

# Algoritmos y Estructuras de Datos II

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

## Trabajo Práctico I

### Grupo: 12

Integrante	LU	Correo electrónico
Pondal, Iván	078/14	ivan.pondal@gmail.com
Paz, Maximiliano León	251/14	m4xileon@gmail.com
Mena, Manuel	313/14	manuelmena1993@gmail.com
Demartino, Francisco	348/14	demartino.francisco@gmail.com

### Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

## 1. TADs Auxiliares

**TAD** pc **ES** nat

**TAD** paquete **ES** tupla(nat id, nat ipOrigen, nat ipDestino, nat prioridad)

**TAD** segmento **ES** tupla(nat, nat, nat, nat)

## 2. TAD DCNET

**TAD** DCNET

**géneros** dcnet

**igualdad observacional**

$$(\forall d, d' : \text{dcnet}) \quad d =_{\text{obs}} d' \iff \left( \begin{array}{l} (\text{topo}(d) =_{\text{obs}} \text{topo}(d')) \wedge \\ ((\forall p : \text{pc})(p \in \text{compus}(\text{topo}(d))) \Rightarrow_{\text{L}} \\ (\text{buffer}(d, p) =_{\text{obs}} \text{buffer}(d', p) \wedge \\ \# \text{paquetesEnviados}(d, p) \\ \# \text{paquetesEnviados}(d', p)) \wedge \\ ((\forall p : \text{paquetes})((\exists c : \text{pc})(c \in \text{compus}(\text{topo}(d')) \wedge \\ c \in \text{compus}(\text{topo}(d')) \wedge_{\text{L}} (p \in \text{buffer}(d, c) \wedge \\ p \in \text{buffer}(d', c))) \Rightarrow_{\text{L}} \\ (\text{recorridoPaquete}(d, p) =_{\text{obs}} \text{recorridoPaquete}(d', p))) \end{array} \right) =_{\text{obs}}$$

**generadores**

CrearRed : topo  $\longrightarrow$  dcnet  
 Seg : dcnet  $\longrightarrow$  dcnet  
 CrearPaquete : dcnet dcn  $\times$  pc p1  $\times$  pc p2  $\times$  paquete  $\longrightarrow$  dcnet  
 $\{(p_1 \in \text{compus}(\text{topo}(\text{dcn})) \wedge p_2 \in \text{compus}(\text{topo}(\text{dcn}))) \wedge_{\text{L}} \text{conectadas}?( \text{topo}(\text{dcn}), p_1, p_2 )\}$

**observadores básicos**

topo : dcnet  $\longrightarrow$  topologia  
 $\# \text{paquetesEnviados}$  : dcnet dcn  $\times$  pc p  $\longrightarrow$  nat  $\{p \in \text{compus}(\text{topo}(\text{dcn}))\}$   
 buffer : dcnet dcn  $\times$  pc p  $\longrightarrow$  conj(paquete)  $\{p \in \text{compus}(\text{topo}(\text{dcn}))\}$

**otras operaciones**

recorridoPaquete : dcnet dcn  $\times$  nat id  $\longrightarrow$  secu(tupla(nat, nat, nat, nat))  
 $\{(paqueteEnTransito?(dcn, id)\}$   
 cortarRecHasta : secu(tupla(nat  $\times$  nat  $\times$  nat  $\times$  nat))  $\times$  nat  $\longrightarrow$  secu(tupla(nat, nat, nat, nat))  
 buscarPaquete : dcnet dcn  $\times$  conj(nat) pcs  $\times$  nat id  $\longrightarrow$  nat  
 $\{pcs = \text{compus}(\text{topo}(\text{dcn})) \wedge (\exists \text{ip} : \text{nat})(\text{ip} \in \text{pcs} \wedge \text{id} \in \text{buffer}(\text{dcn}, \text{ip}))\}$   
 $\pi_1$ Conj : conj(tupla(nat, nat, nat, nat))  $\longrightarrow$  conj(nat)  
 paqueteEnTransito? : dcnet  $\times$  nat  $\longrightarrow$  bool  
 existePaqEnBuffers? : dcnet dcn  $\times$  conj(nat) pcs  $\times$  nat id  $\longrightarrow$  bool  $\{pcs = \text{compus}(\text{topo}(\text{dcn}))\}$   
 perteneceBuffers? : paquete  $\times$  buffers  $\longrightarrow$  bool  
 darPaqueteEnviado : conj(paquete)  $\longrightarrow$  paquete  
 darPrioridad : dcnet dcn  $\times$  nat id  $\longrightarrow$  nat  $\{id \in \text{paquetesEnLaRed}(\text{dcn})\}$   
 buscarPrioridad : nat  $\times$  conj(paquetes)  $\longrightarrow$  nat  
 maxPrioridad : dcnet  $\times$  conj(pc)  $\longrightarrow$  nat

PaquetesConPrioridadKdcnet $\times$ conj(pc) $\times$ nat	$\longrightarrow$ paquete
paquetesEnLaRed : dcn	$\longrightarrow$ conj(paquete)
buscarPaquetesEnLaRed : dcn $\times$ conj(pc)	$\longrightarrow$ conj(paquete)
compuQueMasEnvio : dcn	$\longrightarrow$ pc
laQueMasEnvio : dcn $\times$ conj(pc)	$\longrightarrow$ pc
pasoSeg : topo $\times$ buffers $\times$ buffers	$\longrightarrow$ buffers
regresion : topo $\times$ buffers $\times$ secu(buffers)	$\longrightarrow$ buffers
generarHistoria : dcn $\times$ diccionario(pc $\times$ conj(paquete))	$\longrightarrow$ secu(buffers)
auxDefinir : buffers $\times$ pc $\times$ conj(paquete) $\times$ conj(paquete)	$\longrightarrow$ buffers
auxBorrar : buffers $\times$ pc $\times$ conj(paquete) $\times$ conj(paquete)	$\longrightarrow$ buffers
transacion : topo $\times$ buffers $\times$ conj(pc)	$\longrightarrow$ buffers
envio : topo $\times$ buffers $\times$ ip $\times$ conj(paquete)	$\longrightarrow$ buffers
nuevosPaquetes : buffers $\times$ buffers	$\longrightarrow$ buffers
pasarA : topologia $\times$ pc $\times$ pc	$\longrightarrow$ pc

**axiomas**  $\forall p, p': \text{paquete}, \forall c, c': \text{pc}, \forall dcn: \text{dcn}, \forall t: \text{topologia}$

topo(crearRed(t))	$\equiv t$
topo(seg(dcn))	$\equiv \text{topo(dcn)}$
topo(CrearPaquete(dcn, c, c', p))	$\equiv \text{topo(dcn)}$
#paquetesEnviados(crearRed(t), c)	$\equiv 0$
#paquetesEnviados(seg(dcn), c)	$\equiv \#paquetesEnviados(dcn)$
#paquetesEnviados(CrearPaquete(dcn, o, d, p), c)	$\equiv \text{if } c = o \text{ then } \#paquetesEnviados(dcn, c) + 1$ $\text{else } \#paquetesEnviados(dcn, c)$ $\text{fi}$
buffer(dcn, c)	$\equiv \text{obtener}(c, \text{regresion}(\text{topo(dcn)}, \text{vacio}, \text{generarHistoria(dcn, vacio)}))$
recorridoPaquete(dcn, p)	$\equiv \text{cortarRecHasta}(\text{darCaminoMasCorto}(\text{topo(dcn)}, \text{origen(p)}, \text{destino(p)}), \text{buscar}(\text{compus}(\text{topo(dcn)}), p))$
cortarRecHasta(s, ip)	$\equiv \text{if } \text{vacio?}(s) \vee \text{ip} = \text{ipOrigen}(\text{prim}(s)) \text{ then } \langle \rangle$ $\text{else } \text{prim}(s) \bullet \text{cortarRecHasta}(\text{fin}(s), \text{ip})$ $\text{fi}$
buscarPaquete(dcn, compus, id)	$\equiv \text{if } id \in \pi_1 \text{Conj}(\text{buffer(dcn, dameUno(compus))}) \text{ then } \text{dameUno(compus)}$ $\text{else } \text{buscarPaquete}(\text{sinUno(compus)}, id)$ $\text{fi}$
$\pi_1 \text{Conj}(\text{conjTuplas})$	$\equiv \text{if } \emptyset?(\text{conjTuplas}) \text{ then } \emptyset$ $\text{else } \text{Ag}(\pi_1(\text{dameUno}(\text{conjTuplas})), \pi_1 \text{Conj}(\text{sinUno}(\text{conjTuplas})))$ $\text{fi}$
paqueteEnTransito?(dcn, id)	$\equiv \text{existePqEnBuffers?}(dcn, \text{compus}(\text{topo(dcn)}), id)$

<code>existePaqEnBuffers?(dcn, pcs, id)</code>	$\equiv$ <b>if</b> $\emptyset?(pcs)$ <b>then</b> false <b>else</b> <b>if</b> $id \in \pi_1 \text{Conj}(\text{buffer}(dcn, \text{dameUno}(pcs)))$ <b>then</b> true <b>else</b> <code>existePaqEnBuffers?(dcn, sinUno(pcs), id)</code> <b>fi</b> <b>fi</b>
<code>buscarpaquetesEnLaRed(dcn,cc)</code>	$\equiv$ <b>if</b> $\emptyset?(cc)$ <b>then</b> $\emptyset$ <b>else</b> $\text{buffer}(dcn, \text{dameUno}(cc))$ <span style="float: right;"><math>\cup</math></span> <code>buscarpaquetesEnLaRed(dcn, sinUno(cc))</code> <b>fi</b>
<code>paquetesEnLaRed(dcn)</code>	$\equiv$ <code>buscarpaquetesEnLaRed(dcn, compus(topo(dcn)))</code>
<code>buscarPrioridad(id,cp)</code>	$\equiv$ <b>if</b> $i = \Pi_1(\text{dameUno}(cp))$ <b>then</b> $\Pi_4(\text{dameUno}(cp))$ <b>else</b> <code>darPrioridad(id, sinUno(cp))</code> <b>fi</b>
<code>darPrioridad(dcn,id)</code>	$\equiv$ <code>buscarPrioridad(id, compus(dcn))</code>
<code>darPaqueteEnviado(dcn,cp)</code>	$\equiv$ <code>dameUno(PaquetesConPrioridadK(dcn, cp, maxPrioridad(dcn, cp)))</code>
<code>maxPrioridad(dcn,cp)</code>	$\equiv$ <b>if</b> $\emptyset?(sinUno(cp))$ <b>then</b> <code>darPrioridad(dcn, dameUno(cp))</code> <b>else</b> $\max(\text{darPrioridad}(dcn, \text{dameUno}(cp)),$ $\text{maxPrioridad}(dcn, sinUno(cp)))$ <b>fi</b>
<code>PaquetesConPrioridadK(dcn,cp,k)</code>	$\equiv$ <b>if</b> $\emptyset?(cp)$ <b>then</b> $\emptyset$ <b>else</b> <b>if</b> $\text{darPrioridad}(dcn, \text{dameUno}(cp)) = k$ <b>then</b> $\text{Ag}(\text{dameUno}(cp), \text{PaquetesConPrioridadK}(dcn, sinUno(cp), k))$ <b>else</b> $\text{PaquetesConPrioridadK}(dcn, sinUno(cp), k)$ <b>fi</b> <b>fi</b>
<code>compuQueMasEnvio(dcn)</code>	$\equiv$ <code>laQueMasEnvio(dcn, compus(topo(dcn)))</code>
<code>laQueMasEnvio(dcn,cs)</code>	$\equiv$ <b>if</b> $\emptyset?(sinUno(cs))$ <b>then</b> <code>dameUno(cs)</code> <b>else</b> <b>if</b> $\#paquetesEnviados(dcn, \text{dameUno}(cs)) <$ $\#paquetesEnviados(dcn, \text{laQueMasEnvio}(dcn, sinUno(cs)))$ <b>then</b> <code>laQueMasEnvio(dcn, sinUno(cs))</code> <b>else</b> <code>dameUno(cs)</code> <b>fi</b> <b>fi</b>

perteneceBuffers?(p,bs)	≡ <b>if</b> $\emptyset?(claves(bs))$ <b>then</b> <i>false</i> <b>else</b> <b>if</b> $p \in obtener(dameUno(claves(bs)),bs)$ <b>then</b> <i>true</i> <b>else</b> <i>perteneceBuffers?(p, borrar(dameUno(claves(bs)),bs))</i> <b>fi</b> <b>fi</b>
generarHistoria(crearRed(t),bs)	≡ bs • $\langle \rangle$
generarHistoria(seg(dcn),bs)	≡ bs • generarHistoria(dcn,vaco)
generarHistoria(CrearPaquete(dcn,o,d,p),bs)	≡ <b>if</b> $def?(c,bs)$ <b>then</b> <i>generarHistoria(dcn, definir(c,n</i> <span style="float:right">∪</span> <i>obtener(o,bs),bs))</i> <b>else</b> <i>generarHistoria(dcn, definir(c,n))</i> <b>fi</b>
auxBorrar(bs,c,b,p)	≡ <b>if</b> $\emptyset?(p - \{b\})$ <b>then</b> <i>borrar(c,n)</i> <b>else</b> <i>borrar(c,bs) definir(c,p - \{b\},bs)</i> <b>fi</b>
regresion(t,bs,cbs)	≡ <b>if</b> $vacia?(fin(cbs))$ <b>then</b> <i>pasoSeg(bs,t,prim(cbs))</i> <b>else</b> <i>regresion(t,pasoSeg(bs,t,prim(cbs)),fin(cbs))</i> <b>fi</b>
pasoSeg(t,bs,nbs)	≡ nuevosPaquetes(transacion(t,bs,claves(bs)),nbs)
transacion(t,bs,cp)	≡ <b>if</b> $\emptyset?(cp)$ <b>then</b> <i>bs</i> <b>else</b> <i>transacion(t,envio(t,bs,dameUno(cp)),sinUno(cp))</i> <b>fi</b>
pasarA(t,o,d)	≡ $prim(caminoMin(t,o,d))$
envio(t,bs,ip,cp)	≡ <b>if</b> $\emptyset?(darPaqueteEnviado(cp))$ <b>then</b> <i>bs</i> <b>else</b> <b>if</b> $pasarA(t,ip,destino(darPaqueteEnviado(cp))) =$ <i>destino(darPaqueteEnviado(cp))</i> <b>then</b> <i>envio(t,quitarPaquete(bs,ip),ip,cp</i> <span style="float:right">−</span> <i>darPaqueteEnviado(cp))</i> <b>else</b> <i>envio(t,quitarPaquete(pasarPaquete(bs,ip,darPaqueteEnviado</i> <i>,ip,cp - darPaqueteEnviado(b)))</i> <b>fi</b> <b>fi</b>
nuevosPaquetes(bs,nbs)	≡ <b>if</b> $\emptyset?(claves(nbs))$ <b>then</b> <i>bs</i> <b>else</b> <i>nuevosPaquetes(auxDefinir(bs,dameUno(claves(nbs)),obtener</i> <i>(dameUno(claves(nbs)),nbs),obtener(dameUno</i> <i>(claves(nbs),bs)),sinUno(nbs))</i> <b>fi</b>

**Fin TAD**

### 3. TAD TOPOLOGÍA

Este TAD modela cómo se conectan las computadoras. Las IP son únicas entre compus de la topología. Las compus tienen interfaces numeradas con los naturales de manera consecutiva (todas funcionan perfecto y todo eso, el DC las cuida y mantiene como corresponde).

#### TAD TOPOLOGÍA

**géneros**      topologia

**igualdad observacional**

$$(\forall t, t' : \text{topo}) \left( t =_{\text{obs}} t' \iff \left( \begin{array}{l} (\text{compus}(t) =_{\text{obs}} \text{compus}(t')) \wedge_L \\ ((\forall p : \text{pc}) (p \in \text{compus}(t) \Rightarrow_L ( \\ \quad (\text{cablesEn}(t, p) =_{\text{obs}} \text{cablesEn}(t', p)) \wedge \\ \quad (\# \text{interfaces}(t, p) =_{\text{obs}} \# \text{interfaces}(t', p)) \\ \end{array} \right) \right) \right)$$

**generadores**

NuevaTopo	:		$\longrightarrow$ topologia
Compu	:	topologia $\times$ nat $ip \times$ nat	$\longrightarrow$ topologia $\{ \neg(ip \in \text{compus}(t)) \}$
Cable	:	topologia $\times$ nat $ipA \times$ nat $ifA \times$ nat $ipB \times$ nat $ifB$	$\longrightarrow$ topologia $\left\{ \begin{array}{l} (ipA \in \text{compus}(t) \wedge ipB \in \text{compus}(t)) \wedge_L \\ (ifA < \# \text{interfaces}(t, ipA)) \wedge \\ (ifB < \# \text{interfaces}(t, ipB)) \wedge \\ \neg(ifA \in \text{interfacesOcupadasDe}(t, ipA)) \wedge \\ \neg(ifB \in \text{interfacesOcupadasDe}(t, ipB)) \wedge \\ \neg(ipA \in \text{vecinas}(t, ipB)) \end{array} \right\}$

**observadores básicos**

compus	:	topologia	$\longrightarrow$ conj(nat)
cablesEn	:	topologia $t \times$ nat $ip$	$\longrightarrow$ conj(tupla(nat, nat)) $\{ ip \in \text{compus}(t) \}$
#interfaces	:	topologia $t \times$ nat $ip$	$\longrightarrow$ nat $\{ ip \in \text{compus}(t) \}$

**otras operaciones**

vecinas	:	topologia $t \times$ nat $ip$	$\longrightarrow$ conj(nat) $\{ ip \in \text{compus}(t) \}$
interfacesOcupadasDe	:	topologia $t \times$ nat $ip$	$\longrightarrow$ conj(nat) $\{ ip \in \text{compus}(t) \}$
conectados?	:	topologia $t \times$ nat $ipA \times$ nat $ipB$	$\longrightarrow$ bool $\{ ipA \in \text{compus}(t) \wedge ipB \in \text{compus}(t) \}$
darInterfazConectada	:	conj(tupla(nat, nat)) cablesA $\times$ nat $ipB$	$\longrightarrow$ nat $\{ ipB \in \Pi_2 \text{Conj}(\text{cablesA}) \}$
darSegmento	:	topologia $t \times$ nat $ipA \times$ nat $ipB$	$\longrightarrow$ segmento $\{ ipA \in \text{compus}(t) \wedge_L ipB \in \text{vecinas}(t, ipA) \}$
estáEnRuta?	:	secu(segmento) ruta $\times$ nat $ip$	$\longrightarrow$ bool
darSiguientePc	:	secu(segmento) ruta $\times$ nat $ip$	$\longrightarrow$ nat $\{ \text{estáEnRuta?}(ruta, ip) \}$
darCaminoMasCorto	:	topologia $t \times$ nat $ipA \times$ nat $ipB$	$\longrightarrow$ secu(segmento) $\{ ipA \in \text{compus}(t) \wedge ipB \in \text{compus}(t) \wedge_L \text{conectados?}(t, ipA, ipB) \}$
darRutas	:	topologia $\times$ nat $ipA \times$ nat $ipB \times$ conj(nat) $\times$ secu(segmento)	$\longrightarrow$ conj(secu(segmento)) $\{ ipA \in \text{compus}(t) \wedge ipB \in \text{compus}(t) \}$

darRutasVecinas	: $\text{topologia} \times \text{conj}(\text{nat}) \times \text{nat } ip \times \text{conj}(\text{nat}) \times \text{se-}$ $\text{cu(segmento)}$	$\longrightarrow \text{conj}(\text{secu}(\text{segmento}))$ $\{ip \in \text{compus}(t)\}$
longMenorSec	: $\text{conj}(\text{secu}(\alpha))$	$\longrightarrow \text{nat}$
secusDeLongK	: $\text{conj}(\text{secu}(\alpha)) \times \text{nat}$	$\longrightarrow \text{conj}(\text{secu}(\alpha))$
$\Pi_1 \text{Conj}$	: $\text{conj}(\text{tupla}(\text{nat}, \text{nat}))$	$\longrightarrow \text{conj}(\text{nat})$
$\Pi_2 \text{Conj}$	: $\text{conj}(\text{tupla}(\text{nat}, \text{nat}))$	$\longrightarrow \text{conj}(\text{nat})$
<b>axiomas</b>	$\forall t: \text{topologia}, \forall ipNueva, ip, ipA, ipB, ifA, ifB, cantIfaces, k: \text{nat}, \forall conjDuplas: \text{conj}(\text{tupla}(\text{nat}, \text{nat})), \forall conjCablesIpA: \text{conj}(\text{tupla}(\text{nat}, \text{nat})), \forall cs, rec, vecinas: \text{conj}(\text{nat}), \forall secus: \text{conj}(\text{secu}(\alpha)), \forall sc: \text{conj}(\text{secu}(\alpha)), \forall ruta: \text{secu}(\text{segmento})$	
compus(NuevaTopo)	$\equiv \emptyset$	
compus(Compu( $t, ipNueva, cantIfaces$ ))	$\equiv \text{Ag}(ipNueva, \text{compus}(t))$	
compus(Cable( $t, ipA, ifA, ipB, ifB$ ))	$\equiv \text{compus}(t)$	
cablesEn(NuevaTopo, $ip$ )	$\equiv \emptyset$	
cablesEn(Compu( $t, ipNueva, cantIfaces$ ), $ip$ )	$\equiv \text{cablesEn}(t, ip)$	
cablesEn(Cable( $t, ipA, ifA, ipB, ifB$ ), $ip$ )	$\equiv \text{if } ip = ipA \text{ then } \text{Ag}(\langle ifA, ipB \rangle, \emptyset) \text{ else } \emptyset \text{ fi} \cup$ $\text{if } ip = ipB \text{ then } \text{Ag}(\langle ifB, ipA \rangle, \emptyset) \text{ else } \emptyset \text{ fi} \cup$ $\text{cablesEn}(t, ip)$	
#interfaces(NuevaTopo, $ip$ )	$\equiv 0$	
#interfaces(Compu( $t, ipNueva, cantIfaces$ ), $ip$ )	$\equiv \text{if } ip = ipNueva \text{ then}$ $\quad cantIfaces$ $\text{else}$ $\quad \#interfaces(t, ip)$ $\text{fi}$	
#interfaces(Cable( $t, ipA, ifA, ipB, ifB$ ), $ip$ )	$\equiv \#interfaces(t, ip)$	
interfacesOcupadasDe( $t, ip$ )	$\equiv \Pi_1 \text{Conj}(\text{cablesEn}(t, ip))$	
vecinas( $t, ip$ )	$\equiv \Pi_2 \text{Conj}(\text{cablesEn}(t, ip))$	
conectados?( $t, ipA, ipB$ )	$\equiv \neg \emptyset?(\text{darRutas}(t, ipA, ipB, \emptyset, <>))$	
darInterfazConectada( $conjCablesIpA, ipB$ )	$\equiv \text{if } ipB = \Pi_2(\text{dameUno}(conjCablesIpA)) \text{ then}$ $\quad \Pi_1(\text{dameUno}(conjCablesIpA))$ $\text{else}$ $\quad \text{darInterfazConectada}(\text{sinUno}(conjCablesIpA), ipB)$ $\text{fi}$	
darSegmento( $t, ipA, ipB$ )	$\equiv \langle ipA, \text{darInterfazConectada}(\text{cablesEn}(t, ipA), ipB),$ $ipB, \text{darInterfazConectada}(\text{cablesEn}(t, ipB), ipA) \rangle$	
estáEnRuta?( $ruta, ip$ )	$\equiv \text{if vacía?}(ruta) \text{ then}$ $\quad \text{false}$ $\text{else}$ $\quad \text{if } \Pi_1(\text{prim}(ruta)) = ip \text{ then}$ $\quad \quad \text{true}$ $\quad \text{else}$ $\quad \quad \text{estáEnRuta?}(\text{fin}(ruta), ip)$ $\quad \text{fi}$ $\text{fi}$	
darSiguientePc( $ruta, ip$ )	$\equiv \text{if } \Pi_1(\text{prim}(ruta)) = ip \text{ then}$ $\quad \Pi_3(\text{prim}(ruta))$ $\text{else}$ $\quad \text{darSiguientePc}(\text{fin}(ruta), ip)$ $\text{fi}$	
darCaminoMasCorto( $t, ipA, ipB$ )	$\equiv \text{dameUno}(\text{secusDeLongK}(\text{darRutas}(t, ipA, ipB, \emptyset, <>),$ $\text{longMenorSec}(\text{darRutas}(t, ipA, ipB, \emptyset, <>)))$	

```

darRutas(t, ipA, ipB, rec, ruta)  ≡  if ipB ∈ vecinas(t, ipA) then
    Ag(ruta ∘ darSegmento(t, ipA, ipB) , ∅)
else
    if ∅?(vecinas(t, ipA) - rec) then
        ∅
    else
        darRutas(t, dameUno(vecinas(t, ipA) - rec),
            ipB, Ag(ipA, rec),
            ruta ∘ darSegmento(t, ipA, dameUno(vecinas(t, ipA) - rec))) ∪
            darRutasVecinas(t, sinUno(vecinas(t, ipA) - rec),
            ipB, Ag(ipA, rec),
            ruta ∘ darSegmento(t, ipA, dameUno(vecinas(t, ipA) - rec)))
    fi
fi

darRutasVecinas(t, vecinas, ipB, rec, ruta)  ≡  if ∅?(vecinas) then
    ∅
else
    darRutas(t, dameUno(vecinas), ipB, rec, ruta) ∪
    darRutasVecinas(t, sinUno(vecinas), ipB, rec, ruta)
fi

darCaminoMasCorto(t, ipA, ipB)  ≡  dameUno(secusDeLongK(darRutas(t, ipA, ipB, ∅, <>),
    longMenorSec(darRutas(t, ipA, ipB, ∅, <>)))

secusDeLongK(secus, k)  ≡  if ∅?(secus) then
    ∅
else
    if long(dameUno(secus)) = k then
        dameUno(secus) ∪ secusDeLongK(sinUno(secus), k)
    else
        secusDeLongK(sinUno(secus), k)
    fi
fi

longMenorSec(secus)  ≡  if ∅?(secus) then
    0
else
    min(long(dameUno(secus)),
        longMenorSec(sinUno(secus)))
fi

Π1Conj(conjDuplas)  ≡  if ∅?(conjDuplas) then
    ∅
else
    Ag(Π1(dameUno(conjDuplas)),
        Π1Conj(sinUno(conjDuplas)))
fi

Π2Conj(conjDuplas)  ≡  if ∅?(conjDuplas) then
    ∅
else
    Ag(Π2(dameUno(conjDuplas)),
        Π2Conj(sinUno(conjDuplas)))
fi

```

**Fin TAD**