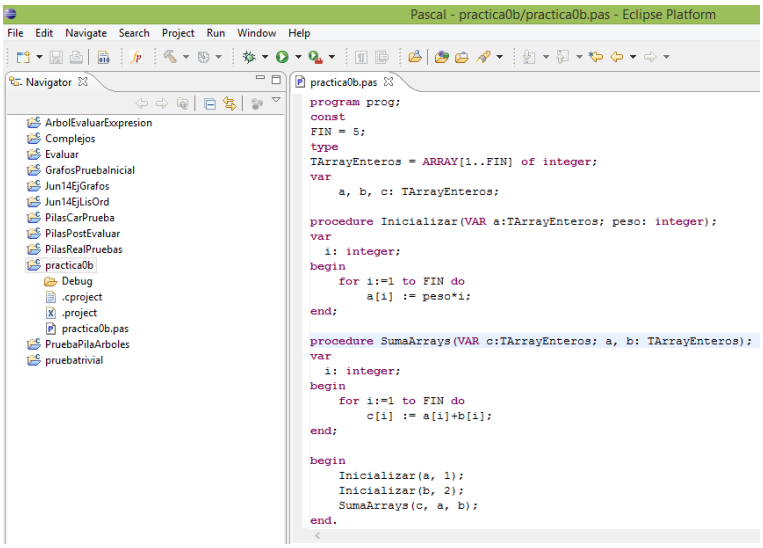
Tutorial para depurar aplicaciones Pascal mediante EclipseGavab

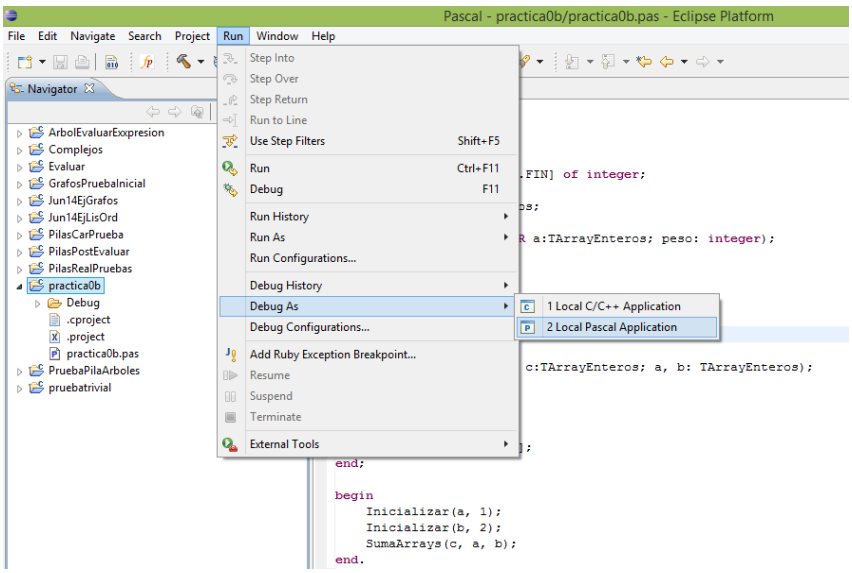
Estructuras de Datos - 1º GII y dobles (Vicálvaro/Móstoles) – Antonio Sanz Montemayor

Con el fin de facilitar la depuración de aplicaciones en EclipseGavab hemos creado un pequeño tutorial con las instrucciones mínimas para comenzar. Una correcta depuración de un programa es vital para un proceso de desarrollo de software en cualquier lenguaje de programación. Partimos de un programa cualquiera, por ejemplo el de la Figura 1 (debe haber sido guardado y compilado, si no compila, no se podrá depurar).

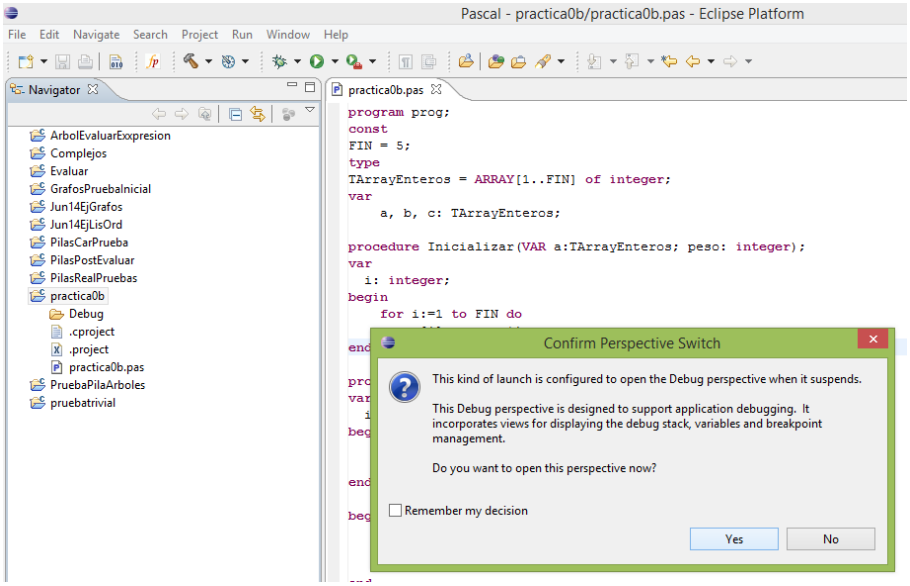


Para empezar la depuración el programa debe compilar, si no lo hace es porque tiene errores en tiempo de compilación (sintácticos o cualquier tipo de error detectable por el compilador, identificador duplicado, inexistencia de definición de tipo utilizado, *type mistmatch*, etc.). Una vez compilado, activamos la depuración:

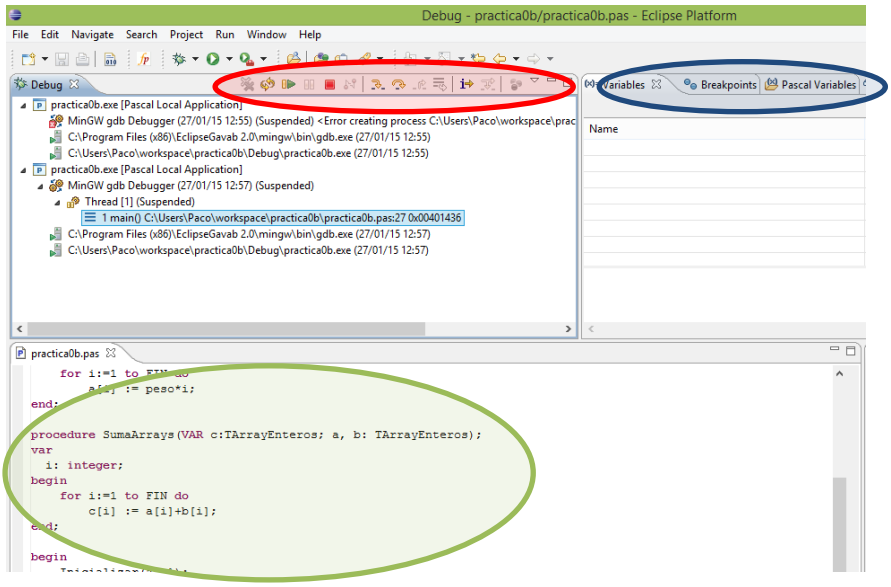
Menu Run → Debug As → Local Pascal Application



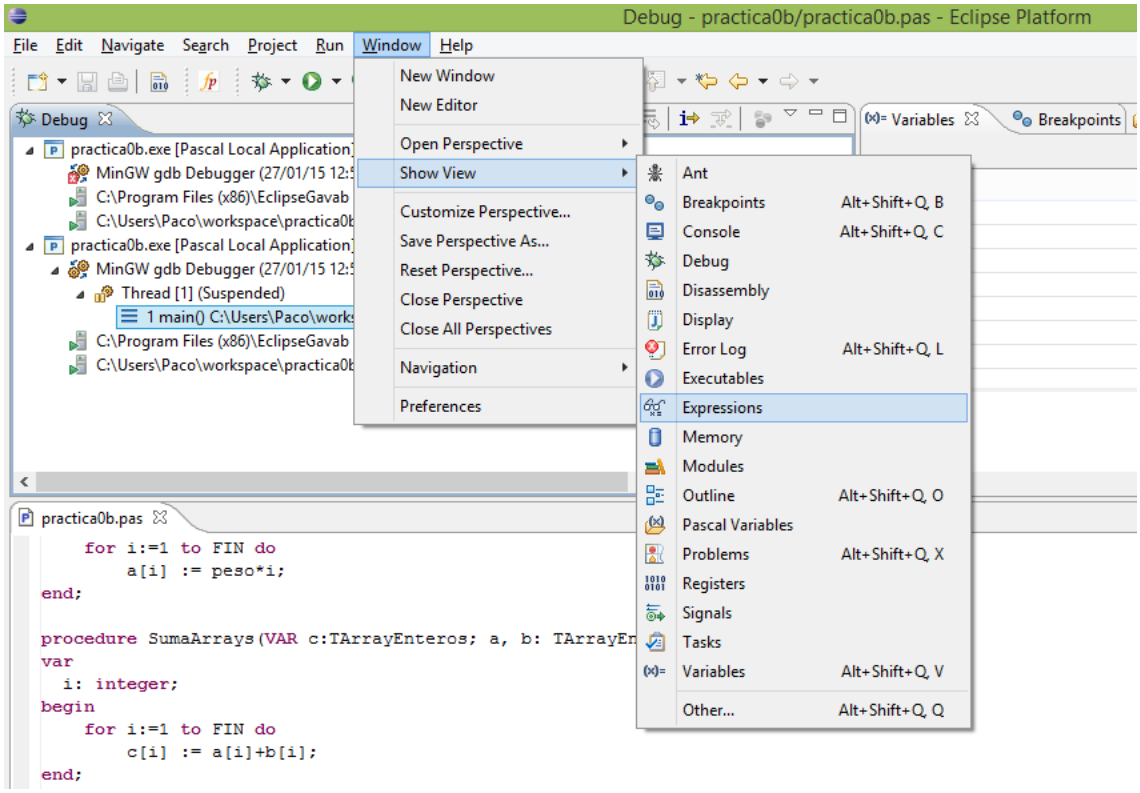
Eclipse nos preguntará si queremos cambiar la vista a modo Depuración. Elegimos Sí.



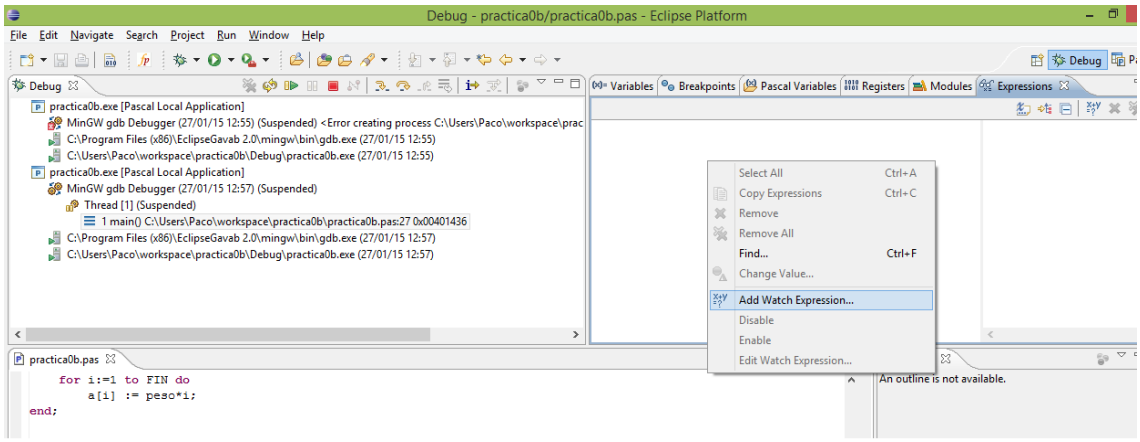
La vista tomará un aspecto parecido al que se muestra a continuación, con el código en la parte baja (globo verde en la imagen), una ventana de visualización de variables (locales a subprogramas), *breakpoints* (puntos de ruptura), etc. (globo azul) y, especialmente interesantes, los nuevos botones habilitados (globo rojo). Especial atención merecen los botones de *Step Into* (para inspección profunda de subprogramas) y *Step Over* (salto de instrucción sin entrar en subprograma)



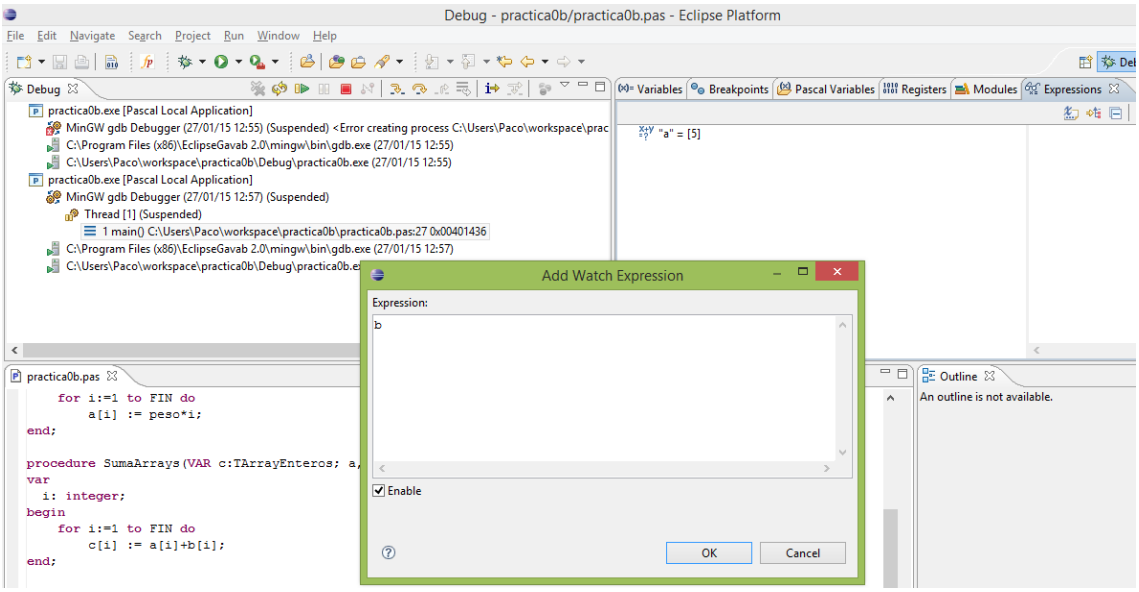
Si no tenemos habilitada la vista de expresiones (*Expressions*) lo hacemos para evaluar el contenido de variables globales en Pascal. Para ello, menú *Window* →*Show View* → *Expressions*



En este momento, podemos añadir alguna variable global a la pestaña de *Expressions* mediante el botón derecho del ratón y "*Add Watch Expressions*" (NOTA: Aunque *Watch Expressions* se utiliza para evaluar expresiones más complejas, podemos utilizarlo para visualizar el contenido de variables globales)

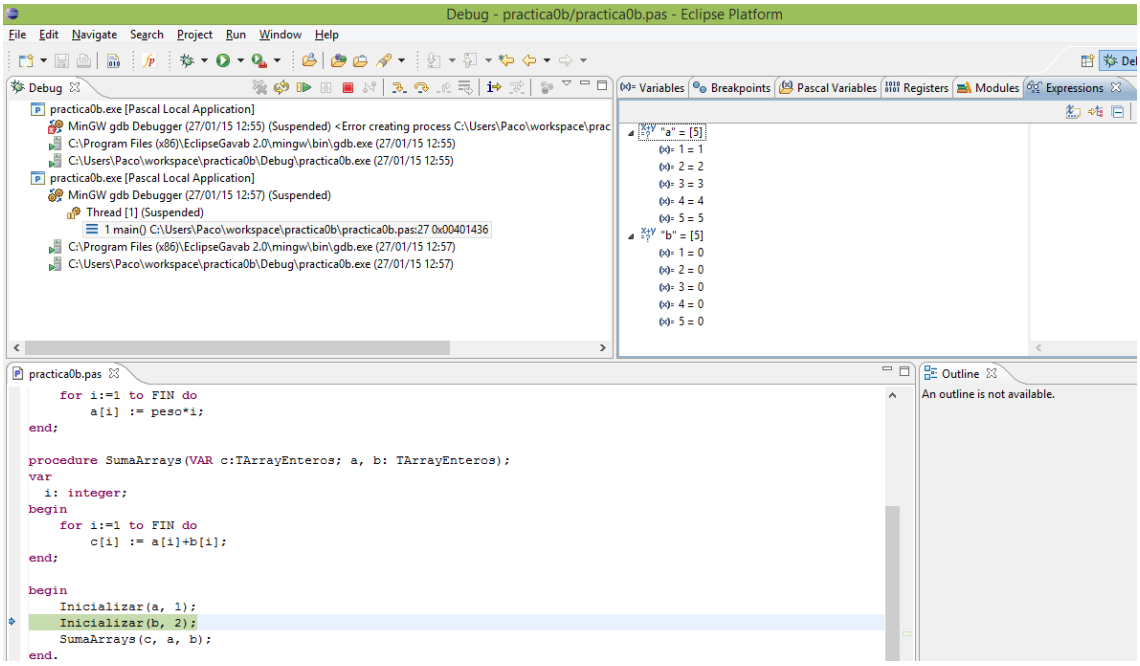


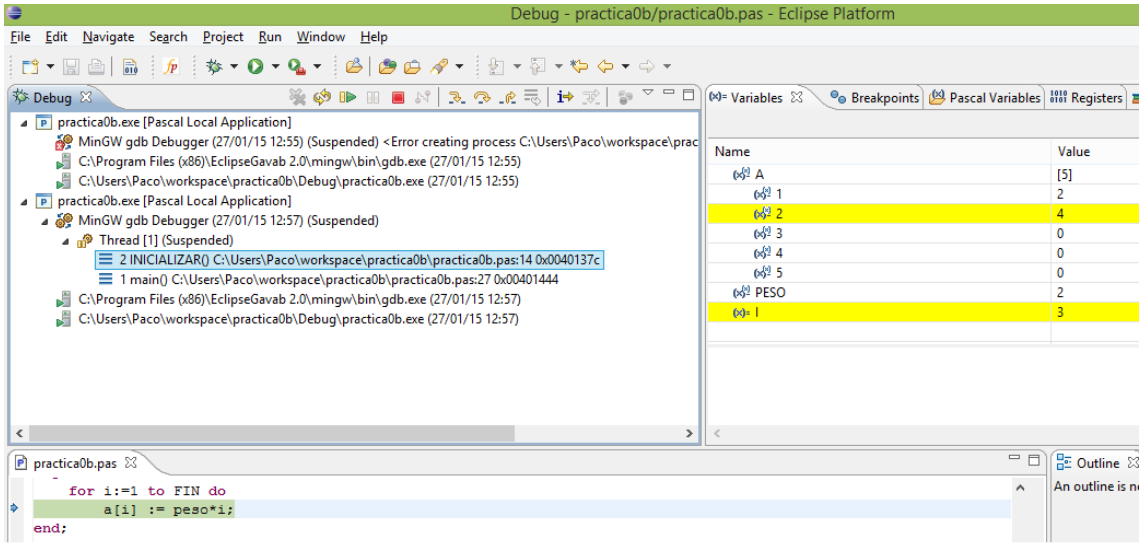
Por ejemplo añadimos consecutivamente las variables globales "a" y "b" (que son de tipo TArrayEnteros).



Podemos ejecutar la primera instrucción pulsando el botón de Step Over y se marcará la segunda instrucción del programa principal, lista para ejecutarse.

En este momento podemos desplegar en la ventana de *Expressions* las variables “a” y “b”, que siendo *arrays* mostrarán su contenido. En este ejemplo, “a” ya ha sido inicializado mientras que “b” aún no (y por defecto Pascal ha inicializado todas sus componentes a 0).



Estando en la segunda instrucción podemos inspeccionar paso a paso la correcta ejecución del código del subprograma Inicializar pulsando el botón *Step Into*. Pulsando varias veces Step

Over (o Step Into) podemos ejecutar varias iteraciones del bucle FOR y ver la actualización de los valores de las variables locales al subprograma (en la pestaña de *Variables*) ”a”, “peso” e “i”. En la imagen se han ejecutado 2 iteraciones y estamos en medio de la tercera iteración, antes de la asignación del valor a la componente 3 del array.