

The background features a large, textured sphere on the left, resembling a planet or moon, with several smaller spheres floating around it. The lower portion of the image shows a grey, metallic-looking surface with geometric shapes and a circular element, possibly representing a console or a base. The overall color palette is muted, with greys, oranges, and purples.

Programación de la Consola Nintendo DS

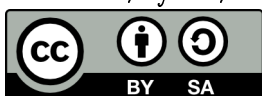
VJ1214 Consolas y dispositivos de videojuegos

Grado en Diseño y Desarrollo de Videojuegos

**Raúl Montoliu Colas
Juan Carlos Fernández Fernández
Maribel Castillo Catalán**

Copyright ©2018 Raúl Montoliu Colas, Juan Carlos Fernández Fernández, Maribel Castillo Catalán.

Esta obra se distribuye bajo la Licencia *Creative Commons Atribución-Compartir Igual 4.0 Internacional*. Puede consultar las condiciones de dicha licencia en: <http://creativecommons.org/licenses/by-sa/4.0/>



Índice general

1	Introducción a la consola Nintendo DS	5
1.1	Las videoconsolas de Nintendo	5
1.2	Hardware de la Nintendo DS (NDS)	6
2	Introducción a la programación de la NDS	9
2.1	Herramientas de desarrollo para programar con la NDS	9
2.2	Instalación del entorno de desarrollo en Windows	11
2.3	Nuestro primer programa para NDS en Windows	13
2.4	Instalación del entorno de desarrollo en Linux	18
2.5	Nuestro primer programa para NDS en Linux	19
	Bibliography	33
	Books	33
	Articles	33

1. Introducción a la consola Nintendo DS

Este capítulo expone en primer lugar el conjunto de consolas de la familia Nintendo y posteriormente describe los principales elementos Hardware de la computadora Nintendo DS, que es la videoconsola a la que se hace referencia el resto de los capítulos de este libro.

1.1 Las videoconsolas de Nintendo

Existe una amplia variedad de consolas de Nintendo:[Barrachina17]

- *GameBoy Advance (GBA)*: tiene un procesador *ARM7TDMI*, de 32 bits, junto con un procesador *Z80*, para dar soporte a los juegos de la *GameBoy* clásica (ver Figura 1.1a).
- *Nintendo DS (NDS)*: tiene un procesador *ARM9* a 66Mhz y un procesador *ARM7* a 33Mhz (ver Figura 1.1b).
- *Nintendo DS Lite*: se diferencia de la *NDS* normal en su aspecto más estilizado, en mejoras de consumo energético y los diferentes niveles de control de brillo de la pantalla (ver Figura 1.2a).
- *Nintendo DSi*: incorpora dos cámaras de baja resolución, pantallas ligeramente mayores, mejor sonido, más memoria y una nueva ranura para tarjetas *Secure Digital (SD)*. A cambio pierde la ranura de compatibilidad con los cartuchos de *GameBoy Advance (Slot2)* (ver Figura 1.2b).
- *Nintendo DSi XL*: conocida también como *DSi XL*, es prácticamente idéntica a la anterior, salvo que su forma es significativamente mayor.
- *Nintendo 3DS*: permite jugar con juegos y ver películas en 3D. Además, la nueva pantalla ofrece imágenes estereoscópicas sin necesidad de gafas especiales para disfrutar del efecto 3D. Incorpora una pantalla táctil, *WiFi*, sensor de movimiento con giroscopio de tres ejes y acelerómetro de tres ejes. (ver Figura 1.3a).
- *Nintendo 2DS*: Conserva las mismas funciones y especificaciones que la *Nintendo 3DS*, salvo que no reproduce los videojuegos con efecto 3D, sino en 2D. Además, mantiene el tamaño de las pantallas de la *Nintendo 3DS* (ver Figura 1.3b)..
- *New Nintendo 3DS*: Esta consola cuenta con botones de colores. Las pantallas de la *New Nintendo 3DS* son 1,2 veces más grandes que las de la *Nintendo 3DS* original, mientras que

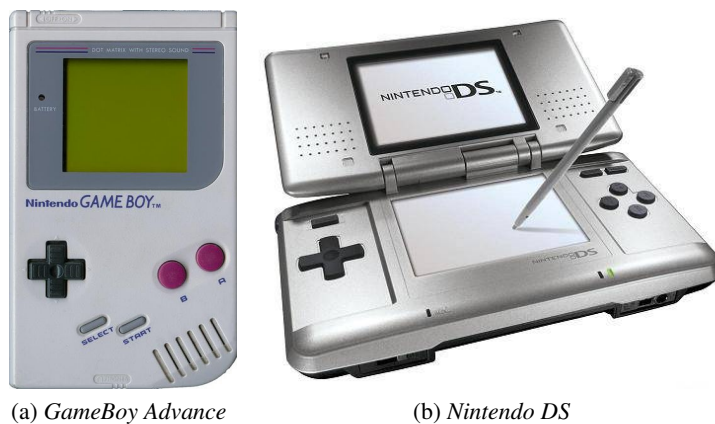


Figura 1.1: Videoconsolas de Nintendo: GBA y NDS



Figura 1.2: Videoconsolas de Nintendo: NDSLite y NDSi

el tamaño de las pantallas de la *New Nintendo 3DS XL* son similares a las de su predecesora. (ver Figura 1.3c). Las ranuras de la tarjeta de juego, del lápiz y del botón de encendido se han trasladado a la base de la consola. Como nuevas características cabe destacar:

- El rastreo facial para que la cámara siga la línea de visión del jugador, de esta forma se amplía la gama de ángulos desde los que se puede ver el efecto 3D estereoscópico del sistema.
- La variación automática del brillo de las pantallas según la iluminación ambiental.
- La transferencia inalámbrica de archivos multimedia entre la consola y un ordenador.
- Una CPU más potente, un ARM11 Dual Core a 532MHz.

1.2 Hardware de la Nintendo DS (NDS)

A continuación se describe el hardware de la *Nintendo DS*:

- **Procesadores:** cuenta con dos procesadores, un ARM9 y un ARM7. El procesador ARM9 se encarga de la lógica principal del programa, mientras que el ARM7, como procesador secundario, se encarga básicamente de gestionar el audio, la red inalámbrica (*WiFi*) y algunas teclas. El hecho de que la consola NDS cuente con dos procesadores implica la generación de dos ejecutables distintos, uno para cada procesador. El ejecutable del ARM7 actúa como esclavo del ARM9, atendiendo peticiones de reproducción de sonido o comunicaciones vía



Figura 1.3: Videoconsolas de Nintendo: N3DS, N2DS y NNew3DS

WiFi.

■ **Memorias:**

- **Memoria principal:** tiene un tamaño de 4 MB. Dicha memoria almacena el ejecutable para el ARM9, así como la gran mayoría de datos del ejecutable. Ambos procesadores pueden acceder a esta memoria en cualquier momento. Si ambos intentan acceder a la vez, será el que tenga mayor prioridad el que accede, quedando el otro a la espera.
- **Memoria de vídeo VRAM:** tiene 656 KB distribuidos en 9 bancos de memoria de vídeo, que se pueden usar con diferentes propósitos. A lo largo de las sesiones de prácticas se verán más detalles de esta memoria de vídeo.
- **Otras memorias:** tiene las pseudo-cachés *WRAM* e *IWRAM* de 96Kb, una memoria *RAM* adicional para la *BIOS* y una memoria virtual para vídeo (*Virtual Video RAM*).

- **Gráficos:** el hardware de vídeo se compone de dos núcleos gráficos 2D, uno principal (*main*) y otro secundario (*sub*). Dichos núcleos se diferencian únicamente en que el motor principal puede *renderizar* tanto la memoria de vídeo virtual sin utilizar el motor 2D, como mapas de bits de 256 colores, así como utilizar el motor 3D para el renderizado de alguno de sus fondos.

- **Sonido:** dispone de altavoces estéreo y cuenta con 16 canales de audio independientes.

- **Comunicación inalámbrica:** soporta el estándar de protocolo de comunicaciones *IEEE 802.11*. El rango de comunicación inalámbrica varía de 10 a 30 metros, dependiendo de las circunstancias.

- **Entrada/Salida:** tiene un puerto para cartuchos de juegos de Nintendo DS y otro para juegos de Game Boy Advance2. La NDS cuenta con una entrada para auriculares estéreo y otra entrada para micrófono.

- **Doble pantalla:** las dos pantallas LCD son de 3 pulgadas. La pantalla inferior emplea tecnología táctil.

- **Temporizador:** cuenta con un reloj de tiempo de real, que puede ser utilizado por una aplicación o juego para definir diferentes respuestas dependiendo de la hora del día.

2. Introducción a la programación de la NDS

Este capítulo está dedicado a la instalación de las herramientas necesarias para poder realizar videojuegos en la consola Nintendo DS. Así mismo se verá como poder realizar un primer programa *Hello World*.

2.1 Herramientas de desarrollo para programar con la NDS

2.1.1 Introducción

Se denomina *homebrew* al software *casero* no oficial realizado por programadores, ya sean aficionados o expertos, para cualquier plataforma. Generalmente, esta plataforma suele ser una videoconsola propietaria. El desarrollo de software *casero* está permitido en cualquiera de las consolas de Nintendo, siempre y cuando sea sin ánimo de lucro. En cualquier caso, se debe señalar que no todas las plataformas permiten el *homebrew*. El desarrollo de software para la Nintendo DS se puede realizar de dos maneras diferentes:

- Utilizando el kit comercial de desarrollo de software (*SDK*) de Nintendo.
- Utilizando *DevkitPro*, que es un conjunto de bibliotecas, compiladores y utilidades para desarrollar software para varias plataformas. Además, es libre y de descarga gratuita.

En los apartados siguientes de esta sección se presentarán las principales herramientas existentes que ayudan al desarrollo de aplicaciones para NDS.

2.1.2 DevkitPro

DevkitPro es un conjunto de bibliotecas, compiladores y utilidades que permiten desarrollar aplicaciones para las consolas *Game Boy Advance (GBA)*, *GP32*, *GP2X*, *Playstation Portable (PSP)*, *Nintendo DS* y *GameCube*. *DevkitPro* cuenta con cuatro *toolchains* que permiten escribir aplicaciones y juegos para las consolas citadas:

- *DevkitARM*: utilizado para el desarrollo de aplicaciones para *GBA*, *GP32* y *Nintendo DS*.
- *DevkitGP2X*: utilizado para el desarrollo de aplicaciones para la *GamePark GP2X*.
- *DevkitPPC*: utilizado para el desarrollo de aplicaciones para la *Nintendo GameCube*.
- *DevkitPSP*: utilizado para el desarrollo de aplicaciones para la *Sony PSP*.

2.1.3 DevkitARM

DevkitARM es un *toolchain* de los lenguajes *C* y *C++*, basado en la colección de compiladores *GNU (GCC)*, que permite crear binarios para la *arquitectura ARM*. Incluye todo lo necesario para crear software para la *Nintendo DS*, *GBA* y *GP32*. Las bibliotecas que incluye *DevkitARM* son las siguientes:

- *LibNDS*: anteriormente conocida como *NDSL*, es una biblioteca creada por Michael Noland y Jason Rogers. Esta biblioteca sirve como base para el desarrollo de programas para la *Nintendo DS*. *LibNDS* soporta casi todas las características de la *DS*, incluyendo la pantalla táctil, el micrófono, el hardware 2D, el hardware 3D y las comunicaciones inalámbricas.
- *LibFAT*: contiene una serie de rutinas para leer y escribir en sistemas de ficheros *FAT (File Allocation Table)* como los de las tarjetas *Secure Digital (SD)*, *MultimediaCard (MMC)* o *CompactFlash (CF)*.
- *DSWifi*: permite a los desarrolladores usar la *WiFi* de la *NDS* de una manera similar a como los ordenadores usan la tarjeta de red inalámbrica.
- *LibGBA*: contiene las funciones necesarias para controlar el hardware de la *Game Boy Advance*.

Algunas de las herramientas más destacadas de *DevkitARM* son las siguientes:

- *Grit (GBA Image Transmogifier)*: es un conversor de imágenes para la *Game Boy Advance* y la *Nintendo DS*. *Grit* acepta multitud de formatos de archivos (*bmp*, *pcx*, *png*, *gif*, *jpeg*, ...) con cualquier profundidad de bits y obtiene los datos para ser usados directamente en el código de un programa para *GBA* o *NDS*. Los datos que genera *Grit* pueden ser datos de una paleta, datos de teselas, datos de un mapa o datos de un gráfico. Los formatos de salida disponibles son, entre otros, archivo *C*, archivo binario o archivo *GNU Assembly*. Esta herramienta se empleará más adelante cuando se estudie la parte gráfica de la *NDS*.
- *arm-eabi-gcc*: es un compilador cruzado que genera código objeto para el *ARM7* y el *ARM9* a partir de código escrito en los lenguajes *C* o *C++*.
- *arm-eabi-ld*: es un enlazador que genera un archivo ejecutable en el formato estándar *ELF* para el entorno de ejecución *ARM7* y *ARM9* a partir del código objeto generado por *arm-eabi-gcc*.
- *arm-eabi-objcopy*: es una herramienta que genera los archivos ejecutables reducidos *.arm7* y *.arm9* a partir del archivo ejecutable con formato *ELF*. Esta herramienta reduce al mínimo las necesidades de memoria de la videoconsola. Para ello, extrae exclusivamente lo necesario para poder ejecutar el programa (instrucciones y datos).
- *ndstool*: combina los archivos ejecutables *.arm7* y *.arm9* en un único archivo con extensión *.nds* añadiendo una cabecera descriptiva al comienzo. Opcionalmente, puede combinar junto con los archivos ejecutables otros datos como, por ejemplo, datos de gráficos.
- *dsbuild*: genera un archivo con extensión *.ds.gba*, que permite arrancar el programa desde el *Slot2* (compatible con *Game Boy Advance*).

2.1.4 Entornos de desarrollo

Se puede definir un *IDE (Integrated Development Environment)* como un programa compuesto por un conjunto de herramientas útiles para un desarrollador de software. Como elementos básicos, un *IDE* cuenta con un editor de código, un compilador/intérprete y un depurador. También puede dar soporte a más de un lenguaje de programación.

Para desarrollar programas para la *NDS* se tienen las siguientes opciones:

- Cualquier entorno de desarrollo en *C/C++* es válido para desarrollar programas para la *NDS*, pero suelen requerir dedicar tiempo a configurar tanto los compiladores como los ajustes necesarios de cada proyecto individual.
- Emplear un *IDE* pensado específicamente para el desarrollo en *Nintendo DS*. Por ejemplo,

Eclipse Ganymede dispone de un *plugin NDS*.

2.1.5 Emuladores

Un *emulador* es un programa que se ejecuta en un computador (sistema anfitrión del emulador) y se encarga de recrear el comportamiento de un computador diferente (sistema objetivo del emulador). La ventaja de utilizar un emulador de NDS es que no se necesita tener ni videoconsola ni cartuchos especiales. Sin embargo, las funcionalidades de la NDS que se soportan dependen del emulador utilizado.

WinDS Pro es un pack de emuladores para la NDS. En concreto dispone de los siguientes:

- Citra: emulador de Nintendo 3DS
- DeSmuME: emulador de Nintendo DS
- No\$gba: emulador de Nintendo DS y Game Boy Advance
- VBA: emulador de Game Boy, Game Boy Color y Game Boy Advance

2.2 Instalación del entorno de desarrollo en Windows

Se puede encontrar información sobre el proceso de instalación en la siguiente página web:

http://snipah.com/index.php?option=com_content&view=article&id=44&Itemid=53

2.2.1 Instalación de *devkitpro*

Se accede a la página web:

<http://devkitpro.org/>

Se pulsa en *For instructions on installing the toolchains see our Getting Started pages* y posteriormente elegir *Windows Installer/Updater package* y descargar la última versión (p. ej. *devkitProUpdater-1.6.0.exe*).

2.2.2 Instalación de *WinDS Pro*

Se puede descargar la última versión de *WinDS Pro* de la siguiente página web:

<https://windsprocentral.blogspot.com.es/2016/10/winds-pro.html>

2.2.3 Instalación de *eclipse* con el *plugin de NDS*

Existen dos posibilidades:

1. Instalar el *eclipse* que ya contiene el *plugin de NDS*.
2. Instalar *eclipse* y después añadir el *plugin NDS ManagedBuilder*.

A continuación se describen los pasos a seguir para ambas opciones.

Instalar *eclipse* que ya contiene el *plugin de NDS*

En la página web indicada al comienzo de esta sección (*Instalación del entorno de desarrollo en Windows*), en concreto en *Full Eclipse packages* se pulsa en *Eclipse Full Package Win32 - Zip-Format* para descargar el fichero

`eclipse-cpp-ganymede-win32_nds.zip`

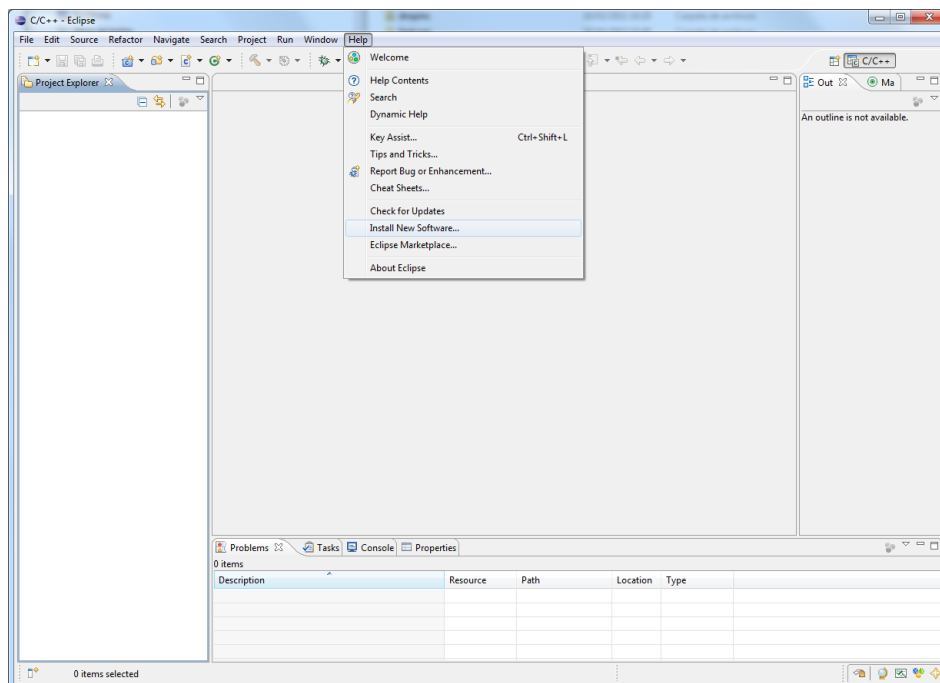


Figura 2.1: Instalación del plugin NDS en eclipse (parte 1).

Instalar *eclipse* y después añadir el plugin *NDS ManagedBuilder*

En este caso el primer paso consiste en la instalación de *eclipse*. Por cuestiones de compatibilidad se escoge la versión *Helios*. Se accede a la página web:

<http://www.eclipse.org/downloads/packages/release/Helios/R>

y se descarga la versión *Eclipse IDE for C/C++ Developers* para el sistema operativo apropiado a las necesidades del equipo con el que se va a trabajar. Una vez descomprimido el fichero ya se tiene instalado *eclipse*. El siguiente paso es instalar el plugin mediante el actualizador del propio *eclipse* que se encuentra en *Help->Install New Software*, tal y como muestra la Figura 2.1.

A continuación en la ventana *Available Software* se pulsa el botón *Add*, e introduce la siguiente información:

- Name: *NDS Manager builder*
- Location: *http://dev.snipah.com/nds/updater*

Esta operación se refleja en la Figura 2.2.

Para que el proceso se realice de forma adecuada hay que tener la precaución de desactivar la opción *Group items by category*. De esta forma aparecerá el software buscado, debiendo activarse las casillas correspondientes a *devkitARM*, tal y como muestra la siguiente Figura 2.3.

Después de pulsar en sucesivos botones *Next* y aceptar la licencia, comienza la instalación del software. Durante dicho proceso puede aparecer la Figura 2.4.

Simplemente se pulsa en *OK* para continuar el proceso de instalación. Una vez finalizada la instalación se debe reiniciar *eclipse*.

Como comprobación de que todo ha ido correctamente, a la hora de crear el proyecto se debe observar que aparece algo parecido a lo mostrado en la Figura 2.5.

Se elige *Empty Project (libnds)*, y si después de pulsar en *Next* aparece lo mostrado en la Figura 2.6, entonces todo está correcto.

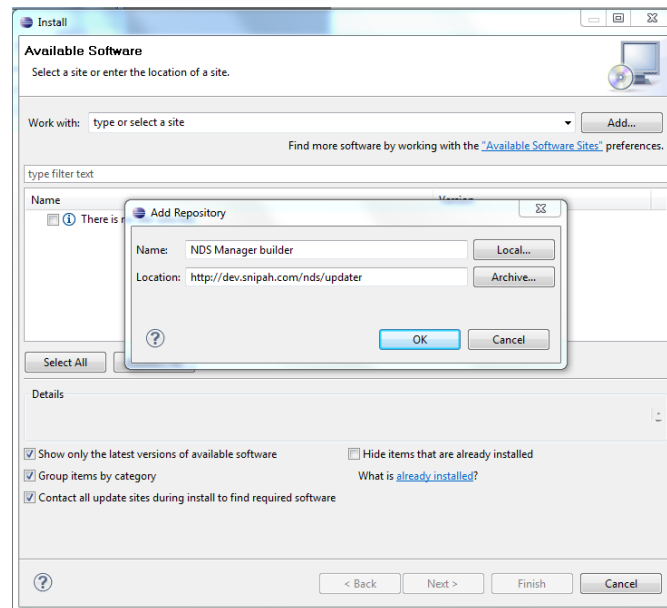


Figura 2.2: Instalación del plugin NDS en eclipse (parte 2).

2.3 Nuestro primer programa para NDS en Windows

En esta sección se van a ver los pasos para realizar nuestro primer programa para la NDS en el sistema operativo *Windows*.

2.3.1 Desarrollar código para la NDS sin emplear Eclipse

En este apartado se va a crear el primer programa en la NDS sin emplear *Eclipse* como herramienta de desarrollo.

Creación de la estructura de ficheros

Se puede emplear como punto de partida el ejemplo *hello_world* que aparece en el directorio *nds* del directorio *examples* de *DevkitPro*. En el laboratorio de prácticas, *DevkitPro* se encuentra en el directorio *C:*. En el directorio donde se vayan a almacenar los programas a desarrollar se crea un nuevo directorio que identifique el programa a desarrollar (p.ej. *ejemplo*). Dentro de ese directorio se crea el directorio *source*, que contendrá los ficheros necesarios para el código a desarrollar (p.ej. *main.c*). En el directorio *ejemplo* se copia el fichero *Makefile* del ejemplo *hello_world* de *DevkitPro*. De esta forma la estructura de ficheros que se tiene es la siguiente:

```
c:\mis_ejemplos
- directorio ejemplo
- fichero Makefile
- directorio source
- fichero main.c
```

Edición del fichero ejemplo

Para familiarizarse con el entorno de desarrollo de aplicaciones para Nintendo DS, se va a utilizar como ejemplo una aplicación en la que aparezca un saludo con el nombre del desarrollador del programa. Para escribir este código se puede emplear cualquier editor de texto. Según esto, el código del programa a desarrollar (*main.c*) es el siguiente:

```
1 #include <nds.h>
2 #include <stdio.h>
3 int main(void) {
```

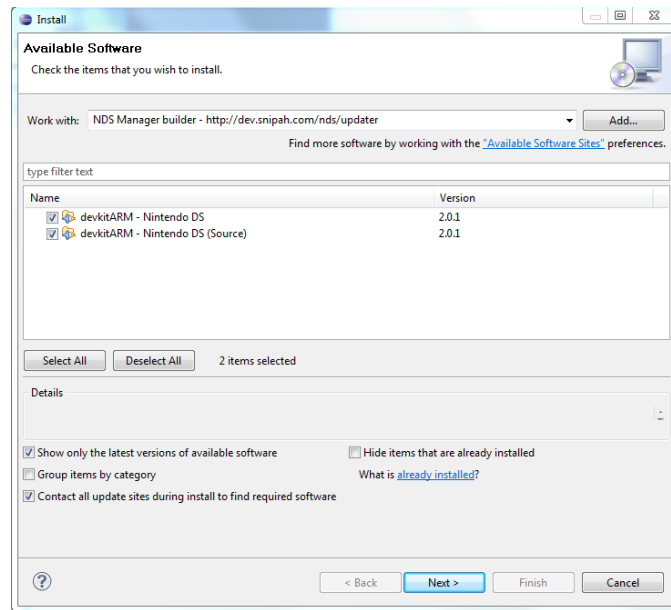


Figura 2.3: Instalación del plugin NDS en eclipse (parte 3).

```

4 consoleDemoInit();
5 iprintf("Hola Juan"); // Imprimir el mensaje
6 while(1) {} // Bucle que no hace nada.
7 }

```

En dicho código cabe destacar lo siguiente:

- `consoleDemoInit`: inicializa una consola de texto predeterminada, sin permitir elegir la pantalla donde se imprime el texto. En este caso, será la pantalla inferior de la videoconsola. Se verán más detalles de cómo seleccionar la pantalla en la que se visualiza información en prácticas posteriores.
- `iprintf("Hola Juan")`: esta función imprime texto con formato, soportando solo números enteros.

Compilación del fichero ejemplo

El siguiente paso es compilar el programa, para ello se abre el *símbolo del sistema*. Una vez se está en el directorio *ejemplo* creado, se ejecuta el comando *make*:

```

C:\mis_ejemplos\ejemplo>dir
29/07/2013 15:10 <DIR> .
29/07/2013 15:10 <DIR> ..
02/04/2012 22:02 4.903 Makefile
29/07/2013 15:10 <DIR> source
                1 archivos      4.903 bytes
                3 dirs 51.779.096.576 bytes libres

C:\mis_ejemplos\ejemplo>make
main.c
arm-none-eabi-gcc -MMD -MP -MF /d/mis_ejemplos/ejemplo/build/main.d -g -Wall
-O2 -march=armv5te -mtune=arm946e-s -fomit-frame-pointer -ffast-ma
th -mthumb -mthumb-interwork -I/d/mis_ejemplos/ejemplo/include
-I/d/mis_ejemplos/ejemplo/build -I/c/devkitPro/libnds/include
-I/d/mis_ejemplos/ejemplo/build -DARM9 -c /d/mis_ejemplos/ejemplo/source/main.c
-o main.o
linking ejemplo.elf
Nintendo DS rom tool 1.50.1 - Jun 19 2012
by Rafael Vuijk, Dave Murphy, Alexei Karpenko

```

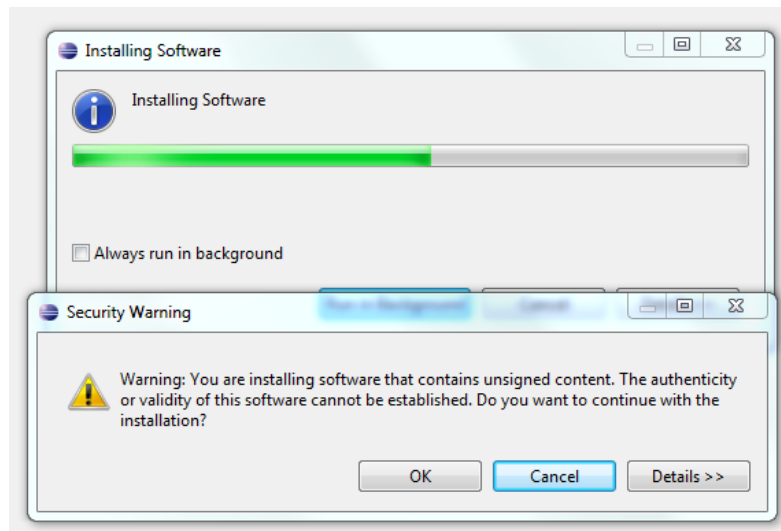


Figura 2.4: Instalación del plugin NDS en eclipse (parte 4).

```
built ... ejemplo.nds
```

Si no se han producido errores de compilación aparecerán los ficheros *ejemplo.elf* y *ejemplo.nds*.

```
C:\mis_ejemplos\ejemplo>dir
29/07/2013  15:15    <DIR>        .
29/07/2013  15:15    <DIR>        ..
29/07/2013  15:15    <DIR>        build
29/07/2013  15:15               234.949 ejemplo.elf
29/07/2013  15:15               134.208 ejemplo.nds
02/04/2012  22:02                4.903 Makefile
29/07/2013  15:10    <DIR>        source
                3 archivos      374.060 bytes
                4 dirs  51.778.568.192 bytes libres
```

El primero (*ejemplo.elf*) es el que contiene la información de depuración, por tanto, el depurador tendrá que trabajar necesariamente con él. Este fichero se empleará en prácticas sucesivas. Sin embargo, la consola (o el emulador) solo será capaz de ejecutar la imagen del cartucho *ejemplo.nds*. También se ha creado el directorio *build*, que por ahora no tiene interés para lo que se está desarrollando. Para borrar todos los ficheros y directorios creados durante la compilación se puede ejecutar *make clean*.

Si se produjese un error relacionado con que no encuentra el compilador se puede realizar lo siguiente:

- Hacer una copia de los siguientes ficheros que se encuentran en *C:\devkitPro\devkitARM\bin*:
 - arm-none-eabi-as
 - arm-none-eabi-g++
 - arm-none-eabi-gcc
 - arm-none-eabi-gdb
 - arm-none-eabi-objcopy
- Renombrar las copias con los siguientes nombres:
 - arm-eabi-as
 - arm-eabi-g++
 - arm-eabi-gcc
 - arm-eabi-gdb

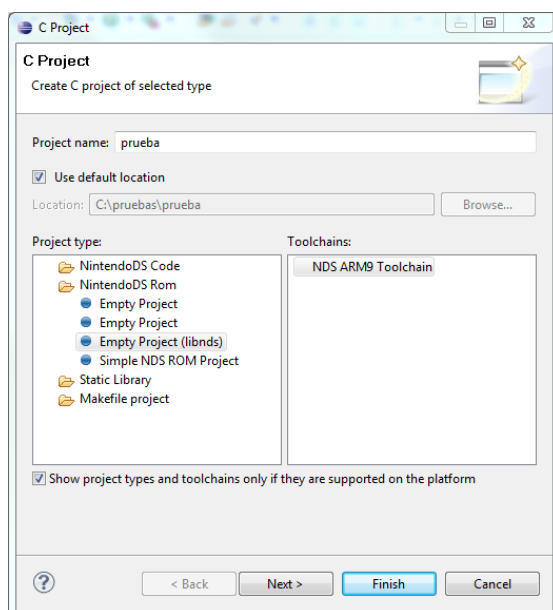


Figura 2.5: Instalación del plugin NDS en eclipse (parte 5).

- arm-eabi-objcopy

Ejecución del fichero ejemplo en el emulador

Si no se ha producido ningún problema en la compilación, la salida del programa se puede ver en el emulador. Una vez abierto *WinDS Pro*, si se escoge el emulador *No\$gba*, se pulsa en *File->Cartridge Menu (File Name)* y se busca el fichero *.nds* que nos interesa. En nuestro caso, y una vez elegido *ejemplo.nds* aparece la ventana en el emulador mostrada en la figura 2.7:

Si se escoge el emulador *DeSmuME*, se pulsa en *File->Open ROM* y se busca el fichero *.nds* que nos interesa.

2.3.2 Desarrollar código para la NDS empleando Eclipse

En este apartado se va a crear el primer programa en la NDS empleando *Eclipse* como herramienta de desarrollo.

Creación de un proyecto para NDS

En el laboratorio de prácticas, *Eclipse* se encuentra en el directorio *C:*. Al iniciar *Eclipse* se pide el directorio donde se almacenará el proyecto a crear, tal como muestra la Figura 2.8.

Para crear un nuevo proyecto se debe pulsar en la ventana principal de *Eclipse* en *File->New->C project* (ver Figura 2.9).

Aparece la ventana (*C Project*) mostrada en la Figura 2.10. en la que se debe configurar lo siguiente:

- El nombre del proyecto (p.ej. *ejemplo*).
- En *Project types* se selecciona *Nintendo DS Rom->Empty Project (libnds)*.
- Se pulsa en *Next*.

Aparece una ventana de *Select Configurations* en la que se pulsa en *Finish*.

Una vez creado el proyecto, si lo que aparece es la siguiente ventana se debe elegir el icono de *workbench*, tal y como se muestra en la Figura 2.11.

De esta forma, el proyecto creado aparecerá en la ventana de *workbench* de *Eclipse* (ver Figura 2.12):

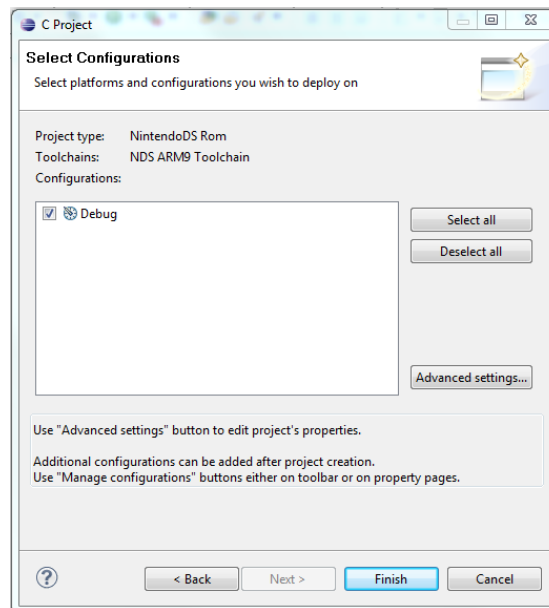


Figura 2.6: Instalación del plugin NDS en eclipse (parte 6).

Esta ventana podría aparecer directamente sin necesidad de elegir el icono de *workbench* de *Eclipse*.

Configuración del proyecto

Para iniciar la configuración del proyecto, se debe pulsar en la ventana principal de *Eclipse* en *Project Properties*, teniendo la precaución de tener activado el proyecto que se desea. Una vez realizada esta operación aparece la ventana (*Properties for ejemplo*) (ver Figura 2.13).

En dicha ventana se debe desplegar la pestaña *C/C++ Build*. Se resalta dicha opción. En el cuadro *Builder* de la pestaña *Builder Settings* se cambia la opción *Builder Type* a *Internal Builder* (ver Figura 2.14).

En la pestaña *C/C++ Build* se resalta *Settings*, apareciendo la ventana que se muestra en la Figura 2.15.

Se escoge *devkitARM C Linker->ARM* y se desactiva la opción *No FPU*. Finalmente se pulsa en *OK* (ver Figura 2.16).

Edición del fichero ejemplo

Para familiarizarse con el entorno de desarrollo de *Eclipse* para NDS, se utiliza el mismo ejemplo que el del apartado anterior. En primer lugar, se debe crear un fichero fuente en *lenguaje C* dentro del proyecto actual. Para ello se pulsa en *File->New->Source File* (ver Figura 2.17).

Aparece una nueva ventana (*New Source File*) (ver Figura 2.18).
en la que se debe realizar lo siguiente:

- En *Source File* se introduce *main.c*.
- En *Template* se selecciona *None*.
- Se pulsa en *Finish*.

A continuación en la ventana que hace referencia a *main.c* se introduce el mismo código que el del apartado 2.3.2.

Compilación del fichero ejemplo

El siguiente paso es compilar el programa, para ello se elige *Project->Build Project*. Si no se han producido errores de compilación aparecerán los ficheros *ejemplo.elf* y *ejemplo.nds* en el directorio *Debug*, tal y como se puede comprobar en la Figura 2.19.



Figura 2.7: Ejecución del programa ejemplo en el emulador *No\$gba*.

Si se produjese un error relacionado con que no encuentra el compilador se puede realizar lo siguiente:

- Hacer una copia de los siguientes ficheros que se encuentran en *C:\devkitPro\devkitARM\bin*:
 - arm-none-eabi-as
 - arm-none-eabi-g++
 - arm-none-eabi-gcc
 - arm-none-eabi-gdb
 - arm-none-eabi-objcopy
- Renombrar las copias con los siguientes nombres:
 - arm-eabi-as
 - arm-eabi-g++
 - arm-eabi-gcc
 - arm-eabi-gdb
 - arm-eabi-objcopy

Ejecución del fichero ejemplo en el emulador

Una vez abierto *WinDS Pro*, si se escoge el emulador *No\$gba*, se pulsa en *File->Cartridge Menu (File Name)* y se busca el fichero *.nds* que nos interesa. En nuestro caso, y una vez elegido *ejemplo.nds* aparece la ventana en el emulador mostrada en la Figura 2.7.

Si se escoge el emulador *DeSmuME*, se pulsa en *File->Open ROM* y se busca el fichero *.nds* que nos interesa.

2.4 Instalación del entorno de desarrollo en Linux

Las operaciones a seguir para instalar el entorno de desarrollo en Linux son las siguientes:

1. Instalación de *devkitpro*. Se accede a la página web:

<http://devkitpro.org/>



Figura 2.8: Creación de un proyecto para NDS usando Eclipse (parte 1).

Se pulsa en *For instructions on installing the toolchains see our Getting Started pages* y posteriormente elegir *Manual instructions for installing devkitARM* y seguir los pasos que aparecen en dicha página web.

2. Instalación de *WinDS Pro*. Se puede descargar la última versión de *WinDS Pro* de la siguiente página web:

<https://windsprocentral.blogspot.com.es/2016/10/winds-pro.html>

3. Instalación de *desmume*. Se pueden seguir los pasos que aparecen en la siguiente página web:

http://wiki.desmume.org/index.php?title=Installing_DeSmuME_from_source_on_Linux

Recomendable emplear la opción *Install desmume from svn*.

4. Instalación de *eclipse Helios*. Se siguen los mismos pasos que los indicados para Windows.
5. Instalación del *plugin NDS ManagedBuilder*. En este caso será necesario instalar el plugin mediante el actualizador del propio Eclipse que se encuentra en *Help->Install New Software*. Para ello se emplea la *url* de actualizaciones del *NDS Managed builder* (<http://dev.snipah.com/nds/updater/>), que se deberá especificar en la ventana mostrada en la Figura 2.20.

Hay que tener la precaución de desactivar la opción *Group items by category*. Después de aceptar la licencia se instalará el plugin. Una vez finalizada la instalación se debe reiniciar Eclipse.

2.5 Nuestro primer programa para NDS en Linux

En esta sección se van a ver los pasos para realizar nuestro primer programa para la NDS en el sistema operativo *Linux*. **En el laboratorio se debe entrar con la cuenta *usuario*.** Antes de nada se debe comprobar que las siguientes variables de entorno se encuentran en el fichero *.bash_profile* del *usuario*:

```
export DEVKITPRO=/opt/devkitpro/
export DEVKITARM=/opt/devkitpro/devkitARM/
```

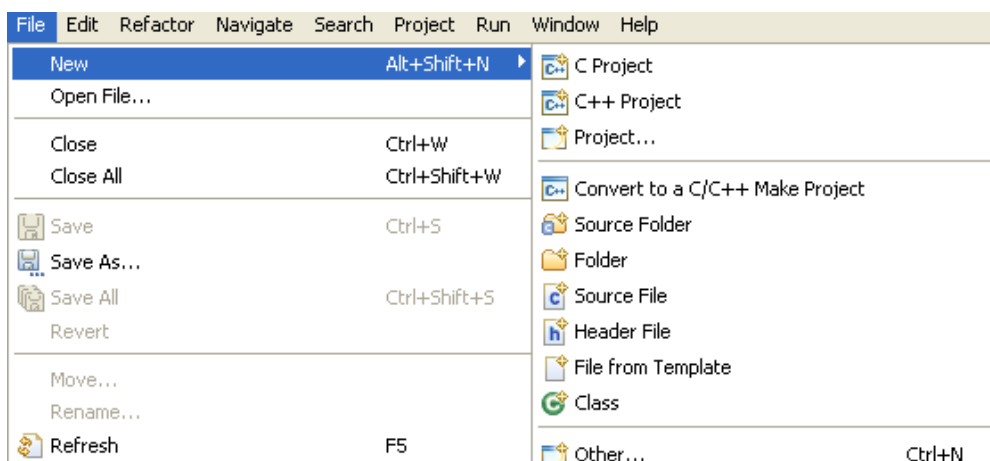


Figura 2.9: Creación de un proyecto para NDS usando Eclipse (parte 2).

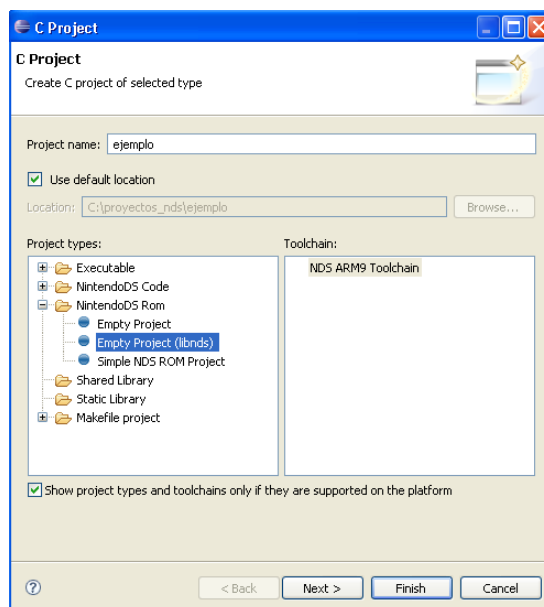


Figura 2.10: Creación de un proyecto para NDS usando Eclipse (parte 3).

2.5.1 Desarrollar código para la NDS sin emplear Eclipse

En este apartado se va a crear el primer programa en la NDS sin emplear *Eclipse* como herramienta de desarrollo.

Creación de la estructura de ficheros

Se puede emplear como punto de partida el ejemplo *hello_world* que aparece en el directorio *nds* del directorio *examples* de *DevkitPro*. En el laboratorio de prácticas, *DevkitPro* se encuentra en el directorio */opt/devkitpro*. En el directorio donde se vayan a almacenar los programas a desarrollar se crea un nuevo directorio que identifique el programa a desarrollar (p.ej. *ejemplo*). Dentro de ese directorio se crea el directorio *source*, que contendrá los ficheros necesarios para el código a desarrollar (p.ej. *main.c*). En el directorio *ejemplo* se copia el fichero *Makefile* del ejemplo *hello_world* de *DevkitPro*. De esta forma la estructura de ficheros que se tiene es la siguiente:

```
/home/usuario/mis_ejemplos
- directorio ejemplo
```



Figura 2.11: Creación de un proyecto para NDS usando Eclipse (parte 4).

- fichero Makefile
- directorio source
- fichero main.c

Edición del fichero ejemplo

Para familiarizarse con el entorno de desarrollo de aplicaciones para Nintendo DS, se va a utilizar como ejemplo una aplicación en la que aparezca un saludo con el nombre del desarrollador del programa. Para escribir este código se puede emplear cualquier editor de texto.

Según esto, el código del programa a desarrollar (*main.c*) es el siguiente:

```

1  #include <nds.h>
2  #include <stdio.h>
3  int main(void)
4  {
5      consoleDemoInit();
6      iprintf("Hola Juan"); // Imprimir el mensaje
7      while(1){} // Bucle que no hace nada.
8      return 0; // Finalizar el programa
9  }

```

Compilación del fichero ejemplo

El siguiente paso es compilar el programa, para ello se abre el *Terminal* (*Sistema->Terminal*). Una vez se está en el directorio *ejemplo* creado, se ejecuta el comando *make*:

```

[usuario@labsop02 ejemplo]# make
main.c
arm-none-eabi-gcc -MMD -MP -MF /root/mis_ejemplos/ejemplo/build/main.d -g -Wall
-O2 -march=armv5te -mtune=arm946e-s -fomit-frame-pointer -ffast-math
-mthumb -mthumb-interwork -I/root/mis_ejemplos/ejemplo/include
-I/root/mis_ejemplos/ejemplo/build -I/opt/devkitpro/libnds/include
-I/root/mis_ejemplos/ejemplo/build -DARM9 -c

```

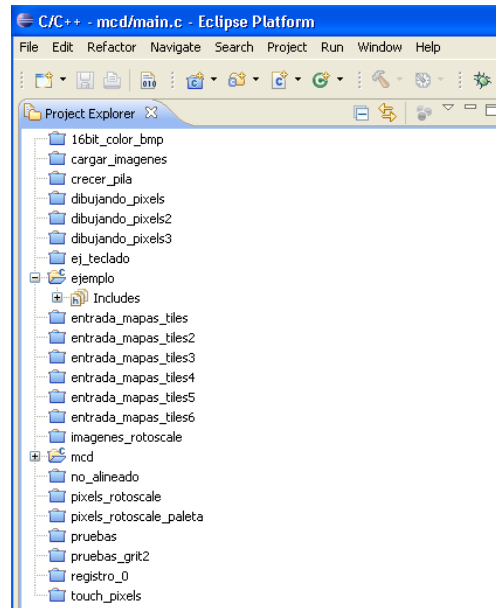


Figura 2.12: Creación de un proyecto para NDS usando Eclipse (parte 5).

```
/root/mis_ejemplos/ejemplo/source/main.c -o main.o
linking ejemplo.elf
Nintendo DS rom tool 1.50.1 - Jun 19 2012
by Rafael Vuijk, Dave Murphy, Alexei Karpenko
built ... ejemplo.nds
```

Si no se han producido errores de compilación aparecerán los ficheros *ejemplo.elf* y *ejemplo.nds*.

```
[usuario@labsop02 ejemplo]# ls -l
total 332
drwxr-xr-x 2 root root 4096 sep 5 18:01 build
-rwxr-xr-x 1 root root 234909 sep 5 18:01 ejemplo.elf
-rw-r--r-- 1 root root 134208 sep 5 18:01 ejemplo.nds
-rwxr-xr-x 1 root root 4903 abr 2 2012 Makefile
drwxr-xr-x 2 root root 4096 sep 5 18:00 source
```

Para borrar todos los ficheros y directorios creados durante la compilación se puede ejecutar *make clean*.

Si se produjese un error relacionado con que no encuentra el compilador se puede realizar lo siguiente:

- Hacer una copia de los siguientes ficheros que se encuentran en */opt/devkitpro/devkitARM/bin*:
 - arm-none-eabi-as
 - arm-none-eabi-g++
 - arm-none-eabi-gcc
 - arm-none-eabi-gdb
 - arm-none-eabi-objcopy
- Renombrar las copias con los siguientes nombres:
 - arm-eabi-as
 - arm-eabi-g++
 - arm-eabi-gcc

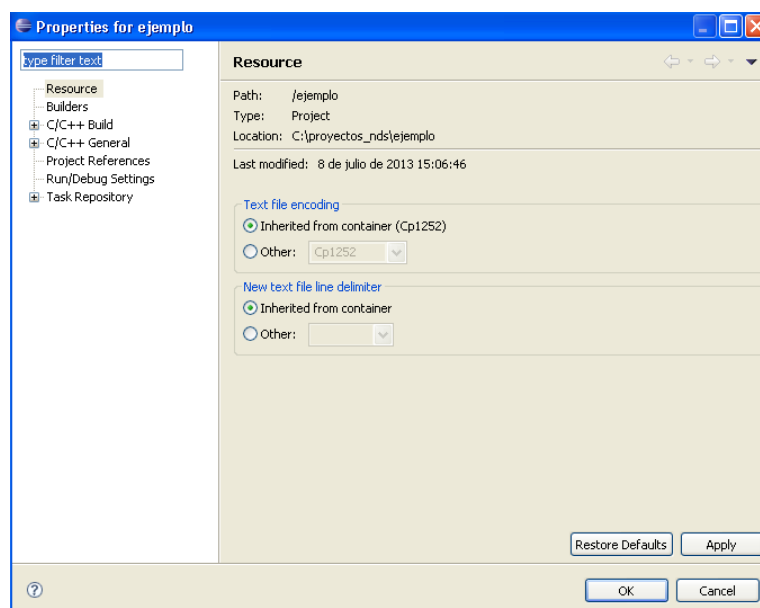


Figura 2.13: Creación de un proyecto para NDS usando Eclipse (parte 6).

- arm-eabi-gdb
- arm-eabi-objcopy

Ejecución del fichero ejemplo en el emulador

Si no se ha producido ningún problema en la compilación, la salida del programa se puede ver en el emulador. En el laboratorio de prácticas, *DeSmuME* se encuentra en el directorio *C:*. Una vez abierto *WinDS Pro*, si se escoge el emulador *No\$gba*, se pulsa en *File->Cartridge Menu (File Name)* y se busca el fichero *.nds* que nos interesa. Si se escoge el emulador *DeSmuME*, se pulsa en *File->Open ROM* y se busca el fichero *.nds* que nos interesa.

2.5.2 Desarrollar código para la NDS empleando Eclipse

En este apartado se va a crear el primer programa en la NDS empleando *Eclipse* como herramienta de desarrollo.

Creación de un proyecto para NDS

En el laboratorio de prácticas, *Eclipse* se encuentra en el directorio */opt/eclipse-helios*. Al iniciar *Eclipse* se pide el directorio donde se almacenará el proyecto a crear (ver Figura 2.21).

Para crear un nuevo proyecto se debe pulsar en la ventana principal de *Eclipse* en *Archivo->Nuevo->Proyecto*. Aparece la ventana (*Proyecto nuevo*) mostrada en la Figura 2.22.

Se escoge *C/C++->Proyecto en C* y se pulsa en *Siguiente*, apareciendo la ventana *Proyecto C* mostrada en la Figura 2.23.

En dicha ventana se debe configurar lo siguiente:

- El nombre de proyecto (p.ej. *ejemplo*).
- En *Project type* se selecciona *NintendoDS Rom->Empty Project (libnds)*.
- Se pulsa en *Siguiente*.

Aparece una ventana de *Select Configurations* en la que se pulsa en *Finalizar*.

De esta forma, el proyecto creado aparecerá en la ventana de *workbench* de *Eclipse* lo que se muestra en la Figura 2.24.

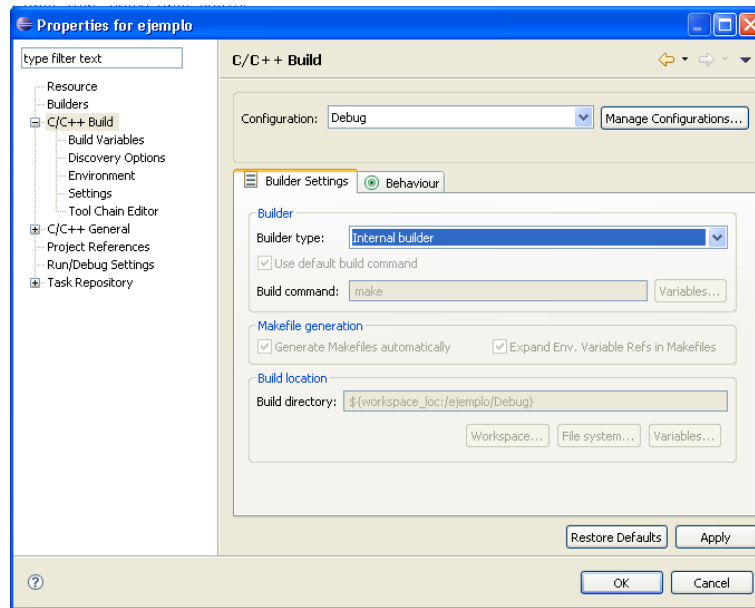


Figura 2.14: Creación de un proyecto para NDS usando Eclipse (parte 7).

Configuración del proyecto

Para iniciar la configuración del proyecto, se debe pulsar en la ventana principal de *Eclipse* en *Proyecto->Propiedades*, teniendo la precaución de tener activado el proyecto que se desea. Una vez realizada esta operación aparece la ventana (*Propiedades de ejemplo*) (ver Figura 2.25).

En dicha ventana se debe desplegar la pestaña *C/C++ Build*. Se resalta dicha opción. En el cuadro *Constructor* de la pestaña *Builder Settings* se cambia la opción *Builder Type* a *Internal Builder* (ver Figura 2.26).

En la pestaña *C/C++ Build* se resalta *Valores*, apareciendo la ventana mostrada en la Figura 2.27.

Se escoge *devkitARM C Linker->ARM* y se desactiva la opción *No FPU*. Finalmente se pulsa en *Aceptar* (ver Figura 2.28).

Edición del fichero ejemplo

Para familiarizarse con el entorno de desarrollo de *Eclipse* para NDS, se utiliza el mismo ejemplo que el del apartado anterior. En primer lugar, se debe crear un fichero fuente en *lenguaje C* dentro del proyecto actual. Para ello se pulsa en *Archivo->Nuevo->Source File*. Aparece una nueva ventana (*New Source File*) (ver Figura 2.29).

en la que se debe realizar lo siguiente:

- En *Source File* se introduce *main.c*.
- En *Template* se selecciona *Ninguno*.
- Se pulsa en *Finalizar*.

A continuación en la ventana que hace referencia a *main.c* se introduce el mismo código que el del apartado 7.1.2.

Compilación del fichero ejemplo

El siguiente paso es compilar el programa, para ello se elige *Proyecto->Construir proyecto*. Si no se han producido errores de compilación aparecerá el fichero *ejemplo.nds* en el directorio *Debug*. Si se produjese un error relacionado con que no encuentra el compilador se puede realizar lo siguiente:

- Hacer una copia de los siguientes ficheros que se encuentran en */opt/devkitpro/devkitARM/bin*:

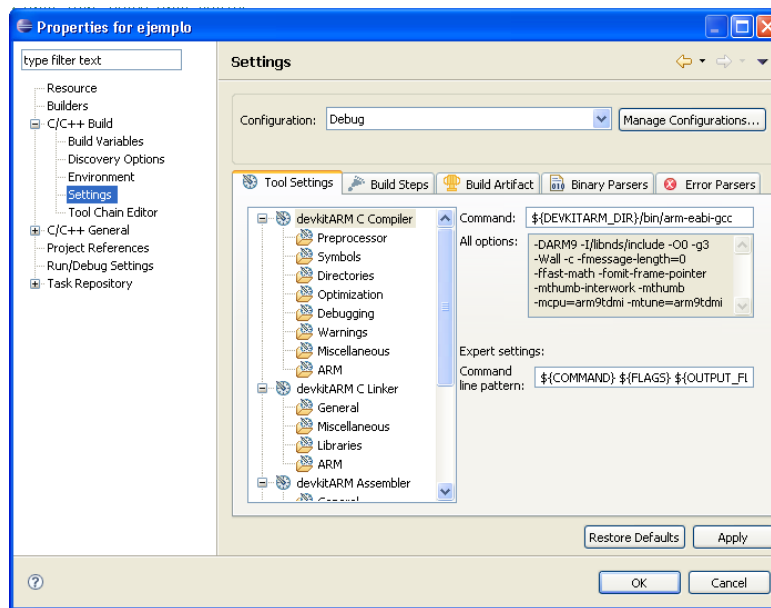


Figura 2.15: Creación de un proyecto para NDS usando Eclipse (parte 8).

- arm-none-eabi-as
- arm-none-eabi-g++
- arm-none-eabi-gcc
- arm-none-eabi-gdb
- arm-none-eabi-objcopy
- Renombrar las copias con los siguientes nombres:
 - arm-eabi-as
 - arm-eabi-g++
 - arm-eabi-gcc
 - arm-eabi-gdb
 - arm-eabi-objcopy

Ejecución del fichero ejemplo en el emulador

Una vez abierto *WinDS Pro*, si se escoge el emulador *No\$gba*, se pulsa en *File->Cartridge Menu (File Name)* y se busca el fichero *.nds* que nos interesa. Si se escoge el emulador *DeSmuME*, se pulsa en *File->Open ROM* y se busca el fichero *.nds* que nos interesa.

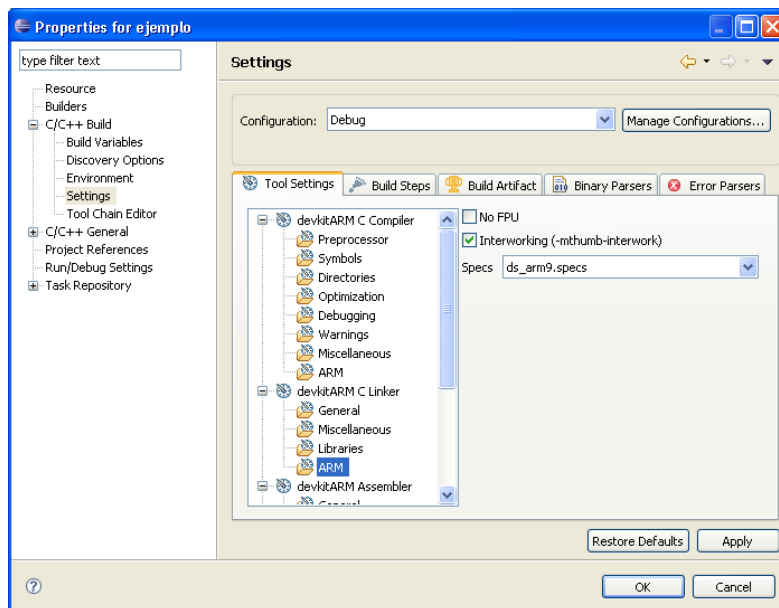


Figura 2.16: Creación de un proyecto para NDS usando Eclipse (parte 9).

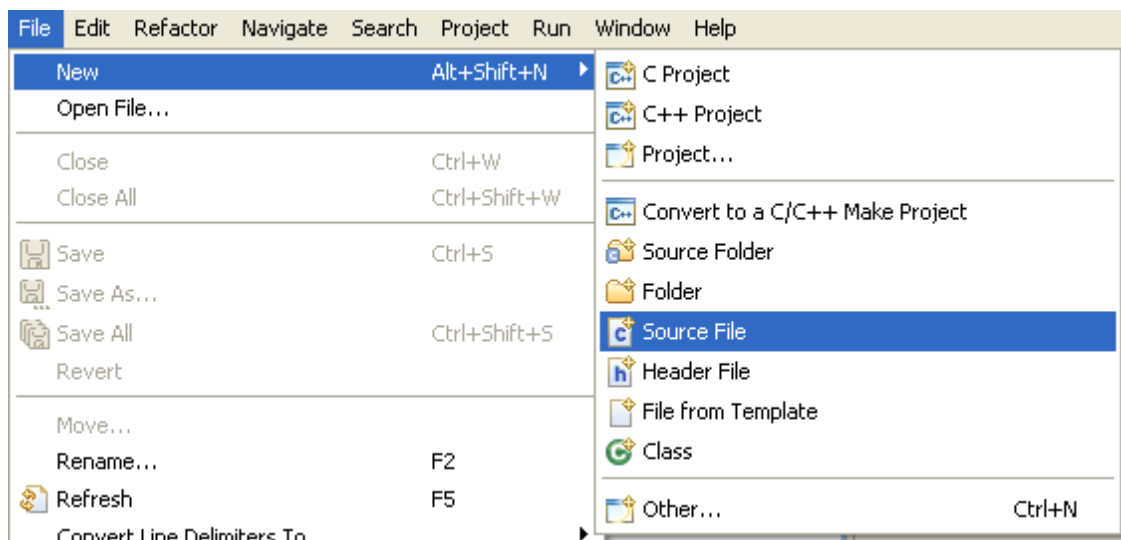


Figura 2.17: Creación de un proyecto para NDS usando Eclipse (parte 10).

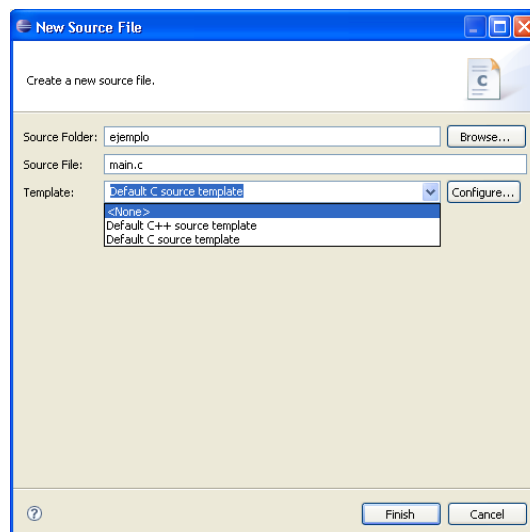


Figura 2.18: Creación de un proyecto para NDS usando Eclipse (parte 11).

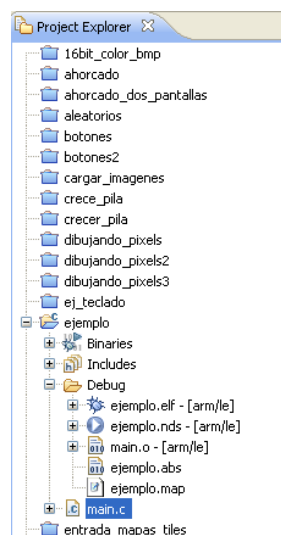


Figura 2.19: Creación de un proyecto para NDS usando Eclipse (parte 12).

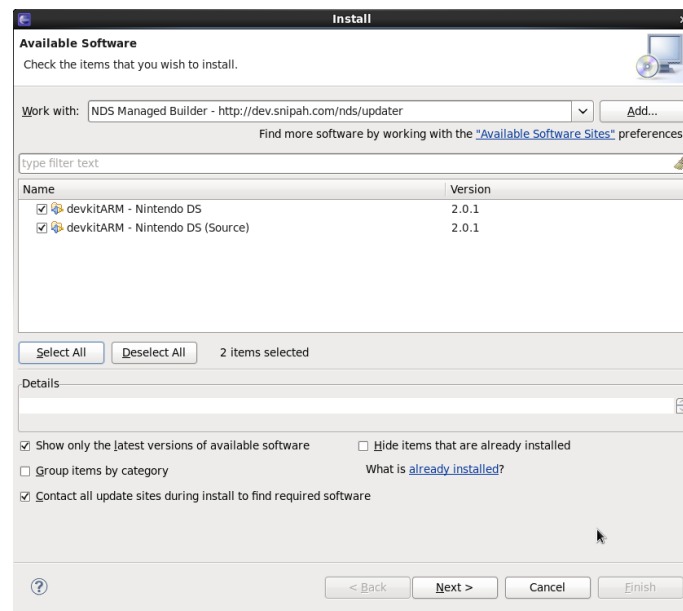
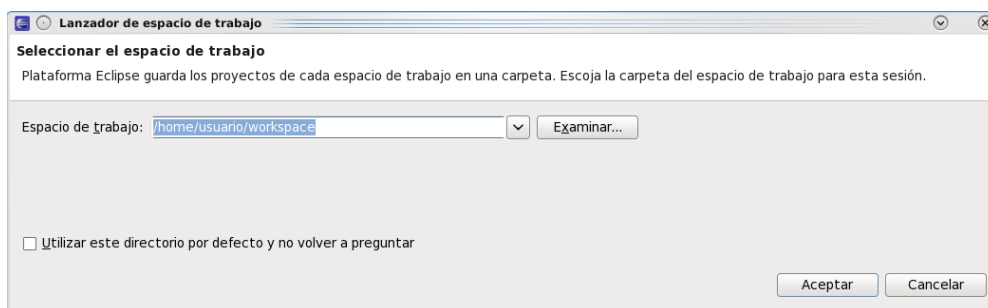
Figura 2.20: Instalación del *plugin NDS ManagedBuilder*

Figura 2.21: Creación de un proyecto para NDS usando Eclipse en linux (parte 1).

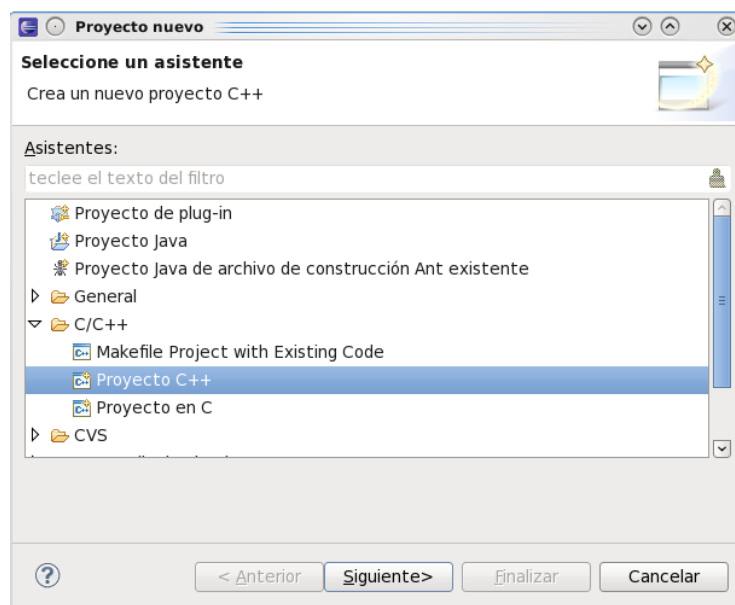


Figura 2.22: Creación de un proyecto para NDS usando Eclipse en linux (parte 2).

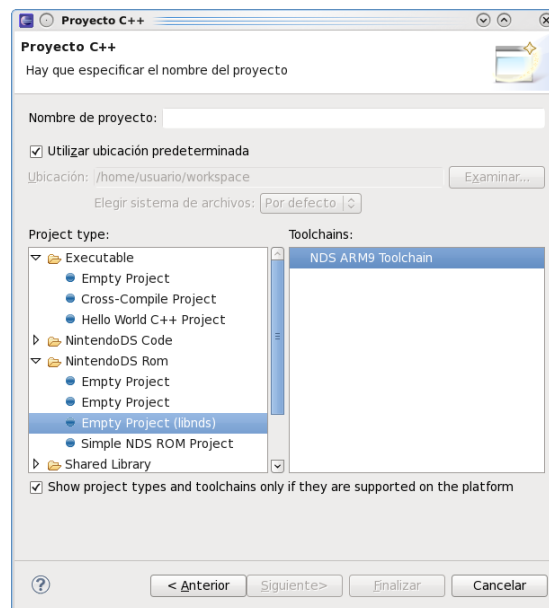


Figura 2.23: Creación de un proyecto para NDS usando Eclipse en linux (parte 3).

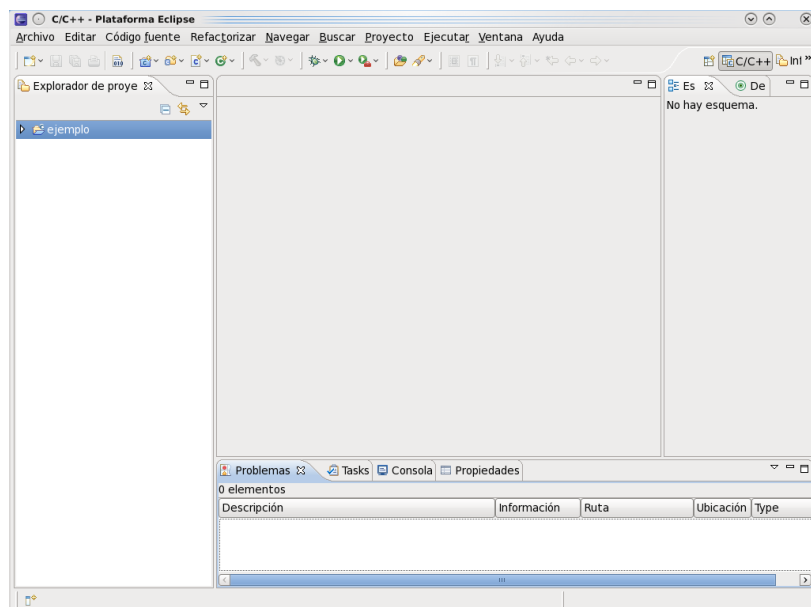


Figura 2.24: Creación de un proyecto para NDS usando Eclipse en linux (parte 4).

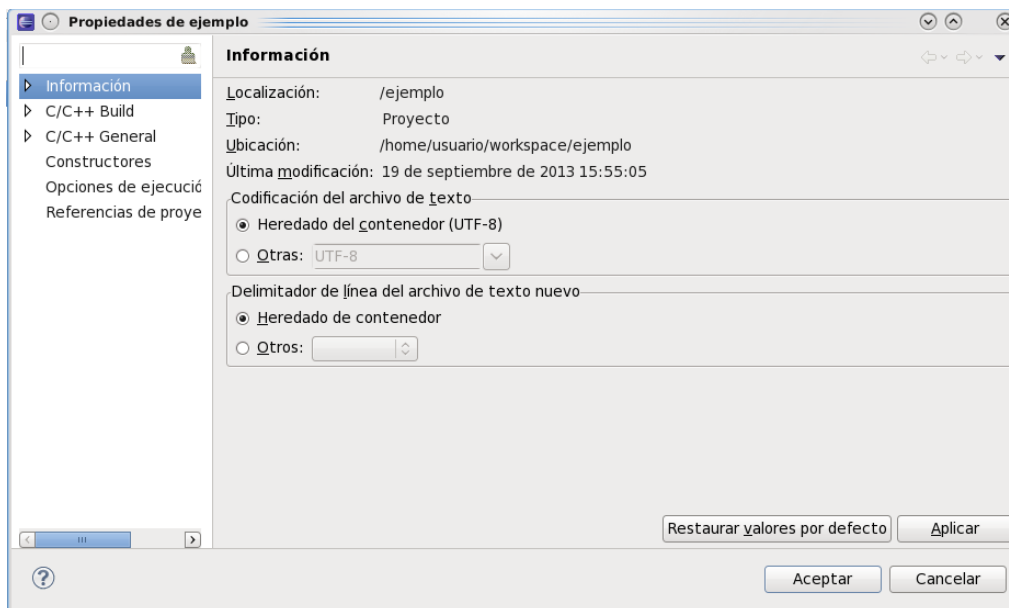


Figura 2.25: Creación de un proyecto para NDS usando Eclipse en linux (parte 5).

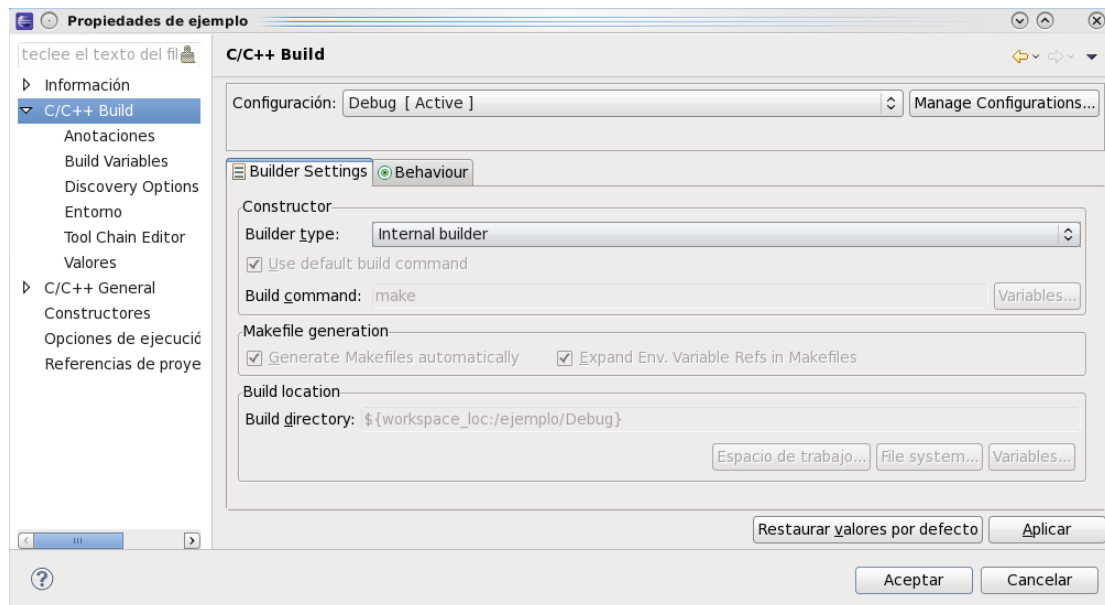


Figura 2.26: Creación de un proyecto para NDS usando Eclipse en linux (parte 6).

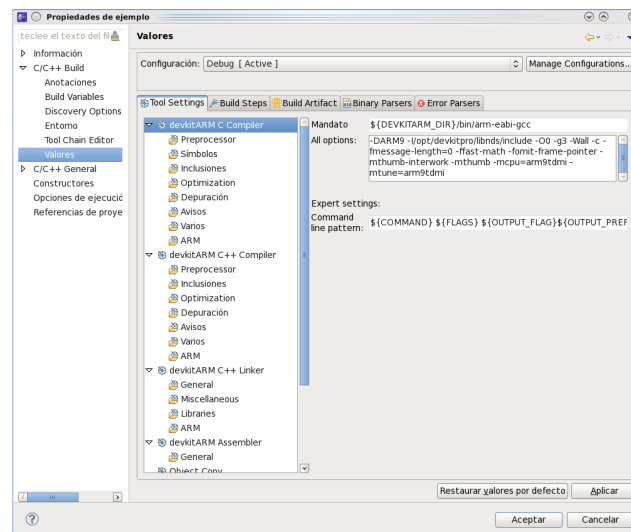


Figura 2.27: Creación de un proyecto para NDS usando Eclipse en linux (parte 7).

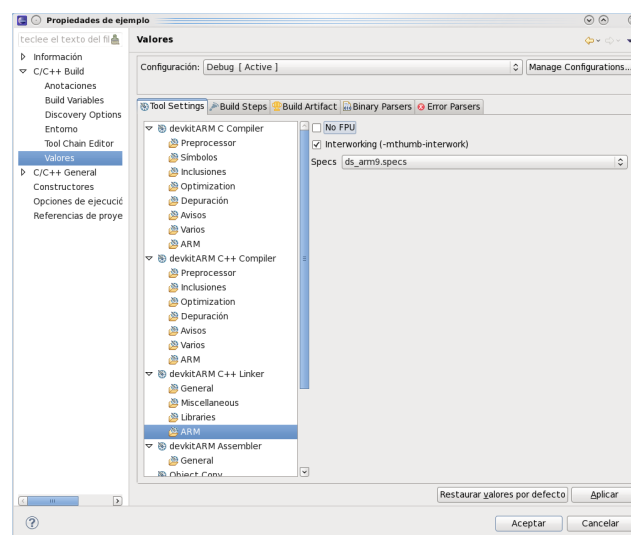


Figura 2.28: Creación de un proyecto para NDS usando Eclipse en linux (parte 8).

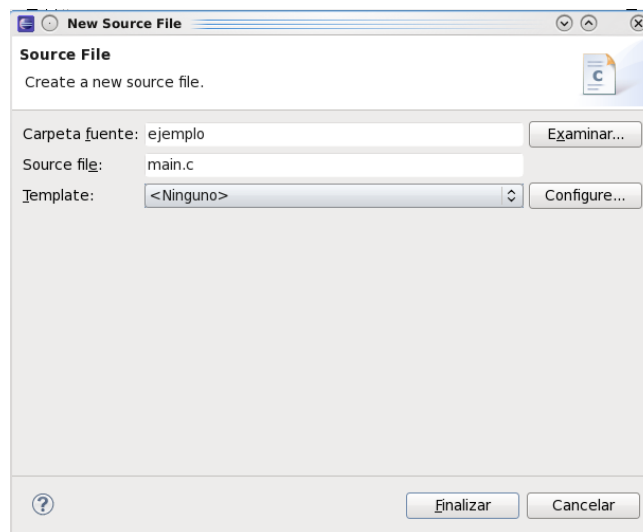


Figura 2.29: Creación de un proyecto para NDS usando Eclipse en linux (parte 9).

Bibliography

Books

Articles