

Ejemplo de @DESMEDIA

En la tabla siguiente se enumeran las ventas de casas efectuadas por una inmobiliaria en el mes de abril. Las edades de las mismas (expresadas en años) se encuentran en el rango denominado EDADES (C2..C8). Se desea hallar la desviación media de las edades de las casas de la tabla:

@DESMEDIA(EDADES) = 15,18

A	A	B	C	D
1	DIRECCION	m ²	EDAD	PRECIO
2	Castilla, 149	210	48	290.000\$
3	Ibiza, 80	200	22	105.000\$
4	Galileo, 88	110	21	135.000\$
5	Orense, 24	240	70	128.000\$
6	Hoyos, 52	95	52	174.000\$
7	Isaac Peral, 8	80	42	195.000\$
8	Fuencarral, 56	90	23	118.000\$

Ejemplo de @CORREL

Se desea averiguar si existe alguna relación entre la altura y el peso de diez individuos seleccionados al azar. Los datos se encuentran en una hoja de trabajo.

@CORREL(A2..A11;B2..B11) = 0,374888

A	A	B
1	ALTURA (cm)	PESO (kg)
2	191	73
3	188	86
4	175	68
5	175	76
6	180	77
7	180	73
8	188	75
9	173	68
10	178	71
11	179	87

Ejemplo de @COVAR

Se desea averiguar si existe alguna relación entre la altura y el peso de diez individuos seleccionados al azar. Los datos se encuentran en una hoja de trabajo.

@COVAR(A2..A11;B2..B11) = 13,82

A	A	B
1	ALTURA (cm)	PESO (kg)
2	191	73
3	188	86
4	175	68
5	175	76
6	180	77
7	180	73
8	188	75
9	173	68
10	178	71
11	179	87

Ejemplo de @@

En este sencillo cuadro de comisiones de ventas, la celda A10 contiene la fórmula

`@SI(C3="G";"C7";@SI(C3="S";"C8";@ERR))`

que dará como resultado una u otra dirección de celda (C7 o C8), dependiendo del código del producto (G o S) incluido en C3. `@@(A10)` insertada en C4 dará como resultado el contenido de la celda cuya dirección proporciona la fórmula insertada en A10.

Si se incluye en C3 algo distinto de uno de los dos códigos de producto válidos, las funciones `@SI` y `@@` darán como resultado ERR.

A	-----	A	-----	B	-----	C	-----
1		CUADRO DE COMISIONES DE VENTAS					
2							
3		Escriba el código del producto:				G	
4		Porcentaje de comisión:				5%	
5							
6		Producto		Código		Comisión	
7		Gálar Pino		G		5%	
8		Superblank Hogar		S		3%	
9							
10		C7					

@ABS

@ABS(x) calcula el valor absoluto (positivo) de x .

Argumentos

x puede ser cualquier valor.

Notas

Utilice -@ABS para imponer que el resultado de la función @ sea negativo.

Ejemplos

@ABS(A5) = 25 cuando la celda A5 contenga el valor 25, -25 o una fórmula cuyo resultado sea 25 o -25.

-@ABS(A5) = -25 cuando la celda A5 contenga el valor 25, -25 o una fórmula cuyo resultado sea 25 o -25.

@ABS(COMIENZO-FIN) = 5,6 cuando COMIENZO y FIN contengan cualquier combinación de valores positivos o negativos cuya diferencia sea 5,6 o -5,6.

@ACUMULADO

@ACUMULADO(*negociación*; *emisión*; *primer_cupón*; *cupón*; [*valor_nominal*]; [*frecuencia*]; [*base*]) calcula el importe acumulado de los intereses para activos financieros que proporcionan pagos periódicos de intereses. La función @ACUMULADO podrá realizar el cálculo para períodos de cupón cortos, estándar y largos.

Argumentos

negociación representa la fecha en que se negocia o transmite el activo. *negociación* deberá ser un número de fecha. Si *negociación* es menor que *emisión*, @ACUMULADO dará como resultado ERR.

emisión representa la fecha de emisión del activo. *emisión* deberá ser un número de fecha.

primer_cupón representa la fecha de pago del primer cupón del activo. *primer_cupón* deberá ser un número de fecha. Si *primer_cupón* es menor o igual que *emisión*, @ACUMULADO dará como resultado ERR.

cupón representa el tipo de interés anual del cupón del activo. *cupón* puede ser cualquier valor positivo o 0.

valor_nominal es un argumento opcional que permite especificar el valor nominal del activo, es decir, el principal que se pagará al vencimiento. *valor_nominal* puede ser cualquier valor positivo. Si se omite el argumento *valor_nominal*, 1-2-3 utilizará el valor 100.

frecuencia es un argumento opcional que permite especificar el número de pagos de cupón por año. *frecuencia* deberá ser cualquiera de los valores indicados en la tabla siguiente:

<u><i>frecuencia</i></u>	<u>Frecuencia de los pagos de cupón</u>
1	Anual
2	Semestral; es el valor predeterminado si se omite este argumento
4	Trimestral
12	Mensual

base es un argumento opcional que permitirá especificar el tipo de base para el recuento de los días que se va a emplear. *base* deberá ser cualquiera de los valores indicados en la tabla siguiente:

<u><i>base</i></u>	<u>Base para el recuento de los días</u>
0	30/360; es el valor predeterminado si se omite este argumento
1	Real/real
2	Real/360
3	Real/365
4	Europeo 30/360

No podrá utilizarse un argumento opcional si se omiten los que le preceden.

Ejemplos

Un bono, cuya fecha de emisión es el 1 de diciembre de 1992 y cuyo primer pago de cupón es el 1 de junio de 1993, va a negociarse el 1 de julio de 1993. El tipo de interés del cupón semestral es del 5,50%, su valor nominal es de 100\$ y el recuento de los días se efectuará de acuerdo con la base 30/360.

Para determinar el importe acumulado de los intereses del bono:

@ACUMULADO(@FECHA(93;7;1);@FECHA(92;12;1);@FECHA(93;6;1);0,055;100;2;0) = 0,458333

Funciones @ similares

@ACUMULADO2 calcula el importe acumulado de los intereses para activos financieros que proporcionan pagos periódicos de intereses, según el sistema utilizado en Japón.

@PRECIO calcula el precio por cada 100\$ de valor nominal para un bono. @RENTABILIDAD calcula la rentabilidad para activos financieros que proporcionan pagos periódicos de intereses. @DURACION calcula la duración anual de un activo y @DURACIONM calcula la duración anual modificada para activos financieros que proporcionan pagos

periódicos de intereses.

@ACOS

@ACOS(x) calcula el arco coseno de un valor. El arco coseno de x es el ángulo cuyo coseno es x . El resultado de **@ACOS** será un ángulo expresado en radianes, comprendido entre 0 y π , ambos inclusive. Este valor corresponde a un ángulo entre 0 y 180 grados.

Argumentos

x es el coseno de un ángulo y puede ser un valor comprendido entre -1 y 1, ambos inclusive.

Ejemplos

El coseno de un ángulo z es 0,5. Para hallar el arco coseno, utilice **@ACOS(0,5)**. El resultado será 1,0472 radianes. Para convertir este valor en grados, utilice **@RADAGRAD(1,0472)**. El resultado, 60 grados, será el valor del ángulo z .

Funciones @ similares

@COS calcula el coseno de un ángulo. @ACOSH calcula el arco coseno hiperbólico de un valor.

@ACOSH

@ACOSH(x) calcula el arco coseno hiperbólico de un valor. El arco coseno hiperbólico de x es el ángulo cuyo coseno hiperbólico es x .

Argumentos

x es el coseno hiperbólico de un ángulo y puede ser cualquier valor mayor o igual que 1.

Ejemplos

@ACOSH(2) = 1,316958

Funciones @ similares

@ACOS calcula el arco coseno de un valor. **@COSH** calcula el coseno hiperbólico de un ángulo.

@ACOT

@ACOT(x) calcula el arco cotangente de un valor. El arco cotangente de x es el ángulo cuya cotangente es x . El resultado de @ACOT será un ángulo expresado en radianes, comprendido entre 0 y π , ambos inclusive. Este valor corresponde a un ángulo entre 0 y 180 grados.

Argumentos

x es la cotangente de un ángulo y puede ser cualquier valor.

Ejemplos

La cotangente de un ángulo z es 1,732051. Para hallar el ángulo z , utilice @ACOT(1,732051). El resultado será 0,523599 radianes. Para convertir este valor en grados, utilice @RADAGRAD(0,523599). El resultado será 30 grados.

Funciones @ similares

@COT calcula la cotangente de un ángulo. @ATAN calcula el arco tangente de un valor.

@ACOTH

@ACOTH(x) calcula el arco cotangente hiperbólica de un valor. El arco cotangente hiperbólica de x es el ángulo cuya cotangente hiperbólica es x.

Argumentos

x es la cotangente hiperbólica de un ángulo y puede ser cualquier valor menor que -1 o mayor que 1.

Ejemplos

@ACOTH(2) = 0,549306

Funciones @ similares

@ACOT calcula el arco cotangente de un valor. @COTH calcula la cotangente hiperbólica de un ángulo.

@ACOSEC

@ACOSEC(x) calcula el arco cosecante de un valor. El arco cosecante de x es el ángulo cuya cosecante es x . El resultado de **@ACOSEC** será un ángulo expresado en radianes, comprendido entre $-\pi/2$ y $\pi/2$. Este valor corresponde a un ángulo entre -90 y 90 grados.

Argumentos

x es la cosecante de un ángulo y puede ser un valor mayor o igual que 1 , o menor o igual que -1 .

Ejemplos

La cosecante de un ángulo z es $1,743447$. Para hallar el ángulo z , utilice **@ACOSEC(1,743447)**. El resultado será $0,610865$ radianes. Para convertir este valor en grados, utilice **@RADAGRAD(0,610865)**. El resultado será 35 grados.

Funciones @ similares

@ACOSECH calcula el arco cosecante hiperbólica de un valor. **@CSC** calcula la cosecante de un ángulo.

@ACOSECH

@ACOSECH(x) calcula el arco cosecante hiperbólica de un valor. El arco cosecante hiperbólica de x es el ángulo cuya cosecante hiperbólica es x .

Argumentos

x es la cosecante hiperbólica de un ángulo y puede ser cualquier valor distinto de 0.

Ejemplos

@ACOSECH(1,54) = 0,61068

Funciones @ similares

@ACOSEC calcula el arco cosecante de un valor. **@COSECH** calcula la cosecante hiperbólica de un ángulo.

@ASEC

@ASEC(x) calcula el arco secante de un valor. El arco secante de x es el ángulo cuya secante es x . El resultado de **@ASEC** será un ángulo expresado en radianes, comprendido entre 0 y π . Este valor corresponde a un ángulo entre 0 y 180 grados.

Argumentos

x es la secante de un ángulo y puede ser un valor menor o igual que -1 , o mayor o igual que 1 .

Ejemplos

En un triángulo rectángulo, la secante de un ángulo z es 2 . Para hallar el ángulo z , utilice **@ASEC(2)**. El resultado será $1,047198$ radianes. Para convertir este valor en grados, utilice **@RADAGRAD(1,047198)**. El resultado será 60 grados.

Funciones @ similares

@SEC calcula la secante de un ángulo. **@ASECH** calcula el arco secante hiperbólica de un valor.

@ASECH

@ASECH(x) calcula el arco secante hiperbólica de un valor. El arco secante hiperbólica de x es el ángulo cuya secante hiperbólica es x.

Argumentos

x es la secante hiperbólica de un ángulo y puede ser cualquier valor mayor que 0 y menor o igual que 1.

Ejemplos

@ASECH(0,5) = 1,316958

Funciones @ similares

@ASEC calcula el arco secante de un ángulo. @SECH calcula la secante hiperbólica de un ángulo.

@ASEN

@ASEN(x) calcula el arco seno de un valor. El arco seno de x es el ángulo cuyo seno es x . El resultado de **@ASEN** será un ángulo expresado en radianes, comprendido entre $-\pi/2$ y $\pi/2$, ambos inclusive. Este valor corresponde a un ángulo entre -90 y 90 grados.

Argumentos

x es el seno de un ángulo y puede ser un valor comprendido entre -1 y 1 , ambos inclusive.

Ejemplos

El seno de un ángulo z es $0,66$. Para hallar el ángulo z , utilice **@ASEN(0,66)**. El resultado será $0,72082$ radianes. Para convertir este valor en grados, utilice **@RADAGRAD(0,72082)**. El resultado será $41,3$ grados.

Funciones @ similares

@SEN calcula el seno de un ángulo. **@ASENH** calcula el arco seno hiperbólico de un valor.

@ASENH

@ASENH(x) calcula el arco seno hiperbólico de un valor. El arco seno hiperbólico de x es el ángulo cuyo seno hiperbólico es x.

Argumentos

x es el seno hiperbólico de un ángulo y puede ser cualquier valor.

Ejemplos

@ASENH(2) = 1,443635

Funciones @ similares

@SENH calcula el seno hiperbólico de un ángulo. @ASEN calcula el arco seno de un valor.

@ATAN

@ATAN(x) calcula el arco tangente de un valor. El arco tangente de x es el ángulo cuya tangente es x . El resultado de @ATAN será un ángulo expresado en radianes, comprendido entre $-\pi/2$ y $\pi/2$, ambos inclusive. Este valor corresponde a un ángulo entre -90 y 90 grados.

Argumentos

x es la tangente de un ángulo y puede ser cualquier valor.

Ejemplos

La tangente de un ángulo z es $2/1$, es decir, 2 . Para hallar el ángulo z , utilice @ATAN(2). El resultado será $1,10715$ radianes. Para convertir este valor en grados, utilice @RADAGRAD($1,10715$). El resultado será $63,4$ grados.

Funciones @ similares

@ATANH calcula el arco tangente hiperbólica de un valor. @ATAN2 calcula el valor de un ángulo cuya tangente es y/x . @TAN calcula la tangente de un ángulo.

@ATAN2

@ATAN2(x;y) calcula el arco tangente de un valor. El arco tangente es el ángulo cuya tangente es y/x . El resultado de @ATAN2 será un ángulo expresado en radianes, comprendido entre $-\pi$ y π , ambos inclusive. Este valor corresponde a un ángulo entre -180 y 180 grados, dependiendo del signo de x e y (las cuatro posibilidades se enumeran en el apartado Notas de este tema de Ayuda).

Argumentos

x e y son valores. Si y es 0, @ATAN2 dará como resultado el valor 0; si tanto x como y son 0, @ATAN2 dará como resultado el valor ERR.

Notas

La lista que aparece a continuación muestra los rangos de valores que puede tener @ATAN2.

- Si x es positivo e y es positivo, el resultado estará comprendido entre 0 y $\pi/2$ (1er cuadrante).
- Si x es negativo e y es positivo, el resultado estará comprendido entre $\pi/2$ y π (2º cuadrante).
- Si x es negativo e y es negativo, el resultado estará comprendido entre $-\pi$ y $-\pi/2$ (3er cuadrante).
- Si x es positivo e y es negativo, el resultado estará comprendido entre $-\pi/2$ y 0 (4º cuadrante).

Si x e y son positivos (primer cuadrante), y si x es positivo e y es negativo (cuarto cuadrante), los resultados serán los mismos que los que proporcionaría la función @ATAN.

Ejemplos

En un triángulo rectángulo, los catetos miden 1 y 2. Para hallar el valor del mayor de los dos ángulos agudos, utilice @ATAN2(1;2). El resultado será 1,10715 radianes. Para convertir este valor en grados, utilice @RADAGRAD(1,10715). El resultado será 63,4 grados.

Funciones @ similares

@ATAN calcula el arco tangente a partir de la tangente x de un ángulo. @TAN calcula la tangente de un ángulo.

@ATANH

@ATANH(x) calcula el arco tangente hiperbólica de un valor. El arco tangente hiperbólica de x es el ángulo cuya tangente hiperbólica es x.

Argumentos

x es la tangente hiperbólica de un ángulo y puede ser cualquier valor comprendido entre -1 y 1.

Ejemplos

@ATANH(0,544736) = 0,610865

Funciones @ similares

@ATAN calcula el arco tangente de un valor. @ATAN2 calcula el valor del ángulo cuya tangente es y/x. @TANH calcula la tangente hiperbólica de un ángulo.

@DESVMEDIA

[@DESVMEDIA](#)(*lista*) calcula la media de las desviaciones absolutas de los valores de *lista*.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas numéricas. Utilice separadores de argumentos para separar los elementos de *lista*.

Si desea más información, consulte el tema [Argumentos de las funciones @ estadísticas](#).

Ejemplos

[@DESVMEDIA](#)

Funciones @ similares

[@STD](#) y [@STDPURA](#) calculan la desviación típica de los valores de una lista. [@DESVCUAD](#) calcula la suma de los cuadrados de las desviaciones de los valores de una lista.

@MEDIA, @MEDIAPURA

@MEDIA(*lista*) calcula la media aritmética de los valores de *lista*.

@MEDIAPURA(*lista*) calcula la media aritmética de los valores de *lista*, pero ignorará las celdas que contengan rótulos.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas numéricas. Utilice separadores de argumentos para separar los elementos de *lista*.

Si desea más información, consulte el tema [Argumentos de las funciones @ estadísticas](#).

Ejemplos

@MEDIA(A2..A5) = 252,75, si A2..A5 contiene los valores 160, 227, 397 y 227.

@MEDIA(A1..A5) = 202,20, si A1..A5 contiene los valores 160, 227, 397 y 227, así como el rótulo Enero. (**@MEDIA** considerará que el rótulo posee el valor cero y lo tendrá en cuenta al efectuar el cálculo).

@MEDIAPURA(A1..A5) = 252,75, ya que **@MEDIAPURA** ignorará el rótulo Enero.

Funciones @ similares

[@BMEDIA](#) halla la media aritmética de los valores de un campo de una tabla de base de datos pertenecientes a registros que cumplan los criterios que se especifiquen. [@MEDIAGEOM](#) calcula la media geométrica y [@MEDIARMON](#) calcula la media armónica de los valores de una lista. [@MEDIANA](#) halla la mediana de los valores de una lista.

@BESSELI, @BESSELJ, @BESSELK, @BESSELY

@BESSELI($x;n$) calcula la función de Bessel de segunda especie de orden entero $Y_n(x)$.

@BESSELJ($x;n$) calcula la función de Bessel de primera especie de orden entero $J_n(x)$.

@BESSELK($x;n$) calcula la función de Bessel de segunda especie de orden entero $K_n(x)$.

@BESSELY($x;n$) calcula la función de Bessel de primera especie de orden entero $Y_n(x)$, también conocida como función de Neumann.

Argumentos

x es el valor para el que se evalúa la función; puede ser cualquier valor.

n es el orden de la función; puede ser cualquier entero positivo. Para @BESSELI y @BESSELJ, n también puede ser 0.

Notas

@BESSELI, @BESSELJ, @BESSELK y @BESSELY calculan el valor aproximado de las funciones respectivas con un margen de error de $\pm 5 \cdot 10^{-8}$.

Las funciones de Bessel se suelen utilizar para resolver problemas de simetría cilíndrica relacionados con la propagación de ondas, la dinámica de fluidos, la elasticidad y la difusión.

Ejemplos

@BESSELI(2;2) = 0,688948

@BESSELJ(1;0) = 0,765198

@BESSELK(3;0) = 0,03474

@BESSELY(1;1) = -0,781213

@BETA

@BETA(z;w) calcula la función beta.

Argumentos

z y w can pueden ser dos valores reales distintos de cero o de un entero negativo.

Notas

El resultado de @BETA tendrá una precisión de un mínimo de seis cifras significativas.

Ejemplos

@BETA(0,5;0,5) = 3,141593

Funciones @ similares

@BETA calcula la función beta incompleta. @GAMMA calcula la función gamma.

@BETAI

@BETAI($a;b;x$) da como resultado la función beta incompleta.

Argumentos

a y b pueden ser dos valores cualesquiera.

x puede ser cualquier valor entre 0 y 1.

Notas

El resultado de @BETAI tendrá una precisión de un mínimo de seis cifras significativas.

Ejemplos

@BETAI(5;0,5;0,668271) = 0,050012

Funciones @ similares

@BETA calcula la función beta.

@BINOMIAL

@BINOMIAL(*intentos*; *aciertos*; *probabilidad*; [*tipo*]) calcula la función de probabilidad binomial (o de Bernoulli) o la distribución binomial acumulativa.

Argumentos

intentos representa el número de intentos independientes; puede ser cualquier entero positivo.

aciertos representa el número de aciertos obtenidos en *intentos*; puede ser cualquier entero positivo o 0, y deberá ser menor o igual que *intentos*.

Si *intentos* y *aciertos* no son enteros, 1-2-3 truncará su parte fraccionaria.

probabilidad representa la probabilidad de acierto en cada intento; puede ser cualquier valor entre 0 y 1.

tipo es un argumento opcional que indica a 1-2-3 si debe calcular la función de probabilidad binomial o la distribución binomial acumulativa.

<i>tipo</i>	1-2-3 calcula
0	La probabilidad de que el número de aciertos sea exactamente igual a <i>aciertos</i> ; es el valor predeterminado si se omite este argumento
1	La probabilidad de que el número de aciertos sea igual a <i>aciertos</i> como máximo
2	La probabilidad de que el número de aciertos sea igual a <i>aciertos</i> como mínimo

Notas

@BINOMIAL calcula la distribución binomial acumulativa con un margen de error de $\pm 3 \cdot 10^{-7}$.

Ejemplos

Se han seleccionado al azar diez bebedores de refrescos de cola para que participen en una degustación a ciegas. A cada uno se le ofrece un vaso de refresco del tipo A y uno del tipo B. Ambos vasos son aparentemente idénticos, y sólo se diferencian en que tienen un código en la parte inferior para identificar el tipo de refresco de cola.

Suponiendo que no hay una tendencia entre los bebedores de refrescos que les induzca a preferir una marca respecto a la otra, la probabilidad de que uno de los participantes prefiera el refresco A es del 50%.

Para determinar la probabilidad de que exactamente 7 de los 10 participantes prefieran el refresco A:

@BINOMIAL(10;7;0,5) = 0,117188, o el 11,72%.

Para determinar la probabilidad de que al menos 7 de los 10 participantes prefieran el refresco A:

@BINOMIAL(10;7;0,5;2) = 0,171875, o el 17,19%.

Funciones @ similares

@CRITBINOMIAL calcula el valor máximo para el cual la distribución binomial acumulativa es menor o igual que un determinado criterio. @COMBIN calcula el número de combinaciones que pueden realizarse con un número determinado de valores. @PERMUT calcula el número de permutaciones que pueden realizarse con una lista de valores.

@CELDA, @INDICADORCELDA

@CELDA(*atributo*;*posición*) proporciona información sobre la primera celda de *posición*.

@INDICADORCELDA(*atributo*) proporciona información sobre la celda actual.

Argumentos

posición puede ser la dirección o el nombre de un rango.

atributo puede ser cualquiera de los elementos enumerados a continuación; deberá ir encerrado entre comillas (" ") y sin acento; también puede ser la dirección o el nombre de una celda que contenga uno de dichos elementos.

<u><i>atributo</i></u>	<u>Resultado</u>
atraves	1 si los datos en la celda están alineados a través de las columnas 0 si los datos en la celda no están alineados a través de las columnas
direccion	Dirección absoluta en forma abreviada (sólo la letra de columna y el número de fila)
colorfondo	El color de fondo, expresado como un entero entre 0 y 239, que especifica uno de los colores de la paleta
negrita	1 si la celda tiene asignado el atributo negrita 0 si la celda no tiene asignado el atributo negrita
bordeinferior	Un entero, entre 0 (sin borde) y 8, que especifica un estilo de línea
colorbordeinferior	Un entero, entre 0 y 239, que especifica un color para la línea
col	Letra de la columna, con un valor entre 1 y 256 (1 para la columna A, 2 para la columna B, etc.)
color	1 si se ha asignado formato a la celda para que los números negativos se representen en color 0 si no se ha asignado formato a la celda para que los números negativos se representen en color
contenido	Contenido de la celda
coord	Dirección absoluta de la celda en forma completa (letra de hoja, letra de columna y número de fila)
tipodatos	Tipo de datos que contiene la celda: b si la celda está vacía v si la celda contiene un número o una fórmula numérica r si la celda contiene un rótulo o una fórmula de texto e si la celda contiene el valor <u>ERR</u> n si la celda contiene el valor <u>ND</u>
fechaarchivo	Valor que indica la fecha y la hora en que se guardó por última vez el archivo que contiene la celda. Incluye tanto el <u>número de fecha</u> (parte entera) como el <u>número de hora</u> (parte decimal)
nombarch	Nombre del archivo que contiene la

	celda, con la ruta de acceso incluida
tipoletra	Tipo de letra de los datos contenidos en la celda
tamaño letra	Tamaño en puntos de los datos contenidos en la celda
formato	<p>Formato de la celda:</p> <p>M0 a M15: Monetario, de 0 a 15 decimales</p> <p>F0 a F15: Fijo, de 0 a 15 decimales</p> <p>G: General, un rótulo o una celda vacía</p> <p>P0 a P15: porcentaje, de 0 a 15 decimales</p> <p>C0 a C15: Científico, de 0 a 15 decimales</p> <p>.0 a .15: Punto (.), de 0 a 15 decimales</p> <p>+: Formato +/-</p> <p>D1: formato de Fecha 31-Dic-92</p> <p>D2: formato de Fecha 31-Dic</p> <p>D3: formato de Fecha Dic-92</p> <p>D4: formato de Fecha 31/12/92</p> <p>D5: formato de Fecha 31/12</p> <p>D6: formato de Hora 11:59:59 AM</p> <p>D7: formato de Hora 11:59 AM</p> <p>D8: formato de Hora 23:59:59</p> <p>D9: formato de Hora 23:59</p> <p>T: Texto</p> <p>O: Oculta</p> <p>R: Rótulo</p> <p>A: Automático</p> <p>--: Color para los números negativos</p> <p>() : Paréntesis</p>
tipofórmula	<p>Tipo de la fórmula contenida en la celda:</p> <p>b si la celda está vacía</p> <p>v si la celda contiene un número</p> <p>r si la celda contiene un rótulo</p> <p>fv si la celda contiene una fórmula numérica</p> <p>fr si la celda contiene una fórmula de cadena</p> <p>fe si la celda contiene una fórmula que da como resultado ERR</p> <p>fn si la celda contiene una fórmula que da como resultado ND</p>
alinearch	<p>Determina la alineación horizontal de los datos en la celda:</p> <p>0 si se ha seleccionado General (rótulos alineados a la izquierda; valores alineados a la derecha)</p> <p>1 para Izquierda</p> <p>2 para Centro</p> <p>3 para Derecha</p> <p>4 para Justificada</p>
altura	Altura de la fila, especificada en puntos
cursiva	1 si la celda tiene asignado el atributo cursiva

	0 si la celda no tiene asignado el atributo cursiva
bordeizquierdo	Un entero, entre 0 (sin borde) y 8, que especifica el estilo de línea
colorbordeizdo	Un entero, de 0 a 239, que especifica el color de la línea
orientacion	Orientación que se aplica al texto girado; se representa como un valor entero entre 0 y 4 y especifica el grado de orientación
parentesis	1 si la celda tiene asignado el formato Paréntesis 0 si la celda no tiene asignado el formato Paréntesis
trama	Un entero, entre 0 y 63, que especifica una trama
colortrama	Color de la trama, expresado como un entero entre 0 y 239, que especifica el color seleccionado de la paleta de colores
prefijo	' si la celda contiene un rótulo alineado a la izquierda " si la celda contiene un rótulo alineado a la derecha ^ si la celda contiene un rótulo centrado \ si la celda contiene un rótulo repetido si la celda contiene un rótulo que no se imprime Ninguno (sin prefijo de rótulo) si la celda está vacía o contiene un valor
proteger	1 si la celda está protegida 0 si la celda no está protegida
bordederecho	Un entero, entre 0 (sin borde) y 15, que especifica el estilo de línea
colorbordedcho	Un entero, entre 0 y 239, que especifica el color de la línea
rotacion	Ángulo de rotación del texto girado, expresado como un valor entero entre 0 y 90 grados
fila	Número de fila (un valor entre 1 y 8192)
hoja	Un valor entre 1 y 256 correspondiente a la letra de la hoja (1 para la hoja A, 2 para la hoja B, etc.)
nombrehoja	El nombre de la hoja o, en caso de no tener nombre, la letra de la hoja
colortexto	Color de los datos de la celda, expresado como un entero, entre 0 y 239, que especifica uno de los colores de la paleta
bordesuperior	Un entero, entre 0 (sin borde) y 8, que especifica el tipo de línea
colorbordesuperior	Un entero, entre 0 y 239, que especifica el color de la línea

tipo	El tipo de datos contenidos en la celda: b si la celda está vacía v si la celda contiene un valor numérico, una <u>fórmula numérica</u> o una <u>fórmula de cadena</u> r si la celda contiene un rótulo
subrayado	El estilo de subrayado expresado como un entero, entre 0 (sin subrayado) y 3, que especifica un estilo de subrayado
alinearv	Alineación vertical de los datos en la celda: 0 para Abajo 1 para Centro 2 para Arriba
ancho	Anchura de la columna
ajustelinea	1 si los datos se ajustan automáticamente en la celda 0 si los datos no se ajustan automáticamente en la celda

Notas

Presione F9 (**CALC**) para recalcular la hoja antes de utilizar @CELDA o @INDICADORCELDA; así podrá estar seguro de que los resultados son correctos.

@CELDA y @INDICADORCELDA resultan de utilidad en las macros y cuando se utilizan conjuntamente con @SI. Utilice @CELDA para comprobar los datos insertados durante la ejecución de una macro y rechazar ciertos tipos de datos, así como para bifurcar la ejecución de la macro hacia una u otra subrutina dependiendo de los datos especificados por el usuario. @CELDA podrá utilizarse también en una aplicación automatizada para cambiar los atributos de celda según los datos insertados por el usuario.

Utilice @INDICADORCELDA para determinar la posición actual del indicador de celda o para evaluar una fórmula basada en el contenido de la celda actual. Después podrá bifurcar el proceso de acuerdo con el contenido o el tipo de celda.

Podrá reemplazar el nombre de atributo tipo2 por tipodatos y el nombre de atributo tipo3 por tipoformula.

Ejemplos

En el siguiente ejemplo se ha utilizado @CELDA junto con @SI y @ERR para arrojar el resultado ERR, si el usuario no escribe un valor en la celda llamada IMPORTE, o bien para obtener el contenido de IMPORTE (un valor), si el usuario especifica un valor.

```
@SI(@CELDA("tipo";IMPORTE)="v";IMPORTE;@ERR)
```

En el siguiente ejemplo, se ha utilizado @INDICADORCELDA en una macro para identificar las celdas vacías de una lista de elementos; si 1-2-3 encuentra una celda vacía durante la ejecución de la macro, emitirá una señal sonora y bifurcará la ejecución de la macro hacia una subrutina.

```
{SI @INDICADORCELDA("tipo")="b"}{BIP}{BIFURCAR Paso2}
```

Funciones @ similares

@INFO da como resultado información acerca de la sesión actual de 1-2-3.

@CAR

@CAR(x) da como resultado el carácter correspondiente al código x del Juego de caracteres multibyte de Lotus (LMBCS).

Argumentos

x es un entero. Aquellos valores que no se correspondan con códigos de caracteres LMBCS darán como resultado ERR. Si x no es un número entero, 1-2-3 lo truncará para convertirlo en entero.

Notas

En el caso de que el monitor no pueda mostrar en pantalla el carácter que corresponda a x, 1-2-3 mostrará un carácter similar siempre que sea posible. Si no existe ningún carácter similar, 1-2-3 mostrará un rectángulo sólido, que representa los caracteres que no se pueden mostrar. Asegúrese de que la impresora puede reproducir los caracteres especificados.

Ejemplos

@CAR(156) = £ (signo de la libra esterlina).

@CAR(D9) = A, si la celda D9 contiene el valor 65.

Funciones @ similares

@CODIGO da como resultado el número del código LMBCS correspondiente a un carácter.

@DISTJI

@DISTJI(x ; *grados_libertad*; [*tipo*]) calcula la distribución ji cuadrado (χ^2).

Argumentos

x es el valor en el cual se desea evaluar la distribución ji cuadrado. El valor que se especifique para x dependerá del valor especificado para *tipo*.

<u>Si tipo es</u>	<u>x es</u>
0	El valor crítico o el límite superior para el valor de la variable aleatoria de la distribución acumulada ji cuadrado; puede ser un valor mayor o igual que 0; es el valor predeterminado si se omite este argumento
1	Una probabilidad (nivel de significación); puede ser un valor entre 0 y 1

grados_libertad es el número de grados de libertad para la muestra. *grados_libertad* deberá ser un entero positivo. Si *grados_libertad* no es un entero, 1-2-3 lo truncará para convertirlo en un entero.

tipo es un argumento opcional que permite indicar a 1-2-3 cómo deberá calcular @DISTJI.

<u>tipo</u>	<u>1-2-3 calcula</u>
0	El nivel de significación correspondiente a x ; es el valor predeterminado si se omite el argumento
1	El valor crítico correspondiente al nivel de significación x

Notas

@DISTJI calcula la distribución acumulada ji cuadrado con un margen de error de $\pm 3 \cdot 10^{-7}$. Si, tras 100 iteraciones de cálculo, @DISTJI no puede determinar el resultado con una precisión de 0,0000001, dará como resultado ERR.

La distribución ji cuadrado es una distribución continua, de un único parámetro y constituye un caso particular de la distribución gamma.

Utilice @DISTJI para comprobar la validez de una hipótesis comparando los valores experimentales con los valores teóricos.

Ejemplos

@DISTJI(12,592;6) = 0,05

@DISTJI(0,05;6;1) = 12,59159

Funciones @ similares

@TESTJI calcula la probabilidad asociada a una verificación mediante ji cuadrado. @DISTF calcula la distribución F .

@DISTT calcula la distribución t de Student.

@ELEGIR

@ELEGIR(x;lista) da como resultado el valor o el rótulo que ocupa la posición *x* en *lista*.

Argumentos

x es un valor. *x* representará un número offset, es decir, la posición ocupada por un elemento en *lista*.

lista puede contener un grupo de valores y rótulos, o direcciones o nombres de celdas que contengan valores y rótulos, separados por separadores de argumentos. 1-2-3 numerará cada entrada de *lista* y elegirá la que corresponda al valor de *x*.

Ejemplos

Una hoja de trabajo contiene una lista de rótulos en el rango A1..A4 y sus respectivos números offset (0;1;2;3) en el rango B1..B4.

@ELEGIR(B3;A1;A2;A3;A4) dará como resultado el rótulo contenido en A3, que será el elemento de la *lista* cuyo número offset es 2 (el valor 2 se encuentra en B3).

Funciones @ similares

@CONSULH y **@CONSULV** hallan entradas en tablas de consulta horizontales o verticales, respectivamente.

@INDICE y **@INTERSECCION** dan como resultado el contenido de una celda situada en la intersección de una columna, fila y hoja especificadas. **@BUSCARCONTENIDO** busca la posición que ocupa en un rango la celda cuyo contenido se especifica. **@BUSCARMAXIMO** da como resultado una referencia absoluta a la celda que contenga el valor más alto de una lista de rangos. **@BUSCARMINIMO** da como resultado una referencia absoluta a la celda que contenga el valor más bajo de una lista de rangos.

@DESPEJAR

@DESPEJAR(*cadena*) elimina los caracteres no imprimibles de *cadena*.

Argumentos

cadena puede ser cualquier cadena de texto encerrada entre comillas (" "), la dirección o el nombre de una celda que contenga un rótulo, o bien una fórmula o una función @ cuyo resultado sea una cadena.

Ejemplos

Se acaban de importar a 1-2-3 datos de un procesador de texto. La celda A45 contiene el rótulo

®Segundo, debemos actuar con rapidez.↵

@DESPEJAR(A45) = Segundo, debemos actuar con rapidez.

Funciones @ similares

@CAR da como resultado el carácter correspondiente a un código en el juego de caracteres LMBCS. @SUPRESP elimina los espacios iniciales y los finales, y reduce a uno los espacios consecutivos de una cadena.

@CODIGO

@CODIGO(cadena) da como resultado el código del Juego de Caracteres Multibyte de Lotus (LMBCS) correspondiente al primer carácter de *cadena*.

Argumentos

cadena puede ser cualquier cadena de texto encerrada entre comillas (" "), la dirección o el nombre de una celda que contenga un texto, o una fórmula o función @ cuyo resultado sea una cadena.

Ejemplos

@CODIGO("A") = 65

@CODIGO(C5) = 77, si C5 contiene el rótulo Martínez, ya que 77 es el código LMBCS de la letra M.

Funciones @ similares

@CAR da como resultado el carácter LMBCS correspondiente a un número de código especificado.

@COLS

@COLS(*rango*) cuenta el número de columnas contenidas en *rango*.

Argumentos

rango es un nombre o una dirección de rango.

Notas

Puede utilizarse **@COLS** con {DESDE} en macros que repitan la misma operación en una serie de columnas para determinar cuando debe detenerse la ejecución de la macro.

Ejemplos

@COLS(D9..J25) = 7, porque en *rango* se incluyen desde la columna D hasta la J (siete columnas).

@COLS(EMPLEADOS) = 2, si EMPLEADOS es el nombre del rango B3..C45.

Funciones @ similares

@CONVERTREF convierte las letras de las hojas o de las columnas de 1-2-3, desde A hasta IV, en números de 1 a 256. **@FILAS** cuenta el número de filas, y **@HOJAS** cuenta el número de hojas de trabajo de un rango.

@COMBIN

@COMBIN($n;r$) calcula el coeficiente binomial para n y r . El coeficiente binomial es el número de combinaciones que pueden realizarse con n elementos tomados de r en r , sin tener en cuenta el orden.

Argumentos

n representa el número total de elementos; puede ser cualquier entero positivo o 0.

r representa el número de elementos de que consta cada combinación; puede ser cualquier entero positivo o 0, y debe ser menor o igual que n .

Si n y r no son enteros, 1-2-3 truncará su parte decimal y los convertirá en enteros.

Notas

@COMBIN calcula el coeficiente binomial con una precisión de $\pm 3 \cdot 10^{-7}$.

Ejemplos

Un recipiente contiene cinco canicas de distintos colores. Se sacan tres canicas al azar. El número de combinaciones de colores posibles es

@COMBIN(5;3) = 10

Funciones @ similares

@BINOMIAL calcula la función de probabilidad binomial o la distribución binomial acumulativa. **@CRITBINOMIAL** calcula el valor máximo para el cual la distribución binomial acumulativa es menor o igual que un determinado criterio. **@PERMUT** calcula el número de permutaciones que pueden realizarse con un número determinado de elementos de un conjunto.

@COORD

@COORD(*hoja;columna;fila;absoluta*) crea una referencia a celda a partir de unos valores que correspondan a la *hoja, columna y fila*.

Argumentos

hoja y *columna* pueden ser cualquier número entero comprendido entre 1 y 256. *hoja* y *columna* corresponden a las letras de la hoja y de la columna (1 para la hoja o la columna A, 2 para la hoja o la columna B, etc).

fila puede ser cualquier número entero comprendido entre 1 y 8192. *fila* corresponde al número de fila.

absoluta puede ser cualquier número entero comprendido entre 1 y 8.

Si *hoja, columna, fila* y *absoluta* no son números enteros, 1-2-3 los truncará para convertirlos en enteros.

Notas

@COORD crea una referencia celda *relativa, absoluta* o *mixta*, según el valor contenido en *absoluta*. La siguiente tabla muestra los posibles valores de *absoluta* y su efecto sobre la dirección de celda A1 de la hoja A.

<u><i>absoluta</i></u>	<u>Valor de @COORD(1;1;1;<i>absoluta</i>)</u>
1	\$A:\$A\$1
2	\$A:A\$1
3	\$A:\$A1
4	\$A:A1
5	A:\$A\$1
6	A:A\$1
7	A:\$A1
8	A:A1

@COORD puede utilizarse conjuntamente con @INDICE, @CONSULV o @CONSULH para crear direcciones de celda a partir de tablas de valores del libro de trabajo activo. También puede utilizarse @COORD con @@ para obtener el valor de la dirección de celda creada por @COORD.

Ejemplos

@COORD(3;7;25;8) dará como resultado la dirección de celda relativa C:G25.

@(@COORD(C1;D1;E4;8)) dará como resultado el valor contenido en la celda A:A4 (si C1 contiene 1, D1 contiene 1 y E4 contiene 4).

Funciones @ similares

@CONVERTREF convierte las letras de las hojas o de las columnas de 1-2-3, desde A hasta IV, en números de 1 a 256.

@CORREL

@CORREL(*rango1*;*rango2*) calcula el coeficiente de correlación de los valores contenidos en *rango1* y en *rango2*.

Argumentos

rango1 y *rango2* son nombres o direcciones de rango. Deberán tener las mismas dimensiones. Si *rango1* y *rango2* no tienen las mismas dimensiones, **@CORREL** dará como resultado **ERR**.

1-2-3 emparejará las celdas de los dos rangos de acuerdo con la posición que ocupen en el rango. Las celdas de los rangos se consideran ordenadas de arriba a abajo, de izquierda a derecha y desde la primera hoja hasta la última.

Notas

Tanto el coeficiente de correlación como la covarianza miden la relación existente entre dos grupos de datos, pero se diferencian en que el primero es independiente de la unidad de medida, mientras que la covarianza sí que depende de ella.

Ejemplos

@CORREL

Funciones @ similares

@COVAR calcula la covarianza existente entre los valores contenidos en dos rangos.

@COS

@COS(z) calcula el coseno del ángulo z. El coseno de un ángulo de un triángulo rectángulo es la relación existente entre el cateto contiguo a dicho ángulo y la hipotenusa. El resultado de @COS será un valor comprendido entre -1 y 1, ambos inclusive.

Argumentos

z es el valor de un ángulo expresado en radianes. Puede ser cualquier valor comprendido entre -2^63 y 2^63 .

Ejemplos

@COS(@GRADARAD(30)) = 0,866 (coseno de un ángulo de 30 grados).

Funciones @ similares

@ACOS calcula el arco coseno de un valor, es decir, el ángulo cuyo coseno es igual a dicho valor. @COSH calcula el coseno hiperbólico de un ángulo.

@COSH

@COSH(z) calcula el coseno hiperbólico del ángulo z. El resultado de @COSH será un valor mayor o igual que 1.

Argumentos

z puede ser cualquier valor comprendido, aproximadamente, entre -709,7827 y 709,7827.

Ejemplos

@COSH(@GRADARAD(30)) = 1,140238 (coseno hiperbólico de un ángulo de 30 grados).

Funciones @ similares

@ACOS calcula el arco coseno de un valor, es decir, el ángulo cuyo coseno es igual a dicho valor. @COS calcula el coseno de un ángulo.

@COT

@COT(z) calcula la cotangente del ángulo z. La cotangente de uno de los ángulos agudos de un triángulo rectángulo es la relación existente entre el cateto adyacente al ángulo y el cateto opuesto al mismo.

Argumentos

z es un ángulo expresado en radianes y puede tener cualquier valor comprendido entre -2^{63} y 2^{63} .

Ejemplos

@COT(@GRADARAD(30)) = 1,73205 (cotangente de un ángulo de 30 grados).

Funciones @ similares

@ACOT calcula el arco cotangente de un valor. @COTH calcula la cotangente hiperbólica de un ángulo. @TAN calcula la tangente de un ángulo.

@COTH

@COTH(z) calcula la cotangente hiperbólica del ángulo z.

Argumentos

z puede ser cualquier valor comprendido, aproximadamente, entre -709,7827 y 709,7827, excepto el 0.

Ejemplos

@COTH(@GRADARAD(30)) = 2,081283 (cotangente hiperbólica de un ángulo de 30 grados).

Funciones @ similares

@ACOTH calcula el arco cotangente hiperbólica de un valor. @COT calcula la cotangente de un ángulo. @TANH calcula la tangente hiperbólica de un ángulo.

@CUENTA, @CUENTAPURA

@CUENTA(*lista*) cuenta las celdas no vacías existentes en una *lista* de rangos.

@CUENTAPURA(*lista*) cuenta las celdas de una *lista* de rangos, excluyendo las que contengan rótulos.

Argumentos

lista puede ser cualquier combinación de direcciones o nombres de rangos, separados por separadores de argumentos.

Si desea más información, consulte también el tema Argumentos de las funciones @ estadísticas.

Notas

@CUENTA cuenta todas las celdas incluidas en *lista* que contengan entradas de cualquier tipo, incluidos rótulos, caracteres de prefijo de rótulo o los valores ERR y ND.

@CUENTA y **@CUENTAPURA** son útiles para detener (o bifurcar) la ejecución de una macro que actúe sobre una serie de rangos cuando el indicador de celda llegue a un rango vacío.

Ejemplos

@CUENTA(A2..A3;A5) = 1, si A2..A3 se encuentra vacío, porque A5, esté o no vacía, es una dirección de una sola celda.

{SI **@CUENTAPURA**(SEPTIEMBRE)=0}{BIFURCAR AHF} se bifurcará a una macro llamada AHF (que calcula los totales año hasta la fecha), si el rango llamado SEPTIEMBRE está vacío o contiene un rótulo, un prefijo de rótulo o una fórmula de cadena.

Funciones @ similares

@BCUENTA y **@BCUENTAPURA** cuentan el número de celdas no vacías correspondientes a un campo de los registros de una tabla de base de datos que cumplan los criterios que se especifiquen.

@COVAR

@COVAR(*rango1*;*rango2*;*[tipo]*) calcula la covarianza de una población o de una muestra constituida por los valores contenidos en *rango1* y en *rango2*.

Argumentos

rango1 y *rango2* son nombres o direcciones de rango, deben tener las mismas dimensiones. Si *rango1* y *rango2* no tienen las mismas dimensiones, @COVAR dará como resultado ERR.

1-2-3 emparejará las celdas de los dos rangos de acuerdo con la posición que ocupen en el rango. Las celdas de los rangos se consideran ordenadas de arriba abajo, de izquierda a derecha y desde la primera hoja hasta la última.

tipo es un argumento opcional que permite indicar a 1-2-3 si se desea calcular la covarianza de una población o de una muestra.

<u>tipo</u>	<u>1-2-3 calcula</u>
0	Covarianza de una población; es el valor predeterminado si se omite este argumento
1	Covarianza de una muestra

Notas

La covarianza es la media aritmética de los productos de las desviaciones de los valores homólogos en dos listas de datos.

Tanto el coeficiente de correlación como la covarianza miden la relación existente entre dos grupos de datos, pero se diferencian en que el primero es independiente de la unidad de medida, mientras que la covarianza sí que depende de ella.

Ejemplos

@COVAR

Funciones @ similares

@VAR y @VARPURA calculan la varianza de una población, y @VARM y @VARM PURA calculan la varianza de una muestra constituida por los valores de una lista. @CORREL calcula el coeficiente de correlación de los valores homólogos contenidos en dos rangos.

@COSEC

@COSEC(z) calcula la cosecante del ángulo z. La cosecante es la función recíproca del seno. El resultado de @COSEC será un valor mayor o igual que 1, o menor o igual que -1.

Argumentos

z es un ángulo expresado en radianes y puede tener cualquier valor comprendido entre -2^{63} y 2^{63} , exceptuando el 0.

Ejemplos

@COSEC(@GRADARAD(30)) = 2 (cosecante de un ángulo de 30 grados).

Funciones @ similares

@ACOSEC calcula el arco cosecante de un valor. @COSECH calcula la cosecante hiperbólica de un ángulo. @SEN calcula el seno de un ángulo.

@COSECH

@COSECH(z) calcula la cosecante hiperbólica del ángulo z. La cosecante hiperbólica es la función recíproca del seno hiperbólico.

Argumentos

z puede ser cualquier valor comprendido entre -709,7827 y 709,7827, ambas cifras aproximadas.

Ejemplos

@COSECH(@GRADARAD(30)) = 1,825306 (cosecante hiperbólica de un ángulo de 30 grados).

Funciones @ similares

@ACOSECH calcula el arco cosecante hiperbólica de un valor. @COSEC calcula la cosecante de un ángulo.

@SENH calcula el seno hiperbólico de un ángulo.

@CPERODO

@CPERODO(*interés;valor_futuro;valor_actual*) calcula el número de periodos necesarios para que una inversión (*valor_actual*) alcance un *valor_futuro* determinado, a un tipo de *interés* compuesto que se supone constante a lo largo de todos los periodos.

Argumentos

interés puede ser cualquier valor mayor que -1 excepto 0.

valor_futuro y *valor_actual* pueden ser valores cualesquiera, pero deberán tener el mismo signo (ambos positivos o ambos negativos).

Ejemplos

Supongamos que acaba de depositar 10.000\$ a un interés anual del 10% (0,10), pagadero mensualmente, y que desea calcular el número de años que tardará la inversión en doblarse.

@CPERODO(0,10/12;20000;10000)/12 = 6,960312.

Es decir, habrán de transcurrir siete años para que se doble la inversión original de 10.000\$.

Dado que **@CPERODO** calcula el número total de periodos de capitalización a interés compuesto, puede que sea necesario incluir el número de periodos en los que se divide el tipo de *interés* con el fin de que el periodo y el tipo de interés se midan según la misma unidad de tiempo. En el ejemplo anterior, se trata de un interés anual del 10%, que se abona mensualmente, por lo que se especifica de la forma 0,10/12 (el *interés* dividido entre el número de periodos de capitalización).

Funciones @ similares

@PERODO y **@NPERODO** calculan el número de pagos periódicos de igual cuantía necesarios para acumular un valor futuro determinado.

@@

@@(*posición*) permite obtener de forma indirecta el contenido de la celda especificada en *posición*.

Argumentos

posición es la dirección o nombre de una celda que contiene una dirección o nombre de celda, o una fórmula cuyo resultado es la dirección o el nombre de una celda.

posición hace referencia a otra celda, cuyo contenido será mostrado por la función **@@** en la celda que contenga dicha función. Cuando *posición* no sea una dirección de celda o un nombre de rango válidos, o cuando sea un rango con varias celdas, la función **@@** dará como resultado **ERR**.

Notas

@@ es útil para crear fórmulas condicionales puesto que su referencia indirecta puede alterar automáticamente su propio valor. Por ejemplo, la fórmula

@SI(D2="S";"D8";@SI(D2="N";"D9";@ERR))

en la celda A10, y la fórmula **@@(A10)** en la celda E2, darán como resultado el contenido de las celdas D8 o D9, o **ERR**, en la celda E2, dependiendo de si D2 contiene S, N u otros valores.

Cuando *posición* haga referencia a una celda que contenga una fórmula, presione F9 (**CALC**) para actualizar la fórmula de **@@** una vez realizado el recálculo automático. Si no se presiona F9 (**CALC**), la fórmula de **@@** dará como resultado el valor 0.

Ejemplos

@@

Funciones @ similares

@CONSULH y **@CONSULV** dan como resultado el contenido de una celda específica en una tabla de consulta horizontal o vertical. **@ELEGIR** da como resultado un valor o un rótulo de *lista* y **@INDICE** da como resultado el contenido de una celda ubicada en la intersección de una hoja, una fila y una columna específicas.

@D360, @DIAS360

@D360(*fecha_inicial;fecha_final*) calcula el número de días existentes entre dos fechas, basándose en un año de 360 días (12 meses de 30 días cada uno).

@DIAS360(*fecha_inicial;fecha_final*) calcula el número de días existentes entre dos fechas, basándose en un año de 360 días y de acuerdo con la normativa del mercado de valores de los Estados Unidos.

Argumentos

fecha_inicial y *fecha_final* son números de fecha.

Notas

La fórmula empleada para calcular **@DIAS360** se ajusta a las modificaciones publicadas en 1990 para la edición de 1986 de la publicación *Standard Security Calculation Methods*, de la Securities Industry Association de los Estados Unidos.

@DIAS360 y **@D360** suelen dar resultados distintos para los mismos datos cuando *fecha_inicial* o *fecha_final* coincide con el último día de un mes.

Ejemplos

@DIAS360(**@FECHA**(89;4;16);**@FECHA**(89;9;25)) = 159

@D360(33290;33524) = 232, el número de días existentes entre el 21 de febrero de 1991 y el 13 de octubre de 1991, basándose en un año de 360 días.

Funciones @ similares

@DIFERFECHA calcula el número de años, meses o días existentes entre dos fechas. **@DIAS** calcula el número de días existentes entre dos fechas utilizando una base determinada para el recuento de los días.

@LABORABLESNETOS calcula el número de días transcurridos entre dos fechas sin contar los fines de semana ni festivos. **@LABORABLE** calcula la fecha correspondiente a un número de días anterior o posterior a una fecha dada, sin contar fines de semana ni festivos. **@MESSIGUIENTE** calcula la fecha de un número de meses anterior o posterior a una fecha determinada.

@FECHA

@FECHA(año;mes;día) calcula el número de fecha correspondiente al *año*, *mes* y *día* especificados.

Argumentos

año es un número entero comprendido entre 0 (año 1900) y 199 (año 2099).

mes es un número entero comprendido entre 1 y 12.

día es un número entero comprendido entre 1 y 31. El valor utilizado para *día* deberá ser un día válido para el *mes* especificado. Por ejemplo, no podrá especificarse 31 como *día* si se utiliza 4 (abril) como *mes*.

Notas

Aunque el 29 de febrero de 1900 no existió (no era un año bisiesto), 1-2-3 asigna automáticamente un número de fecha a este día. Esto no invalidará los cálculos de fecha a menos que se utilicen fechas comprendidas entre el 1 de enero de 1900 y el 1 de marzo de 1900. Si utiliza fechas pertenecientes a ese período, deberá restar 1 a los resultados de las operaciones realizadas.

Si desea que el resultado de un cálculo de la función **@FECHA** se muestre con formato de fecha, asigne a la celda donde se encuentra la función **@FECHA** uno de los formatos de fecha.

Ejemplos

@FECHA(92;2;21) dará como resultado 33655 o 21-Feb-92 en una celda a la que se haya asignado el formato DD-Mmm-AA.

@FECHA(91;2;29) dará como resultado ERR, porque 1991 no fue un año bisiesto.

Funciones @ similares

@VALFECHA calcula el número de fecha correspondiente a una fecha especificada como rótulo. **@VHORA** calcula el número de hora correspondiente a la hora especificada. **@ACTUAL** calcula el número de fecha y hora correspondientes a la fecha y hora actuales.

@DIFERFECHA

@DIFERFECHA(*fecha_inicial;fecha_final;formato*) calcula el número de años, meses o días existentes entre dos números de fecha.

Argumentos

fecha_inicial y *fecha_final* son números de fecha.

formato es uno de los códigos de la tabla siguiente; deberá especificarse en forma de cadena de texto y sirve para especificar el formato que se desee que tenga el resultado de @DIFERFECHA.

<u>format</u>	<u>Dará como resultado el número de</u>
a	Años
m	Meses
d	Días
md	Días, pero ignorando los meses y los años
am	Meses, pero ignorando los años
ad	Días, pero ignorando los años

Ejemplos

En los ejemplos siguientes se emplean las fechas 15 de febrero de 1990 y 15 de septiembre de 1993.

@DIFERFECHA(@FECHA(90;2;15);@FECHA(93;9;15);"m") da como resultado 43, el número de meses existentes entre el 15 de febrero de 1990 y el 15 de septiembre de 1993.

@DIFERFECHA(@FECHA(90;2;15);@FECHA(93;9;15);"md") da como resultado 0, ya que el día del mes para la *fecha_inicial* y la *fecha_final* es el 15.

@DIFERFECHA(@FECHA(90;2;15);@FECHA(93;9;15);"am") da como resultado 7, el número de meses existentes entre febrero y septiembre.

Funciones @ similares

@D360 y @DIAS360 calculan el número de días existentes entre dos fechas, basándose en un año de 360 días (con 12 meses de 30 días cada uno). @DIAS calcula el número de días existentes entre dos fechas, basándose en una determinada base para el recuento de los días.

@INFOFECHA

@INFOFECHA(*fecha*; *atributo*) proporciona información acerca de un número de fecha.

Argumentos

fecha es un número de fecha.

atributo puede ser cualquiera de los números enteros indicados en la tabla siguiente:

<i>atributo</i>	Da como resultado
1	Día de la semana en forma de rótulo, con formato corto (Lun)
2	Día de la semana en forma de rótulo, con formato largo (Lunes)
3	Día de la semana como número entero del 0 (Lunes) al 6 (Domingo)
4	Semana del año como número entero del 1 al 53
5	Mes del año en forma de rótulo, con formato corto (Ene)
6	Mes del año en forma de rótulo, con formato largo (Enero)
7	Número de días existentes en el mes correspondiente a <i>fecha</i>
8	Número de días restantes en el mes correspondiente a <i>fecha</i>
9	Último día del mes correspondiente a <i>fecha</i>
10	Trimestre al que pertenece <i>fecha</i> , como número entero del 1 al 4
11	1 si el año correspondiente a <i>fecha</i> es un año bisiesto; 0 si no es bisiesto
12	Día del año correspondiente a <i>fecha</i> , como número del 1 al 366
13	Número de días restantes en el año correspondiente a <i>fecha</i>

Ejemplos

@INFOFECHA(23063;7) = 28 (el número de días que tuvo el mes de febrero de 1963).

@INFOFECHA(@FECHA(92;10;5);10) = 4 (octubre pertenece al cuarto trimestre).

@VALFECHA

@VALFECHA(cadena) calcula el número de fecha correspondiente a la fecha especificada en *cadena*.

Argumentos

cadena puede ser texto encerrado entre comillas (" "), una fórmula cuyo resultado sea texto, o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo. El argumento *cadena* deberá tener una de las siguientes formas:

- Uno de los formatos de fecha que aparezcan en la lista "Más usados"
- Uno de los cinco formatos aceptados en las celdas de formato General: 31-Dic-96, Dic-96, 31-Dic, así como los formatos corto y largo para fecha, especificados en los parámetros regionales (del país) del sistema operativo.

Notas

Si desea que el resultado de **@VALFECHA** se muestre con formato de fecha, asigne a la celda donde se encuentre la función **@VALFECHA** uno de los formatos de fecha.

@VALFECHA podrá utilizarse con datos importados desde otra aplicación, como un programa de tratamiento de texto.

Ejemplos

@VALFECHA("21-Feb-91") dará como resultado el número de fecha 33290.

@VALFECHA(CUMPLEAÑOS) dará como resultado el número de fecha 20723 si la celda CUMPLEAÑOS contiene la cadena de fecha 25-Sep-56.

Funciones @ similares

@CADENAFECHA convierte un número de fecha en la fecha equivalente y la muestra como rótulo. **@FECHA** calcula el número de fecha correspondiente a una fecha determinada.

@DIA

@DIA(*número_fecha*) calcula el día del mes (un entero del 1 al 31) correspondiente a *número_fecha*.

Argumentos

número_fecha es un valor comprendido entre 1 (1 de enero de 1900) y 73050 (31 de diciembre del 2099).

Notas

Es posible utilizar otra de las funciones @ de fecha para obtener el valor de *número_fecha*.

@DIA podrá proporcionar el argumento *día* a otras funciones @ de fecha que partan de fechas previamente calculadas.

Ejemplos

@DIA(@ACTUAL) = el día actual del mes.

@DIA(D9) = 12 si la celda D9 contiene el número de fecha 33250 (correspondiente a la fecha 12-Ene-91).

Funciones @ similares

@MES calcula el mes correspondiente a un número de fecha, @AÑO calcula el año correspondiente a un número de fecha y @DIASEMANA calcula el día de la semana correspondiente a un número de fecha.

@DIAS

@DIAS(*fecha_inicial*; *fecha_final*; [*base*]) calcula el número de días existentes entre dos fechas utilizando una *base* determinada para el recuento de los días.

Argumentos

fecha_inicial y *fecha_final* son números de fecha. Si *fecha_inicial* es anterior a *fecha_final*, @DIAS dará un resultado positivo. Si *fecha_inicial* es posterior a *fecha_final*, el resultado de @DIAS será negativo. Si *fecha_inicial* y *fecha_final* son iguales, el resultado de @DIAS será 0.

base es un argumento opcional que permite especificar el tipo de base utilizada para el recuento de los días. El argumento *base* deberá ser cualquiera de los valores indicados en la tabla siguiente:

<u>base</u>	<u>Base para el recuento de los días</u>
0	30/360; es el valor predeterminado si se omite el argumento
1	Real/real
2	Real/360
3	Real/365
4	Europeo 30/360

Ejemplos

@DIAS(@FECHA(93;4;16);@FECHA(93;9;25)) = 159, el número de días existentes entre el 16 de abril de 1993 y el 25 de septiembre de 1993, basado en un año de 360 días y doce meses, cada uno de 30 días.

@DIAS(@FECHA(93;4;16);@FECHA(93;9;25);1) = 162, el número de días existentes entre el 16 de abril de 1993 y el 25 de septiembre de 1993, basado en el número real de días existentes en los meses de abril a septiembre.

Funciones @ similares

@DIFERFECHA calcula el número de años, meses o días existentes entre dos fechas. @D360 y @DIAS360 calculan el número de días existentes entre dos fechas, basándose en un año de 360 días. @LABORABLESNETOS calcula el número de días transcurridos entre dos fechas sin contar fines de semana ni días festivos.

@DB

@DB(*coste;valor_residual;vida;período*) calcula, por el método de los saldos decrecientes (Declining Balance), la cuota de amortización de un bien de inversión con un *coste* inicial, una *vida* útil estimada y un *valor_residual* dados, durante un *período* de tiempo determinado.

Argumentos

coste representa el importe pagado por el bien y puede ser cualquier valor positivo o 0. Si *coste* es igual a 0, el resultado de **@DB** será 0.

valor_residual representa el valor estimado para el bien al final de su vida útil y puede ser cualquier valor positivo o 0. Si *valor_residual* es mayor que *coste*, el resultado de **@DB** será negativo.

vida representa el número de períodos que deberán transcurrir para que el bien se deprecie hasta su *valor_residual*; puede ser cualquier valor mayor o igual que 1.

período representa el período de tiempo concreto para el que se desee calcular la cuota de amortización; podrá ser cualquier valor mayor o igual que 1.

Los argumentos *vida* y *período* deberán expresarse en la misma unidad de tiempo, que suele ser años.

Notas

El método de los saldos decrecientes decelera la amortización de un bien en comparación con el método de los saldos decrecientes al 200%; en ambos se considera que en cada período se produce (y se contabiliza) un gasto de amortización menor que en el anterior. La amortización finaliza cuando el valor contable del bien (es decir, el coste total del bien menos el importe total de las cuotas de amortización practicadas durante los períodos anteriores) alcanza el valor residual.

Ejemplos

Acaba de comprarse una máquina de oficina cuyo coste inicial es de 10.000\$, su vida útil es de ocho años y su valor residual después de esos ocho años se estima en 1.200\$. Para calcular los gastos de amortización correspondientes al quinto año por el método de los saldos decrecientes podrá utilizarse la fórmula:

@DB(10000;1200;8;5) = 806,51\$

Funciones @ similares

@VDB calcula la cuota de amortización por el método de los saldos decrecientes al 200% y permite especificar otras tasas de amortización diferentes del 200%. **@DDB** utiliza el método de los saldos decrecientes al 200%, **@SLN** utiliza el método de amortización lineal y **@SYD** utilizará el método de los números dígitos.

@DDB

@DDB(*coste;valor_residual;vida;período*) calcula, por el método de los saldos decrecientes al 200% (Double Declining Balance), la cuota de amortización de un bien de inversión con un *coste* inicial, una *vida* útil estimada y un *valor_residual* dados, durante un *período* de tiempo determinado.

Argumentos

coste representa el importe pagado por el bien y puede ser cualquier valor mayor o igual que *valor_residual*.

valor_residual representa el valor estimado para el bien al final de su vida útil y puede ser cualquier valor.

vida representa el número de períodos que deberán transcurrir para que el bien se deprecie hasta su *valor_residual*; puede ser cualquier valor mayor que 2.

período representa el período de tiempo concreto para el que desea calcularse la cuota de amortización; puede ser cualquier valor mayor o igual que 1.

Los argumentos *vida* y *período* deberán expresarse en la misma unidad de tiempo, que suele ser años.

Notas

El método de los saldos decrecientes al 200% acelerará la amortización de un bien calculando cuotas de amortización decrecientes, lo que equivale a considerar que en cada período se producirá (y se contabilizará) un gasto de amortización menor que en el anterior. La amortización finalizará cuando el valor contable del bien (es decir, el coste total del bien menos el importe total de las cuotas de amortización practicadas durante los períodos anteriores) alcance el valor residual.

Si el valor residual de un bien es relativamente bajo, podrá ocurrir que **@DDB** no lo amortice totalmente al final de su vida útil. En ese caso, podrá utilizarse **@VDB**, ya que esta función siempre amortizará totalmente el bien en la vida útil estimada.

Ejemplos

Se ha realizado la adquisición de una máquina de oficina cuyo coste inicial es de 10.000\$, su vida útil es de ocho años y su valor residual después de esos ocho años se estima en 1.200\$. Para calcular los gastos de amortización correspondientes al quinto año por el método de los saldos decrecientes al 200%, podrá utilizarse la fórmula:

@DDB(10000;1200;8;5) = 791,02\$

Funciones @ similares

@VDB calcula la cuota de amortización por el método de los saldos decrecientes al 200% y permite especificar otras tasas de amortización diferentes del 200%. **@DB** utiliza el método de los saldos decrecientes, **@SLN** utiliza el método de amortización lineal y **@SYD** utilizará el método de los números dígitos.

@DECIMAL

@DECIMAL(*hexadecimal*) convierte un valor hexadecimal en su equivalente decimal con signo.

Argumentos

hexadecimal puede ser cualquier valor comprendido entre 00000000 y FFFFFFFF, ambos inclusive, especificado como texto. Deberá tener un máximo de ocho caracteres y sólo podrá contener los números del 0 al 9 y las letras de la A a la F. Las letras podrán estar en mayúsculas o en minúsculas.

Notas

Los valores hexadecimales del 00000000 al 7FFFFFFF corresponden al 0 y a valores decimales positivos.

Los valores hexadecimales del 80000000 al FFFFFFFF corresponden a valores decimales negativos.

Ejemplos

@DECIMAL("1A") = 26

@DECIMAL("FFFFFFFE") = -2

Funciones @ similares

@HEX convierte números decimales en números hexadecimales.

@GRADARAD

@GRADARAD(*grados*) convierte *grados* en radianes.

Argumentos

grados puede ser cualquier valor.

Ejemplos

@GRADARAD(30) = 0,523599 radianes

@COS(@GRADARAD(45)) = 0,707107 (coseno de un ángulo de 45 grados).

Funciones @ similares

@RADAGRAD convierte radianes en grados.

@DESVCUAD

@DESVCUAD(*lista*) calcula la suma de los cuadrados de las desviaciones de los valores de *lista* con relación a su media aritmética.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas numéricas. Utilice separadores de argumentos para separar los elementos de *lista*.

Consulte también Argumentos de las funciones @ estadísticas.

Ejemplos

@DESVCUAD(2;3;9;8;15;2;1) = 159,4286

Funciones @ similares

@STD y @STDPURA calculan la desviación típica de los valores de una lista.

@FER

@FER(*límite_inferior*;*[límite_superior]*) calcula la función error integrada entre *límite_inferior* y *límite_superior*.

Argumentos

límite_inferior es el límite inferior utilizado para la integración de @FER y puede ser cualquier valor.

límite_superior es un argumento opcional que permite especificar el límite superior utilizado para integrar @FER . Si se omite el argumento *límite_superior*, @FER calcula la integral entre 0 y el *límite_inferior*.

Notas

@FER calcula la función error con un margen de error de $\pm\epsilon 1,2*10^{-7}$.

Ejemplos

@FER(0,7) = 0,677801

@FER(0,8) = 0,742101

@FER(0,7;0,8) = 0,0643 (diferencia entre los dos ejemplos anteriores).

Funciones @ similares

@FERC calcula la función error complementaria. @FERD calcula la derivada de la función error.

@FERC

@FERC(x) calcula la función error complementaria, integrada entre x e ∞ .

Argumentos

x puede ser cualquier valor.

Notas

@FERC(x) es igual a $1 - \text{@FER}(x)$.

@FERC calcula el valor de la función error complementaria con un margen de error de $\pm 3 \cdot 10^{-7}$.

Ejemplos

@FERC(0,7) = 0,322199

Funciones @ similares

@FER calcula la función error integrada entre un límite inferior y uno superior especificados. @FERD calcula la derivada de la función error.

@FERD

@FERD(x) calcula la derivada de la función error.

Argumentos

x puede ser cualquier valor.

Ejemplos

@FERD(0,7) = 0,691275

Funciones @ similares

@FER calcula la función error integrada entre un límite inferior y uno superior especificados.

@ERR

@ERR arroja como resultado el valor ERR.

Notas

La función @ERR resulta útil a la hora de hallar errores en el cálculo de operaciones, pero raramente se utiliza por sí sola. Por ejemplo, si se especifica @ERR como argumento de @SI, se obtendrá el valor ERR cuando existan determinadas condiciones, por ejemplo, cuando una fórmula devuelva un valor no aceptable, como un pago mensual negativo.

Ejemplos

@SI(B14>3;@ERR;B14) = ERR, si el valor contenido en B14 es mayor que 3.

Funciones @ similares

@ND da como resultado el valor ND (no disponible). @ESERR detecta el valor ERR.

@PAR

@PAR(x) redondea el valor x al primer número entero par que encuentre al alejarse de 0.

Argumentos

x puede ser cualquier valor. Si x es un entero par, @PAR dará como resultado x.

Ejemplos

@PAR(2,25) = 4

@PAR(2) = 2

@PAR(-2,25) = -4

Funciones @ similares

@IMPAR redondea un valor al primer número entero impar que encuentre al alejarse de 0. @REDOND, @REDONDABAJO y @REDONDARRIBA redondean un valor a un número determinado de decimales. @REDONDMULTP redondea un valor a un múltiplo determinado. @ENT trunca un valor eliminando su parte decimal y lo convierte en un entero. @TRUNC trunca un valor a partir de una posición decimal determinada.

@IGUAL

@IGUAL(*cadena1*;*cadena2*) compara dos grupos de caracteres para comprobar si son iguales. Si coinciden exactamente, **@IGUAL** dará como resultado 1 (verdad); en caso contrario, dará como resultado 0 (falso).

Argumentos

cadena1 y *cadena2* pueden ser cadenas de texto encerradas entre comillas (" "), fórmulas cuyo resultado sea texto, o direcciones o nombres de celdas que contengan rótulos o fórmulas que den como resultado rótulos.

Notas

@IGUAL proporciona una alternativa de mayor precisión que el operador igual (=) para fórmulas. A diferencia de éste, **@IGUAL** distingue entre letras mayúsculas y minúsculas y letras con o sin acentos.

Es posible utilizar **@IGUAL** para establecer contraseñas para macros comparando los datos especificados por el usuario con una cadena determinada antes de continuar la ejecución de la macro.

Ejemplos

@IGUAL("ATENAS";"Atenas") = 0 (falso).

@IGUAL("Debe";B2) = 1 (verdad), si la celda B2 contiene el rótulo Debe.

@IGUAL("400";400) = ERR, porque *cadena2* no es una cadena, sino un valor.

@EXP

@EXP(x) calcula el valor de la constante e (aproximadamente 2,718282) elevada a la potencia x.

Argumentos

x puede ser un valor comprendido entre -709,7827 y 709,7827, ambas cifras aproximadas.

Notas

Si x es mayor que 709,7827, aproximadamente, o menor que -709,7827, aproximadamente, el valor calculado será demasiado elevado o demasiado bajo para que 1-2-3 pueda almacenarlo, por lo que @EXP dará como resultado ERR.

Ejemplos

@EXP(0,7) = 2,013753

Funciones @ similares

@EXP2 calcula el valor de la constante e elevada a la potencia $-x^2$. @LN es la función inversa de @EXP.

@EXP2

@EXP2(x) calcula el valor de la constante e (aproximadamente 2,718282) elevada a la potencia (-x^2).

Argumentos

x puede ser un valor comprendido entre -106,570 y 106,570, ambas cifras aproximadas.

Notas

Si x es mayor que aproximadamente 106,570 o menor que aproximadamente -106,570, el valor calculado será demasiado elevado o demasiado bajo para que 1-2-3 pueda almacenarlo, por lo que @EXP2 dará como resultado ERR. Si x es menor que aproximadamente -15,098 o mayor que aproximadamente 15,098, 1-2-3 podrá calcular y almacenar el resultado de @EXP2, pero no podrá mostrarlo en pantalla (en la celda se mostrarán una serie de asteriscos).

Ejemplos

@EXP2(0,7) = 0,612626

Funciones @ similares

@EXP calcula el valor de la constante e elevada a una potencia determinada.

@FACT

@FACT(n) calcula el factorial de n .

Argumentos

n puede ser cualquier entero positivo o 0.

Notas

El factorial de n es igual al producto de todos los enteros positivos desde 1 hasta n .

Si n es mayor o igual que 171, el resultado de los cálculos será demasiado elevado para que 1-2-3 pueda almacenarlo y @FACT dará como resultado ERR.

Ejemplos

@FACT(0) = 1

@FACT(5) = 120 (el resultado de la multiplicación $1*2*3*4*5$).

Funciones @ similares

@FACTLN calcula el logaritmo neperiano del factorial de n . @PRODUCTO calcula el producto de los valores de una lista.

@FACTLN

@FACTLN(n) calcula el logaritmo neperiano del factorial de n .

Argumentos

n puede ser cualquier entero positivo o 0.

Notas

El factorial de n es igual al producto de todos los enteros positivos desde 1 hasta n .

Ejemplos

@FACTLN(0) = 0, es decir, el resultado de @LN(1).

@FACTLN(5) = 4,787492, es decir, el resultado de @LN(1*2*3*4*5).

Funciones @ similares

@FACT calcula el factorial de n . @LN calcula el logaritmo neperiano de un valor.

@FALSO

@FALSO arroja como resultado el valor lógico 0 (falso).

Notas

Si una expresión lógica, como por ejemplo $A1=B1$, es verdadera, su valor lógico será 1. Si es falsa, su valor lógico será 0.

La utilización de @FALSO en fórmulas que evalúan condiciones lógicas equivale a utilizar el valor 0, pero @FALSO facilita la lectura de dichas fórmulas.

Utilice la función @FALSO con macros o funciones @ como @SI o @ELEGIR que precisen el valor lógico 0 (falso). @FALSO resulta útil cuando se utiliza como argumento y de @SI, ya que ese argumento representa el valor obtenido cuando no se cumple la condición.

Ejemplos

@SI(A6>500;@VERDAD;@FALSO) = 0 cuando la celda A6 contenga un valor menor o igual que 500.

Funciones @ similares

@VERDAD da como resultado el valor lógico 1.

@DISTF

@DISTF(x ; $grados_libertad1$; $grados_libertad2$; [$tipo$]) calcula la distribución F .

Argumentos

x es el valor en el que se desea evaluar la distribución F . El valor que puede especificarse para x dependerá del valor especificado para $tipo$.

<u>Si $tipo$ es</u>	<u>x es</u>
0	El valor crítico o límite superior para el valor de la distribución F acumulada (es un valor mayor o igual que 0); es el valor predeterminado si se omite este argumento
1	Una probabilidad (un valor de 0 a 1)

$grados_libertad1$ y $grados_libertad2$ representan los números de grados de libertad para la primera y la segunda muestra, respectivamente. Los argumentos $grados_libertad1$ y $grados_libertad2$ deberán ser enteros positivos.

$tipo$ es un argumento opcional que permiten indicar a 1-2-3 el procedimiento de cálculo de la función @DISTF.

<u>$tipo$</u>	<u>1-2-3 calcula</u>
0	El nivel de significación correspondiente al valor crítico x ; es el valor predeterminado si se omite este argumento
1	El valor crítico correspondiente al nivel de significación x

Notas

@DISTF aproxima la distribución F acumulada dentro de un margen de error de $\pm \epsilon 3 \cdot 10^{-7}$. Si, tras 100 iteraciones de cálculo, @DISTF no puede determinar el resultado con una precisión de 0,0000001, dará como resultado ERR.

La distribución F es una distribución continua obtenida a partir de la relación existente entre dos distribuciones ji cuadrado, cada una de ellas dividida entre su número de grados de libertad.

Utilice @DISTF para determinar el grado de variación de dos muestras.

Ejemplos

@DISTF(3,07;8;10) = 0,05

@DISTF(0,05;8;10) = 0,999865

Funciones @ similares

@TESTF calcula la probabilidad asociada a una prueba F . @DISTJI calcula la distribución ji cuadrado. @DISTT calcula la distribución t de Student.

@BUSCAR

@BUSCAR(*cadena_búsqueda;cadena;posición_inicial*) calcula la posición que ocupa *cadena_búsqueda* la primera vez que aparece dentro de *cadena*. 1-2-3 comenzará la búsqueda a partir de la posición indicada por *posición_inicial*.

Argumentos

cadena_búsqueda y *cadena* son cadenas de texto encerradas entre comillas (" "), fórmulas cuyo resultado sea texto, o la dirección o el nombre de una celda que contenga texto o una fórmula cuyo resultado sea un rótulo.

posición_inicial es un número offset.

Notas

Si 1-2-3 no encuentra *cadena_búsqueda* dentro de *cadena*, **@BUSCAR** dará como resultado ERR; lo mismo sucederá si *posición_inicial* es mayor que el número de caracteres de *cadena*, o si *posición_inicial* es un número negativo.

@BUSCAR tendrá en cuenta si los caracteres están escritos en mayúsculas o en minúsculas y si llevan acento o no. Por ejemplo, **@BUSCAR** no hallará la *cadena_búsqueda* "e" en la *cadena* "HOTEL".

@BUSCAR también resulta útil si se utiliza conjuntamente con **@MED** o **@SUSTITUIR** para localizar y extraer o reemplazar cadenas de texto.

Ejemplos

@BUSCAR("P";"Ramiro Pérez";0) = 7 porque la *cadena_búsqueda* P ocupa la posición 7 en la *cadena* Ramiro Pérez.

@VALFUT, @VALFUTURO

@VALFUT(*pagos;interés;plazo*) calcula el valor futuro de una inversión constituida por una serie de *pagos* iguales por los que se retribuye una tasa de *interés* dada, durante el número de períodos indicado en *plazo*.

@VALFUTURO(*pagos;interés;plazo;[tipo];[valor_actual]*) calcula el valor futuro de una inversión que tiene un determinado *valor_actual* y está constituida por una serie de *pagos* iguales por los que se retribuye una tasa de *interés* dada, durante el número de períodos indicado en *plazo*. @VALFUTURO realiza el cálculo para anualidades ordinarias o anticipadas, dependiendo del valor especificado para el argumento *tipo*.

Argumentos

pagos y *plazo* deben ser valores.

Para @VALFUTURO, *plazo* debe ser un valor positivo.

interés debe ser un valor mayor que -1.

tipo es un argumento opcional que permite especificar si se desea realizar el cálculo para una anualidad ordinaria o para una anualidad anticipada.

<u><i>tipo</i></u>	<u>1-2-3 realiza el cálculo para</u>
0	Anualidad ordinaria (los pagos se efectúan al final de cada período); es el valor predeterminado si se omite este argumento
1	Anualidad anticipada (los pagos se efectúan al principio de cada período)

valor_actual es un argumento opcional que permite especificar el valor actual de la serie de pagos futuros, y puede ser cualquier valor. Si se omite el argumento *valor_actual*, 1-2-3 utilizará el valor 0.

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Notas

La unidad de tiempo utilizada para calcular el *interés* deberá coincidir con la utilizada en *plazo*; por ejemplo, si se efectúan los cálculos para pagos mensuales, el interés y el plazo también deberán expresarse en meses.

Normalmente, esto supondrá dividir el tipo de interés por 12 y multiplicar el número de años de *plazo* por 12.

Ejemplos

Se desea depositar durante los próximos 20 años la cantidad de 2.000\$ al año para disponer de ellas tras la jubilación en una cuenta que devenga un 7,5% de interés, capitalizado anualmente. Los intereses se abonan el último día de cada año. Se desea calcular el capital que de que dispondrá la cuenta dentro de 20 años. Los depósitos se realizan el último día de cada año.

@VALFUT(2000;0,075;20) da como resultado 86.609,36\$, el valor de la cuenta al cabo de los 20 años.

Si se realiza el depósito de cada año el primer día del año:

@VALFUTURO(2000;0,075;20;1) = 93.105,06\$

Funciones @ similares

@VALFUT2 calcula el valor futuro de una inversión constituida por una serie de *pagos* iguales por los que se retribuye una tasa de *interés* dada, durante el número de períodos indicado en *plazo*, suponiendo una anualidad anticipada. @VALFUTCANTIDAD da como resultado el valor futuro de una suma global invertida a una tasa determinada durante un número dado de períodos. @VALACT y @VALACTUAL determinan el valor actual de una inversión. @VAN calcula el valor actual neto de una inversión, descontando el valor futuro hasta el valor actual.

Ejemplo de @INDICE

En una tabla llamada AUMENTOS (A3..E8) se muestran los aumentos salariales en una empresa según la evaluación del rendimiento de los empleados.

@INDICE(AUMENTOS;2;3), incluida en una celda con formato porcentaje dará como resultado el valor 5%, correspondiente al aumento salarial de un empleado del nivel salarial 2 que en la evaluación del rendimiento obtuvo la calificación 3.

@INDICE(AUMENTOS;1;2), incluida en una celda con formato porcentaje dará como resultado el valor 7%, correspondiente al aumento salarial de un empleado del nivel salarial 1 que en la evaluación del rendimiento obtuvo la calificación 2.

A	B	C	D	E	
1					
2					
3	Calificación	1	2	3	4
4	1	10%	9%	8%	7%
5	2	7%	6%	5%	4%
6	3	6%	5%	4%	3%
7	4	3%	2%	1%	0%
8	5	0%	0%	0%	0%

@GAMMA

@GAMMA(x) calcula la función gamma.

Argumentos

x puede ser cualquier valor excepto 0 y números enteros negativos.

Notas

@GAMMA calcula aproximadamente la distribución gamma con una precisión de seis cifras significativas.

Ejemplos

@GAMMA(0,5) = 1,772454

@GAMMA(5) = 24

Funciones @ similares

@BETA calcula la función beta. @GAMMAI calcula la función gamma incompleta. @GAMMALN calcula el logaritmo neperiano de la función gamma.

@GAMMAI

@GAMMAI($a;x;[complemento]$) calcula la función gamma incompleta.

Argumentos

a puede ser cualquier valor positivo.

x puede ser cualquier valor positivo o 0.

complemento es un argumento opcional que permite indicar a 1-2-3 el procedimiento para calcular @GAMMAI.

<u>complemento</u>	<u>1-2-3 calcula</u>
0	P($a;x$); es el valor predeterminado si se omite este argumento
1	Q($a;x$); equivale a 1-P($a;x$)

Notas

@GAMMAI calcula aproximadamente la función gamma incompleta con una precisión de seis cifras significativas.

Ejemplos

@GAMMAI(7,5;12,497;1) = 0,050024

Funciones @ similares

@GAMMA calcula la función de distribución gamma. @GAMMALN calcula el logaritmo neperiano de la función gamma.

@GAMMALN

@GAMMALN(x) calcula el logaritmo neperiano de la función gamma.

Argumentos

x puede ser cualquier valor distinto de 0 y de los enteros negativos.

Notas

@GAMMALN calcula aproximadamente el logaritmo neperiano de la función gamma con una precisión de seis cifras significativas.

Ejemplos

@GAMMALN(0,5) = 0,572365

@GAMMALN(5) = 3,178054

Funciones @ similares

@GAMMA calcula la función de distribución gamma. @GAMMAI calcula la función gamma incompleta.

@MEDIAGEOM

@MEDIAGEOM(*lista*) calcula la media geométrica de los valores de una *lista*.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas numéricas. Utilice separadores de argumentos para separar los elementos de *lista*.

Todos los valores de *lista* deben ser mayores que 0.

Consulte también Argumentos de las funciones @ estadísticas.

Notas

La media geométrica de una *lista* de n valores es la raíz n ésimo del producto de los valores de la *lista*.

Para una misma *lista*, el resultado de **@MEDIAGEOM** será menor que el resultado de **@MEDIA**, salvo si todos los valores de *lista* son iguales. En este caso, los resultados de **@MEDIAGEOM** y **@MEDIA** serán iguales.

Ejemplos

@MEDIAGEOM(A1..A4) = 239,1886, si A1..A4 contiene los valores 160, 227, 397 y 227.

Funciones @ similares

@MEDIARMON calcula la media armónica de los valores de una lista. **@MEDIA** y **@MEDIAPURA** calculan la media aritmética de los valores de una lista. **@MEDIANA** calcula la mediana de los valores de una lista.

@MEDIARMON

@MEDIARMON(*lista*) calcula la media armónica de los valores de *lista*.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas numéricas. Utilice separadores de argumentos para separar los elementos de *lista*.

Todos los valores de *lista* deben ser mayores que 0.

Consulte también Argumentos de las funciones @ estadísticas.

Notas

La media armónica de los valores de *lista* es la inversa de la media aritmética de los inversos de los valores de *lista*. Por ejemplo, la media aritmética de 1/2, 1/3 y 1/4 es 13/36; por tanto, la media armónica de 2, 3 y 4 será 36/13.

Para la misma *lista*, el resultado de **@MEDIARMON** siempre será menor que el resultado de **@MEDIAGEOM**.

Ejemplos

@MEDIARMON(25;50;75) = 40,90909

Funciones @ similares

@MEDIAGEOM calcula la media geométrica de los valores de una lista. **@MEDIA** y **@MEDIAPURA** calculan la media aritmética de los valores de una lista. **@MEDIANA** calcula la mediana de los valores de una lista.

@HEX

@HEX(x) convierte un número decimal en su equivalente hexadecimal.

Argumentos

x puede ser cualquier número entero comprendido entre -2.147.483.648 y 2.147.483.647. Si x no es un entero, 1-2-3 truncará su parte decimal y lo convertirá en un entero.

Notas

Los valores hexadecimales del 00000000 al 7FFFFFFF corresponden al 0 y a valores decimales positivos.

Los valores hexadecimales del 80000000 al FFFFFFFF corresponden a valores decimales negativos.

Ejemplos

@HEX(162) = A2

Funciones @ similares

@DECIMAL convierte números hexadecimales en números decimales.

@CONSULH

@CONSULH(*x*; *rango*; *fila*) localiza el contenido de una celda en una fila determinada de una tabla horizontal de consulta: un rango cuya primera fila contiene valores en orden ascendente o r tulos.

Argumentos

x puede ser un valor o una cadena, dependiendo del contenido de la primera fila de la tabla horizontal de consulta.

<u>Primera fila</u>	<u>x</u>
Valores	Cualquier valor mayor o igual que el primer valor contenido en <i>rango</i> . Si es menor, @CONSULH da como resultado el valor ERR. Si <i>x</i> es mayor que el �ltimo valor de la primera fila de <i>rango</i> , @CONSULH se detendr� en la �ltima celda de la fila especificada en <i>fila</i> y dar� como resultado el contenido de esa celda
R�tulos	Texto encerrado entre comillas (" "), una f�rmula cuyo resultado sea texto, o la direcci�n o el nombre de una celda que contenga un r�tulo o una f�rmula cuyo resultado sea un r�tulo. Si <i>x</i> no coincide exactamente con el contenido de ninguna celda de la primera fila de <i>rango</i> , @CONSULH dar� como resultado ERR

rango representa la posici n de la tabla horizontal de consulta y puede ser una direcci n o un nombre de rango. Si se trata de un rango tridimensional, 1-2-3 utilizar  solamente la primera hoja de *rango*.

fila representa el n mero offset correspondiente a la posici n que ocupa la fila en *rango*.

Notas

@CONSULH comparar  *x* con cada celda de la primera fila de la tabla. Cuando 1-2-3 localice en la primera fila una celda que contenga *x* (o, si *x* es un valor, el valor m s aproximado sin superarlo), descender  por esa columna el n mero de filas especificado en *fila* y dar  como resultado el contenido de la celda en la que se detenga.

Ejemplos

@CONSULH

Funciones @ similares

@CONSULV busca el contenido de una celda en una tabla vertical de consulta. @INDICE localiza el contenido de una celda cuando se especifican n meros offset de columna y de fila. @ELEGIR da como resultado el contenido de una celda de una tabla de consulta formada por una  nica fila. @BUSCARCONTENIDO busca la posici n que ocupa en un rango la celda cuyo contenido se especifica. @INTERSECCION busca en un rango el contenido de una celda especificada mediante el t tulo de la columna, la fila y la hoja donde se encuentra. @BUSCARMAXIMO da como resultado una referencia absoluta a la celda que contenga el valor m s alto de una lista de rangos. @BUSCARMINIMO da como resultado una referencia absoluta a la celda que contenga el valor m s bajo de una lista de rangos.

@HORA

@HORA(*número_hora*) extrae la hora, es decir, un valor comprendido entre 0 (medianoche) y 23 (23:00 u 11:00 PM), ambos inclusive, a partir de *número_hora*.

Argumentos

número_hora puede ser un valor comprendido entre 0,000000 (medianoche) y 0,999988 (11:59:59 PM), ambos inclusive. Por lo general, se utiliza junto con otra función **@** de hora que suministra el *número_hora*.

Notas

El cálculo de la hora resulta útil en operaciones que requieren horas enteras, como por ejemplo, el cálculo de salarios pagados por horas, las horas transcurridas desde el comienzo de un proyecto, o bien para incluir en una hoja de trabajo la hora en que se realizó un determinado cambio.

Ejemplos

@HORA(0,51565) = 12 porque 0,51565 es el número de hora correspondiente a las 12:22:32 PM.

@HORA(**@VHORA**(13;45;18)) = 13 (1:00 PM), porque 13 es el argumento *hora* de **@VHORA**(13;45;18).

Funciones @ similares

@MINUTO extrae los minutos y **@SEGUNDO** extrae los segundos de un número de hora.

@SI

`@SI(condición;x;y)` comprueba el valor del argumento *condición* y da como resultado un valor entre dos posibles, dependiendo del resultado de la comprobación. Si *condición* es verdadera, `@SI` da como resultado *x*; si *condición* es falsa, da como resultado *y*.

Argumentos

condición puede ser una fórmula lógica. No obstante, es posible utilizar cualquier fórmula, número, texto encerrado entre comillas (" "), o el nombre o la dirección de una celda como *condición*. 1-2-3 evaluará cualquier *condición* distinta de cero como verdadera, y cualquier *condición* igual a cero como falsa. Las celdas vacías, el texto y los valores ERR y ND son equivalentes a cero cuando se utilizan como *condición*.

x e *y* pueden ser valores, texto encerrado entre comillas (" "), o direcciones o nombres de celda que contengan valores o rótulos.

Notas

`@SI` resulta útil también cuando se utiliza conjuntamente con `@ERR` y `@ND` para localizar errores o datos no existentes en fórmulas. Además, puede utilizarse para evitar la obtención de los valores ERR y ND, o para evitar que se produzcan errores de cálculo en situaciones en las que falten datos o estos no sean del tipo adecuado; por ejemplo, puede usarse para evitar una división por cero.

Es posible anidar varias funciones `@SI` para elaborar una condición compleja. Por ejemplo, la fórmula

```
@SI(TOT>10000;TOT*0,15;@SI(TOT>5000;TOT*0,10;TOT*0,02))
```

anida dos funciones `@SI` para crear una fórmula que calcula el importe de una comisión basándose en tres niveles de ventas: superiores a 10.000\$, superiores a 5.000\$ e inferiores o iguales a 5.000\$

Ejemplos

`@SI(SALDO>=0;SALDO;"Deudor")` da como resultado el valor contenido en la celda denominada SALDO si dicho valor es positivo o 0; si se trata de un valor negativo, dará como resultado el rótulo Deudor.

@INDICE

@INDICE(rango;columna;fila;[hoja]) da como resultado el contenido de la celda situada en una determinada *columna*, *fila* y (opcionalmente) *hoja* de un rango.

Argumentos

rango es la dirección o el nombre de un rango.

columna representa el número offset de la columna utilizada por @INDICE.

fila representa el número offset de la fila utilizada por @INDICE, o la dirección o el nombre de una celda que contenga el valor 0 o un entero positivo.

hoja es un argumento opcional y representa el número offset de la hoja utilizada por @INDICE. Si no se especifica ninguna *hoja*, @INDICE utilizará sólo la primera hoja de *rango*.

Ejemplos

@INDICE

Funciones @ similares

@CONSULH y @CONSULV buscan entradas en tablas de consulta horizontales o verticales, respectivamente.

@ELEGIR busca una entrada en una lista. @BUSCARCONTENIDO busca la posición que ocupa en un rango la celda cuyo contenido se especifica. @INTERSECCION busca en un rango el contenido de una celda especificada mediante el título de la columna, la fila y la hoja donde se encuentra. @BUSCARMAXIMO da como resultado una referencia absoluta a la celda que contenga el valor más alto de una lista de rangos. @BUSCARMINIMO da como resultado una referencia absoluta a la celda que contenga el valor más bajo de una lista de rangos.

@INFO

@INFO(*atributo*) facilita información sobre el sistema para la sesión de trabajo actual de 1-2-3.

Argumentos

atributo puede ser uno de los elementos que se muestran a continuación, insertados como cadena.

<u>atributo</u>	<u>Da como resultado</u>
autor	El nombre de usuario de la persona que guardó el libro actual por primera vez
fecha-creacion	Un <u>número de fecha</u> que corresponde a la fecha en que se guardó el libro actual por primera vez
tiempo-revision	Un <u>número de hora</u> que corresponde al número total de horas y minutos que ha permanecido abierto el libro actual
codretbd	ND
mensajecontbd	ND
recuentoregbd	ND
directorio	La <u>ruta de acceso</u> actual, con la letra de la unidad
ultimo-revisor	El nombre de usuario de la persona que guardó el libro actual por última vez
fecha-ultima-revision	Un número de fecha que corresponde a la fecha en que se guardó el libro actual por última vez
macro-paso	Sí, si está activado el modo Paso; No, si está desactivado
macro-rastrear	Sí, si la ventana Rastrear macro está abierta; No, si no está abierta
memdisp	La cantidad de memoria disponible
modo	El modo actual: 0 Espere 1 Activo 3 Menú 4 Valor 5 Señalar 6 Edición 7 Error 8 Buscar 9 Ficheros 10 Inicio 11 Estado 13 Nombres 99 El resto de los modos (por ejemplo, los definidos por el usuario con <u>{INDICAR}</u>)
numfichs	El número de archivos abiertos en ese momento
origen	La dirección absoluta de la celda superior izquierda de la hoja activa
codretso	ND
versionso	ND
recalc	El modo activo de recálculo expresado con uno de los siguientes rótulos: Automático o Manual

versión	El número de la versión de 1-2-3 para Windows utilizada, compuesto por tres partes: El número de la versión principal, el nivel de actualización y el número de la edición
configurar-nombre-usuario	El nombre de usuario para el correo electrónico o para la red
pantalla-altura	La altura de la pantalla, en píxeles
pantalla-anchura	La anchura de la pantalla, en píxeles
seleccion	La dirección del rango que se encuentre seleccionado, o el nombre del gráfico, del objeto dibujado o de la tabla de consulta que se encuentren seleccionados
seleccion-parte	ND
seleccion-tipo	El tipo de la selección actual: Rango, Dibujo, Consulta o Gráficos
sistema	El nombre del sistema operativo
memtot	La memoria total disponible (la cantidad disponible en el momento actual y la que se esté utilizando)
dirwin	La ruta de acceso al directorio que contenga Windows, incluida la letra de la unidad
número-hojas	El número de hojas en el libro actual
tamaño-hojas	El tamaño del libro actual, en Kilobytes (KB)

Notas

Además de los atributos que aparecen en la lista anterior, *atributo* puede ser cualquiera de los componentes del cuadro de información.

Es preciso recalcular el trabajo (presionando F9 (CALC)) antes de utilizar la función @INFO para garantizar la exactitud de los resultados obtenidos.

@INFO resultará de utilidad en macros cuando se precise suministrar información sobre el estado de 1-2-3 al usuario o a la macro (por ejemplo, para informar al usuario sobre cuál es la ruta de acceso actual en una macro que automatiza el almacenamiento de los libros o para avisarle de que dispone de poca memoria).

Utilice @INFO junto con @SI para comprobar el estado de 1-2-3 y dirigir la actuación de la macro en base a determinadas condiciones. Por ejemplo, para cambiar la ruta de acceso en caso necesario, para eliminar datos innecesarios o para cerrar libros abiertos cuando se dispone de poca memoria.

Ejemplos

@INFO("numfichs") = 2, si hay dos libros abiertos.

@INFO(B4) = 3, si B4 contiene el rótulo "modo" y 1-2-3 se encuentra en el modo Menú.

Funciones @ similares

@CELDA da como resultado información acerca de la primera celda de un rango. @INDICADORCELDA da como resultado información sobre la celda actual.

@ENT

@ENT(x) arroja como resultado la parte entera de x.

Argumentos

x es un valor.

Notas

Utilice el Formato fijo para mostrar en pantalla valores con un número determinado de decimales si desea que 1-2-3 calcule los valores con la máxima precisión posible; no utilice en ese caso la función @ENT.

Ejemplos

@ENT(35,67) = 35

@ENT(@ACTUAL) = número de fecha correspondiente a la fecha actual sin la parte correspondiente a la hora, ya que ésta es un valor decimal.

Funciones @ similares

@REDOND, @REDONDARRIBA y @REDONDABAJO redondean un valor al múltiplo de la potencia determinada de 10 más próximo. @REDONDMULTP redondea un valor a un múltiplo determinado. @PAR redondea un valor al primer número entero par que encuentre al alejarse de 0. @IMPAR redondea un valor al primer número entero impar que encuentre al alejarse de 0. @TRUNC trunca un valor a partir de una posición decimal determinada.

@AMORTINT, @AMORTPRINC

@AMORTINT(*principal;interés;plazo;período_inicial;[período_final;tipo;valor_futuro]*) calcula la suma de las partes correspondientes al interés de los pagos periódicos de un crédito (*principal*) con un tipo de *interés* periódico dado y un determinado número de períodos de pago (*plazo*).

@AMORTPRINC(*principal;interés;plazo;período_inicial;[período_final;tipo;valor_futuro]*) calcula la suma de las partes correspondientes al principal de los pagos periódicos de un crédito (*principal*) con un tipo de *interés* periódico dado y un determinado número de períodos de pago (*plazo*).

Argumentos

principal y *plazo* deben ser valores. *plazo* puede ser cualquier valor excepto 0.

interés debe ser un valor decimal o un porcentaje mayor que -1.

período_inicial es el número del período de pago del crédito en que se desea comenzar a calcular el interés o el principal. Puede ser cualquier valor mayor o igual que 1, pero no puede ser mayor que *plazo*.

período_final es el número del período de pago del crédito en que se desea finalizar el cálculo del interés o el principal. Puede ser cualquier valor mayor que *período_inicial*. Si se omite el argumento *período_final*, éste se igualará a *período_inicial*.

tipo es un argumento opcional que permite especificar si se desea realizar el cálculo para una anualidad ordinaria o para una anualidad anticipada.

<u><i>tipo</i></u>	<u>1-2-3 realiza el cálculo para</u>
0	Anualidad ordinaria (los pagos se efectúan al final de cada período); es el valor predeterminado si se omite este argumento
1	Anualidad anticipada (los pagos se efectúan al principio de cada período)

valor_futuro es un argumento opcional que permite especificar el valor futuro de la serie de pagos. Puede ser cualquier valor. Si se omite el argumento *valor_futuro*, 1-2-3 utilizará el valor 0.

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Notas

La unidad de tiempo utilizada para calcular el *interés* debe coincidir con la utilizada en *plazo*; por ejemplo, si se efectúan los cálculos para pagos mensuales, el interés y el plazo también deberán expresarse en meses. Normalmente, esto supone dividir el tipo de interés por 12 y multiplicar el número de años de *plazo* por 12.

Ejemplos

Suponga que ha pedido un crédito de 8.000\$ a pagar en 3 años con un interés anual del 10,5%, compuesto mensualmente. Los pagos mensuales son de 260,02\$. Para determinar la parte correspondiente al interés de los pagos del último año:

@AMORTINT(8000;0,105/12;36;25;36) = 170,45\$

Para determinar la parte correspondiente al principal de los pagos del último año:

@AMORTPRINC(8000;0,105/12;36;25;36) = 2.949,79\$

Funciones @ similares

@AMORT calcula el importe de los pagos periódicos necesarios para amortizar un crédito.

@TASAINT

@TASAINT(*plazo*; *pagos*; *valor_actual*; [*tipo*; *valor_futuro*; *estimación*]) calcula el tipo (o tasa) de interés compuesto necesario para que una inversión (*valor_actual*) alcance un *valor_futuro* durante un determinado número de períodos de pago (*plazo*).

Argumentos

plazo debe ser un entero positivo.

pagos y *valor_actual* deben ser valores.

tipo es un argumento opcional que permite especificar si se desea realizar el cálculo para una anualidad ordinaria o para una anualidad anticipada.

<u>tipo</u>	<u>1-2-3 realiza el cálculo para</u>
0	Anualidad ordinaria (los pagos se efectúan al final de cada período); es el valor predeterminado si se omite este argumento
1	Anualidad anticipada (los pagos se efectúan al principio de cada período)

valor_futuro es un argumento opcional que permite especificar el valor futuro de la serie de pagos. Puede ser cualquier valor. Si se omite el argumento *valor_futuro*, 1-2-3 utilizará el valor 0.

estimación es un argumento opcional mediante el que puede suministrarse a 1-2-3 una estimación para el tipo de interés. Debe ser un valor entre 0 y 1. Si se omite el argumento *estimación*, 1-2-3 utilizará el valor 0,10 (10%).

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Notas

@TASAINT calculará el tipo de interés mediante aproximaciones sucesivas partiendo del valor del argumento *estimación*; por tanto, deberá especificar en *estimación* un valor que considere razonable como tipo de interés. Como puede haber varias soluciones, si el resultado obtenido es menor que 0 o mayor que 1, continúe probando nuevos valores para *estimación*.

Si @TASAINT no consigue aproximarse al resultado con una precisión de 0,0000001 después de 30 iteraciones de cálculo, dará como resultado ERR. Si especifica otros valores como *estimación* y sigue obteniendo ERR, utilice @VAN para determinar una *estimación* más acertada. Si @VAN da como resultado un valor positivo, significará que su estimación es demasiado baja; si da como resultado un valor negativo, significará que es demasiado elevada. Cuando la estimación es exacta, da como resultado 0.

La unidad de tiempo utilizada para calcular la *estimación* debe coincidir con la utilizada en *plazo*; por ejemplo, si se efectúan los cálculos para pagos mensuales, el interés y el plazo también deberán expresarse en meses. Normalmente, esto supone dividir el tipo de interés por 12 y multiplicar el número de años de *plazo* por 12.

Ejemplos

Suponga que ha depositado 6.000\$ en una cuenta y que desea retirar 100\$ al mes durante ocho años. Para determinar el interés que deben pagarle para poder hacer los reintegros utilice la siguiente fórmula:

@TASAINT(96;100;6000;0;0;0,01) = 0,010623, es decir, un 1,06% compuesto mensualmente.

Funciones @ similares

@VAN calcula el valor actual neto de una serie de flujos de caja futuros. @VALACT y @VALACTUAL calculan el valor actual de una inversión constituida por una serie de pagos iguales. @VALFUT y @VALFUTURO calculan el valor futuro de una inversión. @TASA da como resultado el tipo de interés compuesto necesario para que una inversión alcance en el futuro un valor determinado.

@TIR

@TIR(*estimación;rango*) calcula la tasa interna de rendimiento (beneficio) prevista para una serie de flujos de caja generados por una inversión. La tasa interna de rendimiento es el porcentaje de interés para el cual se cumple que el valor actual de una serie de flujos de caja que se prevén para una inversión coincide con el valor de la inversión inicial.

Argumentos

estimación puede ser un valor decimal o porcentual que represente el cálculo estimado de la tasa interna de rendimiento. En la mayoría de los casos, *estimación* debería ser un porcentaje entre 0 (0%) y 1 (100%). Cuando se trate de flujos de caja muy elevados, asegúrese de que *estimación* sea un valor lo más preciso posible.

rango puede ser la dirección o el nombre de un rango que contenga los flujos de caja. 1-2-3 considerará los números negativos como pagos y los números positivos como cobros. Por lo general, el valor del primer flujo de caja contenido en *rango* es un número negativo (un pago) que representa la inversión. 1-2-3 asignará el valor 0 a las celdas vacías de *rango* y a las que contienen rótulos, y las incluirá en los cálculos.

Notas

Utilice **@TIR** para determinar la rentabilidad de una inversión. Combine **@TIR** con otras funciones **@** financieras, como **@VAN**, para evaluar una inversión.

1-2-3 supondrá que los flujos de caja se obtienen a intervalos regulares e iguales.

@TIR utiliza aproximaciones sucesivas, partiendo del valor de *estimación*, para calcular la tasa interna de rendimiento. Por ello, deberá empezar especificando en *estimación* un valor que considere razonable como tasa interna de rendimiento. Como puede haber varias soluciones, si el resultado obtenido es menor que 0 o mayor que 1, deberá continuar especificando nuevos valores para *estimación*.

Si **@TIR** no consigue aproximarse al resultado con una precisión de 0,0000001 después de 30 iteraciones de cálculo, dará como resultado **ERR**. Si especifica otros valores como *estimación* y sigue obteniendo **ERR**, utilice **@VAN** para determinar una *estimación* más acertada. Si **@VAN** da como resultado un valor positivo, quiere decir que su *estimación* es demasiado baja; si da como resultado un valor negativo, quiere decir que es demasiado elevada. Cuando la *estimación* sea exacta, dará como resultado 0.

Si obtiene diversas tasas, podrá utilizar **@MEDIA** para determinar la media de la tasa interna de rendimiento.

Ejemplos

Para calcular la tasa interna de rendimiento de una inversión inicial de 10.000\$ seguida de 12 pagos mensuales de 1.500\$ cada uno, se especifica la *estimación* (12,00%) en la celda ESTIMACION. Los pagos están contenidos en el rango denominado FLUJOS_CAJA.

@TIR(ESTIMACION;FLUJOS_CAJA) dará como resultado 10,45%, la tasa interna de rendimiento.

Funciones @ similares

@VAN calcula el valor actual neto de una serie de flujos de caja futuros. **@VALACT** y **@VALACTUAL** calculan el valor actual de una inversión constituida por una serie de pagos iguales. **@VALFUT** y **@VALFUTURO** calculan el valor futuro de una inversión. **@TASA** da como resultado el tipo de interés compuesto necesario para que una inversión alcance en el futuro un valor determinado.

@TIRM calcula la tasa interna de rendimiento modificada. **@TIRX** da como resultado la tasa interna de rendimiento para una serie de flujos de caja de entrada y de salida .

@ISAAP, @ISAPP, @ESMACRO

@ISAAP(*nombre*) comprueba si *nombre* es una función complementaria definida de LotusScript. Si lo es, **@ISAAP** dará como resultado 1 (verdad); si no lo es, dará como resultado 0 (falso).

@ISAPP(*nombre*) comprueba si *nombre* es un programa complementario que se encuentra cargado en memoria en el momento actual. Si lo es, **@ISAPP** dará como resultado 1 (verdad); si no lo es, dará como resultado 0 (falso).

@ESMACRO(*nombre*) comprueba si *nombre* es una subrutina global de LotusScript complementaria definida. Si lo es, **@ESMACRO** dará como resultado 1 (verdad); si no lo es, dará como resultado 0 (falso).

Argumentos

nombre es el nombre del archivo de libro complementario (excluyendo la extensión .12A), o de la función o subrutina que se desee comprobar; debe especificarse como texto.

Notas

@ISAPP da como resultado 1 (verdad) sólo cuando el programa complementario se ha cargado utilizando el comando **Archivo - Complementos - Administrar complementos**. **@ISAPP** arrojará dicho resultado para cualquier programa complementario, incluso si contiene solamente funciones **@**.

@ISAAP y **@ESMACRO** darán como resultado el valor verdad si se especifica una función únicamente por el nombre, y si dicha función está en el libro de trabajo actual o en un programa complementario que esté cargado. Para efectuar la comprobación con una función que se encuentre en otro libro, deberá especificarse tanto el nombre del archivo como de la función, como, por ejemplo, "<<miarchivo>>mifunción."

1-2-3 reconocerá las funciones y las subrutinas solamente si sus argumentos y los tipos de los resultados están permitidos. Para números en coma flotante, 1-2-3 no reconocerá el tipo sencillo: todas las funciones y subrutinas deberán utilizar el tipo doble.

Los rangos son argumentos válidos, pero deberán especificarse como argumentos de tipo variante. Los nombres de objetos de 1-2-3 no son tipos de argumentos permitidos.

Ejemplos

@ISAAP("grados") = 1 si GRADOS es una función de scripts complementaria definida.

@ISAAP("bsuma") = 0, porque **@BSUMA** es una función **@** de 1-2-3, no una función de scripts complementaria.

@ISAPP("finanzas") = 1 si el programa complementario FINANZAS se encuentra cargado en memoria.

@ESMACRO("nomina") = 1 si {NOMINA} es una subrutina de scripts complementaria y definida.

@ESERR

@ESERR(x) comprueba si *x* es el valor **ERR**. Si es así, **@ESERR** dará como resultado 1 (verdad); en caso contrario, dará como resultado 0 (falso) .

Argumentos

x puede ser cualquier valor, posición, texto o condición.

Notas

Utilice **@ESERR** para bloquear los errores derivados de la división entre cero. Por ejemplo, la fórmula **@SI(@ESERR(A1/A2),0,A1/A2)** comprobará el resultado de la división A1/A2 (el contenido de la celda A1 dividido entre el contenido de la celda A2). Si el resultado es el valor **ERR**, la fórmula dará como resultado el valor 0. Si el resultado es cualquier otro valor, la fórmula dará como resultado dicho valor.

Ejemplos

La subrutina **CMPCANT** está formada por tres subrutinas más pequeñas que comprueban los datos insertados en las celdas **CANTIDAD** y **PRECIO**. **CMPCANT** comprobará si la celda **CANTIDAD** contiene un valor; si es así, el control se transferirá a la subrutina **ERRORND**. Si **CANTIDAD** no contiene un valor, **NUECANT** solicitará al usuario que indique una nueva entrada y el control se transferirá a **CMPCANT**.

ERRORND utilizará **@ESERR** para comprobar si **CANTIDAD** contiene el valor **ERR**; si es así, dará como resultado 1 (verdad) y solicitará un nuevo valor. Si **CANTIDAD** no contiene el valor **ERR** y **PRECIO** no contiene el valor **ND**, la subrutina multiplicará los valores contenidos en las dos celdas e introducirá el resultado en la celda denominada **TOTAL**.

```
...
CMPCANT {SI @ESNUM(CANTIDAD)}{BIFURCAR ERRORND}
NUECANT {LEERNUMERO "Indique la cantidad: ";CANTIDAD}
        {BIFURCAR CMPCANT}
ERRORND {SI @ESERR(CANTIDAD)}{BIFURCAR NUECANT}
        {SI @ESND(PRECIO)}{LEERNUMERO "Indique el nuevo precio: ";PRECIO}{BIFURCAR
CMPCANT}
        {IR}TOTAL~+CANTIDAD*Precio~
...

```

Funciones @ similares

@ESND comprueba la presencia del valor **ND**.

@ESND

@ESND(x) comprueba si *x* es el valor ND; si es así, **@ESND** dará como resultado 1 (verdad); en caso contrario, dará como resultado 0 (falso).

Argumentos

x puede ser cualquier valor, posición, texto o condición.

Ejemplos

La subrutina **CMPCANT** está formada por tres subrutinas más pequeñas que comprueban los datos insertados en las celdas **CANTIDAD** y **PRECIO**. **CMPCANT** comprobará si la celda **CANTIDAD** contiene un valor; si es así, el control se transferirá a la subrutina **ERRORND**. Si **CANTIDAD** no contiene un valor, **NUECANT** solicitará al usuario que indique una nueva entrada y el control se transferirá a **CMPCANT**.

ERRORND utilizará **@ESND** para comprobar si **PRECIO** contiene el valor **ND**; si es así, dará como resultado 1 (verdad) y solicitará un nuevo valor. Si **PRECIO** no contiene el valor **ND** y **CANTIDAD** no contiene el valor **ERR**, la subrutina multiplicará los valores contenidos en las dos celdas y depositará el resultado en la celda denominada **TOTAL**.

```
...
CMPCANT  {SI @ESNUM(CANTIDAD)}{BIFURCAR ERRORND}
NUECANT  {LEERNUMERO "Indique la cantidad: ";CANTIDAD}
          {BIFURCAR CMPCANT}
ERRORND  {SI @ESERR(CANTIDAD)}{BIFURCAR NUECANT}
          {SI @ESND(PRECIO)}{LEERNUMERO "Indique el nuevo precio: ";PRECIO}{BIFURCAR
CMPCANT}
          {IR}TOTAL~+CANTIDAD*Precio~
...

```

Funciones @ similares

@ESERR comprueba la presencia del valor **ERR**.

@ESNUM

@ESNUM(x) comprueba si x es un valor. Si es un valor, **ND**, **ERR** o una celda vacía, **@ESNUM** dará como resultado 1 (verdad); si es texto, o una celda que contiene un rótulo o una fórmula cuyo resultado sea un rótulo, **@ESNUM** dará como resultado 0 (falso).

Argumentos

x puede ser cualquier valor, posición, texto o condición. Si x es un rango de varias celdas, **@ESNUM** dará como resultado 0 (falso), aunque la primera celda del rango contenga un valor.

Ejemplos

La subrutina **CMPCANT** está formada por tres subrutinas cortas que comprueban las entradas de las celdas denominadas **CANTIDAD** y **PRECIO**.

CMPCANT utilizará **@ESNUM** para comprobar si **CANTIDAD** contiene un valor; si es así, el control del proceso se transferirá a la subrutina **ERRORND**. Si **CANTIDAD** no contiene un valor, **NUECANT** solicitará al usuario que indique una nueva entrada.

```
...
CMPCANT  {SI @ESNUM(CANTIDAD)}{BIFURCAR ERRORND}
NUECANT  {LEERNUMERO "Indique la cantidad: ";CANTIDAD}
         {BIFURCAR CMPCANT}
ERRORND  {SI @ESERR(CANTIDAD)}{BIFURCAR NUECANT}
         {SI @ESND(PRECIO)}{LEERNUMERO "Indique el nuevo precio: ";PRECIO}{BIFURCAR
CMPCANT}
         {IR}TOTAL~+CANTIDAD*Price~
...

```

Funciones @ similares

@ESCADENA comprueba la presencia de rótulos. **@CELDA** y **@INDICADORCELDA** también pueden determinar si una celda contiene un valor o un rótulo.

@ESRANGO

@ESRANGO(*rango*) comprueba si *rango* corresponde a un nombre de rango definido o a una dirección de rango válida (con letras de hoja y columna de A a IV y números de fila de 1 a 8192). Si *rango* es un nombre de rango definido o una dirección de rango válida, @ESRANGO dará como resultado 1 (verdad); en caso contrario, dará como resultado 0 (falso).

Argumentos

rango puede ser una cadena de texto o una dirección de rango.

Notas

@ESRANGO resulta útil combinado con @SI para determinar si una entrada es un nombre de rango válido para realizar bifurcaciones y llamadas a subrutinas con {REMITIR}.

La función @ESRANGO sólo podrá utilizarse con libros que estén cargados en memoria.

Ejemplos

@ESRANGO(A1) = 1 (verdad).

@ESRANGO(+A1) = 0 (falso).

@ESRANGO(A1..C3) = 1 (verdad).

@ESRANGO(VENTAS) = 1 (verdad), si VENTAS es un nombre de rango definido.

@ESRANGO(PRECIO) = 0 (falso), si PRECIO es un nombre de rango no definido.

@ESRANGO(3) = 0 (falso).

@ESRANGO("COMISION") = 0 (falso) porque el argumento es texto encerrado entre comillas (" ").

@ESCADENA

@ESCADENA(x) comprueba si x es una cadena de texto o un rótulo. Si x es una cadena de texto, una celda que contiene un rótulo o una fórmula cuyo resultado es un rótulo, **@ESCADENA** dará como resultado 1 (verdad); si x es un valor, **ERR**, **ND**, o una celda vacía, **@ESCADENA** dará como resultado 0 (falso).

Argumentos

x puede ser cualquier valor, posición, texto o condición.

Ejemplos

La subrutina **CMPCAD** comprueba el contenido de la celda denominada **CLIENTE**. Si **CLIENTE** contiene un rótulo (**@ESCADENA(CLIENTE) = 1**), la subrutina llama a una nueva subrutina denominada **PEDIDO**. Si **CLIENTE** no contiene ningún rótulo, la subrutina solicita al usuario que especifique una nueva entrada.

```
...
CMPCAD      {SI @ESCADENA(CLIENTE)}{BIFURCAR PEDIDO}
            {LEERROTULO "Indique el nombre del CLIENTE: ",CLIENTE}
            {CMPCAD}
...
```

Funciones @ similares

@ESNUM comprueba la presencia de valores. **@CELDA** y **@INDICADORCELDA** también pueden determinar si una celda contiene un valor o un rótulo.

@CURTOSIS

@CURTOSIS(*rango*;*[tipo]*) calcula el coeficiente de curtosis o de apuntamiento para los valores de un *rango*.

Argumentos

rango es el nombre o la dirección de un rango que contiene valores. Si *rango* contiene menos de cuatro valores, @CURTOSIS dará como resultado el valor ERR.

tipo es un argumento opcional que permite especificar si 1-2-3 debe realizar el cálculo para una población o para una muestra

<u><i>tipo</i></u>	<u>1-2-3 calcula</u>
0	El coeficiente de curtosis para una población; es el valor predeterminado si se omite este argumento
1	El coeficiente de curtosis para una muestra

Notas

El coeficiente de curtosis mide la concentración de una distribución con respecto a la media. Un coeficiente de curtosis positivo revelará la presencia de una distribución relativamente concentrada (leptocúrtica); un coeficiente de curtosis negativo revelará la presencia de una distribución relativamente dispersa (platicúrtica).

Ejemplos

El rango DATOS contiene los siguientes valores: 2, 5, 5, 9, 1, 2, 4.

@CURTOSIS(DATOS;1) = 1,021488

@CURTOSIS(DATOS) = -0,32438

Funciones @ similares

@ASIMETRIA calcula el coeficiente de asimetría para los valores de un rango.

@MAYOR

@MAYOR(*rango*;n) halla el *enésimo* valor más elevado de los comprendidos en *rango*.

Argumentos

rango es el nombre o la dirección de un rango que contiene valores.

n puede ser cualquier entero positivo. Si *n* es mayor que el número de valores existentes en *rango*, **@MAYOR** dará como resultado ERR.

Ejemplos

Un rango denominado CALIFICACIONES contiene los siguientes valores: 87, 85, 90, 80, 82, 92, 79, 85, 95, 86.

@MAYOR(CALIFICACIONES;3) dará como resultado 90, la tercera calificación más alta.

Funciones @ similares

@MENOR halla el *enésimo* valor más pequeño de los comprendidos en un rango. **@MAX** y **@MAXPURA** hallan el valor máximo entre los contenidos en un rango. **@MIN** y **@MINPURA** hallan el valor mínimo entre los contenidos en un rango.

@IZDA

@IZDA(cadena;n) da como resultado los *n* primeros caracteres de *cadena*.

Argumentos

cadena puede ser texto encerrado entre comillas (" "), una fórmula cuyo resultado sea texto, o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

n puede ser un número entero positivo o el valor 0. Si *n* es 0, el resultado será una cadena vacía. Si *n* es mayor que la longitud de *cadena*, **@IZDA** dará como resultado la cadena completa.

Notas

@IZDA cuenta los signos de puntuación y los espacios como caracteres.

@IZDA resulta útil para copiar sólo parte de un rótulo en otra celda comenzando por el principio del rótulo (por ejemplo, para separar títulos como Sr. o Sra. de sus nombres correspondientes).

@IZDA puede utilizarse en una macro para extraer parte de los rótulos insertados por el usuario y almacenarlos en una base de datos, utilizarlos en llamadas a subrutinas o para alterar la macro en sí misma.

Utilice **@IZDA** junto con **@BUSCAR** cuando desconozca el valor exacto de *n* o éste pueda variar.

Ejemplos

@IZDA(C_POSTAL;2) = la parte correspondiente a la provincia del código postal español contenido en la celda C_POSTAL.

@IZDA(A1;**@BUSCAR**("*";A1;0)) = el primer nombre de la celda A1 (por ejemplo, Roberto si la celda A1 contiene el nombre Roberto Sáinz). El asterisco(*) representa un espacio.

Funciones @ similares

@MED da como resultado un número de caracteres de *cadena*. **@DERECHA** da como resultado los últimos *n* caracteres de *cadena*.

@LONGITUD

@LONGITUD(*cadena*) cuenta el número de caracteres de *cadena*.

Argumentos

cadena puede ser texto encerrado comillas (" "), una fórmula cuyo resultado sea texto, o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

Notas

@LONGITUD cuenta los signos de puntuación y los espacios como caracteres.

Utilice @LONGITUD junto con @SUPRESP para hallar la longitud de *cadena* sin incluir los espacios iniciales, finales ni los espacios múltiples situados entre dos palabras.

@LONGITUD resulta útil también en cualquier aplicación en la que los rótulos deban tener una longitud determinada, como en el caso de los códigos postales o en los números de orden de compra.

Ejemplos

@LONGITUD("fiscal") = 6.

@LONGITUD(A5&G12) = el número total de caracteres contenidos en las celdas A5 y G12.

@LONGITUD(@SUPRESP(" Sr. Sáenz ")) = 9.

@LN

@LN(x) calcula el logaritmo neperiano (en base e) de x.

Argumentos

x debe ser un valor mayor que 0.

Notas

Un logaritmo neperiano es aquél que utiliza como base el número e (aproximadamente 2,718282).

Ejemplos

@LN(2) = 0,693147

@LN(@EXP(1)) = 1, porque @EXP(1) = 2,718282.

Funciones @ similares

@EXP es la función inversa de @LN. @LOG calcula el logaritmo decimal (en base 10) de x.

@LOG

@LOG(x) calcula el logaritmo decimal (en base 10) de x .

Argumentos

x debe ser un valor mayor que 0.

Ejemplos

$10^{(@LOG(8)/3)} = 2$, la raíz cúbica de 8.

@LOG(B3) = 0,60206, si la celda B3 contiene el valor 4.

Funciones @ similares

@LN calcula el logaritmo neperiano (en base e) de un valor.

@MINUSC

@MINUSC(*cadena*) convierte en minúsculas todas las letras contenidas en *cadena*.

Argumentos

cadena puede ser texto encerrado entre comillas (" "), una fórmula cuyo resultado sea texto, o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

Notas

Si se utiliza un orden de clasificación ASCII (que se establece a través del comando **Archivo - Opciones del usuario - Preferencias de 1-2-3**, con la opción "**Orden de clasificación regional**" de la pestaña **General**), la utilización de las mayúsculas y minúsculas afectará al orden de clasificación de los rótulos, de modo que dos rótulos aparentemente idénticos pueden no aparecer juntos tras ordenar un rango si uno de ellos contiene mayúsculas donde el otro contiene minúsculas.

Ejemplos

@MINUSC("Previsión de Ventas") = previsión de ventas

Funciones @ similares

@MAYUSC convierte en mayúsculas todas las letras contenidas en *cadena*. @NOMPROPIO convierte en mayúscula la primera letra de cada palabra contenida en *cadena* y en minúsculas el resto.

@MAX, @MAXPURA

@MAX(*lista*) halla el valor máximo entre los contenidos en *lista*.

@MAXPURA(*lista*) halla el valor máximo entre los contenidos en *lista*, pero ignora las celdas que contienen rótulos.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas numéricas. Utilice separadores de argumentos para separar los elementos de *lista*.

Consulte también Argumentos de las funciones @ estadísticas.

Ejemplos

Un rango denominado PRUEBA contiene los datos siguientes: -5,- 7, -9, -11, Enero.

@MAX(PRUEBA) dará como resultado 0, el valor del rótulo Enero, como valor máximo del rango PRUEBA.

@MAXPURA(PRUEBA) ignora el rótulo Enero y da como resultado -5 como valor máximo del rango PRUEBA.

Funciones @ similares

@MIN y @MINPURA hallan el valor mínimo entre los contenidos en una lista. @BMAX halla el valor máximo entre los contenidos en un campo de aquellos registros de una tabla de base de datos que cumplen determinados criterios. @MAYOR da como resultado el *enésimo* valor más grande de entre los contenidos en una lista.

@MEDIANA

@MEDIANA(*lista*) da como resultado la mediana de los valores comprendidos en *lista*.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas numéricas. Utilice separadores de argumentos para separar los elementos de *lista*.

Consulte también Argumentos de las funciones @ estadísticas.

Notas

Si *lista* contiene un número impar de valores, **@MEDIANA** dará como resultado el valor situado en la parte central de la lista después de ordenarla. Si *lista* contiene un número par de valores, **@MEDIANA** dará como resultado la media aritmética de los dos valores situados en la parte central de la lista después de ordenarla.

Ejemplos

@MEDIANA(5;12;65;82;9) = 12

@MEDIANA(5;12;65;82;9;78) = 38,50

Funciones @ similares

@MEDIANAPURA da como resultado la mediana de *lista*, ignorando las celdas vacías, los rótulos y las fórmulas cuyos resultados sean rótulos.

@MEDIAGEOM calcula la media geométrica de los valores de una lista. **@MEDIARMON** calcula la media armónica de los valores de una lista. **@MEDIA** y **@MEDIAPURA** calculan la media aritmética de los valores de una lista.

@MED

@MED(cadena;posición_inicial;n) extrae *n* caracteres de *cadena*, a partir del carácter situado en *posición_inicial*.

Argumentos

cadena puede ser texto encerrado entre comillas (" "), una fórmula cuyo resultado sea texto, o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

posición_inicial debe ser un número offset. Si *posición_inicial* resulta ser mayor que la longitud de *cadena*, el resultado de **@MED** será una cadena vacía.

n puede ser cualquier entero positivo o 0. Si *n* es 0, el resultado de **@MED** será una cadena vacía. Si *n* es mayor que la longitud de *cadena*, 1-2-3 dará como resultado todos los caracteres desde *posición_inicial* hasta el final de la *cadena*.

Notas

@MED cuenta los signos de puntuación y los espacios como caracteres.

Si no conoce la longitud de *cadena*, utilice un número elevado para *n*; 1-2-3 no tendrá en cuenta los caracteres sobrantes y devolverá la *cadena* completa a partir de *posición_inicial*.

Para extraer parte de un rótulo cuando no conozca su *posición_inicial*, utilice **@MED** junto con **@BUSCAR**.

@MED resulta útil en macros para almacenar partes de rótulos insertados por el usuario, crear llamadas a subrutinas o alterar la propia macro.

Ejemplos

@MED("Saldo medio en cuenta";6;5) = medio.

Funciones @ similares

@IZDA da como resultado los primeros *n* caracteres de *cadena* y **@DERECHA** da como resultado los últimos *n* caracteres de *cadena*.

@MIN, @MINPURA

@MIN(*lista*) halla el valor mínimo entre los contenidos en *lista*.

@MINPURA(*lista*) halla el valor mínimo entre los contenidos en *lista*, pero ignora las celdas que contienen rótulos.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas numéricas. Utilice separadores de argumentos para separar los elementos de *lista*.

Consulte también Argumentos de las funciones @ estadísticas.

Ejemplos

Un rango denominado PRUEBA contiene los datos siguientes: 5, 7, 9, 11, Enero.

@MIN(PRUEBA) da como resultado 0, el valor del rótulo Enero, como valor mínimo del rango PRUEBA.

@MINPURA(PRUEBA) ignora el rótulo Enero y da como resultado 5 como valor mínimo del rango PRUEBA.

Funciones @ similares

@MAX y **@MAXPURA** hallan el valor máximo entre los contenidos en una lista. **@BMIN** halla el valor mínimo entre los contenidos en un campo de aquellos registros de una tabla de base de datos que cumplen determinados criterios.

@MINUTO

@MINUTO(*número_hora*) extrae la parte correspondiente a los minutos, es decir, un valor entre 0 y 59 a partir de *número_hora*.

Argumentos

número_hora debe ser un valor comprendido entre 0,000000 (medianoche) y 0,999988 (11:59:59 PM), ambos inclusive. Por lo general, se utiliza junto con otra función @ que proporciona el *número_hora*.

Notas

La parte correspondiente a los minutos resulta útil para operaciones en las que sólo intervienen minutos, como el tiempo transcurrido desde el comienzo de una aplicación.

Ejemplos

@MINUTO(0,333) = 59 porque 0,333 es el número de hora de 7:59:31.

@MINUTO(**@VHORA**(11;15;45)) = 15 porque 15 es el argumento *minutos* de **@VHORA**(11;15;45).

Funciones @ similares

@HORA extrae la hora y **@SEGUNDO** extrae los segundos correspondientes a un número de hora.

@TIRM

@TIRM(*rango*; *tipo_financiación*; *tipo_reinversión*; [*tipo*]) calcula la tasa interna de rendimiento modificada (beneficio) prevista para una serie de flujos de caja generados por una inversión.

La tasa interna de rendimiento es el porcentaje de interés para el cual se cumple que el valor actual de una serie de flujos de caja que se prevén para una inversión coincide con el valor de la inversión inicial.

Argumentos

rango puede ser la dirección o el nombre de un rango que contenga los flujos de caja. 1-2-3 considerará los números negativos como pagos y los números positivos como cobros. Dicho argumento debe contener al menos un valor positivo y un valor negativo.

Por lo general, el valor del primer flujo de caja contenido en *rango* es un número negativo (un pago) que representa la inversión. 1-2-3 asignará el valor 0 a las celdas vacías de *rango* y a las que contengan rótulos, y las incluirá en los cálculos.

tipo_financiación es el tipo de interés que se paga por el dinero empleado en los flujos de caja.

tipo_reinversión es el tipo de interés que se recibe al reinvertir los flujos de caja.

tipo_financiación y *tipo_reinversión* pueden ser valores cualesquiera.

tipo especifica el período de tiempo que se aplica a los flujos de caja. Debe ser un entero extraído de la tabla siguiente

<u>tipo</u>	<u>Los flujos de caja se producen</u>
0	Al final de cada período; es el valor predeterminado cuando se omite el argumento
1	Al principio de cada período

Notas

Utilice @TIRM para determinar la rentabilidad de una inversión. Combine @TIRM con otras funciones @ financieras, como @VAN, para evaluar una inversión.

1-2-3 interpretará que los flujos de caja se obtienen a intervalos regulares e iguales.

Solamente la versión 5 y las versiones posteriores a ésta admiten el uso del argumento opcional *tipo* para la función @TIRM. Cuando se almacene una fórmula @TIRM que contenga un argumento *tipo* y, a continuación, se abra el archivo en una versión anterior a las indicadas, la fórmula dará como resultado ERR.

Ejemplos

Suponga que es el propietario de una serie de apartamentos. Hace cinco y seis años, respectivamente, tomó prestados créditos de 100.000\$ al 9,5% de interés anual para comprar los apartamentos. La siguiente relación, que se encuentra almacenada en el rango INGRESOS, contiene los importes de las inversiones iniciales y los ingresos posteriores por el alquiler de los apartamentos, expresados en dólares:

-100.000

-100.000

45.500

47.000

48.500

50.000

50.000

Durante estos cinco años ha obtenido un 11% anual al reinvertir los beneficios.

@TIRM(INGRESOS;0,095;0,11) = 7,96%

La tasa de rendimiento es ligeramente menor cuando la inversión se realiza al principio del año.

@TIRM(INGRESOS;0,095;0,11;1) = 7,70%

Funciones @ similares

@TIR calcula la tasa interna de rendimiento. @TIRX da como resultado la tasa interna de rendimiento para una serie de flujos de caja (cobros y pagos).

@RESTO, @MODULO

@RESTO(x;y) y @MODULO (x;y) calculan el resto (módulo) de x/y .

Argumentos

x es un valor. Si x es 0, @RESTO y @MODULO darán como resultado 0.

y es un valor distinto de cero.

Notas

El resultado de @RESTO es $x - y * @ENT(x/y)$. El signo del resultado (+ o -) será siempre el mismo que el signo x.

El resultado de @MODULO es $x - y * @REDONDABAJO(x/y)$. El signo del resultado (+ o -) será siempre el mismo que el signo de y.

Ejemplos

$$@RESTO(9;4) = 1$$

$$@MODULO(9;4) = 1$$

$$@RESTO(-14;3) = -2$$

$$@MODULO(-14;3) = 1$$

Funciones @ similares

@COCIENTE calcula el resultado de x/y , y trunca la parte decimal del resultado para convertirlo en un entero.

@MES

@MES(*número_fecha*) extrae la parte correspondiente al mes (un número del 1 al 12) a partir de *número_fecha*.

Argumentos

número_fecha debe ser un valor comprendido entre 1 (1 de enero de 1900) y 73050 (31 de diciembre del 2099), ambos inclusive.

Notas

Pueden utilizarse otras funciones @ de fecha para obtener el valor *número_fecha*.

Ejemplos

@MES(**@FECHA**(91;3;27)) = 3 porque 3 es el argumento *mes* de **@FECHA**(91;3;27).

@MES(20181) = 4 porque el número de fecha 20181 corresponde al 02-Abr-55.

@MES(**@ACTUAL**) = el mes actual.

Funciones @ similares

@DIA calcula el día a partir de un número de fecha. **@AÑO** calcula el año a partir de un número de fecha.

Ejemplo de @REGRESION

Suponga que es el encargado de una heladería en un centro turístico y que desea saber si se podría predecir el número de litros de helado que se venderán un día determinado. A su entender, las ventas dependen de tres factores principales: el número de horas de sol, la temperatura a mediodía (en grados centígrados) y el número de autocares que aparcan en un aparcamiento cercano. Para determinar la correlación existente entre estos tres factores y el volumen de ventas, ha recogido los datos durante un período de seis días y los ha insertado en una hoja de trabajo.

A	A	B	C	D
1	Litros vendidos	Sol	Temp	Autocares
2	250	3	29	10
3	545	5	33	7
4	550	5	32	8
5	450	6	29	10
6	605	6	32	11
7	615	7	31	9

@REGRESION(B2..D7;A2..A7;3) = 0,987035

Dado que el valor de r2 es muy cercano a 1, puede tener la certeza de que existe una correlación alta entre las ventas de helados, las horas de sol, la temperatura y el número de autocares.

@N

@N(*rango*) arroja como resultado el contenido de la primera celda de *rango* en forma de valor. Si la celda contiene un rótulo, **@N** dará como resultado el valor 0.

Argumentos

rango puede ser una dirección de rango o de celda, o un nombre de rango.

Notas

@N puede utilizarse junto con otra función **@** o con una fórmula para comprobar si una celda contiene un rótulo cuando debe contener un valor. De esta forma, puede evitarse que las fórmulas den como resultado ERR.

@N resulta útil también en las macros para comprobar los datos insertados por el usuario.

Ejemplos

+100+**@N**(B5..F5) = 885, si la celda B5 contiene el valor 785.

@N(A5)+**@N**(B5) dará como resultado 785, si A5 contiene un rótulo y B5 contiene el valor 785.

Funciones @ similares

@C arroja como resultado el contenido de la primera celda de un rango, en forma de rótulo. **@ESNUM** comprueba si una celda contiene un valor.

@ND

@ND arroja como resultado el valor ND (no disponible).

Notas

Utilice @ND durante la creación de una hoja de trabajo que vaya a contener datos todavía por determinar. Podrá utilizar @ND para señalar las celdas en las que vayan a incluirse los datos, de manera que las fórmulas que hagan referencia a dichas celdas devuelvan el valor ND mientras no se especifiquen los datos correctos.

@ND resulta útil también para averiguar qué fórmulas dependen de un celda determinada.

Ejemplos

@SI(@CELDA("tipo";B14)="b";@ND;B14) dará como resultado el valor ND cuando la celda B14 esté vacía.

Funciones @ similares

@ERR da como resultado el valor ERR. @ESND comprueba la presencia del valor ND.

@ACTUAL

@ACTUAL calcula el número correspondiente a la fecha y hora actuales según el reloj del PC. Este valor está formado por un número de fecha (parte entera) y un número de hora (parte decimal).

Notas

Utilice @ACTUAL junto con las teclas F2 (EDICION) y F9 (CALC) para crear un registro fijo de la fecha y hora actuales, con objeto de que aparezcan en una hoja o de utilizarlo en cálculos para averiguar el tiempo transcurrido.

El valor obtenido con @ACTUAL admite cualquiera de los formatos de fecha y hora existentes. Si se le asigna un formato de fecha, 1-2-3 mostrará únicamente en pantalla la parte correspondiente a la fecha (parte entera). Si se le asigna un formato de hora, 1-2-3 mostrará sólo la parte correspondiente a la hora (parte decimal). En ambos casos, 1-2-3 almacenará y utilizará en los cálculos el número completo de fecha y hora.

1-2-3 recalculará la función @ACTUAL cada vez que se recalcule la hoja.

Ejemplos

@ACTUAL = 31050,5 a las 12:00 PM del 3 de enero de 1985.

@ACTUAL = 33418,395 a las 9:28 AM del 29 de junio de 1991.

Funciones @ similares

@HOY calcula el número de fecha correspondiente a la fecha actual.

@VAN

@VAN(*interés*; *rango*; [*tipo*]) calcula el valor actual neto de una serie de flujos de caja futuros (contenidos en *rango*) actualizados a un tipo de *interés* periódico fijo.

Argumentos

interés es un valor decimal o porcentual mayor que -1.

rango es el rango que contiene los flujos de caja.

tipo determina el período de tiempo para los flujos de caja. Debe ser un entero de los especificados en la tabla siguiente:

<u><i>tipo</i></u>	<u>Los flujos de caja se producen</u>
0	Al final de cada período; es el valor predeterminado cuando se omite este argumento
1	Al principio de cada período

Notas

Utilice @VAN para evaluar una inversión o comparar varias inversiones entre sí. @VAN calcula la inversión inicial necesaria para alcanzar un flujo de caja determinado a un tipo de interés dado.

@VAN dará como resultado ERR si *rango* contiene más de una fila o más de una columna. Por ejemplo, @VAN dará como resultado ERR si *rango* es A1..D25, pero no lo hará si *rango* es A1..D1 (una sola fila) o A1..A25 (una sola columna).

Solamente la versión 5 y las versiones posteriores a ésta admiten el uso del argumento opcional *tipo* para la función @VAN. Si se almacena una fórmula @VAN que contiene un argumento *tipo* y, a continuación, se abre el archivo en una versión de 1-2-3 anterior a las indicadas, la fórmula dará como resultado ERR.

Ejemplos

En el ejemplo siguiente se utiliza @VAN para descontar o actualizar a dólares actuales una serie de flujos de caja irregulares con un tipo de interés anual del 11,5%.

rango es una lista de flujos de caja, uno por mes durante 12 meses, denominado FLUJOS:

0,00\$
0,00\$
2.500,00\$
2.500,00\$
3.000,00\$
5.000,00\$
6.000,00\$
9.000,00\$
3.000,00\$
2.500,00\$
0,00\$
7.500,00\$

Con objeto de proporcionar a la función @VAN el número de períodos correcto, los meses en los que no haya tenido lugar ningún pago deberán incluirse también en *rango*. Dado que los pagos son mensuales, @VAN precisa que el *interés*, incluido en una celda denominada DESCUENTO, esté expresado en forma de porcentaje mensual:

$$0,115/12 = 0,96$$

$$@VAN(DESCUENTO;FLUJOS) = 38.084,13\$$$

El resultado será diferente si la retirada de fondos se realiza al principio de cada período.

$$@VAN(DESCUENTO;FLUJOS;1) = 38.449,10\$$$

Funciones @ similares

@VALACT calcula el valor actual de una inversión, a partir de una serie de pagos iguales. @VALFUT calcula el valor futuro de una inversión.

@VANX da como resultado el valor actual neto de una serie de flujos de caja (cobros y pagos).

@IMPAR

@IMPAR(x) redondea el valor x al primer número entero impar que encuentre al alejarse de 0.

Argumentos

x puede ser cualquier valor. Si x es un entero impar, @IMPAR dará como resultado x.

Ejemplos

@IMPAR(3,25) = 5

@IMPAR(3) = 3

@IMPAR(-3,25) = -5

Funciones @ similares

@PAR redondea un valor al primer número entero par que encuentre al alejarse de 0. @REDOND,

@REDONDABAJO y @REDONDARRIBA redondean un valor a un número determinado de decimales.

@REDONDMULTP redondea un valor a un múltiplo determinado. @ENT trunca un valor y elimina su parte decimal.

@TRUNC trunca un valor a partir de una posición decimal determinada.

@PERCENTILP

@PERCENTILP(*x*; *rango*) calcula el valor correspondiente al percentil *x* para la muestra contenida en *rango*.

Argumentos

x es el percentil cuyo valor se desea hallar. Puede ser un valor entre 0 y 1.

rango es el nombre o la dirección del rango que contiene los valores que van a analizarse.

1-2-3 asignará el valor 0 a todos los rótulos de *rango* y los incluirá en el cálculo de **@PERCENTILP**. 1-2-3 ignorará las celdas vacías existentes en *rango*.

Ejemplos

Un rango denominado CALIFICACIONES contiene los valores siguientes: 87, 85, 90, 80, 82, 92, 79, 85, 95, 86. Se desea hallar la calificación correspondiente al percentil del 90%.

@PERCENTILP(0,9; CALIFICACIONES) = 92,3

Funciones @ similares

@PERCENTILV halla el percentil en que se encuentra un valor determinado teniendo en cuenta los valores contenidos en un rango.

@DECIL da como resultado un decil especificado, y **@CUARTIL** da como resultado un cuartil especificado.

@PERMUT

@PERMUT($n;r$) calcula el número de secuencias ordenadas (permutaciones) de r objetos que pueden seleccionarse de un conjunto de n objetos.

Argumentos

n puede ser cualquier entero positivo o 0.

r puede ser cualquier entero positivo o 0, y no puede ser mayor que n .

Si n y r no son enteros, 1-2-3 truncará su parte decimal para convertirlos en enteros.

Ejemplos

Se van a realizar exámenes a las horas siguientes: 9:00, 10:00 y 11:00 AM; tres de los cinco profesores de un departamento serán los encargados de vigilarlos. La función @ siguiente calcula el número de formas posibles de asignar los profesores.

@PERMUT(5;3) = 60

Funciones @ similares

@COMBIN calcula el número de subconjuntos de r elementos que pueden realizarse con n elementos, sin tener en cuenta el orden.

@PI

@PI da como resultado el valor π (tomado como 3,14159265358979). El valor π es la relación entre la longitud y el diámetro de una circunferencia.

Ejemplos

@PI = 3,1415926536

@PI*4^2 = 50,26548, el área de un círculo cuyo radio es 4.

@AMORT, @AMORTIZ, @AMORTC

@AMORT(*principal;interés;plazo*) calcula el importe de los pagos periódicos necesarios para amortizar un crédito (*principal*), dado un tipo de *interés* periódico y un número de periodos de pago (*plazo*).

@AMORTIZ(*principal;interés;plazo;[tipo;valor_futuro]*) calcula el importe de los pagos periódicos necesarios para amortizar un crédito (*principal*), dado un tipo de *interés* periódico y un número de periodos de pago (*plazo*).

@AMORTIZ realiza el cálculo para anualidades ordinarias o anticipadas, dependiendo del valor especificado para el argumento *tipo*.

@AMORTC(*principal;interés;plazo*) es una modalidad especial de @AMORT que realiza los cálculos de acuerdo con el método utilizado en Canadá.

Argumentos

principal y *plazo* deben ser valores.

interés debe ser un valor decimal o un porcentaje mayor que -1.

tipo es un argumento opcional que permite especificar si se desea realizar el cálculo para una anualidad ordinaria o para una anualidad anticipada.

<u>tipo</u>	<u>1-2-3 realiza el cálculo para</u>
0	Anualidad ordinaria (los pagos se efectúan al final de cada período); es el valor predeterminado si se omite este argumento
1	Anualidad anticipada (los pagos se efectúan al principio de cada período)

valor_futuro es un argumento opcional que permite especificar el valor futuro de la serie de pagos. Puede ser cualquier valor. Si se omite el argumento *valor_futuro*, 1-2-3 utilizará el 0.

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Notas

Para @AMORT y @AMORTIZ, la unidad de tiempo utilizada para calcular el *interés* debe coincidir con la de los períodos utilizados en *plazo*; por ejemplo, si se efectúan los cálculos para pagos mensuales, el interés y el plazo también deben expresarse en meses. Normalmente, esto supone dividir el tipo de interés por 12 y multiplicar el número de años de *plazo* por 12.

Para la función @AMORTC, la unidad de tiempo utilizada para calcular el *interés* es años, mientras que la correspondiente a *plazo* es meses.

Ejemplos

Suponga que está pensando pedir un crédito de 8.000\$ a pagar en 3 años con un interés anual del 10,5%, compuesto mensualmente. Los pagos se efectuarán el último día de cada mes. Para determinar la cuantía del pago mensual:

@AMORT(8000;0,105/12;36) = 260,02\$

Si los pagos se efectúan el primer día de cada mes:

@AMORTIZ(8000;0,105/12;36;1;-2500) = 19.890\$

Si se calcula el pago mensual utilizando @AMORTC:

@AMORTC(8000;0,105;36) = 25.918\$

Funciones @ similares

@AMORT2 calcula el importe de los pagos periódicos necesarios para amortizar un crédito, dado un tipo de *interés* periódico y un número de periodos de pago, suponiendo una anualidad anticipada. @AMORTI calcula la parte correspondiente al interés de un pago periódico constante. @IPE calcula la parte correspondiente al interés de un pago periódico en el que la parte correspondiente al principal es la misma en cada período.

@AMORTINT calcula la suma de las partes correspondientes al interés de los pagos periódicos de un crédito.

@AMORTPRINC calcula la suma de las partes correspondientes al principal de los pagos periódicos de un crédito.

@PERIODO calcula el número de pagos periódicos de un crédito.

@POISSON

@POISSON(*x*; *media*; [*acumulativo*]) calcula la distribución de Poisson.

Argumentos

x es el número de sucesos observados y debe ser un entero positivo o 0.

media es el número de sucesos esperados y debe ser un entero positivo.

Si *x* y *media* no son enteros, 1-2-3 truncará su parte decimal y los convertirá en enteros.

acumulativo es un argumento opcional que permite indicar a 1-2-3 cómo debe calcular la función @POISSON.

<u>acumulativ o</u>	<u>1-2-3 calcula</u>
0	La probabilidad de que se produzca un número de sucesos igual a <i>x</i> ; es el valor predeterminado si se omite este argumento
1	La probabilidad de que se produzca un número de sucesos menor o igual que <i>x</i>

Notas

@POISSON calcula la distribución de Poisson con un margen de error de $\pm 3 \cdot 10^{-7}$.

@POISSON es útil para predecir el número de sucesos que pueden producirse durante un período de tiempo determinado; por ejemplo, el número de visitantes que pueden entrar en un parque de atracciones en una hora.

Ejemplos

Se espera que pasen seis automóviles por un control de peaje en una hora. Para determinar la probabilidad de que pasen como máximo cuatro automóviles en una hora:

@POISSON(4;6;1) = 0,285057, es decir, un 28,51%

Para determinar la probabilidad de que pasen exactamente cuatro automóviles en una hora:

@POISSON(4;6) = 0,133853, es decir, un 13,39%

@PERCENTILV

@PERCENTILV(*x*; *rango*; [*decimales*]) halla el percentil en que se encuentra el valor *x* teniendo en cuenta los valores contenidos en *rango*.

Argumentos

x puede ser cualquier valor.

rango es el nombre o la dirección del rango que contiene los valores que se van a analizar.

decimales es un argumento opcional que permite especificar el número de decimales al que se desea redondear el resultado de **@PERCENTILV**. Debe ser un valor entre 0 y 100. Si se omite el argumento *decimales*, 1-2-3 utilizará el valor 2.

Notas

Si *x* no es uno de los valores de *rango*, 1-2-3 asignará la posición del percentil del 0% al valor más pequeño de *rango*, la posición del percentil del 100% al valor más elevado de *rango* y realizará una interpolación.

Ejemplos

Un rango denominado CALIFICACIONES contiene los valores siguientes: 87, 85, 90, 80, 82, 92, 79, 85, 95, 86. Se desea hallar el percentil en que se encuentra la calificación 90.

@PERCENTILV(90; CALIFICACIONES) = 0,78, es decir, 78%.

Funciones @ similares

@PERCENTILP calcula el valor correspondiente a un percentil determinado para una muestra contenida en una relación de valores.

@PRECIO

@PRECIO(*negociación*; *vencimiento*; *cupón*; *rentabilidad*; [*amortización*; *frecuencia*; *base*]) calcula el precio por cada 100\$ de valor nominal para activos financieros que proporcionan pagos periódicos de intereses.

Argumentos

negociación es la fecha en que se negocia o transmite el activo. Debe ser un número de fecha.

vencimiento es la fecha de vencimiento o amortización del activo. Debe ser un número de fecha. Si *vencimiento* es menor o igual que *negociación*, @PRECIO dará como resultado ERR.

cupón es el tipo de interés anual del cupón del activo. Puede ser cualquier valor positivo o 0.

rentabilidad es la rentabilidad anual. *rentabilidad* puede ser cualquier valor positivo.

amortización es un argumento opcional que permite especificar el valor de amortización del activo por cada 100\$ de valor nominal. Puede ser cualquier valor positivo o 0. Si se omite el argumento *amortización*, 1-2-3 utilizará el valor 100.

frecuencia es un argumento opcional que permite especificar el número de pagos de cupón por año. Debe ser cualquiera de los valores indicados en la tabla siguiente:

<u><i>frecuenci</i></u> <u><i>a</i></u>	<u>Frecuencia de los pagos de cupón</u>
1	Anual
2	Semestral; es el valor predeterminado si se omite este argumento
4	Trimestral
12	Mensual

base es un argumento opcional que permite especificar el tipo de base para el recuento de los días utilizada. Debe ser cualquiera de los valores indicados en la tabla siguiente:

<u><i>base</i></u>	<u>Base para el recuento de los días</u>
0	30/360; es el valor predeterminado si se omite este argumento
1	Real/real
2	Real/360
3	Real/365
4	Europeo 30/360

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Ejemplos

Un bono, cuyo vencimiento es el 1 de diciembre de 1998, se negocia el 1 de julio de 1993. El tipo de interés del cupón semestral es del 5,50%, su rentabilidad es del 5,61% y el recuento de los días se efectúa de acuerdo con la base 30/360.

Para determinar precio del bono:

@PRECIO(@FECHA(93;7;1);@FECHA(98;12;1);0,055;0,0561;100;2;0) = 99,49\$

Funciones @ similares

@ACUMULADO calcula el interés acumulado para activos financieros que proporcionan pagos periódicos de intereses. @RENTABILIDAD calcula la rentabilidad para activos financieros que proporcionan pagos periódicos de intereses. @DURACION calcula la duración anual, y @DURACIONM calcula la duración anual modificada para activos financieros que proporcionan pagos periódicos de intereses. @PRECIO2 calcula el precio por cada 100 yenes de valor nominal para activos financieros que proporcionan pagos periódicos de intereses, según el sistema utilizado en Japón.

@NOMPROPIO

@NOMPROPIO(*cadena*) convierte en mayúscula la primera letra de cada palabra de *cadena* y en minúsculas el resto de las letras.

Argumentos

cadena puede ser texto encerrado entre comillas (" "), una fórmula cuyo resultado sea texto, o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

Notas

@NOMPROPIO resulta útil cuando se combinan datos procedentes de varias fuentes y se desea mantener la coherencia de los rótulos a lo largo de todo el libro. Conviene utilizar @NOMPROPIO en las bases de datos con el fin de que la inicial de todos los nombres vaya en mayúsculas antes de ordenarlos alfabéticamente o antes de utilizarlos para crear etiquetas postales.

Si se utiliza un orden de clasificación ASCII (que se establece mediante el comando **Archivo - Opciones del usuario - Preferencias de 1-2-3**, en "**Orden de clasificación regional**" de la pestaña **General**), la utilización de mayúsculas y minúsculas afectará al orden de clasificación de los rótulos; dos rótulos aparentemente idénticos pueden no aparecer juntos tras ordenar un rango si uno de ellos contiene mayúsculas donde el otro contiene minúsculas.

Ejemplos

@NOMPROPIO(A7&" ; "&G7) dará como resultado Manuel Santos; Gijón, Asturias si A7 contiene el rótulo MANUEL SANTOS, y G7 contiene el rótulo gijón, asturias. Obsérvese que el punto y coma (;) se encuentra entre comillas, por lo que se considera una cadena de texto en lugar de separador de argumentos.

Funciones @ similares

@MINUSC convierte todas las letras de *cadena* en minúsculas. @MAYUSC convierte todas las letras de *cadena* en mayúsculas.

@VALACT, @VALACTUAL

@VALACT(*pagos;interés;plazo*) calcula el valor actual de una inversión constituida por una serie de *pagos* iguales descontados a un tipo de *interés* dado durante el número de periodos indicado en *plazo*.

@VALACTUAL(*pagos;interés;plazo;[tipo;valor_futuro]*) calcula el valor actual de una inversión que tiene un determinado *valor_futuro* y está constituida por una serie de *pagos* iguales descontados a un tipo de *interés* dado durante el número de periodos indicado en *plazo*. @VALACTUAL realiza el cálculo para anualidades ordinarias o anticipadas, dependiendo del valor especificado para el argumento *tipo*.

Argumentos

pagos y *plazo* deben ser valores.

interés debe ser un valor decimal o un porcentaje mayor que -1.

tipo es un argumento opcional que especifica si 1-2-3 debe realizar el cálculo para una anualidad ordinaria o anticipada.

<u><i>tipo</i></u>	<u>1-2-3 realiza el cálculo para</u>
0	Anualidad ordinaria (los pagos se efectúan al final de cada período); es el valor predeterminado si se omite este argumento
1	Anualidad anticipada (los pagos se efectúan al principio de cada período)

valor_futuro es un argumento opcional que permite especificar el valor futuro de la serie de pagos. Puede ser cualquier valor. Si se omite el argumento *valor_futuro*, 1-2-3 utiliza el valor 0.

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Notas

La unidad de tiempo utilizada para calcular el *interés* deberá coincidir con la de los periodos utilizados en *plazo*; por ejemplo, si se efectúan los cálculos para pagos mensuales, el interés y el plazo también deberán expresarse en meses. Normalmente, esto supone dividir el tipo de interés por 12 y multiplicar el número de años de *plazo* por 12.

Utilice @VALACT para evaluar una inversión o comparar varias inversiones entre sí; por ejemplo, una inversión con un pago único y un fondo de pensiones con una serie de pagos periódicos. También podrá utilizar @VALACT junto con @AMORT para crear tablas de amortización de créditos.

@VALACT es la función complementaria de @AMORT, es decir, @VALACT calcula la cuantía de un crédito cuando existe un límite para el importe de los pagos mensuales que puede permitirse el solicitante, mientras que @AMORT calcula el importe de los pagos mensuales teniendo en cuenta la cuantía del crédito que se desea solicitar.

Ejemplos

Suponga que ha ganado un premio de 1.000.000\$ que se cobrará en 20 pagos anuales de 5.000.000\$ al final de cada uno de los próximos 20 años, pero que también le ofrecen la oportunidad de recibir un pago único de 40.000.000\$ en el momento actual, y que desea calcular cuál de ambas opciones resulta más rentable en dólares actuales.

Se supone que si se aceptan los pagos anuales de 50.000\$, se van invirtiendo a un tipo de interés del 8% capitalizado anualmente.

@VALACT(50000;0,08;20) dará como resultado 49.090.700, lo que quiere decir que 1.000.000\$ pagados en veinte años equivalen a 490,907\$ de hoy.

Si recibiera los pagos al principio de cada año:

@VALACTUAL(50000;0,08;20;1) = 530,180\$

Funciones @ similares

@VALFUT y @VALFUTURO calculan el valor futuro de una inversión constituida por una serie de pagos iguales.

@VAN calcula el valor actual neto de una serie de inversiones, descontando el valor futuro al valor actual. @AMORT y @AMORTIZ calculan el importe de los pagos periódicos necesarios para amortizar un crédito a un tipo de interés dado durante un determinado número de periodos de pago.

@VALACT2 calcula el valor actual de una inversión, a partir de una serie de pagos iguales, descontado a un tipo de interés dado durante el número de periodos indicado en plazo, suponiendo una anualidad anticipada.

@VALACTGLOBAL da como resultado el valor actual de una suma fija que se percibirá en el futuro tras un

determinado número de períodos y que se actualizará a un tipo dado de interés.

@COCIENTE

@COCIENTE(x;y) calcula el cociente de x/y y trunca la parte decimal del resultado para convertirlo en un entero.

Argumentos

x puede ser cualquier valor. Si x es igual a 0, @COCIENTE dará como resultado 0.

y puede ser cualquier valor distinto de 0.

Ejemplos

@COCIENTE(7;3) = 2

@COCIENTE(12,25;3,5) = 3

@COCIENTE(-7;3) = -2

Funciones @ similares

@RESTO calcula el resto (módulo) de x/y .

@RADAGRAD

@RADAGRAD(*radianes*) convierte *radianes* en grados.

Argumentos

radianes es un valor.

Ejemplos

@RADAGRAD(0,523599) = 30 grados

Funciones @ similares

@GRADARAD convierte grados en radianes.

@ALEAT

@ALEAT genera un número aleatorio entre 0 y 1. El número obtenido por @ALEAT tiene una precisión de 15 decimales. Cada vez que 1-2-3 recalcula la hoja, @ALEAT genera un nuevo número aleatorio.

Notas

Si desea convertir el valor generado por @ALEAT en un valor fijo, presione F2 (EDICION) y después F9 (CALC). Para obtener un número aleatorio comprendido entre 0 y otro número, multiplique @ALEAT por dicho número. Para obtener números enteros aleatorios utilice @REDOND o @ENT junto con @ALEAT.

Ejemplos

@ALEAT = 0,419501 o cualquier valor comprendido entre 0 y 1.

@ALEAT*10 = 6,933674 o cualquier valor comprendido entre 0 y 10.

@ENT(@ALEAT*50)+1 = 49 o cualquier entero comprendido entre 1 y 50.

@POSICION

@POSICION(*elemento*; *rango*; [*orden*]) calcula la posición relativa o el tamaño de un valor de un rango con relación a los restantes valores del rango.

Argumentos

elemento es el valor cuya posición se desea determinar.

rango es la dirección o el nombre del rango que contiene los valores. El argumento *elemento* debe encontrarse dentro de *rango*.

orden es un argumento opcional que permite especificar el modo en que 1-2-3 debe ordenar el *elemento*. Puede tomar uno de los dos valores siguientes:

<u>orden</u>	<u>1-2-3 considera que los valores de <i>rango</i> están en</u>
0	Orden descendente (del 9 al 1) antes de buscar la posición que ocupa <i>elemento</i> ; es el valor predeterminado si se omite este argumento
1	Orden ascendente (del 1 al 9) antes de buscar la posición que ocupa <i>elemento</i>

Notas

1-2-3 asignará la misma posición a los números que se encuentren repetidos en *rango*. La presencia de números repetidos afectará a la posición de los números de *rango* situados a continuación. Por ejemplo, en el caso de los valores 2, 4, 6, 8, 8, 10, 12, el número 8 aparece dos veces y ocupa la cuarta posición en orden ascendente; el número 10 ocupa la sexta posición en orden ascendente, pero ninguno de los números ocupa la quinta posición.

Ejemplos

El rango denominado VENTAS (A1..A5) contiene los valores siguientes:

5.000\$
4.900\$
5.150\$
4.800\$
4.900\$

@POSICION(4900;VENTAS) = 3, ya que 4.900 es el tercer valor más grande del rango VENTAS. No hay ningún valor que ocupe la cuarta posición.

@POSICION(4900;VENTAS;1) = 2; ya que si VENTAS se clasifica en orden ascendente, 4.900 es el segundo valor más pequeño del rango VENTAS. No hay ningún valor que ocupe la tercera posición.

@TASA

@TASA(*valor_futuro*; *valor_actual*; *plazo*) calcula el tipo de interés periódico necesario para que una inversión (*valor_actual*) alcance un *valor_futuro* al cabo del número de periodos de capitalización especificados en *plazo*.

Argumentos

valor_futuro, *valor_actual* y *plazo* son valores.

Ejemplos

Se han invertido 10.000\$ en un bono con vencimiento a cinco años, valor de vencimiento de 18.000\$ y un interés compuesto mensualmente. Para determinar el tipo de interés de esta inversión se utilizará la siguiente función:

@TASA(18000;10000;60), cuyo resultado, 0,984%, es el tipo de interés periódico (mensual). Para determinar el tipo de interés anual, utilice la fórmula $((1+\text{@TASA}(18000;10000;60))^{12})-1$, que dará como resultado un tipo de interés anual del 12,47%.

Funciones @ similares

@TASAINI calcula el tipo (o tasa) de interés necesario para que la anualidad aumente hasta un valor futuro determinado.

@CONVERTREF

@CONVERTREF(*referencia*) convierte las letras de la A a la IV de las columnas o de las hojas de 1-2-3 en los números del 1 al 256 correspondientes, y viceversa.

Argumentos

referencia corresponde a una hoja o una columna de 1-2-3 y puede ser una letra entre A y IV, especificada como texto, o bien un entero del 1 al 256.

@CONVERTREF no tendrá en cuenta si las letras de *referencia* se especifican en mayúsculas o en minúsculas.

Ejemplos

@CONVERTREF(10) = J

@CONVERTREF("J") = 10

Funciones @ similares

@COLS cuenta el número de columnas existentes en un rango y **@HOJAS** cuenta el número de hojas existentes en un rango. **@COORD** crea una dirección de celda a partir de los valores especificados.

@REGRESION

@REGRESION(*rango_x*; *rango_y*; *atributo*; [*calcular*]) realiza un análisis de regresión lineal múltiple y da como resultado el parámetro de la regresión especificado.

Argumentos

rango_x contiene las variables independientes y representa el nombre o la dirección de un rango que puede contener un máximo de 75 columnas y 8.192 filas.

rango_y contiene el conjunto de valores correspondientes a la variable dependiente y representa el nombre o la dirección de un rango que tiene una sola columna y el mismo número de filas que *rango_x*.

atributo permite especificar cuál de los parámetros de la regresión se desea calcular. Puede ser cualquiera de los valores siguientes:

<i>atributo</i>	1-2-3 calcula
1	Constante
2	Error típico de estimación Y
3	r ² (coeficiente de determinación)
4	Número de observaciones
5	Grados de libertad
101 a 175	Coeficiente de X (pendiente) para la variable independiente especificada mediante <i>atributo</i>
201 a 275	Error típico del coeficiente para la variable independiente especificada mediante <i>atributo</i>

Para los dos últimos atributos, 1-2-3 numerará las variables independientes del *rango_x* de izquierda a derecha y de arriba abajo, comenzando por el número 1.

Por ejemplo, si el *rango_x* es B2..D7, utilice el atributo 201 para hallar el error típico del coeficiente correspondiente a la variable independiente situada en la columna B; emplee el atributo 102 para hallar el coeficiente de X para la variable independiente situada en la columna C.

calcular es un argumento opcional que permite especificar si se ha de calcular el punto de intersección con el eje Y.

<i>calcular</i>	1-2-3
0	Supone que la línea corta al eje Y en el origen de coordenadas (0,0)
1	Calcula el punto de intersección con el eje Y; es el valor predeterminado si se omite este argumento

Notas

Para los mismos datos, la función @REGRESION y Rango - Análisis matemático - Regresión darán el mismo resultado.

Ejemplos

@REGRESION

Funciones @ similares

@ESTIMACION da como resultado un valor estimado para x a partir de la tendencia lineal existente entre los valores del *rango_y* y los del *rango_x*.

@REPETIR

@REPETIR(*cadena;n*) reproduce *cadena* el número de veces especificado en *n*.

Argumentos

cadena puede ser un texto encerrado entre comillas (" "), una fórmula cuyo resultado sea texto, o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

n puede ser cualquier entero positivo.

Notas

@REPETIR reproducirá la cadena tantas veces como se indique, aunque se sobrepase la anchura de la columna actual. Esta posibilidad es la que diferencia esta función @ del carácter de prefijo de rótulo barra invertida (\), que repite un rótulo hasta llenar la columna actual.

Ejemplos

@REPETIR("Hola ";3) da como resultado Hola Hola Hola.

@REPETIR("-";10) da como resultado -----.

@SUSTITUIR

@SUSTITUIR(*cadena_original*; *posición_inicio*; *n*; *cadena_nueva*) sustituye *n* caracteres de *cadena_original* por *cadena_nueva*, a partir de *posición_inicio*.

Argumentos

cadena_original y *cadena_nueva* pueden ser cadenas de texto encerradas entre comillas (" "), fórmulas cuyo resultado sea texto, o direcciones o nombres de celdas que contengan rótulos o fórmulas cuyos resultados sean rótulos.

posición_inicio es el número offset de un carácter de *cadena_original*. Puede ser cualquier valor positivo o 0. Si *posición_inicio* es mayor que la longitud de *cadena_original*, **@SUSTITUIR** agregará *cadena_nueva* al final de *cadena_original*.

n puede ser cualquier entero positivo o 0. Si *n* es 0, **@SUSTITUIR** insertará la *cadena_nueva* en *posición_inicio* sin eliminar ningún carácter de *cadena_original*.

Notas

@SUSTITUIR cuenta los espacios y los signos de puntuación como caracteres. Si utiliza **@SUSTITUIR** para agregar o insertar texto, no olvide incluir los espacios necesarios.

Utilice **@BUSCAR** junto con **@SUSTITUIR** para buscar y sustituir un rótulo o para encontrar una *posición_inicio* desconocida.

@SUSTITUIR resultará útil cuando se necesite sustituir un grupo de caracteres por otro, como, por ejemplo, cuando varíen ciertas cifras de los números de teléfono de una base de datos.

Ejemplos

@SUSTITUIR(CELDA;**@BUSCAR**("-";CELDA;0),1,"/") cambia el rótulo contenido en CELDA, 4-24, por 4/24.

@DERECHA

@DERECHA(*cadena*; *n*) arroja como resultado los últimos *n* caracteres de *cadena*.

Argumentos

cadena puede ser texto encerrado entre comillas (" "), una fórmula cuyo resultado sea texto, o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

n puede ser un entero positivo o 0. Si *n* es 0, el resultado será una cadena vacía. Si *n* es mayor que la longitud de *cadena*, @DERECHA dará como resultado la cadena completa.

Notas

@DERECHA cuenta los signos de puntuación y los espacios como caracteres.

@DERECHA resultará útil cuando se desee copiar sólo una parte de un rótulo en otra celda (por ejemplo, para extraer los apellidos de un campo que contiene nombres y apellidos).

Utilizada en una macro, @DERECHA extraerá partes de rótulos insertados por el usuario para almacenarlos en una base de datos, realizar llamadas a subrutinas, o alterar la propia macro.

Utilice @DERECHA junto con @BUSCAR cuando no conozca el valor exacto de *n*, o cuando éste pueda variar.

Ejemplos

@DERECHA(B3;6) = ventas, si B3 contiene el rótulo Departamento de ventas.

Funciones @ similares

@IZDA da como resultado los *n* primeros caracteres de *cadena*. @MED da como resultado un grupo de caracteres contenidos en *cadena*.

@REDONDMULTP

@REDONDMULTP(*x*; *múltiplo*; [*dirección*]) redondea el valor *x* al *múltiplo* más próximo.

Argumentos

x y *múltiplo* pueden ser valores cualesquiera con el mismo signo.

dirección es un argumento opcional que permite especificar si se desea redondear por exceso o por defecto.

<i>dirección</i>	<u>1-2-3 redondea x</u>
<i>n</i>	
1	Por exceso
0	Al <i>múltiplo</i> más cercano; es el valor predeterminado si se omite este argumento
-1	Por defecto

Ejemplos

@REDONDMULTP(25,37;0,05;1) = 25,40

@REDONDMULTP(25,37;0,05;-1) = 25,35

Funciones @ similares

@ENT trunca un valor y elimina su parte decimal. @REDOND, @REDONDABAJO y @REDONDARRIBA redondean un valor a un número determinado de decimales. @PAR redondea un valor al primer número entero par que encuentre al alejarse de 0. @IMPAR redondea un valor al primer número entero impar que encuentre al alejarse de 0. @TRUNC trunca un valor a partir de una posición decimal determinada.

@REDOND, @REDONDABAJO, @REDONDARRIBA

@REDOND($x;n$) redondea el valor x al múltiplo más próximo de 10 elevado a n .

@REDONDABAJO($x;n;dirección$) redondea el valor x al múltiplo anterior más próximo de 10 elevado a n .

@REDONDARRIBA($x;n;dirección$) redondea el valor x al múltiplo siguiente más próximo de 10 elevado a n .

Argumentos

x puede ser cualquier valor.

n debe ser un valor entre -100 y 100. Si se omite n para @REDONDABAJO y @REDONDARRIBA, 1-2-3 utilizará el valor 0.

Si n es	@REDOND
Positivo	Afecta a la parte decimal del número (a la derecha de la coma decimal). Por ejemplo, si n es 2, 1-2-3 redondeará x a la centésima más cercana.
Negativo	Afecta a la parte entera del número (a la izquierda de la coma decimal). Por ejemplo, si n es -2, 1-2-3 redondea x a la centena más cercana.
0	Redondea x al entero más cercano.

dirección es un argumento opcional que indica cómo se desea que redondee 1-2-3 los valores negativos. Puede ser 0 o 1.

- Para @REDONDARRIBA: Si *dirección* es 0, 1-2-3 redondeará los valores negativos hacia valores más próximos a cero; si *dirección* es 1, 1-2-3 redondeará los valores negativos hacia valores más alejados de cero.
- Para @REDONDABAJO: Si *dirección* es 0, 1-2-3 redondeará los valores negativos hacia valores más alejados de cero; si *dirección* es 1, 1-2-3 redondeará los valores negativos hacia valores más próximos a cero.

Si se omite el argumento *dirección*, 1-2-3 utilizará el valor 0. Si x es positivo, el argumento *dirección* no tendrá efecto alguno.

Notas

Utilice el Formato fijo si desea que los valores se muestren con un número de decimales determinado, pero que 1-2-3 siga operando con ellos con toda la precisión disponible, en lugar de utilizar @REDOND.

Ejemplos

@REDOND(134,578;2) = 134,58

@ROUND(134,578;0) = 135

@REDOND(134,578;-2) = 100

@REDONDABAJO(134,578;2) = 134,57

@REDONDABAJO(134,578;0) = 134

@REDONDABAJO(134,578;-2) = 100

@REDONDARRIBA(134,578;2) = 134,58

@REDONDARRIBA(134,578;0) = 135

@REDONDARRIBA(134,578;-2) = 200

Funciones @ similares

@REDONDMULTP redondea un valor a un múltiplo determinado. @PAR redondea un valor al primer número entero par que encuentre al alejarse de 0. @IMPAR redondea un valor al primer número entero impar que encuentre al alejarse de 0. @ENT trunca un valor y elimina su parte decimal. @TRUNC trunca un valor a partir de una posición decimal determinada.

@FILAS

@FILAS(*rango*) cuenta el número de filas contenidas en *rango*.

Argumentos

rango puede ser la dirección o el nombre de un rango.

Notas

Utilice @FILAS junto con {DESDE} en macros que repitan la misma operación en una serie de filas para determinar cuándo debe detenerse la ejecución de la macro.

Ejemplos

@FILAS(A3..B7) = 5 (filas de la 3 a la 7).

@FILAS(CALIFICACIONES) = 43, si CALIFICACIONES es el rango B3..B45.

Funciones @ similares

@COLS cuenta el número de columnas y @HOJAS cuenta el número de hojas comprendidas en un rango.

Ejemplo de @STD y de @STDM

La tabla siguiente contiene datos sobre la altura y el peso de diez individuos seleccionados al azar. Se desea hallar la desviación típica de sus alturas.

$$@STD(A2..A11) = 5,9$$

Suponga que los sujetos representan una muestra seleccionada al azar de un grupo más amplio de individuos sometidos a la prueba.

$$@STDM(A2..A11) = 6,22$$

A	-----	A	-----	B	--
1	ALTURA	(cm)		PESO	(kg)
2		191		73	
3		188		86	
4		175		68	
5		175		76	
6		180		77	
7		180		73	
8		188		75	
9		173		68	
10		178		71	
11		179		87	

@C

@C(*rango*) arroja como resultado el contenido de la primera celda de *rango* en forma de rótulo.

Argumentos

rango puede ser una dirección de celda o un nombre de rango.

Notas

@C puede utilizarse junto con cualquier función @ de cadena o fórmula de cadena para comprobar si una celda contiene un valor cuando debe contener un rótulo (por ejemplo, una celda que contenga un código postal). Utilice @C para evitar que las fórmulas de cadena den como resultado ERR (por ejemplo, +A1&A2 dará como resultado ERR si cualquiera de las celdas contiene un valor).

Ejemplos

En las instrucciones de macro

```
{SI @C(B6)=""}{BIP}{INDICAR "DEBE ESCRIBIR UN ROTULO"}
```

@C dará como resultado una cadena vacía si la celda B6 contiene un valor o está vacía; 1-2-3 emitirá una señal sonora y cambiará el indicador de modo a DEBE ESCRIBIR UN ROTULO.

Funciones @ similares

@N da como resultado el contenido de la primera celda de *rango* en forma de valor. @ESCADENA comprueba si una celda contiene un rótulo.

@SEC

@SEC(z) calcula la secante del ángulo z. La secante de uno de los ángulos agudos de un triángulo rectángulo es la relación existente entre la hipotenusa y el cateto adyacente a dicho ángulo. La secante es la función recíproca del coseno.

Argumentos

z es un ángulo expresado en radianes y puede tener cualquier valor comprendido entre -2^63 y 2^63 .

Ejemplos

@SEC(@GRADARAD(30)) = 1,154701 (secante de un ángulo de 30 grados).

Funciones @ similares

@ASEC calcula el arco secante de un valor. @ASECH calcula el arco secante hiperbólica de un valor. @COS calcula el coseno de un ángulo. @SECH calcula la secante hiperbólica de un valor.

@SECH

@SECH(z) calcula la secante hiperbólica del ángulo z. La secante hiperbólica es la función recíproca del coseno hiperbólico. El resultado de @SECH es un valor mayor que 0 y menor o igual que 1.

Argumentos

z puede ser cualquier valor comprendido entre -709,7827 y 709,7827, ambos valores aproximados.

Ejemplos

@SECH(@GRADARAD(30)) = 0,87701

Funciones @ similares

@ASECH calcula el arco secante hiperbólica de un valor. @SEC calcula la secante de un valor.

@SEGUNDO

@SEGUNDO(*número_hora*) extrae la parte correspondiente a los segundos, es decir, un entero entre 0 y 59, a partir de *número_hora*.

Argumentos

número_hora debe ser un valor entre 0,000000 (medianoche) y 0,999988 (11:59:59 PM).

Ejemplos

@SEGUNDO(0,333) = 31

@SEGUNDO(**@VHORA**(11;15;45)) = 45, porque 45 es el argumento *segundos* de **@VHORA**(11;15;45).

Funciones @ similares

@HORA extrae la hora y **@MINUTO** los minutos correspondientes a un número de hora.

@SUMASERIE

@SUMASERIE($x;n;m;coeficientes$) calcula la suma de una serie de potencias.

Argumentos

x es el valor cuya serie de potencias se desea calcular.

n es la potencia inicial a la que se va a elevar x .

m es el incremento que se aplicará a n para cada término de la serie.

x , n y m deben ser valores.

coeficientes es un rango que contiene los coeficientes por los que 1-2-3 multiplicará cada una de las potencias de x . El número de celdas del rango *coeficientes* determina el número de términos de la serie. Por ejemplo, si *coeficientes* contiene diez celdas, la serie de potencias contendrá diez términos.

Ejemplos

El rango DATOS contiene los coeficientes siguientes: 0,2, 0,7 y 1,3.

@SUMASERIE(3,5;2;1;DATOS) = 227,5438

@HOJAS

@HOJAS(*rango*) cuenta el número de hojas contenidas en *rango*.

Argumentos

rango puede ser la dirección o el nombre de un rango.

Notas

Utilice **@HOJAS** junto con {DESDE} en macros que repitan una misma operación en una serie de hojas, para determinar cuándo debe detenerse la macro.

Ejemplos

@HOJAS(TRIM_2) = 4 si TRIM_2 es el rango B:B3..E:C45 (hojas B, C, D y E).

Funciones @ similares

@COLS cuenta las columnas y **@FILAS** las filas contenidas en un rango. **@CONVERTREF** convierte las letras de la A a la IV correspondientes a las columnas o a las hojas de 1-2-3 en números del 1 al 256, y viceversa.

@SIGNO

@SIGNO(x) da como resultado 1 si x es un valor positivo, 0 si x es igual a 0 y -1 si x es un valor negativo.

Argumentos

x puede ser cualquier valor.

Ejemplos

$$\text{@SIGNO}(15) = 1$$

$$\text{@SIGNO}(15*0) = 0$$

$$\text{@SIGNO}(-15) = -1$$

@SEN

@SEN(z) calcula el seno del ángulo z. El seno de un ángulo de un triángulo rectángulo es la relación existente entre el cateto opuesto a dicho ángulo y la hipotenusa.

Argumentos

z es un ángulo expresado en radianes y puede ser un valor comprendido entre -2^63 y 2^63 .

Ejemplos

@SEN(@GRADARAD(30)) = 0,5 (seno de un ángulo de 30 grados).

Funciones @ similares

@ASEN calcula el arco seno de un valor. @SENH calcula el seno hiperbólico de un ángulo.

@SENH

@SENH(z) calcula el seno hiperbólico del ángulo z.

Argumentos

z puede ser cualquier valor comprendido entre -709,7827 y 709,7827, ambas cifras aproximadas.

Ejemplos

@SENH(@GRADARAD(30)) = 0,547853

Funciones @ similares

@ASENH calcula el arco seno hiperbólico de un valor. @SEN calcula el seno de un ángulo.

@ASIMETRIA

@ASIMETRIA(*rango*;*[tipo]*) calcula el coeficiente de asimetría para los valores de *rango*.

Argumentos

rango es el nombre o la dirección de un rango que contiene una serie de valores. Si *rango* contiene menos de tres valores, @ASIMETRIA dará como resultado ERR.

tipo es un argumento opcional que permite especificar si 1-2-3 debe calcular el coeficiente de asimetría para una población o para una muestra.

<u><i>tipo</i></u>	<u>1-2-3 calcula</u>
0	El coeficiente de asimetría para una población; es el valor predeterminado si se omite este argumento
1	El coeficiente de asimetría para una muestra

Notas

El coeficiente de asimetría mide el grado de simetría de una distribución alrededor de su media aritmética. Si es positivo, significa que la curva de la distribución tiene una cola alargada hacia la izquierda; si es negativo, significa que la curva de la distribución tiene una cola alargada hacia la derecha.

Ejemplos

El rango DATOS contiene los valores siguientes: 2, 5, 6, 9, 1, 2, 4.

@ASIMETRIA(DATOS) = 0,584816

Funciones @ similares

@CURTOSIS calcula el coeficiente de curtosis para los valores de una lista. @STD y @STDPUA calculan la desviación típica de una población. @VAR y @VARPUA calculan la varianza de una población.

@SLN

@SLN(*coste*; *valor_residual*; *vida*) calcula la cuota de amortización lineal de un bien de inversión con un *coste* inicial, una *vida* útil estimada y un *valor_residual* dados para cada uno de los períodos en que se mide su vida útil.

Argumentos

coste representa el importe pagado por el bien y puede ser cualquier valor.

valor_residual representa el valor estimado del bien al final de su vida útil y puede ser cualquier valor.

vida representa el número de períodos que han de transcurrir para que el bien se deprecie hasta su valor residual; puede ser cualquier valor excepto 0.

Notas

El método de amortización lineal reparte el coste amortizable (el coste de adquisición menos el valor residual) uniformemente entre los períodos (normalmente años) de la vida útil del bien, es decir, el tiempo que transcurre hasta que se amortiza.

Ejemplos

El precio de compra de una máquina de oficina es de 10.000\$, su vida útil de cuatro años y su valor residual después de esos cuatro años se estima en 1.200\$. Se desea calcular la cuota de amortización anual por el método de amortización lineal.

@SLN(10000;1200;4) = 2.200\$

Funciones @ similares

@DB calcula la cuota de amortización de un bien por el método de los saldos decrecientes (Declining Balance);

@DDB calcula la cuota de amortización por el método de los saldos decrecientes al 200% (Double Declining Balance), **@VDB** la calcula por el método de los saldos decrecientes al 200% y permite especificar otras tasas de amortización diferentes del 200% y **@SYD** la calcula por el método de los números dígitos.

@MENOR

@MENOR(*rango*; *n*) halla el *enésimo* valor más pequeño de los comprendidos en *rango*.

Argumentos

rango es el nombre o la dirección de un rango que contiene valores.

n puede ser cualquier entero positivo. Si *n* es mayor que el número de valores existentes en *rango*, **@MENOR** dará como resultado ND.

Ejemplos

Un rango denominado CALIFICACIONES contiene los valores siguientes: 87, 85, 90, 80, 82, 92, 79, 85, 95, 86.

@MENOR(CALIFICACIONES;3) dará como resultado 82, la tercera calificación más baja.

Funciones @ similares

@MAYOR halla el *enésimo* valor más grande de los comprendidos en un rango. **@MAX** y **@MAXPURA** hallan el valor máximo entre los comprendidos en un rango. **@MIN** y **@MINPURA** hallan el valor mínimo entre los comprendidos en un rango.

@RCUAD

@RCUAD(x) da como resultado la raíz cuadrada positiva de x.

Argumentos

x debe ser un valor positivo.

Ejemplos

@RCUAD(B3) = 10, si B3 contiene el valor 100.

@RCUAD(-2) = ERR, porque x es negativo.

@RCUADPORPI

@RCUADPORPI(x) calcula la raíz cuadrada de $x^*\pi$.

Argumentos

x puede ser cualquier valor positivo o 0.

Ejemplos

@RCUADPORPI(0,5) = 1,253314

@RCUADPORPI(2) = 2,506628

Funciones @ similares

@RCUAD calcula la raíz cuadrada positiva de un valor. @PI da como resultado el valor π .

@STD, @STDM, @STDPURA, @STDMPURA

@STD(*lista*) calcula la desviación típica de una *lista* de valores que constituyen una población.

@STDM(*lista*) calcula la desviación típica de una *lista* de valores que constituyen una muestra.

@STDPURA(*lista*) calcula la desviación típica de una *lista* de valores que constituyen una población, pero ignora las celdas que contienen rótulos.

@STDMPURA(*lista*) calcula la desviación típica de una *lista* de valores que constituyen una muestra, pero ignora las celdas que contienen rótulos.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rango que contengan números o fórmulas. Utilice separadores de argumentos para separar los elementos de *lista*.

Consulte también el tema Argumentos de las funciones @ estadísticas.

Notas

@STD y **@STDPURA** utilizan el método *n* o de la población para calcular la desviación típica. Dicho método interpreta que los valores seleccionados constituyen el total de la población. Si se trata tan sólo de una muestra, la desviación típica resultará poco fiable debido a los errores cometidos al obtener la muestra.

@STDM y **@STDMPURA** utilizan el método *n-1* o de la muestra para calcular la desviación típica de una muestra de una población de datos. El resultado obtenido con el método *n-1* es ligeramente superior al obtenido con el método *n*, con lo que quedan compensados los errores cometidos al obtener la muestra y la precisión es mayor.

La desviación típica es la raíz cuadrada de la varianza de todos los valores con relación a la media.

Ejemplos

@STD y @STDM

Funciones @ similares

@BSTD y @BSTDM calculan la desviación típica de los valores contenidos en un campo de una tabla de base de datos. @VAR y @VARPURA calculan la varianza de una lista de valores que constituyen una población. @VARM y @VARMPURA calculan la varianza de una lista de valores que constituyen una muestra.

@CADENA

@CADENA(x;n) convierte el valor x en un rótulo utilizando el formato especificado por n.

Argumentos

x es un valor.

n es un entero extraído de la tabla siguiente:

<u>Valor de n</u>	<u>Formato del resultado</u>
de 0 a 116	<u>Fijo</u> , con n decimales
de 1000 a 1116	<u>Punto</u> , con n-1000 decimales
de -18 a -1	<u>Científico</u> , con @ABS(n) dígitos
de 10001 a 10512	<u>General</u> , hasta un máximo de n-10000 caracteres

Notas

@CADENA no tiene en cuenta los caracteres de formato que 1-2-3 utiliza para mostrar el valor x. Esto incluye todos los símbolos monetarios y demás símbolos numéricos, tanto si los inserta el usuario como si los crea 1-2-3 después de haber seleccionado un formato numérico. Por ejemplo, si la celda A7 contiene el valor 45,23 (especificado en dólares estadounidenses), @CADENA(A7;2) dará como resultado la cadena 45,23.

Ejemplos

@CADENA(203;3) = al rótulo 203,000

@CADENA(1,23587;0) = al rótulo 1

@CADENA(20500;1002) = al rótulo 20.500,00

@CADENA(@PI;-5) = al rótulo 3,1416E+000

@CADENA(123456,789;10008) = 123456,8

Funciones @ similares

@VALOR convierte un número especificado como texto en su valor correspondiente.

@SUBTOTAL

@SUBTOTAL(*lista*) suma los valores contenidos en *lista*. **@SUBTOTAL** se utiliza para indicar a **@TOTAL** las celdas que debe sumar.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas numéricas. Utilice separadores de argumentos para separar los elementos de *lista*.

Consulte también el tema Argumentos de las funciones @ estadísticas.

Ejemplos

@SUBTOTAL(B5..B9) dará como resultado la suma de los valores contenidos en B5..B9.

@SUBTOTAL(VENTAS;M25..R25) dará como resultado la suma de los valores contenidos en el rango VENTAS más los del rango M25..R25.

Funciones @ similares

@SUMA suma los valores contenidos en una lista. **@SUMANEGATIVOS** suma sólo los valores negativos de una lista. **@SUMAPOSITIVOS** suma sólo los valores positivos de una lista.

@SUMA

@SUMA(*lista*) suma los valores contenidos en *lista*.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas. Utilice separadores de argumentos para separar los elementos de *lista*.

Consulte también el tema [Argumentos de las funciones @ estadísticas](#).

Ejemplos

@SUMA(B5..B9) dará como resultado la suma de los valores contenidos en el rango B5..B9.

@SUMA(VENTAS;M25..R25) da como resultado la suma de los valores del rango VENTAS más los del rango M25..R25.

SmartIcons relacionados



Suma los valores del rango adyacente o que aparece resaltado, en el caso de que se incluyan celdas vacías debajo o a la derecha del rango.

Funciones @ similares

[@BSUMA](#) calcula la suma de los valores que cumplen los criterios especificados. [@SUMAN](#) suma cada valor *enésimo* de una lista, a partir de un *número offset*. [@SUBTOTAL](#) calcula la suma de los valores de una lista e indica a [@TOTAL](#) los valores que debe sumar. [@SUMANEGATIVOS](#) suma sólo los valores negativos de una lista. [@SUMAPOSITIVOS](#) suma sólo los valores positivos de una lista.

@SUMAPRODUCTOS

@SUMAPRODUCTOS(*lista*) multiplica los valores contenidos en las celdas homólogas de varios rangos y, a continuación, suma los productos.

Argumentos

lista puede ser cualquier combinación de rangos de igual tamaño y forma que contengan valores. Si el tamaño y la forma de los rangos de *lista* no son iguales, **@SUMAPRODUCTOS** dará como resultado **ERR**.

Consulte también el tema [Argumentos de las funciones @ estadísticas](#).

Notas

Si los rangos de *lista* son columnas, **@SUMAPRODUCTOS** realizará las multiplicaciones por filas. Si los rangos de *lista* son filas, **@SUMAPRODUCTOS** realizará las multiplicaciones por columnas. Si cada rango de *lista* ocupa más de una columna, **@SUMAPRODUCTOS** realizará las multiplicaciones por filas.

Ejemplos

En el ejemplo siguiente se utiliza **@SUMAPRODUCTOS** para calcular el total de las comisiones pagadas a un agente de una inmobiliaria por las viviendas vendidas en el mes de febrero.

lista contiene dos rangos: VENDIDO (D4..D6), con los precios de venta de tres viviendas, y COM (E4..E6) con el porcentaje de comisión sobre el precio de venta recibido por el agente inmobiliario en cada caso:

VENDIDO	COM
25.000\$	0,04
34.580\$	0,05
77.325\$	0,04

@SUMAPRODUCTOS(VENDIDO;COM) = 5.822\$, el total de las comisiones (1.000\$+ 1.729\$+ 3.093\$) recibidas por el agente en concepto de la venta de las viviendas.

Funciones @ similares

@SUMACUADRADOS calcula la suma de los cuadrados de los valores de una lista. **@SUMADIF2** resta los valores correspondientes de las celdas de dos rangos, calcula los cuadrados de las diferencias y, por último, suma los cuadrados.

@SUMACUADRADOS

@SUMACUADRADOS(*lista*) calcula la suma de los cuadrados de los valores de *lista*.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas numéricas. Utilice separadores de argumentos para separar los elementos de *lista*.

Consulte también el tema Argumentos de las funciones @ estadísticas.

Ejemplos

@SUMACUADRADOS(2;4;6) = 56

Funciones @ similares

@SUMA suma los valores contenidos en una lista. **@SUMAPRODUCTOS** multiplica los valores contenidos en las celdas homólogas de varios rangos y, a continuación, suma los productos.

@SUMADIF2

@SUMADIF2(*rango1*;*rango2*) resta de las celdas correspondientes del *rango1* los valores de las celdas del *rango2*, calcula los cuadrados de las diferencias y, por último, suma los cuadrados.

Argumentos

rango1 y *rango2* son rangos de igual tamaño y forma que contienen valores. Si *rango1* y *rango2* no tienen el mismo tamaño y la misma forma, **@SUMADIF2** dará como resultado ERR.

Notas

Si *rango1* y *rango2* son rangos de una sola columna, 1-2-3 calculará las diferencias por filas. Si *rango1* y *rango2* son rangos de varias columnas, 1-2-3 calculará las diferencias por columnas.

Ejemplos

En el ejemplo siguiente, el nombre de *rango1* es MARTES y el de *rango2* MIERCOLES:

MARTES	MIERCOLES
5	3
4	4
7	8

@SUMADIF2(MARTES;MIERCOLES) = 5

Funciones @ similares

@SUMAPRODUCTOS multiplica los valores contenidos en las celdas homólogas de varios rangos y, a continuación, suma los productos. **@SUMACUADRADOS** calcula la suma de los cuadrados de los valores de una lista.

@SYD

@SYD(*coste;valor_residual;vida;período*) calcula, por el método de los números dígitos, la cuota de amortización de un bien de inversión con un *coste* inicial y un *valor_residual* dados para un *período* determinado de su *vida* útil estimada.

Argumentos

coste representa el importe pagado por el bien y puede ser cualquier valor.

valor_residual representa el valor estimado para el bien al final de su vida útil y puede ser cualquier valor.

vida representa el número de períodos que deberán transcurrir para que el bien se deprecie hasta llegar a su valor residual. Puede ser cualquier valor mayor o igual que 1.

período representa el período concreto de tiempo para el que se desea calcular la cuota de amortización y puede ser cualquier valor mayor o igual que 1.

Notas

El método de los números dígitos calcula cuotas de amortización decrecientes, de forma que en cada período se produzca (y se contabilice) un gasto de amortización mayor que en el siguiente (aunque no tanto como con el método de los saldos decrecientes al 200%). El importe amortizable equivale al coste de adquisición menos el valor residual.

Utilice **@SYD** cuando necesite que la cuota de amortización de un bien de inversión sea mayor al principio de su vida útil que al final (por ejemplo, a la hora de preparar un plan de amortización para su aprobación por las autoridades fiscales).

Ejemplos

El precio de compra de una máquina de oficina es de 10.000\$, su vida útil de cuatro años y su valor residual después de esos cuatro años se estima en 1.200\$. Para calcular la cuota de amortización correspondiente al primer año por el método de los números dígitos se utiliza la fórmula **@SYD**(10000;1200;4;1), que da como resultado 3.520\$.

Funciones @ similares

@DDB calcula la cuota de amortización por el método de los saldos decrecientes al 200% (Double Declining Balance). **@VDB** la calcula por el método de los saldos decrecientes al 200% y permite especificar otras tasas de amortización diferentes del 200%, **@DB** utiliza el método de los saldos decrecientes y **@SLN** utiliza el método de la amortización lineal para calcularla.

Ejemplo de @VAR y @VARM

La tabla siguiente contiene datos sobre la altura y el peso de diez individuos seleccionados al azar. Se desea hallar la varianza de sus pesos.

$$@VAR(B2..B11) = 39,04$$

Suponga que los sujetos representan una muestra seleccionada al azar de un grupo más amplio de individuos sometidos a la prueba.

$$@VARM(B2..B11) = 43,38$$

A ----- A ----- B --

1	ALTURA (cm)	PESO (kg)
2	191	73
3	188	86
4	175	68
5	175	76
6	180	77
7	180	73
8	188	75
9	173	68
10	178	71
11	179	87

@TAN

@TAN(z) calcula la tangente del ángulo z. La tangente de un ángulo de un triángulo rectángulo es la relación existente entre el cateto opuesto a dicho ángulo y el cateto adyacente.

Argumentos

z es un ángulo expresado en radianes y puede tener cualquier valor comprendido entre -2^{63} y 2^{63} .

Ejemplos

@TAN(@GRADARAD(35)) = 0,700208 (tangente de un ángulo de 35 grados).

Funciones @ similares

@ATAN calcula el arco tangente a partir de la tangente de un ángulo. @TANH calcula la tangente hiperbólica de un ángulo.

@TANH

@TANH(z) calcula la tangente hiperbólica del ángulo z. La tangente hiperbólica es la relación existente entre el seno hiperbólico y el coseno hiperbólico. El resultado de @TANH es un valor entre -1 y 1.

Argumentos

z puede ser cualquier valor comprendido entre -709,7827 y 709,7827, ambas cifras aproximadas.

Ejemplos

@TANH(@GRADARAD(30)) = 0,480473

Funciones @ similares

@ATANH calcula el arco tangente hiperbólica de un valor. @TAN calcula la tangente de un ángulo.

@DISTT

@DISTT(x ; $grados_libertad$; [$tipo$; $colas$]) calcula la distribución t de Student.

Argumentos

El valor que puede especificarse para x depende del valor especificado para $tipo$.

<u>Si $tipo$ es</u>	<u>x es</u>
0	El valor crítico o el límite superior para el valor de la variable aleatoria de la distribución t acumulada (puede ser cualquier valor); es el valor predeterminado si se omite este argumento
1	Una probabilidad; puede ser un valor entre 0 y 1

$grados_libertad$ es el número de grados de libertad para la muestra. Debe ser un entero positivo.

$tipo$ es un argumento opcional que permite indicar a 1-2-3 cómo debe calcular la función @DISTT.

<u>$tipo$</u>	<u>1-2-3 calcula</u>
0	El nivel de significación correspondiente al valor crítico x ; es el valor predeterminado si se omite este argumento
1	El valor crítico correspondiente al nivel de significación x

$colas$ es un argumento opcional que permite especificar la dirección en que se desea evaluar la distribución t .

<u>$colas$</u>	<u>1-2-3 evalúa</u>
1	Una distribución t unilateral o de una cola
2	Una distribución t bilateral o de dos colas; es el valor predeterminado si se omite este argumento

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Notas

@DISTT calcula la distribución t acumulada con un margen de error de $\pm 3 \cdot 10^{-7}$. Si, tras 100 iteraciones de cálculo, @DISTT no puede determinar el resultado con una precisión de 0,0000001, dará como resultado ERR.

La distribución t de Student es la distribución continua obtenida de la relación existente entre una distribución normal tipificada y la raíz cuadrada del cociente de una distribución ji cuadrado dividida entre el número de grados de libertad.

Ejemplos

@DISTT(2,228;10) = 0,05

@DISTT(0,05;10;1) = 2,228

Funciones @ similares

@DISTJI calcula la distribución ji cuadrado. @DISTF calcula la distribución F . @TESTI calcula la probabilidad asociada a una prueba t de Student.

@PERIODO, @NPERIODO

@PERIODO(*pago;interés;valor_futuro*) calcula el número de *pagos* periódicos necesarios para acumular un determinado *valor_futuro*. Se supone que los pagos son de igual cuantía y que la inversión rinde cierto *interés* periódico.

@NPERIODO(*pago;interés;valor_futuro;[tipo;valor_actual]*) calcula el número de *pagos* periódicos necesarios para acumular un determinado *valor_futuro* partiendo de cierto *valor_actual*. Se supone que los pagos son de igual cuantía y que la inversión rinde cierto *interés* periódico. **@NPERIODO** realiza el cálculo para anualidades ordinarias o anticipadas, dependiendo del valor especificado para el argumento *tipo*.

Argumentos

pago representa el valor de los pagos periódicos (cantidades que se invierten periódicamente). Puede ser cualquier valor excepto el valor 0.

interés representa el tipo de interés periódico. Puede ser cualquier valor mayor que -1.

valor_futuro representa la cantidad que se desea acumular. Puede ser cualquier valor.

tipo es un argumento opcional que permite especificar si se desea realizar el cálculo para una anualidad ordinaria o para una anualidad anticipada.

<u><i>tipo</i></u>	<u>1-2-3 realiza el cálculo para</u>
0	Anualidad ordinaria (los pagos se efectúan al final de cada período); es el valor predeterminado si se omite este argumento
1	Anualidad anticipada (los pagos se efectúan al principio de cada período)

valor_actual es un argumento opcional que permite especificar el valor actual de la serie de pagos futuros. *valor_actual* puede ser cualquier valor. Si se omite el argumento *valor_actual*, 1-2-3 utiliza el 0.

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Notas

Si se utiliza un *valor_futuro* negativo, **@PERIODO** puede emplearse para calcular el tiempo necesario para amortizar un crédito. Por ejemplo, si desea conocer el período de amortización de un crédito de 10.000\$ a un tipo de interés anual del 10% con pagos de 1.174\$ al año, deberá escribir la fórmula

@ABS(@PERIODO(1174,6;0,1;-10000)) cuyo resultado es 20 años.

Ejemplos

Al final de cada año se depositan 2.000\$ en una cuenta de ahorro que devenga un interés anual del 7,5% capitalizado anualmente; para calcular el tiempo necesario para acumular 100.000\$ se utilizará la siguiente fórmula:

@PERIODO(2000;0,075;100000) = 21,5 años

Si los pagos se realizan al principio de cada año:

@NPERIODO(2000;0,075;100000;1) = 20,76 años

Funciones @ similares

@CPERIODO calcula el número de períodos de capitalización necesarios para que una inversión de un solo depósito alcance un valor futuro determinado.

@PERIODO2 calcula el número de períodos necesarios para que una serie de pagos iguales alcancen en el futuro un valor determinado a un tipo de interés periódico, suponiendo una anualidad anticipada.

@VHORA

@VHORA(hora;minutos;segundos) calcula el número de hora correspondiente a unos valores de *hora*, *minutos* y *segundos* específicos.

Argumentos

hora es un entero comprendido entre 0 (medianoche) y 23 (11:00 PM), ambos inclusive.

minutos es un entero comprendido entre 0 y 59, ambos inclusive.

segundos es un entero comprendido entre 0 y 59, ambos inclusive.

Notas

Utilice un formato de hora para que el número de hora se pueda mostrar con la forma de la hora que representa.

Ejemplos

La fórmula $(@VHORA(13;0;0)-@VHORA(9;15;0))*75*24$ calcula lo que debe cobrar un asesor un día determinado restando la hora de inicio (9:15 AM) a la hora de finalización (1:00 PM) y multiplicando el resultado por una tarifa de 75\$ la hora.

Funciones @ similares

@VALHORA convierte rótulos en números de hora.

@VALHORA

@VALHORA(*cadena*) calcula el número de hora correspondiente a la hora especificada en *cadena*.

Argumentos

cadena puede ser texto encerrado entre comillas (" "), una fórmula cuyo resultado sea texto, o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo. *cadena* debe estar en uno de los cuatro formatos de hora de 1-2-3.

Notas

Utilice @VALHORA cuando necesite convertir horas especificadas como rótulos en su número de hora correspondiente con el fin de utilizarlas en cálculos. @VALHORA resulta especialmente útil cuando se trabaja con datos importados de otros programas, como los procesadores de texto.

Utilice un formato de hora para que el número de hora se pueda mostrar con la forma de la hora que representa.

Ejemplos

@VALHORA("08:19:27 AM") = 0,34684

@VALHORA("08:19:27 AM") = 0,34684 con el formato 08:19:27 AM, si la celda tiene asignado el formato 11:59:59 AM/PM.

Funciones @ similares

@VHORA calcula el número de hora de un grupo de valores que representan la hora, los minutos y los segundos.

@HOY

@HOY calcula el número de fecha correspondiente a la fecha actual indicada por el PC.

Notas

1-2-3 recalculará @HOY cada vez que se recalcule la hoja.

Utilice un formato de fecha para que el número de fecha se pueda mostrar con la forma de la fecha que representa.

Ejemplos

@HOY = 31048, el 1 de enero de 1985.

@HOY = 33418, el 29 de junio de 1991.

@SUPRESP

@SUPRESP(*cadena*) elimina los espacios iniciales y finales y reduce a uno los espacios múltiples situados entre dos palabras de *cadena*.

Argumentos

cadena puede ser texto encerrado entre comillas (" "), una fórmula cuyo resultado sea texto, o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

Notas

Utilice @SUPRESP para evitar que las entradas de una base de datos contengan espacios innecesarios que puedan afectar al orden de clasificación cuando ordene un rango.

Ejemplos

@SUPRESP(" 45 3/8") = 45 3/8, tras eliminar el espacio anterior a 45 y los dos espacios existentes entre 45 y 3/8.

@SUPRESP(" San Marcos, 20") = San Marcos, 20, tras eliminar el espacio anterior a San; dos de los tres espacios existentes entre San y Marcos; y uno de los dos espacios existentes entre Marcos, y 20.

Funciones @ similares

@DEFINIRCADENA da como resultado un texto alineado con un número específico de espacios.

@VERDAD

@VERDAD arroja como resultado el valor lógico 1 (verdad).

Notas

Si una expresión lógica como $A1=B1$ es verdadera, su valor lógico es 1. Si es falsa, su valor lógico es 0.

Utilizar @VERDAD en fórmulas que evalúan condiciones lógicas equivale a utilizar el valor 1, pero @VERDAD facilitará la lectura de dichas fórmulas.

Ejemplos

@SI(A6>500;@VERDAD;@FALSO) = 1 cuando la celda A6 contiene un valor mayor que 500.

Funciones @ similares

@FALSO arroja como resultado el valor lógico 0.

@TRUNC

@TRUNC(x;[n]) trunca el valor x a partir de la posición especificada mediante n.

Argumentos

x es un valor.

n es un argumento opcional y puede ser cualquier valor entre -100 y 100.

Si n es	@TRUNC
Positivo	Afecta a la parte decimal del número (a la derecha de la coma decimal). Por ejemplo, si n es 2, 1-2-3 truncará x a la centésima más cercana.
Negativo	Afecta a la parte entera del número (a la izquierda de la coma decimal). Por ejemplo, si n es -2, 1-2-3 truncará x a la centena más cercana.
0	Trunca x al entero más cercano; es el valor predeterminado si se omite este argumento

Notas

Si desea que los valores se muestren con un número de decimales determinado, pero que 1-2-3 siga realizando los cálculos con ellos con toda la precisión disponible, seleccione un formato numérico fijo en lugar de utilizar @TRUNC.

Ejemplos

@TRUNC(123,45) = 123

@TRUNC(-123,45) = -123

@TRUNC(123,45;-2) = 100

@TRUNC(123,45;1) = 123,4

@TRUNC(-123,45;-2) = -100

@TRUNC(-123,45;1) = -123,4

Funciones @ similares

@REDOND, @REDONDABAJO y @REDONDARRIBA redondean un valor a un número determinado de decimales. @REDONDMULTP redondea un valor a un múltiplo determinado. @PAR redondea un valor al primer número entero par que encuentre al alejarse de 0. @IMPAR redondea un valor al primer número entero impar que encuentre al alejarse de 0. @ENT trunca un valor y elimina su parte decimal.

@MAYUSC

@MAYUSC(*cadena*) convierte en mayúsculas todas las letras contenidas en *cadena*.

Argumentos

cadena puede ser texto encerrado entre comillas (" "), una fórmula cuyo resultado sea texto, o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

Notas

La utilización de mayúsculas y minúsculas afecta al orden de clasificación de los rótulos cuando se procede a la ordenación de un rango. Dos rótulos aparentemente idénticos pueden no aparecer juntos tras ordenar un rango si uno de ellos contiene mayúsculas donde el otro contiene minúsculas.

Ejemplos

@MAYUSC("Número de cuenta") = NÚMERO DE CUENTA

@MAYUSC(B2) = AVISO, si B2 contiene el rótulo aviso.

Funciones @ similares

@MINUSC convierte en minúsculas todas las letras de *cadena*. @NOMPROPIO convierte en mayúscula sólo la primera letra de cada palabra de *cadena*.

@VALOR

@VALOR(*cadena*) convierte un número especificado como *cadena* en su valor correspondiente.

Argumentos

cadena puede ser texto encerrado entre comillas (" ") o un rótulo que contenga sólo números. Puede aparecer como número normal (456,7), con formato científico (4,567E2), como número mixto (45 7/8) o con cualquier otro formato (45,67\$).

Notas

@VALOR ignorará los espacios existentes delante y detrás de *cadena*; no obstante, si *cadena* contiene espacios que separen los símbolos de los números (como por ejemplo, \$ 32,85 o 12 %) @VALOR dará como resultado ERR. Si el espacio forma parte del símbolo monetario de forma predeterminada, como es el caso de 12 Pts, @VALOR no dará como resultado el valor ERR.

Si *cadena* es una celda vacía o una cadena vacía, @VALOR dará como resultado 0; si *cadena* contiene caracteres no numéricos, @VALOR dará como resultado el valor ERR.

Si desea sustituir la función @VALOR por el valor que da como resultado, presione F2 (EDICION) y después F9 (CALC).

Cuando se utilice @VALOR no podrán realizarse cálculos dentro del argumento *cadena*. Sin embargo, será posible crear una fórmula con varias funciones @VALOR. Por ejemplo, @VALOR("22"+"20") = 0, pero @VALOR("22")+@VALOR("20") = 42.

Ejemplos

@VALOR("543") = al valor 543.

@VALOR(B3) = al valor 49,75, si la celda B3 contiene el rótulo 49 3/4.

@VALOR("85%") = al valor 0,85.

Funciones @ similares

@CADENA convierte un valor en un rótulo.

@VAR, @VARM, @VARPURA, @VARMPURA

@VAR(*lista*) calcula la varianza de una *lista* de valores que constituyen una población.

@VARM(*lista*) calcula la varianza de una *lista* de valores que constituyen una muestra.

@VARPURA(*lista*) calcula la varianza de una *lista* de valores que constituyen una población, pero ignora las celdas que contienen rótulos.

@VARMPURA(*lista*) calcula la varianza de una *lista* de valores que constituyen una muestra, pero ignora las celdas que contienen rótulos.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas. Utilice separadores de argumentos para separar los elementos de *lista*.

Consulte también el tema [Argumentos de las funciones @ estadísticas](#).

Notas

Las funciones @ relativas a la varianza resultan de utilidad cuando se efectúan pruebas estadísticas para análisis de varianza.

@VAR y **@VARPURA** utilizan el método n o de la población para calcular la varianza. Dicho método interpreta que los valores seleccionados constituyen el total de la población. Si se trata tan sólo de una muestra, la varianza resultará poco fiable debido a los errores cometidos al obtener la muestra.

@VARM y **@VARMPURA** utilizan el método n-1 o de la muestra para calcular la varianza. El resultado obtenido con el método n-1 es ligeramente superior al obtenido con el método n, con lo que quedan compensados los errores cometidos al obtener la muestra y la precisión es mayor.

Ejemplos

[@VAR y @VARM](#)

Funciones @ similares

[@BVAR](#) y [@BVARM](#) calculan la varianza de una serie de valores que cumplen unos determinados criterios y constituyen una población.

@VDB

@VDB(*coste;valor_residual;vida;período_inicial;período_final;[factor_amort;cambio]*) calcula, por el método de los saldos decrecientes al 200% (con la posibilidad de especificar otras tasas de amortización distintas del 200% y si se produce o no el cambio al método de amortización lineal), la cuota de amortización correspondiente al intervalo de tiempo (definido por *período_inicial* y *período_final*) de la *vida* útil estimada de un bien de inversión que tiene un determinado *coste* inicial y un determinado *valor_residual* final.

Argumentos

coste representa el importe pagado por el bien y puede ser cualquier valor mayor que *valor_residual*.

valor_residual representa el valor estimado para el bien al final de su vida útil y puede ser cualquier valor.

vida representa el número de períodos que deberán transcurrir para que el bien se deprecie hasta llegar a su valor residual y puede ser cualquier valor mayor que 0.

período_inicial representa el punto en la vida de un bien en el que se desea iniciar el cálculo de la amortización. Puede ser cualquier valor mayor o igual que 0, siempre que no sea superior a *vida*.

período_final representa el punto en la vida del bien en el que se desea terminar el cálculo de la amortización. Puede ser cualquier valor mayor que *período_inicial*.

período_inicial y *período_final* se refieren a la vida del bien en relación al año fiscal. Por ejemplo, si se desea hallar la cuota de amortización correspondiente al primer año de un bien adquirido al principio del segundo trimestre del año fiscal, *período_inicial* sería 0 y *período_final* sería 0,75 (un año menos un trimestre). @VDB puede utilizarse para realizar cálculos de amortización para varios períodos.

@VDB permite utilizar una opción de período de puesta en funcionamiento para calcular la amortización durante dicho período. @VDB utilizará la parte fraccionaria de *período_inicial* y *período_final* para determinar la opción de período de puesta en funcionamiento. Si tanto *período_inicial* como *período_final* tienen parte fraccionaria, @VDB utilizará la de *período_inicial*.

factor_amort es un argumento opcional que especifica el porcentaje de amortización lineal que se desea utilizar como tasa de amortización. Si no se incluye este argumento, 1-2-3 utilizará el 200%, que es la correspondiente a la función @DDB. El argumento *factor_amort* puede ser cualquier valor mayor o igual que 0, aunque normalmente se utilizan los valores 1,25, 1,50, 1,75 y 2.

cambio es un argumento opcional que se incluye cuando no se desea que @VDB cambie al método de amortización lineal al final de la vida útil del bien. Normalmente se pasa del método de los saldos decrecientes al de amortización lineal cuando con este último se obtienen cuotas mayores que con el primero.

Si cambio es	@VDB
0	Cambia automáticamente a amortización lineal cuando con esta última se obtengan cuotas mayores que con el método de los saldos decrecientes; es el valor predeterminado si se omite este argumento
1	No cambia nunca a amortización lineal

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Notas

El método de los saldos decrecientes mantiene una tasa de amortización estable hasta que el valor residual de un bien es inferior al valor obtenido con la siguiente ecuación:

$$(\text{valor contable} * ((1 - (\text{tasa}/\text{vida}))^{\text{vida}}))$$

donde valor contable = coste - valor residual - total amortizado.

A partir de este momento, 1-2-3 cambiará al método de amortización lineal para el resto de la vida útil del bien con el fin de evitar un valor residual excesivo. Al cambiar al método de amortización lineal, 1-2-3 ajustará el resultado de @VDB siempre que sea necesario para garantizar que la amortización total a lo largo de la vida del bien sea igual a su coste menos su valor residual.

Ejemplos

En el siguiente ejemplo se calcula la cuota de amortización de una máquina de oficina adquirida en el primer

trimestre del año fiscal. La máquina está valorada en 10.000\$, tiene una vida útil de diez años, y un valor residual después de esos diez años estimado en 600\$. Las fórmulas siguientes calculan la cuota de amortización correspondiente a cada año de la vida útil del bien, suponiendo que la legislación fiscal le permite amortizar por el método de los saldos decrecientes con una tasa máxima del 150%. Observe que el cambio al método de amortización lineal se produce en el sexto año.

@VDB(10000;600;10;0,875;1,5)	=	1.312,50\$
@VDB(10000;600;10;0,875;1,875;1,5)	=	1.303,13\$
@VDB(10000;600;10;1,875;2,875;1,5)	=	1.107,66\$
@VDB(10000;600;10;2,875;3,875;1,5)	=	941,51\$
@VDB(10000;600;10;3,875;4,875;1,5)	=	800,28\$
@VDB(10000;600;10;4,875;5,875;1,5)	=	767,79\$
@VDB(10000;600;10;5,875;6,875;1,5)	=	767,79\$
@VDB(10000;600;10;6,875;7,875;1,5)	=	767,79\$
@VDB(10000;600;10;7,875;8,875;1,5)	=	767,79\$
@VDB(10000;600;10;8,875;9,875;1,5)	=	767,79\$
@VDB(10000;600;10;9,875;10,1,5)	=	95,97\$
Amortización total (coste menos valor_residual)		9.400,00\$

Funciones @ similares

@DDB calcula la cuota de amortización por el método de los saldos decrecientes al 200% (Double Declining Balance). @SLN la calcula por el método de amortización lineal y @SYD por el método de los números dígitos.

@CONSULV

@CONSULV(*x*; *rango*; *columna*) busca el contenido de una celda en una columna determinada de una tabla vertical de consulta.

Argumentos

x puede ser un valor o una cadena, dependiendo del contenido de la primera columna de la tabla vertical de consulta.

<u>Primera columna</u>	<u>x</u>
Valores	Cualquier valor mayor o igual que el primer valor contenido en <i>rango</i> . Si es menor, @CONSULV dará como resultado ERR. Si <i>x</i> es mayor que el último valor de la primera columna de <i>rango</i> , @CONSULV se detiene en la última celda de <i>columna</i> y da como resultado el contenido de dicha celda.
Rótulos	Texto encerrado entre comillas (" "), una fórmula cuyo resultado sea un texto, o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo. Si <i>x</i> no coincide exactamente con el contenido de ninguna celda de la primera columna de <i>rango</i> , @CONSULV dará como resultado el valor ERR.

rango representa la posición de la tabla vertical de consulta y puede ser cualquier dirección o nombre de rango. Si se trata de un rango tridimensional, 1-2-3 utilizará solamente la primera hoja del *rango*.

columna representa un número offset correspondiente a la posición que ocupa la columna en *rango*.

Notas

@CONSULV comparará *x* con cada celda de la primera columna de la tabla. Cuando 1-2-3 localice una celda en la primera columna que contenga *x* (o, si *x* es un valor, el valor más aproximado sin superarlo), se desplazará por esa fila el número de columnas especificado en *columna* y dará como resultado el contenido de la celda en la que se detenga.

Ejemplos

@CONSULV

Funciones @ similares

@CONSULH busca el contenido de una celda en una tabla horizontal de consulta. @INDICE busca el contenido de una celda cuando el usuario especifica los números offset de columna y fila. @ELEGIR da como resultado el contenido de una celda de una tabla de consulta formada por una sola fila. @BUSCARCONTENIDO busca la posición que ocupa en un rango la celda cuyo contenido se especifica. @INTERSECCION busca en un rango el contenido de una celda especificada mediante el título de la columna, la fila y la hoja donde se encuentra.

@BUSCARMAXIMO da como resultado una referencia absoluta a la celda que contenga el valor más alto de una lista de rangos. @BUSCARMINIMO da como resultado una referencia absoluta a la celda que contenga el valor más bajo de una lista de rangos.

@DIASEMANA

@DIASEMANA(*número_fecha*) calcula el día de la semana correspondiente a *número_fecha* y lo muestra como un entero del 0 (lunes) al 6 (domingo).

Argumentos

número_fecha es un número de fecha.

Ejemplos

DIASEMANA(@FECHA(91;7;3)) = 2 (miércoles).

Funciones @ similares

@MES calcula el mes correspondiente a un número de fecha. @AÑO calcula el año correspondiente a un número de fecha.

@AÑO

@AÑO(*número_fecha*;*[tipo]*) extrae la parte correspondiente al año, es decir, un entero del 0 (año 1900) al 199 (año 2099), del argumento *número_fecha*.

Argumentos

número_fecha puede ser un número entero, o la dirección o el nombre de una celda que contenga un número entero del 1 (1 de enero de 1900) al 73050 (31 de diciembre del 2099).

tipo puede ser 0 ó 1. Si *tipo* es 1, **@AÑO** dará como resultado el año en forma de número con cuatro dígitos. Si el *tipo* es 0 o si se omite, **@AÑO** dará como resultado el número offset del año a partir de 1900 (por ejemplo, **@AÑO** dará como resultado el valor 123 para representar el año 2023).

Notas

@AÑO puede proporcionar también el argumento *año* para otras funciones **@** de fecha que se basan en fechas previamente calculadas.

Ejemplos

@AÑO(20181) = 55, porque el número de fecha 20181 corresponde al 02-Abr-55.

@AÑO(@ACTUAL) = al año actual

@AÑO(@VALFECHA("14-Feb-92")) = 92

Funciones @ similares

@DIA extrae el día del mes (del 1 al 31) y **@MES** extrae el mes (del 1 al 12), a partir de un número de fecha.

@RENTABILIDAD

@RENTABILIDAD(*negociación*; *vencimiento*; *cupón*; *precio*; [*amortización*; *frecuencia*; *base*]) calcula la rentabilidad de activos financieros que proporcionan pagos periódicos de intereses.

Argumentos

negociación es la fecha en que se negocia o transmite el activo. Debe ser un número de fecha.

vencimiento es la fecha de vencimiento o amortización del activo. Debe ser un número de fecha. Si *vencimiento* es menor o igual que *negociación*, @RENTABILIDAD dará como resultado ERR.

cupón es el tipo de interés anual del cupón del activo. Puede ser cualquier valor positivo o 0.

precio es el precio del activo por cada 100\$ de valor nominal. Puede ser cualquier valor positivo.

amortización es un argumento opcional que permite especificar el valor de amortización (que se cobrará al vencimiento) del activo por cada 100\$ de valor nominal. *amortización* puede ser cualquier valor positivo o 0. Si se omite el argumento *amortización*, 1-2-3 utilizará el valor 100.

frecuencia es un argumento opcional que permite especificar el número de pagos de cupón por año. *frecuencia* debe ser cualquiera de los valores indicados en la tabla siguiente.

<u><i>frecuenci</i></u> <u><i>a</i></u>	<u>Frecuencia de los pagos de cupón</u>
1	Anual
2	Semestral; es el valor predeterminado si se omite este argumento
3	Trimestral
12	Mensual

base es un argumento opcional que permite especificar el tipo de base para el recuento de los días que va a utilizarse.

<u><i>base</i></u>	<u>Base para el recuento de los días</u>
0	30/360; es el valor predeterminado si se omite este argumento
1	Real/real
2	Real/360
3	Real/365
4	Europeo 30/360

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Ejemplos

Un bono, cuyo vencimiento es el 1 de diciembre de 1998, se negocia el 1 de julio de 1993. El tipo de interés del cupón semestral es del 5,50%, su precio es de 99,50\$ y el valor de amortización es de 100\$. El recuento de los días se efectúa de acuerdo con la base 30/360. Para determinar la rentabilidad del bono:

@RENTABILIDAD(@FECHA(93;7;1);@FECHA(98;12;1);0,055;99,5;100;2;0) = 0,056072

Funciones @ similares

@ACUMULADO calcula el interés acumulado para activos financieros que proporcionan pagos periódicos de intereses. @PRECIO calcula el precio por cada 100\$ de valor nominal para activos financieros que proporcionan pagos periódicos de intereses. @DURACION calcula la duración anual para activos financieros que proporcionan pagos periódicos de intereses. @DURACIONM calcula la duración anual modificada para activos financieros que proporcionan pagos periódicos de intereses.

@RENTABILIDAD2 da como resultado la rentabilidad de activos financieros que proporcionan pagos periódicos de intereses, según el sistema utilizado en Japón.

Ejemplo de @TESTJI (Verificación de bondad de ajuste)

@TESTJI(A2..A9;B2..B9) = 0,996882

A	-----	A	-----	B	-----
1		Observados		Esperados	
2		25		23	
3		34		35	
4		87		91	
5		91		88	
6		34		34	
7		23		22	
8		56		60	
9		70		68	

Ejemplo de @TESTJI (Verificación de independencia)

@TESTJI(B3..C5) = 0,080809

A	---	A	-----	B	-----	C	--
1			Áspero			Suave	
2							
3	Mucho		42			51	
4	Normal		25			37	
5	Poco		85			68	

Ejemplo de @BMEDIA

En la tabla de base de datos siguiente (denominada VENTAS) se incluye una relación de las viviendas vendidas por una inmobiliaria en tres ciudades durante los meses de abril y mayo. En el campo PAGADO de dicha tabla se indican los precios de venta en dólares. La fórmula siguiente permite determinar el precio medio de las viviendas vendidas en Córdoba:

@BMEDIA(VENTAS;"PAGADO";CIUDAD="Córdoba") = 365.666,67\$

A	-----	B	-----	C	-----
1	DIRECCION	CIUDAD	PAGADO		
2	Arenal, 12	Córdoba	720.000\$		
3	Los Olivos, 5	Guadalajara	318.000\$		
4	Madrid, 34	Veracruz	332.000\$		
5	Ebro, 65	Córdoba	278.000\$		
6	Mayor, 17	Córdoba	160.000\$		
7	Trafalgar, 3	Córdoba	227.000\$		
8	Arteaga, 87	Guadalajara	397.000\$		
9	Gran Vía, 28	Guadalajara	303.000\$		
10	Alminares, 6	Córdoba	669.000\$		
11	Luna, 15	Córdoba	140.000\$		

Ejemplo de @BCUENTA

En la tabla de base de datos siguiente (denominada VENTAS_ABR) se incluye una relación de los pisos vendidos por una inmobiliaria durante el mes de abril. En el campo CALEF de dicha tabla se indica el tipo de calefacción con que cuenta cada piso. La fórmula siguiente permite determinar el número de pisos que cuentan con calefacción a gas:

@BCUENTA(VENTAS_ABR;"CALEF";CALEF="Gas") = 4

A	-----	A	-----	B	-----	C	-----	D	--
1	DIRECCION			DORM		CALEF		PRECIO	
2	Fuencarral, 56			5		Gas		290.000\$	
3	Ibiza, 80			5		Gasóleo		105.000\$	
4	Galileo, 88			2		Gas		135.000\$	
5	Orense, 24			4		Eléctrica		128.000\$	
6	Oca, 52			2		Gas		174.000\$	
7	Isaac Peral, 8			3		Gas		195.000\$	

Ejemplo de @BEXTRAER

En la tabla de base de datos siguiente (denominada VENTAS) se incluye una relación de las viviendas vendidas por una inmobiliaria en tres ciudades durante los meses de abril y mayo. En el campo COM_PG de dicha tabla se indican las comisiones recibidas por los distintos agentes inmobiliarios. La fórmula siguiente permite determinar la comisión recibida por el vendedor de la vivienda situada en Gran Vía, 28:

@BEXTRAER(VENTAS;"COM_PG";DIRECCION="Gran Vía, 28") = 12.120\$

A	-----	A	-----	B	-----	C	-----
1	DIRECCION		AGENTE		COM_PG		
2	Arenal, 12		A. Téllez		28.800\$		
3	Los Olivos, 5		M. Martín		12.720\$		
4	Madrid, 34		A. Téllez		19.920\$		
5	Ebro, 65		J. Pérez		11.120\$		
6	Mayor, 17		E. Pinto		9.600\$		
7	Trafalgar, 3		M. Martín		13.620\$		
8	Arteaga, 87		E. Pinto		15.880\$		
9	Gran Vía, 28		A. Téllez		12.120\$		

Ejemplo de @BMAX

En la tabla de base de datos siguiente (denominada VENTAS) se incluye una relación de las viviendas vendidas por una inmobiliaria en tres ciudades durante los meses de abril y mayo. En el campo PAGADO de dicha tabla se indican los precios de venta en dólares. La fórmula siguiente permite determinar el precio máximo pagado por una vivienda en Córdoba:

@BMAX(VENTAS;"PAGADO";CIUDAD="Córdoba") = 720.000\$

A	-----	A	-----	B	-----	C	-----
1	DIRECCION		CIUDAD		PAGADO		
2	Arenal, 12		Córdoba		720.000		
3	Los Olivos, 5		Guadalajara		318.000		
4	Madrid, 34		Veracruz		800.000		
5	Ebro, 65		Córdoba		278.000		
6	Mayor, 17		Córdoba		160.000		
7	Trafalgar, 3		Córdoba		227.000		
8	Arteaga, 87		Guadalajara		397.000		
9	Gran Vía, 28		Guadalajara		303.000		
10	Alminares, 6		Córdoba		669.000		
11	Luna, 15		Córdoba		140.000		

Ejemplo de @BMIN

En la tabla de base de datos siguiente (denominada VENTAS) se incluye una relación de las viviendas vendidas por una inmobiliaria en tres ciudades durante los meses de abril y mayo. En el campo PAGADO de dicha tabla se indican los precios de venta en dólares. La fórmula siguiente permite determinar el precio mínimo pagado por una vivienda en Córdoba:

@BMIN(VENTAS;"PAGADO";CIUDAD="Córdoba") = 130.000\$

A	----- A -----	B -----	C -----
1	DIRECCION	CIUDAD	PAGADO
2	Arenal, 12	Córdoba	720.000\$
3	Los Olivos, 5	Guadalajara	318.000\$
4	Madrid, 34	Veracruz	332.000\$
5	Ebro, 65	Córdoba	278.000\$
6	Mayor, 17	Córdoba	160.000\$
7	Trafalgar, 3	Córdoba	227.000\$
8	Arteaga, 87	Guadalajara	397.000\$
9	Gran Vía, 28	Guadalajara	130.000\$
10	Alminares, 6	Córdoba	669.000\$
11	Luna, 15	Córdoba	140.000\$

Ejemplo de @BSTD y @BSTDM

La tabla siguiente contiene datos sobre la altura y el peso de diez individuos seleccionados al azar. La fórmula siguiente permite determinar la desviación típica de las alturas de aquellos individuos que pesan más de 75 kg.

@BSTD(A1..B11;"ALTURA";PESO>75) = 4,716991

Si suponemos que los diez individuos constituyen una muestra seleccionada al azar entre un grupo más numeroso de individuos.

@BSTDM(A1..B11;"ALTURA";PESO>75) = 5,446712

	A -----	A -----	B --
		ALTURA (cms)	PESO (kgs)
1		191	73
2		188	86
3		175	68
4		175	76
5		180	77
6		180	73
7		188	75
8		173	68
9		178	71
10		179	87
11			

Ejemplo de @BSUMA

En la tabla de base de datos siguiente (denominada VENTAS) se incluye una relación de las viviendas vendidas por una inmobiliaria en tres ciudades durante los meses de abril y mayo. En el campo COM_PG de dicha tabla se indican las comisiones recibidas por los distintos agentes inmobiliarios, en dólares. La fórmula siguiente permite determinar el total de las comisiones recibidas por el agente E. Pinto:

@BSUMA(VENTAS;"COM_PG";AGENTE="E. Pinto") = 25.480\$

A ---- A ----- B ----- C -----

1	DIRECCION	AGENTE	COM_PG
2	Arenal, 12	A. Téllez	28.800\$
3	Los Olivos, 5	M. Martín	12.720\$
4	Madrid, 34	A. Téllez	19.920\$
5	Ebro, 65	J. Pérez	11.120\$
6	Mayor, 17	E. Pinto	9.600\$
7	Trafalgar, 3	M. Martín	13.620\$
8	Arteaga, 87	E. Pinto	15.880\$
9	Gran Vía, 28	A. Téllez	12.120\$

Ejemplo de @BVAR y @BVARM

La tabla siguiente contiene datos sobre la altura y el peso de diez individuos seleccionados al azar. La fórmula siguiente permite determinar la varianza de los pesos de aquellos individuos que miden más de 180 cm.

@BVAR(A1..B11;"PESO";ALTURA>180) = 32,66667

Si suponemos que los diez individuos constituyen una muestra seleccionada al azar entre un grupo más numeroso de individuos.

@BVARM(A1..B11;"PESO";ALTURA>180) = 49

A	A	B
1	ALTURA (cms)	PESO (kgs)
2	191	73
3	188	86
4	175	68
5	175	76
6	180	77
7	180	73
8	188	75
9	173	68
10	178	71
11	179	87

Ejemplo de @TESTF

@TESTF(A2..A13;B2..B15) = 0,157348

A	-----	A	-----	B	-----
1		Muestral		Muestra	
2	84,5			1,65	
3	80,7			4,58	
4	34,5			42,6	
5	54,6			4,37	
6	50,5			30,8	
7	33,7			97,7	
8	46,8			87,2	
9	47,6			40,7	
10	22,8			38,4	
11	15,5			10,6	
12	60,6			56,3	
13	80,5			70,5	
14				9,04	
15				97,3	

Ejemplo de @TOTAL

La fórmula @TOTAL situada en la celda A10 calcula la suma de los resultados de todas las celdas de A1..A8 que contienen fórmulas con @SUBTOTAL (A4 y A8).

A	---	A	-----	B	-----
1		10			
2		15			
3					
4		25		@SUBTOTAL (A1..A2)	
5					
6		20			
7		25			
8		45		@SUBTOTAL (A6..A7)	
9					
10		70		@TOTAL (A1..A8)	

Ejemplo de @BUSCARCONTENIDO

La posología de un medicamento se realiza de acuerdo con el peso del paciente. El peso de un paciente en libras (125), se encuentra en la celda denominada PESO_PACIENTE.

@INDICE(A2..C7;2;@BUSCARCONTENIDO(PESO_PACIENTE;A2..C7;1)) = 2

A	----	A	-----	B	-----	C	-----
1		Libras		Kilos		Comprimidos	
2		50		22,5		1,5	
3		100		45,5		2,0	
4		150		68,0		2,5	
5		200		90,5		3,0	
6		250		113,5		3,5	
7		300		136,0		4,0	

Ejemplo de @TESTT

@TESTT(A2..A13;B2..B13) = 0,050022

A	-----	A	-----	B	-----
1		Muestral		Muestra	
2	84,5			65,1	
3	80,7			85,4	
4	34,5			62,4	
5	54,6			73,4	
6	50,5			80,3	
7	33,7			66,7	
8	46,8			87,2	
9	47,6			70,4	
10	22,8			30,2	
11	15,5			60,1	
12	60,6			56,3	
13	80,5			70,5	

Ejemplo de @INTERSECCION

En la tabla de consulta llamada TARIFAS (A2..E7), se muestran las tarifas postales, en dólares, correspondientes al envío de un paquete a diferentes ciudades.

@INTERSECCION(TARIFAS;"Nueva York";1) da como resultado 9,29\$, la tarifa postal correspondiente a un paquete de 1 Kg con destino a Nueva York.

@INTERSECCION(TARIFAS;"París";5) da como resultado 29,00\$, la tarifa postal correspondiente a un paquete de 5 Kg con destino a París.

A	----	A	-----	B	-----	C	-----	D	-----	E	-----
1						PAQUETE	CON	DESTINO		A	-----
2		Kg		Londres		París		Frankfurt		Nueva York	
3		1		\$18.36		\$19.33		\$20.12		\$ 9.29	
4		2		\$20.32		\$21.66		\$22.03		\$11.25	
5		3		\$22.44		\$23.88		\$24.00		\$13.25	
6		4		\$24.14		\$25.26		\$25.75		\$16.85	
7		5		\$28.32		\$29.00		\$29.80		\$19.54	

@TESTJI

@TESTJI(*rango1*;*[rango2]*) realiza un test o verificación de independencia mediante el parámetro estadístico ji cuadrado con los datos contenidos en *rango1*, o una verificación de bondad de ajuste mediante el parámetro estadístico ji cuadrado con los datos contenidos en *rango1* y en *rango2*.

Argumentos

rango1 y *rango2* deben ser rangos del mismo tamaño. Si *rango1* y *rango2* no tienen el mismo tamaño, @TESTJI dará como resultado ERR.

Notas

@TESTJI calcula el valor aproximado de la probabilidad asociada a una verificación mediante ji cuadrado con un margen de error de $\pm 3 \cdot 10^{-7}$.

Ejemplos

@TESTJI: Verificación de independencia

@TESTJI: Verificación de bondad de ajuste

Funciones @ similares

@DISTJI calcula la distribución ji cuadrado. @TESTF realiza una verificación mediante el parámetro *F*, @TESTT realiza una verificación mediante el parámetro *t* de Student y @TESTZ realiza una verificación mediante el parámetro *z*.

@CRITBINOMIAL

@CRITBINOMIAL(*intentos*; *probabilidad*; *alfa*) calcula el valor máximo para el cual la distribución binomial acumulativa es menor o igual que *alfa*.

Argumentos

intentos representa el número de intentos de un experimento de Bernoulli; puede ser cualquier entero positivo o 0.

probabilidad representa la probabilidad de que se produzca un acierto en un único intento de un experimento de Bernoulli; puede ser cualquier valor entre 0 y 1.

alfa representa la probabilidad del criterio; puede ser cualquier valor entre 0 y 1.

Notas

@CRITBINOMIAL calcula el resultado con un margen de error de $\pm 3 \cdot 10^{-7}$.

Ejemplos

Suponga que es el gerente de una pequeña fábrica de filtros de aceite. Los filtros se fabrican en lotes de 100 unidades; la probabilidad de que un filtro cualquiera carezca de defectos es del 85%. Si desea estar seguro en un 99% de que al menos un determinado número de filtros carecen de defectos, puede utilizar la siguiente función.

@CRITBINOMIAL(100;0,85;0,01) = 76 (número de filtros que carecen de defectos)

Funciones @ similares

@BINOMIAL calcula la función de probabilidad binomial o la distribución binomial acumulativa. **@COMBIN** calcula el coeficiente binomial. **@PERMUT** calcula el número de permutaciones que pueden realizarse con un número determinado de elementos de un conjunto.

@BMEDIA

@BMEDIA(*entrada;campo;*[*criterios*]) calcula la media aritmética de los valores contenidos en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *criterios* especificados.

Argumentos

Consulte el tema [Argumentos de las funciones @ de bases de datos.](#)

Ejemplos

[@BMEDIA](#)

Funciones @ similares

[@MEDIA](#) y [@MEDIAPURA](#) calculan la media aritmética de los valores de una lista.

@BCUENTA, @BCUENTAPURA

@BCUENTA(*entrada;campo;*[*critérios*]) cuenta las celdas no vacías existentes en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *critérios* especificados.

@BCUENTAPURA(*entrada;campo;*[*critérios*]) cuenta las celdas que contienen valores existentes en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *critérios* especificados.

Argumentos

Consulte el tema [Argumentos de las funciones @ de bases de datos.](#)

Ejemplos

[@BCUENTA](#)

Funciones @ similares

[@CUENTA](#) y [@CUENTAPURA](#) cuentan las celdas existentes en una lista de rangos.

@BEXTRAER

@BEXTRAER(*entrada;campo;*[*critérios*]) extrae un valor o un rótulo de un *campo* de una tabla de base de datos perteneciente al registro que cumple los *critérios* especificados.

Argumentos

Consulte el tema [Argumentos de las funciones @ de bases de datos](#).

Notas

Si existe más de un registro que cumple los *critérios* especificados, **@BEXTRAER** dará como resultado **ERR**.

Utilice **@BEXTRAER** cuando necesite obtener un valor de un único registro que cumpla unos *critérios* específicos, como por ejemplo, el código de identificación de un determinado empleado de la empresa. **@BEXTRAER** permite obtener este tipo de información automáticamente para su uso en una macro, como argumento en otra función **@**, o como variable en una fórmula.

Ejemplos

[@BEXTRAER](#)

Funciones @ similares

[@CONSULH](#) y [@CONSULV](#) dan como resultado el contenido de una celda determinada incluida en una tabla de consulta horizontal o vertical, respectivamente. [@ELEGIR](#) halla una entrada de una lista. [@INDICE](#) da como resultado el contenido de una celda situada en la intersección de una columna, una fila y una hoja de trabajo determinadas de un rango. [@INTERSECCION](#) da como resultado el contenido de una celda de una tabla basándose en los títulos de la hoja, la fila y la columna donde se encuentra. [@@](#) da como resultado el contenido de una celda especificada en otra celda.

@BMAX

@BMAX(*entrada;campo[criterios]*) selecciona el valor máximo entre los contenidos en un *campo* de una tabla de base de datos, pero únicamente tiene en cuenta los registros que cumplen los *criterios* especificados.

Argumentos

Consulte el tema [Argumentos de las funciones @ de base de datos.](#)

Notas

Puede utilizar @BMAX para hallar la fecha o la hora más reciente entre las incluidas en una lista de fechas o de horas.

Ejemplos

[@BMAX](#)

Funciones @ similares

[@MAX](#) y [@MAXPURA](#) hallan el valor máximo de una lista.

@BMIN

@BMIN(*entrada;campo;[criterios]*) selecciona el valor mínimo entre los contenidos en un *campo* de una tabla de base de datos, pero únicamente tiene en cuenta los registros que cumplen los *criterios* especificados.

Argumentos

Consulte el tema [Argumentos de las funciones @ de base de datos.](#)

Notas

Puede utilizar @BMIN para hallar la fecha o la hora menos reciente entre las incluidas en una lista de fechas o de horas.

Ejemplos

[@BMIN](#)

Funciones @ similares

[@MIN](#) y [@MINPURA](#) hallan el valor mínimo entre los contenidos en una lista.

@BSTD, @BSTDM

[@BSTD](#)(*entrada;campo;*[*criterios*]) calcula la desviación típica de una población constituida por los valores contenidos en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *criterios* especificados.

[@BSTDM](#)(*entrada;campo;*[*criterios*]) calcula la desviación típica de una muestra constituida por los valores contenidos en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *criterios* especificados.

Argumentos

Consulte el tema [Argumentos de las funciones @ de bases de datos](#).

Notas

[@BSTD](#) utiliza el método n para calcular la desviación típica de una población. El método n asume que los valores seleccionados comprenden toda la población. Si los valores representan sólo una muestra de la población, la desviación típica resulta poco fiable a causa de los errores cometidos al obtener la muestra.

La desviación típica es la raíz cuadrada de la varianza de todos los valores con relación a la media aritmética.

Ejemplos

[@BSTD](#) y [@BSTDM](#)

Funciones @ similares

[@STD](#) y [@STDPURA](#) calculan la desviación típica de una serie de valores de un rango que constituyen una población. [@STDM](#) y [@STDMPURA](#) calculan la desviación típica de una serie de valores que constituyen una muestra. [@BVAR](#) calcula la varianza de una población constituida por los valores contenidos en un campo de una tabla de base de datos pertenecientes a los registros que cumplen determinados criterios.

@BSUMA

@BSUMA(*entrada;campo;*[*critérios*]) calcula la suma de los valores contenidos en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *critérios* especificados.

Argumentos

Consulte el tema [Argumentos de las funciones @ de bases de datos.](#)

Ejemplos

@BSUMA

Funciones @ similares

@SUMA calcula la suma de los valores de una lista. @SUMANEGATIVOS suma únicamente los valores negativos de una lista. @SUMAPOSITIVOS suma únicamente los valores positivos de una lista.

@BVAR, @BVARM

@BVAR(*entrada;campo;*[*criterios*]) calcula la varianza de una población constituida por los valores contenidos en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *criterios* especificados.

@BVARM(*entrada;campo;*[*criterios*]) calcula la varianza de una muestra constituida por los valores contenidos en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *criterios* especificados.

Argumentos

Consulte el tema [Argumentos de las funciones @ de bases de datos.](#)

Notas

La varianza mide el grado de dispersión de los valores de una lista con relación a su media aritmética. Cuanto menor sea la varianza, más se aproximarán los valores a la media y más fiable será ésta. Cuando todos los valores de la lista son iguales, la varianza es 0. La varianza es necesaria en ciertas pruebas estadísticas de análisis de varianza.

@BVAR utiliza el método *n*, o de la población, para calcular la varianza. Dicho método considera que los valores seleccionados comprenden el total de la población. Si los valores representan solamente una muestra de dicha población, la varianza resultará poco fiable a causa de los errores cometidos al obtener la muestra.

La varianza es el cuadrado de la desviación típica.

Ejemplos

[@BVAR](#) y [@BVARM](#)

Funciones @ similares

[@VAR](#) y [@VARPURA](#) calculan la varianza de los valores de una lista que constituyen una población. [@BSTD](#) calcula la desviación típica de una población constituida por los valores contenidos en un campo de una tabla de base de datos pertenecientes a los registros que cumplen determinados criterios.

@TESTF

@TESTF(*rango1*;*rango2*) realiza un test o verificación de hipótesis mediante el parámetro estadístico F y da como resultado la probabilidad asociada.

Argumentos

rango1 y *rango2* son los rangos en que se encuentran los datos con los que se desea realizar la verificación y no tienen por qué ser del mismo tamaño.

Notas

@TESTF calcula el valor aproximado de la probabilidad asociada a una verificación mediante F con un margen de error de $\pm 3 \cdot 10^{-7}$.

Utilice **@TESTF** para determinar si dos muestras tienen varianzas distintas.

Ejemplos

@TESTE

Funciones @ similares

@DISTF calcula la distribución F . **@TESTJI** realiza una verificación mediante el parámetro estadístico ji cuadrado, **@TESTI** realiza una verificación mediante el parámetro estadístico t de Student, y **@TESTZ** realiza una verificación mediante el parámetro estadístico z .

@TOTAL

@TOTAL(*lista*) calcula la suma de todas las celdas de *lista* que contienen la función @SUBTOTAL en sus fórmulas.

Argumentos

lista puede ser cualquier combinación de rangos. Utilice separadores de argumentos para separar los elementos de *lista*.

Consulte también el tema Argumentos de las funciones @ estadísticas.

Ejemplos

@TOTAL

Funciones @ similares

@SUMA calcula la suma de los valores de una lista. @BSUMA calcula la suma de los valores contenidos en un campo de una tabla de base de datos pertenecientes a los registros que cumplen determinados criterios.

@SUMANEGATIVOS suma sólo los valores negativos de una lista. @SUMAPOSITIVOS suma sólo los valores positivos de una lista.

@ESARCHIVO

`@ESARCHIVO(nombre_archivo;[tipo])` comprueba si *nombre_archivo* es uno de los archivos existentes en la memoria o en el disco. Si encuentra *nombre_archivo*, `@ESARCHIVO` dará como resultado 1 (verdad); si no encuentra *nombre_archivo*, `@ESARCHIVO` dará como resultado 0 (falso).

Argumentos

nombre_archivo es una cadena de texto con el nombre completo, incluida la extensión, del archivo que se desea comprobar. Será preciso especificar la ruta de acceso en *nombre_archivo*, salvo en el caso de que desee que 1-2-3 busque el archivo en el directorio actual.

tipo permite especificar si se desea buscar *nombre_archivo* en la memoria o en el disco. Si *tipo* es igual a 0, 1-2-3 buscará *nombre_archivo* en la memoria; si *tipo* es igual a 1, 1-2-3 lo buscará en el disco. Si se omite el argumento *tipo*, 1-2-3 utilizará el valor 0.

Ejemplos

`@ESARCHIVO("C:\123W\PRESUP\RENTA_93.123";1) = 1`, si el archivo RENTA_93.123 se encuentra en el directorio C:\123W\PRESUP.

@BUSCARCONTENIDO

@BUSCARCONTENIDO(*contenido_celda*; *rango*; [*tipo*]) da como resultado la posición que ocupa en *rango* la celda cuyo contenido es igual a *contenido_celda*. **@BUSCARCONTENIDO** indica la posición de la celda en forma de número offset.

Argumentos

contenido_celda puede ser un valor o una cadena de texto. Si *contenido_celda* es una cadena de texto, pueden utilizarse caracteres comodín.

rango es el nombre o la dirección de un rango.

tipo es un argumento opcional que permite especificar el modo en que 1-2-3 debe comparar *contenido_celda* con el contenido de las celdas de *rango*.

<u><i>tipo</i></u>	<u>@BUSCARCONTENIDO da como resultado la posición relativa de</u>
0	La primera celda cuyo contenido sea igual a <i>contenido_celda</i> .
1	La celda que contiene el máximo valor que es menor o igual que <i>contenido_celda</i> ; es el valor predeterminado si se omite este argumento. Se considera que los elementos de <i>rango</i> están en orden ascendente.
2	La celda que contiene el mínimo valor que es mayor o igual que <i>contenido_celda</i> . Se considera que los elementos de <i>rango</i> están en orden descendente.

Notas

1-2-3 efectuará la búsqueda en *rango* de arriba a abajo en cada columna y de izquierda a derecha. Si se especifica un rango que comprende varias hojas, 1-2-3 efectuará la búsqueda en la primera hoja del rango, y después continuará con la segunda y sucesivas hasta que encuentre el *contenido_celda* especificado o el final del rango.

Si 1-2-3 no puede hallar *contenido_celda*, **@BUSCARCONTENIDO** dará como resultado el valor ERR.

Si *tipo* es 1 y la primera celda de *rango* contiene un valor que es mayor que *contenido_celda*, **@BUSCARCONTENIDO** dará como resultado ERR.

Si *tipo* es 2 y la primera celda de *rango* contiene un valor que es menor que *contenido_celda*, **@BUSCARCONTENIDO** dará como resultado ERR.

Ejemplos

@BUSCARCONTENIDO

Funciones @ similares

@CONSULH y **@CONSULV** buscan el contenido de una celda en una tabla de consulta horizontal o vertical, respectivamente. **@INDICE** busca el contenido de una celda basándose en los números offset especificados para la columna y la fila donde se encuentra. **@ELEGIR** busca el contenido de una celda en una tabla de consulta de una única fila. **@BUSCARMAXIMO** da como resultado una referencia absoluta a la celda que contenga el valor más alto de una lista de rangos. **@BUSCARMINIMO** da como resultado una referencia absoluta a la celda que contenga el valor más bajo de una lista de rangos.

@NORMAL

@NORMAL(x ;[$media$; std ; $tipo$]) calcula la función de la distribución normal para el valor x

Argumentos

x es el límite superior para el valor de la distribución normal acumulativa. Puede ser cualquier valor; si x es negativo, 1-2-3 lo convertirá en su valor absoluto (positivo).

$media$ es un argumento opcional que permite especificar la media de la distribución. Puede ser cualquier valor positivo o 0. Si se omite el argumento $media$, 1-2-3 utilizará el valor 0.

std es un argumento opcional que permite especificar la desviación típica de la distribución. Puede ser cualquier valor positivo o 0. Si se omite el argumento std , 1-2-3 utilizará el valor 1.

$tipo$ es un argumento opcional que permite especificar la función que @NORMAL debe calcular.

<u>tipo</u>	<u>@NORMAL calcula</u>
0	La función de distribución acumulativa; es el valor predeterminado si se omite este argumento
1	La distribución acumulativa inversa
2	La función densidad de probabilidad

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Notas

@NORMAL calculará el valor aproximado de la función de distribución acumulativa con un margen de error de $\pm 7,5 \cdot 10^{-8}$ y el de la distribución acumulativa inversa con un margen de error de $\pm 4,5 \cdot 10^{-4}$

Ejemplos

@NORMAL(1,96) = 0,9750

@NORMAL(0,975;0;1;1) = 1,96

@NORMAL(1,96;0;1;2) = 0,058441

Funciones @ similares

@DISTJI calcula la distribución ji cuadrado. @DISTF calcula la distribución F. @POISSON calcula la distribución de Poisson. @DISTT calcula la distribución t de Student.

@NOMBRERANGO

@NOMBRERANGO(*celda*) arroja como resultado el nombre del rango donde se encuentra *celda*.

Argumento

celda es la dirección de una celda o el nombre de un rango con una sola celda.

Notas

Si se especifica una celda que pertenece a varios rangos que se solapan, 1-2-3 dará como resultado el primer nombre de rango que encuentre.

Si *celda* no se encuentra en ningún rango con nombre, @NOMBRERANGO dará como resultado ERR.

La función @NOMBRERANGO sólo podrá utilizarse con libros que se encuentren en memoria.

Ejemplos

@NOMBRERANGO(A:A2) da como resultado VENTAS si A:A2 se encuentra en el rango denominado VENTAS.

@INFOESCENARIO, @INFOVERSION

@INFOESCENARIO(*opción;grupo_versiones:[autor]*) proporciona información relativa a un grupo de versiones.

@INFOVERSION(*opción;rango_version;nombre:[autor]*) proporciona información relativa a una versión.

Argumentos

opción es una cadena de texto (sin acentos) que permite indicar a 1-2-3 el tipo de información que debe suministrar.

<u>opción</u>	<u>1-2-3 da como resultado</u>
autor	El nombre de la persona que creó la versión o el grupo de versiones
revisor	El nombre de la persona que modificó por última vez la versión o el grupo de versiones
creacion	La fecha y la hora en que se creó la versión o el grupo de versiones, en forma de <u>número de fecha</u> y de <u>hora</u>
revision	La fecha y la hora en que se modificó por última vez la versión o el grupo de versiones, en forma de número de fecha y hora
comentario	El comentario de la versión o el grupo de versiones; 1-2-3 lo truncará si contiene más de 512 caracteres de un solo byte
oculto	0 (falso) si no se ha ocultado la versión o el grupo de versiones, o 1 (verdad) si se ha ocultado
proteccion	0 (falso) si no se ha protegido la versión o el grupo de versiones, o 1 (verdad) si se ha protegido

grupo_versiones es una cadena de texto que permite especificar el nombre de la versión o el grupo de versiones. Si hay más de una versión o un grupo de versiones con el mismo nombre, 1-2-3 utilizará la versión o el grupo de versiones de más reciente creación.

autor es una cadena de texto que permite especificar el nombre del usuario que creó la versión o el grupo de versiones. 1-2-3 empleará el argumento *autor* para determinar con mayor precisión cuál es la versión o el grupo de versiones que debe utilizar.

rango_version es el nombre del rango que contiene la versión. Debe ser un nombre de rango definido.

Ejemplos

@INFOESCENARIO("comentario";"Solución óptima";"Daniel Martín") dará como resultado el comentario correspondiente al último grupo de versiones cuyo nombre es Solución óptima y que fue creado por Daniel Martín.

@INFOESCENARIO("autor";"Ventas") dará como resultado el nombre del usuario que creó el último grupo de versiones cuyo nombre es Ventas.

@INFOVERSION("creacion";RANGOVENTAS;"Solución óptima";"Daniel Martín") dará como resultado la fecha y la hora en que Daniel Martín creó su última versión para RANGOVENTAS.

@INFOVERSION("revision";RANGOVENTAS;"Asuntos") dará como resultado la fecha y la hora en que se modificó por última vez la versión Asuntos de RANGOVENTAS.

@ULTIMOESCENARIO

@ULTIMOESCENARIO(*nombre_archivo*) da como resultado el nombre del grupo de versiones que se ha visualizado por última vez en un libro durante una sesión de 1-2-3.

Argumentos

nombre_archivo es el nombre completo, incluida la extensión, del archivo de libro que se desea examinar, especificado en forma de texto. Deberá especificar también la ruta de acceso como parte de *nombre_archivo*, a menos que desee que 1-2-3 efectúe la búsqueda en el directorio en el que se encuentre en ese momento.

Notas

Si no se ha visualizado ningún grupo de versiones en *nombre_archivo* durante la sesión actual de 1-2-3, @ULTIMOESCENARIO dará como resultado ERR.

Ejemplos

@ULTIMOESCENARIO("d:\123_97\hojas\julio.wk4") dará como resultado el nombre del grupo de versiones que se haya visualizado por última vez en el archivo de libro JULIO.WK4, que se encuentra en D:\123_97\HOJAS.

@ERRMEDIA

@ERRMEDIA(*rango*) calcula el error típico de la distribución muestral de la media para los valores contenidos en *rango*.

Argumentos

rango es un nombre de rango o una dirección.

Ejemplos

Suponga que el rango PRUEBA contiene los valores 2, 6, 8, 5, 3, 9, 1 y 2.

@ERRMEDIA(PRUEBA) = 1,052209

Funciones @ similares

@MEDIAGEOM calcula la media geométrica de los valores de una lista. **@MEDIARMON** calcula la media armónica de los valores de una lista. **@STD** y **@STDPURA** calculan la desviación típica de los valores de una lista.

@TESTT

@TESTT(*rango1*;*rango2*;*[tipo;colas]*) realiza un test o verificación de hipótesis mediante el parámetro estadístico *t* de Student con los datos contenidos en *rango1* y *rango2*, y da como resultado la probabilidad asociada.

Argumentos

rango1 y *rango2* son rangos que contienen valores.

tipo es un argumento opcional que permite especificar el tipo de verificación *t* que se desea realizar.

<u>tipo</u>	<u>1-2-3 realiza</u>
0	Una verificación <i>t</i> para muestras extraídas de poblaciones con la misma varianza; <i>rango1</i> y <i>rango2</i> no tienen por qué contener el mismo número de celdas; es el valor predeterminado si se omite este argumento
1	Una verificación <i>t</i> para muestras extraídas de poblaciones con varianzas distintas; <i>rango1</i> y <i>rango2</i> no tienen por qué contener el mismo número de celdas
2	Una verificación <i>t</i> para muestras emparejadas; <i>rango1</i> y <i>rango2</i> deben contener el mismo número de celdas

colas es un argumento opcional que permite especificar la dirección en que se desea realizar la verificación *t*.

<u>colas</u>	<u>1-2-3 realiza</u>
1	Una verificación <i>t</i> de una cola
2	Una verificación <i>t</i> de dos colas; es el valor predeterminado si se omite este argumento

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Notas

@TESTT calcula el valor aproximado de la probabilidad asociada a una verificación mediante *t* con un margen de error de $\pm 3 \cdot 10^{-7}$.

Ejemplos

@TESTT

Funciones @ similares

@DISTT calcula la distribución *t* de Student. @TESTJI realiza una verificación mediante el parámetro estadístico *ji* cuadrado, @TESTF realiza una verificación mediante el parámetro estadístico *F*, y @TESTZ realiza una verificación mediante el parámetro estadístico *z*.

@VERSIONACTUAL

@VERSIONACTUAL(*rango*) da como resultado el nombre de la versión actual de un *rango*.

Argumentos

rango es el nombre o la dirección del rango cuyo nombre de versión se desea averiguar.

Notas

Si no hay ninguna versión actual, @VERSIONACTUAL dará como resultado ERR.

Ejemplos

@VERSIONACTUAL(BENEFICIOS) dará como resultado el nombre de la versión actual del rango BENEFICIOS.

@VERSIONDATOS

@VERSIONDATOS(*opción*; *celda*; *rango_versión*; *nombre*; [*autor*]) da como resultado el contenido de una determinada celda de una versión.

Argumentos

opción es una de las cadenas de texto siguientes (sin acento) que sirven para indicar a 1-2-3 en qué forma debe suministrar el contenido de *celda*.

<u>opción</u>	<u>1-2-3 da como resultado</u>
formula	La fórmula contenida en la celda, en forma de rótulo, o bien <u>ERR</u> si la celda no contiene ninguna fórmula
valor	El resultado de la fórmula, si la celda contiene una fórmula El valor o el rótulo, si la celda contiene un valor o un rótulo Una <u>cadena vacía</u> si la celda está vacía

celda es el nombre o la dirección de la celda cuyo contenido se desea obtener. Debe encontrarse en *rango_versión*.

rango_versión es el nombre del rango que contiene la versión. Debe ser un nombre de rango definido.

nombre es una cadena de texto que permite especificar el nombre de la versión. Si hay más de una versión con el mismo *nombre*, 1-2-3 empleará la última que se haya creado.

autor es una cadena de texto que permite especificar el nombre del usuario que creó la versión. 1-2-3 utilizará el argumento *autor* para determinar con mayor precisión cuál es la versión que debe utilizar. Si un *autor* ha creado más de una versión con el mismo nombre, 1-2-3 utilizará la de más reciente creación.

Ejemplos

@VERSIONDATOS("formula";A:B12;VENTAS;"Solución óptima") dará como resultado la fórmula contenida en la celda A:B12 de la última versión creada con el nombre Solución óptima en el rango VENTAS.

@VERSIONDATOS("valor";A:B12;VENTAS;"Solución óptima";"Daniel Martín") dará como resultado el valor o el rótulo contenido en la celda A:B12 de la última versión creada por Daniel Martín con el nombre Solución óptima en el rango VENTAS.

@MEDIAPOND

@MEDIAPOND(*rango_datos*; *rango_proporciones*; [*tipo*]) calcula la media ponderada de los valores contenidos en *rango_datos*.

Argumentos

rango_datos y *rango_proporciones* son nombres o direcciones de rangos que contienen valores y deben tener el mismo tamaño y la misma forma.

Si *rango_datos* y *rango_proporciones* no tienen el mismo tamaño y la misma forma, @MEDIAPOND dará como resultado ERR.

tipo es un valor que determina el modo en que 1-2-3 calcula la media ponderada.

<u><i>tipo</i></u>	<u>1-2-3 divide entre</u>
0	El valor equivalente a la suma de los valores de <i>rango_proporciones</i> ; es el valor predeterminado si se omite este argumento.
1	El valor que indica el número de valores de <i>rango_datos</i> .

Ejemplos

En el ejemplo siguiente se utiliza @MEDIAPOND para calcular la media ponderada de las calificaciones obtenidas por un alumno en una asignatura compuesta por tres partes: Teoría (cuyo valor de estima en el 50% de la calificación final), Problemas (cuyo valor es del 30%) y Laboratorio (cuyo valor es del 20%). La calificación mínima es de 0 puntos y la máxima de 10.

El rango CALIFIC contiene las calificaciones obtenidas por el alumno en cada una de las partes de que se compone la asignatura y BAREMO contiene la contribución relativa de cada una de ellas a la calificación final:

CALIFIC	BAREMO
8	0,5
5	0,3
9	0,2

@MEDIAPOND(CALIFIC;BAREMO) = 7,3 puntos, es decir, la calificación final de acuerdo con el valor proporcional estimado para cada parte de la asignatura.

Funciones @ similares

@SUMAPRODUCTOS multiplica los valores contenidos en las celdas homólogas de varios rangos y, a continuación, suma los productos.

@LABORABLE

@LABORABLE(*fecha_inicial*; *días*; [*rango_festivos*; *fin_de_semana*]) calcula el número de fecha correspondiente a una fecha que es un determinado número de *días* anterior o posterior a *fecha_inicial*, excluyendo los fines de semana y, opcionalmente, los festivos o vacaciones.

Argumentos

fecha_inicial debe ser un número de fecha.

días debe ser un número entero. Si es positivo, 1-2-3 calculará dicho número de días después de *fecha_inicial*; si es negativo, 1-2-3 calculará dicho número de días antes de *fecha_inicial*.

rango_festivos es un argumento opcional que permite especificar los días festivos o de vacaciones que deben excluirse del cálculo efectuado por **@LABORABLE** y es el nombre o la dirección de un rango que contiene números de fecha.

Si se omite el argumento *rango_festivos*, 1-2-3 no excluirá los días festivos al efectuar el cálculo con **@LABORABLE**.

fin_de_semana es un argumento opcional que especifica los días que forman el fin de semana y, por lo tanto, no deben considerarse como laborables debe ser una cadena de texto que se vale de los números enteros de 0 (lunes) a 6 (domingo) para indicar los días que corresponden al fin de semana.

Por ejemplo, "45" indica que los viernes y los sábados son los días correspondientes al fin de semana. Si se omite el argumento *fin_de_semana*, 1-2-3 utilizará "56", lo que indica que considera que el fin de semana lo constituyen el sábado y el domingo. Si no desea especificar un valor para el fin de semana, indique 7.

Notas

Si desea utilizar el argumento *fin_de_semana* pero no *rango_festivos* deberá especificar una celda vacía para el argumento *rango_festivos*.

Ejemplos

Suponga que desea determinar la fecha 30 días laborables posteriores al martes, 1 de noviembre de 1994, teniendo en cuenta que los días 24 y 25 son festivos. En tal caso, deberá indicar dichas fechas en un rango llamado FESTIVOS. Para especificar los sábados y domingos como días pertenecientes al fin de semana omita el argumento *fin_de_semana*.

@LABORABLE(**@FECHA**(94;11;1);30;FESTIVOS) = 34683, es decir, el martes, 15 de diciembre de 1994.

Funciones @ similares

@DIAS360 y **@D360** calculan el número de días existentes entre dos números de fecha. **@LABORABLESNETOS** calcula el número de días existentes entre dos fechas sin contar fines de semana ni festivos. **@MESSIGUIENTE** calcula la fecha de un número de meses anterior o posterior a una fecha determinada.

@INTERSECCION

@INTERSECCION(*rango*; *título_columna*; *título_fila*; [*título_hoja*]) arroja como resultado el contenido de una celda de *rango* situada en la intersección especificada mediante *título_columna*, *título_fila* y, opcionalmente, *título_hoja*.

Argumentos

rango es la dirección o el nombre de un rango.

título_columna es el contenido de una celda de la primera fila de *rango*.

título_fila es el contenido de una celda de la primera columna de *rango*.

título_hoja es un argumento opcional con el contenido de la primera celda de *rango*.

título_columna, *título_fila* y *título_hoja* pueden ser valores o cadenas de texto.

Ejemplos

@INTERSECCION

Funciones @ similares

@ELEGIR busca una entrada en una lista. **@CONSULH** y **@CONSULV** buscan el contenido de una celda en una tabla de consulta horizontal o vertical, respectivamente. **@BUSCARCONTENIDO** da como resultado la posición relativa de una celda en un rango cuando se conoce el contenido de la misma. **@BUSCARMAXIMO** da como resultado una referencia absoluta a la celda que contenga el valor más alto de una lista de rangos.

@BUSCARMINIMO da como resultado una referencia absoluta a la celda que contenga el valor más bajo de una lista de rangos.

@TESTZ

@TESTZ(*rango1*; *media1*; *std1*; [*colas*; *rango2*; *media2*; *std2*]) realiza una verificación mediante el parámetro estadístico *z* con una o dos poblaciones y da como resultado la probabilidad asociada.

Argumentos

rango1 es un rango que contiene el primer (o el único) grupo de datos con el que se desea realizar la verificación.

media1 es la media conocida de la población a la que corresponde *rango1* y puede ser cualquier valor.

std1 es la desviación típica conocida de la población a la que corresponde *rango1*. Puede ser cualquier valor mayor que 0.

colas es un argumento opcional que permite especificar la dirección en que se desea realizar la verificación *z*.

cola	1-2-3 realiza
s	
1	Una verificación <i>z</i> de una cola
2	Una verificación <i>z</i> de dos colas; es el valor predeterminado si se omite este argumento

rango2 es un rango que contiene el segundo grupo de datos con los que se desea realizar la verificación.

media2 es la media conocida de la población a la que corresponde *rango2* y puede ser cualquier valor. Si se omite *media2*, 1-2-3 utilizará el valor 0.

std2 es la desviación típica conocida de la población a la que corresponde *rango2*. Puede ser cualquier valor mayor que 0. Si se omite *std2*, 1-2-3 utilizará el valor 1.

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Notas

@TESTZ calcula el valor aproximado de la probabilidad asociada a una verificación mediante *z* con un margen de error de $\pm 7.5 \cdot 10^{-8}$.

Ejemplos

El rango A1..A8 contiene los ocho valores siguientes: 12, 19, 21, 22, 18, 16, 15, 17. Si la media de los mismos es 16 y la desviación típica de la población es igual a 3,041381, entonces $z = 1,394972$.

@TESTZ(A1..A8;16;3,041381;1) = 0,081512

Funciones @ similares

[@TESTJI](#) realiza una verificación mediante el parámetro estadístico ji cuadrado, [@TESTFI](#) realiza una verificación mediante el parámetro estadístico *F*, y [@TESTTI](#) realiza una verificación mediante el parámetro estadístico *t*.

@CADENAFECHA

@CADENAFECHA(*fecha*) convierte un número de fecha en su fecha equivalente y lo muestra como un rótulo aplicándole el formato de fecha internacional predeterminado.

Argumentos

fecha es un número de fecha.

Notas

El sistema operativo ofrece la posibilidad de cambiar la configuración regional, lo cual permite a su vez cambiar el formato internacional predeterminado para la fecha.

Ejemplos

Si el formato de fecha predeterminado es dd/mm/aa, **@CADENAFECHA**(34635) dará como resultado el rótulo 28/10/94.

Funciones @ similares

@VALFECHA calcula el número de fecha de una fecha especificada en forma de rótulo. **@FECHA** calcula el número de fecha de fecha dada.

@DURACION, @DURACIONM

@DURACION(*negociación;vencimiento;cupón;rentabilidad;[frecuencia;base]*) calcula la duración anual de activos financieros que producen intereses periódicos.

@DURACIONM(*negociación;vencimiento;cupón;rentabilidad;[frecuencia;base]*) calcula la duración anual modificada de activos financieros que producen intereses periódicos.

Argumentos

negociación corresponde a la fecha en la que se negocia el activo y es un número de fecha.

vencimiento es la fecha de vencimiento o amortización del activo y es un número de fecha. Si el valor de *vencimiento* es igual o inferior al de *negociación*, el resultado de las funciones @DURACION y @DURACIONM será ERR.

cupón es el tipo de interés anual del cupón del activo. Puede ser 0 o cualquier número positivo.

rentabilidad es la rentabilidad anual. Puede ser cualquier valor positivo o 0.

frecuencia es un argumento opcional que permite especificar el número de pagos de cupón por año. Debe ser cualquiera de los valores indicados en la tabla siguiente:

<u><i>frecuencia</i></u>	<u>Frecuencia de los pagos de cupón</u>
1	Anual
2	Semestral; es el valor predeterminado si se omite este argumento
4	Trimestral
12	Mensual

base es un argumento opcional que permite especificar el tipo de base para el recuento de los días que va a utilizarse. Es un valor extraído de la tabla siguiente:

<u><i>base</i></u>	<u>Base para el recuento de los días</u>
0	30/360; es el valor predeterminado si se omite este argumento
1	Real/real
2	Real/360
3	Real/365
4	Europeo 30/360

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Notas

Duración es el período de vencimiento de la media ponderada para los flujos de caja de un activo. Las proporciones corresponden al valor actual de cada flujo de caja en forma de fracción del valor actual de todos los flujos de caja.

Ejemplos

La fecha de negociación de un activo es 1 de julio de 1993 y su fecha de vencimiento 1 de diciembre de 1998. El tipo que se aplica al cupón semestral es del 5,50% y la rentabilidad anual es 5,61%. La base para el recuento de los días es 30/360.

Para determinar la duración anual del activo:

@DURACION(@FECHA(93;7;1);@FECHA(98;12;1);0,055;0,0561;2;0) = 4,734591

Para determinar la duración anual modificada del activo:

@DURACIONM(@FECHA(93;7;1);@FECHA(98;12;1);0,055;0,0561;2;0) = 4,60541

Funciones @ similares

@ACUMULADO calcula el importe acumulado de los intereses para activos financieros que producen pagos periódicos de intereses. @PRECIO calcula el precio por cada 100\$ de valor nominal para activos financieros que proporcionan pagos periódicos de intereses. @RENTABILIDAD calcula la rentabilidad de activos financieros que proporcionan pagos periódicos de intereses.

@ESTAVACIA

@ESTAVACIA(x) verifica la *posición* de una celda vacía. Si x corresponde a una celda vacía, la función @ESTAVACIA da como resultado 1 (verdad); si x no es una celda vacía, @ESTAVACIA da como resultado 0 (falso).

Argumentos

x es el nombre o la dirección de una única celda. Si se especifica un rango para x, la función @ESTAVACIA dará como resultado 0 (falso).

Ejemplos

@ESTAVACIA(A1) = 1 si la celda A1 es una celda vacía

@ESTAVACIA(A1) = 0 si la celda A1 contiene el valor 1.963

@ESTAVACIA(A1) = 0 si la celda A1 contiene el rótulo Ingresos

@ESTAVACIA(A1) = 0 si la celda A1 contiene un carácter de prefijo de rótulo

@BUSCARMAXIMO, @BUSCARMINIMO

@BUSCARMAXIMO(*lista_rangos*) da como resultado una referencia absoluta a la celda que contiene el valor máximo de una lista de rangos; la referencia incluye el nombre de archivo del libro.

@BUSCARMINIMO(*lista_rangos*) da como resultado una referencia absoluta a la celda que contiene el valor mínimo de una lista de rangos; la referencia incluye el nombre de archivo del libro.

Argumentos

lista_rangos puede ser cualquier combinación de rangos. Utilice separadores de argumentos para separar las direcciones o nombres de rangos de *lista_rangos*.

Si desea incluir en *lista_rangos* un rango compuesto por una sola celda, deberá asegurarse de que su aspecto es lo más parecido posible a una dirección de rango. Por ejemplo, no indique A1, sino A1..A1.

1-2-3 ignora los rótulos y las celdas vacías incluidas en *lista_rangos*.

Si ninguna de las celdas de *lista_rangos* contiene valores, las funciones **@BUSCARMAXIMO** y **@BUSCARMINIMO** darán como resultado ND.

Ejemplos

Suponga que su directorio 1-2-3 contiene los archivos PID1.WK4, PID2.WK4 y PID3.WK4, cada uno de los cuales guarda información relacionada con presupuestos presentados por distintos vendedores. Los libros se han creado a partir de la misma plantilla por lo que, en cada uno de ellos, la cantidad presupuestada se encuentra en una celda llamada TOTAL.

El resultado de la fórmula siguiente indica la ubicación, incluido el nombre del archivo, del presupuesto más alto:

@BUSCARMAXIMO(<<PID1.WK4>>TOTAL;<<PID2.WK4>>TOTAL;<<PID3.WK4>>TOTAL)

El resultado de la fórmula siguiente indica la ubicación, incluido el nombre del archivo, del presupuesto más bajo:

@BUSCARMINIMO(<<PID1.WK4>>TOTAL;<<PID2.WK4>>TOTAL;<<PID3.WK4>>TOTAL)

Funciones @ similares

@CONSULH y **@CONSULV** buscan el contenido de una celda en una tabla de consulta horizontal o vertical, respectivamente. **@INDICE** busca el contenido de una celda basándose en los números offset especificados para la columna y la fila donde se encuentra. **@ELEGIR** busca el contenido de una celda en una tabla de consulta de una única fila. **@BUSCARCONTENIDO** busca la posición relativa de una celda cuando se conoce el contenido. **@INTERSECCION** busca en un rango el contenido de una celda especificada mediante el título de la columna, la fila y la hoja donde se encuentra.

@LABORABLESNETOS

@LABORABLESNETOS(*fecha_inicial*;*fecha_final*;*[rango_festivos*;*fin_de_semana*]) calcula el número de días que transcurren desde la *fecha_inicial* hasta la *fecha_final*, excluyendo los fines de semana y los festivos o de vacaciones.

Argumentos

fecha_inicial y *fecha_final* son números de fecha.

rango_festivos es un argumento opcional que especifica los días festivos o de vacaciones que no deben incluirse en el cálculo de la función **@LABORABLESNETOS**; es el nombre o la dirección del rango que contiene los números de fecha.

fin_de_semana es un argumento opcional que especifica los días que forman el fin de semana y, por lo tanto, no deben considerarse como laborables; es texto que se vale de los números enteros de 0 (lunes) a 6 (domingos) para indicar los días que corresponden al fin de semana.

Por ejemplo, para excluir los viernes y los sábados de una semana laborable use "45" como argumento *fin_de_semana*. Si omite el argumento *fin_de_semana*, 1-2-3 utilizará "56", lo que significa que considera que el fin de semana lo constituyen el sábado y el domingo. Si no desea especificar un valor para el fin de semana, indique 7.

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Notas

@LABORABLESNETOS incluye en el resultado tanto la *fecha_inicial* como la *fecha_final*.

Si desea utilizar el argumento *fin_de_semana* pero no *rango_festivos* deberá especificar una celda vacía para éste último.

Ejemplos

Suponga que desea determinar el número de días laborables comprendidos entre el martes 1 de noviembre de 1994 y el jueves 1 de diciembre de 1994, teniendo en cuenta que los días 24 y 25 de noviembre son festivos. En tal caso, deberá indicar dichas fechas en un rango llamado FESTIVOS. Para especificar los sábados y domingos como días pertenecientes al fin de semana, omita el argumento *fin_de_semana*.

@LABORABLESNETOS(@FECHA(94;11;1);@FECHA(94;12;1);FESTIVOS) = 21

Funciones @ similares

@DIAS360 y **@D360** calculan el número de días existentes entre dos números de fecha. **@DIAS** calcula el número de días existentes entre dos fechas y, para ello, se vale de una determinada base para el recuento de los días.

@LABORABLE calcula la fecha que corresponde a un número de días anterior o posterior a una fecha determinada sin contar los fines de semana y festivos. **@MESSIGUIENTE** calcula la fecha de un número de meses anterior o posterior a una fecha determinada.

@MESSIGUIENTE

@MESSIGUIENTE(*fecha_inicial*; *meses*; [*día_del_mes*]; [*base*]) calcula el número de fecha que corresponde a la fecha de un determinado número de *meses* anterior o posterior a la *fecha_inicial*.

Argumentos

fecha_inicial es un número de fecha.

meses es un entero. Utilice un número entero positivo para indicar el correspondiente a los meses posteriores a la *fecha_inicial*, o bien un entero negativo para los meses anteriores a la *fecha_inicial*.

día_del_mes es un argumento opcional que especifica el día del mes al que se desea que corresponda el resultado devuelto por la función @MESSIGUIENTE. Es un valor extraído de la tabla siguiente:

<u>día_del_mes</u>	<u>@MESSIGUIENTE da como resultado</u>
0	Una fecha que corresponde al mismo día de <i>fecha_inicial</i> . Si <i>fecha_inicial</i> corresponde a un día que no existe en el mes siguiente (por ejemplo, si la <i>fecha_inicial</i> es 30 de enero de 1994 y el mes siguiente, febrero, tiene 28 días); @MESSIGUIENTE dará como resultado una fecha que corresponda al último día del mes. Es el valor predeterminado cuando se omite este argumento.
1	Una fecha que corresponde al primer día del mes.
2	Una fecha que corresponde al último día del mes.

base es un argumento opcional que especifica el tipo de base para el recuento de los días que va a utilizarse. Es un valor extraído de la tabla siguiente:

<u>base</u>	<u>Base para el recuento de los días</u>
0	30/360
1	Real/real; es el valor predeterminado si se omite este argumento
2	Real/360
3	Real/365
4	Europeo 30/360

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Ejemplos

Suponga que desea determinar la fecha que corresponda al último día del mes siguiente al martes, 7 de abril de 1994.

@MESSIGUIENTE(@FECHA(94;4;7);1;2) = 34485 o martes, 31 de mayo de 1994

Funciones @ similares

@DIAS360 y @D360 calculan el número de días existentes entre dos números de fecha. @LABORABLE calcula la fecha correspondiente a un número especificado de días anterior o posterior a la fecha indicada, sin contar los fines de semana ni los festivos. @LABORABLESNETOS calcula el número de días existentes entre dos fechas, excluyendo los fines de semana y festivos.

@DEFINIRCADENA

@DEFINIRCADENA(*cadena*;*longitud*;*[alineación]*) da como resultado un rótulo de texto del tamaño indicado en *longitud*. El rótulo está formado por *cadena* y un número de espacios en blanco suficientes para poder alinear el texto de *cadena* según lo especificado en *alineación*.

Argumentos

cadena puede ser cualquier cadena de texto.

longitud puede ser cualquier número entero de 1 a 512. Si *longitud* es menor que el número de caracteres de *cadena*, @DEFINIRCADENA dará como resultado *cadena*.

alineación es un argumento opcional que sirve para especificar la forma en la que se debe alinear la *cadena*. Es un valor obtenido de la tabla siguiente:

<i>alineación</i>	1-2-3 alinea el texto de <i>cadena</i>
0	A la izquierda de los espacios sobrantes en <i>longitud</i> ; es el valor predeterminado cuando se omite este argumento.
1	En el centro de los espacios sobrantes en <i>longitud</i> . En el caso de que el número de espacios sobrantes sea impar, 1-2-3 agrega el espacio sobrante a la izquierda de la <i>cadena</i> .
2	A la derecha de los espacios sobrantes en <i>longitud</i> .

La mayoría de las fuentes utilizadas en Windows cuentan con espaciado proporcional. Por lo general, estas fuentes conceden más espacio a las letras que a los espacios en blanco.

Ejemplos

En los ejemplos siguientes, cada símbolo • representa un espacio en blanco.

@DEFINIRCADENA("Elemento nueve, C. de B.";30) = Elemento nueve, C. de B.●●●●●

@DEFINIRCADENA("Elemento nueve, C. de B.";30;1) = ●●●Elemento nueve, C. de B.●●●

@DEFINIRCADENA("Elemento nueve, C. de B.";30;2) = ●●●●●Elemento nueve, C. de B.

Funciones @ similares

@SUPRESP elimina de la cadena los espacios iniciales, finales y reduce a uno los espacios múltiples consecutivos.

@SUMANEGATIVOS, @SUMAPOSITIVOS

@SUMANEGATIVOS(*lista*) suma sólo los valores negativos de *lista*.

@SUMAPOSITIVOS(*lista*) suma sólo los valores positivos de *lista*.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas, direcciones o nombres de rango que incluyan números o fórmulas numéricas. Utilice separadores de argumentos para separar los elementos de *lista*.

Ejemplos

@SUMANEGATIVOS(-2;21;5;12;-2;-7) = -11

@SUMAPOSITIVOS(-2;21;5;12;-2;-7) = 38

Funciones @ similares

@SUMA calcula la suma de los valores de una lista. **@BSUMA** calcula la suma de los valores contenidos en un campo de una tabla de base de datos que cumplan unos criterios determinados. **@SUBTOTAL** calcula los valores de una lista e indica a **@TOTAL** las celdas que debe sumar.

@VALFUTCANTIDAD

@VALFUTCANTIDAD (*principal;interés;plazo;[frecuencia]*) da como resultado el valor futuro de una suma global invertida a un tipo de interés determinado durante un número dado de períodos.

Argumentos

principal representa la cantidad invertida inicialmente y puede ser cualquier valor.

interés representa el tipo de interés periódico y puede ser cualquier valor decimal o porcentual.

plazo representa el número de períodos para los que se realiza la inversión y puede ser cualquier valor.

frecuencia es un argumento opcional y representa la frecuencia de capitalización para cada período. Si se especifica un valor que no sea entero, 1-2-3 truncará su parte decimal. El valor predeterminado para frecuencia, si se omite este argumento, es 1.

No podrá utilizarse un argumento opcional si se omiten los que le preceden.

Notas

Si *plazo* es 1 y *frecuencia* es 1, **@VALFUTCANTIDAD** calculará un tipo de interés simple.

Ejemplo

Un padre deposita 100\$ en una cuenta de ahorro que proporciona un 10% de interés, para que el hijo, que ese día cumple 10 años, lo reciba cuando cumpla 18 años. Mediante **@VALFUTCANTIDAD**, podrá calcular la cantidad que habrá acumulado hasta ese momento la cuenta:

@VALFUTCANTIDAD(100;10%;8) = 214,36\$

Funciones @ similares

@VALFUTCANTIDAD se refiere a un pago único. **@VALFUT** y **@VALFUTUROL** dan como resultado el valor futuro para una anualidad, es decir, una serie de uno o más pagos. **@VALACTGLOBAL** da como resultado el valor actual de una suma global que se percibirá dentro de un determinado número de períodos en el futuro y que se descontará o actualizará con un tipo de interés dado.

@VALACTGLOBAL

@VALACTGLOBAL(*valor_futuro;interés;plazo;[frecuencia]*) da como resultado el valor actual de una suma global que se percibirá dentro de un determinado número de períodos en el futuro y que se descontará o actualizará con un tipo de interés dado.

Argumentos

valor_futuro representa la cantidad que se percibirá en el futuro y puede ser cualquier valor.

interés representa el tipo de descuento periódico y puede ser cualquier valor decimal o porcentual.

plazo representa el número de períodos que tardará en percibirse el principal en el futuro y puede ser cualquier valor.

frecuencia es un argumento opcional y representa la frecuencia de capitalización por cada período de pago. En caso de que se especifique un valor que no sea un número entero, 1-2-3 lo truncará. El valor predeterminado si se omite este argumento es 1.

Notas

Si *plazo* es 1 y *frecuencia* también es 1, **@VALACTGLOBAL** realizará el calculo para un tipo de interés simple.

Ejemplos

Suponga que necesita 10.000\$ para los gastos de universidad, a pagar dentro de 10 años, y que puede recibir un 10% de interés sobre esa cantidad. Podrá utilizar **@VALACTGLOBAL** para determinar la suma global que deberá invertir en la actualidad:

@VALACTGLOBAL(10.000,10%,10) = 3.855,43\$

Funciones @ similares

@VALACT y **@VALACTUAL** dan como resultado el valor actual de una anualidad, es decir, una serie de uno o más pagos. **@VALFUTCANTIDAD** da como resultado el valor futuro de una suma global invertida a un tipo de interés determinado durante un número dado de períodos.

@TIRX

@TIRX(*estimación;flujos_caja;fechas*) da como resultado la tasa interna de rendimiento prevista para una serie de flujos de caja de entrada y de salida.

Argumentos

estimación puede ser un valor decimal o porcentual que represente el cálculo estimado de la tasa interna de rendimiento. En la mayoría de los casos, *estimación* debería ser un porcentaje entre 0 (0%) y 1 (100%). Generalmente, se considera que 10% es una buena estimación. Cuando se trate de flujos de caja muy elevados, asegúrese de que *estimación* sea un valor lo más preciso posible.

flujos_caja es la dirección o el nombre de un rango que contiene los flujos de caja. 1-2-3 considerará los números negativos como flujos de salida o pagos, y los números positivos como flujos de entrada o cobros. Normalmente, la primera cantidad o flujo de caja del rango es un número negativo (un flujo de salida o pago), que representa una inversión. 1-2-3 asignará el valor 0 a todas las celdas vacías y rótulos del rango y los incluirá en los cálculos.

fechas es la dirección o el nombre de un rango que contiene las fechas en las que tienen lugar los correspondientes flujos de caja. Cada fecha debe ser un número de fecha y corresponde a la fecha en que tuvo lugar el correspondiente flujo de del rango *flujos_caja*. Las fechas deben estar en orden ascendente.

Los rangos a los que hacen referencia los argumentos *flujos_caja* y *fechas* deberán tener las mismas dimensiones, es decir, deberán contener el mismo número de celdas. 1-2-3 emparejará las celdas de ambos rangos de acuerdo con la posición que ocupen en el rango. Los rangos se consideran ordenados de arriba a abajo (por filas), de izquierda a derecha (por columnas) y por hojas, de la primera a la última. Si los rangos correspondientes a *flujos_caja* y *fechas* no tienen las mismas dimensiones, @TIRX dará como resultado ERR.

Notas

Los flujos de caja pueden haberse producido a intervalos desiguales. Cada flujo de caja de *flujos_caja* está emparejado con una fecha de *fechas*. El primer flujo de caja y la primera fecha indican el inicio del proceso; el primer flujo de caja no estará actualizado. Los siguientes flujos se actualizarán según la tasa de descuento anual y la fecha en que se originó el flujo.

@TIR actualizará o descontará el primer flujo de caja, mientras que @TIRX no lo hará y utilizará la primera fecha como la de inicio del proceso. Utilice el valor 0 para el primer flujo de caja en @TIRX para simular el mismo método de actualización o descuento que aplica @TIR.

@TIRX permite que los flujos se produzcan a intervalos desiguales; @TIR supone que se originan a intervalos iguales.

Ejemplos

Suponga que realiza una inversión de 50.000\$ el día 13 de septiembre de 1996 y recibe los siguientes pagos: 1.000\$ el 31 de enero de 1997 y 53.000\$ el 14 de junio de 1997. Estos datos se muestran en el rango siguiente:

ESTIMACION	FLUJOS	FECHAS
0,10	-50000	13/09/96
	1000	31/01/97
	53000	14/06/97

@TIRX(ESTIMACION;FLUJOS;FECHAS) = 10,90%

Funciones @ similares

@VANX da como resultado el valor actual neto de una serie de flujos de caja de entrada y de salida. @TIR calcula la tasa interna de rendimiento (beneficio) para una serie de valores de flujos de caja generados por una inversión.

@TIRM calcula la tasa interna de rendimiento modificada.

@VANX

@VANX(*tasa;flujos_caja;fechas*) da como resultado el valor actual neto de una serie de flujos de caja de entrada (cobros) y de salida (pagos).

Argumentos

tasa puede ser cualquier valor mayor que -1 y representa la tasa de descuento o actualización.

flujos_caja es la dirección o el nombre de un rango que contiene los flujos de caja. 1-2-3 considerará los números negativos como flujos de salida y los números positivos como flujos de entrada. Normalmente, la cantidad que indica el primer flujo de caja en el rango es un número negativo (un pago) que representa la inversión. 1-2-3 asignará el valor 0 a todas las celdas vacías y rótulos que existan en el rango y los incluirá en los cálculos.

fechas es la dirección o el nombre de un rango que contiene las fechas en las que tienen lugar los correspondientes flujos de caja. Cada fecha debe ser un número de fecha y corresponde a la fecha en que tuvo lugar el correspondiente flujo en del rango *flujos_caja*. Las fechas deben estar en orden ascendente.

Los rangos a los que hacen referencia los argumentos *flujos_caja* y *fechas* deberán tener las mismas dimensiones, es decir, deberán contener el mismo número de celdas. 1-2-3 emparejará las celdas de ambos rangos de acuerdo con la posición que ocupen en el rango. Los rangos se consideran ordenados de arriba abajo (por filas), de izquierda a derecha (por columnas) y por hojas, de la primera a la última. Si los rangos correspondientes a *flujos_caja* y *fechas* no tienen las mismas dimensiones, @VANX dará como resultado ERR.

Notas

Los flujos de caja pueden haberse producido a intervalos desiguales. Cada flujo de caja de *flujos_caja* está emparejado con una fecha de *fechas*. El primer flujo de caja y la primera fecha indican el inicio del proceso; el primer flujo de caja no estará actualizado. Los siguientes flujos se actualizarán según la *tasa* de descuento anual y la fecha en que se originó el flujo.

Ejemplos

En el ejemplo siguiente se utiliza @VANX para actualizar con el valor actual de los dólares, a 1 de enero de 1996, una serie de distribuciones irregulares invertidas con una tasa de descuento o porcentaje anual del 11,5%. En este ejemplo se utilizarán los datos que aparecen a continuación, donde *flujos_caja* será una lista de flujos contenida en un rango denominado DISTRIBUCIONES, expresado en dólares, y *fechas* será la lista de las fechas en que se producen los flujos de caja, contenida en un rango denominado FECHAS:

DISTRIBUCIONES	FECHAS
0	01/01/96
250.000	01/04/96
250.000	01/05/96
300.000	01/06/96
500.000	01/07/96
600.000	01/08/96
900.000	01/09/96
300.000	01/10/96
250.000	01/11/96
750.000	01/01/97

@VANX(0,115;DISTRIBUCIONES;FECHAS) = 3.821.809,20\$

Funciones @ similares

@VAN calcula el valor actual neto de una inversión, descontando el valor futuro al valor actual. @TIRX da como resultado la tasa interna de rendimiento para una serie de flujos de caja de entrada y de salida.

@ESTACOMPREDIDO

@ESTACOMPREDIDO(*valor*; *límite1*; *límite2*; [*tipo*]) comprueba si *valor* está comprendido entre *límite1* y *límite2*.

@ESTACOMPREDIDO dará como resultado el valor 1 (verdad) si *valor* está comprendido entre *límite1* y *límite2*, y 0 (falso) en caso contrario.

Argumentos

valor, *límite1* y *límite2* pueden ser cualquier valor numérico o cadenas de texto.

tipo es un argumento opcional y puede ser cualquiera de los valores que aparecen en la tabla siguiente:

<u>tipo</u>	<u>Descripción</u>	<u>Algoritmo</u>
0	<i>límite1</i> y <i>límite2</i> están incluidos (es el valor predeterminado si se omite este argumento)	@MIN(<i>límite1</i> , <i>límite2</i>) <= <i>valor</i> <= @MAX(<i>límite1</i> , <i>límite2</i>)
1	Incluye el menor de <i>límite1</i> y <i>límite2</i> y excluye el mayor	@MIN(<i>límite1</i> , <i>límite2</i>) <= <i>valor</i> < @MAX(<i>límite1</i> , <i>límite2</i>)
2	Excluye el menor de <i>límite1</i> y <i>límite2</i> e incluye el mayor	@MIN(<i>límite1</i> , <i>límite2</i>) < <i>valor</i> <= @MAX(<i>límite1</i> , <i>límite2</i>)
3	<i>límite1</i> y <i>límite2</i> están ambos excluidos	@MIN(<i>límite1</i> , <i>límite2</i>) < <i>valor</i> < @MAX(<i>límite1</i> , <i>límite2</i>)

Notas

Si el tipo de datos de *límite1* es diferente del tipo de datos de *límite2*, @ESTACOMPREDIDO dará como resultado ERR.

Si el tipo de datos de *límite1* o de *límite2* es diferente del de *valor*, @ESTACOMPREDIDO dará como resultado el valor 0.

No es posible utilizar un argumento opcional si se omiten los que le preceden.

Ejemplos

@ESTACOMPREDIDO(123;100;200) = 1

@ESTACOMPREDIDO("ABC";"AAA";"BBB") = 1

@ESTACOMPREDIDO(100;100;200;0) = 1

@DECIL

@DECIL(*x*; *rango*) da como resultado un decil determinado.

Argumentos

x es un entero entre 0 y 10, ambos inclusive.

x	Da como resultado	Equivale a...
0	El mínimo	@PERCENTILP(0,0; <i>rango</i>) o @MIN(<i>rango</i>)
1	El primer decil	@PERCENTILP(0,1; <i>rango</i>)
2	El segundo decil	@PERCENTILP(0,2; <i>rango</i>)
3	El tercer decil	@PERCENTILP(0,3; <i>rango</i>)
4	El cuarto decil	@PERCENTILP(0,4; <i>rango</i>)
5	La mediana	@PERCENTILP(0,5; <i>rango</i>) o @MEDIANA(<i>rango</i>)
6	El sexto decil	@PERCENTILP(0,6; <i>rango</i>)
7	El séptimo decil	@PERCENTILP(0,7; <i>rango</i>)
8	El octavo decil	@PERCENTILP(0,8; <i>rango</i>)
9	El noveno decil	@PERCENTILP(0,9; <i>rango</i>)
10	El máximo	@PERCENTILP(1,0; <i>rango</i>) o @MAX(<i>rango</i>)

rango es la dirección o el nombre del rango que contiene los valores.

Notas

@DECIL realiza el cálculo considerando las celdas vacías y las celdas que contienen rótulos como 0.

Ejemplos

Un rango denominado DATOS contiene los siguientes elementos:

34
12
60
128
67
350
206

@DECIL(0;DATOS) = 12,0

@DECIL(1;DATOS) = 25,2

@DECIL(2;DATOS) = 39,2

Funciones @ similares

@PERCENTILP calcula el valor correspondiente al percentil *x* para la muestra contenida en *rango*. @CUARTIL da como resultado un cuartil determinado. @MIN busca el menor valor de una lista. @MAX busca el mayor valor de una lista. @MEDIANA da como resultado la mediana de una lista.

@ESTIMACION

@ESTIMACION(*x*; *rango_y*; *rango_x*) da como resultado un valor estimado para *x* a partir de la tendencia lineal existente entre los valores del *rango_y* y los del *rango_x*.

Argumentos

x puede ser cualquier valor y representa el valor que se desea estimar.

rango_y es el nombre o la dirección de un rango y representa el conjunto de valores correspondientes a la variable dependiente. Si *rango_y* contiene algún dato no numérico, @ESTIMACION dará como resultado ERR.

rango_x es la dirección o el nombre de un rango y representa el conjunto de valores correspondientes a la variable independiente. Si *rango_x* contiene algún dato no numérico, @ESTIMACION dará como resultado ERR.

rango_y y *rango_x* deben tener el mismo tamaño, es decir, deben contener el mismo número de celdas. 1-2-3 emparejará las celdas de ambos rangos de acuerdo con la posición que ocupen en el rango. Los rangos se consideran ordenados de arriba abajo (por filas), de izquierda a derecha (por columnas) y por hojas, de la primera a la última. Si *rango_y* y *rango_x* no tienen el mismo tamaño, @ESTIMACION dará como resultado ERR.

Notas

La función @REGRESION también puede realizar este tipo de cálculos.

Ejemplos

Se desea calcular la función @ESTIMACION para los siguientes datos:

RANGOY	RANGOX
250	3
545	5
550	5
450	6
605	6
615	7

@ESTIMACION(-1;RANGOY;RANGOX) = -26,786

@ESTIMACION(0;RANGOY;RANGOX) = 56,786

Funciones @ similares

@REGRESION realiza un análisis de regresión lineal múltiple y da como resultado el parámetro de la regresión especificado.

@SUMAN

@SUMAN(*número_offset*; *n*; *lista*) suma todos los valores *enésimos* de *lista*, a partir de *número_offset*.

@SUMAN ignorará los valores ERR y ND de las celdas o valores omitidos.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas numéricas. Utilice separadores de argumentos para separar los elementos de *lista*.

número_offset puede ser cualquier entero positivo.

n puede ser cualquier entero positivo.

Notas

@SUMAN da como resultado la suma de (*número_offset*), (*número_offset* + *n*), (*número_offset* + 2*n*), (*número_offset* + 3*n*) ... de *lista*.

@SUMAN(0;1;*lista*) dará el mismo resultado que @SUMA(*lista*).

Ejemplos

@SUMAN(1;3;B5..B15) dará como resultado la suma de los valores contenidos en las celdas B6, B9, B12 y B15.

Funciones @ similares

@SUMA suma los valores de *lista*. @BSUMA calcula la suma de los valores que reúnan unos criterios determinados. @SUBTOTAL suma los valores contenidos en una lista e indica a @TOTAL los valores que debe sumar.

@MEDIANAPURA

@MEDIANAPURA(*lista*) da como resultado la mediana de los valores de *lista*, ignorando las celdas vacías, los rótulos y las fórmulas cuyos resultados sean rótulos.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas numéricas. Utilice separadores de argumentos para separar los elementos de *lista*.

Notas

La mediana es el valor central de *lista* y al obtenerlo se cumple que el número de valores superiores que la mediana es igual que el número de valores inferiores. Si *lista* contiene un número impar de valores, **@MEDIANAPURA** ordenará los valores y dará como resultado el valor central. Si *lista* contiene un número par de valores, **@MEDIANAPURA** ordenará los valores y dará como resultado la media aritmética de los dos valores centrales.

@MEDIANAPURA dará como resultado **ERR** en el caso de que el rango esté vacío o si el rango contiene únicamente rótulos o fórmulas cuyos resultados sean rótulos.

Ejemplos

@MEDIANAPURA(A2..A6) = 12, si las celdas A2..A6 contienen los valores 5, 12, 65, 82 y 9.

@MEDIANAPURA(A1..A6) = 12, si las celdas A1..A6 contienen los valores 5, 12, 65, 82 y 9, y el rótulo "Enero" (**@MEDIANAPURA** ignorará el rótulo). La función **@MEDIANA** asignaría al rótulo el valor 0 para realizar el cálculo, de modo que el resultado de la operación para esos datos sería 10,5.

Funciones @ similares

@MEDIANA da como resultado la mediana de una lista.

@CUARTIL

@CUARTIL(*x*;*rango*) da como resultado un cuartil determinado.

Argumentos

x es un entero entre 0 y 4, ambos inclusive, como se especifica en la tabla siguiente:

<u>x</u>	<u>Da como resultado</u>	<u>Equivale a...</u>
0	El mínimo	@PERCENTILP(0,0; <i>rango</i>) o @MIN(<i>rango</i>)
1	El primer cuartil	@PERCENTILP(0,25; <i>rango</i>)
2	La mediana	@PERCENTILP(0,50; <i>rango</i>) o @MEDIANA(<i>rango</i>)
3	El tercer cuartil	@PERCENTILP(0,75; <i>rango</i>)
4	El máximo	@PERCENTILP(1,00; <i>rango</i>) o @MAX(<i>rango</i>)

rango es la dirección o el nombre del rango que contiene los valores.

Notas

@CUARTIL realiza el cálculo considerando las celdas vacías y las celdas que contienen rótulos como 0.

Ejemplos

DATOS es un rango con el siguiente contenido:

34
12
60
128
67
350
206

@CUARTIL(0;DATOS) = 12

@CUARTIL(1;DATOS) = 47

@CUARTIL(2;DATOS) = 67

@CUARTIL(3;DATOS) = 167

@CUARTIL(4;DATOS) = 350

Funciones @ similares

@PERCENTILP calcula el valor correspondiente al percentil *x* para la muestra contenida en *rango*. @DECIL da como resultado un decil determinado.

@BUSCARB

@BUSCARB(*cadena_búsqueda;cadena;posición_inicial*) calcula la posición (en bytes) que ocupa *cadena_búsqueda* la primera vez que aparece dentro de *cadena*. 1-2-3 comenzará la búsqueda a partir de la posición (en bytes) especificada por *posición_inicial*.

Argumentos

cadena_búsqueda y *cadena* son cadenas de texto encerradas entre comillas (" "), fórmulas cuyo resultado sea texto o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

posición_inicial es un número offset en bytes.

Notas

En el caso de que 1-2-3 no encuentre *cadena_búsqueda* en *cadena*, **@BUSCARB** dará como resultado ERR. Asimismo, **@BUSCARB** dará como resultado ERR si el valor de *posición_inicial* es superior al número de bytes de *cadena*, o si se trata de un valor negativo.

@BUSCARB tiene en cuenta si los caracteres están escritos en mayúsculas o en minúsculas y si están o no acentuados; por ejemplo, **@BUSCARB** no encontrará la *cadena_búsqueda* "e" en la *cadena* "TOLEDO".

@BUSCARB también puede utilizarse junto con **@MEDB** o **@SUSTITUIRB** para hallar y extraer o reemplazar cadenas de texto.

Ejemplos

@BUSCARB("v";"Departamento de ventas";0) = 16 porque *cadena_búsqueda*, es decir, la letra v, ocupa la posición 16 (en bytes) en la *cadena* Departamento de ventas.

Funciones @ similares

@BUSCAR da como resultado la posición en caracteres.

@MEDB

@MEDB(cadena;posición_inicio;n) copia *n* bytes de *cadena*, empezando a partir del dato situado en el byte *posición_inicio*.

Argumentos

cadena puede ser una cadena de texto encerrado entre comillas (" "), una fórmula cuyo resultado sea una cadena de texto o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

posición_inicio es un número offset en bytes. Si *posición_inicio* es un valor mayor que la longitud de *cadena*, el resultado de **@MEDB** será una cadena vacía.

n puede ser cualquier entero positivo o el valor 0. Si *n* es 0, el resultado de **@MEDB** será una cadena vacía. Si *n* es mayor que la longitud de *cadena*, 1-2-3 dará como resultado todos los bytes desde *posición_inicio* hasta el final de *cadena*.

Notas

@MEDB puede utilizarse con conjuntos de caracteres de doble byte (DBCS) como es el caso de los caracteres japoneses.

@MEDB cuenta los signos de puntuación y los espacios como caracteres.

Si se desconoce la longitud de la *cadena*, conviene utilizar un número elevado para *n*; 1-2-3 ignorará los espacios sobrantes y dará como resultado la *cadena* completa a partir de *posición_inicio*.

Si el primer byte del rótulo que se obtiene como resultado es la segunda mitad de un carácter de doble byte o si el último byte de dicho rótulo es la primera mitad de un carácter de doble byte, **@MEDB** reemplazará el carácter incompleto por un espacio.

Ejemplos

@MEDB("Cadena de un solo byte";7;2) = "de"

Funciones @ similares

@MED copia *n* caracteres de la *cadena*, a partir del carácter situado en *posición_inicio*. **@IZDAB** da como resultado los *n* primeros bytes de *cadena*. **@DERECHAB** da como resultado los *n* últimos bytes de *cadena*.

@IZDAB

@IZDAB(*cadena*; *n*) arroja como resultado los *n* primeros bytes de *cadena*.

Argumentos

cadena puede ser una cadena de texto encerrado entre comillas (" "), una fórmula cuyo resultado sea cadena de texto o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

n puede ser un entero positivo o el valor 0. Si *n* es 0, el resultado será una cadena vacía. Si *n* es mayor que la longitud de *cadena* (en bytes), @IZDAB dará como resultado la *cadena* completa.

Notas

@IZDAB cuenta los signos de puntuación y los espacios como caracteres.

@IZDAB se puede utilizar con juegos de caracteres de doble byte (DBCS), como es el caso de los caracteres japoneses. Por ejemplo, si se desea insertar un rótulo en un campo de una base de datos externa mediante 1-2-3 y el campo tiene una longitud limitada en bytes, podrán extraerse partes de dicho rótulo para ajustarlo a los límites impuestos.

Si el último byte del rótulo que se obtiene como resultado es la primera parte de un carácter de doble byte, @IZDAB reemplazará el carácter incompleto por un espacio.

Si *cadena* está formada únicamente por caracteres de un solo byte, @IZDAB dará como resultado el mismo que daría @IZDA.

Ejemplos

@IZDAB("Un solo byte";9) = "Un solo b"

Funciones @ similares

@IZDA da como resultado los *n* primeros caracteres de *cadena*. @MEDB da como resultado *n* bytes del interior de *cadena*. @DERECHAB da como resultado los *n* últimos bytes de *cadena*.

@DERECHAB

@DERECHAB(*cadena*; *n*) arroja como resultado los últimos *n* bytes de *cadena*.

Argumentos

cadena puede ser una cadena de texto encerrada entre comillas (" "), una fórmula cuyo resultado sea una cadena de texto o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

n puede ser cualquier entero positivo o 0. Si *n* es 0, el resultado será una cadena vacía. Si *n* es mayor que la longitud de *cadena*, @DERECHAB dará como resultado la *cadena* completa.

Notas

@DERECHA puede utilizarse para juegos de caracteres de doble byte (DBCS) como es el caso de los caracteres japoneses.

@RIGHTB cuenta los signos de puntuación y los espacios como caracteres.

Si el primer byte del rótulo que se obtiene como resultado es la segunda mitad de un carácter de doble byte,

@DERECHAB reemplazará el carácter incompleto por un espacio.

Utilice @DERECHAB junto con @BUSCARB cuando desconozca el valor exacto de *n* o cuando *n* pueda variar.

Ejemplos

@DERECHAB("Un solo byte";4) = "byte"

Funciones @ similares

@DERECHA da como resultado los *n* últimos caracteres de una cadena. @IZDAB da como resultado los *n* primeros bytes de una cadena. @MEDB da como resultado unos determinados bytes del interior de una cadena.

@LONGITUDB

@LONGITUDB(*cadena*) cuenta el número de bytes de *cadena*.

Argumentos

cadena puede ser una cadena de texto encerrado entre comillas (" "), una fórmula cuyo resultado sea una cadena de texto o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

Notas

@LONGITUDB cuenta los signos de puntuación y los espacios como caracteres.

@LONGITUDB se puede utilizar con conjuntos de caracteres de doble bytes (DBCS) como es el caso de los japoneses. Por ejemplo, si se desea introducir una cadena de texto en un campo de una base de datos externa mediante 1-2-3 y el campo tiene una limitación de longitud en bytes, es posible comprobar si la cadena excede o no dicha limitación.

Ejemplos

@LONGITUDB (A5&G12) = el número total de bytes de las celdas A5 y G12.

Funciones @ similares

@LONGITUD cuenta los caracteres que contiene *cadena*.

@SUSTITUIRB

@SUSTITUIRB(*cadena_original*; *posición_inicio*; *n*; *cadena_nueva*) sustituye *n* bytes de *cadena_original* por *cadena_nueva*, a partir de *posición_inicio*.

Argumentos

cadena_original y *cadena_nueva* pueden ser cadenas de texto encerrado entre comillas (" "), fórmulas cuyo resultado sea una cadena de texto, o direcciones o nombres de celdas que contengan rótulos o fórmulas cuyos resultados sean rótulos.

posición_inicio es el número offset de un carácter de *cadena_original*. Puede ser cualquier valor positivo o 0. Si *posición_inicio* es mayor que la longitud en bytes de *cadena_original*, **@SUSTITUIRB** agregará *cadena_nueva* al final de *cadena_original*.

n puede ser cualquier entero positivo o 0. Si *n* es 0, **@SUSTITUIRB** insertará la *cadena_nueva* en *posición_inicio* sin eliminar ningún carácter de *cadena_original*.

Notas

@SUSTITUIRB calcula las posiciones y la longitud de la cadena en bytes.

@SUSTITUIRB cuenta los espacios y los signos de puntuación como caracteres. Si utiliza **@SUSTITUIR** para agregar o insertar texto, no olvide incluir los espacios necesarios.

Si los datos reemplazados en *cadena_original* comienzan o terminan en la parte central de un carácter de doble byte, **@SUSTITUIRB** reemplazará cada carácter de doble byte incompleto del texto obtenido como resultado por un espacio.

Utilice **@BUSCARB** junto con **@SUSTITUIRB** para buscar y sustituir un rótulo o para calcular una *posición_inicio* desconocida.

@SUSTITUIRB resultará útil cuando se necesite reemplazar un grupo de caracteres por otro, como, por ejemplo, para modificar el prefijo de los número de teléfono de una base de datos.

Ejemplos

@SUSTITUIRB(CELDA;**@BUSCARB**("-";CELDA;0),1,"/") cambia el rótulo contenido en CELDA, 4-24, por 4/24.

Funciones @ similares

@SUSTITUIR reemplaza *n* caracteres de *cadena_original* por *cadena_nueva*, a partir de *posición_inicio*.

@CONVERTFECHA

@CONVERTFECHA(*fecha;tipo_entrada;tipo_salida*) convierte una fecha hijri (árabe), farsi (iraní) o hebrea (judía) en la fecha gregoriana equivalente y viceversa.

Notas

@CONVERTFECHA ha sido creada específicamente para este tipo de lenguajes y sólo funcionará en los PC que admitan versiones bidireccionales de 1-2-3 en estos idiomas como, por ejemplo, en hebreo o en farsi.

@TBUSCAR

@TBUSCAR(*cadena_búsqueda*; *cadena*; *posición_inicial*) calcula la posición de un carácter tailandés lógico en *cadena*; 1-2-3 determinará cuándo aparece por primera vez *cadena_búsqueda* en *cadena* a partir del carácter tailandés que se encuentre en la columna a la que se refiere *posición_inicial*. Esta función operará únicamente sobre caracteres tailandeses lógicos, que pueden constar de 1, 2 o 3 bytes.

Argumentos

cadena_búsqueda y *cadena* pueden ser cadenas de texto encerradas entre comillas (" "), fórmulas cuyos resultados sean cadenas de texto o direcciones o nombres de celdas que contengan rótulos o fórmulas cuyos resultados sean rótulos.

posición_inicial es un número offset.

Notas

@TBUSCAR también resulta de utilidad cuando se utiliza junto con @TMED o @TSUSTITUIR para hallar y extraer o reemplazar una cadena de texto.

Ejemplos

@TBUSCAR("ไทย", "บริษัท ไทยซอฟต์แวร์ จำกัด", 0) = 5

Funciones @ similares

@BUSCAR realiza la misma función que @TBUSCAR, pero se utiliza con cadenas de texto que no son tailandesas.

@TMED

@TMED(*cadena*; *posición_inicio*; *n*) copia *n* caracteres tailandeses lógicos de *cadena* a partir del carácter tailandés que se encuentre en *posición_inicio*. Esta función operará únicamente sobre caracteres tailandeses lógicos, que pueden constar de 1, 2 o 3 bytes.

Argumentos

cadena puede ser una cadena de texto encerrada entre comillas (" "), una fórmula cuyo resultado sea una cadena de texto o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

posición_inicio es un número offset. Si el valor de *posición_inicio* es mayor que la longitud de *cadena*, el resultado de @TMED será una cadena vacía.

n puede ser cualquier entero positivo o 0. Si *n* es 0, el resultado de @TMED será una cadena vacía. Si *n* es mayor que la longitud de *cadena*, @TMED dará como resultado todos los caracteres que encuentre entre *posición_inicio* y el final de *cadena*.

Ejemplos

@TMED("บริษัท ๑๒๓ จำกัด",5,7) = ๑๒๓

Funciones @ similares

@MED realiza la misma función que @TMED pero se utiliza con cadenas de texto que no sean tailandesas.

@IDERECHA da como resultado los *n* últimos caracteres tailandeses de *cadena*. @TIZDA da como resultado los *n* primeros caracteres tailandeses de *cadena*.

@TIZDA

@TIZDA(*cadena*; *n*) arroja como resultado los *n* primeros caracteres tailandeses lógicos de *cadena*. Esta función operará únicamente sobre caracteres tailandeses lógicos, que pueden constar de 1, 2 o 3 bytes.

Argumentos

cadena puede ser una cadena de texto encerrada entre comillas (" "), una fórmula cuyo resultado sea una cadena de texto o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

n puede ser cualquier entero positivo o 0. Si *n* es 0, el resultado será una celda vacía. Si *n* es mayor que la longitud de *cadena*, @TIZDA dará como resultado la *cadena* completa.

Ejemplos

@TIZDA("บริษัท วิทยุสื่อสาร จำกัด",4) = บริษัท

Funciones @ similares

@IZDA realiza la misma función que @TIZDA pero se utiliza con cadenas de texto que no son tailandesas. @TMED da como resultado caracteres tailandeses del interior de *cadena*. @TDERECHA da como resultado los *n* últimos caracteres de *cadena*.

@TDERECHA

@TDERECHA(*cadena*; *n*) arroja como resultado los últimos *n* caracteres tailandeses lógicos de *cadena*. Esta función operará únicamente sobre caracteres tailandeses lógicos, que pueden constar de 1, 2 o 3 bytes.

Argumentos

cadena puede ser una cadena de texto encerrada entre comillas (" "), una fórmula cuyo resultado sea una cadena de texto o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

n puede ser un entero positivo o 0. Si *n* es 0, el resultado será una cadena vacía. Si *n* es mayor que la longitud de *cadena*, @TDERECHA dará como resultado la *cadena* completa.

Ejemplos

@TDERECHA("บริษัท วิทยงอฝท จ่ากัถ",4) = จ่ากัถ

Funciones @ similares

@DERECHA realiza la misma función que @TDERECHA pero se utiliza con cadenas de texto que no son tailandesas. @TMED da como resultado caracteres tailandeses del interior de *cadena*. @TIZDA da como resultado los *n* primeros caracteres tailandeses lógicos de *cadena*.

@TLONGITUD

@TLONGITUD(*cadena*) cuenta el número de caracteres tailandeses lógicos contenidos en *cadena*. Esta función operará únicamente sobre caracteres tailandeses lógicos, que pueden constar de 1, 2 o 3 bytes.

Argumentos

cadena puede ser una cadena de texto encerrada entre comillas (" "), una fórmula cuyo resultado sea una cadena de texto o la dirección o el nombre de una celda que contenga un rótulo o una fórmula cuyo resultado sea un rótulo.

Ejemplos

@TLONGITUD("บริษัท ไทยซอฟต์แวร์ จำกัด") = 17

Funciones @ similares

@LONGITUD realiza la misma función que @TLONGITUD pero se utiliza con cadenas de texto que no son tailandesas.

@TSUSTITUIR

@TSUSTITUIR(*cadena_original*; *posición_inicio*; *n*; *cadena_nueva*) sustituye *n* caracteres tailandeses lógicos de *cadena_original* por el contenido de *cadena_nueva*, a partir del carácter indicado por *posición_inicio*. Esta función operará únicamente sobre caracteres tailandeses lógicos, que pueden constar de 1, 2 o 3 bytes.

Argumentos

cadena_original y *cadena_nueva* pueden ser cadenas de texto encerradas entre comillas (" "), fórmulas cuyos resultados sean cadenas de texto o direcciones o nombres de celdas que contengan rótulos o fórmulas cuyos resultados sean rótulos.

posición_inicio representa el número offset de una columna de *cadena_original*. Puede ser cualquier valor positivo o 0. Si *posición_inicio* es mayor que la longitud de *cadena_original*, @TSUSTITUIR agregará *cadena_nueva* al final de *cadena_original*.

n puede ser cualquier entero positivo o 0. Si *n* es 0, @TSUSTITUIR insertará *cadena_nueva* a partir de *posición_inicio* sin eliminar ninguna columna de *cadena_original*.

Ejemplos

@TSUSTITUIR("บริษัท จำกัด",5,0,"ไทยซอฟท์") = บริษัท ไทยซอฟท์ จำกัด

Funciones @ similares

@SUSTITUIR realiza la misma función que @TSUSTITUIR pero se utiliza con cadenas de texto que no son tailandesas.

@TDDS

@TDDS(*valor_fecha*) convierte *valor_fecha* en el día de la semana en tailandés.

Argumentos

valor_fecha es un número de fecha.

Ejemplos

@TDDS(@FECHA(91,8,16)) = ศุกร์

@TCIFRA

@TCIFRA(*cadena_cifra*) convierte los números arábigos de *cadena_cifra* en una cadena con caracteres numéricos tailandeses.

Argumentos

cadena_cifra es una cadena que contiene números arábigos.

Ejemplos

@TCIFRA("16 สค. 34") = ๑๖ สค. ๓๔

Funciones @ similares

@ECIFRA convierte los caracteres numéricos tailandeses de una cadena de cifra en una cadena de números arábigos equivalente.

@ECIFRA

@ECIFRA(*cadena_cifra*) convierte el número tailandés contenido en *cadena_cifra* en una cadena numérica arábica.

Argumentos

cadena_cifra debe ser una cadena que contenga caracteres numéricos tailandeses.

Notas

Utilice **@ECIFRA** en combinación con **@VALOR** para convertir los datos insertados como cadenas numéricas tailandesas en valores numéricos para poder realizar cálculos.

Ejemplos

@ECIFRA("๑๖ ๙๗. ๓๔") = 16 ๙๗. 34

Funciones @ similares

@TCIFRA convierte los números arábigos contenidos en *cadena-cifra* en una cadena numérica tailandesa.

@VALOR convierte un número especificado como una cadena de texto en su valor numérico correspondiente.

@TNUMEROCADENA

@TNUMEROCADENA(*número*) convierte *número* en una cadena numérica tailandesa deletreada.

Argumentos

número puede ser un entero o un número con coma flotante.

Ejemplos

@TNUMEROCADENA(1000000) = หนึ่งล้านบาทถ้วน

Funciones @ similares

@NUMEROCADENA convierte *número* en un número deletreado (guardado como rótulo) en un formato no tailandés.

@TCADENAFECHA

@TCADENAFECHA(*valor_fecha*) convierte *valor_fecha* en una cadena de fecha tailandesa con formato corto.

Argumentos

valor_fecha es un número de fecha.

Ejemplos

@TCADENAFECHA(@FECHA(91,8,16)) = 16 สค. 34

Funciones @ similares

@TCADENAFECHAL convierte un valor de fecha en una cadena de fecha tailandesa con formato largo.

@TCADENAFECHAL

@TCADENAFECHAL(*valor_fecha*) convierte *valor_fecha* en una cadena de fecha tailandesa con formato largo.

Argumentos

valor_fecha es un número de fecha.

Ejemplos

@TCADENAFECHAL(@FECHA(91,8,16)) = 16 สิงหาคม 2534

Funciones @ similares

@TCADENAFECHA convierte *valor_fecha* en una cadena de fecha tailandesa con formato corto.

@IPE

@IPE(*principal;interés;plazo;período*) calcula la parte correspondiente al interés de un pago periódico en el que la parte correspondiente al principal es la misma en todos los periodos.

Argumentos

principal representa el valor del crédito y puede ser cualquier valor.

interés representa el tipo de interés del crédito y puede ser cualquier valor decimal o porcentual mayor que -1.

plazo representa el número de periodos de pago necesarios para amortizar el crédito. Puede ser cualquier entero positivo. Si se especifica algún valor que no sea entero, 1-2-3 truncará su parte decimal.

período representa los periodos de tiempo que se desean establecer para el pago del interés. Puede ser cualquier entero menor o igual que *plazo*. Si se especifica un valor que no sea un número entero 1-2-3 truncará su parte decimal para convertirlo en entero. El argumento *período* comienza a partir del valor cero (0).

Notas

La unidad de tiempo utilizada para calcular el *interés* debe coincidir con la de los periodos utilizados en *plazo*; por ejemplo, si se efectúan los cálculos para pagos mensuales, el interés y el plazo también deben expresarse en meses. Normalmente, esto supone dividir el tipo de interés por 12 y multiplicar el número de años de *plazo* por 12.

Ejemplos

Suponga que está pensando pedir un crédito de 8.000\$ a pagar en 3 años con un interés anual del 10,5%, compuesto mensualmente. Se desea determinar la parte del primer pago correspondiente al interés:

$$\text{@IPE}(8000;0,105/12;36;0) = 70\$$$

Para determinar el importe del primer pago:

$$\text{@IPE}(8000;0,105/12;36;0) + 8000/36 = 292,22\$$$

Funciones @ similares

@AMORTI calcula la parte correspondiente al interés de un pago periódico cuando el pago es igual en todos los periodos, pero la parte correspondiente al principal aumenta en la misma medida en que disminuye el interés.

@AMORTI

@AMORTI(*principal;interés;plazo;período*) calcula la parte correspondiente al interés de un pago periódico constante.

Argumentos

principal representa el valor del crédito y puede ser cualquier valor.

interés representa el tipo de interés del crédito y puede ser cualquier valor decimal o porcentual mayor que -1.

plazo representa el número de períodos de pago necesarios para amortizar el crédito y puede ser cualquier entero positivo. Si se especifica un valor que no sea entero se truncará su parte decimal para convertirlo en entero.

período representa los períodos de tiempo que se desean establecer para el pago del interés. Puede ser cualquier entero positivo menor o igual que *plazo*. Si se especifica un valor que no sea un número entero 1-2-3 truncará su parte decimal para convertirlo en entero.

Notas

La unidad de tiempo utilizada para calcular el *interés* debe coincidir con la de los períodos utilizados en *plazo*; por ejemplo, si se efectúan los cálculos para pagos mensuales, el interés y el plazo también deben expresarse en meses. Normalmente, esto supone dividir el tipo de interés por 12 y multiplicar el número de años de *plazo* por 12.

@AMORTINT(*principal;interés;plazo;período;período;0*) dará el mismo resultado que

@AMORTI(*principal;interés;plazo;período*).

Ejemplos

Suponga que está pensando pedir un crédito de 8.000\$ a pagar en 3 años con un interés anual del 10,5%, compuesto mensualmente. Los pagos mensuales son de 260,02\$. Se desea determinar la parte del primer pago correspondiente al interés:

@AMORT2(8000;0,105/12;36;1) = 70\$

Funciones @ similares

@AMORT y @AMORTIZ calculan el importe de los pagos periódicos necesarios para amortizar un crédito (*principal*), dado un tipo de interés periódico y un número de períodos de pago, suponiendo una anualidad ordinaria.

@AMORT2 calcula el importe de los pagos periódicos necesarios para amortizar un crédito dado un tipo de *interés* periódico y un número de períodos de pago, suponiendo una anualidad anticipada. @IPE calcula la parte correspondiente al interés de un pago periódico en el que la parte correspondiente al principal es la misma en todos los períodos. @AMORTINT calcula la suma de las partes correspondientes al interés de los pagos periódicos de un crédito a un tipo de interés dado para un determinado número de períodos de pago.

@HALFP

@HALFP(*cadena*) convierte los caracteres japoneses de doble byte de *cadena* en los caracteres correspondientes de un solo byte (ASCII).

Argumentos

cadena es una cadena que contiene los caracteres japoneses de doble byte que se desea convertir. Si *cadena* no es una cadena de texto, **@HALFP** dará como resultado ERR.

Notas

@HALFP convierte los caracteres alfabéticos, números y signos (como +, -, :) del juego de caracteres japoneses de doble byte (DBCS) en sus caracteres correspondientes en código ASCII. Los caracteres ASCII y aquellos para los que no existen valores ASCII no se podrán convertir, y al aplicarles **@HALFP** no experimentarán variación alguna. **@HALFP** tampoco convertirá el juego de caracteres de doble byte del silabario japonés Katakana en los de un solo byte del mismo silabario.

Funciones @ similares

@FULLP convierte caracteres de un solo byte en caracteres japoneses de doble byte.

@FULLP

@FULLP(*cadena*) convierte los caracteres de un solo byte (ASCII) de *cadena* en los caracteres japoneses de doble byte correspondientes.

Argumentos

rótulo es una cadena que contiene los caracteres en código ASCII que se desea convertir. Si *cadena* no es una cadena de texto, @FULLP dará como resultado ERR.

Notas

@FULLP convierte los caracteres ASCII de *rótulo* en los caracteres de doble byte correspondientes. Los caracteres pertenecientes al juego de caracteres de doble byte (DBCS) no se pueden convertir, y al aplicarles @FULLP no experimentarán variación alguna. @FULLP tampoco convertirá el juego de caracteres de un solo byte del silabario japonés Katakana en los de doble byte del mismo silabario.

Funciones @ similares

@HALFP convierte los caracteres japoneses de doble byte en caracteres de un solo byte.

@NUMEROCADENA

@NUMEROCADENA(*número*;*tipo*) convierte *número* en el texto japonés deletreado que corresponda al número, utilizando el formato especificado en *tipo*.

Argumentos

número puede ser cualquier número redondeado entre 0 y 9.999.999.999.999.999. 1-2-3 redondeará *número* al entero más próximo. Si se especifica un número que se encuentre fuera del rango indicado, @NUMEROCADENA dará como resultado ERR.

tipo determina el formato que se va a utilizar y puede ser cualquiera de los valores especificados en la tabla siguiente:

<i>tipo</i>	Da como resultado
1	Notación numérica Kanji ordinario
2	Notación numérica Kanji formal
3	Conversión simple de cifras a números Kanji uno a uno

Ejemplos

@NUMEROCADENA(1234567890,1) = 十二億三千四百五十六万七千八百九十

@NUMEROCADENA(1234567890,2) = 拾貳億參千四百五拾六万七千八百九拾

@NUMEROCADENA(1234567890,3) = 一 二 三 四 五 六 七 八 九 〇

@ACUMULADO2

@ACUMULADO2(*negociación*; *vencimiento*; *cupón*; [*valor_nominal*; *frecuencia*; *emisión*; *primer_cupón*; *tipo*]) calcula el importe acumulado de los intereses para activos financieros que proporcionan pagos periódicos de intereses, según el sistema utilizado en Japón.

Argumentos

negociación representa la fecha en que se negocia o transmite el activo; deberá especificarse como un número de fecha.

vencimiento es la fecha de vencimiento o amortización del activo; deberá especificarse como un número de fecha. Si el valor de *vencimiento* es igual o inferior al de *negociación*, @ACUMULADO2 dará como resultado ERR.

cupón es el tipo de interés anual del cupón del activo. Puede ser 0 o cualquier valor positivo, expresado en forma de número decimal.

valor_nominal es un argumento opcional que permite especificar el valor nominal del activo, es decir, el principal que se pagará al vencimiento. *valor_nominal* puede ser cualquier valor positivo. Si se omite el argumento *valor_nominal*, 1-2-3 utilizará el valor 100.

frecuencia es un argumento opcional que permite especificar el número de pagos de cupón por año. Deberá ser cualquiera de los valores indicados en la tabla siguiente:

<u>frecuencia</u>	<u>Frecuencia de los pagos de cupón</u>
1	Anual
2	Semestral; es el valor predeterminado si se omite este argumento

emisión es un argumento opcional que representa la fecha de emisión del activo; deberá especificarse como un número de fecha. Si el valor de *emisión* es superior al de *negociación*, @ACUMULADO2 dará como resultado ERR.

primer_cupón es un argumento opcional que representa la fecha de pago del primer cupón del activo; deberá especificarse como un número de fecha. Si el valor de *primer_cupón* es igual o inferior al de *emisión* o superior al de *vencimiento*, @ACUMULADO2 dará como resultado ERR.

tipo es un argumento opcional que permite especificar el tipo de bono u obligación. Deberá ser uno de los valores indicados en la tabla siguiente:

<u>tipo</u>	<u>Descripción</u>
0	Activo financiero del gobierno japonés; es el valor predeterminado si se omite este argumento
1	Activo financiero público o de una sociedad

No podrá utilizarse un argumento opcional si se omiten los que le preceden.

Ejemplos

El día 14 de junio de 1996 se adquiere un activo financiero del gobierno japonés por un valor de 100.000 yenes. La fecha de emisión de la obligación es el 1 de mayo de 1990, y la fecha de vencimiento es el 20 de junio del año 2000. El tipo de interés del cupón es del 9%, con pagos semestrales, y la fecha de pago del primer cupón es el 21 de diciembre de 1990.

El importe acumulado de los intereses será:

@ACUMULADO2(@FECHA(96;6;14);@FECHA(2000;6;20);0,09;100000;2;@FECHA(90;5;1);@FECHA(90;12;21)) = 4.339 yenes.

Funciones @ similares

@ACUMULADO calcula el importe acumulado de los intereses para activos financieros que proporcionan pagos periódicos de intereses. @PRECIO2 calcula el precio por cada 100 yenes de valor nominal para activos financieros que proporcionan pagos periódicos de intereses, utilizando el sistema japonés. @RENTABILIDAD2 da como resultado la rentabilidad de activos financieros que proporcionan pagos periódicos de intereses, utilizando el sistema

japonés. @PRECIO calcula el precio por cada 100\$ de valor nominal para activos financieros. @RENTABILIDAD calcula la rentabilidad para activos financieros que proporcionen pagos periódicos de intereses. @DURACION calcula la duración anual de un activo y @DURACIONM calcula la duración anual modificada para activos financieros que proporcionen pagos periódicos de intereses.

@VALFUT2

@VALFUT2(pagos;interés;plazo) calcula el valor futuro de una inversión constituida por una serie de *pagos* iguales por los que se retribuye una tasa de *interés* dada, durante el número de períodos indicado en *plazo* y suponiendo una anualidad anticipada.

Argumentos

pagos representa el importe que debe ser pagado en cada período y puede ser cualquier valor.

plazo representa el número de períodos precisos para efectuar los pagos y puede ser cualquier entero positivo. Si se incluye algún valor que no sea un entero, 1-2-3 truncará su parte decimal para convertirlo en entero.

interés representa el tipo de interés periódico y puede ser cualquier valor decimal o porcentual mayor que -1.

Notas

La unidad de tiempo utilizada para calcular el *interés* deberá coincidir con la utilizada en *plazo*; por ejemplo, si se efectúan los cálculos para pagos mensuales, el interés y el plazo también deberán expresarse en meses.

Normalmente, esto supondrá dividir el tipo de interés por 12 y multiplicar el número de años de *plazo* por 12.

Ejemplos

Se desea depositar en una cuenta la cantidad de 500.000 yenes al año durante los próximos 20 años para disponer de ellas tras la jubilación. La cuenta devenga un interés del 7,5% , capitalizado anualmente; el interés se pagará el último día de cada año. Se desea calcular el valor que tendrá la cuenta dentro de 20 años. La contribución anual se realiza el primer día del año.

@VALFUT2(500000;0,075;20) = 23.276.266 yenes, el valor de la cuenta al final de los 20 años.

El mismo resultado se obtendría aplicando la fórmula **@VALFUTURO**(500000;0,075;20;1).

Funciones @ similares

@VALFUT calcula el valor futuro de una inversión constituida por una serie de pagos iguales por los que se retribuye una tasa de interés dada, durante el número de períodos indicado en *plazo*, suponiendo una anualidad ordinaria.

@VALFUTURO calcula el valor futuro de una inversión que tiene un determinado valor actual, para una anualidad ordinaria o anticipada. **@VALACT** y **@VALACTUAL** determinan el valor actual de una inversión. **@VAN** calcula el valor actual neto de una inversión, descontando el valor futuro hasta el valor actual.

@AMORT2

@AMORT2(*principal;interés;plazo*) calcula el importe de los pagos periódicos necesarios para amortizar un crédito (*principal*) dado un tipo de *interés* periódico y un número de periodos de pago (*plazo*), suponiendo una anualidad anticipada.

Argumentos

principal representa el valor del crédito y puede ser cualquier valor.

interés representa el tipo de interés del crédito y puede ser cualquier valor decimal o porcentual mayor que -1.

plazo representa el número de periodos de pago necesarios para amortizar el crédito y puede ser cualquier entero positivo. Si se especifica un valor que no sea entero se truncará su parte decimal para convertirlo en entero.

Notas

La unidad de tiempo utilizada para calcular el *interés* debe coincidir con la de los periodos utilizados en *plazo*; por ejemplo, si se efectúan los cálculos para pagos mensuales, el interés y el plazo también deben expresarse en meses. Normalmente, esto supone dividir el tipo de interés por 12 y multiplicar el número de años de *plazo* por 12.

A diferencia de **@AMORT**, **@AMORT2** utilizará en el cálculo el final del plazo, en lugar del principio.

Ejemplos

Suponga que está pensando pedir un crédito de 8.000 yenes a pagar en 3 años con un interés anual del 10,5%, compuesto mensualmente. Los pagos se efectuarán el último día de cada mes. Para determinar la cuantía del pago mensual:

@AMORT2(8000;0,105/12;36) = 257,76 yenes

@AMORTIZ(8000;0,105/12;36;1) dará el mismo resultado.

Funciones @ similares

@AMORT y **@AMORTIZ** calculan el importe de los pagos periódicos necesarios para amortizar un crédito (*principal*), dado un tipo de interés periódico y un número de periodos de pago, suponiendo una anualidad ordinaria. **@AMORTI** calcula la parte correspondiente al interés de un pago periódico constante. **@AMORTINT** calcula la suma de las partes correspondientes al interés de los pagos periódicos de un crédito. **@AMORTPRINC** calcula la suma de las partes correspondientes al principal de los pagos periódicos de un crédito. **@PERIODO** calcula el número de pagos periódicos de un crédito. **@PERIODO2** calcula el número de periodos necesarios para que una serie de pagos periódicos iguales alcancen un valor futuro determinado a un tipo de interés dado, suponiendo una anualidad anticipada.

@PRECIO2

@PRECIO2(*negociación*; *vencimiento*; *cupón*; *rentabilidad*; [*amortización*; *base*]) calcula el precio por cada 100 yenes de valor nominal para activos financieros que proporcionan pagos periódicos de intereses, según el sistema utilizado en Japón.

Argumentos

negociación representa la fecha en que se negocia o transmite el activo. Debe ser un número de fecha.

vencimiento representa la fecha de vencimiento o amortización del activo. Debe ser un número de fecha. Si *vencimiento* es menor o igual que *negociación*, @PRECIO2 dará como resultado ERR.

cupón representa la tasa de interés anual del activo financiero, que se especifica como un valor decimal. Puede ser cualquier valor positivo o 0.

rentabilidad representa la rentabilidad anual, que se especifica como un valor decimal. Puede ser cualquier valor positivo.

amortización es un argumento opcional que permite especificar el valor de amortización del activo por cada 100 yenes de valor nominal. Puede ser cualquier valor positivo o 0. Si se omite el argumento *amortización*, 1-2-3 utilizará el valor 100.

base es un argumento opcional que permite especificar el tipo de base para el recuento de los días que va a utilizarse:

<u>base</u>	<u>Base para el recuento de los días</u>
0	30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeo 30/360

Si se incluye el argumento *base*, 1-2-3 efectuará una llamada a la función @PRECIO para calcular el valor.

Ejemplos

Un bono, cuyo vencimiento es el 20 de junio del 2000, se negocia el día 18 de julio de 1996. El tipo de interés del cupón es del 5,50% y la rentabilidad anual es del 5,61%. Para determinar el precio del bono:

@PRECIO2(@FECHA(96;7;18);@FECHA(100;6;20);0,055;0,0561;100) = 99,65 yenes

Funciones @ similares

@PRECIO calcula el precio por cada 100\$ de valor nominal para activos financieros que proporcionan pagos periódicos de intereses. @ACUMULADO calcula el interés acumulado para activos financieros que proporcionan pagos periódicos de intereses. @RENTABILIDAD calcula la rentabilidad de activos financieros que proporcionan pagos periódicos de intereses. @DURACION calcula la duración anual @DURACIONM calcula la duración anual modificada para activos financieros que proporcionan pagos periódicos de intereses. @ACUMULADO2 calcula el interés acumulado para activos financieros que proporcionan pagos periódicos de intereses, según el sistema utilizado en Japón. @RENTABILIDAD2 da como resultado la rentabilidad de activos financieros que proporcionan pagos periódicos de intereses, según el sistema utilizado en Japón.

@VALACT2

@VALACT2(*pagos;interés;plazo*) calcula el valor actual de una inversión constituida por una serie de *pagos* iguales por los que se retribuye una tasa de *interés* dada, durante el número de períodos indicado en *plazo* y suponiendo una anualidad anticipada.

Argumentos

pagos representa el importe que debe ser pagado en cada período y puede ser cualquier valor.

interés representa el tipo de interés periódico y puede ser un valor decimal o porcentual mayor que -1.

plazo representa el número de períodos precisos para efectuar los pagos y puede ser cualquier entero positivo. Si se especifica un valor que no sea un entero, 1-2-3 truncará su parte decimal.

Notas

La unidad de tiempo utilizada para calcular el *interés* debe coincidir con la de los períodos utilizados en *plazo*; por ejemplo, si se efectúan los cálculos para pagos mensuales, el interés y el plazo también deben expresarse en meses. Normalmente, esto supone dividir el tipo de interés por 12 y multiplicar el número de años de *plazo* por 12.

A diferencia de @VALACT, @VALACT2 utilizará el final del plazo para realizar el cálculo, en lugar del principio.

Utilice @VALACT2 para evaluar una inversión o para comparar una inversión con otras. @VALACT2 resulta útil para comparar diferentes tipos de inversiones; por ejemplo, podrá comparar una inversión de un pago único de un fondo de pensiones con una serie de pagos periódicos. Utilice @VALACT2 junto con @AMORT2 para crear una tabla de amortizaciones.

@VALACT2 es la función complementaria de @AMORT: @VALACT2 permitirá al usuario averiguar la magnitud del crédito que podrá solicitar, teniendo en cuenta la restricción del pago mensual que el usuario pueda permitirse; @AMORT realizaría la operación inversa, es decir, informar al usuario acerca de la magnitud del pago mensual, dados ciertos límites en el tamaño del crédito que se solicite.

Ejemplos

Suponga que ha ganado un premio de 1.000.000 de yenes que se cobrarán en 20 pagos anuales de 50.000 yenes al final de cada uno de los próximos 20 años, pero que también le ofrecen la oportunidad de recibir un pago único de 400.000 yenes en el momento actual, y que desea calcular cuál de ambas opciones resulta más rentable en yenes actuales.

Se supone que si se aceptan los pagos anuales de 50.000 yenes, se van invirtiendo a un tipo de interés del 8% capitalizado anualmente.

@VALACT2(50000;0,08;20) dará como resultado 530.180 yenes, lo que quiere decir que 1.000.000 de yenes pagados en veinte años equivalen a 530.180 yenes actuales.

Podrá obtenerse el mismo resultado aplicando @VALACTUAL(50000;0,08;20;1)

Funciones @ similares

@VALACT calcula el valor actual de una inversión, constituida por de una serie de pagos iguales, descontados a un tipo de interés dado durante el número de períodos indicado en plazo, suponiendo una anualidad ordinaria.

@VALACTUAL calcula el valor actual de una inversión que tiene un determinado valor futuro para anualidades ordinarias o anticipadas. @VALFUT y @VALFUTURO calculan el valor futuro de una inversión constituida por una serie de pagos iguales. @VALFUT2 calcula el valor futuro de una inversión constituida por una serie de pagos iguales por los que se retribuye una tasa de interés dada, durante el número de períodos indicado en plazo y suponiendo una anualidad anticipada. @VAN calcula el valor actual neto de una serie de inversiones, descontando el valor futuro al valor actual. @AMORT y @AMORTIZ calculan el importe de los pagos periódicos necesarios para amortizar un crédito a un tipo de interés dado durante un determinado número de períodos de pago. @AMORT2 calcula el importe de los pagos periódicos necesarios para amortizar un crédito dado un tipo de interés periódico y un número de períodos de pago, suponiendo una anualidad anticipada.

@PERIODO2

@PERIODO2(*pago*; *interés*; *valor_futuro*) calcula el número de *pagos* periódicos necesarios para acumular un determinado *valor_futuro*, suponiendo que los pagos son de igual cuantía, que la inversión rinde un tipo de *interés* periódico y que se realiza el cálculo para una anualidad anticipada.

Argumentos

pago representa el valor de las cantidades iguales invertidas periódicamente; puede ser cualquier valor excepto 0.

interés representa el tipo de interés periódico, y puede ser cualquier valor decimal o porcentual mayor que -1.

valor_futuro representa la cantidad que se desea acumular; puede ser cualquier valor.

Notas

Si se utiliza un *valor_futuro* negativo, **@PERIODO2** puede emplearse para calcular el tiempo necesario para amortizar un crédito. Por ejemplo, si desea conocer el período de amortización de un crédito de 10.000\$ a un tipo de interés anual del 10% con pagos de 1.174\$ al año, la fórmula **@ABS(@PERIODO2(1174;0,1;-10000))** calculará que se precisan 16 años para amortizar el crédito.

A diferencia de **@PERIODO**, **@PERIODO2** utiliza el final del período para realizar el cálculo, en lugar del principio.

Ejemplos

Al final de cada año se depositan 200.000 yenes en una cuenta de ahorro que devenga un interés anual del 7,5% capitalizado anualmente; para calcular el tiempo necesario para acumular 10.000.000 yenes se utilizará esta fórmula:

@PERIODO2(200000;0,075;10000000) = 20,76 años

Podrá obtenerse el mismo resultado utilizando la fórmula **@NPER(200000;0,075;10000000;1)**

Funciones @ similares

@PERIODO calcula el número de períodos necesarios para que una serie de pagos periódicos iguales alcancen cierto *valor_futuro* dado un determinado tipo de interés y suponiendo que el cálculo se realiza para una anualidad ordinaria. **@NPERIODO** calcula el número de períodos necesarios para que una serie de pagos iguales alcancen un *valor_futuro* determinado a un tipo de interés dado, pudiendo utilizarse en el cálculo una anualidad ordinaria o una anualidad anticipada. **@CPERIODO** calcula el número de períodos de capitalización necesarios para que una inversión de un solo depósito alcance un *valor_futuro*.

@RENTABILIDAD2

@RENTABILIDAD2(*negociación*; *vencimiento*; *cupón*; *precio*; [*amortización*; *base*]) da como resultado la rentabilidad de activos financieros que proporcionan pagos periódicos de intereses, según el sistema utilizado en Japón.

Argumentos

negociación es la fecha en que se negocia o transmite el activo. Debe ser un número de fecha.

vencimiento es la fecha de vencimiento o amortización del activo. Debe ser un número de fecha. Si *vencimiento* es menor o igual que *negociación*, @RENTABILIDAD2 dará como resultado ERR.

cupón es el tipo de interés anual del activo, representado como un número decimal. Puede ser cualquier valor positivo o 0.

precio es el precio en yenes del activo por cada 100 yenes de valor nominal. Puede ser cualquier valor positivo.

amortización es un argumento opcional que permite especificar el valor de amortización del activo por cada 100 yenes de valor nominal. Puede ser cualquier valor positivo o 0. Si se omite el argumento *amortización*, @RENTABILIDAD2 utilizará el valor 100.

base es un argumento opcional que permite especificar el tipo de base para el recuento de los días que va a utilizarse:

<u>base</u>	<u>Base para el recuento de los días</u>
0	30/360
1	Real/real
2	Real/360
3	Real/365
4	Europeo 30/360

Si se incluye el argumento *base*, 1-2-3 efectuará un llamada a la función @RENTABILIDAD para calcular el valor.

Ejemplos

Un bono, cuyo vencimiento es el 1 de diciembre de 1998, se negocia el 1 de julio de 1993. El tipo de interés del cupón semestral es del 5,50%, su precio es de 99,5 yenes y el valor de amortización es de 100 yenes. Se desea determinar la rentabilidad del bono:

@RENTABILIDAD2(@FECHA(93;7;1);@FECHA(98;12;1);0,055;99,5;100) = 0,056072

Funciones @ similares

@RENTABILIDAD da como resultado la rentabilidad para activos financieros que proporcionan pagos periódicos de intereses.

@ACUMULADO2 calcula el interés acumulado para activos financieros que proporcionan pagos periódicos de intereses, según el sistema utilizado en Japón. @PRECIO2 calcula el precio por cada 100 yenes de valor nominal para activos financieros que proporcionan pagos periódicos de intereses, según el sistema utilizado en Japón.

@ACUMULADO calcula el interés acumulado para activos financieros que proporcionan pagos periódicos de intereses. @PRECIO calcula el precio por cada 100 dólares de valor nominal para activos financieros que proporcionan pagos periódicos de intereses. @DURACION calcula la duración anual para activos financieros que proporcionan pagos periódicos de intereses. @DURACIONM calcula la duración anual modificada para activos financieros que proporcionan pagos periódicos de intereses.

@VINCULODATOS

@VINCULODATOS(*aplicación;archivo;elemento*;*[formato;máx_filas;máx_colsmáx_hojas]*) establece un vínculo OLE entre los datos.

Es posible modificar el vínculo modificando los argumentos de la función **@VINCULODATOS**.

Argumentos

aplicación es una cadena de texto que permite especificar el nombre de una aplicación de Windows que pueda comportarse como servidora de vínculos OLE. Este argumento no se utiliza en 1-2-3 97, pero se proporciona para mantener la compatibilidad con las versiones anteriores de 1-2-3.

archivo es una cadena de texto que permite especificar el nombre del archivo de la aplicación con el que se desea establecer el vínculo.

elemento es una cadena de texto que permite especificar el nombre del elemento del archivo de la aplicación servidora con la que se desea establecer el vínculo. Se trata del elemento de la aplicación servidora desde el cual se desean transferir los datos a través del vínculo.

formato es una cadena de texto que permite especificar uno de los formatos del Portapapeles, a saber: Texto, WK1 o WK3.

Si se omite el formato, 1-2-3 utilizará de forma predeterminada el formato Texto del Portapapeles.

máx_filas, *máx_cols* y *máx_hojas* sirven para especificar el número máximo de filas, columnas y hojas para el rango de destino. Si se omite alguno de estos argumentos, 1-2-3 utilizará tantas filas, columnas u hojas como requiera el rango.

Notas

Siempre que se genere un vínculo mediante el comando **Edición - Pegado especial**, y se elija como tipo de vínculo Texto, WK1 o WK3, **@VINCULODATOS** aparecerá en la primera celda del rango de destino del vínculo.

Los vínculos creados por el usuario recibirán automáticamente los nombres "Vínculo de datos 1", "Vínculo de datos 2" y así sucesivamente.

Consulte también el tema [Creación de un vínculo OLE en 1-2-3](#).

Ejemplos

La siguiente fórmula creará un vínculo OLE con el archivo PRESTAMO.LWP de la aplicación Word Pro, utilizando la función **@VINCULODATOS**.

@VINCULODATOS("WordPro";"C:\LOTUS\WORK\WORDPRO\PRESTAMO.LWP";"!Link_BookMark1")

@PRODUCTO

@PRODUCTO(*lista*) multiplica los valores de *lista*.

Argumentos

lista puede contener cualquier combinación de los elementos siguientes: números, fórmulas numéricas y direcciones o nombres de rangos que contengan números o fórmulas numéricas. Utilice separadores de argumentos para separar los elementos de *lista*.

Consulte también el tema Argumentos de las funciones @ estadísticas.

Ejemplos

@PRODUCTO(2;4;6;8) = 384

Funciones @ similares

@FACT calcula el factorial de un valor. **@SUMA** suma los valores de una lista. **@SUMAPRODUCTOS** multiplica los valores contenidos en las celdas homólogas de varios rangos y, a continuación, suma los productos.

Funciones @ de la A a la Z

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

A

@@(posición)

Permite obtener de forma indirecta el contenido de la celda especificada en *posición*.

@ABS(x)

Calcula el valor absoluto (positivo) de x .

@ACOS(x)

Calcula el arco coseno de un valor. El arco coseno de x es el ángulo cuyo coseno es x .

@ACOSEC(x)

Calcula el arco cosecante de un valor. El arco cosecante de x es el ángulo cuya cosecante es x .

@ACOSECH(x)

Calcula el arco cosecante hiperbólica de un valor. El arco cosecante hiperbólica de x es el ángulo cuya cosecante hiperbólica es x .

@ACOSH(x)

Calcula el arco coseno hiperbólico de un valor. El arco coseno hiperbólico de x es el ángulo cuyo coseno hiperbólico es x .

@ACOT(x)

Calcula el arco cotangente de un valor. El arco cotangente de x es el ángulo cuya cotangente es x .

@ACOTH(x)

Calcula el arco cotangente hiperbólica de un valor. El arco cotangente hiperbólica de x es el ángulo cuya cotangente hiperbólica es x .

@ACTUAL

Calcula el número correspondiente a la fecha (parte entera) y a la hora (parte decimal) según la fecha y hora del reloj del PC.

@ACUMULADO(negociación;emisión;primer_cupón;cupón;[valor_nominal;frecuencia;base])

Calcula el importe acumulado de los intereses para activos financieros que proporcionan pagos periódicos de intereses.

@ACUMULADO2(negociación;vencimiento;cupón;[valor_nominal;frecuencia;emisión;primer_cupón;tipo])

Calcula el importe acumulado de los intereses para activos financieros que proporcionan pagos periódicos de intereses, según el sistema utilizado en Japón.

@ALEAT

Genera un número aleatorio entre 0 y 1 con una precisión de hasta 15 decimales. Cada vez que 1-2-3 recalcula la hoja, @ALEAT genera un nuevo número aleatorio.

@AMORT(principal;interés;plazo)

Calcula el importe de los pagos periódicos necesarios para amortizar un crédito (*principal*), dado un tipo de *interés* periódico y un número de períodos de pago (*plazo*), suponiendo una anualidad ordinaria.

@AMORT2(principal;interés;plazo)

Calcula el importe de los pagos periódicos necesarios para amortizar un crédito (*principal*) dado un tipo de *interés* periódico y un número de períodos de pago (*plazo*), suponiendo una anualidad anticipada.

@AMORTC(principal;interés;plazo)

Calcula el importe de los pagos periódicos necesarios para amortizar un crédito (*principal*), dado un tipo de *interés* periódico y un número de períodos de pago (*plazo*), de acuerdo con el método utilizado en Canadá.

@AMORTI(principal;interés;plazo;período)

Calcula la parte correspondiente al *interés* de un pago periódico constante.

@AMORTINT(principal;interés;plazo;período_inicial;[período_final;tipo;valor_futuro])

Calcula la suma de las partes correspondientes al *interés* de los pagos periódicos de un crédito para un determinado número de períodos de pago (*plazo*)

@AMORTIZ(principal;interés;plazo;[tipo;valor_futuro])

Calcula el importe de los pagos periódicos necesarios para amortizar un crédito (*principal*), dado un tipo de *interés* periódico y un número de períodos de pago (*plazo*).

@AMORTPRINC(principal;interés;plazo;período_inicial;[período_final;tipo;valor_futuro])

Calcula la suma de las partes correspondientes al principal de los pagos periódicos de un crédito (*principal*) con un tipo de *interés* periódico dado y un determinado número de períodos de pago (*plazo*).

@AÑO(número_fecha;[tipo])

Extrae la parte correspondiente al año, es decir, un entero del 0 (año 1900) al 199 (año 2099), a partir de *número_fecha*.

@ASEC(x)

Calcula el arco secante de un valor. El arco secante de x es el ángulo cuya secante es x .

@ASECH(x)

Calcula el arco secante hiperbólica de un valor. El arco secante hiperbólica de x es el ángulo cuya secante hiperbólica es x .

@ASEN(x)

Calcula el arco seno de un valor. El arco seno de x es el ángulo cuyo seno es x .

@ASENH(x)

Calcula el arco seno hiperbólico de un valor. El arco seno hiperbólico de x es el ángulo cuyo seno hiperbólico es x .

@ASIMETRIA(rango;[tipo])

Calcula el coeficiente de asimetría para los valores de *rango*.

@ATAN(x)

Calcula el arco tangente de un valor. El arco tangente de x es el ángulo cuya tangente es x . El resultado de @ATAN será un ángulo, expresado en radianes, comprendido entre $-\pi/2$ y $\pi/2$, ambos inclusive.

@ATAN2(x;y)

Calcula el arco tangente de un valor. El arco tangente es el ángulo cuya tangente es y/x . El resultado de @ATAN2 será un ángulo, expresado en radianes, comprendido entre $-\pi$ y π , dependiendo del signo de x e y .

@ATANH(x)

Calcula el arco tangente hiperbólica de un valor. El arco tangente hiperbólica de x es el ángulo cuya tangente hiperbólica es x .

B

@BCUENTA(entrada;campo;[criterios])

Cuenta las celdas no vacías existentes en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *criterios* especificados.

@BCUENTAPURA(entrada;campo;[criterios])

Cuenta las celdas que contienen valores existentes en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *criterios* especificados.

@BESSEL(*x;n*)

Calcula la función de Bessel de segunda especie de orden entero $I_n(x)$.

@BESSELJ(*x;n*)

Calcula la función de Bessel de primera especie de orden entero $J_n(x)$.

@BESSELK(*x;n*)

Calcula la función de Bessel de segunda especie de orden entero $K_n(x)$.

@BESSELY(*x;n*)

Calcula la función de Bessel de primera especie de orden entero $Y_n(x)$, también conocida como función de Neumann.

@BETA(*z;w*)

Calcula la función beta.

@BETA(*a;b;x*)

Calcula la función beta incompleta.

@BEXTRAER(*entrada;campo;[criterios]*)

Extrae un valor o un rótulo de un *campo* de una tabla de base de datos perteneciente al registro que cumple los *criterios* especificados.

@BINOMIAL(*intentos;aciertos;probabilidad;[tipo]*)

Calcula la función de probabilidad binomial (o de Bernouilli) o la distribución binomial acumulativa.

@BMAX(*entrada;campo[criterios]*)

Selecciona el valor máximo entre los contenidos en un *campo* de una tabla de base de datos, pero únicamente tiene en cuenta los registros que cumplen los *criterios* especificados.

@BMEDIA(*entrada;campo;[criterios]*)

Calcula la media aritmética de los valores contenidos en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *criterios* especificados.

@BMIN(*entrada;campo;[criterios]*)

Selecciona el valor mínimo entre los contenidos en un *campo* de una tabla de base de datos, pero únicamente tiene en cuenta los registros que cumplen los *criterios* especificados.

@BSTD(*entrada;campo;[criterios]*)

Calcula la desviación típica de una población constituida por los valores contenidos en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *criterios* especificados.

@BSTDM(*entrada;campo;[criterios]*)

Calcula la desviación típica de una muestra constituida por los valores contenidos en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *criterios* especificados.

@BSUMA(*entrada;campo;[criterios]*)

Calcula la suma de los valores contenidos en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *criterios* especificados.

@BUSCAR(*cadena_búsqueda;cadena;posición_inicial*)

Calcula la posición que ocupa *cadena_búsqueda* la primera vez que aparece dentro de *cadena*. 1-2-3 comenzará la búsqueda a partir de la posición indicada por *posición_inicial*.

@BUSCARB(*cadena_búsqueda;cadena;posición_inicial*)

Busca la posición (en bytes) que ocupa *cadena_búsqueda* la primera vez que aparece dentro de *cadena* a partir de la posición (en bytes) especificada por *posición_inicial*.

@BUSCARCONTENIDO(*contenido_celda;rango;[tipo]*)

Da como resultado la posición que ocupa en *rango* la celda cuyo contenido es igual a *contenido_celda*.

@BUSCARMAXIMO(*lista_rangos*)

Da como resultado una referencia absoluta a la celda que contiene el valor máximo de una lista de rangos.

@BUSCARMINIMO(*lista_rangos*)

Da como resultado una referencia absoluta a la celda que contiene el valor mínimo de una lista de rangos.

@BVAR(*entrada;campo;[criterios]*)

Calcula la varianza de una población constituida por los valores contenidos en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *criterios* especificados.

@BVARM(*entrada;campo;[criterios]*)

Calcula la varianza de una muestra constituida por los valores contenidos en un *campo* de una tabla de base de datos, pero sólo tiene en cuenta los registros que cumplen los *criterios* especificados.

C

@C(rango)

Arroja como resultado el contenido de la primera celda de *rango* en forma de rótulo.

@CADENA(x;n)

Convierte el valor *x* en un rótulo utilizando el formato especificado por *n*.

@CADENAFECHA(número_fecha)

Convierte un número de fecha en su fecha equivalente y lo muestra como un rótulo aplicándole el formato de fecha internacional predeterminado.

@CAR(x)

Arroja como resultado el carácter correspondiente al código *x* del Juego de caracteres Multibyte de Lotus (LMBCS).

@CELDA(atributo;posición)

Proporciona información sobre la primera celda de *posición*.

@COCIENTE(x;y)

Calcula el cociente de *x/y* y trunca la parte decimal del resultado para convertirlo en un entero.

@CODIGO(cadena)

Arroja como resultado el código del Juego de caracteres Multibyte de Lotus (LMBCS) correspondiente al primer carácter de *cadena*.

@COLS(rango)

Cuenta el número de columnas contenidas en *rango*.

@COMBIN(n;r)

Calcula el coeficiente binomial para *n* y *r*. El coeficiente binomial es el número de combinaciones que pueden realizarse con *n* elementos tomados de *r* en *r*, sin tener en cuenta el orden.

@CONSULH(x;rango;fila)

Busca el contenido de una celda en una fila determinada de una tabla horizontal de consulta, en la columna especificada por *x*.

@CONSULV(x;rango;columna)

Busca el contenido de una celda en una columna determinada de una tabla vertical de consulta.

@CONVERTFECHA(fecha;tipo_entrada;tipo_salida)

Convierte una fecha hijri (árabe), farsi (iraní) o hebrea (judía) en la fecha gregoriana equivalente y viceversa. Sólo para versiones bidireccionales de 1-2-3.

@CONVERTREF(referencia)

Convierte las letras de la A a la IV de las columnas o de las hojas de 1-2-3 en los números del 1 al 256 correspondientes, y viceversa.

@COORD(hoja;columna;fila;absoluta)

Crea una dirección de celda a partir de unos valores que correspondan a la *hoja*, *columna* y *fila*.

@CORREL(rango1;rango2)

Calcula el coeficiente de correlación de los valores contenidos en *rango1* y en *rango2*.

@COS(z)

Calcula el coseno del ángulo *z*. El ángulo deberá expresarse en radianes.

@COSEC(z)

Calcula la cosecante del ángulo *z*. El ángulo deberá expresarse en radianes.

@COSECH(z)

Calcula la cosecante hiperbólica del ángulo *z*. El ángulo deberá expresarse en radianes.

@COSH(z)

Calcula el coseno hiperbólico del ángulo *z*. El resultado de @COSH será un valor mayor o igual que 1. El ángulo deberá expresarse en radianes.

@COT(z)

Calcula la cotangente del ángulo *z*. El ángulo deberá expresarse en radianes.

@COTH(z)

Calcula la cotangente hiperbólica del ángulo *z*. El ángulo deberá expresarse en radianes.

@COVAR(rango1;rango2;[tipo])

Calcula la covarianza de una población o de una muestra constituida por los valores contenidos en *rango1* y en

rango2.

@CPERIODO(*interés;valor_futuro;valor_actual*)

Calcula el número de períodos de capitalización necesarios para que una inversión (*valor_actual*) alcance un *valor_futuro* determinado, a un tipo de *interés* fijo.

@CRITBINOMIAL(*intentos;probabilidad;alfa*)

Calcula el valor máximo para el cual la distribución binomial acumulativa es menor o igual que *alfa*.

@CUARTIL(*x;rango*)

Da como resultado un cuartil determinado.

@CUENTA(*lista*)

Cuenta las celdas no vacías existentes en una *lista* de rangos.

@CUENTAPURA(*lista*)

Cuenta las celdas contenidas en una *lista* de rangos, excluyendo aquellas que contengan rótulos.

@CURTOSIS(*rango;[tipo]*)

Calcula el coeficiente de curtosis o de apuntamiento para los valores de *rango*.

D

@D360(*fecha_inicial;fecha_final*)

Calcula el número de días existentes entre dos números de fecha, basándose en un año de 360 días.

@DB(*coste;valor_residual;vida;período*)

Calcula, por el método de los saldos decrecientes (Declining Balance), la cuota de amortización de un bien de inversión.

@DDB(*coste;valor_residual;vida;período*)

Calcula, por el método de los saldos decrecientes al 200% (Double Declining Balance), la cuota de amortización de un bien de inversión.

@DECIL(*x;rango*)

Da como resultado un decil determinado.

@DECIMAL(*hexadecimal*)

Convierte un valor *hexadecimal* en su equivalente decimal con el signo correspondiente.

@DEFINIRCADENA(*cadena;longitud;[alineación]*)

Da como resultado un rótulo de texto del tamaño indicado en *longitud*.

@DERECHA(*cadena;n*)

Da como resultado los últimos *n* caracteres de *cadena*.

@DERECHAB(*cadena;n*)

Da como resultado los últimos *n* bytes de *cadena*.

@DESPEJAR(*cadena*)

Suprime los caracteres no imprimibles de *cadena*.

@DESVCUAD(*lista*)

Calcula la suma de los cuadrados de las desviaciones de los valores de *lista* con relación a su media aritmética.

@DESVMEDIA(*lista*)

Calcula la desviación media, también denominada desviación absoluta o aritmética, de los valores de *lista*.

@DIA(*número_fecha*)

Calcula el día del mes, un entero del 1 al 31, correspondiente a *número_fecha*.

@DIAS(*fecha_inicial;fecha_final;[base]*)

Calcula el número de días existentes entre *fecha_inicial* y *fecha_final*, utilizando una *base* determinada para el recuento de los días.

@DIAS360(*fecha_inicial;fecha_final*)

Calcula el número de días existentes entre dos fechas, basándose en un año de 360 días y de acuerdo con la normativa del mercado de valores de los Estados Unidos.

@DIASEMANA(*número_fecha*)

Calcula el día de la semana correspondiente a *número_fecha* y lo muestra como un entero del 0 (lunes) al 6 (domingo).

@DIFERFECHA(*fecha_inicial;fecha_final;formato*)

Calcula el número de años, meses o días existentes entre *fecha_inicial* y *fecha_final*.

@DISTE(x;grados_libertad1;grados_libertad2;[tipo])

Calcula la distribución F.

@DISTJI(x;grados_libertad;[tipo])

Calcula la distribución ji cuadrado (χ^2).

@DISTT(x;grados_libertad;[tipo;colas])

Calcula la distribución t de Student.

@DURACION(negociación;vencimiento;cupón;rentabilidad;[frecuencia;base])

Calcula la duración anual de activos financieros que producen intereses periódicos.

@DURACIONM(negociación;vencimiento;cupón;rentabilidad;[frecuencia;base])

Calcula la duración anual modificada de activos financieros que producen intereses periódicos.

E

@ECIFRA(cadena_cifra)

Convierte los caracteres numéricos tailandeses de *cadena_cifra* en una cadena numérica arábica.

@ELEGIR(x;lista)

Da como resultado el objeto enésimo de *lista*, donde n es el valor de x.

@ENT(x)

Arroja como resultado la parte entera de x.

@ERR

Arroja como resultado el valor ERR.

@ERRMEDIA(rango)

Calcula el error típico de la distribución muestral de la media para los valores contenidos en *rango*.

@ESARCHIVO(nombre_archivo;[tipo])

Comprueba si *nombre_archivo* es uno de los archivos existentes en la memoria o en el disco y da como resultado 1 (verdad) o 0 (falso).

@ESCADENA(x)

Comprueba si x es una cadena de texto o una celda que contiene un rótulo o una fórmula cuyo resultado sea un rótulo y da como resultado 1 (verdad) o 0 (falso).

@ESERR(x)

Comprueba si x es el valor ERR y da como resultado 1 (verdad) o 0 (falso).

@ESMACRO(nombre)

Comprueba si *nombre* es una instrucción de macro complementaria definida y da como resultado 1 (verdad) o 0 (falso).

@ESND(x)

Comprueba si x es el valor ND y da como resultado 1 (verdad) o 0 (falso).

@ESNUM(x)

Comprueba si x es un valor, ND, ERR o una celda vacía y da como resultado 1 (verdad) o 0 (falso).

@ESRANGO(rango)

Comprueba si *rango* corresponde a un nombre de rango definido o a una dirección de rango válida y da como resultado 1 (verdad) o 0 (falso).

@ESTACOMPREDIDO(valor;limite1;limite2;[tipo])

Comprueba si *valor* está comprendido entre *limite1* y *limite2* y da como resultado 1 (verdad) o 0 (falso).

@ESTAVACIA(x)

Comprueba si x corresponde a una celda vacía o no; da como resultado 1 (verdad) o 0 (falso).

@ESTIMACION(x;rango_y;rango_x)

Da como resultado una valor estimado para x a partir de la tendencia lineal existente entre los valores del *rango_y* y los del *rango_x*.

@EXP(x)

Calcula el valor de la constante e (aproximadamente 2,718282) elevada a la potencia x.

@EXP2(x)

Calcula el valor de la constante e (aproximadamente 2,718282) elevada a la potencia (-x^2).

F

@FACT(*n*)

Calcula el factorial de *n*.

@FACTLN(*n*)

Calcula el logaritmo neperiano del factorial de *n*.

@FALSO

Arroja como resultado el valor lógico 0 (falso).

@FECHA(*año;mes;día*)

Calcula el número de fecha correspondiente al *año*, *mes* y *día* especificados.

@FER(*límite_inferior*;*límite_superior*)

Calcula la función error integrada entre *límite_inferior* y *límite_superior*.

@FERC(*x*)

Calcula la función error complementaria, integrada entre *x* e ∞ (infinito).

@FERD(*x*)

Calcula la derivada de la función error.

@FILAS(*rango*)

Cuenta el número de filas contenidas en *rango*.

@FULLP(*cadena*)

Convierte los caracteres de un solo byte (ASCII) de *cadena* en los caracteres japoneses de doble byte correspondientes.

G

@GAMMA(*x*)

Calcula la función gamma.

@GAMMAI(*a;x*;*complemento*)

Calcula la función gamma incompleta.

@GAMMALN(*x*)

Calcula el logaritmo neperiano de la función gamma.

@GRADARAD(*grados*)

Convierte *grados* en radianes.

H

@HALFP(*cadena*)

Convierte los caracteres japoneses de doble byte de *cadena* en los caracteres correspondientes de un solo byte (ASCII).

@HEX(*x*)

Convierte un número decimal en su equivalente hexadecimal.

@HOJAS(*rango*)

Cuenta el número de hojas comprendidas en *rango*.

@HORA(*número_hora*)

Extrae la hora, es decir, un valor comprendido entre 0 (medianoche) y 23 (23:00 u 11:00 PM), ambos inclusive, a partir de *número_hora*.

@HOY

Calcula el número de fecha correspondiente a la fecha actual indicada por el PC.

I

@IGUAL(*cadena1;cadena2*)

Da como resultado 1 (verdad) si *cadena1* y *cadena2* coinciden exactamente; en caso contrario, dará como resultado 0 (falso).

@IMPAR(*x*)

Redondea el valor *x* al primer número entero impar que encuentre al alejarse de 0.

@INDICADORCELDA(*atributo*)

Proporciona información sobre la celda actual.

@INDICE(*rango;columna;fila*;*hoja*)

Da como resultado el contenido de la celda situada en una determinada *columna*, *fila* y *hoja* de un rango.

@INFO(*atributo*)

Facilita información sobre el sistema para la sesión actual de trabajo de 1-2-3.

@INFOESCENARIO(*opción;grupo_versiones;[autor]*)

Proporciona información relativa a un grupo de versiones.

@INFOFECHA(*fecha;atributo*)

Proporciona información acerca de un número de *fecha*.

@INFOVERSION(*opción;rango_version;nombre;[autor]*)

Proporciona información relativa a una versión.

@INTERSECCION(*rango;título_columna;título_fila;[título_hoja]*)

Da como resultado el contenido de una celda de *rango* situada en la intersección especificada mediante *título_columna*, *título_fila* y, opcionalmente, *título_hoja*.

@IPE(*principal;interés;plazo;período*)

Calcula la parte correspondiente al interés de un pago periódico en el que la parte correspondiente al principal es la misma en todos los periodos.

@ISAAF(*nombre*)

Comprueba si *nombre* es una función complementaria definida de Lotus Script y da como resultado 1 (verdad) o 0 (falso).

@ISAPP(*nombre*)

Comprueba si *nombre* es un programa complementario que se encuentra cargado en memoria en el momento actual y da como resultado 1 (verdad) o 0 (falso).

@IZDA(*cadena;n*)

Da como resultado los *n* primeros caracteres de *cadena*.

@IZDAB(*cadena;n*)

Da como resultado los *n* primeros bytes de *cadena*.

J

K

L

@LABORABLE(*fecha_inicial;días;[rango_festivos;fin_de_semana]*)

Calcula el número de fecha correspondiente a una fecha que es un determinado número de *días* anterior o posterior a *fecha_inicial*.

@LABORABLESNETOS(*fecha_inicial;fecha_final;[rango_festivos;fin_de_semana]*)

Calcula el número de días que transcurren desde la *fecha_inicial* hasta la *fecha_final*, excluyendo los fines de semana y los festivos o de vacaciones.

@LN(*x*)

Calcula el logaritmo neperiano (en base e) de *x*.

@LOG(*x*)

Calcula el logaritmo decimal (en base 10) de *x*.

@LONGITUD(*cadena*)

Cuenta el número de caracteres de *cadena*.

@LONGITUDB(*cadena*)

Cuenta el número de bytes de *cadena*.

M

@MAX(*lista*)

Halla el valor máximo entre los contenidos en *lista*.

@MAXPURA(*lista*)

Halla el valor máximo entre los contenidos en *lista*, pero ignora las celdas que contienen rótulos.

@MAYOR(*rango;n*)

Halla el *enésimo* valor más elevado de los comprendidos en *rango*.

@MAYUSC(cadena)

Convierte en mayúsculas todas las letras contenidas en *cadena*.

@MED(cadena;posición_inicio;n)

Copia *n* caracteres de *cadena*, a partir del carácter situado en *posición_inicio*.

@MEDB(cadena;posición_inicio;n)

Copia *n* bytes de *cadena*, a partir de los datos situados en el byte *posición_inicio*.

@MEDIA(lista)

Calcula la media aritmética de los valores de *lista*.

@MEDIAGEOM(lista)

Calcula la media geométrica de los valores de *lista*.

@MEDIANA(lista)

Da como resultado la mediana de los valores comprendidos en *lista*.

@MEDIANAPURA(lista)

Da como resultado la mediana de los valores contenidos en *lista*, ignorando las celdas vacías, los rótulos y las fórmulas cuyos resultados sean rótulos.

@MEDIAPOND(rango_datos;rango_proporciones:[tipo])

Calcula la media ponderada de los valores contenidos en *rango_datos*.

@MEDIAPURA(lista)

Calcula la media aritmética de los valores de *lista*, pero ignorará las celdas que contengan rótulos.

@MEDIARMON(lista)

Calcula la media armónica de los valores de *lista*.

@MENOR(rango;n)

Halla el *enésimo* valor más pequeño de los comprendidos en *rango*.

@MES(número_fecha)

Extrae la parte correspondiente al mes (un número del 1 al 12) a partir de *número_fecha*.

@MESSIGUIENTE(fecha_inicial;meses:[día_del_mes;base])

Calcula el número de fecha correspondiente a la fecha de un número de *meses* determinado anterior o posterior a la *fecha_inicial*.

@MIN(lista)

Halla el valor mínimo entre los contenidos en *lista*.

@MINPURA(lista)

Halla el valor mínimo entre los contenidos en *lista*, pero ignora las celdas que contienen rótulos.

@MINUSC(cadena)

Convierte en minúsculas todas las letras contenidas en *cadena*.

@MINUTO(número_hora)

Extrae la parte correspondiente a los minutos, es decir, un valor entre 0 y 59 a partir de *número_hora*.

@MODULO(x;y)

Calcula el resto (módulo) de *x/y*. El signo del resultado coincide con el de *y*.

N

@N(rango)

Arroja como resultado el contenido de la primera celda de *rango* en forma de valor. Si la celda contiene un rótulo,

@N dará como resultado el valor 0.

@ND

Da como resultado el valor ND (no disponible).

@NOMBRERANGO(celda)

Arroja como resultado el nombre del rango donde se encuentra *celda*.

@NOMPROPIO(cadena)

Convierte en mayúscula la primera letra de cada palabra de *cadena* y en minúsculas el resto de las letras.

@NORMAL(x:[media;std;tipo])

Calcula la función de distribución normal para el valor *x*.

@NPERIODO(pago;interés;valor_futuro:[tipo;valor_actua])

Calcula el número de *pagos* periódicos necesarios para acumular un determinado *valor_futuro* si la inversión

rinde cierto *interés* periódico.

@NUMEROCADENA(*número;tipo*)

Convierte *número* en el texto japonés deletreado que corresponda al número, utilizando el formato especificado en *tipo*.

O

P

@PAR(*x*)

Redondea el valor *x* al primer número entero par que encuentre al alejarse de 0.

@PERCENTILP(*x;rango*)

Calcula el valor correspondiente al percentil *x* para la muestra contenida en *rango*.

@PERCENTILV(*x;rango;[decimales]*)

Halla el percentil en que se encuentra el valor *x* teniendo en cuenta los valores contenidos en *rango*.

@PERIODO(*pago;interés;valor_futuro*)

Calcula el número de *pagos* periódicos iguales necesarios para acumular un determinado *valor_futuro* a un tipo de *interés* dado y suponiendo una anualidad ordinaria.

@PERIODO2(*pago;interés;valor_futuro*)

Calcula el número de *pagos* periódicos iguales necesarios para acumular un determinado *valor_futuro*, a un tipo de *interés* dado y suponiendo una anualidad anticipada.

@PERMUT(*n;r*)

Calcula el número secuencias ordenadas (permutaciones) de *r* objetos que pueden seleccionarse de un conjunto de *n* objetos.

@PI

Da como resultado el valor π (tomado como 3,14159265358979).

@POISSON(*x;media;[acumulativo]*)

Calcula la distribución de Poisson.

@POSICION(*elemento;rango;[orden]*)

Calcula la posición relativa o el tamaño de un valor de un *rango* con relación a los restantes valores del rango.

@PRECIO(*negociación;vencimiento;cupón;rentabilidad;[amortización];[frecuencia];[base]*)

Calcula el precio por cada 100\$ de valor nominal para activos financieros que proporcionan pagos periódicos de intereses.

@PRECIO2(*negociación;vencimiento;cupón;rentabilidad;[amortización;base]*)

Calcula el precio por cada 100 yenes de valor nominal para activos financieros que proporcionan pagos periódicos de intereses, según el sistema utilizado en Japón.

@PRODUCTO(*lista*)

Multiplica los valores de *lista*.

Q

R

@RADAGRAD(*radianes*)

Convierte *radianes* en grados.

@RCUAD(*x*)

Da como resultado la raíz cuadrada positiva de *x*.

@RCUADPORPI(*x*)

Calcula la raíz cuadrada de $x*\pi$.

@REDOND(*x;n*)

Redondea el valor *x* al múltiplo más próximo de 10 elevado a *n*.

@REDONDABAJO(*x;n;dirección*)

Redondea el valor *x* al múltiplo anterior más próximo de 10 elevado a *n*.

@REDONDARRIBA(*x;n;[dirección]*)

Redondea el valor x al múltiplo siguiente más próximo de 10 elevado a n .

@REDONDMULTP(x ;múltiplo;[dirección])

Redondea el valor x al *múltiplo* más próximo.

@REGRESION(*rango_x*;rango_y;atributo;[calcular])

Realiza un análisis de regresión lineal múltiple y da como resultado el parámetro de la regresión especificado.

@RENTABILIDAD(*negociación*;vencimiento;cupón;precio;[amortización];[frecuencia;base])

Calcula la rentabilidad de activos financieros que proporcionan pagos periódicos de intereses.

@RENTABILIDAD2(*negociación*;vencimiento;cupón;precio;[amortización;base])

Da como resultado la rentabilidad de activos financieros que proporcionan pagos periódicos de intereses, según el sistema utilizado en Japón.

@REPETIR(*cadena*;n)

Reproduce *cadena* el número de veces especificado en n .

@RESTO(x ;y)

Calcula el resto (módulo) de x/y . El signo del resultado coincide con el de x .

S

@SEC(z)

Calcula la secante del ángulo z .

@SECH(z)

Calcula la secante hiperbólica del ángulo z .

@SEGUNDO(*número_hora*)

Extrae la parte correspondiente a los segundos, es decir, un entero entre 0 y 59, a partir de *número_hora*.

@SEN(z)

Calcula el seno del ángulo z .

@SENH(z)

Calcula el seno hiperbólico del ángulo z .

@SI(*condición*;x;y)

Evalúa *condición* y da como resultado x si *condición* es verdadera o y si *condición* es falsa.

@SIGNO(x)

Da como resultado 1 si x es un valor positivo, 0 si x es igual a 0 y -1 si x es un valor negativo.

@SLN(*coste*;valor_residual;vida)

Calcula la cuota de amortización lineal de un bien de inversión con un *coste* inicial, una *vida* útil estimada y un *valor_residual* dados para cada uno de los períodos en que se mide su vida útil.

@STD(*lista*)

Calcula la desviación típica de una *lista* de valores que constituyen una población.

@STDM(*lista*)

Calcula la desviación típica de una *lista* de valores que constituyen una muestra.

@STDMPURA(*lista*)

Calcula la desviación típica de una *lista* de valores que constituyen una muestra, pero ignora las celdas que contienen rótulos.

@STDPURA(*lista*)

Calcula la desviación típica de una *lista* de valores que constituyen una población, pero ignora las celdas que contengan rótulos.

@SUBTOTAL(*lista*)

Suma los valores contenidos en *lista*. @SUBTOTAL se utiliza para indicar a @TOTAL las celdas que debe sumar.

@SUMA(*lista*)

Suma los valores contenidos en *lista*.

@SUMACUADRADOS(*lista*)

Calcula la suma de los cuadrados de los valores de *lista*.

@SUMADIF2(*rango1*;rango2)

Resta de las celdas correspondientes del *rango1* los valores de las celdas del *rango2*, calcula los cuadrados de las diferencias y, por último, suma los cuadrados.

@SUMAN(*número_offset*;n;*lista*)

Suma todos los valores *enésimos* de *lista*, a partir de *número_offset*.

@SUMANEGATIVOS(*lista*)

Suma sólo los valores negativos de *lista*.

@SUMAPOSITIVOS(*lista*)

Suma sólo los valores positivos de *lista*.

@SUMAPRODUCTOS(*lista*)

Multiplica los valores contenidos en las celdas homólogas de varios rangos de *lista* y, a continuación, suma los productos.

@SUMASERIE(*x;n;m;coeficientes*)

Calcula la suma de una serie de potencias.

@SUPRESP(*cadena*)

Elimina los espacios iniciales y finales y reduce a uno los espacios múltiples situados entre dos palabras de *cadena*.

@SUSTITUIR(*cadena_original;posición_inicial;n;cadena_nueva*)

Sustituye *n* caracteres de *cadena_original* por *cadena_nueva*, a partir de *posición_inicial*.

@SUSTITUIRB(*cadena_original;posición_inicial;n;cadena_nueva*)

Sustituye *n* bytes de *cadena_original* por *cadena_nueva*, a partir de *posición_inicial*.

@SYD(*coste;valor_residual;vida;periodo*)

Calcula, por el método de los números dígitos, la cuota de amortización de un bien de inversión para un *periodo* determinado.

T

@TAN(*z*)

Calcula la tangente del ángulo *z*.

@TANH(*z*)

Calcula la tangente hiperbólica del ángulo *z*.

@TASA(*valor_futuro;valor_actual;plazo*)

Calcula el tipo de interés periódico necesario para que una inversión alcance un *valor_futuro* al cabo del número de periodos de capitalización especificados en *plazo*.

@TASAINT(*plazo;pagos;valor_actual;[tipo;valor_futuro;estimación]*)

Calcula el tipo (o tasa) de interés compuesto necesario para que una inversión (*valor_actual*) alcance un *valor_futuro* durante un determinado número de periodos de pago (*plazo*).

@TBUSCAR(*cadena_búsqueda;cadena;posición_inicial*)

Calcula la posición de un carácter tailandés lógico en *cadena*; 1-2-3 determinará cuándo aparece por primera vez *cadena_búsqueda* en *cadena* a partir del carácter tailandés que se encuentre en la columna a la que se refiere *posición_inicial*. Se utiliza con cadenas de caracteres tailandeses.

@TCADENAFECHA(*valor_fecha*)

Convierte *valor_fecha* en una cadena de fecha tailandesa con el formato corto.

@TCADENAFECHAL(*valor_fecha*)

Convierte *valor_fecha* en una cadena de fecha tailandesa con formato largo.

@TCIFRA(*cadena_cifra*)

Convierte los números arábigos de *cadena_cifra* en una cadena con caracteres numéricos tailandeses.

@TDDS(*valor_fecha*)

Convierte *valor_fecha* en el día de la semana en tailandés.

@TDERECHA(*cadena;n*)

Arroja como resultado los últimos *n* caracteres tailandeses lógicos de *cadena*. Se utiliza con cadenas de caracteres tailandeses.

@TESTF(*rango1;rango2*)

Realiza un test o verificación de hipótesis mediante el parámetro estadístico *F* y da como resultado la probabilidad asociada.

@TESTJI(*rango1;[rango2]*)

Realiza un test o verificación de independencia mediante el parámetro estadístico *ji* cuadrado con los datos contenidos en *rango1*, o una verificación de bondad de ajuste mediante el parámetro estadístico *ji* cuadrado con los datos contenidos en *rango1* y en *rango2*.

@TESTI(*rango1;rango2:[tipo;colas]*)

Realiza un test o verificación de hipótesis mediante el parámetro estadístico *t* de Student con los datos contenidos en *rango1* y *rango2*, y da como resultado la probabilidad asociada.

@TESTZ(*rango1;media1;std1;[colas;rango2;media2;std2]*)

Realiza una verificación mediante el parámetro estadístico *z* con una o dos poblaciones y da como resultado la probabilidad asociada.

@TIR(*estimación;rango*)

Calcula la tasa interna de rendimiento (beneficio) prevista para una serie de flujos de caja generados por una inversión.

@TIRM(*rango;tipo_financiación;tipo_reinversión;[tipo]*)

Calcula la tasa interna de rendimiento modificada (beneficio) prevista para una serie de flujos de caja generados por una inversión.

@TIRX(*estimación;flujos_caja;fechas*)

Da como resultado la tasa interna de rendimiento prevista para una serie de flujos de caja de entrada y de salida.

@TIZDA(*cadena;n*)

Arroja como resultado los *n* primeros caracteres tailandeses lógicos de *cadena*. Se utiliza con cadenas de caracteres tailandeses.

@TLONGITUD(*cadena*)

Cuenta el número de caracteres tailandeses lógicos contenidos en *cadena*. Se utiliza con cadenas de caracteres tailandeses.

@TMED(*cadena;posición_inicial;n*)

Copia *n* caracteres tailandeses lógicos de *cadena* a partir del carácter tailandés que se encuentre en *posición_inicial*. Se utiliza con cadenas de caracteres tailandeses.

@TNUMEROCADENA(*número*)

Convierte *número* en una cadena de número tailandés deletreado.

@TOTAL(*lista*)

Calcula la suma de todas las celdas de *lista* que contienen @SUBTOTAL en sus fórmulas.

@TRUNC(*x;[n]*)

Trunca el valor *x* a partir de la posición especificada mediante *n*.

@TSUSTITUIR(*cadena_original;posición_inicial;n;cadena_nueva*)

Sustituye *n* caracteres tailandeses lógicos de *cadena original* por el contenido de *nueva_cadena*, a partir del carácter indicada por *posición_inicial*. Se utiliza con cadenas de caracteres tailandeses.

U

@ULTIMOESCENARIO(*nombre_archivo*)

Da como resultado el nombre del grupo de versiones que se ha visualizado por última vez en un libro durante una sesión de 1-2-3.

V

@VALACT(*pagos;interés;plazo*)

Calcula el valor actual de una inversión constituida por una serie de *pagos* iguales descontados a un tipo de *interés* dado durante el número de períodos indicado en *plazo*, suponiendo una anualidad ordinaria.

@VALACT2(*pagos;interés;plazo*)

Calcula el valor actual de una inversión constituida por una serie de *pagos* iguales por los que se retribuye una tasa de *interés* dada, durante el número de períodos indicado en *plazo* y suponiendo una anualidad anticipada.

@VALACTGLOBAL(*valor_futuro;interés;plazo;[frecuencia]*)

Da como resultado el valor actual de una suma global que se percibirá dentro de un determinado número de períodos en el futuro (*plazo*) y que se descontará o actualizará a un tipo de *interés* dado.

@VALACTUAL(*pagos;interés;plazo;[tipo;valor_futuro]*)

Calcula el valor actual de una inversión que tiene un determinado *valor_futuro*, para una anualidad ordinaria o anticipada.

@VALFECHA(*cadena*)

Calcula el número de fecha correspondiente a la fecha especificada en *cadena*.

@VALFUT(*pagos;interés;plazo*)

Calcula el valor futuro de una inversión constituida por una serie de *pagos* iguales por los que se retribuye una

tasa de *interés* dada, durante el número de períodos indicado en *plazo*, y suponiendo una anualidad ordinaria.

@VALFUT2(pagos;interés;plazo)

Calcula el valor futuro de una inversión constituida por una serie de *pagos* iguales por los que se retribuye una tasa de *interés* dada, durante el número de períodos indicado en *plazo* y suponiendo una anualidad anticipada.

@VALFUTCANTIDAD(principal;interés;plazo;[frecuencia])

Da como resultado el valor futuro de una suma global invertida a un tipo de interés determinado durante un número dado de períodos.

@VALFUTURO(pagos;interés;plazo;[tipo;valor_ actual])

Calcula el valor futuro de una inversión que tiene un determinado *valor_ actual* , realizando el cálculo para anualidades ordinarias o anticipadas.

@VALHORA(cadena)

Calcula el número de hora correspondiente a la hora especificada en *cadena*.

@VALOR(cadena)

Convierte un número especificado como *cadena* en su valor correspondiente.

@VAN(interés;rango;[tipo])

Calcula el valor actual neto de una serie de flujos de caja futuros (contenidos en *rango*) actualizados a un tipo de *interés* periódico fijo.

@VANX(tasa;flujos_caja;fechas)

Da como resultado el valor actual neto de una serie de flujos de caja de entrada y de salida.

@VAR(lista)

Calcula la varianza de una *lista* de valores que constituyen una población.

@VARM(lista)

Calcula la varianza de una *lista* de valores que constituyen una muestra.

@VARMPURA(lista)

Calcula la varianza de una *lista* de valores que constituyen una muestra, pero ignora las celdas que contienen rótulos.

@VARPURA(lista)

Calcula la varianza de una *lista* de valores que constituyen una población, pero ignora las celdas que contienen rótulos.

@VDB(coste;valor_residual;vida;período_inicial;período_final;[factor_amort;cambio])

Calcula la cuota de amortización correspondiente a un determinado período para un bien de inversión, utilizando el método de los saldos decrecientes al 200%, con posibilidad de otras tasas de amortización distintas del 200%.

@VERDAD

Da como resultado el valor lógico 1 (verdad).

@VERSIONACTUAL(rango)

Da como resultado el nombre de la versión actual de un *rango*.

@VERSIONDATOS(opción;celda;rango_versión;nombre;[autor])

Da como resultado el contenido de una determinada celda de una versión.

@VHORA(hora;minutos;segundos)

Calcula el número de hora correspondiente a unos valores de *hora*, *minutos* y *segundos* específicos.

@VINCULODATOS(aplicación;archivo;elemento;[formato;máx_filas;máx_cols;máx_hojas])

Establece un vínculo OLE entre los datos.

W

X

Y

Z

