



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA QUÍMICA

IMPLEMENTACIÓN DE UNIDADES DE SEPARACIÓN EN EL SIMULADOR DE PROCESOS COCO. APLICACIÓN AL DISEÑO DE UN PROCESO DE REGENERACIÓN DE AGUAS DE LAVADO DE RECUBRIMIENTOS METÁLICOS

AUTOR: Miguel Sanchis Sebastián

TUTOR: Asunción Santafé Moros

COTUTOR: José M. Gozávez Zafrilla

Curso Académico: 2013-14

ÍNDICE DE DOCUMENTOS

I. MEMORIA DESCRIPTIVA	1
II. ANEXOS A LA MEMORIA	61
III. PLIEGO DE CONDICIONES.....	75
IV. PRESUPUESTO	83



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA QUÍMICA

I. MEMORIA DESCRIPTIVA

Curso Académico: 2013-14

ÍNDICE MEMORIA DESCRIPTIVA

1. Objetivos y alcance del proyecto	5
1.1. Establecimiento de los objetivos	5
1.2. Definición del alcance	6
1.3. Estructura de descomposición del trabajo (EDT).....	6
1.4. Planificación del proyecto.....	8
2. Introducción.....	11
2.1. Motivación	11
2.2. Antecedentes y estado actual.....	11
2.2.1. La tecnología CAPE-OPEN	11
2.2.2. Simulación de modelos desarrollados por el usuario.....	13
2.2.3. Simulación de unidades de membrana	13
2.2.4. Simulación de procesos de enjuague.....	14
2.2.5. Estado actual.....	15
2.3. Justificación técnica	17
2.4. Justificación académica.....	17
3. Análisis de alternativas de simuladores	19
3.1. Descripción de las alternativas existentes de simuladores de procesos químicos ...	19
3.1.1. CHEMCAD.....	20
3.1.2. COCO simulator.....	21
3.2. Selección del simulador de procesos	22
4. Descripción de los modelos físicos utilizados	25
4.1. Modelo para una etapa de membrana	25
4.1.1. Cálculo de los fluxes mediante el modelo de disolución-difusión.....	26
4.1.2. Efecto de la concentración por polarización	28
4.2. Modelo para un tanque de lavado.....	30
4.3. Modelo para un equipo de intercambio iónico	31
5. Implementación de funcionalidades adicionales en COCO.....	33
5.1. La unidad MATLAB en COCO	33
5.2. Implementación de una etapa de membrana	36
5.3. Implementación de un tanque de lavado	39
5.4. Implementación de un intercambio iónico.....	40
5.5. Introducción de componentes en la base de datos de COCO	40
6. Simulación de un proceso industrial	43

6.1. Alternativas de mejora del proceso analizadas	43
6.1.1. Modificación de la posición de recirculación.....	43
6.1.2. Utilización de una segunda etapa de membrana	45
6.2. Justificación económica de la mejora planteada	46
6.2.1. Coste inicial del tratamiento de agua	46
6.2.2. Coste del sistema de tratamiento mejorado	48
7. Conclusiones	55
8. Lista de símbolos	57
9. Bibliografía.....	59

1. OBJETIVOS Y ALCANCE DEL PROYECTO

1.1. Establecimiento de los objetivos

Objetivo principal: Introducir unidades de separación adicionales en un simulador de procesos para simular un proceso de regeneración de aguas industriales y además analizar diferentes alternativas de mejora de dicho proceso. Previamente a ello, realizar un análisis de las diferentes alternativas de simulador para determinar cuál de ellas resulta más viable desde el punto de vista de complejidad y coste.

Objetivos parciales:

1. Seleccionar el simulador de procesos
 - 1.1. Búsqueda de diferentes alternativas de simuladores disponibles
 - 1.2. Estudio de la posibilidad de introducir unidades de usuario en cada alternativa
 - 1.3. Realización de una matriz de decisión para seleccionar una alternativa
2. Implementación de unidades de cálculo
 - 2.1. Definición del comportamiento de cada unidad mediante un modelo matemático
 - 2.2. Introducción de los modelos en el simulador de procesos
 - 2.3. Validación del funcionamiento de la unidad de ósmosis inversa con un caso base
3. Simulación de un proceso de regeneración de aguas industriales
 - 3.1. Búsqueda bibliográfica de alternativas de proceso
 - 3.2. Selección del proceso a simular
 - 3.3. Definición de las unidades de proceso y simulación del conjunto
 - 3.4. Planteamiento y justificación de una mejora del proceso mediante el uso del simulador

1.2. Definición del alcance

El objeto de este proyecto es implementar módulos de cálculo de unidades de membrana de ósmosis inversa y otras unidades de separación en un simulador de procesos, con el propósito de llegar a simular un proceso de regeneración de aguas industriales. Una vez conseguido esto, se planteará una mejora del proceso mediante el uso del simulador para así demostrar la utilidad a nivel industrial de esta herramienta informática.

Para alcanzar el objeto del proyecto lo primero que se realizará es una exploración de los distintos simuladores de procesos que existen actualmente, prestando especial atención a la posibilidad de introducir unidades personalizadas en ellos. Atendiendo a diferentes criterios, como por ejemplo coste económico o complejidad en el desarrollo, se deberá seleccionar una de las alternativas de simulador planteadas.

Una vez determinado el simulador de procesos que se utilizará, se deberán desarrollar los modelos que representen el comportamiento de las unidades de separación que no estén incluidas previamente en el programa. Tras finalizar el desarrollo de los modelos, estos deberán implementarse en el simulador para así poder utilizarlos en la simulación.

Dado que el proyecto se centra en las unidades de ósmosis inversa, el modelo de dicha unidad de separación será de una mayor complejidad y precisión que el resto de modelos desarrollados. Además, dicho modelo será validado a través de la resolución de un caso base, lo cual no se realizará en el resto de unidades de separación.

Además de la implementación de las unidades de cálculo, será necesario seleccionar un proceso de regeneración de aguas industriales con el cual pueda ilustrarse la capacidad del simulador para reproducir procesos donde aparezcan unidades de ósmosis inversa mediante el uso de la unidad implementada en él. Así mismo, se planteará una mejora del proceso seleccionado para demostrar la posibilidad de utilizar el simulador de procesos como herramienta de optimización y diseño de procesos industriales.

Es importante destacar que la aplicación de cálculo desarrollada no pretende tener carácter comercial sino ser una herramienta útil y versátil para su uso a nivel universitario, tanto por el personal docente como el alumnado.

Por último, aunque queda fuera del alcance del proyecto, el trabajo realizado a lo largo de este ha dado pie a la realización de un póster que fue enviado al Congreso Iberoamericano de Ciencia y Tecnología de Membranas (CITEM), el cual puede consultarse en el capítulo 3 de los Anexos junto con su abstract correspondiente, adjuntado en el capítulo 4 de este mismo documento.

1.3. Estructura de descomposición del trabajo (EDT)

En la Figura 1 se presenta la estructura de descomposición del trabajo que se ha realizado para este proyecto, en la cual se indican todas las tareas que se llevarán a cabo, estando estas agrupadas en paquetes de trabajo.

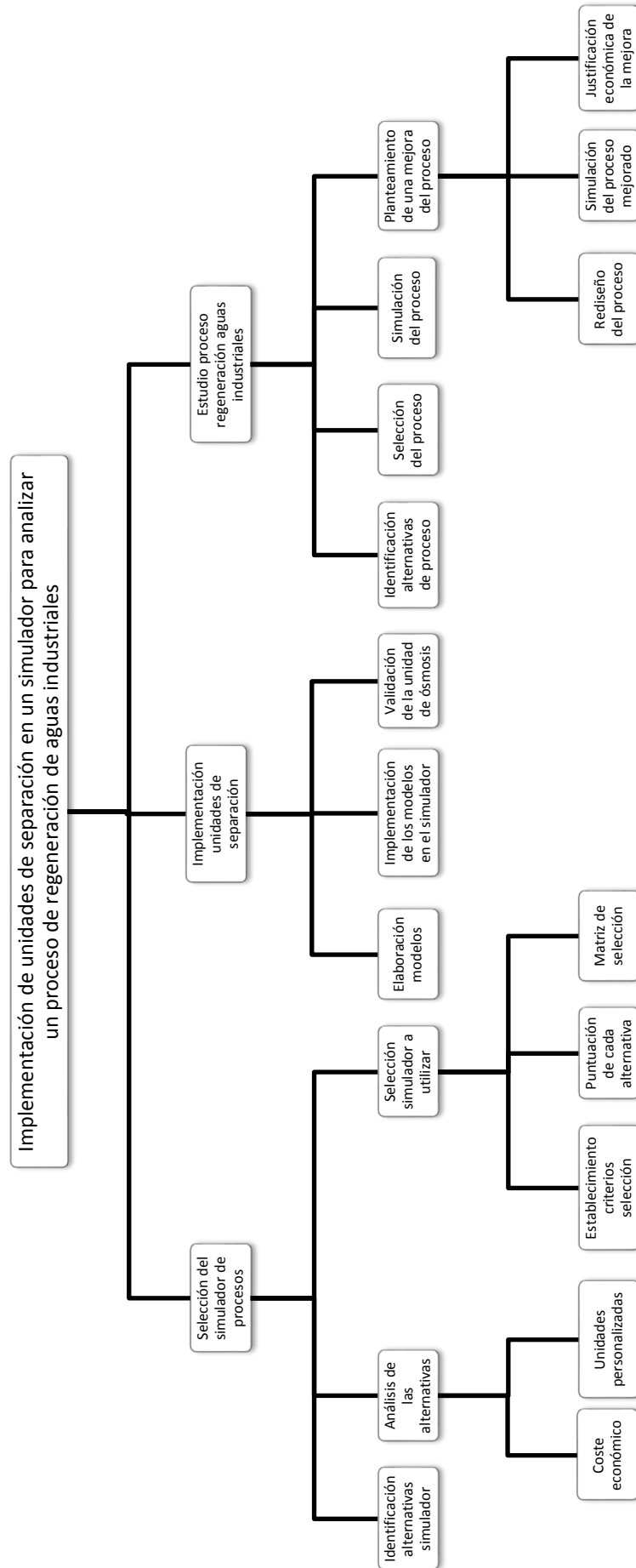


Figura 1. Estructura de descomposición del trabajo del proyecto

1.4. Planificación del proyecto

La planificación de las tareas se ha realizado en base a las duraciones y precedencias que aparecen en la Tabla 1. Para establecer las precedencias, con la intención de realizar una planificación más realista, se ha tomado como criterio que el proyecto es realizado por una Ingeniería, la cual dedica un programador, graduado en Ingeniería Química e Ingeniero Químico a la realización del proyecto.

Tabla 1. Duraciones y precedencias de las tareas a realizar en el proyecto

<i>Fase</i>	<i>Código</i>	<i>Tarea</i>	<i>Duración(h)</i>	<i>Precedencias</i>
Selección del simulador	1.1	Identificación alternativas	5	-
	1.2	Análisis prestaciones	31	1.1
	1.3	Posibilidad de introducir unidades	17	1.1
	1.4	Determinar coste licencia de uso	5	1.2
	1.5	Introducción ud. personalizada sencilla	9	1.3
	1.6	Establecer criterios selección	5	-
	1.7	Selección alternativa	2	1.4, 1.5, 1.6
Modelado unidades separación	2.1	Definición modelo ósmosis inversa	40	-
	2.2	Codificación modelo ósmosis inversa	46	2.1
	2.3	Definición modelo tanque lavado	8	2.1
	2.4	Codificación modelo tanque lavado	5	2.2, 2.3
	2.5	Definición modelo intercambio iónico	11	2.3
	2.6	Codificación modelo intercambio iónico	4	2.4, 2.5
Simulación proceso industrial	3.1	Implementación unidades de separación	33	1.7, 2.6
	3.2	Selección del proceso	13	-
	3.3	Simulación del proceso	17	3.1, 3.2
	3.4	Rediseño del proceso	10	3.2
	3.5	Simulación proceso rediseñado	39	3.3, 3.4

Cabe destacar que, debido a los recursos considerados a la hora de realizar la planificación, la duración del proyecto es inferior a la correspondiente a un Trabajo Final de Grado. Sin embargo, un alumno necesita 300h para realizar todas y cada una de las tareas que aparecen en la Tabla 1, reduciéndose el tiempo de realización como consecuencia de una asignación de recursos mayor.

En la Figura 2 se presenta la información de la Tabla 1 de manera gráfica en forma de diagrama CPM (Critical Path Method). En dicho diagrama se resalta en color rojo el camino crítico, el cual está compuesto por aquellas actividades cuyo retraso implica un retraso en la finalización del proyecto.

Para poder interpretar adecuadamente dicho diagrama a continuación se proporciona una leyenda, así como las expresiones que se emplean para el cálculo de las holguras:

CP	D	FP
Código de la tarea		
CT	H_T/H_L	FT

CP: comienzo más próximo

D: duración

FP: final más próximo

CT: comienzo más tardío

FT: final más tardío

Holgura total (H_T): tiempo que puede retrasarse una tarea sin retrasar la finalización del proyecto. Se obtiene mediante la siguiente expresión:

$$H_T^i = FT^i - FP^i$$

Holgura libre (H_L): tiempo que puede retrasarse una tarea sin reducir la holgura de las tareas siguientes. Se obtiene mediante la expresión:

$$H_L^i = CP^{i+1} - FP^i$$

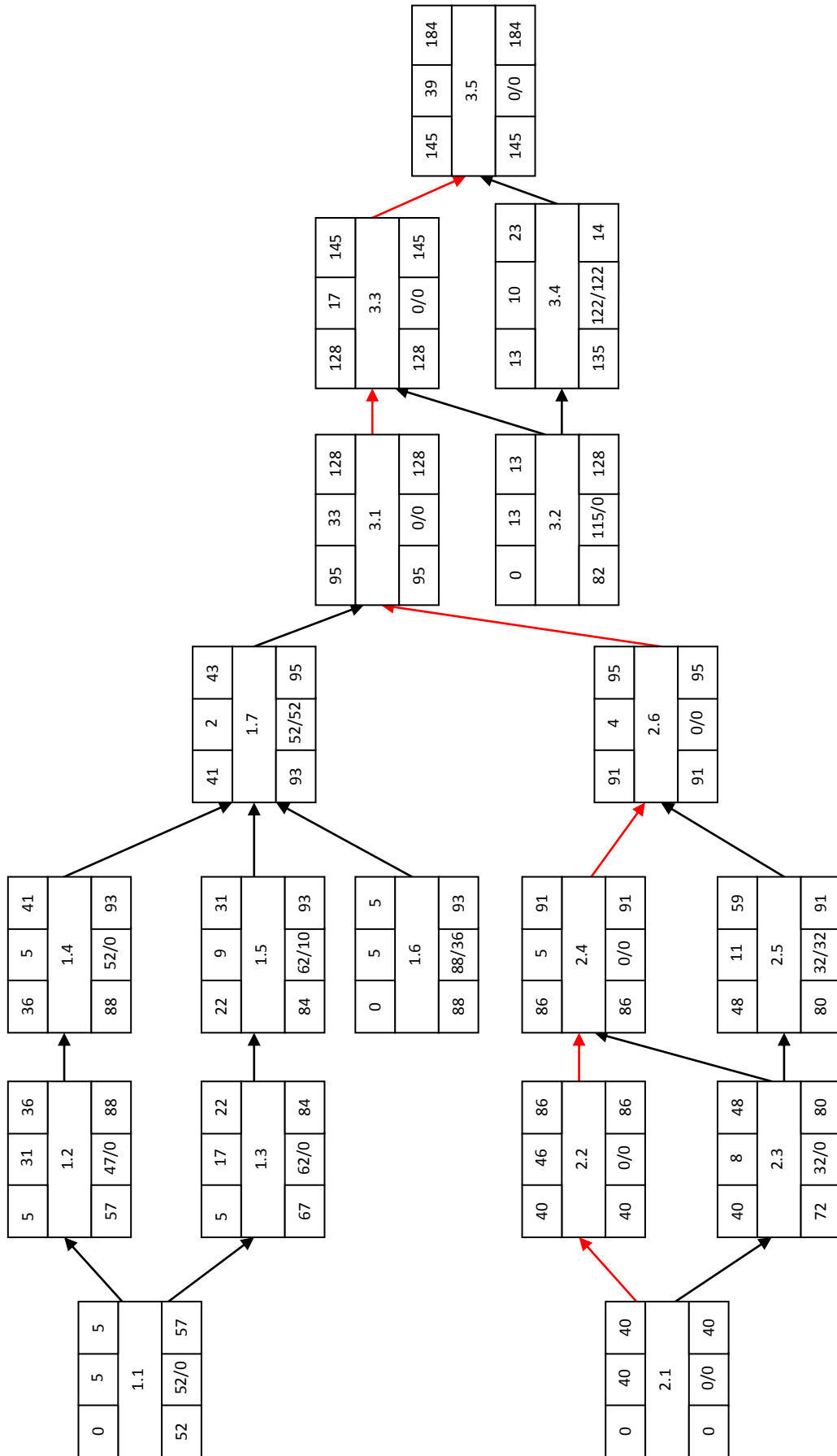


Figura 2. Diagrama CPM del proyecto

2. INTRODUCCIÓN

2.1. Motivación

Los simuladores de procesos se utilizan en una gran variedad de empresas para diseñar y analizar sus procesos productivos, puesto que permiten realizar estas tareas de una forma segura y económica. Estas herramientas permiten variar parámetros del proceso sin riesgo a que este se descontrola así como explorar diferentes modificaciones del mismo sin necesidad de realizar una inversión en un ensayo de planta piloto.

Aunque los simuladores tienen incorporada una gran variedad de modelos, en la mayoría de los casos no contemplan muchas de las unidades de proceso que se emplean en el campo de la Ingeniería Química: separación por membranas, intercambio iónico, adsorción, etc. Por el hecho de carecer de estas unidades, las cuales en la mayoría de los casos son unidades de separación, en muchas ocasiones no es posible utilizar un simulador para estudiar el proceso de una planta química.

En consecuencia, en este proyecto se pretende introducir en un simulador de procesos diferentes modelos para poder simular algunas de las operaciones de separación que habitualmente no están incluidas en estos programas. De esta forma, se busca ampliar las funcionalidades del simulador hasta conseguir una herramienta que permita simular al completo los procesos químicos, no solo teniendo en cuenta las etapas de producción sino también las operaciones de separación destinadas a la purificación y acondicionamiento de corrientes.

Debido a la creciente actividad del Departamento de Ingeniería Química y Nuclear de la Universidad Politécnica de Valencia (UPV) en el campo de la tecnología de membranas, se prestará especial atención a la introducción de unidades de separación de membrana en el simulador de procesos, las cuales no suelen estar incluidas debido a la dependencia que tienen estas unidades de las condiciones en que se operan.

Además, la separación por membranas está cada vez más presente en los procesos industriales, ya que suponen una alternativa eficaz para reducir el impacto ambiental de las plantas químicas. Un ejemplo de ello puede encontrarse en la industria metalmeccánica, donde en la zona de enjuague se utilizan membranas de ósmosis para reducir el consumo de agua, siendo pues este el proceso que se ha seleccionado para demostrar las posibilidades que ofrece el simulador una vez se hayan implementado las nuevas funcionalidades.

2.2. Antecedentes y estado actual

2.2.1. *La tecnología CAPE-OPEN*

Desde que en 1995 la asociación CO-LaN publica su documento de especificaciones CAPE-OPEN, muchos programadores han mostrado interés por ajustar el software que desarrollan a estas normas. El principal objetivo que se persigue con el estándar CAPE-OPEN es garantizar que el software de simulación de procesos posea interfaces abiertas que permitan la interoperabilidad de manera sencilla y barata ^[1].

Las herramientas de modelado de unidades de proceso (PMC: Process Modelling Component) que se ajustan a las especificaciones CAPE-OPEN pueden introducirse en cualquier entorno de simulación (PME: Process Modelling Environment) CAPE-OPEN a través de la creación de una interfaz (middleware). La implementación de estas interfaces se hace a través de tecnologías utilizadas a nivel mundial como lo son COM o CORBA por ejemplo ^[1].

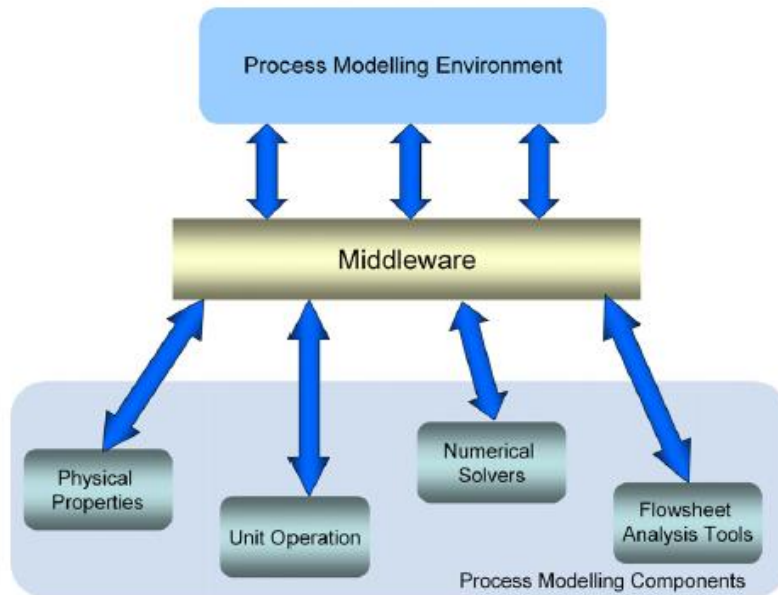


Figura 3. Conexión entre los diferentes componentes CAPE-OPEN

En el año 2008 diversos autores aprovecharon el lanzamiento del entorno .NET framework de Microsoft para implementar una conexión entre PMCs y PMEs. A través del uso de esta tecnología les fue posible introducir en un simulador de procesos tanto unidades de operación como datos de una base termodinámica ^[2]. Con ello se consiguió demostrar que el entorno .NET framework es una alternativa a los que se habían venido utilizando en el pasado (COM, CORBA).

Hasta el 2009, para poder personalizar los simuladores de procesos que cumplieran las especificaciones CAPE-OPEN los usuarios debían implementar ellos mismos un algoritmo que permitiera la conexión entre las diferentes herramientas informáticas. Sin embargo, en la reunión anual de AIChE del 2009 los desarrolladores del simulador de procesos COCO junto con los del programa ChemSep presentaron una alternativa a esto: la unidad MATLAB ^[3]. Esta unidad tiene implementada una conexión entre el entorno de simulación COCO y el programa de cálculo MATLAB, de manera que a través de su uso puede introducirse en COCO cualquier algoritmo escrito en código Matlab sin necesidad de tener los conocimientos de programación necesarios para la creación de un middleware.

A pesar de ello, al año siguiente el principal desarrollador de COCO (Jasper Van Baten) junto con otros autores describió cómo implementar el modelo de un reactor de Gibbs para que este pueda funcionar en cualquier entorno de simulación CAPE-OPEN^[4]. En este trabajo se detalla la forma en que deben implementarse las interfaces de la unidad para que esta se convierta en un PMC capaz de operar con cualquier PME, de manera que los usuarios que prefieran seguir implementando su propio middleware a pesar del lanzamiento de la unidad MATLAB pueden utilizar el texto como guía.

2.2.2. Simulación de modelos desarrollados por el usuario

Como alternativa al uso de tecnología CAPE-OPEN, los simuladores de procesos comerciales ofrecen unidades de operación que permiten introducir modelos desarrollados por el usuario en la simulación. Como ejemplo de ello puede mencionarse las unidades de que dispone CHEMCAD para introducir código Visual Basic (VBA), Fortran o C++.

La principal diferencia con la simulación a través de componentes CAPE-OPEN es que en este caso la libertad de desarrollo que tienen el usuario es menor. La implementación de los modelos debe hacerse de una manera muy concreta, por lo que la aplicación final solo es válida para ser utilizada en un entorno de simulación determinado.

En la Universidad Politécnica de Valencia se han desarrollado diferentes proyectos académicos en los que se explora esta alternativa. Cabe destacar el Proyecto Final de Carrera realizado por Beatriz Gómez en 2004, cuyo objetivo era introducir una unidad de cálculo de membranas en el simulador CHEMCAD a través del uso del lenguaje VBA para poder diseñar una planta de desalación mediante el uso del simulador.

2.2.3. Simulación de unidades de membrana

Para abordar el diseño de instalaciones de membrana, las diferentes casas comerciales ofrecen de manera gratuita sus propias aplicaciones para simular el comportamiento de sus productos. Como ejemplo de este software pueden mencionarse los siguientes programas: ROSA (Filmtec), ROPRO (Koch) e IMSDesign (Hydranautics).

Estas aplicaciones presentan ciertas limitaciones en su uso, ya que están preparadas para realizar una serie de cálculos determinada, por lo que el usuario no puede estudiar libremente los parámetros de la instalación que desee. Además, en estas herramientas solo se puede simular las membranas de la casa comercial que las ha desarrollado.

Sin embargo, el principal motivo por el cual los simuladores de procesos suponen una alternativa más interesante es que en ellos existe la posibilidad de estudiar instalaciones donde las unidades de membrana se combinen con otras unidades de proceso. Para poder realizar esta tarea con las aplicaciones de las casas comerciales habría que combinar el uso dos programas, uno para la unidad de membrana y otro para el resto de unidades, lo cual se convierte en una tarea costosa si existen recirculaciones de la unidad de membrana a otros puntos del proceso.

2.2.4. Simulación de procesos de enjuague

Como trabajo previo en el campo de la simulación de procesos de enjuague de la industria metalmeccánica, cabe destacar el artículo realizado por E. Chilyumova y J. Thöming en 2007, en el cual se realiza una simulación dinámica de un proceso de enjuague ^[5]. El proceso estudiado en este proyecto es una adaptación del que aparece en este trabajo, describiéndose a continuación las principales características de este.

El proceso consiste en un enjuague de piezas metálicas a la salida de un acabado de níquel cuyo objetivo es reducir la concentración de este metal en el agua arrastrada por la pieza. Esto se realiza a través de una serie de siete tanques de lavado donde se sumerge la pieza un determinado tiempo y a contracorriente se introduce un caudal de agua de lavado para retirar las impurezas. En la siguiente tabla se adjuntan los parámetros que caracterizan dicho sistema de lavado.

Tabla 2. Parámetros de la cascada de tanques de lavado

<i>Parámetro</i>	<i>Valor</i>
Concentración NiCl ₂ a la entrada	210 g/L
Razón de dilución	1000
Concentración NiCl ₂ a la entrada	0.21 g/L
Caudal de arrastre	16 L/h

En el proceso de enjuague, además de los propios tanques de lavado, se cuenta con una unidad de membrana, la cual tiene dos objetivos: purificar el agua de lavado para poder reutilizarla, reduciendo así el consumo de agua fresca, y generar una corriente concentrada que pueda recircularse al baño de niquelado, de manera que el impacto ambiental de la planta sea nulo.

Esta unidad de tratamiento por membranas queda definida a través de la siguiente lista de características:

- 1 tubo con 6 módulos SW30-2540 de la casa DOW. Estos módulos de tamaño pequeño se adaptan a los caudales bajos con que se trabaja en el proceso a la vez que permiten tratar las elevadas concentraciones de las corrientes.
- 3 tanques de almacenamiento. Con estos equipos se puede almacenar el agua a tratar durante 23h al día, de manera que se succiona el contenido durante la hora restante con la bomba para introducirlo en la membrana. De esta forma se asegura una velocidad de paso adecuada en el equipo (1000 L/h).
- Bomba 310S. Esta bomba permite aplicar la presión necesaria para vencer la presión osmótica y así conseguir la separación de componentes. Como los módulos son de tamaño pequeño la presión a aplicar es elevada, concretamente de 115 bares.

Adicionalmente, en el último tanque se cuenta con una unidad de intercambio iónico para purificar parte del agua de lavado que sale de dicho tanque, consiguiéndose reducir la concentración de NiCl_2 hasta 0.01 g/L a través del uso de una resina catiónica LEWATIT TP 260. Gracias al uso de esta segunda unidad de tratamiento del agua de lavado puede cumplirse con las especificaciones requeridas en la pieza de salida introduciendo en el sistema 16 L/h de agua fresca.

En la Figura 4 se presenta un esquema del proceso, en el cual puede apreciarse la configuración de los diferentes equipos que lo componen así como los parámetros de operación que definen su comportamiento.

2.2.5. Estado actual

Actualmente muchas empresas y profesionales siguen empleando la tecnología COM, CORBA y .NET framework para crear interfaces que permitan introducir sus PMC en los simuladores de procesos CAPE-OPEN. De hecho, en la página web del simulador COCO existe una sección donde se detalla todas las empresas que poseen PMC que pueden ser introducidos como accesorio en el simulador ^[6].

Además, el propio desarrollador de COCO publicó en 2011 junto con otros autores un artículo en el cual a través de la creación de una de estas interfaces se consigue implementar un algoritmo de la Agencia de Protección Medioambiental de EEUU en el simulador. Con ello se consigue añadir una funcionalidad al programa que permite calcular el impacto ambiental de cualquier proceso simulado en este entorno ^[7].

Sin embargo, algunos autores han optado por las facilidades que ofrece la unidad MATLAB, la cual permite introducir funcionalidades en el simulador COCO sin necesidad de implementar un middleware, a cambio de que dichas funcionalidades estén desarrolladas en código Matlab.

Como ejemplo de ello, D. Peshev y A. Livingston introdujeron una unidad de nanofiltración implementada en Matlab en los simuladores Hysys, Aspen Plus y COCO ^[8]. En su trabajo se demuestra que en cualquiera de los tres casos existe una coincidencia entre los resultados de la simulación y los datos de los ensayos experimentales, por lo que se concluye que cualquiera de los tres es una alternativa válida para simular la unidad implementada.

Dado que la validez de la unidad MATLAB ha quedado demostrada para la implementación de unidades de nanofiltración en COCO ^[8], parece razonable utilizar esta herramienta para implementar otras unidades de separación, ya que de esta forma se ahorra en tiempo de implementación y en conocimientos requeridos para el desarrollo de la aplicación.

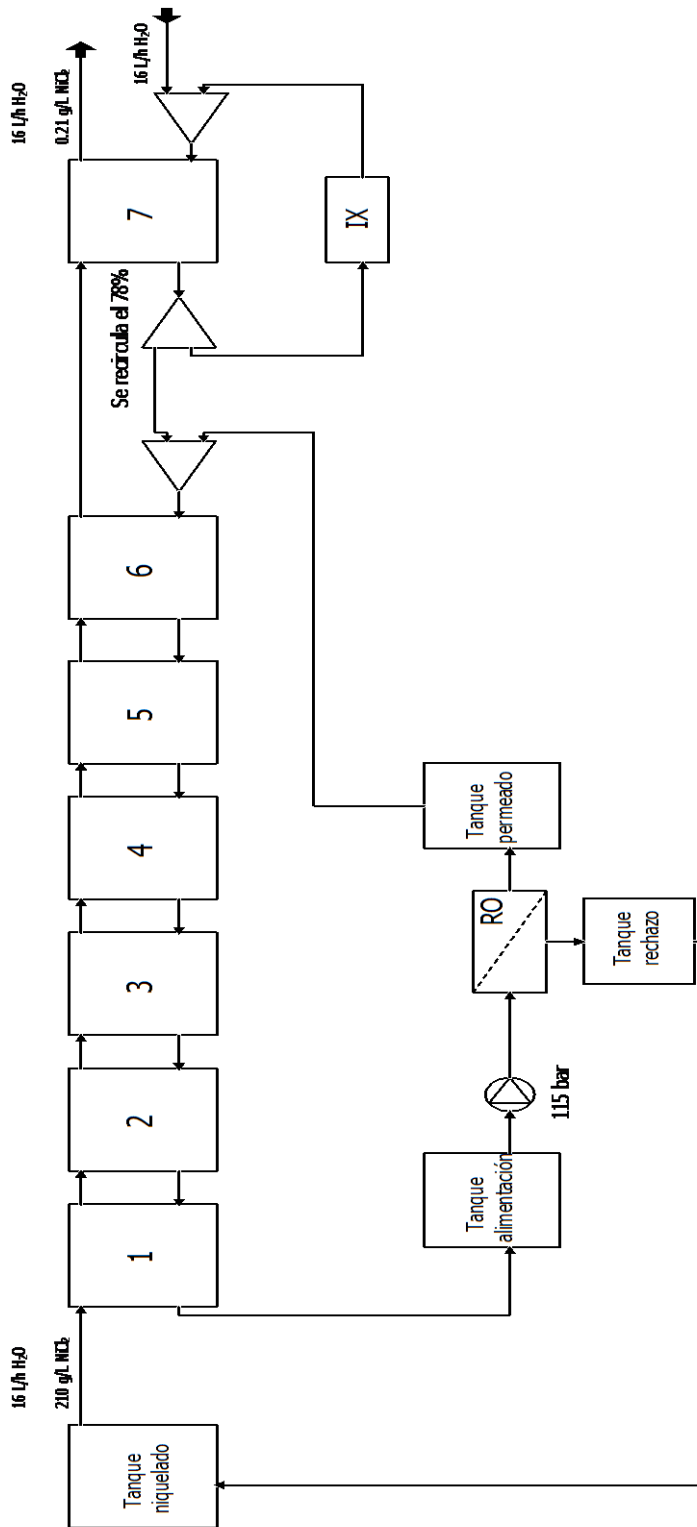


Figura 4. Esquema del proceso de regeneración de aguas industriales estudiado

2.3. Justificación técnica

A través del uso de la unidad MATLAB pueden introducirse diferentes operaciones de separación en el simulador COCO, de manera que se obtiene una herramienta flexible y versátil para analizar procesos químicos. La mayor flexibilidad en el desarrollo del entorno CAPE-OPEN permite implementar una mayor variedad de modelos que en los simuladores comerciales, por lo que con COCO pueden llegarse a simular procesos de muy diversa naturaleza.

Además, pese a que ello no es parte del objeto del proyecto, el trabajo desarrollado es totalmente compatible con cualquier otra aplicación que cumpla las especificaciones CAPE-OPEN. Esto abre posibilidades no disponibles en otros simuladores, como por ejemplo la introducción de algoritmos de optimización en la simulación para determinar las condiciones de operación óptimas.

Otra ventaja de la tecnología utilizada es que es totalmente gratuita, por lo que permite sustituir los ensayos de planta piloto sin necesidad de invertir en una licencia de un software comercial. De esta forma pueden encontrarse mejoras en el diseño de los procesos sin necesidad de realizar una inversión.

Finalmente, con respecto a la utilización del software de las casas de membranas, si bien el programa desarrollado no conlleva un beneficio económico, ya que ambas opciones son gratuitas, este permite realizar análisis de procesos con recirculaciones entre distintas unidades de operación, lo cual permite diseñar instalaciones de membranas que no podrían estudiarse de manera sencilla utilizando las aplicaciones de las casas de membranas.

2.4. Justificación académica

Dado que el proyecto desarrollado se realiza para la obtención del título Grado en Ingeniería Química, este tiene un claro carácter académico. Durante la realización del proyecto se ponen en práctica los conocimientos adquiridos durante la titulación, no solo en el campo de la simulación, sino también en otros como por ejemplo la transferencia de materia o la elaboración de proyectos de ingeniería.

Así mismo, el trabajo realizado consigue demostrar que a nivel académico el simulador COCO es una alternativa para impartir las prácticas de simulación.

En la misma línea, la mayor flexibilidad del entorno CAPE-OPEN abre nuevas posibilidades para la realización de futuros proyectos académicos, en los cuales podría combinarse el trabajo realizado en este proyecto con otras funcionalidades añadidas en el simulador por otro estudiante o personal docente.

3. ANÁLISIS DE ALTERNATIVAS DE SIMULADORES

3.1. Descripción de las alternativas existentes de simuladores de procesos químicos

En el mercado existe una amplia oferta de simuladores de procesos con los que podría abordarse la resolución de este proyecto. En este caso no solo se ha tenido en cuenta los simuladores comerciales ofrecidos por distintas compañías, sino también los simuladores de uso libre desarrollados por diferentes empresas o asociaciones.

Los simuladores comerciales suponen una potente herramienta a la hora de diseñar procesos de ingeniería química dado que tienen implementadas por defecto muchas funcionalidades que se adaptan a las necesidades del usuario, facilitándole así la definición del proceso objeto de estudio. Por otro lado, estos simuladores pertenecen a compañías que dedican sus recursos al desarrollo de dichas funcionalidades y por tanto para poder utilizar estas herramientas es necesario adquirir licencias de uso con un alto coste asociado.

De entre todos los simuladores comerciales, se ha elegido el simulador CHEMCAD de la empresa *Chemstations* por ser el simulador utilizado en el Departamento de Ingeniería Química y Nuclear de la Universidad Politécnica de Valencia así como por ser el simulador comercial con que el alumno ha trabajado previamente en las asignaturas de Grado en Ingeniería Química.

En cuanto a los simuladores de uso libre, en la mayoría de los casos estos son pequeñas aplicaciones cuyo uso queda restringido a cálculos sencillos de Ingeniería Química, como por ejemplo diseño básico de columnas de destilación. Sin embargo, existen algunos simuladores de uso libre que pretenden ser aproximaciones a los simuladores comerciales, proporcionando al usuario un entorno de simulación similar al que puede encontrarse en dichos programas comerciales. Algunos ejemplos de simuladores de uso libre que pueden suponer una alternativa real a los simuladores comerciales son *COCO simulator* y *DWSIM* entre otros.

De entre todos los simuladores de uso libre que proporcionan un entorno de simulación adecuado para la resolución del proyecto, se ha establecido que *COCO simulator* es el que representa una mejor alternativa, por el hecho de ser el simulador con mayores prestaciones y mayor facilidad de ampliación por parte del usuario.

En consecuencia, a lo largo de este apartado se describirán con detalle las características de los simuladores CHEMCAD y *COCO simulator* con el objetivo de poder valorar cuál de ellos representa una mejor alternativa para realizar el proyecto.

3.1.1. CHEMCAD

El simulador de procesos CHEMCAD cuenta con una extensa base de componentes, formada por 2277 sustancias puras ^[9], entre las cuales se encuentran los compuestos iónicos que normalmente están involucrados en los procesos de separación por membrana. Además, el usuario puede ampliar esta base añadiendo componentes nuevos o introduciendo datos termodinámicos experimentales (BIPs) para calcular con ellos los coeficientes de los modelos termodinámicos.

Así mismo, el simulador cuenta con un conjunto de 50 ecuaciones termodinámicas agrupadas en un conjunto de 12 modelos ^[9] para poder determinar las propiedades termodinámicas tanto de sustancias puras como de mezclas. De hecho el simulador cuenta con un módulo independiente (CC-FLASH) con funciones específicas para el cálculo de entalpías, volúmenes específicos, etc. Una de las grandes ventajas de utilizar este simulador es que el usuario no necesita seleccionar el modelo termodinámico a utilizar sino que el propio programa elige el modelo más adecuado en función de la naturaleza de los componentes que intervienen en el proceso y las condiciones de operación.

CHEMCAD posee 56 unidades de proceso diferentes ^[9], las cuales permiten definir de forma rápida y sencilla una gran variedad de procesos químicos. Es por este motivo que este software se utiliza en empresas tan distintas del sector químico como lo son las petroquímicas y las empresas farmacéuticas.

A pesar de la gran diversidad de unidades que ofrece el simulador, entre ellas no se incluyen las unidades de separación por ósmosis inversa. Esto se debe a la complejidad de definir estos procesos como una unidad genérica, ya que el comportamiento de las membranas se ve muy afectado por las características de cada proceso particular. Sin embargo, el simulador ofrece la posibilidad al usuario de implementar sus propias unidades en el mismo, lo cual puede hacerse a través de código Visual Basic (VBA), Fortran o C++.

El código VBA es útil para implementar de forma rápida unidades de proceso sencillas, pero cuando se desea implementar modelos de cierta complejidad de forma rigurosa, como es el caso de las unidades de ósmosis inversa, resulta más interesante el uso de código C++, que es el mismo en que están desarrolladas las propias unidades del simulador.

En consecuencia, para poder implementar el modelo de una unidad de ósmosis inversa en este simulador el usuario no solo debe estar familiarizado con el manejo del mismo sino que además debe tener un buen conocimiento del lenguaje C++, especialmente en cuanto a programación con objetos se refiere. Además, a parte de la propia unidad, es necesario desarrollar una interfaz para crear el cuadro de diálogo que permite ajustar los parámetros de operación de la misma una vez introducida esta en la simulación.

En resumen, podría decirse que CHEMCAD es una herramienta muy flexible que permite definir una gran parte de los procesos químicos que se pueden encontrar a nivel industrial. Sin embargo, la ampliación de las funcionalidades del simulador por parte del usuario no resulta una tarea sencilla, puesto que las unidades introducidas deben tener el mismo rigor que las pertenecientes al simulador, de manera que se requieren amplios conocimientos en el campo de la programación para poder desarrollar el código de la unidad al nivel exigido por el programa.

3.1.2. *COCO simulator*

El simulador de procesos COCO es un simulador de uso libre, lo que significa que su utilización no requiere de pago alguno, incluso aunque se le dé fines comerciales al programa. Dado que la empresa desarrolladora del software no recibe remuneración alguna por el mismo, sus funcionalidades son más limitadas que las de CHEMCAD, puesto que no se puede destinar tantos recursos al desarrollo de las mismas. Es por este motivo que COCO solo dispone de una base de datos de unas 430 sustancias puras, 7 modelos termodinámicos y alrededor de 30 unidades de operación ^[6].

Sin embargo, el simulador está totalmente orientado a la ampliación de sus funcionalidades por parte del usuario, de forma que este pueda adaptar el software de manera sencilla y rápida, pero rigurosa, a sus necesidades particulares. De hecho, este simulador se incluye entre la lista de programas que cumple con las normas CAPE-OPEN, cuyo objetivo es garantizar la máxima interoperabilidad entre distintos softwares de ingeniería de una forma barata, rápida y sencilla para el usuario.

Este simulador de procesos ofrece la posibilidad de introducir componentes de manera sencilla a través del programa PCD manager de ChemSep. Dicho software ofrece diferentes ventanas al usuario para definir paso a paso tantas características como se desee del componente, de manera que puede definirse un componente solo con sus propiedades más básicas o bien definirlo de forma completamente rigurosa, incluso mediante datos experimentales, asignándole un modelo termodinámico con sus coeficientes correspondientes.

Otra de las ventajas de este programa, quizás la más importante, es la posibilidad que ofrece de realizar una conexión con MATLAB o Scilab para definir una unidad de operación personalizada. De esta manera el usuario puede definir unidades que no se encuentren por defecto en el simulador mediante una codificación más sencilla que en el caso de C++, pero con la potencia de cálculo y funciones matemáticas propias de un lenguaje de alto nivel.

En pocas palabras, el simulador COCO proporciona gratuitamente un entorno básico de simulación, el cual debe seguir siendo desarrollado por el propio usuario hasta conseguir las funcionalidades que requiera. Esta tarea puede acometerse de manera muy sencilla a través del uso del programa MATLAB, con el cual están familiarizados la mayoría de los profesionales de ingeniería y que permite realizar un desarrollo mucho más rápido que en C++, si bien la velocidad de cálculo es algo menor.

Por lo tanto, podría decirse que en lugar de emplearse recursos en la adquisición de una licencia de uso, en este caso se emplean recursos humanos en la ampliación del simulador, tarea que resulta ser mucho más extensa que en el caso de CHEMCAD al poseer el simulador libre muchas menos prestaciones que este.

3.2. Selección del simulador de procesos

Para determinar cuál de las dos alternativas es la más adecuada para el estudio del tipo de proceso analizado en este proyecto se aplicará el método de los factores ponderados, en el cual se valora cada una de las alternativas en base a unos criterios de decisión a los cuales se les asigna diferentes grados de importancia.

A continuación se detallan los diferentes criterios de decisión en base a los cuales se valorará cada uno de los simuladores:

1. Base de componentes y modelos termodinámicos: resulta interesante que el simulador cuente con el máximo número de sustancias puras en su base de datos así como de los modelos termodinámicos necesarios para el cálculo de sus propiedades. De especial interés es la existencia del modelo de electrolito, puesto que resulta ser el más adecuado para el cálculo de las propiedades de los compuestos iónicos.
2. Unidades de operación: cuantas más unidades de operación propias posea el simulador menor es la probabilidad de que el usuario necesite definir una unidad de proceso propia, por lo que se valorará positivamente el número de unidades de operación disponibles en cada simulador.
3. Coste: la coyuntura económica restringe fuertemente el coste que pueden alcanzar los proyectos, puesto que los recursos económicos de las empresas están muy limitados. Por este motivo, se primará aquellos programas que representen un menor coste para el proyecto.
4. Unidades personalizadas: es muy probable que se requiera implementar al menos una unidad personalizada en el simulador, ya que prácticamente ninguno de ellos tiene incluidas las unidades de membrana por defecto. En consecuencia, se valora que el programa ofrezca la posibilidad de ampliar sus unidades de proceso de una manera sencilla a la vez que precisa.

De todos los criterios de decisión descritos anteriormente, aquel con una mayor importancia es el último, ya que el hecho de poder introducir fácilmente unidades de operación en el simulador reduce considerablemente el tiempo dedicado a la implementación de las mismas, reduciendo por tanto el coste de la mano de obra de manera muy importante, que suele ser la mayor partida en este tipo de proyectos.

A continuación en importancia está el primer criterio, ya que el hecho de tener los componentes del proceso introducidos en el simulador ahorra tiempo en la implementación del mismo. Sin embargo, la definición de estos componentes solo requiere de datos que pueden encontrarse fácilmente en tablas de propiedades termodinámicas, por lo que la importancia de este criterio es mucho menor que la del último, ya que el ahorro de tiempo que puede conseguirse es significativamente menor.

Por último en cuanto a importancia están el tercer y segundo criterios respectivamente. El criterio sobre las unidades de operación tiene la menor importancia porque es muy probable que el simulador solo se utilice para cálculos sencillos como por ejemplo mezcladores, de manera que aunque el simulador posea muchas unidades de proceso es muy posible que estas no acaben necesitándose, ya que la mayoría de las unidades que aparecen en los procesos de regeneración de aguas industriales no están incluidas en los simuladores.

Esta razón refuerza aún más la decisión de asignar al último criterio una importancia muy superior al resto, dado que con toda seguridad se necesitará implementar más de una unidad de proceso en el simulador.

En base a todos estos motivos, sobre una base de 1, se ha decidido asignar los valores de importancia que aparecen en la siguiente tabla a cada uno de los criterios.

Tabla 3. Importancia asignada a cada uno de los criterios de decisión

<i>Criterio</i>	<i>Importancia</i>
Base componentes y modelos termodinámicos	0.250
Unidades de operación	0.058
Coste	0.142
Unidades personalizadas	0.550

Una vez establecidos los diferentes criterios de decisión y asignado un nivel de importancia a cada uno de ellos puede realizarse la valoración de cada una de las alternativas y así determinar cuál de los dos simuladores representa una mejor opción para su uso en el proyecto, a través de la aplicación de la fórmula de la suma ponderada.

Tabla 4. Valoración de cada una de las alternativas

<i>Criterio</i>	<i>CHEMCAD</i>	<i>COCO</i>
Base componentes y modelos termodinámicos	8	1
Unidades de operación	9	7
Coste	3	10
Unidades personalizadas	6	9

$$CHEMCAD = 0.25 \cdot 8 + 0.058 \cdot 9 + 0.142 \cdot 3 + 0.55 \cdot 6 = 6.248$$

$$COCO = 0.25 \cdot 1 + 0.058 \cdot 7 + 0.142 \cdot 10 + 0.55 \cdot 9 = 7.026$$

A partir de los resultados obtenidos, puede afirmarse que el simulador COCO es la mejor de las alternativas planteadas y por tanto será el que se utilice en el desarrollo del proyecto. En este caso la mayor facilidad de personalización de COCO ha primado sobre las mayores prestaciones de CHEMCAD, dado que para simular un proceso de regeneración de aguas industriales se necesitará introducir varias funcionalidades en cualquiera de los dos simuladores.

El principal motivo de la diferencia entre ambos simuladores es la unidad MATLAB de que dispone COCO. A lo largo del proyecto se deberán desarrollar diferentes modelos de unidades de separación y la posibilidad de poder realizarlos en código Matlab reduce enormemente la complejidad de la tarea en comparación con desarrollarlos en C++.

Por lo tanto, podría decirse que a pesar de tener prestaciones muy inferiores a CHEMCAD, las funcionalidades de COCO resultan ser más útiles para abordar la realización de las tareas de este proyecto.

4. DESCRIPCIÓN DE LOS MODELOS FÍSICOS UTILIZADOS

4.1. Modelo para una etapa de membrana

Las ecuaciones matemáticas que permiten definir el comportamiento de una unidad de ósmosis inversa pueden obtenerse a través de la aplicación de balances de máximo gradiente al sistema. A continuación se presenta el conjunto de ecuaciones que representa dicho comportamiento así como el desarrollo realizado para obtenerlas ^[10].

$$\text{Balance total: } \frac{dv}{dz} = -J_v \cdot \frac{dA}{dV} \quad (1)$$

Dado que la variable de estado que se utiliza es el caudal, es necesario expresar la derivada de la Ec. 1 en función de dicha variable.

$$\frac{dv}{dz} = \frac{d(Q/S_{mod})}{dz} = \frac{dQ}{dz} \cdot \frac{1}{S_{mod}}$$

$$\boxed{\frac{dQ}{dz} = -J_v \cdot S_{mod} \cdot \frac{dA}{dV}} \quad (2)$$

$$\text{Balance de componente: } \frac{d(v \cdot C)}{dz} = -J_s \cdot \frac{dA}{dV} \quad (3)$$

$$\frac{dv}{dz} \cdot C + \frac{dC}{dz} \cdot v = -J_s \cdot \frac{dA}{dV} \quad (4)$$

Combinando la expresión de la Ec. 4 con la de la Ec. 1 se llega a la siguiente ecuación diferencial:

$$-J_v \cdot \frac{dA}{dV} \cdot C + \frac{dC}{dz} \cdot v = -J_s \cdot \frac{dA}{dV}$$

$$\frac{dC}{dz} = \frac{J_v \cdot C - J_s}{v} \cdot \frac{dA}{dV}$$

$$\boxed{\frac{dC}{dz} = \frac{J_v \cdot C - J_s}{Q/S_{mod}} \cdot \frac{dA}{dV}} \quad (5)$$

Además de las Ec. 2 y 5, para definir la variación de las propiedades de la corriente a lo largo de la longitud de la unidad puede introducirse una ecuación más que considere las pérdidas de carga en la unidad, de forma que el sistema de ecuaciones diferenciales que representa el comportamiento de la unidad es el siguiente:

$$\begin{cases} \frac{dQ}{dz} = -J_v \cdot S_{mod} \cdot \frac{dA}{dV} \\ \frac{dC}{dz} = \frac{J_v \cdot C - J_s}{Q/S_{mod}} \cdot \frac{dA}{dV} \\ \frac{dP}{dz} = -k_{perd} \cdot \left(\frac{Q}{S_{mod}}\right)^2 \end{cases}$$

El sistema de ecuaciones diferenciales obtenido permite definir las características de la corriente de rechazo Q_r , C_r y P_r , de manera que a través de la aplicación de balances globalizados pueden obtenerse las características de la corriente de permeado.

$$\text{Balance total: } Q_a = Q_r + Q_p \quad (6)$$

$$Q_p = Q_a - Q_r \quad (7)$$

$$\text{Balance de componente: } Q_a \cdot C_a = Q_r \cdot C_r + Q_p \cdot C_p \quad (8)$$

$$C_p = \frac{Q_a \cdot C_a - Q_r \cdot C_r}{Q_p} \quad (9)$$

En cuanto a la presión de la corriente de permeado, este valor es un dato de entrada que debe especificar el usuario, aunque por defecto se establece un valor de 1 atm, ya que en la mayoría de las situaciones el permeado sale de la unidad a una presión igual a la atmosférica.

4.1.1. Cálculo de los fluxes mediante el modelo de disolución-difusión

Uno de los modelos que más comúnmente se emplea para calcular los fluxes J_v y J_s que aparecen en las ecuaciones diferenciales que definen el comportamiento de una membrana de ósmosis inversa es el modelo de disolución-difusión. En este modelo se considera que los componentes que atraviesan la barrera semipermeable lo hacen disolviéndose primero en ella y después difundiendo a través suyo.

En consecuencia, en el modelo de disolución-difusión se plantean expresiones proporcionales a una fuerza impulsora, que en el caso del flux J_v es la presión y en el caso del flux J_s , la concentración.

$$J_v = L_p \cdot (\Delta P - \Delta\pi) \quad (10)$$

$$J_s = B \cdot (C_a - C_p) \quad (11)$$

En la Ec. 10 el término $\Delta\pi$ representa la presión osmótica de la disolución, que es la mínima presión que debe aplicarse para que se tenga un flujo a través de la membrana. Este valor puede relacionarse con las concentraciones de la alimentación y del permeado a través de la ecuación de Van't Hoff:

$$\Delta\pi = (C_a - C_p) \cdot R_G \cdot T \quad (12)$$

En la Ec. 12 C_a y C_p hacen referencia a la concentración del ión que será separado mediante la membrana, pero sin embargo en el modelo informático se pasa como parámetro de entrada la concentración del compuesto disuelto expresada en kg/m^3 . Por lo tanto, para poder aplicar la ecuación de Van't Hoff en el modelo es necesario introducir un factor de conversión, en el cual se incluye además el término constante $R_G \cdot T$ y que se denomina factor osmótico (α).

$$\alpha = \frac{\text{número iones en el compuesto}}{\text{Peso molecular}} \cdot R_G \cdot T \cdot 10^{-2} \quad (13)$$

$$\Delta\pi = \alpha \cdot (C_a - C_p) \quad (14)$$

Dado que ambos fluxes están pues relacionados con la diferencia de concentraciones entre el alimento y el permeado, para poder calcularlos basta con obtener la concentración que se tendrá en la corriente de permeado. Esto puede hacerse a través del uso de una tercera expresión, según la cual la concentración en el permeado es el cociente entre el flux de soluto y el flux volumétrico.

$$C_p = \frac{J_s}{J_v} = \frac{B \cdot (C_a - C_p)}{L_p \cdot [\Delta P - \alpha \cdot (C_a - C_p)]} \quad (14)$$

Despejando C_p de la Ec. 14:

$$C_p = \frac{1}{2\alpha} \cdot \left[\alpha C_a - \Delta P - \frac{B}{L_p} + \sqrt{\left(\alpha C_a - \Delta P - \frac{B}{L_p} \right)^2 + 4 \cdot \frac{B}{L_p} \cdot \alpha C_a} \right] \quad (15)$$

Por lo tanto, para poder obtener los fluxes J_v y J_s se debe aplicar la Ec. 15 para obtener la concentración en el permeado a partir de las características del alimento y los parámetros de la membrana, calcular la presión osmótica mediante la Ec. 14 y finalmente aplicar las Ec. 10 y 11 para obtener los fluxes.

Es importante remarcar que en el desarrollo para deducir expresiones que permitan obtener los fluxes han aparecido otras expresiones para la concentración de permeado. El valor que se obtiene al aplicar dichas expresiones es análogo al obtenido aplicando el balance globalizado, habiéndose preferido utilizar el balance globalizado para obtenerlo en la implementación informática del modelo.

4.1.2. Efecto de la concentración por polarización

Debido a que las moléculas de disolvente atraviesan con mayor facilidad la barrera semipermeable que las moléculas de soluto, a lo largo del espesor de la membrana se produce un aumento de la concentración. Esto provoca que se genere un flujo difusivo de sentido contrario al flujo convectivo que arrastra las moléculas de soluto hacia la pared de la membrana, de manera que al final se alcanza una situación en la que la concentración en la pared de la membrana no es igual a la de la disolución alimento, conociéndose este efecto como concentración por polarización.

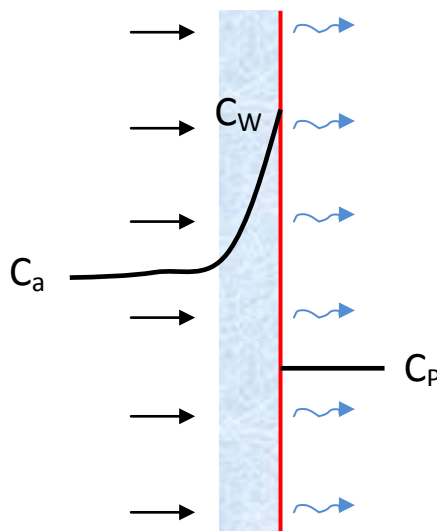


Figura 5. Evolución del perfil de concentración a lo largo de la capa límite de polarización ^[11]

Al tener en cuenta la concentración por polarización las ecuaciones del modelo de disolución-difusión se ven modificadas, puesto que la fuerza impulsora pasa a ser la diferencia entre la concentración en la pared y la concentración en el permeado. Las ecuaciones que permiten calcular los fluxes en este caso son ^[12]:

$$J_v = L_p \cdot (\Delta P - \Delta \pi) \quad (10) \quad \Delta \pi = b_\pi \cdot (C_w - C_p) \quad (15)$$

$$J_v = k_s \cdot \ln \frac{C_w - C_p}{C_a - C_p} \quad (16)$$

$$J_s = B \cdot (C_w - C_p) \quad (17)$$

$$C_p = \frac{J_s}{J_v} = \frac{B \cdot (C_a - C_p)}{k_s \cdot \ln \frac{C_w - C_p}{C_a - C_p}} \quad (18)$$

A pesar de que se tiene una nueva incógnita en el sistema (C_w), también se tiene una ecuación adicional para el cálculo del flux volumétrico, con lo cual es posible resolver el sistema formado por las Ec. 10-15-16-17-18. El problema es que dicho sistema no puede resolverse analíticamente sino que su resolución debe hacerse de manera iterativa siguiendo este proceso:

- Suposición de un valor para la concentración en la pared C_{w_sup} .
- Cálculo de C_p mediante el uso de la Ec. 18.
- Cálculo de J_v mediante el uso de las Ec. 10 y 15.
- Cálculo de la concentración en la pared C_{w_cal} con la Ec. 16.
- Comparación de los valores C_{w_sup} y C_{w_cal} . Si no coinciden volver al primer paso tomando C_{w_cal} como valor supuesto y si coinciden continuar con el procedimiento.
- Cálculo de J_s mediante la Ec. 17

En cuanto a los nuevos coeficientes que aparecen en las expresiones del modelo (k_s y b_π), las expresiones necesarias para su cálculo se detallan a continuación (aplicación secuencial para la obtención de los coeficientes) ^[12]:

$$v = 0.0032 + 3 \cdot 10^{-6} \cdot C_a + 4 \cdot 10^{-9} \cdot C_a^2 \quad (19)$$

$$D_{AB} = 6.725 \cdot 10^{-6} \cdot \exp\left(0.00212 \cdot C_a - \frac{1965}{273.15 + T}\right) \quad (20)$$

$$Sc = \frac{v}{D_{AB}} \quad (21)$$

$$Re = \frac{d_h \cdot (Q/S_{mod})}{\nu} \quad \text{con } d_h = 0.9 \cdot 10^{-3} \quad (22)$$

$$Sh = 0.065 \cdot Re^{0.865} \cdot Sc^{0.25} \quad (23)$$

$$Sh = \frac{k_s \cdot d_h}{D_{AB}} \quad (24)$$

$$\pi = 0.7949 \cdot C_a - 0.0021 \cdot C_a^2 + 7 \cdot 10^{-5} \cdot C_a^3 - 6 \cdot 10^{-7} \cdot C_a^3 \quad (25)$$

$$b_\pi = \frac{\pi}{C_a} \quad (26)$$

4.2. Modelo para un tanque de lavado

Para modelar el tanque de lavado este se considera un tanque de mezcla perfecta, es decir, las propiedades de las corrientes de salida serán las mismas que las del propio tanque. Esta hipótesis no introduce demasiado error en el modelo, puesto que los tanques cuentan con sistemas de aireación que garantizan una mezcla homogénea del contenido de los mismos.

En cuanto a las entradas y salidas del sistema, se considera que entra una corriente de agua de lavado con un caudal Q y concentración de impureza C_L y una corriente con un caudal q y concentración C_0 que representa el caudal de arrastre asociado a la pieza metálica. Respecto a las salidas, se tendrán dos corrientes con los mismos caudales que los de entrada pero con una concentración C igual a la que se alcance en el tanque.

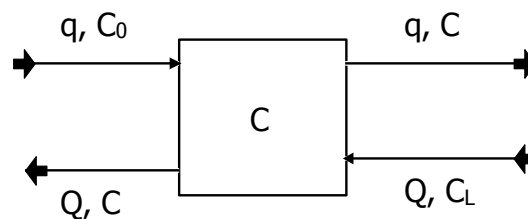


Figura 6. Esquema de un tanque de lavado

La concentración que se alcanza en el tanque, que es la única incógnita del sistema, puede obtenerse a partir de las propiedades de las corrientes de entrada a través de la aplicación de un balance globalizado:

$$q \cdot C_0 + Q \cdot C_L = q \cdot C + Q \cdot C$$

$$C = \frac{q \cdot C_0 + Q \cdot C_L}{q + Q} \quad (27)$$

4.3. Modelo para un equipo de intercambio iónico

El intercambio iónico es una unidad de separación en la que se establece un equilibrio entre el ión a retirar en el líquido y el grupo funcional de la resina. A partir de dicho equilibrio puede obtenerse la concentración del ión a retirar en el líquido a la salida del equipo (C_s), de manera que la corriente de salida tiene el mismo caudal que la de entrada pero con la concentración correspondiente a dicho equilibrio.

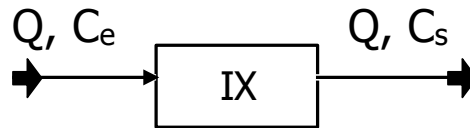
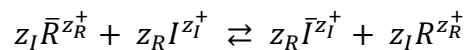


Figura 7. Esquema de un intercambio iónico

Según la teoría de membrana de Donan, el equilibrio que se establece en un intercambio iónico puede definirse mediante la siguiente expresión ^[13]:



Dado que en este tipo de unidades los coeficientes de actividad son difíciles de determinar y normalmente las disoluciones con que se trabaja están próximas a la idealidad, se considera que dichos coeficientes son la unidad ^[13], de manera que la expresión de la constante del equilibrio anterior queda como:

$$(K_C)_R^I = \frac{\bar{C}_I^{z_R} \cdot C_R^{z_I}}{\bar{C}_R^{z_I} \cdot C_I^{z_R}} \quad (28)$$

Por lo tanto, para el caso del proceso a simular $z_A=2$ y $z_B=1$, la expresión que define el equilibrio que se alcanza en el intercambio iónico resulta ser:

$$(K_C)_R^I \cdot \left(\frac{\bar{C}}{C}\right) = \frac{y_I \cdot (1 - x_I)^2}{x_I \cdot (1 - y_I)^2} \quad (29)$$

En la Ec. 29 se aprecia que el equilibrio está influido por la fracción iónica equivalente del ión a retirar en la resina. Esto tiene como consecuencia que la concentración de salida del líquido aumente a lo largo del tiempo, puesto que a medida que se produce el intercambio el valor de x_A se va modificando al depositarse el ión del líquido sobre la resina.

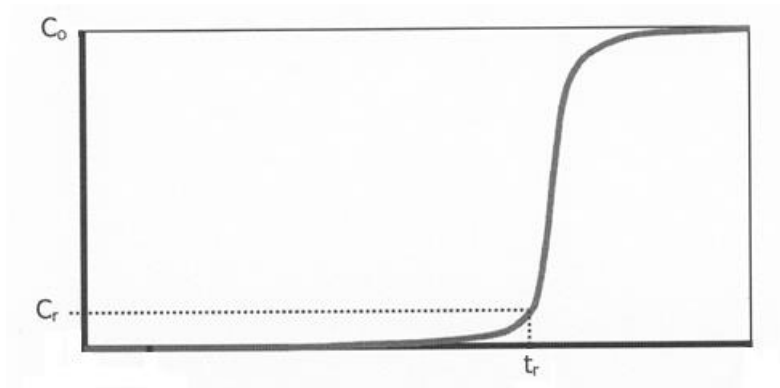


Figura 8. Evolución de la concentración a la salida de un intercambio iónico

En la operación de estos equipos se establece un límite máximo para la concentración de salida, llamado concentración de ruptura (C_r), a partir del cual se detiene la operación con el lecho y se pasa a regenerarlo. Por lo tanto, además de ser un equipo no estacionario, el intercambio iónico es una unidad semicontinua, en la cual se alternan fases de operación y de regeneración.

Además de un cambio en la concentración de la corriente, en un equipo de intercambio iónico se produce un cambio en la presión de la corriente de salida debido a las pérdidas que sufre el fluido al atravesar el lecho. Para el caso de un lecho fijo dichas pérdidas pueden calcularse a través de la siguiente expresión ^[13]:

$$\frac{\Delta P}{L} = \frac{k_1 \cdot S^2 \cdot \mu \cdot (1-\varepsilon)^2}{\varepsilon^3} \cdot U \quad (30)$$

Los simuladores de procesos únicamente permiten simular procesos continuos, por lo que la alternancia entre fase de operación y fase de regeneración no puede incluirse en el modelo de la unidad. Así mismo, el simulador COCO no dispone de la posibilidad de realizar simulaciones dependientes del tiempo, de manera que tampoco puede introducirse en el comportamiento de la unidad el aumento de la concentración en la corriente de salida.

En consecuencia, el modelo que representa el comportamiento de la unidad únicamente consta de un parámetro de entrada donde se establece un valor fijo para la concentración del líquido a la salida del equipo, no habiendo podido implementarse las expresiones del equilibrio del proceso debido a las limitaciones que presenta el simulador.

La Ec. 30 tampoco se ha podido introducir en el modelo de la unidad, dado que en ella aparecen parámetros que dependen de la geometría del equipo, la cual es desconocida. Por lo tanto, se ha considerado que las pérdidas de carga en el equipo son despreciables.

5. IMPLEMENTACIÓN DE FUNCIONALIDADES ADICIONALES EN COCO

5.1. La unidad MATLAB en COCO

El simulador COCO dispone de unidades de usuario que permiten introducir en el simulador unidades de proceso personalizadas. A través del uso de dichas unidades es posible implementar en el simulador unidades de ósmosis inversa por ejemplo. En el simulador COCO se dispone de tres tipos diferentes de unidad de usuario: unidad Excel, unidad MATLAB y unidad Scilab.

Básicamente las unidades de usuario están compuestas por una serie de órdenes, escritas en forma de código informático, que se ejecutan cada vez que el simulador realiza un cálculo de la unidad. La diferencia que existe entre los diferentes tipos de unidad de usuario es el lenguaje en que se escribe dicho código informático. En este proyecto se ha decidido trabajar con la unidad MATLAB, puesto que utiliza un lenguaje de alto nivel cuyo uso está muy extendido.

Antes de comenzar con la escritura del código, para implementar una unidad MATLAB deben definirse los puertos que tendrá dicha unidad, es decir, deben indicarse cuáles serán las corrientes de entrada y salida de la unidad. Esto se hace a través de la pestaña Ports del menú de configuración de la unidad, donde pueden añadirse corrientes de entrada y salida a las cuales se les asigna automáticamente un índice.

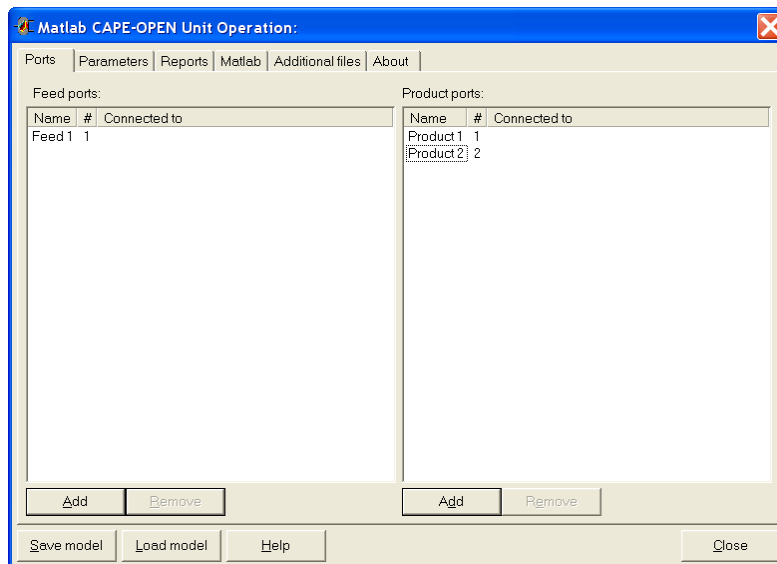


Figura 9. Pestaña Ports del menú de configuración de la unidad MATLAB

Es importante destacar que las corrientes de entrada y salida son reconocidas por el simulador a través del índice que les es asignado, de manera que el nombre de la corriente solo sirve para facilitar al usuario la identificación de cada una de las corrientes pero a efectos de cálculo no tiene ninguna relevancia.

Una vez definidos todos los puertos de la unidad, es necesario indicar los parámetros que la caracterizan, los cuales pueden ser de entrada o de salida. Los parámetros de entrada sirven para definir las características del equipo, como por ejemplo sus dimensiones, mientras que los de salida sirven para indicar el resultado de algún cálculo de la unidad, como puede ser el calor que se necesita aportar a la misma. Tanto los parámetros de entrada como los de salida deben definirse también previamente a la escritura del código, ya que ambos serán utilizados en él y por tanto sus variables deben estar creadas previamente a la escritura de este.

La definición de los parámetros de la unidad MATLAB se hace a través de la pestaña Parameters, donde se permite introducir parámetros debiéndose indicar su nombre, el tipo de variable, si son de salida o de entrada, el valor que tienen así como las unidades en caso de que las tuvieran. Al definir un parámetro en esta pestaña se crea automáticamente una variable del tipo indicado a la cual se le asigna el valor y nombre indicado durante la definición del parámetro. Por este motivo, a la hora de escribir los nombres de los parámetros es importante tener en cuenta las reglas de sintaxis del lenguaje Matlab, ya que de lo contrario se producirá un error cuando se cree la variable correspondiente a dicho parámetro.

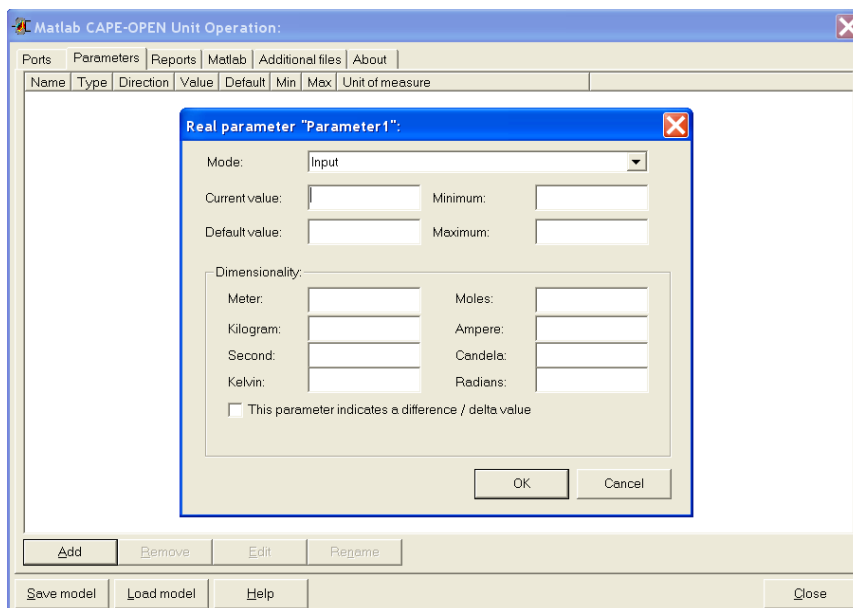


Figura 10. Definición de un parámetro en la unidad MATLAB

Para acabar con la implementación de una unidad MATLAB, deben introducirse en la pestaña Matlab las líneas de código que contengan las órdenes que se ejecutarán en la simulación de dicha unidad. Estas órdenes deben permitir el cálculo de las propiedades de las corrientes de salida así como de los parámetros de salida a partir de las propiedades de las corrientes de entrada y los parámetros de entrada. En consecuencia, la estructura general del código de esta unidad será la siguiente:

1. Lectura de las propiedades de las corrientes de entrada y los valores de los parámetros de entrada. Esto puede hacerse a través de las siguientes instrucciones:

$$\text{FeedProp1} = \text{getFeedProp}(\text{feedIndex}, \text{propName})$$

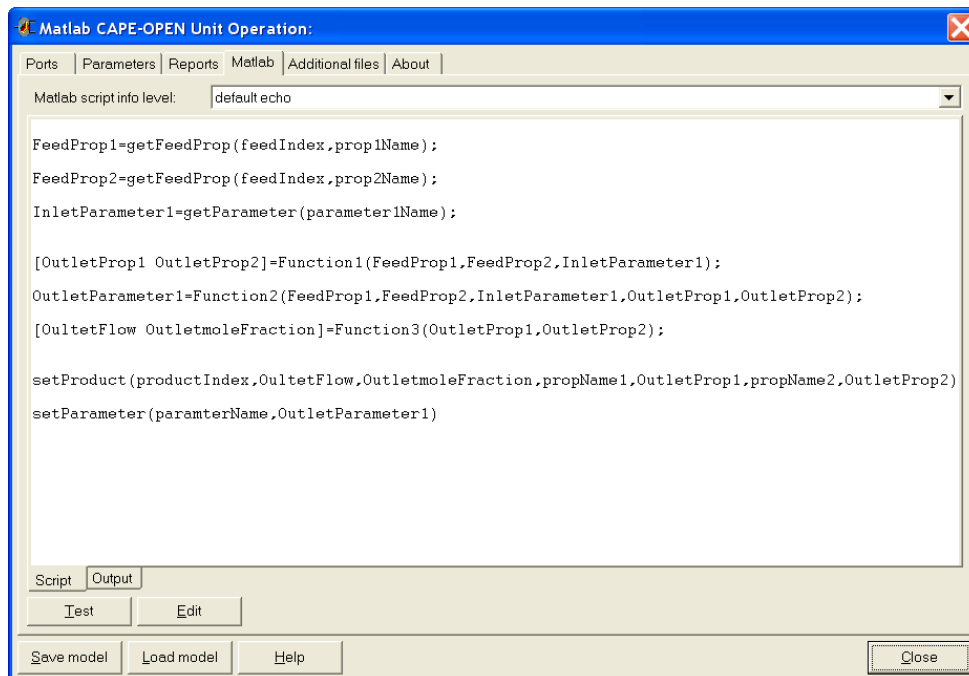
$$\text{InletParameter1} = \text{getParameter}(\text{parameterName})$$

2. Cálculo de las propiedades de las corrientes de salida y los parámetros de salida a través de las ecuaciones del modelo que define el comportamiento de la unidad.
3. Asignación de los resultados en las variables correspondientes para que sean reconocidos por el simulador. Esta asignación también puede hacerse a través del uso de instrucciones de la unidad, las cuales son:

*setProduct(productIndex, totalFlow, moleFraction, propName1, propVal1, ...
 ... propName2, propVal2)*

setParameter(paramterName, value)

Las ecuaciones que permiten el cálculo de las variables de salida pueden escribirse directamente en el cuadro de configuración de la unidad (Figura 11) o bien pueden cargarse a través de la pestaña Additional Files en el caso de que se tenga definida una función en Matlab que ya las contenga (Figura 12). De esta forma, en lugar de escribir el código en el cuadro de configuración de la unidad, basta con cargar el fichero Matlab y llamar a la función correspondiente cuando sea necesario. En las siguientes figuras puede observarse el aspecto general que presenta el código de la unidad una vez realizada la implementación.



The screenshot shows a window titled "Matlab CAPE-OPEN Unit Operation:" with a menu bar containing "Ports", "Parameters", "Reports", "Matlab", "Additional files", and "About". Below the menu bar is a dropdown menu for "Matlab script info level:" set to "default echo". The main area is a script editor containing the following MATLAB code:

```
FeedProp1=getFeedProp(feedIndex,prop1Name);
FeedProp2=getFeedProp(feedIndex,prop2Name);
InletParameter1=getParameter(parameter1Name);

[OutletProp1 OutletProp2]=Function1(FeedProp1,FeedProp2,InletParameter1);
OutletParameter1=Function2(FeedProp1,FeedProp2,InletParameter1,OutletProp1,OutletProp2);
[OutletFlow OutletmoleFraction]=Function3(OutletProp1,OutletProp2);

setProduct(productIndex,OutletFlow,OutletmoleFraction,propName1,OutletProp1,propName2,OutletProp2)
setParameter(paramterName,OutletParameter1)
```

At the bottom of the window, there are tabs for "Script" and "Output", and buttons for "Test", "Edit", "Save model", "Load model", "Help", and "Close".

Figura 11. Aspecto genérico del código de la pestaña MATLAB

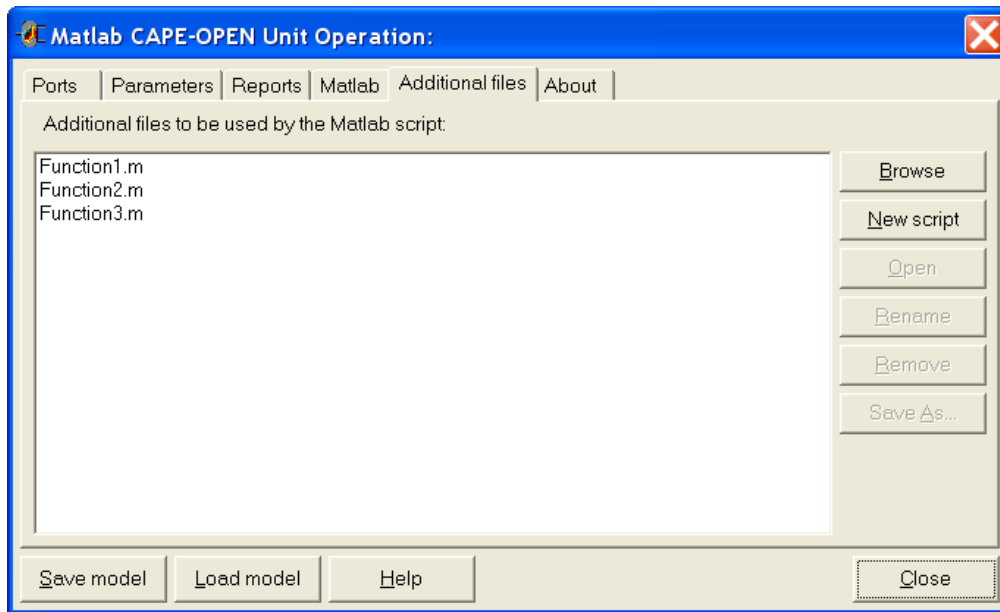


Figura 12. Aspecto genérico de la pestaña Additional Files

5.2. Implementación de una etapa de membrana

Haciendo uso de los conocimientos adquiridos en bibliografía ^[14-16], se ha implementado una membrana a través de la unidad MATLAB. Tras definir que el equipo consta de una entrada y dos salidas, es necesario definir los siguientes parámetros de entrada para poder utilizar posteriormente el modelo de disolución-difusión:

- Características geométricas de la membrana: número de módulos, número de tubos, área del módulo, sección de paso del módulo y longitud del módulo.
- Temperatura de operación del equipo.
- Parámetros del modelo de difusión-disolución: permeabilidad al agua (L_p), permeabilidad al soluto (B), coeficiente de pérdida de presión (k_{perd}) y factor osmótico (parte constante del coeficiente α).

Dado que las unidades de los parámetros deben introducirse mediante análisis dimensional, para facilitar la interpretación de las mismas los parámetros se definen adimensionalmente en la pestaña Parameters y en la pestaña Name del cuadro de diálogo de la unidad se indican las unidades en que deben ser introducidos mediante los comentarios de que se dispone.

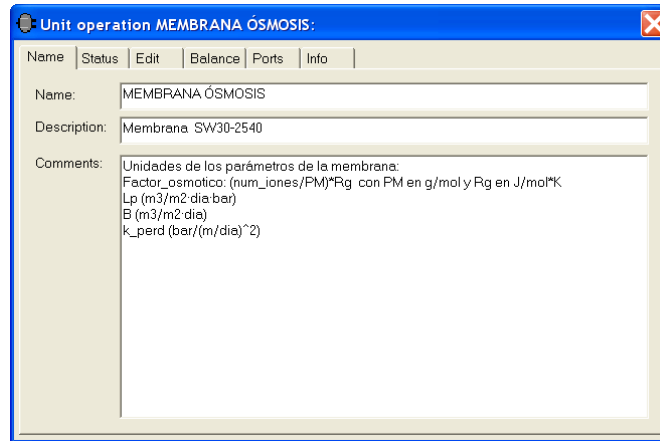


Figura 13. Comentario donde se especifican las unidades en que deben introducirse los parámetros

En cuanto al código que define la etapa de membrana, en la primera parte de lectura de las propiedades de la corriente de entrada es necesario emplear una estructura for-switch que permita separar el flujo de cada componente en una variable diferente, ya que la instrucción de la unidad getFeedProp solo permite almacenar los flujos en un vector sin identificar cuál corresponde a cada componente.

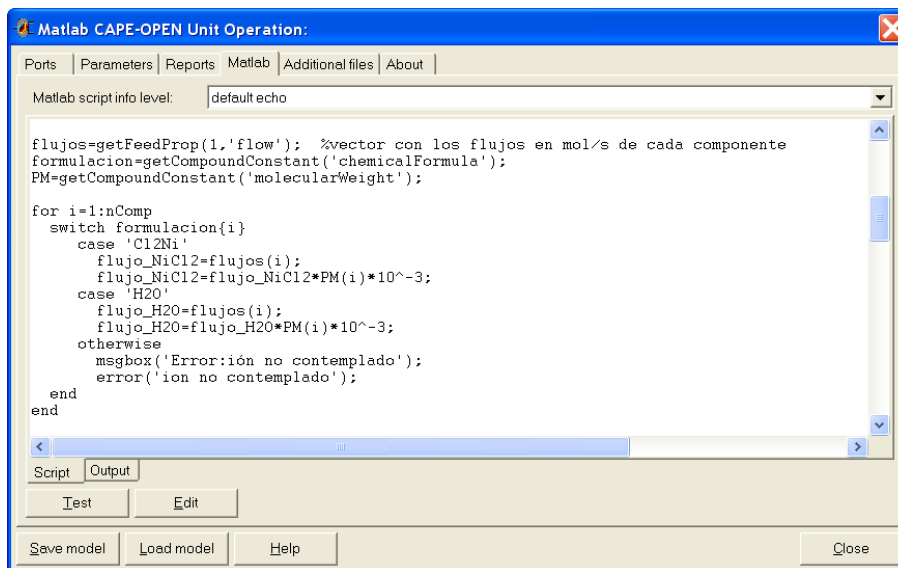


Figura 14. Identificación del flujo de cada componente y asignación en una variable independiente

Una vez se conoce el flujo que corresponde a cada uno de los componentes es posible calcular las variables de entrada que requiere el modelo de disolución-difusión, las cuales son el caudal de agua y la concentración del soluto. Tras definir dichas variables es posible llamar a la función Membrana para obtener los caudales y concentración de las corrientes de rechazo y permeado, debiéndose haber cargado previamente en la pestaña Additional Files los ficheros que contienen las ecuaciones diferenciales necesarias para poder utilizar dicha función, cuyo código informático puede consultarse en el capítulo 1 de los Anexos a la memoria.

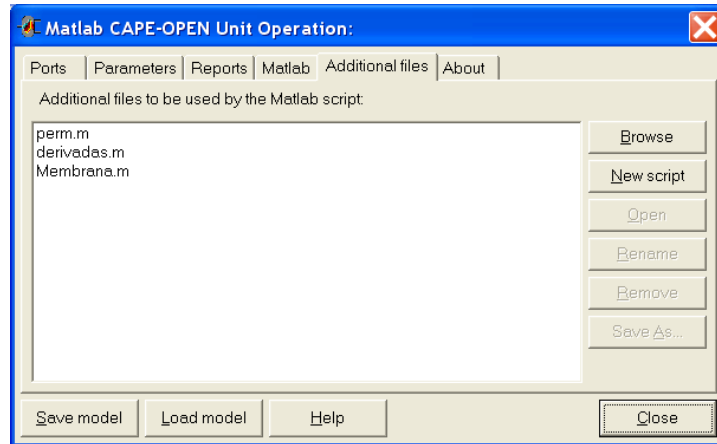


Figura 15. *Ficheros que contienen las ecuaciones diferenciales del modelo de disolución-difusión*

La función Membrana devuelve dos vectores W_r y W_p donde se tienen almacenados el caudal, concentración y presión de las corrientes de rechazo y permeado respectivamente. Por lo tanto, para poder definir las corrientes de salida mediante la instrucción `setProduct` es necesario obtener las propiedades de las corrientes de salida requeridas por dicha instrucción a partir de los resultados de la función Membrana. En la siguiente figura puede observarse las órdenes a ejecutar para realizar dicha operación en la corriente de rechazo.

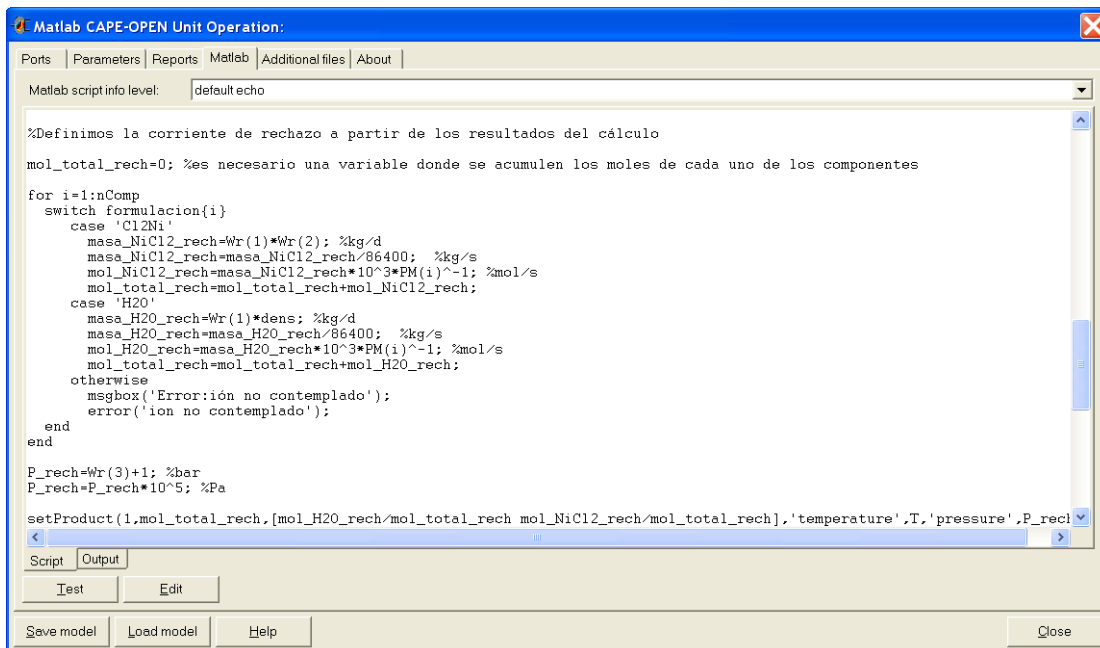


Figura 16. *Definición de la corriente de rechazo a partir de los resultados del modelo de disolución-difusión*

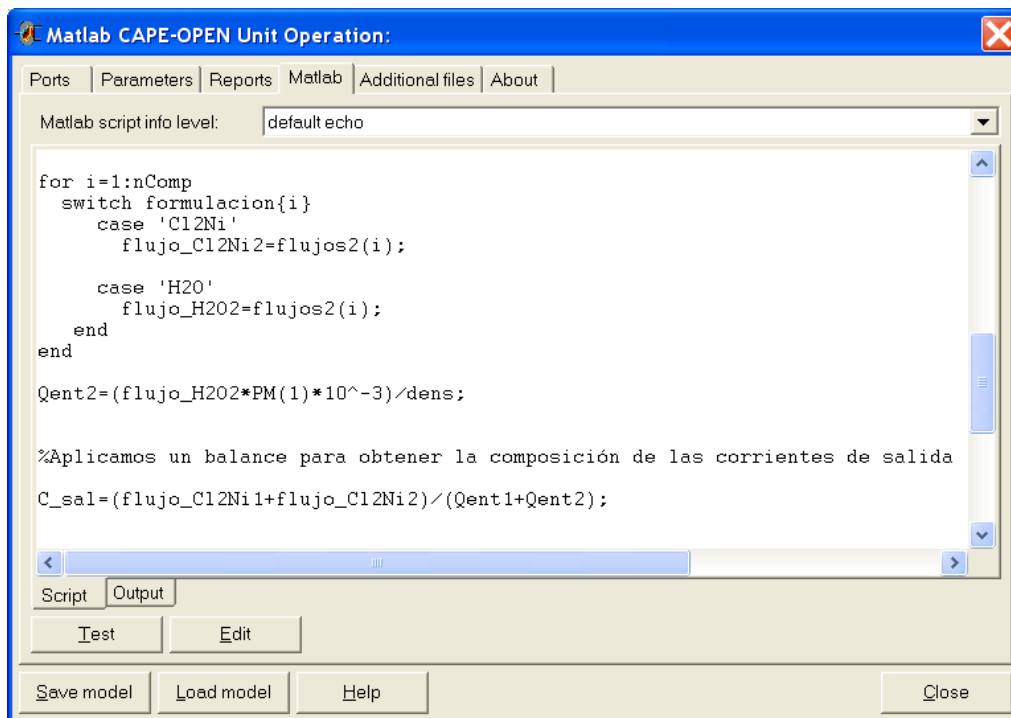
Para finalizar con la implementación del código de la membrana basta con repetir la estructura de la Figura 16 para la corriente de permeado, de manera que todas las corrientes de salida quedan definidas a partir de las corrientes de entrada y los parámetros de la unidad.

En el capítulo 2 de los Anexos puede verificarse la validez de este código, donde se compara el resultado obtenido en COCO con el obtenido en otros programas de cálculo (MATLAB y MATHCAD) al resolver un caso base.

5.3. Implementación de un tanque de lavado

Para obtener los parámetros de entrada al modelo del tanque de lavado solo se necesita utilizar la misma estructura for-switch que se ha utilizado en la etapa de membrana. Con dicha estructura se obtiene el flujo molar de impureza, que es igual al producto de caudal y concentración, así como el flujo molar de agua, que a través de la densidad y peso molecular de esta sustancia puede transformarse a caudal volumétrico.

Por lo tanto, la codificación de esta unidad consiste en aplicar la ecuación del modelo de un tanque de lavado tras obtener los parámetros de entrada correspondientes mediante una estructura for-switch.



```

Matlab CAPE-OPEN Unit Operation:
Ports | Parameters | Reports | Matlab | Additional files | About
Matlab script info level: default echo

for i=1:nComp
    switch formulacion{i}
        case 'Cl2Ni'
            flujo_Cl2Ni2=flujos2(i);

        case 'H2O'
            flujo_H2O2=flujos2(i);
        end
    end
end

Qent2=(flujo_H2O2*PM(1)*10^-3)/dens;

%Aplicamos un balance para obtener la composición de las corrientes de salida
C_sal=(flujo_Cl2Ni1+flujo_Cl2Ni2)/(Qent1+Qent2);
  
```

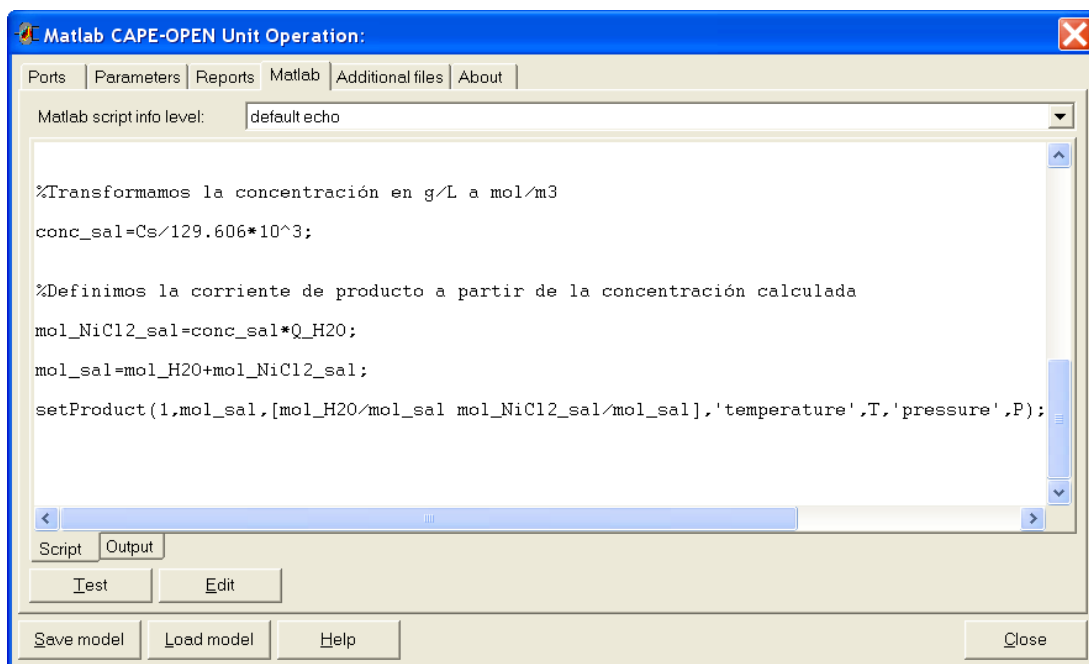
Figura 17. Codificación del modelo de un tanque de lavado en una unidad MATLAB

Como sucede con el resto de unidades MATLAB, para finalizar la implementación basta con definir las diferentes corrientes de salida a partir de los resultados obtenidos del modelo físico que describe la unidad. En este caso la obtención de los parámetros requeridos para dicha definición resulta mucho más sencilla que en el caso de la etapa de membrana, ya que simplemente multiplicando la concentración de salida calculada por el caudal de cada corriente se tienen definidos los flujos molares de todos los componentes en las corrientes de salida.

5.4. Implementación de un intercambio iónico

La implementación de la unidad de intercambio iónico resulta ser la más sencilla de todas dado que, además de la estructura for-switch para separar los flujos utilizada en las otras codificaciones, solo es necesario transformar las unidades de la concentración de salida fijada en los parámetros de la unidad para poder definir todas las características de la corriente de salida.

De la misma forma que sucedía en el caso del tanque de lavado, una vez se conoce la concentración y caudal de la corriente de salida pueden obtenerse el flujo molar de impureza como producto de ambos, teniendo así determinadas todas las propiedades de la corriente de salida necesarias para su definición.



```

Matlab CAPE-OPEN Unit Operation:
Ports | Parameters | Reports | Matlab | Additional files | About |
Matlab script info level: default echo

%Transformamos la concentración en g/L a mol/m3
conc_sal=Cs/129.606*103;

%Definimos la corriente de producto a partir de la concentración calculada
mol_NiCl2_sal=conc_sal*Q_H2O;
mol_sal=mol_H2O+mol_NiCl2_sal;
setProduct(1,mol_sal,[mol_H2O/mol_sal mol_NiCl2_sal/mol_sal],'temperature',T,'pressure',P);
  
```

Figura 18. Codificación del modelo de un intercambio iónico

5.5. Introducción de componentes en la base de datos de COCO

Para poder simular el proceso de lavado objeto de estudio en este proyecto es necesario definir tres nuevos componentes en la base de datos de COCO: Ni^{2+} , Cl^- y NiCl_2 . Esta tarea puede acometerse mediante el servidor termodinámico de COCO (TEA) o bien mediante el programa PCD manager que se instala durante la instalación del propio simulador COCO. En este caso se ha decidido optar por la segunda alternativa, puesto que presenta una interfaz gráfica que facilita la interpretación de los parámetros necesarios para definir el componente.

Lo primero que debe hacerse para introducir un nuevo componente en COCO es definirlo de forma unívoca. Ello se hace a través de la estructura molecular del componente, su peso molecular y su número de identificación CAS. El número CAS correspondiente a cada uno de los compuestos que deben introducirse es ^[17]:

- Ni²⁺: 7440-02-0
- Cl⁻: 16887-00-6
- NiCl₂: 7791-20-0

Una vez definido el componente es necesario caracterizar sus propiedades termodinámicas, lo cual requiere una gran cantidad de información como por ejemplo las propiedades del punto crítico, la evolución de las propiedades físicas con la temperatura, propiedades moleculares como el momento dipolar, etc. En muchos casos esta caracterización requiere de datos experimentales, bien de ensayos propios o de resultados presentados en trabajos de otros autores.

Por este motivo, una caracterización completa de las propiedades termodinámicas de los componentes queda fuera del objeto del proyecto, dado que ello exigiría una cantidad de recursos que no permitiría realizar con éxito el resto de las tareas del mismo. Por tanto solo se han definido las propiedades mínimas que exige el programa, que son la entalpía, energía libre y entropía estándar de formación. En la siguiente tabla se presentan los valores de cada una de estas variables para cada uno de los componentes a introducir ^[18-20].

Tabla 5. Propiedades termodinámicas de los compuestos a introducir en COCO

Compuesto	ΔH^0 (J/kmol)	ΔG^0 (J/kmol)	ΔS^0 (J/kmol·K)
Ni ²⁺	$-6.400 \cdot 10^7$	$-4.556 \cdot 10^7$	-61872.5
Cl ⁻	$-6.674 \cdot 10^8$	$-1.312 \cdot 10^8$	-121577
NiCl ₂	$-3.156 \cdot 10^8$	$-2.721 \cdot 10^8$	-145879

Por último, el programa PCD manager permite definir los parámetros de interacción en caso de se quiera utilizar el modelo UNIFAC para el cálculo de las propiedades del componente o el factor acéntrico si se prefiere emplear el modelo SRK. La complejidad de esta tarea también pone en compromiso el éxito del resto del proyecto, por lo que tampoco se ha abordado su resolución, optando por calcular las propiedades de los componentes con modelos más sencillos que no requieran definir parámetros adicionales.

En todo caso, las únicas propiedades de las corrientes que se tienen en cuenta durante el análisis del proceso son el caudal y la concentración, por lo que una imprecisión en el cálculo de las propiedades termodinámicas de las mismas no afecta de manera significativa a la validez de los resultados obtenidos.

Además, dado que en el proceso no aparecen reacciones químicas ni cambios de fase, así como tampoco variaciones en la temperatura de las corrientes, no se necesita una definición precisa de los componentes para obtener resultados fidedignos en la simulación. Por lo tanto, dado que el tipo de proceso analizado no requiere precisión en este aspecto, se ha decidido no dedicar excesivos recursos al mismo para poder realizar otras tareas más importantes, como por ejemplo la definición de la unidad de ósmosis, con mayor precisión.

6. SIMULACIÓN DE UN PROCESO INDUSTRIAL

6.1. Alternativas de mejora del proceso analizadas

6.1.1. *Modificación de la posición de recirculación*

Mediante el uso de las unidades de proceso implementadas y los componentes añadidos en la base de datos es posible obtener la composición y caudal de todas las corrientes del proceso de enjuague (Figura 21). En estos resultados se observa que la concentración del permeado de la membrana difiere de la concentración de salida del séptimo tanque, por lo que el proceso de mezcla podría mejorarse si dichas concentraciones se asemejarán más.

En consecuencia, la posibilidad de mejora más sencilla que puede plantearse es la modificación de la posición donde se introduce el permeado de la membrana, de manera que se consiga una mezcla de aguas de calidad más similar y así la cadena de tanques funcione más eficientemente, con lo que puede reducirse el caudal recirculado al intercambio iónico y así el coste de regeneración de la resina.

En la Figura 19 se puede observar como a medida que se adelanta la posición donde se introduce el permeado en la cadena de tanques la concentración a la salida del sistema se incrementa. Este incremento implica que se requiere de un mayor uso de la unidad de intercambio iónico para seguir cumpliendo con la razón de dilución, por lo que el coste del tratamiento aumentaría al necesitarse una cantidad mayor de resina.

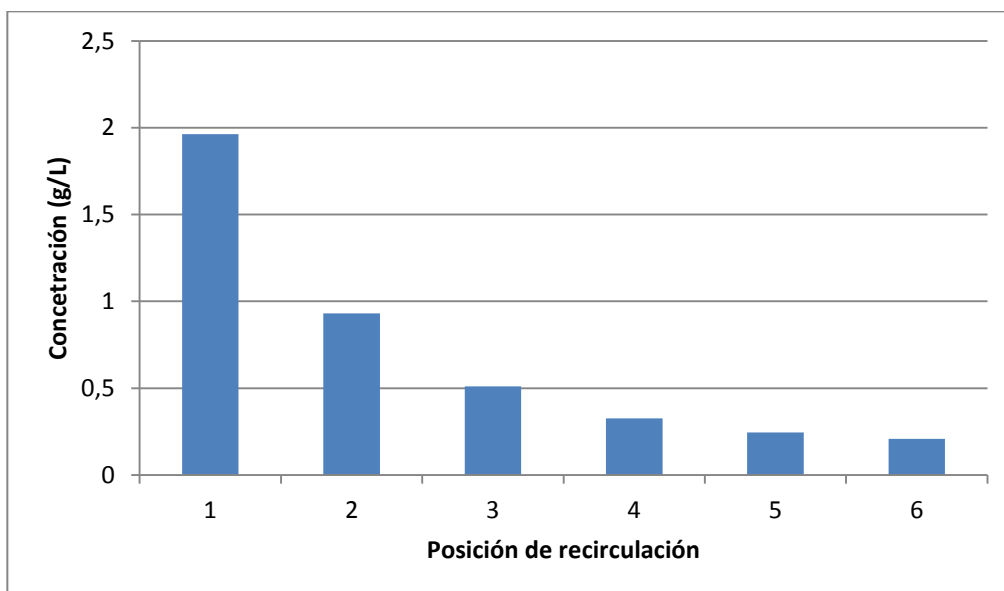


Figura 19. Concentración en la corriente de salida para diferentes posiciones de recirculación

Al adelantar la posición de recirculación el permeado que se obtiene de la unidad de membrana tiene un mayor grado de pureza, por lo que su calidad difiere en mayor medida del agua con que se mezcla. En las Figuras 22 y 23 se presentan los resultados de la simulación para dos de las posiciones estudiadas, pudiéndose observar que la diferencia de concentraciones entre las aguas mezcladas es de 1.930 g/L en la posición 4 y de 6.050 g/L en la 3, en comparación con los 0.795 g/L que se tenía en la posición inicial.

Como ya se ha comentado, esto tiene un efecto negativo sobre la eficiencia en la cadena de lavado, puesto que esta no funciona de manera homogénea sino que se tiene una primera zona con alta carga de trabajo y una segunda zona infrautilizada. Por ejemplo, en la posición 3 antes de la recirculación la concentración se reduce en un 95.64%, mientras que en la zona posterior a esta solo disminuye un 70.97%.

Además, a medida que se adelanta la posición de la recirculación se produce una reducción de la concentración del rechazo de la membrana, tal y como puede observarse en la Figura 20. Este hecho provoca que no pueda recircularse directamente la corriente al baño de niquelado, dado que para poder hacer esto es necesario que la corriente tenga la misma concentración que el tanque donde se recircula. Por lo tanto, al modificar la posición de la recirculación debería verse la corriente de rechazo, por lo que aparece un impacto ambiental en el proceso, o bien someterla a un tratamiento de concentración para poder recircularla al tanque de niquelado.

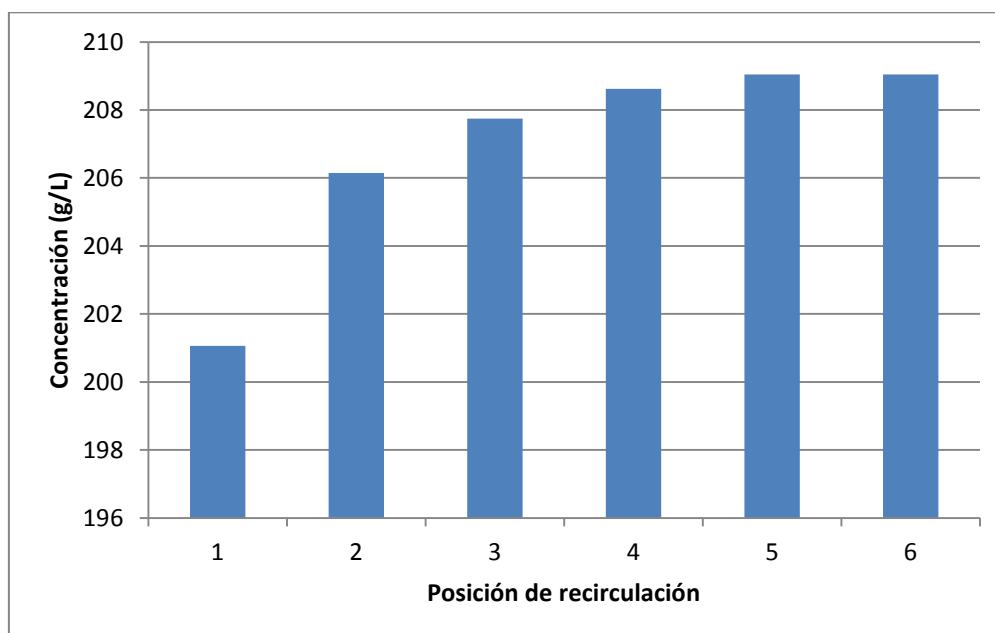


Figura 20. Concentración en la corriente de rechazo para diferentes posiciones de recirculación

En base a estos resultados, puede decirse que cualquier modificación que se haga de la posición de la recirculación empeorará el funcionamiento del proceso, puesto que se producirá una pérdida de eficiencia en la parte final de la cadena de lavado y ello obligará a trasegar un mayor caudal en el intercambio iónico para seguir cumpliendo con la razón de dilución establecida, a la vez que se requiere de un tratamiento del rechazo de la membrana para poder recircularlo y así eliminar todo impacto ambiental del proceso.

6.1.2. Utilización de una segunda etapa de membrana

En la Figura 21 se observa que la concentración de la corriente de permeado es superior a la concentración del agua de lavado con que se mezcla. Por lo tanto, una opción para mejorar el proceso es obtener una corriente de permeado más purificada, de manera que su concentración sea más similar a la del último tanque de lavado y así se tenga un proceso de mezcla más eficiente.

Para conseguir un permeado de mayor pureza puede emplearse una segunda etapa de membrana que trate la corriente de permeado de la primera etapa de membrana, de manera que así se alcanza una mayor conversión a la salida del sistema de membranas. Además, si en lugar de trasegar todo el permeado a la segunda etapa de membrana parte se recircula a la cadena de tanques, se tiene dos corrientes de agua de diferente calidad que pueden inyectarse en puntos diferentes de la cadena de lavado, de forma que, al introducir una mayor cantidad de agua purificada, se tiene un sistema de lavado más eficiente y por tanto puede reducirse el consumo en el intercambio iónico.

La corriente de rechazo que se genera en la segunda etapa de membrana puede recircularse a la entrada del sistema de membranas, de manera que se consigue evitar el impacto ambiental que dicha corriente supone a la vez que se incrementa la velocidad de paso en las membranas.

A través del uso del simulador COCO pueden verificarse todas estas afirmaciones, pudiéndose comprobar en la Figura 24 que una segunda etapa de membrana realmente permite alcanzar la razón de dilución con un menor uso del intercambio iónico. Concretamente se alcanza la razón de dilución trasegando solo 14.769 L/h a través del intercambio iónico, lo que supone una reducción del 73.96% en el caudal de dicho equipo. Esto conlleva una reducción en los costes de operación del proceso, ya que el hecho de trasegar un menor caudal implica que el número de regeneraciones del lecho de resina de intercambio será menor.

Por otro lado, en el proceso se produce un aumento de costes debido al consumo energético de la segunda bomba y a la necesidad de invertir en la compra de una segunda membrana y los tanques que esta requiere. Sin embargo, el intercambio iónico es un tratamiento de agua con un coste muy elevado, de manera que la reducción en el consumo de resina compensará el aumento de costes que supone la introducción de la segunda etapa de membrana.

Las condiciones de operación óptimas serán entonces aquellas en las que el caudal recirculado al intercambio iónico sea el mínimo posible, ya que de esta forma el tratamiento del agua que se hacía con esta unidad pasa a hacerse con una unidad de membrana, lo cual resulta más eficiente desde el punto de vista económico. Por lo tanto, las condiciones de operación óptimas serán aquellas en las que no se requiera del uso de un intercambio iónico (reducción del 100% del caudal recirculado) en el último tanque, quedando el sistema de tratamiento compuesto únicamente por las unidades de membrana.

En la Figura 25 se presenta la combinación de parámetros en la que el sistema de tratamiento permite cumplir con la razón de dilución sin necesidad de utilizar un intercambio iónico.

A la vista de los resultados obtenidos en la simulación, puede afirmarse que la introducción de una segunda etapa de membrana en el proceso permite reducir la carga de trabajo de la unidad de intercambio iónico hasta el punto que esta deja de ser necesaria para alcanzar la razón de dilución de 1000 a la salida del sistema. Por lo tanto, introduciendo esta nueva unidad se obtiene un sistema de tratamiento que cumple con los requisitos de calidad exigidos con un coste de operación menor, puesto que el tratamiento del agua con una etapa de membrana es más barato que con una resina de intercambio iónico.

6.2. Justificación económica de la mejora planteada

6.2.1. Coste inicial del tratamiento de agua

Los costes de operación asociados al sistema de tratamiento de la Figura 19 son los correspondientes al consumo energético de la bomba y al coste de reposición de la resina de intercambio iónico. En relación con el primero de estos costes, los tanques de almacenamiento otorgan la flexibilidad de succionar durante el periodo del día que se desee, de manera que la hora que la bomba tiene que estar succionando agua del tanque de alimentación puede situarse en las horas valle del precio de la energía. Por lo tanto, el coste de la energía consumida en este proceso puede calcularse mediante el uso de la siguiente expresión ^[21]:

$$\text{Coste energía (€/año)} = 6.8324 \cdot \{\text{Potencia instalación (kW)}\} + 0.0866 \cdot \{\text{Consumo (kW/año)}\}$$

Suponiendo que la potencia contratada en la instalación es de 15 kW y empleando el resultado de la simulación, según el cual la bomba consume 4.141 kW/día, el coste energético del sistema de tratamiento resulta ser de:

$$\text{Coste energía} = 6.8324 \cdot 15 + 0.0866 \cdot (4.141 \cdot 365) = 233.38 \text{ €/año}$$

Por otro lado, el coste de la resina LEWATIT TP 260 es de 0.1828 €/g ^[22], de manera que a partir de las condiciones de operación del proceso y las características técnicas de la resina es posible obtener el coste anual de operación de este equipo:

$$\text{Masa NiCl}_2 \text{ depositada} = 56.727 \cdot (0.208 - 0.01) = 11.232 \text{ g/h}$$

$$\begin{aligned} \text{Masa Ni}^{2+} \text{ depositada} &= 11.232 \frac{\text{g NiCl}_2}{\text{h}} \cdot \frac{1 \text{ mol NiCl}_2}{129.606 \text{ g NiCl}_2} \cdot \frac{1 \text{ mol Ni}^{2+}}{1 \text{ mol NiCl}_2} \\ &= 0.0867 \text{ mol Ni}^{2+}/\text{h} = 0.0433 \text{ eq Ni}^{2+}/\text{h} \end{aligned}$$

$$\text{Consumo resina} = \frac{0.0433 \text{ eq Ni}^{2+}/\text{h}}{2.3 \text{ eq Ni}^{2+}/(\text{L resina})} = 0.0188 \text{ (L resina)/h}$$

$$\text{Consumo resina} = 0.0188 \text{ L resina/h} \cdot \frac{720 \text{ g resina}}{1 \text{ L resina}} \cdot \frac{8760 \text{ h}}{1 \text{ año}} = 118739.89 \text{ g resina/año}$$

Suponiendo que el lecho de intercambio iónico tiene un volumen de 1 L y que este puede ser regenerado hasta 30 veces antes de quedar en desuso, el coste asociado a la compra de nueva resina será de:

$$n^{\circ} \text{ reposiciones} = \frac{118739.89}{720 \cdot 30} = 5.497 \approx 6$$

$$\text{Coste resina} = 6 \cdot 720 \cdot 0.1828 = 789.70 \text{ €/año}$$

Además de los costes de operación, en el sistema de tratamiento también aparecen los costes asociados a los equipos que en él se utilizan. En la siguiente tabla se presenta el coste unitario de cada uno de estos equipos, así como su coste anualizado, que se obtiene a través de la siguiente expresión:

$$\text{Coste anual equipo} = \frac{\{\text{Precio unitario equipo}\} \cdot \{\text{Número unidades}\}}{\{\text{Vida útil}\}}$$

Tabla 6. Coste de los equipos del sistema de tratamiento inicial

<i>Nombre equipo</i>	<i>Precio unitario (€)</i>	<i>Número unidades</i>	<i>Vida útil (años)</i>	<i>Coste anual (€/año)</i>
Módulo SW30-2540	144.63 ^[23]	6	4	216.95
Bomba 310S	1118.38 ^[24]	1	15	74.56
Tanque polimérico 1000 L	325	3	15	65

Combinando los costes anualizados de los equipos con los costes de operación previamente calculados es posible determinar el coste inicial del sistema de tratamiento de aguas industriales, el cual queda detallado en la siguiente tabla:

Tabla 7. Coste del sistema de tratamiento inicial

<i>Concepto</i>	<i>Coste anual (€/año)</i>
Etapa de membrana	216.95
Bomba 310S	74.56
Tanques de almacenamiento	65.00
Consumo energético	233.38
Consumo de resina	789.70
Subtotal	1,379.59

6.2.2. Coste del sistema de tratamiento mejorado

Como ya se ha comentado en el apartado anterior, al introducir una segunda unidad de membrana es posible alcanzar la razón de dilución sin utilizar un intercambio iónico. Por lo tanto, los costes de operación quedan reducidos al consumo energético, que en este caso es de 5.537 kW/día debido a que se cuenta con una bomba adicional para la segunda membrana.

$$\text{Coste energía} = 6.8324 \cdot 15 + 0.0866 \cdot (5.537 \cdot 365) = 277.51 \text{ €/año}$$

En cuanto a los costes de los equipos, en este sistema de tratamiento esta partida es mayor debido a que se introduce una segunda etapa de membrana. Por lo tanto, además de los equipos con que ya se contaba serán necesarios otros 6 módulos SW30-2540 para la segunda membrana, una bomba para presurizar el agua y otros 3 tanques de almacenamiento, si bien estos serán de un menor tamaño ya que la cantidad de agua a tratar es menor. Así mismo, el tamaño de los tanques requeridos en la primera etapa de membrana aumenta debido a que debe poder almacenarse también el rechazo recirculado de la segunda etapa de membrana.

Tabla 8. Coste de los equipos del sistema de tratamiento mejorado

<i>Nombre equipo</i>	<i>Precio unitario (€)</i>	<i>Número unidades</i>	<i>Vida útil (años)</i>	<i>Coste anual (€/año)</i>
Módulo SW30-2540	144.63	12	4	433.90
Bomba 310S	1118.38	1	15	74.56
Bomba 2SF20ES	487.35 ^[24]	1	15	32.49
Tanque polimérico 1200 L	350	3	15	70
Tanque polimérico 750 L	300	3	15	60

Tras recalcular los costes de operación y costes anualizados de equipos puede obtenerse el coste asociado al nuevo sistema de tratamiento, el cual se detalla en la siguiente tabla:

Tabla 9. Coste del sistema de tratamiento mejorado

<i>Concepto</i>	<i>Coste anual (€/año)</i>
Etapa de membrana	433.90
Bomba 310S	74.56
Bomba 2SF20ES	32.49
Tanques de almacenamiento	130.00
Consumo energético	277.51
Subtotal	948.46

La introducción de una segunda etapa de membrana permite reducir el coste del tratamiento de regeneración de agua en un 31.25%. Esta reducción de coste justifica la posibilidad de introducir este cambio en el proceso.

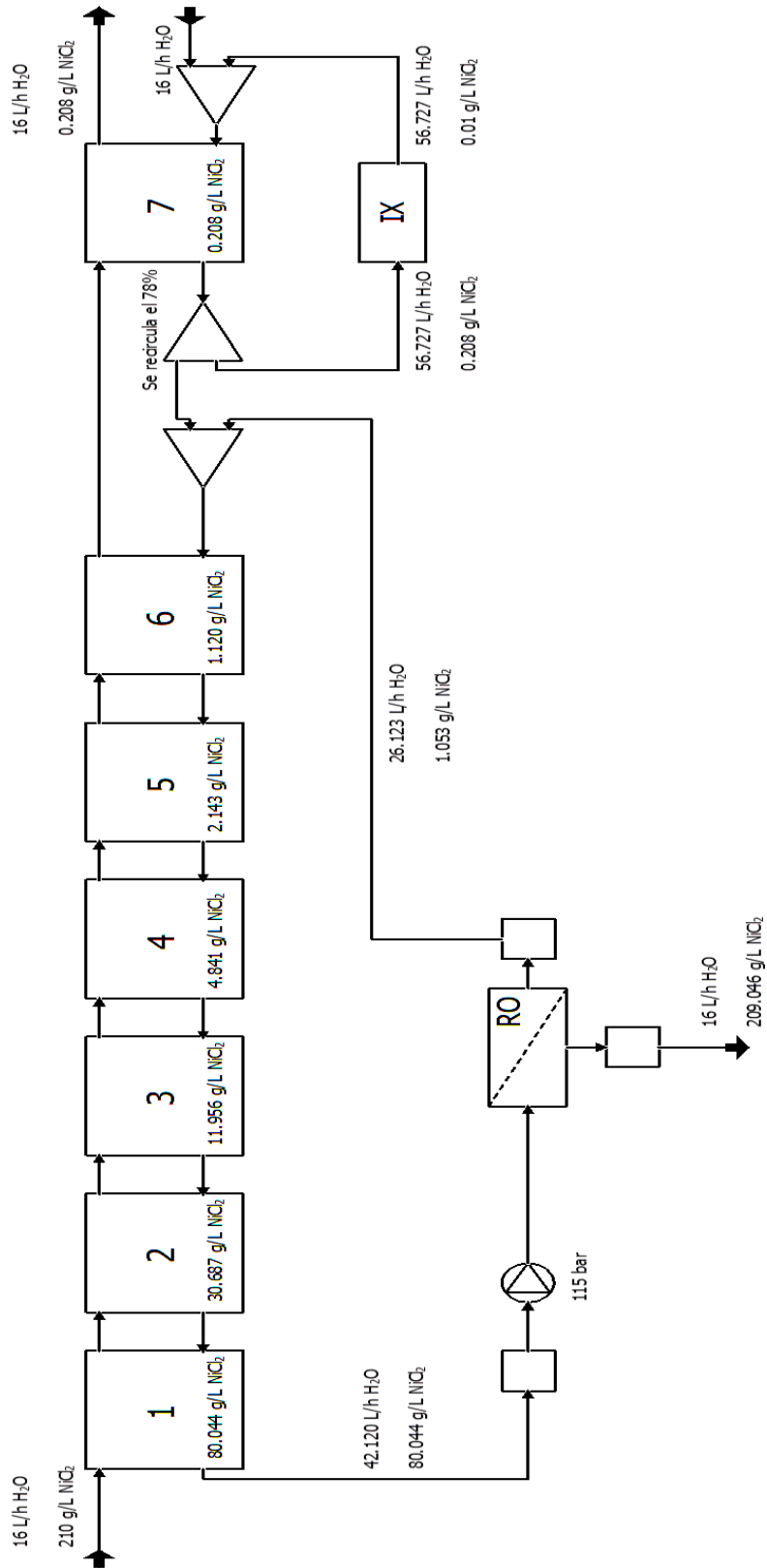


Figura 21. Condiciones de operación del proceso en estado estacionario

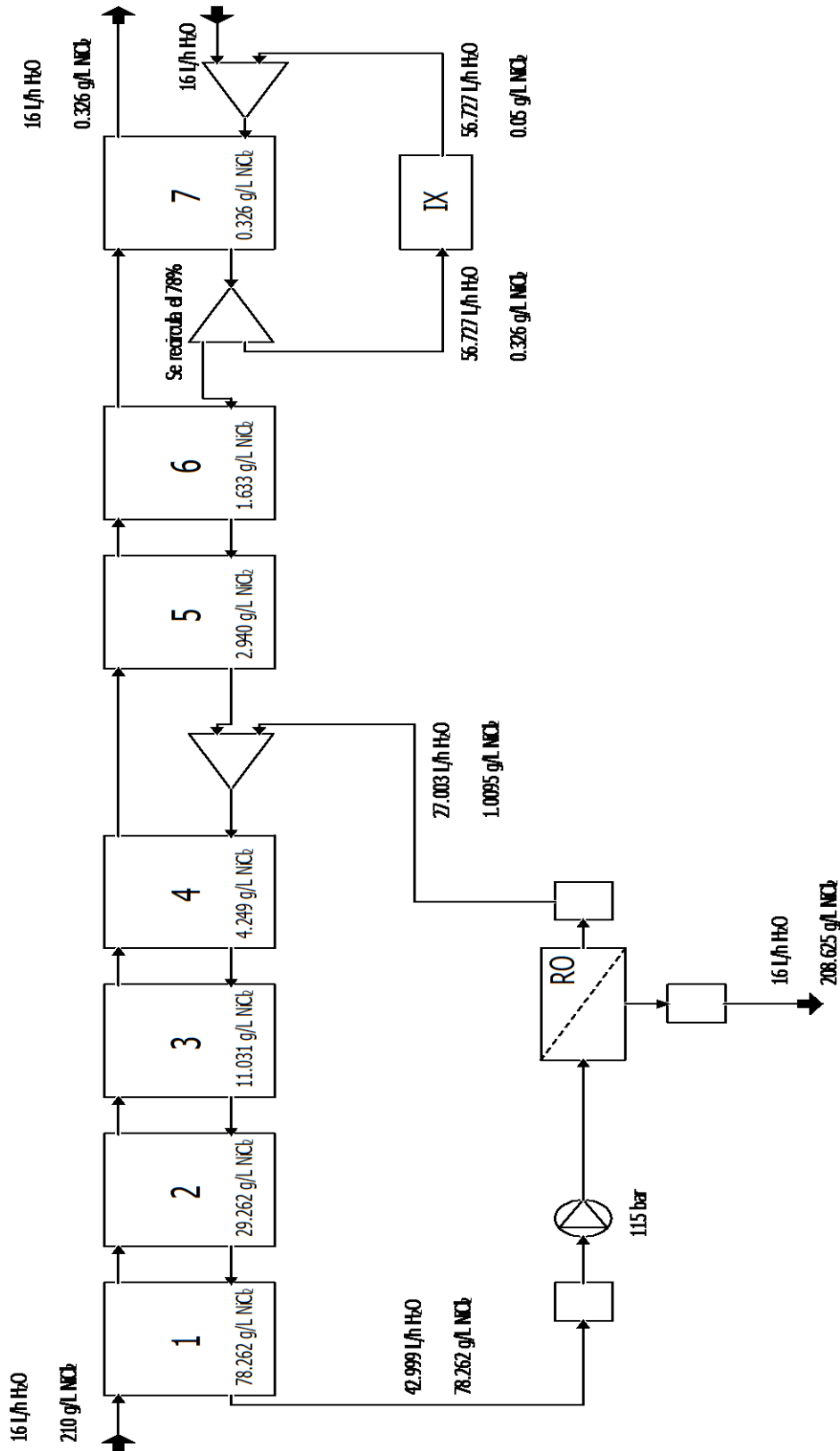


Figura 22. Condiciones de operación recirculando el permeado en la posición 4

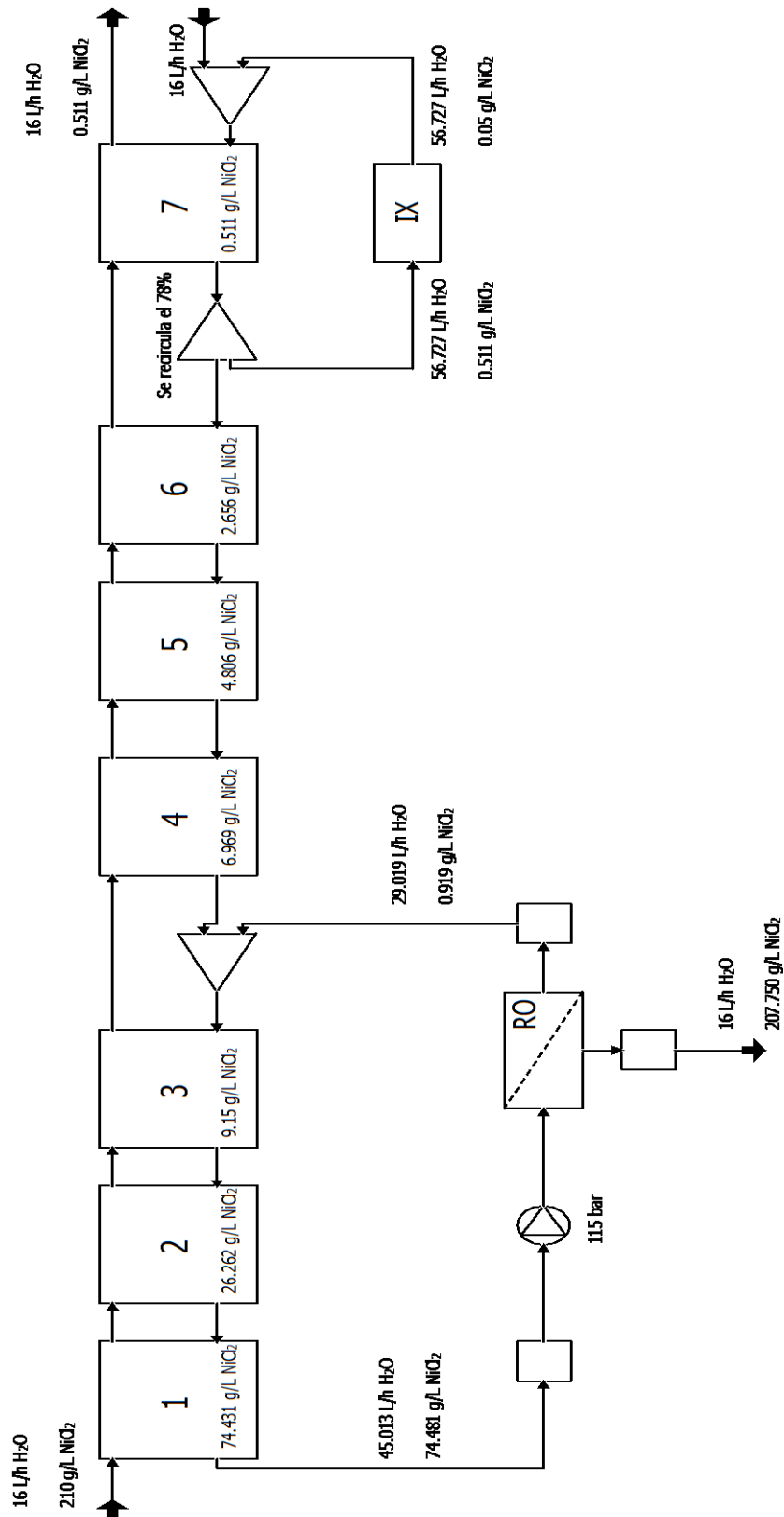


Figura 23. Condiciones de operación recirculando el permeado en la posición 3

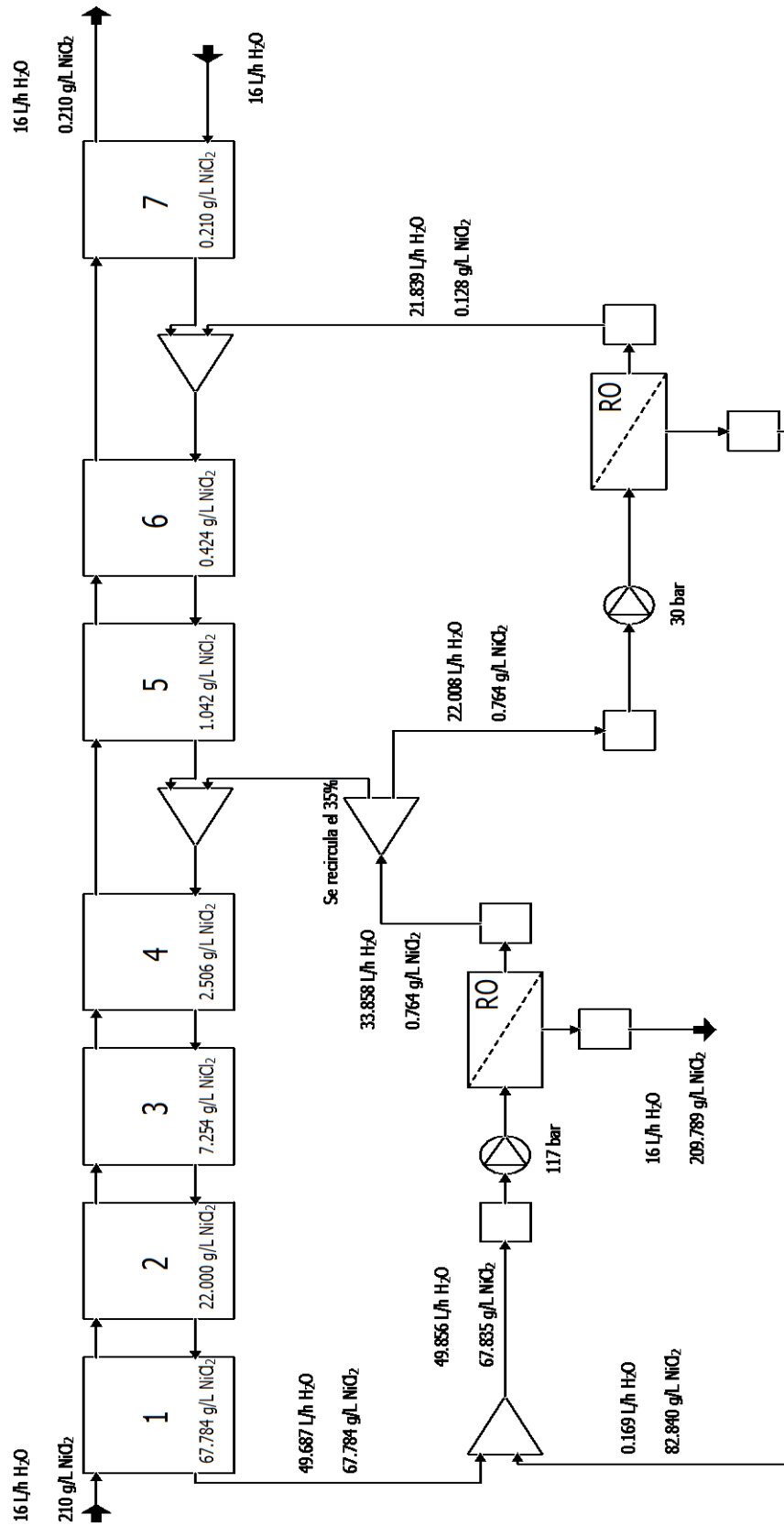


Figura 24. Simulación del proceso con una segunda etapa de membrana

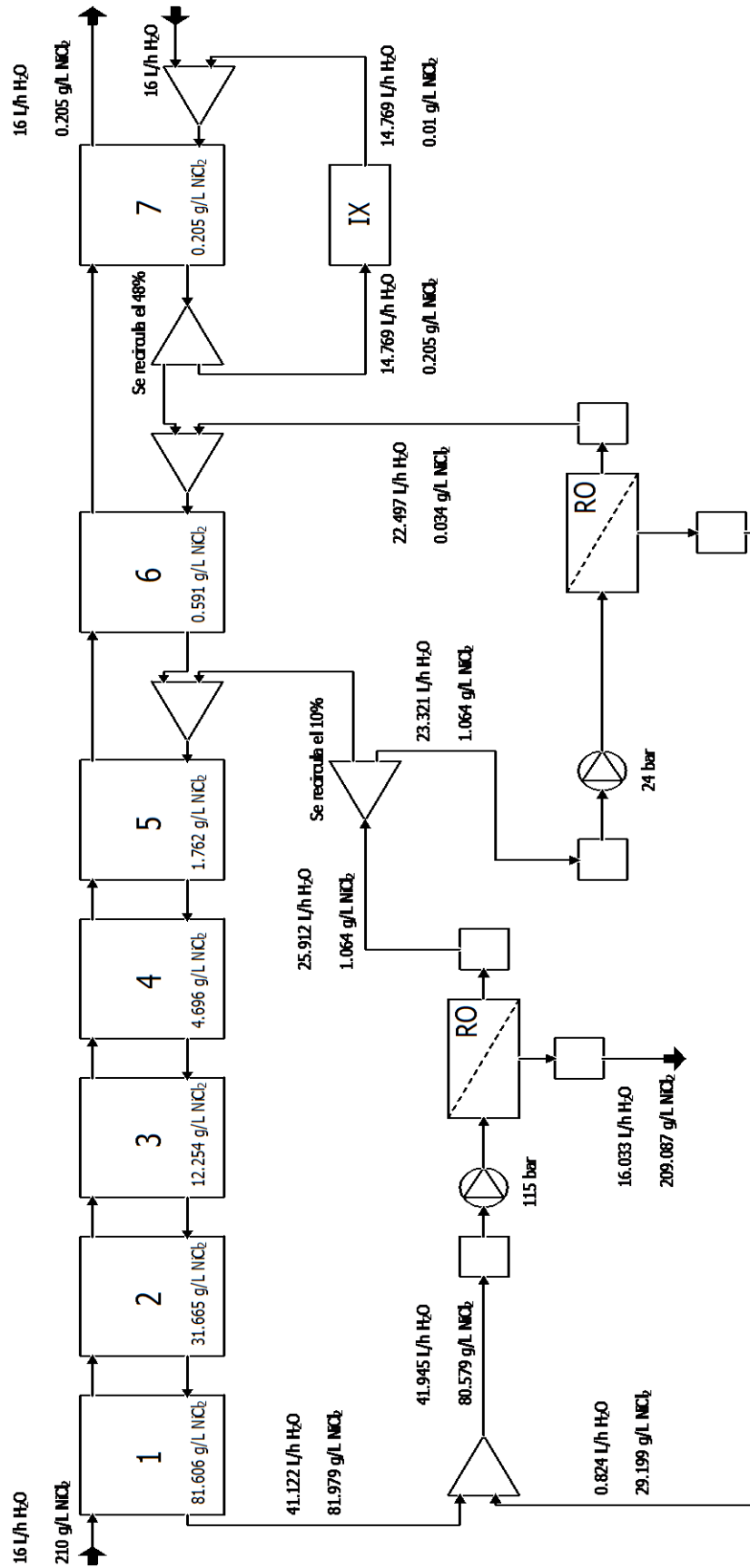


Figura 25. Condiciones de operación óptimas con una segunda etapa de membrana

7. CONCLUSIONES

En relación con la implementación de unidades de cálculo, el trabajo desarrollado ha permitido establecer las siguientes conclusiones:

- A través del uso de la unidad MATLAB proporcionada por la empresa AmsterCHEM se ha podido introducir con éxito tres nuevas unidades en el simulador COCO:
 - Unidad de ósmosis inversa compuesta por diferentes tubos en paralelo, teniendo cada tubo varios módulos en serie.
 - Tanque de lavado operando a contracorriente.
 - Lecho de intercambio iónico (no se tiene en cuenta la alternancia operación/regeneración).
- En el caso de la unidad de ósmosis inversa, el modelo, basado en las ecuaciones de disolución-difusión, incluye el efecto de la concentración por polarización. Dicho modelo consigue reproducir satisfactoriamente el comportamiento de estas unidades, si bien lo hace con una precisión algo menor a la de otros programas de cálculo como por ejemplo MATLAB o MATHCAD.
- La conexión predefinida entre PMC y PME que proporciona la unidad MATLAB facilita enormemente la tarea de introducir el modelo de ósmosis inversa en el simulador. Gracias a ello un usuario sin conocimientos de programación orientada a objetos puede ser capaz de implementar una unidad de complejidad elevada.
- Una vez añadidas las nuevas funcionalidades al simulador COCO (unidades de separación y componentes adicionales) se ha podido representar el comportamiento de una cascada de tanques de lavado, donde se retira el níquel procedente de un tanque de níquelado. Dicho proceso cuenta con diferentes recirculaciones, lo que permite afirmar que con el simulador COCO pueden abordarse cálculos de elevada complejidad.

En cuanto a la mejora del proceso a través del uso del simulador, se han llegado a las siguientes conclusiones:

- Cualquier desplazamiento de la unidad de mezcla produce un aumento de la concentración de salida a la vez que reduce la del rechazo de la membrana, por lo que ya no podría recircularse al tanque de níquelado.
- La posición inicial de recirculación es la óptima y por tanto el proceso no puede mejorarse modificando dicha posición.
- La introducción de una segunda etapa de membrana permite alcanzar la razón de dilución sin necesidad de emplear el lecho de intercambio iónico.
- El proceso con dos unidades de membrana es en torno a un 30% más barato que el proceso inicial de tratamiento.

Por último, puede afirmarse que el simulador COCO es una alternativa a considerar a la hora de analizar y diseñar procesos químicos por los siguientes motivos:

- Permite evaluar procesos con recirculaciones que serían complejos de estudiar utilizando otros métodos.
- La flexibilidad de la tecnología CAPE-OPEN permite que se pueda simular cualquier unidad de proceso, siendo el propio usuario quien pone los límites.
- Al ser software libre es una herramienta accesible a todo el mundo. Ello hace además que sea una buena alternativa para impartir prácticas de simulación, ya que los alumnos tendrían acceso al programa de manera gratuita y sencilla.
- La gran interoperabilidad del simulador hace que sea posible introducir no solo unidades de operación sino cualquier tipo de herramienta informática, como por ejemplo algoritmos de optimización.

8. LISTA DE SÍMBOLOS

Tabla 10. Símbolos y variables utilizados en el desarrollo de los modelos

<i>Símbolo</i>	<i>Variable</i>	<i>Unidades SI</i>
A	Área activa	m ²
B	Permeabilidad al soluto	m ³ /m ² ·s
b _π	Coefficiente osmótico	m ³ ·Pa/kg
C _a	Concentración alimento	kg/m ³
C _p	Concentración permeado	kg/m ³
C _r	Concentración rechazo	kg/m ³
C _w	Concentración pared	kg/m ³
C _I , C _R	Concentración equivalente de ión en el líquido	eq/m ³
\bar{C}_I, \bar{C}_R	Concentración equivalente de ión en la resina	eq/m ³
D _{AB}	Difusividad del componente en el agua	m ² /s
d _h	Diámetro hidráulico	m
I	Ión a retirar disuelto en el líquido	-
\bar{I}	Ión a retirar adherido en la resina	-
J _s	Flux de soluto	kg/m ² ·s
J _v	Flux volumétrico	m ³ /m ² ·s
k _{perd}	Coefficiente de pérdidas	Pa/(m ³ /s) ²
k _s	Coefficiente transferencia de materia	m/s
L	Longitud del lecho de intercambio iónico	m
L _p	Permeabilidad al agua	m ³ /m ² ·s·Pa
P	Presión	Pa
Q	Caudal volumétrico	m ³ /s
R	Grupo funcional resina disuelto en líquido	-
\bar{R}	Grupo funcional resina adherido en resina	-
Re	Número de Reynolds	Adim.
R _G	Constante universal gases perfectos	J/mol·K
Sc	Número de Schmidt	Adim.
Sh	Número de Sherwood	Adim.
S _{mod}	Sección de paso	m ²
T	Temperatura operación	K
U	Velocidad de paso a través del lecho sin la resina presente	m/s
v	Velocidad de paso	m/s
V	Volumen	m ³

<i>Símbolo</i>	<i>Variable</i>	<i>Unidades SI</i>
x_I, x_R	Fracción iónica equivalente en líquido	Adim.
y_I, y_R	Fracción iónica equivalente en la resina	Adim.
z_I, z_R	Carga del ión	Adim.
α	Factor osmótico	$m^3 \cdot Pa/kg$
ϵ	Porosidad	Adim.
μ	Viscosidad cinemática	$kg/m \cdot s$
π	Presión osmótica	Pa
ν	Viscosidad cinemática	m^2/s

9. BIBLIOGRAFÍA

1. <http://www.colan.org/>
2. R. Morales-Rodríguez, R. Gani, S. Déchelotte, A. Vacher, O. Baudouin; Use of CAPE-OPEN standards in the interoperability between modelling tools (MoT) and process simulators (Simulis® Thermodynamics and ProSimPlus), *Chemical Engineering Research and Design*, 86, 823-833 (2008)
3. J. van Baten; Rapid prototyping of unit operation models using generic tools and CAPE-OPEN, *extended abstract, 6th US CAPE-OPEN conference, AIChE annual meeting, Nashville*, November (2009)
4. J. van Baten, R. Szczepanski; A thermodynamic equilibrium reactor model as a CAPE-OPEN unit operation, *Computers and Chemical Engineering*, 35, 1251-1256 (2011)
5. E. Chilyumova, J. Thöming; Dynamic simulation of rinsing and regeneration networks based on high pressure RO, *Desalination*, 207, 45-58 (2007)
6. <http://www.cocosimulator.org/>
7. W.M. Barrett Jr., J. van Baten, T. Martin; Implementation of the waste reduction (WAR) algorithm utilizing flowsheet monitoring, *Computers and Chemical Engineering*, 35, 2680-2686 (2011)
8. D. Peshev, A. Livingston; OSN Designer, a tool for predicting organic solvent nanofiltration technology performance using Aspen One, MATLAB and CAPE-OPEN, *Chemical Engineering Science*, 104, 975-987 (2013)
9. <http://www.chemstations.eu/es/>
10. J.M. Gozálvarez-Zafrilla, A. Santafé Moros; Apuntes de Modelización, Simulación y Optimización de Procesos Químicos, *Documento del Departamento de Ingeniería Química y Nuclear* (2012)
11. Y. Gao, YN. Wang, W. Li, C. Y. Tang; Characterization of internal and external concentration polarizations during forward osmosis processes, *Desalination*, Volume 338, 65-73 (2014)
12. C. Guria, P. K. Bhattacharya, S. K. Gupta; Multi-objective optimization of reverse osmosis desalination units using different adaptation of the non-dominated sorting genetic algorithm, *Computers and Chemical Engineering*, 29, 1977-1995 (2005)
13. R. W. Rousseau; Handbook of separation process technology, *Ed. Advisory Board*, ISBN 0-471-89558-X (1987)

14. J.M. Gozávez-Zafrilla, A. Santafé Moros; Implementación de modelos de procesos químicos y medioambientales, *Editorial UPV*, Ref. 2003.736 (2003)
15. J. van Baten, H. Kooijman, R. Taylor; Flowsheeting for free with COCO, *CACHE News* (2007)
16. J. van Baten; An introduction to COCO, *extended abstract, 4th US CAPE-OPEN conference, AIChE annual meeting, Saltlake City*, November (2007)
17. <https://www.chemindustry.com/apps/chemicals>
18. http://nshs-science.net/chemistry/common/pdf/R-standard_enthalpy_of_formation.pdf
19. E. H. Oelkers, H. C. Helgeson, E. L. Shock, D. A. Sverjensky, J. W. Johnson, V. A. Pokrovskii; Summay of the apparen standard partial molal Gibbs free energies of formation of aqueous species, minerals and gases at pressures 1 to 5000 bars and temperatures 25 to 1000 °C, *American Institute of Physics and American Chemical Society*, Vol. 24, Nº 4 (1995)
20. F. Germán-Borgogno; Compendia de propiedades. Tabla de entalpía de formación, energía libre de Gibbs y entropía de formación de compuestos inorgánicos, *EIQ, BUI, TQ* (2010)
21. http://www.endesaonline.es/ES/empresas/luz/tarifas_electricas_empresas_baja_tension/a_hora/precios/index.asp
22. <http://www.sigmaaldrich.com/catalog/product/fluka/62108?lang=es®ion=ES>
23. <http://www.freshwatersystems.com/p-3849-filmtec-sw30-2540-sea-water-700-gpd-ro-membrane-element.aspx>
24. <http://www.ppe-pressure-washer-parts.com/product/310S>



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA QUÍMICA

II. ANEXOS A LA MEMORIA

Curso Académico: 2013-14

ÍNDICE ANEXOS A LA MEMORIA

1. Código MATLAB del modelo de disolución-difusión.....	65
2. Validación de la unidad de membrana	69
3. Póster presentado en el CITEM 2014.....	71
4. Abstract del póster presentado.....	73

1. CÓDIGO MATLAB DEL MODELO DE DISOLUCIÓN-DIFUSIÓN

En el código Matlab los procedimientos de resolución de ecuaciones se almacenan en funciones, de manera que una vez se tienen definidos los parámetros del problema este se puede resolver simplemente llamando a dichas funciones. En este apartado de los Anexos se describirá el código de las tres funciones (Membrana, derivadas y perm) que permiten resolver una unidad de ósmosis inversa.

La función principal del código desarrollado es la función Membrana, en la cual se resuelven las ecuaciones diferenciales que definen la unidad a través de la función que incluye MATLAB ode45 para obtener las características del rechazo y posteriormente se aplican las ecuaciones de balance globalizado para obtener las del permeado. Las ecuaciones diferenciales son introducidas en el código a través de una llamada a la función derivadas.

Por lo tanto, una vez se conocen las características del alimento y se ha fijado la presión que aporta la bomba y presión de salida del permeado, basta con llamar a la función Membrana para obtener las características de las dos corrientes de salida.

FUNCIÓN MEMBRANA

```
function [Z Wr Wp W Csup Cp] = Membrana(We, Pbomba, Pp)

global L_et n_et

Q_et=We(1)/n_et;

X0=[Q_et We(2) We(3)+Pbomba];

[Z,W] = ode45(@derivadas,[0 L_et],X0);

iter=length(W);
Wr=W(iter,:);
Wr(3)=Wr(3)+Pp;
Wr(1)=Wr(1)*n_et;

Wp(1)=(We(1))-Wr(1);
Wp(2)=(We(1)*We(2)-Wr(1)*Wr(2))/Wp(1);
Wp(3)=Pp;

end

end
```

La función derivadas contiene las ecuaciones diferenciales que definen el comportamiento de la membrana. Dado que para definir dichas ecuaciones es necesario conocer el flux volumétrico y de soluto, al inicio de la función se hace una llamada a la función perm para obtener estos dos parámetros.

FUNCIÓN DERIVADAS

```
function D = derivadas(z,W)

%W=[Q C difP]
global A_mod S_mod L_mod k_perd

[~, Jv Js, ~] = perm(W(1),W(2),W(3));
dAdV=A_mod/(S_mod*L_mod);

D = zeros(3,1);    % a column vector
D(1) = -Jv*dAdV*S_mod;
D(2) = ((Jv*W(2)-Js)/(W(1)/S_mod))*dAdV;
D(3) = -k_perd*(W(1)/S_mod)^2;

end
```

Para finalizar con las funciones que permiten el cálculo de la unidad de ósmosis inversa, se tiene una función, llamada perm, en la cual se han codificado las ecuaciones que permiten el cálculo de los fluxes volumétrico y de soluto. Mediante el procedimiento de cálculo descrito en el apartado 4 de la Memoria, se obtienen ambos fluxes a partir de los datos de la alimentación y presión aplicada.

FUNCIÓN PERM

```
function [Cp Jv Js Cw] = perm(Q,C,difP)

global Lp B T S_mod

%Para hallar el coeficiente de transferencia de materia ks se
efectúan los
%siguientes cálculos:

%viscosidad dinámica (Pa*s)
visc_din=1.234e-06*exp(0.00212*C+(1965/T));

%densidad (kg/m3)
m=1.0069-2.757e-04*(T-273);
dens=498.4*m+sqrt(248000*m^2+752.4*m*C);

%viscosidad cinemática (m2/s)
visc_cin=visc_din/dens;

%mass diffusivity (m2/s)
D_AB=6.725e-06*exp(0.1546e-03*C-(2513/T));

%Diámetro hidráulico
dh=0.9e-03;    %(m)

%Número adimensional Sc:
Sc=visc_cin/D_AB;
```

```

%Número adimensional Reynolds:
Qm3s=Q/86400;
Re=dh*(Qm3s/S_mod)/visc_cin;

%Número adimensional Sh para un módulo de espira enrollada:
Sh=0.065*Re^0.865*Sc^0.25;

%Sh=ks*dh/D_AB

%Coeficiente de transferencia de materia
ks=Sh*D_AB/dh;
ks=ks*86400;

%Ahora que ya tenemos la ks, vamos a hallar el flux de agua
considerando
%el efecto de polarización

%b_pi
pi=0.7949*C-0.0021*C^2+7e-05*C^3-6e-07*C^4;
b_pi=pi/C;

%Inicialización para el cálculo de Cw, Cp y Jv;
Cw=C*1.5;
itermax=2000;
beta=0.1;
epsilon=1e-05;
for i=1:itermax
    a=-(B/Lp)*Cw;
    b=b_pi*Cw-difP-(B/Lp);
    c=b_pi;
    Cp=(4*a*c)/(2*c*(b-sqrt(b^2-4*a*c)));
    Jv=Lp*(difP-b_pi*(Cw-Cp));
    Cw_cal=Cp+(C-Cp)*exp(Jv/ks);
    if abs((Cw-Cw_cal)/Cw)<epsilon, break, end
    Cw=Cw_cal*beta+(1-beta)*Cw;
end

%Flux de soluto:

Js=Jv*Cp;

end

```

Además de las funciones donde se codifica el proceso de cálculo que permite obtener las características de las corrientes de salida de la unidad, se necesita un fichero donde se definan todos los parámetros que caracterizan la misma. En el script datos_membrana se almacenan estos parámetros en variables globalizadas, de manera que una vez definidos en el script pueden ser utilizados en las funciones de cálculo descritas anteriormente.

SCRIPT DATOS_MEMBRANA

```

global T
T=298; %K

global Rg
Rg = 8.314; %J/(mol*K)

%Datos geométricos etapa
global n_mod n_et A_mod S_mod L_mod L_et
n_mod=7; %Nº módulos por tubo
n_et=10; % nº de carcacas
A_mod=8; %m2 Área de un modulo
S_mod=24*10^-4; %m2 sección de paso de un módulo
L_mod=1; %longitud de un módulo

L_et=n_mod*L_mod;

%Datos de la membrana
global Lp B k_perd
Lp =0.023; %en m3/(m2*día*bar) Permeabilidad de la membrana
B=6e-03; %Permeabilidad al soluto (kg·m-2·día-1)
k_perd=3.8e-11; %bar/(m/día)^2

%Datos del sistema
global M_NaCl n_ion alpha
M_NaCl = 58.5; %g/mol
n_ion = 2;
alpha=(n_ion/M_NaCl)*Rg*T*10^-2;

%Datos de la bomba
global Ps
Ps = 40;

```


2. VALIDACIÓN DE LA UNIDAD DE MEMBRANA

Para realizar la implementación de la unidad de cálculo en el simulador de procesos COCO se realizó un desarrollo progresivo del modelo de dicha unidad siguiendo estos pasos:

1. Desarrollo preliminar del modelo de disolución-difusión en MATHCAD. Este primer paso consiste en implementar las ecuaciones diferenciales del modelo en una función de MATHCAD con el objetivo de obtener una idea general de la estructura que debe tener el código del modelo.
2. Desarrollo del modelo de disolución-difusión en MATLAB. Tras definir la estructura básica del código en su diseño preliminar, este se debe implementar en MATLAB con una estructura modular similar a la que tendrá la unidad de cálculo. Dicha implementación se realiza a través de la combinación de varios ficheros: función perm que permite el cálculo de los flujos J_V y J_S , función derivadas que contiene las ecuaciones diferenciales que definen el modelo, script datos_membrana donde se indican los diferentes parámetros requeridos por el modelo y función Membrana donde a través de la combinación de los ficheros anteriores y una función de resolución de ecuaciones diferenciales (ode45) se resuelve la etapa de membrana.
3. Introducción del modelo en el simulador COCO. Además de cargar los scripts perm, derivadas y Membrana en la unidad MATLAB, es necesario introducir las líneas de código que permitan la interacción con las corrientes de entrada y salida de la unidad. Una vez realizado todo esto el modelo queda almacenado en el fichero Membrana.mum para que pueda ser cargado rápidamente en aplicaciones posteriores.

Con la intención de poder validar la unidad de cálculo una vez implementada esta en el simulador, en cada uno de los pasos del desarrollo del modelo se resolvió un caso base para poder comparar así los resultados obtenidos en cada programa posteriormente. Dicho caso base queda definido por los siguientes datos de entrada: corriente de alimento de $100 \text{ m}^3/\text{d}$ con una concentración de $35 \text{ kg}/\text{m}^3$ de NaCl, aportando la bomba una presión de 40 bares.

Tabla 11. Resultados obtenidos en cada programa en la resolución del caso base

<i>Variable</i>	<i>MATHCAD</i>	<i>MATLAB</i>	<i>COCO</i>
Caudal rechazo (m^3/d)	58.65	58.65	53.32
Concentración rechazo (kg/m^3)	57.18	57.18	62.28
Presión rechazo (bar)	40.99	40.99	41.01
Caudal permeado (m^3/d)	41.34	41.34	46.59
Concentración permeado (kg/m^3)	3.53	3.52	3.28
Presión permeado (bar)	1.00	1.00	1.00

Los resultados obtenidos en el simulador COCO difieren ligeramente a los obtenidos en los programas de cálculo MATLAB y MATHCAD. Sin embargo esto no se debe a un error en la introducción del modelo en el simulador, sino a que los programas de cálculo trabajan a doble tolerancia y por tanto el error final de los cálculos es menor. En la medida de lo posible se ha tratado de transformar las expresiones matemáticas del modelo para así evitar expresiones donde el error varíe bruscamente en los cálculos iterativos, de manera que el efecto de no tener doble tolerancia en el simulador se reduzca lo máximo posible.

En definitiva, puede afirmarse que los resultados proporcionados por la unidad de cálculo del simulador consiguen reproducir el comportamiento de una etapa de membrana, si bien la precisión con la que lo hacen es algo menor que si se utilizara un programa de cálculo matemático. Aun así, no se considera que esta pequeña imprecisión afecte de manera importante a los resultados y por tanto se tomará como válida la unidad de cálculo.

3. PÓSTER PRESENTADO EN EL CITEM 2014

Implementation of membrane models on a CAPE-OPEN tool to simulate a process including RO membranes



José M. Gozávez-Zafrilla^{1,2}, Asunción Santafé-Moros^{1,2},
Miguel Sanchis-Sebastià², Joaquín Gomis-Fons²



¹Universitat Politècnica de València. Instituto de Seguridad Industrial Radiológica y Medioambiental (ISIRYM)

²Universitat Politècnica de València. Departamento de Ingeniería Química y Nuclear (DIQN)

Introduction

Process simulators are a useful tool for evaluating different configurations of chemical processes and developing new ones. This software includes many standard units like reactor or distillation towers. However, unit separations like membranes can be very material dependent and they are not usually included in the simulators.

Aim → Implement a RO membrane unit in the COCO environment, which is a free process simulator, as a part of a bigger process in which there are different recycle streams.

Modelling

The RO model is based on the solution-diffusion equations, which have to be solved by an iterative process as the osmotic pressure is taken into account. Moreover, it has also been taken into account polarization effects and pressure losses.

In order to use the model, the following parameters must be specified: solution-diffusion parameters, number of vessels, number of modules per vessel and geometrical characteristics of the module.

The model was developed in MATLAB and the units interchange data with the COFE environment (Fig. 1). Alternatively, the model can be translated into Scilab which is a freeware that can also be interconnected with COCO.

As some ions were not included in the COCO component database, they were introduced using the TEA server provided with the simulator.

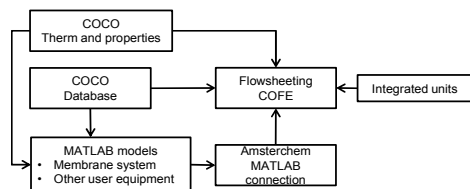


Fig. 1. Communication between models in the COCO environment

Case of study

A rinsing process in which nickel plating waste water is regenerated by the use of a RO membrane unit was studied. The process consists of:

- A rinsing tank cascade using operating in countercurrent mode.
- A RO membrane unit (Filmtec SW30-2540) with one vessel containing six modules in series.
- An ionic exchange unit to achieve the rinsing criterion ($c < 0.21$ g/L)

The simulated case of study process can be seen in Fig. 2. Two different strategies were carried out to analyze the performance of this process:

- Modifying the tank position in which the membrane permeate is recycled to the rinsing cascade.
- Introducing a second membrane unit to replace the ionic exchange one.

For the second strategy, it was necessary to optimize the flow split factor and the pump pressure in order to find the operating conditions that allow the process to meet the quality requirements without using the ionic exchange.

Acknowledgements

The Spanish Ministry of Economy and Competitiveness is kindly acknowledged (Project CTM 2010-20248).

Conclusions

- A membrane unit was successfully implemented in the COCO environment by using MATLAB models.
- COCO simulator is a free alternative way to find improvements for an industrial process rather than using other process simulators or carrying out pilot plant tests.
- Using the COCO simulator the performance of two alternative process using RO regeneration could be compared.

Results

The best recycling position for the permeate stream is shown in Fig.2.

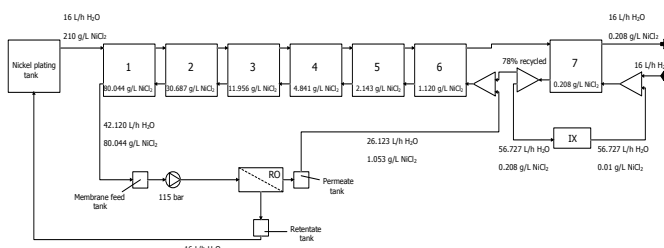


Fig. 2 Operating conditions of the case of study

As it can be seen in Fig. 3, the higher the input position in the tank cascade, the smaller the output concentration of the last tank, what reduces the operating cost of the ionic exchange.

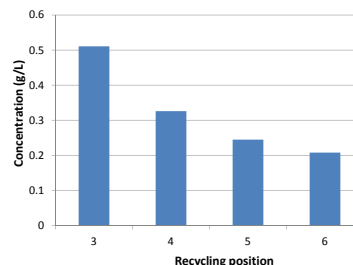


Fig. 3 Concentration of the output stream for different recycling positions

Fig. 4 shows the operating conditions for the process using two RO units:

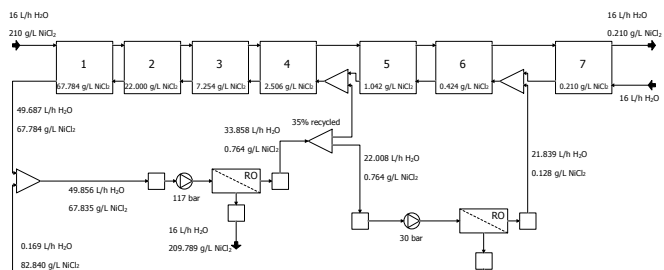


Fig. 4 Operating conditions of the alternative process

Comparing the alternative process with the initial one:

- Membrane units are doubled.
- It is required an additional pump and additional storage tanks.
- Energetic costs are 16% higher.
- The ionic exchange unit is suppressed. Since the ionic exchange unit is the critical unit from the economic point of view, the inclusion of the second membrane unit leads to an important total cost reduction (~ 30%).



4. ABSTRACT DEL PÓSTER PRESENTADO



IX IBERO-AMERICAN CONGRESS ON MEMBRANE SCIENCE AND TECHNOLOGY

Santander- Spain, May 25-28, 2014



BOOK OF ABSTRACTS

CITEM 2014 – IX Ibero-American Congress on Membrane Science and Technology
Book of abstracts

Editors: Alfredo Ortiz and Daniel Gorri

ISBN: 978-84-697-0397-7

Depósito legal: SA-299-2014

IMPLEMENTATION OF MEMBRANE MODELS ON A CAPE-OPEN TOOL TO SIMULATE A PROCESS INCLUDING REVERSE OSMOSIS STAGES

José M. Gozávez-Zafrilla^{1,2}, Asunción Santafé-Moros^{1,2}, Miguel Sanchis-Sebastià², Joaquín Gomis-Fons²

¹*Universitat Politècnica de Valencia. Instituto de Seguridad Industrial Radiológica y Medioambiental (ISIRYM), jmgz@iqn.upv.es*

²*Universitat Politècnica de Valencia. Departamento de Ingeniería Química y Nuclear (DIQN)*

Process simulators are a useful tool for evaluating different configurations of chemical processes and developing new ones. Process simulators include many standard units (equilibrium reactors, distillation towers, etc.) that the engineer can parameterize adapting them to their own process. Usually, the engineer does not need to include physical or thermodynamic data when the compounds are included in the simulator databank. Besides, the simulator includes several thermodynamic models to obtain the properties of mixtures or to perform flash calculations. However, some unit operations are very specific and very material dependent, so there are serious difficulties to define them as general units to be parameterized. That is the case of membrane processes. To face the problem, simulators include the possibility to define user units that liable to be integrated in the process flowsheet to interact with the units integrated in the simulator. To do so, it is necessary to know how the stream information is defined, to correctly interchange information through the unit ports. In a previous work [1], we showed how to implement a simple membrane process model in CHEMCAD simulator using the VisualBasic interface. However, many engineers are not familiar with Visual Basic; and CHEMCAD is a commercial software that cannot be affordable for small applications [2].

In this work, we show the possibility to perform calculations of processes including RO membranes using the freeware environment COCO with models developed in MATLAB. COCO software is based on computer aided process engineering open (CAPE-OPEN) that is a standard for communication between software components. In the COCO environment, the flowsheeting is performed by COFE. The model of the user units is developed in MATLAB Matlab (or alternatively in Scilab which is also freeware) and the units interchange data with the COFE environment (Fig. 1). The communication is possible thanks to software applications developed by AmsterCHEM [3].

As a case of study, we developed a RO unit to be used in the rinsing process, whose COFE flowsheet we show in Fig. 2. In this process, the RO membrane unit recycles water with low salt concentration to the last tank unit where solids coming from the previous tank are rinsed.

The membrane model includes the solution-diffusion model equations to explain volumetric flux and solute permeation as function of pressure, concentration. An iteration is performed to take into account the effects of the difference of osmotic pressure developed on the permeate flux and to account for polarization effects. These equations, valid for a local position of the membrane, are integrated together with a model for pressure charge to obtain the performance of a vessel unit containing several modules in series. The possibility to use robust solvers and many mathematical procedures constitutes one of the strong points of the MATLAB programming. It is important to mention that we must check if the components used are defined, should not be this case, they must be included in the property database using the TEA server provided with COCO. Finally, the membrane stage unit is defined by including the number of vessels and modules per vessel as parameters, together with solution diffusion parameters and other necessary parameters that can be experimentally obtained for any specific membrane.

Once the membrane model is included any typical process simulator capabilities can be used in the COFE environment. As an example, we show a parametric study of the effect of the pressure used in the process on the concentration of the final tank.

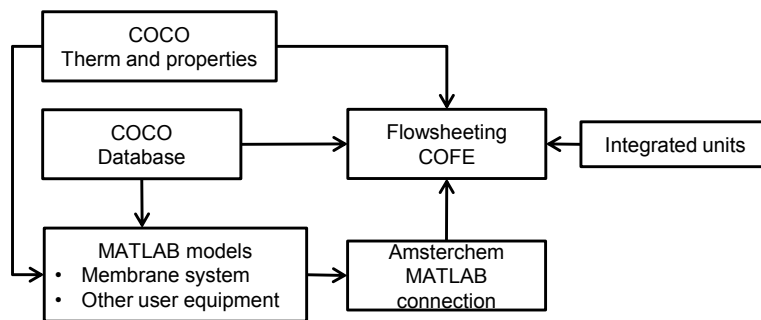


Fig. 1 Communication between models in the COCO environment.

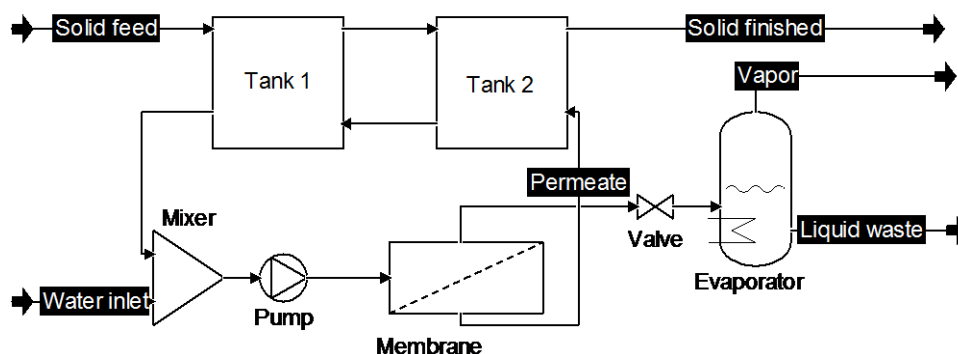
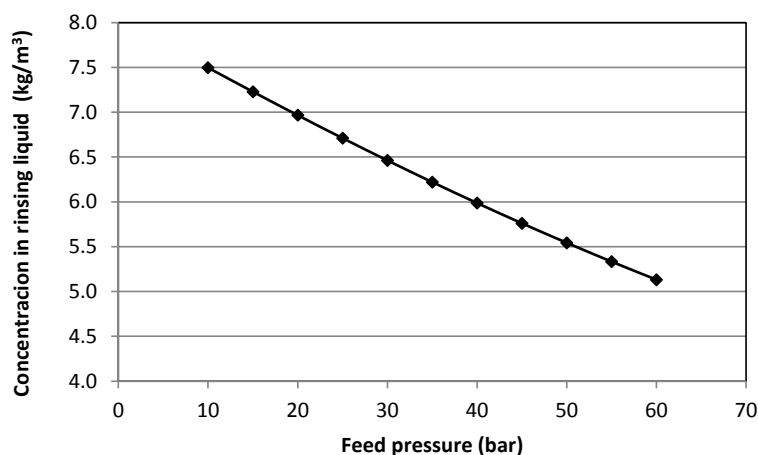


Fig. 2 Flowsheet of rinsing process coupled with membrane for water recycling.



References

- [1] J.M. Gozávez-Zafrilla, B. Gómez-Martínez, A. Santafé Moros; Implementación de unidades de cálculo de membrana en simuladores de procesos, Ing. Quim (Madrid, Spain), 422, 127-130 (2005)
- [2] D. Peshev, A.G. Livingston; OSN Designer, a tool for predicting organic solvent nanofiltration technology performance using Aspen One, MATLAB and CAPE OPEN, Chem. Eng. Sci., 104, 975-987 (2013)
- [3] J.M. van Baten, Rapid prototyping of unit operation models using generic tools and CAPE-OPEN, extended abstract, 6th US CAPE-OPEN conference, AIChE annual meeting, Nashville, November (2009)

Acknowledgements

The Spanish Ministry for Science and Innovation (Project OPTIMEM CTM2010-20248) is kindly acknowledged.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA QUÍMICA

III. PLIEGO DE CONDICIONES

Curso Académico: 2013-14

ÍNDICE DEL PLIEGO DE CONDICIONES

1. Condiciones legales del software utilizado	79
1.1. Simulador COCO.....	79
1.2. Unidad AmsterCHEM	79
1.3. MATLAB.....	79
1.4. MATHCAD.....	80
2. Requisitos técnicos requeridos por el software empleado.....	81
2.1. Unidad AmsterCHEM	81
2.2. MATLAB.....	81
2.3. MATHCAD.....	81
3. Plazos.....	82

1. CONDICIONES LEGALES DEL SOFTWARE UTILIZADO

1.1. Simulador COCO

Las condiciones legales que rigen el uso del simulador de procesos COCO pueden encontrarse en el apartado designado a tal efecto en la página web del simulador, siendo el enlace de dicha página el siguiente: http://www.cocosimulator.org/index_disclaimer.html.

El simulador COCO es freeware y por tanto las restricciones legales que le son aplicables son escasas. En cuanto a su uso no existe ninguna restricción relevante, puesto que el simulador puede utilizarse y distribuirse libremente, si bien la distribución queda restringida por el hecho de que solo puede distribuirse el programa completo sin realizar modificaciones sobre él y sin cobrar nada por ello.

En consecuencia, el trabajo desarrollado con este simulador de procesos podría llegar a comercializarse sin que ello suponga una violación de los términos de la licencia.

1.2. Unidad AmsterCHEM

Las restricciones legales de la licencia de la unidad MATLAB de AmsterCHEM pueden encontrarse en la página web de esta empresa en el siguiente enlace: <http://www.amsterchem.com/matlabunitophelp.php?page=disclaimer.htm>.

Básicamente las condiciones legales establecidas en esta licencia son las mismas que en el caso del simulador COCO, con la excepción de que no se puede hacer un uso comercial de los trabajos desarrollados con esta unidad de proceso.

En caso de que se quiera dar un uso comercial a la unidad MATLAB es necesario registrar el producto enviando un correo a la dirección info@amsterchem.com y pagando una cuota de 400€. Aunque el registro es unipersonal, la persona registrada puede instalar su producto en tantos ordenadores como desee.

De esta manera, mientras la unidad de cálculo desarrollada tenga un uso estrictamente académico esta puede utilizarse y distribuirse libremente. Sin embargo, si se sigue desarrollando hasta obtener una unidad con aplicación comercial sería necesario realizar el registro correspondiente para poder comercializarla.

1.3. MATLAB

Las condiciones legales aplicables al uso del programa MATLAB pueden encontrarse en el fichero "license.txt" que se encuentra en el directorio que se crea durante la instalación del programa. Este documento no está disponible online, pero en la página de la empresa MathWorks (<http://www.mathworks.es/>) disponen de un enlace de contacto donde podría contactarse con la empresa para solicitar el documento.

Las restricciones en el uso del programa son diferentes en función del tipo de licencia de que se disponga: comercial, académica o estudiante. Aunque el proyecto tiene carácter académico, la unidad de cálculo podría seguir desarrollándose hasta llegar a tener aplicación comercial. Por lo tanto, se considerará que las condiciones que en este caso aplican son las de la licencia comercial, ya que de esta forma se deja abierta la posibilidad de continuar desarrollando el trabajo realizado para obtener una aplicación que pueda comercializarse.

Además, las restricciones en el uso también varían en función del tipo de activación que se elija. A continuación se detallan las principales restricciones para cada uno de los tipos de activación que se puede elegir al adquirir una licencia comercial individual:

- *Standalone Named User*: el programa solo puede ser utilizado por el usuario designado, el cual puede utilizarlo en varios ordenadores diferentes. Aunque la ubicación física de los ordenadores no está restringida, el usuario designado solo podrá utilizar el programa en dos terminales simultáneamente. En caso de que sea necesario un registro para acceder al terminal con el programa, el usuario designado no podrá compartir las claves con ningún otro individuo.
- *Designated Computer*: el programa solo puede ser utilizado en el ordenador designado, debiéndose ejecutarse desde la consola de dicho terminal por no más de un usuario a la vez. Es posible redesignar el ordenador, pero ello no puede hacerse más de 4 veces en un período inferior a 12 meses.

1.4. MATHCAD

Las limitaciones legales en el uso del programa MATHCAD se indican al inicio del proceso de instalación de dicho programa (momento en que deben ser aceptadas por el usuario). Posteriormente a la instalación, estas limitaciones pueden consultarse en el menú Help>About Mathcad del programa.

En las condiciones legales de la licencia comercial se indican todas las patentes y marcas de PTC relacionadas con el programa que no pueden violarse durante el uso de este, pero también se indican todas aquellas licencias de otras compañías que deben ser respetadas así como las fuentes donde consultar dichas licencias.

Así mismo, se establecen las restricciones habituales en el uso de una licencia comercial:

- El programa no puede ser copiado ni distribuido por ningún medio.
- No puede cederse el uso del programa a terceras personas.
- El programa no puede utilizarse de forma diferente a la cual se establece en la licencia, a no ser que se disponga de la aprobación escrita de PTC.
- Los manuales de usuario y documentos de iniciación están restringidos por las leyes de copyright de los Estados Unidos.

2. REQUISITOS TÉCNICOS REQUERIDOS POR EL SOFTWARE EMPLEADO

2.1. Unidad AmsterCHEM

En la página web que AmsterCHEM dedica a la unidad MATLAB hay un apartado de requisitos donde se exigen los siguientes requisitos técnicos para poder utilizar este software:

1. La unidad de operación MATLAB funciona con cualquier entorno de simulación CAPE-OPEN que soporte la versión 1.1 termodinámica, entre los cuales se incluye COCO simulator.
2. Debe tenerse instalada una versión de MATLAB 5.3 o superior de 32-bits o bien una versión 7.1 o superior en el caso de 64-bits.

Si se desea confirmar esta información puede hacerse en el siguiente enlace: <http://www.amsterchem.com/matlabunitop.html>.

2.2. MATLAB

En la página web de la empresa MathWorks se indica que los requisitos técnicos que exige la versión R2014a de MATLAB son los siguientes:

1. Sistema operativo: Windows XP o superior, Windows Server 2003 o superior.
2. Procesador: cualquier procesador Intel o AMD x86 que soporte el conjunto de instrucciones SSE2.
3. Espacio en disco: 1 GB si únicamente se instala MATLAB, 3-4 GB para una instalación típica.
4. RAM: mínimo 1024 MB pero se recomienda por lo menos 2048 MB.

En caso de querer consultar estos requisitos en la fuente de origen puede hacerse a través del siguiente enlace: http://www.mathworks.es/support/sysreq/current_release/.

2.3. MATHCAD

En la web de la empresa PTC no se ha conseguido encontrar los requisitos técnicos de la versión MATHCAD 14, que es la que se emplea en el proyecto, pero sí los correspondientes a la versión 13.1. Dichos requisitos puede consultarse en la siguiente dirección web: <http://www.ptc.com/partners/hardware/current/support/mathcad13.htm>.

Ante esta situación, se ha preferido detallar los requisitos de la versión 13.1, los cuales serán muy similares o incluso iguales que los de la versión 14, a buscar los requisitos de la versión utilizada en fuentes de menor fiabilidad.

Tabla 12. Requisitos técnicos MATHCAD 13.1

<i>Característica</i>	<i>Mínimo</i>	<i>Recomendado</i>
Memoria principal	256 MB	512 MB o superior
Velocidad CPU	400 MHz	700 MHz o superior
Espacio disponible en disco	550 MB	
.NET Framework	1.1.4322	
Componentes acceso datos Microsoft (MDAC)	2.6	
Explorador internet	Microsoft Internet Explorer 6.0	
Monitor	Resolución de 1024x768 (o superior) con color de 24-bit o superior	
Red	TCP/IP Adaptador para conexión Ethernet	
Puntero	Ratón compatible con Microsoft u otro dispositivo puntero	
Miscelánea	Puerto CD-ROM o DVD	

3. PLAZOS

- Desde el inicio del proyecto se establece un plazo máximo de 2 semanas para seleccionar el simulador con que se abordará el desarrollo del mismo.
- Las ecuaciones matemáticas que definen el comportamiento de la unidad de membrana, incluyendo el efecto de la concentración por polarización, se establecerán en un plazo de 1 semana y, una vez definidas, se implementarán en MATLAB en no más de 6 días.
- Una vez se tiene el modelo MATLAB, en un plazo máximo de 4 días se implementará dicho modelo en el simulador COCO, incluyendo la validación de la unidad de cálculo.
- El análisis del proceso de regeneración de aguas industriales (simulación y mejora) se realizará en un plazo de hasta 3 semanas, debiéndose realizar en dicho plazo tanto la justificación técnica como económica de la mejora planteada.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA QUÍMICA

IV. PRESUPUESTO

Curso Académico: 2013-14

ÍNDICE DEL PRESUPUESTO

1. Presupuesto de ejecución por contrata	87
2. Presupuesto de ejecución material por capítulos	89
3. Estado de mediciones.....	91
4. Precios descompuestos	95
5. Justificación de los precios unitarios	99

1. PRESUPUESTO DE EJECUCIÓN POR CONTRATA

Capítulo	Importe(€)
Capítulo 1 Selección del simulador de procesos	2,452.25
Capítulo 2 Modelado de las unidades de separación	3,608.58
Capítulo 3 Simulación de un proceso industrial mediante COCO	3,935.64
Presupuesto de ejecución material	9,996.47
5% de gastos generales	499.82
Suma	10,496.29
21% IVA	2,204.22
Presupuesto de ejecución por contrata	12,700.51

2. PRESUPUESTO DE EJECUCIÓN MATERIAL POR CAPÍTULOS

Presupuesto parcial nº 1 Selección del simulador de procesos

Num.	Código	Ud	Denominación	Cantidad	Precio (€)	Total (€)
1.1	Hor1	h	Hora de trabajo dedicada a explorar los distintos simuladores de procesos existentes en el mercado	67.000	33.34	2,233.78
1.2	Hor2	h	Hora de trabajo dedicada a seleccionar el simulador de procesos que se utilizará de entre aquellos cuyo uso es viable en el proyecto	7.000	31.21	218.47
Total presupuesto parcial nº 1 Selección del simulador de procesos :						2,452.25

Presupuesto parcial nº 2 Modelado de las unidades de separación

Num.	Código	Ud	Denominación	Cantidad	Precio (€)	Total (€)
2.1	Mod1	ud	Codificación de un modelo que represente el comportamiento de una unidad de ósmosis inversa. compuesta por varios tubos arrollados en espiral con diversos módulos cada uno de ellos. en lenguaje MATLAB	1.000	1.100.87	1,100.87
2.2	Mod2	ud	Codificación de un modelo que represente el comportamiento de un tanque de lavado en lenguaje MATLAB	1.000	109.30	109.30
2.3	Mod3	ud	Codificación de un modelo que represente el comportamiento de una unidad de intercambio iónico en lenguaje MATLAB	1.000	87.44	87.44
2.4	Hor3	h	Hora de trabajo dedicada a la obtención de las ecuaciones matemáticas y el procedimiento de cálculo que permiten representar el comportamiento de una unidad de ósmosis inversa	40.000	42.94	1,717.60

2.5	Hor4	h	Hora de trabajo dedicada a la obtención de las ecuaciones matemáticas y el procedimiento de cálculo que permiten representar el comportamiento de un tanque de lavado	8.000	31.23	249.84
2.6	Hor5	h	Hora de trabajo dedicada a la obtención de las ecuaciones matemáticas y el procedimiento de cálculo que permiten representar el comportamiento de una unidad de intercambio iónico	11.000	31.23	343.53
Total presupuesto parcial nº 2 Modelado de las unidades de separación:						3,808.58

Presupuesto parcial nº 3 Simulación de un proceso industrial mediante COCO

Num.	Código	Ud	Denominación	Cantidad	Precio (€)	Total (€)
3.1	Hor6	h	Hora de trabajo dedicada a la implementación del modelo MATLAB en un unidad de proceso personalizada del simulador COCO	33.000	20.86	688.38
3.2	Hor7	h	Hora de trabajo dedicada al estudio de un proceso de regeneración de aguas industriales en el cual aparezcan unidades de ósmosis inversa	79.000	35.94	2,839.26
3.3	Lic3	ud	Licencia que permite el uso comercial de la unidad personalizada MATLAB de Amsterchem en el simulador de procesos COCO	1.000	408.00	408.00
Total presupuesto parcial nº 3 Simulación de un proceso industrial mediante COCO :						3,935.64

3. ESTADO DE MEDICIONES

Presupuesto parcial nº 1 Selección del simulador de procesos

Nº	Ud	Descripción	Medición	
			Parcial	Subtotal
1.1	h	Hora de trabajo dedicada a explorar los distintos simuladores de procesos existentes en el mercado		
		Horas		
		Identificación de los diferentes simuladores de procesos disponibles	5	
		Análisis de las prestaciones de cada simulador	31	
		Exploración de la posibilidad de introducir unidades personalizadas	17	
		Determinación del coste de una licencia de cada simulador viable	5	
		Introducción de una unidad personalizada sencilla en cada simulador viable	9	
			67	67
		Total h.....:		67
1.2	h	Hora de trabajo dedicada a seleccionar el simulador de procesos que se utilizará de entre aquellos cuyo uso es viable en el proyecto		
		Horas		
		Establecimiento de los criterios de selección	5	
		Valoración de cada alternativa en base a cada uno de los criterios	1	
		Elaboración de una matriz de selección	1	
			7	7
		Total h.....:		7

Presupuesto parcial nº 2 Modelado de una unidad de ósmosis inversa

Nº	Ud	Descripción	Medición	
2.1	Ud	Codificación de un modelo que represente el comportamiento de una unidad de ósmosis inversa. compuesta por varios tubos arrollados en espiral con diversos módulos cada uno de ellos. en lenguaje MATLAB.		
		Uds.	Parcial	Subtotal
		Modelo de ósmosis inversa	1	
			1	
		Total ud.....:		1
2.2	Ud	Codificación de un modelo que represente el comportamiento de un tanque de lavado en lenguaje MATLAB.		
		Uds.	Parcial	Subtotal
		Modelo tanque de lavado	1	
			1	1
		Total ud.....:		1
2.3	Ud	Codificación de un modelo que represente el comportamiento de una unidad de intercambio iónico en lenguaje MATLAB.		
		Uds.	Parcial	Subtotal
		Modelo intercambio iónico	1	
			1	1
		Total ud.....:		1
2.4	h	Hora de trabajo dedicada a la obtención de las ecuaciones matemáticas y el procedimiento de cálculo que permiten representar el comportamiento de una unidad de ósmosis inversa		
		Horas	Parcial	Subtotal
		Establecer las ecuaciones diferenciales del modelo	2	
		Establecer las ecuaciones del modelo de disolución-difusión	2	
		Introducción del efecto de la polarización en el modelo de disolución-difusión	11	
		Establecer el procedimiento de cálculo para la resolución de las ecuaciones	20	

	Transformación de las expresiones para suavizar la evolución del error	5	5	
			40	40
			Total h.....:	40
2.4	h	Hora de trabajo dedicada a la obtención de las ecuaciones matemáticas y el procedimiento de cálculo que permiten representar el comportamiento de un tanque de lavado		
		Horas	Parcial	Subtotal
	Búsqueda bibliográfica sobre los procesos de enjuague	4	4	
	Planteamiento de las ecuaciones a codificar	4	4	
			8	8
			Total h.....:	8
2.4	h	Hora de trabajo dedicada a la obtención de las ecuaciones matemáticas y el procedimiento de cálculo que permiten representar el comportamiento de una unidad de intercambio iónico		
		Horas	Parcial	Subtotal
	Búsqueda bibliográfica sobre el comportamiento de unidades de intercambio iónico	4	4	
	Planteamiento de las ecuaciones del modelo	4	4	
	Adaptación de las ecuaciones a las limitaciones del simulador	1	1	
			11	11
			Total h.....:	11

Presupuesto parcial nº 3 Simulación de un proceso industrial mediante COCO

Nº	Ud	Descripción	Medición	
3.1	h	Hora de trabajo dedicada a la implementación del modelo MATLAB en un unidad de proceso personalizada del simulador COCO		
			Horas	Parcial Subtotal
		Introducción del modelo de ósmosis inversa en una unidad MATLAB	6	6
		Pruebas de la unidad de cálculo para depurar errores	24	24
		Validación de la unidad de cálculo mediante la resolución de un caso base	3	3
				33 33
			Total h.....:	33
3.2	h	Hora de trabajo dedicada al estudio de un proceso de regeneración de aguas industriales en el cual aparezcan unidades de ósmosis inversa		
			Horas	Parcial Subtotal
		Selección del proceso industrial y recopilación de parámetros	13	13
		Simulación del proceso seleccionado en COCO	15	15
		Revisión de los resultados obtenidos en la simulación	2	2
		Rediseño del proceso para conseguir una reducción en su coste	9	9
		Simulación y optimización del proceso rediseñado	35	35
		Comparación de los resultados de ambas simulaciones y justificación de la mejora	4	4
				79 79
			Total h.....:	79
3.3	Ud	Licencia que permite el uso comercial de la unidad personalizada MATLAB de Amsterchem en el simulador de procesos COCO		
			Total ud.....:	1

4. PRECIOS DESCOMPUESTOS

Nº	Código	Ud	Descripción			Total
1 Selección del simulador de procesos						
1.1	Hor1	h	Hora de trabajo dedicada a explorar los distintos simuladores de procesos existentes en el mercado			
	Maq1	1.000	h	Ordenador (CPU. pantalla. teclado. etc.)	0.049	0.05
	MO1	1.000	h	Graduado en GIQ	30.000	30.00
	MO2	0.100	h	Programador	20.000	2.00
	%	2.000	%	Costes Directos Complementarios	32.050	0.64
		2.000	%	Costes indirectos	32.690	0.65
					Precio total por h.	33.34
1.2	Hor2	h	Hora de trabajo dedicada a seleccionar el simulador de procesos que se utilizará de entre aquellos cuyo uso es viable en el proyecto			
	MO1	1.000	h	Graduado en GIQ	30.000	30.00
	%	2.000	%	Costes Directos Complementarios	30.000	0.60
		2.000	%	Costes indirectos	30.600	0.61
					Precio total por h.	31.21
2 Modelado de las unidades de separación						
2.1	Mod1	ud	Codificación de un modelo que represente el comportamiento de una unidad de ósmosis inversa. compuesta por varios tubos arrollados en espiral con diversos módulos cada uno de ellos. en lenguaje MATLAB			
	Lic1	10.000	h	Licencia MATHCAD	0.124	1.24
	Lic2	36.000	h	Licencia MATLAB	0.962	34.63
	Maq1	46.000	h	Ordenador (CPU. pantalla. teclado. etc.)	0.049	2.25
	MO1	10.000	h	Graduado en GIQ	30.000	300.00
	MO2	36.000	h	Programador	20.000	720.00
	%	2.000	%	Costes Directos Complementarios	1,058.120	21.16
		2.000	%	Costes indirectos	1,079.280	21.59
					Precio total por ud.	1,100.87
2.2	Mod2	ud	Codificación de un modelo que represente el comportamiento de un tanque de lavado en lenguaje MATLAB			
	Lic2	5.000	h	Licencia MATLAB	0.962	4.81
	Maq1	5.000	h	Ordenador (CPU. pantalla. teclado. etc.)	0.049	0.25
	MO2	5.000	h	Programador	20.000	100.00
	%	2.000	%	Costes Directos Complementarios	105.060	2.10
		2.000	%	Costes indirectos	107.160	2.14
					Precio total por ud.	108.30

Nº	Código	Ud	Descripción		Total
	2.3 Mod3	ud	Codificación de un modelo que represente el comportamiento de una unidad de intercambio iónico en lenguaje MATLAB		
	Lic2		4.000	h Licencia MATLAB	0.962 3.85
	Maq1		4.000	h Ordenador (CPU. pantalla. teclado. etc.)	0.049 0.20
	MO2		4.000	h Programador	20.000 80.00
	%		2.000	% Costes Directos Complementarios	84.050 1.68
			2.000	% Costes indirectos	85.730 1.71
				Precio total por ud. 87.44	
	2.4 Hor3	h	Hora de trabajo dedicada a la obtención de las ecuaciones matemáticas y el procedimiento de cálculo que permiten representar el comportamiento de una unidad de ósmosis inversa		
	Maq1		0.500	h Ordenador (CPU. pantalla. teclado. etc.)	0.049 0.02
	MO1		1.000	h Graduado en GIQ	30.000 30.00
	MO3		0.250	h Ingeniero químico superior	45.000 11.25
	%		2.000	% Costes Directos Complementarios	41.270 0.83
			2.000	% Costes indirectos	42.100 0.84
				Precio total por h. 42.94	
	2.5 Hor4	h	Hora de trabajo dedicada a la obtención de las ecuaciones matemáticas y el procedimiento de cálculo que permiten representar el comportamiento de un tanque de lavado		
	Maq1		0.500	h Ordenador (CPU. pantalla. teclado. etc.)	0.049 0.02
	MO1		1.000	h Graduado en GIQ	30.000 30.00
	%		2.000	% Costes Directos Complementarios	30.020 0.60
			2.000	% Costes indirectos	30.620 0.61
				Precio total por h. 31.23	
	2.6 Hor5	h	Hora de trabajo dedicada a la obtención de las ecuaciones matemáticas y el procedimiento de cálculo que permiten representar el comportamiento de una unidad de intercambio iónico		
	Maq1		0.500	h Ordenador (CPU. pantalla. teclado. etc.)	0.049 0.02
	MO1		1.000	h Graduado en GIQ	30.000 30.00
	%		2.000	% Costes Directos Complementarios	30.020 0.60
			2.000	% Costes indirectos	30.620 0.61
				Precio total por h. 31.23	

Nº	Código	Ud	Descripción	Total
3 Simulación de un proceso industrial mediante COCO				
3.1	Hor6	h	Hora de trabajo dedicada a la implementación del modelo MATLAB en un unidad de proceso personalizada del simulador COCO	
	Maq1	1.000	h Ordenador (CPU, pantalla, teclado, etc.)	0.049 0.05
	MO2	1.000	h Programador	20.000 20.00
	%	2.000	% Costes Directos Complementarios	20.050 0.40
		2.000	% Costes indirectos	20.450 0.41
Precio total por h.				20.86
3.2	Hor7	h	Hora de trabajo dedicada al estudio de un proceso de regeneración de aguas industriales en el cual aparezcan unidades de ósmosis inversa	
	Maq1	1.000	h Ordenador (CPU, pantalla, teclado, etc.)	0.049 0.05
	MO1	1.000	h Graduado en GIQ	30.000 30.00
	MO3	0.100	h Ingeniero químico superior	45.000 4.50
	%	2.000	% Costes Directos Complementarios	34.550 0.69
		2.000	% Costes indirectos	35.240 0.70
Precio total por h.				35.94
3.3	Lic3	ud	Licencia que permite el uso comercial de la unidad personalizada MATLAB de Amsterchem en el simulador de procesos COCO	
			Sin descomposición	400.000
		2.000	% Costes indirectos	400.000 8.00
Precio total redondeado por ud.				408.0

5. JUSTIFICACIÓN DE LOS PRECIOS UNITARIOS

Para establecer el precio de los materiales (licencias) y maquinaria que se requieren en el desarrollo del proyecto se ha decidido calcular el coste por hora de uso de cada uno de ellos en base a su precio y vida útil.

En el caso de la maquinaria se ha considerado que el precio de un ordenador es aproximadamente 500 € y que su vida útil está en torno a 5 años. Por lo tanto, considerando que un año está compuesto por 52 semanas con 5 días laborables cada una y 8h de trabajo por día, el coste de utilizar una hora el ordenador resulta ser de:

$$\text{Coste ordenador} = \frac{500}{5 \cdot 52 \cdot 5 \cdot 8} = 0.049 \text{ €/h}$$

El cálculo del coste de las licencias resulta ser un poco más complejo, dado que la adquisición de una licencia permite el uso del programa de forma indefinida. Sin embargo, cada cierto tiempo las empresas de software lanzan nuevas versiones de sus programas, por lo que es necesario adquirir una nueva licencia para así tener el software actualizado y poder mantener de esta forma la calidad de los trabajos realizados.

En consecuencia, se considera que cada vez que sale una versión nueva del software utilizado se adquiere una nueva licencia y por este motivo se asume que una licencia de MATLAB puede utilizarse durante 1 año y una de MATHCAD durante 6 años. De esta manera, el coste de utilizar una hora cada uno de estos programas será de:

$$\text{Coste MATLAB} = \frac{2000}{1 \cdot 52 \cdot 5 \cdot 8} = 0.962 \text{ €/h}$$

$$\text{Coste MATHCAD} = \frac{1540}{6 \cdot 52 \cdot 5 \cdot 8} = 0.124 \text{ €/h}$$

En cuanto a la mano de obra, se ha supuesto que una Ingeniería necesitaría de un programador y un graduado en Ingeniería Química para poder abordar el desarrollo del proyecto, los cuales desempeñarían las tareas que ha realizado el alumno durante el TFG. Así mismo, se ha considerado la necesidad de un Ingeniero Químico con experiencia en el campo para representar el papel que ha tenido el tutor en la elaboración del TFG.

Asciende el presupuesto de ejecución por contrata a la expresada cantidad de DOCE MIL SETECIENTOS EUROS CON CINCUENTA Y UN CÉNTIMOS (12,700.51).

En Valencia a 05 de junio de 2014:

Fdo: Miguel Sanchis Sebastián