

NOTAS

JAVIER ANDRÉS CAUSIL MARTÍNEZ

MATEMÁTICO, DESARROLLADOR FULL STACK Y ANALISTA DE DATOS

<https://github.com/Causil>

2 0 2 2

Índice general

Introducción	vi
I Mi pc	1
1. Instalando programas en mi pc	3
1.1. Terminal	3
1.2. Programas instalados desde software Ubuntu	4
II Python	5
2. Instalaciones en Python	6
2.0.1. Anaconda	6
2.1. Instalando Python	6
2.2. instalando TensorFlow	6
2.3. Instalando Jupyter Notebook	6
3. Numpy	8
3.1. Dimensiones en matrices	8
3.1.1. 0-D Arrays	8
3.1.2. 1-D Arrays	8
3.1.3. 2-D Arrays	9

3.1.4. 3-D Arrays	9
3.2. Numpy data types	11
3.3. Shape de un arreglo	12
3.4. NumPy Array Reshaping	13
3.4.1. Reshape de 1-D a 2-D	13
3.5. Métodos	15
 III Lenguajes del frontend	 16
 4. Figma	 17
 5. JavaScript	 18
 6. Typescript	 19
 7. Next.js	 20
7.1. Navegando entre páginas	21
7.1.1. Páginas en Next.js	21
7.1.2. Crea una nueva página	21
7.1.3. Componente Link	21
7.1.4. Usando “<Link:”	21
7.1.5. Navegación del lado del cliente	22
 IV Bases de Datos	 23
 8. MySQL	 24
8.1. Funciones de control de flujo	24
8.2. Subqueries	26

<i>ÍNDICE GENERAL</i>	iii
V Ciencia de datos	28
8.3. Proyectos data Analysis	30
9. Análisis de datos	31
9.1. ¿Qué tipo de información podemos analizar?	31
9.2. Flujo de trabajo en ciencia de datos: fases, roles y oportunidades laborales	32
9.3. Herramientas para cada etapa del análisis de datos	32
9.4. Python en ciencia de Datos	32
VI Matemáticas	33
10.Ecuaciones diferenciales	34
VII Ingles	35
11.Frases en inglés	36
12. Unknown words	37
13.Inglés A1	39
13.1. Análisis de habilidades	39
13.1.1. A and AN	41
13.1.2. IT'S SENTENCES	42
13.1.3. Plural forms	42
VIII Lecturas de libros	43
13.2. Ultralearning	44
13.2.1. Why Ultralearning Matters	44
Bibliografía	45

Índice de figuras

©EUREKA INFINITY 2022

Eureka infinity es mi proyecto personal, su finalidad es publicar todo lo relacionado con la matemática que yo vaya produciendo relacionado con mi entorno, ya se han cursos, avances de proyectos personales, hasta hoy día tengo poca experiencia profesional, pero con el tiempo, la disciplina y la constancia en el estudio, estaré creciendo gracias a la comunidad.

Introducción

Parte I

Mi pc

La inspiración no se puede provocar pero se puede estimular realizando tareas simples, antes de cometer otras mas complejas en cualquier caso es un fenómeno instintivamente humano y distinto para cada persona.

Capítulo 1

Instalando programas en mi pc

1.1. Terminal

Para instalar node ejecuto el comando

```
sudo apt-get install nodejs
```

Este node es el ultimo estable que se encuentra en los repositorios de Ubuntu, pero descargar la ultima versión ejecutando los siguientes comandos:

```
javier@javier-Lenovo-G40-80:~$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.3/ins
```

```
javier@javier-Lenovo-G40-80:~$ source ~/.bashrc
```

```
javier@javier-Lenovo-G40-80:~$ nvm list-remote
```

```
javier@javier-Lenovo-G40-80:~$ nvm install v16.15.0
```

```
javier@javier-Lenovo-G40-80:~$ sudo apt-get install npm
```

La versión es la necesaria. esa la encontramos en la pagina de nodejs.

Figma:

```
javier@javier-Lenovo-G40-80:~$ sudo snap install figma-linux
```

git:

```
sudo add-apt-repository ppa:git-core/ppa
```

```
sudo apt update
```

```
sudo apt install git
```

```
javier@javier-Lenovo-G40-80:~$ sudo apt-get update
Obj:1 http://co.archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://co.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Obj:4 http://co.archive.ubuntu.com/ubuntu jammy-backports InRelease
Leyendo lista de paquetes... Hecho
javier@javier-Lenovo-G40-80:~$ python3
```

```
javier@javier-Lenovo-G40-80:~$ sudo apt-get update
Obj:1 http://co.archive.ubuntu.com/ubuntu jammy InRelease
Des:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Des:3 http://co.archive.ubuntu.com/ubuntu jammy-updates InRelease [109 kB]
Obj:4 http://co.archive.ubuntu.com/ubuntu jammy-backports InRelease
```

Intalando MySQL

<https://dev.to/gсударshan/how-to-install-mysql-and-workbench-on-ubuntu-20-04-localhost-5828>

<https://dev.mysql.com/downloads/file/?id=509020>

<https://ubuntu.com/server/docs/databases-mysql>

1.2. Programas instalados desde software Ubuntu

Texstudio, vscode, Mysql Workbench Community

Parte II

Python

Capítulo 2

Instalaciones en Python

2.0.1. Anaconda

Contiene un grupo de librerías de python que son Numpy, Scipy, matplotlib, python, pandas, jupyter, scikitLearn, Spyder y para instalar librerías adicionales trae un gestor de paquetes que llama **conda**, podemos hacer la instalación de keras, TensorFlow, PyTorch.

<https://www.anaconda.com/products/distribution>

Vídeo explicando como instalar anaconda. <https://www.youtube.com/watch?v=kymjQ6e4jsQ>

2.1. Instalando Python

2.2. instalando TensorFlow

Para instalar TensorFlow nos podemos guiar de <https://www.tensorflow.org/install/pip>

Si ya tienes el entorno virtual instalado y activado solo es ejecutar en el entorno virtual

```
pip install --upgrade pip
```

2.3. Instalando Jupyter Notebook

Ejecutar Jupyter dentro del entorno virtual,

<https://es.acervolima.com/uso-de-jupyter-notebook-en-un-entorno-virtual/>

Una vez creado el entorno virtual ejecutamos

abrimos el Jupyter Notebook y procedemos a cambiar el kernel en la opción venv que es como se llama nuestro kernel, de esta manera podemos acceder a todos los paquetes instalados en nuestro entorno virtual. Para desactivarlo procedemos a ejecutar el siguiente comando:

```
jupyter-kernelspec uninstall venv
```

Esto fue consultado en [2]. En si solo es cambiar la dirección del kernel.

Pendiente hacer esto mismo en vscode

Capítulo 3

Numpy

3.1. Dimensiones en matrices

3.1.1. 0-D Arrays

0-D arrays, or scalars, are the elements in a array. Each value in an array is a 0-D array.

```
import numpy as np
```

```
arr = np.array(42)
```

```
print(arr)
```

```
output()
```

```
42
```

3.1.2. 1-D Arrays

An array that has 0-D arrays as its elements is called uni-dimensional or 1-D array.

Example

Create a 1-D array containing the values 1,2,3,4,5:

```
import numpy as np
```

```
arr = np.array([1, 2, 3, 4, 5])
```

```
print(arr)
```

```
print(type(arr))
```

```
output()
```

```
[1 2 3 4 5]
<class 'numpy.ndarray'>
```

No confundir con una lista de Python son totalmente diferentes.

3.1.3. 2-D Arrays

An array that has 1-D arrays as its elements is called a 2-D array.

These are often used to represent matrix or 2nd order tensors.

NumPy has a whole sub module dedicated towards matrix operations called `numpy.mat`

Example

Create a 2-D array containing two arrays with the values 1,2,3 and 4,5,6:
`import numpy as np`

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
```

```
print(arr)
```

3.1.4. 3-D Arrays

An array that has 2-D arrays (matrices) as its elements is called 3-D array.

These are often used to represent a 3rd order tensor.

Example

Create a 3-D array with two 2-D arrays, both containing two arrays with the values 1,2,3 and 4,5
`import numpy as np`

```
arr = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])
```

```
print(arr)
```

```
output()
```

```
[ [
  [1 2 3]
  [4 5 6]
]
 [
  [1 2 3]
```



```
[4 5 6]
]
```

Check Number of Dimensions?

NumPy Arrays provides the `ndim` attribute that returns an integer that tells us how many dimensions the array have.

Example

Check how many dimensions the arrays have:

```
import numpy as np

a = np.array(42)
b = np.array([1, 2, 3, 4, 5])
c = np.array([[1, 2, 3], [4, 5, 6]])
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

print(a.ndim)
print(b.ndim)
print(c.ndim)
print(d.ndim)
output()
0
1
2
3
```

Higher Dimensional Arrays

An array can have any number of dimensions.

When the array is created, you can define the number of dimensions by using the `ndmin` argument.

Example

Create an array with 5 dimensions and verify that it has 5 dimensions:

```
import numpy as np

arr = np.array([1, 2, 3, 4], ndmin=5)

print(arr)
print('number of dimensions :', arr.ndim)

output()
```

```
[[[[[1 2 3 4]]]]]  
number of dimensions : 5
```

In this array the innermost dimension (5th dim) has 4 elements, the 4th dim has 1 element that is the vector, the 3rd dim has 1 element that is the matrix with the vector, the 2nd dim has 1 element that is 3D array and 1st dim has 1 element that is a 4D array.

3.2. Numpy data types

Data Types in Python

By default Python have these data types:

strings - used to represent text data, the text is given under quote marks. e.g. "ABCD"

integer - used to represent integer numbers. e.g. -1, -2, -3

float - used to represent real numbers. e.g. 1.2, 42.42

boolean - used to represent True or False.

complex - used to represent complex numbers. e.g. $1.0 + 2.0j$, $1.5 + 2.5j$

Data Types in NumPy

NumPy has some extra data types, and refer to data types with one character, like i for integers, u for unsigned integers etc.

Below is a list of all data types in NumPy and the characters used to represent them.

- i - integer
- b - boolean
- u - unsigned integer
- f - float
- c - complex float
- m - timedelta
- M - datetime
- O - object
- S - string
- U - unicode string
- V - fixed chunk of memory for other type (void)

Shape of an Array

The shape of an array is the number of elements in each dimension.

Get the Shape of an Array

NumPy arrays have an attribute called shape that returns a tuple with each index having the number of corresponding elements.

The shape arrays is return tupla where, first element is

L aforma de un arreglo

Estudiando de https://www.w3schools.com/python/numpy/numpy_array_join.asp y python para Data Science

3.3. Shape de un arreglo

Un array o arreglo de numpy tiene un atributo llamado shape que retorna una tupla donde cada indice tiene el numero correspondiente de elementos, ejemplo:

Example

Print the shape of a 2-D array:

```
import numpy as np
```

```
arr = np.array([[1, 2, 3, 4], [5, 6, 7, 8]])
```

```
print(arr.shape)
```

output:

(2,4)

Nos \ 'indica que tenemos un arreglo de dos elementos de una dimensi\ 'on y cada elemenos de una dimensi\ 'on tiene cuatro elementos.

```
import numpy as np
```

```
arr = np.array([1, 2, 3, 4], ndmin=5)
```

```
print(arr)
```

```
print('shape of array :', arr.shape)
```

```
import numpy as np
```

```
arr = np.array([1, 2, 3, 4], ndmin=5)
```

```
print(arr)
```

```
print('shape of array :', arr.shape)
```

```
[[[[[1 2 3 4]]]]]
```

shape of array : (1, 1, 1, 1, 4)

3.4. NumPy Array Reshaping

Reshaping o reformar un array o arreglo significa cambiar la forma de un array, el shape de un arreglo es el número de elementos en cada dimensión. Al reshaping podemos agregar o eliminar dimensiones o cambiar el número de elementos en cada dimensión.

3.4.1. Reshape de 1-D a 2-D

Ejemplo:

Convirtamos el siguiente array de 1-D con 12 elementos en un arreglo de 2-D, donde la dimensión 2-D tendrá 4 elementos y cada elemento de una dimensión tendrá 3 elementos.

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

newarr = arr.reshape(4, 3)

print(newarr)
```

```
output:
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

Ahora convirtamos el arreglo de 1-D en un arreglo en 3-D, es decir:

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

newarr = arr.reshape(2, 3, 2)

print(newarr)
```

```
output:
```

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
```

```
newarr = arr.reshape(2, 3, 2)
```

```
print(newarr)
```

```
[[[ 1  2]
 [ 3  4]
 [ 5  6]]
```

```
[[ 7  8]
 [ 9 10]
 [11 12]]]
```

El elemento retornado tiene la misma dirección de memoria el arreglo original, es decir sucede lo siguiente:

```
import numpy as np
```

```
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
```

```
newarr = arr.reshape(4, 3)
```

```
newarr[1][0]=2
```

```
print(newarr)
```

```
print(arr)
```

output:

```
[[ 1  2  3]
 [ 2  5  6]
 [ 7  8  9]
 [10 11 12]]
```

```
[ 1  2  3  2  5  6  7  8  9 10 11 12]
```

Dimensión desconocida

Se le permite tener una dimensión desconocida.

Lo que significa que no tiene que especificar un número exacto para una de las dimensiones en el método de reshape.

Pasa -1 como valor y numpy calculará este número por usted.

Example

Convert 1D array with 8 elements to 3D array with 2x2 elements:

```
import numpy as np
```

```
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8])
```

```
newarr = arr.reshape(2, 2, -1)
```

```
print(newarr)
```

```
[[[1 2]
```

```
[3 4]]
```

```
[[5 6]
```

```
[7 8]]]
```

Note: We can not pass -1 to more than one dimension.

Aplanando los arreglos

Aplanar un arreglo multidimensional significa convertir una matriz multidimensional en una matriz 1D.

podemos usar reshape(-1)

Example

Convert the array into a 1D array:

```
import numpy as np
```

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
```

```
newarr = arr.reshape(-1)
```

```
print(newarr)
```

output:

```
[1 2 3 4 5 6]
```

3.5. Métodos

Parte III

Lenguajes del frontend

Capítulo 4

Figma

Capítulo 5

JavaScript

Funciones de Flecha,

ES5 //JSON.parse(JSON.stringify(paddockType));

Capítulo 6

Typescript

Capítulo 7

Next.js

El objetivo de aprender este lenguaje es darle asistencia a la página de As-research, junto con el blog, para esto estaré aprendiendo con fuentes de [3], para crear mis notas de Next.js.

Foro de discusiones: <https://github.com/vercel/next.js/discussions>

Requerimientos del sistema:

Node.js 12.22.0 or later

MacOS, Windows (including WSL), and Linux are supported

Cómo iniciar un proyecto?

Para construir una aplicación web completa con React desde cero, hay muchos detalles importantes que tu necesitas considerar:

1. Código tiene que ser empaquetado usando un empaquetador como webpack y transformarlo utilizando un compilador como Babel.
2. Debes realizar optimizaciones de producción, como la división de código.
3. Es posible que desee renderizar previamente estáticamente algunas páginas para mejorar el rendimiento y el SEO. También es posible que desee utilizar la representación del lado del servidor o la representación del lado del cliente.
4. Es posible que deba escribir algún código del lado del servidor para conectar su aplicación React a su almacén de datos.

```
npx create-next-app nextjs-blog
```

Para recorrer el servidor del proyecto es:

```
npm run dev
```

7.1. Navegando entre páginas

7.1.1. Páginas en Next.js

En Next.js, una pagina es un componente de React exportado desde un archivo en el directorio “pages”.

Las pages son asociadas con una ruta basada en el nombre del archivo. Por ejemplo, en desarrollo:

1. “pages/index.js” es asociado con la ruta “/”.
2. “pages/post/first-post.js” es asociado con la ruta “/posts/first-post”.

7.1.2. Crea una nueva página

Crea el directorio “posts” en la carpeta “pages”.

Crea un archivo llamado “first-post.js” dentro del directorio “posts” con el siguiente contenido:

```
export default function FirstPost() {  
  return <h1>First Post</h1>;  
}
```

La componente puede tener cualquier nombre, pero debes exportar esto como un export “default”.

Si corres el servidor de la aplicación te darás cuenta del cambio.

7.1.3. Componente Link

Al enlazar entre páginas en sitios web, utilizamos la etiqueta de HTML “<a>” .

En Next.js, utilizaremos la componente “Link” de “next/link” para envolver la etiqueta “<a>”. “<Link>” permite realizar navegación del lado del cliente en las diferentes paginas de la aplicación.

7.1.4. Usando “<Link>”

Primero, abrimos “pages/index.js”, e importamos la componente “Link” de “next/link” agregando esta línea en la parte superior:

```
import Link from 'next/link';
```

Utilidad:

```
<h1 className="title">
Read{', '}
<Link href="/posts/first-post">
<a>this page!</a>
</Link>
</h1>
```

Así nuestra componente “pages/posts/first-post.js” que daría de la siguiente manera:

```
import Link from 'next/link';

export default function FirstPost() {
  return (
    <>
    <h1>First Post</h1>
    <h2>
    <Link href="/">
    <a>Back to home</a>
    </Link>
    </h2>
    </>
  );
}
```

atributos de la componente link son href.

7.1.5. Navegación del lado del cliente

La componente “Link” permite en client-side navegar entre dos páginas en la misma app de Next.js

Parte IV

Bases de Datos

Capítulo 8

MySQL

8.1. Funciones de control de flujo

Case: Esta expresión nos permite realizar la condición y devolver el primer valor que cumpla con dicha condición

Ejemplo 8.1.1.

Primer ejemplo

```
select
case 1
when 1 then 'uno'
when 2 then 'dos'
else 'otro n\'umero'
end as valor;
```

segundo ejemplo:

```
select idFactura, idProducto,

case
when cantidad > 2 then 'M\'as de dos productos vendidos'
when cantidad = 2 then 'Dos productos vendidos'
else 'Menos de dos productos vendidos'
end as cantidad
from detalle_factura;
```

Tercer ejemplo

```
select nombre,
case
when email IS NULL then 'No tiene email registrado'
else 'email'
```

```
end as email,  
pais  
from cliente;
```

Podemos ver que es una sentencia muy similar a switch de vuelve el primer caso que cumpla la condición.

IF :

Ejemplo 8.1.2.

Primer ejemplo

```
select if(1 < 2, true, false) as resultado;
```

segundo ejemplo

```
select  
idProducto,  
if(cantidad > 1, cantidad*precioUnitario, precioUnitario) as total  
from detalle_factura;
```

Tercer ejemplo

```
select  
nombre,  
if(fechaIngreso < '2016-12-31', concat(idEmpleado, '-16'),  
if(fechaIngreso < '2017-12-31', concat(idEmpleado, '-17'),  
if(fechaIngreso < '2018-12-31', concat(idEmpleado, '-18'),  
concat(idEmpleado, '-19'))  
)  
)  
as codigo  
from empleado;
```

IFNULL y NULLIF: IFNULL nos permite evaluar una primera expresión, si esta expresión es null, entonces devolverá el segundo valor pasado por parámetro y NULLIF :

Ejemplo 8.1.3.

Primer ejemplo:

```
select ifnull(null, 'texto') as resultado;
```

Segundo ejemplo:

En este ejemplo devuelve los contactos de la tabla cliente en la columna nombre si tiene email nos da el email pero si este campo es null nos devuelve el tel\efono

```
select nombre, ifnull(email, telefono) as contacto  
from cliente;
```


Tercer ejemplo:

```
select nombre,
ifnull( (select email from cliente where idCliente = '14'),
'No tiene email registrado' )
as email
from cliente
where idCliente = '14';

select
nullif(
(select precioUnitario from producto where idProducto = 1),
(select )
)
```

NULLIF:

8.2. Subqueries

Es una declaración select en otra declaración, los subqueries devuelven datos de la consulta principal, los subqueries puede ser utilizados para agregar, actualizar, eliminar, enviar datos.

Ejemplo 8.2.1.

Ejemplo n\úmero 1:

Consiste en traer cuyos empleados tengan mayor salario al promedio:

```
select idEmpleado, nombre, salario
from empleado
where salario > (select avg(salario) from empleado);
```

Ejemplo 2: Seleccionamos los empleados que no pertenezcan al departamento general:

```
select nombre, apellido, idDepartamento
from empleado
where idDepartamento NOT IN (select idDepartamento
                               from departamento
                               where nombre like "%general%")
);
```

Ejemplo 3: facturas de los clientes que pertenezcan a Canada o Brasil:

```
select idCliente, idFactura
from factura
where idCliente IN( select idCliente
                    from cliente
                    where pais = 'canada' or pais = 'Brasil'
                    );
```

Subconsultas:

Ejemplo 8.2.2. `select *`

`from factura`

`where idCliente = (select idCliente from cliente where nombre = 'Jordi');`

`select *`

`from producto`

`where precioUnitario <=`

`(select avg(precioUnitario) from producto where idCategoria = 5)`

`and idCategoria = 5;`

comparando subconsultas

Subconsultas:

Ejemplo 8.2.3.

`select idProducto, nombre`

`from producto`

`where idProducto = ANY (select idProducto from detalle_factura);`

Parte V

Ciencia de datos

¿Qué es data ciencia? Es el proceso de descubrir información valiosa de los datos.

¿cuál es su finalidad?

1. Tomar decisiones y crear estrategias de negocio.
2. Crear productos de software más inteligentes y funcionales.

¿De que trata este proceso?:

1. Obtención de los datos: a través de encuestas
2. Transformar y limpiar los datos.
3. Explorar, analizar y visualizar datos.
4. Usar modelos de machine learning*.
5. Integrar datos e IA a productos de software.

¿Qué es Data Science?

Data science o ciencia de datos es el proceso de descubrir información valiosa de los datos.

¿Cuál es su finalidad? Tomar decisiones y crear estrategias de negocio Crear productos de software más inteligentes y funcionales.

¿De qué trata este proceso?

Obtención de los datos Mediciones Encuestas Internet

Transformar y limpiar los datos Incompletos Formato Incorrecto

Explorar, analizar y visualizar datos Patrones o tendencias Insights Visualizaciones, gráficos o reportes

Usar modelos de machine learning

Machine learning o aprendizaje automático es una rama de inteligencia artificial. Su objetivo es que las computadoras aprendan. En machine learning, las computadoras observan grandes cantidades de datos y construyen un modelo capaz de generar predicciones para resolver problemas.

Integrar datos e IA a productos de software Ponerlos a disposición del usuario final.

La ciencia de datos es una intercepción de conocimiento entre (matemáticas y estadística), (ciencias computacionales) y conocimiento del dominio.

8.3. Proyectos data Analysis

Poner en practica lo mas rápido que se pueda, tener proyectos personales, en que gasto los dineros del mes, que productos consigo cada mes, encontrar anomalías, proyectos con kaggle.

Capítulo 9

Análisis de datos

¿Qué es ciencia de datos y big data? ¿Cómo afectan a mi negocio?

“¿Qué haces en tu trabajo (como científico de datos)? Mi trabajo es crear una solución matemática o estadística para un problema del negocio”

9.1. ¿Qué tipo de información podemos analizar?

Descubrir qué tipos de información existen, qué industrias los usan y qué tipo de acciones podemos tomar a partir de ellos.

Los principales datos que existen son:

Personas: Este tipo de datos lo extraemos de las personas, es decir lo generamos nosotros cuando le damos like a una foto de facebook, de preferencia, tipo de videos, de quien te gusta mas el contenido, subiendo una foto y etiquetando a un compañero.

Transacciones: las monetarias y las no monetarias, cualquier transacción que hago con una tarjeta de crédito o débito, cuando hacemos un pago electrónico o digital queda una huella, queda un registro de quien lo hizo, por que monto lo hizo y en que establecimiento lo hizo, es muy interesante por que las bancas digitales pueden hacerte recomendaciones sobre el tipo de comercio que te podía interesar.

No financieras: las compañías telefónicas identifican cual es tu patrón habitual, cuantas llamadas haces, a que personas llamas, cuanto duran tus llamadas, y a partir de esto te llaman para que no abandones el servicio.

Navegación web: Estas son las famosas cookies, ellas están advirtiendote de la información que van a recoger.

Machine 2 machine: Una conexión de una máquina a otra máquina, la usan las plataformas de transporte, google maps y para hacer la locación entre dispositivos.

Biométricos: Cada vez son mas habituales y únicas, huellas digitales, reconocimiento facial.

9.2. Flujo de trabajo en ciencia de datos: fases, roles y oportunidades laborales

Roles en datos:

Ingeniero de datos: crear bases de datos Hacer que la empresa, hace la conexión de los dispositivos y las bases de datos,

Analista business intelligence: A partir de la información que ha creado el ingeniero de datos va a extraer la data, crear cuadros de control, crear dashboard, monitoreo, va a automatizar estos procedimientos para que cualquier persona de la empresa pueda interpretarla, las herramientas más utilizadas son SQL y Excel. No necesariamente sabe Python.

Data Scientist: Sabe hacer el rol del analista, sabe extraer la información y sabe predecir, con las herramientas de estadística, nos guía a donde vamos.

Data Translator: Nos ayuda a proyectar el equipo, nos ayuda a comunicar con los otros equipos del negocio.

9.3. Herramientas para cada etapa del análisis de datos

El primero es el rol del analista y del ingeniero estas son las personas que crean bases de datos y utilizan SQL, se sintetiza la información de la base de datos.

El científico de datos son herramientas predictivas, son R y Python, R es más estadístico análisis descriptivo,

9.4. Python en ciencia de Datos

Por que numpy para el análisis de datos. Tenemos tres cosas a destacar

1. Un poderoso objeto array multidimensional.
2. Funciones matemáticas

Crear un virtual environments ejecutamos la siguiente línea de comando

```
python3 -m venv my_env
source bin/activate
```

Parte VI

Matemáticas

Capítulo 10

Ecuaciones diferenciales

En este capítulo vamos a entender que es una ecuación diferencial.

Para comenzar a formular lo que es una ecuación diferencial, comencemos explorando lo que es la en sí, la palabra ecuación, de donde proviene, ¿Cómo hemos trascendido los seres humanos en el pensamiento de las ecuaciones? ¿Para que sirven las ecuaciones y por ende donde aplicarlas? Es un tarea muy dura, pero lo vamos a lograr, formulando preguntas para ir llegando a respuestas y de estas respuestas dar el salto a lo que yo llamo familiarización intelectual.

¿Por qué planteamos ecuaciones? tratemos de acercarnos a esta pregunta observando la siguiente ecuación

$$y + 5 = 0 \tag{10.1}$$

¿Qué elementos involucrados conocemos en la ecuación 10.1?

Los números 5 y 0, son dos enteros y la variable y representando lo desconocido en la ecuación, es decir, lo que queremos encontrar, si nos enfocamos en ese sentido, llegaremos a que $y = -5$.

$$y(x) = y'(x) \tag{10.2}$$

Note que tenemos una igualdad de funciones, una ecuación que se cumple solo para las funciones que son soluciones de la misma. ¿Cuales son estas funciones? las funciones $c \exp(x)$, de esta manera podemos definir este tipo de ecuaciones, ¿Por qué es importante las ecuaciones diferenciales? ¿Qué fenómenos describe el cálculo? en intervengan, la distancia, velocidades y aceleraciones.

Definición 10.0.1 (Ecuación diferencial). Una ecuación diferencial es cualquier ecuación que contiene las derivadas de una o más variables dependientes con respecto a una o más variables independientes.

Ejemplo 10.0.2.

$$\frac{dy}{dx} + \frac{dz}{dx} = 2y + z, \tag{10.3}$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial t^2} - 2 \frac{\partial u}{\partial t} \tag{10.4}$$

Parte VII

Ingles

Capítulo 11

Frases en inglés

After you: Después de usted.

Check it out: Échale un vistazo.

I guess so: Supongo que sí.

What a mess: Qué desastre.

Good for you: Bien por ti.

to get one is hopes up

Capítulo 12

Unknown words

Which: cual ; pronombre: que.

Shall: debera.

use: usar.

brief: breve.

either: conjunción : o, adjetive: Cualquiera de los dos, uno u otro; Adverbio: también.

Roughly speaking: mas o menos

behave: comportarse

obey: obedecer

listed: listado

in that: en eso

for most: para la mayoría

may : poder, ser posible.

well : bien

to allow for this : para permitir esto

rather than : más bien que

itself : sí mismo

this means : esto significa

so are : también lo son

such : tal

below : abajo

performance : actuación, el rendimiento, desempeño

us out of : nosotros fuera de

given : dado

on this : en este

unless : a no ser que

must : deber

we leave it : lo dejamos

want : desear

dwel : residir

happen : suceder

least : el menos

Often : Con frecuencia

exposure : exposición

better : mejor

worry : preocuparse

about : sobre

vowels :

consonants :

Capítulo 13

Inglés A1

Marco Común Europeo de Referencia para las Lenguas (MCER):

13.1. Análisis de habilidades

A continuación, encontramos un análisis de los diferentes componentes del idioma y las habilidades individuales que un estudiante debería tener para recibir un certificado A1 de inglés:

Compresión auditiva

Los estudiantes de nivel A1 deberían ser capaces de entender un inglés estándar simple, siempre y cuando sea hablado con claridad por alguien paciente y dispuesto a ayudar. Específicamente, deberían poder reconocer frases y palabras comunes, relacionadas con ellos mismos, su entorno y aquellos cercanos a ellos. Además, deberían entender cosas como los números, direcciones y otras instrucciones muy básicas en inglés.

Welcome everyone to the basic English course. A one for beginners. I'm your teacher, kira Sage, nice to meet you. Who am i? I am from the U.S.A I have taught C level executives and software engineers at companies like google skhynix, Tesla, Sony and more. I have nine years of teaching experience in countries like Usa and Japan. Here are a couple of interesting facts about me. I have visited four of the six Disneyland's around the world. This is me at Star Wars, Galaxy edge in Anaheim California. Also i have Master's degree in teaching T. Saw that means I can teach teachers and learners like you in this course you will learn the alphabet ah and un sentences with it's plural forms, sentences with there are numbers, colors, subject pronouns, professions, greetings, negative and interrogative statements, possessive adjectives. Days of the week, your hobbies, questions with what's your and to review everything from this course we'll have to wrap up class. Here's you'll learn all of the course concepts through worksheets, find the worsheets in the resources section through interactive quizzes and through interactive explanations on an imaginary website called plattso plattso is where we practice english so we can make friends online. Are you ready for this fun adventure? I'll see you in the next class.

Bienvenido todos al curso the Ingles básico. A1 para comenzar

Intento 1:

Welcome everyone to the Black Sea English course A1 for beginners I am here t-shirt carousel nice to me you Hoang I who on my I'm from the USA I have to see level executive a software engineer and companies like Google sky and Tesla Sony and more I cannae jeers of teaching experience in countries like facts about me I had visits of the seeds Disneyland around the world this is me at Star Wars Ga massive degree in teaching t so that means can teach t-shirts in Learners like you in this cour sentence with it's pure Air Force sentence with there are numbers caller surgeon pronounce prof status. Osiria Jetties days of the weave your hobbies question with what you ain't to review ev we all have to grab a glass of the course Concepts world war ships find a wall sheets in the re crisis and throw into righteous play Nations oh imagine a website called black English so we ca ready for this funeral bencher Adventure I also see you in the next class.

Segundo intento:

welcome everyone to the basic English course. a wife for beginners at Food dock. Avon for beginners sorry please thought a blunt for beginners. A1 for beginners. I am your teacher call Ma Kira say Asahi call Mama, nice to meet you. who I who am I. I am from the USA. I hopped out I have towels. IHOP takeout see Liberty Security in Sulphur and genius and companies like Google Skynet call mistakes Le gamma Sony a moor Anne Moore Anne Moore Sony and more. I have 9 years of teaching experience in countries like USA in Japan. Cher here. T R A Copley. here are a couple off interesting facts about me. I had visit visit. I had visited full of the sea Disneyland around the world. this is me and it's sad worse, Galaxy Edge in Anaheim California. also I have. also I had Mercer metzer Meister Meister mustard master. also I heart muscle is degree and teach t. all soup IHOP Massillon he's they greet and teach tea. soda means I can teach t-shirts in layers

<https://www.youtube.com/watch?v=EVNhOAEW784>

<https://howjsay.com/how-to-pronounce-taught> Para pronunciar Taught comience colocando su lengua detrás de sus dientes superiores y agregue el sonidoo AH corto y abierto termine con una T colocando su lengua detrás de sus dientes superiores

Numbers class Two:

Welcome to the 1st Module. This english fundamentals in this module. In this Module you will learn the

1. alphabet.
2. A and N
3. sentences with it's plural forms
4. and sentences with there are

this is the alphabet. Now let's go and learn on platzo.

Welcome to platzo. Now loading oh do you know what this is?

it's capture. We need letters to unlock the capture. Well let's learn in this class.

You will learn the letters of the english alphabet. In the English Alphabet there are 26 letters. Let's practice. I will say a letter and repeat after me. Ready, let's go. A B Bravo. C D E. Excellent F G. Good job H I J K, keep it up. L M magnificent N. Nice work. O P perfect Q R S super T terrific. U V very good W wonderful X. Why Z in some countries this letter is zed.

Now, you know the letters of the alphabet? A B C D E F G H I J K L M N O P Q R S T u v w x, y z. OR ZED. Let's use these letters to unlock the capture on platzo.

The capture says a phrase, not a robot. Let's spell the phrase letter by letter N O T A R O B O T. You did it. You unlock the capture.

Great job. it's your turn. Here is the alphabet but some letters are missing.

Type, the missing letters in the discussion panel. Don't forget to download your worksheet from the resources section that worksheet is your platzo profile. Complete the user name area of your platzo profile on your worksheet, keep this worksheet with you at all times. We will complete your platzo profile in each module of this course. So keep it ready for extra practice. Type your user name in the discussion panel.

My user name is Kira K I R A. I'll see you in the next class.

Email: jcausilmartinez@gmail.com

Number phone: 314 796 81 19.

User name: Javier Andres Causil Martínez

My introduction is in Spanish:

Hola mi nombre es Javier Andrés Causil Martínez, soy de Colombia del departamento de Antioquia en la ciudad de Cauca. Tengo 27 años, soy Matemático de la Universidad de Antioquia y programador, me gusta la música, hacer deporte, aprender cosas nuevas. Siento gran pasión por la música clásica.

13.1.1. A and AN

A and AN refer to only one person, place, or thing, or noun.

An is an article that we use with vowels. Vowels are "a", "e", "i", "o", "u"

Use "an" with these vowels. Now, let's use them together on Platzo.

"A" is the article we use with consonants.

Be careful! there is an exception. The most common exception is with the letter U.

Now you can use “a” and “an” with vowels and consonants.

13.1.2. IT’S SENTENCES

In this class, you will learn sentences with “it’s”.

First, use “it’s” or “it is” with a noun,

it is + a person: a thing: an item: a place

We can make sentences with “it’s, an article, and a noun”.

In this class, you will learn forms plurals,

13.1.3. Plural forms

The most common plural form is to add “s” at the end of the word. Para otras palabras que terminan con “S,SH,CH,X” y “Z”, we add “es” to the end of those words.

Words with “y” are special. because when a “y” is next to a vowel, we just add “S”, when a “y” is next to a consonant, we change the “y” to “i” and add “es”.

Parte VIII

Lecturas de libros

13.2. Ultralearning

Comienza con la ilusión de estudiar en el MIT, pero termina estudiando negocios en la Universidad de Manitoba, una escuela Canadiense de rango medio, ya que era la que podía pagar. Pero al terminar la carrera concluyo que se equivoco de carrera cuatro años mas tarde se da cuenta de una especialización en administración y de informática

13.2.1. Why Ultralearning Matters

Cuándo aplicar el ultraaprendizaje

1. es seguir ultralearning a tiempo parcial.
2. es buscar el ultraaprendizaje durante las brechas en el trabajo y la escuela.
3. Es integrar los principios del ultraaprendizaje en el tiempo y la energía que ya dedica al aprendizaje.

13.2.2. El valor del ultraaprendizaje

La capacidad de adquirir habilidades duras de manera efectiva y eficiente es inmensamente valiosa. No solo eso, sino que las tendencias actuales en economía, educación y tecnología van a exacerbar la diferencia entre quienes tienen esta habilidad y quienes no la tienen.

Bibliografía

- [1] S.C. Brenner, L.R. Scott *The Mathematical Theory of Finite Element Methods*, 3^a ed., USA: Springer, 2008.
- [2] “Uso de Jupyter Notebook en un entorno virtual”, 2022. [En línea]. Disponible en: <https://es.acervolima.com/uso-de-jupyter-notebook-en-un-entorno-virtual/>. [Accedido: 21-jun-2022]
- [3] “Create a Next.js App”, 2022.[En línea]. Disponible en: <https://nextjs.org/learn/basics/create-nextjs-app>. [Accedido: 15-jun-2022]