

# TP1 algo-1

Manuel Lakowsky, Juan Manuel Lenardi, Federico Arienti

Septiembre 2021

## 1. tipos y enumerados

```
type dato =  $\mathbb{Z}$ 
type individuo = seq<dato>
type hogar = seq<dato>
type ephi = seq<individuo>
type ephh = seq<hogar>
type joinHI = seq<hogar  $\times$  individuo>

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, cat_ocup, p47t, ppo4g
}
```

## 2. referencias

```
aux @hogcodusu :  $\mathbb{Z}$  = itemHogar.ord(hogcodusu);
aux @hogaño :  $\mathbb{Z}$  = itemHogar.ord(hogaño);
aux @hogtrimestre :  $\mathbb{Z}$  = itemHogar.ord(hogtrimestre);
aux @hoglatitud :  $\mathbb{Z}$  = itemHogar.ord(hoglatitud);
aux @hoglongitud :  $\mathbb{Z}$  = itemHogar.ord(hoglongitud);
aux @ii7 :  $\mathbb{Z}$  = itemHogar.ord(ii7);
aux @region :  $\mathbb{Z}$  = itemHogar.ord(region);
aux @mas_500 :  $\mathbb{Z}$  = itemHogar.ord(mas_500);
aux @iv1 :  $\mathbb{Z}$  = itemHogar.ord(iv1);
aux @iv2 :  $\mathbb{Z}$  = itemHogar.ord(iv2);
aux @ii2 :  $\mathbb{Z}$  = itemHogar.ord(ii2);
aux @ii3 :  $\mathbb{Z}$  = itemHogar.ord(ii3);

aux @indcodusu :  $\mathbb{Z}$  = itemIndividuo.ord(indcodusu);
aux @componente :  $\mathbb{Z}$  = itemIndividuo.ord(componente);
aux @indaño :  $\mathbb{Z}$  = itemIndividuo.ord(indaño);
aux @indtrimestre :  $\mathbb{Z}$  = itemIndividuo.ord(indtrimestre);
aux @ch4 :  $\mathbb{Z}$  = itemIndividuo.ord(ch4);
aux @ch6 :  $\mathbb{Z}$  = itemIndividuo.ord(ch6);
aux @nivel_ed :  $\mathbb{Z}$  = itemIndividuo.ord(nivel_ed);
aux @cat_ocup :  $\mathbb{Z}$  = itemIndividuo.ord(cat_ocup);
aux @p47t :  $\mathbb{Z}$  = itemIndividuo.ord(p47t);
aux @ppo4g :  $\mathbb{Z}$  = itemIndividuo.ord(ppo4g);
```

## 3. funciones generales

```
pred esMatriz (s: seq<seq<T>>) {
    ( $\forall i : \mathbb{Z}$ )( $0 \leq i < |s| \longrightarrow |s[i]| = |s[0]|$ )
}

pred esTabla (m: seq<seq<T>>, columnas: seq<T>) {
     $|m| > 0 \wedge_L (|m[0]| = |columnas| \wedge esMatriz(m))$ 
}

aux individuosEnHogar (ti: ephi, codusuh:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  =  $\sum_{i=0}^{|ti|-1}$  if ti[i][@indcodusu] = codusuh then 1 else 0 fi;
```

## 4. especificaciones

### 4.1. proc. esEncuestaValida

```
proc esEncuestaValida (in th:  $eph_h$ , in ti:  $eph_i$ , out result: Bool) {  
  Pre {  
    True  
  }  
  Post {  
     $res = true \leftrightarrow validarEncuesta(th, ti)$   
  }  
}
```

#### 4.1.1. funciones auxiliares

/\* tabla hogares \*/

```
pred codigoValidoh (th:  $eph_h$ , ti:  $eph_i$ , i:  $\mathbb{Z}$ ) {  
   $(\exists j : \mathbb{Z})(0 \leq j < |ti| \wedge_L$   
     $th[i][@hogcodusu] = ti[j][@indcodusu]$   
  )  $\wedge$   
   $\neg(\exists k : \mathbb{Z})(0 \leq k < |th| \wedge k \neq i \wedge_L$   
     $th[i][@hogcodusu] = th[k][@hogcodusu]$   
  )  
}
```

```
pred añoYTrimestreCongruenteh (th:  $eph_h$ , i:  $\mathbb{Z}$ ) {  
   $th[i][@hogaño] = th[0][@hogaño]$   
   $th[i][@hogtrimestre] = th[0][@hogtrimestre]$   
}
```

```
pred attEnRangoh (th:  $eph_h$ , i:  $\mathbb{Z}$ ) {  
   $0 \leq th[i][@hogcodusu] \wedge$   
   $-90 \leq th[i][@hoglatitud] \leq 90 \wedge$   
   $-180 \leq th[i][@hoglongitud] \leq 180 \wedge$   
   $1 \leq th[i][@ii7] \leq 3 \wedge$   
   $1 \leq th[i][@region] \leq 6 \wedge$   
   $0 \leq th[i][@mas_500] \leq 1 \wedge$   
   $1 \leq th[i][@iv1] \leq 5 \wedge$   
   $0 < th[i][@ii2] \leq th[i][@iv2] \wedge$   
   $1 \leq th[i][@ii3] \leq 2$   
}
```

/\* tabla individuos \*/

```
pred codigoValidoi (th:  $eph_h$ , ti:  $eph_i$ , i:  $\mathbb{Z}$ ) {  
   $(\exists j : \mathbb{Z})(0 \leq j < |th| \wedge_L$   
     $ti[i][@indcodusu] = th[j][@hogcodusu]$   
  )  $\wedge$   
   $\neg(\exists k : \mathbb{Z})(0 \leq k < |ti| \wedge k \neq i \wedge_L$   
     $ti[i][@indcodusu] = ti[k][@indcodusu] \wedge$   
     $ti[i][@componente] = ti[k][@componente]$   
  )  
}
```

```
pred añoYTrimestreCongruentei (th:  $eph_h$ , ti:  $eph_i$ , i:  $\mathbb{Z}$ ) {  
   $ti[i][@indño] = th[0][@hogaño]$   
   $ti[i][@indtrimestre] = th[0][@hogtrimestre]$   
}
```

}

```

pred attEnRangoi (ti: ephi, i: ℤ) {
  0 ≤ ti[i][@indcodusu] ∧
  0 ≤ ti[i][@componente] < 20 ∧
  1 ≤ ti[i][@ch4] ≤ 2 ∧
  0 ≤ ti[i][@ch6] ∧
  0 ≤ ti[i][@nivel_ed] ≤ 1 ∧
  -1 ≤ ti[i][@estado] ≤ 1 ∧
  0 ≤ ti[i][@cat_ocup] ≤ 4 ∧
  -1 ≤ ti[i][@p47t] ∧
  1 ≤ ti[i][@ppo4g] ≤ 10))
}

```

```

pred validarEncuesta (th: ephh, ti: ephi) {

  /* tabla hogares */

  esTabla(th, itemHogar) ∧L
  (∀i : ℤ)(0 ≤ i < |th| →L
    codigoValidoh(th, ti, i) ∧
    añoYTrimestreCongruenteh(th, i) ∧
    attEnRangoh(th, i)
  ) ∧

  /* tabla individuos */

  esTabla(ti, ItemIndividuo) ∧L
  (∀i : ℤ)(0 ≤ i < |ti| →L
    codigoValidoi(th, ti, i) ∧
    añoYTrimestreCongruentei(th, ti, i) ∧
    attEnRangoi(ti, i) ∧
    individuosEnHogar(th, ti[i][@indcodusu]) ≤ 20
  )
}

```

## 4.2. proc. histHabitacional

### 4.2.1. funciones auxiliares

## 4.3. proc. laCasaEstaQuedandoChica

### 4.3.1. funciones auxiliares

## 4.4. proc. creceElTeleworkingEnCiudadesGrandes

```

proc creceElTeleworkingEnCiudadesGrandes (in t1h: ephh, in t1i: ephi, in t2h: ephh, in t2i: ephi, out res: Bool) {
  Pre {
    validarEncuesta(t1h, t1i) ∧ validarEncuesta(t2h, t2i) ∧L
    t1h[0][@hogaña] = t2h[0][@hogaña] - 1 ∧
    t1h[0][@hogtrimestre] = t2h[0][@hogtrimestre]
  }
  Post {
    res = true ⇔ percentTeleworking(t1h, t1i) < percentTeleworking(t2h, t2i)
  }
}

```

4.4.1. funciones auxiliares

4.5. proc. costoSubsidioMejora

4.5.1. funciones auxiliares