

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica



---

**Bitácora Proyecto 1**

---

Taller de Sistemas Embebidos

Integrantes:

Jorge Andrés Brenes Alfaro

Profesor:

Ing. Johan Carvajal Godínez

20 de octubre de 2021

# Interacción 1

**Fecha:** 23/09/2021

**Tipo de sesión:** Individual

**Actividad:** Se lleva a cabo la instalación de la máquina virtual debido a que se realiza un cambio de computadora ya que la anterior no soportaba la carga y no daba un buen rendimiento

Se hace uso de la aplicación VirtualBox donde se instala el sistema operativo Ubuntu versión 20.04.3 LTS. Se le asigna a la computadora virtual 8GB de RAM, 4 núcleos para el procesador y 255GB de SSD. Además, se lleva la instalación del yocto project y opencv.

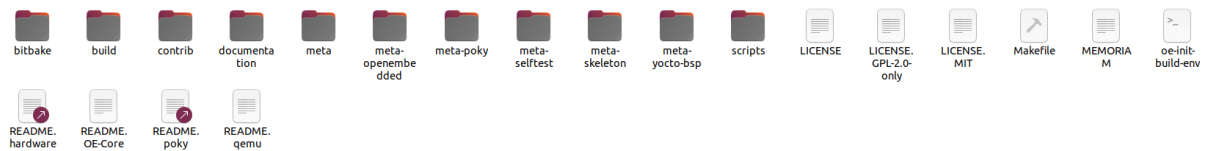


Figura 1: Instalación de Yocto Project.

```
jorge@Jorge-Brenes:~$ python3 -c "import cv2; print(cv2.__version__)"
4.2.0
jorge@Jorge-Brenes:~$
```

Figura 2: Instalación de OpenCV

**Problemas:** La instalación se llevó a cabo sin problemas y la máquina quedó funcionando.

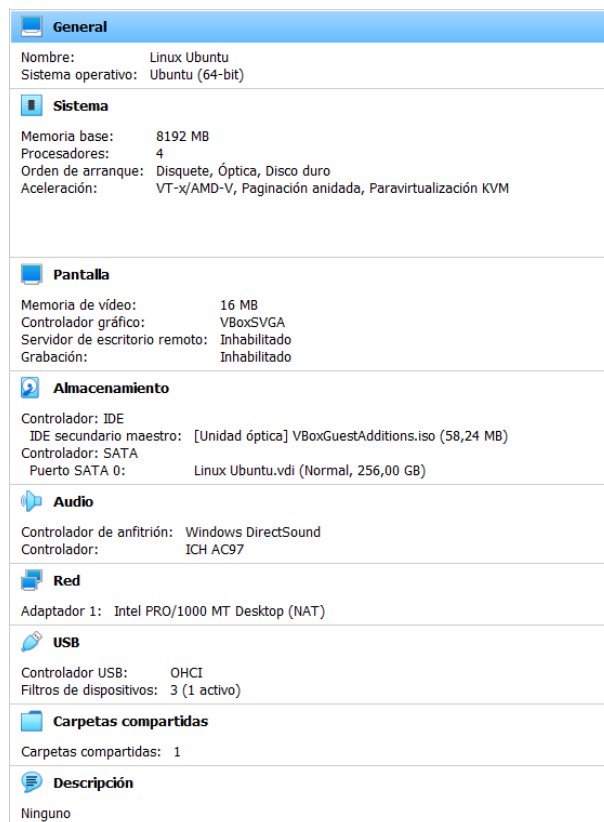


Figura 3: Instalación de la máquina virtual.

## Interacción 2

**Fecha:**

**Tipo de sesión:** Individual

**Actividad:** En esta interacción lo que se realiza principalmente es en una investigación bibliográfica con el fin de aclarar varios puntos acerca del proyecto.

Primeramente, acerca del reconocimiento de emociones en donde se busca algún código predesarrollado en donde ayudaron a tener conocimiento sobre estos para en futuro elegir el que se va a utilizar, para ello se consultó a [2] y [3]. Y se consulta en [6] si ya hay aplicaciones de ello para utilizar.

También se busca información acerca de tensorflow lite, para lo cual se utiliza la referencia [4]. También se averigua sobre como convertir un modelo en tflite, para lo ello se sigue las siguientes referencias [5], [7] y [8].

En cuanto a OpenCV y Yocto Project, ya se conoce su flujo de trabajo por lo que no se indaga mucho en ellos.

**Problemas:** No surgen problemas debido a la naturaleza de la sesión.

## Interacción 3

**Fecha:** 26/09/2021

**Tipo de sesión:** Individual

**Actividad:** Se lleva acabo una reunión grupal para saber que información se ha recopilado, intercambio de ideas, que se entiende hasta el momento del proyecto y determinar una estrategia a seguir para su solución. Se discute sobre la propuesta de diseño y del código de reconocimiento de emociones.

**Problemas:** No surgieron problemas en la sesión.

## Interacción 4

**Fecha:** 30/09/2021

**Tipo de sesión:** Individual

**Actividad:** Se trabaja en la propuesta de diseño donde se realiza la justificación del proyecto y se realiza una separación de apartados del proyecto entre los miembros del grupo.

**Problemas:** No surgen problemas en la sesión.

## Interacción 5

**Fecha:** 01/10/2021

**Tipo de sesión:** Individual

**Actividad:** De igual forma que en la interacción anterior se trabaja en la propuesta de diseño, donde en este caso se trabaja con la sección del análisis del problema y la gestión de requerimientos. Además, se acuerda un código ganador para trabajarlo y hacerle las modificaciones necesarias de tal forma que se simplifique su uso y se pueda adecuar de una mejor forma para los propósitos del proyecto. El código recopilado se encuentra en [1].

**Problemas:** No surgen problemas en la sesión.

## Interacción 6

**Fecha:** 02/10/2021

**Tipo de sesión:** Individual y Grupal

**Actividad:** Se realiza una reunión grupal en donde se verifica el funcionamiento del código encontrado, para llevar a cabo ello se requiere instalar las librerías numpy, opencv y tensorflow, requeridas para correr el programa de reconocimiento de emociones, estas se pueden observar en la Fig. 4. Sin embargo, este se ejecuta en la ventana de comandos de anaconda, debido a que la máquina virtual no reconoce la cámara integrada de la computadora.

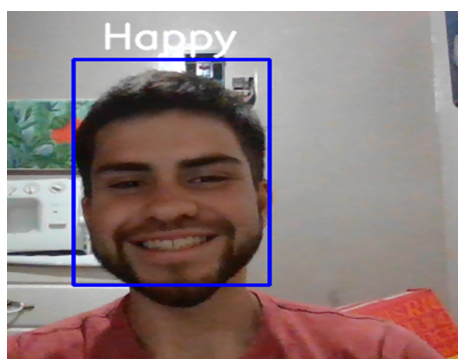
```
jorge@Jorge-Brenes:~$ python3 -c "import cv2; print(cv2.__version__)"
4.2.0
jorge@Jorge-Brenes:~$
```

(a) Instalación de OpenCV

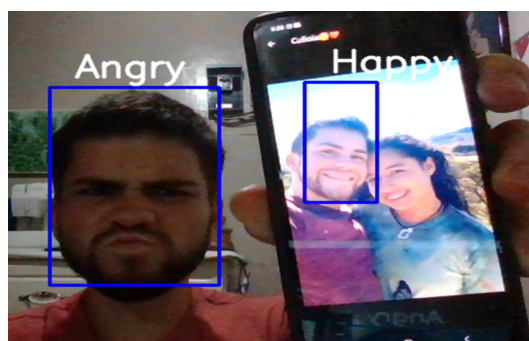
```
jorge@Jorge-Brenes:~$ pip install tensorflow
Collecting tensorflow
  Downloading tensorflow-2.6.0-cp38-cp38-manylinux2010_x86_64.whl (458.4 MB)
    | 23.6 MB 1.8 MB/s eta 0:04:05
```

(b) Instalación de tensorflow

Figura 4: Instalaciones de las librerías numpy, OpenCV y tensorflow.



(a) a



(b) b

Figura 5: Verificación del código de reconocimiento de emociones.

**Problemas:** La máquina virtual no reconoce la cámara integrada de la computadora, para solucionar esto se agrega a los dispositivos, no obstante, sigue sin reconocerla. Igual se presenta el problema de que no se encuentra instalado el comando pip, el cual se soluciona instalándolo, como se muestra en la Fig. 6.

```
jorge@Jorge-Brenes:~$ sudo apt install python3-pip
[sudo] contraseña para jorge:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  libbexpat1-dev libpython3-dev libpython3.8-dev python-pip-whl python3-dev
  python3-setuptools python3-wheel python3.8-dev
Paquetes sugeridos:
  python-setuptools-doc
Se instalarán los siguientes paquetes NUEVOS:
  libbexpat1-dev libpython3-dev libpython3.8-dev python-pip-whl python3-dev
  python3-pip python3-setuptools python3-wheel python3.8-dev
0 actualizados, 9 nuevos se instalarán, 0 para eliminar y 6 no actualizados.
Se necesita descargar 6 967 kB de archivos.
Se utilizarán 26,4 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://cr.archive.ubuntu.com/ubuntu focal/main amd64 libbexpat1-dev amd64 2.2.9-1build1 [116 kB]
Ign:2 http://cr.archive.ubuntu.com/ubuntu focal-updates/main amd64 libpython3.8-dev amd64 3.8.10-0ubuntu1-20
.04
Des:3 http://cr.archive.ubuntu.com/ubuntu focal/main amd64 libpython3-dev amd64 3.8.2-0ubuntu2 [7 236 B]
Des:4 http://cr.archive.ubuntu.com/ubuntu focal-updates/universe amd64 python-pip-whl all 20.0.2-5ubuntu1.6
```

Figura 6: Instalación del comando pip.

## Interacción 7

**Fecha:** 06/10/21

**Tipo de sesión:** Individual

**Actividad:** Se procede a cocinar la imagen de sato en la máquina virtual, en donde esto se lleva a cabo por medio de los siguientes comandos:

- source oe-init-env
- time bitbake core-image-sato

El cocinar la imagen llevo alrededor de 3 horas.

```
jorge@Jorge-Brenes:~/Documentos/poky/build$ time bitbake core-image-sato
Loading cache: 100% |
Loaded 0 entries from dependency cache.
Parsing recipes: 100% |#####| ETA: --:--:--
Parsing of 814 .bb files complete (0 cached, 814 parsed). 1438 targets, 41 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION      = "1.50.0"
BUILD_SYS       = "x86_64-linux"
NATIVELSBSTRING = "ubuntu-20.04"
TARGET_SYS      = "x86_64-poky-linux"
MACHINE         = "qemuarm64"
DISTRO          = "poky"
DISTRO_VERSION  = "3.3.3"
TUNE_FEATURES   = "m64 core2"
TARGET_FPU      = ""
meta
meta-poky
meta-yocto-bsp   = "my-hardknot:b98a57715986cd89b23bd62d8ad35db57de6334"

NOTE: Fetching uninitative binary shln http://downloads.yoctoproject.org/releases/uninitative/3.2/x86_64-native-sdk-libc.tar.xz;sha256sum=3ee8c7d55e2d4c7ae3887cddb97219f97b94efddfee2e24923c0cb0e8ce84c6 (will
check PREMIRRORS first)
Initialising tasks: 100% |#####|
State summary: Wanted 2725 Local 0 Network 0 Missed 2725 Current 0 (0% match, 0% complete)
NOTE: Executing Tasks
WARNING: expat-2.2.10-r0 do_fetch: Failed to fetch URL https://downloads.sourceforge.net/expat/expat-2.2.10.tar.bz2, attempting MIRRORS if available
NOTE: Tasks Summary: Attempted 6851 tasks of which 0 didn't need to be rerun and all succeeded.

Summary: There was 1 WARNING message shown.

real    186m9.210s
user    0m28.527s
sys     0m4.560s
```

Figura 7: Imagen Sato cocinada.

**Problemas:** no surgen problemas la imagen se cocino correctamente y se corre para verificar su funcionamiento, el cual procede con éxito.

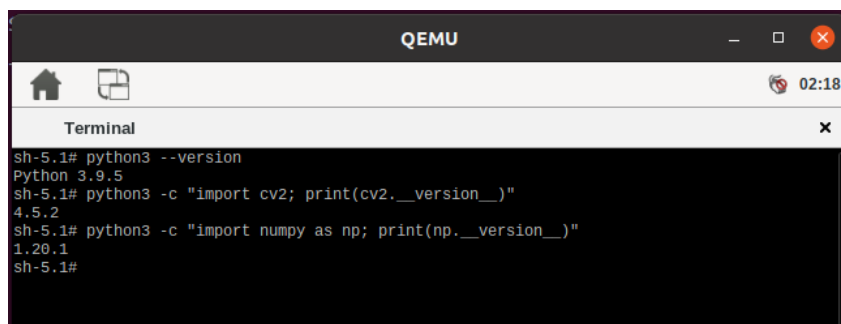
## Interacción 8

**Fecha:** 08/10/2021

**Tipo de sesión:** Grupal

**Actividad:** Junto a la compañera Kimberly se procede hacer la instalación de diversas capas en el yocto Project, donde se agregan las capas de meta-openembedded y meta-oe, las cuales son necesarias para utilizar Python, además de obtener numpy y opencv con dichas capas.

Seguidamente se procede a cocinar la imagen con las nuevas capas añadidas y posteriormente de cocinar la Sato se procede a verificar mediante el Qemu que se tiene las librerías instaladas.



```
QEMU
Terminal
sh-5.1# python3 --version
Python 3.9.5
sh-5.1# python3 -c "import cv2; print(cv2.__version__)"
4.5.2
sh-5.1# python3 -c "import numpy as np; print(np.__version__)"
1.20.1
sh-5.1#
```

Figura 8: Imagen Sato con Python y sus librerías.

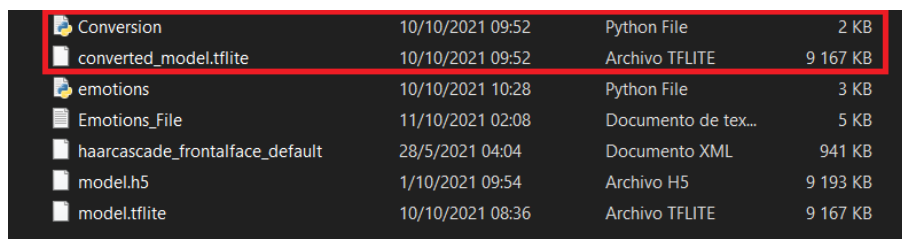
**Problemas:** No surgen problemas, la imagen Sato se ejecuta correctamente con las librerías integradas

## Interacción 9

**Fecha:** 10/10/21

**Tipo de sesión:** Individual

**Actividad:** Se trabaja con el código elegido de reconocimiento de emociones para convertirlo en un modelo de tensorflow lite, esto se puede observar en la Fig. 9. Sin embargo, no se puede imprimir las dimensiones como indica la página de tensorflow lite, por lo que este modelo no se puede integrar al código para realizar la predicción de la emoción.



Conversion	10/10/2021 09:52	Python File	2 KB
converted_model.tflite	10/10/2021 09:52	Archivo TFLITE	9 167 KB
emotions	10/10/2021 10:28	Python File	3 KB
Emotions_File	11/10/2021 02:08	Documento de tex...	5 KB
haarcascade_frontalface_default	28/5/2021 04:04	Documento XML	941 KB
model.h5	1/10/2021 09:54	Archivo H5	9 193 KB
model.tflite	10/10/2021 08:36	Archivo TFLITE	9 167 KB

Figura 9: Conversión modelo a .tflite.

**Problemas:** La conversión del modelo es exitosa, no obstante, no se puede verificar su funcionalidad debido a que no se ha podido integrar en el código.

## Interacción 10

**Fecha:** 11/10/2021

**Tipo de sesión:** Grupal

**Actividad:** Se trabaja con la compañera Kimberly en integrar el tensorflow al yocto project. Para ello, se debe incluir la capa meta-python, meta-tensorflow y en adición, se le agrega meta-raspberry. Estas carpetas se pueden ver agregadas en la Fig. 10.

```
jorge@Jorge-Brenes:~/Documentos/poky/build$ bitbake-layers add-layer ../meta-openembedded/meta-tensorflow
NOTE: Starting bitbake server...
jorge@Jorge-Brenes:~/Documentos/poky/build$ bitbake-layers add-layer ../meta-openembedded/meta-raspberrypi/
NOTE: Starting bitbake server...
jorge@Jorge-Brenes:~/Documentos/poky/build$ bitbake-layers show-layers
NOTE: Starting bitbake server...
layer                path                                     priority
=====
meta                 /home/jorge/Documentos/poky/meta        5
meta-poky            /home/jorge/Documentos/poky/meta-poky   5
meta-yocto-bsp       /home/jorge/Documentos/poky/meta-yocto-bsp 5
meta-oe              /home/jorge/Documentos/poky/meta-openembedded/meta-oe 6
meta-python          /home/jorge/Documentos/poky/meta-openembedded/meta-python 7
meta-tensorflow      /home/jorge/Documentos/poky/meta-openembedded/meta-tensorflow 10
meta-raspberrypi     /home/jorge/Documentos/poky/meta-openembedded/meta-raspberrypi 9
jorge@Jorge-Brenes:~/Documentos/poky/build$
```

Figura 10: Adición de capas meta-python, meta-tensorflow y meta-raspberry.

Además, se agrega el archivo de emotiondetect.py y el model.h5 y el haarcascade\_frontalface\_default, archivos necesarios para la ejecución del programa.

**Problemas:** Al cocinar la imagen, la capa meta-python presenta un error el cual se encuentra en la Fig. 11.

Para la solución de este problema, se solicita la ayuda del compañero Fabián, a quién le logró cocinar bien el meta-python. El compañero me comparte la capa introducida, la cual integro al poky. Dicha acción resuelve el error que se presentaba. De esta forma se logra cocinar, la capa meta-python y meta-tensorflow.



```
jorge@Jorge-Brenes:~/Documentos/poky/build$ time bitbake core-image-sato
Loading cache: 100% |#####| Time: 0:00:00
Loaded 3 entries from dependency cache.
ERROR: ParseError at /home/jorge/Documentos/poky/meta-openembedded/meta-python/recipes-devtools/python/python-functools32_3.2.3-2.bb:8: could not inherit file classes/setuputils.bbclass
Summary: There was 1 ERROR message shown, returning a non-zero exit code.

real    0m1.775s
user    0m0.188s
sys     0m0.033s
jorge@Jorge-Brenes:~/Documentos/poky/build$
```

Figura 11: Error al integrar meta-python.

```
jorge@Jorge-Brenes:~/Documentos/poky/build$ bitbake-layers show-layers
NOTE: Starting bitbake server...
layer      path                                          priority
=====
meta       /home/jorge/Documentos/poky/meta             5
meta-poky  /home/jorge/Documentos/poky/meta-poky        5
meta-yocto-bsp /home/jorge/Documentos/poky/meta-yocto-bsp  5
meta-oe    /home/jorge/Documentos/poky/meta-openembedded/meta-oe  6
meta-python /home/jorge/Documentos/poky/meta-openembedded/meta-python  7
meta-tensorflow /home/jorge/Documentos/poky/meta-openembedded/meta-tensorflow 10
meta-raspberrypi /home/jorge/Documentos/poky/meta-openembedded/meta-raspberrypi 9
jorge@Jorge-Brenes:~/Documentos/poky/build$ time bitbake core-image-sato
Loading cache: 100% |#####| ETA: --:--:--
Loaded 0 entries from dependency cache.
Parsing recipes: 28% |#####| ETA: 0:00:37
```

Figura 12: Imagen sato con las capas agregadas.

Se logra integrar los archivos para la ejecución del programa, no obstante, no se encuentra la ubicación de los archivos `model.h5` y el `haarcascade_frontalface_default`, por otro lado, el `emotiondetect.py` se logra encontrar, como se muestra en la Fig. 13, por lo que la integración de los archivos fue exitosa.

```
du
dumpleases
emotiondetect.py
env
expiry
expr
f2py
f2py3
f2py3.9
faillog
fc-cache
fc-cat
fc-conflist
fc-list
fc-match
pinentry
pinentry-curses
pkcs11-conv
printf
psplash
psplash-default
psplash-write
pulseaudio
pydoc3
pydoc3.9
python3
python3.9
rctest
readbootlog
readlink
```

Figura 13: Archivo Emotiondetect.py en la ejecución de la imagen sato generada.

## Interacción 11

**Fecha:** 12/10/2021

**Tipo de sesión:** Grupal

**Actividad:** Junto a la compañera Kimberly se trata de integrar las dependencias para utilizar tensorflow en la imagen sato.

**Problemas:** No obstante, surge un error por más dependencias que ocupa la librería de tensorflow. Dicho error se muestra en la Fig. 14. Se sigue trabajando en su solución.

```
jorge@Jorge-Brenes:~/documentos/poky/build$ time bitbake core-image-sato
Loading cache: 100% | ETA: --:--:--
Loaded 0 entries from dependency cache.
Parsing recipes: 100% |#####| Time: 0:01:15
Parsing of 2154 .bb files complete (0 cached, 2154 parsed). 3426 targets, 99 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies
ERROR: Nothing PROVIDES 'python3-markdown' (but /home/jorge/documentos/poky/meta-openembedded/meta-tensorflow/recipes-framework/tensorflow/tensorboard/2.6.0.bb RDEPENDS on or otherwise requires it)
NOTE: Runtime target 'python3-markdown' is unbuildable, removing...
Missing or unbuildable dependency chain was: ['python3-markdown']
NOTE: Runtime target 'tensorboard' is unbuildable, removing...
Missing or unbuildable dependency chain was: ['tensorboard', 'python3-markdown']
NOTE: Runtime target 'tensorflow' is unbuildable, removing...
Missing or unbuildable dependency chain was: ['tensorflow', 'tensorboard', 'python3-markdown']
ERROR: Required build target 'core-image-sato' has no buildable providers.
Missing or unbuildable dependency chain was: ['core-image-sato', 'tensorflow', 'tensorboard', 'python3-markdown']

Summary: There were 2 ERROR messages shown, returning a non-zero exit code.

real    1m18.247s
user    0m0.244s
sys     0m0.043s
```

Figura 14: Error tras incorporar dependencias de tensorflow.

## Interacción 12

**Fecha:** 15/10/2021

**Tipo de sesión:** Grupal

**Actividad:** Se realizó un cambio de código en donde haga uso del tensorflow lite, no obstante, este código ocupa la librería de scikit-image para hacer uso de skimage.transfor, la cual no se tenía contemplada anteriormente por lo que se es necesario agregarla al Yocto Project, esta es necesaria para ejecutar el nuevo código.

Por otro lado, se realiza una consulta con el profesor para mostrar lo obtenido hasta el momento y pedirle ayuda con respecto a la integración de tensorflow al yocto. El profesor nos contacta con Frander Díaz para realizar las consultas. Se contacta al compañero y se arregla una reunión para el día siguiente.

Seguidamente de ello, se empieza el informe de entrega.

**Problemas:** Sigue habiendo problemas con agregar tensorflow al yocto. Ahora hay más debido a que la librería scikit-image no se puede integrar, ya que no se encuentra dicha receta.

## Interacción 13

**Fecha:** 16/10/2021

**Tipo de sesión:** Grupal

**Actividad:** Se lleva a cabo una reunión grupal con Frander Díaz en donde se llevan todas las dudas existentes. El compañero nos explica algunos conceptos que no se tenía claros acerca del flujo de trabajo de yocto y nos dió varios tips útiles para desarrollar el proyecto. Aconseja cambiar el código de nuevo debido a que el árbol de dependencias crece al integrar la librería que se necesitaba, complicando así el proyecto.

Terminando la reunión con Frander se comienza una con la compañera Kimberly para cambiar el código al del principio y hacer uso del modelo en tensorflow lite creado anteriormente, lo cual sucede con éxito. Además, se hace la integración de la capa de tensorflow lite en lugar de la de tensorflow

Además, se vuelve a realizar la imagen desde cero para eliminar cosas que no se requieran, causa de la prueba y error y de esta forma simplificar la receta y tener el proyecto más limpio. Mientras se crea la imagen desde cero se escribe los pasos en el tutorial, en donde mi persona iba creando la imagen, la compañera Kimberly iba introduciendo los pasos a seguir en el documento. A continuación, en la Fig. 15 se muestran las capas que se integraron.

```
jorge@Jorge-Brenes:~/Documentos/Proyecto2/poky/build$ bitbake-layers show-layers
NOTE: Starting bitbake server...
layer      path                                          priority
=====
meta       /home/jorge/Documentos/Proyecto2/poky/meta    5
meta-poky  /home/jorge/Documentos/Proyecto2/poky/meta-poky 5
meta-yocto-bsp /home/jorge/Documentos/Proyecto2/poky/meta-yocto-bsp 5
meta-oe    /home/jorge/Documentos/Proyecto2/poky/meta-openembedded/meta-oe 6
meta-python /home/jorge/Documentos/Proyecto2/poky/meta-openembedded/meta-python 7
meta-networking /home/jorge/Documentos/Proyecto2/poky/meta-openembedded/meta-networking 5
meta-raspberrypi /home/jorge/Documentos/Proyecto2/poky/meta-openembedded/meta-raspberrypi 9
meta-tensorflow-lite /home/jorge/Documentos/Proyecto2/poky/meta-openembedded/meta-tensorflow-lite 6
meta-emotiondetect /home/jorge/Documentos/Proyecto2/poky/meta-emotiondetect 5
```

Figura 15: Capas finales para el proyecto. Fuente: Propia.

**Problemas:** Ocurre un error al cocinar la imagen, más específicamente en la capa de tensorflow lite, el error se muestra en la Fig. 16.

```
ERROR: Task (/home/jorge/Documentos/poky/meta-openembedded/meta-tensorflow-lite/recipes-framework/tensorflow-lite/python3-tensorflow-lite_2.6.0.bb:do_install) failed with exit code '1'
NOTE: Tasks Summary: Attempted 7421 tasks of which 7331 didn't need to be rerun and 1 failed.

Summary: 1 task failed:
  /home/jorge/Documentos/poky/meta-openembedded/meta-tensorflow-lite/recipes-framework/tensorflow-lite/python3-tensorflow-lite_2.6.0.bb:do_install
Summary: There was 1 ERROR message shown, returning a non-zero exit code.

real    18m22,815s
user    0m1,622s
sys     0m0,371s
```

Figura 16: Error en la capa de tensorflow lite. Fuente: Propia.

No obstante, al compañero Fabián le logra cocinar bien, por lo que se verifica más profundamente las capas y recetas. Con ello se llega a la conclusión de que el error procede al tener una máquina distinta, por lo que se procede a escribir la máquina correspondiente a la raspberry a utilizar y se procede a cocinar nuevamente, lo que termina en éxito, obteniendo así la imagen final para incorporar en la raspberry.

## Interacción 14

**Fecha:** 17/10/2021

**Tipo de sesión:** Grupal

**Actividad:** Se realiza la integración de la imagen a la tarjeta SD para poder incorporarla a la Raspberry Pi 2.

**Problemas:** Primeramente se cocina la imagen con *time bitbake basic-image-rpi*, sin embargo, no se completo la totalidad de la imagen por lo que se hace un cambio de la imagen cocinada por la sato por lo que se aplica el *time bitbake core-image-sato* terminando esta última con éxito.

Debido a que el proyecto se realiza en una máquina virtual con el sistema operativo de Ubuntu Linux no hay reconocimiento de la tarjeta SD por lo que no se puede hacer el comando para cocinar la imagen en la tarjeta.

Para su solución se sube el archivo *core-image-sato-raspberrypi2.rpi.sdimg* al drive y se descarga en Windows donde se hace uso de una utilidad llamada “Raspberry Pi Imager”, la cual se puede descargar en [9]. Esta aplicación ayuda a cocinar la imagen en la tarjeta SD.

Una vez aplicado esto se procede a probar en la Raspberry 2 lo que concluye en éxito. En la Fig. 17 se muestra una prueba en la Raspberry Pi 2

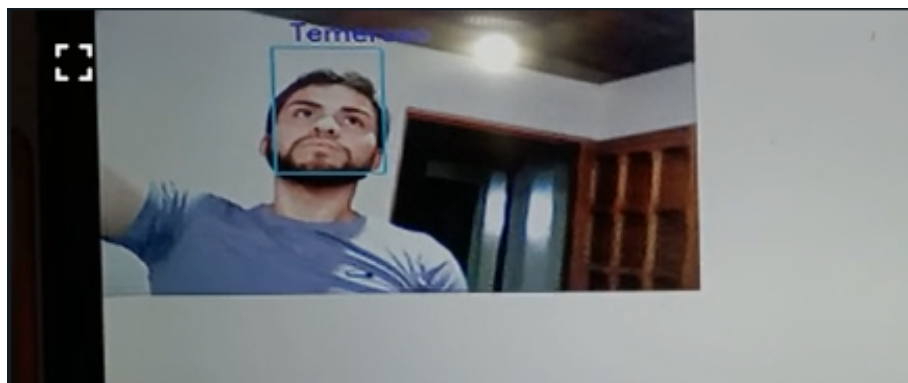


Figura 17: Prueba del proyecto 2 en la Raspberry Pi 2. Fuente: Propia.

## Referencias

- [1] A. Balaji, “Emotion Detection”, s.f. [Online] Disponible en: <https://github.com/atulapra/Emotion-detection>.
- [2] G. Solano, “Reconocimiento de emociones (EigenFaces, FisherFaces, LBPH)—Python – OpenCV”, 2020. [Online] Disponible en: <https://omesva.com/reconocimiento-de-emociones-opencv-python/>.
- [3] Programador Clic, “Reconocimiento de emociones OpenCV (implementado en Python, súper simple)”, s.f. [Online] Disponible en: <https://programmerclick.com/article/43501075874/>.
- [4] TensorFlow, “TensorFlow Lite”, s.f. [Online] Disponible en: <https://www.tensorflow.org/lite/guide?hl=es-419>.
- [5] TensorFlow, “Conversor de TensorFlow Lite”, s.f. [Online] Disponible en: <https://www.tensorflow.org/lite/convert?hl=es>.
- [6] TensorFlow, “Ejemplos de apps de TensorFlow Lite”, s.f. [Online] Disponible en: <https://www.tensorflow.org/lite/examples>.
- [7] Programador Clic, “Escritura y lectura de archivos .h5 (HDF5)”, s.f. [Online] Disponible en: <https://programmerclick.com/article/4092208083/>.
- [8] ICHI.PRO, “Implemente sus modelos de aprendizaje profundo en dispositivos de IoT con TensorFlow lite”, s.f. [Online] Disponible en: <https://ichi.pro/es/implemente-sus-modelos-de-aprendizaje-profundo-en-dispositivos-de-iot-con-tensorflow-lite-169134954367059>.
- [9] Raspberry Pi, “Raspberry Pi OS”, s.f.[Online] Disponible en: <https://www.raspberrypi.com/software/>.