

Projecte de Fi de Carrera
Enginyer Industrial

**Desenvolupament d'eines de suport a la
simulació basada en esdeveniments discrets de
sistemes de fabricació flexible**

ANNEX A: Manual de l'aplicació

ANNEX B: Codi de l'aplicació

Autor: Ariadna Olivar Morente
Director: Jan Rosell Gratacós
Convocatòria: Setembre 2016 (pla 94)



**Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona**



Resum

En aquest annex es presenta, de manera separada degut a la seva extensió, informació necessària per complementar la memòria *Eines de suport a la simulació basada en esdeveniments discrets de sistemes de fabricació*.

L'annex consta de dues parts. La primera és un manual de l'aplicació que ajudarà l'usuari a utilitzar-la més fàcilment i a entendre les limitacions o errors amb què es pot trobar.

La segona part conté tot el codi VBA que controla l'aplicació.

Sumari

| | |
|---|-----------|
| RESUM | 1 |
| SUMARI | 2 |
| 1. MANUAL DE L'APLICACIÓ | 5 |
| 1.1. Pestanya principal (<i>Home</i>)..... | 5 |
| 1.2. Anàlisi del transitori (Transient state analysis)..... | 8 |
| 1.3. Càcul del nombre de rèpliques (<i>Replication number</i>)..... | 12 |
| 1.4. Validació del model (<i>Model validation</i>) | 14 |
| 1.5. Comparació d'alternatives (<i>Alternatives comparison</i>) | 16 |
| 1.6. Disseny factorial (<i>Factorial Design</i>) | 18 |
| 1.7. Índex d'errors | 22 |
| 2. CODI DE L'APLICACIÓ | 23 |
| 2.1. Objectes de Microsoft Excel..... | 23 |
| 2.1.1. Main..... | 23 |
| 2.1.2. Transient State Analysis | 24 |
| 2.1.3. Replication Number | 26 |
| 2.1.4. Model Validation..... | 28 |
| 2.1.5. Alternatives Comparison | 30 |
| 2.1.6. Factorial Design..... | 32 |
| 2.1.7. ThisWorkbook..... | 34 |
| 2.2. Formularis | 34 |
| 2.2.1. TSA_Definition_Form | 34 |
| 2.2.2. RN_OutputVariableSelection_Form | 34 |
| 2.2.3. MV_OutputVariableSelection_Form | 35 |
| 2.2.4. AC_OutputVariableSelection_Form | 35 |
| 2.2.5. FD_Definition_Form..... | 35 |
| 2.2.6. FD_OutputVariableSelection_Form..... | 38 |
| 2.3. Mòduls | 39 |
| 2.3.1. Mòdul 1 | 39 |
| 2.3.2. Mòdul 2 | 39 |
| 2.3.3. Mòdul 3 | 40 |
| 2.3.4. Mòdul 4 | 48 |
| 2.3.5. Mòdul 5 | 55 |
| 2.3.6. Mòdul 6 | 60 |
| 2.3.7. Mòdul 7 | 63 |

| | |
|----------------------|----|
| 2.3.8. Mòdul 8 | 66 |
| 2.3.9. Mòdul 9 | 70 |

1. Manual de l'aplicació

Aquest capítol pretén ser una petita guia per posar en coneixement de l'usuari totes les funcionalitats que incorpora l'aplicació, així com emfatitzar els aspectes que cal tenir en compte a l'hora d'utilitzar-la.

1.1. Pestanya principal (*Home*)

És la primera pantalla que es mostra en accedir a l'aplicació.

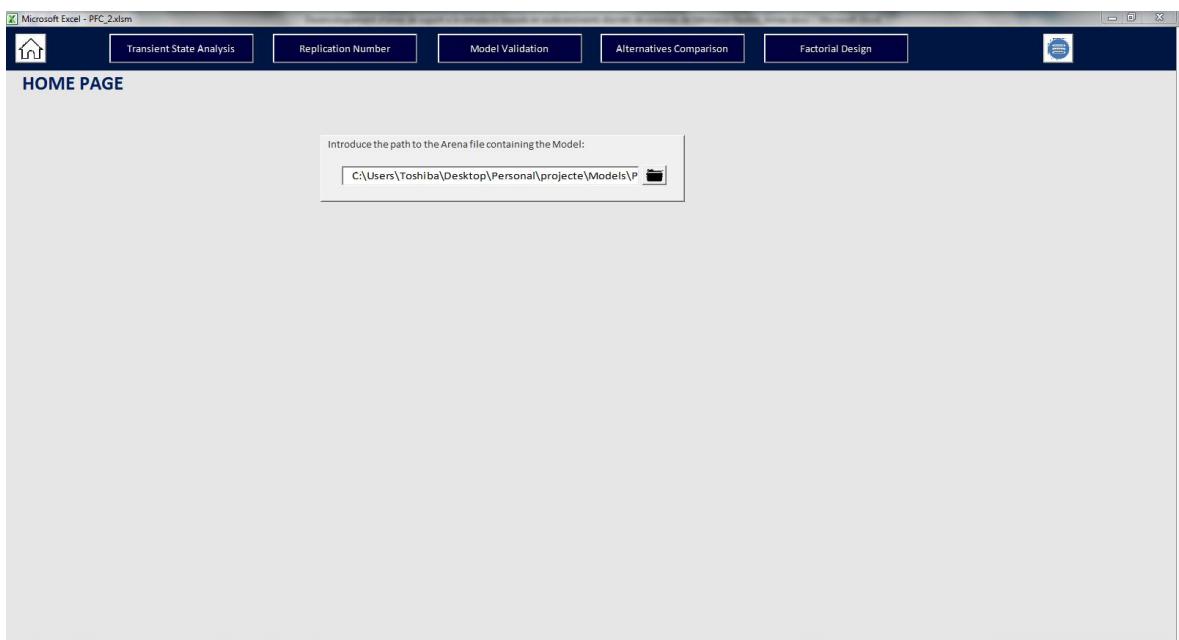


Fig. A1.1. Pantalla principal

El primer que cal fer és indicar la ruta on es troba el fitxer Arena que conté el model amb el què es treballarà. Per fer-ho, es disposa d'un botó que obre l'explorador d'arxius, i filtra els que tenen extensió .doe.

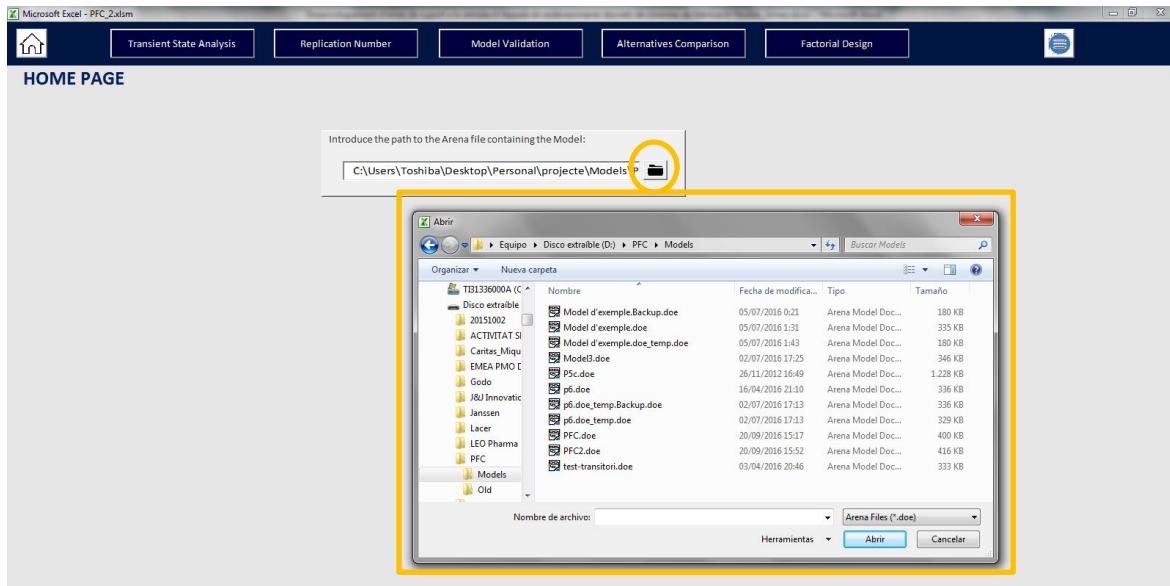


Fig. A1.2. Accés a l'explorador d'arxius

Si s'executa una anàlisi sense haver indicat prèviament el document a partir del qual es vol treballar, o si aquest no existeix, el sistema mostrarà un missatge d'error.

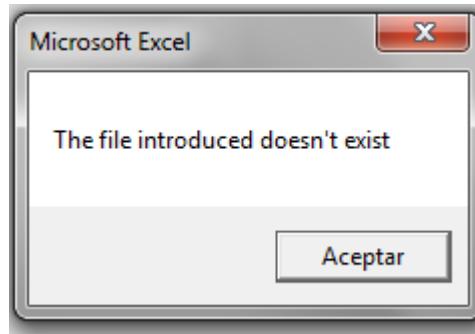


Fig. A1.3. Error mostrat quan el fitxer introduït no és correcte

Un cop seleccionat el fitxer, per poder realitzar un anàlisi cal seleccionar el botó pertinent al panell superior.



Fig. A1.4. Panell de navegació entre les diferents pestanyes

Aquests botons es troben a totes les pestanyes de l'aplicació, i indiquen en cada moment la pantalla que està activa mitjançant un codi de colors (blanc per a la pestanya activa, blau per a la resta).

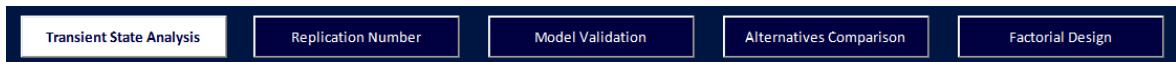


Fig. A1.5. Panell de navegació quan la pestanya activa és la d'anàlisi del transitori

Un altre botó important d'aquesta pestanya és l'encarregat de mostrar o ocultar els menús. Tot i que per utilitzar l'aplicació no és necessari fer ús d'aquests menús, sí que ho és quan es vol introduir alguna modificació en el codi o en la capa de visualització.

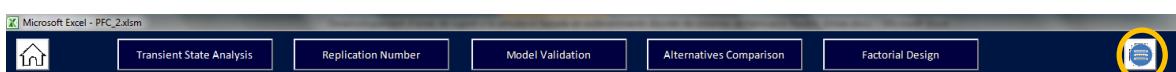


Fig. A1.6. Visió sense la barra de menús d'Excel

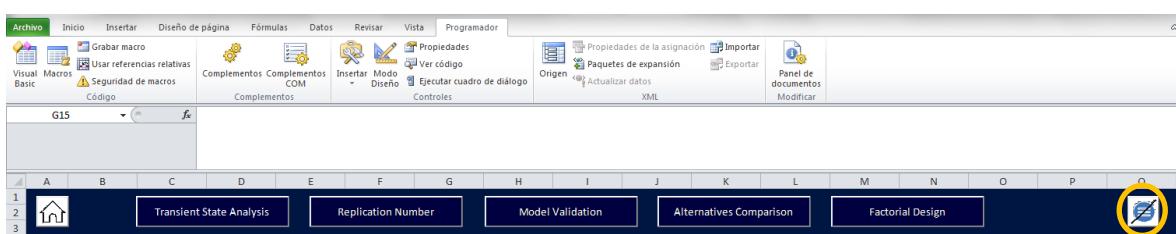


Fig. A1.7. Visió amb la barra de menús d'Excel

Respecte a aquest botó, cal destacar també que abasta tota l'aplicació Excel. És a dir, qualsevol altre full de càcul que estigui obert a la mateixa finestra que aquesta aplicació es veurà afectat per la funció de mostrar o ocultar el menú. Si es vol evitar aquest comportament es pot optar per obrir l'aplicació en una finestra d'Excel separada de la resta de documents.

1.2. Anàlisi del transitori (Transient state analysis)

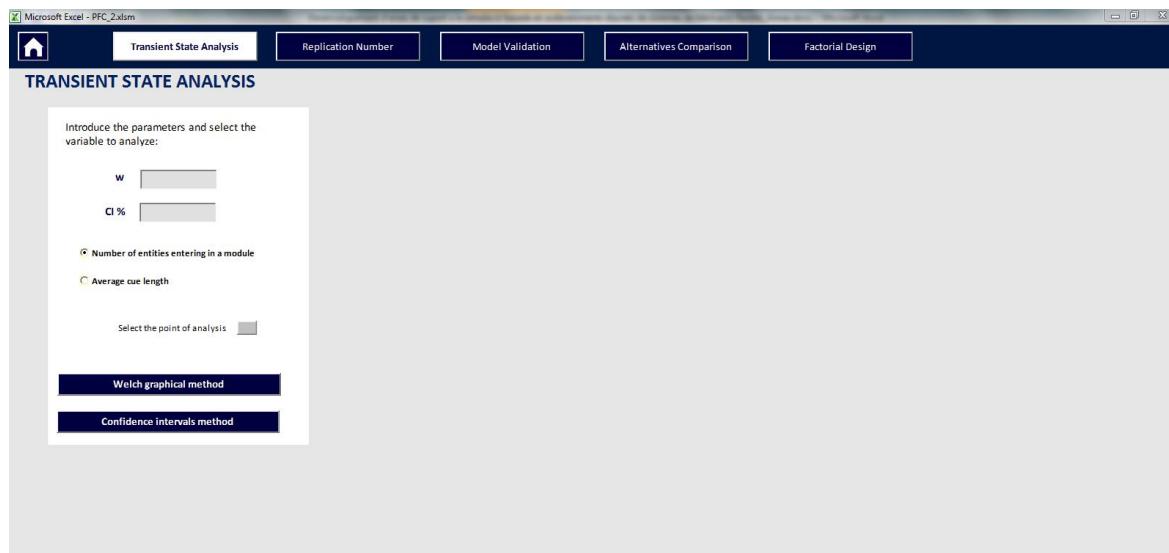


Fig. A1.8. Pantalla d'anàlisi del transitori

Per poder executar aquesta anàlisi cal completar tots els camps que es mostren al panell de definició:

Introduce the parameters and select the variable to analyze:

- 1
- 2
- 3 Number of entities entering in a module
 Average cue length
- 4 Select the point of analysis

Welch graphical method

Confidence intervals method

Fig. A1.9. Panell d'anàlisi del transitori

1. **Paràmetres per a l'anàlisi.** Si es vol realitzar l'anàlisi mitjançant el mètode de Welch, cal introduir un valor adequat pel paràmetre w , mentre que si es vol treballar amb el mètode dels intervals de confiança, el paràmetre necessari és el nivell de confiança. Cal vigilar que els valors introduïts siguin lògics, és a dir, que el percentatge de confiança estigui comprès entre 0 i 1, i que la finestra de temps (w) no superi la meitat del nombre d'observacions.

Pel que fa a la lògica dels valors introduïts, cal fer notar també que el format de número ha de coincidir amb el propi de l'usuari. És a dir, si la configuració numèrica de l'Excel es basa en el sistema europeu, un valor de confiança de 0,95 serà correcte, mentre que si s'escriu 0.95, el sistema generarà un error. En cas que la configuració sigui americana, això succeirà a l'inrevés.

2. **Variable de sortida.** Cal seleccionar si es vol que la variable a analitzar sigui un càcul del nombre d'entitats que entren a un mòdul o del nombre mitjà d'entitats que hi ha entre dos mòduls.
3. **Panell de selecció dels mòduls.** En prémer el botó s'obre un nou panell que té la funció de definir el bloc del model que es prendrà de referència per al càlcul de la variable de sortida.

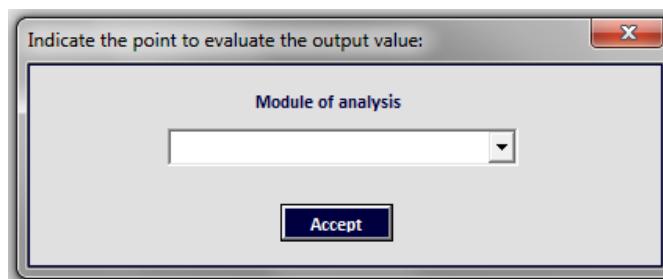


Fig. A1.10. Formulari per indicar el mòdul d'anàlisi

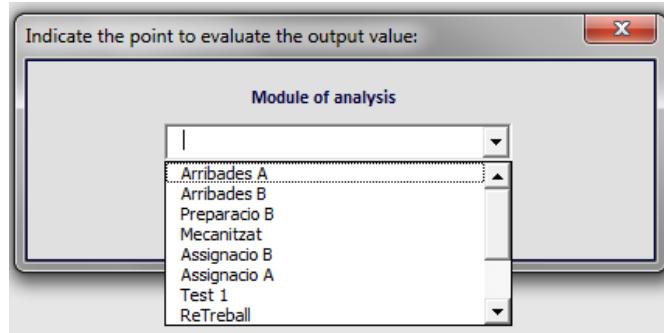


Fig. A1.11. Desplegable amb el llistat de blocs del model

En general, el sistema admet qualsevol mòdul per al càlcul d'entitats que hi entren. En el cas de la longitud de la cua, però, cal evitar els blocs de tipus *decide*, ja que generen un conflicte en no conèixer el flux d'entitats sobre el qual s'ha de fer el càlcul (*true* o *false*). A banda d'aquesta excepció, l'únic que cal tenir en compte és que el punt d'anàlisi tingui lògica (per exemple, intentar calcular les entrades a un bloc de tipus *create* generaria un error que provocaria la interrupció del procés).

Una altra qüestió important és la identificació del blocs en el model. Com s'ha explicat a la Memòria, els canvis que es realitzen al model agafen com a referència el *tag* de cada bloc. Ara bé, de vegades, quan es copia i s'enganxa un bloc per evitar configurar-lo des de zero la còpia manté el mateix *tag* que el mòdul original. Per assegurar el correcte comportament de l'aplicació, és important assegurar-se que no hi hagi dos mòduls amb el mateix *tag* al model.

4. **Mètode de càlcul.** Una vegada s'han introduït tots els valors i seleccionat les opcions correctament, cal prémer un dels dos botons inferiors segons el mètode amb què es vulgui realitzar l'anàlisi. Si el paràmetre introduït és correcte, es generarà el gràfic corresponent.



Fig. A1.12. Resultat de l'anàlisi del transitori

En canvi, si aquest no té un valor lògic, el sistema mostrerà un missatge d'error.

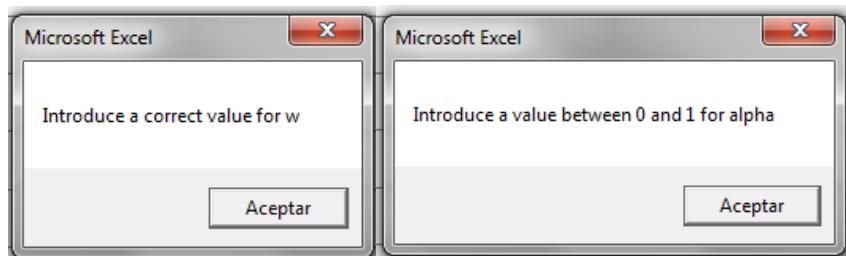


Fig. A1.13. Missatges d'error per a paràmetres incorrectes

1.3. Càlcul del nombre de rèpliques (*Replication number*)

Microsoft Excel - PFC_2.xlsxm

Transient State Analysis Replication Number Model Validation Alternatives Comparison Factorial Design

REPLICATION NUMBER

Introduce a value for the following parameters:

CI %

epsilon

Select a variable to analyze

Replication Number Calculation

Fig. A1.14. Pantalla de càlcul del nombre de rèpliques

En aquest cas, el panell està constituït pels següents elements:

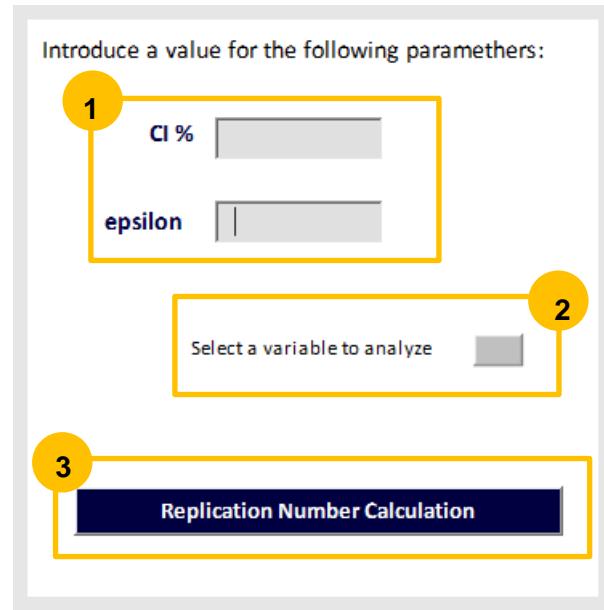


Fig. A1.15. Panell de càlcul del nombre de rèpliques

1. **Paràmetres per a l'anàlisi.** Cal introduir un valor adequat per al percentatge de confiança i per a la meitat de l'amplada de l'interval.
2. **Panell de selecció de la variable de sortida.** En pitjar el botó s'obrirà un nou panell amb totes les variables del model. Per realitzar l'anàlisi, cal triar la variable que es vol prendre com a mesura de comportament del model i prémer acceptar.

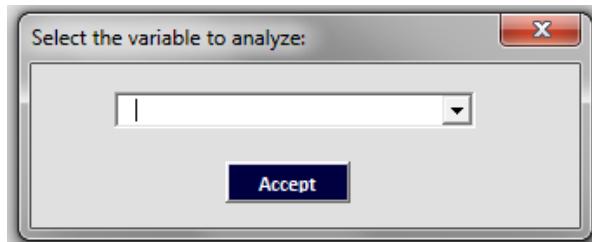


Fig. A1.16. Formulari per indicar la variable de sortida

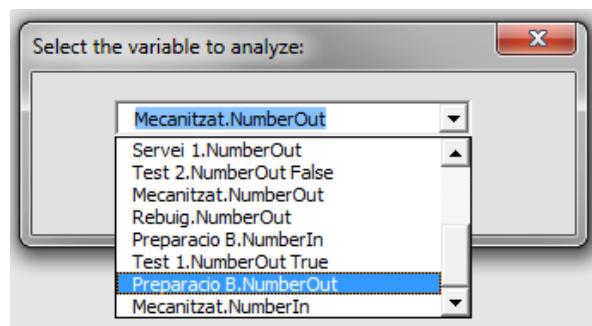


Fig. A1.17. Desplegable amb el llistat de variables del model

3. **Botó d'execució de l'anàlisi.** Una vegada s'han introduït tots els valors i seleccionat la variable, cal prémer el botó que executarà l'anàlisi.

De la mateixa manera que passava en l'anàlisi del transitori, si els paràmetres introduïts no són correctes, el sistema mostrarà un missatge d'error. En canvi, si tots els valors introduïts són correctes, el sistema retorna una imatge amb el resultat obtingut així com el detall de la iteració fins a assolir el resultat desitjat.

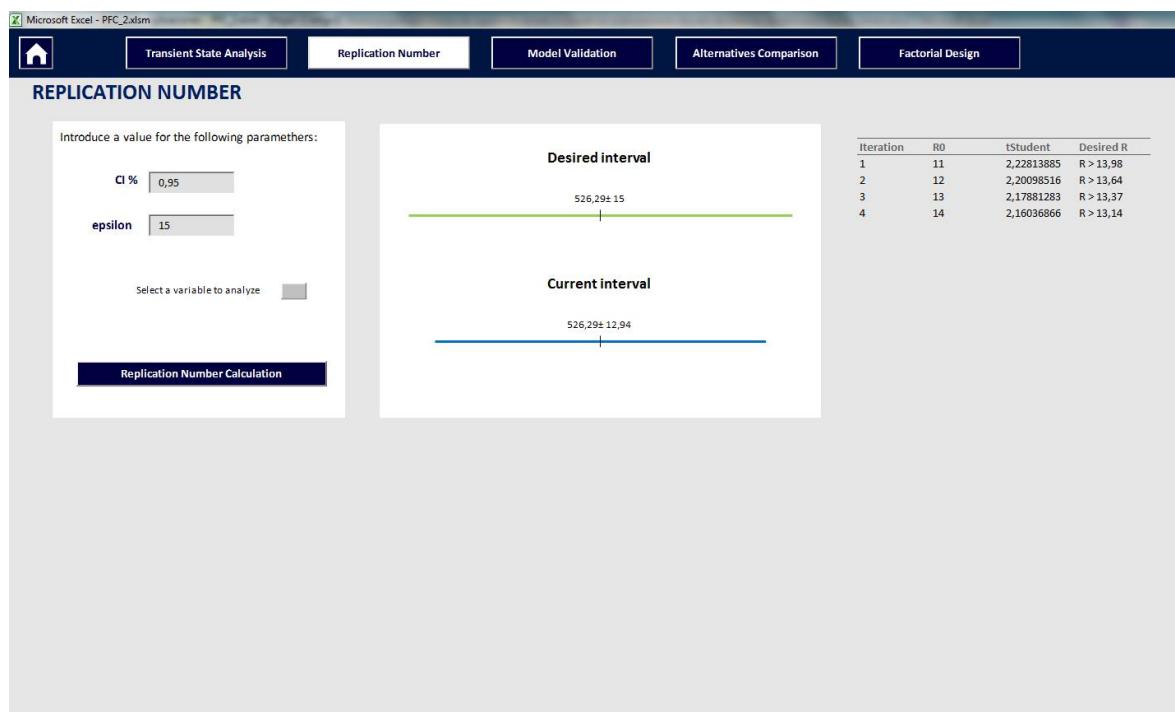


Fig. A1.18. Resultat del càcul del nombre de rèpliques

1.4. Validació del model (*Model validation*)

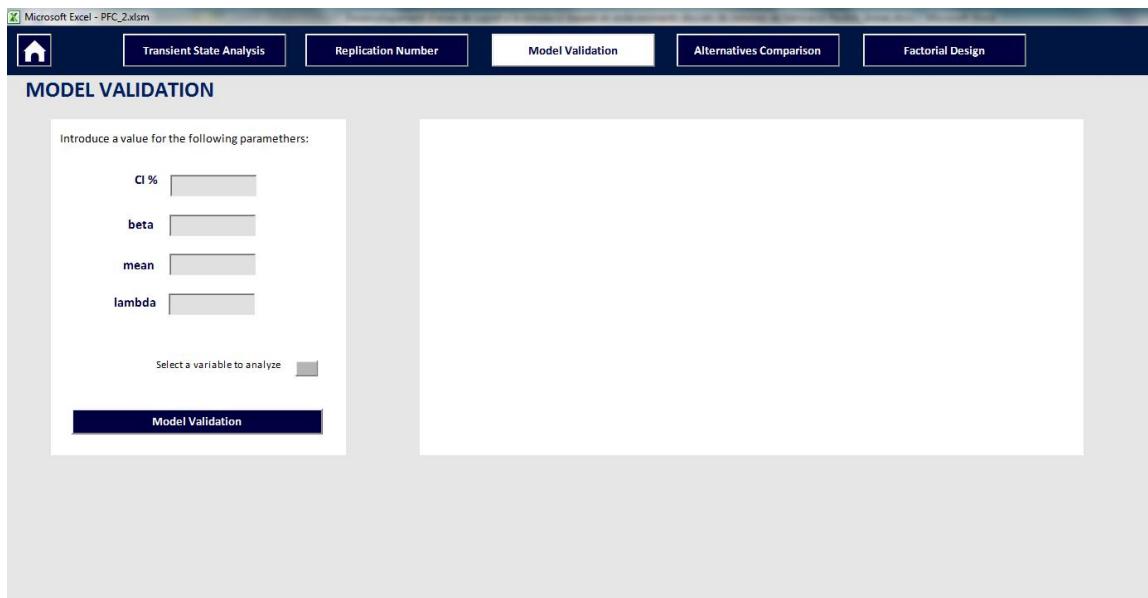


Fig. A1.19. Pantalla de validació del model

Per poder executar aquest anàlisi cal, en primer lloc, indicar els següents paràmetres.

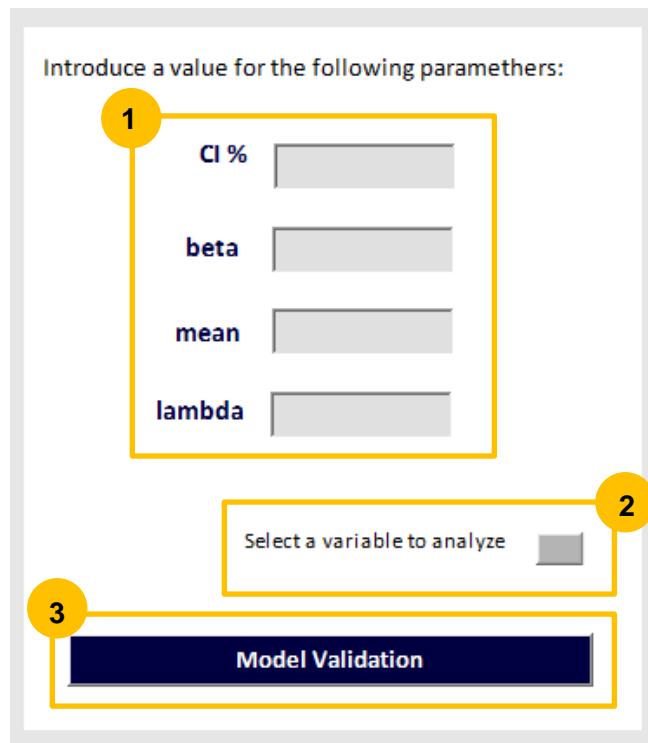


Fig. A1.20. Panell de validació del model

1. **Paràmetres per a l'anàlisi.** Cal introduir un valor adient per al percentatge de confiança, la probabilitat d'estar acceptant un model no vàlid (beta), la mitjana del sistema real i el valor màxim acceptat de la diferència entre la mitjana de la simulació i del sistema real.
2. **Panell de selecció de la variable de sortida.** Igual que en l'anàlisi anterior, aquest botó desplega un formulari amb el llistat de variables del model. Cal seleccionar-ne una i prémer el botó *Accept*.
3. **Botó d'execució de l'anàlisi.** Una vegada s'han introduït tots els valors i seleccionat la variable, aquest és el botó que donarà l'ordre d'executar l'anàlisi. El comportament és el mateix que en els casos anteriors: primerament es comprova que tots els paràmetres siguin correctes i, si és així, es mostren els resultats per pantalla.

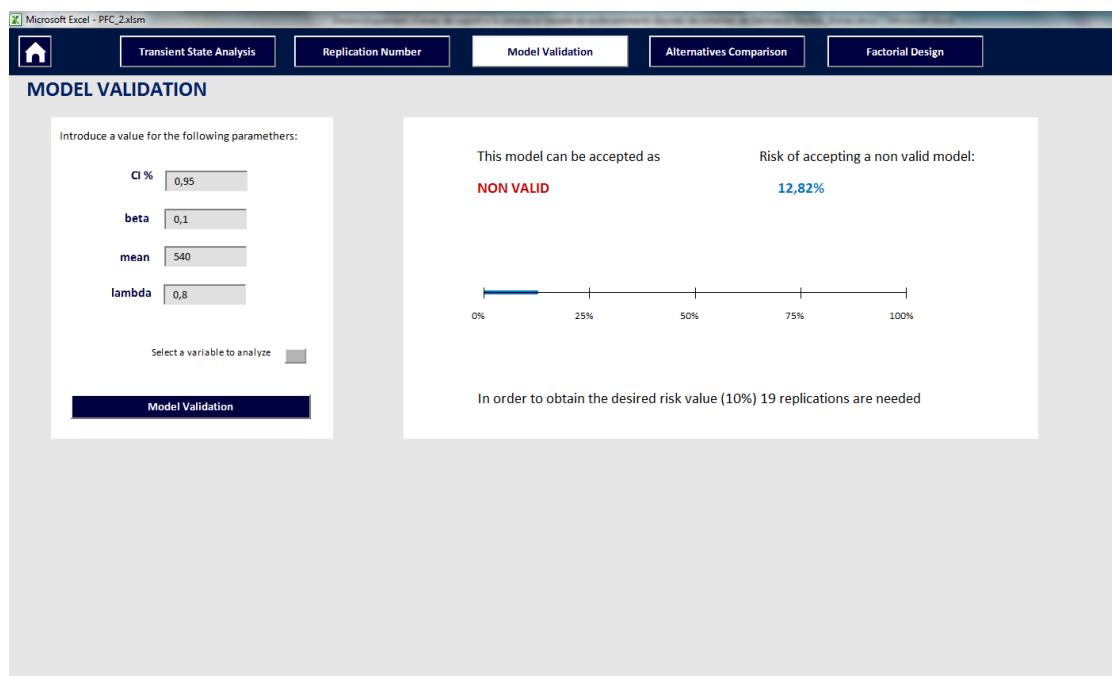


Fig. A1.21. Resultat de la validació del model

1.5. Comparació d'alternatives (*Alternatives comparison*)

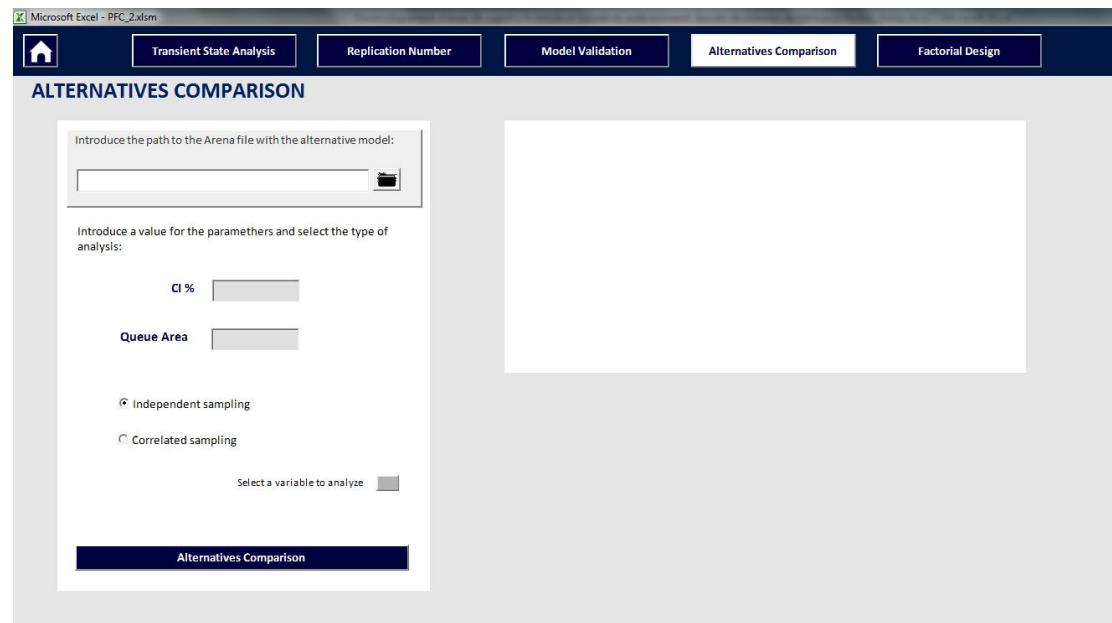


Fig. A1.22. Pantalla de comparació d'alternatives

En aquesta pestanya, el panell de definició de l'experiment inclou els elements següents:



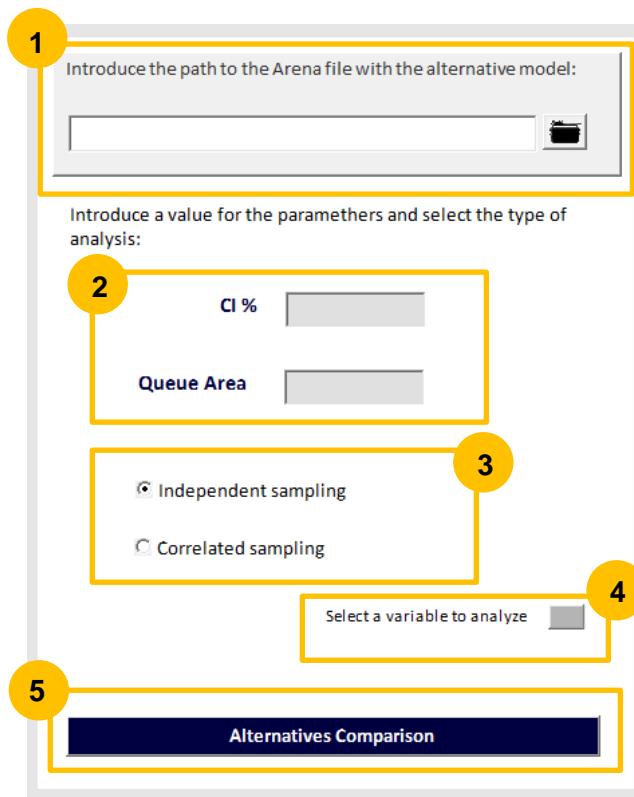


Fig. A1.23. Panell de comparació d'alternatives

1. **Cercador de fitxers.** Aquest element és idèntic al que es troba a la pantalla principal. L'objectiu és poder seleccionar el fitxer que conté el model que es vol comparar amb el de la pestanya *Main*.
2. **Paràmetres per l'anàlisi.** Un cop més cal indicar un valor entre 0 i 1 per al nivell de confiança. L'àrea de cua només es necessària en cas d'estar treballant amb mostreig independent.
3. **Tipus de mostreig.** Tal com s'explica a la memòria adjunta, el desenvolupament matemàtic per saber si dos models són significativament diferents varia en funció de si es treballa amb mostreig independent o amb mostreig correlat. És per això que cal indicar-ho en els selectoris habilitats abans d'executar l'experiment.
4. **Panell de selecció de la variable de sortida.** El funcionament és el mateix que en les analisis descrites anteriorment.
5. **Botó d'execució de l'anàlisi.** De la mateixa manera que passava en els casos anteriors, aquest botó executa l'anàlisi si els paràmetres introduïts són correctes, i

genera un error en cas contrari. Els resultats apareixen a la pantalla de la forma que es mostra a continuació:

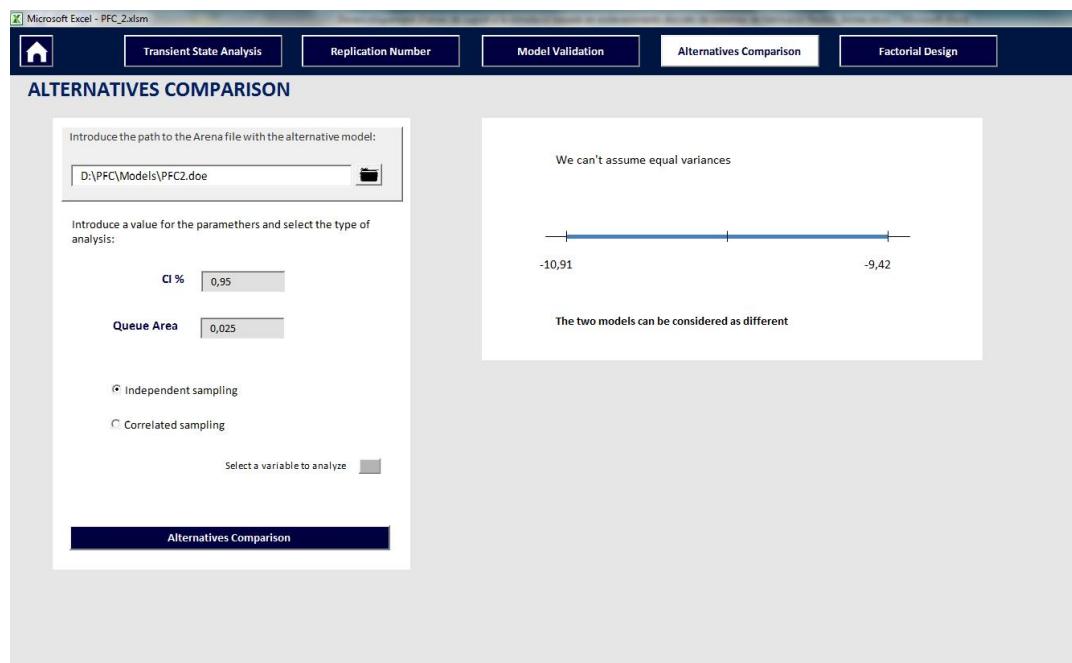


Fig. A1.24. Resultat de la comparació d'alternatives

1.6. Disseny factorial (*Factorial Design*)

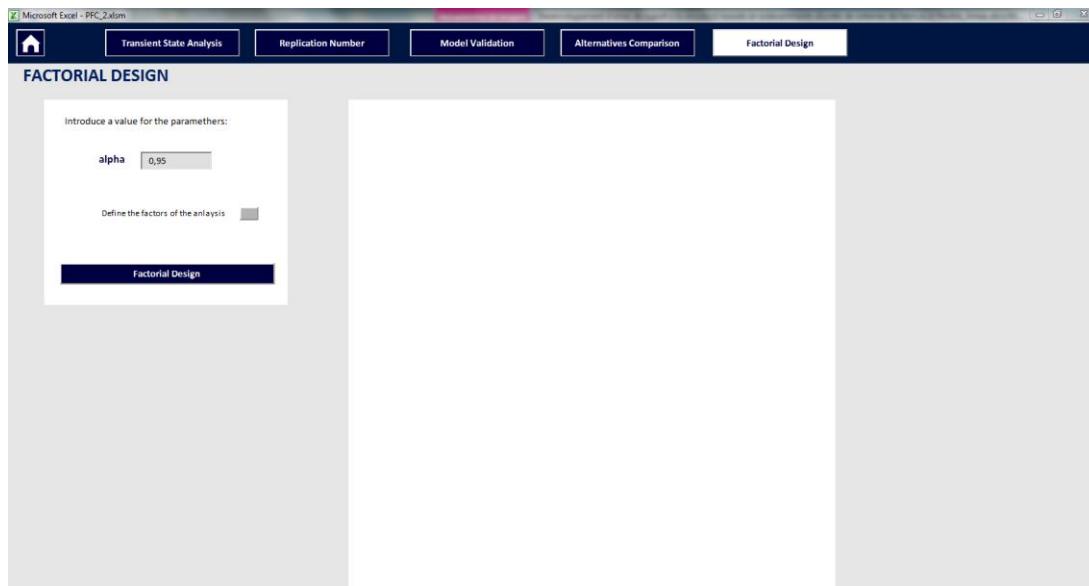


Fig. A1.25. Pantalla de disseny factorial

El panell de disseny factorial consta de tres elements:

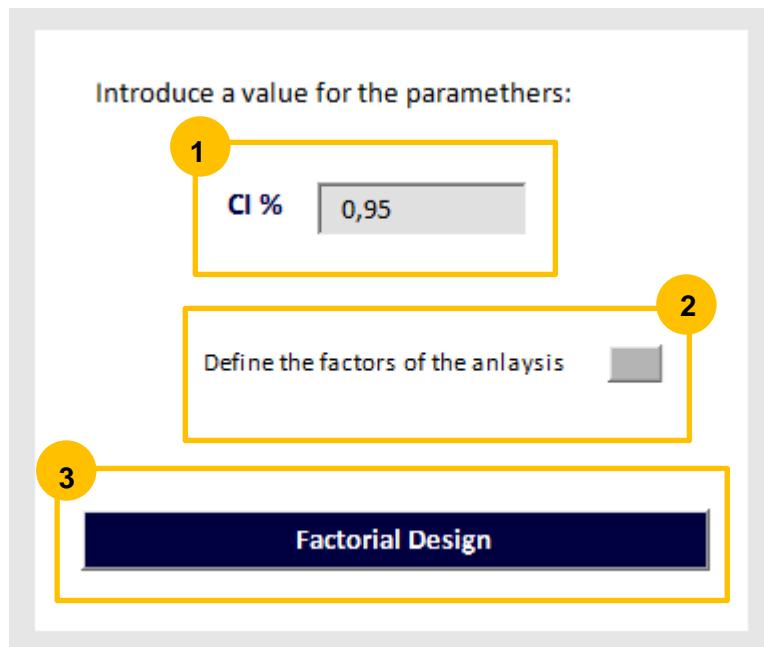


Fig. A1.26. Panell de disseny factorial

1. **Paràmetres per a l'anàlisi.** En aquest cas només es necessita fixar el percentatge de confiança.
2. **Panell de definició dels factors.** Per evitar la realització de múltiples models en Arena s'ha habilitat aquest editor que simplifica en gran mesura la definició dels diferents factors que es volen analitzar.

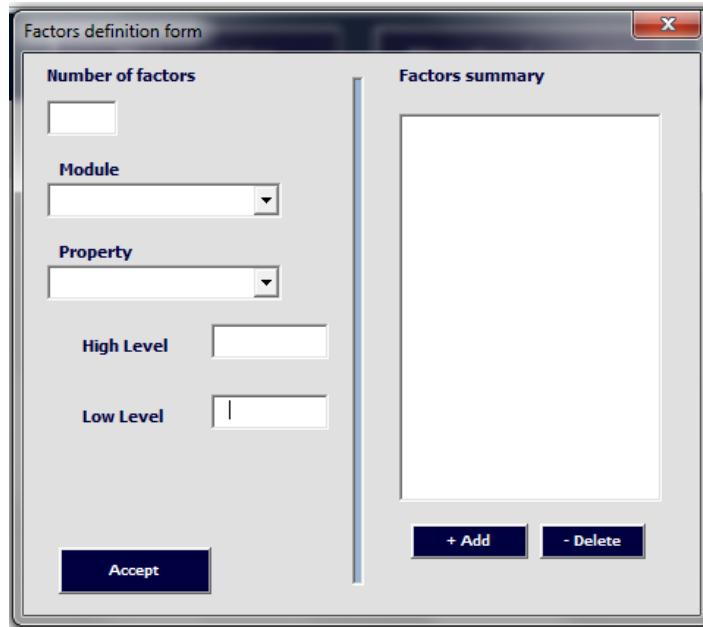


Fig. A1.27. Panell d'edició dels factors

A la part esquerra d'aquest panell es configuren les diferents variables que defineixen completament un factor: bloc, propietat, nivell alt i nivell baix. En prémer el botó **+Add**, el factor definit passa a ser considerat per l'experiment i es mostra resumit a la dreta del panell.

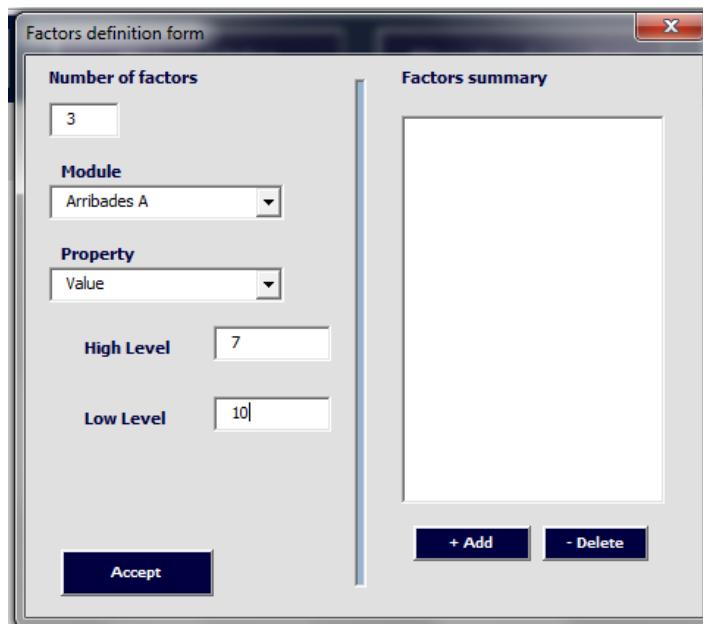


Fig. A1.29. Exemple de definició d'un factor

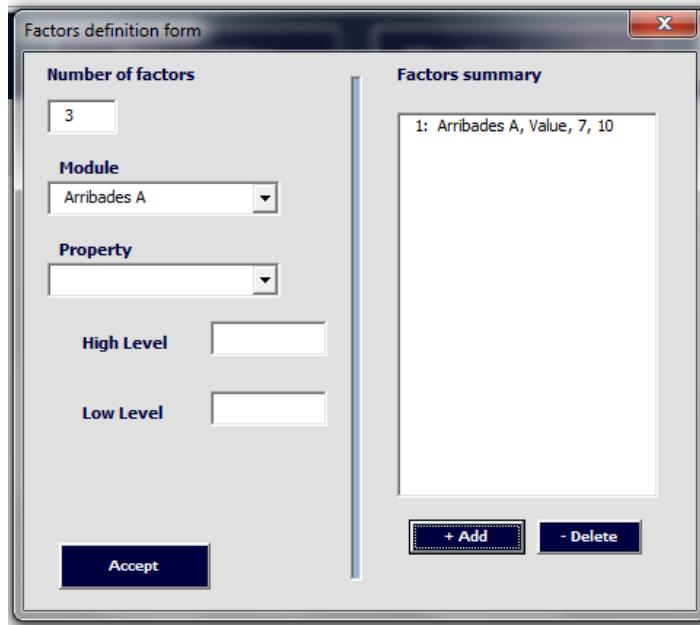


Fig. A1.30. Panell després d'afegir un factor

De la mateixa manera, es pot eliminar un factor de l'experiment seleccionant-lo a la part dreta del panell i prement el botó *Delete*.

Un cop s'han afegit tots els factors, cal prémer el botó *Accept*. És important que el nombre de factors introduït al camp de l'esquerra coincideixi amb el recompte de factors del panell de la dreta. Si això no es compleix, es mostra un missatge d'error que ho adverteix.

3. **Botó d'execució de l'anàlisi.** Com en la resta d'anàlisis, si s'ha complimentat tot el panell de forma correcta, es plasmaran els resultats de forma gràfica.

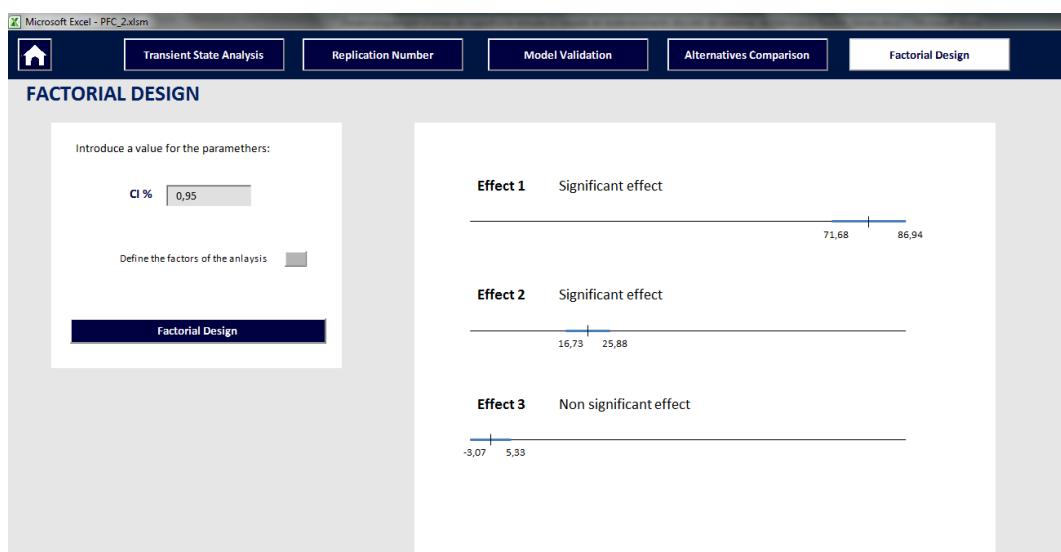


Fig. A1.24. Resultat del disseny factorial

1.7. Índex d'errors

| Codi | Pantalla | Descripció | Missatge |
|------|--------------------------------|--|---|
| 1 | Càlcul del nombre de rèpliques | Valor incorrecte per a l'amplada de l'interval de confiança. | <i>The value for epsilon is not correct</i> |
| 2 | Comparació d'alternatives | L'arxiu de la primera alternativa no existeix. | <i>The file introduced for the first model doesn't exist</i> |
| 3 | Comparació d'alternatives | L'arxiu de la segona alternativa no existeix. | <i>The file introduced for the second model doesn't exist</i> |
| 4 | Determinació del transitori | Valor incorrecte per a la finestra de temps del gràfic de Welch. | <i>Introduce a correct value for w</i> |
| 5 | Determinació del transitori | No s'ha triat el mòdul que serà el focus de l'anàlisi. | <i>The point of analysis must be selected</i> |
| 6 | Disseny Factorial | El nombre de factors introduït no coincideix amb els factors llistats. | <i>The number of factors isn't correct</i> |
| 7 | Validació del model | Valor incorrecte per al paràmetre beta. | <i>The value for beta must be between 0 and 1</i> |
| 8 | Vàries | Valor incorrecte per a l'interval de confiança. | <i>The value for the confidence interval must be between 0 and 1</i> |
| 9 | Vàries | Arxiu no existent. | <i>The file introduced doesn't exist</i> |
| 10 | Vàries | Variable de sortida o paràmetres sense completar. | <i>Select a variable for the analysis and indicate the parameters</i> |
| 11 | Vàries | Falta introduir algun paràmetre. | <i>You must introduce a valid value for all fields</i> |
| 12 | Vàries | No s'ha indicat la variable de sortida. | <i>Select the variable for the analysis</i> |

2. Codi de l'aplicació

2.1. Objectes de Microsoft Excel

2.1.1. Main

```
1 Option Explicit
2 Dim FilePath As String
3
4 'Function to open the file explorer. Only .doe files can be opened
5 Private Sub Select_File_Button_Click()
6 FilePath = Application.GetOpenFilename("Arena Files (*.doe),
*.doe")
7 If (FilePath <> "False") Then
8 FilePath_TextBox.Value = FilePath
9 End If
10 End Sub
11
12 'Return to excel tradicional view
13 Private Sub Show_Menu_Button_Click()
14 Application.ExecuteExcel4Macro "SHOW.TOOLBAR(""Ribbon"",True)"
15 Application.DisplayFormulaBar = True
16 Application.DisplayStatusBar = True
17
18 ActiveWindow.DisplayWorkbookTabs = True
19 ActiveWindow.DisplayHeadings = True
20 ActiveWindow.DisplayHorizontalScrollBar = True
21 ActiveWindow.DisplayVerticalScrollBar = True
22
23 Hide_Menu_Button.Visible = True
24 Show_Menu_Button.Visible = False
25
26 End Sub
27
28 'Set app-style view
29 Private Sub Hide_Menu_Button_Click()
30 Application.ExecuteExcel4Macro "SHOW.TOOLBAR(""Ribbon"",False)"
31 Application.DisplayFormulaBar = False
32 Application.DisplayStatusBar = False
33 ActiveWindow.DisplayWorkbookTabs = False
34 ActiveWindow.DisplayHeadings = False
35 ActiveWindow.DisplayHorizontalScrollBar = False
36 ActiveWindow.DisplayVerticalScrollBar = False
37
38 Hide_Menu_Button.Visible = False
39 Show_Menu_Button.Visible = True
40 End Sub
41
42 'Hide cell headings
43 Private Sub Worksheet_Activate()
44 ActiveWindow.DisplayHeadings = False
45 End Sub
46
47 'Navigation between sheets
```

```

48 'Navigate to transient state analysis screen
49 Private Sub TransientStateAnalysis_Button_Click()
50 ThisWorkbook.Worksheets("Transient State Analysis").Activate
51 End Sub
52
53 'Navigate to alternatives comparison screen
54 Private Sub AlternativesComparison_Button_Click()
55 ThisWorkbook.Worksheets("Alternatives Comparison").Activate
56 End Sub
57
58 'Navigate to factorial design screen
59 Private Sub FactorialDesign_Button_Click()
60 ThisWorkbook.Worksheets("Factorial Design").Activate
61 End Sub
62
63 'Navigate to replication number screen
64 Private Sub ReplicationNumber_Button_Click()
65 ThisWorkbook.Worksheets("Replication Number").Activate
66 End Sub
67
68 'Navigate to model validation screen
69 Private Sub ModelValidation_Button_Click()
70 ThisWorkbook.Worksheets("Model Validation").Activate
71 End Sub

```

2.1.2. Transient State Analysis

```

1 Option Explicit
2 Dim alpha As Double
3 Dim i, w As Integer
4 Dim vModule As String
5
6 'Navigation between sheets
7
8 'Navigate to alternatives comparison screen
9 Private Sub TSA_GotoAC_Button_Click()
10 ThisWorkbook.Worksheets("Alternatives Comparison").Activate
11 End Sub
12
13 'Navigate to factorial design screen
14 Private Sub TSA_GotoFD_Button_Click()
15 ThisWorkbook.Worksheets("Factorial Design").Activate
16 End Sub
17
18 'Navigate to replication number screen
19 Private Sub TSA_GotoRN_Button_Click()
20 ThisWorkbook.Worksheets("Replication Number").Activate
21 End Sub
22
23 'Navigate to model validation screen
24 Private Sub TSA_GotoMV_Button_Click()
25 ThisWorkbook.Worksheets("Model Validation").Activate
26 End Sub
27
28 'Navigate to home screen
29 Private Sub TSA_Home_Button_Click()

```

```

30     ThisWorkbook.Worksheets("Main").Activate
31 End Sub
32
33 'Validation of the analysis definitions and execution of the
34 'analysis by using confidence intervals method
35 Public Sub TSA_IC_Button_Click()
36 If (Hoja2.TSA_alpha_TextBox <> Empty And TSA_alpha_TextBox.Value <
37   1 And TSA_alpha_TextBox.Value > 0) Then
38   alpha = Hoja2.TSA_alpha_TextBox.Value
39   Call IC(alpha) 'Module 4
40 Else
41   MsgBox "The value for the confidence interval must be between 0 and
42   1"
43 End If
44 End Sub
45
46 'Show the module selection form and read the possible modules from
47 'the current model
48 Private Sub TSA_ModuleSelection_Button_Click()
49
50 'Generate a copy of the file introduced on "Intro" sheet to do the
51 'changes in the model
52 vFilePath = Hoja1.FilePath_TextBox.Value
53
54 'Check if the file exists
55 If (Dir(vFilePath) = "") Then
56   MsgBox "The file introduced doesn't exist"
57 Else
58   Set ArenaApp = CreateObject("Arena.Application")
59   If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
60     "_temp.doe") <> "") Then
61     ArenaApp.Models.Close
62   End If
63   Set Model = ArenaApp.Models.Open(vFilePath)
64   Set vSIMAN = Model.SIMAN
65   Set vView = Model.ActiveView
66
67   'Read the modules and add them to the pick list
68   TSA_Definition_Form.TSA_Module1Selection_ComboBox.Clear
69   For i = 1 To Model.Modules.Count
70     If (Model.Modules(i).Shape.Visible) Then
71       vModule = Model.Modules(i).Caption
72       TSA_Definition_Form.TSA_Module1Selection_ComboBox.AddItem (vModule)
73     End If
74   Next i
75   Model.Close
76   TSA_Definition_Form.Show
77 End If
78 End Sub
79
80 'Validation of the analysis definitions and execution of the
81 'analysis by using Welch method
82 Public Sub TSA_welch_Button_Click()
83 If (Hoja2.TSA_w_TextBox <> Empty And TSA_w_TextBox.Value > 1) Then
84   w = Hoja2.TSA_w_TextBox.Value
85   Call Welch(w) 'Module 3
86 Else
87   MsgBox "Introduce a correct value for w"

```

```

81    End If
82    End Sub
83
84    'Set app view
85    Private Sub Worksheet_Activate()
86        ActiveWindow.DisplayHeadings = False
87    End Sub

```

2.1.3. Replication Number

```

1     Option Explicit
2     Dim IC, epsilon As Double
3     Dim i As Integer
4     Dim VariablesList, OPWFile, TextAux, VariablesAux As String
5     Dim IniVar, EndVar As Boolean
6
7     'Code to read the models variables and add them to
8     'AC_OutputVariableSelection_Form
9     Private Sub RN_VariableSelection_Button_Click()
10
11    'Check if the file exists
12    If (Dir(vFilePath) = "") Then
13        MsgBox "The file introduced doesn't exist"
14    Else
15        'Generate a copy of the file introduced on "Intro" sheet to do the
16        'changes in the model
17        vFilePath = Hoja1.FilePath_TextBox.Value
18        Set ArenaApp = CreateObject("Arena.Application")
19        If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
20            "_temp.doe") <> "") Then
21            ArenaApp.Models.Close
22        End If
23        Set Model = ArenaApp.Models.Open(vFilePath)
24        Model.SaveAs (vFilePath & "_temp.doe")
25        Model.Close
26        Set Model = ArenaApp.Models.Open(vFilePath & "_temp.doe")
27        Set vSIMAN = Model.SIMAN
28        Set vView = Model.ActiveView
29
30        'Run the simulation
31        Model.BatchMode = True
32        Model.QuietMode = True
33        Model.PauseBetweenReplications = False
34        Model.Go
35
36        'Open the .opw file to read the variables
37        OPWFile = vFilePath & "_temp.opw"
38        Open OPWFile For Input As #1
39        IniVar = False
40        EndVar = False
41
42        'Read the variables of the model
43        Do Until EOF(1) Or EndVar = True
44        Line Input #1, TextAux
45        If (IniVar = True And InStr(TextAux, "[") = 1) Then
46            EndVar = True
47        End If

```

```

45  If (IniVar = True And EndVar = False) Then
46  VariablesAux = VariablesAux & "/" & TextAux
47  End If
48  If (TextAux = "[VARIABLES]") Then
49  IniVar = True
50  End If
51  Loop
52  Close #1
53  VariablesList = Split(VariablesAux, "/")
54  For i = 1 To UBound(VariablesList)
55  With
56  RN_OutputVariableSelection_Form.RN_OutputVariable.ComboBox.AddItem
57  VariablesList(i)
58  End With
59  Next i
60  Model.Close
61  FileSystem.Kill (vFilePath & "_temp*")
62  RN_OutputVariableSelection_Form.Show
63  End If
64  End Sub
65
66  'Navigate to transient state analysis screen
67  Private Sub RN_Goto_TSA_Button_Click()
68  ThisWorkbook.Worksheets("Transient State Analysis").Activate
69  End Sub
70
71  'Navigate to alternatives comparison screen
72  Private Sub RN_GotoAC_Button_Click()
73  ThisWorkbook.Worksheets("Alternatives Comparison").Activate
74  End Sub
75
76  'Navigate to factorial design screen
77  Private Sub RN_GotoFD_Button_Click()
78  ThisWorkbook.Worksheets("Factorial Design").Activate
79  End Sub
80
81  'Navigate to model validation screen
82  Private Sub RN_GotoMV_Button_Click()
83  ThisWorkbook.Worksheets("Model Validation").Activate
84  End Sub
85
86  'Navigate to home screen
87  Private Sub RN_Home_Button_Click()
88  ThisWorkbook.Worksheets("Main").Activate
89  End Sub
90
91  'Check the parameters and execute the analysis
92  Public Sub RN_Button_Click()
93  If (Hoja3.RN_CI_Textbox <> Empty And Hoja3.RN_Epsilon_TextBox <>
Empty) Then
94  If (Hoja3.RN_CI_Textbox < 0 Or Hoja3.RN_CI_Textbox > 1) Then
95  MsgBox "The value for the confidence interval must be between 0 and
1"
96  ElseIf (Hoja3.RN_Epsilon_TextBox < 0) Then
97  MsgBox "The value for epsilon is not correct"
98  Else
IC = Hoja3.RN_CI_Textbox.Value
epsilon = Hoja3.RN_Epsilon_TextBox.Value

```

```

99 Call N_Rep(IC, epsilon)
100 End If
101 Else
102 'Error message if variable boxes are empty
103 MsgBox "Select a variable for the analysis and indicate the
parameters"
104 End If
105 End Sub
106
107 ' Hyde cells heading
108 Private Sub Worksheet_Activate()
109 ActiveWindow.DisplayHeadings = False
110 End Sub

```

2.1.4. Model Validation

```

1 Option Explicit
2 Dim CI, beta_des, mean, lambda As Double
3 Dim i As Integer
4 Dim VariablesList, OPWFile, TextAux, VariablesAux As String
5 Dim IniVar, EndVar As Boolean
6
7 'Code to read the models variables and add them to
AC_OutputVariableSelection_Form
8 Private Sub MV_VariableSelection_Button_Click()
9
10 'Check if the file exists
11 If (Dir(vFilePath) = "") Then
12 MsgBox "The file introduced doesn't exist"
13 Else
14 'Generate a copy of the file introduced on "Intro" sheet to do the
changes in the model
15 vFilePath = Hoja1.FilePath_TextBox.Value
16 Set ArenaApp = CreateObject("Arena.Application")
17 If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
"_temp.doe") <> "") Then
18 ArenaApp.Models.Close
19 End If
20 Set Model = ArenaApp.Models.Open(vFilePath)
21 Model.SaveAs (vFilePath & "_temp.doe")
22 Model.Close
23 Set Model = ArenaApp.Models.Open(vFilePath & "_temp.doe")
24 Set vSIMAN = Model.SIMAN
25 Set vView = Model.ActiveView
26
27 'Run the simulation
28 Model.BatchMode = True
29 Model.QuietMode = True
30 Model.PauseBetweenReplications = False
31 Model.Go
32
33 'Open the .opw file to read the variables
34 OPWFile = vFilePath & "_temp.opw"
35 Open OPWFile For Input As #1
36 IniVar = False
37 EndVar = False

```

```

38     'Read the variables of the model
39     Do Until EOF(1) Or EndVar = True
40     Line Input #1, TextAux
41     If (IniVar = True And InStr(TextAux, "[") = 1) Then
42     EndVar = True
43     End If
44     If (IniVar = True And EndVar = False) Then
45     VariablesAux = VariablesAux & "/" & TextAux
46     End If
47     If (TextAux = "[VARIABLES]") Then
48     IniVar = True
49     End If
50     Loop
51     Close #1
52     VariablesList = Split(VariablesAux, "/")
53     For i = 1 To UBound(VariablesList)
54     With
55     MV_OutputVariableSelection_Form.MV_OutputVariable_ComboBox.AddItem
56     VariablesList(i)
57     End With
58     Next i
59     Model.Close
60     FileSystem.Kill (vFilePath & "_temp*")
61     MV_OutputVariableSelection_Form.Show
62     End If
63     End Sub
64
65     'Check the parameters and execute the analysis
66     Public Sub MV_Analysis_Button_Click()
67     If (MV_CI_Textbox.Value <> Empty And MV_beta_Textbox.Value <> Empty
68     And MV_mean_Textbox.Value <> Empty And MV_lambda_Textbox.Value <>
69     Empty) Then
70     If (MV_CI_Textbox.Value > 1 Or MV_CI_Textbox.Value < 0) Then
71     MsgBox "The value for the confidence interval must be between 0 and
72     1"
73     ElseIf (MV_CI_Textbox.Value = Empty Or MV_CI_Textbox.Value > 1 Or
74     MV_CI_Textbox.Value < 0) Then
75     MsgBox "The value for beta must be between 0 and 1"
76     End If
77     CI = MV_CI_Textbox.Value
78     beta_des = MV_beta_Textbox.Value
79     mean = MV_mean_Textbox.Value
80     lambda = MV_lambda_Textbox.Value
81     Call Model_Validation(CI, beta_des, mean, lambda)
82     Else
83     MsgBox "You must introduce a valid value for all fields"
84     End If
85     End Sub
86
87     'Navigate to transient state analysis screen
88     Private Sub MV_GotoTSA_Button_Click()
89     ThisWorkbook.Worksheets("Transient State Analysis").Activate
90     End Sub
91
92     'Navigate to alternatives comparison screen
93     Private Sub MV_GotoAC_Button_Click()
94     ThisWorkbook.Worksheets("Alternatives Comparison").Activate

```

```

89 End Sub
90
91 'Navigate to factorial design screen
92 Private Sub MV_GotoFD_Button_Click()
93 ThisWorkbook.Worksheets("Factorial Design").Activate
94 End Sub
95
96 'Navigate to replication number screen
97 Private Sub MV_GotoRN_Button_Click()
98 ThisWorkbook.Worksheets("Replication Number").Activate
99 End Sub
100
101 'Navigate to home screen
102 Private Sub MV_Home_Button_Click()
103 ThisWorkbook.Worksheets("Main").Activate
104 End Sub
105
106 'Hide cells heading
107 Private Sub Worksheet_Activate()
108 ActiveWindow.DisplayHeadings = False
109 End Sub

```

2.1.5. Alternatives Comparison

```

1 Option Explicit
2 Dim i, n As Integer
3 Dim vModule As String
4 Dim IniVar, EndVar As Boolean
5 Dim ArenaApp As Arena.Application
6 Dim nModel, nOperand As Integer
7 Dim VariablesList, OPWFile, TextAux, VariablesAux, vOperand,
8 FilePath As String
9
10 'Check the parameters and execute the analysis
11 Private Sub AC_Analysis_Button_Click()
12 If (AC_CI_TextBox = Empty Or AC_Alpha_TextBox = Empty) Then
13 MsgBox "You must introduce a valid value for all fields"
14 ElseIf (AC_CI_TextBox > 1 Or AC_CI_TextBox < 0) Then
15 MsgBox "The value for the confidence interval must be between 0 and
16 1"
17 ElseIf (AC_Alpha_TextBox > 1 Or AC_Alpha_TextBox < 0) Then
18 MsgBox "The value for the queue area must be between 0 and 1"
19 ElseIf AC_CorrelateSample_OptionButton = True Then
20 Call Alternative_Comparison_Correlated
21 Else: AC_OptionButton1 = True
22 Call Alternative_Comparison_Independent
23 End If
24 End Sub
25
26 'Navigate to transient state screen
27 Private Sub AC_GotoTSA_Button_Click()
28 ThisWorkbook.Worksheets("Transient State Analysis").Activate
29 End Sub
30
31 'Navigate to factorial design screen
32 Private Sub AC_GotoFD_Button_Click()
33 ThisWorkbook.Worksheets("Factorial Design").Activate

```

```

32     End Sub
33
34     'Navigate to replication number screen
35     Private Sub AC_GotoRN_Button_Click()
36         ThisWorkbook.Worksheets("Replication Number").Activate
37     End Sub
38
39     'Navigate to model validation screen
40     Private Sub AC_GotoMV_Button_Click()
41         ThisWorkbook.Worksheets("Model Validation").Activate
42     End Sub
43
44     'Navigate to home screen
45     Private Sub AC_Home_Button_Click()
46         ThisWorkbook.Worksheets("Main").Activate
47     End Sub
48
49     'Code to read the models variables and add them to
50     'AC_OutputVariableSelection_Form
51     Private Sub AC_VariableSelection_Button_Click()
52         AC_OutputVariableSelection_Form.AC_OutputVariable.ComboBox.Clear
53
54         'We generate a copy of the file introduced on "Intro" sheet to do
55         'the changes in the model
56         'Check if the file exists
57         If (Dir(vFilePath) = "") Then
58             MsgBox "The file introduced doesn't exist"
59         Else
60             vFilePath = Hoja1.FilePath_TextBox.Value
61             Set ArenaApp = CreateObject("Arena.Application")
62             If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
63             "_temp.doe") <> "") Then
64                 ArenaApp.Models.Close
65             End If
66             Set Model = ArenaApp.Models.Open(vFilePath)
67             Model.SaveAs (vFilePath & "_temp.doe")
68             Model.Close
69             Set Model = ArenaApp.Models.Open(vFilePath & "_temp.doe")
70             Set vSIMAN = Model.SIMAN
71             Set vView = Model.ActiveView
72
73         'We run the simulation
74         Model.BatchMode = True
75         Model.QuietMode = True
76         Model.PauseBetweenReplications = False
77         Model.Go
78
79         'We open the .opw file to read the variables
80         OPWfile = vFilePath & "_temp.opw"
81         Open OPWfile For Input As #1
82         IniVar = False
83         EndVar = False
84
85         'Read the variables of the model
86         Do Until EOF(1) Or EndVar = True
87             Line Input #1, TextAux
88             If (IniVar = True And InStr(TextAux, "[") = 1) Then

```

```

85 EndVar = True
86 End If
87 If (IniVar = True And EndVar = False) Then
88 VariablesAux = VariablesAux & "/" & TextAux
89 End If
90 If (TextAux = "[VARIABLES]") Then
91 IniVar = True
92 End If
93 Loop
94 Close #1
95 VariablesList = Split(VariablesAux, "/")
96
97 'Add the variables to the next form
98 For i = 1 To UBound(VariablesList)
99 With
100 AC_OutputVariableSelection_Form.AC_OutputVariable_ComboBox.AddItem
VariablesList(i)
101 End With
102 Next i
103 Model.Close
104 FileSystem.Kill (vFilePath & "_temp*")
105 AC_OutputVariableSelection_Form.Show
106 End If
107 End Sub
108
109 'Check if the file for the alternative model is correct
110 Private Sub Select_Alternative_File_Button_Click()
111 FilePath = Application.GetOpenFilename("Arena Files (*.doe),
*.doe")
112 If (FilePath <> "False") Then
113 AC_Model2_Path.Value = FilePath
114 End If
115 End Sub
116
117 'Hyde cells headings
118 Private Sub Worksheet_Activate()
119 ActiveWindow.DisplayHeadings = False
120 End Sub

```

2.1.6. Factorial Design

```

1 Option Explicit
2 Dim i, n As Integer
3 Dim vModule As String
4
5 'Check the parameters and execute the analysis
6 Private Sub FD_Analysis_Button_Click()
7 If (FD_alpha_TextBox.Value > 1 Or FD_alpha_TextBox.Value < 0 Or
FD_alpha_TextBox.Value = Empty) Then
8 MsgBox "The value for the confidence interval must be between 0 and
1"
9 Else
10 Call Factorial_Design
11 End If
12 End Sub

```

```
13  Private Sub FD_ExperimentDefinition_Button_Click()
14
15  'We generate a copy of the file introduced on "Intro" sheet to do
16  'the changes in the model
17  'Check if the file exists
18  If (Dir(vFilePath) = "") Then
19    MsgBox "The file introduced doesn't exist"
20  Else
21    vFilePath = Hoja1.FilePath_TextBox.Value
22    Set ArenaApp = CreateObject("Arena.Application")
23    If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
24    "_temp.doe") <> "") Then
25      ArenaApp.Models.Close
26    End If
27    Set Model = ArenaApp.Models.Open(vFilePath)
28    Set vSIMAN = Model.SIMAN
29    Set vView = Model.ActiveView
30
31  'Read the modules and add them to the pick list
32  n = CInt(Model.Modules.Count)
33
34  For i = 1 To n
35    If (Model.Modules(i).Shape.Visible) Then
36      vModule = Model.Modules(i).Caption
37      FD_Definition_Form.FD_ModuleSelection_ComboBox.AddItem (vModule)
38    End If
39  Next i
40  Model.Close
41  FD_Definition_Form.Show
42  End If
43  End Sub
44
45  'Navigate to transient state screen
46  Private Sub FD_GotoTSA_Button_Click()
47    ThisWorkbook.Worksheets("Transient State Analysis").Activate
48  End Sub
49
50  'Navigate to alternatives comparison screen
51  Private Sub FD_GotoAC_Button_Click()
52    ThisWorkbook.Worksheets("Alternatives Comparison").Activate
53  End Sub
54
55  'Navigate to replication number screen
56  Private Sub FD_GotoRN_Button_Click()
57    ThisWorkbook.Worksheets("Replication Number").Activate
58  End Sub
59
60  'Navigate to validation model screen
61  Private Sub FD_GotoMV_Button_Click()
62    ThisWorkbook.Worksheets("Model Validation").Activate
63  End Sub
64
65  'Navigate to home screen
66  Private Sub FD_Home_Button_Click()
67    ThisWorkbook.Worksheets("Main").Activate
```

```

66 End Sub
67
68 'Hyde cells heading
69 Private Sub Worksheet_Activate()
70 ActiveWindow.DisplayHeadings = False
71 End Sub

```

2.1.7. ThisWorkbook

```

1 Option Explicit
2
3 Private Sub Workbook_Open()
4 Hoja1.Activate
5 Hoja1.Hide_Menu_Button.Visible = False
6 Hoja1.Show_Menu_Button.Visible = True
7 Application.ExecuteExcel4Macro "SHOW.TOOLBAR(""Ribbon"",False)"
8 Application.DisplayFormulaBar = False
9 Application.DisplayStatusBar = False
10 ActiveWindow.DisplayWorkbookTabs = False
11 ActiveWindow.DisplayHeadings = False
12 ActiveWindow.DisplayHorizontalScrollBar = False
13 ActiveWindow.DisplayVerticalScrollBar = False
14 End Sub

```

2.2. Formularis

2.2.1. TSA_Definition_Form

```

1 Option Explicit
2
3 'TSA_Definition_Form
4 Public Sub TSA_Accept_Button_Click()
5 'Check if modules are completed
6 If (TSA_Module1Selection_ComboBox.Value <> Empty) Then
7 TSA_Definition_Form.Hide
8 Else
9 MsgBox "The point of analysis must be selected"
10 End If
11 End Sub

```

2.2.2. RN_OutputVariableSelection_Form

```

1 Option Explicit
2
3 Public Sub RN_Accept_Button_Click()
4 If (RN_OutputVariable_ComboBox.Value <> Empty) Then
5 RN_OutputVariableSelection_Form.Hide
6 Else
7 MsgBox "Select the variable for the analysis"
8 End If
9 End Sub

```

2.2.3. MV_OutputVariableSelection_Form

```

1 Option Explicit
2
3 Private Sub MV_Accept_Button_Click()
4 If (MV_OutputVariable_ComboBox.Value <> Empty) Then
5 MV_OutputVariableSelection_Form.Hide
6 Else
7 MsgBox "Select the variable for the analysis"
8 End If
9 End Sub

```

2.2.4. AC_OutputVariableSelection_Form

```

1 Option Explicit
2
3 Private Sub AC_Accept_Button_Click()
4 If (AC_OutputVariable_ComboBox <> Empty) Then
5 AC_OutputVariableSelection_Form.Hide
6 Else
7 MsgBox "Select the variable for the analysis"
8 End If
9 End Sub

```

2.2.5. FD_Definition_Form

```

1 Option Explicit
2 Dim i, NumOfFactors, FactorNumber, ModuleIndex, nOperand As Integer
3 Dim VariablesList, OPWFile, TextAux, VariablesAux, vModule,
vOperand As String
4 Dim IniVar, EndVar As Boolean
5 Dim FactorModule, FactorProperty, FactorHighValue, FactorLowValue
As Variant
6
7 Private Sub FD_Accept_Button_Click()
8 NumOfFactors = FD_NFactors_Textbox.Value
9 If (NumOfFactors <> FD_Experiments_Listbox.ListCount) Then
10 MsgBox "The number of factors isn't correct"
11 Else
12 'Check if the file exists
13 If (Dir(vFilePath) = "") Then
14 MsgBox "The file introduced doesn't exist"
15 Else
16 'Generate a copy of the file introduced on "Intro" sheet to do the
changes in the model
17 vFilePath = Hoja1.FilePath_TextBox.Value
18 Set ArenaApp = CreateObject("Arena.Application")
19 If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
"_temp.doe") <> "") Then
20 ArenaApp.Models.Close
21 End If
22 Set Model = ArenaApp.Models.Open(vFilePath)
23 Model.SaveAs (vFilePath & "_temp.doe")
24 Model.Close
25 Set Model = ArenaApp.Models.Open(vFilePath & "_temp.doe")

```

```

26 Set vSIMAN = Model.SIMAN
27 Set vView = Model.ActiveView
28 'Run the simulation
29 Model.BatchMode = True
30 Model.QuietMode = True
31 Model.PauseBetweenReplications = False
32 Model.Go
33 'Open the .opw file to read the variables
34 OPWFile = vFilePath & "\_temp.opw"
35 Open OPWFile For Input As #1
36 IniVar = False
37 EndVar = False
38 'Read the variables of the model
39 Do Until EOF(1) Or EndVar = True
40 Line Input #1, TextAux
41 If (IniVar = True And InStr(TextAux, "[")) = 1) Then
42 EndVar = True
43 End If
44 If (IniVar = True And EndVar = False) Then
45 VariablesAux = VariablesAux & "/" & TextAux
46 End If
47 If (TextAux = "[VARIABLES]") Then
48 IniVar = True
49 End If
50 Loop
51 Close #1
52 VariablesList = Split(VariablesAux, "/")
53
54 'Add the variables to the next form
55 For i = 1 To UBound(VariablesList)
56 With FD_OutputVariableSelection_Form.FD_OutputVariable_ComboBox
57 .AddItem VariablesList(i)
58 End With
59 Next i
60 Model.Close
61 FileSystem.Kill (vFilePath & "\_temp*")
62 FD_Definition_Form.Hide
63 FD_OutputVariableSelection_Form.Show
64 End If
65 End If
66 End Sub
67
68 'Code to mix the parameters introduced to get the factor definition
69 Public Sub FD_Add_Button_Click()
70 vModule = FD_Definition_Form.FD_ModuleSelection_ComboBox.Value
71 n = CInt(Model.Modules.Count)
72 i = 1
73 While (Model.Modules(i).Caption <> vModule And i <= n)
74 i = i + 1
75 Wend
76
77 ModuleIndex = i
78 FactorNumber = FD_Experiments_Listbox.ListCount + 1
79 FactorModule = Model.Modules(ModuleIndex).Caption
80 FactorProperty =
81 Model.Modules(ModuleIndex).Definition.Operands(FD_ParameterSelectio
n.ComboBox.ListIndex + 1).Name
82 FactorHighValue = FD_High_TextBox.Value

```

```

82     FactorLowValue = FD_Low_TextBox.Value
83     FD_Experiments_Listbox.AddItem (FactorNumber & ":" & FactorModule
84     & ", " & FactorProperty & ", " & FactorHighValue & ", " &
85     FactorLowValue)
86
87     FD_ParameterSelection_ComboBox.Value = ""
88     FD_High_TextBox.Value = ""
89     FD_Low_TextBox.Value = ""
90
91     FactorMatrix(FactorNumber - 1) = ModuleIndex
92     End Sub
93
94     'Remove a factor previously created
95     Private Sub FD_Delete_Button_Click()
96     For i = 0 To (FD_Experiments_Listbox.ListCount - 1)
97     If (FD_Experiments_Listbox.Selected(i) = True) Then
98         FD_Experiments_Listbox.RemoveItem (i)
99     End If
100    Next i
101    End Sub
102
103    'When a module is selected on the form, get its parameters
104    Private Sub FD_ModuleSelection_ComboBox_Change()
105    'Check if the file exists
106    If (Dir(vFilePath) = "") Then
107        MsgBox "The file introduced doesn't exist"
108    Else
109        Set ArenaApp = CreateObject("Arena.Application")
110        If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
111        "_temp.doe") <> "") Then
112            ArenaApp.Models.Close
113        End If
114        Set Model = ArenaApp.Models.Open(vFilePath)
115        Set vSIMAN = Model.SIMAN
116        Set vView = Model.ActiveView
117
118        FD_ParameterSelection_ComboBox.Clear
119        vModule = FD_Definition_Form.FD_ModuleSelection_ComboBox.Value
120
121        n = CInt(Model.Modules.Count)
122        i = 1
123        While (Model.Modules(i).Caption <> vModule And i <= n)
124            i = i + 1
125        Wend
126        ModuleIndex = i
127
128        n = Model.Modules(ModuleIndex).Definition.Operands.Count
129        For i = 1 To n
130            vOperand =
131            Replace(Replace(Model.Modules(ModuleIndex).Definition.Operands(i).P
132            rompt, "&", ""), ":" , "")
133            FD_ParameterSelection_ComboBox.AddItem (vOperand)
134        Next
135        Model.Close
136    End If
137    End Sub

```

```

131 'When a module and a parameter had been selected on the form, get
132 its default value
133 Private Sub FD_ParameterSelection_ComboBox_Change()
134 If FD_ParameterSelection_ComboBox.Value <> Empty Then
135 'Check if the file exists
136 If (Dir(vFilePath) = "") Then
137 MsgBox "The file introduced doesn't exist"
138 Else
139 vFilePath = Hoja1.FilePath_TextBox.Value
140 Set ArenaApp = CreateObject("Arena.Application")
141 If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
142 "_temp.doe") <> "") Then
143 ArenaApp.Models.Close
144 End If
145 Set Model = ArenaApp.Models.Open(vFilePath)
146 Set vSIMAN = Model.SIMAN
147 Set vView = Model.ActiveView
148 vModule = FD_Definition_Form.FD_ModuleSelection_ComboBox.Value
149 n = CInt(Model.Modules.Count)
150 i = 1
151 While (Model.Modules(i).Caption <> vModule And i <= n)
152 i = i + 1
153 Wend
154 ModuleIndex = i
155 nOperand = FD_ParameterSelection_ComboBox.ListIndex + 1
156 vOperand =
157 Model.Modules(ModuleIndex).Definition.Operands(nOperand).Name
158 FD_High_TextBox = Model.Modules(ModuleIndex).Data(vOperand)
159 End If
160 End If
161 End Sub
162
163 'Read the number of factors
164 Private Sub FD_NFactors_Textbox_Change()
165 ReDim FactorMatrix(CInt(FD_NFactors_Textbox.Value))
166 End Sub

```

2.2.6. ***FD_OutputVariableSelection_Form***

```

1 Option Explicit
2
3 Private Sub FD_Accept_Button_Click()
4 If (FD_OutputVariable_ComboBox <> Empty) Then
5 FD_OutputVariableSelection_Form.Hide
6 Else
7 MsgBox "Select the variable for the analysis"
8 End If
9 End Sub

```

2.3. Mòduls

2.3.1. Mòdul 1

```

1      'Global variables initialization
2      Global Model As Arena.Model
3      Global ArenaApp As Arena.Application
4      Global ModelLogic As Arena.ModelLogic
5      Global vSIMAN As Arena.SIMAN
6      Global vView As Arena.View
7      Global vFilePath As String
8      Global vFilePath2 As String
9      Global exl As New Excel.Application
10
11     Global n As Integer      'Sample length
12     Global XOut() As Double 'Output values
13     Global FactorMatrix() As Integer

```

2.3.2. Mòdul 2

```

1      'Definition of recurring functions
2      Option Explicit
3      Dim Last_Value As Boolean
4      Dim i, n, r As Integer
5      Dim Y, S, exp_term, new_term, sum, x, a, b, t, fbeta As Double
6
7      'Function that returns the count of output values
8      Function Replication_Length(ByVal Column As Integer, ByVal
9          First_Row As Integer) As Integer
9          Last_Value = False
10         i = First_Row
11         n = 0
12         While Last_Value = False
13             n = n + 1
14             If IsEmpty(Cells(i + 1, Column)) = True Then
15                 Last_Value = True
16             End If
17             i = i + 1
18         Wend
19         Replication_Length = n
20     End Function
21
22     'Function that returns the mean value
23     Function mean(ByVal Index As Integer) As Double
24         Y = 0
25         For r = 1 To Model.NumberOfReplications
26             Y = Y + XOut(Index, r)
27         Next r
28         mean = Y / Model.NumberOfReplications
29     End Function
30
31     'Function that returns standard deviation value
32     Function Sd(ByVal Index As Integer, ByVal Y As Double) As Double
33         S = 0
34         For r = 1 To Model.NumberOfReplications

```

```

35 S = S + (XOut(Index, r) - Y) ^ 2
36 Next r
37 If S > 0 Then
38 S = Math.Sqr(S / (Model.NumberOfReplications - 1))
39 End If
40 Sd = S
41 End Function
42
43 'Recursive method for beta value calculation
44 Public Function beta(ByVal lambda As Double, ByVal CI As Double,
        ByVal i As Integer)
45 exp_term = Exp(-(lambda ^ 2) * i / 2)
46 b = (i - 1) / 2
47 t = exl.WorksheetFunction.TInv((1 - CI), (i - 1))
48 x = (t ^ 2) / (i - 1 + (t ^ 2))
49
50 'First terms
51 a = 0.5
52 new_term = (exl.WorksheetFunction.BetaDist(x, a, b))
53
54 'Other terms
55 sum = 0
56 r = 1
57 While new_term > 0.000001
58 sum = sum + new_term
59 a = r + 0.5
60 fbeta = exl.WorksheetFunction.BetaDist(x, a, b)
61 new_term = ((0.5 * i * (lambda ^ 2)) ^ r) * fbeta /
        (exl.WorksheetFunction.Fact(r))
62 r = r + 1
63 Wend
64 beta = exp_term * sum * 100
65 End Function

```

2.3.3. Mòdul 3

```

1 'Welch method definition
2 Option Explicit
3 Dim i, j, Row_Ini, Row, Module, ConnectionCounter As Integer
4 Dim vLastModule As Long
5 Dim ModuleFound As Boolean
6 Dim FirstModule, AssignModule, Assign2Module, CreateTimeModule,
    AssignTimeModule, WriteTimeModule, DisposeTimeModule, vCounter,
    vEntityCounter, File, vPreviousTime, vCurrentTime, vAverageQueue,
    vWIP As Arena.Module
7 Dim ToObject As Arena.Shape
8 Dim Con, Con1, Con2, Con3, Con4, Connection As Arena.Connection
9
10 Public Sub Welch(ByVal w As Integer)
11 'Delete any graph previously created
12 If (Hoja2.ChartObjects.Count > 0) Then
13 Hoja2.ChartObjects.Delete
14 End If
15
16 'Delete previous data
17 Range("A:A").ClearContents

```

```

18  'Generation of the copy of the file introduced on "Main" sheet to
19  do the changes in the model
20  vFilePath = Hoja1.FilePath_TextBox.Value
21  If (Dir(vFilePath) = "") Then 'Check if the file exists
22  MsgBox "The file introduced doesn't exist"
23  Else
24  Set ArenaApp = CreateObject("Arena.Application")
25  If (Dir(vFilePath & "_temp.doe") <> "") Or Dir(vFilePath2 &
26  "_temp.doe") <> "" Then
27  ArenaApp.Models.Close
28  End If
29  Set Model = ArenaApp.Models.Open(vFilePath)
30  Model.SaveAs (vFilePath & "_temp.doe")
31  Model.Close
32  Set Model = ArenaApp.Models.Open(vFilePath & "_temp.doe")
33  Set vSIMAN = Model.SIMAN
34  Set vView = Model.ActiveView
35  'Code for entities entering on a module
36  If (Hoja2.TSA_OutputVariable1 = True) Then
37  vView.AutoConnect = False
38
39  'An Assign module must be created between the two blocks introduced
40  For Module = 1 To Model.Modules.Count
41  If (Model.Modules(Module).Data("Name") =
42  TSA_Definition_Form.TSA_Module1Selection_ComboBox.Value) Then
43  i = Module
44  End If
45  Next Module
46  Set FirstModule = Model.Modules(i)
47
48  'Create the Assign module
49  Set AssignModule = Model.Modules.Create("BasicProcess", "Assign",
50  1700, -800)
51  ReDim EntryConnections(Model.Modules.Count * 4)
52  ConnectionCounter = 0
53
54  'Modification of the module connections to integrate the assign
55  Model.ActiveView.Selection.DeselectAll
56  For Module = 1 To Model.Modules.Count
57  For Each Connection In Model.Modules(Module).FromConnections
58  ConnectionCounter = ConnectionCounter + 1
59  If (Model.Modules.Find(smFindTag, Connection.To) = i) Then
60  Connection.Shape.Selected = True
61  Model.ActiveView.Selection.Delete
62  Set EntryConnections(ConnectionCounter) =
63  Model.Connections.Create(Model.Modules(Module), AssignModule)
64  Model.ActiveView.Selection.DeselectAll
65  End If
66  Next Connection
67  Next Module
68  Set Con = Model.Connections.Create(AssignModule, FirstModule)
69
70  AssignModule.Data("Type(1)") = "Variable"
71  AssignModule.Data("VName(1)") = "counter" 'This variable will count
72  the number of entities entering in the second module

```

```

69 AssignModule.Data("Value(1)") = "counter + 1"
70
71 'A submodel must be created in order to count the time
72 Set CreateTimeModule = Model.Modules.Create("BasicProcess",
73 "Create", 1500, 500)
74 CreateTimeModule.Data("Interarrival Type") = "Constant"
75 CreateTimeModule.Data("Value") = 1
76 CreateTimeModule.Data("Units") = "Hours"
77 Set vCounter = Model.Modules.Create("BasicProcess", "Variable",
78 0)
79 vCounter.Data("Name") = "counter"
80 Set vEntityCounter = Model.Modules.Create("BasicProcess",
81 "Variable", 0, 0)
82 vEntityCounter.Data("Name") = "EntityCounter"
83
84 'Assign Module
85 Set AssignTimeModule = Model.Modules.Create("BasicProcess",
86 "Assign", 2500, 500)
87 AssignTimeModule.Data("Type(1)") = "Variable"
88 AssignTimeModule.Data("VName(1)") = "EntityCounter"
89 AssignTimeModule.Data("Value(1)") = "counter"
90 AssignTimeModule.Data("Type(2)") = "Variable"
91 AssignTimeModule.Data("VName(2)") = "counter"
92 AssignTimeModule.Data("Value(2)") = 0
93
94 'ReadWrite Module
95 Set File = Model.Modules.Create("AdvancedProcess", "File", 0, 0)
96 File.Data("Name") = "ExcelFile"
97 File.Data("Access Type") = "Microsoft Excel 2007 (*.xlsx)"
98 File.Data("OSFile") = ThisWorkbook.Path & "\" & ThisWorkbook.Name
99 File.Data("Recordset Name") = "Transient_State_Analysis_Recordset"
100 File.Data("Named Range") = "[Transient State Analysis$A:B]"
101
102 Set WriteTimeModule = Model.Modules.Create("AdvancedProcess",
103 "ReadWrite", 3500, 500)
104 WriteTimeModule.Data("File Name") = "ExcelFile"
105 WriteTimeModule.Data("Type") = "Write to File"
106 WriteTimeModule.Data("Recordset ID") =
107 "Transient_State_Analysis_Recordset"
108 WriteTimeModule.Data("Variable Type(1)") = "Variable"
109 WriteTimeModule.Data("Var(1)") = "EntityCounter"
110 WriteTimeModule.Data("Variable Type(2)") = "Other"
111 WriteTimeModule.Data("Other(2)") = "NREP"
112
113 Set DisposeTimeModule = Model.Modules.Create("BasicProcess",
114 "Dispose", 4500, 500)
115
116 'Connections between modules
117 Set Con2 = Model.Connections.Create(CreateTimeModule,
118 AssignTimeModule)
119 Set Con3 = Model.Connections.Create(AssignTimeModule,
120 WriteTimeModule)
121 Set Con4 = Model.Connections.Create(WriteTimeModule,
122 DisposeTimeModule)
123
124 Model.ActiveView.Selection.DeselectAll
125 ModuleFound = False
126
127

```

```

118  While ModuleFound = False And Module <= Model.Modules.Count
119  If (Model.Modules(Module).Definition = "File") Then
120  If (Model.Modules(Module).Data("OSFile") = Empty) Then
121  Model.Modules(Module).Shape.Selected = True
122  Model.ActiveView.Selection.Delete
123  ModuleFound = True
124  End If
125  End If
126  Module = Module + 1
127  Wend
128
129  'Code for average cue lenght
130  ElseIf (Hoja2.TSA_OutputVariable2 = True) Then
131
132  'Create the additional modules
133  vView.AutoConnect = False
134
135  'Assign the modules introduced in the form
136  For Module = 1 To Model.Modules.Count
137  If (Model.Modules(Module).Data("Name") =
TSA_Definition_Form.TSA_Module1Selection_ComboBox.Value) Then
138  i = Module
139  End If
140  Next Module
141  Set FirstModule = Model.Modules(i)
142
143  'Create blocs and new connections
144  Set AssignModule = Model.Modules.Create("BasicProcess", "Assign",
1700, -800)
145
146  'Time variables
147  AssignModule.Data("Type(1)") = "Variable"
148  AssignModule.Data("VName(1)") = "Previous_Time"
149  AssignModule.Data("Value(1)") = "Current_Time"
150
151  AssignModule.Data("Type(2)") = "Variable"
152  AssignModule.Data("VName(2)") = "Current_Time"
153  AssignModule.Data("Value(2)") = "TNOW"
154
155  'Queue variable
156  AssignModule.Data("Type(3)") = "Variable"
157  AssignModule.Data("VName(3)") = "Counter"
158  AssignModule.Data("Value(3)") = "Counter + ((Current_Time -
Previous_Time) * WorkInProcess)"
159
160  'WIP variable
161  AssignModule.Data("Type(4)") = "Variable"
162  AssignModule.Data("VName(4)") = "WorkInProcess"
163  AssignModule.Data("Value(4)") = "WorkInProcess + 1"
164
165  'Assign 2
166  Set Assign2Module = Model.Modules.Create("BasicProcess", "Assign",
2200, -800)
167
168  ReDim EntryConnections(Model.Modules.Count * 4)
169  ConnectionCounter = 0
170

```

```

171 'Modification of the module connections to integrate the assign
172 module
173 Model.ActiveView.Selection.DeselectAll
174 For Module = 1 To Model.Modules.Count
175 For Each Connection In Model.Modules(Module).FromConnections
176 ConnectionCounter = ConnectionCounter + 1
177 If (Model.Modules.Find(smFindTag, Connection.To) = i) Then
178 Connection.Shape.Selected = True
179 Model.ActiveView.Selection.Delete
180 Set EntryConnections(ConnectionCounter) =
181 Model.Connections.Create(Model.Modules(Module), AssignModule)
182 Model.ActiveView.Selection.DeselectAll
183 End If
184 Next Connection
185 Next Module
186 Set Con = Model.Connections.Create(AssignModule, FirstModule)
187
188 ReDim ExitConnections(Model.Modules.Count * 4)
189 ConnectionCounter = 0
190
191 'Modification of the module connections to integrate the second
192 assign module
193 Model.ActiveView.Selection.DeselectAll
194 For Module = 1 To Model.Modules.Count
195 For Each Connection In FirstModule.FromConnections
196 ConnectionCounter = ConnectionCounter + 1
197 If (Model.Modules.Find(smFindTag, Connection.To) = Module) Then
198 Connection.Shape.Selected = True
199 Model.ActiveView.Selection.Delete
200 Set ExitConnections(ConnectionCounter) =
201 Model.Connections.Create(Assign2Module, Model.Modules(Module))
202 Model.ActiveView.Selection.DeselectAll
203 End If
204 Next Connection
205 Next Module
206 Set Con1 = Model.Connections.Create(FirstModule, Assign2Module)
207
208 'Time variables
209 Assign2Module.Data("Type(1)") = "Variable"
210 Assign2Module.Data("VName(1)") = "Previous_Time"
211 Assign2Module.Data("Value(1)") = "Current_Time"
212
213 'Queue variable
214 Assign2Module.Data("Type(2)") = "Variable"
215 Assign2Module.Data("VName(2)") = "Current_Time"
216 Assign2Module.Data("Value(2)") = "TNOW"
217
218 'WIP variable
219 Assign2Module.Data("Type(3)") = "Variable"
220 Assign2Module.Data("VName(3)") = "Counter"
221 Assign2Module.Data("Value(3)") = "Counter + ((Current_Time -
222 Previous_Time)* WorkInProcess)"
223 'A submodel must be created in order to count the time

```

```

224 Set vWIP = Model.Modules.Create("BasicProcess", "Variable", 0, 0)
225 vWIP.Data("Name") = "WorkInProcess"
226
227 Set vCurrentTime = Model.Modules.Create("BasicProcess", "Variable",
228 0, 0)
229 vcurrentTime.Data("Name") = "Current_Time"
230
231 Set vCounter = Model.Modules.Create("BasicProcess", "Variable", 0,
232 0)
233 vCounter.Data("Name") = "Counter"
234
235 Set vPreviousTime = Model.Modules.Create("BasicProcess",
236 "Variable", 0, 0)
237 vPreviousTime.Data("Name") = "Previous_Time"
238
239 Set vAverageQueue = Model.Modules.Create("BasicProcess",
240 "Variable", 0, 0)
241 vAverageQueue.Data("Name") = "Average_Queue_Length"
242
243 'Create Module
244 Set CreateTimeModule = Model.Modules.Create("BasicProcess",
245 "Create", 1500, 500)
246 CreateTimeModule.Data("Interarrival Type") = "Constant"
247 CreateTimeModule.Data("Value") = 1
248 CreateTimeModule.Data("Units") = "Hours"
249
250 'Assign Module
251 Set AssignTimeModule = Model.Modules.Create("BasicProcess",
252 "Assign", 2500, 500)
253 AssignTimeModule.Data("Type(1)") = "Variable"
254 AssignTimeModule.Data("VName(1)") = "Average_Queue_Length"
255 AssignTimeModule.Data("Value(1)") = "Counter"
256 AssignTimeModule.Data("Type(2)") = "Variable"
257 AssignTimeModule.Data("VName(2)") = "Counter"
258 AssignTimeModule.Data("Value(2)") = 0
259
260 'Readwrite module
261 Set WriteTimeModule = Model.Modules.Create("AdvancedProcess",
262 "ReadWrite", 3500, 500)
263 vLastModule = Model.Modules.Count
264 WriteTimeModule.Data("File Name") = "ExcelFile"
265 WriteTimeModule.Data("Type") = "Write to File"
266 WriteTimeModule.Data("Recordset ID") =
267 "Transient_Analysis_Recordset"
268 WriteTimeModule.Data("Variable Type") = "Variable"
269 WriteTimeModule.Data("Var") = "Average Queue Length"
270 WriteTimeModule.Data("Variable Type(2)") = "Other"
271 WriteTimeModule.Data("Other(2)") = "NREP"
272
273 'Dispose Module

```

```
273 Set DisposeTimeModule = Model.Modules.Create("BasicProcess",
274 "Dispose", 4500, 500)
275 'New connections
276 Set Con2 = Model.Connections.Create(CreateTimeModule,
AssignTimeModule)
277 Set Con3 = Model.Connections.Create(AssignTimeModule,
WriteTimeModule)
278 Set Con4 = Model.Connections.Create(WriteTimeModule,
DisposeTimeModule)
279
280 Model.ActiveView.Selection.DeselectAll
281 ModuleFound = False
282
283 While ModuleFound = False And Module <= Model.Modules.Count
284 If (Model.Modules(Module).Definition = "File") Then
285 If (Model.Modules(Module).Data("OSFile") = Empty) Then
286 Model.Modules(Module).Shape.Selected = True
287 Model.ActiveView.Selection.Delete
288 ModuleFound = True
289 End If
290 End If
291 Module = Module + 1
292 Wend
293 End If
294
295 'We run the simulation
296 Model.BatchMode = True
297 Model.QuietMode = True
298 Model.PauseBetweenReplications = False
299
300 'Calculation of replication length
301 Model.ActiveView.Selection.DeselectAll
302 Model.End
303 ReDim nAux(Model.NumberOfReplications)
304 Row_Ini = 1
305 Model.Go
306 For i = 1 To Model.NumberOfReplications
307 While CInt(Cells(Row_Ini, 2).Value) = i
308 nAux(i) = nAux(i) + 1
309 Row_Ini = Row_Ini + 1
310 Wend
311 Next i
312 n = nAux(1)
313 For i = 2 To Model.NumberOfReplications
314 If n < nAux(i) Then
315 n = nAux(i)
316 End If
317 Next i
318 If (w > (n / 2)) Then
319 MsgBox "Introduce a correct value for w"Else
320 'Read all the output values
321 Row = 0
322 ReDim XOut(n, Model.NumberOfReplications)
323 For i = 1 To Model.NumberOfReplications
324 For j = 1 To nAux(i)
325 Row = Row + 1
326 XOut(j, i) = Cells(Row, 1).Value
```

```

327     Next j
328     Next i
329
330     'Clear values in column A
331     Range("A:B").ClearContents
332
333     'We calculate the mean for each time unit
334     ReDim Y(n)
335     For i = 1 To n
336         Y(i) = mean(i)
337     Next i
338
339     'Application of Welch method
340     ReDim XWelch(n - w)
341     For i = 1 To w
342         For j = -(i - 1) To (i - 1)
343             XWelch(i) = XWelch(i) + Y(i + j)
344         Next j
345         XWelch(i) = XWelch(i) / (2 * i - 1)
346         Cells(i + 1, 1).Value = XWelch(i)
347     Next i
348     For i = (w + 1) To (n - w)
349         For j = (-1) * w To w
350             XWelch(i) = XWelch(i) + Y(i + j)
351         Next j
352         XWelch(i) = XWelch(i) / (2 * w + 1)
353         Cells(i + 1, 1).Value = XWelch(i)
354     Next i
355
356     'Create the graph
357     Range(Cells(2, 1), Cells(2 + n - w, 1)).Select
358     ActiveSheet.Shapes.AddChart.Select
359     With ActiveChart
360         .ChartType = xlLine
361         .PlotVisibleOnly = False
362         .SetSourceData Source:=Range(Cells(2, 1), Cells(2 + n - w, 1)),
363         PlotBy:=xlColumns
364         .HasTitle = True
365         .ChartTitle.Characters.Text = "Welch graph"
366         .HasAxis(xlValue) = True
367         .HasAxis(xlCategory) = True
368         .HasLegend = False
369         .SeriesCollection(1).MarkerStyle = xlMarkerStyleNone
370         .SeriesCollection(1).Format.Line.Weight = 2
371         .SeriesCollection(1).Format.Line.ForeColor.RGB = RGB(0, 112, 192)
372         .Export Filename:=ThisWorkbook.Path & "\temp_image.jpg"
373         .ChartArea.Format.Fill.ForeColor.RGB = RGB(255, 255, 255)
374         .PlotArea.Format.Fill.ForeColor.RGB = RGB(255, 255, 255)
375         .ChartArea.Format.Line.Visible = msoFalse
376     End With
377     With ActiveChart.Parent
378         .Height = 340
379         .Width = 600
380         .Top = 90
381         .Left = 350
382     End With
383     Model.Close
384     FileSystem.Kill (vFilePath & "_temp*")

```

```

384 End If
385 End If
386 End Sub

```

2.3.4. Mòdul 4

```

1 Option Explicit
2 Dim t As Double
3 Dim i, j, Row_Ini, Row, Module, ConnectionCounter As Integer
4 Dim FirstModule, SecondModule, AssignModule, Assign2Module,
CreateTimeModule, AssignTimeModule, WriteTimeModule,
DisposeTimeModule, vCounter, vEntityCounter, File, vPreviousTime,
vCurrentTime, vAverageQueue, vWIP As Arena.Module
5 Dim ToObject As Arena.Shape
6 Dim ArenaApp As Arena.Application
7 Dim Con, Con1, Con2, Con3, Con4, Connection As Arena.Connection
8 Dim vLastModule As Long
9 Dim ModuleFound As Boolean
10
11 Public Sub IC(alpha)
12 'Delete any graph previously created
13 If (Hoja2.ChartObjects.Count > 0) Then
14 Hoja2.ChartObjects.Delete
15 End If
16
17 'Delete previous data
18 Range("A:A").ClearContents
19
20 'Generation of the copy of the file introduced on "Main" sheet to
do the changes in the model
21 vFilePath = Hoja1.FilePath_TextBox.Value
22
23 If (Dir(vFilePath) = "") Then 'Check if the file exists
MsgBox "The file introduced doesn't exist"
25 Else
26 Set ArenaApp = CreateObject("Arena.Application")
27 If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
"_temp.doe") <> "") Then
28 ArenaApp.Models.Close
29 End If
30 Set Model = ArenaApp.Models.Open(vFilePath)
31 Model.SaveAs (vFilePath & "_temp.doe")
32 Model.Close
33 Set Model = ArenaApp.Models.Open(vFilePath & "_temp.doe")
34 Set vSIMAN = Model.SIMAN
35 Set vView = Model.ActiveView
36
37 'Code for entities entering on a block
38 If (Hoja2.TSA_OutputVariable1 = True) Then
vView.AutoConnect = False
40
41 'An Assign block must be created between the two blocks introduced
42 For Module = 1 To Model.Modules.Count
43 If (Model.Modules(Module).Data("Name") =
TSA_Definition_Form.TSA_Module1Selection_ComboBox.Value) Then
44 i = Module
45 End If

```

```

46     Next Module
47     Set FirstModule = Model.Modules(i)
48
49     'Create the Assign block and create the connections with the other
50     'blocks
51     Set AssignModule = Model.Modules.Create("BasicProcess", "Assign",
52     1700, -800)
53
54     ReDim EntryConnections(Model.Modules.Count * 4)
55     ConnectionCounter = 0
56
57     Model.ActiveView.Selection.DeselectAll
58     For Module = 1 To Model.Modules.Count
59     For Each Connection In Model.Modules(Module).FromConnections
60     ConnectionCounter = ConnectionCounter + 1
61     If (Model.Modules.Find(smFindTag, Connection.To) = i) Then
62     Connection.Shape.Selected = True
63     Model.ActiveView.Selection.Delete
64     Set EntryConnections(ConnectionCounter) =
65     Model.Connections.Create(Model.Modules(Module), AssignModule)
66     Model.ActiveView.Selection.DeselectAll
67     End If
68     Next Connection
69     Next Module
70     Set Con = Model.Connections.Create(AssignModule, FirstModule)
71
72     AssignModule.Data("Type(1)") = "Variable"
73     AssignModule.Data("VName(1)") = "counter" 'This variable will count
74     'the number of entities entering in the second block
75     AssignModule.Data("Value(1)") = "counter + 1"
76
77     'A submodel must be created in order to count the time
78     Set CreateTimeModule = Model.Modules.Create("BasicProcess",
79     "Create", 1500, 500)
80     CreateTimeModule.Data("Interarrival Type") = "Constant"
81     CreateTimeModule.Data("Value") = 1
82     CreateTimeModule.Data("Units") = "Hours"
83     Set vCounter = Model.Modules.Create("BasicProcess", "Variable", 0,
84     0)
85     vCounter.Data("Name") = "counter"
86     Set vEntityCounter = Model.Modules.Create("BasicProcess",
87     "Variable", 0, 0)
88     vEntityCounter.Data("Name") = "EntityCounter"
89
90     'Assign Module
91     Set AssignTimeModule = Model.Modules.Create("BasicProcess",
92     "Assign", 2500, 500)
93     AssignTimeModule.Data("Type(1)") = "Variable"
94     AssignTimeModule.Data("VName(1)") = "EntityCounter"
95     AssignTimeModule.Data("Value(1)") = "counter"
96     AssignTimeModule.Data("Type(2)") = "Variable"
97     AssignTimeModule.Data("VName(2)") = "counter"
98     AssignTimeModule.Data("Value(2)") = 0
99
100    'ReadWrite Module
101    Set File = Model.Modules.Create("AdvancedProcess", "File", 0, 0)
102    File.Data("Name") = "ExcelFile"
103    File.Data("Access Type") = "Microsoft Excel 2007 (*.xlsx)"

```

```
96 File.Data("OSFile") = ThisWorkbook.Path & "\" & ThisWorkbook.Name
97 File.Data("Recordset Name") = "Transient_State_Analysis_Recordset"
98 File.Data("Named Range") = "[Transient State Analysis$A:B]"
99
100 Set WriteTimeModule = Model.Modules.Create("AdvancedProcess",
101     "ReadWrite", 3500, 500)
102 WriteTimeModule.Data("File Name") = "ExcelFile"
103 WriteTimeModule.Data("Type") = "Write to File"
104 WriteTimeModule.Data("Recordset ID") =
105     "Transient_State_Analysis_Recordset"
106 WriteTimeModule.Data("Variable Type(1)") = "Variable"
107 WriteTimeModule.Data("Var(1)") = "EntityCounter"
108 WriteTimeModule.Data("Variable Type(2)") = "Other"
109 WriteTimeModule.Data("Other(2)") = "NREP"
110
111 'Connections between modules
112 Set Con2 = Model.Connections.Create(CreateTimeModule,
113     AssignTimeModule)
114 Set Con3 = Model.Connections.Create(AssignTimeModule,
115     WriteTimeModule)
116 Set Con4 = Model.Connections.Create(WriteTimeModule,
117     DisposeTimeModule)
118
119 Model.ActiveView.Selection.DeselectAll
120 ModuleFound = False
121
122 While ModuleFound = False And Module <= Model.Modules.Count
123 If (Model.Modules(Module).Definition = "File") Then
124 If (Model.Modules(Module).Data("OSFile") = Empty) Then
125 Model.Modules(Module).Shape.Selected = True
126 Model.ActiveView.Selection.Delete
127 ModuleFound = True
128 End If
129 End If
130 Module = Module + 1
131 Wend
132
133 'Code for medium cue lenght
134 ElseIf (Hoja2.TSA_OutputVariable2 = True) Then
135
136 'Create the additional blocks
137 vView.AutoConnect = False
138
139 'Assign the modules introduced in the form
140 For Module = 1 To Model.Modules.Count
141 If (Model.Modules(Module).Data("Name") =
142     TSA_Definition_Form.TSA_Module1Selection_ComboBox.Value) Then
143 i = Module
144 End If
145 Next Module
146
147 Set FirstModule = Model.Modules(i)
148
149 'Create blocs and new connections
```

```

145 Set AssignModule = Model.Modules.Create("BasicProcess", "Assign",
146   1700, -800)
147
148 'Time variables
149 AssignModule.Data("Type(1)") = "Variable"
150 AssignModule.Data("VName(1)") = "Previous_Time"
151 AssignModule.Data("Value(1)") = "Current_Time"
152 AssignModule.Data("Type(2)") = "Variable"
153 AssignModule.Data("VName(2)") = "Current_Time"
154 AssignModule.Data("Value(2)") = "TNOW"
155
156 'Queue variable
157 AssignModule.Data("Type(3)") = "Variable"
158 AssignModule.Data("VName(3)") = "Counter"
159 AssignModule.Data("Value(3)") = "Counter + ((Current_Time -
160 Previous_Time) * WorkInProcess)"
161 'WIP variable
162 AssignModule.Data("Type(4)") = "Variable"
163 AssignModule.Data("VName(4)") = "WorkInProcess"
164 AssignModule.Data("Value(4)") = "WorkInProcess + 1"
165
166 'Assign 2
167 Set Assign2Module = Model.Modules.Create("BasicProcess", "Assign",
168   2200, -800)
169
170
171 'Modification of the module connections to integrate the assign
172 Model.ActiveView.Selection.DeselectAll
173 For Module = 1 To Model.Modules.Count
174 For Each Connection In Model.Modules(Module).FromConnections
175 ConnectionCounter = ConnectionCounter + 1
176 If (Model.Modules.Find(smFindTag, Connection.To) = i) Then
177 Connection.Shape.Selected = True
178 Model.ActiveView.Selection.Delete
179 Set EntryConnections(ConnectionCounter) =
180 Model.Connections.Create(Model.Modules(Module), AssignModule)
181 Model.ActiveView.Selection.DeselectAll
182 End If
183 Next Connection
184 Next Module
185 Set Con = Model.Connections.Create(AssignModule, FirstModule)
186
187 ReDim ExitConnections(Model.Modules.Count * 4)
188 ConnectionCounter = 0
189
190 'Modification of the module connections to integrate the second
191 assign module
192 Model.ActiveView.Selection.DeselectAll
193 For Module = 1 To Model.Modules.Count
194 For Each Connection In FirstModule.FromConnections
195 ConnectionCounter = ConnectionCounter + 1
196 If (Model.Modules.Find(smFindTag, Connection.To) = Module) Then
197 Connection.Shape.Selected = True
198 Model.ActiveView.Selection.Delete

```

```

197 Set ExitConnections(ConnectionCounter) =
Model.Connections.Create(Assign2Module, Model.Modules(Module))
198 Model.ActiveView.Selection.DeselectAll
199 End If
200 Next Connection
201 Next Module
202 Set Con1 = Model.Connections.Create(FirstModule, Assign2Module)
203
204 'Time variables
205 Assign2Module.Data("Type(1)") = "Variable"
206 Assign2Module.Data("VName(1)") = "Previous_Time"
207 Assign2Module.Data("Value(1)") = "Current_Time"
208
209 Assign2Module.Data("Type(2)") = "Variable"
210 Assign2Module.Data("VName(2)") = "Current_Time"
211 Assign2Module.Data("Value(2)") = "TNOW"
212
213 'Queue variable
214 Assign2Module.Data("Type(3)") = "Variable"
215 Assign2Module.Data("VName(3)") = "Counter"
216 Assign2Module.Data("Value(3)") = "Counter + ((Current_Time -
Previous_Time) * WorkInProcess)"
217
218 'WIP variable
219 Assign2Module.Data("Type(4)") = "Variable"
220 Assign2Module.Data("VName(4)") = "WorkInProcess"
221 Assign2Module.Data("Value(4)") = "WorkInProcess - 1"
222
223 'A submodel must be created in order to count the time
224 Set vWIP = Model.Modules.Create("BasicProcess", "Variable", 0, 0)
vWIP.Data("Name") = "WorkInProcess"
225
226 Set vCurrentTime = Model.Modules.Create("BasicProcess", "Variable",
0, 0)
vCurrentTime.Data("Name") = "Current_Time"
227
228 Set vCounter = Model.Modules.Create("BasicProcess", "Variable", 0,
0)
vCounter.Data("Name") = "Counter"
229
230 Set vPreviousTime = Model.Modules.Create("BasicProcess",
"Variable", 0, 0)
vPreviousTime.Data("Name") = "Previous_Time"
231
232 Set vAverageQueue = Model.Modules.Create("BasicProcess",
"Variable", 0, 0)
vAverageQueue.Data("Name") = "Average_Queue_Length"
233
234 'Create Module
235 Set CreateTimeModule = Model.Modules.Create("BasicProcess",
"Create", 1500, 500)
CreateTimeModule.Data("Interarrival Type") = "Constant"
236 CreateTimeModule.Data("Value") = 1
CreateTimeModule.Data("Units") = "Hours"
237
238 'Assign Module
239 Set AssignTimeModule = Model.Modules.Create("BasicProcess",
"Assign", 2500, 500)

```

```

247 AssignTimeModule.Data("Type(1)") = "Variable"
248 AssignTimeModule.Data("VName(1)") = "Average_Queue_Length"
249 AssignTimeModule.Data("Value(1)") = "Counter"
250 AssignTimeModule.Data("Type(2)") = "Variable"
251 AssignTimeModule.Data("VName(2)") = "Counter"
252 AssignTimeModule.Data("Value(2)") = 0
253
254 'Readwrite module
255 Set WriteTimeModule = Model.Modules.Create("AdvancedProcess",
256   "ReadWrite", 3500, 500)
256 vLastModule = Model.Modules.Count
257 WriteTimeModule.Data("File Name") = "ExcelFile"
258 WriteTimeModule.Data("Type") = "Write to File"
259 WriteTimeModule.Data("Recordset ID") =
260   "Transient_Analysis_Recordset"
261 WriteTimeModule.Data("Variable Type") = "Variable"
262 WriteTimeModule.Data("Var") = "Average_Queue_Length"
263 WriteTimeModule.Data("Variable Type(2)") = "Other"
264 WriteTimeModule.Data("Other(2)") = "NREP"
265
266 Set File = Model.Modules(vLastModule)
267 File.Data("Name") = "ExcelFile"
268 File.Data("Access Type") = "Microsoft Excel 2007 (*.xlsx)"
269 File.Data("OSFile") = ThisWorkbook.Path & "\" & ThisWorkbook.Name
270 File.Data("Recordset Name") = "Transient_Analysis_Recordset"
271 File.Data("Named Range") = "[Transient State Analysis$A:B]"
271
272 'Dispose Module
273 Set DisposeTimeModule = Model.Modules.Create("BasicProcess",
274   "Dispose", 4500, 500)
274
275 'New connections
276 Set Con2 = Model.Connections.Create(CreateTimeModule,
277   AssignTimeModule)
277 Set Con3 = Model.Connections.Create(AssignTimeModule,
278   WriteTimeModule)
278 Set Con4 = Model.Connections.Create(WriteTimeModule,
279   DisposeTimeModule)
280 Model.ActiveView.Selection.DeselectAll
281
282 ModuleFound = False
283
284 While ModuleFound = False And Module <= Model.Modules.Count
285 If (Model.Modules(Module).Definition = "File") Then
286 If (Model.Modules(Module).Data("OSFile") = Empty) Then
287 Model.Modules(Module).Shape.Selected = True
288 Model.ActiveView.Selection.Delete
289 ModuleFound = True
290 End If
291 End If
292 Module = Module + 1
293 Wend
294 End If
295 'Executar la simulació
296 Model.BatchMode = True
297 Model.QuietMode = True
298 Model.PauseBetweenReplications = True

```

```

299
300 'Calculation of replication length
301 ReDim nAux(Model.NumberOfReplications)
302 Row_Ini = 1
303 For i = 1 To Model.NumberOfReplications
304 Model.Go
305 nAux(i) = Replication_Length(1, Row_Ini)
306 Row_Ini = Row_Ini + nAux(i)
307 Next i
308 n = nAux(1)
309 For i = 2 To Model.NumberOfReplications
310 If n < nAux(i) Then
311 n = nAux(i)
312 End If
313 Next i
314
315 'Read all the output values
316 Row = 0
317 ReDim XOut(n, Model.NumberOfReplications)
318 For i = 1 To Model.NumberOfReplications
319 For j = 1 To nAux(i)
320 Row = Row + 1
321 XOut(j, i) = Cells(Row, 1).Value
322 Next j
323 Next i
324
325 'Clear values in column A
326 Range("A:A").ClearContents
327
328 'Mean calculation for each time unit
329 ReDim Y(n)
330 For i = 1 To n
331 Y(i) = mean(i)
332 Next i
333
334 'Variance calculation
335 ReDim S(n)
336 For i = 1 To n
337 S(i) = Sd(i, Y(i))
338 Next i
339
340 'Càlcul del interval de confiança
341 t = Excel.Application.WorksheetFunction.TInv((1 - alpha),
Model.NumberOfReplications - 1)
342 ReDim ICHigh(n)
343 ReDim ICLow(n)
344 For i = 1 To n
345 ICHigh(i) = Y(i) + t * S(i) /
(Math.Sqr(Model.NumberOfReplications))
346 ICLow(i) = Y(i) - t * S(i) / (Math.Sqr(Model.NumberOfReplications))
347 Cells(i + 1, 1).Value = ICHigh(i) - ICLow(i)
348 Next i
349
350 'Gràfic intervals de confiança
351 Range("A2").Select
352 Range(Selection, Selection.End(xlDown)).Select
353 ActiveSheet.Shapes.AddChart.Select
354 With ActiveChart

```

```

355     .ChartType = xlLine
356     .PlotVisibleOnly = False
357     .HasLegend = False
358     .HasAxis(xlValue) = True
359     .HasAxis(xlCategory) = True
360     .HasTitle = True
361     .ChartTitle.Characters.Text = "Confidence intervals width"
362     .SeriesCollection(1).Format.Line.Weight = 2
363     .SeriesCollection(1).Format.Line.ForeColor.RGB = RGB(0, 112, 192)
364     .SeriesCollection(1).MarkerStyle = xlMarkerStyleNone
365     .Export Filename:=ThisWorkbook.Path & "\temp_image.jpg"
366     .ChartArea.Format.Fill.ForeColor.RGB = RGB(255, 255, 255)
367     .PlotArea.Format.Fill.ForeColor.RGB = RGB(255, 255, 255)
368     .ChartArea.Format.Line.Visible = msoFalse
369 End With
370 With ActiveChart.Parent
371     .Height = 340
372     .Width = 600
373     .Top = 90
374     .Left = 350
375 End With
376 Model.Close
377 FileSystem.Kill (vFilePath & "_temp*")
378 End If
379 End Sub

```

2.3.5. Mòdul 5

```

1 Option Explicit
2 Dim Y, S, current_e, t, r As Double
3 Dim Last_Iteration As Boolean
4 Dim i, x, R0 As Integer
5 Dim Variable As String
6
7 Public Sub N_Rep(ByVal CI As Double, ByVal e As Double)
8 'Check if the file exists
9 If (Dir(vFilePath) = "") Then
10 MsgBox "The file introduced doesn't exist"
11 Else
12 'We generate a copy of the file introduced on "Intro" sheet to do
13 'the changes in the model
14 vFilePath = Hoja1.FilePath_TextBox.Value
15 Set ArenaApp = CreateObject("Arena.Application")
16 If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
17 "_temp.doe") <> "") Then
18 ArenaApp.Models.Close
19 End If
20 Set Model = ArenaApp.Models.Open(vFilePath)
21 Model.SaveAs (vFilePath & "_temp.doe")
22 Model.Close
23 Set Model = ArenaApp.Models.Open(vFilePath & "_temp.doe")
24 Set vSIMAN = Model.SIMAN
25 Set vView = Model.ActiveView
26
27 'Delete previous executions graphs and data
28 ActiveSheet.EnableSelection = xlNoSelection
29 i = 1

```

```

28 While i <= ActiveSheet.Shapes.Count
29 If (ActiveSheet.Shapes.Item(i).Name = "Rectangle1" Or
ActiveSheet.Shapes.Item(i).Name = "Line1" Or
ActiveSheet.Shapes.Item(i).Name = "Line2" Or
ActiveSheet.Shapes.Item(i).Name = "Line3" Or
ActiveSheet.Shapes.Item(i).Name = "Line4" Or
ActiveSheet.Shapes.Item(i).Name = "Text1" Or
ActiveSheet.Shapes.Item(i).Name = "Text2" Or
ActiveSheet.Shapes.Item(i).Name = "Text_Des_CI" Or
ActiveSheet.Shapes.Item(i).Name = "Text_Current_CI") Then
30 ActiveSheet.Shapes.Item(i).Delete
31 i = i - 1
32 End If
33 i = i + 1
34 Wend
35
36 'Delete previous data
37 Range("N:Q").Select
38 Selection.ClearContents
39 Selection.Borders(xlEdgeBottom).LineStyle = xlNone
40
41 Variable =
RN_OutputVariableSelection_Form.RN_OutputVariable.ComboBox.Value
42
43 'Execute the simulation
44 Model.BatchMode = True
45 Model.QuietMode = True
46 Model.PauseBetweenReplications = True
47
48 ReDim XOut(1, Model.NumberOfReplications)
49 For i = 1 To Model.NumberOfReplications
50 Model.Go
51 XOut(1, i) =
vSIMAN.VariableArrayValue(vSIMAN.SymbolNumber(Variable))
52 Next i
53
54 'Mean calculation for each time unit
55 Y = mean(1)
56
57 'Variance calculation
58 S = Sd(1, Y)
59
60 'Confidence interval width
61 t = exl.WorksheetFunction.TInv((1 - CI),
(Model.NumberOfReplications - 1))
62 current_e = (t * S) / (Math.Sqrt(Model.NumberOfReplications))
63
64 'Iterative process
65 For i = 14 To 17
66 Cells(7, i).Font.Bold = True
a. Cells(7, i).Font.Color = RGB(100, 100, 100)
67 Next i
68 Cells(7, 14).Value = "Iteration"
69 Cells(7, 15).Value = "R0"
70 Cells(7, 16).Value = "tStudent"
71 Cells(7, 17).Value = "Desired R"

```

```

72     Range(Cells(7, 14), Cells(7, 17)).Select
73     With Selection.Borders(xlEdgeBottom)
74         .LineStyle = xlContinuous
75         .Color = -10197916
76         .TintAndShade = 0
77         .Weight = xlThin
78     End With
79     With Selection.Borders(xlEdgeTop)
80         .LineStyle = xlContinuous
81         .Color = -10197916
82         .TintAndShade = 0
83         .Weight = xlThin
84     End With
85
86     'First R estimation
87     t = exl.WorksheetFunction.NormSInv((1 - CI) / 2)
88     R0 = CInt(Round((t * S / e) ^ 2, 0))
89
90     'R0 = Model.NumberOfReplications
91     Last_Iteration = False
92     i = 1
93     While Last_Iteration = False
94         Cells(i + 7, 14).Value = i
95         Cells(i + 7, 14).NumberFormat = "0"
96         Cells(i + 7, 14).HorizontalAlignment = xlLeft
97         Cells(i + 7, 15).Value = R0
98         Cells(i + 7, 15).NumberFormat = "0"
99         Cells(i + 7, 15).HorizontalAlignment = xlLeft
100        t = exl.WorksheetFunction.TInv((1 - CI), (R0 - 1))
101        S = Sd(1, Y)
102        r = (t * S / e) ^ 2
103        If (r > R0) Then 'continua
104        Else:
105            Last_Iteration = True
106        End If
107        R0 = R0 + 1
108        Cells(i + 7, 16).Value = t
109        Cells(i + 7, 16).NumberFormat = "0.00000000"
110        Cells(i + 7, 16).HorizontalAlignment = xlLeft
111        Cells(i + 7, 17).Value = "R > " & Round(r, 2)
112        Cells(i + 7, 17).HorizontalAlignment = xlLeft
113        i = i + 1
114    Wend
115
116    'Draw confidence intervals
117    current_e = Round(current_e, 2)
118    x = 540
119    If (e < current_e) Then
120        With ActiveSheet.Shapes.AddLine(x - 175 * (e / current_e), 170, x +
121            175 * (e / current_e), 170)
122            .Line.Weight = xlThin
123            .Name = "Line1"
124            .Line.ForeColor.RGB = RGB(146, 208, 80)
125        End With
126        With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal, x -
127            175 * (e / current_e), 145, 350 * (e / current_e), 20)
128            .TextFrame.Characters.Text = Round(Y, 2) & "± " & e
129            .TextFrame.Characters.Font.Size = 10

```

```
128 .TextFrame.HorizontalAlignment = xlHAlignCenter
129 .Line.Visible = msoFalse
130 .Fill.Visible = msoFalse
131 .Name = "Text1"
132 End With
133 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal, x
- 175 * (current_e / e), 105, 350 * (current_e / e), 25)
134 .TextFrame.Characters.Text = "Desired interval"
135 .TextFrame.Characters.Font.Size = 14
136 .TextFrame.Characters.Font.Bold = True
137 .TextFrame.HorizontalAlignment = xlHAlignCenter
138 .Line.Visible = msoFalse
139 .Fill.Visible = msoFalse
140 .Name = "Text_Des_CI"
141 End With
142 With ActiveSheet.Shapes.AddLine(365, 285, 715, 285)
143 .Line.Weight = xlThin
144 .Name = "Line2"
145 .Line.ForeColor.RGB = RGB(0, 112, 192)
146 End With
147 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
365, 260, 350, 20)
148 .TextFrame.Characters.Text = Round(Y, 2) & "± " & current_e
149 .TextFrame.Characters.Font.Size = 10
150 .TextFrame.HorizontalAlignment = xlHAlignCenter
151 .Line.Visible = msoFalse
152 .Fill.Visible = msoFalse
153 .Name = "Text2"
154 End With
155 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
365, 220, 350, 25)
156 .TextFrame.Characters.Text = "Current interval"
157 .TextFrame.Characters.Font.Size = 14
158 .TextFrame.Characters.Font.Bold = True
159 .TextFrame.HorizontalAlignment = xlHAlignCenter
160 .Line.Visible = msoFalse
161 .Fill.Visible = msoFalse
162 .Name = "Text_Current_CI"
163 End With
164 Else
165 With ActiveSheet.Shapes.AddLine(365, 170, 715, 170)
166 .Line.Weight = xlThin
167 .Name = "Line1"
168 .Line.ForeColor.RGB = RGB(146, 208, 80)
169 End With
170 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
365, 145, 350, 20)
171 .TextFrame.Characters.Text = Round(Y, 2) & "± " & e
172 .TextFrame.Characters.Font.Size = 10
173 .TextFrame.HorizontalAlignment = xlHAlignCenter
174 .Line.Visible = msoFalse
175 .Fill.Visible = msoFalse
176 .Name = "Text1"
177 End With
178 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
365, 105, 350, 25)
179 .TextFrame.Characters.Text = "Desired interval"
180 .TextFrame.Characters.Font.Size = 14
```

```
181 .TextFrame.Characters.Font.Bold = True
182 .TextFrame.HorizontalAlignment = xlHAlignCenter
183 .Line.Visible = msoFalse
184 .Fill.Visible = msoFalse
185 .Name = "Text_Des_CI"
186 End With
187 With ActiveSheet.Shapes.AddLine(x - 175 * (current_e / e), 285, x +
188 175 * (current_e / e), 285)
189 .Line.Weight = xlThin
190 .Name = "Line2"
191 .Line.ForeColor.RGB = RGB(0, 112, 192)
192 End With
193 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal, x -
194 175 * (current_e / e), 260, 350 * (current_e / e), 20)
195 .TextFrame.Characters.Text = Round(Y, 2) & "± " & current_e
196 .TextFrame.Characters.Font.Size = 10
197 .TextFrame.HorizontalAlignment = xlHAlignCenter
198 .Line.Visible = msoFalse
199 .Fill.Visible = msoFalse
200 .Name = "Text2"
201 End With
202 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal, x -
203 175 * (current_e / e), 220, 350 * (current_e / e), 25)
204 .TextFrame.Characters.Text = "Current interval"
205 .TextFrame.Characters.Font.Size = 14
206 .TextFrame.Characters.Font.Bold = True
207 .TextFrame.HorizontalAlignment = xlHAlignCenter
208 .Line.Visible = msoFalse
209 .Fill.Visible = msoFalse
210 .Name = "Text_Current_CI"
211 End With
212 End If
213 With ActiveSheet.Shapes.AddLine(x, 165, x, 175)
214 .Line.ForeColor.RGB = RGB(0, 0, 0)
215 .Name = "Line3"
216 End With
217 With ActiveSheet.Shapes.AddLine(x, 280, x, 290)
218 .Line.ForeColor.RGB = RGB(0, 0, 0)
219 .Name = "Line4"
220 End With
221 With ActiveSheet.Shapes.AddShape(msoShapeRectangle, x - 200, 88,
222 400, 265)
223 .Fill.ForeColor.RGB = RGB(255, 255, 255)
224 .Line.ForeColor.RGB = RGB(255, 255, 255)
225 While .ZOrderPosition > 1
226 .ZOrder msoSendBackward
227 Wend
228 .Name = "Rectangle1"
229 End With
230 Model.Close
231 FileSystem.Kill (vFilePath & "_temp*")
232 End If
233 End Sub
```

2.3.6. Mòdul 6

```

1 Option Explicit
2 Dim Y, S, t0, t, new_beta, vMessage2 As Double
3 Dim vMessage1, vMessage3 As String
4 Dim vColor As Long
5 Dim n, i As Integer
6 Dim Variable As String
7
8 Public Sub Model_Validation(ByVal CI As Double, ByVal des_beta As
9 Double, ByVal des_mean As Double, ByVal lambda As Double)
10
11 'Delete previous results
12 Range("H:M").Select
13 Selection.ClearContents
14 Cells(5, 7).Value = Empty
15 Cells(7, 7).Value = Empty
16 ActiveSheet.EnableSelection = xlNoSelection
17 i = 1
18 While i <= ActiveSheet.Shapes.Count
19 If (ActiveSheet.Shapes.Item(i).Name = "Line1" Or
20 ActiveSheet.Shapes.Item(i).Name = "Line2" Or
21 ActiveSheet.Shapes.Item(i).Name = "Line3" Or
22 ActiveSheet.Shapes.Item(i).Name = "Line4" Or
23 ActiveSheet.Shapes.Item(i).Name = "Line5" Or
24 ActiveSheet.Shapes.Item(i).Name = "Line6" Or
25 ActiveSheet.Shapes.Item(i).Name = "Line7") Or
26 ActiveSheet.Shapes.Item(i).Name = "Text1" Or
27 ActiveSheet.Shapes.Item(i).Name = "Text2" Or
28 ActiveSheet.Shapes.Item(i).Name = "Text3" Or
29 ActiveSheet.Shapes.Item(i).Name = "Text4" Or
30 ActiveSheet.Shapes.Item(i).Name = "Text5" Then
31     a. ActiveSheet.Shapes.Item(i).Delete
32     b. i = i - 1
33 End If
34 i = i + 1
35 Wend
36
37 'Check if the file exists
38 If (Dir(vFilePath) = "") Then
39 MsgBox "The file introduced doesn't exist"
40 Else
41     'We generate a copy of the file introduced on "Intro" sheet to do
42     'the changes in the model
43     vFilePath = Hoja1.FilePath_TextBox.Value
44     Set ArenaApp = CreateObject("Arena.Application")
45     If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
46     "_temp.doe") <> "") Then
47         ArenaApp.Models.Close
48     End If
49     Set Model = ArenaApp.Models.Open(vFilePath)
50     Model.SaveAs (vFilePath & "_temp.doe")
51     Model.Close
52     Set Model = ArenaApp.Models.Open(vFilePath & "_temp.doe")
53     Set vSIMAN = Model.SIMAN
54     Set vView = Model.ActiveView

```

```

40     Variable =
41     MV_OutputVariableSelection_Form.MV_OutputVariable_ComboBox.Value
42     Model.BatchMode = True
43     Model.QuietMode = True
44     Model.PauseBetweenReplications = True
45     ReDim XOut(1, Model.NumberOfReplications)
46     For i = 1 To Model.NumberOfReplications
47       Model.Go
48       XOut(1, i) =
49         vSIMAN.VariableArrayValue(vSIMAN.SymbolNumber(Variable))
50       Next i
51
52     'Mean calculation for each time unit
53     Y = mean(1)
54
55     'Variance calculation
56     S = Sd(1, Y)
57
58     'Model Validation
59     t0 = (Y - des_mean) / (S / Math.Sqrt(Model.NumberOfReplications))
60     t = exl.WorksheetFunction.TInv((1 - CI),
61       (Model.NumberOfReplications - 1))
62
63     'We calculate current beta value
64     new_beta = beta(lambda, CI, Model.NumberOfReplications)
65
66     If Abs(t0) < t Then
67       vMessage1 = "VALID"
68       vColor = RGB(146, 208, 80)
69     Else
70       vMessage1 = "NON VALID"
71       vColor = RGB(192, 0, 0)
72     End If
73     Cells(8, 9).Value = "This model can be accepted as"
74     Cells(8, 9).Font.Size = 14
75     Cells(10, 9).Value = vMessage1
76     Cells(10, 9).Font.Size = 14
77     Cells(10, 9).Font.Bold = True
78     Cells(10, 9).Font.Color = vColor
79
80     vMessage2 = Round(new_beta, 2) / 100
81     Cells(8, 13).Value = "Risk of accepting a non valid model: "
82     Cells(8, 13).Font.Size = 14
83     Cells(10, 13).Value = vMessage2
84     Cells(10, 13).Font.Color = RGB(0, 112, 192)
85     Cells(10, 13).Font.Bold = True
86     Cells(10, 13).Font.Size = 14
87     Cells(10, 13).NumberFormat = "0.00%"
88
89     'Draw the confidence intervals
90     With ActiveSheet.Shapes.AddLine(450, 250, 450 + (new_beta * 4),
91       250)
92       .Line.Weight = xlThick
93       .Name = "Line1"
94       .Line.ForeColor.RGB = RGB(0, 112, 192)
95     End With
96     With ActiveSheet.Shapes.AddLine(450, 250, 850, 250)

```

```
93 .Line.ForeColor.RGB = RGB(0, 0, 0)
94 .Name = "Line2"
95 End With
96 With ActiveSheet.Shapes.AddLine(450, 245, 450, 255)
97 .Line.ForeColor.RGB = RGB(0, 0, 0)
98 .Name = "Line3"
99 End With
100 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
101 425, 262, 40, 20)
102 .TextFrame.Characters.Text = "0%"
103 .TextFrame.Characters.Font.Size = 10
104 .TextFrame.HorizontalAlignment = xlHAlignCenter
105 .Line.Visible = msoFalse
106 .Fill.Visible = msoFalse
107 .Name = "Text1"
108 End With
109 With ActiveSheet.Shapes.AddLine(550, 245, 550, 255)
110 .Line.ForeColor.RGB = RGB(0, 0, 0)
111 .Name = "Line4"
112 End With
113 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
114 525, 262, 40, 20)
115 .TextFrame.Characters.Text = "25%"
116 .TextFrame.Characters.Font.Size = 10
117 .TextFrame.HorizontalAlignment = xlHAlignCenter
118 .Line.Visible = msoFalse
119 .Fill.Visible = msoFalse
120 .Name = "Text2"
121 End With
122 With ActiveSheet.Shapes.AddLine(650, 245, 650, 255)
123 .Line.ForeColor.RGB = RGB(0, 0, 0)
124 .Name = "Line5"
125 End With
126 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
127 625, 262, 40, 20)
128 .TextFrame.Characters.Text = "50%"
129 .TextFrame.Characters.Font.Size = 10
130 .TextFrame.HorizontalAlignment = xlHAlignCenter
131 .Line.Visible = msoFalse
132 .Fill.Visible = msoFalse
133 .Name = "Text3"
134 End With
135 With ActiveSheet.Shapes.AddLine(750, 245, 750, 255)
136 .Line.ForeColor.RGB = RGB(0, 0, 0)
137 .Name = "Line6"
138 End With
139 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
140 725, 262, 40, 20)
141 .TextFrame.Characters.Text = "75%"
142 .TextFrame.Characters.Font.Size = 10
143 .TextFrame.HorizontalAlignment = xlHAlignCenter
144 .Line.Visible = msoFalse
145 .Fill.Visible = msoFalse
146 .Name = "Text4"
147 End With
148 With ActiveSheet.Shapes.AddLine(850, 245, 850, 255)
149 .Line.ForeColor.RGB = RGB(0, 0, 0)
```

```

147     .Name = "Line7"
148 End With
149 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
150     825, 262, 40, 20)
151     .TextFrame.Characters.Text = "100%"
152     .TextFrame.Characters.Font.Size = 10
153     .TextFrame.HorizontalAlignmnent = xlHAlignCenter
154     .Line.Visible = msoFalse
155     .Fill.Visible = msoFalse
156     .Name = "Text5"
157 End With
158 'Find new number of replications
159 If (new_beta > (des_beta * 100)) Then
160     n = 2
161     While (new_beta > (des_beta * 100))
162         new_beta = beta(lambda, CI, n)
163         n = n + 1
164     Wend
165     vMessage3 = "In order to obtain the desired risk value (" &
166     (des_beta * 100) & "%) " & (n - 1) & " replications are needed"
167     Cells(23, 9).Value = vMessage3
168     Cells(23, 9).Font.Size = 14
169 End If
170 Model.Close
171 FileSystem.Kill (vFilePath & "_temp*")
172 End If
173 End Sub

```

2.3.7. Mòdul 7

```

1 Option Explicit
2 Dim CI, alpha As Double
3 Dim Variable As String
4 Dim Y1, Y2, S1, S2, S As Double, n, i As Integer
5 Dim Dm, es, ICi, ICs As Double
6
7 Public Sub Alternative_Comparison_Correlated()
8
9     'Delete previous results
10    Cells(8, 10).Value = Empty
11    Cells(18, 10).Value = Empty
12
13    ActiveSheet.EnableSelection = xlNoSelection
14    i = 1
15    While i <= ActiveSheet.Shapes.Count
16        If (ActiveSheet.Shapes.Item(i).Name = "Line1" Or
17            ActiveSheet.Shapes.Item(i).Name = "Line2" Or
18            ActiveSheet.Shapes.Item(i).Name = "Line3" Or
19            ActiveSheet.Shapes.Item(i).Name = "Line4" Or
20            ActiveSheet.Shapes.Item(i).Name = "Line5" Or
21            ActiveSheet.Shapes.Item(i).Name = "Text1" Or
22            ActiveSheet.Shapes.Item(i).Name = "Text2") Then
23            ActiveSheet.Shapes.Item(i).Delete
24            i = i - 1
25        End If
26        i = i + 1

```

```

21 Wend
22
23 Variable =
24 AC_OutputVariableSelection_Form.AC_OutputVariable.ComboBox.Value
25 'Check if the file exists
26 If (Dir(vFilePath) = "") Then
27 MsgBox "The file introduced for the first model doesn't exist"
28 Else
29 'We generate a copy of the file introduced on "Intro" sheet to do
30 the changes in the model
31 vFilePath = Hoja1.FilePath_TextBox.Value
32 Set ArenaApp = CreateObject("Arena.Application")
33 If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
34 "_temp.doe") <> "") Then
35 ArenaApp.Models.Close
36 End If
37 Set Model = ArenaApp.Models.Open(vFilePath)
38 Model.SaveAs (vFilePath & "_temp.doe")
39 Model.Close
40 Set Model = ArenaApp.Models.Open(vFilePath & "_temp.doe")
41 Set vSIMAN = Model.SIMAN
42 Set vView = Model.ActiveView
43
44 'We run the simulation for the first alternative
45 Model.BatchMode = True
46 Model.QuietMode = True
47 Model.PauseBetweenReplications = True
48 ReDim XOut(1, Model.NumberOfReplications)
49 ReDim XOut1(1, Model.NumberOfReplications)
50 For i = 1 To Model.NumberOfReplications
51 Model.Go
52 XOut(1, i) =
53 vSIMAN.VariableArrayValue(vSIMAN.SymbolNumber(Variable))
54 XOut1(1, i) =
55 vSIMAN.VariableArrayValue(vSIMAN.SymbolNumber(Variable))
56 Next i
57
58 Model.Close
59 FileSystem.Kill (vFilePath & "_temp*")
60 End If
61
62 'We generate a copy of the file with the comparison model
63 vFilePath2 = Hoja5.AC_Model2_Path
64
65 'Check if the file exists
66 If (Dir(vFilePath2) = "") Then
67 MsgBox "The file introduced for the second model doesn't exist"
68 Else
69 Set ArenaApp = CreateObject("Arena.Application")
70 If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
71 "_temp.doe") <> "") Then
72 ArenaApp.Models.Close
73 End If

```

```

73     Set Model = ArenaApp.Models.Open(vFilePath2)
74     Model.SaveAs (vFilePath2 & "_temp.doe")
75     Model.Close
76     Set Model = ArenaApp.Models.Open(vFilePath2 & "_temp.doe")
77     Set vSIMAN = Model.SIMAN
78     Set vView = Model.ActiveView
79
80     'We run the simulation for the second alternative
81     Model.BatchMode = True
82     Model.QuietMode = True
83     Model.PauseBetweenReplications = True
84
85     For i = 1 To Model.NumberOfReplications
86         Model.GoXOut(1, i) =
87             vSIMAN.VariableArrayValue(vSIMAN.SymbolNumber(Variable))
88     Next i
89     Y2 = mean(1)
90     S2 = Sd(1, Y1)
91
92     Model.Close
93     FileSystem.Kill (vFilePath2 & "_temp*")
94     End If
95
96     If (Dir(vFilePath) = "" Or Dir(vFilePath2) = "") Then
97         MsgBox "The file introduced for the second model doesn't exist"
98     Else
99         'We set the parameters for the analysis
100        alpha = Hoja5.AC_CI_TextBox.Value
101
102        'Calculations
103        ReDim D(n)
104        For i = 1 To n
105            D(i) = XOut1(1, i) - XOut(1, i)
106            Dm = Dm + D(i)
107        Next i
108        Dm = Dm / n
109        For i = 1 To n
110            S = S + (D(i) - Dm) ^ 2
111        Next i
112        S = Math.Sqr(S / (n - 1))
113        es = S / Math.Sqr(n)
114        ICI = (Y1 - Y2) - (exl.WorksheetFunction.TInv(alpha, n - 1) * es)
115        ICs = (Y1 - Y2) + (exl.WorksheetFunction.TInv(alpha, n - 1) * es)
116
117        'We draw the confidence intervals
118        With ActiveSheet.Shapes.AddLine(500, 195, 840, 195)
119            .Line.ForeColor.RGB = RGB(0, 0, 0)
120            .Name = "Line1"
121        End With
122        With ActiveSheet.Shapes.AddLine(520, 195, 820, 195)
123            .Line.Weight = xlThick
124            .Line.ForeColor.RGB = RGB(0, 112, 192)
125            .Name = "Line2"
126        End With
127        With ActiveSheet.Shapes.AddLine(520, 190, 520, 200)
128            .Line.ForeColor.RGB = RGB(0, 0, 0)
129            .Name = "Line3"
130        End With

```

```

130 With ActiveSheet.Shapes.AddLine(670, 190, 670, 200)
131 .Line.ForeColor.RGB = RGB(0, 0, 0)
132 .Name = "Line4"
133 End With
134 With ActiveSheet.Shapes.AddLine(820, 190, 820, 200)
135 .Line.ForeColor.RGB = RGB(0, 0, 0)
136 .Name = "Line5"
137 End With
138 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
139 480, 210, 60, 20)
140 .TextFrame.Characters.Text = Round(ICi, 2)
141 .TextFrame.Characters.Font.Size = 12
142 .TextFrame.HorizontalAlignment = xlHAlignCenter
143 .Line.Visible = msoFalse
144 .Fill.Visible = msoFalse
145 .Name = "Text1"
146 End With
147 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
148 780, 210, 60, 20)
149 .TextFrame.Characters.Text = Round(ICs, 2)
150 .TextFrame.Characters.Font.Size = 12
151 .TextFrame.HorizontalAlignment = xlHAlignCenter
152 .Line.Visible = msoFalse
153 .Fill.Visible = msoFalse
154 .Name = "Text2"
155 End With
156 Cells(18, 10).Font.Bold = True
157 If (ICi < 0) Then
158 If (ICs > 0) Then
159 Cells(18, 10).Value = "The two models can't be considered as
different"
160 Else
161 Cells(18, 10).Value = "The two models can be considered as
different"
162 End If
163 Else
164 Cells(18, 10).Value = "The two models can't be considered as
different"
165 End If
166 End If
167 End Sub

```

2.3.8. Mòdul 8

```

1 Option Explicit
2 Dim Queue_Area, alpha As Double
3 Dim exl As New Excel.Application
4 Dim Variable As String
5 Dim Y1, Y2, S1, S2 As Double, n1, n2, i As Integer
6 Dim EqualVar As Boolean, F As Double
7 Dim es, Sp, ICi, ICs As Double, u As Integer
8
9 Public Sub Alternative_Comparison_Independent()

```

```

10   'Delete previous results
11  Cells(8, 10).Value = Empty
12  Cells(18, 10).Value = Empty
13
14  ActiveSheet.EnableSelection = xlNoSelection
15  i = 1
16  While i <= ActiveSheet.Shapes.Count
17  If (ActiveSheet.Shapes.Item(i).Name = "Line1" Or
18    ActiveSheet.Shapes.Item(i).Name = "Line2" Or
19    ActiveSheet.Shapes.Item(i).Name = "Line3" Or
20    ActiveSheet.Shapes.Item(i).Name = "Line4" Or
21    ActiveSheet.Shapes.Item(i).Name = "Line5" Or
22    ActiveSheet.Shapes.Item(i).Name = "Text1" Or
23    ActiveSheet.Shapes.Item(i).Name = "Text2") Then
24    ActiveSheet.Shapes.Item(i).Delete
25  i = i - 1
26  End If
27  i = i + 1
28 Wend
29
30 Variable =
31 AC_OutputVariableSelection_Form.AC_OutputVariable.ComboBox.Value
32
33 'We generate a copy of the file introduced on "Intro" sheet to do
34 the changes in the model
35 'Check if the file exists
36 If (Dir(vFilePath) = "") Then
37 MsgBox "The file introduced for the first model doesn't exist"
38 Else
39 vFilePath = Hoja1.FilePath_TextBox.Value
40 Set ArenaApp = CreateObject("Arena.Application")
41 If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
42 "_temp.doe") <> "") Then
43 ArenaApp.Models.Close
44 End If
45 Set Model = ArenaApp.Models.Open(vFilePath)
46 Model.SaveAs (vFilePath & "_temp.doe")
47 Model.Close
48 Set Model = ArenaApp.Models.Open(vFilePath & "_temp.doe")
49 Set vSIMAN = Model.SIMAN
50 Set vView = Model.ActiveView
51
52 'We run the simulation for the first alternative
53 Model.BatchMode = True
54 Model.QuietMode = True
55 Model.PauseBetweenReplications = True
56 ReDim XOut(1, Model.NumberOfReplications)
57 ReDim XOut1(1, Model.NumberOfReplications)
58 For i = 1 To Model.NumberOfReplications
59 Model.Go
60 XOut(1, i) =
61 vSIMAN.VariableArrayValue(vSIMAN.SymbolNumber(Variable))
62 XOut1(1, i) =
63 vSIMAN.VariableArrayValue(vSIMAN.SymbolNumber(Variable))
64 Next i

```

```

54 n1 = Model.NumberOfReplications
55 Y1 = mean(1)
56 S1 = Sd(1, Y1)
57
58 Model.Close
59 FileSystem.Kill (vFilePath & "_temp*")
60 End If
61
62 'We generate a copy of the file with the comparison model
63 vFilePath2 = Hoja5.AC_Model2_Path
64
65 'Check if the file exists
66 If (Dir(vFilePath2) = "") Then
67 MsgBox "The file introduced for the second model doesn't exist"
68 Else
69 Set ArenaApp = CreateObject("Arena.Application")
70 If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
71 "& _temp.doe") <> "") Then
72 ArenaApp.Models.Close
73 End If
74 Set Model = ArenaApp.Models.Open(vFilePath2)
75 Model.SaveAs (vFilePath2 & "_temp.doe")
76 Model.Close
77 Set Model = ArenaApp.Models.Open(vFilePath2 & "_temp.doe")
78 Set vSIMAN = Model.SIMAN
79 Set vView = Model.ActiveView
80
81 'We run the simulation for the second alternative
82 Model.BatchMode = True
83 Model.QuietMode = True
84 Model.PauseBetweenReplications = True
85 For i = 1 To Model.NumberOfReplications
86 Model.Go
87 XOut(1, i) =
88 vSIMAN.VariableArrayValue(vSIMAN.SymbolNumber(Variable))
89 Next i
90 n2 = Model.NumberOfReplications
91 Y2 = mean(1)
92 S2 = Sd(1, Y1)
93
94 Model.Close
95 FileSystem.Kill (vFilePath2 & "_temp*")
96 End If
97
98 If (Dir(vFilePath) = "" Or Dir(vFilePath2) = "") Then
99 MsgBox "The file introduced for the second model doesn't exist"
100 Else
101 'We set the parameters for the analysis
102 alpha = Hoja5.AC_CI_TextBox.Value
103 Queue_Area = 1 - Hoja5.AC_Alpha_TextBox.Value
104
105 'We need to know whether we are working with equal variances or not
106 If S2 > S1 Then
107 F = (S2 ^ 2) / (S1 ^ 2)
108 EqualVar = F < (exl.WorksheetFunction.FInv(Queue_Area, (n1 - 1),
109 (n2 - 1)))
110 Else
111 F = (S1 ^ 2) / (S2 ^ 2)

```

```

109 EqualVar = F < (exl.WorksheetFunction.FInv(Queue_Area, (n2 - 1),
110 (n1 - 1)))
111 End If
112 Cells(8, 10).Font.Size = 12
113 If EqualVar Then 'Case equal variances
114 Cells(8, 10).Value = "We can assume equal variances"
115 u = n1 + n2 - 2
116 Sp = (((n1 - 1) * (S1 ^ 2)) + ((n2 - 1) * (S2 ^ 2))) / u
117 es = Sp * Math.Sqrt((1 / n1) + (1 / n2))
118 ICi = (Y1 - Y2) - (exl.WorksheetFunction.TInv(alpha, u) * es)
119 ICs = (Y1 - Y2) + (exl.WorksheetFunction.TInv(alpha, u) * es)
120 Else ' Different Variances
121 Cells(8, 10).Value = "We can't assume equal variances"
122 u = (((((S1 ^ 2) / n1) + ((S2 ^ 2) / n2)) ^ 2) / (((((S1 ^ 2) / n1)
123 ^ 2) / (n1 - 1)) + (((((S2 ^ 2) / n2) ^ 2) / (n2 - 1))))
124 es = Math.Sqrt(((S1 ^ 2) / n1) + ((S2 ^ 2) / n2))
125 ICi = (Y1 - Y2) - (exl.WorksheetFunction.TInv(alpha, u) * es)
126 ICs = (Y1 - Y2) + (exl.WorksheetFunction.TInv(alpha, u) * es)
127 End If
128 'We draw the confidence intervals
129 With ActiveSheet.Shapes.AddLine(500, 195, 840, 195)
130 .Line.ForeColor.RGB = RGB(0, 0, 0)
131 .Name = "Line1"
132 End With
133 With ActiveSheet.Shapes.AddLine(520, 195, 820, 195)
134 .Line.Weight = xlThick
135 .Line.ForeColor.RGB = RGB(0, 112, 192)
136 .Name = "Line2"
137 End With
138 With ActiveSheet.Shapes.AddLine(520, 190, 520, 200)
139 .Line.ForeColor.RGB = RGB(0, 0, 0)
140 .Name = "Line3"
141 End With
142 With ActiveSheet.Shapes.AddLine(670, 190, 670, 200)
143 .Line.ForeColor.RGB = RGB(0, 0, 0)
144 .Name = "Line4"
145 End With
146 With ActiveSheet.Shapes.AddLine(820, 190, 820, 200)
147 .Line.ForeColor.RGB = RGB(0, 0, 0)
148 .Name = "Line5"
149 End With
150 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
151 480, 210, 60, 20)
152 .TextFrame.Characters.Text = Round(ICi, 2)
153 .TextFrame.Characters.Font.Size = 12
154 .TextFrame.Horizontal_ALIGNMENT = xlHAlignCenter
155 .Line.Visible = msoFalse
156 .Fill.Visible = msoFalse
157 .Name = "Text1"
158 End With
159 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
160 780, 210, 60, 20)
161 .TextFrame.Characters.Text = Round(ICs, 2)
162 .TextFrame.Characters.Font.Size = 12
163 .TextFrame.Horizontal_ALIGNMENT = xlHAlignCenter
164 .Line.Visible = msoFalse

```

```

163 .Fill.Visible = msoFalse
164 .Name = "Text2"
165 End With
166
167 Cells(18, 10).Font.Bold = True
168 If (ICi < 0) Then
169 If (ICs > 0) Then
170 Cells(18, 10).Value = "The two models can't be considered as
different"
171 Else
172 Cells(18, 10).Value = "The two models can be considered as
different"
173 End If
174 Else
175 Cells(18, 10).Value = "The two models can't be considered as
different"
176 End If
177 End If
178 End Sub

```

2.3.9. Mòdul 9

```

1 Option Explicit
2 Dim alpha, t, MaxIChigh, MinICLow, ScaleFactor As Double
3 Dim i, j, k, n(), Experiments, Row, Count, r As Integer
4 Dim TextChain, Variable, vEffectResult As String
5
6 Public Sub Factorial_Design()
7 'Check if the file exists
8 If (Dir(vFilePath) = "") Then
9 MsgBox "The file introduced doesn't exist"
10 Else
11 'We generate a copy of the file introduced on "Intro" sheet to do
the changes in the model
12 vFilePath = Hojal.FilePath_TextBox.Value
13 Set ArenaApp = CreateObject("Arena.Application")
14 If (Dir(vFilePath & "_temp.doe") <> "" Or Dir(vFilePath2 &
"_temp.doe") <> "") Then
15 ArenaApp.Models.Close
16 End If
17 Set Model = ArenaApp.Models.Open(vFilePath)
18 Model.SaveAs (vFilePath & "_temp.doe")
19 Model.Close
20 Set Model = ArenaApp.Models.Open(vFilePath & "_temp.doe")
21 Set vSIMAN = Model.SIMAN
22 Set vView = Model.ActiveView
23
24 'Delete previous results
25 Columns("H:J").Select
26 Selection.ClearContents
27
28 ActiveSheet.EnableSelection = xlNoSelection
29 i = 1
30 While i <= ActiveSheet.Shapes.Count
31 If (Left(ActiveSheet.Shapes.Item(i).Name, 5) = "Line1" Or
Left(ActiveSheet.Shapes.Item(i).Name, 5) = "Line2" Or
Left(ActiveSheet.Shapes.Item(i).Name, 5) = "Line3" Or

```

```

1    Left(ActiveSheet.Shapes.Item(i).Name, 5) = "Text1" Or
2    Left(ActiveSheet.Shapes.Item(i).Name, 5) = "Text2" Or
3    Left(ActiveSheet.Shapes.Item(i).Name, 6) = "Effect" Or
4    Left(ActiveSheet.Shapes.Item(i).Name, 6) = "Result") Then
5        ActiveSheet.Shapes.Item(i).Delete
6        i = i - 1
7    End If
8    i = i + 1
9    Wend
10   Variable =
11   FD_OutputVariableSelection_Form.FD_OutputVariable_ComboBox.Value
12   Experiments = FD_Definition_Form.FD_Experiments_Listbox.ListCount
13   r = Model.NumberOfReplications
14
15   'Construct the matrix that will be the guide for execute the
16   'different simulations
17   ReDim M(2 ^ Experiments, Experiments)
18   For i = 1 To Experiments
19       Row = 1
20       Count = 0
21       While Count < 2 ^ (Experiments - 1)
22           For j = 1 To (2 ^ Experiments) / 2 ^ i
23               M(Row, i) = 3
24               M(Row + (2 ^ Experiments) / (2 ^ i), i) = 4
25               Count = Count + 1
26               Row = Row + 1
27           Next j
28           For j = 1 To (2 ^ Experiments) / 2 ^ i
29               Row = Row + 1
30           Next j
31       Wend
32       Next i
33
34   'Construct a matrix with the factors and its parameters
35   ReDim
36   FactorsMatrix(FD_Definition_Form.FD_Experiments_Listbox.ListCount,
37   4)
38   ReDim
39   ParametersPosition(FD_Definition_Form.FD_Experiments_Listbox.ListCo
40   unt, 3)
41   For i = 0 To Experiments - 1
42       TextChain = FD_Definition_Form.FD_Experiments_Listbox.List(i)
43       ParametersPosition(i, 0) = InStr(TextChain, ",")
44       For j = 1 To 2
45           ParametersPosition(i, j) = InStr(ParametersPosition(i, j - 1) + 1,
46           TextChain, ",")
47       Next j
48       FactorsMatrix(i, 1) = FactorMatrix(i)
49       FactorsMatrix(i, 2) = Trim(Mid(TextChain, ParametersPosition(i, 0)
50       + 1, ParametersPosition(i, 1) - ParametersPosition(i, 0) - 1))
51       FactorsMatrix(i, 3) = Mid(TextChain, ParametersPosition(i, 1) + 1,
52       ParametersPosition(i, 2) - ParametersPosition(i, 1) - 1)
53       FactorsMatrix(i, 4) = Right(TextChain, Len(TextChain) -
54       ParametersPosition(i, 2))
55   Next i
56
57   ReDim n(Experiments)
58   ReDim Yh(Experiments)

```

```

76  ReDim Sh(Experiments)
77  ReDim Y1(Experiments)
78  ReDim S1(Experiments)
79
80  'Run the simulations for all combinations and save the results
81  ReDim XOutAux(2 ^ Experiments, r)
82
83  Model.BatchMode = True
84  Model.QuietMode = True
85  Model.PauseBetweenReplications = True
86
87  'Apply the configuration to the model
88  For j = 1 To 2 ^ Experiments
89    For i = 1 To Experiments
90      Model.Modules(FactorsMatrix(i - 1, 1)).Data(FactorsMatrix(i - 1,
91        2)) = CStr(FactorsMatrix(i - 1, M(j, i)))
92    Next i
93  Model.Save
94  For k = 1 To r
95    Model.Go
96    XOutAux(j, k) =
97      vSIMAN.VariableArrayValue(vSIMAN.SymbolNumber(Variable))
98  Next k
99  Model.End
100 Next j
101
102 ReDim XOut(Experiments, r)
103 For i = 1 To Experiments
104   For k = 1 To r
105     Row = 1
106     Count = 0
107     While Count < 2 ^ (Experiments - 1)
108       For j = 1 To (2 ^ Experiments) / 2 ^ i
109         XOut(i, k) = XOut(i, k) + XOutAux(Row, k) - XOutAux(Row + (2 ^
110           Experiments) / (2 ^ i), k)
111       Count = Count + 1
112     Row = Row + 1
113   Next j
114   For j = 1 To (2 ^ Experiments) / 2 ^ i
115     Row = Row + 1
116   Next j
117   XOut(i, k) = XOut(i, k) / 2 ^ (Experiments - 1)
118 Wend
119 Next k
120 Next i
121
122 'Calculate the confidence interval for each experiment
123 ReDim Y(Experiments)
124 ReDim S(Experiments)
125 ReDim ICHigh(Experiments)
126 ReDim ICLow(Experiments)
127
128 alpha = Hoja6.FD_alpha_TextBox.Value
129 t = exl.WorksheetFunction.TInv((1 - alpha), r - 1)
130 For i = 1 To Experiments
131   Y(i) = mean(i)
132   S(i) = Sd(i, Y(i))

```

```

131 ICHigh(i) = Y(i) + t * S(i) / (Math.Sqrt(r))
132 ICLow(i) = Y(i) - t * S(i) / (Math.Sqrt(r))
133
134 If (ICHHigh(i) > 0 And ICLow(i) < 0) Then
135 vEffectResult = "Non significant effect"
136 Else
137 vEffectResult = "Significant effect"
138 End If
139
140 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
141 425, 30 + (100 * i), 75, 20)
142 .TextFrame.Characters.Text = "Effect " & i
143 .TextFrame.Characters.Font.Size = 14
144 .TextFrame.Characters.Font.Bold = True
145 .TextFrame.HorizontalAlignment = xlHAlignLeft
146 .Line.Visible = msoFalse
147 .Fill.Visible = msoFalse
148 .Name = "Effect" & CStr(i)
149 End With
150
151 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,
152 500, 30 + (100 * i), 150, 20)
153 .TextFrame.Characters.Text = vEffectResult
154 .TextFrame.Characters.Font.Size = 14
155 .TextFrame.HorizontalAlignment = xlHAlignLeft
156 .Line.Visible = msoFalse
157 .Fill.Visible = msoFalse
158 .Name = "Result" & CStr(i)
159 End With
160 Next i
161
162 MaxICHHigh = ICHHigh(1)
163 MinICLow = ICLow(1)
164 For i = 2 To Experiments
165 If (ICHHigh(i) > MaxICHHigh) Then
166 MaxICHHigh = ICHHigh(i)
167 End If
168 If (ICLow(i) < MinICLow) Then
169 MinICLow = ICLow(i)
170 End If
171 Next i
172 If (MaxICHHigh < 0) Then
173 MaxICHHigh = 0
174 If (MinICLow > 0) Then
175 MinICLow = 0
176 End If
177 End If
178 ScaleFactor = 400 / (MaxICHHigh - MinICLow)
179
180 'Draw the confidence intervals
181 For i = 1 To Experiments
182 With ActiveSheet.Shapes.AddLine(425, 75 + (100 * i), 825, 75 + (100
* i))
183 .Line.ForeColor.RGB = RGB(0, 0, 0)
184 .Name = "Line1" & CStr(i)
185 End With

```

```
185 With ActiveSheet.Shapes.AddLine(425 + ScaleFactor * (ICLow(i) -  
MinICLow), 75 + (100 * i), 425 + ScaleFactor * (ICLow(i) -  
MinICLow) + ScaleFactor * (ICHIGH(i) - ICLow(i)), 75 + (100 * i))  
186 .Line.Weight = xlThin  
187 .Line.ForeColor.RGB = RGB(0, 112, 192)  
188 .Name = "Line2" & CStr(i)  
189 End With  
190  
191 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,  
410 + ScaleFactor * (ICLow(i) - MinICLow), 77 + (100 * i), 40, 20)  
192 .TextFrame.Characters.Text = CStr(Round(ICLow(i), 2))  
193 .TextFrame.Characters.Font.Size = 10  
194 .TextFrame.HorizontalAlignment = xlHAlignCenter  
195 .Line.Visible = msoFalse  
196 .Fill.Visible = msoFalse  
197 .Name = "Text1" & CStr(i)  
198 End With  
199  
200 With ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal,  
410 + ScaleFactor * (ICLow(i) - MinICLow) + ScaleFactor *  
(ICHIGH(i) - ICLow(i)), 77 + (100 * i), 40, 20)  
201 .TextFrame.Characters.Text = CStr(Round(ICHigh(i), 2))  
202 .TextFrame.Characters.Font.Size = 10  
203 .TextFrame.HorizontalAlignment = xlHAlignCenter  
204 .Line.Visible = msoFalse  
205 .Fill.Visible = msoFalse  
206 .Name = "Text2" & CStr(i)  
207 End With  
208  
209 With ActiveSheet.Shapes.AddLine(425 + ScaleFactor * (ICLow(i) -  
MinICLow) + ScaleFactor * (ICHIGH(i) - ICLow(i)) / 2, 70 + (100 *  
i), 425 + ScaleFactor * (ICLow(i) - MinICLow) + ScaleFactor *  
(ICHIGH(i) - ICLow(i)) / 2, 80 + (100 * i))  
210 .Line.ForeColor.RGB = RGB(0, 0, 0)  
211 .Name = "Line3" & CStr(i)  
212 End With  
213  
214 Next i  
215 Model.Close  
216 FileSystem.Kill (vFilePath & "_temp*")  
217 End If  
218 End Sub
```

