

LU:
 Apellidos:
 Nombres:
 Orden:
 Turno:

Aclaraciones: El parcial NO es a libro abierto. Cualquier decisión de interpretación que se tome debe ser aclarada y justificada. Para aprobar se requieren al menos 60 puntos. **Entregar cada ejercicio en hoja separada.**
Importante: Para la resolución del parcial NO es necesario ni está permitido el uso de acum.

```

tipo Cargo = Presidente, Senador, Diputado, Gobernador, Inten-
dente;
tipo Partido=String;
tipo DNI=Z;
tipo Nombre=String;
tipo Politico=(Nombre, Partido);

tipo Boleta {
  observador lista (b: Boleta) : Z;
  observador candidatos (b: Boleta) : [(Politico, Cargo)];
  invariante todosDelMismoPartido :
    ( $\forall c, d \leftarrow \text{candidatos}(b) \text{sgd}(prm(c)) == \text{sgd}(prm(d))$ );
  invariante listaValida : lista(b) > 0;
  invariante alMenosUnCandidato : |candidatos(b)| > 0;
  invariante sinRepetidos(primeros(candidatos))
     $\wedge$  sinRepetidos(segundos(candidatos));
}

tipo Mesa {
  observador votantes (m: Mesa) : [DNI];
  observador yaVoto (m: Mesa, d: DNI) : Bool;
  requiere  $d \in \text{votantes}(m)$ ;
  observador voto (m: Mesa, d: DNI) : Boleta;
  requiere  $d \in \text{votantes}(m) \wedge \text{yaVoto}(m, d)$ ;
  invariante DNIsValidos : ( $\forall v \leftarrow \text{votantes}(m) v > 0$ );
  invariante sinVotantesRepetidos : sinRepetidos(votantes(m));
}

tipo Eleccion {
  observador padron (e: Eleccion) : [DNI];
  observador boletas (e: Eleccion) : [Boleta];
  observador mesas (e: Eleccion) : [Mesa];
  invariante nadieVotaDosVeces : sinRepetidos(padron(e));
  invariante mesasYBoletasUnicas : ...;
  invariante padronYMesasConsistentes :
    mismos(padron(e), concat([votantes(m) | m  $\leftarrow$  mesas(e)]));
  invariante soloVotanBoletasDisponibles : ...;
}

aux sinRepetidos (l : [T]) : Bool = ( $\forall i \leftarrow [0..|l|] l_i \notin l(i..|l|)$ );
aux primeros (l : [(T, U)]) : [T] = [prm(x) | x  $\leftarrow$  l];
aux segundos (l : [(T, U)]) : [U] = [sgd(x) | x  $\leftarrow$  l];
    
```

Ejercicio 1. [30 puntos]

- [10 p.] Especificar el aux `mesasIguales (m1, m2: Mesa) = Bool`; que indica si m_1 y m_2 representan la misma mesa. Es decir, si los votantes son los mismos y votaron boletas con los mismos candidatos.
- [10 p.] Completar el invariante `mesasYBoletasUnicas` del tipo Elección, que garantiza que ninguna boleta es igual a otra, y que ninguna de las mesas es igual a otra.
- [10 p.] Completar el invariante `soloVotanBoletasDisponibles` del tipo Elección, que garantiza que todas aquellas personas que ya votaron, votaron una boleta disponible.

Ejercicio 2. [25 puntos]

Especificar el problema `mesaMasInfluyente (e: Eleccion) = result : Mesa`. Este problema devuelve la mesa con más votantes, considerando únicamente aquellas mesas en las que todos sus votantes se presentaron.

Ejercicio 3. [30 puntos]

Especificar el problema `vetar (e: Eleccion, p: Partido) = result : [DNI]`. Este problema modifica la elección e de forma tal que se elimine la boleta correspondiente al partido p . Notar que si alguna persona ya había votado a ese partido, esa persona debería volver a emitir su voto. Es decir, para esa persona el observador `yaVoto` deberá devolver *False*. Además, el problema devuelve la lista de aquellas personas que ya habían emitido su voto y habían votado al partido p .

Ejercicio 4. [15 puntos]

(Ejercicio 23.2 de la de la práctica 4)

Especificar el problema `picos (l:[Z]) = result : [Z]`, que devuelve la lista con las posiciones de los picos de l , en orden ascendente. Un *pico* es un elemento de la lista que es mayor estricto que el elemento que está a su derecha (si es que tiene) y a su izquierda (si es que tiene). Las posiciones de l se numeran desde 0. Por ejemplo:

- `picos([1, 0, 9, 3, 7, 8, 5, 3, 3, 3, 2, 7, 1]) = [0, 2, 5, 11]`
- `picos([6, 4, 2, 2, 3]) = [0, 4]`