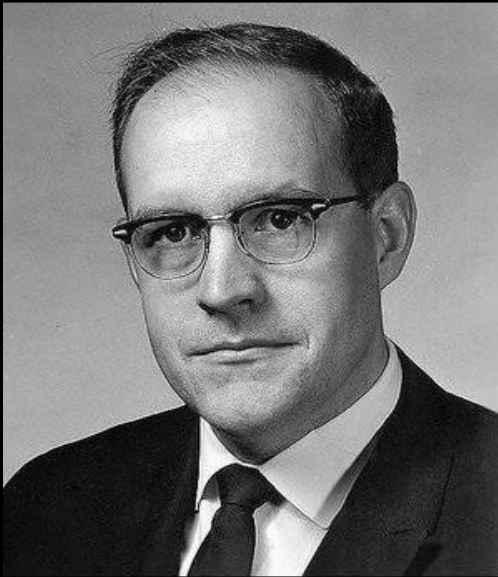
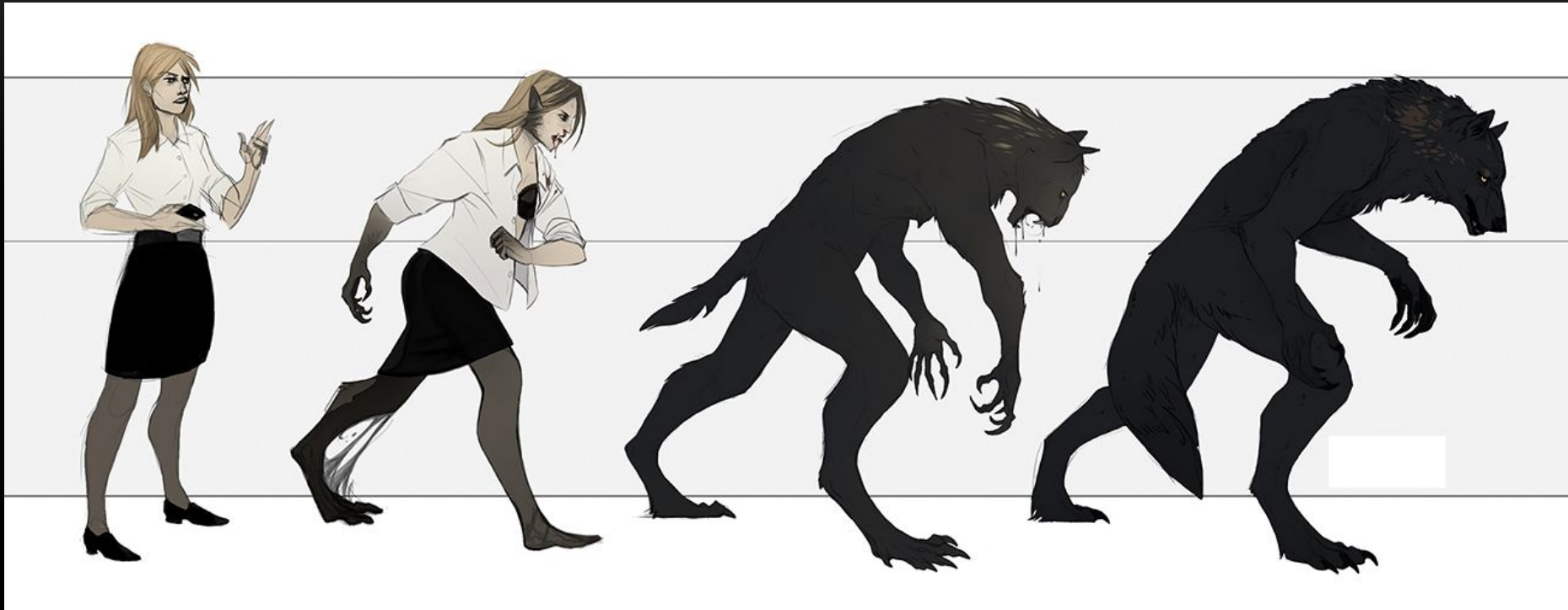


NO HAY BALAS DE PLATA



No hay desarrollo único, ya sea en la tecnología o la gestión técnica, que por sí mismo incluso promete una mejora de un orden de magnitud dentro de una década de la productividad, en la fiabilidad, en la simplicidad.

—Frederick P. Brooks Jr.

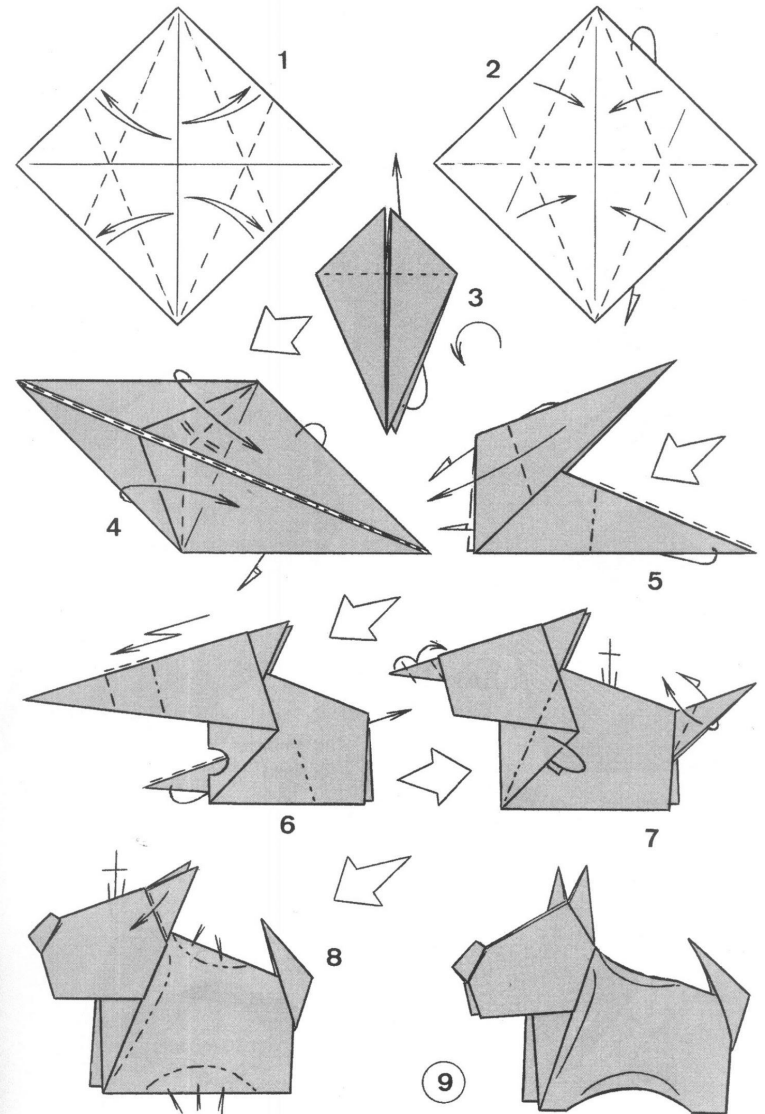


De lo Familiar al HORROR

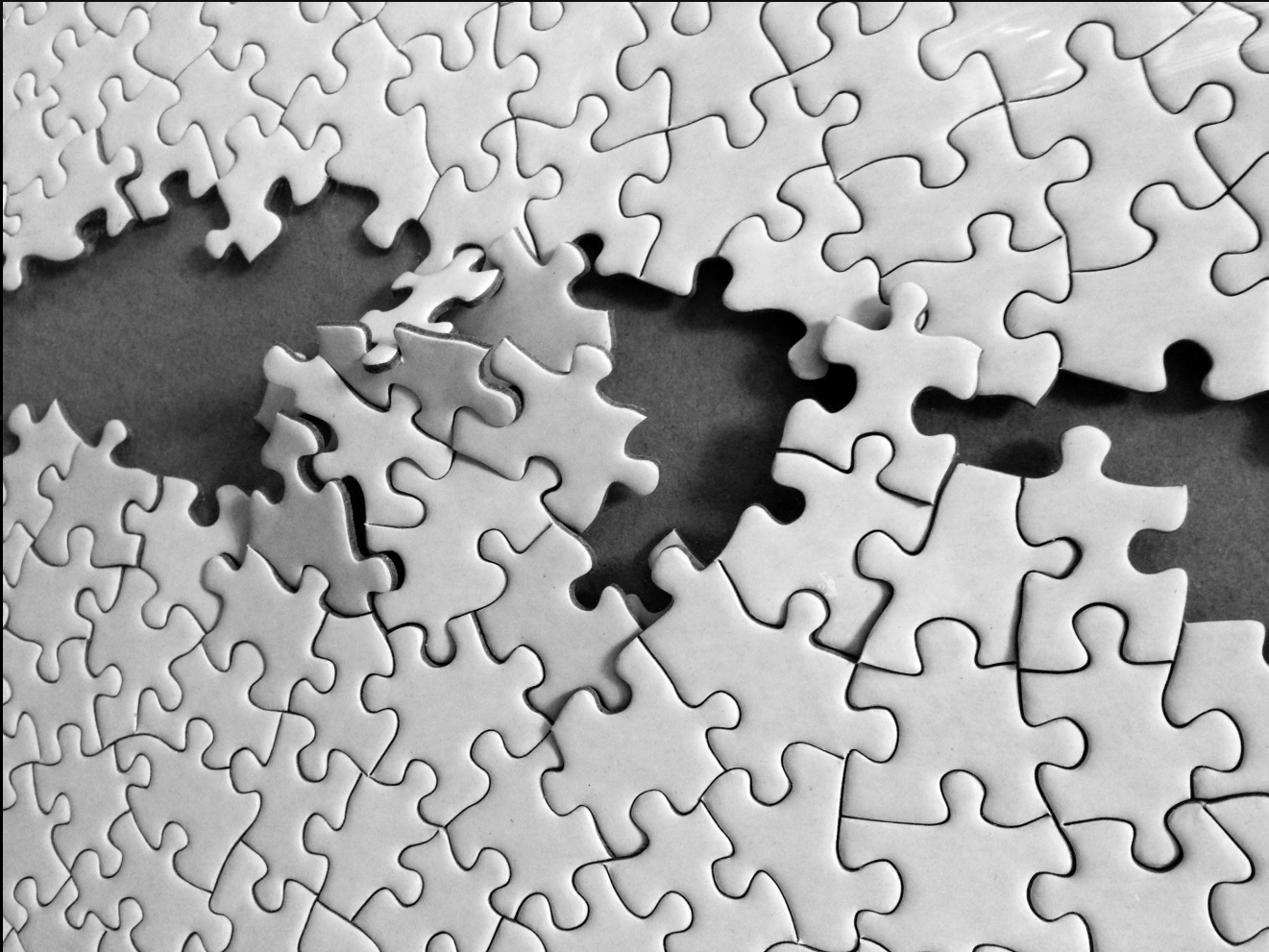


LITTLE TERRIER DOG

by Francisco Javier Caboblanco



Complejidad



Adaptabilidad



Modificabilidad



COMPUTADORES

- o Antes
- o TRS-80 Model III
- o Commodore SX-64
- o Apple IIGS
- o computadoras
- o Actualmente
- o Portátiles
- o Pantalla plana
- o Táctiles
- o Mesa



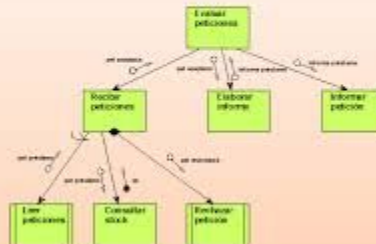
Invisibilidad



=



Diseño Estructurado: DIAGRAMA DE ESTRUCTURA. Ejemplo de diagrama de estructuras



AVANCES HISTÓRICOS FRENTE A DIFICULTADES ACCIDENTALES

- LENGUAJES DE ALTO NIVEL

Lenguaje de Alto Nivel C	Lenguaje Ensamblador
<pre>SWITCH TYPE) { case 'a': type=type+10; break; case 'b': type= type+20; break; default: break;}</pre>	<pre>MOV1 R1 = Type LD4 R2 = [R1] ;; cmp.eq P1, P2 = 'a', R2 cmp.eq P3, P4 = 'b', R2 ;; (P1) Add R2 = 10, R2 (P3) Add R2 = 20, R2 ;; st4 (R1) = R2 default::</pre>

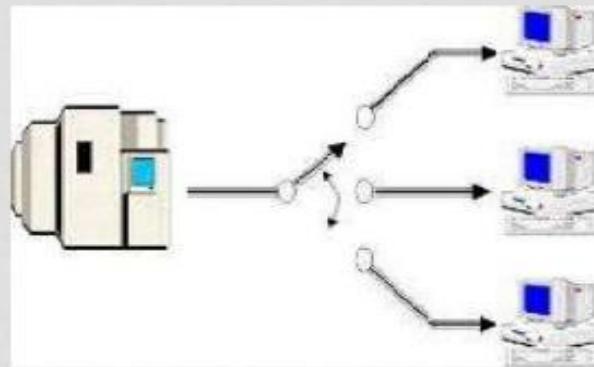
- SISTEMAS DE TIEMPO COMPARTIDO

SISTEMAS OPERATIVOS DE TIEMPO COMPARTIDO

- **Sistemas Operativos de tiempo compartido**

Permiten la simulación de que el sistema y sus recursos son todos para cada usuario.

**REDUCE
CONSIDERABLEMENTE LOS
TIEMPOS DE RESPUESTA
PARA EL USUARIO**



- SISTEMAS DE PROGRAMACIÓN UNIFICADOS



ESPERANZAS POR LAS BALAS

8888	8888	8888
3b83	5788	8888
d61f	7abd	8818
8b06	e8f7	88aa
8a18	880c	e841
a948	5862	5884

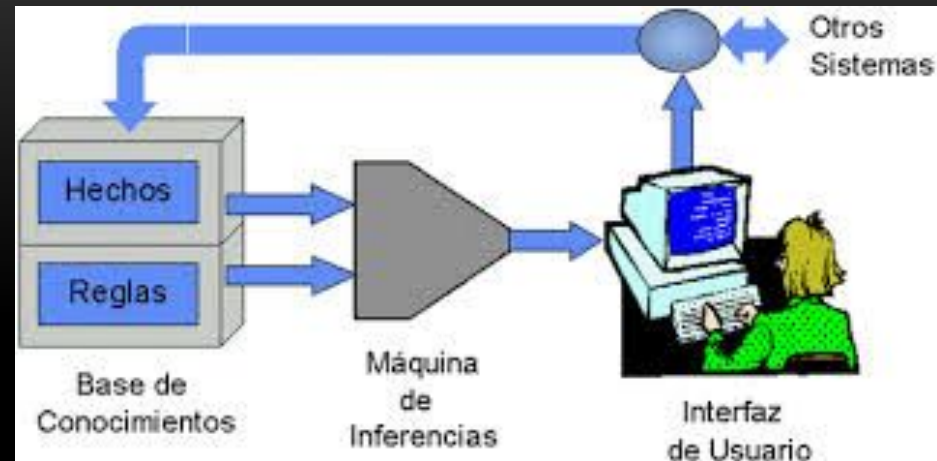


PROGRAMACIÓN ORIENTADA A OBJETOS

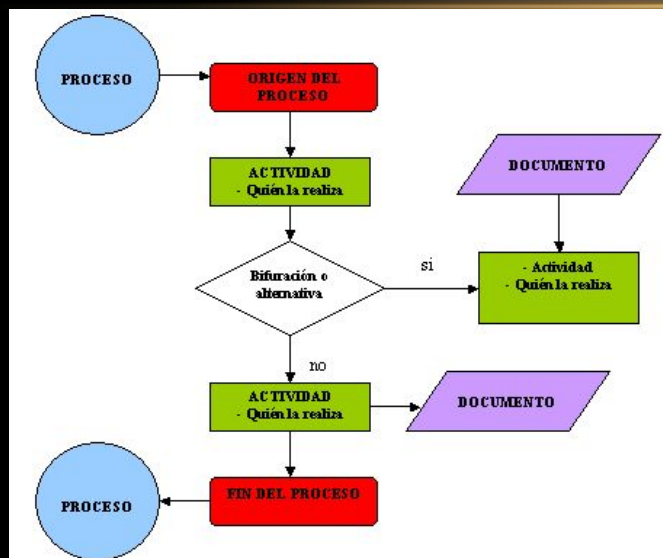


```
<Un código HTML >  
  <limpio es>  
    <fundamental>  
  </para>  
</una campaña>  
</exitosa>
```





IA



Dividendo			
575	25	Divisor	
- 250	10		
- 325	+ 10		
- 250			
- 75	2		
- 50			
- 25	1		
- 25			
Resto 0	23		

Verificación de Programas



¿La especificación está completa es consistente?

Mejoras en Ambientes y herramientas



Utilidades marginales.

Estaciones de trabajo



No generan soluciones mágicas.

Ataques prometedores a la esencia conceptual



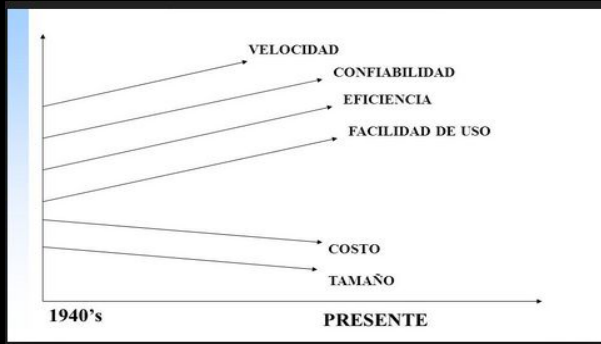
Atacar

$$\text{Tiempo_de_ejecución} = \sum (\text{frecuencia})_i \times (\text{tiempo})_i$$

COMPRAR VS CONSTRUIR



Hardware Evolución.



Retos inherentes al desarrollo



Software del Mercado

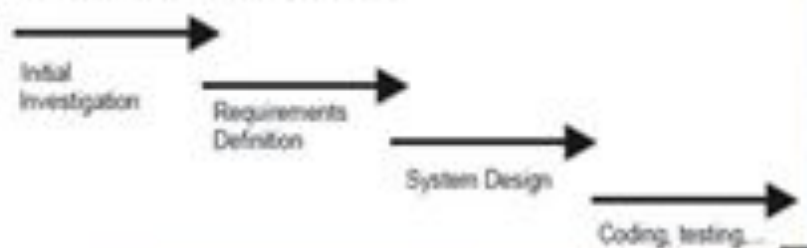


Adaptar soluciones

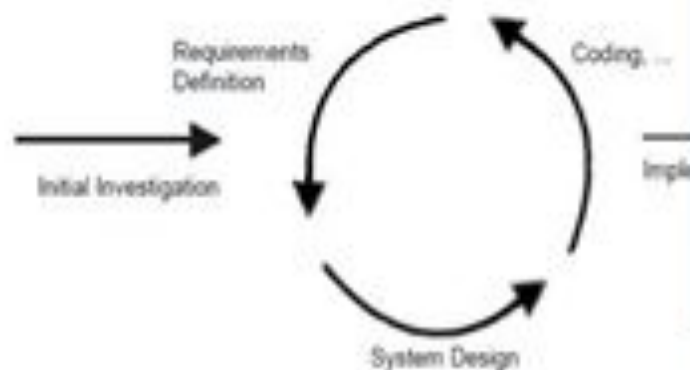


Refinamiento de requerimientos y prototipado veloz

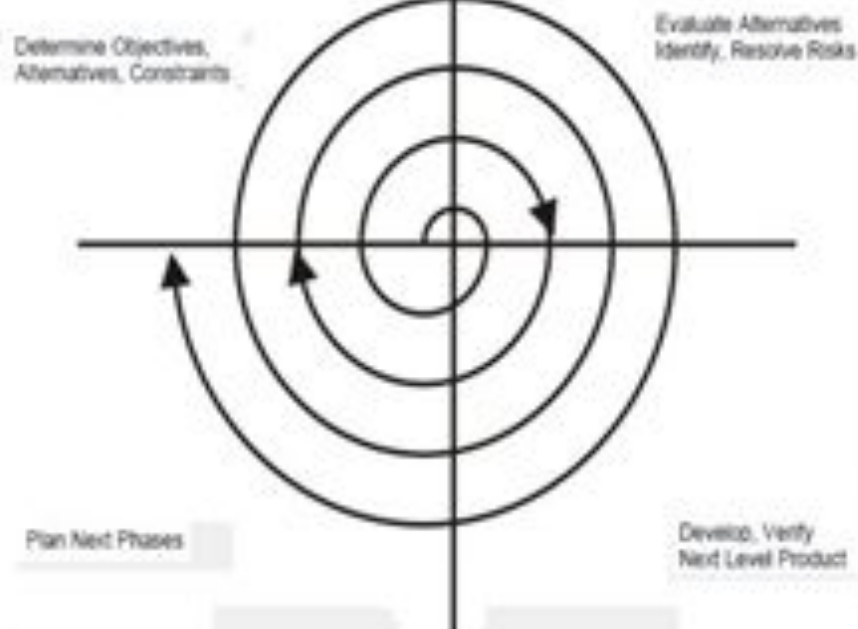
Waterfall



Prototyping



Spiral



Cultivar el software (prototipos + incremental)



Diseñadores brillantes



