**Universidad de San Carlos de Guatemala**

Facultad de Ingeniería

Laboratorio Estructura de Datos

Sección B

**Manual técnico**

**Proyecto1**

Franklin Orlando Noj Perez

202200089

5to Semestre

Ing. En Ciencias y Sistemas

**FORTRAN**

Fortran, acrónimo de "Formula Translation", es un lenguaje de programación de alto nivel diseñado principalmente para cálculos numéricos y científicos. Fue desarrollado por IBM en la década de 1950 y ha experimentado varias versiones desde entonces. Fortran es conocido por su eficiencia en el manejo de operaciones matemáticas y su capacidad para trabajar con grandes conjuntos de datos.

En términos de soporte de computadoras, Fortran ha sido ampliamente adoptado en sistemas mainframe y supercomputadoras, así como en plataformas más modernas. Es compatible con una variedad de arquitecturas de procesadores, incluyendo x86, ARM y POWER, lo que permite su ejecución en una amplia gama de hardware.

En cuanto al consumo de recursos, Fortran es conocido por ser eficiente en términos de rendimiento y uso de memoria. Su diseño está orientado a la velocidad y la capacidad de procesar grandes cantidades de datos, lo que lo hace adecuado para aplicaciones científicas y de ingeniería que requieren cálculos intensivos.

**Consejos para Programar en Fortran:**

Optimización de Cálculos Numéricos: Fortran es conocido por su capacidad para realizar cálculos numéricos eficientes. Aprovecha las características específicas del lenguaje, como las matrices y las operaciones vectorizadas, para optimizar el rendimiento.

Uso de Módulos: Fortran soporta la programación modular, lo que facilita la organización del código en módulos. Utilizar esta característica puede mejorar la legibilidad y mantenibilidad del código.

Gestión de Memoria: Aunque Fortran maneja automáticamente la gestión de memoria en muchos casos, es importante entender cómo funciona para evitar posibles problemas de rendimiento.

**Ventajas de Fortran:**

Eficiencia: Fortran está diseñado para ser eficiente en términos de velocidad de ejecución y uso de memoria, lo que lo hace ideal para aplicaciones científicas y de ingeniería.

Herencia en Computación Científica: Dado que ha sido utilizado históricamente en el ámbito científico, Fortran tiene una amplia base de código existente y numerosas bibliotecas especializadas.

Compatibilidad con Hardware Diverso: Fortran es compatible con diversas arquitecturas de hardware, permitiendo su ejecución en una amplia variedad de sistemas, desde supercomputadoras hasta computadoras personales.

**Desventajas de Fortran:**

Sintaxis Antigua: La sintaxis de Fortran puede parecer anticuada en comparación con lenguajes más modernos, lo que puede dificultar la adopción para nuevos programadores.

Menos Orientado a Objetos: Aunque se han introducido características orientadas a objetos en versiones más recientes, Fortran tradicionalmente carece de las características avanzadas de programación orientada a objetos presentes en algunos lenguajes más modernos.

Arquitectura y Evolución:

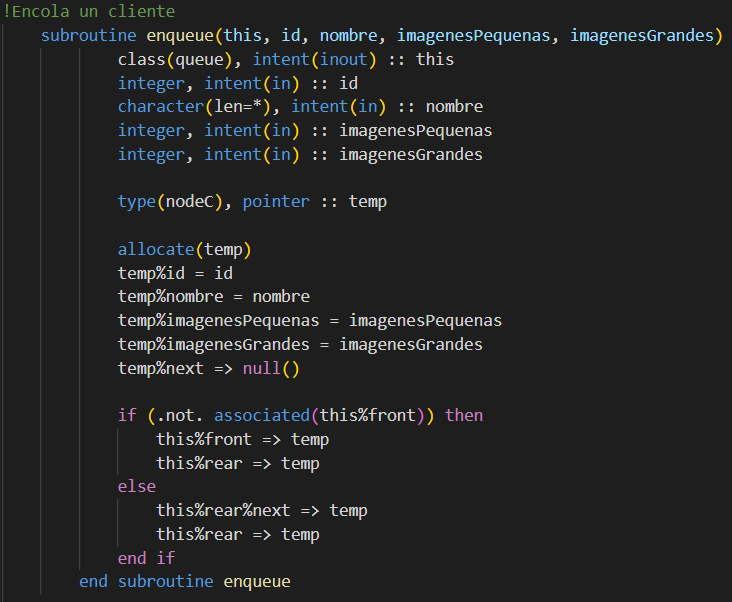
Fortran fue diseñado originalmente por IBM en la década de 1950. A lo largo de las décadas, ha experimentado múltiples revisiones y actualizaciones para adaptarse a las necesidades cambiantes de la computación científica y de ingeniería.

**Inventor e Historia:**

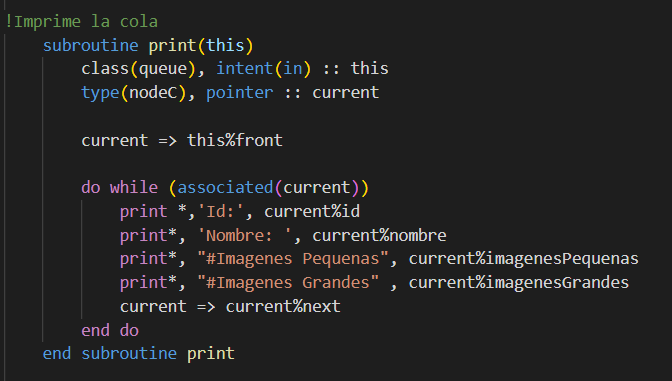
John W. Backus lideró el equipo de desarrollo de Fortran en IBM en la década de 1950. Fortran I, la primera versión, se lanzó en 1957. Backus recibió el Premio Turing en 1977 por su contribución al diseño de lenguajes de programación, incluido Fortran.

En resumen, Fortran sigue siendo una herramienta poderosa para cálculos numéricos y científicos, con una rica historia, una amplia base de usuarios y una evolución continua para adaptarse a las demandas de la computación moderna.

Modulo para agregar elementos a una cola desarrollada en Fortran 90



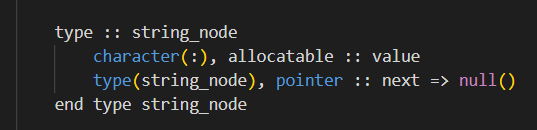
Esta cola recibe a los clientes que van a ingresar a las ventanillas de atención



Esta subrutina

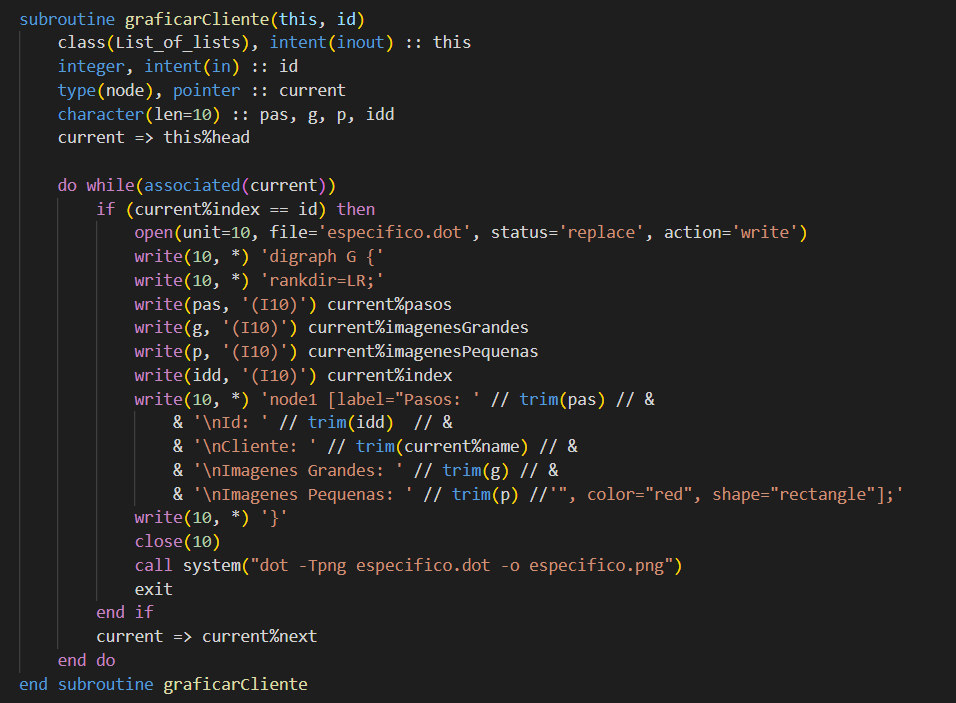
Imprime la cola de los clientes

Encolados

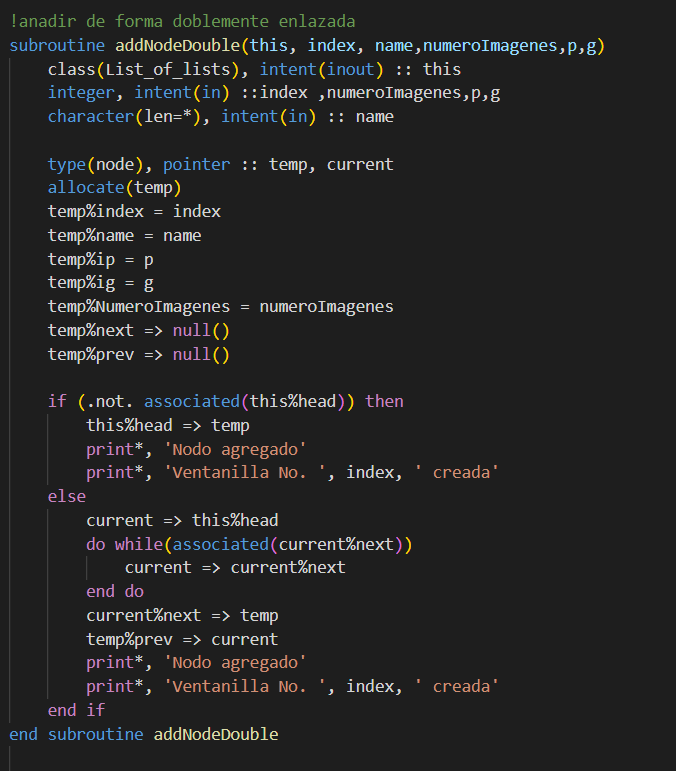


Este trozo de código es el que simula a un objeto, es lo mas parecido que hay a objetos en Fortran 90

***Los módulos simulas clases en Fortran***

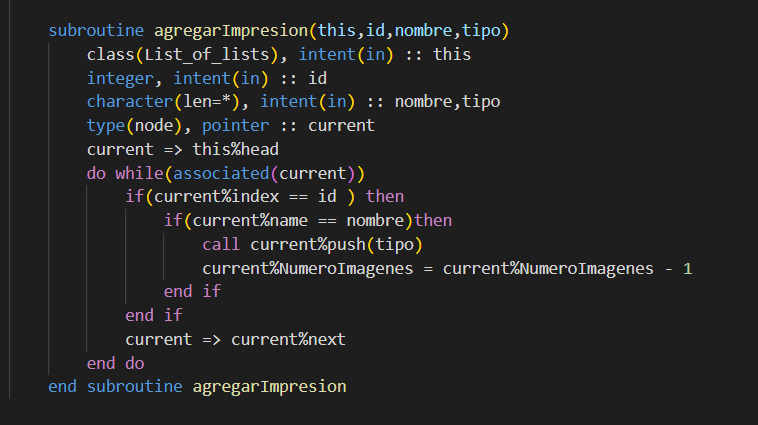
******

La subrutina anterior grafica a un cliente en específico, buscandolo mediante su id, el cual en el programa es único



Subrutina que agrega elementos a la lista de manera doblemente enlazada

***Se utilizo un mismo esqueleto, para la reducción de código.***

******

***Con este método se logro simular la lista de listas en Fortran 90***



Subrutina para actualizar nodos, se utilizó esta lógica para varios métodos, se utilizaron boleanos para poder simular/ver si determinado nodo de lista/cola/pila estaba ocupado, para poder ir ejecutando las acciones requeridas según fuera el caso.



**Se utilizo el IDE Visual studio code**

**Se utilizo la extensión C/C++**

**Modern Fortran**

**Code Runner**

***Para el versiona miento de hizo uso de Git Hub***

***Repositorio del proyecto***

***https://github.com/FranklinONP/EDD\_PROYECTO\_202200089***