



Universidad del
Rosario



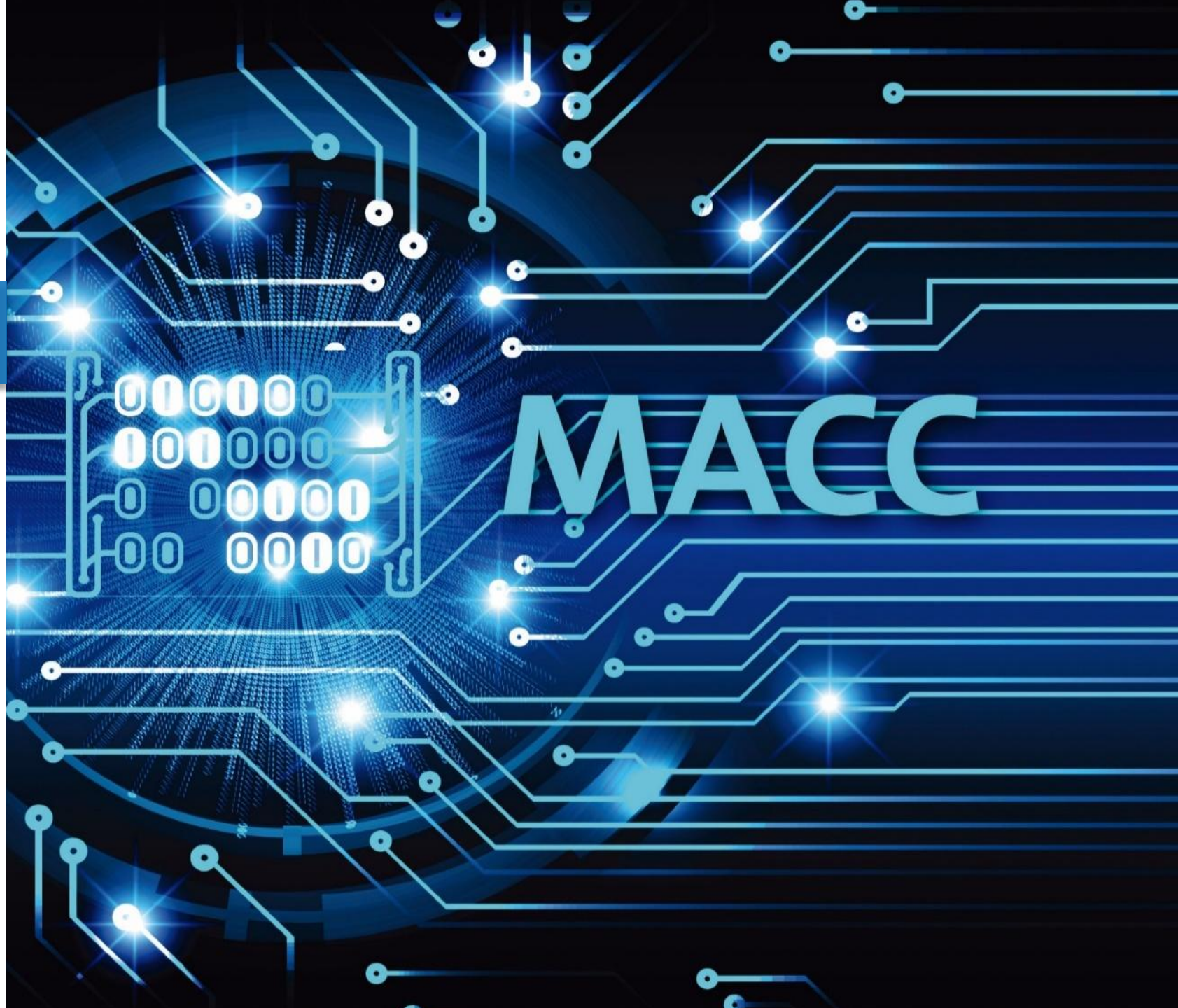
MACC
Matemáticas Aplicadas y
Ciencias de la Computación

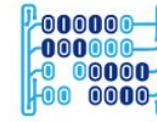
Hacking Mobile Platforms

Hacking Ético

Daniel Orlando Díaz López, PhD

Profesor principal
Departamento MACC
Universidad del Rosario
danielo.diaz@urosario.edu.co



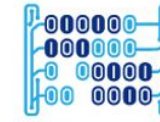


- DBI es inyectar código extraño en una aplicación existente para alterar su comportamiento
- Hacer DBI es diferente a hacer “explotación” o “Debugging”
- Por medio de DBI se puede:
 - Acceder a memoria
 - Sobrescribir funciones
 - Invocar funciones
 - Usar objetos
 - Interceptar funciones
- Usaremos la herramienta **FRIDA** para hacer DBI sobre aplicaciones Android
- Usaremos **Anbox** para emular un dispositivo móvil Android y evitar dañar un dispositivo real



Laboratorio

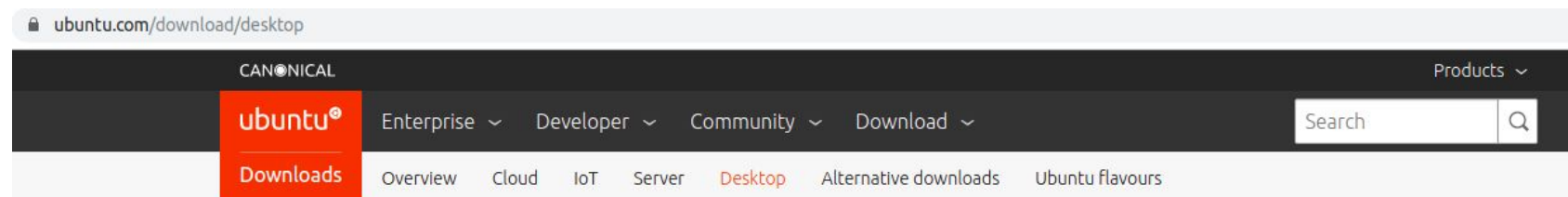
1. Instalar un emulador de dispositivos móviles Android: Anbox
2. Instalar la herramienta de hacking FRIDA
3. Validar la ejecución de comandos de FRIDA
4. Documentar con capturas de pantalla cada uno de los pasos realizados
5. Responder las preguntas planteadas que se encuentran al final



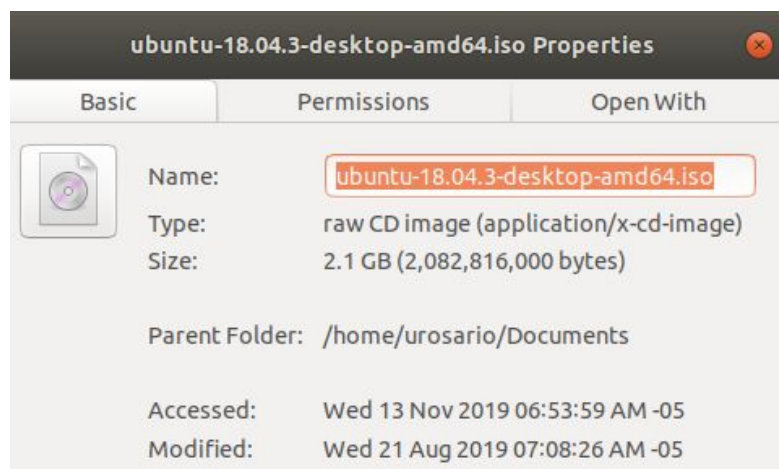
Iniciar Ubuntu 18.04 en modo live

Realizar diferentes tipos de ataques a una red inalámbrica

1. Descargar la última versión estable de Ubuntu en formato ISO



Download Ubuntu Desktop



Ubuntu 18.04.3 LTS

Download the latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years, until April 2023, of free security and maintenance updates, guaranteed.

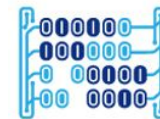
[Ubuntu 18.04 LTS release notes](#)

Recommended system requirements:

- ✓ 2 GHz dual core processor or better
- ✓ 4 GB system memory
- ✓ 25 GB of free hard drive space
- ✓ Either a DVD drive or a USB port for the installer media
- ✓ Internet access is helpful

Download

For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors, and past releases [see our alternative downloads](#).



Crear una USB booteable usando como archivo .ISO, la imagen de Ubuntu que se acaba de descargar. Necesitamos una memoria USB de al menos 4 Gb!

```
urosario@CNP77151: ~  
File Edit View Search Terminal Help  
(base) urosario@CNP77151:~$ echo "deb https://deb.etcher.io stable etcher" | sudo  
tee /etc/apt/sources.list.d/balena-etcher.list  
  
(base) urosario@CNP77151:~$ sudo apt-key adv --keyserver keyserver.ubuntu.com --re  
cv-keys 379CE192D401AB61  
  
(base) urosario@CNP77151:~$ sudo apt-get update  
  
(base) urosario@CNP77151:~$ sudo apt-get install balena-etcher-electron  
  
(base) urosario@CNP77151:~$ balena-etcher-electron
```

Seguir los pasos que se indican en el siguiente recurso

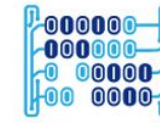
Desde un equipo Linux: : <https://github.com/balena-io/etcher>

Desde un equipo Windows: <https://tutorials.ubuntu.com/tutorial/tutorial-create-a-usb-stick-on-windows#0>

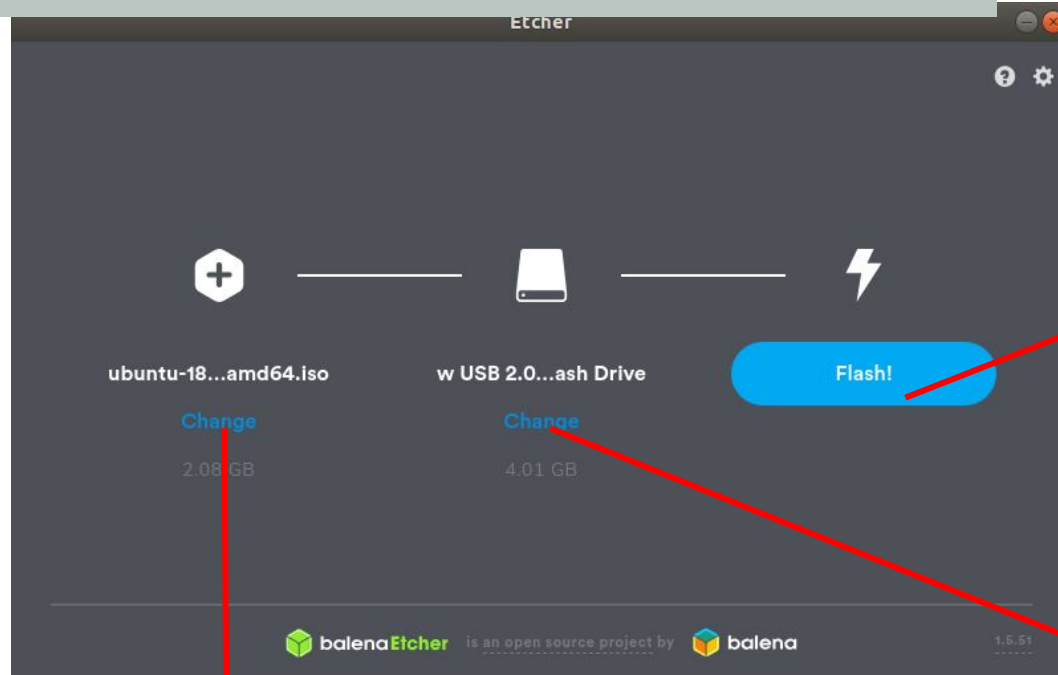
Dynamic Binary Instrumentation (DBI)



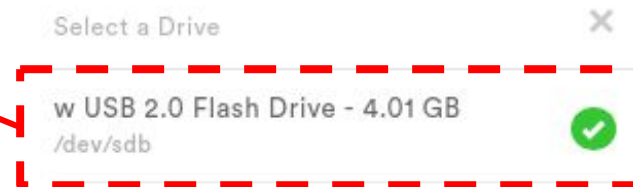
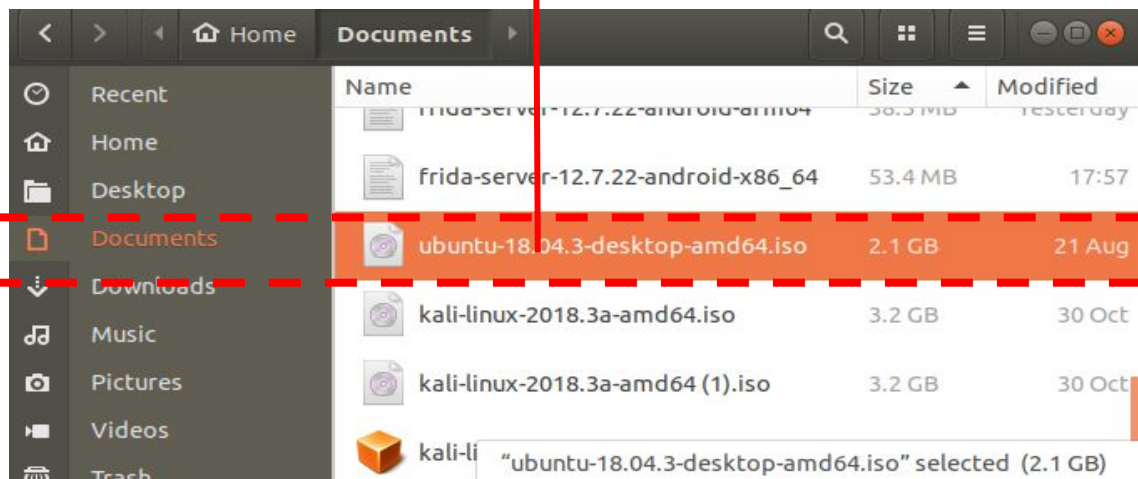
Universidad del
Rosario

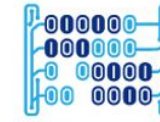


MACC
Matemáticas Aplicadas y
Ciencias de la Computación

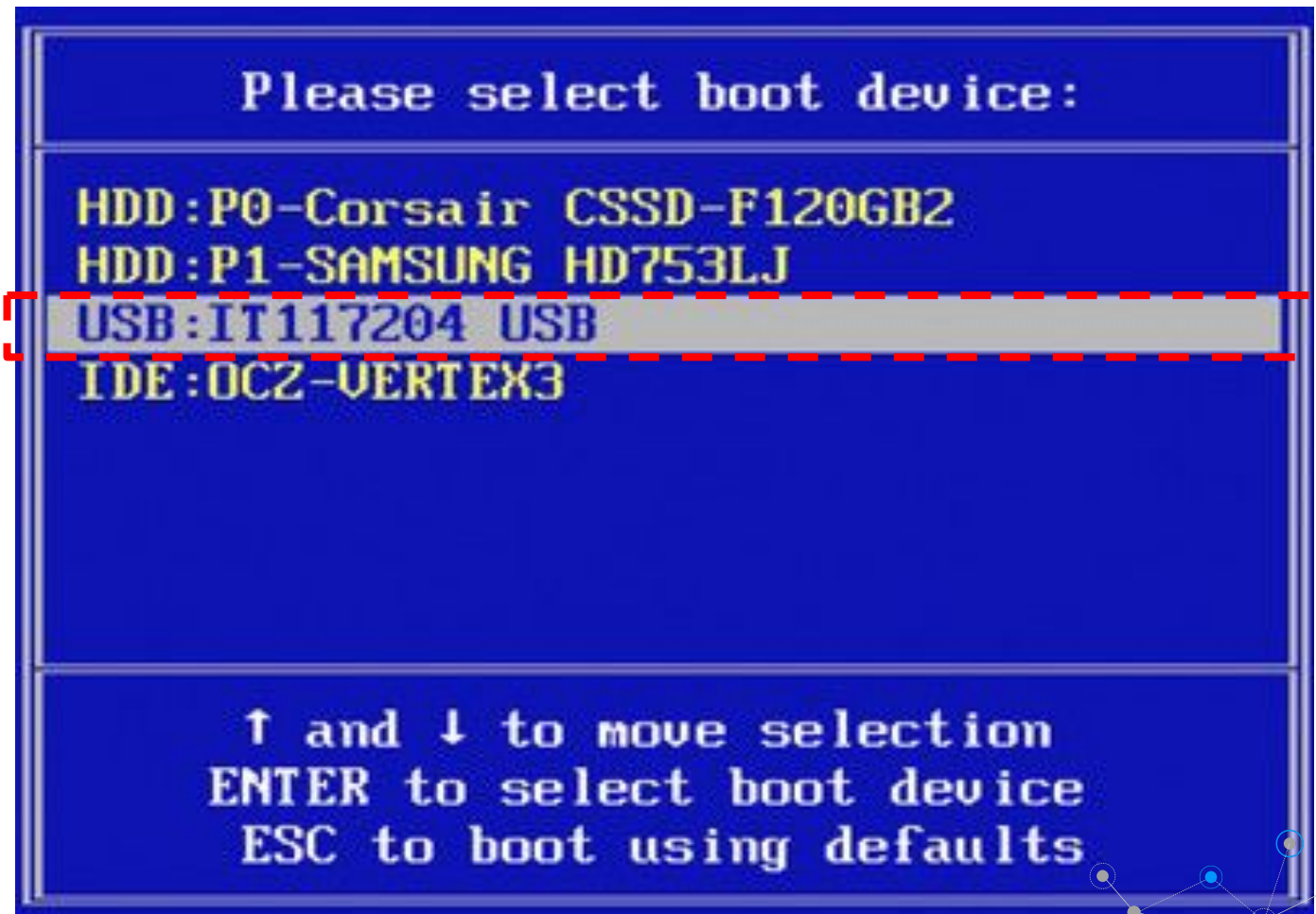


Este botón nos permite iniciar la creación de la memoria USB Bootable!

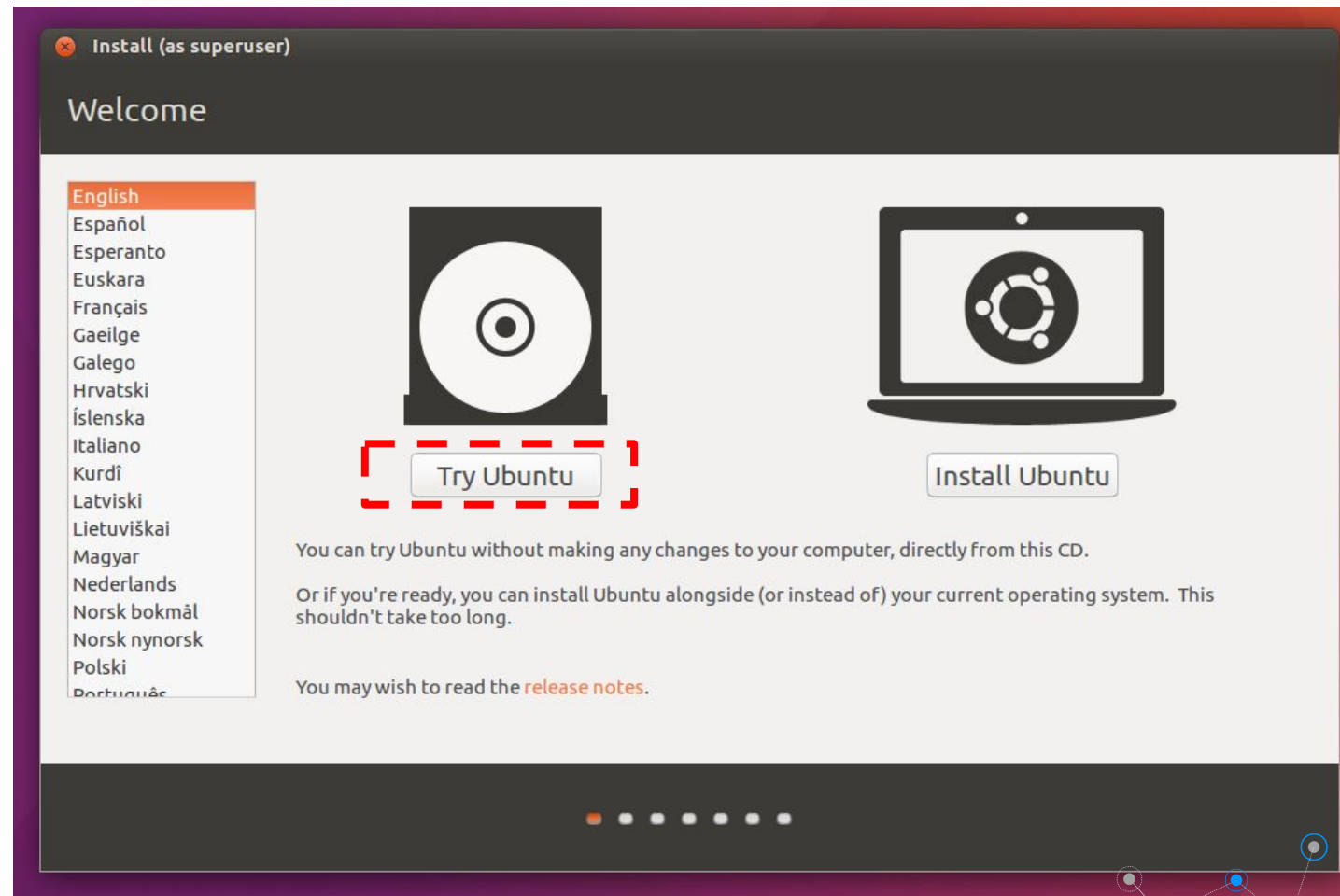




Reiniciar su computador (la memoria con Ubuntu debe estar puesta) y cuando el PC esté encendiendo presionar la tecla **F12** (dependiendo del modelo de su computador puede ser otra otra, sino funciona con F12 intentar con F2, F8, F10) para entrar la “Selección del dispositivo de arranque” y seleccionar la memoria USB en la que está instalado Ubuntu.

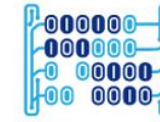


Iniciar Ubuntu en modo **LIVE CD** -> ¡Esto es como si tuvieramos Ubuntu instalado en el computador, pero tan pronto reiniciemos volveremos a tener nuestra instalación anterior y perderemos cualquier archivo guardado en Ubuntu!



Quien desee realizar el laboratorio en los equipos de la universidad puede tener dificultades para instalar ETCHER (la herramienta que usamos para crear la USB booteable), por ello pueden venir a mi oficina y con gusto les puedo prestar alguna de las USB booteables que ya tengo preparadas con Ubuntu 18.04.



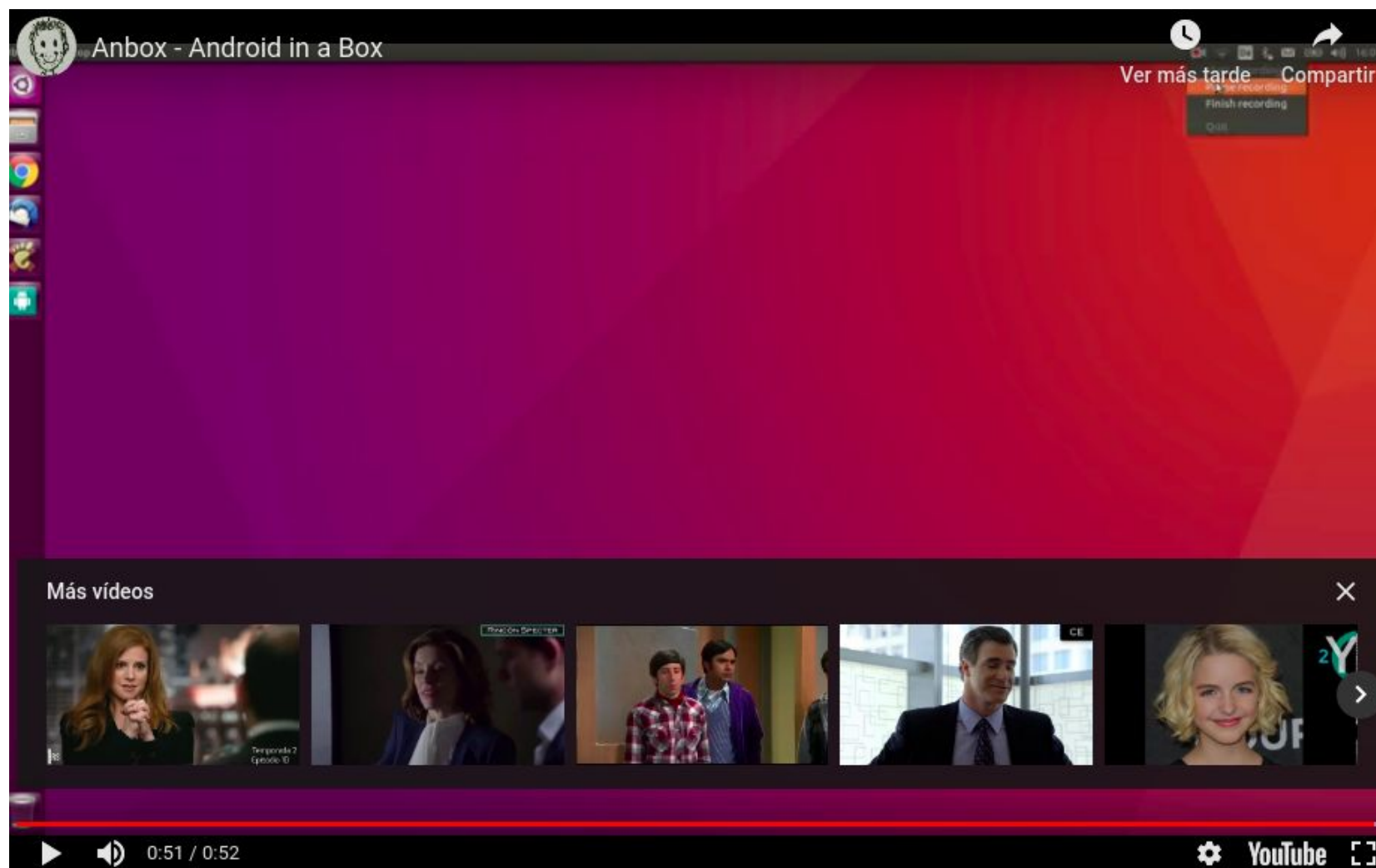


Anbox Emulator

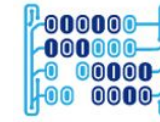
ANBOX es la herramienta que utilizaremos para emular un dispositivo Android a atacar, así evitamos dañar un dispositivo real



- Introducción a Anbox



<https://youtu.be/MbmiHnasGWg>



- Instalar módulos necesarios por el kernel: https://docs.anbox.io/userguide/install_kernel_modules.html

```
$ sudo add-apt-repository ppa:morphis/anbox-support  
$ sudo apt update  
$ sudo apt install linux-headers-generic anbox-modules-dkms
```

- Iniciar los módulos del kernel requeridos para la virtualización

```
$ sudo modprobe ashmem_linux  
$ sudo modprobe binder_linux
```

- Validar que los módulos del kernel hayan quedado cargados

```
(base) urosario@CNP77151:~$ ls -1 /dev/{ashmem,binder}  
/dev/ashmem  
/dev/binder  
(base) urosario@CNP77151:~$
```

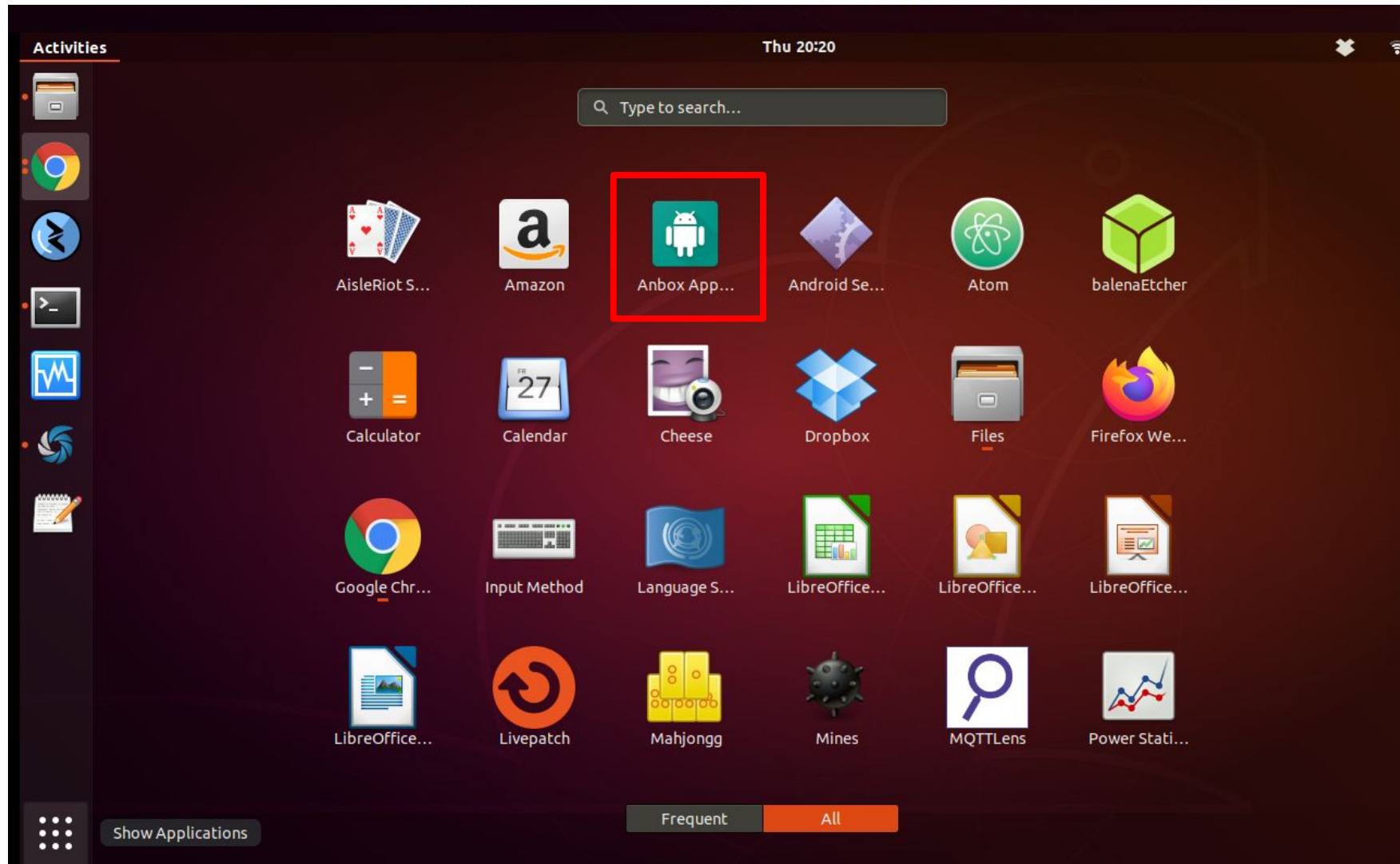
- Con los prerequisites cumplidos en el anterior slide, ya podemos proceder a instalar Anbox. <https://docs.anbox.io/userguide/install.html>

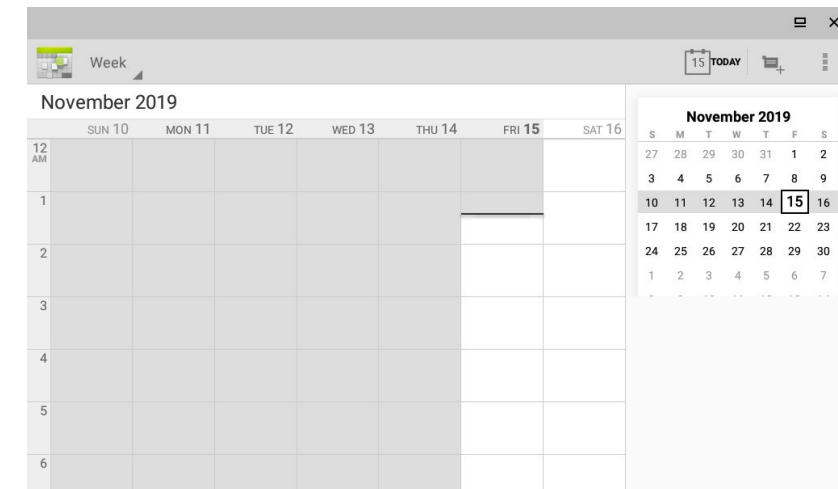
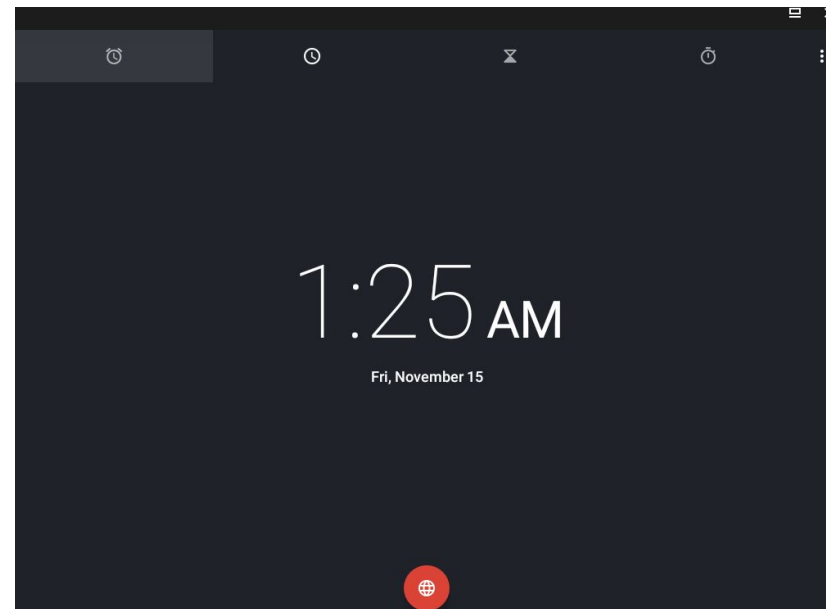
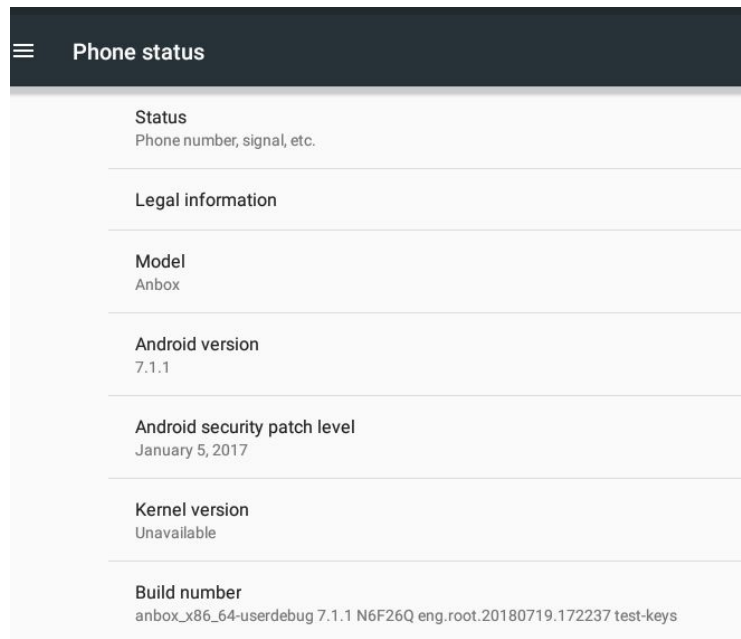
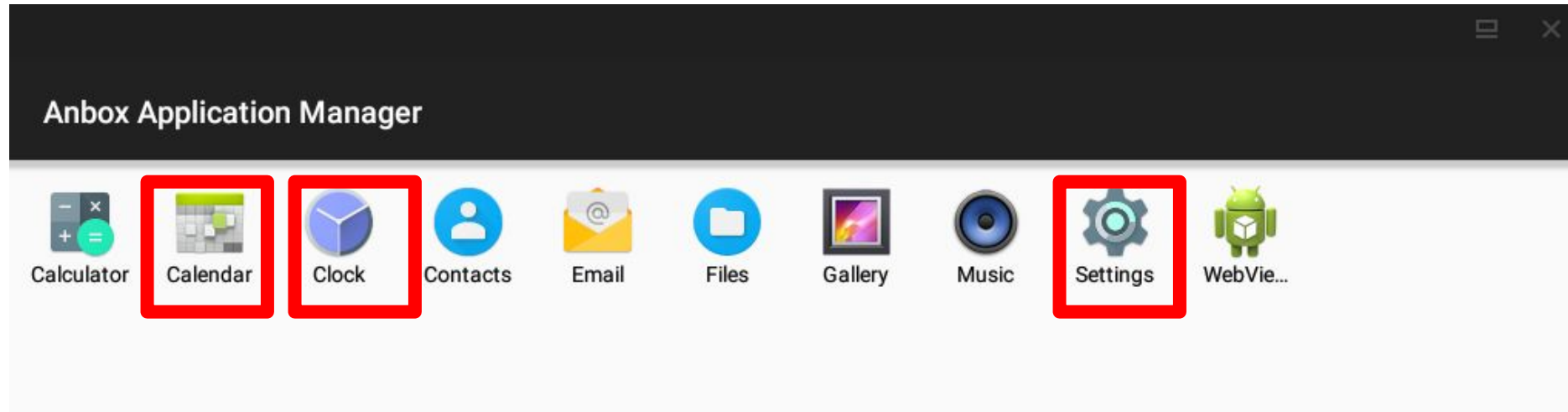
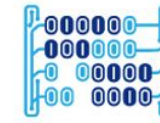
```
$ snap install --devmode --beta anbox
```

- Validamos que la instalación haya sido correcta

```
(base) urosario@CNP77151:~$ snap info anbox
name:      anbox
summary:   Android in a Box
publisher: morphis
contact:   https://anbox.io
license:   unset
description: |
  Runtime for Android applications which runs a full Android system
  in a container using Linux namespaces (user, ipc, net, mount) to
  separate the Android system fully from the host.
```

- Ahora ya podemos ejecutar Anbox





- Ahora debemos instalar ADB (Android Debug Bridge), el cual nos permitirá comunicarnos desde la máquina física con el dispositivo emulado

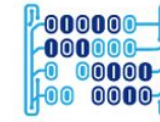
```
(base) urosario@CNP77151:~$ sudo apt install adb
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  android-libadb android-libbase android-libboringssl android-libcrypto-utils
  android-libcutils android-liblog android-sdk-platform-tools-common
```

- Para ver los dispositivos móviles “reales” o emulados mediante ANBOX que están conectados al computador físico lo hacemos por medio del siguiente comando:

```
(base) urosario@CNP77151:~$ adb devices
List of devices attached
* daemon not running; starting now at tcp:5037
* daemon started successfully
emulator-5558    device
```

Este es el dispositivo
emulado!!

<https://developer.android.com/studio/command-line/adb>



FRIDA

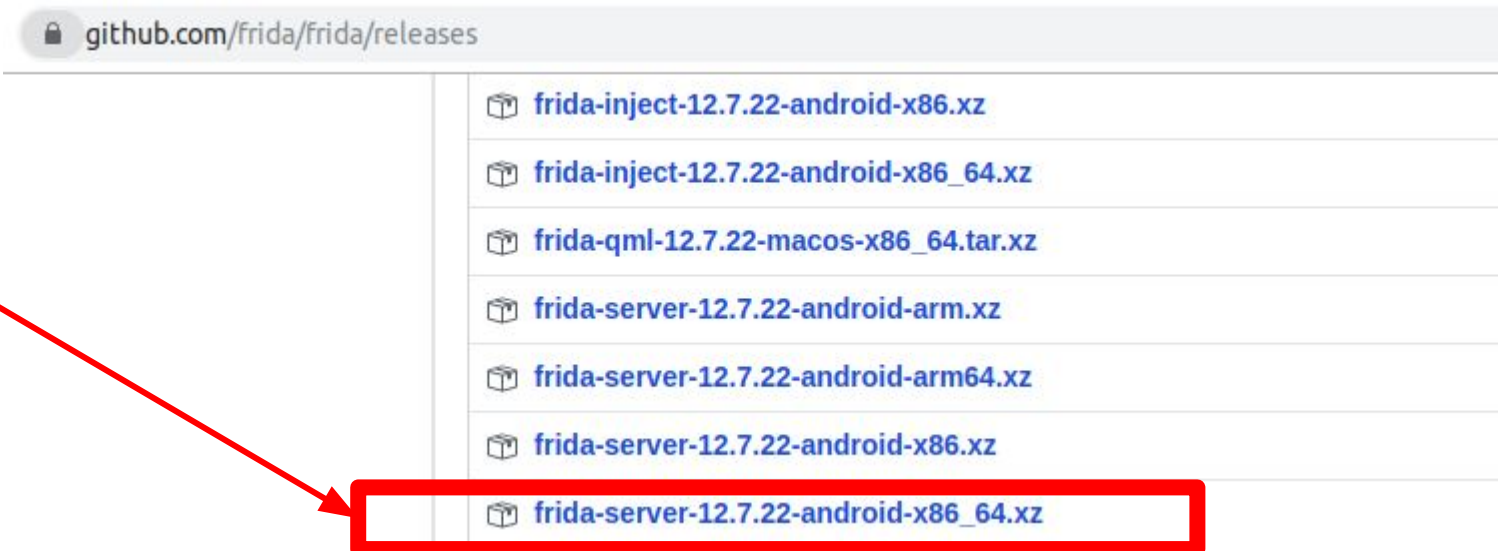
FRIDA es la herramienta que utilizaremos para inyectar código a una aplicación
móvil
vulnerable

```
(base) urosario@CNP77151:~$ npm install frida
loadDep:strip-json-commen |
```

- Descargar FRIDA server - <https://github.com/frida/frida/releases>

Escoger la versión adecuada según la arquitectura del dispositivo móvil (X86, X86_64, arm), en nuestro caso será la x86_64

Esta es la ruta de destino de la transferencia en el dispositivo emulado



- Instalar FRIDA server en el dispositivo emulado utilizando la herramienta “adb” previamente instalada y el argumento “push”

```
(base) urosario@CNP77151:~/Documents$ adb push frida-server-12.7.22-android-x86_64 /data/local/tmp/frida-server
frida-server-12.7.22-android-x86_64: 1.... 97.4 MB/s (53368072 bytes in 0.523s)
```


- Iniciar una consola en el dispositivo emulado usando el comando “adb shell”
- Después de iniciar la consola ejecutar el comando “su” para cambiar a superusuario

```
(base) urosario@CNP77151:~/Documents$ adb shell
generic_x86:/ $ su
generic_x86:/ # ls
acct          init.environ.rc      sdcard
bugreports    init.goldfish.rc     seapp_contexts
cache         init.ranchu-encrypt.rc selinux_version
charger       init.ranchu-noencrypt.rc sepolicy
config        init.ranchu.rc        service_contexts
d             init.rc              storage
data          init.usb.configfs.rc sys
default.prop  init.usb.rc          system
dev           init.zygote32.rc     ueventd.goldfish.rc
etc           mnt                  ueventd.ranchu.rc
file_contexts.bin oem                  ueventd.rc
fstab.goldfish proc                 var
fstab.ranchu-encrypt property_contexts   vendor
fstab.ranchu-noencrypt root
init          sbin
generic_x86:/ #
```

→ Cambio a superusuario!

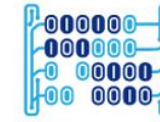
- En este punto ya estamos dentro del dispositivo móvil y acabamos de rootear el dispositivo (acceder como usuario root)

- Revisemos que el archivo de FRIDA server que transferimos hace 2 slides esta en la ruta hacia donde lo transferimos

```
x86_64:/ # cd /data/local/tmp/  
x86_64:/data/local/tmp # ls  
frida-server re.frida.server
```

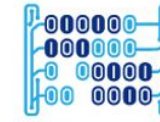
- Ahora asignarle permisos de ejecución al archivo y ejecutarlo

```
x86_64:/data/local/tmp # chmod 755 frida-server  
x86_64:/data/local/tmp # ./frida-server  
[
```



- En este punto ya podemos ejecutar diferentes comandos desde la consola del equipos físico para acceder a los datos del equipos emulado, por ejemplo: en una consola del equipo físico (no del dispositivo móvil) ejecutar el siguiente comando:
frida-ps -U

```
(base) urosario@CNP77151:~/Documents$ frida-ps -U
PID  Name
-----
1322  adbd
2823  android.process.acore
1406  audioserver
1407  camerasever
1856  com.android.inputmethod.latin
2021  com.android.phone
2438  com.android.printspooler
3024  com.android.providers.calendar
1904  com.android.systemui
```



Preguntas

Android Studio

- ¿Que significa “Rootear” un equipo móvil?
- ¿Que es un APK?
- ¿Cuántos mercados de android apps existen? ¿Cual es más seguro?
- ¿Que es una arquitectura ARM? ¿Que diferencias hay entre un procesador ARM y uno Intel?
- ¿Cual es el procesador de su teléfono?

Emulator

- ¿Qué es una instantánea?
- ¿Qué es Quick Boot (contrario a Inicio en frío)?
- ¿Qué limitaciones tiene un emulador?

Frida

- Investigue cual el resultado del comando Frida-ps -U
- ¿Cuales son las funcionalidades de Frida?
- ¿Que otros comandos están disponibles en Frida?

Links de referencia:

<https://www.androidauthority.com/arm-vs-x86-key-differences-explained-568718/>
<https://www.frida.re/docs/modes/>
<https://www.frida.re/docs/hacking/>
<https://www.frida.re/docs/frida-cli/>



Universidad del
Rosario



MACC



HINNT

¡Gracias!