



TP de Especificación

Análisis Habitacional Argentino

8 de Septiembre de 2021

Algoritmos y Estructuras de Datos I

Grupo 01

Integrante	LU	Correo electrónico
Freire, Guido	978/21	gfreire@dc.uba.ar
Hernandez, Agustin	37/20	ahernandez@dc.uba.ar
Machulsky, Joaquin	521/21	jmachulsky@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>

1. Problemas

1.1. Ejercicio 1

```

proc esEncuestaVálida (in th : ephh, in ti : ephi, out result : Bool) {
  Pre {true}
  Post {
    esMatriz(th) ∧ (∀h : ℤ)(rango(th, h) →L |th[h]| = 12) ∧
    esMatriz(ti) ∧ (∀i : ℤ)(rango(ti, i) →L |ti[i]| = 11) ∧
    valoresValidosHogares(th) ∧ valoresValidosIndividuos(ti) ∧
    individuosConHogaresYHogaresConIndividuos(th, ti) ∧
    (∀h : ℤ)(cantidadDeAparicionesHogar(h, ti) ≤ 20) ∧ componenteValido(ti) ∧
    noHayHogarRepetido(th) ∧ noHayIndividuoRepetido(ti) ∧
    masAmbientesQueHabitacionesParaMimir(th) ∧
    periodoValido(th, ti)}

  pred valoresValidosHogares (th: ephh){
    (∀h : ℤ)(rango(th, h) →L
      th[h][@HOGCODUSU] > 0 ∧ th[h][@HOGAÑO] > 0 ∧ 1 ≤ th[h][@HOGTRIMESTRE] ≤ 4 ∧
      1 ≤ th[h][@II7] ≤ 3 ∧ 1 ≤ th[h][@REGION] ≤ 6 ∧ 0 ≤ th[h][@MAS_500] ≤ 1 ∧
      1 ≤ th[h][@IV1] ≤ 5 ∧ th[h][@IV2] ≥ 0 ∧ th[h][@II2] ≥ 0 ∧ 1 ≤ th[h][@II3] ≤ 2)}

  pred valoresValidosIndividuos (ti: ephi){
    (∀i : ℤ)(rango(ti, i) →L
      ti[i][@INDCODUSU] > 0 ∧ ti[i][@COMPONENTE] > 0 ∧
      ti[i][@INDAÑO] ≥ 0 ∧ 1 ≤ ti[i][@INDTRIMESTRE] ≤ 4 ∧
      1 ≤ th[h][@CH4] ≤ 2 ∧ ti[i][@CH6] ≥ 1 ∧
      0 ≤ th[h][@NIVELED] ≤ 1 ∧ -1 ≤ th[h][@ESTADO] ≤ 1 ∧
      0 ≤ th[h][@CATOCUP] ≤ 4 ∧ th[h][@p47T] ≥ -1 ∧ 1 ≤ th[h][@PP04G] ≤ 10)}

  pred individuosConHogaresYHogaresConIndividuos (th: ephh, ti : ephi){
    (∀h : ℤ)(cantidadDeAparicionesHogar(h, th) > 0 ↔ cantidadDeAparicionesHogar(h, ti) > 0)}

  pred componenteValido (ti: ephi){
    (∀i : individuo)(i ∈ ti ∧ i[@COMPONENTE] > 1 →L
      (∃j : individuo)(j ∈ ti ∧ i[@INDCODUSU] = j[@INDCODUSU] ∧L
        i[@COMPONENTE] = j[@COMPONENTE] - 1))}

  pred masAmbientesQueHabitacionesParaMimir (th: ephh){
    / * mimir = dormir * /
    (∀h : ℤ)(rango(th, h) →L th[h][@IV2] ≥ th[h][@II2])}

  pred lugarValido (th: ephh){
    (∀h : ℤ)(rango(th, h) →L (th[h][@HOGLATITUD], th[h][@HOGLONGITUD] : ℤ))}

  pred periodoValido (th: ephh, ti : ephi){
    (∀h, i : ℤ)(rango(th, h) ∧ rango(ti, i) →L th[h][@HOGAÑO] = ti[i][@INDAÑO] ∧
      th[h][@HOGTRIMESTRE] = ti[i][@INDTRIMESTRE])}
}

```

1.2. Ejercicio 2

```

proc histHabitacional (in th: ephh, in ti : ephi, in region : ℤ, out res : seq(ℤ)){
  Pre {esEncuestaValida(th, ti) ∧ 1 ≤ region ≤ 6}
  Post {
    / * *|res| = maximo numero de habitaciones en region */
    (∃i : ℤ)(rango(th, i) ∧L (∀j : ℤ)(rango(th, j) →L
      th[j][@IV2] ≤ th[i][@IV2] ∧ th[i][@IV2] + 1 = |res|)) ∧
    (∀i : ℤ)(rango(res, i) →L res[i] = cantidadCasasConHabitacionesRegion(th, region, i))}

  aux cantidadCasasConHabitacionesRegion (th: ephh, region : ℤ, h : ℤ) : ℤ =
    ∑i=0|th|-1 if (th[i][@REGION] = region ∧ th[i][@IV1] = 1 ∧ t[i][@IV2] = h) then 1 else 0 fi;
}

```

1.3. Ejercicio 3

```

proc laCasaEstaQuedandoChica (in th: ephh, in ti: ephi, out res: seq(ℝ)){
  Pre {esEncuestaValida(th, ti)}
  Post {|res| = 6 ∧
    (∀r: ℤ)(1 ≤ r ≤ 6 →L
      hacinamientoCriticoPorRegion(th, ti, r) · inversoDe(cantidadDeAparicionesRegion(r, th)) = res[r - 1])}

  aux hacinamientoCriticoPorRegion (th: ephh, ti: ephi, region: ℤ) : ℤ =
    ∑i=0|th|-1 if th[i][@REGION] = r ∧ th[i][@MAS_500] = 0 ∧ th[i][@IV1] = 1 ∧ esHacinamientoCritico(th[i], ti)
      then 1 else 0 fi;

  pred esHacinamientoCritico (h: hogar, ti: ephi){
    hogar[@IV2] = 0 ∨L  $\frac{\text{cantidadDeAparicionesHogar}(h[\text{@HOGCODUSU}], ti)}{\text{hogar}[\text{@IV2}]} > 3$ }

  aux cantidadDeAparicionesRegion (r: ℤ, th: ephh) : ℤ =
    ∑i=0|th|-1 if (th[i][@REGION] = r ∧ th[i][@MAS_500] = 0 ∧ th[i][@IV1] = 1) then 1 else 0 fi;
}

```

1.4. Ejercicio 4

```

proc creceElTeleworkingEnCiudades (in t1h: ephi, in t1i: ephi, in t2h: ephh, in t2i: ephi, out res: Bool){
  Pre {esEncuestaValida(t1h, t1i) ∧ esEncuestaValida(t2h, t2i) ∧ (t1h[0][@HOGAÑO] < t2h[0][@HOGAÑO]) ∧
    t1i[0][@INDTRIMESTRE] = t2i[0][@INDTRIMESTRE]}
  Post {res = true ↔ proporcionTeleworkers(t1h, t1i) < proporcionTeleworkers(t2h, t2i)}

  aux individuos500Casa (th: ephh, ti: ephi) : ℤ =
    ∑i=0|th|-1 if th[i][@MAS_500] = 1 ∧ (th[i][@iIV1] = 1 ∨ th[i][@IV1] = 2)
      then cantidadDeAparicionesHogar(th[i][@HOGCODUSU], ti) else 0 fi;

  pred aptoTelework (h: dato, th: ephh){
    (∃i: ℤ)(rango(th, i) ∧L (th[i][@HOGCODUSU] = h ∧
      th[i][@MAS_500] = 1 ∧ th[i][@II3] = 1 ∧ (th[i][@IV1] = 1 ∨ th[i][@IV1] = 2)))}

  aux teleworkers (th: ephh, ti: ephi) : ℤ =
    ∑i=0|ti|-1 if ti[i][@PP04G] = 6 ∧ aptoTelework(ti[i][@INDCODUSU], th) then 1 else 0 fi;

  aux proporcionTeleworkers (in th: ephh, in ti: ephi) : ℝ =
    teleworkers(th, ti) · inversoDe(individuos500Casa(th, ti));
}

```

1.5. Ejercicio 5

```

proc costoSubsidioMejora (in th: ephh, in ti: ephi, in monto: ℤ, out res: ℤ){
  Pre {esEncuestaValida(th, ti) ∧ monto > 0}
  Post {res ≥ 0 ∧ res = monto · cantidadCasasSubsidiabiles(th, ti)}

  aux cantidadCasasSubsidiabiles (th: ephh, ti: ephi) : ℤ =
    ∑i=0|th|-1 if th[i][@II7] = 1 ∧ th[i][@IV1] = 1 ∧ (th[i][@II2] < cantidadAparicionesHogar(th[i][@HOGCODUSU], ti) - 2)
      then 1 else 0 fi;
}

```

1.6. Ejercicio 6

```

proc generarJoin (in th: ephh, in ti: ephi, out junta: joinHI){
  Pre {esEncuestaValida(th, ti)}
  Post {|junta| = |ti| ∧
    (∀j: ℤ)(rango(junta, j) →L j0 ∈ th ∧ j1 ∈ ti ∧ j0[@HOGCODUSU] = j1[@INDCODUSU]) ∧
    (∀ind: individuo)(ind ∈ ti →L (∃tupla: joinHI)(tupla ∈ junta ∧L tupla1 = individuo))}
}

```

1.7. Ejercicio 7

```

proc ordenarRegionYTipo (inout th : ephh, inout ti : ephi) {
  Pre {esEncuestaValida(th, ti) ∧ th = th0 ∧ ti = ti0}
  Post {|th| = |th0| ∧ |ti| = |ti0| ∧ hogaresOrdenados(th, th0) ∧L individuosOrdenados(ti, ti0, th, th0)}

  pred hogaresOrdenados (th: ephh, th0 : ephh) {
    (∀h1, h2 : hogar)(h1 ∈ th0 ∧ h2 ∈ th0 →L
      (h1[@REGION] < h2[@REGION] →L tieneIndiceAscendente(th, h1, h2)) ∧
      (h1[@REGION] = h2[@REGION] ∧ h1[@HOGCODUSU] < h2[@HOGCODUSU] →
        tieneIndiceAscendente(th, h1, h2)))}

  pred individuosOrdenados (ti : ephi, ti0 : ephi, th : ephh, th0 : ephh) {
    (∀i1, i2 : individuo)(i1 ∈ ti0 ∧ i2 ∈ ti0 →L
      (∃h1, h2 : ℤ)(rango(th, h1) ∧ rango(th, h2) ∧L viveEn(i1, th[h1]) ∧ viveEn(i2, th[h2]) ∧
        (h1 < h2 →L tieneIndiceAscendente(ti, i1, i2)) ∧
        (h1 = h2 ∧ i1[@COMPONENTE] < i2[@COMPONENTE] →L tieneIndiceAscendente(ti, i1, i2))))))}

  pred tieneIndiceAscendente (t: seq⟨seq⟨dato⟩⟩, e1 : seq⟨dato⟩, e2 : seq⟨dato⟩) {
    /*e1 esta ubicado antes que e2*/
    (∃i, j : ℤ)(rango(t, i) ∧ rango(t, j) ∧L t[i] = e1 ∧ t[j] = e2 ∧ i < j)

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

```

1.8. Ejercicio 8

```

proc muestraHomogenea (in th: ephh, in ti : ephi, out res : seq⟨hogar⟩) {
  Pre {esEncuestaValida(th, ti)}
  Post {(∃seq : seq⟨hogar⟩)(|seq| ≥ 3 ∧L ingresosOrdenados(seq, ti) ∧ correspondeTh(seq, th) ∧
    ¬(∃seq' : seq⟨hogar⟩)(ingresosOrdenados(seq, ti) ∧ correspondeTh(seq, th) ∧ |seq'| > |seq|) ∧ res = seq) ∨
    res = <>}}

  pred ingresosOrdenados (seq: seq⟨hogar⟩, ti: ephi) {
    (∃d : ℤ)(0 ≤ d ∧L (∀h : ℤ)(0 ≤ h < |seq| - 1 →L ingresos(seq[h + 1], ti) - ingresos(seq[h], ti) = d))}

  aux ingresos (h: hogar, ti: ephi) : ℤ =
    /*Si los ingresos son -1 suma 0*/
    ∑i=0|ti|-1 if ti[i][@INDCODUSU] = h[@HOGCODUSU] ∧ ti[i][@p47T] ≠ -1 then ti[i][@p47T] else 0 fi;

  pred correspondeTh (seq: seq⟨hogar⟩, th: ephh) {
    (∀i : ℤ)(rango(seq, i) →L
      (∃h : th)(h = seq[i] ∧ (∀i' : ℤ)(rango(seq, i') →L seq[i] = seq[i'] ↔ i = i')))}
}

```

1.9. Ejercicio 9

```

proc corregirRegion (inout th: ephh, ∈ ti : ephi) {
  Pre {esEncuestaValida(th, ti) ∧ th0 = th}
  Post {|th| = |th0| ∧ (∀i : ℤ)(rango(th0, i) →L
    (th0[i][@REGION] = 1 → th[i][@REGION] = 5) ∧
    (th0[i][@REGION] ≠ 1 → th0[i] = th[i]) ∧
    todoLoDemasIgual(th0[i], th[i]))}

  pred todoDemasIgual (h0 : hogar, h : hogar) {
    (∀j : ℤ)(rango(h0, j) ∧ j ≠ @REGION ∧L h0[j] = h[j])}
}

```

1.10. Ejercicio 10

```

proc histogramaDeAnillosConcentricos (in th: ephh, in centro:  $\mathbb{Z} \times \mathbb{Z}$ , in distancias: seq( $\mathbb{Z}$ ), result: seq( $\mathbb{Z}$ )) {
  Pre {distanciasValidas(distancias)}
  Post { |res| = |distancias|  $\wedge$  res[0] = cantidadDeCasasEntreDistancias(0, distancias[0], th)  $\wedge$ 
        ( $\forall i: \mathbb{Z}$ ) (1  $\leq i < |distancias| \rightarrow_L$  res[i] = cantidadDeCasasEntreDistancias(distancias[i-1], distancias[i], th)) }

  pred distanciasValidas (distancias: seq( $\mathbb{Z}$ )) {
    |distancias| > 0  $\wedge$  ( $\forall i: \mathbb{Z}$ ) (0  $\leq i < |distancias| - 1 \rightarrow_L$  0 < distancias[i] < distancias[i+1]) }

  aux euclideana (c:  $\mathbb{Z} \times \mathbb{Z}$ , h: hogar) :  $\mathbb{R} = \sqrt{(c_1 - h[@LATITUD])^2 + (c_2 - h[@LONGITUD])^2}$ ;

  aux cantidadDeCasasEntreDistancias (d0:  $\mathbb{Z}$ , d1:  $\mathbb{Z}$ , th: ephh) :  $\mathbb{Z} =$ 
     $\sum_{i=0}^{|th|-1}$  if d0 < euclideana(c, th[i])  $\leq$  d1 then 1 else 0 fi;
}

```

1.11. Ejercicio 11

```

proc quitarIndividuos (inout th: ephh, inout ti: ephi, in busqueda: seq((ItemIndividuo, dato)), result: (ephh, ephi)) {
  Pre { encuestaValida(th, ti)  $\wedge$  busquedaValida(busqueda)  $\wedge$  th = th0  $\wedge$  ti = ti0 }
  Post { ( $\forall$  tupla: seq((ItemIndividuo, dato))) (
    ( $\forall i: individuo$ ) (
      i  $\in$  ti0  $\wedge$  i[ord(tupla0)] = tupla1  $\rightarrow$  i  $\in$  result1  $\wedge$ 
      i  $\in$  ti0  $\wedge$  i[ord(tupla0)]  $\neq$  tupla1  $\rightarrow$  i  $\in$  ti  $\wedge$ 
      i  $\notin$  ti0  $\rightarrow$  (i  $\notin$  result1  $\wedge$  i  $\notin$  ti)
    )  $\wedge_L$  ( $\forall h: hogar$ ) (
      h  $\in$  th0  $\wedge$  ( $\exists i: result$ ) (i[@INDCODUSU] = h[@HOGCODUSU])  $\rightarrow$  h  $\in$  result  $\wedge$ 
      h  $\in$  th0  $\wedge$  ( $\nexists i: result$ ) (i[@INDCODUSU] = h[@HOGCODUSU])  $\rightarrow$  h  $\in$  th  $\wedge$ 
      h  $\notin$  th0  $\rightarrow$  (h  $\notin$  result0  $\wedge$  h  $\notin$  th)
    )  $\wedge$ 
    /**Si bien el comportamiento de noHayHogar/IndividuoRepetido es distinto, logran el mismo efecto**/
    noHayIndividuoRepetido(result1)  $\wedge$  noHayHogarRepetido(result0)
  ) }

  pred busquedaValida (s: seq((ItemIndividuo, dato))) {
    ( $\forall i, j: \mathbb{Z}$ ) (rango(s, i)  $\wedge$  rango(s, j)  $\wedge$  i  $\neq$  j  $\rightarrow_L$  s[i]0  $\neq$  s[j]0)  $\wedge$ 
    ( $\forall i: \mathbb{Z}$ ) (rango(s, i)  $\rightarrow_L$ 
      (s[i]0 = INDCODUSU  $\wedge$  s[i]1 > 0)  $\vee$ 
      (s[i]0 = COMPONENTE  $\wedge$  s[i]1 > 0)  $\vee$ 
      (s[i]0 = IND AÑO  $\wedge$  s[i]1  $\geq$  0)  $\vee$ 
      (s[i]0 = INDTRIMESTRE  $\wedge$  1  $\leq$  s[i]1  $\leq$  4)  $\vee$ 
      (s[i]0 = CH4  $\wedge$  1  $\leq$  s[i]1  $\leq$  2)  $\vee$ 
      (s[i]0 = CH6  $\wedge$  s[i]1  $\geq$  1)  $\vee$ 
      (s[i]0 = NIVEL ED  $\wedge$  0  $\leq$  s[i]1  $\leq$  1)  $\vee$ 
      (s[i]0 = ESTADO  $\wedge$  -1  $\leq$  s[i]1  $\leq$  1)  $\vee$ 
      (s[i]0 = CAT_OCUP  $\wedge$  0  $\leq$  s[i]1  $\leq$  4)  $\vee$ 
      (s[i]0 = p47T  $\wedge$  (s[i]1  $\geq$  0  $\vee$  s[i]1 = -1)  $\vee$ 
      (s[i]0 = PP04G  $\wedge$  1  $\leq$  s[i]1  $\leq$  10))
  }
}

```

2. Predicados y Auxiliares generales

```

pred noHayHogarRepetido (th: ephh) {
  (∀h : ℤ)(0 ≤ cantidadDeAparicionesHogar(h, th) ≤ 1)}

pred noHayIndividuoRepetido (ti: ephi) {
  (∀i, j : individuo)(i, j ∈ ti →L
    i[@INDCODUSU] ≠ j[@INDCODUSU] ∨ i[@COMPONENTE] ≠ j[@COMPONENTE])}

aux cantidadDeAparicionesHogar (h: ℤ, t: seq(seq<dato>)) : ℤ =
  ∑i=0|t|-1 if t[i][0] = h then 1 else 0 fi;

pred esMatriz (m: seq(seq<dato>)) {
  |m| > 0 ∧L |m[0]| > 0 ∧L (∀j : ℤ)(0 ≤ j < filas(m) →L |m[0]| = |m[j]|)}

aux inversoDe (n: ℤ) : ℤ =
  if n = 0 then 1 else  $\frac{1}{n}$  fi;

aux filas (m : seq(seq<ℤ>)) : ℤ = |m|;

pred rango (s: seq<ℤ>, i : ℤ) {0 ≤ i < |s|}

aux @HOGCODUSU : ℤ = ord(HOGCODUSU);
aux @HOGAÑO : ℤ = ord(HOGAÑO);
aux @HOGTRIMESTRE : ℤ = ord(HOGTRIMESTRE);
aux @HOGLATITUD : ℤ = ord(HOGLATITUD);
aux @HONGLONGITUD : ℤ = ord(HONGLONGITUD);
aux @II7 : ℤ = ord(II7);
aux @REGION : ℤ = ord(REGION);
aux @MAS_500 : ℤ = ord(MAS_500);
aux @IV1 : ℤ = ord(IV1);
aux @IV2 : ℤ = ord(IV2);
aux @II2 : ℤ = ord(II2);
aux @II3 : ℤ = ord(II3);
aux @INDCODUSU : ℤ = ord(INDCODUSU);
aux @COMPONENTE : ℤ = ord(COMPONENTE);
aux @INDAÑO : ℤ = ord(INDAÑO);
aux @INDTRIMESTRE : ℤ = ord(INDTRIMESTRE);
aux @CH4 : ℤ = ord(CH4);
aux @CH6 : ℤ = ord(CH6);
aux @NIVEL_ED : ℤ = ord(NIVEL_ED);
aux @ESTADO : ℤ = ord(ESTADO);
aux @CAT_OCUP : ℤ = ord(CAT_OCUP);
aux @P47T : ℤ = ord(P47T);
aux @PP04G : ℤ = ord(PP04G);

```

3. Decisiones tomadas

1. Si bien se presume que los valores del atributo componente son correctos, especificamos componenteValido, que se asegura que estos tienen valores válidos.
2. Se asume que el atributo IV2, correspondiente a la tabla Hogares, que define la cantidad total de ambientes o habitaciones, puede tomar valores a partir de 0 en adelante.
3. En el ejercicio 3 se considera a un hogar hacinamiento crítico si el mismo posee 0 habitaciones, es decir que la componente IV2, de la tabla Hogares, de un hogar encuestado, es igual a 0.
4. En el ejercicio 7 se verifica que los hogares de th0 están en th porque sino no tendría índice correspondiente y tieneIndiceAscendente evaluaría a False.
5. En los ejercicios 7 y 11 se opta por cuantificar sobre TODOS los hogares e individuos para luego exigir que estos pertenezcan a th0/ti0, de esta manera hacemos generalizaciones sobre los elementos de las tablas.

6. En el ejercicio 8 se contempla el hecho de que los ingresos pueden ser "no informados" (el valor correspondiente al atributo es -1), en ese caso se decide darle un ingreso nulo.