



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

NOCIONES DE MATLAB

N° de práctica: 1

Tema: Nociones de Matlab

Nombre completo del alumno		Firma
N° de brigada:	Fecha de elaboración:	Grupo:

Elaborado por:	Revisado por:	Autorizado por:	Vigente desde:
Profesor 1	Ing. Benjamín Ramírez Hernández	Dr. Paul Rolando Maya Ortiz	28 de noviembre de 2015



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

1. Seguridad en la ejecución

	Peligro o Fuente de energía	Riesgo asociado
	Descargas eléctricas	Daño al equipo

CONTENIDO

- I. Objetivo
- II. Antecedentes
- III. Material y Equipo
- IV. Desarrollo de la Práctica
 - IV.1 Introducción
 - IV.2 Estructura de los Datos en MatLab
 - IV.3 Polinomios y Funciones de Transferencia
 - IV.4 Representaciones Gráficas
- V. Hoja de Resultados



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

I. Objetivo

Al finalizar la práctica el estudiante utilizará los comandos básicos de **MATLAB** para el manejo de funciones de transferencia, para graficar funciones así como para el análisis de sistemas de control.

II. Antecedentes

El estudiante deberá contar con conocimientos matemáticos relativos a los cursos de Álgebra, Ecuaciones Diferenciales, conocer el manejo matemático de los sistemas lineales invariantes en el tiempo así como las bases de la solución numérica de problemas matemáticos. Deberá contar con conocimientos básicos de programación en algún lenguaje de alto nivel. Se recomienda que cuente con una computadora con la versión R2007a de **MATLAB**.

III. Material y Equipo

Computadora PC y software MatLab versión R2007a.

IV. Desarrollo de la Práctica

I Actividad 1

IV.1 Introducción

MATLAB es una herramienta de simulación y cálculo matemático de tipo numérico (básicamente matricial) que puede utilizarse para resolver modelos que representan sistemas dinámicos. La capacidad de cálculo de **MATLAB** se basa en sus múltiples funciones (comandos). En esta práctica se mostrará la utilidad de algunas de ellas.



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

Al abrir MATLAB hay tres pantallas aparece una pantalla con tres ventanas:

La primera ventana de la izquierda, es la ventana de arranque (launch pad) en donde se localizan todos los directorios y demos. La segunda ventana a la izquierda, donde se lleva un registro de los comandos que se han ejecutado, así como de las variables y parámetros utilizados, ventana de historia de comandos (command history). La tercera ventana a la derecha, la ventana de comandos (command window) se considera la ventana principal y es precisamente donde se declaran las variables y comandos de un programa y donde se ubica el símbolo '>>', como se muestra en la figura 1.

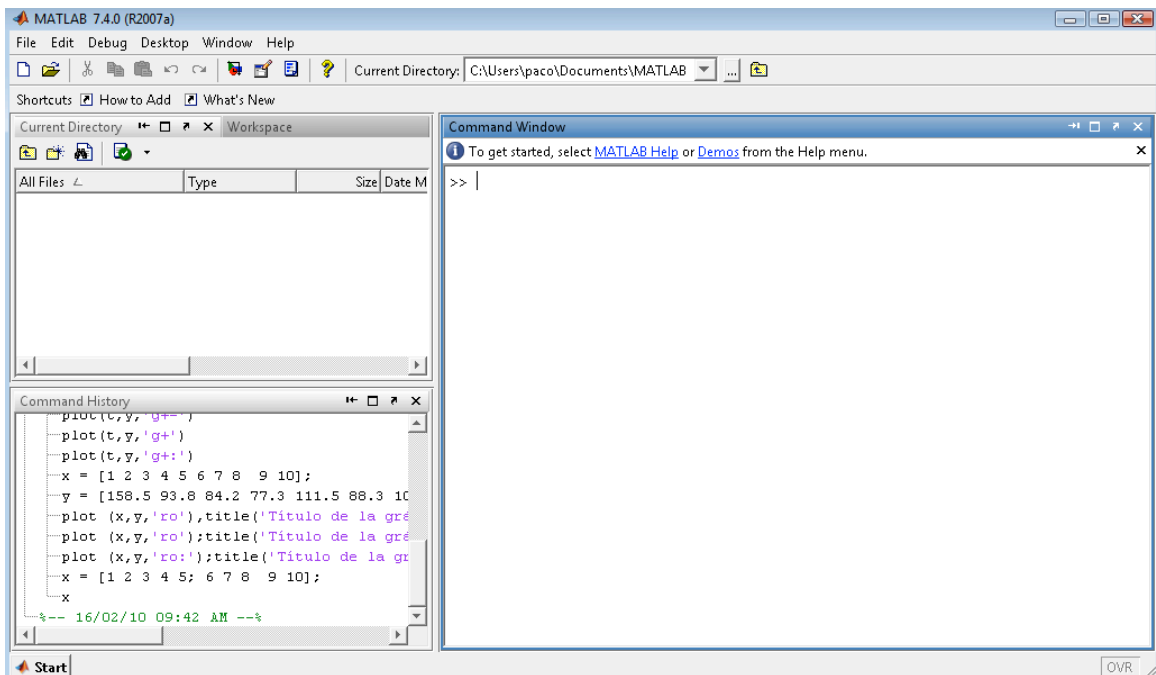


Fig. 1 Pantalla de MATLAB R2007a

El indicador o Prompt representado por el símbolo “>>” en la ventana de comandos indica que MATLAB está en espera de que se introduzca un comando, una variable o la



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

asignación de un parámetro. Para salir de MATLAB cuando sea pertinente use los comandos quit o exit.

La sintaxis general de MATLAB es la siguiente

```
[salida1,salida2, ... ] = nombre de comando (entrada1, entrada2,...);
```

donde las variables de salida se encierran entre paréntesis cuadrados (corchetes) y las de entrada entre paréntesis normales. Para una sola salida los corchetes son opcionales. Si no hubiera al final de la sentencia un punto y coma, el programa nos mostraría el resultado recogido en cada una de las salidas.

El programa MATLAB hace diferencia entre **mayúsculas** y **minúsculas**.

Las variables generadas durante una sesión se van almacenando en el "espacio de trabajo" del propio programa. Si se saliera de éste sin salvarlas previamente las variables se perderían. Para guardarlas se utiliza el comando "save + nombre-archivo". Por ejemplo:

```
» save prac_00
```

MatLab automáticamente agrega la extensión *.mat. Para recuperar los resultados de una sesión anterior se teclaea el comando:

```
» load prac_00
```

y coloca su contenido en el "espacio de trabajo". Para consultar el contenido del "espacio de trabajo" se teclaea el comando:

```
» who ("whos" proporciona información adicional a cada variable)
```

Si se desea borrar alguna de las variables del "espacio de trabajo" se utilizará:



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

```
» clear variable 1
```

Si sólo se tecléa » clear se borrarían todas las variables existentes.

Para consultar los archivos que están contenidos en el "directorio de trabajo" desde el cual se inicia la sesión de MATLAB (c:>matlab\trabajo) se usa el comando:

```
» dir
```

II Actividad 2

IV.2 Estructura de los Datos en MatLab

INGRESO DE DATOS

La forma de ingresar los datos, puede ser de alguno de los tipos siguientes: Un escalar, un vector o una matriz.

Un sólo dato, por ejemplo $A=8.7$ es un **escalar**.

Si una matriz tiene una sola renglón (escribiendo los caracteres separados por comas o espacios) o una sola columna (escribiendo los caracteres separados por punto y coma), entonces se ingresó un **vector**; conocidos como un vector renglón o un vector columna.

```
B=[12.56 36.47]
```

MATRIZ



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

El tamaño de una matriz se especifica por el número de renglones y de columnas; así, W es una matriz de 3×3 . Un dato de una matriz se puede identificar por los subíndices; así W_{13} representa el dato 90 de la matriz W , si una matriz contiene m renglones y n columnas, entonces contiene un total de $m \times n$ valores; así, W es una matriz de tamaño 3×3 .

$$W = \begin{bmatrix} 1 & 34 & 90 \\ 59 & 12 & 67 \\ 22 & -9 & 37 \end{bmatrix}$$

ESTILO

Matlab es sensible a la diferencia entre mayúsculas y minúsculas, así que los nombres “Dato, DATO y dato” representan tres variables distintas.

DEFINICIÓN DE UNA MATRIZ

La forma más sencilla de definir una matriz es usar una lista de números, como:

$$A=[8.7]$$

$$B=[12.56 \ 36.47]$$

La matriz W se puede ingresar como:

$$W = [1 \ 34 \ 90; 59 \ 12 \ 67; 22 \ -9 \ 37];$$

$$W = [1 \ 34 \ 90 \\ 59 \ 12 \ 67 \\ 22 \ -9 \ 37]$$

PUNTOS SUSPENSIVOS



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

Si hay demasiados números en un renglón de una matriz para que quepan en una línea, podemos continuar la instrucción en la siguiente línea, **pero se requiere una coma y tres puntos** al final de la línea para indicar que la renglón debe continuar.

Ejemplo:

$H = [-2,0,-3,4,-3,-4,5,0,0,2,1,1,1,3,4,-0.2]$

Que también se puede escribir como:

$H = [-2,0,-3,4,-3,-4,...$

$5,0,0,2,1,1,1,3,4,-0.2]$

MATLAB también permite definir una matriz que ya se definió:

Ejemplo:

$B = [1.5,4.1]$

$D = [-4,B]$ este comando equivale a

$D = [-4,1.5,4.1]$

También podemos modificar los valores de una matriz o agregar valores adicionales usando una referencia a un lugar específico. Por ejemplo:

$D(3) = 6$;Cambia el tercer valor de la matriz D del valor 4.1 por 6, para quedar como:

$D = [-4,1.5,6]$

Así también logramos extender una matriz definiendo nuevos elementos.



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

Si ejecutamos el siguiente comando

$D(4)=2.5$; La matriz D tendrá cuatro valores en lugar de tres, así D se verá como

$D = [-4, 1.5, 6, 2.5]$;

Escriba las siguientes sentencias:

- 1.- $B=[2;4;6;10]$ % Lo que define un vector columna.
- 2.- $C=[5,3,5;6,2,-3]$ % Lo que define una matriz de 2x3.
- 3.- $E=[3,5,10,0;0,0,...$
 $0,3;3,9,9,8]$ % La coma y tres puntos para
continuar la línea.
- 4.- $T=[4,24,9]$
 $Q=[T,0,T]$ % Intercala el valor de 0 entre los dos vectores.
- 5.- $V=[C(2,1);B]$ % De la matriz C selecciona el valor de C_{21} y lo
agrega al vector B.
- 6.- $A(2,1)=-3$ % Crea una matriz A y le asigna el Valor de -3
al elemento a_{21} .

EL OPERADOR DE DOS PUNTOS



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

Es útil para generar matrices nuevas.

Si se usa el operador dos puntos para separar dos enteros, el operador de dos puntos generará todos los enteros entre los dos enteros especificados.

`n=1:10` % Este operador es especialmente útil para generar
los índices de tiempo de una señal en tiempo discreto.

También se usa el operador dos puntos para separar tres números, el operador de dos puntos generará valores entre el primer número y el tercero, usando el segundo número como incremento:

`t=0.0:0.5:6.0` % Este operador es especialmente útil para generar los
índices de tiempo continuo o un dominio de una
función analógica.

El incremento también puede ser negativo

`r=15:-1:0`

FUNCIONES ESPECIALES PARA GENERAR MATRICES NUEVAS.



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

El comando *size* devuelve dos argumentos escalares que representan el número de renglones y el número de columnas, podemos usar *size* para generar una matriz de ceros que tenga el mismo tamaño que la otra matriz ejemplo:

```
W=[4,3,2;4,6,3];
```

```
F=zeros(size(W))
```

El comando *ones* genera una matriz que solo contiene unos.

```
C= [1, 2, 3; 4, 2, 5];
```

```
D = ones(size(C))
```

OPERACIONES CON ESCALARES, VECTORES Y MATRICES:

Ingrese el enunciado de asignación siguiente;

```
a=3;
```

```
b=[2,6,4];
```

```
c=[4,10,2];
```

los vectores y matrices pueden multiplicarse por un escalar :



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

$a*b$

ans es el nombre dado a un resultado cuando el usuario no le asigna un nombre.

Un escalar puede sumarse a un vector o matriz:

$a+c$

La suma de un escalar a un vector o matriz sólo añade el escalar a cada elemento del vector o matriz la resta se define de manera similar:

$a-b$

Los vectores y matrices se suman (o restan) como lo hacemos en matemáticas, esto es, los dos vectores o matrices deben tener la misma forma para sumarse o restarse a menos que uno de ellos sea un matriz de $1*1$, un escalar, como lo vimos antes :

$b+c$

$c-b$

Los vectores y las matrices se multiplican de acuerdo a las reglas del álgebra matricial:

$b*c$

Este resultado ilustra un error común en Matlab. Las matrices deben ser conformables para ser multiplicadas utilizando el operador $*$. La premultiplicación de un vector renglón de $1*3$ como c por un vector renglón de $1*3$ como b no está definida. Sin embargo si c se transpusiera a un vector columna $3*1$, se definiría la multiplicación. La transposición se efectúa con el operador $'$ (apostrofe).

c'

$b*c'$

este es el producto bc^T .

OPERACIONES ELEMENTO POR ELEMENTO:



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

A menudo es muy útil multiplicar dos vectores o matrices de la misma forma, elemento por elemento en vez de utilizar las reglas usuales de la multiplicación de matrices. Ese tipo de multiplicación se denomina multiplicación del arreglo en Matlab y se lleva a cabo utilizando el operador `.*`

Las operaciones elemento por elemento, u operaciones de arreglos, no sólo se aplican a operaciones entre dos matrices del mismo tamaño, si no también a operaciones entre un escalar y un no escalar.

Así pruebe las instrucciones de cada uno de los siguientes juegos:

```
A=[4,3,2,4,6];
```

```
C=3.*A
```

```
G=A./5
```

Las vectores resultantes C y G tendrán el mismo tamaño que A.

A fin de ilustrar las operaciones de arreglos para vectores, considere los dos siguientes vectores renglón:

```
B=[-6,3,-2,4,6];
```

Calculemos el producto de arreglos de A y B usando las siguientes instrucciones:

```
L=A.*B
```

El comando de división de arreglos,

```
F=A./B
```

Asimismo la exponenciación de arreglos también es una operación elemento por elemento ejemplos;



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

$$Z=A.^2$$

$$M=A.^B$$

También podemos usar una base escalar con un exponente vector

como;

$$P=3.^A$$

Los ejemplos anteriores utilizaron vectores, pero las mismas reglas se aplican a matrices con renglones y columnas, como lo podrás constatar declarando la matriz 'd' ejecutando las operaciones indicadas:

$$d=[1:6;-1:-1:-6];$$

$$f=d.*6$$

$$w=d.^2$$

IV.3 Polinomios y Funciones de Transferencia

MatLab representa a los polinomios como si fueran vectores, los cuales contienen los coeficientes del polinomio ordenados de la mayor a la menor potencia. Por ejemplo, para el polinomio, $s^3 + 2s^2 + 3s + 4$ se debe crear el vector.

$$\gg p = [1 \ 2 \ 3 \ 4];$$

Las raíces de la ecuación algebraica que se obtiene al igualar el polinomio a cero se pueden obtener utilizando la instrucción:

$$\gg \text{roots}(p)$$

si se cuenta con las raíces se puede formar el vector que representa al polinomio en MatLab mediante la instrucción **poly(•)**, por ejemplo:

$$\gg p2 = \text{poly}([-1 \ -2])$$



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

Para evaluar el polinomio en un punto se utiliza la instrucción:

» `ps=polyval(p,s)`

donde p es el vector que representa al polinomio en MatLab y s es el punto en el cual se desea evaluar. Puede evaluarse al polinomio en un conjunto de valores si s se define como un vector.

Para la multiplicación de polinomios se maneja la instrucción:

» `c =conv([1 1],[1 2])`

y para la división:

» `d = deconv([1 3 2],[1 1])`

Para que MatLab presente en forma de fracción se utiliza

» `printsys(num, den)`

Actividad 3

IV.4 Representaciones Gráficas

GRAFICA x-y:



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

Gráficas x vs. y sencillas. Suponga que se desea graficar los resultados de un experimento, y se cuenta con una tabla de datos.

x	y
1	158.5
2	93.8
3	84.2
4	77.3
5	111.5
6	88.3
7	100.1
8	65.6
9	90.5
10	149.4

Pruebe generar esta gráfica: creando un vector llamado x con los Si se agrega un tercer argumento en el comando datos:

$x = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10];$

y un vector llamado y con los datos:

$y = [158.5\ 93.8\ 84.2\ 77.3\ 111.5\ 88.3\ 100.1\ 65.6\ 90.5\ 149.4];$



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

plot (x, y)

Se genera automáticamente la gráfica.

También es posible dibujar una retícula sobre la gráfica mediante el uso del comando:

grid

Varias instrucciones para el análisis de sistemas generan automáticamente gráficas. Un último aspecto que será importante en la obtención de representaciones gráficas es la forma de incluir textos en ellas, las principales instrucciones que pueden utilizarse son

- title escribe un título en la parte superior para nombrar a la gráfica.
- xlabel permite escribir un nombre al eje x de la gráfica.
- ylabel permite escribir un nombre para el eje y de la gráfica.
- legend indica lo que representa cada curva de la gráfica.

Se pueden incluir un Título de la gráfica y etiquetas para los ejes coordenados, de la forma siguiente:

```
title('Titulo de la grafica')
```

```
xlabel('Variable independiente')
```

```
ylabel('Variable dependiente')
```

plot éste define el color y el estilo de la gráfica. Es una cadena la que determina el color de la línea, estilo de la misma y los símbolos (si los hay) utilizado para los puntos de marca, por tanto intente dar elegancia a su grafica anterior agregando al comando plot:

```
plot(x,y,'ro');grid
```



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

LISTA DE CARACTERES

COLOR		MARCADOR		ESTILO DE LINEA	
Y	amarillo	.	punto	-	continua
M	magenta	o	circulo	:	punteada
C	cian	x	marca x	-.	guión-punto
R	rojo	+	más	--	Discontinua (guionada)
G	verde	*	asterisco		
B	azul	s	cuadrado		
W	blanco	d	diamante		
K	negro	v	triangulo(hacia abajo)		
		^	triangulo (hacia arriba)		

Graficas lineales y logarítmicas: La mayor parte de las gráficas que generamos dan por hecho que los ejes x vs. y se dividen en intervalos espaciados uniformemente; estas gráficas se llaman gráficas lineales. Una escala logarítmica (de base 10) es útil cuando una variable abarca varios órdenes de magnitud.

Los comandos MATLAB para generar gráficas lineales y logarítmicas de los vectores x y y son los siguientes:



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

plot(x,y)	Genera una gráfica lineal con los valores de x y y .
semilogx(x,y)	Genera una gráfica de los valores de x y y usando una escala logarítmica para x y una escala lineal para y .
semilogy(x,y)	Genera una gráfica de los valores de x y y usando una escala lineal para x y una escala logarítmica para y .
loglog(x,y)	Genera una gráfica de los valores de x y y usando escalas logarítmicas tanto para x como para y .

Graficas múltiples: Una forma sencilla de generar curvas múltiples en la misma gráfica es usar múltiples argumentos en un comando de graficación, como en:

plot(t,x,t,y) Al ejecutarse este programa, se traza la curva correspondiente a x versus t , y luego se traza en la misma grafica la curva correspondiente y versus t .

Código No.1

```
v1=[0:0.001:.7979];
w1=298;
q=1.6022e-19;
n=1.5;
k=1.38e-23;
num1=q.*v1;
den1=n*k*w1;
M1=num1./den1;
x1=1e-9.*exp(M1);
v2=[0:0.0001:.6379];
w2=358;
num2=q.*v2;
den2=n*k*w2;
M2=num2./den2;
x2=1024e-9.*exp(M2);
```



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

```
plot(v1,x1,'b',v2,x2,'r');title('Ecuación del diodo'); legend('TEMPERATURA 298 K',  
'TEMPERATURA 358 K');grid
```

Subgráficas: El comando subplot permite dividir la pantalla de gráficos en ventanas. Las posibles divisiones pueden ser dos ventanas o cuatro ventanas o incluso 8 ventanas en una hoja.. Los argumentos del comando subplot son tres enteros: (m, n, p). Los dígitos m y n especifican que la ventana de gráficos se divida en una retícula de m por n ventanas más pequeñas, y el dígito p especifica la p-ésima ventana para la gráfica actual (donde se ubica esta gráfica) . Las ventanas se numeran de izquierda a derecha y de arriba abajo.

Por ejemplo el siguiente comando subplot(2,1,1);plot(x,y) especifica que la ventana de gráficos se divida en una gráfica superior y una inferior (dos líneas de gráficas en una columna) y que la gráfica actual plot(x,y) se coloque en la ventana superior ver código No. 2.

El siguiente juego de instrucciones genera cuatro gráficas que ilustran la función del comando subplot empleando escalas lineales y logarítmica.

Código No. 2

```
t=[0:0.001:1];  
f=2;  
w=2*pi*f;  
x=((1/2).*(1+cos(2*w.*t)));  
subplot(2,2,1);plot(t,x,'r','linewidth',2);grid  
subplot(2,2,2);semilogx(t,x,'b','linewidth',2);grid  
subplot(2,2,3);semilogy(t,x,'r','linewidth',2);grid  
20
```



Manual de Prácticas

Elementos de Control

Secretaría/División: División de Ingeniería Eléctrica

Área/Departamento: Control y Robótica

```
subplot(2,2,4);loglog(t,x,'b','linewidth',2);grid
```

Código No. 3

```
t=[-6*pi:0.001:6*pi];  
w1=2/3;  
w2=1/2;  
w3=1/3;  
x1=2.*sin(w1.*t);  
x2=3.*sin(w2.*t);  
x3=4.*sin(w3.*t);  
plot(t,x1,'b',t,x2,'r',t,x3,'k','linewidth',2);title('Cada senoidal completa un numero entero de ciclos');gris
```

V. Hoja de Resultados

Nombre: _____

Incluya los resultados que se obtienen al correr los códigos 1, 2 y 3.