

---

PROGRAMACION

---

**GUIA: Conociendo el entorno de BORLAND C++ 3.1**

Para comenzar a trabajar, Ud. deberá cambiar de directorio para ubicarse en **C:\BORLANDC\BIN**. Una vez ubicado en este directorio ingrese **BC** y luego presione **ENTER**. Esto lo pondrá en el entorno de **BORLAND C++ 3.1 (BC)**.

Debido a que **BC** inicia sus sesiones de la misma forma en que se cerró la última vez, puede ocurrir que Ud. se encuentre con la pantalla principal de BC vacía o con una ventana abierta con el programa con el que se estuvo trabajando en la sesión anterior. Si se da este caso, presione **ALT + F3** para cerrar dicha ventana.

**IMPORTANTE:** El lenguaje C es muy poderoso, lo que lo hace peligroso en caso de un uso incorrecto del mismo. Por lo tanto, no intente hacer "pruebas" que no estén indicadas en esta guía sin consultar previamente con su instructor. En el caso del entorno, también se debe ser cuidadoso pues hay ciertos comandos que pueden afectar el funcionamiento de su programa y/o complicarle el trabajo al próximo usuario. Ud. no es el único que trabajará con **BORLAND C** en esta máquina.

**MENU**

La pantalla de **BORLAND C++** presenta un menú de barras (denominado MENU PRINCIPAL) con las siguientes opciones:

≡ File Edit Search Run Compile Debug Project Options Window Help

Cada una de estas opciones a su vez tiene una serie de opciones llamadas **comandos**. Estos comandos son expuestos a través de menús desplegables (pop ups) para poder acceder a ellos. Toda tarea que se desee realizar en el entorno se debe hacer por medio de dichos comandos. Si un comando aparece en color más tenue que los otros, quiere decir que está desactivado (es decir, no se lo puede usar en ese contexto).

En el nombre de cada opción hay una letra de distinto color a las restantes llamada **HOT KEY** (tecla clave). En el caso de los comandos, a la par de algunos de ellos se encuentra la combinación de teclas que se deben presionar para acceder directamente a ellos. Esta combinación de teclas se denomina **SHORT CUT** (atajo).

Para acceder a los comandos existen diferentes caminos:

**1)** se activa el menú principal presionando la tecla **F10**, luego se desplaza el cursor hasta la opción deseada y se presiona **ENTER** para confirmar (también se llega a lo mismo presionando la tecla clave correspondiente una vez activado el menú principal). Esto abre el pop up correspondiente permitiendo que, por medio de la tecla ↓ se mueva el cursor hasta el comando y se presione **ENTER** para confirmar.

**2)** se presiona la tecla **ALT** junto con la tecla clave (**HOT KEY**) de la opción donde se encuentre el comando. Al abrirse el menú desplegable se presiona la tecla clave correspondiente al comando.

**3)** en el caso que exista, se usa el atajo correspondiente al comando.

**VENTANAS**

En el entorno de **BORLAND C++**, la mayoría de las tareas se realizan en **ventanas**. Las ventanas no son otra cosa que pantallas que se pueden mover, modificar de tamaño, superponer, abrir y cerrar entre otras cosas. Esto facilita enormemente las tareas de edición, debugging (eliminación de "bichos" o errores del programa), comprobación del comportamiento del programa, etc..

En **BORLAND C++** se pueden abrir varias ventanas simultáneamente (esta es la gran ventaja con respecto a trabajar con una única pantalla como se hacía antes) permitiendo que

## PROGRAMACION

---

en una de ellas se pueda editar el programa mientras en otra se pueda observar la salida que dé como resultado correrlo y en otra se puedan ver los eventuales mensajes de error.

De todas las ventanas que estén abiertas sólo una puede estar activa. La ventana activa es aquella en la que se está trabajando actualmente. Por lo general, cualquiera de los comandos que se elija se aplican a esta ventana. Lo mismo ocurre con el texto que uno trate de ingresar por medio del teclado.

Hay distintos tipos de ventanas de acuerdo a la tarea específica que se pueda desarrollar sobre cada una de ellas. Dichos tipos son :

- **VENTANA DE EDICION ( EDIT )** : donde se editan los programas
- **VENTANA DE MENSAJES ( MESSAGES )** : donde **BORLAND C++** muestra los mensajes de error o de advertencia .
- **VENTANA DE SALIDA ( OUTPUT )** : donde **BORLAND C++** muestra las salidas de texto ( no gráficos ) de los programas escritos por los usuarios.
- **VENTANA DE OBSERVACION ( WATCH )** : donde se puede observar el comportamiento de expresiones a través del cambio de sus valores a medida que avanza el programa.
- **PANTALLA DEL USUARIO ( USER SCREEN )** : es como la ventana de salida pero abarca toda la pantalla .

Además están las ventanas **REGISTER** y **PROJECT** que por ahora no nos interesan.

### LA LINEA DE STATUS

Esta línea aparece en el fondo de la pantalla de **BORLAND C++**. Su funcionamiento es el siguiente:

- indica lo que el entorno está haciendo en un dado momento. Ej., cuando se salva un programa muestra el mensaje : " saving archivo ... "

- recuerda algunas de las teclas claves o atajos que se pueden usar para acceder a ciertos comandos .- da una breve descripción de lo que hace el comando sobre el que está el cursor.

El contenido de esta línea cambia a medida que se cambia de tipo de ventana o de actividad .

### PASOS PARA LA CREACION Y EJECUCION DE UN PROGRAMA

Para crear y ejecutar un programa se necesitan dar los siguientes pasos :

- 1) Editar el programa fuente y grabarlo.
- 2) Compilar el programa fuente a fin de generar el programa objeto.
- 3) Hacer funcionar el programa ejecutable.

Con esta breve introducción Ud. ya está en condiciones de comenzar a trabajar. Su tarea consistirá en editar el programa que se encuentra a continuación, compilarlo, probarlo, detectar los errores (si los hubiere), ponerlo a punto y hacerlo funcionar.

### EJEMPLO.CPP

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

```
/* Problema: de una cantidad de días dados, calcular cuantas semanas y quincenas hay en ellos */
```

---

**PROGRAMACION**

---

```
main()
{
    int dia, sem, quin;           /* declaro las variables */

    printf("Ingrese cantidad de dias"); /* mensaje de solicitud de datos al usuario */
    scanf("%d",&dia);           /* lectura de datos */
    sem = dia*7;                 /* sector de calculos*/
    quin = dia/15;
    printf ("Hay %d semanas y %d quincenas",sem,quin); /*escritura de los resultados */
    return 0;
}
```

**INSTRUCCIONES**

Para trabajar con este ejercicio siga las siguientes instrucciones:

- 1) Presione **ALT + F** para acceder al menú de archivos (**FILE**). Luego presione **C** para acceder al comando **Change Dir** (Cambio de Directorio por defecto). Cuando **BORLAND C++** le solicite el nuevo directorio, ingrese **C:\BORLANDC\P0609** y presione **ENTER**.
- 2) Presione **ALT + F** y luego **N** para abrir un nuevo archivo donde comenzará a ingresar el programa arriba indicado.
- 3) Una vez que termine de ingresar el programa presione **F2** para guardarlo (use el nombre **EJEMPLO1.CPP**).
- 4) Presione **ALT + F9** para compilar el programa. También puede hacerlo presionando **ALT + C** y luego **C** nuevamente. Aparecerá entonces un recuadro con el título **COMPILING** donde mostrará información sobre la compilación. Apriete cualquier tecla para salir del recuadro. **BORLAND C++** activará la ventana de mensajes para indicarse la existencia de un error.
- 5) Presione **ENTER** para salir de la ventana de mensajes y volver a la ventana de edición donde **BORLAND C++** le mostrará un mensaje en otro color indicando el lugar y tipo de error. Elimine los caracteres **/\*** y **\*/** de la línea donde se encuentra la expresión "return (0)" y repita el paso 4).
- 6) Presione **CTRL + F9** para hacer correr su programa.
- 7) Presione **ALT + W** y luego **O** para abrir la ventana de salida donde podrá observar el resultado de hacer correr su programa.
- 8) Presione **F6** para saltar de la ventana de salida a la ventana de mensajes y vuelva a presionar **F6** para saltar a la ventana de edición.

**NOTA :** la notación **AAA + BBB** implica mantener simultáneamente presionadas las teclas **AAA** y **BBB**

---

**Notas sobre los procesos de edición y compilación**

Para darle instrucciones en código C a una computadora necesitamos de un editor y un compilador. Un editor es parecido a un procesador de palabras, permite ingresar el código original del programa que se llama código fuente. También hace posible que el programa se pueda almacenar en memoria y en disco, corregir mediante opciones tales como mover, copiar, insertar texto, borrar.

El compilador es el encargado de traducir las instrucciones escritas en lenguaje C a código que pueda ser interpretado por la computadora.

---

PROGRAMACION

---

Estos dos elementos: editor y compilador están presentes en todos los Entornos Integrados de Desarrollo (IDE). El programador debe adquirir destreza en el manejo del IDE, lo cual es más simple que aprender a programar en código C.

Si el programador no usa un IDE, deberá proveerse de un compilador de código C. En general, el compilador C está incluido en los compiladores de C++, ya que éste último está basado en C y se diferencia porque usa una técnica llamada programación orientada a objetos (OOP). Por lo tanto, un compilador como el Visual C++ o el Borland C++ reconocerá el código fuente escrito en C, bastará configurarlo de manera adecuada.

Todos los programas escritos en código C se almacenan en archivos cuya extensión es C. Si usa un compilador de C++, la extensión será CPP. Solo después de escribir, compilar y corregir su programa en C podrá ejecutarlos y disponer de los resultados esperados.

Antes de ejecutar el programa, el mismo pasa por una última etapa que se denomina vinculación (linking). El objetivo de esta etapa es determinar las direcciones de memoria asignadas a las variables y el código, también establecer los nexos o vínculos necesarios para que el programa pueda interactuar con los módulos, que podrán ser otros programas. Dicha etapa, en el IDE es transparente para el usuario.

### Preguntas de autoevaluación

1. ¿Usó el compilador en esta práctica? ¿Para que lo usó? ¿Podía prescindir de su uso? Defina qué es un compilador.
2. Mientras realizaba el trabajo indicado bajo el título **INSTRUCCIONES**, ¿Encontró errores? ¿De qué tipo? ¿Los pudo detectar fácilmente?
3. El lenguaje C es un lenguaje de alto nivel ¿por qué?
4. ¿Qué versión del lenguaje C ha usado en el desarrollo del práctico?
5. Analice minuciosamente el programa y encuentre los elementos básicos de un programa escrito en código C.
6. ¿Por qué se usa sangría al escribir un programa? Si no le gusta usar sangría, ¿Qué propuesta tiene Ud. para mejorar la legibilidad y comprensión del texto?
7. Escriba el mismo programa en una sola línea. Para abreviar texto, suprima los comentarios ¿Qué puede concluir?