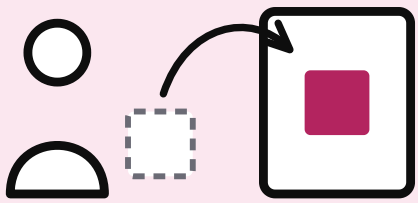
Concepto

Ley de Tesler

Toda aplicación tiene una parte de complejidad que no se puede eliminar, solo trasladar. La cuestión es quién se ocupa de ella: la persona o el equipo que la crea.



En inglés

Tesler's law (law of conservation of complexity)

Origen

Larry Tesler (hacia 1984, según él mismo); difundida por Tognazzini (1998) y Saffer (2010)

Qué problema aborda

Explica por qué simplificar una interfaz no hace desaparecer el trabajo: alguien tiene que hacerlo, y conviene que no sea quien la usa.

Un ejemplo

Un ejemplo ilustrativo: escribir la dirección para un envío.

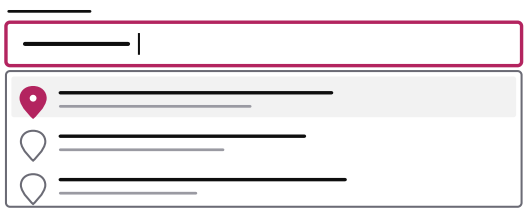
La complejidad, para la persona



Cinco campos: el trabajo es de la persona

Cinco campos que llenar y un formato que adivinar. **Cada persona hace el trabajo.**

La complejidad, para el sistema



Un campo: el sistema completa el resto

Un solo campo que propone la dirección completa mientras escribes. **El trabajo lo hace el sistema.**

La dirección sigue siendo igual de compleja: solo cambió quién se ocupa de ella.

Por qué pasa

1 

Cada tarea tiene una parte de complejidad que no se puede quitar.

2 

Si el sistema no se ocupa de ella, tiene que ocuparse la persona.

3 

Mejor que el equipo la resuelva una vez que cada persona cada vez que usa el producto.

Aplicaciones en diseño

Úsalo así

Antes de pedir un dato o una decisión, pregúntate si el sistema podría resolverlo por la persona.

Evita

Dar por simplificado algo que solo se ha escondido y que la persona tendrá que resolver después.

Para no caer

Trasladar no es quitar el control: si el sistema decide por la persona, que pueda ver y corregir lo que hizo.

Hacia 1984

Lo que dice la investigación

Larry Tesler formuló la ley hacia 1984: toda aplicación tiene una complejidad que no se puede reducir, y la cuestión es si la asume la persona, quien desarrolla la aplicación o quien desarrolla la plataforma (Tesler, s. f.).

● ○ ○ **Estudio único.** No hay estudios que la midan: es un principio de la práctica, conocido por el propio Tesler y por quienes lo citaron. Tognazzini (1998) añade un matiz: cuando algo se simplifica, la gente se anima con tareas más complejas.

Conexiones

Se usa en

- Diseño de interacción

Relacionado con

- Carga cognitiva
- Divulgación progresiva
- Opciones por defecto

Autores y obras

hacia 1984	**Larry Tesler**	Formuló la ley de conservación de la complejidad; no la publicó en un artículo de la época.
1998	**Bruce Tognazzini**	La citó en su columna «The Complexity Paradox» y añadió que, al simplificar, la gente busca tareas más difíciles.
2010	**Dan Saffer**	Recogió en Designing for Interaction una entrevista en la que Tesler explica la ley y defiende que la complejidad la asuma quien programa, no quien usa.
2020	**Jon Yablonski**	La reunió con otras leyes de la experiencia de usuario en el libro Laws of UX.

Fuentes

Fuentes originales

- Tesler, L. (s. f.). The law of conservation of complexity (Tesler's law). Larry Tesler (sitio web mantenido por su familia). Ver fuente (https://www.nomodes.com/computing-history)

Fuentes secundarias

- Tognazzini, B. (1998, septiembre). The complexity paradox. AskTog. Ver fuente (https://asktog.com/columns/011complexity.html)
- Saffer, D. (2010). Designing for interaction: Creating innovative applications and devices (2.ª ed.). New Riders.
- Yablonski, J. (2020). Laws of UX: Using psychology to design better products & services. O'Reilly Media. Ver fuente (https://www.oreilly.com/library/view/laws-of-ux/9781492055303/)

Última revisión: 26 de octubre de 2026

Atlas público de diseño, de Adriana Ojeda. Texto e ilustraciones bajo licencia CC BY 4.0.