

Informática II para Bachillerato

Introducción a la programación en MATLAB

José Luis Alonzo Velázquez

CIMAT

Sesión 1

Inicio de curso

Temario:

- 1 MATLAB
- 2 Programación en MATLAB
- 3 Variables
- 4 Ciclos
- 5 Funciones (recursividad)
- 6 Graficación 2D y 3D
- 7 Algoritmos
- 8 Manejo de datos
- 9 Solución de problemas
- 10 Hojas de Cálculo (Excel)

Evaluación

Tipo	Frecuencia	Porcentaje de la evaluación final
Tareas	≈ 1 por semana	40 %
Practicás	≈ 1 por semana	0 %
Proyecto(s)	en 2 partes	30 %
Exámenes	3 exámenes	30 %

Noticias y Material del Curso

Información sobre la clase centralizada en mi página web

http://www.cimat.mx/~pepe/cursos/informaticaII_bachillerato_2016/

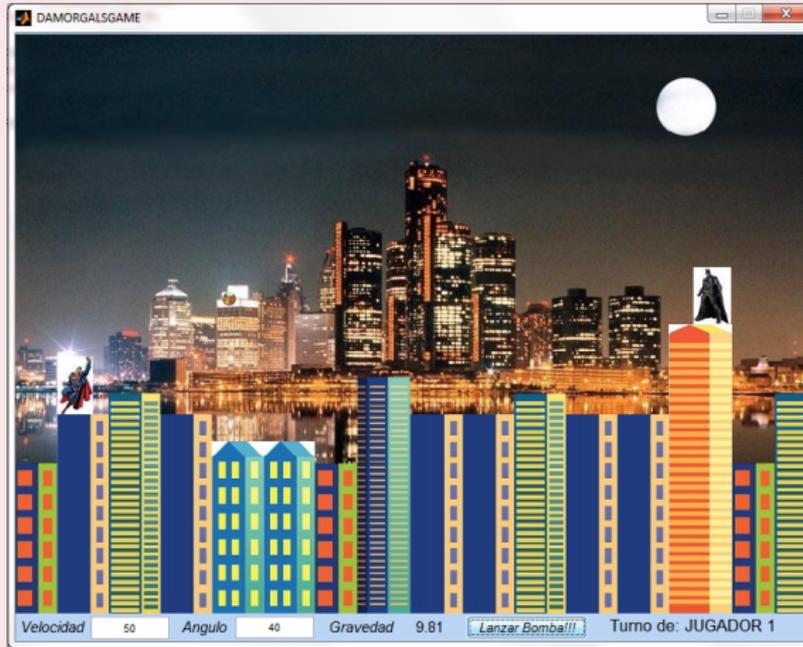
- 1 tareas y clases en pdfs,
- 2 notificaciones diversas, errata sobre las tareas. . .

Por clase tendremos a lo sumo 2 descansos de aproximadamente 10 min.

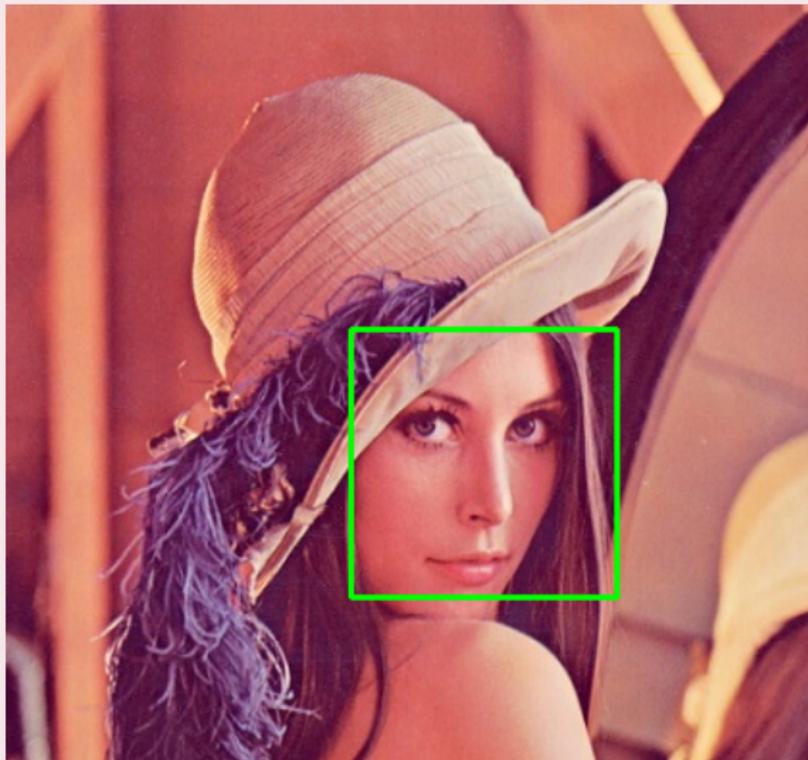
Sobre la clase: Tareas

- 1 Tienen 1 semana para terminarlas, por ejemplo:
- 2 miércoles 7 octubre, 15h45 – > martes 14 octubre, 23h59.
- 3 Las tareas se subirán a la siguiente página web(a partir de la tarea 3 para entonces les asignaré un usuario):
<http://www.pepemxl.my-place.us/evaluador/>
- 4 Formato de entrega de tareas via mail:
ApellidoPaterno_PrimerNombre_Tarea###.zip
ejemplo: Razo Rodriguez Daniel Alberto tarea 3
el formato correspondiente seria: Razo_Daniel_Tarea_03.zip

Juegos



Detector de rostros



Objetivo del Curso

El curso fue diseñado con el propósito de desarrollar en el alumno, las capacidades y habilidades necesarias para diseñar y realizar programas de cómputo que le sirvan de apoyo sustancial a su desempeño académico y profesional, mediante el ejercicio de proyectos de cómputo con aplicación a diversos temas de la ingeniería, dependiendo del empeño de los alumnos el curso podría cambiar su orientación para serles de gran ayuda en sus futuras carreras.

Lenguaje de Programación

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. Ahora a cada instrucción le corresponde una acción de procesador. El lenguaje utilizado por el procesador se denomina lenguaje máquina. Se trata de datos tal como llegan al procesador, que consisten en una serie de 0 y 1 (datos binarios).

- 1 Monitor
- 2 Placa base
- 3 Procesador
- 4 Puertos ATA
- 5 Memoria principal (RAM)
- 6 Placas de expansión
- 7 Fuente de alimentación
- 8 Unidad de almacenamiento óptico
- 9 Disco duro, Unidad de estado sólido
- 10 Teclado
- 11 Ratón

Lenguaje Máquina

El lenguaje máquina, por lo tanto, no es comprensible para los seres humanos, razón por la cual se han desarrollado lenguajes intermediarios comprensibles para el hombre. El código escrito en este tipo de lenguaje se transforma en código máquina para que el procesador pueda procesarlo.

Primer Lenguaje

El **ensamblador** es muy similar al lenguaje máquina, pero los desarrolladores pueden comprenderlo. No obstante, este lenguaje se parece tanto al lenguaje máquina que depende estrictamente del tipo de procesador utilizado (cada tipo de procesador puede tener su propio lenguaje máquina). Así, un programa desarrollado para un equipo no puede ser portado a otro tipo de equipo.

Portabilidad

El término **portabilidad** describe la capacidad de usar un programa de software en diferentes tipos de equipos. Para poder utilizar un programa de software escrito en un código ensamblador en otro tipo de equipo, a veces será necesario volver a escribir todo el programa!!!!

Interpretado

Un lenguaje de programación es, por definición, diferente al lenguaje máquina. Por lo tanto, debe traducirse para que el procesador pueda comprenderlo. Un programa escrito en un lenguaje interpretado requiere de un programa auxiliar (el intérprete), que traduce los comandos de los programas según sea necesario.

Compilado

Un programa escrito en un lenguaje **compilado** se traduce a través de un programa anexo llamado compilador que, a su vez, crea un nuevo archivo independiente que no necesita ningún otro programa para ejecutarse a sí mismo. Este archivo se llama ejecutable.

Un programa escrito en un lenguaje compilado posee la ventaja de no necesitar un programa anexo para ser ejecutado una vez que ha sido compilado. Además, como sólo es necesaria una traducción, la ejecución se vuelve más rápida. Sin embargo, no es tan flexible como un programa escrito en lenguaje interpretado, ya que cada modificación del archivo fuente (el archivo comprensible para los seres humanos: el archivo a compilar) requiere de la compilación del programa para aplicar los cambios.

BASIC	Programación educativa	Interpretado
C	Programación de sistema	Compilado
C++	POO	Compilado
Cobol	Administración	Compilado
Fortran	Cálculo	Compilado
Java	Programación Internet	Intermediario
MATLAB	Cálculos matemáticos	Interpretado
Pascal	Educación	Compilado
PHP	Desarrollo de sitios	Interpretado
Perl	Procesamiento de cadenas	Interpretado
Python	Cálculos matemáticos	Interpretado

Por que aprender MATLAB/SCILAB

Este es un buen lenguaje para aprender ya que cuenta con una sintaxis sencilla que facilita su aprendizaje. Además evita el uso de punteros (una cosa normalmente complicada para los alumnos, incluso para los profesionales). La ventaja es que estos lenguajes pueden ser utilizados en diversas plataformas y que, una vez dominados. Estos lenguajes son aún muy utilizados en el mundo profesional de la ingeniería.

Sobre visual basic

La mayoría de los profesionales que han trabajado con él lo desaconsejan. Este crea malas costumbres de programación (no es necesaria la declaración de variables, ninguna noción de valor/referencia, tentativa fracasada de programar con objetos). Para terminar, no existe ninguna versión gratuita, por lo tanto necesariamente hay que pagar. El sólo punto positivo que le reconocería es su lado lúdico y la facilidad de hacer interfaces gráficas para Windows.

¿Qué es MATLAB?

MATLAB (abreviatura de MATrix LABoratory, “laboratorio de matrices”) es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio. Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux. Es una de las herramientas comerciales más populares para cálculos numéricos en matemáticas, estadística, ciencias, ingeniería, etc. Para la siguiente clase dire en la página cual será la versión oficial del curso.

¿Qué es SCILAB?

Scilab es un software matemático, con un lenguaje de programación de alto nivel, para cálculo científico, interactivo de libre uso y disponible en múltiples sistemas operativos (Mac OS X, GNU/Linux, Windows). Desarrollado por INRIA (Institut National de Recherche en Informatique et en Automatique) y la ENPC (École Nationale des Ponts et Chaussées) desde 1990 hasta 2008, después otros tomaron su desarrollo.

¿Por qué SCILAB?

La licencia de MATLAB es sumamente costosa, por lo cual es complicado disponer de esta aplicación, sin embargo SCILAB es gratuita y ofrece muchas de las funcionalidades de MATLAB. Así que para quien lo desee le permitiré entregar sus tareas en SCILAB.

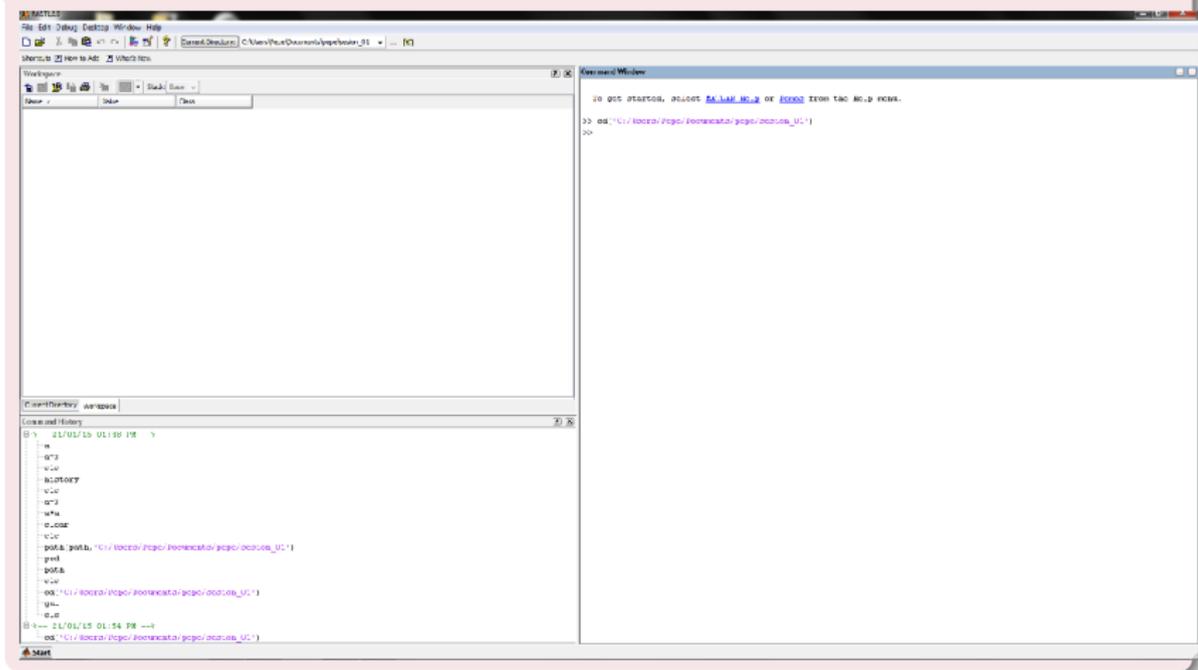
Para descargar SCILAB:

<http://www.scilab.org/download/5.5.1>

Equivalencias entre MATLAB y SCILAB

http://help.scilab.org/docs/5.5.1/fr_FR/section_36184e52ee88ad558380be4e92d3de21.html

MATLAB 7.0(2004)



Declaración de Variables

Una de las ventajas de matlab es que no es necesario declarar variables, las variables son creadas al vuelo, es decir mientras se utilizan obtienen las propiedades necesarias.

Ejemplo:

```
a = 3
```

```
a =
```

```
3
```

Si uno no desea que la asignación o función no sea mostrada en pantalla, basta con usar ;

```
a = 3;
```

Operadores Básicos

+,-,*,/,^,.

Algunos comandos básicos para esta clase:

- pwd nos dice el directorio actual de trabajo.
- cd('C:/Users/Pepe/Documents/pepe/sesion_01') nos permite cambiar el directorio actual de trabajo.
- path(path, 'C:/Users/Pepe/Documents/pepe/sesion_01') nos permite agregar un directorio al path de MATLAB(SOLO MATLAB)
- clc limpia pantalla.
- clear limpia memoria.
- who, whos permite saber que variables se encuentran actualmente usadas.
- % Con este carácter indicamos en MATLAB que la línea será ignorada.(En scilab es //)

Algunos comandos básicos para esta clase:

- `disp('Hola Mundo')` esta instrucción sirve para mandar un mensaje a consola.
- `sqrt(x)` Calcula la raíz cuadrada de x .
- `sin(x)` Calcula la función $\sin(x)$.
- `cos(x)` Calcula la función $\cos(x)$.
- `rand()` Genera un numero aleatorio.
- `inv()` Genera el inverso de una matriz cuadrada.
- `eps` Nos dice el cero maquina.(Solo MATLAB)
- `realmin,realmax`

Formatos Numéricos:

- short coma fija con 4 decimales (defecto)
- long coma fija con 15 decimales
- hex cifras hexadecimales
- bank números con dos cifras decimales
- short e notación científica con 4 decimales
- short g notación científica o decimal, dependiendo del valor
- long e notación científica con 15 decimales
- long g notación científica o decimal, dependiendo del valor
- rational expresa los números racionales como cocientes de enteros

Realizar las siguientes operaciones en MATLAB:

- `disp('Hola Mundo');`
- $3+4$
- $3-4$
- $3*4$
- $3/4$
- 3^4
- `format rational;`
- $3/4$
- `long g;`

Archivos .m

Los archivos .m se pueden utilizar para guardar una serie de instrucciones que queremos realizar secuencialmente. Al teclear el nombre del archivo o ejecutarlo desde pantalla, se correrán todas las instrucciones en él.

Arreglos de datos, ejemplo:

```
a=[1:10]
```

```
a=[1:2:10]
```

```
a=['a':'z']
```

```
a=['a':2:'z']
```

```
a=rand(1,5)
```

```
size(a)
```

Diferencia con respecto a otros lenguajes:

- División por cero.
- Errores lógicos. `a='hola'; a+a`
- Redefinición de variables.
- Lógica 1.

Comparaciones

TRUE(T o 1) FALSE(F o 0)

a = 3

a > 5

a < 5

a == 5

a ~= 5

Actividad:

Crea una variable que contenga los números del 1 al 10. Y compara dicha variable con el número 3.^a

^aSe vale preguntar cualquier cosa que tenga que ver con el curso, me reservo el derecho de responder durante los exámenes.

Operadores Lógicos

- `&` and (función equivalente: `and(A,B)`). Se evalúan siempre ambos operandos, y el resultado es `true` sólo si ambos son `true`.
- `&&` and breve: si el primer operando es `false` ya no se evalúa el segundo, pues el resultado final ya no puede ser más que `false`.
- `|` or (función equivalente: `or(A,B)`). Se evalúan siempre ambos operandos, y el resultado es `false` sólo si ambos son `false`.
- `||` or breve: si el primer operando es `true` ya no se evalúa el segundo, pues el resultado final no puede ser más que `true`. `~` negación lógica (función equivalente: `not(A)`)
- `xor(A,B)` realiza un “or exclusivo”, es decir, devuelve 0 en el caso en que ambos sean 1 ó ambos sean 0.

¿Dudas?