



TECNOLÓGICO  
NACIONAL DE MÉXICO



PROGRAMA INSTITUCIONAL DE FORMACIÓN DOCENTE

# Fundamentos de Machine Learning

Cómo aprenden las máquinas

Ildeberto de los Santos Ruiz

<https://isantosruiz.github.io/home>

Tecnológico Nacional de México  
Instituto Tecnológico de Tuxtla Gutiérrez

**Turix-Dynamics** Diagnosis and Control Group

# Agenda

LUN Introducción. Los datos.

MAR Las matemáticas: Estadística e inferencia bayesiana.

MIÉ Los algoritmos: Regresión.

JUE Los algoritmos: Clustering y clasificación.

VIE Los algoritmos: Deep learning.

# Setup

Para descargar los archivos de datos y materiales que se usarán en el curso:

>\_ Terminal

Clonación inicial

```
git clone https://github.com/isantosruiz/machine
```

# Setup

Para descargar los archivos de datos y materiales que se usarán en el curso:

>\_ Terminal

Clonación inicial

```
git clone https://github.com/isantosruiz/machine
```

La carpeta de archivos se actualizará conforme avance el curso. Puede clonar nuevamente el repositorio en cualquier momento:

>\_ Terminal

Clonación subsecuente

```
git clone https://github.com/isantosruiz/machine temporal
```

La carpeta “**temporal**” se requiere para que Git no reescriba la carpeta “**machine**” existente. Puede borrarla después de copiar los archivos actualizados.



## Cursos online a su ritmo:

- <https://matlabacademy.mathworks.com/es/details/matlab-onramp/gettingstarted>
- <https://matlabacademy.mathworks.com/es/details/statistics-onramp/orst>
- <https://matlabacademy.mathworks.com/es/details/machine-learning-onramp/machinelearning>
- <https://matlabacademy.mathworks.com/es/details/deep-learning-onramp/deeplearning>
- <https://matlabacademy.mathworks.com/es/details/curve-fitting-onramp/orcf>
- <https://matlabacademy.mathworks.com/es/details/optimization-onramp/optim>
- <https://matlabacademy.mathworks.com/es/details/object-oriented-programming-onramp/oroop>

# *Introducción*

# Qué es una **máquina**



Association for  
Computing Machinery

# Qué es una máquina



Association for  
Computing Machinery



## Qué es una **máquina**

- Para la ACM, el término “máquina” generalmente se refiere a un sistema de computación o un dispositivo que realiza operaciones lógicas, matemáticas o algorítmicas. Esto puede incluir computadoras, servidores, y cualquier otro dispositivo capaz de ejecutar instrucciones de un programa o algoritmo.

## Qué es una **máquina**

- Para la ACM, el término “máquina” generalmente se refiere a un sistema de computación o un dispositivo que realiza operaciones lógicas, matemáticas o algorítmicas. Esto puede incluir computadoras, servidores, y cualquier otro dispositivo capaz de ejecutar instrucciones de un programa o algoritmo.
- La ACM se enfoca en el avance de la informática y la tecnología, por lo que el término “máquina” está intrínsecamente ligado a dispositivos que procesan información y toman decisiones o realizan acciones basadas en ese procesamiento.

## Qué es una **máquina**

- Para la ACM, el término “máquina” generalmente se refiere a un sistema de computación o un dispositivo que realiza operaciones lógicas, matemáticas o algorítmicas. Esto puede incluir computadoras, servidores, y cualquier otro dispositivo capaz de ejecutar instrucciones de un programa o algoritmo.
- La ACM se enfoca en el avance de la informática y la tecnología, por lo que el término “máquina” está intrínsecamente ligado a dispositivos que procesan información y toman decisiones o realizan acciones basadas en ese procesamiento.
- Por ejemplo, una “máquina” en este sentido podría ser un robot, una computadora que ejecuta un algoritmo de inteligencia artificial, o incluso una red de servidores que procesan datos a gran escala.

# Qué es Machine Learning

- ☞ Machine Learning es una técnica de IA que enseña a las computadoras a hacer lo que es natural para los humanos y los animales: **aprender de la experiencia.**

# Qué es Machine Learning

- ☞ Machine Learning es una técnica de IA que enseña a las computadoras a hacer lo que es natural para los humanos y los animales: **aprender de la experiencia**.
- ☞ Los algoritmos de Machine Learning usan métodos computacionales para “aprender” información directamente a partir de datos, sin depender de una ecuación predeterminada como modelo.

# Qué es Machine Learning

- 👉 Machine Learning es una técnica de IA que enseña a las computadoras a hacer lo que es natural para los humanos y los animales: **aprender de la experiencia**.
- 👉 Los algoritmos de Machine Learning usan métodos computacionales para “aprender” información directamente a partir de datos, sin depender de una ecuación predeterminada como modelo.
- 👉 Los algoritmos mejoran su rendimiento de manera adaptativa a medida que aumenta el número de muestras disponibles para el aprendizaje.

# Qué es Machine Learning

- ☞ Machine Learning es una técnica de IA que enseña a las computadoras a hacer lo que es natural para los humanos y los animales: **aprender de la experiencia**.
- ☞ Los algoritmos de Machine Learning usan métodos computacionales para “aprender” información directamente a partir de datos, sin depender de una ecuación predeterminada como modelo.
- ☞ Los algoritmos mejoran su rendimiento de manera adaptativa a medida que aumenta el número de muestras disponibles para el aprendizaje.
- ☞ Deep Learning es una variedad especializada de Machine Learning.

**machine learning**

aprendizaje de máquinas

**aprendizaje automático**

aprendizaje computacional

# Tomando decisiones sin Machine Learning



```
# Clasificación basada en comparaciones para exportación o consumo local
def clasificar_manzana():
    peso = obtener_peso()
    tamano = obtener_tamano()
    color = obtener_color()

    # Criterios para exportación:
    # Peso superior a 180 gramos, tamaño superior a 9 cm y color rojo
    if peso > 180 and tamano > 9 and color.lower() == "rojo":
        categoria = "exportación"
    else:
        categoria = "consumo local"

    return categoria

# Mostrar la clasificación de una manzana
categoria = clasificar_manzana()
print(f"La manzana ha sido clasificada para {categoria}.")
```

# Tomando decisiones sin Machine Learning



```
# Clasificación basada en comparaciones para exportación o consumo local
def clasificar_manzana():
    peso = obtener_peso()
    tamaño = obtener_tamaño()
    color = obtener_color()

    # Criterios para exportación:
    # Peso superior a 180 gramos, tamaño superior a 9 cm y color rojo
    if peso > 180 and tamaño > 9 and color.lower() == "rojo":
        categoría = "exportación"
    else:
        categoría = "consumo local"

    return categoría

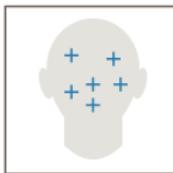
# Mostrar la clasificación de una manzana
categoría = clasificar_manzana()
print(f"La manzana ha sido clasificada para {categoría}.")
```

¿Cómo definir los límites para las comparaciones?

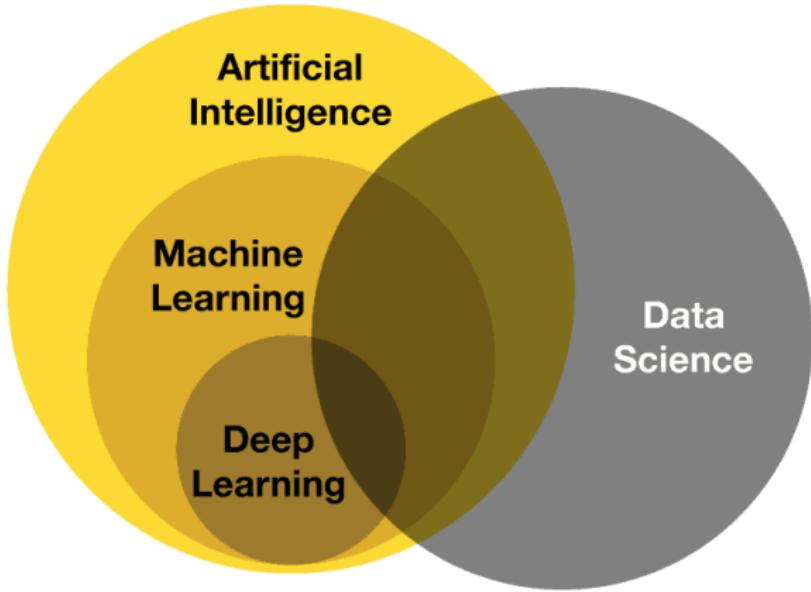
# Aplicaciones del mundo real

Con el aumento del big data, el aprendizaje automático se ha vuelto particularmente importante para resolver problemas en áreas como:

- ① Finanzas computacionales, para calificación crediticia y comercio algorítmico,
- ② Procesamiento de imágenes y visión por computadora, para reconocimiento facial, detección de movimiento y detección de objetos,
- ③ Biología computacional, para detección de tumores, descubrimiento de medicamentos y secuenciación de ADN,
- ④ Producción de energía, para previsión de precios y carga,
- ⑤ Automotriz, aeroespacial y manufactura, para mantenimiento predictivo,
- ⑥ Procesamiento de lenguaje natural.

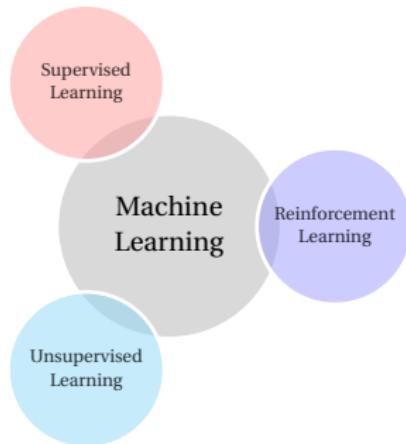


# Contexto del Machine Learning



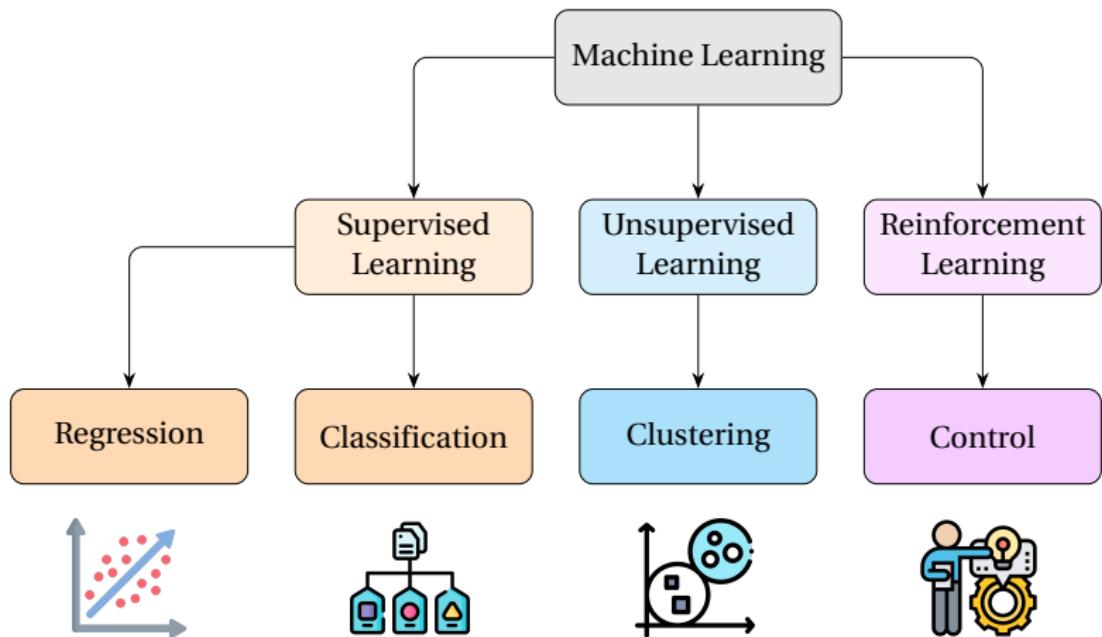
# Tipos de Machine Learning

- En **aprendizaje supervisado** se entrena un modelo con datos de entrada y salida conocidos para predecir salidas futuras.
- El **aprendizaje no supervisado** identifica patrones ocultos o estructuras intrínsecas en datos de entrada sin salidas conocidas.



- En el **aprendizaje por refuerzo** un agente aprende a tomar decisiones mediante la interacción con un entorno, recibiendo recompensas o castigos para mejorar su rendimiento.

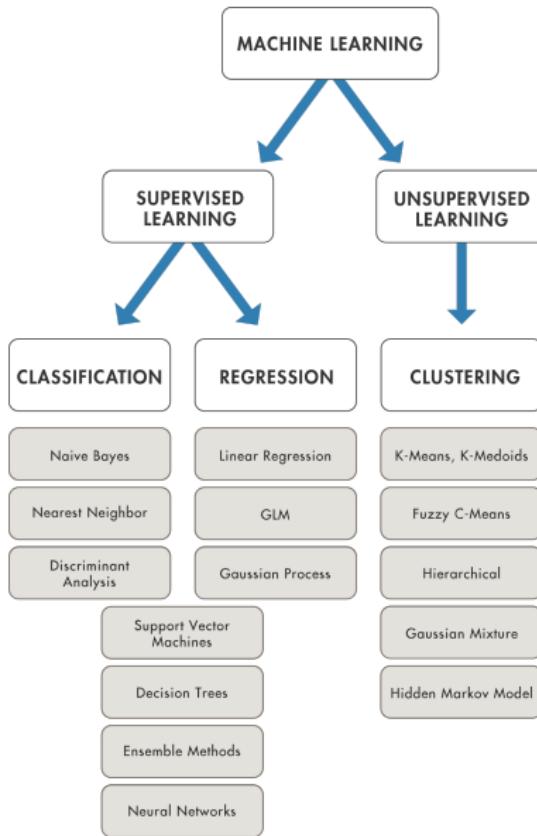
# Tipos de Machine Learning



# Cómo elegir el algoritmo adecuado

- Seleccionar el algoritmo adecuado puede resultar abrumador: existen infinidad de algoritmos de aprendizaje supervisado y no supervisado, y cada uno adopta un enfoque de aprendizaje distinto.
- No existe un método universal que se aplique a todas las situaciones. Encontrar el algoritmo adecuado es una cuestión de prueba y error; **ni los analistas de datos más experimentados pueden saber si un determinado algoritmo funcionará sin probarlo antes.**
- Seleccionar el algoritmo también depende del tamaño y tipo de datos con que se trabaje, la información que se desee obtener de los datos y cómo se vaya a emplear la información.

# Técnicas de Machine Learning

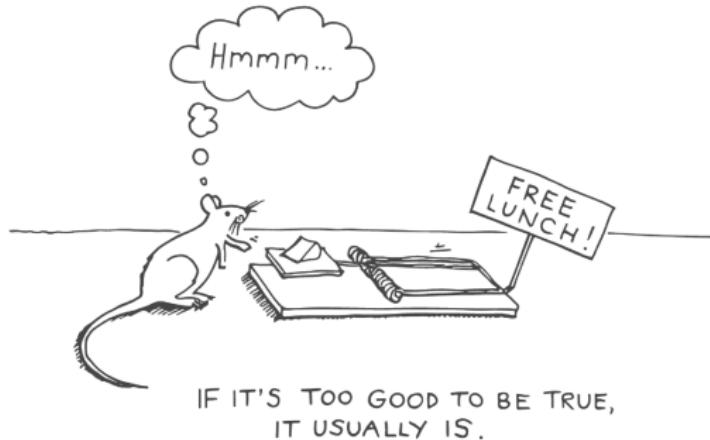


# Supervisado *versus* no supervisado

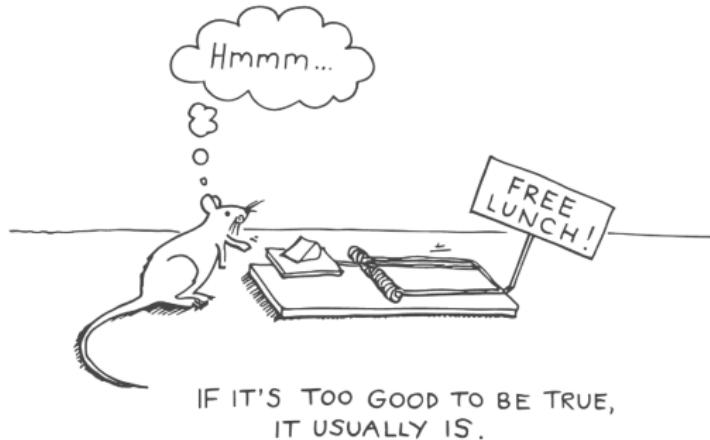
Pautas para elegir entre aprendizaje supervisado y no supervisado:

- Seleccione el aprendizaje supervisado si necesita entrenar un modelo para realizar una predicción. Por ejemplo, el valor futuro de una variable continua, tal como los grados de temperatura o la cotización de acciones, o bien una clasificación, tal como identificar marcas de automóviles en un vídeo de webcam.
- Seleccione el aprendizaje no supervisado si necesita explorar datos y desea entrenar un modelo para obtener una representación interna; por ejemplo, dividir datos en clusters.

# No Free Lunch Theorem

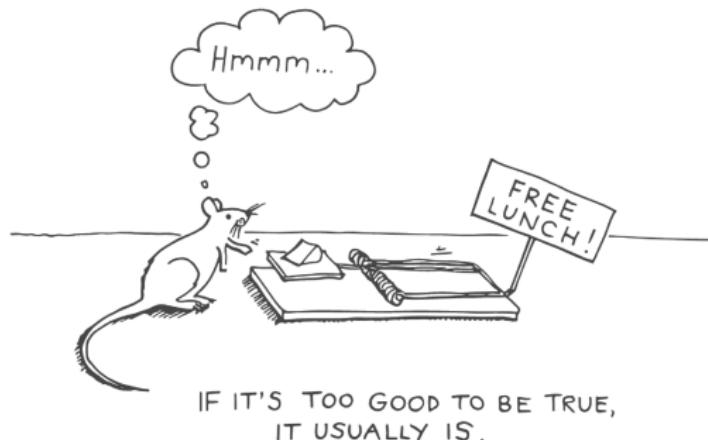


# No Free Lunch Theorem



El “No Free Lunch Theorem” en Machine Learning establece que **no existe un algoritmo de aprendizaje universalmente superior para todas las posibles tareas o problemas.**

# No Free Lunch Theorem



El “No Free Lunch Theorem” en Machine Learning establece que **no existe un algoritmo de aprendizaje universalmente superior para todas las posibles tareas o problemas.**

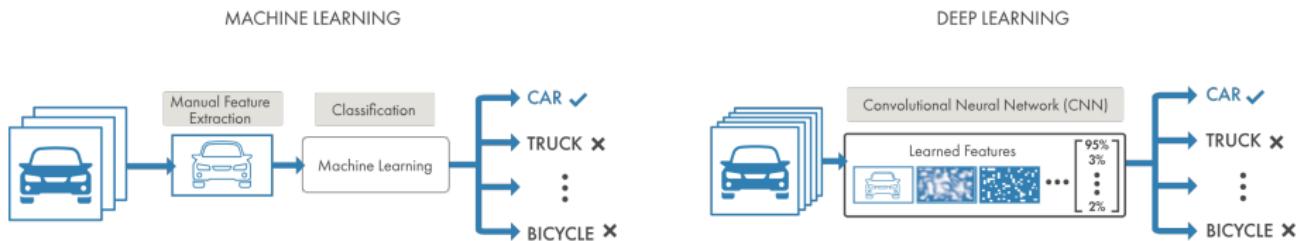
Este teorema destaca la importancia de adaptar y personalizar las soluciones de machine learning a cada caso particular.

# Diferencia entre Machine Learning y Deep Learning

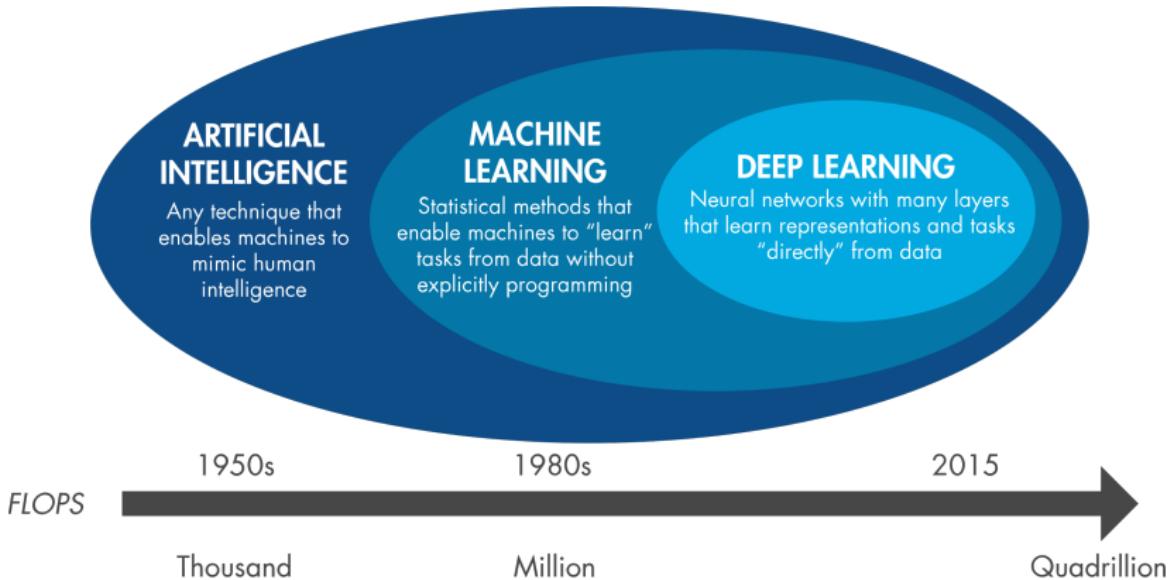
Deep Learning es una variedad especializada de Machine Learning.

En el análisis de imágenes, el flujo de trabajo de **Machine Learning** comienza con la extracción manual de características relevantes de imágenes. Estas características se utilizan luego para crear un modelo que categoriza los objetos presentes en la imagen.

El flujo de trabajo de **Deep Learning** extrae las características relevantes directamente de las imágenes. Además, Deep Learning realiza un “aprendizaje de extremo a extremo”: se proporciona una tarea y datos sin procesar a una red, que aprende a realizar la tarea automáticamente.



Con Machine Learning, la selección de características y el clasificador para categorizarlas se realiza manualmente. Con Deep Learning, los pasos de extracción de características y modelado son automáticos.



Con el tiempo, el rendimiento de las computadoras ha evolucionado a través de la inteligencia artificial, el aprendizaje automático y el aprendizaje profundo.

# Percepción



## Dishcraft Robotics Takes Over Dishwashing From Humans

Washing dishes is a problem that robots can solve, at least in commercial kitchens

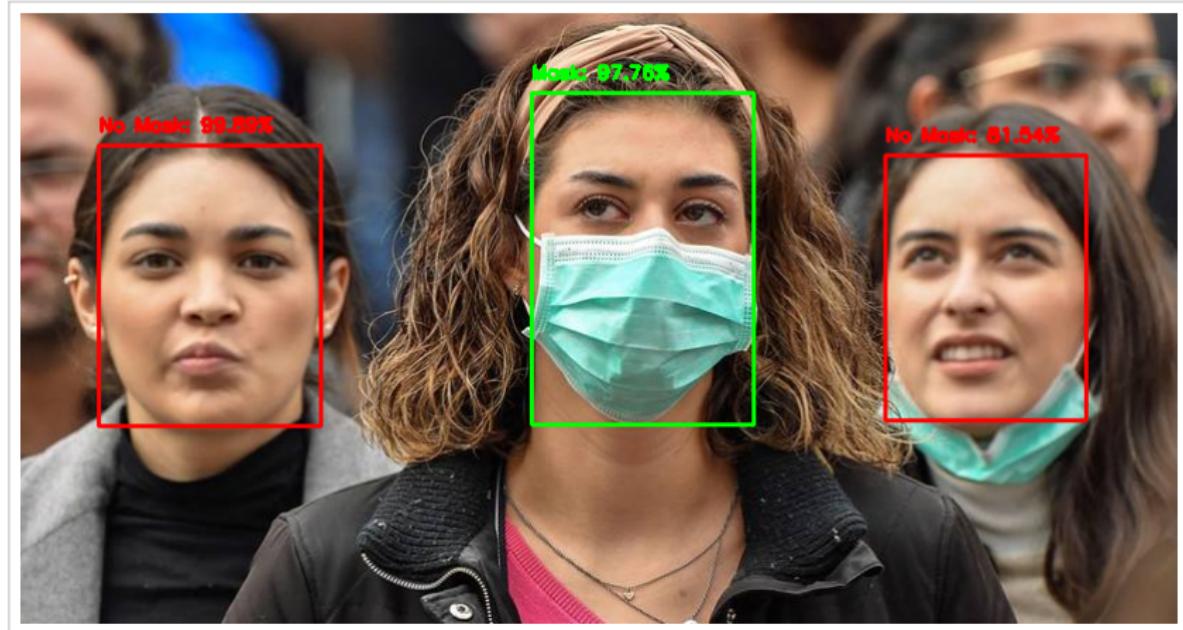
---

By Evan Ackerman



Image: Dishcraft Robotics

# Reconocimiento de patrones



# Diagnóstico

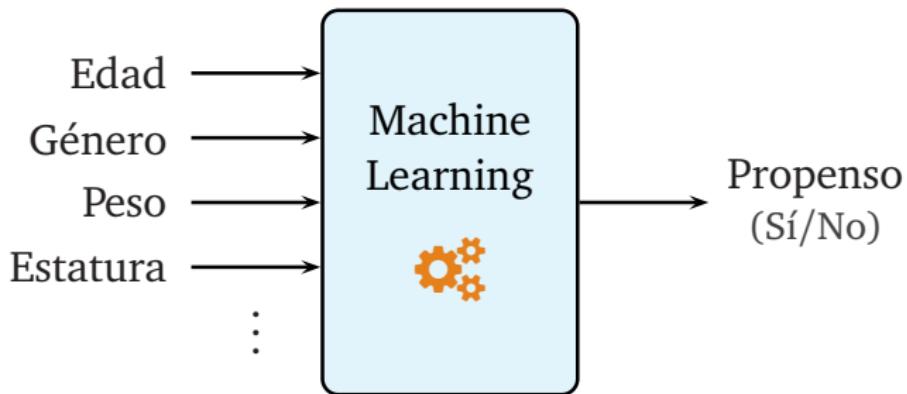


¿Es propenso a ser diabético?

# Diagnóstico



¿Es propenso a ser diabético?



# Toma de decisiones

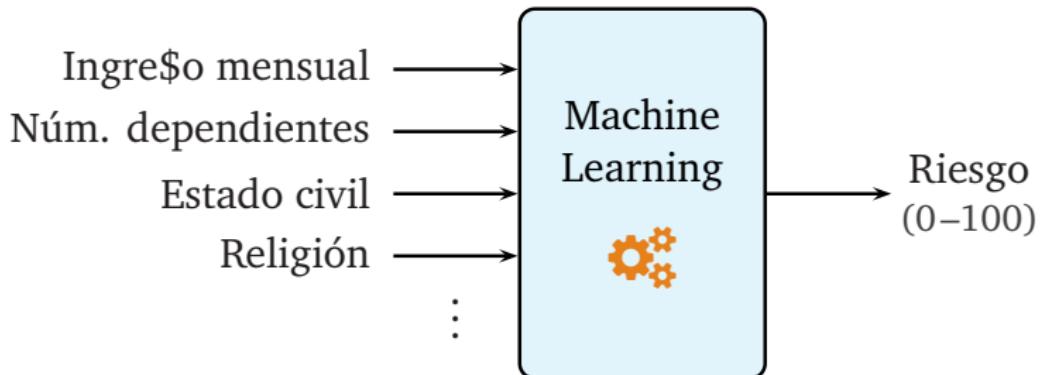


¿Debe aprobarse una solicitud de crédito?

# Toma de decisiones



¿Debe aprobarse una solicitud de crédito?



# Conducción autónoma



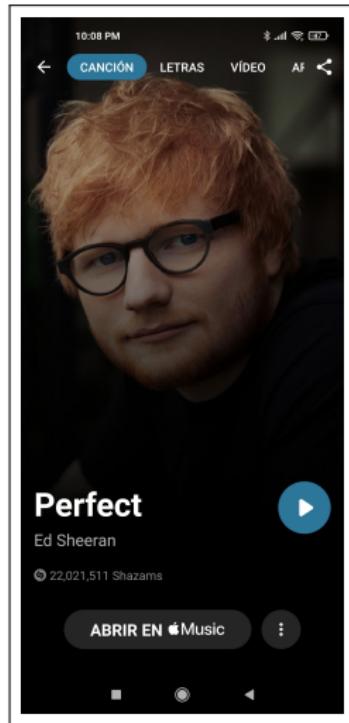
# Más datos, más preguntas, mejores respuestas

Los algoritmos de Machine Learning identifican patrones naturales en los datos, que generan conocimiento, ayudan a tomar mejores decisiones y realizar predicciones más acertadas. Se utilizan a diario para tomar decisiones cruciales en diagnósticos médicos, inversiones, predicción de carga energética y otros ámbitos. Por ejemplo, las plataformas de streaming dependen de Machine Learning para filtrar millones de opciones y ofrecer recomendaciones sobre canciones o películas. Los comerciantes lo utilizan para obtener información sobre el comportamiento de compra de sus clientes.

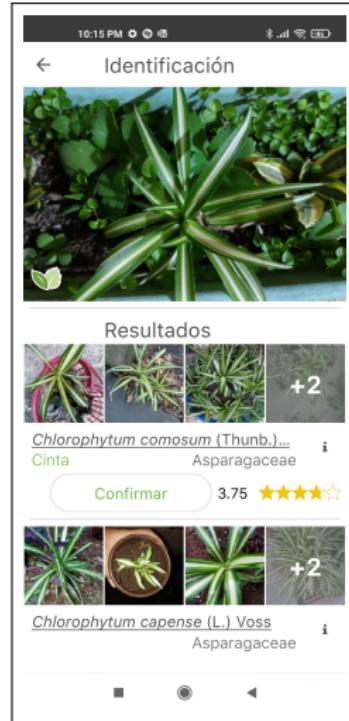
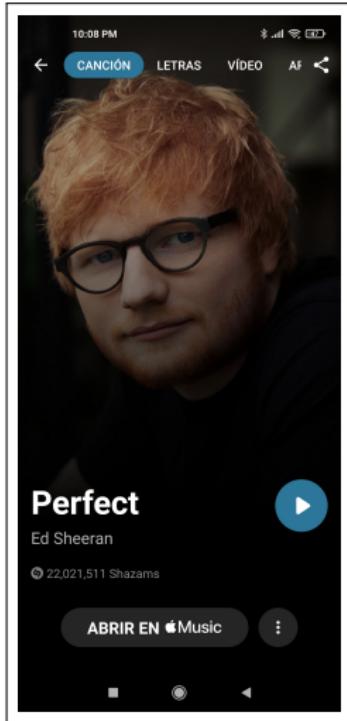
## Cuándo utilizar Machine Learning

Utilice Machine Learning cuando se enfrente a tareas o problemas complejos que impliquen una gran cantidad de datos y variables, pero no disponga de fórmulas o ecuaciones para solucionarlos.

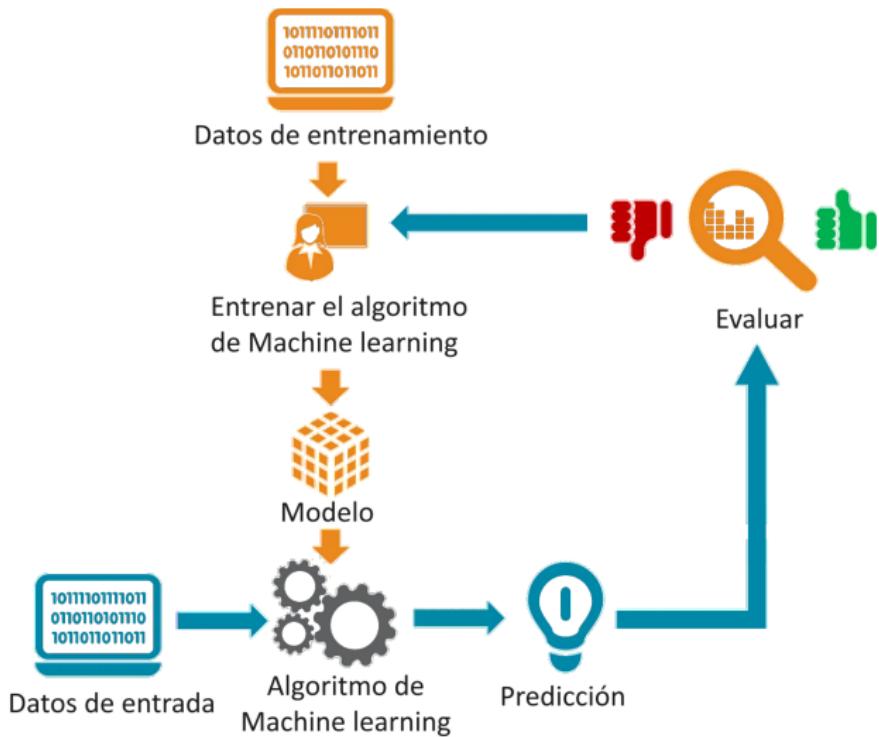
# Apps de Machine Learning



# Apps de Machine Learning



# Cómo funciona el aprendizaje automático



# Lo que puede fallar, fallará

NUEVAS  
TECNOLOGÍAS

## La Inteligencia artificial arruina un partido de fútbol al confundir la calva del árbitro con la pelota



- El partido se retransmitió con cámaras IA sin operadores humanos



# MATLAB | Python

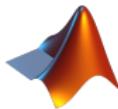


```
f = @(x) 5*x - x.^2;  
  
x0 = 1;  
disp("f(" + x0 + ") = " + f(x0))  
  
x = linspace(0, 5, 51)';  
y = f(x);  
  
plot(x, y, '*')  
  
t = table(x, y);  
disp(t)
```



```
f = lambda x : 5*x - x**2  
  
x0 = 1  
print(f"f({x0}) = {f(x0)}")  
  
import numpy as np  
x = np.linspace(0, 5, 52).transpose()  
y = f(x)  
  
from matplotlib import pyplot as plt  
plt.plot(x, y, '*')  
plt.show()  
  
import pandas as pd  
df = pd.DataFrame({'x': x, 'y': y})  
print(df)
```

# Toolboxes de MATLAB



- ① Statistics and Machine Learning Toolbox
- ② Deep Learning Toolbox
- ③ Reinforcement Learning Toolbox
- ④ Curve Fitting Toolbox
- ⑤ Text Analytics Toolbox

# Paquetes de Python



- ① NumPy (`numpy`) : Matrices y álgebra lineal
- ② Matplotlib (`matplotlib.pyplot`) : Gráficos
- ③ Pandas (`pandas`) : Manipulación y análisis de datos
- ④ scikit-learn (`sklearn`) : Aprendizaje automático
- ⑤ TensorFlow (`tensorflow`) : Aprendizaje automático/profundo
- ⑥ Keras (`tensorflow.keras`) : Aprendizaje profundo
- ⑦ PyTorch (`torch`) : Aprendizaje profundo

# *Los datos*

# Structured Query Language (SQL)

## DuckDB

```
1 CREATE TABLE patients AS
2 SELECT *
3 FROM read_csv_auto('patients.csv');
4 DESCRIBE patients;
5 SELECT *
6 FROM patients
7 WHERE Age < 40
8 ORDER BY Age ASC;
9 SELECT
10    MIN(Weight) AS min_weight,
11    MAX(Weight) AS max_weight,
12    MIN(Height) AS min_height,
13    MAX(Height) AS max_height
14 FROM patients
15 WHERE Gender = 'Male';
```

```
16 SELECT
17    AVG(Weight)/AVG(Height) AS ratioF
18 FROM patients
19 WHERE Gender = 'Female';
20 SELECT
21    AVG(Weight)/AVG(Height) AS ratioM
22 FROM patients
23 WHERE Gender = 'Male';
24 COPY (
25 SELECT Weight, Height, Gender
26 FROM patients
27 WHERE Age < 40
28 ORDER BY Age ASC
29 ) TO 'young_patients.csv' (FORMAT
30 CSV, HEADER, DELIMITER ',');
```

<https://duckdb.org/docs/installation>

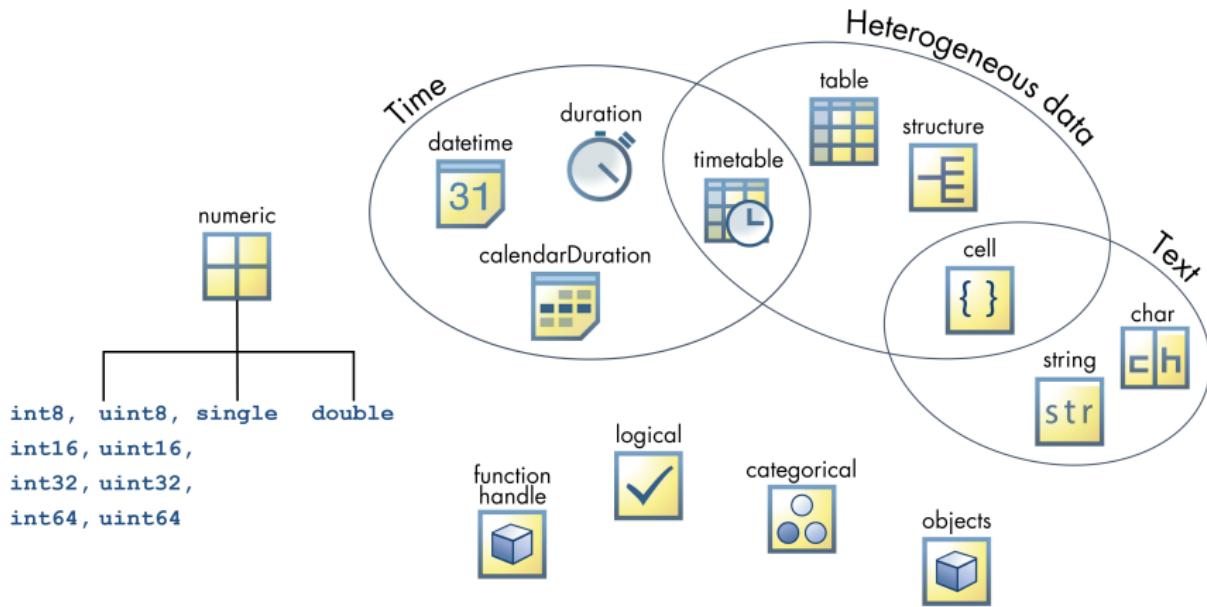
# Tipos de datos

- ① Lógicos: `true`, `false`
- ② Números enteros: `0`, `5`, `-1`
- ③ Números reales: `3.1416`, `-2.34e-5`
- ④ Caracteres: `'a'`, `'Z'`,  `'+'`
- ⑤ Cadenas de texto: `"Hello, World!"`
- ⑥ Categóricos: `{cash, card, crypto}`

# Datos estructurados

- ① Arreglos
- ② Listas
- ③ Estructuras
- ④ Diccionarios
- ⑤ Tablas
- ⑥ Objetos
- ⑦ Audios
- ⑧ Imágenes

# Tipos de datos MATLAB



# Arreglos y listas

MATLAB

```
names = ["Mary", "Pedrito", "John", "Cristina", "Richie"];
names(2)
names(2:end-1)
names'
```

# Arreglos y listas

MATLAB

```
names = ["Mary", "Pedrito", "John", "Cristina", "Richie"];
names(2)
names(2:end-1)
names'

names = {"Mary", "Pedrito", "John", "Cristina", "Richie"};
names'
string(names')
string(names)'
```

# Arreglos y listas



```
names = ["Mary", "Pedrito", "John", "Cristina", "Richie"];
names(2)
names(2:end-1)
names'

names = {"Mary", "Pedrito", "John", "Cristina", "Richie"};
names'
string(names')
string(names)'

things = {"Pedrito", 5, '@', [1, 2, 3]}
things(3)
things(end)
things{3}
things{end}
```

# Arreglos y listas



```
names = ["Mary", "Pedrito", "John", "Cristina", "Richie"];
names(2)
names(2:end-1)
names'

names = {"Mary", "Pedrito", "John", "Cristina", "Richie"};
names'
string(names')
string(names)'

things = {"Pedrito", 5, '@', [1, 2, 3]}
things(3)
things(end)
things{3}
things{end}

words = ['Juan', 'Pérez'] % Se unen
words = ['Juan'; 'Pérez'] % iError!
```

# Arreglos multidimensionales



```
A = [1, 2, 3; 4, 5, 6]          % matriz de 2x3
M = magic(5)                  % matriz "mágica"
heatmap(M)                    % mapa de calor
N = M'                         % transpuesta
r1 = M(1,:)                   % primer renglón
r3 = M(3,:)                   % tercer renglón
c2 = M(:,2)                   % segunda columna
cf = M(:,end)                 % última columna
S = M(2:4,[1,5])              % submatriz
d = diag(M)                   % diagonal principal de la matriz
C = zeros(8, 4)                % 8 renglones y 4 columnas de ceros
U = ones(8, 4)                 % 8 renglones y 4 columnas de unos
T = [magic(3); [1, 2, 3]]      % supermatriz
x = 1:5                        % [1, 2, 3, 4, 5]
y = 1:0.2:2                    % [1, 1.2, 1.4, 1.6, 1.8, 2]
l = linspace(0, 10, 11)         % [0, 1, 2, ..., 9, 10]
```

# Estructuras

MATLAB

```
s.name = "John";
s.lastName = "Smith";
s.age = 35;
s.course(1) = "Math";
s.course(2) = "Physics";
s.course(3) = "Arts";
% s.course = ["Math", "Physics", "Arts"];
```

# Estructuras

MATLAB

```
s.name = "John";
s.lastName = "Smith";
s.age = 35;
s.course(1) = "Math";
s.course(2) = "Physics";
s.course(3) = "Arts";
% s.course = ["Math", "Physics", "Arts"];

disp(s)
disp(s.course)
txt = jsonencode(s,"PrettyPrint",true);
disp(txt)
```

# Estructuras

MATLAB

```
s.name = "John";
s.lastName = "Smith";
s.age = 35;
s.course(1) = "Math";
s.course(2) = "Physics";
s.course(3) = "Arts";
% s.course = ["Math", "Physics", "Arts"];

disp(s)
disp(s.course)
txt = jsonencode(s,"PrettyPrint",true);
disp(txt)

s(2).name = "Juanita";
disp(s)
disp(s(1))
disp(s(2))
```

# Estructuras



```
s.name = "John";
s.lastName = "Smith";
s.age = 35;
s.course(1) = "Math";
s.course(2) = "Physics";
s.course(3) = "Arts";
% s.course = ["Math", "Physics", "Arts"];

disp(s)
disp(s.course)
txt = jsonencode(s,"PrettyPrint",true);
disp(txt)

s(2).name = "Juanita";
disp(s)
disp(s(1))
disp(s(2))

txt = '{"aula": "Q3", "temperatura": 27, "humedad": 85}'
s = jsondecode(txt)
```

# Diccionarios

Un diccionario en programación es una estructura de datos que almacena pares de **clave-valor**. Es similar a un diccionario en la vida real, donde cada palabra (clave) está asociada con una definición (valor).

**key → value**



```
name = ["John Smith", "Juan Pérez", "Giuseppe Rossi"];
grades = {[80,75,90], [90,100,95], [50,80,65]};
student = dictionary(name, grades)

gradesJohn = student("John Smith")
cell2mat(gradesJohn)
gradesJohn{1}
gradesJohn = student{"John Smith"}
```



# Lectura de archivos de imagen



```
I = imread('coins.png');           % imagen en escala de grises
size(I)
imshow(I)
colormap jet

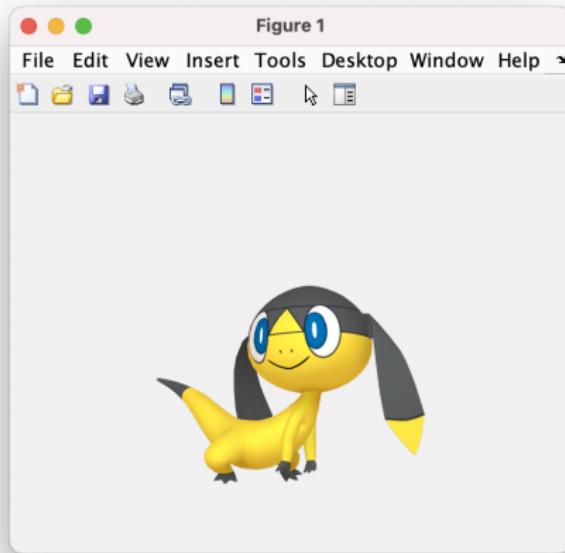
[X, map] = imread('corn.tif'); % imagen indexada
size(X)
imshow(X, map)
heatmap(X(1:10,1:10))
colormap gray; colorbar off

X = imread('ngc6543a.jpg');      % imagen RGB
size(X)
imshow(X)
R = X(:, :, 1); G = X(:, :, 2); B = X(:, :, 3);
X = imresize(X, [227, 227]);
imwrite(X, 'modified.png')
```



# Imágenes RGBA

```
MATLAB  
[img,~,alpha] = imread('694.png');  
h = imshow(img); h.AlphaData = alpha;
```



# Lectura de archivos de audio

```
MATLAB

info = audioinfo('apollo13.wav')
[y,Fs] = audioread('apollo13.wav');
sound(y, Fs)

out = py.gtts.gTTS('¡Hola, Mundo!', lang='es');
out.save('saludo.mp3');
[y, Fs] = audioread('saludo.mp3');
saludo = audioplayer(y, Fs);
saludo.play()

Fs = 32000;
audiowrite('saludo32k.mp3', y, Fs)
system('start saludo32k.mp3')
plot(y)
```



# Lectura de textos, listas, tablas y estructuras



```
fid = fopen('basura.txt','r');
while ~feof(fid)
    txt = fgetl(fid);
    disp(txt)
end
fclose(fid);

c = fileread('basura.txt')          % txt
c = readcell('basura.txt')          % txt
t = readtable('patients.xls')        % xls, csv
s = readstruct('books.xml')          % xml, json
```

# Operando con datos tabulares

MATLAB

```
t = readtable('patients.xls');
whos t % [nRows, nCols] = size(t)
head(t)
tail(t)
summary(t)
```



# Operando con datos tabulares

MATLAB

```
t = readtable('patients.xls');
whos t % [nRows, nCols] = size(t)
head(t)
tail(t)
summary(t)
t.Properties.VariableNames
t.Properties.RowNames = compose('Patient %d',1:100);
head(t)
```



CSV



XLS

# Operando con datos tabulares



```
t = readtable('patients.xls');
whos t    % [nRows, nCols] = size(t)
head(t)
tail(t)
summary(t)
t.Properties.VariableNames
t.Properties.RowNames = compose('Patient %d',1:100);
head(t)
t(3,2)
t('Patient 3',{'Gender','Smoker'}) % t('Patient 3',["Gender","Smoker"])
t(randperm(100,5),:)
datasample(t,5)
```



# Operando con datos tabulares



```
t = readtable('patients.xls');
whos t    % [nRows, nCols] = size(t)
head(t)
tail(t)
summary(t)

t.Properties.VariableNames
t.Properties.RowNames = compose('Patient %d',1:100);
head(t)

t(3,2)
t('Patient 3',{'Gender','Smoker'}) % t('Patient 3',[ "Gender", "Smoker"])
t(randperm(100,5),:)
datasample(t,5)

tbl = sortrows(t(:,["Age", "Height", "Weight", "Gender"]), ...
    ["Height", "Weight", "Age"], "ascend")
writetable(tbl, 'filteredPatients.csv')
```



# Dataset fisheriris

Iris virginica



Iris versicolor



Iris setosa



# Dataset fisheriris

Iris virginica



Iris versicolor



Iris setosa



Sepal Length	Sepal Width	Petal Length	Petal Width	Species
6.8	3	5.5	2.1	virginica
4.6	3.6	1	0.2	setosa
6.3	2.3	4.4	1.3	versicolor
6.3	3.3	6	2.5	virginica
5.4	3	4.5	1.5	versicolor
5.1	3.5	1.4	0.3	setosa
:	:	:	:	:
:	:	:	:	:

# Dataset fisheriris

Iris virginica



Iris versicolor



Iris setosa



Sepal Length	Sepal Width	Petal Length	Petal Width	Species
6.8	3	5.5	2.1	virginica
4.6	3.6	1	0.2	setosa
6.3	2.3	4.4	1.3	versicolor
6.3	3.3	6	2.5	virginica
5.4	3	4.5	1.5	versicolor
5.1	3.5	1.4	0.3	setosa
:	:	:	:	:
:	:	:	:	:



```
data = readtable('fisheriris.csv') % lee como tabla  
data = load('fisheriris.mat')      % lee como estructura
```

## Dataset **fisheriris**



```
load fisheriris           % binario "fisheriris.mat"  
whos                      % buscar en el workspace
```

# Dataset fisheriris



```
load fisheriris % binario "fisheriris.mat"  
whos % buscar en el workspace  
  
iris = load('fisheriris.mat') % forma recomendada  
X = iris.meas; % matriz 150x4  
y = iris.species; % vector 150x1  
gscatter(X(:,3), X(:,4))  
gscatter(X(:,3), X(:,4), y)
```

# Dataset fisheriris



```
load fisheriris % binario "fisheriris.mat"  
whos % buscar en el workspace  
  
iris = load('fisheriris.mat') % forma recomendada  
X = iris.meas; % matriz 150x4  
y = iris.species; % vector 150x1  
gscatter(X(:,3), X(:,4))  
gscatter(X(:,3), X(:,4), y)  
  
unique(y) % tipos de iris en el dataset  
y_cat = categorical(y); % char >>> categorical
```

# Dataset fisheriris



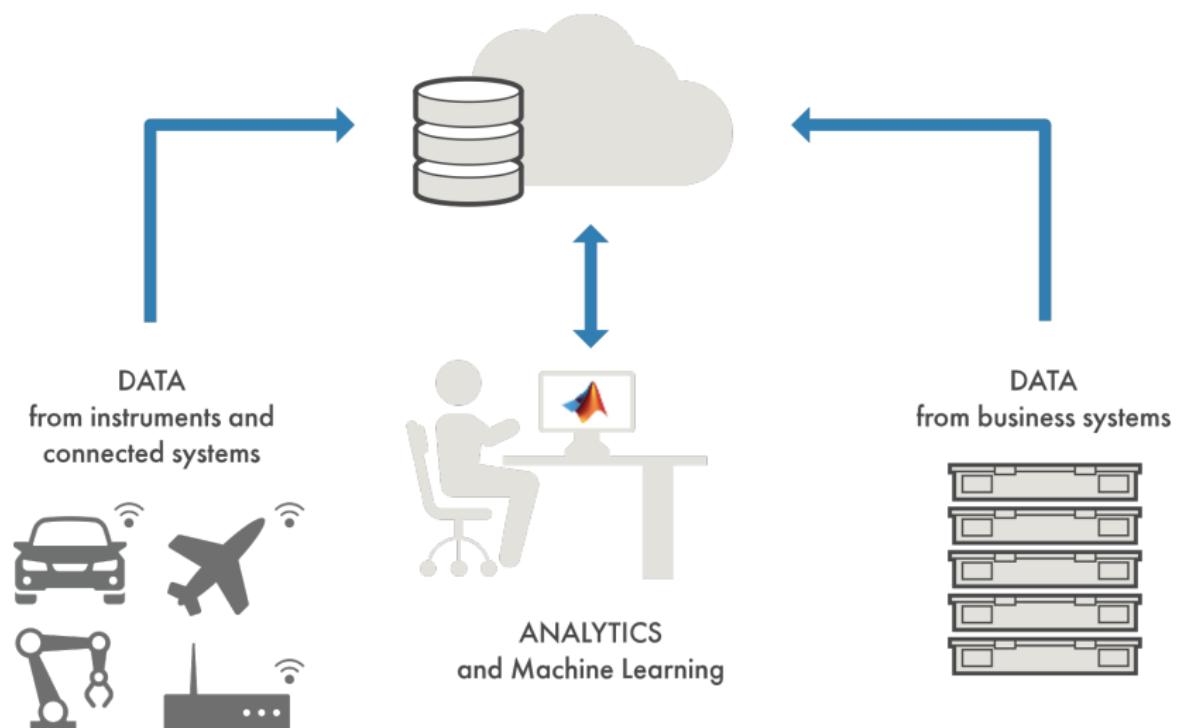
```
load fisheriris % binario "fisheriris.mat"  
whos % buscar en el workspace  
  
iris = load('fisheriris.mat') % forma recomendada  
X = iris.meas; % matriz 150x4  
y = iris.species; % vector 150x1  
gscatter(X(:,3), X(:,4))  
gscatter(X(:,3), X(:,4), y)  
  
unique(y) % tipos de iris en el dataset  
y_cat = categorical(y); % char >>> categorical  
  
y_setosa = double(strcmp(y, 'setosa'));  
y_setosa = double(string(y) == "setosa"); % captura la "setosidad"  
unique(y_setosa)  
gscatter(X(:,3), X(:,4), y_setosa)
```

# Dataset fisheriris



```
load fisheriris % binario "fisheriris.mat"
whos % buscar en el workspace
iris = load('fisheriris.mat') % forma recomendada
X = iris.meas; % matriz 150x4
y = iris.species; % vector 150x1
gscatter(X(:,3), X(:,4))
gscatter(X(:,3), X(:,4), y)
unique(y) % tipos de iris en el dataset
y_cat = categorical(y); % char >> categorical
y_setosa = double(strcmp(y, 'setosa'));
y_setosa = double(string(y) == "setosa");
unique(y_setosa) % captura la "setosidad"
gscatter(X(:,3), X(:,4), y_setosa)
t = table(X(:,3), X(:,4), y_setosa); % dos predictores, dos clases
t.Properties.VariableNames = [ ...
    "PetalLength", "PetalWidth", "Setosity"];
writetable(t, "fisheriris2.csv")
writetable(t, "fisheriris2.xls")
```

# Datos para Analítica y Machine Learning



## Web Data

### RESTful Web Service

<b>webread</b>	Read data
<b>webwrite</b>	Write data
<b>websave</b>	Save data to file
<b>weboptions</b>	Specify options such as authentication and timeout

### JSON

<b>jsondecode</b>	<b>jsonencode</b>
-------------------	-------------------

### HTTP Messaging

Use the HTTP interface for more complex

web communication: **>>**

```
body = ...
matlab.net.http.MessageBody(x);

request = ...
matlab.net.http.RequestMessage ...
(method,header,body);
```

# JavaScript Object Notation (JSON)

- **Números:** Se permiten números negativos y opcionalmente pueden contener parte fraccional separada por puntos. Ejemplo: `123.456`
- **Cadenas:** Representan secuencias de cero o más caracteres. Se ponen entre doble comilla y se permiten cadenas de escape. Ejemplo: `"Hola"`
- **Booleanos:** Representan valores booleanos y pueden tener dos valores: `true` y `false`
- **Nulos:** Se representan con el valor `null`
- **Array:** Representa una lista ordenada de cero o más valores los cuales pueden ser de cualquier tipo. Los valores se separan por comas y el vector se mete entre corchetes. Ejemplo `["juan", "pedro", "jacinto"]`
- **Objetos:** Son colecciones no ordenadas de pares de la forma `<nombre>:<valor>` separados por comas y puestas entre llaves. El nombre tiene que ser una cadena entre comillas dobles. El valor puede ser de cualquier tipo.

# Ejemplo JSON

```
{  
    "departamento": 8,  
    "nombre_departamento": "Ventas",  
    "director": {  
        "nombre": "María",  
        "apellido": "Hernández"  
    },  
    "empleados": [  
        {  
            "nombre": "Juan",  
            "apellido": "Pérez"  
        },  
        {  
            "nombre": "John",  
            "apellido": "Smith"  
        }  
    ]  
}
```

# Extensible Markup Language (XML)

XML es un lenguaje de marcado flexible y basado en texto, diseñado para almacenar y transportar datos de manera estructurada y legible tanto para humanos como para máquinas.

- Es ampliamente utilizado en la interoperabilidad de sistemas en la web, aplicaciones móviles y bases de datos.
- Es un **metalenguaje** que permite definir lenguajes de marcas específicos.

```
<person>
    <name>Juan Pérez</name>
    <age>69</age>
    <email>juan.perez@example.com</email>
</person>
```



## email.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<email>
    <header>
        <from>juan.perez@example.com</from>
        <to>maria.gonzalez@example.com</to>
        <subject>Reunión de Seguimiento</subject>
        <date>2024-07-16T10:30:00</date>
    </header>
    <body>
        <p>Estimada María,</p>
        <p>Estoy coordinando una reunión para discutir los avances del proyecto.
            ¿Podrías confirmar tu disponibilidad para el viernes 19 de julio a
            las 10:00 AM?</p>
        <p>Quedo atento a tu respuesta.</p>
        <p>Saludos cordiales,</p>
        <p>Juan Pérez</p>
    </body>
    <attachments>
        <file name="informe.pdf" />
    </attachments>
</email>
```

# Parsing de XML en MATLAB

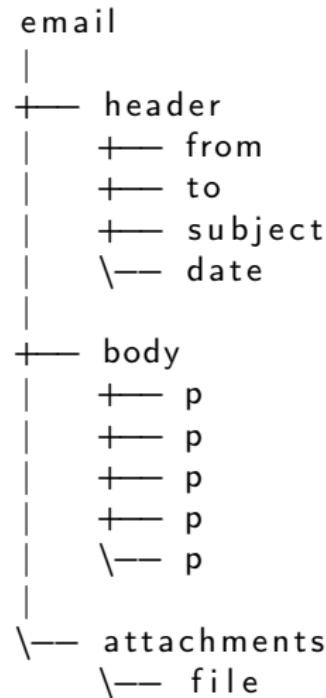
MATLAB

```
filename = 'email.xml';
text = fileread(filename)
email = readstruct(filename)
disp(email.header)
disp(email.body.p(3))
disp(email.body.p')
```

# Parsing de XML en MATLAB

MATLAB

```
filename = 'email.xml';
text = fileread(filename)
email = readstruct(filename)
disp(email.header)
disp(email.body.p(3))
disp(email.body.p')
```



# HTTP requests

**Propósito:** Crear mensajes de solicitud HTTP y enviarlos a un servidor para su procesamiento. Se recibe una respuesta.



```
request = matlab.net.http.RequestMessage('GET');
url = 'https://isantosruiz.github.io/home/test.html';
uri = matlab.net.URI(url);
options = matlab.net.http.HTTPOptions('ConnectTimeout', 20);
response = request.send(uri, options)
char(response)
response.Body.Data
```

Códigos de respuesta:

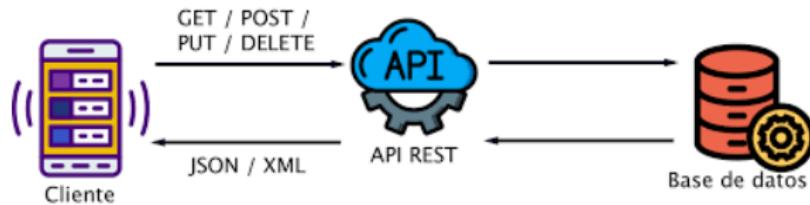
👍 200 OK

👎 404 Not Found

# API REST

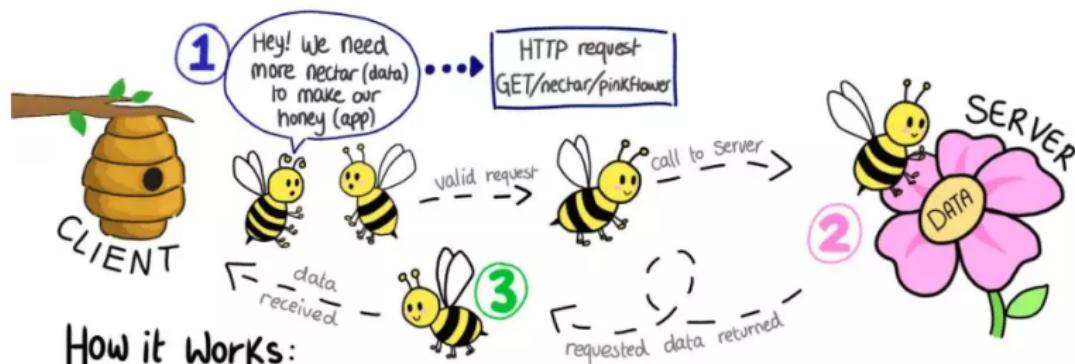
Una interfaz es **una capa que conecta dos sistemas**. Una API es una interfaz que conecta aplicaciones **para que comparten información**, por ejemplo, la API de mapas de Google, entre otras.

REST es una arquitectura para APIs que se conectan **vía web con el protocolo HTTP**. Permite acceder a la información con identificadores únicos (URLs).



# What is an API ?

An application programming interface allows two programs to communicate. On the web, APIs sit between an application and a web server, and facilitate the transfer of data.



## How it Works:

### ① Request

API call is initiated by the Client application via a HTTP request

### ② Receive

Our worker bee acts as an API, going to a flower (server) to collect nectar (data)

### ③ Response

The API transfers the requested data back to the requesting application, usually in JSON format

# Simplificando las peticiones HTTP con `webread`

## DolarAPI

MATLAB

```
url = 'https://mx.dolarapi.com/v1/cotizaciones/usd';
data = webread(url);
disp(data)
c = data.compra;
v = data.venta;
```

# Simplificando las peticiones HTTP con `webread`

## DolarAPI



```
url = 'https://mx.dolarapi.com/v1/cotizaciones/usd';
data = webread(url);
disp(data)
c = data.compra;
v = data.venta;
```



```
import requests
url = "https://mx.dolarapi.com/v1/cotizaciones/usd"
response = requests.get(url)
data = response.json()
print(data)
c = data["compra"]
v = data["venta"]
```

# Transferencia de estado representacional (REST)

## The RESTful Pokémon API



```
url = 'https://pokeapi.co/api/v2/pokemon?offset=0&limit=50';
```

# Transferencia de estado representacional (REST)

## The RESTful Pokémon API



```
url = 'https://pokeapi.co/api/v2/pokemon?offset=0&limit=50';
data = webread(url)
names = string({data.results.name'});
urls = string({data.results.url'});
t = table(names,urls);
disp(t)
```

# Transferencia de estado representacional (REST)

## The RESTful Pokémon API



```
url = 'https://pokeapi.co/api/v2/pokemon?offset=0&limit=50';
data = webread(url)
names = string({data.results.name'});
urls = string({data.results.url'});
t = table(names,urls);
disp(t)
url = t.urls(t.names == "pikachu")
```

# Transferencia de estado representacional (REST)

## The RESTful Pokémon API

MATLAB

PokéAPI

```
url = 'https://pokeapi.co/api/v2/pokemon?offset=0&limit=50';
data = webread(url)
names = string({data.results.name'});
urls = string({data.results.url'});
t = table(names,urls);
disp(t)
url = t.urls(t.names == "pikachu")
data = webread(url)
w = data.weight;
h = data.height;
url_img = data.sprites.other.home.front_default
```

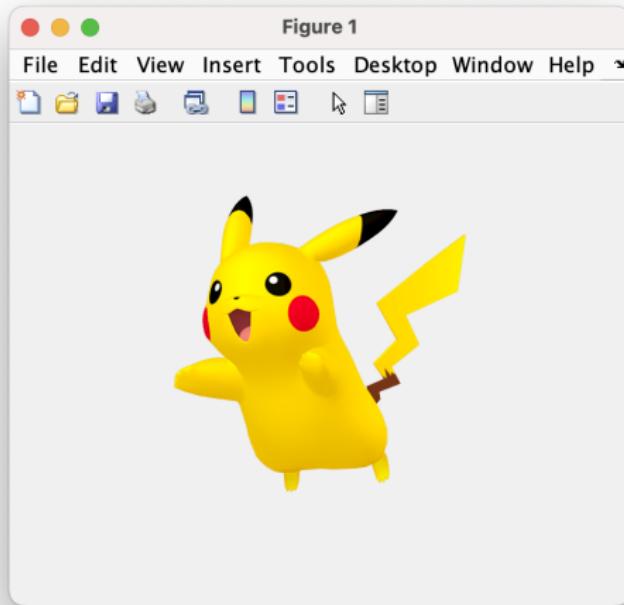
# Transferencia de estado representacional (REST)

## The RESTful Pokémon API



```
url = 'https://pokeapi.co/api/v2/pokemon?offset=0&limit=50';
data = webread(url)
names = string({data.results.name'});
urls = string({data.results.url'});
t = table(names,urls);
disp(t)
url = t.urls(t.names == "pikachu")
data = webread(url)
w = data.weight;
h = data.height;
url_img = data.sprites.other.home.front_default
disp("Peso: " + w/10 + " kg, Altura: " + h/10 + " m")
[img,~,alpha] = imread(url_img);
handle_img = imshow(img);
handle_img.AlphaData = im2double(alpha);
```

# The RESTful Pokémon API



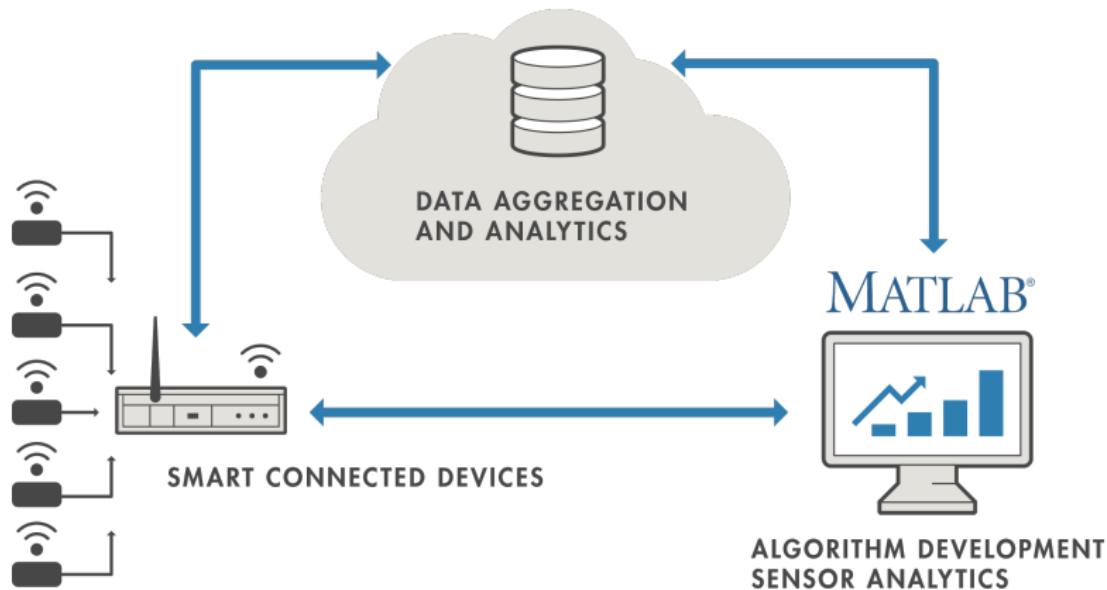
# Actividad 1: REST, creación y manipulación de tablas

Con datos de los pokémon obtenidos mediante la RESTful Pokémon API (disponible en la URL <https://pokeapi.co/api/v2/pokemon>), elaborar una tabla con la siguiente estructura:

Name	Height	Weight
<b>Favorite 1</b>	0.7	5.9
<b>Favorite 2</b>	0.7	6.5
<b>Favorite 3</b>	0.4	4.0
<b>Favorite 4</b>	0.4	4.0
<b>Favorite 5</b>	0.5	9.0
<b>Favorite 6</b>	0.5	9.0
<b>Favorite 7</b>	0.5	9.0
<b>Favorite 8</b>	0.8	10.0

La tabla debe ser ordenada en forma ascendente respecto a la altura y al peso de los pokémon (en ese orden). Los ocho pokémon favoritos a incluir en la tabla son “charmander”, “ho-oh”, “eevee”, “mew”, “pikachu”, “rattata”, “squirtle” y “tepig”. Guarde la tabla en los formatos CSV, Excel y binario de MATLAB.

# Comunicación para IoT mediante REST



# API de ThingSpeak



```
url = 'https://api.thingspeak.com/channels/12397/feeds.json?results=1'
data = webread(url)
data.channel
data.feeds
disp(string(data.channel.field3) + ": " + data.feeds.field3)
disp(string(data.channel.field4) + ": " + data.feeds.field4)
lat = str2num(data.channel.latitude);
lon = str2num(data.channel.longitude);
geoplot(lat, lon, 'r*')
geobasemap('street')
geolimits([lat-0.5, lat+0.5], [lon-0.5, lon+0.5])
```

MathWorks Weather Station, West Garage, Natick, MA 01760, USA

<https://www.hackster.io/mawrey/solar-powered-cellular-weather-station-e5a3ed>

# API de ThingSpeak



```
channel_id = "12397";
read_api_key = "0123456789ABCDE";
num_results = 24*60;

url = "https://api.thingspeak.com/channels/" + channel_id + "/feeds.json";
params = {"api_key", read_api_key, "results", num_results};
data = webread(url, params{:});

humidity = double(string({data.feeds.field3}'));
time = datetime({data.feeds.created_at}', Format="uuuu-MM-dd'T'HH:mm:ss.SS'Z'");

tblHumidity = table(time, humidity);
disp(tail(tblHumidity))
plot(time, humidity, LineWidth=1)
ylim([25,100])
```

# API de OpenWeather



```
q = "Tuxtla Gutiérrez, MX";
appid = fileread("appid.key"); % appid = "Your appID"
base_url = "https://api.openweathermap.org/data/2.5/weather";
data = webread(base_url + "?q=" + q + "&appid=" + appid);
tmp = round(data.main.temp - 273.15);
hmd = data.main.humidity;
desc = string(data.weather.description);
icon_filename = string(data.weather.icon) + "@4x.png";
url = "http://openweathermap.org/img/wn/" + icon_filename;
websave(icon_filename, url);
[img, ~, alpha] = imread(icon_filename);
h = imshow(img); h.AlphaData = im2double(alpha);
info = desc + ", " + hmd + "%, " + tmp + " C";
title(h.Parent, info)
set(gcf, 'Color', 'w', 'MenuBar', 'none')
```

## Actividad 2: Análisis de datos

Descargar la base de datos más reciente de casos de **influenza, COVID-19 y otros virus respiratorios** del portal de Datos Abiertos de la Dirección General de Epidemiología:



[http://datosabiertos.salud.gob.mx/gobmx/salud/datos\\_abiertos/datos\\_abiertos\\_influenza\\_covid19.zip](http://datosabiertos.salud.gob.mx/gobmx/salud/datos_abiertos/datos_abiertos_influenza_covid19.zip)



[http://datosabiertos.salud.gob.mx/gobmx/salud/datos\\_abiertos/diccionario\\_datos\\_abiertos.zip](http://datosabiertos.salud.gob.mx/gobmx/salud/datos_abiertos/diccionario_datos_abiertos.zip)

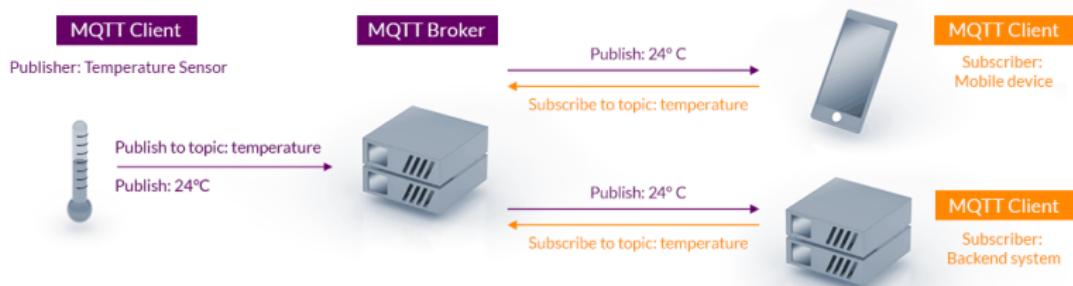


```
url = 'http://datosabiertos.salud.gob.mx/gobmx/salud/datos_abiertos/datos_abiertos_influenza_covid19.zip'; % la cadena no se parte
zip_filename = websave('covid19.zip',url)
data_filename = unzip(zip_filename)
% unzip(url)
```

- ① ¿Cuál es el porcentaje de confirmados de SARS-COV-2 que han fallecido?
- ② Dibuje un histograma de la edad de los confirmados y otro de la edad de los fallecidos.
- ③ ¿Qué fracción de los confirmados fallecidos son mujeres? Dibuje un diagrama de pie que muestre la distribución por sexo en los confirmados fallecidos.
- ④ ¿Cuál es la proporción intubados/confirmados a nivel nacional y cuál es esa proporción para residentes en Chiapas?
- ⑤ Dibuje un diagrama de burbuja que muestre en un mapa el número de casos confirmados en los municipios de “TUXTLA GUTIÉRREZ”, “TONALÁ” y “COMITÁN DE DOMÍNGUEZ”.
- ⑥ ¿En qué porcentaje aumenta la mortalidad en los diabéticos respecto al total de casos confirmados?

# MQTT

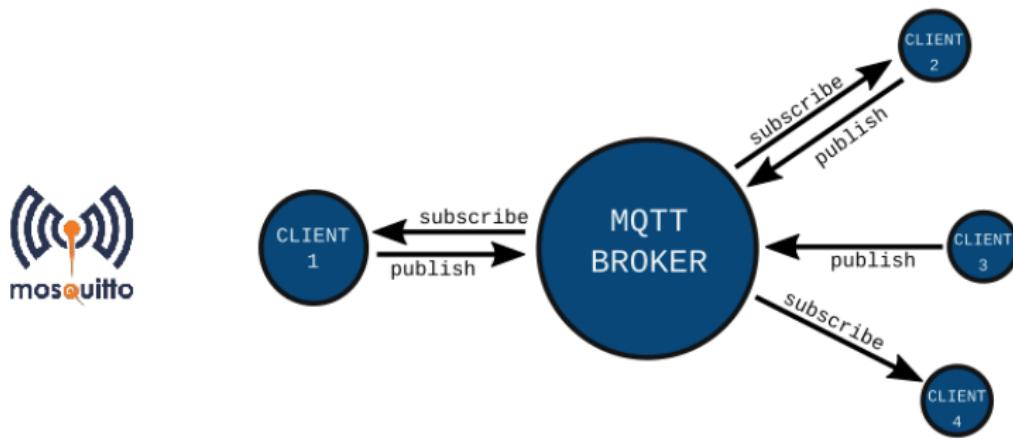
MQTT es un protocolo de mensajería que se utiliza para la comunicación de un equipo a otro. Los sensores inteligentes, los dispositivos portátiles y otros dispositivos de Internet de las cosas (IoT) generalmente tienen que transmitir y recibir datos a través de una red con recursos restringidos y un ancho de banda limitado.



**MQTT no es RESTful.** REST es un enfoque arquitectónico para la comunicación de red que utiliza el modelo de comunicación solicitud-respuesta entre los emisores y receptores de mensajes. Por el contrario, MQTT utiliza el modelo de comunicación **publicación-suscripción** en la capa de aplicación y requiere una **conexión TCP** permanente para transmitir mensajes de manera **push**.

# Mosquitto

**Mosquitto** (Eclipse Foundation) es un broker de código abierto que implementa el protocolo MQTT. **Mosquitto** es liviano y es adecuado para su uso en todos los dispositivos, desde computadoras de placa única de bajo consumo hasta servidores completos.



<https://mosquitto.org>

>\_ Terminal 1

```
mosquitto_pub -h test.mosquitto.org -t machine/temperatura -m "45.2 C"
```

>\_ Terminal 2

```
mosquitto_sub -h test.mosquitto.org -t machine/temperatura
```



```
mqttBroker = 'tcp://test.mosquitto.org:1883';
topic = 'machine/temperatura';
mqttClient = mqttclient(mqttBroker);
subscribe(mqttClient, topic);
disp('Esperando mensajes...');
while true
    msg = read(mqttClient);
    if ~isempty(msg)
        disp("Mensaje en " + msg.Topic + ": " + msg.Data);
    end
    pause(1)
end
```



Este formato de archivo (sufijo `.parquet`) admite una compresión y codificación muy eficientes en datos orientados a columnas. Con este formato, los esquemas de compresión se pueden especificar por columna, lo que permite una compresión y codificación eficientes de los datos.

<https://parquet.apache.org/>



Este formato de archivo (sufijo `.parquet`) admite una compresión y codificación muy eficientes en datos orientados a columnas. Con este formato, los esquemas de compresión se pueden especificar por columna, lo que permite una compresión y codificación eficientes de los datos.

<https://parquet.apache.org/>

Funciones de MATLAB para Parquet:

- `parquetread()` lee datos en columnas de un archivo Parquet.
- `parquetwrite()` escribe datos en columnas en un archivo Parquet.
- `parquetinfo()` obtiene información sobre un archivo Parquet.

A large, colorful word cloud centered around the words "thank you". The word "thank" is in red, "you" is in green, and "you" is in blue. Numerous other words in different languages are scattered around, such as "danke" in German, "merci" in French, "gracias" in Spanish, "mochchakkeram" in Korean, and many others. The background is white.