Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2021.

AUTORIZADOR DE DTE'S SAT

Carnet: 202001923 – Anthony Alexander Aquino Santiago

Resumen

Un software capaz de procesar datos de usuarios de una entidad gubernamental es esencial para la modernización de dichas entidades.

Al construir la solución del problema se pudo evidenciar que la misma podría impactar de forma positiva en las muchas entidades gubernamentales, como las municipalidades, etc.

Las solicitudes que se procesan en alguna empresa o centro gubernamental requieren de un análisis adecuado para poder determinar qué se hace con las mismas. Esto se facilita desarrollando un servicio, como el de este proyecto, que evita procesar las solicitudes que llegan a la SAT de forma manual, ahorrando así recursos y tiempo, y también se crea un proceso más eficiente, generando confianza en la empresa o entidad.

Es importante implementar un servicio óptimo para lograr los beneficios esperados, así como también implementar seguridad informática, ya que el servicio puede sufrir ataques que intercepten los datos procesados.

Palabras clave : Servicio, seguridad, solicitud, proceso, recursos.

Abstract

A software capable of processing user data of a government entity is essential for the modernization of these entities.

When constructing the solution to the problem, it was shown that it could positively impact many government entities, such as municipalities, etc.

The requests that are processed in a company or government center require an adequate analysis to be able to determine what is done with them. This is facilitated by developing a service, such as the one in this project, that avoids processing the requests that arrive at the SAT manually, thus saving resources and time, and also creating a more efficient process, generating trust in the company or entity.

It is important to implement an optimal service to achieve the expected benefits, as well as to implement computer security, since the service can suffer attacks that intercept the processed data.

Keywords: *Service, security, process, request, resources.*

Introducción

La implementación de los lenguajes de programación en el desarrollo de servicios web para las industrias, es esencial, ya que se deben usar las herramientas que proporcionan para crear un servicio útil. Entre las herramientas que podrían utilizarse están: creación de interfaces gráficas, procesamiento masivo de información ingresada, uso de lógica - matemática para la toma de decisiones en el procesamiento de la información, creación de archivos de salida, etc.

Un ejemplo de lo anterior es la implementación de un servicio que recibe solicitudes masivamente y que autoriza determinadas solicitudes que cumplen con ciertas condiciones.

Para desarrollar un servicio web es indispensable un *frontend* amigable con el usuario para que el servicio pueda ser utilizado correctamente, así como también un *backend* funcional que procese los datos correctamente. La mejor opción para desarrollar un servicio web es tener un equipo de trabajo para cada parte del servicio. Uno para el *frontend* y uno para el *backend*.

Desarrollo del tema

En el desarrollo del servicio de autorizaciones de Documentos Tributarios Electrónicos (DTE's) para la Superintendencia de Administración Tributaria (S.A.T.), se implementaron algoritmos de programación utilizando el lenguaje de programación Python y los *frameworks* Flask y Django. Se usó el paradigma de programación orientado a objetos, puesto que su uso se acoplaba muy bien a los requerimientos del servicio. Se creó una clase Fecha para luego crear objetos de tipo fecha , con los siguientes atributos:

-fecha: para guardar la fecha con el formato DD/MM/AAAA, de tipo String.

-recibidas: para guardar la cantidad de facturas recibidas en dicha fecha.

-errores: para guardar la cantidad de facturas recibidas con errores en dicha fecha.

-correctas: para guardar la cantidad de facturas recibidas sin errores en dicha fecha.

-emisores: para guardar la cantidad de nits emisores diferentes en cada factura recibida sin errores en dicha fecha.

-receptores: para guardar la cantidad de nits receptores diferentes en cada factura recibida sin errores en dicha fecha.

-autorizaciones: para guardar objetos de tipo Factura sin errores recibidos en dicha fecha.

En el caso del último atributo, se creó una clase de tipo Factura y se fueron guardando objetos en una lista, la cual conforma el atributo autorizaciones.

Se puede observar en la figura 1 el método constructor de la clase Fecha.

```
class Fecha:
    def __init__(self, fecha, recibidas, errores, correctas, emisores, receptores, autorizaciones):
        self.fecha = fecha
        self.recibidas = recibidas
        self.errores = errores
        self.correctas = correctas
        self.emisores= emisores
        self.receptores = receptores
        self.autorizaciones = autorizaciones #Este atributo va a contener la lista de facturas aprobadas
```

Figura 1. Constructor de la clase Fecha.

Fuente: elaboración propia, 2021.

Se puede observar en la figura 2 el método constructor de la clase Factura.

```
class Factura:
    def __init__(self, tiempo, referencia, nit_emisor, nit_receptor, valor, iva, total, codigo):
        self.tiempo = tiempo
        self.referencia = referencia
        self.nit_emisor= nit_emisor
        self.nit_receptor = nit_receptor
        self.valor= valor
        self.iva = iva
        self.total = total
        self.codigo = codigo
```

Figura 1. Constructor de la clase Factura.

Fuente: elaboración propia, 2021.

También, como en todo desarrollo de un software, se utilizaron variables. Se listan las más relevantes a continuación.

Tabla I.

Algunas variables utilizadas

Nombre	Descripción
facturas	De tipo lista. Guardó objetos de tipo fecha, con todos sus atributos.
productos	De tipo lista. Guardó objetos de tipo factura, con todos sus atributos.
f1	De tipo String. Guardó un string fecha que representó una fecha de inicio para generar el resumen de un rango de fechas.
f2	De tipo String. Guardó un string fecha que representó una fecha de finalización para generar el resumen de un rango de fechas.
fI	De tipo String. Guardó un string fecha que representó una fecha de inicio para generar el resumen de un rango de fechas.
fF	De tipo String. Guardó un string fecha que representó una fecha de finalización para generar el resumen de un rango de fechas.

Fuente: elaboración propia, 2021.

Se listan los métodos más importantes en el desarrollo del simulador.

Tabla II.

Métodos utilizados

Nombre	Descripción
cargar(ruta)	Permitió procesar el archivo de entrada xml para insertar los datos leídos en cada una de las variables definidas. También se asignaron

	atributos de objetos en este método. Como parámetro recibió la ruta del archivo de entrada.
salida()	Generó el archivo autorizaciones.xml en donde se pusieron todos los datos de salida.
llenarF()	Llenó la lista de fechas para poder generar el archivo de salida.

Fuente: elaboración propia, 2021.

En la creación de los métodos, se tuvo que elegir la solución más óptima para resolver el problema, realizando esquemas y pruebas para determinar qué herramientas (estructuras de control, ciclos) utilizar en cada uno de los mismos.

Conclusiones

Se puede concluir que los lenguajes de programación como Python son útiles para desarrollar servicios web, en la parte del backend.

Utilizar un framework para el frontend es muy útil para poder tener un mejor control de lo que se muestra en cada página web en html.

Referencias bibliográficas

M. Carreño, et al. (2012). *Problemas Resueltos de Listas.* Universidad Autónoma de Baja California.

Graphviz. (10 de agosto del 2021). *What is Graphviz?* https://graphviz.org/#what-is-graphviz

Apéndice:

```
                    <img src="masnup-icon.svg" class="navbar-logo-img" alt="">
                    <h3 style="text-align: center; color:snow;"> SAT </h3>
                    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span></button>
                    <div class="collapse navbar-collapse" id="navbarSupportedContent">
                        <ul class="navbar-nav ms-auto mt-2 mt-lg-0">
                            <li class="nav-item dropdown">
                                <a class="nav-link dropdown-toggle" id="navbarDropdown" href="#" data-bs-toggle="dropdown"
aria-haspopup="true" aria-expanded="true" style="color: snow;">Peticiones</a>
                                <div class="dropdown-menu dropdown-menu-end" role="menu" aria-labelledby="navbarDropdown">
                                    <div class="dropdown-divider"></div>
                                    <a class="dropdown-item" href="{% url 'agregar_fechas' %}">Resumen de movimientos por
fecha</a>
                                    <div class="dropdown-divider"></div>
                                    <a class="dropdown-item" href="{% url 'rango_fechas' %}">Ver Resumen de totales y sin
IVA <br> (Rango) </a>
                                </div>
                                <li class="nav-item"><a style="color: snow;" class="dropdown-item" href="{% url 'ayuda'
%}">Ayuda</a>
                            </li>
                        </ul>
                    </div>
                </div>
            </nav>
            <!-- Page content-->
            <div class="container-fluid">
                <main>
                    <div class="container-fluid">
                        <h1 style="text-align: center; color:cornflowerblue;">Superintendencia de Administración
Tributaria</h1>
                        <h3 style="text-align: center; color:purple;">Autorizador de Documentos Tributarios Electrónicos
(DTE)</h1>
                        <h6>Cargar Archivo</h6>
                        <input type="file" accept="text/plain" onchange="cargarArchivo(event)">
```

Figura 3. Código en html de la página de inicio del servicio.

Fuente: elaboración propia, 2021.