



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Trabajo Práctico de Especificación

Análisis Habitacional Argentino

08/09/2021

Lc. Computación

Integrante	LU	Correo electrónico
Simon, Martino	374/21	martinosimon@gmail.com
Ramis, Ezequiel	881/21	ezequielramis.hello@gmail.com



**Facultad de Ciencias Exactas y Naturales**  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (++54 +11) 4576-3300

<http://www.exactas.uba.ar>

# Índice

1. Auxiliares para base de datos y generales	2
2. Especificación	3
2.1. Ejercicio 1 . . . . .	3
2.2. Ejercicio 2 . . . . .	5
2.3. Ejercicio 3 . . . . .	5
2.4. Ejercicio 4 . . . . .	6
2.5. Ejercicio 5 . . . . .	7
2.6. Ejercicio 6 . . . . .	7
2.7. Ejercicio 7 . . . . .	7
2.8. Ejercicio 8 . . . . .	8
2.9. Ejercicio 9 . . . . .	8
2.10. Ejercicio 10 . . . . .	9
2.11. Ejercicio 11 . . . . .	9

# 1. Auxiliares para base de datos y generales

Antes de empezar con la especificación de procedimientos, vamos a definir auxiliares para acceder más declarativamente a los atributos de cada tabla, y para uso general:

```
/* Tabla Hogares */
```

```
aux @hogCodusu :  $\mathbb{Z}$  = ord(HOGCODUSU);
aux @hogAño :  $\mathbb{Z}$  = ord(HOGAÑO);
aux @hogTrimestre :  $\mathbb{Z}$  = ord(HOGTRIMESTRE);
aux @hogLatitud :  $\mathbb{Z}$  = ord(HOGLATITUD);
aux @hogLongitud :  $\mathbb{Z}$  = ord(HOGLONGITUD);
aux @ii7 :  $\mathbb{Z}$  = ord(II7);
aux @region :  $\mathbb{Z}$  = ord(REGION);
aux @mas500 :  $\mathbb{Z}$  = ord(MAS_500);
aux @iv1 :  $\mathbb{Z}$  = ord(IV1);
aux @iv2 :  $\mathbb{Z}$  = ord(IV2);
aux @ii2 :  $\mathbb{Z}$  = ord(II2);
aux @ii3 :  $\mathbb{Z}$  = ord(II3);
```

```
/* Tabla Personas */
```

```
aux @indCodusu :  $\mathbb{Z}$  = ord(INDCODUSU);
aux @componente :  $\mathbb{Z}$  = ord(COMPONENTE);
aux @indAño :  $\mathbb{Z}$  = ord(INDAÑO);
aux @indTrimestre :  $\mathbb{Z}$  = ord(INDTRIMESTRE);
aux @ch4 :  $\mathbb{Z}$  = ord(CH4);
aux @ch6 :  $\mathbb{Z}$  = ord(CH6);
aux @nivelEd :  $\mathbb{Z}$  = ord(NIVELED);
aux @estado :  $\mathbb{Z}$  = ord(ESTADO);
aux @cat0cup :  $\mathbb{Z}$  = ord(CATOCUP);
aux @p47t :  $\mathbb{Z}$  = ord(P47T);
aux @pp04g :  $\mathbb{Z}$  = ord(PP04G);
```

```
/* Varios */
```

```
aux #atributosHogar :  $\mathbb{Z}$  = 12;
aux #atributosPersona :  $\mathbb{Z}$  = 11;
aux #habitaciones (h: hogar) : dato = h[@iv2];
aux #dormitorios (h: hogar) : dato = h[@ii2];
aux #personasEnHogar (ti:  $eph_i$ , h: hogar) :  $\mathbb{Z}$  =  $\sum_{i=0}^{|ti|-1}$  if personaEnHogar(ti[i], h) then 1 else 0 fi;
pred esCasa (h: hogar) {
  h[@iv1] = 1
}
pred personaEnHogar (i: individuo, h: hogar) {
  i[@indCodusu] = h[@hogCodusu]
}
pred hogaresConTrimestresIguales (h: hogar, g: hogar) {
  h[@hogTrimestre] = g[@hogTrimestre]
}
```

## 2. Especificación

### 2.1. Ejercicio 1

```
proc esEncuestaVálida (in th:  $eph_h$ , in ti:  $eph_i$ , out result: Bool) {  
  Pre {True}  
  Post {result = true  $\leftrightarrow$  encuestaValida(th, ti)}  
}  
  
pred encuestaValida (th:  $eph_h$ , ti:  $eph_i$ ) {  
  esMatriz(th)  $\wedge$  esMatriz(ti)  $\wedge$   
  |th| > 0  $\wedge$  |ti| > 0  $\wedge_L$   
  esTablaCompleta(th, #atributosHogar)  $\wedge$  esTablaCompleta(ti, #atributosPersona)  $\wedge_L$   
   $\neg$ hayIndividuoSinHogar(th, ti)  $\wedge$   $\neg$ hayHogarSinIndividuo(th, ti)  $\wedge$   
   $\neg$ hayIndividuoRepetido(ti)  $\wedge$   $\neg$ hayHogarRepetido(th)  $\wedge$   
  mismaFechaDeRelevamiento(th, ti)  $\wedge$   
  todosHogaresConMenosDe21Integrantes(th, ti)  $\wedge$   
  todosHogaresConMasHabitacionesQueDormitorios(th)  $\wedge$   
  todosAtributosEnRango(th, ti)  $\wedge$   
  atributosValidos(th, ti)  
}  
  
pred esMatriz (s: seq<seq<dato>>) {  
  |s| > 0  $\longrightarrow_L \neg(\exists t : seq<dato>)$  (  
    (t  $\in$  s)  $\wedge$  |head(s)|  $\neq$  |t|  
  )  
}  
  
pred esTablaCompleta (s: seq<seq<dato>>, l:  $\mathbb{Z}$ ) {  
  ( $\forall f : seq<dato>)$  (  
    f  $\in$  s  $\longrightarrow$  |f| = l  
  )  
}  
  
pred hayIndividuoSinHogar (th:  $eph_h$ , ti:  $eph_i$ ) {  
  ( $\exists i : individuo$ ) (  
    i  $\in$  ti  $\wedge_L$   
     $\neg(\exists h : hogar)$  (  
      h  $\in$  th  $\wedge_L$  personaEnHogar(i, h)  
    )  
  )  
}  
  
pred hayHogarSinIndividuo (th:  $eph_h$ , ti:  $eph_i$ ) {  
  ( $\exists h : hogar$ ) (  
    h  $\in$  th  $\wedge_L$   
     $\neg(\exists i : individuo)$  (  
      i  $\in$  ti  $\wedge_L$  personaEnHogar(i, h)  
    )  
  )  
}  
  
pred hayIndividuoRepetido (s:  $eph_i$ ) {  
  ( $\exists a, b : individuo$ ) (  
    (a  $\in$  s  $\wedge$  b  $\in$  s  $\wedge$  a  $\neq$  b)  $\wedge_L$  a[@indCodusu] = b[@indCodusu]  $\wedge$  a[@componente] = b[@componente]  
  )  
}
```

```

}
pred hayHogarRepetido (s:  $eph_h$ ) {
  ( $\exists a, b : hogar$ ) (
    ( $a \in s \wedge b \in s \wedge a \neq b$ )  $\wedge_L a[@hogCodusu] = b[@hogCodusu]$ 
  )
}
pred mismaFechaDeRelevamiento (th:  $eph_h$ , ti:  $eph_i$ ) {
  ( $|th| > 0 \wedge |ti| > 0$ )  $\rightarrow_L$  (
    ( $\forall h : hogar$ ) (
       $h \in th \rightarrow_L$  (
         $h[@hogAño] = head(h)[@hogAño] \wedge$ 
         $hogaresConTrimestresIguales(h, head(th))$ 
      )
    )  $\wedge$ 
    ( $\forall i : individuo$ ) (
       $i \in ti \rightarrow_L$  (
         $i[@indAño] = head(ti)[@indAño] \wedge$ 
         $i[@indTrimestre] = head(ti)[@indTrimestre] \wedge$ 
         $i[@indAño] = head(th)[@hogAño] \wedge$ 
         $i[@indTrimestre] = head(th)[@hogTrimestre]$ 
      )
    )
  )
}
pred todosHogaresConMenosDe21Integrantes (th:  $eph_h$ , ti:  $eph_i$ ) {
  ( $\forall h : hogar$ ) (
     $h \in th \rightarrow_L 21 > \#personasEnHogar(ti, h)$ 
  )
}
pred todosHogaresConMasHabitacionesQueDormitorios (th:  $eph_h$ ) {
  ( $\forall h : hogar$ ) (
     $h \in th \rightarrow_L \#habitaciones(h) \geq \#dormitorios(h)$ 
  )
}
pred todosAtributosEnRango (th:  $eph_h$ , ti:  $eph_i$ ) {
  ( $\forall h : hogar$ ) (
     $h \in th \rightarrow_L$ 
     $1 \leq h[@ii7] \leq 3 \wedge$ 
     $1 \leq h[@region] \leq 6 \wedge$ 
     $0 \leq h[@mas500] \leq 1 \wedge$ 
     $1 \leq h[@iv1] \leq 5 \wedge$ 
     $1 \leq h[@ii3] \leq 2$ 
  )  $\wedge$ 
  ( $\forall i : individuo$ ) (
     $i \in ti \rightarrow_L$ 
     $1 \leq i[@ch4] \leq 2 \wedge$ 
     $0 \leq i[@nivelEd] \leq 1 \wedge$ 
     $-1 \leq i[@estado] \leq 1 \wedge$ 
     $0 \leq i[@catOcup] \leq 4 \wedge$ 
     $1 \leq i[@pp04g] \leq 10 \wedge$ 
     $-1 \leq i[@p47t]$ 
  )
}

```

```

}
pred atributosValidos (th:  $eph_h$ , ti:  $eph_i$ ) {
  ( $\forall h : hogar$ ) (
     $h \in th \rightarrow_L$ 
     $0 < h[@hogCodusu] \wedge$ 
     $1800 \leq h[@hogAño] \wedge$ 
     $1 \leq h[@hogTrimestre] \leq 4 \wedge$ 
     $1 \leq \#habitaciones(h) \wedge$ 
     $1 \leq \#dormitorios(h)$ 
  )  $\wedge$ 
  ( $\forall i : individuo$ ) (
     $i \in ti \rightarrow_L$ 
     $0 < i[@indCodusu] \wedge$ 
     $1 \leq i[@componente] \wedge$ 
     $1800 \leq i[@indAño] \wedge$ 
     $1 \leq i[@indTrimestre] \leq 4 \wedge$ 
     $0 \leq i[@ch6]$ 
  )
}

```

## 2.2. Ejercicio 2

```

proc histHabitacional (in th:  $eph_h$ , in ti:  $eph_i$ , in region:  $\mathbb{Z}$ , out res:  $seq\langle\mathbb{Z}\rangle$ ) {
  Pre { $encuestaValida(th, ti) \wedge 1 \leq region \leq 6$ }
  Post { $longitudDeHistograma(th, region, res) \wedge$ 
    ( $\forall i : \mathbb{Z}$ ) ( $0 \leq i < |res| \rightarrow_L res[i] = \#casaConNHabitacionesEnRegion(th, i + 1, region)$ )
  }
}

aux #casaConNHabitacionesEnRegion (th:  $eph_h$ , n:  $\mathbb{Z}$ , r: dato) :  $\mathbb{Z} =$ 
   $\sum_{i=0}^{|th|-1} \text{if } esCasa(th[i]) \wedge \#habitaciones(th[i]) = n \wedge th[i][@region] = r \text{ then } 1 \text{ else } 0 \text{ fi}$ 
;
pred longitudDeHistograma (th:  $eph_h$ , r:  $\mathbb{Z}$ , res:  $seq\langle\mathbb{Z}\rangle$ ) {
  ( $\exists h : hogar$ ) (
     $h \in th \wedge_L esCasa(h) \wedge h[@region] = r \wedge$ 
    ( $\forall g : hogar$ ) (
      ( $g \in th \wedge_L esCasa(g) \wedge g[@region] = r$ )  $\rightarrow \#habitaciones(h) \geq \#habitaciones(g)$ 
    )  $\wedge$ 
     $\#habitaciones(h) = |res|$ 
  )
}

```

## 2.3. Ejercicio 3

```

proc laCasaEstaQuedandoChica (in th:  $eph_h$ , in ti:  $eph_i$ , out res:  $seq\langle\mathbb{R}\rangle$ ) {
  Pre { $encuestaValida(th, ti)$ }
  Post { $|res| = 6 \wedge_L$ 
    ( $\forall i : \mathbb{Z}$ ) (
       $0 \leq i < |res| \rightarrow_L ($ 
        ( $\#casasPorRegion(th, i + 1) \neq 0 \rightarrow$ 
           $res[i] = \#casasCriticasPorRegion(th, ti, i + 1) / \#casasPorRegion(th, i + 1)$ 
        )
      )
    }
  }

```

```

    ) ∧
    (#casasPorRegion(th, i + 1) = 0 → res[i] = 0)
  )
}
}
}
aux #casasPorRegion (th: ephh, r: ℤ) : ℤ =
  ∑i=0|th|-1 if esCasa(th[i]) ∧ th[i][@region] = r ∧ ¬estaEnGranAglomerado(th[i]) then 1 else 0 fi
;
aux #casasCriticasPorRegion (th: ephh, ti: ephi, r: ℤ) : ℤ =
  ∑i=0|th|-1 if esCasaCritica(th[i], ti) ∧ th[i][@region] = r then 1 else 0 fi
;
pred esCasaCritica (h: hogar, ti: ephi) {
  esCasa(h) ∧ esHacinamientoCritico(h, ti) ∧ ¬estaEnGranAglomerado(h)
}
pred esHacinamientoCritico (h: hogar, ti: ephi) {
  #personasEnHogar(ti, h) / #dormitorios(h) > 3
}
pred estaEnGranAglomerado (h: hogar) {
  h[@mas500] = 1
}

```

## 2.4. Ejercicio 4

```

proc creceElTeleworkingEnCiudadesGrandes (in t1h: ephh, in t1i: ephi, in t2h: ephh, in t2i: ephi, out res: Bool) {
  Pre {(encuestaValida(t1h, t1i) ∧ encuestaValida(t2h, t2i)) ∧L (
    esHogarConAñoPrevio(head(t1h), head(t2h)) ∧
    hogaresConTrimestresIguales(head(t1h), head(t2h))
  )}
  Post {res = true ↔ teleworkingEnCiudadesGrandes(t1h, t1i) < teleworkingEnCiudadesGrandes(t2h, t2i)}
}

pred esHogarConAñoPrevio (h: hogar, g: hogar) {
  h[@hogAño] < g[@hogAño]
}

aux teleworkingEnCiudadesGrandes (th: ephh, ti: ephi) : ℝ =
  if #personasTrabajandoEnCiudadesGrandes(th, ti) ≠ 0 then
    #personasTrabajandoADistanciaEnCiudadesGrandes(th, ti) / #personasTrabajandoEnCiudadesGrandes(th, ti)
  else 0 fi
;
aux #personasTrabajandoADistanciaEnCiudadesGrandes (th: ephh, ti: ephi) : ℤ =
  ∑i=0|ti|-1 if viveEnHogarAptoParaTeleworking(ti[i], th) ∧ trabajaADistancia(ti[i]) then 1 else 0 fi
;
aux #personasTrabajandoEnCiudadesGrandes (th: ephh, ti: ephi) : ℤ =
  ∑i=0|ti|-1 if viveEnCiudadGrande(ti[i], th) ∧ trabaja(ti[i]) then 1 else 0 fi
;
pred viveEnHogarAptoParaTeleworking (i: individuo, th: ephh) {
  (∃h : hogar) (
    h ∈ th ∧L
    personaEnHogar(i, h) ∧ esCasaODepto(h) ∧ h[@mas500] = 1 ∧ h[@ii3] = 1
  )
}

```

```

}
pred esCasaODepto (h: hogar) {
   $h[@iv1] = 1 \vee h[@iv1] = 2$ 
}
pred trabajaADistancia (i: individuo) {
   $i[@pp04g] = 6 \wedge trabaja(i)$ 
}
pred trabaja (i: individuo) {
   $i[@estado] = 1$ 
}
pred viveEnCiudadGrande (i: individuo, th:  $eph_h$ ) {
  ( $\exists h : hogar$ ) (
     $h \in th \wedge_L$ 
     $personaEnHogar(i, h) \wedge esCasaODepto(h) \wedge h[@mas500] = 1$ 
  )
}

```

## 2.5. Ejercicio 5

```

proc costoSubsidioMejora (in th:  $eph_h$ , in ti:  $eph_i$ , in monto:  $\mathbb{Z}$ , out res:  $\mathbb{Z}$ ) {
  Pre { $encuestaValida(th, ti) \wedge monto > 0$ }
  Post { $res = \sum_{i=0}^{|th|-1} \text{if } esHogarSubsidiado(th[i], ti) \text{ then } monto \text{ else } 0 \text{ fi}$ }
}
pred esHogarSubsidiado (h: hogar, ti:  $eph_i$ ) {
   $esCasa(h) \wedge h[@ii7] = 1 \wedge \#dormitorios(h) < \#personasEnHogar(ti, h) - 2$ 
}

```

## 2.6. Ejercicio 6

```

proc generarJoin (in th:  $eph_h$ , in ti:  $eph_i$ , out junta:  $joinHI$ ) {
  Pre { $encuestaValida(th, ti)$ }
  Post { $(\forall hi : hogar \times individuo)($ 
     $hi \in junta \longrightarrow ($ 
       $hi_0 \in th \wedge hi_1 \in ti \wedge_L hi_0[@hogCodusu] = hi_1[@indCodusu]$ 
    )
  ) }
}

```

## 2.7. Ejercicio 7

```

proc ordenarRegionYTipo (inout th:  $eph_h$ , inout ti:  $eph_i$ ) {
  Pre { $encuestaValida(th, ti) \wedge th = TH \wedge ti = TI$ }
  Post {
     $|th| = |TH| \wedge |ti| = |TI| \wedge_L$ 
     $(\forall h : hogar)(h \in th \leftrightarrow h \in TH) \wedge$ 
     $(\forall i : individuo)(i \in ti \leftrightarrow i \in TI) \wedge$ 
     $hogaresOrdenados(th) \wedge individuosOrdenados(th, ti)$ 
  }
}
pred hogaresOrdenados (th:  $eph_h$ ) {
   $(\forall i : \mathbb{Z})(0 \leq i < |th| - 1 \longrightarrow_L ($ 

```



```

    th[i][@region] ≤ th[i + 1][@region] ∧
    (th[i][@region] = th[i + 1][@region] → th[i][@hogCodusu] < th[i + 1][@hogCodusu])
  )
}
pred individuosOrdenados (th: ephh, ti: ephi) {
  (∀i : ℤ)(0 ≤ i < |ti| - 1 →L
    indicePorCodusu(th, ti[i][@indCodusu]) = indicePorCodusu(th, ti[i + 1][@indCodusu]) → (
      ti[i][@indCodusu] = ti[i + 1][@indCodusu] ∧ ti[i][@componente] < ti[i + 1][@componente]
    ) ∧
    indicePorCodusu(th, ti[i][@indCodusu]) > indicePorCodusu(th, ti[i + 1][@indCodusu]) → (
      ti[i][@indCodusu] > ti[i + 1][@indCodusu]
    ) ∧
    indicePorCodusu(th, ti[i][@indCodusu]) < indicePorCodusu(th, ti[i + 1][@indCodusu]) → (
      ti[i][@indCodusu] < ti[i + 1][@indCodusu]
    )
  )
}
/* Asumo que la lista tiene elementos únicos, y que el elemento pertenece a la lista */
aux indicePorCodusu (s: ephh, e: dato) : ℤ = ∑i=0|s|-1 if s[i][@hogCodusu] = e then i else 0 fi ;

```

## 2.8. Ejercicio 8

```

proc muestraHomogenea (in th: ephh, in ti: ephi, out res: seq⟨hogar⟩) {
  Pre { encuestaValida(th, ti) }
  Post { (∃s : seq⟨hogar⟩) (
    esHomogenea(s, th, ti) ∧
    (∀t : seq⟨hogar⟩)(esHomogenea(t, th, ti) → |s| ≥ |t|) ∧ (
      (|s| < 3 → res = <>) ∧
      (|s| ≥ 3 → res = s)
    )
  ) }
}

pred esHomogenea (s: seq⟨hogar⟩, th: ephh, ti: ephi) {
  mismaDiferencia(s) ∧ ordenadaPorIngresos(s, ti) ∧ (∀h : hogar)(h ∈ s → h ∈ th)
}

pred mismaDiferencia (s: seq⟨hogar⟩) {
  (∀i : ℤ)(1 ≤ i < |s| - 1 →L diferenciaDeIngresos(s[i], s[i - 1], ti) = diferenciaDeIngresos(s[i], s[i + 1], ti))
}

pred ordenadaPorIngresos (s: seq⟨hogar⟩, ti: ephi) {
  (∀i : ℤ)(0 ≤ i < |s| - 1 →L ingresosEnHogar(s[i], ti) ≤ ingresosEnHogar(s[i + 1], ti))
}

aux diferenciaDeIngresos (a, b: hogar, ti: ephi) : ℤ = |ingresosEnHogar(a, ti) - ingresosEnHogar(b, ti)| ;
aux ingresosEnHogar (h: hogar, ti: ephi) : ℤ =
  ∑i=0|ti|-1 if personaEnHogar(ti[i], h) ∧ ti[i][@p47T] ≠ -1 then ti[i][@p47T] else 0 fi
;

```

## 2.9. Ejercicio 9

```

proc corregirRegion (inout th: ephh, in ti: ephi) {

```

```

Pre {encuestaValida(th, ti) ∧ th = TH}
Post {|th| = |TH| ∧L (∀i :  $\mathbb{Z}$ ) (0 ≤ i < |th| →L
  (TH[i][@region] = 1 →
    th[i][@region] = 5 ∧
    th[i][@hogCodusu] = TH[i][@hogCodusu] ∧
    th[i][@hogAño] = TH[i][@hogAño] ∧
    th[i][@hogTrimestre] = TH[i][@hogTrimestre] ∧
    th[i][@hogLatitud] = TH[i][@hogLatitud] ∧
    th[i][@hogLongitud] = TH[i][@hogLongitud] ∧
    th[i][@ii7] = TH[i][@ii7] ∧
    th[i][@mas500] = TH[i][@mas500] ∧
    th[i][@iv1] = TH[i][@iv1] ∧
    th[i][@iv2] = TH[i][@iv2] ∧
    th[i][@ii2] = TH[i][@ii2] ∧
    th[i][@ii3] = TH[i][@ii3]
  ) ∧
  (TH[i][@region] ≠ 1 → th[i] = TH[i])
)}
}

```

## 2.10. Ejercicio 10

```

proc histogramaDeAnillosConcentricos (in th: ephh, in centro:  $\mathbb{Z} \times \mathbb{Z}$ , in distancias: seq( $\mathbb{Z}$ ), out result: seq( $\mathbb{Z}$ )) {
  Pre {
    |distancias| > 0 ∧
    (∀i :  $\mathbb{Z}$ ) (0 ≤ i < |distancias| - 1 →L 0 < distancias[i] < distancias[i + 1])
  }
  Post {|result| = |distancias| ∧L
    (∀i :  $\mathbb{Z}$ ) (
      0 ≤ i < |result| →L (
        (i = 0 → res[i] = hogaresEnRadio(th, centro, -1, distancias[0])) ∧
        (i ≠ 0 → res[i] = hogaresEnRadio(th, centro, distancias[i - 1], distancias[i]))
      )
    )
  }
}

aux hogaresEnRadio (th: ephh, centro:  $\mathbb{Z} \times \mathbb{Z}$ , desde:  $\mathbb{Z}$ , hasta:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  =
   $\sum_{j=0}^{|th|-1}$  if desde < distanciaEuclidiana(centro, (th[j][@hogLatitud], th[j][@hogLongitud])) ≤ hasta then 1 else 0 fi
;
aux distanciaEuclidiana (c1, c2:  $\mathbb{Z} \times \mathbb{Z}$ ) :  $\mathbb{R}$  =  $\sqrt{(c2_0 - c1_0)^2 + (c2_1 - c1_1)^2}$ ;

```

## 2.11. Ejercicio 11

```

proc quitarIndividuos (inout th: ephh, inout ti: ephi, in busqueda: seq(ItemIndividuo × dato), out result: ephh × ephi)
{
  Pre {encuestaValida(th, ti) ∧ busquedaValida(busqueda) ∧ th = TH ∧ ti = TI}
  Post {0 ≤ |th| ≤ |TH| ∧ 0 ≤ |ti| ≤ |TI| ∧ |result| = |TI| - |ti| ∧L
    (∀h : hogar) (
      (h ∈ TH ∧ hogarLlenoDeCoincidentes(TI, h, busqueda)) ↔ (h ∈ result0 ∧ h ∉ th)
    ) ∧
    (∀i : individuo) (

```

```

        (i ∈ TI ∧ coincideConTerminos(i, busqueda)) ↔ (i ∈ result1 ∧ i ∉ ti)
    )
}
}
aux #personasEnHogarCoincidentes (ti: ephi, h: hogar, busqueda: seq⟨ItemIndividuo × dato⟩) : ℤ =
    ∑j=0|ti|-1 if personaEnHogar(ti[j], h) ∧ coincideConTerminos(ti[j], busqueda) then 1 else 0 fi
;
pred hogarLlenoDeCoincidentes (ti: ephi, h: hogar, busqueda: seq⟨ItemIndividuo × dato⟩) {
    #personasEnHogar(ti, h) = #personasEnHogarCoincidentes(ti, h, busqueda)
}
pred busquedaValida (busqueda: seq⟨ItemIndividuo × dato⟩) {
    (∀b : ItemIndividuo × dato) (
        b ∈ busqueda → def(ord(b0))
    ) ∧L
    ¬(∃i, j : ℤ) (
        0 ≤ i, j < |busqueda| ∧ i ≠ j ∧L ord(busqueda[i]0) = ord(busqueda[j]0)
    )
}
pred coincideConTerminos (i : individuo, busqueda: seq⟨ItemIndividuo × dato⟩) {
    (∀t : ℤ) (
        0 ≤ t < |busqueda| →L i[ord(busqueda[t]0)] = busqueda[t]1
    )
}

```