

A decorative vertical bar on the left side of the page, with a light blue arrow pointing right from its center. The arrow contains the date "2-9-2020".

2-9-2020

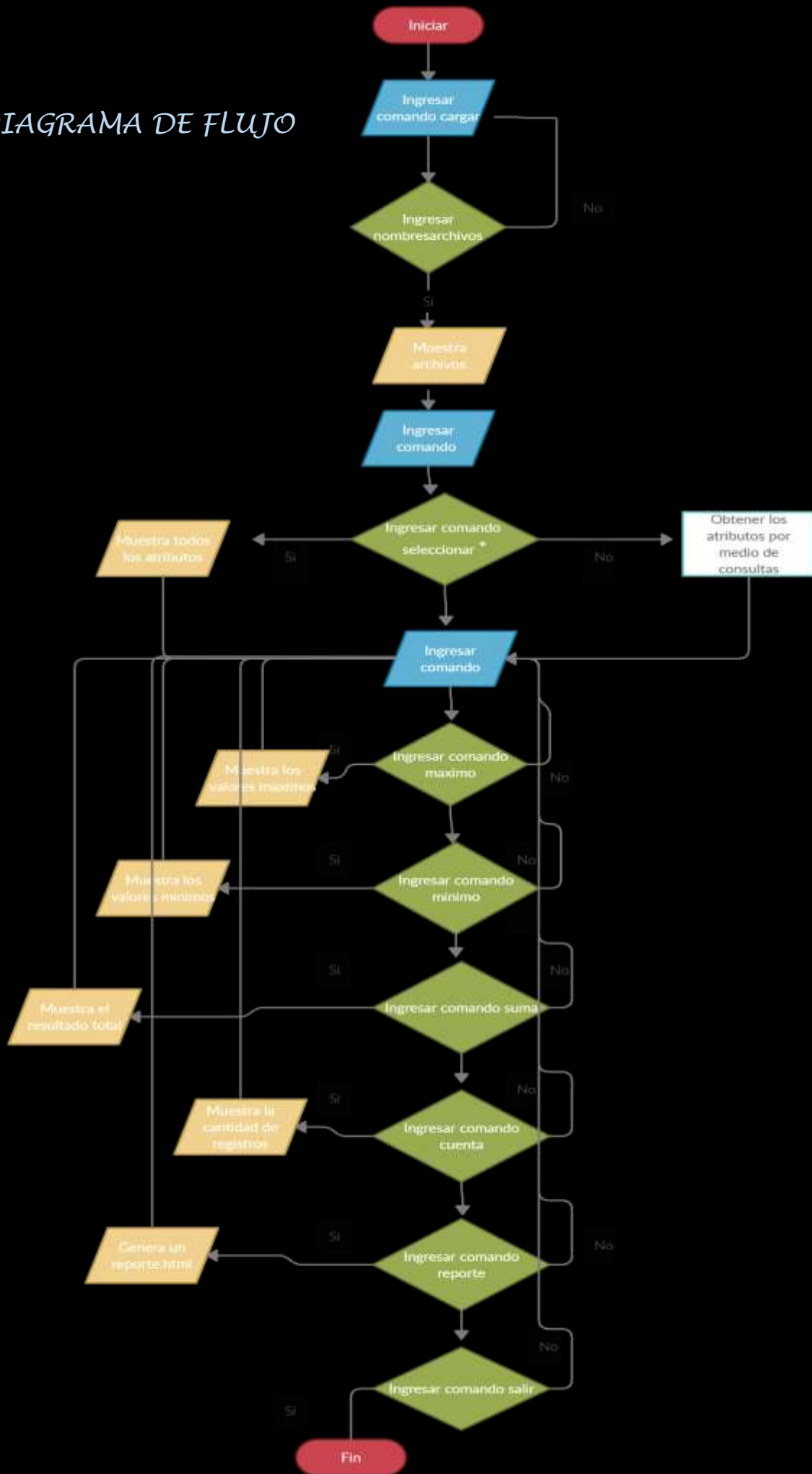
MANUAL TÉCNICO

Práctica Lenguajes Formales de
Programación

A series of thin, light blue wavy lines in the bottom left corner, resembling stylized grass or reeds.

Luis Ruben Quisquinay Hernandez
201701084

DIAGRAMA DE FLUJO



Archivos a utilizar:

```
▼ PRACTICA LENGUAJES
  > __pycache__
  {} archivo1.json
  {} archivo2.json
  📄 operadores.py
  <> reporte.html
  📄 reporte.py
  📄 Simpleql.py
```

Funciones a utilizar en el archivo Simpleql.py:

```
def funciones():
    |
    function = True
    while function:

        op = input("Escribe una funcion ").split(" ")
        op[0] = op[0].lower()

        if op[0] == "cargar":
            cargar(op)

        elif op[0] == "seleccionar":
            seleccionar (op)

        elif op[0] == "maximo":
            maximo(op)

        elif op[0] == "minimo":
            minimo(op)

        elif op[0] == "suma":
            suma(op)                                # Utilizo el .lower

        elif op [0] == "cuenta":
            cuenta(op)

        elif op[0] == "reportar":
            generar_reporte(op[1])
        elif op[0] == "salir":
            break

        else :
            print("Error de comando")

funciones()
```

Método cargar:

```
def cargar(op):
    op.remove("cargar")
    #nombres_archivos = op[0].split(",")
    for x in op:
        efe = x
        print (x)
        if x.__contains__(","):
            efe = x.replace(",", "")
        with open (efe) as file:
            data = json.load(file)
            registros.extend(data)
            print(registros)
```

Método Seleccionar:

```
def seleccionar (op):
    #print(len(registros))
    op.remove("seleccionar")
    #arregloaux = op[0].split(" ")
    # nombre,edad,promedio,activo DONDE nombre = "Ruben"
    if op[0] == "+":
        for x in registros:
            print( "Nombre---->" +x["nombre"],"Edad---->" +str(x["edad"],"Activo---->" +x["activo"],"Promedio---->" +str(x["promedio"]) )
```

Método Máximo:

```
def maximo(op):

    op.remove("maximo")

    if op[0] == "edad":
        items = [reg ["edad"] for reg in registros]
        print(max(items))

    elif op[0] == "promedio":
        item1 = [reg ["promedio"] for reg in registros]
        print(float(max(item1)))
```

Método Mínimo:

```
def minimo(op):  
    op.remove("minimo")  
  
    if op[0] == "edad":  
        items = [reg ["edad"] for reg in registros]  
        print (min(items))  
  
    elif op[0] == "promedio":  
        items = [reg ["promedio"] for reg in registros]  
        print (float(min(items)))
```

Método Suma:

```
def suma(op):  
  
    op.remove ("suma")  
  
    if op[0] == "edad":  
        items = [reg ["edad"]for reg in registros]  
        print (sum(items))  
  
    elif op[0]==("promedio"):  
        items = [reg ["promedio"]for reg in registros]  
        print (float(sum(items)))
```

Método Cuenta:

```
def cuenta (op):  
    print(len(registros))
```

Método Reportar:

Para este método se utilizó otro archivo llamado `reporte.py` con el objetivo de hacerlo más ordenado y utilizando cómodamente las etiquetas de `html`.

Se importaron las librerías `json` para trabajar los archivos, `webbrowser` para trabajar `html`

```
import json
import webbrowser
from os import linesep
import os

def reporte(registros,nreg):

    report = open("reporte.html","w") # para abrir el archivo
    report.write("<!DOCTYPE html>"
+ "<html>"
+ "<head>"
+ "<title>REPORTE</title>"
+ "<meta charset='utf-8'>"
+ "<meta name='viewport' content='width=device-width, initial-scale=1'>"
+ "<link rel='stylesheet' href='https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css'> "
+ "<style>"
+ "body {"
+ "background-color: Slategray;"
+ "}"
+ "</style>"
+ "</head>"
+ "<body>"
+ "<div class='container'>"
+ "<h1>REPORTE </h1>"
+ "<table class='table'>"
+ "<thead>"
+ "<tr>"
+ "<th>Nombre</th>"
+ "<th>Edad</th>"
+ "<th>Activo</th>"
+ "<th>Promedio </th>"
+ "</tr>"
+ "</thead>"
+ "<tbody>"
+ linesep
```

Se encerraron las etiquetas de `html` con comillas para que no se reconocieran como tokens

```

    cont_alternacion = 0
    for i in range (0, nreg):

        clasetr = ""
        if cont_alternacion == 0:
            clasetr = "<tr class=\"table-primary\">"
            cont_alternacion += 1
        elif cont_alternacion == 1:
            clasetr = "<tr class=\"table-success\">"
            cont_alternacion += 1
        elif cont_alternacion == 2:
            clasetr = "<tr class=\"table-danger\">"
            cont_alternacion +=1

        elif cont_alternacion == 3:
            clasetr = "<tr class=\"table-info\">"
            cont_alternacion = 0

        x = registros[i]

        report.write(clasetr + "\n"
+ "<td>" + x["nombre"] + "</td>"
+ "<td>" + str(x["edad"]) + "</td>"
+ "<td>" + x["activo"] + "</td>"
+ "<td>" + str(x["promedio"]) + "</td>"

        + "</tr>"
+ linesep

    )

    report.write(
        "</tbody>"
+ "</table>"
+ "</div>"

```

Se utiliza la instrucción for para recorrer y mostrar los datos del archivo json colocándolo en un rango ya que si se indica una cantidad de archivos que no cuenta el archivo, el programa indicará error.

```

    )

    report.write(
        |         | "</tbody>"
    + "</table>"
    + "</div>"

    + "</body>"
    + "</html>"
    )
    report.close()
    os.system("reporte.html")

```

En el archivo reporte.py se elabora el método generar reporte para luego ser llamado.

Utilizando como parámetro nreg ya que será la encargada de la cantidad de archivos que el usuario coloque.

```

def generar_reporte (nreg):
    try:

        nreg=int(nreg)
        if nreg <= len(registros) :

            reporte(registros,nreg)
        else:
            print("No hay pan")

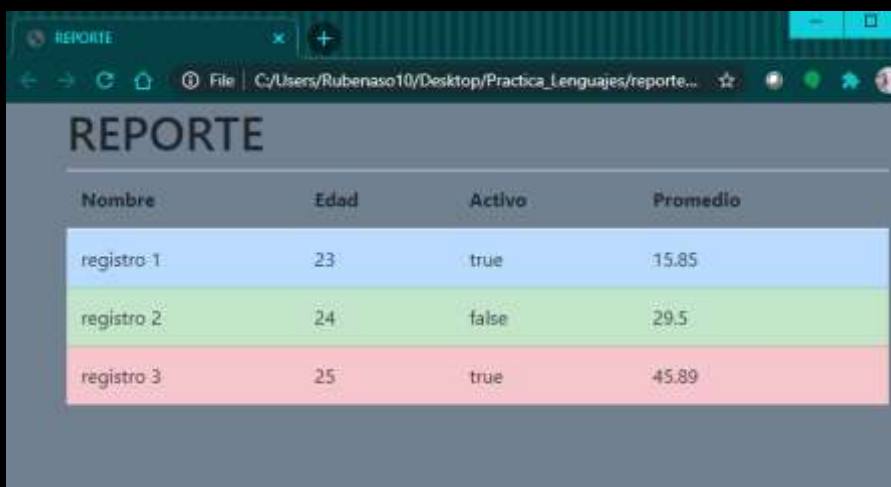
    except:
        print("ERROR SE ESPERABA UN VALOR NUMERICO ")

```


Ejemplo de archivo a cargar:

```
1  [
2      {
3          "nombre" : "registro 1 ",
4          "activo" : "true",
5          "edad": 23,
6          "promedio": 15.85
7      },
8
9
10     ],
11
12     {
13         "nombre" : "registro 2 ",
14         "activo" : "false",
15         "edad": 24,
16         "promedio": 29.50
17     },
18
19     },
20
21     {
22         "nombre" : "registro 3 ",
23         "activo" : "true",
24         "edad": 25,
25         "promedio": 45.89
26     },
27
28     },
29
30 ]
31
32
33
34
35
```

Ejemplo de mostrar archivo en el navegador:



The screenshot shows a web browser window with a single tab titled 'REPORTE'. The address bar shows the file path 'C:/Users/Rubenaso10/Desktop/Practica_Lenguajes/reporte...'. The main content area displays a table with the following data:

Nombre	Edad	Activo	Promedio
registro 1	23	true	15.85
registro 2	24	false	29.5
registro 3	25	true	45.89