

# **PRODIN 1.0**

## **SOFTWARE DIDÁCTICO DE PROGRAMACIÓN DINÁMICA EN DISEÑO DE PROCESOS**

# I

## INTRODUCCIÓN

En la formación de ingenieros químicos, la aplicación de la optimización de procesos es fundamental. Establece el conjunto de condiciones en las que se debe controlar y operar el proceso para tener lo mejor tanto en funcionamiento como en la calidad del producto final, así como procedimientos seguros. Un curso de diseño de procesos a nivel licenciatura incluye la iniciación en el aprendizaje de algunos de los métodos de optimización más simples, ya que aun existiendo muchas técnicas, éstas no son debidamente analizadas debido a que el alumno no tiene el conocimiento para atacar los problemas que se presentan en procesos industriales, la mayoría de ellos son altamente no - lineales y multivariantes, y para resolverlos son necesarios métodos matemáticos rigurosos que en muchos casos sólo pueden ser aplicados por medio de la computadora.

En la asignatura de Diseño de Procesos al alumno frecuentemente se le dificulta asimilar el Método de Programación Dinámica. Esta técnica se basa en el principio de Bellman y es aplicable a sistemas sin reciclo de información, para practicarlo el alumno debe comprender sus fundamentos, plantear y resolver cada problema en forma particular; y por lo tanto, no se puede mecanizar, lo que constituye su dificultad principal. Además estos deben estar relacionados con la ingeniería química actual, en la cuál el análisis de los sistemas de producción por medio de modelos matemáticos se está haciendo una práctica obligada debido al ahorro que significa el tener una idea previa de lo que se debe o no implementar en una industria. La optimización mediante estos modelos es una tarea muy importante.

Actualmente las computadoras se han convertido en un gran aliado en el proceso enseñanza - aprendizaje de las más diversas áreas de estudio, una herramienta de simulación para éste tipo de sistemas sería de gran ayuda a los estudiantes en la comprensión de este tópico

tan importante en su formación, ya que los programas de computo interactivos estimulan el razonamiento de los problemas que se presentan a través de la pantalla.

El objetivo de PRODÍN 1.0 es ayudar al alumno a comprender los principios del método de Programación Dinámica al resolver ejemplos de distinta dificultad y temática, en forma interactiva con la computadora.

## II

# PROGRAMACIÓN DINÁMICA

A continuación se presenta un resumen de los conceptos básicos para la aplicación del método de Programación Dinámica en diseño de procesos.

### 2.1 Antecedentes

La Programación Dinámica es una técnica desarrollada por Richard Bellman a mediados de los 50's que puede ser aplicada en la optimización de estructuras en serie (Harmon, 1973). El Principio de Optimización de Bellman establece que:

*"Dado un sistema acíclico, éste se optimiza si cada componente o unidad se optimiza a su vez para todo el conjunto de posibles valores de las variables que provienen de las etapas anteriores"* (Jiménez, 2000)

Algunos problemas típicos que se pueden resolver con Programación Dinámica son los sistemas en serie, los cuales pueden ser representados como una serie de unidades interconectadas, o que por la estructura de su modelo matemático son fácilmente dissociadas. Por ejemplo, una serie de tanques agitados en un proceso químico, o cualquier otra estructura secuencial en un proceso complejo. La descomposición del sistema puede ser llevada a cabo a través del modelo matemático con el cual es representado, no importa si consiste de una sola unidad. Otro ejemplo son los problemas que involucran el reparto de un recurso; en estos problemas una cantidad debe ser distribuida de acuerdo al tiempo y el espacio para optimizar la totalidad del sistema (Sanford, 1964).

## 2.2 Descripción del Método

La ventaja del método de Programación Dinámica es que se obtiene una secuencia óptima de decisiones, ya que proporciona el conjunto de variables de diseño adecuadas para que el comportamiento del sistema sea de acuerdo con la función objetivo. Se utiliza sólo en sistemas donde no hay recirculación de información y/o componentes, de manera que una decisión tomada en una etapa del sistema sólo afecta los equipos posteriores. La Figura 2.1 muestra como la Programación Dinámica se aplica en orden opuesto al flujo de materiales. Esto significa que se debe iniciar el análisis del sistema por el último equipo del tren de separación, y el primer equipo se analizará una vez suboptimizadas todas las etapas posteriores. Una vez concluida esta revisión y después de haber tomado las decisiones que se crean adecuadas, se procede a aplicar el flujo de información de acuerdo con el problema original para comprobar que la solución es la correcta (Jiménez, 2000).

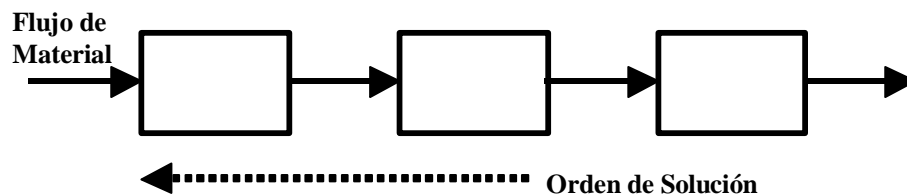


Figura 2.1 Sentido de Solución en Programación Dinámica

Como ejemplo se tiene el sistema descrito en la Figura 2.2, donde los vectores  $\bar{s}_i$  representan las variables de estado saliendo de cada etapa  $i$ , y los vectores  $\bar{d}_i$  al conjunto de variables de diseño para cada etapa  $i$ . El sistema mostrado tiene  $N$  etapas, donde  $\bar{s}_0$  caracteriza la alimentación del sistema, y  $\bar{s}_N$  a la corriente saliendo de la misma etapa y entrando en la Etapa  $N-1$ . Las etapas se enumeran en ese sentido por conveniencia.

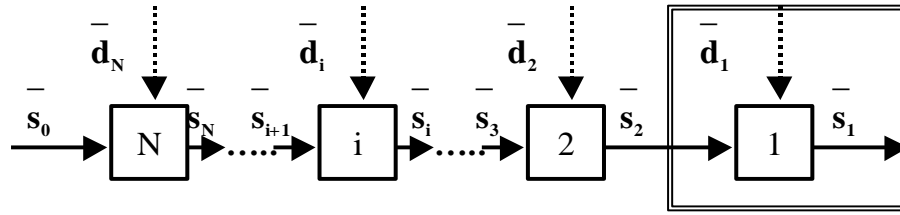


Figura 2.2 Representación de un sistema de N etapas.

Para cada etapa es necesario identificar las variables que representan los grados de libertad que forman parte del vector de variables de estado, y las que forman parte del vector de variables de diseño en cada etapa. Si aplicamos Programación Dinámica, el objetivo es optimizar el sistema como un todo, para cada etapa se busca el valor del vector  $\bar{s}_i$ , como el vector que caracteriza los valores de salida de cada unidad de la serie que permita obtener el vector  $\bar{d}_i$  que contenga los valores óptimos de las variables de diseño en dicha etapa. Esto para cada conjunto posible de valores de entrada  $\bar{s}_{i+1}$ .

Siguiendo el principio de Bellman, la optimización se lleva acabo de la siguiente forma:

1. Se establece una Función Objetivo Global, ésta es una expresión que representa la meta a la cual se quiere llegar al optimizar el sistema en su conjunto, por ejemplo, maximizar utilidades o minimizar gastos. La siguiente expresión es una representación burda de la ganancia máxima de un sistema de N etapas.

$$F\left(\bar{s}_0, \bar{s}_N, \bar{s}_{N-1}, \dots, \bar{s}_2, \bar{s}_1\right) = \text{Max}_{\bar{d}_0, \bar{d}_N, \bar{d}_{N-1}, \dots, \bar{d}_2, \bar{d}_1} \left[ \begin{array}{l} \text{(costo unitario del producto)} - \\ \text{(gastos de producción por unidad de producto)} \end{array} \right]$$

2. Como parte del método de Programación Dinámica se analiza en forma aislada cada una

de las etapas que forman parte del sistema; así, el problema original es convertido en un conjunto de subsistemas donde la Función Objetivo Global se fragmenta de tal forma que se tenga una función objetivo para cada uno de los subsistemas, y al final del procedimiento en su conjunto cumplan con lo establecido inicialmente. Puede haber diferentes formas de establecer las funciones objetivo individuales para cada etapa, una forma de hacerlo se explica en la siguiente sección.

3. Se inicia el procedimiento analizando la Etapa 1 en forma aislada y basándose en su función objetivo se busca el conjunto de valores de  $\overline{s_1}$  que optimicen  $\overline{d_1}$  para cualquier valor posible de  $\overline{s_2}$ .
4. Se repite la misma operación para la Etapa 2, buscando el valor de  $\overline{s_2}$  que optimice  $\overline{d_2}$  para los valores posibles de  $\overline{s_3}$ . En este nivel de optimización no se toma en cuenta la Etapa 1, ya que tenemos los valores de  $\overline{s_1}$  que optimizan el vector  $\overline{d_1}$ , para los posibles valores de  $\overline{s_2}$ .
5. El procedimiento continúa aguas arriba hasta llegar a la Etapa N, donde se obtienen los valores óptimos de  $\overline{d_N}$  para los posibles valores de  $\overline{s_0}$ , buscando el valor adecuado de  $\overline{s_N}$ . Ya que  $\overline{s_0}$  está compuesto por las condiciones de la alimentación, generalmente se conoce su valor, por lo cual sólo se busca el  $\overline{s_N}$  que optimice  $\overline{d_N}$ .
6. Una vez encontrado el  $\overline{d_N}$  óptimo, se observa el  $\overline{s_N}$  que le corresponde, así tenemos los valores a utilizar en la Etapa N. Para la Etapa N-1; de acuerdo al  $\overline{s_N}$  elegido se busca el valor  $\overline{s_{N-1}}$  que corresponda a su vez a  $\overline{d_{N-1}}$  suboptimizado. Se repite el procedimiento aguas abajo hasta llegar a la Etapa 1.

La característica principal de la técnica es que ya determinados los valores óptimos de

una etapa dada para diferentes características de alimentación, estos valores permanecen fijos e inalterables para el resto de las unidades. Es importante conocer algunos parámetros útiles para trabajar en el esquema de cálculo:

$$\overline{\mathbf{d}_i^*} \left( \overline{\mathbf{s}_{i+1}} \right) = \text{valor óptimo de las variables de diseño } \overline{\mathbf{d}_i} \text{ para la alimentación } \overline{\mathbf{s}_{i+1}}.$$

$$\mathbf{P}_i \left( \overline{\mathbf{s}_{i+1}} \right) = \text{función objetivo para las etapas } 1, 2, \dots, N \text{ con una alimentación } \overline{\mathbf{s}_{i+1}}.$$

$$\overline{\mathbf{s}_{i+1}} \quad \text{en la etapa } i \quad = \quad \mathbf{Max}_{\overline{\mathbf{d}_1}, \overline{\mathbf{d}_2}, \dots, \overline{\mathbf{d}_i}} \left[ \sum_{j=1}^i \mathbf{F}_j \left( \overline{\mathbf{s}_{i+1}}, \overline{\mathbf{d}_j} \right) \right] \quad (2.1)$$

Con estas definiciones, se establece la función objetivo para la optimización de la Etapa 1, se analizan los valores de alimentación posibles, y se busca el conjunto de valores de salida que den el valor óptimo de  $\overline{\mathbf{d}_1}$ :

$$\mathbf{P}_1 \left( \overline{\mathbf{s}_2} \right) = \mathbf{Max}_{\overline{\mathbf{d}_1}} \left[ \mathbf{F}_1 \left( \overline{\mathbf{s}_2}, \overline{\mathbf{d}_1} \right) \right]$$

(2.2)

La optimización de las dos últimas etapas requiere la búsqueda de un valor máximo de  $\mathbf{P}_2(\overline{\mathbf{s}_3})$  para obtener  $\overline{\mathbf{d}_2}$  óptimo, encontrando el valor adecuado  $\overline{\mathbf{s}_2}$  para los valores posibles de  $\overline{\mathbf{s}_3}$ :

$$\mathbf{P}_2 \left( \overline{\mathbf{s}_3} \right) = \mathbf{Max}_{\overline{\mathbf{s}_2}, \overline{\mathbf{s}_1}} \left[ \mathbf{F}_2 \left( \overline{\mathbf{s}_3}, \overline{\mathbf{d}_2} \right) + \mathbf{F}_1 \left( \overline{\mathbf{s}_2}, \overline{\mathbf{d}_1} \right) \right] = \mathbf{Max}_{\overline{\mathbf{s}_2}} \left[ \mathbf{F}_2 \left( \overline{\mathbf{s}_3}, \overline{\mathbf{d}_2} \right) + \mathbf{P}_1 \left( \overline{\mathbf{s}_2} \right) \right]$$



(2.3)

Siguiendo el mismo procedimiento lo único que se tiene que hacer en cualquier etapa  $i$ , es resolver el problema de una sola etapa dado por:

$$\mathbf{P}_i\left(\overline{s_{i+1}}\right)=\underset{\overline{s_i}, \overline{s_{i-1}}, \dots, \overline{s_2}, \overline{s_1}}{\mathbf{Max}}\left[\sum_1^i \mathbf{F}_i\left(\overline{s_{i+1}}, \overline{\mathbf{d}_i}\right)\right]=\underset{\overline{s_i}}{\mathbf{Max}}\left[\mathbf{F}_i\left(\overline{s_{i+1}}, \overline{\mathbf{d}_i}\right)+\mathbf{P}_{i-1}\left(\overline{s_i}\right)\right]$$

(2.4)

Cuando se llega a la Etapa  $N$ , en el inicio de la secuencia, el valor óptimo de  $\overline{\mathbf{d}_N}$  puede ser encontrado con sólo una alimentación fija, o por un conjunto de valores posibles para este vector con un esfuerzo extra. Los conceptos definidos quedarán más claros al analizar los casos de estudio.

# III

## EJEMPLO No. 1

### EXTRACCIÓN

Se busca encontrar la cantidad optima de solvente de lavado  $W(i)$  en un proceso de extracción como el de la Figura 3.1.

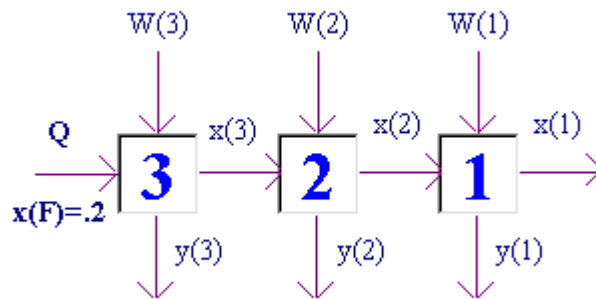


Figura (3.1). Representación Esquemática del Sistema

La composición de la alimentación es  $x(F)=0.2$ , y la alimentación es  $Q=1000 \text{ lb/h}$ ; se propone la siguiente función objetivo:

$$\text{Max}_{\{W(1), W(2), W(3)\}} [Q(x(F) - x(1)) - \lambda(W(1) + W(2) + W(3))] \quad (3.1)$$

Donde:  $Q[x(F)-x(1)]$  = Cantidad de soluto removida.

$\lambda$  = Costo del solvente de lavado/costo del soluto extraído. Se supondrá un valor de **0.05**.

### **Establecer relaciones de diseño por etapa:**

Balances de Materia.

Total: Suponiendo que los solventes no son miscibles,

$$Q_{\text{ent}} = Q_{\text{sal}} = Q$$

(3.2)

$$W(i)_{\text{entra}} = W(i)_{\text{sale}} = W(i) \quad (3.3)$$

Soluto:

$$Qx(i+1) = Qx(i) + Wy(i) \quad (3.4)$$

Relación de Equilibrio:

$$y(i) = 617.854[x(i)]^4 - 196.9[x(i)]^3 + 8.328[x(i)]^2 - .0003 \quad (3.5)$$

Tenemos dos ecuaciones independientes para cada etapa (3.4 y 3.5), y cuatro variables  $W(i)$ ,  $x(i+1)$ ,  $x(i)$ ,  $y(i)$ ),  $Q$  es una cantidad fija,  $y(i)$  depende de  $x(i)$ , y  $x(i+1)$  se consume en el análisis de cada etapa pues ya ha sido analizada previamente debido a la forma de solución que plantea Bellman. La variable de diseño natural para el sistema es  $W(i)$  pero es más práctico utilizar  $x(i)$  para la optimización pues se pueden acotar sus valores que varían entre 0.0 y 0.2. Una vez encontrados los valores óptimos de  $x(2)$ ,  $W(i)$  se determina con la Ecuación 3.4.

$$\text{Grados de Libertad} = \text{No. de Variables} - \text{No. de Ecuaciones} \quad (3.6)$$

# IV

## EJEMPLO No. 2

### DESTILACIÓN

### Síntesis Óptima de Sistemas de Separación

Una de las etapas más importantes en el diseño de procesos químicos es la síntesis de sistemas de separación. Para separar una mezcla de varios componentes existen varias alternativas; el problema entonces es el de detectar aquella que cumpla con los requerimientos de separación a un costo mínimo. En este trabajo se estudia la síntesis óptima de sistemas de destilación para separar una mezcla de multicomponentes. Usando programación dinámica, se detectan las secuencias de separación más económicas en función de la presión de diseño del tren de destilación.

**Se presenta el siguiente problema:** Se tiene una mezcla de cinco componentes, se desea encontrar la secuencia optima de separación, así como sus condiciones de operación.

Tabla 4.1 Características del sistema

<b>Mezcla de Alimentación:</b>		
Flujo de Entrada = 907.2 Kg mol/h		
<b>Componente</b>		<b>Fracción mol</b>
Propano	A	0.05
Iso – butano	B	0.15
N – butano	C	0.25
Iso – pentano	D	0.20
N – pentano	E	0.35
<b>Tipo de Separador:</b> Columna de Destilación		
<b>Material de construcción:</b> Acero al Carbón		

Tabla 4.2 Características Predeterminadas

<b>Valores Supuestos:</b>
$\eta = 80\%$ en todas las columnas
R = valor óptimo con respecto al costo total anual.
Recuperación de los componentes claves = 98% (para ambos en todas las columnas.)
Horas de operación por año = 8500
Vida del proyecto = 10 años.

Tabla 4.3 Costo de servicios

Servicio (Vapor)	Costo (\$/millón de kcal)	Servicio (Enfriamiento)	Costo (\$/millon de kcal)
(28.23 atm)	4.29	Amoníaco (1 °C)	6.91
(11.22 atm)	3.24	Amoníaco (-17.78 °C)	12.43
(4.08 atm)	2.4	Amoníaco (-21.67 °C)	16.59
(1.7 atm)	1.75	Agua (32.2 °C)	0.28

Tabla 4.4 Ecuaciones de Costo y Diseño

**Costo Base para la Optimización:**

$$\text{costo total anual} = \text{costo anual de operación} + \frac{\text{costo total de instalación de equipo}}{\text{vida del proyecto}}$$

**Número Mínimo de Etapas,  $S_m$ :**

$$\frac{\text{kg mol de } i \text{ en el destilado}}{\text{kg mol de } i \text{ en el fondo}} = (\alpha_{i,r})^{S_m} \frac{\text{kg mol de } r \text{ en el destilado}}{\text{kg mol de } r \text{ en el fondo}}$$

**Relación Mínima de Reflujo,  $R_m$ :**

$$\sum_{i=1}^N \frac{\alpha_i Z_{i,F}}{\alpha_i - \theta} = 1 - \phi$$

$$R_m = \sum_{i=1}^N \frac{\alpha_i X_{i,D}}{\alpha_i - \theta} - 1$$

**Número de Etapas:**

Para  $\frac{R - R_m}{R + 1} < 0.125$ ;

$$\frac{S - S_m}{S + 1} = 0.5039 - .5968 \left( \frac{R - R_m}{R + 1} \right) - 0.0908 \log \left( \frac{R - R_m}{R + 1} \right)$$

Para  $\frac{R - R_m}{R + 1} > 0.125$  ;

$$\frac{S - S_m}{S + 1} = 0.6257 - .9868 \left( \frac{R - R_m}{R + 1} \right) - 0.5160 \left( \frac{R - R_m}{R + 1} \right)^2 - .1738 \left( \frac{R - R_m}{R + 1} \right)^3$$

**Diámetro de Columna,  $D_C$ :**

$$D_C = \left[ \left( \frac{4}{\pi V} \right) (D) (R + 1) (22.2) \left( \frac{T_{DV}}{273} \right) \left( \frac{1}{P} \right) \left( \frac{1}{3600} \right) \right]^{1/2}$$

Donde:  $V = 0.761 \left( \frac{1}{P} \right)^{1/2}$

Altura de la Columna,  $H_C$ :  $H_C = 0.61 \left( \frac{S}{\eta} \right) + 4.27$

**Costo**

**de**

**Columnas:**

$$\text{costo de la columna instalada} = 4.34 \left[ 762 D_C \left( \frac{H_C}{12.2} \right)^{0.68} \right]$$

Para  $P > 3.4$  atm, el factor de corrección es:  $[1 + 0.0147(P - 3.4)]$

$$\text{costo de platos instalados} = 70 \left( \frac{s}{\eta} \right) \left( \frac{D_C}{1.22} \right)^{1.9}$$

costo de instrumentación = 4,000.00

costo de mantenimiento = .02 \* costo total de la columna

**Costo de Intercambiador de Calor:**

$$\text{costo total} = 3.39 \left[ 9000 \left( \frac{A}{92.1} \right)^{0.65} \right]$$

Para  $P < 10.2$  atm, factor de corrección:  $[1 + 0.0147(P - 10.2)]$ .

costo de operación del intercambiador =  $8500 (C_u Q) + .02 * \text{costo total}$

\*\*\*\*\*

Se desprecian los costos de compresión

$$K = A_1 + A_2 * T + A_3 * T^2 + A_4 * T^3$$

$$HL = C1 + C2 * T + C3 * T^2 + C4 * T^3$$

$$U = 0.0009765 \text{ kcal/hr m}^2$$

$DT = 15^{\circ}\text{C}$



P = 8.0 atm	Propano	iso-Butano	n-Butano	iso-Pentano	n-Pentano
Para K					
A 1	34.487455200000	17.134359700000	11.330412400000	1.380518450000	-.279498975000
A 2	-.329717425000	-.151834483000	-.094747398000	-.002297884220	.012770713400
A 3	.001002815280	.000415174730	.000232315882	-.000041436023	-.000084752684
A 4	-.000000914947	-.000000312320	-.000000128285	.000000118013	.000000155373
Para HV					
B 1	8621.920030000000	-5195.297180000000	-7714.691130000000	-21309.297000000000	-23831.348400000000
B 2	-75.790864500000	34.527261300000	56.957681900000	160.566162000000	181.979513000000
B 3	.246258000000	-.036893023300	-.095523271400	-.364340852000	-.419953063000
B 4	-.000204641395	.000053420000	.000104251350	.000351950744	.000400382750
Para HL					
C 1	8621.920030000000	-5195.297180000000	-7714.691130000000	-21309.297000000000	-23831.348400000000
C 2	-75.790864500000	34.527261300000	56.957681900000	160.566162000000	181.979513000000
C 3	.246258000000	-.036893023300	-.095523271400	-.364340852000	-.419953063000
C 4	-.000204641395	.000053420000	.000104251350	.000351950744	.000400382750

P = 10.0 atm	Propano	iso-Butano	n-Butano	iso-Pentano	n-Pentano
Para K					
A 1	26.794282700000	15.212114460000	10.640655530000	18.788097200000	10.648201000000
A 2	-.259350334000	-.134523356000	-.089163500000	-.145891186000	-.078322645380
A 3	.000796358200	.000369264020	.000223282710	.000355352437	.000170793626
A 4	-.000000731347	-.000000283973	-.000000136280	-.000000251787	-.000000087800
Para HV					
B 1	108708.325000000000	-36483.061700000000	-17982.864000000000	52238.310000000000	66831.340000000000
B 2	881.012950000000	286.672550000000	137.204804900000	-445.686100000000	-564.375000000000
B 3	-2.339006910000	-.712790300000	-.306346234000	1.285031000000	1.608881000000
B 4	.002109472000	.000655040173	.000289160000	-.001131786960	-.001423616300
Para HL					
C 1	108708.325000000000	-36483.061700000000	-17982.864000000000	52238.310000000000	66831.340000000000
C 2	881.012950000000	286.672550000000	137.204804900000	-445.686100000000	-564.375000000000
C 3	-2.339006910000	-.712790300000	-.306346234000	1.285031000000	1.608881000000
C 4	.002109472000	.000655040173	.000289160000	-.001131786960	-.001423616300

P = 12.0 atm	Propano	iso-Butano	n-Butano	iso-Pentano	n-Pentano
Para K					
A 1	29.272912000000	14.739910000000	10.445450000000	2.389669000000	1.027367300000
A 2	-.275811000000	-.129577520000	-.087651030000	-.013470600000	-.001220406700
A 3	.000833364500	.000356911840	.000223489600	.000004156868	-.000031079609
A 4	-.000000768160	-.000000281667	-.000000147467	.000000052227	.000000083133
Para HV					
B 1	130784.202000000000	15818.503000000000	56039.610000000000	198613.600000000000	229020.152000000000
B 2	1071.885600000000	-141.583820000000	-407.522360000000	-1655.665100000000	-1905.597140000000
B 3	-2.881329400000	.445472670000	1.342090400000	4.587787250000	5.271286070000
B 4	.002616454000	-.000381520100	.001190067000	-.004112401100	-.004729978000
Para HL					
C 1	130784.202000000000	15818.503000000000	56039.610000000000	198613.600000000000	229020.152000000000
C 2	1071.885600000000	-141.583820000000	-407.522360000000	-1655.665100000000	-1905.597140000000
C 3	-2.881329400000	.445472670000	1.342090400000	4.587787250000	5.271286070000
C 4	.002616454000	-.000381520100	.001190067000	-.004112401100	-.004729978000

P = 15.0 atm	Propano	iso-Butano	n-Butano	iso-Pentano	n-Pentano
Para K					
A 1	50.077259700000	26.017442460000	25.897364000000	18.787750000000	16.481404300000
A 2	-.450254571000	-.218818042000	-.210646815000	-.143527310000	-.123886534000
A 3	.001316188730	.000595230507	.000552849500	.000350497900	.000295823300
A 4	-.000001218120	-.000000500773	-.000000447627	-.000000259093	-.000000210733
Para HV					
B 1	245239.410000000000	-96223.685000000000	-55976.890000000000	88977.574200000000	119701.248000000000
B 2	1750.672100000000	676.993964000000	389.836500000000	-659.538821000000	-879.488900000000
B 3	-4.117828000000	-1.546231420000	-.861646840000	1.649857200000	2.175328800000
B 4	.003258105400	.001231533000	.000690239730	-.001287426000	-.001703319300
Para HL					
C 1	245239.410000000000	-96223.685000000000	-55976.890000000000	88977.574200000000	119701.248000000000
C 2	1750.672100000000	676.993964000000	389.836500000000	-659.538821000000	-879.488900000000
C 3	-4.117828000000	-1.546231420000	-.861646840000	1.649857200000	2.175328800000
C 4	.003258105400	.001231533000	.000690239730	-.001287426000	-.001703319300

# V

## P R O D Í N 1.0

### 5.1 Entorno del Programa

PRODÍN 1.0 es un programa realizado en Microsoft Visual Basic 6.0. Este lenguaje permite que la aplicación sea ejecutada en ambiente Windows 95, 98, y 2000; Se seleccionó esta plataforma por que la mayoría de los alumnos tienen acceso a una computadora PC compatible. Se utilizó este lenguaje debido a la versatilidad de herramientas que pueden ser usados en la aplicación creada, además de que es relativamente fácil aprender a manejarlo en forma eficiente. Una de las ventajas de Microsoft Visual Basic 6.0 es el asistente automático que permite la creación de un programa de instalación, y el correspondiente para la desinstalación de la aplicación con los archivos y librerías necesarias para que el programa creado pueda ser ejecutado en cualquier computadora que tenga Windows versión 95 o superior.

#### INSTRUCCIONES DE INSTALACIÓN

- Se tienen disponibles 3 discos de instalación, que contienen el archivo instalar.zip fragmentado.
- Introducir disco 3 en unidad A.
- Entrar a Explorador de Windows (Explorer), examinar unidad A. Aparece icono “Instalar”
- Dar click derecho sobre “Instalar”
- Aparece un menú, hacer click sobre “Extract to...”
- En el cuadro de nombre Winzip hacer click en botón “I Agree”, o en botón “WinZip Classic”, dependiendo de la versión que tengas en tu PC
- Aparece recuadro titulado Extract, en “Extract to” teclea c:\Prodin\_1\support y haz click en botón “Extract”.
- Winzip solicitará cada uno de los 3 discos

- Ir a ‘Inicio’(Start)/’Ejecutar’(Run)
- En Abrir(Open) teclear [C:\PRODIN\_1\Support\setup.exe] con lo cual se iniciará el asistente de instalación que proporciona las instrucciones necesarias para descargar los archivos necesarios para instalar PRODÍN 1.0.
- Se genera la carpeta PRODIN 1.0, y para ejecutar el programa se debe elegir el icono PRODIN\_1.EXE.

NOTA: en caso de que al tratar de iniciar setup.exe el programa indique que no se tienen disponibles archivos \*.dll, copiar los archivos \*.dll encontrados en la carpeta C:\PRODIN\_1\Support en la carpeta C:\windows\system.

El programa ejecutable tiene una extensión aproximada de 3 MB, y para su instalación son necesarios 6 MB de espacio libre en el disco duro, ya que generará archivos que permiten que los discos de instalación no sean necesarios. Una de las ventajas de PRODÍN 1.0 es que está diseñado de modo que se ajusta automáticamente a cualquier resolución y configuración de color de la pantalla.

## 5.2 Descripción del Programa

PRODÍN 1.0 permite al usuario resolver uno de los dos problemas incluidos en forma interactiva, seleccionando las respuestas correctas a las diversas preguntas que se plantean, de una serie de alternativas que se presentan en pantalla. Si contesta erróneamente tendrá la oportunidad de corregir su respuesta, si comete una falla nuevamente, aparecerá la respuesta correcta, así como una breve explicación del error cometido. El alumno tiene la posibilidad de avanzar o regresar en la resolución del problema, según lo considere necesario por medio de los comandos accesibles mediante el ratón y los distintos menús.

Inicialmente se planeó este trabajo de forma que se incluyeran varios ejemplos, pero debido a diversos factores el desarrollo del programa se alargó y se decidió acotar el alcance a dos problemas que

se consideran representativos por la diferencia que existe en la forma de resolver ambos.

Se debe enfatizar que no se pretende crear un programa de solución general, sino una herramienta para auxiliar en el aprendizaje del método de Programación Dinámica en la materia de Diseño de Procesos, y PRODÍN está limitado a dos ejemplos hasta ahora. Por lo tanto, es una herramienta con fines exclusivamente didácticos, por lo cual el programa será proporcionado gratuitamente a los estudiantes de ingeniería química internos y externos que lo soliciten al Instituto Tecnológico de Celaya a través de los autores.

## REFERENCIAS

GARCÍA (2000), *Software Didactico de Programación Dinámica en Diseño de Procesos*. (Tesis para Obtención de Grado de Maestria en Ciencias en Ingeniería Química) Departamento de Ingeniería Química. Instituto Tecnológico de Celaya. Celaya Guanajuato.

HENLEY y Seader (1988), *Operaciones de Separación por Etapas de Equilibrio en Ingeniería Química*. Editorial Reverté S.A. España.

HOROWITZ y Sartaj (1978), *Fundamental of Computer Algorithms*. Computer Science Press. Estados Unidos de Norteamérica.

JIMÉNEZ (2000), *Notas de Diseño de Procesos*. (texto en elaboración) Departamento de Ingeniería Química. Instituto Tecnológico de Celaya. Celaya Guanajuato.

RAY y Szekely (1973), *Process Optimization (with Applications in Metallurgy and Chemical Engineering)*. Wiley-Interscience Publication. State University of New York at Buffalo.

SANFORD (1964), *Dynamic Programming in Chemical Engineering and Process Control*. Academic Press. Houston Texas.

TISCAREÑO y Jiménez (1988), *Síntesis Óptima de Sistemas de Separación*. Ingeniería Química. Mayo de 1988, Pgs. 172-177. España.