

4. Estructura del patrón Decorador (*Decorator*).

5. Estructura del patrón Fábrica Abstracta (*Abstract Factory*)

6. Estructura del patrón Visitante (*Visitor*). Usando un diagrama de secuencia, presente las colaboraciones típicas entre sus participantes.

Parte II: Diseño

Se desea construir un *sistema de control de versiones*. La idea central del sistema es el concepto de *Proyecto*, una colección de información relacionada con un desarrollo concreto. Los proyectos tienen *Versiones* y cada versión agrupa una serie de *Ficheros*, organizados de forma jerárquica en *Directorios*. Cada fichero permite conocer su contenido (como una cadena de caracteres), su tamaño, la fecha de su última modificación y qué *Usuario* la ha realizado.

Dentro de una versión, las sucesivas *Modificaciones* de un determinado fichero se consiguen sustituyendo el fichero original por una combinación del fichero original y la diferencia introducida. Hay dos tipos de modificaciones a considerar: (a) *Añadir* una cadena de caracteres a partir de un punto del fichero y *Eliminar* una serie de caracteres a partir de un punto del fichero. A mayores del cambio realizado, resulta interesante conocer qué usuario ha realizado la modificación y cuándo.

Cada proyecto tiene una *Versión Actual*, que se encuentra *Abierta*, esto es, que puede sufrir modificaciones. El resto de versiones del proyecto se encuentran *Cerradas*, es decir, disponibles sólo para lectura. Normalmente, cualquier usuario puede leer los ficheros de la versión actual, salvo que un usuario concreto haya *Bloqueado* la versión, en cuyo caso sólo este usuario estará autorizado a leer los ficheros. Las modificaciones del contenido de la versión, por otro lado, sólo pueden ser realizadas cuando la versión está bloqueada. El proceso de cierre de la versión actual, sólo posible cuando la versión esté abierta, implica la creación de una nueva versión actual con un contenido idéntico al de la versión previa. Es interesante, dada una versión, conocer cuál fue la versión que la originó.

Para posibilitar la búsqueda de proyectos en el sistema, se define un *Motor de Búsqueda* en el que se permite recorrer, a partir de un cierto *Criterio*, todos los proyectos relevantes. Uno de los criterios más interesantes es determinar aquellos proyectos modificados después de una determinada fecha o aquellos que tienen bloqueada la versión actual.

Dado que el control de versiones debe integrarse con distintos tipos de interfaz de usuario, se requiere que se defina un mecanismo que permita la notificación de los cambios a las posibles vistas del sistema. De la misma forma, se sugiere la inclusión de algún mecanismo que permita representar de forma genérica las operaciones que se pueden realizar sobre el sistema para facilitar en el futuro el desarrollo de dichas posibles interfaces de usuario.

Se pide:

- Diseño del sistema, justificando sus decisiones en base a los patrones empleados