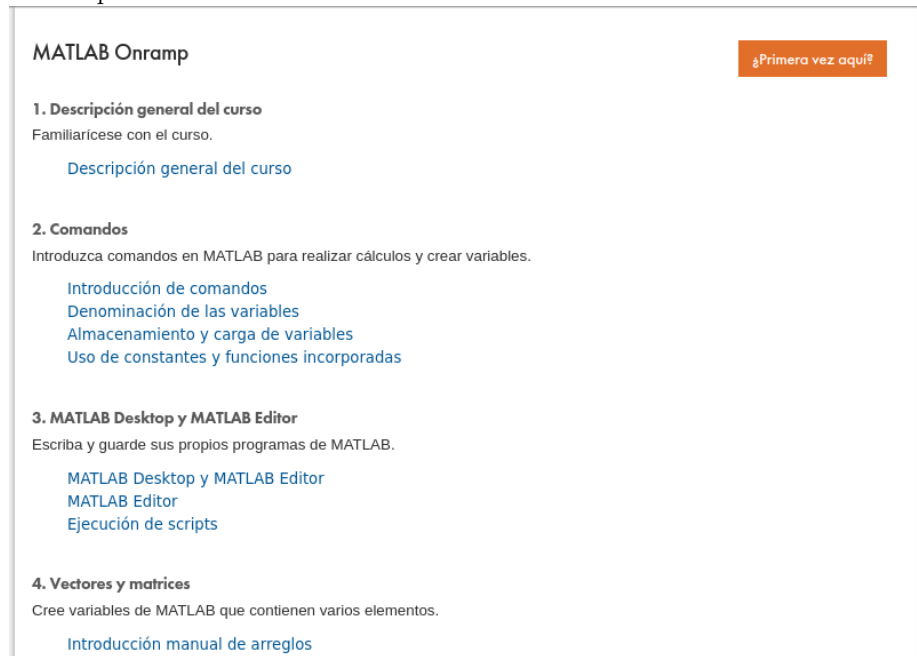


# Tutorial de MATLAB

Emmanuel Peto Gutiérrez

30 de diciembre de 2020

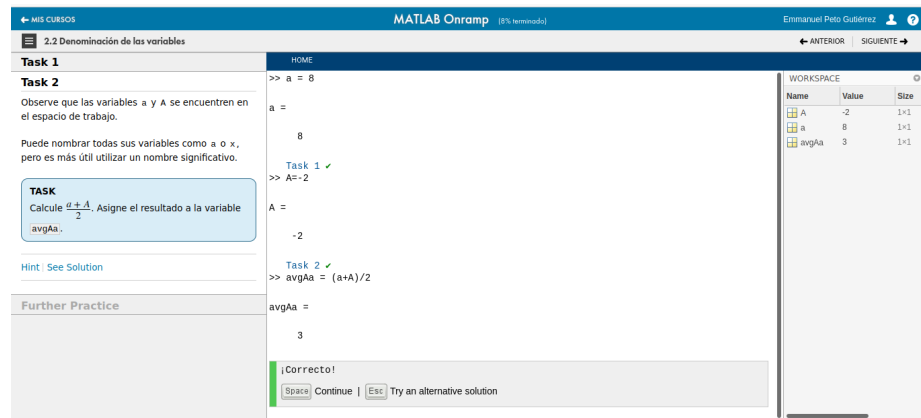
El primer paso es crear una cuenta con el correo de ciencias para tener una licencia en MATLAB. Después, en MATLAB online, se presiona el botón *Learn MATLAB* para ir al tutorial.



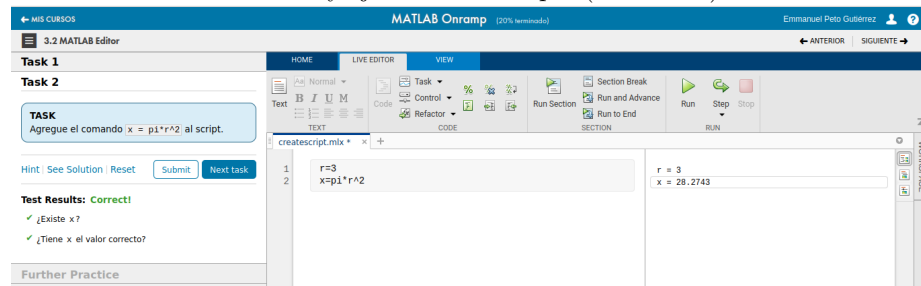
The screenshot shows the MATLAB Onramp tutorial interface. At the top left, it says "MATLAB Onramp". In the top right corner, there is an orange button that says "¿Primera vez aquí?". The main content is organized into four sections:

- 1. Descripción general del curso**  
Familiarícese con el curso.  
[Descripción general del curso](#)
- 2. Comandos**  
Introduzca comandos en MATLAB para realizar cálculos y crear variables.
  - [Introducción de comandos](#)
  - [Denominación de las variables](#)
  - [Almacenamiento y carga de variables](#)
  - [Uso de constantes y funciones incorporadas](#)
- 3. MATLAB Desktop y MATLAB Editor**  
Escriba y guarde sus propios programas de MATLAB.
  - [MATLAB Desktop y MATLAB Editor](#)
  - [MATLAB Editor](#)
  - [Ejecución de scripts](#)
- 4. Vectores y matrices**  
Cree variables de MATLAB que contienen varios elementos.
  - [Introducción manual de arreglos](#)

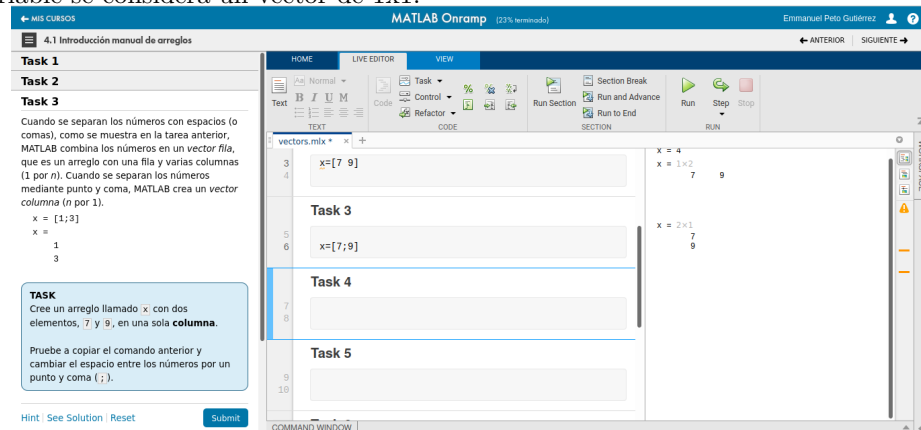
Sección 2: La primera parte es sobre los comandos de MATLAB. Se aprende sobre el uso de variables (como en cualquier lenguaje de programación) y funciones nativas de MATLAB.



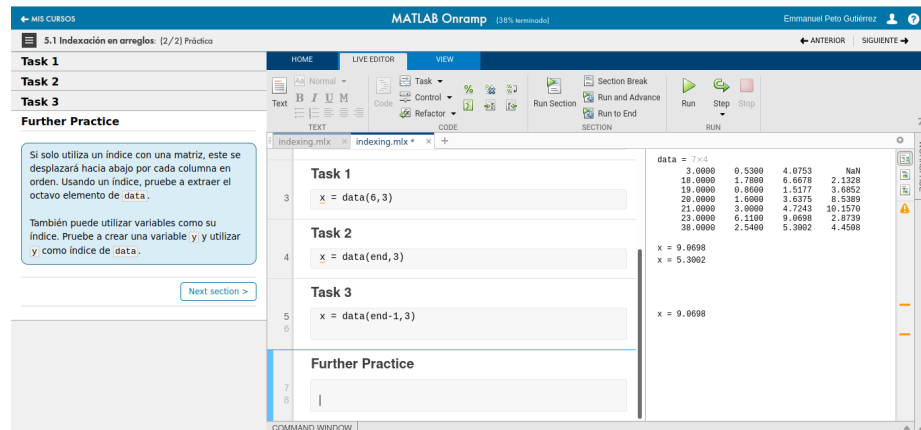
### Sección 3: Sobre la edición y ejecución de scripts (o archivos) de MATLAB.



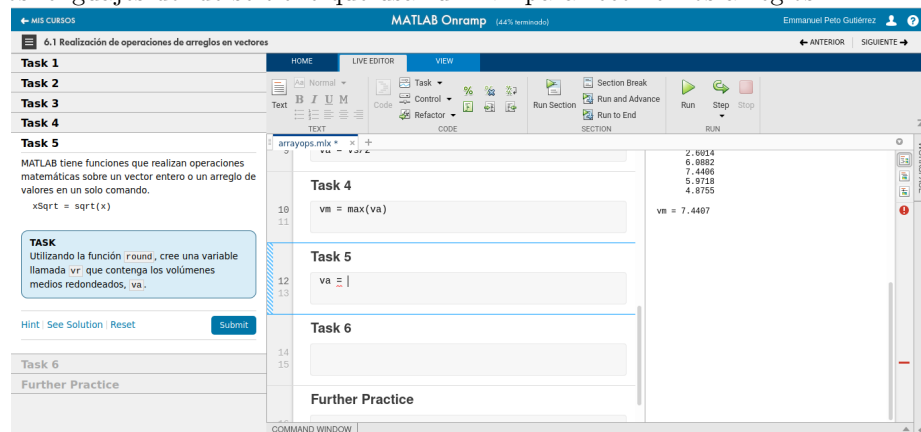
### Sección 4: Es sobre la creación de vectores y matrices. En MATLAB cualquier variable se considera un vector de 1x1.



### Sección 5: Trata sobre la obtención y modificación de datos en un arreglo o matriz.



Sección 6: Enseñan cómo realizar operaciones sobre arreglos con un solo comando; por ejemplo, producto punto de dos vectores. Esto es diferente a otros lenguajes donde se tiene que usar un **for** para recorrer los arreglos.



Sección 7: En MATLAB, una misma función puede dar resultados diferentes, dependiendo del parámetro que se le pase y el tipo de variable a la que se le asigna el resultado.

**Task 1**

La función `size` se puede aplicar a una matriz para producir una o dos variables de salida. Use corchetes `[ ]` para obtener más de una salida.

```
[xrow, xcol] = size(x)
```

**TASK**

Cree las variables `dr` y `dc`, que contienen respectivamente el número de filas y de columnas de la variable `data`.

Hint: See Solution | Reset | Submit

**Task 2**

```
[dr, dc] = size(data)
```

**Task 3**

**Further Practice**

Workspace:

```
data =
    23.0000    6.1100    9.0688    2.8739
    38.0000    2.5400    5.3082    4.4588
    7         4
```

```
ds =
    7     4
```

## Sección 8: Muestra cómo ver la documentación de MATLAB.

**Task 1**

La documentación de MATLAB contiene ejemplos e información que pueden ayudarlo a trabajar en sus propios problemas.

**TASK**

Utilice la documentación de `randi` para completar la tarea siguiente.

Cree una matriz llamada `x` que:

- Contenga números enteros aleatorios entre 1 y 20
- Tenga 5 filas
- Tenga 7 columnas

Hint: See Solution | Reset | Submit | Next task

**Test Results: Correct!**

- ✓ ¿Existe la variable `x`?
- ✓ ¿Está `x` asignada correctamente?

**Task 1**

```
x = randi(20,5,7)
```

**Further Practice**

Workspace:

```
x =
    17     2     4     3    14    16     ...
    19     6    20     9     1    15
     3    11    20    19    17     8
    10    26    19    16    19    14
    13    20    17    20    14     4
```

## Sección 9: Muestra cómo hacer gráficas de puntos y líneas con los valores en un arreglo o una matriz.

**Task 1**

**Task 2**

**Task 3**

**Task 4**

Mientras el estado de persistencia (`hold on`) esté activado, las gráficas seguirán representándose en los mismos ejes. Para volver al comportamiento de representación predeterminado, en el que cada gráfica tiene sus propios ejes, introduzca `hold off`.

**TASK**

Introduzca el comando `hold off`.

Hint: See Solution | Reset | Submit

**Task 3**

```
hold on
plot(sample, mass1, 'ks')
```

**Task 4**

```
hold off
```

**Task 5**

**Task 6**

Workspace:

```
sample =
     0     5    10    15    20    25    30    35    40
mass1 =
     0     5    10    15    20    25    30    35    40
```

## Sección 10: Se realiza un proyecto sobre el consumo de energía eléctrica, usando los conceptos de las secciones anteriores.

**MATLAB Onramp** [61% terminado] Emmanuel Peto Gutiérrez

**10.1 Proyecto: Consumo eléctrico**

**Task 1**

**Task 2**

**Task 3**

**Task 4**

**TASK**  
Los datos comerciales y los datos industriales se almacenan en la segunda y tercera columna, respectivamente. Cree las variables `comm` e `ind`, que contendrán la segunda y tercera columna de `usage`.

Hint: See Solution | Reset | Submit | Next task

**Test Results: Correct!**

- ✓ ¿Existe la variable `comm`?
- ✓ ¿Existe la variable `ind`?
- ✓ ¿Está `comm` asignada correctamente?
- ✓ ¿Está `ind` asignada correctamente?

**Task 5**

**Task 6**

**consumptionplot.mlx**

```

1 load("electricity.mat")
2 usage

Task 2
3 usage(2,3) = 2.74

Task 3
5 res = usage(:,1)

Task 4
7 comm = usage(:,2)
8 ind = usage(:,3)

```

**usage = 23x3**

3.0484	2.5848	2.6480
2.8619	2.5538	2.7480
3.2662	2.7884	2.7345
3.3420	2.8997	2.8919
3.3555	2.9167	2.7960
3.4234	2.9299	2.8579
3.5268	3.1561	2.8597
3.9185	3.3957	2.8757
3.9718	3.4761	2.9984
3.8549	3.4748	2.8998
...	...	...
3.0484	...	...
2.8619	...	...
3.2662	...	...
3.3420	...	...
3.3555	...	...
3.4234	...	...
3.5268	...	...
3.9185	...	...
3.9718	...	...
3.8549	...	...

**res = 23x1**

3.0484
2.8619
3.2662
3.3420
3.3555
3.4234
3.5268
3.9185
3.9718
3.8549
...
3.0484
2.8619
3.2662
3.3420
3.3555
3.4234
3.5268
3.9185
3.9718
3.8549

Sección 11: Es sobre la importación de datos hacia un script de MATLAB. Los datos pueden venir de un archivo de texto, una tabla en excel u otro script de MATLAB.

**MATLAB Onramp** [70% terminado] Emmanuel Peto Gutiérrez

**11.2 Importación de datos como una tabla**  
ella en el panel de salida de un script en tiempo real. Por ejemplo, puede ordenar una tabla utilizando una de sus variables.

Mass	
2.15	Sort Smallest to Largest
11.86	Sort Largest to Smallest
1.30	

Cuando esté satisfecho con la tabla, puede convertir los cambios en permanentes si actualiza el código en el script.

Code ↩  
elements = sortrows(elements, 'Mass')  
Update Code

**TASK**  
Ordene la tabla de la masa más pequeña a la más grande. A continuación, actualice el código del script antes de hacer clic en **Submit** (Enviar).

Hint: See Solution | Reset | Submit

**importedtable.mlx**

```

1 load datafile
2 elements

Task 1
3 d = elements.Density

Task 2 & 3
5 elements.Mass = elements.Density .* elements.Volume1
6 elements = sortrows(elements, 'Mass')

Further Practice
8

```

**d = 7x1**

9.5399
1.7899
8.8600
1.6900
3.0900
6.1190
2.5499

**elements = 7x3 table**

	Element	Density	Volume1
1	"Potassium"	0.8600	1.51
2	"Lithium"	0.5300	4.07
3	"Calcium"	1.6000	3.63
4	"Argon"	1.7800	6.66
5	"Strontium"	2.5400	5.30
6	"Scandium"	3.0900	4.72
7	"Vanadium"	6.1100	9.06

Sección 12: Uso de expresiones lógicas para extraer los datos que cumplan con cierto predicado en un arreglo.

**MATLAB Onramp** [73% terminado] Emmanuel Peto Gutiérrez

**12.1 Indexación lógica**

**Task 1**

**Task 2**

**Task 3**

**Task 4**  
También puede utilizar la indexación lógica con dos vectores diferentes.  
`v = sample(v1 > 6)`  
`s =`  
18  
23

**TASK**  
Cree una variable `s` que contenga los elementos de `sample` correspondientes a cuando `v1` es menor que 4.

Hint: See Solution | Reset | Submit

**logicals.mlx**

```

4 test = p1 < 4

Task 2
6 v1
7 test = v1 < 4

Task 3
9 v = v1(v1 < 4)

Task 4
11 s = sample(v1 < 4)
12

```

**test = logical**

1
---

**v1 = 7x1**

4.0753
6.8678
1.5177
3.6375
4.7243
9.0688
5.3992

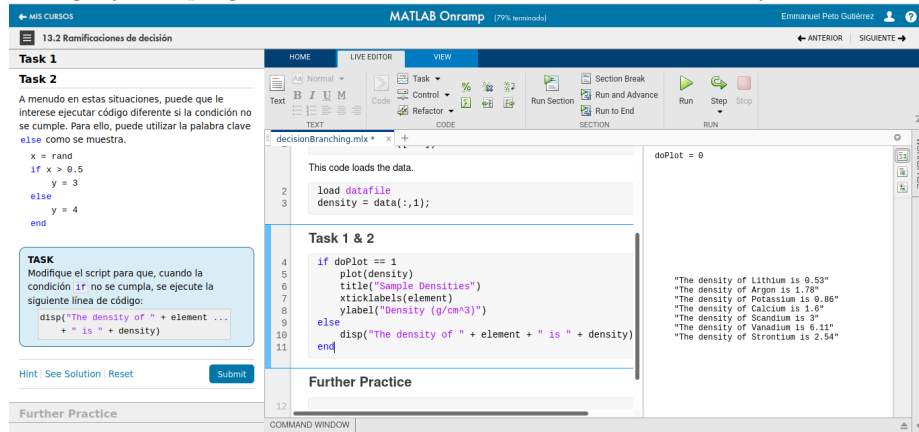
**test = 7x1 logical array**

0
0
1
0
0
0
0

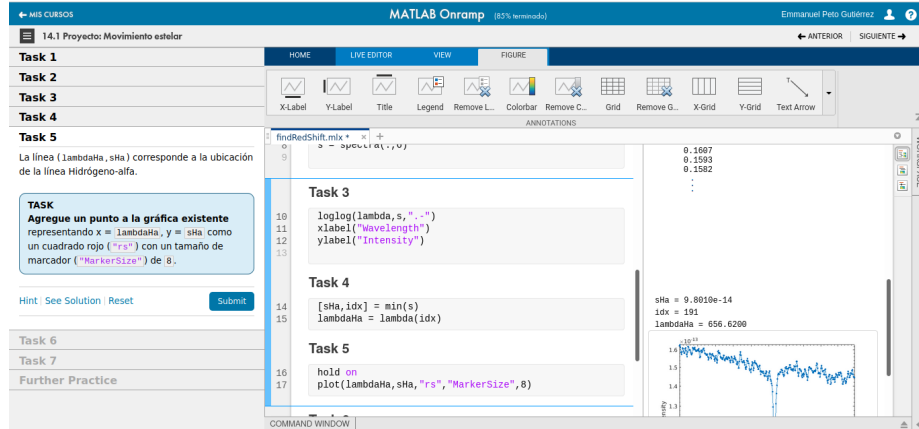
**v = 2x1**

1.5177
3.6375

Sección 13: Muestra el uso de ciertas estructuras de control que se usan en otros lenguajes de programación. Las estructuras son: if, else, elseif y for.



Sección 14: Se realiza un proyecto de "Movimiento estelar" usando lo aprendido hasta ahora.



Al final hay que responder una encuesta corta para completar el tutorial al 100 %.

- ✓ [Importación de datos como una tabla](#)

## 12. Arreglos lógicos

Use expresiones lógicas que lo ayuden a extraer elementos de interés de los arreglos de MATLAB.

- ✓ [Indexación lógica](#)

## 13. Programación

Escriba programas que ejecuten código en función de alguna condición.

- ✓ [Construcciones de programación](#)
- ✓ [Ramificaciones de decisión](#)
- ✓ [Bucles for](#)

## 14. Proyecto final

Incorpore en un proyecto los conceptos que ha aprendido.

- ✓ [Proyecto: Movimiento estelar](#)
- ✓ [Proyecto: Movimiento estelar II](#)

## 15. Conclusión

Aprenda los pasos siguientes y haga comentarios sobre el curso.

- ✓ [Recursos Adicionales](#)
- ✓ [Encuesta](#)