

# Big Data & Cloud



GOBIERNO  
DE ESPAÑA

VICEPRESIDENCIA  
PRIMERA DEL GOBIERNO  
MINISTERIO  
DE ASUNTOS ECONÓMICOS  
Y TRANSFORMACIÓN DIGITAL

SECRETARÍA DE ESTADO  
DE DIGITALIZACIÓN  
E INTELIGENCIA ARTIFICIAL

red.es

Centro de  
Referencia Nacional  
en Comercio Electrónico  
y Marketing

CRN  
Digital



UNIÓN EUROPEA

Barrabés

The Valley

"El FSE invierte en tu futuro"  
Fondo Social Europeo

# Big Data & Cloud



GOBIERNO  
DE ESPAÑA

VICERRENCIA  
PRIMERA DEL GOBIERNO  
MINISTERIO  
DE ASUNTOS ECONÓMICOS  
Y TRANSFORMACIÓN DIGITAL

SECRETARÍA DE ESTADO  
DE DIGITALIZACIÓN  
E INTELIGENCIA ARTIFICIAL

red.es

Centro de  
Referencia y Apoyo  
en Carrera Docente  
y Maestría

GRN  
Digital

GARANTÍA  
JUVENIL



UNIÓN EUROPEA

Barrabés

The Valley

"El FSE invierte en tu futuro"  
Fondo Social Europeo

# Big Data & Cloud



GOBIERNO  
DE ESPAÑA

VICEPRESIDENCIA  
PRIMERA DEL GOBIERNO  
MINISTERIO  
DE ASUNTOS ECONÓMICOS  
Y TRANSFORMACIÓN DIGITAL

SECRETARÍA DE ESTADO  
DE DIGITALIZACIÓN  
E INTELIGENCIA ARTIFICIAL

red.es

Centro de  
Referencia Nacional  
en Comercio Electrónico  
y Marketing

CRN  
Digital



UNIÓN EUROPEA



"El FSE invierte en tu futuro"  
Fondo Social Europeo



# METODOLOGÍA

- **Aula Virtual (20 horas “solo”)**

**Si se atiende y se practica, tiempo suficiente para asentar fundamentos.**

- **Conceptos teóricos**  
**(los necesarios y contexto)**
- **Actividades en clase**  
**(learning by doing)**
- **Evaluación final:**  
**Cuestionario**  
**(una respuesta correcta)**



# El lenguaje Base de Datos (relacionales) es SQL

NAME	TYPE	WEIGHT
bulbasaur	grass	15
charmander	fire	19
squirtle	water	20
pikachu	electric	13
oddish	grass	12
snorlax	normal	1014
mewtwo	psychic	269

```
SELECT name, type  
FROM pokemon  
WHERE type = 'grass';
```

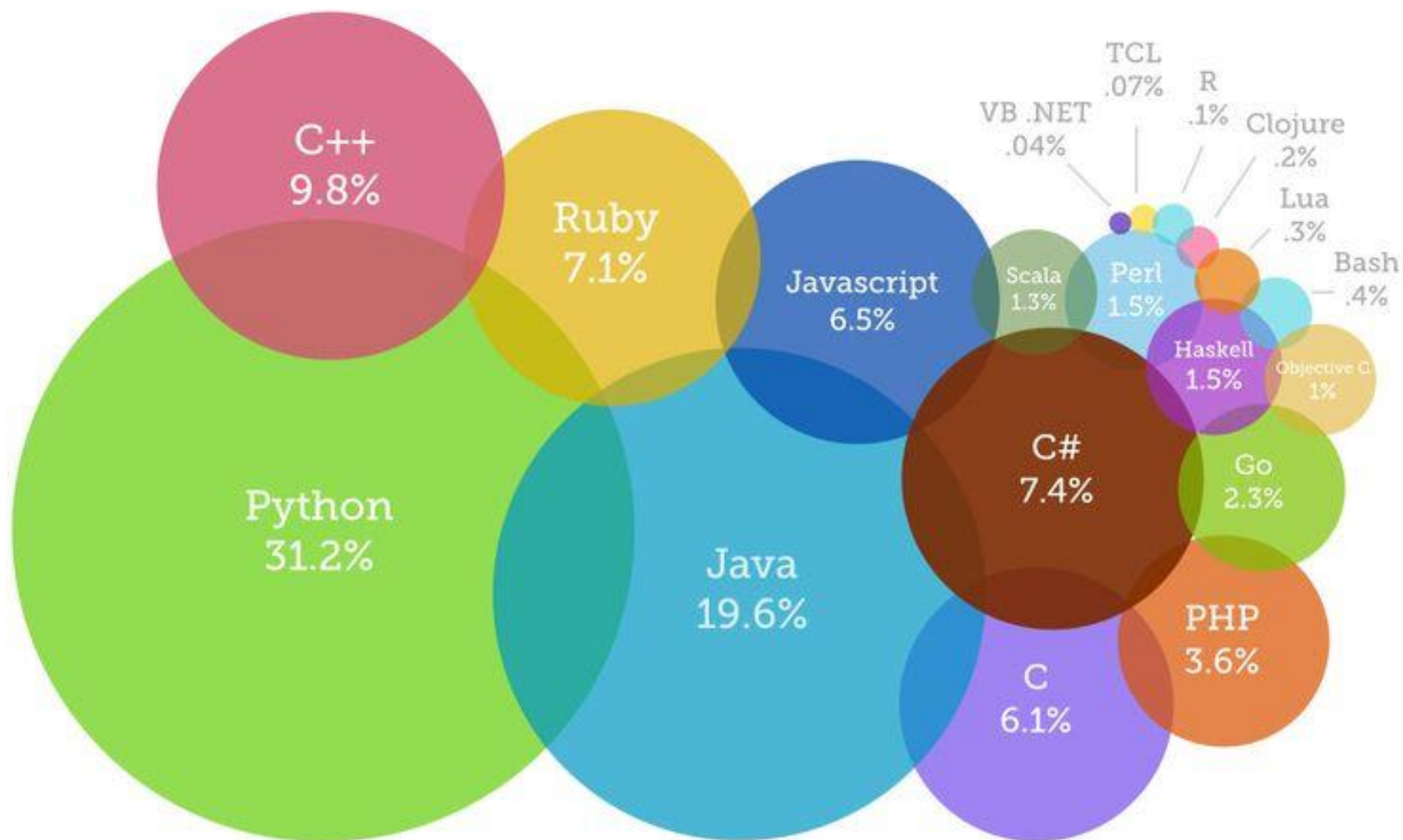
NAME	TYPE
bulbasaur	grass
oddish	grass

TÍTULO DE LA PRESENTACIÓN

# Python



# Selección del lenguaje de programación



## ¿Por qué Python?

*Lenguaje de **alto nivel**, fácil de aprender*

***Expresivo y legible***

*Sintaxis elegante y tipado dinámico y fuerte*

***Multiparadigma***

***Interpretado***

***Multiplataforma***

***“APRENDER A MONTAR EN BICICLETA”***



**PYTHON**



# Aprender un lenguaje de programación



TÍTULO DE LA PRESENTACIÓN

IDE:

¿Qué es un IDE?



# ¿Qué es un IDE?

## IDE

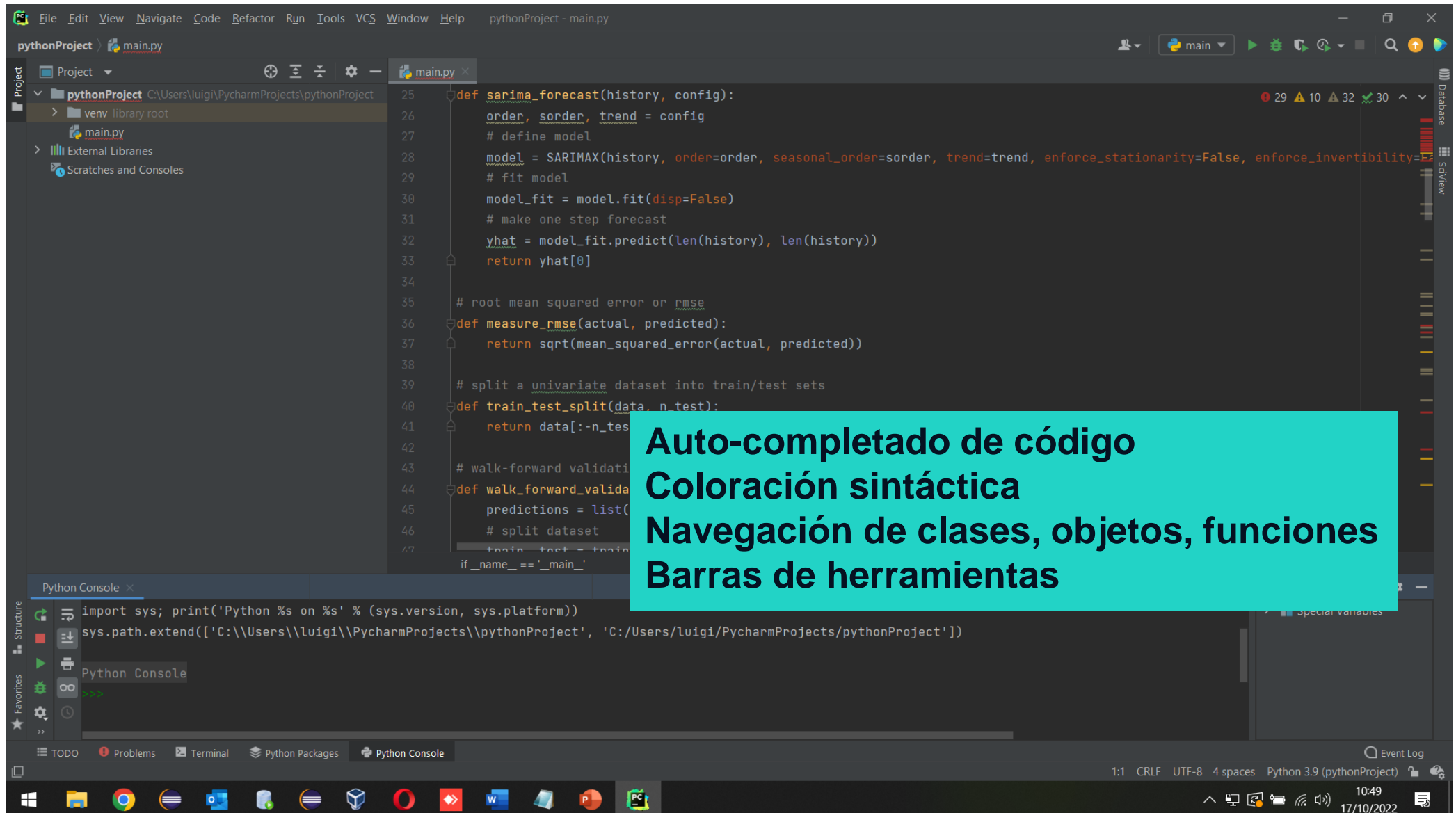
Es un software que nos permite desarrollar de una forma **más fácil**.



```
C:\Windows\system32\cmd.exe - py
D:\>py
Python 3.7.0 (default, Jun 28 2018, 08:04:48) [MSC v.1912 64 bit (AMD64)]
conda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 10 + 8
18
>>>
>>> 90 / 7
12.857142857142858
>>>
>>> 90 // 7
12
>>>
```



# Ventajas de usar un IDE:



# CRISP-DM

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
<b>Determine Business Objectives</b> <i>Background</i> <i>Business Objectives</i> <i>Business Success Criteria</i>	<b>Collect Initial Data</b> <i>Initial Data Collection Report</i>	<b>Select Data</b> <i>Rationale for Inclusion/Exclusion</i>	<b>Select Modeling Techniques</b> <i>Modeling Technique</i> <i>Modeling Assumptions</i>	<b>Evaluate Results</b> <i>Assessment of Data Mining Results w.r.t. Business Success Criteria</i> <i>Approved Models</i>	<b>Plan Deployment</b> <i>Deployment Plan</i>
<b>Assess Situation</b> <i>Inventory of Resources</i> <i>Requirements, Assumptions, and Constraints</i> <i>Risks and Contingencies</i> <i>Terminology</i> <i>Costs and Benefits</i>	<b>Describe Data</b> <i>Data Description Report</i>	<b>Clean Data</b> <i>Data Cleaning Report</i>	<b>Generate Test Design</b> <i>Test Design</i>	<b>Review Process</b> <i>Review of Process</i>	<b>Plan Monitoring and Maintenance</b> <i>Monitoring and Maintenance Plan</i>
<b>Determine Data Mining Goals</b> <i>Data Mining Goals</i> <i>Data Mining Success Criteria</i>	<b>Explore Data</b> <i>Data Exploration Report</i>	<b>Construct Data</b> <i>Derived Attributes</i> <i>Generated Records</i>	<b>Build Model</b> <i>Parameter Settings</i> <i>Models</i> <i>Model Descriptions</i>	<b>Determine Next Steps</b> <i>List of Possible Actions</i> <i>Decision</i>	<b>Produce Final Report</b> <i>Final Report</i> <i>Final Presentation</i>
<b>Produce Project Plan</b> <i>Project Plan</i> <i>Initial Assessment of Tools and Techniques</i>	<b>Verify Data Quality</b> <i>Data Quality Report</i>	<b>Integrate Data</b> <i>Merged Data</i>	<b>Assess Model</b> <i>Model Assessment</i> <i>Revised Parameter Settings</i>		<b>Review Project</b> <i>Experience</i> <i>Documentation</i>
	<b>Format Data</b> <i>Reformatted Data</i> <i>Dataset</i> <i>Dataset Description</i>				

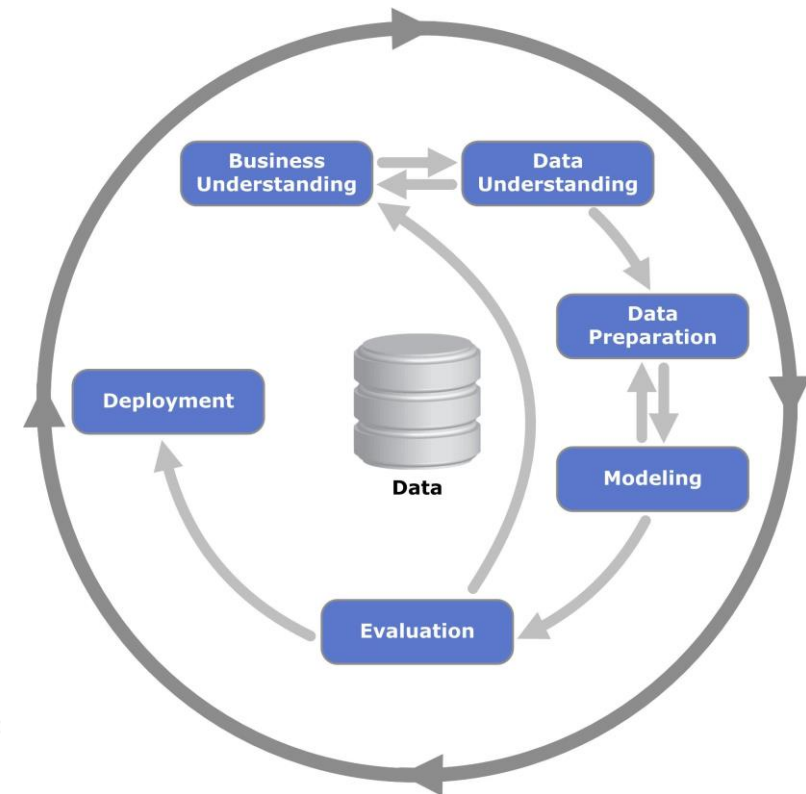


Figure 3: Generic tasks (bold) and outputs (italic) of the CRISP-DM reference model



# VARIABLES

Una variable es donde se almacenan y se recuperan los datos de un programa.

Existen distintos tipos de variables: texto, número, booleanos...

```
1 #Ejemplos de Variables
2 nombre = 'Luis'
3 dia = 2
4 soleado = False
```



```
nombreVariable1 = valor1
nombreVariable2 = valor2
...
```

# LISTAS

```
>>> jugadores = ['Batalla', 'Driussi', 'Alario', 'Mora', 'Auzqui', 'Rojas', 'Ponzio']
>>> jugadores
['Batalla', 'Driussi', 'Alario', 'Mora', 'Auzqui', 'Rojas', 'Ponzio']
>>> len(jugadores)
7
>>> jugadores[0]
'Batalla'
>>> jugadores[6]
'Ponzio'
>>> jugadores[-1]
'Ponzio'
>>> jugadores[3]
'Mora'
>>>
```

```
>>> lista = ["RIN", "LOIRA", "TAJO", "DANUBIO", "ELBA"]
>>> lista.insert(2, "TÍBER")
>>> lista
['RIN', 'LOIRA', 'TÍBER', 'TAJO', 'DANUBIO', 'ELBA']
>>> lista = ["RIN", "LOIRA", "TAJO", "DANUBIO", "ELBA"]
>>> lista + ["TÍBER"] ①
['RIN', 'LOIRA', 'TAJO', 'DANUBIO', 'ELBA', 'TÍBER']
```

DATA FUNDAMENTALS

# FUNCIONES



**MALAMADO**

Receta: Inés De Los Santos

2 oz de Whisky Bourbon  
1 oz de Malamado Viognier

1 dash de almíbar

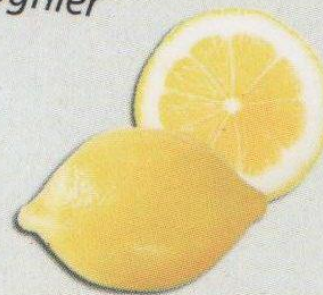
2 gajos de lima

2 gajos de limón

Gaseosa limón

Método: batido

Vaso super highball



En una coctelera aplastar los gajos de lima y limón, luego colocar los hielos y batir todos los ingredientes (menos la gaseosa). Servir en un vaso super highball y terminar con gaseosa lima-limón.

# FUNCIONES

Una función te permite definir un bloque de **código reutilizable** que se puede ejecutar muchas veces dentro de tu programa.

Una de las grandes ventajas de usar funciones en tu código es que reduce el número total de líneas de código en tu proyecto.

```
1 #Ejemplo de función
2 def cuadrado(n):
3     return n*n
4
5 cuadrado(2)
```

4



```
def nombreFuncion( ):
    instruc1
    instruc2
    ...
```



**¿Para qué repetir una y otra vez bloques de código una y otra vez?**

```
# función sin parámetros o retorno de valores
def diHola():
    print("Hello!")

diHola() # llamada a la función, 'Hello!' se muestra en la consola

# función con un parámetro
def holaConNombre(name):
    print("Hello " + name + "!")

holaConNombre("Ada") # llamada a la función, 'Hello Ada!' se muestra en la consola
```

**Por parámetro y sin parámetro (ARGUMENTOS)**



# FUNCIONES (Python)

## Funciones Built-in

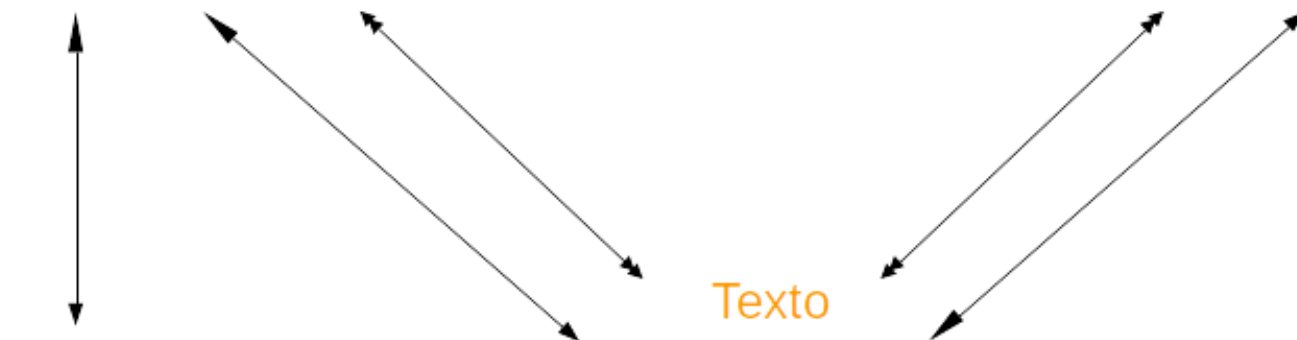
El intérprete de Python tiene una serie de funciones y tipos incluidos en él que están siempre disponibles. Están listados aquí en orden alfabético.

		Funciones Built-in		
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	<b><u>Funciones nativas y propias:</u></b>			
bin()				
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	<b><u>¡YA PODEMOS USARLAS!</u></b>			
bytes()				
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

# FUNCIONES (Python)

## Función Print (imprimir):

**Print ("Hola, esto es una cadena de texto")**



Función

Parámetros

Función ()  
Función (Parámetros)  
Función ("TEXTO")





DATA FUNDAMENTALS

# FUNCIONES (Python)

**Pip es nuestra gran aliada**



# NOSOTROS USAREMOS ESTO

Las librerías serán nuestro mayor aliado

**import solucion**





[Help](#)

[Sponsors](#)

[Log in](#)

[Register](#)

# Find, install and publish Python packages with the Python Package Index



Or [browse projects](#)

352,453 projects

3,180,044 releases

5,487,650 files

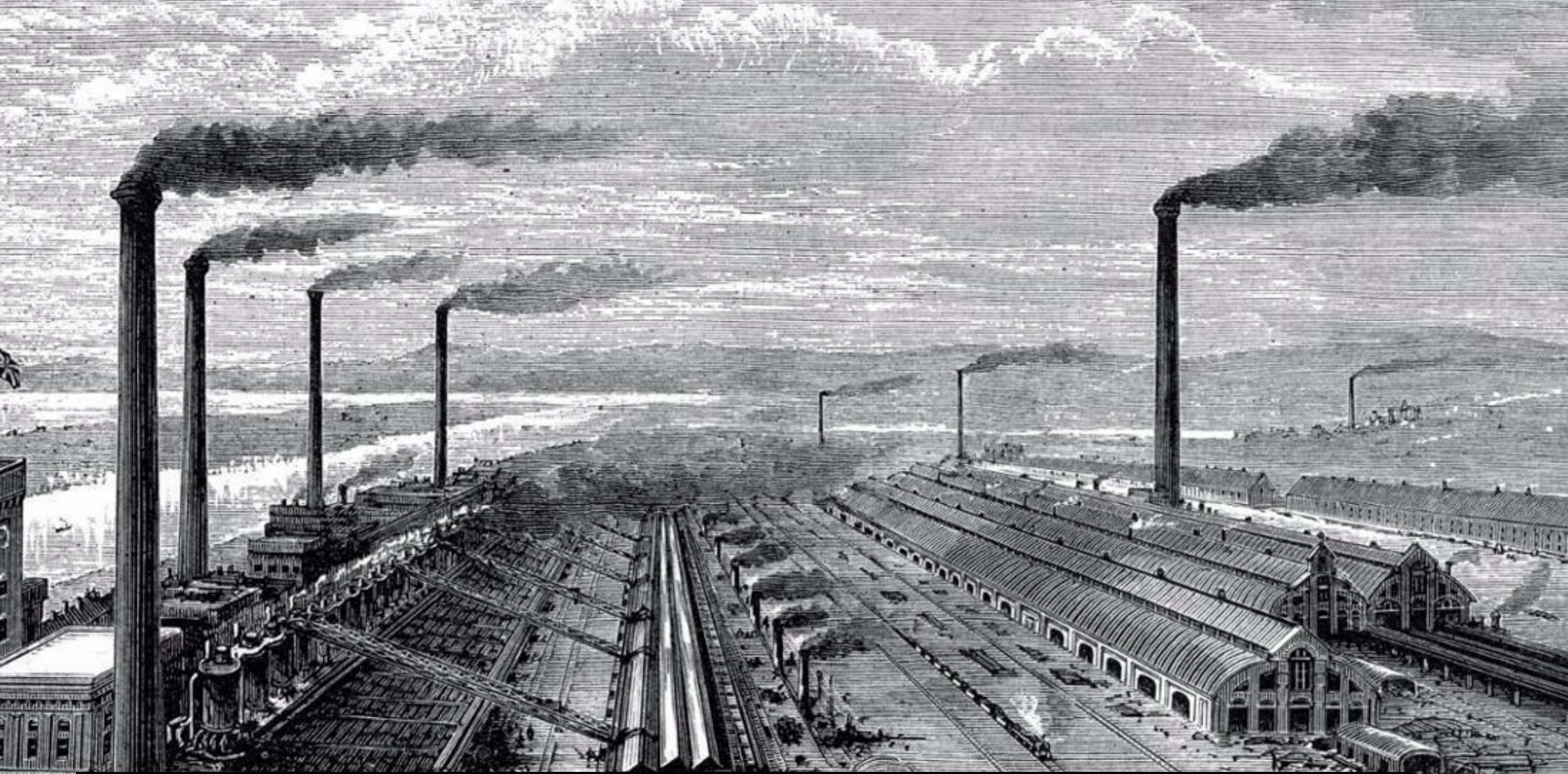
566,399 users



Data Science & Machine Learning

# Inteligencia Artificial



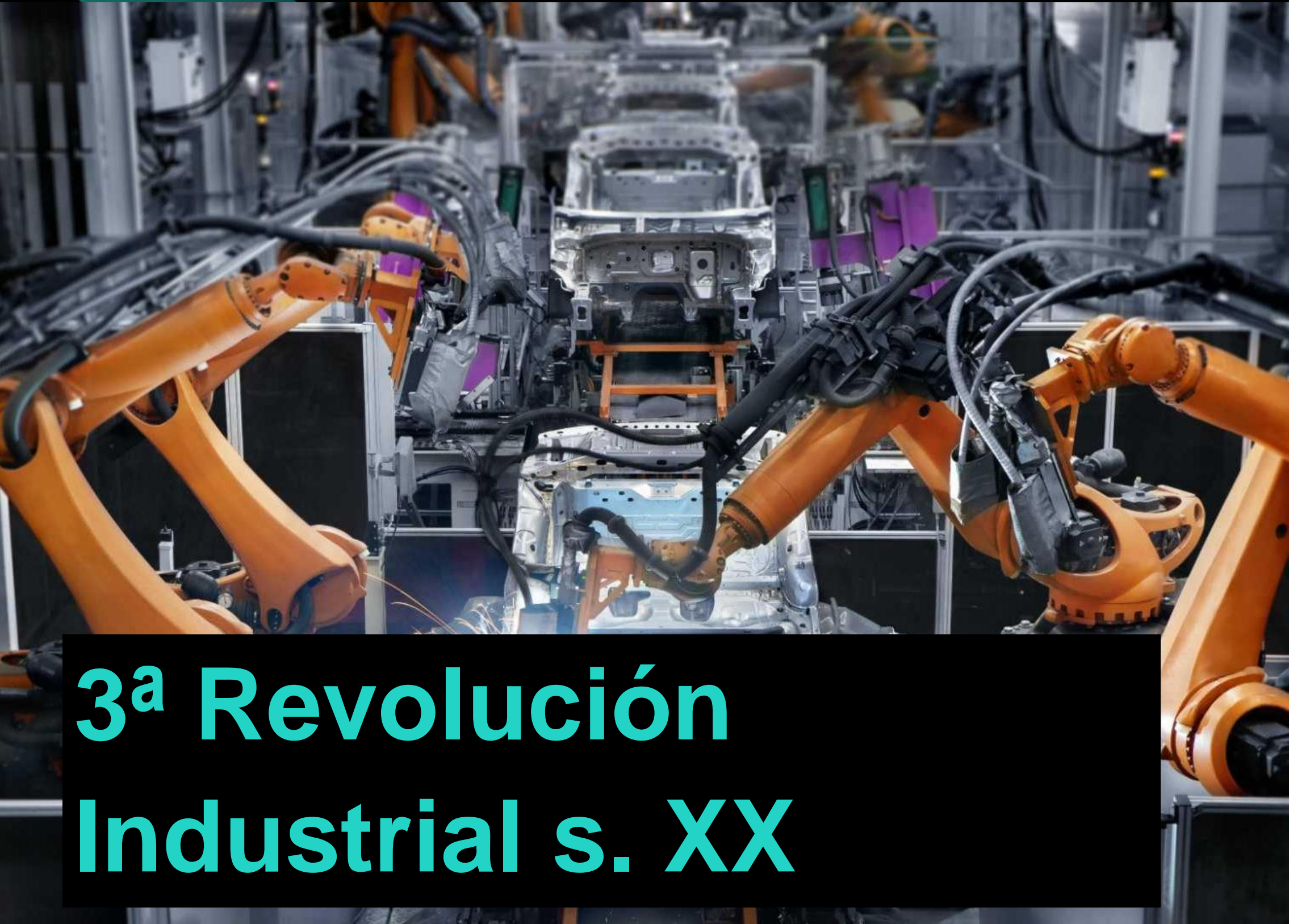


# 1ª Revolución Industrial s. XVIII





# 2ª Revolución Industrial s. XIX



# 3ª Revolución Industrial s. XX



The image shows a warehouse environment with several orange Kiva mobile robots. The robots are positioned on a polished floor, and in the background, there are tall industrial shelving units filled with various items. The robots have circular labels with the number '1348' and a logo. The text '4a Revolución Industrial s. XXI' is overlaid on the image in a large, white, sans-serif font.

# 4a Revolución Industrial s. XXI



# ¿Cuál es la diferencia?



# Inteligencia artificial



**Computer Vision**

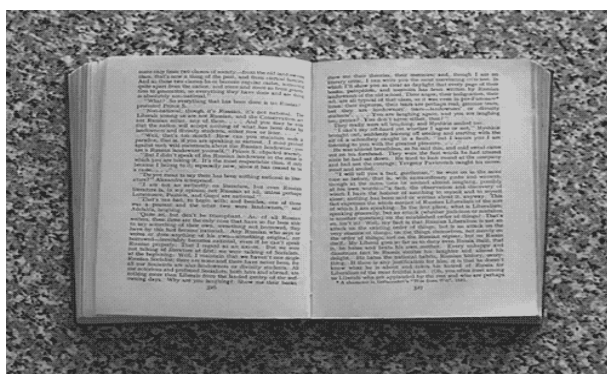
# Inteligencia artificial



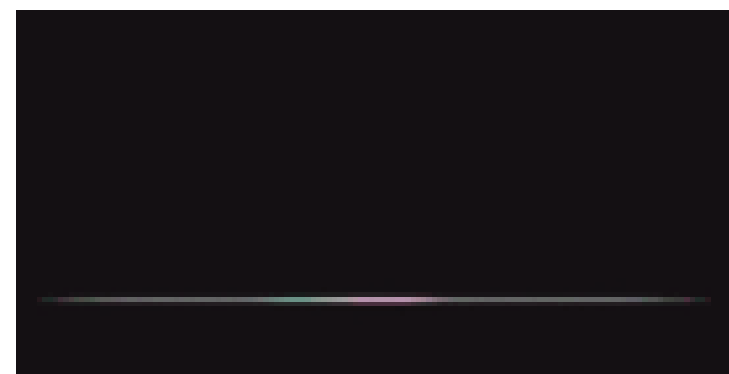
**COMPUTER VISION**  
Habilidad de ver



**ROBÓTICA**  
Habilidad de movilidad



**NLP**  
Habilidad de leer



**SPEECH**  
Habilidad de leer