

The background is a teal color with a white circuit board pattern. A diagonal white line runs from the top-left corner to the bottom-right corner, dividing the image into two triangular sections. The word "ANTIPATRONES" is written in white, bold, uppercase letters across the center of the image.

**ANTIPATRONES**

# OBJETIVOS

- ① Comprender qué son los antipatrones.
- ① Comprender qué es el *refactoring*.
- ① Identificar el uso de antipatrones comunes.

Dennis S. Cohn Muroy, Mag. Ing.  
Pontificia Universidad Católica del Perú

# ¿QUÉ SON LOS ANTIPATRONES?

Tomémonos unos minutos ...





1

# DEFINICIONES

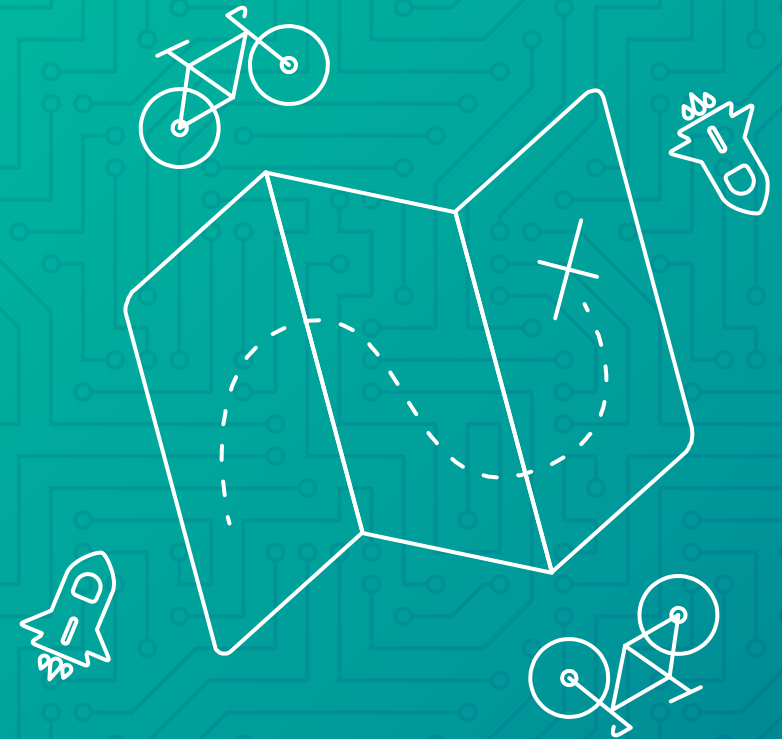
¿Qué es un Antipatrón?

# ¿QUÉ ES UN ANTIPATRÓN?

- ⦿ Es una “solución” a un problema que tiene consecuencias negativas inesperadas.
- ⦿ Las consecuencias negativas se suelen presentar en etapas posteriores a la puesta en producción de la solución.
  - ⦿ Corrección de errores.
  - ⦿ Implementación de nuevas funcionalidades.
  - ⦿ Reemplazo de componentes.

# SE REQUIEREN 2 SOLUCIONES

Solución inicial & la solución que corrige las consecuencias negativas.



# ORDEN DE EVENTOS

Érase una vez ...

Se detecta un problema a ser solucionado.

SOLUCIÓN 1

Consecuencias negativas

Se genera un problema como consecuencia de haber aplicado el antipatrón.

REFACTORING

PROBLEMA INICIAL

Antipatrón

Se aplica un antipatrón para resolver el problema inicial.  
Se ve bien ... por ahora.

NUEVO PROBLEMA

Refactoring

Remedia los problemas generados por la Solución 1.

# 2

## ANTIPATRONES PARA EL DESARROLLO

A continuación, algunos de los antipatrones más usados ...



# SPAGHETTI CODE

Pieza de código compleja y no documentada que es difícil de mantener.

**Causas:** Desconocimiento, Flojera.



# ¿CUÁLES SON LOS SÍNTOMAS DEL SPAGHETTI CODE?

- Muchos métodos de objetos sin parámetros.
- Variables globales innecesarias.
- Los nombres de los objetos son nombres de procesos.
- Relación mínima entre objetos.
- Dificultad para reutilizar el código.
- No se utiliza ni herencia ni polimorfismo.



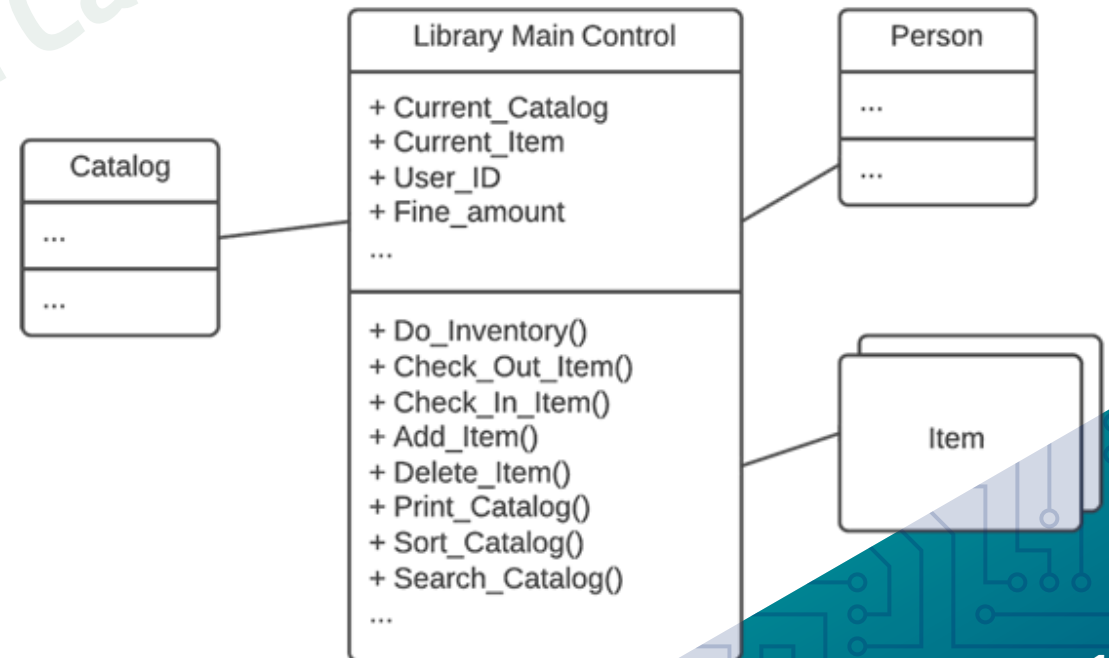
Es más fácil volver a  
implementar esta función  
que modificarla.

(Todos los programadores, alguna vez)

# THE BLOB

Una clase contiene un gran número de responsabilidades que deberían de recaer en otras clases.

**Causas:** Prisa, Flojera.



# ¿CUÁLES SON LOS SÍNTOMAS DEL THE BLOB?

- Una clase con muchos atributos y métodos (> 60).
- Colección de atributos no relacionados.
- Carencia de un diseño orientado a objetos.
- Relación mínima entre objetos.
- Dificultad para reutilizar el código.



Esta clase es el corazón de  
nuestro diseño/arquitectura.

(Todos los programadores, alguna vez)

# CUT & PASTE

“Reutilización” de código a través del cortado (o copiado) y pegado del código fuente.

Causas: Flojera.



# ¿CUÁLES SON LOS SÍNTOMAS DEL CUT & PASTE?

- Se sigue presentando el mismo error en varias secciones de la aplicación.
- Mayor costo de mantenimiento.

Dennis S. Cohn Muroy, Mag. Ing.  
Pontificia Universidad Católica del Perú





¿Ese error no había sido  
corregido? ¿Por qué vuelve  
a pasar?

(Los clientes/usuarios, alguna vez)

# GOLDEN HAMMER

Construir toda una solución/soluciones alrededor de una única herramienta.

**Causas:** Ignorancia, Orgullo, Mente Cerrada.



# ¿CUÁLES SON LOS SÍNTOMAS DEL GOLDEN HAMMER?

- ⦿ Problemas de desempeño y escalabilidad.
- ⦿ Arquitectura es descrita por un único producto.
- ⦿ Requerimientos acomodados al producto.
- ⦿ Conocimiento muy específico.



Nuestra base de datos es  
nuestra arquitectura.

(Ex-arquitecto de software, alguna vez)

# LAVA FLOW

Desarrollos legados sin documentación o capacidades para ser mantenidos.

**Causas:** Avaricia, Envidia, Flojera.



# ¿CUÁLES SON LOS SÍNTOMAS DEL LAVA FLOW?

- Falta de documentación de estructuras y funciones complejas.
- Proyectos académicos o experimentales desplegados en ambientes productivos.
- Variables y código sin justificación.
- Código comentado sin justificación.



¡Oh! Esa funcionalidad fue implementada por 'X' (ya no trabaja en la empresa); luego la estuvo viendo 'Y' (fue transferido a otra área). Mejor no la toques ... ¡Suerte!

(En todo lugar, alguna vez)

# OBSOLECENCIA CONTINUA

La tecnología cambia con rapidez.

Dificultades para actualizar las librerías de los sistemas.

**Causas:** Entorno cambiante.





# ¿CUÁLES SON LOS SÍNTOMAS DE LA OBSOLECENCIA CONTINUA?

- ⦿ Poco uso de estándares abiertos (consenso de la industria).
- ⦿ No se ha actualizado la aplicación en un tiempo.

Dennis S. Cohn Muro, Mag. Ing.  
Pontificia Universidad Católica del Perú



¿Se tienen que actualizar las  
librerías de nuestros programas?

(Todos los programadores, alguna vez)

3

# REFERENCIAS

# BIBLIOGRAFÍA

- Freeman, E., & Robson, E. (2020). Head First Design Patterns. O'Reilly Media.
- Antipatterns. (n.d.-b). <https://sourcemaking.com/antipatterns>

## Créditos:

- Plantilla de la presentación por [SlidesCarnival](#)
- Fotografías por [Unsplash](#)
- Diseño del fondo [Hero Patterns](#)