Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 2do. Semestre 2024

# ENSAYO PROYECTO 2 DE INTRODUCCION A LA PROGRAMACION 2

**202300673 – Álvaro Gabriel Ceballos Gil**

**Resumen**

En el presente ensayo se encuentra una explicación clara y detallada de todos los procesos y herramientas utilizados para la implementación del proyecto 2 de introducción a la programación 2.
Se utilizó Flask como framework, y para poder representar la arquitectura de cliente-servidor se utilizó flask para ambos, y un archivo html para la realización de los cambios de pantallas.

Se desarrolló una solución en la que se obtenían distintos productos de un xml, y el programa debía de ser capaz de calcular el tiempo óptimo (tiempo mínimo) en los que podía ensamblar todos los componentes, exportando los resultados en gráficas y documentos xml y html. Para la implementación de la solución se hizo uso de múltiples tdas, entre los más utilizados estuvo la lista enlazada simple, la lista enlazada doble, y la cola.

También se hizo uso de jinja2 en ocasiones, para poder implementar algormitmos de programación dentro de los archivos html.

**Palabras clave**

Framework, lista enlazada, Flask, html, phyton

***Abstract***

*In this essay you will find a clear and detailed explanation of all the processes and tools used for the implementation of project 2, introduction to programming 2.*

*Flask was used as a framework, and in order to represent the client-server architecture, Flask was used for both, and an HTML file was used to make the screen changes.*

*A solution was developed in which different products were obtained from an xml, and the program had to be able to calculate the optimal time (minimum time) in which it could assemble all the components, exporting the results in graphs and xml and html documents. . For the implementation of the solution, multiple tdas were used, among the most used were the single linked list, the double linked list, and the queue.*

***Keywords***

*Framework, linked list, Flask, html, phyton*

## Introducción

En el presente ensayo se encuentra una explicación clara y detallada de todos los procesos y herramientas utilizados para la implementación del proyecto 2 de introducción a la programación 2.

Dentro del flujo de trabajo del proyecto, se pueden observar y apreciar la implementación de múltiples listas enlazadas para darle solución al proyecto.

Se hizo una amplia utilización de Flask, debido a que en el presente proyecto se utilizó Flask tanto para backend como para frontend.

## Desarrollo del tema

El presente proyecto nace de la necesidad de una compañía de contar con un software capaz de calcular el tiempo óptimo en el que una máquina con brazo robótico sería capaz de ensamblar todas las piezas, alternando entre líneas y componentes, teniendo limitantes, como el hecho de que no podía ensamblar 2 productos a la vez. El desarrollo de este programa capaz de calcular el tiempo óptimo del ensamblaje de un producto resulta muy útil y efectivo, es por eso que por medio de graphviz se puede exportar un archiovo que indica cuanto falta para el ensamblaje del producto, y un archivo html y xml para documentar los procesos utilizados.
Para el desarrollo del programa se implementaron múltiples TDA's, entre los más utilizados están los siguientes:

- Listas enlazadas simples: Para poder almacenar los datos individuales de la matriz, se ha hecho el uso de una lista enlazada simple. La lista enlazada simple cuenta con un apuntador inicial, y un nodo el cual contiene un apuntador que apuntará a los otros nodos. Para poder imprimirlo correctamente, se ha creado una clase dato, la cual contenía su posición en x, posición en y, y por último el dato. Y de esta manera se ha logrado la correcta impresión de la matriz por medio del archivo XML

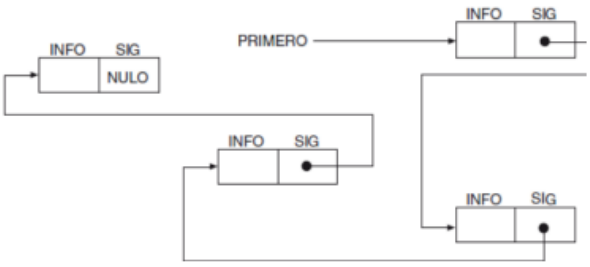



*Figura 1. Estructura de lista enlazada simple*

*Fuente: Ingenierio Marlon (2024).*

Una vez culminada la realización de la s listas enlazadas simples para extraer los datos del xml, se procedió con la implementación del algoritmo capaz de leer los atributos que contiene dentro el xml.

Una vez leído el xml, se procedió con la implementación de una cola para almacenar la

elaboración de los distintos productos provenientes del xml.

- .Colas

A diferencia de las pilas, las colas se basan en el método en el que el primero en entrar, es el primero en salir. En este método se contarán con 2 apuntadores, uno en el frente y otro en el pondo de la cola.

Algunos de los usos que se le pueden dar a las colas, son las colas para consumo de servicios web o de impresión

Insertar: Al momento en el que se vayan a insertar datos, el primer dato tendrá tanto al frente y al fondo apuntando hacia él, ya que es el único dato en la cola. Si se llegaran a insertar más datos, el primero tendrá a "frente" como apuntador y el último en insertarse tendrá "fondo".

Eliminar: para eliminar, como el primer en entrar es el primero en salir, el procedimiento que se realiza es que el apuntador de "frente" apuntará al siguiente nodo de la cola, y de esta forma se eliminaría al primero en entrar.


*Figura 2. Procedimiento de inserción de datos en colas*

*Fuente: Ingenierio Marlon (2024)*

Una vez implementada la cola, por medio de strp y Split se separó la cola en filas y columnas, y se desarrolló una lista de listas. Donde en la cabeza todos tenemos los nodos correspondientes a las líneas de producción, y cada nodo de la lista de líneas de producción tiene una lista enlazada doble de componentes. Es por eso que se proseguirá con la explicación de la lista enlazada doble

- Lista enlazada doble

  Una lista enlazada doble es una estructura de datos donde cada nodo tiene dos punteros: uno que apunta al siguiente nodo y otro que apunta al nodo anterior. Esto permite recorrer la lista en ambas direcciones, hacia adelante y hacia atrás. Esta estructura permite recorrer la lista desde cualquier nodo en ambas direcciones (hacia adelante o hacia atrás), lo que la hace más flexible que una lista enlazada simple. También facilita operaciones como insertar o eliminar nodos

en el medio de la lista, ya que los enlaces a los nodos vecinos están disponibles en ambas direcciones.

Es por estos motivos que se optó por la implementación de una lista enlazada doble, ya que de esta forma podíamos retroceder en caso de ser necesario para poder realizar el proceso de ensamblaje.

Estas fueron las tdas principales utilizadas en el proyecto, ahora se explicarán las herramientas que fueron muy útiles en el desarrollo de este proyecto.

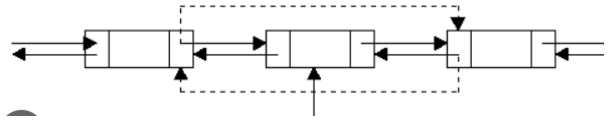


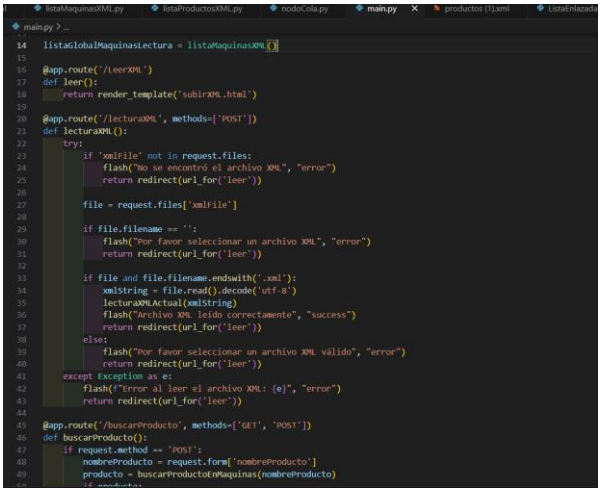
*Figura 2. Procedimiento de una lista enlazada doble*

*Fuente: Universidad de granada (2024)*

- Flask

  Flask es un microframework web escrito en Python. Es ligero y sencillo, diseñado para facilitar el desarrollo de aplicaciones web. A diferencia de frameworks más grandes como Django, Flask no incluye muchas funcionalidades predefinidas, lo que permite a los desarrolladores agregar solo los componentes que necesiten (como manejo de bases de datos o autenticación). Es flexible y adecuado para proyectos de cualquier tamaño, ofreciendo control total sobre el flujo de la aplicación y el manejo de rutas, plantillas, y peticiones HTTP.

  Flask es un framwework más orientado a frontend, pero este también puede ser perfectamente adaptable al frontend.



```python
listaGlobalMaquinasLectura = listaMaquinasXML()

@app.route('/LeerXML')
def leer():
    return render_template('subirXML.html')

@app.route('/lecturaXML', methods=['POST'])
def lecturaXML():
    try:
        if 'xmlFile' not in request.files:
            flash("No se encontró el archivo XML", "error")
            return redirect(url_for('leer'))

        file = request.files['xmlFile']

        if file.filename == '':
            flash("Por favor seleccionar un archivo XML", "error")
            return redirect(url_for('leer'))

        if file and file.filename.endswith('.xml'):
            xmlString = file.read().decode('utf-8')
            lecturaXMLActual(xmlString)
            flash("Archivo XML leído correctamente", "success")
            return redirect(url_for('leer'))
        else:
            flash("Por favor seleccionar un archivo XML válido", "error")
            return redirect(url_for('leer'))
    except Exception as e:
        flash(f"Error al leer el archivo XML: {e}", "error")
        return redirect(url_for('leer'))

@app.route('/buscarProducto', methods=['GET', 'POST'])
def buscarProducto():
    if request.method == 'POST':
        nombreProducto = request.form['nombreProducto']
        producto = buscarProductoEnMaquinas(nombreProducto)
```

*Figura 4. Enpoins en Flask*

*Elaboración Propia (2024)*

- HTML y Jinja 2

  HTML con Jinja2 es la combinación de plantillas HTML y el motor de plantillas Jinja2. Jinja2 permite insertar lógica de programación dentro del código HTML, como variables, bucles, y condicionales. Esto es útil en aplicaciones web, ya que permite generar dinámicamente el contenido de las páginas.

Por ejemplo, en Flask puedes usar una plantilla HTML con Jinja2 para mostrar datos desde el backend, iterar sobre listas, o mostrar diferentes vistas basadas en condiciones, todo sin salir del archivo HTML. Jinja2 facilita la integración de lógica en las plantillas, haciéndolas más interactivas y dinámicas.

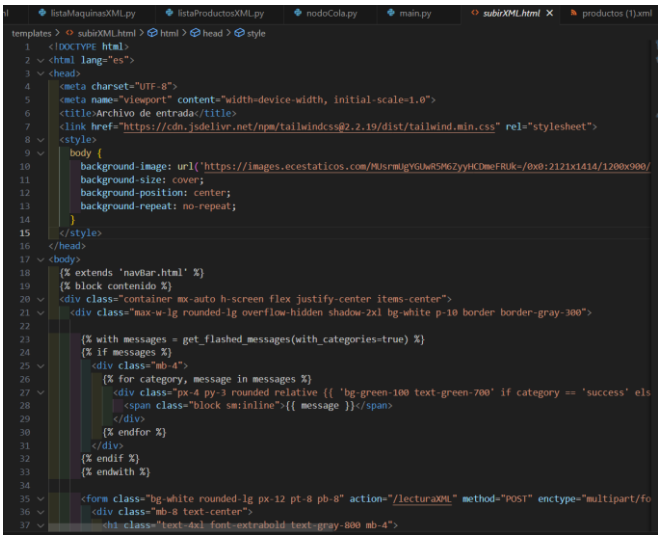



*Figura 5. HTML y Jinja2*

*Elaboración Propia (2024)*

- d. Archivos XML

  Un archivo de lenguaje de marcado extensible.

  Es un documento basado en texto que se puede

  guardar con la extensión .xml. Puede escribir

  XML de forma similar a otros archivos de texto.

  Sirve para el manejo de información por medio

  de archivos estructurados.

  Un archivo cml cuenta con una estructura

  específica, se maneja por medio se etiquetas.

  Las etiquetas <xml></xml> se utilizan para

  marcar el principio y el final de un archivo

  XML. El contenido de estas etiquetas también

  se denomina documento XML. Es la primera

  etiqueta que cualquier software buscará para

  procesar código XML.

  Dependiendo de la estructura que se dese

  implementar, se van colocando las etiquetas.

profundizar en la temática expuesta.

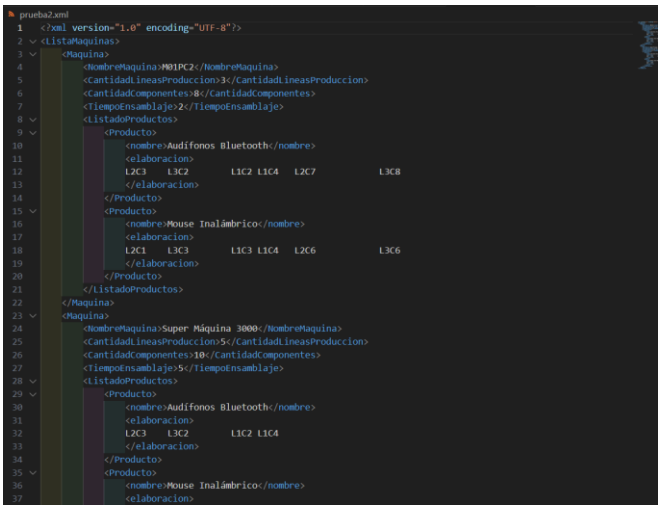



*Figura 5. XML*

*Elaboración Propia (2024)*

## Referencias bibliográficas

Exposito López, D., García Soto, A., & Gomez Antonio Jose, M. (1999-2000). *Listas doblemente enlazadas*. Universidad de Granada. https://ccia.ugr.es/~jfv/ed1/tedi/cdrom/docs/ldoble.html

## Apéndice

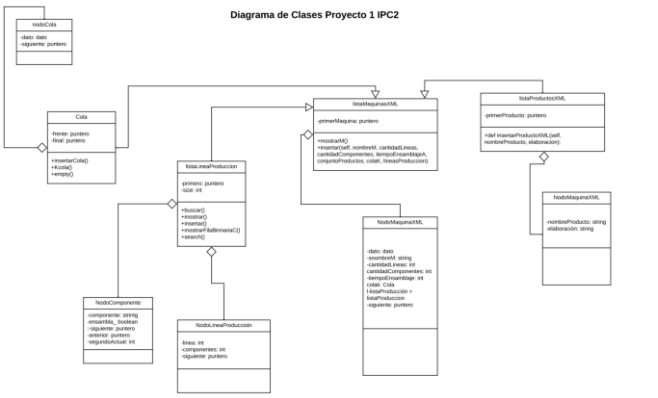



**Figura 11. Diagrama de Clases Fuente: Elaboración propia**

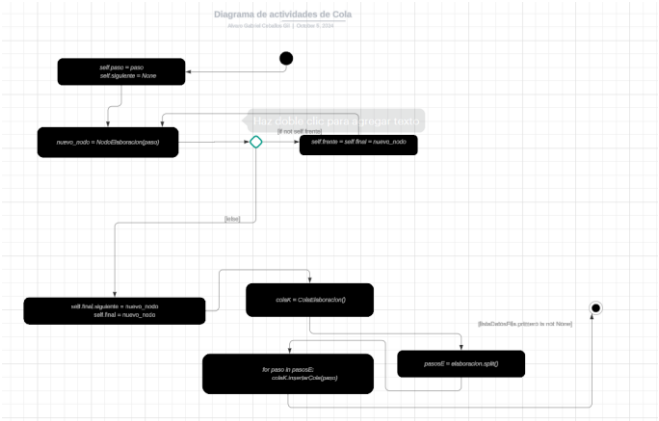



**Figura 11. Diagrama Actividades: Elaboración propia**