



15 al 30 de septiembre de 2015

Creación de *CDF's* para la enseñanza del tema de funciones con *Wolfram Mathematica*

Experiencias y recursos en educación virtual 2.0. Los cursos
MOOC abiertos masivos en línea: Comunicación de
experiencias, evaluación e impacto de esta nueva tendencia

Enrique Vílchez Quesada
Universidad Nacional de Costa Rica
Escuela de Informática

Resumen: los *CDF's* (*Computable Document Format*) constituyen archivos interactivos a través de los cuáles es posible generar aplicaciones Web utilizando el conocido software comercial *Mathematica*. Con la simple instalación de un *plug in*, cualquier usuario de este tipo de recursos, puede visualizar conceptos, o eventualmente manipular objetos que representan estructuras matemáticas de distinta índole. Con el presente taller se espera que los participantes comprendan los principios básicos para la creación de este tipo de recursos didácticos enfocados en el tema de funciones. Las funciones constituyen ya desde hace varias décadas, un reto pedagógico cuando se analiza la forma en cómo usualmente es abordado el contenido en los salones de clase a nivel nacional. En este sentido la generación de *CDF's* brinda nuevas opciones didácticas utilizando una metodología de carácter complementario basada en lecciones de laboratorio.

Palabras clave: *cdf's*, *Wolfram*, *Mathematica*, funciones, enseñanza.

1. Introducción



15 al 30 de septiembre de 2015

Los *Computable Document Format* constituyen una novedosa forma de compartir aplicaciones desarrolladas en el conocido software comercial *Mathematica* a través de la instalación de un *plug in* denominado *Wolfram CDF Player*.

La empresa *Wolfram Research* distribuye de manera gratuita el *plug in* a través de la dirección electrónica: <http://www.wolfram.com/cdf-player/> en su versión 10.2 para distintas plataformas: Windows, MAC y Linux.

En el presente documento se socializan trece aplicaciones *CDF's* elaboradas por el autor, para trabajar contenidos relacionados con el concepto de función, pares ordenados, funciones polinomiales, rectas y parábolas. Además, se exponen algunos principios básicos sobre el uso del software comercial *Mathematica* y el comando *Manipulate*, aspectos esenciales en la construcción de *CDF's*.

2. Prerrequisitos

Manejo básico del software *Mathematica*.

3. Objetivos

- Mostrar comandos básicos del software *Mathematica*.
- Comprender los principios esenciales para la creación de un *CDF*.
- Aplicar los conocimientos generales vinculados con el desarrollo de *CDF's*, en la creación de aplicaciones para el tema de funciones.
- Analizar ejemplos de *CDF's* creados por el autor, desde un punto de vista didáctico y tecnológico.

4. Contenido

La generación de *CDF's* requiere un conocimiento previo sobre algunos comandos de uso frecuente del software *Mathematica*. Se iniciará con ciertos ejemplos clave en esta dirección.

4.1 Conocimientos básicos de *Wolfram Mathematica*

- Simplificar una expresión: **Simplify**[(6 x² - 10)/2 + (6 x² - 6 x + 12)/-6 - (8 x² + 12 x - 20)/4]
- Expandir una expresión: **Expand**[(1 - x)¹⁰]
- Factorizar: **Factor**[12 x^(2 m) - 7 x^m - 12]
- Multiplicar expresiones algebraicas fraccionarias: **Simplify**[(a² - b²)/(2 a² - 3 a b + b²)*(2 a² + 5 a b - 3 b²)/(a² + 4 a b + 3 b²)*(a² - 2 a b - 3 b²)/(a² - 4 a b + 3 b²)]
- Sumar expresiones algebraicas fraccionarias: **Together**[(2 - (a + 5)/(a + 2))/(a² - 1) + (a² - (3 a² - a)/(a + 2))/(a³ + 1)]
- Evaluar: **FullSimplify**[1 + 1/x (1 + 1/x (1 + 1/x (1 + (1 + 1/x)/x)))] /. x -> 1 + y²]
- Resolver una ecuación de manera exacta: **Solve**[10 x² + 59 x + 62 == 0 && x <= 0, x]
- Resolver una ecuación en formato decimal: **NSolve**[10 x² + 59 x + 62 == 0, x]
- Resolver una inecuación: **Reduce**[10 x² + 59 x + 62 >= 0, x, Reals]
- Crear una función: **A**[r_] := (2 r - 1)/(r² + 2 r - 3); {A[3], A[1/2], A[-1], **Simplify**[A[x² + 1]]}
- Graficar una lista de pares ordenados: **ListPlot**[{0, -1}, {1, 2}, {2, 4}], **PlotStyle** -> **PointSize**[0.025]
- Graficar funciones: **Plot**[{x/(x² + 1), x²}, {x, -4, 4}, **PlotRange** -> {-1, 4}]
- Hallar donde decrece 1/(x+1): **Reduce**[D[1/(x + 1), x] < 0, x]
- Crear un método para determinar si un número es positivo o cero: **Positivo**[num_] := **If**[num >= 0, True, False]



15 al 30 de septiembre de 2015

- $1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{6} + \frac{1}{8} - \frac{1}{10} + \frac{1}{12} - \dots$
 Cacular: `SumaSerie[n_] := Module[{suma = 1, signo = -1}, For[i = 1, i <= n, suma = suma + signo/(2* i); signo = -signo; i++]; suma]`

4.2 Creación de CDF's

Los *Computable Document Format* se crean utilizando principalmente el comando del software *Mathematica* denominado *Manipulate*. La siguiente sección muestra algunos ejemplos sobre sus posibles usos.

4.2.1 Ejemplos de uso básico: *Manipulate*

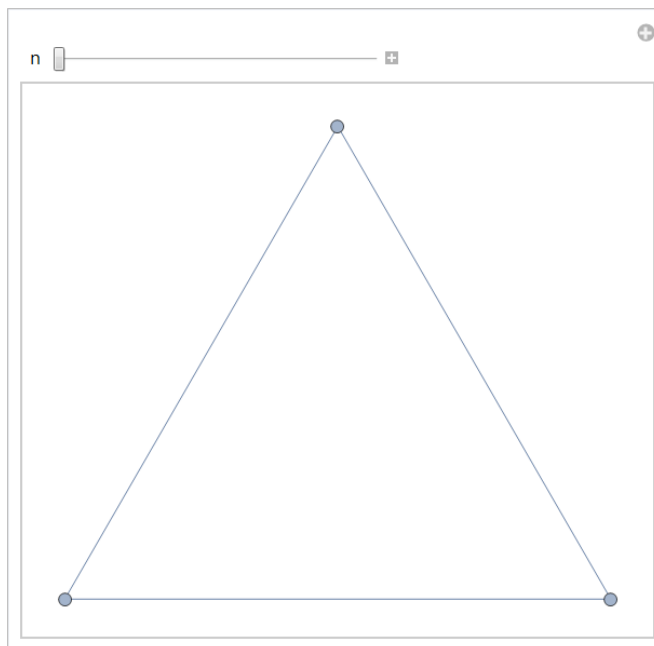
- Suma algebraica: `Manipulate[Row[{n (x - 1)/(x + 1), "+", m (x + 1)/(x - 1), "=", Together[n (x - 1)/(x + 1) + m (x + 1)/(x - 1)]}, {n, 1/2, 1/3, 1/144}, {m, 1/2, 1/3, 1/144}]`

$$\frac{-1 + x}{2 (1 + x)} + \frac{1 + x}{2 (-1 + x)} = \frac{1 + x^2}{(-1 + x) (1 + x)}$$

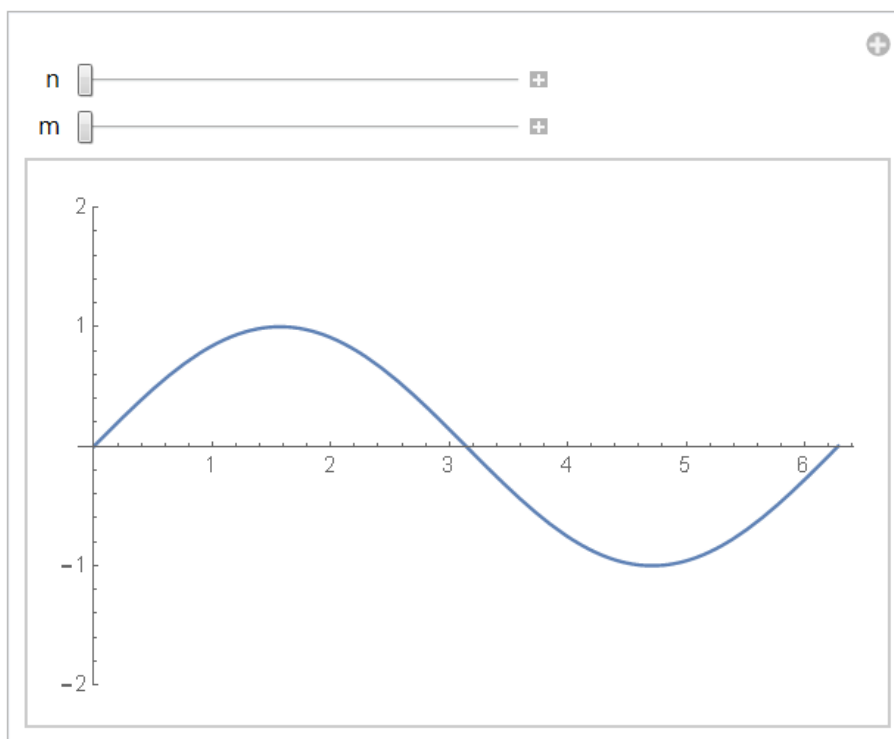
- Expandir un binomio: `Manipulate[Expand[(a + b)^n], {n, 1, 100, 1, Appearance -> "Labeled"}]`

$$a + b$$

- Mostrar un grafo completo: `Manipulate[CompleteGraph[n], {n, 3, 30, 1}]`



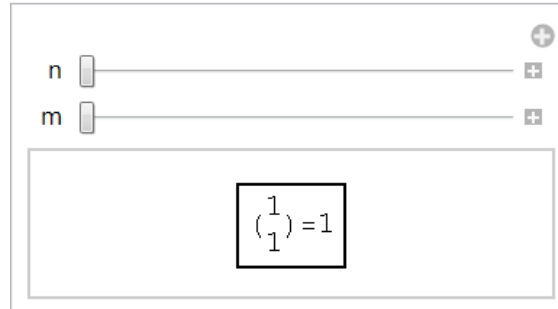
- Graficar una función: **Manipulate[Plot[n Sin[m*x], {x, 0, 2 \[Pi]}, PlotRange -> {-2, 2}], {n, 1, 2, 0.1}, {m, 1, 10, 1}]**



- Cálculo del binomial: **Manipulate[If[m > n, m = n, Framed[Row[{"(", Column[{n, m}, Center], ")="}, Binomial[n, m]]]], {n, 1, 10, 1}, {m, 1, 10, 1}, Alignment -> Center]**



15 al 30 de septiembre de 2015



En general el comando *Manipulate* como se observa en los ejemplos genera un tipo especial de deslizador denominado “*manipulador*”. En un *manipulador* se puede especificar en qué valor se inicia. Además, posee controladores visibles al usuario al presionar el “+” contiguo al *manipulador*. Por otra parte, el “+” del *frame* permite adicionalmente activar un autorun y generar marcas de recorrido.

Manipulate posee los siguientes atributos:

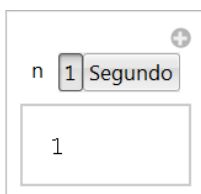
- Appearance ->"Labeled": muestra en pantalla el valor del parámetro.
- Appearance->"Open": en un manipulador abre los controladores.
- Alignment->Center: centra el contenido.
- Framed: muestra en un cuadro el contenido.
- Row: ordena el contenido de una ventana en filas.
- Column: ordena el contenido de una ventana en columnas.
- Grid: ordena el contenido de una ventana en filas y columnas.
- ControlPlacement->Left (Right, Top, Bottom): da ubicación a los manipuladores y cualquier otro controlador.
- Delimiter: añade delimitadores para acomodar los controladores.
- "Mensaje": añade a la ventana de salida el string "Mensaje", se puede emplear junto con Style["Mensaje", 12, Bold].
- Initialization:->(): permite inicializar variables y funciones.
- Locator: añade puntos de localización.

- **Dynamic:** los objetos de visualización pueden quedar junto a los controladores usando este comando.

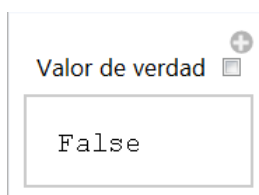
4.2.2 Tipos de controladores en *Manipulate*

Los *manipuladores* no son la única opción de control al emplear el comando *Manipulate*. Es posible crear también: botones, popmenú y check.

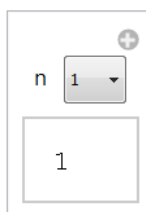
- **Botones:** `{n, {B1, B2}}` añade dos botones B1 y B2. Ejemplo: **`Manipulate[n, {n, {1, 2 -> "Segundo"}}]`**



- **Check:** `{n, {True, False}}`: agrega un check. Ejemplo: **`Manipulate[n, {{n, False, "Valor de verdad"}, {True, False}}]`**



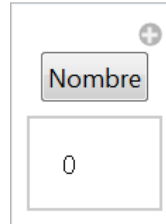
- **Popmenú:** `{n, {B1, B2}, ControlType->PopupMenu}`: añade un combo o popmenú con B1 y B2. El *ControlType->SetterBar* fuerza la creación de una serie de botones. El *ControlType->Manipulator* define un controlador como un *manipulador*. El *ControlType->Slider* añade un *slider*. El *slider* y el *manipulador* se diferencian pues el slider no tiene controladores (play, pause, etc). Ejemplo: **`Manipulate[n, {n, {1, 2}, ControlType -> PopupMenu}]`**



- **Button:** un botón poder ser insertado también, mediante el uso del comando *Button*. Ejemplo: **`Manipulate[n, Button["Nombre", n += 4], Initialization :-> (n = 0)]`**



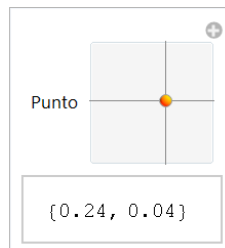
15 al 30 de septiembre de 2015



- Campos de texto: los campos de texto (input field) en el *Wofram Player Free* aceptan datos exclusivamente numéricos. Por ejemplo: **Manipulate[n, {{n, 1, "Dato"}}]**



- Slider 2d: $\{\{n, \{0,0\}, \text{"Punto"}\}, \{x_{\min}, y_{\min}\}, \{x_{\max}, y_{\max}\}\}$ crea un slider 2d ubicado en la posición (0,0). Por ejemplo: **Manipulate[n, {{n, {0, 0}, "Punto"}, {-1, -1}, {1, 1}}]**



4.2.3 Recomendación para la creación de CDF's

Antes de salvar un archivo *.nb* de *Mathematica* al formato *CDF* se recomienda seguir los siguientes pasos con el objetivo de limpiar las variables que fueron utilizadas.

1. Ejecute el comando `Quit[]`, o bien, reinicie el *kernel*: `Evaluation/Quit kernel/Local`.
2. Desactive la barra de sugerencias en: `Edit -> Preferences` y desmarque la opción "Show Suggestions Bar after last output".

3. Corra el notebook.
4. Esconda el código fuente.
5. Desactive la edición de las celdas:
 - a. `nb=First[Notebooks["Nombre del archivo"]]`
 - b. `SetOptions[nb, Editable→False, ShowCellBracket→False]`
6. Salve como un *CDF*.

4.2.4 Ejemplos programados por el autor

Se presentan una serie de *CDF*'s elaborados por el autor del presente trabajo, específicamente para abordar contenidos en el campo de las funciones. Los ejemplos constituyen aplicaciones *CDF*'s bastante elaboradas y por ende, muestran los alcances que este tipo de archivos pueden brindar con fines didácticos.

1. **Tabla (crea una tabla de forma pseudoaleatoria con el objetivo de que el estudiante determine si corresponde o no a una función):**

```
Manipulate[Column[{If[m == 0, If[x == True, F1 = RandomAlfabeto[n];
F2 = RandomEnteros[n], F1 = RandomEnteros[n]; F2 =
RandomEnteros[n]; Grid[{Prepend[F1, "x"], Prepend[F2, "y"]}, Frame -
> All], If[x == True, F1 = RandomAlfabeto[n]; F2 = RandomEnterosE[n],
F1 = RandomEnteros[n]; F2 = RandomEnterosE[n]; Grid[{Prepend[F1,
"x"], Prepend[F2, "y"]}, Frame -> All]], If[s == True,
If[Length[DeleteDuplicates[F1]] == n && m == 0, "Es función", "No es
función"], If[s == True, If[Length[DeleteDuplicates[F1]] == n && m == 0
&& Length[DeleteDuplicates[F2]] == n, "Es inyectiva",
If[Length[DeleteDuplicates[F1]] == n && m == 0, "No es inyectiva"]]],
Button["Generar", m = RandomInteger[]; d++], Center], {{n, 2,
"N[Degree] de pares ordenados"}, 2, 10, 1}, Delimiter, {{x, False,
"Alfabeto x"}, {True, False}}, Delimiter, {{s, False, "Solución"}, {True,
False}}, Delimiter, {{se, True, "Semilla"}, {True, False}}, Delimiter,
Alignment -> Center, Initialization :-> (m = 0; d = 0; RandomEnteros[n_]
```



15 al 30 de septiembre de 2015

```

:= Module[{v = RandomInteger[]}, If[v == 0, If[se, SeedRandom[d]]; L =
RandomInteger[{1, n}, n]; L, If[se, SeedRandom[d]]; L =
RandomInteger[{1, n}, n]; While[Length[DeleteDuplicates[L]] !=
Length[L], If[se, d++; SeedRandom[d]]; L = RandomInteger[{1, n}, n]];
L]]; RandomAlfabeto[n_] := If[n <= 26, Module[{v = RandomInteger[]},
If[v == 0, If[se, SeedRandom[d]]; L = RandomInteger[{1, n}, n]; Alf =
Alphabet[]; For[i = 1, i <= Length[L], L[[i]] = Alf[[L[[i]]]]; i++]; L, If[se,
SeedRandom[d]]; L = RandomInteger[{1, n}, n];
While[Length[DeleteDuplicates[L]] != Length[L], If[se, d++;
SeedRandom[d]]; L = RandomInteger[{1, n}, n]; Alf = Alphabet[]; For[i
= 1, i <= Length[L], L[[i]] = Alf[[L[[i]]]]; i++]; L]]; RandomEnterosE[n_]
:= Module[{}, If[se, SeedRandom[d]]; L = RandomInteger[{1, n}, n - 1]; L
= Insert[L, " ", RandomInteger[{1, n - 1}]]; L]]
    
```

Nº de pares ordenados +

Alfabeto x

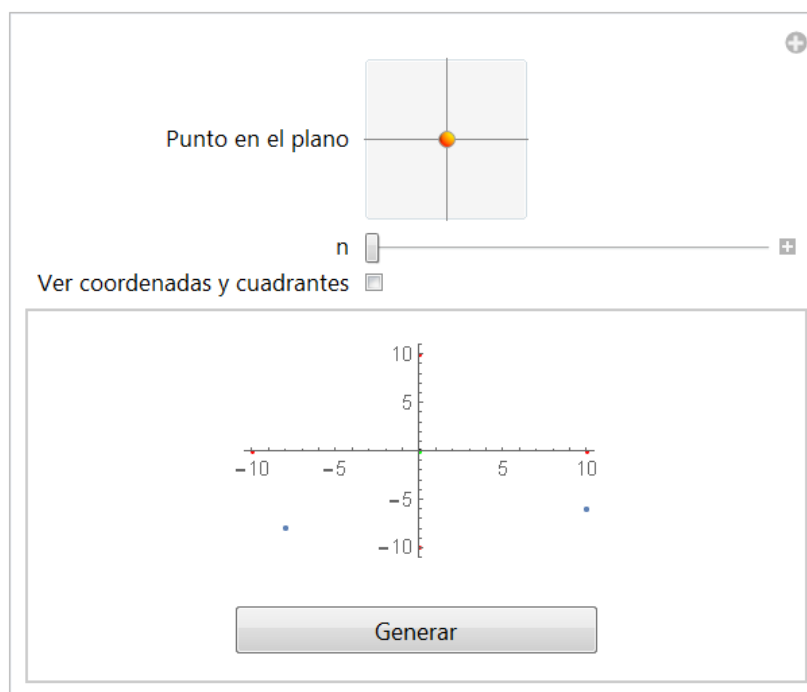
Solución

Semilla

| | | |
|---|---|---|
| x | 2 | 1 |
| y | 2 | 1 |

2. Pares (genera de manera automática una serie de pares ordenados donde el estudiante determina sus coordenadas y posición de acuerdo

a los cuadrantes): `Manipulate[Column[{Show[ListPlot[{{-10, 0}, {10, 0}, {0, -10}, {0, 10}}, PlotStyle -> {Red, PointSize[Small]}], ListPlot[GL[n]], Graphics[{Dotted, Line[{{p[[1]], 0}, p]}, Line[{{0, p[[2]], p}]}], Graphics[{PointSize[Small], Point[{p}, VertexColors -> {Green}]}]], If[m == True, LL, Button["Generar", d++; p = {0, 0}], Center], {{p, {0, 0}, "Punto en el plano"}, {-10, -10}, {10, 10}}, {n, 2, 10, 1}, {{m, False, "Ver coordenadas y cuadrantes"}, {True, False}}, Alignment -> Center, Initialization :-> (d = 0; GL[n_] := Module[{}, SeedRandom[d]; L = RandomInteger[{-10, 10}, {n, 2}]; LL = {}; For[i = 1, i <= Length[L], If[L[[i, 1]] > 0 && L[[i, 2]] > 0, LL = Append[LL, {L[[i]], "I"}]; If[L[[i, 1]] < 0 && L[[i, 2]] > 0, LL = Append[LL, {L[[i]], "II"}]; If[L[[i, 1]] < 0 && L[[i, 2]] < 0, LL = Append[LL, {L[[i]], "III"}]; If[L[[i, 1]] > 0 && L[[i, 2]] < 0, LL = Append[LL, {L[[i]], "IV"}]; If[L[[i, 1]] == 0 && L[[i, 2]] != 0, LL = Append[LL, {L[[i]], "Eje y"}]; If[L[[i, 1]] != 0 && L[[i, 2]] == 0, LL = Append[LL, {L[[i]], "Eje x"}]; i++; L))`



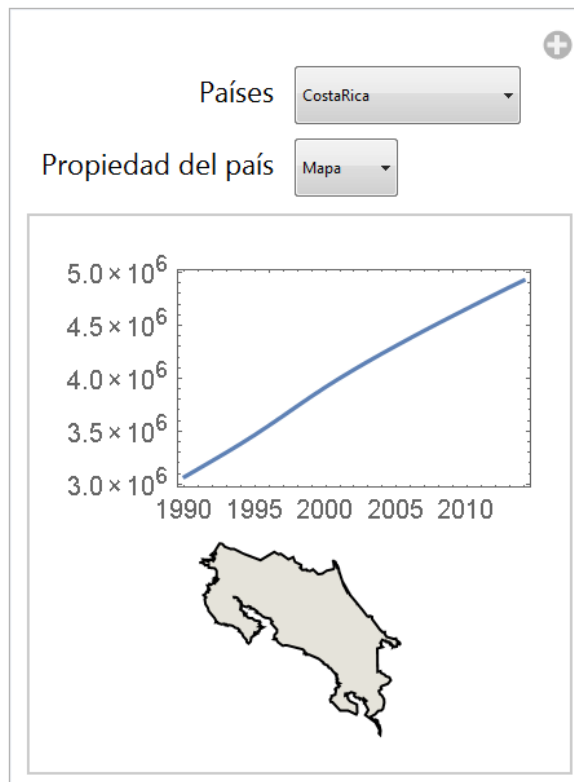
3. Población (muestra la gráfica años/población de distintos países del mundo, solo corre en *Mathematica* por el uso del comando *CountryData*): `Manipulate[Column[{DateListPlot[CountryData[n,`



15 al 30 de septiembre de 2015

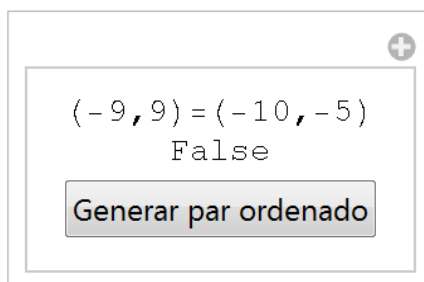
```

{"Population", {1990, 2014}}, AxesLabel -> Automatic], Propiedad[n,
m]}, Center], {{n, "CostaRica", "Países"}, Países[], ControlType ->
PopupMenu}, {{m, "Mapa", "Propiedad del país"}, {"Código", "Capital",
"Bandera", "Área", "Mapa", "Regiones"}, ControlType -> PopupMenu},
Alignment -> Center, Initialization :-> (L = CountryData[]; Países[] :=
Module[{Lista = {}}, For[i = 1, i <= Length[L], Lista = Append[List,
ToString[L[[i, 2]]]; i++]; Lista = Union[List, {"World"}]; Lista];
Propiedad[n_, m_] := Module[{}, Switch[m, "Código", CountryData[n,
"CallingCode"], "Capital", CountryData[n, "CapitalCity"], "Bandera",
CountryData[n, "Flag"], "Área", CountryData[n, "Area"], "Mapa",
CountryData[n, "Shape"], "Regiones", CountryData[n, "Regions"]]])]
    
```



4. Igualdad (aplicación creada para comprender el concepto de igualdad entre pares ordenados): Manipulate[Column[{Row[{"(", x, ",", y, ")"}],

```
z, ",", w, ")"]], {x, y} == {z, w}, Button["Generar par ordenado", vl =
RandomInteger[]; Generador[vl]], Center], Alignment -> Center,
Initialization :-> (x = RandomInteger[{-10, 10}]; y = RandomInteger[{-10,
10}]; z = RandomInteger[{-10, 10}]; w = RandomInteger[{-10, 10}]; vl =
RandomInteger[]; Generador[vl_] := Module[{}, If[vl == 0, x =
RandomInteger[{-10, 10}]; y = RandomInteger[{-10, 10}]; z =
RandomInteger[{-10, 10}]; w = RandomInteger[{-10, 10}], x =
RandomInteger[{-10, 10}]; z = x; y = RandomInteger[{-10, 10}]; w = y]]]
```



5. Interpolación (al ingresar una serie de pares ordenados encuentra un polinomio de mejor ajuste): `Manipulate[Column[{If[Length[Puntos] >= 2, If[bandera == 1 && validos, Show[Plot[InterpolatingPolynomial[Puntos, x], {x, Min[vec], Max[vec]}, PlotRange -> Max[Abs[ovec]] + 1], ListPlot[Puntos, PlotStyle -> {Red, PointSize[Medium]}]], If[bandera == 1, "Los pares ordenados no forman una función, presione Clear y vuelva a iniciar"]], If[obandera == 1, Puntos], If[Length[Puntos] >= 2, If[bandera == 1 && validos, Expand[InterpolatingPolynomial[Puntos, x]], "Debe ingresar al menos dos puntos"], Center], {{a, 0}}, {{b, 0}}, Delimiter, Button["Añadir par ordenado", bandera = 0; obandera = 1; validos = False; vec = Append[vec, a]; ovec = Append[ovec, b]; Puntos = Append[Puntos, {a, b}]], Button["Interpolación", bandera = 1; If[Length[DeleteDuplicates[vec]] == Length[vec], validos = True]], Delimiter, Button["Clear", bandera = 0; obandera = 0; Puntos = {}; vec = {}; ovec = {}; validos = False], Alignment -> Center, Initialization :-> (Puntos = {}; bandera = 0; obandera = 0; vec = {}; ovec = {}; validos = False)]`



15 al 30 de septiembre de 2015

+

a

b

Añadir par ordenado

Interpolación

Clear

Debe ingresar al menos dos puntos

6. Polinomial (analiza una función polinomial cualesquiera, mostrando su gráfica, intersecciones con los ejes y una tabla de valores):

```
Manipulate[If[NumberQ[Subscript[a, 5]] && NumberQ[Subscript[a, 4]]
&& NumberQ[Subscript[a, 3]] && NumberQ[Subscript[a, 2]] &&
NumberQ[Subscript[a, 1]] && NumberQ[Subscript[a, 0]], Column[{If[n,
Ejx = 10; Ejj = 10; Show[{Plot[Sum[Subscript[a, j] x^j, {j, 0, 5}], {x, -Ejx,
Ejx}, PlotRange -> Ejj], ListPlot[{{pt, Sum[Subscript[a, j] x^j, {j, 0, 5}] /.
x -> pt}}, PlotStyle -> {Red, PointSize[Medium]}]], If[NumberQ[Ejx] &&
NumberQ[Ejj], Show[{Plot[Sum[Subscript[a, j] x^j, {j, 0, 5}], {x, -Ejx,
Ejx}, PlotRange -> Ejj], ListPlot[{{pt, Sum[Subscript[a, j] x^j, {j, 0, 5}] /.
x -> pt}}, PlotStyle -> {Red, PointSize[Medium]}]], "Debe ingresar
valores en los campos de texto Eje x y Eje y"}], Tabla[m,
Sum[Subscript[a, j] x^j, {j, 0, 5}], Sum[Subscript[a, j] x^j, {j, 0, 5}],
If[interx && deci == False, Solve[Sum[Subscript[a, j] x^j, {j, 0, 5}] == 0,
x, Reals], If[interx, NSolve[Sum[Subscript[a, j] x^j, {j, 0, 5}] == 0, x,
Reals]]], If[intery, InY[Subscript[a, 0]]], Center], For[i = 0, i <= 5,
```

```

If[NumberQ[Subscript[a, i]] == False, Subscript[a, i] = 0]; i++],
Style["Coeficientes numéricos", Bold, Medium],
Panel[Column[{Row[{"!\(\(*SubscriptBox[\(a), \(\(5)\)\)=",
InputField[Dynamic[Subscript[a, 5]], Number]}],
Row[{"!\(\(*SubscriptBox[\(a), \(\(4)\)\)=",
InputField[Dynamic[Subscript[a, 4]], Number]}],
Row[{"!\(\(*SubscriptBox[\(a), \(\(3)\)\)=",
InputField[Dynamic[Subscript[a, 3]], Number]}],
Row[{"!\(\(*SubscriptBox[\(a), \(\(2)\)\)=",
InputField[Dynamic[Subscript[a, 2]], Number]}],
Row[{"!\(\(*SubscriptBox[\(a), \(\(1)\)\)=",
InputField[Dynamic[Subscript[a, 1]], Number]}],
Row[{"!\(\(*SubscriptBox[\(a), \(\(0)\)\)=",
InputField[Dynamic[Subscript[a, 0]], Number]}]}, Center]], Delimiter,
Style["Intervalos de graficación", Bold, Medium],
Panel[Column[{Row[{"Eje x=", InputField[Dynamic[Ejx], Number]}],
Row[{"Eje y", InputField[Dynamic[Ejy], Number]}]}, Center]], Delimiter,
{{n, True, "Ejes automático"}, {True, False}}, {{m, 1, "Tabla"}, 1, 21, 1},
Delimiter, {{interx, False, "Intersección (es) con el eje x"}, {True,
False}}, {{intery, False, "Intersección con el eje y"}, {True, False}},
{{deci, False, "Con decimales"}, {True, False}}, {{pt, 0, "Punto móvil"}, -
Ejx, Ejx, 0.1}, Alignment -> Center, Initialization -> (Subscript[a, 5] = 0;
Subscript[a, 4] = 0; Subscript[a, 3] = 0; Subscript[a, 2] = 0; Subscript[a,
1] = 0; Subscript[a, 0] = 0; Ejx = 10; Ejy = 10; Tabla[m_, Exp_] :=
Module[{}, TablaPre = Table[h, {h, -10, 10}]; TablaImg = Table[Exp /. x ->
h, {h, -10, 10}]; Grid[{Prepend[Take[TablaPre, m], "x"],
Prepend[Take[TablaImg, m], "y"]}], Frame -> All]; InY[v_] := Module[{},
StringJoin["(0,", ToString[v], ")"]]]]

```



15 al 30 de septiembre de 2015

Coefficientes numéricos

$a_5 =$

$a_4 =$

$a_3 =$

$a_2 =$

$a_1 =$

$a_0 =$

Intervalos de graficación

Eje x=

Eje y

Ejes automático

Tabla

Intersección (es) con el eje x

Intersección con el eje y

Con decimales

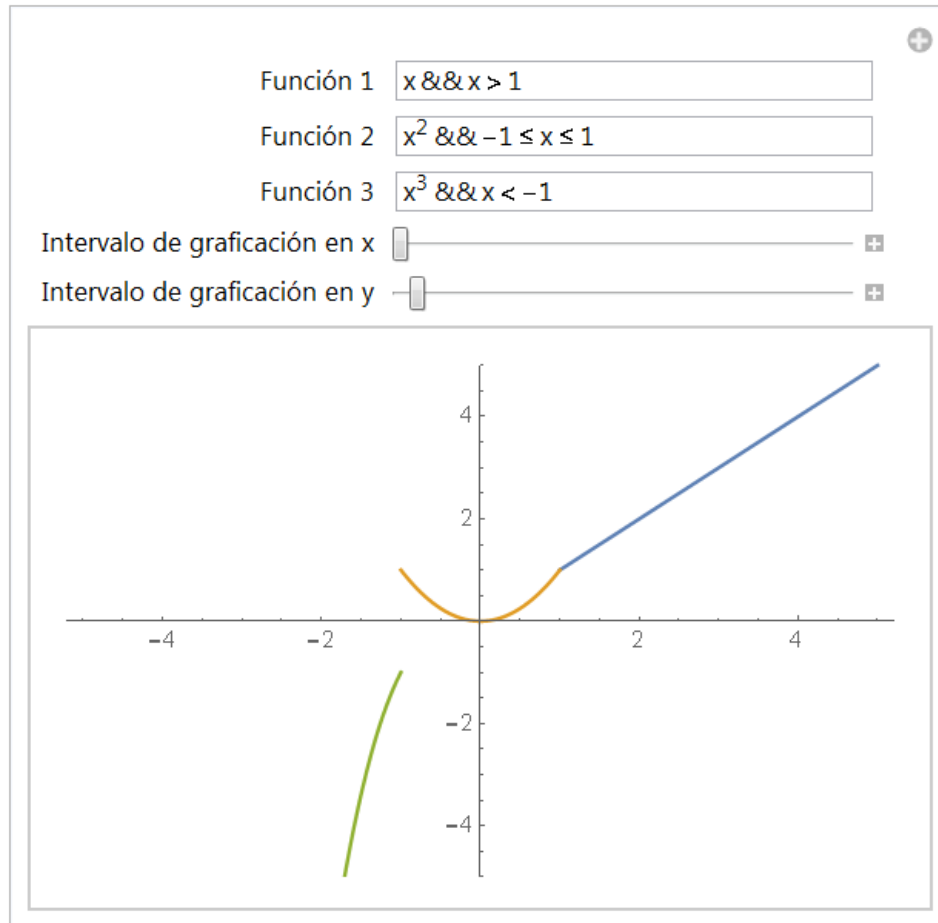
Punto móvil

| | |
|---|-----|
| x | -10 |
| y | 0 |

0

7. Graficador (genera un graficador, solo corre en *Wolfram Player Pro*):
- ```
Manipulate[Plot[{f, g, h}, {x, -a, a}, PlotRange -> b], {{f, x && x > 1, "Función 1"}}, {{g, x^2 && -1 <= x <= 1, "Función 2"}}, {{h, x^3 && x < -1, "Función 3"}}, {{a, 5, "Intervalo de graficación en x"}, 5, 100, 5}, {{b, 5, "Intervalo de graficación en y"}, 1, 100, 1}]
```

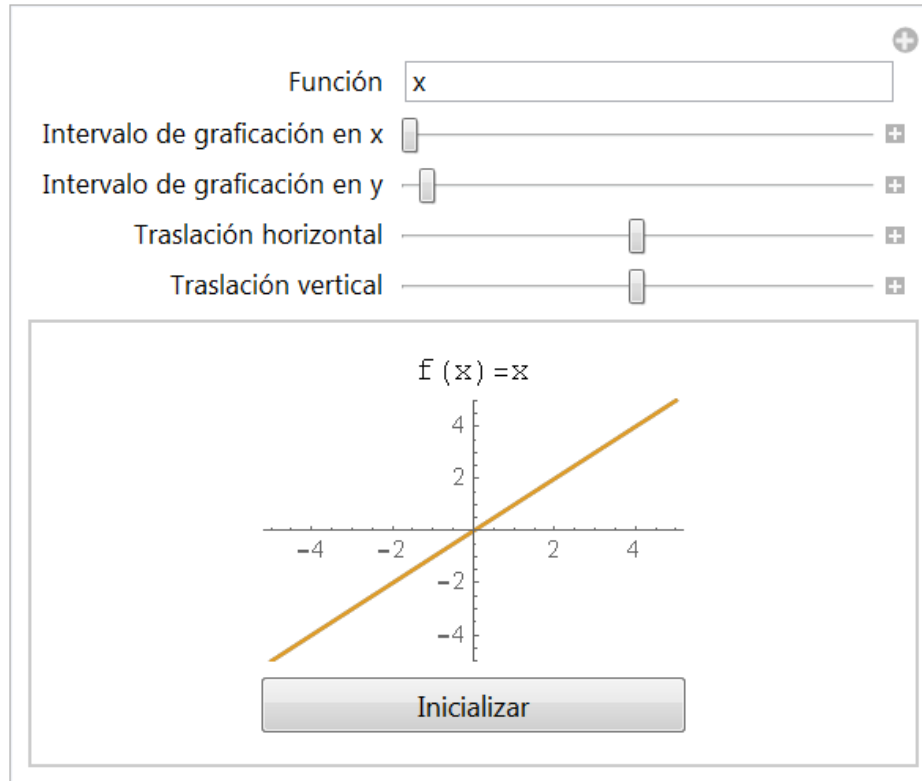




8. **Traslaciones verticales y horizontales (analiza el concepto de traslación vertical y horizontal para una función, solo corre en *Wolfram Player Pro*):** `Manipulate[Column[{Row[{"f(", x + th, ")="}, tv + f /. x -> x + th], Plot[{f, Evaluate[tv + f /. x -> x + th]}, {x, -a, a}, PlotRange -> b], Button["Inicializar", tv = 0; th = 0]}, Center], {{f, x, "Función"}}, {{a, 5, "Intervalo de graficación en x"}, 5, 100, 5}, {{b, 5, "Intervalo de graficación en y"}, 1, 100, 1}, {{th, 0, "Traslación horizontal"}, -100, 100, 1}, {{tv, 0, "Traslación vertical"}, -100, 100, 1}, Alignment -> Center]`



15 al 30 de septiembre de 2015



9. Monotonía (muestra como un intervalo y gráficamente, el crecimiento y decrecimiento de una función, solo corre en *Wolfram Player Pro*):
- ```

Manipulate[Column[{Show[Plot[f, {x, -a, a}, PlotRange -> b],
If[NumberQ[D[f, x]] == True && D[f, x] > 0, NumberLinePlot[Interval[{-Infinity, Infinity}], PlotStyle -> Red],
If[Total[StringCount[StringSplit[ToString[Reduce[D[f, x] > 0, x]]], "F"]] == 0 && Total[StringCount[StringSplit[ToString[Reduce[D[f, x] > 0, x]]], "\[Element]"]] == 0, NumberLinePlot[GenerandoIntervalos[Reduce[D[f, x] > 0, x]], PlotStyle -> Red],
If[Total[StringCount[StringSplit[ToString[Reduce[D[f, x] > 0, x]]], "\[Element]"]] != 0, NumberLinePlot[Interval[{-Infinity, Infinity}], PlotStyle -> Red], ListPlot[{{0, 0}}, PlotStyle -> {Transparent}]]],
If[NumberQ[D[f, x]] == True && D[f, x] < 0, NumberLinePlot[Interval[{-

```

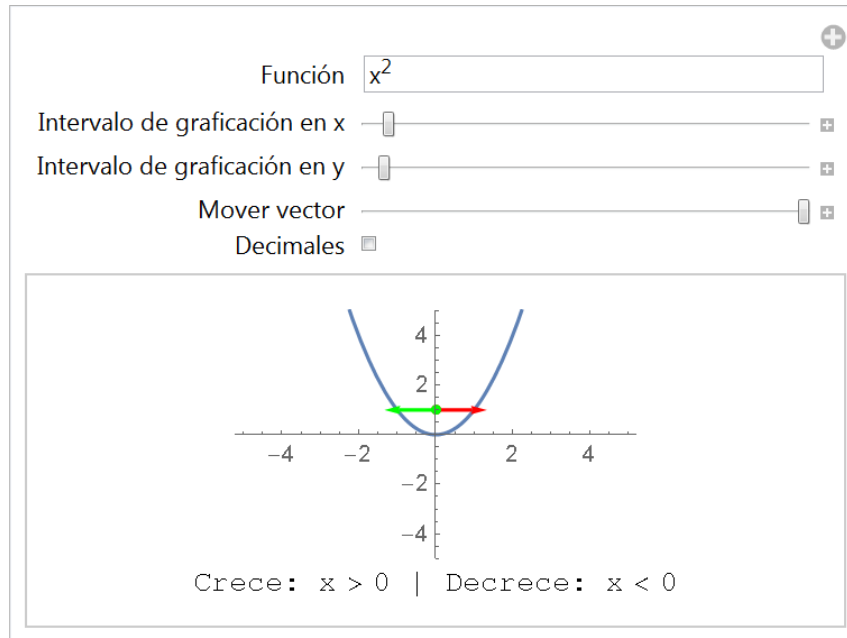
```

Infinity, Infinity}}, PlotStyle -> Green],
If[Total[StringCount[StringSplit[ToString[Reduce[D[f, x] < 0, x]], "F"]]
== 0 && Total[StringCount[StringSplit[ToString[Reduce[D[f, x] < 0, x]],
"\[Element]"]] == 0, NumberLinePlot[GenerandoIntervalos[Reduce[D[f,
x] < 0, x]], PlotStyle -> Green],
If[Total[StringCount[StringSplit[ToString[Reduce[D[f, x] < 0, x]],
"\[Element]"]] != 0, NumberLinePlot[Interval[{-Infinity, Infinity}],
PlotStyle -> Green], ListPlot[{{0, 0}}, PlotStyle -> {Transparent}]]],
Graphics[Arrow[{{c, Evaluate[f /. x -> c]}, {c + 2, Evaluate[f /. x -> c] + 2
D[f, x] /. x -> c}}]], Row[{"Crece: ", If[dec, N[Reduce[D[f, x] > 0, x]],
Reduce[D[f, x] > 0, x]], " | Decrece: ", If[dec, N[Reduce[D[f, x] < 0, x]],
Reduce[D[f, x] < 0, x]]}], Center], {{f, x^2, "Función"}, {{a, 5, "Intervalo
de graficación en x"}, 0.1, 100, 0.2}, {{b, 5, "Intervalo de graficación en
y"}, 1, 100, 1}, {{c, 5, "Mover vector"}, -a, a, 0.2}, {{dec, False,
"Decimales"}, {True, False}}, Alignment -> Center, Initialization :->
(GenerandoIntervalos[L_] := Module[{Lista = {}, Intervalo}, Intervalo[v_]
:= Module[{}, If[Total[StringCount[StringSplit[ToString[v]], "<"]] == 1,
Inter = {-Infinity, v[[2]]}; If[Total[StringCount[StringSplit[ToString[v]],
">"]] == 1, Inter = {v[[2], Infinity}];
If[Total[StringCount[StringSplit[ToString[v]], "<"]] == 2, Inter = {v[[1]],
v[[5]]}; Return[Inter]]; If[Total[StringCount[StringSplit[ToString[L],
"|"]] == 0, Lista = Interval[Intervalo[L]], Lista =
Interval[Intervalo[L[[1]]]]; For[i = 2, i <= Length[L], Lista =
IntervalUnion[Listas, Interval[Intervalo[L[[i]]]]; i++]; Listas]]

```



15 al 30 de septiembre de 2015



10. Recta puntos (grafica una recta dados dos puntos y la analiza de acuerdo a sus intersecciones y monotonía): `Manipulate[If[NumberQ[a] && NumberQ[b], Column[{If[Length[Puntos] == 2, P = Puntos[[1]]; A = Puntos[[2]] - Puntos[[1]]; Show[ParametricPlot[P + t A, {t, 0, 1}, AxesOrigin -> {0, 0}], ListPlot[Puntos, PlotStyle -> {Red, PointSize[Medium]}], "Debe ingresar dos puntos distintos"], Row[{"Puntos: ", Puntos}], Row[{"Pendiente: ", If[Length[Puntos] == 2, If[A[[1]] != 0, Simplify[A[[2]]/A[[1]], "no existe"}], Row[{"Intersección con el eje y: ", If[Length[Puntos] == 2, If[A[[1]] != 0, Row[{"(0,", Simplify[P[[2]] - A[[2]]/A[[1]] P[[1]], ")"}], If[P[[1]] == 0 && A[[1]] == 0, "eje y", If[A[[1]] == 0, "no hay"}]]], Row[{"Ecuación de la recta: ", If[Length[Puntos] == 2, If[A[[1]] != 0, Row[{"y = ", Simplify[A[[2]]/A[[1]] ToString[x] + P[[2]] - A[[2]]/A[[1]] P[[1]]}], Row[{"x = ", Simplify[P[[1]]}]]], Row[{"Intersección con el eje x: ", If[Length[Puntos] == 2, If[A[[1]] != 0, If[Puntos[[1, 2]] != Puntos[[2, 2]], Row[{"(", Simplify[-(P[[2]] - A[[2]]/A[[1]] P[[1])]/(A[[2]]/A[[1]]), ", ",`

```

"0)"}], If[Puntos[[1, 2]] == 0, "eje x", "no hay"]], Row[{"(", P[[1]], ",",
"0)"}]]], Row[{"Monotonía: ", If[Length[Puntos] == 2, If[A[[1]] != 0,
If[A[[2]]/A[[1]] > 0, "estrictamente creciente", If[A[[2]]/A[[1]] < 0,
"estrictamente decreciente", "constante"]], "no es función"]]}],
Center], a = 0; b = 0], {{a, 0}}, {{b, 0}}, Delimiter, Button["Añadir par
ordenado", If[Length[Puntos] < 2, Puntos = Append[Puntos, {a, b}];
Puntos = DeleteDuplicates[Puntos]]], Delimiter, Button["Clear", Puntos
= {}], Alignment -> Center, Initialization -> (Puntos = {})]

```

11. Recta pendiente (grafica una recta dado un punto y su pendiente y la analiza de acuerdo a sus intersecciones y monotonía):

```

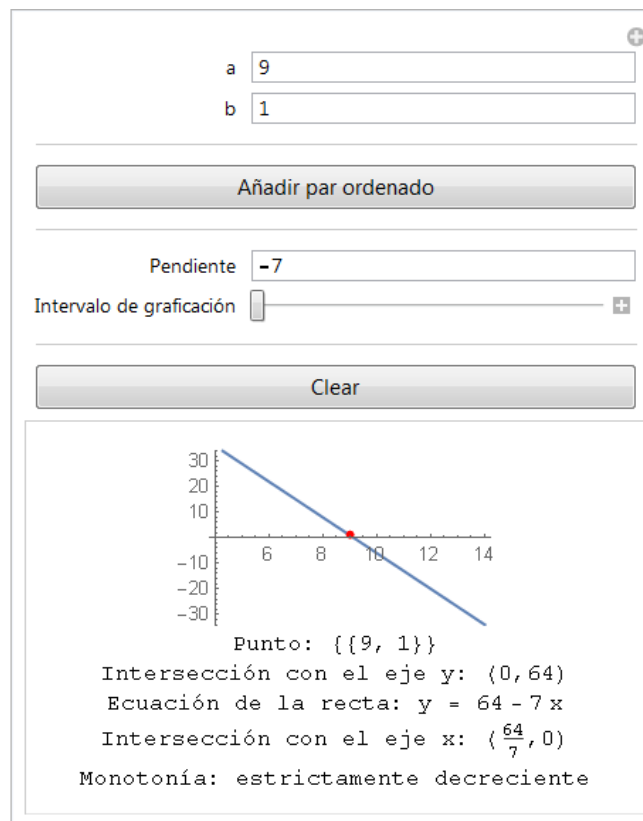
Manipulate[If[NumberQ[a] && NumberQ[b] && NumberQ[m],
Column[If[Length[Punto] == 1, P = Punto[[1]]; Show[Plot[m x + (P[[2]] -
m P[[1]]), {x, P[[1]] - g, P[[1]] + g}, PlotRange -> (m (P[[1]] + g) + (P[[2]] -
m P[[1]])], ListPlot[Punto, PlotStyle -> {Red, PointSize[Medium]}]],
"Debe ingresar un punto"], Row[{"Punto: ", Punto}],
Row[{"Intersección con el eje y: ", If[Length[Punto] == 1, Row[{"(0,",
P[[2]] - m P[[1]], ")"}]]], Row[{"Ecuación de la recta: ", If[Length[Punto]
== 1, Row[{"y = ", Simplify[m ToString[x] + P[[2]] - m P[[1]]}]]],
Row[{"Intersección con el eje x: ", If[Length[Punto] == 1 && m != 0,

```



15 al 30 de septiembre de 2015

```
Row[{"(", Simplify[-((P[[2]] - m P[[1]])/m)], ", ", "0)"}], If[m == 0 &&
Length[Punto] == 1, If[Punto[[1]] == {0, 0}, "eje x", "no hay"]]],
Row[{"Monotonía: ", If[Length[Punto] == 1, If[m > 0, "estrictamente
creciente", If[m < 0, "estrictamente decreciente", "constante"]]]}],
Center], a = 0; b = 0; m = 0], {{a, 0}}, {{b, 0}}, Delimiter, Button["Añadir
par ordenado", If[Length[Punto] < 1, Punto = Append[Punto, {a, b}];
Punto = DeleteDuplicates[Punto]]], Delimiter, {{m, 0, "Pendiente"}}, {{g,
5, "Intervalo de graficación"}, 5, 100, 5}, Delimiter, Button["Clear",
Punto = {}; m = 0], Alignment -> Center, Initialization :-> (Punto = {})]
```



12. Cuadrática (analiza una función cuadrática):

```
Manipulate[If[NumberQ[Subscript[a, 2]] && NumberQ[Subscript[a, 1]]
&& NumberQ[Subscript[a, 0]], Column[{If[n, E]jx = 10; E]jy = 10;
Show[{Plot[Sum[Subscript[a, j] x^j, {j, 0, 2}], {x, -E]jx, E]jx}, PlotRange ->
```

```

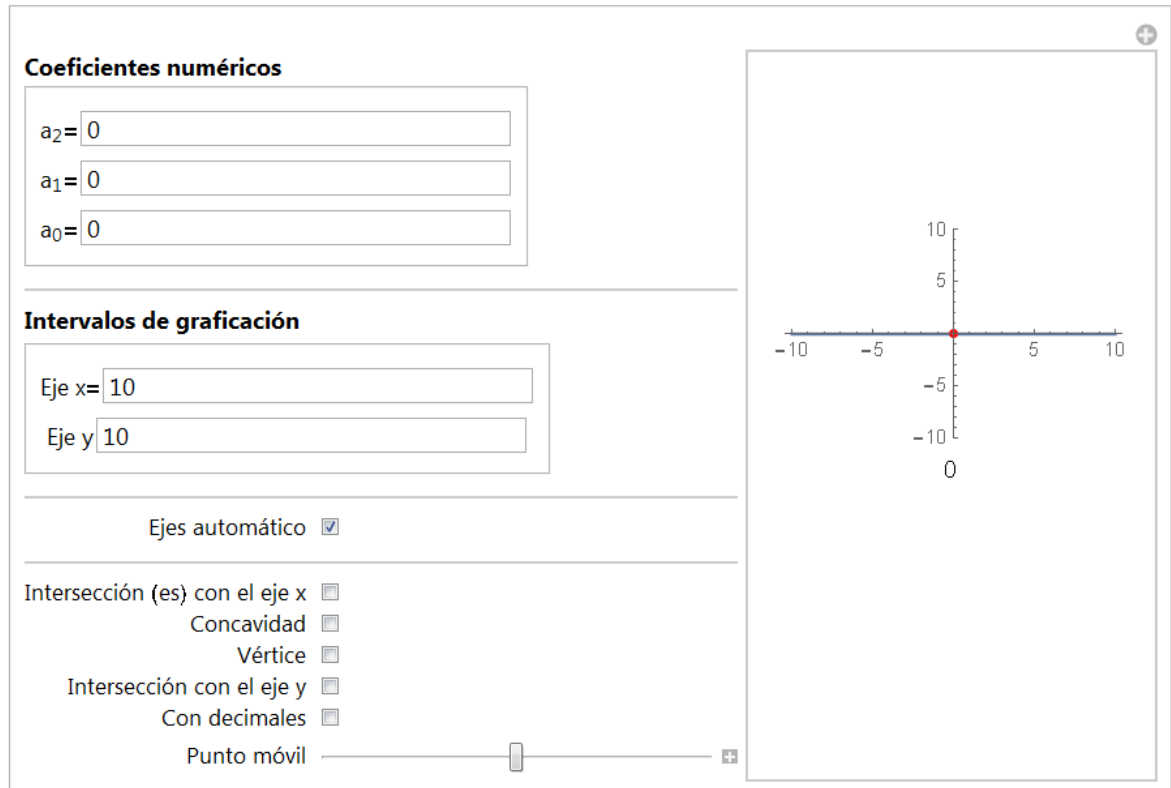
Ejy], ListPlot[{{pt, Sum[Subscript[a, j] x^j, {j, 0, 2}] /. x -> pt}}, PlotStyle
-> {Red, PointSize[Medium]}]], If[NumberQ[Ejx] && NumberQ[Ejy],
Show[{Plot[Sum[Subscript[a, j] x^j, {j, 0, 2}], {x, -Ejx, Ejx}, PlotRange ->
Ejy], ListPlot[{{pt, Sum[Subscript[a, j] x^j, {j, 0, 2}] /. x -> pt}}, PlotStyle
-> {Red, PointSize[Medium]}]], "Debe ingresar valores en los campos
de texto Eje x y Eje y"], Simplify[Sum[Subscript[a, j] x^j, {j, 0, 2}]],
If[interx && deci == False, Solve[Sum[Subscript[a, j] x^j, {j, 0, 2}] == 0,
x, Reals], If[interx, NSolve[Sum[Subscript[a, j] x^j, {j, 0, 2}] == 0, x,
Reals]], If[concavidad, If[Subscript[a, 2] > 0, "Cóncava hacia arriba",
If[Subscript[a, 2] < 0, "Cóncava hacia abajo", "No corresponde a una
función cuadrática"]]], If[vertice && Subscript[a, 2] != 0, Row[{"V = ("},
Simplify[-Subscript[a, 1]/(2 Subscript[a, 2])], ",", Simplify[-
(((Subscript[a, 1])^2 - 4 Subscript[a, 2] Subscript[a, 0])/(4 Subscript[a,
2]))], ")"}]], If[intery, InY[Subscript[a, 0]]], Center], For[i = 0, i <= 2,
If[NumberQ[Subscript[a, i]] == False, Subscript[a, i] = 0]; i++]],
Style["Coeficientes numéricos", Bold, Medium],
Panel[Column[{Row[{"\\(\\*SubscriptBox[\\(a), \\(2\\)]\\)="},
InputField[Dynamic[Subscript[a, 2]], Number]],
Row[{"\\(\\*SubscriptBox[\\(a), \\(1\\)]\\)="},
InputField[Dynamic[Subscript[a, 1]], Number]],
Row[{"\\(\\*SubscriptBox[\\(a), \\(0\\)]\\)="},
InputField[Dynamic[Subscript[a, 0]], Number]]], Center]], Delimiter,
Style["Intervalos de graficación", Bold, Medium],
Panel[Column[{Row[{"Eje x=", InputField[Dynamic[Ejx], Number]}],
Row[{"Eje y", InputField[Dynamic[Ejy], Number]}], Center]], Delimiter,
{{n, True, "Ejes automático"}, {True, False}}, Delimiter, {{interx, False,
"Intersección (es) con el eje x"}, {True, False}}, {{concavidad, False,
"Concavidad"}, {True, False}}, {{vertice, False, "Vértice"}, {True,
False}}, {{intery, False, "Intersección con el eje y"}, {True, False}},
{{deci, False, "Con decimales"}, {True, False}}, {{pt, 0, "Punto móvil"}, -
Ejx, Ejx, 0.1}, Alignment -> Center, Initialization :-> (Subscript[a, 2] = 0;

```



15 al 30 de septiembre de 2015

Subscript[a, 1] = 0; Subscript[a, 0] = 0; Ejsx = 10; Ejy = 10; InY[v_] :=
Module[{}, StringJoin["(0,", ToString[v], ")"]]]



13. Intersecciones (determina la intersección entre dos rectas, dos parábolas, o bien, una recta y una parábola, que correspondan a funciones): Manipulate[If[NumberQ[Subscript[a, 2]] && NumberQ[Subscript[a, 1]] && NumberQ[Subscript[a, 0]] && NumberQ[Subscript[b, 2]] && NumberQ[Subscript[b, 1]] && NumberQ[Subscript[b, 0]], Column[{If[na, Ejxa = 10; Ejya = 10; If[nb, Ejxb = 10; Ejyb = 10; Show[{Plot[{Sum[Subscript[a, j] x^j, {j, 0, 2}], Sum[Subscript[b, j] x^j, {j, 0, 2}], {x, -Ejxa, Ejxa}, PlotStyle -> {Blue, Orange}, PlotRange -> Ejya], ListPlot[{{pta, Sum[Subscript[a, j] x^j, {j, 0, 2}] /. x -> pta}}, PlotStyle -> {Red, PointSize[Medium]}], ListPlot[{{ptb, Sum[Subscript[b, j] x^j, {j, 0, 2}] /. x -> ptb}}, PlotStyle -> {Green,


```

PointSize[Medium]]}], If[NumberQ[Ejxb] && NumberQ[Ejyb],
Show[{Plot[Sum[Subscript[b, j] x^j, {j, 0, 2}], {x, -Max[{Ejxa, Ejxb}],
Max[{Ejxa, Ejxb}]}, PlotStyle -> {Orange}, PlotRange -> Max[Ejya,
Ejyb]], Plot[Sum[Subscript[a, j] x^j, {j, 0, 2}], {x, -Max[{Ejxa, Ejxb}],
Max[{Ejxa, Ejxb}]}, PlotStyle -> {Blue}, PlotRange -> Max[Ejya, Ejyb]],
ListPlot[{{pta, Sum[Subscript[a, j] x^j, {j, 0, 2}] /. x -> pta}}, PlotStyle ->
{Red, PointSize[Medium]}], ListPlot[{{ptb, Sum[Subscript[b, j] x^j, {j, 0,
2}] /. x -> ptb}}, PlotStyle -> {Green, PointSize[Medium]}]], "Debe
ingresar valores en los campos de texto Eje x y Eje y"], If[nb, Ejxb =
10; Ejyb = 10; If[NumberQ[Ejxa] && NumberQ[Ejya],
Show[{Plot[Sum[Subscript[a, j] x^j, {j, 0, 2}], {x, -Max[{Ejxa, Ejxb}],
Max[{Ejxa, Ejxb}]}, PlotStyle -> {Blue}, PlotRange -> Max[Ejya, Ejyb]],
Plot[Sum[Subscript[b, j] x^j, {j, 0, 2}], {x, -Max[{Ejxa, Ejxb}], Max[{Ejxa,
Ejxb}]}, PlotStyle -> {Orange}, PlotRange -> Max[Ejya, Ejyb]],
ListPlot[{{pta, Sum[Subscript[a, j] x^j, {j, 0, 2}] /. x -> pta}}, PlotStyle ->
{Red, PointSize[Medium]}], ListPlot[{{ptb, Sum[Subscript[b, j] x^j, {j, 0,
2}] /. x -> ptb}}, PlotStyle -> {Green, PointSize[Medium]}]], "Debe
ingresar valores en los campos de texto Eje x y Eje y",
If[NumberQ[Ejxa] && NumberQ[Ejya] && NumberQ[Ejxb] &&
NumberQ[Ejyb], Show[{Plot[Sum[Subscript[a, j] x^j, {j, 0, 2}], {x, -
Max[{Ejxa, Ejxb}], Max[{Ejxa, Ejxb}]}, PlotStyle -> {Blue}, PlotRange ->
Max[Ejya, Ejyb]], Plot[Sum[Subscript[b, j] x^j, {j, 0, 2}], {x, -Max[{Ejxa,
Ejxb}], Max[{Ejxa, Ejxb}]}, PlotStyle -> {Orange}, PlotRange ->
Max[Ejya, Ejyb]], ListPlot[{{pta, Sum[Subscript[a, j] x^j, {j, 0, 2}] /. x ->
pta}}, PlotStyle -> {Red, PointSize[Medium]}], ListPlot[{{ptb,
Sum[Subscript[b, j] x^j, {j, 0, 2}] /. x -> ptb}}, PlotStyle -> {Green,
PointSize[Medium]}]], "Debe ingresar valores en los campos de texto
Eje x y Eje
y"]], Row[{"f(x) = ", Simplify[Sum[Subscript[a, j] x^j, {j, 0, 2}]]}],
Row[{"g(x) = ", Simplify[Sum[Subscript[b, j] x^j, {j, 0, 2}]]}],
Row[{"Intersección(es): ", If[And[Or[Subscript[a, 2] != 0, Subscript[a, 1]

```



15 al 30 de septiembre de 2015

```

!= 0, Subscript[a, 0] != 0], Or[Subscript[b, 2] != 0, Subscript[b, 1] != 0,
Subscript[b, 0] != 0]], If[dec == False, Conjunto = {}; sol =
Solve[Sum[Subscript[a, j] x^j, {j, 0, 2}] == Sum[Subscript[b, j] x^j, {j, 0,
2}], x, Reals]; For[i = 1, i <= Length[sol], Conjunto = Append[Conjunto,
{x, Simplify[Sum[Subscript[a, j] x^j, {j, 0, 2}]]} /. sol[[i]]; i++];
If[Conjunto == {}, "no hay", Conjunto], Conjunto = {}; sol =
NSolve[Sum[Subscript[a, j] x^j, {j, 0, 2}] == Sum[Subscript[b, j] x^j, {j,
0, 2}], x, Reals]; For[i = 1, i <= Length[sol], Conjunto =
Append[Conjunto, FullSimplify[{x, Simplify[Sum[Subscript[a, j] x^j, {j,
0, 2}]]} /. sol[[i]]]; i++]; If[Conjunto == {}, "no hay", Conjunto]]]],
Center], For[i = 0, i <= 2, If[NumberQ[Subscript[a, i]] == False,
Subscript[a, i] = 0]; If[NumberQ[Subscript[b, i]] == False, Subscript[b, i]
= 0]; i++]], Style["Función f(x)", Bold, Medium], Delimiter,
Style["Coeficientes numéricos", Bold, Medium],
Panel[Column[{Row[{"\(\^*SubscriptBox[a], \{2\}\)=",
InputField[Dynamic[Subscript[a, 2]], Number]}],
Row[{"\(\^*SubscriptBox[a], \{1\}\)=",
InputField[Dynamic[Subscript[a, 1]], Number]}],
Row[{"\(\^*SubscriptBox[a], \{0\}\)=",
InputField[Dynamic[Subscript[a, 0]], Number]}]], Center]], Delimiter,
Style["Intervalos de graficación", Bold, Medium],
Panel[Column[{Row[{"Eje x=", InputField[Dynamic[Ejxa], Number]}],
Row[{"Eje y", InputField[Dynamic[Ejya], Number]}]], Center]],
Delimiter, {{na, True, "Ejes automático"}, {True, False}}, Delimiter,
{{pta, 0, "Punto móvil"}, -Ejxa, Ejxa, 0.1}, Delimiter, Style["Función
g(x)", Bold, Medium], Delimiter, Style["Coeficientes numéricos", Bold,
Medium], Panel[Column[{Row[{"\(\^*SubscriptBox[b], \{2\}\)=",
InputField[Dynamic[Subscript[b, 2]], Number]}],

```

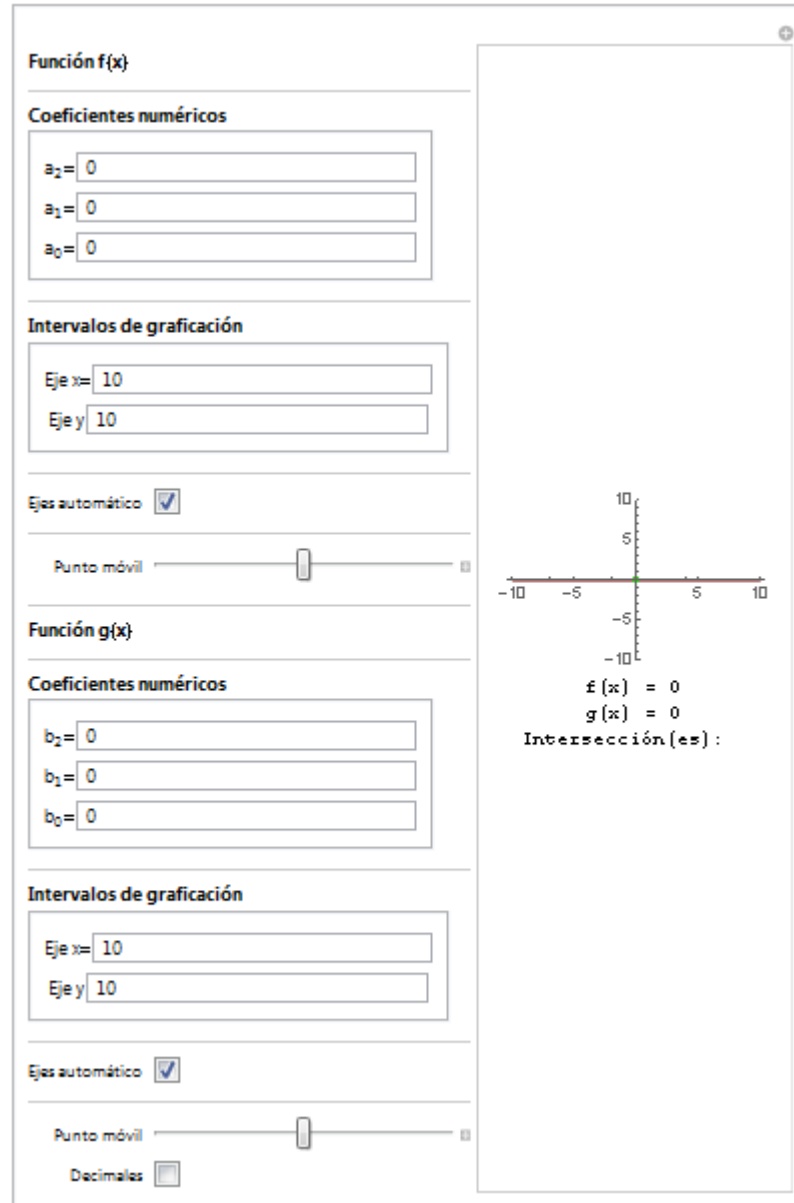
```

Row[{"!\(*SubscriptBox[(b), (1)]\)=",
InputField[Dynamic[Subscript[b, 1]], Number]],
Row[{"!\(*SubscriptBox[(b), (0)]\)=",
InputField[Dynamic[Subscript[b, 0]], Number]]}, Center]], Delimiter,
Style["Intervalos de graficación", Bold, Medium],
Panel[Column[{Row[{"Eje x=", InputField[Dynamic[Ejxb], Number]}],
Row[{"Eje y", InputField[Dynamic[Ejyb], Number]}]}, Center]],
Delimiter, {{nb, True, "Ejes automático"}, {True, False}}, Delimiter,
{{ptb, 0, "Punto móvil"}, -Ejxb, Ejxb, 0.1}, {{dec, False, "Decimales"},
{True, False}}, Alignment -> Center, Initialization :-> (Subscript[a, 2] =
0; Subscript[a, 1] = 0; Subscript[a, 0] = 0; Subscript[b, 2] = 0;
Subscript[b, 1] = 0; Subscript[b, 0] = 0; Ejxa = 10; Ejya = 10; Ejxb = 10;
Ejyb = 10)]

```



15 al 30 de septiembre de 2015



4.2.5 Compartir un CDF

Desde *Wolfram Cloud* es posible compartir un CDF. En *Mathematica* se ejecuta `CloudDeploy[Manipulate ...]`, lo anterior genera una dirección electrónica que carga el CDF en la nube. Esta dirección puede ser guardada en código QR para el uso de los estudiantes. Por ejemplo, este es el código QR de **Tabla**:



Los códigos *QR* pueden ser generados desde *Mathematica*, en los archivos del taller, la aplicación *Code.nb* automatiza este proceso.

5. Conclusiones

El presente trabajo representa un esfuerzo de implementación de una tecnología contemporánea promovida por la compañía *Wolfram Research*. La creación de *CDF's* constituye un recurso didáctico de carácter interactivo que puede favorecer la enseñanza y el aprendizaje de la teoría de funciones y en general, de la educación matemática en cualquier área. Se espera que el presente documento contribuya con la formación del profesorado en esta disciplina y promueva estrategias de enseñanza más innovadoras en un ámbito nacional y regional.

6. Referencias

- Vílchez, E. y Ávila J. (2012). *Matemática elemental apoyada con software*. España: Editorial Académica Española.
- Vílchez, E. (2015). *Estructuras discretas con Mathematica*. México: Editorial Alfaomega.
- Vílchez, E. (2012). *Álgebra lineal apoyada con Mathematica*. Costa Rica: Editorial Tecnológica.
- Wolfram *Mathematica* 10: Documentation Center *Mathematica* functions and tutorials. Obtenido el 1 de julio del 2015: <http://reference.wolfram.com/language/>.

Currículum:



15 al 30 de septiembre de 2015

Master en Tecnología e Informática Educativa, Licenciado en la enseñanza de la matemática, docente e investigador de la Escuela de Informática de la Universidad Nacional de Costa Rica.

