Informática

Clase 1: Introducción

Mario Merino Martínez
mario.merino@upm.es

Escuela de Ingeniería Aeronáutica y del Espacio
Universidad Politécnica de Madrid

6 de septiembre de 2011

Índice

Index

¿Qué es programar?

Un ordenador es una herramienta enormemente tonta, pero increíblemente rápida

- **Programar** es dar instrucciones de forma secuencial a un ordenador para que realice una tarea por nosotros.
- El ordenador sólo entiende instrucciones en forma de 1s y 0s (código máquina).
- **No queremos escribir en binario**: inventamos los lenguajes de programación (convenientes y fáciles de utilizar) y un programa (compilador) que traduzca lo que escribimos a código máquina (".exe").

Lenguajes de programación

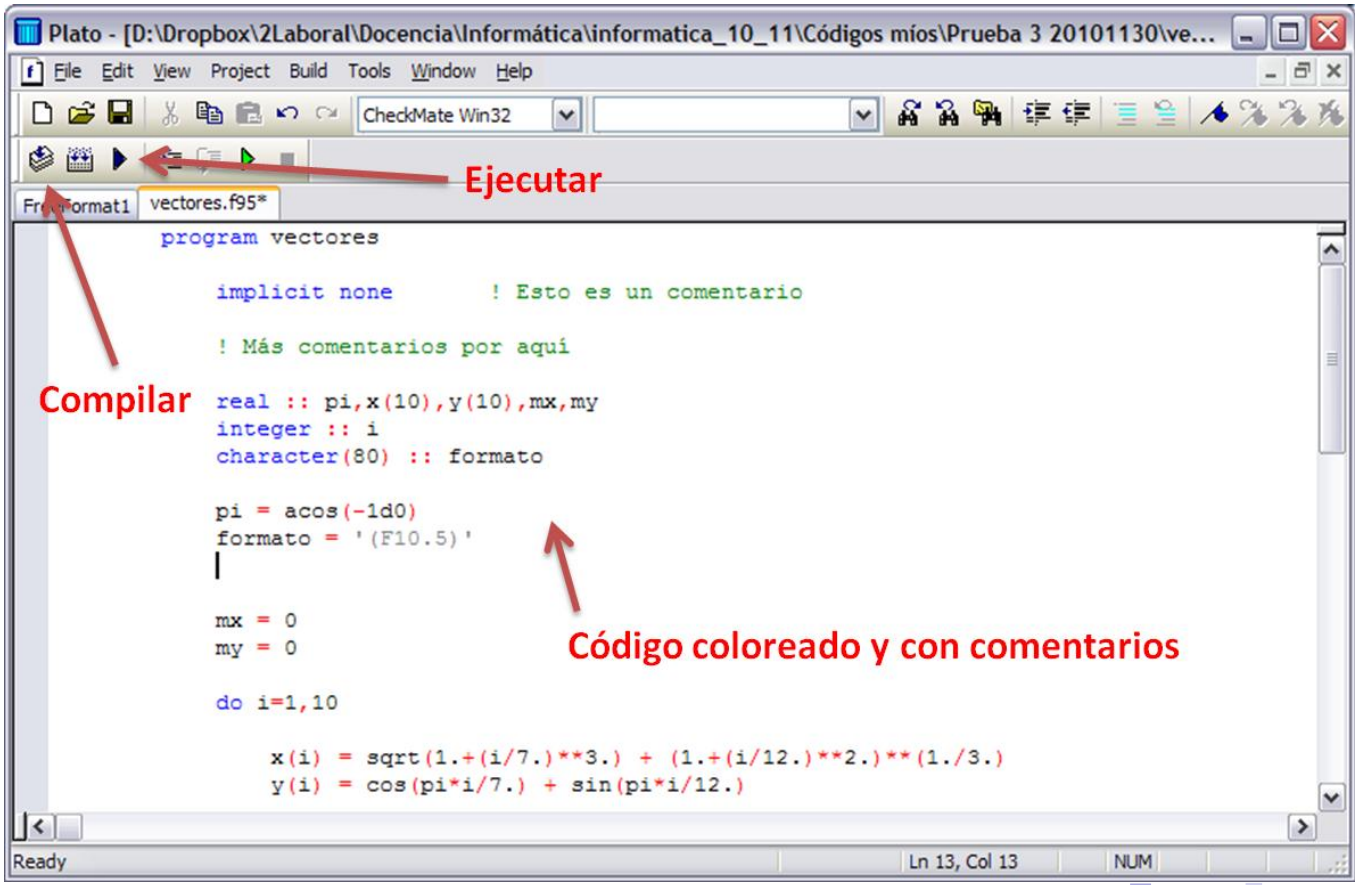
- Cada lenguaje tiene su vocabulario y sintaxis propios.
- Los compiladores **no son inteligentes**: no admiten fallos ni ambigüedades.
- Todos los lenguajes puede hacer "de todo", pero cada uno está **optimizado para un tipo de tarea**.

Fortran (de "*formula translation*") es el **lenguaje de programación científico e ingenieril** por excelencia. Se inventó por y para el cálculo numérico.

Index

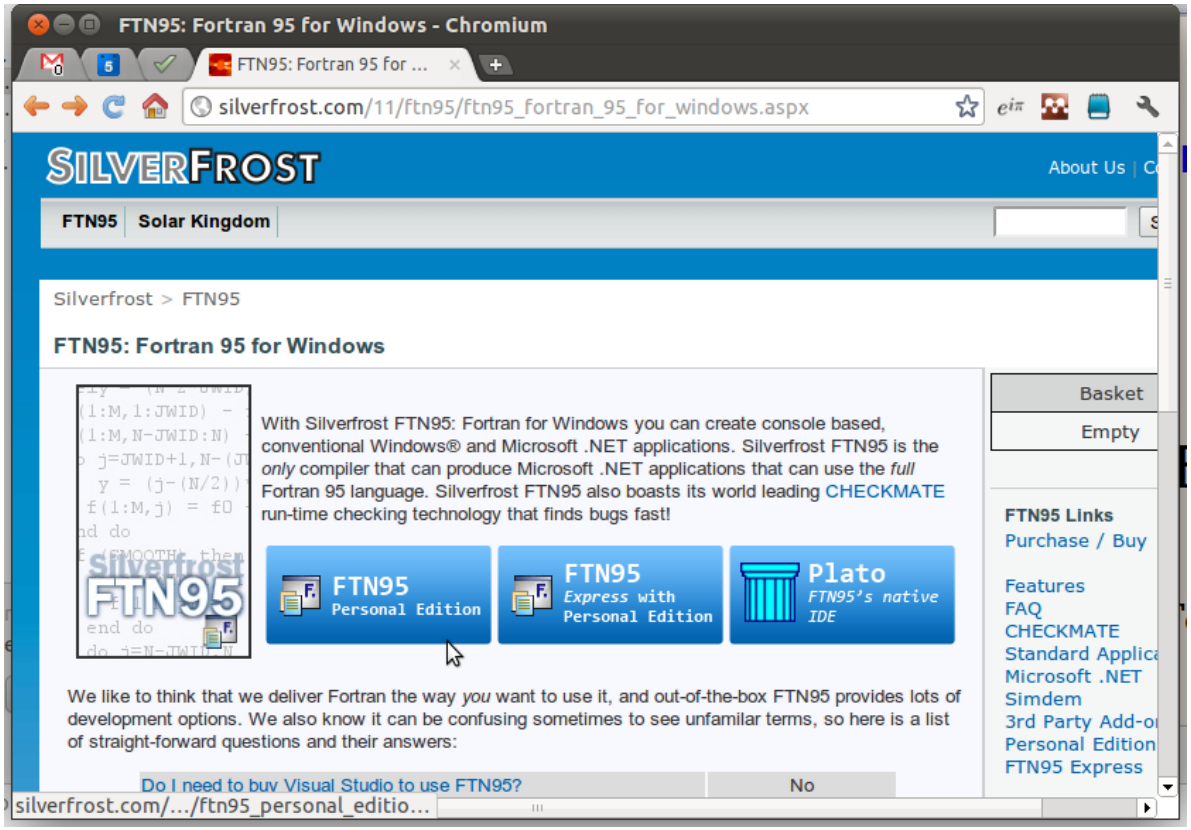
Cómo trabajar en Fortran

Para escribir el código de un programa, sirve cualquier editor de texto. ¡Pero mejor utilizar una herramienta especializada!



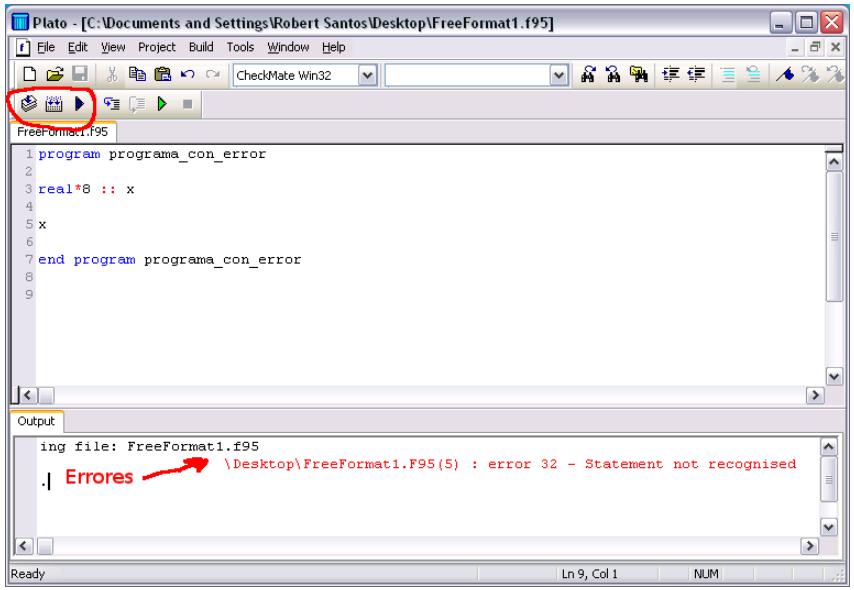
Instalación de un "Entorno de Desarrollo"

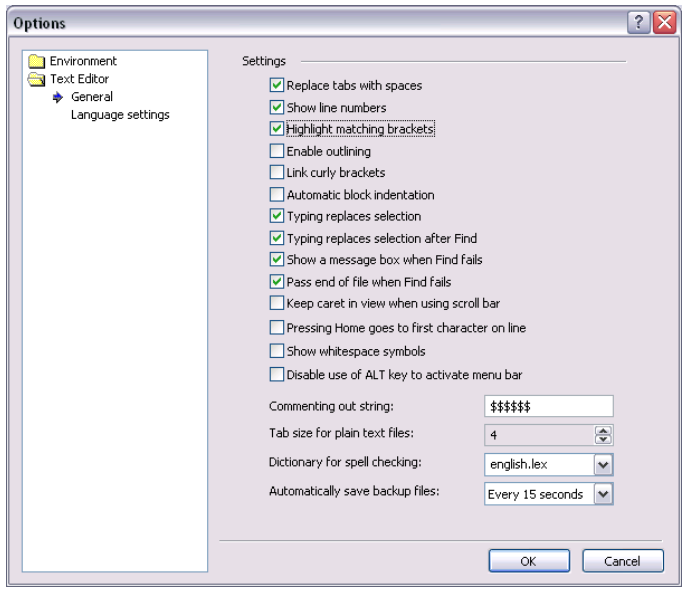
- Hay varios: usaremos FTN95 (gratuito) [alternativas]
- Id a http://silverfrost.com/ y descargar la **"personal edition"**. Ejecutadlo y seguid los pasos de la instalación



Uso de FTN95

- Crear nuevo documento del tipo "**Free fortran**"
- Tras escribir el código, pulsad sobre **"compile", "build" y "execute"**
- Configurad **opciones** a vuestro gusto (Tools|Options)





*Imprescindibles

- Invertir un poco de tiempo familiarizándose con el FTN95
- "`CTRL + F5`" equivale a compile + build + run
- Explorar todas las opciones de configuración. **Muy útiles:**
  - Show line numbers
  - Highlight matching brackets
- Familiarizarse con los errores y warnings más habituales del compilador

MUY IMPORTANTE: APRENDED A INTERPRETAR LOS ERRORES PARA PODER CORREGIRLOS

Index

Estructura de un programa en Fortran

- Un programa en Fortran es un **fichero de texto** con una **lista secuencial y finita de instrucciones**. El programa empieza y termina siempre así:

```
program nombre_del_programa
  ! Declaración de variables (líneas "!" son comentarios)
  ! Instrucciones SECUENCIALES
end program nombre_del_programa
```

- Fortran no distingue entre mayúsculas y minúsculas, e ignora espacios y líneas en blanco extra.

Index

Tipos y declaración I

Una **variable** es un **nombre** al que le asociamos un **número o dato**, para poder referirnos a él en el código de forma genérica. Hay que **declarar cada variable que vayamos a usar en la parte inicial del programa**, para que el compilador sepa que existe y de qué tipo es. Existen varios tipos de variables:

- Variables enteras: `integer`
- Variables reales, precisión simple: `real*4`
- Variables reales, precisión doble (más decimales): `real*8`
- Otros tipos (`complex`, `character`, `logical`...)

Siempre hay que: (1) **declarar**, (2) **asignar valores**, (3) **usar**

Tipos y declaración II

Se declaran así, **al principio del programa**:

```
program nombre_del_programa
  implicit none ! ponedlo SIEMPRE
  integer :: i, n
  real*4 :: s, temp
  real*8 :: h, x, y
  ! ...
end program nombre_del_programa
```

Los nombres sólo pueden tener A–Z, 1–9, "_". No "ñ". Han de empezar por letra; max 31 caracteres

`Implicit none` evita que Fortran asuma tipos si olvidamos declarar una variable (peligroso!)

Asignación de valores

Una vez las variables han sido declaradas, se les puede **dar un valor** con el **operador asignación** "=":

```
! ...
integer :: n
real*4 :: x, h
real*8 :: y
real*4, parameter :: cte = 3
! ... (parameter implica que el valor no se puede cambiar)
n = 34
h = 4.91234
x = 5.   ! Los reales SIEMPRE con punto o exponente (1e0)
y = 1d0  ! El exponente con "d" se utiliza para doble precisión
! ...
```

Uso en expresiones matemáticas

Tras declarar y asignar un valor, podemos usar las variables en expresiones matemáticas
Una variable **se puede sobreescribir cuantas veces se desee**, y utilizar en operaciones y funciones:

```
! ...
x = 2*x + x + 124.5d0
! ...
```

Este ejemplo calcula $2x + x + 124.5$*, y después lo guarda en la variable* x*,* ***olvidando su valor anterior****.*

Index

Leer y escribir por pantalla

Nuestro código puede **escribir** cualquier cosa por pantalla usando:

```
write(*,*) 'Hola. El valor de x es: ', x
```

```
print *, 'Hola. El valor de x es: ', x
```

- Las cadenas de texto se escriben entre ' '.
- También pueden escribirse variables (x, etc). Separar con comas.

Se puede pedir al usuario que introduzca por **teclado** un número y guardarlo en una variable "n" con:

```
read(*,*) n
```

Ejemplo 1: calcular el cuadrado de un número

Practicamos declaración y uso de variables, y las instrucciones **read** y **write**.

```
program ejemplo1
  implicit none
  real*8 :: x, cuadrado

  write(*,*) 'Por favor, introduzca un numero x'
  read(*,*) x   ! El usuario escribe número
  cuadrado = x*x   ! Calculamos x^2
  write(*,*) 'El numero introducido es: ', x
  write(*,*) 'Su cuadrado es: ', cuadrado
end program ejemplo1
```

Index

Operadores algebraicos (+,-,*,/,**)

Suma, resta, producto, cociente y potencia se escriben así: `+`, `-`, `*`, `/`, `**`. La potencia tiene **prioridad** sobre producto y cociente, y estos sobre suma y resta. Se pueden usar **paréntesis** para establecer el orden de las operaciones:

`5.+2.*4.**(-3)` es distinto de `((5.+2.)*4.)**(-3)`

¡OJO! EL COCIENTE DE ENTEROS PRODUCE COMO RESULTADO UN ENTERO. SI SE QUIERE UN RESULTADO REAL, NO OLVIDAR EL PUNTO NUNCA (o usar DBLE).

```
! ...
x = 2/3 ! El resultado es 0
x = 2./3. ! El resultado es 0.66666...
x = dble(2)/dble(3) ! El resultado es 0.66666...
! ...
```

*Imprescindibles I

- Usar **SIEMPRE** `implicit none`
- Si el exponente es **real** (simple o doble), la base ha de ser ≥ 0

```
! ...
x = (-2) ** (1./2.)   ! Da error
x = (-2) ** 3.   ! Da error también
x = (-2) ** 3   ! Esto sí funciona
! ...
```

- Leer **COMPLETAMENTE** los errores del compilador: **nos dicen qué es lo que pasa.**

*Imprescindibles II

- No olvidar **NUNCA** el punto "." en real*4 y "d" en real*8.

En fortran $1 \neq 1. \neq 1d0$

- No hacer líneas demasiado largas. Fortran sólo lee hasta la columna 132 (continuar línea con &).
- Escribir códigos profusamente **comentados** (!)