

Comencemos a programar con  
**VBA - Access**

Entrega **02**

Entorno de desarrollo

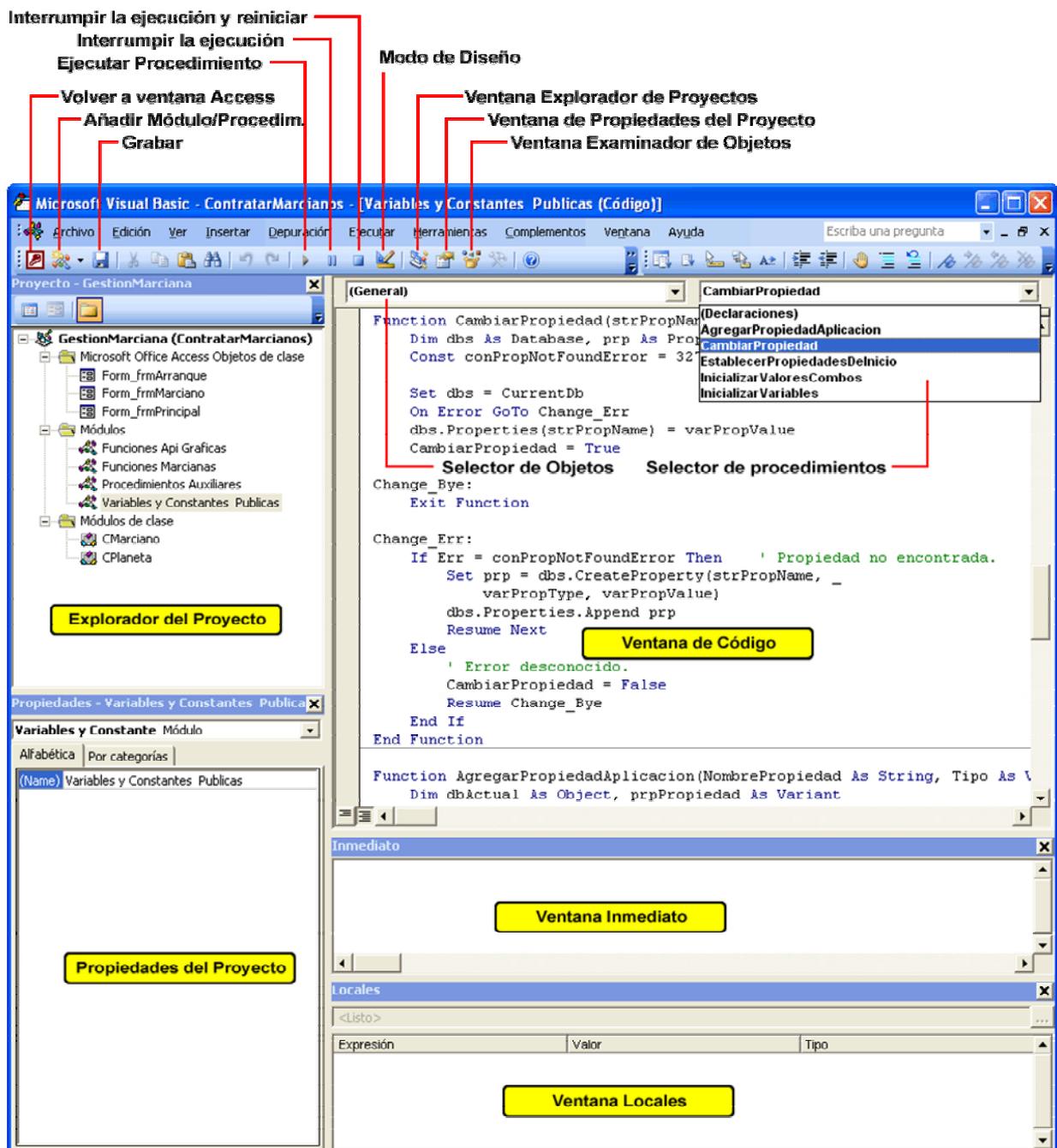
## Entorno de desarrollo

Ya hemos comentado que VBA para Access tiene un entorno de desarrollo similar al de otras aplicaciones con VBA.

Este entorno recibe el nombre de IDE, que significa algo así como Entorno de Desarrollo Integrado. En el "idioma bárbaro" se escribe Integrated Development Environment.

Este entorno consta de una serie de ventanas y botones, algunos de los cuales se muestran en el siguiente gráfico.

### Ventanas y botones del Editor de código VBA



## Las ventanas, por el momento más importantes para nosotros, son

- **Ventana de Código**

Es la ventana en la que se escribe todo el código que va a gestionar nuestra aplicación.

Podemos seleccionar el módulo que se mostrará en cada momento, por ejemplo seleccionándolo con un doble clic en la ventana del Explorador del Proyecto.

- **Ventana del Explorador del Proyecto**

La ventana del proyecto nos muestra todos los objetos que poseen módulos.

Entre ellos tenemos los módulos de Formularios e Informes, los módulos estándar, y los módulos de clase.

En esta primera parte del cursillo vamos a trabajar con los módulos asociados a formularios y los módulos estándar.

Si en la ventana del proyecto hacemos doble clic sobre cualquiera de los elementos que aparecen, abriremos su módulo correspondiente.

Por ejemplo, si en la base de datos correspondiente al gráfico presionáramos (doble clic) el formulario **Form\_frmMarciano** nos abriría el módulo asociado al formulario frmMarciano.

Si presionamos sobre **Funciones Marcianas** nos abriría el módulo estándar Funciones Marcianas y si presionáramos sobre **CMarciano**, nos abriría el módulo de clase CMarciano.

Fijaros que el icono asociado a cada módulo cambia en función del tipo que sea.

- **Ventana Inmediato**

Esta es una ventana desde la que podemos ordenar que se ejecuten determinadas órdenes, o imprimir desde el código determinados valores para efectuar comprobaciones. Recibe también el nombre de **Ventana de Depuración**.

Su mayor utilidad es para la depuración del código, es decir para efectuar comprobaciones de errores. De hecho es un objeto que recibe el nombre de **Debug**.

- **Ventana Locales**

Esta es una ventana que también sirve para la comprobación y el seguimiento de los valores que van tomando determinados parámetros, variables, etc. durante la ejecución del código.

Por tanto comparte la función de depuración con la ventana Inmediato.

- **Ventana Propiedades**

En esta ventana podemos ver las propiedades de los diferentes objetos que intervienen en el proyecto, al estilo de la ventana de propiedades del Formulario ó Informe en el modo diseño.

Los módulos, como Objetos de Access, también tienen sus propiedades.

Existen otras ventanas no comentadas aquí, pero que no dejan de ser importantes. Entre ellas podemos destacar la Ventana del **Examinador de Objetos**, o la Ventana de **Inspección**, de las que hablaremos más adelante.

## Uso de la ventana Inmediato

Antes de nada vamos a hacer lo siguiente:

Creamos una nueva base de datos a la que llamamos Entrega02.mdb

Vamos a Módulos y presionamos en el botón [Nuevo].

Nos creará un módulo con el original nombre de **Módulo1**.

Vamos a examinar las ventanas que se abren y las comparamos con el gráfico de la página anterior 02-2.

¿Identificas lo que se ha explicado en los puntos anteriores?

Si por un casual no apareciera la ventana Inmediato, haz lo siguiente:

Vete al menú Ver y presiona en la opción [Ventana Inmediato]; o más fácil aún, presiona la combinación de teclas [Ctrl] + [G].

Vamos a mantener las ventanas de **Código**, la de **Inmediato** y la del **Explorador del proyecto**. Las otras, si te molestan, puedes cerrarlas. Más tarde las podrás abrir con los botones o las opciones de menú correspondientes (Menú Ver > y seleccionar su opción).

¿Para qué sirve la ventana Inmediato?

Vamos a hacer un ejemplo.

Pincha con el cursor en ella, escribe lo siguiente:

```
Print 4 + 5
```

y dale a la tecla [Enter]

Y ¡oh sorpresa! Escribe 9 en la siguiente línea.

Ahora escribe ? 4 + 5 más [Enter]

Y vuelve a escribir la cifra 9

Escribe ? 4 \* 5 más [Enter]

¿Adivinas qué va a escribir? ¡Premio! 20

Ahora vamos a hacerlo un poco más complicado

```
a = 2 : b = 4 : Print a * b
```

Pues sí, te escribe 8 en la siguiente línea.

¿Qué pasará si escribo ? a^b y aprieto [Enter]?

¡Sorpresa! Ha escrito 16 que es 2 elevado a 4

Si ahora presiono la tecla de la barra de menú que tiene la forma de un cuadradito azul y que en el gráfico la identifico con Interrumpir la ejecución y reiniciar, si vuelvo a escribir

```
? a^b
```

como resultado me da 1.

Pensemos un poco sobre estos sorprendentes resultados.

Tras la primera línea

```
a = 2 : b = 4 : Print a * b
```

parece que Access recuerda que existe “algo” llamado **a**, que se le ha dado el valor 2 y algo llamado **b** que tiene el valor 4, como se demuestra tras el resultado de escribir la segunda vez ? a^b

Pero resulta que tras presionar el botón de Reiniciar (el cuadradito azul) resulta que Access se equivoca y da como resultado **1**.

Pues no, Access no se ha equivocado. El exponente **b** tiene el valor **0**, y cualquier número elevado al exponente **0** da como resultado **1**.

En realidad éste no es un gran ejemplo, ya que estaríamos en el caso singular de 0 elevado a cero, que no está definido en el campo de los números reales.

Pero no vamos a ser más “papistas que el Papa”.

Vamos a hacer ahora otro experimento.

Escribimos en la ventana Inmediato:

```
a = "2" : b = "4" : ? a + b
```

Tras presionar la tecla Enter el resultado es ¡chan, chan....!

Si se te ha ocurrido que sería 6 te puedo decir que estás equivocado.

Si tu respuesta ha sido 24, has acertado; pero yo me pregunto ¿No estás perdiendo el tiempo leyendo este capítulo? ¡so listo...!

Bromas aparte expliquemos el resultado.

Como introducción hemos usado unos elementos que se llaman **Operadores**.

Los que hemos usado por ahora son:

- + Suma
- \* Producto
- ^ Exponenciación

El símbolo **:** no es el operador de división. En realidad es un simple separador de instrucciones. El operador de división es **/**.

Hay otros muchos más que ya iremos viendo poco a poco.

¿Por qué al escribir `a = "2" : b = "4" : ? a + b`, nos da **24**?

Vamos a fijarnos en estos pequeños detalles `a = "2"` y `b = "4"`

La cifra **2** está entre 2 comillas, lo mismo que la cifra **4**.

Esto hace que VBA considere el 2 y el 4, no como números, sino como texto, también llamado cadena.

Tras esto el operador **+** no efectúa una suma numérica, sino que enlaza dos cadenas de texto. Si enlazamos la cadena `"2"` con la cadena `"4"`, nos da la cadena `"24"`.

Para los “listos” que ya se sabían el resultado, el resto que no haga ni caso de este párrafo, les informo que esto, en lenguajes más avanzados, se llama Sobrecarga de Operadores, lo que equivale a decir que el operador **+** está **Sobrecargado**.

En resumidas cuentas, que con números hace la operación de **Suma numérica** y con cadenas de texto la operación de **enlazar cadenas**.

Vamos a seguir con “pequeñas sorpresas”.

Escribamos algo tan inocente como:

```
Print 2,4 * 2
```

¡Sorpresa!. ¿Qué ha pasado?

Que nos imprime

2

8

¿Por qué no nos ha dado como resultado 4,8 que es lo que le daría a un buen cristiano, judío ó musulmán, (o lo que sea siempre que sea bueno)?.

De toda la vida la operación 2,4 \* 2 ha dado como resultado 4,8

¡La culpa de esto la tienen que tener los americanos!

Y no es broma.

Sólo un pequeño matiz:

Los americanos utilizan la coma como separador de miles.

En cambio el separador decimal, en vez de la coma, es el punto.

En una sentencia Print, cuando encuentra una coma entre dos valores, intercala una tabulación.

Probemos ahora:

`Print 2.4 * 2` y nos da 4,8

¡Nueva sorpresa!

-Ahora lo hace bien pero ¿no hemos quedado que el separador decimal es el punto?-.

-¿Por qué ahora me muestra el resultado decimal con una coma como separador?-.

-Cosas de Windows-.

Como sabe que al otro lado de la pantalla hay un hispano, le muestra el resultado "en cristiano"; Excepto para algunos PCs. De México, Florida y otras "colonias".

-Borro lo de colonias-

-¿En qué estaría yo pensando?-

La pregunta ahora es

-¿Y cómo lo sabe?-.

-Fácil; para eso está la Configuración Regional de Windows-.

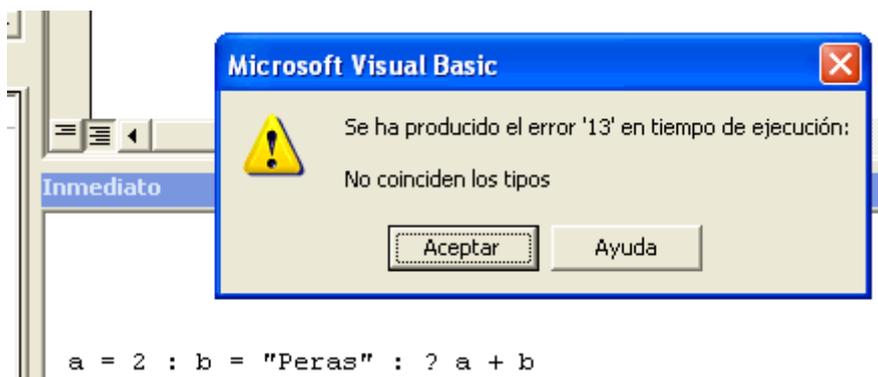
Vamos a dejar de divagar y sigamos experimentando.

Escribimos en la ventana Inmediato lo siguiente:

`a = 2 : b = "Peras" : Print a + b`

-¡No salimos de sustos!-

-¿Qué he roto?-



Tranquilo que no has roto nada.

¿No nos enseñaron en el colegio que no había que sumar Peras con Manzanas?

Le estamos diciendo a VBA que sume un número con un texto, y lógicamente se queja.

Cambiamos la línea

```
a = "2" : b = " Peras" : ? a + b
```

Ahora sí imprime bien

```
2 Peras
```

Al poner el 2 entre comillas, lo hemos cambiado al modo texto y entonces ha funcionado.

Si hacemos

```
a = "Dos" : b = " Peras" : ? a + b
```

Nos imprimirá

```
Dos Peras
```

-A ver profe si me aclaro-

Tenemos “algo” llamado **a** que le da lo mismo ser **Número** que **Texto**.

¿Cómo se come esto?

Tranquilo “pequeño saltamontes”. No seas impaciente. La respuesta en las próximas entregas.