

SUMARIO

SUMARIO	1
A. CÓDIGO DEL PROGRAMA WITNESS	3
A.1. Código del modelo de simulación	3
A.1.1. Elemento Vehículo	3
A.1.2. Elemento almacén	5
A.1.3 Elemento máquina	8
A.2. Código del algoritmo	17
A.2.1. Elemento Velocidad	18
A.2.2. Elemento Velocidad1	18
A.2.3. Elemento Veriv	19
A.3. Obtención de resultados en el modelo de Witness	24
A.3.1. Elemento Escribir	25
A.3.2. Elemento Escribir1	26
A.4. Inicialización de las variables	32





A. Código del programa Witness

A.1. Código del modelo de simulación

En este apartado se detallará el código C, utilizado en Witness para el desarrollo del modelo. En la Fig. A.1, se puede ver una imagen extraída del software Witness del modelo que se ha simulado con los nombres de cada elemento.

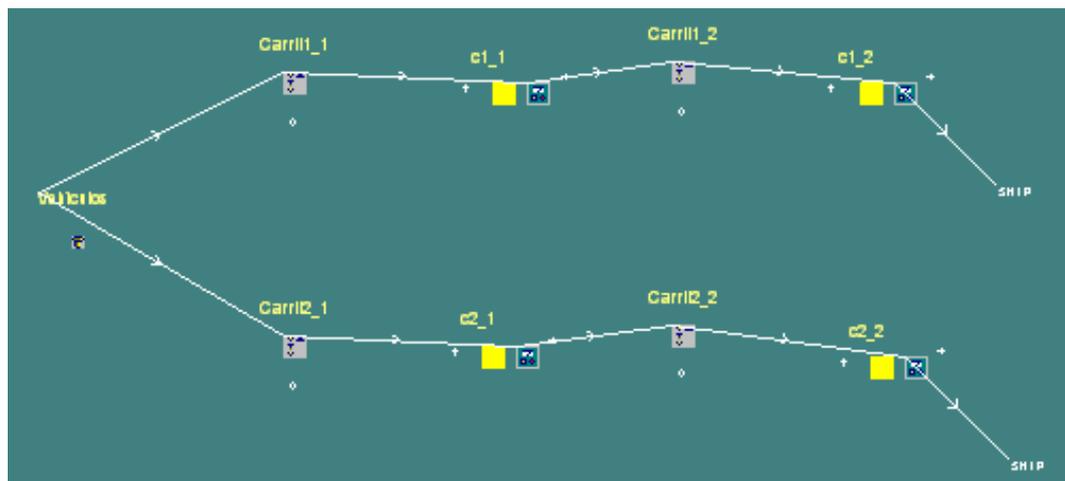


Fig. A.1 – Imagen del modelo resultante en Witness.

A continuación se expondrá el código que compone cada uno de los elementos, así como las propiedades más destacables que deben tener los elementos, para funcionar correctamente. Para ello se definen en primer lugar las propiedades y el código del elemento Vehículo, así como de un almacén, como Carril1_1 y una máquina c1_1 y después a partir de estos dos últimos elementos se extrapolará a todos los demás.

A.1.1. Elemento Vehículo

El elemento Vehículo está definido como una parte que circula a través del sistema. En la Fig. A.2, se pueden ver las propiedades más destacables del elemento.



Fig. A.2 – Propiedades elementales del elemento Vehículos en el software Witness.

A continuación se expondrá el código que se encuentra en las diferentes partes, en primer lugar se expondrá el código de Actions on Create y en segundo lugar el del To.

Actions on Create

```

tsv = TIME - tsv1
tsv1 = TIME
nc = UNIFORM (0,100,pse)
IF nc <= 50
    c = 1
ELSEIF nc > 50
    c = 2
ENDIF
panel = vp1
IF c = 1
    IF vcambio = 0
        IF vp1 >= 70
            v = NORMAL (vmedia,4,pse)
        ELSEIF vp1 < 70
            v = NORMAL (vmedia,6,pse)
        ENDIF
    ELSEIF vcambio = 1
        IF vp1 >= 70
            v = NORMAL (vp1 - 2,4,pse)
        ELSEIF vp1 < 70
            v = NORMAL (vp1 - 2,6,pse)
        ENDIF
    
```



```

    ENDIF
ELSEIF c = 2
    IF vcambio = 0
        IF vp1 >= 70
            v = NORMAL (vmedia - 2,4,pse)
        ELSEIF vp1 < 70
            v = NORMAL (vmedia - 2,6,pse)
        ENDIF
    ELSEIF vcambio = 1
        IF vp1 >= 70
            v = NORMAL (vp1 - 4,4,pse)
        ELSEIF vp1 < 70
            v = NORMAL (vp1 - 4,6,pse)
        ENDIF
    ENDIF
ENDIF
ENDIF
ntv = UNIFORM (0,100,pse + 1)
IF ntv <= tv1
    tv = 1
    d = 4
ELSEIF tv1 < ntv AND ntv <= tv1 + tv2
    tv = 2
    d = 4
ELSEIF tv1 + tv2 < ntv AND ntv <= tv1 + tv2 + tv3
    tv = 3
    d = 6
ELSEIF ntv > tv1 + tv2 + tv3
    tv = 4
    d = 8
ENDIF

```

To

```

IF c = 1
    PUSH to Carril1_1
ELSEIF c = 2
    PUSH to Carril2_1
ELSE
    Wait
ENDIF

```

A.1.2. Elemento almacén

Los elementos Carril1_1, Carril2_1, Carril1_2 y Carril2_2 están definidos como almacenes donde se acumulan las piezas del sistema, en este caso los vehículos. En la Fig. A.3, se pueden ver las propiedades más destacables de estos elementos, en concreto las del elemento Carril1_1, pero son las mismas para el resto de los elementos anteriores.



Fig. A.3 – Propiedades elementales del elemento Carril1_1 en el software Witness.

El código que se puede encontrar dentro de las propiedades del elemento almacén, como el que se puede encontrar en las pestañas Actions on Input y Actions on Output, para el elemento Carril1_1 se puede ver a continuación.

Actions on Input

```

te = TIME
tcarril = TIME + s * 3600 / v
nv1_1 = nv1_1 + 1
nv = nv1_1
nie1 = nie1 + 1
RecordRealValue (te_1_1,te)
IF nie1 >= ni
    nie_1 = nie_1 + 1
    l1e = ni * 3600 / (RNAMEVAL (te_1_1,nie1) - RNAMEVAL (te_1_1,nie_1))
ENDIF
RecordRealValue (te_11,te)
nite = nite + 1
IF nite >= ni
    nite_1 = nite_1 + 1
    lte = ni * 3600 / (RNAMEVAL (te_11,nite) - RNAMEVAL (te_11,nite_1))
ENDIF
ltev = lte
RecordRealValue (lte_1_1,lte)
RecordRealValue (v_1_1,v)
RecordRealValue (tv_1_1,tv)
RecordRealValue (d_1_1,d)
RecordRealValue (tsv_1_1,tsv)

```



RecordRealValue (tcarril_1_1,tcarril)
 RecordRealValue (nv_1_1,nv)
 RecordRealValue (l1e_1,l1e)

El código para el resto de elementos almacén del modelo, es decir, Carril2_1, Carril1_2 y Carril2_2, tiene la misma estructura que el visto anteriormente, únicamente varían algunas de las variables que se pueden encontrar. Por ello, en la Tabla A.1, se puede encontrar una relación de las variables que varían así como su nueva nomenclatura.

Variable en c1_1	Variable en c2_2	Variable en c1_2	Variable en c2_2
nv1_1	nv2_1	nv1_2	nv2_2
nv	nv	nv_2	nv_2
nie1	nie2	nie1_2	nie2_2
te_1_1	te_2_1	te_1_2	tc2_2
nie_1	nie_2	nie_1_2	tc2_1_2
l1e	l2e	l1e_2	l2e_2
te_11	te_11	te_22	te_22
nite	nite	nite_2	nite2_2
nite_1	nite_1	nite_1_2	nite_2
lte	lte	lte_2	lte_2
lte_1_1	lte_2_1	lte_1_2	lte_1_2
v_1_1	v_2_1	v_1_2	v_2_2
tv_1_1	tv_2_1	tv_1_2	tv_2_2
d_1_1	d_2_1	d_1_2	d_2_2
tsv_1_1	tsv_2_1	tsv_1_2	tsv_2_2
tcarril_1_1	tcarril_2_1	tcarril_1_2	tcarril_2_2
nv_1_1	nv_2_1	nv_1_2	nv_2_22
l1e_1	l2e_1	l1e_1_2	l2e_22

Tabla A.1 – Cambio de nomenclatura del código anterior para los diferentes elementos.

La única diferencia que existe, se encuentra en el código de los elementos Carril1_2 y Carril2_2, donde se incluye, al inicio del código anterior, una parte de código donde se vuelva a generar una nueva velocidad del vehículo, si existe una variación en los límites de circulación respecto al límite de velocidad observado en el primer tramo.

Los códigos para los dos elementos son diferentes, ya que la velocidad en los dos carriles es diferente, en el segundo carril es algo menor, por ello se puede ver una parte del código Carril1_2 a continuación:



```

IF vp2 > panel OR vp2 < panel
  IF vp2 >= 70
    v = NORMAL (vp1 - 5,1,pse)
  ELSEIF vp2 < 70
    v = NORMAL (vp1 - 6,4,pse)
  ENDIF
ENDIF

```

Y a continuación se puede ver la parte del código referida al cambio en la generación de velocidad, del elemento Carril2_2:

```

IF vp2 > panel OR vp2 < panel
  IF vp1 >= 70
    v = NORMAL (vp2 - 5,1,pse)
  ELSEIF vp2 < 70
    v = NORMAL (vp2 - 8,4,pse)
  ENDIF
ENDIF

```

A la salida del elemento también se realizan acciones. En este caso la estructura del código es la misma respecto a los cuatro elementos, la única variación es el cambio de las variables, como se puede ver en la Tabla A.2, a continuación se presenta el código y las modificaciones que presentan los diferentes elementos. A continuación se puede observar el código del elemento Carril1_1:

Actions on ouput

```

dif = TIME - tcarril
RecordRealValue (dif2_2,dif)

```

Variable en c1_1	Variable en c2_2	Variable en c1_2	Variable en c2_2
dif1	dif2	dif1_2	dif2_2

Tabla A.2 – Cambio de nomenclatura del código de Actions on Ouput para los elementos.

A.1.3 Elemento máquina

Los elementos c1_1, c2_1, c1_2 y c2_2 están definidos como máquinas en el sistema. En la Fig. A.4, se pueden ver las propiedades más destacables de estos elementos, en concreto las del elemento c1_1, pero son las mismas para el resto de los elementos anteriores



Fig. A.4 – Propiedades elementales del elemento c1_1 en el software Witness.

El código que se puede encontrar dentro de las propiedades del elemento almacén, como el que se puede encontrar en las pestañas Actions on Input y Actions on Output, para el elemento Carril1_1 se puede ver a continuación.

Actions on Input

```

IF nveh2 = 0
    vm2 = v
ENDIF
IF vm2 >= 115
    tc2_1 = (d120_115 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 110 AND vm2 < 115
    tc2_1 = (d115_110 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 105 AND vm2 < 110
    tc2_1 = (d110_105 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 100 AND vm2 < 105
    tc2_1 = (d105_100 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 95 AND vm2 < 100
    tc2_1 = (d100_95 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 90 AND vm2 < 95
    tc2_1 = (d95_90 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 85 AND vm2 < 90
    tc2_1 = (d90_85 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 80 AND vm2 < 85
    tc2_1 = (d85_80 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 75 AND vm2 < 80
    tc2_1 = (d80_75 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 70 AND vm2 < 75

```



```

    tc2_1 = (d75_70 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 65 AND vm2 < 70
    tc2_1 = (d70_65 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 60 AND vm2 < 65
    tc2_1 = (d65_60 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 55 AND vm2 < 60
    tc2_1 = (d60_55 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 50 AND vm2 < 55
    tc2_1 = (d55_50 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 45 AND vm2 < 50
    tc2_1 = (d50_45 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 40 AND vm2 < 45
    tc2_1 = (d45_40 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 35 AND vm2 < 40
    tc2_1 = (d40_35 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 30 AND vm2 < 35
    tc2_1 = (d35_30 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 25 AND vm2 < 30
    tc2_1 = (d30_25 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 20 AND vm2 < 25
    tc2_1 = (d25_20 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 15 AND vm2 < 20
    tc2_1 = (d20_15 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 10 AND vm2 < 15
    tc2_1 = (d15_10 - d) * 3.6 * tcn / v
ELSEIF vm2 >= 5 AND vm2 < 10
    tc2_1 = (d10_5 - d) * 3.6 * tcn / v
ELSEIF vm2 < 5
    tc2_1 = (d5_0 - d) * 3.6 * tcn / v
ENDIF
tc = tc2_1
RecordRealValue (tc2_1_1,tc)

```

El código para el resto de elementos máquinas del modelo, es decir, c2_1, c1_2 y c2_2, tiene la misma estructura que el visto anteriormente, únicamente varían algunas de las variables que se pueden encontrar. Por ello, en la Tabla A.3, se puede encontrar una relación de las variables que varían así como su nueva nomenclatura.

Variable en c1_1	Variable en c2_2	Variable en c1_2	Variable en c2_2
nveh1	nveh2	nveh1_2	nveh2_2
vm1	vm2	vm1_2	vm2_2
tc1_1	tc2_1	tc1_2	tc2_2
tc1_1_1	tc2_1_1	tc1_2_1	tc2_2

Tabla A.3 – Cambio de nomenclatura del código de Actions on Input para los elementos.



Actions on Ouput

```

ts = TIME
RecordRealValue (ts_2_1,ts)
RecordRealValue (ts_1,ts)
tr = ts - te - tc
RecordRealValue (tr_2_1,tr)
RecordRealValue (tr_1,tr)
vm = s * 3600 / tr
RecordRealValue (vm_2_1,vm)
RecordRealValue (vm_1,vm)
demora = tr - s * 3600 / v
RecordRealValue (demora_2_1,demora)
RecordRealValue (demora_1,demora)
demora_t2 = demora_t2 + demora
RecordRealValue (demora_t2_1,demora_t2)
RecordRealValue (demorat_1,demora_t2)
demora_t = demora_t + demora
nveh2 = nveh2 + 1
IF nveh2 = 1
    tr2 = tr
    demora_m2 = demora
    vm2 = vm
ELSEIF nveh2 > 1 AND nveh2 < ni
    tr2_m = tr2
    tr2 = (tr2_m * (nveh2 - 1) + tr) / nveh2
    demora_2m = demora_m2
    demora_m2 = (demora_2m * (nveh2 - 1) + demora) / nveh2
    vm2_m = vm2
    vm2 = (vm2_m * (nveh2 - 1) + vm) / nveh2
ELSEIF nveh2 >= ni
    nvm2_1 = nvm2_1 + 1
    nvmi2_1 = nvm2_1
    str2_1 = 0
    WHILE nvmi2_1 <= nveh2
        str2_1 = str2_1 + RNAMEVAL (tr_2_1,nvmi2_1)
        nvmi2_1 = nvmi2_1 + 1
    ENDWHILE
    tr2 = str2_1 / ni
    nvm2_1 = nvm2_1
    sdemoram2_1 = 0
    WHILE nvmi2_1 <= nveh2
        sdemoram2_1 = sdemoram2_1 + RNAMEVAL (demora_2_1,nvmi2_1)
        nvmi2_1 = nvmi2_1 + 1
    ENDWHILE
    demora_m2 = sdemoram2_1 / ni
    nvmi2_1 = nvm2_1
    svm2_1 = 0
    WHILE nvmi2_1 <= nveh2
        svm2_1 = svm2_1 + RNAMEVAL (vm_2_1,nvmi2_1)
        nvmi2_1 = nvmi2_1 + 1
    ENDWHILE

```



```

        vm2 = svm2_1 / ni
ENDIF
RecordRealValue (tr2_1,tr2)
RecordRealValue (demora_m2_1,demora_m2)
RecordRealValue (vm2_1,vm2)
nis2 = nis2 + 1
IF nis2 >= ni
    nis_2 = nis_2 + 1
    l2s = ni * 3600 / (RNAMEVAL (ts_2_1,nis2) - RNAMEVAL (ts_2_1,nis_2))
ENDIF
nsi1 = nsi1 + 1
IF TIME > tcongi
    ins1 = nsi1 * 3600 / tcong
    tcongi = tcongi + tcong
    nsi1 = 0
ENDIF
RecordRealValue (ins_1,ins1)
nits = nits + 1
IF nits < ni
    sv1s10 = sv1s10 + vm
    v1s = sv1s10 / nits
    sdemora1m10 = sdemora1m10 + demora
    demoram1 = sdemora1m10 / nits
    tr1m10 = tr1m10 + tr
    trt1 = tr1m10 / nits
ELSEIF nits >= ni
    nits_1 = nits_1 + 1
    lts = ni * 3600 / (RNAMEVAL (ts_1,nits) - RNAMEVAL (ts_1,nits_1))
    nitsv_1 = nits_1
    svts11 = 0
    WHILE nitsv_1 <= nits
        svts11 = svts11 + RNAMEVAL (vm_1,nitsv_1)
        nitsv_1 = nitsv_1 + 1
    ENDWHILE
    v1s = svts11 / ni
    k1 = lts / v1s
    oc1 = k1 * 100 * 0.005
    nitsv_1 = nits_1
    sdemora1 = 0
    WHILE nitsv_1 <= nits
        sdemora1 = sdemora1 + RNAMEVAL (demora_1,nitsv_1)
        nitsv_1 = nitsv_1 + 1
    ENDWHILE
    demoram1 = sdemora1 / ni
    nitsv_1 = nits_1
    str1 = 0
    WHILE nitsv_1 <= nits
        str1 = str1 + RNAMEVAL (tr_1,nitsv_1)
        nitsv_1 = nitsv_1 + 1
    ENDWHILE
    trt1 = str1 / ni

```



```

ENDIF
RecordRealValue (I2s_1,I2s)
RecordRealValue (Its_2_1,Its)
RecordRealValue (Its_1,Its)
RecordRealValue (v2s_1,v1s)
RecordRealValue (vs1t,v1s)
RecordRealValue (k2_1,k1)
RecordRealValue (k_1,k1)
RecordRealValue (demoram_1,demoramt1)
RecordRealValue (trm_1,trt1)
RecordRealValue (oc_1,oc1)
IF cont = 1
  IF tv = 1
    nox = (0.0929 - 0.00149 * v + 6.53e-006 * v * v) / (1 - 0.0122 * v - 3.97e-005
* v * v)
    pm10 = 0.00695
    co = (71.7 - 0.248 * v) / (1 + 35.4 * v - 0.248 * v * v)
    hc = (0.0557 - 0.0011 * v + 1.25e-005 * v * v) / (1 + 0.0365 * v - 0.000188 * v
* v)
    fc = (0.0217 + 0.253 * v + 0.00965 * v * v) / (1 + 0.096 * v - 0.000421 * v * v)
    co2 = 44.011 * (fc / (12.011 + 1.008 * 2) - co / 28.011)
    noxv_1 = noxv_1 + nox * k1
    pm10v_1 = pm10v_1 + pm10 * k1
    co_v1 = co_v1 + co * k1
    hcv_1 = hcv_1 + hc * k1
    fcv_1 = fcv_1 + fc * k1
    co2v_1 = co2v_1 + co2 * k1
  ELSEIF tv = 2
    nox = (2.82 + 0.0669 * v - 0.000463 * v * v) / (1 + 0.198 * v - 0.00143 * v * v)
    pm10 = 0.0512 - 0.00088 * v + 8.12e-006 * v * v
    co = (0.169 - 0.00292 * v + 1.25e-005 * v * v) / 1 + 1.1 / v
    hc = (0.0965 - 0.000238 * v + 1.93e-006 * v * v) / (1 + 0.103 * v - 7.24e-005 *
v * v)
    fc = (0.0162 + 2.18 * v - 0.0128 * v * v) / (1 + 0.123 * v - 0.000776 * v * v)
    co2 = 44.011 * (fc / (12.011 + 1.008 * 1.8) - co / 28.011)
    noxv_1 = noxv_1 + nox * k1
    pm10v_1 = pm10v_1 + pm10 * k1
    co_v1 = co_v1 + co * k1
    hcv_1 = hcv_1 + hc * k1
    fcv_1 = fcv_1 + fc * k1
    co2v_1 = co2v_1 + co2 * k1
  ELSEIF tv = 3
    nox = (0.000241 * v * v - 0.03181 * v + 2.0247) * (1 - 0.16)
    pm10 = (4.5e-005 * v * v - 0.004885 * v + 0.1932) * (1 - 0.33)
    co = (0.000223 * v * v - 0.026 * v + 1.076) * (1 - 0.18)
    hc = (1.75e-005 * v * v - 0.00284 * v + 0.2162) * (1 - 0.38)
    fc = 0.0198 * v * v - 2.506 * v + 137.42
    co2 = 44.011 * (fc / (12.011 + 1.008 * 2) - co / 28.011)
    noxv_1 = noxv_1 + nox * k1
    pm10v_1 = pm10v_1 + pm10 * k1
    co_v1 = co_v1 + co * k1

```



```

    hcv_1 = hcv_1 + hc * k1
    fcv_1 = fcv_1 + fc * k1
    co2v_1 = co2v_1 + co2 * k1
ELSEIF tv = 4
    d = 8
    nox = 0.0367 + 0.139 * EX (-0.0436 * v) + 0.0425 * EX (-0.138 * v)
    pm10 = EX (1.617 - 3.956 / v) - 0.928 * LN (v)
    co = -2.563 * LN (v) + 13.322
    hc = 1e-005 * v * v * v * v - 0.0038 * v * v * v + 0.3798 * v * v - 15.75 * v +
317.28
    fc = 191.108 + 700.027 * EX (-0.05283 * v) + 3813.803 * EX (-0.4522 * v)
    co2 = 44.011 * (fc / (12.011 + 1.008 * 2) - co / 28.011)
    noxv_1 = noxv_1 + nox * k1
    pm10v_1 = pm10v_1 + pm10 * k1
    co_v1 = co_v1 + co * k1
    hcv_1 = hcv_1 + hc * k1
    fcv_1 = fcv_1 + fc * k1
    co2v_1 = co2v_1 + co2 * k1
ENDIF
RecordRealValue (noxv_1_,noxv_1)
RecordRealValue (pm10v_1_,pm10v_1)
RecordRealValue (co_v1_,co_v1)
RecordRealValue (hcv_1_,hcv_1)
RecordRealValue (fcv_1_,fcv_1)
RecordRealValue (co2v_1_,co2v_1)
RecordRealValue (nox1_1v,nox)
RecordRealValue (pm101_1v,pm10)
RecordRealValue (co1_1,co)
RecordRealValue (hc1_1,hc)
RecordRealValue (fc1_1,fc)
RecordRealValue (co21_1,co2)
RecordRealValue (nox1_t,nox)
RecordRealValue (pm101_t,pm10)
RecordRealValue (co1_t,co)
RecordRealValue (hc1_t,hc)
RecordRealValue (fc1_t,fc)
RecordRealValue (co21_t,co2)
ENDIF
RecordRealValue (lte_1,ltev)
RecordRealValue (v_1,v)
RecordRealValue (tv_1,tv)
RecordRealValue (d_1,d)
RecordRealValue (tsv_1,c)
RecordRealValue (tcarril_1,tcarril)
RecordRealValue (tc_1,tc)
RecordRealValue (te_1,te)
RecordRealValue (v_p1,vp1)

```



El código para el resto de elementos máquinas del modelo, es decir, c2_1, c1_2 y c2_2, únicamente varían las variables. La nueva nomenclatura de las variables para los diferentes elementos se puede encontrar en la Tabla A.4, en la Tabla A.5 y en la Tabla A.6.

Variable en c1_1	Variable en c2_2	Variable en c1_2	Variable en c2_2
ts_1_1	ts_2_1	ts_1_2	ts_2_2
ts_1	ts_1	ts_2	ts_2
tr_1	tr_1	tr_2	tr_2
tr_1_1	tr_2_1	tr_1_2	tr_2_2
vm_1	vm_1	vm_2	vm_2
vm_1_1	vm_2_1	vm_1_2	vm_2_2
demora_1	demora_1	demora_2	demora_2
demora_1_1	demora_2_1	demora_1_2	demora_2_2
demora_t1	demora_t1	demora_t2	demora_t2
demora_t1_1	demora_t2_1	demora_t1_2	demora_t2_2
demorat_1	demorat_1	demorat_2	demorat_2
tr1	tr2	tr1_2	tr2_2
demora_m1	demora_m2	demora_m2	demora_m2
vm1	vm2	vm1_2	vm2_2
tr1_m	tr2_m	tr1_m_2	tr2_m_2
demora_1m	demora_2m	demora_1m_2	demora_2m_2
vm1_m	vm2_m	vm1_m_2	vm2_m_2
nvm1_1	nvm2_1	nvm1_2	nvm2_2
nvmi1_1	nvmi2_1	nvmi1_2	nvmi2_2
str1_1	str2_1	str1_2	str2_2
sdemoram1_1	sdemoram2_1	sdemoram1_2	sdemoram2_2
svm1_1	svm2_1	svm1_2	svm2_2
tr1_1	tr2_1	tr1_2	tr2_22
demora_m1_1	demora_m2_1	demora_m1_2	demora_m2_2
vm1_1	vm2_1	vm12	vm2_22
nis1	nis2	nis1_2	nis2_2
l1s	l2s	l1s_2	l2s_2
nis_1	nis_2	nis_1_2	nis_2_2
nits	nits	nits_2	nits_2
sv1s10	sv1s10	sv2s10	sv2s10
v1s	v1s	v2s	v2s
sdemora1m10	sdemora1m10	sdemora2m10	sdemora2m10
demoramt1	demoramt1	demoramt2	demoramt2
tr1m10	tr1m10	tr2m10	tr2m10
trt1	trt1	trt2	trt2
nits_1	nits_1	nits_2	nits_2
nitsv_1	nitsv_1	nitsv_2	nitsv_2
svts11	svts11	svts22	svts22

Tabla A.4 – Variables que intervienen en el código de los diversos elementos máquina (I).



Variable en c1_1	Variable en c2_2	Variable en c1_2	Variable en c2_2
lts	lts	lts_2	lts_2
sdemora1	sdemora1	sdemora2	sdemora2
str1	str1	str2	str2
l1s_1	l2s_1	l1s_12	ls_12
lts_1_1	lts_2_1	lts_1_2	lts_2_2
lts_1	lts_1	lts_12	lts_12
v1s_1	v2s_1	v2s_1	v2s_2
vs1t	vs1t	vs2t	vs2t
k1	k1	k2	k2
k_1	k_1	k_2	k_2
k1_1	k2_1	k1_2	k2_2
demoram_1	demoram_1	demoram_2	demoram_2
trm_1	trm_1	trm_2	trm_2
oc_1	oc_1	oc_2	oc_2
noxv_1	noxv_1	noxv_2_	noxv_2_
pm10v_1	pm10v_1	pm10v_2_	pm10v_2_
co_v1	co_v1	co_v2_	co_v2_
hcv_1	hcv_1	hcv_2_	hcv_2_
fcv_1	fcv_1	fcv_2_	fcv_2_
co2v_1	co2v_1	co2v_2_	co2v_2_
nox1_1v	nox1_1v	nox1_2v	nox1_2v
pm101_1v	pm101_1v	pm101_2v	pm101_2v
co1_1	co1_1	co1_2	co1_2
hc1_1	hc1_1	hc1_2	hc1_2
fc1_1	fc1_1	fc1_2	fc1_2
co21_1	co21_1	co21_2	co21_2
nox1_t	nox1_t	nox2_t	nox2_t
pm101_t	pm101_t	pm102_t	pm102_t
co1_t	co1_t	co2_t	co2_t
hc1_t	hc1_t	hc2_t	hc2_t
fc1_t	fc1_t	fc2_t	fc2_t
co21_t	co21_t	co22_t	co22_t
lte_1	lte_1	lte_12	lte_12
v_1	v_1	v_2	v_2
tv_1	tv_1	tv_2	tv_2
d_1	d_1	d_2	d_2
tsv_1	tsv_1	tsv_2	tsv_2
tcarril_1	tcarril_1	tcarril_2	tcarril_2
tc_1	tc_1	tc_2	tc_2
te_1	te_1	te_2	te_2

Tabla A.5 – Variables que intervienen en el código de los diversos elementos máquina (II).



Variable en c1_1	Variable en c2_2	Variable en c1_2	Variable en c2_2
v_p1	v_p1	v_p2	v_p2
noxv_1_	noxv_1_	noxv_2_	noxv_2_
pm10v_1_	pm10v_1_	pm10v_2_	pm10v_2_
co_v1_	co_v1_	co_v2_	co_v2_
hcv_1_	hcv_1_	hcv_2_	hcv_2_
fcv_1_	fcv_1_	fcv_2_	fcv_2_
co2v_1_	co2v_1_	co2v_2_	co2v_2_

Tabla A.6 – Variables que intervienen en el código de los diversos elementos máquina (III).

Para dirigir las piezas en el modelo existen dos acciones From y To, el código para todas ellas es una única línea y para los dos casos se puede encontrar en la Tabla A.7, ya que es diferente en función del elemento.

Elemento	From	To
c1_1	PULL from Carril1_1	PUSH to Carril1_2
c2_1	PULL from Carril2_1	PUSH to Carril2_2
c1_2	PULL from Carril1_2	PUSH to SHIP
c2_2	PULL from Carril2_2	PUSH to SHIP

Tabla A.7 – Código de las acciones From y To de los diferentes elementos máquina.

A.2. Código del algoritmo

En la Fig. A.5, se puede ver una imagen del programa Witness donde se puede ver el esquema utilizado para representar el algoritmo. En este apartado se presentará el código de los diferentes elementos.



Fig. A.5 – Imagen del software Witness que representa el esquema del algoritmo.



A.2.1. Elemento Velocidad

El elemento Velocidad está definido como una parte que circula a través del sistema. En la Fig. A.6, se pueden ver las propiedades más destacables del elemento.

Fig. A.6 – Propiedades elementales del elemento Velocidad en el software Witness.

A continuación se expondrá el código que se encuentra en las diferentes partes, en primer lugar se expondrá el código de To.

To

```

IF vari > TIME
    PUSH to Velocidad1
ELSEIF vari <= TIME AND cong >= 1
    PUSH to Veriv
ELSE
    Wait
ENDIF

```

A.2.2. Elemento Velocidad1

El elemento Velocidad1 está definido como una parte que circula a través del sistema. En la Fig. A.7, se pueden ver las propiedades más destacables del elemento.



Fig. A.7 – Propiedades elementales del elemento Velocidad1 en el software Witness.

A continuación se expondrá el código que se encuentra en las diferentes partes, en primer lugar se expondrá el código de To.

To

PUSH to SHIP

A.2.3. Elemento Veriv

El elemento Veriv está definido como una parte que circula a través del sistema. En la Fig. A.8, se pueden ver las propiedades más destacables del elemento.



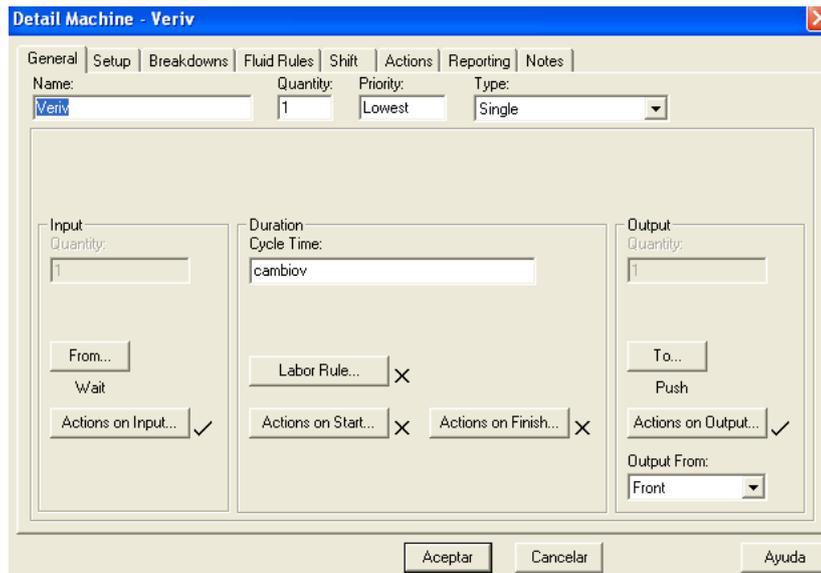


Fig. A.8 – Propiedades elementales del elemento Veriv en el software Witness.

A continuación se expondrá el código que se encuentra en las diferentes partes, en primer lugar se expondrá el código de Actions on Input, Actions on Output y To.

Actions on Input

```

IF cong = 1
  IF oc1 < oc120_110
    vpp1 = 120
  ELSEIF v1s < 110 AND v1s >= 100 AND oc1 >= oc120_110
    vpp1 = 110
  ELSEIF oc1 < oc110_100
    vpp1 = 110
  ELSEIF v1s < 100 AND v1s >= 90 AND oc1 >= oc110_100
    vpp1 = 100
  ELSEIF oc1 < oc100_90
    vpp1 = 100
  ELSEIF v1s < 90 AND v1s >= 80 AND oc1 >= oc100_90
    vpp1 = 90
  ELSEIF oc1 < oc90_80
    vpp1 = 90
  ELSEIF v1s < 80 AND v1s >= 75 AND oc1 >= oc90_80
    vpp1 = 80
  ELSEIF oc1 < oc80_75
    vpp1 = 80
  ELSEIF v1s < 75 AND v1s >= 70 AND oc1 >= oc80_75
    vpp1 = 75
  ELSEIF oc1 < oc75_70
    vpp1 = 75
  ELSEIF v1s < 70 AND v1s >= 65 AND oc1 >= oc75_70

```



```

        vpp1 = 70
    ELSEIF oc1 < oc70_65
        vpp1 = 70
    ELSEIF v1s < 65 AND v1s >= 60 AND oc1 >= oc70_65
        vpp1 = 65
    ELSEIF oc1 < oc65_60
        vpp1 = 65
    ELSEIF v1s < 60 AND v1s >= 55 AND oc1 >= oc65_60
        vpp1 = 60
    ELSEIF oc1 < oc60_55
        vpp1 = 60
    ELSEIF v1s < 55 AND v1s >= 50 AND oc1 >= oc60_55
        vpp1 = 55
    ELSEIF oc1 < oc55_50
        vpp1 = 55
    ELSEIF v1s < 50 AND v1s >= 45 AND oc1 >= oc55_50
        vpp1 = 50
    ELSEIF oc1 < oc50_45
        vpp1 = 50
    ELSEIF v1s < 45 AND v1s >= 40 AND oc1 >= oc50_45
        vpp1 = 45
    ELSEIF oc1 < oc45_40
        vpp1 = 45
    ELSEIF v1s < 40 AND oc1 >= oc45_40
        vpp1 = 40
    ENDIF
RecordRealValue (vp_p1,vpp1)
IF vpp1 > 80
    IF vp1 >= 80
        IF vpp1 - vp1 > 10
            vpp1 = vp1 + 10
        ELSEIF vp1 - vpp1 > 10
            vpp1 = vp1 - 10
        ENDIF
    ELSEIF vp1 < 80
        IF vpp1 - vp1 > 5
            vpp1 = vp1 + 5
        ELSEIF vp1 - vpp1 > 5
            vpp1 = vp1 - 5
        ENDIF
    ENDIF
ELSEIF vpp1 <= 80
    IF vpp1 - vp1 > 5
        vpp1 = vp1 + 5
    ELSEIF vp1 - vpp1 > 5
        vpp1 = vp1 - 5
    ENDIF
ENDIF
RecordRealValue (vpp1_,vpp1)
RecordRealValue (v1st_,v1s)
RecordRealValue (oc_1_,oc1)

```



```
IF oc2 < oc120_110
    vpp2 = 120
ELSEIF v2s < 110 AND v2s >= 100 AND oc2 >= oc120_110
    vpp2 = 110
ELSEIF oc2 < oc110_100
    vpp2 = 110
ELSEIF v2s < 100 AND v2s >= 90 AND oc2 >= oc110_100
    vpp2 = 100
ELSEIF oc2 < oc100_90
    vpp2 = 100
ELSEIF v2s < 90 AND v2s >= 80 AND oc2 >= oc100_90
    vpp2 = 90
ELSEIF oc2 < oc90_80
    vpp2 = 90
ELSEIF v2s < 80 AND v2s >= 75 AND oc2 >= oc90_80
    vpp2 = 80
ELSEIF oc2 < oc80_75
    vpp2 = 80
ELSEIF v2s < 75 AND v2s >= 70 AND oc2 >= oc80_75
    vpp2 = 75
ELSEIF oc2 < oc75_70
    vpp2 = 75
ELSEIF v2s < 70 AND v2s >= 65 AND oc2 >= oc75_70
    vpp2 = 70
ELSEIF oc2 < oc70_65
    vpp2 = 70
ELSEIF v2s < 65 AND v2s >= 60 AND oc2 >= oc70_65
    vpp2 = 65
ELSEIF oc2 < oc65_60
    vpp2 = 65
ELSEIF v2s < 60 AND v2s >= 55 AND oc2 >= oc65_60
    vpp2 = 60
ELSEIF oc2 < oc60_55
    vpp2 = 60
ELSEIF v2s < 55 AND v2s >= 50 AND oc2 >= oc60_55
    vpp2 = 55
ELSEIF oc2 < oc55_50
    vpp2 = 55
ELSEIF v2s < 50 AND v2s >= 45 AND oc2 >= oc55_50
    vpp2 = 50
ELSEIF oc2 < oc50_45
    vpp2 = 50
ELSEIF v2s < 45 AND v2s >= 40 AND oc2 >= oc50_45
    vpp2 = 45
ELSEIF oc2 < oc45_40
    vpp2 = 45
ELSEIF v2s < 40 AND oc2 >= oc45_40
    vpp2 = 40
ENDIF
RecordRealValue (vp_p2,vpp2)
IF vpp2 > 80
```



```

IF vp2 >= 80
  IF vpp2 - vp2 > 10
    vpp2 = vp2 + 10
  ELSEIF vp2 - vpp2 > 10
    vpp2 = vp2 - 10
  ENDIF
ELSEIF vp2 < 80
  IF vpp2 - vp2 > 5
    vpp2 = vp2 + 5
  ELSEIF vp2 - vpp2 > 5
    vpp2 = vp2 - 5
  ENDIF
ENDIF
ELSEIF vpp2 <= 80
  IF vpp2 - vp2 > 5
    vpp2 = vp2 + 5
  ELSEIF vp2 - vpp2 > 5
    vpp2 = vp2 - 5
  ENDIF
ENDIF
RecordRealValue (vpp2_,vpp2)
RecordRealValue (v2st_,v2s)
RecordRealValue (oc_2_,oc2)
IF vpp1 > vpp2
  IF vpp2 >= 80 AND vpp1 >= 80
    IF vpp1 - vpp2 > 10
      IF vpp1 >= vp1
        IF vp1 - vpp2 > 10
          vpp1 = vp1 - 10
        ELSEIF vp1 - vpp2 = 10
          vpp1 = vp1
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ELSEIF vpp2 <= 80 AND vpp1 > 80
  IF vpp2 = 80
    IF vpp1 >= vp1
      IF vp1 - vpp2 > 10
        vpp1 = vp1 - 10
      ELSEIF vp1 - vpp2 = 10
        vpp1 = vp1
      ENDIF
    ENDIF
  ELSEIF vp2 < 80
    IF vpp1 >= vp1
      IF vp1 - vpp2 > 5
        vpp1 = vp1 - 5
      ELSEIF vp1 - vpp2 = 5
        vpp1 = vp1
      ENDIF
    ENDIF
  ENDIF
ENDIF

```



```

                ENDIF
            ENDIF
        ELSEIF vpp2 <= 80 AND vpp1 <= 80
            IF vpp1 - vpp2 > 5
                IF vpp1 >= vp1
                    IF vp1 - vpp2 > 5
                        vpp1 = vp1 - 5
                    ELSEIF vp1 - vpp2 = 5
                        vpp1 = vp1
                    ENDIF
                ENDIF
            ENDIF
        ENDIF
    ENDIF
    RecordRealValue (vpp1_f,vpp1)
    tvtime = TIME
    RecordRealValue (timec, tvtime)
    IF vpp1 > vp1
        vp1 = vpp1
        vcambio = 1
    ENDIF
    IF vpp2 > vp2
        vp2 = vpp2
        vcambio = 1
    ENDIF
    vari = vari + tcambio
ENDIF

```

Actions on Ouput

```

IF vpp1 < vp1
    vp1 = vpp1
    vcambio = 1
ENDIF
IF vpp2 < vp2
    vcambio = 1
    vp2 = vpp2
ENDIF
RecordRealValue (vp1_,vp1)
RecordRealValue (vp2_,vp2)

```

To

PUSH to SHIP

A.3. Obtención de resultados en el modelo de Witness



Los resultados que se obtienen en las diferentes simulaciones, se exportan a un archivo Excel, donde se pueden tratar con mayor facilidad. Para ello, se ha realizado un esquema, la imagen del cual se puede ver en la Fig. A.9, para que se exporten los datos a Excel, en un orden determinado y al terminar la simulación.



Fig. A.9 – Imagen del software Witness que representa el esquema de los elementos para la obtención de los resultados.

Para la obtención de los resultados se simula hasta un segundo más tarde del tiempo deseado y se para la simulación, entonces se exportan todos los resultados de las variables deseadas. Para que se realicen estas acciones a continuación se puede ver un esquema detallado de los elementos.

A.3.1. Elemento Escribir

El elemento Escribir está definido como una parte que circula a través del sistema. En la Fig. A.10, se pueden ver las propiedades más destacables del elemento.

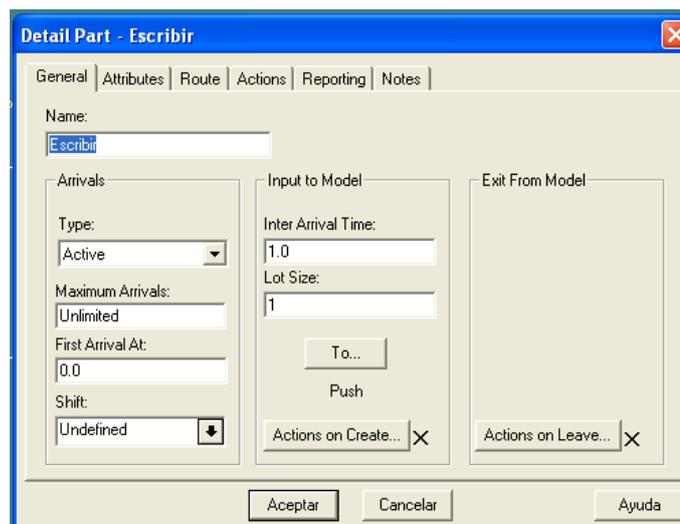


Fig. A.10 – Propiedades del elemento Escribir en el software Witness.



A continuación se expondrá el código que se encuentra en las diferentes partes, en primer lugar se expondrá el código de To.

To

PUSH to Escribir1

A.3.2. Elemento Escribir1

El elemento Escribir1 está definido como una parte que circula a través del sistema. En la Fig. A.11, se pueden ver las propiedades más destacables del elemento.

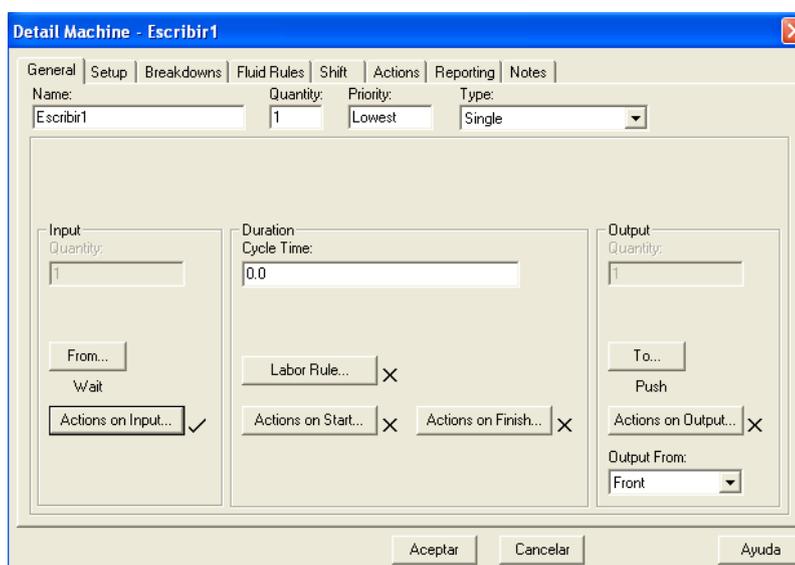


Fig. A.11 – Propiedades del elemento Escribir1 en el software Witness.

A continuación se expondrá el código que se encuentra en las diferentes partes, en primer lugar se expondrá el código de Actions on Input y To.

```
IF TIME >= 4600
```

```
    t = 2.4
```

```
    vmedia = 56.25
```

```
ENDIF
```

```
IF TIME > 8200
```

```
    IF total = 0
```

```
        XLWriteArray ("pfcn.xls", "sim1", "A2:A65000", v_1_1)
```

```
        XLWriteArray ("pfcn.xls", "sim1", "B2:B65000", tv_1_1)
```

```
        XLWriteArray ("pfcn.xls", "sim1", "C2:C65000", tsv_1_1)
```

```
        XLWriteArray ("pfcn.xls", "sim1", "D2:D65000", te_1_1)
```

```
        XLWriteArray ("pfcn.xls", "sim1", "E2:E65000", ts_1_1)
```

```
        XLWriteArray ("pfcn.xls", "sim1", "F2:F65000", demora_1_1)
```

```
        XLWriteArray ("pfcn.xls", "sim1", "G2:G65000", tr1_1)
```



XLWriteArray ("pfcn.xls","sim1","H2:H65000",vm_1_1)
XLWriteArray ("pfcn.xls","sim1","I2:I65000",vm1_1)
XLWriteArray ("pfcn.xls","sim1","J2:J65000",v1s_1)
XLWriteArray ("pfcn.xls","sim1","K2:K65000",k1_1)
XLWriteArray ("pfcn.xls","sim1","L2:L65000",tcarril_1_1)
XLWriteArray ("pfcn.xls","sim1","M2:M65000",d_1_1)
XLWriteArray ("pfcn.xls","sim1","N2:N65000",demora_t1_1)
XLWriteArray ("pfcn.xls","sim1","O2:O65000",demora_m1_1)
XLWriteArray ("pfcn.xls","sim1","P2:P65000",tr1_1)
XLWriteArray ("pfcn.xls","sim1","Q2:Q65000",tc1_1_1)
XLWriteArray ("pfcn.xls","sim1","R2:R65000",dif1)
XLWriteArray ("pfcn.xls","sim1","S2:S65000",nv_1_1)
XLWriteArray ("pfcn.xls","sim1","T2:T65000",l1e_1)
XLWriteArray ("pfcn.xls","sim1","U2:U65000",l1s_1)
XLWriteArray ("pfcn.xls","sim1","V2:V65000",lte_1_1)
XLWriteArray ("pfcn.xls","sim1","W2:W65000",lts_1_1)
XLWriteArray ("pfcn.xls","sim1","AL2:AL65000",v_2_1)
XLWriteArray ("pfcn.xls","sim1","AM2:AM65000",tv_2_1)
XLWriteArray ("pfcn.xls","sim1","AN2:AN65000",tsv_2_1)
XLWriteArray ("pfcn.xls","sim1","AO2:AO65000",te_2_1)
XLWriteArray ("pfcn.xls","sim1","AP2:AP65000",ts_2_1)
XLWriteArray ("pfcn.xls","sim1","AQ2:AQ65000",demora_2_1)
XLWriteArray ("pfcn.xls","sim1","AR2:AR65000",tr2_1)
XLWriteArray ("pfcn.xls","sim1","AS2:AS65000",vm_2_1)
XLWriteArray ("pfcn.xls","sim1","AT2:AT65000",vm2_1)
XLWriteArray ("pfcn.xls","sim1","AU2:AU65000",v2s_1)
XLWriteArray ("pfcn.xls","sim1","AV2:AV65000",k2_1)
XLWriteArray ("pfcn.xls","sim1","AW2:AW65000",tcarril_2_1)
XLWriteArray ("pfcn.xls","sim1","AX2:AX65000",d_2_1)
XLWriteArray ("pfcn.xls","sim1","AY2:AY65000",demora_t2_1)
XLWriteArray ("pfcn.xls","sim1","AZ2:AZ65000",demora_m2_1)
XLWriteArray ("pfcn.xls","sim1","BA2:BA65000",tr2_1)
XLWriteArray ("pfcn.xls","sim1","BB2:BB65000",tc2_1_1)
XLWriteArray ("pfcn.xls","sim1","BC2:BC65000",dif2)
XLWriteArray ("pfcn.xls","sim1","BD2:BD65000",nv_1_1)
XLWriteArray ("pfcn.xls","sim1","BE2:BE65000",l2e_1)
XLWriteArray ("pfcn.xls","sim1","BF2:BF65000",l2s_1)
XLWriteArray ("pfcn.xls","sim1","BG2:BG65000",lte_2_1)
XLWriteArray ("pfcn.xls","sim1","BH2:BH65000",lts_2_1)
XLWriteArray ("pfcn.xls","sim1","X2:X65000",nox1_1v)
XLWriteArray ("pfcn.xls","sim1","Y2:Y65000",pm101_1v)
XLWriteArray ("pfcn.xls","sim1","Z2:Z65000",co1_1)
XLWriteArray ("pfcn.xls","sim1","AA2:AA65000",hc1_1)
XLWriteArray ("pfcn.xls","sim1","AB2:AB65000",fc1_1)
XLWriteArray ("pfcn.xls","sim1","AC2:AC65000",co21_1)
XLWriteArray ("pfcn.xls","sim1","BI2:BI65000",nox2_1v)
XLWriteArray ("pfcn.xls","sim1","BJ2:BJ65000",pm102_1v)
XLWriteArray ("pfcn.xls","sim1","BK2:BK65000",co2_1)
XLWriteArray ("pfcn.xls","sim1","BL2:BL65000",hc2_1)
XLWriteArray ("pfcn.xls","sim1","BM2:BM65000",fc2_1)
XLWriteArray ("pfcn.xls","sim1","BN2:BN65000",co22_1)



```
XLWriteArray ("pfcn.xls", "sim2", "A2:A65000", v_1_2)
XLWriteArray ("pfcn.xls", "sim2", "B2:B65000", tv_1_2)
XLWriteArray ("pfcn.xls", "sim2", "C2:C65000", tsv_1_2)
XLWriteArray ("pfcn.xls", "sim2", "D2:D65000", te_1_2)
XLWriteArray ("pfcn.xls", "sim2", "E2:E65000", ts_1_2)
XLWriteArray ("pfcn.xls", "sim2", "F2:F65000", demora_1_2)
XLWriteArray ("pfcn.xls", "sim2", "G2:G65000", tr1_2)
XLWriteArray ("pfcn.xls", "sim2", "H2:H65000", vm_1_2)
XLWriteArray ("pfcn.xls", "sim2", "I2:I65000", vm12)
XLWriteArray ("pfcn.xls", "sim2", "J2:J65000", v1s_2)
XLWriteArray ("pfcn.xls", "sim2", "K2:K65000", k1_2)
XLWriteArray ("pfcn.xls", "sim2", "L2:L65000", tcarril_1_2)
XLWriteArray ("pfcn.xls", "sim2", "M2:M65000", d_1_2)
XLWriteArray ("pfcn.xls", "sim2", "N2:N65000", demora_t12)
XLWriteArray ("pfcn.xls", "sim2", "O2:O65000", demora_m12)
XLWriteArray ("pfcn.xls", "sim2", "P2:P65000", tr12)
XLWriteArray ("pfcn.xls", "sim2", "Q2:Q65000", tc1_2_1)
XLWriteArray ("pfcn.xls", "sim2", "R2:R65000", dif1_2)
XLWriteArray ("pfcn.xls", "sim2", "S2:S65000", nv_1_2)
XLWriteArray ("pfcn.xls", "sim2", "T2:T65000", l1e_2)
XLWriteArray ("pfcn.xls", "sim2", "U2:U65000", l1s_12)
XLWriteArray ("pfcn.xls", "sim2", "V2:V65000", lte_1_2)
XLWriteArray ("pfcn.xls", "sim2", "W2:W65000", lts_1_2)
XLWriteArray ("pfcn.xls", "sim2", "AL2:AL65000", v_2_2)
XLWriteArray ("pfcn.xls", "sim2", "AM2:AM65000", tv_2_2)
XLWriteArray ("pfcn.xls", "sim2", "AN2:AN65000", tsv_2_2)
XLWriteArray ("pfcn.xls", "sim2", "AO2:AO65000", te_2_2)
XLWriteArray ("pfcn.xls", "sim2", "AP2:AP65000", ts_2_2)
XLWriteArray ("pfcn.xls", "sim2", "AQ2:AQ65000", demora_2_2)
XLWriteArray ("pfcn.xls", "sim2", "AR2:AR65000", tr2_2)
XLWriteArray ("pfcn.xls", "sim2", "AS2:AS65000", vm_2_2)
XLWriteArray ("pfcn.xls", "sim2", "AT2:AT65000", vm2_22)
XLWriteArray ("pfcn.xls", "sim2", "AU2:AU65000", v2s_2)
XLWriteArray ("pfcn.xls", "sim2", "AV2:AV65000", k2_2)
XLWriteArray ("pfcn.xls", "sim2", "AW2:AW65000", tcarril_2_2)
XLWriteArray ("pfcn.xls", "sim2", "AX2:AX65000", d_2_2)
XLWriteArray ("pfcn.xls", "sim2", "AY2:AY65000", demora_t2_22)
XLWriteArray ("pfcn.xls", "sim2", "AZ2:AZ65000", demora_m2_22)
XLWriteArray ("pfcn.xls", "sim2", "BA2:BA65000", tr2_22)
XLWriteArray ("pfcn.xls", "sim2", "BB2:BB65000", tc2_2_1)
XLWriteArray ("pfcn.xls", "sim2", "BC2:BC65000", dif2_2)
XLWriteArray ("pfcn.xls", "sim2", "BD2:BD65000", nv_2_22)
XLWriteArray ("pfcn.xls", "sim2", "BE2:BE65000", l2e_22)
XLWriteArray ("pfcn.xls", "sim2", "BF2:BF65000", l2s_22)
XLWriteArray ("pfcn.xls", "sim2", "BG2:BG65000", lte_12)
XLWriteArray ("pfcn.xls", "sim2", "BH2:BH65000", lts_12)
XLWriteArray ("pfcn.xls", "sim2", "X2:X65000", nox1_2v)
XLWriteArray ("pfcn.xls", "sim2", "Y2:Y65000", pm101_2v)
XLWriteArray ("pfcn.xls", "sim2", "Z2:Z65000", co1_2)
XLWriteArray ("pfcn.xls", "sim2", "AA2:AA65000", hc1_2)
XLWriteArray ("pfcn.xls", "sim2", "AB2:AB65000", fc1_2)
```



XLWriteArray ("pfcn.xls", "sim2", "AC2:AC65000", co21_2)
XLWriteArray ("pfcn.xls", "sim2", "BI2:BI65000", nox2_2v)
XLWriteArray ("pfcn.xls", "sim2", "BJ2:BJ65000", pm102_2v)
XLWriteArray ("pfcn.xls", "sim2", "BK2:BK65000", co2_2)
XLWriteArray ("pfcn.xls", "sim2", "BL2:BL65000", hc2_2)
XLWriteArray ("pfcn.xls", "sim2", "BM2:M65000", fc2_2)
XLWriteArray ("pfcn.xls", "sim2", "BN2:BN65000", co22_2)
XLWriteArray ("pfcn.xls", "sim", "A2:A65000", v_1)
XLWriteArray ("pfcn.xls", "sim", "B2:B65000", tv_1)
XLWriteArray ("pfcn.xls", "sim", "C2:C65000", tsv_1)
XLWriteArray ("pfcn.xls", "sim", "D2:D65000", te_1)
XLWriteArray ("pfcn.xls", "sim", "E2:E65000", ts_1)
XLWriteArray ("pfcn.xls", "sim", "F2:F65000", demora_1)
XLWriteArray ("pfcn.xls", "sim", "G2:G65000", tr_1)
XLWriteArray ("pfcn.xls", "sim", "H2:H65000", vm_1)
XLWriteArray ("pfcn.xls", "sim", "I2:I65000", vs1t)
XLWriteArray ("pfcn.xls", "sim", "J2:J65000", k_1)
XLWriteArray ("pfcn.xls", "sim", "K2:K65000", oc_1)
XLWriteArray ("pfcn.xls", "sim", "L2:L65000", v_p1)
XLWriteArray ("pfcn.xls", "sim", "M2:M65000", tcarril_1)
XLWriteArray ("pfcn.xls", "sim", "N2:N65000", demorat_1)
XLWriteArray ("pfcn.xls", "sim", "O2:O65000", demoram_1)
XLWriteArray ("pfcn.xls", "sim", "P2:P65000", trm_1)
XLWriteArray ("pfcn.xls", "sim", "Q2:Q65000", tc_1)
XLWriteArray ("pfcn.xls", "sim", "R2:R65000", lte_1)
XLWriteArray ("pfcn.xls", "sim", "S2:S65000", lts_1)
XLWriteArray ("pfcn.xls", "sim", "AL2:AL65000", v_2)
XLWriteArray ("pfcn.xls", "sim", "AM2:AM65000", tv_2)
XLWriteArray ("pfcn.xls", "sim", "AN2:AN65000", tsv_2)
XLWriteArray ("pfcn.xls", "sim", "AO2:AO65000", te_2)
XLWriteArray ("pfcn.xls", "sim", "AP2:AP65000", ts_2)
XLWriteArray ("pfcn.xls", "sim", "AQ2:AQ65000", demora_2)
XLWriteArray ("pfcn.xls", "sim", "AR2:AR65000", tr_2)
XLWriteArray ("pfcn.xls", "sim", "AS2:AS65000", vm_2)
XLWriteArray ("pfcn.xls", "sim", "AT2:AT65000", vs2t)
XLWriteArray ("pfcn.xls", "sim", "AU2:AU65000", k_2)

XLWriteArray ("pfcn.xls", "sim", "AV2:AV65000", oc_2)
XLWriteArray ("pfcn.xls", "sim", "AW2:AW65000", v_p2)
XLWriteArray ("pfcn.xls", "sim", "AX2:AX65000", tcarril_2)
XLWriteArray ("pfcn.xls", "sim", "AY2:AY65000", demorat_2)
XLWriteArray ("pfcn.xls", "sim", "AZ2:AZ65000", demoram_2)
XLWriteArray ("pfcn.xls", "sim", "BA2:BA65000", trm_2)
XLWriteArray ("pfcn.xls", "sim", "BB2:BB65000", tc_2)
XLWriteArray ("pfcn.xls", "sim", "BC2:BC65000", lte_12)
XLWriteArray ("pfcn.xls", "sim", "BD2:BD65000", lts_12)
XLWriteArray ("pfcn.xls", "sim", "S2:S65000", nox1_t)
XLWriteArray ("pfcn.xls", "sim", "T2:T65000", pm101_t)
XLWriteArray ("pfcn.xls", "sim", "U2:U65000", co1_t)
XLWriteArray ("pfcn.xls", "sim", "V2:V65000", hc1_t)
XLWriteArray ("pfcn.xls", "sim", "W2:W65000", fc1_t)



```

XLWriteArray ("pfcn.xls", "sim", "X2:X65000", co21_t)
XLWriteArray ("pfcn.xls", "sim", "Y2:Y65000", vp1_t)
XLWriteArray ("pfcn.xls", "sim", "Z2:Z65000", noxv_1_)
XLWriteArray ("pfcn.xls", "sim", "AA2:AA65000", pm10v_1_)
XLWriteArray ("pfcn.xls", "sim", "AB2:AB65000", co_v1_)
XLWriteArray ("pfcn.xls", "sim", "AC2:AC65000", hcv_1_)
XLWriteArray ("pfcn.xls", "sim", "AD2:AD65000", fcv_1_)
XLWriteArray ("pfcn.xls", "sim", "AE2:AE65000", co2v_1_)
XLWriteArray ("pfcn.xls", "sim", "BD2:BD65000", nox2_t)
XLWriteArray ("pfcn.xls", "sim", "BE2:BE65000", pm102_t)
XLWriteArray ("pfcn.xls", "sim", "BF2:BF65000", co2_t)
XLWriteArray ("pfcn.xls", "sim", "BG2:BG65000", hc2_t)
XLWriteArray ("pfcn.xls", "sim", "BH2:BH65000", fc2_t)
XLWriteArray ("pfcn.xls", "sim", "BI2:BI65000", co22_t)
XLWriteArray ("pfcn.xls", "sim", "BK2:BK65000", noxv_2_)
XLWriteArray ("pfcn.xls", "sim", "BL2:BL65000", pm10v_2_)
XLWriteArray ("pfcn.xls", "sim", "BM2:BM65000", co_v2_)
XLWriteArray ("pfcn.xls", "sim", "BN2:BN65000", hcv_2_)
XLWriteArray ("pfcn.xls", "sim", "BO2:BO65000", fcv_2)
XLWriteArray ("pfcn.xls", "sim", "BP2:BP65000", co2v_2)
IF cong = 1
    XLWriteArray ("pfcn.xls", "simc", "A2:A65000", vpp1_)
    XLWriteArray ("pfcn.xls", "simc", "B2:B65000", vpp1_f)
    XLWriteArray ("pfcn.xls", "simc", "C2:C65000", v1st_)
    XLWriteArray ("pfcn.xls", "simc", "D2:D65000", oc_1_)
    XLWriteArray ("pfcn.xls", "simc", "E2:E65000", vp1_)
    XLWriteArray ("pfcn.xls", "simc", "F2:F65000", vpp2_)
    XLWriteArray ("pfcn.xls", "simc", "G2:G65000", timec)
    XLWriteArray ("pfcn.xls", "simc", "H2:H65000", v2st_)
    XLWriteArray ("pfcn.xls", "simc", "I2:I65000", oc_2_)
    XLWriteArray ("pfcn.xls", "simc", "J2:J65000", vp2_)
ENDIF
ELSEIF total = 1
    XLWriteArray ("pfcn.xls", "sim", "A2:A65000", v_1)
    XLWriteArray ("pfcn.xls", "sim", "B2:B65000", tv_1)
    XLWriteArray ("pfcn.xls", "sim", "C2:C65000", tsv_1)
    XLWriteArray ("pfcn.xls", "sim", "D2:D65000", te_1)
    XLWriteArray ("pfcn.xls", "sim", "E2:E65000", ts_1)
    XLWriteArray ("pfcn.xls", "sim", "F2:F65000", demora_1)
    XLWriteArray ("pfcn.xls", "sim", "G2:G65000", tr_1)
    XLWriteArray ("pfcn.xls", "sim", "H2:H65000", vm_1)
    XLWriteArray ("pfcn.xls", "sim", "I2:I65000", vs1t)
    XLWriteArray ("pfcn.xls", "sim", "J2:J65000", k_1)
    XLWriteArray ("pfcn.xls", "sim", "K2:K65000", oc_1)
    XLWriteArray ("pfcn.xls", "sim", "L2:L65000", v_p1)
    XLWriteArray ("pfcn.xls", "sim", "M2:M65000", tcarril_1)
    XLWriteArray ("pfcn.xls", "sim", "N2:N65000", demorat_1)
    XLWriteArray ("pfcn.xls", "sim", "O2:O65000", demoram_1)
    XLWriteArray ("pfcn.xls", "sim", "P2:P65000", trm_1)
    XLWriteArray ("pfcn.xls", "sim", "Q2:Q65000", tc_1)
    XLWriteArray ("pfcn.xls", "sim", "R2:R65000", lte_1)

```



```

XLWriteArray ("pfcn.xls","sim","S2:S65000",lts_1)
XLWriteArray ("pfcn.xls","sim","T2:T65000",ine_1)
XLWriteArray ("pfcn.xls","sim","V2:V65000",ins_1)
XLWriteArray ("pfcn.xls","sim","AL2:AL65000",v_2)
XLWriteArray ("pfcn.xls","sim","AM2:AM65000",tv_2)
XLWriteArray ("pfcn.xls","sim","AN2:AN65000",tsv_2)
XLWriteArray ("pfcn.xls","sim","AO2:AO65000",te_2)
XLWriteArray ("pfcn.xls","sim","AP2:AP65000",ts_2)
XLWriteArray ("pfcn.xls","sim","AQ2:AQ65000",demora_2)
XLWriteArray ("pfcn.xls","sim","AR2:AR65000",tr_2)
XLWriteArray ("pfcn.xls","sim","AS2:AS65000",vm_2)
XLWriteArray ("pfcn.xls","sim","AT2:AT65000",vs2t)
XLWriteArray ("pfcn.xls","sim","AU2:AU65000",k_2)
XLWriteArray ("pfcn.xls","sim","AV2:AV65000",oc_2)
XLWriteArray ("pfcn.xls","sim","AW2:AW65000",v_p2)
XLWriteArray ("pfcn.xls","sim","AX2:AX65000",tcarril_2)
XLWriteArray ("pfcn.xls","sim","AY2:AY65000",demorat_2)
XLWriteArray ("pfcn.xls","sim","AZ2:AZ65000",demoram_2)
XLWriteArray ("pfcn.xls","sim","BA2:BA65000",trm_2)
XLWriteArray ("pfcn.xls","sim","BB2:BB65000",tc_2)
XLWriteArray ("pfcn.xls","sim","BC2:BC65000",lte_12)
XLWriteArray ("pfcn.xls","sim","BD2:BD65000",lts_12)
XLWriteArray ("pfcn.xls","sim","T2:T65000",nox1_t)
XLWriteArray ("pfcn.xls","sim","U2:U65000",pm101_t)
XLWriteArray ("pfcn.xls","sim","V2:V65000",co1_t)
XLWriteArray ("pfcn.xls","sim","W2:W65000",hc1_t)
XLWriteArray ("pfcn.xls","sim","X2:X65000",fc1_t)
XLWriteArray ("pfcn.xls","sim","Y2:Y65000",co21_t)
XLWriteArray ("pfcn.xls","sim","Z2:Z65000",vp1_t)
XLWriteArray ("pfcn.xls","sim","AA2:AA65000",noxv_1_)
XLWriteArray ("pfcn.xls","sim","AB2:AB65000",pm10v_1_)
XLWriteArray ("pfcn.xls","sim","AC2:AC65000",co_v1_)
XLWriteArray ("pfcn.xls","sim","AD2:AD65000",hcv_1_)
XLWriteArray ("pfcn.xls","sim","AE2:AE65000",fcv_1_)
XLWriteArray ("pfcn.xls","sim","AF2:AF65000",co2v_1_)
XLWriteArray ("pfcn.xls","sim","BE2:BE65000",nox2_t)
XLWriteArray ("pfcn.xls","sim","BF2:BF65000",pm102_t)
XLWriteArray ("pfcn.xls","sim","BG2:BG65000",co2_t)
XLWriteArray ("pfcn.xls","sim","BH2:BH65000",hc2_t)
XLWriteArray ("pfcn.xls","sim","BI2:BI65000",fc2_t)
XLWriteArray ("pfcn.xls","sim","BJ2:BJ65000",co22_t)
XLWriteArray ("pfcn.xls","sim","BK2:BK65000",noxv_2_)
XLWriteArray ("pfcn.xls","sim","BL2:BL65000",pm10v_2_)
XLWriteArray ("pfcn.xls","sim","BM2:BM65000",co_v2_)
XLWriteArray ("pfcn.xls","sim","BN2:BN65000",hcv_2_)
XLWriteArray ("pfcn.xls","sim","BO2:BO65000",fcv_2)
XLWriteArray ("pfcn.xls","sim","BP2:BP65000",co2v_2)
IF cong = 1
    XLWriteArray ("pfcn.xls","simc","A2:A65000",vpp1_)
    XLWriteArray ("pfcn.xls","simc","B2:B65000",vpp1_f)
    XLWriteArray ("pfcn.xls","simc","C2:C65000",v1st_)

```



```

XLWriteArray ("pfcn.xls", "simc", "D2:D65000", oc_1_)
XLWriteArray ("pfcn.xls", "simc", "E2:E65000", vp1_)
XLWriteArray ("pfcn.xls", "simc", "F2:F65000", vpp2_)
XLWriteArray ("pfcn.xls", "simc", "G2:G65000", timec)
XLWriteArray ("pfcn.xls", "simc", "H2:H65000", v2st_)
XLWriteArray ("pfcn.xls", "simc", "I2:I65000", oc_2_)
XLWriteArray ("pfcn.xls", "simc", "J2:J65000", vp2_)
    ENDIF
  ENDIF
ENDIF

```

To

PUSH to SHIP

A.4. Inicialización de las variables

A continuación se puede observar el valor inicial de algunas de las variables que se encuentran en el modelo de simulación.

```

pse = 2
t = 4.135
vp1 = 80
vp2 = 80
tv1 = 40.97
tv2 = 50.65
tv3 = 5.58
tv4 = 2.79
s1 = 0.689
s2 = 0.711
d120_115 = 100
d115_110 = 98
d110_105 = 93
d105_100 = 90
d100_95 = 88
d95_90 = 84
d90_85 = 80
d85_80 = 75
d80_75 = 70
d75_70 = 65
d70_65 = 58
d65_60 = 50
d60_55 = 40
d55_50 = 33
d50_45 = 27

```



d45_40 = 23.25
d40_35 = 20
d35_30 = 18
d30_25 = 16
d25_20 = 14.5
d20_15 = 13
d15_10 = 11
d10_5 = 9
d5_0 = 6
ti = 30
ni = 40
tcn = 3
total = 1
cong = 1
vari = 60
tcambio = 60
oc120_110 = 1
oc110_100 = 1.5
oc100_90 = 2.1
oc90_80 = 3
oc80_75 = 3.8
oc75_70 = 6
oc70_65 = 7
oc65_60 = 8
oc60_55 = 10
oc55_50 = 16
oc50_45 = 19
oc45_40 = 25
vmedia = 74.25
cont = 1

