

# TADs en 2 niveles

## Clase práctica

Algoritmos y Estructuras de Datos II

# Ejercicio 1

## Reproductor de música

Se nos pide modelar mediante TADs un servicio de música en internet (iSpotify). El servicio provee un catálogo de canciones. Cada canción pertenece a un género musical. Los usuarios pueden reproducir canciones del catálogo.

Se nos pide saber cuantas veces se reprodujo cada canción y cuantas canciones se reprodujeron de cada género.

# Resolución

## TAD ifiSpot

**géneros**      ifiSpot

### observadores básicos

catalogo : ifiSpot  $\longrightarrow$   $\text{dicc}(\text{cancion}, \text{genero})$

reproducciones :  $\text{cancion } c \times \text{ifiSpot } s \longrightarrow \text{nat}$

$\{c \in \text{claves}(\text{catalogo}(s))\}$

### generadores

nuevolfiSpot :  $\text{dicc}(\text{cancion} \times \text{genero}) \longrightarrow \text{ifiSpot}$

reproducir :  $\text{cancion } c \times \text{ifiSpot } s \longrightarrow \text{ifiSpot}$

$\{c \in \text{claves}(\text{catalogo}(s))\}$

### otras operaciones

reproduccionesGenero :  $\text{genero} \times \text{ifiSpot} \longrightarrow \text{nat}$

**Fin TAD**

## axiomas

$\text{catalogo}(\text{nuevolfiSpot}(\text{cat})) \equiv \text{cat}$

$\text{catalogo}(\text{reproducir}(c,s)) \equiv \text{catalogo}(s)$

$\text{reproducciones}(c, \text{nuevolfiSpot}(\text{cat})) \equiv 0$

$\text{reproducciones}(c, \text{reproducir}(c', s)) \equiv \text{if } c = c' \text{ then } 1 \text{ else } 0 \text{ fi} + \text{reproducciones}(c, s)$

$\text{reproduccionesGenero}(g, \text{nuevolfiSpot}(\text{cat})) \equiv 0$

$\text{reproduccionesGenero}(g, \text{reproducir}(i, c)) \equiv \text{reproduccionesGenero}(g, i) + (\text{if obtener}(c, \text{catalogo}(i)) = g \text{ then } 1 \text{ else } 0 \text{ fi})$

## axiomas

$\text{catalogo}(\text{nuevolfiSpot}(\text{cat})) \equiv \text{cat}$

$\text{catalogo}(\text{reproducir}(c,s)) \equiv \text{catalogo}(s)$

$\text{reproducciones}(c, \text{nuevolfiSpot}(\text{cat})) \equiv 0$

$\text{reproducciones}(c, \text{reproducir}(c', s)) \equiv \text{if } c = c' \text{ then } 1 \text{ else } 0 \text{ fi} + \text{reproducciones}(c, s)$

$\text{reproduccionesGenero}(g, \text{nuevolfiSpot}(\text{cat})) \equiv 0$

$\text{reproduccionesGenero}(g, \text{reproducir}(i, c)) \equiv \text{reproduccionesGenero}(g, i) + (\text{if obtener}(c, \text{catalogo}(i)) = g \text{ then } 1 \text{ else } 0 \text{ fi})$

Siempre **cuidado** al axiomatizar sobre generadores.

## axiomas

$\text{catalogo}(\text{nuevolfiSpot}(\text{cat})) \equiv \text{cat}$

$\text{catalogo}(\text{reproducir}(c,s)) \equiv \text{catalogo}(s)$

$\text{reproducciones}(c, \text{nuevolfiSpot}(\text{cat})) \equiv 0$

$\text{reproducciones}(c, \text{reproducir}(c', s)) \equiv \text{if } c = c' \text{ then } 1 \text{ else } 0 \text{ fi} + \text{reproducciones}(c, s)$

$\text{reproduccionesGenero}(g, \text{nuevolfiSpot}(\text{cat})) \equiv 0$

$\text{reproduccionesGenero}(g, \text{reproducir}(i, c)) \equiv \text{reproduccionesGenero}(g, i) + (\text{if obtener}(c, \text{catalogo}(i)) = g \text{ then } 1 \text{ else } 0 \text{ fi})$

Siempre **cuidado** al axiomatizar sobre generadores.

¿Cual es el TAD de segundo nivel en este caso?

## Ejercicio 2

En la facultad de exactas se decidieron organizar competencias de acertijos matemáticos. Los acertijos se categorizan con un número de complejidad del 1 al 5 y se organizan formando un laberinto.

El objetivo de la competencia es ser el primero en atravesar el laberinto (ver ejemplo). Todos los jugadores comienzan en un acertijo inicial. Al resolverlo tienen la posibilidad de avanzar en el laberinto. Cuando un jugador resuelve un acertijo obtiene acceso a uno o más acertijos nuevos. No obstante, el jugador debe elegir con cuál de estos nuevos acertijos quiere enfrentarse. Esta decisión lo restringe a un camino en el laberinto. Todos los caminos posibles terminan en un acertijo final. No existen ciclos en el laberinto.

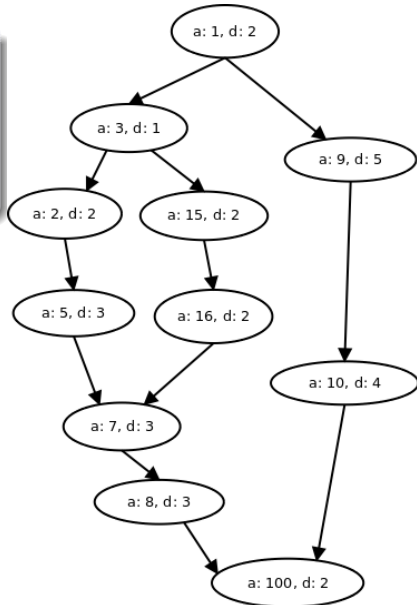
Los jugadores tienen completo conocimiento de las relaciones entre los acertijos y sus dificultades. La temporada solo termina al ser resuelto el acertijo final y el ganador es quién lo haga.

Se pide especificar el TAD Temporada para que maneje la información administrativa de cada competencia. Se deben conocer los acertijos y la relación ellos, así como sus dificultades. Además debe mantener la información sobre los jugadores, en qué acertijo se encuentran y cuáles resolvieron, e imponer las restricciones del laberinto a la hora de moverse por él. La temporada debe finalizar cuando un jugador resuelve el acertijo final, luego de esto el jugador pasa a ser el ganador de la temporada y ningún otro jugador puede resolver acertijos.

# Ejemplo de laberinto de acertijos

El laberinto de acertijos requerido para una temporada tiene ciertas restricciones:

- Hay un único acertijo final y un único acertijo inicial
- Siguiendo un camino del laberinto no puedo llegar a un acertijo ya resuelto





# Resolución

**TAD** acert es nat

**TAD** Laberinto

**géneros**            lab

**observadores básicos**

acertijos : lab  $\longrightarrow$  conj(acert)

dificultad : lab  $\mid \times$  acert a  $\longrightarrow$  nat  $\{a \in \text{acertijos}(l)\}$

opciones : lab  $\mid \times$  acert a  $\longrightarrow$  conj(acert)  $\{a \in \text{acertijos}(l)\}$

**generadores**

nuevoLab : acert a  $\times$  nat d  $\longrightarrow$  lab  $\{1 \leq d \leq 5\}$

agAcertijo : lab  $\mid \times$  acert acj  $\times$  nat dif  $\times$  conj(acert) prev  $\longrightarrow$  lab

$$\left\{ \begin{array}{l} \text{prev} \subseteq \text{acertijos}(l) \wedge 1 \leq \text{dif} \leq 5 \wedge \neg \text{acj} \in \text{acertijos}(l) \wedge \#(\text{prev}) \\ \geq 1 \end{array} \right\}$$

**otras operaciones**

acertijoInicial : lab  $\longrightarrow$  acert

acertijosFinales : lab  $\longrightarrow$  conj(acert)

filtrarAcertijosFinales : lab  $\mid \times$  conj(acert) as  $\longrightarrow$  conj(acert)

$\{as \subseteq \text{acertijos}(l)\}$

## axiomas

$\text{acertijos}(\text{nuevoLab}(a, d)) \equiv \text{Ag}(a, \emptyset)$

$\text{acertijos}(\text{agAcertijo}(l, \text{acj}, \text{dif}, \text{prev})) \equiv \text{Ag}(\text{acj}, \text{acertijos}(l))$

$\text{dificultad}(\text{nuevoLab}(a, d), a') \equiv d$

$\text{dificultad}(\text{agAcertijo}(l, \text{acj}, \text{dif}, \text{prev}), a') \equiv \text{if } \text{acj} = a' \text{ then } \text{dif} \text{ else } \text{dificultad}(l, a') \text{ fi}$

$\text{opciones}(\text{nuevoLab}(a, d), a') \equiv \emptyset$

$\text{opciones}(\text{agAcertijo}(l, \text{acj}, \text{dif}, \text{prev}), a') \equiv \text{if } a' \in \text{prev} \text{ then } \text{Ag}(\text{acj}, \emptyset) \text{ else } \emptyset \text{ fi}$   
 $\cup$   
 $\text{if } a' \in \text{acertijos}(l) \text{ then } \text{opciones}(l, a') \text{ else } \emptyset \text{ fi}$

$\text{acertijoInicial}(\text{nuevoLab}(a, d)) \equiv a$

$\text{acertijoInicial}(\text{agAcertijo}(l, \text{acj}, \text{dif}, \text{prev})) \equiv \text{acertijoInicial}(l)$

$\text{acertijosFinales}(l) \equiv \text{filtrarAcertijosFinales}(l, \text{acertijos}(l))$

## Axiomas de los acertijos finales del Laberinto

$\text{acertijosFinales}(l) \equiv \text{filtrarAcertijosFinales}(l, \text{acertijos}(l))$

$\text{filtrarAcertijosFinales}(l, as) \equiv$  **if**  $\emptyset?(as)$   
    **then**  $as$   
    **else** (  
        **if**  $\emptyset?(\text{opciones}(\text{dameUno}(as), l))$   
        **then**  $\text{Ag}(\text{dameUno}(as), \emptyset)$   
        **else**  $\emptyset$   
        **fi**  
         $\cup \text{filtrarAcertijosFinales}(l, \text{sinUno}(as))$   
    )  
    **fi**

**TAD** jugador es nat

**TAD** Temporada

**géneros**            temp

**observadores básicos**

jugadores : temp  $\longrightarrow$  conj(jugador)

acertados : temp t  $\times$  jugador j  $\longrightarrow$  conj(acert)  $\{j \in \text{jugadores}(t)\}$

actual : temp t  $\times$  jugador j  $\longrightarrow$  acert  $\{j \in \text{jugadores}(t)\}$

lab : temp  $\longrightarrow$  lab

**generadores**

initTemp : conj(jugador) js  $\times$  lab l  $\longrightarrow$  temp  $\{\#(\text{acertijosFinales}(l)) = 1\}$

anotarYproxAcertijo : temp t  $\times$  jugador j  $\times$  acert elec  $\longrightarrow$  temp

$$\left\{ \begin{array}{l} j \in \text{jugadores}(t) \wedge_L \neg \text{actual}(t, j) \in \text{acertijosFinales}(\text{laberinto}(t)) \wedge \\ \text{elec} \in \text{opciones}(\text{laberinto}(t), \text{actual}(t, j)) \wedge \neg \text{finalizada}(t) \end{array} \right\}$$

anotarAcertijoFinal : temp t  $\times$  jugador j  $\longrightarrow$  temp

$$\left\{ \begin{array}{l} j \in \text{jugadores}(t) \wedge_L \text{actual}(t, j) \in \text{acertijosFinales}(\text{laberinto}(t)) \wedge \\ \neg \text{finalizada}(l) \end{array} \right\}$$

**otras operaciones**

finalizada : temp  $\longrightarrow$  bool

**Fin TAD**

## axiomas

$$\text{jugadores}(\text{initTemp}(\text{js}, l)) \equiv \text{js}$$
$$\text{jugadores}(\text{anotarYproxAcertijo}(t, n, j)) \equiv \text{jugadores}(t)$$
$$\text{jugadores}(\text{anotarAcertijoFinal}(t, j)) \equiv \text{jugadores}(t)$$
$$\text{acertados}(\text{initTemp}(js, l), j') \equiv \emptyset$$

```

acertados(anotarYproxAcertijo(t, j, elec), j')  $\equiv$  if j' = j then
    Ag(actual(t, j), acertados(t, j'))
else
    acertados(t, j')
fi

```

```

acertados(anotarAcertijoFinal(t, j), j')  $\equiv$  if j' = j then
    Ag(actual(t, j), acertados(t, j'))
else
    acertados(t, j')
fi

```

$$\text{actual}(\text{initTemp}(\text{js}, l), j') \equiv \text{acertijolnicial}(l)$$

```

actual(anotarYproxAcertijo(t, j, elec), j')  ≡  if j = j'
                                                    then elec
                                                    else actual(t, j')
                                                    fi

```

$$\text{actual}(\text{anotarAcertijoFinal}(t, j), j') \equiv \text{actual}^{\mathbb{A}}(t, j')$$

### **axiomas**

$\text{lab}(\text{initTemp}(\text{js}, \text{l})) \equiv \text{l}$

$\text{lab}(\text{anotarYproxAcertijo}(\text{t}, \text{j}, \text{elec})) \equiv \text{lab}(\text{t})$

$\text{lab}(\text{anotarAcertijoFinal}(\text{t}, \text{j})) \equiv \text{lab}(\text{t})$

$\text{finalizada}(\text{initTemp}(\text{js}, \text{l})) \equiv \text{false}$

$\text{finalizada}(\text{anotarYproxAcertijo}(\text{t}, \text{j}, \text{elec})) \equiv \text{false}$

$\text{finalizada}(\text{anotarAcertijoFinal}(\text{t}, \text{j})) \equiv \text{true}$

# Cerrando

## TADs en 2 niveles

- Identificar la **parte estática** de la **parte dinámica** del problema
- Analizar si conviene partir en 2 niveles para sumar claridad a la especificación
- Las partes tienen que tener comportamiento marcadamente distinto y separable

# Cerrando

## TADs en 2 niveles

- Identificar la **parte estática** de la **parte dinámica** del problema
- Analizar si conviene partir en 2 niveles para sumar claridad a la especificación
- Las partes tienen que tener comportamiento marcadamente distinto y separable

## Algunas consejos y conclusiones finales...

- Cuándo conviene usar los TAD básicos y cuándo TADs más específicos.
- Analizar las “partes” o “etapas” de un problema y como impactan en un TAD.
- Verificar si hay partes inmutables o siempre están cambiando.
- Identificar en el problema “entidades” con comportamiento/responsabilidades/propiedades asociadas y como se relacionan con otras “entidades”.



Es todo por hoy...

Preguntas ???