



Un hombre no es otra cosa que lo  
que hace de sí mismo.

Jean Paul Sartre

## Capítulo 3.

### Descripción del programa.

#### 3.1 Lo nuevo en ésta versión.

Como resultado de este trabajo se tiene una versión nueva del programa que simula el proceso de recuperación de azufre, a esta nueva versión se le ha dado el nombre de *WinClaus*. El programa fue escrito íntegramente en el lenguaje llamado *Delphi*. En el apéndice B se muestra el procedimiento a seguir para efectuar la simulación de una planta de recuperación de azufre con *WinClaus*.

En la nueva versión se hicieron las siguientes mejoras.

En la interfaz del usuario:

- El programa es una aplicación MDI para *Windows* de 32 bits. Cuenta con una ventana principal que funciona como contenedor de todas las ventanas secundarias.
- Se permite que el usuario cree sus propios esquemas de proceso o bien modifique alguno existente.
- La información sobre configuración del proceso y sus características, se puede almacenar en archivos cuyos nombres siguen la convención de *Windows* de 32 bits.
- El sistema de unidades se establece según las necesidades del usuario, en el caso de las presiones el programa es capaz de manejar tanto presiones absolutas o manométricas.
- Se pueden especificar las composiciones de las corrientes alimentadas en cualquier base, ya sea como fracción o flujo de los componentes. Las unidades de flujo total son a elección del usuario.
- El reporte de los componentes en las corrientes puede hacerse en términos de fracciones o de flujos, estos últimos en cualquier conjunto de unidades.
- Los valores de temperatura y presión considerados como estándar, así como la presión local pueden ser modificados por el usuario.
- La captura de datos se hace mediante ventanas. En el caso de opciones múltiples no se recurre a palabras clave, todo el manejo se hace mediante listas desplegables.
- La simulación se puede hacer en dos modalidades, una *muda* en donde el programa efectúa la simulación sin ninguna interacción con el usuario, y la otra mediante un *cuadro de controles* donde el usuario puede ver los mensajes que el programa genera durante la ejecución.



- Se cuenta con un sistema de verificación que revisa los datos proporcionados por el usuario con el fin de detectar errores u omisiones.
- El reporte de la simulación puede ser global, por módulos o por corrientes.
- Todos los resultados pueden ser almacenados en archivos con formato de texto enriquecido -RTF-, archivos de texto plano -ASCII- o bien enviados a impresión.
- El programa cuenta herramientas adicionales como una calculadora y un pequeño cuaderno de notas. Todas las ventanas que muestran resultados tienen la capacidad de copiar el texto seleccionado al *portapapeles*, con lo que los resultados pueden ser exportados a cualquier otra aplicación de *Windows*.
- Los diagramas de flujo pueden ser impresos directamente desde el programa.
- Se cuenta con ayuda general sobre las funciones del programa. La ayuda puede consultarse directamente en el ambiente de *WinClaus*.

En la estructura interna del programa:

- Se mejoró de manera significativa la *modularidad* del programa.
- Se rediseñó el núcleo del programa, encargado de llevar a cabo la simulación modular.
- Se mejoró la biblioteca de funciones para el cálculo de propiedades físicas y termodinámicas.
- Se implantó un sistema de manejo de errores que permite una ejecución sin que el programa se *congele* o aborte en caso de ocurrir algún error.

En los modelos de las unidades de proceso:

- Se mejoraron las rutinas que calculan tanto la cantidad de agua contenida en el aire y el gas ácido como la temperatura de rocío del azufre.
- Se mejoró el modelo que simula tanto a los hornos de reacción como a los convertidores catalíticos, implementando una *minimización modificada de la energía libre de Gibbs*, con lo que se toman en cuenta las reacciones limitadas cinéticamente que ocurren en las unidades de proceso. Se establecieron dos factores, el índice térmico y el índice cinético, que tienen como fin ajustar los resultados del horno de reacción a los datos de planta y modelar la desactivación del catalizador en los convertidores catalíticos.



- Se reescribió la rutina que simula a la caldera con el fin de que contemplara los fenómenos que ocurren en su interior. Se implantó el concepto de *temperatura de bloqueo*.
- Se mejoró la rutina que simula a los condensadores de azufre con el fin de considerar el equilibrio físico y el arrastre de azufre debido a la formación de niebla.
- Se programaron dos módulos totalmente nuevos que tienen como fin analizar los esquemas de proceso. Uno de ellos permite observar el efecto de manipular una variable dentro de un intervalo sobre otras variables, y el otro funciona como un control por retroalimentación, que permite que el valor de una variable se ajuste a especificación del usuario mediante la manipulación de otra variable dentro del esquema de proceso.

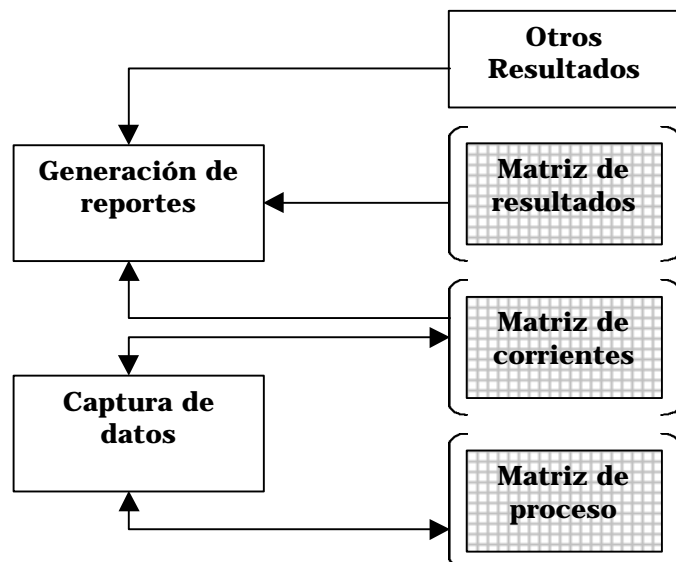
### 3.2 Estructura interna del programa.

En el programa coexisten dos conjuntos de rutinas: unas tienen como fin la gestión de la interfaz de usuario, y las otras se encargan de calcular y tienen un contacto muy limitado con las rutinas encargadas de la gestión de la interfaz de usuario.

#### 3.2.1 Rutinas encargadas de la interfaz de usuario.

Estas rutinas tienen como fin establecer una comunicación apropiada entre el usuario y la sección del programa encargada de efectuar la simulación. Esta comunicación tiene que ver básicamente a la captura de datos y a la generación de reportes. Es también mediante la interfaz de usuario que puede establecerse o modificarse la topología del proceso.

Figura 3.2.1.  
Estructura de la  
interfaz de usuario.

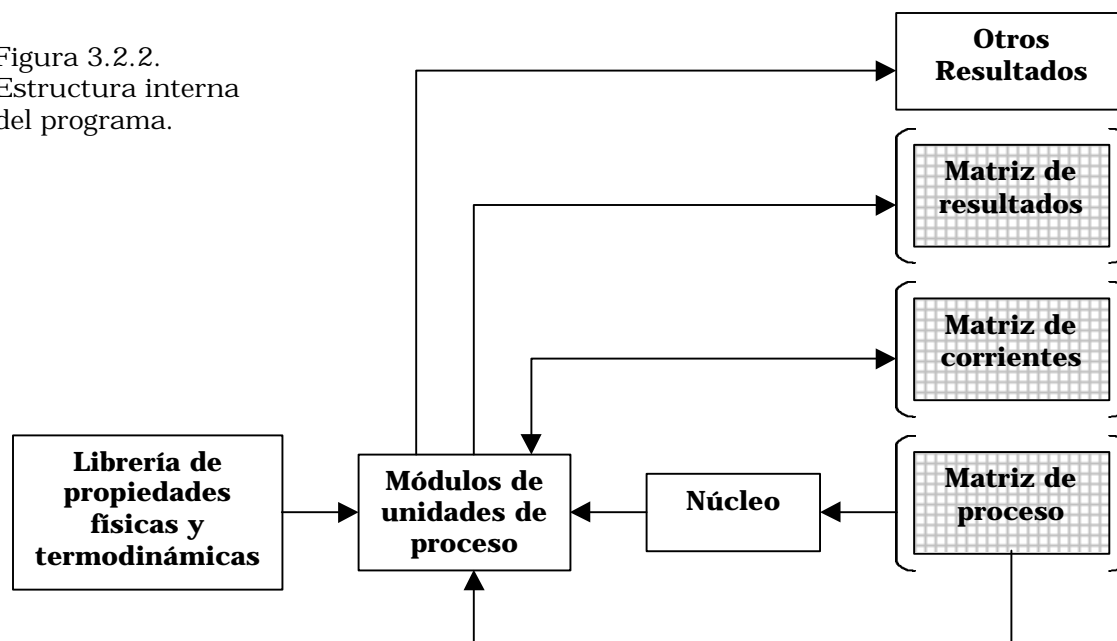




### 3.2.2 Rutinas encargadas de efectuar la simulación.

Esta sección del programa se encarga de la simulación del proceso. La parte principal esta constituida por un *núcleo* que se encarga de coordinar a todas las rutinas necesarias durante la simulación. El núcleo es también es el encargado de generar todos los resultados y mensajes que posteriormente son mostrados en pantalla por la interfaz de usuario.

Figura 3.2.2.  
Estructura interna  
del programa.

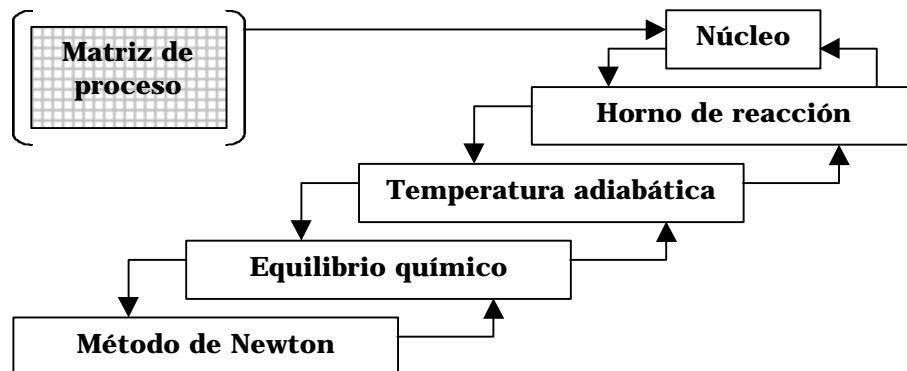


### 3.2.3 Manejo de errores.

El programa posee un sistema que permite el manejo de errores, este se desarrolló mediante el empleo de dos técnicas: la comprobación después de la ejecución y el manejo por excepción. La comprobación después de la ejecución se implanta en las rutinas donde se presume que pueden ocurrir errores. Entre los resultados devueltos por las rutinas se incluyen códigos de error que permiten la toma de decisiones respecto a la ejecución del programa. Para ilustrar el funcionamiento del manejo de errores se considera el caso de la simulación del horno de reacción.



Figura 3.2.3.  
Ejemplo  
aplicación del  
manejo de  
errores.



En este ejemplo el *núcleo* llama a la rutina encargada de simular al horno de reacción, esta a su vez llama a la rutina encargada de calcular la temperatura de reacción adiabática, común al horno de reacción y a los convertidores catalíticos. Dentro de la rutina encargada de calcular la temperatura adiabática, es necesario conocer las composiciones en el equilibrio por lo que se le llama repetidamente a la rutina que lo calcula. Dentro de la rutina que calcula el equilibrio químico se llama a una rutina que aplica el método de Newton que resuelve un conjunto de ecuaciones no lineales. Si suponemos que en la resolución del módulo ocurre un error en el nivel mas bajo, es decir durante la aplicación del método de Newton donde por ejemplo no se encontrara la solución al sistema de ecuaciones no lineales, se retorna un código de error al nivel superior siguiente y así sucesivamente hasta el *núcleo*. Es el *núcleo* quien toma la decisión final sobre continuar con la simulación o detener la ejecución.

El manejo por excepción hace uso de la gestión de errores propia de *Delphi*. Una excepción es una condición de error o un evento que interrumpe la ejecución normal de una aplicación, ejemplos de excepciones son la falta de memoria, la división entre cero o los errores en la lectura y escritura de un fichero. Los errores así generados se manejan tal como se describió en el párrafo anterior.



