

Compte rendu de stage semaine 1 :

Rappel du projet :

Création d'une application de comptabilité avec visual basic.

Jour 1 (Lundi 26 mai) :

Visite du lieu de stage puis explication de ma mission. Découverte du code vba de l'ancien stagiaire.

Jour 2 (Mardi 27 mai) :

Décomposition du code salaire query + Logigramme.

Programme déjà existant (pas fait moi-même) qui extrait les données d'une table et qui renvoi les informations suivantes :

- Cle
- Nom
- Date initiale de l'engagement
- Rubrique de la pièce
- Ctre fin
- Centre de coûts
- Elt OTP
- Cmp.Ex.Bud
- Fournisseur
- Nom du tiers
- Libellé du flux
- Cde N-1
- CDE
- SF
- date SF
- Facturation
- Date Facturation
- Payé
- Date payé
- Statut (En cours, Soldés, Annulé, Partiel).

Les paiements en cours ont deux couleurs en fonction de leur retard (Orange ou Rouge).

Résumé détaillé du code :

1. Déclaration des variables et des chemins de dossiers :

```
Sub CreerResultatPowerQuery()  
    Dim TabGes As Variant  
  
    'Définition des chemins des dossiers  
    filepathMaj = "Y:\GAF\Shared\Situation Saic\EXTRACT_SIFAC\ZSUIVI_EXEC2024.txt"  
    filepathGes = "Y:\gestion_financiere\Documentation\Compta\INPT\compta 2005\Gestionnaires\" ' Chemin du dépôt gestionnaire  
  
    datee = 2025 'Year(Date)
```

2. Récupération du contenu de Resultat query :

```
On va chercher les données dans Resultat query  
  
Colonnes'  
ocol_TabResultat = ThisWorkbook.Sheets("Resultat query").Cells(1, ThisWorkbook.Sheets("Resultat query").Columns.Count).End(xlToLeft).Column  
Lignes'  
olig_TabResultat = (ThisWorkbook.Sheets("Resultat query").Cells(ThisWorkbook.Sheets("Resultat query").Rows.Count, 1).End(xlUp).Row)
```

3. Récupération des gestionnaires depuis Annuaire.

```

nblig_ges = 20
'Recupère les gestionnaires de la feuille annuaire
ReDim TabGes(nblig_ges, 1)|
For i = 2 To nblig_ges
    TabGes(i, 1) = Sheets("Annuaire").Cells(i, "AC")
Next i

```

4. Création d'un nouveau tableau

```

ReDim TabNouveau(1 To nblig_TabResultat, 1 To nbcol_TabResultat)
For k = 2 To UBound(TabGes, 1)
    Application.DisplayAlerts = False

```

5. Ajout des Gestionnaires depuis TabGes

```

For k = 2 To UBound(TabGes, 1)
    Application.DisplayAlerts = False
    nom = TabGes(k, 1)
    'Remplace les accents et caractères spéciaux..
    nom = Replace(nom, "é", "e")
    nom = Replace(nom, "ç", "c")
    nom = Replace(nom, "ë", "e")
    If Dir(filePathGes & "Situation SAIC " & datee & "\" & nom & " " & datee & ".xlsx") <> "" Then ' Si il y a déjà un fichier au dépôt
        Set Classeur = Workbooks.Open(filePathGes & "Situation SAIC " & datee & "\" & nom & " " & datee & ".xlsx", UpdateLinks:=False)
    Else
        GoTo Skip
    End If
    Set feuilleSource = Classeur.Sheets(1)
    DerniereLigne = feuilleSource.Cells(feuilleSource.Rows.Count, 1).End(xlUp).Row

    ' Récupère tout le tableau du fichier
    dataSource = feuilleSource.Range(feuilleSource.Cells(1, 1), feuilleSource.Cells(DerniereLigne, nbcol_TabResultat)).Value

```

6. Mise à jour des données dans TabRésultat

```

If Dir(filePathGes & "Situation SAIC " & datee & "\" & nom & " " & datee & ".xlsx") <> "" Then ' Si il y a déjà un fichier au dépôt
    Set Classeur = Workbooks.Open(filePathGes & "Situation SAIC " & datee & "\" & nom & " " & datee & ".xlsx", UpdateLinks:=False)
Else
    GoTo Skip
End If
Set feuilleSource = Classeur.Sheets(1)
DerniereLigne = feuilleSource.Cells(feuilleSource.Rows.Count, 1).End(xlUp).Row

' Récupère tout le tableau du fichier
dataSource = feuilleSource.Range(feuilleSource.Cells(1, 1), feuilleSource.Cells(DerniereLigne, nbcol_TabResultat)).Value

```

7. Détermination de la dernière ligne et colonne et transformation de cette la feuille en tableau

```

'On transforme la feuille en tableau'
Set feuilleSource = Classeur.Sheets(1)
'On définit la dernière ligne de ce tableau'
DerniereLigne = feuilleSource.Cells(feuilleSource.Rows.Count, 1).End(xlUp).Row

' Récupère tout le tableau du fichier
dataSource = feuilleSource.Range(feuilleSource.Cells(1, 1), feuilleSource.Cells(DerniereLigne, nbcol_TabResultat)).Value

```

8. Ajout du total dans la dernière ligne du tableau

```

' Mise en place des totaux en derniere ligne
With Sheets("Resultat")
    .Cells(DerniereLigne + 1, 35).FormulaR1C1 = "=SUBTOTAL(9, R2C:R[-1]C)"
    .Cells(DerniereLigne + 1, 36).FormulaR1C1 = "=SUBTOTAL(9, R2C:R[-1]C)"
    .Cells(DerniereLigne + 1, 37).FormulaR1C1 = "=SUBTOTAL(9, R2C:R[-1]C)"
    .Cells(DerniereLigne + 1, 39).FormulaR1C1 = "=SUBTOTAL(9, R2C:R[-1]C)"
    .Cells(DerniereLigne + 1, 41).FormulaR1C1 = "=SUBTOTAL(9, R2C:R[-1]C)"
End With

```

9. Conversion du texte en date là où c'est nécessaire.

```

Range("AL3:AL" & DerniereLigne).TextToColumns Destination:=Feuille.Range("AL3:AL" & DerniereLigne), DataType:=xlDelimited, TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, Semicolon:=False, Comma:=False, _
Space:=False, Other:=False, FieldInfo:=Array(1, xlDMYFormat), _
TrailingMinusNumbers:=True

Range("AN3:AN" & DerniereLigne).TextToColumns Destination:=Feuille.Range("AN3:AN" & DerniereLigne), DataType:=xlDelimited, TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, Semicolon:=False, Comma:=False, _
Space:=False, Other:=False, FieldInfo:=Array(1, xlDMYFormat), _
TrailingMinusNumbers:=True

Range("AP3:AP" & DerniereLigne).TextToColumns Destination:=Feuille.Range("AP3:AP" & DerniereLigne), DataType:=xlDelimited, TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, Semicolon:=False, Comma:=False, _
Space:=False, Other:=False, FieldInfo:=Array(1, xlDMYFormat), _
TrailingMinusNumbers:=True

```

10. Enregistrement du nouveau classeur.

```

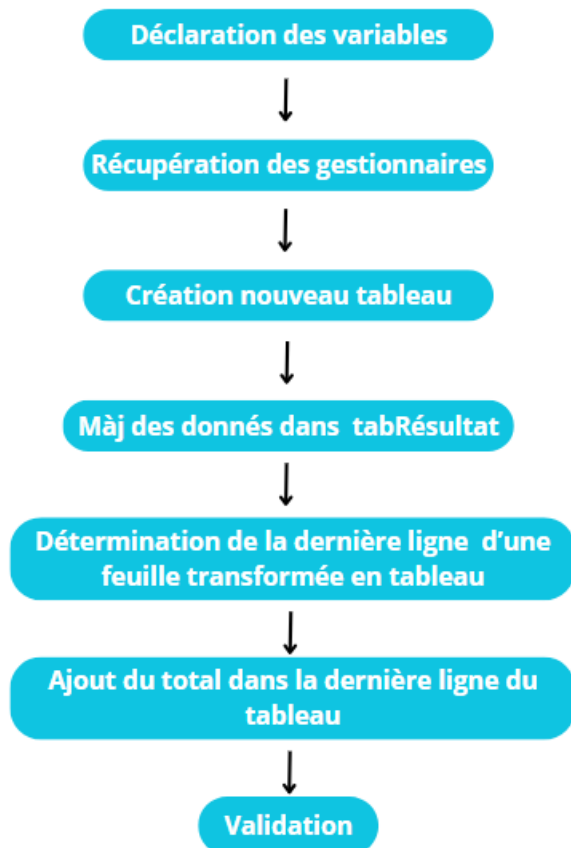
' Enregistre le nouveau classeur
Application.DisplayAlerts = False
Classeur.Sheets("Resultat").SaveAs filePath & "TabResultAncien" & datee & ".xlsx"
Application.DisplayAlerts = True

Classeur.Close SaveChanges:=False
End Sub

```

Logigramme du code :

Logigramme code (Résult Query)



Installation d'autoIt pour la création de script d'automatisation
Création d'un script d'automatisation avec auto it.

Jour 3 (Mercredi 28 mai) :

Accès à geslab finalisé.

Geslab = logiciel de suivi financier. Onglet éditions qui permet d'extraire des données selon des critères précis.

Liste des états

Domaine	Catégorie	Etat
Budget	Origines	ORI_1 : Bilan des origines (Niveau fin)
		ORI_2 : Bilan d'une origine de regroupement
		SOR_2 : Synthèse d'une origine - Historique des commandes et des factures
	Entités dépen­sières	EDP_1 : Bilan des entités dépen­sières de niveau fin
		EDP_2 : Bilan des dépenses par entités dépen­sières (Regroupement)
		EDP_3 : Bilan financier d'une entité dépen­sière par origine
		SED_2 : Synthèse d'une entité dépen­sière - Historique des commandes et des factures
	Justifications	JUS_1 : Préparation à la justification financière ou au montage de dossier de tarification
		JUS_2 : Préparation à la justification financière ou au montage de dossier de tarification (flux 4)
Dépenses	Achats	CAR_1 : Multi-Exercice - Etat récapitulatif par UF, OT, NB
		COM_1 : Commandes par origine
		COM_2 : Commandes par entité dépen­sière
		COM_3 : Commandes partiellement facturées et/ou commandes certifiées non facturées
		EDP_4 : Dépenses par entité dépen­sière et par matière
		FAC_1 : Factures par origine
		FAC_2 : Factures par entité dépen­sière
		LDE_1 : Détail des lignes de commande
		LDE_2 : Suivi des services faits par ligne commande
	Missions	MIS_1_2 : Multi-Exercice - Récapitulatif des dépenses de mission par agent (4 ans)
		MIS_3 : Liste des frais de mission par entité dépen­sière
		MIS_4 : Liste des missions pour calculer les émissions CO2
	Nature d'achats	NAT_2 : Bilan des dépenses par entité dépen­sière et par nature
Inventaire	Inventaire sous GESLAB "Web"	INV_1 : Multi-Exercice - Liste de l'inventaire
Fin D'exercice	Pré-Bascule	COM_FDX : Liste des commandes par anticipation
	Post-Bascule	REP_FDX : Liste des répartitions de reports en attente

Extraction des données utilisées par le projet. L'application vba devra mettre en forme les données dans des tableau et faciliter leur manipulation.

Le jeudi et vendredi sont fériés marquant la fin de la première semaine de stage.

Compte rendu de stage semaine 2 :

Rappel du projet : Création d'une application de comptabilité avec visual basic.

Jour 1 (lundi 2 juin):

Manipulation de power query pour fusionner différentes données extraites de geslab.

Création de code vba basic pour : séparer les tables par utilisateurs, effacer les modifications du tableur, actualiser les connexions power query.

Jour 2 (mardi 3 juin):

Amélioration du code VBA de hier, ajout d'exportation d'utilisateur en feuilles séparées vers des dossiers choisis par l'utilisateur, création d'un UserForm qui permettra de faire des recherches plus précises.

Ces macros VBA se sont faites sur un classeur excel qui ne sera pas celui de l'application finale. Il s'agissait d'un entraînement et d'une prise en main de VBA.

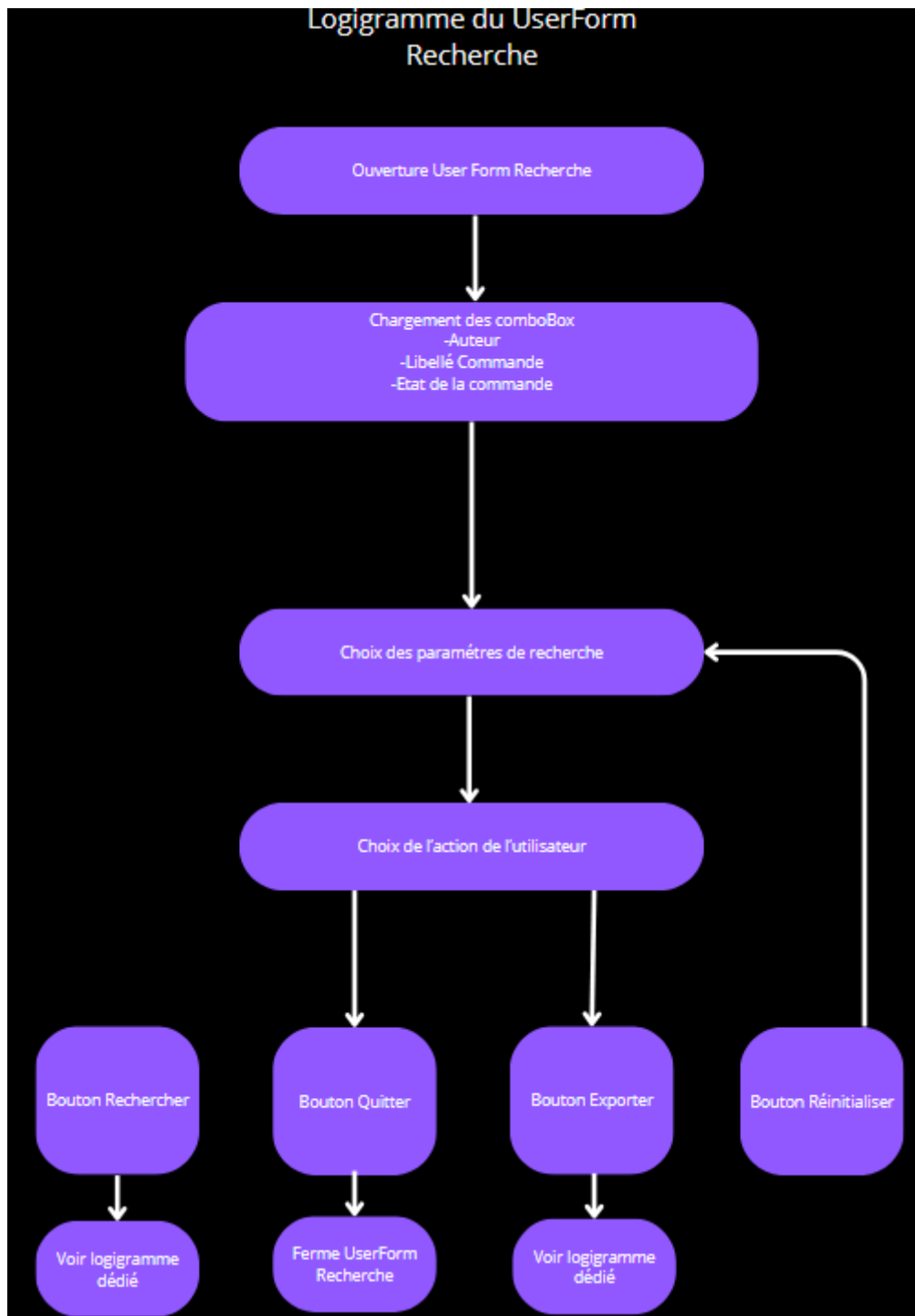
Les macros finales seront faites sur un classeur que je ne me mettrai pas en forme, une de mes tutrices de stages le fera. Je ne ferai que la partie VBA quand j'y aurai accès.

Jour 3 (mercredi 5 juin):

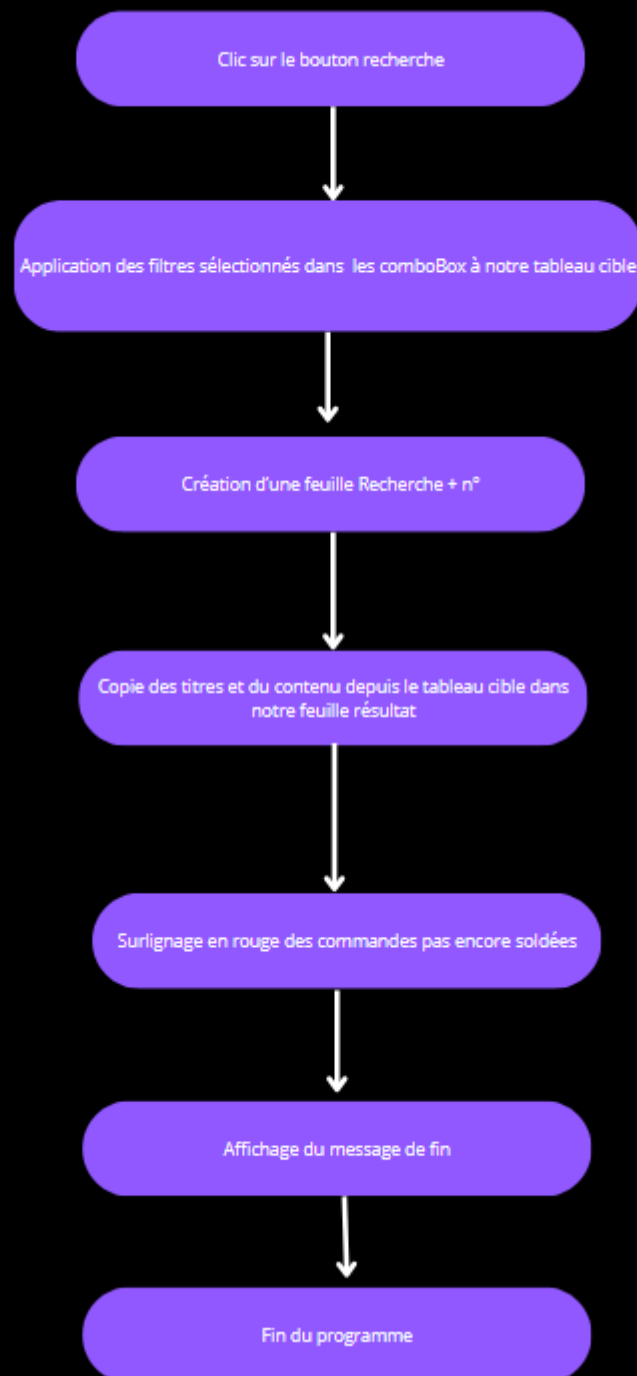
Création de logigramme pour mon userForm pour mieux comprendre son fonctionnement.

Après consultation l'idée du UserForm a été abandonnée trop lente et n'apporte pas de réelle utilité à l'application finale.

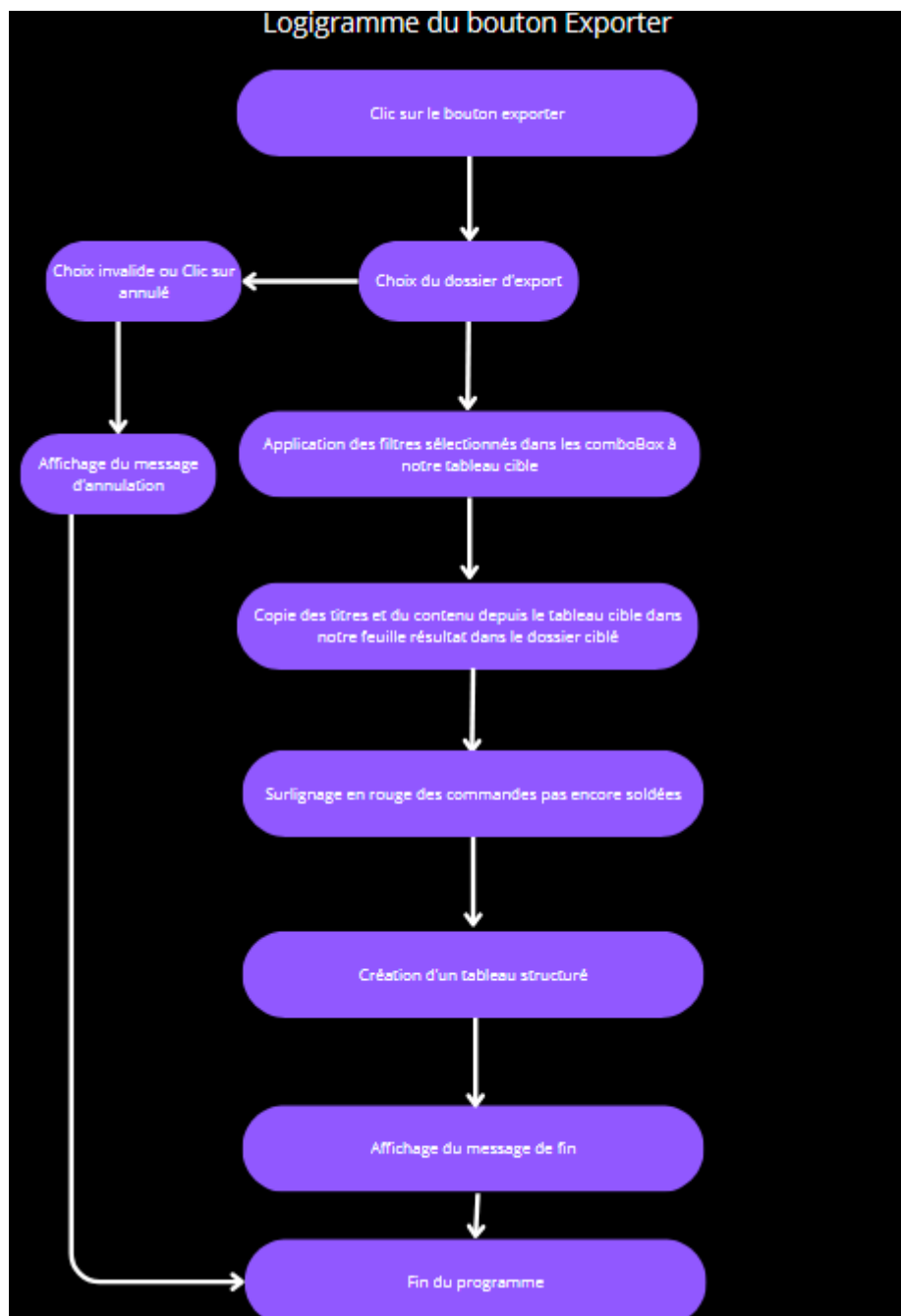
Logigramme du User Form (il contient deux boutons qui lancent d'autres processus avec leur propre logigramme) :



Logigramme du bouton Recherche



Logigramme du bouton Exporter



Jour 4 (jeudi 5 juin):

Création d'un logigramme pour toutes les macros de l'ancien stagiaire avant adaptation à ma mission :

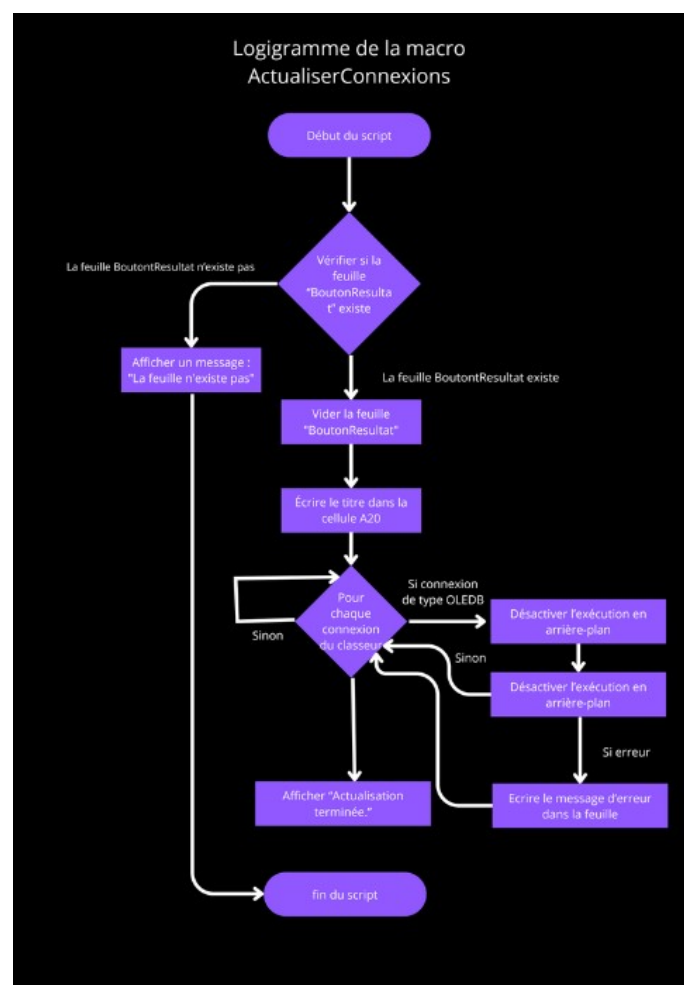
- envoi Mail,
- Actualisation,
- Power Query
- Interface
- Eclater les conventions

Cette étape peut sembler inutile mais j'éprouve vraiment des difficultés à comprendre le code de l'ancien stagiaire pas parce qu'il est mal annoté ou commenté n'y parce que mes responsables expliquent mal, mais parce que les concepts sont un peu compliqués pour moi, la construction de logigrammes détaillés va me permettre d'écrire plus facilement mon propre code.

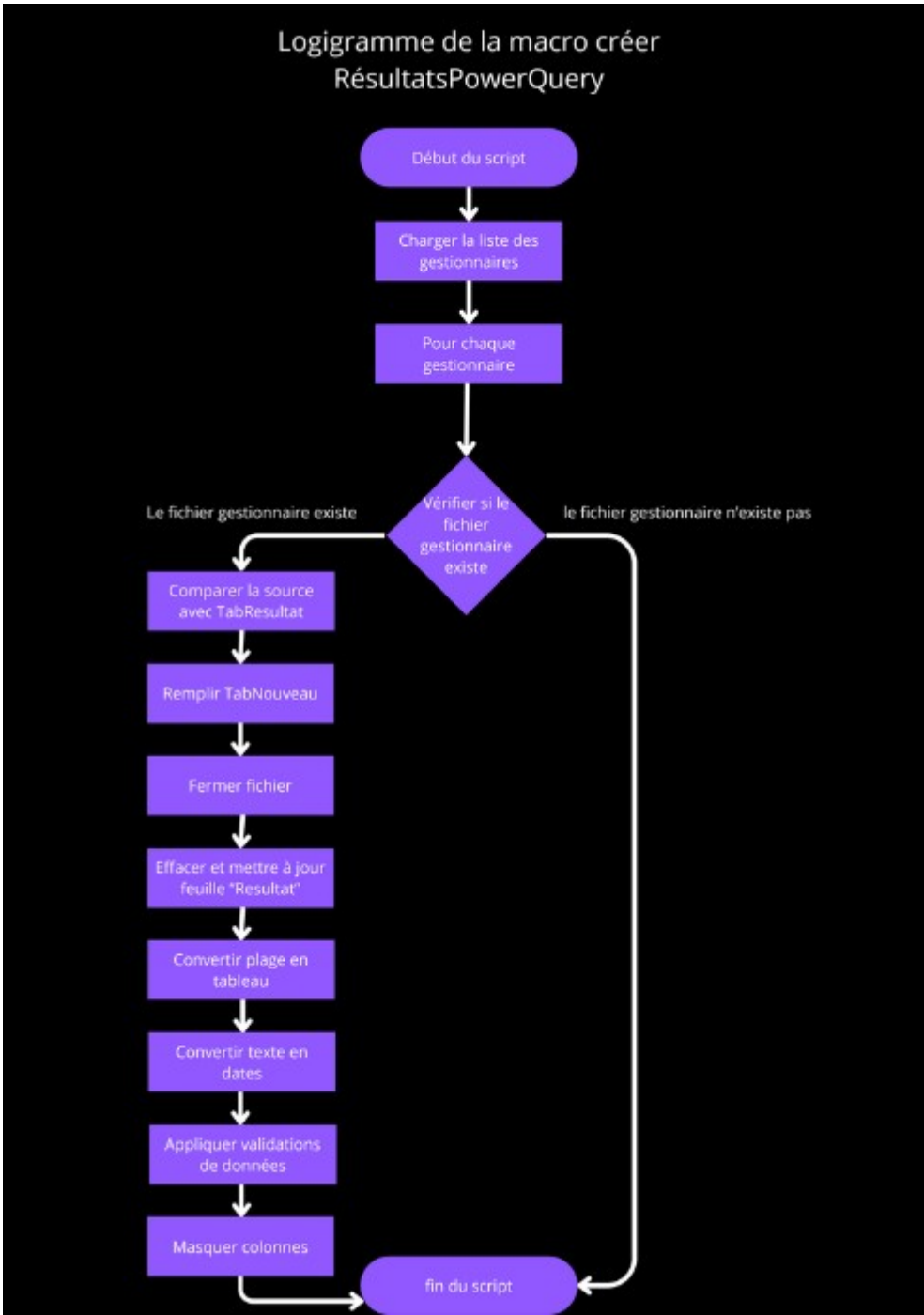
J'ai aussi modifié la nomenclature de mes logigrammes

- Ovale = début ou fin d'un processus.
- Rectangle = étape d'un processus
- Rectangle orange = Boucle
- Losange = choix dans le processus.

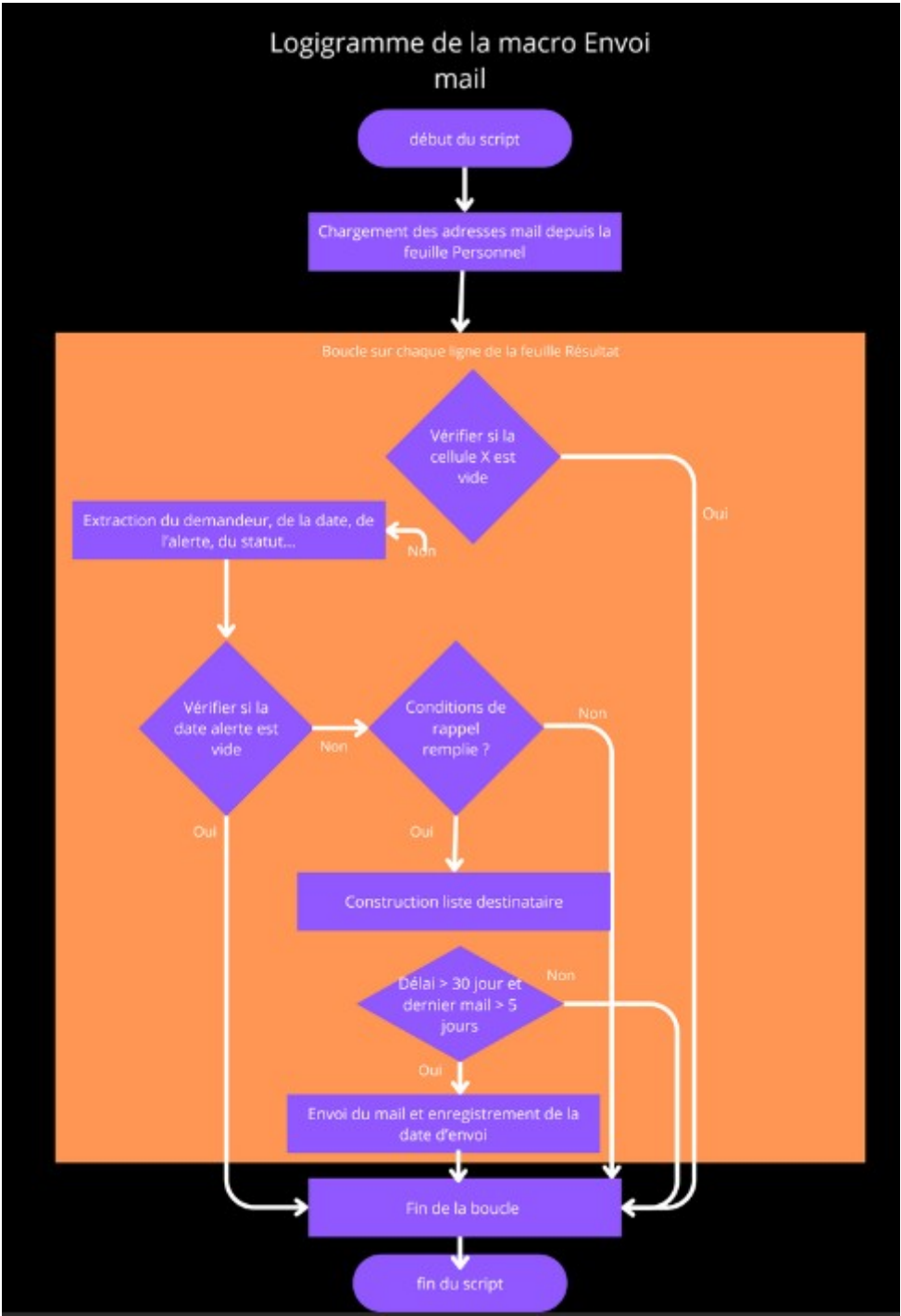
Macro d'actualisation :



Macros Créer résultat PowerQuery :

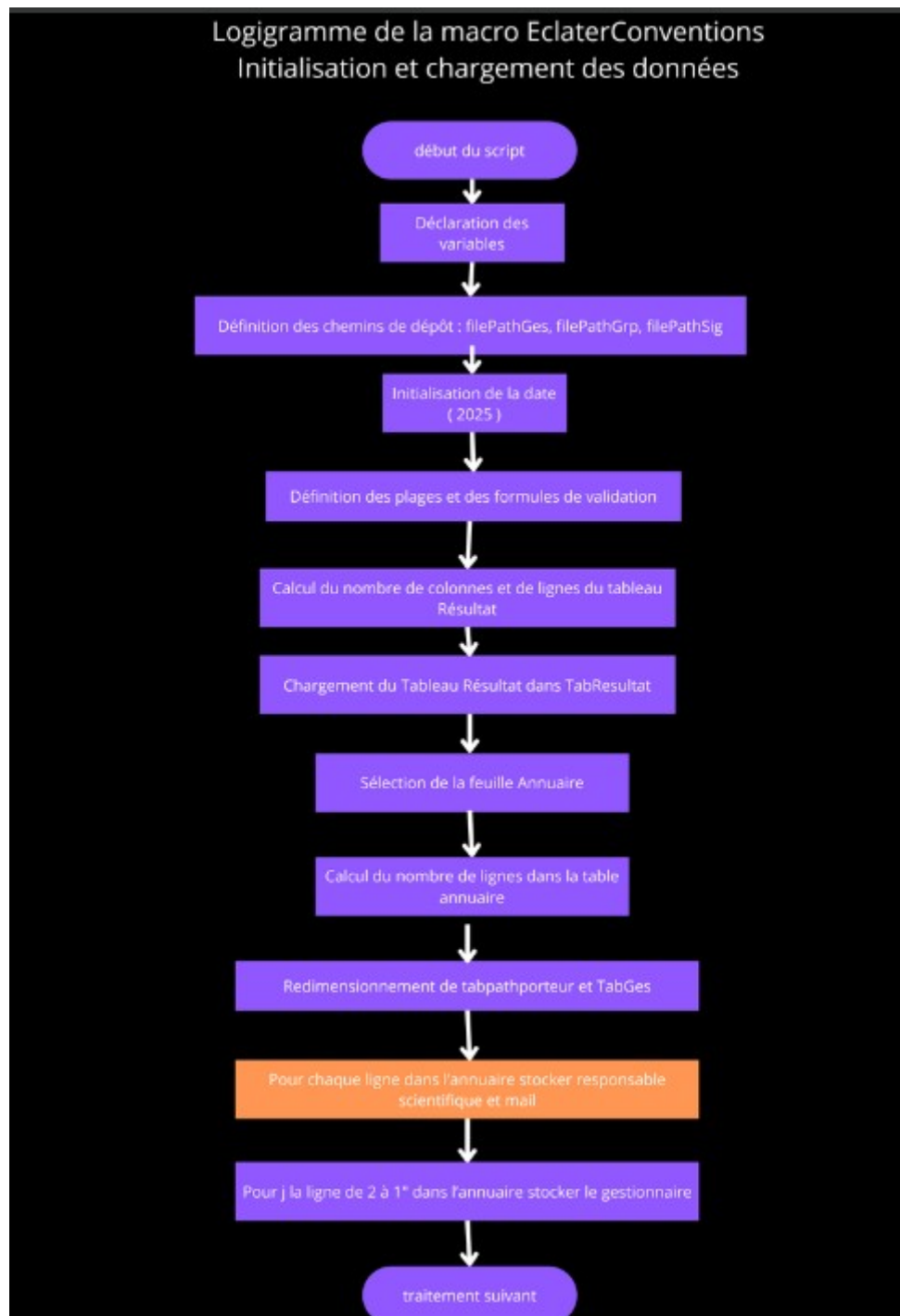


Macros envoie mail :

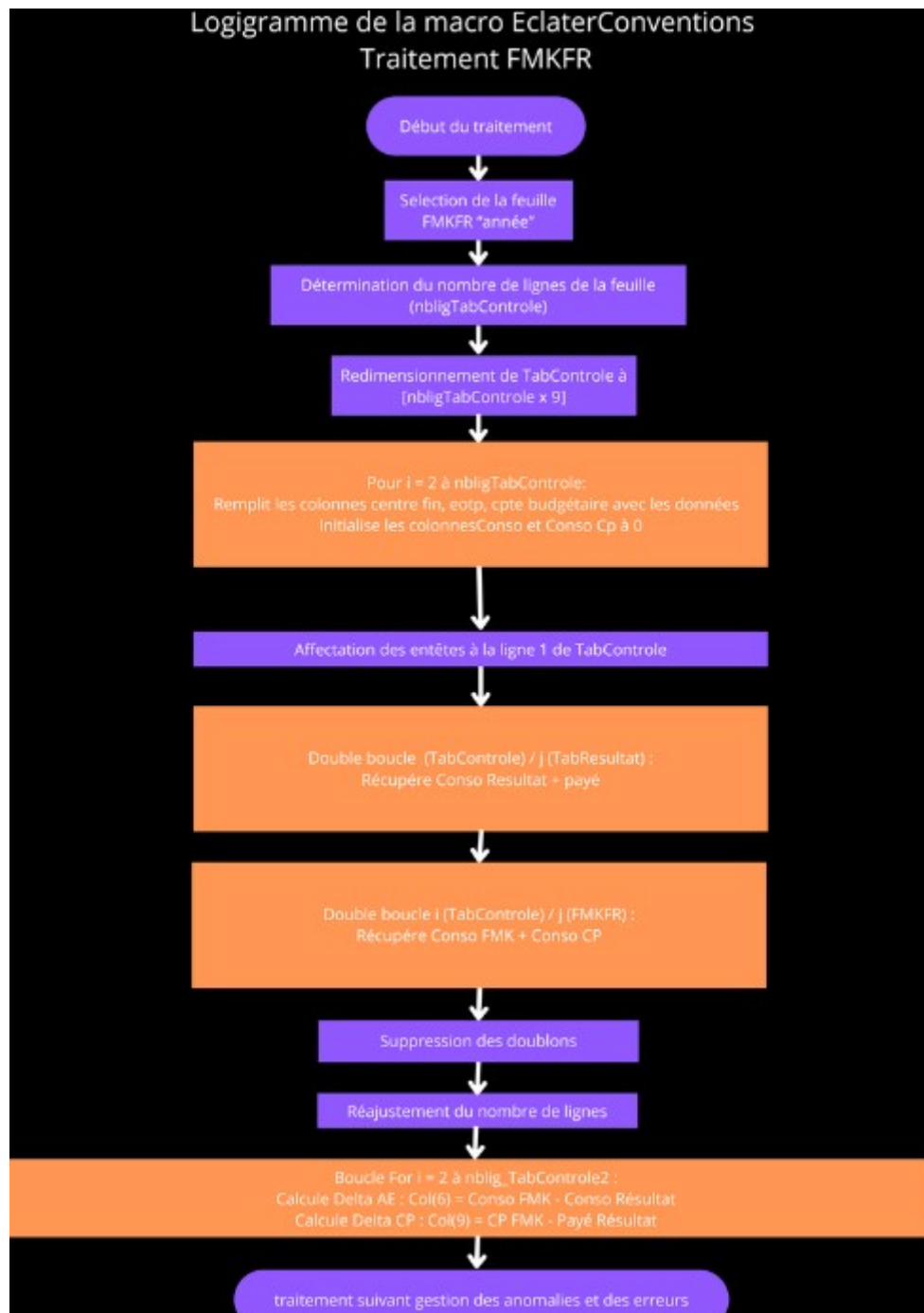


La macro éclater convention c'est faite en plusieurs parties car elle contient de nombreux processus.

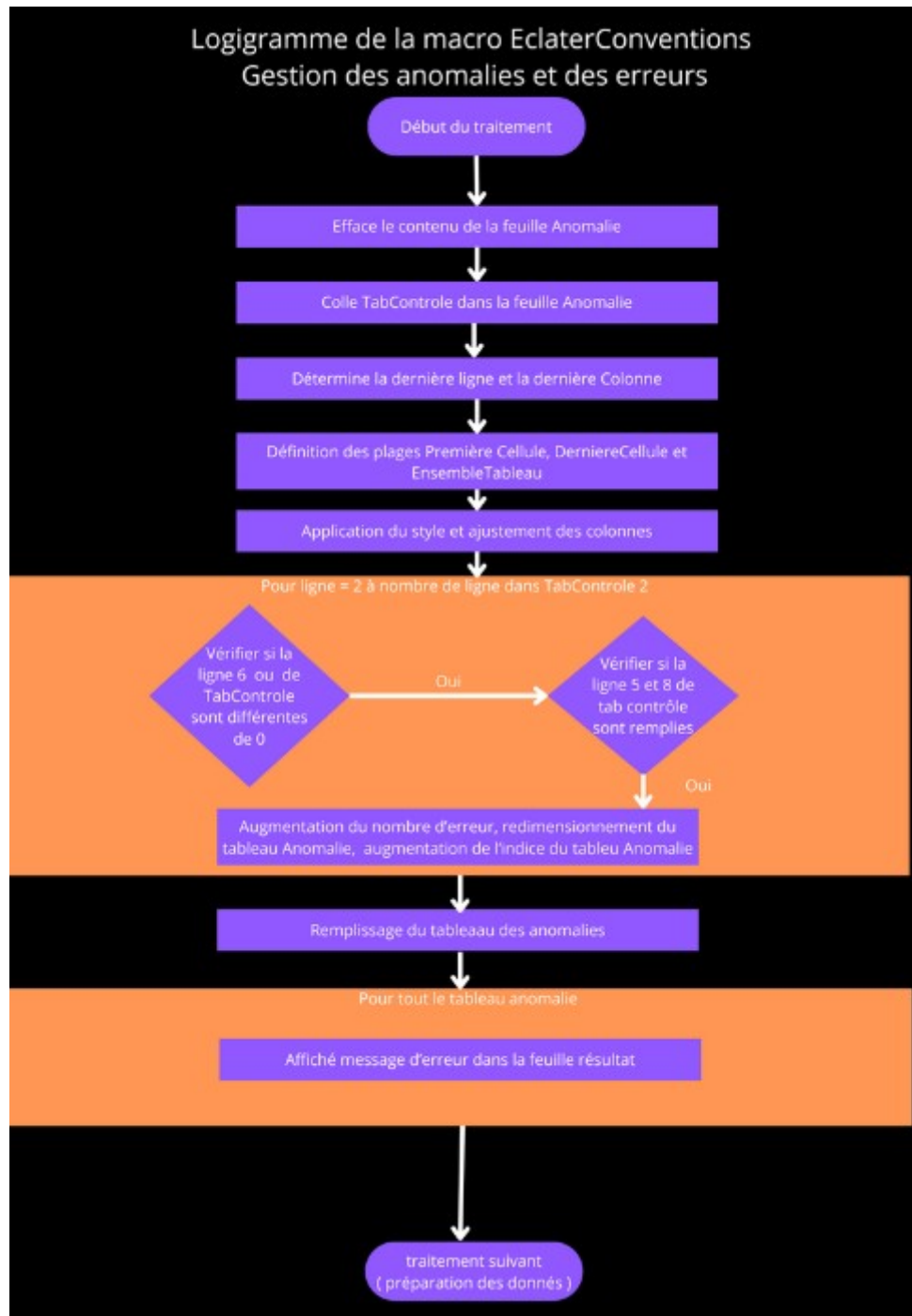
1. Chargement des données :



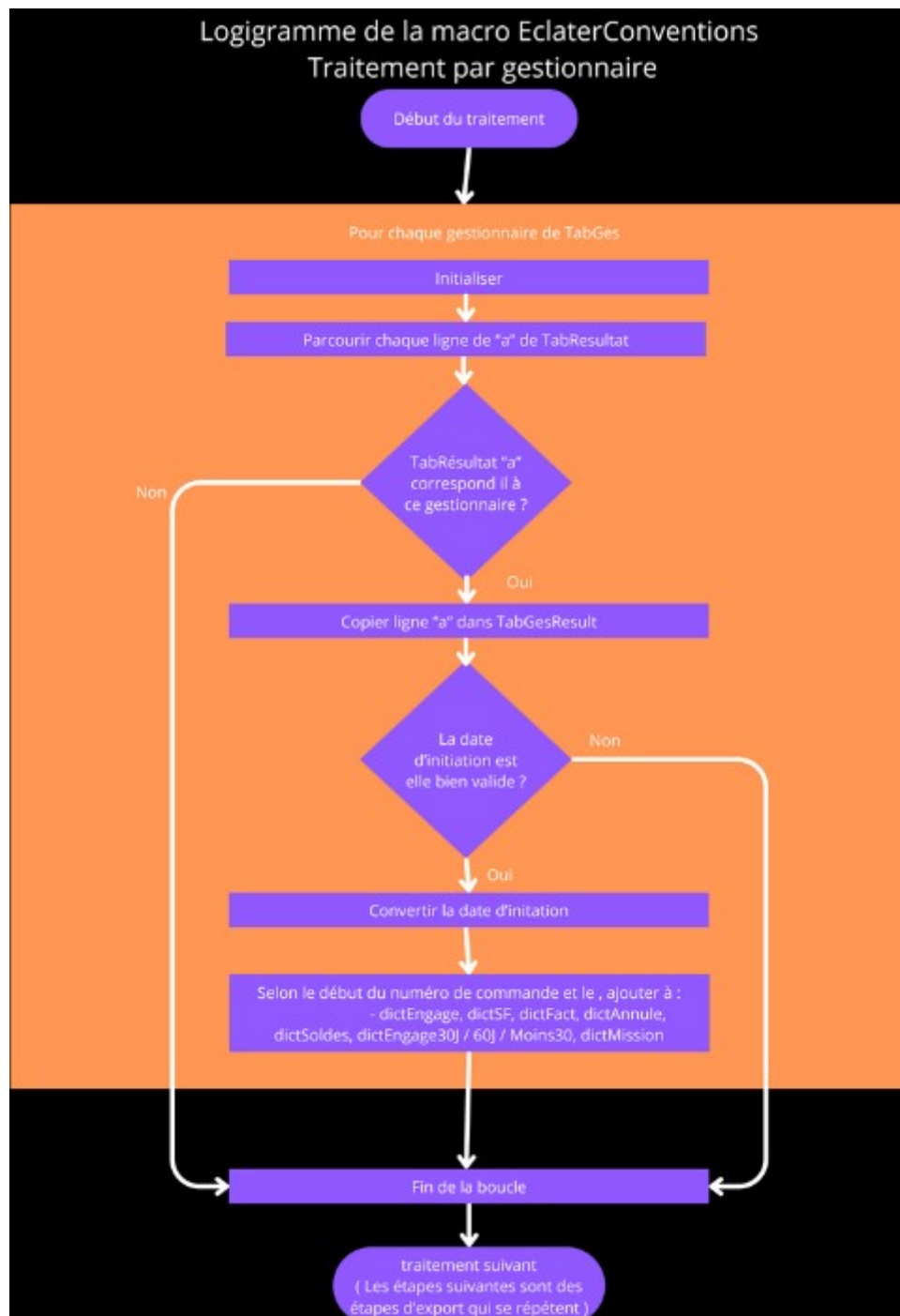
2. Traitement FMKFR



3. Gestion des anomalies et des erreurs :



4. Traitement par gestionnaire.



Jour 5 (vendredi 6 juin):

Annotation directe du code du stagiaire précédent. Sa gestion était sur les commandes de l'inp les miennes seront sur celles du CNRS. Il suffit donc de changer la valeur de chaque cellule pour que cela correspondent bien à l'organisation du CNRS.

Avec le code annoté et les logigrammes la mise en place de l'application sera plus simple (normalement).

Début de réécriture du code CréerResultatPowerQuery avec les données spécifiques au CNRS.

Compte rendu de stage semaine 3 :

Rappel du projet : Création d'une application de comptabilité avec visual basic, le code pour la partie INP est déjà fait ma mission consiste à l'adapter pour le coté CNRS.

Jour 1 (Mardi 10 juin):

Accès au classeur excel du CNRS. Début d'adaptation de la macro CréerRésultat Powerquery, cette macro mets à jour le classeur excel avec les données les plus récentes et crée une feuille qu'on appellera Resultat Query. Cette macro avait déjà était développé pour le coté INP l'adaptation a donc était rapide.

Jour 2 (mercredi 11 juin):

Fin du code VBA pour la macro CréerRésultatPowerquery ? Début de la macro EclaterConvention. Cette macro consiste à créer un nouveau classeur excel pour chaque gestionnaire, groupe, et porteur. Ce classeur excel va réunir toutes les commandes fait par cette personne ainsi qu'un tableau de bord récapitulatif (avec les commandes, soldés, en attente, le total...).

Le développement a commencé par l'éclatement des gestionnaire.

Jour 3 (jeudi 12 juin):

Fin du travail sur l'éclatement des gestionnaire mais sans tableau de bord pour l'instant. Encore une fois cette macro existe déjà pour l'inp. Le travail principal a était une adaptation.

Jour 4 (vendredi 13 juin):

Création du tableau de bord pour les gestionnaire. Il manque à présent l'éclatement par groupe et par porteur.

Compte rendu de stage semaine 4 :

Rappel du projet : Création d'une application de comptabilité avec visual basic, le code pour la partie INP est déjà fait ma mission consiste à l'adapter pour le coté CNRS.

Jour 1 (Lundi 16 juin):

Quelques changements sur la macro de création de tableau de bord des gestionnaires. Développement de l'éclatement par groupe et par porteur. Fin de l'éclatement par groupe mais pas par porteur.

Jour 2 (Mardi 17 juin):

Fin de l'éclatement par porteur. Ensuite j'ai fait des tests de dépôts dans des dossiers cibles. Les test fonctionnent.

Jour 3 (Mercredi 18 juin):

Début de l'intégration des macros à l'application finale. Une fois intégrés au classeur excel de comptabilité les macros du CNRS et de l'INP fonctionneront en même temps. Les chemins de dépôts de ma macro d'éclatement de gestionnaires avaient quelques problèmes dans l'application finale. Ces problèmes ont été corrigés avec l'aide de ma tutrice de stage.

Jour 4 (Jeudi 19 juin):

Modification de l'éclatement des gestionnaires pour qu'il ne supprime pas les anciens fichiers du même nom mais qu'il les ouvre et ajoute deux nouvelles feuilles. De cette manière les fichiers éclatés auront les informations de l'INP et du CNRS en même temps. Les chemins de dépôts pour l'éclatement des groupes et des porteurs avaient également quelques problèmes. Début de leur correction.

Jour 5 (Vendredi 20 juin):

Fin de la correction des chemins pour l'éclatement des groupes et des porteurs. Ajout d'un éclatement par SIG.

Compte rendu de stage semaine 5 :

Rappel du projet : Création d'une application de comptabilité avec visual basic, le code pour la partie INP est déjà fait ma mission consiste à l'adapter pour le coté CNRS.

Jour 1 (Lundi 23 juin) :

Test de l'éclatement des fichiers pour les porteurs, les Sig et les groupes du coté CNRS de l'application. L'éclatement des porteurs avait quelques problèmes (le chemin de dépôt n'était pas bien défini). Le problème a été corrigé.

Jour 2 (Mardi 24 juin) :

Ajout de plages de validations dans les dossiers éclatés par gestionnaires. Permet à chaque gestionnaire de changer le gestionnaire d'une commande. Modification du code de l'ancien stagiaire pour l'adapter aux nouvelles exigences (crée un seul fichier éclaté avec deux feuilles une pour le CNRS et une pour l'INP.)

Jour 3 (Mercredi 25 juin) :

Test d'éclatement pour le CNRS et pour l'INP pour les gestionnaires, les groupes, les SIG et les porteurs. Aucun problème à signaler sur les gestionnaires, groupes et SIG. Quelques soucis d'enregistrement du côté CNRS, finalement corrigé.

Jour 4 (Jeudi 26 juin) :

Quelques arrangements de visibilité du coté gestionnaire. Caché les feuilles annuaires et personnel pour plus de lisibilité. + effacement des anciennes feuilles annuaire et personnel à chaque fois que le code se fait.

Jour 5 (Vendredi 27 juin) :

Ajout de bouton sur le classeur excel pour faciliter le lancement des macros, plus ajout de commentaire dans le code pour le rendre plus simple à adapter et comprendre.

Les boutons sont aussi là pour faciliter l'utilisation pour des gens qui ne connaissent pas bien l'utilité de chaque macro.

Création d'une macro qui ne fait que l'éclatement des gestionnaires sans les autres éléments Voici l'exemple du code d'éclatement des gestionnaires (l'éclatement des autres fichiers suis une structure similaire) :

' Cette macro ne fait que l'éclatement des gestionnaires pour un traitement plus rapide

```
Sub EclaterGestionnaireCNRS()
```

```
    ' Déclaration des variables pour stocker les données et manipuler les
```

```

feuilles Excel
Dim TabGeneral, TabNouveau, TabResultat, TabGesResultUnique,
TabGesResultUniqueSansDoublons, TabAnnuaire, TabControle As Variant
Dim TabSansDoublons, dataSource, PlagesValidation, FormulesValidation,
tableauAnomalie, TabGesResult, Values As Variant
Dim nblig_tabGeneral, nbcol_tabGeneral, nblig_TabAnomalie, DerniereColonne,
DerniereLigne, nbligIncrement, i, j As Long
Dim PremiereCellule, EnsembleTableau, DerniereCellule As Range
Dim tolerance As Double
Dim nbErreur, ligne As Integer
Dim message As String
Dim filePathGes As String, datee As Integer

' Chemins des fichiers pour les gestionnaires
' filePathGes = "Y:\gestion_financiere\Documentation\Compta\INPT\compta 2005\
Gestionnaires\Situation SAIC "
filePathGes = "C:\Users\stagiaire\Desktop\Stagiaire-Adrien- M-C\
TestdépôtGestionnaires\Situation SAIC 2025"

datee = 2025 ' Année dans laquelle le traitement est effectué

' Plages et formules de validation
PlagesValidation = Array("D3:D", "E3:E", "F3:F", "G3:G", "I3:I", "AV3:AV")
FormulesValidation = Array("=Annuaire!$AD$2:$AD$13", "=Annuaire!
$AC$2:$AC$15", "=Annuaire!$AE$2:$AE$85", _
    "=Annuaire!$AH$2:$AH$138", "=Annuaire!$AG$2:$AG$14", "=Annuaire!
$AL$2:$AL$5")

' Récupération du tableau des résultats
nbcol_TabResultat = ThisWorkbook.Sheets("Resultat CNRS").Cells(2,
ThisWorkbook.Sheets("Resultat CNRS").Columns.Count).End(xlToLeft).Column
nblig_TabResultat = (ThisWorkbook.Sheets("Resultat
CNRS").Cells(ThisWorkbook.Sheets("Resultat CNRS").Rows.Count, 2).End(xlUp).Row)
- 1
TabResultat = ThisWorkbook.Sheets("Resultat
CNRS").Range(ThisWorkbook.Sheets("Resultat CNRS").Cells(1, 1),
ThisWorkbook.Sheets("Resultat CNRS").Cells(nblig_TabResultat,
nbcol_TabResultat)).Value

Sheets("Annuaire").Select

' Calcul du nombre de lignes et récupération des données pour les porteurs
et gestionnaires
nblig_tabannuaire = Cells(Rows.Count, 6).End(xlUp).Row
ReDim TabGes(15, 1)

For j = 2 To 15
    TabGes(j, 1) = RTrim(Cells(j, "AC").Value)
Next j

nblig_TabGes = UBound(TabGes, 1)
ReDim TabGesResult(1 To nblig_TabResultat, 1 To nbcol_TabResultat)

' Eclatement des Gestionnaires
For i = 1 To nbcol_TabResultat
    TabGesResult(1, i) = TabResultat(2, i)
Next i
For i = 2 To nblig_TabGes
    nbligIncrement = 1
    If TabGes(i, 1) = "Denis Bourrel" Then
        nom = "Mutualisé"
    Else

```

```

        nom = TabGes(i, 1)
End If
nom = Replace(nom, "é", "e")
nom = Replace(nom, "ç", "c")
nom = Replace(nom, "è", "e")
Dim fichierComplet As String
'fichierComplet = filePathGes & datee & "\" & nom & "_" & datee &
".xlsx"
fichierComplet = filePathGes & "\" & nom & "_" & datee & ".xlsx"
Dim classeurExiste As Boolean
classeurExiste = Dir(fichierComplet) <> ""
Application.AskToUpdateLinks = False
Dim Classeur As Workbook

' Création ou ouverture du fichier Excel correspondant au gestionnaire
If classeurExiste Then
    Set Classeur = Workbooks.Open(fichierComplet, UpdateLinks:=0)
Else
    Set Classeur = Workbooks.Add
End If
Application.DisplayAlerts = False ' Désactiver les alertes
Dim Feuille As Worksheet
Dim tableauDeBord As Worksheet

' Suppression et recréation des feuilles nécessaires
On Error Resume Next
Classeur.Sheets("Resultat_CNRS").Delete
On Error GoTo 0
Set Feuille =
Classeur.Sheets.Add(After:=Classeur.Sheets(Classeur.Sheets.Count))
Feuille.Name = "Resultat_CNRS"

On Error Resume Next
Classeur.Sheets("Tableau_de_bord_CNRS").Delete
On Error GoTo 0
Set tableauDeBord =
Classeur.Sheets.Add(After:=Classeur.Sheets(Classeur.Sheets.Count))
tableauDeBord.Name = "Tableau_de_bord_CNRS"
On Error Resume Next
Classeur.Sheets("Feuil1").Delete
On Error GoTo 0

' Suppression des anciennes feuilles Annuaire et Personnel si elles
existent
On Error Resume Next
Classeur.Sheets("Annuaire").Delete
Classeur.Sheets("Personnel").Delete
On Error GoTo 0

On Error Resume Next
Classeur.Sheets("Annuaire CNRS").Delete
Classeur.Sheets("Personnel CNRS").Delete
On Error GoTo 0

' Copier et renommer les feuilles Annuaire et Personnel
ThisWorkbook.Sheets("Annuaire").Copy After:=Classeur.Sheets(1)
ActiveSheet.Name = "Annuaire CNRS"

ThisWorkbook.Sheets("Personnel").Copy After:=Classeur.Sheets(1)
ActiveSheet.Name = "Personnel CNRS"

' Cacher les feuilles Annuaire CNRS et Personnel CNRS.
Classeur.Sheets("Annuaire CNRS").Visible = xlSheetHidden
Classeur.Sheets("Personnel CNRS").Visible = xlSheetHidden

```

```

' Création des dictionnaires pour stocker les informations des commandes
Set dictEngage = CreateObject("Scripting.Dictionary")
Set dictSF = CreateObject("Scripting.Dictionary")
Set dictFact = CreateObject("Scripting.Dictionary")
Set dictSoldes = CreateObject("Scripting.Dictionary")
Set dictAnnule = CreateObject("Scripting.Dictionary")
Set dictEngage30J = CreateObject("Scripting.Dictionary")
Set dictEngage60J = CreateObject("Scripting.Dictionary")
Set dictEngageMoins30J = CreateObject("Scripting.Dictionary")

' Parcourir les données pour extraire les informations pertinentes
For a = 2 To nblig_TabResultat
    Numero_de_flux = TabResultat(a, 1)
    Rubrique_de_la_piece = TabResultat(a, 32)
    CdeN_1 = TabResultat(a, 36)
    Cde = TabResultat(a, 37)
    Fact = TabResultat(a, 45)
    Paye = TabResultat(a, 47)
    Date_certification = TabResultat(a, 41)
    dateInit = TabResultat(a, 31)
    dateInit = Replace(dateInit, ".", "/")
    Reste_a_constater = TabResultat(a, 42)
    statut = TabResultat(a, 48)

    ' Filtrage et traitement des commandes
    If ((TabResultat(a, 5) = TabGes(i, 1) And TabGes(i, 1) <> "Denis
Bourrel") _
    Or ((TabResultat(a, 5) = ErrNA Or TabResultat(a, 5) = "Denis
Bourrel" Or TabResultat(a, 5) = "Mutualisé") _
    And (TabGes(i, 1) = "Mutualisé" Or TabGes(i, 1) = "Denis Bourrel"))))
Then
    nbligIncrement = nbligIncrement + 1
    For j = 1 To nbcol_TabResultat
        TabGesResult(nbligIncrement, j) = TabResultat(a, j)
    Next j
    If IsDate(dateInit) Then
        dateInit = CDate(dateInit)
        If Rubrique_de_la_piece = "COMMANDE" Then
            valeur = Numero_de_flux

            ' Traitement des commandes engagées
            If (CdeN_1 + Cde) > 0 And Paye = 0 And Reste_a_constater
> 0 And Fact = 0 And statut = "En cours" Then
                If Not dictEngage.exists(valeur) Then dictEngage.Add
valeur, True

                If Date < dateInit + 30 Then
                    If Not dictEngageMoins30J.exists(valeur) Then
dictEngageMoins30J.Add valeur, True
                End If
                If Date >= dateInit + 30 And Date < dateInit + 60
Then
                    If Not dictEngage30J.exists(valeur) Then
dictEngage30J.Add valeur, True
                End If
                If Date > dateInit + 60 Then
                    If Not dictEngage60J.exists(valeur) Then
dictEngage60J.Add valeur, True
                End If
            End If

            ' Traitement des commandes SF, facturées, annulées et
soldées
            If Date_certification <> "" And (CdeN_1 + Cde) > 0 And

```

```

Paye = 0 And Fact = 0 And Reste_a_constater = 0 Then
    If Not dictSF.exists(valeur) Then dictSF.Add valeur,
True
    End If
    If (CdeN_1 + Cde) > 0 And Paye = 0 And Reste_a_constater
= 0 And Fact > 0 Then
        If Not dictFact.exists(valeur) Then dictFact.Add
valeur, True
        End If
        If statut = "Annulé" Then
            If Not dictAnnule.exists(valeur) Then dictAnnule.Add
valeur, True
        End If
        If statut = "Soldé" Then
            If Not dictSoldes.exists(valeur) Then dictSoldes.Add
valeur, True
        End If
    End If
End If
End If
Next a

' Création du tableau de bord pour chaque gestionnaire
explications = Array("Nombre de commandes détaillées", "SF = Service
fait (les commandes qui ont été livrées)", "Facture = Les commandes qui ont été
facturées", "Soldés = Les commandes qui ont été payés")
tableauDeBord.Range("A1:A4").Value = Application.Transpose(explications)

headers = Array("Cde Engagé", "SF", "Facture", "NB Soldé", "NB Annulé",
"Total")
tableauDeBord.Range("B6:G6").Value = headers
tableauDeBord.Range("A7:A10").Value = Application.Transpose(Array("Nb
Cde Total", "Moins de 30J", "Plus de 30J", "Plus de 60J"))

ReDim Values(3, 5)
Values(0, 0) = dictEngage.Count
Values(0, 1) = dictSF.Count
Values(0, 2) = dictFact.Count
Values(0, 3) = dictSoldes.Count
Values(0, 4) = dictAnnule.Count
Values(0, 5) = ""
Values(1, 0) = dictEngageMoins30J.Count
Values(1, 1) = ""
Values(1, 2) = ""
Values(1, 3) = ""
Values(1, 4) = ""
Values(1, 5) = ""
Values(2, 0) = dictEngage30J.Count
Values(2, 1) = ""
Values(2, 2) = ""
Values(2, 3) = ""
Values(2, 4) = ""
Values(2, 5) = ""
Values(3, 0) = dictEngage60J.Count
Values(3, 1) = ""
Values(3, 2) = ""
Values(3, 3) = ""
Values(3, 4) = ""
Values(3, 5) = ""

Set tableauRange = tableauDeBord.Range("B7:G10")
tableauRange.Value = Values
tableauDeBord.Range("G7").FormulaLocal = "=SOMME(B7:F7)"
tableauDeBord.ListObjects.Add(xlSrcRange, tableauDeBord.Range("B6:G10"),

```

```

xlYes).TableStyle = "TableStyleMedium2"

Set Feuille = Classeur.Sheets("Resultat_CNRS")
'Connaître les anomalies
TabGesResultSansDoublons = SupDoublonsArray(TabGesResult, 28, 30, 31)
message = "Anomalie dans le(s) eotp suivants : "
For k = 1 To nblig_TabAnomalie
    For j = 2 To UBound(TabGesResultSansDoublons)
        If tableauAnomalie(k, 1) = TabGesResultSansDoublons(j, 28) And
tableauAnomalie(k, 2) = TabGesResultSansDoublons(j, 30) And tableauAnomalie(k,
3) = TabGesResultSansDoublons(j, 31) Then
            message = message & tableauAnomalie(k, 2) + " | "
        End If
    Next j
Next k

'Affichage des données
Feuille.Range("A2").Resize(nbligIncrement, nbcol_TabResultat).Value =
TabGesResult
If message = "Anomalie dans le(s) eotp suivants : " Then
    message = ""
End If

Feuille.Range("D1") = message
Feuille.Range("A1") = "Mise à jour le : " & CStr(Now)
Feuille.Range("A1").Font.Bold = True
DerniereLigne = Feuille.Cells(Feuille.Rows.Count, "A").End(xlUp).Row
DerniereColonne = Feuille.Cells(2,
Feuille.Columns.Count).End(xlToLeft).Column
Set PremiereCellule = Feuille.Cells(2, 1)
Set DerniereCellule = Feuille.Cells(DerniereLigne, DerniereColonne)
Set EnsembleTableau = Feuille.Range(PremiereCellule, DerniereCellule)
Dim tbl As ListObject
For Each tbl In Feuille.ListObjects
    tbl.Delete
Next tbl

'Définition du style de la feuille Résultat_CNRS ( TableStyleMedium2 =
bleu )

Feuille.ListObjects.Add(xlSrcRange, EnsembleTableau, , xlYes).TableStyle
= "TableStyleMedium2"

Feuille.Cells(DerniereLigne + 1, 36).FormulaR1C1 = "=SUBTOTAL(9, R2C:R[-
1]C)"
' Cde N-1€
Feuille.Cells(DerniereLigne + 1, 37).FormulaR1C1 = "=SUBTOTAL(9, R2C:R[-
1]C)"
' CDE €
Feuille.Cells(DerniereLigne + 1, 38).FormulaR1C1 = "=SUBTOTAL(9, R2C:R[-
1]C)"
' Réservé ligne €
Feuille.Cells(DerniereLigne + 1, 45).FormulaR1C1 = "=SUBTOTAL(9, R2C:R[-
1]C)"
' Fact €
Feuille.Cells(DerniereLigne + 1, 47).FormulaR1C1 = "=SUBTOTAL(9, R2C:R[-
1]C)"
' Payé
Feuille.Name = "Resultat_CNRS"

Set plageTableau = Feuille.Range("A3:AV" & DerniereLigne)

' Mise en forme des commandes en cours depuis 30 et 60 jours
Set plageTableau = Feuille.Range("A3:AV" & DerniereLigne)

'Rouge ( en cours depuis plus de 60 jours )
With plageTableau.FormatConditions.Add(Type:=xlExpression,
Formula1:= _
    "=ET($AF3=""Commande""; $AV3=""En cours""; ($AJ3+$AK3)>0;

```



```

$AS3=0; $AP3>0; $AU3=0; AUJOURDHUI()-$AE3>=60)")
    .Interior.Color = RGB(225, 90, 80)
End With

    ' Orange ( en cours depuis plus de 30 jours )
    With plageTableau.FormatConditions.Add(Type:=xlExpression,
Formula1:= _
    "=ET($AF3=""Commande""; $AV3=""En cours""; ($AJ3+$AK3)>0;
$AS3=0; $AP3>0; $AU3=0; AUJOURDHUI()-$AE3>30; AUJOURDHUI()-$AE3<60)")
    .Interior.Color = RGB(255, 165, 0)
End With

    ' Recréer les plages de validation
    PlagesValidation = Array("D3:D", "E3:E", "F3:F", "G3:G", "I3:I",
"AV3:AV")

    ' Donner la bonne source au plages de validation ( Ici annuaire CNRS )
    FormulesValidation = Array("='Annuaire CNRS'!$AD$2:$AD$13",
"='Annuaire CNRS'!$AC$2:$AC$15", "='Annuaire CNRS'!$AE$2:$AE$85", _
"='Annuaire CNRS'!$AH$2:$AH$138", "='Annuaire CNRS'!$AG$2:$AG$14",
"='Annuaire CNRS'!$AL$2:$AL$5")

    ' Appliquer les règles de validation dans la Feuille
    For idx = LBound(PlagesValidation) To UBound(PlagesValidation)
        With Feuille.Range(PlagesValidation(idx) & DerniereLigne).Validation
            .Delete
            .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, _
                Formula1:=FormulesValidation(idx)
            .ErrorTitle = "saisi non conforme"
            .InputMessage = ""
            .ErrorMessage = "écrit mieux"
            .IgnoreBlank = True
            .InCellDropdown = True
            .ShowInput = True
            .ShowError = True
        End With
    Next idx

    'Formatage des colonnes
    Feuille.Columns.AutoFit
    Feuille.Columns("J:W").Hidden = True
    Feuille.Columns("A").ColumnWidth = 31
    Feuille.Columns("B").ColumnWidth = 20
    Feuille.Columns("C").ColumnWidth = 9
    Feuille.Columns("D").ColumnWidth = 25
    Feuille.Columns("F").ColumnWidth = 35
    Feuille.Columns("Z").ColumnWidth = 12
    Feuille.Columns("AA").ColumnWidth = 40
    Feuille.Columns("AD").ColumnWidth = 20
    Feuille.Columns("AE").ColumnWidth = 30
    Feuille.Columns("AG").ColumnWidth = 40
    Feuille.Columns("AH").ColumnWidth = 45
    Feuille.Columns("AR").ColumnWidth = 15

    Feuille.Range("AJ3:AJ" & DerniereLigne).NumberFormat = "#,##0.00"
    Feuille.Range("AK3:AK" & DerniereLigne).NumberFormat = "#,##0.00"
    Feuille.Range("AL3:AL" & DerniereLigne).NumberFormat = "#,##0.00"
    Feuille.Range("AS3:AS" & DerniereLigne).NumberFormat = "#,##0.00"
    Feuille.Range("AU3:AU" & DerniereLigne).NumberFormat = "#,##0.00"

    Application.DisplayAlerts = True ' Réactiver les alertes
    Application.DisplayAlerts = False
    Classeur.SaveAs Filename:=fichierComplet, FileFormat:=xlOpenXMLWorkbook
    Application.DisplayAlerts = True

```

Classeur.Close SaveChanges:=False

```

' Réinitialisation sans supprimer la première ligne
For y = 2 To UBound(TabGesResult, 1) ' Commencer à y = 2 pour éviter la
première ligne
    For Z = LBound(TabGesResult, 2) To UBound(TabGesResult, 2)
        TabGesResult(y, Z) = Empty
    Next Z
Next y
Next i
End Sub

```

Cette macro crée un fichier récapitulatif pour chaque gestionnaire avec deux feuilles :

Une feuille résultat :

Mise à jour le : 26/06/2025 11:21:15									
Numero de flux	N° commande BFC	Nom	Groupe	Gestionnaire	Porteur	Matière	Auteur	SIG	Code
0266L020324_45213	2011125854	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	3110 Denis BOURREL			DB11
0266L020618_45741	2011604332	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			RA15
0266L020324_45214	2011125854	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	3110 Denis BOURREL			DC12
0266L020618_45746	2011604332	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			RA15
0266L020618_45736	2011604332	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			NB51
0266L020618_45743	2011604332	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			RA15
0266L020618_45747	2011604332	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			RA15
0266L020618_45740	2011604332	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			RA15
0266L020618_45739	2011604332	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			RA15
0266L020618_45738	2011604332	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			RA15
0266L020390_45123	2011228532	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			IE02
0266L020618_45742	2011604332	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			RA15
0266L020390_45124	2011228532	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			IE02
0266L020618_45745	2011604332	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			RA15
0266L020390_45125	2011228532	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			IE02
0266L020418_45193	2011298672	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1160 Karine DEFRETIN			EC14
0266L020665_45913	2011603044	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1150 Karine DEFRETIN			FC01
0266L020665_45912	2011603044	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1150 Karine DEFRETIN			FC01
0266L020512_45492	2011387447	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			IG02
0266L020512_45493	2011387447	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			IG02
0266L020665_45914	2011603044	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1150 Karine DEFRETIN			FC01
0266L020512_45495	2011387447	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			IG02
0266L020665_45915	2011603044	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1150 Karine DEFRETIN			FC01
0266L020512_45496	2011387447	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			IG02
0266L020512_45497	2011387447	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			IG02
0266L020513_45498	2011387450	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1190 Karine DEFRETIN			II01
0266L020513_45499	2011387450	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1190 Karine DEFRETIN			II01
0266L020547_45571	2011441042	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	2120 Karine DEFRETIN			IA24
0266L020548_45572	2011441037	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	2110 Karine DEFRETIN			OA11
0266L020617_45730	2011604331	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			AB01
0266L020617_45731	2011604331	DR14	Affaires Générales	Denis Bourrel	Responsable Affaires Générales	1170 Karine DEFRETIN			AB01

Et une feuille tableau de bord :

Nombre de commandes détaillées						
SF = Service fait (les commandes qui ont été livrées)						
Facture = Les commandes qui ont été facturées						
Soldés = Les commandes qui ont été payés						
	Cde Engagé	SF	Facture	NB Soldé	NB Annulé	Total
Nb Cde Total	20	0	31	13	0	64
Moins de 30J	0					
Plus de 30J	0					
Plus de 60J	20					

Le fichier crée est ensuite nommé en fonction du nom du gestionnaire et de l'année de traitement. Le code du coté INP va ouvrir ce fichier et lui ajouter deux feuilles similaires, mais pour l'INP permettant un suivi des commandes. Les colonnes Groupe, gestionnaires, porteur, matière et SIG contiennent des plages de validations qui vont permettre à chaque gestionnaire de changer les paramètres d'une commande.

Maintenant voilà à quoi ressemble la page avec les boutons pour lancer les macros. L'objectif est de faire une page compréhensible pour chaque gestionnaire, sans avoir à lancé chaque macro individuellement :

Les traitements de cette page se font uniquement pour le CNRS

Permet de mettre à jour la feuille résultat CNRS Avec les informations les plus récentes

Résultat query CNRS

Permet de faire un traitement complet (mise à jour et éclatement) des commandes du CNRS (Pour les gestionnaires, les Groupes, les porteurs, les SIG)

Eclater les fichiers CNRS

Permet de faire un traitement (mise à jour et éclatement) pour les gestionnaires du CNRS

Eclater les gestionnaires CNRS