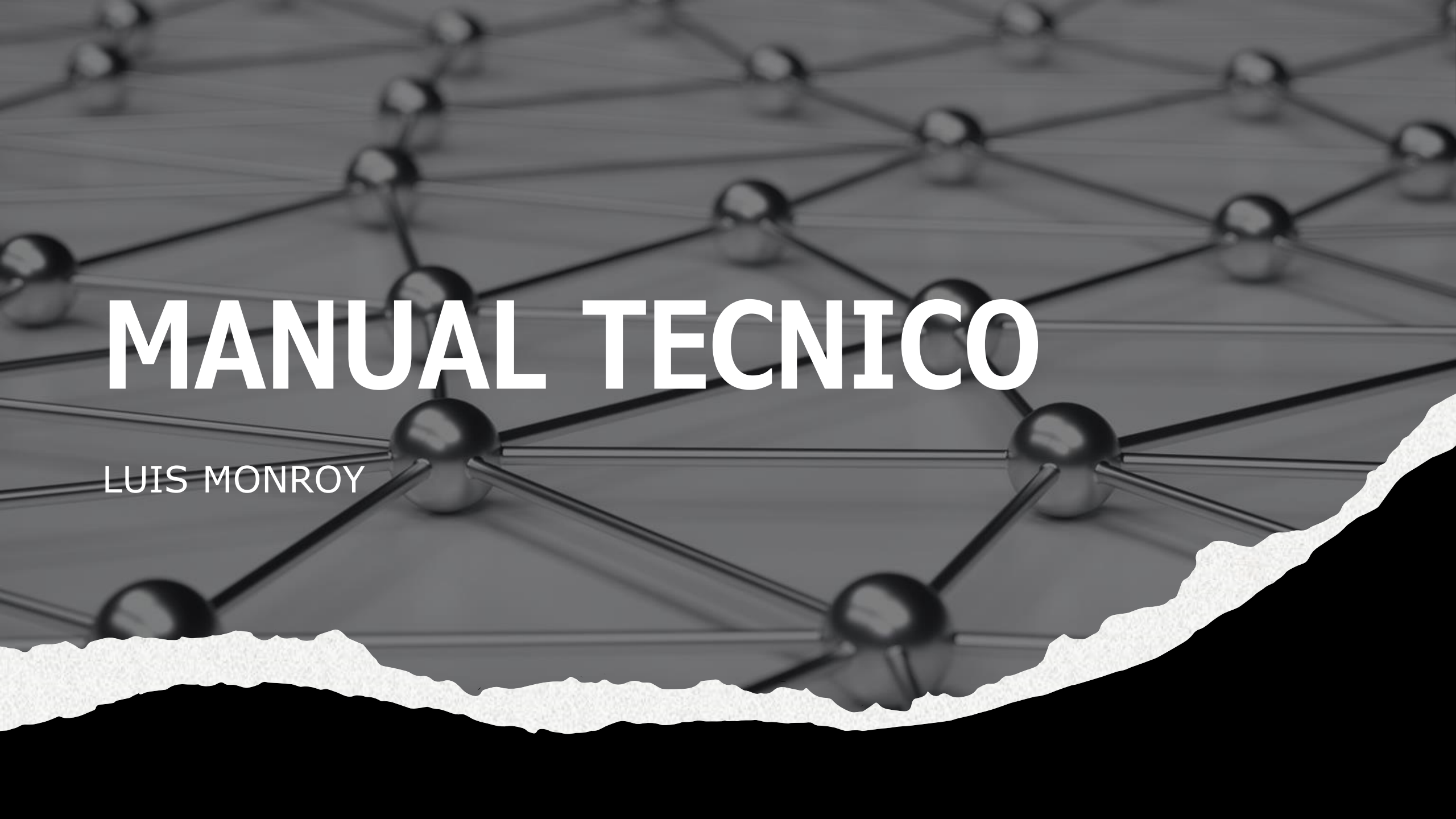

MANUAL TECNICO

LUIS MONROY

CLASS MENU INICIAL()

```
class Menuinicial():
    def __init__(self):...
    def opciones(self):...
    def Seleccion(self,lista,valor):...
    def Todas(self,lista):...
    def Todas_Archivo(self,lista):...
```

```
def opciones(self):
    Funcion = Funciones()
    lista = []
    bandera = True
    while bandera:
        print("Elija una opcion:\n"
              "1. CARGAR ARCHIVO DE ENTRADA\n"
              "2. DESPLEGAR LISTAS ORDENADAS\n"
              "3. DESPLEGAR BUSQUEDAS\n"
              "4. DESPLEGAR TODAS\n"
              "5. DESPLEGAR TODAS A ARCHIVO\n"
              "6. SALIR"
              )
        opcion = int(input("@>> "))
        if opcion == 1:
            ventana = Tk()
            ventana.withdraw()
            ventana.update()
            path = askopenfilename()
            if path:
                lista = Funcion.open(path)
        if opcion == 2:
            self.Seleccion(lista,"O")
        if opcion == 3:
            self.Seleccion(lista,"B")
```

METODO OPCIONES()

- Este metodo se encarga de desplegar el menú inicial y obtener la opción que se desea realizar si en caso no se elige una opción correcta se volverá a pedir la opción, tambien es el encargado de interactuar con los demás métodos, la petición de cada opción se genera por medio de un while

METODO SELECCIÓN()

```
def Seleccion(self,lista,valor):
    for i in lista:
        if i[0] == valor:
            print(i[1:])
```

- Es un metodo auxiliar que permite imprimir los resultados según se requiera, su principal funcion es la obtención de una cadena almacenada en un vector y se muestra el resultado si el valor es O se muestran las listas ordenas y si el valor es B se muestran las listas en las cuales se busco un elemento

METODO TODAS()

- Obtiene todas las listas almacenadas y sus respectivas funciones según el formato establecido.

```
def Todas(self,lista):
    for i in lista:
        print(i[1:])
```

METODO TODAS A ARCHIVO()

- Este metodo se encarga de generar un archivo json y abre una ventana en el navegador, por medio de la funcion webbrowser

CLASS FUNCIONES()

```
def open(self,archivo):
    archivo = open(archivo,"r")
    listado = []
    for lineas in archivo.readlines():
        self.cadena(self.espacios(lineas),listado)
    archivo.close()
    return listado
                string,listado):...
```

METODO OPEN()

- Este es el metodo encargado de abrir el archivo y leer línea por línea, retorna una lista cargada con todas las funciones en la estructura indicada

METODO CADENA()

- Es el metodo encargado de descomponer la cadena proveniente de open e interpretar las instrucciones asi como de dar formato según la operación indicada, utiliza appen() para cargar cada solución a una lista referenciada

METODO ESPACIOS()

```
def espacios(self,listado):
    string = ""
    for i in range(len(listado)):
        if listado[i] != " ":
            string += listado[i]
    return string
```

Este es un metodo complementario que elimina los espacios en todas las líneas del archivo

METODO ORDENAR()

- Este metodo se encarga de realizar un ordenamiento tipo burbuja y retorna una lista de caracteres

```
def ordenar(self,listado):
    for i in range(len(listado)-1):
        for j in range(i+1,len(listado)):
            if listado[i] > listado[j]:
                temp = listado[i]
                listado[i] = listado[j]
                listado[j] = temp

    return self.iteracion2(listado)
```

METODO BUSCAR()

```
def buscar(self,listado1,listado2):
    encontrado = ""
    numeros = []
    for i in listado1:
        for j in range(len(listado2)):
            if listado2[j].isdigit:
                if listado2[j]== i:
                    numeros.append(str(j+1))
    encontrado = ",".join(numeros)
    if encontrado == "":
        encontrado = "NO ENCONTRADO"
    return encontrado
```

Este metodo se encarga de buscar los elementos que se piden y devuelve los elementos si se encontraron, de lo contrario devuele NO ENCONTRADO

METODO ITERACION() E ITERACION2()

```python
def iteracion(self,cadena):
    for i in range(len(cadena)):
        cadena[i] = int(cadena[i])
    return cadena
def iteracion2(self,numeros):
    for i in range(len(numeros)):
        numeros[i] = str(numeros[i])
    return numeros
```

Estos métodos son complementarios y su única funcion es transformar una lista de numeros a caracteres o viceversa.

Paradigmas utilizados

- POO (Orientado a objetos)
- Funcional

Orientado a objetos

- Aunque no se implementan mayores características de la programación orientada a objetos se utilizan clases y métodos como la clase Funciones y la clase Menu cada una posee sus atributos y sus métodos.

Funcional

El Proyecto se Desarrollo practicamente por medio de funciones que permiten la solucion de los problemas presentados.