

# Algoritmos y Estructuras de Datos II

Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

## Trabajo Práctico de Especificación

### Grupo 1

Integrante	LU	Correo electrónico
Bálsamo, Facundo	874/10	facundobalsamo@gmail.com
Lasso, Nicolás	892/10	lasso.nico@gmail.com
Rodríguez, Agustín	120/10	agustinrodriguez90@hotmail.com
Tripodi, Guido	843/10	guido.tripodi@hotmail.com

### Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

# 1. TAD LINKLINKIT

## TAD LINKLINKIT

**géneros** linkLinkIT

**exporta** linkLinkIT, generadores, observadores

**usa** ARBOLCATEGORIAS, LINK, CATEGORIA, FECHA

### igualdad observacional

$$(\forall it, it' : \text{linkLinkIT}) \left( it =_{\text{obs}} it' \iff \left( \begin{array}{l} \text{categorias}(it) =_{\text{obs}} \text{categorias}(it') \wedge \\ \text{accesosXDia}(it) =_{\text{obs}} \text{accesosXDia}(it') \wedge \\ \text{linksXCategoria}(it) =_{\text{obs}} \text{linksXCategoria}(it') \\ \text{fechaAct}(it) =_{\text{obs}} \text{fechaAct}(it') \end{array} \right) \right)$$

### observadores básicos

**categorias** : linkLinkIT  $\longrightarrow$  arbolDeCategorias

**accesosXDia** : linkLinkIT  $it \times \text{link } l \times \text{fecha } f \longrightarrow \text{nat}$   
 $\{l \in \text{LinkLinkIT}(it) \ \&\& \ \text{fechaAct}(it) - f \leq 2\}$

**linksXCategoria** : linkLinkIT  $it \times \text{categoria } c \longrightarrow \text{conj}(\text{link})$   $\{c \in \text{dameTodos}(\text{categorias}(it))\}$

**fechaAct** : linkLinkIT  $it \longrightarrow \text{fecha}$

### generadores

**iniciar** : arbolCategorias  $\longrightarrow$  linkLinkIT  $\{-\emptyset? \text{DameTodos}(a)\}$

**agregarLink** : linkLinkIT  $it \times \text{link } l \times \text{categoria } c \longrightarrow \text{linkLinkIT}$   
 $\{\neg(l \in \text{links}(it)) \ \&\& \ c \in \text{dameTodos}(\text{categorias}(it))\}$

**accederLink** : linkLinkIT  $it \times \text{link } l \times \text{fecha } f \longrightarrow \text{linkLinkIT}$   
 $\{l \in \text{links}(it) \ \&\& \ f \geq \text{ultimoAcceso}(it)\}$

### otras operaciones

**links** : linkLinkIT  $\longrightarrow \text{conj}(\text{link})$

**#LinksXCategoria** : linkLinkIT  $it \times \text{categoria } c \longrightarrow \text{nat}$   $\{c \in \text{dameTodos}(\text{categorias}(it))\}$

**accesosUlt3Dias** : linkLinkIT  $\times \text{link} \longrightarrow \text{nat}$

**todosLosLinks** :  $\text{conj}(\text{categoria}) \times \text{linkLinkIT} \longrightarrow \text{conj}(\text{link})$

**actualizandoAccesos** :  $\text{fecha} \times \text{linkLinkIT} \times \text{link} \longrightarrow \text{nat}$

**axiomas**  $\forall it, it' : \text{linklinkIT}$   
 $\forall a : \text{arbolDeCategorias}$   
 $\forall c : \text{categoria}$   
 $\forall l : \text{link}$   
 $\forall f : \text{fecha}$

$\text{categorias}(\text{iniciar}(a)) \equiv a$

$\text{categorias}(\text{agregarLink}(it, l, c)) \equiv \text{categorias}(it)$

$\text{categorias}(\text{accederLink}(it, l, f)) \equiv \text{categorias}(it)$

$\text{accesosXDia}(\text{iniciar}(a), l, f) \equiv \emptyset$

$\text{accesosXDia}(\text{agregarLink}(it, l, c), l', f) \equiv \text{accesos}(it)$

accesosXDia(accederLink(it, l, f),l',f')	≡	<b>if</b> l = l' <b>then</b> <b>if</b> f = f' <b>then</b> 1 + accesosXDia(it,l',f') <b>else</b> accesosXDia(it,l',f') <b>fi</b> <b>else</b> accesosXDia(it,l',f') <b>fi</b>
linksXCategoria(iniciar(a), c)	≡	∅
linksXCategoria(agregarLink(it, l, c),c')	≡	<b>if</b> c = c' <b>then</b> Ag(l, linksXCategoria(it,c)) <b>else</b> linksXCategoria(it,c') <b>fi</b>
linksXCategoria(accederLink(it, l, f),c')	≡	linksXCategoria(it,c')
fechaAct(iniciar(a))	≡	0
fechaAct(agregarLink(it,l,c))	≡	fechaAct(it)
fechaAct(accederLink(it,l,f))	≡	f
links(it)	≡	todosLosLinks((dameTodos(categorias(it))),it)
todosLosLinks(c,it)	≡	<b>if</b> ∅?(c) <b>then</b> ∅ <b>else</b> linksXCategoria(it,dameUno(c))      ∪      linksXCategoria(it,sinUno(c)) <b>fi</b>
#LinksXCategoria(it,c)	≡	#LinksXCategoria(it,c)
accesosUlt3Dias(it,l)	≡	actualizandoAccesos(fechaAct(it),it,l)
actualizandoAccesos(f,it,l)	≡	<b>if</b> f = 0 <b>then</b> accesosXDia(it,l,f)+actualizandoAccesos(f,it,l) <b>else</b> <b>if</b> fechaAct(it) - f ≤ 2 <b>then</b> accesosXDia(it,l,f) + actualizandoAccesos(f-1,it,l) <b>else</b> actualizandoAccesos(f,it,l) <b>fi</b> <b>fi</b>

**Fin TAD**

## 2. TAD ARBOLDECATEGORIAS

### TAD ARBOLDECATEGORIAS

**géneros**      arbolDeCategorias

**exporta**      arbolDeCategorias, generadores, observadores

**usa**            CATEGORIA

#### igualdad observacional

$$(\forall ac, ac' : \text{arbolDeCategorias}) \left( ac =_{\text{obs}} ac' \iff \left( \text{raiz}(ac) =_{\text{obs}} \text{raiz}(ac') \wedge \left( \text{hijos}(ac) =_{\text{obs}} \text{hijos}(ac') \right) \right) \right)$$

#### observadores básicos

raiz            : arbolDeCategorias                       $\longrightarrow$  categoria

hijos          : arbolDeCategorias                       $\longrightarrow$  conj(arbolDeCategorias)

#### generadores

AC            : conj(arbolDeCategorias)  $a \times$  categoria  $c \longrightarrow$  arbolDeCategorias       $\{-estaEnAlguno(c, a)\}$

#### otras operaciones

dameTodos    : arbolDeCategorias                       $\longrightarrow$  conj(categoria)

losFamiliares : categoria  $c \times$  arbolDeCategorias  $a \longrightarrow$  conj(categoria)       $\{c \in dameTodos(ac)\}$

estaEnAlguno : arbolDeCategorias  $\times$  categoria  $\longrightarrow$  bool

losParientes : categoria  $\times$  conj(arbolDeCategorias)  $\longrightarrow$  conj(categoria)

todosLosDeAbajo arbolDeCategorias  $\longrightarrow$  conj(categoria)

**axiomas**       $\forall a: \text{arbolDeCategorias}$   
                   $\forall c: \text{categoria}$

raiz(AC(a,c))                       $\equiv c$

hijos(AC(a,c))                      $\equiv a$

estaEnAlguno(AC(a,c),c')  $\equiv$  **if**  $c = c' \parallel c' \in a$  **then** true **else** false **fi**

losFamiliares(c,a)                 $\equiv$  **if**  $c = \text{raiz}(a)$  **then** dameTodos(a) **else** losParientes(c,hijos(a)) **fi**

losParientes(c,conj)               $\equiv$  **if**  $\emptyset?(\text{conj})$  **then**  
     $\emptyset$   
    **else**  
    **if** estaEnAlguno(dameUno(conj),c) **then**  
    losFamiliares(c,dameUno(conj))  
    **else**  
    losParientes(c,sinUno(conj))  
    **fi**  
    **fi**

dameTodos(a)                       $\equiv$  Ag(raiz(ac),todosLosDeAbajo(hijos(a)))

todosLosDeAbajo(a)                $\equiv$  **if**  $\emptyset?(a)$  **then**  
     $\emptyset$   
    **else**  
    dameTodos(dameUno(a))  $\cup$  todosLosDeAbajo(sinUno(a))  
    **fi**

estaEnAlguno(a,c)                 $\equiv$  **if**  $c \in \text{dameTodos}(a)$  **then** true **else** false **fi**

**Fin TAD**

### 3. Renombres

**TAD CATEGORIA**

es String

**Fin TAD**

**TAD LINK**

es String

**Fin TAD**

**TAD FECHA**

es Nat

**Fin TAD**