

Laboratorio Remoto para Sistemas Embebidos

Diseño de alto nivel



2021

Historial de revisiones

Nombre	Fecha	Descripción de los cambios	Versión
M. Sebastián Tobar	18/05/20	Emisión inicial	0
M. Sebastián Tobar	03/07/20	Se incorporan modificaciones realizadas durante el uso del laboratorio en las prácticas de RTOS para TD3.	1
M. Sebastián Tobar	02/11/20	Correcciones y adecuaciones para publicación.	2
M. Sebastián Tobar	11/03/21	Tipo de documento pasa de Memoria descriptiva a Diseño de alto nivel. Incorporación de controlador de dominio.	3

Contexto

Desde hace algunos años, las prácticas del tema Sistemas Operativos de Tiempo Real de la cátedra Técnicas Digitales III se realizan de manera presencial utilizando plataformas de hardware y software determinadas. En los últimos años la plataforma elegida ha sido la Computadora Industrial Abierta Argentina en su versión educativa (EDU-CIAA-NXP). Este hardware es suministrado por la cátedra durante las horas de práctica para que los alumnos desarrollen las actividades propuestas. En general, los alumnos solo tienen acceso a esta plataforma durante el horario de las prácticas, lo que limita sus posibilidades de repasar o profundizar el manejo de esta. Desafortunadamente, la situación actual de cuarentena obligatoria por la pandemia mundial de COVID-19 ha acentuado esta situación, imposibilitando de hecho el desarrollo de las actividades de laboratorio.

Propuesta

A partir de la problemática descripta, se planteó la posibilidad de desarrollar un laboratorio remoto de sistemas embebidos. El objetivo de este laboratorio es brindar a los estudiantes acceso a entornos de desarrollo y prueba de sistemas embebidos 24 horas al día, siete días a la semana de manera remota.

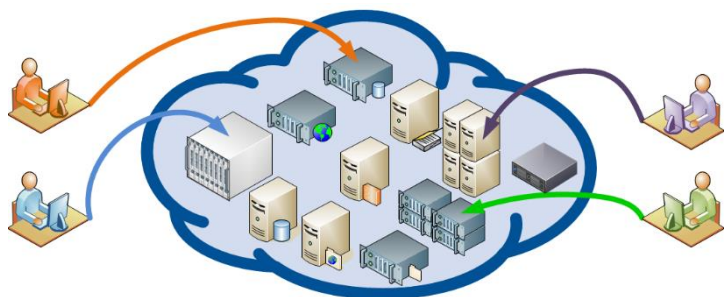
Descripción general

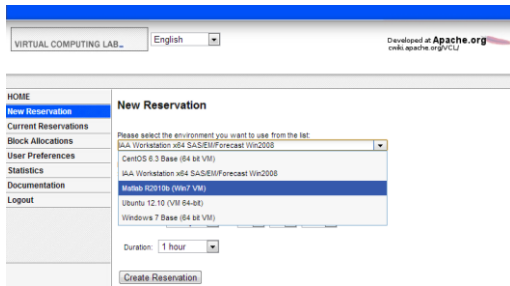
Laboratorio Remoto para Sistemas Embebidos es un conjunto de tecnologías libres orquestadas para suministrar acceso remoto a entornos de desarrollo y prueba de sistemas embebidos.

Tecnologías utilizadas

Apache VCL

Apache VCL (Virtual Computing Lab) es una plataforma de computación en la nube libre y de código abierto cuyo objetivo principal es brindar entornos informáticos personalizados a los usuarios. Estos entornos pueden ser algo tan sencillo como una máquina virtual ejecutando software de oficina o algo tan poderoso como un cluster corriendo complejas simulaciones.





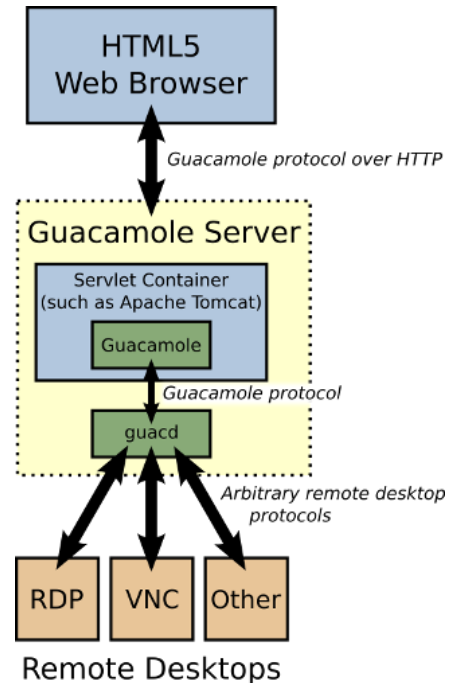
VCL soporta la entrega de distintos tipos de recursos incluyendo máquinas bare-metal, máquinas virtuales alojadas en distintos hipervisores y computadoras de laboratorio tradicionales.

La interfaz de usuario consiste en un autoservicio web. A través de este portal, los usuarios pueden seleccionar los ambientes y hacer reservas.

Apache Guacamole

Es una herramienta libre y de código abierto que permite acceso remoto a equipos usando únicamente un navegador web, sin la necesidad de ningún cliente. Guacamole no es una aplicación web monolítica, sino que está compuesta de varias partes. La aplicación web intenta ser sencilla, con la mayor parte del trabajo llevada a cabo por los componentes de menor nivel.

Los usuarios se conectan al servidor Guacamole con su navegador. El cliente Guacamole, escrito en JavaScript, es servido a los usuarios por Tomcat. Una vez cargado, el cliente se conecta nuevamente al servidor sobre HTTP usando el protocolo Guacamole. La aplicación web desplegada en el servidor recibe este protocolo y lo reenvía a guacd, el proxy Guacamole nativo. Este proxy interpreta el protocolo Guacamole conectándose a servidores de escritorio remoto en nombre del usuario. El protocolo Guacamole junto con guacd brindan abstracción del protocolo subyacente, ya que ni el cliente Guacamole ni la aplicación web deben conocer el protocolo de escritorio usado.



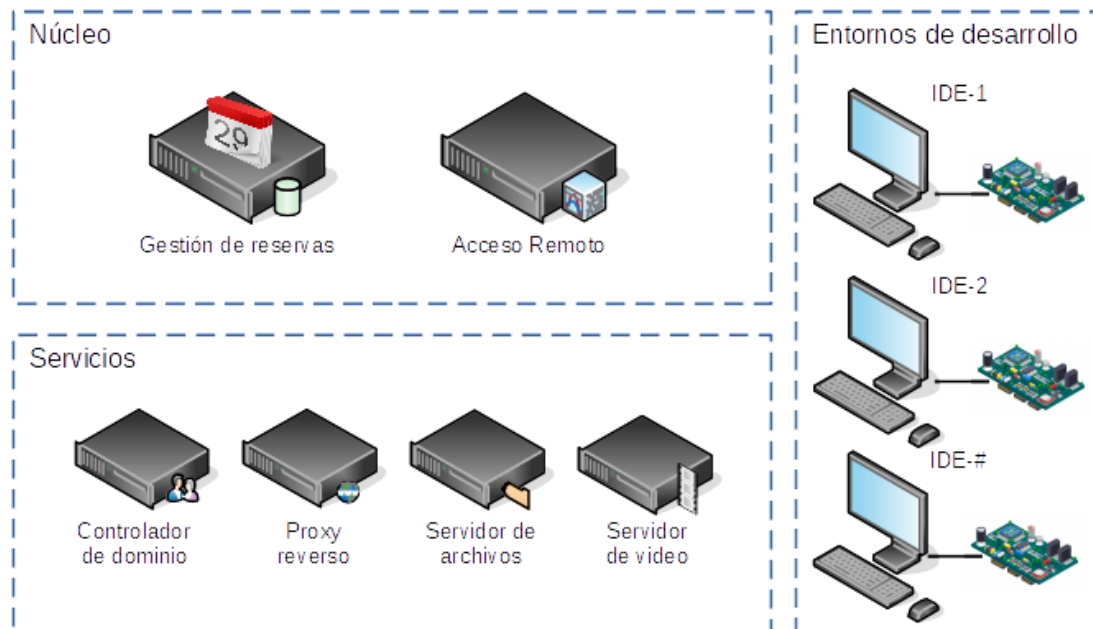
Linux containers (LXC)



LXC es un método de virtualización a nivel de sistema operativo para la ejecución de múltiples sistemas Linux de manera aislada (contenedores). El objetivo del proyecto es ofrecer un ambiente similar al que se obtendría con una máquina virtual, pero sin la sobrecarga que acarrea la ejecución de un kernel adicional y la simulación del hardware.

Arquitectura

El LRSE posee una arquitectura modular, integrando una serie de tecnologías abiertas que permiten alcanzar los requisitos de diseño del sistema. Los componentes fundamentales del laboratorio son la gestión de reservas y el acceso remoto. Dentro de los servicios se incorporan la gestión de archivos de los usuarios y el video en vivo. Finalmente se encuentran los entornos de desarrollo, que constituyen el recurso gestionado por el sistema.



Entornos de desarrollo

Los entornos de desarrollo (IDE) para la EDU-CIAA corren Ubuntu 20.04. Cada uno de estos entornos dispone de una plataforma embebida EDU-CIAA dedicada, la cual es utilizada por los estudiantes para probar las aplicaciones desarrolladas. Dentro de cada entorno, los estudiantes poseen un espacio privado para almacenamiento de archivos (/home). Este espacio de almacenamiento se encuentra sincronizado, a través de un servidor de archivos, entre los distintos entornos de desarrollo. De esta manera, independientemente del entorno al que acceda el estudiante, siempre dispondrá de sus archivos de trabajo.

Gestión de reservas

Los recursos demandados por los entornos de desarrollo son esencialmente computacionales. Por este motivo resulta conveniente la utilización de Apache VCL (Virtual Computing Lab) para la gestión de los entornos. Para el caso particular de este proyecto, Apache VCL gestiona los entornos de desarrollo de la EDU-CIAA.

Acceso remoto

El entorno de desarrollo implementado en el proyecto es accesible a través de emuladores de terminal. Con el objetivo de simplificar el acceso al laboratorio, se optó por la interfaz de acceso web Apache Guacamole, que brinda emuladores de terminal dentro del navegador, no requiriendo ningún software adicional.

Controlador de dominio

El controlador de dominio, basado en Zentyal, es el encargado de autenticar los usuarios a través de LDAP. Por otra parte, a través de los grupos asignados en el controlador de dominio a cada usuario, se gestionan los recursos a los que tienen acceso los usuarios en cada plataforma, en particular VCL y Guacamole.

Proxy reverso

Brinda servicios web en nombre de los servidores VCL y Guacamole. Almacena y gestiona los certificados Let's Encrypt para establecer conexiones seguras reconocidas por navegadores estándar.

Servidor de archivos

El servidor de archivos permite a los estudiantes acceder a su espacio de almacenamiento privado en el laboratorio a través de un cliente SFTP. Esta funcionalidad brinda al estudiante la flexibilidad de desarrollar código localmente y conectarse al laboratorio remoto solo para hacer pruebas en la placa EDU-CIAA.

Servidor de video

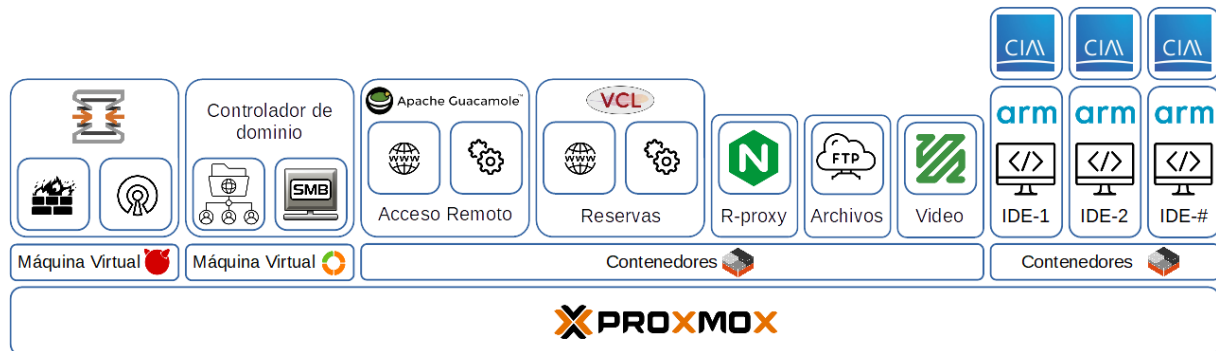
El servidor de video recibe el stream de una cámara IP que captura la parte superior de las placas EDU-CIAA, superpone una imagen con información relevante y envía el stream a la transmisión en vivo del canal de Youtube de la cátedra. Se utiliza la aplicación de código abierto ffmpeg.

Gestión de los entornos

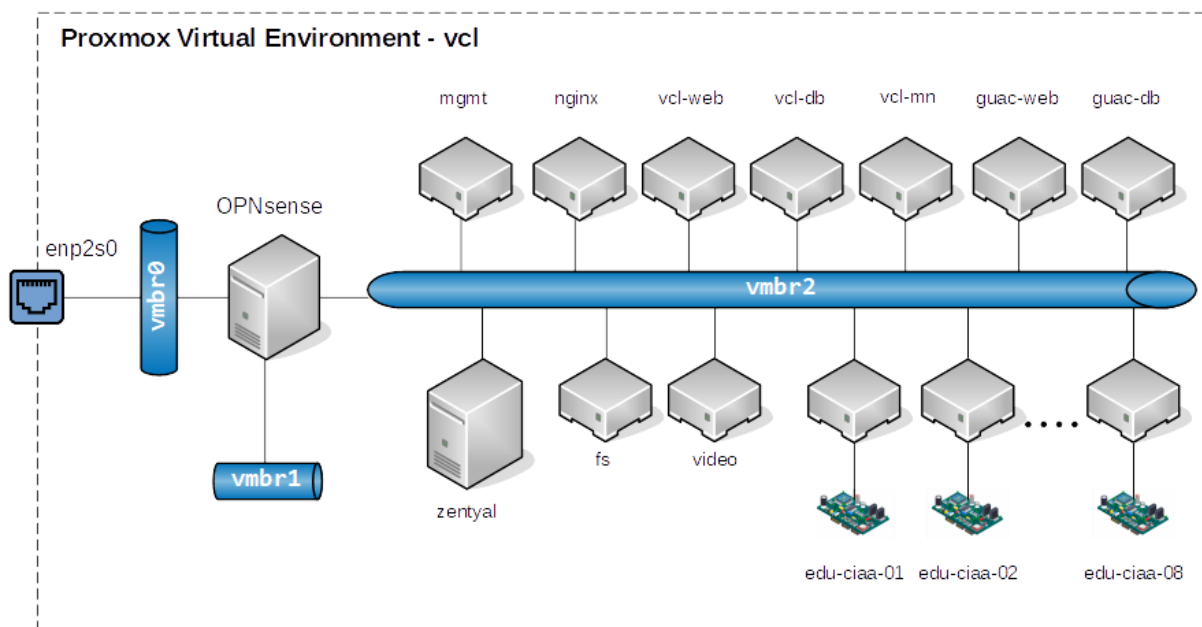
Apache VCL gestiona los entornos de desarrollo ejecutando el demonio vclcliend en cada máquina administrada. Al establecerse una reserva, el nodo de management de VCL (vcl-mn) se conecta a través de ssh a la máquina correspondiente con el usuario vclstaff y almacena la información de las reservas en archivos de texto plano dentro del home de vclstaff. El demonio vclcliend toma esta información y ejecuta una instancia de ssh-server con la opción AllowUsers <usuario>, permitiendo de esta manera que solo el usuario dueño de la reserva pueda iniciar sesión. Esta instancia de ssh-server espera conexiones en el puerto 22, por lo que el servidor ssh original de la máquina se configura para recibir conexiones en otro puerto para evitar conflictos.

Infraestructura del prototipo

El prototipo se desplegó íntegramente sobre la plataforma de virtualización Proxmox utilizando un único servidor y cuatro placas EDU-CIAA:



OPNSense y Zentyal se desplegaron utilizando máquinas virtuales, mientras que el resto de las aplicaciones utilizan contenedores LXC al igual que los entornos de desarrollo. El networking se resuelve a través de los recursos de Proxmox y utilizando OPNSense como firewall.



Hardware del prototipo

Servidor:

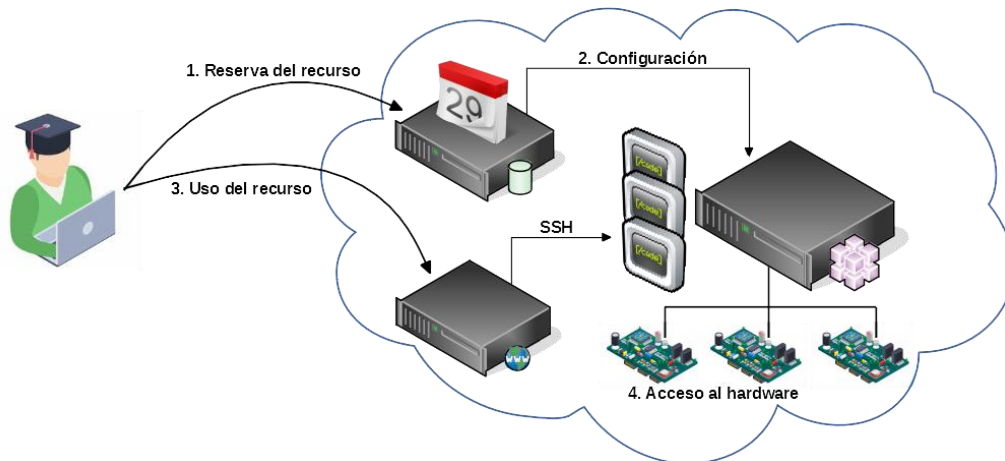
CPU(s): 8 x Intel(R) Core(TM) i7-4790S CPU @ 3.20GHz (1 Socket)
 RAM: 8192 MB
 HD: /dev/sda/ 931.5G

Sistemas embebidos:

EDU-CIAA-NXP

Caso de uso

Los estudiantes acceden al portal de autoservicio de VCL. Allí seleccionan el día, hora, tiempo y ambiente que necesitan utilizar. Si hay disponibilidad, el sistema agenda el pedido y genera credenciales para que el estudiante pueda iniciar sesión. Posteriormente, el estudiante accede a Guacamole el día y hora agendada, selecciona el recurso asignado y se identifica con las credenciales asignadas.



El estudiante tiene a su disposición hasta cinco terminales SSH simultáneas para conectarse al entorno de desarrollo, el cual tendrá vinculada una placa EDU-CIAA-NXP. En tiempo de ejecución, la interacción con la placa se podrá efectuar a través de los dos puertos seriales disponibles en la misma y a través de la interfaz de depuración.

Configuraciones

Apache Guacamole

Se utiliza una versión 1.3 de Guacamole compilada desde los archivos fuente. Corre sobre Java 11 y utiliza una base de datos Mysql donde se almacenan las conexiones. Tiene instaladas las extensiones para autenticación LDAP.

Dentro de Guacamole se configuran de manera manual las conexiones a los entornos de desarrollo EDU-CIAA. A través de los grupos se asignan permisos a las conexiones. Los grupos definidos en Guacamole también se definen de forma manual y deben tener el mismo nombre que en Zentyal.

Apache VCL

Se encuentra instalado sobre CentOS 7. La autenticación de usuarios se realiza a través de LDAP. Los recursos se asignan a través de grupos, los cuales deben coincidir de manera exacta con los existentes en Zentyal.

Servidor de archivos

Los homes de los usuarios residen en el servidor de archivos. De esta manera, independientemente del entorno al que se conecten, siempre accederán a sus archivos. En cada entorno de desarrollo los homes se montan con autofs.

El acceso SFTP se realiza a través del servidor de archivos. El acceso SSH está limitado a la transferencia de archivos, no siendo posible otorgar un shell a los usuarios.

```
/etc/passwd
```

```
Shell: /bin/false
```

Servidor de video

La aplicación ffmpeg recupera el flujo de video de una cámara IP que captura las placas EDU-CIAA y lo retransmite al canal en vivo de Youtube de la cátedra. ffmpeg se configura como servicio de tal manera de reiniciar el streaming si por algún motivo la aplicación se detiene. Existen dos servidores, uno principal y uno de respaldo.

EDU-CIAA-NXP

Esta placa utiliza el chip FT2232H como interfaz de programación y de comunicación serial. La configuración de este chip se almacena en una memoria EEPROM la cual no ha sido grabada en fábrica en las placas EDU-CIAA, motivo por el cual el chip trabaja con los parámetros por defecto. Es posible configurar, entre otros, los siguientes parámetros:

- Manufacturer
- Product Description
- Serial Number
- Tipo de controlador

Para poder identificar las placas y vincularlas de manera unívoca con los entornos de desarrollo virtuales, es necesaria la configuración del parámetro *Serial Number*. Esta programación se realiza con la herramienta FT Prog suministrada por el fabricante. **Se configura el parámetro *Serial Number* con el número de serie impreso en la propia placa EDU-CIAA.**

OpenOCD

OpenOCD utiliza el primer dispositivo USB FTDI que encuentra listado para establecer comunicación con la placa. Se debe configurar el parámetro `ftdi_serial` con el número de serie de la EDU-CIAA-NXP correspondiente a cada entorno para que OpenOCD utilice la placa prevista. El archivo de configuración se encuentra en:

```
~/firmware_v3/scripts/openocd/lpc4337_old.cfg
```

USB

Es necesario que los contenedores con el ambiente de desarrollo de la EDU-CIAA tengan acceso al bus USB para poder comunicarse con las placas. Por otra parte, si bien al conectar la placa se generan dos dispositivos seriales (`ttyUSB0` y `ttyUSB1`) solamente `ttyUSB1` es realmente un puerto serie, `ttyUSB0` es para programación y se elimina la primera vez que OpenOCD utiliza el puerto. Esto se debe a la falta de configuración de la EEPROM del chip FTDI.

El acceso al puerto serie `ttyUSB1` desde los contenedores se logra mediante reglas para `udev`. Se identifican las placas mediante el número de serie y se crean enlaces simbólicos, de tal

manera que los puertos tengan siempre el mismo nombre. Esta configuración se encuentra en el archivo:

```
/usr/lib/udev/rules.d/96-educiaa.rules
```

Adicionalmente se monta el enlace simbólico en el contenedor:

```
lxc.mount.entry: /dev/ttyUSB### dev/ttyUSB1 none bind,optional,create=file
```

En el archivo:

```
/etc/pve/lxc/###.conf
```

OpenOCD accede al puerto de programación a través de los archivos de dispositivo localizados en:

```
/dev/bus/usb/<bus number>/<device number>
```

Para brindarle acceso a las placas se monta el directorio completo del bus:

```
lxc.mount.entry: /dev/bus/usb/### dev/bus/usb/### none  
bind,optional,create=dir
```

En el archivo:

```
/etc/pve/lxc/###.conf
```

Por defecto OpenOCD utiliza el primer dispositivo FTDI que lista, motivo por el cual se debe ajustar el parámetro `ftdi_serial` en el archivo de configuración:

```
~/firmware_v3/scripts/openocd/lpc4337_old.cfg
```

Contenedores IDE EDU-CIAA

```
SO:          Ubuntu 20.04 LTS (Focal Fossa)
RAM:         1024 MB
vCPU:        1
Disk:        20 GB
Privileged:  Sí
```


```
root@edu-ciaa-02:~# arm-none-eabi-gcc --version
arm-none-eabi-gcc (15:9-2019-q4-0ubuntu1) 9.2.1 20191025 (release) [ARM/arm-9-branch revision 277599]
```

Contenedores VCL

```
SO:          CentOS Linux release 7.7.1908 (Core)
RAM:         1024 MB
vCPU:        1
Disk:        10 GB
Privileged:  No
```

Contenedor fs

```
SO:          CentOS Linux release 7.7.1908 (Core)
```

	Diseño de alto nivel	Página 11 de 11
	Laboratorio Remoto para Sistemas Embebidos	

RAM: 1024 MB

vCPU: 1

Disk: 10 GB

Privileged: No

Máquina virtual Docker

SO: Ubuntu 20.04 LTS (Focal Fossa)

RAM: 3072 MB

vCPU: 2

Disk: 32 GB

Máquina virtual OPNSense

SO: FreeBSD (Appliance)

RAM: 1024 MB

vCPU: 2

Disk: 32 GB