DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Practico 1 - Especificacion

Especificacion de Elecciones Nacionales

17 de septiembre de 2023 — Algoritmos y Estructura de Datos

Alias: KBRBTMZSQLHZMHGIXKDS

Integrante	LU	Correo electrónico
Palomino, Leonardo	418/21	lpalomino2300@gmail.com
Medina Herrera, Facundo	1308/21	facundomehe@gmail.com
Seirgalea, Tobías Ezequiel	078/23	tobyseirgalea@gmail.com
Gutierrez Cruz, Cristian	226/21	cristiangutierrezcruz8@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. Definicion de Tipos

type $escrutinio = seq\langle\mathbb{Z}\rangle$

type $cant\ bancas = \mathbb{Z}$

type $dHondt = seq\langle seq\langle \mathsf{Bool}\rangle\rangle$

type $listas = seq\langle seq\langle dni : \mathbb{Z} \times genero : \mathbb{Z}\rangle\rangle$

2. Problemas

2.1. Ejercicio 1

`proc hayBallotage` (escrutinio : $seq\langle\mathbb{Z}\rangle$) : Bool
`requiere` $\{escrutinioValido(escrutinio)\}$
`asegura` $\{\neg(\exists i : \mathbb{Z})(0 \leq i < |escrutinio| \ \wedge \ porcentajeVotos(escrutinio[i]) > 45) \wedge$
$\neg(\exists i : \mathbb{Z})(0 \leq i < |escrutinio| \ \wedge \ porcentajeVotos(escrutinio[i]) > 40) \ \wedge \ diferenciaMayorA10(escrutinio[i], i)\}$

`aux porcentajeDeVotos` (e: escrutinio, i: $\mathbb{Z}$) : $\mathbb{Z} = \frac{e[i]*100}{\sum_{i=0}^{|e|-1} e[i]}$;
`pred diferenciaMayorA10` (e: escrutinio: $\mathbb{Z}$) {
$\neg(\exists a : \mathbb{Z})(0 \leq a < |escrutinio| \ \wedge_L \ |porcentajeDeVotos(e[i]) - porcentajeDeVotos(e[a])) > 10|$
}

`pred escrutinioValido` (e: escrutinio: $\mathbb{Z}$) {
$|escrutinio| > 1 \ \wedge_L \ (\exists i : \mathbb{Z})(0 \leq i < |escrutinio| - 1 \wedge_L porcentajeDeVotos(e[i]) > 3) \ \wedge_L \ \neg(\exists i, j : \mathbb{Z})(0 \leq i, j <$
$|escrutinio| - 1 \ \wedge_L \ i \neq j \ \wedge \ e[i] == e[j])$
}

2.2. Ejercicio 2

`proc hayFraude` (in escrutinio_presidencial :$seq\langle\mathbb{Z}\rangle$, in escrutinio_senadores :$seq\langle\mathbb{Z}\rangle$, in escrutinio_diputados :$seq\langle\mathbb{Z}\rangle$) : Bool
`requiere` $\{escrutinioValido(escrutinio\_presidencial) \ \wedge \ escrutinioValido(escrutinio\_senadores)$
$\wedge \ escrutinioValido(escrutinio\_diputados)\}$
`asegura` $\{res = (cantidadVotos(escrutinio\_presidencial) \neq cantidadVotos(escrutinio\_senadores)) \vee$
$(cantidadVotos(escrutinio\_senadores) \neq cantidadVotos(escrutinio\_diputados)) \vee$
$(cantidadVotos(escrutinio\_diputados) \neq cantidadVotos(escrutinio\_presidencial))\}$

`aux cantidadVotos` (in escrutinio : $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z} = \sum_{i=0}^{|escrutinio|-1} escrutinio[i]$;

2.3. Ejercicio 3

`proc obtenerSenadoresEnProvincia` (in escrutinio :$seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ x $\mathbb{Z}$
`requiere` $\{escrutinioValido(escrutinio)\}$
`asegura` $\{(masVotado(escrutinio), masVotado(setAt(escrutinio, masVotado(escrutinio), 0))) = res\}$

`aux masVotado` (in escrutinio : $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z} = (\forall i : \mathbb{Z})$ (
$0 \leq i < |escrutinio| \longrightarrow escrutinio[i] < escrutinio[res]$
) ;

2.4. Ejercicio 4

`proc calcularDHondtEnProvincia` (in cant_bancas, in escrutinio :$seq\langle\mathbb{Z}\rangle$) : $seq\langle seq\langle\mathbb{Z}\rangle\rangle$
`requiere` $\{escrutinioValido(escrutinio) \wedge cant\_bancas > 0\}$
`asegura` $\{(\forall i : \mathbb{Z})(\forall bancas : \mathbb{Z})(0 \leq i < |escrutinio| - 1 \wedge 0 \leq bancas < cant\_bancas) \longrightarrow_L$
$res[i][bancas] = \frac{escrutinio[i]}{bancas+1}\}$

2.5. Ejercicio 5

proc obtenerDiputadosEnProvincia (in cant_bancas, in escrutinio: $seq\langle\mathbb{Z}\rangle$,in d: dHondt) : $seq\langle\mathbb{Z}\rangle$
requiere $\{cant\_bancas > 0\}$
asegura $\{(\forall i : \mathbb{Z})(0 \leq i < |res| \longrightarrow_L res[i] = cantBancasObtenidas(d, i)\}$
aux cantBancasObtenidas (in d: dHondt, in partido:$\mathbb{Z}$) : $\mathbb{Z}$ =
$\sum_{j=0}^{|d|}$if $d[partido][j] \in numerosMaximosDeMatriz(d)$ then 1 else 0 fi ;

aux numerosDeMatriz (in d: dHondt, in cant_bancas: $\mathbb{Z}$) : $seq\langle\mathbb{Z}\rangle$ =
$(\forall i : \mathbb{Z})(0 \leq i < |d| \wedge_L superaElUmbral(d, i) \longrightarrow_L concat(numerosDeMatriz, dHont[i])$;
aux ordenarSecuencia (in s : $seq\langle\mathbb{Z}\rangle$) : $seq\langle\mathbb{Z}\rangle$ =
$(\forall i : \mathbb{Z})(0 \leq i < |s| - 1 \longrightarrow_L$ if $s[i] \geq [i + 1]$ then $concat(ordenarSecuencia(s), s[i])$ else $skip$ fi ;

aux numerosMayoresDeMatriz (in s: $seq\langle\mathbb{Z}\rangle$, in cant_bancas: $\mathbb{Z}$) : $seq\langle\mathbb{Z}\rangle$ =
$(\forall i : \mathbb{Z})(0 \leq i < cant\_bancas \longrightarrow_L$
$concat(numeroMayoresDeMatriz(ordenarSecuencia(numerosDeMatriz), cant\_bancas), s[i])$;

aux porcentajeDeVotosMatriz (d: dHondt, partido: $\mathbb{Z}$) : $\mathbb{Z}$ = $\frac{d[partido][0]*100}{\sum_{partido=0}^{|d|-2} d[partido][0]}$;

pred superaElUmbral (d: dHondt, partido: $\mathbb{Z}$) {
$porcentajeDeVotosMatriz(d, partido) > 3$
}

2.6. Ejercicio 6

proc validarListasDiputadosEnProvincia (in cant_bancas : $\mathbb{Z}$, in listas : $seq\langle seq\langle dni : \mathbb{Z} \times genero : \mathbb{Z}\rangle\rangle$) :
requiere $\{cant\_bancas > 0 \wedge (\forall i : \mathbb{Z})(|listas[i]| > 1\}$
asegura $\{((\forall i : \mathbb{Z})(0 \leq i < |listas|) \rightarrow ((|listas[i]| = cant\_bancas) \wedge (respetaAlternancia(listas[i]))))\}$

pred respetaAlternancia (in listaPartido : $seq\langle dni : \mathbb{Z} \times genero : \mathbb{Z}\rangle$) {
$res = ((\forall j : \mathbb{Z})(0 \leq j < |listaPartido| - 1) \longrightarrow_L (((listaPartido[j]_1 = 1) \wedge (listaPartido[j + 1]_2 = 2)) \vee$
$((lista[j]_1 = 2) \wedge (lista[j + 1]_2 = 1))$
}

3. Algoritmos

```
1  res := True;
2  i := 1;
3  segundo := escrutinio[0];
4  primero := escrutinio[0];
5  cantidadVotos := 0;
6  while (i < escrutinio.size()) do
7      cantidadVotos := cantidadVotos + escrutinio[i];
8      if (escrutinio[i] > primero && i != (escrutinio.size()-1))
9          segundo := primero;
10         primero := escrutinio[i];
11     else
12         skip;
13     endif
14     if (escrutinio[i] < primero && escrutinio[i] > segundo && (i != (escrutinio.size()-1)))
15         segundo := escrutinio[i];
16     else
17         skip;
18     endif
19     i := i + 1;
20 endwhile
21 if ((((primero * 100)/cantidadVotos)) > 45 )
22         res := False;
23 endif
24 if (((((primero * 100)/cantidadVotos))) > 40 && (|primero - segundo| * (100/cantidadVotos)) > 10)
```

```
25          res := False;
26  endif
```

Código 1: Algoritmo para hayBallotage

```
1   i := 0;
2   j := 0;
3   k := 0;
4   res := False;
5   votosPresidenciales := 0;
6   votosSenadores := 0;
7   votosDiputados := 0;
8
9   while (i < escrutinio_presidencial.size()) do
10      votosPresidenciales := votosPresidenciales + escrutinio_presidencial[i]
11      i := i + 1
12  endwhile
13
14  while (j < escrutinio_senadores.size()) do
15      votosSenadores := votosSenadores + escrutinio_senadores[i]
16      j := j + 1
17  endwhile
18
19  while (k < escrutinio_Diputados.size()) do
20      votosDiputados := votosDiputados + escrutinio_diputados[i]
21      k := k + 1
22  endwhile
23
24  if(votosPresidenciales != votosSenadores)
25      res = True
26  else
27      skip
28  endif
29
30  if(votosPresidenciales != votosDiputados)
31      res = True
32  else
33      skip
34  endif
35
36  if(votosSenadores != votosDiputados)
37      res = True
38  else
39      skip
40  endif
```

Código 2: Algoritmo para hayFraude

```
1   i := 1;
2   segundo := escrutinio[0];
3   primero := escrutinio[0];
4   indiceGanador := 0;
5       while (i < escrutinio.size()) do
6           if (escrutinio[i] > primero && i != (escrutinio.size() - 1))
7               primero := escrutinio[i];
8               indiceGanador := i;
9           else
10              skip;
11          endif
12          i := i + 1;
13      endwhile
14      i := 1;
15      while (i < escrutinio.size()) do
16      if (escrutinio[i] > segundo && i != (escrutinio.size() - 1) && i != indiceGanador)
```

```
17          segundo := escrutinio[i];
18          else
19              skip;
20      endif
21      i := i + 1;
22      endwhile
23  res := (primero, segundo);
```

Código 3: Algoritmo para obtenerSenadoresEnProvincia

```
1  i := 0;
2  j := 0;
3  res := True;
4  while (i < listas.size()) do
5      if (listas[i].size() = cant_bancas)
6          while (j < listas[i].size() - 1)
7              if (listas[i][j]1 != listas[i][j + 1]1)
8                  skip
9              else
10                 res := False;
11             endif
12         j := j + 1;
13         endwhile
14     else
15         res = False;
16     endif
17     i := i + 1;
18 endwhile
```

Código 4: Algoritmo para validarListasDiputadosEnProvincia

4. Correctitud

4.1. Correctitud del algoritmo ObtenerSenadoresEnProvincia

$$P_c \longrightarrow I$$

$P_c \equiv (indiceGanador = 0, i = 1, primero = escrutinio[0], segundo = escrutinio[0])$
$I \equiv (0 \leq i \leq |escrutinio| \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
Para probar $P_c \longrightarrow I$ asumimos como cierta P_c y reemplazamos en el invariante para ver si llegamos a algo True.
$P_c \longrightarrow I \equiv (0 \leq 1 \leq |escrutinio| \wedge (\forall k : \mathbb{Z})\ (0 \leq k < 1) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\equiv (True \wedge (\forall k : \mathbb{Z})\ (k = 0) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\equiv (True \wedge (\forall k : \mathbb{Z})\ (k = 0) \longrightarrow_L (escrutinio[0] \leq escrutinio[0] \leq escrutinio[0])$
$\equiv (True \wedge True \wedge True)$
Por lo tanto $P_c \longrightarrow I$.

$$(I \wedge \neg B) \longrightarrow Q_c$$

$Q_c \equiv (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$I \wedge \neg B \equiv (0 \leq i \leq |escrutinio| \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero) \wedge i \geq |escrutinio|$
$\equiv (i = |escrutinio| \wedge (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
Separo en casos $k = i,\ k \neq i$
$\equiv (i = |escrutinio| \wedge (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio| \wedge i = k) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\wedge (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio| \wedge i \neq k) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\equiv (i = |escrutinio| \wedge (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio| \wedge i = k) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\wedge (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
Pero $Q_c \equiv (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$ *Que coincide con el caso* $k \neq i$.
Por lo tanto la implicación es válida.
Luego $(I \wedge \neg B) \longrightarrow Q_c$

$$\{I \wedge B\} S \{I\}$$

i := i + 1;

$I \equiv (0 \leq i+1 \leq |escrutinio| \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$I \equiv (-1 \leq i \leq |escrutinio| - 1 \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$

```
if (escrutinio[i] > primero && i != (escrutinio.size() - 1))
        primero := escrutinio[i];
        indiceGanador := i;
else
        skip;
endif
```

$def(B) \equiv def(escrutinio[i] > primero \wedge i \neq |escrutinio|) \equiv (0 \leq i < |escrutinio|)$
$B \wedge wp(primero := escrutinio[i], indiceGanador := i, I)$
$\equiv (escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (primero := escrutinio[i], wp(indiceGanador := i,$
$(-1 \leq i \leq |escrutinio| - 1)) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\equiv (escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (primero := escrutinio[i], (-1 \leq i \leq |escrutinio| - 1))$
$\wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero) \wedge indiceGanador = i;$
$\equiv (escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (primero := escrutinio[i], (-1 \leq i \leq |escrutinio| - 1))$
$\wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero) \wedge indiceGanador = i;$
$\equiv (escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (-1 \leq i \leq |escrutinio| - 1)) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L$
$(escrutinio[k] \leq segundo \leq escrutinio[i]) \wedge indiceGanador = i;$

Ahora calculo $\neg B \wedge WP(skip, I)$:
$\equiv \neg(escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (-1 \leq i \leq |escrutinio| - 1)$
$\wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$

Distribuyo def(B)

def(B) $\wedge_L$ $((B \wedge (s1, Q)) \vee (\neg B \wedge (s2, Q)) \equiv (def(B) \wedge_L B \wedge (s1, Q)) \vee (def(B) \wedge_L \neg B \wedge (s2, Q))$

Desarrollamos (def(B) $\wedge B \wedge (s1, Q)$)
$\equiv (escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (0 \leq i \leq |escrutinio| - 1))$
$\wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo \leq escrutinio[i]) \wedge indiceGanador = i;$

Ahora desarrollamos (def(B) $\wedge((\neg B \wedge (s2, Q))$

$\equiv \neg(escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (0 \leq i \leq |escrutinio| - 1)$
$\wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$

Llegamos a que

$\equiv (escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (0 \leq i \leq |escrutinio| - 1))$
$\wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo \leq escrutinio[i]) \wedge indiceGanador = i;$
$\bigvee$
$\neg(escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (0 \leq i \leq |escrutinio| - 1)$
$\wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$

$\equiv ((escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (0 \leq i \leq |escrutinio| - 1))$
$\wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo \leq escrutinio[i]) \wedge indiceGanador = i))$
$\bigvee$
$\neg(escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (0 \leq i \leq |escrutinio| - 1)$
$\wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$

$\equiv ((escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (0 \leq i \leq |escrutinio| - 1))$
$\wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo \leq escrutinio[i]) \wedge indiceGanador = i))$
$\bigvee$
$\neg(escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (0 \leq i \leq |escrutinio| - 1)$
$\wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$

Dividimos los casos k=i, k $\neq$ i

$\equiv ((escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (0 \leq i \leq |escrutinio| - 1))$
$\wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1 \wedge k \neq i) \longrightarrow_L escrutinio[k] \leq segundo \leq escrutinio[i] \wedge indiceGanador = i))$

$\wedge\ (\forall k : \mathbb{Z})\ (0 \leq k < i+1 \wedge k = i) \longrightarrow_L (escrutinio[k] \leq segundo \leq escrutinio[i]) \wedge indiceGanador = i))$
$\bigvee$
$\neg(escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (0 \leq i \leq |escrutinio| - 1),$
$\wedge\ (\forall k : \mathbb{Z})\ (0 \leq k < i+1 \wedge k = i) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\wedge\ (\forall k : \mathbb{Z})\ (0 \leq k < i+1 \wedge k \neq i) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$

$\equiv ((escrutinio[i] > primero \wedge i \neq |escrutinio|) \wedge (0 \leq i \leq |escrutinio| - 1))$
$\wedge\ (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L escrutinio[k] \leq segundo \leq escrutinio[i] \wedge Indice = i))$
$\wedge\ (\forall k : \mathbb{Z})\ (True) \longrightarrow_L escrutinio[k] \leq segundo \leq escrutinio[i] \wedge indiceGanador = i))$
$\bigvee$
$\neg(escrutinio[i] > primero \wedge i \neq |escrutinio|)$
$\wedge\ (\forall k : \mathbb{Z})\ (0 \leq k \leq |escrutinio| - 1) \longrightarrow_L escrutinio[k] \leq segundo \leq primero$
$\wedge\ (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L escrutinio[k] \leq segundo \leq primero$

Veamos si implica al Invariante inicial. Escribimos Invariante inicial, asumimos Invariante post ciclo como cierto y reemplazamos en el inicial hasta llegar a algo que sea True.
I $\equiv (0 \leq i \leq |escrutinio| \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\equiv (True \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\equiv (True \wedge True)$
Por lo tanto se preserva el Invariante a lo largo del primer ciclo y este es correcto.

Estudiamos 2do ciclo

I $\equiv (0 \leq i \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq segundo)$
$P_c \equiv i = 1 \wedge (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq escrutinio[indiceGanador])$
$Q_c \equiv (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq segundo)$

$$P_c \longrightarrow I$$

Probamos $P_c \longrightarrow$ I: Asumiendo como True Pc y reemplazando en I para ver si llegamos a algo cierto.
I $\equiv (0 \leq 1 \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < 1) \longrightarrow_L (escrutinio[k] \leq segundo)$
$I \equiv (True) \wedge (\forall k : \mathbb{Z})\ (k = 0) \longrightarrow_L (escrutinio[0] \leq segundo)$
$I \equiv (\forall k : \mathbb{Z})\ (k = 0) \longrightarrow_L (True)$
$I \equiv True$

$$(I \wedge \neg B) \longrightarrow Q_c$$

Vemos si $(I \wedge \neg B) \longrightarrow Q_c$, con el mismo método de antes.
I $\equiv (0 \leq i \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq segundo) \wedge i \geq |escrutinio|$
$I \equiv (i = |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq segundo)$
$I \equiv (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq segundo) \equiv$ Q_c
Por lo tanto se cumple.

$$\{I \wedge B\} S \{I\}$$

Vemos si el Invariante vale durante el ciclo con $\{I \wedge B\}S\}$:

```
        i := i + 1;
```
I $\equiv (0 \leq i+1 \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i+1) \longrightarrow_L (escrutinio[k] \leq segundo)$

```
        if (escrutinio[i] > primero && i != (escrutinio.size() - 1))
                primero := escrutinio[i];
                indiceGanador := i;
        else
                skip;
        endif
```

Aplicando $wp(if\ B\ then\ s1\ else\ s2\ fi) \equiv def(B) \wedge_L (B \wedge wp(s1, I)) \vee (\neg B \wedge wp(s2, I))$
$def(B) \equiv def(escrutinio[i] > segundo \wedge i \neq |escrutinio| \wedge i \neq indiceGanador) \equiv (0 \leq i \leq |escrutinio|)$

$B \wedge wp(segundo := escrutinio[i], I)$

$\equiv (escrutinio[i] > segundo \wedge i \neq |escrutinio| \wedge i \neq indiceGanador) \wedge wp(segundo := escrutinio[i],\ I)$
$\equiv (escrutinio[i] > segundo \wedge i \neq |escrutinio| \wedge i \neq indiceGanador) \wedge (0 \leq i \leq |escrutinio|)$
$\wedge\ (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq escrutinio[i])$
$\equiv (i \neq indiceGanador) \wedge (0 \leq i \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] < escrutinio[i])$
$\neg B \wedge wp(skip, I) \equiv \neg B \wedge I \equiv \neg(escrutinio[i] > segundo \wedge i \neq |escrutinio| \wedge i \neq indiceGanador)$
$\wedge\ (0 \leq i + 1 \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i + 1) \longrightarrow_L (escrutinio[k] \leq segundo)$
$\equiv \neg(escrutinio[i] > segundo) \vee \neg(i \neq |escrutinio|) \vee \neg(i \neq indiceGanador)) \wedge (0 \leq i + 1 \leq |escrutinio|)$
$\wedge\ (\forall k : \mathbb{Z})\ (0 \leq k < i + 1) \longrightarrow_L (escrutinio[k] \leq segundo)$
$\equiv (escrutinio[i] \leq segundo) \vee (i = |escrutinio|) \vee (i = indiceGanador)) \wedge (0 \leq i + 1 \leq |escrutinio|)$
$\wedge\ (\forall k : \mathbb{Z})\ (0 \leq k < i + 1) \longrightarrow_L (escrutinio[k] \leq segundo)$
$\equiv (escrutinio[i] \leq segundo) \vee (i = |escrutinio|) \vee (i = indiceGanador)) \wedge (-1 \leq i \leq |escrutinio| - 1)$
$\wedge\ (\forall k : \mathbb{Z})\ (0 \leq k < i + 1) \longrightarrow_L (escrutinio[k] \leq segundo)$
$\equiv (escrutinio[i] \leq segundo) \vee (False) \vee (i = indiceGanador)) \wedge (-1 \leq i \leq |escrutinio| - 1)$
$\wedge\ (\forall k : \mathbb{Z})\ (0 \leq k < i + 1) \longrightarrow_L (escrutinio[k] \leq segundo)$
$\equiv (escrutinio[i] \leq segundo) \vee (i = indiceGanador)) \wedge (-1 \leq i \leq |escrutinio| - 1)$
$\wedge\ (\forall k : \mathbb{Z})\ (0 \leq k < i + 1) \longrightarrow_L (escrutinio[k] \leq segundo)$
Uso propiedad distributiva del $\wedge_L$ *con* $def(B)$ $\wedge_L (B \wedge wp(s1, I)) \vee (\neg B \wedge wp(s2, I)) \equiv def(B) \wedge_L (B \wedge wp(s1, I) \vee (def(B) \wedge_L \neg B \wedge wp(s2, I))$
Obteniendo que
$\equiv (i \neq indiceGanador) \wedge (0 \leq i < |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] < escrutinio[i])$
$\vee$
$(escrutinio[i] \leq segundo) \vee (i = indiceGanador)) \wedge (0 \leq i \leq |escrutinio| - 1)$
$\wedge\ (\forall k : \mathbb{Z})\ (0 \leq k < i + 1) \longrightarrow_L (escrutinio[k] \leq segundo)$
$\equiv (i \neq indiceGanador) \wedge (0 \leq i < |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] < escrutinio[i])$
$\vee$
$(escrutinio[i] \leq segundo) \vee (i = indiceGanador)) \wedge (0 \leq i \leq |escrutinio| - 1)$
$\wedge\ (\forall k : \mathbb{Z})\ (0 \leq k < i + 1) \longrightarrow_L (escrutinio[k] \leq segundo)$

$I_{inicial} \equiv I(0 \leq i \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq segundo)$
Observamos lo siguiente el Invariante modificado es un caso particular del inicial:
(i $\neq indiceGanador) \wedge (0 \leq i < |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] < escrutinio[i])$
Por lo tanto {I $\wedge B\} \longrightarrow I$. Luego el ciclo es parcialmente correcto.

Ahora verificamos si el ciclo finaliza con el teorema de terminación.

$$\{I \wedge B \wedge fv = v_0\} S \{fv < v_0\}$$

$\{I \wedge B \wedge fv = v_0\} \equiv (0 \leq i \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i \longrightarrow_L (escrutinio[k] \leq segundo \leq primero))$
$\wedge\ (i < |escrutinio|) \wedge (|escrutinio| - i =v_0)$

i := i + 1;
$\equiv (0 \leq i + 1 \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i + 1) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\wedge\ (i + 1 < |escrutinio|) \wedge (|escrutinio| - (i + 1) =v_0)$
$\equiv (0 \leq i + 1 \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i + 1 \longrightarrow_L (escrutinio[k] \leq segundo \leq primero))$
$\wedge\ (i + 1 < |escrutinio|) \wedge (|escrutinio| - i - 1 <v_0)$
$\equiv (0 \leq i + 1 \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i + 1) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\wedge\ (i + 1 < |escrutinio|) \wedge (|escrutinio| - i - 1 < |escrutinio| - i)$
Basta ver que :
$|escrutinio| - i - 1 < |escrutinio| - i \equiv True.$

$$(I \wedge fv \leq 0) \longrightarrow \neg B$$

I $\wedge fv \leq 0 \equiv (0 \leq i \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)) \wedge |escrutinio| - i \leq 0$
$\equiv (0 \leq i \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero) \wedge (|escrutinio| \leq i)$
$\equiv (i = |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\equiv (i = |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
Ahora queremos ver si $(I \wedge fv \leq 0) \longrightarrow \neg B$
$\neg B \equiv i \geq |escrutinio|$
Asumimos $I \wedge fv \leq 0$ *como cierto, reemplazamos en* $\neg B$ *y vemos si llegamos a algo* $True.$
Pero $i = |escrutinio| \ \longrightarrow\ i \geq |escrutinio|$
Por lo tanto el primer ciclo cumple con correctitud pues satisface teorema invariante y teorema de terminación.
Verifico finalización del segundo ciclo

```
while (i < escrutinio.size()) do
    if (escrutinio[i] > segundo && i != (escrutinio.size() - 1) && i != indiceGanador)
        segundo := escrutinio[i];
    else
        skip;
    endif
i := i + 1;
endwhile
```

I $\wedge B \wedge fv =$v$_0$}S{fv $< v_0$}

$fv \equiv |escrutinio| - i$
Calculamos $wp(S, fv$ ¡ v$_0)$:
{I $\wedge B \wedge fv =$v$_0$} $\equiv (0 \leq i \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq segundo)$
$\wedge\ i < |escrutinio| \wedge |escrutinio| - i =v_0$
i := i + 1;
$\equiv (0 \leq i + 1 \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i + 1) \longrightarrow_L (escrutinio[k] \leq segundo)$
$\wedge (i + 1 < |escrutinio|)\ \wedge (|escrutinio| - (i + 1) =v_0)$
$\equiv (0 \leq i + 1 \leq |escrutinio|) \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i + 1) \longrightarrow_L (escrutinio[k] \leq segundo)$
$\wedge\ (i + 1\ < |escrutinio| \wedge |escrutinio| - i - 1 =v_0)$
Pero recordemos que $v_0 =$ —escrutinio— - i
Basta ver que
$|escrutinio| - (i + 1) < v_0 \equiv |escrutinio| - i - 1 < |escrutinio| - i \equiv True.$

Vemos si I $\wedge 0 \leq fv \longrightarrow \neg B$
$I \wedge 0 \leq fv \equiv 0 \leq i \leq |escrutinio| \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L escrutinio[k] \leq segundo \wedge 0 \geq |escrutinio| - i$
$I \wedge 0 \leq fv \equiv 0 \leq i \leq |escrutinio| \wedge (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L escrutinio[k] \leq segundo \wedge i \geq |escrutinio|$
$I \wedge 0 \leq fv \equiv (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L escrutinio[k] \leq segundo \wedge i = |escrutinio|$
Esto implica $\neg B$, *pues* $i = |escrutinio| \longrightarrow i \geq |escrutinio|$ *que es* $\neg B$.
Concluyendo que el ciclo es correcto.
Para poder afirmar la correctitud del programa debemos demostrar las siguientes implicaciones :

Pre $\longrightarrow$ wp(codigo previo al primer ciclo, P$_c$ *del primer ciclo*)

P_c 1*er ciclo* $\longrightarrow$ $wp($1*er ciclo*, Q_c *del primer ciclo*$)$ *Ya demostrado con el Invariante.*

Q_c 1*er ciclo* $\longrightarrow$ $wp(i := 0,\ P_c$ 2*do ciclo*$)$

P_c 2*do ciclo* $\longrightarrow$ $wp($2*do ciclo*, Q_c *del segundo ciclo*$)$ *Ya demostrado con el Invariante.*

Q_c 2*do ciclo* $\longrightarrow$ $wp(res := (primero,\ segundo),\ Post)$

Q_c 2do ciclo $\equiv (\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq segundo)$
$wp(res := (primero, segundo), res = (masVotado(escrutinio), masVotado(setAt(escrutinio, masVotado(escrutinio), 0)))$
$\equiv (primero, segundo) = (masVotado(escrutinio), masVotado(setAt(escrutinio, masVotado(escrutinio), 0)))$
$\equiv (\forall i : \mathbb{Z})\ (0 \leq i < |escrutinio| \longrightarrow (escrutinio[i] < escrutinio[primero])) \wedge ((\forall j : \mathbb{Z})\ (0 \leq j < |escrutinio|$
$\wedge\ i \neq j \longrightarrow escrutinio[j] < escrutinio[segundo]))$
Vemos que $(\forall k : \mathbb{Z})\ (0 \leq k < i) \longrightarrow_L (escrutinio[k] \leq segundo)$
Es un caso particular de $wp(res := (primero, segundo),\ res =$
$(masVotado(escrutinio), masVotado(setAt(escrutinio, masVotado(escrutinio), 0))$

Por lo tanto Q_c 2do ciclo $\longrightarrow$wp(res:= (primero, segundo), res =
(masVotado(escrutinio),masVotado(setAt(escrutinio, masVotado(escrutinio),0)))

P_c 2do ciclo $\equiv i = 1 \wedge (\forall k : \mathbb{Z})\ (0 \leq< |escrutinio|) \longrightarrow_L (escrutinio[k] \leq escrutinio[indiceGanador])$

wp(i:=1, P_c 2do ciclo) $\equiv 1 = 1 \wedge (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq escrutinio[indiceGanador])$

$\equiv True \wedge (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq escrutinio[indiceGanador])$

$\equiv (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq escrutinio[indiceGanador])$

¿Es cierto que Q_c primer ciclo $\longrightarrow$wp(i:=1, P_c 2do ciclo)?
Asumimos wp(i:=1, P_c 2do ciclo) como True y reemplazando en Q_c primer ciclo vemos si llegamos a algo cierto.
Q_c primer ciclo $\equiv (cantidadVotos = (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\equiv True \wedge (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\equiv (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$Pero\ escrutinio[indiceGanador] = primero,$
$por\ lo\ tanto\ nuestra\ wp(i := 1$,$P_c$ 2do ciclo$) \equiv (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq primero)$
$Volviendo\ a$ Q_c del primer ciclo...
$\equiv True \wedge (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$\equiv (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio|) \longrightarrow_L (escrutinio[k] \leq segundo \leq primero)$
$Vemos\ que$ Q_c es un caso particular de wp(i:=1, P_c 2do ciclo), por lo tanto $Q_c \longrightarrow$wp(i:=1, P_c 2do ciclo).

Ahora probemos que Pre $\longrightarrow$wp(codigo previo al ciclo, P_c del primer ciclo).
$Pc \equiv (escrutinioValido(escrutinio), indiceGanador = 0, i = 0, primero = escrutinio[0], segundo = escrutinio[0],$
$wp(indiceGanador := 0, primero := escrutinio[0], segundo := escrutinio[0], i := 0, P_c)$

$wp(indiceGanador := 0, wp(primero := escrutinio[0], wp(segundo := escrutinio[0], i := 0, P_c))))$
$\equiv (escrutinioValido(escrutinio), indiceGanador = 0, i = 0, primero = escrutinio[0], segundo = escrutinio[0])$
Pero se ve que
(escrutinioValido(escrutinio) $\longrightarrow$(escrutinioValido(escrutinio), indiceGanador = 0, i = 0, primero = escrutinio[0], segundo = escrutinio[0])

Tras haber probado:

- Pre $\longrightarrow$ wp(codigo previo al primer ciclo, P_c del primer ciclo)
- P_c 1er ciclo $\longrightarrow$ wp(1er ciclo,Q_c del primer ciclo) Ya demostrado con el Invariante.
- Q_c 1er ciclo $\longrightarrow$ wp(i:=0, P_c 2do ciclo)
- P_c 2do ciclo $\longrightarrow$ wp(2do ciclo,Q_c del segundo ciclo) Ya demostrado con el Invariante.
- Q_c 2do ciclo $\longrightarrow$ wp(res:= (primero, segundo), Post)

El algoritmo obtenerSenadoresEnProvincia es correcto respecto a su especificación, dado que al probar estas implicaciones, por corolario de monotonia sabemos que Pre $\longrightarrow$wp(programa completo,Post)

4.2. Correctitud del algoritmo hayFraude

$P_c \equiv$ i = 0, votos = 0, escrutinioValido(escrutinio)
$Q_c \equiv i = |escrutinio| \wedge\ votos = \sum_{j=0}^{|escrutinio|-1} esc[j]$
I $\equiv 0 \leq i \leq |escrutinio| \wedge votos = \sum_{j=0}^{i-1} esc[j]$

$P_c \longrightarrow$I Asumo P_c como True. Reemplazo en I y veo si llega a True
$\equiv 0 \leq i \leq |escrutinio| \wedge votos = \sum_{j=0}^{i-1} escrutinio[j]$
$\equiv 0 \leq 0 \leq |escrutinio| \wedge votos = \sum_{j=0}^{-1} escrutinio[j]$
$\equiv votos = 0$
$\equiv True$

$$(I \wedge \neg B) \longrightarrow Qc$$

$B \equiv i < |escrutinio| \longrightarrow \neg B \equiv i \geq |escrutinio|$
$I \wedge \neg B \equiv 0 \leq i \leq |escrutinio| \wedge votos = \sum_{j=0}^{i-1} escrutinio[j] \wedge i \geq |escrutinio|$
$\equiv i = |escrutinio| \wedge votos = \sum_{j=0}^{i-1} escrutinio[j]$
$\equiv votos = \sum_{j=0}^{|escrutinio|-1} escrutinio[j] \wedge i = |escrutinio|$

¿Es cierto $I \wedge \neg B \longrightarrow Q_c$? Asumo $I \wedge \neg B$ como cierto, reemplazo en Q_c y veo si llego a True

$Q_c \equiv i = |escrutinio| \wedge votos = \sum_{j=0}^{|escrutinio|-1} esc[j] \equiv I \wedge \neg B$ **CIERTO**

$$\{I \wedge B\} S \{I\}$$

$I \wedge B \equiv 0 \leq i < |escrutinio| \wedge votos = \sum_{j=0}^{i-1} escrutinio[j]$
$\equiv def(i+1) \wedge wp(i = i+1, I)$
$\equiv 0 \leq i+1 < |escrutinio| \wedge votos = \sum_{j=0}^{i} escrutinio[j]$
$\equiv -1 \leq i < |escrutinio| - 1 \wedge votos = \sum_{j=0}^{i} escrutinio[j]$
$\equiv -1 \leq i \leq |escrutinio| - 2 \wedge votos = \sum_{j=0}^{i} escrutinio[j]$
$\equiv def(votos) votos + escrutinio[i]) \wedge wp(votos = votos + escrutinio[i], I)$
$\equiv 0 \leq i < |escrutinio| \wedge -1 \leq i \leq |escrutinio| - 2 \wedge votos + escrutinio[i] = \sum_{j=0}^{i} escrutinio[j]$
$I_2 \equiv 0 \leq i \leq |escrutinio| - 2 \wedge votos = \sum_{j=0}^{i-1} escrutinio[j]$

¿$I_2 \longrightarrow I$? Asumo como True I_2. Reemplazo en I y veo si llego a True

$I_2 \equiv 0 \leq i \leq |escrutinio| - 2 \wedge votos = \sum_{j=0}^{i-1} escrutinio[j]$
$I \equiv 0 \leq i \leq |escrutinio| \wedge votos = \sum_{j=0}^{i-1} escrutinio[j]$
$I \equiv 0 \leq i \leq |escrutinio| \wedge True$
$\equiv 0 \leq i \leq |escrutinio|$

$0 \leq i \leq |escrutinio| - 2 \longrightarrow 0 \leq i \leq |escrutinio| \equiv$ True $\{I \wedge B\} S \{I\}$

Prueba Terminacion
$\{I \wedge B \wedge fv = v_0\}S\{\ fv < v_0\}$ definimos $fv = |escrutinio| - i$
$I \wedge B \wedge fv = v_0 \equiv 0 \leq i \leq |escrutinio| \wedge votos = \sum_{j=0}^{i-1} escrutinio[j] \wedge i < |escrutinio| \wedge |escrutinio| - i < v_0$
$\equiv 0 \leq i < |escrutinio| \wedge votos = \sum_{j=0}^{i-1} escrutinio[j] \wedge |escrutinio| - i < v_0$
$\equiv 0 \leq i+1 < |escrutinio| \wedge votos = \sum_{j=0}^{i-1+1} escrutinio[j] \wedge |escrutinio| - i - 1 < v_0$
$|escrutinio| - i - 1 < |escrutinio| - i$
$\equiv True$

$$\{I \wedge fv \leq 0 \longrightarrow \neg B\}$$

$I \wedge fv \leq 0 \equiv 0 \leq i \leq |escrutinio| \wedge votos = \sum_{j=0}^{i-1} escrutinio[j] \wedge |escrutinio| - i \leq 0$
$\equiv 0 \leq i \leq |escrutinio| \wedge votos \sum_{j=0}^{i-1} escrutinio[j] \wedge |escrutinio| \leq i \equiv i = |escrutinio| \wedge votos = \sum_{j=0}^{i-1} escrutinio[j] \wedge |escrutinio| \leq i$
$\equiv i = |escrutinio| \wedge votos = \sum_{j=0}^{i-1} escrutinio[j]$
$\equiv votos = \sum_{j=0}^{|escrutinio|-i} escrutinio[j] \wedge i = |escrutinio|$

Observar que
$i = |escrutinio| \longrightarrow \neg B$
$i = |escrutinio| \longrightarrow i \geq |escrutinio|$
True

El ciclo es correcto y finaliza

Debemos probar las siguientes implicaciones para poder afirmar que el programa es correcto respecto a su especificacion:

- Pre $\longrightarrow$wp(codigo previo al primer ciclo, Pc del primer ciclo)
- Pc 1er ciclo $\longrightarrow$wp(1er ciclo,Qc del primer ciclo) Ya demostrado con el Invariante.
- Qc 1er ciclo $\longrightarrow$Pc 2do ciclo
- Pc 2do ciclo $\longrightarrow$wp(2do ciclo,Qc del segundo ciclo) Ya demostrado con el Invariante.
- Qc 2do ciclo $\longrightarrow$Pc 3er ciclo

- Pc 3er ciclo $\longrightarrow$wp(3er ciclo, Qc del tercer ciclo) Ya demostrado con el Invariante.

- Qc 3er ciclo $\longrightarrow$wp(codigo post ciclo, postcondicion)

Qc 3er ciclo $\longrightarrow$wp(codigo post ciclo, postcondicion)

Qc 3er ciclo $\equiv i = |escrutinio\_presidencial| \land j = |escrutinio\_senadores| \land k = |escrutinio_diputados| \land res = False \land (\forall l : \mathbb{Z})\ (0 \leq l < |escrutinio|) \longrightarrow_L (votosPresidenciales = \sum_{j=0}^{|escrutinio|-1} escrutinio[j]) \land (\forall m : \mathbb{Z})\ (0 \leq m < |escrutinio|) \longrightarrow_L (votosSenadores = \sum_{j=0}^{|escrutinio|-1} escrutinio[j]) \land (\forall n : \mathbb{Z})\ (0 \leq n < |escrutinio|) \longrightarrow_L (votosDiputados = \sum_{j=0}^{|escrutinio|-1} escrutinio[j])$

$wp(if(votosPresidenciales \neq votosSenadores) \lor (votosPresidenciales \neq votosDiputados) \lor (votosSenadores \neq votosDiputados)$ then res:= True else skip, postcondicion)

$\equiv def(B) \land_L (B \land wp(res := True, postcondicion)) \lor (\neg B \land wp(skip, postcondicion))$
$def((votosPresidenciales \neq votosSenadores) \lor (votosPresidenciales \neq votosDiputados) \lor$ (votosSenadores $\neq votosDiputados) \equiv True$

Entonces $def(B) \land_L (B \land wp(res := True, postcondicion)) \lor (\neg B \land wp(skip, postcondicion))$

$\equiv True \land_L (B \land wp(res := True, postcondicion)) \lor (\neg B \land wp(skip, postcondicion))$

$\equiv (B \land wp(res := True, postcondicion)) \lor (\neg B \land wp(skip, postcondicion))$

$\equiv (B \land True = (cantidadVotos(escrutinio\_presidencial) \neq cantidadVotos(escrutinio\_senadores)) \lor (cantidadVotos(escrutinio\_senadores) \neq cantidadVotos(escrutinio\_diputados)) \lor (cantidadVotos(escrutinio\_diputados) \neq cantidadVotos(escrutinio\_presidencial)))$

$\equiv ((cantidadVotos(escrutinio\_presidencial) \neq cantidadVotos(escrutinio\_senadores)) \lor (cantidadVotos(escrutinio\_senadores) \neq cantidadVotos(escrutinio\_diputados)) \lor (cantidadVotos(escrutinio\_diputados) \neq cantidadVotos(escrutinio\_presidencial))) \land True = (cantidadVotos(escrutinio\_presidencial) \neq cantidadVotos(escrutinio\_senadores)) \lor (cantidadVotos(escrutinio\_senadores) \neq cantidadVotos(escrutinio\_diputados)) \lor (cantidadVotos(escrutinio\_diputados) \neq cantidadVotos(escrutinio\_presidencial)))$
$\equiv True$

Entonces solo quedaria True $\lor(\neg B \land wp(skip, postcondicion)) \equiv True$.
Pero Qc 3er ciclo $\longrightarrow$ $True$.

Podemos pensar a Qc 3er ciclo como...

$i = |escrutinio\_presidencial| \land j = |escrutinio\_senadores| \land k = |escrutinio\_diputados| \land$ res = False $\land (\forall l : \mathbb{Z})\ (0 \leq l < |escrutinio|) \longrightarrow_L votosPresidenciales = \sum_{j=0}^{|escrutinio|-1} escrutinio[j] \land (\forall m : \mathbb{Z})\ (0 \leq m < |escrutinio|) \longrightarrow_L (votosSenadores = \sum_{j=0}^{|escrutinio|-1} escrutinio[j]) \land (\forall n : \mathbb{Z})\ (0 \leq n < |escrutinio|) \longrightarrow_L (votosDiputados = \sum_{j=0}^{|escrutinio|-1} escrutinio[j]) \land True$

Asi queda mas claro que Qc 3er ciclo $\longrightarrow$wp(codigo post ciclo, postcondicion).

Ahora queremos ver que Qc 2do ciclo $\longrightarrow$ Pc 3er ciclo):

Basta ver que Qc 2do ciclo $\longrightarrow$ Pc 3er ciclo:
Q_c 2do ciclo $\equiv i = |escrutinio\_presidencial| \land j = |escrutinio\_senadores| \land k = 0 \land res = False \land (\forall l : \mathbb{Z})\ (0 \leq l < |escrutinio|) \longrightarrow_L votosPresidenciales = \sum_{j=0}^{|escrutinio|-1} escrutinio[j] \land (\forall m : \mathbb{Z})\ (0 \leq m < |escrutinio|) \longrightarrow_L (votosSenadores = \sum_{j=0}^{|escrutinio|-1} escrutinio[j])$
P_c 3er ciclo $\equiv i = |escrutinio\_presidencial| \land j = |escrutinio\_senadores| \land k = 0 \land res = False \land (\forall l : \mathbb{Z})\ (0 \leq l < |escrutinio|) \longrightarrow_L votosPresidenciales = \sum_{j=0}^{|escrutinio|-1} escrutinio[j] \land (\forall m : \mathbb{Z})\ (0 \leq m < |escrutinio|) \longrightarrow_L (votosSenadores = \sum_{j=0}^{|escrutinio|-1} escrutinio[j])$

Son equivalentes, entonces la implicacion es valida.

Q_c 1er ciclo $\longrightarrow$ P_c 2do ciclo:
Q_c 1er ciclo $\equiv i = |escrutinio\_presidencial| \land j = 0 \land k = 0 \land res = False \land$
$(\forall l : \mathbb{Z})\ ((0 \leq l < |escrutinio|) \longrightarrow_L (votosPresidenciales = \sum_{j=0}^{|escrutinio|-1} escrutinio[j])\ )$
P_c 2do ciclo $\equiv i = |escrutinio\_presidencial| \land j = 0 \land k = 0 \land res = False \land$
$(\forall l : \mathbb{Z})\ ((0 \leq l < |escrutinio|) \longrightarrow_L (votosPresidenciales = \sum_{j=0}^{|escrutinio|-1} escrutinio[j])\ )$
Son equivalentes por lo tanto la implicacion es válida. Elegimos los Qc y Pc de forma conveniente para evitar pasos extra. Dado que no hay otras instrucciones entre el final de un ciclo y el inicio del otro, podemos concluir que en ambos instantes de la ejecución se conoce la misma información.
Solo queda probar que Pre $\longrightarrow$ wp(codigo previo al primer ciclo, Pc del primer ciclo):
wp(codigo previo al primer ciclo, P_c del primer ciclo)
$\equiv wp(i := 0, j := 0, k := 0, res := False, votosPresidenciales := 0, votosSenadores := 0, votosDiputados := 0$,$P_c$ primer ciclo)
$\equiv (i := 0, j := 0, k := 0, res := False, votosPresidenciales := 0, votosSenadores := 0, votosDiputados := 0$,$P_c$ primer ciclo, $i = 0 \land j = 0 \land k = 0 \land votosDiputados = 0 \land votosSenadores = 0 \land votosPresidenciales = 0 \land res = False$)
$\equiv True$
Debemos ver que $Pre \longrightarrow True$ *pero como podemos asumir lo declarado en pre como* $True$*, entonces la implicacion es valida.*