



# TP de Especificación

## Análisis Habitacional Argentino

1 de Abril de 2019

Algoritmos y Estructuras de Datos I

### Grupo 30

Integrante	LU	Correo electrónico
Venegas, David	783/18	venegasr.david@gmail.com
Bustamante, Luis	43/18	luisbustamante097@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>

# 1. Problemas

```

proc EncuestaValida (in th:  $eph_h$ , in ti:  $eph_i$ , out result: Bool) {
  Pre {True}
  Post {result = true  $\leftrightarrow$  esEncuestaValida(th, ti)}
  pred esEncuestaValida (th:  $eph_h$ , ti:  $eph_i$ ) {
    (sonMatrices(th, ti)  $\wedge$  noSonMatricesVacias(th, ti)  $\wedge$  cantidadDeVariablesValidas(th, ti))  $\wedge_L$ 
    (hogaresEIndividuosAsociados(th, ti)  $\wedge$ 
    noHayHogaresRepetidosNiIndividuosRepetidos(th, ti)  $\wedge$ 
    añoHogarEsElMismo(th)  $\wedge$ 
    trimestreHogarEsElMismo(th)  $\wedge$ 
    añoIndividuoEsElMismo(ti)  $\wedge$ 
    trimestreIndividuoEsElMismo(ti)  $\wedge$ 
    cantidadDeMiembrosPorHogarMenorA20(ti)  $\wedge$ 
    IV2MayorOIgualII2(th)  $\wedge$ 
    atributosCategoricosEnRango(th, ti) )
  }
}

proc histHabitacional (in th:  $eph_h$ , in ti:  $eph_i$ , in region: dato, out res: seq( $\mathbb{Z}$ )) {
  Pre {esEncuestaValida(th, ti)  $\wedge$  1  $\leq$  region  $\leq$  6}
  Post {esHistogramaCorrecto(res, th, region)}
  pred esHistogramaCorrecto (hist: seq( $\mathbb{Z}$ ), th:  $eph_h$ , region: dato) {
    (esLaMayorCantidadDeHabitacionesEnRegion(th, region, |hist| - 1))  $\wedge_L$ 
    (( $\forall i : \mathbb{Z}$ )(0  $\leq$  i < |hist|  $\rightarrow_L$  #casasConNHabitaciones(th, region, i) = hist[i]))
  }
  pred esLaMayorCantidadDeHabitacionesEnRegion (th:  $eph_h$ , region: dato, max:  $\mathbb{Z}$ ) {
    ( $\exists i : \mathbb{Z}$ )(0  $\leq$  i < filas(th)  $\wedge_L$ 
    (esRegionIngresada(th, i, region)  $\wedge_L$  esCasa(th, i)  $\wedge$  th[j][@iv2] = max)  $\wedge$ 
    ( $\forall i : \mathbb{Z}$ )(0  $\leq$  i < filas(th)  $\rightarrow_L$ 
    (esRegionIngresada(th, i, region)  $\wedge_L$  esCasa(th, i)  $\rightarrow_L$  th[i][@iv2]  $\leq$  max)
    )
  }
  aux #casasConNHabitaciones (th:  $eph_h$ , region: dato, n:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  =
     $\sum_{i=0}^{filas(th)-1}$  (if (esRegionIngresada(th, i, region)  $\wedge$ 
    esCasa(th, i)  $\wedge$  esCasaConNHabitaciones(th, i, n)
    then 1 else 0 fi);
  pred esCasaConNHabitaciones (th:  $eph_h$ , fila:  $\mathbb{Z}$ , n:  $\mathbb{Z}$ ) {
    th[fila][@iv2] = n
  }
}

proc laCasaEstaQuedandoChica (in th:  $eph_h$ , in ti:  $eph_i$ , out res: seq( $\mathbb{R}$ )) {
  Pre {esEncuestaValida(th, ti)  $\wedge$  #hogaresTotalesEnRegion(th, region)! = 0}
  Post {|res| = 6  $\wedge_L$  secuenciaCorrecta(th, ti, res)}
  pred secuenciaCorrecta (th:  $eph_h$ , ti:  $eph_i$ , sec: seq( $\mathbb{R}$ )) {
    ( $\forall i : \mathbb{Z}$ )(0  $\leq$  i < |sec|  $\rightarrow_L$  sec[i] = proporcionDeHogaresHacinadosEnRegion(th, ti, i + 1))
    /*Mandamos el parametro "i+1" porque eso es lo que corresponde para @region*/
  }
  aux proporcionDeHogaresHacinadosEnRegion (th:  $eph_h$ , ti:  $eph_i$ , region: dato) :  $\mathbb{R}$  =
    #hogaresHacinadosEnRegion(th, ti, region) / #hogaresTotalesEnRegion(th, region);
  aux #hogaresTotalesEnRegion (th:  $eph_h$ , region: dato) :  $\mathbb{Z}$  =
     $\sum_{i=0}^{filas(th)-1}$  (if esCasaEnCiudadChicaEnRegionIngresada(th, i, region) then 1 else 0 fi);
  pred esCasaEnCiudadChicaEnRegionIngresada (th:  $eph_h$ , fila:  $\mathbb{Z}$ , region: dato) {
    esCasa(th, i)  $\wedge$  ( $\neg$  esCiudadGrande(th, i))  $\wedge$  esRegionIngresada(th, i, region)
  }
  aux #hogaresHacinadosEnRegion (th:  $eph_h$ , ti:  $eph_i$ , region: dato) :  $\mathbb{Z}$  =
     $\sum_{i=0}^{filas(th)-1}$  (if esHogarHacinado(th, ti, i)  $\wedge$  esCasaEnCiudadChicaEnRegionIngresada(th, i, region)
    then 1 else 0 fi);
  pred esHogarHacinado (th:  $eph_h$ , ti:  $eph_i$ , filaHogar:  $\mathbb{Z}$ ) {
    (personasPorCuarto(th, ti, filaHogar) > 3)
  }
}

```

```

aux personasPorCuarto (th: ephh, ti: ephi, filaHogar:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  =
  #individuosPorHogar(ti, th[filaHogar][@hogcodusu]) / th[filaHogar][@ii2];
aux #individuosPorHogar (ti: ephi, codigoHogar: dato) :  $\mathbb{Z}$  =
   $\sum_{i=0}^{filas(ti)-1} (\text{if } codigoHogar = ti[i][@hogcodusu] \text{ then } 1 \text{ else } 0 \text{ fi});$ 
}

proc creceElTeleworkingEnCiudadesGrandes (in t1h: ephh, in t1i: ephi, in t2h: ephh, in t2i: ephi, out res: Bool) {
  Pre {esEncuestaValida(th, ti)  $\wedge_L$  encuestasInteranualesValidas(t1h, t1i, t2h, t2i)  $\wedge$ 
    personasQueTrabajanEnCiudadesGrandes(t2h, t2i))! = 0  $\wedge$ 
    personasQueTrabajanEnCiudadesGrandes(t1h, t1i))! = 0}
  Post {res = true  $\iff$  proporcionInteranualDeTeleworking(t1h, t1i, t2h, t2i) > 0}
  pred encuestasInteranualesValidas (t1h: ephh, t1i: ephi, t2h: ephh, t2i: ephi) {
    (t1h[0][@hogaño] < t2h[0][@hogaño]  $\wedge$  t1h[0][@hogtrimestre] = t2h[0][@hogtrimestre])  $\wedge$ 
    (t1i[0][@indaño] < t2i[0][@indaño]  $\wedge$  t1i[0][@indtrimestre] = t2i[0][@indtrimestre])
  }
  aux proporcionInteranualDeTeleworking (t1h: ephh, t1i: ephi, t2h: ephh, t2i: ephi) :  $\mathbb{R}$  =
    (personasDeTeleWorkingEnCiudadesGrandes(t2h, t2i) /
    personasQueTrabajanEnCiudadesGrandes(t2h, t2i)) -
    (personasDeTeleWorkingEnCiudadesGrandes(t1h, t1i) /
    personasQueTrabajanEnCiudadesGrandes(t1h, t1i));
  aux personasQueTrabajanEnCiudadesGrandes (th: ephh, ti: ephi) :  $\mathbb{Z}$  =
     $\sum_{i=0}^{filas(ti)-1} (\text{if } (esCiudadGrande(th, filaDeHogar(th, ti, i)) \wedge personaTrabaja(ti, i)) \text{ then } 1 \text{ else } 0 \text{ fi});$ 
  pred personaTrabaja (ti: ephi, filaDePersona:  $\mathbb{Z}$ ) {
    th[filaDePersona][@estado] = 1
  }
  aux filaDeHogar (th: ephh, ti: ephi, filaDePersona:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  =
    /* Mediante esta función puedo retornar la fila del hogar de una persona mediante su Indcodusu*/
    devolverFilaDeHogcodusuSegunIndcodosu(th, ti[filaDePersona][@indcodosu]);
  aux personasDeTeleWorkingEnCiudadesGrandes (th: ephh, ti: ephi) :  $\mathbb{Z}$  =
     $\sum_{i=0}^{filas(ti)-1} (\text{if } (esPersonaDeTeleWorkingEnCiudadGrande(th, ti, i)) \text{ then } 1 \text{ else } 0 \text{ fi});$ 
  pred esPersonaDeTeleWorkingEnCiudadGrande (th: ephh, ti: ephi, filaDePersona:  $\mathbb{Z}$ ) {
    trabajaEnCasa(ti, filaDePersona)  $\wedge$  esCiudadGrande(th, filaDeHogar(th, ti, filaDePersona))  $\wedge$ 
    esHogarDeTeleWorking(th, filaDeHogar(th, ti, filaDePersona))
  }
  pred trabajaEnCasa (ti: ephi, filaDePersona:  $\mathbb{Z}$ ) {
    th[filaDePersona][@ppo4g] = 6
  }
  pred esHogarDeTeleWorking (th: ephh, ti: ephi, filaDePersona:  $\mathbb{Z}$ ) {
    esCasaODepartamento(th, filaDeHogar(th, ti, filaDePersona))  $\wedge$ 
    tieneLugarParaTrabajar(th, filaDeHogar(th, ti, filaDePersona))
  }
  pred esCasaODepartamento (th: ephh, filaDeHogar:  $\mathbb{Z}$ ) {
    th[filaDeHogar][@iv1] = 1  $\vee$  th[filaDeHogar][@iv1] = 2
  }
  pred tieneLugarParaTrabajar (th: ephh, filaDeHogar:  $\mathbb{Z}$ ) {
    th[filaDeHogar][@ii3] = 1
  }
}

proc ordenarRegionYtipo (inout th: ephh, inout ti: ephi) {
  Pre {esEncuestaValida(th, ti)  $\wedge$  (th = th0  $\wedge$  ti = ti0)}
  Post {( |th| = |th0|  $\wedge$  |ti| = |ti0|)  $\wedge_L$ 
    (matricesDeDatosIguales(th, th0)  $\wedge$  matricesDeDatosIguales(ti, ti0))  $\wedge$ 
    estanAmbasMatricesOrdenadas(th, ti)}
  pred matricesDeDatosIguales (m1, m2: seq(seq( $\mathbb{Z}$ ))) {
    matrizContenidaEn(m1, m2)  $\wedge$  matrizContenidaEn(m2, m1)
  }
  pred matrizContenidaEn (m1, m2: seq(seq( $\mathbb{Z}$ ))) {
    ( $\forall i : \mathbb{Z}$ )(0  $\leq i < filas(m1) \longrightarrow_L$ 
    ( $j : \mathbb{Z}$ )(0  $\leq j < filas(m2) \wedge_L m1[i] = m2[j]$ ))
  }
}

```

```

pred estanAmbasMatricesOrdenadas (th: ephh, ti: ephi) {
    estaMatrizHogaresOrdenada(th) ∧ estaMatrizIndividuosOrdenada(th, ti)
}

pred estaMatrizHogaresOrdenada (th: ephh) {
    estaOrdenadaPorRegion(th) ∧ estaOrdenadaPorCadaRegionDeFormaCreciente(th)
}

pred estaOrdenadaPorRegion (th: ephh) {
    (∀i : Z)(0 ≤ i < (filas(th) - 1) →L
        (th[i + 1][@region] = th[i][@region]) ∨
        (th[i + 1][@region] = th[i][@region] + 1))
}

pred estaOrdenadaPorCadaRegionDeFormaCreciente (th: ephh) {
    (∀i : Z)( (0 ≤ i < (filas(th) - 1) ∧L (th[i][@region] = th[i + 1][@region]) ) →L
        (th[i][@hogcodusu] < th[i + 1][@hogcodusu] + 1))
}

pred estaMatrizIndividuosOrdenada (th: ephh, ti: ephi) {
    estaOrdenadaPorHogcodusu(th, ti) ∧ estaOrdenadaPorComponente(th, ti)
}

pred estaOrdenadaPorHogcodusu (th: ephh, ti: ephi) {
    (∀i : Z)( (0 ≤ i < (filas(ti) - 1) →L
        (ti[i][@indcodusu] = ti[i + 1][@indcodusu]) ∨
        (ti[i][@indcodusu] = siguienteIndcodusuEnListaHogares(th, ti[i][@indcodusu])
        )
    )
}

aux siguienteIndcodusuEnListaHogares (th: ephh, incodusuBuscado: dato) : dato =
    th[ devolverFilaDeHogcodusuSegunIndcodusu(th, incodusuBuscado) + 1 ][@hogcodusu]
/*El "+1" es porque necesito el Hogcodusu siguiente al que se esta buscando*/;

pred estaOrdenadaPorComponente (th: ephh, ti: ephi) {
    (∀i : Z)( (0 ≤ i < (filas(ti) - 1) ∧L (ti[i + 1][@indcodusu] = ti[i][@indcodusu]) ) →L
        (ti[i][@componente] < ti[i + 1][@componente]) )
}

}

proc muestraHomogenea (in th: ephh, in ti: ephi, out res: seq(hogar)) {
    Pre {esEncuestaValida(th, ti)}
    Post {secuenciaPerteneceAEncuesta(res, th) ∧ noHayElementosRepetidosEnSecuencia ∧
        ordenadaDeMenorAMayorPorIngresosTotales(th, ti, res) ∧ diferenciaDosADosEsIgual(th, ti, res) ∧
        esLaSecuenciasMasLarga(th, ti, res) ∧ secuenciaEsMayorA3ElementosOEsVacía(res)
    }

    pred ordenadaDeMenorAMayorPorIngresosTotales (th: ephh, ti: ephi, seq: seq(dato)) {
        (∀i : Z)( (0 ≤ i < |seq| - 1) →L (ingresoHogar(th, ti, seq[i]) < ingresoHogar(th, ti, seq[i + 1]))
    }

    pred diferenciaDosADosEsIgual (th: ephh, ti: ephi, seq: seq(dato)) {
        (∀i : Z)(0 ≤ i < seq - 1 →L ((∀j : Z)(1 ≤ j < seq →L
            ingresoHogar(th, ti, seq[i + 1]) - ingresoHogar(th, ti, seq[i]) =
            ingresoHogar(th, ti, seq[j]) - ingresoHogar(th, ti, seq[j - 1]))
        )
    }

    aux ingresoHogar (th: ephh, ti: ephi, hogar: dato) : Z =
        ∑i=0filas(th)-1 (if (ti[i][@incodusu] = hogar ∧ ti[i][@p47t] = -1 ∧ then ti[i][@p47t] else 0 fi);

    pred secuenciaEsMayorA3ElementosOEsVacía (seq: seq(dato)) {
        |seq| > 3 ∨ seq = [ ]
    }

    pred esLaSecuenciasMasLarga (th: ephh, ti: ephi, res: seq(dato)) {
        ¬(∃seq : seq(dato))(3 ≤ |seq| ≤ filas(th)) ∧L
        ordenadaDeMenorAMayorPorIngresosTotales(th, ti, seq) ∧
        diferenciaDosADosEsIgual(th, ti, seq) →L |seq| > |res|
    }

    pred noHayElementosRepetidosEnSecuencia (sec: seq(hogar)) {
        (∀i : Z)( (0 ≤ i < |sec| →L ¬(∃i : Z)(0 ≤ i < |sec| ∧L sec[i] = sec[j]) )
    }

    pred secuenciaPerteneceAEncuesta (sec: seq(hogar), th: ephh) {
        (∀i : Z)( (0 ≤ i < |sec| →L ¬(∃i : Z)(0 ≤ i < filas(th) ∧L sec[i] = th[j]) )
    }
}

```

```

}

proc corregirRegion (inout th:  $eph_h$ , inout ti:  $eph_i$ ) {
  Pre { $esEncuestaValida(th, ti) \wedge th = th0$ }
  Post { $|th| = |th0| \wedge |th[0]| = |th0[0]| \wedge_L matrizCambiadaCorrectamente(th, th0)$ }
  pred matrizCambiadaCorrectamente (th:  $eph_h$ , th0:  $eph_h$ ) {
     $matrizDeDatosCorrectos(th, th0) \wedge cambioDeRegionCorrecto(th, th0)$ 
  }
  pred matrizDeDatosCorrectos (th:  $eph_h$ , th0:  $eph_h$ ) {
     $(\forall i : \mathbb{Z})(0 \leq i < filas(th)) \longrightarrow_L$ 
     $(\forall j : \mathbb{Z})(0 \leq j < columnas(th) \wedge_L \neg esRegionGBA(th, i, j)) \longrightarrow_L$ 
     $th[i][j] = th0[i][j])$ 
  }
  pred cambioDeRegionCorrecto (th:  $eph_h$ , th0:  $eph_h$ ) {
     $(\forall i : \mathbb{Z})(0 \leq i < filas(th0)) \wedge_L esRegionGBA(th, i, @region) \longrightarrow_L th[i][@region] = 5)$ 
  }
  pred esRegionGBA (th:  $eph_h$ , fila:  $\mathbb{Z}$ , columna:  $\mathbb{Z}$ ) {
     $columna = @region \wedge th[fila][columna] = 1$ 
  }
}

```

## 2. Predicados y Auxiliares generales

```

aux @hogcodusu :  $\mathbb{Z} = ord(HOGCODUSU)$ ;
aux @hogafio :  $\mathbb{Z} = ord(HOGAÑO)$ ;
aux @hogtrimestre :  $\mathbb{Z} = ord(HOGTRIMESTRE)$ ;
aux @ii7 :  $\mathbb{Z} = ord(II7)$ ;
aux @region :  $\mathbb{Z} = ord(REGION)$ ;
aux @mas_500 :  $\mathbb{Z} = ord(MAS_500)$ ;
aux @iv1 :  $\mathbb{Z} = ord(IV1)$ ;
aux @iv2 :  $\mathbb{Z} = ord(IV2)$ ;
aux @ii2 :  $\mathbb{Z} = ord(II2)$ ;
aux @ii3 :  $\mathbb{Z} = ord(II3)$ ;
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 @nivel_ed :  $\mathbb{Z} = ord(NIVEL_ED)$ ;
aux @estado :  $\mathbb{Z} = ord(ESTADO)$ ;
aux @cat_ocup :  $\mathbb{Z} = ord(CAT_OCUP)$ ;
aux @p47t :  $\mathbb{Z} = ord(P47T)$ ;
aux @pp04g :  $\mathbb{Z} = ord(PP04G)$ ;

pred sonMatrices (th:  $eph_h$ , ti:  $eph_i$ ) {
   $esMatriz(ti) \wedge esMatriz(th)$ 
}

pred esMatriz (m:  $seq(seq(dato))$ ) {
   $(\forall i : \mathbb{Z})(0 \leq i < |m| \longrightarrow_L ((\forall j : \mathbb{Z})(0 \leq j < |m| \longrightarrow_L |m[i]| = |m[j]|))$ 
}

pred noEsMatrizVacía (m:  $seq(seq(dato))$ ) {
   $|m| > 0$ 
}

pred noSonMatricesVacías (th:  $eph_h$ , ti:  $eph_i$ ) {
   $noEsMatrizVacía(ti) \wedge noEsMatrizVacía(th)$ 
}

pred cantidadDeVariablesValidas (th:  $eph_h$ , ti:  $eph_i$ ) {
   $(columnas(th[0]) = 10) \wedge (columnas(ti[0]) = 11)$ 
}

```

```

}
aux filas (m:seq⟨seq⟨dato⟩⟩) :  $\mathbb{Z}$  = |m|;
aux columnas (m:seq⟨seq⟨dato⟩⟩) :  $\mathbb{Z}$  = if(filas(m) > 0) then |m[0]| else 0 fi;

pred hogaresEIndividuosAsociados (th: ephh, ti: ephi) {
  (∀i :  $\mathbb{Z}$ )( (0 ≤ i < filas(th)) →L (∃j :  $\mathbb{Z}$ )( (0 ≤ j < filas(ti)) ∧L
    (th[i][@hogcodusu] = ti[j][@indcodusu])) ) ∧
  (∀i :  $\mathbb{Z}$ )( (0 ≤ i < filas(ti)) →L (∃j :  $\mathbb{Z}$ )( (0 ≤ j < filas(th)) ∧L
    (ti[i][@indodusu] = th[j][@hogcodusu])) ) )
}

pred noHayHogaresNiIndividuosRepetidos (th: ephh, ti: ephi) {
  noHayHogaresRepetidos(th) ∧ noHayIndividuosRepetidos(ti)
}

pred noHayHogaresRepetidos (th: ephh) {
  (∀i :  $\mathbb{Z}$ )( (0 ≤ i < filas(th)) →L
    ¬(∃j :  $\mathbb{Z}$ )( (i < j < filas(th)) ∧L (th[i][@hogcodusu] = th[j][@hogcodusu])) ) )
}

pred noHayIndividuosRepetidos (ti: ephi) {
  (∀i :  $\mathbb{Z}$ )( (0 ≤ i < filas(th)) →L
    ¬(∃j :  $\mathbb{Z}$ )( (i < j < filas(ti)) ∧L
      (ti[i][@indcodusu] = ti[j][@indcodusu]) ∧
      (ti[i][@componente] = ti[j][@componente])) ) )
}

pred añoHogarEsElMismo (th: ephh) {
  (∀i :  $\mathbb{Z}$ )( (0 ≤ i < filas(th) - 1) →L
    ¬(∃j :  $\mathbb{Z}$ )( (i < j < filas(th) - 1) ∧L (th[i][@hogaño]! = th[j][@hogaño])) ) )
}

pred trimestreHogarEsElMismo (th: ephh) {
  (∀i :  $\mathbb{Z}$ )( (0 ≤ i < filas(th) - 1) →L
    ¬(∃j :  $\mathbb{Z}$ )( (i < j < filas(th)) ∧L (th[i][@hogtrimestre]! = th[j][@hogtrimestre])) ) )
}

pred añoIndividuoEsElMismo (ti: ephi) {
  (∀i :  $\mathbb{Z}$ )( (0 ≤ i < filas(th) - 1) →L
    ¬(∃j :  $\mathbb{Z}$ )( (i < j < filas(ti)) ∧L (ti[i][@indaño]! = ti[j][@indaño])) ) )
}

pred trimestreIndividuoEsElMismo (ti: ephi) {
  (∀i :  $\mathbb{Z}$ )( (0 ≤ i < filas(th) - 1) →L
    ¬(∃j :  $\mathbb{Z}$ )( (i < j < filas(ti) - 1) ∧L (ti[i][@indtrimestre]! = ti[j][@indtrimestre])) ) )
}

pred cantidadDeMiembroDeHogarMenorA20 (ti: ephi) {
  (∀i :  $\mathbb{Z}$ )( (0 ≤ i < filas(ti)) →L (#apariciones(ti, @componente, ti[i][@componente]) < 20) )
}

aux #apariciones (m: seq⟨seq⟨ $\mathbb{Z}$ ⟩⟩, columna:  $\mathbb{Z}$ , elem: dato) :  $\mathbb{Z}$  =
  ∑i=0m-1 (if m[i][columna] = elem then 1 else 0 fi);

pred IV2MayorOIgualIII2 (th: ephh) {
  (∀i :  $\mathbb{Z}$ )( (0 ≤ i < filas(th)) →L th[i][@iv2] ≤ ti[i][@ii2] )
}

pred enRango (m: seq⟨seq⟨dato⟩⟩, min:  $\mathbb{Z}$ , max:  $\mathbb{Z}$ , orden:  $\mathbb{Z}$ ) {
  (∀i :  $\mathbb{Z}$ )( (0 ≤ i < filas(th)) →L (min ≤ m[i][orden] ≤ max) )
}

pred atributosCategoricosEnRango (th: ephh, thi: ephi) {
  enRango(th, 1, 6, @region) ∧ enRango(th, 1, 3, @ii2) ∧ enRango(th, 0, 1, @mas_500) ∧
  enRango(th, 1, 5, @iv1) ∧ enRango(th, 1, 2, @ii3) ∧
  enRango(ti, 1, 2, @ch4) ∧ enRango(ti, 0, 1, @nivel_ed) ∧ enRango(ti, -1, 1, @estado) ∧
  enRango(ti, 0, 4, @cat_ocup) ∧ enRango(ti, 1, 10, @ppo4g)
}

pred esCiudadGrande (th: ephh, filaDeHogar:  $\mathbb{Z}$ ) {
  th[filaDeHogar][@mas_500] = 1
}

pred esRegionIngresada (th: ephh, fila:  $\mathbb{Z}$ , region: dato) {

```

```

    th[filas][@region] = region
  }
pred esCasa (th: ephh, fila:  $\mathbb{Z}$ ) {
  th[filas][@iv1] = 1
}
aux devolverFilaDeHogcodusuSegunIndcodosu (th: ephh, indcodusuBuscado: dato) :  $\mathbb{Z}$  =
 $\sum_{i=0}^{filas(th)-1} (if (th[i][@hogcodosu] = indcodusuBuscado) then i else 0 fi);$ 

```

### 3. Decisiones tomadas

1. En Ejercicio 1: En “*pred añoHogarEsElMismo, pred trimestreHogarEsElMismo, pred añoIndividuoEsElMismo, pred trimestreIndividuoEsElMismo*” el cuantificador  $i$  se toma menor estricto que la cantidad de filas - 1 porque se compara que el valor de una posición en la matriz sea igual al valor de otra posición en la matriz dada por el cuantificador  $j$  que si es menor estricto que la cantidad de filas solamente.
2. En Ejercicio 2: Tomamos como “cantidad de habitaciones” al total de habitaciones de la casa (valor dado por @iv2) y asumimos que la primera casilla de histograma en la posición cero, cuando existen cero habitaciones puede quedar igual a cero.
3. En Ejercicio 3: Tomamos la proporción de cada región solamente en base a hogares de tipo casa y en ciudades pequeñas.
4. En Ejercicio 4: En “*aux proporcionInteranualDeTeleworking*” consideramos que la proporción está dada sobre el total de personas que trabajan en ciudades grandes, y no sobre el total de personas que trabajan en general.
5. En Ejercicio 5: En “*pred estaOrdenadaPorComponente*” suponemos que no hay componentes faltantes en un hogar.
6. En Ejercicio 6: En “*aux ingresoHogar*” suponemos que si no fue informado el ingreso total por individuo (es igual a -1), no lo contamos para el ingreso total por hogar.