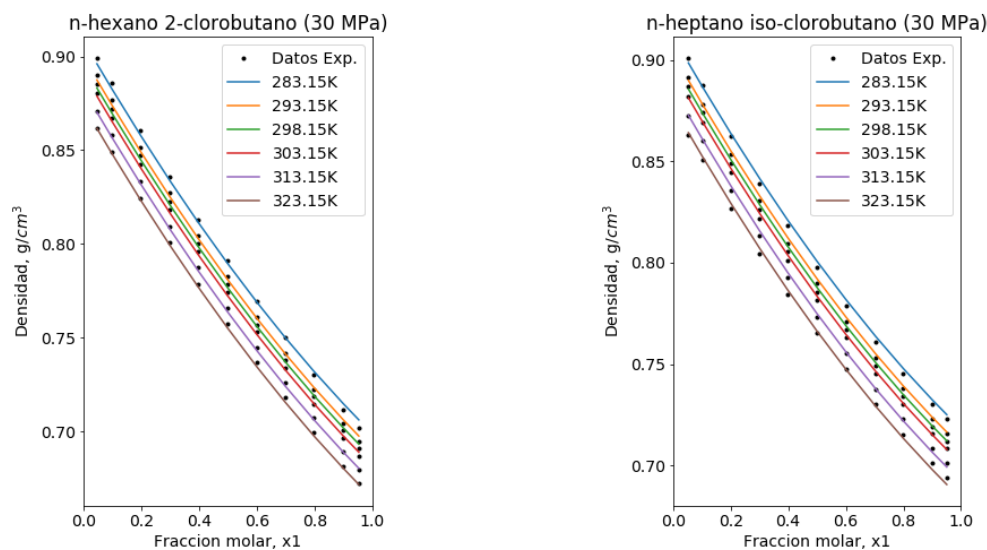
Tabla 4. Coeficientes para el modelo de mezclas binarias alcano-cloroalcano

Coefficiente	Valor	Coefficiente	Valor	Coefficiente	Valor
$a_0$	0,947278	$a_5(C_2 - Cl)$	0,077970	$e$	-3,5913e-6
$a_1(C_1 - C_2)$	-0,001768	$a_6(C_3 - Cl)$	0,060036	$f$	-0,001011
$a_2(C_1 - C_3)$	0,002652	$b_0$	-0,084885	$g$	4,9877e-6
$a_3(C_2 - C_2)$	0,019148	$c$	0,014277		
$a_4(C_1 - Cl)$	0,088873	$d$	-0,000432		

La Figura 6 muestra la variación de la densidad estimada por el modelo de mezclas con respecto a la fracción molar de alcano, a diferentes temperaturas y 30 MPa de presión, y los datos de densidad experimental a las mismas condiciones. En general

se observa un mayor ajuste entre los datos experimentales y el modelo para fracciones molares bajas, es decir, para mezclas con mayor concentración del cloroalcano.

Figura 6. Variación de la densidad experimental (·) y modelada (-) con respecto a la fracción molar, a diferentes temperaturas y 30 MPa para las mezclas n-he y 2-clorobutano (izda.) y n-heptano e iso-clorobutano (dcha.).



#### 4.4. PREDICCIÓN DE PROPIEDADES TERMOFÍSICAS DE ALCANOS Y CLOROALCANOS

Los modelos obtenidos se utilizaron para predecir la expansibilidad, compresibilidad y presión interna de los compuestos. En la Tabla 5 se registran las desviaciones estándar y DRCM de propiedades termofísicas con respecto a valores de las mismas propiedades provenientes del sistema de correlación TRIDEN [12]. A pesar de que las magnitudes de la expansibilidad isobárica y compresibilidad isotérmica son pequeñas (orden de magnitud:  $10^{-3} \text{ K}^{-1}$  y  $10^{-3} \text{ MPa}$ ), los modelos permiten predecir la expansibilidad isobárica con un error del 2,2-11,2% y la compresibilidad isotérmica con un error del 5,3-13,5%. En el caso de la presión interna, los errores

están entre 3,8-13,2%. El error en la presión interna es un reflejo de los errores en la expansibilidad y la compresibilidad, ya que estos se utilizan para calcularla.

Tabla 5. Desviación estándar y DRCM de expansibilidad isobárica, compresibilidad isotérmica y presión interna de alcanos y cloroalcanos.

Líquido	$\sigma(\alpha_p)$ ( $\text{kK}^{-1}$ )	DRCM %	$\sigma(\kappa_T)$ ( $\text{MPa}^{-1}$ )	DRCM %	$\sigma(p_i)$ (MPa)	DRCM %
1-clorobutano	0.0338	2.9358	0.1240	11.2516	0.0271	8.3405
1-cloropropano	0.1209	9.1042	0.0627	5.3145	0.0270	8.1441
2,2-dimetilbutano	0.0590	4.0016	0.2936	13.4987	0.0303	13.2454
2,3-dimetilbutano	0.0433	3.3163	0.1806	8.8397	0.0190	7.9375
2-clorobutano	0.0316	2.7214	0.0698	6.1179	0.0117	3.7852
2-cloropropano	0.1621	11.1504	0.1907	12.9965	0.0127	3.9861
2-metilpentano	0.0464	3.3996	0.1930	9.2527	0.0204	8.4912
3-metilpentano	0.0407	3.2465	0.1330	6.9586	0.0140	5.6520
iso-clorobutano	0.0251	2.2224	0.0743	6.6412	0.0163	5.1848
n-heptano	0.0711	5.9927	0.1058	8.9654	0.0153	5.6609
n-hexano	0.0383	3.0953	0.1163	6.3788	0.0144	5.7053
ter-clorobutano	0.0698	4.8893	0.1933	11.0758	0.0224	8.3409

#### 4.5. PREDICCIÓN DE PROPIEDADES DE EXCESO PARA SISTEMAS BINARIOS ALCANO-CLOROALCANO

Las propiedades de exceso también se estimaron a partir de los modelos de las mezclas y las sustancias puras. En las Tablas 6 y 7 se muestran las desviaciones estándar con respecto a los valores reportados por Guerrero Amaya [12], quien estimó propiedades de exceso mediante el sistema de correlación TRIDEN y Redlich-Kister. Cabe resaltar que las propiedades de exceso son bastante pequeñas, indicando que las mezclas estudiadas tienen un comportamiento cercano a la idealidad.

Tabla 6. Desviación estándar de propiedades de exceso (volumen, expansibilidad isobárica y compresibilidad isotérmica) para mezclas con n-hexano

<b>Sistemas Binarios (n-hexano + cloroalcano)</b>	$\sigma(V^E)$ ( $cm^3/mol$ )	$\sigma(\alpha_p^E)$ ( $kK^{-1}$ )	$\sigma(\kappa_T^E)$ ( $MPa^{-1}$ )
1-clorobutano	0,1975	0,0843	0,0755
2-clorobutano	0,2828	0,0887	0,0722
iso-clorobutano	0,2354	0,0879	0,0742
ter-clorobutano	0,4627	0,0896	0,0672

Tabla 7. Desviación estándar de propiedades de exceso (volumen, expansibilidad isobárica y compresibilidad isotérmica) para mezclas con n-heptano

<b>Sistemas Binarios (n-heptano + cloroalcano)</b>	$\sigma(V^E)$ ( $cm^3/mol$ )	$\sigma(\alpha_p^E)$ ( $kK^{-1}$ )	$\sigma(\kappa_T^E)$ ( $MPa^{-1}$ )
1-clorobutano	0,4460	0,0875	0,0707
2-clorobutano	0,3509	0,0780	0,0767
iso-clorobutano	0,4152	0,0861	0,0758
ter-clorobutano	0,1318	0,0867	0,0691

#### 4.6. VALIDACIÓN DE LOS MODELOS

Se obtuvieron datos adicionales en literatura de alcanos, cloroalcanos y mezclas binarias diferentes a los usados en el entrenamiento de los modelos. Se utilizaron los modelos de sustancias puras y mezclas obtenidos en el presente trabajo para predecir las densidades de los nuevos compuestos y mezclas. Adicionalmente, se compararon las densidades obtenidas con las obtenidas por la ecuación de estado de Peng-Robinson y el modelo COSTALD [10] para los nuevos compuestos y mezclas. Los modelos de Peng-Robinson y COSTALD se implementaron directamente en Python usando la librería Thermo [45].

#### 4.6.1. Validación del modelo de densidad de alcanos

El modelo de densidad de alcanos se probó con datos de literatura para n-pentano, iso-pentano, n-octano, iso-octano, ciclooctano, n-decano, n-nonano, n-undecano, n-dodecano y n-tridecano [41] [46]. En las Figuras 7 y 8, se presentan los resultados para el n-pentano a diferentes temperaturas y presiones, y para el iso-pentano, n-nonano, n-undecano y n-dodecano a 20°C y 1 atm. Para los otros compuestos véase el **Anexo D**.

Se evidencia que el modelo se puede extrapolar a alcanos diferentes al heptano y al hexano y sus isómeros (alcanos usados para entrenar el modelo) y realizar predicciones de densidad con DRCM inferiores a 1.8% para cadenas de hasta 9 carbonos. También se demuestra que el modelo se puede extrapolar a temperaturas y presiones superiores a las usadas para crear el modelo. Sin embargo, existen ciertas condiciones en las que se presentan errores significativos. La primera es cuando las presiones superan los 120 MPa (casi el doble de la presión máxima usada para el modelo) y las temperaturas superan los 430 K (100°C por encima de la temperatura máxima usada para el modelo). La segunda es cuando hay alcanos con más de dos ramificaciones en su estructura molecular. La tercera es cuando hay alcanos cíclicos. La cuarta es cuando hay alcanos con más de 12 carbonos.

Se observa que, en general, el modelo propuesto tiene un mejor ajuste a los datos experimentales que el de Peng-Robinson. A pesar de que el modelo COSTALD se ajusta mejor a los datos, éste tiene la limitación de no funcionar a temperaturas superiores a la temperatura crítica de la molécula. Por ello, el modelo COSTLAND, no calculó la densidad del n-pentano a 520.45 K. Por el contrario, el modelo de alcanos dio resultados satisfactorios a esta temperatura hasta ~80 MPa.

Figura 7. Comparación de la variación de la densidad del n-pentano experimental con la estimada por (de izda. a dcha.) el modelo de propuesto, Peng Robinson y COSTALD.

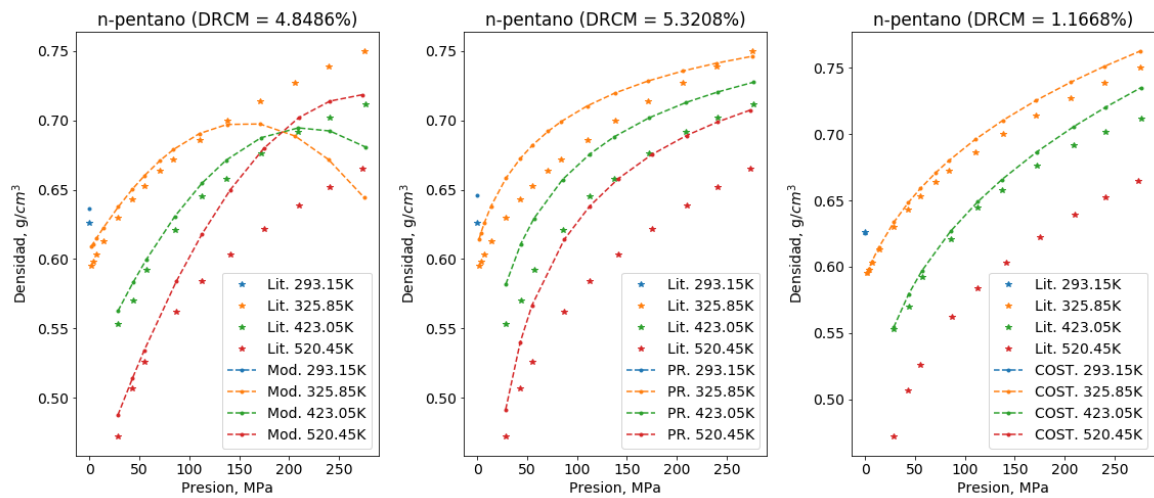
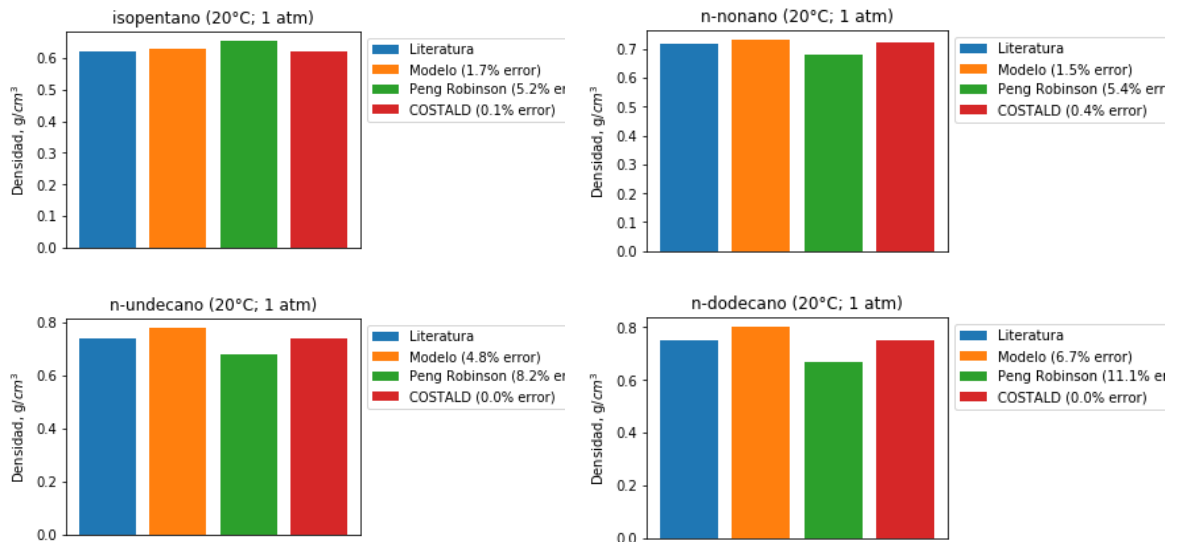


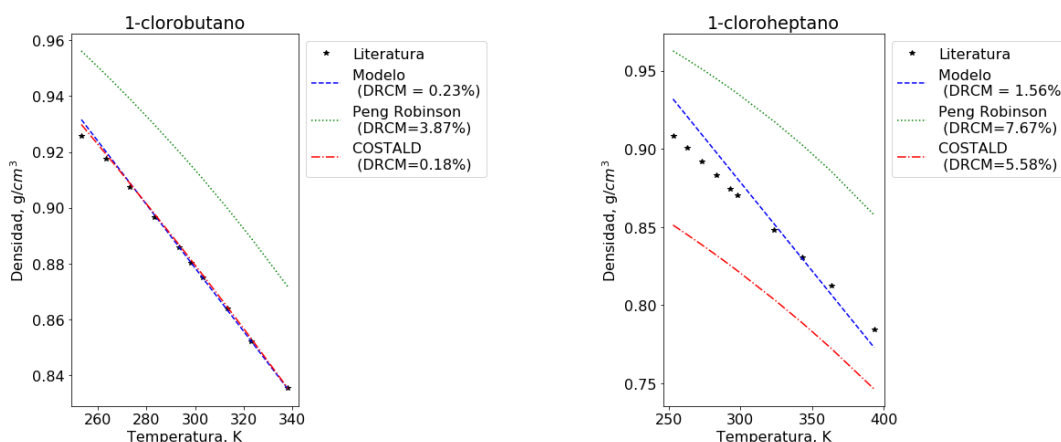
Figura 8. Comparación de las densidades del isopentano (sup. izda.), nonano (sup. dcha.), undecano (inf. izda.) y dodecano (inf. dcha.) con el modelo de alcanos, Peng-Robinson y COSTALD.



#### 4.6.2. Validación del modelo de densidad de cloroalcanos

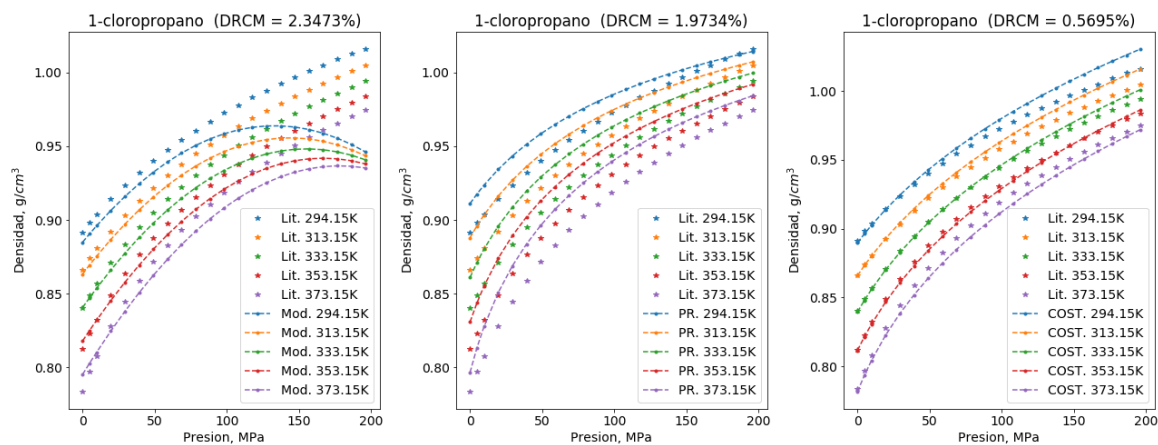
El modelo de cloroalcanos se verificó con los siguientes líquidos: 1-cloropropano, 1-clorobutano, 1-cloropentano, 1-cloroheptano, 1-clorooctano, 1-clorodecano, 1-clorotetradecano, 1,4-diclorobutano y 1,6-diclorohexano [27] [25]. Para el 1-cloropropano y el 1-clorobutano, el modelo se validó a temperaturas en un rango superior al de los datos experimentales con los que se creó el modelo. En las Figuras 9 y 10, se muestran los resultados para el 1-clorobutano, 1-cloroheptano y 1-cloropropano. Para los otros compuestos véase el **Anexo D**.

Figura 9. Comparación de la densidad del 1-clorobutano (izda.) y del 1-cloroheptano (dcha.) con el modelo de cloroalcanos, Peng-Robinson y COSTALD.



En general, el modelo se ajusta bastante bien para los 1-cloroalcanos con una DRCM inferior a 2,5%. En todos los casos, el modelo supera a Peng-Robinson y en algunos como el 1-cloroheptano y 1-clorooctano, el modelo supera a COSTALD. Nuevamente, se observa la limitación del modelo de los cloroalcanos al llegar a presiones superiores a 120 MPa. Adicionalmente, el modelo está limitado a cloroalcanos con un solo átomo de cloro. Estas limitaciones se deben a los datos experimentales con los que se entrenó el modelo y se podrían superar al entrenar el modelo de aprendizaje automático con datos adicionales de literatura.

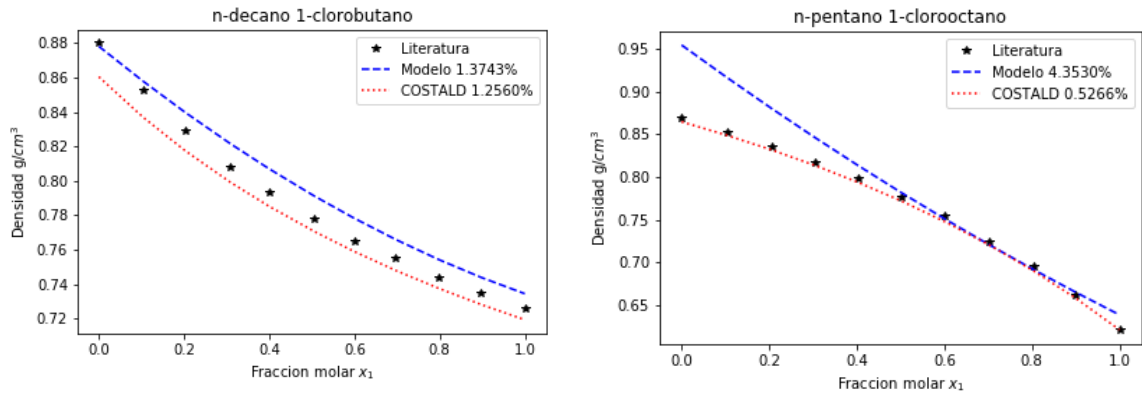
Figura 10. Comparación de la densidad experimental del 1-cloropropano con la densidad estimada por (de izda a dcha.) el modelo, Peng-Robinson y COSTALD.



#### 4.6.3. Validación del modelo de densidad de mezclas binarias

El modelo de las mezclas se validó para las mezclas binarias n-pentano/1-clorobutano, n-decano/1-clorobutano, n-pentano/1-clorooctano y n-decano/1-clorooctano a condiciones ambiente [47]. En la Figura 11 se muestran los resultados para dos de las mezclas. Las otras dos están en el **Anexo D**. Para las mezclas de 1-clorobutano, el modelo da resultados comparables a COSTALD, aun cuando el alcano de la mezcla tenga 10 carbonos (el modelo sólo se entrenó con 6 y 7 carbonos). Para cloroalcanos con más de 4 carbonos, el modelo COSTALD supera el modelo de mezclas, especialmente a bajas concentraciones del alcano. Nuevamente, esta limitación se podría superar al entrenar el modelo con datos nuevos que comprendan cloroalcanos con más de 4 carbonos.

Figura 11. Comparación de la densidad de las mezclas binarias n-decano y 1-clorobutano (izda.) y n-pentano y 1-clorooctano (dcha.) con el modelo de mezclas, Peng-Robinson y COSTALD.



Nota: En el **Anexo E**, se registró el código de programación usado para el modelamiento y comparación de los resultados obtenidos.

## 5. CONCLUSIONES

- Los modelos de densidad propuestos para alcanos, cloroalcanos, y sus mezclas binarias presentaron desviaciones relativas cuadráticas medias menores de 0,34% para los compuestos con los que se entrenó el modelo y menores de 2,4% a condiciones de presión y temperatura más elevadas (120 MPa y 400K), con respecto a las presiones y temperaturas de los datos experimentales usados por los modelos. Esto indica un alto grado de confiabilidad en las predicciones de densidad de los modelos.
- Al comparar los modelos propuestos en el presente trabajo de grado con otros métodos de predicción de densidad, se evidencian grandes ventajas al ser sencillos, requerir pocos datos para su entrenamiento y tener la habilidad de permitir extrapolaciones a diferentes condiciones de presión, temperatura y variedades de compuestos. Además, permiten estimar las propiedades termofísicas de los compuestos puros y las propiedades de exceso mediante ecuaciones sencillas en términos de los parámetros del modelo y las condiciones de presión y temperatura.
- A pesar de que el modelamiento realizado es una estimación estadística, los parámetros obtenidos dan información acerca de la influencia de la estructura molecular y las condiciones de operación en la densidad de líquidos y sistemas binarios, acorde a la información termodinámica conocida.

## RECOMENDACIONES

- Se sugiere entrenar los modelos obtenidos mediante el aprendizaje automático con los datos obtenidos de la literatura. Dicho entrenamiento, permitiría superar las limitaciones observadas durante la validación de los modelos. De esta manera, el modelo se podría expandir a cloroalcanos con más de un cloro, sustancias y mezclas con más carbonos y sistemas a mayores temperaturas y presiones.
- Los principios de modelamiento utilizados en el presente trabajo se pueden expandir para incluir otros sistemas (alquenos, aromáticos, cetonas, ácidos carboxílicos, aminas y alcoholes, entre otros).
- Se recomienda probar las limitaciones de los modelos con mezclas ternarias y sistemas menos ideales. Si se observan pérdidas significativas en la precisión, se recomienda entrenar el modelo con algunos datos de dichos sistemas.

## BIBLIOGRAFÍA

- [1] ASSAEL, Marc J. The Importance of Thermophysical Properties in Optimum Design and Energy Saving. En: Energy and Environment. Tokio: Springer, 2001. p.162-178. ISBN 978-4-431-68327-8.
- [2] GLEN, Norman. The Importance of Understanding Thermophysical Properties Data and Effects on Engineering Calculations. NEL - Flow Measurement Services, 2014. p.14.
- [3] HERGES, Rainer y KLOSE, Kirsten. Alkanes: Structure and Constitution. [En línea]. Wiley Information Services GmbH – ChemgaPedia. (Recuperado en 5 marzo 2018.) Disponible en [http://www.chemgapedia.de/vsengine/vlu/vsc/en/ch/2/vlu/alkane/alkane\\_struktur.vlu.html](http://www.chemgapedia.de/vsengine/vlu/vsc/en/ch/2/vlu/alkane/alkane_struktur.vlu.html).
- [4] NATH, Sukhendu, *et al.* Steady-State and Time-Resolved Studies on Photoinduced Interaction of Phenothiazine and 10-Methylphenothiazine with Chloroalkanes. En: Journal of Physical Chemistry A. Junio, 1998, vol.102 no.29., p. 5822-5830.
- [5] GONZÁLEZ-PRIOR, J., *et al.* Mecanismo de oxidación de cloroalcanos sobre un catalizador Co<sub>3</sub>O<sub>4</sub>/SBA. En: CICat 2016 Congreso Iberoamericano de Catálisis (25, 18-23, septiembre: Montevideo, Uruguay). CICat, 2016.
- [6] SCHMID, Bastian, SCHEDEMANN, Andre y GMEHLING, Jürgen. Extension of the VTPR Group Contribution Equation of State: Group Interaction

Parameters for Additional 192 Group Combinations and Typical Results. En: *Industrial & Engineering Chemistry Research*. Febrero, 2014, vol.53 no.8., p. 3393-3405.

- [7] TWU, Chornng H., *et al.* A cubic equation of state with a new alpha function and a new mixing rule. En: *Fluid Phase Equilibria*. Diciembre, 1991, vol.69, p. 33-50.
- [8] AALTO, Mika, *et al.* An improved correlation for compressed liquid densities of hydrocarbons. Part 1. Pure compounds. En: *Fluid Phase Equilibria*. Enero, 1996, vol.114 no.1-2., p. 1-19.
- [9] AALTO, Mika, *et al.* An improved correlation for compressed liquid densities of hydrocarbons. Part 2. Mixtures. En: *Fluid Phase Equilibria*. Enero, 1996, vol.114 no.1-2., p. 21-35.
- [10] THOMSON, G. H., BROBST, K. R. y HANKINSON, R. W. An improved correlation for densities of compressed liquids and liquid mixtures. En: *AIChE Journal*. Julio, 1982, vol.28 no.4., p. 671-676.
- [11] MAIR, Carolyn, *et al.* An investigation of machine learning based prediction systems. En: *Journal of Systems and Software*. Julio, 2000, vol53 no.1., p. 23-29.
- [12] GUERRERO AMAYA, Hernando. Estudio ppT de alcanos, cloroalcanos y de sus mezclas binarias. Tesis de Doctor en Ciencias Químicas. Aragón: Universidad de Zaragoza. Facultad de Ciencias, 2012.

- [13] Smith, Janice Gorzynski. Alkanes. En: Organic Chemistry. 2 ed. Nueva Deli: Tata McGraw Hill, 2009. p. 114-159.
- [14] ATKINS, Peter y JONES, Loretta. Properties of Alkanes. En: Chemical Principles, The Quest for Insight. 4 ed. Nueva York: W. H. Freeman and Company, 2008. p. 737-756.
- [15] GARY, James H. y HANDWERK, Glenn E. Refinery Products. En: Petroleum Refining. Nueva York: Marcel Dekker, Inc, 1994. p. 2.
- [16] VAN DER LAAN, Gerard Pieter. Kinetics, selectivity and scale up of the Fischer-Tropsch synthesis. Tesis de Doctor en Matemáticas y Ciencias Naturales. Groningen: University of Groningen. Facultad de Matemáticas y Ciencias Naturales, 1999, 251p.
- [17] POE, Sarah L., *et al.* Solving the Clogging Problem: Precipitate-Forming Reactions in Flow. En: Angewandte Chemie International Edition. Febrero, 2006, vol.45 no.10., p. 1544-1548.
- [18] SICAIRE, Anne-Gaëlle, *et al.* Alternative Bio-Based Solvents for Extraction of Fat and Oils: Solubility Prediction, Global Yield, Extraction Kinetics, Chemical Composition and Cost of Manufacturing. En: International Journal of Molecular Sciences. Abril, 2015, vol.16 no.4., p. 8430-8453.
- [19] STANDARD OIL DEVELOPMENT CO. Paraffin wax candle. Inventor: KNOX, William T. Fecha de solicitud: 29 julio 1949. Estados Unidos de América. US 2697926. 29 diciembre 1954.

- [20] BALDWIN, E. A., *et al.* Use of lipids in edible coatings for food products. En: Food Technology. Junio, 1997, p. 56-64.
- [21] OOSTDIJK, John, KUIPERS, Johan y GANNI, Ramaprasad. Agilent J&W CP-Select 624 Hexane for Best Separation of Solvents and Hexane Isomers. En: Agilent Technologies. Noviembre, 2015.
- [22] PRTR ESPAÑA. Cloroalcanos (C10-C13). [En línea]. Ministerio de Agricultura y Pesca, Alimentación y Medio Ambiente. (Recuperado en 26 abril 2018.) Disponible en <http://www.prtr-es.es/Cloroalcanos-C10C13,15618,11,2007.html>.
- [23] ATKINS, Peter y JONES, Loretta. Dipole-Dipole Forces. En: Chemical Principles, The Quest for Insight. 4 ed. Nueva York: W. H. Freeman and Company, 2008. p. 179-180.
- [24] NATIONAL CENTER FOR BIOTECHNOLOGY INFORMATION. Methylene Chloride. [En línea]. PubChem. 16 septiembre 2004. (Recuperado en 25 abril 2018). Disponible en <https://pubchem.ncbi.nlm.nih.gov/compound/dichloromethane>.
- [25] BLOTNIKOV, Mikhail F., NERUCHEV, Yuriy A. y RYSHKOVA, Olga S. Density of Some 1-Chloroalkanes within the Temperature Range from (253.15 to 423.15) K. En: Journal of chemical & Engineering Data. Octubre, 2007, vol.52 no.6., p. 2514-2516.

- [26] ICIS. Methylene chloride prices, markets & analysis. [En línea]. Chemicals. (Recuperado en 25 abril 2018.). Disponible en <https://www.icis.com/chemicals/methylene-chloride>.
- [27] MELENT'EV, Vyacheslav V. y POSTNIKOV, Eugene B. Speed of Sound and Density of 1-Chloropropane in the Range of Temperatures 180-373 K and Pressures up to 196.1 MPa. En: Journal of Chemical & Engineering Data. Agosto, 2017, vol.62 no.10., p. 3409-3413.
- [28] OXYCHEM TECHNICAL SERVICES DEPARTMENT. Methyl Chloride End Use Applications. [En línea]. Dallas: Occidental Chemical Corporation. 2015. (Recuperado en 25 abril 2018.) Disponible en <http://www.oxy.com/OurBusinesses/Chemicals/Products/Documents/methylchloride/510-104%20Methyl%20Chloride%20Applications.pdf>.
- [29] GARCÍA, J., *et al.* UNIFAC calculation of thermodynamic properties of binary 1-chloroalkane + alkane and  $\alpha,\omega$ -dichloroalkane + alkane mixtures: Comparison with Nitta-Chao and DISQUAC predictions. En: Canadian Journal of Chemistry. Mayo, 2003, vol.81 no.5., p. 394-405.
- [30] Smith, Janice Gorzynski. Radical Reactions. En: Organic Chemistry. 2 ed. Nueva Deli: Tata McGraw Hill, 2009. p. 536-569.
- [31] CONTI, G, *et al.* Excess thermodynamic properties of asymmetric multicomponent mixtures: Predictive models and microscopic insight for the

system ethanol + tetrahydrofuran + cyclohexane at 25 C. En: Pure & Applied Chemistry. Septiembre, 1995, vol.67 no.11., p. 1849-1854.

- [32] LISA, Gabriela, CURTEANU, Silvia y LISA, Catalin. Prediction of Excess Thermodynamic Properties from experimental refractive index of binary mixtures 2. Artificial Neural Network Modelling. En: Revue Roumaine de Chimie. Septiembre, 2008, vol.53 no.9., p. 859-867.
- [33] LISA, Gabriela y LISA, Catalin. Prediction of excess thermodynamic properties from experimental refractive index of binary mixtures 1. Water- Propionic acid mixtures at 290.15, 300.15 and 310.15 K. En: Revue Roumaine de Chimie. Julio, 2007, vol.52 no.7., p. 647-653.
- [34] BLAS, Felipe J. Excess Thermodynamic Properties of Chainlike Mixtures. 1. Predictions from the Soft-SAFT Equation of State and Molecular Simulation. En: Journal of Physical Chemistry B. Septiembre, 2000, vol.104 no.39., p. 9239-9248.
- [35] ZHAO, Nan, *et al.* Group Contribution Method for Evaluation of Volumetric Properties of Ionic Liquids Using Experimental Data Recommended by Mathematical Gnostics. En: Industrial & Engineering Chemistry Research. Mayo, 2017, vol.56 no.23., p. 6827–6840.
- [36] LAZZUS, Juan A. A Group Contribution Method to Predict  $\rho$ -T-P of Ionic Liquids. En: Chemical Engineering Communications. Febrero, 2010, vol.197 no.7., p. 974-1015.

- [37] GARDAS, Ramesh L. y COUTINHO, Joao A.P. Extension of the Ye and Shreeve group contribution method for density estimation of ionic liquids in a wide range of temperatures and pressures. En: Fluid Phase Equilibria. Enero, 2008, vol.263 no.1., p. 26-32.
- [38] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. The 2017 Top Programming Languages [En línea]. IEEE Spectrum. 18 de julio de 2017. (Recuperado en 31 enero 2018.) Disponible en <https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>.
- [39] MULLER, Andreas. Scikit-learn, Machine Learning in Python [En línea]. (Recuperado en 3 abril 2018.) Disponible en <http://scikit-learn.org/stable/>
- [40] CENGEL, Yunus y CIMBALA, John. Properties of Fluids. En: Fluid Mechanics. 2 ed. New York: McGraw-Hill, 2006, p. 37-63.
- [41] WU, Yue. High-Pressure and High-Temperature Density Measurements of n-Octane, 2,2,4-Trimethylpentane, Cyclooctane, n-Decane and Toluene. Tesis de Magíster en Ciencias, mención n Ingeniería. Virginia: Virginia Commonwealth University, 2010.
- [42] GOHARSHADI, Elaheh K y NAZARI, Fariba. Computation of internal pressure of liquids using a statistical mechanical equation os state. En: Fluid Phase Equilibria. Septiembre, 2001, vol.187-188., p. 425-431.

- [43] SMITH, J M; VAN NESS, H C y ABBOTT, M M. Solution Thermodynamics: Theory. En: Introduction to Chemical Engineering Thermodynamics. 7 ed. Nueva Deli: Tata McGraw-Hill, 2010. p. 378-420.
- [44] BENSON, George C. y KIYOHARA, Osamu. Thermodynamics of aqueous mixtures of nonelectrolytes. I. Excess volumes of water-n-alcohol mixtures at several temperature. En: Journal of Solution Chemistry. Octubre, 1980, vol.9 no.10., p. 791-804.
- [45] BELL, Caleb. Thermo: Chemical properties component of Chemical Engineering Design Library (ChEDL) [En línea]. 2016. (Recuperado en 17 marzo 2018.) Disponible en <https://github.com/CalebBell/thermo>
- [46] NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION. Database of Hazardous Materials [En línea]. USA. (Recuperado en 1 marzo 2018.) Disponible en <https://cameochemicals.noaa.gov>
- [47] AUCEJO, Antonio, *et al.* Viscosity of Some n -Alkane/1-Chloroalkane Binary Liquid Mixtures. En: Journal of Chemical Engineering Data. 1986, vol.31 no.2., p. 143-145.

## ANEXOS

**Anexo A.** Matrices de los datos utilizados para el entrenamiento de los modelos de densidad.

En la Tabla A1 se registraron las presiones y temperaturas de los datos experimentales usados para el modelamiento de densidad de alcanos; en la Tabla A2 se registraron las presiones y temperaturas de los datos experimentales usados para el modelamiento de densidad de cloroalcanos y, por último, la Tabla A3 contiene las fracciones molares, presiones y temperaturas de los datos experimentales con los que se entrenó el modelo de densidad para sistemas binarios alcano-cloroalcano.

Los datos de densidad experimental usados para el presente trabajo de grado se encuentran en la tesis doctoral “**Estudio ppT de alcanos, cloroalcanos y de sus mezclas binarias**” [12].

Tabla A1. Matriz de datos experimentales usados para entrenar el modelo de densidad de alcanos.

Dato	Compuesto	Temperatura (K)	Presión (MPa)
1	2,2-dimetilbutano	313.15	5
2		308.15	50
3		293.15	65
4		293.15	10
5	2,3-dimetilbutano	328.15	60
6		328.15	5
7		303.15	5
8		323.15	40
9		303.15	30
10		283.15	20
11	2-metilpentano	328.15	10
12		308.15	65
13		323.15	65
14		328.15	5
15		288.15	10
16		293.15	2
17	3-metilpentano	283.15	20
18		328.15	5
19		308.15	2
20		318.15	40
21	n-heptano	328.15	10
22		298.15	40
23		293.15	20
24		308.15	65
25		288.15	7
26		293.15	7
27	n-hexano	303.15	5
28		288.15	65
29		283.15	0.1
30		298.15	50
31		283.15	7
32		283.15	40

Tabla A2. Matriz de datos experimentales usados para entrenar el modelo de densidad de cloroalcanos.

Dato	Compuesto	Temperatura (K)	Presión (MPa)
1	1-clorobutano	328.15	40
2		303.15	5
3		293.15	0.1
4		303.15	0.1
5		298.15	60
6		328.15	2
7		298.15	7
8	1-cloropropano	288.15	30
9		308.15	60
10		318.15	7
11		303.15	40
12	2-clorobutano	308.15	60
13		288.15	30
14		293.15	20
15		318.15	5
16	2-cloropropano	288.15	20
17		303.15	50
18		288.15	7
19	iso-clorobutano	303.15	30
20		293.15	20
21		283.15	0.1
22		298.15	30
23	ter-clorobutano	293.15	50
24		308.15	60
25		283.15	30
26		313.15	2
27		303.15	20
28		328.15	65
29		303.15	10

Tabla A3. Matriz de datos experimentales usados para entrenar el modelo de densidad de sistemas binarios alcano-cloroalcano.

Dato	Sistema	Temperatura (K)	Presión (MPa)	Fracción molar
1	n-heptano y 1-clorobutano	313.15	7	0.9494
2		298.15	2	0.9494
3		323.15	50	0.9011
4		313.15	2	0.6012
5		298.15	7	0.6012
6		323.15	50	0.2013
7		313.15	5	0.5026
8		303.15	5	0.6012
9		323.15	2	0.4008
10		293.15	7	0.2013
11		283.15	0.1	0.0994
12		293.15	5	0.5026
13		283.15	60	0.0508
14		303.15	30	0.2013
15		323.15	65	0.0994
16		303.15	50	0.5026
17		323.15	0.1	0.9494
18		323.15	50	0.0508
19		303.15	30	0.9011
20		283.15	50	0.7995
21		298.15	65	0.9011
22		293.15	2	0.7995
23		293.15	2	0.5026
24		323.15	50	0.0994
25		303.15	20	0.3015
26		283.15	65	0.9011
27		313.15	7	0.9011
28		283.15	5	0.9494
29		293.15	20	0.9011
30		313.15	0.1	0.7003
31		303.15	5	0.9011
32	n-heptano y 2-clorobutano	313.15	40	0.5017
33		298.15	2	0.5017
34		323.15	30	0.1968
35		303.15	2	0.7977
36		303.15	2	0.8986

Tabla A3. Matriz de datos experimentales usados para entrenar el modelo de densidad de sistemas binarios alcano-cloroalcano (continuación).

Dato	Sistema	Temperatura (K)	Presión (MPa)	Fracción molar
37		283.15	65	0.1018
38		293.15	60	0.7977
39		293.15	0.1	0.8986
40		323.15	2	0.952
41		298.15	2	0.6008
42		298.15	60	0.3006
43		298.15	50	0.5017
44		323.15	65	0.4029
45		283.15	0.1	0.0524
46		298.15	40	0.1018
47		283.15	65	0.1968
48		313.15	65	0.0524
49		283.15	2	0.6008
50		303.15	20	0.3006
51		303.15	30	0.7977
52		313.15	0.1	0.1018
53		293.15	50	0.7977
54	n-heptano y 2-clorobutano	313.15	40	0.8986
55		323.15	20	0.952
56		323.15	5	0.3006
57		293.15	2	0.952
58		283.15	0.1	0.952
59		298.15	40	0.7023
60		298.15	30	0.7977
61		323.15	50	0.3006
62		303.15	7	0.1968
63		313.15	30	0.3006
64		323.15	40	0.6008
65		293.15	65	0.0524
66		323.15	10	0.8986
67		323.15	5	0.0524
68		283.15	50	0.8986
69		283.15	0.1	0.6008
70		293.15	60	0.1968
71		323.15	10	0.1968
72		323.15	60	0.3006

Tabla A3. Matriz de datos experimentales usados para entrenar el modelo de densidad de sistemas binarios alcano-cloroalcano (continuación).

Dato	Sistema	Temperatura (K)	Presión (MPa)	Fracción molar
73		298.15	60	0.4986
74		298.15	10	0.3981
75		298.15	5	0.4986
76		293.15	65	0.2009
77		283.15	65	0.3981
78		313.15	65	0.3981
79		313.15	10	0.7038
80		323.15	40	0.3009
81		298.15	40	0.8995
82		303.15	5	0.8995
83		283.15	20	0.7038
84		283.15	5	0.2009
85		303.15	0.1	0.7038
86		303.15	30	0.5998
87		323.15	30	0.5998
88		298.15	20	0.1023
89		298.15	40	0.4986
90	n-heptano y iso-clorobutano	323.15	2	0.5998
91		298.15	40	0.0531
92		323.15	40	0.3981
93		313.15	60	0.1023
94		293.15	40	0.3009
95		303.15	7	0.3981
96		293.15	40	0.7038
97		283.15	50	0.0531
98		323.15	20	0.9507
99		283.15	0.1	0.7993
100		283.15	40	0.8995
101		283.15	60	0.9507
102		293.15	20	0.4986
103		293.15	65	0.4986
104		303.15	0.1	0.8995
105		283.15	30	0.4986
106		313.15	40	0.1023
107		293.15	60	0.9507
108		293.15	10	0.1023
109		313.15	65	0.1023

Tabla A3. Matriz de datos experimentales usados para entrenar el modelo de densidad de sistemas binarios alcano-cloroalcano (continuación).

Dato	Sistema	Temperatura (K)	Presión (MPa)	Fracción molar
110	n-heptano y iso-clorobutano	283.15	7	0.9507
111		323.15	7	0.3009
112		293.15	5	0.9507
113	n-heptano y ter-clorobutano	313.15	2	0.9032
114		298.15	5	0.7995
115		313.15	2	0.0554
116		283.15	2	0.3016
117		323.15	2	0.0554
118		298.15	50	0.0554
119		303.15	10	0.3016
120		283.15	10	0.7995
121		323.15	5	0.0554
122		313.15	30	0.4097
123		303.15	40	0.7995
124		293.15	10	0.3016
125		303.15	7	0.9032
126		293.15	7	0.9032
127		303.15	60	0.7995
128		293.15	30	0.3016
129		283.15	60	0.498
130		283.15	60	0.4097
131		313.15	0.1	0.1017
132		303.15	5	0.5983
133		323.15	20	0.4097
134		313.15	30	0.0554
135		298.15	20	0.2004
136		293.15	0.1	0.2004
137		313.15	30	0.9482
138		293.15	65	0.0554
139		313.15	65	0.9482
140		323.15	20	0.1017
141		313.15	2	0.1017
142		313.15	20	0.5983
143		313.15	30	0.498
144		283.15	50	0.9032
145		283.15	20	0.498
146	313.15	5	0.5983	

Tabla A3. Matriz de datos experimentales usados para entrenar el modelo de densidad de sistemas binarios alcano-cloroalcano (continuación).

Dato	Sistema	Temperatura (K)	Presión (MPa)	Fracción molar
147		323.15	50	0.497
148		303.15	10	0.202
149		323.15	30	0.5957
150		323.15	30	0.9497
151		298.15	5	0.0483
152		283.15	50	0.3026
153		283.15	5	0.9017
154		298.15	5	0.5957
155		283.15	65	0.9497
156		303.15	2	0.8012
157		283.15	10	0.9017
158		323.15	50	0.099
159		293.15	7	0.9497
160	n-hexano y 1-clorobutano	303.15	60	0.9017
161		313.15	2	0.9497
162		323.15	40	0.8012
163		298.15	10	0.6998
164		298.15	40	0.5957
165		298.15	20	0.099
166		313.15	5	0.9017
167		293.15	5	0.8012
168		283.15	10	0.0483
169		283.15	60	0.5957
170		303.15	10	0.5957
171		313.15	50	0.8012
172		313.15	30	0.8012
173		303.15	20	0.3026
174		323.15	7	0.3996
175	313.15	20	0.3996	
176	n-hexano y 2-clorobutano	298.15	10	0.6013
177		303.15	65	0.6013
178		298.15	60	0.9532
179		323.15	7	0.0982
180		313.15	65	0.2995
181		313.15	10	0.047
182		293.15	20	0.8997
183		323.15	5	0.4979

Tabla A3. Matriz de datos experimentales usados para entrenar el modelo de densidad de sistemas binarios alcano-cloroalcano (continuación).

Dato	Sistema	Temperatura (K)	Presión (MPa)	Fracción molar
184		323.15	60	0.6013
185		298.15	7	0.8997
186		298.15	50	0.6973
187		298.15	60	0.6973
188		293.15	5	0.047
189		323.15	30	0.4979
190		298.15	5	0.047
191		303.15	20	0.047
192		323.15	50	0.0982
193		313.15	20	0.0982
194		298.15	60	0.047
195		283.15	30	0.8997
196		323.15	60	0.3983
197		323.15	65	0.6013
198		293.15	2	0.0982
199		303.15	20	0.9532
200	n-hexano y 2-clorobutano	313.15	50	0.3983
201		283.15	7	0.9532
202		293.15	7	0.4979
203		283.15	30	0.4979
204		323.15	65	0.6973
205		283.15	50	0.3983
206		293.15	10	0.0982
207		303.15	2	0.7985
208		303.15	20	0.0982
209		298.15	20	0.9532
210		283.15	20	0.6973
211		298.15	30	0.3983
212		293.15	7	0.047
213		323.15	20	0.8997
214		313.15	40	0.2995
215		293.15	30	0.9532
216		283.15	2	0.3983
217		313.15	10	0.0982
218	n-hexano y iso-clorobutano	313.15	0.1	0.8991
219		303.15	2	0.5991

Tabla A3. Matriz de datos experimentales usados para entrenar el modelo de densidad de sistemas binarios alcano-cloroalcano (continuación).

Dato	Sistema	Temperatura (K)	Presión (MPa)	Fracción molar
220		313.15	60	0.0479
221		298.15	10	0.401
222		323.15	10	0.953
223		298.15	40	0.8991
224		283.15	5	0.5991
225		313.15	10	0.5039
226		298.15	30	0.7018
227		303.15	30	0.1971
228		298.15	7	0.7971
229		323.15	50	0.2992
230		303.15	60	0.401
231		323.15	10	0.0986
232		293.15	7	0.8991
233		293.15	60	0.401
234		313.15	0.1	0.5039
235		293.15	7	0.7018
236	n-hexano y iso-clorobutano	298.15	5	0.5039
237		298.15	2	0.2992
238		298.15	0.1	0.1971
239		298.15	40	0.953
240		313.15	5	0.953
241		283.15	50	0.7018
242		323.15	40	0.401
243		298.15	20	0.1971
244		303.15	40	0.5039
245		293.15	10	0.8991
246		303.15	65	0.1971
247		293.15	20	0.7971
248		303.15	0.1	0.2992
249		303.15	2	0.0479
250		293.15	20	0.1971
251		313.15	0.1	0.953
252		298.15	65	0.5991
253		298.15	40	0.0479
254		323.15	0.1	0.7018

Tabla A3. Matriz de datos experimentales usados para entrenar el modelo de densidad de sistemas binarios alcano-cloroalcano (continuación).

Dato	Sistema	Temperatura (K)	Presión (MPa)	Fracción molar
255		298.15	5	0.6989
256		303.15	10	0.3998
257		303.15	65	0.5004
258		313.15	30	0.3998
259		283.15	20	0.3998
260		293.15	60	0.1991
261		323.15	50	0.9015
262		283.15	65	0.0981
263		323.15	7	0.3998
264		303.15	60	0.6989
265		293.15	10	0.7977
266		283.15	10	0.3009
267		298.15	60	0.9465
268		303.15	30	0.3998
269		303.15	20	0.1991
270		293.15	50	0.9015
271		298.15	40	0.3998
272	n-hexano y ter-	298.15	5	0.1991
273	clorobutano	283.15	40	0.0981
274		293.15	5	0.5989
275		283.15	65	0.9465
276		303.15	5	0.0504
277		303.15	10	0.1991
278		313.15	2	0.1991
279		283.15	7	0.5004
280		323.15	7	0.6989
281		283.15	5	0.3009
282		323.15	65	0.5004
283		323.15	50	0.0504
284		298.15	50	0.0504
285		293.15	10	0.3009
286		313.15	20	0.0981
287		293.15	5	0.1991
288		313.15	60	0.5004
289		293.15	20	0.5989
290		298.15	2	0.9465

## Anexo B. Grupos de contribución de algunos alcanos y cloroalcanos.

En la Tabla B1 se registran los grupos de contribución de los compuestos usados para el entrenamiento de los modelos de densidades propuestos, a su vez, la Tabla B2 muestra los grupos de contribución de los compuestos con lo que se realizó la comparación de resultados de densidades en el presente trabajo de grado.

Tabla B1. Grupos de contribución de alcanos y cloroalcanos usados para el modelamiento de densidad.

Compuesto	C <sub>1</sub> -C <sub>2</sub>	C <sub>1</sub> -C <sub>3</sub>	C <sub>1</sub> -C <sub>4</sub>	C <sub>2</sub> -C <sub>2</sub>	C <sub>2</sub> -C <sub>3</sub>	C <sub>2</sub> -C <sub>4</sub>	C <sub>3</sub> -C <sub>3</sub>	C <sub>1</sub> -Cl	C <sub>2</sub> -Cl	C <sub>3</sub> -Cl
n-hexano	2	-	-	3	-	-	-	-	-	-
n-heptano	2	-	-	4	-	-	-	-	-	-
2-metilpentano	1	2	-	1	1	-	-	-	-	-
3-metilpentano	2	1	-	-	2	-	-	-	-	-
2,3-dimetilbutano	-	4	-	-	-	-	1	-	-	-
2,2-dimetilbutano	1	-	3	-	-	1	-	-	-	-
1-cloropropano	2	-	-	-	-	-	-	1	-	-
2-cloropropano	2	-	-	-	-	-	-	-	1	-
1-clorobutano	2	-	-	1	-	-	-	1	-	-
2-clorobutano	2	-	-	1	-	-	-	-	1	-
iso-clorobutano	-	3	-	-	-	-	-	1	-	-
ter-clorobutano	-	3	-	-	-	-	-	-	-	1

Tabla B2. Grupos de contribución de algunos alcanos y cloroalcanos.

Compuesto	C <sub>1</sub> -C <sub>2</sub>	C <sub>1</sub> -C <sub>3</sub>	C <sub>1</sub> -C <sub>4</sub>	C <sub>2</sub> -C <sub>2</sub>	C <sub>2</sub> -C <sub>3</sub>	C <sub>2</sub> -C <sub>4</sub>	C <sub>1</sub> -Cl
1,4-diclorobutano	2	-	-	1	-	-	2
1,6-diclorohexano	2	-	-	3	-	-	2
2,2,4-trimetilpentano	0	1	1	-	1	1	-
ciclooctano	-	-	-	7	-	-	-
isopentano	1	2	-	-	1	-	-
n-decano	2	-	-	7	-	-	-
n-dodecano	2	-	-	9	-	-	-
n-nonano	2	-	-	6	-	-	-
n-octano	2	-	-	5	-	-	-
n-pentano	2	-	-	2	-	-	-
n-undecano	2	-	-	8	-	-	-
1-cloropropano	2	-	-	-	-	-	1
1-clorobutano	2	-	-	1	-	-	1
1-cloropentano	2	-	-	2	-	-	1
1-cloroheptano	2	-	-	4	-	-	1
1-clorooctano	2	-	-	5	-	-	1
1-clorodecano	2	-	-	7	-	-	1
1-clorotradecano	2	-	-	11	-	-	1
n-tridecano	2	-	-	10	-	-	-

### Anexo C. Gráficas de variación de densidad experimental y modelada.

- Alcanos: Las Figuras C1 a la C3 representan la variación de la densidad de algunos alcanos, tanto experimental como modelada, con respecto a la presión, para diferentes temperaturas.
- Cloroalcanos: Las Figuras C4 a la C6 representan la variación de la densidad de algunos cloroalcanos, tanto experimental como modelada, con respecto a la presión, para diferentes temperaturas.
- Sistemas binarios alcano-cloroalcano: Las Figuras C7 a la C10 representan la variación de la densidad de algunas mezclas binarias, tanto experimental como modelada, con respecto a la fracción molar, a 30 MPa y diferentes temperaturas.

Figura C1. Variación de la densidad experimental (·) y modelada (-) con respecto a la presión, a diferentes temperaturas para el n-hexano (izda.) y n-heptano (dcha.).

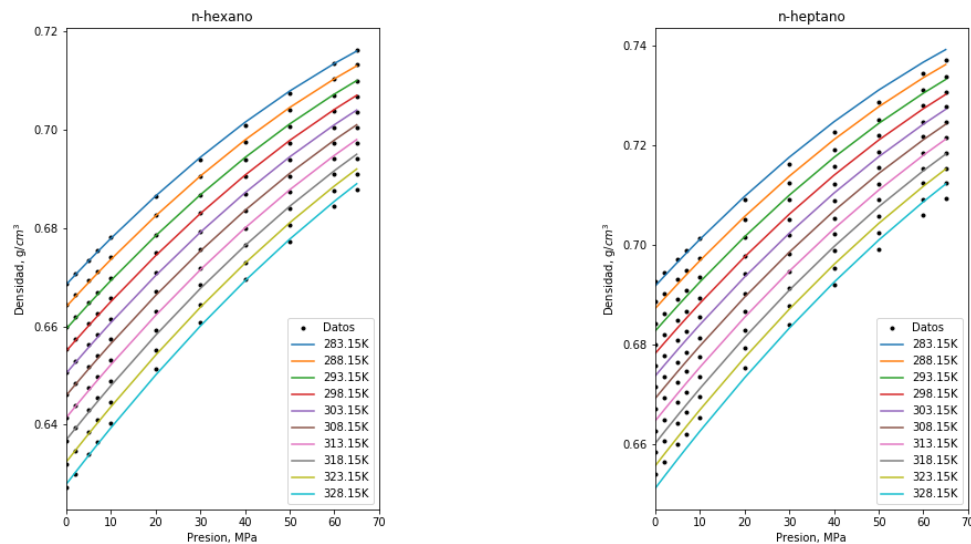


Figura C2. Variación de la densidad experimental (·) y modelada (-) con respecto a la presión, a diferentes temperaturas para el 2-metilpentano (izda.) y 3-metilpentano (dcha.).

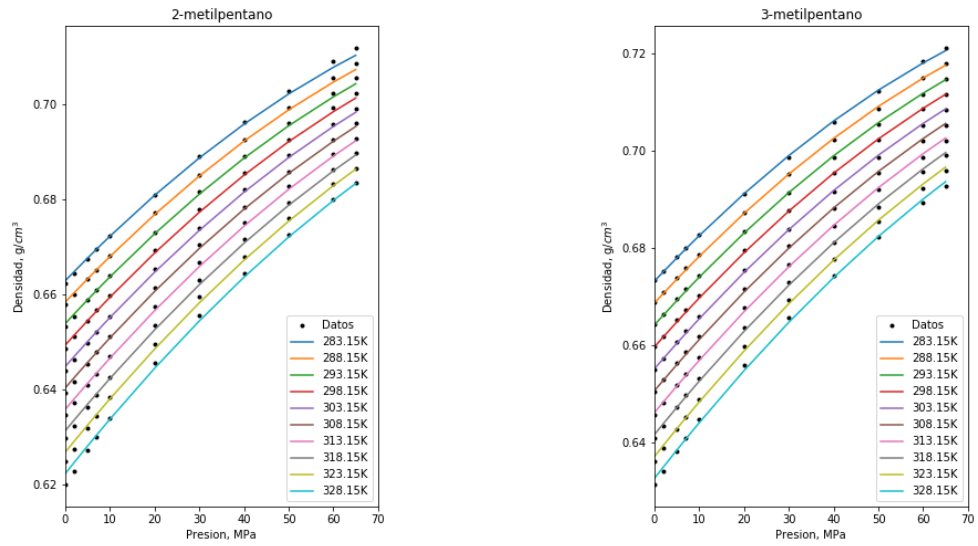


Figura C3. Variación de la densidad experimental (·) y modelada (-) con respecto a la presión, a diferentes temperaturas para el 2,3-dimetilbutano (izda.) y 2,2-dimetilbutano (dcha.).

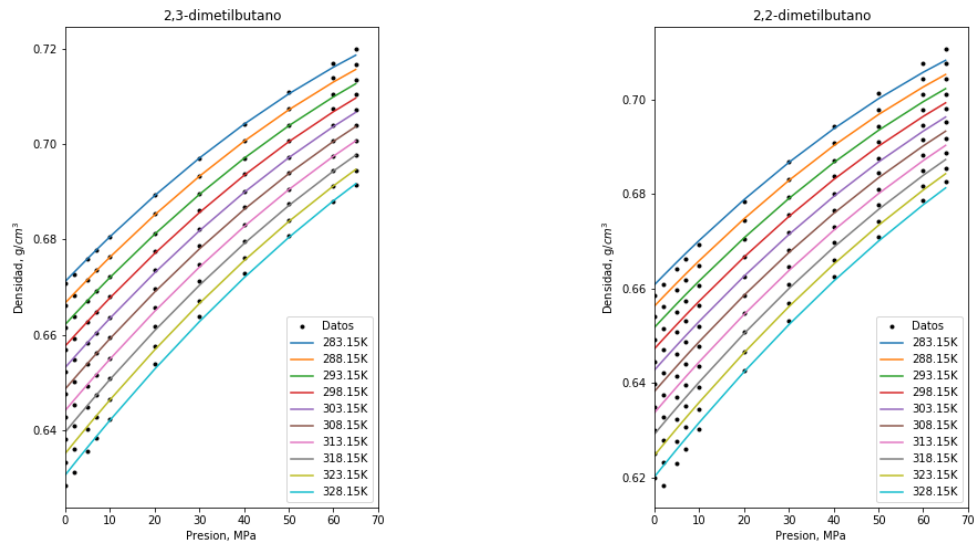


Figura C4. Variación de la densidad experimental (·) y modelada (-) con respecto a la presión, a diferentes temperaturas para el 1-cloropropano (izda.) y 2-cloropropano (dcha.).

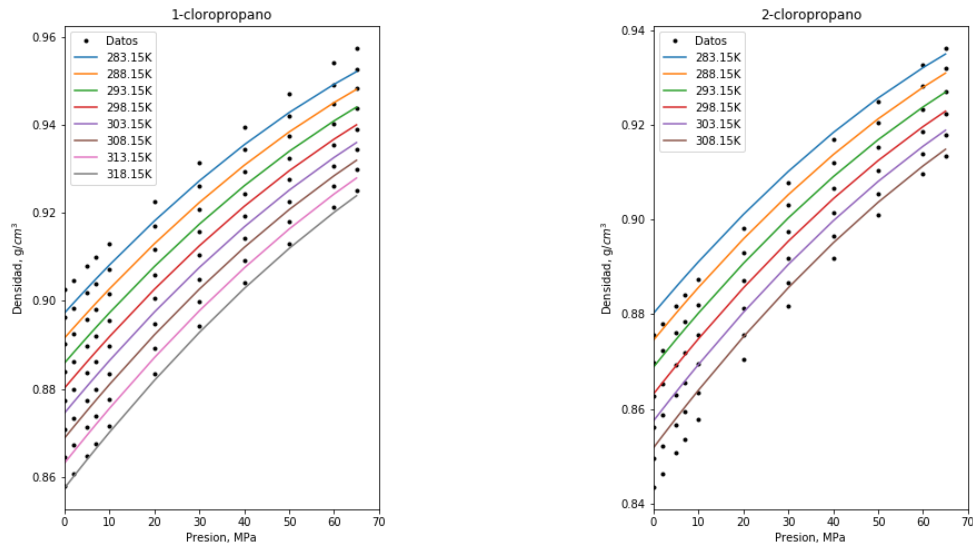


Figura C5. Variación de la densidad experimental (·) y modelada (-) con respecto a la presión, a diferentes temperaturas para el 1-clorobutano (izda.) y 2-clorobutano (dcha.).

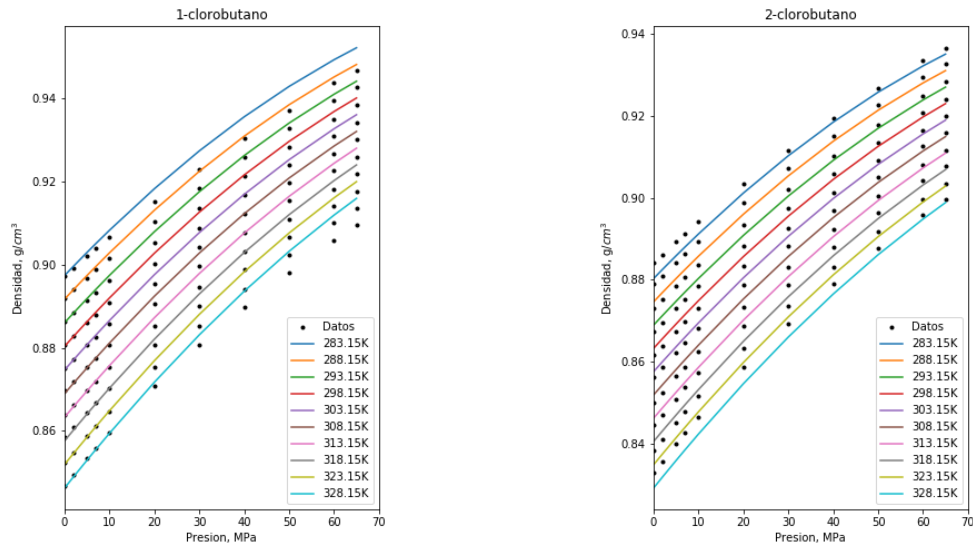


Figura C6. Variación de la densidad experimental (·) y modelada (-) con respecto a la presión, a diferentes temperaturas para el iso-clorobutano (izda.) y ter-clorobutano (dcha.).

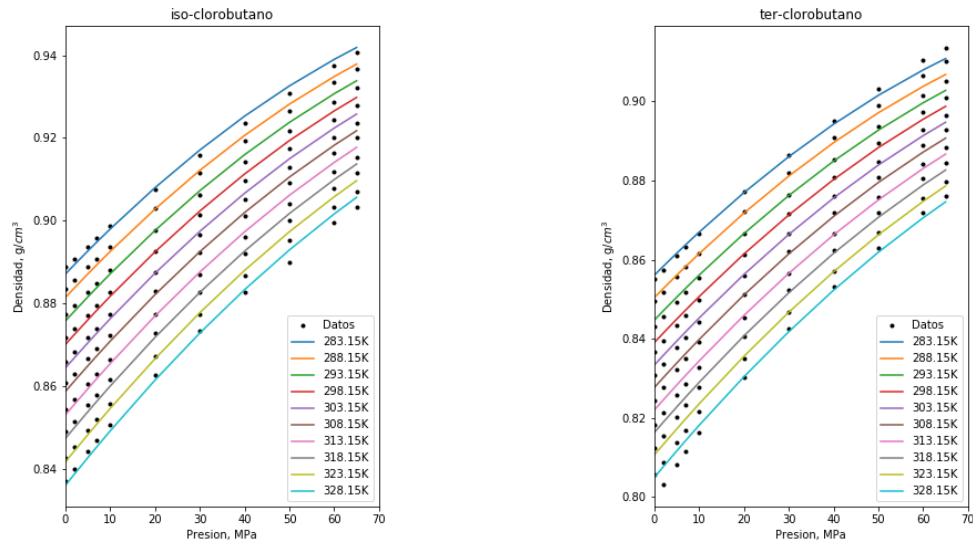


Figura C7. Variación de la densidad experimental (·) y modelada (-) con respecto a la fracción molar de n-hexano, a diferentes temperaturas y 30 MPa para las mezclas con 1-clorobutano (izda.) y 2-clorobutano (dcha.).

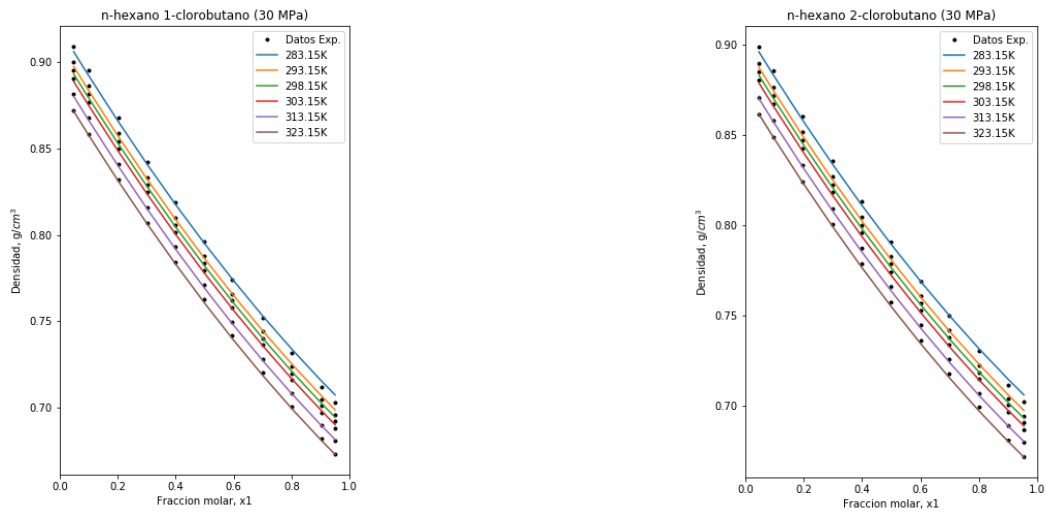


Figura C8. Variación de la densidad experimental (·) y modelada (-) con respecto a la fracción molar de n-hexano, a diferentes temperaturas y 30 MPa para las mezclas con iso-clorobutano (izda.) y ter-clorobutano (dcha.).

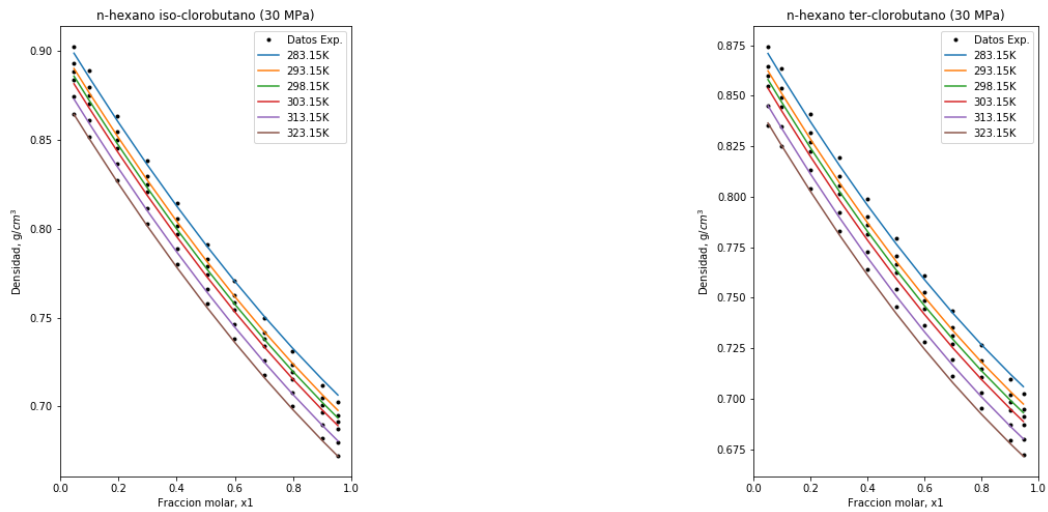


Figura C9. Variación de la densidad experimental (·) y modelada (-) con respecto a la fracción molar de n-heptano, a diferentes temperaturas y 30 MPa para las mezclas con 1-clorobutano (izda.) y 2-clorobutano (dcha.).

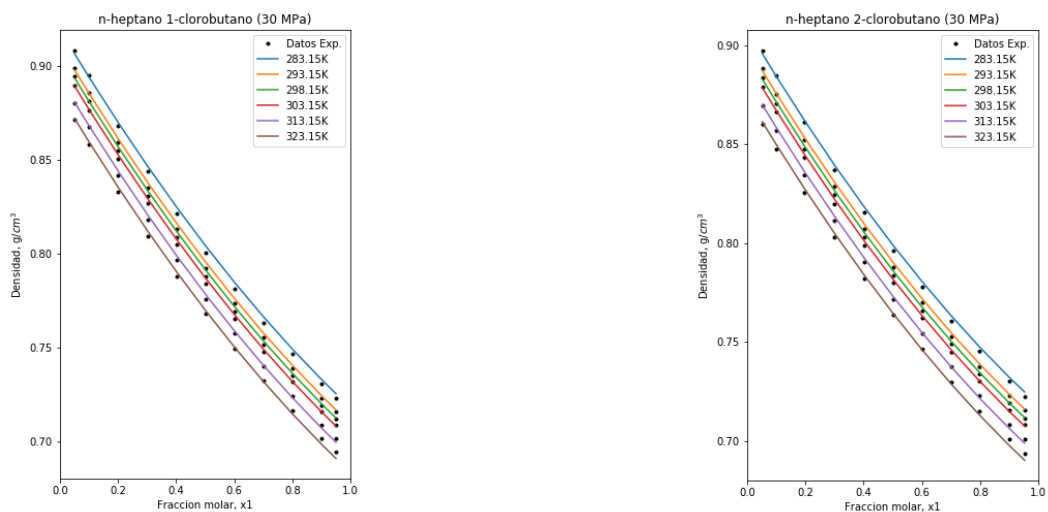
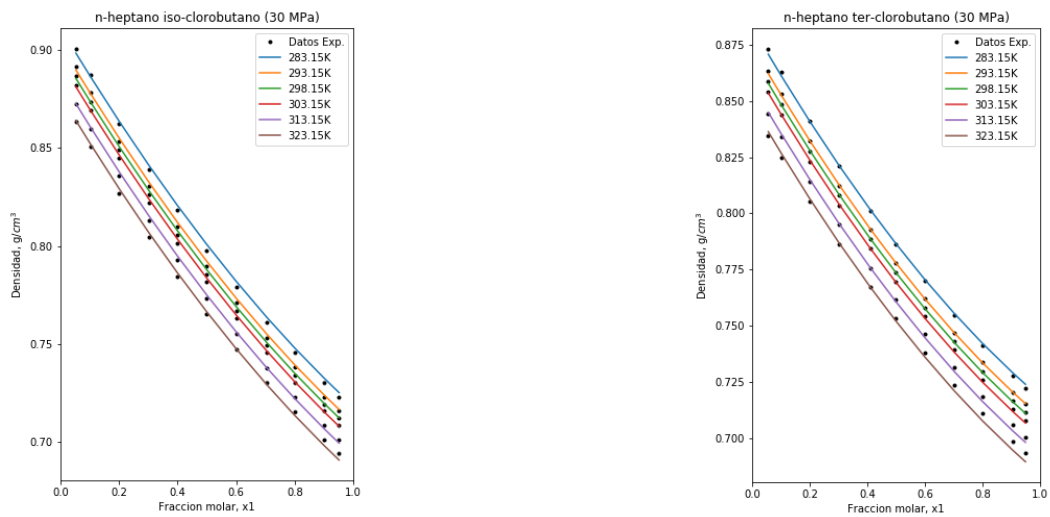


Figura C10. Variación de la densidad experimental (·) y modelada (-) con respecto a la fracción molar de n-heptano, a diferentes temperaturas y 30 MPa para las mezclas con iso-clorobutano (izda.) y ter-clorobutano (dcha.).



**Anexo D.** Gráficas de comparación de los modelos propuestos con datos provenientes de la literatura y estimaciones de Peng-Robinson y COSTALD.

Las Figuras D1 a la D5 representan la variación de la densidad con respecto a la presión a diferentes temperaturas para algunos alcanos según datos encontrados en la literatura y densidades estimadas por el modelo de alcanos propuesto, por Peng-Robinson y por COSTALD, de igual manera, en las Figuras D6 a la D8 se comparan las densidades de algunos alcanos a condiciones ambientales.

La Figura D9 representan la variación de la densidad con respecto a la presión a diferentes temperaturas para el 1-cloropropano según datos encontrados en la literatura y densidades estimadas por el modelo de alcanos propuesto, por Peng-Robinson y por COSTALD, a su vez, en las Figuras D10 a la D13 se representa la variación de la densidad con respecto a la temperatura a presión atmosférica, para diferentes cloroalcanos.

En la Figura D14 se muestra la variación de la densidad con respecto a las fracciones molares a condiciones ambientales de algunos sistemas binarios alcano-cloroalcano según datos encontrados en la literatura y estimaciones del modelo propuesto para sistemas binarios alcano-cloroalcano y estimaciones de COSTALD.

Figura D1. Comparación de la variación de la densidad experimental del n-pentano con respecto a la presión, a diferentes temperaturas con la densidad estimada por (de izda. a dcha.) el modelo propuesto, Peng-Robinson y COSTALD.

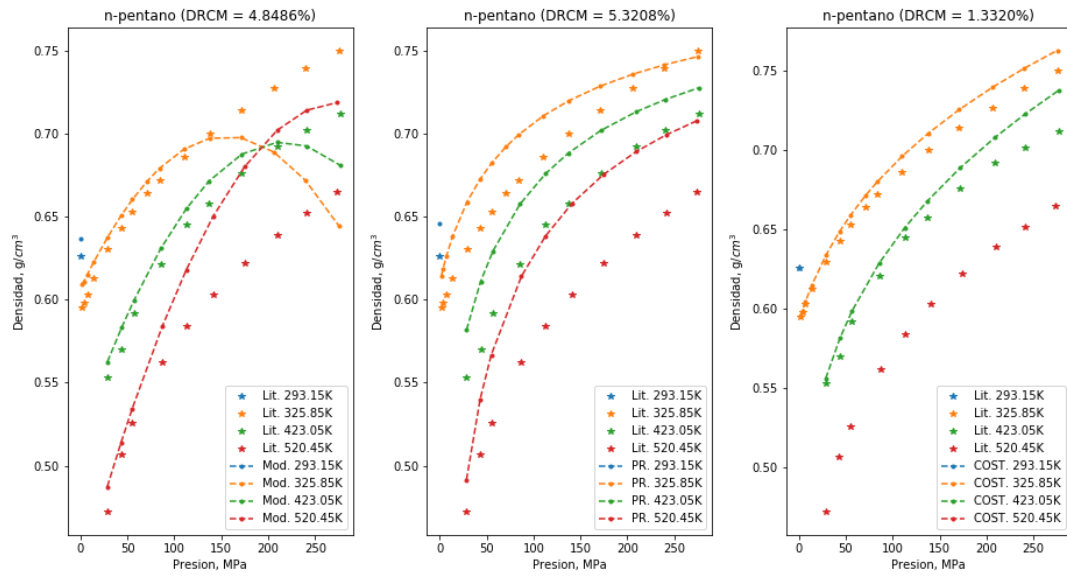


Figura D2. Comparación de la variación de la densidad experimental del n-octano con respecto a la presión, a diferentes temperaturas con la densidad estimada por (de izda. a dcha.) el modelo propuesto, Peng-Robinson y COSTALD.

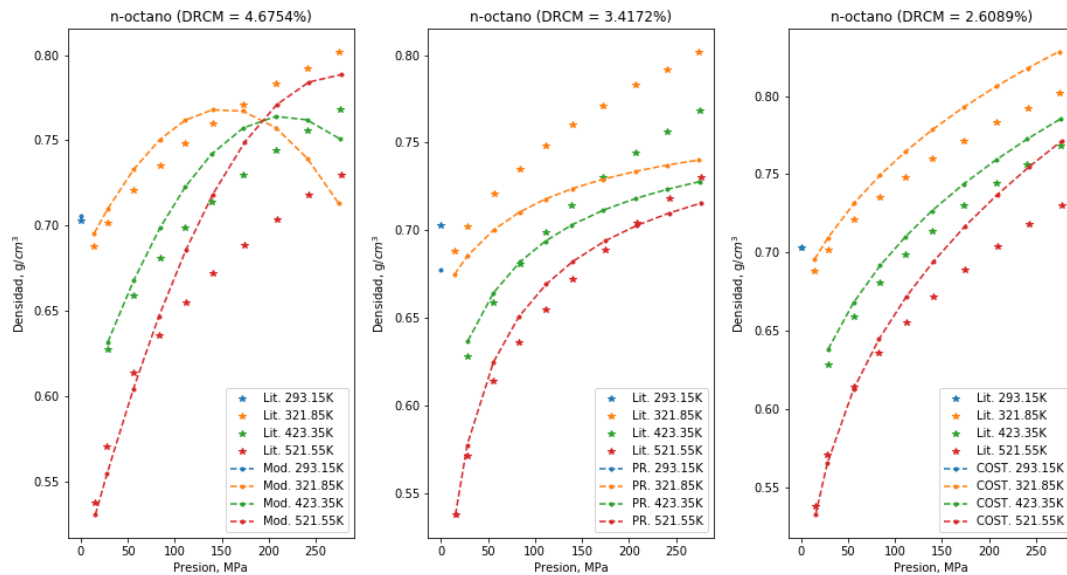


Figura D3. Comparación de la variación de la densidad experimental del 2,2,4-trimetilpentano con respecto a la presión, a diferentes temperaturas con la densidad estimada por (de izda. a dcha.) el modelo propuesto, Peng-Robinson y COSTALD.

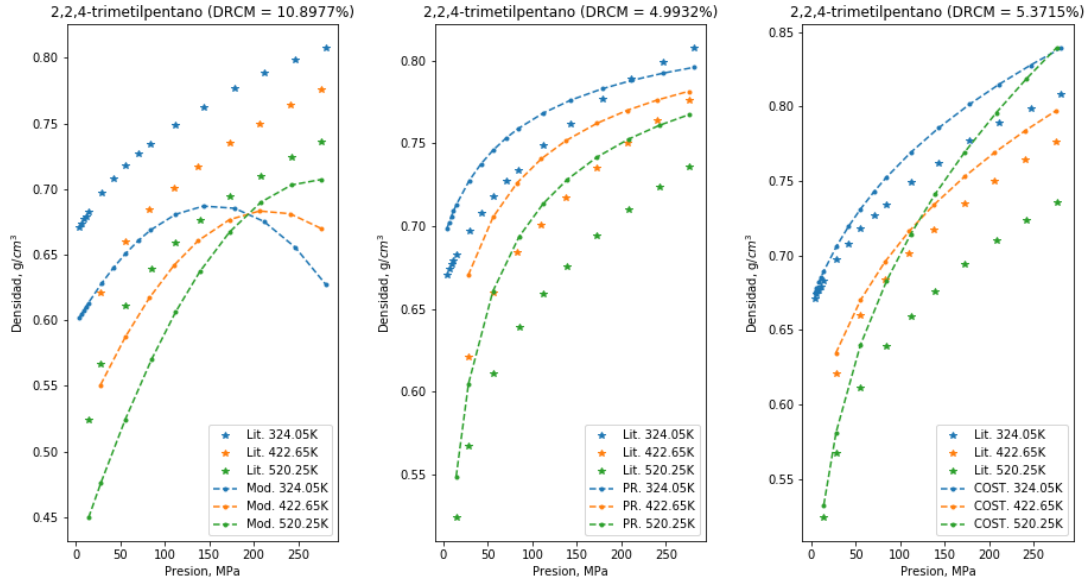


Figura D4. Comparación de la variación de la densidad experimental del ciclooctano con respecto a la presión, a diferentes temperaturas con la densidad estimada por (de izda. a dcha.) el modelo propuesto, Peng-Robinson y COSTALD.

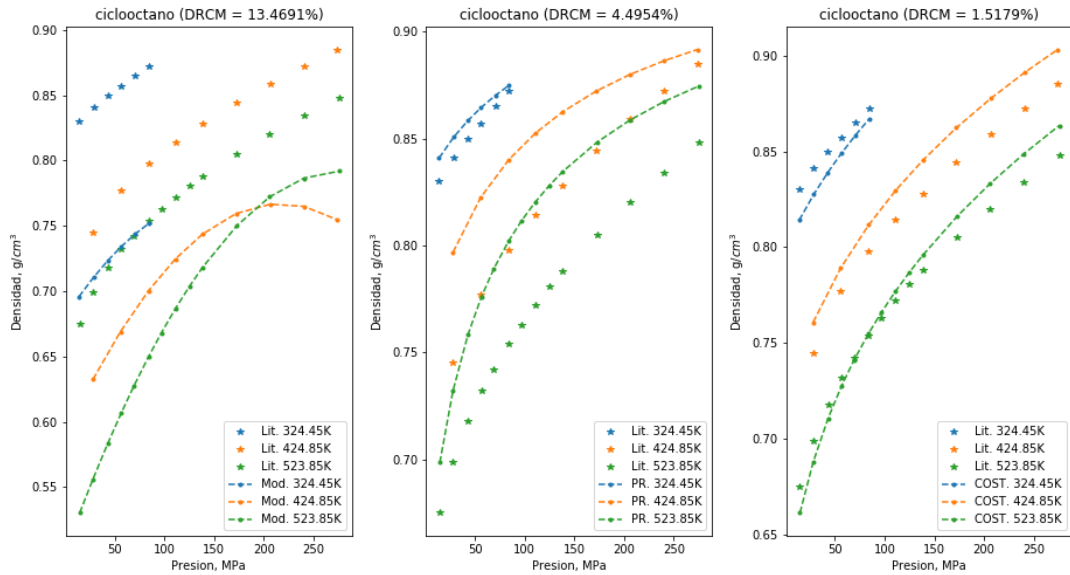


Figura D5. Comparación de la variación de la densidad experimental del n-decano con respecto a la presión, a diferentes temperaturas con la densidad estimada por (de izda. a dcha.) el modelo propuesto, Peng-Robinson y COSTALD.

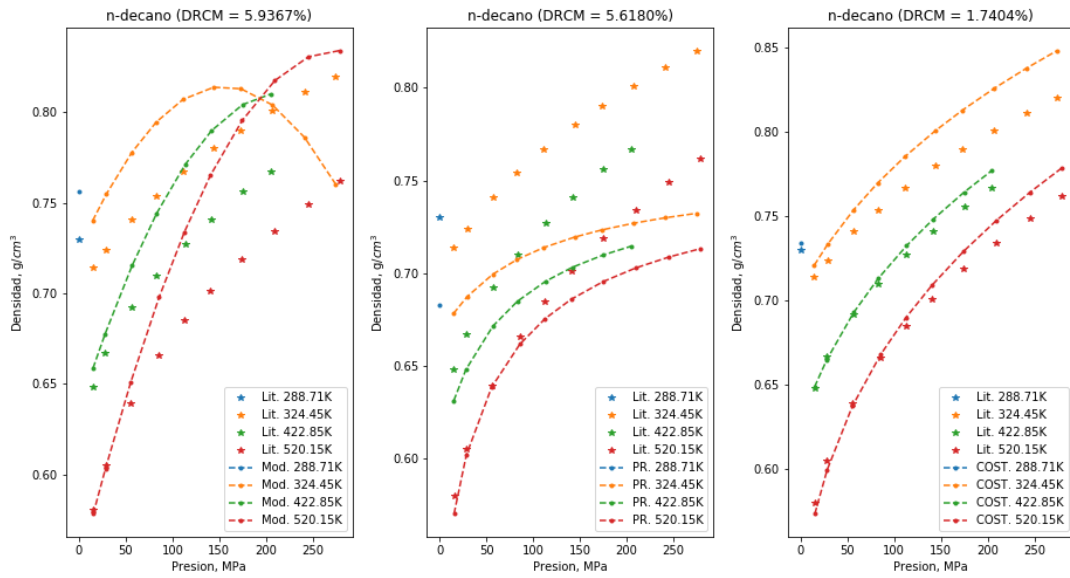


Figura D6. Densidad experimental y estimada del isopentano (izda.) y n-nonano (dcha.) a 20 °C y 1 atm.

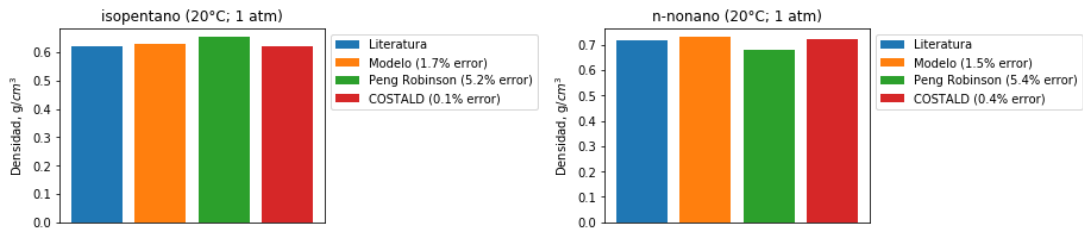


Figura D7. Densidad experimental y estimada del n-undecano (izda.) y n-dodecano (dcha.) a 20 °C y 1 atm.

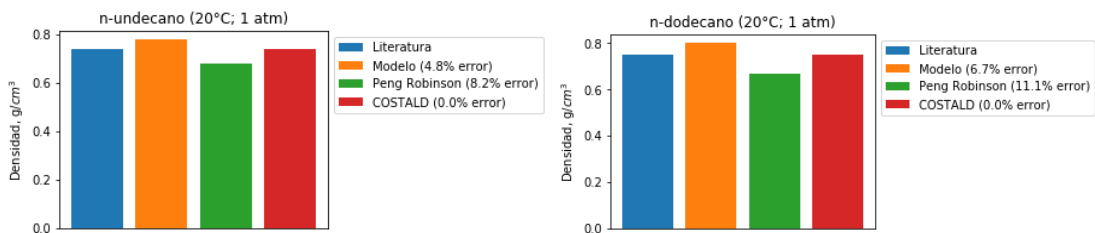


Figura D8. Densidad experimental y estimada del n-tridecano a 20 °C y 1 atm.

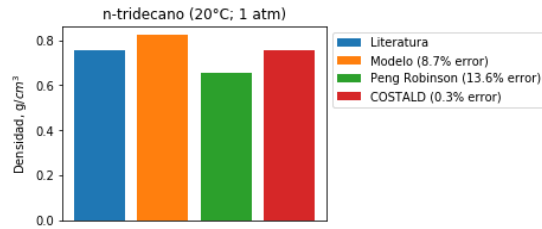


Figura D9. Comparación de la variación de la densidad experimental del 1-cloropropano con respecto a la presión, a diferentes temperaturas con la densidad estimada por (de izda. a dcha.) el modelo propuesto, Peng-Robinson y COSTALD.

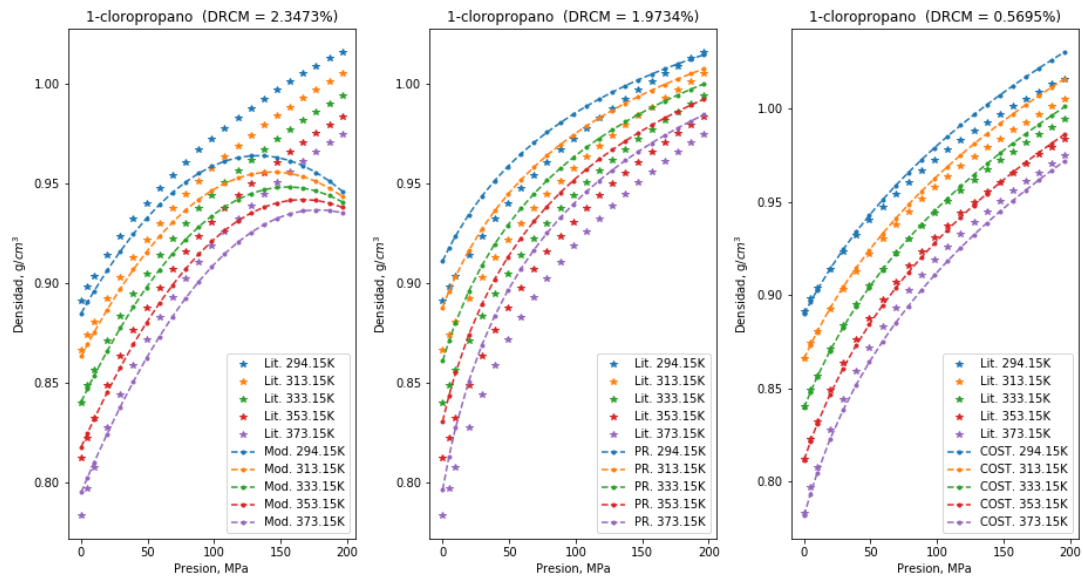


Figura D10. Variación de la densidad experimental y estimada con respecto a la temperatura a presión atmosférica del 1-cloropropano (izda.) y 1-clorobutano (dcha.).

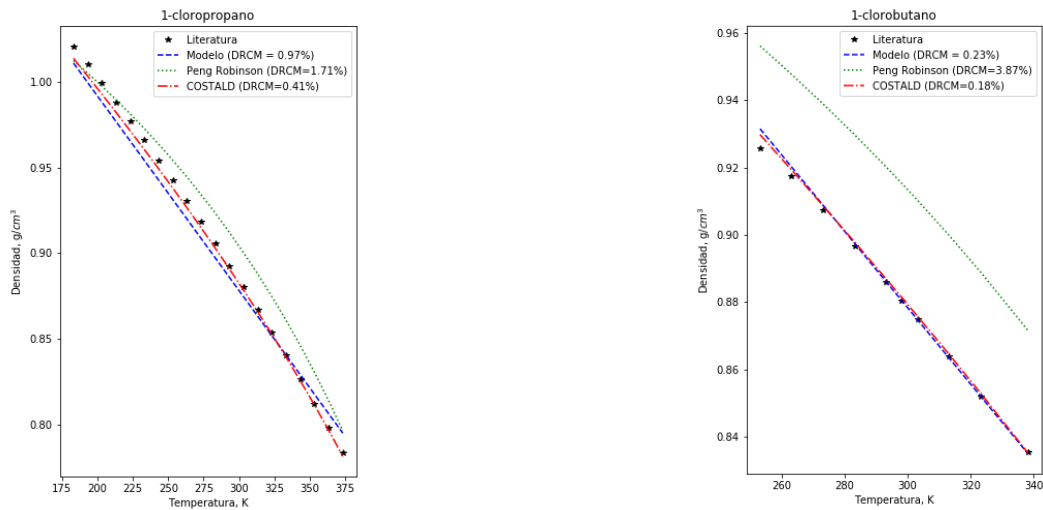


Figura D11. Variación de la densidad experimental y estimada con respecto a la temperatura a presión atmosférica del 1-cloropentano (izda.) y 1-cloroheptano (dcha.).

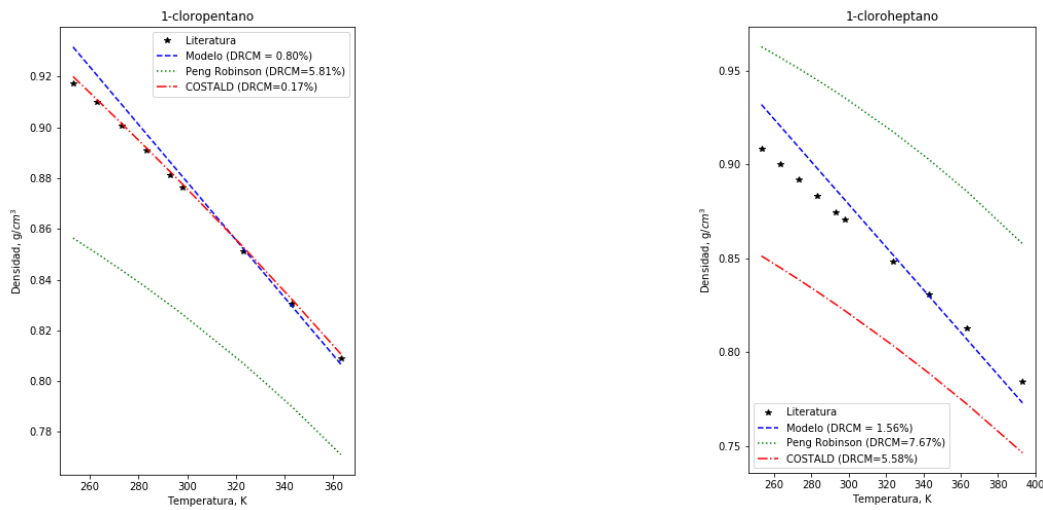


Figura D12. Variación de la densidad experimental y estimada con respecto a la temperatura a presión atmosférica del 1-clorooctano (izda.) y 1-clorodecano (dcha.).

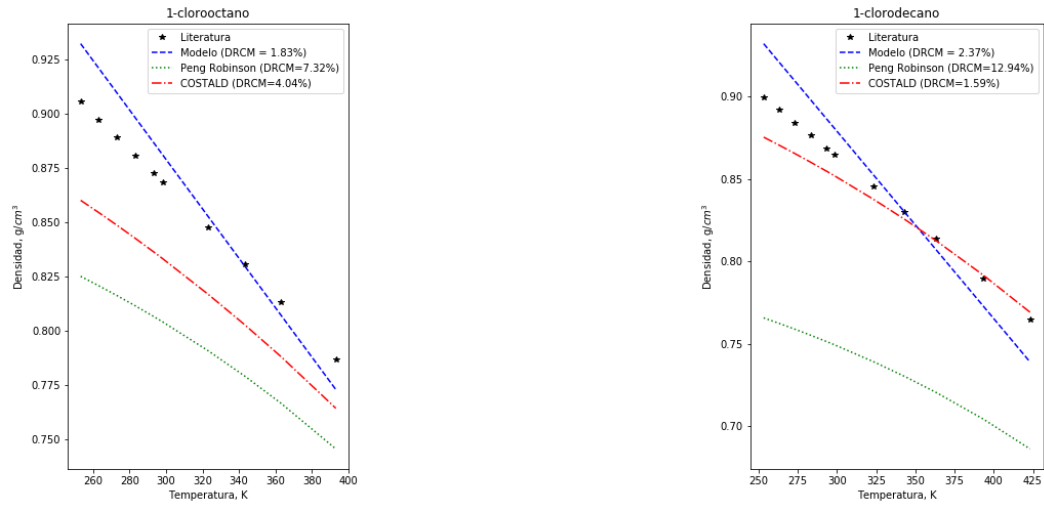


Figura D13. Variación de la densidad experimental y estimada con respecto a la temperatura a presión atmosférica del 1-clorotetradecano.

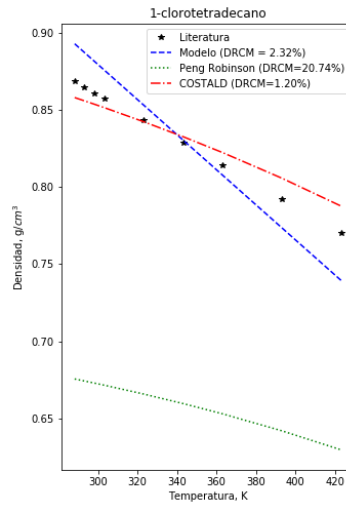
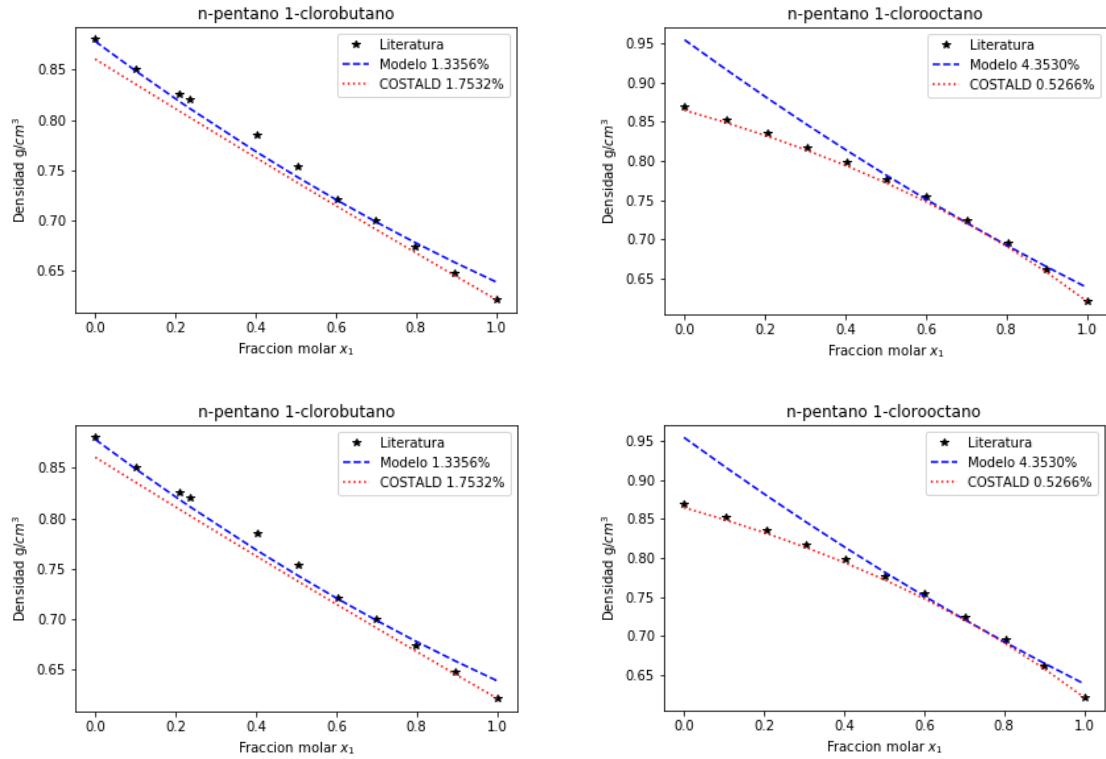


Figura D14. Variación de la densidad experimental y estimada a 25°C y 1 atm respecto a la fracción molar para los sistemas: n-pentano y 1-clorobutano (sup. izda.), n-pentano y 1-clorooctano (sup. dcha.), n-pentano y 1-clorobutano (inf. izda.), n-pentano y 1-clorooctano (inf. dcha.).



## Anexo E. Códigos de programación en Python.

- **Modelo de densidad de alcanos y cloroalcanos**

```
# ##### Importar las librerías pertinentes
import os
import pickle
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from Funciones import *

# ##### Importar las tablas y guardarlas en un diccionario de Python
# Las Tablas 7-18 son para los compuestos puros (guardadas como .txt).
# * _DatosTablas_ = Diccionario con cada tabla (guardada bajo el nombre
del compuesto estudiado)
# * _compuestos_ = vector con los nombres de los compuestos de las tablas
(en el mismo orden que las tablas).
# * _tablas_ = vector con los nombres de los .txt.

#Obtener la dirección de la carpeta Modelo Lineal
path=os.path.abspath('').[:-7]

#Crear el diccionario para almacenar las tablas
DatosTablas={}

compuestos,tablas=DensCompPuros() #Funcion de Funciones.py

#Importar cada tabla y guardar en diccionario
for tabla,comp in zip(tablas,compuestos):
    DatosTablas[comp]=pd.read_csv(path + 'Tablas\\Compuestos puros\\' +
tabla,index_col=0)
    DatosTablas[comp].index.name=comp

print('Verificar que las tablas se importaron correctamente. Por
ejemplo:')
DatosTablas['2,2-dimetilbutano']

# ##### Importar los grupos de contribucion de los compuestos estudiados
GruposContr=pd.read_csv(path + 'Grupos y
PM\\GruposContribucion.csv',index_col=0).fillna(0)
GruposContr

# ##### Organizar los datos para entrenar el modelo
#Funcion para aplanar los datos
def CrearDatosFilas(comps):
    #Crear el diccionario para guardar las tablas
    DatosFilas={}
    #Hacer un ciclo para organizar las tablas de cada compuesto
    for comp in comps:
        #Extraer las presiones (indices de cada tabla)
```

```

P=np.array(DatosTablas[comp].index,dtype=float)
#Extraer las temperaturas (nombres de columnas de cada tabla)
T=np.array(DatosTablas[comp].columns,dtype=float)
#Obtener todas las combinaciones de T y P
Temp,Pres=np.meshgrid(T,P)
#Extraer las densidades (valores de la tabla)
Dens=DatosTablas[comp].values

#Aplanar las matrices y unir las en una sola matriz
#Agregar T*P y P^2

flat=np.column_stack((Temp.ravel(),Pres.ravel(),Temp.ravel()*Pres.ravel(),
Pres.ravel()**2,Dens.ravel()))
#Guardar la matriz como un DataFrame

flat_data=pd.DataFrame(flat,columns=['Temperatura','Presion','T*P','P^2',
'Densidad'])
#Guardar el nombre del compuesto en la ultima columna (llamada
'Compuesto')
flat_data['Compuesto']=comp

#Obtener el numero de filas
nfilas=flat.shape[0]
#Obtener el numero de grupos de contribucion
ncols=GruposContr.columns.shape[0]
#Obtener los grupos de contribucion del compuesto
grupos=GruposContr.loc[comp]
#Copiar los grupos n veces, reorganizar en una matriz y guardar
en un DataFrame
copy_grupos=np.tile(grupos,nfilas)
groups_data=np.resize(copy_grupos,[nfilas,ncols])
groups_data=pd.DataFrame(groups_data,columns=GruposContr.columns)

#Concatenar las tablas aplanadas y las de los grupos
horizontalmente
DatosFilas[comp]=pd.concat([groups_data,flat_data],axis=1)
return DatosFilas

#Crear tabla aplanada para alcanos, cloroalcanos y todos los compuestos
DatosFilasAlc=CrearDatosFilas(compuestos[:6])
DatosFilasCl=CrearDatosFilas(compuestos[6:])
DatosFilas=CrearDatosFilas(compuestos)

#Guardar el diccionario con todos los compuestos
with open(path + 'Pickle y CSV\\DatosDensidadP.pickle', 'wb') as handle:
    pickle.dump(DatosFilas, handle, protocol=pickle.HIGHEST_PROTOCOL)

print('Un ejemplo de una tabla organizada (n-hexano) es:')
DatosFilas['n-hexano'].tail()

#Concatenar cada tabla de los diccionarios para tener una tabla maestra
para alcanos y una para cloroalcanos.
#Se deben remover las filas que tengan NaN (donde faltaba el valor de la
densidad).

```

```

#Se debe resetear el indice para que la enumeracion de las filas no tenga huecos.
TodosDatosAlc=pd.concat(DatosFilasAlc,axis=0,ignore_index=True).dropna().reset_index(drop=True)
TodosDatosCl=pd.concat(DatosFilasCl,axis=0,ignore_index=True).dropna().reset_index(drop=True)

# #### Crear el modelo lineal
#Crear los modelos para los alcanos y para los cloroalcanos.
#Se crean 100 modelos y se escoge el mejor.
modeloAlc,modelo_linealAlc,xAlc,yAlc,DRCMAlc,AleaAlc=
OptimizarModelo(TodosDatosAlc.iloc[:, :-2],TodosDatosAlc.iloc[:, -2],100)
modeloCl,modelo_linealCl,xCl,yCl,DRCMCl,AleaCl=
OptimizarModelo(TodosDatosCl.iloc[:, :-2],TodosDatosCl.iloc[:, -2],100)

#Obtener datos que se usaron para entrenar el modelo óptimo:
x_train, _, y_train, _ = train_test_split(TodosDatosAlc, yAlc,
test_size=0.95,random_state=AleaAlc)
AlcTrain=x_train[['Compuesto', 'Temperatura', 'Presion']].sort_values('Compuesto').reset_index(drop=True)

x_train, _, y_train, _ = train_test_split(TodosDatosCl, yCl,
test_size=0.95,random_state=AleaCl)
ClTrain=x_train[['Compuesto', 'Temperatura', 'Presion']].sort_values('Compuesto').reset_index(drop=True)

#Imprimir los modelos junto con los datos que se utilizaron
print()
print('Modelo Alcanos:')
print(modeloAlc)
print()
print('Entrenado con los siguientes datos:')
print(AlcTrain)
print()
print('Modelo Cloroalcanos:')
print()
print(modeloCl)
print()
print('Entrenado con los siguientes datos:')
print(ClTrain)

#Guardar los datos de entrenamiento como .csv
AlcTrain.to_csv(path + 'Modelos\\DatosTrainAlc.csv')
ClTrain.to_csv(path + 'Modelos\\DatosTrainCl.csv')

#Guardar los modelos como .csv
modeloAlc.to_csv(path + 'Modelos\\ModeloPuroAlc.csv')
modeloCl.to_csv(path + 'Modelos\\ModeloPuroCl.csv')

#Graficar las densidades calculadas vs las densidades experimentales
GraficarModelo(modelo_linealAlc,xAlc,yAlc,'Alcanos',path)
GraficarModelo(modelo_linealCl,xCl,yCl,'Cloroalcanos',path)

#Graficar las densidades de cada compuesto

```

```

#en cada isoterma vs los datos experimentales
def GraficarModelo2(DatosFilas,modelo_lineal):
    #Obtener los compuestos del diccionario
    comps=DatosFilas.keys()
    #Hacer un ciclo para crear una tabla para cada compuesto
    for comp in comps:
        #Extraer la tabla del compuesto
        tabla=DatosFilas[comp].copy()
        #Extraer los atributos (variables independientes)
        atributos = tabla.columns[:-2]
        #Calcular la densidad con el modelo
        tabla['Densidad Calc'] = modelo_lineal.predict(tabla[atributos])

        #Crear una figura
        fig, ax = plt.subplots(figsize=(5,7))
        #Graficar los valores experimentales de densidad (puntos negros)
        ax.plot(tabla['Presion'],tabla['Densidad'],'k.')

        #Usar la presion como el indice de la tabla (para que sea el eje
x)
        tabla.set_index('Presion', inplace=True)
        #Agrupar las densidades calculadas segun la temperatura
(isotermas) y graficar en la misma figura
        tabla.groupby('Temperatura')[['Densidad Calc']].plot(ax=ax)
        #Crear leyenda de temperaturas
        leyenda = [str(temp) + 'K' for temp in
tabla.Temperatura.unique()]
        #Agregar 'Datos Exp.' para los puntos experimentales en la
leyenda
        leyenda = ['Datos Exp.'] + leyenda
        #Insertar leyenda
        plt.legend(leyenda)
        #Insertar titulos para los ejes y para la grafica
        plt.xlabel('Presion, MPa')
        plt.ylabel('Densidad, g/$cm^3$')
        plt.title(comp)
        #Ajustar el rango del eje x
        plt.xlim([0,70])
        #Guardar la grafica
        plt.tight_layout()
        plt.savefig(path + 'Graficas\\Sustancias Puras\\' + comp +
'.png')
        plt.show()

GraficarModelo2(DatosFilasAlc,modelo_linealAlc)
GraficarModelo2(DatosFilasCl,modelo_linealCl)

```

- **Modelo de densidad de mezclas binarias alcano-cloroalcano**

```

# ##### Importar librerias pertinentes
import os
import pickle
import numpy as np
import pandas as pd

```

```

import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from Funciones import *

# ##### Importar las tablas y guardarlas en un diccionario de Python
#
# Las Tablas 19-26 son para las mezclas binarias (guardadas como .txt).
# * _DatosTablas_ = Diccionario con cada tabla de  $\rho$  (guardada bajo
el nombre de la mezcla estudiada)
# * _DatosTablas2_ = Diccionario con cada tabla de  $V_E$  (guardada bajo
el nombre de la mezcla estudiada)
# * _compuestos_ = vector con los nombres de las mezclas de las tablas
(en el mismo orden que las tablas).
# * _tablas_ = vector con los nombres de los .txt.

#Obtener la direccion de la carpeta Modelo Lineal
path=os.path.abspath('').[:-7]

#Crear el diccionario para almacenar las tablas de densidad y VE
DatosTablas={}
DatosTablas2={}

compuestos,tablas=DensVEMezclas() #Funcion de Funciones.py

#Vector de temperaturas estudiadas
Temps=['283.15', '293.15', '298.15', '303.15', '313.15', '323.15']
for tabla,comp in zip(tablas,compuestos):
    nombre=path + 'Tablas\\Mezclas binarias\\' + tabla

DatosTablas[comp]=pd.read_csv(nombre,index_col=0).iloc[:,[0,2,4,6,8,10]]
    DatosTablas[comp].index.name=comp
    DatosTablas[comp].columns=Temps

DatosTablas2[comp]=pd.read_csv(nombre,index_col=0).iloc[:,[1,3,5,7,9,11]]
    DatosTablas2[comp].index.name=comp
    DatosTablas2[comp].columns=Temps

print('Verificar que las tablas se importaron correctamente. Por
ejemplo:')
DatosTablas['n-hexano 1-clorobutano (0.1 MPa)']

# ##### Importar los grupos de contribucion de los compuestos estudiados

GruposContr=pd.read_csv(path + 'Grupos y
PM\\GruposContribucion.csv',index_col=0).fillna(0)
GruposContr

# ##### Organizar los datos para entrenar el modelo
#Funcion para aplanar los datos
def CrearDatosFilas(DatosDict,Variable):
    #DatosDict es el diccionario de las tablas iniciales
    #Variable es 'Densidad' o 'Volumen Exceso'
    #Crear el diccionario para guardar las tablas

```

```

DatosFilas={}
#Obtener las mezclas del diccionario
Mezc=DatosDict.keys()
#Hacer un ciclo para organizar las tablas de cada mezcla
for mez in Mezc:
    #Extraer las fracciones molares (indices de cada tabla)
    X=np.array(DatosDict[mez].index,dtype=float)
    #Extraer las temperaturas (nombres de las columnas)
    T=np.array(DatosDict[mez].columns,dtype=float)
    #Obtener todas las combinaciones de T y X
    Temp,frac=np.meshgrid(T,X)
    #Extraer las densidades o VE
    Var=DatosDict[mez].values

    #Aplanar las matrices y unir las en una sola matriz
    #Agregar X^2

flat=np.column_stack((Var.ravel(),frac.ravel(),frac.ravel()*2,Temp.ravel()
()))
    #Guardar la matriz como un DataFrame
    flat_data=pd.DataFrame(flat,columns=[Variable,'Fraccion
molar','X^2','Temperatura'])
    #Extraer el valor de la presion (del nombre de la tabla)
    P=float(mez.split("(")[1].split(" ")[0])

    #Obtener el numero de filas
    nfilas=flat.shape[0]
    #Agregar columnas para la P, T*P y P^2
    flat_data['Presion']=np.ones(nfilas)*P

flat_data['T*P']=flat_data['Temperatura'].values*flat_data['Presion'].val
ues
    flat_data['P^2']=flat_data['Presion'].values**2

    #Extraer los nombres de los 2 compuestos de la mezcla
    Comp1=mez.split(" ")[0]
    Comp2=mez.split(" ")[1].split(" ")[0]
    #Extraer los grupos de contribucion de cada compuesto
    GC1=GruposContr.loc[Comp1]
    GC2=GruposContr.loc[Comp2]
    #Obtener el numero de grupos de contribucion
    ncols=len(GC1)
    #Calcular los grupos de contribucion de la mezcla
    for GC in GruposContr.columns:
        flat_data[GC]=GC1[GC]*flat_data['Fraccion molar']+GC2[GC]*(1-
flat_data['Fraccion molar'])
    #Guardar el nombre de la mezcla en la ultima columna
    flat_data['Mezcla']=Comp1+' '+Comp2

    #Guardar dataframe en el diccionario
    DatosFilas[mez]=flat_data
return DatosFilas

#Crear los diccionarios con todas las densidades y volúmenes de exceso

```

```

DatosFilas=CrearDatosFilas(DatosTablas,'Densidad')
DatosFilas2=CrearDatosFilas(DatosTablas2,'Volumen Exceso')

#Guardar estos diccionarios como archivos .pickle
with open(path + 'Pickle y CSV\\DatosDensidadM.pickle', 'wb') as handle:
    pickle.dump(DatosFilas, handle, protocol=pickle.HIGHEST_PROTOCOL)
with open(path + 'Pickle y CSV\\DatosVE.pickle', 'wb') as handle:
    pickle.dump(DatosFilas2, handle, protocol=pickle.HIGHEST_PROTOCOL)

print('Un ejemplo de una tabla organizada (n-hexano 1-clorobutano (0.1
MPa)) es:')
DatosFilas['n-hexano 1-clorobutano (0.1 MPa)'].tail()

#Concatenar cada tabla del diccionario para tener una tabla maestra de
densidades.
#Se deben remover las filas que tengan NaN (donde faltaba el valor de la
densidad).
#Se debe resetear el indice para que la enumeracion de las filas no tenga
huecos.
TodosDatos=pd.concat(DatosFilas,axis=0,ignore_index=True).dropna().reset_
index(drop=True) #concatenar tablas y remover NaN

# #### Crear el modelo lineal
#Crear el modelo para las mezclas
#Se crean 200 modelos y se escoge el mejor.
modelo,modelo_lineal,x,y,RMSE,Aleatorio=OptimizarModelo(TodosDatos.iloc[:
,1:-1],TodosDatos.iloc[:,0],200)

#Obtener datos que se usaron para entrenar el modelo óptimo:
x_train,_, y_train,_ = train_test_split(TodosDatos, y,
test_size=0.95,random_state=Aleatorio)
Train=x_train[['Mezcla','Temperatura','Presion','Fraccion
molar']].sort_values('Mezcla').reset_index(drop=True)

#imprimir modelo junto los datos que se utilizaron para crearlo
print('Modelo Mezclas')
print()
print(modelo)
print()
print('Entrenado con los siguientes datos:')
print(Train)

#Guardar los datos de entrenamiento como .csv
Train.to_csv(path + 'Modelos\\DatosTrainMez.csv')

#guardar el modelo
modelo.to_csv(path + 'Modelos\\ModeloMezclas.csv')

# #### Graficar los modelos para verificar su precision
#Graficar las densidades calculadas vs las densidades experimentales
GraficarModelo(modelo_lineal,x,y,'Mezclas',path)

#Graficar las densidades de las mezclas a 30 MPa
#para cada isoterma vs los datos experimentales

```

```

def GraficarModelo2(DatosFilas,modelo_lineal):
    #Hacer un ciclo para crear una tabla para cada compuesto
    for key in compuestos[4::11]:
        #Extraer la tabla de la mezcla
        tabla=DatosFilas[key].copy()
        #Extraer los atributos (variables independientes)
        atributos = tabla.columns[1:-1]
        #Calcular la densidad con el modelo
        tabla['Densidad Calc'] = modelo_lineal.predict(tabla[atributos])
#aplicar el modelo para predecir densidad

        #Crear una figura
        fig, ax = plt.subplots(figsize=(5,7)) #crear figura
        #Graficar los valores experimentales de densidad (puntos negros)
        ax.plot(tabla['Fraccion molar'],tabla['Densidad'],'k.') #graficar
densidad

        #Usar la fraccion molar como el indice de la tabla (para que sea
el eje x)
        tabla.set_index('Fraccion molar', inplace=True)
        #Agrupar las densidades calculadas segun la temperatura
(isotermas) y graficar en la misma figura
        tabla.groupby('Temperatura')[['Densidad Calc']].plot(ax=ax)
        #Crear leyenda de temperaturas
        leyenda = [str(temp) + 'K' for temp in
tabla.Temperatura.unique()]
        #Agregar 'Datos Exp.' para los puntos experimentales en la
leyenda
        leyenda = ['Datos Exp.'] + leyenda
        #Insertar leyenda
        plt.legend(leyenda)
        #Insertar titulos para los ejes y para la grafica
        plt.xlabel('Fraccion molar, x1')
        plt.ylabel('Densidad, g/$cm^3$')
        plt.title(key)
        #Ajustar el rango del eje x
        plt.xlim([0,1])
        #Guardar la grafica
        plt.tight_layout()
        plt.savefig(path + 'Graficas\\Mezclas\\' + key + '.png')
        plt.show()

GraficarModelo2(DatosFilas,modelo_lineal)

```

- **Densidad y volumen de exceso de mezclas**

```

# ##### Importar librerias y tablas pertinentes
import os
import pickle
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

```

```

from scipy.interpolate import interp1d
from Funciones import *

#Obtener la direccion de la carpeta Modelo Lineal
path=os.path.abspath('').[:-7]

#Importar diccionarios de densidad de mezclas, VE y densidades puras
with open(path + 'Pickle y CSV\\DatosDensidadM.pickle', 'rb') as handle:
    DatosDensidadM = pickle.load(handle)
with open(path + 'Pickle y CSV\\DatosVE.pickle', 'rb') as handle:
    DatosVE = pickle.load(handle)
with open(path + 'Pickle y CSV\\DatosDensidadP.pickle', 'rb') as handle:
    DatosDensidadP = pickle.load(handle)

compuestos=MezclasNombres() #funcion de Funciones.py

#Nombres de las mezclas
nombres_tablas=['n-hexano 1-clorobutano','n-hexano 2-clorobutano','n-
hexano iso-clorobutano','n-hexano ter-clorobutano',
heptano 1-clorobutano','n-heptano 2-clorobutano','n-heptano iso-
clorobutano','n-heptano ter-clorobutano']

#Importar tabla de grupos de contribucion
GruposContr=pd.read_csv(path + 'Grupos y
PM\\GruposContribucion.csv',index_col=0).fillna(0)

#Importar modelo de mezclas y ordenar las filas
ModeloMezclas=pd.read_csv(path +
'Modelos\\ModeloMezclas.csv',index_col=0)
ModeloMezclas=ModeloMezclas.reindex(['C1-C2', 'C1-C3', 'C1-C4', 'C2-C2',
'C2-C3', 'C2-C4', 'C3-C3', 'C1-C1',
'C2-C1', 'C3-C1', 'Temperatura', 'Presion', 'T*P', 'P^2','Fraccion
molar','X^2','Intercepto'])

#Importar modelos puros y ordenar las filas
ModeloPuroAlc=pd.read_csv(path +
'Modelos\\ModeloPuroAlc.csv',index_col=0).fillna(0)
ModeloPuroAlc=ModeloPuroAlc.reindex(['C1-C2', 'C1-C3', 'C1-C4', 'C2-C2',
'C2-C3', 'C2-C4', 'C3-C3', 'C1-C1',
'C2-C1', 'C3-C1', 'Temperatura', 'Presion', 'T*P',
'P^2','Intercepto'])

#Importar modelos puros y ordenar las filas
ModeloPuroCl=pd.read_csv(path +
'Modelos\\ModeloPuroCl.csv',index_col=0).fillna(0)
ModeloPuroCl=ModeloPuroCl.reindex(['C1-C2', 'C1-C3', 'C1-C4', 'C2-C2',
'C2-C3', 'C2-C4', 'C3-C3', 'C1-C1',
'C2-C1', 'C3-C1', 'Temperatura', 'Presion', 'T*P',
'P^2','Intercepto'])

#Importar pesos moleculares
PesosMoleculares=pd.read_csv(path + 'Grupos y
PM\\PesosMoleculares.csv',index_col=0).fillna(0)
PesosMoleculares

```

```

# #### Calcular densidades y volúmenes de exceso
# Crear diccionarios vacíos para guardar las densidades y VE de cada
mezcla
DensidadesMaestro={}
VEMaestro={}
# Definir si se usará el modelo de mezclas para las sustancias puras
(True)
# o si se usarán tanto los modelos de puros como el modelo de mezclas
(False)
SoloMezclas=True

# Para cada mezcla del listado de mezclas
for nombre in nombres_tablas:
    # MEZCLAS
    # Crear un dataframe para densidades y otro para VE
    TablaD=pd.DataFrame()
    TablaVE=pd.DataFrame()

    # Extraer información de DatosDensidadM y DatosVE para la mezcla en
todas los valores de presión
    # (concatenar cada tabla que existe para la mezcla)
    for mez in [tbl for tbl in compuestos if nombre in tbl]:

TablaD=pd.concat([TablaD,DatosDensidadM[mez]],axis=0,ignore_index=True).r
eset_index(drop=True)
        TablaVE=pd.concat([TablaVE,DatosVE[mez]['Volumen
Exceso']],axis=0,ignore_index=True).reset_index(drop=True)

        # Ordenar las columnas de la tabla de densidades para calcular
densidades
        Tabla=TablaD[['C1-C2', 'C1-C3', 'C1-C4', 'C2-C2', 'C2-C3', 'C2-C4',
'C3-C3', 'C1-C1',
        'C2-C1', 'C3-C1', 'Temperatura', 'Presion', 'T*P', 'P^2','Fraccion
molar','X^2']].copy()
        # Extraer el número de filas
        nfilas=Tabla.shape[0]
        # Agregar columna para el intercepto (1)
        Tabla['Intercepto']=np.ones(nfilas)
        # Calcular densidades y guardar en nueva columna
        Tabla['Densidad mezcla']=Tabla.values @ ModeloMezclas.values
        # Guardar densidades experimentales en nueva columna
        Tabla['Densidad mezcla real']=TablaD['Densidad']

# SUSTANCIAS PURAS
# Extraer nombres, grupos de contribución y pesos moleculares de los
compuestos
Comp1=nombre.split(" ")[0]
Comp2=nombre.split(" ")[1]
GC1=GruposContr.loc[Comp1]
GC2=GruposContr.loc[Comp2]
M1=PesosMoleculares.loc[Comp1]
M2=PesosMoleculares.loc[Comp2]

```

```

#Copiar T, P, T*P, P^2 y crear tablas para cada compuesto
Tabla_comp1=Tabla[['Temperatura','Presion','T*P','P^2','Intercepto']].copy()

Tabla_comp2=Tabla[['Temperatura','Presion','T*P','P^2','Intercepto']].copy()

#Obtener el numero de grupos de contribucion
ncols=GruposContr.columns.shape[0]

#SUSTANCIA PURA 1
#Copiar el grupo de contribucion n veces y reorganizar en una matriz
copy_groups=np.tile(GC1,nfilas)
groups_data=np.resize(copy_groups,[nfilas,ncols])
#Guardar la matriz en una tabla
groups_data=pd.DataFrame(groups_data,columns=GruposContr.columns)

#Unir la tabla del compuesto 1 con la de los grupos de contribucion
Tabla_comp1=pd.concat([groups_data,Tabla_comp1],axis=1)

#Calcular la densidad del compuesto 1
if SoloMezclas: #Calcular con el modelo de mezclas (para que VE = 0
en x1=0 o x1=1)
    Tabla_comp1['Fraccion molar']=1
    Tabla_comp1['X^2']=1
    Tabla_comp1=Tabla_comp1[['C1-C2', 'C1-C3', 'C1-C4', 'C2-C2', 'C2-
C3', 'C2-C4', 'C3-C3', 'C1-C1',
    'C2-C1', 'C3-C1', 'Temperatura', 'Presion', 'T*P',
    'P^2','Fraccion molar','X^2','Intercepto']]
    Tabla_comp1['Densidad 1']=Tabla_comp1.values @
ModeloMezclas.values
else: #Calcular con el modelo de alcanos
    Tabla_comp1['Densidad 1']=Tabla_comp1.values @
ModeloPuroAlc.values

#Agrega columnas para la densidad y volumen real del compuesto 1
#Se busca la densidad de DatosDensidadP donde coincidan el compuesto,
T y P
Tabla_comp1['Densidad 1
real']=pd.merge(Tabla_comp1,DatosDensidadP[Comp1],on=['Temperatura','Pres
ion'],how='left')['Densidad']
Tabla_comp1['Volumen 1 real']=M1.values/Tabla_comp1['Densidad 1
real'].values
#Calcular el volumen del compuesto 1
Tabla_comp1['Volumen 1']=M1.values/Tabla_comp1['Densidad 1'].values

#SUSTANCIA PURA 2
#Copiar el grupo de contribucion n veces y reorganizar en una matriz
copy_groups=np.tile(GC2,nfilas)
groups_data=np.resize(copy_groups,[nfilas,ncols])
#Guardar la matriz en una tabla
groups_data=pd.DataFrame(groups_data,columns=GruposContr.columns)

```

```

#Unir la tabla del compuesto 1 con la de los grupos de contribucion
Tabla_comp2=pd.concat([groups_data,Tabla_comp2],axis=1)

#Calcular la densidad del compuesto 2
if SoloMezclas: #Calcular con el modelo de mezclas (para que VE = 0
en x1=0 o x1=1)
    Tabla_comp2['Fraccion molar']=0
    Tabla_comp2['X^2']=0
    Tabla_comp2=Tabla_comp2[['C1-C2', 'C1-C3', 'C1-C4', 'C2-C2', 'C2-
C3', 'C2-C4', 'C3-C3', 'C1-C1',
    'C2-C1', 'C3-C1', 'Temperatura', 'Presion', 'T*P',
    'P^2','Fraccion molar','X^2','Intercepto']]
    Tabla_comp2['Densidad 2']=Tabla_comp2.values @
ModeloMezclas.values #ModeloPuroCl.values
else: #Calcular con el modelo de cloroalcanos
    Tabla_comp2['Densidad 2']=Tabla_comp2.values @
ModeloPuroCl.values

#Agrega columnas para la densidad y volumen real del compuesto 2
#Se busca la densidad de DatosDensidadP donde coincidan el compuesto,
T y P
Tabla_comp2['Densidad 2
real']=pd.merge(Tabla_comp2,DatosDensidadP[Comp2],on=['Temperatura','Pres
ion'],how='left')['Densidad']
Tabla_comp2['Volumen 2 real']=M2.values/Tabla_comp2['Densidad 2
real'].values
#Calcular el volumen del compuesto 2
Tabla_comp2['Volumen 2']=M2.values/Tabla_comp2['Densidad 2'].values

#VOLUMEN MOLAR DE MEZCLAS
#Extraer fracciones molares
X1=Tabla['Fraccion molar'].values
X2=1-Tabla['Fraccion molar'].values
#calcular el volumen de la mezcla (basada en los modelos)
Tabla['Volumen mezcla']=(M1.values*X1+M2.values*X2)/Tabla['Densidad
mezcla'].values
#Calcular el volumen de la mezcla ideal (basada en las densidades
experimentales)
Tabla['Volumen mezcla ideal']=X1*Tabla_comp1['Volumen 1
real'].values+X2*Tabla_comp2['Volumen 2 real'].values
#Calcular el volumen de la mezcla real
Tabla['Volumen mezcla
real']=(M1.values*X1+M2.values*X2)/Tabla['Densidad mezcla real'].values
#Calcular el VE (basado en los modelos)
Tabla['VE calc']=Tabla['Volumen mezcla']-X1*Tabla_comp1['Volumen
1'].values-X2*Tabla_comp2['Volumen 2'].values
#Extraer el VE real (de las tablas) y calcularlo a partir de las
densidades experimentales
Tabla['VE real']=TablaVE.values
Tabla['VE real2']=Tabla['Volumen mezcla real']-Tabla['Fraccion
molar']*Tabla_comp1['Volumen 1 real']-(1-Tabla['Fraccion
molar'])*Tabla_comp2['Volumen 2 real']

```

```

#UNIR TODO
#Concatenar las tablas y separar en densidades y volúmenes
Densidades=pd.concat([Tabla[['Temperatura','Presion','Fraccion
molar','Densidad mezcla','Densidad mezcla real']],
Tabla_comp1[['Densidad 1','Densidad 1 real']],
Tabla_comp2[['Densidad 2','Densidad 2 real']],axis=1)
Densidades['Mezcla']=Comp1+' '+Comp2
VolumenExceso=pd.concat([Tabla[['Temperatura','Presion','Fraccion
molar','VE calc','VE real','VE real2','Volumen mezcla',
'Volumen mezcla ideal','Volumen
mezcla real']],\
                        Tabla_comp1[['Volumen 1','Volumen 1 real']],\
                        Tabla_comp2[['Volumen 2','Volumen 2
real']]],axis=1)
VolumenExceso['Mezcla']=Comp1+' '+Comp2

#Calcular la desviación para cada mezcla
if SoloMezclas:
    DesvMezcla=fDesviacion(VolumenExceso['VE real'],VolumenExceso['VE
calc'],13)
else:
    DesvMezcla=fDesviacion(VolumenExceso['VE real'],VolumenExceso['VE
calc'],36)
print('La desviación de la mezcla %s es: %.4f' % (Comp1+'
'+Comp2,DesvMezcla))
#GUARDAR EN EL DICcionario
DensidadesMaestro[nombre]=Densidades
VEMaestro[nombre]=VolumenExceso

#Concatenar los diccionarios
Todas_densidades=pd.concat(DensidadesMaestro,axis=0,ignore_index=True).re
set_index(drop=True)
Todas_VE=pd.concat(VEMaestro,axis=0,ignore_index=True).reset_index(drop=T
rue)

#Guardar tablas de densidades y VE como .csv
Todas_densidades.to_csv(path + 'Pickle y CSV\\DensidadesMaestro.csv')
Todas_VE.to_csv(path + 'Pickle y CSV\\VEMaestro.csv')

Todas_densidades.tail()

Todas_VE.head()

# #### Graficar resultados
#Graficar el VE calculado vs el VE obtenido experimentalmente
#Graficar cada mezcla con un color diferente
for mezcla in Todas_VE.Mezcla.unique():
    plt.plot(Todas_VE[Todas_VE.Mezcla==mezcla]['VE
real'],Todas_VE[Todas_VE.Mezcla==mezcla]['VE calc'],'.',markersize=3)
#Graficar línea y=x
plt.plot([0,0.6],[0,0.6],'k-')
#insertar títulos en los ejes y leyendas
plt.xlabel('VE real')
plt.ylabel('VE calculado')

```

```

lgd=plt.legend(list(Todas_VE.Mezcla.unique())+['y=x'], bbox_to_anchor=(1,
1))
plt.savefig(path + 'Graficas\\Propiedades Exceso\Volumen
Exceso.png',bbox_extra_artists=(lgd,), bbox_inches='tight')
plt.show()

#Comparar el volumen de mezcla real con el ideal (asumiendo que VE=0)
x=Todas_VE['Volumen mezcla real']
y=Todas_VE['Volumen mezcla ideal']
plt.plot(x,y, '.')
plt.plot([x.min(),x.max()], [x.min(),x.max()])
plt.xlabel('Volumen Mezcla, $cm^3$/mol')
plt.ylabel('$x_{1V_{m1}}+x_{1V_{m1}}$, $cm^3$/mol')
plt.legend(['VE','y=x'])
plt.savefig(path + 'Graficas\\Propiedades Exceso\Volumen Exceso 2.png')
plt.show()

#Calcular la desviacion y el DRCM
DM=fDesviacion(x,y,36)
print('La desviacion total del volumen de las mezclas es: %.4f' % DM)

DCM=fDRCM(x,y)
print('El DRCM del volumen de las mezclas es: %.4f%%' % DCM)

#Graficar los Volumenes de exceso calculados junto con los obtenidos
experimentalmente
#Se hace la grafica para T=303.15K y P=0.1 y 65 MPa
for temp,press in [(303.15,0.1),(303.15,65)]:
    #Graficar las mezclas de hexano y luego las mezclas de heptano
    for comp1 in ['n-hexano','n-heptano']:
        #Crear figura
        fig, ax = plt.subplots(figsize=(5,7))
        #Graficar VE calculadas
        for nombre in [nom for nom in nombres_tablas if comp1 in nom]:
            #Filtrar la tabla de VE para la mezcla a la T y P deseadas
            VE_datos=Todas_VE[(Todas_VE['Mezcla']==nombre) &
(Todas_VE['Temperatura']==temp) & (Todas_VE['Presion']==press)].copy()
            #Agregar el VE en x1=0 y x1=1 (debe ser 0)
            last_row=VE_datos.shape[0]
            VE_datos.loc[last_row+1,'Fraccion molar']=0 #agregar x1=0
            VE_datos.loc[last_row+1,'VE calc']=0 #agregar VE=0
            VE_datos.loc[last_row+2,'Fraccion molar']=1 #agregar x1=1
            VE_datos.loc[last_row+2,'VE calc']=0 #agregar VE=0
            #Ordenar segun la fraccion molar
            VE_datos=VE_datos.sort_values(by='Fraccion molar')

            #Realizar una interpolacion cubica para suavizar la curva a
graficar
            x=VE_datos['Fraccion molar']
            y=VE_datos['VE calc']
            y_smooth=interp1d(x,y,kind='cubic')
            x_smooth=np.linspace(x.min(),x.max())

            #Graficar curva suavizada

```

```

ax.plot(x_smooth,y_smooth(x_smooth))

#Resetear el ciclo de colores para que los datos obtenidos
experimentalmente coincidan en los colores
plt.gca().set_prop_cycle(None)

#Graficar VE obtenidas experimentalmente
for nombre in [nom for nom in nombres_tablas if compl in nom]:
    #Filtrar la tabla de VE para la mezcla a la T y P deseadas
    VE_datos=Todas_VE[(Todas_VE['Mezcla']==nombre) &
(Todas_VE['Temperatura']==temp) & (Todas_VE['Presion']==press)].copy()
    #Graficar datos reales
    x=VE_datos['Fraccion molar']
    y_real=VE_datos['VE real']
    ax.plot(x,y_real, '.')

#Insertar titulos en los ejes y en la figura
plt.xlabel('Fraccion molar, $x_1$')
plt.ylabel('$V^E$, $cm^3/mol$')
plt.title(compl+' : '+str(temp)+'K'+ ' '+str(press)+'MPa')
#Fijar el rango para del eje x
plt.xlim([0,1])
#Insertar la leyenda y ponerla fuera del grafico
#ley0 son los calculados, ley1 son los experimentales
ley0 = ['Calc. ' + nom.split(" ")[1] for nom in nombres_tablas if
compl in nom]
ley1 = ['Exp. ' + nom.split(" ")[1] for nom in nombres_tablas if
compl in nom]
leyenda=ley0 + ley1
lgd=ax.legend(leyenda, bbox_to_anchor=(1, 1))
#Guardar la grafica
fig.savefig(path + 'Graficas\\Propiedades Exceso\\ ' + compl + '
+ cloroalcanos ' +str(temp)+'K'+ ' '+str(press)+'MPa.png',
            bbox_extra_artists=(lgd,), bbox_inches='tight')
plt.show()

```

- **Validación de los modelos**

```

# #### Importar librerias y tablas pertinentes
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from thermo.chemical import Chemical
from thermo.mixture import Mixture
from scipy.optimize import fsolve
from Funciones import *

#Obtener la direccion de la carpeta Modelo Lineal
path=os.path.abspath('')[:-7]

#Importar modelos puros

```

```

ModeloPuroAlc=pd.read_csv(path +
'Modelos\\ModeloPuroAlc.csv',index_col=0)
ModeloPuroAlc=ModeloPuroAlc.reindex(['C1-C2', 'C1-C3', 'C1-C4', 'C2-C2',
'C2-C3', 'C2-C4', 'C3-C3', 'C1-C1',
'C2-C1', 'C3-C1', 'Temperatura', 'Presion', 'T*P',
'P^2','Intercepto'])

ModeloPuroCl=pd.read_csv(path + 'Modelos\\ModeloPuroCl.csv',index_col=0)
ModeloPuroCl=ModeloPuroCl.reindex(['C1-C2', 'C1-C3', 'C1-C4', 'C2-C2',
'C2-C3', 'C2-C4', 'C3-C3', 'C1-C1',
'C2-C1', 'C3-C1', 'Temperatura', 'Presion', 'T*P',
'P^2','Intercepto'])

#Importar modelo de mezclas
ModeloMezclas=pd.read_csv(path +
'Modelos\\ModeloMezclas.csv',index_col=0)
ModeloMezclas=ModeloMezclas.reindex(['C1-C2', 'C1-C3', 'C1-C4', 'C2-C2',
'C2-C3', 'C2-C4', 'C3-C3', 'C1-C1',
'C2-C1', 'C3-C1', 'Temperatura', 'Presion', 'T*P', 'P^2','Fraccion
molar','X^2','Intercepto'])

#Importar los grupos de contribucion
GruposContr2=pd.read_csv(path + 'Grupos y
PM\\GruposContribucion2.csv',index_col=0).fillna(0)
#Hacer una copia para 1-cloropropano para los datos de 1-cloropropano.csv
GruposContr2.loc['1-cloropropano ',:]=GruposContr2.loc['1-
cloropropano',:]
GruposContr2

# #### Importar datos de literatura
#Importar los datos de literatura de sustancias puras
OtrosCompuestos0=pd.read_csv(path + 'Datos
literatura\\OtrosCompuestos.csv',index_col=0)
print('Numero de datos: ' + str(OtrosCompuestos0.shape[0]))
OtrosCompuestos0.head()

#Importar 1-cloropropano.csv
cloroprop=pd.read_csv(path + 'Datos literatura\\1-
cloropropano.csv',index_col=0)
cloroprop.head()

#Importar datos de mezclas
OtrasMezclas=pd.read_csv(path + 'Datos
literatura\\OtrasMezclas.csv').fillna(0)
print('Numero de datos: ' + str(OtrasMezclas.shape[0]))
OtrasMezclas.head()

# ### Organizar datos de literatura
#Organizar datos de 1-cloropropano.csv en filas
#Leer las presiones
pres=np.array(cloroprop.index,dtype=float)
#Leer las temperaturas
temp=np.array(cloroprop.columns,dtype=float)
#Sacar las combinaciones de T y P

```

```

T,P=np.meshgrid(temp,pres)
#Leer las densidades
rho=cloroprop.values
#Crear un vector con '1-cloropropano '
comp=np.array(['1-cloropropano '])
comparray=np.tile(comp,rho.ravel().shape[0])
#Crear un vector con la fuente de la literatura
fuente=np.array(['Melentev, 2017'])
fuentearray=np.tile(fuente,rho.ravel().shape[0])
#Crear una tabla con las densidades, T y P
CP=np.column_stack((rho.ravel(),T.ravel(),P.ravel()))
#Guardar en un dataframe con el nombre del compuesto como el indice y las
mismas columnas de OtrosCompuestos0
CPdataframe=pd.DataFrame(CP,index=comparray,columns=OtrosCompuestos0.colu
ms[:-1])
#Agregar columna con la fuente
CPdataframe['Fuente']=fuentearray

#Unir CPdataframe con OtrosCompuestos
OtrosCompuestos=pd.concat([OtrosCompuestos0,CPdataframe],axis=0)
print('Numero de datos: ' + str(OtrosCompuestos.shape[0]))
OtrosCompuestos.tail()

# #### Crear tablas con las variables independientes para el modelo
#COMPUESTOS PUROS
#Copiar la tabla OtrosCompuestos sin incluir la fuente
Datos=OtrosCompuestos[['Densidad','Temperatura','Presion']].copy()
#Obtener los nombres de cada fila (compuestos)
comps=Datos.index
#Insertar los grupos de contribucion para cada compuesto (cada fila)
Datos[GruposContr2.columns.values]=GruposContr2.loc[comps]
#Extraer los nombres de las columnas de la tabla Datos
cols=Datos.columns.tolist()
#Reorganizar orden de las columnas para coincidir con el orden de los
parametros del modelo
Datos=Datos[[cols[0]]+cols[3:]+cols[1:3]]
#Insertar columnas con T*P, P^2 y el intercepto (una columna de unos)
Datos['T*P']=Datos['Temperatura'].values*Datos['Presion'].values
Datos['P^2']=Datos['Presion'].values**2
Datos['Intercepto']=np.ones(Datos.shape[0])
#Mostrar las ultimas 5 filas de Datos
Datos.tail()

#Separar los cloroalcanos de los alcanos
compuestos_clorindados=['cloro' in comp for comp in Datos.index.values]
Cloroalcanos=Datos[compuestos_clorindados]
compuestos_alcanos=['cloro' not in comp for comp in Datos.index.values]
Alcanos=Datos[compuestos_alcanos]
#Mostrar las ultimas 5 filas de los cloroalcanos
Cloroalcanos.tail()

#Mostrar las ultimas 5 filas de los alcanos
Alcanos.tail()

```

```

#MEZCLAS
#Copiar la tabla OtrosCompuestos sin incluir la fuente
DatosM=OtrasMezclas[['Densidad','Fraccion
molar','Temperatura','Presion','Mezcla']].copy()
#Agregar columnas par los grupos de contribucion e inicializar con zeros
DatosM[GruposContr2.columns.values]=pd.DataFrame(np.zeros([DatosM.shape[0]
],len(GruposContr2.columns)))
#Calcular los grupos de contribucion para cada mezcla (cada fila)
for i in range(DatosM.shape[0]):
    #Extraer los nombres de los compuestos que forman la mezcla
    comp1=OtrasMezclas.loc[i,'Mezcla'].split(" ")[0]
    comp2=OtrasMezclas.loc[i,'Mezcla'].split(" ")[1]
    #Extraer las fracciones molares de cada mezcla
    x1=OtrasMezclas.loc[i,'Fraccion molar']
    x2=1-x1
    #Calcular los grupos de contribucion de cada mezcla

DatosM.loc[i,GruposContr2.columns.values]=x1*GruposContr2.loc[comp1].valu
es+x2*GruposContr2.loc[comp2].values
#Agregar las columnas T*P, P^2, X^2 y el intercepto (una columna de unos)
DatosM['T*P']=DatosM['Temperatura'].values*DatosM['Presion'].values
DatosM['P^2']=DatosM['Presion'].values**2
DatosM['X^2']=DatosM['Fraccion molar']**2
DatosM['Intercepto']=np.ones(DatosM.shape[0])
#Reorganizar las columnas de DatosM para que esten en el mismo orden que
los parametros del modelo
DatosM=DatosM[['Densidad','C1-C2', 'C1-C3', 'C1-C4', 'C2-C2', 'C2-C3',
'C2-C4', 'C3-C3', 'C1-C1', 'C2-C1', 'C3-C1', 'Temperatura',
'Presion', 'T*P', 'P^2','Fraccion molar','X^2','Intercepto','Mezcla']]
#Mostrar las ultimas 5 filas de DatosM
DatosM.tail()

# #### Calcular las densidades usando los modelos
#COMPUESTOS PUROS
#Crear una copia de las tablas Cloroalcanos y Alcanos para mostrar T, P,
rho real, rho calculado
Cloroalcanos2=Cloroalcanos[['Temperatura','Presion','Densidad']].copy()
Alcanos2=Alcanos[['Temperatura','Presion','Densidad']].copy()
#Calcular densidades de cloroalcanos y alcanos usando los modelos
respectivos.
#Agregar resultados a las tablas respectivas.
Cloroalcanos2['Densidad calc']=Cloroalcanos.iloc[:,1:].values @
ModeloPuroCl
Alcanos2['Densidad calc']=Alcanos.iloc[:,1:].values @ ModeloPuroAlc
#Unir las dos tablas de alcanos y cloroalcanos
Puros=pd.concat([Cloroalcanos2,Alcanos2],axis=0)

#MEZCLAS
#Crear una copia de la tabla DatosM para mostrar nombre de mezcla, x1, T,
P, rho real, rho calculado
Mezclas=DatosM[['Mezcla','Fraccion
molar','Temperatura','Presion','Densidad']].copy()
#Calcular densidades de las mezclas y agregar a la tabla Mezclas
Mezclas['Densidad calc']=DatosM.iloc[:,1:-1].values @ ModeloMezclas

```

```

#Mostrar resultados de los alcanos y cloroalcanos
Puros.tail()

#Mostrar resultados de las mezclas
Mezclas.tail()

# ##### Calcular densidades de sustancias puras usando Peng-Robinson y
COSTALD
#Separar los compuestos en compuestos donde varia la presion, donde varia
la temperatura y donde estan a condiciones ambiente
compuestosP=['n-pentano','n-octano','2,2,4-
trimetilpentano','ciclooctano','n-decano','1-cloropropano ']
compuestosT=['1-cloropropano','1-clorobutano','1-cloropentano','1-
cloroheptano','1-clorooctano','1-clorodecano','1-clorotetradecano']
compuestosAmb=['1,4-diclorobutano', '1,6-diclorohexano', 'isopentano',
'n-nonano', 'n-undecano', 'n-dodecano', 'n-tridecano']
#Todos los compuestos
TodosCompuestos=compuestosP + compuestosT + compuestosAmb
#Nombres en ingles (para buscar en el paquete de thermo)
AllComponents=['n-pentane','n-octane','isooctane','cyclooctane','n-
decane','1-chloropropane ', '1-chloropropane','1-chlorobutane',
'1-chloropentane','1-chloroheptane','1-chlorooctane','1-
chlorodecane','1-chlorotetradecane','1,4-dichlorobutane',
'1,6-dichlorohexane', 'isopentane', 'n-nonane',
'undecane', 'n-dodecane', 'n-tridecane']

#Modelo Peng Robinson
def PengRobinson(V,*params):
    P,T,Pc,Tc,omega=params
    R = 8.3145
    a = 0.45724*R**2*Tc**2/Pc
    b = 0.0778*R*Tc/Pc
    kappa = 0.37464+1.54226*omega-0.26992*omega**2
    Tr = T/Tc
    alpha = (1+kappa*(1-Tr**0.5))**2
    PR = R*T/(V-b) - alpha*a/(V**2+2*b*V-b**2)
    return P-PR

#Hallar densidad de los liquidos puros
#Agregar nueva columna a Puros
# Puros['SRK']=np.zeros(Puros.shape[0])
# from thermo.eos import PR,SRK
Puros['COSTALD']=np.ones(Puros.shape[0])*np.nan
Puros['Peng Robinson']=np.zeros(Puros.shape[0])
for i in range(Puros.shape[0]):
    #Extraer el compuesto (nombre de la fila de la tabla Puros)
    #Coincidir el nombre en espanol con el nombre en ingles
    (AllComponents)
    comp_ind=TodosCompuestos.index(Puros.index[i])
    comp=Chemical(AllComponents[comp_ind])
    #Extraer T y P
    T=Puros.iloc[i,0]
    P=Puros.iloc[i,1]*1e6

```

```

#Hallar propiedades del compuesto
Tc=comp.Tc
Pc=comp.Pc
omega=comp.omega
MW=comp.MW
#Calcular valor inicial para liquidos (1.1*b)
R = 8.3145
b = 0.0778*R*Tc/Pc
V0=1.1*b
#Hallar volumen molar del liquido
Vm=fsolve(PengRobinson,V0,args=(P,T,Pc,Tc,omega))
#Calcular la densidad (g/cc)
rho=MW/Vm/1e6
#Guardar en la ultima columna de Puros
# comp.set_eos(T=T,P=P,eos=SRK)
# comp.VolumeLiquid.set_user_methods_P('EOS')
# comp.calculate(T=T,P=P)
if T<=Tc:
    Puros.iloc[i,-
2]=comp.MW/comp.VolumeLiquid.calculate_P(P=P,T=T,method='COSTALD_COMPRESS
ED')/1e6 #comp.rhol/1e3
    Puros.iloc[i,-1]=rho

#Mostrar nueva tabla de densidades puras
Puros.head()

# #### Calcular densidades de las mezclas usando COSTALD
#Crear una nueva columna vacia
Mezclas['COSTALD']=np.ones(Mezclas.shape[0])*np.nan
#Para cada fila de la tabla de Mezclas, calcular la densidad
for i in range(Mezclas.shape[0]):
    #Extraer nombre de la mezcla
    mezcla=Mezclas.loc[i,'Mezcla']
    #Extraer los compuestos de la mezcla
    comp1=mezcla.split(" ")[0]
    comp2=mezcla.split(" ")[1]
    #Traducir los nombres a ingles para buscar en thermo
    comp1_ind=TodosCompuestos.index(comp1)
    comp2_ind=TodosCompuestos.index(comp2)
    c1=AllComponents[comp1_ind]
    c2=AllComponents[comp2_ind]
    #Extraer la fraccion molar
    x1=Mezclas['Fraccion molar'].values[i]
    x2=1-x1
    #Crear mezcla
    mez=Mixture([c1,c2],zs=[x1,x2],T=298.15,P=101325)
    #Calcular densidad

rho=mez.MW/mez.VolumeLiquidMixture.calculate(P=101325,T=298.15,zs=mez.zs,
ws=mez.ws,method='COSTALD mixture')/1e6
    #Guardar resultado
    Mezclas.loc[i,'COSTALD']=rho

#Mostrar tabla de mezclas

```

```

Mezclas.head()

# #### Graficar los resultados
#COMPUESTOS PUROS
#Graficar cada compuesto
for comp in TodosCompuestos:
    #Filtrar los datos segun el compuesto
    DatosComp=Puros[Puros.index==comp]
    #Calcular DRCM para el modelo y para PR
    y=DatosComp['Densidad']
    y_pred=DatosComp['Densidad calc']
    y_pred2=DatosComp['Peng Robinson']
    y_pred3=DatosComp['COSTALD']
    DRCMval=fDRCM(y,y_pred)
    DRCMval2=fDRCM(y,y_pred2)
    DRCMval3=fDRCM(y,y_pred3)
    #Calcular la desviacion (cloroalcanos = modelo con 11 pametros;
alcanos = 12 parametros)
    if comp in compuestos_clorindados:
        r=11
    else:
        r=12
    if len(y)>r:
        Desv=fDesviacion(y,y_pred,r)
        Desv2=fDesviacion(y,y_pred2,3)
        Desv3=fDesviacion(y,y_pred3,12) #COSTALD tiene 12 parametros
    else: #Si solo es un punto de densidad, se ignora el r
        Desv=fDesviacion(y,y_pred,0)

    #Graficar los compuestos que estan a condiciones ambiente
    if comp in compuestosAmb:
        #Crear la figura
        fig,ax=plt.subplots(figsize=(4,3))
        #Graficar el valor de literatura, modelo y PR
        plt.bar(0,y)
        plt.bar(1,y_pred)
        plt.bar(2,y_pred2)
        plt.bar(3,y_pred3)
        #Titulos en el eje y y en la grafica
        plt.ylabel('Densidad, g/$cm^3$')
        plt.title(comp+ ' (20°C; 1 atm)')
        #Quitar nombres en eje x

plt.tick_params(axis='x',which='both',bottom=False,top=False,labelbottom=
False)
    #Insertar leyenda
    lgd=ax.legend(['Literatura','Modelo (%.1f%% error)' %
DRCMval,'Peng Robinson (%.1f%% error)' % DRCMval2,
                'COSTALD (%.1f%% error)' %
DRCMval3],bbox_to_anchor=(1, 1))
    #Guardar y mostrar grafico
    fig.savefig(path + 'Graficas\\Otros\\' + comp + '.png',
                bbox_extra_artists=(lgd,), bbox_inches='tight')
    plt.show()

```

```

#Copiar la tabla de datos del compuesto para las graficas
tabla=DatosComp.copy()
#Graficar los compuestos en los que varia la presion
if comp in compuestosP:
    #Crear la figura con 2 graficos
    fig, (ax1,ax2,ax3) = plt.subplots(nrows=1,ncols=3,figsize=(13,7))

    #Fijar la presion como el indice (para que sea el eje x)
    tabla.set_index('Presion', inplace=True)

    #Agrupar los datos segun la temperatura (isotermas) y graficar
    las densidades de literatura
    tabla.groupby('Temperatura')[['Densidad']].plot(ax=ax1,style='*')
    #Resetear el ciclo de colores para mostrar los datos calculados
    con los mismos colores
    ax1.set_prop_cycle(None)
    #Agrupar los datos segun la temperatura (isotermas) y graficar
    las densidades calculadas
    tabla.groupby('Temperatura')[['Densidad
    calc']].plot(ax=ax1,style='.-')

    #Agrupar los datos segun la temperatura (isotermas) y graficar
    las densidades de literatura
    tabla.groupby('Temperatura')[['Densidad']].plot(ax=ax2,style='*')
    #Resetear el ciclo de colores para mostrar los datos calculados
    con los mismos colores
    ax2.set_prop_cycle(None)
    #Agrupar los datos segun la temperatura (isotermas) y graficar
    las densidades de PR
    tabla.groupby('Temperatura')[['Peng
    Robinson']].plot(ax=ax2,style='.-')

    #Agrupar los datos segun la temperatura (isotermas) y graficar
    las densidades de literatura
    tabla.groupby('Temperatura')[['Densidad']].plot(ax=ax3,style='*')
    #Resetear el ciclo de colores para mostrar los datos calculados
    con los mismos colores
    ax3.set_prop_cycle(None)
    #Agrupar los datos segun la temperatura (isotermas) y graficar
    las densidades de COSTALD
    tabla.groupby('Temperatura')[['COSTALD']].plot(ax=ax3,style='.-
    ')

    #Crear la leyenda para cada isoterma de la literatura y calculada
    con el modelo
    leyenda0 = ['Lit. ' + str(round(temp,2)) + 'K' for temp in
    tabla.Temperatura.unique()]
    leyenda1 = ['Mod. ' + str(round(temp,2)) + 'K' for temp in
    tabla.Temperatura.unique()]
    leyenda2 = ['PR. ' + str(round(temp,2)) + 'K' for temp in
    tabla.Temperatura.unique()]
    leyenda3 = ['COST. ' + str(round(temp,2)) + 'K' for temp in
    tabla.Temperatura.unique()]
    ax1.legend(leyenda0 + leyenda1)

```

```

ax2.legend(leyenda0 + leyenda2)
ax3.legend(leyenda0 + leyenda3)
#Insertar titulos en los ejes y para la grafica donde se muestre
el compuesto, DRCM y desviacion
ax1.set_xlabel('Presion, MPa')
ax1.set_ylabel('Densidad, g/$cm^3$')
ax2.set_xlabel('Presion, MPa')
ax2.set_ylabel('Densidad, g/$cm^3$')
ax3.set_xlabel('Presion, MPa')
ax3.set_ylabel('Densidad, g/$cm^3$')
ax1.set_title('%s (DRCM = %.4f%%)' % (comp,DRCMval))
ax2.set_title('%s (DRCM = %.4f%%)' % (comp,DRCMval2))
ax3.set_title('%s (DRCM = %.4f%%)' % (comp,DRCMval3))
#Guardar la grafica
plt.tight_layout()
fig.savefig(path + 'Graficas\\Otros\\' + comp + '.png')
plt.show()
#Graficar los compuestos en los que varia la temperatura
if comp in compuestosT:
    #Crear la figura
    fig, ax = plt.subplots(figsize=(5,7))
    #Fijar la temperatura como el indice (para que sea el eje x)
    tabla.set_index('Temperatura', inplace=True)
    #Agrupar los datos segun la presion (curvas isobaricas) y
graficar las densidades de literatura
    tabla.groupby('Presion')[['Densidad']].plot(ax=ax, style='k*')
    #Agrupar los datos segun la presion (curvas isobaricas) y
graficar las densidades calculadas
    tabla.groupby('Presion')[['Densidad calc']].plot(ax=ax, style='b--
')
    #Agrupar los datos segun la presion (curvas isobaricas) y
graficar las densidades de PR
    tabla.groupby('Presion')[['Peng
Robinson']].plot(ax=ax, style='g:')
    #Agrupar los datos segun la presion (curvas isobaricas) y
graficar las densidades de COSTALD
    tabla.groupby('Presion')[['COSTALD']].plot(ax=ax, style='r-.')
    #Crear la leyenda para las densidades de la literatura y las
calculadas con el modelo
    plt.legend(['Literatura', 'Modelo (DRCM = %.2f%%)' % DRCMval, 'Peng
Robinson (DRCM=%.2f%%)' % DRCMval2,
               'COSTALD (DRCM=%.2f%%)' % DRCMval3])
    #Insertar titulos en los ejes y para la grafica donde se muestre
el compuesto, DRCM y desviacion
    plt.xlabel('Temperatura, K')
    plt.ylabel('Densidad, g/$cm^3$')
    plt.title(comp)
    #Guardar la grafica
    plt.tight_layout()
    plt.savefig(path + 'Graficas\\Otros\\' + comp + '.png')
    plt.show()

#MEZCLAS
#Hacer una grafica para cada mezcla

```

```

for mezcla in Mezclas.Mezcla.unique():
    #Filtrar la tabla de mezclas para la mezcla a graficar
    tabla=Mezclas[Mezclas.Mezcla==mezcla].copy()
    #Calcular DRCM
    DRCM=fDRCM(tabla['Densidad'],tabla['Densidad calc'])
    DRCM2=fDRCM(tabla['Densidad'],tabla['COSTALD'])
    #Graficar datos de literatura
    plt.plot(tabla['Fraccion molar'],tabla['Densidad'],'k*')
    #Graficar densidades calculadas
    plt.plot(tabla['Fraccion molar'],tabla['Densidad calc'],'b--')
    plt.plot(tabla['Fraccion molar'],tabla['COSTALD'],'r:')
    plt.xlabel('Fraccion molar $x_1$')
    plt.ylabel('Densidad g/$cm^3$')
    plt.title(mezcla)
    plt.legend(['Literatura','Modelo %.4f%%' % DRCM,'COSTALD %.4f%%' %
DRCM2])
    plt.savefig(path + 'Graficas\\Otros\\' + mezcla + '.png')
    plt.show()

```

- **Cálculo simbólico de propiedades termofísicas**

```

# #### Importar librerías pertinentes
import os
import pickle
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sympy as sp
from sympy import init_printing
from Funciones import *

# #### Calcular densidades, volúmenes y derivadas (simbólicamente)
#Crear variables simbolicas
init_printing()
A, B, C, D, E, F, G, H, p, Te, m1, m2, x1, x2=sp.symbols('A B C D E F G H p T M1 M2 x1
x2')

#Escribir modelo de densidad simbolico (sustancias puras)
Dp=A+B*p+C*p**2+D*Te+E*p*Te+F
#Escribir modelo de densidad simbolico (mezclas)
Dm=A+B*p+C*p**2+D*Te+E*p*Te+F*x1+G*x1**2+H

#Calcular volumen de sustancias puras simbolico
Vp=m1/Dp
#Calcular volumen de mezclas simbolico
Vm=(m1*x1+m2*x2)/Dm

#Derivar el volumen de sustancias puras con respecto a T
dVpdT=sp.diff(Vp,Te)
#Derivar el volumen de mezclas con respecto a T
dVmdT=sp.diff(Vm,Te)

#Derivar el volumen de sustancias puras con respecto a P
dVpdp=sp.diff(Vp,p)

```

```

#Derivar el volumen de mezclas con respecto a P
dVmdp=sp.diff(Vm,p)

# ##### Calcular expansibilidad isobarica ( $\alpha$ ) de sustancias puras y
mezclas (simbólicamente)

#Expansibilidad isobarica (Alpha) de sustancias puras
AlphaS=1/Vp*dVpdT
#Expansibilidad isobarica (Alpha) de mezclas
AlphaM=1/Vm*dVmdT

#Mostrar AlphaS
AlphaS

#Mostrar AlphaM
AlphaM

# ##### Calcular compresibilidad isotermica ( $\kappa$ ) de sustancias puras
y mezclas (simbólicamente)
#Compresibilidad isotermica (kappa) de sustancias puras
KappaS=-1/Vp*dVpdp
#Compresibilidad isotermica (kappa) de mezclas
KappaM=-1/Vm*dVmdp

#Mostrar KappaS
sp.simplify(KappaS)

#Mostrar KappaM
sp.simplify(KappaM)

# ##### Calcular presion interna ( $P_i$ ) de sustancias puras y mezclas
(simbólicamente)

#Presion interna de sustancias puras
PiS=sp.simplify(AlphaS*Te/KappaS)-p
#Presion interna de mezclas
PiM=sp.simplify(AlphaM*Te/KappaM)-p

#Mostrar PiS
PiS

#Mostrar PiM
PiM

```

- **Cálculo de propiedades termofísicas de sustancias puras**

```

# ##### Importar librerias y CSV pertinentes
import os
import pickle
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from Funciones import *

```

```

#Obtener la direccion de la carpeta Modelo Lineal
path=os.path.abspath('').[:-7]

#Importar modelos
ModeloPuroAlc=pd.read_csv(path +
'Modelos\\ModeloPuroAlc.csv',index_col=0)
ModeloPuroAlc=ModeloPuroAlc.reindex(['C1-C2', 'C1-C3', 'C1-C4', 'C2-C2',
'C2-C3', 'C2-C4', 'C3-C3', 'C1-C1',
'C2-C1', 'C3-C1', 'Temperatura','Presion','T*P',
'P^2','Intercepto'])
ModeloPuroCl=pd.read_csv(path + 'Modelos\\ModeloPuroCl.csv',index_col=0)
ModeloPuroCl=ModeloPuroCl.reindex(['C1-C2', 'C1-C3', 'C1-C4', 'C2-C2',
'C2-C3', 'C2-C4', 'C3-C3', 'C1-C1',
'C2-C1', 'C3-C1', 'Temperatura','Presion','T*P',
'P^2','Intercepto'])

#Importar datos puros de densidad
with open(path + 'Pickle y CSV\\DatosDensidadP.pickle', 'rb') as handle:
    DatosDensidadP = pickle.load(handle)

# ##### Importar valores de  $\alpha$ ,  $\kappa$  y  $P_i$  de sustancias puras
#Obtener la direccion de la carpeta Modelo Lineal
path=os.path.abspath('').[:-7]

#Crear diccionarios para almacenar las tablas con las
#propiedades termofisicas de compuestos puros
DatosTablasAlpha={}
DatosTablasKappa={}
DatosTablasPi={}

compuestos,tablasAlpha,tablasKappa,tablasPi=PropCompPuros() #funcion de
Funciones.py

#Importar tablas en los diccionarios
for tabla,comp in zip(tablasAlpha,compuestos):
    DatosTablasAlpha[comp]=pd.read_csv(path + 'Tablas\\Propiedades\\' +
tabla,index_col=0)
    DatosTablasAlpha[comp].index.name=comp

for tabla,comp in zip(tablasKappa,compuestos):
    DatosTablasKappa[comp]=pd.read_csv(path + 'Tablas\\Propiedades\\' +
tabla,index_col=0)
    DatosTablasKappa[comp].index.name=comp

for tabla,comp in zip(tablasPi,compuestos):
    DatosTablasPi[comp]=pd.read_csv(path + 'Tablas\\Propiedades\\' +
tabla,index_col=0)
    DatosTablasPi[comp].index.name=comp

print('Verificar que las tablas se importaron correctamente. Por
ejemplo:')
DatosTablasAlpha['2,2-dimetilbutano'].head()

```

```

print('Verificar que las tablas se importaron correctamente. Por
ejemplo:')
DatosTablasKappa['2,2-dimetilbutano'].head()

print('Verificar que las tablas se importaron correctamente. Por
ejemplo:')
DatosTablasPi['2,2-dimetilbutano'].tail()

# ##### Organizar datos importados de sustancias puras
#Funcion para aplanar los datos
def CrearDatosFilas(DatosDict,Variable):
    #DatosDict es el diccionario de las tablas iniciales
    #Variable es 'Alpha','Kappa' o 'Presion interna'
    #Crear el diccionario para guardar las tablas
    DatosFilas={}
    #Hacer un ciclo para cada compuesto (keys del diccionario)
    for key in DatosDict.keys():
        #Extraer las presiones (indices de cada tabla)
        P=np.array(DatosDict[key].index,dtype=float)
        #Extraer las temperaturas (nombres de columnas de cada tabla)
        T=np.array(DatosDict[key].columns,dtype=float)
        #Obtener todas las combinaciones de T y P
        Temp,Pres=np.meshgrid(T,P)
        #Extraer las Alpha, Kappa o Presion interna
        Var=DatosDict[key].values

        #Aplanar las matrices y unir las en una sola matriz
        flat=np.column_stack((Var.ravel(),Temp.ravel(),Pres.ravel()))
        #Guardar la matriz como un DataFrame

    flat_data=pd.DataFrame(flat,columns=[Variable,'Temperatura','Presion'])
        #Guardar el nombre del compuesto en la ultima columna (llamada
'Compuesto')
        flat_data['Compuesto']=key

        #Guardar la tabla aplanada en el diccionario
        DatosFilas[key]=flat_data
    return DatosFilas

#Crear los diccionarios con todos los valores de alpha, kappa y pi
DatosFilasAlpha=CrearDatosFilas(DatosTablasAlpha,'Alpha')
DatosFilasKappa=CrearDatosFilas(DatosTablasKappa,'Kappa')
DatosFilasPi=CrearDatosFilas(DatosTablasPi,'Presion interna')

#Concatenar las tablas de los diccionarios para tener tablas maestras
TodosDatosAlpha=pd.concat(DatosFilasAlpha,axis=0,ignore_index=True).dropn
a().reset_index(drop=True)
TodosDatosKappa=pd.concat(DatosFilasKappa,axis=0,ignore_index=True).dropn
a().reset_index(drop=True)
TodosDatosPi=pd.concat(DatosFilasPi,axis=0,ignore_index=True).dropna().re
set_index(drop=True)

print('Ejemplo de una tabla maestra:')
TodosDatosPi.head()

```

```

# #### Calcular  $\alpha$  de sustancias puras
#Calcular alpha de las sustancias puros
#Inicializar el vector donde se guardaran los valores de alpha
(inicializar con ceros)
AlphaCal=np.zeros(TodosDatosAlpha.shape[0])
#Para cada fila de TodosDatosAlpha, calcular el alpha usando el modelo
for i in range(TodosDatosAlpha.shape[0]):
    #Extraer el compuesto, T y P de la fila
    compuesto=TodosDatosAlpha.loc[i,'Compuesto']
    temperatura=TodosDatosAlpha.loc[i,'Temperatura']
    presion=TodosDatosAlpha.loc[i,'Presion']
    #Extraer los parametros (variables independietes) a los mismos
valores de T y P
    datos=DatosDensidadP[compuesto]
    datos=datos[(datos['Temperatura']==temperatura) &
(datos['Presion']==presion)]
    #Agregar un 1 al final de los parametros para el intercepto
    parametros=np.append(datos[datos.columns[:-2]].values,[1])

    #Calcular la densidad, e y d segun el tipo de compuesto
    if 'cloro' in compuesto:
        densidad = parametros @ ModeloPuroCl.values
        e=ModeloPuroCl.loc['T*P']
        d=ModeloPuroCl.loc['Temperatura']
    else:
        densidad = parametros @ ModeloPuroAlc.values
        e=ModeloPuroAlc.loc['T*P']
        d=ModeloPuroAlc.loc['Temperatura']

    #Usando e, d, P y densidad, calcular alpha (1/kK)
    AlphaCal[i]=(-presion*e-d)/densidad*1000

#Guardar el vector AlphaCal en una columna de TodosDatosAlpha
TodosDatosAlpha['Alpha Calc']=AlphaCal
TodosDatosAlpha.head()

# #### Calcular  $\kappa$  de sustancias puras
#Calcular kappa de las sustancias puros
#Inicializar el vector donde se guardaran los valores de kappa
(inicializar con ceros)
KappCal=np.zeros(TodosDatosKappa.shape[0])
#Para cada fila de TodosDatosAlpha, calcular el kappa usando el modelo
for i in range(TodosDatosKappa.shape[0]):
    #Extraer el compuesto, T y P de la fila
    compuesto=TodosDatosKappa.loc[i,'Compuesto']
    temperatura=TodosDatosKappa.loc[i,'Temperatura']
    presion=TodosDatosKappa.loc[i,'Presion']
    #Extraer los parametros (variables independietes) a los mismos
valores de T y P
    datos=DatosDensidadP[compuesto]
    datos=datos[(datos['Temperatura']==temperatura) &
(datos['Presion']==presion)]
    #Agregar un 1 al final de los parametros para el intercepto

```

```

parametros=np.append(datos[datos.columns[:-2]].values,[1])

#Calcular la densidad, b, c y e segun el tipo de compuesto
if 'cloro' in compuesto:
    densidad = parametros @ ModeloPuroCl.values
    b=ModeloPuroCl.loc['Presion']
    c=ModeloPuroCl.loc['P^2']
    e=ModeloPuroCl.loc['T*P']
else:
    densidad = parametros @ ModeloPuroAlc.values
    b=ModeloPuroAlc.loc['Presion']
    c=ModeloPuroAlc.loc['P^2']
    e=ModeloPuroAlc.loc['T*P']

#Usando b, c, e, P, T y densidad, calcular kappa (1/TPa)
KappCal[i]=(b+2*c*presion+e*temperatura)/densidad*1e6

#Guardar el vector KappCal en una columna de TodosDatosKappa
TodosDatosKappa['Kappa Calc']=KappCal
TodosDatosKappa.head()

# ##### Calcular $P_i$ de sustancias puras
#Calcular Pi (MPa) de las sustancias puros usando alpha y kappa
TodosDatosPi['Pi Calc']=TodosDatosAlpha['Alpha
Calc']*1000*TodosDatosAlpha['Temperatura']/TodosDatosKappa['Kappa Calc']-
TodosDatosAlpha['Presion']
TodosDatosPi.head()

# ##### Comparar propiedades termofísicas calculadas y reales de
sustancias puras
#Funcion para calcular la desviacion y DRCM para cada compuesto
def CompProp(Tabla):
    #Crear vector de compuestos
    comps=Tabla.Compuesto.unique()
    #Crear una tabla para guardar los resultados

tabla=pd.DataFrame(np.zeros([len(comps),3]),columns=['Compuesto','Desviac
ion','DRCM,%'])

for comp,i in zip(comps,range(len(comps))):
    #Filtrar tabla de datos por compuesto
    Tabla0=Tabla[Tabla.Compuesto==comp].copy()
    #Valores reales (x) y valores calculados (y)
    x=Tabla0.iloc[:,0]
    y=Tabla0.iloc[:,-1]
    #Calcular la desviacion (11 parametros para cloroalcanos; 12 para
alcanos)
    if 'cloro' in comp:
        Desv=fDesviacion(x,y,11)
    else:
        Desv=fDesviacion(x,y,12)
    DRCM=fDRCM(x,y)
    tabla.iloc[i,0]=comp
    tabla.iloc[i,1]=Desv

```

```

        tabla.iloc[i,2]=DRCM
    return tabla.set_index('Compuesto')

#Comparar los valores de alpha (1/kK)
AlphaComp=CompProp(TodosDatosAlpha)
AlphaComp.to_csv(path + 'Pickle y CSV\\Alpha comparacion.csv')
AlphaComp

#Comparar los valores de kappa (1/GPa)
TodosDatosKappa2=TodosDatosKappa.copy()
TodosDatosKappa2['Kappa']=TodosDatosKappa2['Kappa']/1e3
TodosDatosKappa2['Kappa Calc']=TodosDatosKappa2['Kappa Calc']/1e3
KappaComp=CompProp(TodosDatosKappa2)
KappaComp.to_csv(path + 'Pickle y CSV\\Kappa comparacion.csv')
KappaComp

#Comparar los valores de Pi (GPa)
TodosDatosPi2=TodosDatosPi.copy()
TodosDatosPi2['Presion interna']=TodosDatosPi2['Presion interna']/1e3
TodosDatosPi2['Pi Calc']=TodosDatosPi2['Pi Calc']/1e3
PiComp=CompProp(TodosDatosPi2)
PiComp.to_csv(path + 'Pickle y CSV\\Pi comparacion.csv')
PiComp

# #### Graficar propiedades termofísicas calculadas y reales de
sustancias puras

GraficarProp(TodosDatosAlpha,'Alpha, 1/kK',path,'Propiedades')
GraficarProp(TodosDatosKappa,'Kappa, 1/TPa',path,'Propiedades')
GraficarProp(TodosDatosPi,'Pi, MPa',path,'Propiedades')

```

- **Cálculo de propiedades termofísicas de mezclas**

```

# #### Importar librerías y CSV pertinentes
import os
import pickle
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from Funciones import *

#Obtener la dirección de la carpeta Modelo Lineal
path=os.path.abspath('').[:-7]

#Importar densidades y VE
DensidadesMaestro=pd.read_csv(path+'Pickle y
CSV\\DensidadesMaestro.csv',index_col=0)
VEMaestro=pd.read_csv(path+'Pickle y CSV\\VEMaestro.csv',index_col=0)

#Importar modelos
ModeloPuroAlc=pd.read_csv(path +
'Modelos\\ModeloPuroAlc.csv',index_col=0)
ModeloPuroCl=pd.read_csv(path + 'Modelos\\ModeloPuroCl.csv',index_col=0)

```

```

ModeloMezclas=pd.read_csv(path +
'Modelos\\ModeloMezclas.csv',index_col=0)

# ##### Calcular  $\alpha^E$  de mezclas
#Calcular alpha para las sustancias que forman las mezclas
Alpha1=(-ModeloPuroAlc.loc['Temperatura'].values-
ModeloPuroAlc.loc['T*P'].values*DensidadesMaestro['Presion'].values)/Dens
idadesMaestro['Densidad 1'].values

Alpha2=(-ModeloPuroCl.loc['Temperatura'].values-
ModeloPuroCl.loc['T*P'].values*DensidadesMaestro['Presion'].values)/Densi
dadesMaestro['Densidad 2'].values

#Calcular alpha de la mezcla
AlphaM=(-ModeloMezclas.loc['Temperatura'].values-
ModeloMezclas.loc['T*P'].values*DensidadesMaestro['Presion'].values)/Dens
idadesMaestro['Densidad mezcla'].values

#Obtener los volúmenes de compuestos puros y las fracciones molares
V1=VEMaestro['Volumen 1'].values
V2=VEMaestro['Volumen 2'].values
X1=VEMaestro['Fraccion molar'].values
X2=1-VEMaestro['Fraccion molar'].values
#Calcular fracciones del volumen ideal (Phi1 y Phi2)
Phi1=X1*V1/(X1*V1+X2*V2)
Phi2=X2*V2/(X1*V1+X2*V2)

#Calcular alpha de exceso
AlphaE=AlphaM-(Phi1*Alpha1+Phi2*Alpha2)

#Crear tabla para guardar resultados
TablaAlphaEC=pd.DataFrame()
#Guardar x1, T, P, alpha exceso (unidades: 1/kK) y nombre de mezcla
TablaAlphaEC['Fraccion molar']=VEMaestro['Fraccion molar']
TablaAlphaEC['Temperatura']=VEMaestro['Temperatura']
TablaAlphaEC['Presion']=VEMaestro['Presion']
TablaAlphaEC['Alpha de exceso cal']=AlphaE*1000
TablaAlphaEC['Mezcla']=VEMaestro['Mezcla']

#Guardar y mostrar resultados
TablaAlphaEC.head()

# ##### Calcular  $\kappa^E$  de mezclas
#Importar T, P, densidad compuesto 1, densidad compuesto 2 y densidad
mezcla (calculadas las densidades)
T=DensidadesMaestro['Temperatura'].values
P=DensidadesMaestro['Presion'].values
D1=DensidadesMaestro['Densidad 1'].values
D2=DensidadesMaestro['Densidad 2'].values
DM=DensidadesMaestro['Densidad mezcla'].values

#Calcular kappa para los compuestos que forman la mezcla
Kappal=(ModeloPuroAlc.loc['Presion'].values+2*ModeloPuroAlc.loc['P^2'].va
lues*P+
ModeloPuroAlc.loc['T*P'].values*T)/D1

```

```

Kappa2=(ModeloPuroCl.loc['Presion'].values+2*ModeloPuroCl.loc['P^2'].values*P+
        ModeloPuroCl.loc['T*P'].values*T)/D2

#Calculo de kappa para las mezclas
KappaM=(ModeloMezclas.loc['Presion'].values+2*ModeloMezclas.loc['P^2'].values*P+
        ModeloMezclas.loc['T*P'].values*T)/DM

#Calculo de kappa de exceso (mismos Phi's del calculo anterior)
KappaE=KappaM-(Phi1*Kappa1+Phi2*Kappa2)

#Crear una tabla para guardar los datos
TablaKappaEC=pd.DataFrame()

#Guardar x1, T, P, kappaE (1/TPa), nombre de mezcla
TablaKappaEC['Fraccion molar']=VEMaestro['Fraccion molar']
TablaKappaEC['Temperatura']=VEMaestro['Temperatura']
TablaKappaEC['Presion']=VEMaestro['Presion']
TablaKappaEC['Kappa de exceso cal']=KappaE*1e6
TablaKappaEC['Mezcla']=VEMaestro['Mezcla']

#Guardar y mostrar resultados
TablaKappaEC.head()

# #### Calcular  $P_i^E$  de mezclas
#Calculo de presion interna (MPa) de sustancias puras
Pi1=Alpha1*T/Kappa1-P
Pi2=Alpha2*T/Kappa2-P

#Calcular la presion interna de mezclas
PiM=AlphaM*T/KappaM-P

#Calcular las fracciones de compresibilidad para calcular Pi de exceso
Psi1=X1*Kappa1*V1/(X1*Kappa1*V1+X2*Kappa2*V2)
Psi2=X2*Kappa2*V2/(X1*Kappa1*V1+X2*Kappa2*V2)

#Calcular Pi de exceso
PiE=PiM-(Psi1*Pi1+Psi2*Pi2)

#Crear tabla para guardar los resultados del Pi
TablaPiEC=pd.DataFrame()
#Extraer x1, T, P, PiE (MPa), nombre de mezcla
TablaPiEC['Fraccion molar']=VEMaestro['Fraccion molar']
TablaPiEC['Temperatura']=VEMaestro['Temperatura']
TablaPiEC['Presion']=VEMaestro['Presion']
TablaPiEC['Pi de exceso cal']=PiE
TablaPiEC['Mezcla']=VEMaestro['Mezcla']

#Mostrar resultados
TablaPiEC.head()

# #### Importar tablas de  $\alpha^E$  y  $\kappa^E$  reales (Redlich-Kister)
compuestos,tablas=PropEMezclas() #funcion de Funciones.py

#Crear diccionarios para guardar alphaE y kappaE

```

```

DatosAlphaE={}
DatosKappaE={}

#Vector de temperaturas
Temps=['283.15', '293.15', '298.15', '303.15', '313.15', '323.15']

#Importar tablas 77-84
for tabla,comp in zip(tablas,compuestos):
    nombre=path + 'Tablas\\Propiedades exceso\\' + tabla

DatosAlphaE[comp]=pd.read_csv(nombre,index_col=0).iloc[:,[0,2,4,6,8,10]]
    DatosAlphaE[comp].index.name=comp
    DatosAlphaE[comp].columns=Temps

DatosKappaE[comp]=pd.read_csv(nombre,index_col=0).iloc[:,[1,3,5,7,9,11]]
    DatosKappaE[comp].index.name=comp
    DatosKappaE[comp].columns=Temps

print('Verificar que las tablas se importaron correctamente. Por
ejemplo:')
DatosAlphaE['n-hexano 1-clorobutano (0.1 MPa)']

# #### Organizar las tablas de  $\alpha^E$  y  $\kappa^E$  reales en filas

#Funcion para aplanar los datos
def CrearDatosFilas(DatosDict,Variable):
    #DatosDict es el diccionario de las tablas iniciales
    #Variable es 'AlphaE RK' o 'KappaE RK'
    #crear el diccionario para guardar las tablas
    DatosFilas={}
    #Para cada tabla de DatosDict
    for key in DatosDict.keys():
        #Extraer las fracciones molares (nombres de las filas)
        X=np.array(DatosDict[key].index,dtype=float)
        #Extraer las temperaturas (nombres de las columnas)
        T=np.array(DatosDict[key].columns,dtype=float)
        #Obtener todas las combinaciones de T y X
        Temp,frac=np.meshgrid(T,X)
        #Extraer AlphaE o KappaE
        Var=DatosDict[key].values
        #Aplanar las matrices, unir las en una sola matriz y guardar en un
        DataFrame
        flat=np.column_stack((Var.ravel(),Temp.ravel(),frac.ravel()))

flat_data=pd.DataFrame(flat,columns=[Variable,'Temperatura','Fraccion
molar'])
    #Extraer el valor de la presion (del nombre de la tabla)
    P=float(key.split("(")[1].split(" ")[0])
    #Obtener el numero de filas y agregar columna de P
    nfilas=flat.shape[0]
    flat_data['Presion']=np.ones(nfilas)*P
    #Insertar el nombre de la mezcla
    flat_data['Mezcla']=key.split("(")[0]

```

```

        #Guardar tabla aplanada en nuevo diccionario
        DatosFilas[key]=flat_data
    return DatosFilas

#Aplicar CrearDatosFilas para alphaE y kappaE
DatosAlphaE_RK=CrearDatosFilas(DatosAlphaE,'Alpha de exceso RK')
DatosKappaE_RK=CrearDatosFilas(DatosKappaE,'Kappa de exceso RK')

#Concatenar tablas
Todas_AlphaRK=pd.concat(DatosAlphaE_RK,axis=0,ignore_index=True).dropna()
.reset_index(drop=True)
Todas_KappaRK=pd.concat(DatosKappaE_RK,axis=0,ignore_index=True).dropna()
.reset_index(drop=True)

#Verificar las nuevas tablas maestras
Todas_AlphaRK.head()

# ##### Comparar propiedades termofísicas de exceso reales y calculadas
#Unir tablas de propiedades de exceso calculadas y reales
AlphaETotal=pd.merge(Todas_AlphaRK,TablaAlphaEC,on=['Fraccion
molar','Temperatura','Presion','Mezcla'],how='left')
KappaETotal=pd.merge(Todas_KappaRK,TablaKappaEC,on=['Fraccion
molar','Temperatura','Presion','Mezcla'],how='left')

#Verificar nueva tabla
AlphaETotal.head()

#Funcion para calcular la desviacion y DRCM para cada mezcla
def MezProp(Tabla):
    #Crear vector de mezcla
    Mezc=Tabla.Mezcla.unique()
    #Crear una tabla para guardar los resultados

tabla=pd.DataFrame(np.zeros([len(Mezc),3]),columns=['Mezcla','Desviacion',
'DRCM,%'])

    for mezcla,i in zip(Mezc,range(len(Mezc))):
        #Filtrar tabla de datos por mezcla
        #y quitar los que tienen la propiedad de exceso = 0 (evitar
numeros infinitos)
        Tabla0=Tabla[(Tabla.Mezcla==mezcla) & (Tabla.iloc[:,0] !=
0)].copy()
        #Valores reales (x) y valores calculados (y)
        x=Tabla0.iloc[:,0]
        y=Tabla0.iloc[:,-1]
        #Calcular la desviacion (36 parametros entre los 3 modelos de
densidades)
        Desv=fDesviacion(x,y,36)
        DRCM=fDRCM(x,y)
        tabla.iloc[i,0]=mezcla
        tabla.iloc[i,1]=Desv
        tabla.iloc[i,2]=DRCM
    return tabla.set_index('Mezcla')

```

```

#Comparar alpha's de exceso (1/kK)
MezProp(AlphaETotal)

#Comparar kappa's de exceso (1/GPa)
KappaETotal2=KappaETotal.copy()
KappaETotal2['Kappa de exceso RK']=KappaETotal2['Kappa de exceso RK']/1e3
KappaETotal2['Kappa de exceso cal']=KappaETotal2['Kappa de exceso
cal']/1e3
MezProp(KappaETotal2)

# #### Graficar propiedades termofísicas de exceso reales y calculadas
GraficarProp(AlphaETotal,'Alpha Exceso, 1/kK',path,'Propiedades Exceso')
GraficarProp(KappaETotal,'Kappa Exceso, 1/TPa',path,'Propiedades Exceso')

```

- **Librería de funciones**

```

def fDesviacion(y_real,y_calc,r):
    #y_real = vector con los datos reales/experimentales
    #y_calc = vector con los valores calculados
    #r = numero de parametros
    import numpy as np
    return np.sqrt(np.sum((y_real-y_calc)**2)/(y_real.shape[0]-r))

def fDRCM(y_real,y_calc):
    #y_real = vector con los datos reales/experimentales
    #y_calc = vector con los valores calculados
    import numpy as np
    return 100*np.sqrt(1/len(y_real)*np.sum(((y_real-y_calc)/y_real)**2))

def ModeloLineal(x,y,Aleatorio,Porcentaje=0.05):
    from sklearn.linear_model import LinearRegression
    from sklearn.model_selection import train_test_split
    import numpy as np
    import pandas as pd

    #x = tabla con todos los atributos
    #y = vector con el objetivo
    #Aleatorio es un numero entero para dividir los datos en datos para
    entrenar y datos para validar
    #Porcentaje es el porcentaje de datos para entrenar el modelo

    #separar los datos en los que se usaran para entrenar y los que se
    usaran para validar
    test_size=1-Porcentaje
    x_train, x_test, y_train, y_test = train_test_split(x, y,
    test_size=test_size,random_state=Aleatorio)

    #crear el modelo lineal
    modelo_lineal = LinearRegression()
    #entrenar el modelo the model
    modelo_lineal.fit(x_train, y_train)
    #verificar que tan bien se ajusta el modelo
    y_prediccion = modelo_lineal.predict(x_test)

```

```

DRCM = fDRCM(y_test, y_prediccion)

#guardar los coeficientes del modelo en una tabla
coeficientes =
np.append(modelo_lineal.coef_, modelo_lineal.intercept_)

modelo=pd.DataFrame(coeficientes, index=np.append(x.columns, 'Intercepto'),
columns=['Value'])
modelo.index.name='Variable'

#ordenar los coeficientes de mayor a menor (segun su valor absoluto)
orden_nuevo=modelo['Value'].abs().sort_values(ascending=False).index
modelo=modelo.reindex(orden_nuevo)

return modelo, modelo_lineal, x, y, DRCM

def OptimizarModelo(x, y, cantidad):
    #x = tabla con todos los parametros
    #y = tabla (vector) con el objetivo
    #cantidad = cantidad de veces que se va a correr la regresion para
    hallar la mejor opcion

    incumbent=100 #valor inicial del error
    mejor=[0,0,0,0,0] #inicializar el vector resultado
    for i in range(cantidad):
        Aleatorio=i #correr con todos los valores aleatorios desde 0
        hasta cantidad
        modelo, modelo_lineal, x, y, DRCM=ModeloLineal(x, y, Aleatorio) #correr
        el modelo lineal
        if DRCM<incumbent: #si el error es menor que 100 (incumbent),
        guardar los resultados
            incumbent=DRCM
            mejor=[modelo, modelo_lineal, x, y, DRCM, Aleatorio]
            altorio=Aleatorio
    return mejor #dar el mejor resultado (modelo, modelo_lineal, x, y, DRCM)

def GraficarModelo(modelo_lineal, x, y, tipo, path):
    #Graficar densidades modeladas vs densidades experimentales
    #modelo_lineal = modelo de machine learning
    #x = datos reales
    #y = datos calculados
    #tipo = 'Alcanos', 'Cloroalcanos' o 'Mezclas'
    import matplotlib.pyplot as plt
    from sklearn.metrics import r2_score

    y_pred=modelo_lineal.predict(x)
    DRCM=fDRCM(y, y_pred)
    R2=r2_score(y_true = y, y_pred = y_pred)
    plt.plot(y, y_pred, '.')
    plt.plot([min(y), max(y)], [min(y), max(y)])
    plt.xlabel('Densidad Experimental, g/$cm^3$')
    plt.ylabel('Densidad Modelo, g/$cm^3$')
    plt.legend(['Datos', 'y=x'])
    plt.title('Modelo de %s ($R^2$=%.4f; DRCM=%.4f%%)' % (tipo, R2, DRCM))

```

```

plt.tight_layout()
plt.savefig(path + 'Graficas\\Sustancias Puras\\' + tipo + '.png')
plt.show()

def GraficarProp(tabla,propiedad,path,carpeta):
    #Funcion para graficar las propiedades calculadas vs las reales
    #tabla = tabla con datos
    #propiedad = 'Alpha', 'Kappa', 'Alpha de Exceso', 'Kappa de Exceso'
    #Reales = columna 0
    #Calculadas = columna -1 (ultima)
    import matplotlib.pyplot as plt
    x=tabla.iloc[:,0]
    y=tabla.iloc[:,-1]
    plt.plot(x,y, '.')
    plt.plot([x.min(),x.max()], [x.min(),x.max()])
    plt.xlabel('Real')
    plt.ylabel('Calculada')
    plt.legend([propiedad,'y=x'])
    #Guardar grafica
    plt.tight_layout()
    plt.savefig(path + 'Graficas\\' + carpeta + '\\\'' +
propiedad.split(",")[0] + '.png')
    plt.show()

def CompuestosNombres():
    #Nombres de los compuestos estudiados inicialmente
    return ['n-hexano','n-heptano','2-metilpentano','3-
metilpentano','2,3-dimetilbutano',\
'2,2-dimetilbutano','1-cloropropano','2-cloropropano','1-
clorobutano','2-clorobutano',\
'iso-clorobutano','ter-clorobutano']

def DensCompPuros():
    #Funcion para obtener los nombres de los compuestos puros
    #y los nombres de las tablas donde estan guardadas los valores de
densidad

    #nombres de los compuestos de las tablas de densidades de compuestos
puros
    Compuestos=CompuestosNombres()

    #nombres de archivos donde estan guardadas las tablas de densidades
de compuestos puros
    Tablas=['Tabla7.txt', 'Tabla8.txt', 'Tabla9.txt', 'Tabla10.txt',
'Tabla11.txt', 'Tabla12.txt', \
'Tabla13.txt', 'Tabla14.txt', 'Tabla15.txt', 'Tabla16.txt',
'Tabla17.txt', 'Tabla18.txt']

    return Compuestos,Tablas

def PropCompPuros():
    #Funcion para obtener los nombres de los compuestos puros
    #y los nombres de las tablas donde estan guardadas los valores de
alpha, kappa y Pi

```

```

#nombres de los compuestos de las tablas de propiedades de compuestos
puros
Compuestos=CompuestosNombres()

tablasAlpha=['Tabla41.txt', 'Tabla42.txt', 'Tabla43.txt',
'Tabla44.txt', 'Tabla45.txt', 'Tabla46.txt', \
'Tabla47.txt', 'Tabla48.txt', 'Tabla49.txt', 'Tabla50.txt',
'Tabla51.txt', 'Tabla52.txt']

tablasKappa=['Tabla53.txt', 'Tabla54.txt', 'Tabla55.txt',
'Tabla56.txt', 'Tabla57.txt', 'Tabla58.txt',\
'Tabla59.txt', 'Tabla60.txt', 'Tabla61.txt', 'Tabla62.txt',
'Tabla63.txt', 'Tabla64.txt']

tablasPi=['Tabla65.txt', 'Tabla66.txt', 'Tabla67.txt', 'Tabla68.txt',
'Tabla69.txt', 'Tabla70.txt',\
'Tabla71.txt', 'Tabla72.txt', 'Tabla73.txt', 'Tabla74.txt',
'Tabla75.txt', 'Tabla76.txt']

return Compuestos,tablasAlpha,tablasKappa,tablasPi

def MezclasNombres():
#Nombres de las mezclas

#las mezclas estan compuestas por uno de los alcanos de la lista "a"
y un cloroalcano de la lista "b"
#la lista c contiene todas las presiones posibles en MPa
a=['n-hexano', 'n-heptano']
b=['1-clorobutano', '2-clorobutano', 'iso-clorobutano', 'ter-
clorobutano']
c0=[0.1, 10, 2, 20, 30, 40, 5, 50, 60, 65, 7]
#agregar parentesis y "MPa" a cada valor de presion
c=["("+str(p)+" MPa)" for p in c0]

#hacer todas las combinaciones posibles entre a,b,c
#el resultado es una lista con grupos (a,b,c)
import itertools
x=list(itertools.product(*[a,b,c]))
#unir todos los grupos para que esten en el formato: "alcano
cloroalcano (z MPa)"
y=[" ".join(row) for row in x]

return y

def DensVEMezclas():
#Funcion para obtener los nombres de las mezclas
#y los nombres de las tablas donde estan guardadas los valores de
densidad y VE

#nombres de las mezclas de las tablas de densidades de mezclas
Compuestos=MezclasNombres()

#las tablas de mezclas van de la 19 hasta la 26

```

```

#a es una lista con los numeros del 19 hasta el 26 (en texto)
#b es una lista con las presiones en texto para cada tabla
#en x se unen las palabras "Tabla" + NumeroTabla + "_" + Presion +
".txt"
#(para crear los nombres de los archivos con las tablas para las
mezclas)
a=[str(i) for i in range(19,27)]
b=['01','10','2','20','30','40','5','50','60','65','7']
x=['Tabla'+ai+'_'+bi+'.txt' for ai in a for bi in b]

#nombres de las tablas de densidades de mezclas
Tablas=x

return Compuestos,Tablas

def PropEMezclas():
#Funcion para obtener los nombres de las mezclas
#y los nombres de las tablas donde estan guardadas los valores de
alphaE y kappaE

#nombres de las mezclas de las tablas de densidades de mezclas
Compuestos=MezclasNombres()

#las tablas de mezclas van de la 77 hasta la 84
#a es una lista con los numeros del 77 hasta el 84 (en texto)
#b es una lista con las presiones en texto para cada tabla
#en x se unen las palabras "Tabla" + NumeroTabla + "_" + Presion +
".txt"
#(para crear los nombres de los archivos con las tablas para las
mezclas)
a=[str(i) for i in range(77,85)]
b=['01','10','2','20','30','40','5','50','60','65','7']
x=['Tabla'+ai+'_'+bi+'.txt' for ai in a for bi in b]

#nombres de las tablas de densidades de mezclas
Tablas=x

return Compuestos,Tablas

```