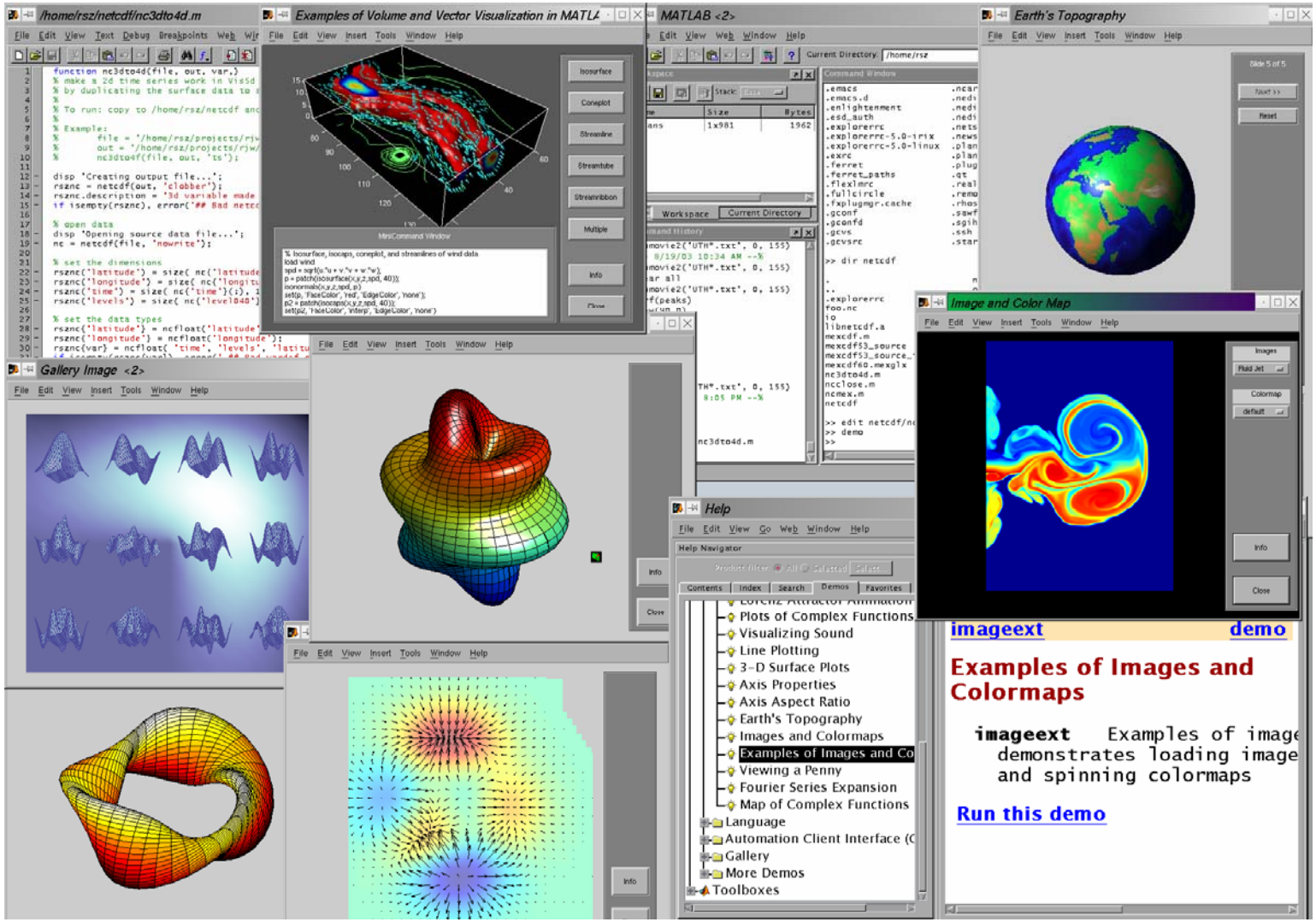
CEU
Universidad
San Pablo

CURSO de UTILIZACIÓN PRÁCTICA de MATLAB

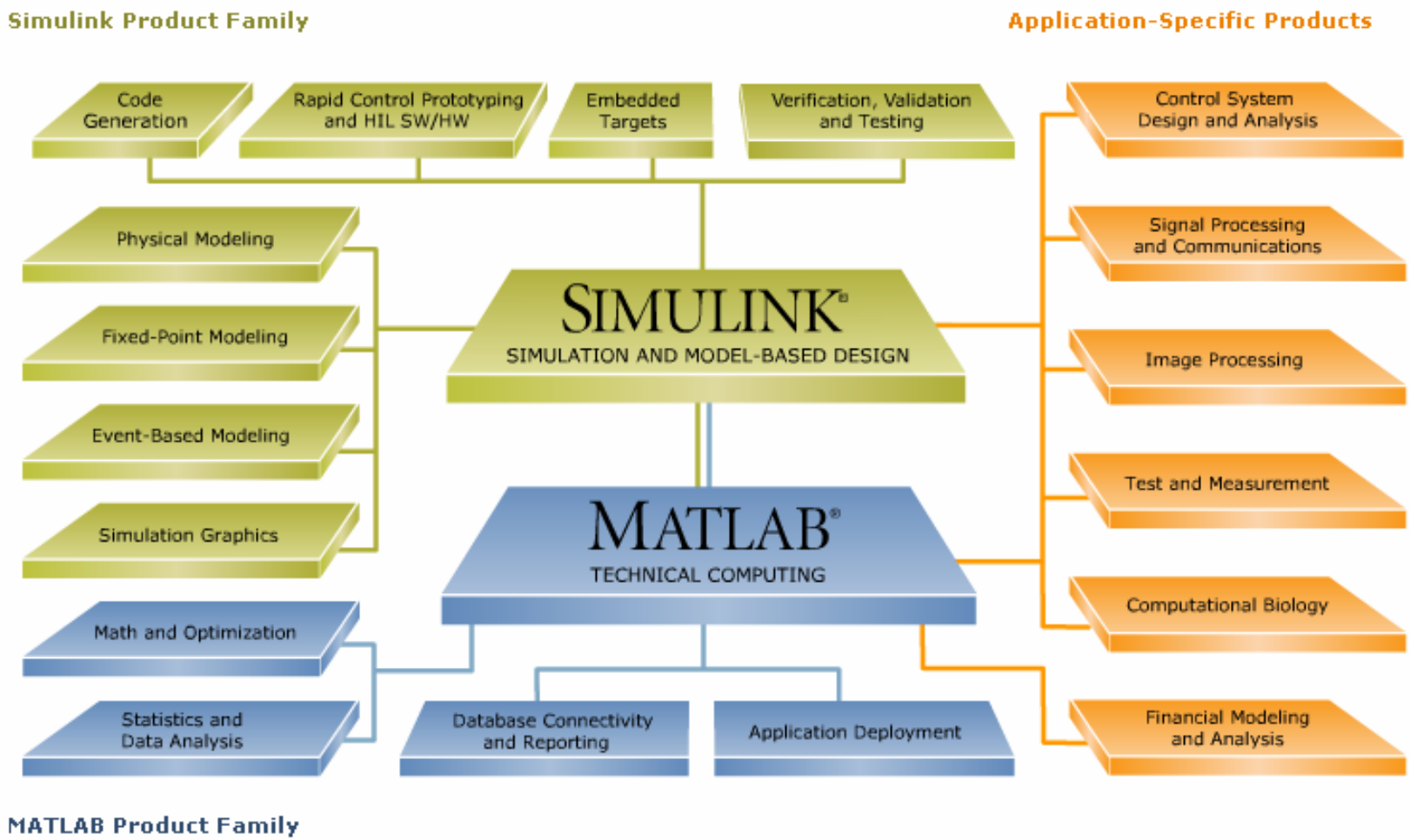
Sesión 0

Carlos Óscar Sánchez Sorzano, Ph.D.
Madrid, July 17th 2006

MATLAB como paquete de programas



MATLAB como paquete de programas



MATLAB como paquete de programas

MATLAB

Editor - G:\MATLAB\toolbox\matlab\datafun\hist.m

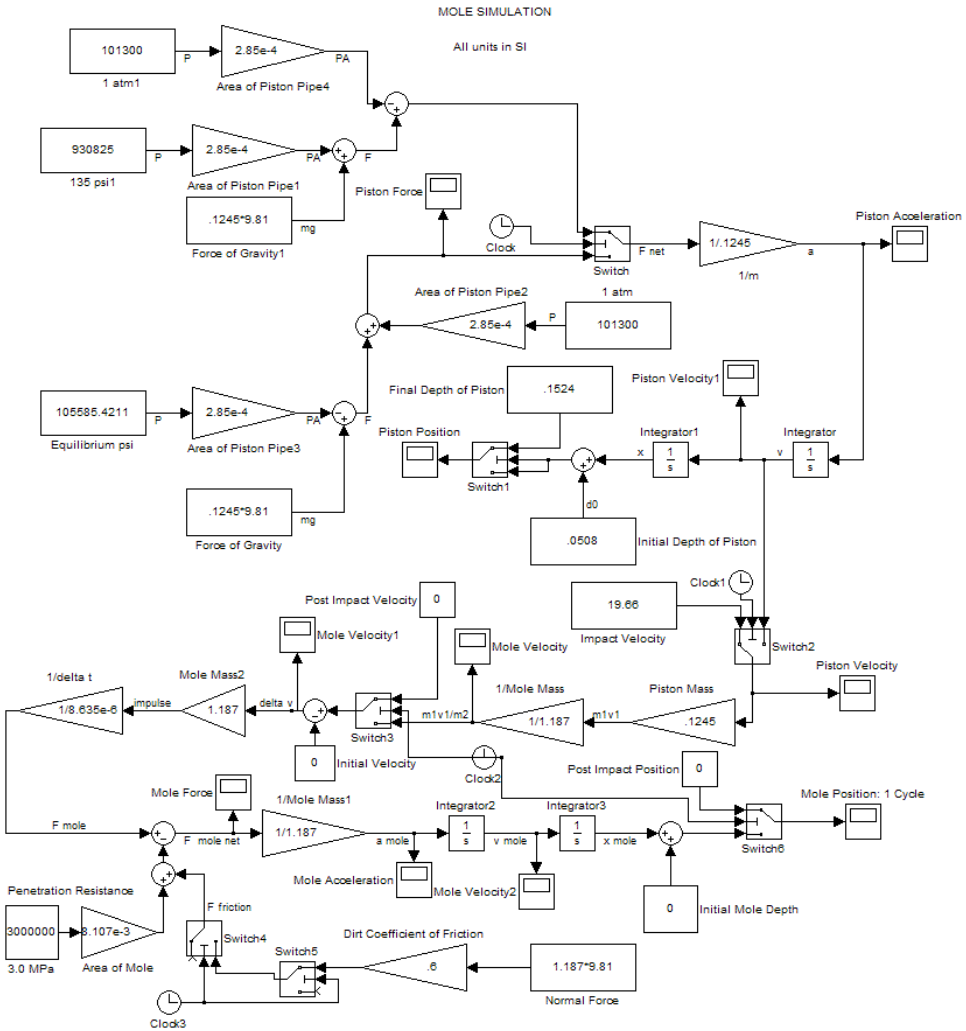
File Edit Text Go Cell Tools Debug Desktop Window Help

```
36
37 -  if nargs == 1
38 -      x = 10;
39 -  else
40 -      x = args{2};
41 -  end
42
43 -  if min(size(y))==1, y = y(:); end
44 -  if ~ishistnumeric(x) || ~ishistnumeric(y)
45 -      error('MATLAB:hist:InvalidInput', 'Input arguments must be numeric.')
46 -  end
47
48 -  if isempty(y),
49 -      if length(x) == 1,
50 -         x = 1:double(x);
51 -      end
52 -      nn = zeros(size(x)); % No elements to count
53        %  Set miny, maxy for call to bar below.
54 -      miny = [];
55 -      maxy = [];
56 -  else
57        %  Ignore NaN when computing miny and maxy.
58 -      ind = ~isnan(y);
59 -      miny = min(y(ind));
60 -      maxy = max(y(ind));
61        %  miny, maxy are empty only if all entries in y are NaNs.  In this case,
62        %  max and min would return NaN, thus we set miny and maxy accordingly.
63 -      if (isempty(miny))
64 -        miny = NaN;
65 -        maxy = NaN;
66 -      end
67 -      if length(x) == 1
68 -            if miny == maxy,
69 -                miny = miny - floor(x/2) - 0.5;
```

hist Ln 1 Col 1 OVR

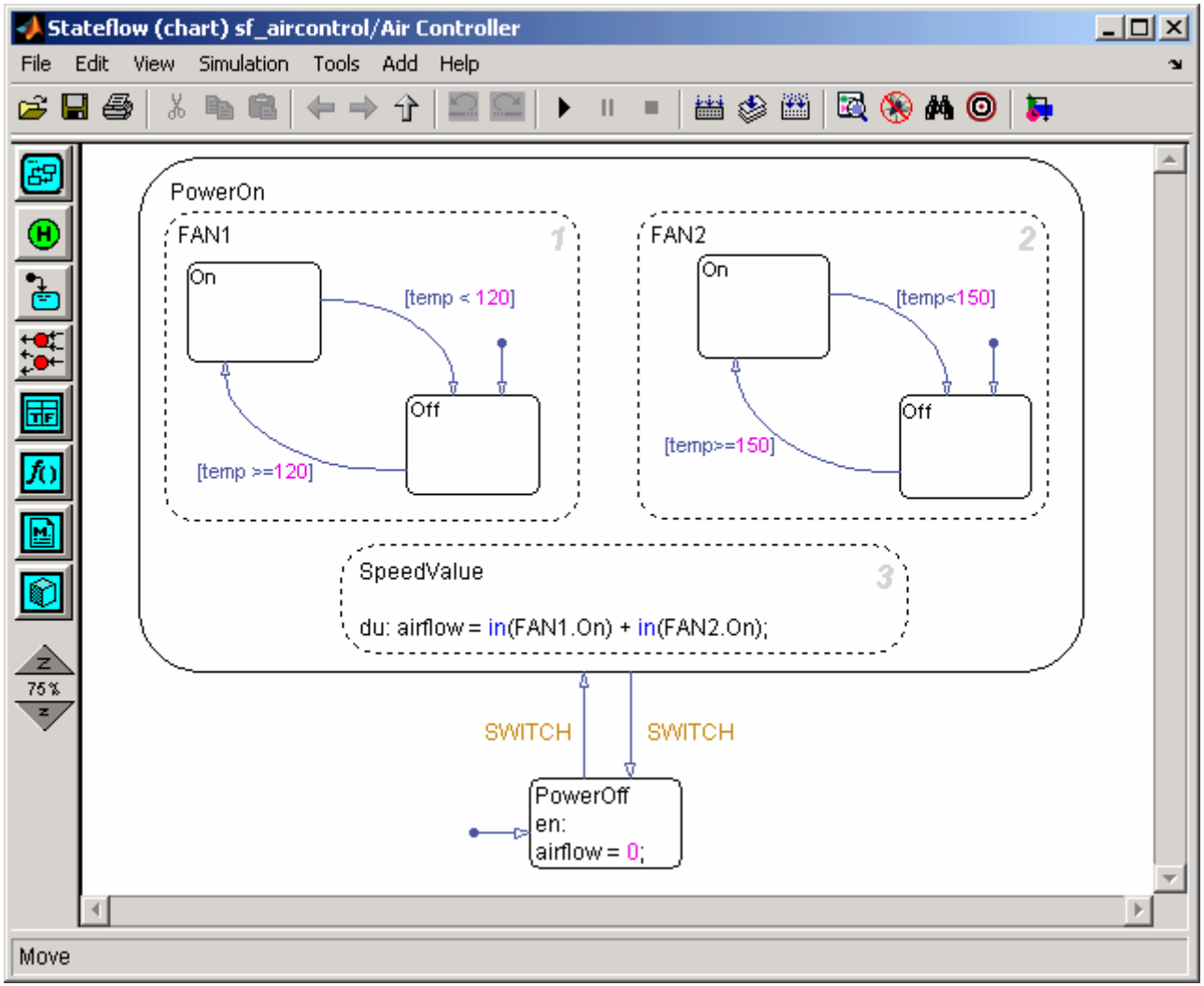
MATLAB como paquete de programas

Simulink



MATLAB como paquete de programas

Stateflow



Un poco de historia

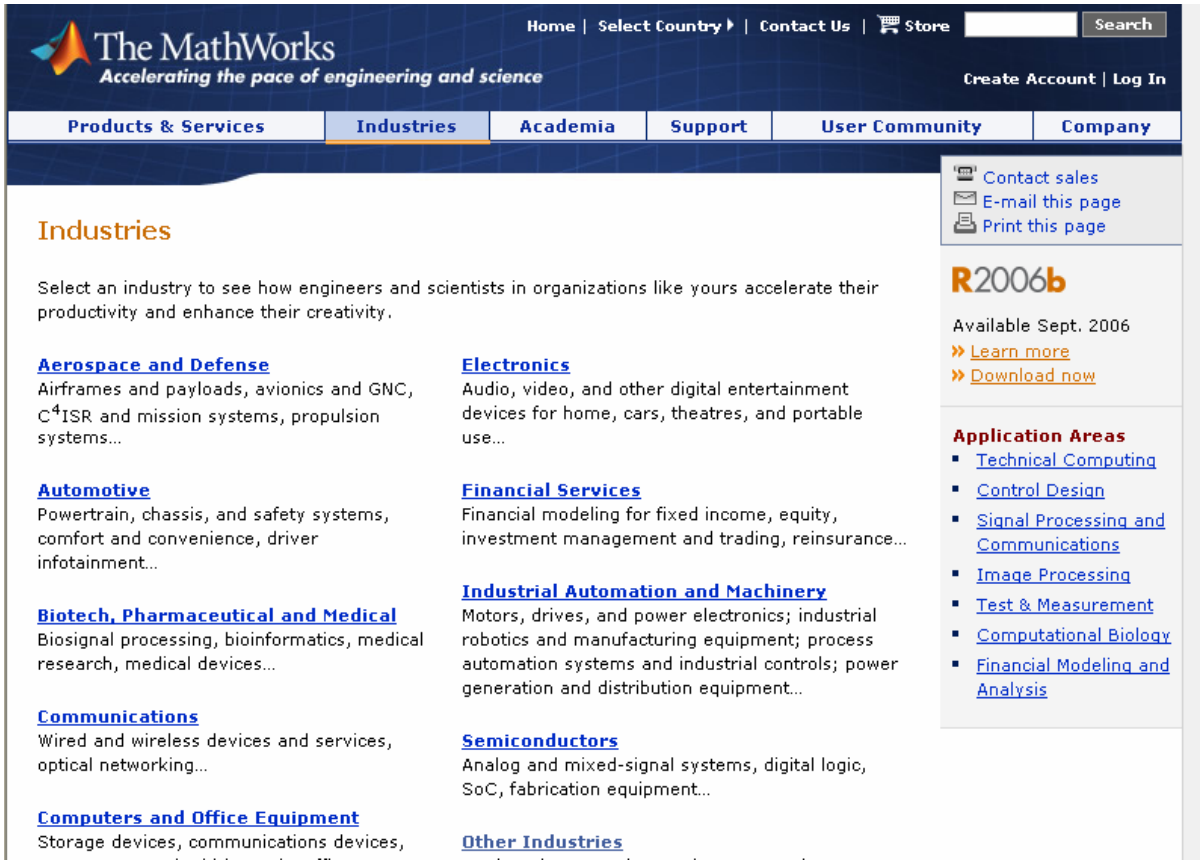
- MATLAB se inventó en los años 70 para poder acceder a las funciones de álgebra lineal de LINPACK y EISPACK sin tener que aprender Fortran.
- Su autor fue Cleve Moler (Univ. New Mexico).
- En 1983, Jack Little (Univ. Stanford) se unió al proyecto y comenzó a comercializarse.

Ventajas y desventajas de MATLAB

- Ventajas
  - Amplio soporte matemático
  - Alta precisión
  - Amplio soporte de funciones ya desarrolladas
  - Rápido prototipado
  - Integración con dispositivos hardware
  - Una comunidad muy extendida
  - Magnífica ayuda
  - Comercial
- Desventajas
  - Gestión “oscura” de la memoria
  - Problemas eventuales de velocidad
  - Comercial
  - Distribución de ejecutables

Motivación para este curso

Google	MATLAB	Michael Jackson
Septiembre 2006	33.300.000	38.500.000



Planteamiento del curso

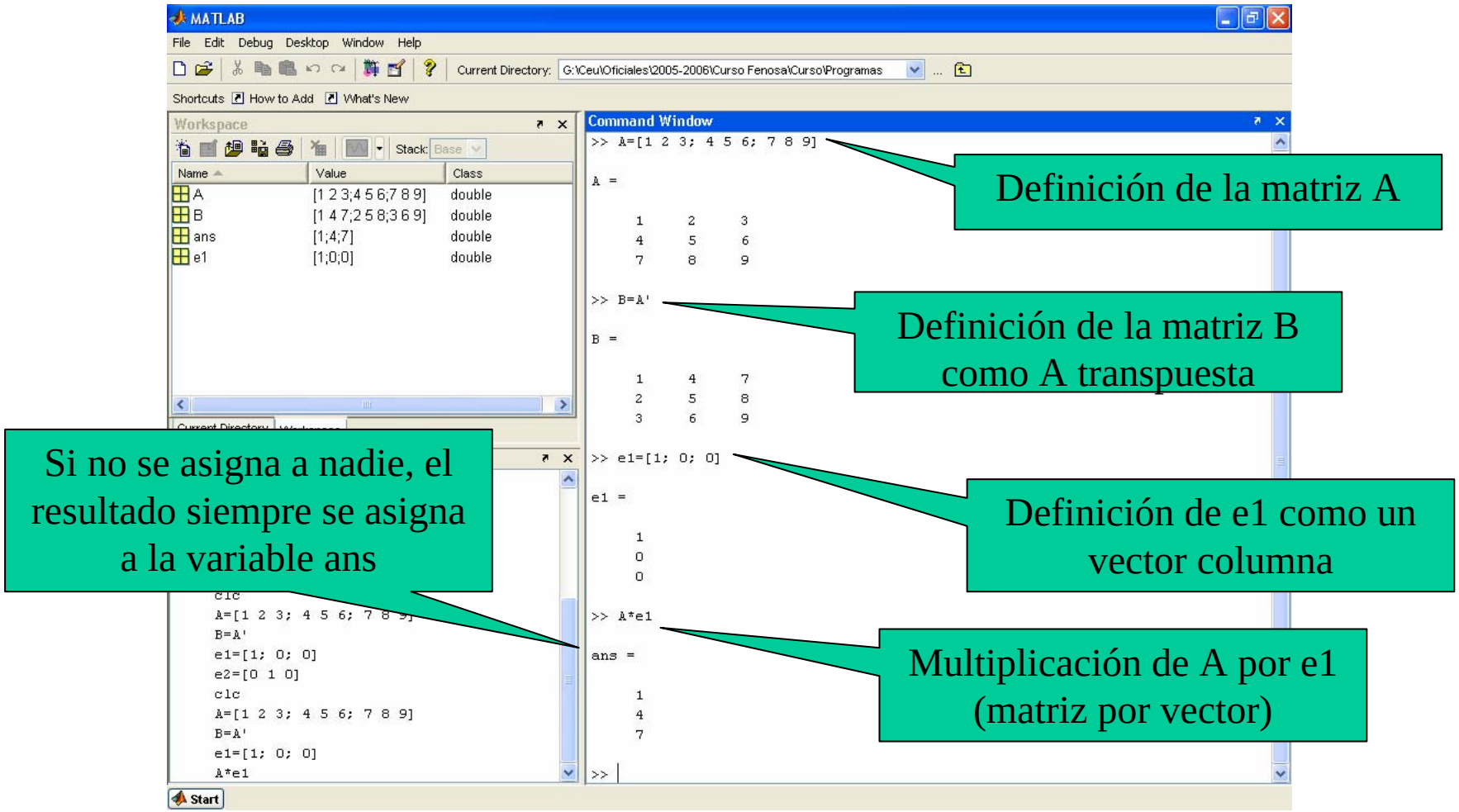
Programa

- Operaciones con matrices y vectores
- Funciones de librería
- Otros tipos de datos en MATLAB: cadenas, hipermatrices, estructuras, celdas
- Programación en MATLAB
- Generación de documentación HTML automáticamente
- Gráficos bidimensionales
- Gráficos tridimensionales
- Interfaces de usuario en MATLAB
- Generación de programas autónomos
- Interacción de MATLAB con Office y Visual Basic
- Librerías de interés práctico: optimización, análisis estadístico, redes neuronales, lógica difusa, ajuste de curvas, sistemas de control, control predictivo, análisis financiero, análisis de series temporales financieras, conexión a bases de datos, generación de informes, sistemas eléctricos de potencia, modelos cuantitativos de energía, etc.

Cronograma del curso

- Sesión 1: Operaciones con matrices y vectores. Funciones de librería.
- Sesión 2: Otros tipos de datos en MATLAB. Programación en MATLAB.
- Sesión 3: Gráficos bidimensionales. Gráficos tridimensionales.
- Sesión 4: Interfaces de usuario en MATLAB. Generación de programas autónomos
- Sesión 5: Librerías de interés práctico
- Sesión 6: Interacción de MATLAB con Office
- Sesión 7: Desarrollo de un proyecto

Sesión 1: Operaciones con matrices y vectores. Funciones de librería.

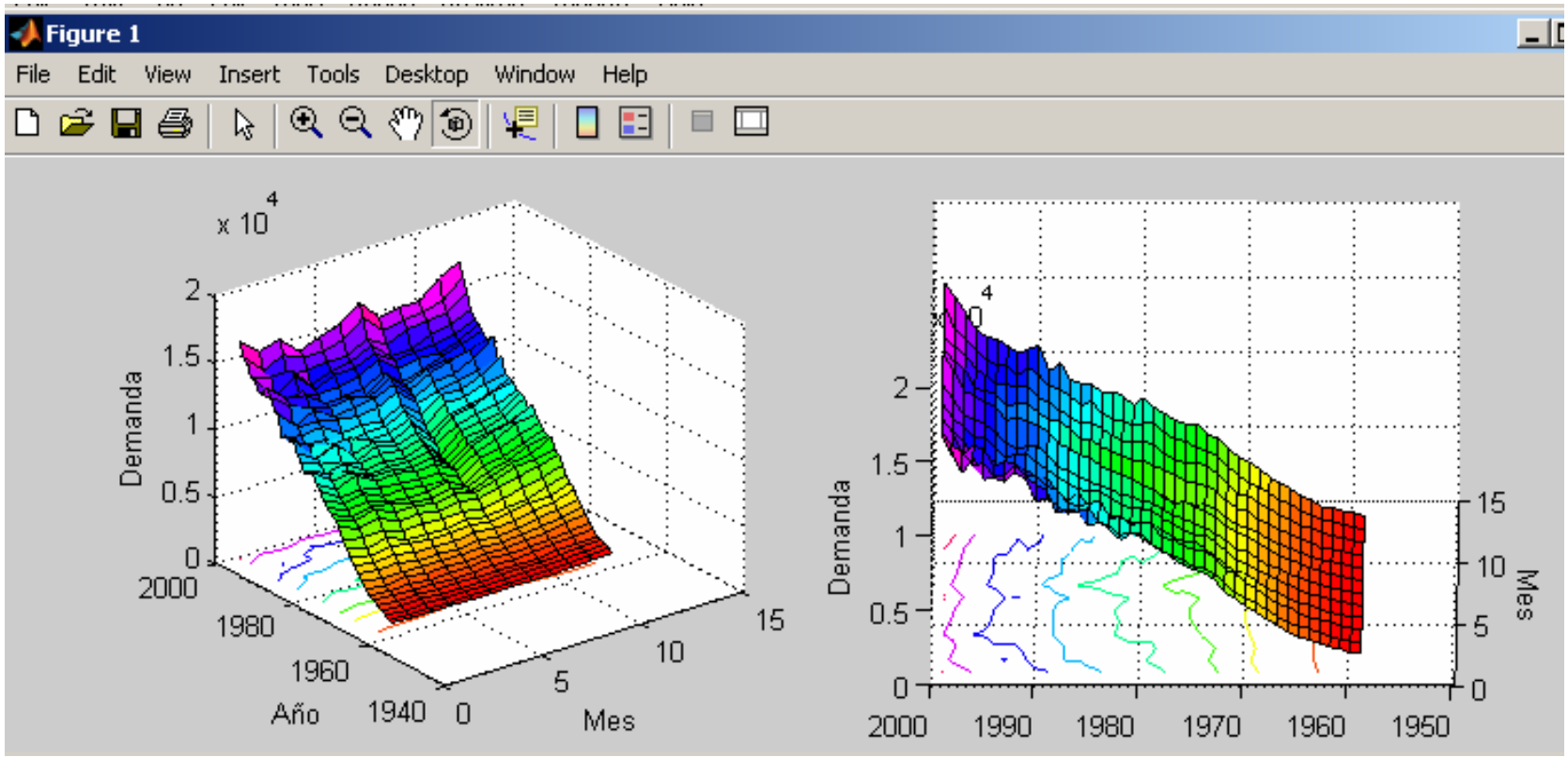


Sesión 2: Otros tipos de datos en MATLAB. Programación en MATLAB.

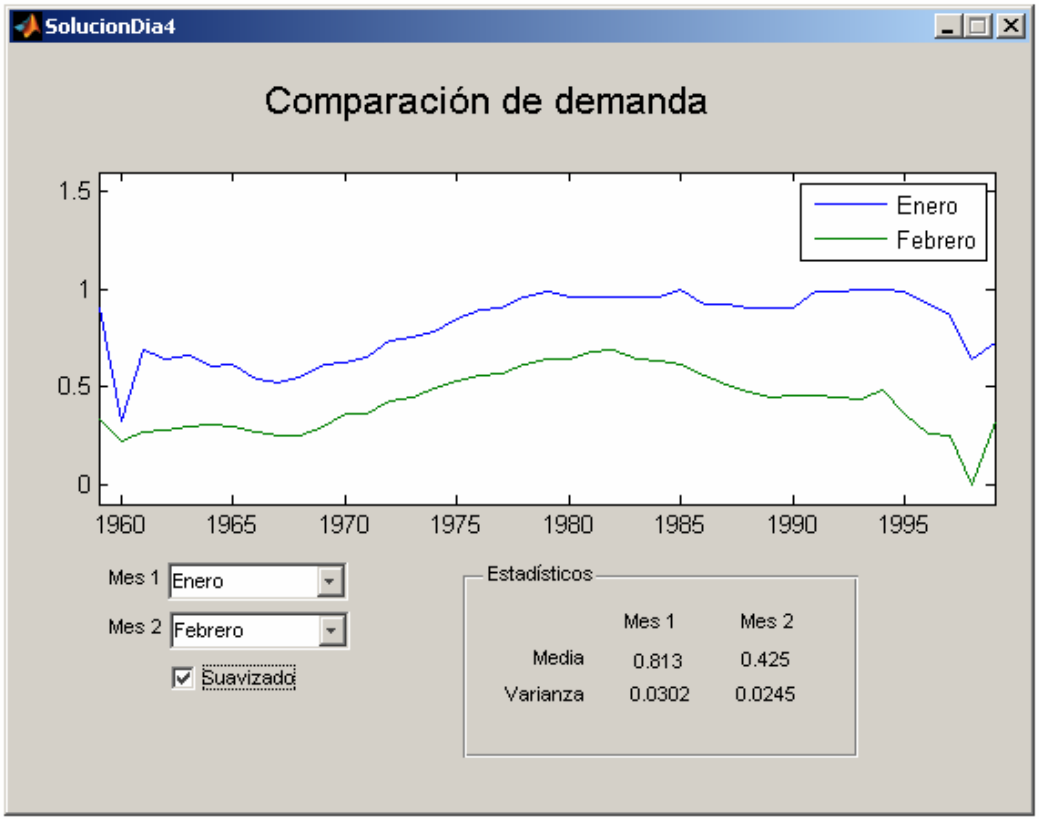
Editor - G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas\SolucionDia1_celdas.m

```
%% Esto es un ejemplo de cómo se programa por celdas
% Únicamente se puede hacer con scripts
%
% El objetivo de este ejemplo es representar los datos
% de demanda eléctrica entre 1959 y 1999

%% Limpiemos todos los datos anteriores
clear   % Limpiar el workspace
clc     % Limpiar la consola

%% Cargar la evolución de la demanda
load('Evolucion demanda.txt')
agnos_demanda=Evolucion_demanda(:,1);
demanda_anual=Evolucion_demanda(:,14);

%% Cargar la evolución de la población
load('Evolucion poblacion.txt')
agnos_poblacion=Evolucion_poblacion(:,1);
poblacion_anual=Evolucion_poblacion(:,2);

%% Mostrar ambas tendencias
plot(agnos_demanda, demanda_anual, agnos_poblacion, poblacion_anual*10);
legend('Demanda anual(GWh)','Población anual (100 hab)')

%% Interpolar la población anualmente entre 1959-1999
poblacion_interpolada=interp1(agnos_poblacion,poblacion_anual,...
    agnos_demanda,'spline');
figure
plot(agnos_demanda, demanda_anual, agnos_demanda, poblacion_interpolada*10);
legend('Demanda anual(GWh)','Población anual (100 hab)')
```

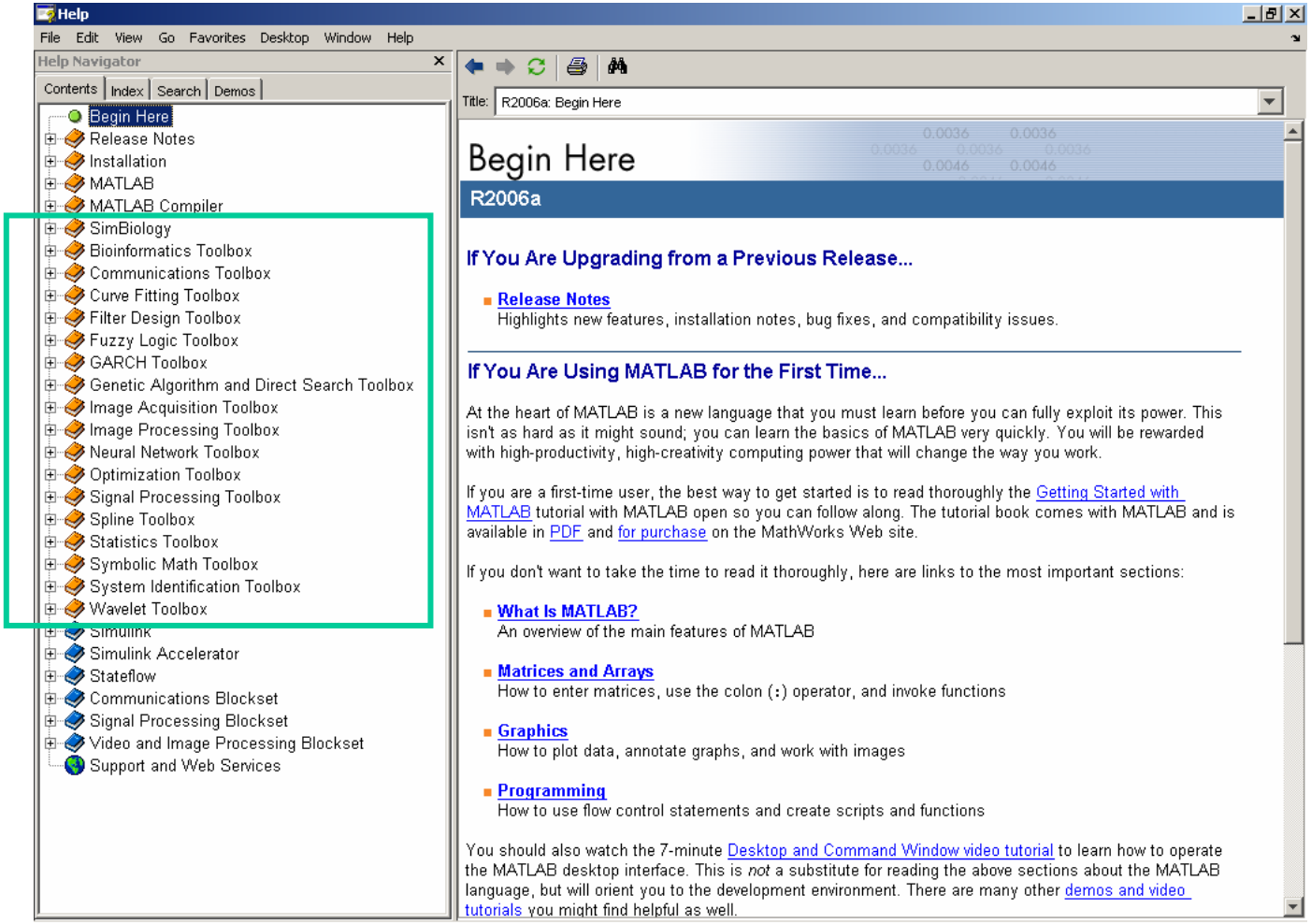
Sesión 3: Gráficos bidimensionales. Gráficos tridimensionales.



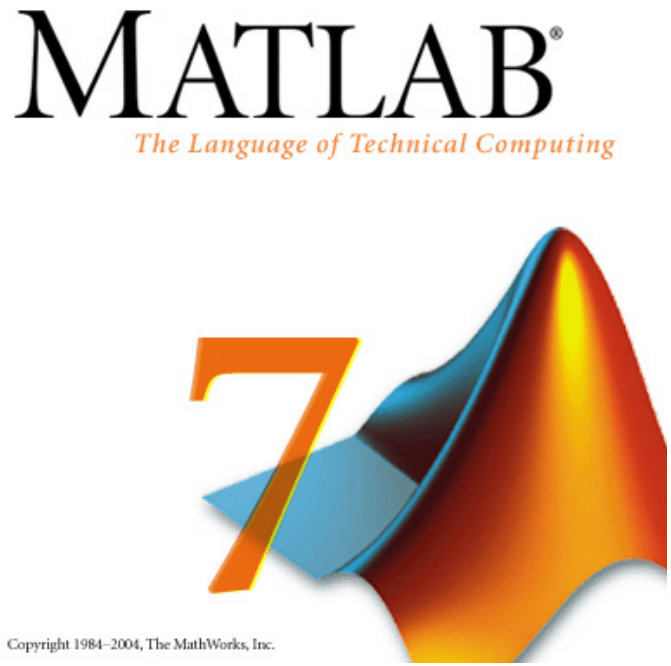
Sesión 4: Interfaces de usuario en MATLAB. Generación de programas autónomos

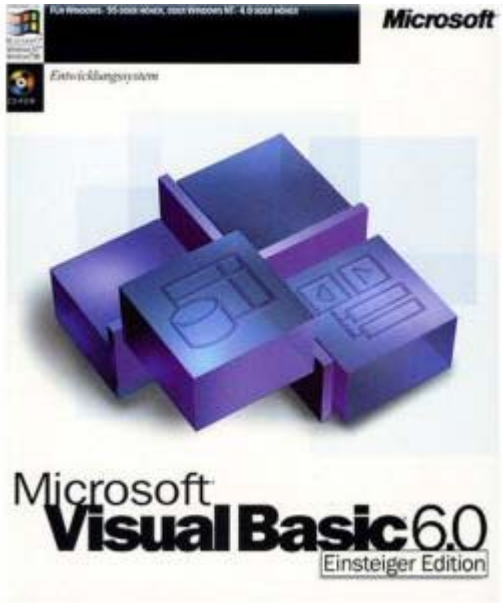


Sesión 5: Librerías de interés práctico



Sesión 6: Interacción de MATLAB con Office y Visual Basic





Sesión 7: Desarrollo de un proyecto

Organización temporal

	Martes	Miércoles	Jueves
9’00	Sesión 0	Sesión 3	Sesión 7
11’00			
11’30	Sesión 1	Sesión 4	Sesión 7
14’00			
16’00	Sesión 2	Sesión 5	
14’00			
16’00	Sesión 2	Sesión 6	
14’00			

Lecturas recomendadas

- Mathworks, Getting Started (187 págs)
- Mathworks, Programming tips (66 págs)
- Mathworks, Quick reference (12 págs)

Recursos

The MathWorks - Support - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

Dirección http://www.mathworks.com/support/ Ir Vínculos

Google references MATLAB help Ir Marcadores PageRank 217 bloqueados Corrector ortográfico Traducir Configuración

The MathWorks
Accelerating the pace of engineering and science

Home | Select Country | Contact Us | Store Search

Create Account | Log In

Products & Services | Industries | Academia | Support | User Community | Company

Contact support
E-mail this page
Print this page

Support

Browse by Product

View All Product Support Resources
Full Product List

Installation and Licensing
Installation Troubleshooting
Get My Passcode(PLP)/License File
System Requirements

License Manager Errors
License Manager Errors

Search Support

Search
Entire Support Site
for:

Manage Account

My Account - Licenses, Orders...
Create an Account
Account Benefits
Customer Service Options

Browse by Resource

Documentation
Product Documentation
Index of MATLAB Examples
Function List - All Products
Tech Notes / How-To Guides

Bug Reports

User Community
File Exchange
MATLAB Newsgroup
Link Exchange

Additional Resources
MATLAB and Simulink Based Books

New Support Resources
Learn how to effectively use MathWorks products and quickly resolve technical problems.
» Learn more

R2006b
Available Sept. 2006
» Learn more
» Download now

Track Your Support Requests Online
Create and monitor customer service and technical support requests.
» Learn more

Provide Feedback

(Quedan 1 elemento) Abriendo página http://www.mathworks.com/support/... Internet

Recursos

The MathWorks - Online Documentation, R2006b - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

Dirección http://www.mathworks.com/access/helpdesk/help/helpdesk.html Ir Vínculos

Google references MATLAB help Ir Marcadores PageRank 217 bloqueados Corrector ortográfico Traducir Configuración

The MathWorks
Accelerating the pace of engineering and science

Home | Select Country | Contact Us | Store Search

Create Account | Log In

Products & Services | Industries | Academia | Support | User Community | Company

E-mail this page
Print this page

Documentation

Documentation for MathWorks Products, R2006b

Alphabetical list by product

Release Notes for R2006b
Installation Guides
License Administration Guide
Function List - All Products

Search

R2006b Documentation

for:

advanced search search tips

MATLAB
MATLAB® (including *External Interfaces/API, GUIDE, Handle Graphics, File I/O, Notebook*)
MATLAB® Builder for .NET
MATLAB® Builder for Excel
MATLAB® Builder for Java™
MATLAB® Compiler
Distributed Computing Toolbox
MATLAB® Distributed Computing Engine
MATLAB® Report Generator
SimBiology™
SystemTest

MATLAB Toolboxes
Aerospace Toolbox

Simulink
Simulink®
Embedded Target for Infineon C166® Microcontrollers
Embedded Target for Motorola® HC12
Embedded Target for Motorola® MPC555
Embedded Target for TI C2000™ DSP
Embedded Target for TI C6000™ DSP
Link for Code Composer Studio™ Development Tools
Link for Incisive®
Link for ModelSim®
Link for TASKING®
Real-Time Windows Target
Real-Time Workshop®

R2006b
Available Sept. 2006
» Learn more
» Download now

Provide Feedback
If you are a MATLAB user in the Boston area, provide Web site feedback and earn $100.
» Sign up today

Internet

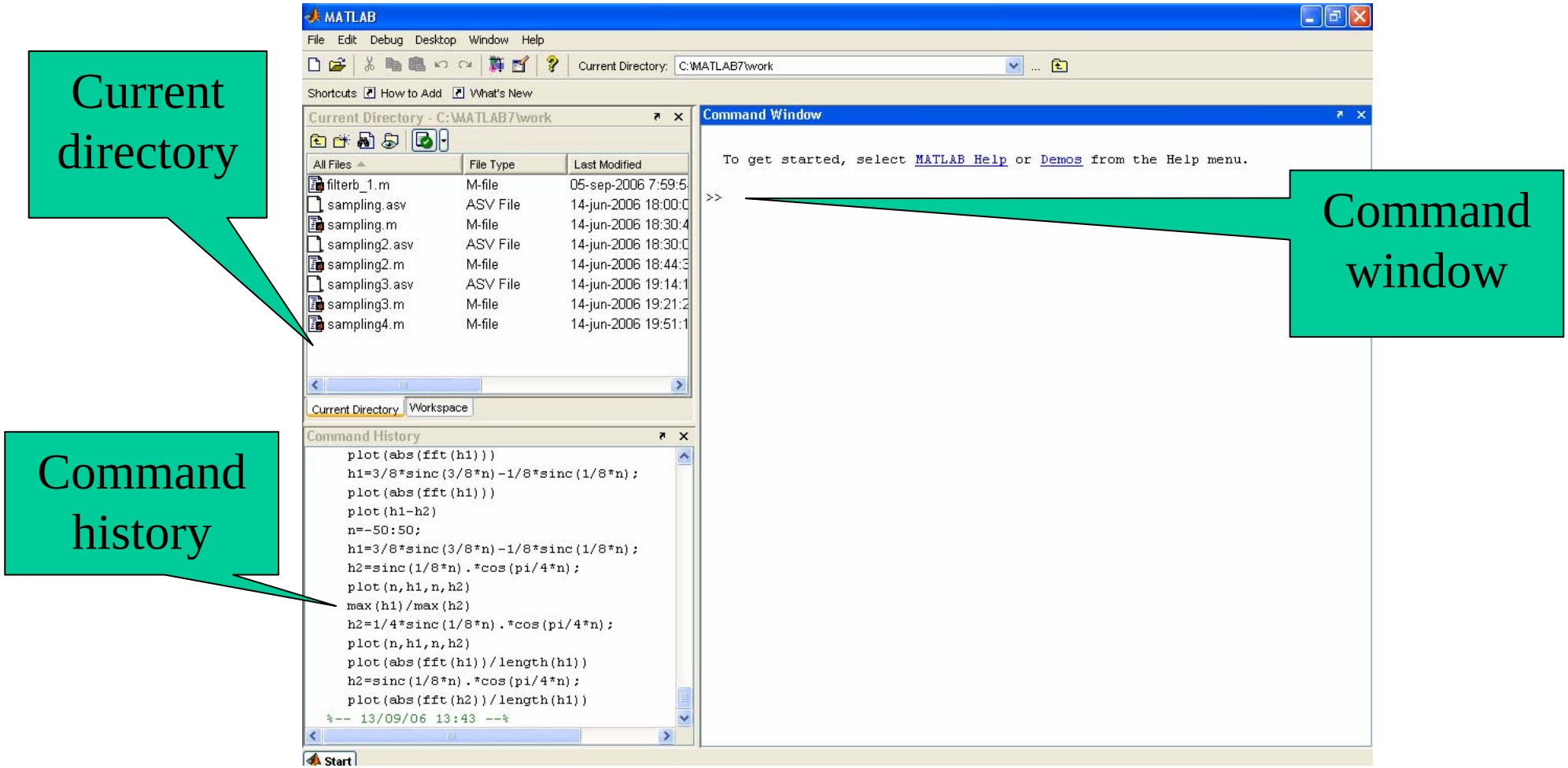
Recursos


MATLAB Central - Home - Microsoft Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda
Atrás Búsqueda Favoritos
Dirección http://www.mathworks.com/matlabcentral/
Ir Vínculos
Google references MATLAB help Ir Marcadores PageRank 217 bloqueados Corrector ortográfico Traducir Configuración
MATLAB CENTRAL
An open exchange for the MATLAB and Simulink user community
Hosted by The MathWorks
Search: MATLAB Central Go
File Exchange | MATLAB Newsgroup | Link Exchange | Blogs | Contest | MathWorks.com
File Exchange
User-contributed code library
Recently Added Files...
Whats New - MATLAB - R2006b - Ke...
Knight_Tour - Krishna Lalit...
structure tensor - introduction ...
MATLAB Newsgroup
MATLAB Usenet newsgroup access
Most Active Threads...
Reassigning Commands Should NOT Be...
Displaying final numerical results...
cellfun mean w subset indexes
Now Available
R2006b
Learn more
Download now
Job Opening
Web Community Business Analyst
Free Webinars
Open Architecture Solutions for Rapid Prototyping and HIL Using Simulink
October 5, 2006
Link to Us
Link to MATLAB Central
MATLAB Screensaver
Download Now
Link Exchange
MATLAB related web sites
Recently Added Links...
MAT-file Viewer plugin for Total Command...
Hplot (Claudio Gambelli)
Video seminar : Direct methods for spars...
MATLAB Central Blogs
Doug's Pick of the Week
Latest pick: Simulink pick
Loren on the Art of MATLAB
Latest entry: Working with Low Level File I/O and Encodings
Steve on Image Processing
Latest entry: Unusual red-eye reduction technique
Public Submission Policy
NOTICE: Any content you submit to MATLAB Central, including personal information, is not subject to the protections which may be afforded information collected unde
including: incluyendo
athWorks, Inc. Web site. You are entirely responsible for all content that you uplo
otherwise make available via MATLAB Central. The MathWorks does not control the content posted by visitors to MATLAB Central and, does not guarantee the accuracy, integrity, or quality of such content. Under no circumstances will The MathWorks be liable in any way for any content not authored by The MathWorks, or any loss or damage of any kind incurred as a result of the use of any content posted, e-mailed, transmitted or otherwise
(Quedan 1 elemento) Abriendo página http://www.mathworks.com/matlabcentral/...
Internet

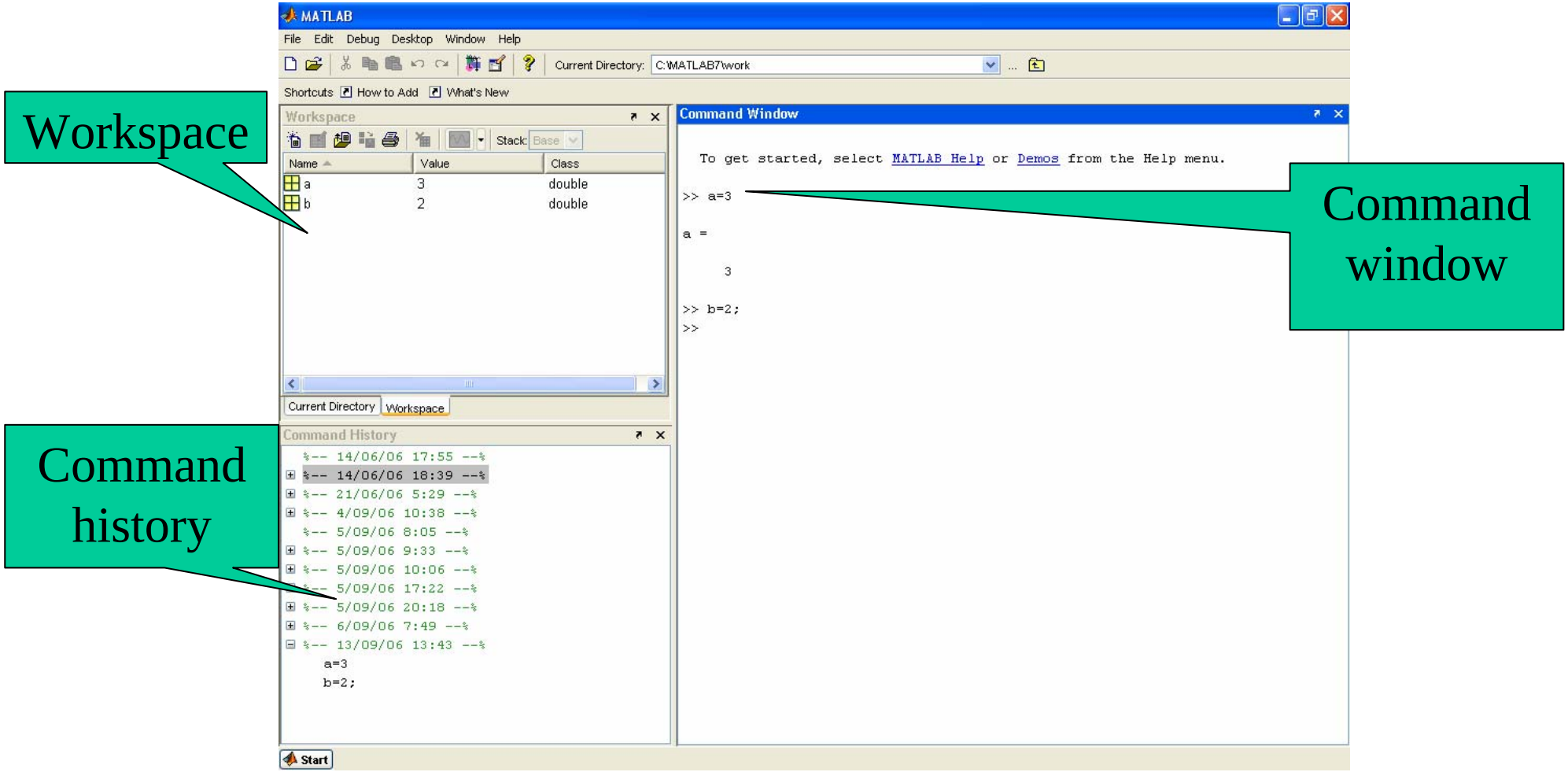
Bibliografía

- Mathworks. *MATLAB online help*
- J. Atencia, R. Néstar. *Aprenda MATLAB 6.0 como si estuviera en primero*. Univ. Navarra, 2001.
- C. Pérez. *MATLAB y sus aplicaciones en las ciencias y la ingeniería*. Prentice Hall, 2002
- G. Amos. *MATLAB: una introducción con ejemplos prácticos*. Reverte, 2006.

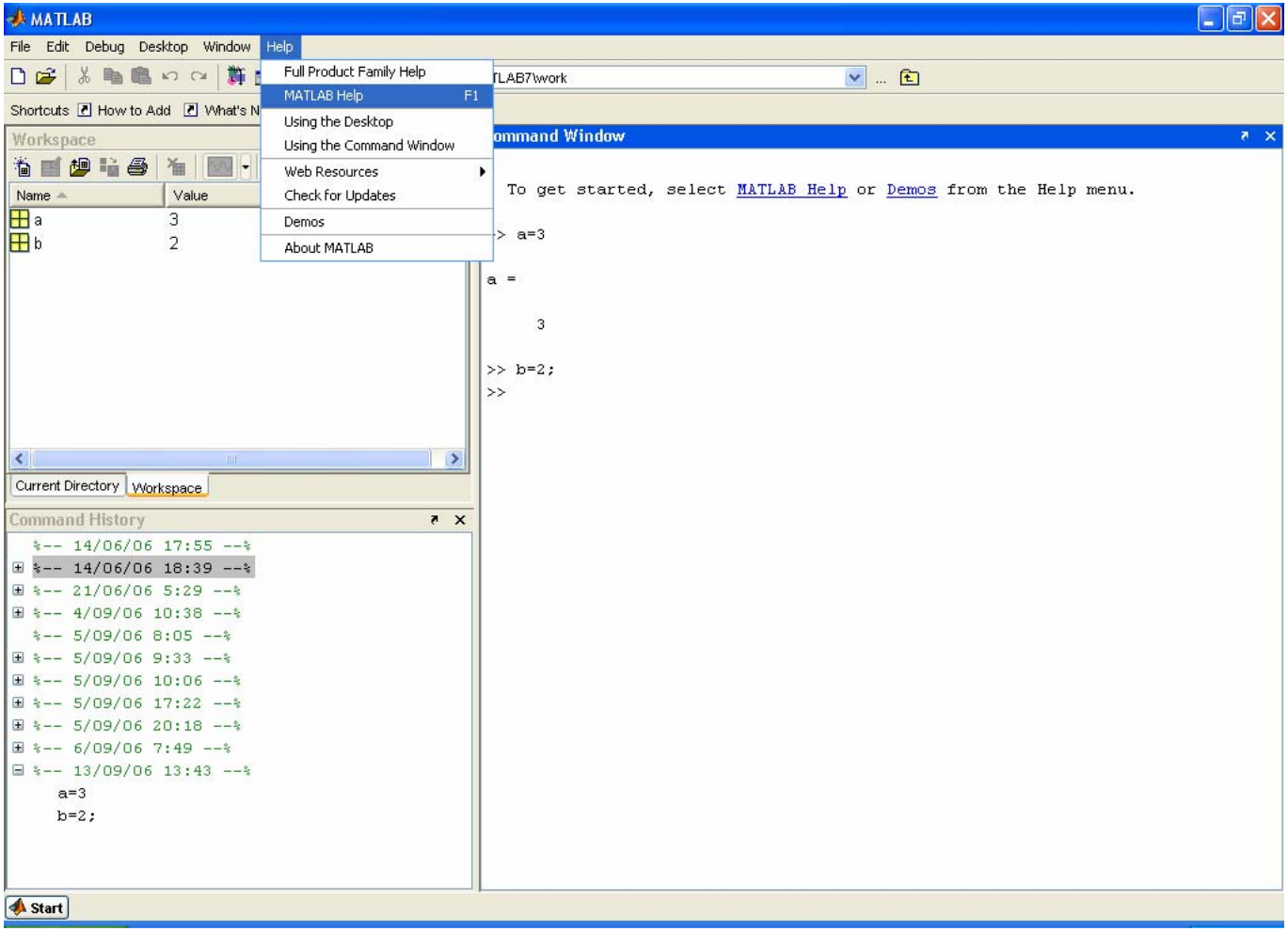
Entorno de MATLAB



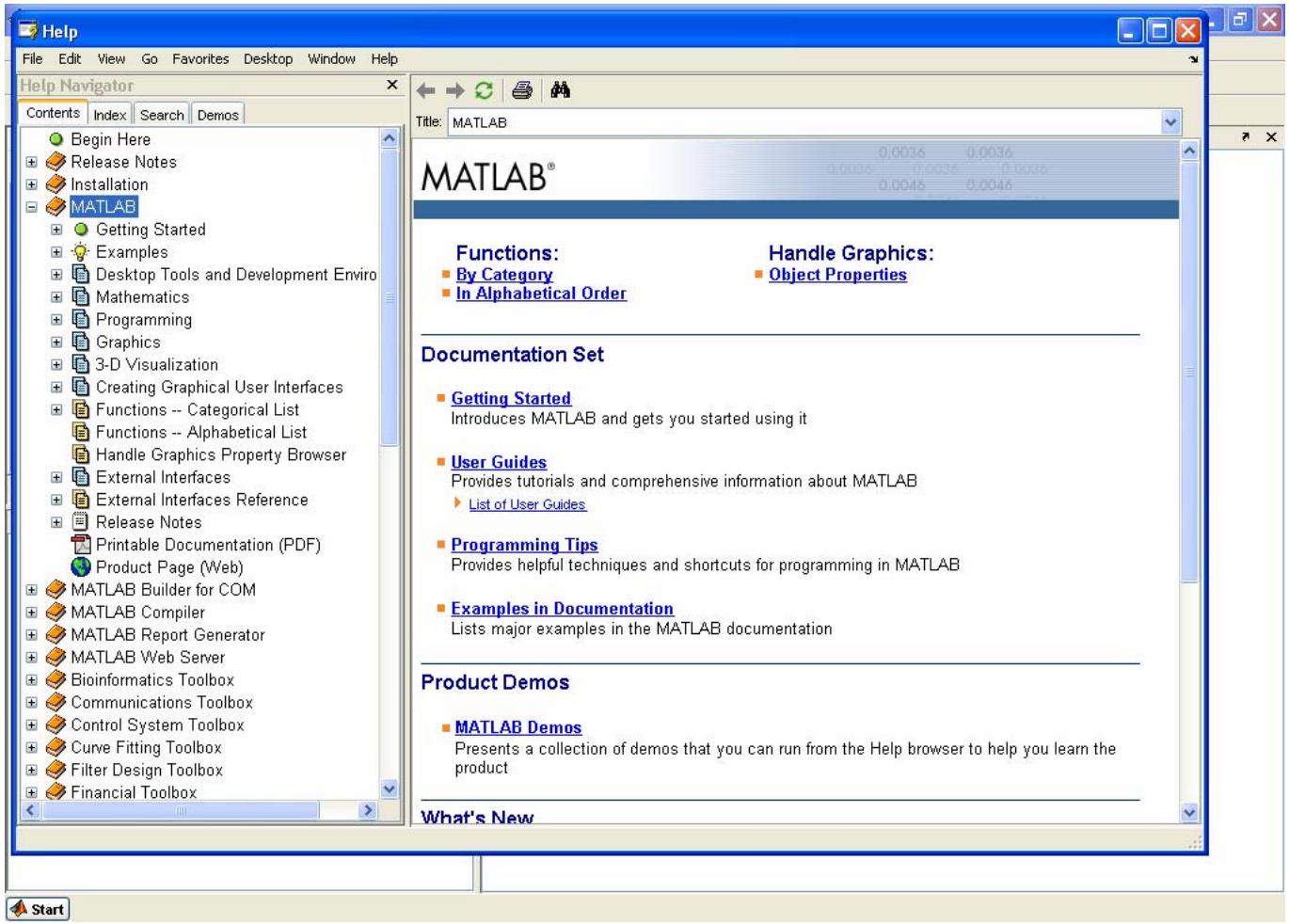
Entorno de MATLAB



Entorno de MATLAB: Ayuda



Entorno de MATLAB: Ayuda



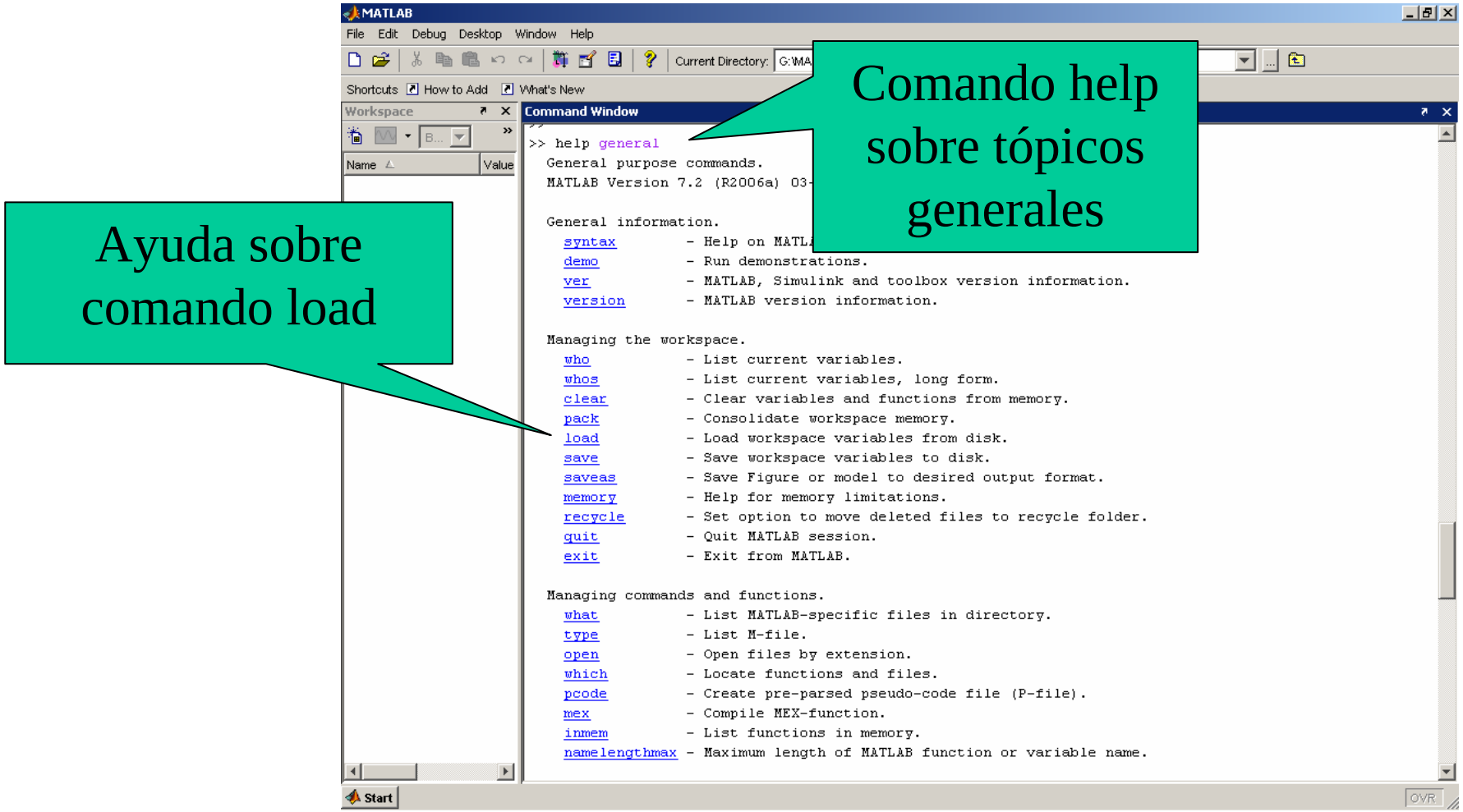
Entorno de MATLAB: Ayuda

Comando help

Ayuda sobre tópicos generales

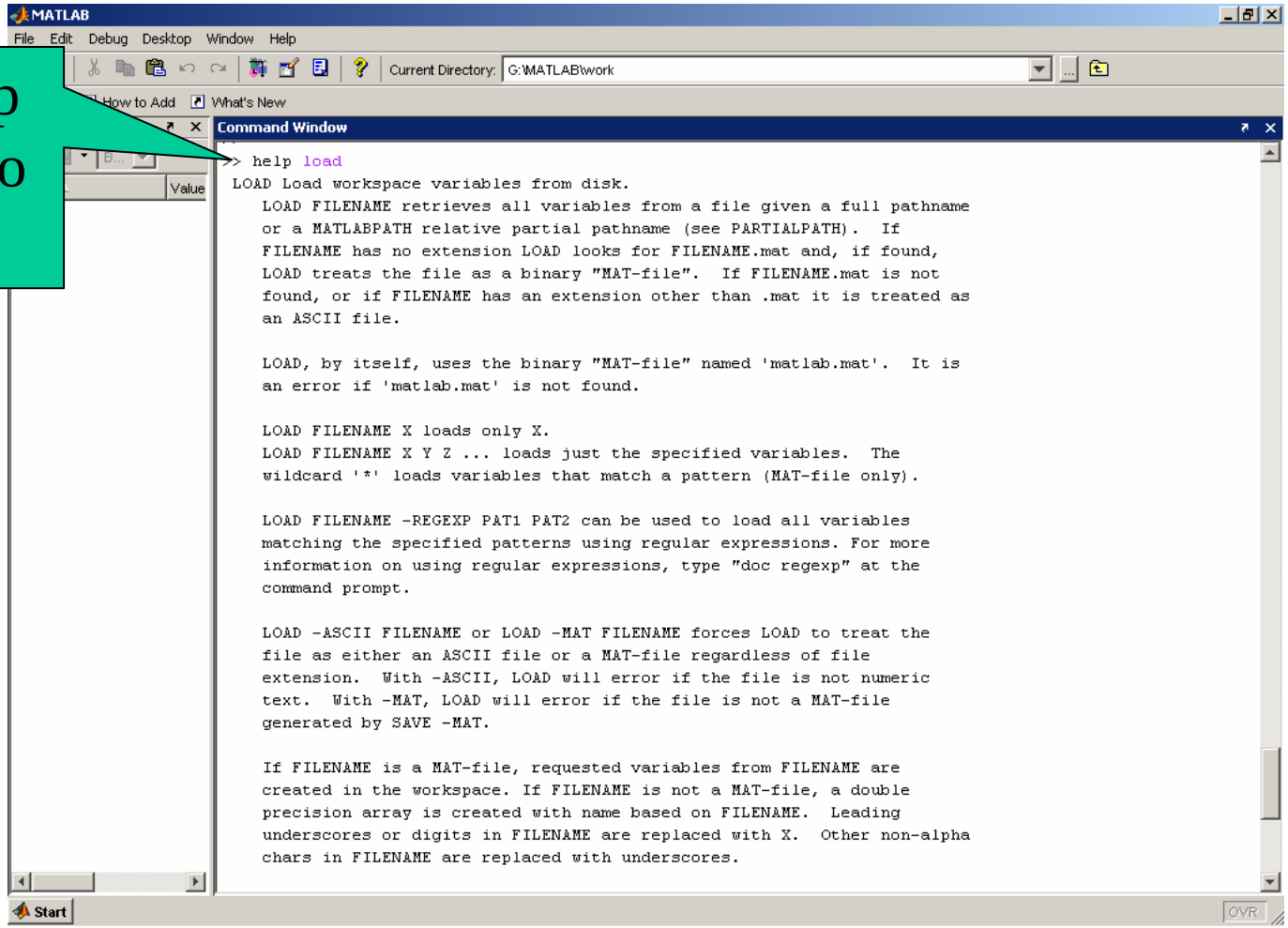
```
>> help
HELP topics

Ceu\MyMatlabStuff     -  (No table of contents file)
matlab\general        -  General purpose commands.
matlab\ops            -  Operators and special characters.
matlab\lang           -  Programming language constructs.
matlab\elmat          -  Elementary matrices and matrix manipulation.
matlab\elfun          -  Elementary math functions.
matlab\specfun        -  Specialized math functions.
matlab\matfun         -  Matrix functions - numerical linear algebra.
matlab\datafun        -  Data analysis and Fourier transforms.
matlab\polyfun        -  Interpolation and polynomials.
matlab\funfun         -  Function functions and ODE solvers.
matlab\sparfun        -  Sparse matrices.
matlab\scribe         -  Annotation and Plot Editing.
matlab\graph2d        -  Two dimensional graphs.
matlab\graph3d        -  Three dimensional graphs.
matlab\specgraph      -  Specialized graphs.
matlab\graphics       -  Handle Graphics.
matlab\uitools        -  Graphical user interface tools.
matlab\strfun         -  Character strings.
matlab\imagesci       -  Image and scientific data input/output.
matlab\iofun          -  File input and output.
matlab\audiovideo     -  Audio and Video support.
matlab\timefun        -  Time and dates.
matlab\datatypes      -  Data types and structures.
matlab\verctrl        -  Version control.
matlab\codetools      -  Commands for creating and debugging code.
matlab\helptools      -  Help commands.
matlab\winfun         -  Windows Operating System Interface Files (COM/DDE)
matlab\demos          -  Examples and demonstrations.
matlab\timeseries     -  Time series data visualization and exploration.
```

Entorno de MATLAB: Ayuda

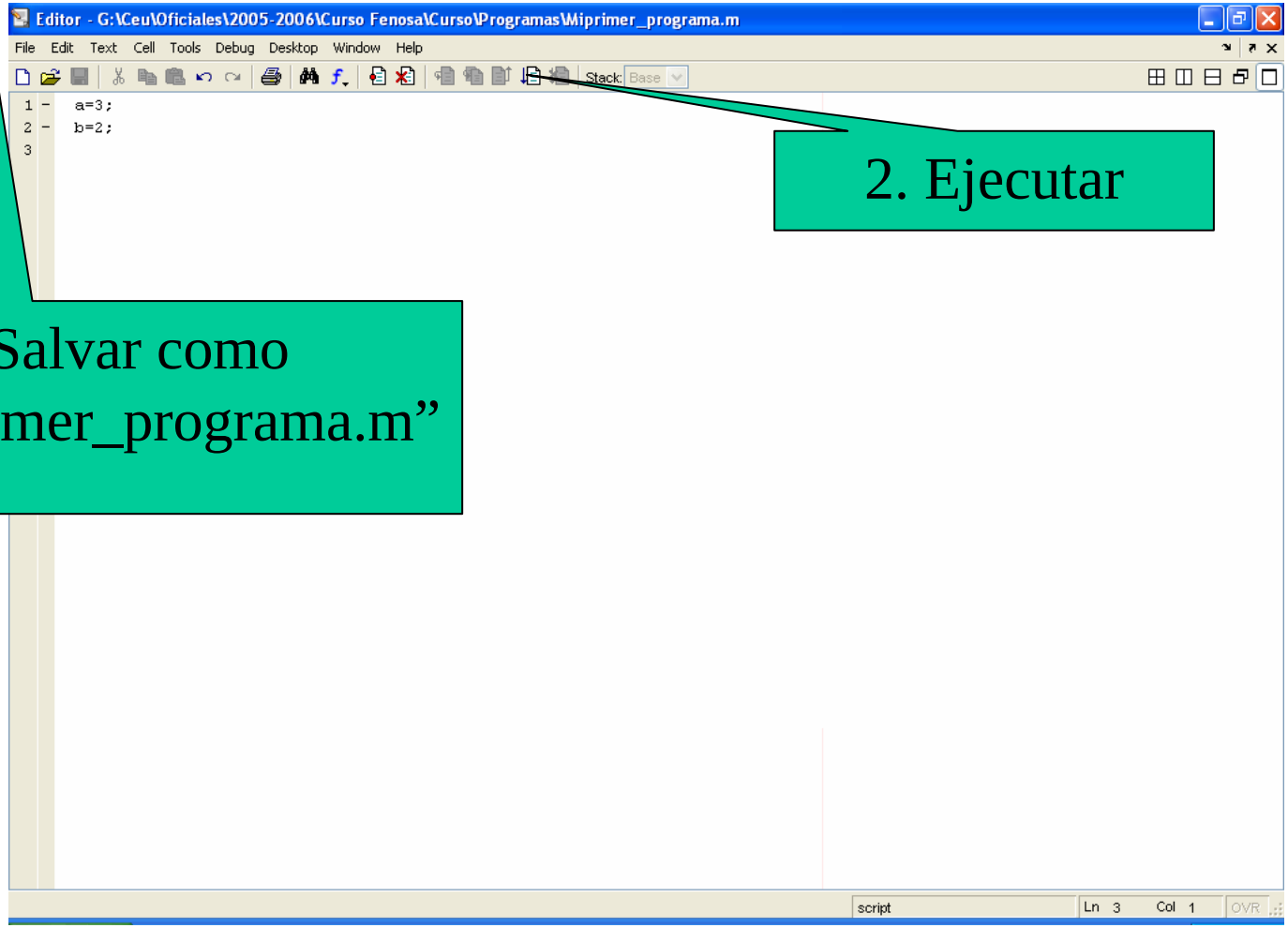


Entorno de MATLAB: Ayuda

Comando help sobre comando load



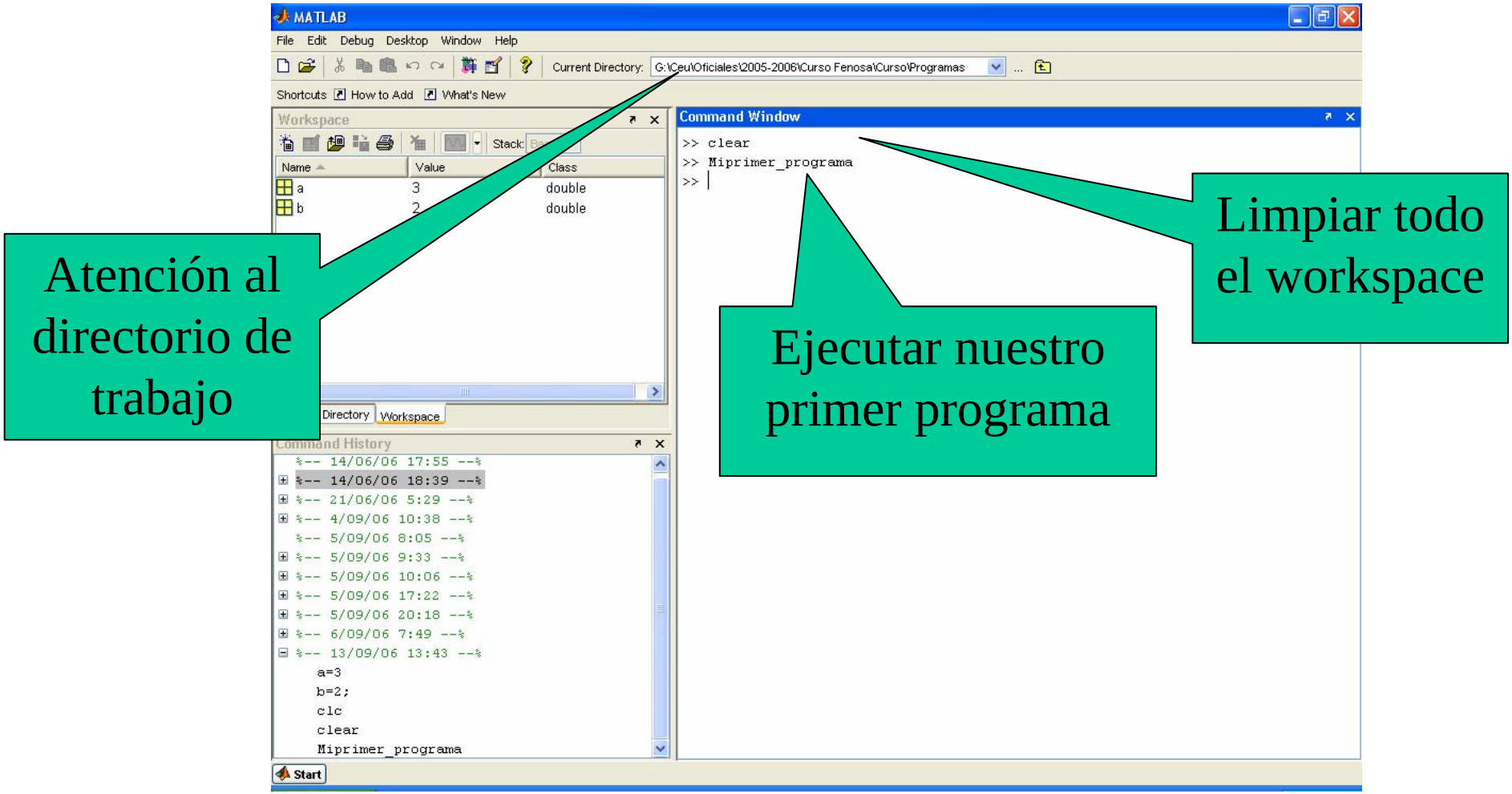
Entorno de MATLAB: El editor



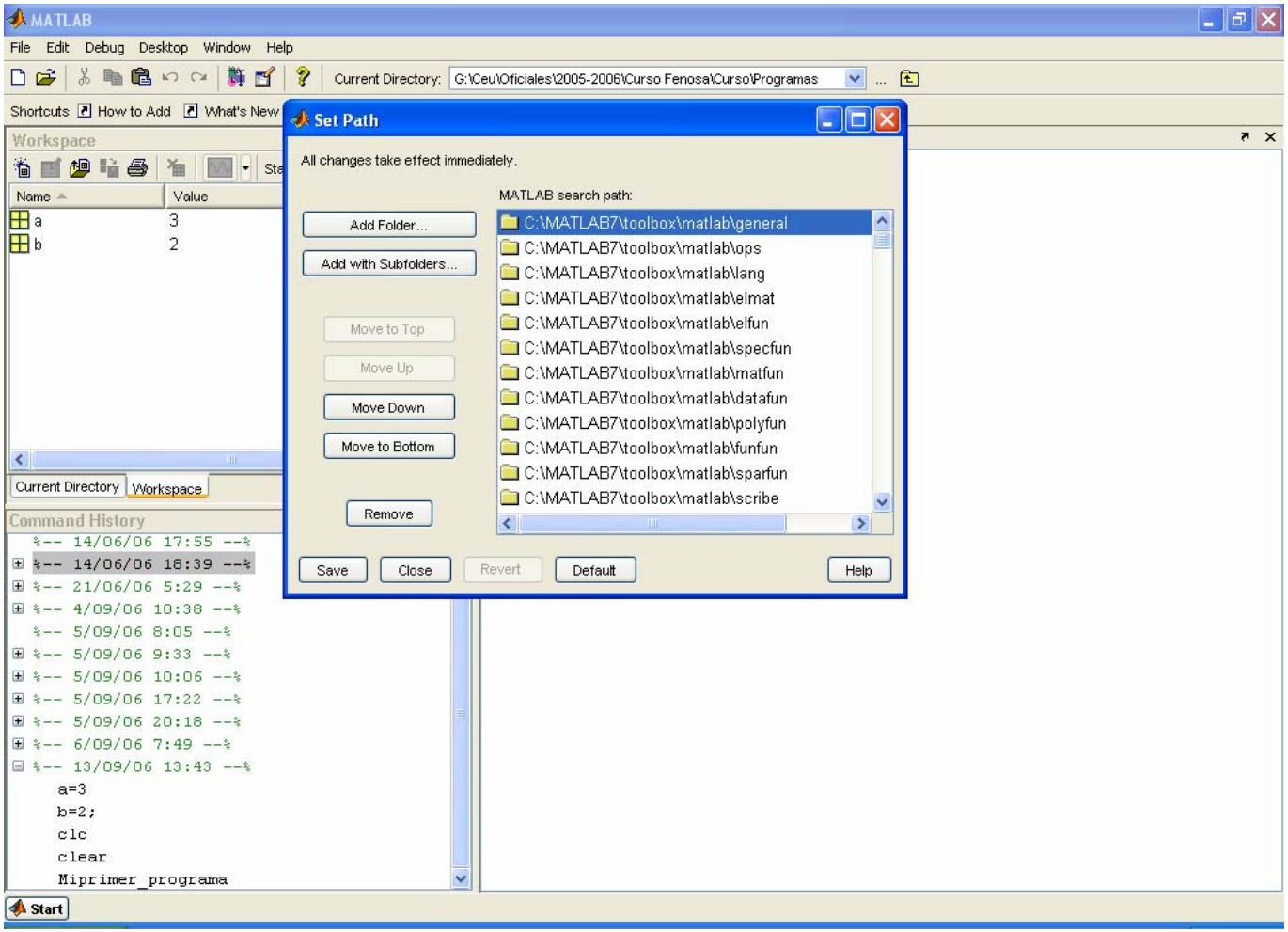
1. Salvar como "Miprimer_programa.m"

2. Ejecutar

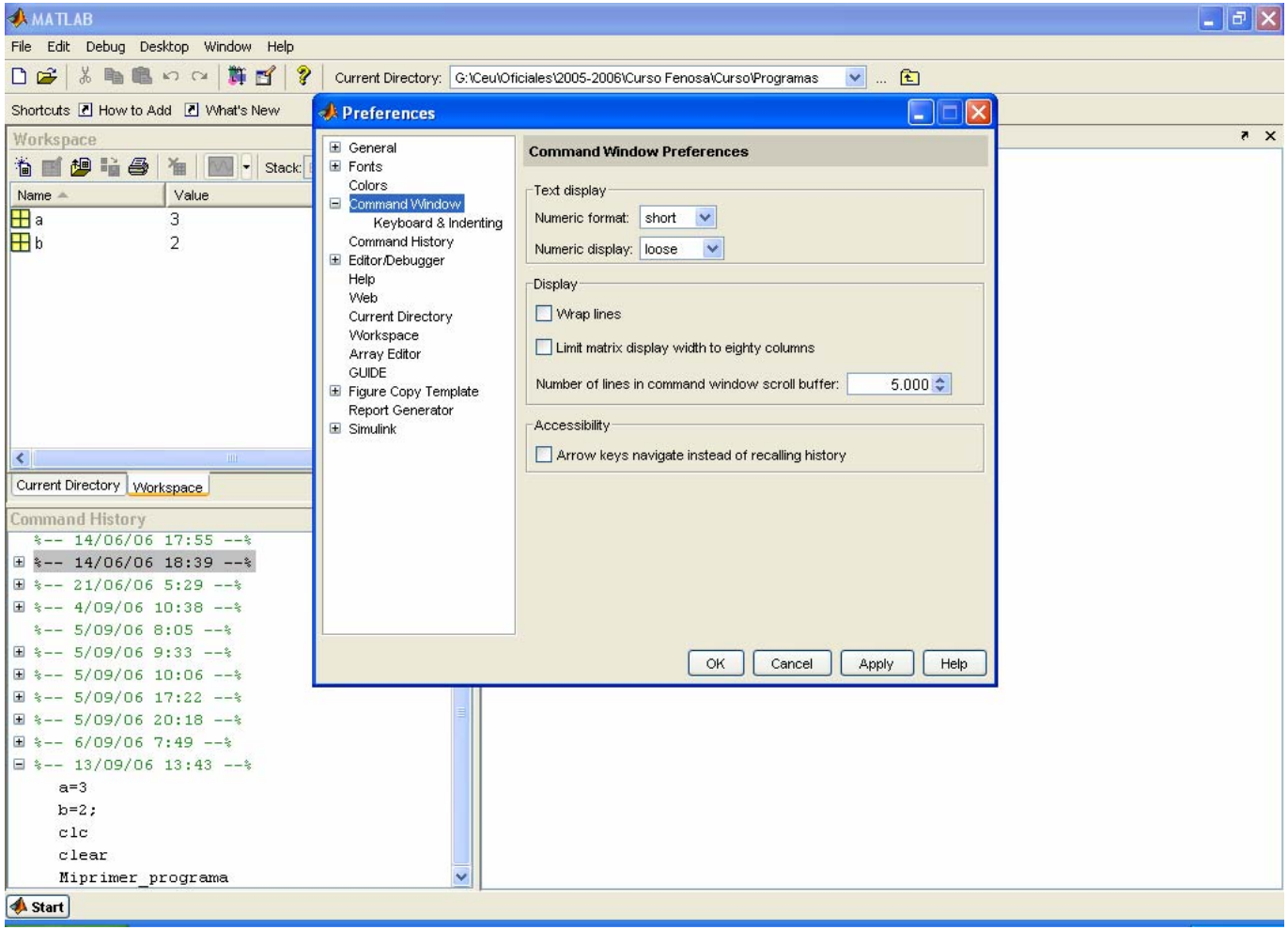
Entorno de MATLAB



Entorno de MATLAB: File -> Set Path



Entorno de MATLAB: File -> Preferences



CEU
Universidad
San Pablo

CURSO de UTILIZACIÓN PRÁCTICA de MATLAB

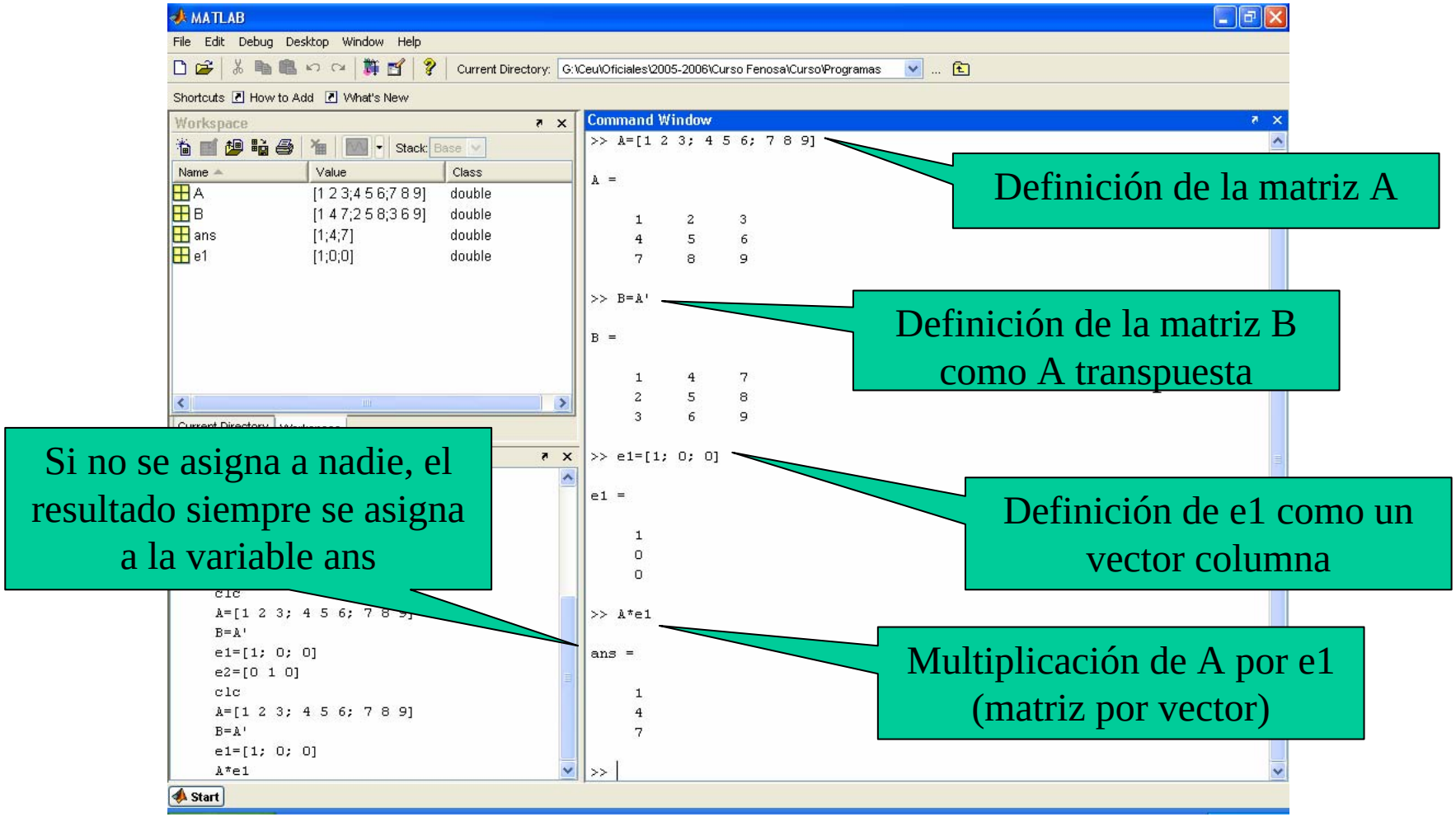
Sesión 1

Carlos Óscar Sánchez Sorzano, Ph.D.
Madrid, July 17th 2006

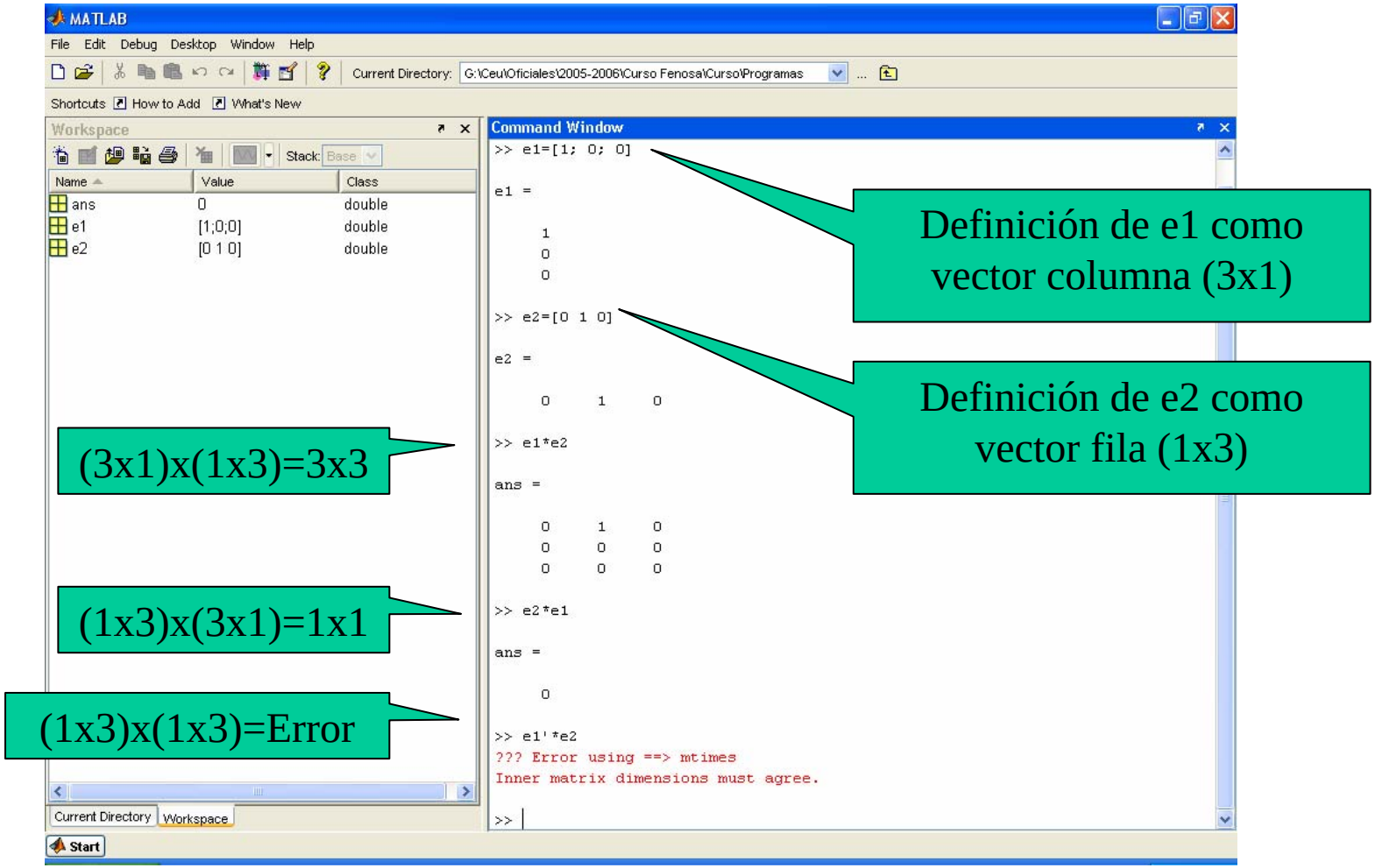
Cronograma del curso

- **Día 1: Operaciones con matrices y vectores. Funciones de librería.**
- Día 2: Otros tipos de datos en MATLAB. Programación en MATLAB.
- Día 3: Gráficos bidimensionales. Gráficos tridimensionales.
- Día 4: Interfaces de usuario en MATLAB. Generación de programas autónomos
- Día 5: Librerías de interés práctico
- Día 6: Interacción de MATLAB con Office y Visual Basic
- Día 7: Desarrollo de un proyecto

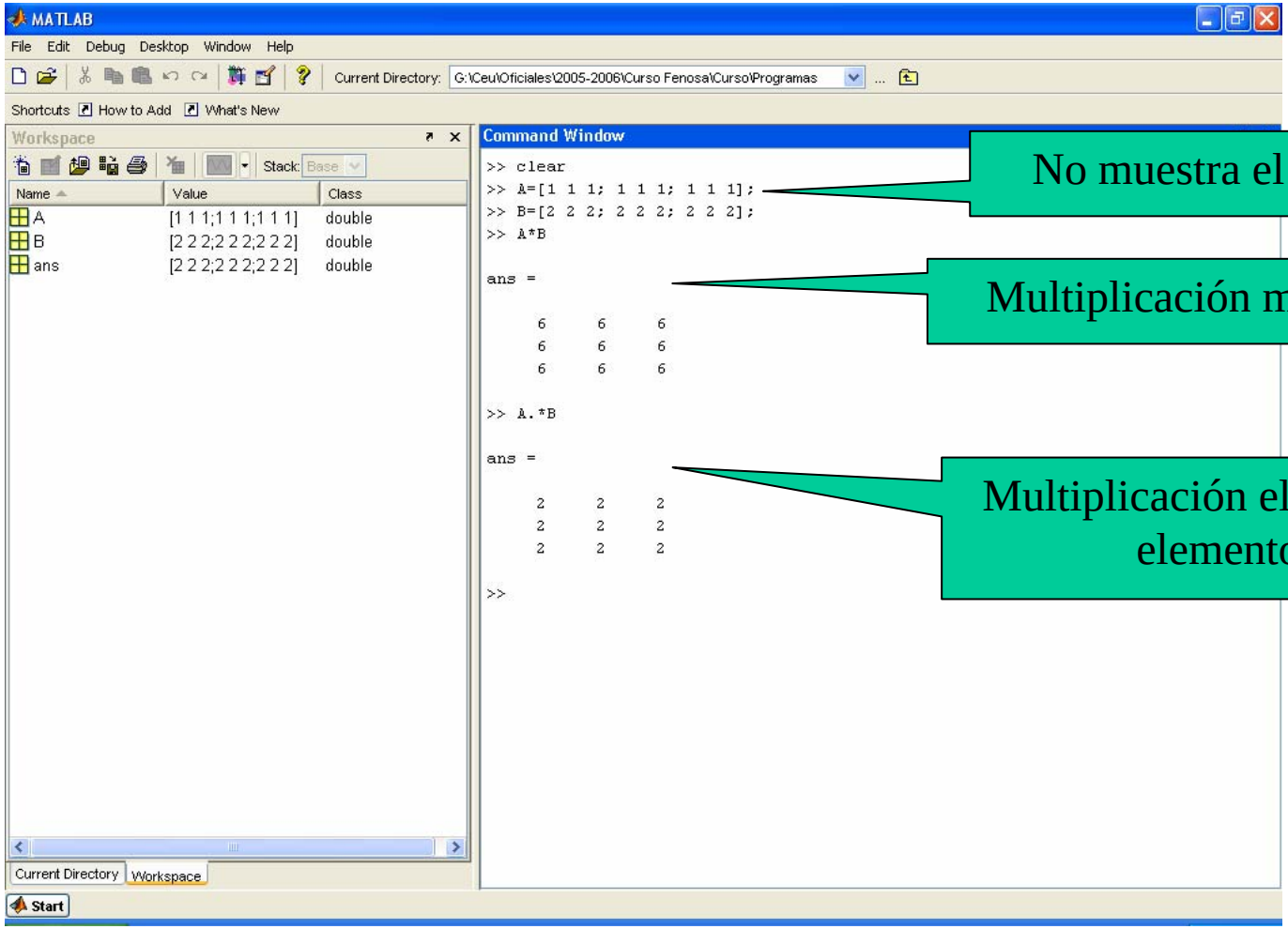
Definición de vectores y matrices



Definición de vectores y matrices



Operaciones con vectores y matrices



No muestra el resultado

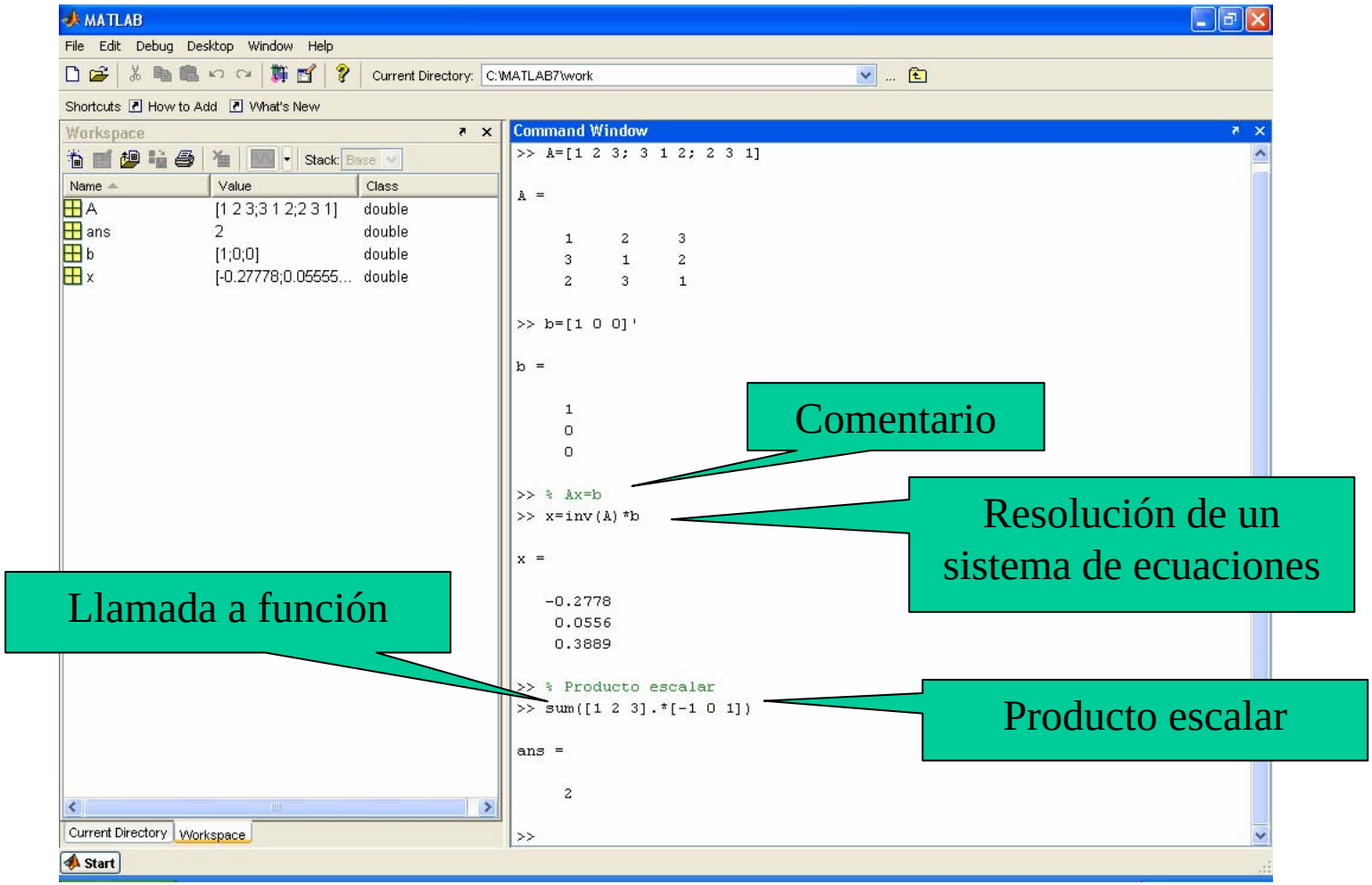
Multiplicación matricial

Multiplicación elemento a elemento

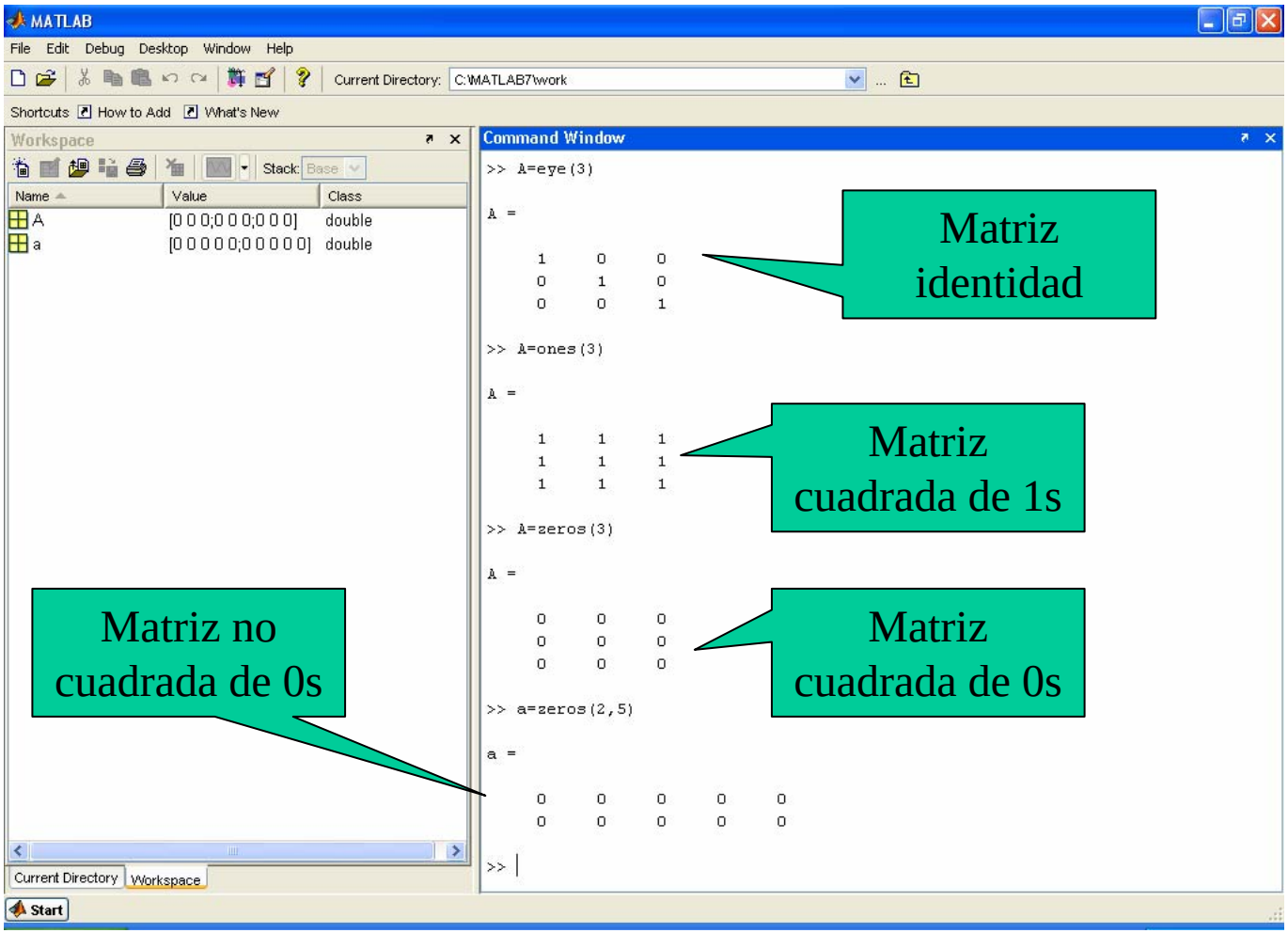
Operaciones con vectores y matrices

- + adición o suma
- – sustracción o resta
- * multiplicación
- ' traspuesta
- ^ potenciación
- \ división-izquierda
- / división-derecha
- .* producto elemento a elemento
- ./ y .\ división elemento a elemento
- .^ elevar a una potencia elemento a elemento

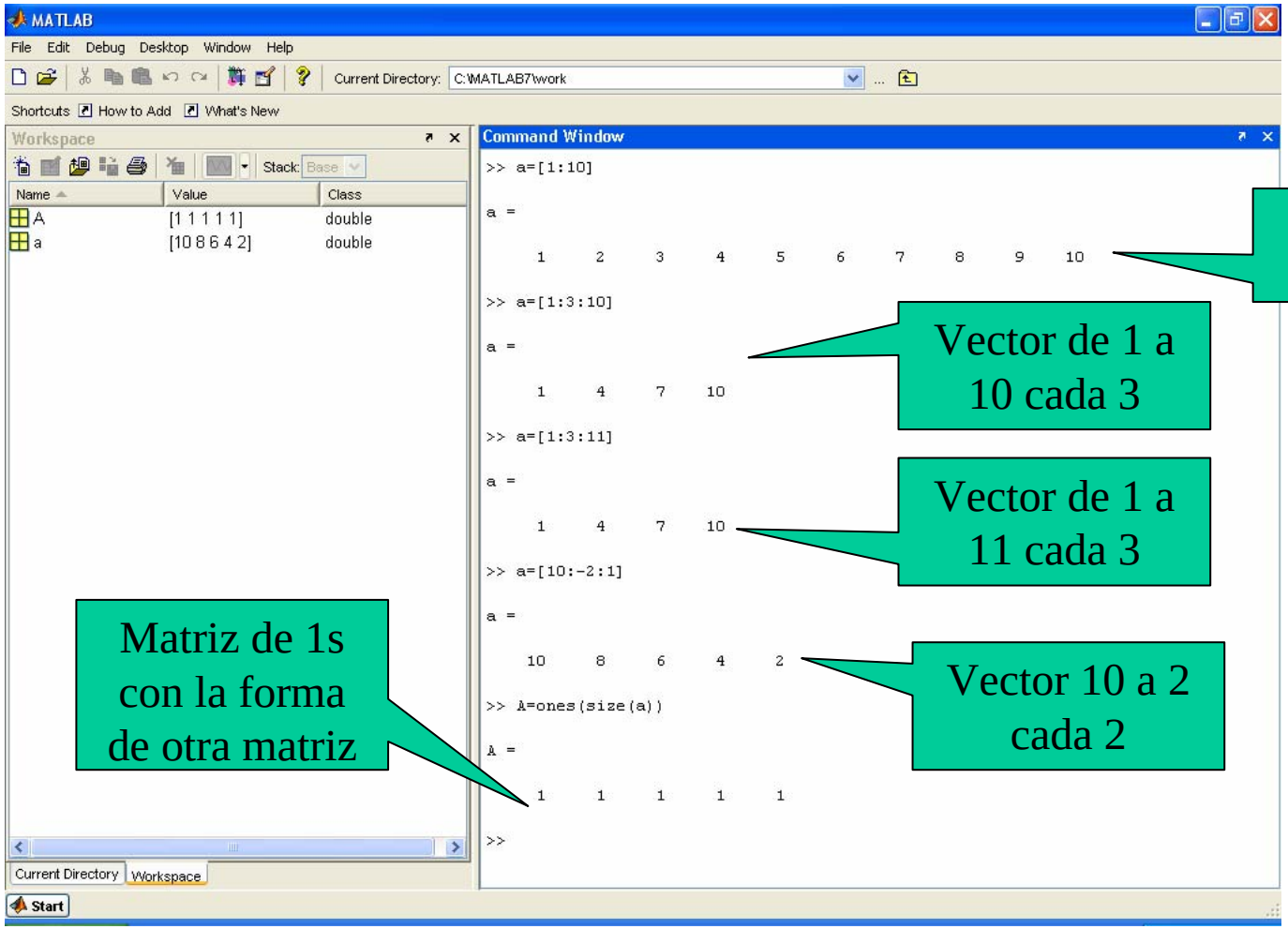
Un poco de álgebra



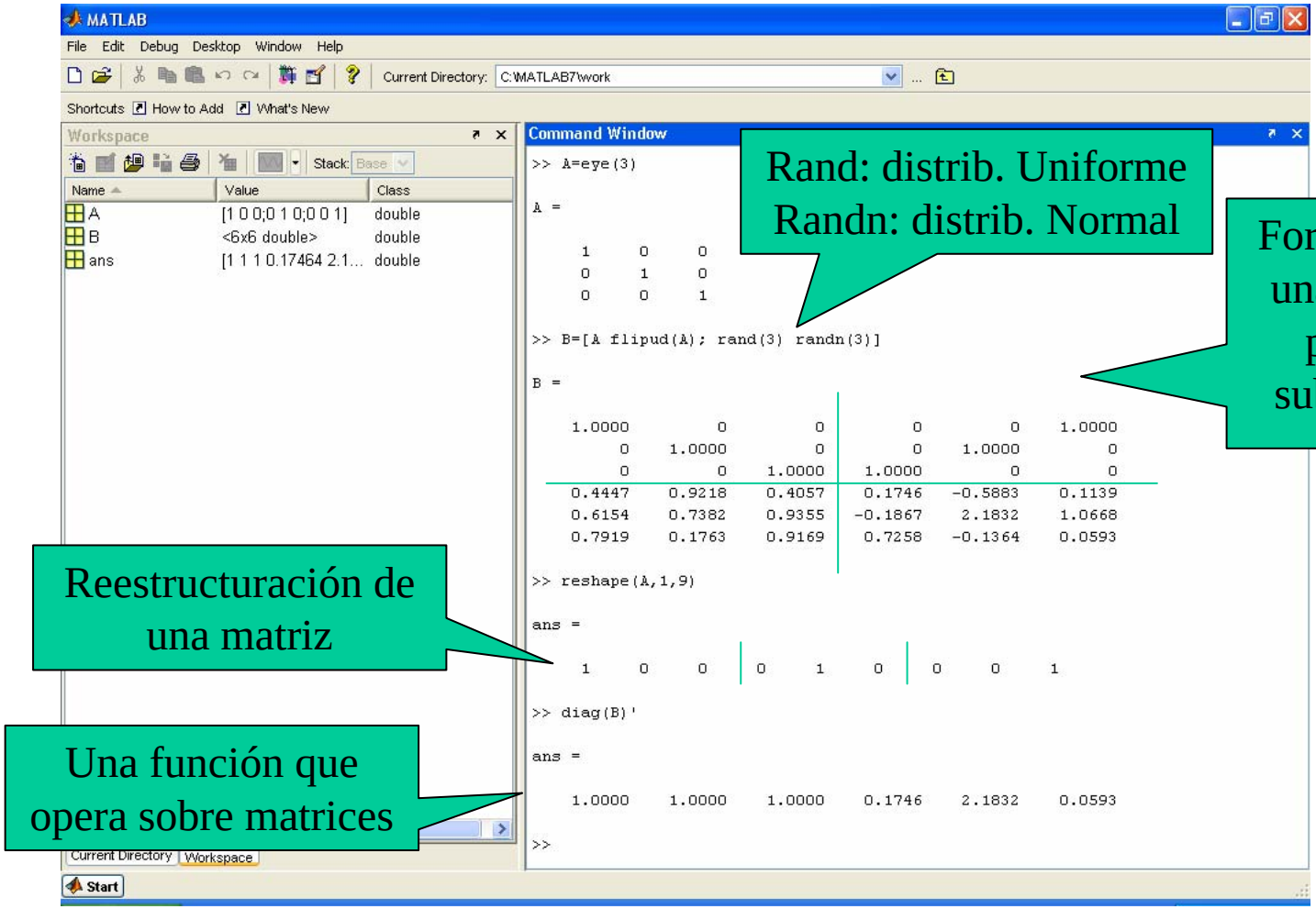
Definición de vectores y matrices



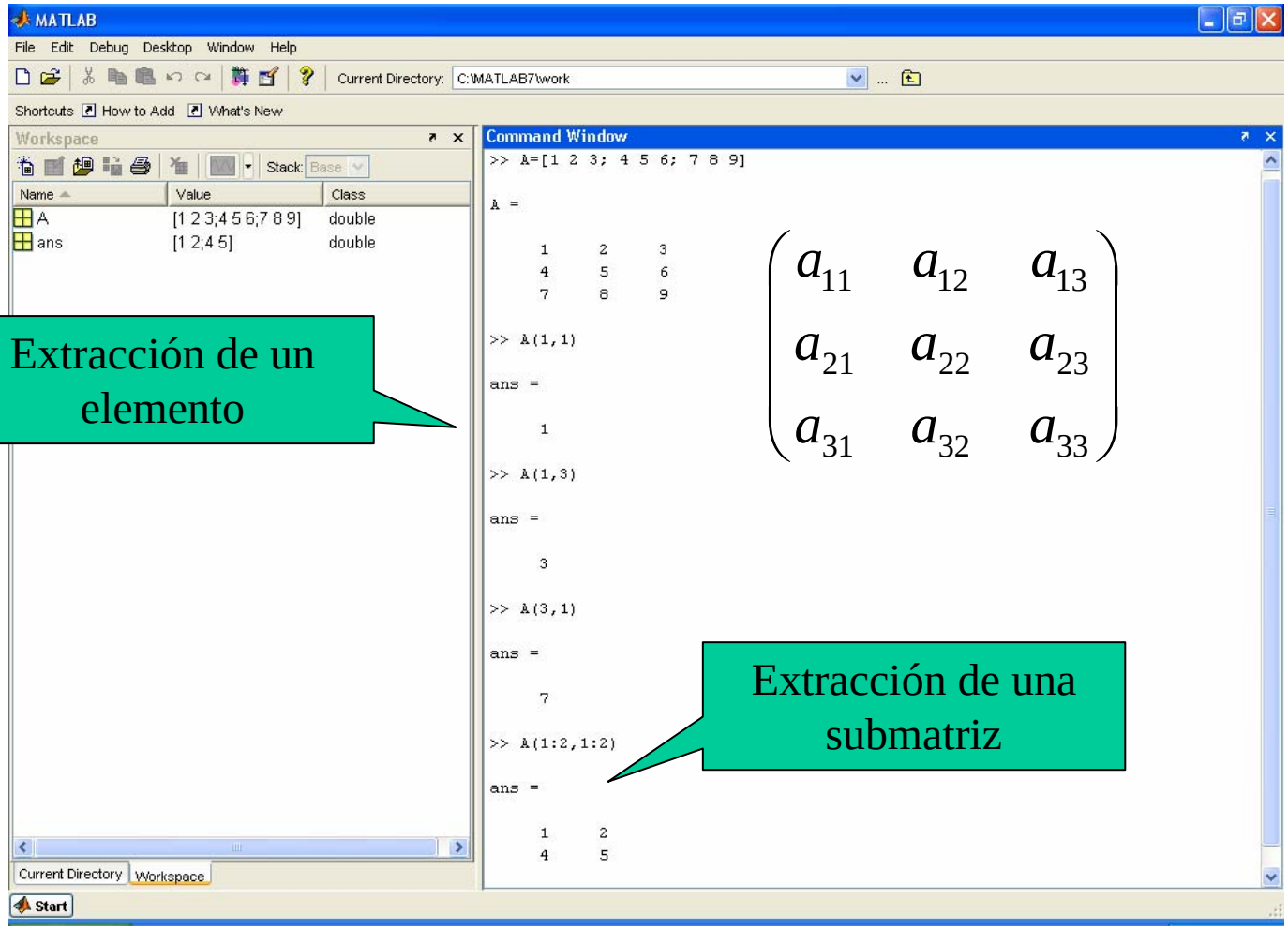
Definición de vectores y matrices



Definición de vectores y matrices



Indexación de matrices

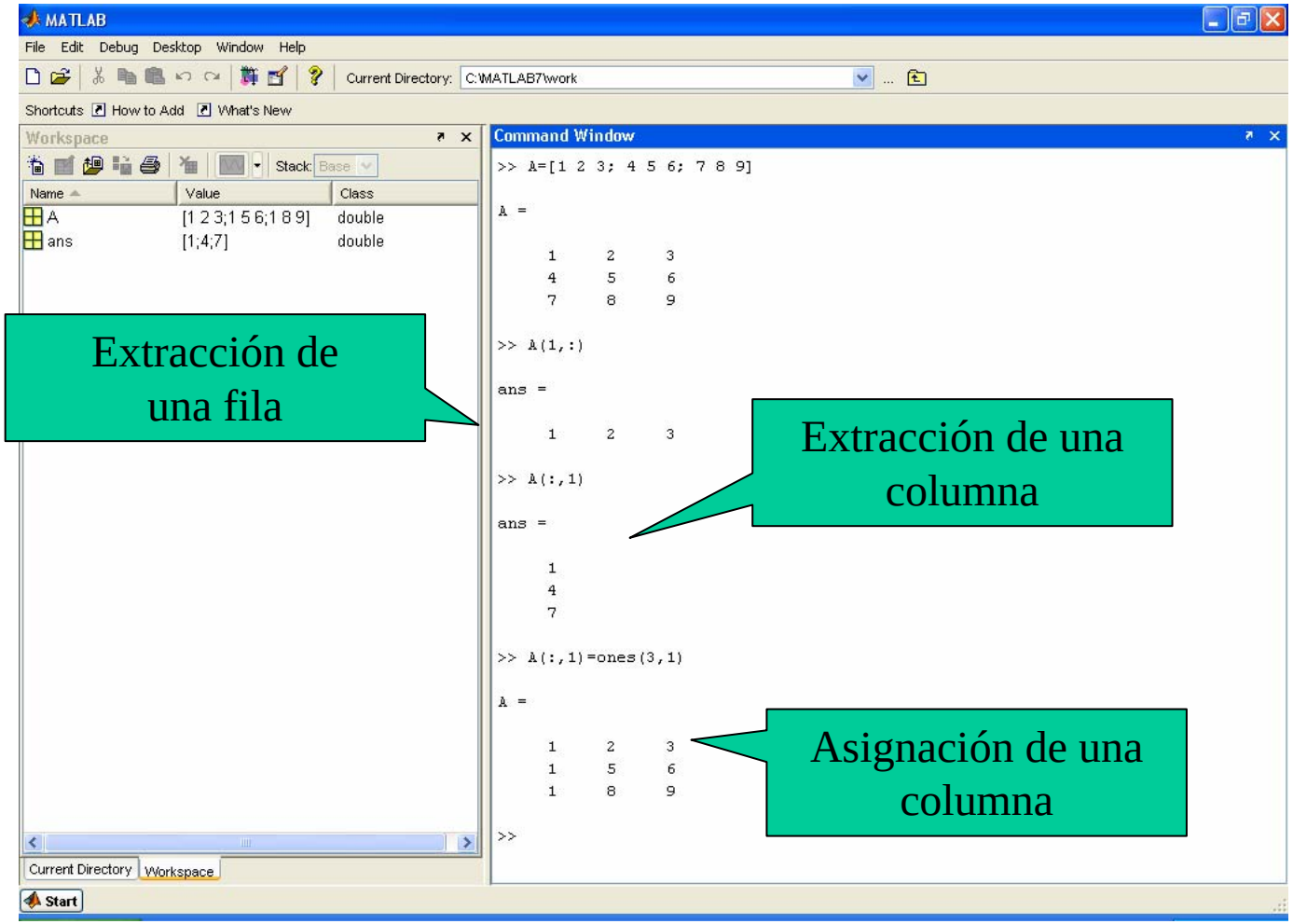


Extracción de un elemento

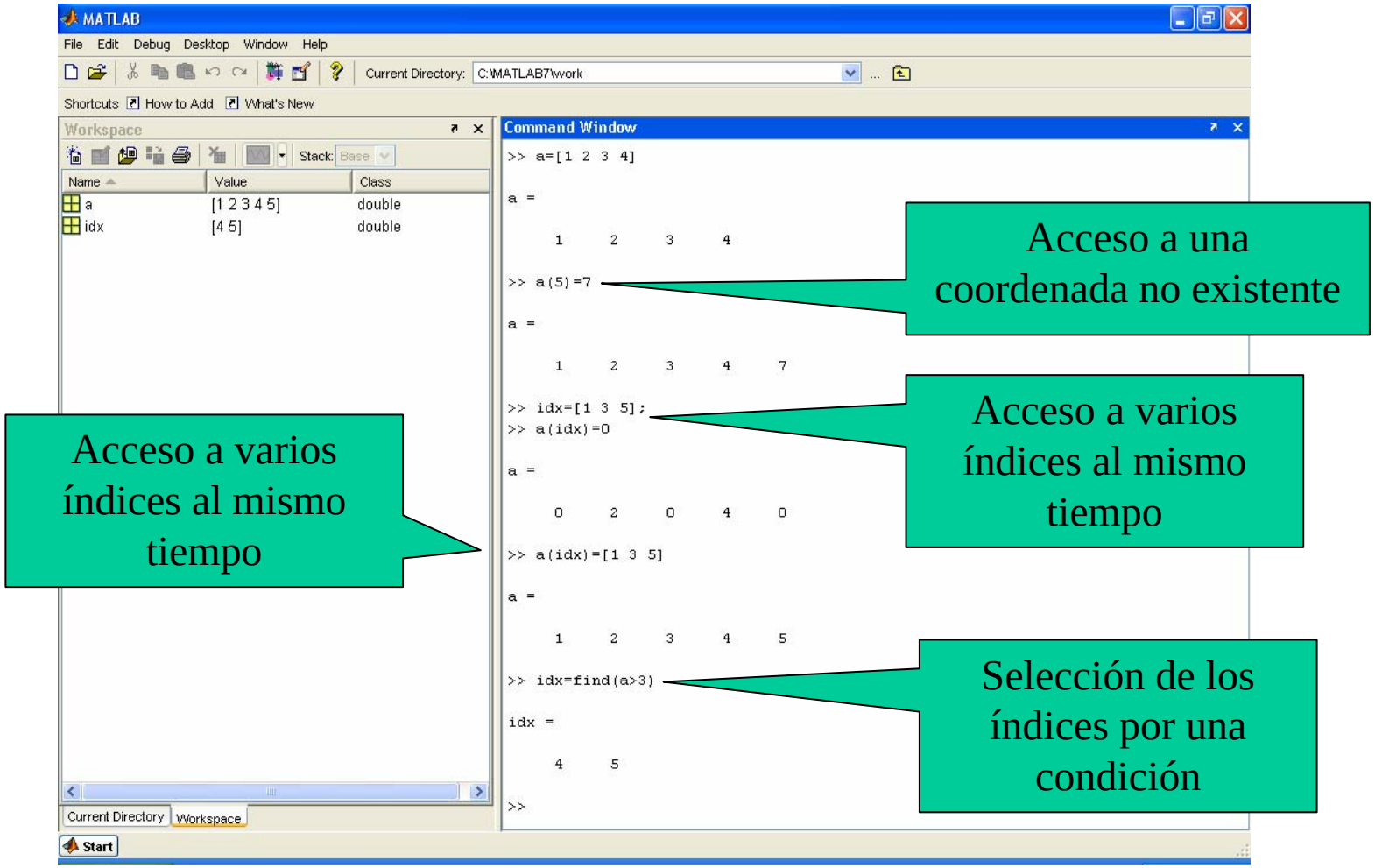
Extracción de una submatriz

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

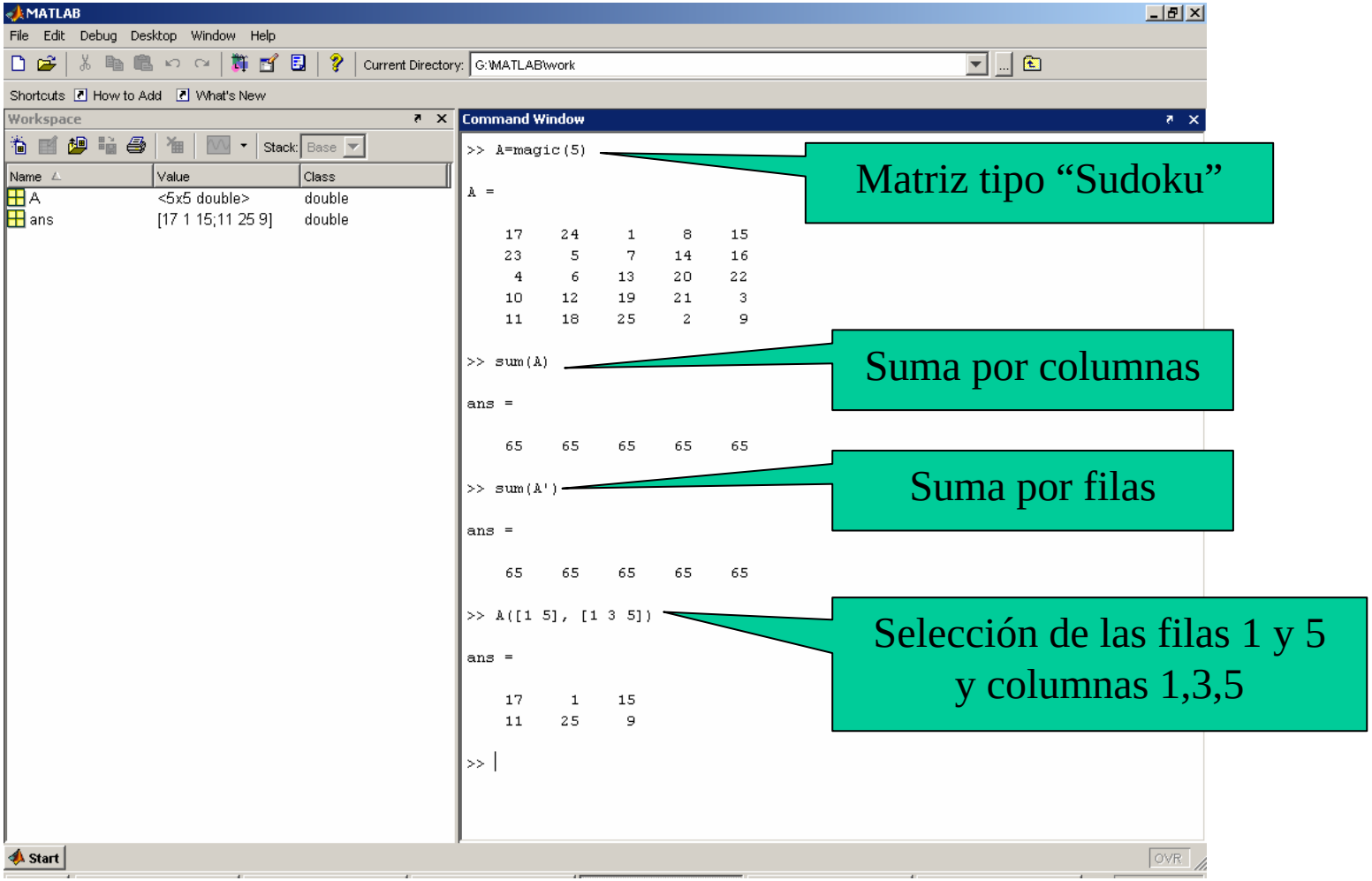
Indexación de matrices



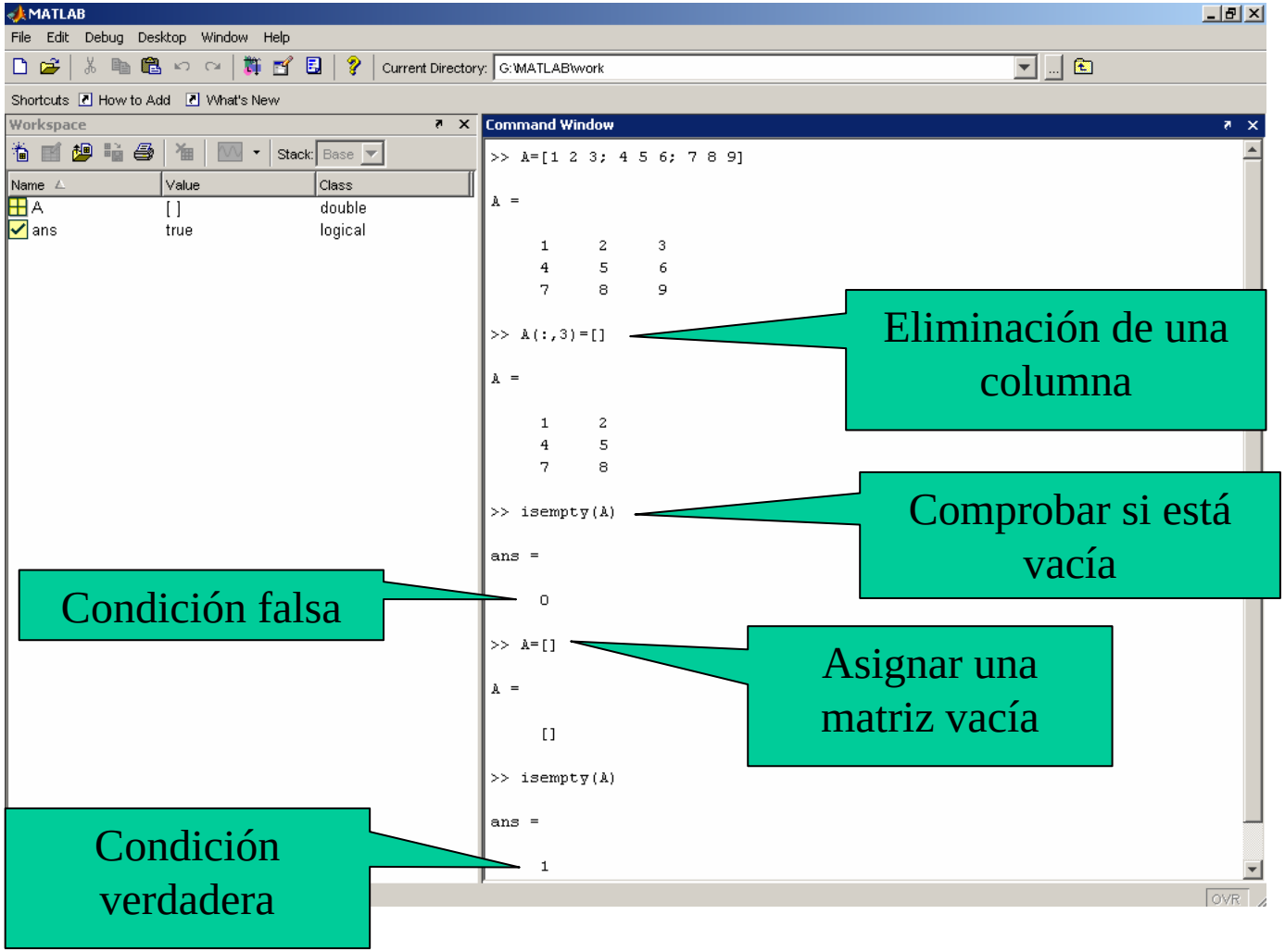
Indexación de matrices



Indexación de matrices



Indexación de matrices



Condiciones

MATLAB

File Edit Debug Desktop Window Help

Current Directory: G:\MATLAB\work

Shortcuts How to Add What's New

Workspace

Stack: Base

Name	Value	Class
A	<5x5 double>	double
ans	0.52	double

Command Window

```
>> A=randn(5)

A =

    0.8580   -0.3999    0.6686   -1.6041    0.5287
    1.2540    0.6900    1.1908    0.2573    0.2193
   -1.5937    0.8156   -1.2025   -1.0565   -0.9219
   -1.4410    0.7119   -0.0198    1.4151   -2.1707
    0.5711    1.2902   -0.1567   -0.8051   -0.0592

>> A>0

ans =

     1     0     1     0     1
     1     1     1     1     1
     0     1     0     0     0
     0     1     0     1     0
     1     1     0     0     0

>> size(A)

ans =

     5     5

>> sum(sum(A>0))/(size(A,1)*size(A,2))

ans =

    0.5200

>>
```

Start

OVR

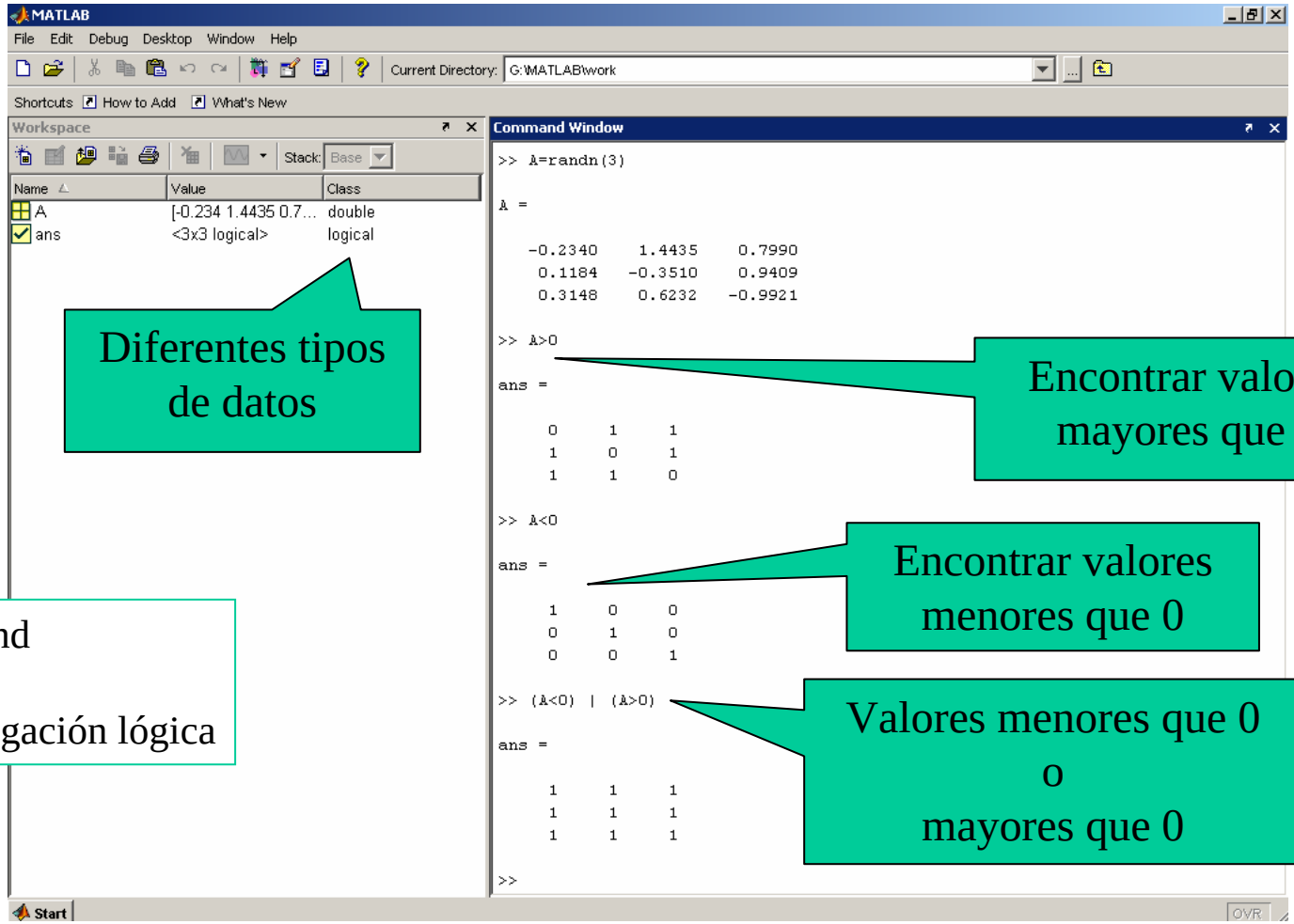
< menor que
> mayor que
<= menor o igual que
>= mayor o igual que
== igual que
~= distinto que (~=Alt-Graph 126)

Encontrar valores mayores que 0

Pedir tamaño de la matriz

Porcentaje de números mayores que 0

Condiciones



Diferentes tipos de datos

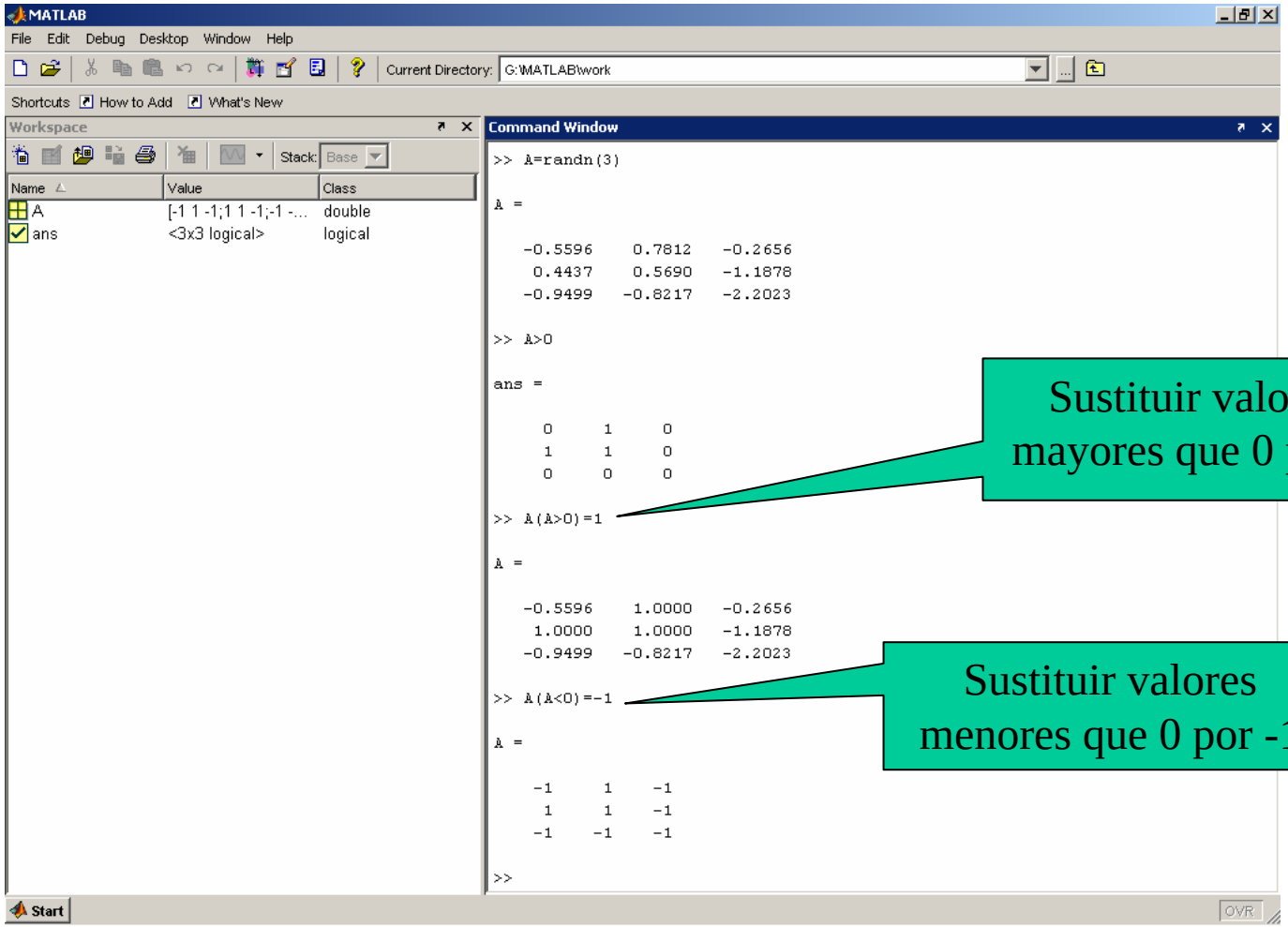
Encontrar valores mayores que 0

Encontrar valores menores que 0

Valores menores que 0
o
mayores que 0

& and
| or
~ negación lógica

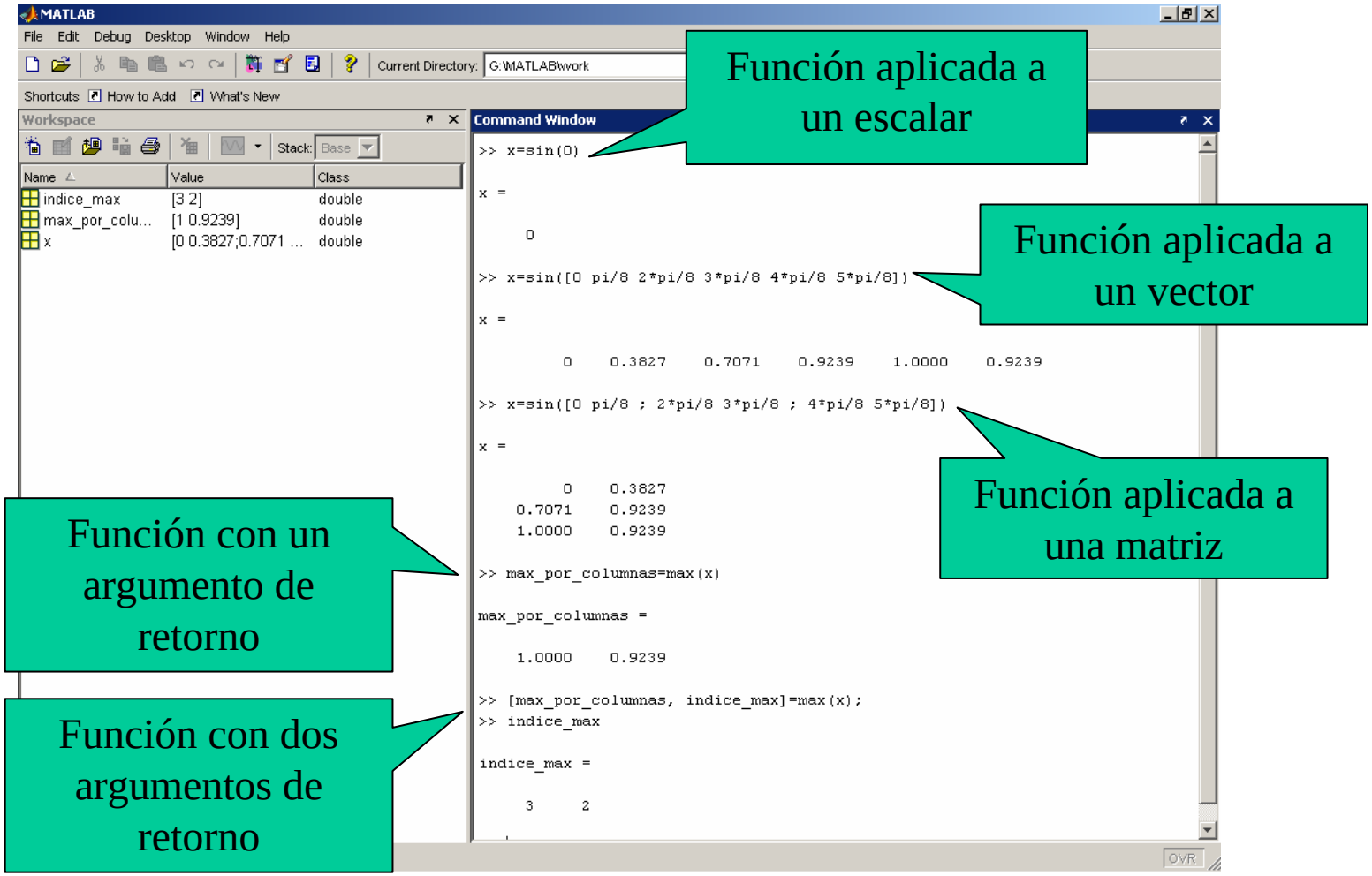
Condiciones e indexación



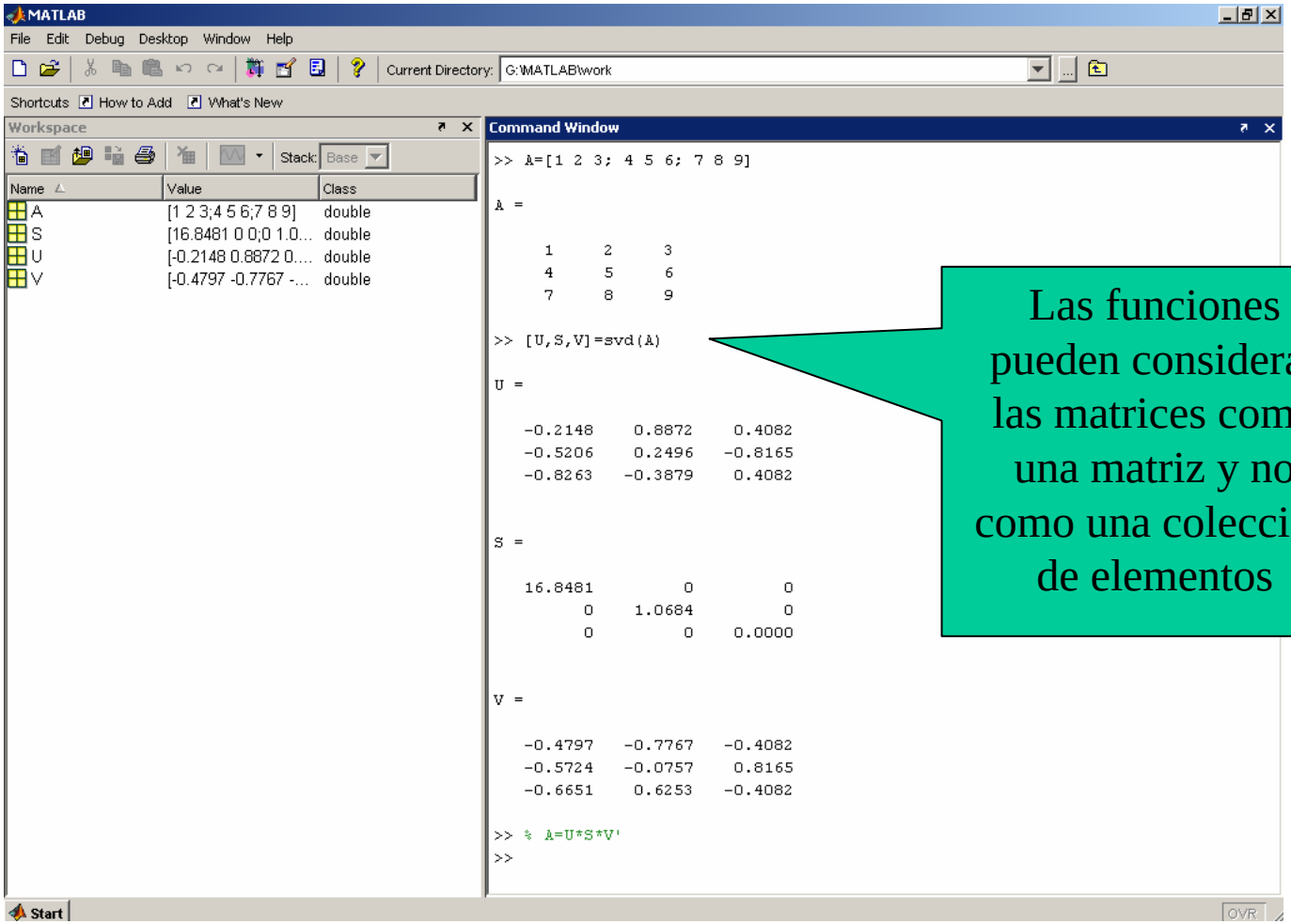
Sustituir valores mayores que 0 por 1

Sustituir valores menores que 0 por -1

Funciones


MATLAB
File Edit Debug Desktop Window Help
Current Directory: G:\MATLAB\work
Shortcuts How to Add What's New
Workspace
Stack: Base
Name Value Class
indice_max [3 2] double
max_por_colu... [1 0.9239] double
x [0 0.3827;0.7071 ... double
Command Window
>> x=sin(0)
x =
0
>> x=sin([0 pi/8 2*pi/8 3*pi/8 4*pi/8 5*pi/8])
x =
0 0.3827 0.7071 0.9239 1.0000 0.9239
>> x=sin([0 pi/8 ; 2*pi/8 3*pi/8 ; 4*pi/8 5*pi/8])
x =
0 0.3827
0.7071 0.9239
1.0000 0.9239
>> max_por_columnas=max(x)
max_por_columnas =
1.0000 0.9239
>> [max_por_columnas, indice_max]=max(x);
>> indice_max
indice_max =
3 2
OVR
Función aplicada a un escalar
Función aplicada a un vector
Función aplicada a una matriz
Función con un argumento de retorno
Función con dos argumentos de retorno

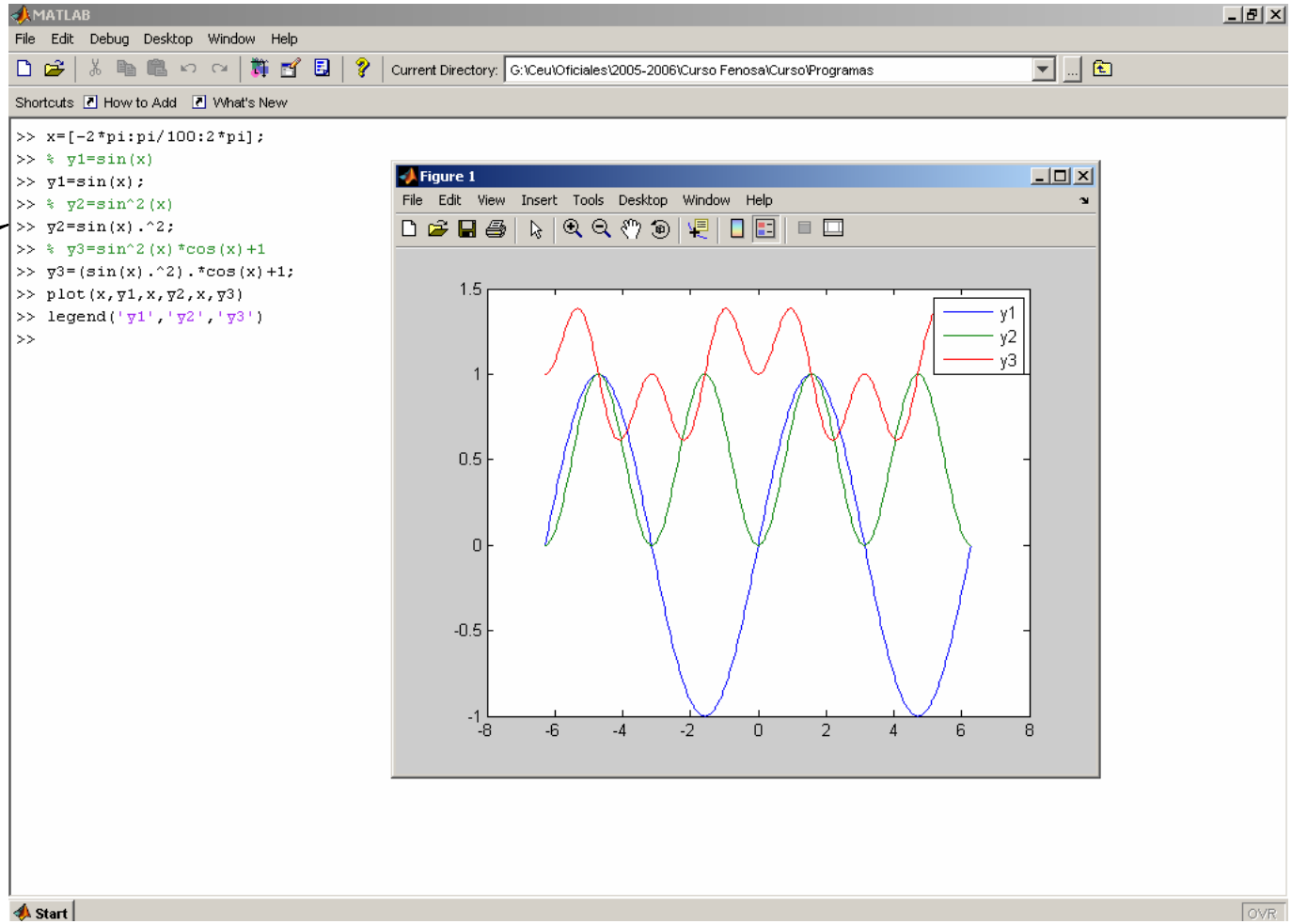
Funciones



Las funciones pueden considerar las matrices como una matriz y no como una colección de elementos

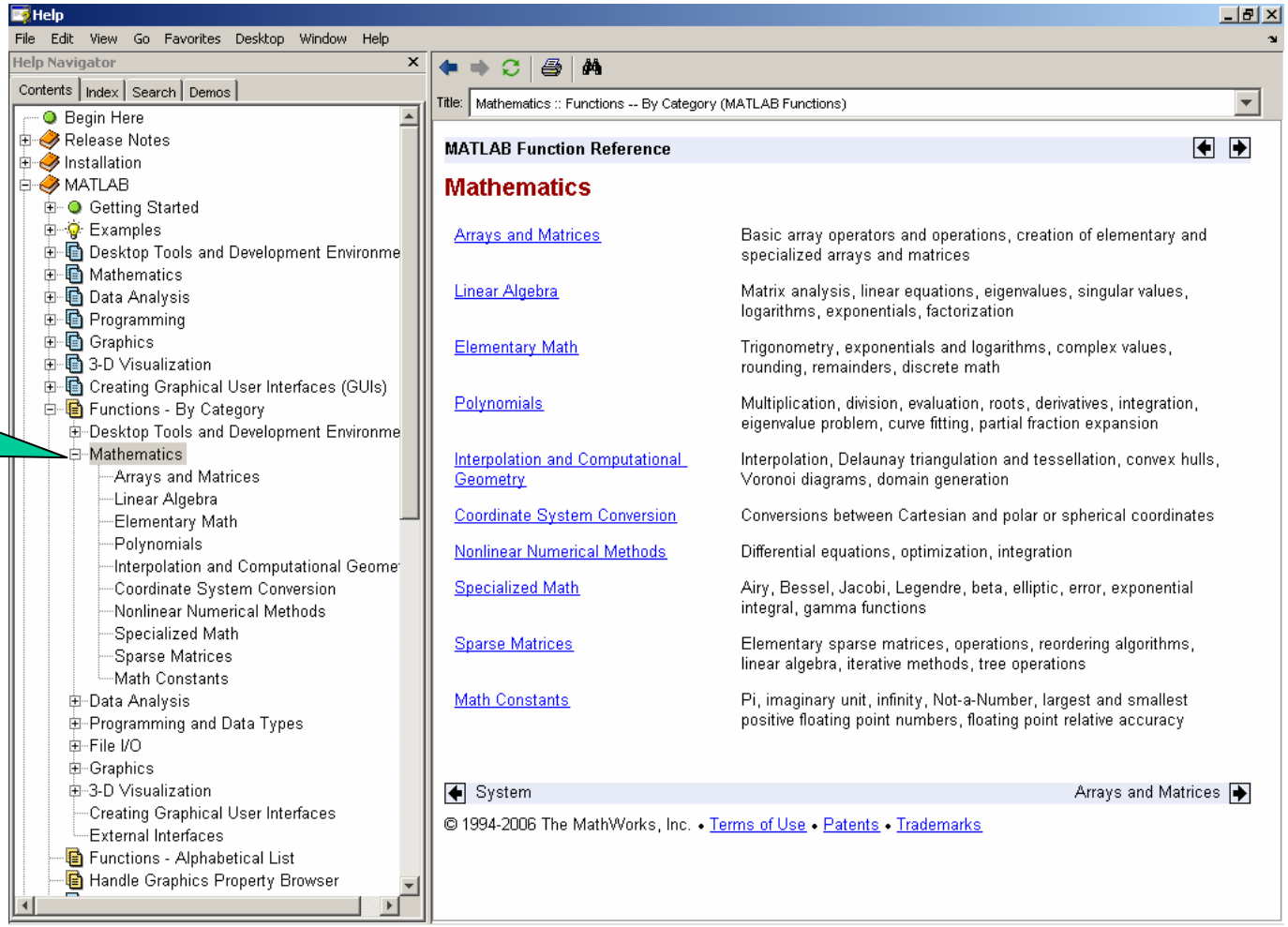
Funciones

Operaciones elemento a elemento

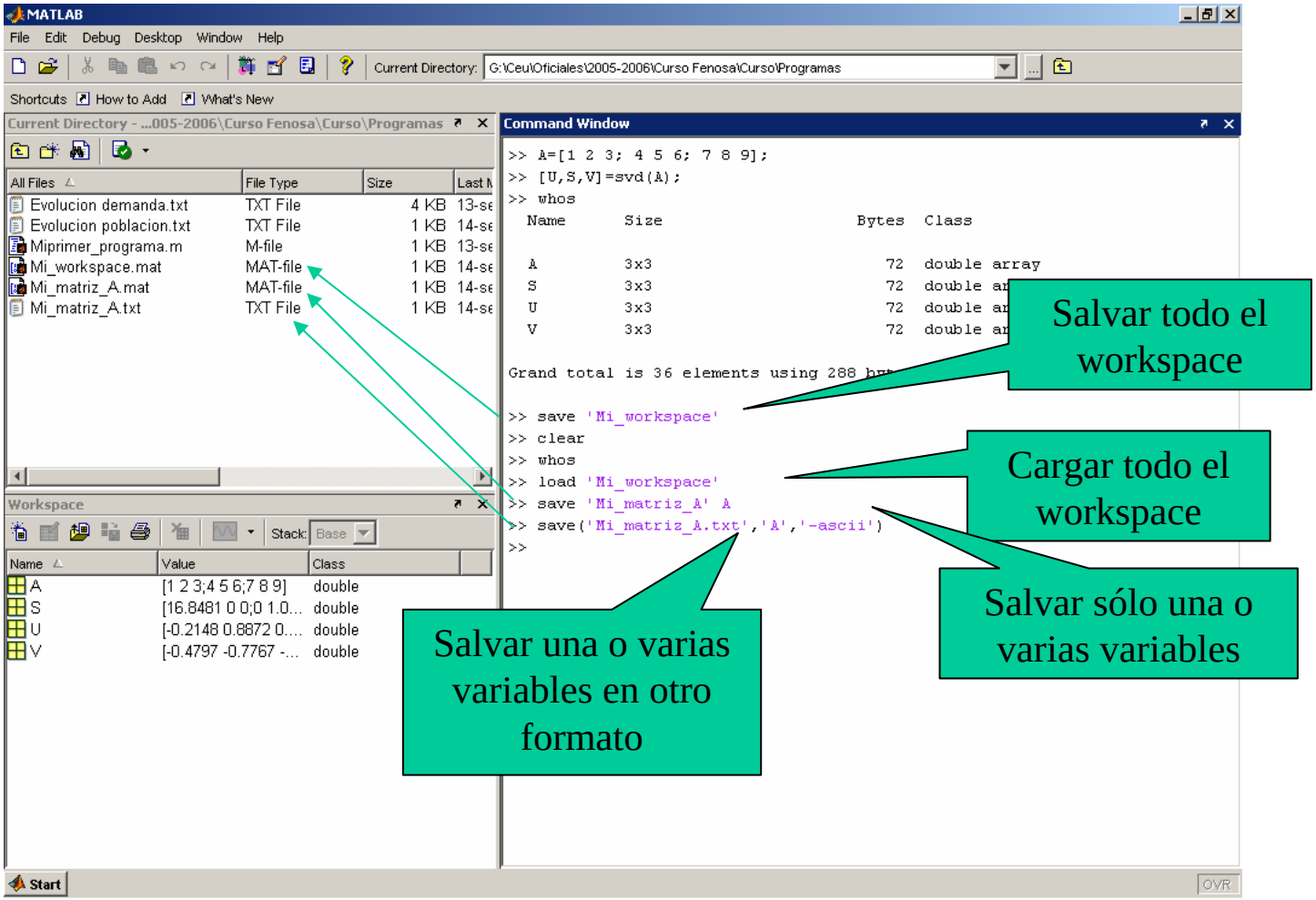


Funciones

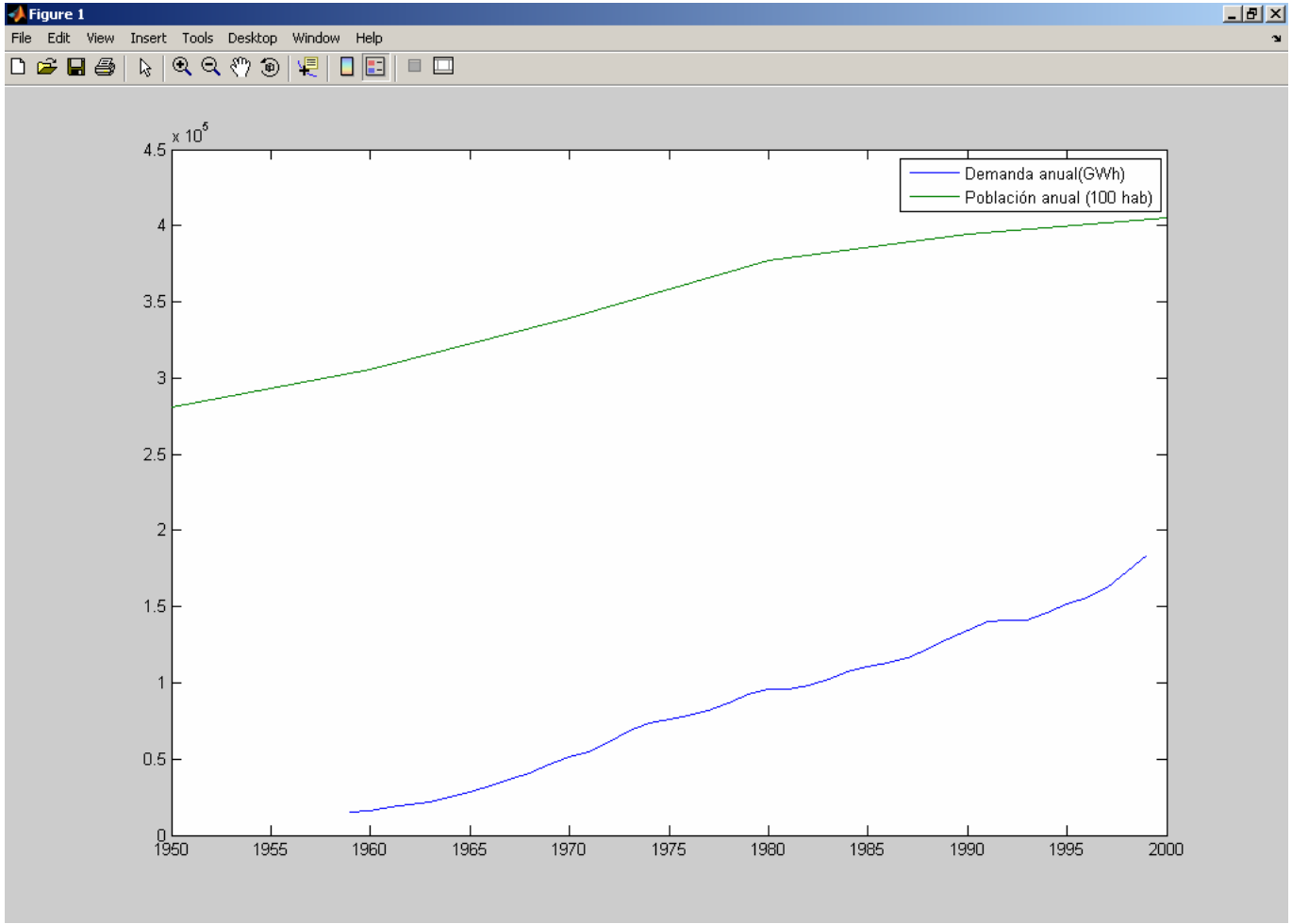
Explorar las funciones matemáticas



Entrada/Salida



Ejercicio final



Ejercicio final

1. Cargar los datos de evolución de la demanda eléctrica y la población española desde 1950 a 2000
   1. ‘Evolucion demanda.txt’ (Fuente: Red Eléctrica de España):
      1. Columna 1: Año
      2. Columna 2-13: Demanda mensual (GWh)
      3. Columna 14: Demanda anual (GWh)
   2. ‘Evolucion poblacion.txt’ (Fuente: Instituto Nacional de Estadística):
      1. Columna 1: Año
      2. Columna 2: Población (Miles de habitantes)
2. Mostrar en un mismo gráfico ambas tendencias de forma anual
   (Sugerencia: puede que haga falta escalar alguna de las series)
3. Poner ambas series en la misma base de tiempos (1959-1999)
   (Sugerencia: buscar en la ayuda las funciones de interpolación de MATLAB)

CEU
Universidad
San Pablo

CURSO de UTILIZACIÓN PRÁCTICA de MATLAB

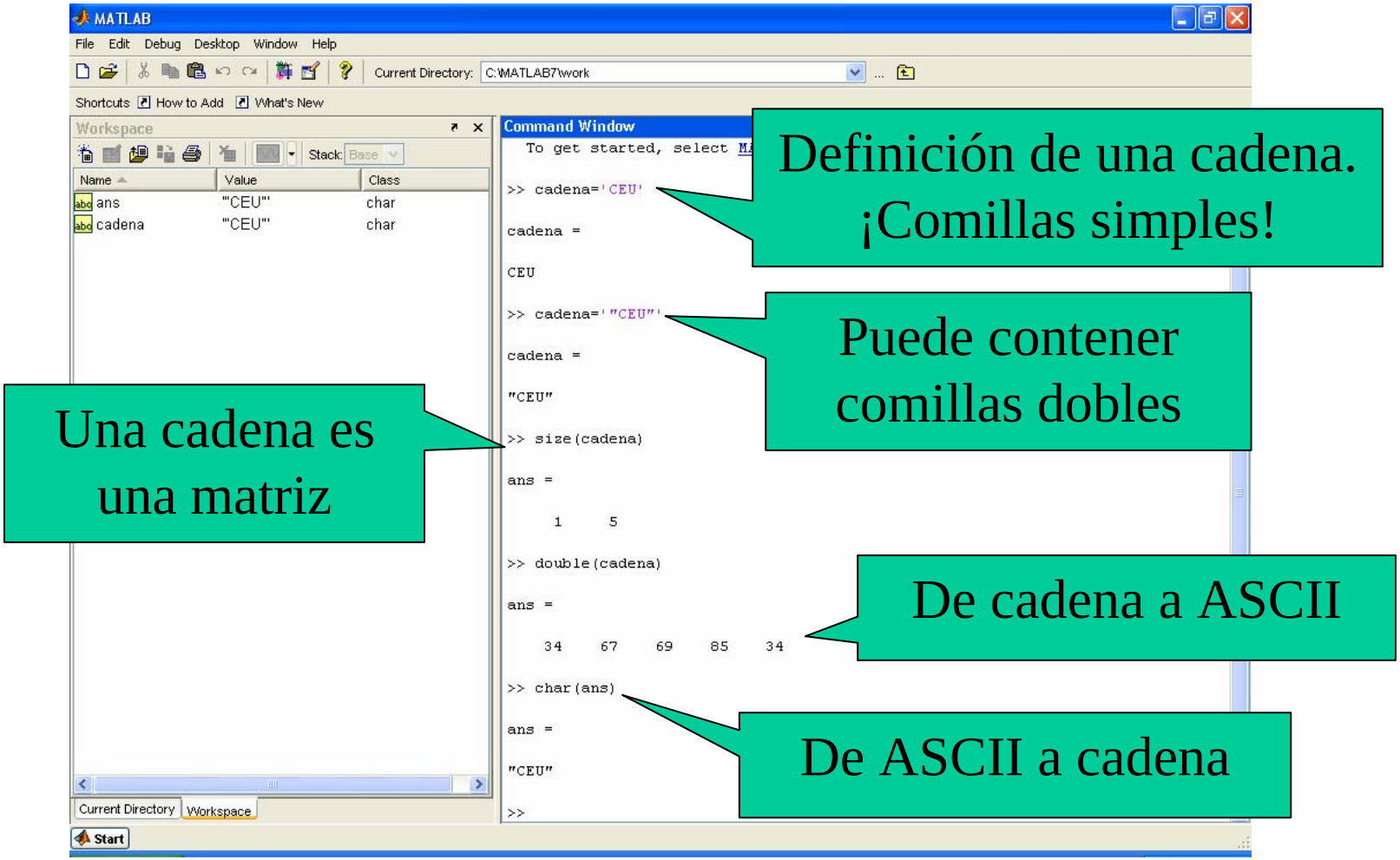
Sesión 2

Carlos Óscar Sánchez Sorzano, Ph.D.
Madrid, July 17th 2006

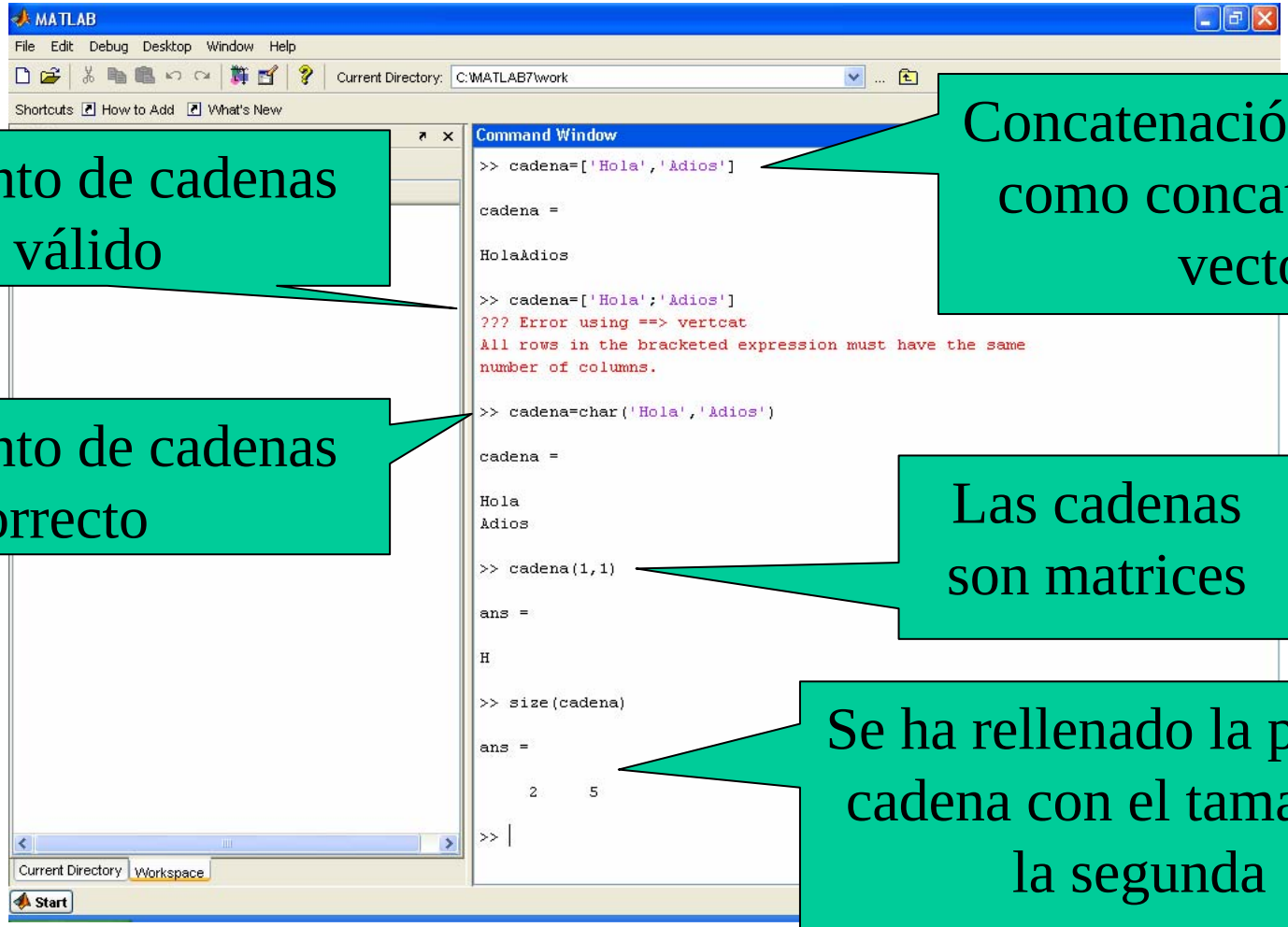
Cronograma del curso

- Día 1: Operaciones con matrices y vectores. Funciones de librería.
- **Día 2: Otros tipos de datos en MATLAB. Programación en MATLAB.**
- Día 3: Gráficos bidimensionales. Gráficos tridimensionales.
- Día 4: Interfaces de usuario en MATLAB. Generación de programas autónomos
- Día 5: Librerías de interés práctico
- Día 6: Interacción de MATLAB con Office y Visual Basic
- Día 7: Desarrollo de un proyecto

Tipos de datos: Cadenas



Tipos de datos: Cadenas



Apilamiento de cadenas no válido

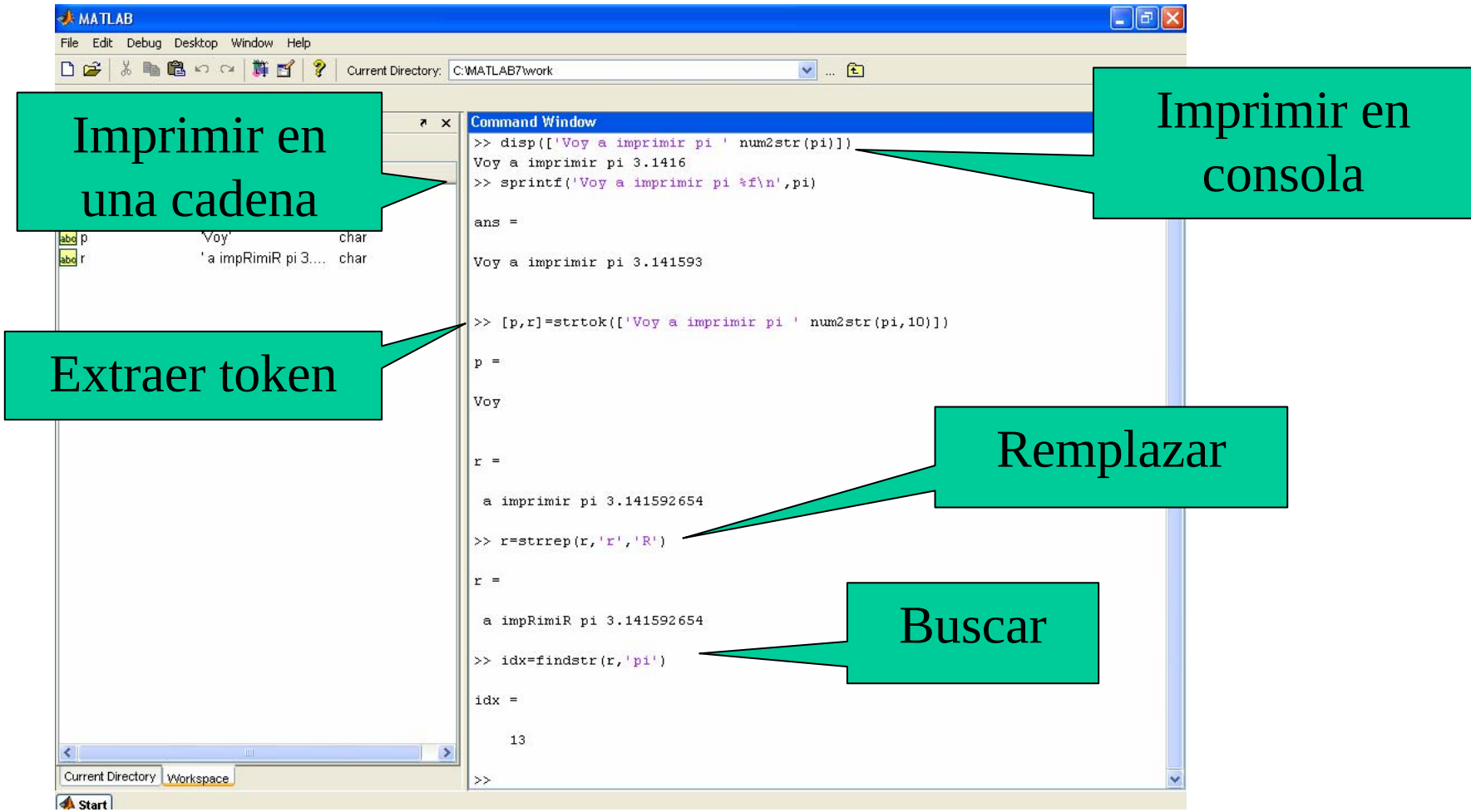
Concatenación de cadenas como concatenación de vectores

Apilamiento de cadenas correcto

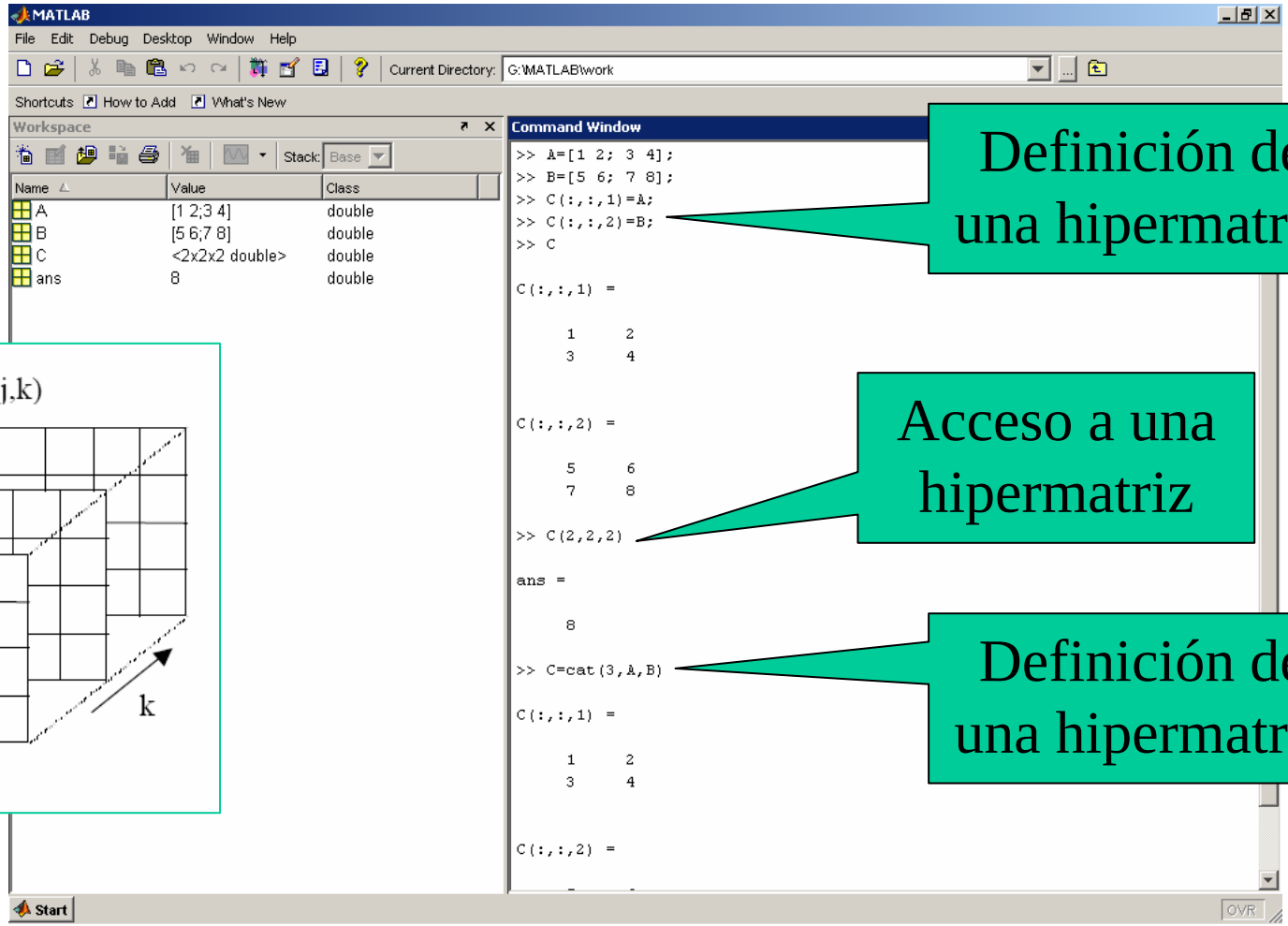
Las cadenas son matrices

Se ha rellenado la primera cadena con el tamaño de la segunda

Tipos de datos: Cadenas



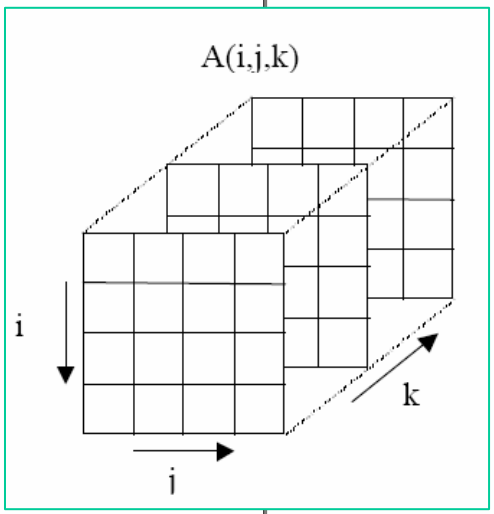
Tipos de datos: Hipermatrices



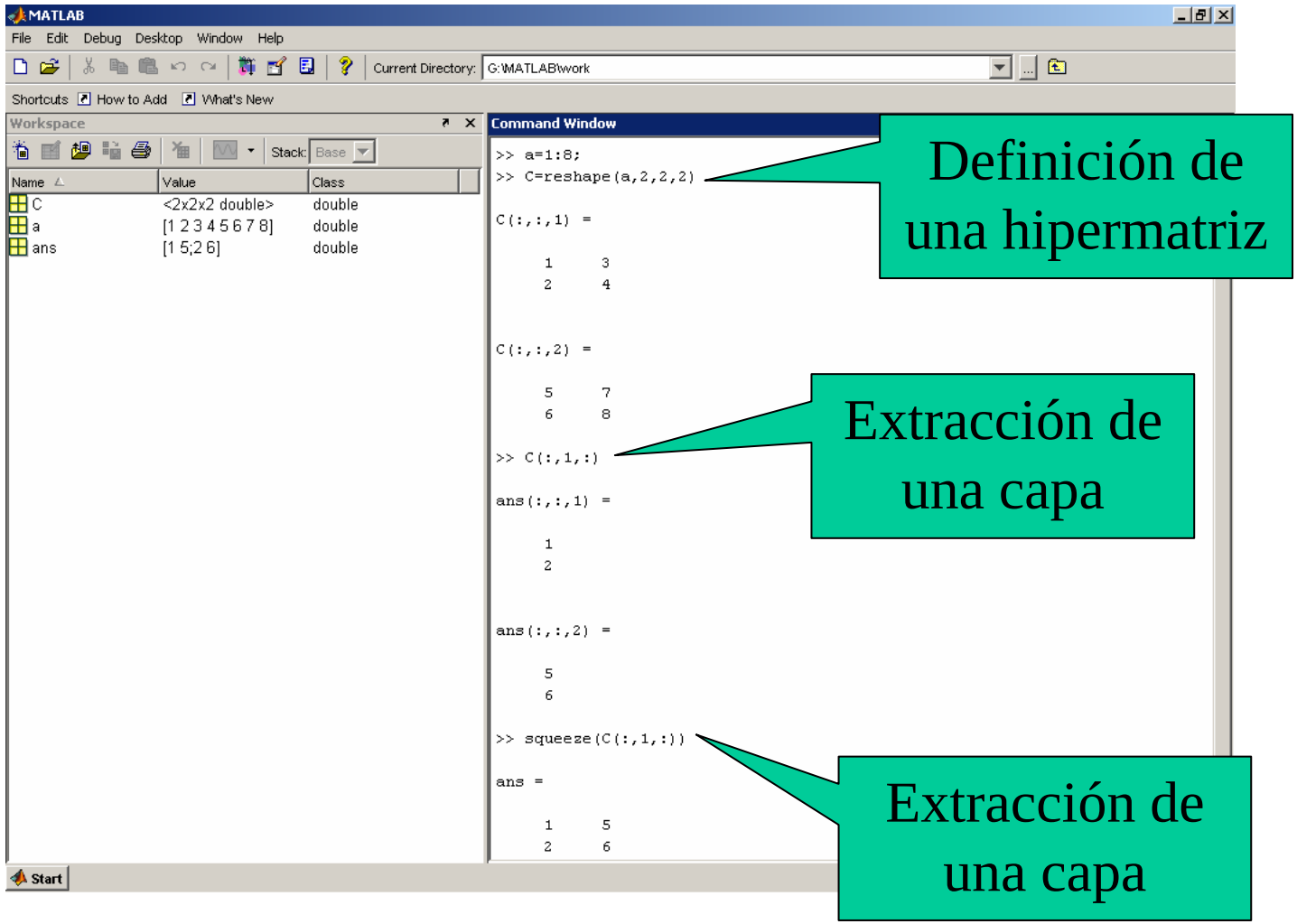
Definición de una hipermatriz

Acceso a una hipermatriz

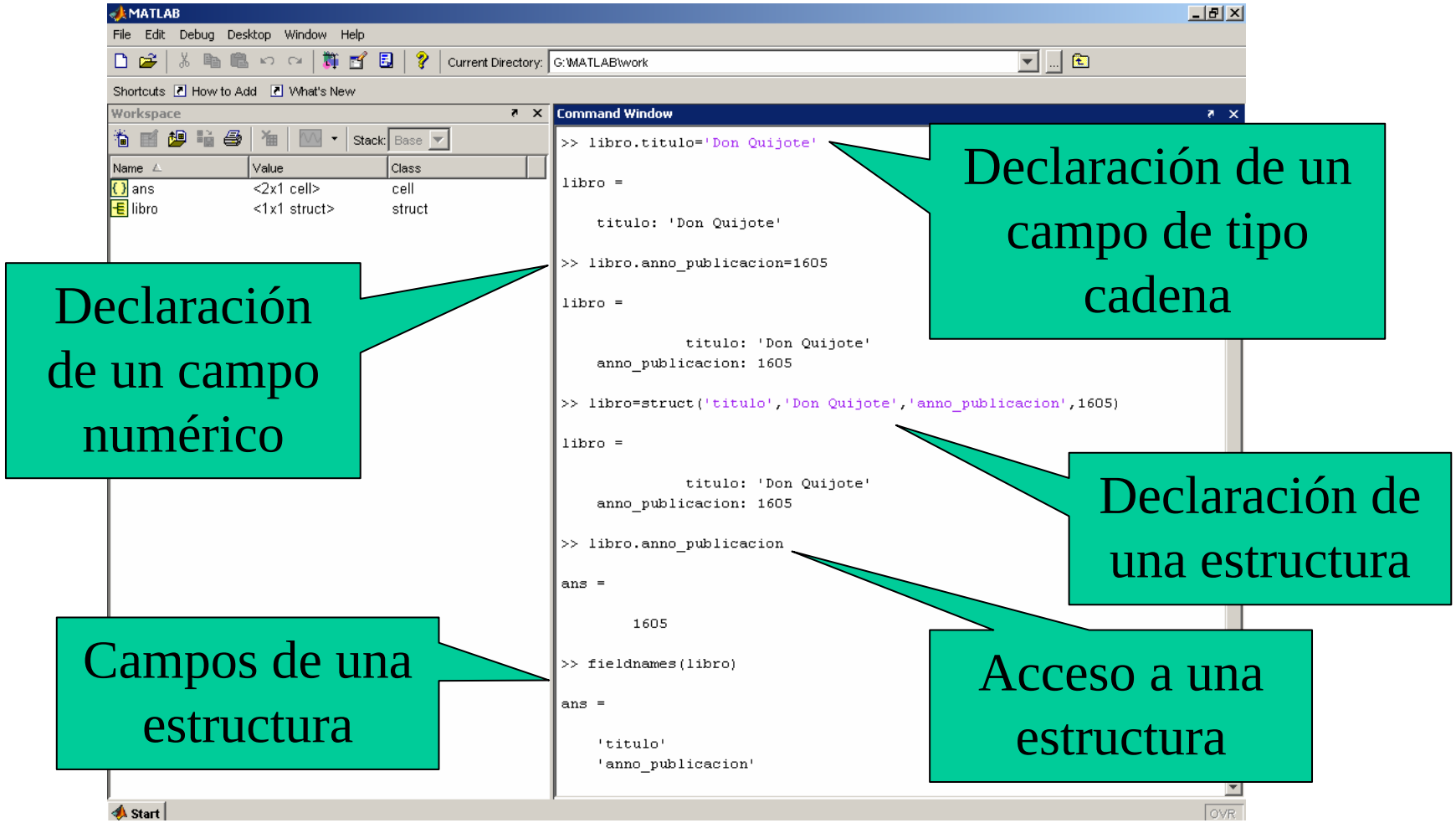
Definición de una hipermatriz



Tipos de datos: Hipermatrices



Tipos de datos: Estructuras



Tipos de datos: Estructuras

MATLAB

File Edit Debug Desktop Window Help

Current Directory: G:\MATLAB\work

Shortcuts How to Add What's New

Workspace

Stack: Base

Name	Value	Class
ans	<1x1 struct>	struct
biblioteca	<1x1 struct>	struct

Command Window

```
>> biblioteca.ubicacion='Mi casa';
>> biblioteca.libro(2)=struct('titulo','Don Quijote')

biblioteca =

    ubicacion: 'Mi casa'
        libro: [1x2 struct]

>> biblioteca.libro

ans =

1x2 struct array with fields:
    titulo

>> biblioteca.libro(2)

ans =

    titulo: 'Don Quijote'

>> biblioteca.libro(1)=struct('titulo','Novelas ejemplares')

biblioteca =

    ubicacion: 'Mi casa'
        libro: [1x2 struct]

>>
```

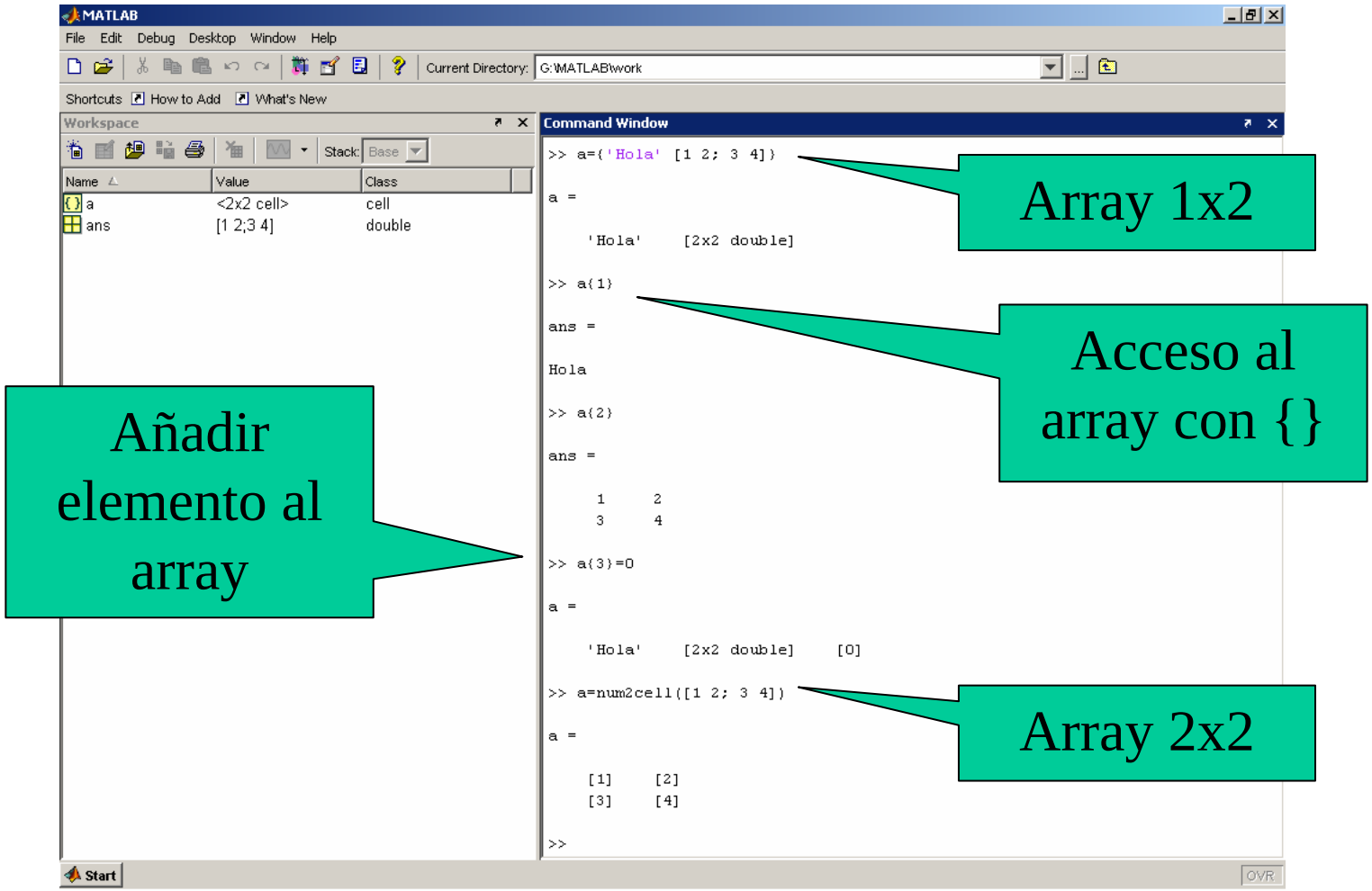
Start

Estructuras anidadas

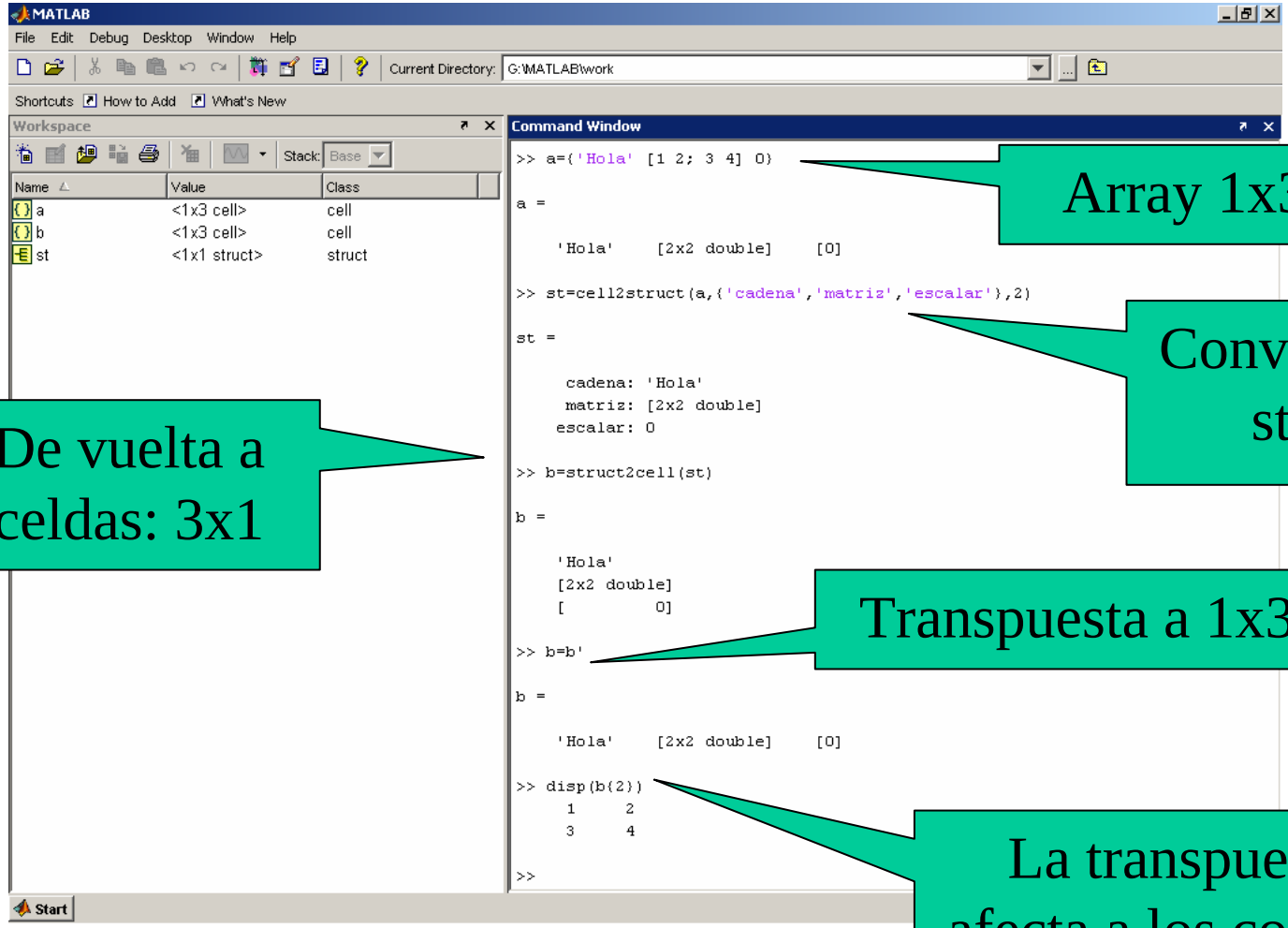
Array de estructuras con el primer elemento vacío

Completar el primer elemento del array

Tipos de datos: Arrays de celdas



Tipos de datos: Arrays de celdas y estructuras



Array 1x3

Conversión a struct

De vuelta a celdas: 3x1

Transpuesta a 1x3

La transpuesta no afecta a los contenidos

Programación: Control

```
for i=1:5
   for j=1:5
      A(i,j)=1/(i+j-1);
   end
end
```

```
A=rand(2,3)
for columna=A
   disp(columna)
end
```

```
A=zeros(3,4);
for i=1:size(A,1)
   for j=1:size(A,2)
      A(i,j)=abs(i-j);
   end
end
```

```
for x=1.0:-0.1:0.0
   disp(x)
   if x<0.5 break; end
end
```

```
for i=1:5
   for j=1:5
      if i==j
         A(i,j)=2;
      elseif abs(i-j)==1
         A(i,j)=-1;
      else
         A(i,j)=0;
      end
   end
end
```

```
a=[10 20 30 40];
for i=1:length(a)
   a(i)=a(i)/i;
end
a

a=[10 20 30 40]./[1:length(a)]
```

Programación: Control

```
method = 'Bilinear';

switch lower(method)
   case {'linear','bilinear'}
        disp('Method is linear')
   case 'cubic'
        disp('Method is cubic')
   case 'nearest'
        disp('Method is nearest')
   otherwise
        disp('Unknown method.')
end
```

```
% Precisión de la máquina
eps=1;
while (1+eps)>1
   eps=eps/2;
end
eps=eps*2
```

```
% Contar las líneas de código de magic.m
fid=fopen('magic.m', 'r');
count=0;
while ~feof(fid)
   linea = fgetl(fid);
   if isempty(linea) | strncmp(linea,'%',1)
       continue
   end
   count=count+1;
end
fclose(fid);
disp(sprintf('%d líneas', count));
```

Programación: Funciones

```
function [mean,stdev]=stat(x)
   % Esta función calcula la media y la desviación típica de x
   % Uso: [mean,stdev]=stat(x)

   % El fichero debe llamarse stat.m

   n=length(x);
   mean=sum(x)/n;
   stdev=sqrt(sum((x-mean).^2/n));
end

help stat
[media,desviacion]=stat([1 2 3 4 5]);
```

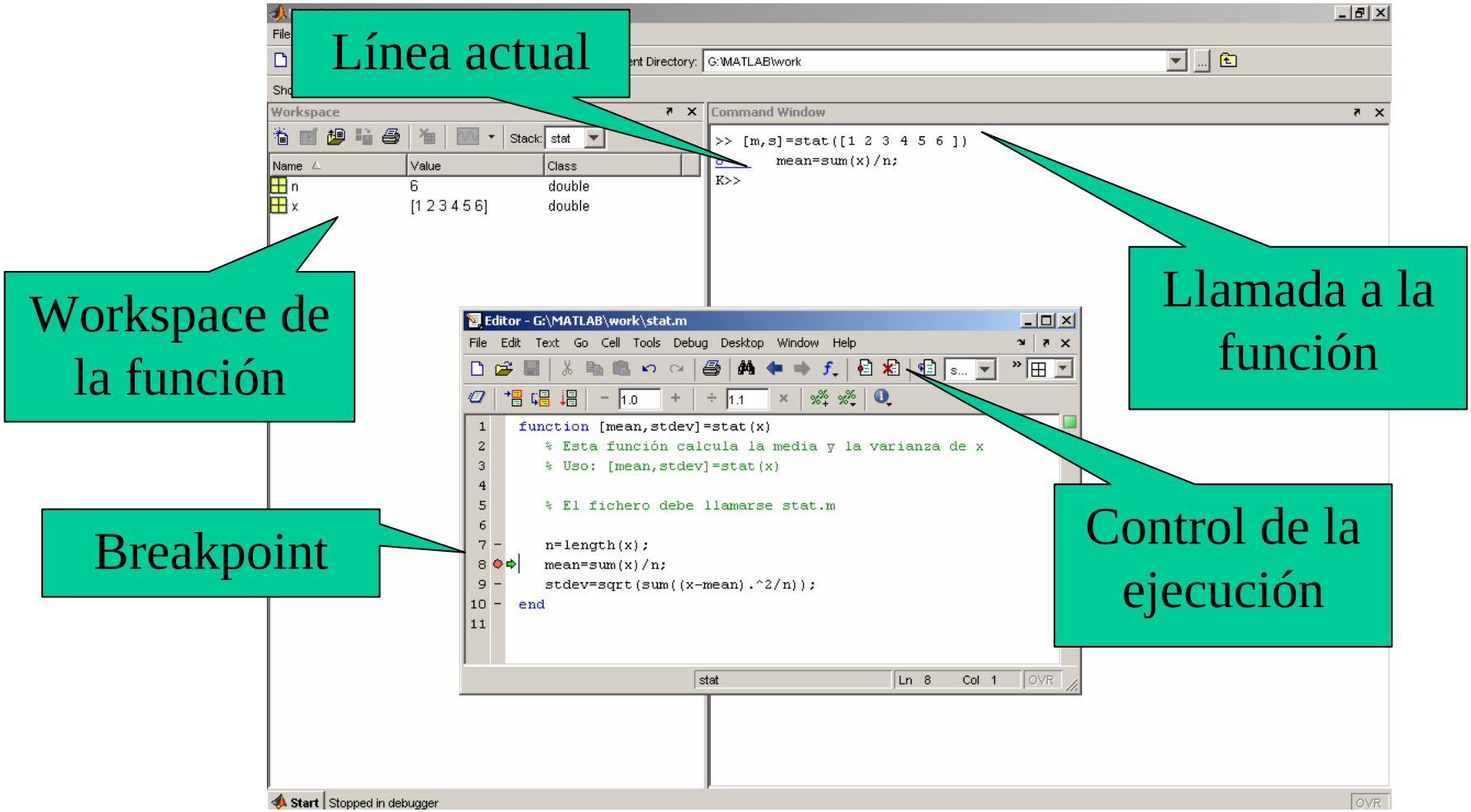
```
function [mean,stdev]=stat(x)
   n=length(x);
   mean=media(x);
   stdev=sqrt(sum((x-mean).^2/n));

function mean=media(x)
   mean=sum(x)/length(x);
```

Subfunción

```
function [mean,stdev]=stat(x)
   n=length(x);
   mean=media();
   stdev=sqrt(sum((x-mean).^2/n));

   function mean=media()
      mean=sum(x)/length(x);
   end
end
```

Función anidada

Entorno de MATLAB: Debugger



Programación: Funciones

```
function varargout=atan3(varargin)
   if nargin==1
      rad=atan(varargin{1});
   elseif nargin==2
      rad=atan2(varargin{1},varargin{2});
   else
      disp('Error: más de dos argumentos')
      return
   end
   varargout{1}=rad;
   varargout{2}=rad*180/pi;
end
```

```
function [x0, y0] = myplot(x, y, npts, angle, subdiv)
% MYPLOT  Plot a function.
% MYPLOT(x, y, npts, angle, subdiv)
%     The first two input arguments are
%     required; the other three have default values.
 ...
if nargin < 5, subdiv = 20; end
if nargin < 4, angle = 10; end
if nargin < 3, npts = 25; end
 ...
if nargout == 0
     plot(x, y)
else
     x0 = x;
     y0 = y;
end
```

Programación: funciones vs scripts (batches)

houdini

```
% File: houdini.m
m = magic(4); % Assign 4x4 magic square to m.
t = m .^ 3;   % Cube each element of m.
disp(t);      % Display the value of t.
```

Workspace: m,t

houdini(4)

```
% File: houdini.m
function houdini(sz)
m = magic(sz); % Assign 4x4 magic square to m.
t = m .^ 3;    % Cube each element of m.
disp(t);       % Display the value of t.
```

Workspace:

[m,t]=houdini(4)

```
% File: houdini.m
function [m,t]=houdini(sz)
m = magic(sz); % Assign 4x4 magic square to m.
t = m .^ 3;    % Cube each element of m.
disp(t);       % Display the value of t.
```

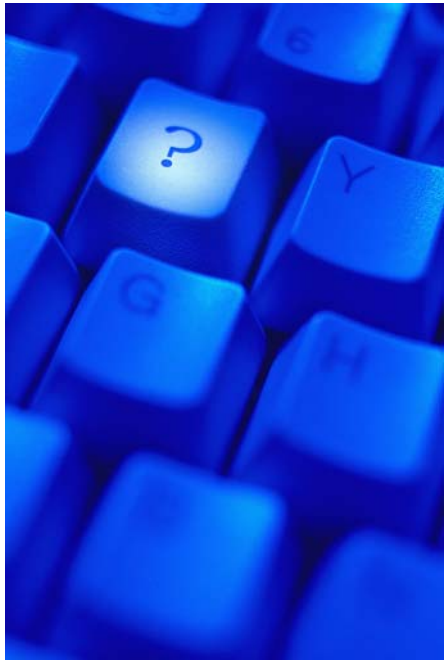
Workspace: m,t

Programación: Control

```
function matrixMultiply(A, B)
try
   X = A * B
catch
   disp '** Error multiplying A * B'
end

A = [1  2  3; 6  7  2; 0  1  5];
B = [9  5  6; 0  4  9];
matrixMultiply(A, B)
```

Programación: Entrada/Salida

- Consola/Teclado:
  - input/disp
  - echo, diary
- Cadena
  - sscanf,strread/sprintf
- Fichero:
  - load/save
  - dlmread/dlmwrite
  - tblread/tblwrite
  - xlsread/xlswrite
  - fopen, fread, fgetl, textscan/fwrite, fclose
  - importdata
  - uiimport
- Clipboard:
  - clipboard



Programación: Ejecución por celdas

Ejecución por celdas

Editor - G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas\SolucionDia1_celdas.m

File Edit Text Go Cell Tools Debug Desktop Window Help

This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.

```
%% Esto es un ejemplo de cómo se programa por celdas
% Únicamente se puede hacer con scripts
%
% El objetivo de este ejemplo es representar los datos
% de demanda eléctrica entre 1959 y 1999

%% Limpiemos todos los datos anteriores
clear   % Limpiar el workspace
clc     % Limpiar la consola

%% Cargar la evolución de la demanda
load('Evolucion demanda.txt')
agnos_demanda=Evolucion_demanda(:,1);
demanda_anual=Evolucion_demanda(:,14);

%% Cargar la evolución de la población
load('Evolucion poblacion.txt')
agnos_poblacion=Evolucion_poblacion(:,1);
poblacion_anual=Evolucion_poblacion(:,2);

%% Mostrar ambas tendencias
plot(agnos_demanda, demanda_anual, agnos_poblacion, poblacion_anual*10);
legend('Demanda anual(GWh)','Población anual (100 hab)')

%% Interpolar la población anualmente entre 1959-1999
poblacion_interpolada=interp1(agnos_poblacion,poblacion_anual,...
    agnos_demanda,'spline');
figure
plot(agnos_demanda, demanda_anual, agnos_demanda, poblacion_interpolada*10);
legend('Demanda anual(GWh)','Población anual (100 hab)')
```

Untitled | SolucionDia1_celda... | SolucionDia1.m

script Ln 13 Col 14 OVR

Programación: Documentación

- MiToolBox (Directorio)
  - contents.m
  - mifuncion1.m
  - mifuncion2.m
  - …

```
help MiToolBox
```

```
% Esta toolbox sirve para ...
%
% Familia de funciones 1
%    mifuncion1   -  Sirve para ...
%    mifuncion2   -  Sirve para ...
...
```

```
help mifuncion1
```

```
function y=mifuncion1(x)
% Esta función no hace nada
% Uso: y=mifuncion1(x)
% Entradas:
%     x   : vector de entrada para
%           no hacer nada con él
% Salidas:
%     y   : vector de salida
y=x
```

Programación: Publicación HTML de scripts

Esto es un ejemplo de cómo se programa por celdas

File Edit View Go Debug Desktop Window Help

Location: G:/Ceu/Oficiales/2005-2006/Curso Fenosa/Curso/Programas/html/SolucionDia1_celdas.html

Esto es un ejemplo de cómo se programa por celdas

Únicamente se puede hacer con scripts

El objetivo de este ejemplo es representar los datos de demanda eléctrica entre 1959 y 1999

Contents

Limpiemos todos los datos anteriores

```
clear   % Limpiar el workspace
clc     % Limpiar la consola
```

Cargar la evolución de la demanda

```
load('Evolucion demanda.txt')
agnos_demanda=Evolucion_demanda(:,1);
demanda_anual=Evolucion_demanda(:,14);
```

Cargar la evolución de la población

```
load('Evolucion poblacion.txt')
agnos_poblacion=Evolucion_poblacion(:,1);
poblacion_anual=Evolucion_poblacion(:,2);
```

Mostrar ambas tendencias

```
plot(agnos_demanda, demanda_anual, agnos_poblacion, poblacion_anual*10);
legend('Demanda anual(GWh)','Población anual (100 hab)')
```

Programación: Publicación HTML de scripts

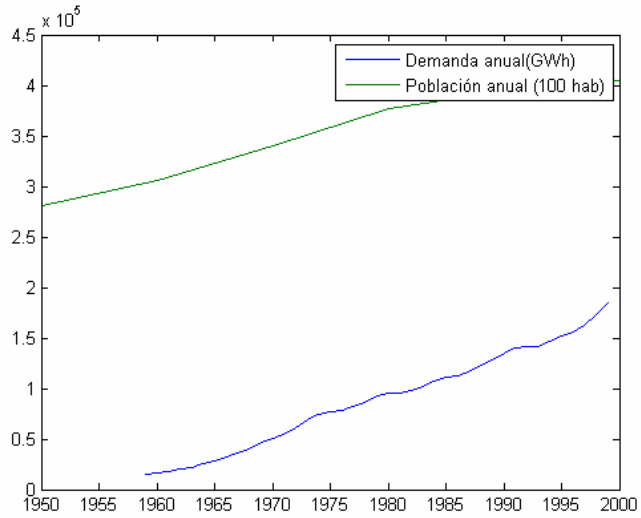
Esto es un ejemplo de cómo se programa por celdas

File Edit View Go Debug Desktop Window Help

Location: G:/Ceu/Oficiales/2005-2006/Curso Fenosa/Curso/Programas/html/SolucionDia1_celdas.html

Mostrar ambas tendencias

```
plot(agnos_demanda, demanda_anual, agnos_poblacion, poblacion_anual*10);
legend('Demanda anual(GWh)','Población anual (100 hab)')
```



Interpolar la población anualmente entre 1959-1999

```
poblacion_interpolada=interp1(agnos_poblacion,poblacion_anual,...
    agnos_demanda,'spline');
figure
```

Ejercicio final 1: Entrada/Salida

```
# Año  Ene   Feb   Mar   Abr   May   Jun   Jul   Ago   Sep   Oct   Nov   Dic  Total
1959  1345  1246  1252  1254  1188  1202  1243  1202  1192  1270  1312  1361  15067

# Década de los 60
1960  1369  1331  1351  1252  1321  1275  1325  1322  1336  1393  1456  1610  16341
1961  1622  1422  1528  1433  1488  1418  1467  1506  1514  1573  1653  1711  18335
1962  1741  1639  1755  1557  1572  1585  1653  1653  1642  1730  1828  1902  20257
...
```

- Ejercicio:
  - Leer el fichero “Evolucion demanda comentada.txt” separando los años, los datos mensuales y los datos totales.
  - Pedir al usuario el mes y los años que desea representar
  - Representar los datos solicitados

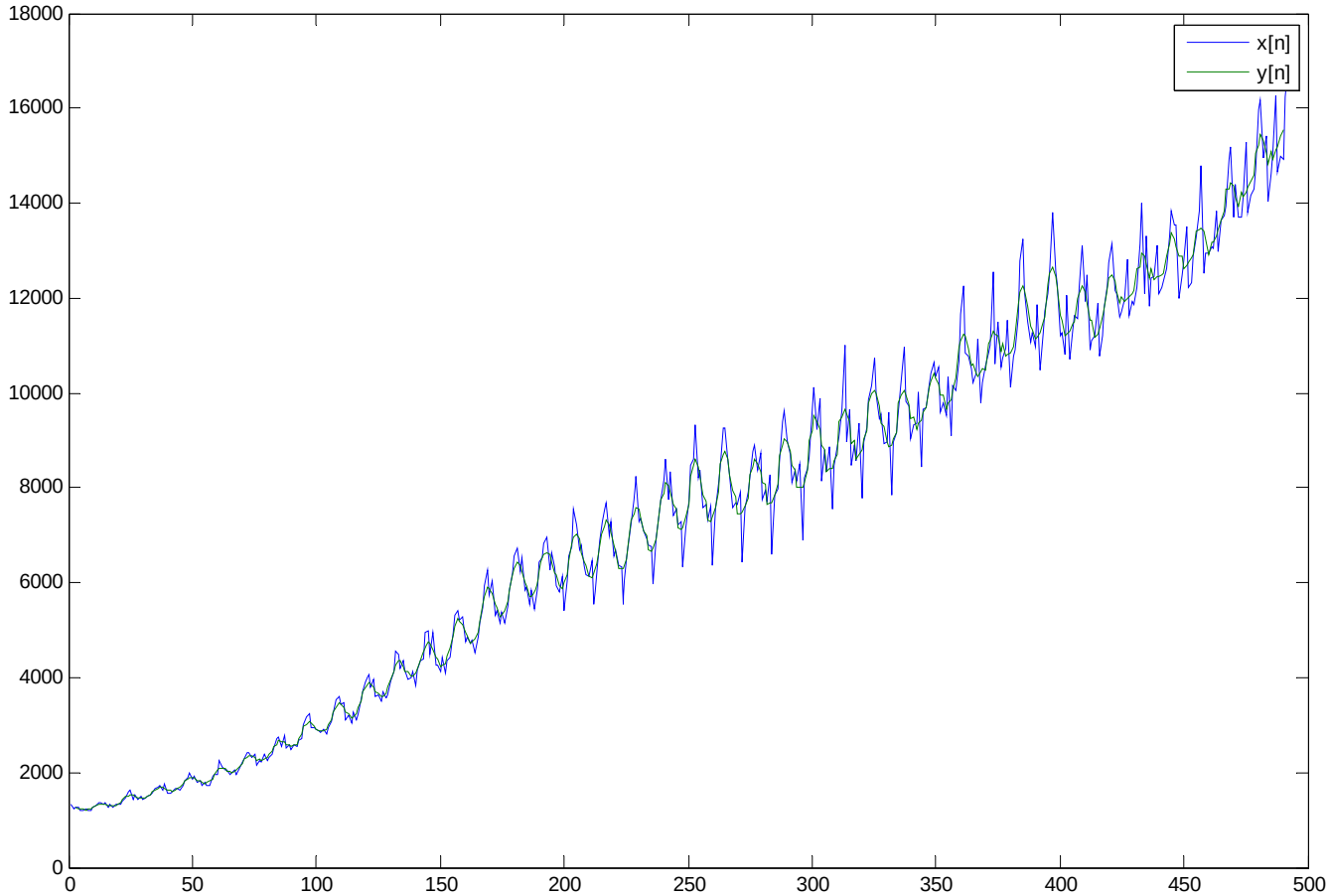
Ejercicio final 2: Cálculo científico

- Ejercicio:
  - Leer el fichero “Evolucion demanda comentada.txt” por medio de una función que recibe el nombre de fichero como parámetro
  - Denominemos x[n] a los datos mensuales
    - n=1 para enero de 1959
    - n=2 para febrero de 1959
    - …
    - n=12 para diciembre de 1959
    - n=13 para enero de 1960
    - etc.
  - Suavizar los datos leídos mediante la fórmula

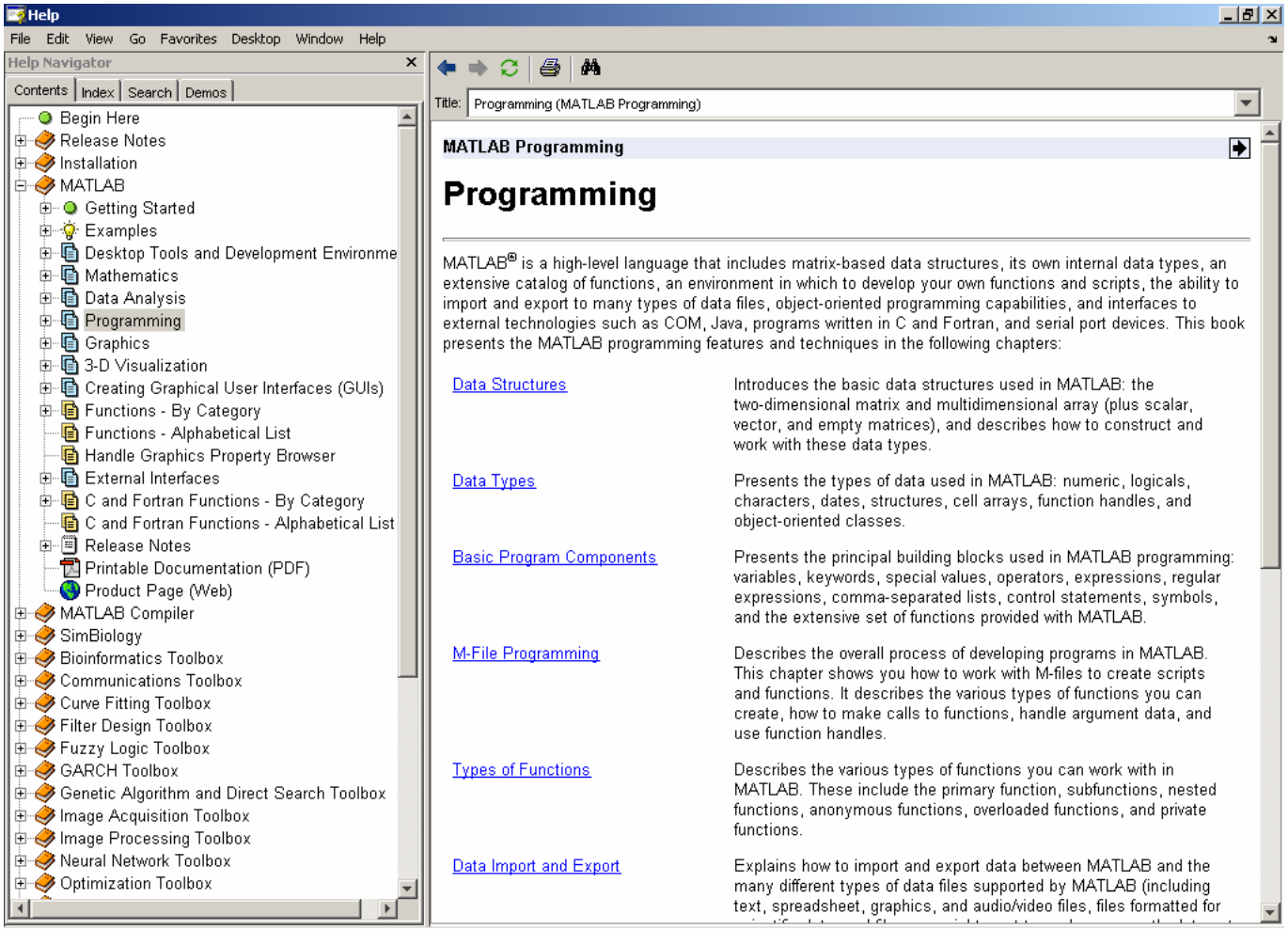
$$y[n] = \frac{x[n-2] + x[n-1] + x[n] + x[n+1] + x[n+2]}{5}$$

  - Representar la demanda y la demanda suavizada en un mismo gráfico

Ejercicio 2: Cálculo científico



Y si sobra tiempo …



CEU
Universidad
San Pablo

CURSO de UTILIZACIÓN PRÁCTICA de MATLAB

Sesión 3

Carlos Óscar Sánchez Sorzano, Ph.D.
Madrid, July 17th 2006

Cronograma del curso

- Día 1: Operaciones con matrices y vectores. Funciones de librería.
- Día 2: Otros tipos de datos en MATLAB. Programación en MATLAB.
- **Día 3: Gráficos bidimensionales. Gráficos tridimensionales.**
- Día 4: Interfaces de usuario en MATLAB. Generación de programas autónomos
- Día 5: Librerías de interés práctico
- Día 6: Interacción de MATLAB con Office y Visual Basic
- Día 7: Desarrollo de un proyecto

Gráficos bidimensionales

Etiquetas de los ejes

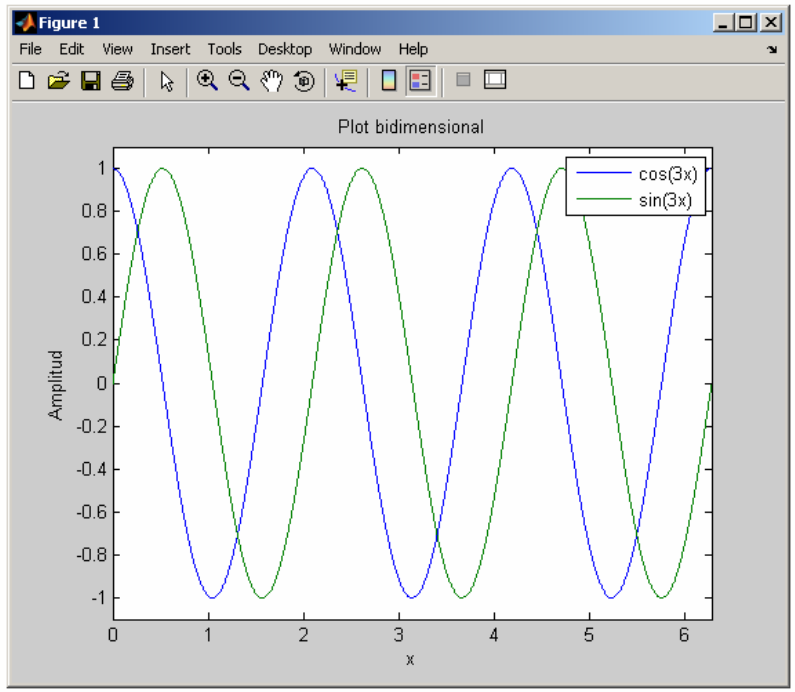
Leyenda

Título

Ejes

Plot

```
>> x=[0:0.001:2*pi];
>> plot(x,cos(3*x),x,sin(3*x))
>> xlabel('x');
>> legend('cos(3x)','sin(3x)')
>> ylabel('Amplitud')
>> title('Plot bidimensional')
>> axis([0 2*pi -1.1 1.1])
>>
```

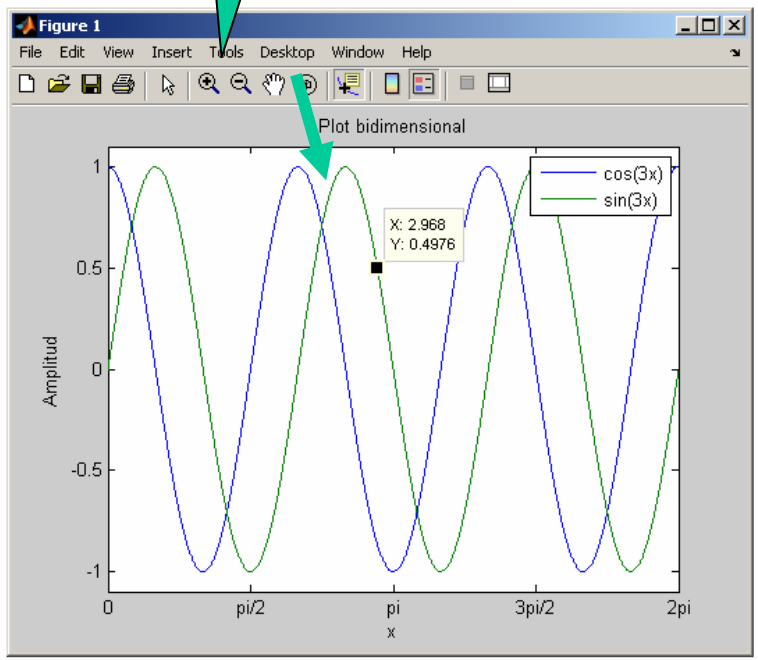


Gráficos bidimensionales

Zoom

Marcas de los ejes

```
>> x=[0:0.001:2*pi];
>> plot(x,cos(3*x),x,sin(3*x))
>> xlabel('x');
>> legend('cos(3x)','sin(3x)')
>> ylabel('Amplitud')
>> title('Plot bidimensional')
>> axis([0 2*pi -1.1 1.1])
>> set(gca,'ytick',[-1 -0.5 0 0.5 1])
>> set(gca,'xtick',[0 pi/2 pi 3*pi/2 2*pi])
>> set(gca,'XTickLabel',{'0';'pi/2';'pi';'3pi/2';'2pi'})
>>
```



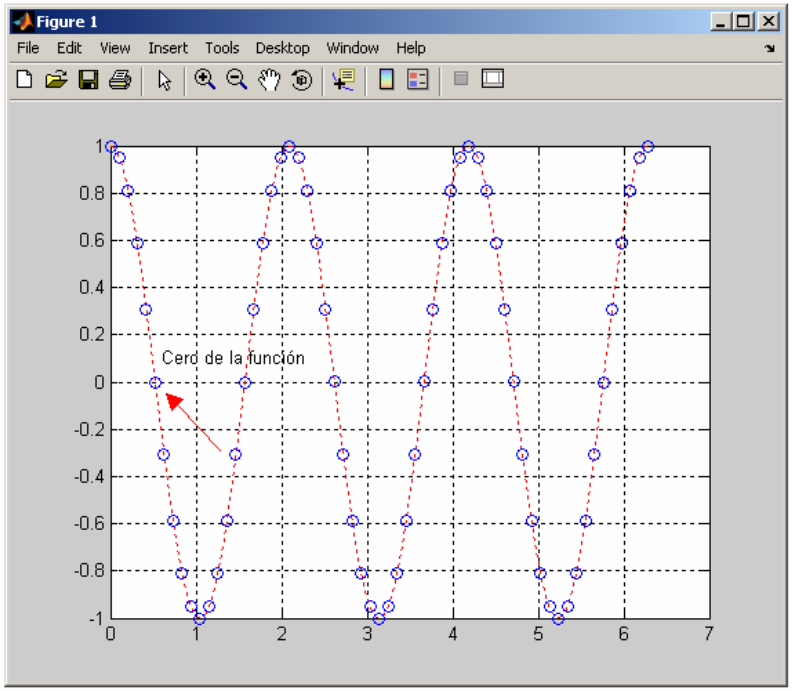
Gráficos bidimensionales

Grid

Handle

Anotaciones

```
>> x1=[0:0.001:2*pi];
>> x2=[0:pi/30:2*pi];
>> plot(x1,cos(3*x1),'r:',x2,cos(3*x2),'bo')
>> grid
>> handle_arrow=annotation('arrow',0.13+[1/7 1/14],[4/10 5/10],'Color',[1 0 0])

handle_arrow =

  161.0106

>> text(0.6,0.1,'Cero de la función')
>> set(handle_arrow,'HeadStyle','plain')
>>
```

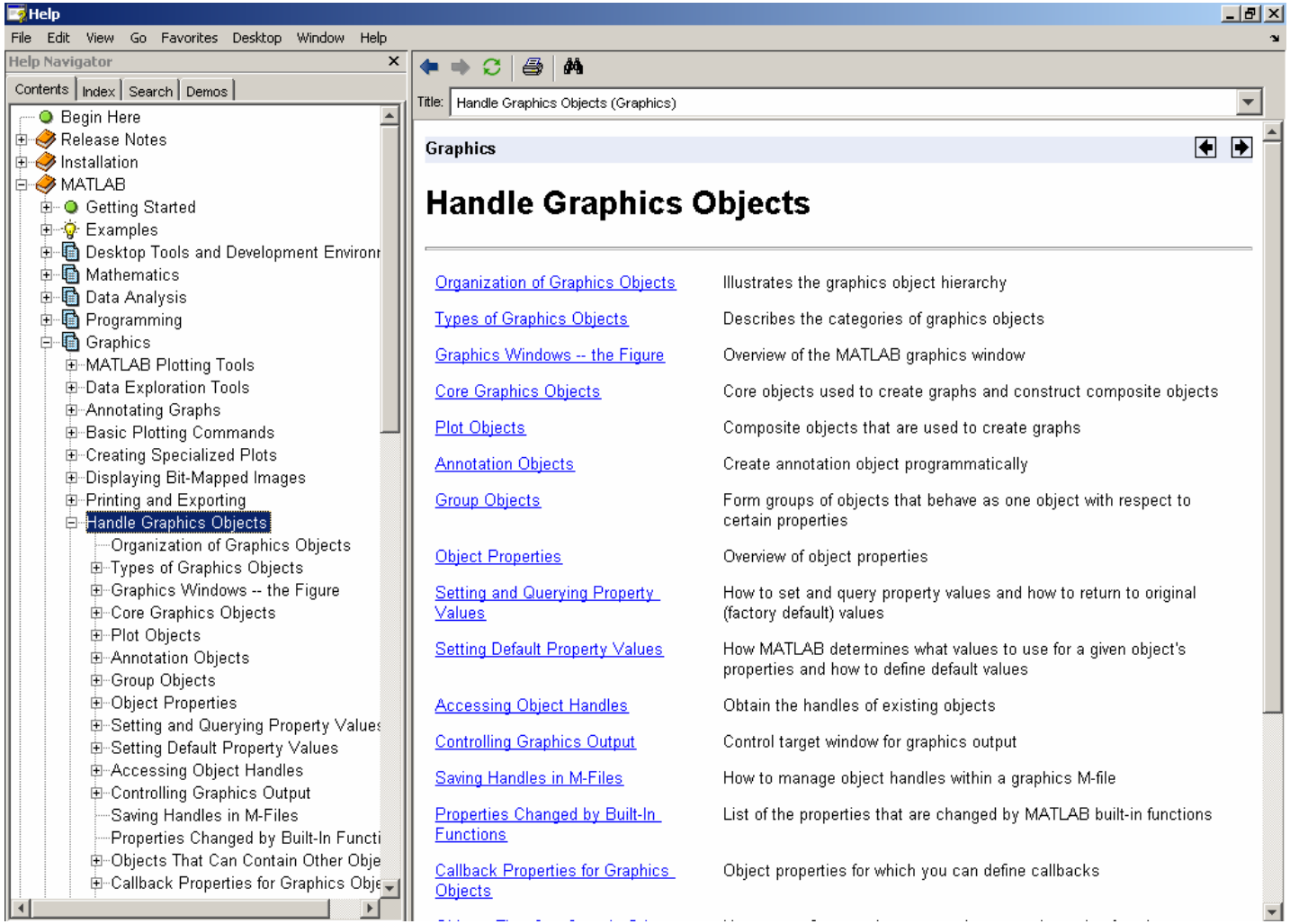


Gráficos bidimensionales

Símbolo	Color	Símbolo	Marcadores (markers)
y	yellow	.	puntos
m	magenta	o	círculos
c	cyan	x	marcas en x
r	red	+	marcas en +
g	green	*	marcas en *
b	blue	s	marcas cuadradas (square)
w	white	d	marcas en diamante (diamond)
k	black	^	triángulo apuntando arriba
		v	triángulo apuntando abajo
Símbolo	**Estilo de línea**	>	triángulo apuntando a la dcha
-	líneas continuas	<	triángulo apuntando a la izda
:	líneas a puntos	p	estrella de 5 puntas
-.	líneas a barra-punto	h	estrella se seis puntas
--	líneas a trazos		

Tabla 1. Colores, markers y estilos de línea.

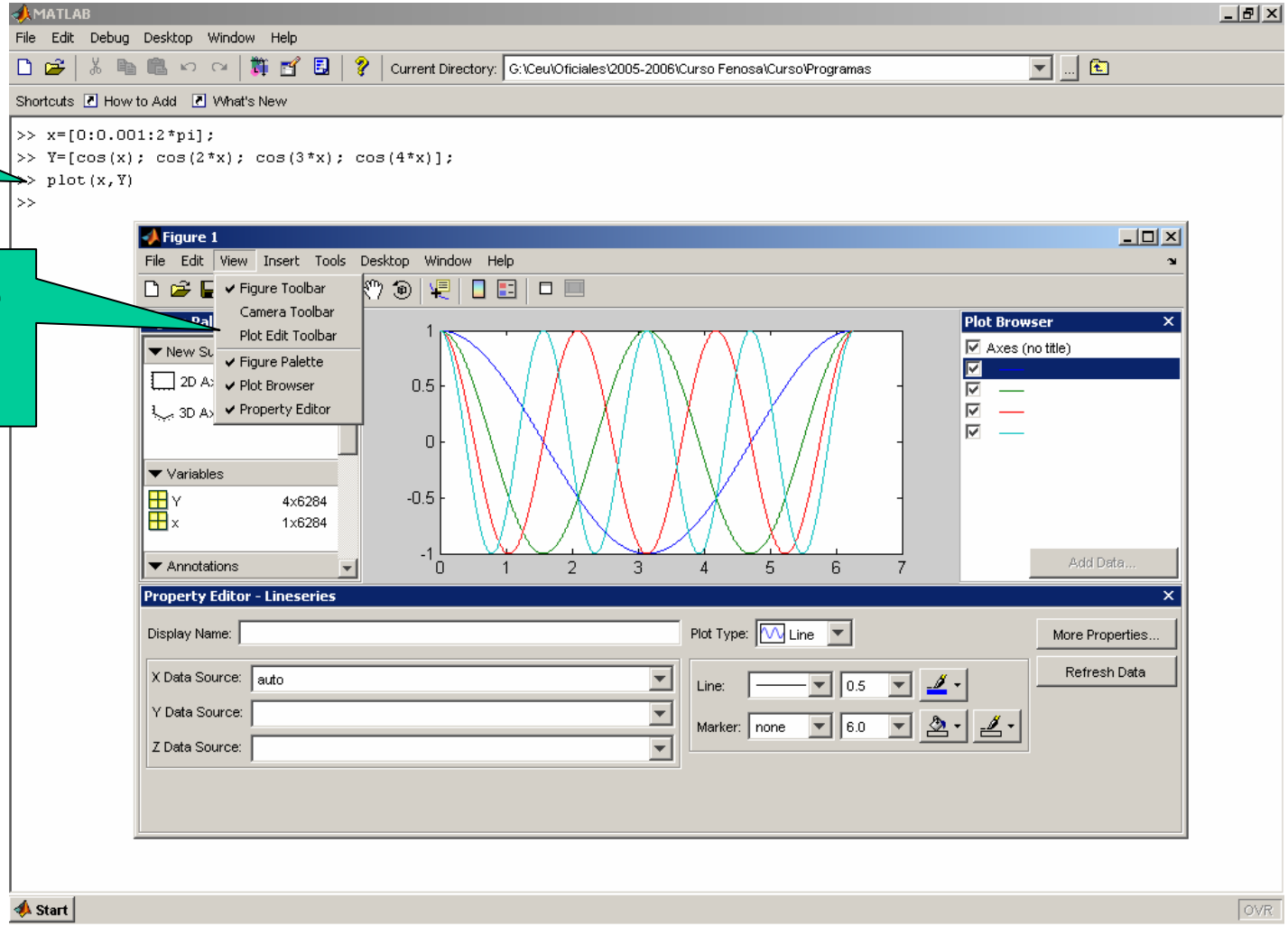
Gráficos bidimensionales



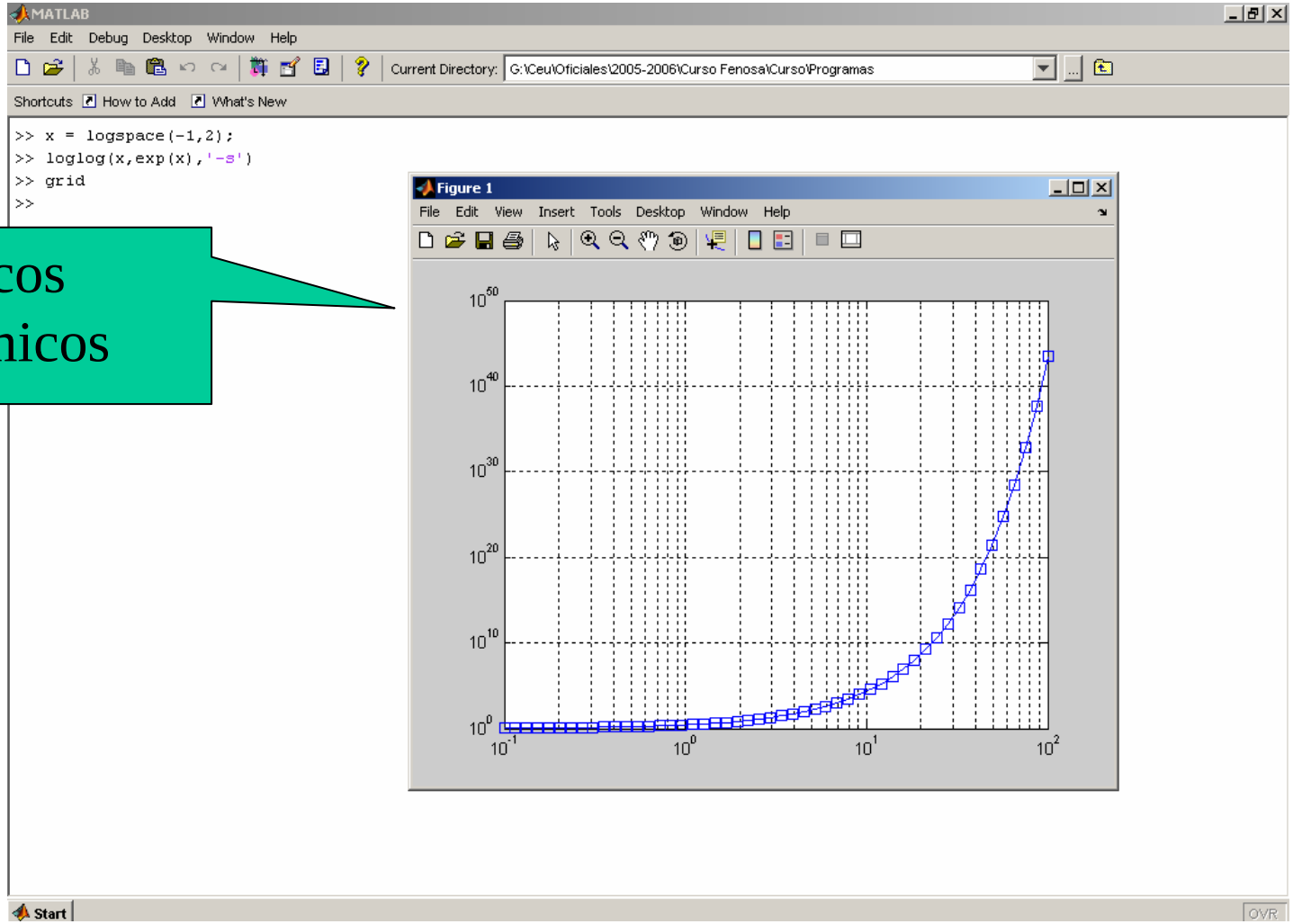
Gráficos bidimensionales

Plot de las filas de una matriz

Gestión gráfica de las propiedades

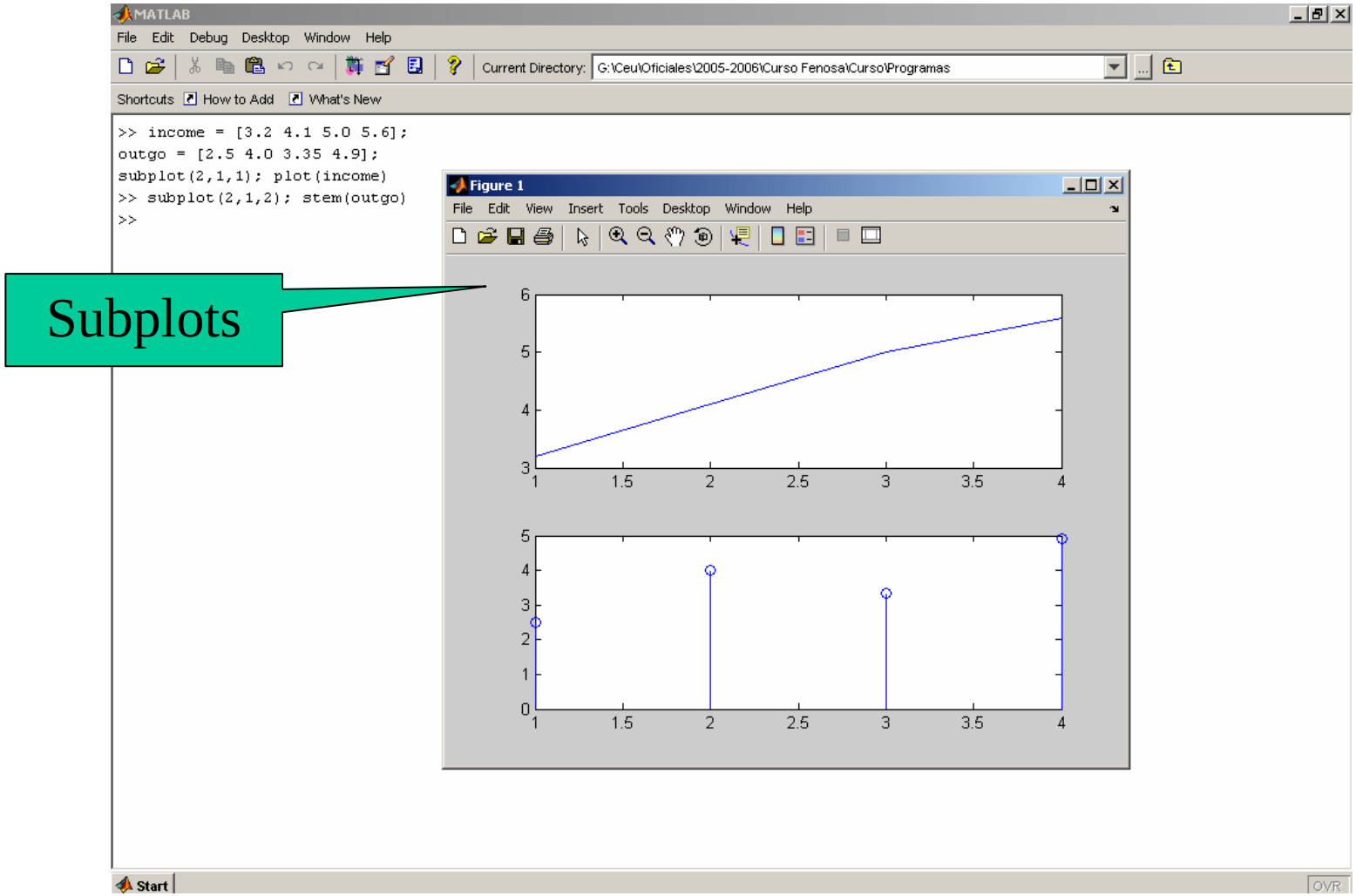


Gráficos bidimensionales



Gráficos logarítmicos

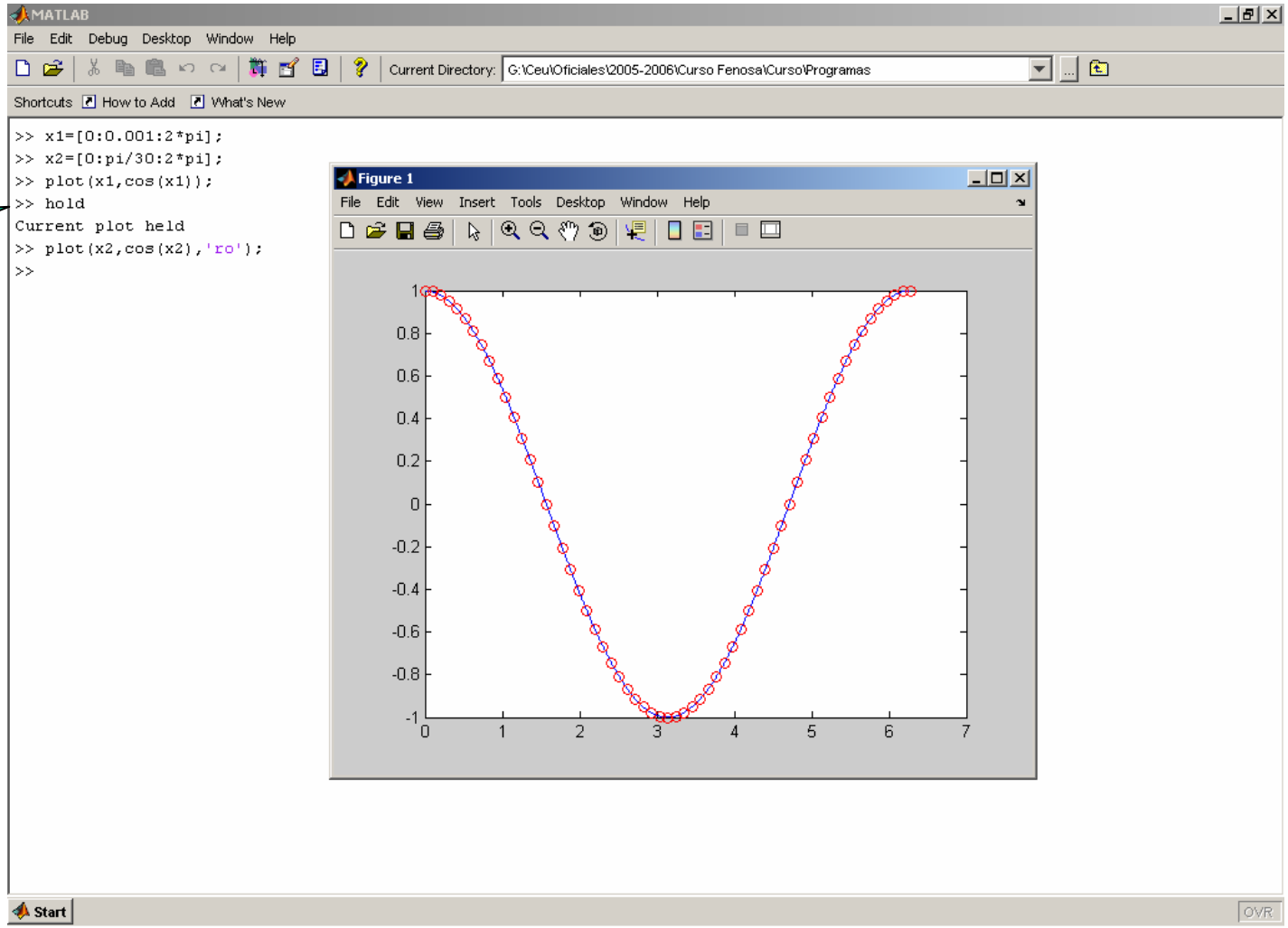
Gráficos bidimensionales



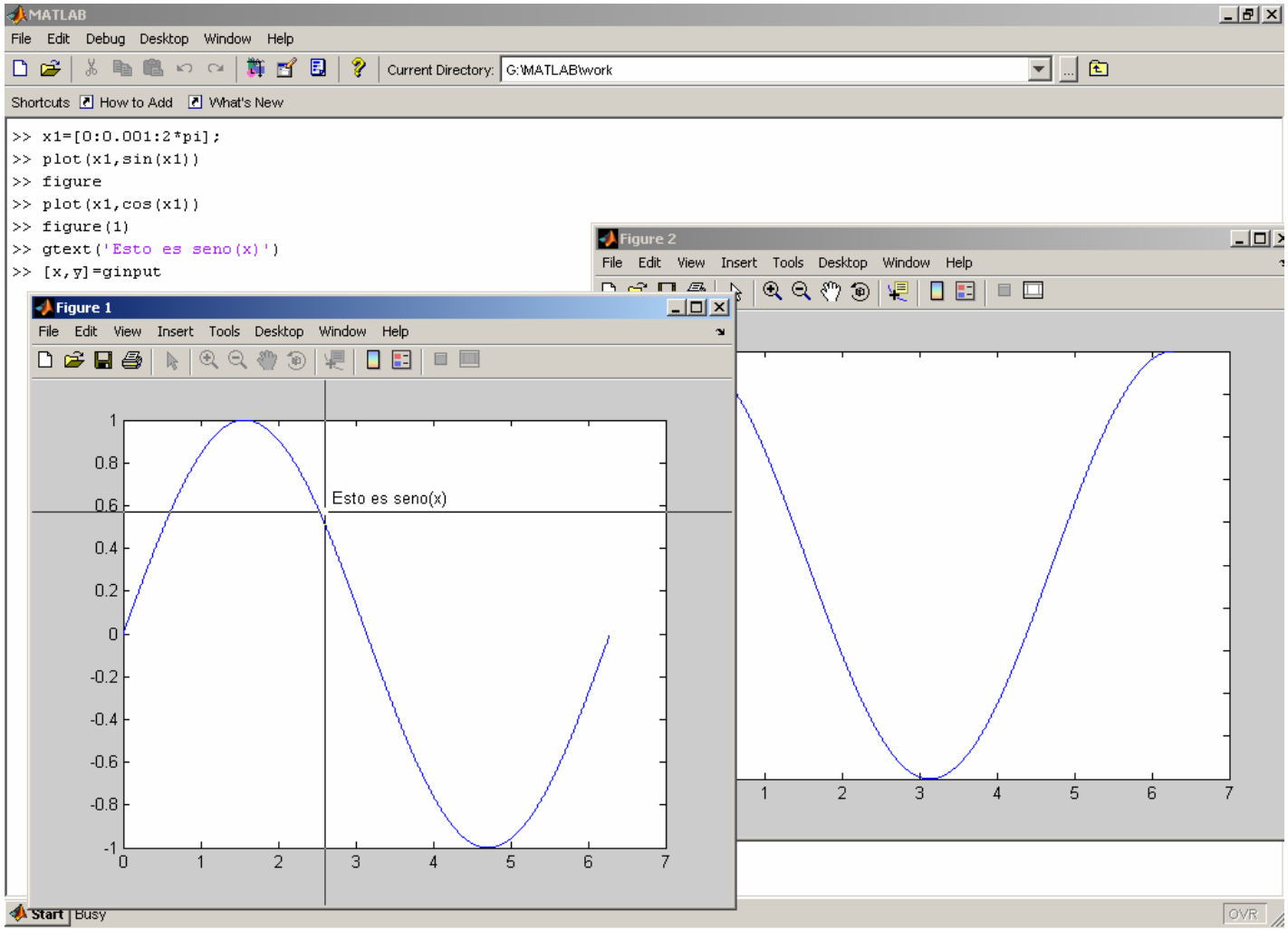
Subplots

Gráficos bidimensionales

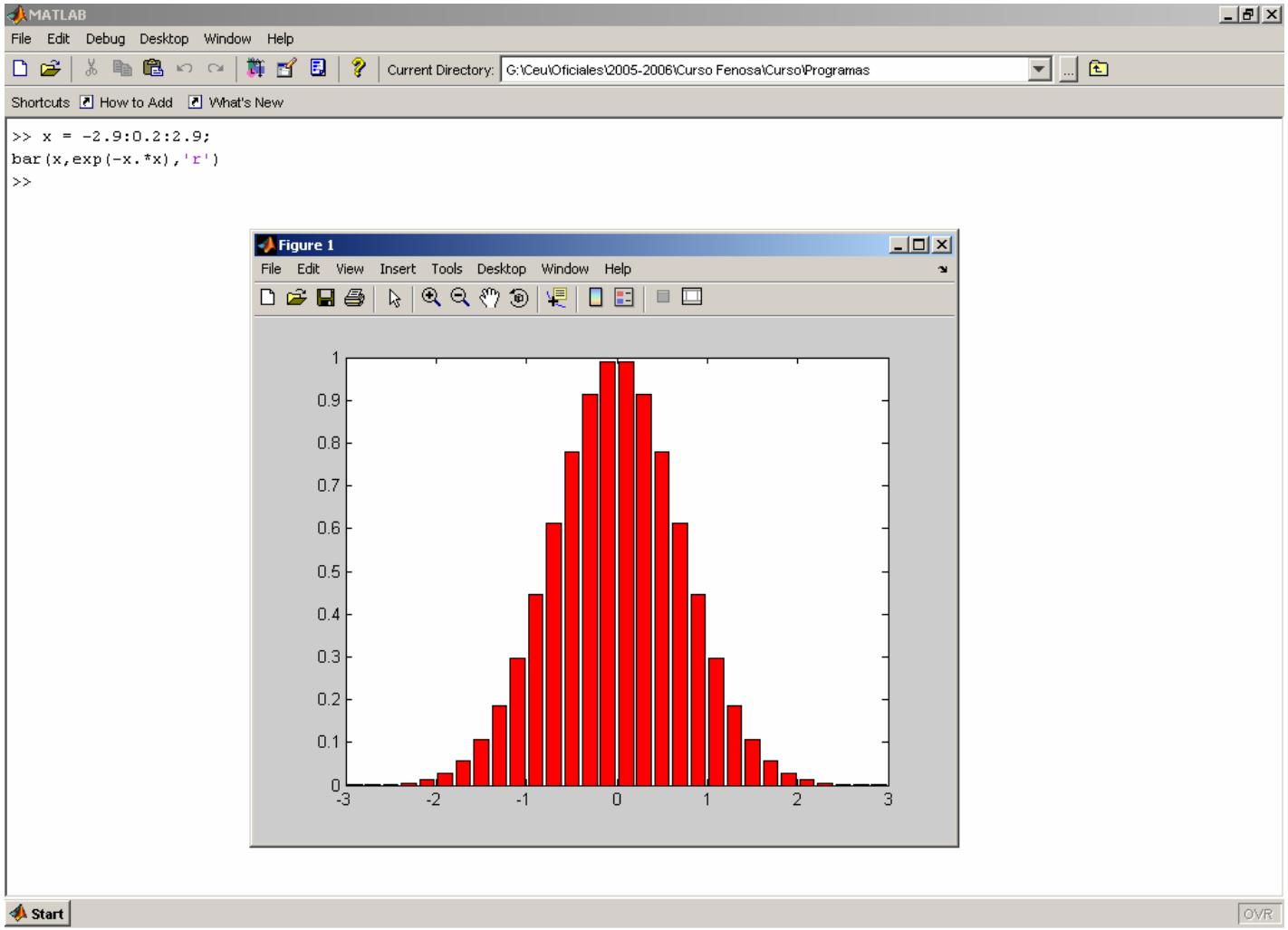
hold



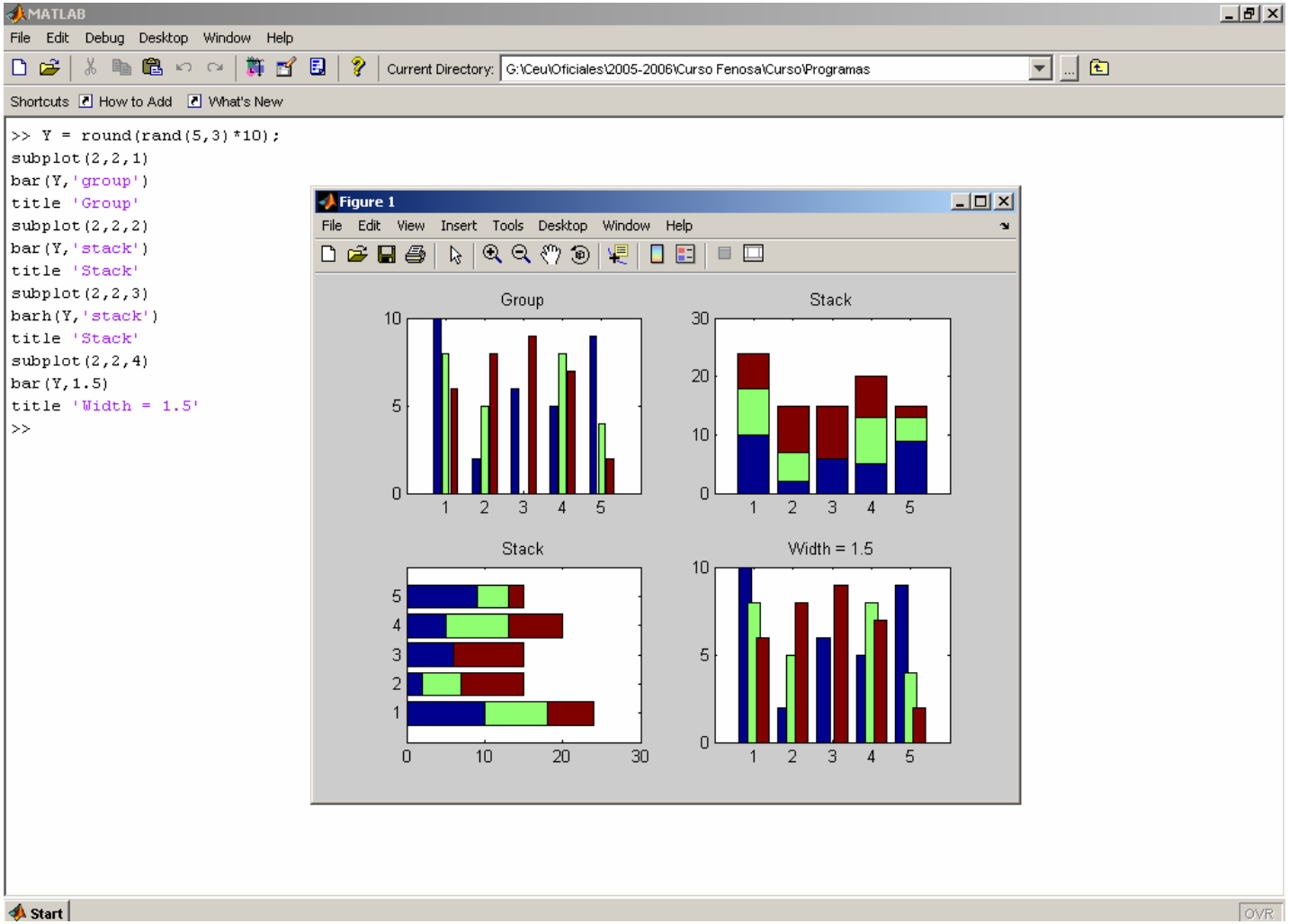
Gráficos bidimensionales



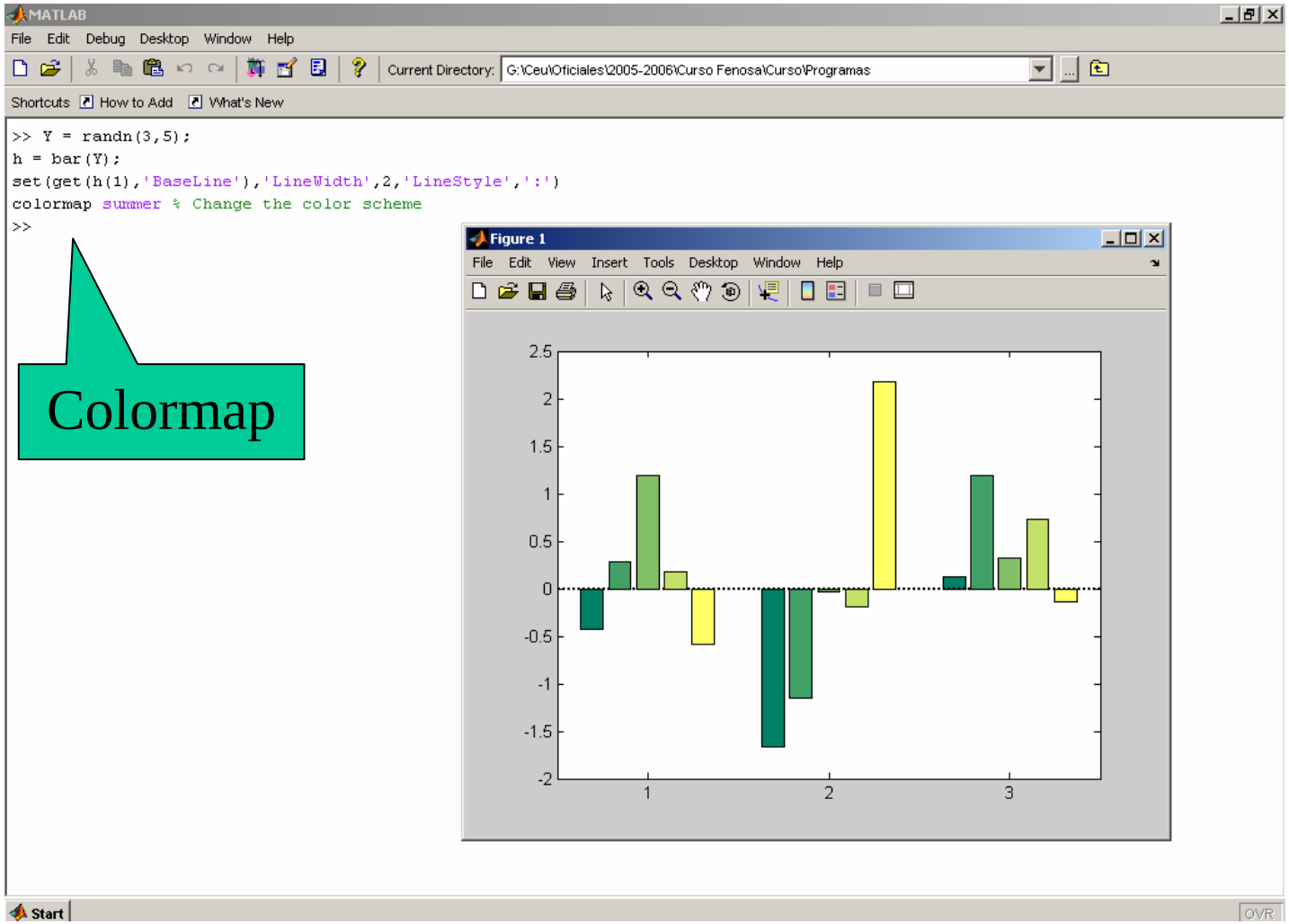
Gráficos bidimensionales



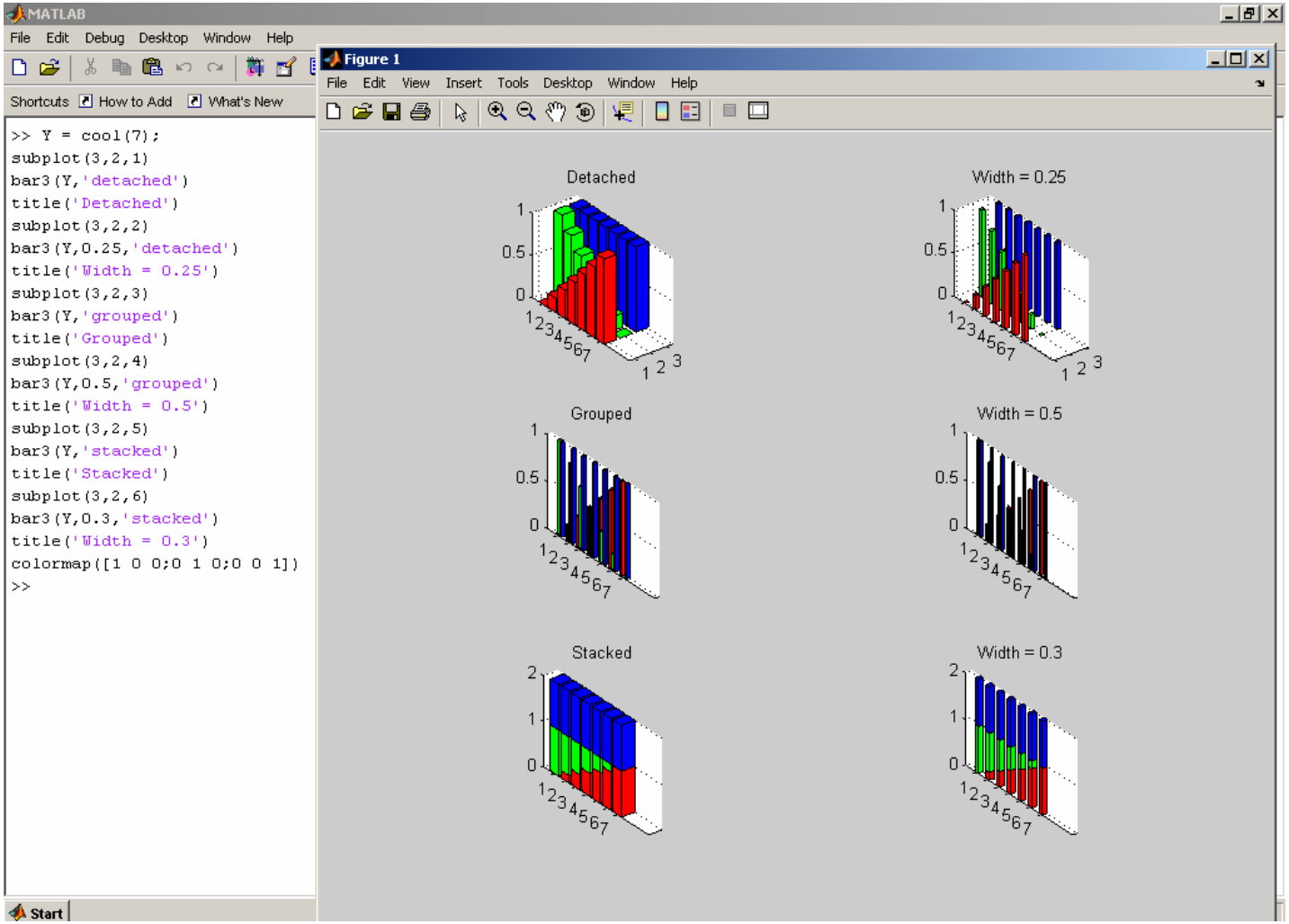
Gráficos bidimensionales



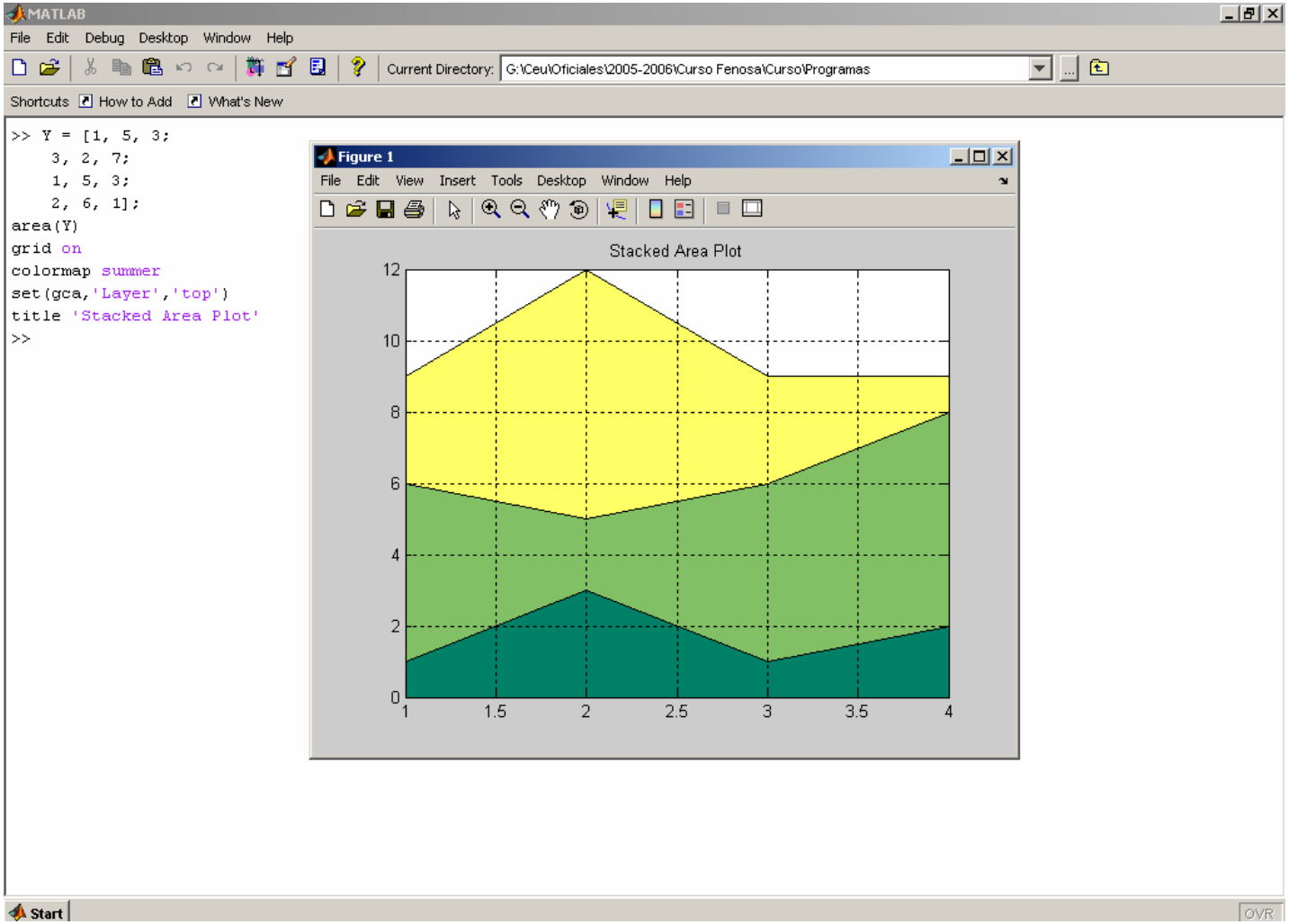
Gráficos bidimensionales



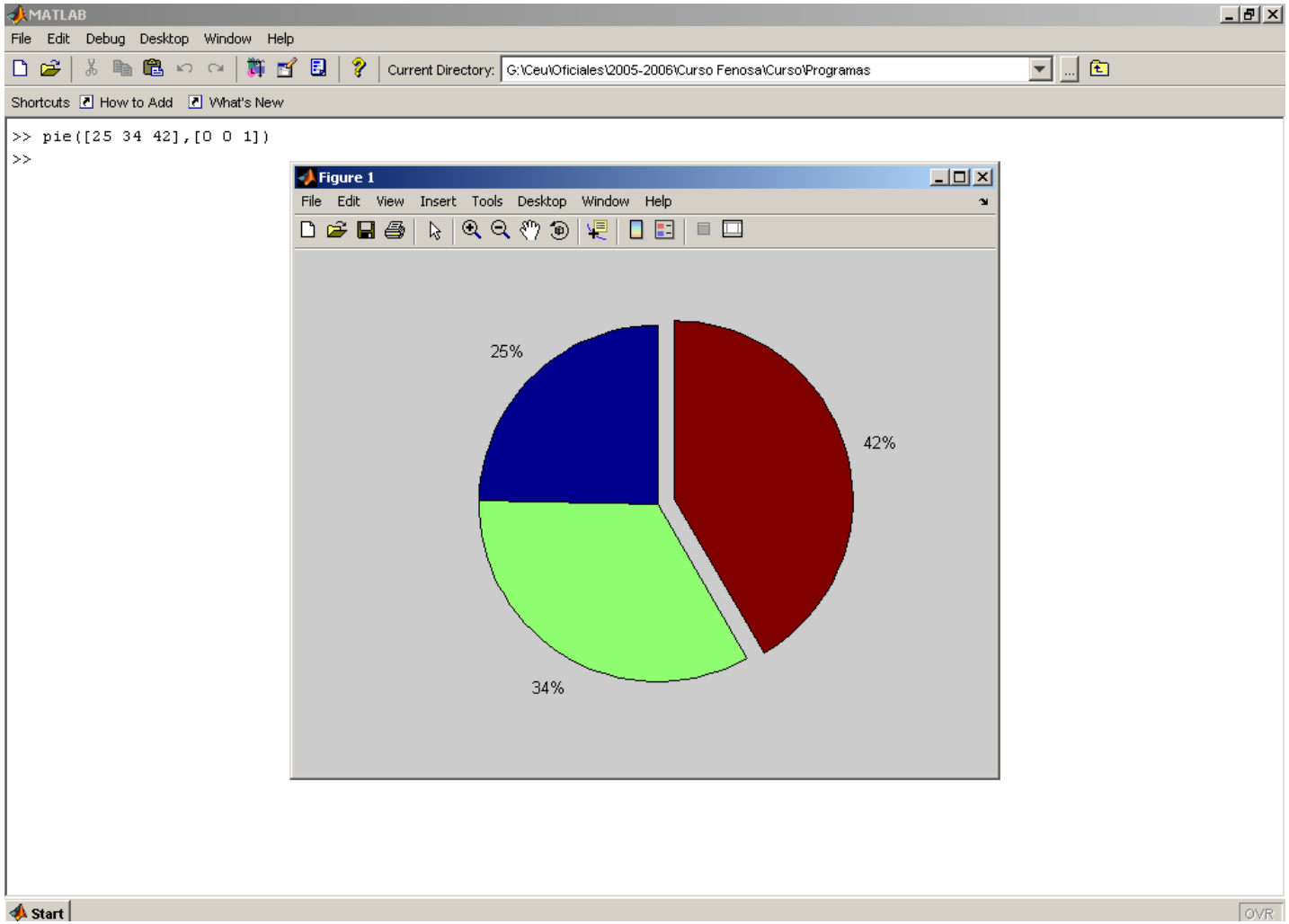
Gráficos bidimensionales



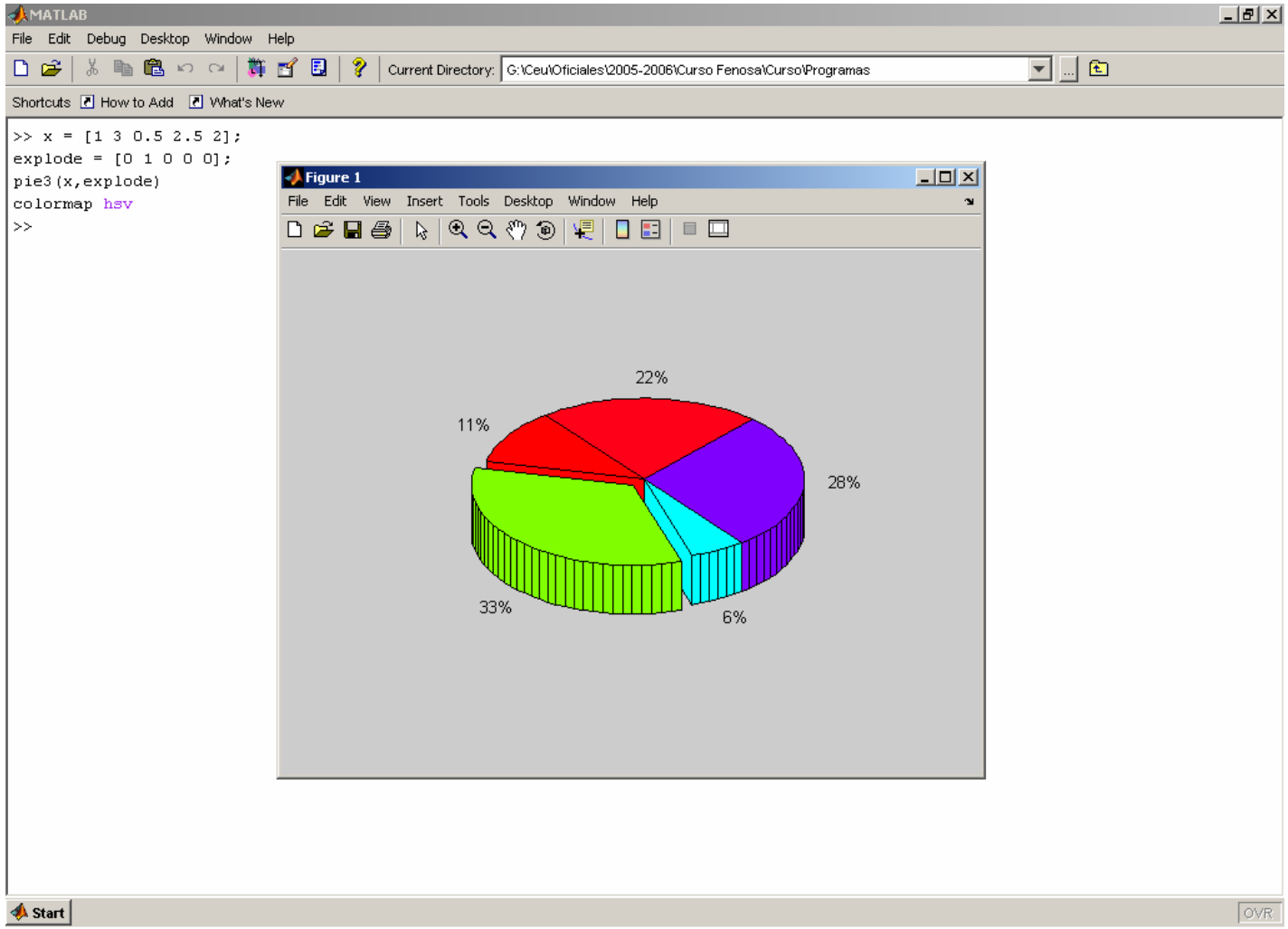
Gráficos bidimensionales



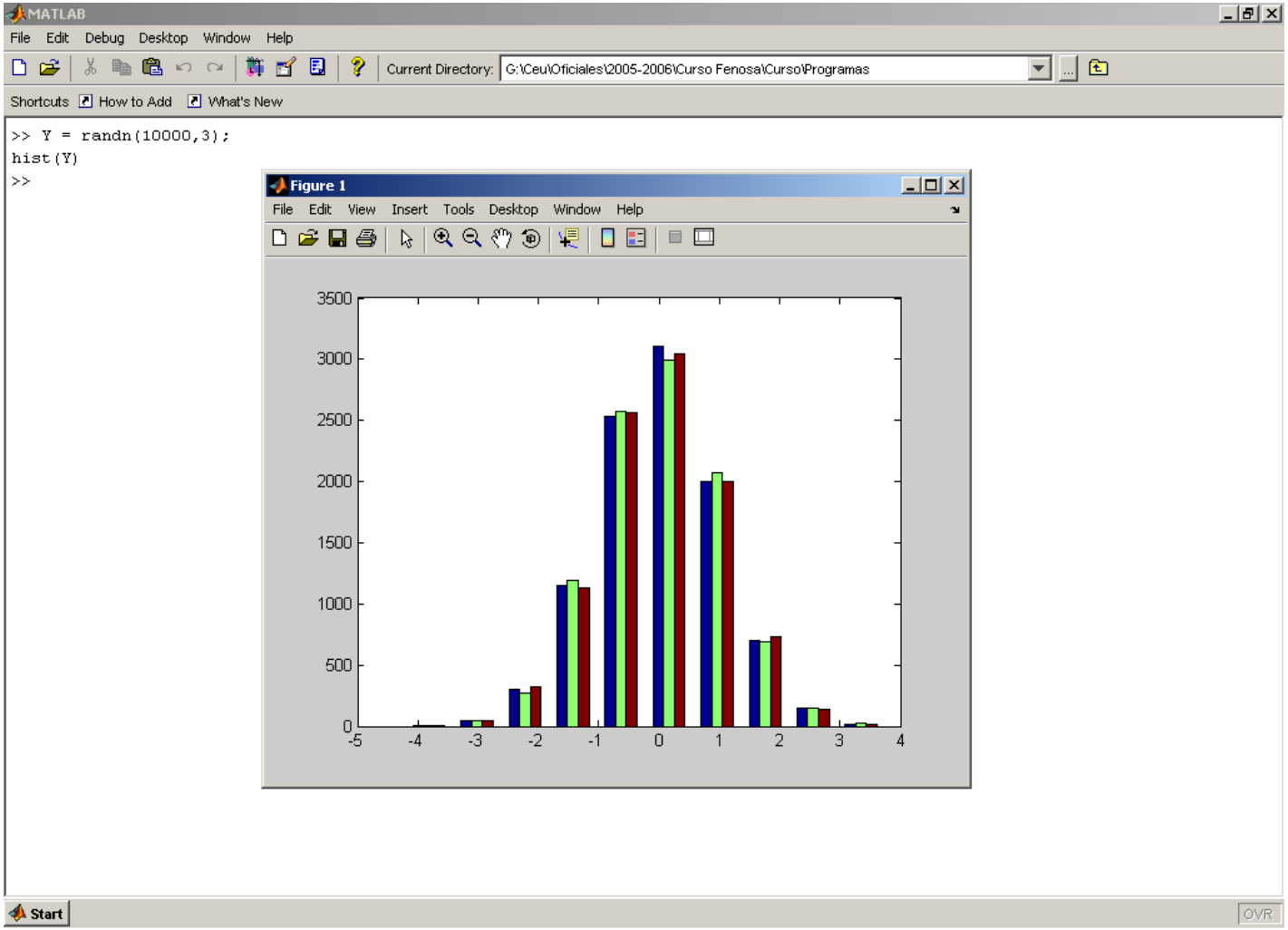
Gráficos bidimensionales



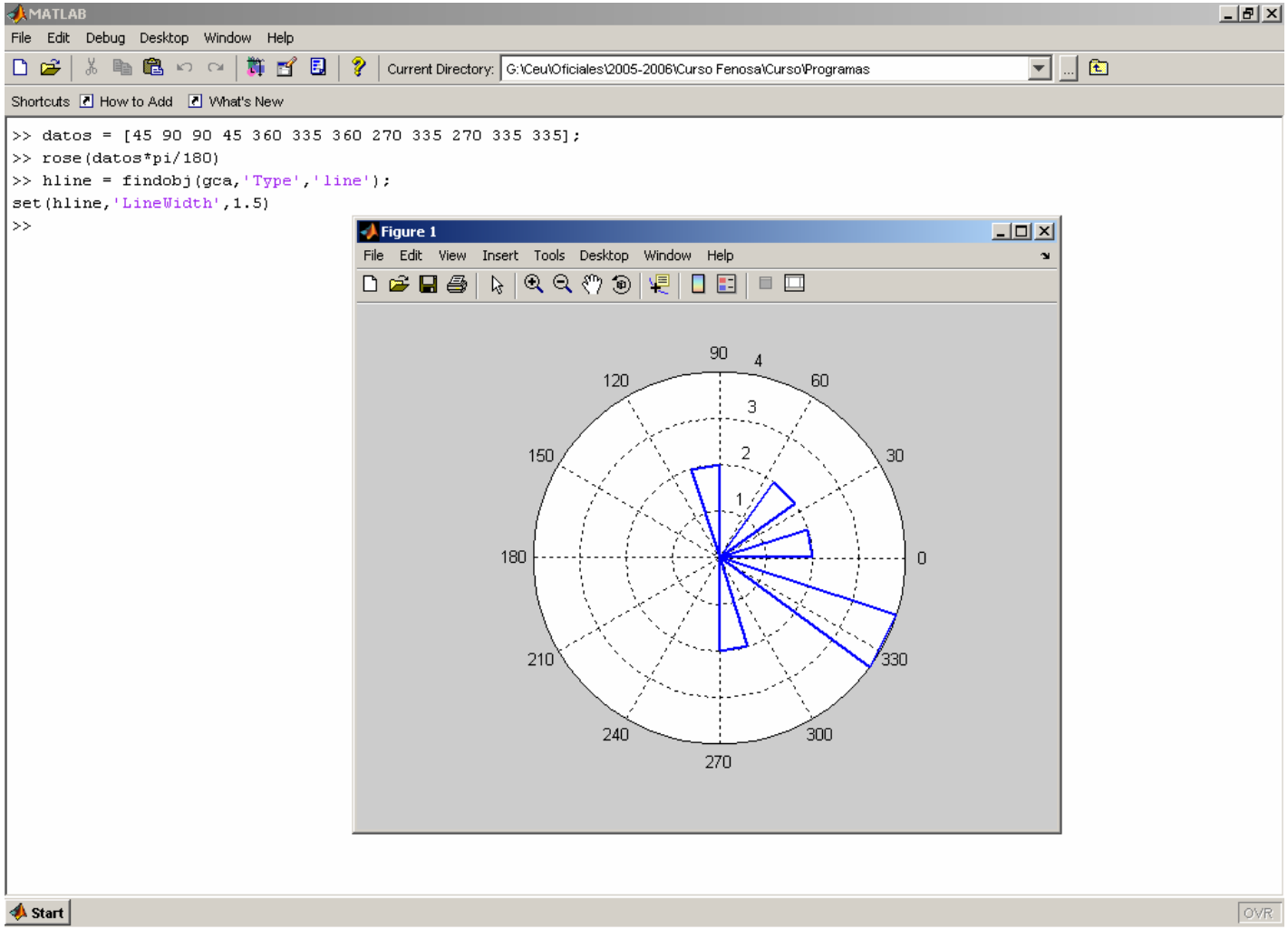
Gráficos bidimensionales



Gráficos bidimensionales

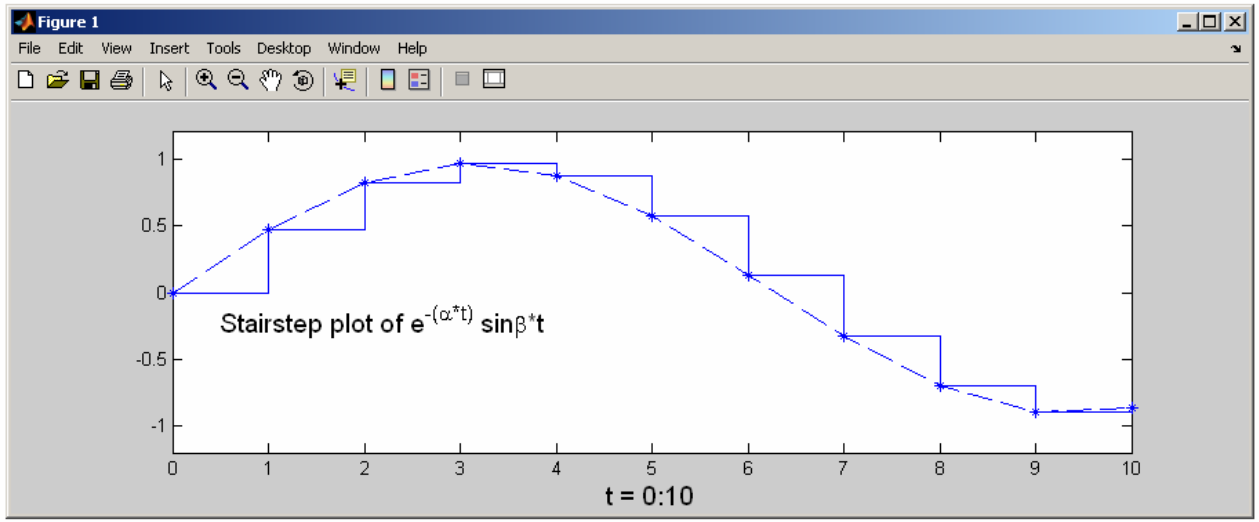


Gráficos bidimensionales

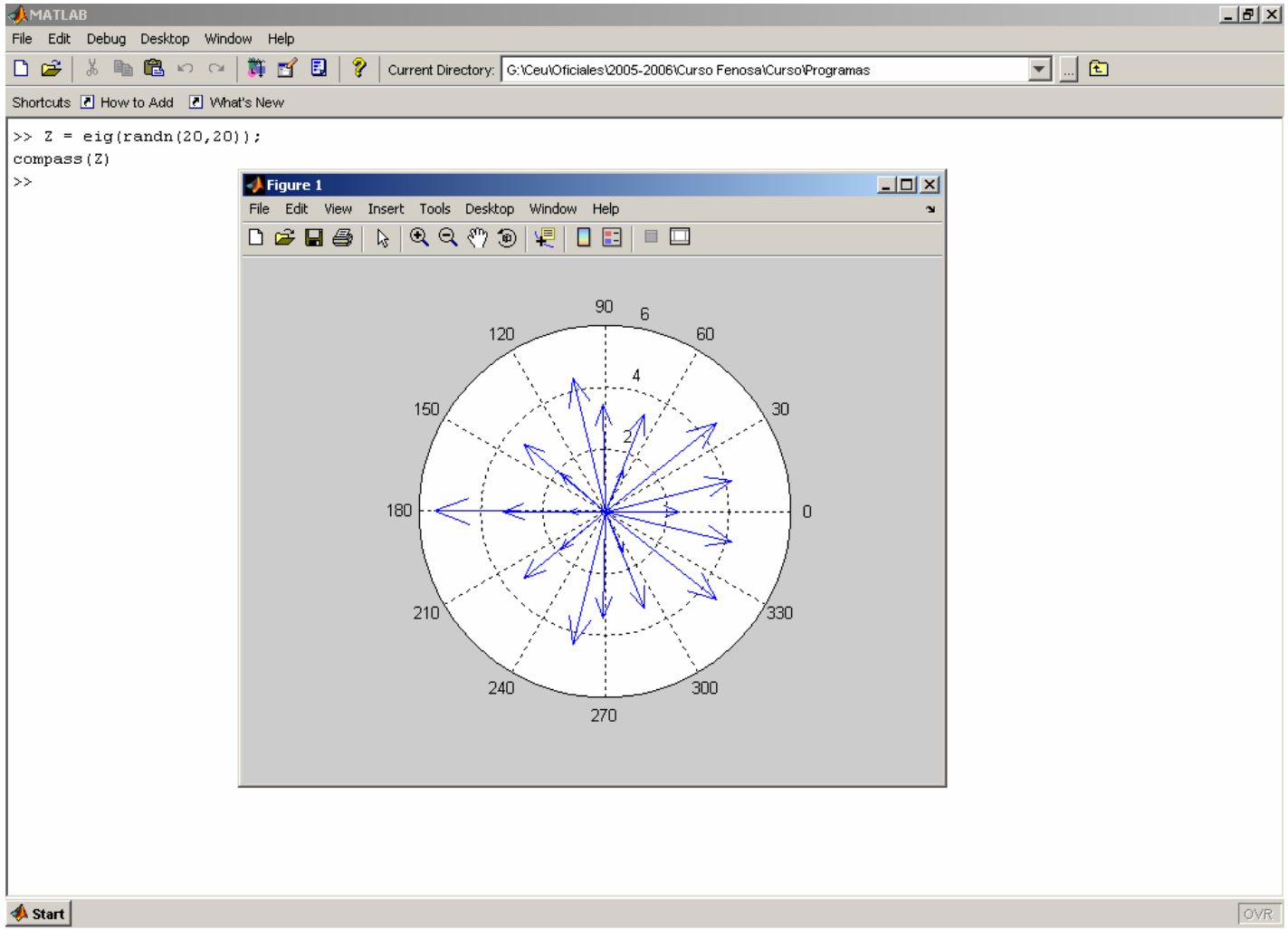


Gráficos bidimensionales

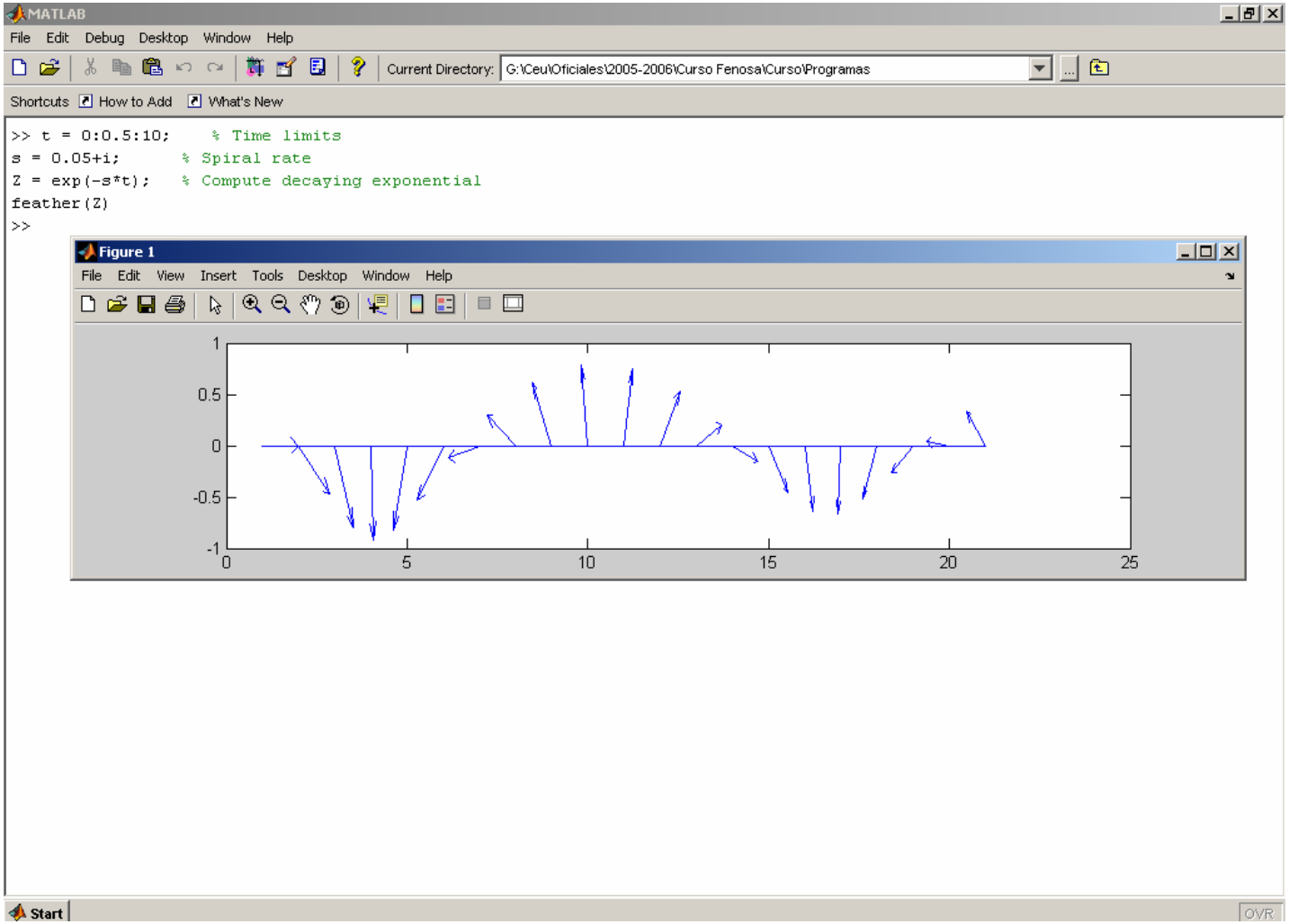
```
>> alpha = 0.01;
beta = 0.5;
t = 0:10;
f = exp(-alpha*t).*sin(beta*t);
>> stairs(t,f)
hold on
plot(t,f,'--*')
hold off
>> label = 'Stairstep plot of e^{-(\alpha*t)} sin\beta*t';
text(0.5,-0.2,label,'FontSize',14)
xlabel('t = 0:10','FontSize',14)
axis([0 10 -1.2 1.2])
>>
```



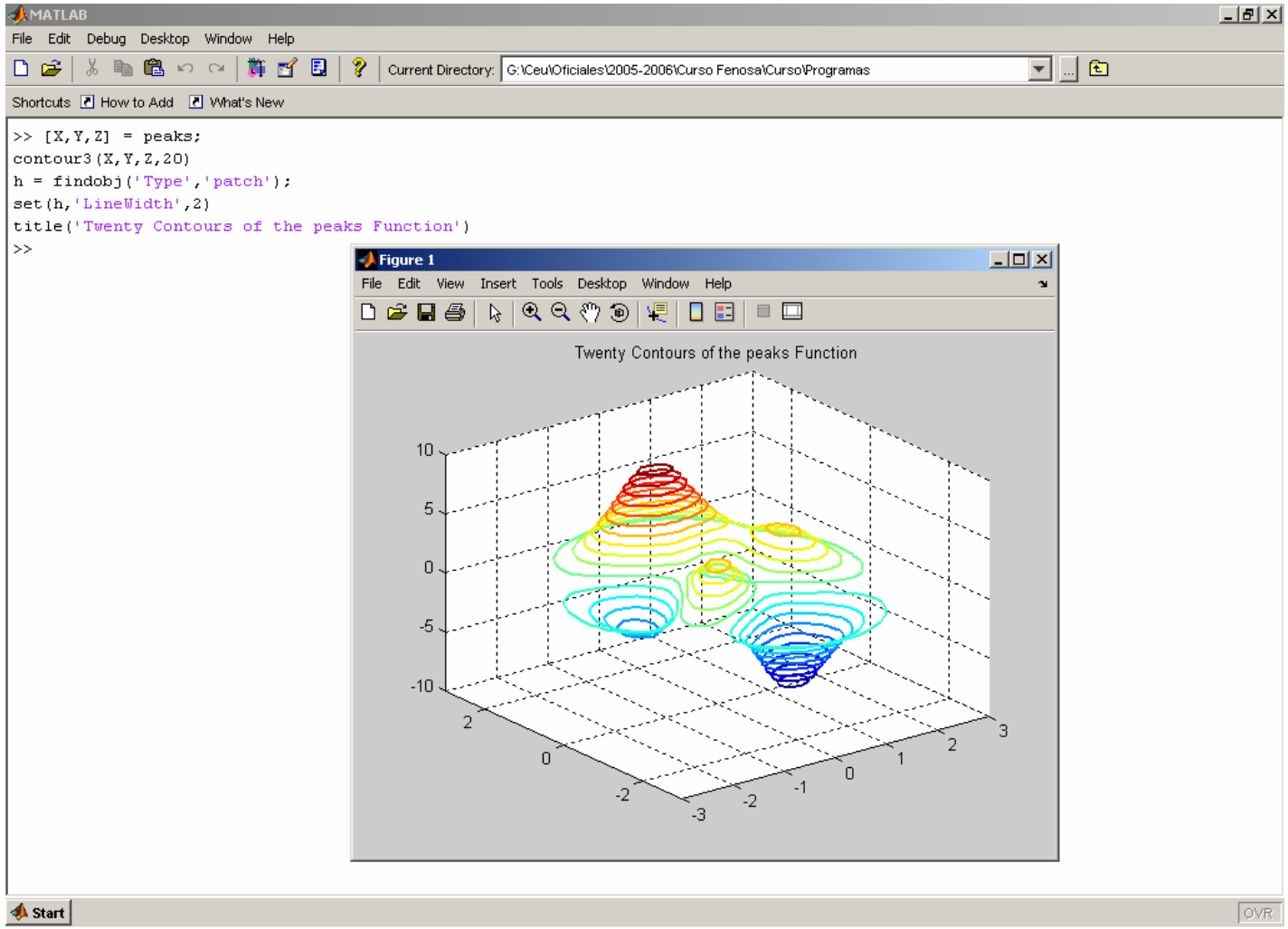
Gráficos bidimensionales



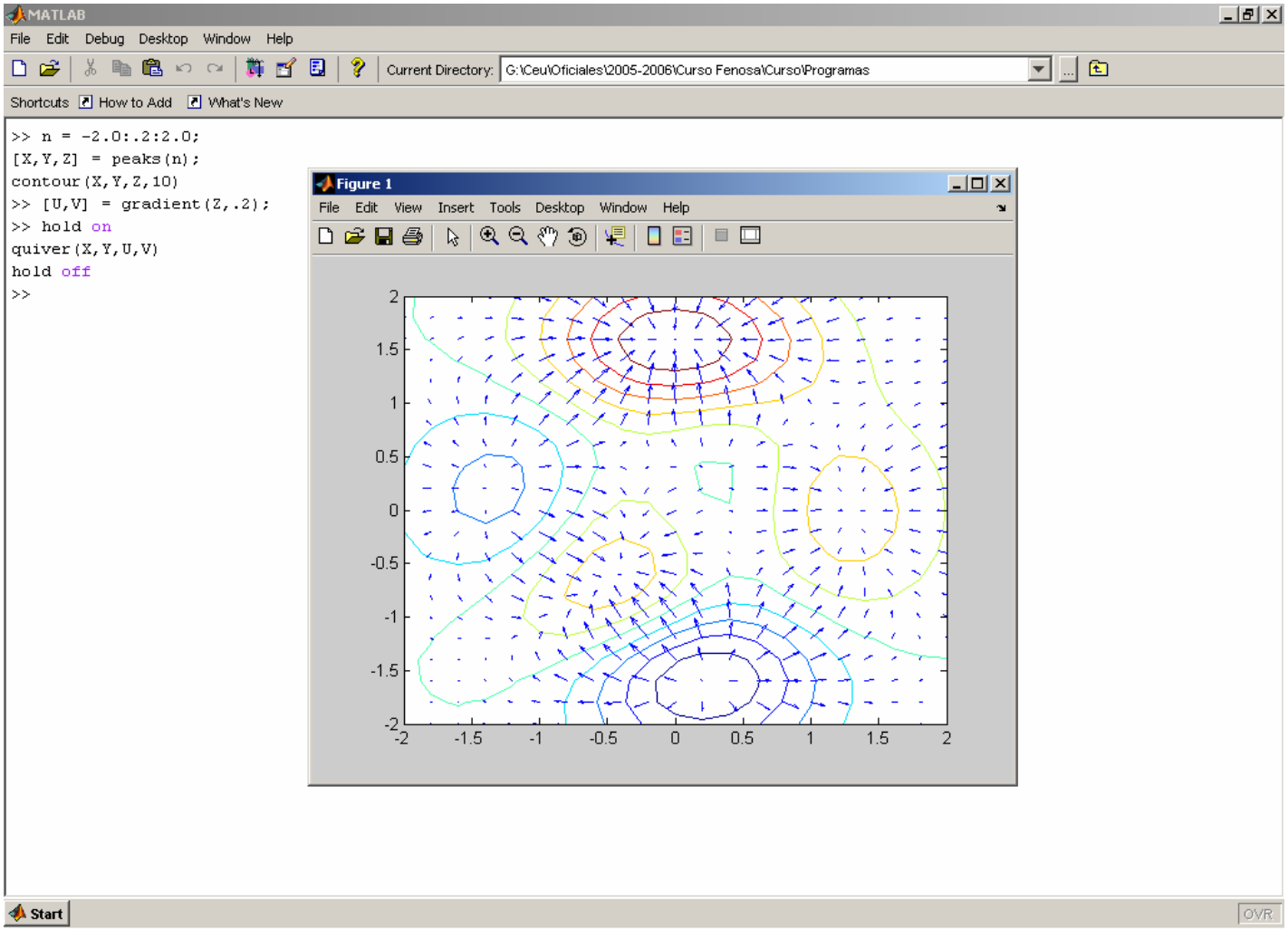
Gráficos bidimensionales



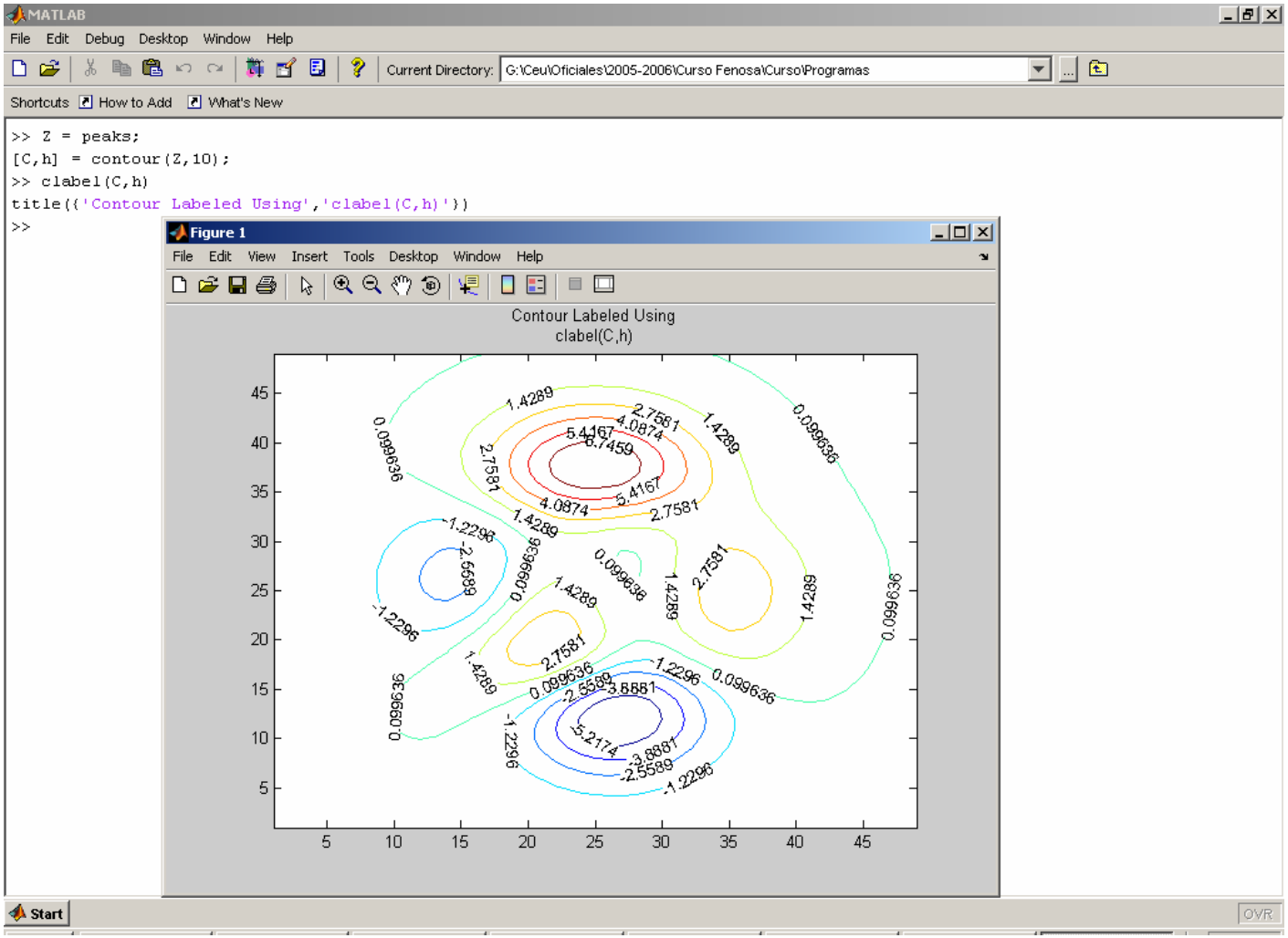
Gráficos tridimensionales



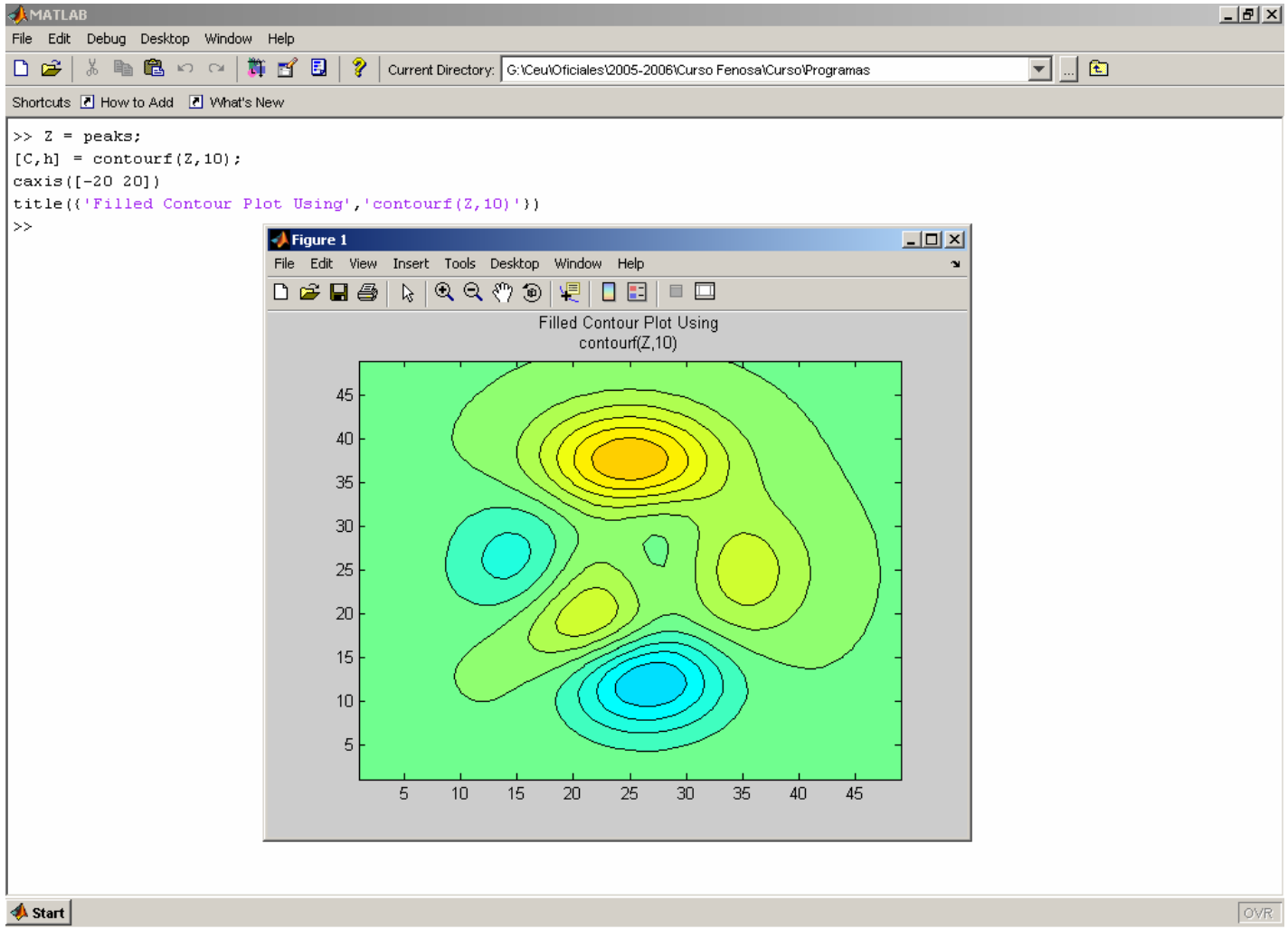
Gráficos bidimensionales



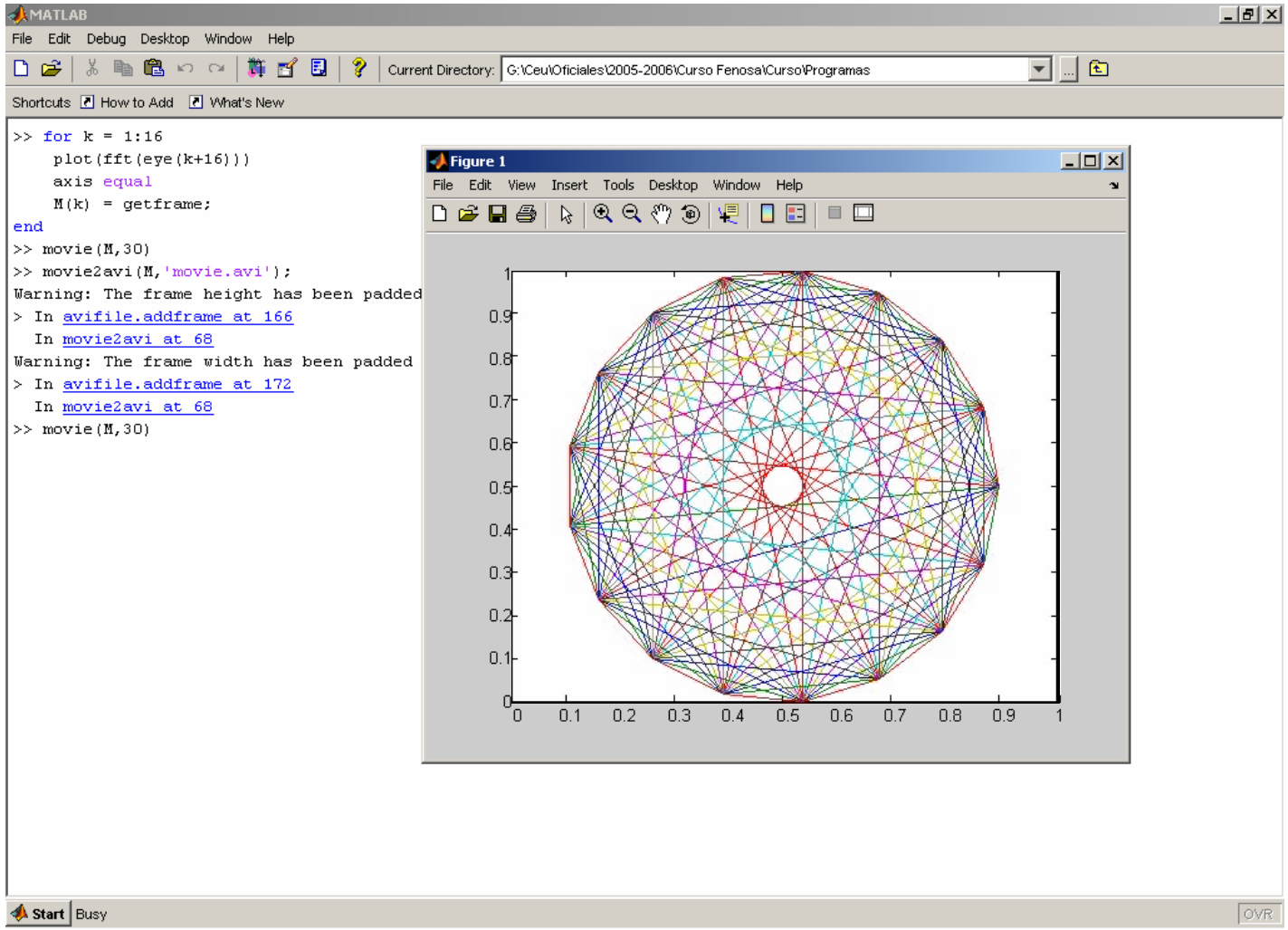
Gráficos bidimensionales



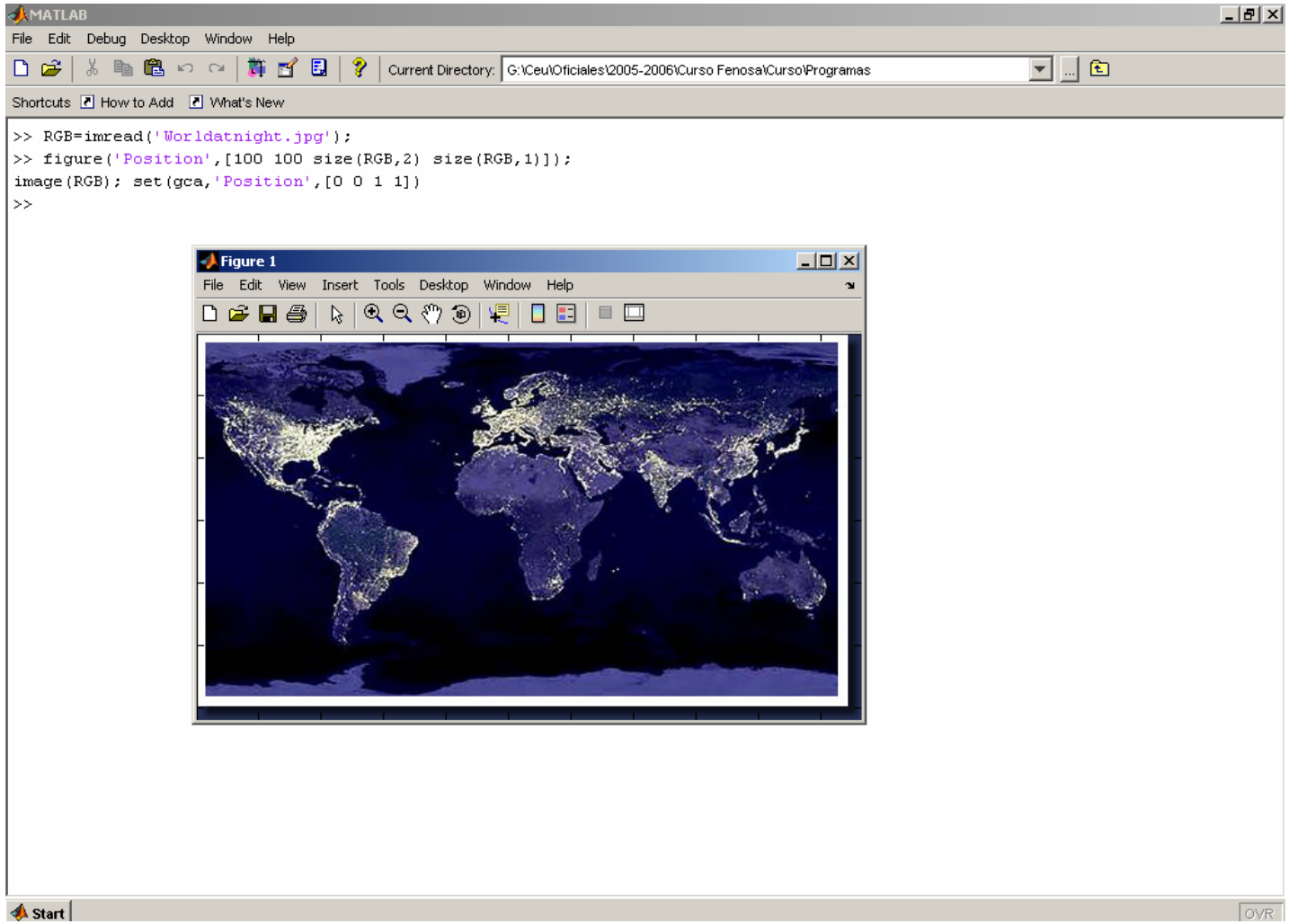
Gráficos bidimensionales



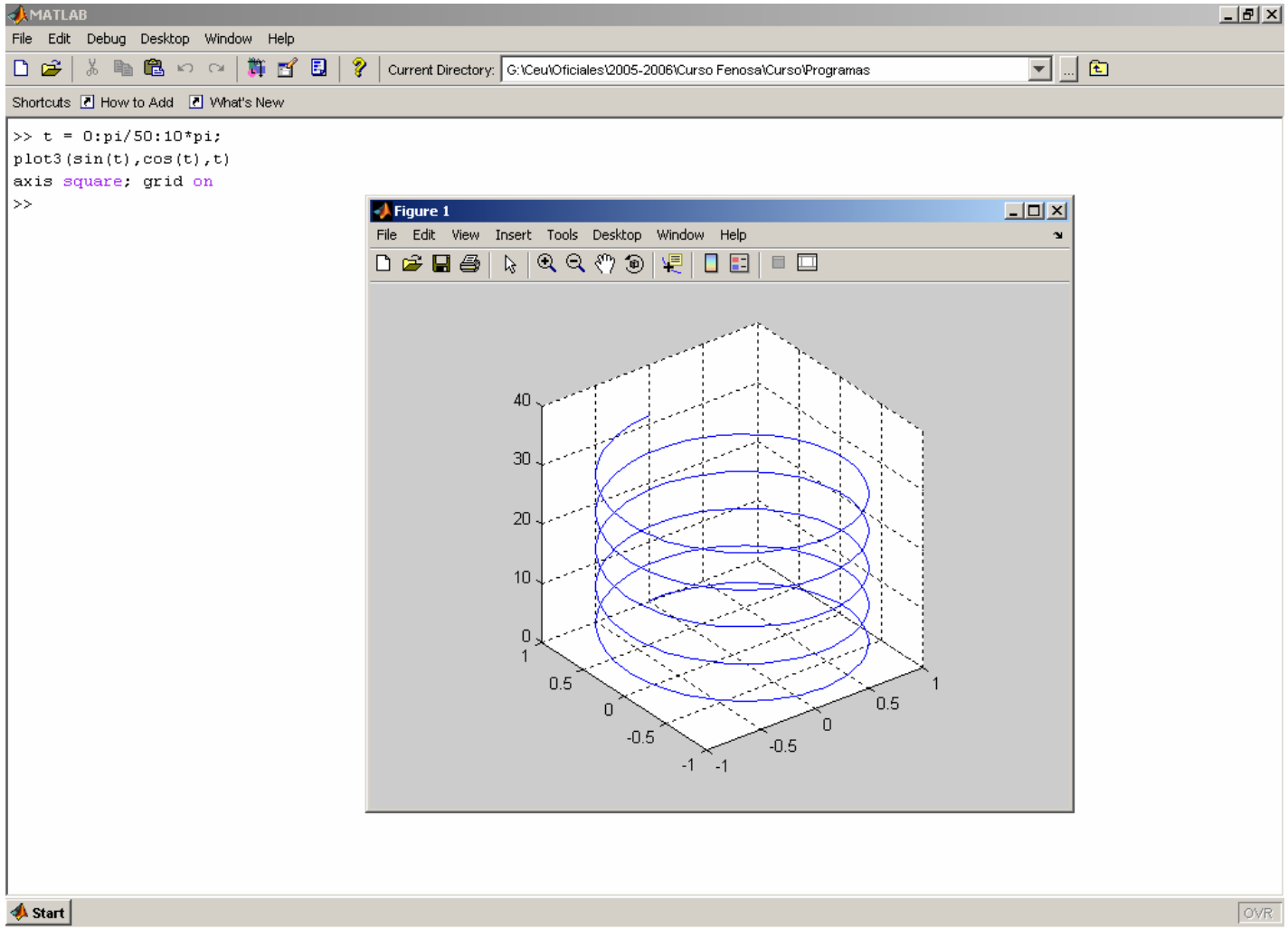
Animaciones



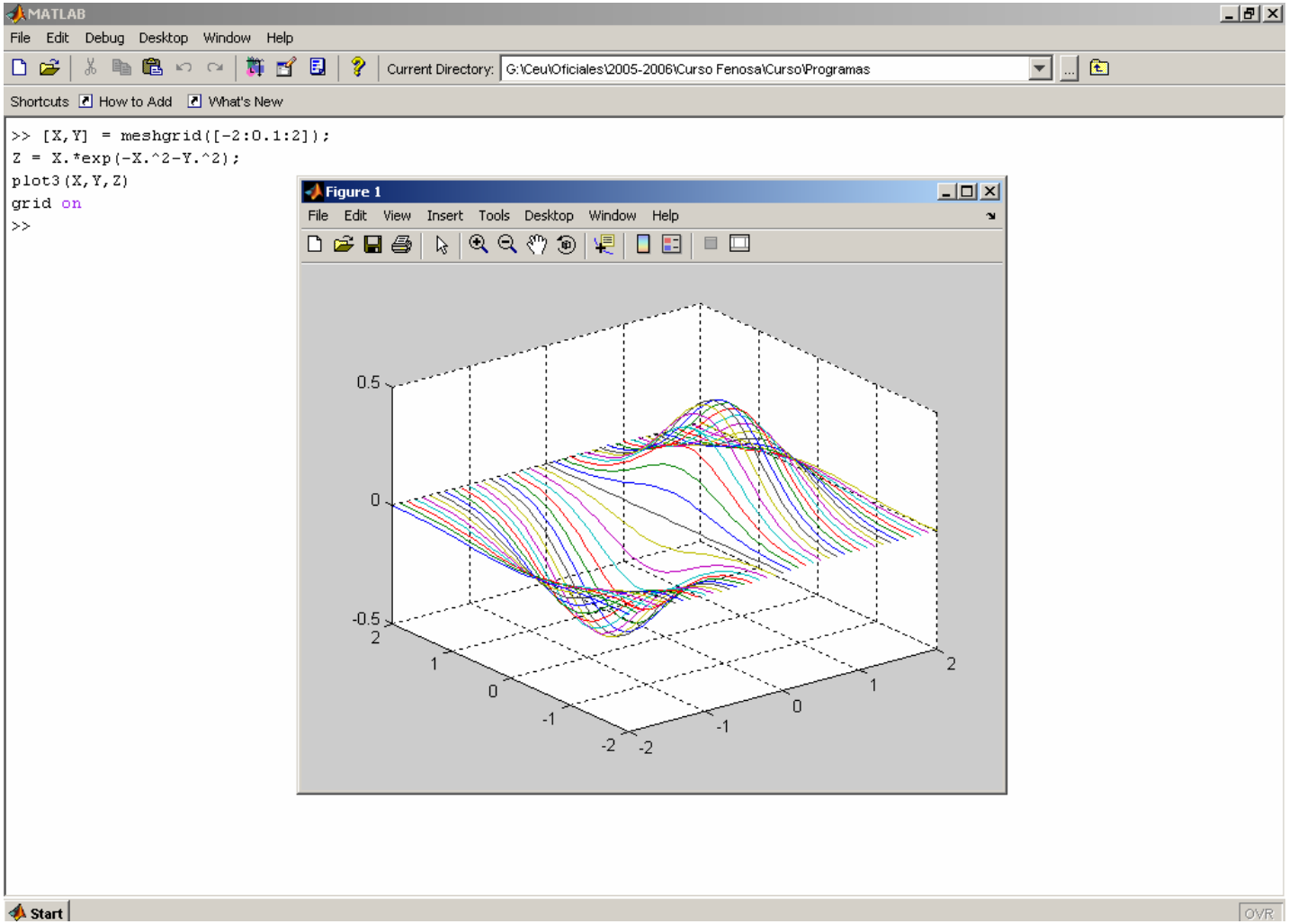
Imágenes


MATLAB
File Edit Debug Desktop Window Help
Current Directory: G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas
Shortcuts How to Add What's New
>> RGB=imread('Worldatnight.jpg');
>> figure('Position',[100 100 size(RGB,2) size(RGB,1)]);
image(RGB); set(gca,'Position',[0 0 1 1])
>>
Figure 1
File Edit View Insert Tools Desktop Window Help
Start
OVR

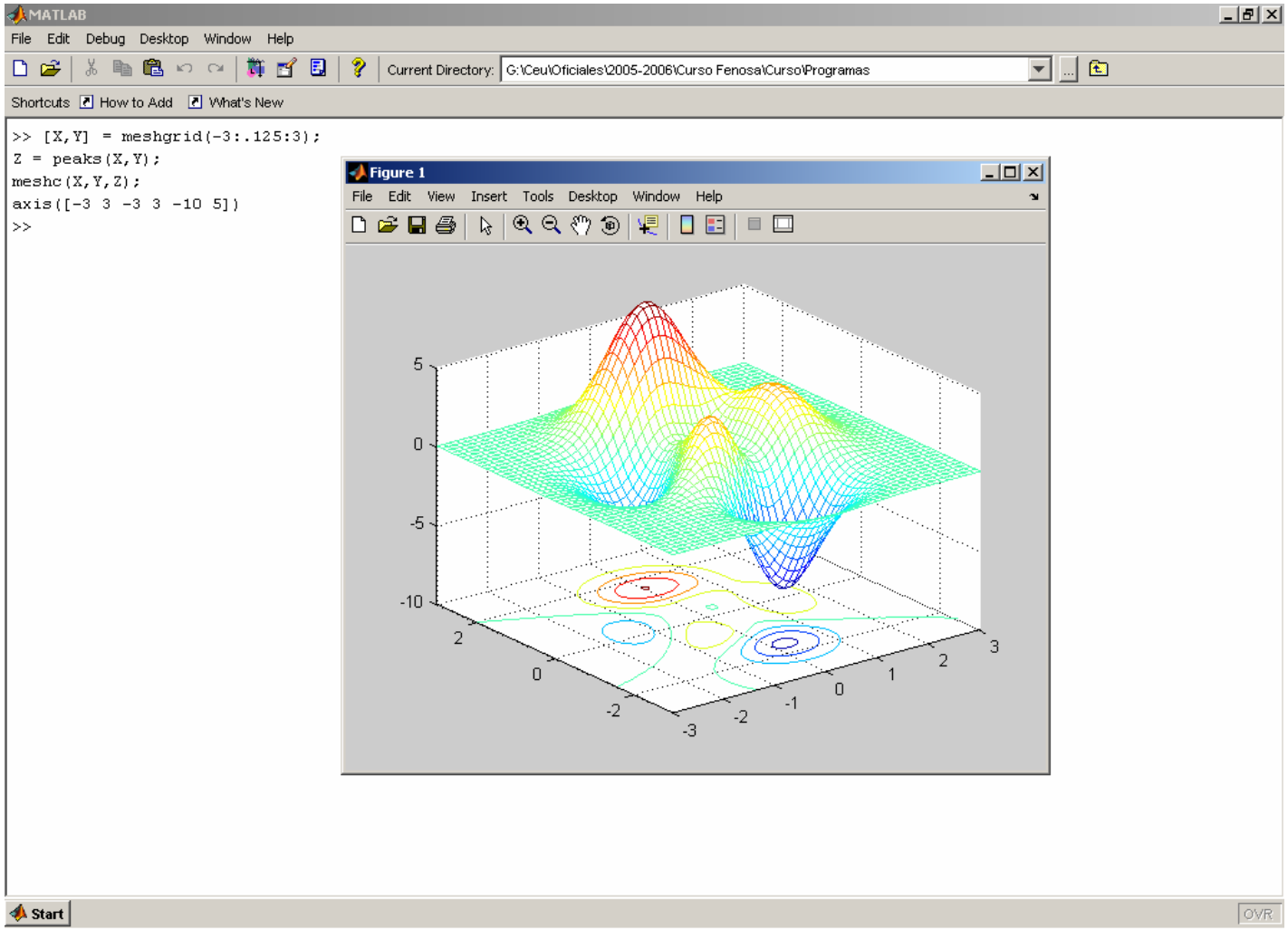
Gráficos tridimensionales



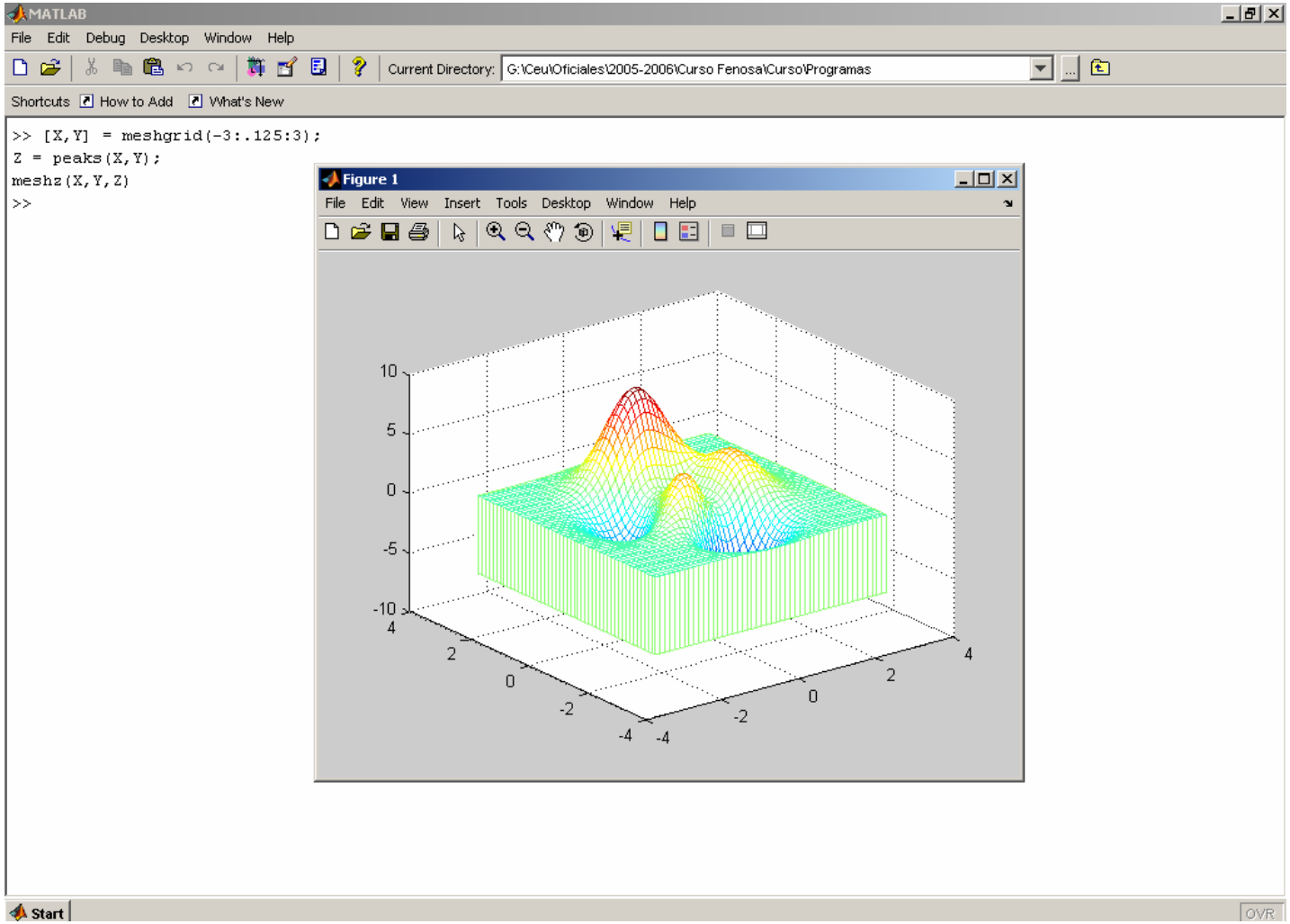
Gráficos tridimensionales



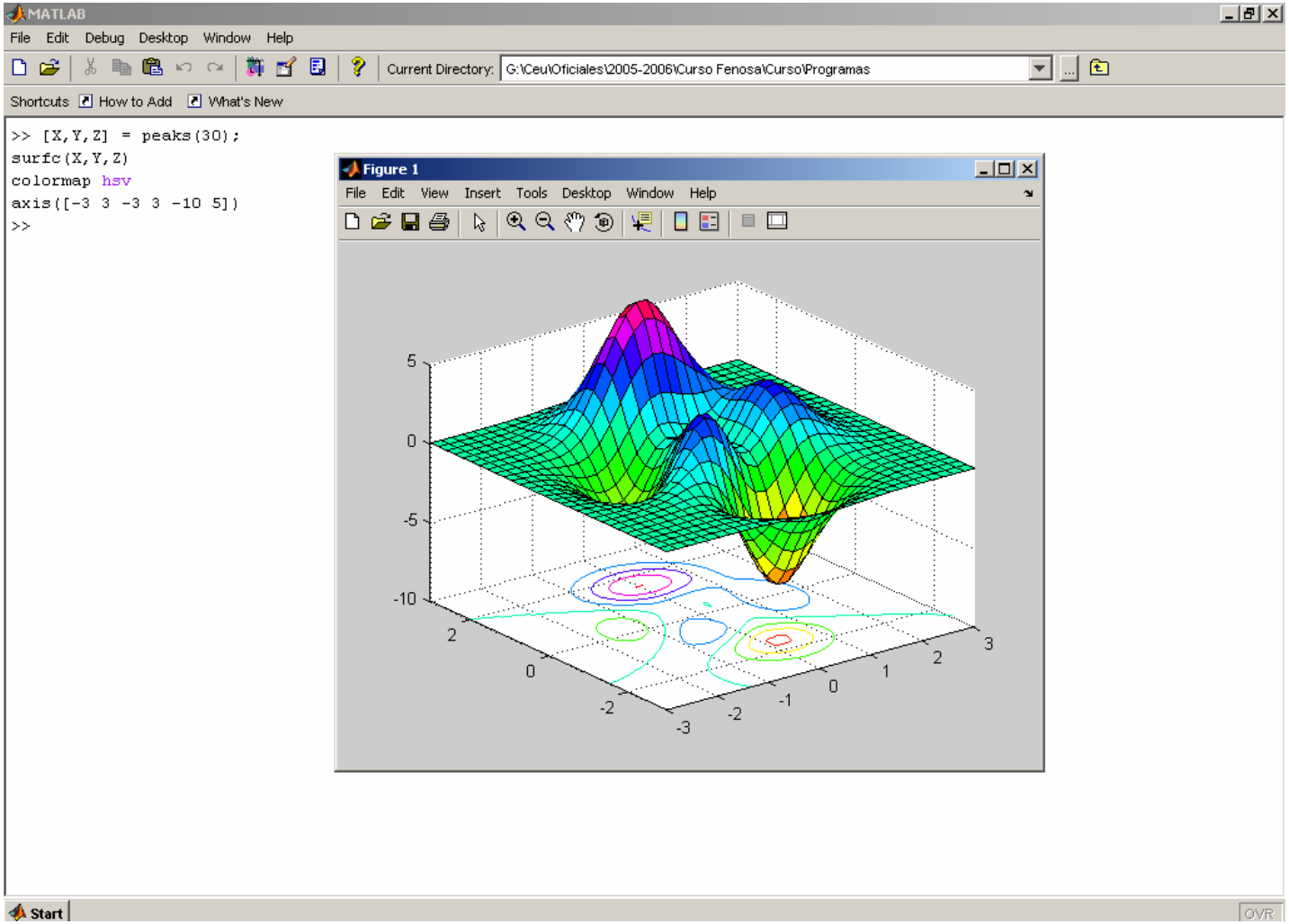
Gráficos tridimensionales



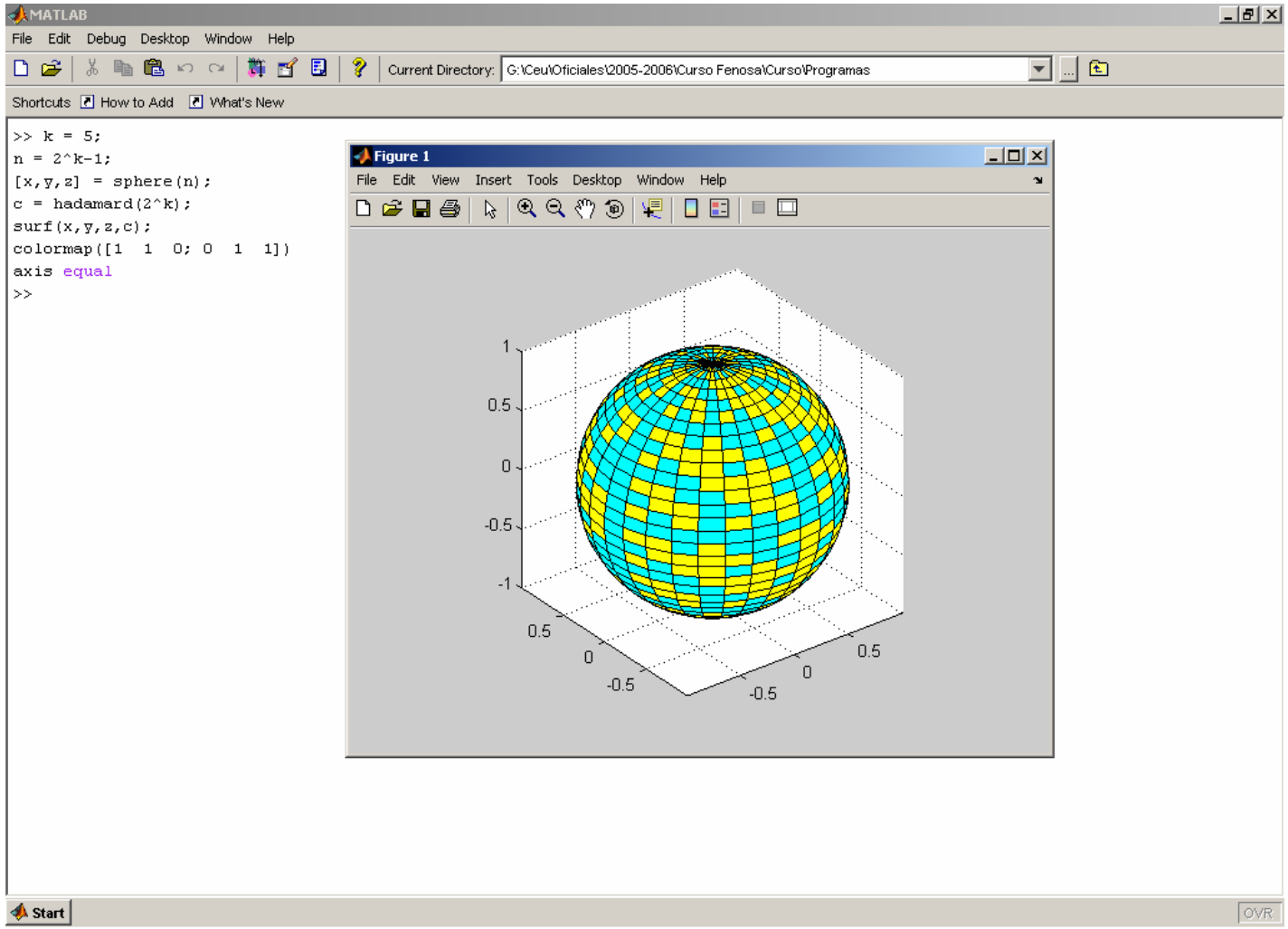
Gráficos tridimensionales



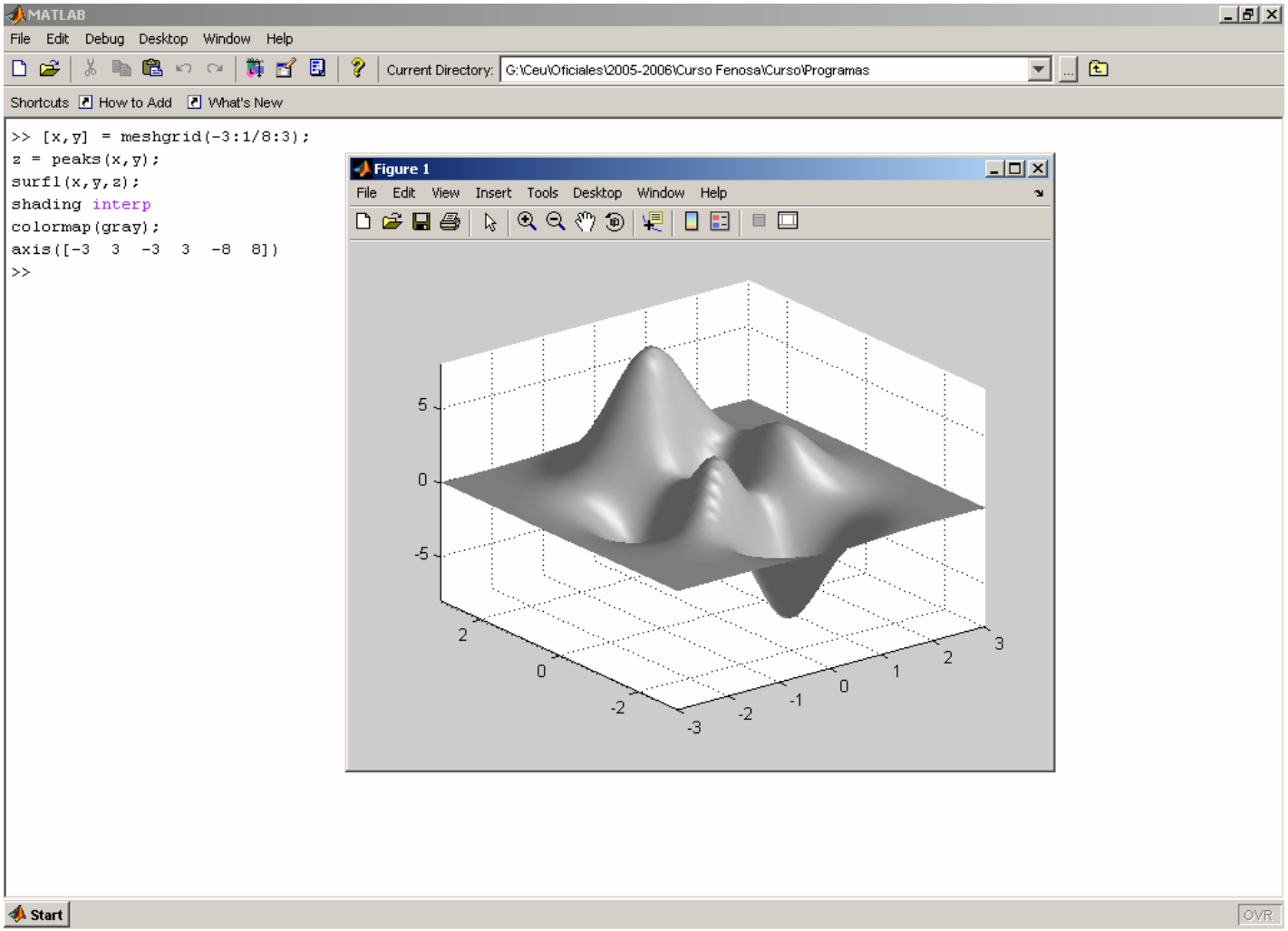
Gráficos tridimensionales



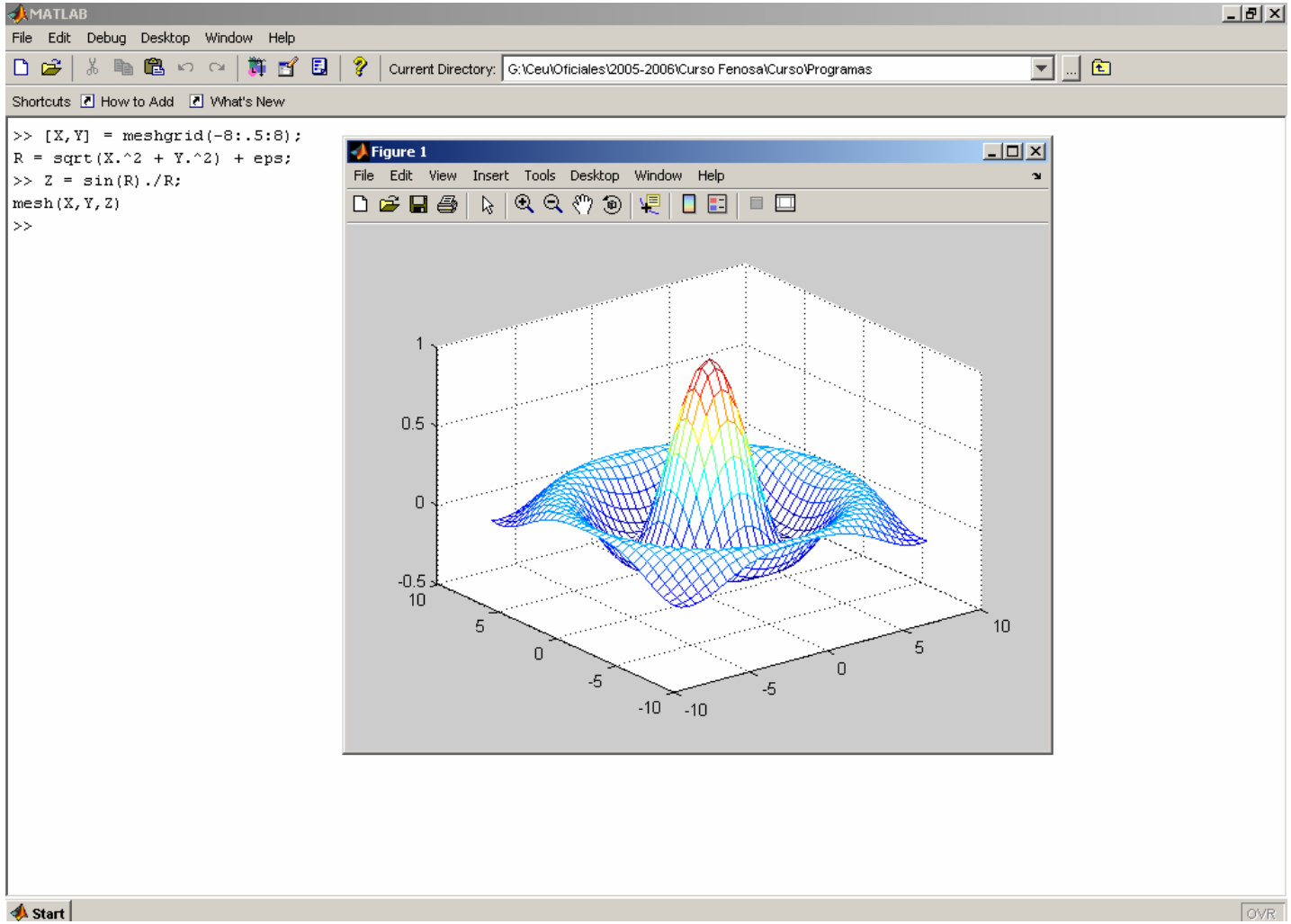
Gráficos tridimensionales



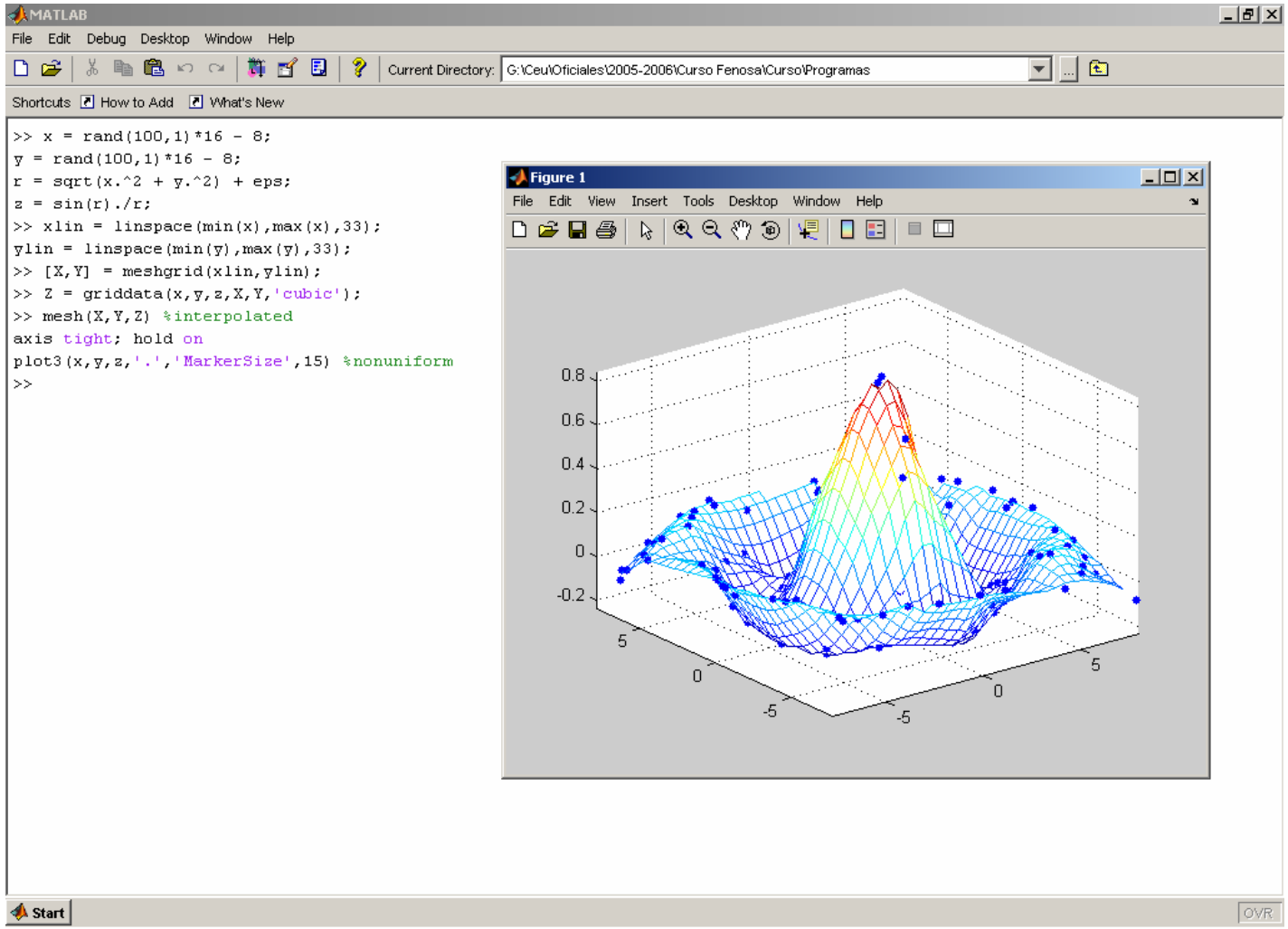
Gráficos tridimensionales



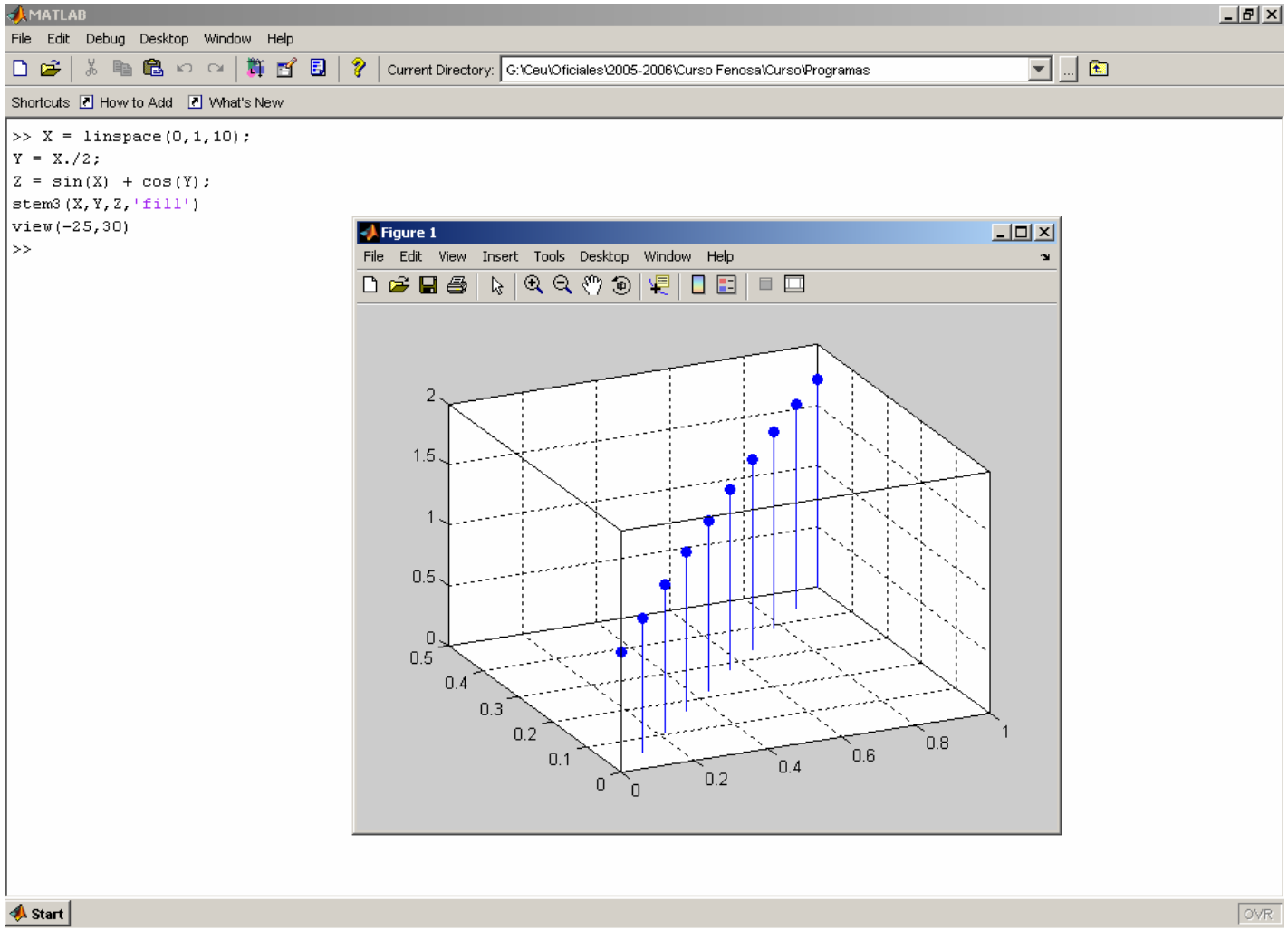
Gráficos tridimensionales



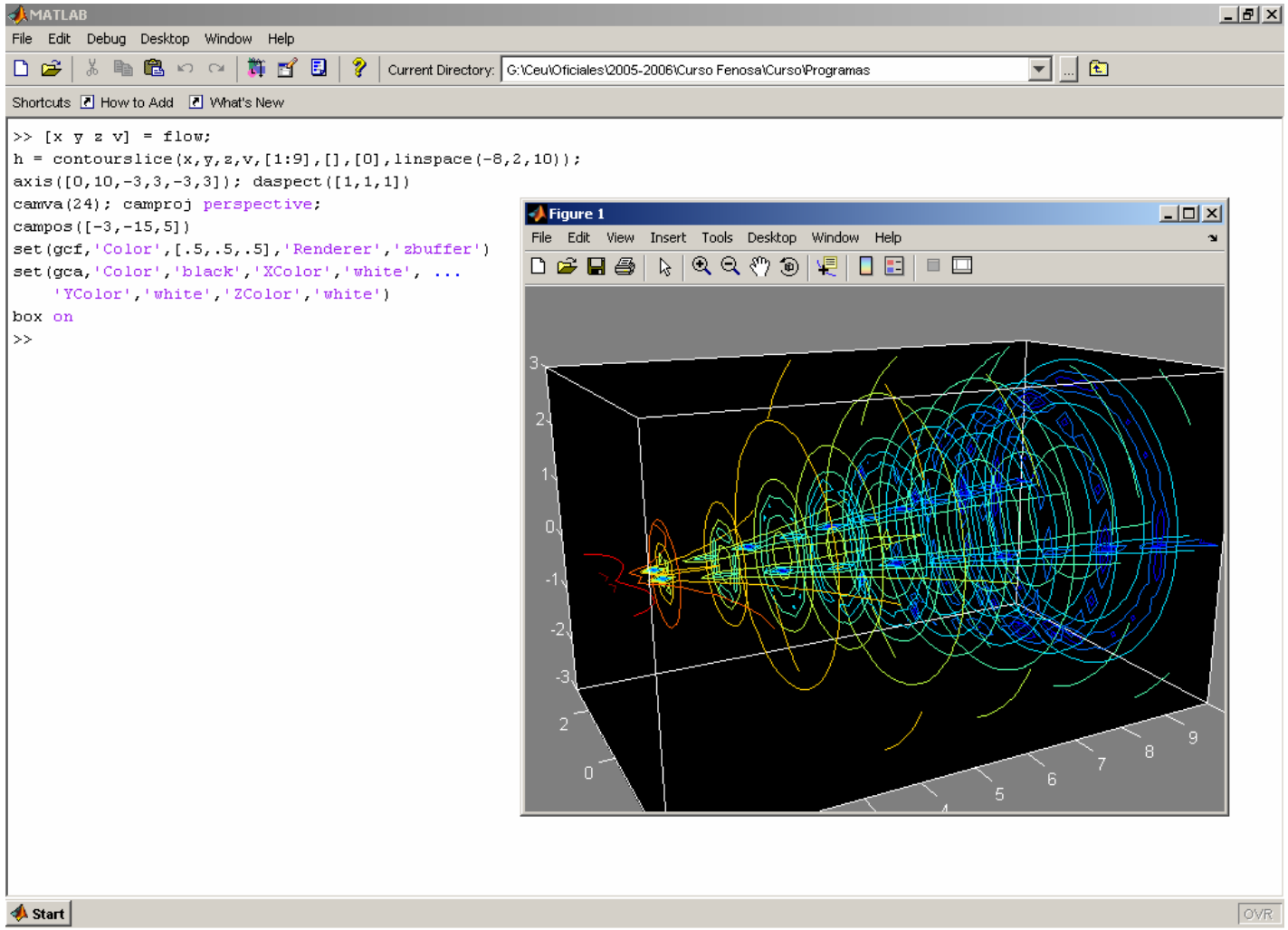
Gráficos tridimensionales



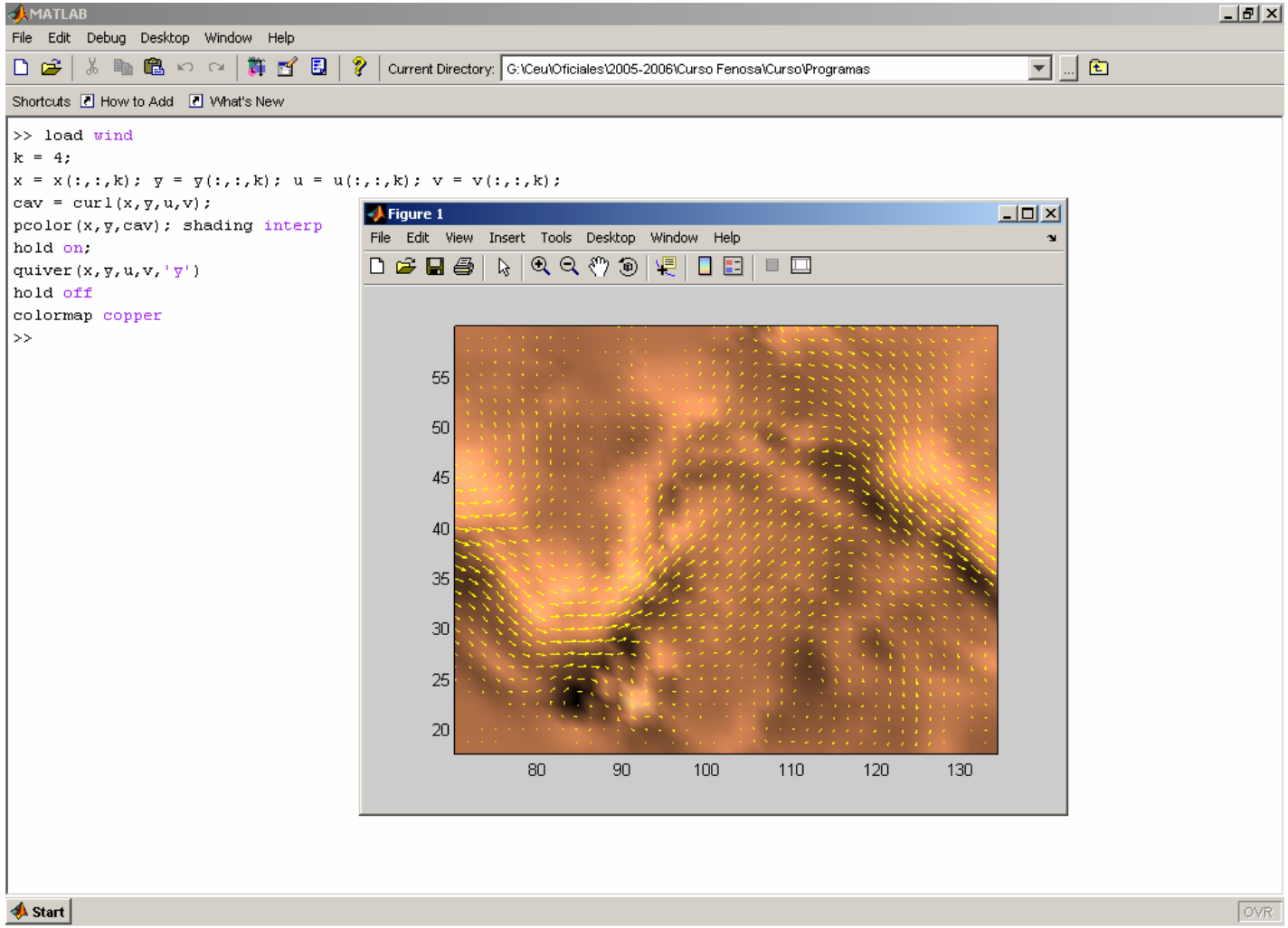
Gráficos tridimensionales



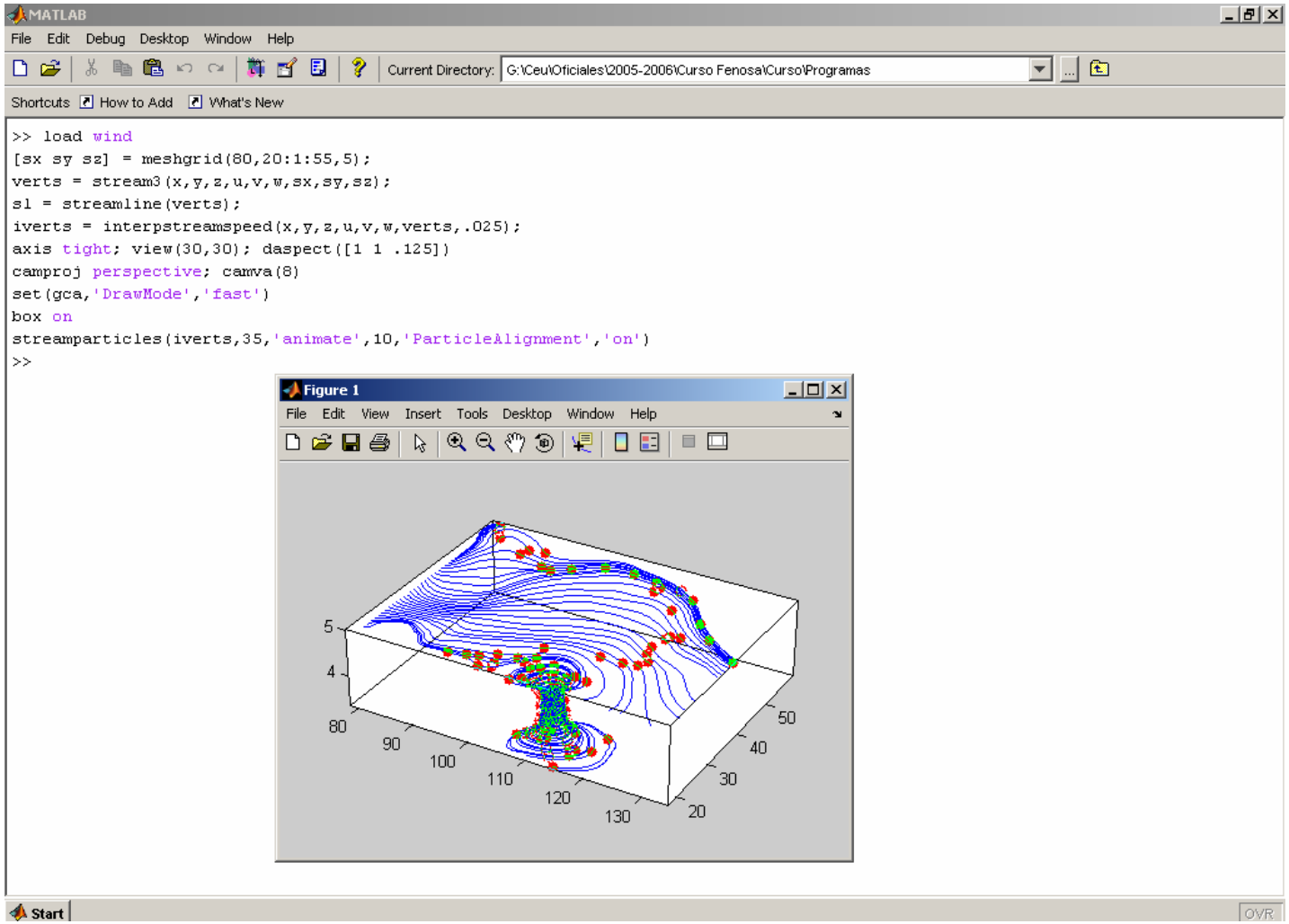
Gráficos tridimensionales



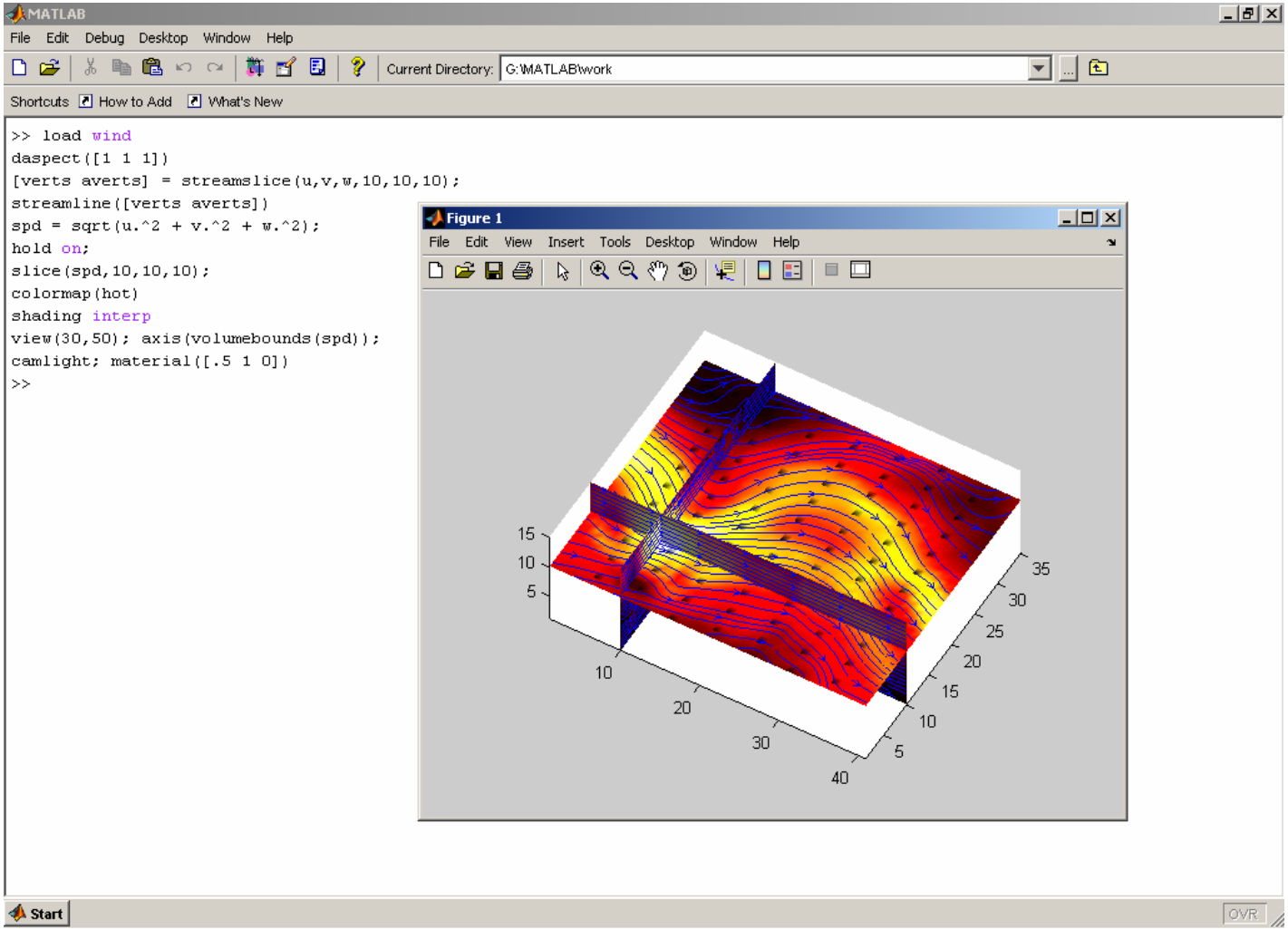
Gráficos tridimensionales



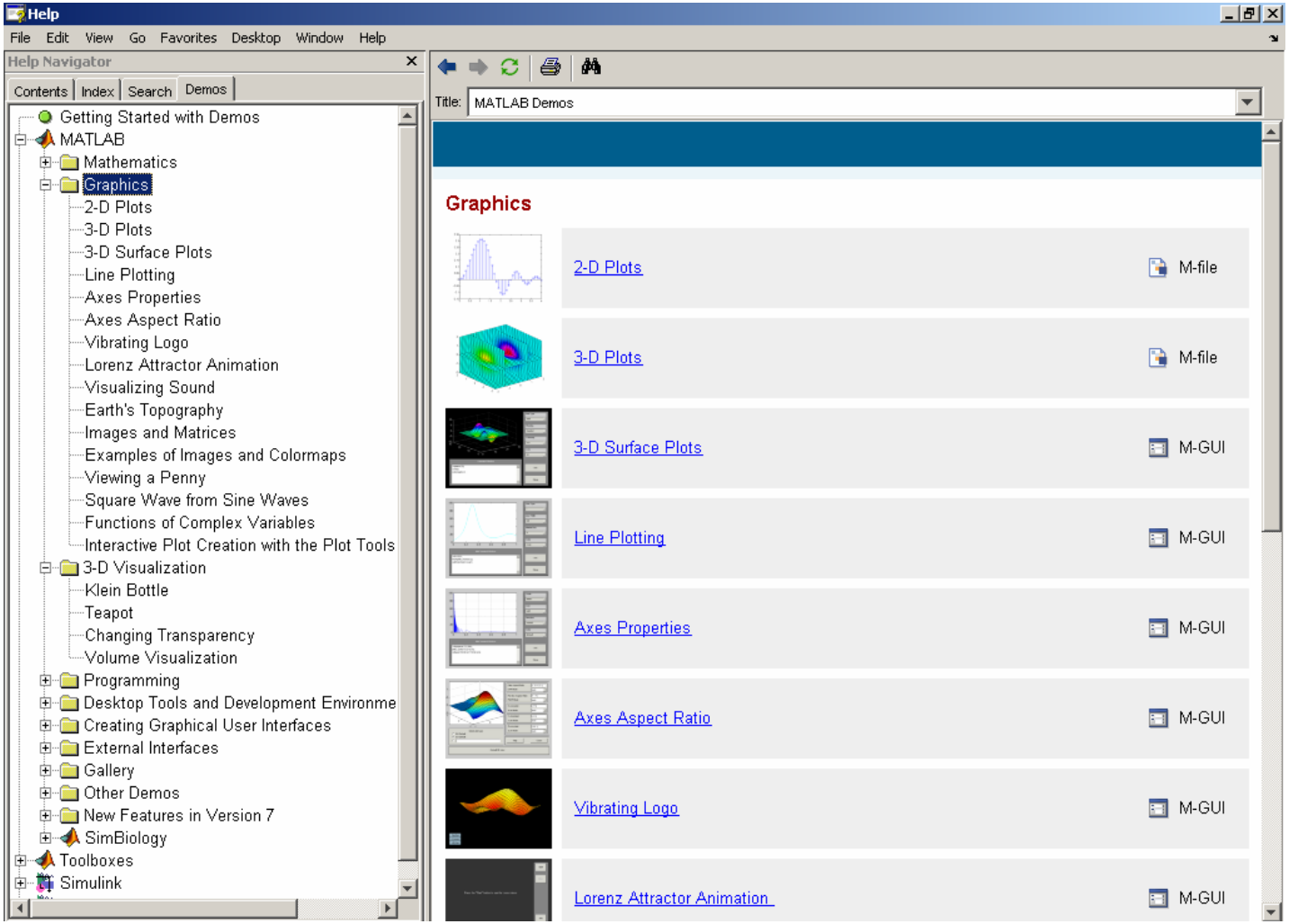
Gráficos tridimensionales



Gráficos tridimensionales

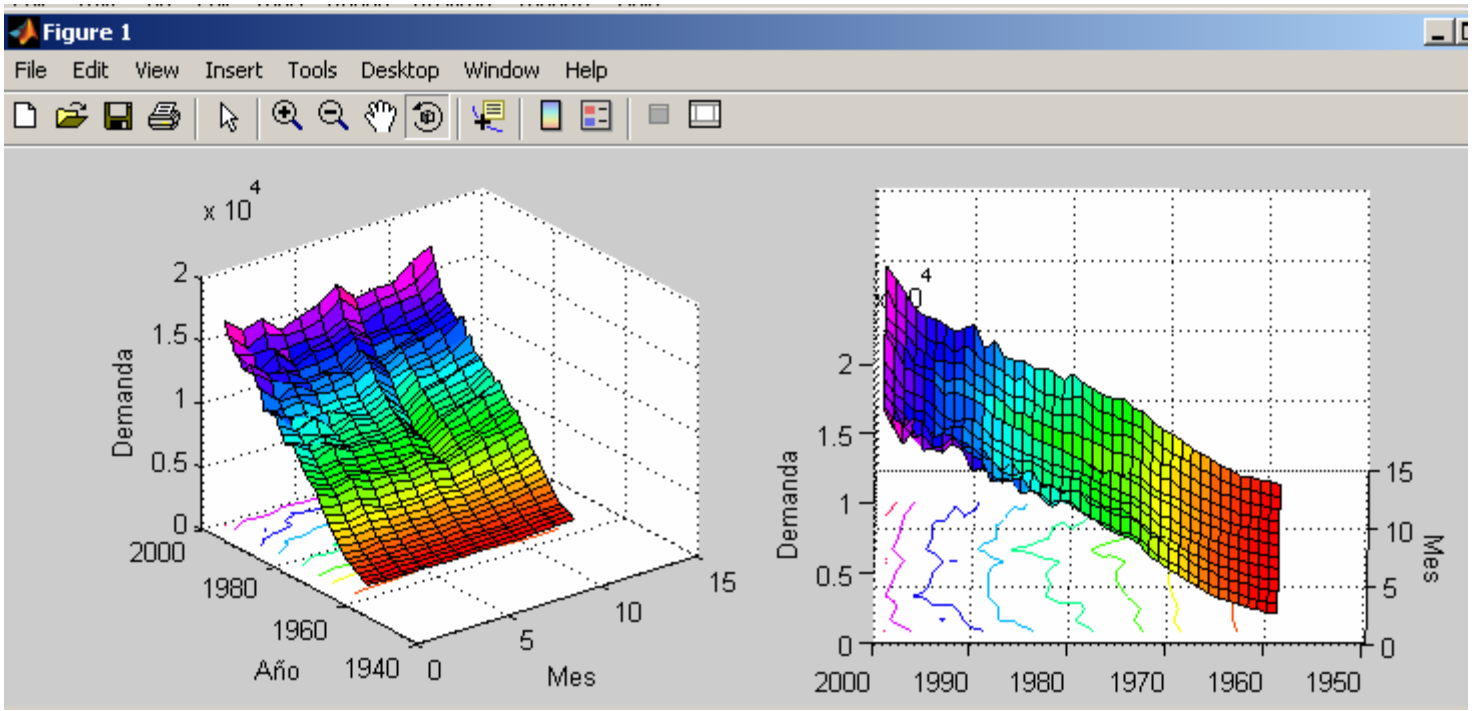


Demos


Help
File Edit View Go Favorites Desktop Window Help
Help Navigator
Contents Index Search Demos
Getting Started with Demos
MATLAB
Mathematics
Graphics
2-D Plots
3-D Plots
3-D Surface Plots
Line Plotting
Axes Properties
Axes Aspect Ratio
Vibrating Logo
Lorenz Attractor Animation
Visualizing Sound
Earth's Topography
Images and Matrices
Examples of Images and Colormaps
Viewing a Penny
Square Wave from Sine Waves
Functions of Complex Variables
Interactive Plot Creation with the Plot Tools
3-D Visualization
Klein Bottle
Teapot
Changing Transparency
Volume Visualization
Programming
Desktop Tools and Development Environme
Creating Graphical User Interfaces
External Interfaces
Gallery
Other Demos
New Features in Version 7
SimBiology
Toolboxes
Simulink
Title: MATLAB Demos
Graphics
2-D Plots M-file
3-D Plots M-file
3-D Surface Plots M-GUI
Line Plotting M-GUI
Axes Properties M-GUI
Axes Aspect Ratio M-GUI
Vibrating Logo M-GUI
Lorenz Attractor Animation M-GUI

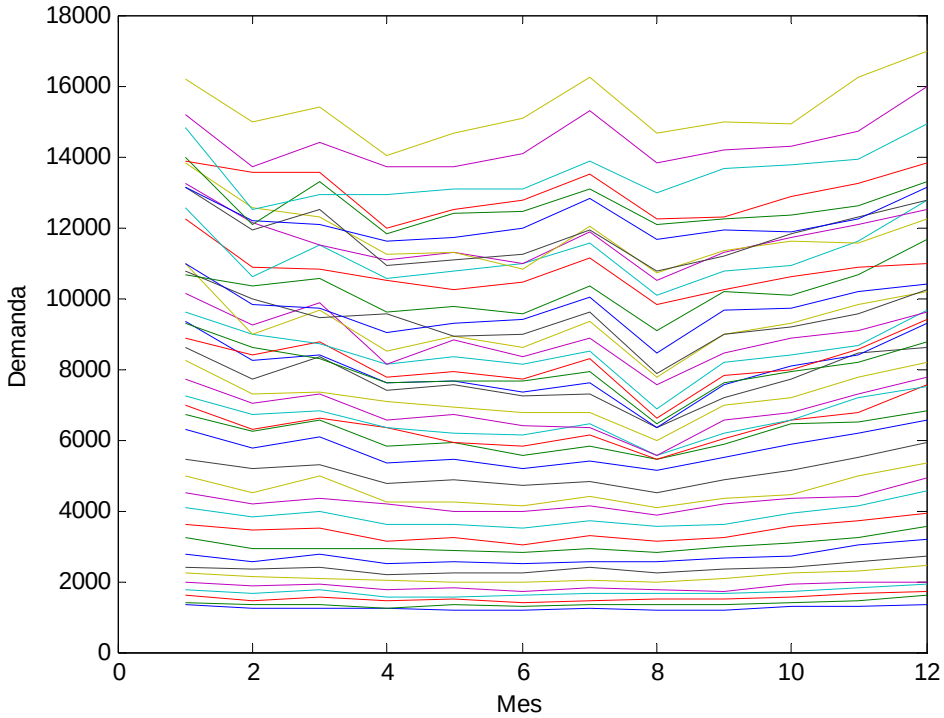
Ejercicio Final 1:

- Representar la demanda frente a los años y los meses



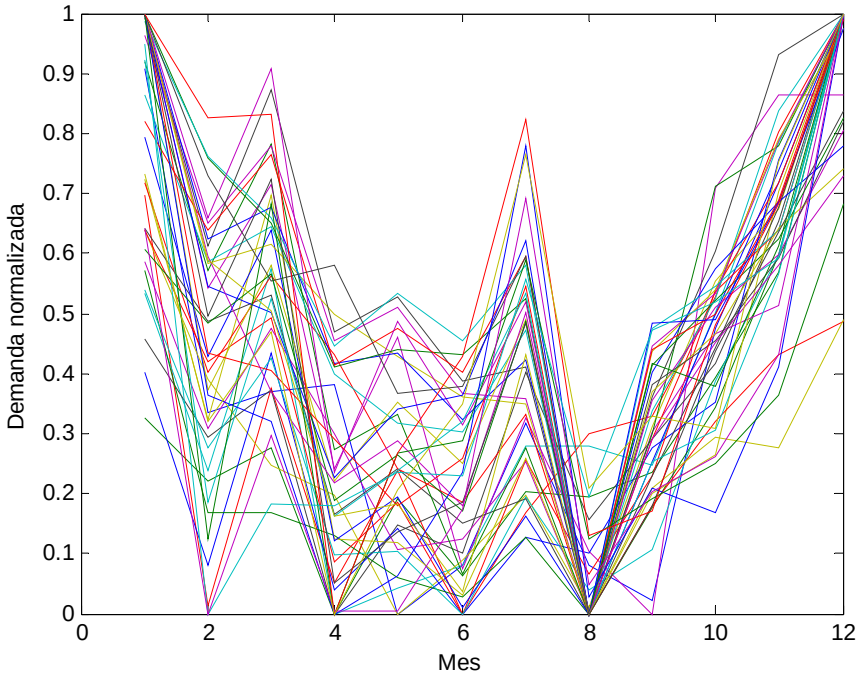
Ejercicio Final 2

- Representar en un mismo gráfico todos los consumos anuales



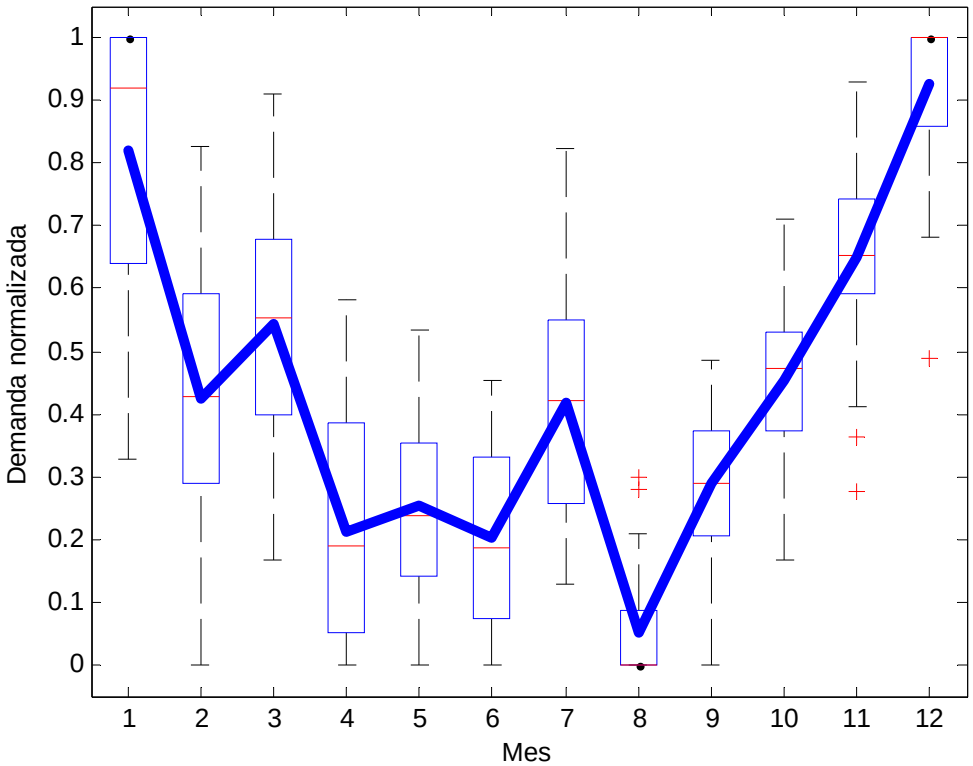
Ejercicio Final 3

- Representar en un mismo gráfico todos los consumos anuales normalizados de forma que todos estén entre 0 y 1



Ejercicio Final 4

- Representar un boxplot de los datos normalizados y superponer la media



CEU
Universidad
San Pablo

CURSO de UTILIZACIÓN PRÁCTICA de MATLAB

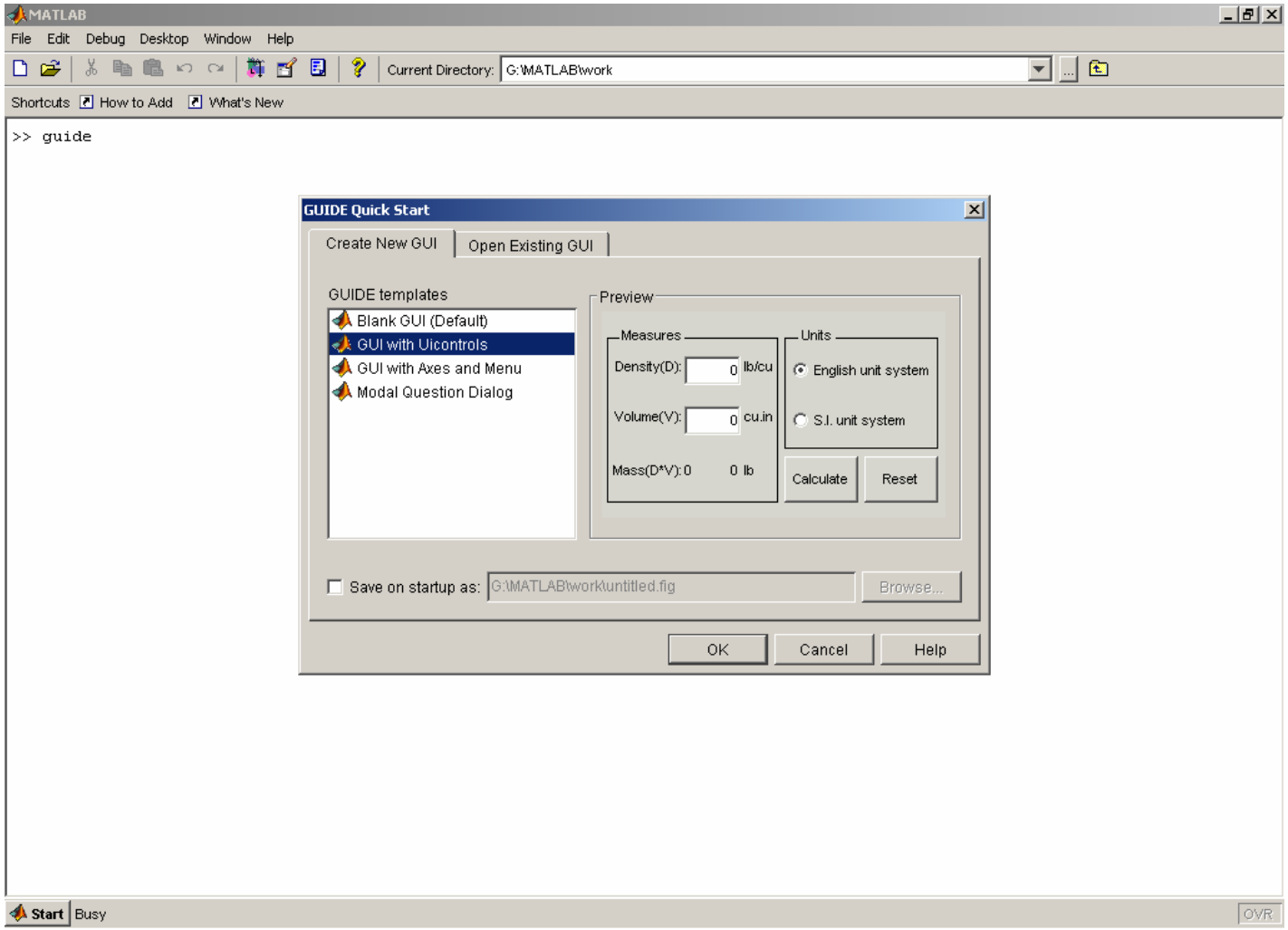
Sesión 4

Carlos Óscar Sánchez Sorzano, Ph.D.
Madrid, July 17th 2006

Cronograma del curso

- Día 1: Operaciones con matrices y vectores. Funciones de librería.
- Día 2: Otros tipos de datos en MATLAB. Programación en MATLAB.
- Día 3: Gráficos bidimensionales. Gráficos tridimensionales.
- **Día 4: Interfaces de usuario en MATLAB. Generación de programas autónomos**
- Día 5: Librerías de interés práctico
- Día 6: Interacción de MATLAB con Office y Visual Basic
- Día 7: Desarrollo de un proyecto

Graphical User Interfaces: GUIs



GUIs

Diseño del interfaz

Botón de ejecución: Salvar como Mi_primer_GUI

Interfaz ejecutándose

Código generado

Editor - G:\Ceu\Oficiales\2005-2006\Curso Fenosa\

File Edit Text Go Cell Tools Debug Desktop Wind

```
function varargout = Mi_primer_
```

Mi_primer_GUI.fig

File Edit View Layout Tools Help

Measures
Density(D): 0 lb/cu.in
Volume(V): 0 cu.in
Mass(D*V): 0 lb
Units
English Unit System
S.I. Unit System
Calculate
Reset

Mi_primer_GUI

Measures
Density(D): 10 kg/cu.m
Volume(V): 5 cu.m
Mass(D*V): 50 kg
Units
English Unit System
S.I. Unit System
Calculate
Reset

```
MI_PRIMER_GUI or raises
operty value pairs are
ion gets called.  An
operty application
gFcn via varargin.

allows only one

%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Mi_primer_GU

% Last Modified by GUIDE v2.5 23-Sep-2006 19:50:54

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Mi_primer_GUI_OpeningFcn, ...
                   'gui_OutputFcn',  @Mi_primer_GUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
```

Mi_primer_GUI Ln 1 Col 1 OVR

GUIs

Función Mi_primer_GUI

Editor - G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas\Mi_primer_GUI.m

File Edit Text Go Cell Tools Debug Desktop Window Help

```
function varargout = Mi_primer_GUI(varargin)
% MI_PRIMER_GUI M-file for Mi_primer_GUI.fig
%      MI_PRIMER_GUI, by itself, creates a new MI_PRIMER_GUI or raises the existing
%      singleton*.
%
%      H = MI_PRIMER_GUI returns the handle to a new MI_PRIMER_GUI or the handle to
%      the existing singleton*.
%
%      MI_PRIMER_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in MI_PRIMER_GUI.M with the given input arguments.
%
%      MI_PRIMER_GUI('Property','Value',...) creates a new MI_PRIMER_GUI or raises
%      the existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before Mi_primer_GUI_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to Mi_primer_GUI_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Mi_primer_GUI

% Last Modified by GUIDE v2.5 23-Sep-2006 19:50:54

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Mi_primer_GUI_OpeningFcn, ...
                   'gui_OutputFcn',  @Mi_primer_GUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
```

Mi_primer_GUI / Mi_primer_GUI_O... Ln 48 Col 2 OVR

GUIs

Editor - G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas\Mi_primer_GUI.m

File Edit Text Go Cell Tools Debug Desktop Window Help

```
12  %       MI_PRIMER_GUI('Property','Value',...) creates a new MI_PRIMER_GUI or raises
13  %       the existing singleton*.  Starting from the left, property value pairs are
14  %       applied to the GUI before Mi_primer_GUI_OpeningFunction gets called.  An
15  %       unrecognized property name or invalid value makes property application
16  %       stop.  All inputs are passed to Mi_primer_GUI_OpeningFcn via varargin.
17  %
18  %       *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
19  %       instance to run (singleton)".
20  %
21  % See also: GUIDE, GUIDATA, GUIHANDLES
22
23  % Edit the above text to modify the response to help Mi_primer_GUI
24
25  % Last Modified by GUIDE v2.5 23-Sep-2006 19:50:54
26
27  % Begin initialization code - DO NOT EDIT
28  gui_Singleton = 1;
29  gui_State = struct('gui_Name',       mfilename, ...
30                     'gui_Singleton',  gui_Singleton, ...
31                     'gui_OpeningFcn', @Mi_primer_GUI_OpeningFcn, ...
32                     'gui_OutputFcn',  @Mi_primer_GUI_OutputFcn, ...
33                     'gui_LayoutFcn',  [] , ...
34                     'gui_Callback',   []);
35  if nargin && ischar(varargin{1})
36      gui_State.gui_Callback = str2func(varargin{1});
37  end
38
39  if nargout
40      [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
41  else
42      gui_mainfcn(gui_State, varargin{:});
43  end
44  % End initialization code - DO NOT EDIT
45
```

Mi_primer_GUI | Ln 23 | Col 67 | OVR

Código de inicialización. Generado automáticamente

GUIs

Editor - G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas\Mi_primer_GUI.m

```
42      gui_mainfcn(gui_State, varargin{:});
43  end
44  % End initialization code - DO NOT EDIT
45
46  % --- Executes just before Mi_primer_GUI is made visible.
47  function Mi_primer_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
48  % This function has no output args, see OutputFcn.
49  % hObject    handle to figure
50  % eventdata  reserved - to be defined in a future version of MATLAB
51  % handles    structure with handles and user data (see GUIDATA)
52  % varargin   command line arguments to Mi_primer_GUI (see VARARGIN)
53
54  % Choose default command line output for Mi_primer_GUI
55  handles.output = hObject;
56
57  % Update handles structure
58  guidata(hObject, handles);
59
60  initialize_gui(hObject, handles, false);
61
62  % UIWAIT makes Mi_primer_GUI wait for user response (see UIRESUME)
63  % uiwait(handles.figure1);
64
65
66  % --- Outputs from this function are returned to the command line.
67  function varargout = Mi_primer_GUI_OutputFcn(hObject, eventdata, handles)
68  % varargout  cell array for returning output args (see VARARGOUT);
69  % hObject    handle to figure
70  % eventdata  reserved - to be defined in a future version of MATLAB
71  % handles    structure with handles and user data (see GUIDATA)
72
73  % Get default command line output from handles structure
74  varargout{1} = handles.output;
75
```

Función llamada antes de abrir el interfaz

Función llamada al cerrar el interfaz

GUIs

Editor - G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas\Mi_primer_GUI.m

```
172 -  else
173 -      set(handles.text4, 'String', 'kg/cu.m');
174 -      set(handles.text5, 'String', 'cu.m');
175 -      set(handles.text6, 'String', 'kg');
176 -  end
177
178    % --------------------------------------------------------------------
179    function initialize_gui(fig_handle, handles, isreset)
180    % If the metricdata field is present and the reset flag is false, it means
181    % we are we are just re-initializing a GUI by calling it from the cmd line
182    % while it is up. So, bail out as we dont want to reset the data.
183 -  if isfield(handles, 'metricdata') && ~isreset
184 -      return;
185 -  end
186
187 -  handles.metricdata.density = 0;
188 -  handles.metricdata.volume  = 0;
189
190 -  set(handles.density, 'String', handles.metricdata.density);
191 -  set(handles.volume,  'String', handles.metricdata.volume);
192 -  set(handles.mass, 'String', 0);
193
194 -  set(handles.unitgroup, 'SelectedObject', handles.english);
195
196 -  set(handles.text4, 'String', 'lb/cu.in');
197 -  set(handles.text5, 'String', 'cu.in');
198 -  set(handles.text6, 'String', 'lb');
199
200    % Update handles structure
201 -  guidata(handles.figure1, handles);
202
203
204
205
```

'initialize' found at 60:1 | Mi_primer_GUI / Mi_primer_GUI_O... | Ln 60 | Col 11 | OVR

Función de inicialización

GUIs

Editor - G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas\Mi_primer_GUI.m

```
% --- Executes during object creation, after setting all properties.
function density_CreateFcn(hObject, eventdata, handles)
% hObject    handle to density (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background, change
%       'usewhitebg' to 0 to use default.  See ISPC and COMPUTER.
usewhitebg = 1;
if usewhitebg
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function density_Callback(hObject, eventdata, handles)
% hObject    handle to density (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of density as text
%        str2double(get(hObject,'String')) returns contents of density as a double
density = str2double(get(hObject, 'String'));
if isnan(density)
    set(hObject, 'String', 0);
    errordlg('Input must be a number','Error');
end

% Save the new density value
handles.metricdata.density = density;
guidata(hObject,handles)
```

Función llamada al crear el objeto density

Función llamada al modificar el objeto density

GUIs

Editor - G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas\Mi_primer_GUI.m

```
% --- Executes during object creation, after setting all properties.
function volume_CreateFcn(hObject, eventdata, handles)
% hObject    handle to volume (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background, change
%       'usewhitebg' to 0 to use default.  See ISPC and COMPUTER.
usewhitebg = 1;
if usewhitebg
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function volume_Callback(hObject, eventdata, handles)
% hObject    handle to volume (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of volume as text
%        str2double(get(hObject,'String')) returns contents of volume as a double
volume = str2double(get(hObject, 'String'));
if isnan(volume)
    set(hObject, 'String', 0);
    errordlg('Input must be a number','Error');
end

% Save the new volume value
handles.metricdata.volume = volume;
guidata(hObject,handles)
```

Función llamada al crear el objeto volume

Función llamada al modificar el objeto volume

GUIs

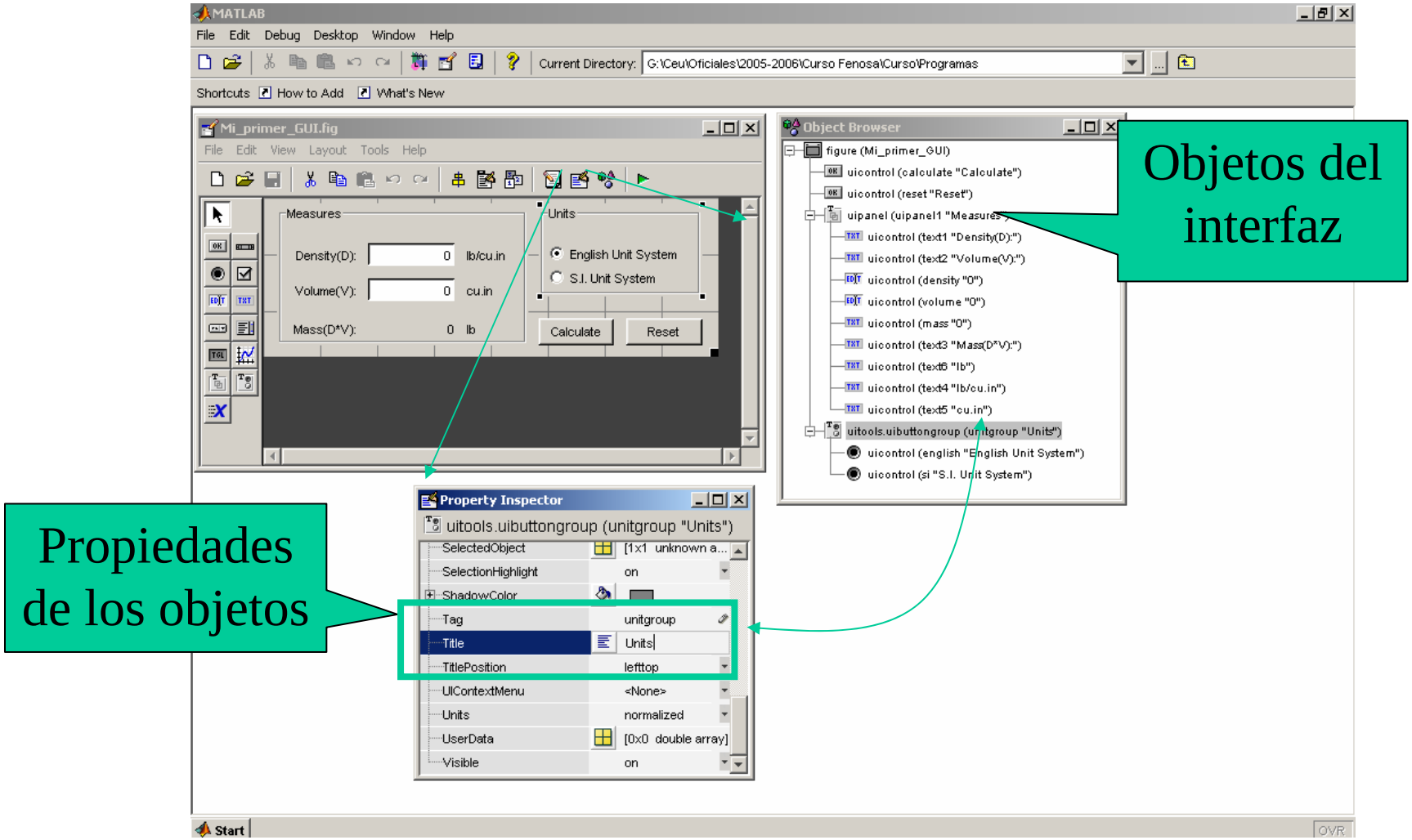
Editor - G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas\Mi_primer_GUI.m

```
% --- Executes on button press in calculate.
function calculate_Callback(hObject, eventdata, handles)
% hObject    handle to calculate (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

mass = handles.metricdata.density * handles.metricdata.volume;
set(handles.mass, 'String', mass);

% --- Executes on button press in reset.
function reset_Callback(hObject, eventdata, handles)
% hObject    handle to reset (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

initialize_gui(gcbf, handles, true);

% --------------------------------------------------------------------
function unitgroup_SelectionChangeFcn(hObject, eventdata, handles)
% hObject    handle to unitgroup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

if (hObject == handles.english)
    set(handles.text4, 'String', 'lb/cu.in');
    set(handles.text5, 'String', 'cu.in');
    set(handles.text6, 'String', 'lb');
else
    set(handles.text4, 'String', 'kg/cu.m');
    set(handles.text5, 'String', 'cu.m');
    set(handles.text6, 'String', 'kg');
end

% --------------------------------------------------------------------
```

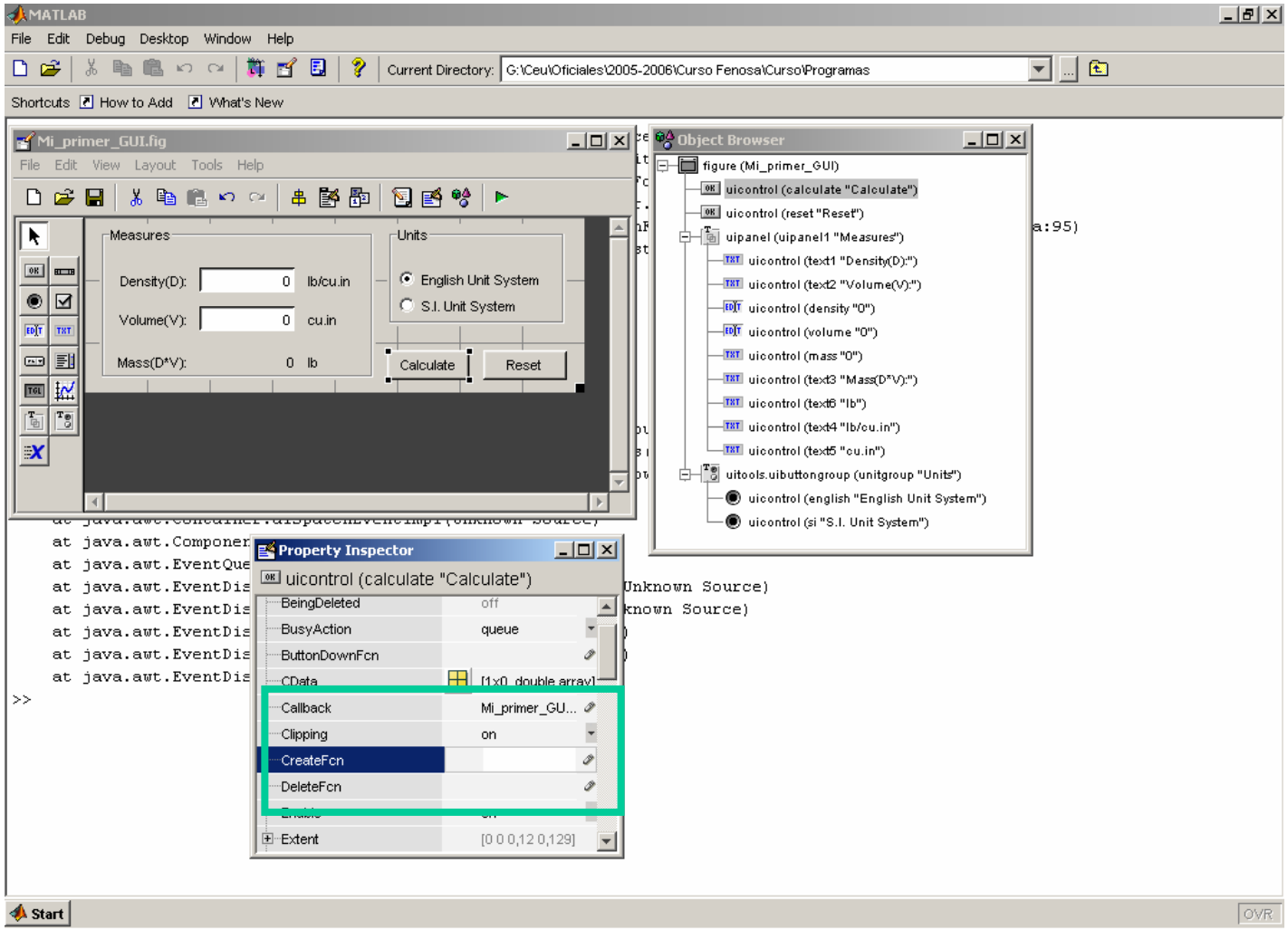
Funciones llamadas al pulsar “Calculate” o “Reset”

Función llamada cuando cambian los botones de unidades

GUIs

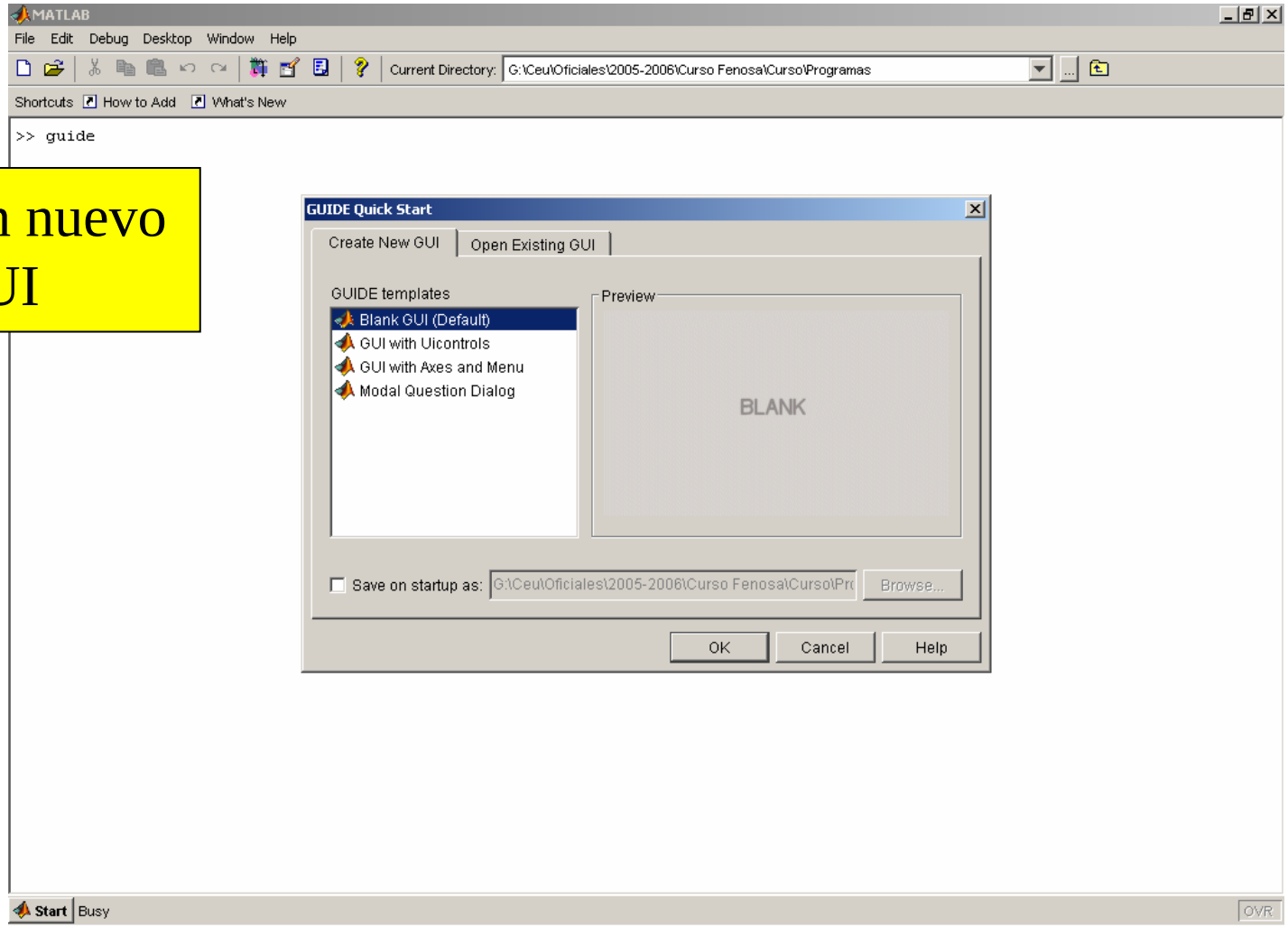

MATLAB
File Edit Debug Desktop Window Help
Current Directory: G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas
Shortcuts How to Add What's New
Mi_primer_GUI.fig
File Edit View Layout Tools Help
Measures
Density(D): 0 lb/cu.in
Volume(V): 0 cu.in
Mass(D*V): 0 lb
Units
English Unit System
S.I. Unit System
Calculate
Reset
Object Browser
figure (Mi_primer_GUI)
uicontrol (calculate "Calculate")
uicontrol (reset "Reset")
uipanel (uipanel1 "Measures")
uicontrol (text1 "Density(D):")
uicontrol (text2 "Volume(V):")
uicontrol (density "0")
uicontrol (volume "0")
uicontrol (mass "0")
uicontrol (text3 "Mass(D*V):")
uicontrol (text6 "lb")
uicontrol (text4 "lb/cu.in")
uicontrol (text5 "cu.in")
uitools.uibuttongroup (unitgroup "Units")
uicontrol (english "English Unit System")
uicontrol (si "S.I. Unit System")
Objetos del interfaz
Property Inspector
uitools.uibuttongroup (unitgroup "Units")
SelectedObject [1x1 unknown a...
SelectionHighlight on
ShadowColor
Tag unitgroup
Title Units
TitlePosition lefttop
UIContextMenu <None>
Units normalized
UserData [0x0 double array]
Visible on
Propiedades de los objetos
Start
OVR

GUIs


MATLAB
File Edit Debug Desktop Window Help
Current Directory: G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas
Shortcuts How to Add What's New
Mi_primer_GUI.fig
File Edit View Layout Tools Help
Measures
Density(D): 0 lb/cu.in
Volume(V): 0 cu.in
Mass(D*V): 0 lb
Units
English Unit System
S.I. Unit System
Calculate
Reset
Object Browser
figure (Mi_primer_GUI)
uicontrol (calculate "Calculate")
uicontrol (reset "Reset")
uipanel (uipanel1 "Measures")
uicontrol (text1 "Density(D):")
uicontrol (text2 "Volume(V):")
uicontrol (density "0")
uicontrol (volume "0")
uicontrol (mass "0")
uicontrol (text3 "Mass(D*V):")
uicontrol (text6 "lb")
uicontrol (text4 "lb/cu.in")
uicontrol (text5 "cu.in")
uitools.uibuttongroup (unitgroup "Units")
uicontrol (english "English Unit System")
uicontrol (si "S.I. Unit System")
Property Inspector
uicontrol (calculate "Calculate")
BeingDeleted off
BusyAction queue
ButtonDownFcn
CData [1x0 double array]
Callback Mi_primer_GU...
Clipping on
CreateFcn
DeleteFcn
Extent [0 0 0,12 0,129]
at java.awt.Componer
at java.awt.EventQue
at java.awt.EventDis
Unknown Source)
known Source)
>>
Start
OVR

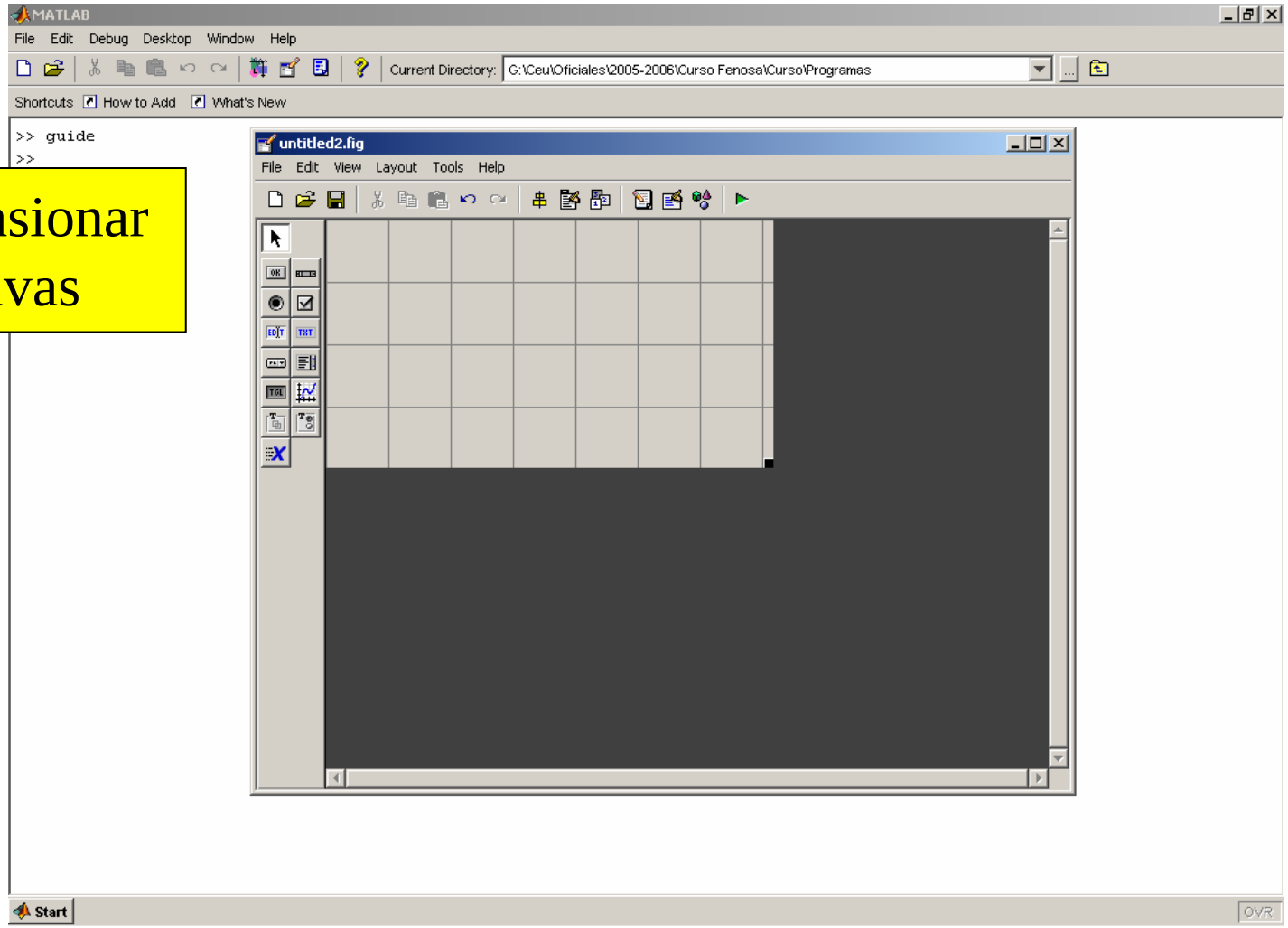
GUIs: Creación

Crear un nuevo GUI



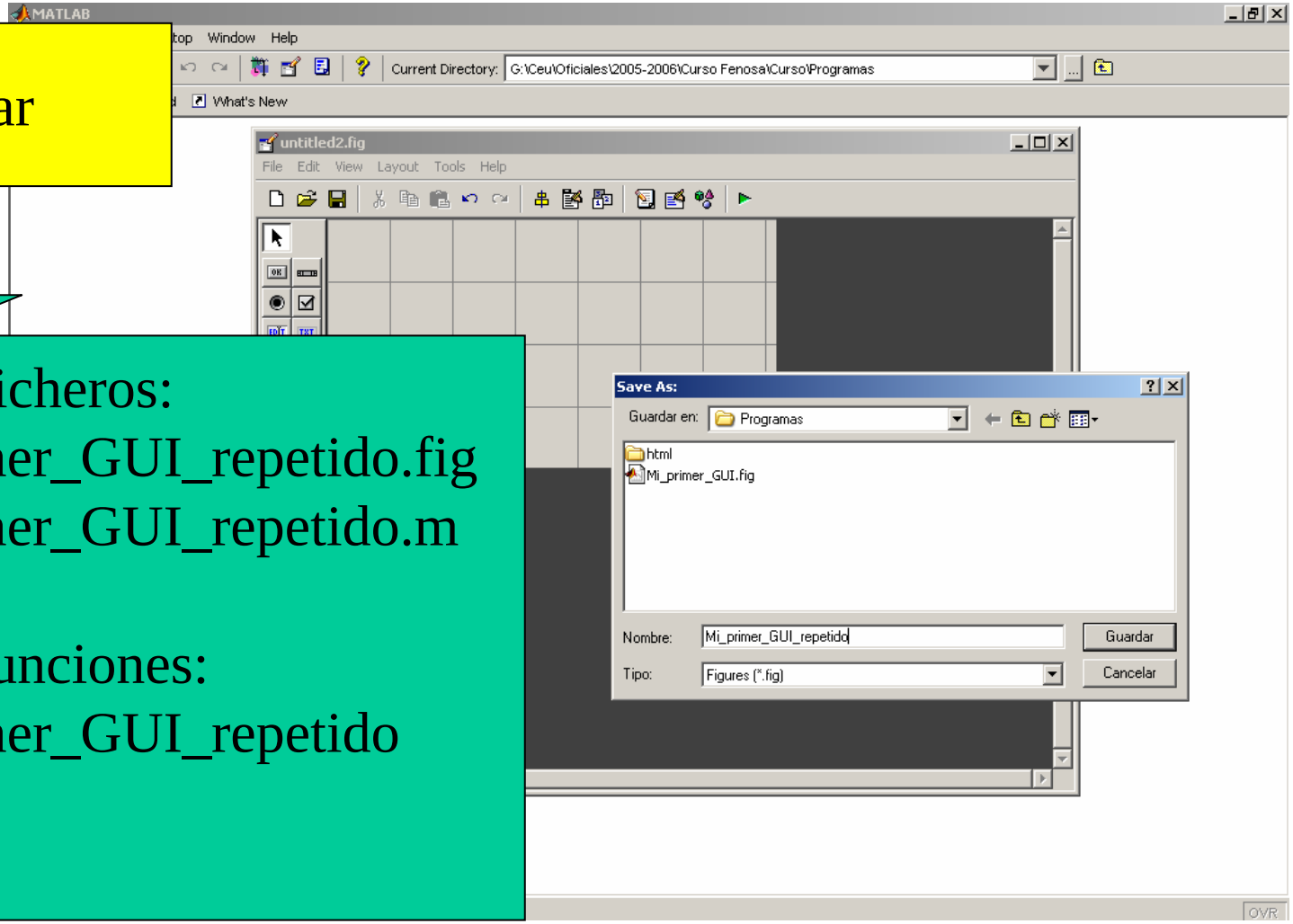
GUIs: Creación

Redimensionar el canvas



GUIs: Creación

Salvar



Crea los ficheros:

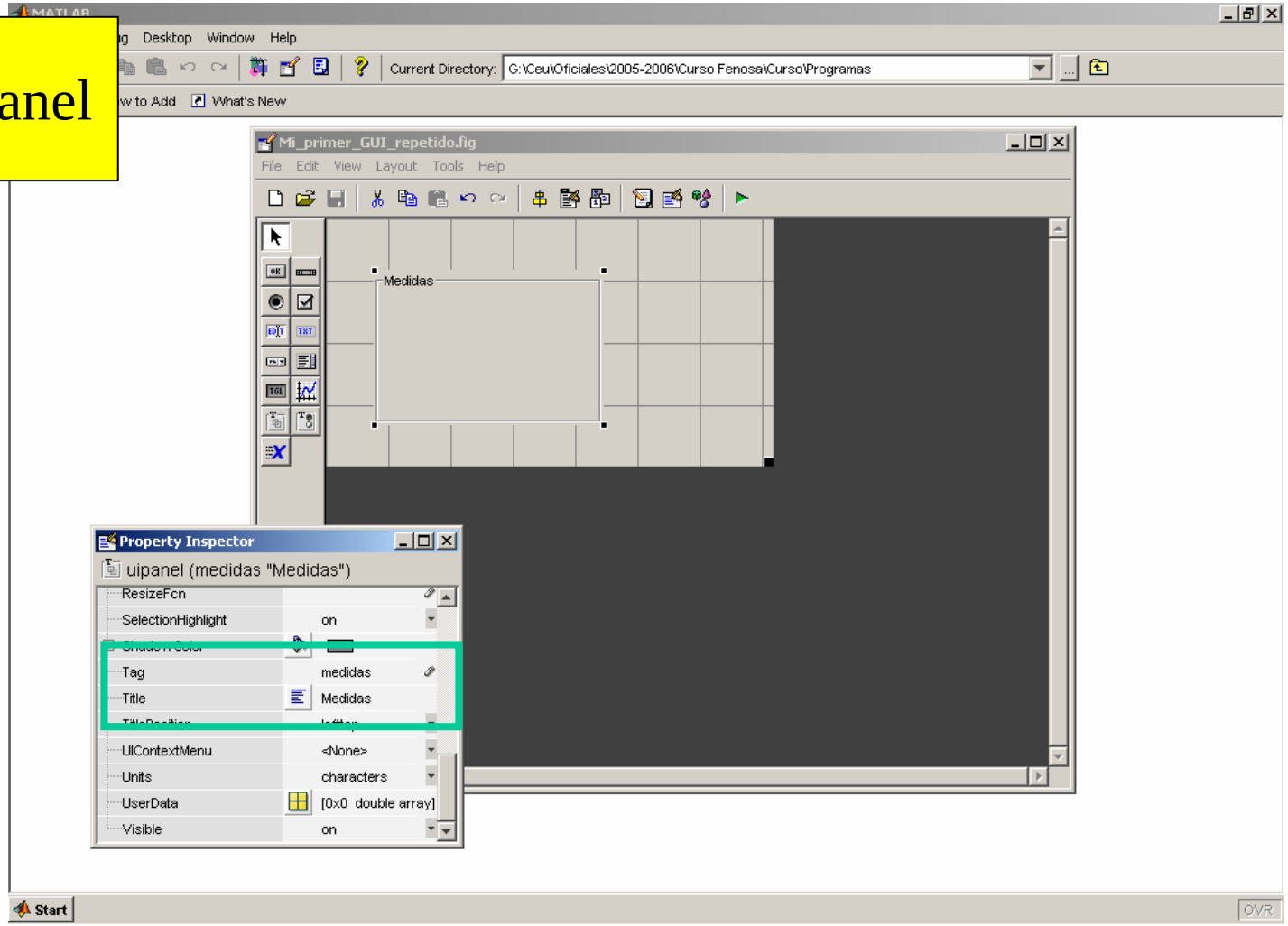
- Mi_primer_GUI_repetido.fig
- Mi_primer_GUI_repetido.m

Crea las funciones:

- Mi_primer_GUI_repetido
- Opening
- Closing

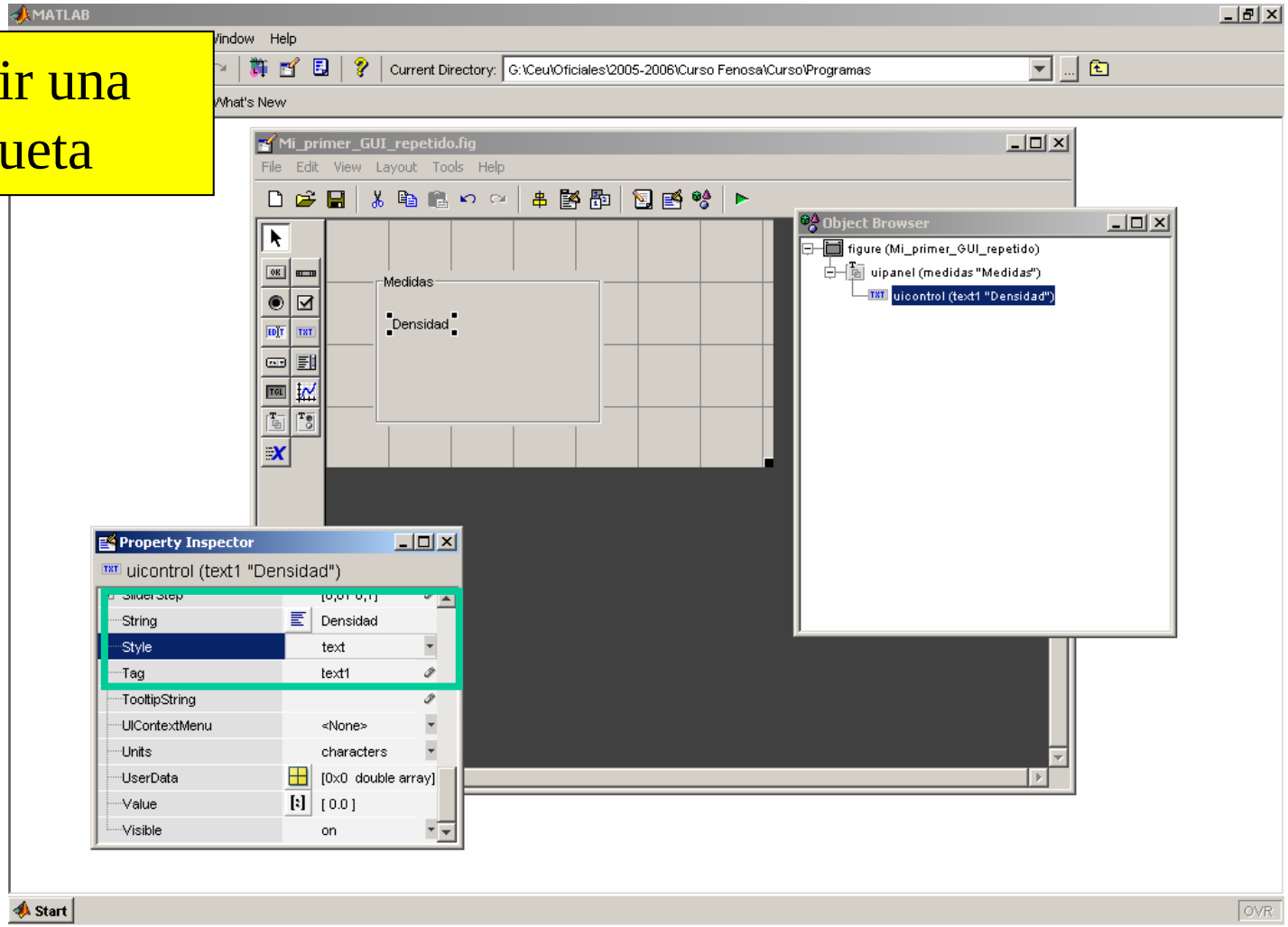
GUIs: Creación

Añadir un panel



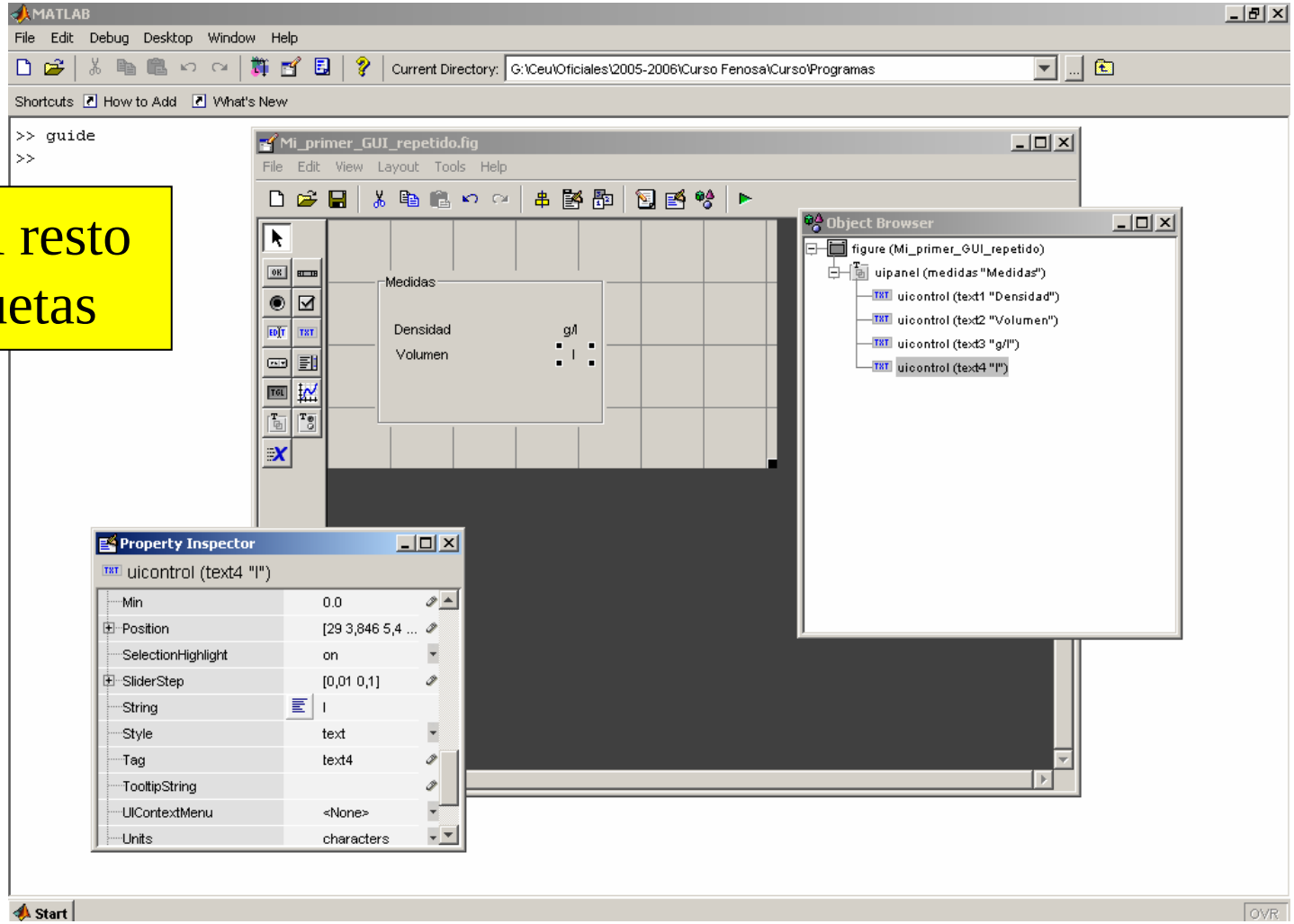
GUIs: Creación

Añadir una etiqueta



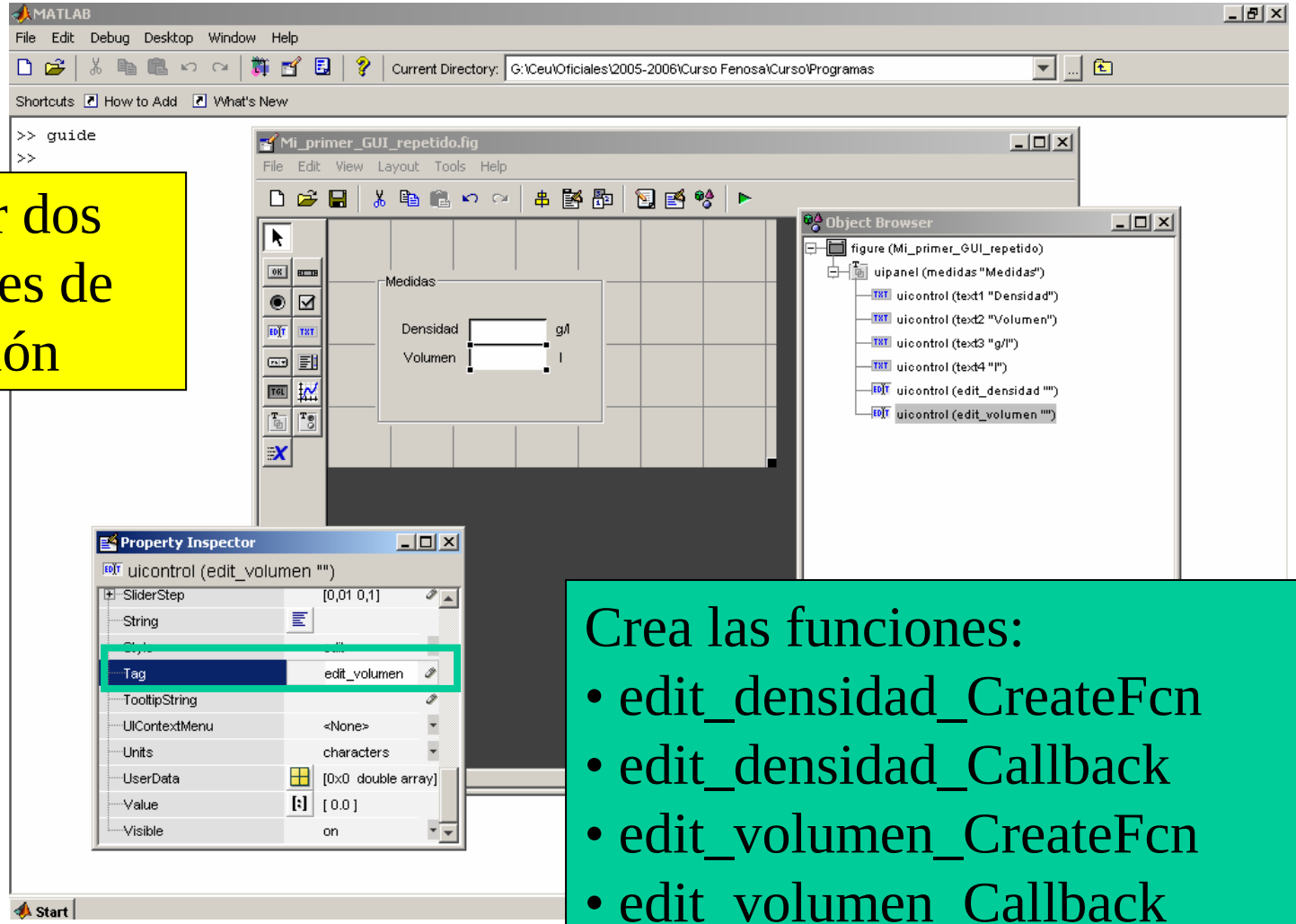
GUIs: Creación

Añadir el resto de etiquetas



GUIs: Creación

Añadir dos variables de edición



Crea las funciones:

- edit_densidad_CreateFcn
- edit_densidad_Callback
- edit_volumen_CreateFcn
- edit_volumen_Callback

GUIs: Creación

Editor - G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas\Mi_primer_GUI_repetido.m*

```
77   function edit_densidad_Callback(hObject, eventdata, handles)
78   % hObject    handle to edit_densidad (see GCBO)
79   % eventdata  reserved - to be defined in a future version of MATLAB
80   % handles    structure with handles and user data (see GUIDATA)
81
82   % Hints: get(hObject,'String') returns contents of edit_densidad as text
83   %        str2double(get(hObject,'String')) returns contents of edit_densidad as a double
84
85   % Recoger el valor del interfaz
86 - densidad = str2double(get(hObject, 'String'));
87 - if isnan(density)
88 -     set(hObject, 'String', 0);
89 -     errordlg('La entrada debe ser un número','Error');
90 - end
91
92   % Guardar el nuevo valor en handles
93 - handles.densidad = densidad;
94 - guidata(hObject,handles)
95
96   % --- Executes during object creation, after setting all properties.
97   function edit_densidad_CreateFcn(hObject, eventdata, handles)
98   % hObject    handle to edit_densidad (see GCBO)
99   % eventdata  reserved - to be defined in a future version of MATLAB
100  % handles    empty - handles not created until after all CreateFcns called
101
102  % Hint: edit controls usually have a white background on Windows.
103  %       See ISPC and COMPUTER.
104 - if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
105 -     set(hObject,'BackgroundColor','white');
106 - end
```

Modificar la función Callback

GUIs: Creación

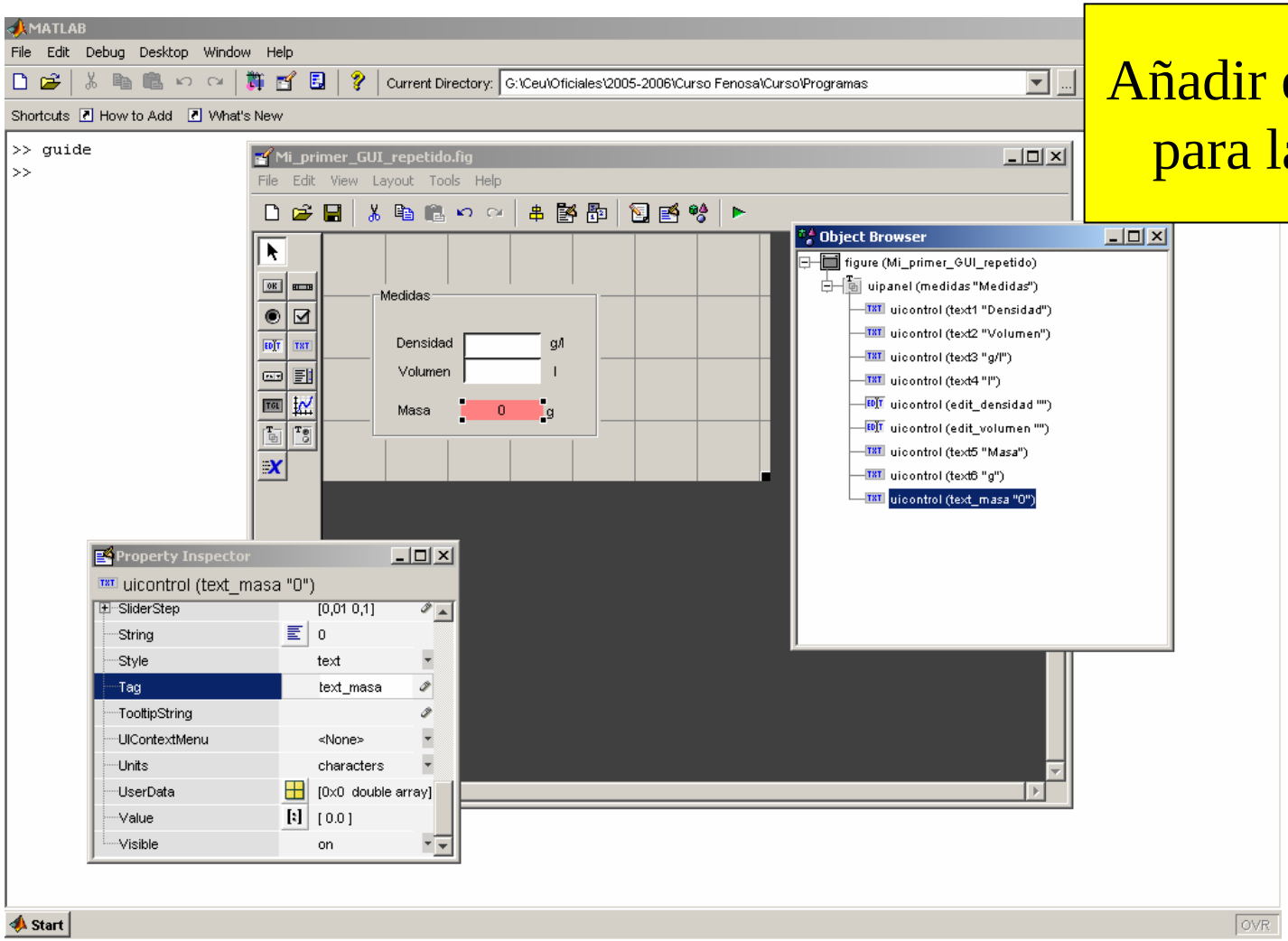
Editor - G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas\Mi_primer_GUI_repetido.m

```
47   % --- Executes just before Mi_primer_GUI_repetido is made visible.
48   function Mi_primer_GUI_repetido_OpeningFcn(hObject, eventdata, handles, varargin)
49   % This function has no output args, see OutputFcn.
50   % hObject    handle to figure
51   % eventdata  reserved - to be defined in a future version of MATLAB
52   % handles    structure with handles and user data (see GUIDATA)
53   % varargin   command line arguments to Mi_primer_GUI_repetido (see VARARGIN)
54
55   % Choose default command line output for Mi_primer_GUI_repetido
56 - handles.output = hObject;
57
58   % Update handles structure
59 - guidata(hObject, handles);
60
61   % Inicializar variables del GUI
62 - inicializar_gui(hObject,handles);
63
64   % UIWAIT makes Mi_primer_GUI_repetido wait for user response (see UIRESUME)
65   % uiwait(handles.figure1);
```

```
138
139  % ---------------------------------------------------------------
140  function inicializar_gui(fig_handle, handles)
141
142 - handles.densidad = 0;
143 - handles.volumen = 0;
144
145 - set(handles.edit_densidad, 'String', handles.densidad);
146 - set(handles.edit_volumen,  'String', handles.volumen);
147
148  % Update handles structure
149 - guidata(handles.figure1, handles);
150
151
```

Añadir una función de inicialización

GUIs: Creación



Añadir etiquetas para la masa

GUIs: Creación

Editor - G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas\Mi_primer_GUI_repetido.m

```
121 -  end
122
123    % Guardar el nuevo valor en handles
124 -  handles.volumen = volumen;
125 -  guidata(hObject,handles)
126
127    % --- Executes during object creation, after setting all properties.
128    function edit_volumen_CreateFcn(hObject, eventdata, handles)
129    % hObject    handle to edit_volumen (see GCBO)
130    % eventdata  reserved - to be defined in a future version of MATLAB
131    % handles    empty - handles not created until after all CreateFcns called
132
133    % Hint: edit controls usually have a white background on Windows.
134    %       See ISPC and COMPUTER.
135 -  if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
136 -      set(hObject,'BackgroundColor','white');
137 -  end
138
139    % --------------------------------------------------------------------
140    function inicializar_gui(fig_handle, handles)
141
142 -  handles.densidad = 0;
143 -  handles.volumen = 0;
144
145 -  set(handles.edit_densidad, 'String', handles.densidad);
146 -  set(handles.edit_volumen,  'String', handles.volumen);
147
148 -  set(handles.text_masa, 'String', '0');
149
150    % Update handles structure
151 -  guidata(handles.figure1, handles);
152
153
```

Modificar la función de inicialización

GUIs: Creación

Editor - G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas\Mi_primer_GUI_repetido.m

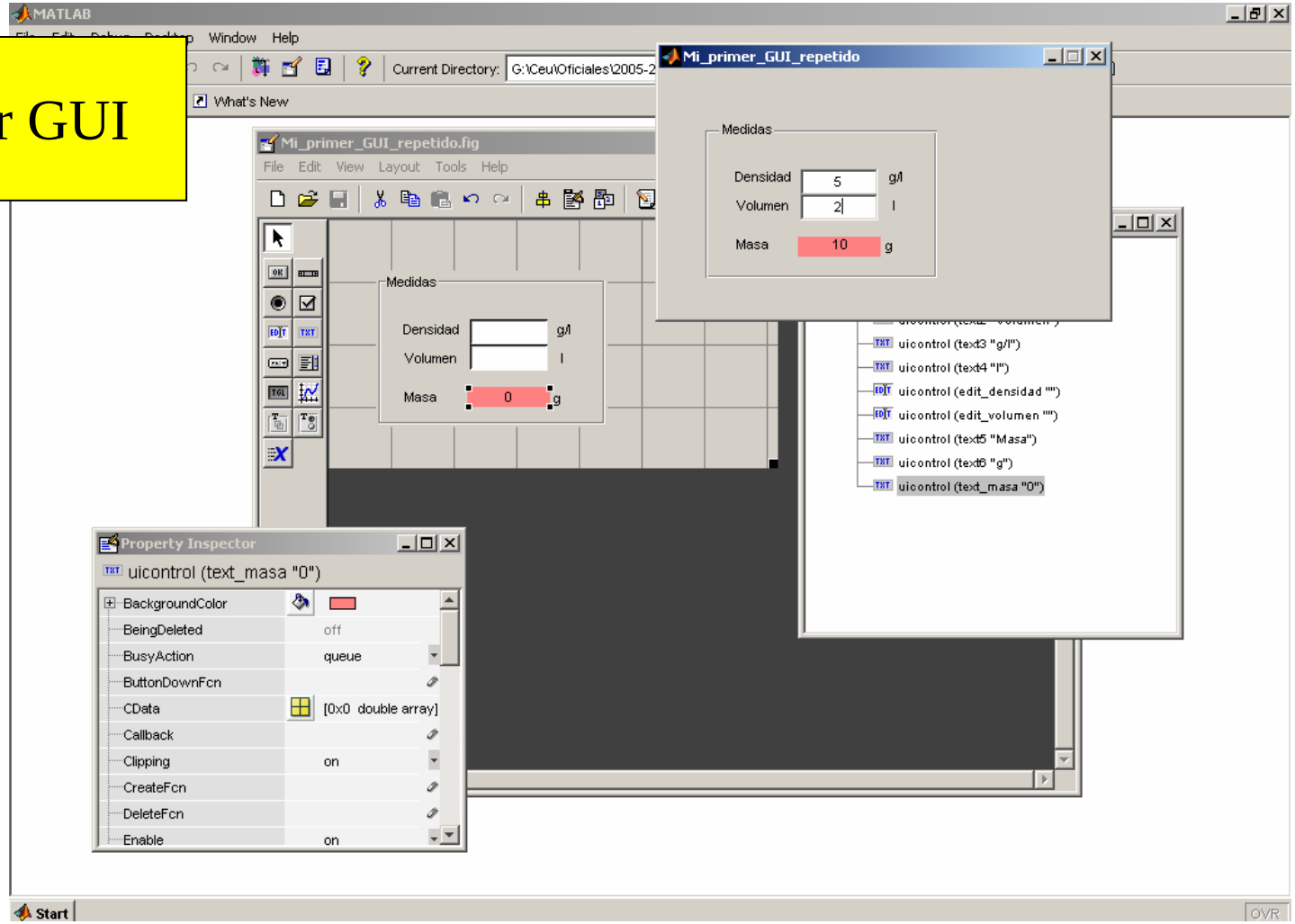
```
% Get default command line output from handles structure
varargout{1} = handles.output;

function edit_densidad_Callback(hObject, eventdata, handles)
% hObject    handle to edit_densidad (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_densidad as text
%        str2double(get(hObject,'String')) returns contents of edit_densidad as a double

% Recoger el valor del interfaz
densidad = str2double(get(hObject, 'String'));
if isnan(density)
    set(hObject, 'String', 0);
    errordlg('La entrada debe ser un número','Error');
end

% Guardar el nuevo valor en handles
handles.densidad = densidad;
guidata(hObject,handles)

% Actualizar masa
update_masa(handles)

% --- Executes during object creation, after setting all properties.
function edit_densidad_CreateFcn(hObject, eventdata, handles)
```

```
guidata(handles.figure1, handles);

% Actualizar la masa en la interfaz
function update_masa(handles)
set(handles.text_masa,'String',num2str(handles.densidad*handles.volumen))
```

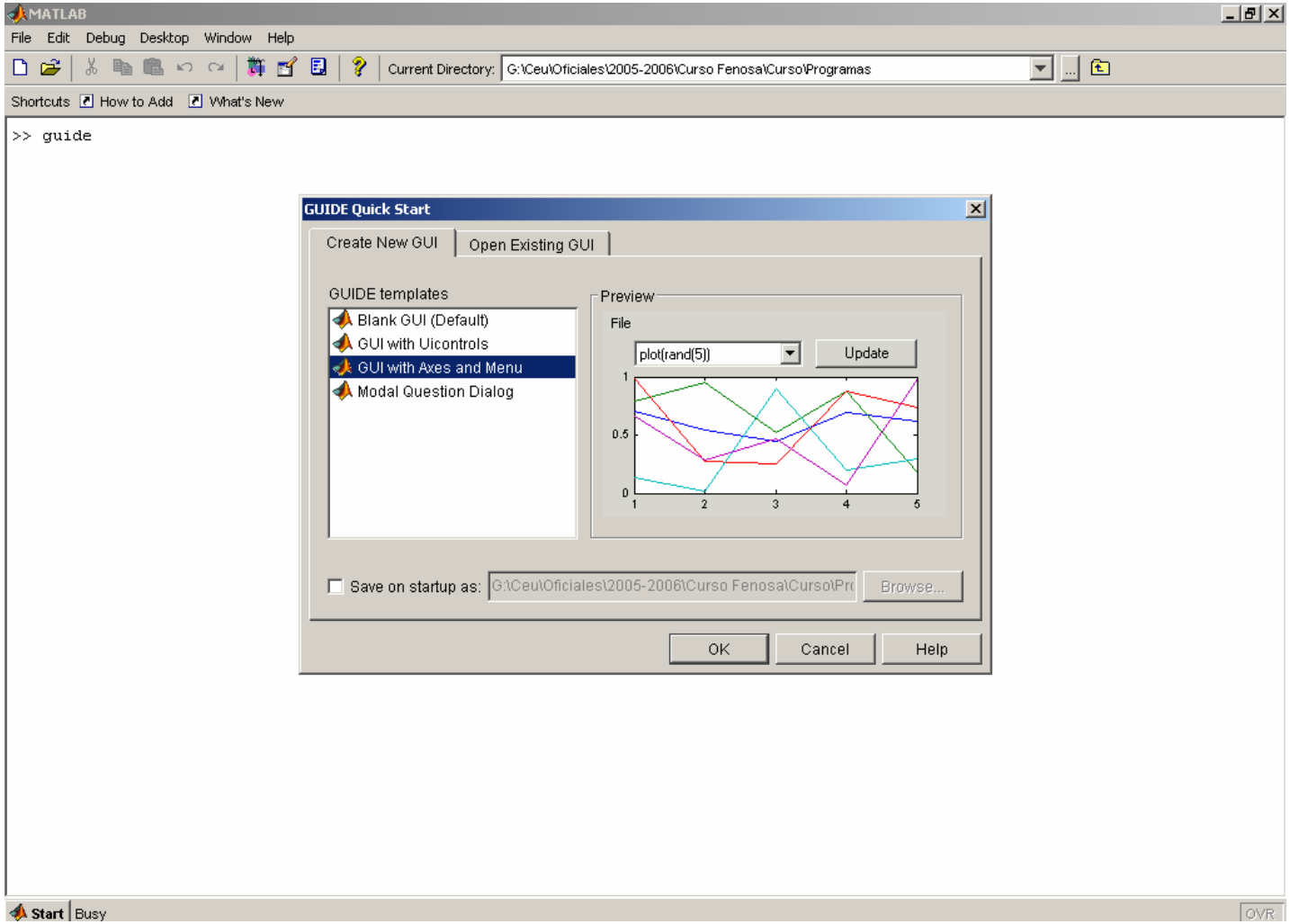
Crear función update_masa

GUIs: Creación

Ejecutar GUI



GUIs


MATLAB
File Edit Debug Desktop Window Help
Current Directory: G:\Ceu\Oficiales\2005-2006\Curso Fenosa\Curso\Programas
Shortcuts How to Add What's New
>> guide
GUIDE Quick Start
Create New GUI
Open Existing GUI
GUIDE templates
Blank GUI (Default)
GUI with Uicontrols
GUI with Axes and Menu
Modal Question Dialog
Preview
File
plot(rand(5))
Update
Save on startup as:
Browse...
OK
Cancel
Help
Start
Busy
OVR

GUIs

Help

File Edit View Go Favorites Desktop Window Help

Help Navigator

Contents | Index | Search | Demos

- Begin Here
- Release Notes
- Installation
- MATLAB
  - Getting Started
  - Examples
  - Desktop Tools and Development Environme
  - Mathematics
  - Data Analysis
  - Programming
  - Graphics
  - 3-D Visualization
  - Creating Graphical User Interfaces (GUIs)
  - Functions - By Category
    - Desktop Tools and Development Environme
    - Mathematics
    - Data Analysis
    - Programming and Data Types
    - File I/O
    - Graphics
    - 3-D Visualization
    - Creating Graphical User Interfaces
    - External Interfaces
  - Functions - Alphabetical List
  - Handle Graphics Property Browser
  - External Interfaces
  - C and Fortran Functions - By Category
  - C and Fortran Functions - Alphabetical List
  - Release Notes
  - Printable Documentation (PDF)
  - Product Page (Web)
- MATLAB Compiler
- SimBiology
- Bioinformatics Toolbox
- Communications Toolbox

Title: Creating Graphical User Interfaces :: Functions -- By Category (MATLAB Functions)

MATLAB Function Reference

Creating Graphical User Interfaces

Predefined Dialog Boxes	Dialog boxes for error, user input, waiting, etc.
Deploying User Interfaces	Launching GUIs, creating the handles structure
Developing User Interfaces	Starting GUIDE, managing application data, getting user input
User Interface Objects	Creating GUI components
Finding Objects from Callbacks	Finding object handles from within callbacks functions
GUI Utility Functions	Moving objects, text wrapping
Controlling Program Execution	Wait and resume based on user input

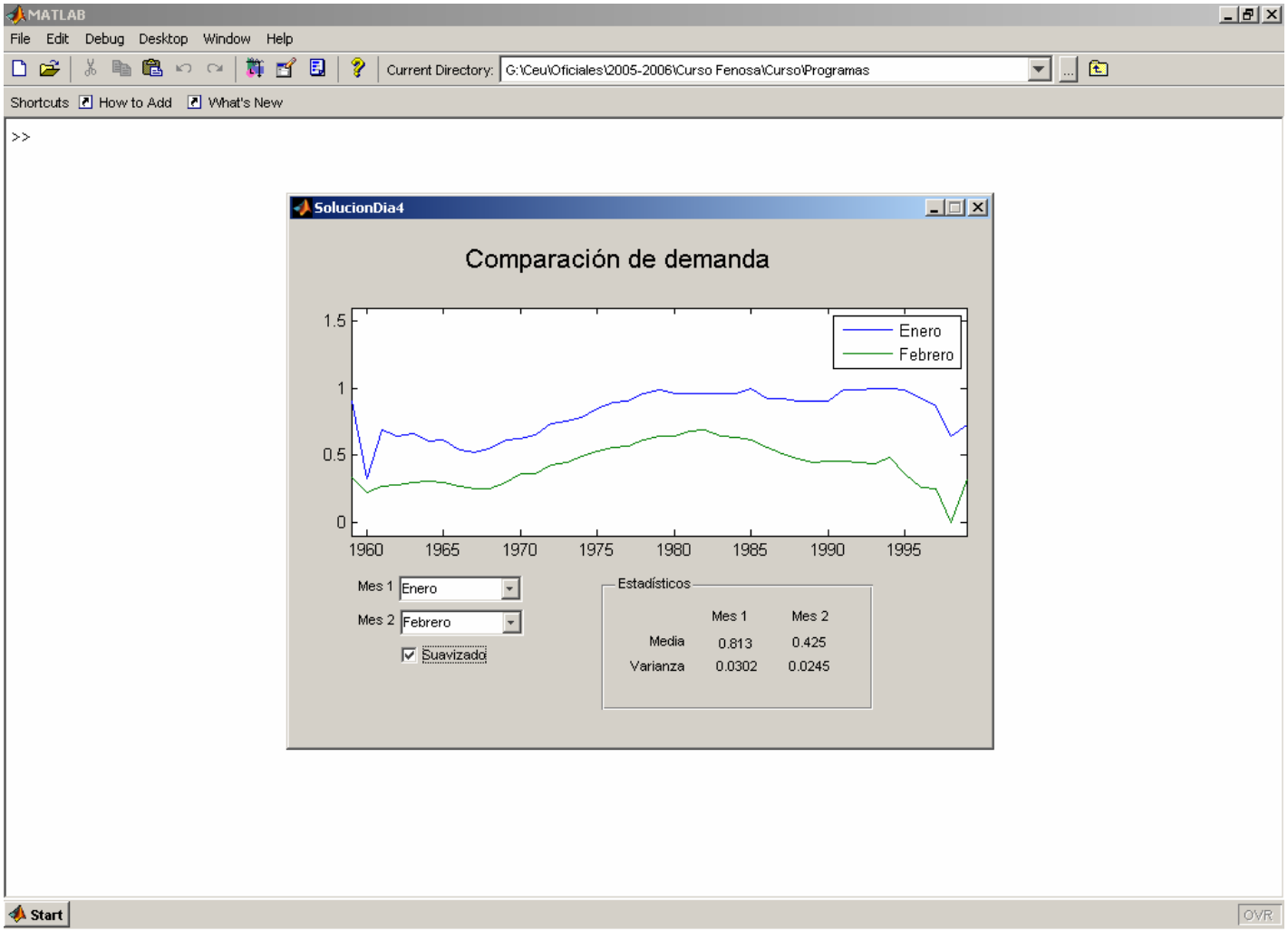
Predefined Dialog Boxes

dialog	Create and open dialog box
errordlg	Create and open error dialog box
helpdlg	Create and open help dialog box
inputdlg	Create and open input dialog box
listdlg	Create and open list-selection dialog box
msgbox	Create and open message dialog box
pagesetupdlg	Open page setup dialog box
printdlg	Open print dialog box
questdlg	Open question dialog box
uigetdir	Open standard dialog box for retrieving directory
uigetfile	Open standard dialog box for retrieving files
uigetpref	Open dialog box for retrieving preferences
uiopen	Open file selection dialog box with appropriate file filters
uiputfile	Open standard dialog box for saving files
uisave	Open standard dialog box for saving workspace variables
uisetcolor	Open standard dialog box for setting object's ColorSpec

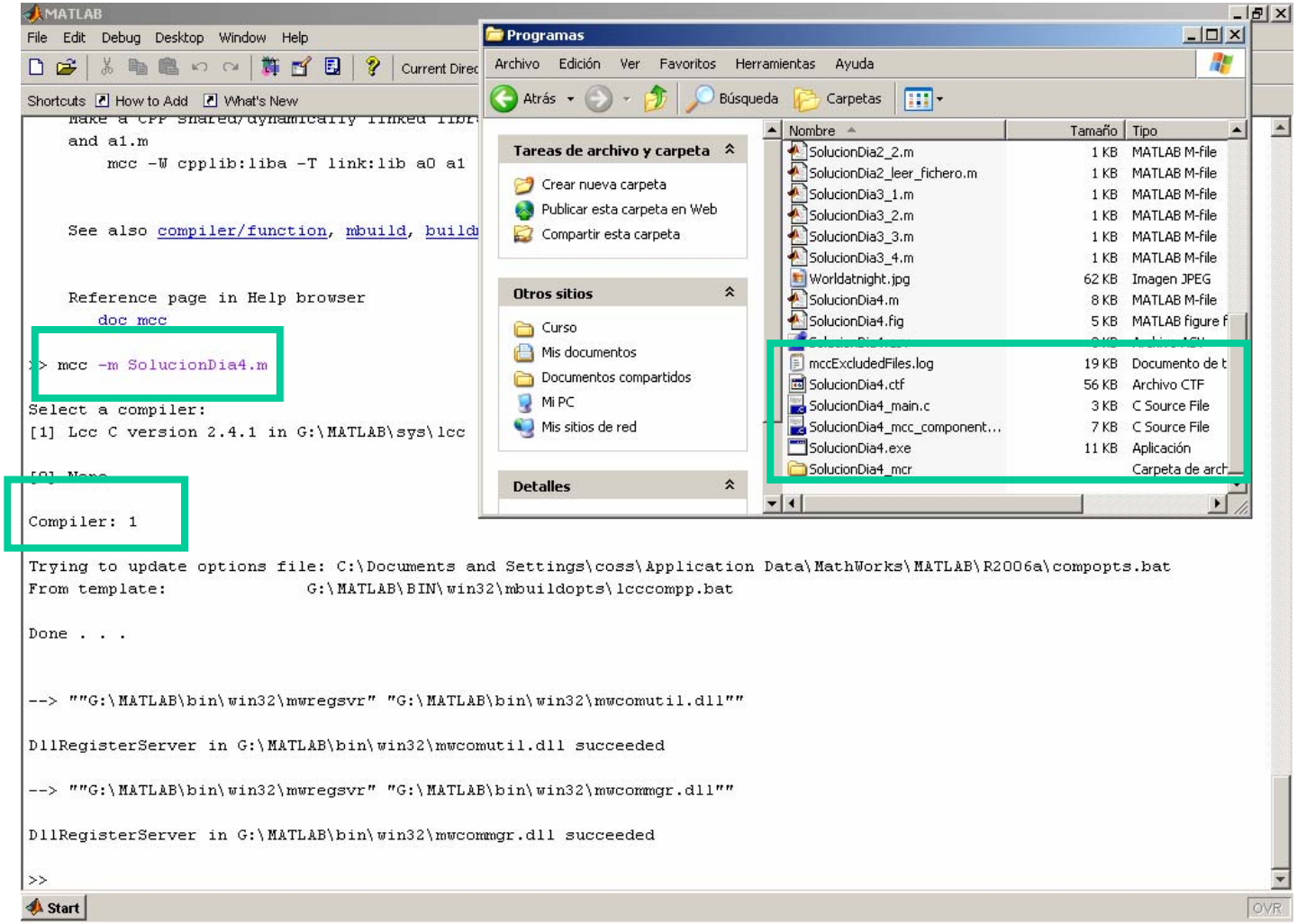
Ejercicio final

- Implementar un GUI que permita comparar la demanda normalizada de dos meses.
- La comparación debe poder realizarse opcionalmente con una función de suavizado.
- Mostrar en un panel aparte la media y la varianza de los meses solicitados.

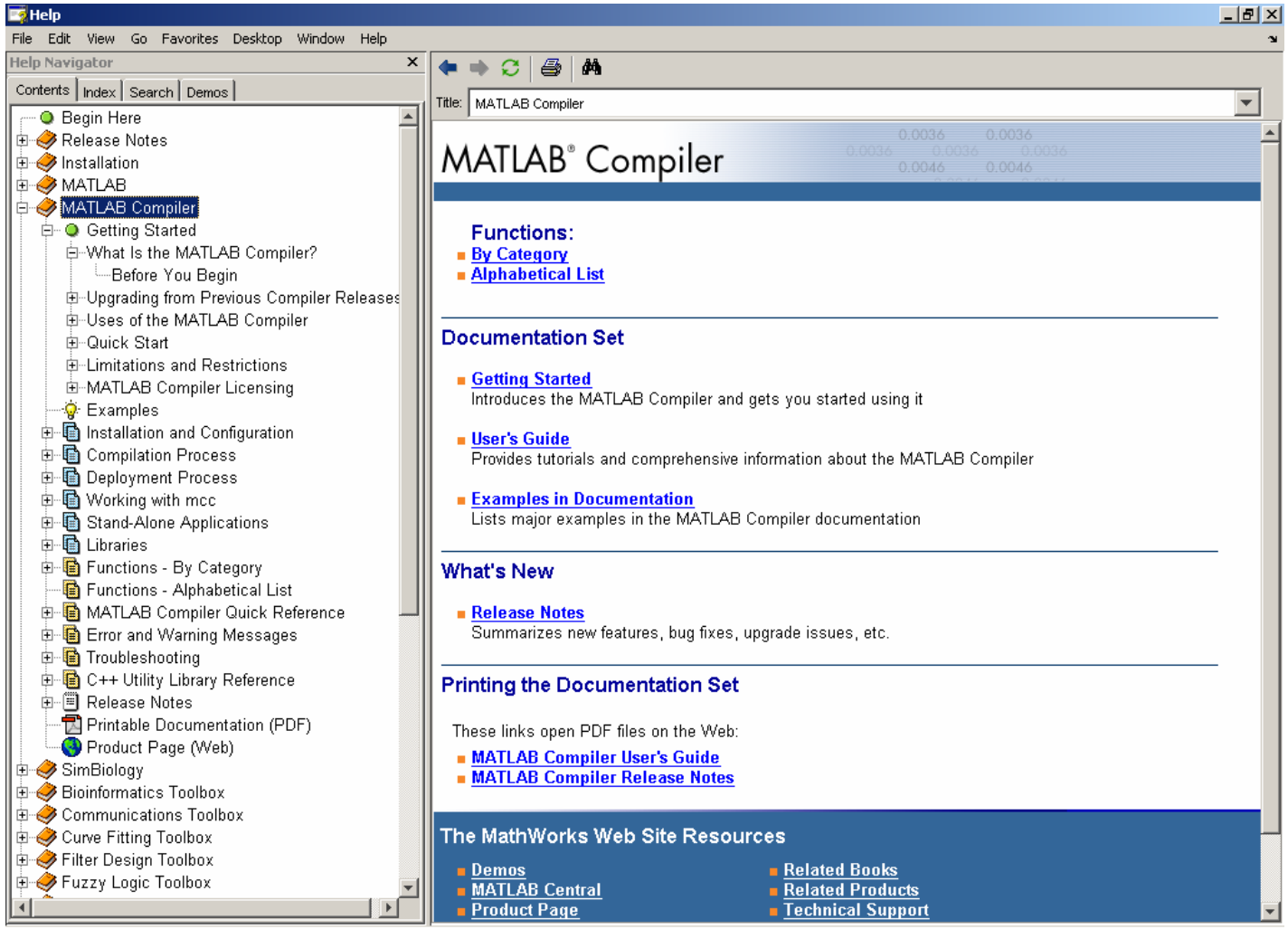
Ejercicio final



Generación de autoejecutables



Generación de autoejecutables



CEU
Universidad
San Pablo

CURSO de UTILIZACIÓN PRÁCTICA de MATLAB

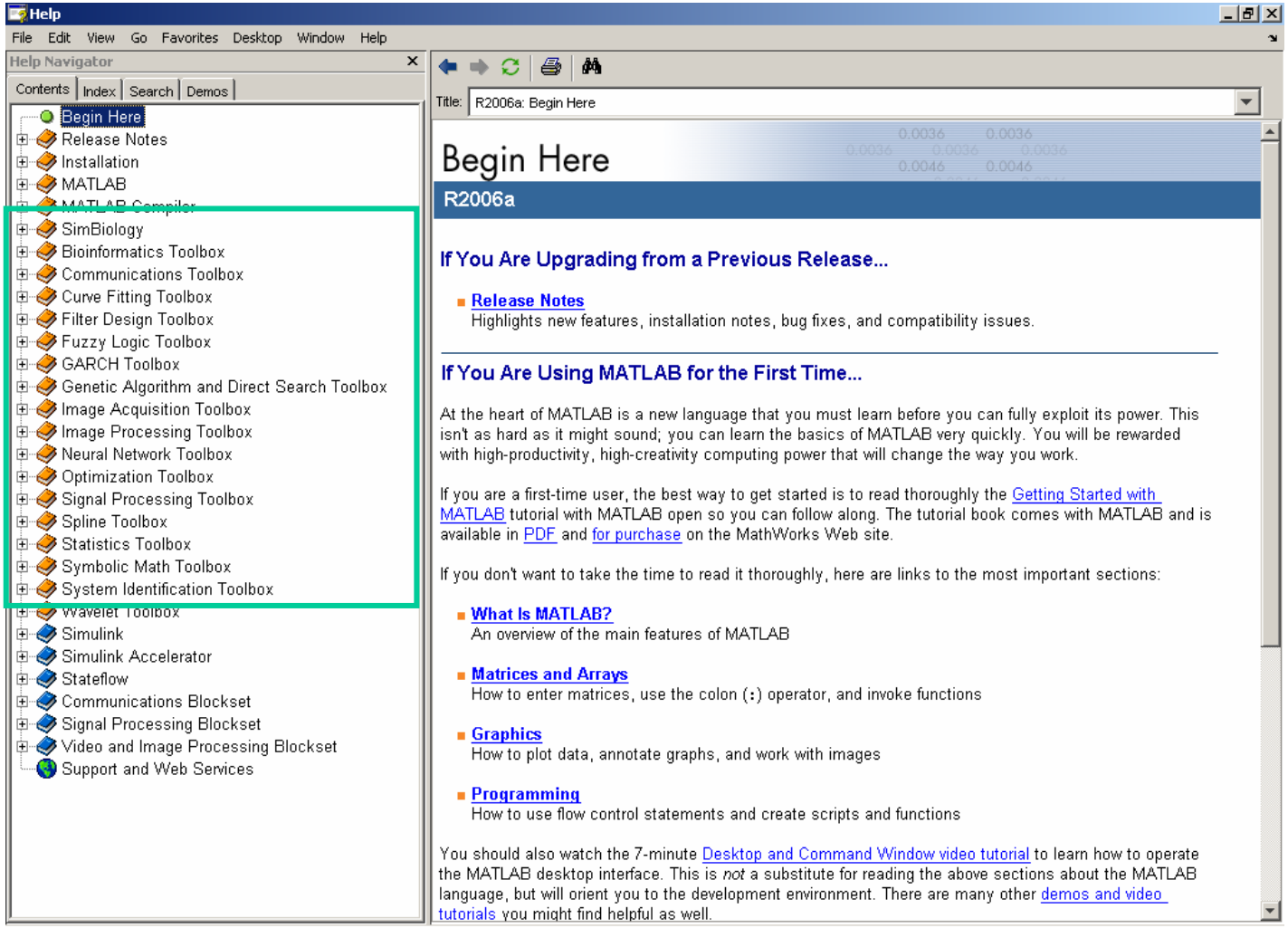
Sesión 5

Carlos Óscar Sánchez Sorzano, Ph.D.
Madrid, July 17th 2006

Cronograma del curso

- Día 1: Operaciones con matrices y vectores. Funciones de librería.
- Día 2: Otros tipos de datos en MATLAB. Programación en MATLAB.
- Día 3: Gráficos bidimensionales. Gráficos tridimensionales.
- Día 4: Interfaces de usuario en MATLAB. Generación de programas autónomos
- **Día 5: Librerías de interés práctico**
- Día 6: Interacción de MATLAB con Office y Visual Basic
- Día 7: Desarrollo de un proyecto

Librerías de interés práctico



Librerías de interés práctico

The MathWorks - Technical Computing - An introduction to technical computing with MATLAB - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

Dirección http://www.mathworks.com/applications/tech_computing/ Ir Vínculos

Google Ir Marcadores PageRank 217 bloqueados Corrector ortográfico Traducir Configuración

The MathWorks
Accelerating the pace of engineering and science

Home | Select Country | Contact Us | Store Search

Create Account | Log In

Products & Services | Industries | Academia | Support | User Community | Company

- Technical Computing Main Page
- Description
- Demos and Webinars
- User Stories
- Training
- Recommended Products
- Technical Literature
- News
- Events

Technical Computing

Mathematical computation, analysis, visualization, and algorithm development

MATLAB and companion toolboxes provide engineers, scientists, mathematicians, and educators with an environment for technical computing applications. These products serve a broad range of tasks across a variety of industries from automotive and electronics to industrial equipment and telecommunications.

Description Topics

- The Power of Technical Computing with MATLAB
- Data Analysis and Exploration
- Mathematical Algorithms to Build On
- Modeling and Simulation
- Visualization and Image Processing
- Programming and Application Development

Free Technical Kit

Contact sales
E-mail this page
Print this page

R2006b

Available Sept. 2006
» Learn more
» Download now

Free Seminar
Managing Risk with Extreme Value Theory and Copulas: A MATLAB Financial Case Study
new
» Learn more

Upcoming Webinar
Introduction to MATLAB®
new
» Register today

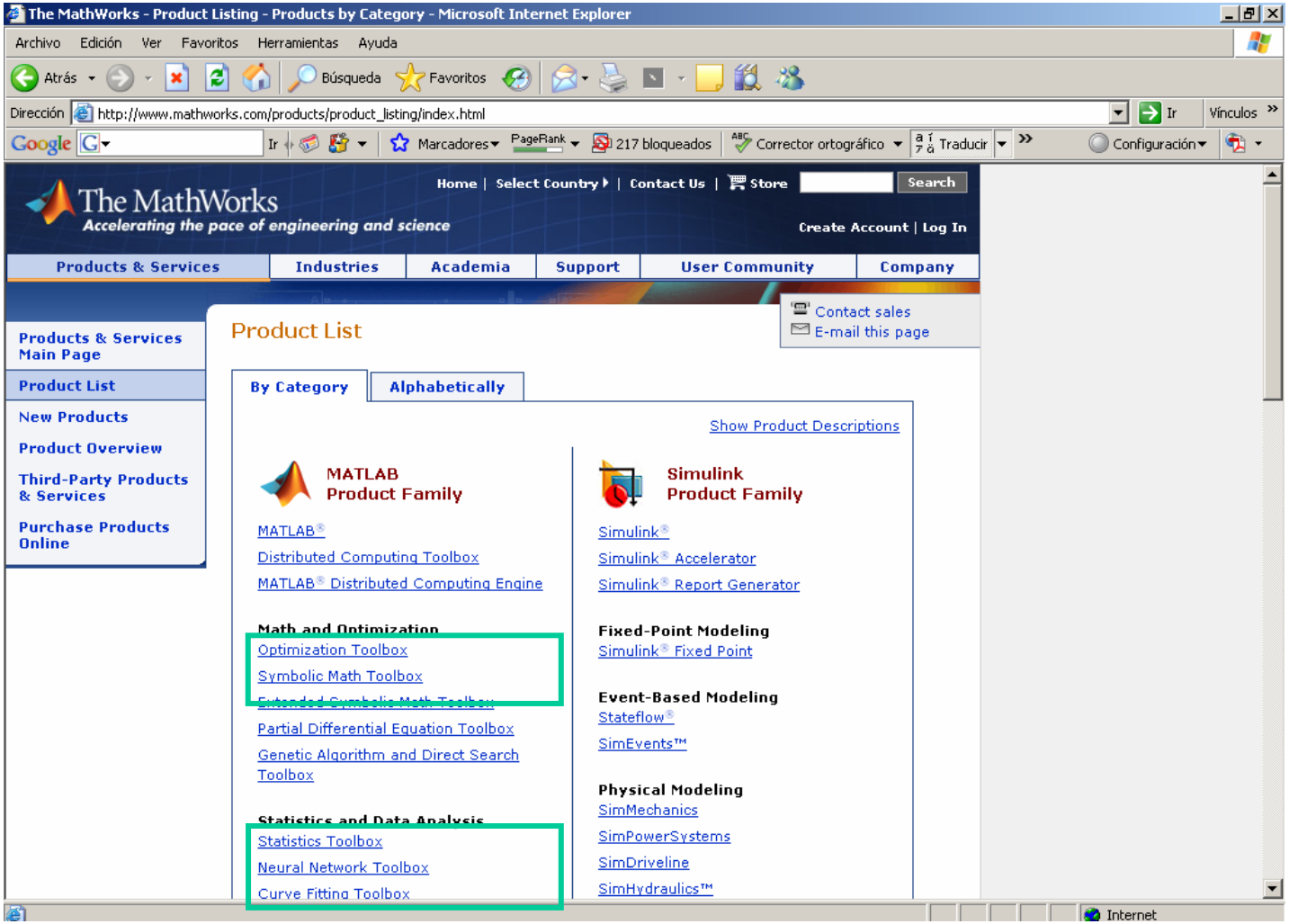
Internet

Librerías de interés práctico

Librerías de interés práctico

Librerías de interés práctico

Librerías de interés práctico



Librerías de interés práctico

The MathWorks - Product Listing - Products by Category - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

Dirección http://www.mathworks.com/products/product_listing/index.html Ir Vínculos

Google Ir Marcadores PageRank 217 bloqueados Corrector ortográfico Traducir Configuración

Toolbox

Statistics and Data Analysis
Statistics Toolbox
Neural Network Toolbox
Curve Fitting Toolbox
Spline Toolbox
Model-Based Calibration Toolbox

Control System Design and Analysis
Control System Toolbox
System Identification Toolbox
Fuzzy Logic Toolbox
Robust Control Toolbox
Model Predictive Control Toolbox
Aerospace Toolbox

Signal Processing and Communications
Signal Processing Toolbox
Communications Toolbox
Filter Design Toolbox
Filter Design HDL Coder
Wavelet Toolbox
Fixed-Point Toolbox
RF Toolbox
Link for Code Composer Studio™
Link for ModelSim®
Link for Incisive®

Physical Modeling
SimMechanics
SimPowerSystems
SimDriveline
SimHydraulics™

Simulation Graphics
Virtual Reality Toolbox
Gauges Blockset

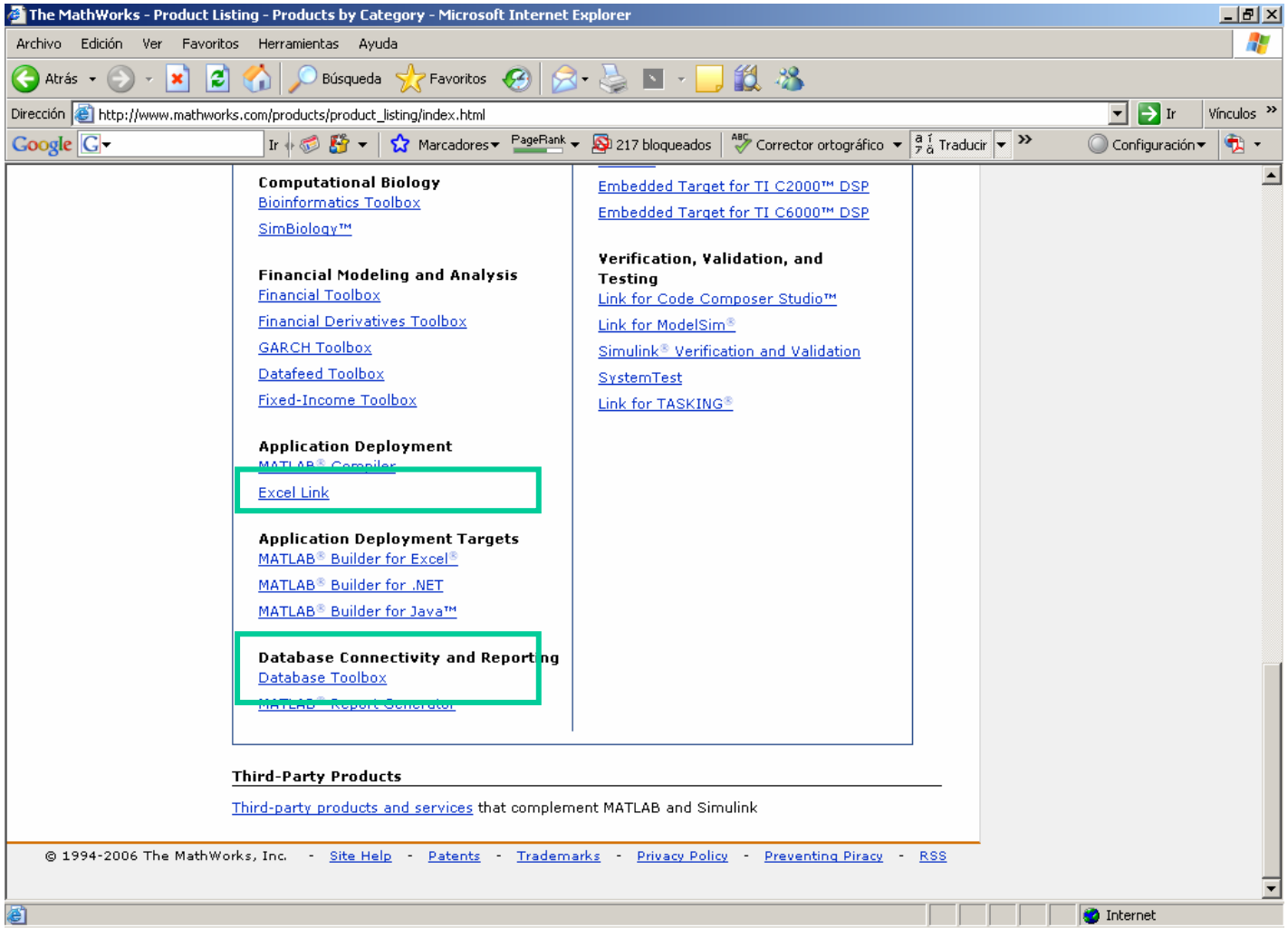
Control System Design and Analysis
Simulink® Control Design
Simulink® Response Optimization
Simulink® Parameter Estimation
Aerospace Blockset

Signal Processing and Communications
Signal Processing Blockset
Communications Blockset
RF Blockset
Video and Image Processing Blockset

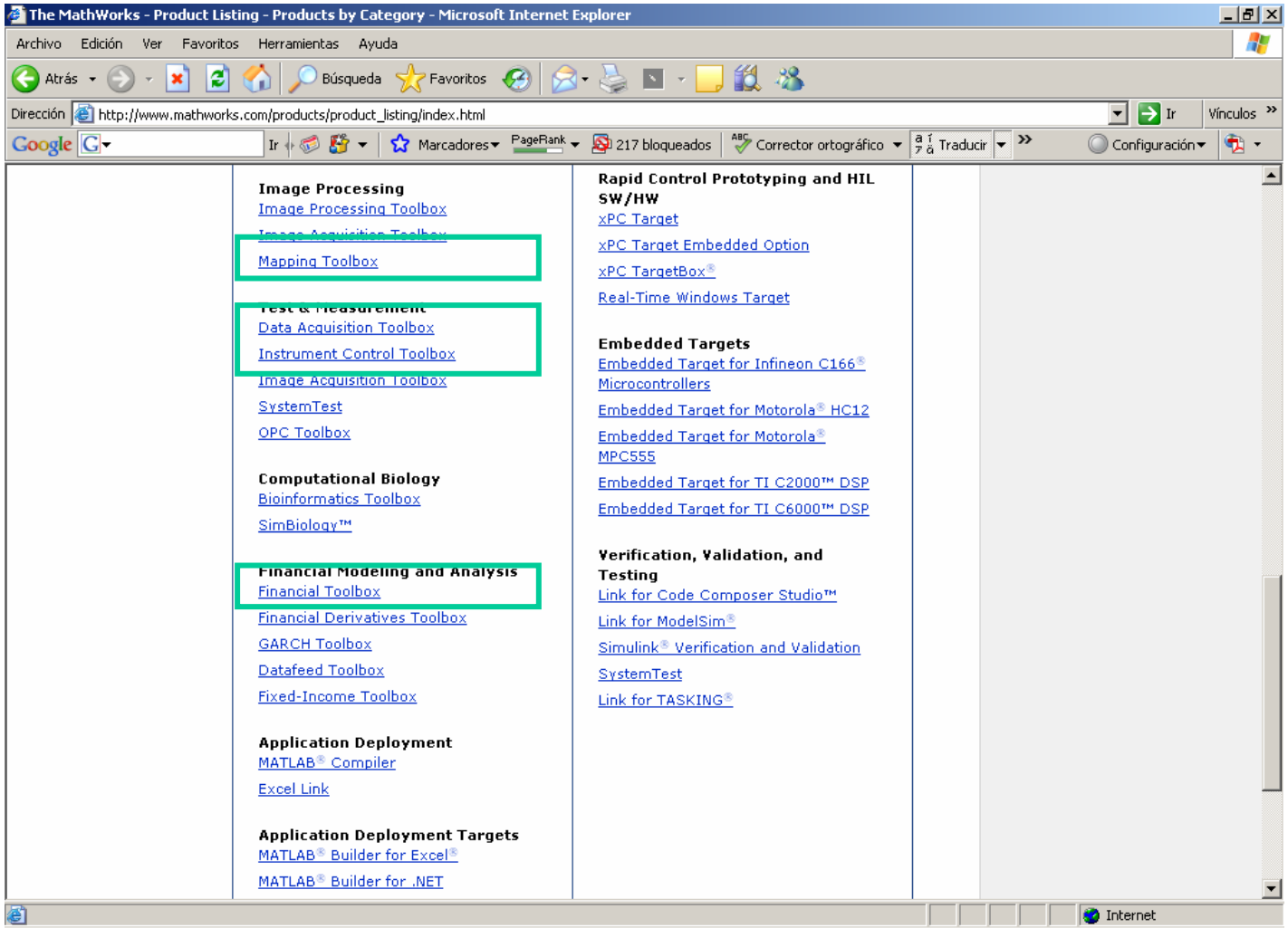
Code Generation
Real-Time Workshop®
Real-Time Workshop® Embedded Coder
Stateflow® Coder
Simulink® HDL Coder

Internet

Librerías de interés práctico



Librerías de interés práctico



CEU
Universidad
San Pablo

CURSO de UTILIZACIÓN PRÁCTICA de MATLAB

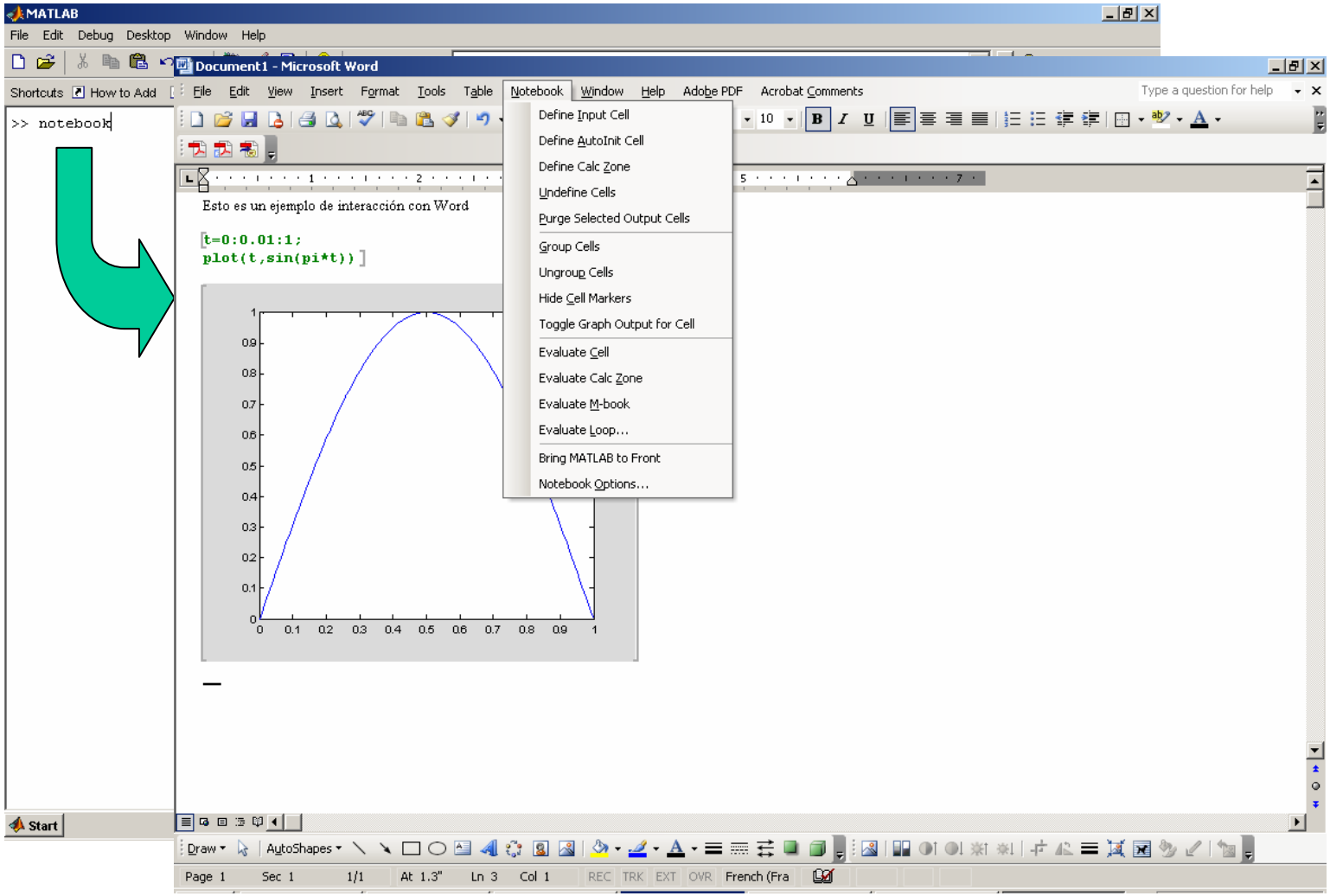
Sesión 6

Carlos Óscar Sánchez Sorzano, Ph.D.
Madrid, July 17th 2006

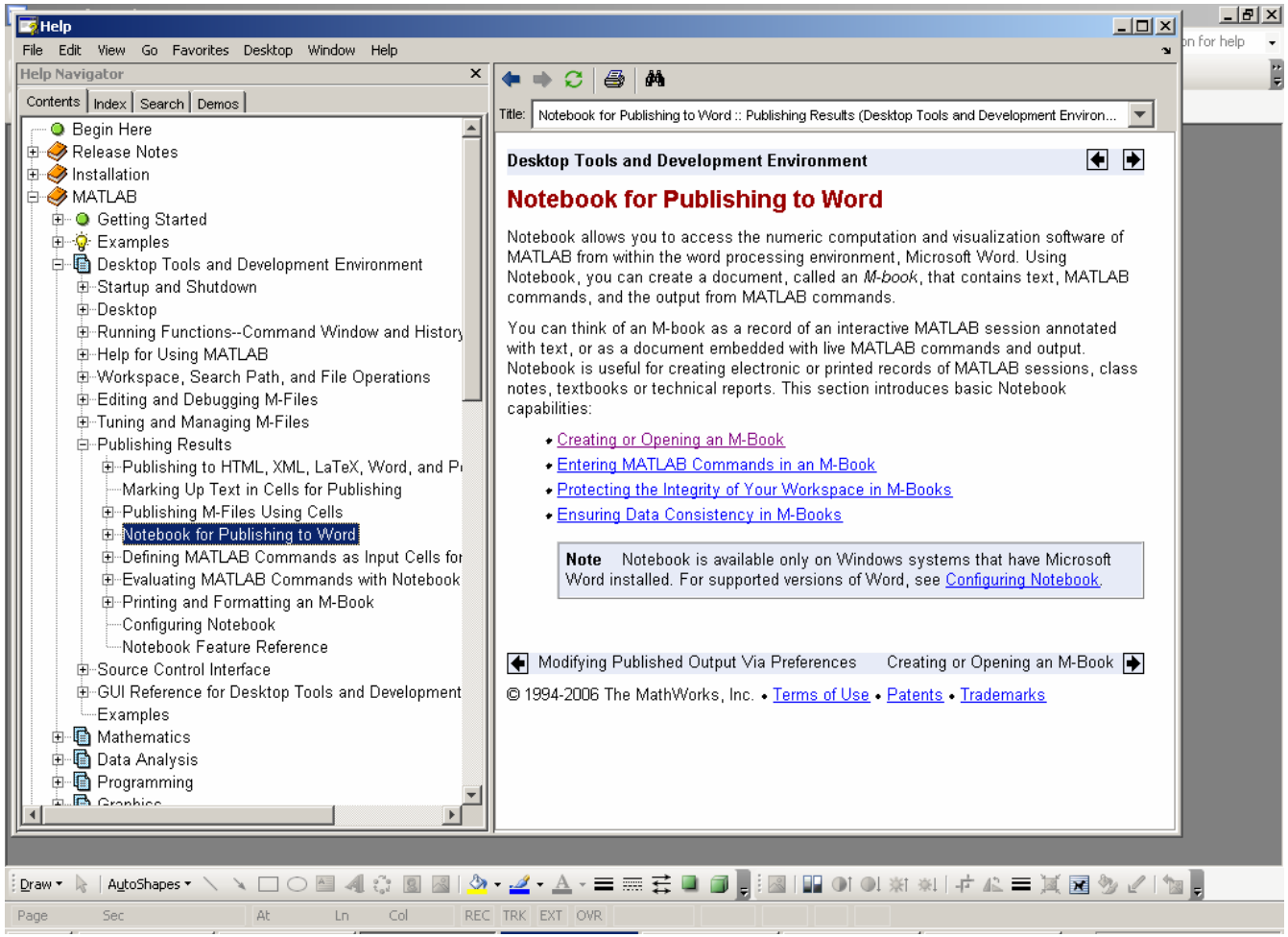
Cronograma del curso

- Día 1: Operaciones con matrices y vectores. Funciones de librería.
- Día 2: Otros tipos de datos en MATLAB. Programación en MATLAB.
- Día 3: Gráficos bidimensionales. Gráficos tridimensionales.
- Día 4: Interfaces de usuario en MATLAB. Generación de programas autónomos
- Día 5: Librerías de interés práctico
- **Día 6: Interacción de MATLAB con Office y Visual Basic**
- Día 7: Desarrollo de un proyecto

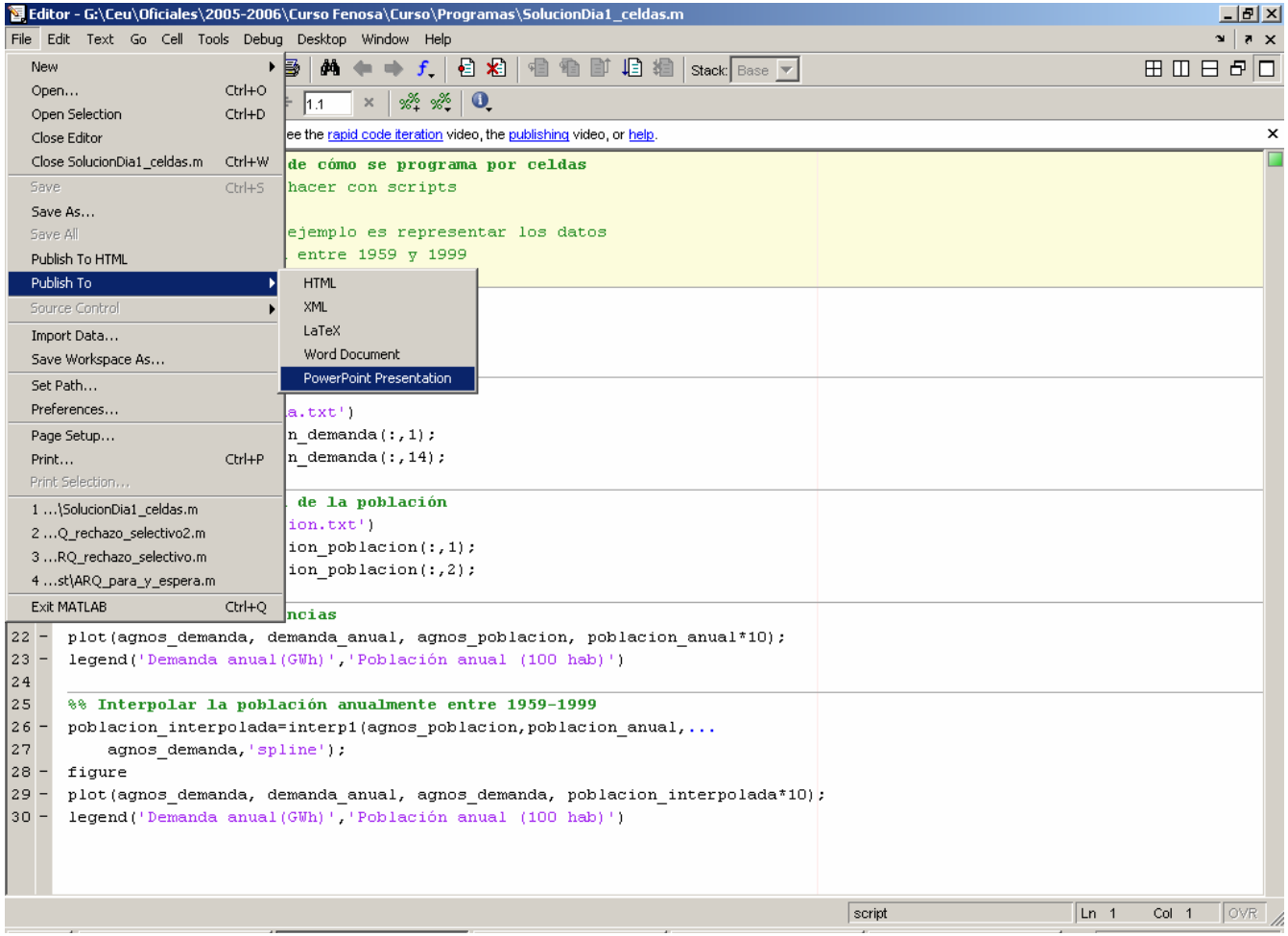
Interacción con Word



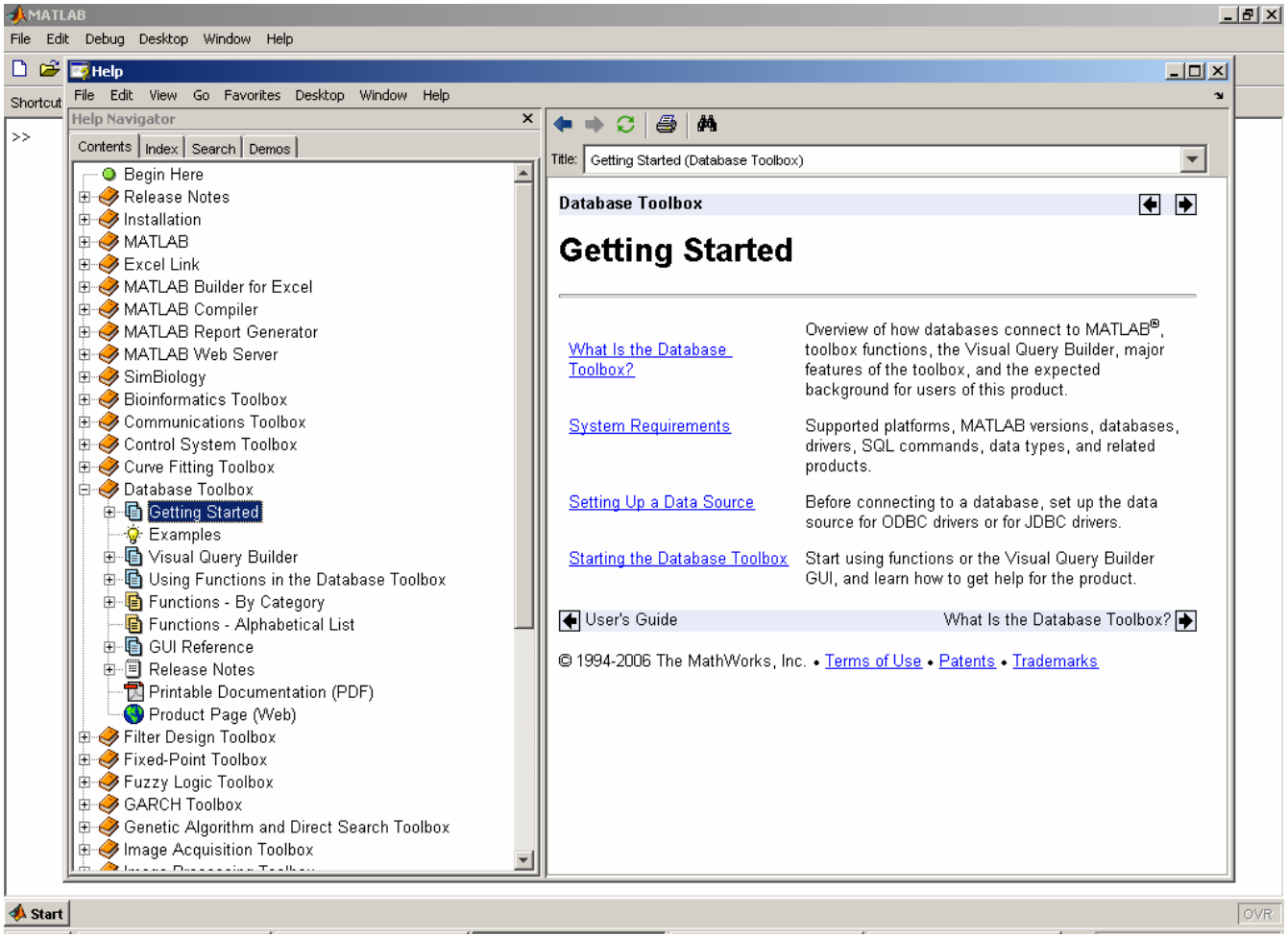
Interacción con Word



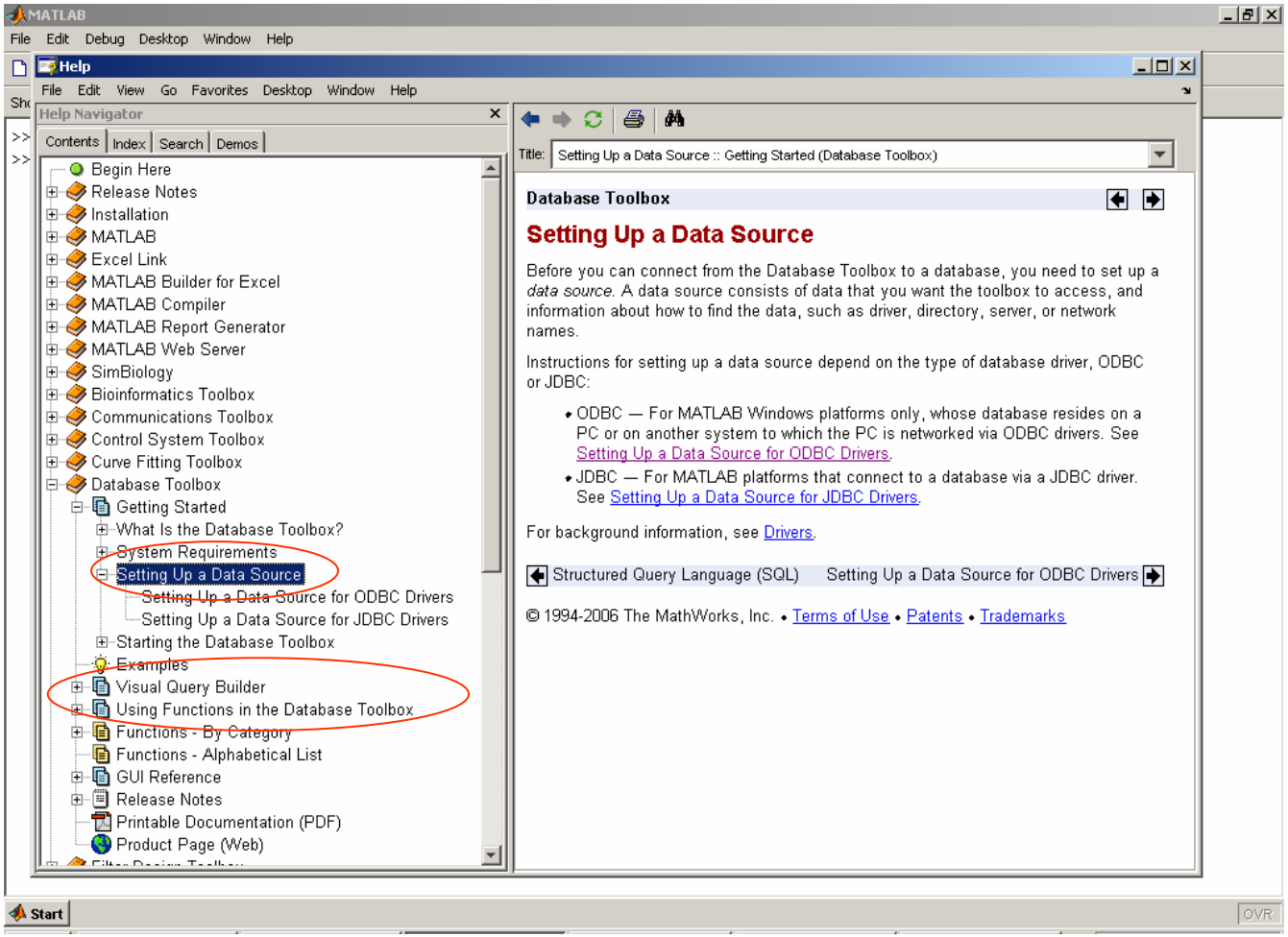
Interacción con Powerpoint



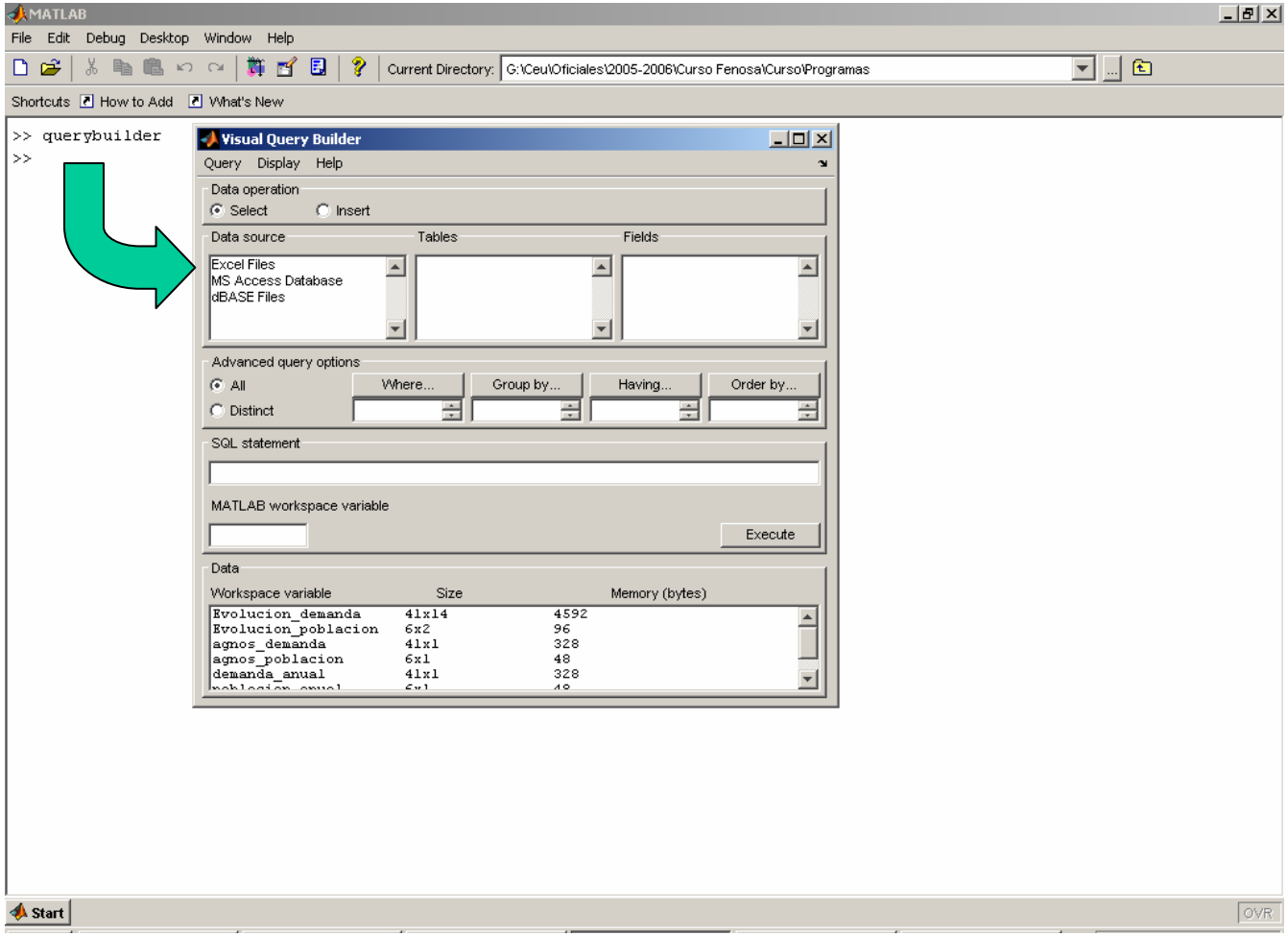
Interacción con Access



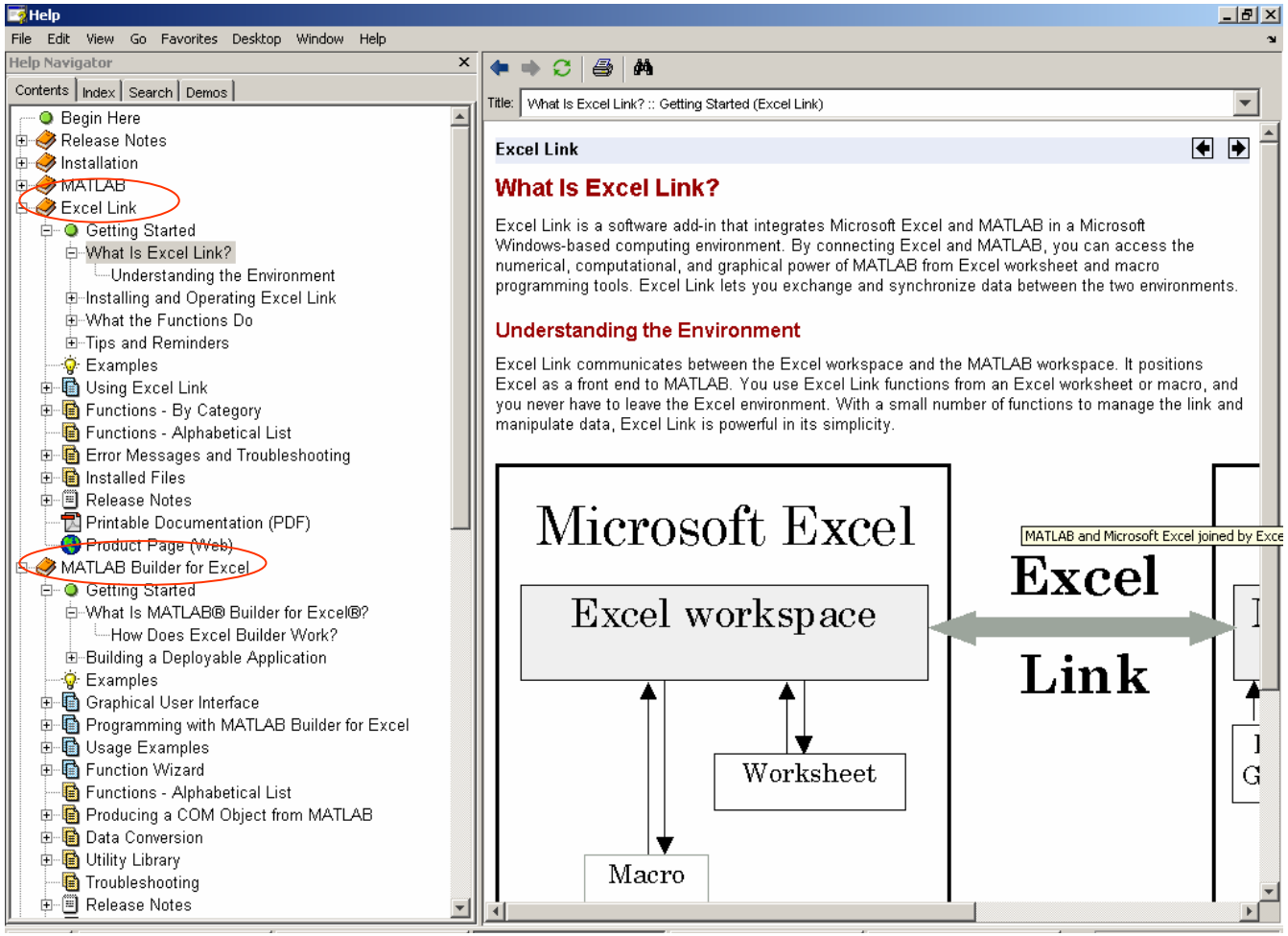
Interacción con Access



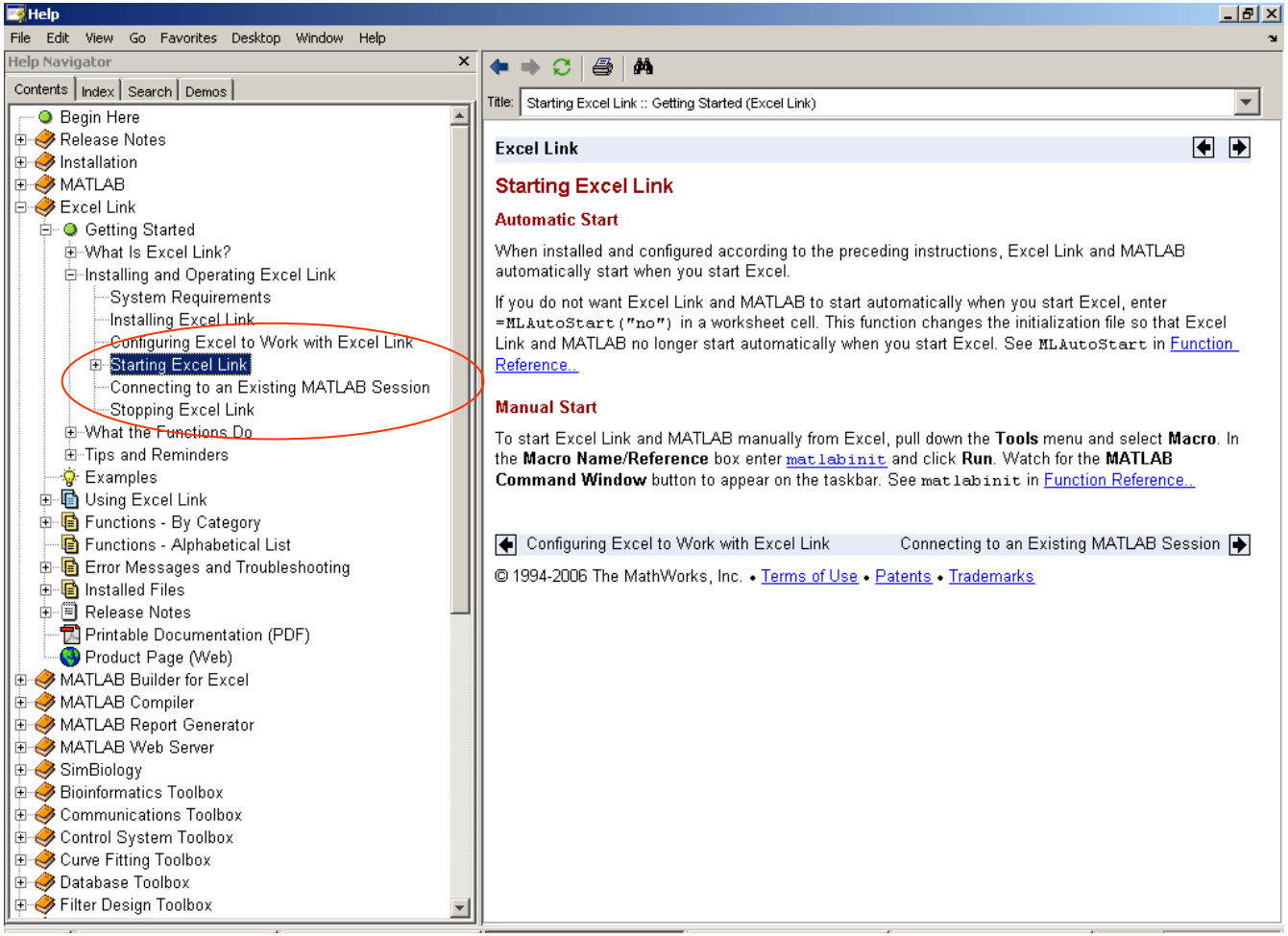
Interacción con Access



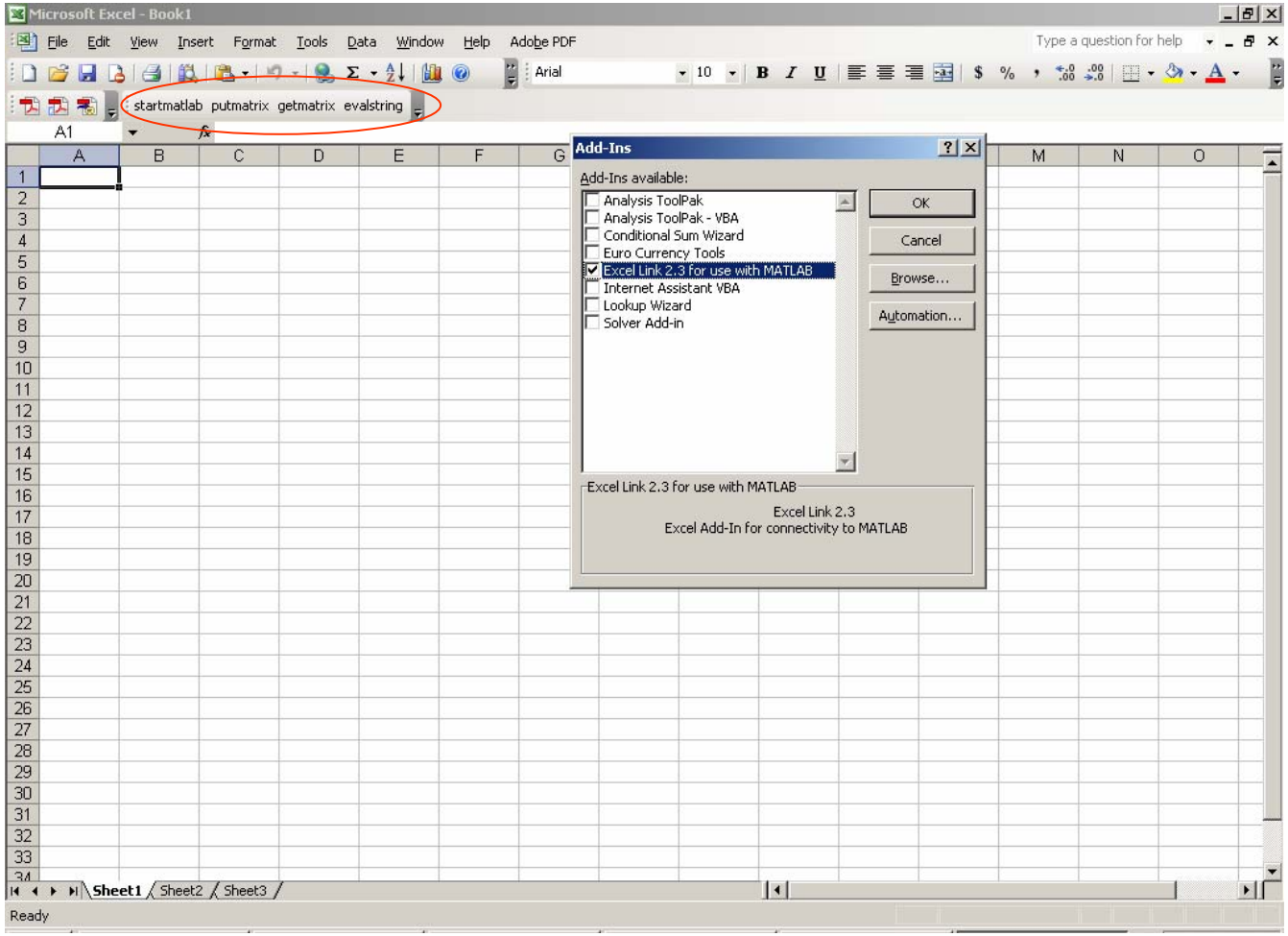
Interacción con Excel: Excel Link



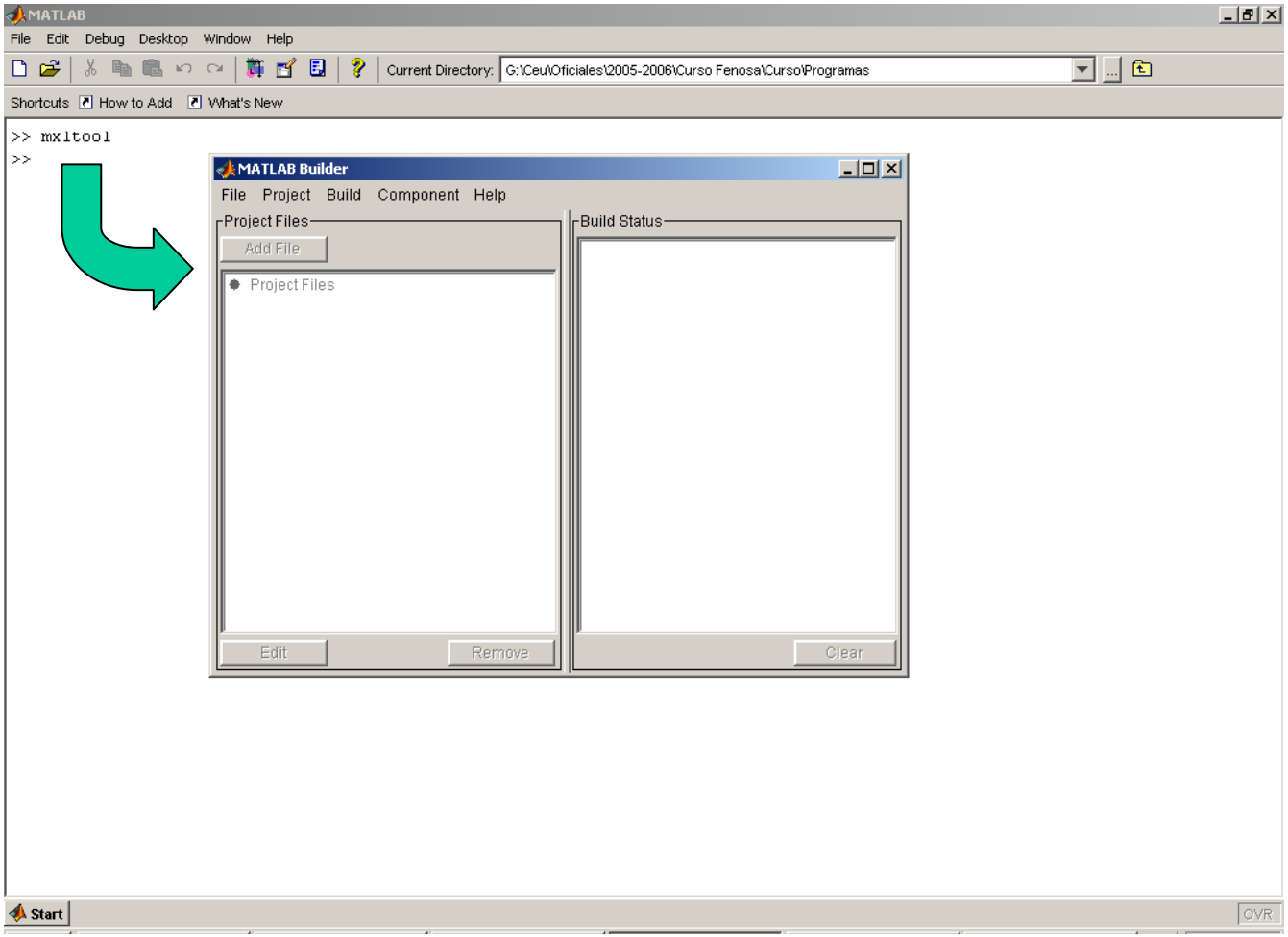
Interacción con Excel: Excel Link



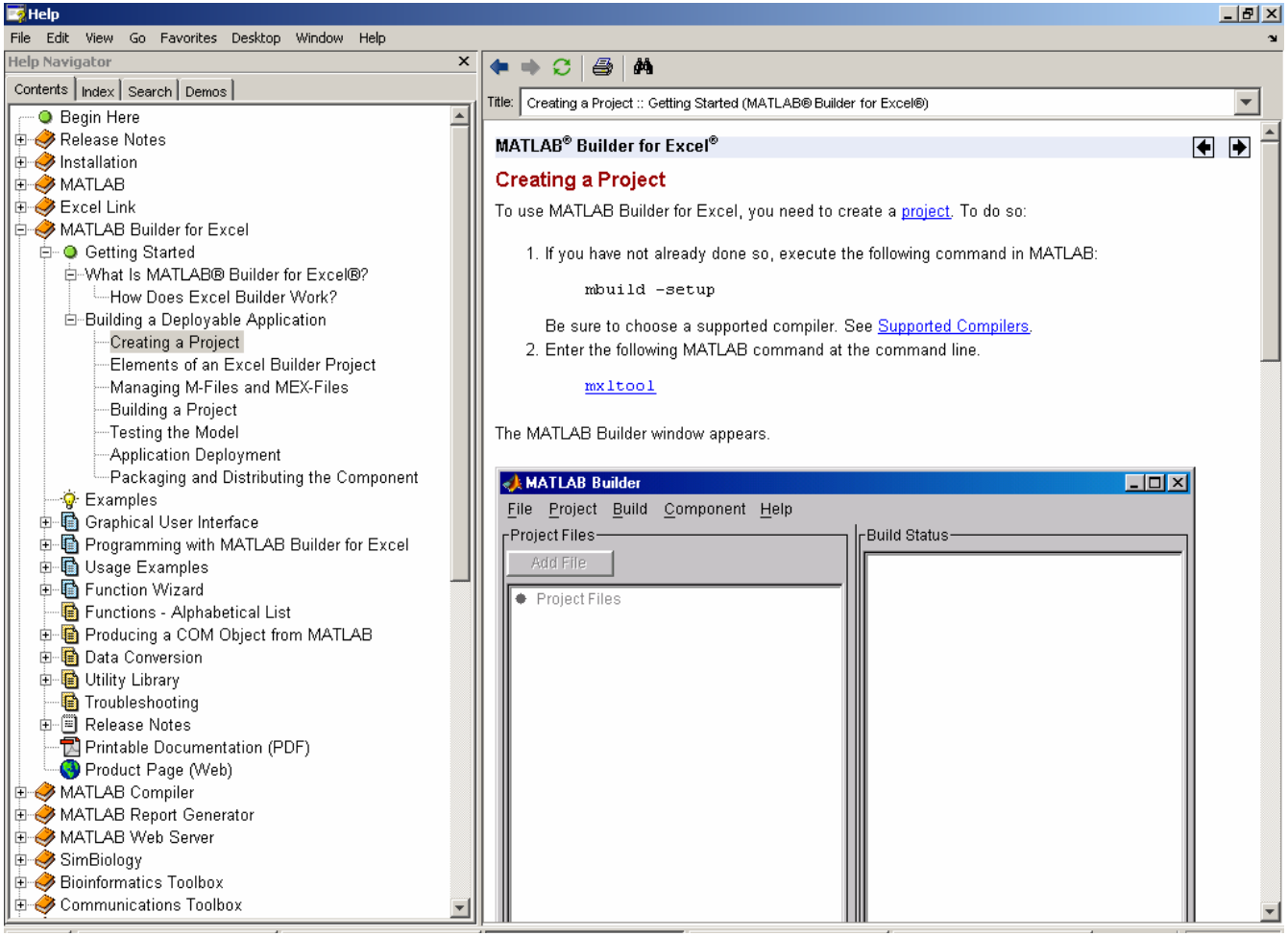
Interacción con Excel: Excel Link



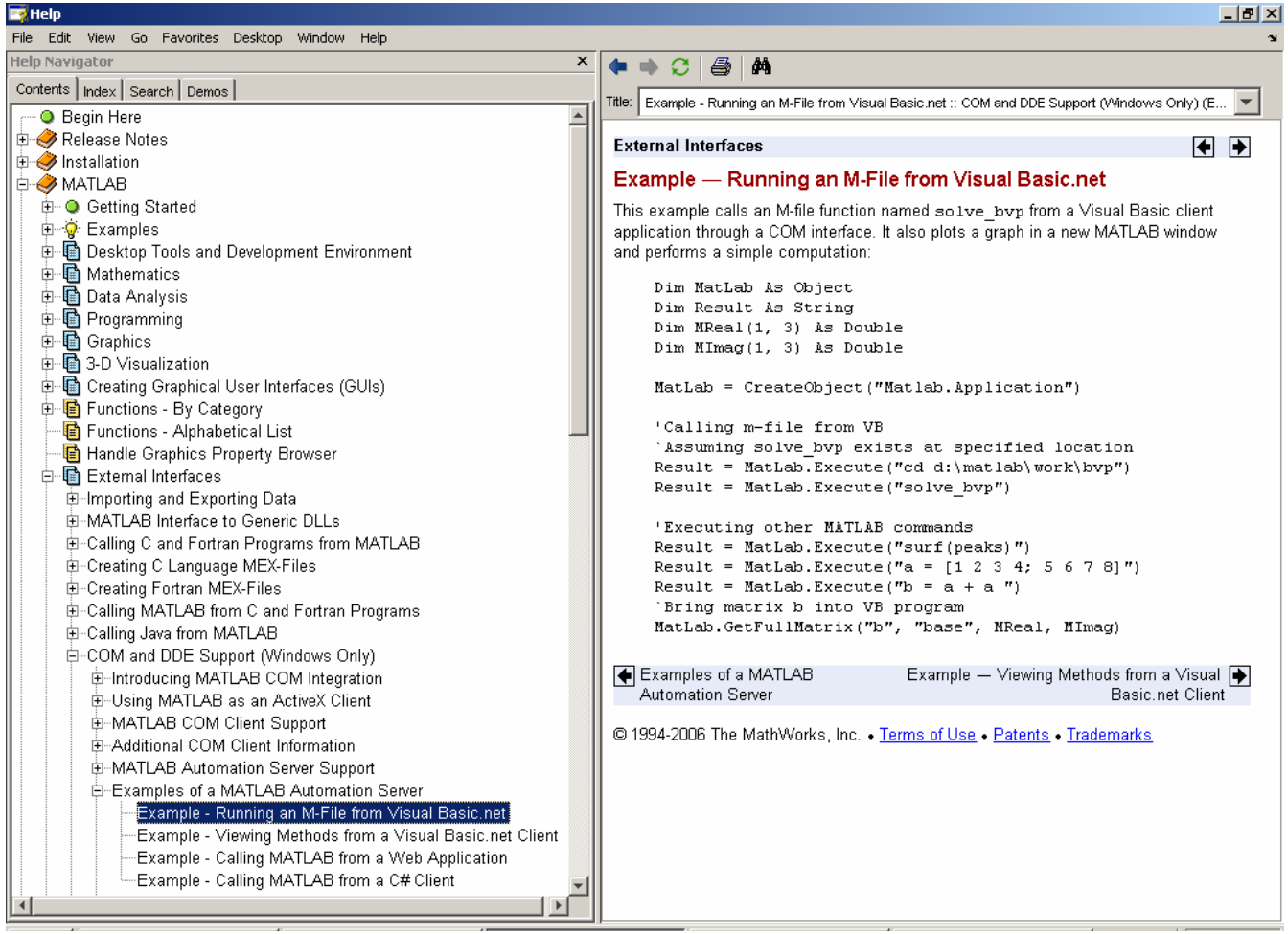
Interacción con Excel: Excel Builder



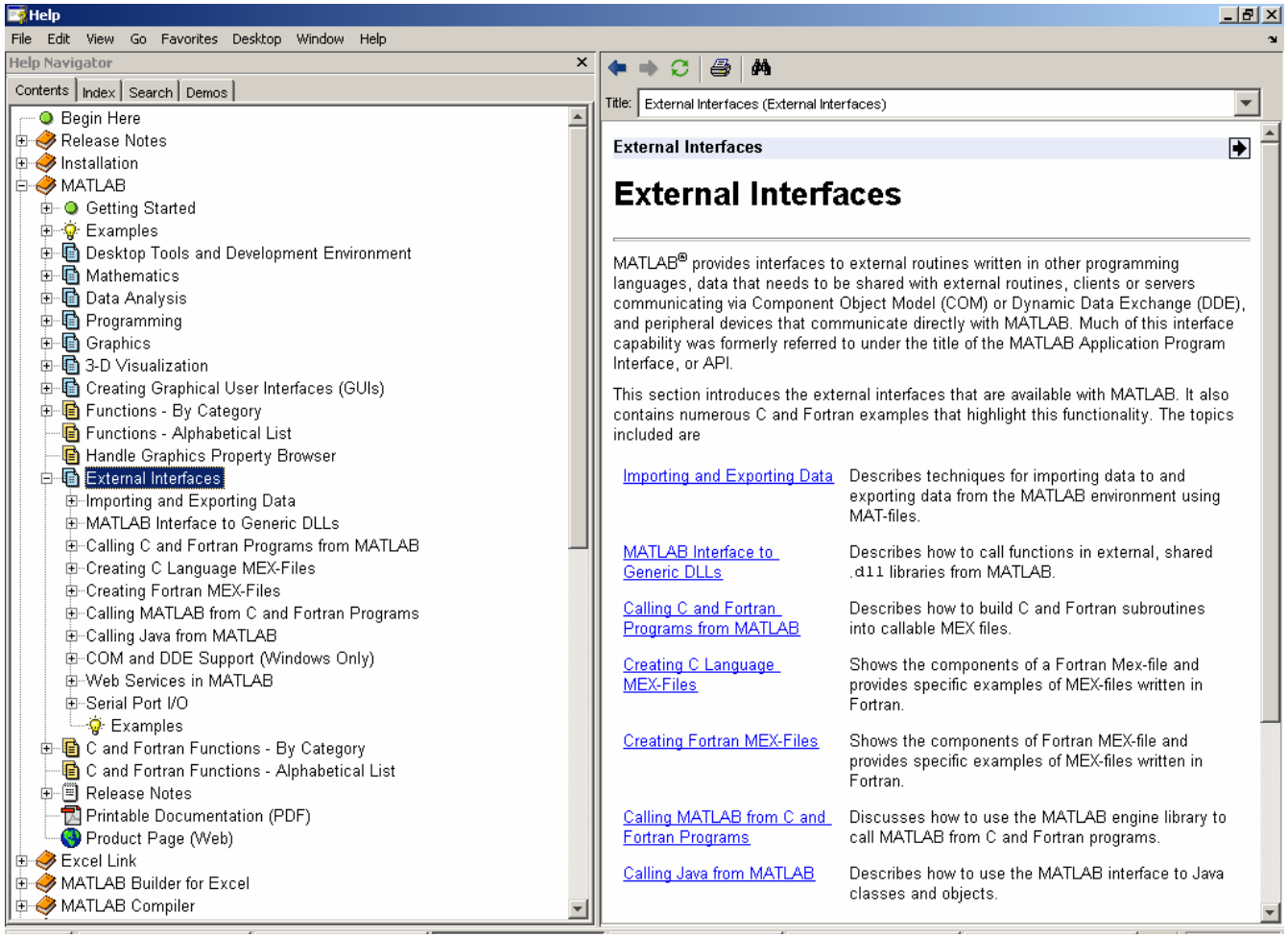
Interacción con Excel: Excel Builder



Interacción con Visual Basic



Ejercicio


Help
File Edit View Go Favorites Desktop Window Help
Help Navigator
Contents Index Search Demos
Begin Here
Release Notes
Installation
MATLAB
Getting Started
Examples
Desktop Tools and Development Environment
Mathematics
Data Analysis
Programming
Graphics
3-D Visualization
Creating Graphical User Interfaces (GUIs)
Functions - By Category
Functions - Alphabetical List
Handle Graphics Property Browser
External Interfaces
Importing and Exporting Data
MATLAB Interface to Generic DLLs
Calling C and Fortran Programs from MATLAB
Creating C Language MEX-Files
Creating Fortran MEX-Files
Calling MATLAB from C and Fortran Programs
Calling Java from MATLAB
COM and DDE Support (Windows Only)
Web Services in MATLAB
Serial Port I/O
Examples
C and Fortran Functions - By Category
C and Fortran Functions - Alphabetical List
Release Notes
Printable Documentation (PDF)
Product Page (Web)
Excel Link
MATLAB Builder for Excel
MATLAB Compiler
Title: External Interfaces (External Interfaces)
External Interfaces
External Interfaces
MATLAB® provides interfaces to external routines written in other programming languages, data that needs to be shared with external routines, clients or servers communicating via Component Object Model (COM) or Dynamic Data Exchange (DDE), and peripheral devices that communicate directly with MATLAB. Much of this interface capability was formerly referred to under the title of the MATLAB Application Program Interface, or API.
This section introduces the external interfaces that are available with MATLAB. It also contains numerous C and Fortran examples that highlight this functionality. The topics included are
Importing and Exporting Data
Describes techniques for importing data to and exporting data from the MATLAB environment using MAT-files.
MATLAB Interface to Generic DLLs
Describes how to call functions in external, shared .dll libraries from MATLAB.
Calling C and Fortran Programs from MATLAB
Describes how to build C and Fortran subroutines into callable MEX files.
Creating C Language MEX-Files
Shows the components of a Fortran Mex-file and provides specific examples of MEX-files written in Fortran.
Creating Fortran MEX-Files
Shows the components of Fortran MEX-file and provides specific examples of MEX-files written in Fortran.
Calling MATLAB from C and Fortran Programs
Discusses how to use the MATLAB engine library to call MATLAB from C and Fortran programs.
Calling Java from MATLAB
Describes how to use the MATLAB interface to Java classes and objects.

CEU
Universidad
San Pablo

CURSO de UTILIZACIÓN PRÁCTICA de MATLAB

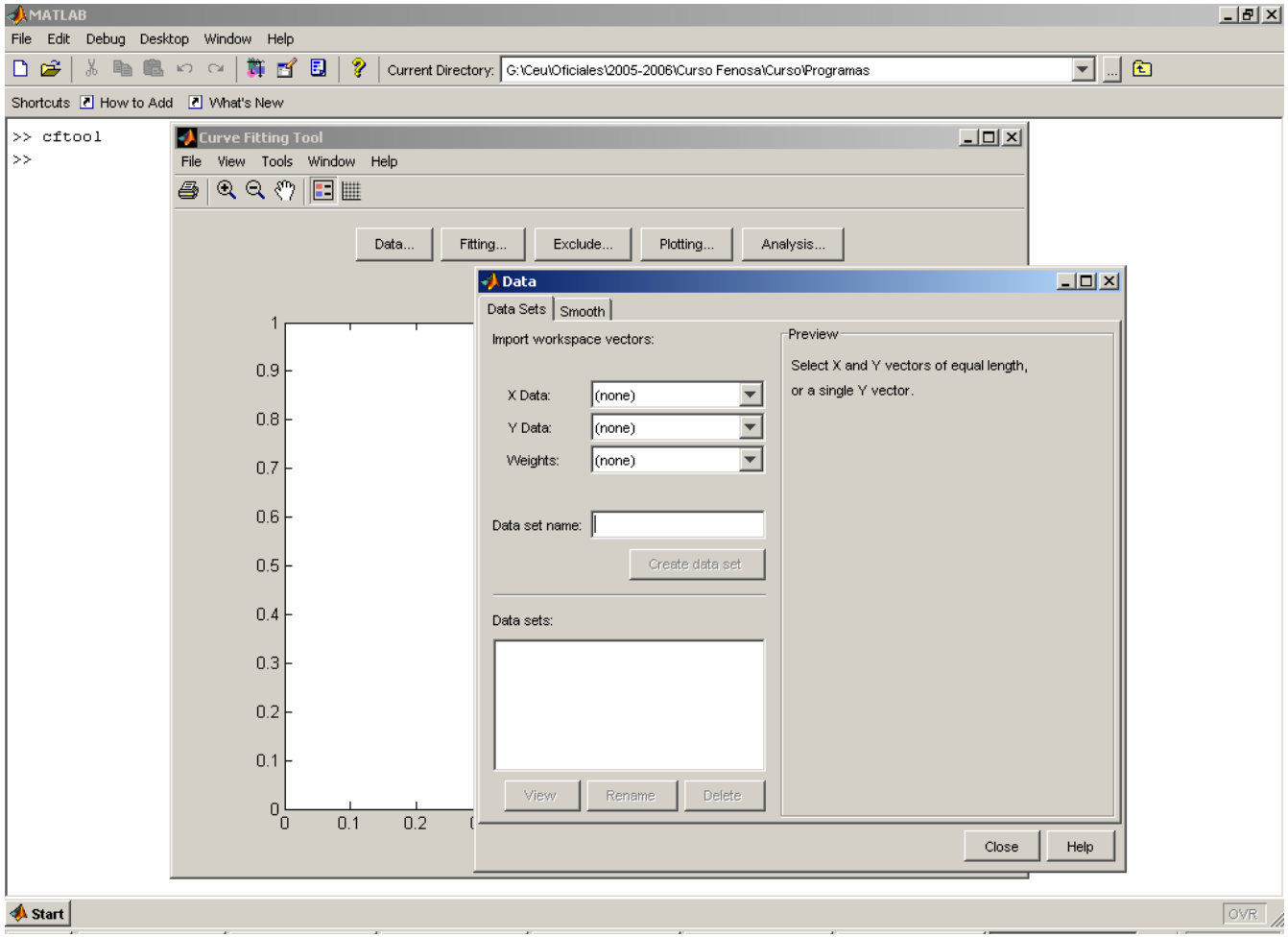
Sesión 7

Carlos Óscar Sánchez Sorzano, Ph.D.
Madrid, July 17th 2006

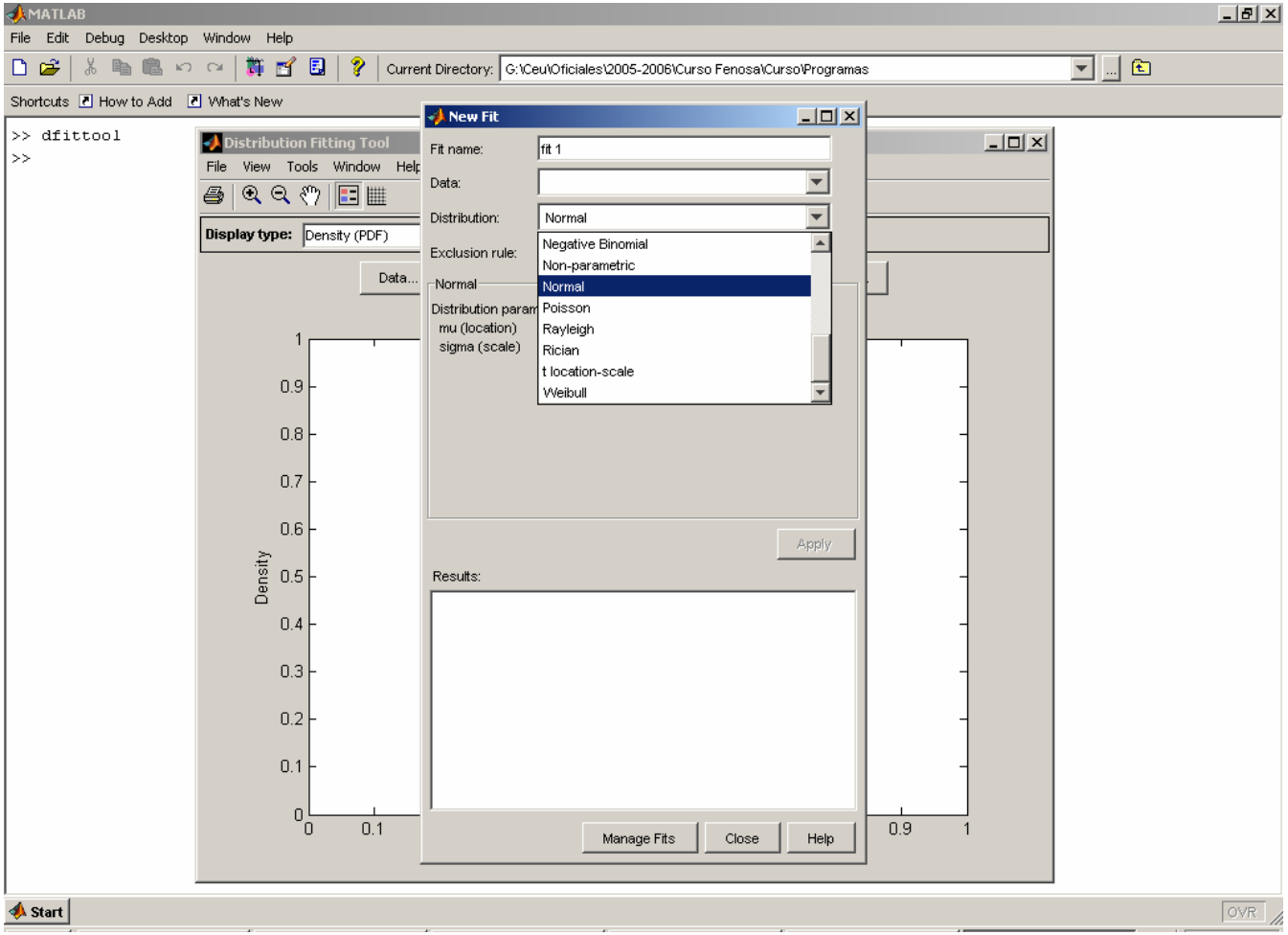
Cronograma del curso

- Día 1: Operaciones con matrices y vectores. Funciones de librería.
- Día 2: Otros tipos de datos en MATLAB. Programación en MATLAB.
- Día 3: Gráficos bidimensionales. Gráficos tridimensionales.
- Día 4: Interfaces de usuario en MATLAB. Generación de programas autónomos
- Día 5: Librerías de interés práctico
- Día 6: Interacción de MATLAB con Office y Visual Basic
- **Día 7: Desarrollo de un proyecto**

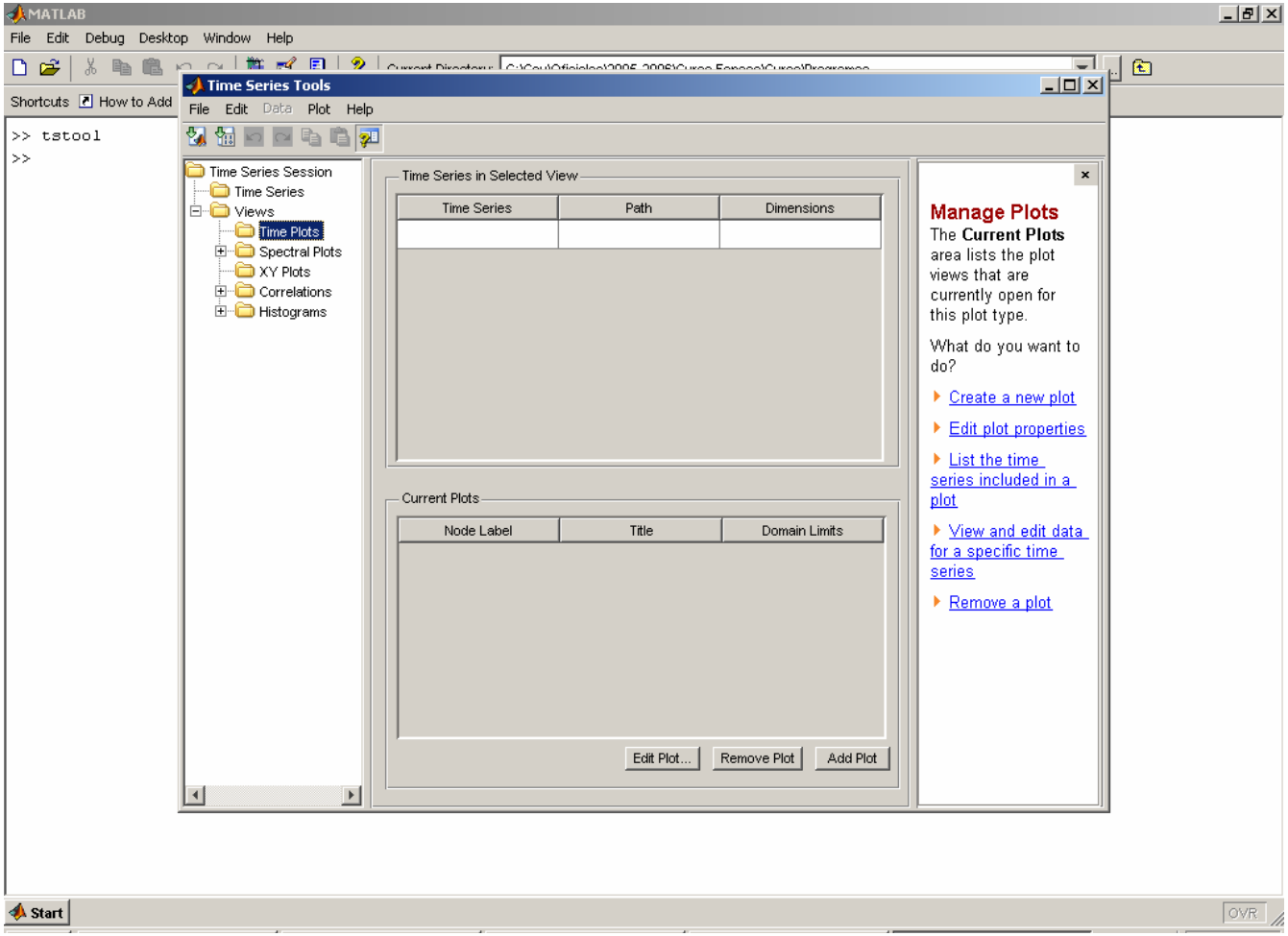
Algo de series temporales: Curve Fitting Tool



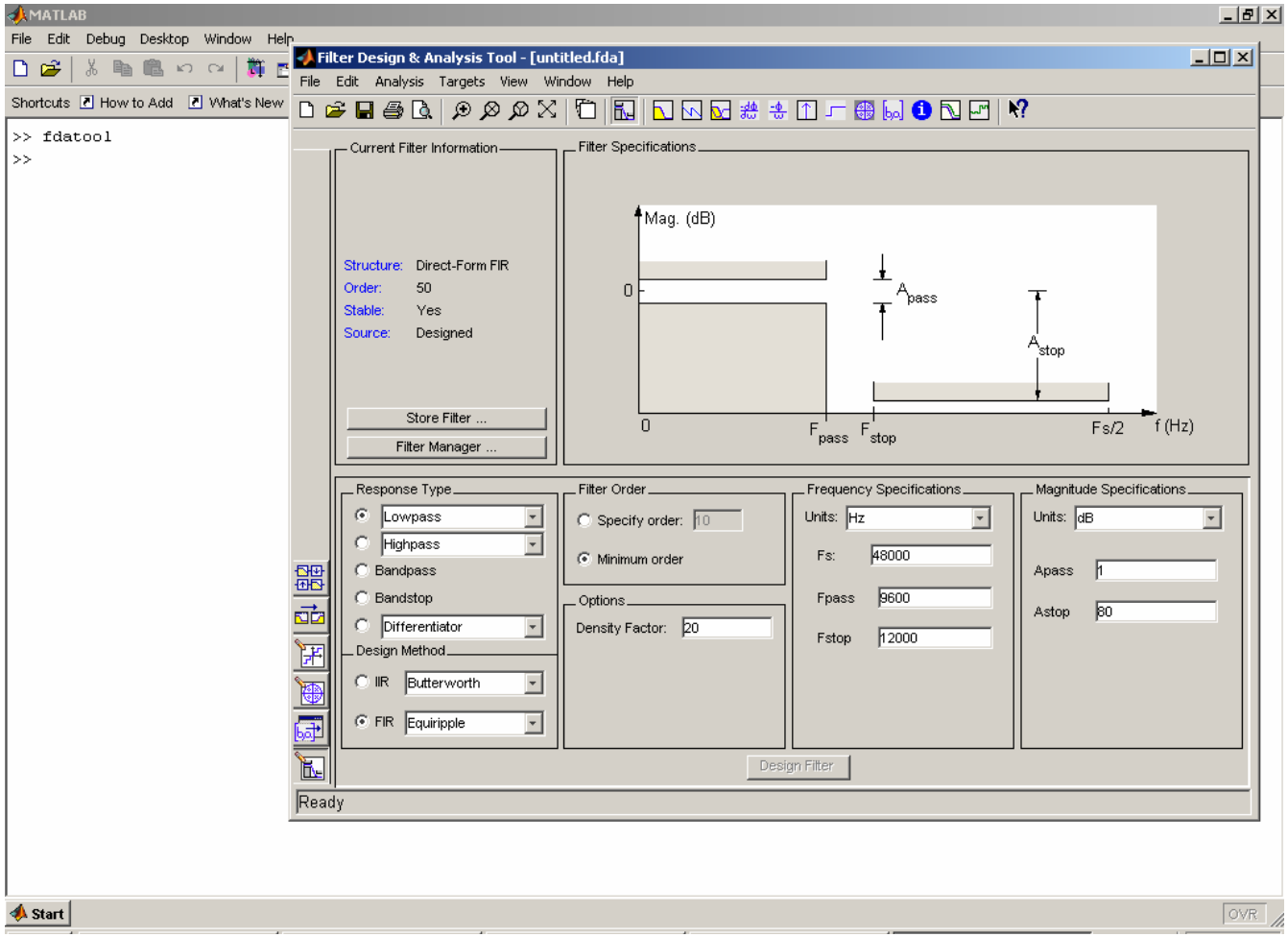
Algo de series temporales: Distribution Fitting Tool



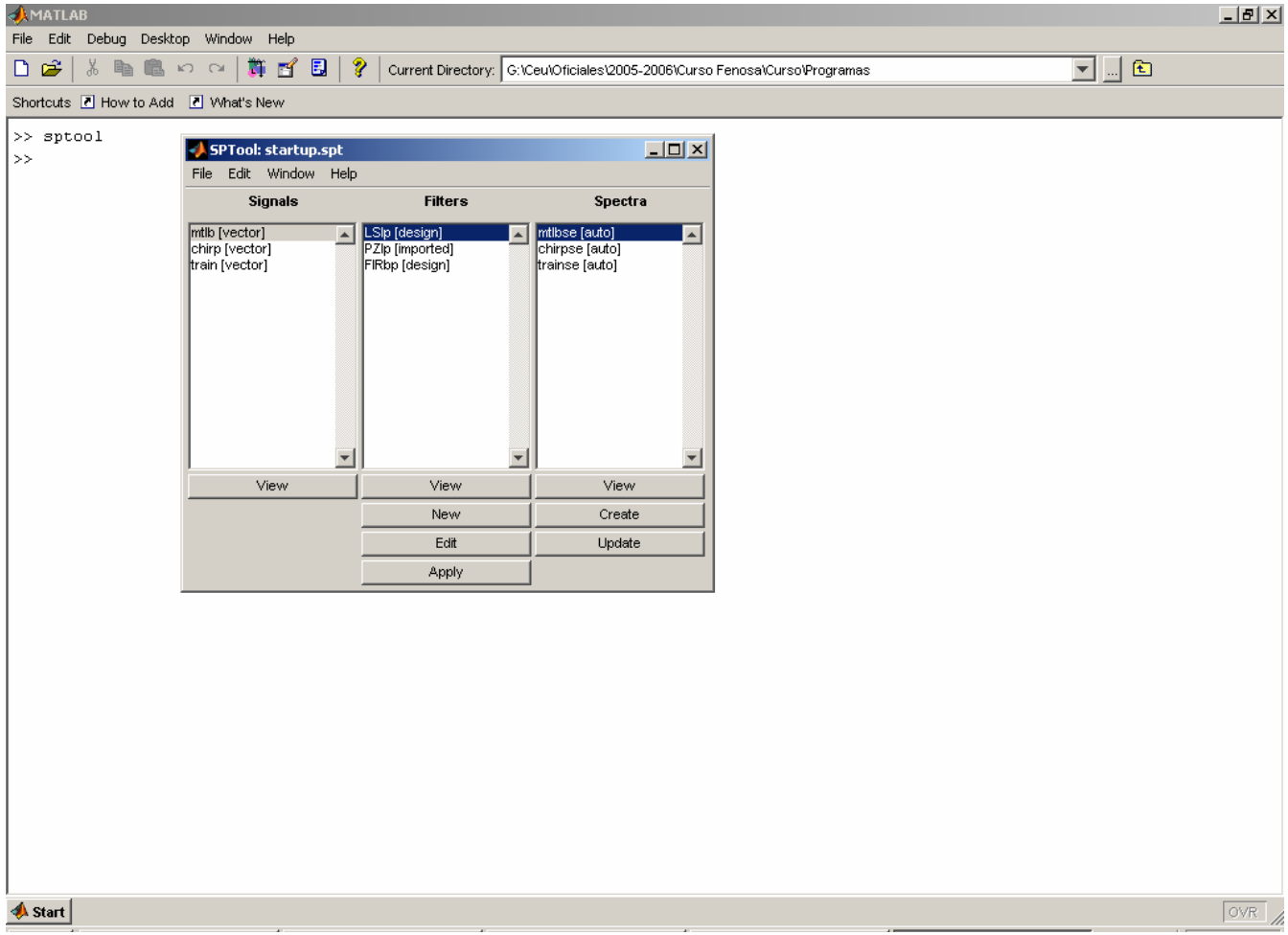
Algo de series temporales: Time Series Tool



Algo de series temporales: Filter Design Tool



Algo de series temporales: Signal Processing Tool



Su nombre aquí