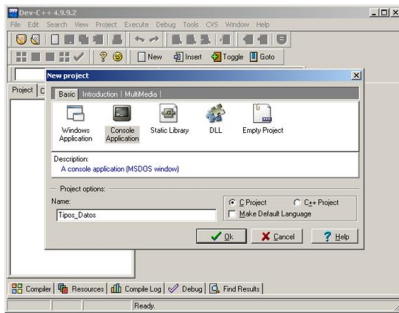


Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00

There are no translations available.



El primer objetivo de un programa informático es el manejo de datos. Un dato es toda aquella información que se puede tratar en un programa informático.

En este artículo vamos a conocer los tipos de datos que podemos manejar programando en C.

Existen unos datos de entrada y unos datos de salida.

Los datos de entrada se manipulan en el programa produciendo unos datos de salida.



Un dato dentro de un programa se caracteriza por llevar asociado un identificador, un tipo y un valor.

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00

- Identificador: Nombre para referenciar al dato dentro del programa
- Tipo: el tipo de un dato determina el rango de valores que puede tomar el dato y su ocupación en memoria durante la ejecución del programa
- Valor: Será un elemento determinado dentro del rango de valores permitidos por el tipo de dato definido.

Algunos ejemplos de datos son: la edad, el saldo de una cuenta bancaria, el nombre de una persona, la letra del piso de una dirección, etc.

A continuación vamos a describir los distintos tipos de datos que existen.

TIPOS DE DATOS

Hay dos clases de tipos de datos: tipos fundamentales y tipos derivados.

Únicamente vamos a ver los tipos de datos fundamentales.

Tipo entero: representa números enteros con o sin signo, que estarán compuestos por los dígitos del 0 al 9, pudiendo ser precedidos por los signos + o -.

Algunos ejemplo de datos enteros son: 0, 23, -176, -1, etc.

Para definir variables en C se antepone la palabra reservada del tipo al identificador de la variable. El tipo de datos entero se define en el lenguaje de programación C por la palabra reservada *int*.

Para definir un tipo de dato en C se escribe lo siguiente:

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00

```
int nombre_variable = valor;
```

No es necesario que la variable tenga un valor predeterminado. Se puede definir sin asignarle ningún valor.

Si tenemos varios datos que son del mismo tipo, se pueden definir todas en la misma línea de código escribiendo un único *int*, separando el nombre de las variables por “,”. Una vez que se haya acabado de definir variables, se cierra la línea de código con “;”,

Por ejemplo: `int edad = 24;`

```
int edad;
```

```
int edad, num, contador;
```

Tipo real: Se emplean para representar números reales (con decimales).

Para definir datos reales se antepone la palabra reservada *float* al identificador de la variable.

```
float identificador = valor;
```

Por ejemplo: `float numero1, numero2;`

```
float numero3 = 123.43;
```

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00

```
float numero3;
```

Tipo carácter: Este tipo de datos se emplea para representar un carácter perteneciente a un determinado código utilizado por el ordenador (normalmente el código ASCII).

Para representar este tipo de dato se antepone la palabra reservada *char* al identificador de la variable.

```
Char identificador = 'valor';
```

Una constante tipo char se representa como un solo carácter encerrado entre comillas simples.

```
Por ejemplo: char letra, letra2;
```

```
char letra='a';
```

Tipo cadena de caracteres: una cadena de caracteres es un número de caracteres consecutivos (incluso ninguno) encerrado entre unos delimitadores determinados, que en el lenguaje C son las comillas dobles.

Para definir variables de tipo cadena, estas se definen como vectores de caracteres, esto es, anteponiendo la palabra reservada *char* al identificador de la variable, y después entre corchetes la longitud máxima de cadena.

```
Char identificador[cantidad] = " mensaje ";
```

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00

Por ejemplo: `char cadena[20];`

`char cadena[20] = "Hola mundo";`

`char cadena[] = "HOLA";`

En la siguiente tabla se hace un resumen de los distintos tipos de datos:

Tipo de dato

Palabra reservada

Ejemplo

Entero

Int

Int numero=0;

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00

Real

Float

Float numero=12.2;

Carácter

Char

Char letra = 'a';

Cadena de carácter

Char

Char palabra[10] = "HOLA";

Lectura y Escritura de Datos

Para poder leer y escribir datos en el lenguaje de programación C existen una serie de funciones agrupadas en un conjunto de librerías de código objeto, que constituyen la llamada biblioteca estándar del lenguaje. En el caso concreto de las funciones de entrada y salida (lectura y escritura), su archivo de cabecera es *stdio.h*.

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00

Existen varias funciones que realizan la entrada y salida de datos en el lenguaje de programación C, pero nos vamos a centrar solamente en dos de ellas: *printf()* y *scanf()*.

Funcion printf()

La función *printf()* sirve para escribir datos en la pantalla con un formato determinado.

El prototipo de esta función es la siguiente:

```
int printf(const char *formato, arg1, arg2, ..., argn)
```

donde:

formato: hace referencia a una cadena de caracteres que contiene información sobre el formato de salida.

Arg1, arg2, ..., argn: argumentos que representan los datos de salida

La cadena de control puede constar de hasta tres tipos de información:

- texto: que será mostrado por pantalla tal cual este escrito
- Secuencias de escape: secuencias reservadas propias del lenguaje. Van precedidas por `"` y las más importantes son `'n'` que introduce un salto de línea, y `'t'` que introduce un tabulador.

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00

- Especificadores de formato: son ciertos caracteres precedidos del carácter tanto por ciento '%'.

A continuación se muestra una tabla con estos especificadores de formato dependiendo del tipo de dato a los que representan:

Tipo de dato

Especificadores de formato

Comentario

Int

%d

Entero con signo

Float

%f

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00

Real con signo

Char

%c

carácter

Char [n]

%s

Cadena de caracteres

Ejemplos:

```
printf("n Valores: %d, %f", entero, real);
```

Salida por pantalla:

(Retorno de carro)Valores: 2, 2.3

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00

```
printf(“%st%dn%f”, concepto, num, coste);
```

Salida por pantalla:

Cremallera 1234

12.345

Función scanf()

La función *scanf()* se puede considerar de alguna manera como la inversa de la función *printf()*, pues sirve para introducir datos desde el teclado con un formato determinado.

El prototipo de esta función es el siguiente:

```
int scanf(const char *formato, arg1, arg2, ..., argn);
```

donde:

- **formato:** hace referencia a una cadena de caracteres (cadena de control) que contiene información sobre el formato de los datos que se van a introducir mediante el teclado.
- **arg1, arg2, ..., argn:** son los argumentos que representan los datos de entrada.

La cadena de control puede constar de:

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00

- Especificadores de formato, que son ciertos caracteres precedido por el carácter tanto por ciento '%'.
 - Caracteres de espacio
 - Otros caracteres no espacios, para entradas con formato

A continuación se muestra una tabla con estos especificadores de formato dependiendo del tipo de dato a los que representan:

Tipo de dato

Especificadores de formato

Comentario

Int

%d

Entero con signo

Float

%f

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00

Real con signo

Char

%c

carácter

Char [n]

%s

Cadena de caracteres

Ejemplos:

```
scanf("%d", numero);
```

```
scanf("%d %s %f", numero, &cadena, &real);
```

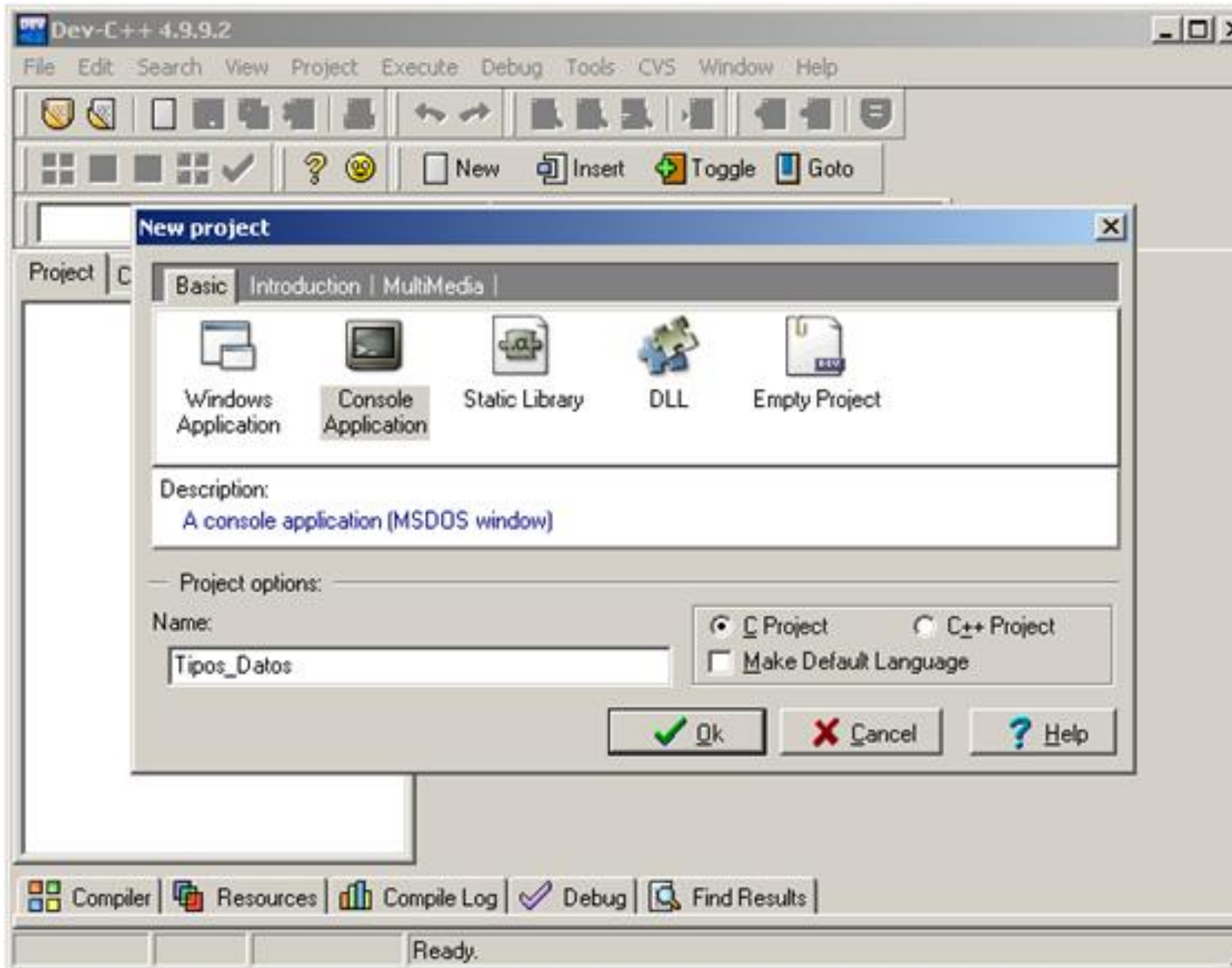
Programa Escritura por Pantalla

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00

Vamos a hacer un programa que muestre el valor de los distintos tipos de datos que hemos visto anteriormente. El valor de los datos lo vamos a dar ya definido en el código.

Lo primero que tenemos que hacer es crear un nuevo proyecto:

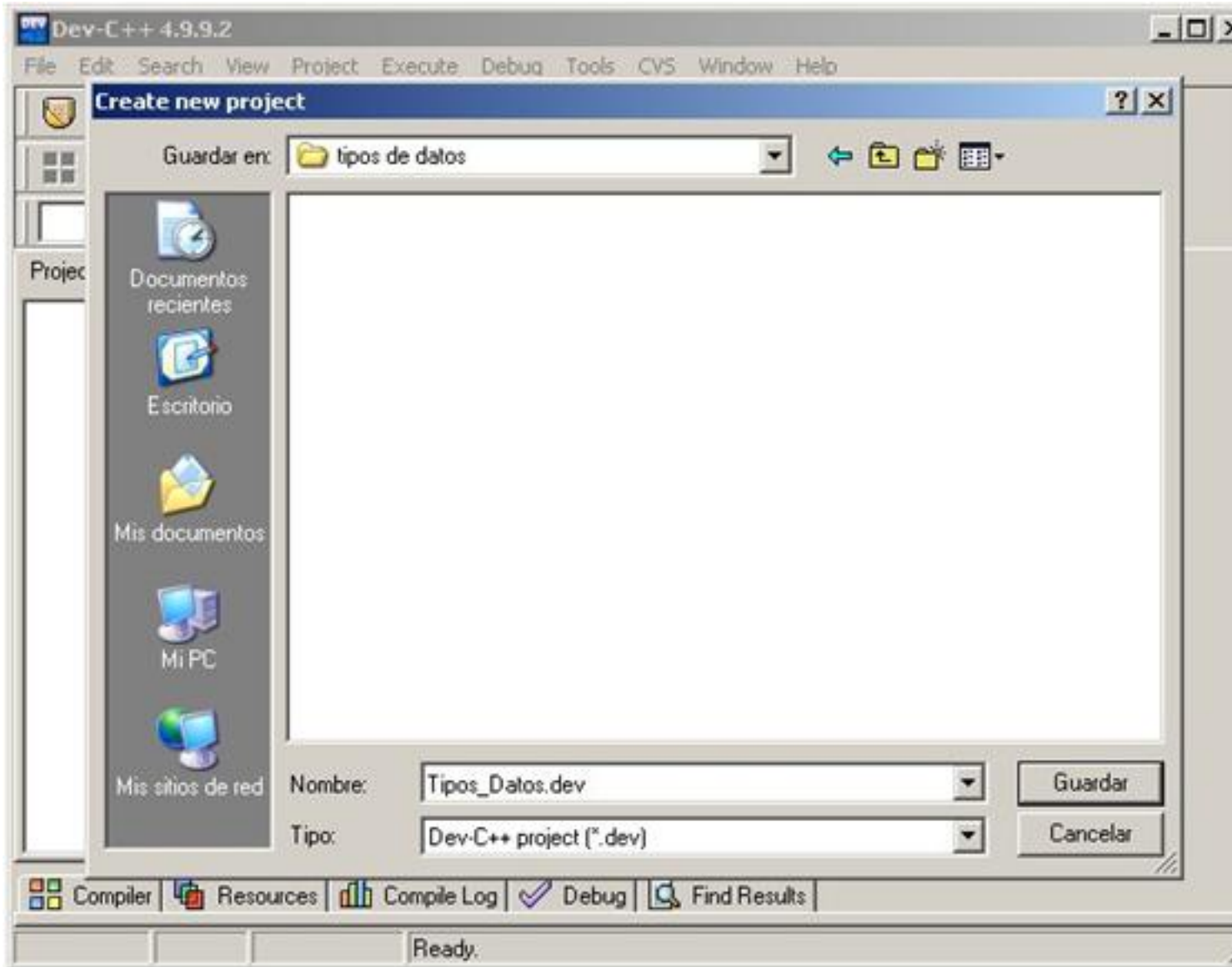


Creamos un nuevo proyecto que sea *Console Application*, seleccionamos C Project y le ponemos el nombre que queramos, que en este caso es:
Tipo_Datos

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00

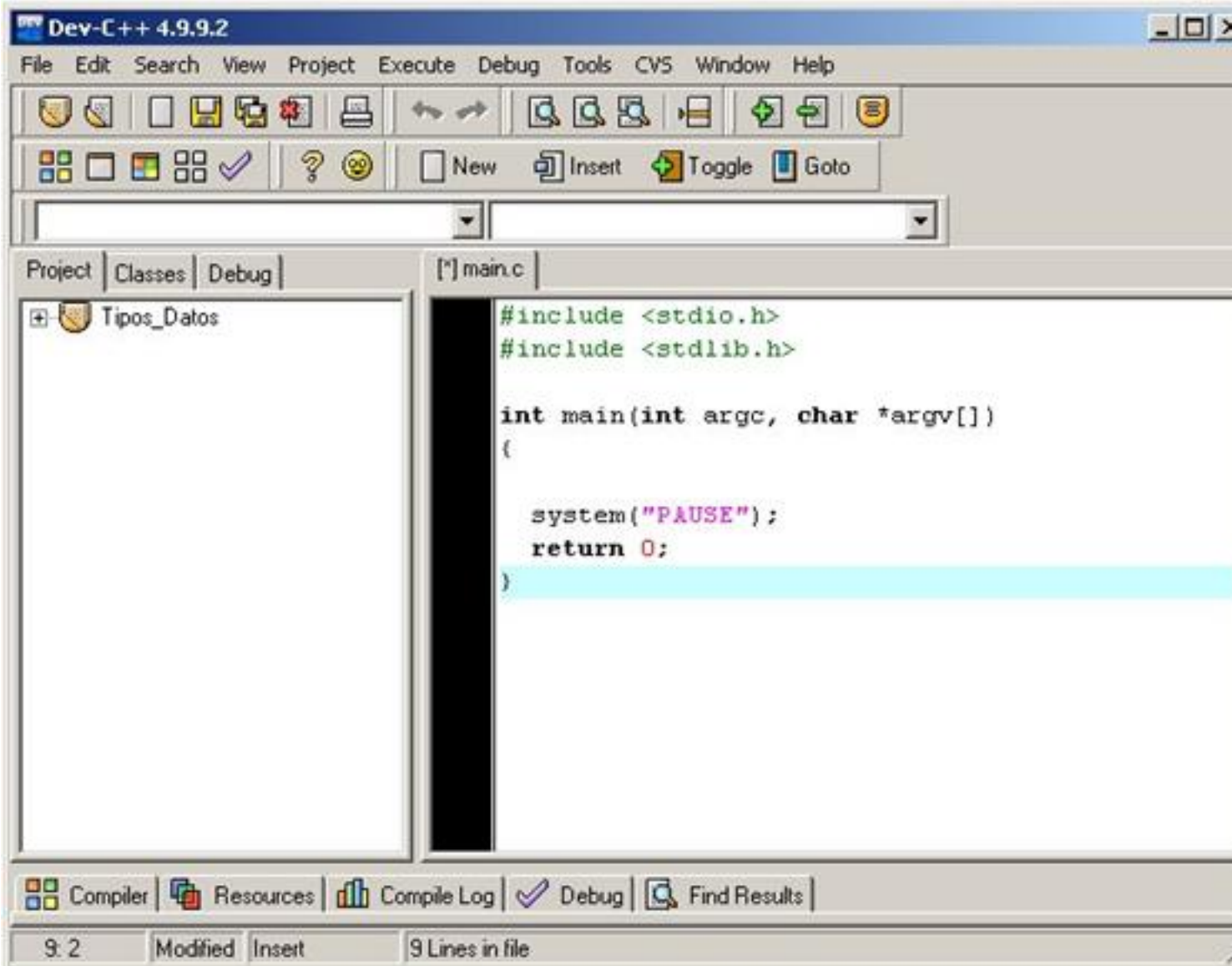
A continuación indicamos donde queremos guardar el proyecto que hemos creado.



Por defecto, Dev-C++ crea el archivo *main.c*, pero lo borramos ya que queremos aprender a programar desde el principio.

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00



The screenshot shows the Dev-C++ 4.9.9.2 IDE interface. The main window displays the following C code in the file `main.c`:

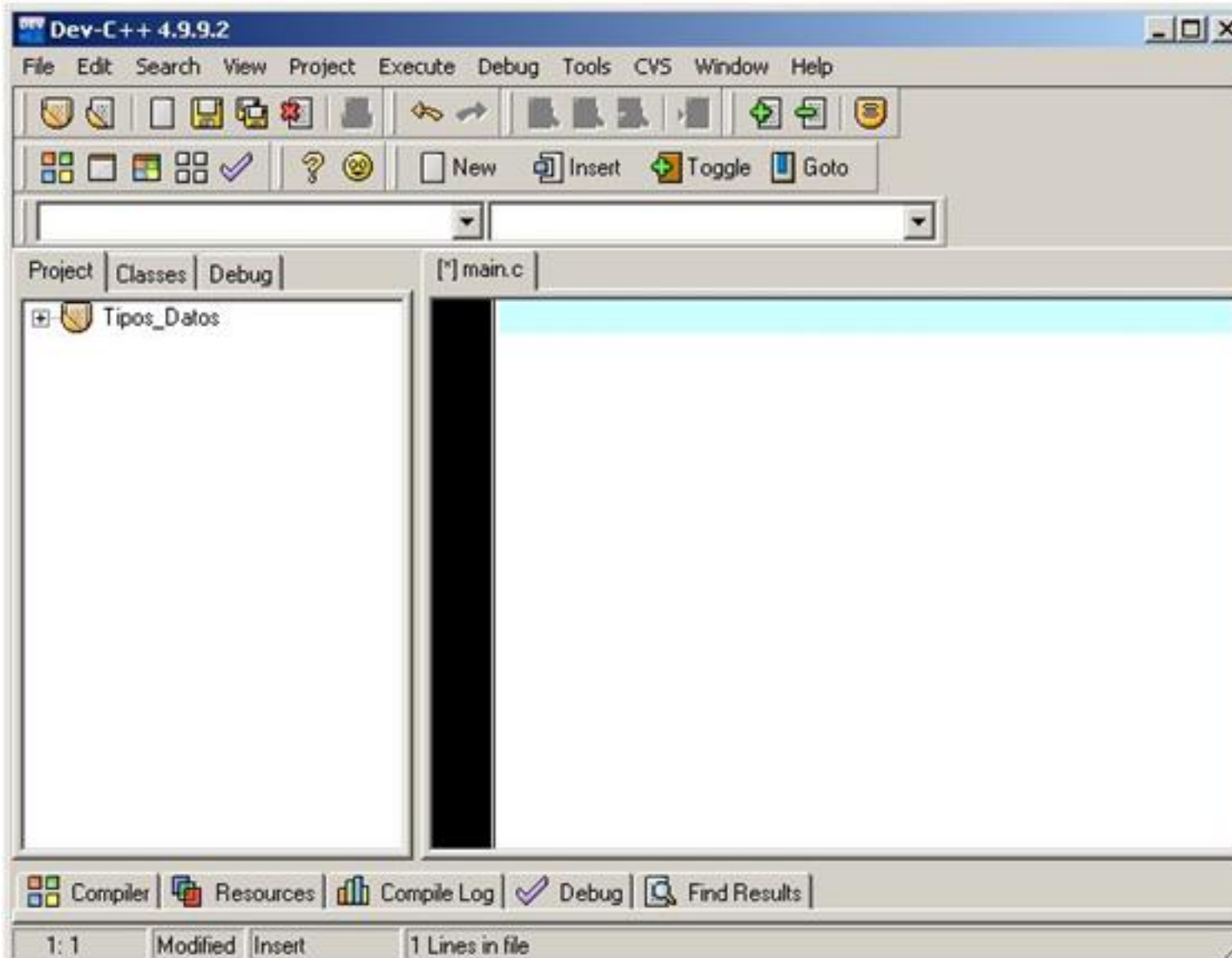
```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
(
    system("PAUSE");
    return 0;
)
```

The code is displayed in a monospaced font with syntax highlighting. The `system("PAUSE");` line is highlighted in light blue. The IDE's menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar contains various icons for file operations and development tools. The Project Explorer on the left shows a project named 'Tipos_Datos'. The status bar at the bottom indicates '9: 2 Modified Insert 9 Lines in file'.

Tipos de datos en C

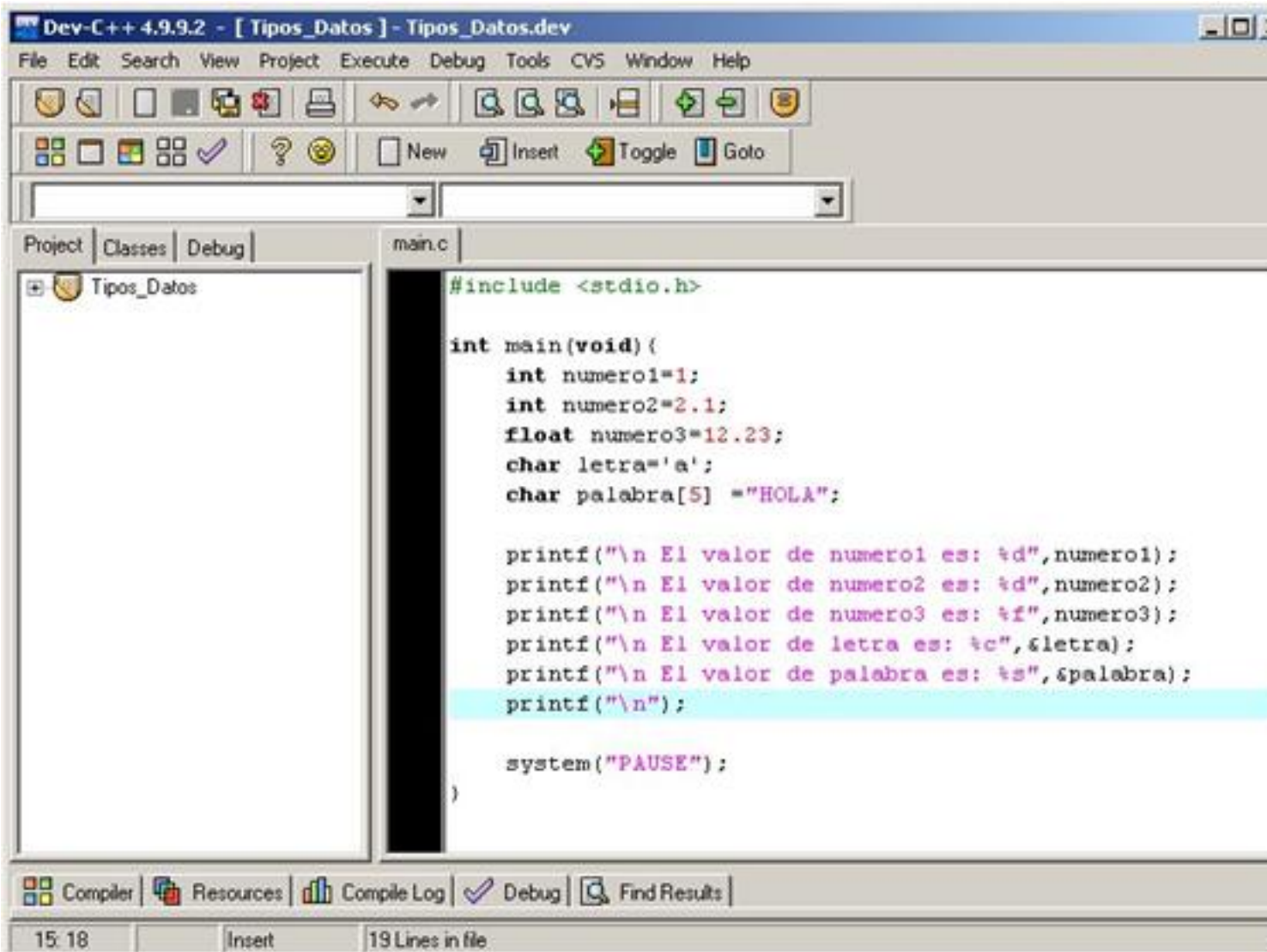
Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00



Algunos ejemplos de tipos de datos en C son los siguientes: `int` es el tipo entero, `float` es el tipo de punto flotante, `double` es el tipo de punto flotante de precisión doble, `char` es el tipo de carácter, `void` es el tipo de tipo vacío, `long` es el tipo de entero largo, `short` es el tipo de entero corto, `signed` es el tipo de entero con signo, `unsigned` es el tipo de entero sin signo, `signed char` es el tipo de carácter con signo, `unsigned char` es el tipo de carácter sin signo, `signed short` es el tipo de entero corto con signo, `unsigned short` es el tipo de entero corto sin signo, `signed long` es el tipo de entero largo con signo, `unsigned long` es el tipo de entero largo sin signo, `signed int` es el tipo de entero con signo, `unsigned int` es el tipo de entero sin signo, `signed float` es el tipo de punto flotante con signo, `unsigned float` es el tipo de punto flotante sin signo, `signed double` es el tipo de punto flotante de precisión doble con signo, `unsigned double` es el tipo de punto flotante de precisión doble sin signo, `signed char*` es el tipo de puntero a carácter con signo, `unsigned char*` es el tipo de puntero a carácter sin signo, `signed short*` es el tipo de puntero a entero corto con signo, `unsigned short*` es el tipo de puntero a entero corto sin signo, `signed long*` es el tipo de puntero a entero largo con signo, `unsigned long*` es el tipo de puntero a entero largo sin signo, `signed int*` es el tipo de puntero a entero con signo, `unsigned int*` es el tipo de puntero a entero sin signo, `signed float*` es el tipo de puntero a punto flotante con signo, `unsigned float*` es el tipo de puntero a punto flotante sin signo, `signed double*` es el tipo de puntero a punto flotante de precisión doble con signo, `unsigned double*` es el tipo de puntero a punto flotante de precisión doble sin signo, `signed char**` es el tipo de puntero a puntero a carácter con signo, `unsigned char**` es el tipo de puntero a puntero a carácter sin signo, `signed short**` es el tipo de puntero a puntero a entero corto con signo, `unsigned short**` es el tipo de puntero a puntero a entero corto sin signo, `signed long**` es el tipo de puntero a puntero a entero largo con signo, `unsigned long**` es el tipo de puntero a puntero a entero largo sin signo, `signed int**` es el tipo de puntero a puntero a entero con signo, `unsigned int**` es el tipo de puntero a puntero a entero sin signo, `signed float**` es el tipo de puntero a puntero a punto flotante con signo, `unsigned float**` es el tipo de puntero a puntero a punto flotante sin signo, `signed double**` es el tipo de puntero a puntero a punto flotante de precisión doble con signo, `unsigned double**` es el tipo de puntero a puntero a punto flotante de precisión doble sin signo.

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00



```
#include <stdio.h>

int main(void){
    int numero1=1;
    int numero2=2.1;
    float numero3=12.23;
    char letra='a';
    char palabra[5] ="HOLA";

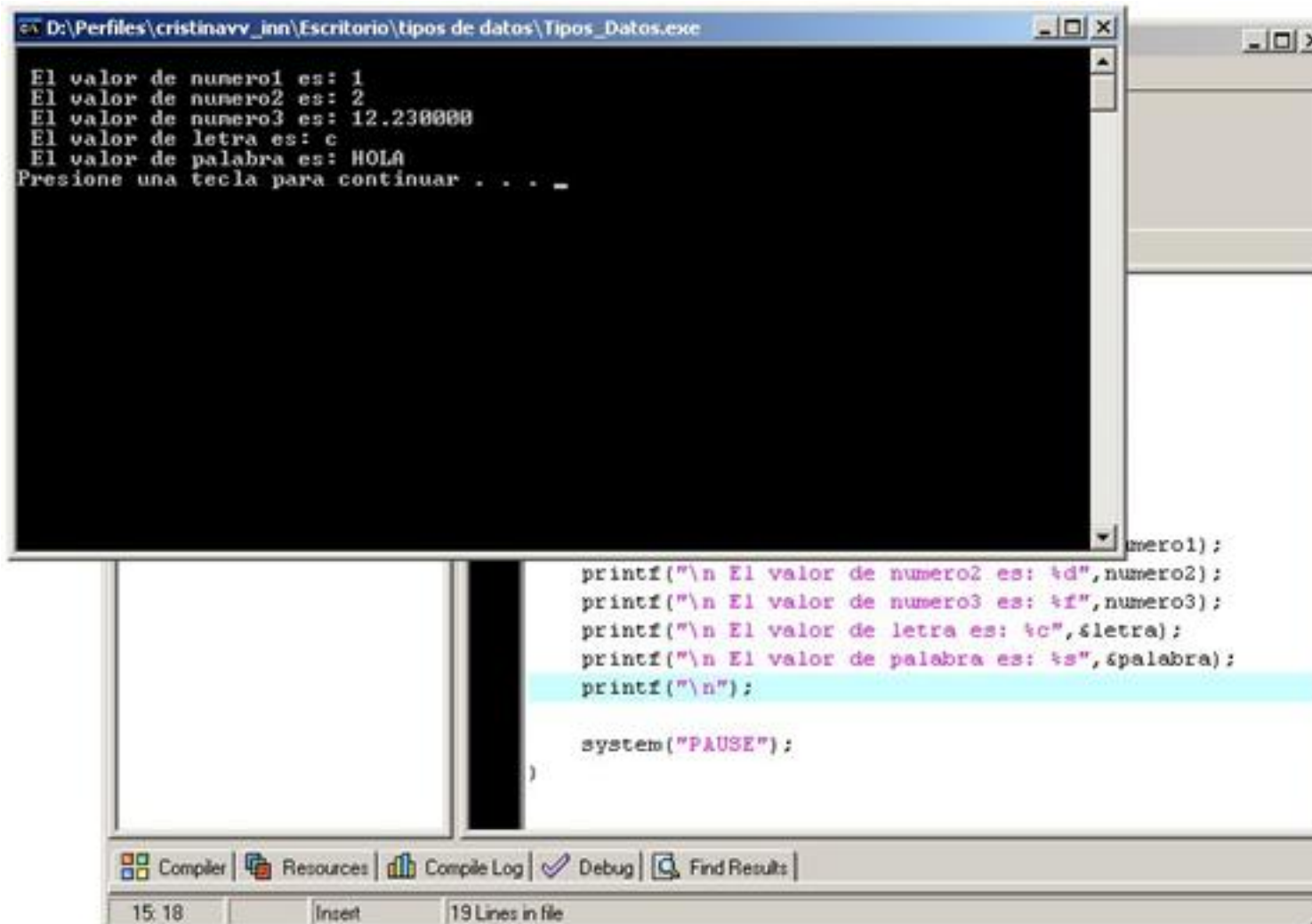
    printf("\n El valor de numero1 es: %d", numero1);
    printf("\n El valor de numero2 es: %d", numero2);
    printf("\n El valor de numero3 es: %f", numero3);
    printf("\n El valor de letra es: %c", &letra);
    printf("\n El valor de palabra es: %s", &palabra);
    printf("\n");

    system("PAUSE");
}
```

El resultado es el siguiente el programa para comprobar que todo es correcto.

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00



```
El valor de numero1 es: 1
El valor de numero2 es: 2
El valor de numero3 es: 12.238008
El valor de letra es: c
El valor de palabra es: HOLA
Presione una tecla para continuar . . . _

printf("\n El valor de numero2 es: %d", numero2);
printf("\n El valor de numero3 es: %f", numero3);
printf("\n El valor de letra es: %c", &letra);
printf("\n El valor de palabra es: %s", &palabra);
printf("\n");

system("PAUSE");
```

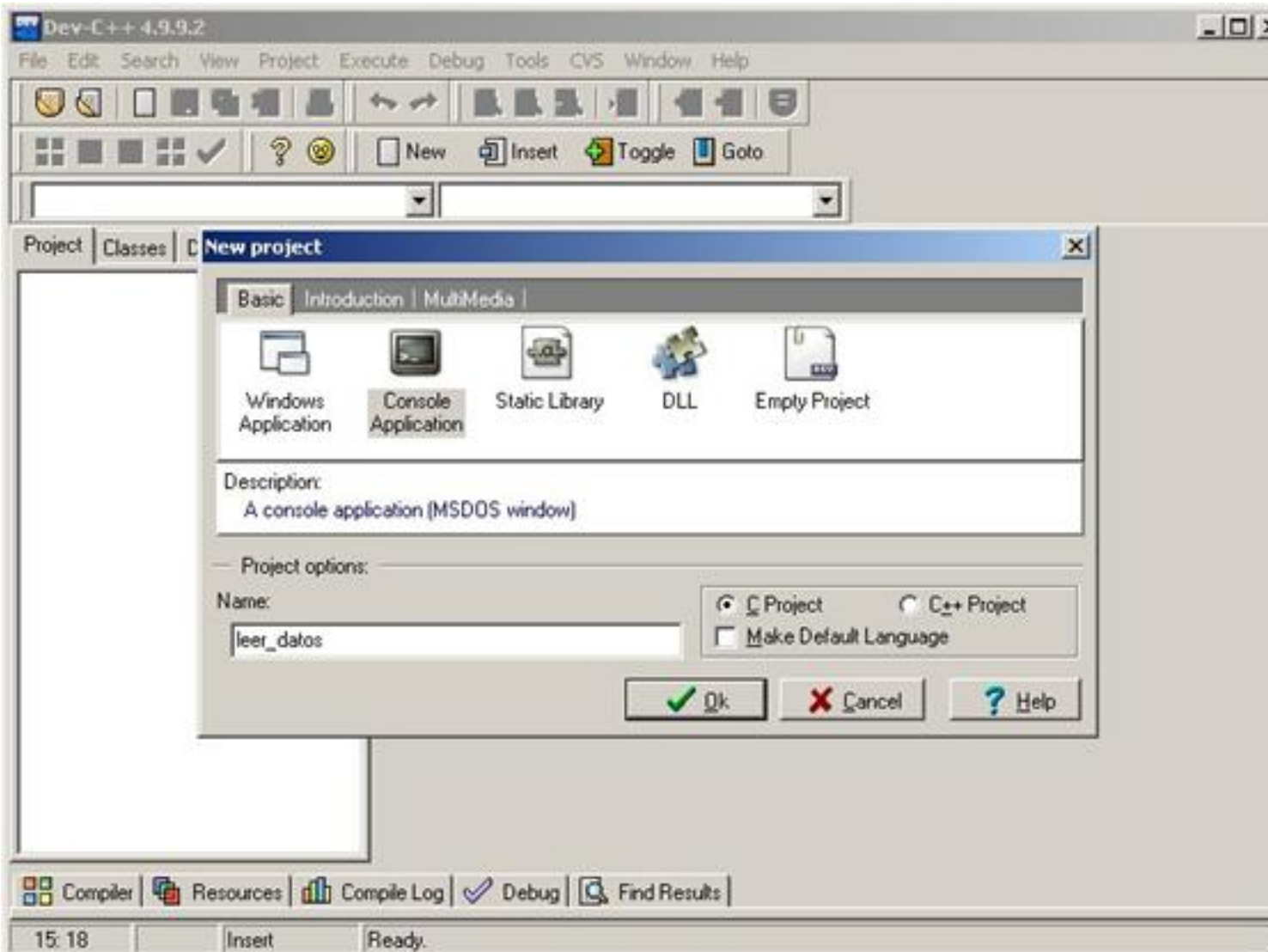
Programa de Lectura y Escritura de Datos

Vamos a hacer un programa que muestre el valor de los distintos tipos de datos que hemos visto anteriormente. El valor de los datos lo vamos a dar ya definido en el código.

Lo primero que tenemos que hacer es crear un nuevo proyecto:

Tipos de datos en C

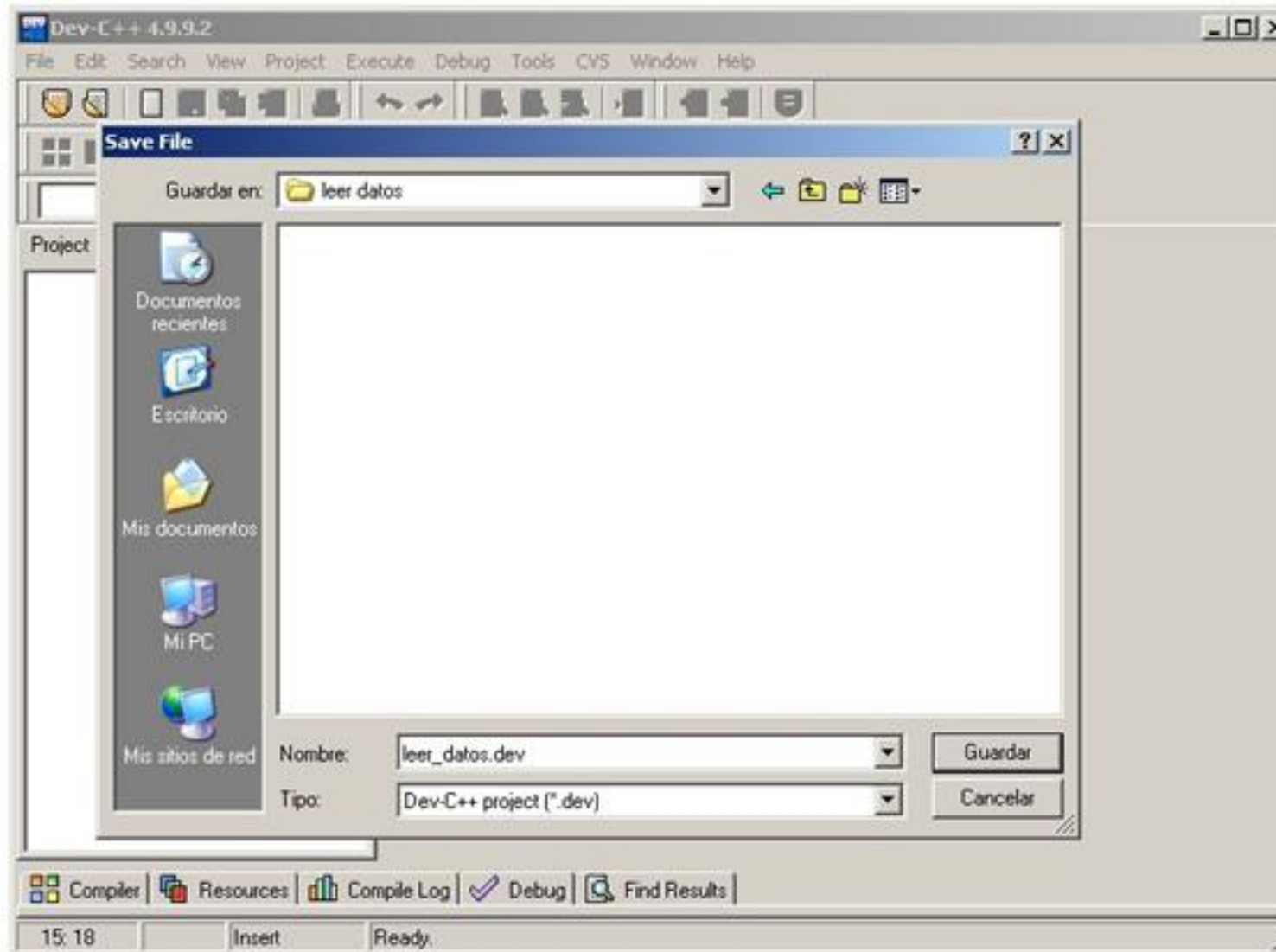
Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00



Antes de iniciar el proyecto que se creará se debe seleccionar el tipo de proyecto y seleccionar el lenguaje C++ y el

Tipos de datos en C

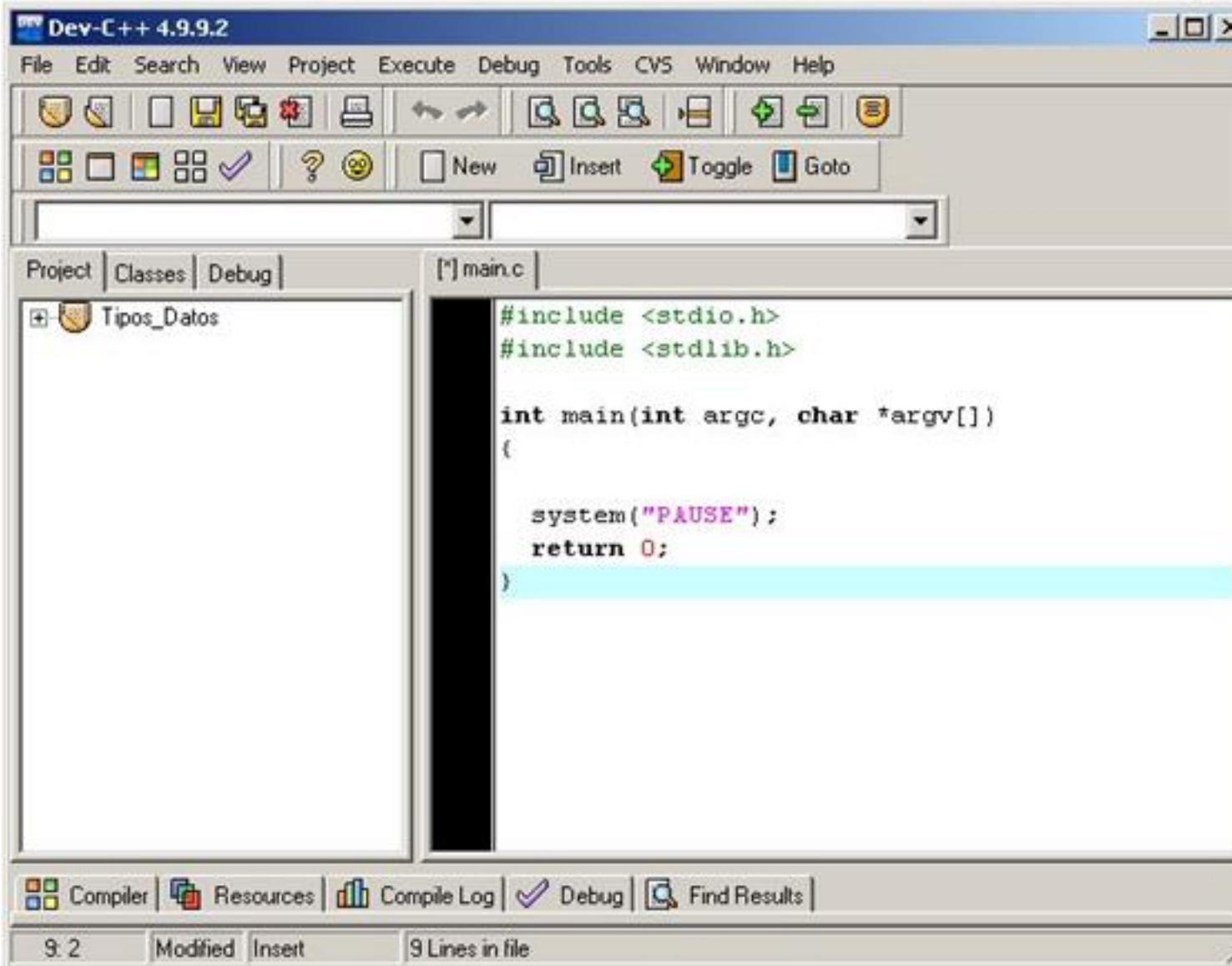
Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00



Programado desde el principio el archivo *main.c*, pero lo borramos ya que queremos aprender a

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00



The image shows the Dev-C++ 4.9.9.2 IDE interface. The main window displays a C program in the file `main.c`. The code is as follows:

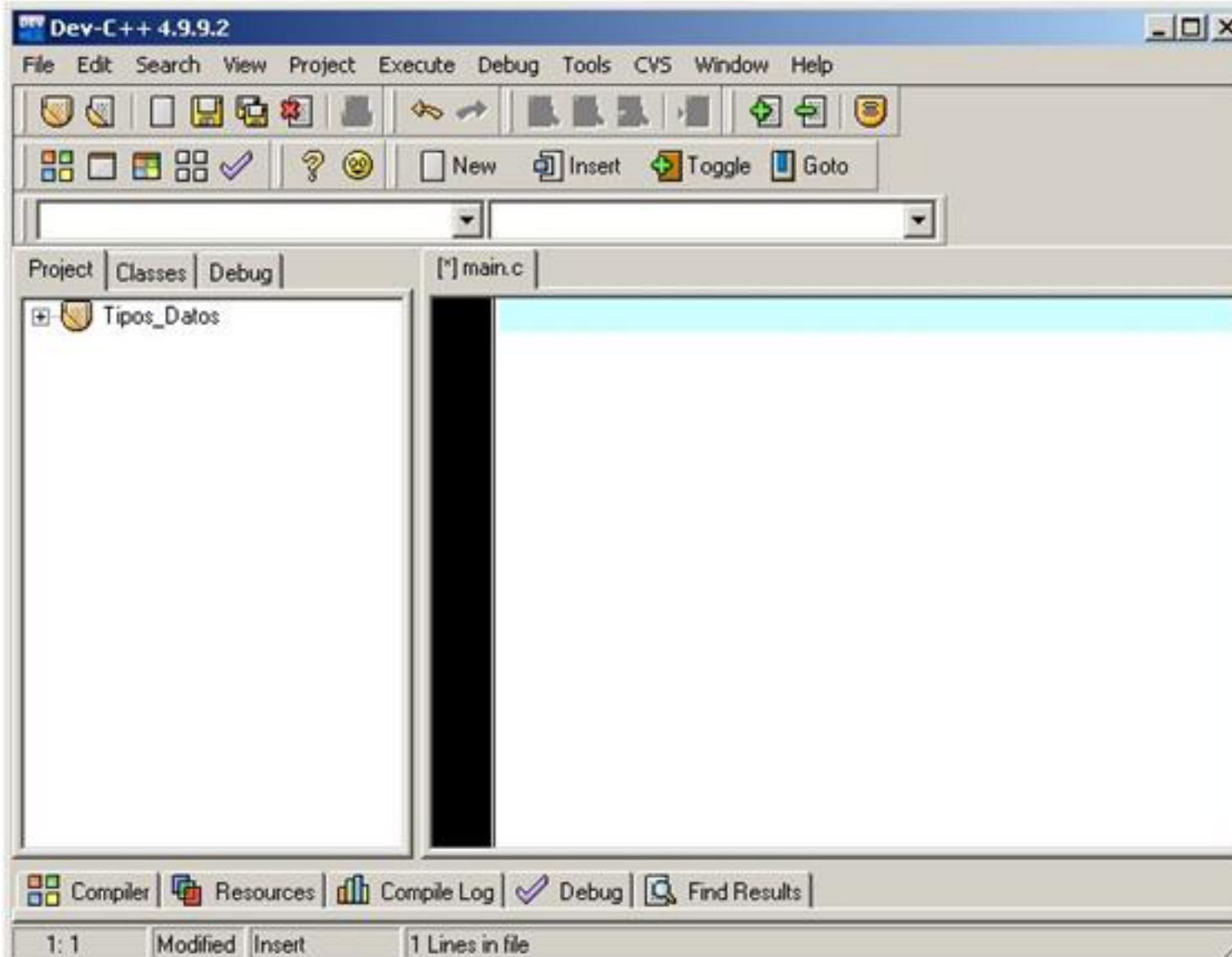
```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
(
    system("PAUSE");
    return 0;
)
```

The IDE includes a menu bar (File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, Help), a toolbar with various icons, and a project explorer on the left showing a project named "Tipos_Datos". The status bar at the bottom indicates "9: 2 Modified Insert 9 Lines in file".

Tipos de datos en C

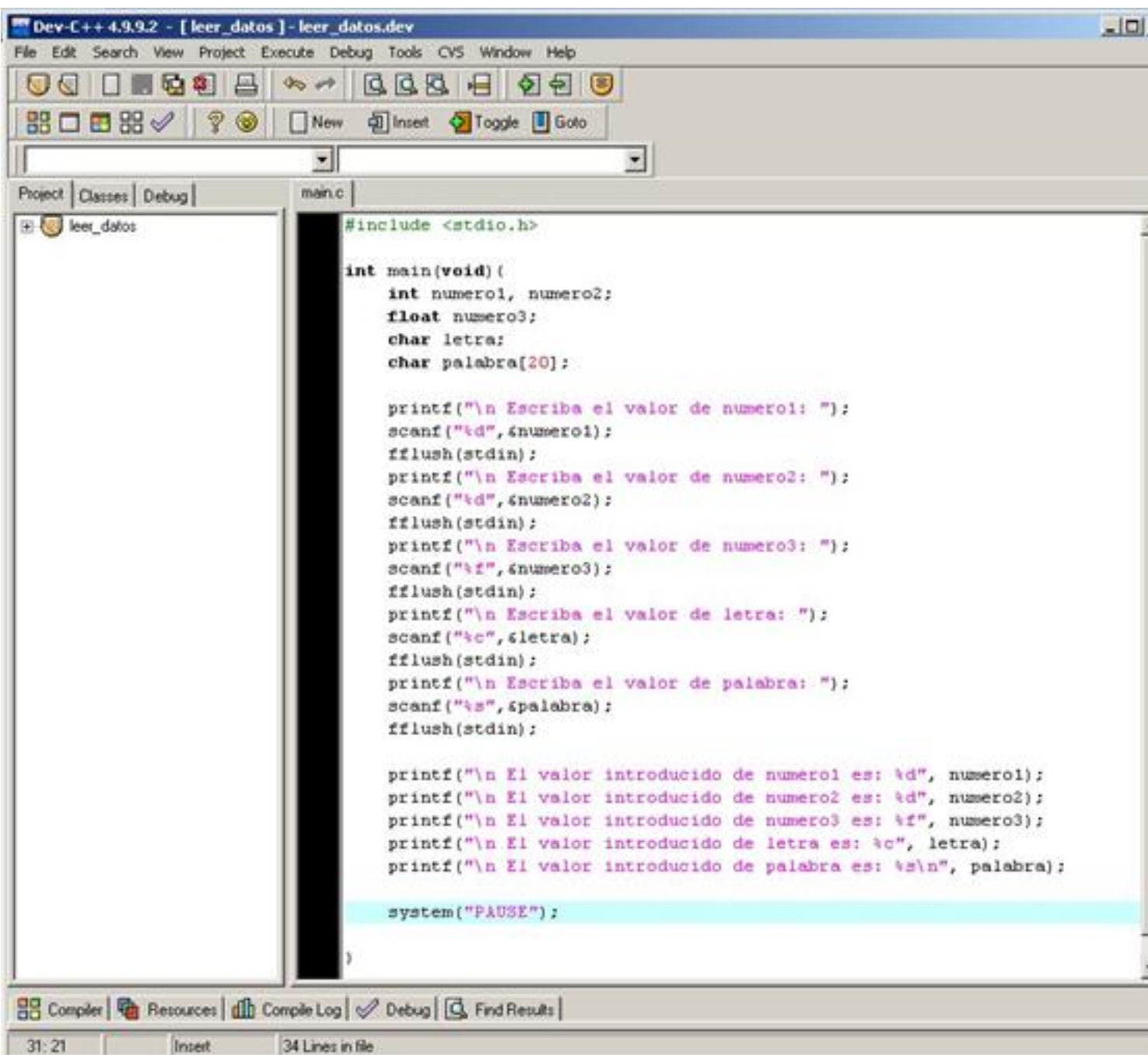
Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00



Algunos de los tipos de datos más comunes en C son los enteros e imprimiendo las funciones de
librería de C para imprimir los datos de un tipo de dato. En la sección de código que buscamos, el
usuario ha escrito el código que muestra los valores de los datos de un tipo de dato. En la sección de
código que buscamos, el usuario ha escrito el código que muestra los valores de los datos de un tipo de dato.

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00



The screenshot shows the Dev-C++ 4.9.9.2 IDE with a C program in the editor. The program is titled 'leer_datos' and is located in 'main.c'. The code includes the standard input/output header and defines a main function that prompts the user for four different types of data: two integers, a float, a character, and a string. Each input is followed by a flush of the standard input stream. The program then prints the values back to the user in a formatted manner. The last line of the program is a system call to pause the execution.

```
#include <stdio.h>

int main(void) {
    int numero1, numero2;
    float numero3;
    char letra;
    char palabra[20];

    printf("\n Escriba el valor de numero1: ");
    scanf("%d", &numero1);
    fflush(stdin);
    printf("\n Escriba el valor de numero2: ");
    scanf("%d", &numero2);
    fflush(stdin);
    printf("\n Escriba el valor de numero3: ");
    scanf("%f", &numero3);
    fflush(stdin);
    printf("\n Escriba el valor de letra: ");
    scanf("%c", &letra);
    fflush(stdin);
    printf("\n Escriba el valor de palabra: ");
    scanf("%s", &palabra);
    fflush(stdin);

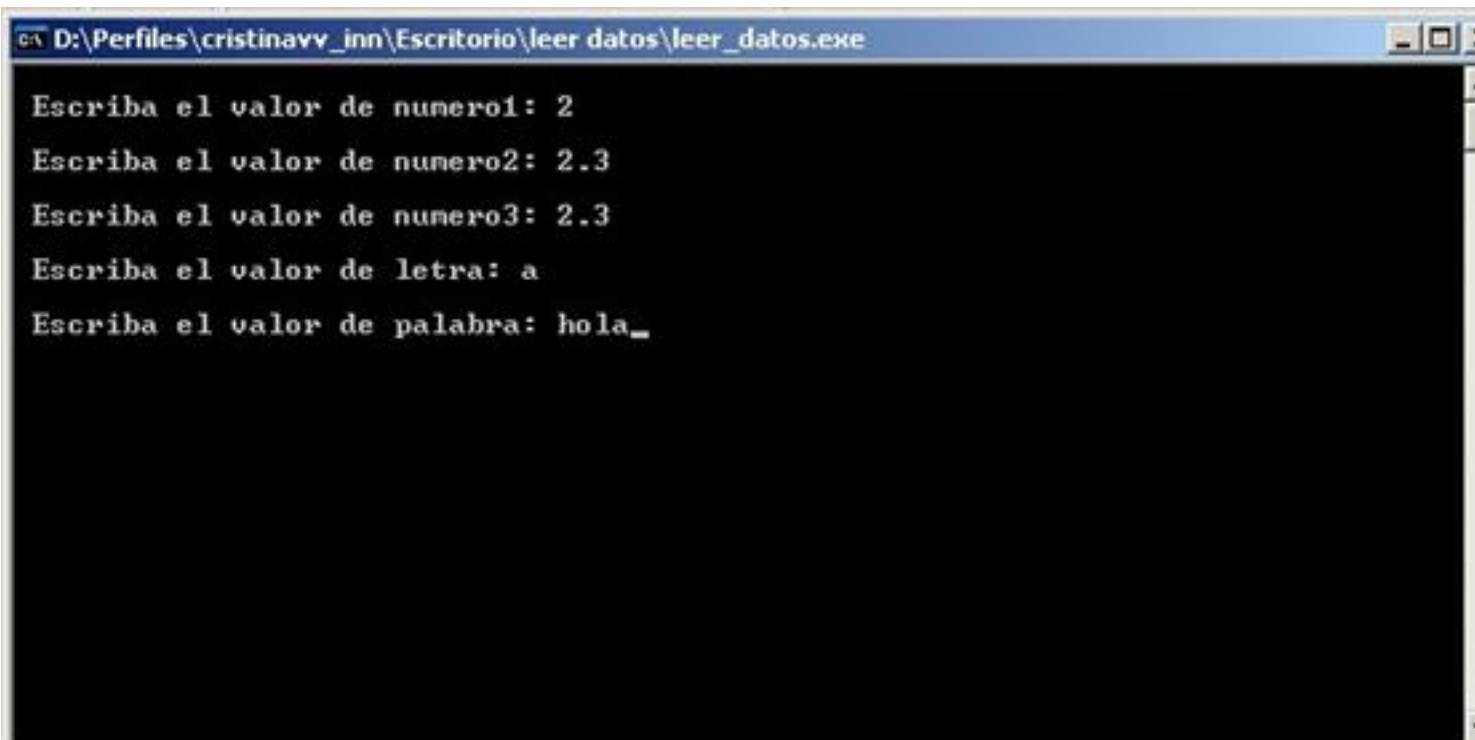
    printf("\n El valor introducido de numero1 es: %d", numero1);
    printf("\n El valor introducido de numero2 es: %d", numero2);
    printf("\n El valor introducido de numero3 es: %f", numero3);
    printf("\n El valor introducido de letra es: %c", letra);
    printf("\n El valor introducido de palabra es: %s\n", palabra);

    system("PAUSE");
}
```

El resultado es el siguiente:  el programa para comprobar que todo es correcto.

Tipos de datos en C

Écrit par Autor Cristina Villoria
Mardi, 07 Juin 2011 00:00



```
D:\Perfiles\cristinavv_inn\Escritorio\leer datos\leer_datos.exe

Escriba el valor de numero1: 2
Escriba el valor de numero2: 2.3
Escriba el valor de numero3: 2.3
Escriba el valor de letra: a
Escriba el valor de palabra: hola_
```

Una vez introducidos todos los valores, se muestran por pantalla:



```
D:\Perfiles\cristinavv_inn\Escritorio\leer datos\leer_datos.exe

Escriba el valor de numero1: 2
Escriba el valor de numero2: 2.3
Escriba el valor de numero3: 2.3
Escriba el valor de letra: a
Escriba el valor de palabra: hola

El valor introducido de numero1 es: 2
El valor introducido de numero2 es: 2
El valor introducido de numero3 es: 2.300000
El valor introducido de letra es: a
El valor introducido de palabra es: hola
Presione una tecla para continuar . . . _
```