

[Inicio](#) » [Ciencia](#) » [Computación](#) » Ley de Demeter : No hables con extraños

COMPUTACIÓN

Ley de Demeter : No hables con extraños

por [Jose Ramon Pascual](#) | Publicada [10 junio, 2019](#) | [1 Comentario](#)

La Ley de Demeter ([LoD](#) por las siglas en inglés) o **Principio del Menor Conocimiento**, es una buena práctica de diseño que busca minimizar el acoplamiento entre componentes en programación orientada a objetos.

Lo podríamos llamar Principio de Endogamia, porque favorece la relación con lo que está dentro y no lo de fuera, pero se suele asociar con la frase «No hables con extraños». Y esto es porque, a partir de ciertas reglas sencillas, lo que busca es que un objeto solo se relacione consigo mismo todo lo posible. O bien, asuma tan poco como sea posible la estructura o propiedades de cualquier otro.

Es decir, intentar que cada objeto sepa lo menos posible de otros ajenos. De ahí lo de Principio del Menor Conocimiento.

En resumen, la Ley de Demeter dice que un método M de un objeto O solo puede llamar a métodos de:

- El propio Objeto O.
- Un argumento del método. Los parámetros que recibe el método.
- Cualquier objeto creado/instanciado dentro del propio método. Es decir, las variables locales al método.
- Cualquier propiedad/campo del Objeto O.

**Disrupción Tecnológica***Blog de divulgación de Ciencia y Tecnología*

Es decir, no llamar a métodos de los objetos devueltos por otros métodos.

Para mas información este interesante artículo de David Bock titulado «[The Paperboy, The Wallet and The Law of Demeter](#)»

Ejemplo

Por ejemplo, hay que evitar las cadenas de métodos de la forma:

```
objetoX.metodoDeX().metodoDeY().metodoDeZ();
```

Que expone el código a cambios en la clase del ObjetoX, los cambios en la clase del ObjetoY, y en la del ObjetoZ.

Además de los problemas de mantenibilidad del código, crea en la clase X una dependencia de la clase Y y Z. Esta dependencia es acoplamiento de la clase X a la estructura interna que está asumiendo de la clase Y y Z. Para usar la clase X necesitas también la Y y la Z.

Para estas llamadas se debería encapsular el acceso a las propiedades/métodos que se necesitan.

Por ejemplo, en la clase X debería existir un método que devuelva la composición:

```
objetoX.metodoXYZ();
```

Ventajas de aplicar la Ley de Demeter

Los beneficios de usar esta ley son evidentes en cuanto a la eliminación del acoplamiento, pero se usa bastante poco por la cantidad de código a generar para redirigir llamadas a componentes internos del objeto.

- Limitar el acoplamiento y dependencias entre clases.
- Código más fácil de probar.
- Código mas mantenible y robusto.
- Mas sencillo de reutilizar.

algoritmos

buenas prácticas

AUTOR



Jose Ramon Pascual

Ingeniero Informático de Sistemas por la Universidad de Málaga. Grado en Ingeniería Informática por la U. Internacional de La Rioja. Máster en Dirección de Proyectos, y Posgrados en Biotecnología, Bioingeniería, e-Health y Telemedicina por UNED. Comencé trabajando en automatización industrial en el sector de automoción, pero pronto pasé al sector sanidad, donde he trabajado durante 13 años en proyectos de sistemas de información vinculados a 79 hospitales de España de los servicios de sanidad de 15 comunidades autónomas, y Portugal. Tras 10 años de experiencia en interoperabilidad en sanidad con HL7 y **Mirth Connect**, he decidido divulgar experiencias sobre este motor.

TAMBIÉN TE PUEDE INTERESAR



Deja un comentario

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con *



☐ GUARDA MI NOMBRE, CORREO ELECTRÓNICO Y WEB EN ESTE NAVEGADOR PARA LA PRÓXIMA VEZ QUE COMENTE.

☐ RECIBIR UN CORREO ELECTRÓNICO CON LOS SIGUIENTES COMENTARIOS A ESTA ENTRADA.

☐ RECIBIR UN CORREO ELECTRÓNICO CON CADA NUEVA ENTRADA.

PUBLICAR EL COMENTARIO

Una idea sobre “Ley de Demeter : No hables con extraños”

1 Comentario

Jim

2 abril, 2020, 11:43 am

Thanks for your personal marvelous posting! I really enjoyed reading it, you happen to be a great author.

I will be sure to bookmark your blog and definitely will come back in the future. I want to encourage yourself to continue your great writing, have a nice evening!

< [DÍA MUNDIAL DE LOS OCÉANOS](#)

[ACOPLAMIENTO Y COHESIÓN](#) >

Etiquetas

- Agujero negro (1)
- algoritmos (3)
- Arquitectura (3)
- Basura espacial (1)
- buenas prácticas (3)
- Cambio horario (2)
- carcinógenos (1)
- Como (8)
- Database (4)
- desinfección (1)
- e-Salud (7)
- ecologismo (5)
- epidemia (1)
- ER7 (2)
- Error (1)
- Event horizon telescope (1)
- eventos (5)
- excel (1)
- HL7 (7)
- Horizonte de sucesos (1)
- Howto (8)
- Interoperabilidad (17)
- IOT (1)
- JDBC (1)
- Layer (1)
- m87 (1)
- Mirth Connect (10)
- modularización (1)
- Open Source (5)
- optimización (2)
- Oracle (4)
- PL/SQL (2)
- productos químicos (1)
- Raspberry (1)
- Resiliencia (1)
- riesgos (1)
- Salud (6)
- SOA (2)
- SQL (2)
- tecnologica (1)
- toxicidad (1)
- Transcripción (1)
- vacunas (1)
- Web Service (2)
- XML (2)