

COMENTARIO TÉCNICO

# *Buceando en los MCUs Freescale.....*

Por Ing. Daniel Di Lella  
Dedicated Field Application Engineer  
[www.edudevices.com.ar](http://www.edudevices.com.ar)  
[dilella@arnet.com.ar](mailto:dilella@arnet.com.ar)



[www.edudevices.com.ar](http://www.edudevices.com.ar)

*“Serie Flexis”....  
..... “Como migrar de 8 a 32 Bits sin  
traumas”*

**Primera Entrega.**



¡Hola queridos lectores!, como habrán notado, hemos cambiado el título de nuestra sección por *“Buceando en los MCUs Freescale”* debido a que el crecimiento de otras familias de microcontroladores de Freescale Semiconductor ha ido ganando espacio en el mercado y lo justo es, tratar de abarcar las alternativas disponibles en otras muy interesantes familias de microcontroladores de 8 a 32 bits.

Si bien la familia **HC908** es una de las más populares del mercado, otras familias están creciendo a pasos acelerados y van cubriendo “nichos” en donde la familia HC908 no podía llegar. Este es el caso de la familia **HC9S08**, de muy bajo consumo y de prestaciones muy superiores a la HC908. Esta familia “abre la puerta” al mundo de los 32 bits gracias a la nueva “**Serie Flexis**” de 8 / 32 Bits que permite una rápida migración de un proyecto de 8 Bits al mundo de los 32 Bits.

En artículos anteriores, hemos visto las distintas prestaciones y características de algunos derivados de la familia HC9S08, también hemos visto que toda esta familia es **100% código compatible con la HC908**, lo que a la hora de la migración de una a otra, facilita la tarea del diseñador y además garantiza la “reusabilidad” de la mayor parte de las rutinas armadas y probadas para la familia HC908, tema “no menor” para aquellos que sabemos de la importancia de la “confiabilidad” de una rutina y las cientos o miles de horas que se invierten en lograr que un sistema funcione en forma estable.

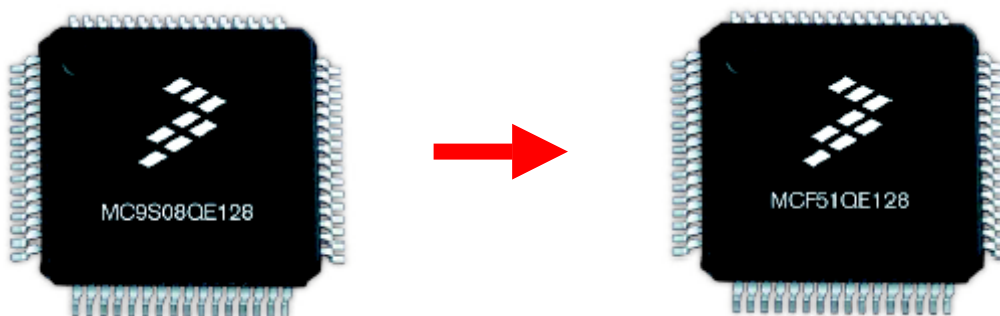
Ahora bien... **¿Que sucede cuando un proyecto en particular no tiene “un techo” de crecimiento?**

En algunos proyectos, es muy difícil prever las prestaciones de este a lo largo del tiempo, aun con el “hardware” totalmente definido, existen aplicaciones que crecen considerablemente debido a que el usuario necesita más y más prestaciones gráficas, de accesibilidad, conectividad con otros productos, etc., etc.

Ante semejante desafío, el mundo de 8 Bits nos queda “chico” cuando nuestra aplicación tiene que recurrir a cálculos complejos, manejo de “stacks” de conectividad, y otras tareas complejas en donde el mundo de 32 Bits puede acometerlas sin siquiera “despeinarse”.

Hasta aquí pareciera que migrar de 8 bits a 32 Bits sería la respuesta lógica e inmediata que dispone el diseñador ante este tipo de proyectos..... Pero... **¿Porqué frecuentemente no es así?**

Lamentablemente, hasta la aparición de la **Serie Flexis**, el diseñador se enfrentaba a una serie de problemas que hacían muy difícil la migración y por ello un proyecto “aguantaba” todo lo posible con soluciones de 8 bits cuando lo razonable era migrar a 32 Bits.



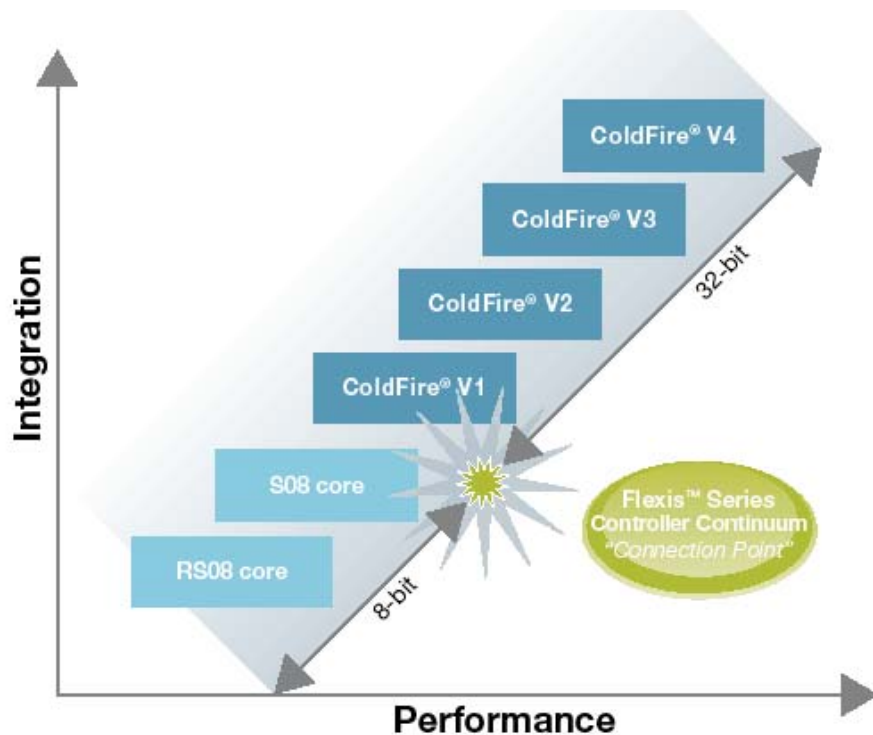
### Se pueden resumir las dificultades de migración en los siguientes puntos:

- Uso de diferentes herramientas de software (entornos) cuando se migra de aplicaciones de 8 Bits a 32 Bits. Esto obliga a invertir un tiempo significativo en el aprendizaje de características y funcionalidades de cada uno de los entornos según cuán diferentes sean uno de otro. Las librerías y derivados pueden necesitar de actualización cuando pasamos de uno a otro mundo.
- Las herramientas de hardware suelen ser diferentes, con diferentes conexiones y con un costo extra que hay que tener en cuenta.
- El cambio en el programa implica el uso de documentación diferente, nombres de registros distintos y periféricos distintos.
- La configuración de la mayoría de los periféricos del mundo de 32 Bits son más complejos de configurar y usar que los del mundo de 8 Bits, lo que acentúa la brecha entre ambos mundos.
- Muchos dispositivos no mantienen compatibilidad entre pines y además se manejan con tensiones distintas, lo que genera un nuevo diseño de la placa de circuito impreso (PCB) y contar con espacio extra para adaptar tensiones de trabajo.
- Modos de consumo muy diferentes pueden encontrarse en ambos mundos, porque son fuertemente dependientes del tipo de arquitectura utilizada en cada uno de ellos.

Todas estas dificultades son las que a menudo pueden hacer que un proyecto pase del “desastre” al “éxito” rotundo, según como cada diseñador pueda manejarse con ellas.

### *La Serie Flexis, un punto de conexión en el “Controller Continuum”...*

Si algo ha caracterizado a los microcontroladores de Freescale (ex Motorola), es que desde sus comienzos existe una línea de pensamiento de hacer sencilla la migración desde una familia a otra, conservando muchas veces compatibilidad de código, arquitecturas de funcionamiento, y manejo de las excepciones, esto ocurre desde las primeras apariciones del microprocesador de 8 Bits MC6800 y del microprocesador de 32 Bit MC68000. La aparición de la nueva Serie Flexis, es el punto de unión en la amplia oferta de soluciones de 8 a 32 Bits que posee hoy la firma, denominada “**Controller Continuum**”, dando la idea de una “continuidad” en la oferta de distintas familias de MCUs, una para cada aplicación.



Un número de factores hacen de la nueva Serie Flexis, el punto “revolucionario” que facilita la migración de uno a otro mundo. Alguno de ellos podemos citarlos a continuación:

- Una cantidad de periféricos comunes entre sí, todos del mundo de 8 bits, garantiza una total compatibilidad de la aplicación en 8 y en 32 Bits, además de facilitar el manejo de dichos periféricos por parte del diseñador acostumbrado al mundo de 8 bits.
- Herramientas de Hardware y Software comunes, que permiten una fácil migración desde los 8 Bits (familia HC9S08) a los 32 Bits (Familia “V1” ColdFire (CFV1)). La misma versión de CodeWarrior para ambos “Cores”.
- Compatibilidad entre pines, lo que permite reemplazar el microcontrolador sin cambiar nada del hardware cuando se crece de 8 a 32 Bits en la aplicación.

Los pilares de la Serie Flexis descansan sobre las bases de las familias **HC9S08** para el mundo de los 8 Bits y la familia **“V1” ColdFire (MCF51)** para el mundo de los 32 Bits.

La familia “V1” ColdFire es una versión adaptada de la familia ColdFire y de su antecesora la MC68000.

La arquitectura del set de instrucciones (ISA) del ColdFire original ha sido modificada en esta nueva versión “V1”, no solo mejorando las prestaciones en operaciones de 32 bits, sino que además expandiendo y mejorando las operaciones con 8 y 16 Bits. Esto es muy importante teniendo en cuenta que de esta forma se optimiza el manejo de las aplicaciones que vienen del mundo de los 8 o 16 Bits, logrando eficiencia de código y mayor velocidad de ejecución de instrucciones que no son típicas en un procesador de 32 Bits.

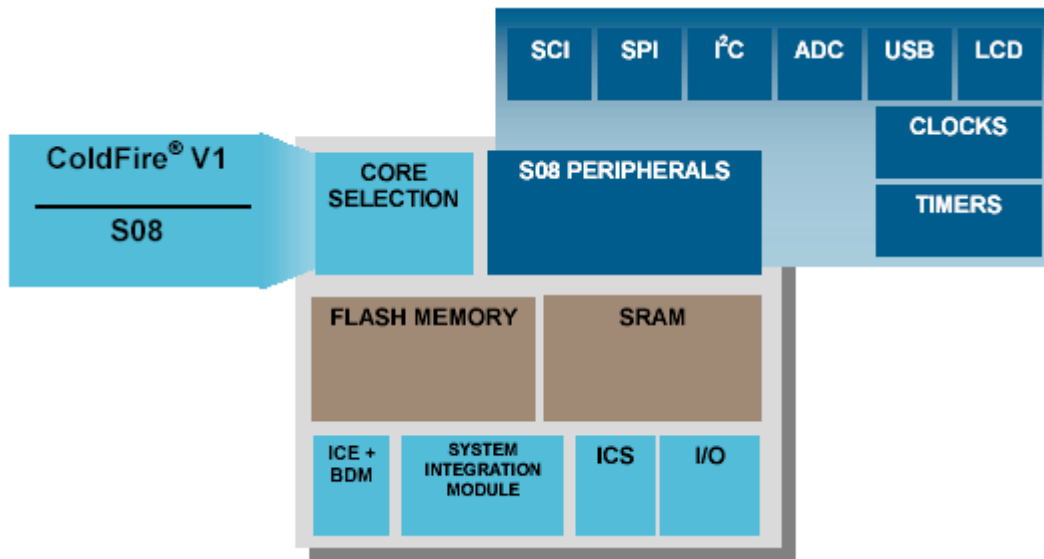
## Pero ¿Cuándo utilizar HC9S08 y cuando ColdFire V1?

Si bien no hay una respuesta única y categórica, podemos citar algunas condiciones que hacen más aconsejable el uso de una u otra familia dentro de la serie flexis en un proyecto en particular. A continuación podemos ver un cuadro con el resumen de las mismas.

Usar HC9S08 cuando:	Usar ambas cuando:	Usar ColdFire V1 cuando:
<ul style="list-style-type: none"><li>• Se requiere muy bajo consumo con mínimo requerimiento de funciones matemáticas.</li><li>• Muy pocos pines en uso o es preferible el encapsulado más pequeño.</li><li>• No hay una exigencia de alta performance o periféricos especiales.</li><li>• El proyecto es muy sensible al costo.</li><li>• En el proyecto hay otros dispositivos de 8 Bits que realizan otras funciones.</li></ul>	<ul style="list-style-type: none"><li>• Se diseña para un amplio portafolio de aplicaciones con distintas prestaciones y precios en un mismo sistema básico.</li></ul>	<ul style="list-style-type: none"><li>• En lo inmediato o en lo cercano se necesita alta capacidad de procesamiento matemático.</li><li>• En lo inmediato / cercano se necesita alta velocidad de comunicación con interfaces del tipo USB o Ethernet.</li><li>• Se tiene pensado en el futuro migrar a aplicaciones con 32 Bits típicas de los modelos V2, V3, V4</li><li>• Se necesita una ruta amigable a memorias Flash superiores a los 128 Kbytes.</li></ul>

### *Metiéndonos en el Hardware....*

Una de las grandes ventajas que posee la Serie Flexis es el hecho de compartir TODO el hardware entorno al “Core” elegido, o sea, solo cambia el “Core” (8 o 32Bits) y el resto permanece sin cambios. Como se había mencionado anteriormente, el hardware en común pertenece al mundo de los 8 Bits (HC9S08), con lo que se gana en facilidad de uso de los periféricos y además permite utilizar herramientas comunes entre la familia HC9S08 tradicional y la nueva Serie Flexis, ya que comparten los circuitos de BDM e ICS, haciendo que cualquier herramienta apta para HC9S08 lo sea para la nueva Serie Flexis. Por ejemplo, solo se necesita el *CodeWarrior 6.x* para trabajar con ambas familias y efectuar la migración en muy pocos pasos. Esta particularidad se puede observar en el diagrama en bloques genérico que se muestra a continuación.



Estructura “típica” común entre las familias HC9S08 y ColdFire V1 en la Serie Flexis.

*Cambio en pocos “Clicks” del “Core” con CodeWarrior 6.x .....*

Gracias al entorno **CodeWarrior 6.x** en común que poseen los “Core” HC9S08 y ColdFire V1, solo se necesita unos pocos “clicks” para pasar de 8 Bits a 32 Bits. Por ejemplo, un proyecto puede pasar de HC9S08 a ColdFire V1 en solo 4 pasos:

1. Elija la opción “Change MCU / Connections” (Globo “1” de la Figura 1).
2. Seleccione el MCU ColdFire V1 correcto para migrar la aplicación (Globo “2”).
3. Seleccione el modo de conexión “P&E Multilink” si es que se está usando dicho BDM (Globo “3”).
4. Haga “Click” en el botón “Finish”.

El entorno CodeWarrior 6.x se encarga de generar y adaptar las librerías cuando se migra de un core a otro. En las figuras 2 y 3 se pueden observar los archivos generados para cada “Core” en forma automática.

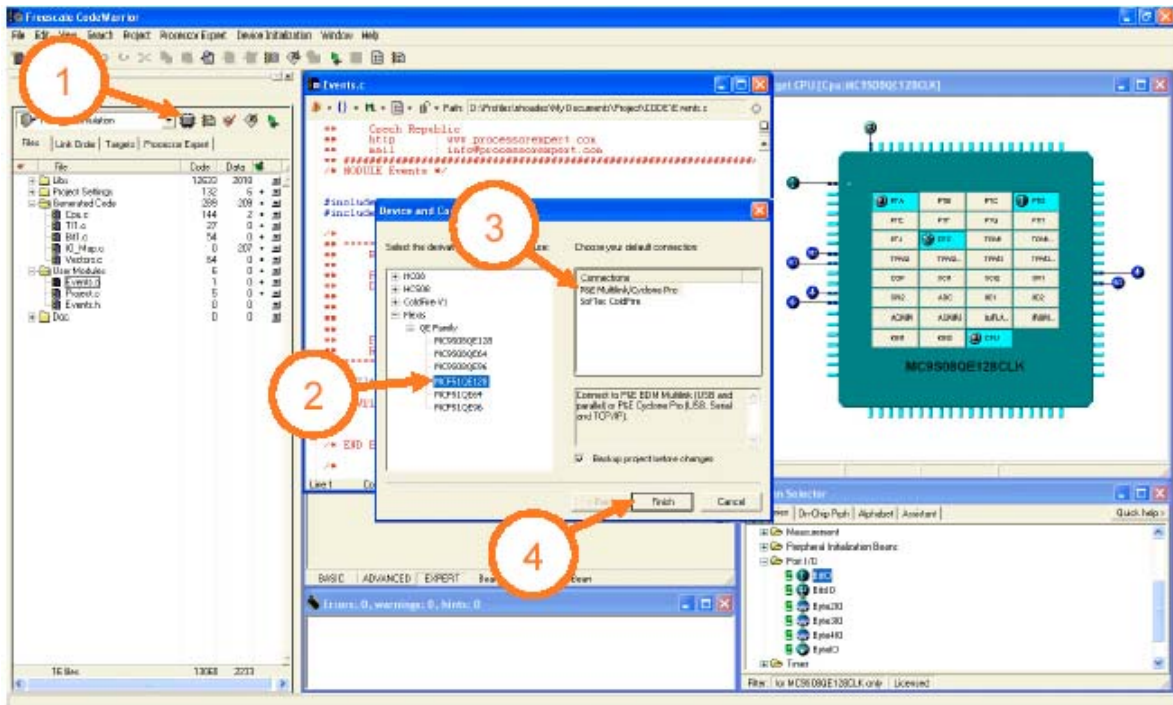


Figura 1.- Migración de “Core” en el entorno “CodeWarrior 6.0”.

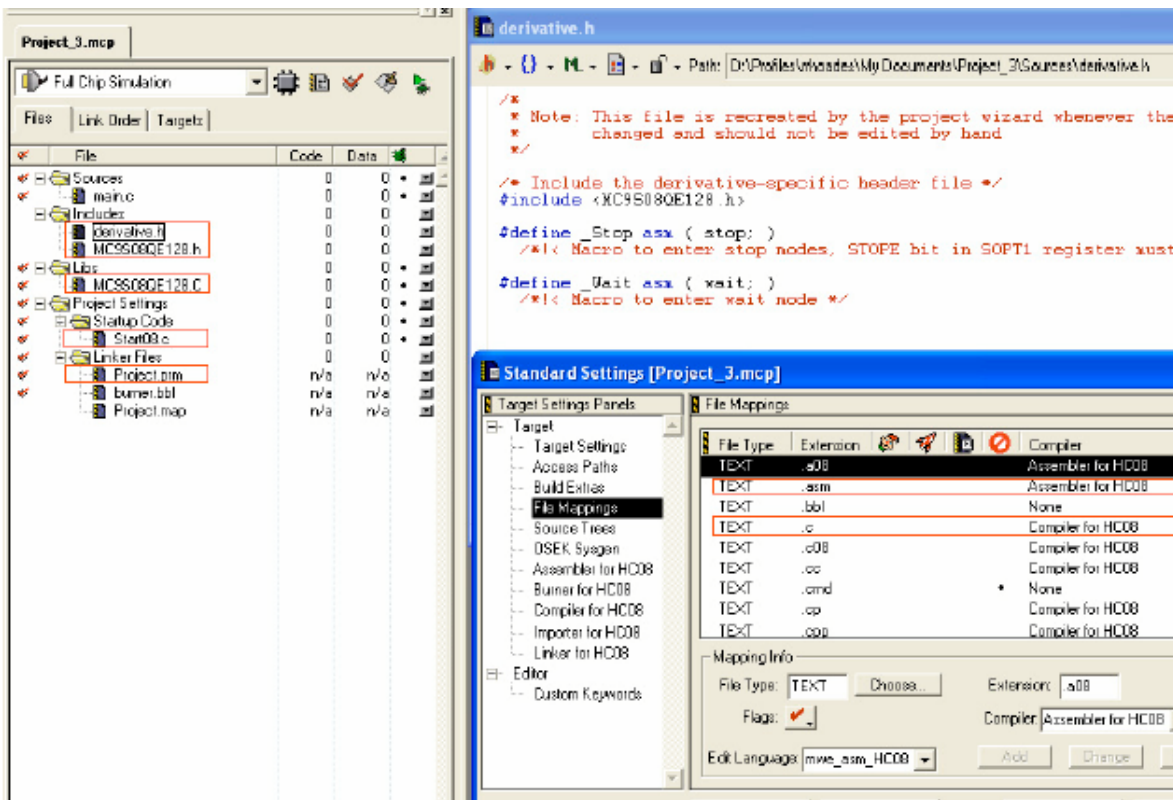


Figura 2.- Archivos de un proyecto básico con HC9S08.



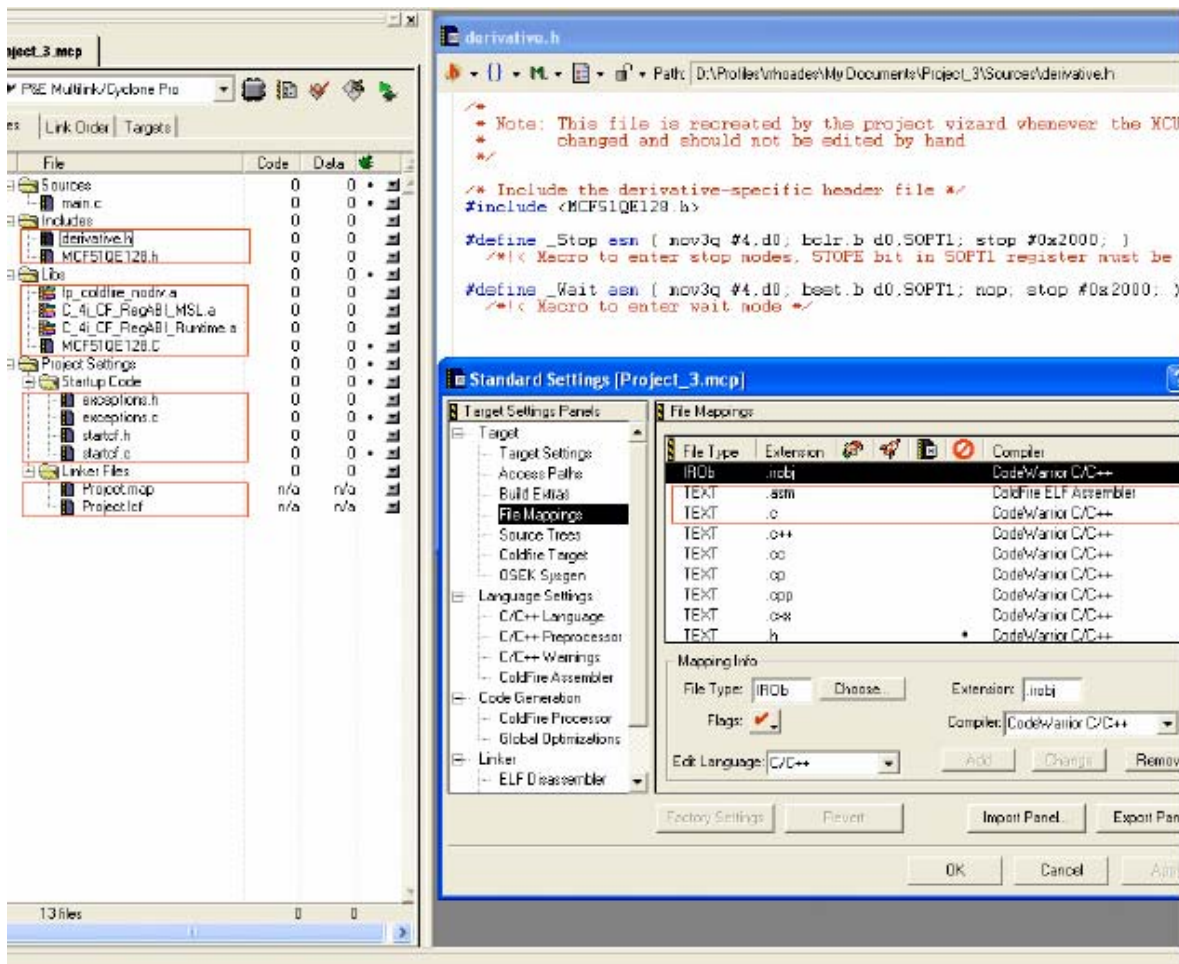


Figura 3.- Archivos de un proyecto básico con ColdFire V1.

*Continuará.....*

*..... Hasta la próxima!!!!*

[www.edudevices.com.ar](http://www.edudevices.com.ar)

