

## Capítulo 6

### El modelo de programación

El desarrollo de una aplicación engloba varias fases de un proceso complejo que no se limita únicamente a la implementación de una serie de modelos matemáticos (dotados de sentido físico), sino que conlleva abordar temas tanto de forma anterior como posterior. Los primeros, están relacionados con la fase de análisis y diseño de la aplicación; los otros, con la optimización de la programación, las modificaciones o mejoras y la ampliación.

Como podrá entender, una fase no puede soportar toda la carga de la aplicación (no se puede programar a “lo loco, por ejemplo), siendo el conjunto de todas ellas el que hace que una determinada aplicación funcione de forma adecuada y eficiente.

Los Capítulos 6, 7 y 8 se dedican a cada una de estas fases del desarrollo. En éste se plantean las bases de la primera fase: el análisis y diseño de una aplicación; el Capítulo 7 se dedica a la programación en sus términos más implícitos; y, finalmente, en el Capítulo 8 se describen las posibles mejoras, modificaciones y ampliaciones, así como se evalúa de alguna manera la eficiencia de la aplicación.

#### 6.1 Los patrones de diseño

La programación estructura contiene una serie de ventajas muy variadas, junto a los lenguaje OOP, se consigue intensificar éstas y otras nuevas. La principal es la independencia del código y el modelo.

Tome como ejemplo el esquema de la figura 6.1, que representa a una aplicación que permite realizar integrales definidas de una función en un intervalo, y el resultado se muestra por algún medio. Responda a una pregunta: En la figura 6.1, ¿ve usted algún código de programación? La respuesta es obvia: No. Sin embargo, detrás de cada uno de los bloques debe existir para que la aplicación tenga utilidad.

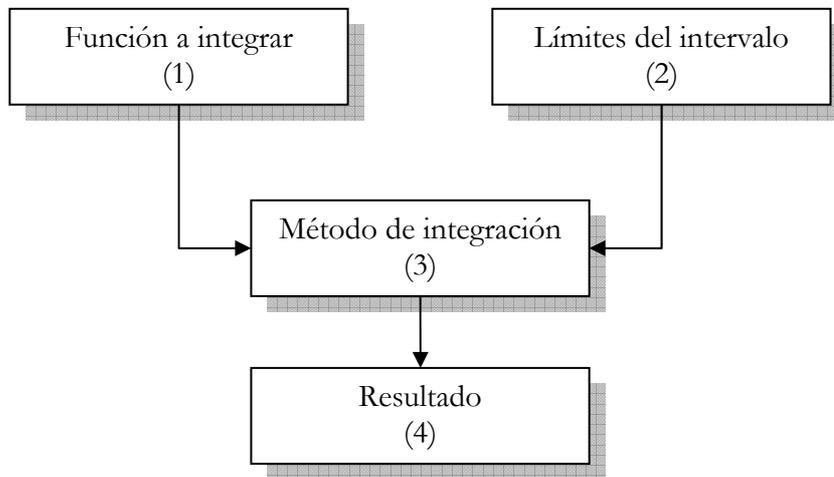


Figura 6.1: Esquema de la programación estructura junto a la programación OOP.

El esquema de la figura anterior representa el modelo, mientras que el código es la implementación de cada uno de los bloques que la componen.

### 6.1.1 El Modelo–Vista–Controlador (MVC)

Con el desarrollo de las técnicas informáticas han ido surgiendo nuevos lenguajes de programación, nuevas capacidades y nuevas tecnologías; pero lo más importante ha sido el desarrollo de nuevos diseños. Un buen diseño permite aprovechar al máximo las posibilidades de cada lenguaje (como ejemplo, basta decir que gran parte del S.O. Windows que actualmente usamos ha sido desarrollado con el lenguaje C).

Un buen diseño, sigue lo que se denomina un “patrón de diseño”, que impone una serie de pautas para que las aplicaciones desarrolladas funcionen de una determinada manera independientemente de su función y utilidad.

El Modelo-Vista-Controlador (MVC) es uno de los patrones de diseño más extendidos en el desarrollo de software. Se trata de una forma de diseñar aplicaciones de manera que se separen cada uno de los componentes:

- Datos
- Interfaz de usuario
- Lógica de control

Esta forma de diseñar aplicaciones permite que modificaciones en el componente de la vista tengan una mínima repercusión en el modelo de datos. El éxito de este patrón de diseño se debe a que muchos programadores realizan una interfaz de usuario muy robusta para conseguir una estabilidad de la aplicación, modificando según cada cliente los componentes de vista y datos.

Por lo general, desarrollar una aplicación siguiendo un patrón del tipo MVC, conlleva la definición de tres módulos separados:

- Modelo
- Vista
- Control

Cada uno de ellos interactúa entre sí dependiendo de las necesidades de la aplicación. Se muestra un diagrama con las comunicaciones entre cada módulo en la figura 6.2.

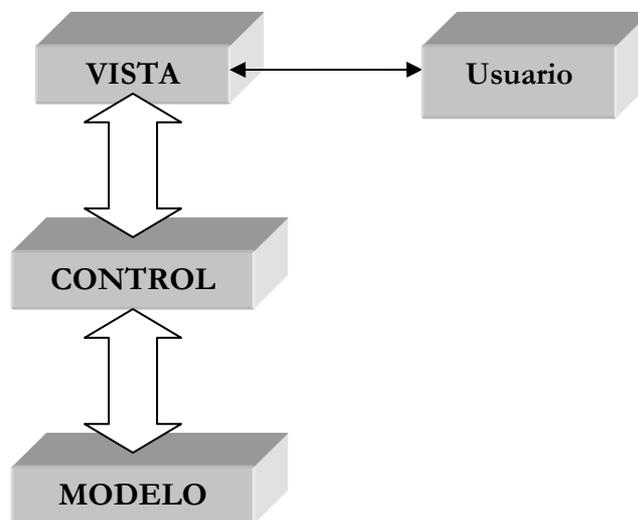


Figura 6.2: Comunicaciones en un patrón de diseño del tipo MVC.

El proceso se inicia cuando el usuario realiza una acción sobre la aplicación (por ejemplo, pulsa un botón), esta acción es recogida por la vista y enviada al controlador que identifica la acción a realizar (p.e., iniciar un proceso); a su vez el control transfiere las órdenes adecuadas al modelo, accediendo a los datos necesarios.

Las páginas web presentan un buen ejemplo para comprender el funcionamiento de un patrón MVC. El usuario interactúa con la aplicación de alguna manera (pulsando un botón para seleccionar un artículo que quiere comprar, por ejemplo), el control recibe la petición enviada por la vista y efectúa las operaciones necesarias para transmitir la petición al modelo (p.e., actualizar la lista de la compra del usuario). El modelo devuelve la información requerida y se realiza el flujo en sentido inverso. En la OOP, el control delega a los objetos de la vista la tarea de actualizar los datos de la interfaz de usuario, de forma que:

- la vista obtiene sus datos del modelo

- el modelo no debe tener conocimiento directo de la vista
- en algunas aplicaciones, la vista y el modelo no pueden comunicarse directamente sino a través del controlador.

Tras ello, la vista queda a la espera de nuevas instrucciones del usuario para repetir el ciclo de comunicaciones.

## 6.2 La programación estructurada en *Fretting Fatigue*

Dentro de cada uno de los módulos que forma la aplicación *Fretting Fatigue* puede observarse algunos criterios comunes a todos ellos y que permiten realizar modificaciones con poco esfuerzo. En este apartado se pretende describir dichos criterios así como ilustrar al programador sobre las tareas realizadas.

### 6.2.1 Esquema general: Módulo Inicial (de Presentación)

El esquema desde la ventana inicial se muestra en la figura 6.3; desde aquí se puede acceder a cada uno de los módulos que componen la aplicación.

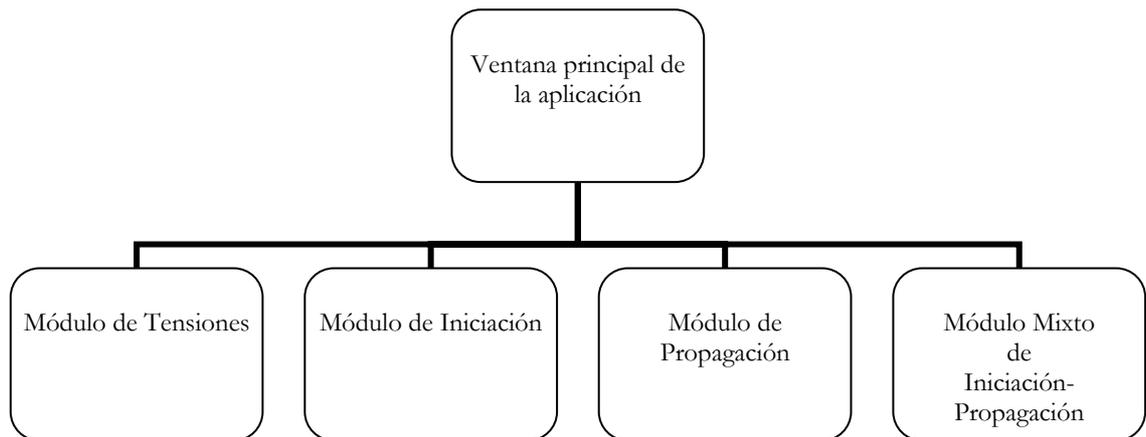


Figura 6.3: Esquema general de la aplicación.

Desde un punto de vista más centrado en la ventana principal, la figura 6.4 muestra las posibles opciones que pueden darse. Las flechas indican el sentido de la información.

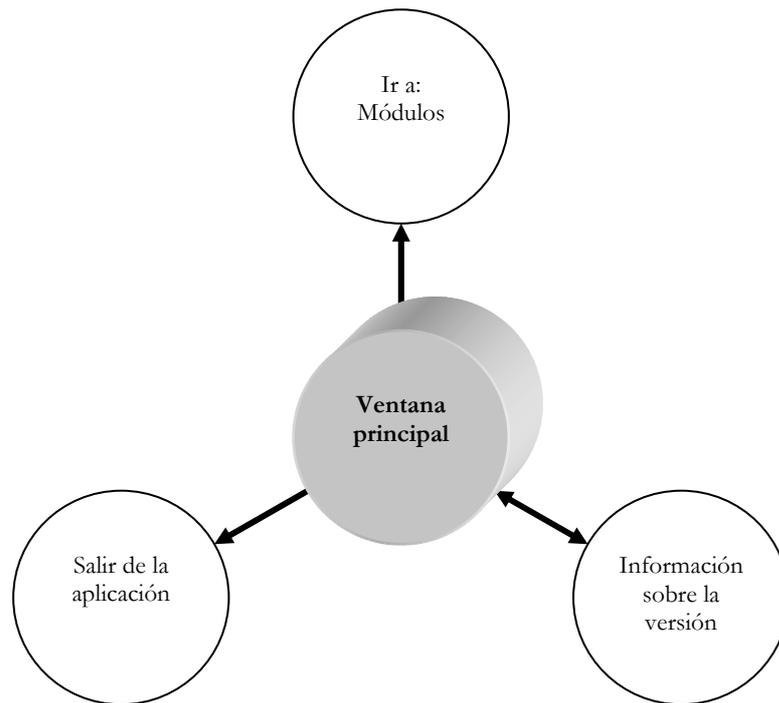


Figura 6.4: Opciones de la ventana principal.

### 6.2.2 Módulo de Tensiones

La ventana inicial no tiene mayor interés, aquí que se describe a continuación cada uno de los módulos implementados. El Módulo de Tensiones es uno de los más importantes desarrollados, ya que sobre él se sustenta una parte importante del funcionamiento de la aplicación: el fichero de Datos Iniciales.

En la figura 6.5 se muestra las distintas posibilidades que permite el módulo, así como la dirección de las comunicaciones que tienen lugar durante el proceso de cálculo.

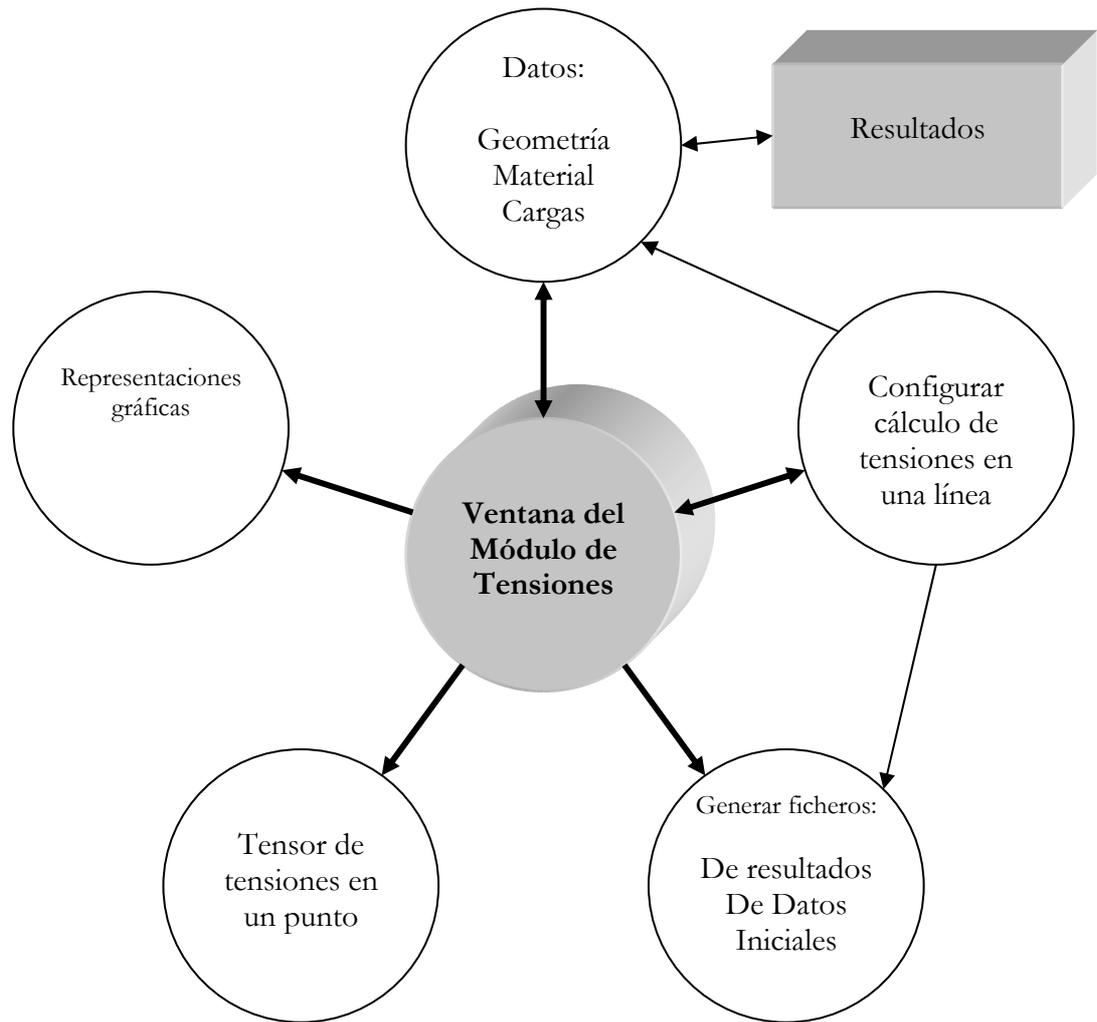


Figura 6.5: Esquema del modelos para el Módulo de Tensiones.

Puede observarse en la figura 6.5 que desde la ventana del módulo puede configurarse los datos referentes a la geometría, las cargas y el material; así como el método de cálculo y la forma de obtener los resultados (de forma gráfica, en un fichero o de ambas formas).

### 6.2.3 Módulo de Iniciación

Este módulo depende directamente del anterior, en el sentido de que trabaja con el fichero de Datos Iniciales generados por aquél.

Su estructura, así mismo, es bastante modular, ya que cada uno de los criterios posee un formulario propio para obtener los resultados. Dependiendo de cada criterio, las opciones de resultados varían.

La figura 6.6 muestra un esquema del modelo de programación; pueden observarse las distintas posibilidades, así como la repetición de las partes, lo que favorece su implementación informática.

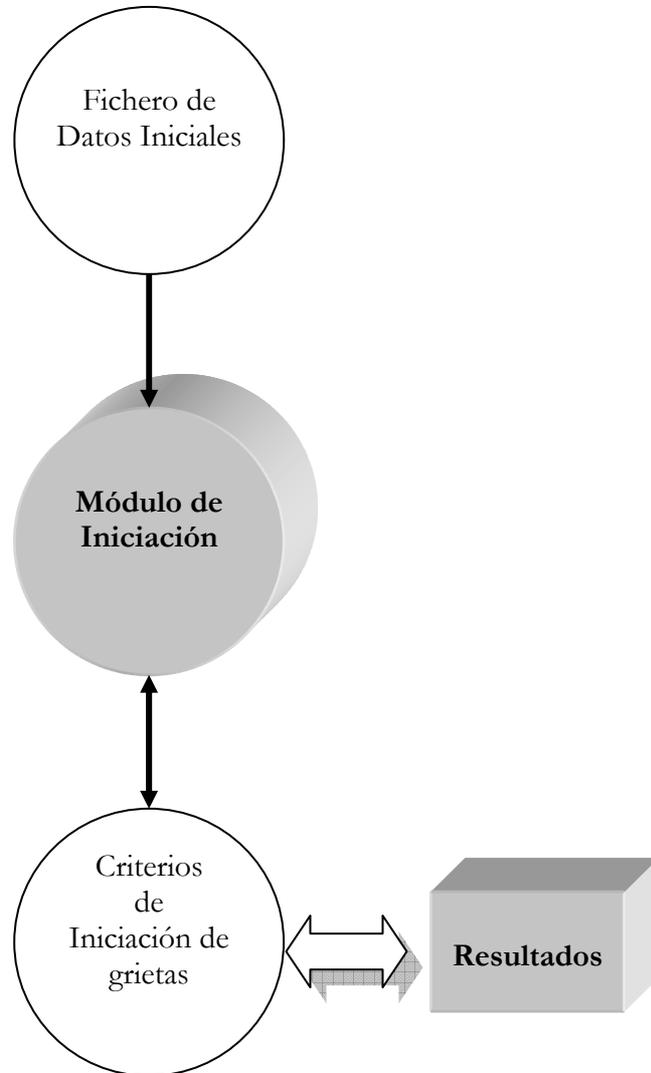


Figura 6.6: Esquema del modelo del Módulo de Tensiones.

#### 6.2.4 Módulo de Propagación

En este caso, la principal diferencia es la cantidad de información necesaria para proceder a calcular los resultados. Está por un lado el fichero de Datos Iniciales, y por otro las propiedades correspondientes a la longitud de iniciación, el espesor de probeta, el camino y el punto de iniciación de grietas. Cada una de éstas dispone de un método adecuado de adquisición.

La figura 6.7 muestra el esquema correspondiente a este módulo. Observe que se requiere mucha información para iniciar el cálculo. Por lo demás, no conlleva mayor complicación.

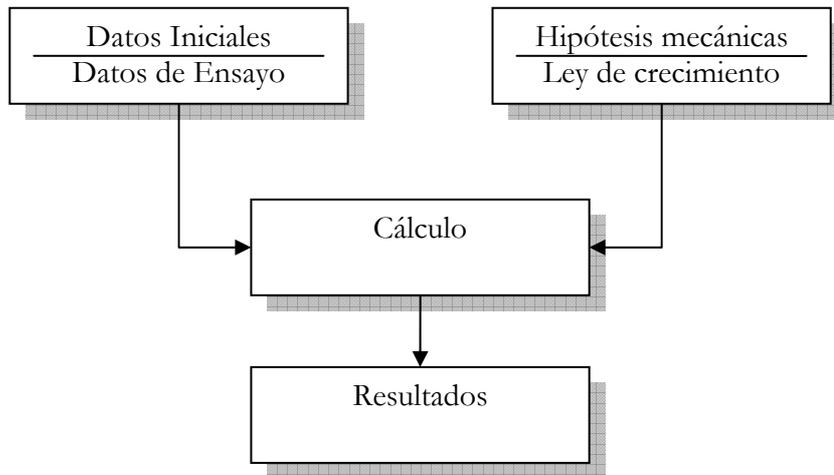


Figura 6.7: Esquema del modelo del Módulo de Propagación

### 6.2.5 Módulo Mixto

Para finalizar el capítulo se aborda el modelo del principal módulo de la aplicación. Éste tiene de particular que puede operar a partir de dos datos diferentes: desde un fichero de Datos Iniciales (un único ensayo) y desde un fichero de Datos de Ensayos (varios ensayos).

En la figura 6.8 puede observarse que el esquema del modelo correspondiente en el que se pueden distinguir dos líneas de cálculo que en apariencia discurren de forma paralela. Cada una de ellas representa las posibilidades del módulo de trabajar con un único ensayo o con varios de forma automática.

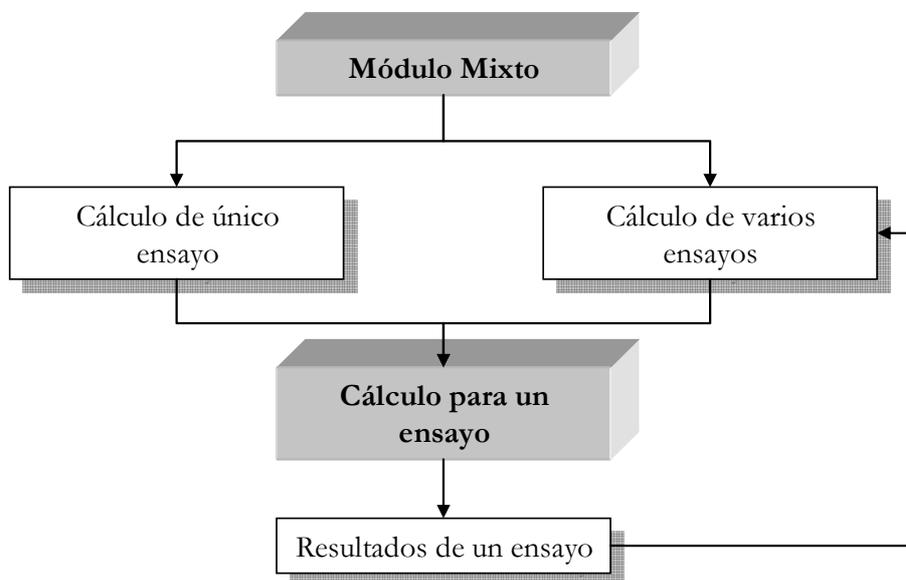


Figura 6.8: Esquema del modelo del Módulo Mixto.