

# REPRESENTACIÓN DE GRÁFICAS Y TABLAS DE DATOS CON MOVIMIENTO EN EXCEL

**JUAN JESÚS BERNAL GARCÍA**

[juanjesus.bernal@upct.es](mailto:juanjesus.bernal@upct.es)

*Universidad Politécnica de Cartagena (UPT)/Métodos Cuantitativos e Informáticos  
C/Real,3. 30201 Cartagena*

**SOLEDAD M<sup>a</sup>. MARTÍNEZ M<sup>a</sup> DOLORES**

[soledad.martinez@upct.es](mailto:soledad.martinez@upct.es)

*Universidad Politécnica de Cartagena/Métodos Cuantitativos e Informáticos  
C/Real,3. 30201 Cartagena*

**PABLO BERNAL SOTO**

[pablo.bernal@gmail.com](mailto:pablo.bernal@gmail.com)

*UPCT /Grupo de Investigación de "Métodos cuantitativos e informáticos para la economía y la empresa  
C/Real,3. 30201 Cartagena*

RESUMEN: Continuando con anteriores trabajos sobre representación de funciones en Excel, se trata de recurrir a opciones especiales y a la programación en VBA para dotar a dichas gráficas de automatismos que nos permitan darles movimiento, tanto de rotación como de avance lineal, pudiendo generar una vista en cualquier posición. Las gráficas así elaboradas sirven tanto para visionar dinámicamente, no solo las funciones matemáticas, sino también la representación temporal de datos económico-financieros, llegando incluso a emular los gráficos de movimiento (denominados Motion Chart) de Google Drive.

*Palabras clave:* Excel, representación de funciones, gráficos en movimiento.

ABSTRACT: Following previous works on graphical representation of functions, this article introduces special options and VBA programming methods to provide motion capabilities to these graphs, such as spinning or lineal advance, which allow to set a the chart view from any position. These charts serve to display not only mathematical functions but also timeline representation of economic and financial data, even to emulate the Motion Charts of Google Drive.

*Keywords:* Excel, graphical representation of functions, motion charts

## 1. Introducción

Excel, a semejanza de otras hojas de cálculo, cuenta con numerosas funciones de tipo matemático que permiten representaciones gráficas de calidad, así como es posible mostrar de forma visual Tablas de datos, no obstante estas son de tipo estático. En un artículo anterior se expuso cómo representar una función en Excel, simplemente escribiendo su expresión en una celda [1]. En el presente trabajo abordaremos cómo lograr que dichas gráficas puedan ser modificadas a partir de unos controles, para que sea posible mostrar distintas visiones espaciales de las mismas, mediante rotaciones y avances lineales. Los efectos conseguidos son aplicables tanto para funciones de tipo matemático como económico-empresarial, lo cual las hace interesante para la investigación y para la docencia.

### 1.1. Antecedentes

Ciertamente existen programas en el mercado para la representación móvil de gráficas, normalmente de tipo matemático; algunos son de software libre en Internet, y de entre ellos el más conocido es el GeoGebra [2], que en su versión 4 incluye una versión Web, tratándose fundamentalmente de representaciones geométricas, donde la movilidad se consigue variando la posición de un punto en tiempo real. Existen no obstante, páginas en Internet donde se pueden encontrar ejemplos de gráficas en 3D, también a partir de esta aplicación [3] (Figura 1). Además existen complementos para dotar a Excel de mayor potencia a la hora de representar funciones, como la aplicación XLFit [4], pero dichas soluciones, lógicamente, no son gratuitas.

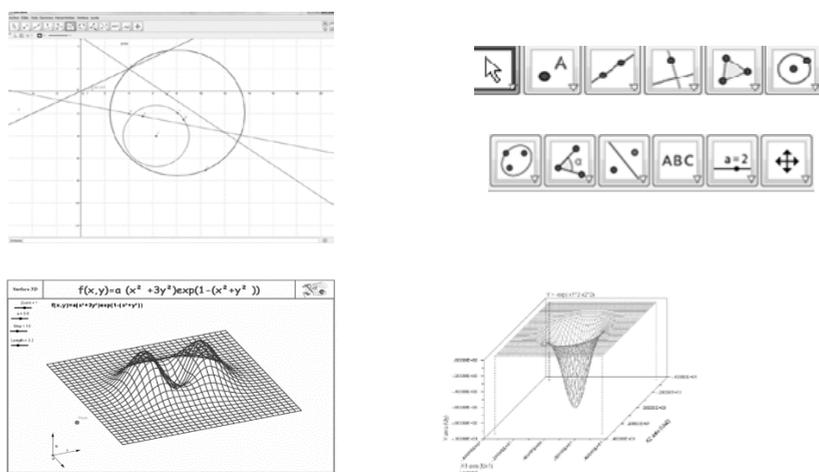


Figura 1: Ejemplos en GeoGebra Web y en 3D y en XLFit

## 2. Gráficos en movimiento

Veamos a continuación, distintas propuestas para conseguir gráficos en movimiento:

### 2.1. Gráficas sin refresco de pantalla

Cuando se varían los datos de una Tabla, el gráfico correspondiente lo hace de forma global, es decir, se muestra en la hoja modificada, pero puede resultar interesante, sobre todo si se desea visualizar un proceso temporal, ir mostrando la evolución instantánea del mismo. La actualización continua del gráfico proporciona una secuencia tipo vídeo de dicha evolución, por esta razón le dimos el nombre de “osciloscopio”. Así, en la *Figura 2*, se muestra la simulación del número pi, mediante un algoritmo de simulación, para lo cual se elaboró una macro en VBA (*Visual Basic for Applications*) [5], [6] y [7], que realiza trescientas tiradas, almacenando en una Tabla el resultado obtenido para cada una de ellas; se añade de esta forma una línea en la misma y un punto en el gráfico correspondiente, lo que proporciona una actualización o moviendo en tiempo real, que se consigue desactivando el “refresco de pantalla” de la macro, mediante la instrucción *Application.ScreenUpdating = False* [8]

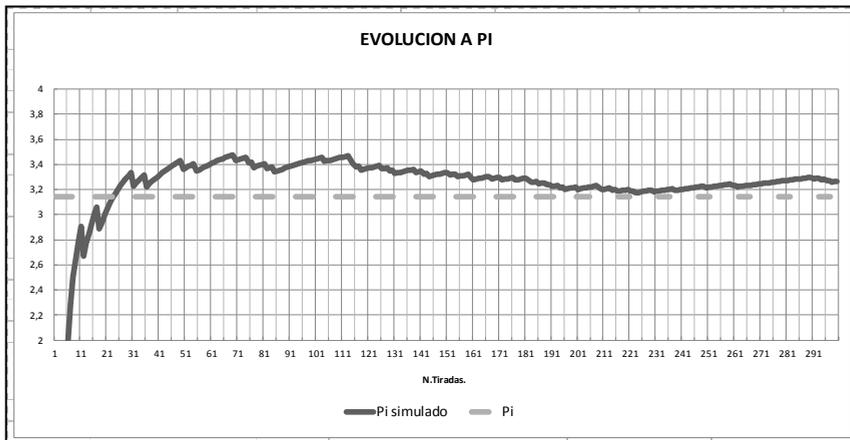


Figura 2: Representación en tiempo real de una gráfica

### 2.2 Gráficas automáticas con barra de desplazamiento

En otros trabajos explicamos cómo conseguir variaciones automáticas en los gráficos gracias al uso de botones de formulario y/o “controlesX” [8], y los hemos añadido a la representación automática citada [1], con lo cual hemos conseguido contar con una hoja que representa de forma automática una ecuación y permite cambiar sus parámetros.

A modo de ejemplo se ha representado las curvas de ingresos y costes totales, por cuya diferencia determinaremos el beneficio. Para unos costes fijos y variables prefijados, y un precio de venta dado, las rectas dependerán del número de unidades vendidas

(Figura 3). Partiendo de la gráfica superior, podemos, mediante una “barra de desplazamiento”, dotar de movimiento al gráfico al variar el número de unidades vendidas, consiguiendo que para el caso de 8 unidades vendidas, encontremos el beneficio cero o umbral de rentabilidad.

Se podría utilizar un programa de captura de pantalla como *Camtasia Studio* [9], para grabar la evolución del proceso.

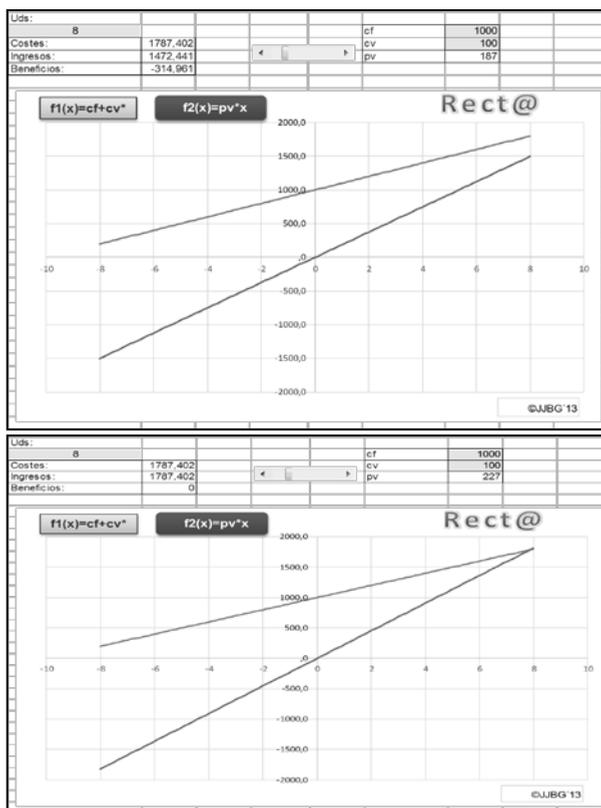


Figura 3: Líneas de costes totales e ingresos

### 2.3 Movimiento con VBA

La alternativa a los controles anteriores es la programación mediante código en VBA, es posible realizar un ciclo con variaciones incrementales de una función dada; así en el ejemplo presentado SENO(X), la Figura 4 muestra una instantánea de dicha gráfica en movimiento. Con la finalidad de poder fijar un momento dado en la representación, disponiendo así de una “foto fija”, se ha añadido en la macro realizada, y que se adjunta en el Anexo A, unas líneas de código que detienen el movimiento al volver a pulsar el

botón de “mover/parar”.

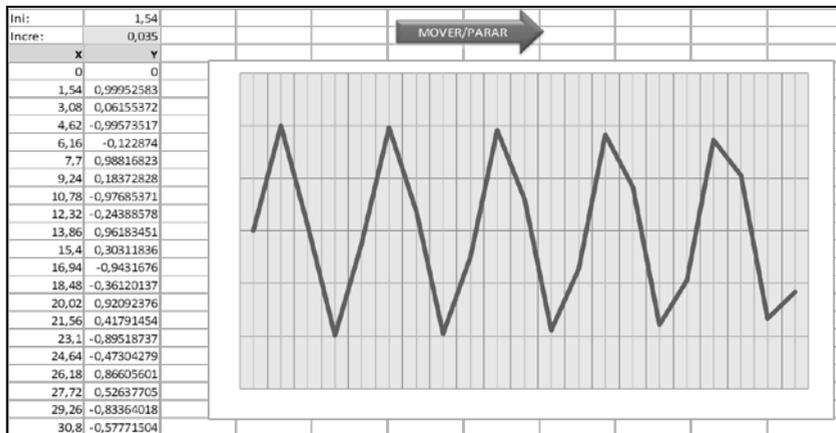


Figura 4: Gráfica SENO(X) en movimiento

## 2.4 Movimientos espaciales

Si representamos una función en 3D, lo que es posible en Excel mediante la elaboración de una Tabla de dos dimensiones, podremos programar macros para conseguir rotar en sentido horizontal, voltear verticalmente, o variar la perspectiva de dicha gráfica. A modo de ejemplo, en la Figura 5 se ha representado la parábola hiperbólica:  $z=x^2-y^2$ , y en la Figura 6:  $z=x^2+y^2$ ; además se han añadido “formas” o “botones” [10] que accionan los macros citados (*Anexo B*). Se ha incluido asimismo en todos los casos, la posibilidad de parada, con lo cual podemos obtener distintas vistas en 3D. También es posible, en el visor de VBA pulsar {F5} y realizar el proceso paso a paso, lo que permite visualizarlo a voluntad.

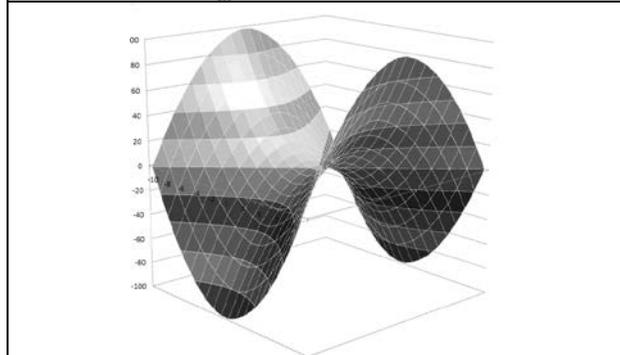
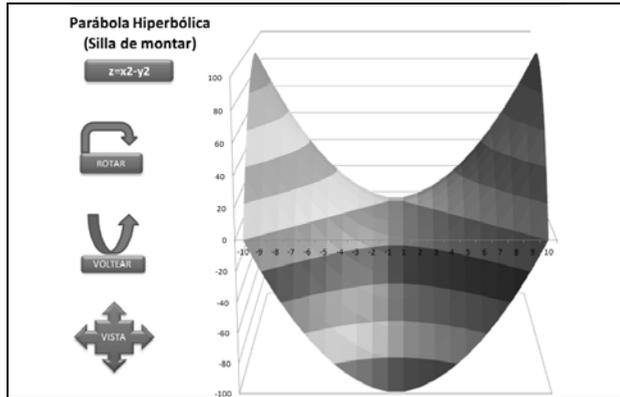


Figura 5: Gráfica  $z=x^2-y^2$  en 3D con distintas vistas

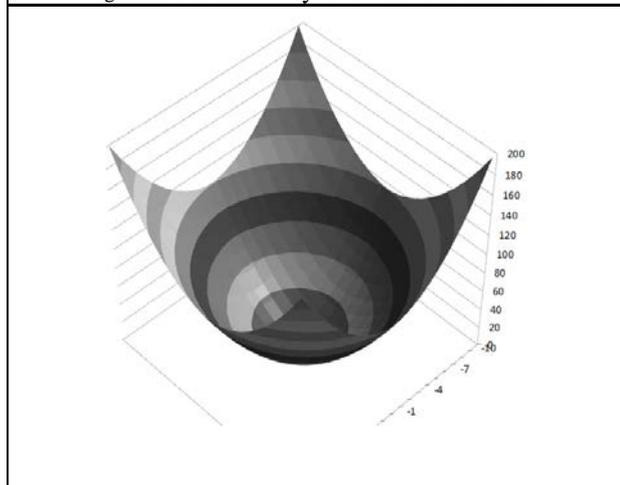


Figura 6: Gráfica  $z=x^2+y^2$  en 3D

### 2.5. Gráficos con datos temporales

Si algún tipo de datos se presta al movimiento, son aquellos de tipo económico-empresarial, o series temporales. En el supuesto que aquí presentamos, se dispone de una larga lista de ventas realizadas comparadas con las previsiones de demanda, ordenadas por fechas; en concreto hablamos de mil entradas desde el 2009 al 2012.

El programa realizado permite elegir el día de inicio, el número de observaciones a mostrar a la vez, y el incremento o día de desplazamiento. La *Figura 7* presenta parte de la Tabla de datos, la entrada de parámetros citados, una barra de desplazamiento para variar el incremento, más una instantánea de la gráfica, ya que el parar el macro nos permite también mostrar los 20 días seleccionados. Se ha incluido la programación en VBA en el *Anexo C*.

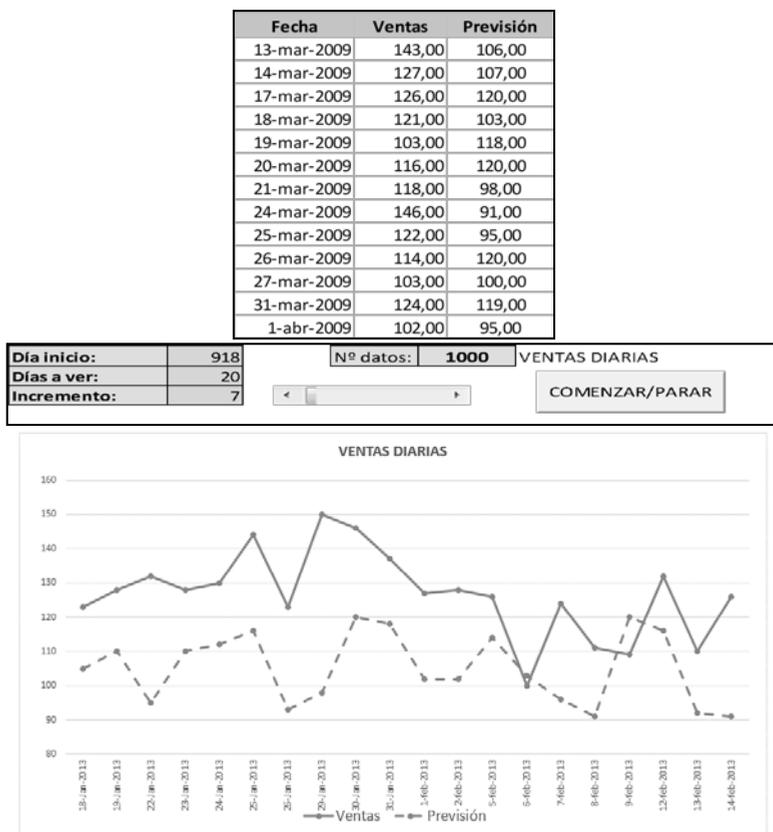


Figura 7 Gráfica en movimiento de ventas reales y esperadas

### 3. Gráfico de movimiento (Motion Chart) en Excel con VBA

Continuando con la anterior idea de dotar a los datos empresariales de movimiento, señalar que *Google Drive* (antes *Google Doc*) [11], ya cuenta con la posibilidad de mostrar gráficos de burbuja en movimiento (*Figura 8*).

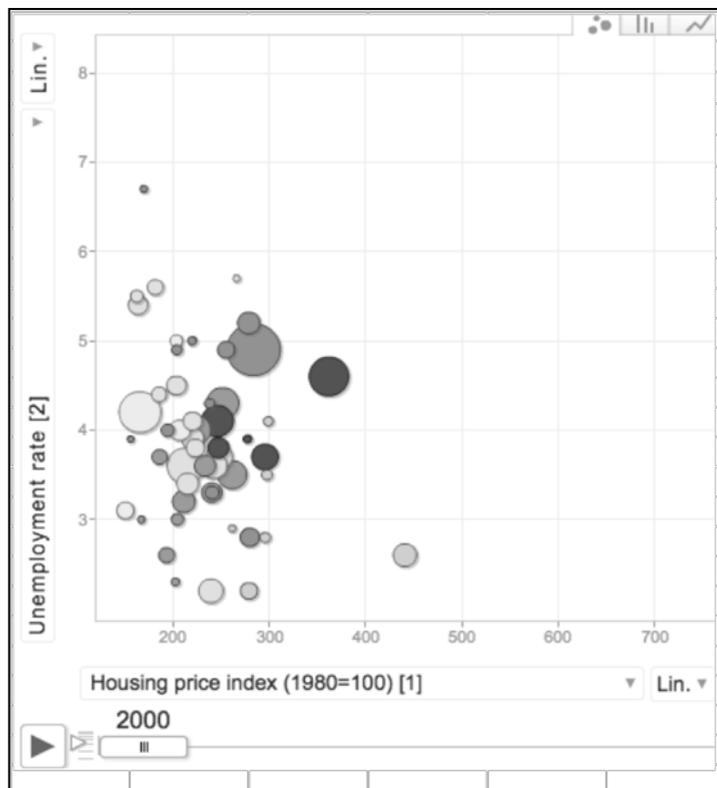


Figura 8 Motion Graf de Google drive

Nos propusimos emular con Excel ese tipo de gráficos, para lo cual recurrimos a los de burbuja 3D, que contienen una doble información, según su posición en el plano, y su tamaño. Para ello se parte de una Tabla con los valores de las ventas y el beneficio neto (BN) obtenido, en un listado por años y meses (*Figura 9*).

Meses	Año	número mes	Ventas	BN
enero	2010	1	29.155,10 €	380.536,71 €
febrero	2010	2	29.155,02 €	380.536,71 €
marzo	2010	3	44.264,54 €	380.536,71 €
abril	2010	4	27.279,94 €	380.536,71 €
mayo	2010	5	33.945,16 €	380.536,71 €
junio	2010	6	36.168,90 €	380.536,71 €
julio	2010	7	34.516,12 €	380.536,71 €
agosto	2010	8	15.878,73 €	380.536,71 €
septiembre	2010	9	27.886,96 €	380.536,71 €
octubre	2010	10	29.173,12 €	380.536,71 €
noviembre	2010	11	40.826,75 €	380.536,71 €
diciembre	2010	12	32.286,37 €	380.536,71 €

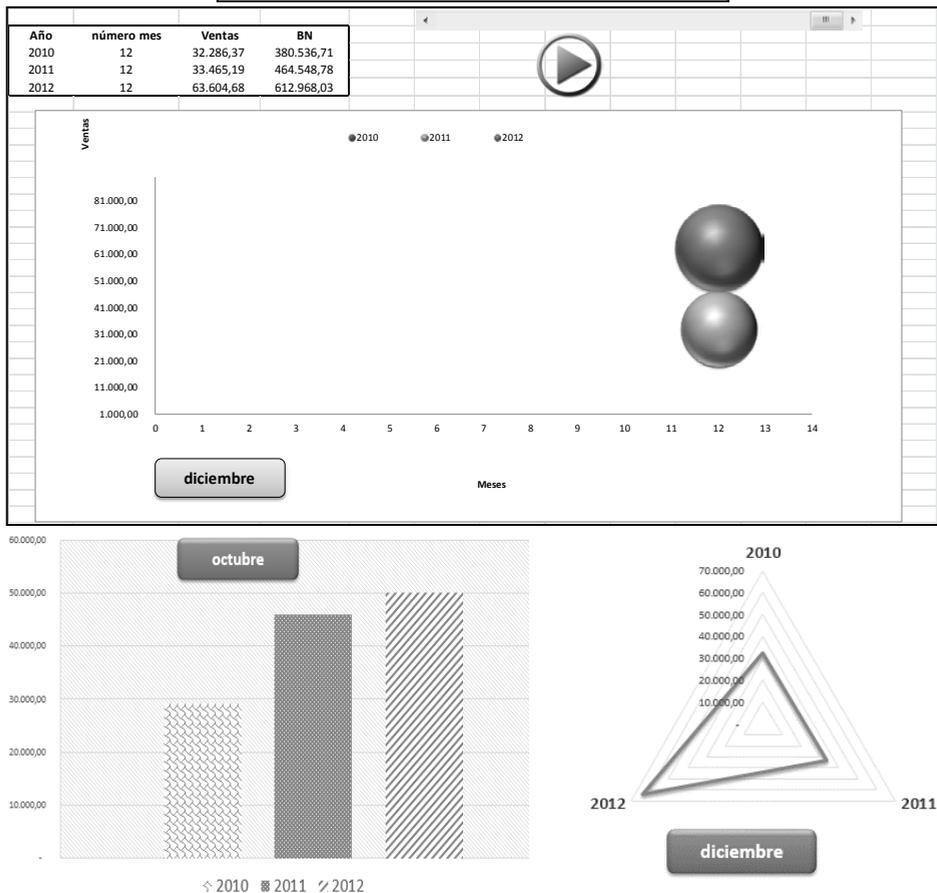


Figura 9: Gráfico en movimiento tipo Motion Char de bolas, barras y radial

Tras preparar una Tabla intermedia, que va extrayendo los datos de tres en tres (por año), podemos representar el gráfico de burbujas, al que posteriormente hemos dotado de una barra de desplazamiento. Si se desea, podemos variar los datos mostrados, o pulsar el icono de avance (>), que activa la macro correspondiente. Como mejora hemos

añadido la forma de texto variable con el mes [8], y en el código la instrucción (Sleep n), nos permite ralentizar el movimiento todo lo deseado para apreciar con detalle la evolución del mismo (*Anexo D*). Se ha probado, con esta técnica, otros tipos de gráficos en movimiento, resultando los más interesantes los de tipo barras y radial.

#### 4. Futuras mejoras

Seguimos trabajando para obtener mejoras en esta línea de investigación, siendo la más inmediata la de unir las dos técnicas presentadas en Rect@, la de representación automática de gráficas en Excel [1] con la de dotarlas a su vez de movimiento, del presente trabajo.

#### 5. Conclusiones

Volvemos a recalcar que la utilización avanzada de la hoja de cálculo Excel, permite conseguir mayores prestaciones, en un entorno muy versátil. Lo que lo hace asequible para la docencia, además de para la investigación y el mundo empresarial.

#### Referencias Bibliográficas (Ejemplos)

1. Bernal García, J.J, Martínez M<sup>a</sup> Dolores, S.M. y Bernal Soto, P. Representación automática de funciones en Excel y su aplicación docente, Revista Rect@. Vol. **12**. Pág. 141-157. 20112-412. 2012.
2. <http://www.geogebra.org/cms/es/download/> (13/05/13)
3. <http://dmentrard.free.fr/GEOGEBRA/Maths/Surface3D/surfexpoMD.html> (13/05/13)
4. <http://idbs-xlfit5.software.informer.com/> (13/05/13)
5. Walkenbach,, J. Excel 2010.Programación con VBA, *Anaya Multimedia.-Wiley* (Madrid, 2011)
6. Walkenbach, J. , Excel 2010. Los mejores trucos, *Anaya multimedia*, (Madrid, 2011)
7. Jelen, B. y Syrstad,T. Excel. Macros y VBA. Trucos esenciales, *Ed Anaya*, (Madrid, 2008)
8. Bernal García, J.J. Aportaciones para la mejora de la presentación grafica de datos cuantitativos en Excel, Revista Rect@. Vol. **16**: 41242-412, XVI Congreso ASEPUMA y IV Encuentro Internacional. (Cartagena, 2008).
9. <http://www.techsmith.com/camtasia.html> (13/05/13)

10. Bernal García, J.J. Nuevas contribuciones a la mejora de la representación gráfica en Excel, *Anales ASEPUMA. N° 18.2010*, XVIII Congreso ASEPUMA y VI Encuentro Internacional (Santiago de Compostela, 2010)
11. <http://support.google.com/drive/bin/answer.py?hl=en&answer=1047434> (13/05/13)

## Anexo A

### Código VBA para función SENO() en movimiento

```

Option Explicit
Public MovtriIsRunning As Boolean
Sub movtri()
If MovtriIsRunning Then
    MovtriIsRunning = False
End
End If
MovtriIsRunning = True
Dim i As Long, incre As Double
Range("ini") = 0
incre = Range("incre")
For i = 1 To 150
    DoEvents
    Range("ini") = Range("ini") + incre
DoEvents
StopIt:
    Next i
    Rem Range("ini") = 0
End Sub

```

## Anexo B

### Código VBA para ROTAR, VOLTEAR y PERSPECTIVA

```

Sub Rotar()
Dim i As Long
If SilladuraIsRunning Then
    SilladuraIsRunning = False
End
End If

```

```
SilladuraIsRunning = True
With ThisWorkbook.Sheets("Silladura").ChartObjects(1).Chart
    For i = 0 To 360 Step 8
        .Rotation = i
        'Elevation = i - 90
        DoEvents
    Next i
End With
SilladuraIsRunning = False
End
```

```
SubSub voltrear()
    Dim i As Long
    ' Elevation range: -90 to +90
    If SilladuraIsRunning Then
        SilladuraIsRunning = False
        End
    End If
    SilladuraIsRunning = True
    With ThisWorkbook.Sheets("Silladura").ChartObjects(1).Chart
        For i = -90 To 90 Step 2
            .Elevation = i
            Application.Wait (Now + 0.000002)
            DoEvents
        Next i
        .Elevation = 15
    End With
    SilladuraIsRunning = False
End Sub
```

```
Sub perspectiva()
    Dim i As Long
    ' Perspective range = 0 to 100
    If SilladuraIsRunning Then
        SilladuraIsRunning = False
        End
    End If
    SilladuraIsRunning = True
```

```
With ThisWorkbook.Sheets("Silladura").ChartObjects(1).Chart
  For i = 0 To 100 Step 2
    .Perspective = i
    DoEvents
  Next i
  .Perspective = 30
End With
SilladuraIsRunning = False
End Sub
```

### **Anexo C**

Public AnimationInProgress As Boolean

```
Sub AnimateChart()
  Dim Dstar As Long, r As Long
  If AnimationInProgress Then
    AnimationInProgress = False
  End
End If
AnimationInProgress = True
Dstar = Range("diainicio")
For r = Dstar To Range("ndatos") - Range("ndiasven") Step
Range("Incr")
  Range("diainicio") = r
  DoEvents
Next r
AnimationInProgress = False
End Sub
```

### **Anexo D**

**Código VBA para Motion Graf para Rect@**

```
Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Private Sub Play()
  Dim i As Integer
  For i = 1 To 12:
    If grafmovi_rectaIsRunning Then
```

```
    grafmovi_rectaIsRunning = False
    End
End If
    Range("d5").Value = i
    Application.Calculate
    Sleep (500)
Next
End Sub
```