

A continuación se muestran los pasos a seguir para construir el ejercicio:

1. Abra Descartes e introduzca un pulsador con identificador y nombre a , valor inicial 2 e incrementos discretos de 0.1.
2. Duplique el pulsador pero con un identificador b . Asigne al nuevo pulsador el nombre b y dele un valor inicial de 1.
3. Introduzca un gráfico tipo texto que devuelva el resultado de comparar la variable a con la b para los operadores $>$, $<$, $=$, $!=$, $>=$ y $<=$, uno en cada renglón. Haga que el texto muestre 1 decimal de forma *no fija*.
4. Agregue una instrucción al algoritmo *CALCULOS* que asigne a la variable *mostraramarillo* el valor de 1 si a es distinto de b y de lo contrario le asigne 0.
5. Agregue un punto amarillo de tamaño 10 en el origen del plano cartesiano y que sólo se muestre cuando la variable *mostraramarillo* vale 1.
6. Reduzca lo más posible el texto en el campo *dibujar si* del punto.
7. Duplique el punto amarillo pero ahora asígnele la coordenada (1,0) y un color rojo. Haga que este punto sólo se muestre si la variable *mostrarrojo* es verdadera. Dicha variable es verdadera cuando a es mayor que b y ambas son positivas.
8. Encuentre un nuevo hogar para las asignaciones actualmente en el algoritmo *CALCULOS* para que no se ejecuten cada vez que el usuario hace cualquier cosa sobre el interactivo. Ello con el fin de optimizar la función de la escena.
9. Haga que las asignaciones de valor a las variables *mostraramarillo* y *mostrarrojo* se hagan desde el principio al cargar el interactivo.

A continuación se presentan los resultados esperados de cada paso de este ejercicio, así como algunas observaciones.

1. Es necesario moverse al selector *Controles* en el editor de configuraciones y oprimir el botón $+$. En el menú de la ventana emergente hay que seleccionar *pulsador* e introducir el identificador a . Una vez agregado, el campo *valor* debe cambiarse a 2. Se debe marcar también el checkbox *discreto* y se debe cambiar el campo *decimales* a 1 para asegurar que los incrementos serán discretos y de 0.1 unidades.
Tras aplicar los cambios, el pulsador a aparece en el sur del interactivo y respeta las condiciones dadas al cambiarle su valor.
2. Habiendo seleccionado el pulsador a en el selector *Controles*, se pulsa el botón $*$ para duplicarlo. En la ventana emergente se da el nuevo identificador b a este control. Una vez agregado se cambia el campo *nombre* del duplicado por b y su campo *valor* por 1.
Tras aplicar los cambios, ambos pulsadores aparecen en el sur del interactivo y respetan las condiciones impuestas cada que se cambian sus valores.
3. Es necesario moverse al selector *Gráficos* en el editor de configuraciones para agregar el gráfico tipo texto. Una vez agregado, se pulsa el botón T en dicho gráfico. En la ventana de introducción de texto se debe tener
El resultado de la comparación $[a]>[b]$ es $[a>b]$
El resultado de la comparación $[a]<[b]$ es $[a<b]$
El resultado de la comparación $[a]==[b]$ es $[a==b]$
El resultado de la comparación $[a]!=[b]$ es $[a!=b]$
El resultado de la comparación $[a]>=[b]$ es $[a>=b]$
El resultado de la comparación $[a]<=[b]$ es $[a<=b]$

Una vez que se acepta la introducción del texto, hay que colocar 1 en el campo *decimales* del gráfico tipo texto y quitar la marca al checkbox *fijo*. De esta forma, sólo se mostrará un decimal sólo cuando sea significativo.

Tras aplicar los cambios, el interactivo muestra los 6 renglones de texto. El texto se mantiene igual hasta que el valor de b iguala o rebasa al de a . Observe que el comportamiento de las

comparaciones es el esperado.

4. Es necesario moverse al selector *Programa* y seleccionar el algoritmo *CALCULOS* en el panel de algoritmos a la izquierda. Una vez seleccionado hay que introducir el siguiente texto en el panel *hacer* del algoritmo

```
mostraramarillo=(a!=b)?1:0
```

Tras aplicar los cambios, el interactivo no muestra cambio alguno puesto que no se ha indicado que la asignación a la nueva variable *mostraramarillo* se manifieste explícitamente.

5. Hay que moverse al selector *Gráficos* y agregar el gráfico tipo *punto*. Una vez agregado se le cambia el color amarillo usando el botón a la derecha de *color* que muestra la herramienta de control de colores. El campo *tamaño* del gráfico tipo punto se cambia por 10. Es necesario agregar una condición en el campo *dibujar si* del punto. Recordamos que sólo si el valor de la expresión en este campo es 1 se trazará el gráfico. Por lo mismo, podemos introducir ahí

```
mostraramarillo==1
```

Tras aplicar los cambios, el texto sigue igual y se pinta el punto, que por defecto está colocado en el origen. No obstante, si se mueve el valor de uno de los pulsadores de tal forma que ambos pulsadores tengan el mismo valor, entonces el punto desaparece. Si vuelven a ser diferentes, el punto vuelve a mostrarse.

6. Recordamos que la variable *mostraramarillo* de por sí ya tiene valores 1 o 0 por la asignación que se le hizo. Así pues, es suficiente introducir sólo el texto *mostraramarillo* en el campo *dibujar si* del gráfico punto y el interactivo debe seguir comportándose igual.

Tras hacer eso y aplicar los cambios, el gráfico sigue comportándose igual.

7. Primero hay que seleccionar el punto amarillo en el selector *Gráficos* y oprimir el botón * para duplicarlo. Luego hay que cambiar el campo *expresión* del duplicado por $(a > b) \wedge (b > 0)$. Se cambia su color por rojo usando el botón a la derecha de *color*. El campo *dibujar si* se cambia por *mostrarrojo*. Posteriormente se agrega el texto *mostrarrojo* = $(a > b) \wedge (b > 0)$ en el algoritmo *CALCULOS*. De tal forma que el panel de texto *hacer* del algoritmo *CALCULOS* queda:

```
mostraramarillo=(a!=b)?1:0
```

```
mostrarrojo=(a>b)&(b>0)?1:0
```

Tras aplicar los cambios, todo sigue igual con la excepción de un punto rojo que aparece a la derecha del amarillo. Si se modifican los pulsadores de tal forma que *a* sea menor que *b*, el punto desaparece. Igualmente, si cualquiera de los dos es negativo, también desaparece.

Note que la condición usada resume que ambos deben ser positivos y que *a* debe ser mayor que *b* para que la variable *mostrarrojo* valga 1. Por cierto, note que las asignaciones, dado que ambas variables sólo pueden llevar valor 0 ó 1, bien podrían ser:

```
mostraramarillo=(a!=b)
```

```
mostrarrojo=(a>b)&(b>0)
```

Es decir, la asignación de valores de la condicional como tal es la misma que si se incluye el $?1:0$, por lo que se puede omitir. De hacerlo así, el interactivo seguirá comportándose igual.

8. Recordamos que las instrucciones dadas en el algoritmo *CALCULOS* se hacen cada vez que el usuario hace **cualquier cosa** en la escena. Por ejemplo, pulsar el espacio, moverlo, y muchas otras que no competen directamente al funcionamiento de los pulsadores. Así, sería buen restringir dichas asignaciones sólo a los pulsadores. Ello se logra si se asigna la acción *calcular* a cada pulsador y ambas asignaciones se cortan de *CALCULOS* y se pegan en la ventana que se abre si se pulsa el botón a la derecha de *parámetro* para cada pulsador.

Tras aplicar los cambios, el interactivo no muestra los puntos de inicio. Es necesario pulsar cualquier a de los pulsadores para que aparezcan. Posterior a esto, los puntos siguen comportándose como antes. El que no se vean los puntos inicialmente corresponde a que *mostraramarillo* y *mostrarrojo* al principio no tienen valores (previamente se los asignaba *CALCULOS*, dado que ese algoritmo se realiza también al cargar el interactivo. Al ya no estar las asignaciones de valor a estas variables en *CALCULOS*, la primera vez que se hacen es hasta que se usa un pulsador.

9. Se pueden copiar ambos renglones de las asignaciones al algoritmo *INICIO* con el fin de que se realicen una vez al empezar el interactivo. Con ello se logra que aparezcan los puntos desde un

principio, y se mantiene la optimización del interactivo.

NOTA: Si usó la escena *Conds_Escena.html* para revisar este ejercicio, notará que en el algoritmo *INICIO* de la misma hay tres líneas que no corresponden a nada hecho en este ejercicio. Puede ignorarlas, debido a que dichas líneas responden sólo a comunicación entre la escena y el contenedor *index.html* para su correcto despliegue en pantalla.