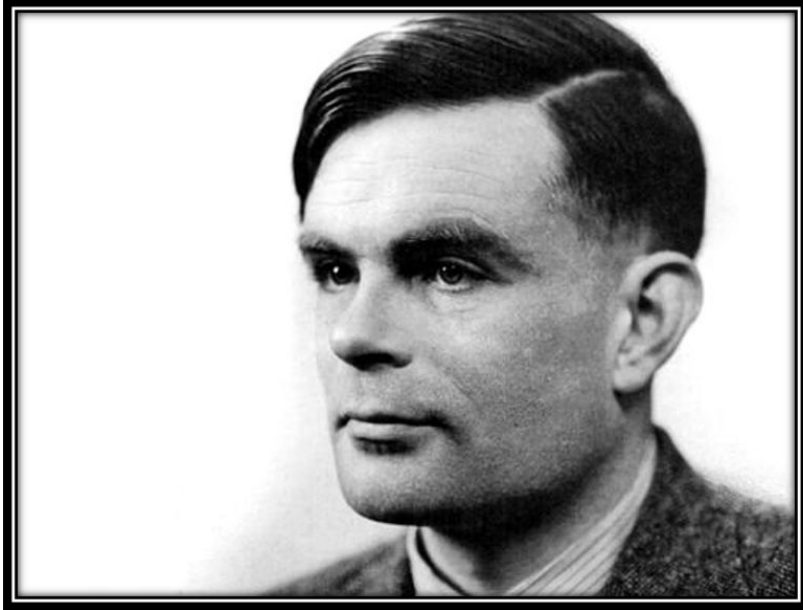


¿Qué es Software?



Software



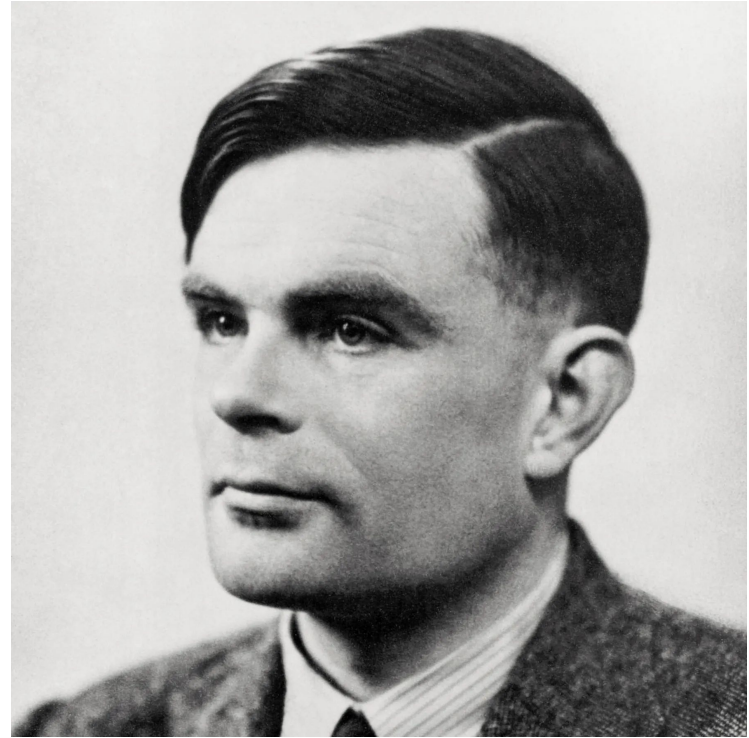
Alan Turing



Alonso Church



Conjunto de
Instrucciones que dado
cierto input genera cierto
output



Si Software es un “Conjunto de Instrucciones” ...

¿Qué es “buen software”?

¿Cómo organizamos el “software”?



(**Conocimiento** expresado en un)

Modelo Computable

de un ***Dominio de Problema***

de la Realidad



Realidad

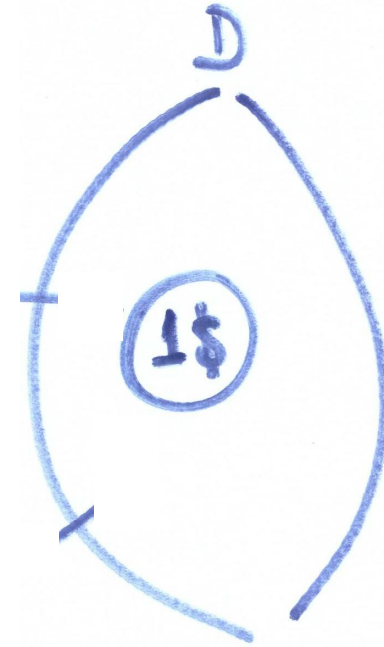


R



Dominio de Problema





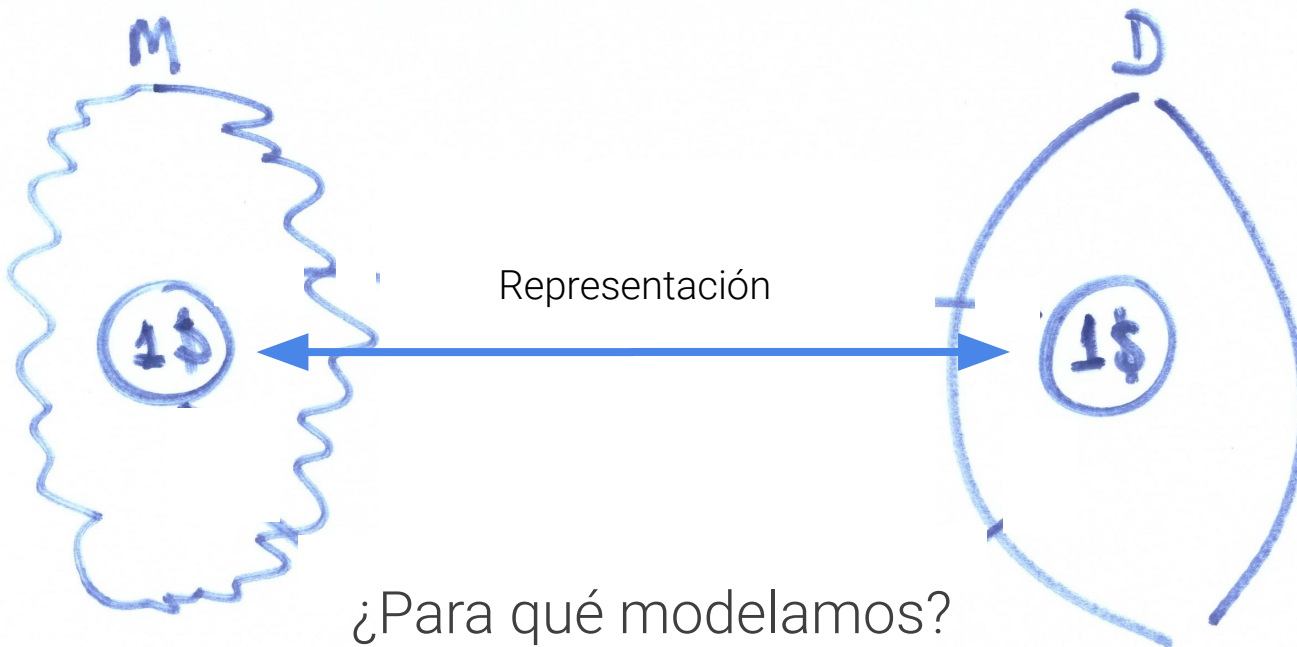
Arbitrario
-
Lenguaje
Natural
-
Ambiguo
Contextual

R



Modelo





Arbitrario
-
Lenguaje Natural
-
Ambiguo Contextual

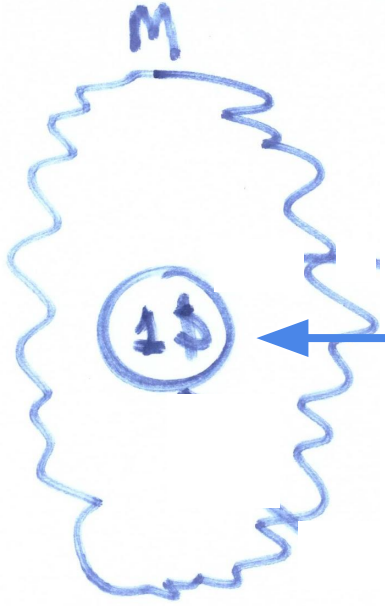
¿Para qué modelamos?
¿En qué lenguaje modelamos?
¿Qué tipo de modelos hacemos?

R



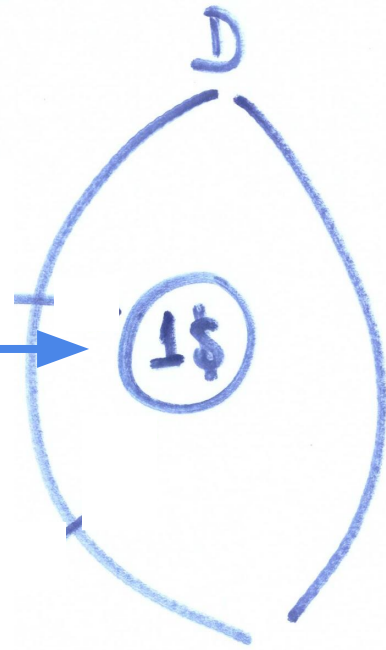
Modelo Computable





Lenguaje
Formal
-
Ejecutable
-
Código
Fuente

Representación



Arbitrario
-
Lenguaje
Natural
-
Ambiguo
Contextual

R



- **Realidad:** Un conjunto de cosas
- **Cosa:** Todo aquello acerca de lo cual podemos decir algo
- **Dominio de Conocimiento:** Un recorte de la realidad que representa el negocio que estamos modelando
 - Dominios de negocio vs. dominios naturales



- **Modelo:** Representación del conocimiento construido acerca de un dominio
- **Computable:** Que puede ejecutar en una máquina de Turing Formal, a-contextual
 - Característica esencial: No solo especifica el qué sino que además implementa el cómo



Si Software es un “Modelo” ...

¿Qué es “buen software”?

¿Cómo organizamos el “Software”?
(rails way vs. modelo way)



“Buen Software” == “Buen Modelo”

Lo debemos organizar según la
organización del Dominio de Problema

