UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENEIRIA EN CIENCIAS Y SISTEMAS

LENGUAJES FORMALES Y DE PROGRAMACIÓN

SECCION “B+”

PROYECTO 1

MANUAL DE TECNICO

JUAN PABLO GONZALEZ LEAL

201901374

21/03/2021

Main

```
def menu():
    print('Lenguajes Formales de Programacion seccion')
    print('Juan Pablo Gonzalez Leal')
    print('Carne: 201901374')
    while True:
        print('1. Cargar menu')
        print('2. Cargar o rden')
        print('3. Generar menu')
        print('4. Generar factura')
        print('5. Generar árbol')
        print('6. Salir')
        opcion=int(input('ELIJA UNA OPCION'))
        if opcion==1:
            cargar_menu()
            titulo.clear()
            seccion.clear()
            datos.clear()
            total_datos.clear()
            descripcionProducto.clear()
            cantidad.clear()
            precioProducto.clear()
            nombreProducto.clear()
            cantidadProductos=0

            tokens.clear()
            errores.clear()
            identificador_id.clear()
            columnas.clear()
            filas.clear()
            Total_tokens.clear()
            lexema.clear()
            No.clear()
            contador_filas=0
            cantidad_tokens=0
            cantidad_errores=0

        elif opcion ==2:
            cargar_orden()
            nombre_cliente.clear()
            nit_cliente.clear()
            direccion_cliente.clear()
            propina_cliente.clear()
            propina_clienteInt.clear()
                cantidad_comprada.clear()
                identificador_comprado.clear()

                No_factura.clear()
                Lexema_factura.clear()
                fila_factura.clear()
                columna_factura.clear()
                tokensfactura.clear()
                cantidadtokens_factura=0
                contadorfilas_factura=0

            elif opcion==3:
                generar_menu()

            elif opcion==4:
                generar_factura()

            elif opcion==5:
                generar_arbol()

            elif opcion==6:
                    break
```

Funciones:
Menú(): Este es un mensaje que se muestra en consola para poder seleccionar una opción.

Cargar_menu():Esta función llama una ventana para poder recoger la ruta del archivo menu que deseamos analizar.

Caragar_orden():Esta función llama una ventana para poder recoger la ruta del archivo orden que deseamos analizar.

Generar_menu():Pasamos la ruta de que obtuvimos en cargar_menu() y este hace que lee cada linea se pasen al archivo analizador, hasta finalizar que todo este correctamente.

Generar_factura():Pasamos la ruta de que obtuvimos en cargar_orden() y este hace que lee cada línea se pasen al archivo analizador_orden, hasta finalizar que todo este correctamente.

Validacion_Tokens():Pasamos la ruta de que obtuvimos en analizar_token() y este hace que lee cada línea se pasen al archivo analizador_orden, hasta finalizar que todo este correctamente con los tokens encontrados.

Validacion_Tokens_Factura():Pasamos la ruta de que obtuvimos en token_factura() y este hace que lee cada línea se pasen al archivo analizador_orden, hasta finalizar que todo este correctamente con los tokens encontrados.

Generar_arbol():
Generamos un grafico de las

Variables globales
Factura=0
numero=0

numero2=0
graphviz=0
factura_lista=[0]
ruta1=""
ruta2=""

Analizador

```
    def analizar(cadena,contador):
        if contador==0:
            Titulo(cadena)

        inicial=cadena[0]
        if inicial=="'":
            global cantidadProductos
            cantidadProductos += 1
            Nombre_seccion(cadena)

        if inicial=="[":
            Opcion_menu(cadena)
            cantidad.append(cantidadProductos)
```

```
def Titulo(cadena):
    contador=0
    paso1=True
    paso2=False
    cajon=""
    cajon2=""
    error1=""
    for i in cadena:
        caracter=i
        if paso1==True:
            if caracter=="=":
                if cajon=="restaurante" or cajon=="RESTAURANTE" or cajon=="Restaurante":
                    paso1=False
                    paso2=True
                    cajon="Nombre restaurante = "
                else:
                    error1=("No existepalabra restaurante")
            else:
                cajon=cajon+caracter
        elif paso2==True:
            if caracter=="'":
                contador +=1
                if contador==2:
                    pass
                else:
                    error1=error1+"no contiene comillas"
            else:
                cajon2=cajon2+caracter
    titulo.append(cajon2)
```

```
    def Nombre_seccion(cadena):
        contador=0
        cajon=""
        cajon2=""
        paso1=True
        paso2=False
        error2=""
        for i in cadena:
            caracter=i
            if paso1==True:
                if caracter=="'":
                    contador += 1
                    if contador==2:
                        paso1=False
                        paso2=True
                else:
                    cajon=cajon+caracter
            elif paso2==True:
                if caracter==":" :
                    cajon2="Nombre Seccion = "
                else:
                    error2="No existe el caracter :"
        seccion.append(cajon)
```

Variables globales
titulo=[]
seccion=[]
datos=[]
total_datos=[]
descripcionProducto=[]
cantidad=[]
precioProducto=[]
nombreProducto=[]
cantidadProductos=0

analizar(cadena,contador):
el contador sirve para leer en la fila que estamos analizando, y la cadena para saber cómo inicia la cadena e ingresarla en una condición y pasara la cadena a un método y que realice su respectivo análisis.

Titulo(cadena):
Analizamos la primera línea de entrada donde se encontrará el título del restaurante

Nombre_seccion(cadena)
Analizamos la sección de los productos que ingresaran luego por el identificador

```
def Opcion_menu(cadena):
    contador=0
    contador2=0
    paso1=False
    paso2=False
    paso3=False
    paso4=False
    paso5=False
    ultimo=""
    id=""
    nombre=""
    precio=""
    descripcion=""
    for i in cadena:
        caracter=i
        if caracter=="[":
            contador += 1
            paso1=True
        elif paso1==True:
            if caracter==";":
                paso1=False
                paso2=True
            else:
                id=id+caracter
        elif paso2==True:
            if caracter==";":
                paso2=False
                paso3=True
            else:
                if caracter=="'":
                    pass
                else:
                    nombre=nombre+caracter
```

Opción_menu(cadena):
Analizamos atreves de la condición que se hace al inicio sabemos que este es un producto a ingresar, este lo dividimos en pasos para ir ingresando de forma ordenada el identificador, nombre del producto y la descripción del mismo

Tener en cuenta que al finalizar de analizar cada línea se agrega a una lista para poder generar los html

Analizador_orden

```
def analizar_orden(cadena,contador):
    if contador==0:
        DatosPersonales(cadena)
    else:
        Pedidos(cadena)

def DatosPersonales(cadena):
    paso1=True
    contador=0
    contador2=0
    nombre=""
    nit=""
    direccion=""
    porcentaje=""
    porcentaje_int=0
    for i in cadena:
        caracter=i
        if paso1==True:
            if caracter=="'":
                contador += 1
            elif caracter==",":
                contador2 += 1
            else:
                if contador==1:
                    if caracter==" ":
                        pass
                    else:
                        nombre=nombre+caracter
                elif contador==3:
                    if caracter==" ":
                        pass
                    else:
                        nit=nit+caracter
                elif contador==5:
                    if caracter==" ":
                        pass
                    else:
                        direccion=direccion+caracter
                elif contador2==3:
                    if caracter=="%":
                        porcentaje_int=int(porcentaje)
                    else:
                        if caracter==" ":
```

Variables Globales
nombre_cliente=[]
nit_cliente=[]
direccion_cliente=[]
propina_cliente=[]
propina_clienteInt=[]
cantidad_comprada=[]
identificador_comprado=[]

analizar_orden(cadena,contador):
Pasamos la cadena a un método por medio de condiciones

DatosPesonales(cadena):
Realizamos varias condiciones para obtener datos por separado y almacenarlos en de forma ordenada.

```
                            pass
                        else:
                            porcentaje=porcentaje+caracter
        nombre_cliente.append(nombre)
        nit_cliente.append(nit)
        direccion_cliente.append(direccion)
        propina_cliente.append(porcentaje)
        propina_clienteInt.append(porcentaje_int)


    def Pedidos(cadena):
        global posicion_contador
        paso1=True
        paso2=False
        cantidad=""
        identificador=""
        for i in cadena:
            caracter=i
            if paso1==True:
                if caracter == ",":
                    cantidad_int=int(cantidad)
                    paso1=False
                    paso2=True
                else:
                    if caracter==" ":
                        pass
                    else:
                        cantidad=cantidad+caracter
            elif paso2==True:
                if caracter==" ":
                    pass
                else:
                    identificador=identificador+caracter

        cantidad_comprada.append(cantidad_int)
        identificador_comprado.append(identificador)
```

Pedidos(cadena)
En este apartado analizamos la cantidad comprada y por medio del identificador sabremos que producto se compró.

Al finalizar cada línea se agregan en listas

Analizar_token

Este se analiza de primero antes de realizar anlizador, ya que este verifica que todos los datos estén correctos haciendo mas validaciones para obtener una tabla de todos los datos que se ingresaron.

Token_factura

Este se analiza de primero antes de realizar anlizador_orden, ya que este verifica que todos los datos estén correctos haciendo mas validaciones para obtener una tabla de todos los datos que se ingresaron.

HTML GENERADOS DESDE PYTHON

html_menu

```
from analizador import *
def crear():
    crear=open(titulo[0].replace("\n","")+".html","w")
    crear.write("<html lang=\"es\">")
    crear.write("<head>")
    crear.write("<meta charset=\"ISO 8859-1\">")
    crear.write("<meta name=\"viewport\" content=\"width=, initial-scale=1.0\">")
    crear.write("<link rel=\"stylesheet\" href=\"https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css\" integrity=\"sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9
    crear.write("<title>"+titulo[0]+"</title>")
    crear.write("</head>")
    crear.write("<style>body {background-color: #FFCA33;}</style>")
    crear.write("<body>")
    crear.write("<div class=\"container\"><div class=\"row justify-content-center\"><div class=\"col-4\"><h1>"+titulo[0]+"</h1></div></div></div>")
    x=len(datos)
    y=1
    for i in seccion:
        crear.write("<div class=\"row\"><div class=\"col align-self-start\"><h3>"+i+"</h3></div></div><br>")
        for i in range(0,x):
            if cantidad[i]==y:
                crear.write("<div class=\"row\"><div class=\"col align-self-start\"><h3>"+datos[i]+"</h3></div></div>")
                crear.write("<div class=\"row\"><div class=\"col-md-9 offset-md-3\"><h4>"+descripcionProducto[i]+"</h4></div></div>")
        crear.write("<br>")
        crear.write("<hr>")
        y += 1
    crear.write("</div>")
    crear.write("<script src=\"https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js\" integrity=\"sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6\" crossorigin=\
    crear.write("<script src=\"https://code.jquery.com/jquery-3.4.1.slim.min.js\" integrity=\"sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n\" crossorigin=\"anonymous\"></script>
    crear.write("<script src=\"https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js\" integrity=\"sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo\" crossorigin=\"a
    crear.write("</body>")
    crear.write("</html>")
    crear.close()
    print('archivo creado')
```

html_factura

```
from analizador_orden import*
from analizador import*
from analizar_token import*
import datetime
ahora=datetime.datetime.now()

def crear(Factura):
    crear=open("Factura#"+str(Factura)+".html","w")
    crear.write("<html lang=\"es\">")
    crear.write("<html lang=\"es\" dir=\"ltr\"><head><meta charset=\"ISO-8859-1\"><link href=\"https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css\" rel=\"stylesheet\" integri
    crear.write("<title>Factura#"+str(Factura)+"</title></head><style>body{background:#8ba987 url('https://cronica.com.gt/wp-content/uploads/2019/09/1.jpg') no-repeat center center;background-size:100
    div></h3>")
    crear.write("<h5 class=\"card-title\"><div class=\"row justify-content-center\">Factura No."+str(Factura)+"</div></h5>")
    crear.write("<h5 class=\"card-title\"><div class=\"row justify-content-center\">Fecha: "+str(ahora)+"</div></h5>")
    crear.write("<h5 class=\"card-subtitle mb-2 text-muted\">Datos Cliente</h5><h5 class=\"card-subtitle mb-2 text-muted\">Nombre: "+nombre_cliente[0]+"</h5><h5 class=\"card-subtitle mb-2 text-muted\"
    crear.write("<h5 class=\"card-subtitle mb-2 text-muted\"> Direccion: "+direccion_cliente[0]+"</h5><br><h5 class=\"card-subtitle mb-2 text-muted\">Descripcion</h5><table class=\"table\"><thead><tr>"
    crear.write("<th scope=\"col\">Cantidad</th><th scope=\"col\">Concepto</th><th scope=\"col\">Precio</th><th scope=\"col\">Total</th></tr></thead><tbody>")
    SUBTOTAL=0
    TOTAL=0
    x=len(identificador_id)
    y=len(cantidad_comprada)
    for i in range(0,y):
        crear.write("<tr><th scope=\"row\">"+str(cantidad_comprada[i])+"</th>")
        j=identificador_id.index(identificador_comprado[i].replace("\n",""))
        crear.write("<td>"+nombreProducto[j]+"</td>")
        crear.write("<td>"+str(precioProducto[j])+"</td>")
        TotalComprado=(precioProducto[j]*cantidad_comprada[i])
        crear.write("<td>"+str(TotalComprado)+"</td></tr>")
        SUBTOTAL=SUBTOTAL+TotalComprado

    crear.write("</tbody><thead><tr><td scope=\"row\" colspan=\"3\">Sub total</td>")
    crear.write("<td scope=\"row\" colspan=\"2\">Q "+str(SUBTOTAL)+"</td>")
    crear.write("<tr><td scope=\"row\" colspan=\"3\">Propina("+propina_cliente[0]+"%)</td>")
    propina=(propina_clienteInt[0]/100)*SUBTOTAL
    crear.write("<td scope=\"row\" colspan=\"1\">Q "+str(propina)+"</td></tr></thead>")
    TOTAL=propina+SUBTOTAL
    crear.write("<thead><tr><th scope=\"col\" colspan=\"3\">Total</th>")
    crear.write("<th scope=\"col\" colspan=\"1\">"+str(TOTAL)+"</th></tr></thead>")
    crear.write("</table></div></div></div></div></body></html>")
    crear.close()
    print('archivo creado')
```

html_tokens

```
from analizar_token import*
def crear(Tokens):
    crear=open("TokensMenu#"+str(Tokens)+".html","w")
    crear.write("<html lang=\"es\">")
    crear.write("<head><meta charset=\"ISO-8859-1\"><link href=\"https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css\" rel=\"stylesheet\" integrity=\"sha384-BmbxuPwQa2lc/FVzBc
    crear.write("<div class=\"container\"><div class=\"row justify-content-center\"><div class=\"col-4\"><h1>Tabla Tokens# "+str(Tokens)+"</h1></div></div>")
    crear.write("<table class=\"table\"><thead><tr class=\"table-dark\"><th scope=\"col\">No</th><th scope=\"col\">Fila</th><th scope=\"col\">Columna</th><th scope=\"col\">Carácter</th><th scope=\"col\
    x=len(No)
    for i in range(0,x):
        crear.write("<tr><th scope=\"row\">"+No[i]+"</th><td>"+lexema[i]+"</td><td>"+filas[i]+"</td><td>"+columnas[i]+"</td><td>"+Total_tokens[i]+"</td></tr>")

    crear.write("</table></div></body></html>")
    crear.close()
    print('archivo creado')
```

html_token_factura

```
from token_factura import*
def crear(Tokens):
    crear=open("TokensFactura#"+str(Tokens)+".html","w")
    crear.write("<html lang=\"es\">")
    crear.write("<head><meta charset=\"ISO-8859-1\"><link href=\"https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css\" rel=\"stylesheet\" integrity=\"sha384-BmbxuPwQa2lc/FVzBc
    crear.write("<div class=\"container\"><div class=\"row justify-content-center\"><div class=\"col-4\"><h1>Tabla Tokens# "+str(Tokens)+"</h1></div></div>")
    crear.write("<table class=\"table\"><thead><tr class=\"table-dark\"><th scope=\"col\">No</th><th scope=\"col\">Fila</th><th scope=\"col\">Columna</th><th scope=\"col\">Carácter</th><th scope=\"col\
    x=len(No_factura)
    for i in range(0,x):
        crear.write("<tr><th scope=\"row\">"+No_factura[i]+"</th><td>"+Lexema_factura[i]+"</td><td>"+fila_factura[i]+"</td><td>"+columna_factura[i]+"</td><td>"+tokensfactura[i]+"</td></tr>")
    crear.write("</table></div></body></html>")
    crear.close()
    print('archivo creado')
```

Paradigmas utilizados

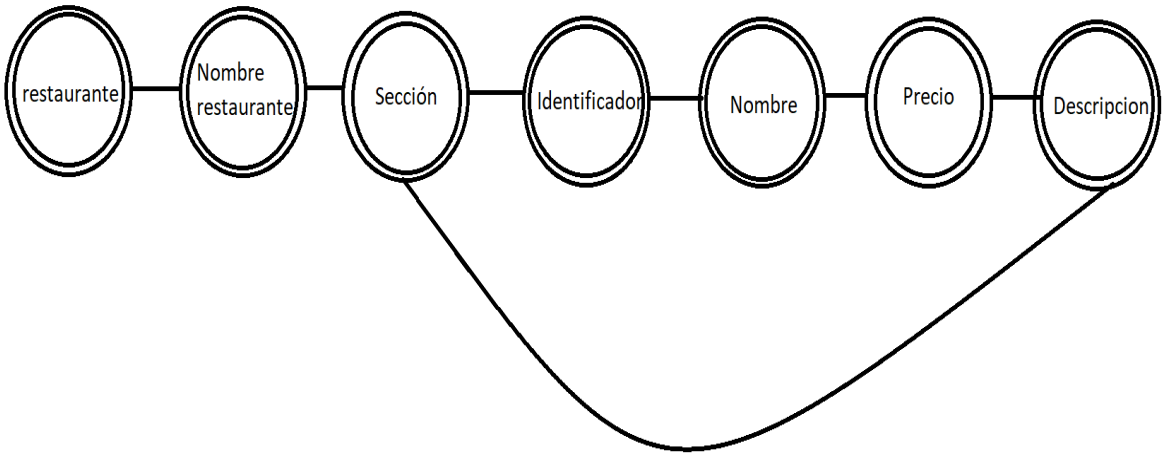
Lógico: En la realización de métodos y condiciones al utilizar if .

Imperativo: Al inicializar listas vacías y variables tanto de tipo Sting como boléanos.

Dirigido por eventos: Al hacer una entrada de datos por el usuario tanto al escribir un numero a la opción del menú, como a la hora de escribir el nombre del archivo que se creara como HTML.

Orientado a aspectos: El programa o software se divide en módulos o archivos de distinto nombre realizando cada archivo actividades diferentes

AFD MENU



AFD ORDEN