



Anexo Editor de Fórmulas

1. Finalidad del Editor de Fórmulas.

El Editor de Fórmulas de LicitEval constituye la herramienta central para la definición y aplicación de criterios automáticos de valoración.

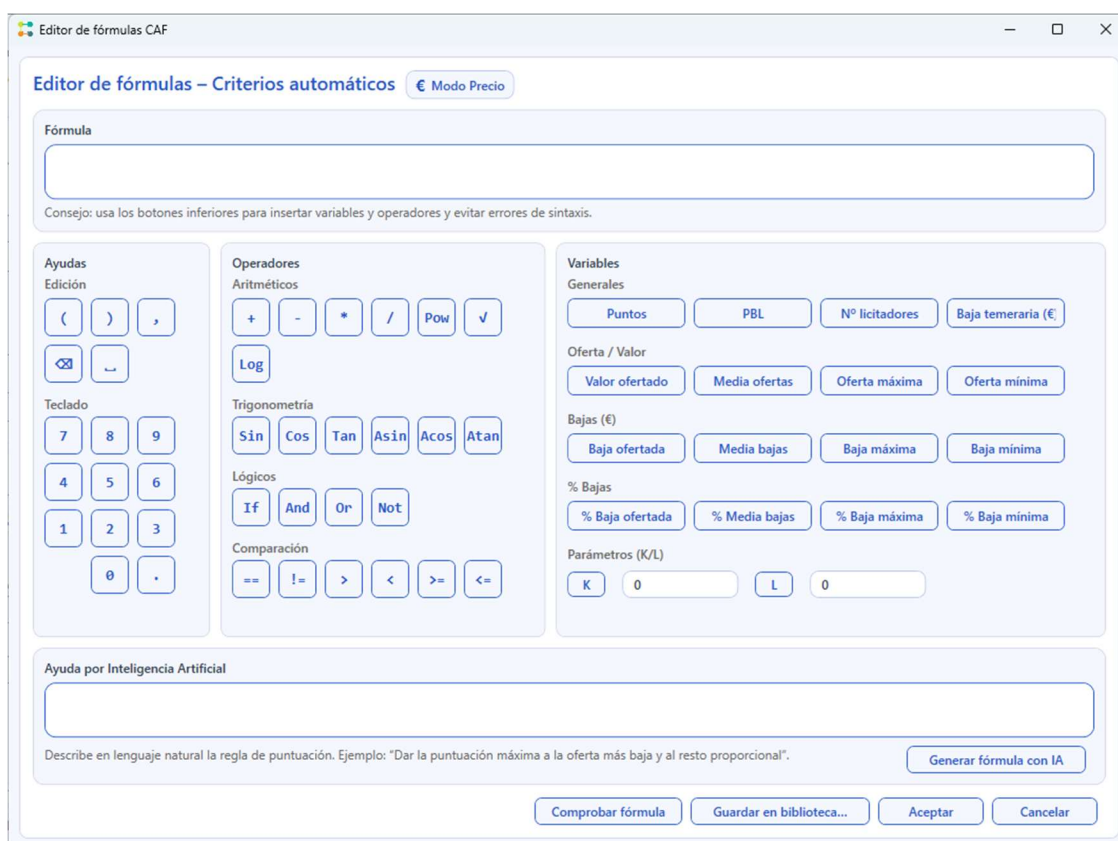


Imagen 1. Editor de fórmulas.

Su función es permitir que el órgano de contratación traduzca al sistema la regla matemática prevista en el pliego, garantizando:

- Transparencia en el cálculo.
- Reproducibilidad de resultados.
- Coherencia con el texto del pliego.
- Automatización completa del proceso de puntuación.

A través del Editor de Fórmulas es posible:

- Generar expresiones matemáticas.



- Introducir condiciones lógicas.
- Aplicar umbrales y tramos.
- Ajustar parámetros dinámicos.
- Validar la sintaxis antes de aplicar el cálculo.
- Generar fórmulas mediante asistencia por Inteligencia Artificial.

La clave para aprovechar al máximo el generador está en entender su sintaxis y los operadores que admite.

2. Estructura de la Ventana del Editor

La ventana del Editor de Fórmulas está organizada en bloques funcionales claramente diferenciados.

2.1 Área principal de expresión

Es el campo donde se escribe la fórmula completa. La fórmula se puede escribir directamente con el teclado, pero para evitar errores, se recomienda encarecidamente introducirla con las variables, operadores y ayudas de edición que nos brinda la plataforma.

La expresión debe devolver siempre un valor numérico, que será la puntuación obtenida por el licitador en ese criterio.

Ejemplo:

```
If([Valor] <= 0, 0, [Puntos] * ([OfertaMinima] / [Valor]))
```

2.2 Panel de variables

En este panel se muestran las variables disponibles para el criterio activo.

Las variables pueden insertarse automáticamente haciendo clic sobre ellas, evitando errores de escritura.

Las variables disponibles dependen del tipo de criterio:

- Criterio económico (Modo Precio)
- Criterio técnico o numérico (Modo Fórmula)

Cuando la fórmula admite parámetros configurables, aparecen campos específicos para introducir:

- K → coeficiente o exponente de ajuste.
- L → valor umbral o límite.

Estos parámetros permiten modificar el comportamiento de la fórmula sin alterar su estructura.

Estas son las variables que podremos utilizar. Siempre estarán encerradas entre corchetes.



Operador	Descripción
[Puntos]	Puntos asignados en el criterio
[PBL]	Precio Base de Licitación
[Valor]	Valor de la oferta que consigna el licitador para ese criterio
[NumLicitadores]	Nº de licitadores admitidos en esta fase
[ImporteBajaTemeraria]	Importe a partir del cual se considera baja temeraria
[MediaOfertas]	Valor medio de todas las ofertas presentadas para ese criterio
[OfertaMinima]	Valor mínimo de todas las ofertas presentadas para ese criterio
[OfertaMaxima]	Valor máximo de todas las ofertas presentadas para ese criterio
[Baja]	Valor de la baja que consigna el licitador para ese criterio
[MediaBajas]	Valor medio de todas las bajas presentadas para ese criterio
[BajaMinima]	Valor mínimo de todas las bajas presentadas para ese criterio
[BajaMaxima]	Valor máximo de todas las bajas presentadas para ese criterio
[%Baja]	Valor del porcentaje de baja que consigna el licitador para ese criterio
[%MediaBajas]	Valor medio de todos los porcentajes de bajas para ese criterio
[%BajaMinima]	Valor mínimo de todos los porcentajes de bajas para ese criterio
[%BajaMaxima]	Valor máximo de todos los porcentajes de bajas para ese criterio

2.3 Operadores

Se pueden insertar operadores aritméticos, trigonométricos, lógicos o de comparación desde los botones habilitados al efecto.

Operadores aritméticos:

Operador	Descripción	Ejemplos de sintaxis
+	suma	5 + 3
-	Resto	10 - 2
X	Multiplicación	4 * 7
/	División	15 / 3

- **Potencia:** Pow(base, exponente)

Ejemplos:

Pow(2, 3) Resultado → 8



$\text{Pow}([\text{Oferta}], 2)$ Resultado \rightarrow Cuadrado de la oferta.

$\text{Pow}(100, 0.5)$ Resultado $\rightarrow 10$

- **Raíz cuadrada:** $\text{Sqrt}(\text{valor})$

Ejemplos:

$\text{Sqrt}([\text{PBL}])$ Resultado \rightarrow Raíz del parámetro.

$\text{Sqrt}(16)$ Resultado $\rightarrow 4$

Para utilizar raíz cúbica o cuadrática, utilizaremos la función “Potencia”

- **Raíz cúbica:** $\text{Pow}(27, 1/3)$ Resultado $\rightarrow 3$

- **Raíz cuarta:** $\text{Pow}(16, 1/4)$ Resultado $\rightarrow 2$

- **Logaritmo:** $\text{Log}(\text{valor}, \text{base})$

Ejemplos:

$\text{Log}(1)$ Resultado $\rightarrow 0$ (porque $\ln(1)=0$)

$\text{Log}([\text{PBL}])$ Calcula \ln del parámetro [PBL].

$\text{Log}(1000, 10)$ Resultado $\rightarrow 3$

$\text{Log}([\text{Oferta}], 10)$ Aplica log base 10 a la variable.

- **Seno, Coseno, Tangente, Arcoseno, Arcocoseno, Arcotangente:** $\text{Sin}(\text{valor})$

Todas estas funciones trabajan en radianes, NO en grados.

$\text{Sin}(0)$ Resultado $\rightarrow 0$

$\text{Cos}(0)$ Resultado $\rightarrow 1$

$\text{Tan}(0)$ Resultado $\rightarrow 0$

$\text{Asin}(1)$ Resultado $\rightarrow \text{PI} / 2 \approx 1.5708$

$\text{Acos}(0)$ Resultado $\rightarrow \text{PI} / 2$

$\text{Atan}(1)$ Resultado $\rightarrow \text{PI} / 4$



- **Operadores condicionales y lógicos con If**

Los operadores condicionales y lógicos son esenciales cuando se trabaja con condiciones en fórmulas. Se admiten operadores como And, Or, Not para manejar evaluaciones booleanas.

El operador **If** en el generador de fórmulas se utiliza para evaluar expresiones condicionales. Funciona de forma similar al operador **If** de Excel, con la salvedad que el separador entre argumentos que utiliza Excel es “;” y el separador utilizado en la LicitEval es “,”.

Un **If** anidado en LicitEval es cuando colocamos un **If** dentro de otro. Esto nos permite evaluar una serie de condiciones de forma secuencial. Cuando anidamos **If**, colocamos otro **If** en la posición de **valor_si_falso**, o en **valor_si_verdadero** si queremos evaluar más de una condición dependiendo del resultado.

Sintaxis del If: If(condición, valor_si_verdadero, valor_si_falso)

Operador	Descripción
And	Verdadero si ambos son verdaderos
Or	Verdadero si uno es verdadero
Not	Invierte el valor booleano
==	Igualdad, igual que
!=	Desigualdad, distinto que
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que

- **Uso de Paréntesis y Comas en If**

- **Paréntesis:** Se usan para agrupar expresiones y evitar ambigüedades. En condicionales como **If**, asegúrese de que la condición esté bien delimitada.
- **Comas:** Las comas separan los tres componentes del **If**: condición, valor si verdadero, valor si falso. Deben estar bien ubicadas, ya que se requiere una estructura precisa.

- **Ejemplos de uso del If**

- **Ejemplo simple:**

If(5 > 3, 10, 20)

Si 5 > 3 es verdad, devuelve 10. Si es falso, devuelve 20.

- **Ejemplo de fórmula combinada con If operadores lógicos y matemáticos:**

If([Valor] > 10 and [Puntos] < 50, [Puntos] + 10, [Puntos] - 10)



Si el valor es mayor que 10 y los puntos son menores que 50, suma 10 a los puntos.

De lo contrario, resta 10 a los puntos.

- **Ejemplo de If anidados** Tres condiciones en secuencia.

`If([Valor] > 10, 500, If([Valor] > 5, 200, 100))`

Primero se evalúa si `[Valor] > 10`. Si es verdadero, el resultado es 500.

Si `[Valor] <= 10`, pasa al siguiente If anidado, evaluando si `[Valor] > 5`.

- Si `[Valor] > 5`, el resultado es 200.
- Si `[Valor] <= 5`, el resultado final es 100.

Este If anidado se puede expandir para manejar condiciones más específicas.

En cada paso, si una condición es verdadera, se obtiene un resultado y se detiene la evaluación. Si es falsa, pase a la siguiente condición.

- **Ejemplo Complejo con If Anidados y Valores entre Corchetes**

Supongamos que tenemos un sistema de evaluación para clasificar una puntuación en función de media de las ofertas presentadas. Si la media de las ofertas presentadas son mayores o iguales a 50, se le aplicarán diferentes umbrales de evaluación que si la media de ofertas presentadas menor de 50.

`If([Media] >= 50, If([Valor] >= 80, 20, If([Valor] >= 50, 10, 5)), If([Valor] >= 60, 20, If([Valor] >= 30, 10, 5)))`

Primero evalúa la media de las ofertas presentadas:

- Si `[Media] >= 50`, pasa a evaluar la puntuación:
 - Si `[Valor] >= 80`, devuelve 20.
 - Si `[Valor] >= 50`(y es menor de 80), devuelve 10.
 - Si no cumple ninguna de las anteriores, devuelve 5.
- Si `[Media] < 50`, evalúa una escala diferente:
 - Si `[Valor] >= 60`, devuelve 20.
 - Si `[Valor] >= 30`(y es menor de 70), devuelve 10.
 - De lo contrario, devuelve 5.

Este tipo de fórmula es útil cuando necesitamos una lógica más compleja, como aplicar diferentes criterios de evaluación según la media de las ofertas y la oferta presentada, u otros factores.

- **Consejos para Trabajar con If Anidados**

Agrupar con paréntesis: Al anidar If, los paréntesis son importantes para agrupar cada condicional correctamente.

Mantén una estructura ordenada: Demasiados If anidados pueden volverse difíciles de leer. Trata de simplificar la lógica o de usar estructuras alternativas cuando sea posible.



○ **Prioridad de los Operadores**

LicitEval sigue reglas estándar de precedencia de operadores. A continuación, se muestra la prioridad de los operadores de mayor a menor:

Prioridad	Operador
1	() paréntesis
2	*, /
3	+,-
4	<, <=, >, >=
5	==, !=
6	And, Or, Not

○ **Ejemplo con Prioridad de Operadores.**

$$3 + 5 * 2$$

Este ejemplo evalúa primero $5 * 2$, luego suma 3. El resultado es 13.

Para cambiar la prioridad, puedes usar paréntesis:

$$(3 + 5) * 2$$

Ahora se evalúa primero $3 + 5$, y luego se multiplica por 2, resultando en 16.

○ **Errores Comunes al Usar el generador de fórmulas**

Para evitar errores en tus fórmulas:

- Uso de comas y paréntesis: Asegúrese de que las funciones con If, contengan sus argumentos separados por comas y estén bien delimitadas por paréntesis.
 - Correcto: `If(5 > 3, 10, 20)`
 - Incorrecto: `If(5 > 3 10 20)`
- Comparaciones bien formadas: Cuando utilice operadores como `>`, `<`, `==`, asegúrese de que ambas partes de la expresión estén bien definidas.
 - Correcto: `If([Valor] > 10, 1, 0)`
 - Incorrecto: `If([Valor] 10, 1, 0)`
- Uso de operadores lógicos: LicitEval evalúa estrictamente las expresiones booleanas, por lo que es importante usar And, Or y Not de manera correcta.
 - Correcto: `If([Valor] > 10 And [Media] < 20, 1, 0)`
 - Incorrecto: `If([Valor] > 10 y [Media] < 20, 1, 0)`



2.4 Ayudas

Permiten escribir valores numéricos, y otros elementos como para generar la fórmula.

Paréntesis: Se usan para agrupar expresiones y evitar ambigüedades.

Comas: Las comas separan los tres componentes del If: condición, valor si verdadero, valor si falso. Deben estar bien ubicadas, ya que se requiere una estructura precisa. Es similar al uso de punto y coma de Excel.

2.5 Ayuda por Inteligencia Artificial y Botón “Generar fórmula con IA”

Permite redactar en lenguaje natural la lógica del criterio y obtener una propuesta de fórmula.

Ejemplo de entrada:

“Otorgar la máxima puntuación a la oferta más baja y penalizar progresivamente el resto.”

La IA genera una fórmula compatible con el sistema.

Debe revisarse y validarse siempre antes de su uso.

2.6 Botón Comprobar fórmula.

El botón Comprobar fórmula realiza una validación y comprobación automática de la expresión:

- Verifica que la sintaxis es correcta.
- Comprueba que las variables existen.
- Detecta paréntesis no cerrados.
- Identifica operadores mal formados.

Es obligatorio validar antes de guardar la fórmula.

2.7 Botón Guardar en biblioteca...

Este botón nos abrirá una nueva ventana que, una vez generada la fórmula, nos permitirá guardarla en la biblioteca de usuario, especificándole los siguientes parámetros

- Nombre específico de la fórmula.
- Descripción detallada.
- Insertar adjuntos como una imagen y un gráfico.
- Consignar si utiliza valores K y/o L, dándoles un valor predefinido.



3. Flujo Completo de Trabajo

1. Determinar el tipo de criterio.
2. Redactar la regla matemática prevista en el pliego.
3. Introducir o generar la fórmula.
4. Validar.
5. Comprobar resultados.
6. Guardar y aplicar.

4. Ejemplo Real Completo

Datos del supuesto

- Puntos máximos ([Puntos]): 50
- PBL: 100.000 € (*en este ejemplo no entra en la fórmula; solo es contexto del expediente*)
- Ofertas ([Valor]):

A: 80.000 € B: 90.000 € C: 100.000 €

1) Fórmula proporcional clásica

$[Puntos] * ([OfertaMinima] / [Valor])$

○ Cálculo

A) Oferta 80.000

- Cociente = $80.000 / 80.000 = 1$
- Puntos = $50 * 1 = 50$

B) Oferta 90.000

- Cociente = $80.000 / 90.000 = 0,888888...$
- Puntos = $50 * 0,888888... = 44,44$

C) Oferta 100.000

- Cociente = $80.000 / 100.000 = 0,8$
- Puntos = $50 * 0,8 = 40,00$

○ Resultados:

- 80.000 → **50,00**
- 90.000 → **44,44**
- 100.000 → **40,00**



2) Fórmula con K = 3 (penalización no lineal)

La idea es “endurecer” la caída de puntos cuando la oferta se aleja de la mínima, elevando el cociente a una potencia K.

$[Puntos] * Pow([OfertaMínima] / [Valor], K)$

○ Cálculo.

A) Oferta 80.000

- Cociente = $80.000 / 80.000 = 1$
- Potencia = $1^3 = 1$
- Puntos = $50 * 1 = 50,00$

B) Oferta 90.000

- Cociente = $80.000 / 90.000 = 0,888888...$
- Potencia = $(0,888888...) ^3 = 0,7023319616...$
- Puntos = $50 * 0,7023319616... = 35,12$

C) Oferta 100.000

- Cociente = $80.000 / 100.000 = 0,8$
- Potencia = $0,8^3 = 0,512$
- Puntos = $50 * 0,512 = 25,60$

○ Resultados (K=3):

- 80.000 → **50,00**
- 90.000 → **35,12**
- 100.000 → **25,60**

3) Comparativa

Oferta	Clásica	K=3	Diferencia
80.000	50,00	50,00	0,00
90.000	44,44	35,12	9,32
100.000	40,00	25,60	14,40

Interpretación:

- Con la fórmula clásica, subir de 80k a 100k “solo” baja de 50 a 40.
- Con K=3, esa subida baja de 50 a 25,6: castiga mucho más a las ofertas más altas.
- La oferta mínima siempre queda en 50 puntos (porque el cociente es 1).



5. Recomendaciones.

- Documentar siempre la lógica de la fórmula.
- Verificar coherencia con el pliego.
- Evitar complejidad innecesaria.
- Controlar divisiones por cero.
- Validar antes de guardar.
- Probar con datos reales del expediente.

El Desarrollador no se hace responsable de los posibles perjuicios que pudiera ocasionar al Usuario, derivados de los fallos o errores de cálculo y/o resultados obtenidos por la Aplicación.