

Desarrollo de programas

Jaime Sánchez. Sistemas Informáticos y Computación, UCM

Desarrollo de programas

Dos ideas previas sencillas:

- ▶ El objetivo de un programa es **resolver un problema**.
- ▶ El ordenador *no piensa*, solo ejecuta un programa dado.

Un programa tiene que estar **bien escrito** de acuerdo con la sintaxis del lenguaje, para que pueda ser ejecutado.

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, outlined font. The '99' is larger and more prominent. The text is set against a background of overlapping light blue and orange geometric shapes, possibly representing a map or abstract design.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

- - -

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Programas correctos (I)

Un programa es correcto si hace **exactamente** la tarea para la que ha sido diseñado.

Formalmente: es correcto si se comporta exactamente según una especificación dada. La especificación define o describe lo que debe hacer hacer el programa.

A su vez la especificación puede ser:

- ▶ Formal (utilizando lenguajes formales de especificación, que utilizan la lógica matemática). Por ejemplo, con el modelo de **precondición y postcondición** y la lógica de **Hoare**.
- ▶ Menos formal, utilizando lenguaje natural.

En cualquier caso la especificación debe ser absolutamente precisa a la hora de describir **qué debe hacer el programa**.

Jaime Sánchez. Sistemas Informáticos y Computación, UCM

Programas correctos (II)

En particular, un programa correcto:

- ▶ Debe **funcionar para todos los casos previstos**, dando el resultado esperado según la especificación
- ▶ Debe estar **completamente libre de errores**.
- ▶ Debe **terminar** adecuadamente

Además, la buena metodología de programación busca que los programas:

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

ello se consigue que sean más fáciles de entender, reutilizar

Desarrollo de programas. Metodología.

1. **Análisis y Especificación** del problema (**¿QUÉ?**):
 - ▶ extraer información relevante, eliminar ambigüedades del planteamiento
 - ▶ identificar los datos de entrada o **input** y los datos de salida o **output** y qué hay que hacer
2. **Diseño** de un **algoritmo** (**¿CÓMO?**): precisar los pasos para obtener la solución requerida (output) a partir de la entrada (input).
 - ▶ Partir de planteamiento general prescindiendo de detalles (dejar pendientes subproblemas más pequeños). Después abordar estos subproblemas con la misma técnica $\rightsquigarrow$ **Diseño descendente** o **divide y vencerás** o **aproximación por refinamientos sucesivos**.
3. **Implementar** el algoritmo en un lenguaje concreto (como C#, en nuestro caso). Compilarlo, corregir posibles errores de sintáxis, ...
4. **Probar y depurar (test)**: comprobar el funcionamiento del programa con una batería de ejemplos *intentando* cubrir toda la casuística posible.
 - ▶ Otra alternativa **verificación formal** de la corrección del algoritmo $\rightsquigarrow$ demostración formal (matemática) utilizando la lógica de Hoare, verificadores (semi)-automáticos de programas...

Desarrollo de programas (II)

Es habitual completar las 4 fases anteriores e iterar, es decir, volver a la fase 1, repasar la especificación (a veces el diseño del algoritmo o la propia implementación requieren modificar la especificación), adaptar la implementación, etc....

En el *ciclo de vida de un programa*, puede incluirse la fase 5:

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

Cartagena99

Ejemplo (I)

Problema:

averiguar el más pequeño entre dos números dados en cualquier orden

Primer paso, **Análisis y especificación:**

- ▶ Información irrelevante?: “dados en cualquier orden”
- ▶ Ambigüedad?: ¿cuál es el más pequeño entre 6 y 6?...
 - ▶ suponemos que el usuario (cliente) desea 6 como respuesta (o se lo preguntamos para aclararlo)
- ▶ Imprecisión?: los números dados, ¿son naturales, enteros, reales, complejos?...
 - ▶ supondremos que son enteros (o preguntamos al cliente)

Jaime Sánchez. Sistemas Informáticos y Computación, UCM

Ejemplo (II)

Especificación (distintas formas de hacerla):

- ▶ en lenguaje natural (español), pero precisa:
determinar el mínimo entre dos enteros dados
- ▶ más formal, apoyada en lenguaje lógico/matemático:

dados $x, y \in \mathbb{Z}$ determinar $z = \min(x, y)$ siendo

$$\min(x, y) = \begin{cases} x & \text{si } x \leq y \\ y & \text{en otro caso} \end{cases}$$

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

- ▶ output: el más pequeño de ellos, mínimo entre x y y

Ejemplo (III)

Segundo paso, diseño del algoritmo: secuencia de acciones a realizar.

- ▶ Aproximación I:
solicitar de teclado los números de entrada x e y
calcular en z el mínimo de x e y
escribir z en pantalla
- ▶ Aproximación II (refinamiento de I):
solicitar de teclado los números de entrada x e y
si $x \leq y$ hacer $z = x$
en caso contrario hacer $z = y$
escribir z en pantalla

Jaime Sánchez. Sistemas Informáticos y Computación, UCM

Ejemplo (III)

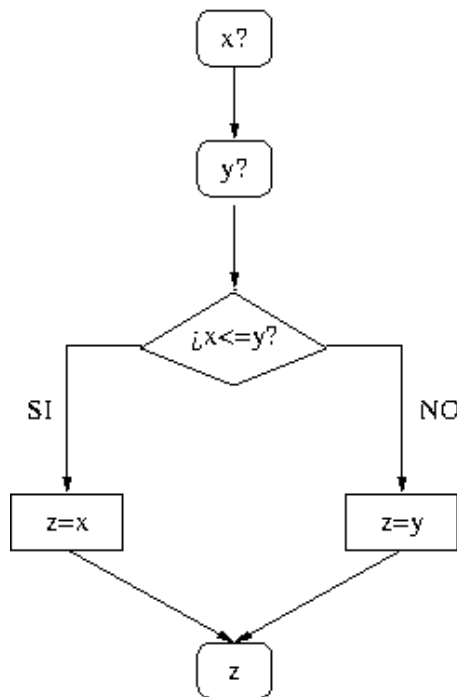
- ▶ Aproximación III (refinamiento de II):
escribir ' $x?$ ' en pantalla
leer el valor de x de teclado
escribir ' $y?$ ' en pantalla
leer el valor de y de teclado
si $x \leq y$ hacer $z = x$

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

Cartagena99

Diagramas de flujo para algoritmos: otra forma de presentar los algoritmos (no la utilizaremos habitualmente):



Jaime Sánchez. Sistemas Informáticos y Computación, UCM

Ejemplo (IV)

Ahora habría que hacer la implementación en un lenguaje concreto. Por ejemplo, en Python se podría hacer como (ahora no nos importan los detalles del lenguaje):

```
print "Primer entero: ",  
x = int(input())  
print "Segundo entero: ",  
y = int(input())  
if x <= y:
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

Ejemplo (V)

Quinto paso, **test**.

Primer entero: 8

Segundo entero: -17

El menor es: -17

Esto es simplemente un test, no un prueba de corrección.

Verificación formal: precondición $\xrightarrow{\text{programa}}$ postcondición

$\{P_0 \equiv x, y, z \in \mathbb{Z}\}$

if (x<=y) then z:=x

else z:=y

$\{P_1 \equiv z = \min(x, y)\}$

La **lógica de Hoare** define el comportamiento de cada instrucción.

Jaime Sánchez. Sistemas Informáticos y Computación, UCM

Ejemplo (VI)

Mantenimiento. El usuario pide solicita cambios:

- ▶ Cambio en la especificación: los números en vez de enteros que sean reales (ampliación de la funcionalidad).
- ▶ Cambios de petición y presentación de datos: que pida los dos números a la vez, que escriba en salida también los números de entrada.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

- - -

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Otro ejemplo

Problema:

calcular la suma de los n primeros naturales

► **Análisis y especificación:**

dado $n \in \mathbb{N}$ evaluar $1 + 2 + \dots + n$ (en términos matemáticos $\sum_{i=1}^n i$)

input: $n \in \mathbb{N}$; output: $1 + 2 + \dots + n$

► **Algoritmo:** (sin aplicar la fórmula conocida)

Aproximación I:

solicitar de teclado el valor n

calcular $Suma = 1 + 2 + \dots + n$

escribir $Suma$ en pantalla

Jaime Sánchez. Sistemas Informáticos y Computación, UCM

Refinamiento

Refinamiento I (sólo del cálculo en sí):

inicializar $Suma = 0$

repetir desde $i = 1$ hasta n

$Suma = Suma + i$

Implementación (ahora en Pascal):

```
{Este programa calcula la suma...}  
program sumatorio;  
var i,n,suma: integer;
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

- - -

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

Luego depuración, mantenimiento...

Compiladores

Jaime Sánchez. Sistemas Informáticos y Computación, UCM

Nuestro primer programa en C#: “hola mundo!”

```
using System;
class HolaMundo
{
    static void Main()
    {
        Console.WriteLine ("Hola mundo!!");
    }
}
```

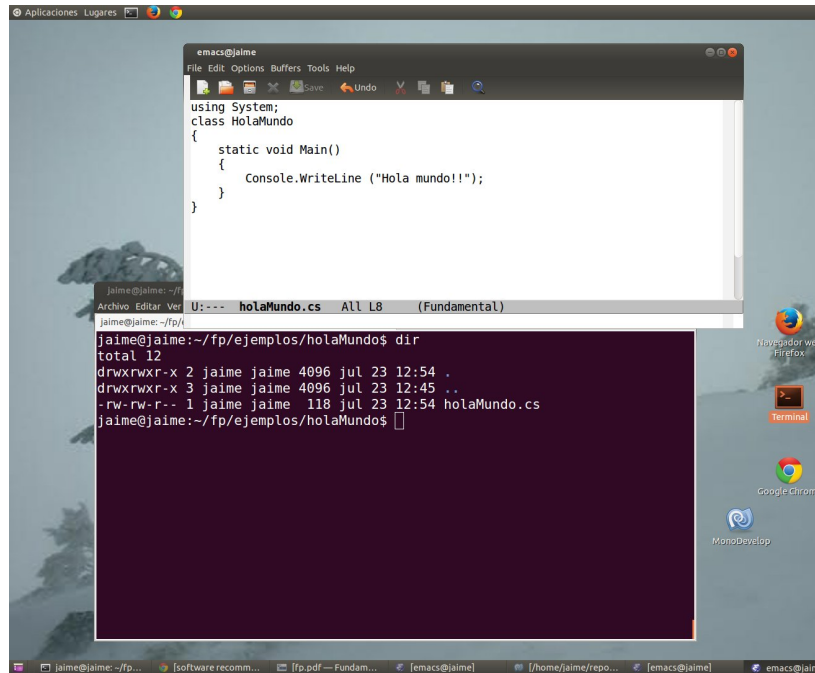
No es necesario entender este programa por ahora...

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

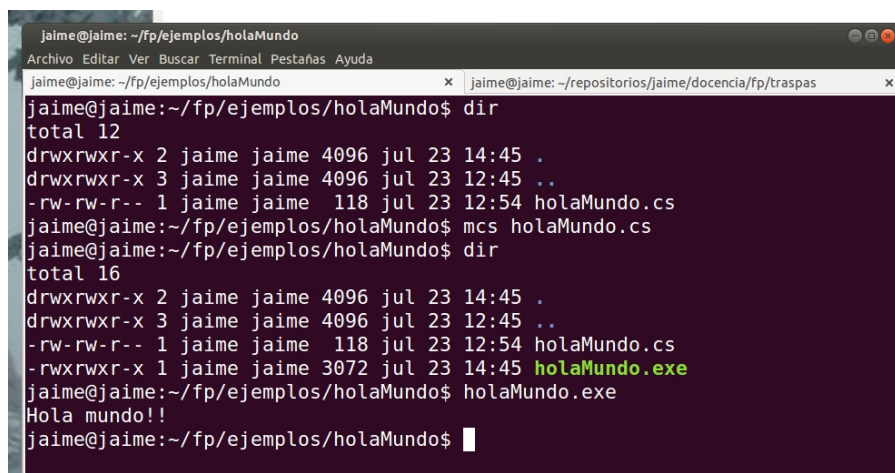
Compilando desde línea de comandos (I)



La línea de comandos todavía existe! (en Linux, Windows, Mac...)

Jaime Sánchez. Sistemas Informáticos y Computación, UCM

Compilando desde línea de comandos (II)



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

> holaMundo.exe

Pero...

- ▶ ¿Qué significa exactamente **compilar** un programa?
- ▶ ¿Qué es un **compilador**?
- ▶ ¿Qué significa **ejecutar** un programa?
- ▶ ¿Quién compila?
- ▶ ¿Quién ejecuta?

Jaime Sánchez. Sistemas Informáticos y Computación, UCM

¿Qué significa exactamente **compilar** un programa?

Compilar es **traducir**: se traduce el **código fuente** escrito en un lenguaje de programación (como C#) **código objeto**. Este código objeto:

- ▶ puede ser código binario para una máquina real (CPU) directamente ejecutable por la misma (ceros y unos que “entiende” el ordenador, i.e., instrucciones para el microprocesador)
- ▶ o puede ser **código para una máquina virtual (bytecode)**, que puede ser fácilmente convertible en código ejecutable

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

En concreto, C# se compila/traduce a un lenguaje intermedio para

máquina virtual (Common Intermediate Language, CIL) utilizado en la

Plataforma .NET (usamos el compilador mcs)

Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Cartagena99

¿Qué significa ejecutar un programa? ¿Quién ejecuta?

- ▶ Ejecutar un programa es hacerlo funcionar en el ordenador. Para ello, el **sistema operativo** (linux, windows,...) lo carga en memoria y la CPU efectúa las instrucciones de ese programa.
- ▶ El sistema operativo (SO) es el que ejecuta los programas en el ordenador.
- ▶ El sistema operativo, a su vez es un programa de base (kernel del SO) que utiliza a su vez un conjunto de programas (más o menos básicos).

Jaime Sánchez. Sistemas Informáticos y Computación, UCM

Lenguajes de programación y compiladores

- Las ideas de compilador y lenguaje de programación están muy ligadas, pero no son lo mismo:
 - ▶ En general puede haber distintos compiladores para un mismo lenguaje de programación. Por ejemplo, para el lenguaje C# hay varios compiladores como *MS VisualStudio* o *monodevelop*.

The logo for Cartagena99, featuring the text 'Cartagena99' in a stylized, outlined font. The '99' is larger and more prominent. The text is set against a background of overlapping light blue and orange geometric shapes.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

- - -

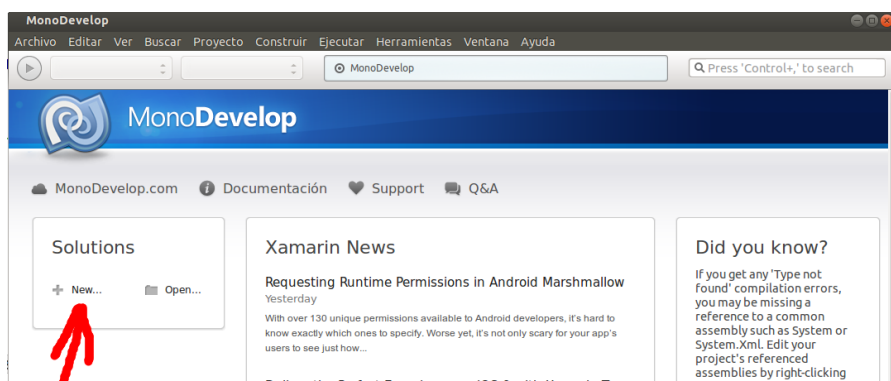
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Entornos de desarrollo

- En la actualidad es muy frecuente que los compiladores se integren en **entornos integrados de desarrollo**: entorno gráfico para el **desarrollo de programas** que incluye:
 - ▶ Editor de texto:
 - ▶ con resaltado de sintaxis, auto-completado inteligente de código, herramientas de construcción automáticas (plantillas de programa, etc).
 - ▶ El **compilador** propiamente dicho. No se lanza desde línea de comandos, sino con botones (a golpe de click de ratón).
 - ▶ Otras herramientas adaptadas al lenguaje concreto: enlazador de librerías, navegador de clases, ayuda sensitiva, etc.
 - ▶ **Depurador de código** (debugger). Es una herramienta que permite hacer trazas de ejecución del programa: ejecutar el programa *paso a paso* (seguir el flujo de ejecución) viendo en el contenido de las variables, la pila de ejecución, etc.

Es un gran aliado para el programador para detectar y corregir errores.
El entorno **MonoDevelop** para C#

Ventana de inicio

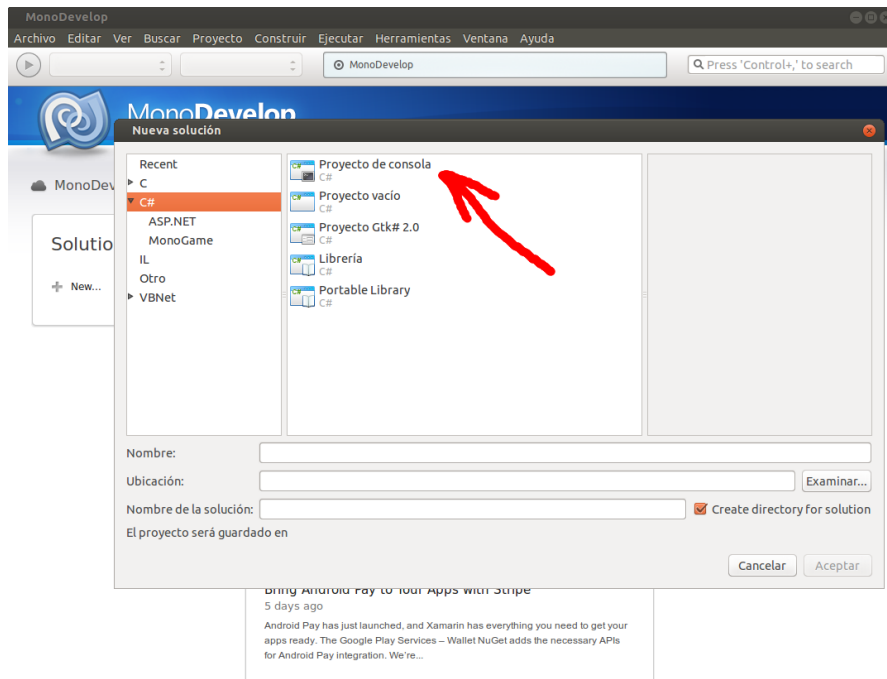


**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

El entorno MonoDevelop para C#

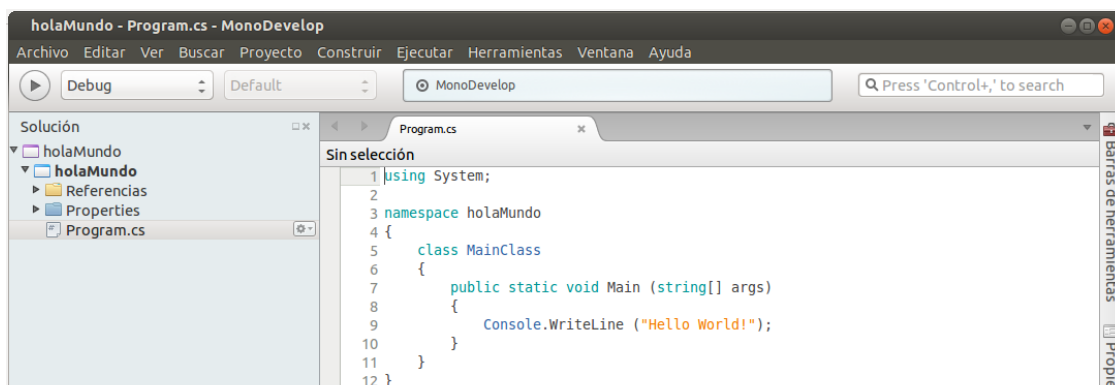
Nuestra solución, en este caso, será un proyecto de consola



Tenemos que dar nombre al proyecto, a la solución y una ubicación en disco duro.

El entorno MonoDevelop para C#

El entorno genera automáticamente una plantilla por defecto para nuestro programa:



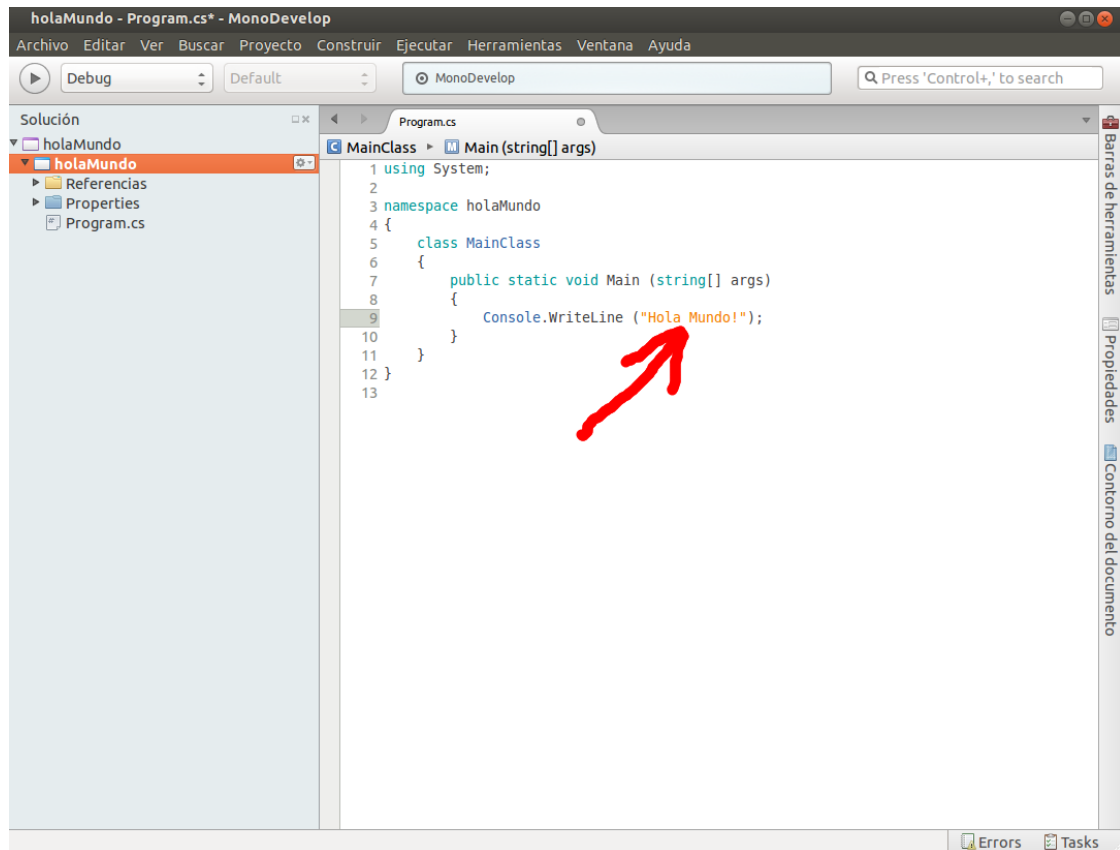
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

El entorno MonoDevelop para C#

Escribimos nuestro programa: en este caso es una modificación trivial de la plantilla.



El entorno MonoDevelop para C#

Damos al play!

- ▶ El entorno lanza el compilador de C#
- ▶ Produce un ejecutable
- ▶ Y lo ejecuta



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

- - -

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

