



Trabajo Práctico de Especificación (TPE)

Análisis Habitacional Argentino

18 de noviembre de 2021

Algoritmos y Estructuras de Datos 1

Grupo 2

Integrante	LU	Correo electrónico
Valentina Durán	974/21	valentinad01@gmail.com
Rafael Romani	775/21	rafaromani243@gmail.com
Jonathan Bekenstein	348/11	jbekenstein@dc.uba.ar



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. Ejercicios	2
1.1. Primera parte	2
1.1.1. Ejercicio 1: esEncuestaValida	2
1.1.2. Ejercicio 2: histHabitacional	4
1.1.3. Ejercicio 3: laCasaEstaQuedandoChica	4
1.1.4. Ejercicio 4: creceElTeleworkingEnCiudadesGrandes	5
1.1.5. Ejercicio 5: costoSubsidioMejora	6
1.2. Segunda parte	7
1.2.1. Ejercicio 6: generarJoin	7
1.2.2. Ejercicio 7: ordenarRegionYTipo	7
1.2.3. Ejercicio 8: muestraHomogenea	8
1.2.4. Ejercicio 9: corregirRegion	9
1.2.5. Ejercicio 10: histogramaDeAnillosConcentricos	9
1.2.6. Ejercicio 11: quitarIndividuos	10
2. Predicados y auxiliares generales	11
2.1. Enums	11
2.2. Predicados	11
2.3. Auxiliares	14
3. Decisiones tomadas	15

1. Ejercicios

1.1. Primera parte

1.1.1. Ejercicio 1: esEncuestaValida

```
proc esEncuestaValida (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:  $eph_i$ ) {  
    tablaHogaresValida(th)  $\wedge$  tablaIndividuosValida(ti)  $\wedge_L$  tablasRelacionadasDeFormaValida(th, ti)  
}  
  
pred tablaHogaresValida (th:  $eph_h$ ) {  
    esMatrizNoVacua(th)  
     $\wedge$  tieneNCantidadDeColumnas(th, 12)  
     $\wedge_L$  sinHogaresRepetidos(th)  
     $\wedge$  suficientesHabitacionesParaDormir(th)  
     $\wedge$  todosLosAtributosCategoricosDeTodosLosHogaresEnRango(th)  
}  
  
pred tablaIndividuosValida (ti:  $eph_i$ ) {  
    esMatrizNoVacua(ti)  
     $\wedge$  tieneNCantidadDeColumnas(ti, 11)  
     $\wedge_L$  sinIndividuosRepetidos(ti)  
     $\wedge$  menosIndividuosPorHogar(ti, 20)  
     $\wedge$  todosLosAtributosCategoricosDeTodosLosIndividuosEnRango(ti)  
}  
  
pred tablasRelacionadasDeFormaValida (th:  $eph_h$ , ti:  $eph_i$ ) {  
    individuosYHogaresRelacionados(th, ti)  
     $\wedge$  todoLosRegistrosConMismoAñoYTrimestre(th, ti)  
}  
  
pred sinHogaresRepetidos (th:  $eph_h$ ) {  
     $(\forall i : \mathbb{Z})(0 \leq i < |th| \longrightarrow_L \neg(\exists j : \mathbb{Z})(0 \leq j < |th| \wedge i \neq j \wedge_L th[i][ord(HOGCODUSU)] = th[j][ord(HOGCODUSU)]))$   
}  
  
pred sinIndividuosRepetidos (ti:  $eph_i$ ) {  
     $(\forall i : \mathbb{Z})(0 \leq i < |ti| \longrightarrow_L \neg(\exists j : \mathbb{Z})($   
         $0 \leq j < |ti| \wedge i \neq j$   
         $\wedge_L ti[i][ord(INDCODUSU)] = ti[j][ord(INDCODUSU)]$   
         $\wedge ti[i][ord(COMPONENTE)] = ti[j][ord(COMPONENTE)]$   
     $))$   
}
```

```

pred todosLosRegistrosConMismoAñoYTrimestre (th:  $eph_h$ , ti:  $eph_i$ ) {
  todasLasFilasConElMismoValor(th, ord(HOGAÑO))
  ∧ todasLasFilasConElMismoValor(ti, ord(INDAÑO))
  ∧ todasLasFilasConElMismoValor(th, ord(HOGTRIMESTRE))
  ∧ todasLasFilasConElMismoValor(ti, ord(INDTRIMESTRE))
  ∧ th[0][ord(HOGAÑO)] = ti[0][ord(INDAÑO)]
  ∧ th[0][ord(HOGTRIMESTRE)] = ti[0][ord(INDTRIMESTRE)]
}

pred menosIndividuosPorHogar (ti:  $eph_i$ , cota:  $\mathbb{Z}$ ) {
  ( $\forall i : individuo$ )( $i \in ti \longrightarrow_L i[ord(COMPONENTE)] \leq cota$ )
}

pred suficientesHabitacionesParaDormir (th:  $eph_h$ ) {
  ( $\forall h : hogar$ )( $h \in th \longrightarrow_L cantidadDeHabitaciones(h) \geq cantidadDeHabitacionesParaDormir(h)$ )
}

pred todosLosAtributosCategoricosDeTodosLosHogaresEnRango (th:  $eph_h$ ) {
  ( $\forall h : hogar$ )( $h \in th \longrightarrow_L todosLosAtributosCategoricosDeUnHogarEnRango(h)$ )
}

pred todosLosAtributosCategoricosDeTodosLosIndividuosEnRango (ti:  $eph_i$ ) {
  ( $\forall i : individuo$ )( $i \in ti \longrightarrow_L todosLosAtributosCategoricosDeUnIndividuoEnRango(i)$ )
}

pred todosLosAtributosCategoricosDeUnHogarEnRango (h: hogar) {
  codigoHogarValido(h[ord(HOGCODUSU)])
  ∧ trimestreValido(h[ord(HOGTRIMESTRE)])
  ∧ regimenDeTenenciaDeHogarValido(h[ord(II7)])
  ∧ regionValido(h[ord(REGION)])
  ∧ poblacionMasDe500kValido(h[ord(MAS_500)])
  ∧ tipoDeHogarValido(h[ord(IV1)])
  ∧ cantidadDeAmbientesValido(h[ord(IV2)])
  ∧ cantidadDeAmbientesParaDormirValido(h[ord(II2)])
  ∧ utilizaAmbienteParaTrabajarValido(h[ord(II3)])
}

pred todosLosAtributosCategoricosDeUnIndividuoEnRango (i: individuo) {
  codigoHogarValido(i[ord(INDCODUSU)])
  ∧ componenteValido(i[ord(COMPONENTE)])
  ∧ trimestreValido(i[ord(INDTRIMESTRE)])
  ∧ generoValido(i[ord(CH4)])
  ∧ edadValido(i[ord(CH6)])
  ∧ estudiosUniversitariosCompletoValido(i[ord(NIVEL_ED)])
  ∧ condicionDeActividadValido(i[ord(ESTADO)])
  ∧ categoriaOcupacionalValido(i[ord(CAT_OCUP)])
  ∧ montoDeIngresoIndividualValido(i[ord(P47T)])
  ∧ lugarDondeRealizaTareasValido(i[ord(PP04G)])
}

```

1.1.2. Ejercicio 2: histHabitacional

```

proc histHabitacional (in th:  $eph_h$ , in ti:  $eph_i$ , in region:  $dato$ , out res:  $seq(\mathbb{Z})$ ) {
  Pre { $encuestaValida(th, ti)$ }
  Post {
     $hayCasasEnLaUltimaPosicionDelHistograma(res)$ 
     $\wedge cadaPosicionDelHistogramaTieneLaCantidadCorrectaDeCasas(th, region, res)$ 
     $\wedge noHayCasasConMasHabitacionesQueElMaximoDelHistograma(th, region, |res| - 1)$ 
  }
}

pred hayCasasEnLaUltimaPosicionDelHistograma (hist:  $seq(\mathbb{Z})$ ) {
   $|hist| > 0 \longrightarrow_L hist[|hist| - 1] \geq 1$ 
}

pred cadaPosicionDelHistogramaTieneLaCantidadCorrectaDeCasas (th:  $eph_h$ , region:  $dato$ , res:  $seq(\mathbb{Z})$ ) {
   $(\forall n : \mathbb{Z})(0 \leq n < |res| \longrightarrow_L res[n] = cantidadDeCasasEnLaRegionConNHabitaciones(th, region, n))$ 
}

pred noHayCasasConMasHabitacionesQueElMaximoDelHistograma (th:  $eph_h$ , region:  $dato$ , maxHabitaciones:  $\mathbb{Z}$ ) {
   $\neg(\exists h : hogar)(h \in th \wedge_L estaEnLaRegion(h, region) \wedge cantidadDeHabitaciones(h) > maxHabitaciones)$ 
}

aux cantidadDeCasasEnLaRegionConNHabitaciones (th:  $eph_h$ , region:  $dato$ , n:  $\mathbb{Z}$ ) :  $\mathbb{Z} =$ 
   $\#\{h \in th : esCasaEnLaRegionConNHabitaciones(h, region, n)\};$ 

pred esCasaEnLaRegionConNHabitaciones (h:  $hogar$ , region:  $dato$ , n:  $\mathbb{Z}$ ) {
   $esCasa(h)$ 
   $\wedge estaEnLaRegion(h, region)$ 
   $\wedge cantidadDeHabitaciones(h) = n$ 
}

```

1.1.3. Ejercicio 3: laCasaEstaQuedandoChica

```

proc laCasaEstaQuedandoChica (in th:  $eph_h$ , in ti:  $eph_i$ , out res:  $seq(\mathbb{Z})$ ) {
  Pre { $encuestaValida(th, ti)$ }
  Post {
     $|res| = 6$ 
     $\wedge (\forall i : \mathbb{Z})(0 \leq i < |res| \longrightarrow_L res[i] = proporcionDeCasasConHacinamientoCritico(th, ti, i + 1))$ 
  }
}

```

```

aux proporcionDeCasasConHacinamientoCritico (th:  $eph_h$ , ti:  $eph_i$ , region:  $dato$ ) :  $\mathbb{R}$  =
  if cantidadDeCasasDePuebloEnLaRegion(th, region) = 0 then
    0
  else
     $\frac{cantidadDeCasasDePuebloEnLaRegionConHacinamientoCritico(th, ti, region)}{cantidadDeCasasDePuebloEnLaRegion(th, region)}$ 
  fi ;

aux cantidadDeCasasDePuebloEnLaRegionConHacinamientoCritico (th:  $eph_h$ , ti:  $eph_i$ , region:  $dato$ ) :  $\mathbb{Z}$  =
   $\#\{h \in th : esCasaDePuebloEnLaRegionConHacinamientoCritico(ti, h, region)\}$  ;

aux cantidadDeCasasDePuebloEnLaRegion (th:  $eph_h$ , region:  $dato$ ) :  $\mathbb{Z}$  =
   $\#\{h \in th : esCasaDePuebloEnLaRegion(h, region)\}$  ;

pred esCasaDePuebloEnLaRegionConHacinamientoCritico (ti:  $eph_i$ , h:  $hogar$ , region:  $dato$ ) {
   $esCasaDePuebloEnLaRegion(h, region) \wedge conHacinamientoCritico(h, ti)$ 
}

pred esCasaDePuebloEnLaRegion (h:  $hogar$ , region:  $dato$ ) {
   $esCasa(h) \wedge poblacionMenosDe500k(h) \wedge estaEnLaRegion(h, region)$ 
}

pred conHacinamientoCritico (h:  $hogar$ , ti:  $eph_i$ ) {
   $cantidadDeHabitacionesParaDormir(h) = 0 \vee 3 < \frac{cantidadDeConvivientes(h, ti)}{cantidadDeHabitacionesParaDormir(h)}$ 
}

```

1.1.4. Ejercicio 4: creceElTeleworkingEnCiudadesGrandes

```

proc creceElTeleworkingEnCiudadesGrandes (in t1h:  $eph_h$ , in t1i:  $eph_i$ , in t2h:  $eph_h$ , in t2i:  $eph_i$ , out res: Bool) {
  Pre {
     $encuestaValida(t1h, t1i) \wedge encuestaValida(t2h, t2i)$ 
     $\wedge_L primerEncuestaEsAnteriorALaSegunda(t1h, t2h)$ 
     $\wedge sonEncuestasDelMismoTrimestre(t1h, t2h)$ 
  }
  Post {  $res = true \leftrightarrow proporcionTeleworking(t2h, t2i) > proporcionTeleworking(t1h, t1i)$  }
}

pred primerEncuestaEsAnteriorALaSegunda (t1h:  $eph_h$ , t2h:  $eph_h$ ) {
   $t1h[0][ord(HOGAÑO)] < t2h[0][ord(HOGAÑO)]$ 
}

pred sonEncuestasDelMismoTrimestre (t1h:  $eph_h$ , t2h:  $eph_h$ ) {
   $t1h[0][ord(HOGTRIMESTRE)] = t2h[0][ord(HOGTRIMESTRE)]$ 
}

```

```

aux proporcionTeleworking (th:  $eph_h$ , ti:  $eph_i$ ) :  $\mathbb{R}$  =
  if cantidadTotalValidoParaTeleworking(th, ti) = 0 then
    0
  else
     $\frac{cantidadTeleworking(th, ti)}{cantidadTotalValidoParaTeleworking(th, ti)}$ 
  fi ;

aux cantidadTeleworking (th:  $eph_h$ , ti:  $eph_i$ ) :  $\mathbb{Z}$  = # {
   $i \in ti : tieneTrabajo(i) \wedge tieneHogarValidoParaTeleworking(th, i[ord(INDCODUSU)])$ 
   $\wedge tieneLugarDeTrabajo(th, i[ord(INDCODUSU)])$ 
   $\wedge trabajaEnElHogar(i)$ 
};

aux cantidadTotalValidoParaTeleworking (th:  $eph_h$ , ti:  $eph_i$ ) :  $\mathbb{Z}$  =
  # {  $i \in ti : tieneTrabajo(i) \wedge tieneHogarValidoParaTeleworking(th, i[ord(INDCODUSO)])$  } ;

pred tieneTrabajo (i: individuo) {
   $i[ord(ESTADO)] = 1$ 
}

pred tieneHogarValidoParaTeleworking (th:  $eph_h$ , codigoHogar: dato) {
   $(\exists h : hogar)(h \in th \wedge_L h[ord(HOGCODUSU)] = codigoHogar \wedge poblacionMasDe500k(h) \wedge (esCasa(h) \vee esDepto(h)))$ 
}

pred tieneLugarDeTrabajo (th:  $eph_h$ , codigoHogar: dato) {
   $(\exists h : hogar)(h \in th \wedge_L h[ord(HOGCODUSU)] = codigoHogar \wedge h[ord(II3)] = 1)$ 
}

pred trabajaEnElHogar (i: individuo) {
   $i[ord(PP04G)] = 6$ 
}

```

1.1.5. Ejercicio 5: costoSubsidioMejora

```

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 {  $res = monto * cantidadDeHogaresSubsidiados(th, ti)$  }
}

aux cantidadDeHogaresSubsidiados (th:  $eph_h$ , ti:  $eph_i$ ) :  $\mathbb{Z}$  = # {  $h \in th : cumpleRequisitosParaSubsidio(h, ti)$  } ;

pred cumpleRequisitosParaSubsidio (h: hogar, ti:  $eph_i$ ) {
   $esCasa(h) \wedge tenenciaPropia(h) \wedge cantidadDeHabitacionesParaDormir(h) < cantidadDeConvivientes(h, ti) - 2$ 
}

```

1.2. Segunda parte

1.2.1. Ejercicio 6: generarJoin

```
proc generarJoin (in th:  $eph_h$ , in ti:  $eph_i$ , out junta:  $joinHI$ ) {  
  Pre { $encuestaValida(th, ti)$ }  
  Post { $|junta| = |ti| \wedge (\forall i : individuo)(i \in ti \longrightarrow_L existeTuplaHogarIndividuo(i, th, junta))$ }  
}  
  
pred existeTuplaHogarIndividuo (i:  $individuo$ , th:  $eph_h$ , junta:  $joinHI$ ) {  
   $(\exists tupla : hogar \times individuo)(tupla \in junta \wedge (tupla)_0 \in th \wedge (tupla)_1 = i \wedge_L viveEnElHogar((tupla)_0, (tupla)_1))$   
}
```

1.2.2. Ejercicio 7: ordenarRegionYTipo

```
proc ordenarRegionYTipo (inout th:  $eph_h$ , inout ti:  $eph_i$ ) {  
  Pre { $encuestaValida(th, ti) \wedge th = TH_0 \wedge ti = TI_0$ }  
  Post {  
     $esPermutacion(th, TH_0)$   
     $\wedge esPermutacion(ti, TI_0)$   
     $\wedge losHogaresEstanOrdenados(th)$   
     $\wedge losIndividuosEstanOrdenados(th, ti)$   
  }  
}  
  
pred losHogaresEstanOrdenados (th:  $eph_h$ ) {  
   $(\forall i : \mathbb{Z})(0 \leq i < |th| - 1 \longrightarrow_L$   
     $ordenadoPorRegion(th[i], th[i + 1])$   
     $\vee ordenadoPorCodigoEnLaMismaRegion(th[i], th[i + 1])$   
  )  
}  
  
pred losIndividuosEstanOrdenados (th:  $eph_h$ , ti:  $eph_i$ ) {  
   $(\forall i : \mathbb{Z})(0 \leq i < |ti| - 1 \longrightarrow_L$   
     $ordenadoPorCodigoHogar(th, ti[i], ti[i + 1])$   
     $\vee ordenadoPorComponenteEnElMismoHogar(ti[i], ti[i + 1])$   
  )  
}  
  
pred ordenadoPorRegion (h1:  $hogar$ , h2:  $hogar$ ) {  
   $h1[ord(REGION)] < h2[ord(REGION)]$   
}  
  
pred ordenadoPorCodigoEnLaMismaRegion (h1:  $hogar$ , h2:  $hogar$ ) {  
   $mismaRegion(h1, h2) \wedge h1[ord(HOGCODUSU)] < h2[ord(HOGCODUSU)]$   
}
```



```

pred ordenadoPorCodigoHogar (th:  $eph_h$ , i1:  $individuo$ , i2:  $individuo$ ) {
   $|th| = 1 \vee (\exists j_1, j_2 : \mathbb{Z})(0 \leq j_1 < j_2 < |th| \wedge_L viveEnElHogar(th[j_1], i1) \wedge viveEnElHogar(th[j_2], i2))$ 
}

pred ordenadoPorComponenteEnElMismoHogar (i1:  $individuo$ , i2:  $individuo$ ) {
   $mismoHogar(i1, i2) \wedge i1[ord(COMPONENTE)] < i2[ord(COMPONENTE)]$ 
}

```

1.2.3. Ejercicio 8: muestraHomogenea

```

proc muestraHomogenea (in th:  $eph_h$ , in ti:  $eph_i$ , out res:  $seq\langle hogar \rangle$ ) {
  Pre {  $encuestaValida(th, ti)$  }
  Post {
     $(|res| \geq 3 \wedge esMuestraHomogenea(th, ti, res) \wedge esLaMuestraHomogeneaMasGrande(th, ti, res))$ 
     $\vee (|res| = 0 \wedge \neg(\exists muestra : seq\langle hogar \rangle)(esMuestraHomogenea(th, ti, muestra) \wedge |muestra| \geq 3))$ 
  }
}

pred esMuestraHomogenea (th:  $eph_h$ , ti:  $eph_i$ , muestra:  $seq\langle hogar \rangle$ ) {
   $estaIncluida(th, muestra)$ 
   $\wedge_L ordenadoPorIngresos(ti, muestra)$ 
   $\wedge diferenciaDeIngresosConstante(ti, muestra)$ 
}

pred esLaMuestraHomogeneaMasGrande (th:  $eph_h$ , ti:  $eph_i$ , muestra:  $seq\langle hogar \rangle$ ) {
   $\neg(\exists otraMuestra : seq\langle hogar \rangle)(esMuestraHomogenea(otraMuestra) \wedge |otraMuestra| > |muestra|)$ 
}

pred ordenadoPorIngresos (ti:  $eph_i$ , hogares:  $seq\langle hogar \rangle$ ) {
   $(\forall i : \mathbb{Z})(0 \leq i < |hogares| - 1 \longrightarrow_L ingresosDelHogar(ti, hogares[i]) \leq ingresosDelHogar(ti, hogares[i + 1]))$ 
}

pred diferenciaDeIngresosConstante (ti:  $eph_i$ , hogares:  $seq\langle hogar \rangle$ ) {
   $(\forall i : \mathbb{Z})(1 \leq i < |hogares| - 1 \longrightarrow_L$ 
     $diferenciaDeIngresosConElSiguiente(ti, hogares, i) = diferenciaDeIngresosConElSiguiente(ti, hogares, 0)$ 
  )
}

aux ingresosDelHogar (ti:  $eph_i$ , hogar:  $hogar$ ) :  $\mathbb{Z}$  =
   $\sum_{i=0}^{|ti|-1} \text{if } viveEnElHogar(hogar, ti[i]) \wedge ingresos(ti[i]) > 0 \text{ then } ingresos(ti[i]) \text{ else } 0 \text{ fi};$ 

aux diferenciaDeIngresosConElSiguiente (ti:  $eph_i$ , hogares:  $seq\langle hogar \rangle$ , i:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  =
   $ingresosDelHogar(ti, hogares[i + 1]) - ingresosDelHogar(ti, hogares[i]);$ 

```

1.2.4. Ejercicio 9: corregirRegion

```

proc corregirRegion (inout th:  $eph_h$ , in ti:  $eph_i$ ) {
  Pre { $encuestaValida(th, ti) \wedge th = TH_0$ }
  Post {
     $|th| = |TH_0|$ 
     $\wedge (\forall h : \text{hogar})(h \in TH_0 \longrightarrow_L \text{hogarDeGranBuenosAiresAPampeana}(th, h) \vee \text{hogarSinModificacion}(th, h))$ 
     $\wedge \text{losHogaresMantienenElMismoOrden}(th, TH_0)$ 
  }
}

pred hogarDeGranBuenosAiresAPampeana (th:  $eph_h$ , h:  $hogar$ ) {
   $estaEnLaRegion(h, ord(GRAN\_BUENOS\_AIRES)) \wedge setAt(h, ord(REGION), ord(PAMPEANA)) \in th$ 
}

pred hogarSinModificacion (th:  $eph_h$ , h:  $hogar$ ) {
   $\neg estaEnLaRegion(h, ord(GRAN\_BUENOS\_AIRES)) \wedge h \in th$ 
}

pred losHogaresMantienenElMismoOrden (th1:  $eph_h$ , th2:  $eph_h$ ) {
   $(\forall i : \mathbb{Z})(0 \leq i < |th1| \longrightarrow_L th1[i][ord(HOGCODUSU)] = th2[i][ord(HOGCODUSU)])$ 
}

```

1.2.5. Ejercicio 10: histogramaDeAnillosConcentricos

```

proc histogramaDeAnillosConcentricos (in th:  $eph_h$ , in centro:  $\mathbb{Z} \times \mathbb{Z}$ , in distancias:  $seq(\mathbb{Z})$ , out result:  $seq(\mathbb{Z})$ ) {
  Pre { $tablaHogaresValida(th) \wedge |distancias| > 0 \wedge esEstrictamenteCreciente(distancias)$ }
  Post {
     $|res| = |distancias|$ 
     $\wedge_L res[0] = cantidadDeCasasDentroDeLaCircunferencia(th, centro, distancias[0])$ 
     $\wedge (\forall i : \mathbb{Z})(1 \leq i < |distancias| \longrightarrow_L res[i] = cantidadDeCasasEnElAnillo(th, centro, distancias[i-1], distancias[i]))$ 
  }
}

aux cantidadDeCasasEnElAnillo (th:  $eph_h$ , centro:  $\mathbb{Z} \times \mathbb{Z}$ , radioInterno:  $\mathbb{Z}$ , radioExterno:  $\mathbb{Z}$ ) :  $\mathbb{Z} =$ 
   $cantidadDeCasasDentroDeLaCircunferencia(th, centro, radioExterno)$ 
   $- cantidadDeCasasDentroDeLaCircunferencia(th, centro, radioInterno);$ 

aux cantidadDeCasasDentroDeLaCircunferencia (th:  $eph_h$ , centro:  $\mathbb{Z} \times \mathbb{Z}$ , radio:  $\mathbb{Z}$ ) :  $\mathbb{Z} =$ 
   $\#\{h \in th : distanciaEntrePuntos((centro)_0, (centro)_1, h[ord(HOGLATITUD)], h[ord(HOGLONGITUD)]) \leq cuadrado(radio)\};$ 

aux distanciaEntrePuntos (x1:  $\mathbb{Z}$ , y1:  $\mathbb{Z}$ , x2:  $\mathbb{Z}$ , y2:  $\mathbb{Z}$ ) :  $\mathbb{Z} = cuadrado(x1 - x2) + cuadrado(y1 - y2);$ 

```

1.2.6. Ejercicio 11: quitarIndividuos

```

proc quitarIndividuos (inout th:  $eph_h$ , inout ti:  $eph_i$ , in busqueda:  $seq\langle (ItemIndividuo, dato) \rangle$ , out result:  $(eph_h, eph_i)$ ) {
  Pre {
    encuestaValida(th, ti)
     $\wedge$  busquedaValida(busqueda)
     $\wedge$  th =  $TH_0$ 
     $\wedge$  ti =  $TI_0$ 
  }
  Post {
    losIndividuosEncontradosEstanEnLaSalida( $TI_0$ , busqueda, (result)1)
     $\wedge$  individuosY HogaresRelacionados((result)0, (result)1)
     $\wedge$  noFaltanNiSobranElementosRespectoALaEntrada( $TH_0$ , th, (result)0)
     $\wedge$  noFaltanNiSobranElementosRespectoALaEntrada( $TI_0$ , ti, (result)1)
  }
}

pred busquedaValida (busqueda:  $seq\langle (ItemIndividuo, dato) \rangle$ ) {
  todosLosValoresBuscadosSonValidos(busqueda)  $\wedge$  sinItemIndividuoRepetidos(busqueda)
}

pred todosLosValoresBuscadosSonValidos (busqueda:  $seq\langle (ItemIndividuo, dato) \rangle$ ) {
   $(\forall i : \mathbb{Z})(0 \leq i < |busqueda| \longrightarrow_L \text{valorValidoParaElItemAsociado}((busqueda[i])_0, (busqueda[i])_1))$ 
}

pred sinItemIndividuoRepetidos (busqueda:  $seq\langle (ItemIndividuo, dato) \rangle$ ) {
   $(\forall i : \mathbb{Z})(0 \leq i < |busqueda| \longrightarrow_L \neg(\exists j : \mathbb{Z})(0 \leq j < |busqueda| \wedge i \neq j \wedge_L (busqueda[i])_0 = (busqueda[j])_0))$ 
}

pred valorValidoParaElItemAsociado (item: ItemIndividuo, valor: dato) {
  (item = INDCODUSU  $\wedge$  codigoHogarValido(valor))
   $\vee$  (item = COMPONENTE  $\wedge$  componenteValido(valor))
   $\vee$  (item = INDTRIMESTRE  $\wedge$  trimestreValido(valor))
   $\vee$  (item = CH4  $\wedge$  generoValido(valor))
   $\vee$  (item = CH6  $\wedge$  edadValido(valor))
   $\vee$  (item = NIVEL_ED  $\wedge$  estudiosUniversitariosCompletosValido(valor))
   $\vee$  (item = ESTADO  $\wedge$  condicionDeActividadValido(valor))
   $\vee$  (item = CAT_OCUP  $\wedge$  categoriaOcupacionalValido(valor))
   $\vee$  (item = P47T  $\wedge$  montoDeIngresoIndividualValido(valor))
   $\vee$  (item = PP04G  $\wedge$  lugarDondeRealizaTareasValido(valor))
}

pred losIndividuosEncontradosEstanEnLaSalida (ti:  $eph_i$ , busqueda:  $seq\langle (ItemIndividuo, dato) \rangle$ , result:  $eph_i$ ) {
   $(\forall i : individuo)(i \in ti \wedge_L \text{coincideConLaBusqueda}(i, busqueda) \leftrightarrow i \in result)$ 
}

pred noFaltanNiSobranElementosRespectoALaEntrada (entrada:  $seq\langle T \rangle$ , salida:  $seq\langle T \rangle$ , resultados:  $seq\langle T \rangle$ ) {
  esPermutacion(entrada, salida ++ resultados)
}

```

```

pred coincideConLaBusqueda (individuo: individuo, busqueda: seq((ItemIndividuo, dato))) {
  ( $\forall i : \mathbb{Z}$ )( $0 \leq i < |busqueda| \longrightarrow_L coincideConUnItemDeLaBusqueda(individuo, (busqueda[i])_0, (busqueda[i])_1)$ )
}

pred coincideConUnItemDeLaBusqueda (individuo: individuo, item: ItemIndividuo, valor: dato) {
  individuo[ord(item)] = valor
}

```

2. Predicados y auxiliares generales

2.1. Enums

```

enum ItemHogar {
  HOGCODUSU, HOGAÑO, HOGTRIMESTRE, HOGLATITUD, HOGLONGITUD, II7, REGION, MAS_500, IV1, IV2, II2, II3
}

enum ItemIndividuo {
  INDCODUSU, COMPONENTE, INDAÑO, INDTRIMESTRE, CH4, CH6, NIVEL_ED, ESTADO, CAT_OCUP, P47T, PP04G
}

enum TiposDeHogar {
  SIN_DATO, CASA, DEPARTAMENTO, PIEZA_DE_INQUILINATO, PIEZA_EN_HOTEL, LOCAL
}

enum Regiones {
  SIN_DATO, GRAN_BUENOS_AIRES, NOA, NEA, CUYO, PAMPEANA, PATAGONIA
}

```

2.2. Predicados

```

pred cantidadDeAmbientesParaDormirValido (valor: dato) {
   $0 \leq valor$ 
}

pred cantidadDeAmbientesValido (valor: dato) {
   $1 \leq valor$ 
}

pred categoriaOcupacionalValido (valor: dato) {
   $0 \leq valor \leq 4$ 
}

pred codigoHogarValido (valor: dato) {
   $1 \leq valor$ 
}

```

```

pred componenteValido (valor: dato) {
   $1 \leq valor$ 
}

pred condicionDeActividadValido (valor: dato) {
   $-1 \leq valor \leq 1$ 
}

pred edadValido (valor: dato) {
   $0 \leq valor$ 
}

pred esCasa (h: hogar) {
   $h[ord(IV1)] = ord(CASA)$ 
}

pred esDepto (h: hogar) {
   $h[ord(IV1)] = ord(DEPARTAMENTO)$ 
}

pred esEstrictamenteCreciente (s: seq⟨Z⟩) {
   $(\forall i : \mathbb{Z})(0 \leq i < |s| - 1 \longrightarrow_L s[i] < s[i + 1])$ 
}

pred esMatriz (m: seq⟨seq⟨Z⟩⟩) {
   $(\forall i : \mathbb{Z})(0 \leq i < |m| \longrightarrow_L |m[i]| > 0 \wedge |m[i]| = |m[0]|)$ 
}

pred esMatrizNoVacía (m: seq⟨seq⟨Z⟩⟩) {
   $esMatriz(m) \wedge |m| > 0$ 
}

pred esPermutacion (l1: seq⟨T⟩, l2: seq⟨T⟩) {
   $(\forall e : T)(cantidadDeApariciones(l1, e) = cantidadDeApariciones(l2, e))$ 
}

pred estaEnLaRegion (h: hogar, region: dato) {
   $h[ord(REGION)] = region$ 
}

pred estaIncluida (s: seq⟨T⟩, sub: seq⟨T⟩) {
   $(\forall e : T)(e \in sub \longrightarrow_L cantidadDeApariciones(sub, e) \leq cantidadDeApariciones(s, e))$ 
}

pred estudiosUniversitariosCompletosValido (valor: dato) {
   $0 \leq valor \leq 1$ 
}

```

```

pred existeFilaConUnValor (m: seq<seq<Z>>, columna: Z, valor: Z) {
  ( $\exists i : Z$ )( $0 \leq i < |m| \wedge_L m[i][columna] = valor$ )
}

pred generoValido (valor: dato) {
   $1 \leq valor \leq 2$ 
}

pred individuosYHogaresRelacionados (th: ephh, ti: ephi) {
  ( $\forall h : hogar$ )( $h \in th \rightarrow_L existeFilaConUnValor(ti, ord(INDCODUSU), h[ord(HOGCODUSU)])$ )
   $\wedge (\forall i : individuo)((i \in ti \rightarrow_L existeFilaConUnValor(th, ord(HOGCODUSU), i[ord(INDCODUSU)]))$ )
}

pred lugarDondeRealizaTareasValido (valor: dato) {
   $1 \leq valor \leq 10$ 
}

pred mismaRegion (h1: hogar, h2: hogar) {
   $h1[ord(REGION)] = h2[ord(REGION)]$ 
}

pred mismoHogar (i1: individuo, i2: individuo) {
   $i1[ord(INDCODUSU)] = i2[ord(INDCODUSU)]$ 
}

pred montoDeIngresoIndividualValido (valor: dato) {
   $-1 \leq valor$ 
}

pred poblacionMasDe500k (h: hogar) {
   $h[ord(MAS_500)] = 1$ 
}

pred poblacionMasDe500kValido (valor: dato) {
   $0 \leq valor \leq 1$ 
}

pred poblacionMenosDe500k (h: hogar) {
   $h[ord(MAS_500)] = 0$ 
}

pred regimenDeTenenciaDeHogarValido (valor: dato) {
   $1 \leq valor \leq 3$ 
}

```

```

pred regionValido (valor: dato) {
  1 ≤ valor ≤ 6
}

pred tenenciaPropia (h: hogar) {
  h[ord(II7)] = 1
}

pred tieneNCantidadDeColumnas (m: seq⟨seq⟨Z⟩⟩, n: Z) {
  (∀i : Z)(0 ≤ i < |m| →L |m[i]| = n)
}

pred tipoDeHogarValido (valor: dato) {
  1 ≤ valor ≤ 5
}

pred trimestreValido (valor: dato) {
  1 ≤ valor ≤ 4
}

pred todasLasFilasConElMismoValor (m: seq⟨seq⟨Z⟩⟩, columna: Z) {
  (∀i : Z)(0 ≤ i < |m| - 1 →L m[i][columna] = m[i + 1][columna])
}

pred utilizaAmbienteParaTrabajarValido (valor: dato) {
  1 ≤ valor ≤ 2
}

pred viveEnElHogar (h: hogar, i: individuo) {
  i[ord(INDCODUSU)] = h[ord(HOGCODUSU)]
}

```

2.3. Auxiliares

```

aux cantidadDeApariciones (l: seq⟨T⟩, e: T) : Z = #{x ∈ l : x = e};

aux cantidadDeConvivientes (h: hogar, ti: ephi) : Z = #{i ∈ ti : viveEnElHogar(h, i)};

aux cantidadDeHabitaciones (h: hogar) : Z = h[ord(IV2)];

aux cantidadDeHabitacionesParaDormir (h: hogar) : Z = h[ord(II2)];

aux cuadrado (x: Z) : Z = x * x;

aux ingresos (i: individuo) : Z = i[ord(P47T)];

```

3. Decisiones tomadas

- En ningún ejercicio se valida el año de los hogares o de los individuos porque no hay un criterio claro para hacerlo.
- A falta de información se asumió que el componente de un individuo pertenece a los naturales.
- La edad de los individuos se consideró desde 0 para tener en cuenta a los recién nacidos.
- Para calcular la proporción de teleworking se consideró como individuos totales a los que viven en casas o departamentos localizados en ciudades de más de 500 mil habitantes. Y para contabilizar a los que realizan teleworking, se consideró a los individuos que realizan principalmente sus tareas en su hogar y además tienen una habitación exclusivamente para trabajar.
- En el ejercicio 10 se consideró como dentro del anillo a los hogares ubicados sobre el borde exterior del mismo.
- En el ejercicio 11 no se garantiza que la encuesta siga siendo válida luego de ejecutar el programa, pues al quitar los hogares de los individuos encontrados, existe la posibilidad de que hayan quedado individuos de ese mismo hogar en la tabla de individuos. La consigna no indica qué hacer en esta situación, y especificar que la encuesta siga siendo válida implicaría quitar individuos de la tabla de individuos que no cumplan el criterio de búsqueda, o duplicar los hogares.