

Resumen

Durante las últimas décadas, el auge en los campos de la tecnología y la computación, ha permitido al ser humano simplificar y acelerar muchas de las tareas llevadas a cabo, automatizando los procesos que las componen. Hoy en día, y cada vez más, el mundo de las aplicaciones está a la orden del día, y la informática es un campo con el que nos relacionamos constantemente y a diario, de manera consciente o inconsciente.

En la actualidad, existen aplicaciones para prácticamente cualquier cosa, independientemente de la plataforma en dónde las usemos (ordenadores, *smartphones*, tabletas, televisores...). Es por este motivo, que en un afán de comprender el funcionamiento y el lenguaje de las computadoras mediante la programación, en este proyecto se emprende largo camino con el fin de poder aplicar los conocimientos de informática aprendidos durante su elaboración, para crear los cimientos de un programa que facilite el análisis en estudios relacionados con el fenómeno físicoquímico de la adsorción.

En este proyecto se ha desarrollado una aplicación compatible con Windows utilizando el lenguaje de programación Python, a fin de analizar los resultados de ensayos experimentales de adsorción.

En la primera parte del trabajo, se exponen las bases teóricas en que se fundamenta el fenómeno de la adsorción, así como los modelos más representativos que describen la naturaleza de estas reacciones.

En segundo lugar, se introduce también a la programación mediante Python, describiendo los métodos y ventajas que ofrece este lenguaje de programación.

Tras presentar la parte teórica del trabajo y aplicando los conceptos adquiridos, se procede con una exposición detallada del proceso de elaboración del programa. También se presentan los manuales y tutoriales correspondientes para su uso e instalación, además de proponer las principales vías de mejora para continuar con su desarrollo. El programa creado se aplicará, finalmente, a resultados de ensayos de adsorción en sistemas químicos reales.

Sumario

RESUMEN	1
SUMARIO	3
1. GLOSARIO	7
2. PREFACIO	11
2.2. Origen del proyecto	11
2.3. Motivación del proyecto.....	11
3. INTRODUCCIÓN	13
3.1. Objetivos del proyecto	13
3.2. Alcance del proyecto	14
3.3. Requisitos previos	15
4. ADSORCIÓN	16
4.1. Definición.....	16
4.2. Tipos de Adsorción.....	17
4.3. Estudio de la adsorción	18
4.4. Cinéticas de adsorción	19
4.4.1. Modelo cinético de pseudo primer orden	19
4.4.2. Modelo cinético de pseudo segundo orden.....	20
4.4.3. Modelo cinético de Elovich.....	21
4.4.4. Modelo de difusión intraparticular	22
4.5. Isotermas de adsorción	22
4.5.1. Isoterma de Langmuir	22
4.5.2. Isoterma de Freundlich	23
4.5.3. Isoterma de Temkin	24
5. INTRODUCCIÓN A LA PROGRAMACIÓN CON PYTHON	25
5.1. ¿Qué es Python?	25
5.2. Programación Orientada a Objetos (POO)	25
5.3. Clases y objetos	26
5.4. ¿Por qué usar Python?	27
5.5. Aplicaciones de Python en la actualidad	28
6. DESARROLLO DEL PROGRAMA	29
6.1. Necesidad de creación del programa.....	29
6.2. Estructura del programa.....	29

6.3.	Herramientas y paquetes usados	31
6.3.1.	Numpy y Scipy	32
6.3.2.	Matplotlib.....	32
6.3.3.	PyQt5.....	32
6.3.4.	cx_Freeze	33
6.3.5.	Distribución de herramientas	33
6.4.	Procesamiento de los datos.....	34
6.4.1.	Datos de entrada	34
6.4.2.	Transformación de los datos.....	36
6.4.3.	Linealización de las ecuaciones y obtención de los parámetros	37
6.4.4.	Obtención del modelo teórico	38
6.5.	Representación gráfica de los resultados.....	39
6.6.	Creación de la Interfaz gráfica (GUI)	40
6.6.1.	Diseño de la Interfaz gráfica de usuario.....	41
6.6.2.	Conexión de la interfaz con el algoritmo del programa	46
6.7.	Empaquetamiento del programa y creación de un instalador	50
6.7.1.	Archivo ejecutable *.exe.....	50
6.7.2.	Creación del instalador para <i>Windows</i> *.msi.....	51
7.	OTRAS IMPLEMENTACIONES	52
7.1.	Icono de la aplicación	52
7.2.	Mensajes de aviso al usuario.....	52
7.3.	Incrustación de los gráficos en una nueva ventana de la interfaz	53
7.4.	Botón Reiniciar.....	53
7.5.	Barra de herramientas: Instrucciones.....	54
8.	VÍAS DE MEJORA Y AMPLIACIÓN	56
8.1.	Ampliación del análisis estadístico descriptivo	56
8.2.	Implementación de otros modelos y tipos de estudio	57
8.3.	Exportación de los resultados a <i>Excel</i>	57
8.4.	Mejora de las opciones de edición de los gráficos	58
9.	COMPROBACIÓN DE FUNCIONAMIENTO DEL PROGRAMA	59
9.1.	Estudio de la cinética de adsorción	59
9.1.1.	Caso de estudio con <i>Excel</i>	59
9.1.2.	Caso de estudio mediante la aplicación creada.....	61
9.1.3.	Comparación de métodos de cálculo.....	64

9.2. Estudio de isotermas de adsorción	65
9.2.1. Caso de estudio con <i>Excel</i>	65
9.2.2. Caso de estudio mediante la aplicación creada	67
9.2.3. Comparación de los métodos de cálculo	70
10. COSTES	72
11. IMPACTO AMBIENTAL	74
12. PLANIFICACIÓN	75
CONCLUSIONES	76
AGRADECIMIENTOS	77
BIBLIOGRAFÍA	78
ANEXO A	80
A.1. Diagrama de funcionamiento del programa	80
A.2. Manual de usuario	81
ANEXO B	93
B.1. Estructura del algoritmo.....	93
B.2. Scripts de <i>Python</i>	94

1. Glosario

Resumen de los principales símbolos, signos, abreviaturas, acrónimos y términos del proyecto:

α : velocidad inicial de adsorción.

Adsorbato: sustancia que se adhiere sobre la superficie de un adsorbente.

Adsorbente: Es un sólido que tiene la capacidad de retener sobre su superficie un componente presente en corrientes líquidas o gaseosas.

Algoritmo de programación: Es una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico.

Array: es un tipo de dato estructurado que permite almacenar un conjunto de datos en la memoria del dispositivo, todos ellos del mismo tipo y relacionados.

β : constante relacionada con la superficie cubierta y la energía de activación por quimiadsorción

Biblioteca (*Library*): En informática, es una colección o conjunto de subprogramas usados para desarrollar software que ofrecen funcionalidades muy específicas al usuario, a modo de subrutinas que funcionan de manera autónoma.

Binding: En el campo de la programación, un binding es una adaptación de una biblioteca para ser usada en un lenguaje de programación distinto de aquel en el que ha sido escrita.

c_1 : constante de unión de equilibrio correspondiente a la máxima energía de enlace.

c_2 : constante relacionada con el calor de adsorción.

C_e : concentración de adsorbato en el equilibrio

Clase: En programación, una clase es una plantilla para la creación de objetos de datos según un modelo predefinido.

Float: los tipos de datos float o de punto flotante son los que representan a los números reales

Framework: En el desarrollo de software, un *framework* o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software.

GUI (Interfaz gráfica de usuario): En informática, es un conjunto de formas y métodos que posibilitan a los usuarios la interacción con un sistema empleando para esto elementos gráficos e imágenes. Estas formas gráficas tales como botones, ventanas, iconos y menús permiten realizar distintas funciones y acciones de manera intuitiva y dinámica, reemplazando la introducción de líneas de comandos en el sistema operativo.

Integer: es un tipo de datos que representa los números enteros; puede tomar valores positivos y negativos, sin parte decimal.

k_1 : constante de velocidad de adsorción de Lagergen [h^{-1}].

k_2 : constante de velocidad de pseudo segundo orden

k_F : constante de Freundlich, relacionada con la capacidad de adsorción

k_L : constante de Langmuir, que relaciona la afinidad del soluto con el adsorbente.

Layout: En programación, es un elemento que representa el diseño de la interfaz de usuario y la organización de elementos gráficos como una actividad, un espacio o un widget. Se encarga de actuar como una plantilla para establecer un orden visual que facilite la comunicación del usuario con la interfaz de la aplicación.

n : parámetro para caracterizar la intensidad de adsorción.

Paradigma de programación: Un paradigma de programación provee y determina la visión y los métodos de un programador en la construcción de un programa o subprogramas. Representa un enfoque particular o filosofía para tratar de resolver uno o varios problemas claramente delimitados.

Python: es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Script: es un archivo que contiene un guión con el conjunto de órdenes y de pasos para el procesamiento de datos, escritos en lenguaje de código de programación y almacenado en un archivo de texto plano.

String: un string o cadena de caracteres es un tipo de dato que contiene una secuencia ordenada de caracteres utilizado habitualmente para el almacenamiento y procesado de palabras, frases o cualquier otra sucesión de caracteres.

q_e : cantidad de soluto adsorbida por unidad de superficie de adsorbente en el equilibrio

q_t : cantidad de soluto adsorbido por unidad de superficie de adsorbente a un tiempo t

t : tiempo de contacto [h]

Q_{max} : constante relacionada con la capacidad máxima de adsorción.

Widget: En informática, un widget o artilugio es una pequeña aplicación o programa, pensado para mostrar información y dar fácil acceso a funciones que se utilicen frecuentemente. Suelen presentarse de manera muy visual y ofrecen una gran variedad de funcionalidades.

2. Prefacio

2.2. Origen del proyecto

El planteamiento inicial de este trabajo surgió de la idea propuesta por el profesor (y director del proyecto) Francisco Javier Giménez, junto con el apoyo del profesor Lluís Solano, también codirector del trabajo.

El principal objetivo de éste era poder realizar el análisis de resultados obtenidos en ensayos experimentales de adsorción, de manera que una vez obtenidos éstos, él fuese capaz de interpretarlos desde el punto de vista fisicoquímico.

En sus inicios, la idea principal del trabajo consistía en la implementación de herramientas informáticas con el fin de automatizar los cálculos respectivos al análisis de los datos recogidos durante estos ensayos, permitiendo así deducir el comportamiento de las especies de estudio en sistemas químicos reales.

No obstante, debido a la gran variedad de estudios posibles, y de las distintas formas de presentar los resultados, fue necesario acotar algunos aspectos previos con tal de que el proyecto fuese viable. En primer lugar, el trabajo se enfocaría únicamente a estudios relacionados con la cinética y las isothermas de adsorción de sustancias químicas en disolución, incluyendo los modelos más representativos utilizados en cada uno de ellos. Además, se definió que estos resultados serían presentados gráficamente, para una mejor interpretación y visualización. De este modo, los gráficos obtenidos podrían ser incluidos en artículos que el profesor, junto con otros compañeros y alumnos, estuvieran desarrollando.

Finalmente, se supo que el director del proyecto contemplaba también la posibilidad de, en un futuro, ampliar el alcance de este proyecto a otros tipos de estudios de adsorción similares. Considerando esto, se llegó a la conclusión de que la opción más efectiva sería desarrollar íntegramente un programa informático, que permitiese llevar a cabo estos estudios de la manera más simplificada posible, sentando las bases y estructurándolo de forma que pudiera ser mejorado más adelante por otros alumnos en futuros proyectos.

2.3. Motivación del proyecto

El factor principal que motivó la realización de este proyecto fue el gran reto que suponía sumergirse en el mundo de la programación. Si bien es cierto que durante el transcurso de la carrera universitaria se impartió una asignatura referente a la informática, los conocimientos

adquiridos en ella eran totalmente insuficientes para llevar a cabo un proyecto de estas dimensiones. La pasión por parte del autor por la tecnología, y el auge de la informática en muchos ámbitos de la ingeniería en la actualidad, fueron determinantes en la decisión de tirar hacia adelante este proyecto. Este trabajo ofrecía la oportunidad de poder aprender realmente a conocer e interpretar el lenguaje de programación, y además, poder aplicar la formación adquirida a un caso real, con el desarrollo de una aplicación informática.

Otro de los grandes motivos que impulsó a llevar a cabo este proyecto, fue la posibilidad de ampliar los conocimientos fisicoquímicos adquiridos durante la titulación referente al fenómeno de la adsorción. En ella, se habían visto los conceptos básicos que describen éste fenómeno en la asignatura de *Química-Física*, además de realizar algunos ensayos relacionados en las diversas asignaturas de *Experimentación en Ingeniería Química*. La realización de este proyecto, suponía poder profundizar acerca del comportamiento de éste fenómeno en sistemas químicos reales, y conocer los modelos teóricos y mecanismos que describen estas reacciones.

3. Introducción

3.1. Objetivos del proyecto

Como se ha mencionado anteriormente, el principal objetivo que pretende cumplir este proyecto es el desarrollo de una aplicación que facilite y permita realizar análisis de los resultados obtenidos en ensayos experimentales de adsorción. Esta herramienta debe buscar satisfacer los siguientes requisitos:

1. Caracterización e interpretación química de los datos de procedencia.
2. Recopilación y categorización de los modelos de adsorción más representativos para cada tipo de estudio.
3. Aplicación de técnicas informáticas y estadísticas para el procesamiento de datos.
4. Automatización del cálculo numérico de las ecuaciones matemáticas para cada modelo.
5. Representación gráfica de los resultados obtenidos.
6. Simplicidad y facilidad de uso del programa.
7. Fiabilidad de resultados, y comprobación de los mismos.

El primer objetivo es definir cuáles son los parámetros de control más relevantes a tener en cuenta en los ensayos de adsorción para su análisis, con el fin de poder determinar bajo qué tipo de mecanismos se rigen estas reacciones.

El segundo objetivo se trata de recolectar los modelos más representativos y contrastados para describir el fenómeno de adsorción en los ensayos experimentales, además de sus respectivas ecuaciones matemáticas y parámetros característicos.

El tercer objetivo marcado consiste en desarrollar una herramienta mediante la aplicación de técnicas informáticas que, dados los valores recopilados en los ensayos, permita realizar y automatizar los procesos necesarios para el cálculo de resultados de cada modelo. Además, debe permitir también representar gráficamente estos resultados de manera automática, ofreciendo así una mejor visualización de los mismos.

El siguiente objetivo que se pretende cumplir es facilitar la utilización de la herramienta creada, ofreciendo a los usuarios un entorno amigable y un manejo simple de las distintas funcionalidades integradas con tal de obtener los resultados deseados.

Por último, deberá comprobarse la fiabilidad de los resultados obtenidos a través de la herramienta desarrollada. Para ello, al final del trabajo se realizará un estudio con datos provenientes de un caso ensayo concreto facilitado por el director, dónde se compararán los resultados obtenidos en ésta frente a otros métodos de cálculo utilizados habitualmente (*Excel*).

3.2. Alcance del proyecto

El planteamiento de este proyecto por parte del director Javier Giménez pretende crear la base para que pueda ser llevado a cabo por varias personas en serie, aumentando progresivamente el contenido del mismo hasta cumplir todos los objetivos pretendidos. Por lo tanto, puede considerarse que éste establece el inicio de un proyecto mayor, fragmentado en varios trabajos y que busca desarrollar a largo plazo una herramienta de cálculo y análisis completa para ensayos fisicoquímicos concretos. Además, se cuenta con su aprobación, puesto que será él quien permita la continuidad del proyecto, siendo conocedor de que con los recursos requeridos en este trabajo (horas de dedicación) no se pueden alcanzar la totalidad de los objetivos.

En primera instancia, la definición del alcance era algo difícil de definir, puesto que no se sabía hasta que punto de desarrollo podría llegarse en esta primera fase del proyecto. Sin embargo, tras las primeras reuniones se acordó por ambas partes que éste debería intentar satisfacer lo siguiente:

- Documentación teórica del fenómeno de adsorción y evaluación de los datos.
- Desarrollo de algún método que permita la automatización de cálculos matemáticos y estadísticos para el análisis de los datos de procedencia.
- Desarrollo de una herramienta que permita la representación bidimensional de los resultados, y la comparación entre los distintos casos posibles.
- Creación de una interfaz de usuario simple e intuitiva.
- Redactado de documentación detallada referente a la elaboración y el funcionamiento del programa, para permitir continuar con su desarrollo.
- Realización de un estudio de comparación de resultados a partir de datos facilitados por el director.
- Propuesta de vías de mejora y ampliación de la herramienta desarrollada.
- Redactado del manual de usuario.

3.3. Requisitos previos

En una de las siguientes reuniones con los directores del proyecto, se debatió acerca de qué lenguaje de programación sería el empleado en el desarrollo de la herramienta. Ésta debería ser capaz de poder satisfacer todas las funciones expuestas previamente. Por experiencia previa en otros trabajos, la cantidad de información disponible en internet y aconsejado por ambos profesores, finalmente se acordó que el lenguaje utilizado sería *Python*.

Debido a los limitados conocimientos previos en programación y con tal de poder asimilar los conceptos necesarios para desarrollar una aplicación como la requerida, fueron necesarias muchas horas de investigación y de autoaprendizaje mediante cursos y tutoriales. Éstas no se incluyen en la realización del proyecto, sino que forman parte del interés propio del autor por este campo.

Durante este proceso, fue de gran ayuda la colaboración de Lluís Solano, dando soporte en todos los temas referentes a la programación, además de guiar hacia buen puerto al autor con tal de conseguir los objetivos propuestos, dada la dificultad que suponía crear un programa desde cero para un principiante en esta materia.

4. Adsorción

4.1. Definición

La adsorción es definida como un proceso mediante el cual se extrae materia de una fase y se concentra sobre la superficie de otra (generalmente sólida). Este fenómeno puede producirse en las siguientes: parejas de fases, líquido-sólido, líquido-líquido, gas-sólido y gas-líquido.

En la práctica, se entiende por adsorción la eliminación de uno o más componentes presentes en una fase líquida o gas, mediante un sólido. Cualquier especie que se concentre en la superficie o que se adsorba se denomina adsorbato, y el material sobre el cual se adhiere, se adsorbe, se llama adsorbente. El proceso inverso a la adsorción se conoce como desorción.

La atracción que provoca que las moléculas en disolución queden retenidas sobre la superficie de un sólido puede ser de naturaleza muy diversa (química, física y/o eléctrica) y tiene lugar en la interfase sólido-disolución. Los principales factores que afectan a la adsorción son el tamaño, la forma y la polaridad tanto de las moléculas del fluido implicado como de las partículas sólidas.

La adsorción no debe confundirse con la absorción, ya que la característica que las distingue es que en un proceso de adsorción el fluido no se incorpora al volumen del material, sino que es retenido en su superficie. En cambio, en la absorción, las moléculas o átomos de una fase penetran casi uniformemente en los de otra fase constituyendo una solución. Factores a tener en cuenta en la absorción son la densidad, viscosidad y tensión superficial por parte del líquido, y la presencia de porosidad, espacios interlaminares y/o canales estructurales en el sólido.

Además de estos fenómenos comúnmente conocidos como procesos de sorción (adsorción, absorción y desorción), cabe mencionar también el fenómeno de intercambio iónico, que supone un intercambio de una sustancia o ion por otra sobre la superficie del sólido. A diferencia de lo que sucede con los procesos de adsorción, donde las ecuaciones que la describen consideran solamente una especie química, de manera que la distribución del soluto entre la disolución y el sólido responde a una relación simple, lineal o no; en las ecuaciones para el intercambio iónico, se tienen en cuenta todos los iones que compiten por los lugares de intercambio [1].

En la Fig. 4.1 se puede observar un esquema gráfico de éstos fenómenos:

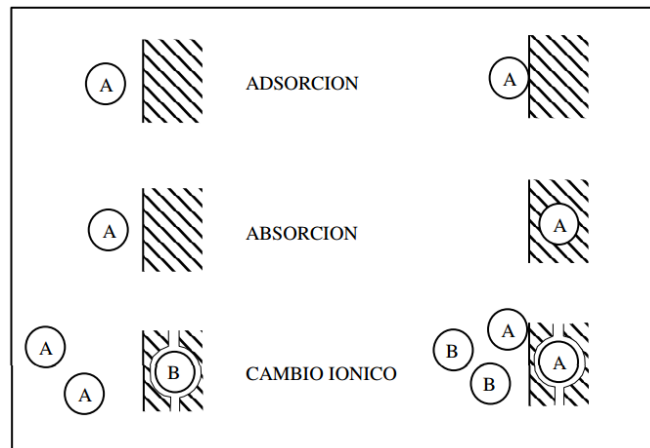


Fig. 4.1. Esquema gráfico de los procesos de adsorción, absorción e intercambio iónico

4.2. Tipos de Adsorción

Una vez definido el concepto de adsorción, cabe distinguir entre tres tipos existentes de adsorción según la naturaleza de la atracción que se produzca en la interacción entre el adsorbato y el adsorbente:

- **Adsorción eléctrica o por intercambio:** es un proceso mediante el cual los iones de una sustancia se concentran en una superficie como resultado de la atracción electrostática en los lugares cargados de la dicha. Suponiendo dos adsorbatos iónicos posibles, a igualdad de otros factores, la carga del ion es el factor determinante en la adsorción por intercambio. Para iones de igual carga, el tamaño molecular (o radio de solvatación) determina el orden de preferencia para la adsorción.
- **Adsorción física o fisisorción:** se trata del proceso de adsorción que tiene lugar debido a las fuerzas de Van der Waals. En este caso, la molécula adsorbida no está fija en un lugar específico de la superficie, sino más bien está libre de trasladarse a través de la interfase. En general, este tipo de adsorción predomina a temperaturas bajas. La adsorción de la mayoría de las sustancias orgánicas en el agua con carbón activado se consideran de naturaleza física.
- **Adsorción química o quimisorción:** también conocida como adsorción activa, ésta se da cuando el adsorbato sufre una interacción química con el adsorbente. Las energías de adsorción en este caso son elevadas, del orden de las de un enlace químico, debido a que el adsorbato forma unos enlaces fuertes localizados en los centros activos del adsorbente. Esta adsorción suele estar favorecida a elevadas temperaturas.

Aunque se presenten de forma separada, la mayor parte de los fenómenos de adsorción que tienen lugar son combinaciones de las tres formas presentadas; por lo que a menudo, es difícil distinguir entre la adsorción química y física [2].

4.3. Estudio de la adsorción

Cuando se hacen estudios acerca de los fenómenos de adsorción, es necesario prestar especial atención a las propiedades del adsorbente, debido a que su función principal es la de atrapar los átomos, iones o moléculas de soluto presentes en una disolución. Por ello, es de gran interés conocer los aspectos más importantes que debe cumplir un buen adsorbente para su uso en los ensayos de adsorción:

- El adsorbente deberá tener una gran capacidad de retención.
- Además de poseer una velocidad rápida de adsorción.
- Tener una gran superficie específica (porosidad) o volumen de microporos.
- Estructura porosa para que las moléculas del adsorbato se transporten fácilmente hacia los sitios activos de la superficie.
- Poseer una buena capacidad de regeneración, para permitir su reutilización y hacer los ensayos de adsorción más efectivos y económicos.

Los estudios de adsorción se llevan a cabo, generalmente, mediante dos tipos de experimentos distintos: los ensayos en batch (en discontinuo) y en columna (continuo).

- En los **experimentos denominados en batch**, los estudios se centran en la medida de la eficacia de adsorción para la eliminación de sustancias específicas, así como de la capacidad máxima de adsorción por parte del adsorbente.

Para ello, se coloca una cantidad conocida del mismo en contacto con una disolución con una concentración determinada de soluto, y se pone en agitación para mantener el adsorbato en suspensión. Pasado un tiempo, durante el cual se supone que se alcanza el equilibrio, se determina la concentración de iones de interés contenido en el líquido sobrenadante, considerando que la diferencia entre ésta y la concentración inicial es la cantidad de soluto adsorbida sobre el suelo.

- En los **ensayos en columna de lecho fijo**, se hace pasar la disolución con los iones o átomos de interés (fase móvil), por un conducto que contiene el adsorbente (fase sólida). La principal diferencia con los experimentos en batch, es debida a que en los estudios de lecho fijo no se alcanzan condiciones de equilibrio, ya que continuamente ingresa a la columna una disolución de alimentación.

En el ámbito industrial generalmente se trabaja bajo condiciones continuas, por lo que los estudios en columna suelen proporcionar una aplicación más práctica, en procesos como el tratamiento de aguas residuales o la adsorción de gases.

4.4. Cinéticas de adsorción

Predecir la velocidad a la que la adsorción se lleva a cabo para un sistema dado, es probablemente el factor más importante para el diseño del mismo, junto con el tiempo de residencia del adsorbato y las dimensiones del reactor. Además, el estudio de la cinética de adsorción aporta ideas acerca de los mecanismos por los que se rigen las reacciones.

Con el fin de estudiar los mecanismos de adsorción, se han presentado numerosos modelos cinéticos: algunos de ellos describen el orden de reacción de los sistemas de adsorción basándose en la concentración de la disolución; otros, en cambio, se basan en la capacidad del adsorbente.

En este proyecto, se centra la atención en las ecuaciones cinéticas de adsorción basadas en la capacidad del sólido (adsorbente) para un sistema de líquidos y sólidos. A continuación, se describen los modelos existentes más representativos que describen este fenómeno. Estos son el modelo de pseudo-primero orden, de pseudo-segundo orden, el modelo de Elovich, y el de difusión intraparticular [3].

Todos los modelos presentados a continuación pretenden ser incorporados al programa desarrollado en este trabajo, pero tratándose de una primera versión tan sólo se implementaran algunos de ellos.

4.4.1. Modelo cinético de pseudo primer orden

El modelo cinético de pseudo primer orden, propuesto por Lagergren, ha sido ampliamente utilizado para describir la cinética de adsorción [4]. Éste viene dado por la ecuación:

$$\frac{dq}{dt} = k_1 \cdot (q_e - q) \quad (\text{Ec. 4.1})$$

Donde:

k_1 : constante de velocidad de adsorción de Lagergen [h^{-1}].

q_t : cantidad de soluto adsorbida a un tiempo t [mol/m^2]

q_e : cantidad de soluto adsorbida en el equilibrio [mol/m^2]

Integrando la ecuación anterior con respecto a las condiciones de contorno:

$$\left[\begin{array}{l} t = 0 \rightarrow q = 0 \\ t = t \rightarrow q = q_e \end{array} \right.$$

Resulta:

$$\log(q_e - q_t) = \log(q_e) - \frac{k_1}{2,303} \quad (\text{Ec. 4.2})$$

4.4.2. Modelo cinético de pseudo segundo orden

El modelo de pseudo segundo orden fue desarrollado por Ho y McKay, y publicado en 1999. En este caso, la ecuación basada en el equilibrio de adsorción se expresa como:

Las variables de la ecuación hacen referencia a:

$$\frac{dq}{dt} = k_2 \cdot (q_e - q_t)^2 \quad (\text{Ec. 4.3})$$

k_2 : constante de velocidad de pseudo segundo orden [$m^2 \cdot mol^{-1} \cdot h^{-1}$].

q_t : cantidad de soluto adsorbido por unidad de superficie de adsorbente a un tiempo t [$mol \text{ adsorbato} / m^2 \text{ adsorbente}$]

q_e : cantidad de soluto adsorbida por unidad de superficie de adsorbente en el equilibrio [$mol \text{ adsorbato} / m^2 \text{ adsorbente}$]

t : tiempo de contacto [h]

Donde, integrando la ecuación anterior y manteniendo las mismas condiciones de contorno que las descritas para el modelo de pseudo primer orden, se obtiene:

$$\frac{t}{q_t} = \frac{1}{k_2 \cdot q_e^2} + \frac{1}{q_e} \cdot t \quad (\text{Ec. 4.4})$$

La constante de velocidad de este modelo (k_2) es determinada experimentalmente de la pendiente y de la ordenada en el origen de la gráfica t/q_t versus t , donde también es posible obtener el valor de concentración en el equilibrio (q_e) [5].

4.4.3. Modelo cinético de Elovich

El modelo de Elovich es aplicado principalmente en procesos de quimisorción, para determinar la cinética de adsorción de fases líquidas o gas sobre superficies sólidas [6]. La ecuación que lo describe viene definida por:

Donde:

$$\frac{dq}{dt} = \alpha \cdot e^{-\beta \cdot q} \quad (\text{Ec. 4.5})$$

α : velocidad inicial de adsorción [$mol \cdot g^{-1} \cdot h^{-1}$].

β : esta constante está relacionada con la superficie cubierta y la energía de activación por quimisorción [m^2 / mol].

Integrando la ecuación, y aplicando las mismas condiciones de contorno descritas para los demás modelos, se obtiene una ecuación linealizada que debe ajustarse a los datos experimentales:

$$q_t = \frac{1}{\beta} \cdot \ln(\alpha \cdot \beta) + \frac{1}{\beta} \cdot \ln(t) \quad (\text{Ec. 4.6})$$

4.4.4. Modelo de difusión intraparticular

En este modelo desarrollado por Weber y Morris, está basada en el transporte del adsorbato a través de la estructura interna de los poros del adsorbente y de la propia difusión en el sólido, y ha sido utilizada para determinar si la difusión intraparticular es la etapa controlante en la dinámica de los procesos de adsorción [7].

La ecuación que describe este mecanismo viene dada por:

$$q_t = k \cdot \sqrt{t} + C \quad (\text{Ec. 4.7})$$

Donde k hace referencia a la velocidad de difusión intraparticular [$\text{mol} \cdot \text{g}^{-1} \cdot \text{h}^{-1/2}$].

4.5. Isotermas de adsorción

Las isotermas de adsorción describen el mecanismo de la interacción entre el adsorbato y el adsorbente, y proveen información acerca del equilibrio del proceso de adsorción, así como de la capacidad de retención del adsorbente de estudio.

El objeto de estudio de las isotermas se centra en analizar la relación entre la cantidad de adsorbato adsorbida por un adsorbente (q) y la concentración en el equilibrio (C) o presión a una temperatura constante de ese adsorbato [8].

Entre los modelos desarrollados para describir la relación entre la cantidad adsorbida y la concentración o presión en el equilibrio destacan las isotermas de Langmuir, Freundlich y Temkin [9].

4.5.1. Isotherma de Langmuir

La isoterma de adsorción de Langmuir es un modelo teórico utilizado ampliamente en los estudios de adsorción en batch, puesto que permite obtener información sobre el mecanismo de adsorción. En este modelo, se asume que la adsorción se produce en monocapa, y tiene lugar en sitios específicos de la superficie del adsorbente de manera homogénea. Por tanto, considera que todos los sitios son idénticos y tan sólo puede ser adsorbida una molécula por sitio [10].

El modelo de isoterma de Langmuir viene representado por la siguiente ecuación:

$$q_e = \frac{Q_{max} \cdot k_L \cdot C_e}{1 + k_L \cdot C_e} \quad (\text{Ec. 4.8})$$

Donde:

q_e : concentración de soluto adsorbido por unidad de superficie de adsorbente en el equilibrio.

C_e : concentración de adsorbato en el equilibrio

k_L : constante de Langmuir, que relaciona la afinidad del soluto con el adsorbente.

Q_{max} : constante relacionada con la capacidad máxima de adsorción.

Y su forma lineal se expresa de la siguiente manera:

$$\frac{C_e}{q_e} = \frac{1}{k_L \cdot Q_{max}} + \frac{C_e}{Q_{max}} \quad (\text{Ec. 4.9})$$

4.5.2. Isotherma de Freundlich

La isoterma de Freundlich es un modelo empírico, el cual supone que la superficie del adsorbente es heterogénea y que se produce un crecimiento en multicapas. También se considera que los sitios de adsorción tienen distintas afinidades; en primer lugar, se ocupan los sitios de adsorción por enlaces más fuertes, y a medida que los sitios activos incrementan su ocupación, las fuerzas de enlace decrecen [11].

La ecuación del modelo de Freundlich se expresa de la siguiente manera:

$$q_e = K_F \cdot C_e^{1/n} \quad (\text{Ec. 4.10})$$

Dónde:

K_F : constante de Freundlich, relacionada con la capacidad de adsorción

n : parámetro para caracterizar la intensidad de adsorción.

Su forma linealizada corresponde a la ecuación:

$$\log(q_e) = \log(k_F) + \frac{1}{n} \cdot \log(C_e) \quad (\text{Ec. 4.11})$$

4.5.3. Isoterma de Temkin

La isoterma de Temkin considera que la entalpía molar de adsorción disminuye linealmente a medida que aumenta el grado de recubrimiento debido a las interacciones adsorbato-adsorbente. Además supone que la adsorción posee una distribución uniforme de las energías de unión, hasta un cierto valor de energía de enlace máximo [12]. La isoterma de Temkin representa viene expresada por la siguiente ecuación:

$$q_e = c_1 \cdot \ln(c_2 \cdot C_e) \quad (\text{Ec. 4.12})$$

Donde:

c_1 : constante de unión de equilibrio correspondiente a la máxima energía de enlace [$L \cdot mg^{-1}$].
 c_2 : constante relacionada con el calor de adsorción.

Representando gráficamente q_e versus $\ln(C_e)$, es posible obtener las constantes c_1 de la pendiente, y c_2 de la ordenada en el origen.

La linealización de la ecuación de la isoterma de Temkin da lugar a la siguiente expresión:

$$q_e = c_1 \cdot \ln(c_2) + c_1 \cdot \ln(C_e) \quad (\text{Ec. 4.13})$$

5. Introducción a la programación con Python

5.1. ¿Qué es Python?

Python es un lenguaje de programación que fue creado por un científico holandés llamado Guido van Rossum a principios de los años 90.

Se trata de un lenguaje interpretado, interactivo y orientado a objetos que ofrece una gran cantidad de estructuras de datos de alto nivel por medio de un tipado dinámico y fuerte, además de ser multiparadigma y multiplataforma [13].

Python es un lenguaje **dinámicamente tipado**. Es decir, en él no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que dicho tipo de dato será determinado en el momento (o tiempo) de ejecución según el valor que se le asigne a la variable. Además, el tipo de la variable puede cambiar si se le asigna un valor de otro tipo.

Por otro lado, Python es un lenguaje **fuertemente tipado**. Por lo tanto, si se posee una variable de tipo entero, no podrá tratarla como una cadena de texto a menos que se realice una conversión previamente de manera explícita.

Además, Python se trata de un lenguaje de programación **multiparadigma**, ya que soporta la orientación a objetos, la programación imperativa, y, en menor medida, la programación funcional.

También se trata de un lenguaje de programación **multiplataforma**, el cuál puede usarse en distintas plataformas como son Microsoft Windows, GNU/Linux, Unix, MacOs, Solaris...

5.2. Programación Orientada a Objetos (POO)

Como se ha mencionado en apartados anteriores, Python es un lenguaje de programación orientado a objetos (POO). Ésta tiene sus orígenes en la década de 1960, pero es a partir de los 80 cuando se convierte en el principal paradigma de la programación de un nuevo software, hasta día de hoy. En el ámbito de la programación, un paradigma es conocido como una forma específica de hacer las cosas que facilita la solución a unos problemas determinados [14].

No obstante, anteriormente ya existían otros paradigmas de la programación como la programación imperativa y la programación funcional.

La POO surgió para que fuera posible representar elementos de la vida real en un programa de una forma entendible, y para facilitar el mantenimiento y reutilización de los programas.

5.3. Clases y objetos

Un programa Orientado a Objetos se basa en una agrupación de objetos de distintas clases que interactúan entre si y que, en conjunto, consiguen que un programa cumpla con su propósito.

Para comenzar, es necesario comprender que todo lo que nos rodea se puede considerar un objeto, incluso nosotros mismos nos podemos considerar objetos, cada uno distinto a los demás. Por lo tanto, un objeto es algo que se visualiza, es utilizado y que juega un papel o un rol. Un objeto no tiene que ser necesariamente algo concreto o tangible, sino que puede ser totalmente abstracto, y también describir un proceso.

- En programación, un **objeto** es un elemento de programa como pueden serlo variables o funciones. Las variables almacenan datos, y las funciones ejecutan tareas. En cambio, un objeto puede almacenar datos y ejecutar tareas. De ahora en adelante, a las variables se les llamará **atributos** del objeto; y a las funciones, **métodos**.

Dependiendo del problema a tratar, diferentes aspectos de un objeto son significativos. Así, los atributos describen propiedades y características de un objeto. Pero éste también posee métodos que actuarán sobre estos atributos o datos.

- El concepto de **clase**, define como representar de forma abstracta a un objeto, es decir, es la plantilla sobre la cual se crea ese objeto.

Para entender la filosofía en la que se basa la POO, es mejor hacerlo a través de un simple ejemplo, con tal de afianzar los conceptos anteriormente expuestos y que más adelante aparecerán durante el desarrollo del programa creado en este proyecto:

Una clase llamada “coche” podría definir que un coche es todo aquel objeto que tiene las siguientes características: marca, modelo, año, color, número de puertas, etc. Éstos podrían ser algunos de los **atributos** que podría contener la clase coche.

De la misma manera, la clase coche podría definir también que todo coche tiene comportamientos en común como son arrancar, acelerar o frenar. Estas acciones que puede realizar cualquier coche, conformarían los **métodos** del coche.

Una vez definidos todos los atributos y métodos de la clase coche, y asignándoles un valor específico a cada uno de ellos, el resultado sería un **objeto** coche en concreto, con todas sus características definidas [15].

5.4. ¿Por qué usar Python?

En este apartado, se tratarán de exponer algunas de las principales ventajas que ofrece la programación en Python frente a otros lenguajes de programación:

- En la creación de un nuevo programa, si éste no utiliza bibliotecas específicas para una determinada plataforma (Windows, Linux, MacOs, etc), el programa en cuestión podrá funcionar en todos los demás sistemas sin la necesidad de realizar grandes cambios.
- Algunos de los lenguajes de programación fueron creados para trabajar en un ámbito específico, como por ejemplo el lenguaje PHP, que fue ideado para desarrollar aplicaciones Web. Sin embargo, éste no es el caso de Python, ya que fue diseñado sin ningún propósito en concreto, y este hecho hace de él un lenguaje versátil y multipropósito.
- Además, una de las principales características de Python es que es un lenguaje de programación que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje de máquina. Esto provee a Python de una sintaxis que favorece un código legible, y que es mucho más fácil de comprender para el usuario.
- Python es *open source*, es decir que cualquiera puede contribuir a su desarrollo y divulgación. Además, no es necesario pagar ninguna licencia para distribuir el software desarrollado con éste lenguaje. Hasta su intérprete es distribuido de forma gratuita para las diferentes plataformas existentes.

En resumen, Python es considerado por muchos usuarios un lenguaje muy productivo, además de ser potente y flexible. Por otro lado, su sintaxis clara y concisa hace que no sea necesario dedicar tiempo a su compilación, puesto que es interpretado.

5.5. Aplicaciones de Python en la actualidad

Actualmente, el lenguaje de programación Python es uno de los más empleados para el desarrollo de software. En los últimos años su utilización ha ido creciendo constantemente entre los usuarios por el hecho de ser un lenguaje de programación de alto nivel, interpretado, y con una gran variedad de fines y propósitos.

Este lenguaje abre un amplio abanico de posibilidades, permitiendo desarrollar software para comunicaciones de red, aplicaciones de escritorio con interfaz gráfica de usuario (GUI), crear juegos, y por supuesto, para aplicaciones web y para smartphones. Pero no sólo se utiliza en sectores tecnológicos, sino que también es empleado en otros ámbitos como el financiero, el científico, o en la educación.

Numerosas empresas mundialmente conocidas hacen un uso intensivo de Python para desarrollar sus productos y servicios. Entre ellas destacan empresas y organizaciones del calibre de Google, la NASA, Walt Disney, IBM, Red Hat, Industrial Light&Magic y Nokia, éste último lo emplea en su conocido sistema operativo Symbian.

Esto demuestra que este lenguaje puede ser utilizado en distintos tipos de sectores con independencia de su actividad empresarial **[16]**.

6. Desarrollo del programa

6.1. Necesidad de creación del programa

La motivación que surgió para enfocar el desarrollo de este trabajo en torno al mundo de la programación, fue la necesidad de crear un programa que procesara y graficara los resultados experimentales obtenidos en ensayos de adsorción.

Para ello, los datos recogidos experimentalmente deberían ser procesados mediante cálculos, siguiendo los distintos modelos expuestos anteriormente que describen los fenómenos de adsorción, y poder así obtener gráficos de los resultados de manera inmediata, ahorrando tiempo en el análisis de los mismos.

Además, el hecho de que éste fuera un programa creado desde cero, suponía que debería poseer una estructura lógica y comprensible, para que en un futuro pudiera ser ampliado y actualizado.

6.2. Estructura del programa

En primer lugar, fue necesario definir una primera estructura mediante diagramas, con tal de definir las principales funcionalidades contenidas en la aplicación. En ella, el usuario debería introducir datos experimentales concretos recogidos en ensayos de adsorción, y el programa debería procesarlos, realizar cálculos sobre los mismos y ofrecer unos resultados, en este caso se determinó que deberían presentarse de forma gráfica:

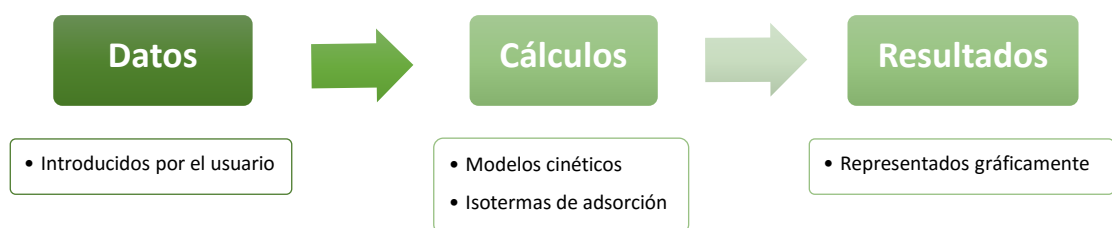


Fig. 6.1. Diagrama de estructura básica del programa.

Una vez planteadas las funcionalidades del programa, se procedió a continuar con el diseño de la aplicación. Para que el usuario pudiera interactuar con él sin necesidad de poseer ningún tipo de experiencia previa en programación, se decidió que habría que crear una interfaz gráfica de usuario (GUI, por sus siglas en inglés: *Graphical User Interface*). Así, mediante un simple formulario, el usuario podrá introducir fácilmente los datos a analizar, y seleccionar el tipo de modelo deseado a calcular.

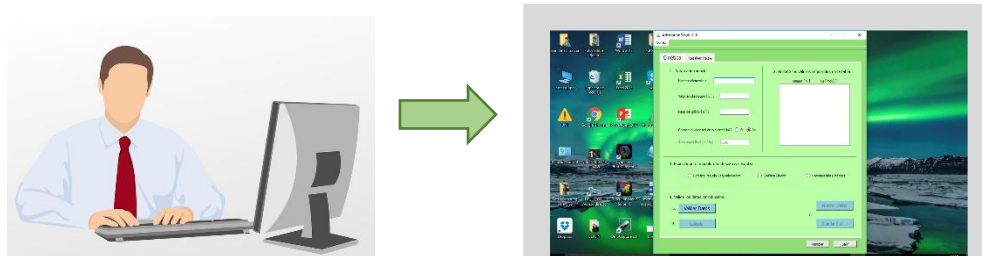


Fig. 6.2. Ilustración de la interacción usuario-GUI en el programa.

Al tratarse de una primera versión del programa, y debido a la gran cantidad de modelos existentes para estudiar los fenómenos de adsorción, se decidió acotar el número de modelos disponibles, seleccionando entre ellos los más representativos según el tipo de estudio. Los modelos elegidos se dividieron en dos grandes bloques:

- **Modelos cinéticos:** el modelo cinético de pseudo-segundo orden y el modelo de Elovich fueron los seleccionados.
- **Isotermas de adsorción:** En este caso, se escogió la isoterma de Langmuir y la isoterma de Freundlich.

Debido a que los dos grupos de estudio pertenecen a distintos tipos de ensayos y requieren distintos datos de entrada, se decidió que el código del programa también trataría a los dos bloques de estudio por separado. Es decir, que habría un código independiente para los modelos cinéticos, y otro para las isotermas de adsorción. De esta manera, se evitarían posibles conflictos en la futura compilación del programa, y éste estaría dotado de una estructura interna más fácilmente comprensible y ampliable.

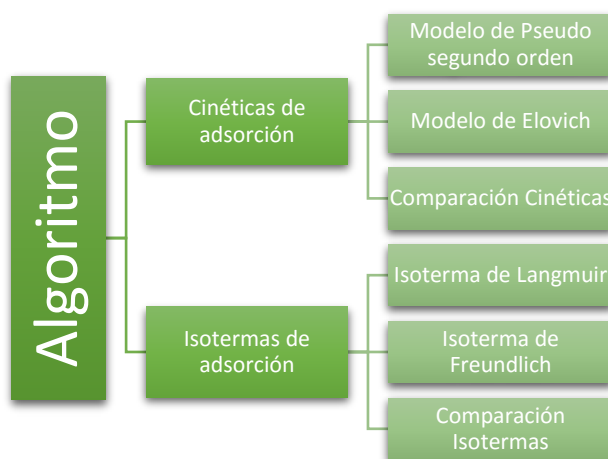


Fig. 6.3. Estructura interna del algoritmo referente a los modelos implementados.

Los datos recogidos durante los ensayos de adsorción deberían ser procesados de la siguiente manera: los datos de entrada serían linealizados siguiendo las ecuaciones del modelo en cuestión, obteniendo las constantes respectivas a cada modelo, así como su coeficiente de correlación (R^2). Posteriormente, las constantes obtenidas por ajuste lineal, serían empleadas para el cálculo de los distintos modelos, y se compararían con datos teóricos.

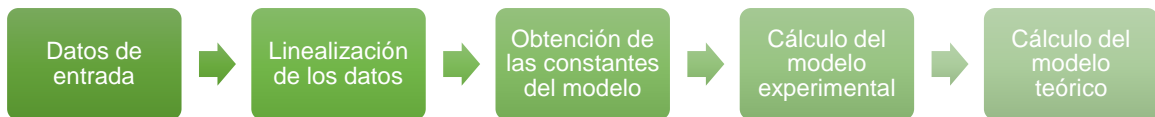


Fig. 6.4. Diagrama de procesos de cálculo del programa.

Finalmente, los resultados obtenidos serían presentados al usuario mediante gráficos. Para cada modelo seleccionado, se representarían los datos linealizados en un gráfico, y el modelo experimental y teórico en otro. Además, también cabría la posibilidad de representar los diferentes modelos disponibles en un mismo gráfico comparativo, para cada grupo de estudio mencionado anteriormente.

En resumen, una vez concretadas y delimitadas las funcionalidades de esta primera versión de un programa para el análisis de estudios experimentales de adsorción, la estructura inicial fue ampliada y definida, tal como se muestra a continuación:

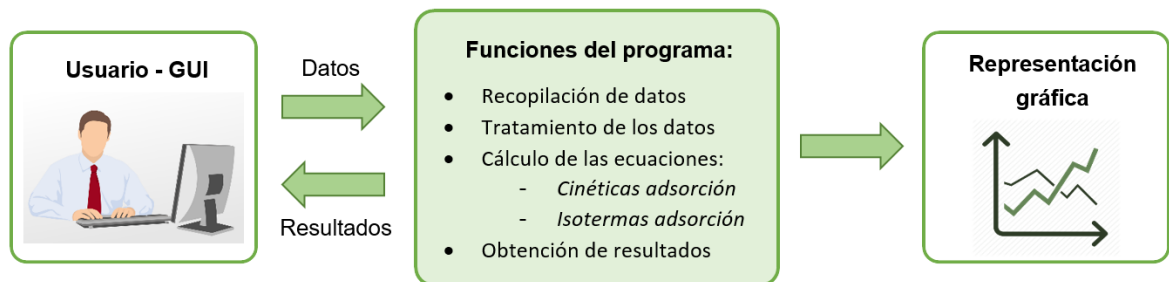


Fig. 6.5. Diagrama de estructura definida del programa.

6.3. Herramientas y paquetes usados

Para llevar a cabo las acciones que la aplicación deberá realizar, fueron necesarias un conjunto de herramientas o bibliotecas específicas para *Python*, puesto que éstas no vienen incluidas en su versión estándar, y su ausencia dificultaría en gran medida la elaboración de la aplicación objeto de este proyecto.

Por este motivo, se exponen a continuación los paquetes utilizados durante el desarrollo de la aplicación, con la descripción de los mismos y su función dentro del programa. Es necesario mencionar que todos ellos pueden ser obtenidos de manera gratuita.

6.3.1. Numpy y Scipy

Scipy es una librería de herramientas numéricas, científicas y de ingeniería para *Python* que es distribuida de forma libre. Este módulo confiere al lenguaje general de *Python* capacidades de cálculo numérico de gran capacidad, permitiendo así la optimización de funciones, integración, resolución de ecuaciones diferenciales ordinarias, estadística descriptiva y otros muchos aspectos. *Scipy* además incluye en su núcleo a *Numpy*, un módulo utilizado para el manejo de matrices y grandes conjuntos de valores numéricos.

Si desea ampliar la información, podrá encontrar la documentación completa de *Scipy* y *Numpy* en <http://docs.scipy.org/>.

6.3.2. Matplotlib

Matplotlib es probablemente el módulo más usado para la representación de gráficos y figuras bidimensionales a partir de datos contenidos en matrices y *arrays*. A través de esta biblioteca, es posible generar gráficos, histogramas, gráficos de barras, diagramas de dispersión, etc, con tan sólo unas pocas líneas de código. Además, contiene una gran variedad de posibilidades de edición y personalización de gráficos, como son la inclusión de títulos, fórmulas, anotaciones y leyendas, entre otros.

Dirigiéndose a su página web <http://matplotlib.org/> encontrará toda la información relativa a esta biblioteca, y donde dispondrá de numerosos tutoriales, ejemplos de gráficos y plantillas con su código de programación correspondiente.

6.3.3. PyQt5

PyQt es una adaptación de la biblioteca gráfica *Qt* para el lenguaje de programación *Python*. *Qt* utiliza de forma nativa el lenguaje de programación C++ pero puede ser empleado en otros lenguajes de programación a través de adaptaciones o *bindings*. Esta biblioteca multiplataforma es ampliamente usada para el desarrollo de aplicaciones con interfaz gráfica de usuario, pero también en el manejo de *XML*, las comunicaciones de red, bases de datos *SQL*, navegación web y otras tecnologías informáticas.

Actualmente existen dos versiones de *PyQt* que son: *PyQt4* y *PyQt5*. Sus desarrolladores recomiendan el uso de *PyQt5* puesto que se trata de la versión más reciente, y además, se

ha dejado de dar soporte al desarrollo de *PyQt4*. Por este motivo, en el desarrollo del programa se utilizará la última versión de *PyQt5* para *Windows*.

Si lo desea, puede descargar la versión más reciente en la página web de sus desarrolladores <https://www.riverbankcomputing.com/software/pyqt/download5>, donde también encontrará la documentación de referencia de esta biblioteca.

6.3.4. **cx_Freeze**

Cx_Freeze es un conjunto de *scripts* y módulos para la “congelación” de los códigos de programación de Python en ejecutables. Esta herramienta se utiliza para recopilar todos los archivos de código que forman el programa y crea un archivo ejecutable. Su funcionamiento es muy parecido a otras bibliotecas como *py2exe* y *pyInstaller*, pero a diferencia de estas dos herramientas, *cx_Freeze* es multiplataforma. Además también es compatible con las versiones de *Python 2.6* o superior, incluyendo *Python 3*.

Para más información acerca de la herramienta *cx_Freeze* consulte su página web <http://cx-freeze.sourceforge.net/>.

6.3.5. **Distribución de herramientas**

Una vez definidas las bibliotecas de soporte que se utilizarán, en la siguiente figura se muestra un esquema con su papel dentro del programa:

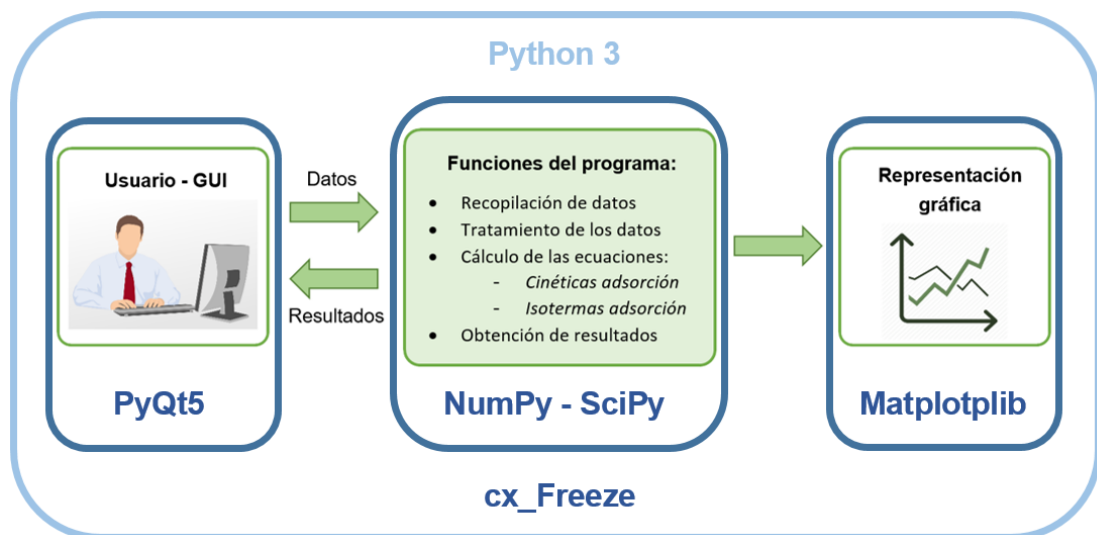


Fig. 6.6. Diagrama de bibliotecas utilizadas en el programa.

Teniendo clara la idea de funcionamiento del programa, se prosiguió con el desarrollo de cada uno de los bloques que componen el algoritmo de la aplicación.

6.4. Procesamiento de los datos

En este bloque se determinaron los cálculos que deberían ser realizados por parte del programa, una vez introducidos los valores experimentales de estudio, con el fin de obtener los resultados propuestos.

6.4.1. Datos de entrada

Primeramente, fue necesario concretar qué datos serían requeridos por el programa para que éste pudiera cumplir con su función. Se estableció que los datos de entrada se introducirían separadamente, en función de si los ensayos realizados hacían referencia a estudios de la cinética de adsorción, o bien, a las isotermas de adsorción. Los datos de entrada requeridos al usuario en el programa son los que siguen a continuación, distinguiéndolos en función del objeto de estudio:

Cinéticas de adsorción		Isotermas de adsorción													
Título del gráfico		Título del gráfico													
Volumen de disolución en [L]		Volumen de disolución en [L]													
Peso del sólido adsorbente en [g]		Peso del sólido adsorbente en [g]													
Área superficial del adsorbente en [m ² /g]		Área superficial del adsorbente en [m ² /g]													
Tabla de valores experimentales:		Tabla de valores experimentales:													
<table border="1"> <thead> <tr> <th>Tiempo en [h]</th> <th>Concentración de sólido [mol/dm³]</th> </tr> </thead> <tbody> <tr> <td>0*</td> <td>...</td> </tr> <tr> <td>...</td> <td>...</td> </tr> </tbody> </table>	Tiempo en [h]	Concentración de sólido [mol/dm ³]	0*	<table border="1"> <thead> <tr> <th>Concentración inicial [mol/dm³]</th> <th>Concentración en el equilibrio [mol/dm³]</th> </tr> </thead> <tbody> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>...</td> <td>...</td> </tr> </tbody> </table>	Concentración inicial [mol/dm ³]	Concentración en el equilibrio [mol/dm ³]		
Tiempo en [h]	Concentración de sólido [mol/dm ³]														
0*	...														
...	...														
Concentración inicial [mol/dm ³]	Concentración en el equilibrio [mol/dm ³]														
...	...														
...	...														
*Es importante que el primer valor de concentración de sólido introducido en la tabla sea a tiempo 0h, es decir, la concentración inicial.															
Selección del modelo a analizar: <ul style="list-style-type: none"> • Modelo de pseudo segundo orden • Modelo de Elovich • Comparación de modelos 		Selección del modelo a analizar: <ul style="list-style-type: none"> • Isoterma de Langmuir • Isoterma de Freundlich • Comparación de isotermas 													

Tabla. 6.1. Tabla de datos de entrada definidos para cada tipo de estudio.

Con el fin de que el programa pueda interpretar dichos datos, éstos deberán ser introducidos por el usuario de una manera determinada, teniendo en cuenta para cada uno de ellos el formato adecuado para poder ser procesados. El tipo de datos y su método de introducción se determinan a continuación:

Nombre de la variable	Tipo de dato								
Título del estudio	String → Cadena de caracteres (<i>Ejemplo:</i> "Goetita"/ "Hematita")								
Volumen de disolución en [L]	Float → Número decimal (<i>Ejemplo:</i> 1.00 / 1.00E+00)								
Peso del sólido adsorbente en [g]	Float → Número decimal								
Área superficial del adsorbente en [m ² /g]	Float → Número decimal								
<p>Tabla de valores experimentales:</p> <table border="1" data-bbox="308 1014 866 1328"> <thead> <tr> <th data-bbox="308 1014 587 1108">X</th> <th data-bbox="587 1014 866 1108">Y</th> </tr> </thead> <tbody> <tr> <td data-bbox="308 1108 587 1182">...</td> <td data-bbox="587 1108 866 1182">...</td> </tr> <tr> <td data-bbox="308 1182 587 1256">...</td> <td data-bbox="587 1182 866 1256">...</td> </tr> <tr> <td data-bbox="308 1256 587 1328">...</td> <td data-bbox="587 1256 866 1328">...</td> </tr> </tbody> </table>	X	Y	<p>Matriz ($i \times j$) de valores Float , donde: i = número de filas j = número de columnas (=2)</p> <p>Los valores se introducen conjuntamente para cada fila "i", usando como separador entre los valores de cada columna, el espacio " " .</p> <p><i>Ejemplo:</i> $i = 1 \rightarrow (1.00 \quad 2.00)$ $i = 2 \rightarrow (1.00E+0 \quad 2.00E-00)$</p>
X	Y								
...	...								
...	...								
...	...								
<p>Selección del modelo a analizar:</p> <p>A = 1 → valor de la variable correspondiente a los modelos de cinética de adsorción.</p> <p>A = 2 → variable correspondiente a los modelos de isotermas de adsorción</p> <p>numero_modelo → variable numérica que hace referencia a un modelo en concreto, en función del valor de A.</p>	<p>Integer o int → Número entero</p> <p>A cada modelo se le asignará una correspondencia con un número entero, a fin de poder identificarlo.</p> <p><i>Ejemplo:</i> Modelo de Elovich → A = 1 numero_modelo = 2 Isoterma de Langmuir → A = 2 numero_modelo = 1</p>								

Tabla 6.2. Tabla detallada del tipo de datos definidos para cada variable de entrada.

Cabe remarcar una vez más que, aunque gran parte de las variables del programa sean las mismas para cada uno de los dos tipos de estudio presentados (cinéticos y de isotermas), éstas se tratarán por separado, tal y como se estableció desde el principio, y se considerarán como variables independientes.

6.4.2. Transformación de los datos

Una vez recopilados los datos, el programa realizará un conjunto de transformaciones y adaptaciones de los mismos en función de las ecuaciones de cada modelo (cinético o de isotermas). El proceso para todos ellos es muy parecido, y sigue este orden:

Para cuantificar el proceso de adsorción, inicialmente se halla el parámetro q , que relaciona la cantidad de soluto adsorbido por unidad de adsorbente, teniendo en cuenta el área superficial del adsorbente:

$$q_{t,e} = \frac{(C_i - C_{t,e}) \cdot V_{disolución}}{m_{adsorbente}} \cdot \frac{1}{A_{sup}} \left[\frac{mol}{m^2} \right] \quad (\text{Ec. 6.1})$$

Este parámetro es calculado para cada valor de concentración introducido en la tabla de entrada. En caso de que el usuario desconozca el valor del área superficial (A_{sup}), éste tomará por valor 1 y el parámetro q será presentado en las unidades $\left[\frac{mol}{g} \right]$. A continuación, se presentan las transformaciones realizadas para cada tipo de estudio:

Tiempo [h]	Concentración [mol/dm ³]
...	...

Tiempo [h]	Parámetro q_t [mol/m ²] ó [mol/g]
...	...

Tabla. 6.3. Transformación de tabla de entrada para las cinéticas de adsorción.

Concentración inicial [mol/dm ³]	Concentración equilibrio [mol/dm ³]
...	...

Concentración equilibrio [mol/dm ³]	Parámetro q_e [mol/m ²] ó [mol/g]
...	...

Tabla. 6.4. Transformación de tabla de entrada para las isotermas de adsorción.

6.4.3. Linealización de las ecuaciones y obtención de los parámetros

Una vez transformados los valores de la tabla introducida por el usuario, se emplearán los valores de q obtenidos para realizar una linealización de los datos siguiendo las ecuaciones del modelo en cuestión. Con los resultados obtenidos, será posible obtener la pendiente de la recta resultante, su coordenada en el origen y el factor de correlación entre los datos introducidos. Estos parámetros permitirán calcular las constantes respectivas de cada modelo, así como evaluar la validez del modelo seleccionado con los datos de estudio.

Véase a continuación, un esquema del proceso, poniendo como ejemplo el utilizado para el modelo cinético de pseudo segundo orden:

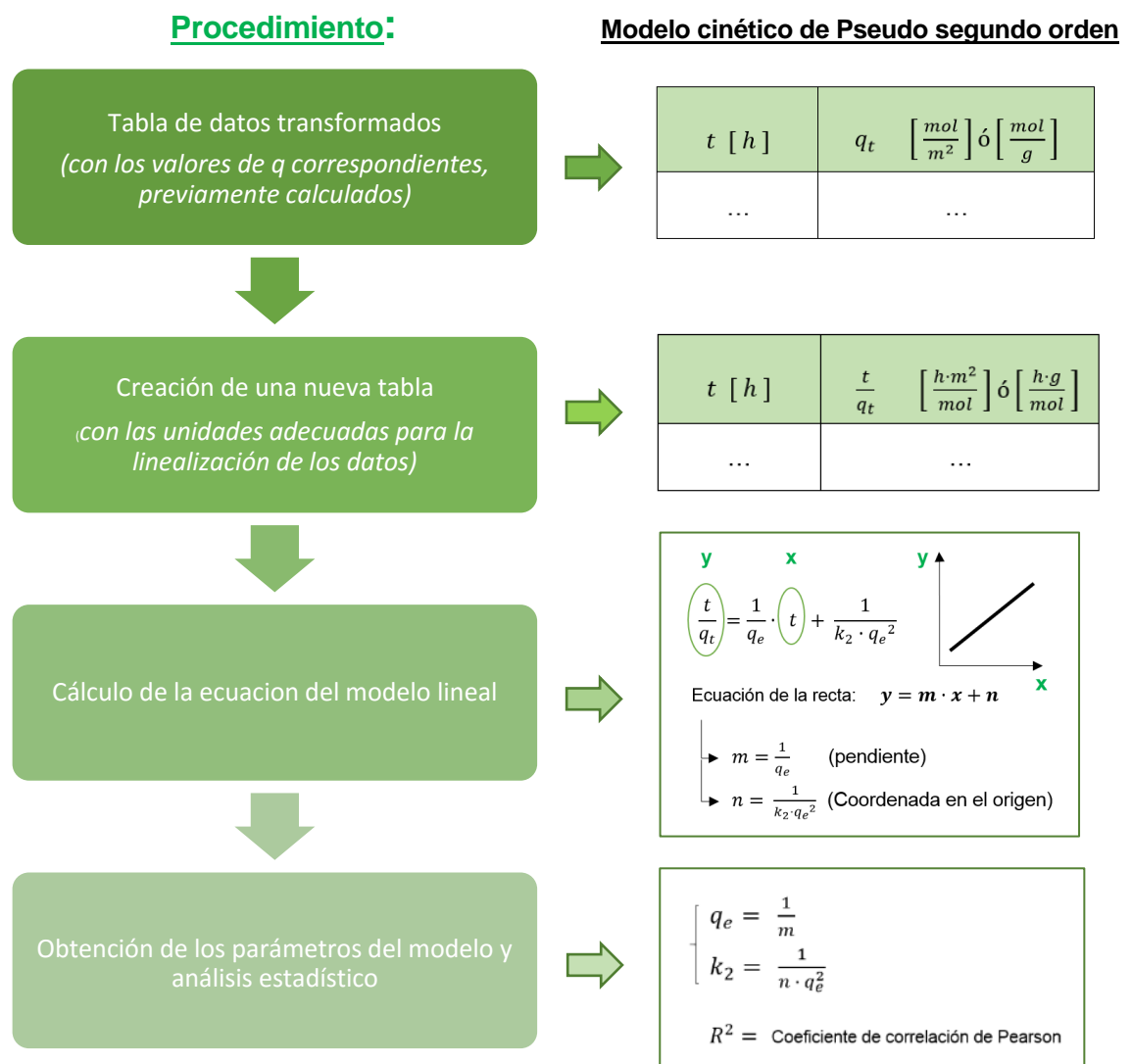


Fig. 6.7. Diagrama del procedimiento interno de cálculo del programa. Ejemplo de cálculo para el caso del modelo cinético de pseudo segundo orden.

6.4.4. Obtención del modelo teórico

Finalmente, y una vez obtenidas las constantes relativas a cada modelo, éstas se utilizan en el cálculo de la ecuación teórica que describe dicho modelo. En este caso, los datos de salida del programa se presentan en forma de tabla con los siguientes valores prefijados en función del tipo de estudio de adsorción:

- En los estudios de la **cinética de adsorción**, la tabla resultante tiene las siguientes unidades:

$t [h]$	$q_t \left[\frac{mol}{m^2} \right] \text{ ó } \left[\frac{mol}{g} \right]$
...	...

Tabla. 6.5. Tabla de resultados obtenidos para el caso de cinéticas de adsorción.

- En cuanto a las **isotermas de adsorción**, la tabla de salida contiene las variables que siguen a continuación:

$C_e \left[\frac{mol}{dm^3} \right]$	$q_t \left[\frac{mol}{m^2} \right] \text{ ó } \left[\frac{mol}{g} \right]$
...	...

Tabla. 6.6 .Tabla de resultados obtenidos para el caso de isotermas de adsorción.

6.5. Representación gráfica de los resultados

Una vez obtenidos los resultados en formato numérico, además de calcular las distintas constantes de cada modelo, y los valores en forma de tablas, se procedería con la graficación de los mismos. En este caso se usaría la biblioteca de Python *Matplotlib*, ya que ofrece la posibilidad de representar gráficos en 2D, y cuenta con numerosas posibilidades de edición de los mismos.

Para cada modelo seleccionado, con independencia del tipo de estudio realizado (cinético o de isotermas), se representarían dos gráficos diferentes en la misma pantalla:

- El primero, situado a la izquierda, contendría la representación lineal del modelo, y sería obtenido a partir del ajuste de los datos experimentales con la ecuación del modelo linealizado. En él, deberían aparecer conjuntamente la ecuación de la recta y el coeficiente de correlación R^2 obtenidos. Además, también serían visibles las constantes obtenidas para el modelo en cuestión, así como su ecuación en forma linealizada.
- A la derecha, se observaría otro gráfico en el que se compararían las curvas obtenidas con datos experimentales frente al modelo teórico, dadas las constantes obtenidas en el caso concreto de estudio.

La apariencia del gráfico final, incluyendo ambas representaciones en la misma ventana, es la que se muestra en la imagen siguiente (Fig 6.8.), poniendo por ejemplo un caso de estudio de la cinética de adsorción:

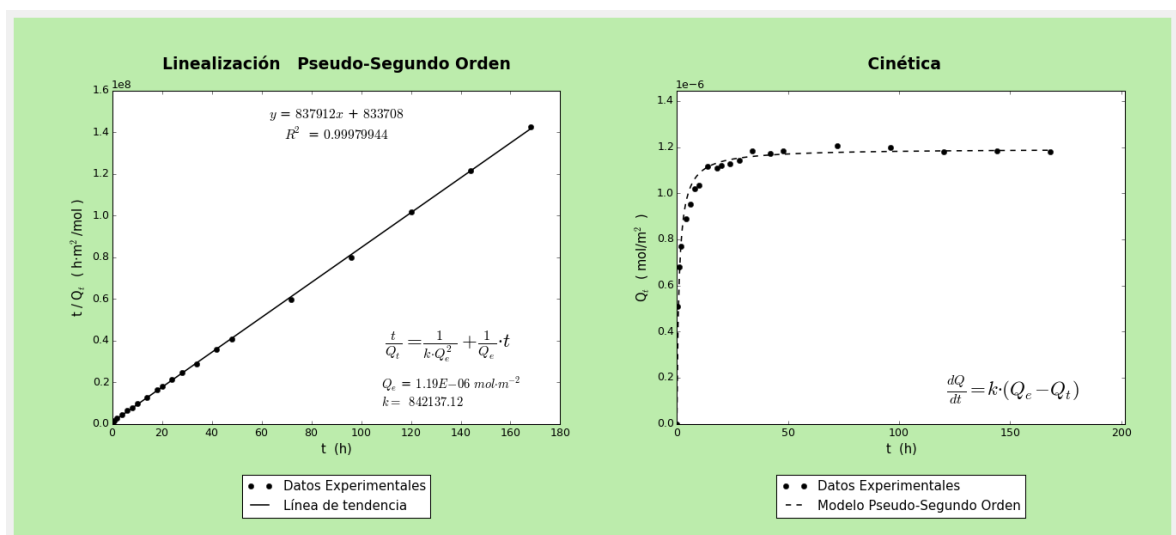


Fig. 6.8. Formato y apariencia de los gráficos obtenidos a través del programa, poniendo como ejemplo el modelo cinético de pseudo segundo orden.

Además, tanto en los estudios de cinética como en los referentes a las isotermas de adsorción, será posible visualizar a parte otro gráfico, en el que se representarán todos los modelos disponibles comparándolos con los datos experimentales. Este último gráfico pretende ayudar al usuario a determinar a qué modelo se ajustan mejor los datos recopilados durante los ensayos.

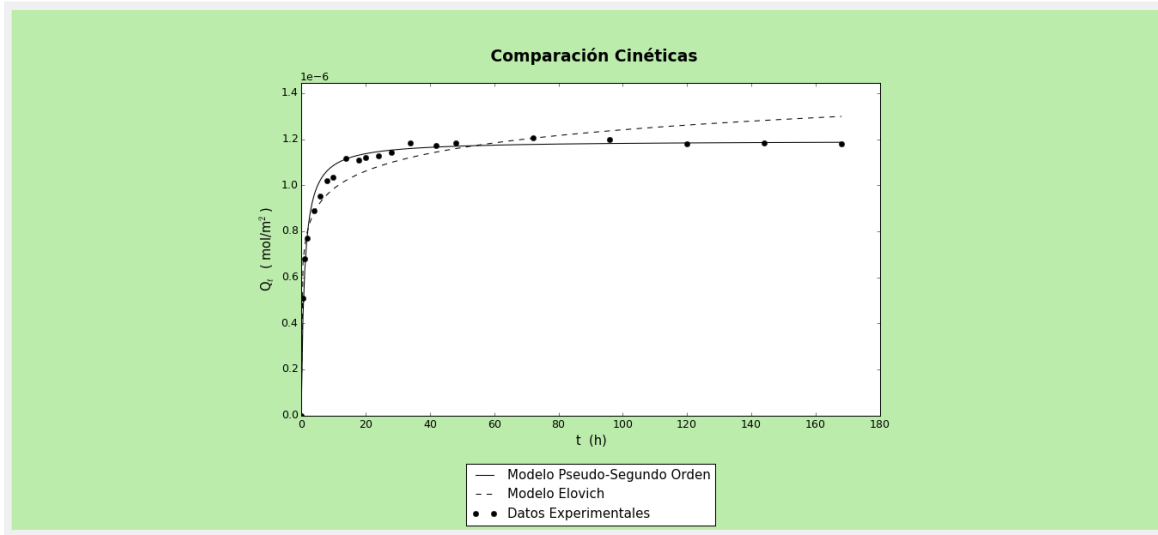


Fig. 6.9. Aspecto del gráfico comparativo entre modelos obtenido con el programa.

6.6. Creación de la Interfaz gráfica (GUI)

Llegados a este punto, en el que el programa ya cumplía con el propósito de realizar los cálculos que se plantearon desde un inicio, hubo la necesidad de crear una interfaz gráfica de usuario, con tal de facilitar la interacción entre el usuario y el programa.

Este apartado del proyecto se dividió en dos partes: la primera hace referencia al diseño de la interfaz gráfica creada mediante la aplicación *Qt Creator* (incluida en la librería *PyQt*) y la segunda, trata sobre la implementación de la interfaz de usuario con el algoritmo del programa.

6.6.1. Diseño de la Interfaz gráfica de usuario

6.6.1.1. Qt Creator y primera interfaz

La creación de la interfaz gráfica, como se ha mencionado en el apartado anterior, se elaboró mediante el programa *Qt Creator*. Esta herramienta permite diseñar la “pantalla” de la aplicación, a través de la cual el usuario podrá comunicarse con el programa. La principal ventaja que ofrece es que el diseño de la interfaz se realiza de manera gráfica, y paralelamente *Qt Creator* elabora el código correspondiente en lenguaje de programación. De este modo, se ahorra mucho tiempo en la confección de la interfaz, puesto que de lo contrario, sería necesario escribir el código de programación manualmente.

Tras practicar y entender mejor el funcionamiento de esta herramienta, se creó una primera interfaz inicial formada por una ventana, y posteriormente se fueron añadiendo distintos elementos que podrían ser de utilidad en este proyecto [17]. Entre ellos, destacan: las etiquetas, botones, editores y campos de texto. En el siguiente apartado se explicará que son cada uno de estos elementos, y sus principales funciones.

A continuación, se muestra una imagen (Fig 6.10) en la cual se puede observar el aspecto de la aplicación *Qt Creator*, así como la interfaz inicial creada:

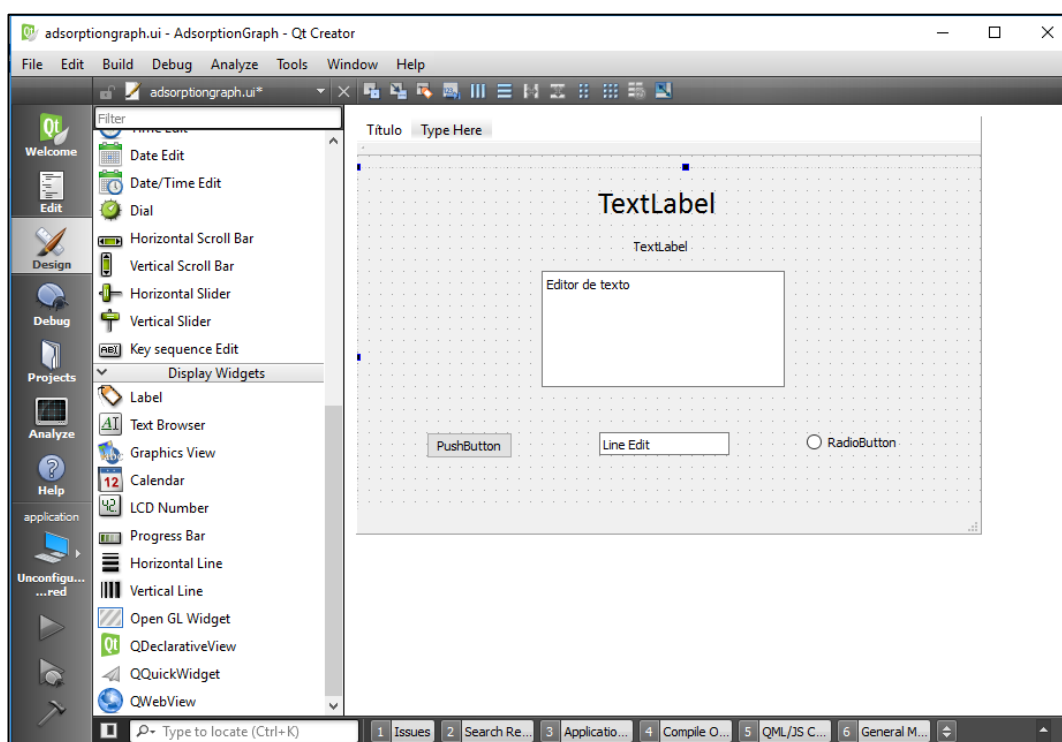


Fig. 6.10. Aspecto de la interfaz de diseño del programa *Qt Creator*.

Para añadir cada elemento, tan sólo es necesario arrastrar los *widgets* de la barra de herramientas que aparece a la izquierda y situarlos en la ventana de diseño. Posteriormente, es posible editar su forma, tamaño y color al deseo del diseñador.

Cabe mencionar que el código de la interfaz creada través de la aplicación *Qt Creator* es extraído con la extensión de interfaz gráfica (*.ui*). No obstante, este archivo se puede convertir fácilmente a código de *Python* (*.py*) mediante una simple línea de comando en la terminal del sistema (*cmd*).

6.6.1.2. Widgets implementados

A continuación, se describen las principales clases de widgets que han sido utilizadas para la elaboración de la interfaz de usuario:

- **QLabel:** También conocido como *Etiqueta*, esta clase de widget permite la escritura de texto o imágenes. Se utiliza principalmente para la inclusión de títulos, descripciones, y otros elementos de texto de carácter informativo para el usuario.

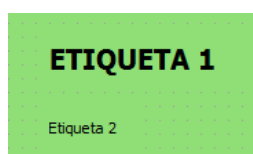


Fig. 6.11. Aspecto widget *QLabel*.

- **QPushButton:** Este widget proporciona un botón de comando. Es probablemente el elemento más comúnmente utilizado en cualquier interfaz gráfica de usuario. Pulsando en el botón se pueden realizar diversas acciones. Entre los casos más típicos de botones destacan el “*Aceptar*”, “*Salir*”, “*Ayuda*”, etc.

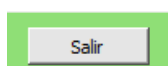


Fig. 6.12. Aspecto widget botón (*QPushButton*).

- **QRadioButton:** Se trata de un botón de opción que se puede activar (marcado) o desactivar (sin marcar). Los botones de radio normalmente se presentan al usuario en grupo, y acompañados de una etiqueta de texto. Estos representan una selección entre diversas opciones. Si el usuario selecciona otro botón, el botón marcado previamente se desconecta.

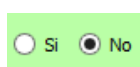


Fig. 6.13. Aspecto widget *QRadioButton*.

- **QPlainTextEdit:** Esta clase provee un widget que es utilizado para editar y mostrar texto sin formato. Este editor de texto está optimizado para manejar documentos de gran tamaño y para responder con rapidez a la entrada del usuario. Además incluye una colección de funciones útiles como “*Cortar*” y “*Pegar*”, “*Deshacer*” y “*Rehacer*”, etc.

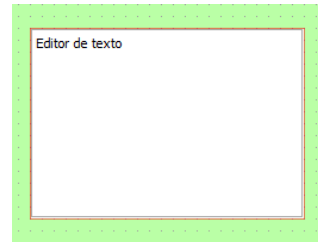


Fig. 6.14. Aspecto widget de editor de texto plano (*QPlainTextEdit*).

- **QLineEdit:** Similar al *QPlainTextEdit*, este widget es un editor de texto pero de una sola línea, que permite al usuario introducir y editar texto sin formato.

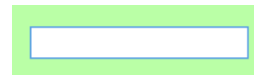


Fig. 6.15. Aspecto widget editor de línea de texto (*QLineEdit*).

- **QTabWidget:** La clase *QTabWidget* proporciona una barra de pestañas con su respectiva “área de página” que se utiliza para mostrar los elementos relacionados de cada pestaña. Además, ésta última está asociada con un widget o conjunto de widgets diferentes. Al seleccionar una de las pestañas, se muestra el área de la página correspondiente, quedando ocultas el resto.

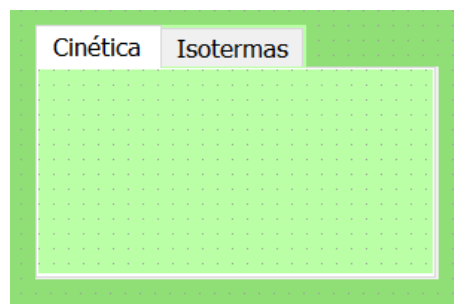


Fig. 6.16. Aspecto widget barra de pestañas (*QTabWidget*).

6.6.1.3. Aspecto final de la interfaz

Una vez definidos los widgets que integraría la interfaz, se procedió con la organización de estos en la ventana de la aplicación, a fin de disponerlos de manera ordenada y visual para facilitar la tarea de introducción al usuario.

Siguiendo las pautas establecidas desde el inicio de la creación del programa, se definió que la ventana de la interfaz englobaría dos pestañas independientes: una para los estudios de cinética, y otro para los relacionados con las isotermas. Dentro de cada pestaña, se dispondría de un formulario que incluiría los distintos campos que el usuario debería rellenar para cada tipo de ensayo.

En cada formulario, se establecería un orden de introducción de los datos y de interacción con los diferentes widgets, con descripciones para facilitar el correcto uso del programa:

1. En primer lugar el usuario deberá rellenar los campos correspondientes a los siguientes parámetros: *Nombre elemento* (título de los gráficos), *Volumen de disolución*, *Peso del sólido* y *Área superficial*, éste último en función de si se conoce su valor.
2. El segundo paso será rellenar la tabla con los datos obtenidos en los ensayos.
3. Seleccionar una de las opciones entre los diferentes modelos disponibles.
4. Interactuar con los botones que ejecutan las diferentes funcionalidades del programa: *Validar Datos*, *Calcular*, *Mostrar Gráfico* y *Guardar gráfico*.

A continuación, se muestran imágenes del diseño de la ventana principal de la aplicación, en la que se puede observar el contenido de cada una de las pestañas, y la disposición de los diferentes widgets en ella:

AdsorptionGraph v1.0 - [Preview]

Cinética | Isotermas

1. Datos a introducir

Nombre elemento :

Volumen disolución [L] :

Peso del sólido [g] :

Conoce el valor del área superficial ? Si No

Área superficial [m2/g] :

2. Rellene los valores requeridos en la tabla:

Tiempo [h]	Cs [mol/L]
--------------	--------------

3. Especifique el modelo que desea representar :

Cinética Pseudo-Segundo Orden Cinética Elovich Comparación Cinéticas

4. Valide los datos introducidos:

5.

6.

7.

Fig. 6.17. Aspecto de la interfaz gráfica: Formulario cinética de adsorción.

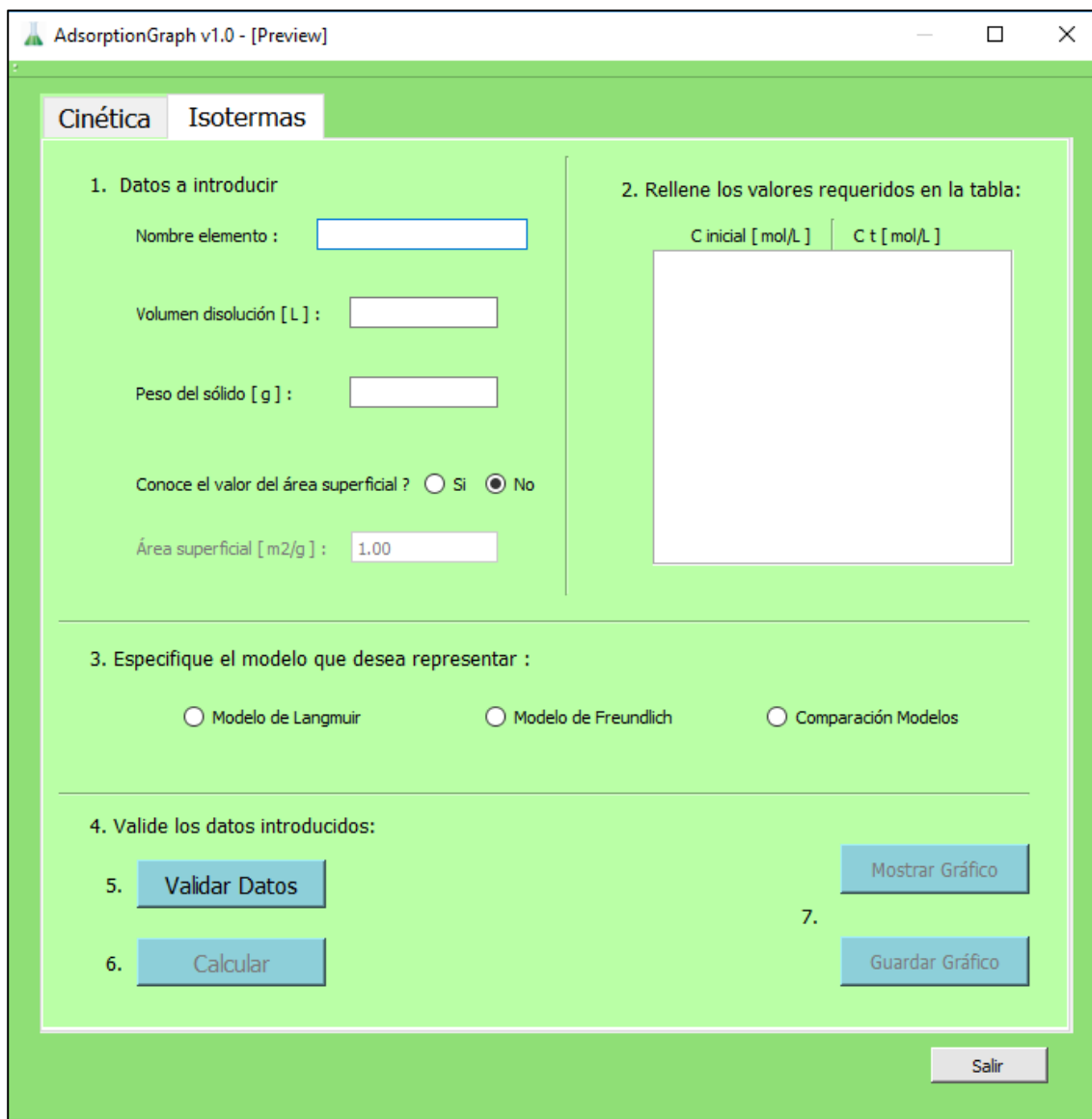


Fig. 6.7. Aspecto de la interfaz gráfica: Formulario isotermas de adsorción.

6.6.2. Conexión de la interfaz con el algoritmo del programa

Tras haber definido el diseño de la interfaz, surgió una nueva problemática: cómo conectar los distintos elementos de la interfaz con el algoritmo principal del programa. Hasta este momento, se había creado una interfaz con varios widgets, pero estos no tenían ninguna función asignada.

El siguiente paso en la construcción del programa consistía entonces en conseguir que los datos que el usuario había rellenado en los campos de texto fueran almacenados en el programa, y que al pulsar cada botón se ejecutaran las acciones pertinentes. Para que esto fuera posible fue necesario recurrir a una característica de *PyQt*, concretamente al mecanismo de señales y slots, explicado en el apartado 7.6.2.2.

6.6.2.1. Almacenamiento de los datos contenidos en campos de texto

El almacenamiento de cada uno de los datos introducidos en los campos y en el editor de textos, se llevaría a cabo al hacer clic al botón “*Validar datos*”. Una vez pulsado este botón, el programa se encargaría de leer uno a uno el valor de todas las casillas, y le asignaría el valor introducido a cada una de sus variables correspondientes en el programa.

Para tratar los datos en forma de tabla del editor de texto, la situación fue algo compleja, dado que no se permitía leer directamente su contenido y almacenarlo en forma de matriz.

Por este motivo, se optó por crear una función específica llamada “*LeerMatriz()*”, a través de la cual se copiaría el contenido íntegro del editor de texto automáticamente en un bloc de notas (.txt). A continuación, tras guardar el archivo con nombre “*datosTabla.txt*”, el programa lo abriría internamente de nuevo y leería cada una de las filas de texto contenidas, atribuyendo los valores a su posición relativa dentro de una matriz. Para distinguir entre los valores de cada columna pertenecientes a una misma fila, se asignó como separador la barra espaciadora y la tecla tabulador.

6.6.2.2. “Signals” y “Slots”

El mecanismo de señales y slots es una de las características más importantes de la librería de *PyQt*. Éstas se usan para las comunicaciones entre los distintos objetos de una interfaz gráfica. Generalmente, la señal (signal) parte de un objeto "emisor" y llega a un objeto "receptor". Éste último decide si ejecuta un slot (o acción) y finaliza el proceso, o emite una nueva señal que propaga el evento hacia otro objeto receptor. El proceso se puede repetir.

Como ejemplo sencillo, imagine un botón de "Salir". Cuando el usuario lo pulsa, el botón genera la señal *clicked()*, y en respuesta a esa señal el sistema ejecuta un slot *close()* que cierra la ventana de la aplicación [18].

Procediendo de esta misma manera para cada uno de los botones creados, y siguiendo el mecanismo de señales y slots, se definieron sus funciones correspondientes descritas a continuación:

- **Radio botones “Sí” y “No”:** Ambos botones están conectados el uno con el otro, y hacen referencia a si se conoce o no, el valor del área superficial del adsorbente. En caso afirmativo (seleccionando el botón “Sí”), se habilita la edición del campo de texto correspondiente al valor de área superficial. En caso de seleccionar la opción “No”, este campo de texto queda desactivado y al área superficial se le asigna automáticamente el valor 1. Su función principal es la de concretar en qué unidades se presentarán los resultados de los gráficos, condicionados por la participación de la variable área superficial en los cálculos.

The figure shows two side-by-side screenshots of a web form. Both have a light green background. The left screenshot shows the question 'Conoce el valor del área superficial?' with radio buttons for 'Si' (unselected) and 'No' (selected). Below it is a text input field labeled 'Área superficial [m2/g]:' containing the value '1.00'. The right screenshot shows the same question but with 'Si' selected and 'No' unselected. The text input field below it also contains '1.00'.

Fig. 6.19. Funcionalidad botones de selección “Sí”/”No”

- **Radio botones de elección del modelo:** Según el botón del modelo que se seleccione, se asigna un valor numérico correspondiente a dicho modelo, para que cuando se ejecute el algoritmo principal, tenga en cuenta cuál es el modelo que debe ser calculado.

The figure shows a horizontal row of three radio buttons on a light green background. The first button, labeled 'Cinética Pseudo-Segundo Orden', is selected. The second button is labeled 'Cinética Elovich' and the third is 'Comparación Cinéticas'. Both are unselected.

Fig. 6.80. Botones de elección: modelos cinéticos.

The figure shows a horizontal row of three radio buttons on a light green background. The first button, labeled 'Modelo de Langmuir', is unselected. The second button, labeled 'Modelo de Freundlich', is selected. The third button, labeled 'Comparación Modelos', is unselected.

Fig. 6.21. Botones de elección: modelos cinéticos.

- **Botón “Validar datos”:** Este botón posee múltiples funciones. En primer lugar, tal y como se ha comentado anteriormente, cuando se hace clic sobre él todos los valores introducidos en los campos de texto de la interfaz son almacenados y asignados a su variable correspondiente. Posteriormente, si dichos valores han sido introducidos en el formato correcto, se activa el botón “Calcular”.

Además, si los botones “Mostrar gráfico” y “Guardar Gráfico” se encuentran activos, los desactiva para que no puedan ser pulsados.

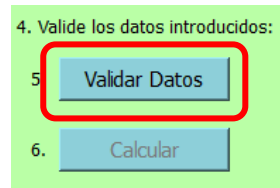


Fig. 6.22. Botón “Validar Datos”.

- **Botón “Calcular”**: Inicialmente este botón se encuentra desactivado, y tan sólo se activa si se reúnen las condiciones necesarias descritas para el botón “Validar datos”. Una vez pulsado, comienza la ejecución del algoritmo encargado de procesar los datos y realizar los cálculos para el modelo seleccionado. Si la ejecución del algoritmo se completa sin incidencias, entonces se activan también los botones “Mostrar Gráfico” y “Guardar Gráfico”.

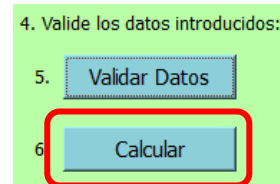


Fig. 6.23. Botón “Calcular”.

- **Botón “Mostrar Gráfico”**: Una vez activado, pulsando este botón se abre una nueva ventana con el gráfico representado, a partir de los datos introducidos y del modelo seleccionado.

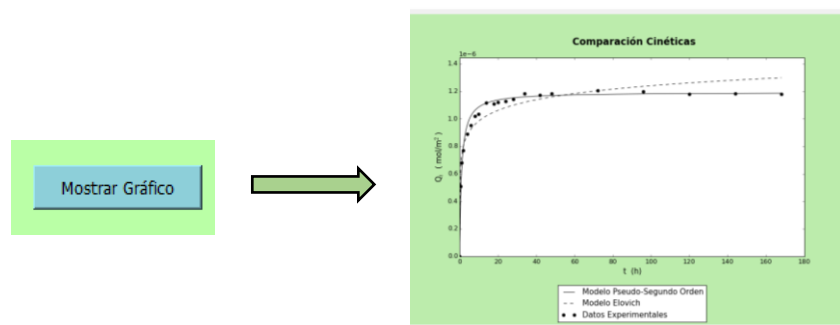


Fig. 6.24. Botón “Mostrar Gráfico” y funciones.

- **Botón “Guardar Gráfico”**: Una vez activado, pulsando este botón se abre un cuadro de dialogo que permite guardar una imagen (en formato .jpg o .png) del gráfico obtenido, en el directorio del ordenador que se desee. También es posible editar el nombre con el que será guardado dicha imagen.

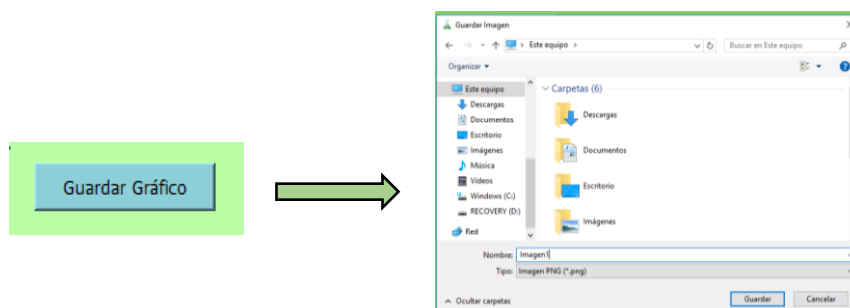


Fig. 6.25. Botón “Guardar gráfico” y funciones.

- **Botón “Salir”**: Tal y como se ha comentado anteriormente, al pulsar este botón se cierra la ventana del programa, y se detiene la ejecución del algoritmo.

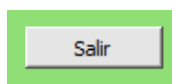


Fig. 6.26. Botón “Salir”.

6.7. Empaquetamiento del programa y creación de un instalador

Finalmente, tras conseguir enlazar la interfaz de usuario con el algoritmo principal, se obtuvo una primera versión completa de la aplicación que funcionara correctamente. Pero había un problema: el programa tan sólo podía ejecutarse desde el ordenador en el que se había programado.

6.7.1. Archivo ejecutable *.exe

Para poder ejecutar el programa en otros ordenadores con sistema operativo *Windows*, hacía falta convertirlo a un archivo ejecutable *.exe. Para ello, se eligió el programa *Cx_Freeze*, el cual a partir de los archivos que contienen el código del programa y de las bibliotecas necesarias, genera un archivo con todo lo necesario para poder ejecutar el programa en

cualquier ordenador con Windows, aunque éste no disponga de *Python* o de las bibliotecas instaladas.

Para generar este archivo ejecutable, primero hubo que descargar para *Windows* el *Python* 3.5 y después las bibliotecas necesarias que se habían usado durante el desarrollo del programa. Éstas eran *numpy*, *scipy*, *matplotlib* y *Pyqt4*. Con todas ellas descargadas y siguiendo un video-tutorial, fue posible generar el archivo que permitía su ejecución total en *Windows*.

El único inconveniente era el peso del archivo *.exe, puesto que superaba los 200Mb y tardaba mucho en iniciarse ejecutándolo desde otro ordenador.

6.7.2. Creación del instalador para *Windows* *.msi.

Tras investigar acerca de cómo reducir el tamaño del archivo resultante, se encontró otra vía de empaquetamiento de la aplicación a través de *Cx_Freeze*. En esta ocasión, el programa generaba una carpeta conteniendo todos los archivos necesarios para poder ejecutar el programa en otro ordenador, incluyendo el ejecutable. De este modo, la carpeta creada continuaba ocupando unos 170Mb, pero esta vez el archivo *.exe que incluía no superaba los 2Mb.

Ésta fue una gran mejora debido a que agilizaba la ejecución y funcionamiento del programa. Por el contrario, se hacía menos cómodo para el usuario ya que aumentaba la complejidad para copiar el programa creado en su ordenador.

Por último, tras insistir y profundizar en este tema, se encontró una solución óptima mediante *Cx_Freeze*, que consistía en generar un archivo en formato de instalador para *Windows* (*.msi). Éste se encargaría de crear automáticamente la carpeta anteriormente mencionada, en el directorio del ordenador de destino deseado. Además una vez instalado el programa, a través de este mismo instalador era posible reparar o desinstalar el programa. Así, se consiguió también reducir el tamaño del archivo final comprimido hasta los 60Mb, manteniendo el peso del ejecutable en 2Mb, facilitando así su distribución.

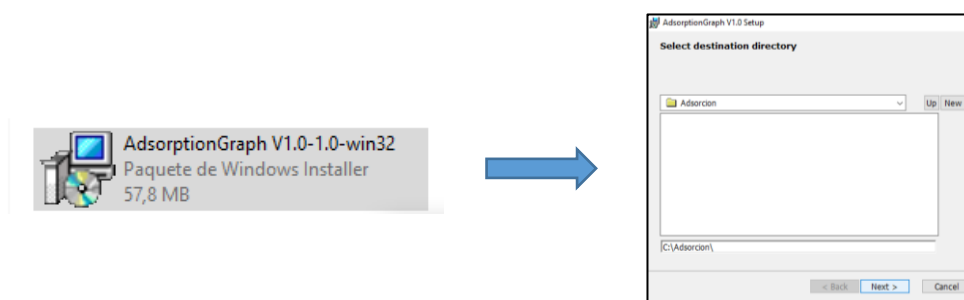


Fig. 6.27. Archivo y menú del instalador.

7. Otras implementaciones

7.1. Icono de la aplicación

Una de las primeras mejoras implementadas fue añadir un icono a la aplicación. La imagen elegida consistía en un Erlenmeyer haciendo referencia a la temática del programa: los ensayos experimentales en química. Además, la tonalidad verde del icono iba a juego con la paleta de colores seleccionada para el menú del programa. Su implementación fue posible una vez más gracias a *Qt Creator*, donde existe la opción de incluir una imagen como icono del programa.

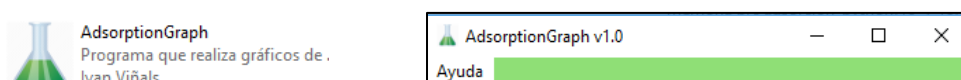


Fig. 7.1. Apariencia del icono de la aplicación.

7.2. Mensajes de aviso al usuario

Esta implementación surgió con el fin de ayudar al usuario a hacer un correcto uso del programa. De este modo, cuando el usuario no rellena correctamente los campos necesarios, o se olvida de completar alguno de ellos, el sistema responde con un mensaje alertando de lo sucedido. Si existen campos incompletos, al pulsar el botón “*Validar Datos*”, se indica lo siguiente: “*Revise los datos introducidos*”. Además también aparece un mensaje de error en cada campo incompleto, para informar al usuario de cuál es el dato que falta por completar o corregir.

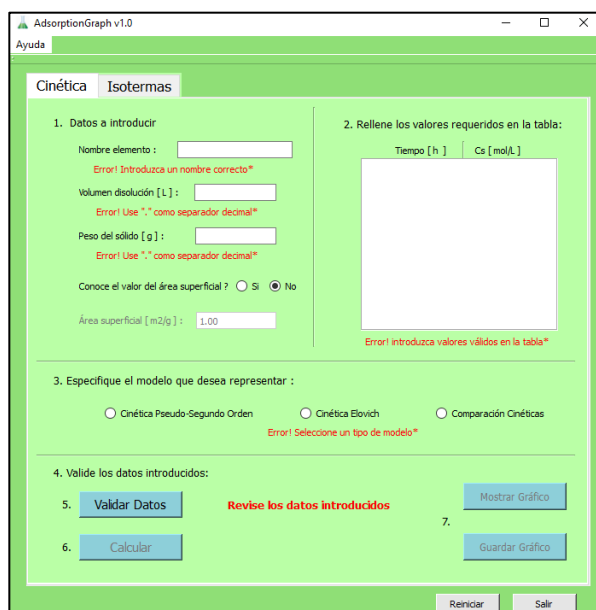


Fig. 7.2. Mensajes de error de la aplicación.

A medida que el usuario va añadiendo los datos que faltan, estos mensajes desaparecen al pulsar nuevamente en “*Validar Datos*”. Una vez todos los datos estén correctamente introducidos, tras volver a pulsar el botón para validar los datos, el mensaje que aparece es el siguiente: “*Los datos introducidos son válidos*”.

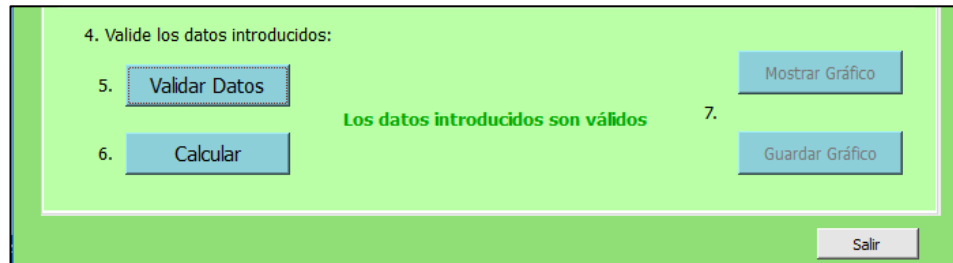


Fig. 7.3. Mensaje que aparece cuando los datos introducidos son válidos.

7.3. Incrustación de los gráficos en una nueva ventana de la interfaz

La aplicación creada en el proyecto tiene como objetivo presentar los resultados de los estudios realizados en forma de gráficos. Inicialmente, los gráficos que se obtenían no podían modificarse, tan sólo visualizarse en una ventana a parte, y ser exportados en formato de imagen.

Por consiguiente, este apartado surgió con la finalidad de que los gráficos obtenidos fueran editables por parte del usuario, por lo menos a un nivel básico. De este modo, se incluyó una barra de herramientas en la parte inferior del gráfico (Fig. 7.4.) que permitía realizar cambios sobre el mismo, entre los que destacan: redimensionar, ampliar/reducir, mover, guardar, etc., con tal que el usuario pudiera personalizarlos.

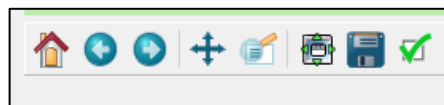


Fig. 7.4. Barra de herramientas para la edición de los gráficos.

7.4. Botón Reiniciar

La implementación del botón Reiniciar surgió para facilitar y agilizar el trabajo al usuario. Al pulsar dicho botón, todos los campos introducidos quedaban vacíos y los mensajes de aviso desaparecían, dejando así el programa en su estado inicial. De este modo, se podría empezar un nuevo caso de estudio sin tener que perder tiempo eliminando uno a uno todos los campos.

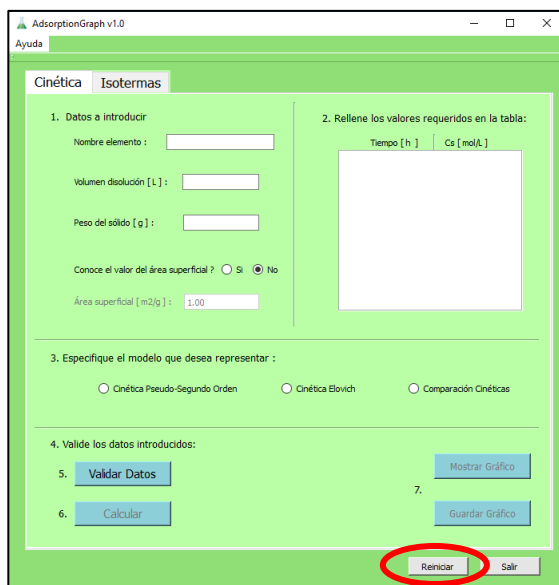


Fig. 7.5. Botón “Reiniciar” incrustado en la interfaz gráfica.

7.5. Barra de herramientas: Instrucciones.

Este apartado se creó con tal de solventar cualquier duda al usuario acerca del funcionamiento del programa. Por ello, se añadió una barra de herramientas en la parte superior de la ventana que contenía un apartado con el nombre “Ayuda”. Pulsando dicho apartado, aparece una opción llamada “Instrucciones”, en la cual haciendo ‘clic’ se abre un diálogo en una nueva ventana, conteniendo todas las instrucciones necesarias para poder hacer un correcto uso del programa.

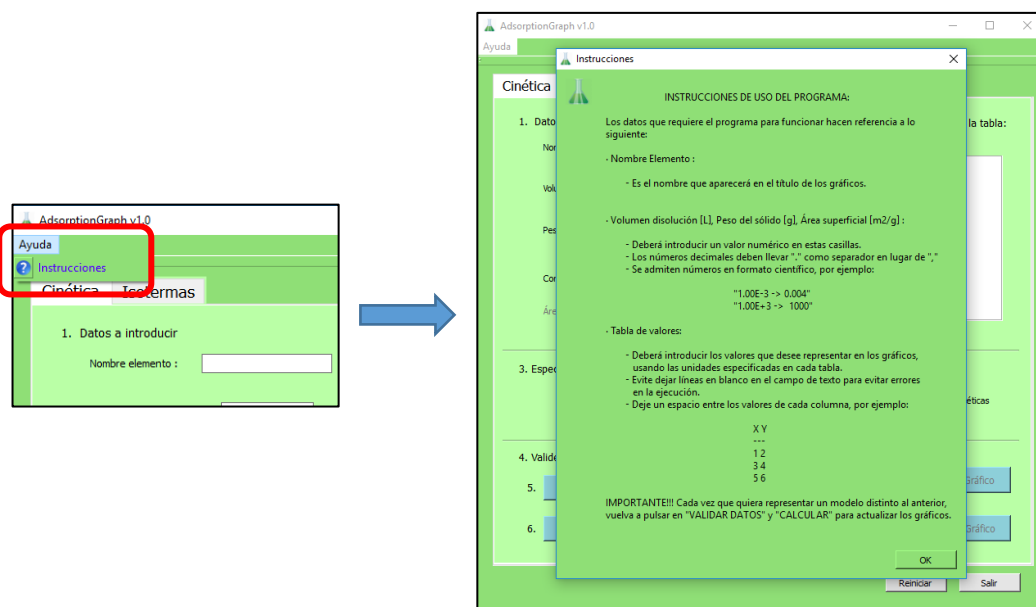


Fig. 7.6. Menú de instrucciones en la ventana principal del programa.

Este mismo tipo de opción se implementó también en la ventana correspondiente a los gráficos representados. De esta manera, las instrucciones permitían informar sobre la edición y la navegación a través de los gráficos.

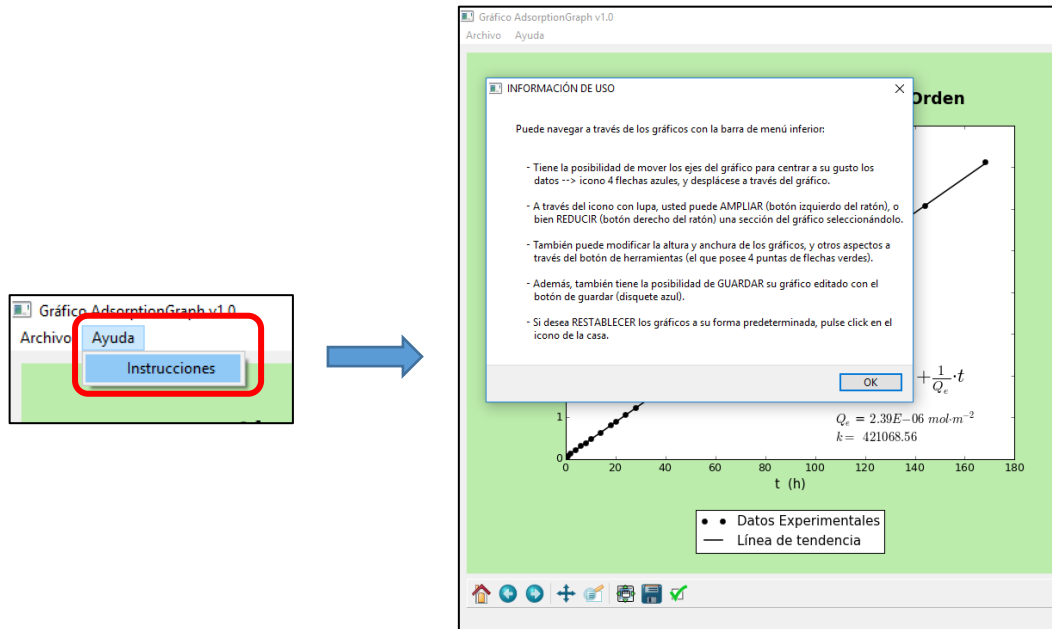


Fig. 7.7. Menú de instrucciones en la ventana de gráficos.

8. Vías de mejora y ampliación

Una vez terminada la versión básica del programa creado en este proyecto, donde los datos se tratan de manera eficaz y se obtienen los resultados esperados, interesaba plantear algunas vías de mejora, con el fin de abrir nuevos caminos para poder continuar en el desarrollo de la aplicación, y disponer en un futuro de una mayor cantidad de herramientas que lo hicieran extensible a más tipos de estudios.

A pesar de que se ha conseguido implementar una gran variedad de herramientas para el uso del programa, existen muchas otras que podrían resultar muy útiles con tal de ampliar las funcionalidades ya desarrolladas. Debido a que éste fue creado con la intención de poder ser ampliado en un futuro, en los siguientes apartados se proponen algunas de las opciones de mejora más interesantes que surgieron durante su creación. Tras estudiar la viabilidad para ser añadidas al programa actual, fueron descartadas por el trabajo extra que suponía su inclusión.

8.1. Ampliación del análisis estadístico descriptivo

El análisis estadístico que se realiza en el programa sirve para determinar la correlación existente entre los datos introducidos y la ecuación de la recta relativa a los distintos modelos presentados. En éste solamente se contemplan los siguientes parámetros, indispensables para discernir entre la existencia o no de un ajuste lineal de los datos: la pendiente de la recta, sus coordenadas en el origen y el factor de correlación de Pearson (R^2).

No obstante, existen otros parámetros estadísticos que podrían resultar de gran interés de cara a una futura ampliación del programa, con tal de ofrecer al usuario un análisis descriptivo más completo. Entre ellos, sería útil incluir los que se mencionan a continuación:

- Número de datos (n)
- Media (\bar{x})
- Desviación estándar (s)
- Desviación de la media ($\frac{s}{\sqrt{n}}$)
- Máximos y mínimos

La mayoría de estos elementos pueden ser calculados fácilmente gracias a la biblioteca de *SciPy* (<https://www.scipy.org>), aunque también es posible obtenerlos a través de otras bibliotecas como *matplotlib* (<http://matplotlib.org>).

8.2. Implementación de otros modelos y tipos de estudio

Una gran mejora que ampliaría en gran medida el uso y la utilidad del programa sería la inclusión de más modelos para los estudios de cinética de adsorción y las isothermas. Así, sería posible comparar los datos con un mayor número de modelos, y en consecuencia, los resultados ofrecerían soluciones más contrastadas. Algunos de los más representativos ya han sido expuestos anteriormente en el trabajo, y gracias a la estructuración del programa resultaría una tarea relativamente fácil añadirlos.

A continuación, se detallan en la siguiente tabla (Tabla 8.1.) algunos de los modelos teóricos que podrían ser de utilidad para ampliar en número a los que ya se han implementado:

Cinéticas de adsorción	Isothermas de adsorción
Modelo de difusión intraparticular	Isoterma de Temkin
Modelo de pseudo primer orden	Isothermas B.E.T.

Tabla. 8.1. Modelos cinéticos y de isothermas a implementar en futuras actualizaciones.

Por otro lado, sería necesario añadir una opción que permita añadir otras series de valores con tal de realizar un estudio comparativo entre ensayos, pudiendo visualizar los distintos casos simultáneamente y en el mismo gráfico.

Además, también cabría la posibilidad de incluir nuevas pestañas en el programa con distintos tipos de estudios de la adsorción, como los referentes a los diagramas de pH, utilizados para estudiar el efecto del pH sobre la cantidad adsorbida de iones. En este caso, se mide la concentración de soluto en el adsorbato para diferentes valores de pH, estableciendo un mismo tiempo de contacto para cada uno de los ensayos.

8.3. Exportación de los resultados a *Excel*

Otra herramienta que podría considerarse muy productiva, por no decir necesaria, y que no se ha podido incluir en esta versión del programa debido al tiempo y la complejidad que conlleva, sería la posibilidad de exportar los resultados numéricos obtenidos a una hoja (o plantilla) de *Excel*.

Teniendo en cuenta que en esta primera versión los resultados tan sólo pueden ser visualizados de manera gráfica, sería muy útil ofrecer al usuario la opción de almacenarlos y poder disponer de ellos para un análisis más profundo.

Mediante un simple botón llamado por ejemplo “*Exportar resultados*”, podría crearse una función que al pulsar sobre él, se encargara de recopilar todos los valores obtenidos a través de los cálculos realizados, y los guardara en un documento de *Excel*, rellenando las celdas correspondientes en una plantilla previamente diseñada.

8.4. Mejora de las opciones de edición de los gráficos

La edición de los gráficos representados mediante la aplicación es bastante útil pero a su vez limitada en la versión desarrollada. En ella, se permite navegar a través de los gráficos pudiendo realizar varias acciones sobre los mismos. Entre ellos destacan: moverse a través del gráfico, redimensionar su tamaño, ampliar o reducir una zona determinada y conocer las coordenadas de un punto en concreto del mismo. Además, también es posible modificar el color y el estilo de los marcadores, así como los títulos del gráfico y los de los ejes.

Pero es cierto que se encuentran a faltar varias herramientas de edición con tal de poder sacar un mayor provecho de los gráficos que se obtienen. Algunas de ellas, son las que se plantean a continuación:

- Seleccionar si se quieren incluir o no en el gráfico, de forma independiente, los siguientes elementos: ecuaciones del modelo, constantes obtenidas y sus factores de correlación, y la leyenda.
- Variar la posición relativa de la leyenda, las ecuaciones y los parámetros obtenidos respecto al gráfico.
- Posibilidad de mostrar/ocultar otras series de datos representadas en el mismo gráfico, a modo de comparación.

9. Comprobación de funcionamiento del programa

Con el programa ya terminado y listo para ser distribuido, lo último que quedaba por hacer era una comprobación final para corroborar la fiabilidad de resultados ofrecidos en la aplicación, realizando un estudio con datos provenientes de artículos facilitados por el director.

Dicha comprobación consistiría en utilizar esos mismos datos de estudio en el programa, y comparar los resultados obtenidos en varios casos con los publicados en el artículo en cuestión. Para ello, se dispuso del documento de *Excel* original en el que fueron calculados.

Antes de empezar, cabe comentar que en el artículo de referencia se realizó un estudio de adsorción sobre el arsénico As(III). El estudio se centra principalmente en la determinación de los mecanismos cinéticos y las isotermas de adsorción del arsénico sobre cada uno de los siguientes minerales: goetita, hematita y magnetita. Los datos del ensayo son los contenidos en el fichero *arsenico3.xls*.

9.1. Estudio de la cinética de adsorción

9.1.1. Caso de estudio con Excel

En este documento, el principal modelo cinético empleado es el de pseudo-segundo orden, por lo que se usará como ejemplo para la comprobación. Además, se tomará como referencia el caso estudiado para la goetita con tal de presentar imágenes y capturas de pantalla que justifiquen la comparación.

Los datos de partida para realizar los cálculos, son los que se muestran dentro del documento, señalados en la Fig 9.1.

En el documento de Excel, primeramente se realiza la transformación de datos para obtener el parámetro q_t , necesario para el cálculo del modelo. A continuación, en la Fig. 9.2., se muestra la recta obtenida a través del ajuste lineal del modelo, junto con los correspondientes valores de las constantes q_e y k del modelo cinético de pseudo segundo orden.

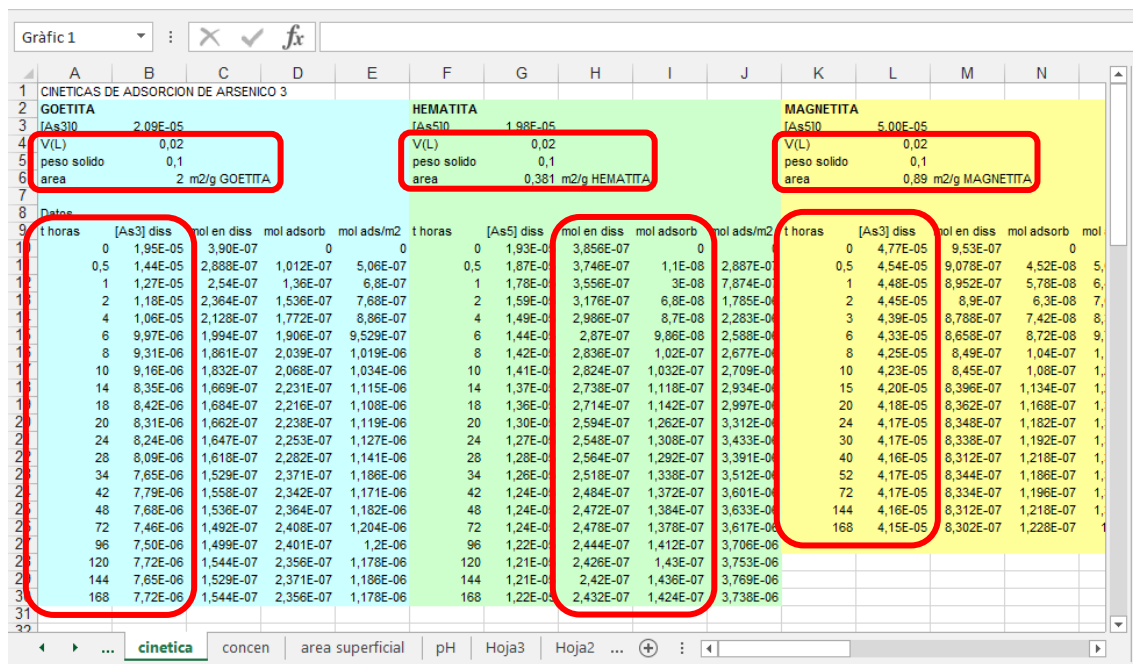


Fig. 9.1. Datos de partida para los cálculos cinéticos en el fichero de Excel.

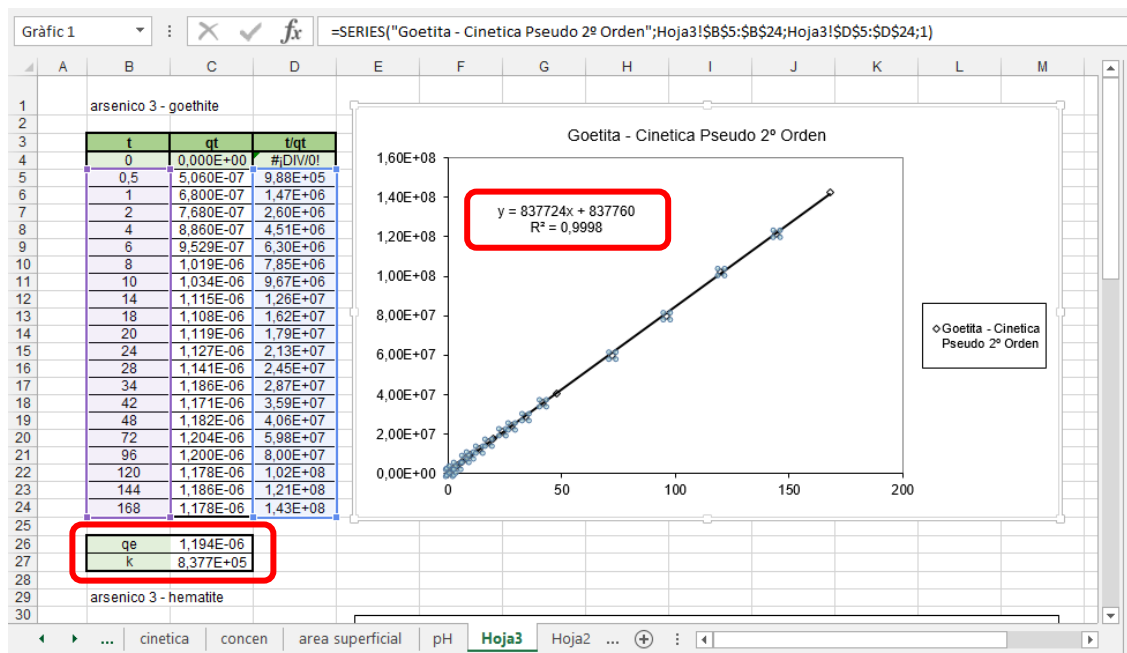


Fig. 9.2. Parámetros del modelo cinético de pseudo segundo orden obtenidos tras la representación lineal de los datos para el caso de la goetita, en Excel.

Se observa que la recta resultante posee un factor de correlación muy cercano a 1 ($R^2=0,9998$), por lo que se puede deducir que los datos de estudio se ajustan en gran medida a la ecuación del modelo cinético de pseudo segundo orden.

A partir de las constantes del modelo obtenidas se procede con el cálculo del modelo teórico, comparando la curva obtenida con los datos experimentales, tal y como se observa en la Fig 9.3.:

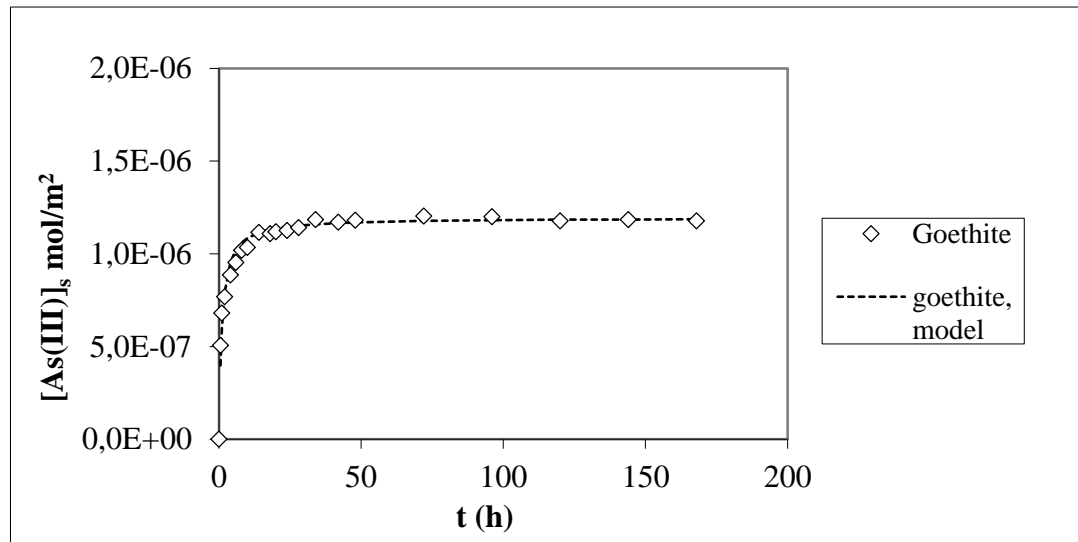


Fig. 9.3. Gráfico comparativo entre datos experimentales y el modelo cinético de pseudo segundo orden obtenido en *Excel*.

De este gráfico se puede deducir que los valores obtenidos en el ensayo de adsorción del arsénico (III) sobre la goetita se comportan siguiendo la ecuación cinética teórica del modelo pseudo-segundo orden. También es posible determinar el valor de concentración de As(III) en el equilibrio, el cual es de aproximadamente $1,2 \cdot 10^{-6}$ mol/m².

9.1.2. Caso de estudio mediante la aplicación creada

Una vez expuestos los resultados obtenidos mediante el uso de Excel, se procedió a la obtención de los resultados mediante el uso del programa creado durante el transcurso de este proyecto.

En primer lugar, se introdujeron en el programa los mismos datos provenientes del documento de referencia para cada mineral, de forma separada. Se seleccionó el modelo cinético de pseudo-segundo orden y se procedió a generar los gráficos con los resultados.

Fig. 9.4. Datos introducidos en la aplicación para el caso de estudio de la goetita.

A continuación, se muestra el gráfico obtenido en la aplicación en la Fig. 9.5., dónde pueden observarse de manera conjunta los diagramas referentes a la linealización y al cálculo del modelo teórico para el caso de la goetita:

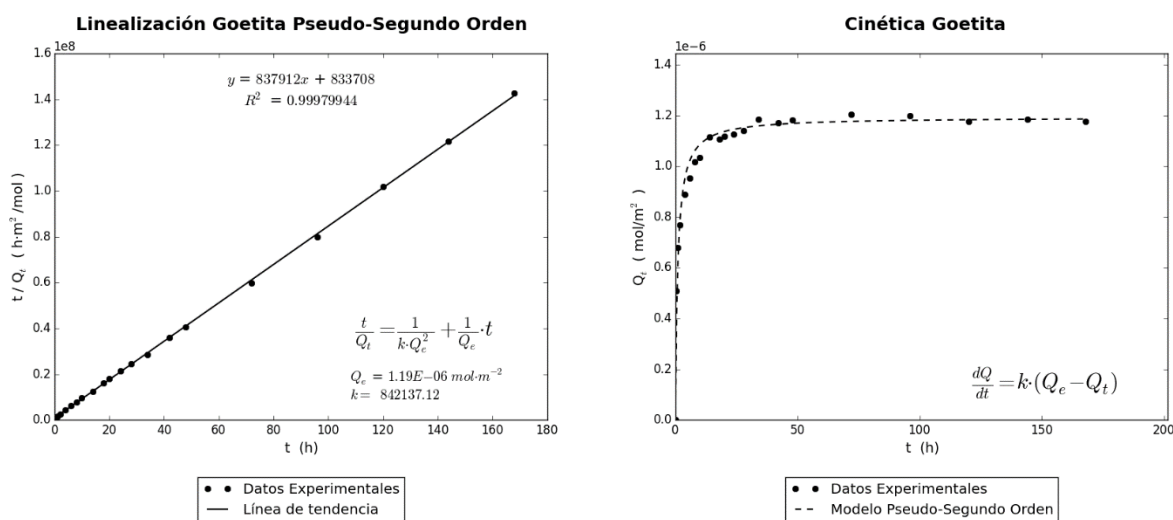


Fig. 9.5. Gráficos cinéticos obtenidos a través de la aplicación para el modelo de pseudo segundo orden.

Del mismo modo, se seleccionó también el cálculo del modelo cinético de Elovich, a fin de comparar los resultados anteriores. En la Fig 9.6. se observa el gráfico obtenido según el modelo de Elovich:

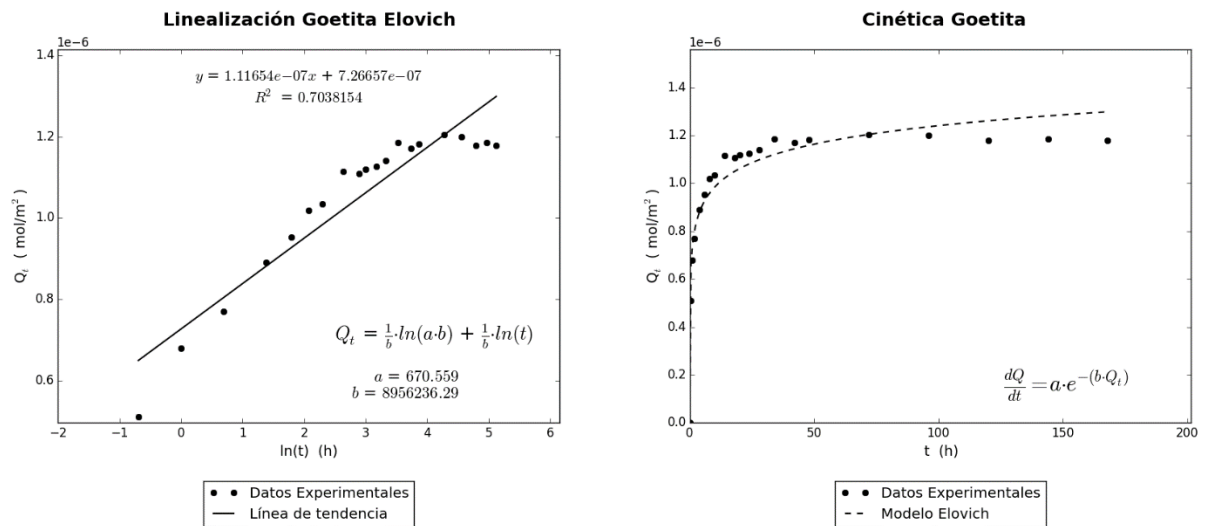


Fig. 9.6. Gráficos cinéticos obtenidos a través de la aplicación para el modelo de Elovich.

Finalmente, se compararon los resultados obtenidos en ambos modelos. Para ello, se seleccionó la opción de “*Comparación cinéticas*” de la aplicación, con tal de decidir a qué modelo se ajustan más los datos experimentales. En la siguiente imagen obtenida mediante el programa (Fig. 9.7.) se comparan las curvas teóricas de cada modelo:

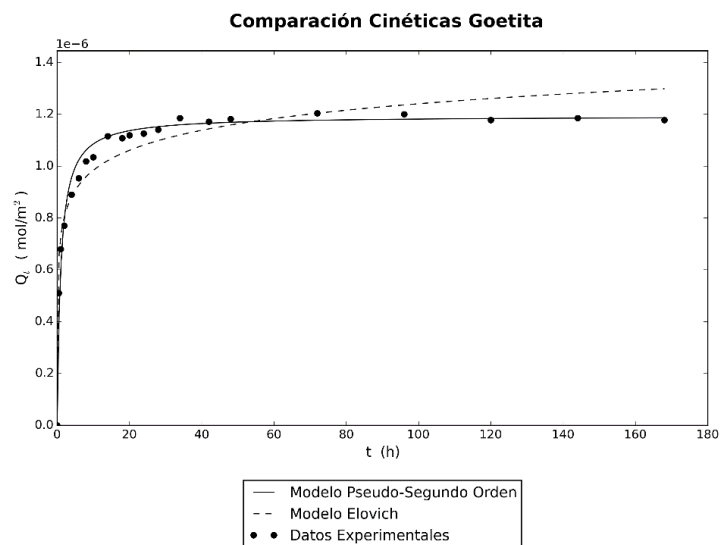


Fig. 9.7. Gráfico comparativo de modelos cinéticos obtenidos a través de la aplicación.

Goetita	Modelo Pseudo segundo orden	Modelo Elovich
Ecuación de la recta	$y = 837912x + 833708$	$y = 1,117 \cdot 10^{-6}x + 7,266 \cdot 10^{-7}$
Factor de correlación	$R^2 = 0,9998$	$R^2 = 0,7038$

Tabla. 9.1. Tabla comparativa de los modelos cinéticos para el caso de la goetita.

A través del gráfico comparativo de modelos, y de los factores de correlación de Pearson (R^2) de cada modelo, puede afirmarse que los datos experimentales de ensayo describen un sistema químico que se ajusta mejor a los mecanismos cinéticos descritos por el modelo de pseudo-segundo orden, en detrimento a los descritos por el modelo de Elovich.

9.1.3. Comparación de métodos de cálculo

En este apartado se pretende comprobar la fiabilidad de cálculo del programa comparándolo con los resultados obtenidos y corroborados mediante el documento de Excel. Esta comparación se realiza procediendo del mismo modo para los tres minerales de estudio, de manera que así se dispone de más muestras para comprobar la validez del sistema de cálculo de la aplicación. En las siguientes tablas para cada mineral, se observan los valores significativos obtenidos en cada caso:

Goetita	Documento Excel	Programa AdsorptionGraph
Ecuación de la recta	$y = 837724x + 833760$	$y = 837912x + 833708$
Factor de correlación	$R^2 = 0,9998$	$R^2 = 0,99979$
q_e	$1,194 \cdot 10^{-6}$	$1,193 \cdot 10^{-6}$
k	837688,00	842137,12

Tabla. 9.2. Tabla comparativa entre métodos de cálculo para el caso de la goetita.

Hematita	Documento Excel	Programa AdsorptionGraph
Ecuación de la recta	$y = 260059x + 945543$	$y = 259830x + 920333$
Factor de correlación	$R^2 = 0,99963$	$R^2 = 0,99938$
q_e	$3,845 \cdot 10^{-6}$	$3,848 \cdot 10^{-6}$
k	71525,76	73355,69

Tabla. 9.3. Tabla comparativa entre métodos de cálculo para el caso de la hematita.

Magnetita	Documento Excel	Programa AdsorptionGraph
Ecuación de la recta	$y = 720902x + 1072075$	$y = 716502x + 1056020$
Factor de correlación	$R^2 = 0,99987$	$R^2 = 0,99977$
q_e	$1,387 \cdot 10^{-6}$	$1,396 \cdot 10^{-6}$
k	484761,65	486141,63

Tabla. 9.4. Tabla comparativa entre métodos de cálculo para el caso de la magnetita.

Los valores obtenidos en ambos métodos de cálculo son prácticamente iguales, como se puede observar en las tablas adjuntas, por lo que se concluye que los resultados que ofrece el programa en el cálculo de las cinéticas de adsorción son realmente fiables. Las pequeñas diferencias que se observan entre ellos pueden ser debidas al empleo de distintos herramientas de cálculo estadístico, pero se considera que no alteran el resultado final.

9.2. Estudio de isotermas de adsorción

9.2.1. Caso de estudio con *Excel*

Para el caso de estudio de las isotermas de adsorción, se escogió como referencia el modelo de Langmuir en la comparación.

La distribución del documento de Excel es el que se observa en la Fig 9.8., dónde se señalan los datos de entrada de concentración de cada mineral utilizados para los cálculos:

VARIACIÓN CON CONCENTRACIÓN INICIAL DE ARSENICO 3												LANGMUIR
GOETITA				HEMATITA				MAGNETITA				
V(L) 0,02				V(L) 0,02				V(L) 0,02				
peso solido 0,1				peso solido 0,1				peso solido 0,1				
area 2 m2/g GOETITA				area 0,381 m2/g HEMATITA				area 0,89 m2/g MAGNETITA				
[As] inicial	[As5] diss	mol adsorb	mol ads/m2	[As] inicial	[As5] diss	mol adsorb	mol ads/m2	[As] inicial	[As5] diss	mol en diss	mol adsorb/m2	
8,48E-06	5,23E-07	1,59E-07	7,96E-07	8,48E-06	4,23E-06	8,50E-08	2,23097E-07	3,00E-06	1,18E-06	3,64E-08	4,09E-07	
2,75E-05	1,19E-05	3,12E-07	1,56E-06	2,75E-05	1,72E-05	2,06E-07	5,40682E-07	2,43E-05	2,04E-05	7,80E-08	8,76E-07	
4,69E-05	2,82E-05	3,74E-07	1,87E-06	4,69E-05	3,60E-05	2,18E-07	5,72178E-07	4,46E-05	3,88E-05	1,16E-07	1,30E-06	
6,60E-05	4,30E-05	4,60E-07	2,30E-06	6,60E-05	5,36E-05	2,48E-07	6,50919E-07	9,33E-05	8,46E-05	1,74E-07	1,96E-06	
9,47E-05	7,08E-05	4,78E-07	2,39E-06	9,47E-05	8,10E-05	2,74E-07	7,1916E-07	2,91E-04	2,79E-04	2,40E-07	2,70E-06	
2,91E-04	2,65E-04	5,20E-07	2,60E-06	2,92E-04	2,75E-04	3,40E-07	8,92388E-07	4,90E-04	4,77E-04	2,60E-07	2,92E-06	
4,70E-04	4,46E-04	4,80E-07	2,40E-06	4,88E-04	4,71E-04	3,40E-07	8,92388E-07	6,83E-04	6,70E-04	2,60E-07	2,92E-06	
[As5]diss	[As5]diss/ qe			[As5]diss	[As5]diss/ qe			[As5]diss	[As5]diss/ qe			
5,23E-07	6,57E-01			4,23E-06	1,90E+00			1,18E-06	2,89E+00			
1,19E-05	7,63E+00			1,72E-05	3,18E+00			2,04E-05	2,33E+01			
2,82E-05	1,51E+01			3,60E-05	6,29E+00			3,88E-05	2,98E+01			
4,30E-05	1,87E+01			5,36E-05	8,23E+00			8,46E-05	4,33E+01			
7,08E-05	2,96E+01			8,10E-05	1,13E+01			2,79E-04	1,03E+02			
2,65E-04	1,02E+02			2,75E-04	3,08E+01			4,77E-04	1,63E+02			
4,46E-04	1,86E+02			4,71E-04	5,28E+01			6,70E-04	2,29E+02			

Fig. 9.8. Datos de partida para el cálculo de las isotermas de adsorción, en Excel.

Una vez calculada a ecuación lineal del modelo de Langmuir, se obtienen las rectas resultantes junto con su ecuación para cada mineral. A través del factor de correlación R², se puede deducir que existe un buen ajuste entre los distintos datos con este modelo:

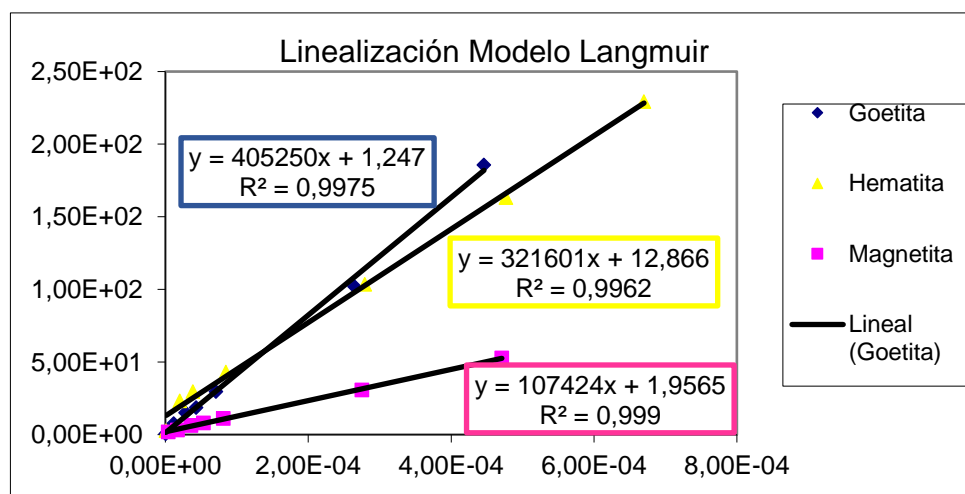


Fig. 9.9. Gráfico del ajuste lineal de los datos para el caso de la isoterma de Langmuir, en Excel.

Finalmente, se elabora el gráfico representado en la Fig. 9.10., correspondiente a la comparación del modelo teórico de Langmuir frente a los datos experimentales:

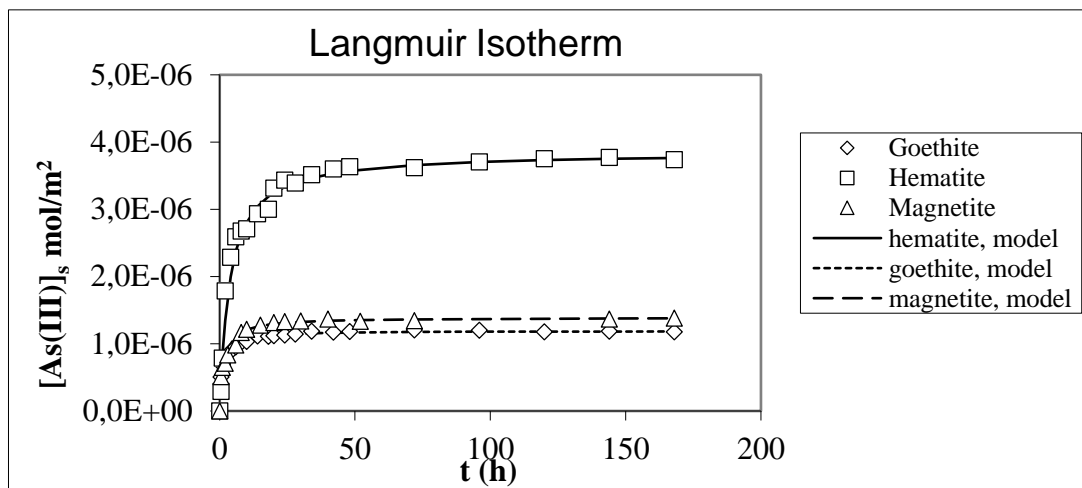


Fig. 9.10. Gráfico comparativo entre datos experimentales y el modelo cinético de pseudo segundo orden obtenido en *Excel*.

9.2.2. Caso de estudio mediante la aplicación creada

Procediendo de igual manera que para las cinéticas de adsorción, se utilizará el programa para el cálculo y comparación de las isotermas de Langmuir y Freundlich.

The screenshot shows the "AdsorptionGraph v1.0" application window. The "Cinética" tab is selected. The interface is divided into several sections:

- Datos a introducir:**
 - Nombre elemento: GOETITA
 - Volumen disolución [L]: 0.02
 - Peso del sólido [g]: 0.1
 - Conoce el valor del área superficial? Sí No
 - Área superficial [m²/g]: 2
- Rellene los valores requeridos en la tabla:**

C inicial [mol/L]	C t [mol/L]
8,48E-06	5,23E-07
2,75E-05	1,19E-05
4,69E-05	2,82E-05
6,60E-05	4,30E-05
9,47E-05	7,08E-05
2,91E-04	2,65E-04
4,70E-04	4,46E-04
- Especifique el modelo que desea representar:**
 - Modelo de Langmuir
 - Modelo de Freundlich
 - Comparación Modelos
- Valide los datos introducidos:**
 - 5. Validar Datos (button)
 - 6. Calcular (button)
 - 7. Guardar Gráfico (button)
 - Mostrar Gráfico (button)

At the bottom, there are buttons for "Reiniciar" and "Salir". A status message reads: "Los datos introducidos son válidos".

Fig. 9.11. Datos introducidos en la aplicación para el caso de estudio de la goetita.

Tras introducir los datos correspondientes para cada mineral en la aplicación, y seleccionando los diferentes modelos, se generaron los gráficos correspondientes a cada modelo de isoterma. En las figuras 9.12. y 9.13 se muestran los gráficos obtenidos para la goetita:

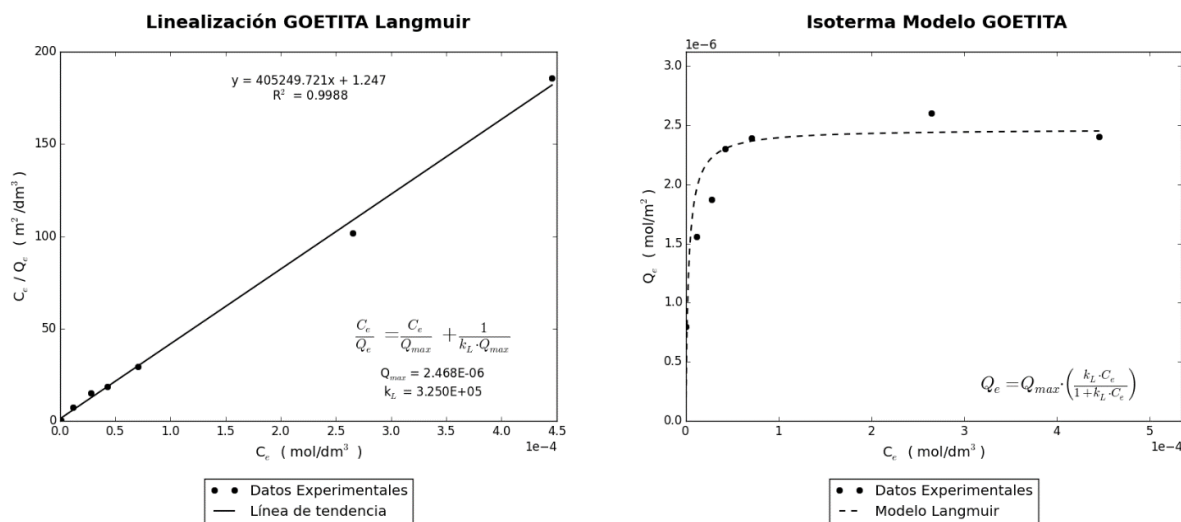


Fig. 9.12. Gráficos obtenidos a través de la aplicación para la isoterma de Langmuir.

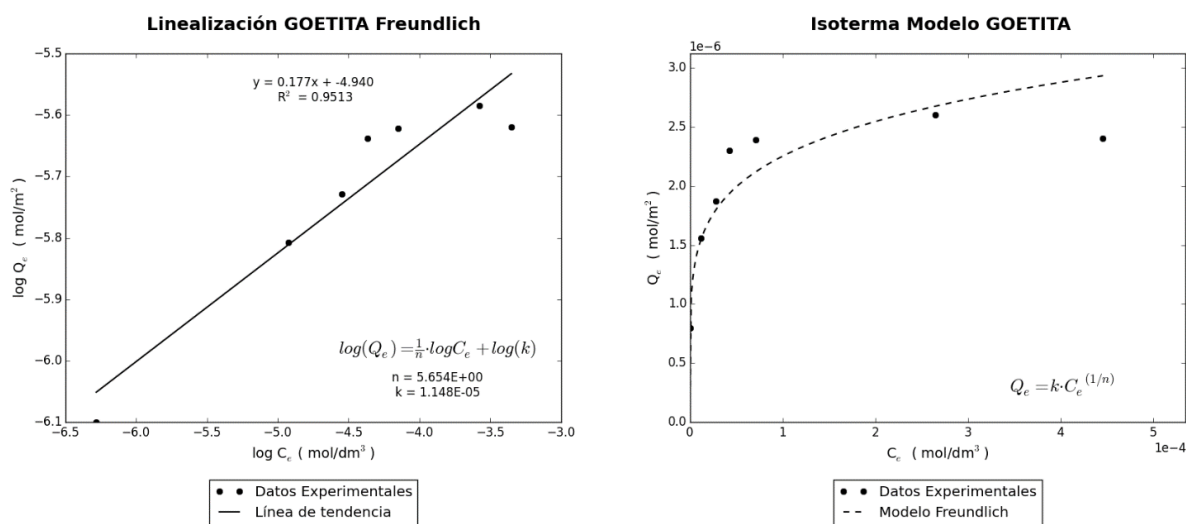


Fig. 9.12. Gráficos obtenidos a través de la aplicación para la isoterma de Freundlich.

A partir de los coeficientes de correlación obtenidos para cada modelo, y tras generar el gráfico comparativo seleccionando la opción de “*Comparación Modelos*” en el programa, es posible determinar a qué modelo se asemejan más los datos experimentales:

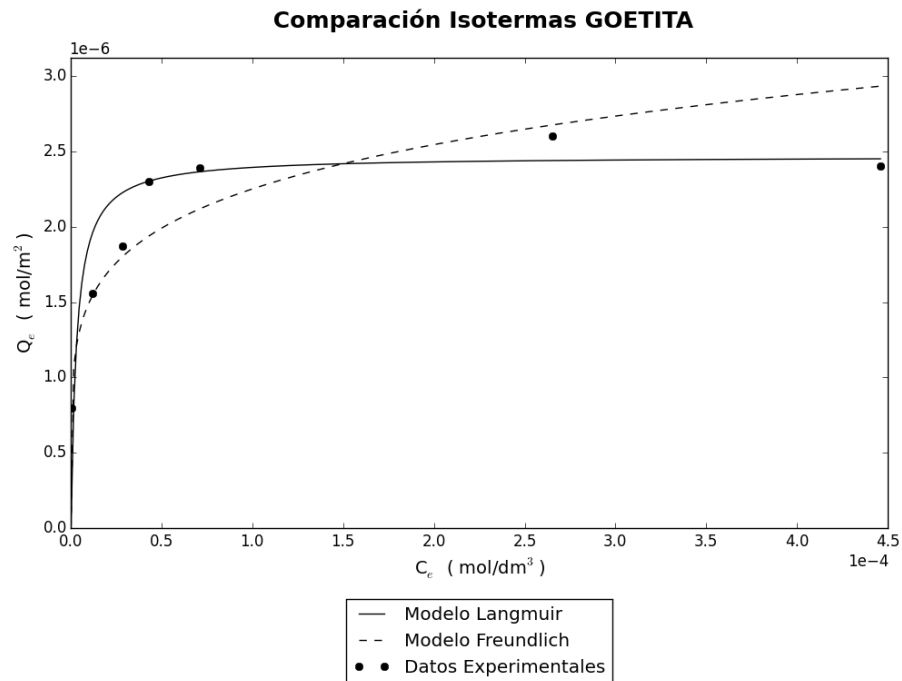


Fig. 9.14. Gráfico comparativo de modelos de isotermas obtenidos a través de la aplicación.

Goetita	Modelo Langmuir	Modelo Freundlich
Ecuación de la recta	$y = 405249,7x + 1,247$	$y = 0,177x - 4,940$
Factor de correlación	$R^2 = 0,9988$	$R^2 = 0,9513$

Tabla. 9.5. Tabla comparativa de los modelos de isoterma para el caso de la goetita.

Los resultados indican un mejor ajuste de los datos de estudio para la isoterma de Langmuir, puesto que ésta posee un valor de R^2 superior al obtenido para el modelo de Freundlich. También es posible llegar a esta conclusión observando el gráfico comparativo (Fig. 9.14.).

9.2.3. Comparación de los métodos de cálculo

Por último, se realizó una comparación de los resultados obtenidos con las dos herramientas de cálculo utilizadas. Para ello, se empleó como ejemplo el modelo de isoterma de Langmuir, puesto que es el único modelo común calculado en ambos casos.

Las siguientes tablas corroboran la gran fiabilidad de cálculo que posee el programa, debido a que como se puede observar en ellas, los resultados obtenidos tanto en Excel como en la aplicación son idénticos:

Goetita	Documento Excel	Programa AdsorptionGraph
Ecuación de la recta	$y = 405250x + 1,247$	$y = 405249,7x + 1,247$
Factor de correlación	$R^2 = 0,9975$	$R^2 = 0,9988$
q_{\max}	$2,468 \cdot 10^{-6}$	$2,468 \cdot 10^{-6}$
k_L	$3,250 \cdot 10^5$	$3,250 \cdot 10^5$

Tabla. 9.6. Tabla comparativa entre métodos de cálculo para el caso de la goetita.

Hematita	Documento Excel	Programa AdsorptionGraph
Ecuación de la recta	$y = 107424x + 1,956$	$y = 107424,03x + 1,957$
Factor de correlación	$R^2 = 0,9990$	$R^2 = 0,9995$
q_{\max}	$9,309 \cdot 10^{-6}$	$9,309 \cdot 10^{-6}$
k_L	54906	54910

Tabla. 9.7. Tabla comparativa entre métodos de cálculo para el caso de la hematita.

Magnetita	Documento Excel	Programa AdsorptionGraph
Ecuación de la recta	$y = 321601x + 12,866$	$y = 321601,339x + 12,866$
Factor de correlación	$R^2 = 0,9962$	$R^2 = 0,9981$
q_{\max}	$3,109 \cdot 10^{-6}$	$3,109 \cdot 10^{-6}$
k_L	$2,4996 \cdot 10^4$	$2,5 \cdot 10^4$

Tabla. 9.8. Tabla comparativa entre métodos de cálculo para el caso de la magnetita.

10. Costes

Los costes asociados a la elaboración de este proyecto se descomponen principalmente en dos partes: los costes de la mano de obra, y los costes asociados a su realización. Debido a que el trabajo se ha realizado íntegramente con ordenador, tan sólo se contemplarán como costes materiales los derivados de la impresión del mismo, y de los CDs grabados necesarios para su presentación.

En primer lugar, en referencia a los **costes de la mano de obra**, estos vienen dados por las horas trabajadas por la persona que ha realizado el proyecto. Teniendo en cuenta que para su elaboración se realizaron una media de unas 8 horas semanales durante aproximadamente un periodo de 16 meses hasta el momento de la entrega de la memoria escrita. Todo ello suma un total de 512 horas dedicadas al proyecto, sin tener en cuenta muchas otras enfocadas al aprendizaje necesario en programación con *Python* por deseo propio del autor. Considerando que un ingeniero cobrase alrededor de 30€/h por la realización de un proyecto, el coste de la mano de obra por todo el trabajo ascendería a **15.360€**.

En segundo lugar, los **costes asociados a la realización del trabajo** engloban los gastos directos derivados de su confección. Al tratarse de un programa informático, su valor es mucho menos significativo que los asociados a la mano de obra. Además estos costes son bastante más reducidos que en otros tipos de proyectos, puesto que no se han utilizado otros medios a parte de un ordenador portátil. Estos costes engloban lo siguiente:

- Costes eléctricos: Los costes eléctricos han sido calculados a partir del consumo del ordenador portátil (120W) y del coste del *kWh*, teniendo en cuenta el total de horas trabajadas (512h). En consecuencia, el consumo eléctrico total del proyecto se obtiene de la siguiente manera:

$$0,12 \text{ kW} \cdot 512 \text{ h} = 61,44 \text{ kWh}$$

Suponiendo que el precio aproximado del *kWh* es de 0,15€/kWh, el coste total eléctrico por la elaboración del proyecto es de unos **9,22€**.

- Costes de amortización del ordenador: Partiendo de que el tiempo de vida medio de un ordenador portátil es de 6 años, y que el precio de adquisición del equipo en cuestión utilizado ronda los 800€, es posible calcular los gastos de amortización derivados por el uso del mismo durante la realización del trabajo:

$$\frac{512 \text{ h trabajadas}}{6 \text{ años} \cdot 8760 \frac{\text{h}}{\text{año}}} \cdot 800\text{€} = 7,80\text{€}$$

- Costes materiales de impresión y encuadernación: Finalmente, hay que añadir el coste de impresión de la memoria escrita, el papel usado, la encuadernación del mismo y los CDs utilizados. Teniendo en cuenta que para la presentación se requiere un total de 3 copias completas del trabajo, se estima que su coste sería de unos **70€**.

Por lo tanto, se estima que los costes totales del proyecto ascienden a **15.447,02€**.

11. Impacto ambiental

En este apartado se analizan las consecuencias ambientales que conlleva la realización de este proyecto. Cabe mencionar que el impacto ambiental generado durante su desarrollo es prácticamente menospreciable comparado con trabajos de ámbito más industrial. No obstante, es necesario hacer hincapié en las principales fuentes que causan el impacto.

En primer lugar, referentes al proceso de desarrollo del programa y la redacción de la memoria, se tienen que considerar las emisiones de CO₂ derivadas del consumo eléctrico, ya que contribuyen al efecto invernadero. Según los datos de la Unión Europea, que cifran en 0,39kg de emisiones de CO₂ para la producción de 1kWh, y teniendo en cuenta el consumo eléctrico calculado anteriormente en el apartado de costes, se estima que las emisiones derivadas de este proyecto representan 23,65kg de CO₂.

Por otro lado, para la impresión y presentación de la memoria podría considerarse la cantidad de papel utilizada, que representa una de las principales causas de deforestación.

Finalmente, es necesario mencionar que la aplicación desarrollada pretende contribuir a la mejora de los estudios de investigación sobre la remediación de especies contaminantes, como el arsénico. Como consecuencia de estos estudios, y aunque quede fuera del alcance de este proyecto, hay que tener en cuenta el impacto ambiental derivado del uso futuro del programa. Puesto que el programa se basa en el análisis de datos procedentes de estudios experimentales, se tendrán que considerar los componentes utilizados y los residuos generados durante la realización de los ensayos.

12. Planificación

La planificación llevada a cabo para la realización del trabajo es la que se observa en el diagrama de Gantt, mostrado a continuación (Figura X.X):

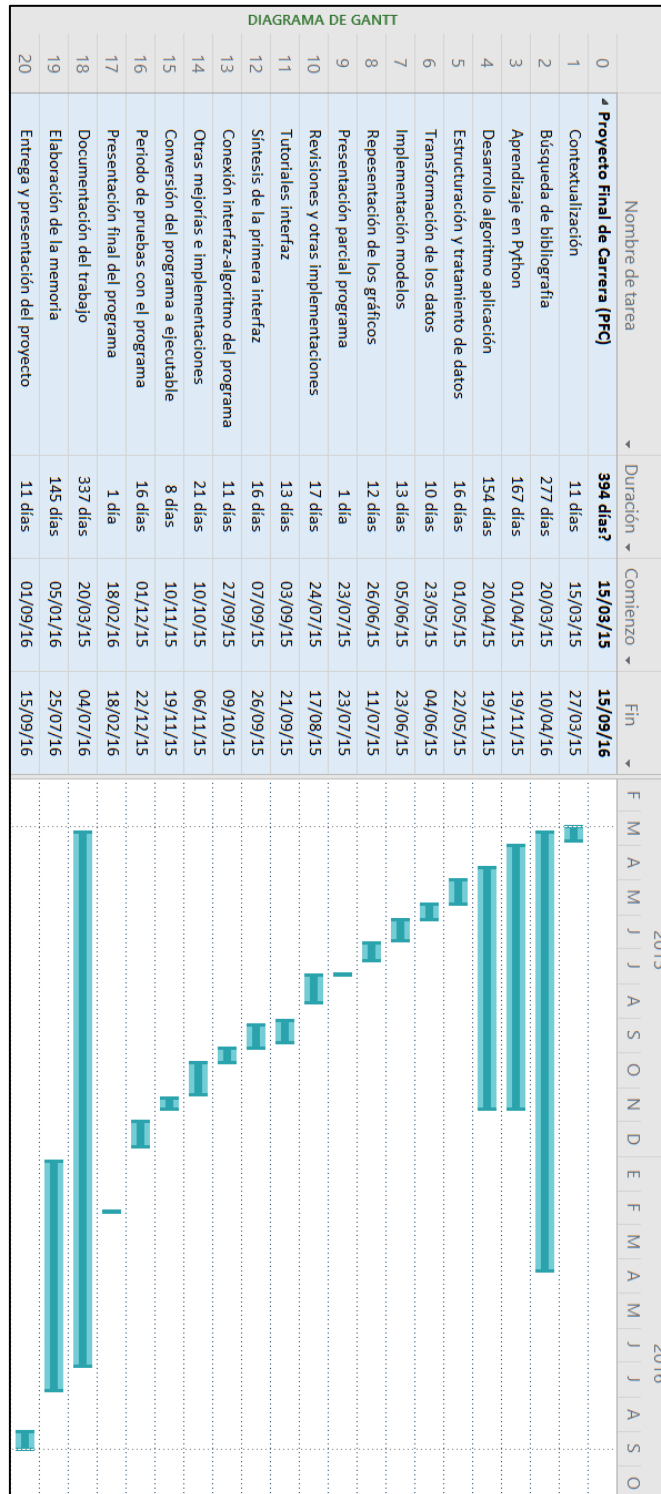


Fig. 12.1. Diagrama de Gantt referente a la planificación del proyecto.

Conclusiones

Este trabajo, considerado como la primera fase de un proyecto mayor, satisface la gran mayoría de los objetivos propuestos.

El primero a destacar, y quizás el más importante, es que se ha conseguido crear una herramienta informática compacta y funcional que resuelve el problema planteado. La aplicación en cuestión, permite la caracterización de los mecanismos de adsorción presentes entre sustancias durante este tipo de ensayos experimentales.

Mediante el programa, es posible calcular algunos de los modelos existentes que describen este fenómeno de manera muy rápida. Los datos son procesados de manera automática y los resultados que se obtienen son presentados de forma gráfica para ofrecer una mejor interpretación. Todo ello, con el único propósito de ayudar a determinar los parámetros y factores clave en estudios de adsorción, como son el orden cinético de reacción y las condiciones de equilibrio.

En segundo lugar, cabe mencionar que la aplicación final está pensada para fomentar su uso entre la gente que entienda sobre estudios de adsorción, por lo que posee una interfaz visual ordenada y muy intuitiva, que facilita la interacción con el programa para todo aquél que no posea conocimientos avanzados en informática. Además, en el caso de que éste presente dificultades durante su utilización, el programa incorpora numerosas ayudas, y también se dispone de manuales con las instrucciones de uso.

Si bien es cierto que las funcionalidades implementadas están focalizadas a analizar unos tipos de ensayo de adsorción muy concretos, y que los datos requeridos para ser procesados deben ser introducidos de una manera específica, las opciones que brinda el programa responden de manera eficaz y ofrecen resultados precisos de forma ágil, tal y como se ha demostrado durante el trabajo. No obstante, la fiabilidad de éstos está evidentemente condicionada por los datos de los que se parte.

Por último, observando el proyecto en perspectiva y con los conocimientos actuales adquiridos a medida que se avanzaba en su desarrollo, seguramente se habría diseñado el programa de manera algo diferente, dotándolo de una estructura de código más robusta, pulida y versátil. Pero aun así, considerando la elevada dificultad que suponía este reto desde su inicio, se puede concluir que el programa cumple con creces las expectativas establecidas inicialmente: establece una base sólida de partida, posee una gran capacidad de mejora y ofrece bastantes facilidades para que otros puedan continuar con su desarrollo.

Agradecimientos

Me gustaría que estas líneas sirvieran para expresar mi más sincero agradecimiento a todas esas personas que con su ayuda han colaborado en la realización del presente trabajo, en especial a los directores de este proyecto, Francisco Javier Giménez y Lluís Solano, por el interés mostrado y por la orientación, el seguimiento y la supervisión continua del trabajo.

Este proyecto supone un punto y final a un largo camino recorrido tanto personalmente como profesionalmente. Sin lugar a dudas, el papel de mis compañeros de universidad y de mis amigos de toda la vida ha sido fundamental para llegar hasta aquí.

Tampoco puedo olvidarme de mi familia. Es difícil describir con palabras el agradecimiento que siento hacia ellos por su apoyo y motivación constantes. Gracias a todos.

Bibliografía

- [1] Alonso Quintans D., *“Reutilización de raspo procedente de la industria vinícola para la extracción de metales en efluentes líquidos”*, Universidad Politécnica de Catalunya, 20-34, 2004
- [2] Ayora Salvo V., *“Estudio experimental del sistema Hematita/Fosfato para su aplicación a la fertilización de suelos agrícolas”*, Universitat Politècnica de Catalunya, 36-45, 2013
- [3] Y.-S. Ho, *“Review of second-order models for adsorption systems”*, Journal of Hazardous Materials, vol. 136, pp. 681-689, 2006.
- [4] Y.-S. Ho, *“Citation review of Lagergren kinetic rate equation on adsorption reactions”*, Scientometrics, vol.59, nº1, pp. 171-177, 2004.
- [5] I. Matus, L. Paniagua y M. Benavente, *“Estudio de la cinética de adsorción de Cu y Hg a partir de soluciones binarias usando quitosano”*, Nexo: Revista científica, vol. 24, No. 1, pp. 20-32, 2011
- [6] I. W. Tan, a L. Ahmad, and B. H. Hameed, *“Adsorption isotherms, kinetics, thermodynamics and desorption studies of 2, 4, 6-trichlorophenol on oil palm empty fruit bunch-based activated carbon”*, Journal of Hazardous Materials, vol. 164, n.º 2-3, pp. 473-82, 2009.
- [7] M. Arami, N. Y. Limaee, N. M. Mahmoodi, y N. S. Tabrizi, *“Equilibrium and kinetics studies for the adsorption of direct and acid dyes from aqueous solution by soy meal hull”*, Journal of Hazardous Materials, vol. 135, n.º 1-3, pp. 171-9, 2006.
- [8] Sureda R., Martinez-Lladó X., Rovira M., de Pablo J., Casas I., Giménez J., *“Sorption of strontium on uranyl peroxide: Implications for a high-level nuclear waste repository”*, Journal of Hazardous Materials, vol. 181, pp. 881-885, 2010.
- [9] Y.-S. Ho, W.-T. Chiu, and C.-C. Wang, *“Regression analysis for the sorption isotherms of basic dyes on sugarcane dust”*, Bioresource technology, vol. 96, n.º 11, pp. 1285-91, 2005.

- [10] Rovira, M., Giménez J., Martínez-Lladó X., de Pablo J., Martí V., Duro L., “*Sorption of selenium(IV) and selenium(VI) onto natural iron oxides: goethite and hematite*”. *Journal of Hazardous Materials*, vol. 150, pp. 279-284, 2008.
- [11] Martínez M., Giménez J., de Pablo J., Rovira M., Duro L., “*Sorption of selenium(IV) and selenium(VI) onto magnetite*”, *Applied Surface Science*, vol. 252, pp 3767-3773, 2006
- [12] G. Limousin, J.-P. Gaudet, L. Charlet, S. Szenknect, V. Barthès, and M. Krimissa, “*Sorption isotherms: A review on physical bases, modeling and measurement*”, *Applied Geochemistry*, vol. 22, n.º 2, pp. 249-275, 2007.
- [13] Gonzalez Duque R., “*Python para todos*” (Libro online)
[<https://launchpadlibrarian.net/18980633/Python%20para%20todos.pdf>]
- [14] “*Tutorial de Python: Programación orientada a objetos*”
[<https://frontendlabs.io/1305--tutorial-basico-de-python-parte-iv-programacion-orientada-a-objetos>]
- [15] “*Codejobs: Comunidad tecnológica relacionada con el aprendizaje en Python*”
[<https://www.codejobs.biz/es/blog/2013/03/02/que-es-python>]
- [16] “*Curso online: Introducción a Python para científicos e ingenieros*”
[<http://cacheme.org/curso-online-python-cientifico-ingenieros/>]
- [17] Jesús Conde, “*Curso Aplicaciones GUI Python con PyQt*”
[<https://www.youtube.com/playlist?list=PLEtcGQaT56cj81xiNCrJnoAjc66uZVXDa>]
- [18] “*Mecanismo de conexiones de señales y slots*”
[<http://aicastell.blogspot.com.es/2012/01/el-mecanismo-de-conexiones-senales-y.html>]

Anexo A

A.1. Diagrama de funcionamiento del programa

El diagrama de funcionamiento básico del programa es el que se observa en la Figura A.1. En el manual de usuario que encontrará más adelante, se explica detalladamente cada bloque.

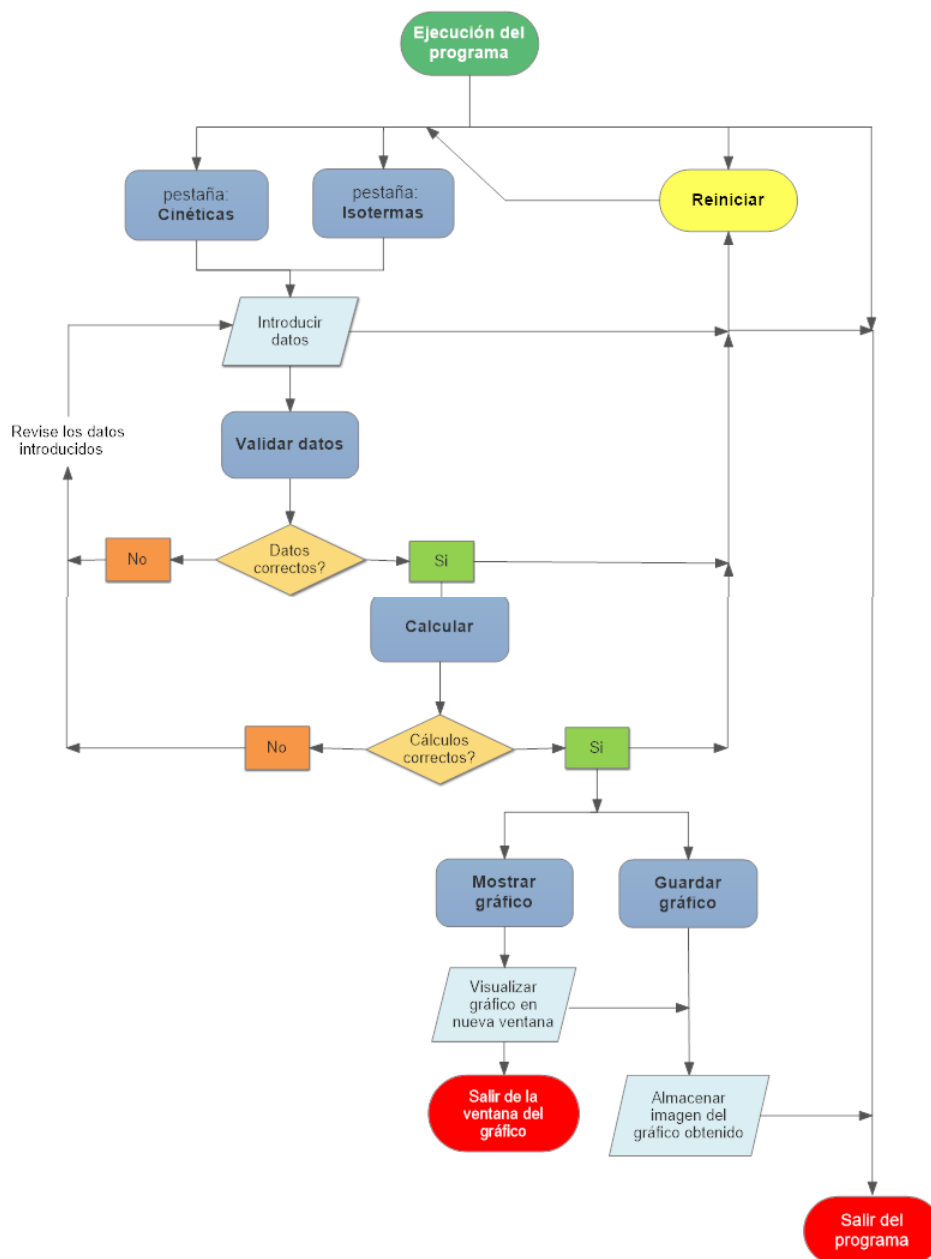


Figura. A.1. Diagrama de funcionamiento del programa.

A.2. Manual de usuario

A continuación se presenta el manual enfocado al usuario, que ha sido diseñado con el fin de facilitar el uso del programa, y conocer las distintas funcionalidades y opciones que ofrece. En este, se incluyen además el manual de instalación del programa, un tutorial paso a paso de su utilización, y un diagrama de funcionamiento del mismo. En el caso de que se desee una explicación más profunda acerca de su funcionamiento, se puede consultar el apartado de desarrollo del programa, o bien el anexo siguiente referente al algoritmo de programación que contiene la aplicación.

- **Instalación del programa**

Requisitos:

- Poseer el archivo del instalador previamente copiado en el ordenador en el cuál se quiera instalar el programa.

Procedimiento:

El programa viene incluido por defecto en un instalador (archivo en formato **.msi*) que se encarga de crear automáticamente una carpeta con todos los archivos necesarios para poder funcionar en cualquier ordenador *Windows*.

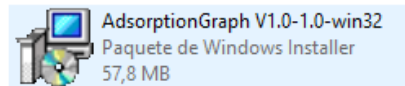


Figura. A.2. Archivo instalador del programa.

Haciendo doble clic sobre él, se abre la ventana de diálogo del instalador. Siguiendo los pasos que se detallan a continuación, resulta muy sencillo instalar el programa en su propio ordenador personal:

1. Seleccionar el directorio en el que se desea crear la carpeta con el contenido del programa. La ruta sugerida por defecto se encuentra en el directorio de Windows (*C:*) y la carpeta creada se llamará "*Adsorción*". No obstante, es posible renombrar la carpeta del programa y modificar el directorio en el que será instalado el programa.

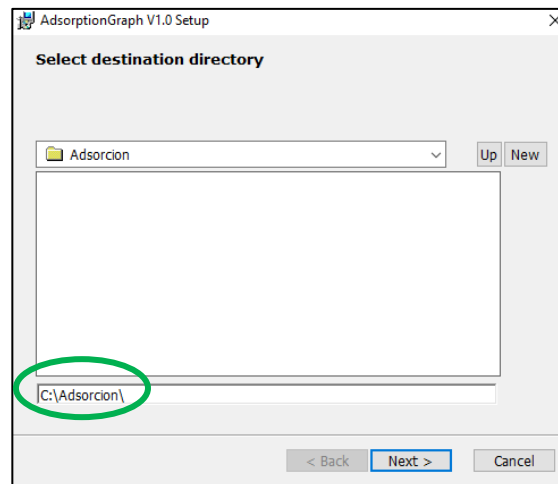


Figura. A.3. Selección del directorio del programa en el instalador.

2. Tras haber definido el directorio y la carpeta de destino del programa, pulse sobre el botón "Next". A continuación, aparecerá una pantalla con la barra de progreso para la instalación de los archivos del programa.

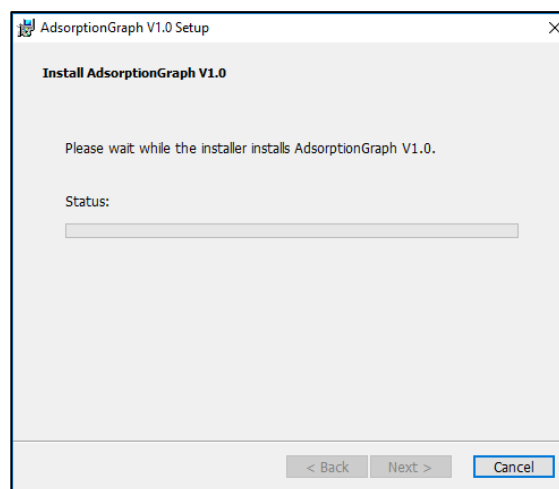


Figura. A.4. Inicio de la instalación del programa.

3. Antes de que el instalador empiece a copiar los archivos a nuestro ordenador, será requerido un permiso de usuario. Para iniciar la instalación, tan sólo es necesario confirmar la instalación haciendo clic en el botón “*Si*”.

Nota: Este paso de confirmación puede tardar unos segundos en aparecer dependiendo del ordenador.

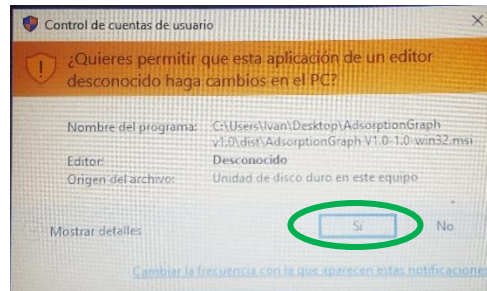


Figura. A.5. Permiso de usuario para la instalación de la aplicación.

4. A continuación, se iniciará la instalación del programa. Una vez finalizada, pulse sobre el botón “*Finish*” y la ventana del instalador se cerrará.

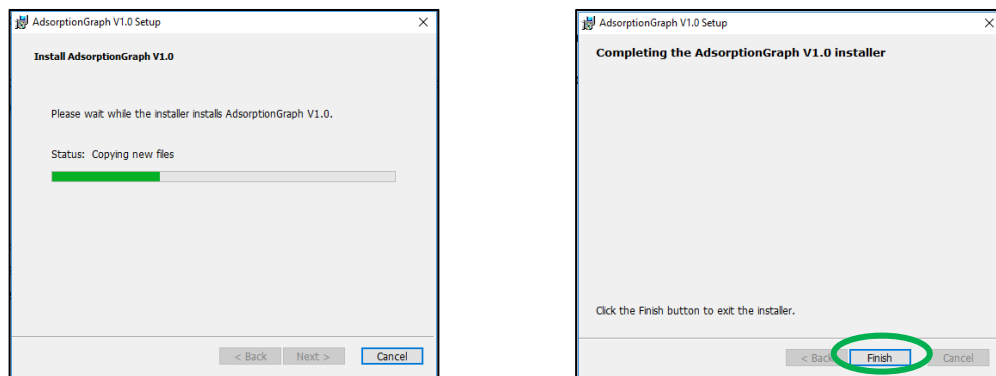


Figura. A.6. Finalización de la instalación del programa.

5. Una vez completada la instalación, diríjase a la carpeta dónde se ha instalado el programa (Por defecto se encuentra en “*C:\Adsorción*”). En ella aparecerán todos los archivos que se han copiado para que pueda funcionar en su ordenador. Entre ellos, se encuentra el archivo “*AdsorptionGraph*”, que es el ejecutable (*.exe*) de la aplicación. Haciendo doble clic sobre él, se ejecutará la aplicación.

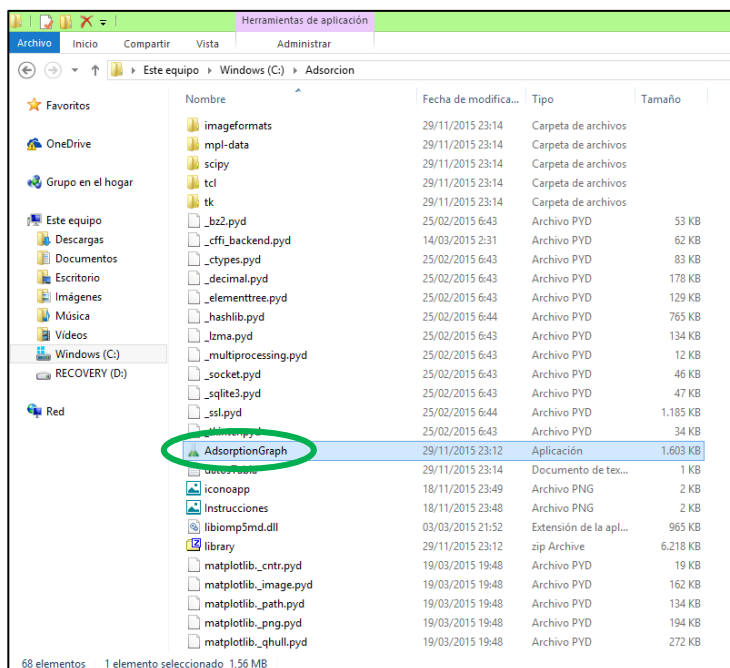


Figura. A.7. Localización del archivo ejecutable de la aplicación.

6. Con el fin de hacer la aplicación más accesible y sin necesidad de tener que dirigirse a la carpeta creada cada vez que se quiera usar el programa, se recomienda crear un acceso directo a este archivo ejecutable, y copiarlo en el *Escritorio* del ordenador. Para ello, basta con pulsar el botón derecho del ratón sobre el archivo *AdsorptionGraph* y seleccionar la opción del desplegable “**Crear acceso directo**”. Finalmente, copie el acceso directo de la aplicación en el *Escritorio* del ordenador.

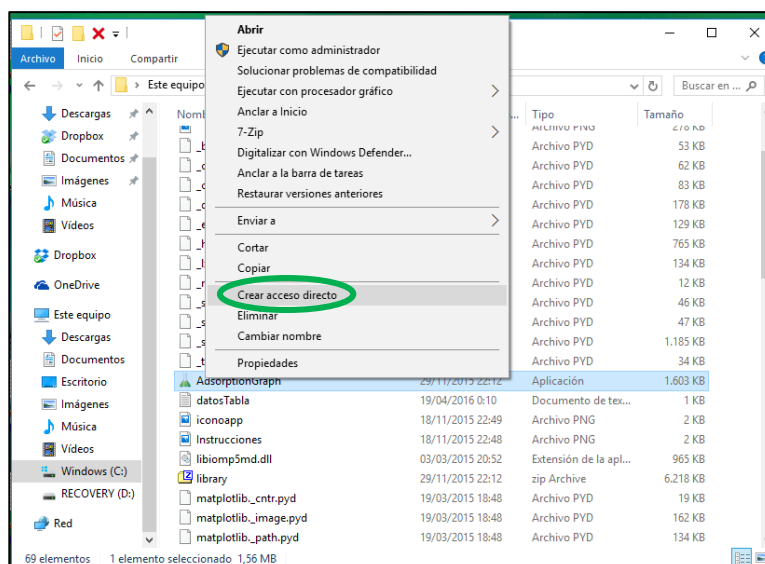


Figura. A.8. Creación de un acceso directo a la aplicación.

ssl.pyd	25/02/2015 5:44	Archivo PYD	1.185 KB
tkinter.pyd	25/02/2015 5:43	Archivo PYD	34 KB
AdsorptionGraph - Acceso directo	07/05/2016 13:49	Acceso directo	2 KB
AdsorptionGraph	29/11/2015 22:12	Aplicación	1.603 KB
datosTabla	19/04/2016 0:10	Documento de tex...	1 KB
iconoapp	18/11/2015 22:49	Archivo PNG	2 KB
Instrucciones	18/11/2015 22:48	Archivo PNG	2 KB
libiomp5md.dll	03/03/2015 20:52	Extensión de la apl...	965 KB

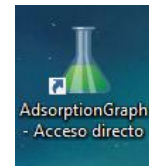


Figura. A.9. Acceso directo de la aplicación

- Al hacer doble clic sobre el icono del acceso directo se ejecutará el programa, aparecerá la ventana del menú principal, y ¡podrá empezar a usarlo!

- Reparación y desinstalación del programa**

Requisitos:

- Tener la aplicación instalada previamente en el ordenador.
- Poseer el archivo del instalador en el ordenador en el cuál se disponga a reparar o desinstalar el programa.

Procedimiento:

El asistente instalador, también ofrece la posibilidad de reparar o desinstalar correctamente el programa procediendo del mismo modo que para la instalación del mismo.

- Si experimenta errores o malfuncionamiento con el uso del programa, es posible que deba a que alguno de los archivos que lo compone haya sido modificado y/o eliminado por error. En este caso, seleccionando la opción de **“Repair”**, el instalador reparará los archivos del programa dañados, para que pueda volver a funcionar correctamente.
- Sin embargo, si desea sencillamente eliminar la aplicación de su ordenador, junto con todos los archivos asociados a ésta, seleccione en este caso la casilla **“Remove”**.

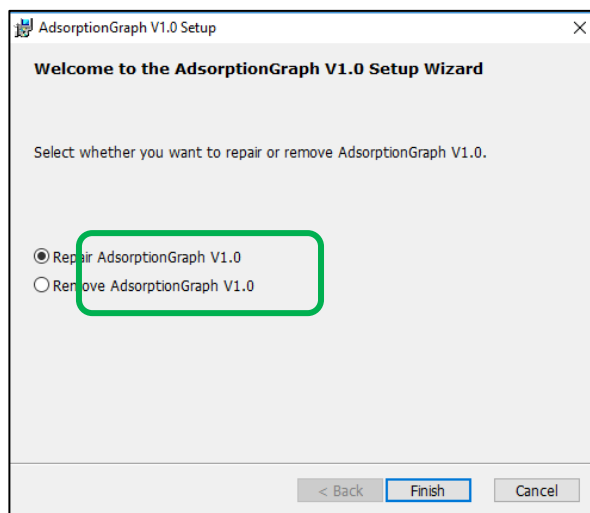


Figura. A.10. Selección para reparar o eliminar el programa.

Tras seleccionar la opción deseada, haga clic sobre el botón **“Finish”**. Seguidamente, se iniciará el proceso de reparación o desinstalación del programa. Una vez completada pulse de nuevo en **“Finish”** para salir del instalador.

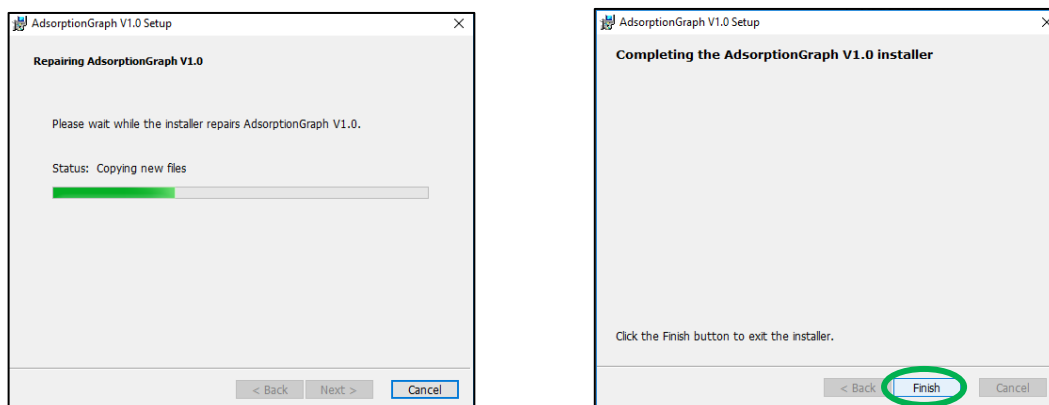


Figura. A.11. Finalizar la reparación/eliminación del programa.

- **Instrucciones de uso del programa**

En este apartado se facilitarán las instrucciones de uso del programa y algunos consejos con el fin de poder hacer un correcto uso del mismo. Si lo desea, puede consultar también las instrucciones en el apartado **“Ayuda/Instrucciones”** que encontrará en la barra del menú principal del programa. Los pasos a seguir son los siguientes:

- 1- En primer lugar, seleccione la pestaña “**Cinética**” o bien “**Isotermas**”, según el tipo de estudio que desee realizar.
- 2- A continuación, deberá rellenar las casillas indicadas del formulario. Los datos que requiere el programa para funcionar hacen referencia a lo siguiente:
 - **Nombre Elemento:** Es el nombre que aparecerá en el título de los gráficos. Se admiten tanto números como letras. Si no desea introducir ningún título, rellene esta casilla con un espacio “ ”.
 - **Volumen disolución [L] , Peso del sólido [g] , Área superficial [m²/g]:**
 - Deberá introducir un valor numérico en estas casillas.
 - Los números decimales deben llevar “.” como separador en lugar de “,”.
 - Se admiten números en formato científico.
Ejemplo: 1.00E-3 → 0.001
 - **Tabla de valores:**
 - Deberá introducir los valores que desee representar en los gráficos, utilizando las unidades especificadas en cada tabla (según si elige un estudio de cinética o isotermas).
 - Evite dejar líneas en blanco en el campo de texto para evitar errores en la ejecución.
 - Deje un espacio entre los valores de cada columna. También puede usar la tecla de *Tabulador*.
 - En este caso, los valores con decimales pueden ser introducidos usando los separadores “.” y “,”. También se aceptan números en formato científico.
 - Si posee los valores de la tabla en un archivo de *Bloc de notas* (*.txt) o en un *Excel*, puede introducirlos fácilmente copiándolos y pegándolos directamente en el campo de texto de la tabla.
- 3- Seleccione un modelo de la lista entre los disponibles.
- 4- Pulse el botón “**Validar datos**”.
 - Si le aparece el mensaje “**Revise los datos introducidos**”, revíselos. El programa le indicará la casilla o casillas que no han sido introducidas correctamente.
 - En caso contrario, aparecerá el mensaje “**Los datos introducidos son correctos**”. En este caso, continúe al siguiente paso.
- 5- Haga clic sobre el botón “**Calcular**”. Si los datos han sido calculados de manera correcta, se activarán los botones “**Mostrar gráfico**” y “**Guardar gráfico**”. En caso de no activarse dichos botones, es posible que con los datos introducidos no sea posible representar el modelo seleccionado, o bien haya ocurrido algún error de cálculo.
- 6- Mediante el botón “**Mostrar gráfico**”, puede visualizar el gráfico obtenido para el modelo seleccionado y los datos introducidos en una nueva ventana.

7- Si desea guardar el gráfico directamente, pulse sobre el botón **“Guardar gráfico”**. A continuación, podrá guardar el gráfico en formato de imagen (*.jpg o *.png) en el directorio que desee, y renombrar la imagen guardada.

Otros consejos útiles:

- Si desea calcular y representar un modelo distinto con los mismos datos introducidos, seleccione otro de los modelos, y repita la secuencia **“Validar datos”-“Calcular”- “Mostrar gráfico”/”Guardar gráfico”**.
- Pulsando el botón **“Reiniciar”** se borrarán todos los valores de las casillas, y el estado del programa volverá a su situación inicial.

• **Herramientas de edición de los gráficos**

En la ventana correspondiente a los resultados representados, además de visualizar los gráficos usted tiene la posibilidad de poder editar y retocar algunos aspectos concretos. Aunque se trate de un editor de gráficos muy básico, éste puede resultar de gran utilidad, puesto que le permitirá ajustar el aspecto final de los gráficos una vez obtenidos, y también guardar la imagen correspondiente con los cambios realizados.

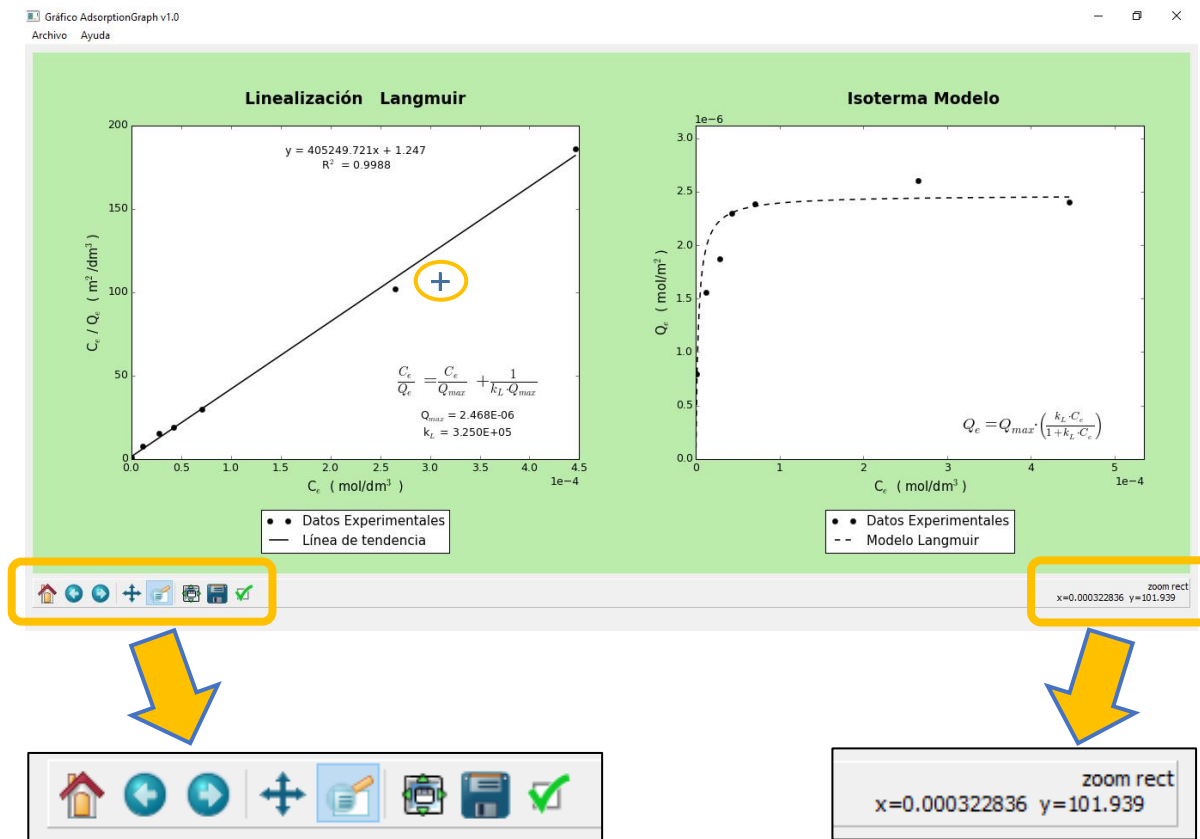






Figura. A.12. Herramientas de edición del gráfico y visualización de las coordenadas del puntero.


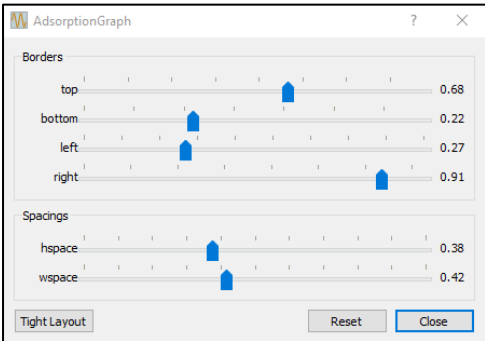


Situando el cursor sobre el área del gráfico, podrá consultar en todo momento las coordenadas de posición del ratón sobre el mismo, en la esquina inferior derecha de la ventana.

Los botones que encontrará en la esquina inferior izquierda le servirán para interactuar con los gráficos.

Si lo desea, podrá consultar también las instrucciones en el apartado **“Ayuda/Instrucciones”** que encontrará en la barra del menú principal del programa.

A continuación, se explica el significado de cada uno de ellos junto con su funcionalidad de edición:

	<p>Botón “Home” (Casa): con este botón podrá deshacer todos los cambios realizados, restableciendo así el gráfico a su aspecto representado por defecto.</p>
	<p>Botones “Deshacer” y “Rehacer”: como su nombre indica, estos botones le permitirán volver hacia atrás y hacia adelante en los distintos cambios realizados.</p>
	<p>Botón “Mover”: esta herramienta le permitirá desplazarse a través del gráfico. Mantenga pulsados los siguientes botones del ratón mientras se desplaza por el gráfico para realizar las acciones pertinentes:</p> <p><u>Botón izquierdo</u> → sirve para mover los ejes del gráfico, ajustando así el área representada en ellos, pero sin variar su escala.</p> <p><u>Botón derecho</u> → se utiliza para extender o acortar la escala de los ejes, representando así un mayor o menor rango de valores de los ejes, empleando en el mismo tamaño de gráfico.</p>
	<p>Botón “Zoom”: con este botón se podrá ampliar o reducir el área representada en los gráficos. Seleccione un recuadro del área representada, manteniendo pulsados los siguientes botones del ratón mientras se desplaza por el gráfico:</p> <p><u>Botón izquierdo</u> → Ampliar el recuadro seleccionado.</p> <p><u>Botón derecho</u> → Reducir el área de selección.</p>

	<p>Botón “Configurar márgenes”: pulsando este botón se abrirá una ventana con las distintas opciones de configuración. Está en inglés, pero a continuación se describen los parámetros para modificar los márgenes de los gráficos:</p> <div style="text-align: center;">  </div> <p style="text-align: center;">Figura. A.13. Parámetros de edición de los márgenes.</p> <p>Los parámetros que modifique serán aplicados en tiempo real en los gráficos, para su visualización. Una vez terminada la edición, cierre la ventana o haga clic en “Close”.</p> <p><u>Borders:</u> podrá ensanchar o estrechar todo el contenido de los gráficos, ajustando los bordes por arriba (<i>top</i>), abajo (<i>bottom</i>), izquierda (<i>left</i>) y derecha (<i>right</i>).</p> <p><u>Spacings:</u> sirve para aumentar o reducir el espaciado entre gráficos, en el caso de que sea representado más de uno a la vez en la misma ventana. El espaciado entre gráficos puede ser horizontal (<i>wspace</i>) o vertical (<i>hspace</i>).</p>
	<p>Botón “Guardar”: Haciendo clic sobre este botón podrá guardar la ventana del gráfico formato de imagen, manteniendo los cambios realizados. También puede realizar esta acción desde la ventana principal del programa.</p>
	<p>Botón “Personalizar” o “Customize”: Pulsando el icono de este botón accederá a una ventana en la que tendrá que seleccionar el gráfico que desee personalizar:</p>

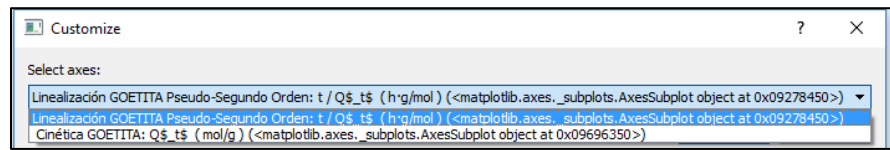


Figura. A.14. Selección del gráfico a personalizar.

A continuación se abrirá un nuevo menú con dos opciones distintas de edición, separadas por pestañas:

Axis: Permite la edición del título del gráfico, así como el rango de valores de los ejes, su escala, y el nombre de cada eje.

Curves: Seleccionando la serie de valores deseada, podrá editar el estilo de los valores representados, permitiendo modificar la geometría, el tamaño y el color tanto de las líneas como de los marcadores.

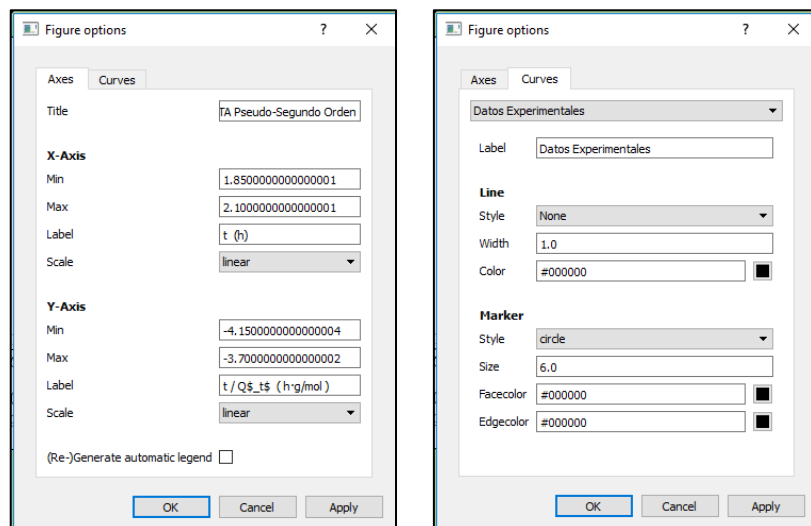


Figura. A.15. Menú pestañas de personalización: “Axis” y “Curves”.

Para aplicar los cambios pulse en “OK” o bien en “Apply”. Si desea restablecer los cambios realizados, siga los siguientes pasos:

1. Cierre el gráfico actual.
2. En el menú principal, pulse “Calcular”.
3. Vuelva a generar el gráfico por defecto pulsando el botón “Mostrar gráficos”.

Anexo B

B.1. Estructura del algoritmo

El esquema siguiente muestra la organización de los ficheros que contienen el algoritmo del programa. Como puede observarse, los diferentes *scripts* están relacionados siguiendo una estructura jerárquica, donde “*Interfaz_grafica.py*” es el documento principal. Cada uno de estos archivos posee una funcionalidad bien definida, que se detallará en el apartado siguiente.

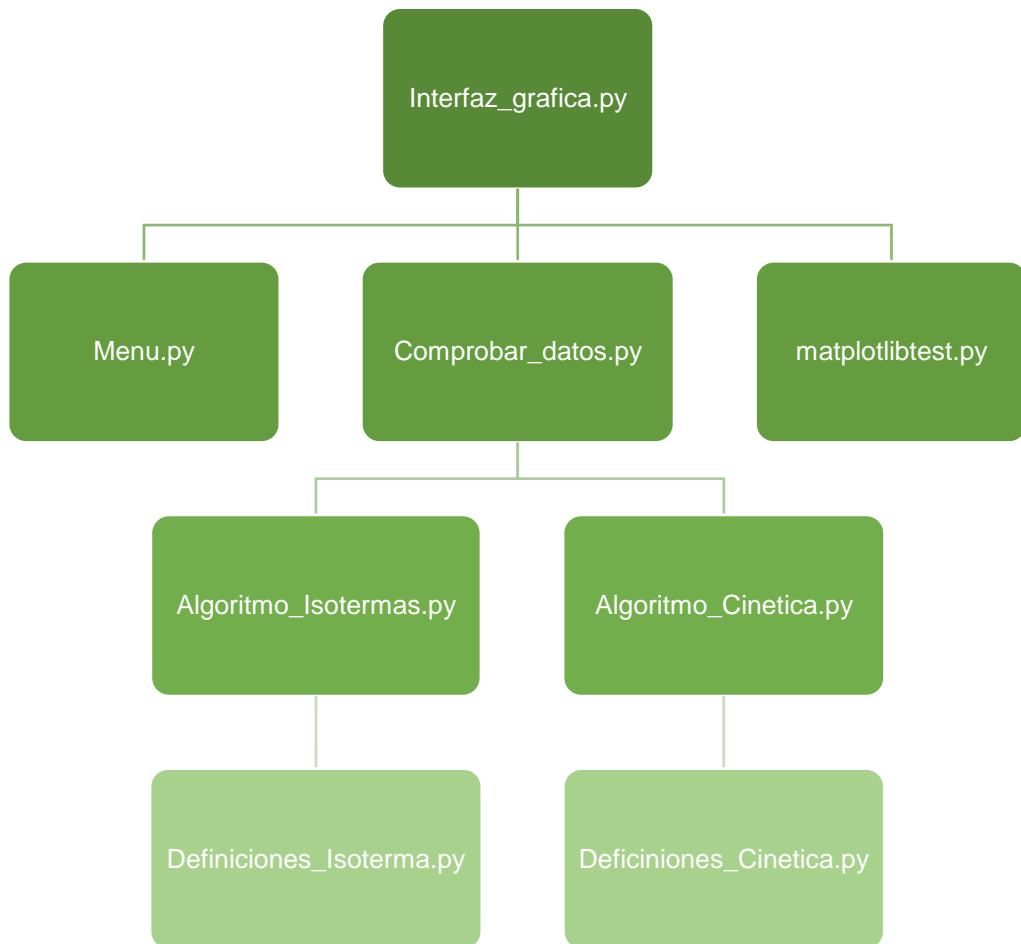


Figura. B.1. Estructura y organización de los archivos de *Python* que componen el algoritmo del programa.

B.2. Scripts de Python

Definiciones_Cinetica.py

Archivo: Definiciones_Cinetica.py

Bibliotecas necesarias: *numpy, pylab, matplotlib*

Funciones:

1. Unidades_Elovich (AreaSuperficial)

Función: Asigna los valores de las etiquetas de los gráficos adecuándolos al modelo cinético de *Elovich*: el título del gráfico, el nombre de los ejes y las unidades de los valores representados. En función de si se conoce o no el valor del área superficial, las unidades que presenta el gráfico serán diferentes.

También contiene la ecuación del modelo de *Elovich* a partir de la cual se calculan los valores de concentración para cada valor de tiempo introducido (variable *y_modelo*).

Entrada: AreaSuperficial

Salida: Area_superficial, xlabel_lineal, xlabel_modelo, ylabel_lineal, ylabel_modelo, nombre_modelo, label_modelo, y_modelo

2. Unidades_PseudoSegundo_Orden (AreaSuperficial)

Función: Procediendo del mismo modo que para la función “*Unidades_Elovich*”, ésta asigna los valores correspondientes a las etiquetas de los gráficos adaptándolos al modelo cinético de *pseudo segundo orden*: el nombre del gráfico, la etiqueta de los ejes y las unidades en que se presentan. En función de si se conoce o no el valor del área superficial, éstos tomarán unos valores u otros.

Además, también ofrece la ecuación del modelo de *pseudo segundo orden* para el cálculo de concentraciones (variable *y_modelo*).

Entrada: AreaSuperficial

Salida: Area_superficial, xlabel_lineal, xlabel_modelo, ylabel_lineal, ylabel_modelo, nombre_modelo, label_modelo, y_modelo

3. Linealizacion_Cinetica (numero_modelo, matriz, Vol_disolucion, Area_superficial, Peso_solido)

Función: Calcula una matriz con los valores linealizados siguiendo la ecuación del modelo seleccionado. Posteriormente, con la matriz obtenida realiza un ajuste

lineal y obtiene los valores de la pendiente (m) y la ordenada en el origen (n) de la recta resultante.

Entrada: AreaSuperficial

Salida: m , n , t_{Qe} (matriz de tiempo frente a concentración en el equilibrio), matrizresultado (matriz t_{Qe} traspuesta), Q_e (matriz de concentraciones en el equilibrio)

4. Parametros_Modelo_Cinetica (numero_modelo, mineral)

Función: Calcula los valores de las constantes correspondientes al modelo cinético seleccionado.

Entrada: numero_modelo, mineral (título del caso de estudio)

Salida: Modelo de *Elovich* \rightarrow Conc_eq, K
Modelo *pseudo segundo orden* \rightarrow a, b

5. Grafico_Linealizacion_Cinetica (fig, mineral, xdatos_lineal, ydatos_lineal, numero_modelo)

Función: Representa el gráfico de linealización del modelo seleccionado.

Entrada: fig, mineral, xdatos_lineal, ydatos_lineal, numero_modelo

Salida: ax1 (Gráfico de linealización)

6. Modelizacion_Cinetica (matriz, matrix, numero_modelo)

Función: Realiza los cálculos de la ecuación del modelo para obtener los valores de concentración del modelo teórico seleccionado.

Entrada: matriz, matrix, numero_modelo

Salida: cinetica_modelo, cinetica_modelizada, x_cinetica, y_cinetica, Q_e

7. Grafico_Modelizacion_Cinetica (fig, mineral, xdatos_modelo, numero_modelo)

Función: Representa un gráfico dónde se visualiza el modelo teórico y los valores del modelo experimental introducidos.

Entrada: fig, mineral, xdatos_modelo, numero_modelo

Salida: ax2 (Gráfico de modelización)

8. Comparacion_Cinetica (fig, xdatos_modelo, numero_modelo)

Función: Representa un nuevo gráfico donde se visualizan todos los modelos cinéticos teóricos.

Entrada: fig, xdatos_modelo, numero_modelo

Salida: ax3 (Gráfico de comparación)

9. Grafico_Comparacion_Cinetica (fig, mineral)

Función: Añade la serie de valores experimentales al gráfico de comparación y establece los títulos del gráfico y de sus ejes.

Entrada: fig, mineral

Salida: ax3 (Gráfico de comparación)

Algoritmo_Cinetica.py

Archivo: Algoritmo_Cinetica.py

Bibliotecas necesarias: numpy, pylab, Definiciones_Cinetica, Interfaz_grafica

Funciones:

1. EjecutarC (numero_modelo, Area_superficial, Peso_solido, Vol_disolucion, mineral, matriz, AreaSuperficial)

Función: Recopila los datos introducidos pertenecientes a los estudios de cinética de adsorción y ejecuta los algoritmos correspondientes al modelo seleccionado, realizando los cálculos pertinentes.

Entrada: numero_modelo, Area_superficial, Peso_solido, Vol_disolucion, mineral, matriz, AreaSuperficial

Salida: Según el modelo cinético de adsorción seleccionado, representa el gráfico cinético resultante.

Definiciones_Isoterma.py

Archivo: Definiciones_Isoterma.py

Bibliotecas necesarias: *numpy, pylab, matplotlib*

Funciones:

1. Unidades_Langmuir (AreaSuperficial)

Función: Asigna los valores siguientes de las etiquetas de los gráficos correspondientes a la isoterma de *Langmuir*: el título del gráfico, el nombre de los ejes y las unidades de los valores representados. En función de si se conoce o no el valor del área superficial, las unidades que presente el gráfico serán diferentes. También contiene la ecuación de *Langmuir* a partir de la cual se calculan los valores de concentraciones en el equilibrio (variable *y_modelo*).

Entrada: AreaSuperficial

Salida: Area_superficial, xlabel_lineal, xlabel_modelo, ylabel_lineal, ylabel_modelo, nombre_modelo, label_modelo, y_modelo

2. Unidades_Freundlich (AreaSuperficial)

Función: Procediendo del mismo modo que para la función “*Unidades_Langmuir*”, ésta asigna los valores correspondientes a las etiquetas de los gráficos adaptándolos al modelo de la isoterma de *Freundlich*: el nombre del gráfico, la etiqueta de los ejes y las unidades en que se presentan. En función de si se conoce o no el valor del área superficial, éstos tomarán unos valores u otros. Además, también ofrece la ecuación de *Freundlich* para el cálculo de concentraciones (variable *y_modelo*).

Entrada: AreaSuperficial

Salida: Area_superficial, xlabel_lineal, xlabel_modelo, ylabel_lineal, ylabel_modelo, nombre_modelo, label_modelo, y_modelo

3. Linealizacion_Isoterma (numero_modelo, matriz, Vol_disolucion, Area_superficial, Peso_solido)

Función: Calcula una matriz con los valores linealizados siguiendo la ecuación del modelo seleccionado. Posteriormente, con la matriz obtenida realiza un ajuste lineal y obtiene los valores de la pendiente (*m*) y la ordenada en el origen (*n*) de la recta resultante.

Entrada: AreaSuperficial

Salida: m, n, Ce_Qe (matriz de tiempo frente a concentración en el equilibrio), matriz resultado (matriz Ce_Qe traspuesta), Qe (matriz de concentraciones en el equilibrio)

4. Parametros_Modelo_Isoterma (numero_modelo, mineral)

Función: Calcula los valores de las constantes correspondientes al modelo de isoterma seleccionado.

Entrada: numero_modelo, mineral (título del caso de estudio)

Salida: Isoterma de Langmuir → Conc_max, KL
Isoterma de Freundlich → K, N

5. Grafico_Linealizacion_Isoterma (fig, mineral, xdatos_lineal, ydatos_lineal, numero_modelo)

Función: Representa el gráfico de linealización del modelo seleccionado.

Entrada: fig, mineral, xdatos_lineal, ydatos_lineal, numero_modelo

Salida: ax1 (Gráfico de linealización)

6. Modelizacion_Isoterma (matriz, matrix, numero_modelo)

Función: Realiza los cálculos de la ecuación del modelo para obtener los valores de concentración del modelo teórico seleccionado.

Entrada: matriz, matrix, numero_modelo

Salida: isoterma_modelo, isoterma_modelizada, x_isoterma, y_isoterma, Qe

7. Grafico_Modelizacion_Isoterma (fig, mineral, xdatos_modelo, numero_modelo)

Función: Representa un gráfico donde se visualiza el modelo teórico y los valores del modelo experimental introducidos.

Entrada: fig, mineral, xdatos_modelo, numero_modelo

Salida: ax2 (Gráfico de modelización)

8. Comparacion_Isotermas (fig, xdatos_modelo, numero_modelo)

Función: Representa un nuevo gráfico donde se visualizan todas las isotermas de adsorción.

Entrada: fig, xdatos_modelo, numero_modelo

Salida: ax3 (Gráfico de comparación)

9. Grafico_Comparacion_Isoterma (fig, mineral)

Función: Añade la serie de valores experimentales al gráfico de comparación y establece los títulos del gráfico y de sus ejes.

Entrada: fig, mineral

Salida: ax3 (Gráfico de comparación), nombre modelo

Algoritmo_Isotermas.py

Archivo: Algoritmo_Isotermas.py

Bibliotecas necesarias: *numpy, pylab, Definiciones_Isoterma, Interfaz_grafica*

Funciones:

1. EjecutarI (numero_modelo, Area_superficial, Peso_solido, Vol_disolucion, mineral, matriz, AreaSuperficial)

Función: Recopila los datos introducidos pertenecientes a los estudios de isotermas de adsorción y ejecuta los algoritmos correspondientes al modelo seleccionado, realizando los cálculos pertinentes.

Entrada: numero_modelo, Area_superficial, Peso_solido, Vol_disolucion, mineral, matriz, AreaSuperficial

Salida: Según el modelo de isoterma de adsorción seleccionado, representa el gráfico de la isoterma resultante.

Comprobar_Datos.py

Archivo: Comprobar_datos.py

Bibliotecas necesarias: numpy, PyQt4, Menu, Algoritmo_Cinetica, Algoritmo_Isotermas

Funciones:

Clase ManejoDatos (QtGui.QMainWindow)

1. ComprobarDatos (self, A)

Función: En esta función se declara una variable para cada widget contenido en la interfaz gráfica. También asigna un valor a la variable "numero_modelo" en función del modelo seleccionado, tanto para las cinéticas ($A=1$) como para las isotermas de adsorción ($A=2$).

Finalmente, activa un mensaje en función de si los datos introducidos son válidos o no.

Entrada: A

Salida: Activa/Desactiva los siguientes mensajes de error (etiquetas de la interfaz gráfica): ErrorModelol, ErrorModeloC, DatosIncorrectos

2. LeerDatos (self, ValorElemento, ErrorElemento, ValorVolumen, ErrorVolumen, ValorPeso, ErrorPeso, ValorAreaSup, ErrorAreaSup, A, DatosCorrectos, DatosIncorrectos, Tabla, ErrorTabla)

Función: Almacena los valores introducidos en los diferentes campos del formulario por parte del usuario, y en función de si pueden ser almacenados o no en el formato de dato especificado, muestran su respectivo mensaje de error en la interfaz gráfica.

Entrada: self, ValorElemento, ErrorElemento, ValorVolumen, ErrorVolumen, ValorPeso, ErrorPeso, ValorAreaSup, ErrorAreaSup, A, DatosCorrectos, DatosIncorrectos, Tabla, ErrorTabla

Salida: LeerTabla, Activa/Desactiva los siguientes mensajes de error (etiquetas de la interfaz gráfica): ErrorElemento, ErrorPeso, ErrorVolumen, ErrorAreaSup

3. Leer_matriz (self, A, DatosIncorrectos, Tabla, ErrorTabla)

Función: Abre el archivo de texto previamente creado y lee una a una todas las líneas de texto a medida que va almacenando los valores introducidos en una matriz. En caso de producirse algún error durante este proceso, muestra su particular mensaje de error en la interfaz gráfica.

Entrada: A, DatosIncorrectos, Tabla, ErrorTabla

Salida: matriz, Activa/Desactiva mensaje de error ErrorTabla

Interfaz_grafica.py

Archivo: Interfaz_grafica.py

Bibliotecas necesarias: *numpy, PyQt4, Menu, Comprobar_datos*

Funciones:

Clase MiFormulario(QtGui.QMainWindow)

1. __init__ (self)

Función: Contiene las señales y acciones a realizar de todos los elementos de la interfaz gráfica, asignándoles a cada una su función correspondiente.

2. Instrucciones (self)

Función: Esta función contiene el texto con las instrucciones que se muestran en la ventana principal de la interfaz gráfica.

3. RadioAreaSuperficialC (self)

Función: Se encarga de reestablecer el valor inicial del área superficial que se muestra en la pestaña de Cinéticas de la interfaz gráfica, adoptando 1 por valor.

4. RadioAreaSuperficialI (self)

Función: Se encarga de reestablecer el valor inicial del área superficial que se muestra en la pestaña de Isotermas de la interfaz gráfica, adoptando 1 por valor.

5. ValidarCinetica (self)

Biblioteca: *Comprobar_datos*

Función: En función del mensaje obtenido sobre el almacenamiento de los datos de la interfaz gráfica (Datos Correctos/Incorrectos) pertenecientes a la pestaña de Cinéticas, activa/desactiva los botones que permiten continuar con la resolución del caso y realizar los cálculos respectivos.

6. ValidarIsoterma (self)

Biblioteca: Comprobar_datos

Función: En función del mensaje obtenido sobre el almacenamiento de los datos de la interfaz gráfica (Datos Correctos/Incorrectos) pertenecientes a la pestaña de Isotermas, activa/desactiva los botones que permiten continuar con la resolución del caso y realizar los cálculos respectivos.

7. CalcularCinetica (self)

Biblioteca: Comprobar_datos

Función: Una vez pulsado el botón de “Calcular” de la pestaña Cinéticas, ejecuta el algoritmo contenido en la biblioteca *Algoritmo_Cinetica* y si no se produce ningún error, activa los botones de “Mostrar Grafico” y “Guardar Gráfico” de la pestaña mencionada.

8. CalcularCinetica (self)

Biblioteca: Comprobar_datos

Función: Una vez pulsado el botón de “Calcular” de la pestaña Isotermas, ejecuta el algoritmo contenido en la biblioteca *Algoritmo_Isoterma* y si no se produce ningún error, activa los botones de “Mostrar Grafico” y “Guardar Gráfico” pertenecientes a dicha pestaña.

9. GraficoCinetica (self)

Biblioteca: matplotlibtest, Definiciones_Cinetica, Algoritmo_Cinetica

Función: Abre/Actualiza la ventana de la interfaz gráfica que contiene los gráficos con los resultados obtenidos para el caso cinético de estudio.

10. GraficoIsoterma (self)

Biblioteca: matplotlibtest, Definiciones_Isoterma, Algoritmo_Isotermas

Función: Abre/Actualiza la ventana de la interfaz gráfica que contiene los gráficos con los resultados obtenidos para el caso de isoterma estudiado.

11. GuardarImagenCinetica (self)

Biblioteca: Algoritmo_Cinetica

Función: Abre un cuadro de diálogo para guardar el gráfico cinético obtenido y se encarga de almacenarlo en formato de imagen con el nombre especificado.

12. GuardarImagenIsoterma (self)

Biblioteca: Algoritmo_Isotermas

Función: Abre un cuadro de diálogo para guardar el gráfico de isoterma obtenido y se encarga de almacenarlo en formato de imagen con el nombre especificado.

13. Reiniciar (self)

Función: Reestablece todos los elementos de la interfaz gráfica a su estado inicial en el que se encuentran al ejecutar el programa.

matplotlibtest.py

Archivo: matplotlibtest.py

Bibliotecas necesarias: PyQt4, numpy, matplotlib, Algoritmo_Isotermas, Algoritmo_Cinetica, Comprobar_datos

Funciones:

Clase Qt4MplCanvas(FigureCanvas)

1. **__init__ (self)**

Función: Según si se trata de un caso de estudio cinético ($C=1$) o de isoterma ($C=2$) imprime el gráfico respectivo obtenido en la ventana creada.

Clase ApplicationWindow(QtGui.QMainWindow)

1. **__init__ (self)**

Función: crea la ventana donde se presentan los gráficos, e inicializa todos los elementos de ésta.

2. fileQuit(self):

Función: Contiene la acción de cerrar la ventana creada.

3. closeEvent(self, ce):

Función: Realiza la acción de cerrar la ventana creada

4. about(self):

Función: Muestra y contiene las instrucciones de uso que aparecen en el apartado de “Ayuda” de la ventana del gráfico.

Menu.py

Archivo: Menu.py

Bibliotecas necesarias: PyQt4

Función: Este archivo contiene toda la información sobre el diseño y la apariencia de la interfaz gráfica de usuario: el tipo de objeto, tamaño, estilo, color, fuente... de todos los elementos que la conforman.

Este archivo se obtuvo a partir de *QtCreator*, el cual permite crear el diseño de la interfaz de forma gráfica, y posteriormente generar un archivo traducido a código de programación.

setup.py

Archivo: setup.py

Bibliotecas necesarias: cx_Freeze

Función: Este documento hace referencia al constructor del ejecutable. En él se definen todas las instrucciones y métodos para la creación del archivo ejecutable. También se encarga de recopilar todos los archivos y bibliotecas para que el programa pueda funcionar, y contiene información acerca de la aplicación: el nombre, el autor, la versión, las imágenes de los iconos, etc.