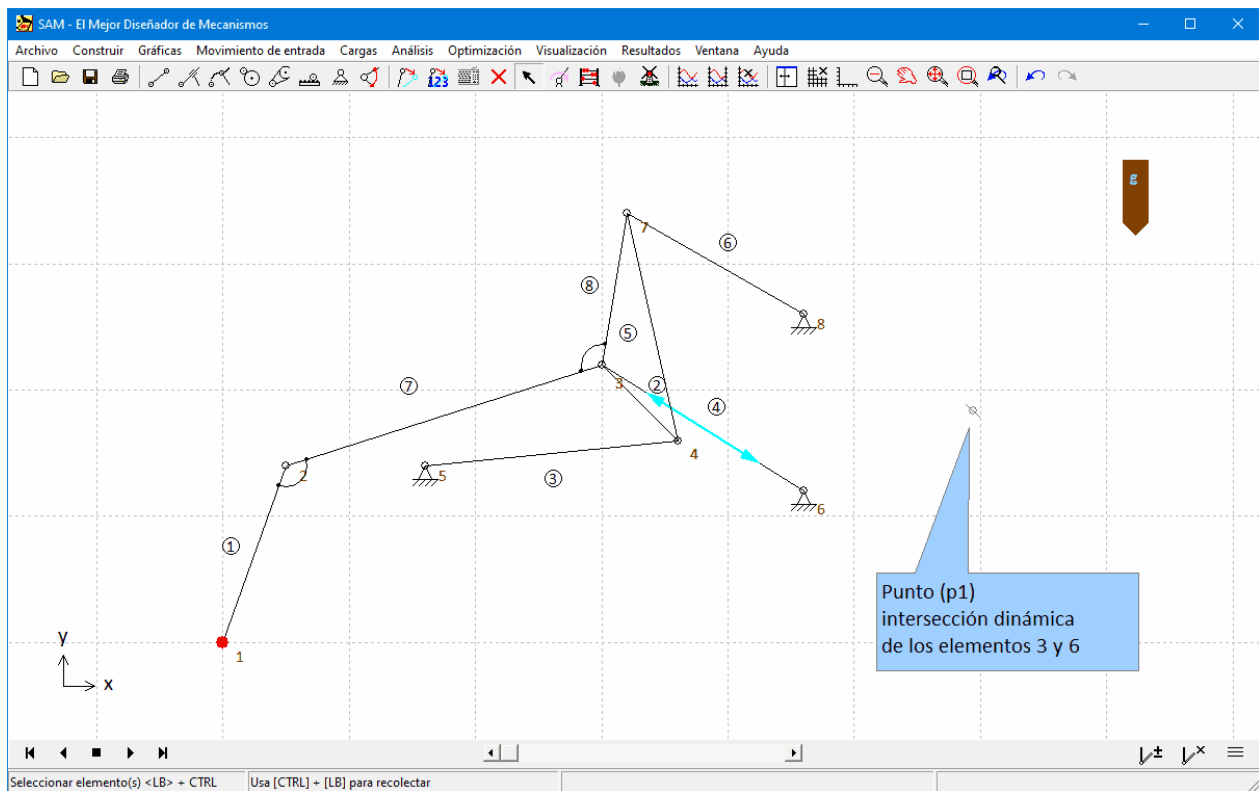


## ¿Qué es nuevo en SAM 8.3?

### Puntos dinámicos

La posibilidad de construir puntos a través de la intersección de líneas e incluso la intersección de ejes se ha vuelto dinámica. Al mover los objetos utilizados en la creación de un punto, las coordenadas xy de ese punto se adaptan automáticamente. Eso es válido para la situación en la que un objeto se mueve en la posición de referencia, pero también cuando se realiza una animación.

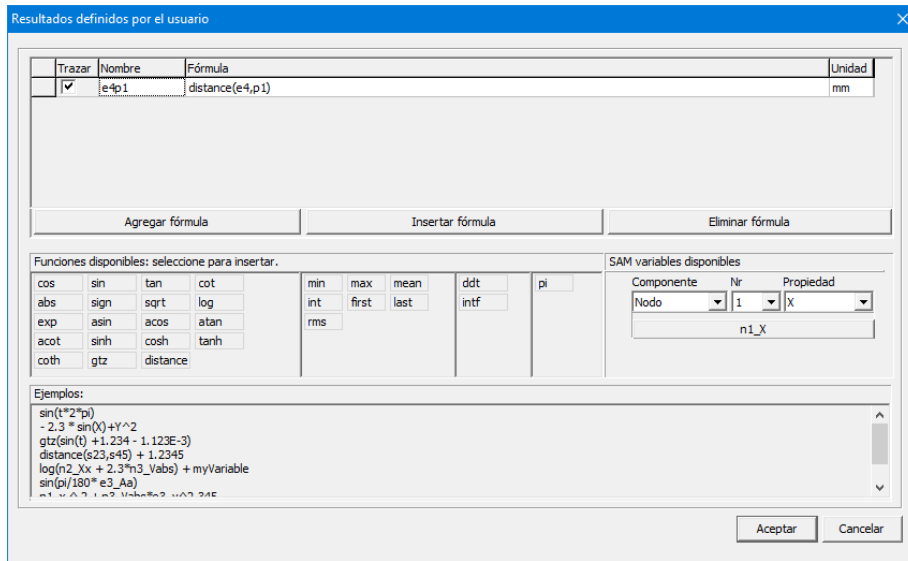
De esta manera, se puede crear, por ejemplo, el punto p1 como la intersección de la viga 3 y la viga 6, que constituye el centro de rotación momentáneo del triángulo formado por el acoplador triangular. Cuando el mecanismo está animado y se mueve de una posición a la siguiente, la ubicación del punto p1 se adapta automáticamente.



**El punto p1 se crea como una intersección de las dos líneas (infinitas) correspondientes a los elementos de viga 3 y 6. El punto es dinámico y se mueve durante una animación.**

### Distancia

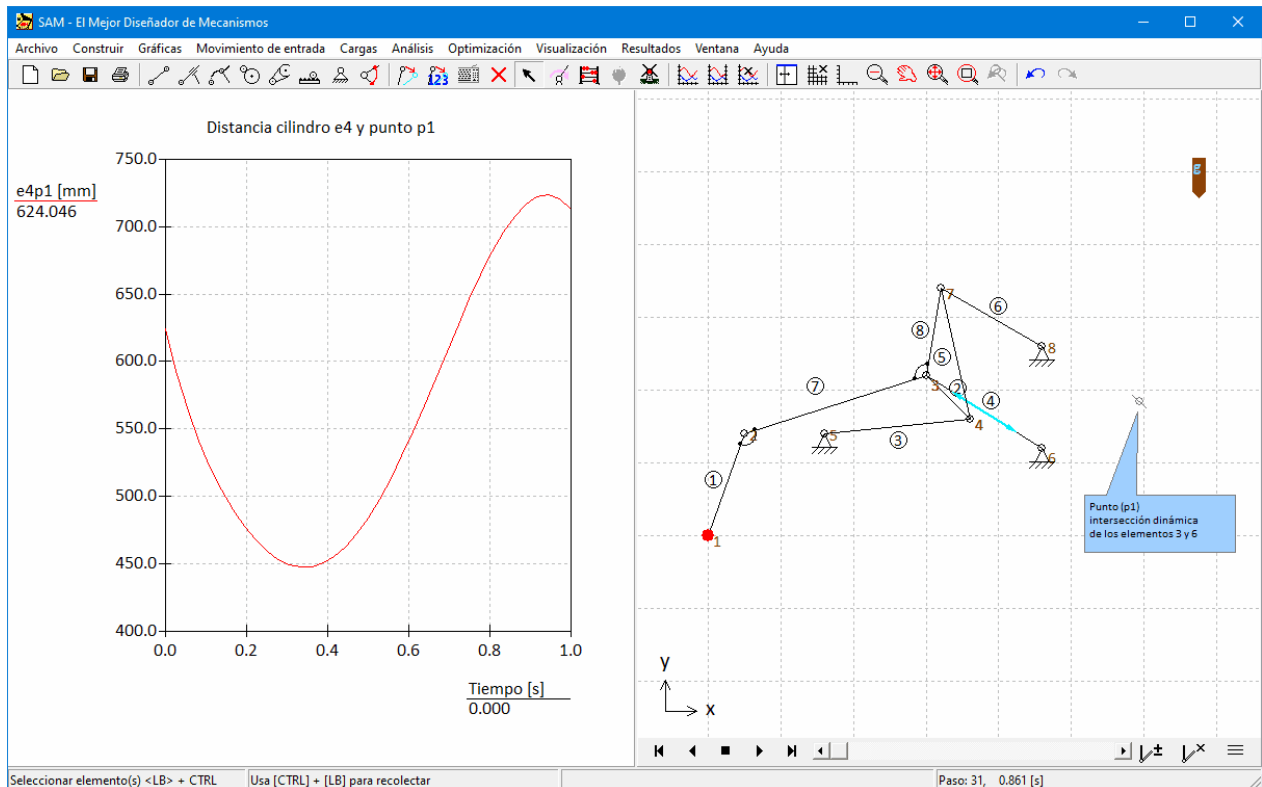
El módulo de posprocesamiento (Resultados > Definido por el usuario...) se ha ampliado con una nueva función: `distance(objeto1,objeto2)`, en la que un objeto puede ser un nodo, un elemento, un punto o una forma.



Definición de la distancia entre el elemento e4 y el punto p1.

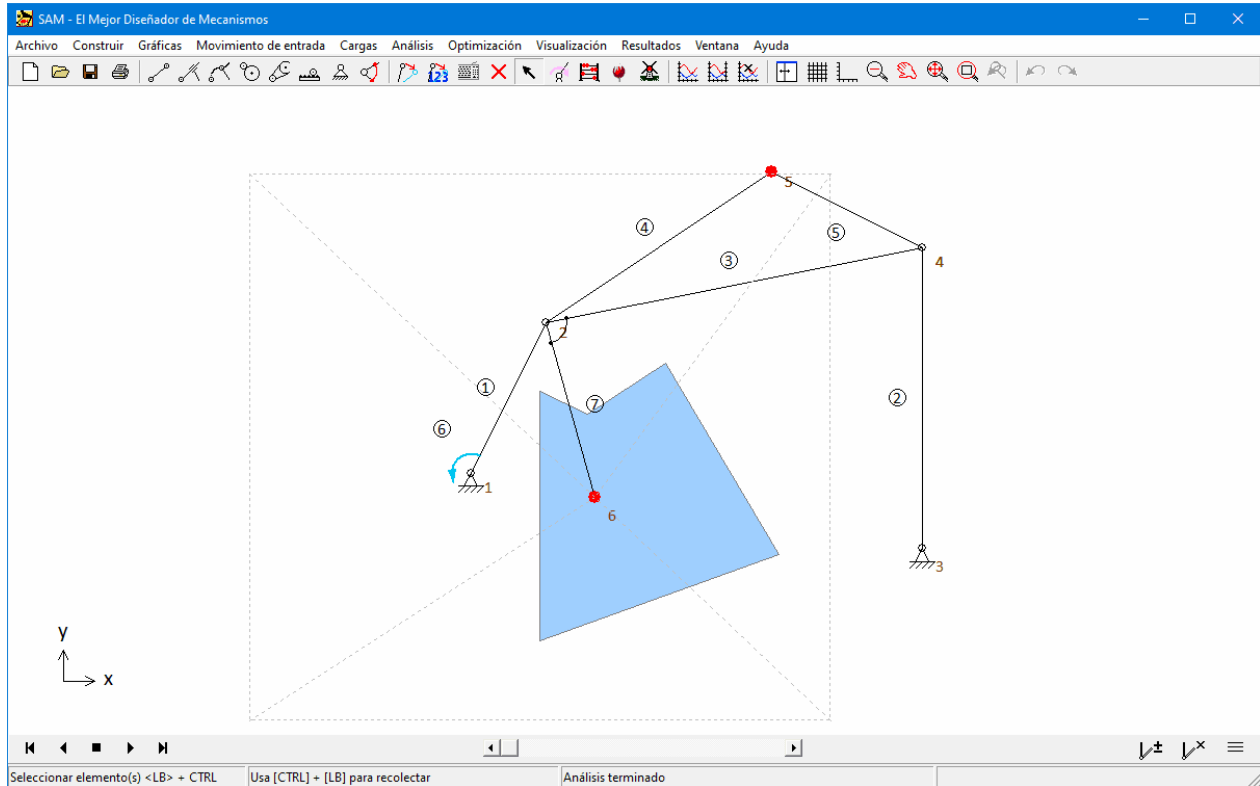
El uso de esta característica es múltiple:

1. En el ejemplo anterior puede ser útil graficar la distancia entre el cilindro impulsor (elemento e4) y el punto dinámico p1 para monitorear si esa distancia es siempre positiva y no llega a cero, ya que en esa situación el mecanismo estaría bloqueado. .

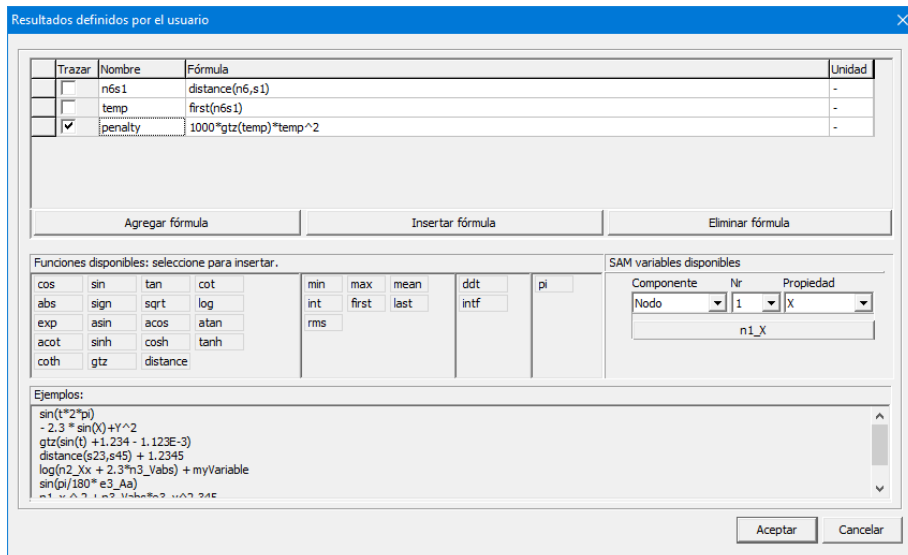


El mecanismo y la gráfica de la distancia entre el elemento e4 y el punto p1 como función del tiempo se traza al lado del mecanismo

2. Durante una optimización, la ubicación xy de un nodo puede ser un parámetro de optimización. Usando la función de distancia, se puede definir una función de penalización para la optimización con el fin de restringir la ubicación xy de ese nodo a cualquier polígono.



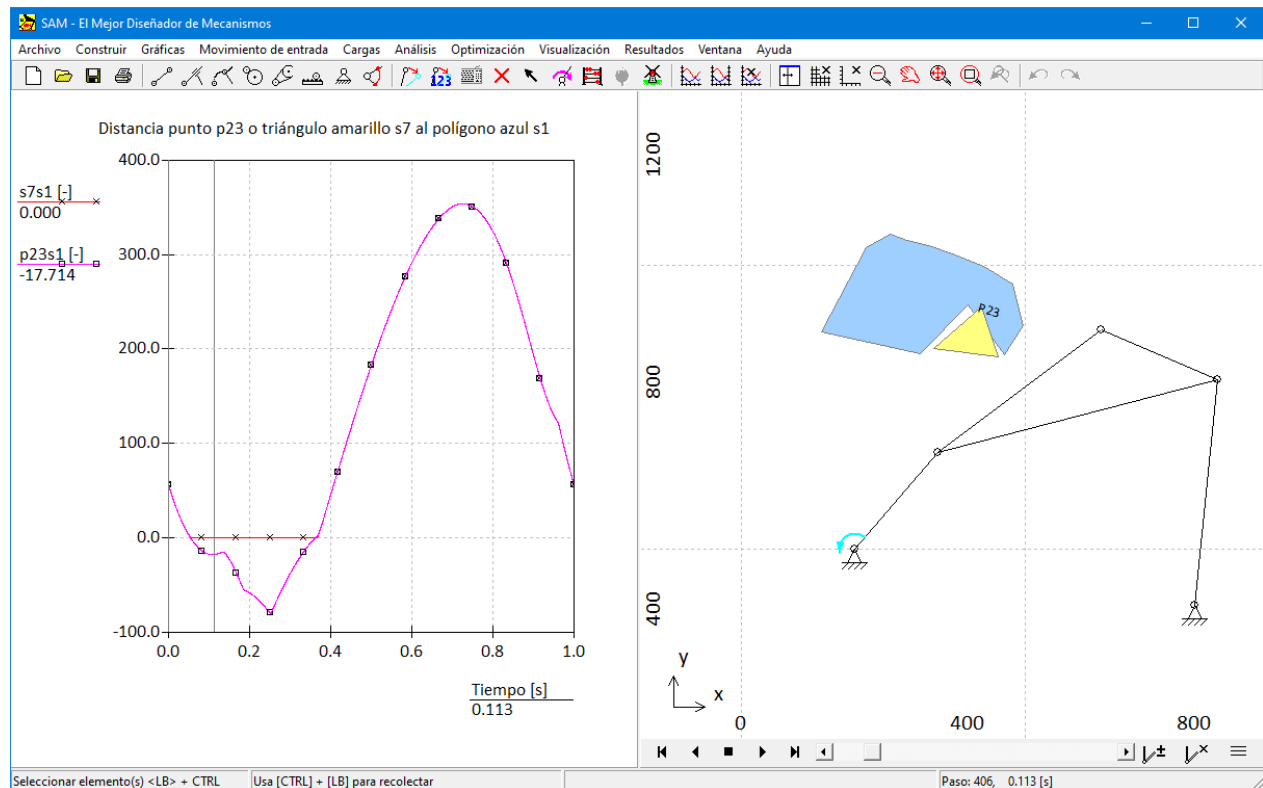
**Ejemplo de una optimización en la que el nodo n6 es un parámetro de diseño pero debe estar restringido al área definida por el polígono (forma s1)**



**Definición de la función de penalización para restringir el nodo n6 al polígono s1. Tan pronto como el nodo n6 se encuentra fuera del polígono en la posición inicial del mecanismo, la penalización es mayor que 0.**

### 3. Detección de colisión entre objetos

La función de distancia se puede utilizar para detectar la colisión entre objetos y, según el tipo de objetos, incluso la profundidad de penetración (o la distancia mínima de retracción para evitar la colisión). En el siguiente ejemplo, el polígono azul es una forma estacionaria  $s1$ , mientras que el triángulo amarillo (forma  $s7$ ) está unido al acoplador. En el gráfico se ve la variable  $s7s1$ , que corresponde a la distancia entre las dos formas y la variable  $p23s1$ , que corresponde a la distancia entre el punto  $p23$  del triángulo y el polígono azul. En el caso de dos formas en colisión, la distancia se establece en cero, mientras que en el caso de un punto o nodo que se mueve hacia una forma, tanto la distancia positiva como la negativa (distancia negativa = profundidad de penetración) se muestran en el gráfico.



Ejemplo de detección de colisión mediante el uso de la función de distancia.

### Copiar/Pegar (Ctrl-C/Ctrl-V) para gráficos/formas

Ahora se puede usar Ctrl-C y Ctrl-V para copiar y luego pegar cualquier forma.