

---

## PROYECTO NO.2 PRIMER SEMESTRE 2021

---

201700747 – Dennis Alexander Gamboa Stokes

### Resumen

La Programación Orientada (POO) a objetos permite que el código sea reutilizable, organizado y fácil de mantener. Sigue el principio de desarrollo de software utilizado por muchos programadores DRY (Don't Repeat Yourself), para evitar duplicar el código y crear de esta manera programas eficientes mientras Graphviz es una herramienta utilizada por muchas personas que permite la creación de graficas usando un lenguaje específico.

Se investigo Django(usado como front-end) y Flask (usado como back-end)para poder entender sus métodos y poder crear 2 servidores para poder crear una aplicación web con el uso de Programación orientada a Objetos como también se investigo la estructura XML para la obtención de datos como también como guardarlos.

### Palabras clave

Django,Flask,XML,POO,Plotly

### Abstract

Object Oriented Programming (OOP) allows code to be reusable, organized and easy to maintain. It follows the software development principle used by many programmers DRY (Don't Repeat Yourself), to avoid duplicating the code and thus create efficient programs while Graphviz is a tool used by many people that allows the creation of graphs using a language specific.

Django and Flask were investigated to understand their methods and to be able to create 2 servers to create a web application with the use of Object-Oriented Programming, as well as the XML structure to obtain data as well as how to save them.

.

### Keywords

Django,Flask,XML,POO,Plotly

## Introducción

En el Presente trabajo o programa se mostrará la información principal de los métodos utilizados en la Programación orientada a objetos (POO) usando Django como Front-end y Flask como Back-end .

Se mostrara los datos en estructuras XML como también

El uso de Plotly que es la herramienta usada mundialmente para la creación de graficas que cualquier tipo.

## Desarrollo del tema

### XML:

Extensible Markup Language (XML) se utiliza para describir datos. El estándar XML es una forma flexible de crear formatos de información y compartir electrónicamente datos estructurados a través de la Internet pública, así como a través de redes corporativas.

#### Usos:

**Intercambio de datos entre sistemas**, uno de los objetivos fundamentales de XML es permitir la posibilidad de intercambiar datos de forma estructurada entre diferentes sistemas. Al tratarse de un formato de texto plano y ser un lenguaje estandarizado, hace que esta transferencia sea muy ágil e independiente de la plataforma utilizada.

**Base de datos**, XML permite guardar datos de forma estandarizada para luego poder ser tratados por multitud de lenguajes diferentes. Su manejo es mucho más sencillo que bases de datos como MySQL y mucho más rico que utilizar ficheros de texto planos.

**Conversor**, actualmente son muchos los formatos que ofrecen servicios de conversión a XML, como PDF, HTML, .text, .docx o XHTML.

<? xml version = "1.0" standalone = "yes"?>

<conversacion>

<saludo> ¡Hola, mundo! </saludo>

<respuesta> Cuiden al planeta </respuesta>

</conversacion>.

*Figura 1.*Estructura de XML.

Fuente: elaboración propia

## PLOTLY:

La biblioteca plotly de Python es una biblioteca de trazado interactiva de código abierto que admite más de 40 tipos de gráficos únicos que cubren una amplia gama de casos de uso estadísticos, financieros, geográficos, científicos y tridimensionales. Construido sobre la biblioteca Plotly JavaScript (plotly.js), plotly permite a los usuarios de Python crear hermosas visualizaciones interactivas basadas en la web que pueden mostrarse en cuadernos de Jupyter, guardarse en archivos HTML independientes o servirse como parte de una web pura construida por Python aplicaciones que utilizan Dash. La biblioteca de Python plotly a veces se denomina "plotly.py" para diferenciarla de la biblioteca de JavaScript. Gracias a la integración profunda con la utilidad de exportación de imágenes de orca, plotly también proporciona un gran soporte para contextos no web, incluidos editores de escritorio (por ejemplo, QtConsole, Spyder, PyCharm) y publicación de documentos estáticos (por ejemplo, exportar cuadernos a PDF con imágenes vectoriales de alta calidad).

## FLASK:

Flask es un marco de micro web escrito en Python. Se clasifica como un microframework porque no requiere herramientas o bibliotecas particulares. No tiene una capa de abstracción de base de datos, validación de formularios ni ningún otro componente donde las bibliotecas de terceros preexistentes proporcionan funciones comunes. Sin embargo, Flask admite extensiones que pueden agregar funciones de la aplicación como si estuvieran implementadas en el propio Flask. Existen extensiones para mapeadores relacionales de objetos, validación de formularios, manejo de carga, varias tecnologías de autenticación

abierta y varias herramientas comunes relacionadas con el marco.

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World"

if __name__ == "__main__":
    app.run(debug=False)
```

## DJANGO:

Django es un marco web de Python de alto nivel que fomenta el desarrollo rápido y el diseño limpio y pragmático. Creado por desarrolladores experimentados, se encarga de gran parte de la molestia del desarrollo web, por lo que puede concentrarse en escribir su aplicación sin necesidad de reinventar la rueda. Es gratis y de código abierto.

Ridículamente rápido.

Django fue diseñado para ayudar a los desarrolladores a llevar las aplicaciones desde el concepto hasta su finalización lo más rápido posible.

Tranquilizadoramente seguro.

Django se toma la seguridad en serio y ayuda a los desarrolladores a evitar muchos errores de seguridad comunes.

Extremadamente escalable.

Algunos de los sitios más activos de la Web aprovechan la capacidad de Django para escalar de forma rápida y flexible.

```
from django.contrib import admin

from django.urls import include, path
```

```
urlpatterns = [

    path('polls/',
        include('polls.urls')),

    path('admin/', admin.site.urls),

]

from django.http import HttpResponse
```

```
def index(request):

    return HttpResponse("Hello, world.
        You're at the polls index.")
```

## Estructura de Programación:

### Estructura Condicional:

Permite alterar la secuencia normal de pasos en un paso específico del Algoritmo, para crear 2 alternativas de bloques de ejecución, de manera excluyente entre ambos. En otras palabras: Solo uno de los 2 bloques se ejecutará, nunca ambos bloques y permite decidir por cuál alternativa seguirá el flujo del programa dependiendo del resultado de la evaluación de una condición. Para establecer condiciones complejas se utilizan los operadores relacionales y lógicos

### Estructura cíclica:

Se llaman problemas repetitivos o cíclicos a aquellos en cuya solución es necesario utilizar un mismo conjunto de acciones que se puedan ejecutar una cantidad específica de veces. Esta cantidad puede ser

## CONCLUSIONES

- FLASK es un framework minimalista escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código.
- PLOTLY es una herramienta que permite graficar diferentes figuras como de barras o pie chart.
- Archivos XML nos permiten hacer una carga masiva de datos en un programa con una estructura muy básica, y fácil de manejar, así como importar datos de un programa
- Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles
- Estructuras Secuencial permite que una instrucción siga a otra en secuencia.
- Estructuras condicionales son instrucciones que se pueden ejecutar o no en función del valor de una condición.
- Estructuras cíclicas permite ejecutar fragmentos de Código un umero limitado de veces

## Referencias bibliográficas

Máximo 5 referencias en orden alfabético.

- Estructura de XML y usos en <http://www.maestrosdelweb.com/xmlusos/>
- Estructura de Graphviz, Disponible en <https://graphviz.org/>
- Programación orientada a objetos, Disponible en <https://desarrolloweb.com/articulos/499.php#:~:text=La%20programaci%C3%B3n%20Orientada%20a%20objetos%20se%20define%20como%20un%20paradigma,los%20objetivos%20de%20las%20aplicaciones.>
- Programación orientada a objetos, Disponible en <https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/>
- <http://www.rdebug.com/2010/10/matriz-ortogonal-estructura-de-datos-en.html#:~:text=una%20matriz%20ortogonal%20es%20una,las%20columnas%20representan%20los%20modelos.>
- <https://www.djangoproject.com/>
- <https://flask.palletsprojects.com/en/1.1.x/>
- Estructuras de programación secuenciales, cíclicas y condicionales Disponible en <https://prezi.com/fmivkws5biwl/estructuras-condicionales-secuenciales-y-ciclicas/>