INTERACCIÓN Y EXPERIENCIA DE USUARIO

End-user development

End-User Programming

De utilizar a crear

01



Necesidad propia

Detectar un problema

→

02



Construcción

Adaptar una herramienta

→

03



Solución

Probar en contexto

¿Por qué *end-user*?

¿Por qué algunos ingenieros prefieren Python y otros Excel?

End-user programming

01



Usuario

Conoce la tarea

→

02



Programa

Expresa la solución

→

03



Uso real

Verifica el resultado

Alternativas al texto: Representaciones accesibles y límites de la abstracción



Why Source Code is Always Text | Dan Vanderkam

SquiggleConf

- Why Source Code is Always Text: alternativas al texto, desde 14:30

Computational Thinking

¿Recuerdan a Vannevar Bush?



A SCIENTIST OF THE FUTURE RECORDS EXPERIMENTS WITH A TINY CAMERA FITTED WITH UNIVERSAL-FOCUS LENS. THE SMALL SQUARE IN THE EYEGLASS AT THE LEFT SIGHTS THE OBJECT

AS WE MAY THINK

A TOP U. S. SCIENTIST FORESEES A POSSIBLE FUTURE WORLD IN WHICH MAN-MADE MACHINES WILL START TO THINK

by VANNEVAR BUSH

DIRECTOR OF THE OFFICE OF SCIENTIFIC RESEARCH AND DEVELOPMENT

Condensed from The Atlantic Monthly, July 1945

This has not been a scientists' war; it has been a war in which all have had a part. The scientists, burying their old professional competition in the demand of a common cause, have shared greatly and learned much. It has been exhilarating to work in effective partnership. What are the scientists to do next?

For the biologists, and particularly for the medical scientists, there can be little indecision, for their war work has hardly required them to leave the old paths. Many indeed have been able to carry on their war research in their

new, and the effort to bridge between disciplines is correspondingly superficial.

Professionally our methods of transmitting and reviewing the results of research are generations old and by now are totally inadequate for their purpose. If the aggregate time spent in writing scholarly works and in reading them could be evaluated, the ratio between these amounts of time might well be startling. Those who conscientiously attempt to keep abreast of current thought, even in restricted fields, by close

Pensamiento computacional

01



Descomponer

Dividir el problema

Separar una tarea compleja en problemas manejables.

→

02



Abstraer

Elegir lo relevante

Conservar los datos y relaciones que necesita la solución.

→

03



Algoritmo

Ordenar los pasos

Definir una secuencia de operaciones que pueda ejecutarse.

→

04



Evaluar

Probar y corregir

Observar el uso y valorar la experiencia de las personas.

Complejidad: poder razonar sobre lo construido



Why Can't We Make Simple Software? - Peter van Hardenberg

Handmade Cities

- 1:40
- 33:20

Low code

Low code

01



Bloques · composición

Configuración visual

Componer lo habitual

Combinar componentes y ajustar sus propiedades.

02



Código · instrucciones

Código

Resolver lo específico

Expresar operaciones que la configuración no cubre.

Low code: Poder bajar de nivel cuando la herramienta no basta



Why Source Code is Always Text | Dan Vanderkam

SquiggleConf

- Why Source Code is Always Text: abstracciones y herramientas, desde 20:40

Embodiment o Encarnación

¿Excel usa alguna encarnación?

Embodiment: pensar con el cuerpo



Manipular → percibir → comprender


score: 0.02
estat: 0.06
hbond: 0.00
vdw: -0.08
ds: 0.04

Programación literaria

Programación literaria

01



Explicación

Por qué y para quién

02

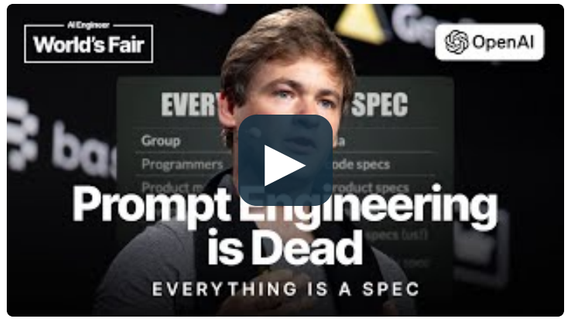


Código

Cómo se ejecuta

Expresar operaciones que la configuración no cubre.

Especificación: Conservar la intención además del resultado generado



The New Code — Sean Grove, OpenAI

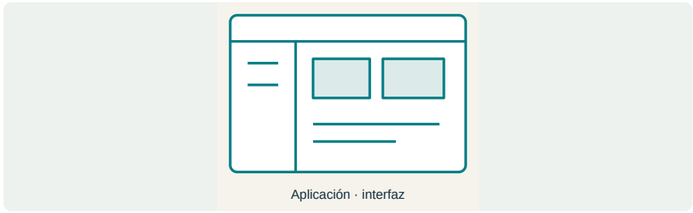
AI Engineer

- The New Code: conservar la especificación, desde 5:08

Herramientas

Herramientas GUI

01

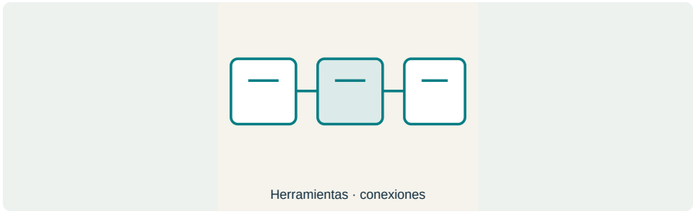


Elegir

Un componente

→

02

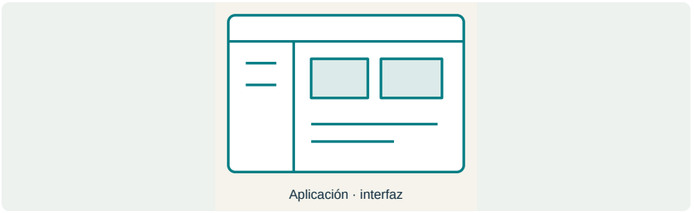


Configurar

Sus propiedades

→

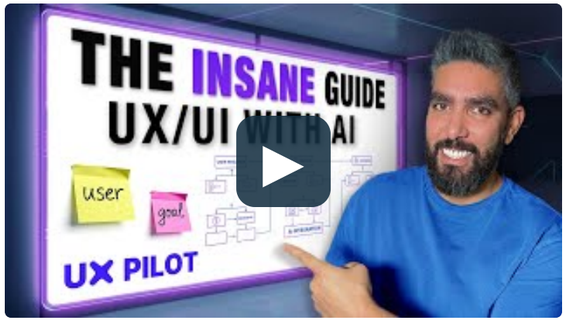
03



Probar

La interacción

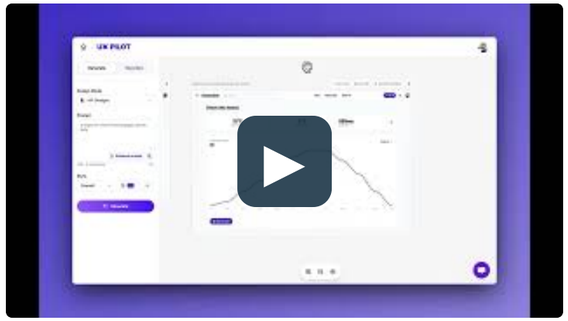
Diseño asistido por IA: de la intención a pantallas revisables



Design An App With AI | UX Pilot Tips & Tricks

AI Tooltip

- idea, 1:41
- wireframes, 7:47
- documentación, 12:06



Demo of UX Pilot AI

UX Pilot AI

Visionado complementario · sin transcripción disponible

Lenguajes de bloques

01  Al recibir un evento

02  Consultar datos

03  Tomar una decisión

04  Actualizar la interfaz

Lenguajes de flujo de datos

01



Entrada

Datos

Facilitar que la persona comunique lo que quiere conseguir.

→

02



Transformación

Operación

→

03



Salida

Resultado

Mostrar resultados que puedan inspeccionarse y corregirse.

Lenguajes data-flow

Lenguajes mixtos

01



Bloques · composición

Representación visual

Estructura y conexiones

02



Código · instrucciones

Representación textual

Detalle y precisión

Diseñar para quien conoce el problema

¿Qué debe poder cambiar el usuario?

Identificad una necesidad propia y elegid una herramienta.