Funciones de conversión

Asc: Devuelve un tipo de datos Integer que representa el código de carácter correspondiente a la primera letra de una cadena.

Sintaxis: Asc(cadena)

cadena Cualquier expresión de cadena válida.

CBool: Convierte una expresión en tipo de datos Boolean.

Sintaxis: CBool(expr)

expr Cualquier cadena o expresión numérica válida.

CByte: Convierte una expresión en tipo de datos Byte.

Sintaxis: CByte(expr)

expr Cualquier valor Integer entre 0 y 255.

CCur: Convierte una expresión en tipo de datos Currency.

Sintaxis: CCur(expr)

expr Cualquier expresión numérica entre –922.337.203.685.477,5808 y 922.337.203.685.477,5807.

CDate: Convierte una expresión en tipo de datos Date.

Sintaxis: CDate(expr)

expr Cualquier expresión de fecha válida.

CDbl: Convierte una expresión en tipo de datos Double.

Sintaxis: CDbl(expr)

expr Cualquier expresión numérica entre –1,79769313486232E308 y –4,94065645841247E–324 para valores negativos o entre 4,94065645841247E–324 y 1,79769313486232E308 para valores positivos.

CDec: Convierte una expresión en tipo de datos Decimal.

Sintaxis: CDec(expr)

expr Cualquier expresión numérica a partir de +/ –79.228.162.514,264.337.593.543.950.335 para números en escala de cero (es decir, sin decimales). Para números con 28 decimales, el intervalo es +/ –7,9228162514264337593543950335. El número distinto de cero más pequeño posible es 0,0000000000000000000000000001.

Chr: Devuelve un tipo de datos String que contiene el carácter asociado con el código de carácter especificado.

Sintaxis: Chr(códcar)

códcar Un tipo de datos Long que identifica un carácter.

CInt: Convierte una expresión en tipo de datos Integer.

Sintaxis: CInt(expr)

Expression Cualquier expresión numérica entre–32.768 y 32.767; se redondean las fracciones.

CLng: Convierte una expresión en tipo de datos Long.

Sintaxis: CLng(expr)

Expression Cualquier expresión numérica entre–2.147.483.648 y 2.147.483.647; se redondean las fracciones.

CSng: Convierte una expresión en tipo de datos Single.

Sintaxis: CSng(expr)

expr Cualquier expresión numérica entre –3,402823E38 y –1,401298E–45 para valores negativos y entre 1,401298E–45 y 3,402823E38 para valores positivos.

CStr: Convierte una expresión en tipo de datos String.

Sintaxis: CStr(expr)

expr Cualquier cadena o expresión numérica válida.

CVar: Convierte una expresión en tipo de datos Variant.

Sintaxis: CVar(expr)

Mismo intervalo que Double para los valores numéricos. Mismo intervalo que String para los valores no numéricos.

DateSerial: Devuelve un tipo de dato Variant (Date) para un año, mes y día especificados.

Sintaxis: DateSerial(año, mes, día)

año Obligatorio; Integer. Número entre 100 y 9999 (ambos inclusive) o una expresión numérica.

mes Obligatorio; Integer. Cualquier expresión numérica.

día Obligatorio; Integer. Cualquier expresión numérica.

DateValue: Devuelve un tipo de dato Variant (Date).

Sintaxis: DateValue(fecha)

fecha Obligatorio; normalmente una expresión de cadena que representa una fecha entre el 1 de enero de 100 y el 31 de diciembre de 9999. Sin embargo, fecha también puede ser cualquier expresión que pueda representar una fecha, una hora o una fecha y una hora, en ese intervalo.

Day: Devuelve un tipo de dato Variant (Integer) que especifica un número completo entre 1 y 31, ambos inclusive, que representa el día del mes.

Sintaxis: Day(fecha)

fecha Obligatorio; cualquier tipo de datos Variant, expresión numérica, expresión de cadena o combinación de ellas que pueda representar una fecha. Si fecha contiene un valor Null, se devolverá Null.

Hex: Devuelve un tipo de datos String que representa el valor hexadecimal de un número.

Sintaxis: Hex(número)

número Obligatorio; cualquier cadena o expresión numérica válida.

Hour: Devuelve un tipo de datos Variant (Integer) que especifica un número entero entre 0 y 23 (ambos inclusive) y representa la hora del día.

Sintaxis: Hour(hora)

hora Obligatorio; cualquier tipo de datos Variant, expresión numérica, expresión de cadena o combinación de ellas que pueda representar una hora. Si hora contiene un valor Null, se devolverá Null.

Minute: Devuelve un valor de tipo Variant (Integer) que especifica un número entero entre 0 y 59, ambos inclusive, que representa el minuto de la hora.

Sintaxis: Minute(hora)

hora Obligatorio; cualquier tipo de datos Variant, expresión numérica, expresión de cadena o combinación de ellas que pueda representar una hora. Si hora contiene un valor Null, se devolverá Null.

Month: Devuelve un valor de tipo Variant (Integer) que especifica un número entero entre 1 y 12, ambos inclusive, y representa el mes del año.

Sintaxis: Month(fecha)

fecha Obligatorio; cualquier tipo de datos Variant, expresión numérica, expresión de cadena o combinación de ellas que pueda representar una hora. Si fecha contiene un valor Null, se devolverá Null.

Oct: Devuelve un valor de tipo Variant (String) que representa el valor octal de un número.

Sintaxis: Oct(número)

número Obligatorio; cualquier cadena o expresión numérica válida.

ProjDateConv: Convierte un valor en una fecha.

Sintaxis: ProjDateConv(expr, formato_fecha)

expresión Obligatorio; Variant. La expresión que se va a convertir en una fecha.

formato_fecha Opcional; Long. El formato de fecha predeterminado es pjDateDefault, pero puede sustituir una de las siguientes constantes pjDateFormat (formato de fecha aplicado 25/9/07 a las 12:33 p. m.):

pjDateDefault: el formato predeterminado. Se establece en la pestaña Vista del cuadro de diálogo Opciones (menú Herramientas).

pjDate_mm_dd_aa_hh_mmAM: 25/9/07 12:33 p. m.

pjDate_mm_dd_aa: 9/25/07

pjDate_mm_dd_aaaa: 9/25/2007

pjDate_mmmm_dd_aaaa_hh_mmAM: 25 de septiembre de 2007, 12:33 p. m.

pjDate_mmmm_dd_aaaa: 25 de septiembre de 2007

pjDate_mmm_dd_hh_mmAM: 25 sep, 12:33 p. m.

pjDate_mmm_dd_aaa: 25 sep ’07

pjDate_mmmm_dd: 25 de septiembre

pjDate_mmm_dd: 25 sep

pjDate_ddd_mm_dd_aa_hh_mmAM: Mar 25/9/07, 12:33 p. m.

pjDate_ddd_mm_dd_aa: Mar 25/9/07

pjDate_ddd_mmm_dd_aaa: Mar 25 sep '07

pjDate_ddd_hh_mmAM: Mar 12:33 p. m.

pjDate_mm_dd: 25/9

pjDate_dd: 25

pjDate_hh_mmAM: 12:33 p. m.

pjDate_ddd_mmm_dd: Mar 25 sep

pjDate_ddd_mm_dd: Mar 25/9

pjDate_ddd_dd: Mar 25

pjDate_Www_dd: S40/2

pjDate_Www_dd_aa_hh_mmAM: S40/2/07, 12:33 p. m.

ProjDurConv: Convierte una expresión en un valor de duración en las unidades especificadas.

Sintaxis: ProjDurConv(expresión, unidades_duración)

expresión Obligatorio; Variant. La expresión que se va a convertir en duración.

unidades_duración Opcional; Long. Las unidades usadas para expresar la duración. Si no se especifica un valor para unidades_duración, el valor predeterminado será el tipo de unidades especificado en la opción Mostrar duración en de la pestaña Programación, en el cuadro de diálogo Opciones del menú Herramientas. El valor de unidades_duración puede ser una de las siguientes constantes de pjFormatUnit:

pjMinutes: pjElapsedMinutes

pjHours: pjElapsedHours

pjDays: pjElapsedDays

pjWeeks: pjElapsedWeeks

pjMonths: pjElapsedMonths

pjMinutesEstimated: pjElapsedMinutesEstimated

pjHoursEstimated: pjElapsedHoursEstimated

pjDaysEstimated: pjElapsedDaysEstimated

pjWeeksEstimated: pjElapsedWeeksEstimated

pjMonthsEstimated: pjElapsedMonthsEstimated

Second: Devuelve un valor de tipo Variant (Integer) que especifica un número entero entre 0 y 59, ambos inclusive, que representa el segundo del minuto.

Sintaxis: Second(hora)

hora Obligatorio; cualquier tipo de datos Variant, expresión numérica, expresión de cadena o combinación de ellas que pueda representar una hora. Si hora contiene un valor Null, se devolverá Null.

Str: Devuelve un valor de tipo Variant (String) que representa un número.

Sintaxis: Str(número)

número Obligatorio; tipo de datos Long que contiene cualquier expresión numérica válida.

StrConv: Devuelve un valor de tipo Variant (String) que se ha convertido según lo especificado.

Sintaxis: StrConv(cadena, conversión, código_idioma)

cadena Obligatorio; expresión de cadena que se va a convertir.

conversión Obligatorio; Entero. La suma de valores que especifican el tipo de conversión que se va a realizar.

código_idioma Opcional; el identificador de configuración regional, si es diferente del identificador de configuración regional del sistema. (El del sistema es el predeterminado.)

TimeSerial: Devuelve un tipo de datos Variant (Date) que contiene la hora para una hora, minuto y segundo concretos.

Sintaxis: TimeSerial(hora, minuto, segundo)

hour Obligatorio; Variant (Integer). Número entre 0 (12:00 A.M.) y 23 (11:00 P.M.), ambos incluidos o una expresión numérica.

minuto Obligatorio; Variant (Integer). Cualquier expresión numérica.

segundo Obligatorio; Variant (Integer). Cualquier expresión numérica.

TimeValue: Devuelve un tipo de datos Variant (Date) que contiene la hora.

Sintaxis: TimeValue(hora)

agota Necesario Normalmente, una expresión de cadena que representa una hora de 0:00:00 (12:00:00 A.M.) a 23:59:59 (11:59:59), ambos inclusive. Sin embargo, Time también puede ser cualquier expresión que representa una hora de ese intervalo. Si hora contiene un valor Null, se devolverá Null.

Val: Devuelve los números incluidos en una cadena como un valor numérico del tipo apropiado.

Sintaxis: Val(cadena)

cadena Obligatorio; cualquier expresión de cadena válida.

Weekday: Devuelve un valor de tipo Variant (Integer) que contiene un número entero que representa el día de la semana.

Sintaxis: Weekday(fecha[, primer_día_semana])

fecha Obligatorio; Variant, expresión numérica, expresión de cadena o cualquier combinación de ellas que pueda representar una fecha. Si fecha contiene un valor Null, se devolverá Null.

primer_día_semana Opcional; una constante que especifica el primer día de la semana. Si no se especifica ningún valor, se supone que es el domingo.

Year: Devuelve un valor de tipo Variant (Integer) que contiene un número entero que representa el año.

Sintaxis: Year(fecha)

fecha Obligatorio; cualquier tipo de datos Variant, expresión numérica, expresión de cadena o combinación de ellas que pueda representar una fecha. Si fecha contiene un valor Null, se devolverá Null.

Funciones de fecha y hora

CDate: Convierte una expresión en tipo de datos Date.

Sintaxis: CDate(expr)

expr Cualquier expresión de fecha válida.

Date: Devuelve un tipo de dato Variant (Date) que contiene la fecha actual de sistema.

Sintaxis: Date

DateAdd: Devuelve un tipo de dato Variant (Date) que contiene una fecha a la que se ha agregado un intervalo de tiempo especificado.

Sintaxis: DateAdd(intervalo, número, fecha)

intervalo Obligatorio; expresión de cadena que es el intervalo de tiempo que se desea agregar, como "m" o "d" .

número Obligatorio; expresión numérica que es el número de intervalos que se desea agregar. Puede ser positivo (para obtener fechas en el futuro) o negativo (para obtener fechas en el pasado).

fecha Obligatorio; Variant (Date) o texto literal que representa la fecha a la que se agrega el intervalo.

DateDiff: Devuelve un tipo de datos Variant (Long) que especifica el número de intervalos de tiempo entre las dos fechas especificadas.

Sintaxis: DateDiff(intervalo, fecha1, fecha2[, primer_día_semana[, primera_semana_año]])

intervalo Obligatorio; expresión de cadena que es el intervalo de tiempo empleado para calcular la diferencia entre fecha1 y fecha2.

fecha1, fecha2 Obligatorio; Variant (Date). Dos fechas que se desea utilizar en el cálculo.

primer_día_semana Opcional; una constante que especifica el primer día de la semana. Si no se especifica ningún valor, se supone que es el domingo.

primera_semana_año Opcional; una constante que especifica la primera semana del año. Si no se especifica, se entiende que la primera semana es aquella que contiene el día 1 de enero.

DatePart: Devuelve un tipo de dato Variant (Integer) que contiene la parte especificada de una fecha dada.

Sintaxis: DatePart(intervalo, fecha[, primer_día_semana[, primera_semana_año]])

intervalo Obligatorio; expresión de cadena que es el intervalo de tiempo que se desea devolver.

fecha Obligatorio; valor Variant (Date) que se desea evaluar.

primer_día_semana Opcional; una constante que especifica el primer día de la semana. Si no se especifica ningún valor, se supone que es el domingo.

primera_semana_año Opcional; una constante que especifica la primera semana del año. Si no se especifica, se entiende que la primera semana es aquella que contiene el día 1 de enero.

DateSerial: Devuelve un tipo de dato Variant (Date) para un año, mes y día especificados.

Sintaxis: DateSerial(año, mes, día)

año Obligatorio; Integer. Número entre 100 y 9999 (ambos inclusive) o una expresión numérica.

mes Obligatorio; Integer. Cualquier expresión numérica.

día Obligatorio; Integer. Cualquier expresión numérica.

DateValue: Devuelve un tipo de dato Variant (Date).

Sintaxis: DateValue(fecha)

fecha Obligatorio; normalmente una expresión de cadena que representa una fecha entre el 1 de enero de 100 y el 31 de diciembre de 9999. Sin embargo, fecha también puede ser cualquier expresión que pueda representar una fecha, una hora o una fecha y una hora, en ese intervalo.

Day: Devuelve un tipo de dato Variant (Integer) que especifica un número completo entre 1 y 31, ambos inclusive, que representa el día del mes.

Sintaxis: Day(fecha)

fecha Obligatorio; cualquier tipo de datos Variant, expresión numérica, expresión de cadena o combinación de ellas que pueda representar una fecha. Si fecha contiene un valor Null, se devolverá Null.

Hour: Devuelve un tipo de datos Variant (Integer) que especifica un número entero entre 0 y 23 (ambos inclusive) y representa la hora del día.

Sintaxis: Hour(hora)

hora Obligatorio; cualquier tipo de datos Variant, expresión numérica, expresión de cadena o combinación de ellas que pueda representar una hora. Si hora contiene un valor Null, se devolverá Null.

IsDate: Devuelve un valor de tipo Boolean que indica si una expresión puede convertirse en una fecha.

Sintaxis: IsDate(expr)

expr Obligatorio; cualquier tipo de datos Variant que contenga una expresión de fecha o de cadena reconocible como una fecha o una hora.

Minute: Devuelve un valor de tipo Variant (Integer) que especifica un número entero entre 0 y 59, ambos inclusive, que representa el minuto de la hora.

Sintaxis: Minute(hora)

hora Obligatorio; cualquier tipo de datos Variant, expresión numérica, expresión de cadena o combinación de ellas que pueda representar una hora. Si hora contiene un valor Null, se devolverá Null.

Month: Devuelve un valor de tipo Variant (Integer) que especifica un número entero entre 1 y 12, ambos inclusive, y representa el mes del año.

Sintaxis: Month(fecha)

fecha Obligatorio; cualquier tipo de datos Variant, expresión numérica, expresión de cadena o combinación de ellas que pueda representar una hora. Si fecha contiene un valor Null, se devolverá Null.

Now: Devuelve un valor de tipo Variant (Date) que especifica la fecha y la hora actuales conforme a la fecha y la hora del sistema del equipo.

Sintaxis: Now

ProjDateAdd: Agrega una duración a una fecha y devuelve una nueva fecha.

Sintaxis: ProjDateAdd(fecha, duración, calendario)

fecha Obligatorio; Variant. La fecha original a la que se agrega la duración.

duración Obligatorio; Variant. La duración que se agrega a una fecha.

calendario Opcional; String. El calendario que se utiliza al calcular la nueva fecha. Si no se especifica calendario, el valor predeterminado para el recurso actual será el calendario del recurso, y para la tarea actual, el calendario de la tarea o el calendario estándar si no hay ningún calendario. En Project Server se utilizará el calendario estándar independientemente de qué calendario se especifique en la cadena calendario. Si utiliza esta función en una fórmula creada en Project Web App y la compara con la misma fórmula creada en Project Profesional, deberá hacer pruebas para comprobar si obtiene los resultados esperados.

Nota: Si se quiere restar siete días a una fecha especificada, la fórmula siguiente funciona correctamente en Project Profesional 2010: ProjDateAdd("24/9/2010", "–7d"). Pero, si ejecuta la misma fórmula en Project Server 2010, el resultado será 24/9/2010, no 17/9/2010. Para crear fórmulas que funcionen correctamente en Project Profesional 2010 y Project Server 2010 tiene que evitar aplicar parámetros negativos en las funciones ProjDateAdd y ProjDateSub.

ProjDateConv: Convierte un valor en una fecha.

Sintaxis: ProjDateConv(expr, formato_fecha)

expresión Obligatorio; Variant. La expresión que se va a convertir en una fecha.

formato_fecha Opcional; Long. El formato de fecha predeterminado es pjDateDefault, pero puede sustituir una de las siguientes constantes pjDateFormat (formato de fecha aplicado 25/9/07 a las 12:33 p. m.):

pjDateDefault: el formato predeterminado. Se establece en la pestaña Vista del cuadro de diálogo Opciones (menú Herramientas).

pjDate_mm_dd_aa_hh_mmAM: 25/9/07 12:33 p. m.

pjDate_mm_dd_aa: 9/25/07

pjDate_mm_dd_aaaa: 9/25/2007

pjDate_mmmm_dd_aaaa_hh_mmAM: 25 de septiembre de 2007, 12:33 p. m.

pjDate_mmmm_dd_aaaa: 25 de septiembre de 2007

pjDate_mmm_dd_hh_mmAM: 25 sep, 12:33 p. m.

pjDate_mmm_dd_aaa: 25 sep '07

pjDate_mmmm_dd: 25 de septiembre

pjDate_mmm_dd: 25 sep

pjDate_ddd_mm_dd_aa_hh_mmAM: Mar 25/9/07, 12:33 p. m.

pjDate_ddd_mm_dd_aa: Mar 25.09.03

pjDate_ddd_mmm_dd_aaa: Mar 25 sep '07

pjDate_ddd_hh_mmAM: Mar 12:33 p. m.

pjDate_mm_dd: 25/9

pjDate_dd: 25

pjDate_hh_mmAM: 12:33 p. m.

pjDate_ddd_mmm_dd: Mar 25 sep

pjDate_ddd_mm_dd: Mar 25/9

pjDate_ddd_dd: Mar 25

pjDate_Www_dd: S40/2

pjDate_Www_dd_aa_hh_mmAM: S40/2/07, 12:33 p. m.

ProjDateDiff: Devuelve la duración entre dos fechas en minutos.

Sintaxis: ProjDateDiff(fecha1, fecha2, calendario)

fecha1 Obligatorio; Variant. La fecha que se utiliza como comienzo de la duración.

fecha2 Obligatorio; Variant. La fecha que se utiliza como final de la duración.

calendario Opcional; Cadena. El calendario que se va a usar al calcular la duración. Si no se especifica el calendario , el valor predeterminado para el recurso actual es el calendario de recursos, o bien, para la tarea actual, el calendario de tareas (o el calendario estándar si no hay calendario de tareas para Project Server, se usará el calendario estándar, independientemente del calendario que se especifique en la cadena de calendario . Si utiliza esta función en una fórmula creada en Project Web App y la compara con la misma fórmula creada en

Project Professional, deberá hacer pruebas para comprobar si obtiene los resultados esperados.

ProjDateSub: Devuelve la fecha que precede a otra fecha con una duración especificada.

Sintaxis: ProjDateSub(fecha, duración, calendario)

fecha Obligatorio; Variant. La fecha original de la que se resta la duración.

duración Obligatorio; Variant. La duración que se resta de la fecha.

calendario Opcional; Cadena. El calendario que se utiliza al calcular la diferencia entre fechas. Si no se especifica calendario, el valor predeterminado para el recurso actual será el calendario del recurso y, para la tarea actual, el calendario de la tarea (o el calendario estándar si no hay ningún calendario de tareas). En Project Server se utilizará el calendario estándar independientemente de qué calendario se especifique en la cadena calendario. Si utiliza esta función en una fórmula creada en Project Web App y la compara con la misma fórmula creada en Project Professional, deberá hacer pruebas para comprobar si obtiene los resultados esperados.

Nota: Si se quiere restar siete días a una fecha especificada, la fórmula siguiente funciona correctamente en Project Profesional 2010: ProjDateAdd("24/9/2010", "–7d"). Pero, si ejecuta la misma fórmula en Project Server 2010, el resultado será 24/9/2010, no 17/9/2010. Para crear fórmulas que funcionen correctamente en Project Profesional 2010 y Project Server 2010 tiene que evitar aplicar parámetros negativos en las funciones ProjDateAdd y ProjDateSub.

ProjDateValue: Devuelve el valor de fecha de una expresión.

Sintaxis: ProjDateValue(expr)

expr Obligatorio; Variant. La expresión que se representará como fecha.

ProjDurConv: Convierte una expresión en un valor de duración en las unidades especificadas.

Sintaxis: ProjDurConv(expresión, unidades_duración)

expresión Obligatorio; Variant. La expresión que se va a convertir en duración.

unidades_duración Opcional; Long. Las unidades usadas para expresar la duración. Si no se especifica un valor para unidades_duración, el valor predeterminado será el tipo de unidades especificado en la opción Mostrar duración en de la pestaña Programación, en el cuadro de diálogo Opciones del menú Herramientas. El valor de unidades_duración puede ser una de las siguientes constantes de pjFormatUnit:

pjMinutes: pjElapsedMinutes

pjHours: pjElapsedHours

pjDays: pjElapsedDays

pjWeeks: pjElapsedWeeks

pjMonths: pjElapsedMonths

pjMinutesEstimated: pjElapsedMinutesEstimated

pjHoursEstimated: pjElapsedHoursEstimated

pjDaysEstimated: pjElapsedDaysEstimated

pjWeeksEstimated: pjElapsedWeeksEstimated

pjMonthsEstimated: pjElapsedMonthsEstimated

ProjDurValue: Devuelve el número de minutos de una duración.

Sintaxis: ProjDurValue(duración)

duración Obligatorio; Variant. La duración que se expresará en minutos.

Second: Devuelve un valor de tipo Variant (Integer) que especifica un número entero entre 0 y 59, ambos inclusive, que representa el segundo del minuto.

Sintaxis: Second(hora)

hora Obligatorio; cualquier tipo de datos Variant, expresión numérica, expresión de cadena o combinación de ellas que pueda representar una hora. Si hora contiene un valor Null, se devolverá Null.

Time: Devuelve un valor de tipo Variant (Fecha) que indica la hora actual del sistema.

Sintaxis: Time

Timer: Devuelve un valor de tipo Single que representa el número de segundos transcurridos desde la medianoche.

Sintaxis: Cronómetro

TimeSerial: Devuelve un tipo de datos Variant (Date) que contiene la hora para una hora, minuto y segundo concretos.

Sintaxis: TimeSerial(hora, minuto, segundo)

hour Obligatorio; Variant (Integer). Número entre 0 (12:00 A.M.) y 23 (11:00 P.M.), ambos incluidos o una expresión numérica.

minuto Obligatorio; Variant (Integer). Cualquier expresión numérica.

segundo Obligatorio; Variant (Integer). Cualquier expresión numérica.

TimeValue: Devuelve un tipo de datos Variant (Date) que contiene la hora.

Sintaxis: TimeValue(hora)

hora Necesario Normalmente, una expresión de cadena que representa una hora de 0:00:00 (12:00:00 A.M.) a 23:59:59 (11:59:59), ambos inclusive. Sin embargo, Time también puede ser cualquier expresión que representa una hora de ese intervalo. Si hora contiene un valor Null, se devolverá Null.

Weekday: Devuelve un valor de tipo Variant (Integer) que contiene un número entero que representa el día de la semana.

Sintaxis: Weekday(fecha[, primer_día_semana])

fecha Obligatorio; Variant, expresión numérica, expresión de cadena o cualquier combinación de ellas que pueda representar una fecha. Si fecha contiene un valor Null, se devolverá Null.

primer_día_semana Opcional; una constante que especifica el primer día de la semana. Si no se especifica ningún valor, se supone que es el domingo.

Year: Devuelve un valor de tipo Variant (Integer) que contiene un número entero que representa el año.

Sintaxis: Year(fecha)

fecha Obligatorio; cualquier tipo de datos Variant, expresión numérica, expresión de cadena o combinación de ellas que pueda representar una fecha. Si fecha contiene un valor Null, se devolverá Null.

Funciones generales

Choose: Selecciona y devuelve un valor de una lista de argumentos.

Sintaxis: Choose(índice, opción–1[, opción–2, ... [, opción-n]])

índice Obligatorio; campo o expresión numérica que da como resultado un valor entre 1 y el número de opciones disponibles.

opción Obligatorio; expresión de tipo Variant que contiene una de las posibles opciones.

IIf: Devuelve una de dos partes dependiendo de la evaluación de una expresión.

Sintaxis: IIf(expr, parte_verdadera, parte_falsa)

expr Obligatorio; expresión tipo Variant que se desea evaluar.

parte_verdadera Obligatorio; valor o expresión que se devolverá si la expresión correspondiente tiene un valor True.

parte_falsa Obligatorio; valor o expresión que se devolverá si la expresión correspondiente tiene un valor False.

IsNumeric: Devuelve un valor Boolean que indica si una expresión puede evaluarse como número.

Sintaxis: IsNumeric(expr)

expresión Obligatorio; Variant que contiene una expresión numérica o de cadena.

IsNull: Devuelve un valor Boolean que indica si una expresión no contiene datos válidos (Null).

Sintaxis: IsNull(expresión)

expresión Obligatorio; Variant que contiene una expresión numérica o de cadena.

Switch: Evalúa una lista de expresiones y devuelve un tipo de datos Variant o una expresión asociada con la primera expresión de la lista que tenga el valor True.

Sintaxis: Switch(expr–1, valor–1[, expr–2, valor–2, … [, expr-n,valor-n]])

Funciones matemáticas

Abs: Devuelve un valor del mismo tipo que se le haya pasado y que especifica el valor absoluto de un número.

Sintaxis: Abs(número)

número Obligatorio; cualquier expresión numérica válida. Si número contiene Null, se devolverá Null; si es una variable no inicializada, se devolverá el valor cero.

Atn: Devuelve un valor Double que especifica el arco tangente de un número.

Sintaxis: Atn(número)

número Obligatorio; tipo de datos Double o cualquier expresión numérica válida.

Cos: Devuelve un tipo de dato Double que especifica el coseno de un ángulo.

Sintaxis: Cos(número)

número Obligatorio; Double o cualquier expresión numérica válida que exprese un ángulo en radianes.

Exp: Devuelve un tipo de dato Double que especifica el número e (base de los logaritmos naturales) elevado a una potencia.

Sintaxis: Exp(número)

número Obligatorio; tipo de datos Double o cualquier expresión numérica válida.

Fix: Devuelve la parte entera de un número. Si el número es negativo, devuelve el primer entero negativo igual o mayor a número.

Sintaxis: Fix(número)

número Obligatorio; tipo de dato Double o cualquier expresión numérica válida. Si número contiene Null, se devolverá Null.

Int: Devuelve la parte entera de un número. Si número es negativo, devuelve el primer entero negativo menor o igual a número.

Sintaxis: Int(número)

número Obligatorio; tipo de datos Double o cualquier expresión numérica válida. Si número contiene Null, se devolverá Null.

Log: Devuelve un tipo de datos Double que especifica el logaritmo natural de un número.

Sintaxis: Log(número)

número Obligatorio; tipo de datos Double o cualquier expresión numérica mayor que cero.

Rnd: Devuelve un tipo de datos Single que contiene un número aleatorio.

Sintaxis: Rnd(número)

número Obligatorio; tipo de datos Single o cualquier expresión numérica válida.

Sgn: Devuelve un valor de tipo Variant (Integer) que indica el signo de un número.

Sintaxis: Sgn(número)

número Obligatorio; cualquier expresión numérica válida.

Los valores devueltos son los siguientes:

Si el número es mayor que cero, Sgn devuelve 1.

Si el número es igual a cero, Sgn devuelve 0.

Si el número es menor que cero, Sgn devuelve –1.

Sin: Devuelve un tipo de datos Double que especifica el seno de un número.

Sintaxis: Sin(número)

número Obligatorio; tipo de datos Double o cualquier expresión numérica válida que exprese un ángulo en radianes.

Sqr: Devuelve un tipo de datos Double que especifica la raíz cuadrada de un número.

Sintaxis: Sqr(número)

número Obligatorio; tipo de datos Double o cualquier expresión numérica igual o mayor que cero.

Tan: Devuelve un tipo de valor Double que especifica la tangente de un número.

Sintaxis: Tan(número)

número Obligatorio; tipo de datos Double o cualquier expresión numérica válida que exprese un ángulo en radianes.

Funciones de texto

Asc: Devuelve un tipo de datos Integer que representa el código de carácter correspondiente a la primera letra de una cadena.

Sintaxis: Asc(cadena)

Cadena Cualquier expresión de cadena válida.

Chr: Devuelve un tipo de datos String que contiene el carácter asociado con el código de carácter especificado.

Sintaxis: Chr(códcar)

códcar Un tipo de datos Long que identifica un carácter.

Format: Devuelve un tipo de dato Variant (String) que contiene una expresión con formato conforme a las indicaciones incluidas en una expresión de formato.

Sintaxis: Format(expresión[, formato[, primer_día_semana[, primera_semana_año]]])

expresión Obligatorio; cualquier expresión válida.

formato Opcional; una expresión de formato válida definida por el usuario o con nombre.

primer_día_semana Opcional; una Constant que especifica el primer día de la semana.

primera_semana_año Opcional; una Constant que especifica la primera semana del año.

Instr: Devuelve un valor de tipo Variant (Long) que especifica la posición de la primera aparición de una cadena dentro de otra.

Sintaxis: Instr([inicio,]cadena1, cadena2[, comparar])

inicio Opcional; expresión numérica que establece la posición de inicio de cada búsqueda. Si se omite, la búsqueda comienza en la posición del primer carácter. Si inicio contiene Null, se producirá un error. El argumento inicio es necesario si se especifica comparar.

cadena1 Obligatorio; la expresión de cadena que se está buscando.

cadena2 Obligatorio; la expresión de cadena en la que se está haciendo la búsqueda.

comparar Opcional; especifica el tipo de comparación de cadena. Si comparar es Null, se producirá un error. Si comparar se omite, el valor Opcióncomparar determinará el tipo de comparación.

LCase: Devuelve un tipo de datos String que se ha convertido a minúsculas.

Sintaxis: LCase(cadena)

cadena Obligatorio; cualquier expresión de cadena válida. Si cadena contiene Null, se devolverá Null.

Left: Devuelve un valor de tipo Variant (String) que contiene un número específico de caracteres a partir del lado izquierdo de una cadena.

Sintaxis: Left(cadena, largo)

cadena Obligatorio. Expresión de cadena de la cual se devuelven los caracteres situados más a la izquierda. Si cadena contiene Null, se devolverá Null.

largo Obligatorio; valor de tipo Variant (Long). Expresión numérica que indica cuántos caracteres devolver. Si es 0, se devolverá una cadena de longitud cero (“”). Si es mayor o igual que el número de caracteres de cadena, se devolverá toda la cadena.

Len: Devuelve un tipo de datos Long que contiene el número de caracteres de una cadena o el número de bytes necesarios para almacenar una variable.

Sintaxis: Len(cadena, nombrevar)

cadena Cualquier expresión de cadena válida. Si cadena contiene Null, se devolverá Null.

nombrevar Cualquier nombre de variable válido. Si nombrevar contiene Null, se devolverá Null. Si nombrevar es un tipo de datos Variant, Len lo tratará como si fuera un tipo de datos String y devolverá siempre el número de caracteres que contenga.

LTrim: Devuelve un tipo de datos Variant (String) que contiene una copia de una cadena especificada sin espacios a la izquierda.

Sintaxis: LTrim(cadena)

cadena Obligatorio; cualquier expresión de cadena válida. Si cadena contiene un valor Null, se devolverá Null.

Mid: Devuelve un valor de tipo Variant (String) que contiene un número especificado de caracteres de una cadena.

Sintaxis: Mid(cadena, inicio[, largo])

cadena Obligatorio; expresión de cadena de la cual se devuelven caracteres. Si cadena contiene un valor Null, se devolverá Null.

inicio Obligatorio; Largo. Posición del carácter de la cadena donde comienza la parte que se desea seleccionar. Si inicio es mayor que el número de caracteres de cadena, Mid devolverá una cadena de longitud cero (“”).

largo Opcional; tipo de datos Variant (Largo). Número de caracteres que se va a devolver. Si se omite o en el texto hay menos caracteres que los de largo (incluyendo el carácter de inicio), se devolverán todos los caracteres desde la posición inicio hasta el final de la cadena.

Right: Devuelve un valor de tipo Variant (String) que contiene un número especificado de caracteres del lado derecho de una cadena.

Sintaxis: Right(cadena, largo)

cadena Obligatorio; expresión de cadena de la cual se devuelven los caracteres situados más a la derecha. Si cadena contiene un valor Null, se devolverá Null.

longitud Obligatorio; tipo de datos Variant (Long). Expresión numérica que indica cuántos caracteres devolver. Si es 0, se devolverá una cadena de longitud cero (“”). Si es mayor o igual que el número de caracteres de cadena, se devolverá toda la cadena.

RTrim: Devuelve un tipo de datos Variant (String) que contiene una copia de una cadena especificada sin espacios finales.

Sintaxis: RTrim(cadena)

cadena Obligatorio; cualquier expresión de cadena válida. Si cadena contiene Null, se devolverá Null.

Space:Devuelve un valor de tipo Variant (String) que se compone del número especificado de espacios.

Sintaxis: Space(número)

número Obligatorio; número de espacios que se desea en la cadena.

StrComp: Devuelve un valor de tipo Variant (Integer) que indica el resultado de una comparación de cadena .

Sintaxis: StrComp(cadena1, cadena2[, comparar])

cadena1 Obligatorio; cualquier expresión de cadena válida.

cadena2 Obligatorio; cualquier expresión de cadena válida.

comparar Opcional; especifica el tipo de comparación de cadena. Si el argumento comparar es Null, se producirá un error.

StrConv: Devuelve un valor de tipo Variant (String) que se ha convertido según lo especificado.

Sintaxis: StrConv(cadena, conversión, código_idioma)

cadena Obligatorio; expresión de cadena que se va a convertir.

conversión Obligatorio; Entero. La suma de valores que especifican el tipo de conversión que se va a realizar.

código_idioma Opcional; el identificador de configuración regional, si es diferente del identificador de configuración regional del sistema. (El del sistema es el predeterminado.)

String: Devuelve un valor de tipo Variant (String) que contiene una cadena de caracteres repetidos con la longitud especificada.

Sintaxis: String(número, carácter)

número Obligatorio; tipo de datos Long. Longitud de la cadena devuelta. Si el número contiene Null, se devolverá Null.

carácter Obligatorio; tipo de datos Variant. Código de carácter que especifica el carácter o expresión de cadena cuyo primer carácter se utiliza para crear la cadena devuelta. Si carácter contiene Null, se devolverá Null.

Trim: Devuelve un tipo de datos Variant (String) que contiene una copia de una cadena especificada sin espacios a la izquierda ni finales.

Sintaxis: Trim(cadena)

cadena Obligatorio; cualquier expresión de cadena válida. Si cadena contiene Null, se devolverá Null.

UCase: Devuelve un tipo de datos Variant (String) que contiene la cadena especificada, convertida a mayúsculas.

Sintaxis: UCase(cadena)

cadena Obligatorio; cualquier expresión de cadena válida. Si cadena contiene Null, se devolverá Null.

Fuente de información: Soporte de Microsoft. Link: https://support.microsoft.com/es-es/office/funciones-de-project-para-campos-personalizados-en-el-escritorio-de-project-7e525143-380f-4083-8d5a-3ecc6ba44f22