```
Sub CheckCourseIntersections()
   Dim wb1 As Workbook, wb2 As Workbook
   Dim ws1 As Worksheet, ws2 As Worksheet
   Dim rng1 As Range, filterRange As Range
   Dim cell1 As Range
   Dim nrc1 As String, nrc2 As String
   Dim filteredNRC As Object
   Dim tipo1 As String, tipo2 As String
   Dim dayHeaders As Variant
   Dim timeRange1 As String, timeRange2 As String
   Dim start1 As Date, end1 As Date, start2 As Date, end2 As Date
   Dim dia1 As String, dia2 As String
   Dim intersectionList As Collection
   Dim filePath As String
   Dim result As String
   Dim colNRC As Integer, colTipo As Integer, colDia As Integer
   Dim i As Integer

   ' Initialize constants
   dayHeaders = Array("LUNES", "MARTES", "MIERCOLES", "JUEVES", "VIERNES") ' days of the week headers in the excel with the course data

   ' Set workbook and worksheet for the first Excel
   Set wb1 = ThisWorkbook
   Set ws1 = wb1.Sheets(1)
   Set rng1 = ws1.Range("A2:A" & ws1.Cells(ws1.Rows.Count, "A").End(xlUp).Row) ' Range excluding header (the header starts at A1)
   ' Find column indices dynamically in the first workbook
   Dim colNRC1 As Integer, colNRC2 As Integer, colTope As Integer
   colNRC1 = FindColumn(ws1, "NRC")    ' Change this if the column containing the NRC changes the header name
   colNRC2 = FindColumn(ws1, "TOPE DE HORARIO CURSO 2")    ' Change this if the column containing the NRC of the second course changes header name
   colTope = FindColumn(ws1, "TOPES") ' This is the output column where it will add the data like CLAS-CLAS, this column must exist
   colCantidadTopes = FindColumn(ws1, "CANTIDAD TOPES") ' This is the output column where there will be the ammount of intersections
   ' Open the second Excel
   filePath = Application.GetOpenFilename("Excel Files (*.xls; *.xlsx), *.xls; *.xlsx", , "Select the Second Workbook")
   If filePath = "False" Then Exit Sub ' Exit if no file is selected
   Set wb2 = Workbooks.Open(filePath)
   Set ws2 = wb2.Sheets(1) ' Adjust sheet index or name as needed
   Set filterRange = ws2.Rows(14) ' Assume headers are in row 14

   ' Find column indices dynamically in the second workbook
   colNRC = FindColumn(ws2, "NRC", 14) ' The column that contains the NRC in the horario excel
   colTipo = FindColumn(ws2, "TIPO DE REUNIÓN", 14) ' The columna with the type of the course (CLAS, AYUD, LABT, etc
   colDia = FindColumn(ws2, "INICIO", 14) ' start date for the row, like the day a test happens or classes start

   Dim verification As Integer
   ' Loop through NRC pairs in the first workbook
```

```
For Each cell1 In rng1
    ws2.Rows.Hidden = False ' Unhide all rows
    nrc1 = ws1.Cells(cell1.Row, colNRC1).Value
    nrc2 = ws1.Cells(cell1.Row, colNRC2).Value
    If nrc1 <> "" And nrc2 <> "" Then

        ' Get the filtered rows from the second workbook
        Set filteredNRC = RegularFilterNRC(ws2, nrc1, nrc2, colNRC, filterRange)

        ' Check time and date intersections
        Set intersectionList = New Collection

        verification = 0
        For Each nrcKey1 In filteredNRC.Keys
            For Each nrcKey2 In filteredNRC.Keys
                If nrcKey1 <> nrcKey2 And verification = 0 Then
                    verification = 1

                    Set intersectionList = New Collection
                    For Each row1 In filteredNRC(nrcKey1)
                        For Each row2 In filteredNRC(nrcKey2)
                            For i = LBound(dayHeaders) To UBound(dayHeaders)
                                Dim dayCol As Integer
                                dayCol = FindColumn(ws2, CStr(dayHeaders(i)), 14)

                                timeRange1 = ws2.Cells(row1, dayCol).Value
                timeRange2 = ws2.Cells(row2, dayCol).Value

                If timeRange1 <> "" And timeRange2 <> "" Then
                    ParseTimeRange timeRange1, start1, end1
                    ParseTimeRange timeRange2, start2, end2

                    If (start1 < end2) And (end1 > start2) Then
                        tipo1 = ws2.Cells(row1, colTipo).Value
                        tipo2 = ws2.Cells(row2, colTipo).Value

                        If Not (tipo1 = "CLAS" Or tipo1 = "AYUD" Or tipo1 = "LABT") Or Not (tipo2 = "CLAS" Or tipo2 = "AYUD" Or tipo2 = "LABT") Then ' here goes all of the evaluation that are not in a specific day, but on a semester basis
                            dia1 = ws2.Cells(row1, colDia).Value
                            dia2 = ws2.Cells(row2, colDia).Value
                            If dia1 = dia2 Then
                                intersectionList.Add tipo1 & "-" & tipo2
                            End If
                        Else
                            intersectionList.Add tipo1 & "-" & tipo2
                        End If
                    End If
                End If
            Next i
        Next row2
    Next row1
```

```
      If intersectionList.Count > 0 Then
        result = ""
        For Each Item In intersectionList
          result = result & Item & ", "
        Next Item
        result = Left(result, Len(result) - 2)
      Else
        result = "No intersection"
      End If

      ws1.Cells(cell1.Row, colTope).Value = result
    End If
  Next nrcKey2
Next nrcKey1

If intersectionList.Count > 0 Then
  result = ""
  For Each Item In intersectionList
    result = result & Item & ", "
  Next Item
  result = Left(result, Len(result) - 2)
Else
  result = "No intersection"
End If

ws1.Cells(cell1.Row, colTope).Value = result
```

```
            ws1.Cells(cell1.Row, colCantidadTopes).Value = intersectionList.Count
        End If
    Next cell1

    ws2.AutoFilterMode = False
    ws2.Rows.Hidden = False ' Unhide all rows
    wb2.Close SaveChanges:=False
    MsgBox "Processing completed!", vbInformation
End Sub


Function FindColumn(ws As Worksheet, header As String, Optional headerRow As Long = 1) As Integer
    Dim cell As Range
    On Error Resume Next

    ' Iterate through the header row to find the matching header
    For Each cell In ws.Rows(headerRow).Cells
        If Trim(Replace(cell.Value, Chr(10), " ")) = Trim(Replace(header, Chr(10), " ")) Then
            FindColumn = cell.Column
            Exit Function
        End If
    Next cell

    ' If no match is found, raise an error
    MsgBox "Header '" & header & "' not found in row " & headerRow & "!", vbCritical
End Function
```

```vb
Sub ParseTimeRange(timeRange As String, ByRef startTime As Date, ByRef endTime As Date)
    Dim times() As String
    times = Split(timeRange, "-")
    If UBound(times) = 1 Then
        startTime = TimeValue(times(0))
        endTime = TimeValue(times(1))
    Else
        Err.Raise vbObjectError + 513, , "Invalid time range: " & timeRange
    End If
End Sub

Function RegularFilterNRC(ws2 As Worksheet, nrc1 As String, nrc2 As String, colNRC As Integer, filterRange As Range) As Object
    Dim i As Long
    Dim filteredRows As Object
    Set filteredRows = CreateObject("Scripting.Dictionary")

    ' Loop through all rows (starting from row 15 to avoid header)
    For i = 15 To ws2.Cells(ws2.Rows.Count, colNRC).End(xlUp).Row
        ' Check if the NRC value in the current row matches either nrc1 or nrc2
        If ws2.Cells(i, colNRC).Value <> nrc1 And ws2.Cells(i, colNRC).Value <> nrc2 Then
            ws2.Rows(i).Hidden = True
        Else
            ' Make sure the row is visible if it matches either nrc1 or nrc2
            ' Debug.Print "Row num=" & i

            ' If the NRC key doesn't exist in the dictionary, add it with an empty collection
            If Not filteredRows.Exists(ws2.Cells(i, colNRC).Value) Then
                filteredRows.Add ws2.Cells(i, colNRC).Value, New Collection
            End If

            ' Add the row number to the collection for the corresponding NRC
            filteredRows(ws2.Cells(i, colNRC).Value).Add i
            ws2.Rows(i).Hidden = False
        End If
    Next i

    ' Return the dictionary of filtered rows
    Set RegularFilterNRC = filteredRows
End Function
```