Programación en FORTRAN

Nivel Básico - Sesión 1

Martin Josemaría Vuelta Rojas

16 de enero de 2018

SoftButterfly

Contenido

Preliminares

Acerca del curso

El curso básico de Fortran tiene por objetivo

Acerca del curso

El curso básico de Fortran tiene por objetivo

- Que obtengan un conocimiento sólido de las carácteristicas de Fortran.

Acerca del curso

El curso básico de Fortran tiene por objetivo

- Que obtengan un conocimiento sólido de las carácteristicas de Fortran.
- Que se familiarizen con el flujo de infromación en los programas desarrolados en Fortran.

Acerca del curso

El curso básico de Fortran tiene por objetivo

- Que obtengan un conocimiento sólido de las carácteristicas de Fortran.
- Que se familiarizen con el flujo de infromación en los programas desarrolados en Fortran.
- Que tengan una introducción nuevas caracteristicas de los estándares recientes Fortran 2003 y 2008.

Acerca del curso

El curso básico de Fortran tiene por objetivo

- Que obtengan un conocimiento sólido de las carácteristicas de Fortran.
- Que se familiarizen con el flujo de infromación en los programas desarrolados en Fortran.
- Que tengan una introducción nuevas caracteristicas de los estándares recientes Fortran 2003 y 2008.
- Que puedan contruir su entorno de darrollo para programar en Fortran cómodamente.

Acerca del curso

Metodología

Acerca del curso

Metodología

- Exposiciones dialogadas utilizando, apuntes, elementos de proyección fija.

Acerca del curso

Metodología

- Exposiciones dialogadas utilizando, apuntes, elementos de proyección fija.
- Talleres prácticos grupales en la elaboración de programas.

Acerca del curso

Metodología

- Exposiciones dialogadas utilizando, apuntes, elementos de proyección fija.
- Talleres prácticos grupales en la elaboración de programas.
- Las clases serán eminentemente prácticas en una relación 70% práctica - 30% teórica.

Acerca del curso

Referencias

Acerca del curso

Referencias

- I. Chivers, J. Sleightholme, *Introduction to Programming with Fortran. With Coverage of Fortran 90, 95, 2003, 2008 and 77*, Springer-Verlag London, 2012.

Acerca del curso

Referencias

- I. Chivers, J. Sleightholme, *Introduction to Programming with Fortran. With Coverage of Fortran 90, 95, 2003, 2008 and 77*, Springer-Verlag London, 2012.
- Michael Metcalf, John Reid, Malcolm Cohen, *Modern Fortran Explained*, Oxford University Press, USA 2011

Acerca del curso

Referencias

- I. Chivers, J. Sleightholme, *Introduction to Programming with Fortran. With Coverage of Fortran 90, 95, 2003, 2008 and 77*, Springer-Verlag London, 2012.
- Michael Metcalf, John Reid, Malcolm Cohen, *Modern Fortran Explained*, Oxford University Press, USA 2011
- Morten Hjorth-Jensen, *Computational Physics, Lecture Notes Fall 2015*, Department of Physics, University of Oslo. 2015.

Acerca del curso

Materiales

Acerca del curso

Materiales

- Repositorio del curso

Acerca del curso

Materiales

Repositorio del curso

https://github.com/zodiacfireworks/course–fortran-basic

Acerca del curso

Materiales

Repositorio del curso

- https://github.com/zodiacfireworks/course–fortran-basic
- https://github.com/zodiacfireworks/course–fortran-intermediate

Acerca del curso

Materiales

Repositorio del curso

- https://github.com/zodiacfireworks/course–fortran-basic
- https://github.com/zodiacfireworks/course–fortran-intermediate

¡Importante!

Acerca del curso

Materiales

Repositorio del curso

- https://github.com/zodiacfireworks/course–fortran-basic
- https://github.com/zodiacfireworks/course–fortran-intermediate

¡Importante!

Cada alumno debera tener una cuenta en GitHub

Acerca del curso

Materiales

Repositorio del curso

- https://github.com/zodiacfireworks/course–fortran-basic
- https://github.com/zodiacfireworks/course–fortran-intermediate

¡Importante!

Cada alumno debera tener una cuenta en GitHub

- https://github.com

Acerca del instructor

Martín Josemaría Vuelta Rojas

Acerca del instructor

Martín Josemaría Vuelta Rojas

- Software Developer

Acerca del instructor

Martín Josemaría Vuelta Rojas

- Software Developer
- Web Developer

Acerca del instructor

Martín Josemaría Vuelta Rojas

- Software Developer
- Web Developer
- Investigador

Acerca del instructor

Martín Josemaría Vuelta Rojas

- Software Developer
- Web Developer
- Investigador
- Programador

Acerca del instructor

Martín Josemaría Vuelta Rojas

- Software Developer
- Web Developer
- Investigador
- Programador
  - En investigación:

Acerca del instructor

Martín Josemaría Vuelta Rojas

- Software Developer
- Web Developer
- Investigador
- Programador
  - En investigación:
    - C, C++, Fortran, Python, R, Julia, Mathematica, Matlab, LaTeX

Acerca del instructor

Martín Josemaría Vuelta Rojas

- Software Developer
- Web Developer
- Investigador
- Programador
  - En investigación:
    - C, C++, Fortran, Python, R, Julia, Mathematica, Matlab, LaTeX
  - En web:

Acerca del instructor

Martín Josemaría Vuelta Rojas

- Software Developer
- Web Developer
- Investigador
- Programador
  - En investigación:
    - C, C++, Fortran, Python, R, Julia, Mathematica, Matlab, LaTeX
  - En web:
    - HTML, CSS, JavaScript, Python

Acerca del instructor

Martín Josemaría Vuelta Rojas

- Software Developer
- Web Developer
- Investigador
- Programador
  - En investigación:
    - C, C++, Fortran, Python, R, Julia, Mathematica, Matlab, LaTeX
  - En web:
    - HTML, CSS, JavaScript, Python
  - En mobile:

Acerca del instructor

Martín Josemaría Vuelta Rojas

- Software Developer
- Web Developer
- Investigador
- Programador
  - En investigación:
    - C, C++, Fortran, Python, R, Julia, Mathematica, Matlab, LaTeX
  - En web:
    - HTML, CSS, JavaScript, Python
  - En mobile:
    - Kotlin, Java, C++

Acerca del instructor

Martín Josemaría Vuelta Rojas

- Software Developer
- Web Developer
- Investigador
- Programador
  - En investigación:
    - C, C++, Fortran, Python, R, Julia, Mathematica, Matlab, LaTeX
  - En web:
    - HTML, CSS, JavaScript, Python
  - En mobile:
    - Kotlin, Java, C++
  - Hobbie:

Acerca del instructor

Martín Josemaría Vuelta Rojas

- Software Developer
- Web Developer
- Investigador
- Programador
  - En investigación:
    - C, C++, Fortran, Python, R, Julia, Mathematica, Matlab, LaTeX
  - En web:
    - HTML, CSS, JavaScript, Python
  - En mobile:
    - Kotlin, Java, C++
  - Hobbie:
    - Scala, Pixie, Clojure, Elixir, Haskel, Oz, Kotlin, ...

Acerca del instructor

Martín Josemaría Vuelta Rojas

Acerca del instructor

Martín Josemaría Vuelta Rojas

- SoftButterfly

Acerca del instructor

Martín Josemaría Vuelta Rojas

- SoftButterfly
- HackSpace Perú

Acerca del instructor

Martín Josemaría Vuelta Rojas

- SoftButterfly
- HackSpace Perú
- Jupyter Notebook

Acerca del instructor

Martín Josemaría Vuelta Rojas

- SoftButterfly
- HackSpace Perú
- Jupyter Notebook
- Fedora

Acerca del instructor

Martín Josemaría Vuelta Rojas

- SoftButterfly
- HackSpace Perú
- Jupyter Notebook
- Fedora
- GNOME

Acerca del instructor

Martín Josemaría Vuelta Rojas

- SoftButterfly
- HackSpace Perú
- Jupyter Notebook
- Fedora
- GNOME
- UNMSM

Acerca del instructor

Martín Josemaría Vuelta Rojas

Acerca del instructor

Martín Josemaría Vuelta Rojas

+51 982 042 088

Acerca del instructor

Martín Josemaría Vuelta Rojas

+51 982 042 088

martin.vuelta@gmail.com

Acerca del instructor

Martín Josemaría Vuelta Rojas

+51 982 042 088

martin.vuelta@gmail.com

martinvuelta

Acerca del instructor

Martín Josemaría Vuelta Rojas

+51 982 042 088

martin.vuelta@gmail.com

martinvuelta

zodiacfireworks

Introducción histórica

Orígenes

En el origen ...

Orígenes

En el origen ...

- Código máquina en notación octal

Orígenes

En el origen ...

- Código máquina en notación octal
- Conocimiento muy detallado del hardware

Orígenes

En el origen ...

- Código máquina en notación octal
- Conocimiento muy detallado del hardware

A inicios de los 50s

Orígenes

En el origen ...

- Código máquina en notación octal
- Conocimiento muy detallado del hardware

A inicios de los 50s

- Assembler

Orígenes

En el origen ...

- Código máquina en notación octal
- Conocimiento muy detallado del hardware

A inicios de los 50s

- Assembler
- Menos laborioso que el código máquina

Orígenes

En el origen ...

- Código máquina en notación octal
- Conocimiento muy detallado del hardware

A inicios de los 50s

- Assembler
- Menos laborioso que el código máquina
- Conocimiento detallado del hardware

Orígenes

En el origen ...

- Código máquina en notación octal
- Conocimiento muy detallado del hardware

A inicios de los 50s

- Assembler
- Menos laborioso que el código máquina
- Conocimiento detallado del hardware

El panorama general ...

Orígenes

En el origen ...

- Código máquina en notación octal
- Conocimiento muy detallado del hardware

A inicios de los 50s

- Assembler
- Menos laborioso que el código máquina
- Conocimiento detallado del hardware

El panorama general ...

- Conocimiento del harware

Orígenes

En el origen ...

- Código máquina en notación octal
- Conocimiento muy detallado del hardware

A inicios de los 50s

- Assembler
- Menos laborioso que el código máquina
- Conocimiento detallado del hardware

El panorama general ...

- Conocimiento del harware
- Facilidad de cometer errores

Orígenes

En el origen ...

- Código máquina en notación octal
- Conocimiento muy detallado del hardware

A inicios de los 50s

- Assembler
- Menos laborioso que el código máquina
- Conocimiento detallado del hardware

El panorama general ...

- Conocimiento del harware
- Facilidad de cometer errores
- Encontrar los errores en los programas era bastante difícil

Orígenes

Génesis 1953 ...

Orígenes

Génesis 1953 ...

John Backus envía una carta a su jefe en IBM pidiendo permiso para investigar una *mejor forma* de programar las computadoras. La carta contenia un esboo de projecto con un tiempo de desarrollo de 6 meses.

Orígenes

Génesis 1953 ...

John Backus envía una carta a su jefe en IBM pidiendo permiso para investigar una *mejor forma* de programar las computadoras. La carta contenia un esboo de projecto con un tiempo de desarrollo de 6 meses.

Así emepzo el proyecto que daría origen a Fortran

Orígenes

"The project completion was always six months away!"
John Backus

Orígenes

1957

Orígenes

1957

En febrero FORTRAN, el primer lenguaje de programación de alto nivel, fue anunciado al mundo por John Backus y su equipo de IBM en la Western Joint Computer Conference celebrada en Los Ángeles.

Orígenes

1957

En febrero FORTRAN, el primer lenguaje de programación de alto nivel, fue anunciado al mundo por John Backus y su equipo de IBM en la Western Joint Computer Conference celebrada en Los Ángeles.

A mediados de abril de 1957 tuvo lugar la primera entrega del compilador de FORTRAN para IBM 704 a Westinghouse Bettis para su uso en el diseño de reactores nucleares.

Las versiones de FORTRAN

Orígenes

Las versiones de FORTRAN

1957 FORTRAN I

Orígenes

Las versiones de FORTRAN

1957 FORTRAN I

1958 FORTRAN II

Orígenes

Las versiones de FORTRAN

1957 FORTRAN I

1958 FORTRAN II

1958 FORTRAN III (No disponible al público)

Orígenes

Las versiones de FORTRAN

1957 FORTRAN I

1958 FORTRAN II

1958 FORTRAN III (No disponible al público)

1961 FORTRAN IV (Una versión mejorada de FORTRAN II)

Algoritmo TPK en FORTRAN 0

```
1           DIMENSION A(11)
2           READ A
3     2     DO 3,8,11 J=1,11
4     3     I=11-J
5           Y=SQRT(ABS(A(I+1)))+5*A(I+1)**3
6           IF (400>=Y) 8,4
7     4     PRINT I,999.
8           GOTO 2
9     8     PRINT I,Y
10    11    STOP
```

Algoritmo TPK en FORTRAN I

```
C       THE TPK ALGORITHM
C       FORTRAN I STYLE
        FUNF(T)=SQRTF(ABSF(T))+5.0*T**3
        DIMENSION A(11)
    1   FORMAT(6F12.4)
        READ 1,A
        DO 10 J=1,11
        I=11-J
        Y=FUNF(A(I+1))
        IF(400.0-Y)4,8,8
    4   PRINT 5,I
    5   FORMAT(I10,10H TOO LARGE)
        GOTO 10
    8   PRINT 9,I,Y
    9   FORMAT(I10,F12.7)
   10   CONTINUE
        STOP 52525
```

La estándarización

La estándarización

1962 El primer comité de estandarización de la ASA (Ahora ANSI) se re reune.

La estándarización

1962 El primer comité de estandarización de la ASA (Ahora ANSI) se re reune.

1966 Publicación del ANSI X3.91966 (FORTRAN 66), el primer estándar.

La estándarización

1962 El primer comité de estandarización de la ASA (Ahora ANSI) se re reune.

1966 Publicación del ANSI X3.91966 (FORTRAN 66), el primer estándar.

1978 Publicación del ANSI X3.91978 (FORTRAN 77), tambien publicado como ISO 1539:1980.

La estándarización

1962 El primer comité de estandarización de la ASA (Ahora ANSI) se re reune.

1966 Publicación del ANSI X3.91966 (FORTRAN 66), el primer estándar.

1978 Publicación del ANSI X3.91978 (FORTRAN 77), tambien publicado como ISO 1539:1980.

1991 ISO/IEC 1539:1991 (Fortran 90)

La estándarización

1962 El primer comité de estandarización de la ASA (Ahora ANSI) se re reune.

1966 Publicación del ANSI X3.91966 (FORTRAN 66), el primer estándar.

1978 Publicación del ANSI X3.91978 (FORTRAN 77), tambien publicado como ISO 1539:1980.

1991 ISO/IEC 1539:1991 (Fortran 90)

1997 ISO/IEC 15391:1997 (Fortran 95)

La estándarización

1962 El primer comité de estandarización de la ASA (Ahora ANSI) se re reune.

1966 Publicación del ANSI X3.91966 (FORTRAN 66), el primer estándar.

1978 Publicación del ANSI X3.91978 (FORTRAN 77), tambien publicado como ISO 1539:1980.

1991 ISO/IEC 1539:1991 (Fortran 90)

1997 ISO/IEC 15391:1997 (Fortran 95)

2004 ISO/IEC 15391:2004 (Fortran 2003)

La estándarización

1962 El primer comité de estandarización de la ASA (Ahora ANSI) se re reune.

1966 Publicación del ANSI X3.91966 (FORTRAN 66), el primer estándar.

1978 Publicación del ANSI X3.91978 (FORTRAN 77), tambien publicado como ISO 1539:1980.

1991 ISO/IEC 1539:1991 (Fortran 90)

1997 ISO/IEC 15391:1997 (Fortran 95)

2004 ISO/IEC 15391:2004 (Fortran 2003)

2010 ISO/IEC 15391:2010 (Fortran 2008)

Algoritmo TPK en FORTRAN 77

```
1            PROGRAM TPK
2     C      THE TPK ALGORITHM
3     C      FORTRAN 77 STYLE
4            REAL A(0:10)
5            READ (5,*) A
6            DO 10 I = 10, 0, -1
7                    Y = FUN(A(I))
8                    IF ( Y . LT. 400) THEN
9                            WRITE(6,9) I,Y
10       9                   FORMAT(I10. F12.6)
11                   ELSE
12                           WRITE (6,5) I
13       5                   FORMAT(I10,' TOO LARGE')
14                   ENDIF
15       10      CONTINUE
16           END
17
18           REAL FUNCTION FUN(T)
19           REAL T
20           FUN = SQRT(ABS(T)) + 5.0*T**3
21           END
```

Algoritmo TPK en FORTRAN 90

```
1           PROGRAM TPK
2     !     The TPK Algorithm
3     !     Fortran 90 style
4           IMPLICIT NONE
5           INTEGER           :: I
6           REAL                        :: Y
7           REAL, DIMENSION(0:10)       :: A
8           READ (*,*) A
9           DO I = 10, 0, -1            ! Backwards
10                 Y = FUN(A(I))
11                 IF ( Y < 400.0 ) THEN
12                       WRITE(*,*) I, Y
13                 ELSE
14                       WRITE(*,*) I, ' Too large'
15                 END IF
16          END DO
17          CONTAINS                    ! Local function
18                 FUNCTION FUN(T)
19                 REAL  :: FUN
20                 REAL, INTENT(IN) :: T
21                 FUN = SQRT(ABS(T)) + 5.0*T**3
22                 END FUNCTION FUN
23          END PROGRAM TPK
```

Algoritmo TPK en FORTRAN 90

```
1          module Functions
2          public :: fun
3          contains
4             function fun(t) result (r)
5                real, intent(in) :: t
6                real  :: r
7                r = sqrt(abs(t)) + 5.0*t**3
8             end function fun
9          end module Functions
10
11         program TPK
12   !     The TPK Algorithm
13   !     F95 style
14         use Functions
15         integer       :: i
16         real                    :: y
17         real, dimension(0:10) :: a
18         read *, a
19         do i = 10, 0, -1       ! Backwards
20            y = fun(a(i))
21            if ( y < 400.0 ) then
22               print *, i, y
23            else
24               print *, i, " Too large"
```

En la actualidad ...

"I don't know what the programming language of the year 2000 will look like, but I know it will be called FORTRAN."
Charles Anthony Richard Hoare
Circa 1982

En la actualidad ...

Aplicaciones

En la actualidad ...

Aplicaciones

- Predicción del clima

En la actualidad ...

Aplicaciones

- Predicción del clima
- Análisis de datos de sísmicos para la exploración de depositos de gas y petroleo

En la actualidad ...

Aplicaciones

- Predicción del clima
- Análisis de datos de sísmicos para la exploración de depositos de gas y petroleo
- Análisis fianciero

En la actualidad ...

Aplicaciones

- Predicción del clima
- Análisis de datos de sísmicos para la exploración de depositos de gas y petroleo
- Análisis fianciero
- Simulacion de choques vehiculares

En la actualidad ...

Aplicaciones

- Predicción del clima
- Análisis de datos de sísmicos para la exploración de depositos de gas y petroleo
- Análisis fianciero
- Simulacion de choques vehiculares
- Análisis de datos de sondas espaciales

En la actualidad ...

Aplicaciones

- Predicción del clima
- Análisis de datos de sísmicos para la exploración de depositos de gas y petroleo
- Análisis fianciero
- Simulacion de choques vehiculares
- Análisis de datos de sondas espaciales
- Modelación de armas nucleares

En la actualidad ...

Aplicaciones

- Predicción del clima
- Análisis de datos de sísmicos para la exploración de depositos de gas y petroleo
- Análisis fianciero
- Simulacion de choques vehiculares
- Análisis de datos de sondas espaciales
- Modelación de armas nucleares
- Dinámica de fluidos computacionales

En la actualidad ...

Aplicaciones

- Predicción del clima
- Análisis de datos de sísmicos para la exploración de depositos de gas y petroleo
- Análisis fianciero
- Simulacion de choques vehiculares
- Análisis de datos de sondas espaciales
- Modelación de armas nucleares
- Dinámica de fluidos computacionales
- “Numerical Wind Tunnel”

En la actualidad ...

En la actualidad ...

- Intel

En la actualidad ...

- Intel
- IBM

En la actualidad ...

- Intel
- IBM
- NVIDIA

El futuro

El futuro

- **WG5:** https://wg5-fortran.org/

El futuro

- **WG5:** https://wg5-fortran.org/
- **Fortran Wiki:** http://fortranwiki.org/

El futuro

- **WG5:** https://wg5-fortran.org/
- **Fortran Wiki:** http://fortranwiki.org/
- **WG5:** https://wg5-fortran.org/

Entorno de desarrollo

Entorno de desarrollo

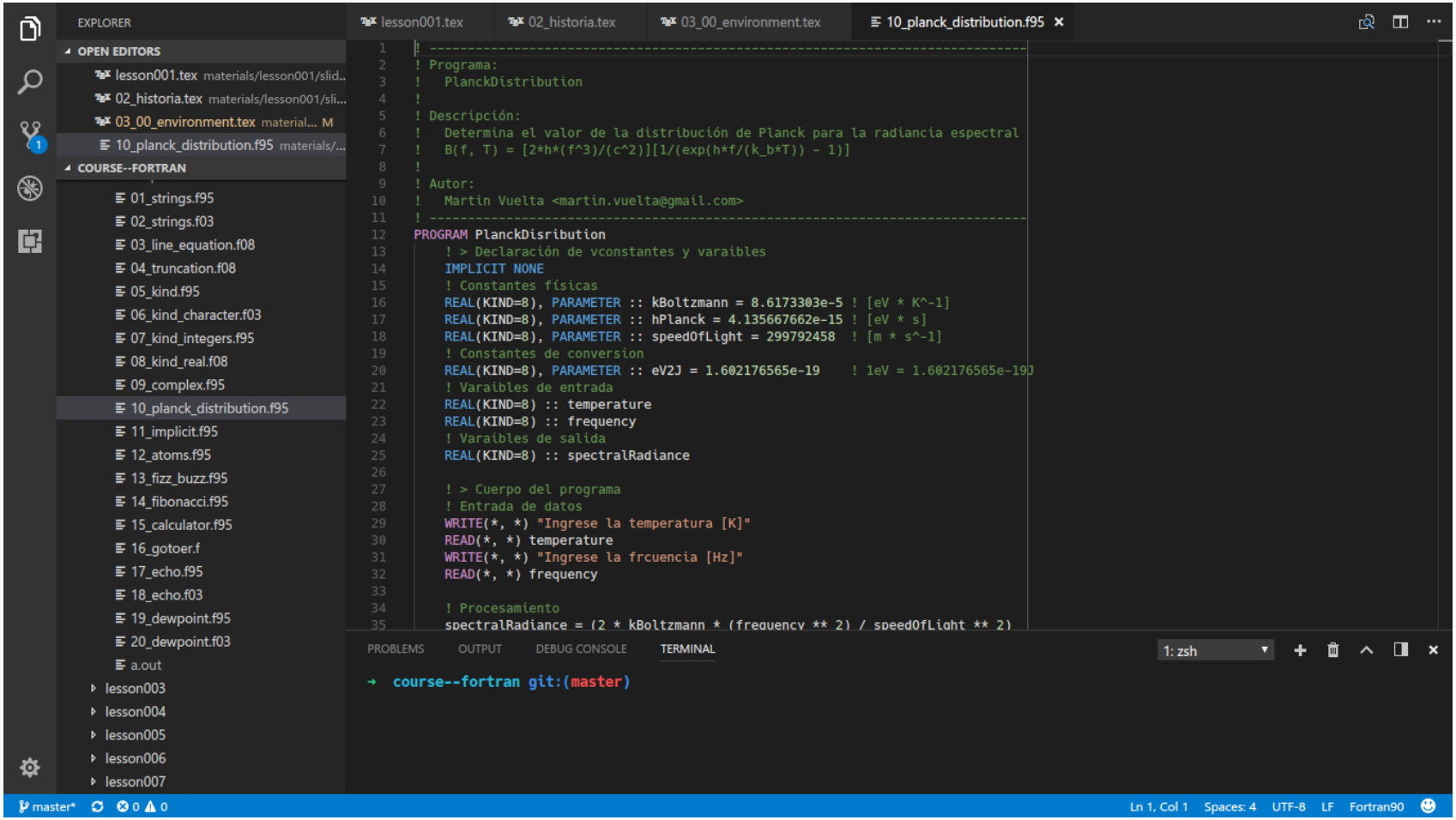


Figura 1: Entorno de desarrollo VS Code

Entorno de desarrollo

Que tiene un entorno de desarrollo

Entorno de desarrollo

Que tiene un entorno de desarrollo

- Editor de código

Entorno de desarrollo

Que tiene un entorno de desarrollo

- Editor de código
- Compiladores o intérpretes

Entorno de desarrollo

Que tiene un entorno de desarrollo

- Editor de código
- Compiladores o intérpretes
- Debugger

Entorno de desarrollo

Que tiene un entorno de desarrollo

- Editor de código
- Compiladores o intérpretes
- Debugger
- Otras utilidades

Entorno de desarrollo

Editor de código

Entorno de desarrollo

Editor de código

- VS Code + Fortran Package [✔]

Entorno de desarrollo

Editor de código

- VS Code + Fortran Package [✔]
- Sublime Text

Entorno de desarrollo

Editor de código

- VS Code + Fortran Package [✔]
- Sublime Text
- Atom

Entorno de desarrollo

Editor de código

- VS Code + Fortran Package [✔]
- Sublime Text
- Atom
- Vim

Entorno de desarrollo

Editor de código

- VS Code + Fortran Package [✔]
- Sublime Text
- Atom
- Vim
- Emacs

Entorno de desarrollo

Editor de código

- VS Code + Fortran Package [✔]
- Sublime Text
- Atom
- Vim
- Emacs
- ...

Entorno de desarrollo

Compilador

Entorno de desarrollo

Compilador

- GFortran [✔]

Entorno de desarrollo

Compilador

- GFortran [✔]
- Intel

Entorno de desarrollo

Compilador

- GFortran [✔]
- Intel
- IBM

Entorno de desarrollo

Compilador

- GFortran [✔]
- Intel
- IBM
- Oracle

Entorno de desarrollo

Compilador

- GFortran [✔]
- Intel
- IBM
- Oracle
- ...

Entorno de desarrollo

Debuggers

Entorno de desarrollo

Debuggers

- gdb [✔]

Entorno de desarrollo

Debuggers

- gdb [✔]
- idb

Entorno de desarrollo

Debuggers

- gdb [✔]
- idb
- ddd

Entorno de desarrollo

Debuggers

- gdb [✔]
- idb
- ddd
- totalview

Entorno de desarrollo

Debuggers

- gdb [✔]
- idb
- ddd
- totalview
- ...

Instalación de editor de código

VS Code en Ubuntu y derivados

Instalación de editor de código

VS Code en Ubuntu y derivados

- Descargar el paquete .deb desde
  https://code.visualstudio.com/download

Instalación de editor de código

VS Code en Ubuntu y derivados

- Descargar el paquete .deb desde https://code.visualstudio.com/download
- Instalar el paquete desde la terminal

```
sudo dpkg -i <nombre del archivo>.deb
sudo apt-get install -f
```

Instalación de editor de código

VS Code en Ubuntu y derivados

- Descargar el paquete .deb desde https://code.visualstudio.com/download
- Instalar el paquete desde la terminal

```
sudo dpkg -i <nombre del archivo>.deb
sudo apt-get install -f
```

- Actualizar el paquete e instalar code

```
sudo apt-get update
sudo apt-get install code # o también  "code-insiders"
```

Instalación de editor de código

VS Code en Fedora/CentOS

Instalación de editor de código

VS Code en Fedora/CentOS

- Descargar el paquete .rpm desde
  https://code.visualstudio.com/download

Instalación de editor de código

VS Code en Fedora/CentOS

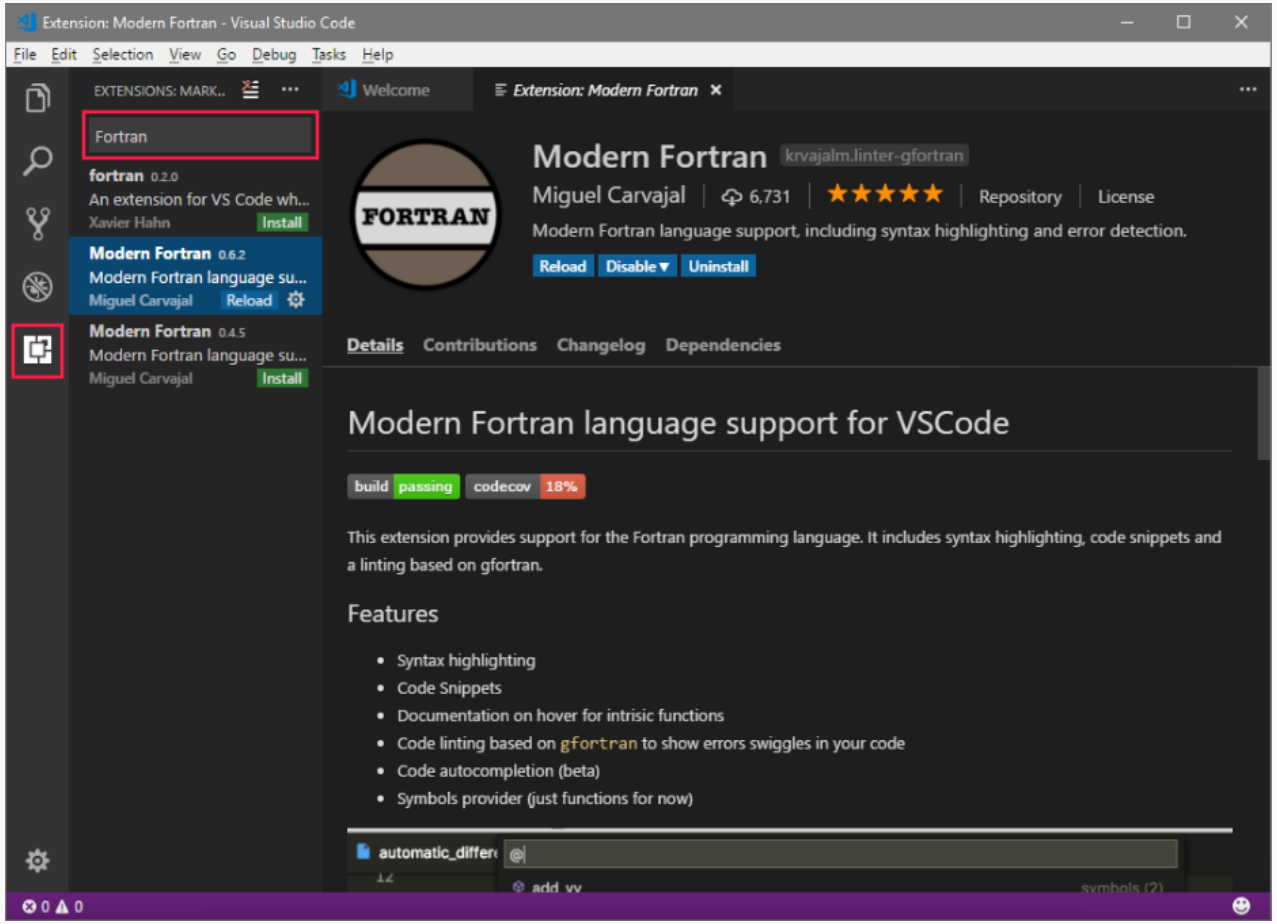
- Descargar el paquete .rpm desde https://code.visualstudio.com/download
- Instalar el paquete desde la terminal según la versión Fedora/CentOs (yum o dnf)

```
yum check-update
sudo yum install <nombre del archivo>.rpm
```

```
dnf check-update
sudo dnf install <nombre del archivo>.rpm
```

Fortran en VS Code

- En la pestaña 'Extensiones' (Ctrl+Shift+X), instalar 'Modern Fortran 0.6.2' encontrado con el buscador y recargar la extensión.



Extensión Modern Fortran 0.6.2 en VS Code

Instalación de compilador y debuger

Compilador y debugger en Ubuntu y derivados

Instalación de compilador y debuger

Compilador y debugger en Ubuntu y derivados

- **Instalación de Gfortran**

```
1 sudo apt-get install gfortran
```

Instalación de compilador y debuger

Compilador y debugger en Ubuntu y derivados

- **Instalación de Gfortran**

```
1 sudo apt-get install gfortran
```

- **Instalación del paquete binutils**

```
1 sudo apt-get update
2 sudo apt-get install binutils
```

Instalación de compilador y debuger

Compilador y debugger en Ubuntu y derivados

- **Instalación de Gfortran**

```
sudo apt-get install gfortran
```

- **Instalación del paquete binutils**

```
sudo apt-get update
sudo apt-get install binutils
```

- **Instalación del paquete build-essential**

```
sudo apt-get update
sudo apt-get install build-essential
```

Instalación de compilador y debuger

Compilador y debugger en Fedora/CentOS

Instalación de compilador y debuger

Compilador y debugger en Fedora/CentOS

- **Instalación de Gfortran**

```
1 yum install gcc-gfortran
```

Instalación de compilador y debuger

Compilador y debugger en Fedora/CentOS

- **Instalación de Gfortran**

```
yum install gcc-gfortran
```

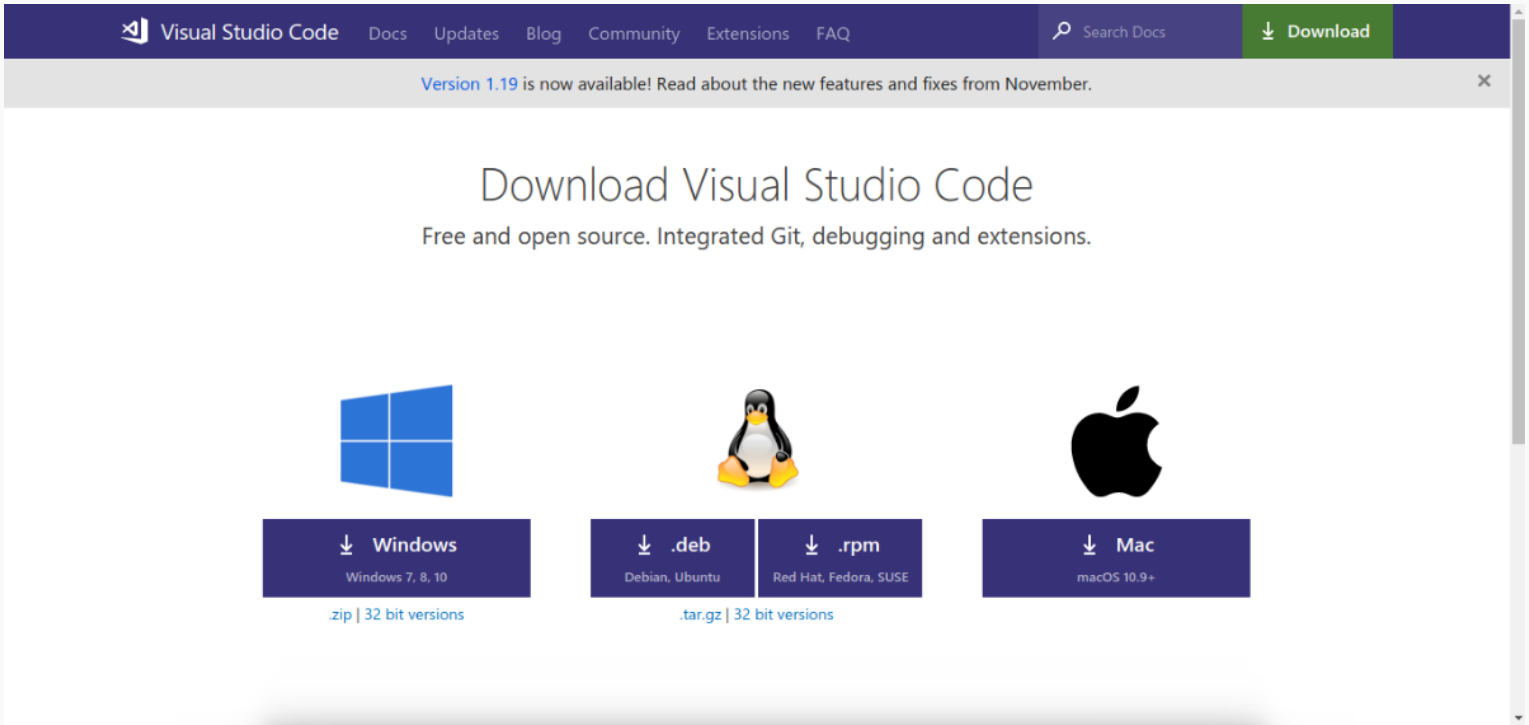
- **Instalación del paquete Development tools**

```
yum clean all
yum groupinstall "Development tools"
```

Instalación de editor de código

VS Code en Windows

- Descargar el instalador desde
  `https://code.visualstudio.com/download`

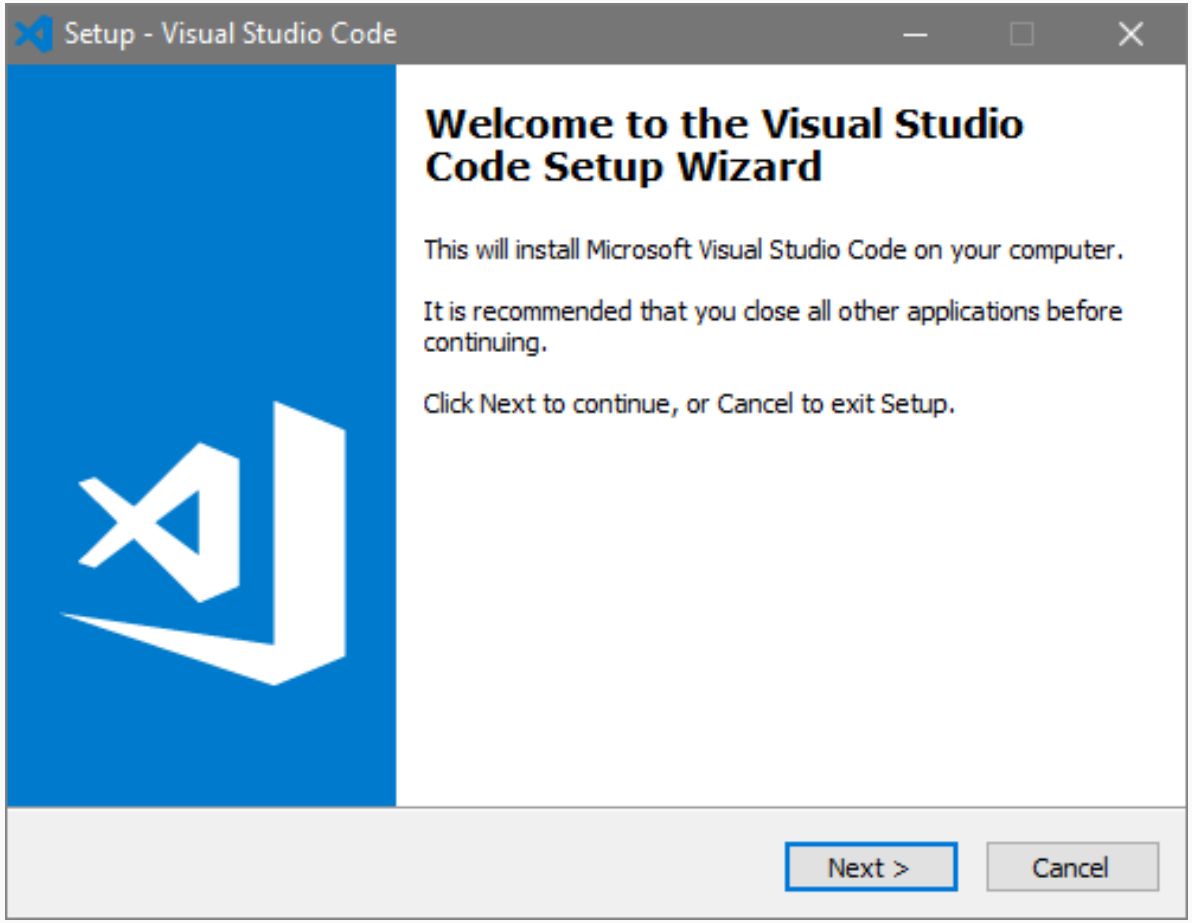


Página oficial de Visual Studio Code

Instalación de editor de código

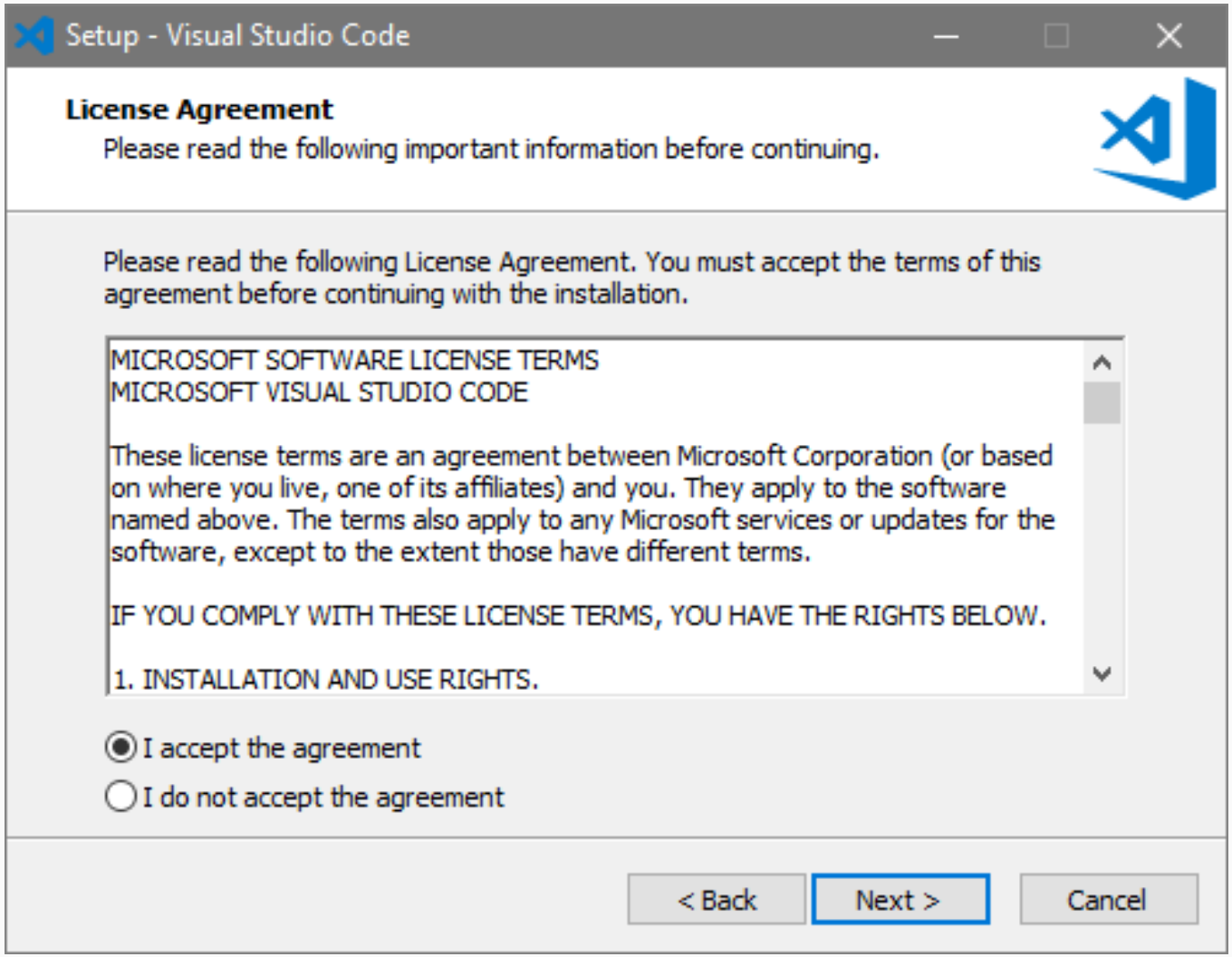
VS Code en Windows

- Iniciar el instalador y seguir los siguientes pasos:



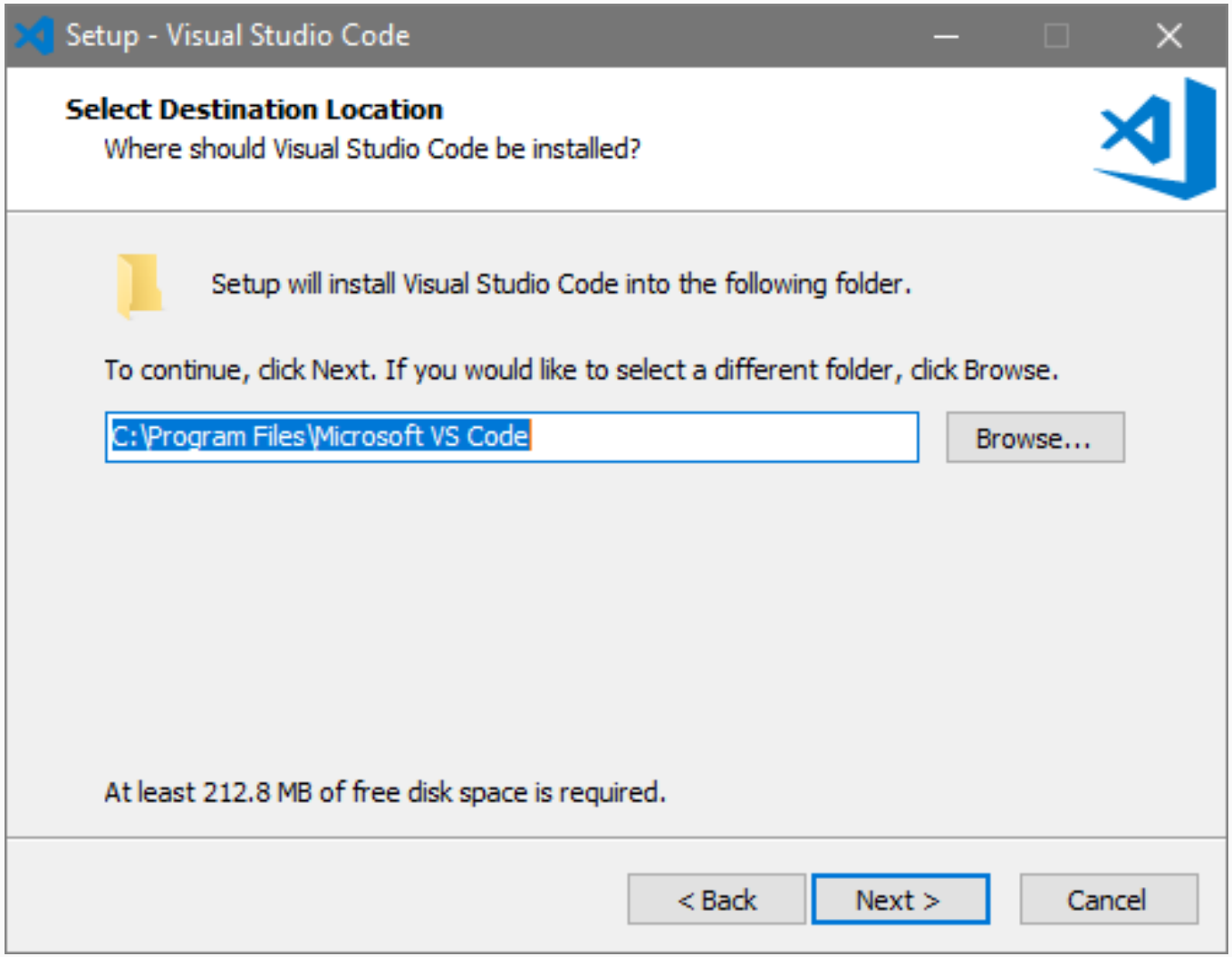
Instalación de editor de código

VS Code en Windows



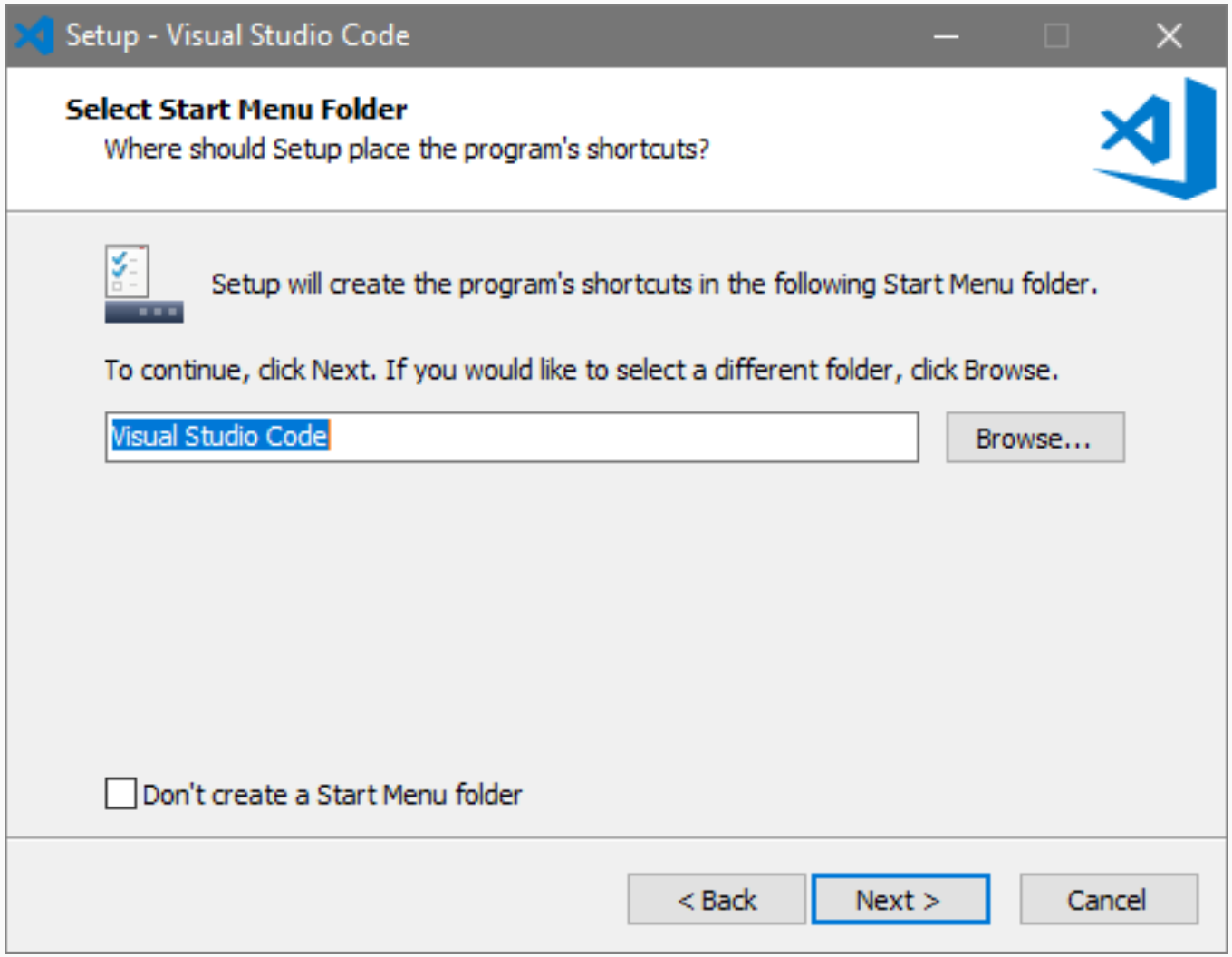
Instalación de editor de código

VS Code en Windows



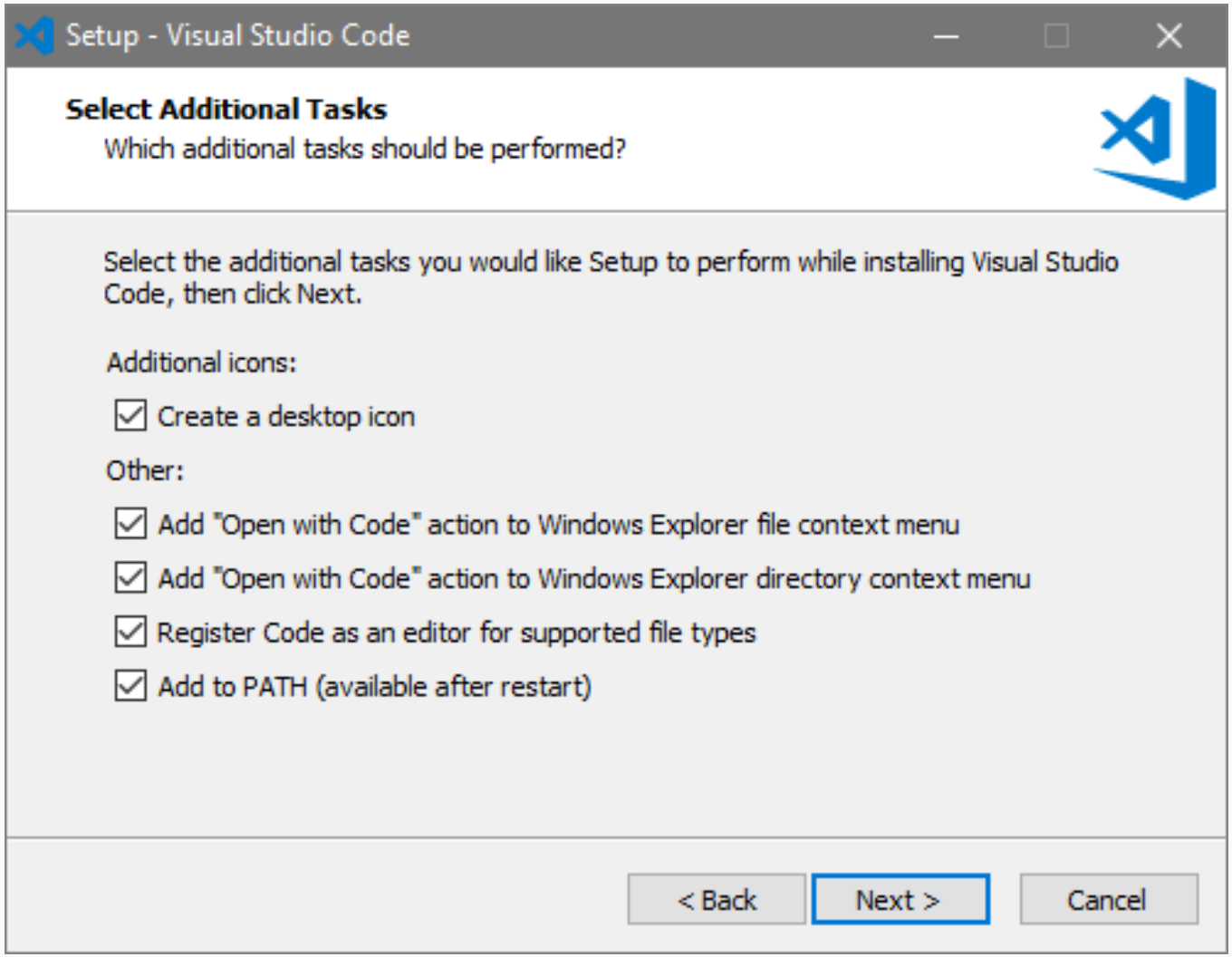
Instalación de editor de código

VS Code en Windows



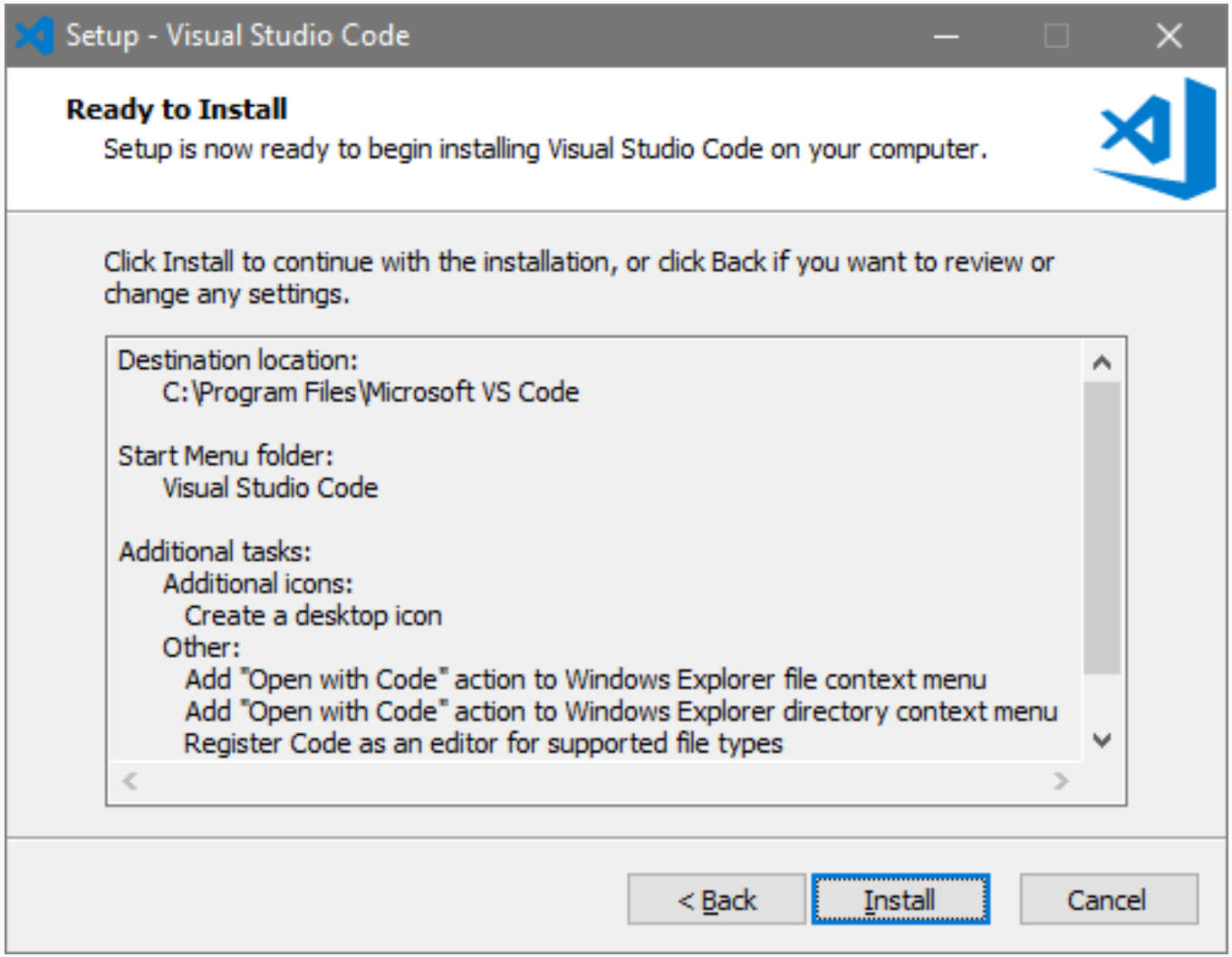
Instalación de editor de código

VS Code en Windows



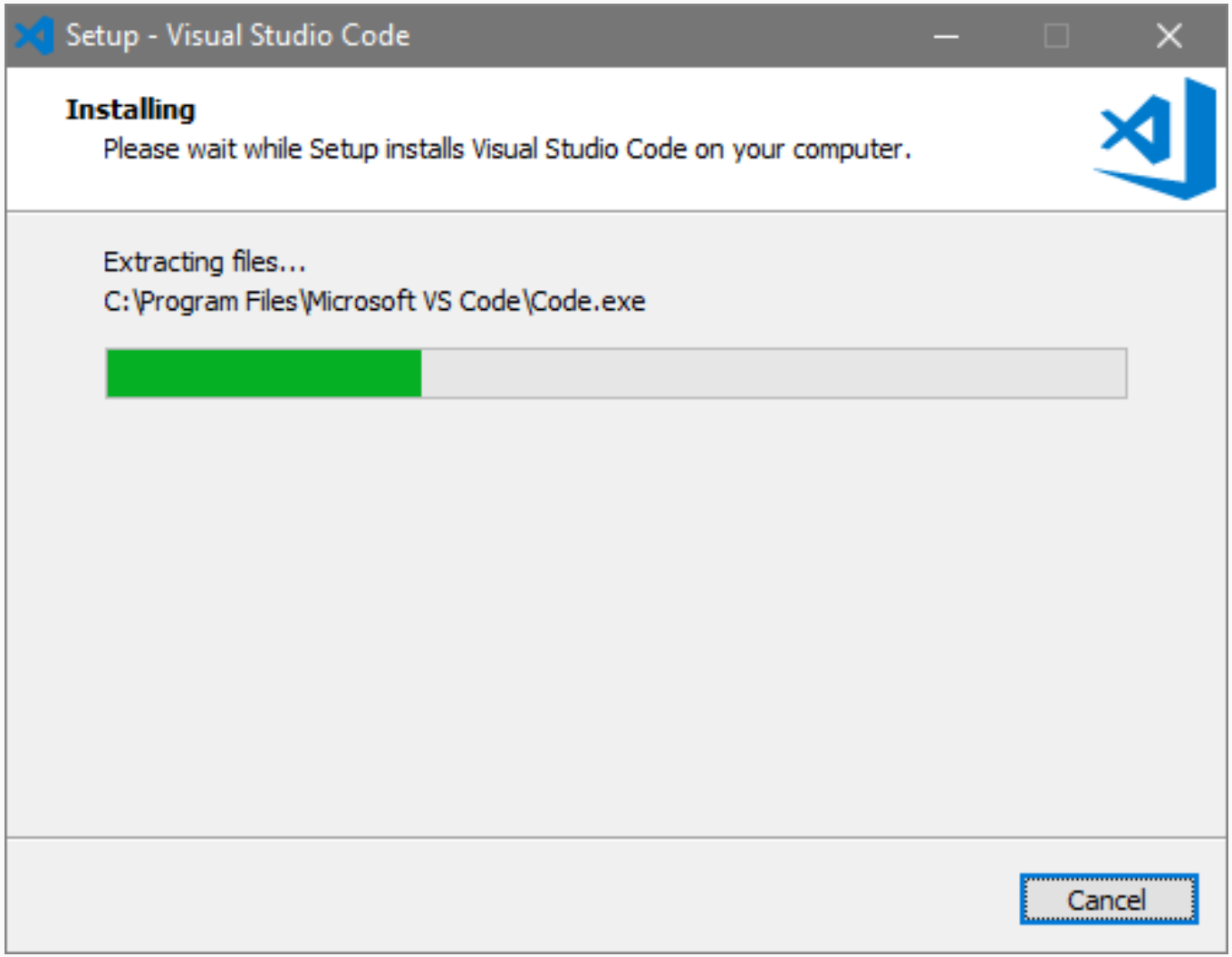
Instalación de editor de código

VS Code en Windows



Instalación de editor de código

VS Code en Windows



Instalación de editor de código

VS Code en Windows

Instalación de compilador y debuger

Instalación del compilador TDM-GCC

Instalación de compilador y debuger

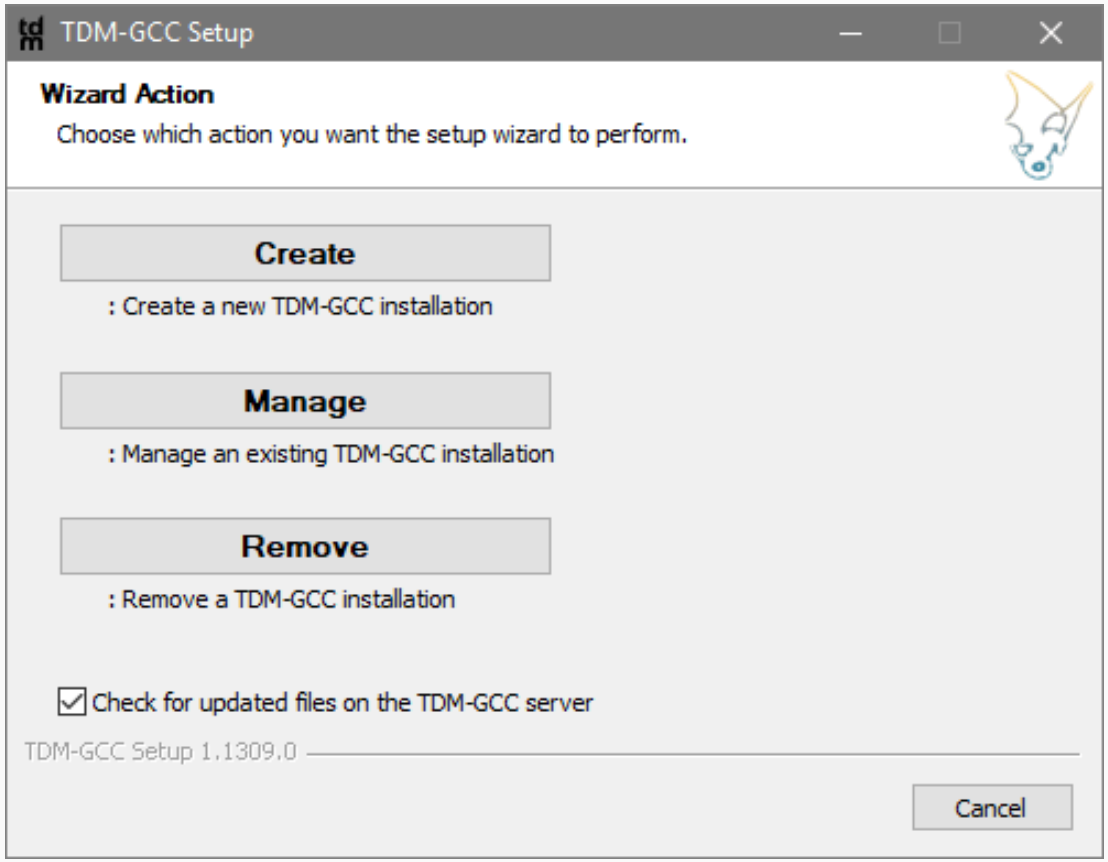
Instalación del compilador TDM-GCC

- Descargar el paquete TDM-GCC según la versión de windows (32-64 bits) en http://tdm-gcc.tdragon.net/

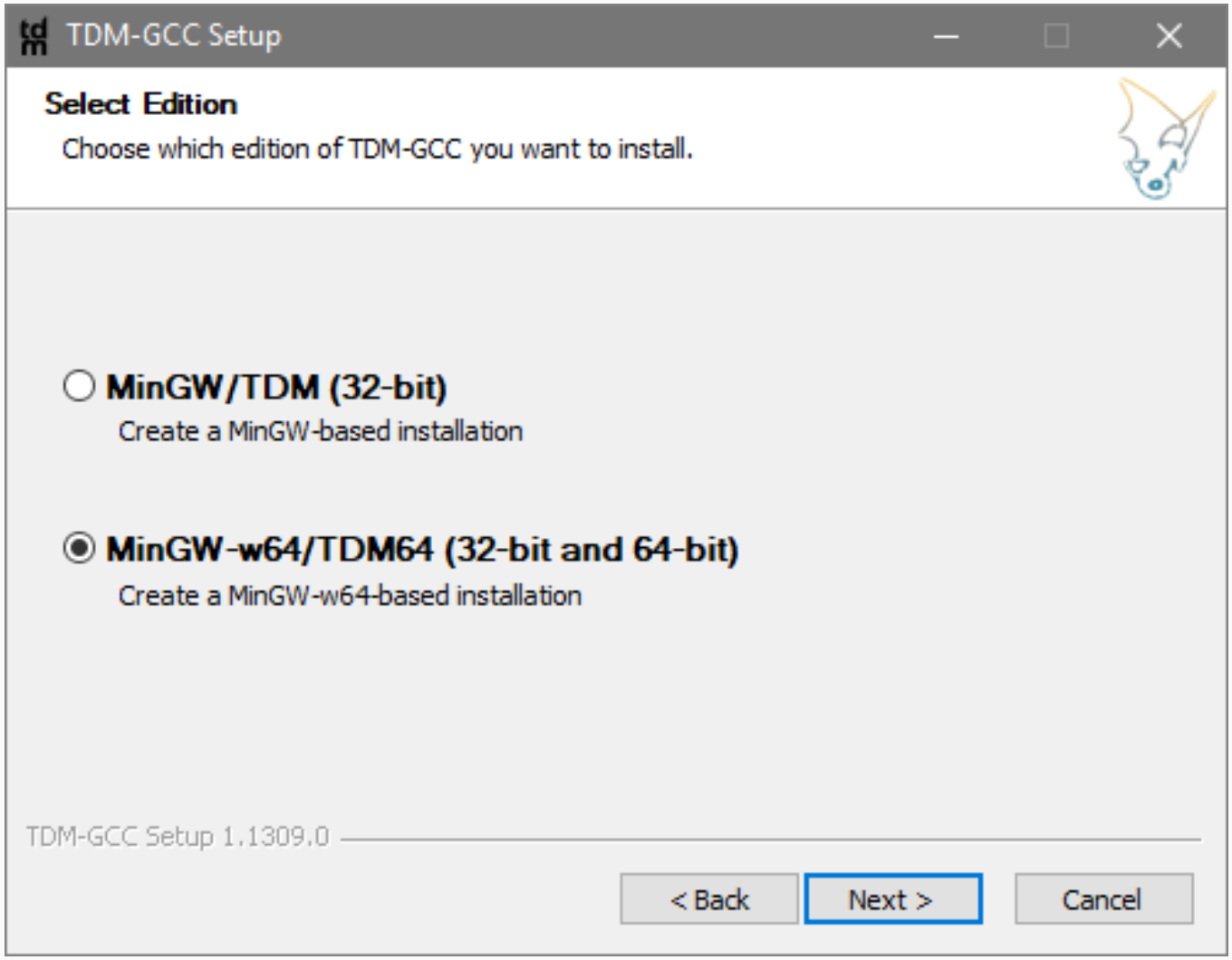
Instalación de compilador y debuger

Instalación del compilador TDM-GCC

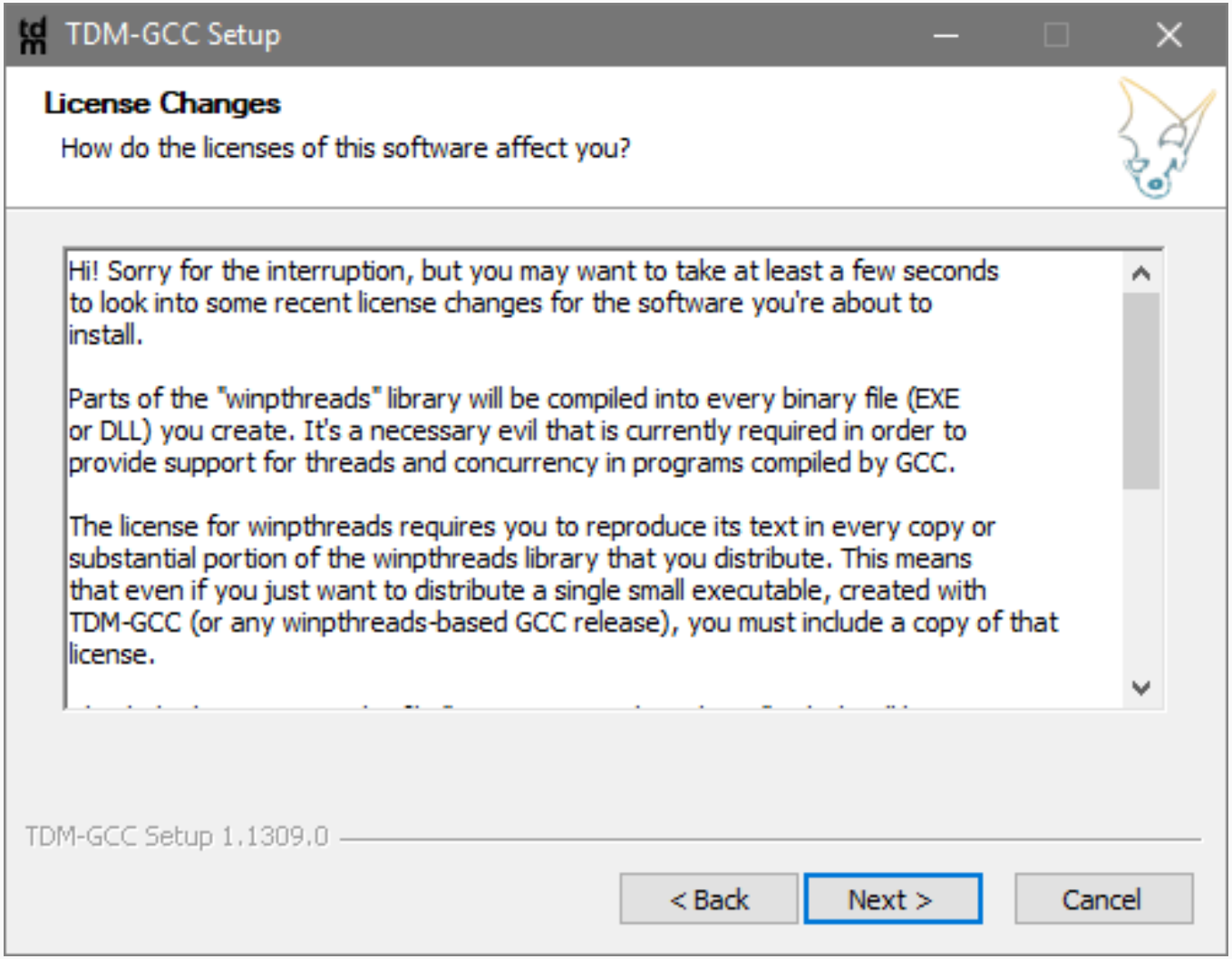
- Descargar el paquete TDM-GCC según la versión de windows (32-64 bits) en http://tdm-gcc.tdragon.net/
- Iniciar el instalador y seguir los siguientes pasos:



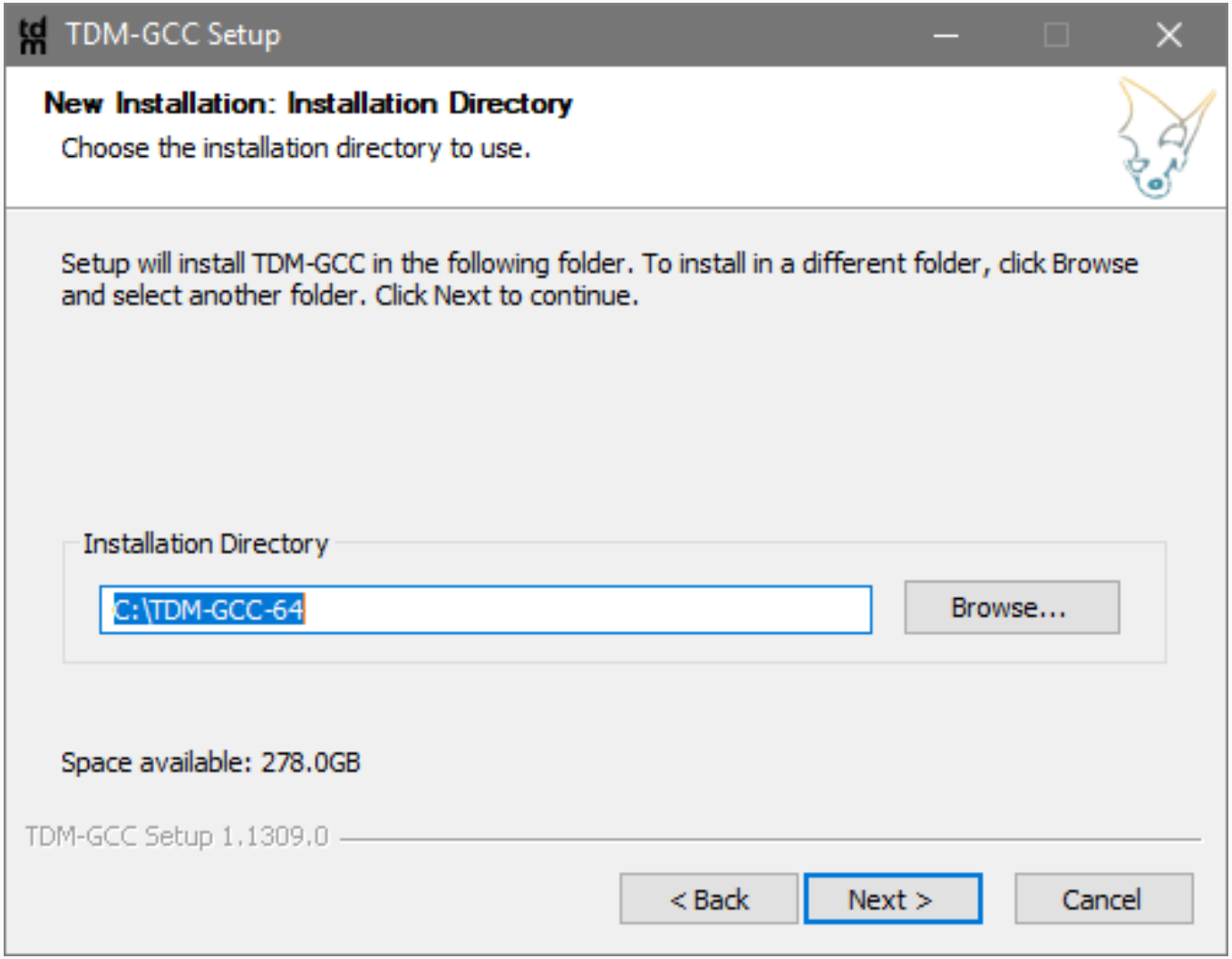
Instalación del compilador TDM-GCC



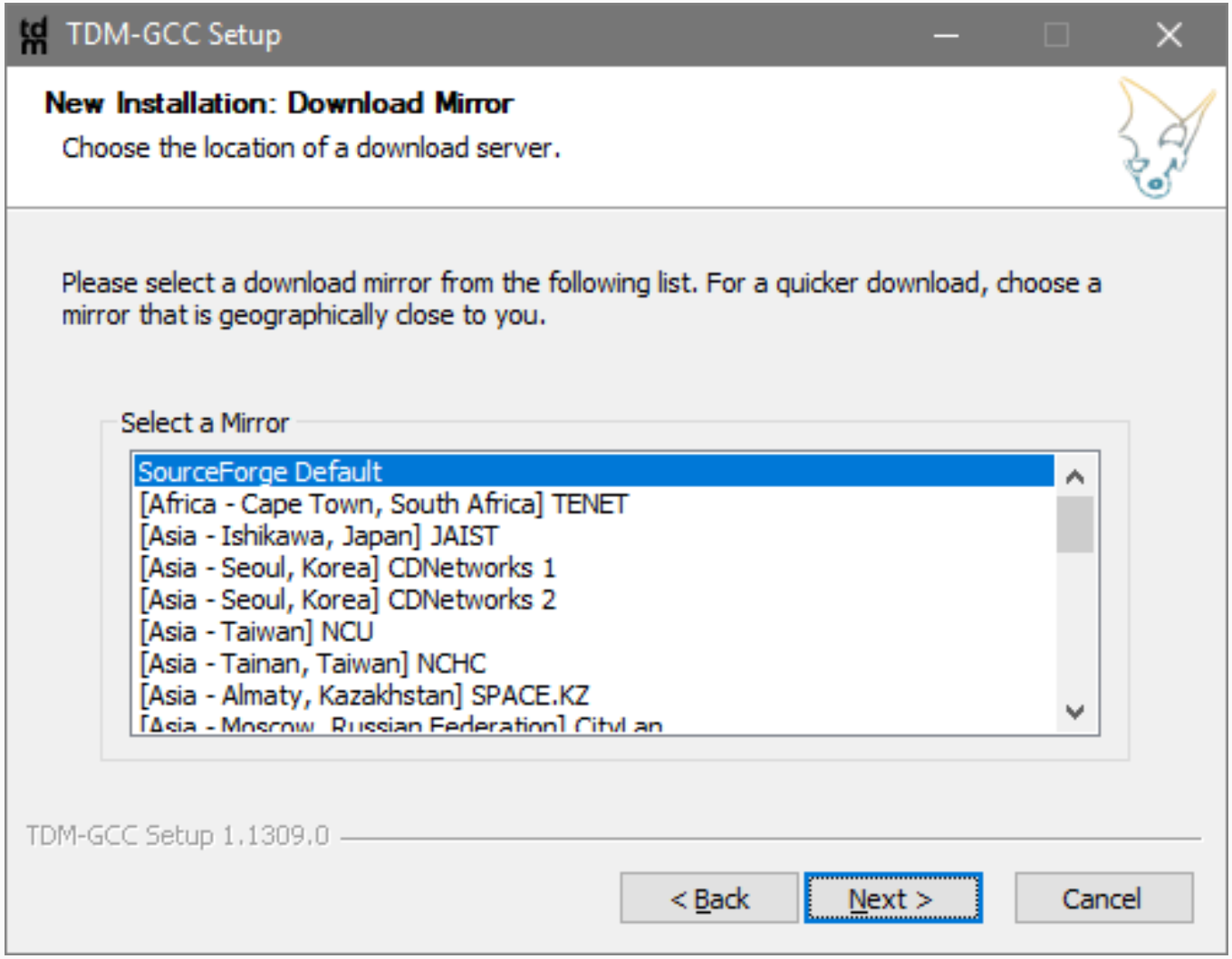
Instalación del compilador TDM-GCC



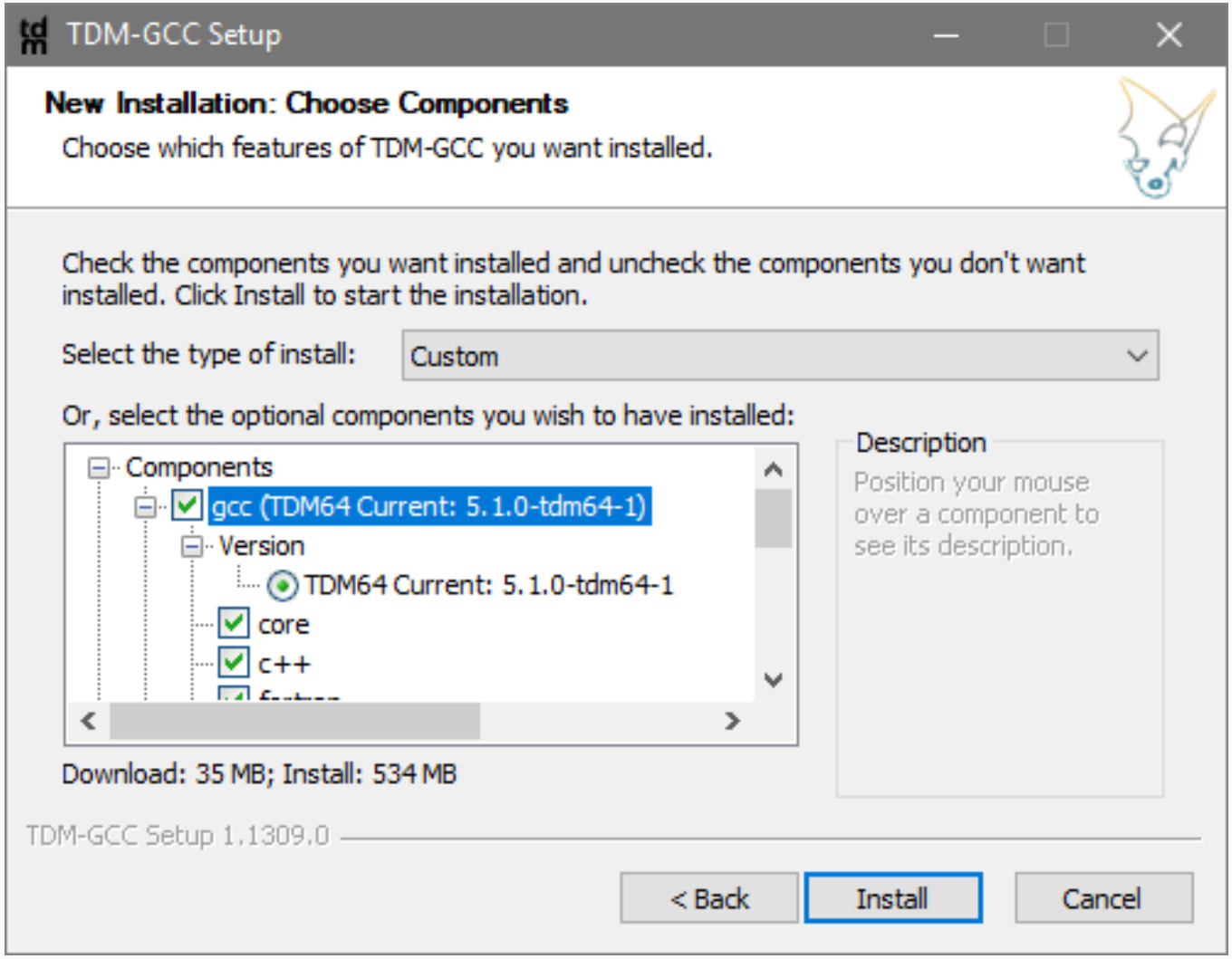
Instalación del compilador TDM-GCC



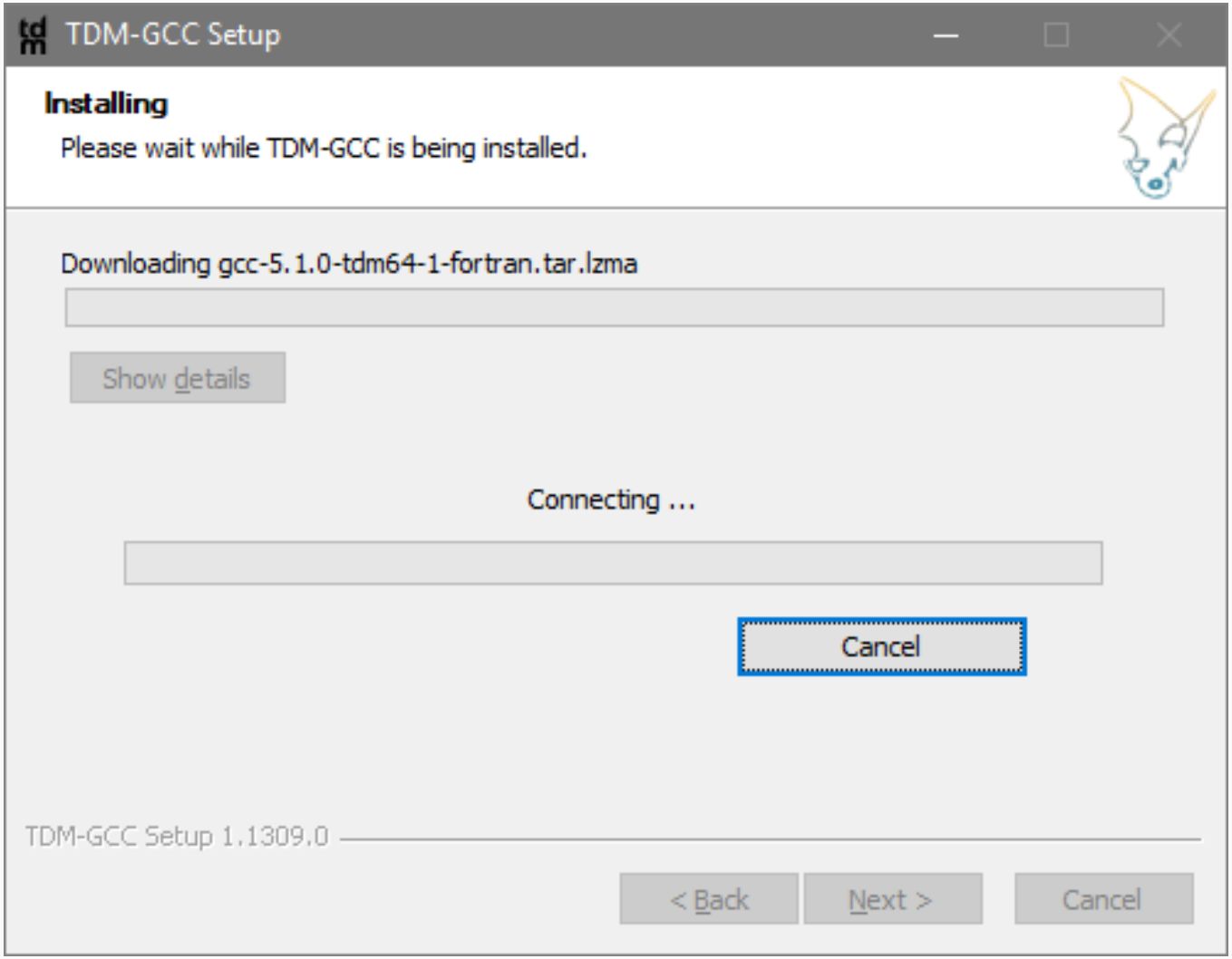
Instalación del compilador TDM-GCC



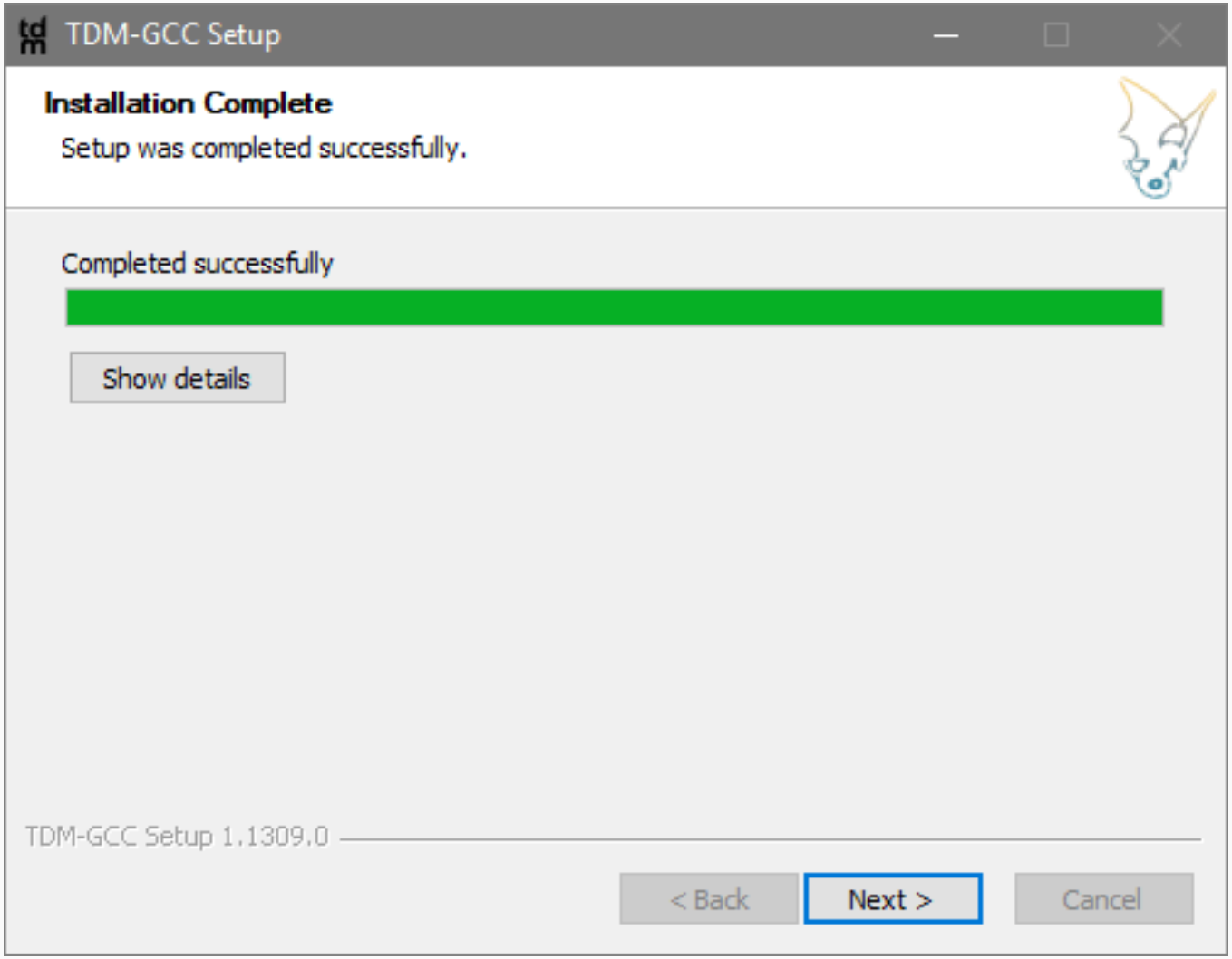
Instalación del compilador TDM-GCC



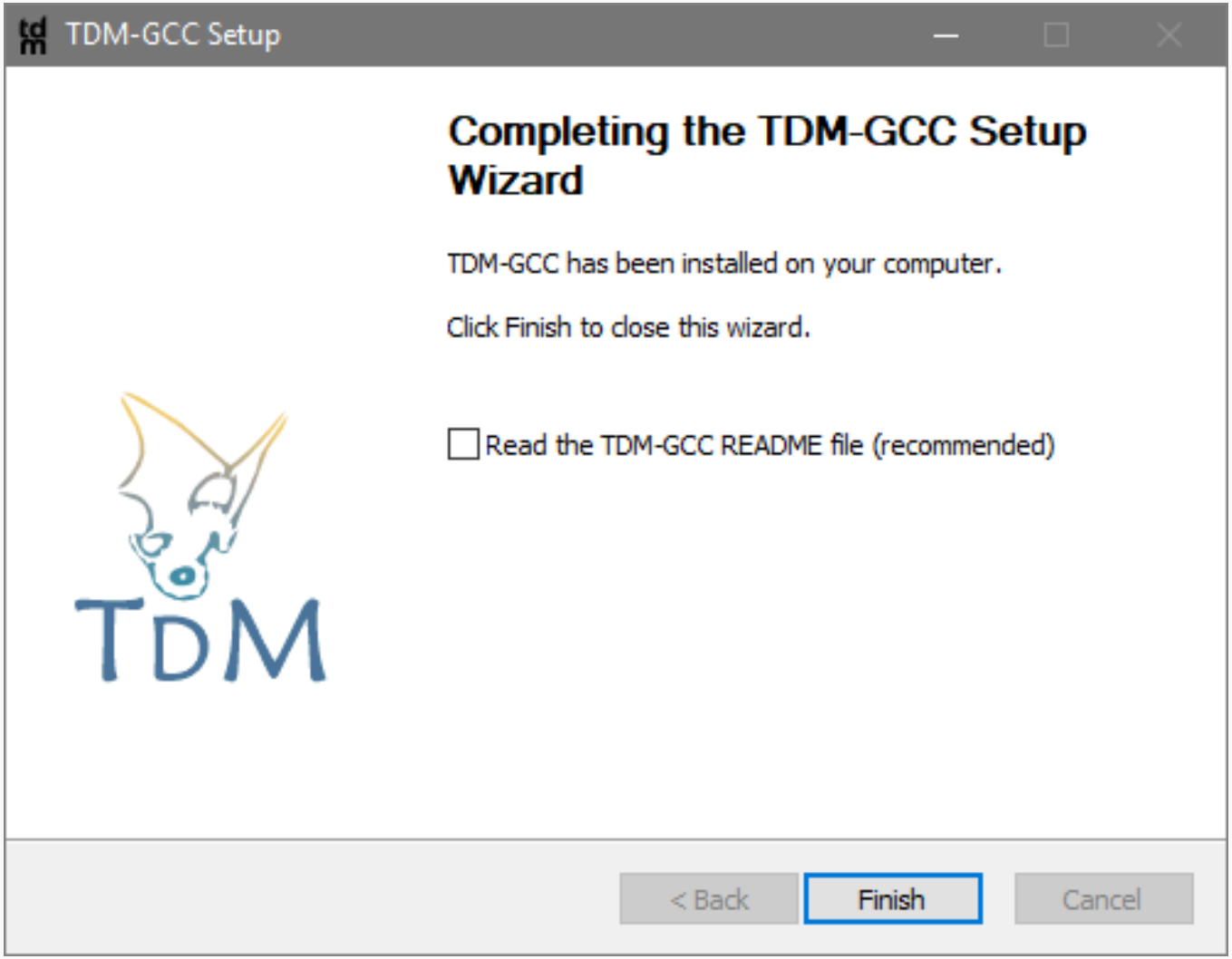
Instalación del compilador TDM-GCC



Instalación del compilador TDM-GCC



Instalación del compilador TDM-GCC



Conceptos básicos

Hello world! y el proceso de compilación

Hello world!

Hello world! y el proceso de compilación

Hello world!

- Se llama programa a un conjunto de instrucciones, realizadas computacionalmente en un tiempo determinado, aplicadas en la introducción, procesamiento o salida de datos.

Hello world! y el proceso de compilación

Hello world!

- Se llama programa a un conjunto de instrucciones, realizadas computacionalmente en un tiempo determinado, aplicadas en la introducción, procesamiento o salida de datos.
- Un programa en fortran tiene la siguiente forma:

Hello world! y el proceso de compilación

Hello world!

- Se llama programa a un conjunto de instrucciones, realizadas computacionalmente en un tiempo determinado, aplicadas en la introducción, procesamiento o salida de datos.
- Un programa en fortran tiene la siguiente forma:

```
1  PROGRAM hello_world
2     WRITE(*, *) Message
3  END PROGRAM hello_world
```

Scientific Hello world!

Scientific Hello world!

```
1   PROGRAM hello_world
2   IMPLICIT NONE
3
4   ! Angulo de entrada
5   REAL(KIND=4) :: theta
6
7   ! Resultado de aplicar la función seno
8   REAL(KIND=4) :: sin_of_theta
9
10  ! Mensaje
11  CHARACTER(len=*), PARAMETER :: Message = 'Hello World'
12
13  WRITE(*, *) 'Ingrese un ángulo [rad]: '
14  READ(*, *) theta
15  sin_of_theta = SIN(theta)
16
17  WRITE(*, *) Message
18  WRITE(*, *) "sin(", theta, ") = ", sin_of_theta
19  END PROGRAM hello_world
```

Proceso de compilación

Proceso de compilación

- El proceso de compilación se puede resumir en dos pasos

Proceso de compilación

- El proceso de compilación se puede resumir en dos pasos
  - Compilación

Proceso de compilación

- El proceso de compilación se puede resumir en dos pasos
- Compilación
- Enlazado

Proceso de compilación

- El proceso de compilación se puede resumir en dos pasos
  - Compilación
  - Enlazado
- El proceso de compilación en fortran posee la siguiente sintaxis:

```
fcomp [options] file1 [file2] [...] [fileN]
```

Proceso de compilación

- El proceso de compilación se puede resumir en dos pasos
  - Compilación
  - Enlazado
- El proceso de compilación en fortran posee la siguiente sintaxis:

```
fcomp [options] file1 [file2] [...] [fileN]
```

  - fcomp $\Rightarrow$ denota el comando para llamar al compilador. (gfortran, ifort, ...)

Proceso de compilación

- El proceso de compilación se puede resumir en dos pasos
  - Compilación
  - Enlazado
- El proceso de compilación en fortran posee la siguiente sintaxis:

```
fcomp [options] file1 [file2] [...] [fileN]
```

  - fcomp $\Rightarrow$ denota el comando para llamar al compilador. (gfortran, ifort, ...)
  - options $\Rightarrow$ opciones que permite el compilador. (-o, -f, -c, ...)

Proceso de compilación

- El proceso de compilación se puede resumir en dos pasos
  - Compilación
  - Enlazado
- El proceso de compilación en fortran posee la siguiente sintaxis:

```
fcomp [options] file1 [file2] [...] [fileN]
```

  - fcomp ⇒ denota el comando para llamar al compilador. (gfortran, ifort, ...)
  - options ⇒ opciones que permite el compilador. (-o, -f, -c, ...)
  - file ⇒ denota el archivo con su respectiva extensión (.f90, .f95, .o, ...)

Proceso de compilación

- El proceso de compilación se puede resumir en dos pasos
  - Compilación
  - Enlazado
- El proceso de compilación en fortran posee la siguiente sintaxis:

```
fcomp [options] file1 [file2] [...] [fileN]
```

  - fcomp ⇒ denota el comando para llamar al compilador. (gfortran, ifort, ...)
  - options ⇒ opciones que permite el compilador. (-o, -f, -c, ...)
  - file ⇒ denota el archivo con su respectiva extensión (.f90, .f95, .o, ...)
- Finalmente se obtiene un producto final.

Compilación

Compilador

Compilación

Compilador

- Programa escrito en un lenguaje de programación, que a su vez, traduce y genera otro programa en otro lenguaje de programación (código máquina), ambos equivalentes.

Compilación

Compilador

- Programa escrito en un lenguaje de programación, que a su vez, traduce y genera otro programa en otro lenguaje de programación (código máquina), ambos equivalentes.
- En principio, al usar un compilador, se busca traducir y simplificar un lenguaje de mayor complejidad a uno mucho más cotidiano y manejable en términos informáticos.

Compilación

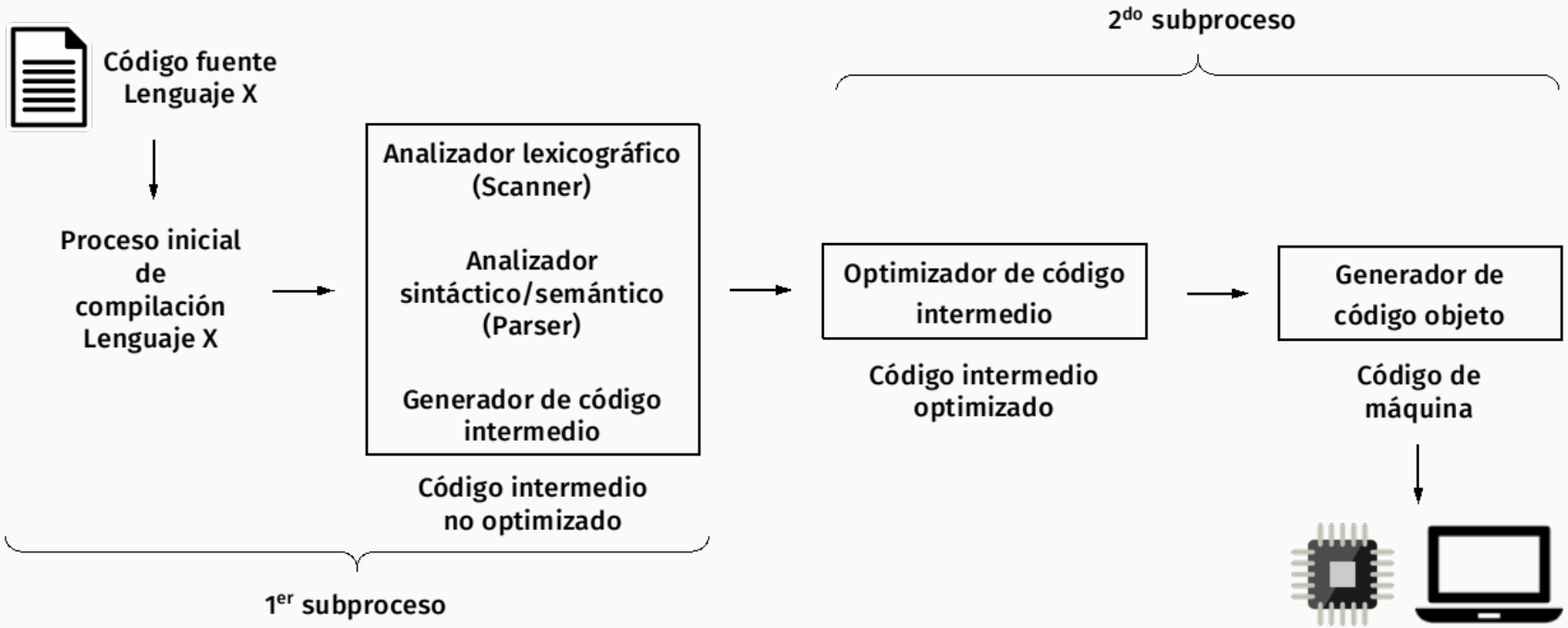


Figura 2: Diagrama de bloques del proceso de compilación

Enlazado

Enlazado?

Enlazado

Enlazado?

-

Productos finales

Ejecutables?

Productos finales

Ejecutables?

-

Formatos de escritura en Fortran

Formatos de escritura de un programa de Fortran

Formatos de escritura de un programa de Fortran

- Se crea el código fuente en un editor de textos y se guarda con su respectiva extensión (.f, .f90, .f95).

```
editor nombre.extensión
```

Formatos de escritura de un programa de Fortran

- Se crea el código fuente en un editor de textos y se guarda con su respectiva extensión (.f, .f90, .f95).

```
editor nombre.extensión
```

- Se emplean caracteres ASCI en el código fuente (alfanuméricos, símbolos y espacio)

Formatos de escritura de un programa de Fortran

- Se crea el código fuente en un editor de textos y se guarda con su respectiva extensión (.f, .f90, .f95).

```
editor nombre.extensión
```

- Se emplean caracteres ASCI en el código fuente (alfanuméricos, símbolos y espacio)
- No hay distinción entre mayúsculas y minúsculas.

Formatos de escritura de un programa de Fortran

- Se crea el código fuente en un editor de textos y se guarda con su respectiva extensión (.f, .f90, .f95).

```
editor nombre.extensión
```

- Se emplean caracteres ASCI en el código fuente (alfanuméricos, símbolos y espacio)
- No hay distinción entre mayúsculas y minúsculas.
- Los caracteres propios del español no pueden ser utilizadas en las instrucciones (á, é, ñ)

Formatos de escritura de un programa de Fortran

Formato libre

Formatos de escritura de un programa de Fortran

Formato libre

- Aplicable a versiones Fortran90 en adelante de extensión f90, f95...

Formatos de escritura de un programa de Fortran

Formato libre

- Aplicable a versiones Fortran90 en adelante de extensión f90, f95...
- Permite un máximo de 132 caracteres.

Formatos de escritura de un programa de Fortran

Formato libre

- Aplicable a versiones Fortran90 en adelante de extensión f90, f95...
- Permite un máximo de 132 caracteres.
- Se emplea el caracter (!) para empezar un comentario, sin que este se tome en cuentra durante la compilación.

Formatos de escritura de un programa de Fortran

Formato libre

- Aplicable a versiones Fortran90 en adelante de extensión f90, f95...
- Permite un máximo de 132 caracteres.
- Se emplea el caracter (!) para empezar un comentario, sin que este se tome en cuentra durante la compilación.

```
1   <instruccion> ! <comentario>
```

Formatos de escritura de un programa de Fortran

Formato libre

- Aplicable a versiones Fortran90 en adelante de extensión f90, f95...
- Permite un máximo de 132 caracteres.
- Se emplea el caracter (!) para empezar un comentario, sin que este se tome en cuentra durante la compilación.

```
1   <instruccion> ! <comentario>
```

- Se emplea el caracter (;) para separar dos instrucciones.

Formatos de escritura de un programa de Fortran

Formato libre

- Aplicable a versiones Fortran90 en adelante de extensión f90, f95...
- Permite un máximo de 132 caracteres.
- Se emplea el caracter (!) para empezar un comentario, sin que este se tome en cuentra durante la compilación.

```
<instruccion> ! <comentario>
```

- Se emplea el caracter (;) para separar dos instrucciones.

```
<instruccion>; <instruccion>
```

Formatos de escritura de un programa de Fortran

- En caso falte espacio en una línea, el caracter (&) permite continuar las instrucciones en la siguiente línea, colocándolo al final de la línea anterior y al inicio de la línea por comenzar.

Formatos de escritura de un programa de Fortran

- En caso falte espacio en una línea, el caracter (&) permite continuar las instrucciones en la siguiente línea, colocándolo al final de la línea anterior y al inicio de la línea por comenzar.

```
1     <instruccion 1> &
2   & <instruccion 1> &
3   & <instruccion 1>
4     <instruccion 2>
```

Formatos de escritura de un programa de Fortran

- En caso falte espacio en una línea, el caracter (&) permite continuar las instrucciones en la siguiente línea, colocándolo al final de la línea anterior y al inicio de la línea por comenzar.

```
1     <instruccion 1> &
2   & <instruccion 1> &
3   & <instruccion 1>
4     <instruccion 2>
```

- Las variables y unidades programaticas deben de ser nobradas empleando una combinacion de maximo 31 caracteres siendo el primer caracter una letra.

Formatos de escritura de un programa de Fortran

- En caso falte espacio en una línea, el caracter (&) permite continuar las instrucciones en la siguiente línea, colocándolo al final de la línea anterior y al inicio de la línea por comenzar.

```
1     <instruccion 1> &
2   & <instruccion 1> &
3   & <instruccion 1>
4     <instruccion 2>
```

- Las variables y unidades programaticas deben de ser nobradas empleando una combinacion de maximo 31 caracteres siendo el primer caracter una letra.

```
1   PROGRAM TPK95
```

Formatos de escritura de un programa de Fortran

Formato fijo

Formatos de escritura de un programa de Fortran

Formato fijo

- Aplicable a versiones anteriores a Fortran90 de extensión f, for...

Formatos de escritura de un programa de Fortran

Formato fijo

- Aplicable a versiones anteriores a Fortran90 de extensión f, for...
- Permite un máximo de 72 caracteres.

Formatos de escritura de un programa de Fortran

Formato fijo

- Aplicable a versiones anteriores a Fortran90 de extensión f, for...
- Permite un máximo de 72 caracteres.
- Se emplea el caracter (C) para comenzar un comentario.

Formatos de escritura de un programa de Fortran

Formato fijo

- Aplicable a versiones anteriores a Fortran90 de extensión f, for...
- Permite un máximo de 72 caracteres.
- Se emplea el caracter (C) para comenzar un comentario.

```
1   c --- Comentario
```

Formatos de escritura de un programa de Fortran

Formato fijo

- Aplicable a versiones anteriores a Fortran90 de extensión f, for...
- Permite un máximo de 72 caracteres.
- Se emplea el caracter (C) para comenzar un comentario.

```
1   c --- Comentario
```

- Se permite una sola instrucción por línea, comenzando a partir del séptimo caracter.

Formatos de escritura de un programa de Fortran

Formato fijo

- Aplicable a versiones anteriores a Fortran90 de extensión f, for...
- Permite un máximo de 72 caracteres.
- Se emplea el caracter (C) para comenzar un comentario.

```
1  c --- Comentario
```

- Se permite una sola instrucción por línea, comenzando a partir del séptimo caracter.

```
1  c23456789
2        <Instruccion>
```

Formatos de escritura de un programa de Fortran

- La continuación de una instrucción en una siguiente línea está dada por algún caracter no alfanumérico en el sexto caracter.

```
1  c23456789
2        <Instruccion muy larga>
3       &<Continuacion de instruccion muy larga>
```

Formatos de escritura de un programa de Fortran

- La continuación de una instrucción en una siguiente línea está dada por algún caracter no alfanumérico en el sexto caracter.

```
1  c23456789
2        <Instruccion muy larga>
3       &<Continuacion de instruccion muy larga>
```

- Los primeros cinco caracteres en una línea de instrucción se pueden emplear en caso se requiera una etiqueta (número positivo diferente de cero con un máximo de cinco dígitos).

```
1  c23456789
2  75289 <Instruccion muy larga>
3       &<Continuacion de instruccion muy larga>
```

Formatos de escritura de un programa de Fortran

- La continuación de una instrucción en una siguiente línea está dada por algún caracter no alfanumérico en el sexto caracter.

```
c23456789
      <Instruccion muy larga>
     &<Continuacion de instruccion muy larga>
```

- Los primeros cinco caracteres en una línea de instrucción se pueden emplear en caso se requiera una etiqueta (número positivo diferente de cero con un máximo de cinco dígitos).

```
c23456789
75289 <Instruccion muy larga>
     &<Continuacion de instruccion muy larga>
```

- Las variables deben de ser nobradas empleando una combinacion de maximo 6 caracteres siendo el primer caracter una letra.

```
c23456789
   12 REAL*8 FUNCTION Fib
```

Estructura de un programa en Fortran

Estructura de un programa en Fortran

Estructura de un programa en Fortran

- Un programa en Fortran está ordenado lógica y jerarquicamente en unidades programáticas.

Estructura de un programa en Fortran

- Un programa en Fortran está ordenado lógica y jerarquicamente en unidades programáticas.
- Tiene una unidad principal, llamada programa principal, la cual contiene las instrucciones que definen el objetivo del programa.

Estructura de un programa en Fortran

- Un programa en Fortran está ordenado lógica y jerarquicamente en unidades programáticas.
- Tiene una unidad principal, llamada programa principal, la cual contiene las instrucciones que definen el objetivo del programa.
- Es posible recurrir a subprogramas, que agrupados constituyen una instrucción del programa principal.

Estructura de un programa en Fortran

- Un programa en Fortran está ordenado lógica y jerarquicamente en unidades programáticas.
- Tiene una unidad principal, llamada programa principal, la cual contiene las instrucciones que definen el objetivo del programa.
- Es posible recurrir a subprogramas, que agrupados constituyen una instrucción del programa principal.

```
1  PROGRAM          ejemplo    !nombre del programa          !sentencia no ejecutable
2  IMPLICIT NONE               !declara todas las variables !sentencia no ejecutable
3  INTEGER          i          !declaración variables        !sentencia no ejecutable
4                   i = 3      !código del programa          !sentencia ejecutable
5  WRITE(*,*)       ' i =',i   !código del programa          !sentencia ejecutable
6  END PROGRAM ejemplo                                       !sentencia ejecutable
```

Estructura de un programa en Fortran

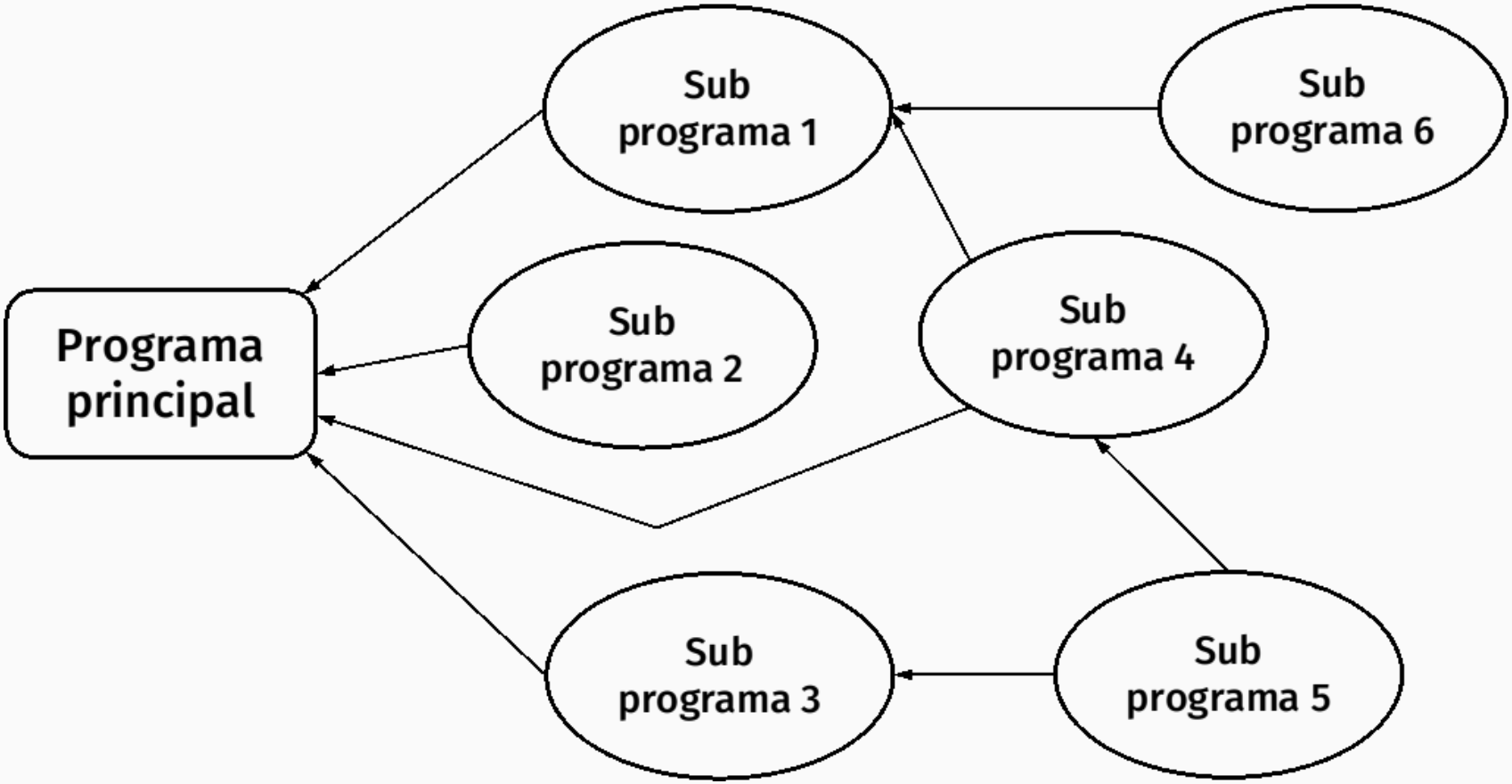


Figura 3: Esquema de la estructura de un programa en Fotran

Expresividad: el algoritmo TPK

El algoritmo TPK

El algoritmo TPK

- El algoritmo TPK es el "ejemplo inicial" para una secuencia de aprendizaje en programación. Fue empleado en el informe técnico The Early Development of Programming Languages por Donald Knuth y Luis Trabb del departamento de ciencia computacional de la Universidad de Stanford.

El algoritmo TPK

```
1   MODULE Functions
2   PUBLIC :: fun
3   CONTAINS
4       FUNCTION fun(t) result (r)
5           REAL, INTENT(IN) :: t
6           REAL  :: r
7           r = SQRT(ABS(t)) + 5.0*t**3
8       END FUNCTION fun
9   END MODULE Functions
10
11  PROGRAM TPK95
12  ! The TPK Algorithm
13  ! F style
14  USE Functions
15  INTEGER :: i
16  REAL    :: y
17  REAL, DIMENSION(0:10) :: a
18  READ *, a
19  DO i = 10, 0, -1 ! Bucle DO con contador hacia atrás
20      y = fun(a(i))
21      IF ( y < 400.0 ) THEN
22          PRINT *, i, y
23      ELSE
24          PRINT *, i, " Too large"
25      END IF
26  END DO
```