

---

## Software de servicios web

---

202010316 – Joseph Jeferson Marroquín Monroy

### Resumen

Este software es una aplicación solicitada por La Superintendencia de Administración Tributaria, se implementa distintos servicios web donde se lee un archivo de entrada y el programa lo procesa generando reportes en pdf o gráficos.

Por medio de este software se tiene un mejor control de las facturas que se registran en determinadas fechas y al mismo tiempo poder generar graficas o reportes de lo analizado.

### Palabras clave

Programación orientada a objetos (POO), XML, base de datos, JSON

### Abstract

*This software is an application requested by the Superintendence of Tax Administration, it implements different web services where an input file is read and the program processes it generating reports in pdf or graphics.*

*By means of this software you have a better control of the invoices that are recorded on certain dates and at the same time be able to generate graphs or reports of the analyzed.*

### Keywords

*Object oriented programming (POO), XML, database, JSON*

## Introducción

Desarrollamos un software que sigue un algoritmo el cual lee un archivo de entrada y lo analiza reconociendo que la facturas que se mandan tengan una estructura correcta, nit valido y así llevar un mejor control de lo que se registra.

Usamos programación orientada a objetos (POO) para guardar datos como la fecha o el gasto total para después poder graficarlo y generar reportes.

## Desarrollo del tema

Una clase es un modelo para crear a partir de ella objetos, es la encargada de almacenar la información, crear características a partir del objeto, agrupándose los objetos en una misma clase.

Una función es un bloque de código reutilizable que se encarga de realizar determinada tarea.

### POO

La Programación Orientada a Objetos es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos se utilizan como metáfora para emular las entidades reales del negocio a modelar.

## DJANGO

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como modelo–vista–controlador.

### Flask

Flask es un framework minimalista escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código. Está basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2 y tiene una licencia BSD.

### XML

XML, siglas en inglés de eXtensible Markup Language, traducido como 'Lenguaje de Mercado Extensible' o 'Lenguaje de Marcas Extensible', es un metalenguaje que permite definir lenguajes de marcas desarrollado por el World Wide Web Consortium utilizado para almacenar datos en forma legible

### Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional.

## Implementación de librerías

```
from flask import Flask,request,jsonify
from flask_cors import CORS
from entrada import Entrada
import json,re
import xml.etree.ElementTree as xml
import xml.etree.ElementTree as ET
import re
from matplotlib import pyplot as plt
```

Usamos xml.etree.ElementTree la cual se encarga de analizar o crear documento con extensión XML representándolos como un árbol, tambien nos ayudamos de matplotlib para realizar nuestras gráficas, la librería re no ayuda para identificar la elaboración a seguir de cada producto por medio de expresiones regulares.

## Implementación de clases

```
def crearEntrada(fecha,total):
    entrada.append(Entrada(fecha,total))
```

En la función de crearEntrada la usamos para guardar datos como la fecha o el gasto total de la factura lo cual vamos a usar para realizar nuestras graficas.

```
def cargamasiva(data):
```

Esta función la usamos para leer el archivo de entrada y así crear nuestro archivo de salida analizando todos los datos verificando que tenga la estructura correcta y los nit estén bien dados.

```
def graficaR(data):
```

Esta función es la encargada de graficar según la fecha dada, mostrando los datos en una gráfica con puntos.

```
def graficaRango(dataI,dataF):
```

Esta función lee un rango de fechas las cuales vamos a graficar mostrándolos en una grafica de puntos.

```
@app.route('/cargar',methods=['POST'])
```

Usamos endpoints para conectar flask con el frontend, mandando una ruta la cual ejecutara una determinada función.

```
class Entrada:
```

Usamos la clase entrada para crear nuestro constructor como los gets and sets para guardar nuestros datos.

```
function cargar(){
```

Nos ayudamos de javascript para realizar funciones y tambien conectar el backend con el frontend, la función cargar manda los datos del archivo de entrada para analizarlo.

```
documentInput.addEventListener('change', (e)=> {
```

Nos ayudamos de javascript para que al momento de cargar un archivo por un input file el contenido del archivo se muestre en un área de texto.

```
function rangoPDF(){
```

Nos ayudamos de javascript para realizar los reportes en pdf, generándolos al darle clic a un botón.

```
function graficarRango(){
```

Usamos javascript para mandar la fecha ingresada y así poder graficar según el total de la factura de acuerdo a las fechas dadas.

```
<script src="https://raw.githubusercontent.com/eKoopmans/html2pdf/master/dist/html2pdf.bundle.js"></script>
```

Importamos la librería de html2pdf para poder generar los reportes solicitados.

```
<button class="content-button" onclick="graficarRango()">Graficar</button>
```

Usamos onclick para llamar a la función de nuestro archivo de javascript.

## Conclusiones

La implementación de un software para que procese las facturas hace un trabajo más óptimo y eficiente.

Esta aplicación nos representa la conexión entre un archivo de salida como base de datos para mantener persistencia en los datos.

El uso de framework como lo es Django o flask nos proporciona una ayuda al realizar nuestro frontend o backend y hacer el desarrollo de un proyecto más óptimo.

## Referencias bibliográficas

Métodos Marplotlib [online] Disponible en:

<https://programmerclick.com/article/3751558386/>

Como hacer graficos [online] Disponible en:

<http://how.okpedia.org/es/python/como-dibujar-un-punto-en-el-grafico-de-python>

## Anexos

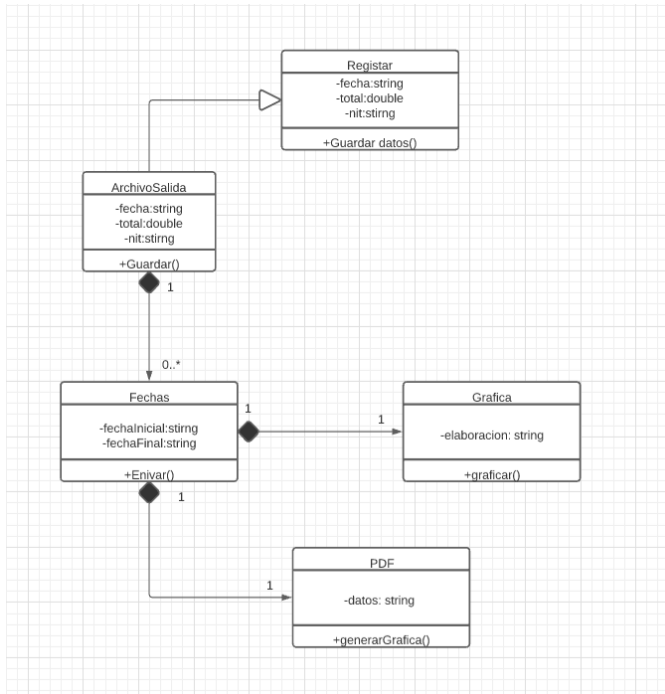


Figura 1. Diagrama de clases

Fuente: elaboración propia.

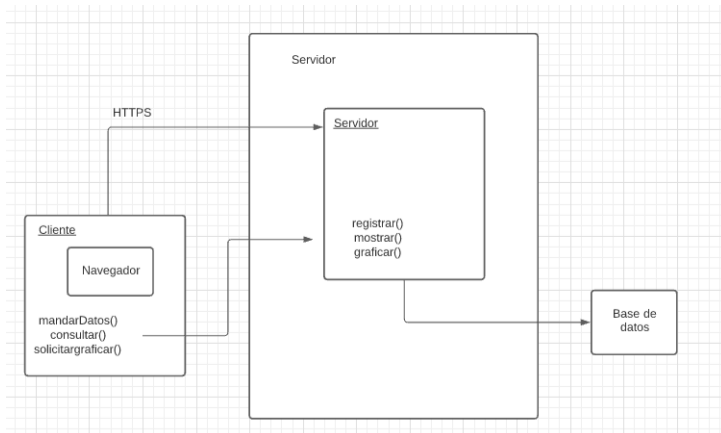


Figura 1. Arquitectura

Fuente: elaboración propia.