

CLASE 10 ALGORITMOS

Una de las excelentes herramientas del Proyecto Descartes es la posibilidad de usar algoritmos. Si bien se trata de algoritmos sencillos, posibilita incluir cálculos muy útiles para el diseño de algunas escenas. En esta clase usaremos esta herramienta para el cálculo de sumatorias, entre ellas las integrales... las sumatorias por excelencia.

Ciclos DO-WHILE

Antes de realizar nuestro primer ejercicio de sumatorias, conviene comprender el funcionamiento lógico de los llamados ciclos DO-WHILE o HACER MIENTRAS.

La estructura lógica del ciclo es la siguiente:

```
DO WHILE Expresión
    Proposición 1
    Proposición 2
    Proposición n
END DO
```

Las proposiciones que están dentro del ciclo se ejecutarán mientras la **Expresión** sea verdadera. Veamos un ejemplo con una sumatoria sencilla:

$$\sum_{i=0}^5 i$$

Para el caso de las sumatorias, es necesario asignarle valores iniciales a las variables a utilizar. En este caso tendremos dos variables (i y S), las cuales se deben iniciar (o inicializar) en cero. Vemos como sería el algoritmo:

```
S=0
i=0
HACER MIENTRAS i ≤ 5
    S = S + i
    i = i + 1
FIN MIENTRAS
Muestre S
```

Si seguimos la secuencia lógica del algoritmo, las instrucciones que se ejecutarían serían las siguientes:

Ciclo	0	1	2	3	4	5
S	0	1	3	6	10	15
i	0	1	2	3	4	5

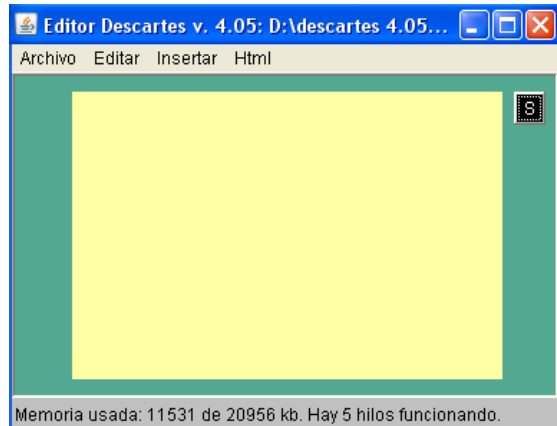
Es decir, mientras que $i \leq 5$ se ejecutan 5 ciclos.

Actividad 1. Diseñar una escena que calcule e imprima sumatorias de la forma:

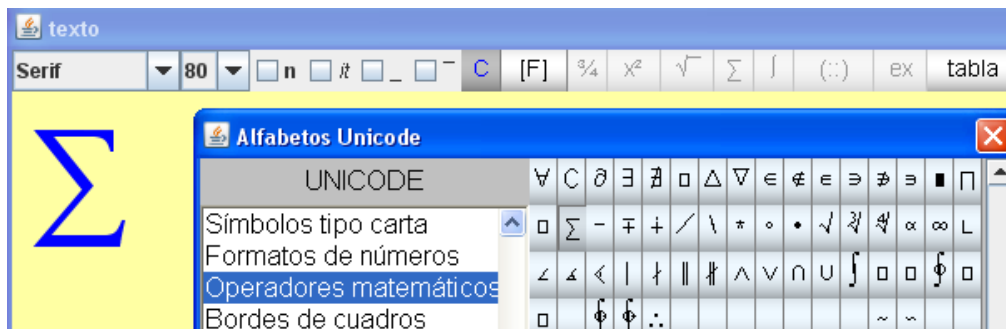
$$\sum_{i=a}^b i$$

1.1 Crea un nuevo archivo con el nombre clase 10a. Agrega una escena Descartes4 con su espacio 2D con un fondo de color claro.

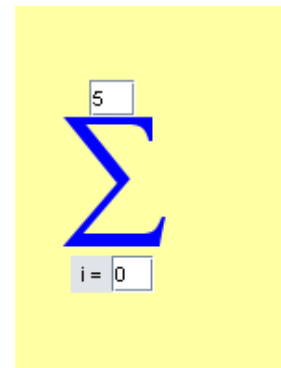
1.2 Desde la opción controles cambia las dimensiones del *applet* a 300x200 y haz clic en aceptar. Algunas escenas, por su tamaño, nos obligan a reducir considerablemente las dimensiones del *applet*, esta actividad es un ejemplo de ello.



Añade, desde la opción gráficos, un texto que sólo tendrá el símbolo de sumatoria. Para ello, haz clic en el botón texto y luego en el botón tabla. Elige **Operadores matemáticos** y luego el símbolo Σ . A este símbolo asigne un tamaño de 80 y un color azul. Finalmente, ubícalo en la posición (20, 50).



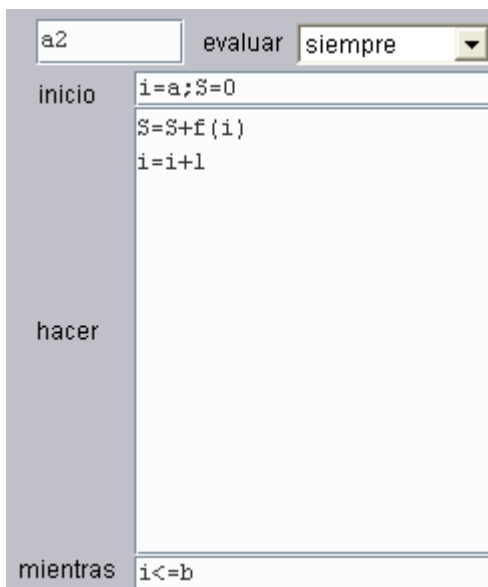
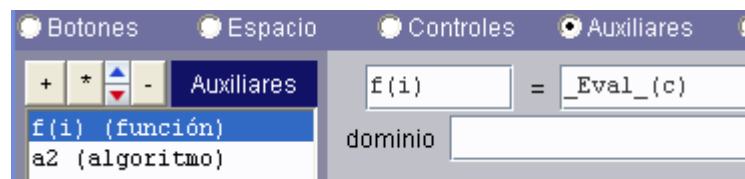
1.3 Los límites de la sumatoria (a y b) los agregaremos desde la opción controles. Para el control **a**, añadimos un control numérico tipo **campo de texto** al interior de la escena con estos atributos: id = a, nombre = "i =", pos = (30, 125, 45, 20), valor = 0, incr = 1. Para el control **b**, apodemos asignarle un valor de 0 inicial y un incremento de 1. Para el control **b**, añadimos un control numérico tipo **campo de texto** al interior de la escena con estos atributos: id = b, nombre = "" (en blanco), pos = (40,40,25,20) , valor = 5, incr = 1. Al hacer clic, observará una escena como la de la imagen de la derecha.



1.4 La expresión de la sumatoria la crearemos con un control similar a los anteriores, pero que tiene una característica especial de Descartes. Añade un control numérico tipo campo de texto con id = c, sin nombre y valor 'i' (la expresión va entre comillas simples, observa la imagen de abajo). En la casilla **pos** escribiremos (90, 75, 80, 30), el nacho de 80 parece exagerado para la expresión 'i'; sin embargo, hemos asignado este valor para permitir cambiar la expresión por otra más extensa... He aquí la característica especial, que comprenderás mejor en los siguientes pasos.



1.5 En la opción **Auxiliares** crearemos una función y el algoritmo de la sumatoria. La función captura la expresión que hemos añadido en el paso anterior; para ello, añade la función con nombre f(i) e igual a `_Eval_(c)`.



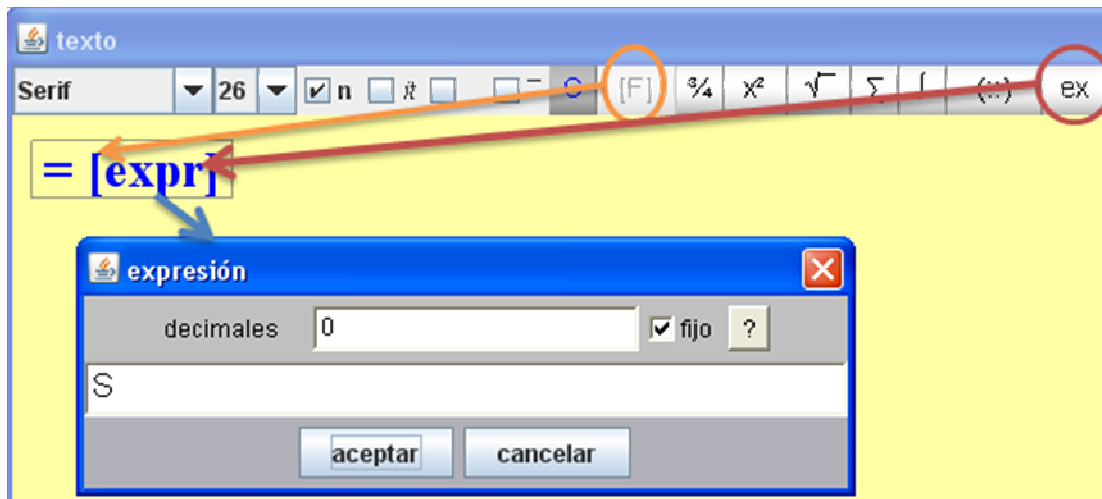
Por otra parte, el ciclo DO-WHILE puede codificarse en Descartes usando el auxiliar **algoritmo** en la opción **Auxiliares**. Añade esta auxiliar y escribe los datos en las casillas, tal como parecen en la imagen de la izquierda.

Las dos primeras instrucciones de inicialización de variables del ejemplo inicial se pueden escribir en la casilla de **inicio** separadas por “;”. Dado que el límite inferior es **a** y no cero como en el ejemplo, inicializamos la variable **i** en dicho límite inferior. Las proposiciones del cuerpo del ciclo (**hacer**) las escribimos tal como lo hicimos en el ejemplo. La expresión lógica del ciclo se escribe en la casilla **mientras**.

Analiza el algoritmo haciendo una prueba como lo hicimos al inicio de esta clase.

1.6 Insertaremos un texto para que imprima el resultado. Veamos como: añade un texto desde la opción gráficos de tamaño 26, tipo de letra Serif, luego haz clic en el botón **texto**. Escribe el símbolo = y después el valor de la sumatoria (S), que escribiremos a través de los siguientes pasos: haz clic en la opción F (ver imagen de abajo), lo cual hará que

aparezca un recuadro; luego haz clic en la opción **ex**, que hará aparecer la expresión “**expr**”; haz doble clic en esta expresión para ingresar la variable S con cero decimales. Finalmente haz clic en los botones aceptar y ubica el texto en la posición (170, 75).

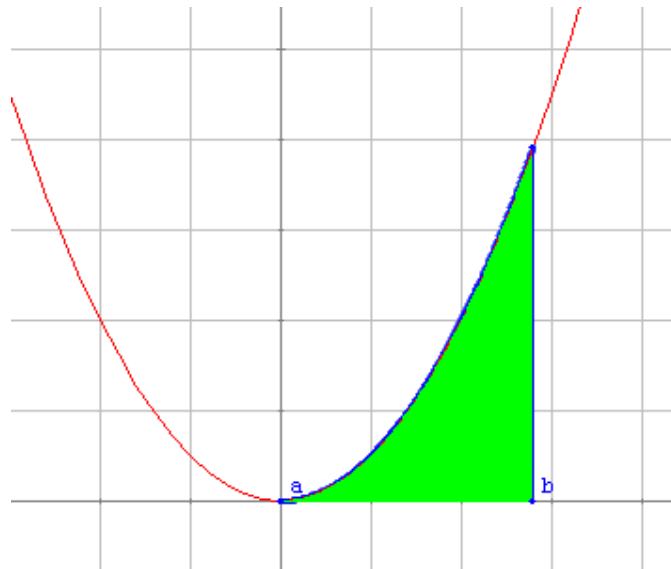


La característica especial del control **c** nos permite cambiar la expresión para obtener diferentes resultados, tal como se observa en la siguiente imagen.

$\sum_{i=0}^5 i = 15$	$\sum_{i=2}^{10} 3i-5 = 117$
$\sum_{i=1}^3 i^2 = 14$	$\sum_{i=1}^6 3i^3-i = 1302$

Actividad 2. Diseñar una escena que muestre la superficie bajo una curva y calcule su área.

Para esta actividad necesitamos de algunos conceptos previos. El primero lo trabajamos en la actividad 1 referido a los algoritmos para calcular sumatorias. El segundo es el concepto de integral definida a partir del cálculo del área bajo una curva $y=f(x)$ entre a y b , tal como lo indica la siguiente figura:



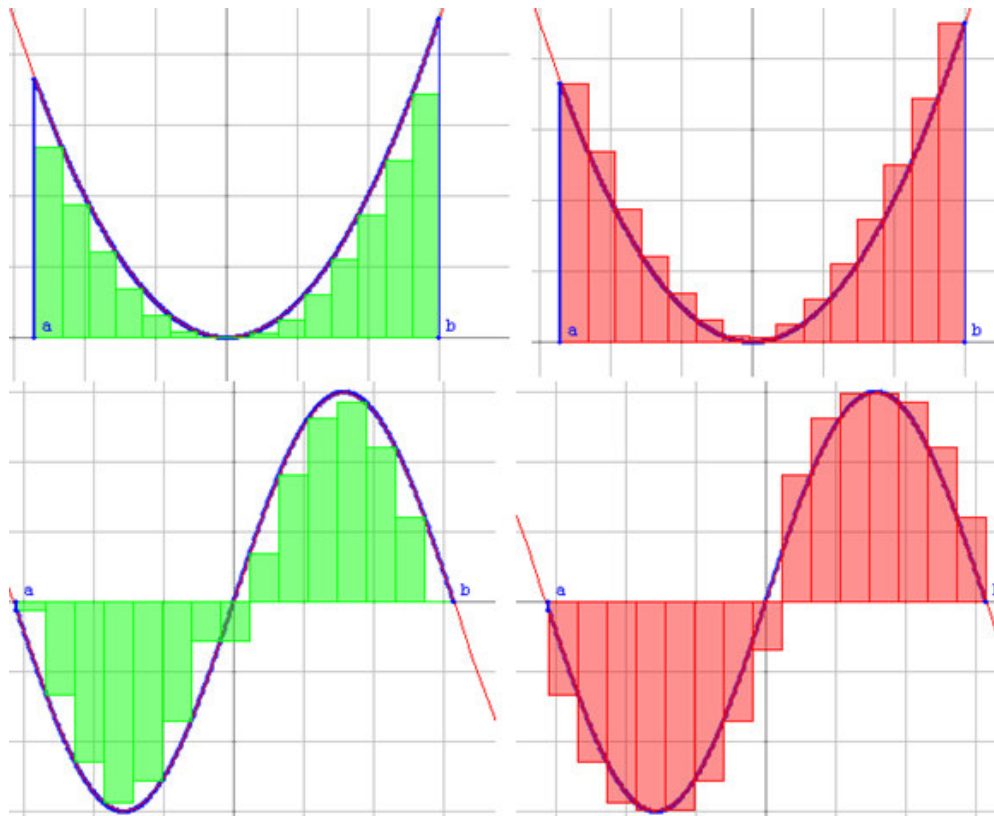
El cálculo del Área lo facilitó el matemático alemán Georg Friedrich Bernhard Riemann con su conocida integral o sumatoria de Riemann. Nuestro objetivo no es explicar este concepto de cálculo integral, sino aplicarlo en esta actividad. Sin embargo, si deseas repasar el concepto, puedes ir a este blog: <http://integrandoconpaco2.blogspot.com/2007/08/integral-definida-sesin-2.html>.

Hecho esto podemos usar la siguiente expresión para calcular el área acotada por $f(x)$, $x = a$, $x = b$ y el eje x .

$$A = \lim_{n \rightarrow \infty} \sum_{i=1}^n \frac{(b-a)}{n} f(x_i)$$

Como observas se trata de una sumatoria muy grande. Es decir, de muchos ciclos. El Nippe Descartes sólo acepta un máximo de 100.000 ciclos, mucho más pequeño que infinito. Pero, lo de infinito es un problema entre filosófico y matemático que para las integrales que vamos a calcular es irrelevante. Con un $n = 10.000$ es más que suficiente para ello.

Calcular la sumatoria anterior no es problema con la herramienta que nos suministra Descartes... los algoritmos. Pero no todo es color de rosa, en el diseño de esta actividad se nos presentarán varios problemas. Para comprenderlos observemos las siguientes imágenes:



Estas imágenes representan dos opciones de la sumatoria de Riemann: suma de rectángulos inferiores y suma de rectángulos superiores... primer problema, pero más de decisión que de análisis. Por otra parte, debemos garantizar que sumamos rectángulos inferiores o superiores cuando hay cambio en el sentido de la curva (primeras dos imágenes)... segundo problema, éste sí es de análisis. Igualmente debemos garantizar lo mismo cuando hay cambio de signo en la función (dos últimas imágenes)... tercer problema.

Vamos, entonces, a resolver nuestros problemas de cálculo.

2.1 Crea un archivo nuevo con el nombre clase 10b y una escena Descartes4 con su espacio 2D por defecto. Cambia las dimensiones del *applet* a 600x400, fija el espacio y desactiva los botones.

2.2 Agrega los siguientes controles:

Controles numérico tipo pulsador para las variables **a** y **b** (los límites izquierdo y derecho de la superficie), con los siguientes valores iniciales: $a = 0$ y $b = 3$.

Control numérico tipo pulsador **n** con valor inicial de 5, incremento 5, **mínimo = 5**, y cero decimales. Hago énfasis en la última característica de **n**, dado que de no limitarlo por debajo podemos incurrir en una división por cero (ver sumatoria de Riemann). Por dicho error supimos lo de los 100.000 ciclos permitidos.

Control numérico tipo pulsador **Superior** con valor inicial de 0, incremento 1, mínimo = 0, máximo = 1, y cero decimales. Es un selector que usaremos para cambiar de tipo de rectángulos.

2.3 Auxiliares. Con cinco auxiliares resolveremos nuestros problemas de cálculo.

F: Es una variable cuyo contenido es nuestra función inicial. Para esta actividad hemos escogido $0.5x^2$. Activamos la opción **editable** de tal forma que el usuario pueda ingresar otra función.

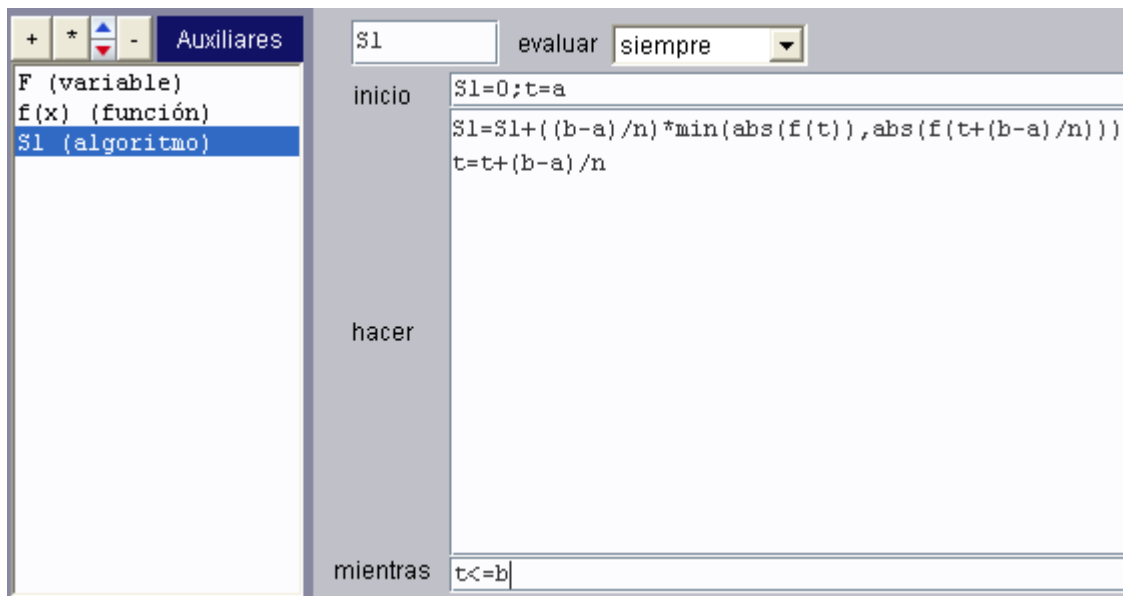


f(x): Es nuestra función a emplear en el resto de la actividad. Es uno de los trucos que suministra el Nippe Descartes, al permitir usar F como variable y asignarle a f(x) su contenido. Es decir, $f(x) = F$.



Ahora vienen nuestras tres auxiliares estrella: S1 para calcular la sumatoria de triángulos interiores, S2 para calcular la sumatoria de triángulos superiores y S para calcular la integral con una sumatoria que va desde cero hasta infinito... bueno... hasta 10.000.

S1: Aquí resolvemos el primer problema. Hemos elegido rectángulos por debajo. Te queda como tarea los rectángulos superiores. Observa la siguiente imagen y la explicación a continuación:



Los datos de **inicio** y de **mientras** son iguales a los de la primera actividad. Así que ya debes comprender el porqué. Sólo hemos cambiado la variable i por la variable t.

Lo interesante es analizar las expresiones del **hacer**:

$$S1=S1+((b-a)/n)*\min(\text{abs}(f(t)),\text{abs}(f(t+(b-a)/n)))$$
$$t=t+(b-a)/n$$

Para ello observa la figura de la derecha. El número de n rectángulos iguales tiene como base $\Delta x = (b-a)/n$.

Cada rectángulo tiene como altura $f(t)$. La suma del área de todos estos rectángulos es entonces:

$$S1 = S1 + (b-a)/n * f(t)$$

Pero, ¿qué pasa cuando hay cambio de sentido de la curva? Debemos garantizar que se tome el valor mínimo entre una ordenada y otra. Es decir, entre $f(t)$ y $f(t+\Delta x)$ o mejor, entre $f(t)$ y $f(t+(b-a)/n)$. Eso se logra con la función **min(x,y)** (también existe la función **max(x,y)**). Entonces queda resuelto nuestro segundo problema:

$$S1 = S1 + ((b-a)/n) * \min(f(t), f(t+(b-a)/n))$$

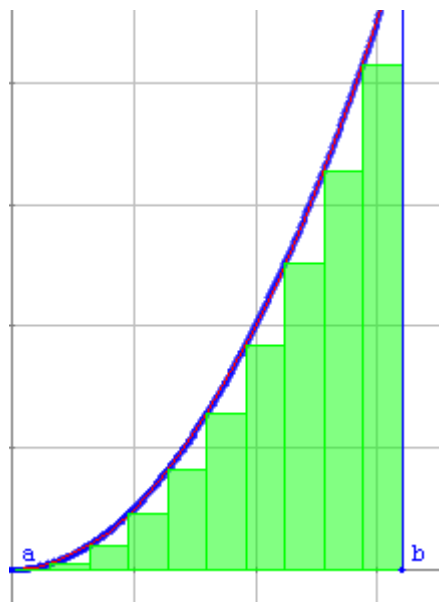
Viene el “tercer pero”... Pero, ¿qué pasa cuando hay cambio de signo?

Debemos garantizar que no se obtengan áreas negativas o erróneas. Eso se logra con la función valor absoluto **abs(x)**. Finalmente obtenemos:

$$S1 = S1 + ((b-a)/n) * \min(\text{abs}(f(t)), \text{abs}(f(t+(b-a)/n)))$$

S: Esta variable auxiliar nos calculará el valor de la integral. Observa la figura y analiza sus datos:

2.4 resultados. Agrega algunos textos de tal forma que la escena nos muestre los resultados obtenidos. Algo así como lo de la imagen de la derecha. El efecto del título se logra creando dos textos desplazados un pixel; es decir, un primer texto con el título INTEGRAL DEFINIDA de color negro en la posición (5, 10) y otro texto igual de color rojo en la posición (4,9).



INTEGRAL DEFINIDA

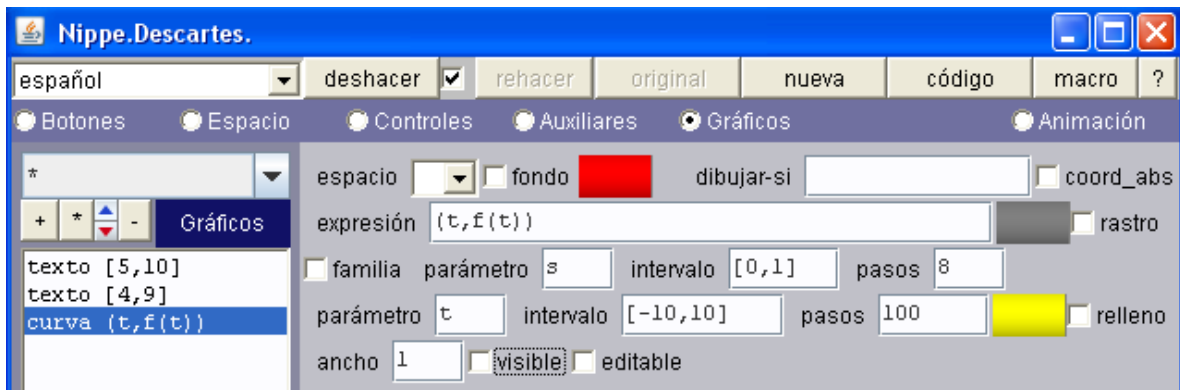
Suma superior = 9,83

Suma inferior = 5,94

Integral = 4,50

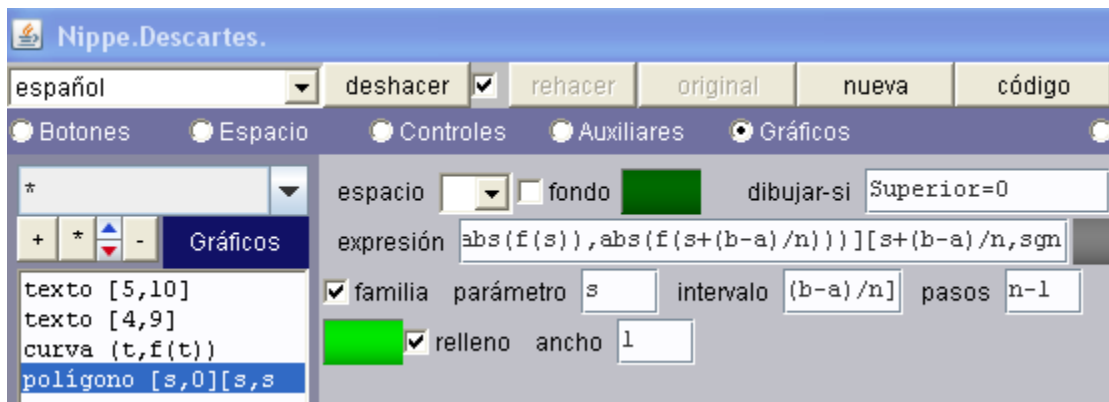
2.5 Inserción de gráficos. Vamos ahora a ver cómo graficamos la función y los rectángulos.

Agregamos inicialmente una curva (opción gráficos) que represente la función. Ver imagen siguiente:



La curva se define con la expresión $(t, f(t))$. El parámetro t lo hemos delimitado en el intervalo $[-10, 10]$ y para 100 pasos. Igualmente podemos definir la curva $(t, f(t))$ con otro color y en el intervalo $[a, b]$. Esto permitirá representar la porción de curva bajo la cual se dibujará la superficie.

Ahora dibujaremos nuestros rectángulos inferiores. Para ello agregamos el gráfico llamado **polígono** con los siguientes atributos: el borde del polígono tiene un color verde oscuro y un relleno de color verde, se dibuja sí el control Superior es cero. Como se trata de dibujar n polígonos, usamos la opción familia utilizando el parámetro s que varía en el intervalo $[a, b-(b-a)/n]$. Finalmente, asignamos una expresión que permitirá dibujar lo n rectángulos.

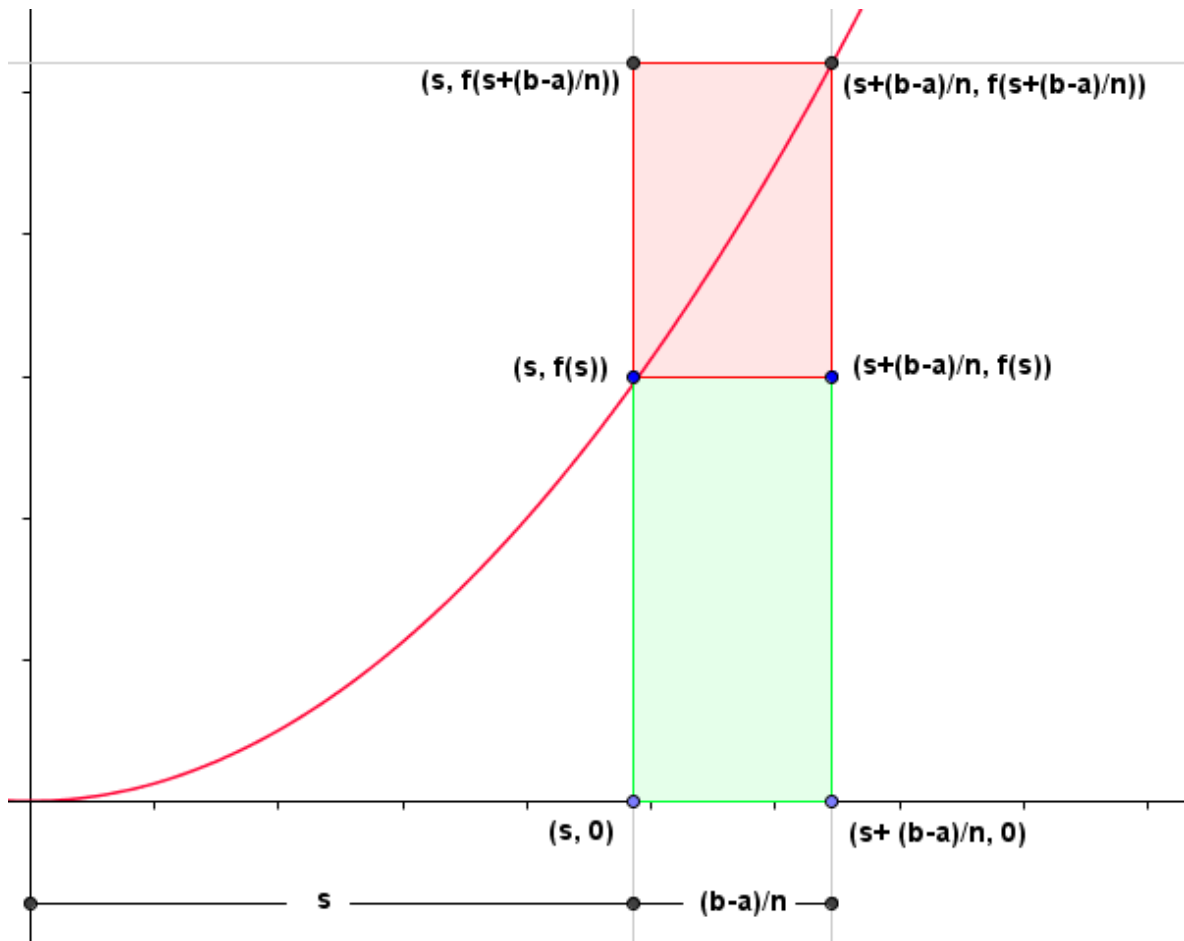


La expresión que vamos a usar es la siguiente:

$$[s,0][s,\text{sgn}(f(s))\cdot\min(\text{abs}(f(s)),\text{abs}(f(s+(b-a)/n)))] [s+(b-a)/n,\text{sgn}(f(s))\cdot\min(\text{abs}(f(s)),\text{abs}(f(s+(b-a)/n)))] [s+(b-a)/n,0][s,0]$$

Algo complicada ¿no?

Con la siguiente figura entenderás porqué dicha expresión



Para conformar nuestro rectángulo interior (color verde) necesitamos cinco puntos (el primero y el último son los mismos):

- P1 [s, 0]
- P2 [s, f(s)]
- P3 [s+(b-a)/n, 0]
- P4 [s+(b-a)/n, f(s)]
- P5 [s,0]

Se usan corchetes en lugar de paréntesis cuando usamos variables en los puntos

Con la gráfica anterior esto funciona perfectamente. Vuelve el pero... Pero, ¿qué pasa cuando hay cambio de sentido de la curva?

Lo que sucede es que se nos convierte en un rectángulo superior (color rojo). Por ello debemos garantizar que los puntos P2 y P3, efectivamente sean del rectángulo inferior. En ese sentido, los cambiaremos por:

- P2 [s, min(abs(f(s)),abs(f(s+(b-a)/n)))]
- P3 [s+(b-a)/n, min(abs(f(s)),abs(f(s+(b-a)/n)))]

Con la función **min(x, y)** garantizamos que tome la ordenada menor para estos puntos y con la función **abs(x)** permitimos que se comparen los valores absolutos.

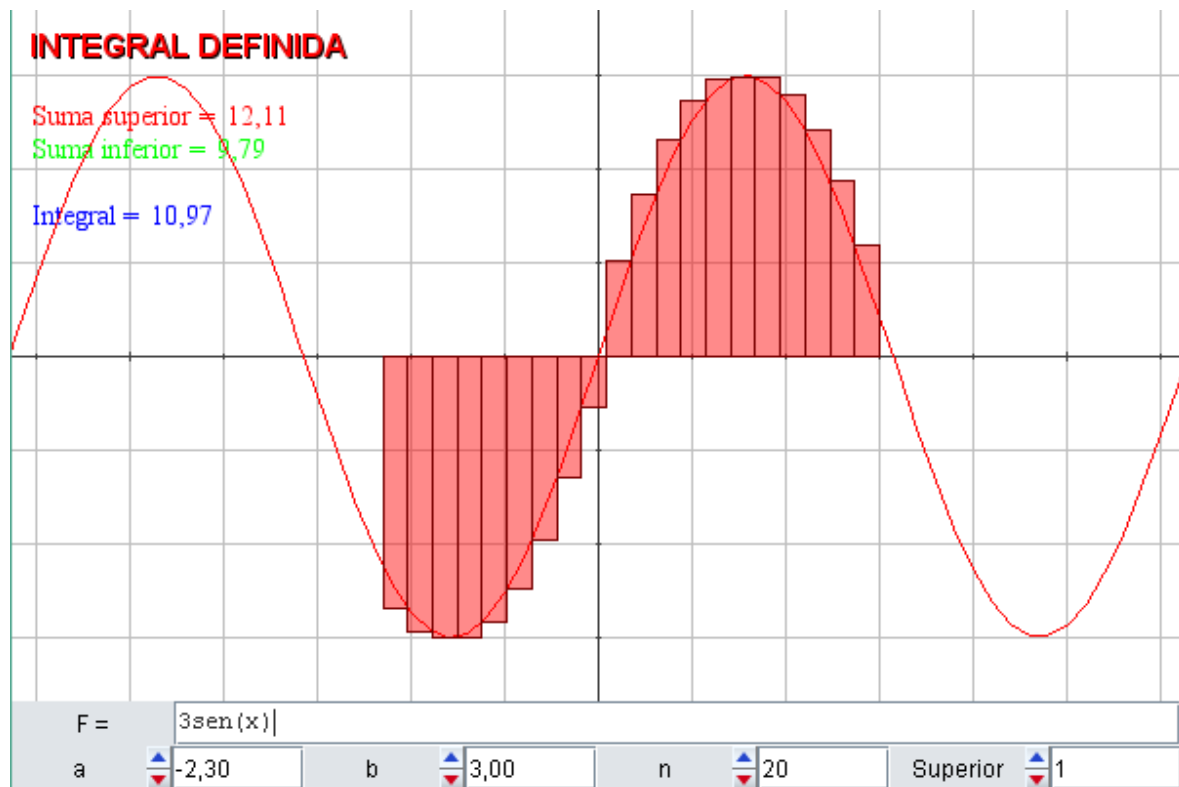
El último pero... Pero, ¿qué pasa cuando hay cambio de signo?

Que todos los rectángulos van a quedar por encima del eje x. Es aquí donde usamos la función **sgn(x)** que nos garantiza la conservación del signo de la ordenada. Finalmente nuestros cinco puntos serían:

- P1 [s, 0]
- P2 [s, sgn(f(s))*min(abs(f(s)),abs(f(s+(b-a)/n)))]
- P3 [s+(b-a)/n, sgn(f(s))*min(abs(f(s)),abs(f(s+(b-a)/n)))]
- P4 [s+(b-a)/n, f(s)]
- P5 [s, 0]

Terminamos de configurar nuestra **familia** de rectángulos definiendo el intervalo [a, b-(b-a)/n], activando la casilla **familia**, asignando el color al rectángulo y, en **dibujar-si** escribiendo la expresión Superior=0.

Bueno, te queda como tarea los rectángulos superiores; para ello, te recomendamos usar un color con transparencia, como se observa en la siguiente imagen.



Hasta pronto. En www.descartes3d.blogspot podrás practicar con el *applet* final

Juan Guillermo Rivera Berrío