**Juan Diego Gómez Triana**  
**Patrones de software - Corte 2**  
**Documentación patrón Fábrica Abstracta**

El patrón Abstract Factory permite crear familias de objetos relacionados sin especificar sus clases concretas. Funciona a través de una interfaz que declara métodos de creación para diferentes productos (por ejemplo, botones y cuadros de texto), y luego cada subclase concreta implementa estos métodos para retornar los objetos de acuerdo a un estilo o configuración específica.

Este patrón se usa principalmente cuando se desea que un sistema sea independiente de cómo se crean, componen y representan sus objetos. De esta forma, se logra una separación clara entre la lógica de creación de los objetos y su utilización dentro del programa, permitiendo así una mayor flexibilidad y escalabilidad en el diseño del software.

En el ejemplo desarrollado con C# con Windows Forms, se implementa una aplicación que permite cambiar visualmente entre dos temas: Light (claro) y Dark (oscuro). Para ello, se definen dos tipos de productos: IButton y ITextBox con cada tipo de tema, que representan los controles gráficos de la interfaz. Luego se crean dos fábricas concretas (LightFactory y DarkFactory) que implementan la interfaz IUIFactory para construir los objetos con los estilos correspondientes.

Cuando el usuario elige un tema desde un ComboBox y presiona el botón de generar UI, se instancia la fábrica adecuada según la selección. Esta fábrica luego produce un botón y un cuadro de texto con el estilo elegido, los cuales se agregan dinámicamente al formulario principal. Así se demuestra cómo el patrón Abstract Factory permite crear componentes con diferentes estilos sin modificar el código que los utiliza, manteniendo el principio de abierto/cerrado de la programación orientada a objetos.