

Universidad De San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas



Lenguajes Formales y de Programación

Sección -B

MANUAL TÉCNICO

Kevin Gerardo Ruíz Yupe
201903791
23/09/2021

Objetivos

General:

Enfatizar en el uso de clases, variables y métodos escritos en la aplicación para usuarios con conocimiento específico en el tema. Asimismo, para edición de código en futuras actualizaciones.

Específicos:

- Mostrar el IDE utilizado para el desarrollo del programa
- Describir el proceso de solución, así como las herramientas utilizadas durante el proyecto.

Introducción

Este manual pretende explicar y describir los temas que se desarrollan durante el proceso de creación del programa. De la misma forma, se indica los requerimientos del sistema, el IDE utilizado y las librerías que se importaron.

La principal función de esta aplicación es la obtención de datos a partir de un archivo con extensión .pxla, recorriendo cada carácter utilizando ciclos y condicionales.

Se usa también librerías externas para mostrar las imágenes.

Descripción de la solución

La aplicación es una Clase de nombre ***Application***, donde ***constructor*** y el ***método create_widget*** genera la interfaz gráfica.

```
class Application():  
  
    def __init__(self):
```

```
def __init__(self):  
    self.root = Tk()  
    self.root.title('Bitxelart')  
    self.root.iconbitmap('Images/Icono.ico')  
    self.root.config(bg='#e5e7e9')  
    self.root.state('zoomed')  
  
def create_widgets(self):  
    Menu=ttk.Notebook(self.root)  
    s =ttk.Style()  
    s.theme_use(['default'])  
    s.configure('TNotebook', tabposition='n',background='#17202a')  
    s.configure('TNotebook.Tab', font=('Segoe UI', 12,'bold'),background='#3498db', foreground='white',  
    s.map('TNotebook.Tab', background=[('selected', '#2471a3'),('active', "#3498db")], foreground=[("sel  
#s.theme_use("default")
```

Aquí se hace una instancia de NoteBook. Este widget permite navegar entre pestañas como un menú en sitios web.

```
Menu=ttk.Notebook(self.root)  
s =ttk.Style()  
s.theme_use(['default'])  
s.configure('TNotebook', tabposition='n',background='#17202a')  
s.configure('TNotebook.Tab', font=('Segoe UI', 12,'bold'),background='#3498db', foreground='white', borderwidth=0)  
s.map('TNotebook.Tab', background=[('selected', '#2471a3'),('active', "#3498db")], foreground=[("selected", 'white')])  
#s.theme_use("default")
```

En esta clase se crea la parte gráfica del apartado Cargar Archivo. De igual forma, se crea el método general de Análisis: la función es recorrer el archivo de entrada carácter por carácter.

```
from os import error, linesep,mkdir  
import textwrap  
from tkinter import *  
from tkinter import filedialog  
from tkinter import messagebox  
from Error import Error  
from html2image import Html2Image  
from Token import Token  
from VerImagen import ListaTitulos  
class CargarArchivo():
```

```

        err=Error('Sintactico',fila,columna,'Se esperaba una T',c)
        self.ListaErrores.append(err)
    elif estado==3:
        if self.isLetra(c) and ord(c)==85:
            lexActual=lexActual+c
            tokenPalabraR=tokenPalabraR+c
            estado=4
        else:
            if ord(c) == 32 or ord(c) == 10 or ord(c) == 9 or c == '~':
                pass
            elif self.isSimbolo(c) or ord(c)!=85:
                err=Error('Sintactico',fila,columna,'Se esperaba una U',c)
                self.ListaErrores.append(err)
    elif estado==4:
        if self.isLetra(c) and ord(c)==76:
            lexActual=lexActual+c
            tokenPalabraR=tokenPalabraR+c
            estado=5
        else:
            if ord(c) == 32 or ord(c) == 10 or ord(c) == 9 or c == '~':
                pass
            elif self.isSimbolo(c) or ord(c)!=76:
                err=Error('Sintactico',fila,columna,'Se esperaba una V',c)
                self.ListaErrores.append(err)

self.BooleanosAux=[]
self.Posiciones=[]
self.PosicionesTotales=[]
self.PosicionesAux=[]
self.PosicionesAux2=[]

```

Estos métodos verifican que el carácter de entrada se encuentra dentro de los rangos permitidos, si existe, el método retorna un valor Verdadero.

```

#Letra
def isLetra(self,caracter):
    if((ord(caracter) >= 65 and ord(caracter) <= 90) or (ord(caracter) >= 97 and ord(caracter) <= 122) or ord(caracter) == 164 or ord(caracter) == 165):
        return True
    else:
        return False
#Numero
def isNumero(self,caracter):
    if ((ord(caracter) >= 48 and ord(caracter) <= 57)):
        return True
    else:
        return False
#Simbolo
def isSimbolo(self,caracter):
    if (ord(caracter)==35 or ord(caracter)==34 or ord(caracter)==44 or ord(caracter)==59 or ord(caracter)==61 or ord(caracter)==64 or ord(caracter)==65 or ord(caracter)==66 or ord(caracter)==67 or ord(caracter)==68 or ord(caracter)==69 or ord(caracter)==70 or ord(caracter)==71 or ord(caracter)==72 or ord(caracter)==73 or ord(caracter)==74 or ord(caracter)==75 or ord(caracter)==76 or ord(caracter)==77 or ord(caracter)==78 or ord(caracter)==79 or ord(caracter)==80 or ord(caracter)==81 or ord(caracter)==82 or ord(caracter)==83 or ord(caracter)==84 or ord(caracter)==85 or ord(caracter)==86 or ord(caracter)==87 or ord(caracter)==88 or ord(caracter)==89 or ord(caracter)==90 or ord(caracter)==91 or ord(caracter)==92 or ord(caracter)==93 or ord(caracter)==94 or ord(caracter)==95 or ord(caracter)==96 or ord(caracter)==97 or ord(caracter)==98 or ord(caracter)==99 or ord(caracter)==100 or ord(caracter)==101 or ord(caracter)==102 or ord(caracter)==103 or ord(caracter)==104 or ord(caracter)==105 or ord(caracter)==106 or ord(caracter)==107 or ord(caracter)==108 or ord(caracter)==109 or ord(caracter)==110 or ord(caracter)==111 or ord(caracter)==112 or ord(caracter)==113 or ord(caracter)==114 or ord(caracter)==115 or ord(caracter)==116 or ord(caracter)==117 or ord(caracter)==118 or ord(caracter)==119 or ord(caracter)==120 or ord(caracter)==121 or ord(caracter)==122 or ord(caracter)==123 or ord(caracter)==124 or ord(caracter)==125 or ord(caracter)==126 or ord(caracter)==127 or ord(caracter)==128 or ord(caracter)==129 or ord(caracter)==130 or ord(caracter)==131 or ord(caracter)==132 or ord(caracter)==133 or ord(caracter)==134 or ord(caracter)==135 or ord(caracter)==136 or ord(caracter)==137 or ord(caracter)==138 or ord(caracter)==139 or ord(caracter)==140 or ord(caracter)==141 or ord(caracter)==142 or ord(caracter)==143 or ord(caracter)==144 or ord(caracter)==145 or ord(caracter)==146 or ord(caracter)==147 or ord(caracter)==148 or ord(caracter)==149 or ord(caracter)==150 or ord(caracter)==151 or ord(caracter)==152 or ord(caracter)==153 or ord(caracter)==154 or ord(caracter)==155 or ord(caracter)==156 or ord(caracter)==157 or ord(caracter)==158 or ord(caracter)==159 or ord(caracter)==160 or ord(caracter)==161 or ord(caracter)==162 or ord(caracter)==163 or ord(caracter)==164 or ord(caracter)==165 or ord(caracter)==166 or ord(caracter)==167 or ord(caracter)==168 or ord(caracter)==169 or ord(caracter)==170 or ord(caracter)==171 or ord(caracter)==172 or ord(caracter)==173 or ord(caracter)==174 or ord(caracter)==175 or ord(caracter)==176 or ord(caracter)==177 or ord(caracter)==178 or ord(caracter)==179 or ord(caracter)==180 or ord(caracter)==181 or ord(caracter)==182 or ord(caracter)==183 or ord(caracter)==184 or ord(caracter)==185 or ord(caracter)==186 or ord(caracter)==187 or ord(caracter)==188 or ord(caracter)==189 or ord(caracter)==190 or ord(caracter)==191 or ord(caracter)==192 or ord(caracter)==193 or ord(caracter)==194 or ord(caracter)==195 or ord(caracter)==196 or ord(caracter)==197 or ord(caracter)==198 or ord(caracter)==199 or ord(caracter)==200 or ord(caracter)==201 or ord(caracter)==202 or ord(caracter)==203 or ord(caracter)==204 or ord(caracter)==205 or ord(caracter)==206 or ord(caracter)==207 or ord(caracter)==208 or ord(caracter)==209 or ord(caracter)==210 or ord(caracter)==211 or ord(caracter)==212 or ord(caracter)==213 or ord(caracter)==214 or ord(caracter)==215 or ord(caracter)==216 or ord(caracter)==217 or ord(caracter)==218 or ord(caracter)==219 or ord(caracter)==220 or ord(caracter)==221 or ord(caracter)==222 or ord(caracter)==223 or ord(caracter)==224 or ord(caracter)==225 or ord(caracter)==226 or ord(caracter)==227 or ord(caracter)==228 or ord(caracter)==229 or ord(caracter)==230 or ord(caracter)==231 or ord(caracter)==232 or ord(caracter)==233 or ord(caracter)==234 or ord(caracter)==235 or ord(caracter)==236 or ord(caracter)==237 or ord(caracter)==238 or ord(caracter)==239 or ord(caracter)==240 or ord(caracter)==241 or ord(caracter)==242 or ord(caracter)==243 or ord(caracter)==244 or ord(caracter)==245 or ord(caracter)==246 or ord(caracter)==247 or ord(caracter)==248 or ord(caracter)==249 or ord(caracter)==250 or ord(caracter)==251 or ord(caracter)==252 or ord(caracter)==253 or ord(caracter)==254 or ord(caracter)==255):
        return True
    else:
        return False
#Hexadecimal
def isHexadecimal(self,caracter):
    if ((ord(caracter) >= 65 and ord(caracter) <= 70) or (ord(caracter) >= 97 and ord(caracter) <= 102) or (ord(caracter) >= 48 and ord(caracter) <= 57)):
        return True
    else:
        return False

```

En el método Analizar se crean diferentes condiciones para el archivo de entrada, de este modo se lleva un control con los estados. Cada estado sigue el proceso del autómata finito determinista mostrado posteriormente.

```

        err=Error('Sintactico',fila,columna,'Se esperaba una T',c)
        self.ListaErrores.append(err)
    elif estado==3:
        if self.isLetra(c) and ord(c)==85:
            lexActual=lexActual+c
            tokenPalabraR=tokenPalabraR+c
            estado=4
        else:
            if ord(c) == 32 or ord(c) == 10 or ord(c) == 9 or c == '~':
                pass
            elif self.isSimbolo(c) or ord(c)!=85:
                err=Error('Sintactico',fila,columna,'Se esperaba una U',c)
                self.ListaErrores.append(err)
    elif estado==4:
        if self.isLetra(c) and ord(c)==76:
            lexActual=lexActual+c
            tokenPalabraR=tokenPalabraR+c
            estado=5
        else:
            if ord(c) == 32 or ord(c) == 10 or ord(c) == 9 or c == '~':

```

Este método genera los archivos .html con las celdas coloreadas para cada imagen. Crea los directorios para guardar cada archivo así como su respectiva imagen .png

```

def Generarimagenes(self):
    self.Pos=[]
    for a in range(len(self.Filas)):
        for f in range(self.Filas[a]):
            for c in range(self.Columnas[a]):
                nuevoPos=[c,f]
                self.Pos.append([nuevoPos])

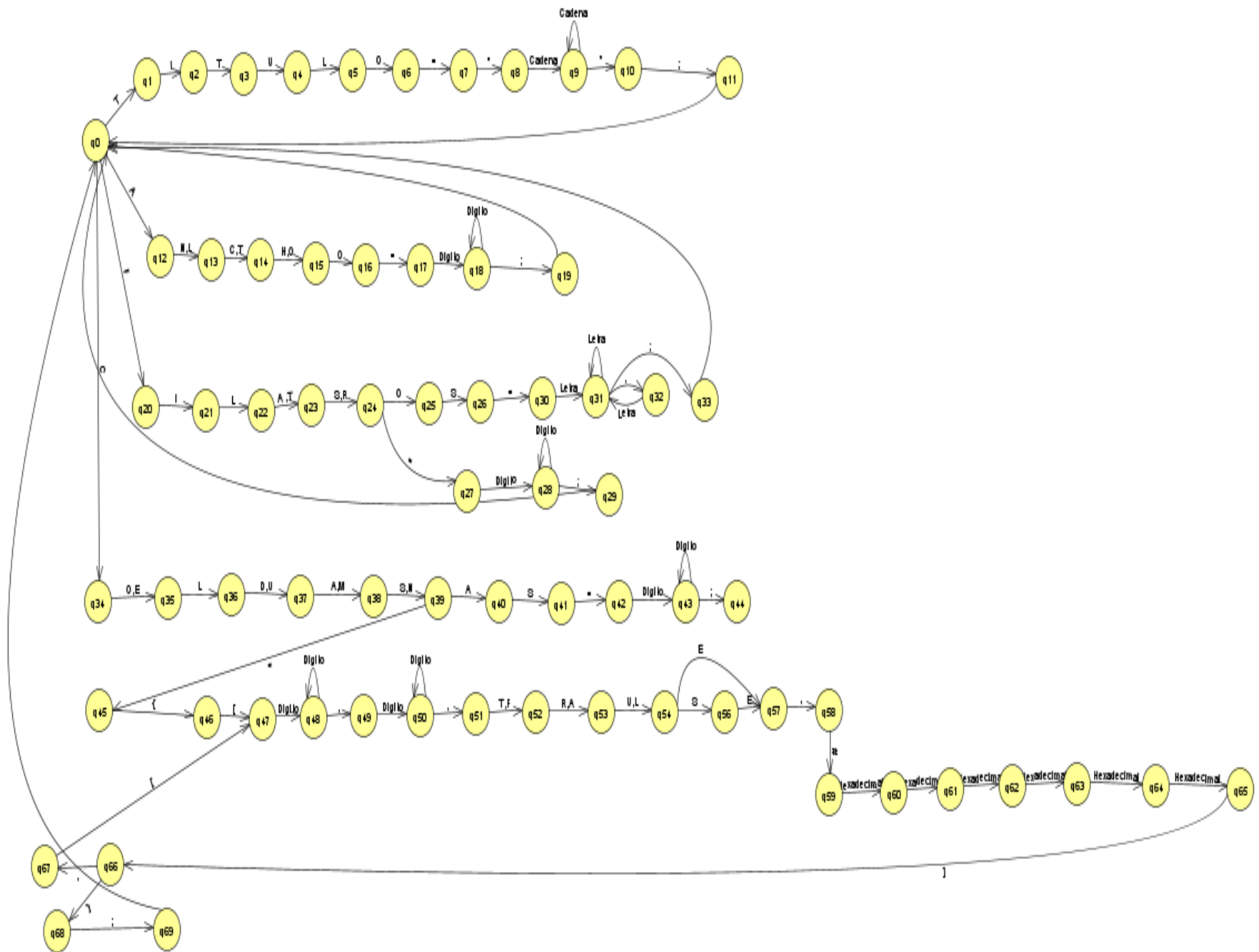
    pos1=self.Pos.copy()
    self.PosicionesTotales.append(pos1)
    self.Pos.clear()

    txt=''
    for i in self.Titulos:
        mkdir(f'{i}')

    for a in range(len(self.Titulos)):
        f=open(f'{self.Titulos[a]}/Original.html','w')
        txt=""

```

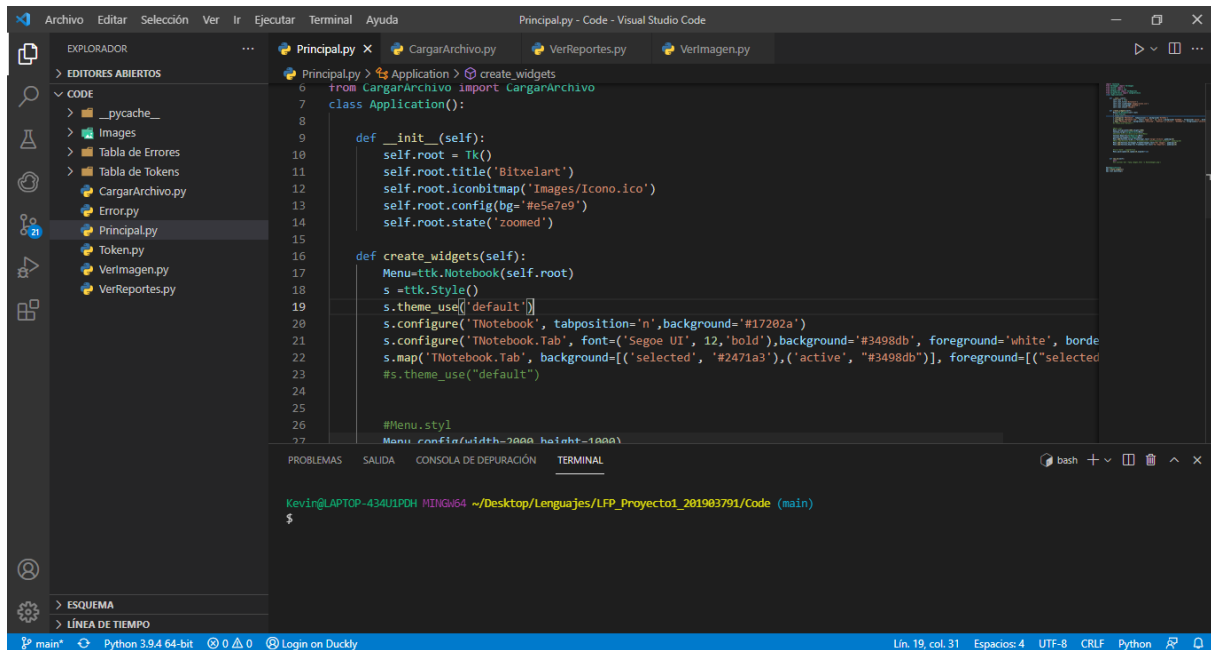
q55



TITULO	(TITULO'=""Cadena+"";')\$
ANCHO	(ANCHO'='Digito+;')\$
ALTO	(ALTO'='Digito+;')\$
FILAS	(FILAS'='Digito+;')\$
COLUMNAS	(COLUMNAS'='Digito+;')\$
CELDA	(CELDA'='{'(['Digito+,Digito+,TRUE FALSE,#HHHHHHH])+'}';')\$
FILTROS	(FILTROS'=(LETRA+)(LETRA+',')*';')\$

Editor de código utilizado para la aplicación

Visual Studio Code de Microsoft



- Modern Operating System:
 - Windows 7 or 10
 - Mac OS X 10.11 or higher, 64-bit
 - Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
- x86 64-bit CPU (Intel / AMD architecture)
- 4 GB RAM
- 5 GB free disk space

Librerías utilizadas

```
import textwrap
from VerImagen import VerImagen
from tkinter import *
from tkinter import ttk
```

- **Tkinter:** Se considera un estándar para la interfaz gráfica de usuario para Python y es el que viene por defecto con la instalación para Microsoft Windows.

```
from html2image import Html2Image
import webbrowser
```

- **webbrowser:** El pequeño módulo estándar webbrowser permite abrir un documento o sitio web en el navegador por defecto en sistemas Unix, Windows y Mac OS X. Posibilita la visualización en una nueva pestaña, ventana o en la actual, siempre que sea posible
- **html2image:** toma captura a los navegadores web existentes para generar imágenes a partir de URL y de cadenas o archivos HTML + CSS.