

Programación con Octave/Octave-UPM (I)

Mario Bermejo

CLUB DE INFORMÁTICA CAMINOS

18 diciembre 2014



Índice

- 1 Breve introducción a la programación
 - Conceptos
 - Tipos y estructuras de datos
 - Operadores
 - Estructuras de control o control de flujo
- 2 Instalación
 - Octave
 - Octave-UPM
- 3 Bibliografía

Definiciones

Lenguaje de programación

Cualquier lenguaje artificial que puede utilizarse para definir una **secuencia** de instrucciones para su procesamiento por un ordenador, o expresa las condiciones lógicas o funcionales que debe cumplir la solución del programa informático.

Programa o aplicación de software

Solución informática para resolver de forma automática un determinado problema. El proceso que se sigue en la resolución viene determinado por el algoritmo que implementa el programa.



Definiciones y conceptos

Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación.

Componentes

Componentes básicos de lenguaje de programación

- Léxico: vocabulario o conjunto de símbolos
- Sintaxis: indica cómo realizar las construcciones del lenguaje
- Semántica: determina el significado de una construcción correcta

Pasos

La programación se define también como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos, a través de los siguientes pasos:

- El desarrollo lógico del programa para resolver un problema en particular.
- Escritura de la lógica del programa empleando un lenguaje de programación específico (codificación del programa).
- Ensamblaje o compilación del programa hasta convertirlo en lenguaje de máquina.
- Prueba y depuración del programa.
- Desarrollo de la documentación

Clasificación de lenguajes de programación

Según el nivel de abstracción

- Bajo nivel
- Alto nivel

Según la implementación

- Compilados
- Interpretados

Según la finalidad del lenguaje

- Científicos
- Ingenieriles
- De gestión
- De inteligencia artificial
- Multipropósito

Compiladores e intérpretes

Compilador: traduce el programa fuente escrito en lenguaje de alto nivel a un programa objeto en lenguaje ensamblador o máquina.

Se utiliza el compilador sólo una vez para traducir el programa al lenguaje máquina, que ya es ejecutable.

Intérprete: va traduciendo y ejecutando las sentencias una por una. La ejecución del programa fuente sólo es posible cuando el intérprete lo va traduciendo.

Depurador: permite ejecutar un programa compilado paso a paso, con el fin de ver lo que falla e ir eliminando errores.

Emulador: es un programa capaz de interpretar el lenguaje máquina contenido en un fichero ejecutable que utiliza el lenguaje de otro procesador.



Dato

Es cualquier objeto abstracto manipulable por un computador

Un dato es almacenado en una **constante** cuando no es necesario alterar su valor durante la ejecución del programa, mientras que se almacena en una **variable** del tipo adecuado cuando son susceptibles de modificación

Estructura de datos

Conjunto de variables de un determinado tipo, agrupadas y organizadas de alguna manera para representar un comportamiento

Las estructuras de datos pueden ser estáticas o dinámicas:

- Estáticas: El tamaño en memoria es fijo. No se pueden ampliar
- Dinámicas: El tamaño es variable. Se pueden ampliar o reducir.

Tipos de datos: primitivos (I)

Tipos de datos primitivos

Los tipos de datos hacen referencia al tipo de información que se trabaja, donde la unidad mínima de almacenamiento es el dato, también se puede considerar como el rango de valores que puede tomar una variable durante la ejecución del programa.

- Caracter
- Numérico
- Lógicos (Booleanos)

Tipos de datos: primitivos (II)

Caracteres

El tipo de dato carácter es un dígito individual el cual se puede representar como numéricos (0 al 9), letras (a-z) y símbolo (\$, _).

Tipo de dato	Rango	Tamaño de bits
char	0 a 65536	16 bits

Tipos de datos: primitivos (III)

Numéricos

- **Entero**: son los valores que no tienen punto decimal, pueden ser positivos o negativos y el cero.
- **Reales**: estos caracteres almacenan números muy grandes que poseen parte entera y parte decimal.

ENTEROS

tipo de dato: byte tamaño = 8 bits

tipo de dato: short tamaño = 16 bits

tipo de dato: int tamaño = 32 bits

tipo de dato: long tamaño = 64 bits

REALES

tipo de dato: float tamaño = 32 bits

tipo de dato: double tamaño = 64 bits

Tipos de datos: primitivos (IV)

Booleanos

Este tipo de dato se emplea para valores lógicos, los podemos definir como datos comparativos dicha comparación devuelve resultados lógicos.

tipo de dato: `boolean` Rango= `true/false`

Tipos de datos: compuestos

Tipos de datos compuestos

Derivados de más de un elemento de tipo primitivo. Se puede hacer de varias maneras, resultando estructuras de datos.

- **Array o lista:** un array es una colección ordenada de valores. Una lista puede contener cualquier cosa. Dependiendo del lenguaje se permiten tipos diferentes dentro del array o no. Ejm: vectores y matrices.
- **Tupla:** una tupla es una secuencia de valores agrupados. Una tupla sirve para agrupar, como si fueran un único valor, varios valores que, por su naturaleza, deben ir juntos. Ejm: coordenadas.
- **Conjuntos:** un conjunto es una colección desordenada sin elementos duplicados. Usados para comprobación de pertenencia, unión, intersección o diferencia.



Tipos de datos

Cadena

- Es una secuencia de caracteres. Se usa habitualmente para representar palabras y texto. Cualquier caracter ascii puede formar parte de la cadena (excepto caracteres especiales dependiendo del lenguaje, que hay que escribir con sintaxis especial)

Otros tipos

- Punteros
- Funciones



Tipos de datos

Tipos de datos abstractos

- Stack o pila
- Tree o árbol
- Graph o grafo

Tipos de datos objetos

- Objetos
- Clases

Utility data

- Tiempo
- Fecha
- Moneda
- ...

Operadores Relacionales

- $<$ menor que
- $>$ mayor que
- $<=$ menor o igual que
- $>=$ mayor o igual que
- $==$ igual que
- \sim distinto que

Operadores Lógicos

- & and
- | or
- ~ negación lógica

Operador de asignación

Muy importante

El operador de asignación: =

El operador asignación asigna el valor o tipo de dato de la derecha a la variable de la izquierda

Operadores matemáticos

- + la suma
- – la resta
- * la multiplicación
- / la división
- % el módulo (resto de la división)
- ** la potencia («elevado a»)

Estructuras de control

Concepto

Las estructuras de control permiten modificar el flujo de ejecución de las instrucciones de un programa.

Con las estructuras de control se puede:

- de acuerdo a una condición, ejecutar un grupo u otro de sentencias (If-Then-Else y Select-Case)
- Ejecutar un grupo de sentencias mientras exista una condición (Do-While)
- Ejecutar un grupo de sentencias hasta que exista una condición (Do-Until)
- Ejecutar un grupo de sentencias un número determinado de veces (For-Next)
- ...

Estructuras de control de selección (I)

Selección if simple

Se trata de una estructura de control que permite redirigir un curso de acción según la evaluación de una condición simple, sea falsa o verdadera.

Si la condición es verdadera, se ejecuta el bloque de sentencias 1, de lo contrario, se ejecuta el bloque de sentencias 2.

```
IF (Condición) THEN
    (Bloque de sentencias 1)
ELSE
    (Bloque de sentencias 2)
END IF
```



Estructuras de control de selección (II)

Selección if múltiple

Se pueden plantear múltiples concisiones simultáneamente, si se cumple la (Condición 1), se ejecuta (Bloque de sentencias 1) en caso contrario se comprueba la (Condición 2), si es cierta se ejecuta (Bloque de sentencias 2), y así sucesivamente hasta n condiciones, si ninguna de ellas es cumple se ejecuta (Bloque de sentencias else).

```
IF (Condición 1) THEN
    (Bloque de sentencias 1)
ELSEIF (Condición 2) THEN
    (Bloque de sentencias 2)
.....
ELSEIF (Condición n) THEN
    (Bloque de sentencias n)
ELSE
    (Bloque de sentencias ELSE)
END IF
```

Estructuras de control de selección (II)

Sentencia Select-Case

Esta sentencia permite ejecutar una de entre varias acciones en función del valor de una expresión. Es una alternativa a if then else cuando se compara la misma expresión con diferentes valores.

Se evalúa la expresión, dando como resultado un número. Luego, se recorren los «Case» dentro de la estructura buscando que el número coincida con uno de los valores. Es necesario que coincidan todos sus valores. Cuando se encuentra la primera coincidencia, se ejecuta el bloque de sentencias correspondiente y se sale de la estructura Select-Case. Si no se encuentra ninguna coincidencia con ningún valor, se ejecuta el bloque de sentencias de la sección «Case Else».



Estructuras de control de selección (II)

Sentencia Select-Case

```
SELECT (Expresión)
  CASE Valor1
    (Bloque de sentencias 1)
  CASE Valor2
    (Bloque de sentencias 2)
  CASE Valor n
    (Bloque de sentencias n)
  CASE ELSE
    (Bloque de sentencias "Else")
END SELECT
```

Estructuras de control iterativas (I)

Las estructuras de control iterativas o de repetición, inician o repiten un bloque de instrucciones si se cumple una condición o mientras se cumple una condición.
También se denominan bucles

Sintaxis Do-While

Mientras la condición sea verdadera, se ejecutarán las sentencias del bloque.

```
DO WHILE (Condición)
    (Bloque de sentencias)
LOOP
```

que también puede expresarse:

```
WHILE (Condición)
    (Bloque de sentencias)
WEND
```



Estructuras de control iterativas (II)

Sentencia Do-Until

Se ejecuta el bloque de sentencias, hasta que la condición sea verdadera

```
DO
  (Bloque de sentencias)
LOOP UNTIL (Condición)
```

Estructuras de control iterativas (III)

Sentencia For-Next

La sentencia For da lugar a un lazo o bucle, y permite ejecutar un conjunto de sentencias cierto número de veces.

Primero, se evalúan las expresiones 1 y 2, dando como resultado dos números. La variable del bucle recorrerá los valores desde el número dado por la expresión 1 hasta el número dado por la expresión 2. El bloque de sentencias se ejecutará en cada uno de los valores que tome la variable del bucle.

```
FOR (Variable) = (Expresión1) TO (Expresión2) STEP (Salto)
    (Bloque de sentencias)
NEXT
```



Estructuras anidadas

Las estructuras de control básicas pueden anidarse, es decir pueden ponerse una dentro de otra.

Estructura For-Next dentro de una estructura If-Then-Else

```
IF A > B THEN
  FOR X = 1 TO 5
    (Bloque de sentencias 1)
  NEXT
ELSE
  (Bloque de instrucciones 2)
END IF
```

Estructuras anidadas

Estructura If-Then-Else dentro de estructura For-Next

```
FOR x = 10 TO 20 STEP 2
  IF A == C THEN
    (Bloque de instrucciones)
  ELSE
    (Bloque de instrucciones)
  END IF
NEXT
```

Estructuras anidadas

Estructura For-Next que está dentro de estructura Do-While

```
DO WHILE A > 0
  FOR X = 1 TO 10
    (Bloque de instrucciones)
  NEXT
  A = A - 1
LOOP
```

Octave











Octave: sourceforge - Octave3.4.3_gcc4.5.2-install.exe

<http://sourceforge.net/projects/octave/files/Octave Windows binaries/Octave 3.4.3 for Windows MinGW Installer/>

Looking for the latest version? [Download econometrics-1.1.1.tar.gz \(26.9 kB\)](#)

[Home](#) / [Octave Windows binaries](#) / Octave 3.4.3 for Windows MinGW Installer



Name	Modified	Size	Downloads
↑ Parent folder			
README	2011-12-30	6.0 kB	3  
Octave3.4.3_gcc4.5.2-install.exe	2011-12-24	147.1 MB	324  
Octave3.4.3_gcc4.5.2_pkgs_20111...	2011-12-08	44.8 MB	11  
Octave3.4.3_gcc4.5.2_20111025.7z	2011-12-08	140.3 MB	7  
libreadline.dll.a.7z	2011-12-08	17.2 kB	7  
Totals: 5 Items		332.2 MB	352

There are 2 options for installing the mingw32 build of octave for MS Windows:

- The installer: Octave3.4.3_gcc4.5.2-install.exe
 - You may change the default install directory. Do not choose one with spaces, it will cause problems for Octave's pkg function.
 - It does not include the octaveforge packages. Use the pkgs 7z archive if these are required.
 - It includes the main Octave archive and the libreadline archive



Octave-UPM: información

Información:

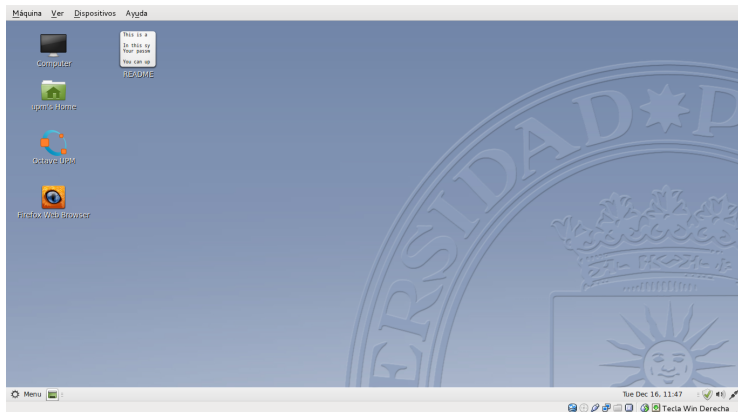
- Web de Israel Herraiz: www.herraiz.org
- <http://mat.camino.upm.es/octave/>
- Video Instalación de Octave UPM:
www.youtube.com/watch?v=Y8u6uHXWBrw
- Curso de Introducción a la Programación: Videos de Israel Herraiz



Octave-UPM: Información

Instalación en máquina virtual bajo VirtualBox:

- https://mat.caminos.upm.es/wiki/Tutorial_de_instalación_de-Octave_UPM_con_máquina_virtual



Octave-UPM: Instalación paso a paso

Descargando paquetes necesarios:



Octave UPM

Octave UPM es una versión personalizada de GNU Octave para las clases de Informática en la ETSI de la Universidad Politécnica de Madrid.

[GNU Octave](#) es un entorno de matemática computacional que intenta mantener un alto nivel de caso de Octave UPM, la configuración por defecto elegida para el programa mejora ligeramente la comparado con usar Octave directamente.

Aunque esta versión personalizada está pensada para las clases en nuestra Escuela, es probable que personas que estén buscando una alternativa libre y gratuita a MATLAB, que esté en español o inglés, GNU/Linux, con una instalación limpia y sencilla, es probable que Octave UPM sea lo que estás buscando.

GNU Octave y Octave UPM se distribuyen como software libre bajo la [Licencia Pública General de GNU](#). Lo puedes instalar en tu ordenador sin restricciones, y que puedes modificarlo y redistribuirlo siempre que cumplas con sus términos y condiciones.

Instalación

Windows

Se distribuyen dos paquetes: GNU Octave edición UPM y un paquete de bibliotecas adicionales.

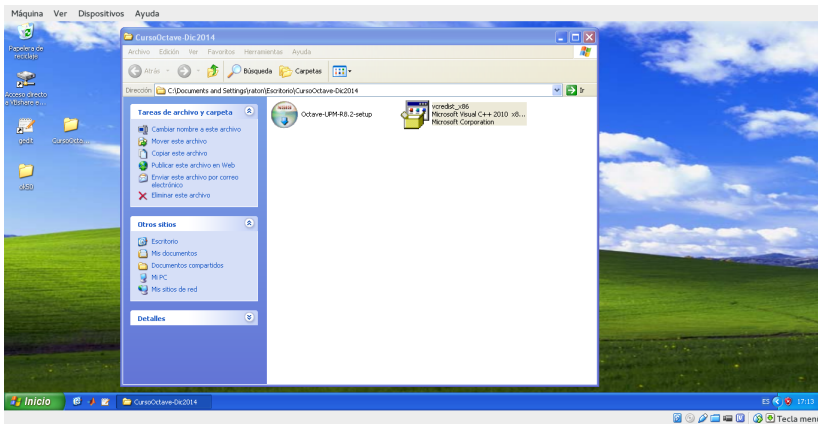
- GNU Octave 3.6.2 edición UPM (Release 8.2) (EXE 42 MB)
- Bibliotecas adicionales (EXE 4.9 MB)

En algunas versiones de Windows, el paquete de bibliotecas adicionales ya vienen incorporado de serie.



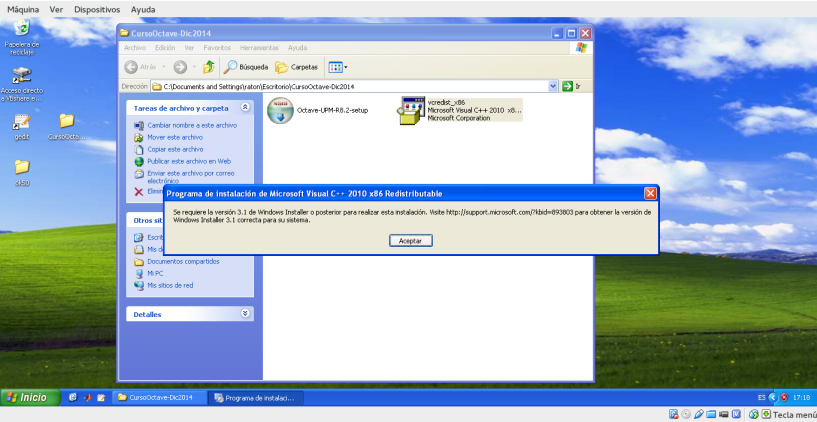
Octave-UPM: Instalación paso a paso

Instalar vcredist_x86



Octave-UPM: Instalación paso a paso

Para versiones de XP puede ser necesario Windows Instaler 3.1.



Octave-UPM: Instalación paso a paso

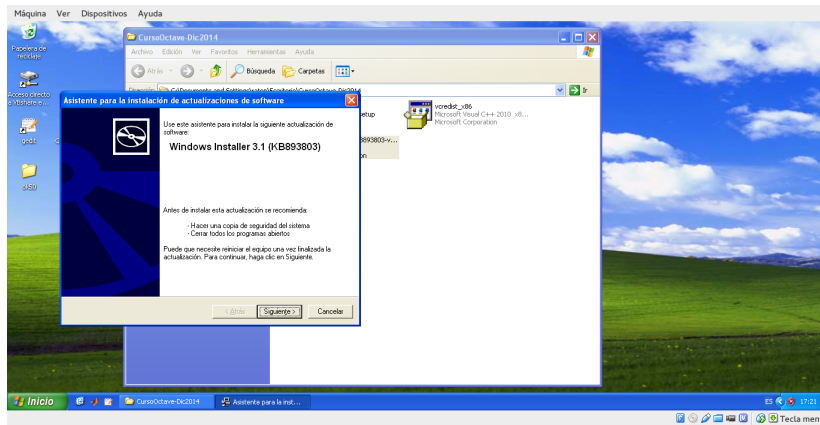
Descargando Windows Instaler 3.1.



The screenshot shows a web browser window with the URL www.microsoft.com/es-es/download/confirmation.aspx?id=25. Below the URL bar is a blue banner with the text: "Al usar este sitio acepta el uso de cookies para análisis, contenido personalizado y publicidad". The main content area features the Microsoft logo and the heading "Download Center". A navigation menu includes links for "Comprar", "Productos", "Categorías", "Soporte técnico", and "Seguridad". The main message reads: "Gracias por descargar el producto". Below this, the product name "Windows Installer 3.1 Redistributable (v2) - Español" is displayed in red. A note states: "Si la descarga no comienza al cabo de 30 segundos, [Haga clic aquí](#)". At the bottom, there is a link with a plus icon: "Instrucciones de instalación".

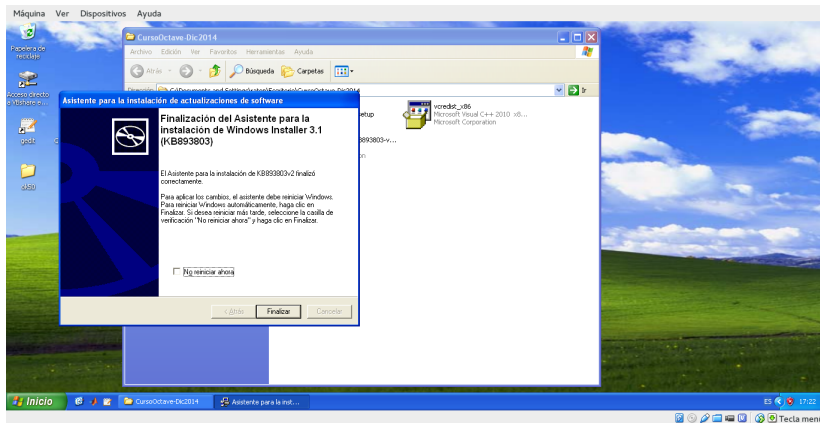
Octave-UPM: Instalación paso a paso

Si es necesario instalamos Windows Instaler 3.1



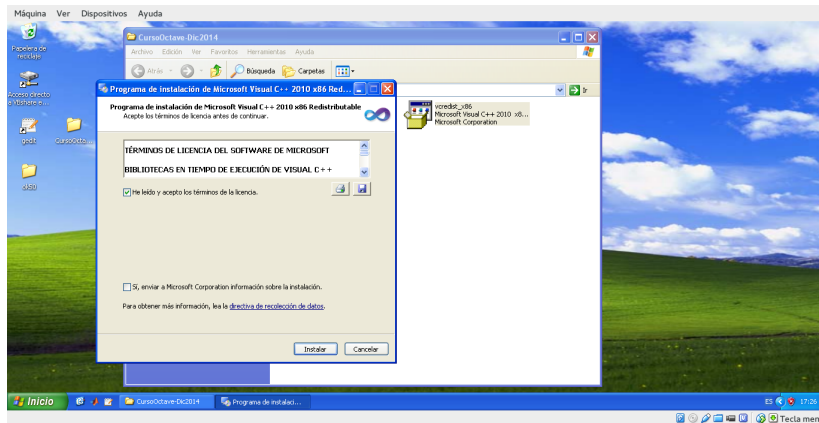
Octave-UPM: Instalación paso a paso

Obtenemos esta pantalla de correcta instalación de Windows Installer 3.1.



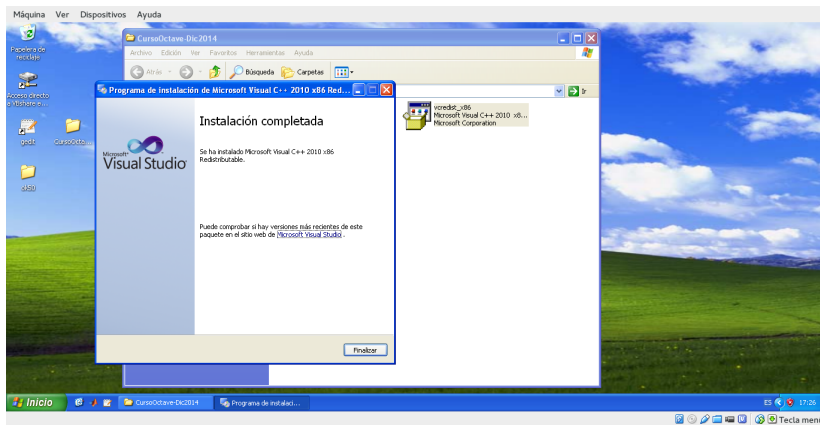
Octave-UPM: Instalación paso a paso

Volvemos al paso de instalar `vc_redist_x86`



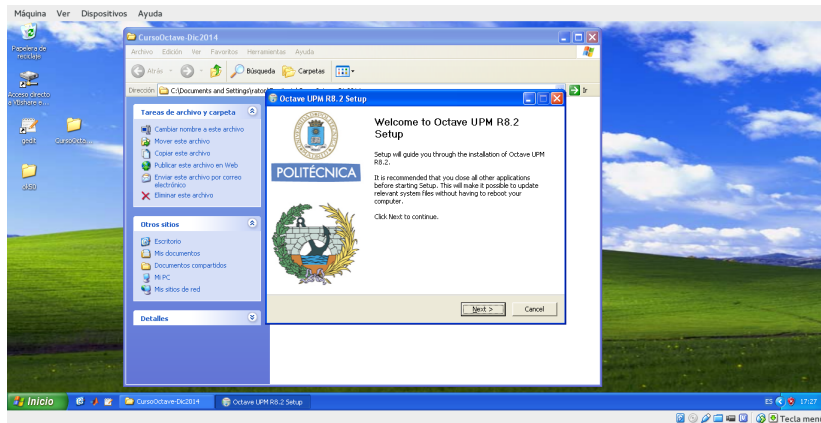
Octave-UPM: Instalación paso a paso

Al finalizar la instalación de `vc_redist_x86` obtenemos esta pantalla.



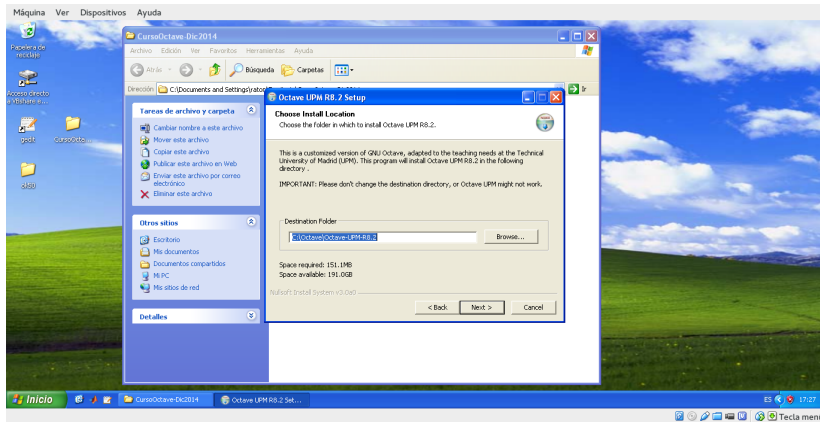
Octave-UPM: Instalación paso a paso

Ahora instalamos Octave UPM



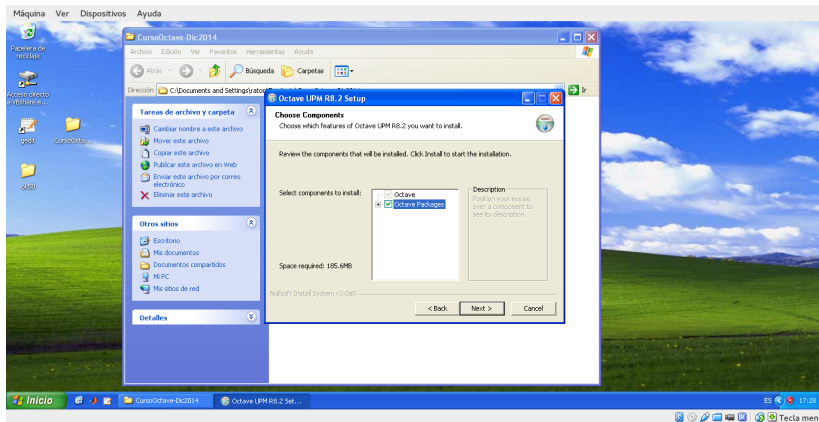
Octave-UPM: Instalación paso a paso

Elegimos una ruta de instalación o dejamos la que aparece por defecto.



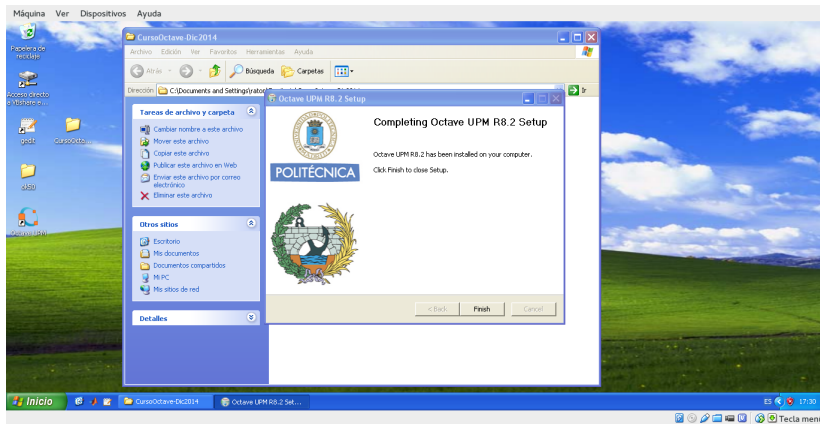
Octave-UPM: Instalación paso a paso

Elegimos instalar también los *Octave Packages*.



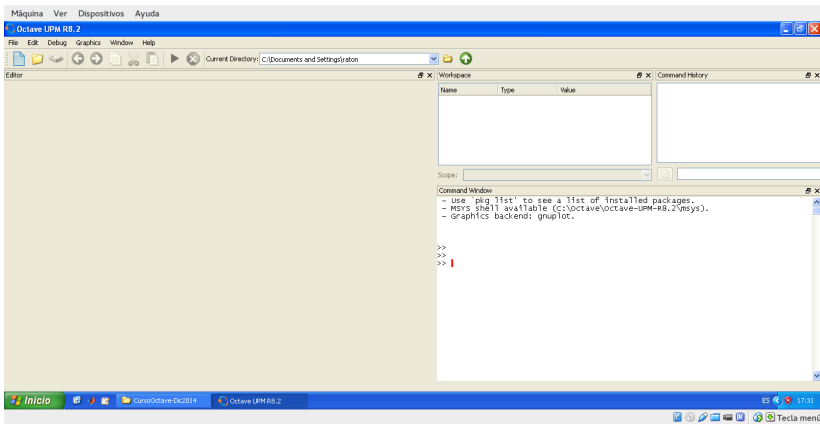
Octave-UPM: Instalación paso a paso

Debemos obtener este mensaje de finalización de la instalación. Aparecerá un nuevo icono que es el de Octave UPM.



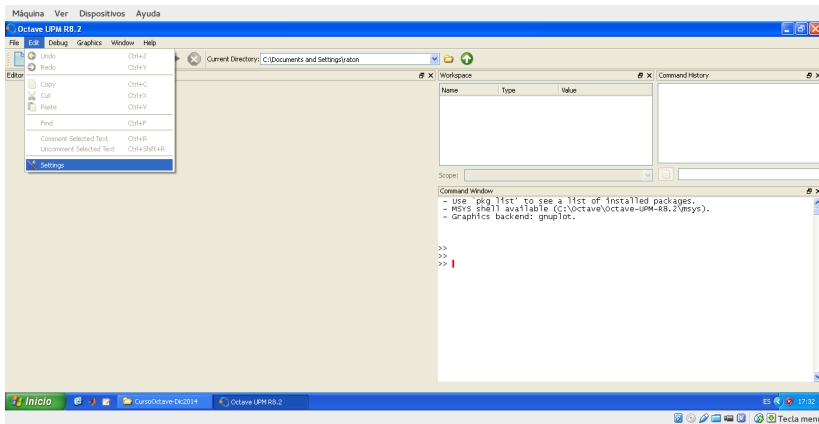
Octave-UPM: Instalación paso a paso

Iniciamos el programa:



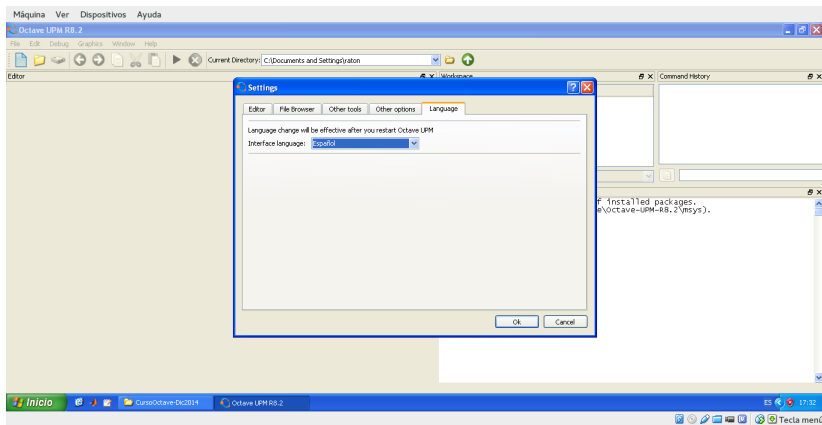
Octave-UPM: Instalación paso a paso

Podemos poner la interfaz en español.



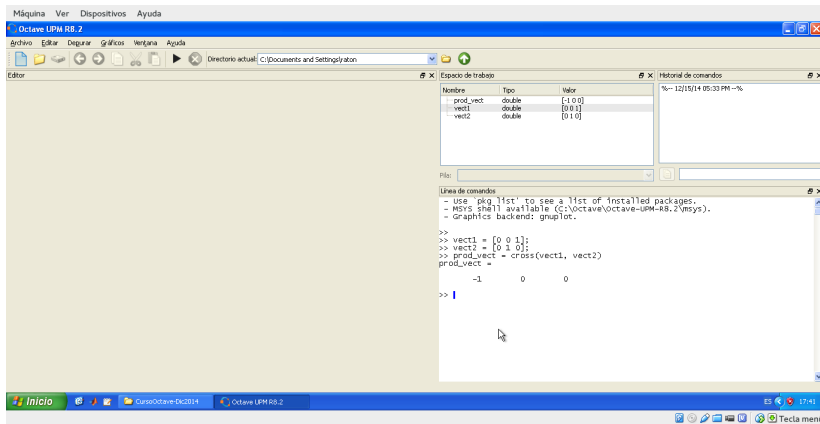
Octave-UPM: Instalación paso a paso

Elegimos la interfaz en español. Es posible que sea necesario reiniciar el programa.







Octave-UPM: Instalación paso a paso

Ya podemos usarlo.



Fuentes y bibliografía

-  *GNU Octave*. (Manual en pdf)
- *Manual de iniciación a GNU Octave*. J.M Valiente Cifuentes.
- *Informática Básica*. F.J. Martín Martínez. Ed. Ra-Ma. 2004
- *Matlab y sus aplicaciones en las ciencias y la ingeniería*. César Pérez. Ed Pearson Prentice Hall. 2003
- *Cálculo científico con Matlab y Octave*. A.Quarteroni, F.Saleri. Ed Springer. 2006.
- *Mástering Matlab*. Duane Hanselman, Bruce Littlefield. Ed Pearson Prentice Hall. 2005
- Wikipedia:  1,  2,  3
- <http://progra.usm.cl/apunte/materia/index.html>
- http://en.wikipedia.org/wiki/Data_type