

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

1. En una nueva escena de Descartes agregue un vector con 20 entradas con identificador *Dts* (de datos) y deje su panel de introducción de datos vacío. Guarde su interactivo html en una carpeta de su preferencia y con el nombre que guste.
2. Haga una función con nombre *AsignaValores()*, sin argumentos, de tipo algoritmo, que no devuelva un valor, y que se encargue de dar valores aleatorios a las 20 entradas del vector. Cada valor debe ir entre 0 y 10 en los reales.
3. Agregue un botón con nombre *Generar datos* e identificador *GenBt*, que ejecute la función como parte de su acción.
4. Agregue un gráfico tipo texto. Haga que se comporte como una familia de renglones. Cada renglón habrá de mostrar el valor de cada una de las entradas del vector.
5. Agregue una función algorítmica *CalcProm()*, que no tenga argumentos y que encuentre el promedio de los valores.
6. Haga que un texto despliegue el promedio.
7. Haga que la acción del botón también asigne a la variable *promedio* lo que devuelve la función *CalcProm()*.
8. Repita todo el proceso de los últimos pasos (desde crear la nueva función hasta desplegar el valor) pero en vez de hacerlo para una función que calcule el promedio, hágalo para una función *CalcDsvSt()* que calcula la desviación estándar de los datos.
9. Oprima el botón *Definiciones* hasta arriba del panel mostrado en el selector *Definiciones* para mostrar la ventana con el código fuente correspondiente a este selector. **Corte** las dos últimas líneas, que corresponden a los cálculos del promedio y la desviación estándar. Acepte los cambios para cerrar la ventana emergente. Pegue el texto cortado en un archivo de texto y guárdelo con el nombre *motor* en una carpeta *biblioteca* a la altura del html del interactivo. De permitirlo su editor de texto, guarde el archivo motor en el editor de texto bajo codificación UTF-8. Si se encuentra usando un editor de texto que haga la distinción entre *UTF-8* y *UTF-8 sin BOM* (como, por ejemplo, *Notepad++*), elija la opción *UTF-8 sin BOM*.
10. Agregue una definición tipo *biblioteca*. En su campo *archivo* agregue la ruta al archivo *motor* (o *motor.txt* en caso que no haya podido quitar la extensión). Aplique los cambios y guarde el interactivo.
11. Abra su archivo html del interactivo con un editor de texto.
12. Actualmente, si visita el selector *Definiciones*, el menú debajo del mismo muestra el símbolo *. Es decir, no está mostrando el **contenido** de una biblioteca en particular. Despliegue el contenido de dicho menú y seleccione la opción *biblioteca/motor*, que es la biblioteca que recién guardó como archivo de texto.

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

1. Es necesario pulsar el botón + sobre el panel en el selector *Definiciones*. En la ventana emergente hay que seleccionar *vector* como tipo de definición y se puede introducir el identificador directamente ahí. Una vez agregado el vector, su campo *tamaño* del vector debe ajustarse a 20. Se elimina el texto que trae por defecto el vector en el panel de introducción.
Tras aplicar los cambios, aún no se visualiza nada nuevo en el editor. El interactivo queda guardado con el nombre y ubicación decidida por el usuario.
2. Se debe pulsar el botón + sobre el panel del selector *Definiciones*. En el menú la ventana emergente se selecciona *función* y en la misma ventana se le asigna el nombre deseado. Una vez creada la función, se limpia el texto a la derecha del signo igual y se marca el checkbox *algoritmo*.
En el campo *inicio* se inicializa el contador *i* en cero, mismo que se usará para cada una de las 20 entradas del vector:

```
i=0
```

En el panel *hacer* se introduce el siguiente texto, el cual contiene las instrucciones para asignar los valores a una entrada determinada:

```
Dts[i]=10*rnd
```

```
i=i+1
```

Observe que el contador ha de aumentar en valor una unidad cada vez que se completa un ciclo. En el campo *mientras* se incluye la condición en que se indica si la función ha de repetirse o no:

```
i<20
```

Se considera dicha condición ya que el vector tiene 20 entradas.

Tras aplicar los cambios, no se observan cambios en el interactivo.

3. Es preciso oprimir el botón + sobre el panel en el selector *Controles*. En la ventana emergente, hay que seleccionar la opción *botón* del menú y darle el identificador deseado.

Una vez añadido el control, se cambia el *nombre* de *GenBt* a *Genera datos*, se selecciona *calcular* como su *acción* y en el *parámetro* de la acción se introduce *AsignaValores()*.

Tras aplicar los cambios, el botón aparece en el sur de la escena, pero aún no se visualizan cambios al oprimirlo.

4. Se pulsa el botón + sobre el panel del selector *Gráficos*. En el menú de la ventana emergente se selecciona *texto*.

Una vez agregado, su campo *expresión* se cambia por $(20, 20+16*s)$ para que cada nuevo renglón aparezca 16 px debajo del anterior. Se marca el checkbox *familia*, el *intervalo* se cambia a $(0, 19)$ (con el objeto de que contenga 20 entradas) y el campo *pasos* se cambia por 19. El campo *texto* debe contener:

```
Dato [s] = [Dts[s]]
```

Tras aplicar los cambios, el texto se despliega en el interactivo. Todos los datos empiezan con cero, que es de esperarse. No obstante, al presionar el botón, se llenan los datos con valores distintos de cero.

5. Se agrega la función en el selector *Definiciones* con el nombre indicado. La función debe llevar un contador. Se puede usar el mismo contador *i* que para la función que genera los datos. Dado que no hay posibilidad de que la función se ejecute simultáneamente o dentro de la que genera los datos, el contador puede llamarse igual. Así, el campo *inicio* debe contener:

```
i=0; suma=0, ya que se requerirá tener también una variable suma, que suma los datos del vector, y que ha de estar inicializada en 0 antes de calcular un promedio.
```

El panel *hacer* de la función *CalcProm()* debe contener:

```
suma=suma+Dts[i]
```

```
i=i+1
```

Note que, en cada realización del ciclo, a la variable *suma* se le agrega el valor del *i*-ésimo dato del vector *Dts*.

La condición en *mientras* de la función sigue siendo la misma (recorrer el ceroésimo dato al decimonoveno).

El valor a devolver, que se especifica a la derecha del signo igual, debe contener la división de la suma entre el número de datos. Es decir:

```
suma/20
```

Tras aplicar los cambios, no aparece cambio alguno en el funcionamiento del interactivo, dado que no se ha agregado un texto que muestre el valor del promedio.

6. Se agrega un nuevo gráfico tipo *texto*. Su expresión puede ajustarse a $(300, 20)$ para que no se encime con los datos. El texto a mostrar es *El promedio es [promedio]*.

Tras aplicar los cambios, el nuevo texto aparece, pero siempre muestra el valor de cero pues la variable *promedio* no tiene aún un valor asignado.

7. Se pulsa el botón a la derecha del campo *parámetro* en el control tipo *botón*. Ello lanza una ventana en la que se puede introducir la nueva instrucción. La acción para el botón queda, así pues:

```
AsignaValores()
```

```
promedio=CalcProm()
```

Tras aplicar los cambios, cada que se oprime el botón para asignar datos, se generan nuevos datos en el vector y el promedio se actualiza dependiendo de éstos.

8. Se deja al lector entender la razón de la estrategia presentada a continuación:

La nueva función debe devolver:

```
sqrt (suma/20)
```

Su campo *inicio* debe llevar:

```
i=0; suma=0
```

Su panel *hacer* debe llevar:

```
suma=suma+(promedio-Dts[i])^2
```

```
i=i+1
```

Su campo *mientras* debe llevar:

```
i<20
```

Las instrucciones en el parámetro del botón deben ahora contener:

```
AsignaValores()
```

```
promedio=CalcProm()
```

```
dsvst=CalcDsvSt()
```

El *texto* en el gráfico debe ahora contener:

```
El promedio es [promedio].
```

```
La desviación estándar es [dsvst].
```

Tras aplicar los cambios, cada que se aprieta el botón se generan nuevos datos. El texto al centro muestra el promedio y la desviación estándar de los datos actuales.

9. Con un editor de texto se genera un archivo *motor* en la carpeta *biblioteca* creada a la altura de donde se guarde el interactivo. Los datos son cortados y pegados en el archivo *motor* y éste es guardado. Note que, por comodidad, éste archivo no debe llevar extensión, sólo el nombre *motor*. Puede eliminar la extensión *txt* manualmente. De no serle posible, deje dicha extensión. Tras aplicar los cambios y presionar el botón para generar los datos, los textos para el promedio y desviación estándar muestran siempre el valor de cero, dado que cortó las funciones relacionadas a sus respectivos cálculos y ya no existen.

10. El campo *archivo* de la biblioteca creada en *Definiciones* debe llevar el texto *biblioteca/motor* (o *biblioteca/motor.txt* en su defecto).

Tras aplicar los cambios, el interactivo sigue funcionando bien a pesar de no tener explícitamente las funciones *CalcProm()* y *CalcDsvSt()* en su interior.

11. El editor de texto muestra un bloque de script casi al final del documento que hace referencia al motor. El motor ha sido embebido en el html y contiene los detalles de las dos funciones que contiene el motor, pero no se muestra en el editor de Descartes como tal. Aún así, al estar embebido, la escena sigue funcionando como se espera.

12. Cuando se selecciona *biblioteca/motor* dentro del menú en el selector *Definiciones*, aparecen en el panel izquierdo del selector las funciones *CalcProm()* y *CalcDsvSt()*. Dependiendo de la función que se visite, se muestra la función en el editor de configuraciones tal de la misma forma en que se ve cuando se edita directamente fuera de la biblioteca.

Cualquier cambio hecho en dichas funciones dentro de la biblioteca *motor* en el editor de configuraciones, si se aplica y guarda la escena, dicho cambio se aplicará tanto en el archivo *motor* (o *motor.txt* en su defecto) como en el código del bloque de script en el archivo html.