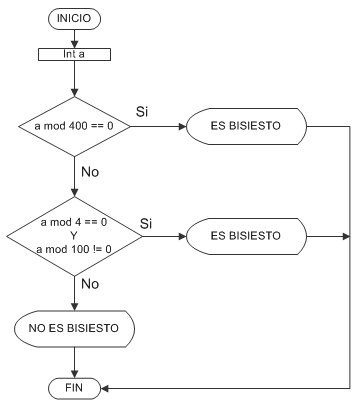
**COMPLEJIDAD CICLOMÁTICA**

*(Cyclomatic Complexity*) es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa. Es una de las métricas de software de mayor aceptación, ya que ha sido concebida para ser independiente del lenguaje.

Esta métrica, propuesta por Thomas McCabe en 1976, se basa en el diagrama de flujo determinado por las estructuras de control de un determinado código. De dicho análisis se puede obtener una medida cuantitativa de la dificultad de crear pruebas automáticas del código y también es una medición orientativa de la fiabilidad del mismo. El resultado obtenido en el cálculo de la complejidad ciclomática define el **número de caminos independientes dentro de un fragmento de código** y determina la cota superior del número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez.



La medida resultante puede ser utilizada en el desarrollo, mantenimiento y reingeniería para estimar el riesgo, costo y estabilidad. Algunos estudios experimentales indican la existencia de distintas relaciones entre la métrica de McCabe y el número de errores existentes en el código fuente, así como el tiempo requerido para encontrar y corregir esos errores.

La complejidad ciclomática puede ser aplicada en varias áreas incluyendo:

* Análisis de **riesgo en desarrollo** de código: Mientras el código está en desarrollo, su complejidad puede ser medida para estimar el riesgo inherente.
* Análisis de **riesgo de cambio** durante la fase de mantenimiento: La complejidad del código tiende a incrementarse a medida que es mantenido durante el tiempo. Midiendo la complejidad antes y después de un cambio propuesto, puede ayudar a decidir cómo minimizar el riesgo del cambio.
* **Planificación de Pruebas**: El análisis matemático ha demostrado que la complejidad ciclomática indica el número exacto de casos de prueba necesarios para probar cada punto de decisión en un programa.
* **Reingeniería**: Provee conocimiento de la estructura del código operacional de un sistema. El riesgo involucrado en la reingeniería de una pieza de código está relacionado con su complejidad.

Tomado de: <https://es.wikipedia.org>

http://formandobits.com/2012/11/complejidad-ciclomatica-nuestro-codigo-es-facil-de-mantener-y-probar/

### SERVICE WORKER DE SERVICIO DE REFERENCIA

Un service Worker hereda todas las limitaciones y los comportamientos disponibles para service Workeres Compartidos de HTML5. Puede crear XMLHttpRequests, usar WebSockets, recibir mensajes de ventanas y del navegador, usar IndexedDB, y publicar mensajes a otras ventanas.

El service Worker puede usar los métodos ononline, onoffline, y navigator.online y las propiedades que están disponibles para todos los service Workeres para obtener una notifiación del estado online/offline de los navegadores.

Además de los métodos estándar, los Trabajdores de Servicio tienen acceso a fucniones adicionales, todas las cuales están implementadas usando mensajes enviados y recibidos por un service Worker "puertos de mensaje". Así como los puertos de mensaje son inherentemente asíncronos, cualquier mensaje que necesite una respuesta involucrará dos mensajes - uno para la pregunta y otro para la respuesta. No todos los mensajes requieren respuesta - esto es parte de la especificación del mensaje. Los mensajes que no requieran una respuesta son análogos a un 'evento' no-solicitado. Si un mensaje si requiere una respuesta, entonces la respuesta siempre deberá ser enviada en el mismo puerto de la pregunta y en general, el 'asunto' de la respuesta será el asunto de la pregunta con "-respuesta" adjunto.

Se espera que los service Workers de Servicio provean una función en un ámbito global, nombrado onconnect. El navegador invocará onconnect en el tiempo de arranque, que dura solo un evento. El service Worker accesa la propiedad ports de este evento para extraer una comunicación estable hacia el puerto del navegador.

Tomado de: https://developer.mozilla.org/es/docs/Social\_API/Service\_worker\_API\_reference