

# Guía de desarrollo para los Laboratorios del curso

## 1 Introducción

Para la realización de los laboratorios y los proyectos se requiere:

- Descarga del repositorio Contiki-NG.
- Instalación de las herramientas de desarrollo.

La instalación de las herramientas de desarrollo implica utilizar un entorno Linux o MacOS. Por lo tanto se proponen las siguientes opciones dependiendo de cada situación particular:

1. **Docker:**<sup>1</sup> es la opción recomendada por los docentes del curso y oficial del proyecto Contiki-NG. Implica:

1. **Linux y macOS:**

- **Instalación de Docker,**
- **Descarga de la imagen contiker/contiki-ng y**
- **Clonación del repositorio contiki-ng.**

2. **Windows:**

- **Instalación de VMWare,**
- **Descarga de máquina virtual Linux** proporcionada por los docentes del curso.

Nota: La instalación de docker directamente sobre Windows es posible pero no permite la interacción con hardware, por lo que esta opción no se considera en este curso.

2. Instalación nativa de las herramientas de desarrollo (**sin soporte por parte de los docentes del curso**):

1. Instalación nativa en Linux: [Toolchain installation on Linux](#).
2. Instalación nativa en MacOS: [Toolchain installation on macOS](#).
3. Windows: Máquina virtual de linux con instalación nativa de herramientas, análogo al [Toolchain installation on Linux](#).

**Nota:** El entorno de desarrollo se encuentra instalado en las computadoras del Laboratorio de Software del IIE (entorno Linux, usuario `rsi`, clave `rsi` seguido del número dos mil dieciséis), por lo que su uso está disponible para los estudiantes que así lo deseen. El único inconveniente es que aún no se encuentra funcionando la aplicación UniFlash (la cual no debería ser necesaria para utilizar los Launchpad).

Esta guía se divide en tres partes:

- **Guía de instalación:** que apunta a contener todo lo necesario para realizar la instalación inicial del entorno de desarrollo. Pretende ser una *guía de un sólo uso* en el sentido de que una vez realizada no debería ser necesario volver a consultarla.

---

<sup>1</sup>Docker es un software libre y de código abierto que permite el manejo de contenedores; se trata de un nivel intermedio entre la instalación *nativa* de programas, y la instalación de una máquina virtual que se abstraiga completamente del sistema operativo anfitrión. En la sección **Guía de uso** se detallan aspectos relativos al uso de este software para los laboratorios del curso.

- **Guía de uso:** que contiene los comandos más bien *cotidianos* para el desarrollo sobre Contiki-NG y puede ser necesario consultar en diversas ocasiones.
- **Anexo y solución de problemas:** es la sección final orientada a contener aspectos que no deberían ser necesarios para el desarrollo del curso pero que apuntan a tener en cuenta posibles imprevistos.

**Nota:** Tanto la presente guía como las letras de los laboratorios contienen enlaces (a distintos sitios web o secciones del propio documento) pensados para utilizarse en la versión digital; muchas veces no se explicitan por lo que no serán visibles en la versión impresa.

---

### Notas sobre actualización del curso (desde 2023):

- Se actualiza la plataforma de software (OS y pila de comunicación) utilizando Contiki-NG en lugar de Contiki-OS.
- Se actualizan las plataformas hardware utilizando kits basados en los SoC CC2650 y CC1350 (lauchpad y sensortag).
- Se continúa utilizando el simulador Cooja.
- En los laboratorios se hará referencia a lauchpad (LP-CC1350 y LP-CC2650) y sensortag (ST-CC1350 y ST-CC2650) para ejecución en hardware y *cooja* para simular en Cooja.
- El sitio de referencia es <https://www.contiki-ng.org/> en particular para acceder al [código](#) y a la [documentación](#). Este último es el material de preferencia.
- El texto de referencia pasa a ser “Practical Contiki-NG” (disponible en <https://foco.timbo.org.uy>). En segundo lugar se recomienda también el libro “IoT in five Days” aunque es para la Contki-OS.

## 2 Guía de instalación

### 2.1 Linux y macOS

En caso de contar con una computadora Linux o macOS, lo más práctico es instalar Docker. A continuación se desarrolla una guía simplificada, por más información se puede consultar la guía en [Contiki-NG->Getting-Started->Docker](#).

#### 2.1.1 Instalación de Docker

Docker viene en dos versiones:

- *Docker Engine*: sería la versión tradicional de docker, se utiliza mediante línea de comandos.
- *Docker Desktop*: es una versión más amigable, con entorno gráfico y que incluye *Docker Engine*.

Esta guía está basada en *Docker Engine*, y dado que puede haber algunas diferencias o incompatibilidades con *Docker Desktop*, se recomienda usar la primera opción, dejando la segunda para quienes quieran aventurarse bajo su propio riesgo.

La instalación de *Docker Engine* se realiza en general a través de los repositorios de software de cualquier distribución, aunque Ubuntu requiere algunos pasos extra. Por más información consulte la [Guía oficial](#).

#### 2.1.2 Descarga de la imagen contiker/contiki-ng

Descargar la imagen de Contiki-NG utilizando el siguiente comando (no importa en qué directorio se ejecute, descarga la imagen en forma centralizada en el sistema, requiere que el servicio `docker` esté activo):

```
docker pull contiker/contiki-ng
```

### 2.1.3 Clonación del repositorio contiki-ng

Descargar el código de Contiki-NG utilizando el siguiente comando (esto creará un directorio `contiki-ng` donde se instalará el software de Contiki-NG, y será el lugar donde se guardarán nuestros proyectos):

```
git clone https://github.com/contiki-ng/contiki-ng.git
cd contiki-ng
git submodule update --init --recursive
```

Además, para facilitar el manejo del contenedor de Contiki-NG se debe descartar el script `contiker` creado por los docentes (dentro del directorio `contiki-ng`):

```
# Utilizando wget:
wget https://rsi.pages.fing.edu.uy/laboratorios-rsi/Entorno-NG/contiker
# o curl:
curl -O https://rsi.pages.fing.edu.uy/laboratorios-rsi/Entorno-NG/contiker
```

y configurarlo como ejecutable:

```
chmod a+x contiker
```

## 2.2 Windows

### 2.2.1 Instalación de VMWare

Descargar e instalar VMWare Workstation Player del [sitio oficial](#).

### 2.2.2 Descarga de máquina virtual Linux

- Descargar la [imagen de Linux](#) creada por los docentes del curso y extraerla en una carpeta.
- Abrir VMWare Workstation Player y hacer click en “Open a Virtual Machine”.
- Seleccionar dentro de la carpeta en la que se descomprimió la VM.
- Hacer click en “Play Virtual Machine” y cuando pregunte si la VM fue copiada o movida, seleccionar “I Copied It”.
- El usuario es `user` y la clave es `rsi`.
- La aplicación de línea de comandos se llama *Sakura*.
- Contiki-NG se encuentra en la carpeta `contiki-ng`:

```
cd ~/contiki-ng/
```

### 2.2.3 Compartir archivos entre la máquina virtual y la anfitriona

Para configurar una carpeta compartida primero debe existir en el sistema operativo anfitrión (Windows). Luego, en la configuración de la máquina virtual (VMWare), en la pestaña *Options* hay una sección llamada *Shared Folders* que debe habilitarse (*Always enabled*) y agregar allí las carpetas que se quieran compartir.

Habiendo hecho esto, en la máquina virtual, deben aparecer estos directorios dentro de la carpeta

```
/home/user/contiki-ng/compartido
```

## 2.3 Instalación nativa

Para la instalación nativa de las herramientas se sugiere seguir las instrucciones de la documentación de `contiki-ng`:

- Instalación nativa en Linux: [Toolchain installation on Linux](#).

- Instalación nativa en MacOS: [Toolchain installation on macOS](#).

Adicionalmente se requerirá la instalación de [Eclipse Californium \(cf-browser\)](#) así como lo referido a la sección [Clonación del repositorio contiki-ng](#).

## 3 Guía de uso

Esta guía de uso asume que se completaron los pasos de la sección [Guía de Instalación](#) y que, ya sea dentro de una máquina virtual o en el sistema operativo anfitrión, se tiene un directorio `contiki-ng` con el software de Contiki-NG y el *script* `contiker`.

Se asume que los comandos aquí mencionados, a no ser que se explicita lo contrario, se ejecutan dentro de este directorio `contiki-ng`. En caso de ser necesario se puede utilizar el comando `cd` para cambiar de directorio, por ejemplo:

```
cd ~/contiki-ng
```

### 3.1 Inicio del contenedor `contiker/contiki-ng`

La mayor parte de la guía trabaja dentro del contenedor `contiker/contiki-ng`. Para introducirse en este entorno se debe ejecutar el comando

```
./contiker
```

Algunos comandos pueden ejecutarse desde *afuera* del contenedor, por ejemplo, para abrir el simulador Cooja se puede ejecutar

```
./contiker cooja
```

que es equivalente a ejecutar primero

```
./contiker
```

y luego

```
cooja
```

En forma análoga, para abrir el cliente gráfico CoAP (utilizado en el Laboratorio 3) se puede ejecutar

```
./contiker cf-browser
```

que es equivalente a ejecutar primero

```
./contiker
```

y luego

```
cf-browser
```

### 3.2 Compilación del firmware

**Dentro del contenedor** moverse a la carpeta del proyecto que se quiere programar, por ejemplo, para probar el ejemplo de los LEDs en el LP-CC2650:

```
cd examples/dev/leds
```

Luego compilar el programa con `make`, por ejemplo:

```
make TARGET=cc26x0-cc13x0 BOARD=launchpad/cc1350 leds-example
```

(donde los valores `TARGET` y `BOARD` deben estar acordes a la placa para la cual se está desarrollando).

El resumen de los comandos disponibles se puede consultar en la [hoja de referencia \(cheat-sheet\)](#). En particular se recomienda usar el comando `savetarget` para facilitar el desarrollo.

### 3.3 Programar las placas [LP|ST]-[CC1350|CC2650]

Nos referiremos con estos nombres a las placas:

- `LAUNCHXL-CC1350` (LP-CC1350),
- `LAUNCHXL-CC2650` (LP-CC2650),
- `CC1350STK` (ST-CC1350) y
- `CC2650STK` (ST-CC2650).

Hay esencialmente dos formas de programar la placa:

- Utilizando Contiki-NG mediante el bootloader serial en ROM.
- Utilizando UniFlash mediante la interfaz JTAG.

#### 3.3.1 Placas LP-[CC1350|CC2650]

Estas placas permiten ambas formas de programación, pero por practicidad ésta se realizará mediante el bootloader serial disponible en la ROM de las placas. Podría suceder (aunque no debería) que el bootloader quede bloqueado, para lo cual se necesitará borrar el contenido de la memoria flash de los dispositivos (para desbloquear bootloader) usando UniFlash (ver apartado correspondiente en la sección [Anexo y solución de problemas](#)).

Una vez conectado el dispositivo, este debe estar visible en Linux (o dentro del contenedor) bajo el nombre `/dev/ttyACM2`. Se puede verificar que el dispositivo se encuentra conectado correctamente ejecutando el siguiente comando:

```
ls /dev/ttyACM*
```

Además, estando **dentro del mismo directorio del ejemplo a compilar**, el comando

```
make motelist-all
```

debe listar las placas conectadas.

Muchas veces sucede que luego de conectar la placa a la computadora, ésta no es reconocida dentro del contenedor: se debe salir del contenedor (ejecutando simplemente el comando `exit`) y volver a iniciarlo.

Para programar la placa, primero se debe activar el modo programación, para lo cual se deben presionar los botones `Reset` y `BTN-1` simultáneamente, para luego soltar primero el botón `Reset` y un segundo después el `BTN-1`. Esto debería dejar la placa en modo programación, con sólo un LED verde encendido que indica que la placa está energizada.

Luego se debe utilizar el comando `make` especificando los correspondientes parámetros `TARGET` y `BOARD`, por ejemplo:

```
make TARGET=cc26x0-cc13x0 BOARD=launchpad/cc1350 leds-example.upload
```

En caso que el sistema no seleccione correctamente la placa, o que tengamos más de una placa conectadas al mismo tiempo, se puede utilizar el parámetro `PORT` para especificar el puerto en el comando `make`, por ejemplo:

```
make TARGET=cc26x0-cc13x0 BOARD=launchpad/cc1350 PORT=/dev/ttyACM0 leds-example.upload
```

<sup>2</sup>Cada Launchpad muestra dos puertos seriales, por ejemplo `/dev/ttyACM0` y `/dev/ttyACM1`.

### 3.3.2 Placas SP-[CC1350|CC2650]

Estas placas no permiten la programación utilizando bootloader serial (no cuentan con conexión serial via USB), por lo que la programación se realizará mediante la interfaz JTAG utilizando el programador J-Link EDU Mini y el software UniFlash.

Pasos a seguir:

1. Utilizando el cable adecuado, conectar la placa Sensortag con el programador J-Link EDU Mini. Verificar la correcta polaridad del cable: la línea roja debe ir del lado donde está el número 1 o el punto grande (los extremos del cable son indistinguibles entre sí).
2. La placa Sensortag debe tener la pila puesta y en contacto (quitar plástico aislante).
3. Conectar el J-Link EDU Mini al puerto USB de la computadora.
4. Abrir el software UniFlash:
  1. Seleccionar el dispositivo CC1350F128 o CC2650F128 según corresponda (en su versión On-Chip).
  2. Seleccionar el programador SEGGER J-Link Emulator.
  3. Click en *Start*.
  4. Seleccionar la imagen `.elf` a grabar. Esta se encuentra en una subruta dentro del directorio `build` donde se compiló el proyecto (se debe activar la opción de ver todos los archivos para que aparezca en el explorador).
5. Para que el programa se ejecute, se debe desconectar el programador, quitar la pila del Sensortag y volverla a colocar.

**Nota:** Se recomienda que el código a utilizar prevea el encendido de LEDs a modo de diagnóstico.

## 3.4 Simulación en Cooja

En la documentación de Contiki-NG hay un [tutorial sobre cooja](#), se recomienda su lectura.

### 3.4.1 Visualizar LEDs y pulsar botón en Cooja

En la simulación, se puede hacer click derecho sobre el nodo, opción *Mote tools for Contiki*, y hacer click en *Mote Interface Viewer...* Esto despliega una ventana que permite observar los LEDs y pulsar el botón, entre otras funcionalidades.

Otra manera de visualizar los LEDs es ir a *View* en la ventana *Network* y seleccionar *LEDs*.

## 4 Anexo y solución de problemas

### 4.1 Problemas en la programación de las placas LP-[CC1350|CC2650]

Si al programar la placa, el comando devuelve

```
Connecting to target...
ERROR: Timeout waiting for ACK/NACK after 'Synch (0x55 0x55)'
```

es porque no pudo conectarse con la placa. Esto puede ser por múltiples causas:

- el sistema operativo (o la máquina virtual) no reconoció el dispositivo: revisar que se encuentre el dispositivo en `/dev/ttyACMx` y que sea reconocido por contiki mediante el comando `make motelist-all` (puede ser necesario salir de todos los contenedores activos y volver a iniciarlos).
- el proceso de dejar la placa en modo programación no resultó satisfactorio: simplemente volver a intentarlo.
- el bootloader quedó bloqueado: ir al apartado [Desbloquear bootloader: borrar la memoria flash utilizando UniFlash](#).

## 4.2 Desbloquear bootloader: borrar la memoria flash utilizando UniFlash

En casos en que el bootloader quedó bloqueado, ya sea porque no quedaron bien configuradas las banderas de desbloqueo en la compilación de Contiki (deberían quedar configuradas por defecto si se utilizan correctamente los parámetros `TARGET` y `BOARD`), o porque es la primera vez que utilizamos la placa, se debe utilizar la aplicación UniFlash de Texas Instruments para borrar el contenido de la memoria flash.

Para ello descargar la aplicación de [UniFlash](#).

En el caso de los LP-[1350|2650] al abrir UniFlash los va a autodetectar, y al presionar Start aparecen varias opciones. Para borrar la memoria flash ir a “Settings & Utilities”, luego ir al apartado “Manual erase” y borrar la flash completa. Esto automáticamente dejará el dispositivo en modo programación hasta que se le grabe un nuevo firmware.

## 4.3 Programar la placa Re-Mote

Para esta plataforma no hay que relizar previamente el desbloqueo del bootloader ni activar el modo programación con la secuencia botón-reset.

Una vez conectado, este debe estar visible en Linux bajo el nombre `/dev/ttyUSBx`. Se puede verificar que el dispositivo se encuentra conectado correctamente ejecutando el siguiente comando:

```
ls /dev/ttyUSB*
```

Dentro del contenedor, moverse a la carpeta del proyecto que se quiere programar, por ejemplo, para probar el ejemplo de los LEDs:

```
cd examples/dev/leds
```

Luego se debe utilizar el comando `make` especificando el correspondiente parámetro `TARGET` en este caso:

```
make TARGET=zoul leds-example
```

Para cargarlo en la plataforma se ejecuta `make` más el `TARGET` y el nombre el archivo más la extensión `upload`, por ejemplo:

```
make TARGET=zoul leds-example.upload
```

## 4.4 Problemas al iniciar Cooja

Si la inicialización de Cooja presenta errores por no poder conectarse al `DISPLAY`, puede que en linux sea necesario habilitar esto ejecutando previamente el comando (desde fuera del contenedor):

```
xhost +local:
```

Si se está trabajando detrás de un proxy, por ejemplo desde la red interna de FIng, se debe crear el archivo `contiki-ng/tools/cooja/gradle.properties` con el siguiente contenido:

```
systemProp.http.proxyHost=proxy.fing.edu.uy
systemProp.http.proxyPort=3128
systemProp.https.proxyHost=proxy.fing.edu.uy
systemProp.https.proxyPort=3128
```

## 4.5 Problemas al compartir directorios entre el sistema operativo *anfitrión* y el *invitado*

El compartir directorios es posible gracias a que en el arranque de la máquina virtual se ejecuta el comando

```
/usr/bin/vmhgfs-fuse .host:/ /home/user/contiki-ng/compartido -o subtype=vmhgfs-fuse,allow_other
```

Esto fue configurado en el menú principal del sistema linux, sección *Menú de control->Ajustes->Aplicaciones predeterminadas para LXSession*, pestaña *Inicio automático*.

Este comando bien puede ejecutarse *a mano* y los parámetros pueden ser modificados a gusto. Más información en la [documentación oficial de VMware](#).

Los materiales de este curso fueron parcialmente financiados por:



Co-funded by the  
Erasmus+ Programme  
of the European Union