**TEMA6**

**1. Decimos que en un sistema PDV una clase Venta es responsable de crear instancias de la clase LineaDeVenta:**

→ Porque según el patrón Creador tiene información para inicializar LineaDeVenta.

→ Porque según el patrón Experto Venta tiene la información para llevar a cabo la responsabilidad.

**→ Porque según el patrón Creador de Venta de hecho contiene LineasDeVenta y cumple una de las condiciones para ser creador.**

→ Porque según el patrón Factoría Venta es una clase que nos inventamos para asignarle las responsabilidades de creación

**2.Cuando decidimos que Registro reciba todos los mensajes externos del sistema:**

→ Estamos utilizando el patrón Controlador que separa la Vista del Modelo y además el patrón Fabricación Pura porque Registro es una clase que no estaba en el Modelo de Dominio.

**→ Estamos utilizando el patrón Controlador de Fachada ya que separa la Vista del Modelo y además Registro que ya estaba en el Modelo de Dominio y represenat ael conjunto.**

→ Estamos siguiendo el Patrón Controlador de Casos de Uso ya que Registro es una Fabricación Pura que sirve para manejar todos los eventos de sistema para este caso de uso.

**3.Un adaptador es:**

**→ Sigue el patrón Fabricación Pura pues es una clase que no estaba en el Modelo de Dominio.**

**→ Sigue el patrón Indirección pues su propósito es desacoplar el sistema en desarrollo de sistemas de terceras partes.**

→ Sigue el patrón Estrategia pues es una política que puede cambiar.

**4.En el sistema de PDV una clase que sirva para aplicar reglas de descuento.**

**→ Sigue el patrón Estrategia y es una Fabricación Pura.**

→ Sigue el patrón Estrategia y es un Adaptador.

→ Sigue el patrón Estrategia y es una Indirección.

**→ Sigue el patrón Estrategia y es un mecanismo de Variaciones Protegidas.**

**→ Sigue el patrón Estrategia y hace uso del Polimorfismo dado que todas las estrategias de descuento implementan la misma interfaz.**

**5.El patrón Factoría se utiliza para crear objetos cuando existe o una lófica compleja o deseo de separar responsabilidades.**

**→ Por eso los Adaptadores siempre se crean desde una Factoría.**

→ Los Adaptadores no se crean desde una Factoría pero las Estrategias sí.

→ Los Estrategias no se crean desde una Factoría pero los Adaptadores sí.

**→ Tanto las Estrategias como los Adaptadores se crean desde una Factoría.**

**6.El patrón Composite:**

→ Se usa por ejemplo siempre que aparezca un multiobjeto en la interacción a la hora de asignar las responsabilidades.

**→ Sólo cuando objetos atómicos y compuestos implementan la misma interfaz.**