

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-27

# **Sistema basado en FPGA para la adquisición de datos de un sensor IP**

Yuri Solandy Hoyos Marín

Ingeniería Electrónica

Luis Fernando Castaño Londoño

**INSTITUTO TECNOLÓGICO METROPOLITANO**

**Julio de 2016**

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## RESUMEN

---

Las diferentes formas de integrar y controlar procesos en el contexto de la domótica e inmótica, requieren la comunicación a través de una red entre los dispositivos que intervienen. En este trabajo se presenta la implementación de un sistema orientado al manejo de sensores sobre IP, conformado por un módulo de adquisición de datos y un sistema de desarrollo basado en FPGA (Zedboard) conectados a través de una red de área local LAN. La adquisición de datos se realiza empleando un sensor de temperatura y un dispositivo Arduino. Este último se encarga de la conversión análogo digital (A/D) de la señal entregada por el sensor y facilita la transmisión de los datos sobre IP. Un sistema operativo embebido instalado sobre la Zedboard permite la configuración de la dirección IP y la ejecución de las aplicaciones necesarias para transmitir, visualizar y procesar la información. Finalmente empleando un *script* ejecutado en el sistema operativo embebido realiza la secuencia de instrucciones para la lectura y visualización de la información adquirida del sensor.

*Palabras clave:* Dispositivo Arduino, FPGA, Sistema operativo embebido.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## RECONOCIMIENTOS

---

Agradezco a Dios la posibilidad de alcanzar un logro más en mi desarrollo profesional, en compañía de mis padres, hermanas y mi esposo, ellos han sido parte fundamental en el transcurrir de mi vida. Doy gracias por encontrarme en el camino buenos compañeros y buenas experiencias que acompañaron todo mi proceso formación.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## ACRÓNIMOS

---

FPGA Field Programmable Gate Array.

IP Internet Protocol.

SD Secure Digital.

PC personal computer.

A/D. Análogo Digital.

LAN local Area Network.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## TABLA DE CONTENIDO

1. INTRODUCCIÓN.....	6
2. MARCO TEÓRICO.....	7
3. METODOLOGÍA .....	8
3.1. Instalación y configuración del Sistema Operativo embebido.....	8
3.2. Instrucciones de conexión y programación del Arduino para la Adquisición de datos del sensor. ....	26
3.3. Rutina de procesos .....	30
4. RESULTADOS Y DISCUSIÓN .....	34
5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO .....	37
REFERENCIAS .....	38
APÉNDICE.....	39

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# 1. INTRODUCCIÓN

---

La domótica e inmótica toman lugar en la automatización de funciones en el hogar y en edificios. Se caracterizan por emplear un medio de comunicación entre los equipos electrónicos que intervienen para proporcionar funciones de control. La tecnología Ethernet aporta los parámetros de estandarización bajo el protocolo TCP/IP en el proceso de recepción y transmisión de datos. No reemplaza protocolos como MODBUS o serial, pero posee ventajas de transferencias a mayor velocidad, acceso al medio con diferentes datos en la misma conexión e integración de redes existentes. Se aborda el tema de adquisición de datos de un sensor que corresponde a una etapa de las etapas del proceso de automatización como requisito previo antes de implementar una acción de control.

Este proyecto de grado tiene como objetivo implementar un sistema para la adquisición de datos de un sensor con la tecnología Ethernet y los sistemas embebidos basados en FPGA.

En el capítulo 2 se realiza la descripción de los sistemas basados en FPGA y de adquisición de datos con el fin de contextualizar las definiciones en lo referente al desarrollo del proyecto. La metodología empleada se especifica en el capítulo 3 donde se definen las etapas que intervienen en la implementación de la red de comunicación entre el sistema basado en FPGA y el sensor IP. Se establecen las pautas e instrucciones para la instalación de un sistema operativo embebido Linux, conexiones y configuración del sensor en el Arduino y los parámetros de configuración de la red LAN.

Los resultados obtenidos se encuentran en el capítulo 4, donde se definen las ventajas que ofrece el sistema operativo embebido Linux a través del uso de los controladores de los puertos y las aplicaciones requeridas para la asignación de dirección IP y la lectura de la información entregada por el sensor empleando una consola de terminal.

Finalmente el capítulo 5 reúne las conclusiones y mejoras encontradas al ejecutar el proyecto de investigación.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 2. MARCO TEÓRICO

---

La implementación del control del sensor IP empleado en el desarrollo de este trabajo hace uso de un sistema basado en SoC FPGA y un sistema para la adquisición de datos con la tecnología Ethernet. Los SoC FPGA son dispositivos lógicos programables que cuentan con un gran potencial de procesamiento. Se destacan de otros tipos de FPGA porque incluyen un procesador embebido. Son empleados en aplicaciones de comunicación implementadas sobre sistemas y entornos de desarrollo que poseen controladores Ethernet y librerías para realizar intercambio de información (Franco, O., 2012). Se pueden adquirir a menor costo que un equipo de cómputo y físicamente son más pequeños, logrando optimizar espacio al momento de diseñar un sistema de control.

Por otra parte el sistema de adquisición de datos es el encargado de leer la información de la señal analógica que entrega un sensor y convertirla en un dato digital para ser procesado a través de un sistema computacional. En la actualidad es común el uso de sistemas Arduino que se utilizan para la adquisición de datos. El diseño de esta tarjeta se basa en microcontroladores que incluyen un conversor análogo digital. El entorno de desarrollo (Arduino IDE) permite a través de su interfaz gráfica configurar las herramientas asociadas a los periféricos físicos de la tarjeta, tales como pines de entrada y salida y asignación de una dirección IP. Esta última permite dar un nombre al Arduino para ser identificado en una red LAN.

El sistema desarrollado utiliza la conexión a través de una red alámbrica de área local (LAN) para la comunicación entre los dispositivos. El intercambio de información o recursos se basa en estándar de comunicaciones IEEE 802.3, comúnmente conocido como Ethernet (Tanenbaum, 1997).

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 3. METODOLOGÍA

---

El propósito de este trabajo es el desarrollo de un sistema embebido basado en FPGA para la adquisición de datos de un sensor IP. La transferencia del dato con el sensor utiliza las características de una red de área local LAN.

La implementación es realizada en una tarjeta Zedboard Zynq®-7000 fabricada por Xilinx. Físicamente es una tarjeta electrónica que incluye un dispositivo SoC FPGA con un procesador Dual ARM® Cortex™-A9 MPCore™, puertos de comunicación Ethernet y Serial, tarjeta SD con capacidad de memoria de 4GB, periféricos de entrada y salida de datos análogos y digitales, conectores de audio y video.

La metodología empleada se divide en tres etapas:

1. Instalación y configuración del Sistema Operativo embebido Linux.
2. Instrucciones de conexión y programación del Arduino para la adquisición de datos del sensor de temperatura.
3. Rutina de procesos de lectura y escritura.

### 3.1. Instalación y configuración del Sistema Operativo embebido

---

El sistema operativo embebido instalado es una distribución libre de Linux llamada Petalinux. Se crea un archivo ejecutable denominado imagen del sistema operativo, el cual es instalado en la Zedboard FPGA desde la tarjeta SD.

A continuación se describen los pasos a seguir en un equipo PC con distribución de Linux.

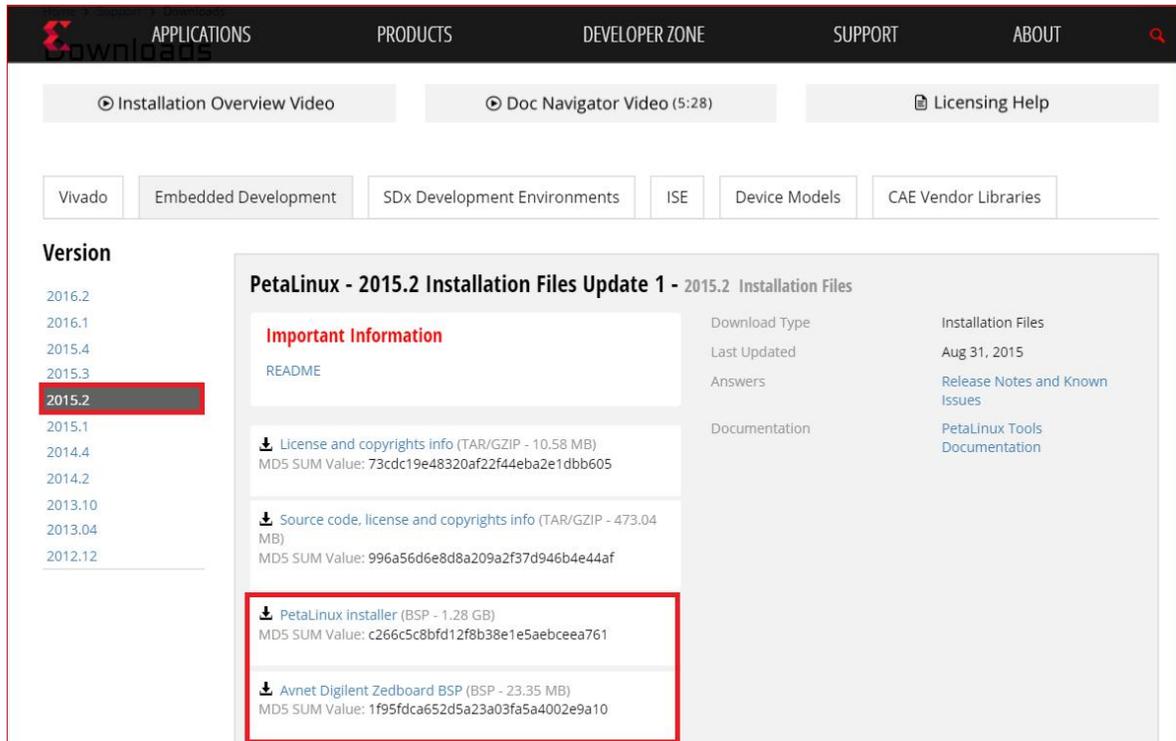
Descargar los archivos instaladores de la distribución de Linux compatible para tarjetas Zedboard FPGA desde el sitio web de Xilinx. Son archivos alojados en la

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

página web del fabricante.

<http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html>

## PetaLinux Zedboard BSP



The screenshot shows the Xilinx download page for PetaLinux. The navigation bar includes 'APPLICATIONS', 'PRODUCTS', 'DEVELOPER ZONE', 'SUPPORT', and 'ABOUT'. Below the navigation bar are video links for 'Installation Overview Video', 'Doc Navigator Video (5:28)', and 'Licensing Help'. A secondary navigation bar includes 'Vivado', 'Embedded Development', 'SDx Development Environments', 'ISE', 'Device Models', and 'CAE Vendor Libraries'. The main content area is titled 'PetaLinux - 2015.2 Installation Files Update 1 - 2015.2 Installation Files'. On the left, a 'Version' list shows versions from 2012.12 to 2016.2, with 2015.2 highlighted. The main content area contains an 'Important Information' section with a 'README' link, and a list of download links with their MD5 SUM values:

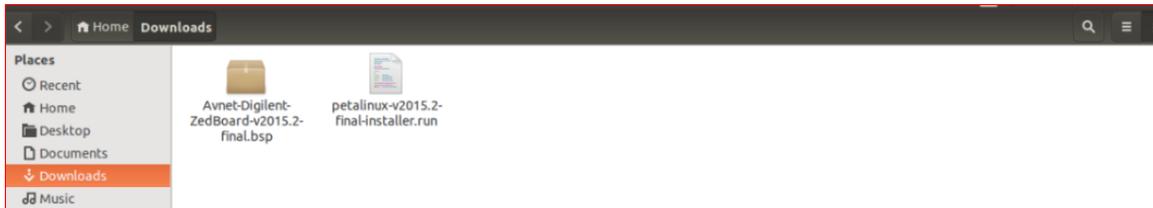
- License and copyrights info (TAR/GZIP - 10.58 MB)  
MD5 SUM Value: 73cdc19e48320af22f44eba2e1dbb605
- Source code, license and copyrights info (TAR/GZIP - 473.04 MB)  
MD5 SUM Value: 996a56d6e8d8a209a2f37d946b4e44af
- PetaLinux installer (BSP - 1.28 GB)**  
MD5 SUM Value: c266c5c8bfd12f8b38e1e5aebceea761
- Avnet Digilent Zedboard BSP (BSP - 23.35 MB)  
MD5 SUM Value: 1f95fdca652d5a23a03fa5a4002e9a10

On the right side of the page, there are links for 'Download Type', 'Installation Files', 'Last Updated' (Aug 31, 2015), 'Answers', 'Release Notes and Known Issues', 'Documentation', and 'PetaLinux Tools Documentation'.

*Imagen 1 Descarga Instaladores Petalinux y Zedboard BSP.*

⇒ Se verifica en la carpeta de descargas del equipo los dos archivos extraídos.

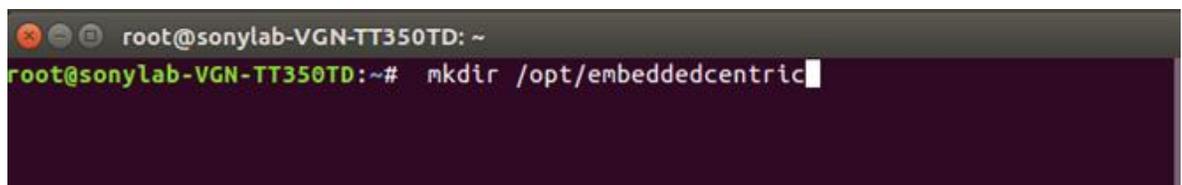
 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



*Imagen 2 Ubicación de archivos en la carpeta de descarga del PC.*

- ⇒ Abrir consola o terminal de Linux, ingresar la contraseña de administrador que tiene establecido el PC. Con el comando `sudo bash`  
Con este comando en la consola se trabaja como como *root* o administrador evitando ingresar credenciales cada vez que se realice algún proceso de configuración.
- ⇒ Se crea una carpeta en el directorio *opt*, en Linux este directorio se considera como ubicación convencional y se utiliza para almacenar aplicaciones y paquetes.  
Se utiliza en comando `mkdir`, se ubica en el directorio *opt* y la carpeta tendrá el nombre *embeddedcentric*:

```
mkdir /opt/embeddedcentric
```

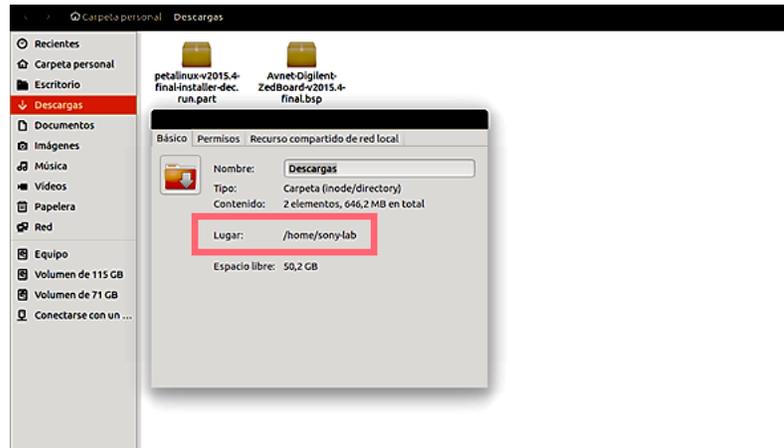


*Imagen 3 Comando mkdir para crear una carpeta en el directorio opt.*

- ⇒ Se copian los archivos *petalinux-v2015.2.1-final-installer.run* and *Avnet-Digilent-ZedBoard-v2015.2.1*, ubicados en la carpeta de descargas del equipo PC,

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

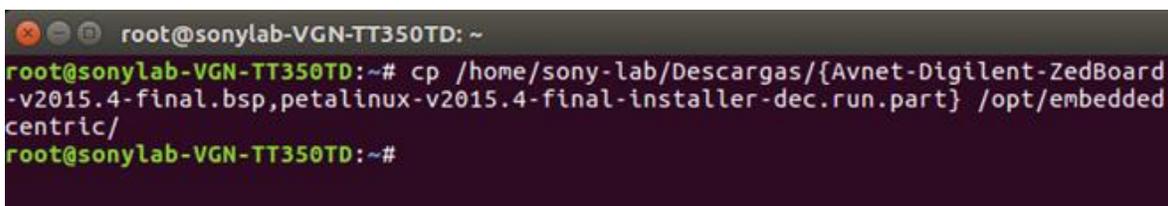
previamente extraídos de la página de Xilinx. Se debe tener en cuenta el lugar de ubicación origen (carpeta de descargas) y destino (carpeta creada en *opt*), para utilizar el comando *cp*, copiar archivos entre carpetas en Linux.



*Imagen 4 Ubicación archivos en carpeta descargas.*

Se utiliza el comando:

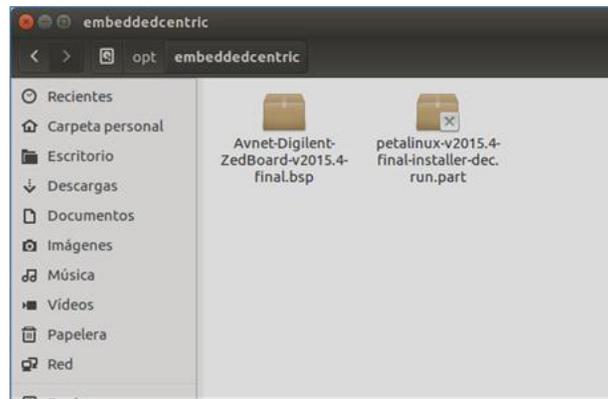
```
cp /home/sony-lab/Descargas/{Avnet-Digilent-ZedBoard-v2015.4-
final.bsp,petalinux-v2015.2.1-final-installer.run}
/opt/embeddedcentric/
```



*Imagen 5 Comando cp para copiar archivos de la carpeta de descargas a la carpeta creada en el directorio opt.*

Confirmar en la ubicación de la carpeta creada en el directorio *opt*, el contenido de los archivos seleccionados desde la consola.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



*Imagen 6 En la carpeta creada en el directorio opt se visualizan los archivos copiados por consola*

⇒ Descargar e instalar las últimas versiones de todos los paquetes que están actualmente en el PC, con el comando `apt-get update` se descargan por la consola desde la dirección `ubuntu.com`, este proceso tarda unos segundos, no interrumpir, al finalizar saldrá un mensaje indicando que se ha finalizado.

```

root@sonylab-VGN-TT350TD: ~
root@sonylab-VGN-TT350TD:~# apt-get update
Des:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [94,5 kB]
Obj:2 http://co.archive.ubuntu.com/ubuntu xenial InRelease
Obj:3 http://co.archive.ubuntu.com/ubuntu xenial-updates InRelease
Obj:4 http://co.archive.ubuntu.com/ubuntu xenial-backports InRelease
Descargados 94,5 kB en 1s (80,6 kB/s)
Leyendo lista de paquetes... Hecho
root@sonylab-VGN-TT350TD:~#

```

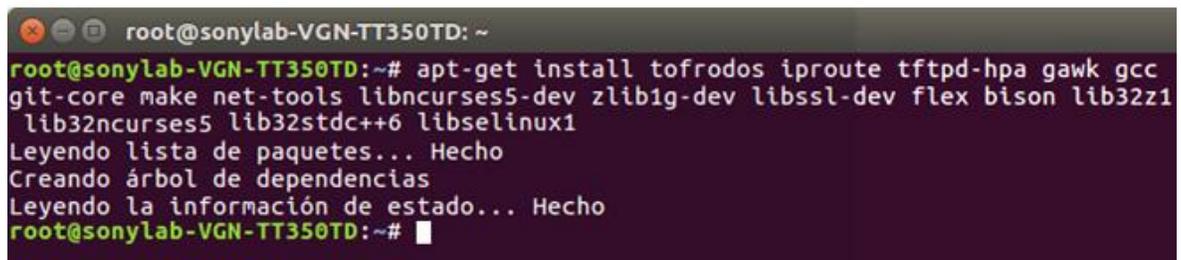
*Imagen 7 Descarga de versiones en el PC desde Ubuntu, con el comando apt-get update*

⇒ En las versiones de petalinux es necesario actualizar las herramientas de desarrollo y librerías. Para realizar este proceso se ejecuta el comando a continuación en la consola de Linux, se debe contar con acceso a internet desde el equipo PC, las librerías recomendadas son las siguientes: `-libncurses5 - dev - dev`

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

*zlib1g libssl - dev flexión bisontes lib32z1 lib32ncurses5 lib32bz2-1.0 lib32stdc ++ 6 libselinux.*

Se realiza la instalación de librerías recomendadas con el comando apt-get install tofrodos iproute tftpd-hpa gawk gcc git-core make net-tools libncurses5-dev zlib1g-dev libssl-dev flex bison lib32z1 lib32ncurses5 lib32stdc++6 libselinux1



```

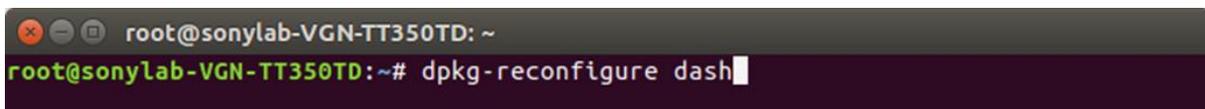
root@sonylab-VGN-TT350TD: ~
root@sonylab-VGN-TT350TD:~# apt-get install tofrodos iproute tftpd-hpa gawk gcc
git-core make net-tools libncurses5-dev zlib1g-dev libssl-dev flex bison lib32z1
lib32ncurses5 lib32stdc++6 libselinux1
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
root@sonylab-VGN-TT350TD:~# █

```

*Imagen 8 Instalación de librerías recomendadas para Ubuntu*

En caso de repetir el comando anterior se puede observar el mensaje que indica que ya cada librería ya se encuentra en la versión más reciente.

⇒ Cuando se instalan las librerías, se configura el bash (hace referencia a la forma en la que el sistema operativo interpreta los comandos, otra manera es utilizando el *dash*) en la consola por medio del comando, *dpkg-reconfigure dash*, se deshabilita el *dash* y por defecto queda seleccionado el *bash*.



```

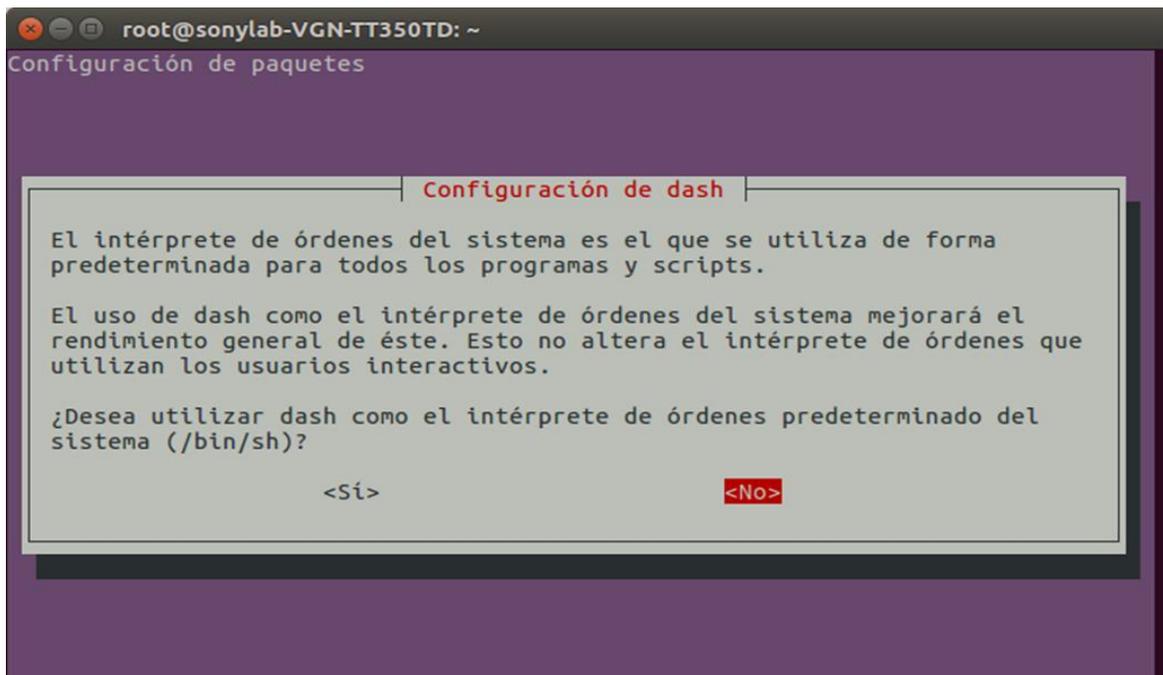
root@sonylab-VGN-TT350TD: ~
root@sonylab-VGN-TT350TD:~# dpkg-reconfigure dash █

```

*Imagen 9 Se deshabilita el dash.*

	<p style="text-align: center;">INFORME FINAL DE TRABAJO DE GRADO</p>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Al ejecutar el comando se habilita una ventana con la opción de seleccionar el *dash* como predeterminado, se selecciona la opción NO, como lo muestra la imagen 10, de esta forma se configura el *bash*.



*Imagen 10 Seleccionar el bash por defecto.*

Hasta este momento se han realizado las configuraciones del sistema operativo (Ubuntu). En el PC. Los pasos que hemos realizado pueden ir o no en el orden propuesto, pues no requieren ser secuenciales.

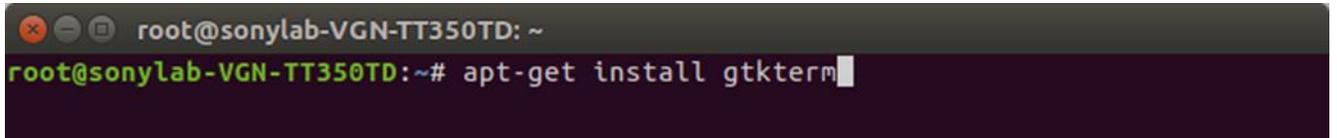
La interfaz sobre la cual se va a visualizar el sistema operativo (proyecto petalinux), en la Zedboard, será por consola, para ello se debe preparar la herramienta editor de texto, configurada a la velocidad de operación de la Zedboard.

Como la Zedboard se conecta por puerto serial con el PC, el editor de texto debe soportar las características de este tipo de comunicación, siendo velocidad 115200 y puerto /dev/ttyACM0. La aplicación para Linux que cumple con estas condiciones es gtkterm.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

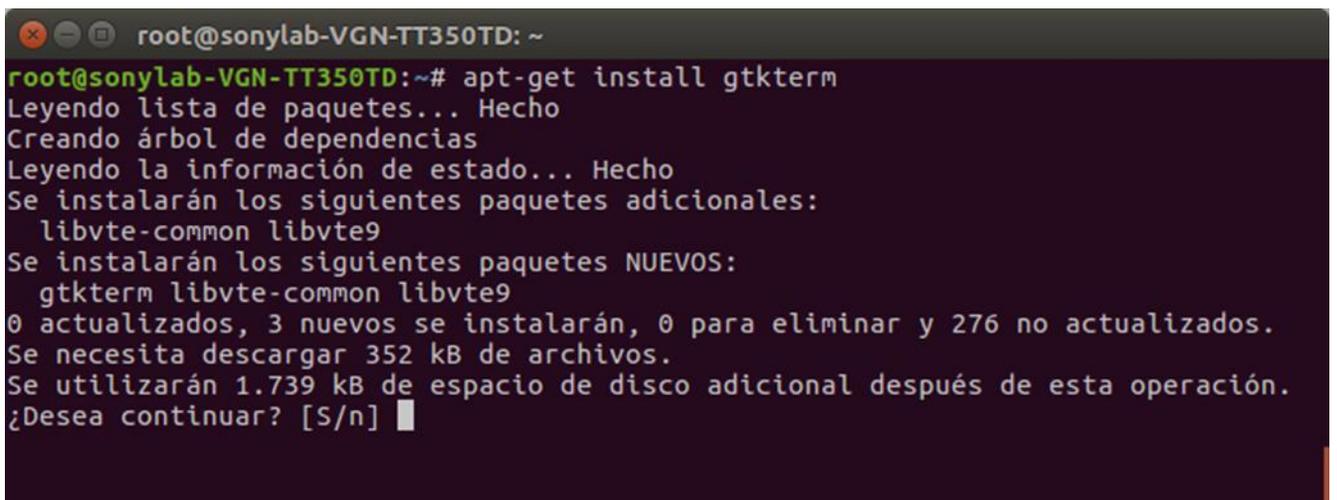
⇒ Para la instalación de la aplicación del terminal GtkTerm utilizamos el siguiente comando.

```
apt-get install gtkterm
```



```
root@sonylab-VGN-TT350TD: ~
root@sonylab-VGN-TT350TD:~# apt-get install gtkterm
```

*Imagen 11 Instalación del terminal serial para la zedboard.*



```
root@sonylab-VGN-TT350TD: ~
root@sonylab-VGN-TT350TD:~# apt-get install gtkterm
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 libvte-common libvte9
Se instalarán los siguientes paquetes NUEVOS:
 gtkterm libvte-common libvte9
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 276 no actualizados.
Se necesita descargar 352 kB de archivos.
Se utilizarán 1.739 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

*Imagen 12 Se acepta la condición del uso de memoria que requiere utilizar la descarga.*

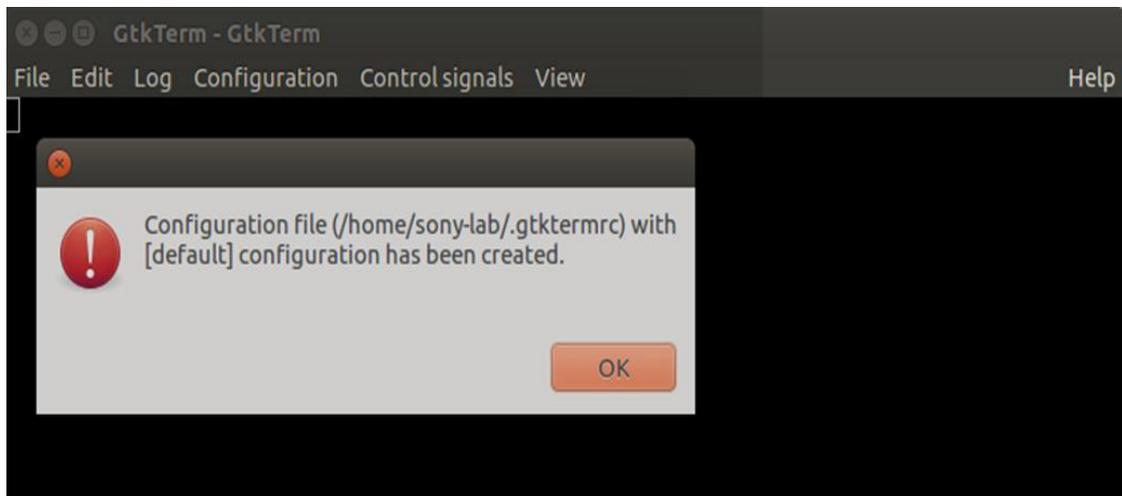
⇒ Para configurar los parámetros establecidos para la comunicación serial se ejecuta el editor de texto, se modifica el valor de velocidad y puerto, se guardan los cambios y se cierra el editor.

El comando utilizado es: gtkterm

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```
root@sonylab-VGN-TT350TD: ~
root@sonylab-VGN-TT350TD:~# gkterm
```

*Imagen 13 Comando para abrir el editor de texto.*



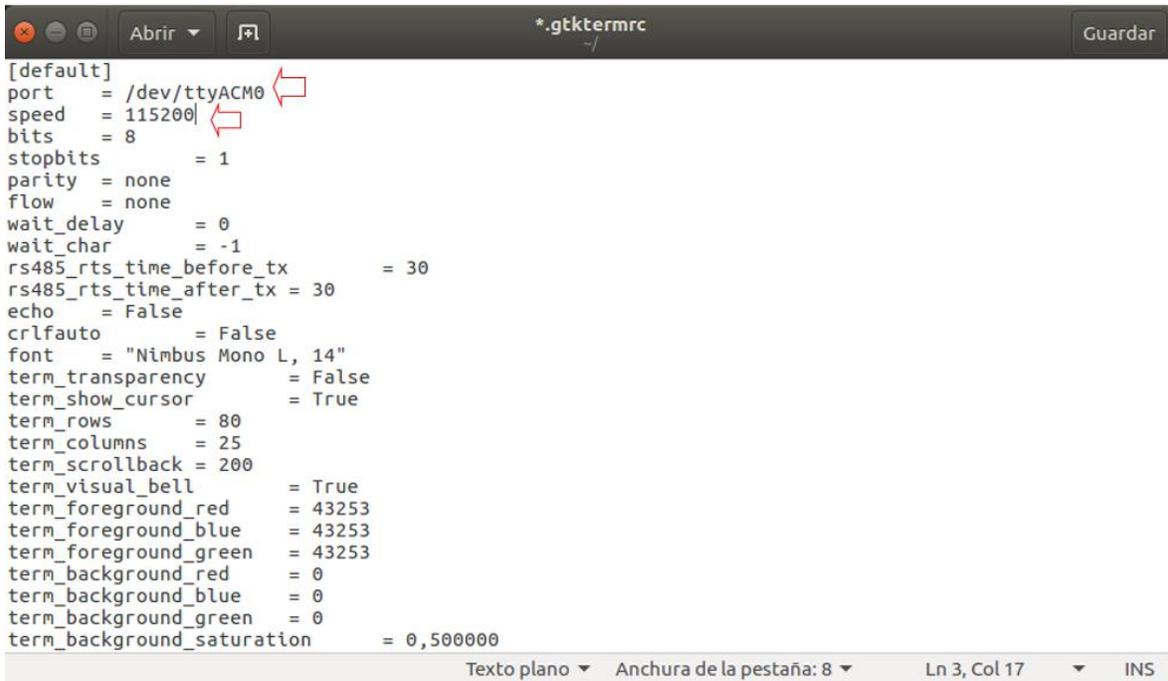
*Imagen 14 Se acepta mensaje informativo para continuar procedimiento*

Con el comando `gedit .gktermrc` editamos los valores.

```
root@sonylab-VGN-TT350TD: ~
root@sonylab-VGN-TT350TD:~# gedit .gktermrc
```

*Imagen 15 Comando para editar valores de velocidad y puerto de la consola*

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



```

[default]
port      = /dev/ttyACM0
speed     = 115200
bits      = 8
stopbits  = 1
parity    = none
flow      = none
wait_delay = 0
wait_char = -1
rs485_rts_time_before_tx = 30
rs485_rts_time_after_tx = 30
echo      = False
crlfauto  = False
font      = "Nimbus Mono L, 14"
term_transparency = False
term_show_cursor = True
term_rows = 80
term_columns = 25
term_scrollback = 200
term_visual_bell = True
term_foreground_red = 43253
term_foreground_blue = 43253
term_foreground_green = 43253
term_background_red = 0
term_background_blue = 0
term_background_green = 0
term_background_saturation = 0,500000

```

Texto plano ▾ Anchura de la pestaña: 8 ▾ Ln 3, Col 17 ▾ INS

*Imagen 16 Se ingresan los valores de velocidad y puerto en el editor de texto, se guardan los cambios y se cierra el editor .*

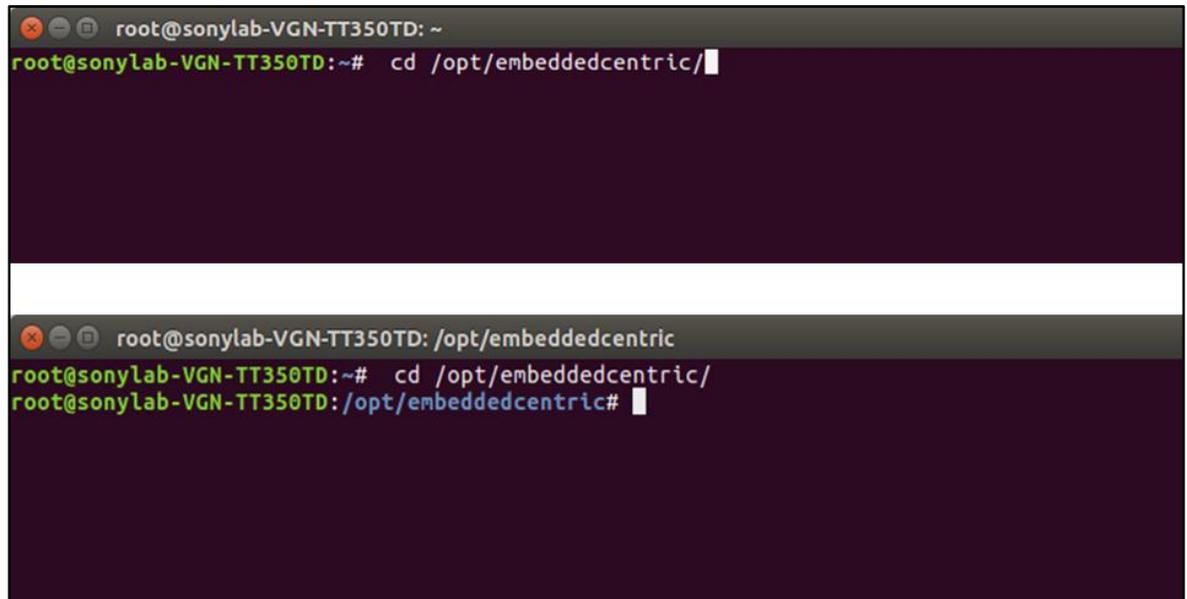
En los siguientes pasos se instalan los archivos descargados desde la página web de Xilinx, fabricante de la tarjeta zedboard, para crear el proyecto petalinux, que será la imagen del sistema operativo o archivo boot para correr desde la memoria SD en la zedboard.

⇒ Con el comando `cd`, cambiamos el directorio de ubicación sobre el cual vamos a trabajar, es decir donde tenemos la copia de los archivos.

Ejecutamos el siguiente comando.

```
cd /opt/embeddedcentric
```

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



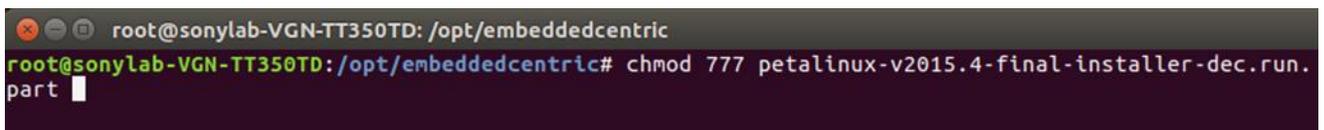
```

root@sonylab-VGN-TT350TD: ~
root@sonylab-VGN-TT350TD:~# cd /opt/embeddedcentric/

```

*Imagen 17 Cambiar al directorio opt en la carpeta llamada embeddedcentric*

⇒ El comando `chmod 777 petalinux-v2015.2.1-final-installer.run` otorga permisos de lectura y visualización de archivos a la carpeta de petalinux mencionada.



```

root@sonylab-VGN-TT350TD: /opt/embeddedcentric
root@sonylab-VGN-TT350TD:/opt/embeddedcentric# chmod 777 petalinux-v2015.4-final-installer-dec.run.
part

```

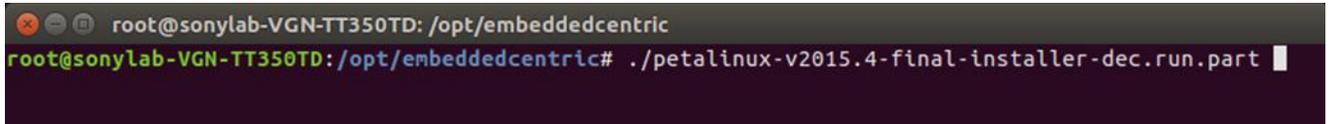
*Imagen 18 Comando chmod 777 para asignar todos los permisos sobre el archivo petalinux descargado.*

⇒ Para lanzar el instalador anteponeamos al nombre del archivo la línea “./”, quedando la línea de comando de la siguiente manera:

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

`./petalinux-v2015.4.1-final-installer.run`

*Nota: Se acepta la licencia para continuar con la instalación.*



```

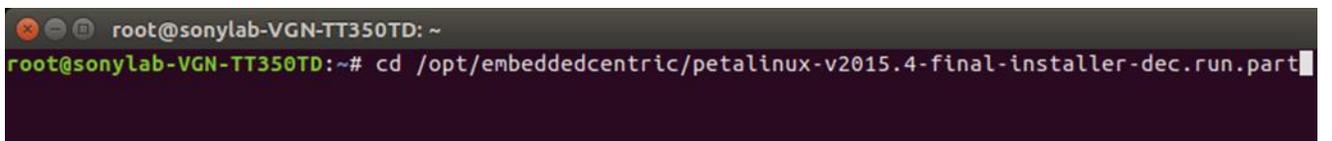
root@sonylab-VGN-TT350TD: /opt/embeddedcentric
root@sonylab-VGN-TT350TD:/opt/embeddedcentric# ./petalinux-v2015.4-final-installer-dec.run.part

```

*Imagen 19 Se ejecuta el instalador de petalinux.*

⇒ Nuevamente hacemos un cambio de directorio, pero en este caso sobre el archivo de instalación del petalinux que realizamos en el paso anterior:

```
cd /opt/embeddedcentric/petalinux-v2015.2.1-final
```



```

root@sonylab-VGN-TT350TD: ~
root@sonylab-VGN-TT350TD:~# cd /opt/embeddedcentric/petalinux-v2015.4-final-installer-dec.run.part

```

*Imagen 20 Cambio de directorio sobre el archivo petalinux instalado.*

⇒ Se habilitan permisos nuevamente con el comando `chmod 777`, a los archivos de herramientas del Ubuntu.

Utilizamos el comando: `chmod 777 ./settings.sh`

Seguidamente con el comando `source ./settings.sh` se cargan las herramientas disponibles en el PC.

Nota: Es importante además de cargar la herramienta de configuración del sistema operativo, cargar las herramientas de compatibilidad de la memoria SD. De esta manera garantizamos que al momento de crear el proyecto Petalinux, éste tenga el formato correspondiente para ser ejecutado sobre el formato SD.

Lo anterior lo realizamos con el comando:

```
source /opt/Xilinx/SDK/2014.4/settings64.sh
```

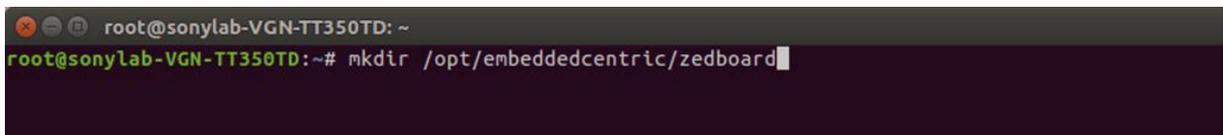
	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Se crea el proyecto petalinux, incluye los parámetros necesarios del sistema operativo que veremos reflejado en la Zedboard, así finalmente tendremos el archivo ejecutable con la imagen del SO.

⇒ Se crea una nueva carpeta en el directorio opt, dentro de la carpeta que estamos trabajando llamada embeddedcentric.

Utilizamos el siguiente comando.

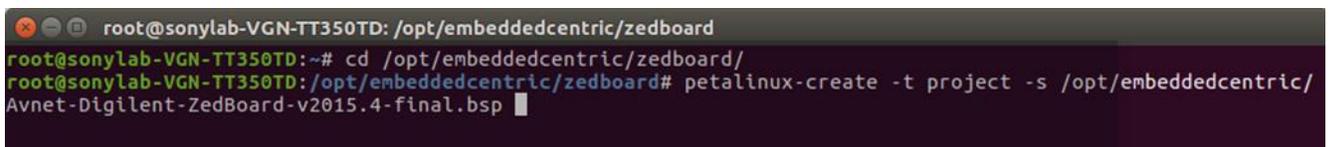
```
mkdir /opt/embeddedcentric/Zedboard
```



```
root@sonylab-VGN-TT350TD: ~
root@sonylab-VGN-TT350TD:~# mkdir /opt/embeddedcentric/zedboard
```

*Imagen 21 Carpeta con nombre zedboard, para almacenar el proyecto petalinux.*

⇒ Se cambia de directorio a la carpeta que se acaba de crear. Con el comando `petalinux-create -t project -s /opt/embeddedcentric/Avnet-Digil`, se crea el proyecto petalinux, se especifica que es un proyecto basado en tipo de archivo BSP y se indica la ubicación donde se encuentra el archivo descargado previamente.



```
root@sonylab-VGN-TT350TD: /opt/embeddedcentric/zedboard
root@sonylab-VGN-TT350TD:~# cd /opt/embeddedcentric/zedboard/
root@sonylab-VGN-TT350TD:/opt/embeddedcentric/zedboard# petalinux-create -t project -s /opt/embeddedcentric/Avnet-Digilent-ZedBoard-v2015.4-final.bsp
```

*Imagen 22 Proyecto PetaLinux, se crea indicando la ubicación de archivo tipo BSP.*

⇒ La herramienta `echo $PETALINUX`, internamente realiza un proceso para verificar el estado de las fuentes cargadas anteriormente. Se debe tener permisos de administrador para poder ejecutar el proceso, esto se verifica

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

al final del directorio con el símbolo #. De no ser así se reemplaza por la línea de comando `sudo su` o `sudo bash`.

```

root@sonylab-VGN-TT350TD: /opt/embeddedcentric/zedboard
root@sonylab-VGN-TT350TD:/opt/embeddedcentric/zedboard# echo $PETALINUX /opt/embeddedcentric/petalinux-v2015.4-final-installer-dec.run.part 3-

```

*Imagen 23 Comprobar el estado de las fuentes con el comando `echo $PETALINUX`*

⇒ Cambiamos nuevamente de directorio, con la siguiente línea de comando.

`cd /opt/embeddedcentric/zedboard/Avnet-Digilent-ZedBoard-2015.2/`

```

root@sonylab-VGN-TT350TD: /opt/embeddedcentric/zedboard
root@sonylab-VGN-TT350TD:/opt/embeddedcentric/zedboard# cd /opt/embeddedcentric/zedboard/Avnet-Digilent-ZedBoard-v2015.4-final.bsp

```

*Imagen 24 Cambiar al directorio de ubicación de la carpeta zedboard en el archivo tipo BSP, copiado anteriormente.*

En los siguientes pasos, vamos a seleccionar las diferentes aplicaciones del sistema operativo para operar sobre la zedboard, principalmente la interacción con los periféricos de la tarjeta y puertos de comunicación.

La aplicación *gpio-demo*, permite que haya comunicación entre los periféricos como lectura y escritura en los switches, *leds* y pulsadores.

⇒ Con la línea de comando `petalinux-config -c rootfs`, se abre la ventana de configuración y seleccionados las aplicaciones que se van a utilizar:

1. *gpio-demo*
2. Filesystem Packages. En esta segunda aplicación seleccionamos:
  - a. Console/Network
  - b. dropbear
  - c. dropbear-openssh-sftp-server.

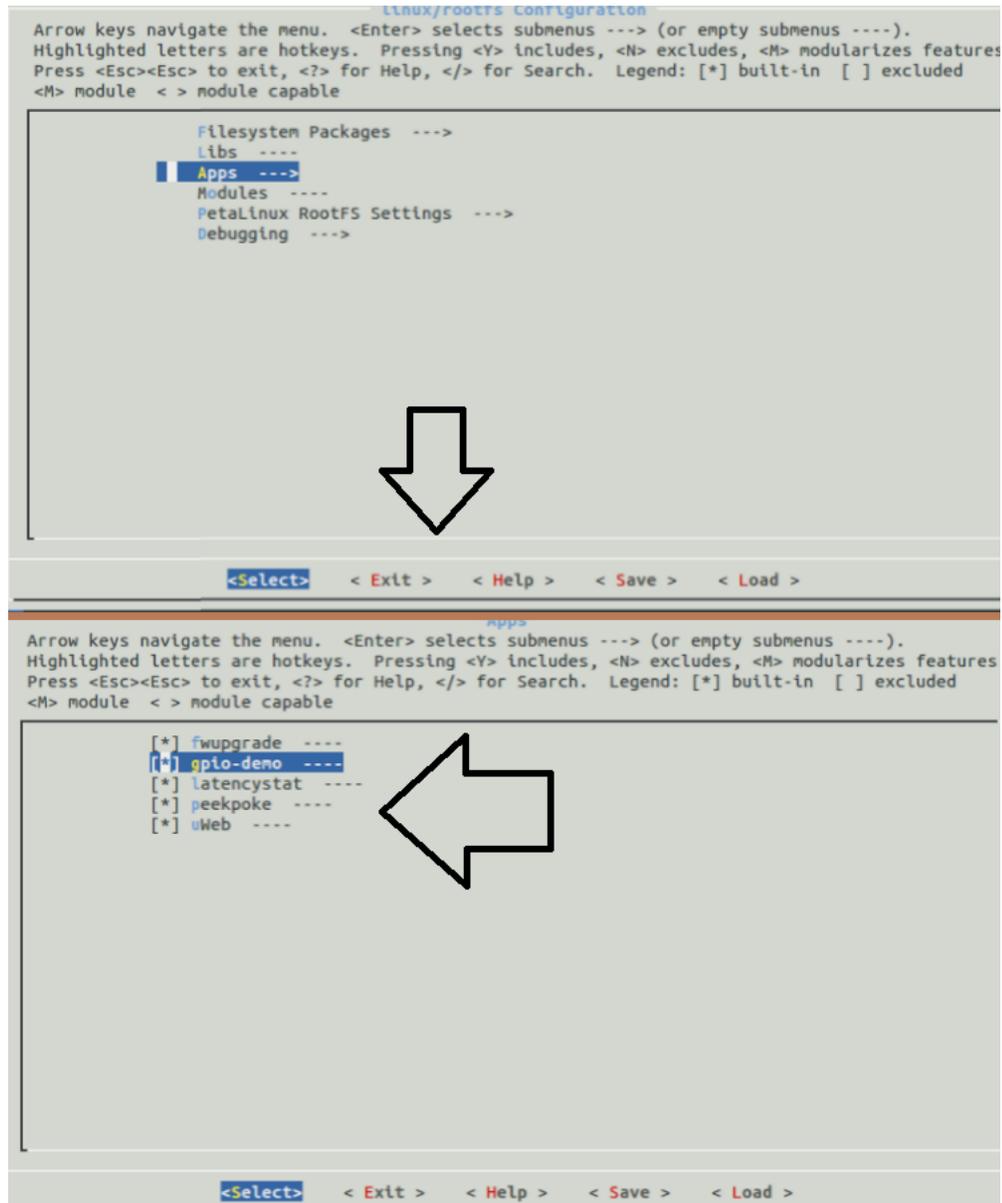


Imagen 25 Aplicación 1. gpio-demo.

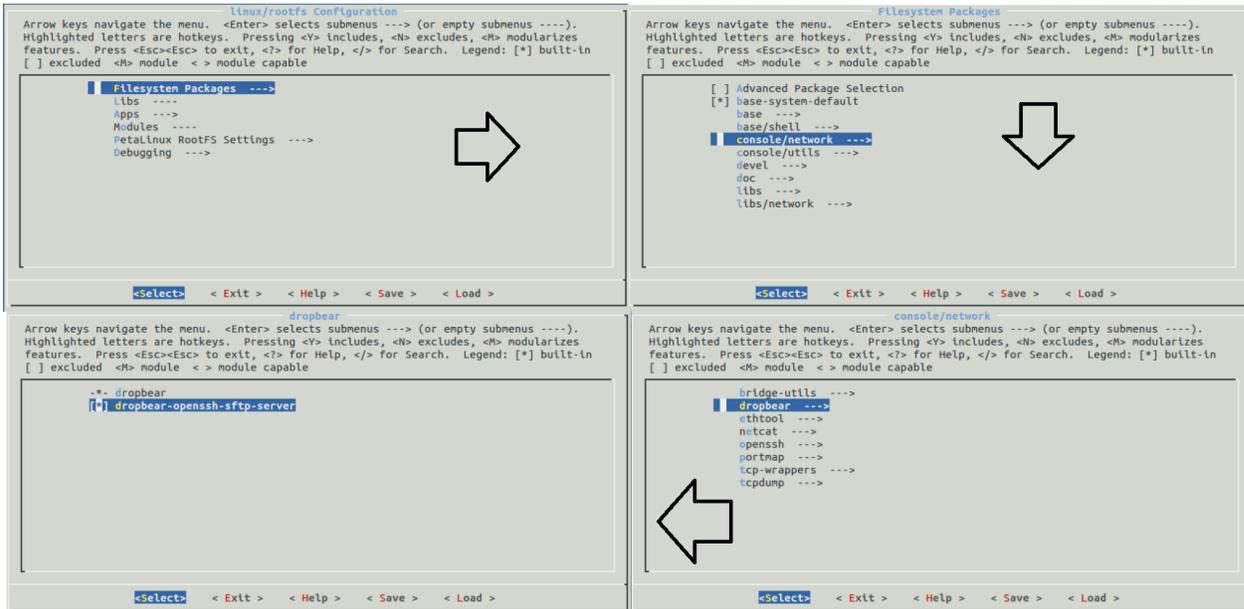


Imagen 26 Aplicación 2. Filesystem Packages

⇒ Para la configuración de las herramientas de red dentro de la ruta Filesystem Packages -> base -> busybox -> busybox-http

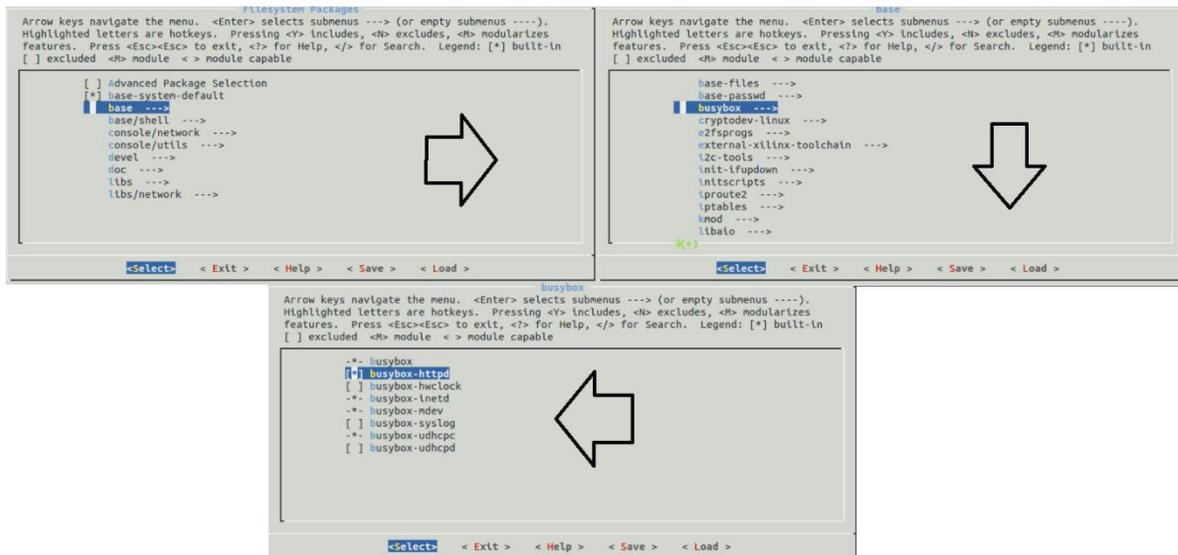


Imagen 27 Configuración de las herramientas de red http

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

⇒ Una vez configuradas las aplicaciones, se genera el archivo de la imagen del sistema operativo que se va a ejecutar desde la tarjeta zedboard en una memoria de almacenamiento SD.

De manera secuencial se lanzan las siguientes líneas de comando.

- `cd /opt/embeddedcentric/zedboard/Avnet-Digilent-ZedBoard-2015.2/`
- `petalinux-build`
- `petalinux-package -boot -format BIN -force -fsbl /opt/embeddedcentric/zedboard/ Avnet-Digilent-ZedBoard-v2015.4/images/linux/zynq_fsbl.elf -fpga /opt/embeddedcentric/zedboard/ Avnet-Digilent-ZedBoard-v2015.4/images/linux/download.bit -uboot`

⇒ En este momento ya tenemos el archivo BOOT.BIN, lo podemos comprobar en el directorio opt.



*Imagen 28 Archivo BOOT.BIN.*

⇒ Este archivo es uno de los que vamos a copiar en la tarjeta SD que incluye el kit de la zedboard para correr el sistema operativo desde ésta. El segundo archivo corresponde a la imagen, con el nombre *image.ud*, lo ubicamos en el directorio `embeddedcentric/zedboard/Avnet-Digilent-ZedBoard-2015.2/images/Linux`.

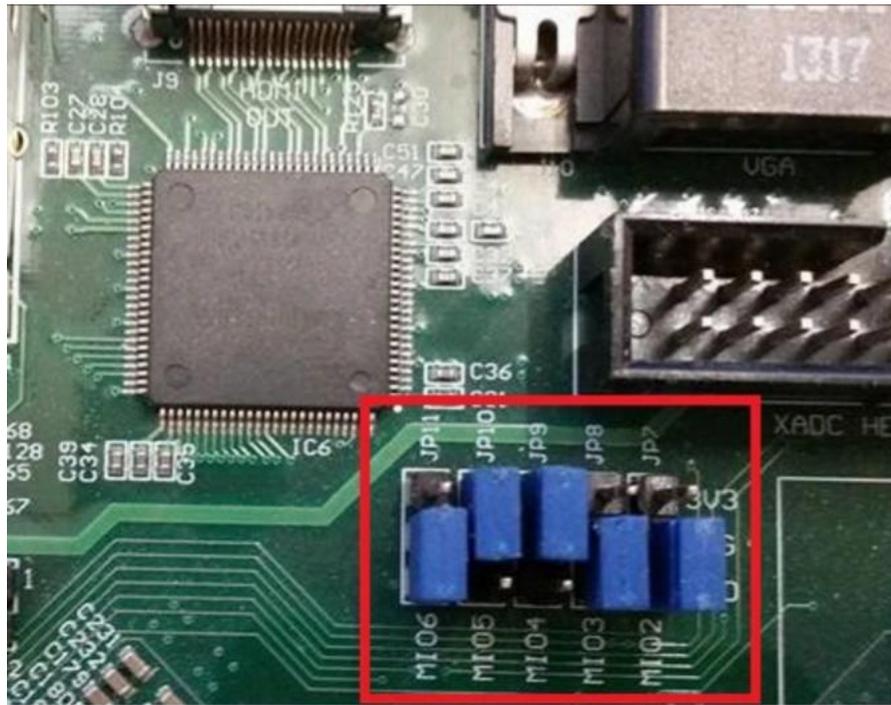
Lo que debemos tener en cuenta en la zedboard antes de insertar la memoria SD, es la conexión de los *jumpers* de configuración para que sea compatible.

JP9(MIO[4]) -----High

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

JP10(MIO[5]) -----High

Tal como lo muestra la siguiente imagen.



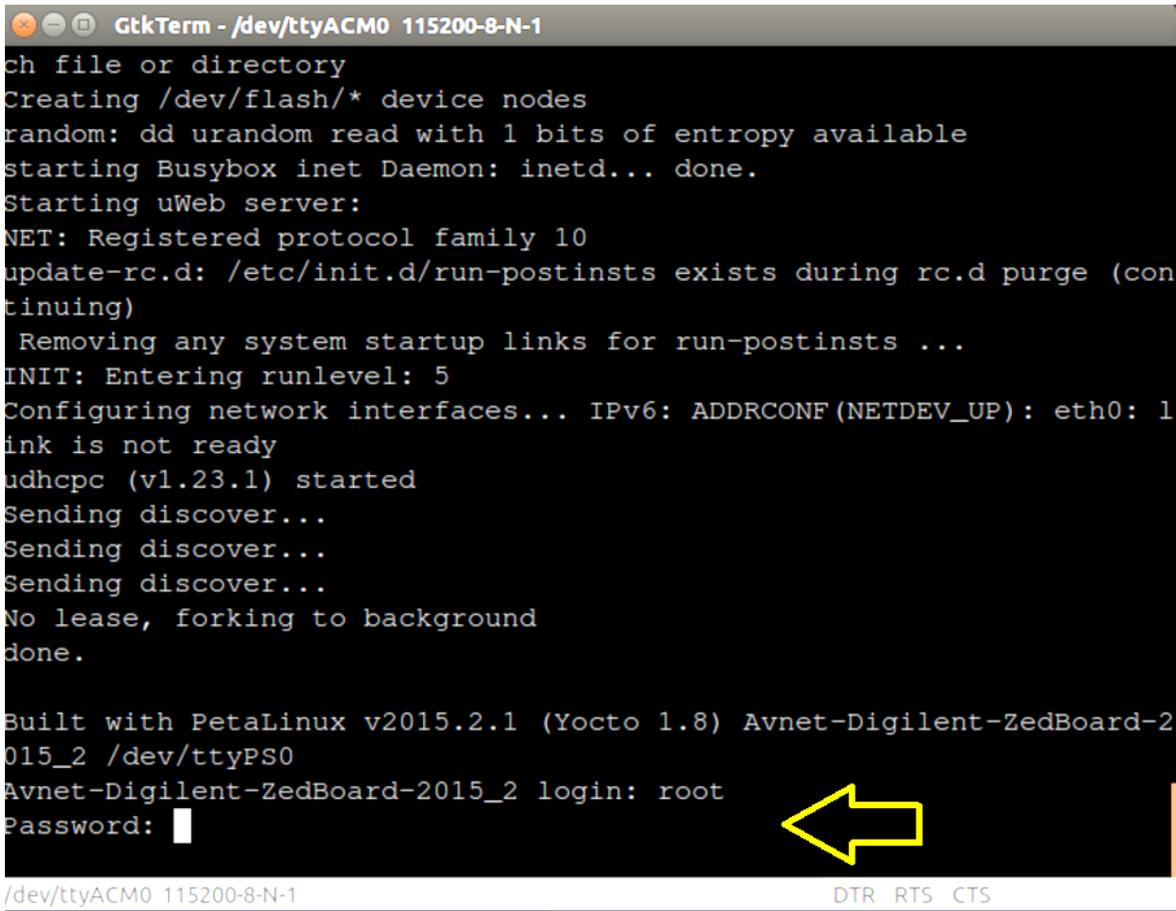
*Imagen 29 Conexión de jumpers de configuración para conectar la memoria SD.*

- ⇒ Conectamos la tarjeta zedboard al PC por medio del puerto serial – USB (UART).
- ⇒ Se inserta la memoria SD a la zedboard.
- ⇒ Se enciende la zedboard y oprimimos el botón pulsador BTN7 PS-RST, éste cumple la función de reiniciar el sistema operativo que tenemos alojado en la SD.
- ⇒ En la consola del PC ejecutamos la línea de comando:
  - `chmod 666 /dev/ttyACM0`
- Seguidamente la línea de comando:
  - `gtkterm`

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Con este último en el PC vamos a visualizar una nueva consola, por medio de esta vamos a visualizar el Linux que estamos ejecutando por medio de la memoria SD en la zedboard.

El progreso de inicio va a solicitar un usuario y contraseña, para ambos casos vamos a ingresar: root



```

ch file or directory
Creating /dev/flash/* device nodes
random: dd urandom read with 1 bits of entropy available
starting Busybox inet Daemon: inetd... done.
Starting uWeb server:
NET: Registered protocol family 10
update-rc.d: /etc/init.d/run-postinsts exists during rc.d purge (continuing)
Removing any system startup links for run-postinsts ...
INIT: Entering runlevel: 5
Configuring network interfaces... IPv6: ADDRCONF (NETDEV_UP): eth0: link is not ready
udhcpd (v1.23.1) started
Sending discover...
Sending discover...
Sending discover...
No lease, forking to background
done.

Built with PetaLinux v2015.2.1 (Yocto 1.8) Avnet-Digilent-ZedBoard-2015_2 /dev/ttyPS0
Avnet-Digilent-ZedBoard-2015_2 login: root
Password: █

```

*Imagen 30 Consola del Linux ejecutado desde la memoria SD.*

De esta manera damos por terminada esta primera etapa de configuración, ya contamos con el sistema operativo corriendo en la zedboard desde la memoria SD.

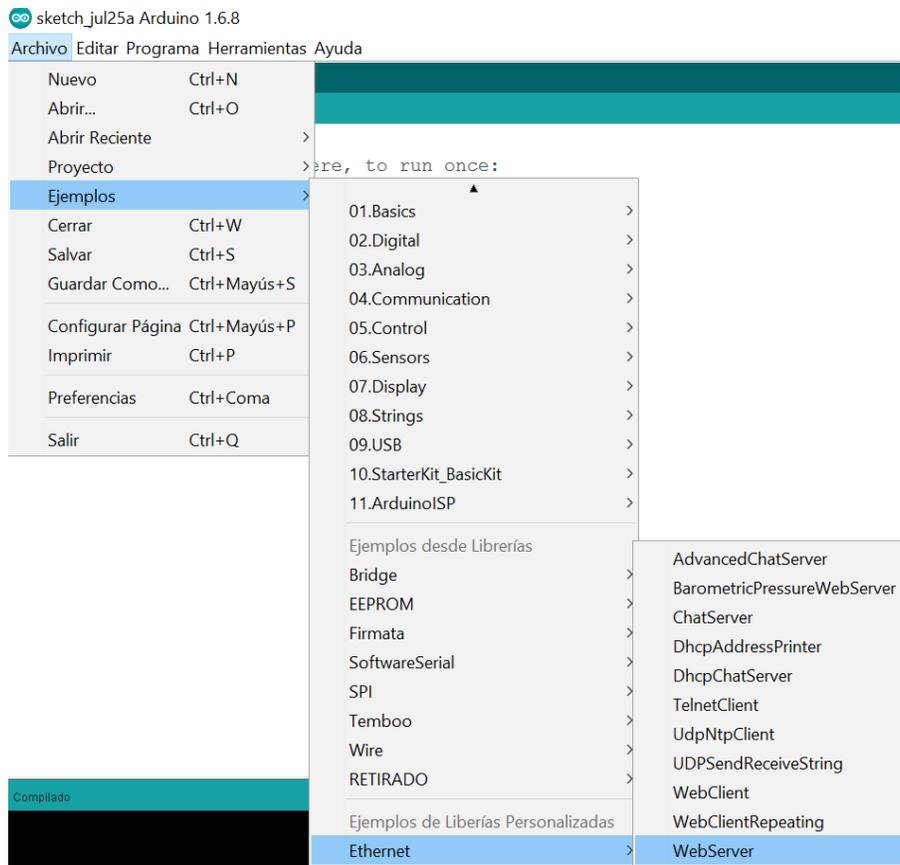
### 3.2. Instrucciones de conexión y programación del Arduino para la Adquisición de datos del sensor.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Luego de contar con el archivo ejecutable se realiza la configuración de la tecnología Ethernet para el sistema de adquisición de datos.

Este procedimiento se hace directamente en el Arduino. Para este proyecto se utiliza la referencia Arduino uno con el modulo adicional Shield Ethernet, ya que éste incluye el puerto RJ-45.

En el equipo PC se instala el programa Arduino, que se descarga directamente de la página del fabricante. Una vez instalado nos basamos en el programa ejemplo webserver, ubicado en el grupo Ethernet.

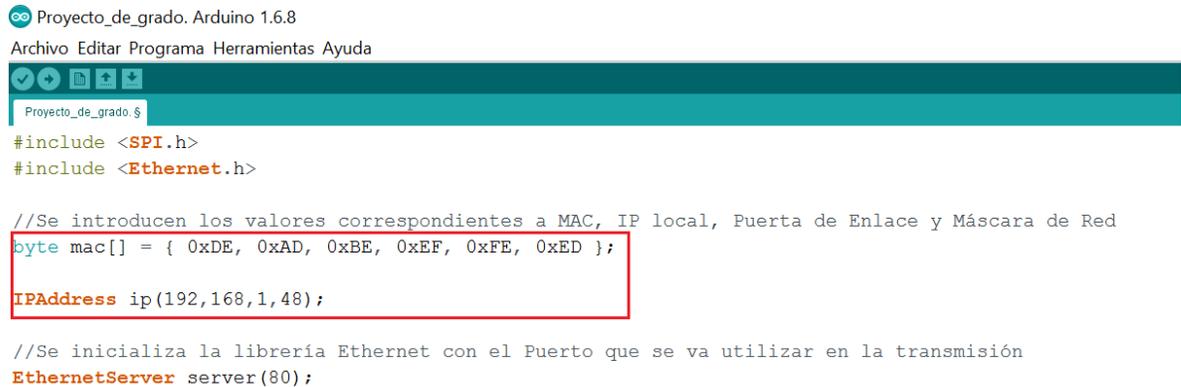


*Imagen 31 Cargar ejemplo webserver desde el programa Arduino.*

Sobre este código se realizan las modificaciones necesarias para la lectura del sensor LM35 conectado al Arduino, entrada análoga A0.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

⇒ Asignar dirección IP al Arduino. Debemos garantizar que el valor de MAC se encuentre habilitado, ya que en una red LAN es necesaria la dirección IP y MAC del dispositivo.



```

Proyecto_de_grado. Arduino 1.6.8
Archivo Editar Programa Herramientas Ayuda
Proyecto_de_grado. §
#include <SPI.h>
#include <Ethernet.h>

//Se introducen los valores correspondientes a MAC, IP local, Puerta de Enlace y Máscara de Red
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,1,48);

//Se inicializa la librería Ethernet con el Puerto que se va utilizar en la transmisión
EthernetServer server(80);

```

*Imagen 32 Asignar dirección IP al Arduino.*

⇒ Se escala el valor del dato del sensor de acuerdo a la información recibida por el convertor Análogo digital (ADC) del Arduino, esta modificación se realiza en la línea de código como lo muestra la siguiente imagen, declarando una variable tipo entero, "INT", que corresponde al valor que hay en el puerto A0 del Arduino donde está conectado el sensor.



	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.3. Rutina de procesos

---

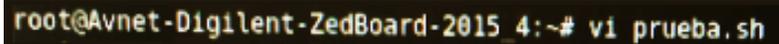
Finalmente realizamos la rutina de procesos para observar en consola la información de datos obtenidos del sensor y transmitidos a través del protocolo Ethernet.

Desde el terminal gkterm, se configura una rutina de procesos en la cual nos direcciona a la información publicada en la dirección IP, asignada al sistema de adquisición de datos (Arduino), con el valor del sensor. La herramienta que se utiliza es un editor de texto llamado *vi*, dentro del editor vamos a copiar los comandos para visualizar la lectura del dato del sensor.

La sintaxis para crear el archivo en el editor es: *vi nombre\_del\_archivo.sh*, al nombre le agregamos la extensión *.sh*, ésta hace referencia al tipo de archivo, que para esta oportunidad va a ser tipo ejecutable.

Después de creado el archivo con escribir, *./ nombre del mismo*, entramos al editor.

⇒ `vi prueba.sh`

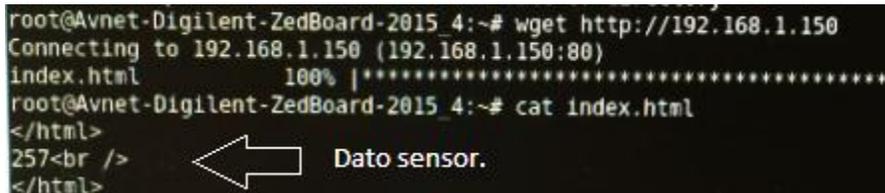


```
root@Avnet-Digilent-ZedBoard-2015_4:~# vi prueba.sh
```

*Imagen 35 Crear archivo vi para rutina de procesos.*

El arduino tiene configurada una dirección IP, para acceder al dato lo hacemos con el comando `wget http://direccion_ip`.

⇒ `wget http://192.168.1.150`



```
root@Avnet-Digilent-ZedBoard-2015_4:~# wget http://192.168.1.150
Connecting to 192.168.1.150 (192.168.1.150:80)
index.html 100% |*****
root@Avnet-Digilent-ZedBoard-2015_4:~# cat index.html
</html>
257<br />
</html>
```

*Imagen 36 Conexión con la IP del sistema de adquisición de datos con el comando wget.*

Si requerimos un segundo intento de conexión con el sistema de adquisición de datos, nos va a generar un error, como lo muestra la siguiente imagen. Para

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

poderlo corregir es necesario respaldar el dato en otro archivo diferente al generado (index.html), en el siguiente paso se indica como crearlo, con formato de archivo de texto y lo nombramos 'datosensor.txt' utilizando el comando *touch*.

```
root@Avnet-Digilent-ZedBoard-2015_4:~# wget http://192.168.1.150
Connecting to 192.168.1.150 (192.168.1.150:80)
wget: can't open 'index.html': File exists
```

*Imagen 37 Error de conexión con el comando wget.*

⇒ touch datosensor.txt

```
root@Avnet-Digilent-ZedBoard-2015_4:~# touch datosensor.txt
```

*Imagen 38 Crear archivo de texto con el nombre 'datosensor.txt' para almacenar el dato leído del sensor*

⇒ Para guardar el dato en el archivo creado anteriormente, utilizamos el comando *echo*.

```
echo index.html > datosensor.txt
```

Para almacenar mas datos sin borrar el anterior utilizamos:

```
echo index.html >> datosensor.txt
```

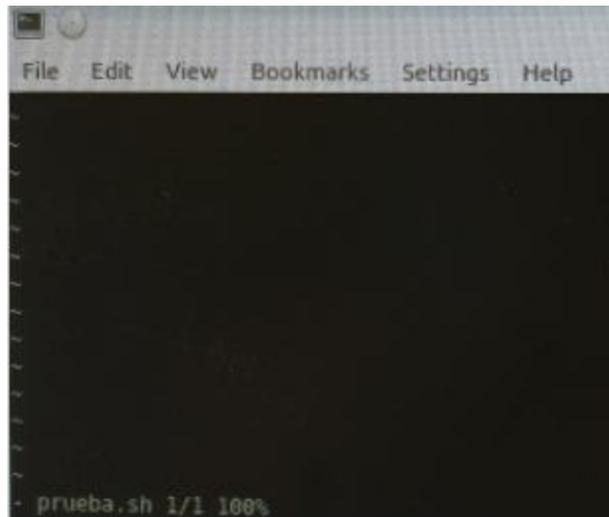
⇒ Con el comando *cat* leemos la información del archivo requerido, sea *index.html* o *datosensor.txt*

```
root@Avnet-Digilent-ZedBoard-2015_4:~# touch datosensor.txt
root@Avnet-Digilent-ZedBoard-2015_4:~# echo index.html > datosensor.txt
root@Avnet-Digilent-ZedBoard-2015_4:~# cat index.html
</html>
257<br />
</html>
root@Avnet-Digilent-ZedBoard-2015_4:~# cat datosensor.txt
index.html
```

*Imagen 39 Lectura de dato con el comando cat.*

⇒ Con el comando *./prueba.sh* abrimos el archivo ejecutable para establecer la rutina de procesos.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

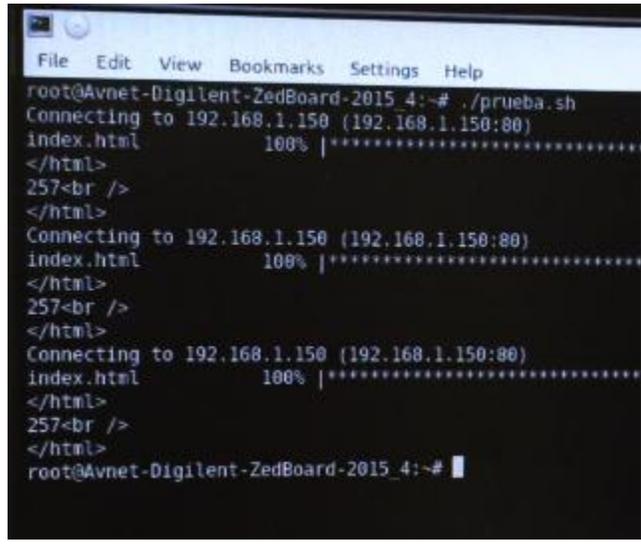


*Imagen 40 Editor de texto del archivo pruebas.sh*

- ⇒ Con la tecla “Insert” habilitamos el editor para escribir.
- ⇒ Para guardar utilizamos la tecla “ESC”, seguidamente insertamos “:” y finalmente la tecla “x”.

Una vez editada la rutina en la consola *gtkterm* la cargamos nuevamente con la línea con el comando, “./prueba.sh” y podemos observar cómo se van ejecutando los procesos que hemos incluido.

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



*Imagen 41 Rutina de proceso en ejecución.*

Para la imagen anterior en la rutina se asignaron tres lecturas con un intervalo de tiempo en cada una.

Este intervalo lo agregamos con el comando “sleep 2”, que indica que después de la lectura espere 2 segundos y vuelva a realizar el proceso.



*Imagen 42 Código de la rutina para la lectura de tres datos almacenados en index.html*

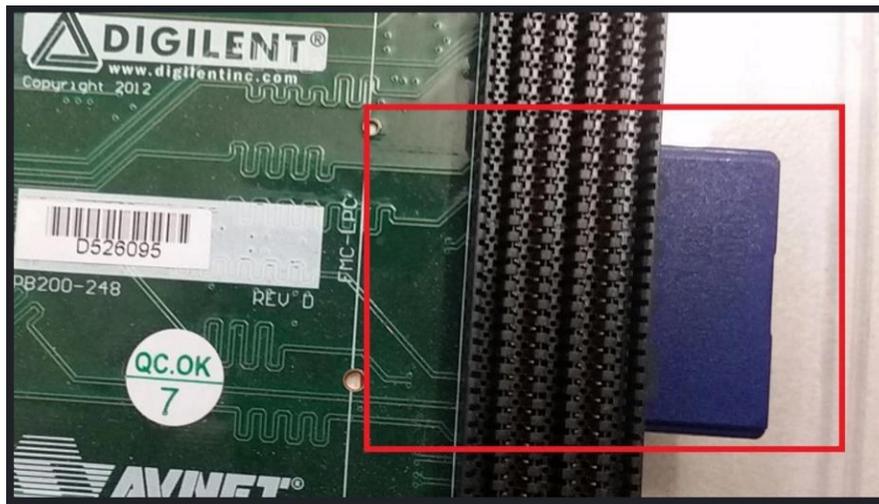
	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 4. RESULTADOS Y DISCUSIÓN

Los archivos creados para la imagen del sistema operativo embebido se guardan en la memoria SD para ser ejecutados desde la Zedboard.



*Evidencia 1 Archivos de la imagen ejecutable del sistema operativo.*



*Evidencia 2 Conexión física de la memoria SD en la tarjeta Zedboard.*

En la tarjeta Zedboard se ejecuta el sistema operativo embebido Linux-Ubuntu. A continuación se observa que es necesario ingresar el Password de acceso definido en las configuraciones determinadas al momento de crear la imagen para el usuario "root".

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

GtkTerm - /dev/ttyACM0 115200-8-N-1
ch file or directory
Creating /dev/flash/* device nodes
random: dd urandom read with 1 bits of entropy available
starting Busybox inet Daemon: inetd... done.
Starting uWeb server:
NET: Registered protocol family 10
update-rc.d: /etc/init.d/run-postinsts exists during rc.d purge (con
tinuing)
Removing any system startup links for run-postinsts ...
INIT: Entering runlevel: 5
Configuring network interfaces... IPv6: ADDRCONF (NETDEV_UP): eth0: 1
ink is not ready
udhcpd (v1.23.1) started
Sending discover...
Sending discover...
Sending discover...
No lease, forking to background
done.

Built with PetaLinux v2015.2.1 (Yocto 1.8) Avnet-Digilent-ZedBoard-2
015_2 /dev/ttyPS0
Avnet-Digilent-ZedBoard-2015_2 login: root
Password: █

```

DTR RTS CTS

*Evidencia 3 Visualización del sistema operativo desde el terminal de consola ejecutado la zedboard desde la memoria SD.*

En el terminal de consola se evidencia la dirección IP asignada al Arduino y el valor del dato leído a través del sistema de adquisición de datos y el *script* elaborado.

Con la línea “Connecting to 192.168.1.150 (192.168.1.150:80)” que se visualiza en la *Evidencia 3* se comprueba que existe comunicación entre la Zedboard y la dirección IP asignada al Arduino. El puerto 80 hace referencia a la comunicación a través del protocolo Ethernet.

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

File Edit View Bookmarks Settings Help
root@Avnet-Digilent-ZedBoard-2015_4:~# ./prueba.sh
Connecting to 192.168.1.150 (192.168.1.150:80)
index.html 100% |*****
</html>
257<br />
</html>
Connecting to 192.168.1.150 (192.168.1.150:80)
index.html 100% |*****
</html>
257<br />
</html>
Connecting to 192.168.1.150 (192.168.1.150:80)
index.html 100% |*****
</html>
257<br />
</html>
root@Avnet-Digilent-ZedBoard-2015_4:~#

```

*Evidencia 4 Lectura y transmisión del dato del sensor.*

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

---

- Se configura una distribución del sistema operativo Linux Ubuntu, para ser ejecutada desde la memoria SD en la tarjeta Zedboard. Se identifica que la zedboard es el equivalente al PC o Host de una red LAN convencional, pues cuenta con un procesador, puertos de comunicación y periféricos integrados físicamente que pueden ser utilizados según se requiera.
- A través de la consola gtkterm, configurada como herramienta en la imagen del sistema operativo, se interconecta una red LAN entre dos dispositivos electrónicos, la tarjeta Zedboard en sistemas embebidos basados en FPGA y el Arduino en sistema de adquisición de datos para un sensor.
- Después de generar la imagen del sistema operativo y se requiera realizar una modificación en las aplicaciones o fuentes de Linux, es necesario volver a cargar la configuración del archivo imagen. Por ejemplo si creamos una imagen con solo la aplicación *gpio*, que envía y lee datos de los switches, led y push button, y posteriormente se requiera agregar la aplicación que permite la configuración de red, es necesario volver a realizar el proceso completo de crear la imagen y agregar los cambios, reemplazando el archivo BOOT.BIN e imagen.ud en la memoria SD.
- Se recomienda tener clara la ubicación en el PC de la fuente que carga Xilinx, para que el archivo quede compatible con el formato de memoria SD. Pues de continuar con el proceso sin esta característica no es posible generar el archivo de imagen de sistema operativo.
- A futuro se puede estudiar la forma de mejorar la rutina de procesos, utilizando un editor de texto más completo y con más herramientas de visualización, como por ejemplo Python.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## REFERENCIAS

---

Arduino.cc. (2016). Arduino - ArduinoBoardUno. [Online] Available at:  
<https://www.arduino.cc/en/Main/ArduinoBoardUno> [Accessed 8 Jul. 2016].

Franco, O. (2012). Diseño y programación de sistemas embebidos con el núcleo microblaze.  
 [Place of publication not identified]: Editorial Acad Mica Espa.

Tanenbaum, A. and Morales Peake, D. (1997). Redes de computadoras. México: Prentice-Hall Hispanoamericana.

Xilinx.com. (2016). Downloads. [online] Available at:  
<http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html> [Accessed 26 Jul. 2016].

Zedboard.org. (2016). ZedBoard | Zedboard. [online] Available at:  
<http://zedboard.org/product/zedboard> [Accessed 20 Jul. 2016].

Embedded Centric. (2015). Embedded Operating Systems. [online] Available at:  
<https://embeddedcentric.com/embedded-operating-systems/> [Accessed 26 Jul. 2016].

Arduino.cc. (2016). Arduino - Home. [online] Available at: <https://www.arduino.cc/>  
 [Accessed 26 Jul. 2016].

Xilinx.com. (2016). Downloads. [online] Available at:  
<http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html> [Accessed 26 Jul. 2016]

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## APÉNDICE

---

Apéndice A: Escalar el valor leído del sensor en el sistema de adquisición de datos con el Arduino uno.

En el proyecto se utilizó el sensor de temperatura LM35, este sensor entrega un valor de medición de 1°C por cada mili voltio mV. El sistema de adquisición de datos, Arduino, internamente tiene el modulo conversor análogo digital ADC, convierte el voltaje de entrada de 0 voltios valor mínimo a 5 voltios valor máximo, en valores enteros entre 1-1024, esto porque el Arduino en cada entrada análoga tiene un dato de 10 bits.

Con un dato de n bits se representan hasta  $2^n$  valores digitales =  $2^{10} = 1024$ .

0 voltios equivale al valor digital 1 en el Arduino después de la conversión análogo digital.

- 5 voltios equivale al valor digital 1024 en el Arduino después de la conversión análogo digital.

Para determinar qué valor de voltaje corresponde la lectura del dato a la salida del sensor, se utiliza la siguiente ecuación:

$$\text{Voltaje} = \frac{\text{valor ADC}}{\text{Valor Maximo ADC}} * \text{valor maximo de voltaje en el puerto del arduino}$$

$$\text{Voltaje} = \frac{\text{valor ADC}}{1024} * 5$$

Como el valor entregado por el sensor están en relación de mV.

$$\text{Voltaje mV Sensor} = \frac{\text{valor ADC}}{1024} * 5000$$

Esta relación se agrega en el código de Arduino de la siguiente manera.

```
int sensorReading = analogRead(A0); // En la variable sensorReading se
    almacena el valor leído en el puerto A0 del Arduino.

float mV = (sensorReading/1024.0)*5000;
float Celcius = mV/10;
```

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## Apéndice B: Código Arduino para adquisición de datos del sensor Lm 35.

```

#include <SPI.h>
#include <Ethernet.h>

//Se introducen los valores correspondientes a MAC, IP local, Puerta de Enlace y
Máscara de Red
//byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

IPAddress ip(192,168,1,48);

//IPAddress gateway(192, 168, 1, 1);

//IPAddress subnet(255, 255, 255, 0);

//Se inicializa la librería Ethernet con el Puerto que se va utilizar en la
transmisión
EthernetServer server(80);

void setup() {
  // Se establece comunicación con el monitor serial (para ver si el sistema está
funcionando correctamente)
  Serial.begin(9600);

  // while (!Serial) {
  //   ; // Se espera a que se conecte el puerto serie (solo para Arduino
Leonardo)
  // }

  // Comienzo de la conexión
  //Ethernet.begin(mac, ip);//gateway, subnet);
  server.begin();
  //Se muestra por el monitor serial que la conexión está establecida
  Serial.print("server is at ");
  Serial.println(Ethernet.localIP());
}

void loop() {
  // Se espera a que alguien acceda a la página web
  EthernetClient client = server.available();
  if (client) {
    Serial.println("new client");
    // una petición http termina con una línea en blanco
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        //Si se llega al final de la línea se recibe un nuevo carácter de
//línea y si se trata de una línea en blanco la petición http ha
terminado
        //así que se envía la respuesta
        if (c == '\n' && currentLineIsBlank) {
          // cabecera típica http
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");

```

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

    client.println("Connection: close"); // se cierra la conexión una vez
se ha respondido a la petición
    client.println("Refresh: 5"); // se refresca la página automáticamente cada
5 segundos
    client.println();
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");

    // se muestran en la web los valores de las entradas analógicas
/* for (int analogChannel = 0; analogChannel < 6; analogChannel++) {
    int sensorReading = analogRead(analogChannel);
    client.print("analog input ");
    client.print(analogChannel);
    client.print(" is ");
    client.print(sensorReading);
    client.println("<br />");
}*/

    int sensorReading = analogRead(A0);

//client.print("Entrada Analoga 0 es: ");

    float mV = (sensorReading/1024.0)*5000;
    float Celcius = mV/10;
    client.print(Celcius);
    client.println("<br />");

    client.println("</html>");
    break;
}
if (c == '\n') {
    // se comienza una nueva línea
    currentLineIsBlank = true;
}
else if (c != '\r') {
    // se ha obtenido un carácter en la línea actual
    currentLineIsBlank = false;
}
}
}
// se le da tiempo al navegador para recibir los datos
delay(1);
// se cierra la conexión
client.stop();
Serial.println("client disconnected");
}
}

```

ITM Institución Universitaria		MODALIDAD TRABAJO DE GRADO PRODUCTO OBTENIDO EN TALLERES O LABORATORIOS DEL ITM			Código	FDE 146
Registro de actividades y cumplimiento de horas / Talleres o Laboratorios de DOCENCIA					Versión	02
					Fecha	2015-09-30
1030559506						
Documento de identidad:						
Nombre completo del estudiante:						
Programa académico ITM:						
Nombre completo del Docente Asesor:						
Fecha de iniciación del producto (aaaa/mm/dd):						
Nombre Taller o Laboratorio:						
Campus:						
YURI SOLANDY HOYOS MARIN						
INGENIERIA ELECTRONICA						
LUIS FERNANDO CASTAÑO LONDOÑO						
Fecha de terminación del producto (aaaa/mm/dd):						
SISTEMAS DE CONTROL Y ROBOTICA						
LABORATORIO BLOQUE O						
FRATERNIDAD						
Fecha	Actividad desempeñada por el estudiante	Hora Ingreso	Hora salida	Total horas	Firma Laboratorista	Firma Estudiante
7 Marzo 2016	Introducción al uso de la tarjeta zedboard Xilinx Zynq@-7000, se explora informacion en el manual en cuanto a puertos que tiene el dispositivo y aplicaciones	18:00	20:00	2	Luis Flauter L	[Firma]
10 Marzo 2016	El profesor explica como programar la tarjeta zedboard desde el vivado. Se realizan pruebas	18:00	20:00	2	Luis Flauter L	[Firma]
14 Marzo 2016	Se explora la pagina de embeddedcentric.com, se empieza a trabajar la guia embedded operating systems, para crear la imagen de un sistema operativo, linux, ejecutable para tarjeta SD que trae incluida la tarjeta zedboard. Se realiza descarga de instaladores desde la pagina del fabricante para los pasos propuestos en la guia.	11:00	15:00	4	Luis Flauter L	[Firma]
28 Marzo 2016	Se avanza en la primera etapa de configuracion de linux en el equipo de computo del laboratorio, estudiando los comandos de consola de linux, se crean las carpetas y se almacenan los archivos descargados de la pagina del fabricante en el directorio opt.	18:00	20:00	2	Luis Flauter L	[Firma]
4 Abril 2016	Se estudia y se realiza la configuracion del segundo paso para la compilacion de un proyecto petalinux, se descargan librerias y se actualizan las politicas en el equipo de computo.	18:00	20:00	2	Luis Flauter L	[Firma]
9 Abril 2016		11:00	15:00	4	Luis Flauter L	[Firma]
11 Abril 2016	Se estudia, se descarga y se configura la terminal gikterm, para la comunicacion de la tarjeta zedboard con el equipo de computo del laboratorio.	18:00	20:00	2	Luis Flauter L	[Firma]
16 Abril 2016	Se ejecuta el archivo petalinux para extraer la fuentes y codigos para la configuracion del proyecto petalinux, que ejecuta el sistema operativo para luego crear la imagen ejecutable, es decir el archivo boot.	11:00	15:00	4	Luis Flauter L	[Firma]
18 Abril 2016	Se crea el nombre al proyecto petalinux, se configuracion las aplicaciones del sistema operativo a utilizar, como gpio-demo, que permite la interaccion con la tarjeta zedboard desde la consola y el terminal gikterm	18:00	20:00	2	Luis Flauter L	[Firma]
23 Abril 2016		11:00	15:00	4	Luis Flauter L	[Firma]
25 Abril 2016	Se comienza a crear la imagen del sistema operativo en formato sdk, y al momento de compilar genera error, que solicita la ubiacion de un complemento para la imagen sdk.	18:00	20:00	2	Luis Flauter L	[Firma]

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-27

FIRMA ESTUDIANTE: 

FIRMA ASESOR: 

FECHA ENTREGA: 08-08-2016

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD \_\_\_\_\_

RECHAZADO \_\_\_      ACEPTADO \_\_\_      ACEPTADO CON MODIFICACIONES \_\_\_

ACTA NO. \_\_\_\_\_

FECHA ENTREGA: \_\_\_\_\_

FIRMA CONSEJO DE FACULTAD \_\_\_\_\_

ACTA NO. \_\_\_\_\_

FECHA ENTREGA: \_\_\_\_\_