

# 1. INTRODUCCIÓN

- 1.1 Qué es el RBC
- 1.2 Del razonamiento analógico al razonamiento basado en casos
- 1.3 Etapas
- 1.4 Ejemplos
- 1.5 Aplicaciones
- 1.6 El modelo cognitivo

---

---

---

---

---

---

---

---

## 1.1 Qué es el RBC?

- RBC es razonar por recuerdo
- RBC resuelve problemas nuevos adaptando soluciones que se han usado en la solución de problemas viejos
- RBC es una nueva aproximación a la resolución de problemas y el aprendizaje
- RBC es una forma con que las personas resolvemos problemas usando casos a la vez que una manera para construir máquinas que los usen

---

---

---

---

---

---

---

---

## 1.2 Analogía

### Intuición

“Si dos situaciones son similares en algún aspecto, entonces pueden serlo en otro”.

+ Si conocemos el funcionamiento de un circuito RLC, es fácil comprender un sistema de fuerzas.

- Los antiguos querían construir máquinas para volar observando como movían las alas los pájaros

---

---

---

---

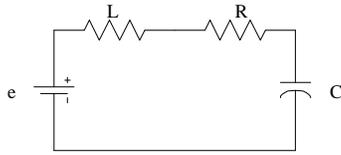
---

---

---

---

## Analogía +. Circuito RLC



$$e = L \frac{\partial^2 q}{\partial t^2} + R \frac{\partial q}{\partial t} + \frac{1}{C} q$$

Udg/Dpt. EIA - Disseny de sistemes de supervisió

7

---

---

---

---

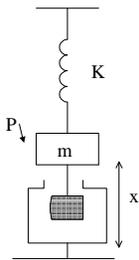
---

---

---

---

## Analogía +. Sistema de fuerzas



Fuerza  $p = e$  Tensión  
 Masa  $m = L$  Inductancia  
 Coef. Elasticidad  $f = R$  Resistencia  
 Desplazamiento  $x = q$  Carga

$$p = m \frac{\partial^2 x}{\partial t^2} + f \frac{\partial x}{\partial t} + Kx$$

Udg/Dpt. EIA - Disseny de sistemes de supervisió

8

---

---

---

---

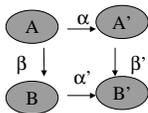
---

---

---

---

## Definiciones



$\beta$ : Causalidad entre A y B  
 $\alpha$ : Similitud entre A y A'  
 $\beta'$ : Razonamiento analógico

Udg/Dpt. EIA - Disseny de sistemes de supervisió

9

---

---

---

---

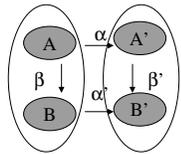
---

---

---

---

## Definiciones



dominio base      dominio objetivo  
*el sistema solar*      *el átomo de hidrógeno*

10

---

---

---

---

---

---

---

---

## Definiciones

$$\frac{\alpha(S) \wedge \beta(S)}{\alpha(T) \quad \beta(T)}$$

Analogía: proceso de inferir una conclusión  $\beta$  sobre una situación/objeto T por el hecho que T comparte una propiedad (o conjunto de propiedades)  $\alpha$  con otra situación/objeto S que tiene la propiedad  $\beta$

Udg/Dpt. EIA - Disseny de sistemes de supervisió

11

---

---

---

---

---

---

---

---

## Ejemplo: Analogía entre el agua de un río y la corriente eléctrica

$\beta$ : el agua se mueve desde los lugares altos hacia los bajos

$\alpha$ : el flujo del agua es similar a la corriente eléctrica

$\beta'$ : la corriente eléctrica fluye desde el potencial alto hacia el bajo

Udg/Dpt. EIA - Disseny de sistemes de supervisió

12

---

---

---

---

---

---

---

---

## Procedimiento analógico (modelo unificado)

1. Recuperación
2. Elaboración
3. Correspondencia
4. Justificación
5. Consolidación o aprendizaje

Udg/Dpt. EIA - Disseny de sistemes de supervisió

13

---

---

---

---

---

---

---

---

## Modelo de Greiner

Inferencia analógica útil ( $\mid \sim$ ): búsqueda de las suposiciones/conjeturas sobre A que aporten algo a la solución de un problema P

$$T, A \sim B \mid \sim \varphi(A)$$

T: teoría

A~B: posible analogía ( $\alpha$ )

$\varphi(A)$ : resultado de la analogía ( $\beta$ )

Udg/Dpt. EIA - Disseny de sistemes de supervisió

14

---

---

---

---

---

---

---

---

## Propiedades del modelo de Greiner

- Novedad:  $T \mid \neq \varphi(A)$
- Consistencia:  $T \mid \neq \neg\varphi(A)$
- Fundamentación:  $T \mid = \varphi(B)$
- Utilidad:  $T \cup \varphi(A) \mid = P$

Udg/Dpt. EIA - Disseny de sistemes de supervisió

15

---

---

---

---

---

---

---

---

### Ejemplo (I)

Flujo A=3

Flujo B = 9

Flujo C = ?

Udg/Dpt. EIA - Disseny de sistemes de supervisió 16

---

---

---

---

---

---

---

---

### Ejemplo (II)

- Teoría disponible:  $T_{ce}$  (teoría sobre la corriente eléctrica)
- Novedad:  $T_{ce} \neq \text{Flujo C}$
- Suposición analógica:  
corriente eléctrica ~flujo del agua

Udg/Dpt. EIA - Disseny de sistemes de supervisió 17

---

---

---

---

---

---

---

---

### Ejemplo (III)

Conocimiento sobre el dominio de la corriente eléctrica: 1ª Ley de Kirchoff (K1)

I1

I2

$I3 = I1 + I2$

Udg/Dpt. EIA - Disseny de sistemes de supervisió 18

---

---

---

---

---

---

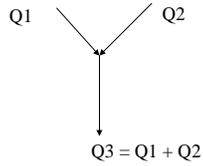
---

---

## Ejemplo (IV)

Aplicamos la K1 sobre el flujo

$T_{ce}$ , Corriente~Flujo | ~ K1(Flujo)



Udg/Dpt. EIA - Disseny de sistemes de supervisió

19

---

---

---

---

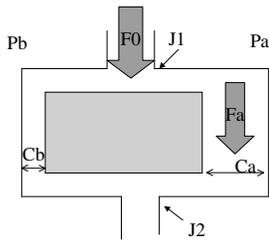
---

---

---

---

## Complejidad de la inferencia analógica útil



Udg/Dpt. EIA - Disseny de sistemes de supervisió

20

---

---

---

---

---

---

---

---

## Analogías necesarias

K1: primera ley de Kirchoff

K2: segunda ley de Kirchoff

Ohm: Ley de Ohm

Cons: Ley de conservación

Udg/Dpt. EIA - Disseny de sistemes de supervisió

21

---

---

---

---

---

---

---

---

## Fórmula analógica

Basada en la combinación de todas las analogías necesarias

$\varphi$ (corriente, potencial, resistencia, resistor)

$\varphi$ (flujo, presión, tubo, características)

---

---

---

---

---

---

---

---

## Dificultades

- Encontrar la correspondencia adecuada entre variables
- Conocer que propiedades de la corriente eléctrica se mantienen en hidráulica

---

---

---

---

---

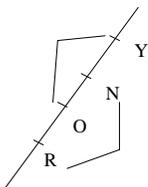
---

---

---

## Analogía transformacional

Equivalencia de segmentos



Si RO y NY son iguales,  
demostrar que RN y OY  
son iguales

$$RO = NY$$

$$ON = ON$$

$$RO + ON = NY + ON$$

$$RO + ON = ON + NY$$

$$RN = OY$$

---

---

---

---

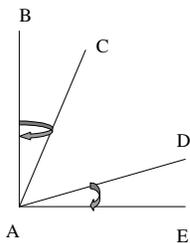
---

---

---

---

## Equivalencia de ángulos



Si BAC y DAE son iguales,  
demostrar que BAD y CAE  
lo son.

$$\begin{aligned} BAC &= DAE \\ CAD &= CAD \\ BAC + CAD &= DAE + CAD \\ BAC + CAD &= CAD + DAE \\ BAD &= CAE \end{aligned}$$

Udg/Dpt. EIA - Disseny de sistemes de supervisió

25

---

---

---

---

---

---

---

---

## Analogía transformacional

La aplicación estricta del mismo conjunto de operadores, en el mismo orden, es una solución analógica.

⇒ El modelo de analogía transformacional resuelve la analogía mediante un procedimiento de búsqueda de operadores.

Udg/Dpt. EIA - Disseny de sistemes de supervisió

26

---

---

---

---

---

---

---

---

## Análisis de medios a fines (*means-ends analysis*).

- Similitud entre problemas: estado inicial, final, restricciones, aplicabilidad de los operadores.
- Estrategias: búsqueda de precedentes, transformación de la solución, métodos de modificación de soluciones (permutaciones, inserciones, etc.)

Udg/Dpt. EIA - Disseny de sistemes de supervisió

27

---

---

---

---

---

---

---

---

## ARIES

- Implementación del método de analogía transformacional
- Métrica de similitud
- Operadores: inserción de un operador en la secuencia, eliminación, expansión de la solución, sustitución de operadores equivalentes, etc.
- Organización de la memoria
- La búsqueda de t-operadores puede resultar muy compleja.

Udg/Dpt. EIA - Disseny de sistemes de supervisió

28

---

---

---

---

---

---

---

---

## Analogía y Razonamiento basado en casos

Razonamiento analógico:  
inter-dominio



Razonamiento basado en casos:  
intra-dominio

Udg/Dpt. EIA - Disseny de sistemes de supervisió

29

---

---

---

---

---

---

---

---

## Razonamiento basado en casos

- Experto vs. principiante
- Aprendizaje vs. razonamiento
- Fundamentos analógicos
  - intra-dominio
  - aprendizaje

Udg/Dpt. EIA - Disseny de sistemes de supervisió

30

---

---

---

---

---

---

---

---

## RBC

- Ventajas:
  - técnica de adquisición de conocimientos
  - técnica de resolución de problemas
  - dominios de difícil estructuración
- Inconvenientes
  - memoria
  - manipulación de la incertidumbre

---

---

---

---

---

---

---

---

## 1.3 Etapas

- Recuperación
- Correspondencia (*Matching*)
- Adaptación
- Evaluación
- Aprendizaje

---

---

---

---

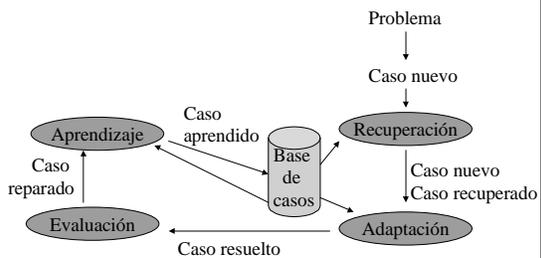
---

---

---

---

## Etapas



---

---

---

---

---

---

---

---

## 1.4. Ejemplo. RBC para la generación de programas

Objetivo: generar nuevos programas a través de la adaptación de viejos

Input: comportamiento deseado del programa

Output: código

Librería de casos: trozos de código con anotaciones

---

---

---

---

---

---

---

---

## Caso en la librería

Código =

```
(DEFUN F1 (N)
```

```
  (DO ((I 1 (+ I 1))
```

```
    (S 0 (+ I S)))
```

```
    ((> I N) S)))
```

Especificación =

“Sumar desde 1 hasta N”

Acumulador = +

Fuente =

Rango .tipo = entero

.inicio = 0

.final = N

.incr = +

.paso = 1

---

---

---

---

---

---

---

---

## Nuevo problema

Comportamiento deseado:

multiplicar desde 1 hasta N

Especificación = “Multiplica desde 1 hasta N”

Acumulador = \*

Fuente = Rango .tipo = entero

.inicio = 1

.final = N

.incr = +

.paso = 1

---

---

---

---

---

---

---

---

## Recuperación y adaptación

- Recuperado: F1
- Diferencias:
  - Acumulador:
    - caso nuevo: \*
    - caso recuperado: +
- Resolución de diferencias: substituir + por \*

---

---

---

---

---

---

---

---

## Resultado

```
(DEFUN F2 (N)
  (DO ((I 1 (+ I 1))
      (S 0 (* I S)))
    (> I N) S)))
```

---

---

---

---

---

---

---

---

## Evaluación

<i>Expresión</i>	<i>Res. Esperado</i>	<i>Res. Actual</i>
(F2 5)	120	0

Explicación del fallo: (DEFUN F2 (N)

```
(DO ((I 1 (+ I 1))
    (S 0 (* I S)))
  (> I N) S)))
```

(S 0 (\* I S)) genera  $0 * 1 * 2 * 3 * 4 * 5 = 0$

---

---

---

---

---

---

---

---

## Reparación

- Substituir 0 en (S 0 (\* I S)) por 1.
- Nuevo resultado

```
(DEFUN F3 (N)
  (DO ((I 1 (+ I 1))
      (S 1 (* I S)))
    (> I N ) S)))
```

---

---

---

---

---

---

---

---

## Aprendizaje: índices y almacenamiento

Código =	Especificación =
(DEFUN F3 (N)	“Multiplica desde 1 hasta N”
(DO ((I 1 (+ I 1))	Acumulador = *
(S 1 (+ I S)))	Fuente =
(> I N ) S)))	Rango .tipo = entero
	.inicio = 1
	.final = N
	.incr = +
	.PASO = 1

---

---

---

---

---

---

---

---

## Conclusiones +

- Se ha generado código real
- Hay muy poco conocimiento sobre programación
- Las mismas reglas de adaptación pueden funcionar para otros trozos de código LISP
- Reglas similares pueden funcionar con otros lenguajes (PASCAL)

---

---

---

---

---

---

---

---

## Advertencias

- Es un tipo de programación pensado para “patches” en un lenguaje no habitual del programador
- No hay garantía de que la solución generada funcione
- Se necesitaría mucho más conocimiento para adaptar diferentes códigos a objetivos múltiples
- Existe una dependencia crítica en la forma de indizar el código (especificaciones correctas).

---

---

---

---

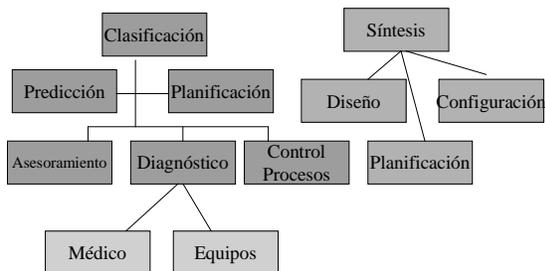
---

---

---

---

## 1.5 Aplicaciones



---

---

---

---

---

---

---

---

## 1.6 El modelo cognitivo

- RBC: una metodología para modelar el proceso del pensamiento y razonamiento humano
  - RBC: una metodología para construir sistemas inteligentes
- ⇒ Desde el punto de vista de la ingeniería interesa la construcción de sistemas inteligentes

---

---

---

---

---

---

---

---