UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN 

FACULTAD DE CIENCIAS FISICO MATEMATICAS

Diseño Orientado a Objetos

Investigacion AntiPateones

Maestro: Miguel Salazar

INTEGRANTE

Gilberto Alejandro Contreras Silva – 1683471

GRUPO: 00 AULA: 409

FECHA: Monterrey, N.L., 28 de abril de 2017

**Antipatrones**

También definen un vocabulario de la industria para resolver procesos comunes y buscar una manera de solucionarlo. La diferencia es que son malas soluciones, estos parecen ser buenas ideas al momento de pensarlas, más sin embargo a la hora de implementarlas estas fallan. Este puede ser creado por un desarrollador o administrador con no tanto conocimiento en el tema. Por eso es que los anti patrones proveen experiencia en el mundo real, con los problemas recurrentes que se puedan presentar. Los anti patrones proporcionan herramientas para soluciones comunes en la industria de software. Algunas de las platillas de anti patrones:

*The Blob:*

Es una sola clase que hace todo, lo que significa que es una clase grande, contiene muchos métodos no relacionados y sin argumentos. Y puede tener como consecuencia la pérdida de las ventajas de la orientación a objetos. Hay una dependencia total a esa clase. Y tiene difícil rehúso o testeo.

*Lava Flow:*

Grandes cantidades de código y con poca claridad de su función. Se vuelve complicado corregir los problemas ya que el desorden va creciendo geométricamente. Como identificar: Grandes cantidades de código y con poca claridad de su función. Se vuelve complicado corregir los problemas ya que el desorden va creciendo geométricamente. Y las consecuencias que esto puede causar: Mala presentación en el sistema. Dificultad para extenderlo o depurarlo. En caso de ser rehusado el problema se traslada.

*Poltergeists:*

Emplear objetos cuyo único propósito es pasar la información a terceros objetos. Se identifica porque tiene un gran número de clases pequeñas poco descriptivas, con usos limitados o responsabilidades muy puntuales y tiene como consecuencia una mala presentación del sistema.

*Golden Hammer:*

Se refiere al uso de una tecnología, patrón, arquitectura etc... Para cualquier problema o situación incluso cuando es evidente que no va ser útil. Por lo que las soluciones encontradas son inferiores en calidad a otras existentes.

*Spaghetti Code:*

Código en el cual el flujo de control es muy variado y complicado. En OO, los objetos son básicamente funciones con poca interacción entre ellos. Por lo que un cuerpo de código realizando más de una función, es más fácil reescribir todo que modificar algo, falta documentación. Y provoca que cerca del 50% del tiempo de las operaciones de mantenimiento de este tipo de aplicaciones se invierte en descubrir cómo funcionan.