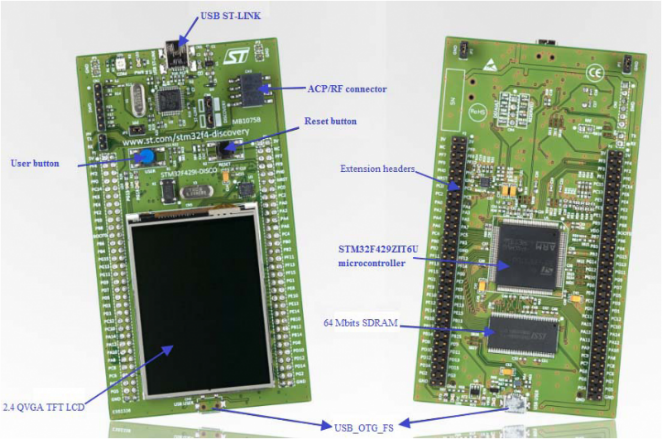
|  |
| --- |
| centro universitario de ciencias exáctas e ingenierías |
| Reglas Misra |
| Tarea 01 COMPLETADA |
|  |
| **Aldo Alexandro Vargas Meza** |



|  |
| --- |
|  |

**Sistema Embebido**

Un sistema embebido (integrado, incrustado) es un sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas, frecuentemente en un sistema de computación en tiempo real. Al contrario de lo que ocurre con los ordenadores de propósito general (como por ejemplo una computadora personal o PC) que están diseñados para cubrir un amplio rango de necesidades, los sistemas embebidos se diseñan para cubrir necesidades específicas.



En un sistema embebido la mayoría de los componentes se encuentran incluidos en la placa base (tarjeta de vídeo, audio, módem, etc.) y muchas veces los dispositivos resultantes no tienen el aspecto de lo que se suele asociar a una computadora. Algunos ejemplos de sistemas embebidos podrían ser dispositivos como un taxímetro, un sistema de control de acceso, la electrónica que controla una máquina expendedora o el sistema de control de una fotocopiadora entre otras múltiples aplicaciones.

**Misra**

MISRA C es un conjunto de recomendaciones o directrices para el desarrollo de software en lenguaje C desarrollado por MISRA (The Motor Industry Software Reliability Association). Su finalidad es proveer portabilidad, seguridad y fiabilidad al código fuente en el contexto de software embebido.

Es aceptado como conjunto de buenas prácticas en sectores como el automovilístico, ferroviario, aeroespacial, telecomunicaciones, equipos médicos y otros. También hay un conjunto de directrices para C++ conocido como MISRA C++. MISRA C no es un estándar abierto; los documentos con las directrices se adquieren mediante compra.

Muchas de las reglas de MISRA C pueden ser efectivamente caracterizadas como recomendaciones o directrices ya que, bajo ciertas condiciones, el desarrollador puede desviarse de las reglas sin dejar su código de ser conforme con el estándar MISRA C. Dichas desviaciones deben ser documentadas o bien en el mismo código o bien en un archivo separado, indicando la regla incumplida y una argumentación al respecto. La desviación respecto a una regla no puede tener un efecto negativo en la seguridad del sistema.

Las reglas se clasifican en tres categorías según su prioridad:

* Reglas/directivas preceptivas (mandatory): La regla debe ser cumplida. No se admiten desviaciones.
* Reglas/directivas requeridas (required): La regla debería ser cumplida. Las desviaciones respecto de la regla deben ser solicitadas mediante un proceso de desviación, deben ser justificadas técnicamente, comprobadas y documentadas.
* Reglas/directivas recomendadas (advisory): La regla debería ser cumplida. Las desviaciones respecto de la regla no deben ser solicitdas formalmente, pero sí deben ser documentadas.

La clasificación de las normas es distinta según se escriba el código de manera manual o de manera automática (Anexo E: Applicability to automatically generated code). La envergadura de las normas a aplicar no varía cuando se evalúa código generado automáticamente, sino que sólo varía el rigor con el que se aplican.

La publicación más reciente de MISRA C contiene 143 reglas que pueden ser agrupadas en la

siguientes categorías.

● Límites de complejidad

● Mejores prácticas

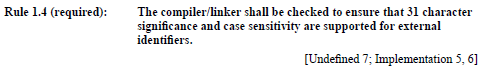
● Producción de código mantenible

● Evitar el uso de funciones propensas a fallar

● Evitar diferencias de ejecución entre compiladores

**Ejemplos de reglas MisraC**

Directivas requeridas:

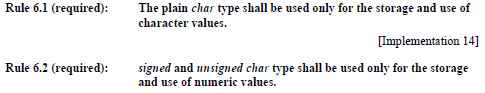




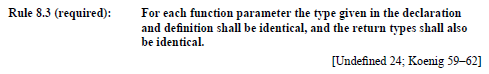


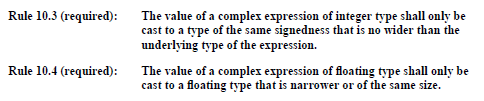
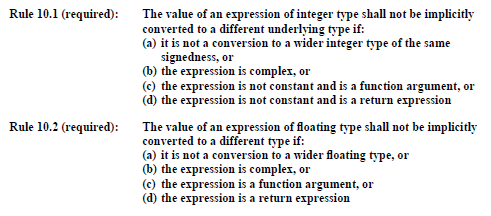




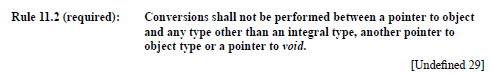


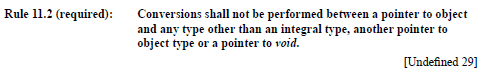












Directivas recomendadas:







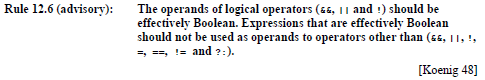


























**Bibliografía:**

* MISRAC 2004 Guidelines for the use of the C language in critical systems.