

Universidad Nacional de La Matanza  
Departamento de Ingeniería e Investigaciones Tecnológicas  
Código: C170

Título del Proyecto:

*ANÁLISIS DE TÉCNICAS DE STREAMING DE VIDEO - IMPLEMENTACIÓN SOBRE IPV6*

Programa de Investigación:

*PROINCE*

Director del Proyecto:

*GIULIANELLI, Daniel Alberto*

Co-Director:

*RODRÍGUEZ, Rocío Andrea*

Integrantes del Proyecto:

DOCENTES: *BLANCO, Gabriel Esteban; VERA, Pablo Martín; MARKO, Isabel Beatriz; FERNANDEZ, Víctor Manuel; TRIGUEROS, Artemisa; CRUZADO, Graciela; MORENO, Edgardo Javier*

ALUMNOS: *CAMMARANO, Pablo; CORNEJO, María Antonella; DOGLIOTTI, Mariano Gastón; VALLES Federico Ezequiel; VALLES, Gabriela Yanina*

Fecha de inicio: *01/01/2015*

Fecha de finalización: *31/12/2016*

Resumen:

En el presente proyecto se analizarán protocolos sobre IPv6 para el manejo de Video mediante Streaming con el objetivo de construir una aplicación que permita en forma nativa establecer comunicación en primera instancia dentro de una LAN y luego al exterior de la universidad (siendo uno de los nodos de interés para la primera fase de pruebas la UTN-FRM). Lograr este objetivo implicará analizar la fragmentación de video, técnicas para comprimir, prioridad de paquetes para evitar retardos, etc. También se plantea como objetivo generar un prototipo de hardware para motorizar cámaras sencillas USB las cuales puedan detectar el sonido ambiente y en base a ello enfocar al orador, generándose un prototipo el que será montado en el laboratorio del GIDFIS (Grupo de Investigación, Desarrollo y Formación en Innovación de Software), para lo cual se comprarán algunos componentes de hardware. Este prototipo será autónomo teniendo instalado en software a construir pudiéndose entonces enchufar a un monitor externo o televisor.

Palabras claves:

IPv6, Streaming, Video, Prototipo, Hardware Dedicado

Área de conocimiento: Ingeniería de Comunicaciones Electrónica y Control

Código de Área de Conocimiento: 1800

Disciplina: Computación

Código de Disciplina: 1802

Campo de Aplicación: Computación

Código de Campo de Aplicación: 1802



## Análisis de Técnicas De Streaming De Video - Implementación sobre IPV6

### Resumen

---

En el presente proyecto se analizarán protocolos sobre IPv6 para el manejo de Video mediante Streaming con el objetivo de construir una aplicación que permita en forma nativa establecer comunicación en primera instancia dentro de una LAN y luego al exterior de la universidad (siendo uno de los nodos de interés para la primera fase de pruebas la UTN-FRM). Lograr este objetivo implicará analizar la fragmentación de video, técnicas para comprimir, prioridad de paquetes para evitar retardos, etc. También se plantea como objetivo generar un prototipo de hardware para motorizar cámaras sencillas USB las cuales puedan detectar el sonido ambiente y en base a ello enfocar al orador, generándose un prototipo el que será montado en el laboratorio del GIDFIS (Grupo de Investigación, Desarrollo y Formación en Innovación de Software), para lo cual se comprarán algunos componentes de hardware. Este prototipo será autónomo teniendo instalado en software a construir pudiéndose entonces enchufar a un monitor externo o televisor.

**Palabras Clave:** IPv6, Streaming, Video, Prototipo



## 1. Estructura

En este apartado se presenta la estructura del presente informe, la cual está basada en las actividades planificadas inicialmente en el protocolo del proyecto.

### 1. [Estructura](#)

### 2. [Introducción](#)

#### 2.1 [Selección del Tema](#)

#### 2.2 [Definición del Problema](#)

#### 2.3 [Justificación del Estudio](#)

#### 2.4 [Alcances del Trabajo](#)

#### 2.5 [Objetivos](#)

#### 2.6 [Hipótesis](#)

### 3. [Desarrollo](#)

#### 3.1. [Lugar y Tiempo de la Investigación](#)

#### 3.2. [Diseño de la Investigación](#)

#### 3.3. [Etapas Ejecutadas](#)

##### 3.3.1 [ETAPA 1: ESTADO DEL ARTE](#)

###### 3.3.1.1 [Tarea 1: Revisión del Protocolo IPv6](#)

###### 3.3.1.2 [Tareas 2 y 3: Proyectos Actuales en el Área y Trabajos Relacionados](#)

###### 3.3.1.3 [Tarea 4: Software Existente en el mercado - Comparación de Características y Prestaciones](#)

##### 3.3.2 [ETAPA 2: STREAMING DE VIDEO SOBRE IPv6](#)

###### 3.3.2.1 [Tarea 5 y 6: Análisis de Protocolos para Streaming y Compresión de Video](#)

###### 3.3.2.2 [Tarea 7: Técnicas de Calidad de Servicio y Disminución de Retardo](#)

###### 3.3.2.3 [Tarea 8: Multicast en IPv6](#)

##### 3.3.3 [ETAPA 3: ENTORNO DE DESARROLLO](#)

###### 3.3.3.1 [Tarea 9: Análisis de Entorno de Desarrollo y Versionado](#)

###### 3.3.3.2 [Tarea 10: Manejo de IPv6 en el Entorno de Desarrollo](#)

###### 3.3.3.3 [Tarea 11: Manejo de Sensores](#)

##### 3.3.4 [ETAPA 4: ANÁLISIS DE QoS BASADO EN AUDIO/VIDEO STREAMING EN TIEMPO REAL](#)

##### 3.3.5 [ETAPA 5: EVALUACIÓN DE IMPLEMENTACIÓN DE HARDWARE](#)

###### 3.3.5.1 [Tarea 12: Análisis de los sensores y sus posibilidades en IPv6](#)



- 3.3.5.2 [Tarea 13: Evaluación Integración de un controlador de motorización para cámara USB<sup>1</sup>](#)
- 3.3.6 [ETAPA 7: DESARROLLO DE HARDWARE DEDICADO](#)
  - 3.3.6.1 [Tarea 14: Desarrollo con Sensores](#)
  - 3.3.6.2 [Tarea 15: Implementación](#)
  - 3.3.6.3 [Tarea 16: Pruebas](#)
- 3.3.7 [ETAPA 8: DESARROLLO DE APLICACIÓN DE STREAMING SOBRE UNA LAN](#)
  - 3.3.7.1 [Tarea 17: Generación de servidor de streaming sobre IPv6](#)
  - 3.3.7.2 [Tarea 18: Generación de cliente/player de streaming IPv6](#)
- 3.3.8 [ETAPA 9: PRUEBAS DE APLICACIÓN DE STREAMING](#)
- 3.3.9 [ETAPA 10: IMPLEMENTACIÓN<sup>2</sup>](#)
- 4. [Producción Científico-Tecnológica](#)
- 5. [Conclusiones](#)
- 6. [Bibliografía](#)
- 7. [Anexo A – Tabla Comparativa Software Existente](#)
- 8. [Anexo B - Instructivo de Instalación del Entorno de Desarrollo para Raspberry PI](#)
- 9. [Anexo C - Análisis y comparativa de retardos en técnicas de streaming](#)

---

<sup>1</sup> Se excluye la ETAPA 6 por ser la elaboración del informe de avance

<sup>2</sup> Se excluye la ETAPA 11 por ser la elaboración de este informe final

## 2. Introducción

### 2.1. Selección del Tema

En la Universidad Nacional de La Matanza se cuenta con conectividad a IPv6 y algunos equipos de trabajo ya han realizado investigaciones en el área. De hecho este proyecto se enmarca bajo el programa “Conectividad Ipv6” iniciado en el 2013, mediante el cual se atienden cuestiones de hardware (infraestructura) y software (servicios y aplicaciones). Es posible profundizar en las técnicas de streaming de video y generar por otra parte prototipos que permitan motorizar las cámaras para dirigirlas hacia el orador dentro de un salón con diversas personas y múltiples oradores. Aprovechando las ventajas de IPv6 además del conocimiento a generarse podrá tenerse como resultado una solución que incluya el desarrollo de software y hardware.

Actualmente se continúan utilizando técnicas de transición entre IPv4 e IPv6 es decir que conviven ambos protocolos. Si bien son protocolos de red, muchas de las características que incorpora IPv6 pueden favorecer a diversas aplicaciones. En el caso de streaming de video por ejemplo en VTC (Video Tele-Conferencias) la posibilidad de multicasts, calidad de servicio, entre otras, hacen la necesidad de construir una aplicación nativa.

La **figura 1** muestra como es la conectividad actual que cuenta el laboratorio 1 del Instituto de Investigación y Desarrollo, donde se lleva a cabo el presente proyecto.

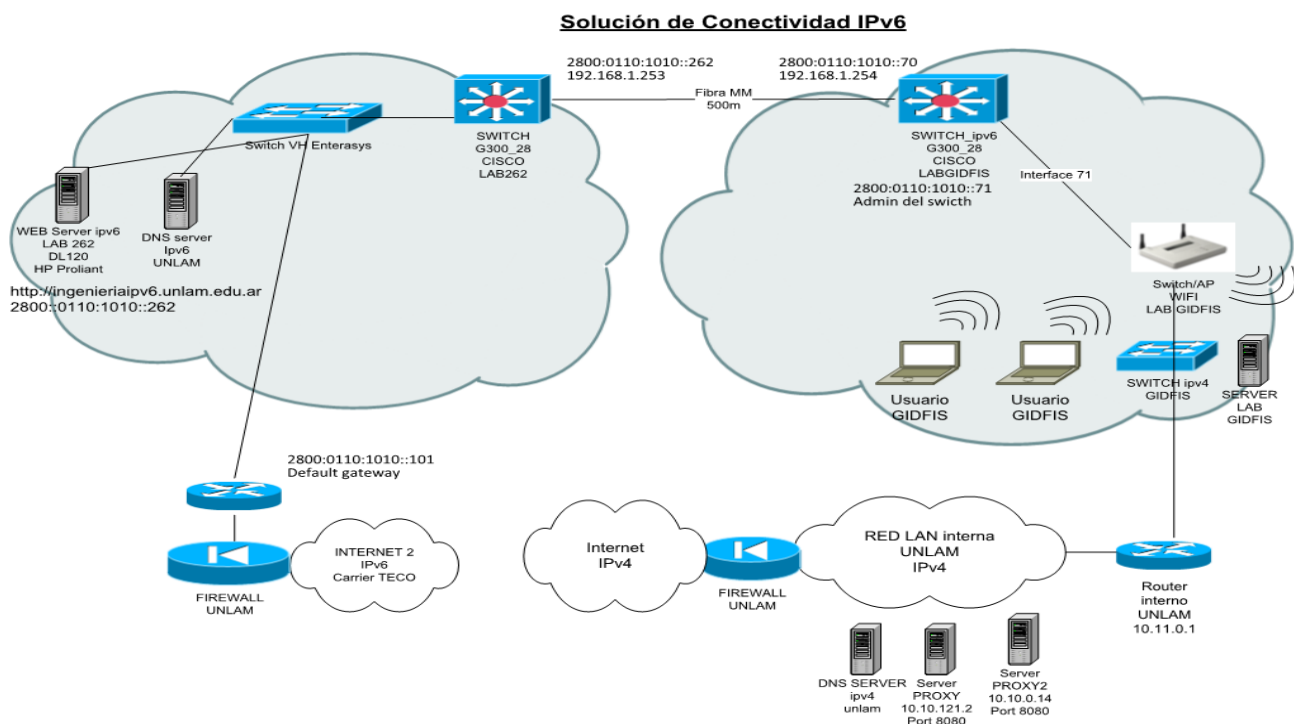


Figura 1. Mapa de Conectividad IPv6



## 2.2. Definición del Problema

Además de la escases de aplicaciones nativas para IPv6, el costo de las soluciones existentes para acceder a aplicaciones dedicadas de video teleconferencia son muy elevados. En algunos casos el software en cuestión se vende con hardware exclusivo, la compra del paquete (hardware + software) es un costo elevado, por otra parte en la Universidad Nacional de La Matanza se había comprado unas cámaras USB con software provisto por el fabricante para video-teleconferencias (VTC) en IPv6, las cuales ya no cuentan con garantía y por otra parte con el paso del tiempo el fabricante ha desestimado proveer actualizaciones del software para los sistemas operativos actuales (teniendo actualmente soporte para Windows XP ó anteriores). Entonces poder tener un software propietario abierto que pueda ser actualizado y por otra parte contar con cámaras actuales de mejor calidad y además motorizadas es una motivación importante que se suma a la posibilidad de profundizar en los conocimientos sobre el protocolo ipv6 y construir una solución integrar que junta conocimientos de desarrollo y también de hardware para generar un prototipo funcional.

## 2.3. Justificación del Estudio

Gradualmente IPv6 irá reemplazando la conectividad con el protocolo IPv4, la dificultad básica reside en poder actualizar todo el hardware que sólo puede funcionar con el protocolo anterior. Esta actualización lógicamente se va produciendo en forma paulatina (tomándose mayor tiempo que el planificado, lo que provoca que haya áreas en las que se han agotado las direcciones IPv4 asignadas [AZA07]). Cuando efectivamente IPv6 sea el protocolo de uso tradicional, surgirá un siguiente interrogante que es cómo desde las aplicaciones se pueden aprovechar las ventajas que este protocolo ofrece. La importancia del proyecto reside en el conocimiento que se profundizará sobre IPv6 en cuanto a streaming de video.

## 2.4. Alcances del Trabajo

En este proyecto se atienden principalmente tres cuestiones: (1) análisis del protocolo IPv6; (2) desarrollo de un prototipo hardware para VTC; (3) desarrollo de un software nativo en IPv6 para el hardware en cuestión. Para lo cual el tiempo del proyecto ha sido dividido en 11 etapas con tareas asociadas, las cuales fueron planteadas en el protocolo inicial del proyecto. Se ha respetado dicho cronograma (el cual se vuelca nuevamente en el ítem 3.1).



## 2.5. Objetivos

- Implementación de una solución de streaming en una LAN sobre IPv6
- Implementación de la solución con comunicación con un nodo externo
- Análisis de protocolos, Implementaciones de Hardware sobre IPv6
- Evaluación de técnicas para streaming de video
- Método para la compresión y descompresión

## 2.6. Hipótesis

Si bien IPv6 es un protocolo de red no sólo es importante analizar a nivel de hardware su implementación sino también construir aplicaciones nativas que permitan aprovechar sus características. Existen pocas aplicaciones nativas dedicadas.

## 3. Desarrollo

### 3.1. Lugar y Tiempo de la Investigación

Las tareas se realizan dentro del laboratorio 1 del Instituto de Investigación y Desarrollo, designado por el Departamento de Ingeniería e Investigaciones Tecnológicas. Siendo este proyecto una línea de investigación del GIDFIS (Grupo de Investigación Desarrollo y Formación en Innovación de Software).

Los tiempos de las tareas se llevaron a cabo en base al GANTT (ver **tabla 1** y **2**) diseñado previamente en el momento de la presentación del protocolo del presente proyecto. Se le han agregado numeración a las tareas indicadas con una T delante del número, como puede verse hay 14 tareas contenidas en las distintas actividades previstas las cuales tienen número sin letra identificadora, siendo 5 etapas las cuales se describirán en forma resumida en el presente informe. Descartándose la etapa 6 que es la elaboración del informe de avance.

La **tabla 1** se corresponde con el primer año de la investigación, hasta la actividad 6 inclusive. En la **tabla 2** están contenidas las actividades del segundo año, actividades numeradas desde 15 a 19 (planificadas para ser ejecutadas en el año 2016).

**Tabla 1.** Gantt previsto para el primer año del Proyecto (2015)

Actividades / Responsables 1er Año	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7	Mes 8	Mes 9	Mes 10	Mes 11	Mes 12
<b>1. Estado del Arte</b>	X	X	X									
T1. Revisión del Protocolo	x											
T2. Proyectos Actuales en el área		x	x									
T3. Trabajos Relacionados			x									
T4. Software Existente en el Mercado, tabla comparativa de características y Prestaciones												
<b>2. Streaming de Video sobre IPv6</b>				X	X	X	X					
T5. Análisis de Protocolos para Streaming de Video				x	x							
T6. Análisis de Técnicas para Comprimir Videos					x							
T7. Técnicas de Calidad de Servicio y Disminución de Retardo						x	x					
T8. Multicast en IPv6						x	x					
<b>3. Entorno de Desarrollo</b>								X	X	X		
T9. Análisis de Entorno de Desarrollo y Versionado								x	x			
T10. Manejo de IPv6 en el entorno de Desarrollo									x	x		
T11. Manejo de Sensores										x		
<b>4. Análisis de QoS basado en audio/video streaming en tiempo real</b>										x		
<b>5. Evaluación de Implementación de Hardware</b>								X	X	X	X	
T12. Análisis de los sensores y sus posibilidades en IPv6								x	x			
T13. Evaluación Integración de un controlador de motorización para cámaras									x	x	x	
<b>6. Informe de Avance</b>												X

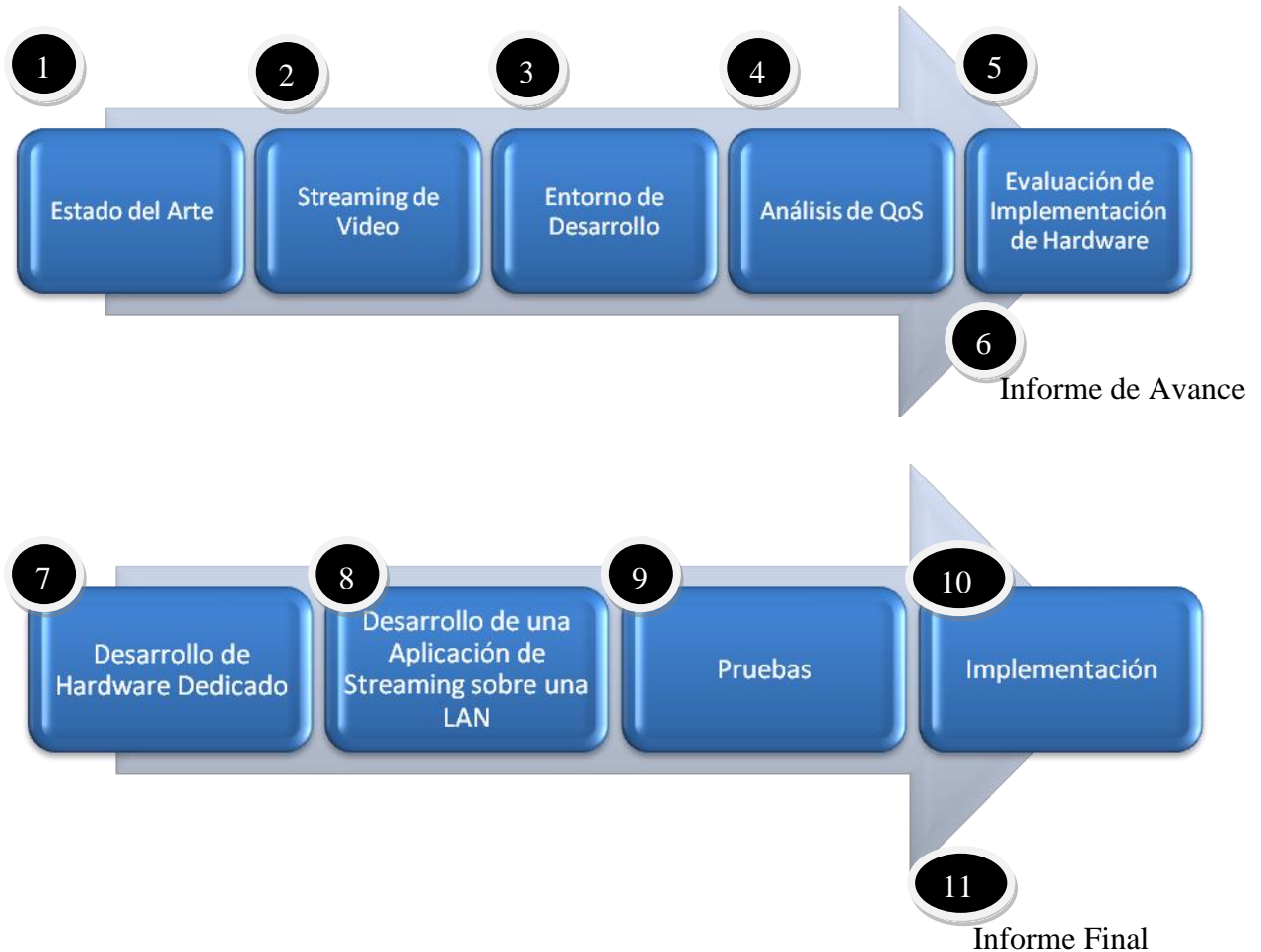
**Tabla 2.** Gantt previsto para el segundo año del Proyecto (2016)

Actividades / Responsables 2do Año	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7	Mes 8	Mes 9	Mes 10	Mes 11	Mes 12
<b>7. Desarrollo de Hardware Dedicado</b>	X	X	X	X	X	X	X	X				
T14. Desarrollo con Sensores	x	x	x	x	x	x						
T15. Implementación						x	x	x				
T16. Pruebas						x		x				
<b>8. Desarrollo de aplicación de streaming sobre una LAN</b>			X	X	X	X	X	X				
T17. Generación de servidor de streaming sobre IPv6			x	x	x							
T18. Generación de cliente/player de streaming IPv6						x	x	x				
<b>9. Pruebas de Aplicación de Streaming</b>									X	X	X	
<b>10. Implementación</b>											X	X
<b>11. Informe Final</b>												X



### 3.2. Diseño de la Investigación

Como puede observarse, en el GANTT presentado en el apartado anterior, las tareas del proyecto se encuentran divididas por Etapas. En la **figura 2** se muestran las 6 etapas realizadas en el primer año de ejecución del proyecto (las cuales se presentaron en la tabla 1), las siguientes etapas corresponden al segundo año del proyecto.



**Figura 2.** Etapas desarrolladas a lo largo del proyecto

### 3.3. Etapas Ejecutadas

En este apartado se presenta en forma resumida el resultado de cada una de las tareas ejecutadas a lo largo del proyecto.



### 3.3.1. ETAPA 1: ESTADO DEL ARTE

#### 3.3.1.1. Tarea 1: Revisión del Protocolo

Si bien desde su creación se anunció que IPv6 no reemplazaría a IPv4, todo parece indicar que esto pueda ocurrir para solucionar el problema del agotamiento de las direcciones IP. Algunas predicciones indicaban que antes del 2010 IPv4 debía estar sustituido previo al agotamiento de direcciones. En el 2007 la LACNIC mencionaba al “2011 como el año en el que se recomienda a todos los proveedores de Internet de la región a tener bloques de direcciones IPv6, ya en uso, para servicios de producción” [AZA07]. Sin embargo en la actualidad aún muchos usuarios y proveedores de servicios trabajan bajo IPv4, otros están implementando mecanismos de transición: Doble Pila, Entubamiento (Tunneling) ó Traducción de Encabezados. Para poder trabajar internamente con IPv4 y salir al exterior con una IPv6.

IPv6 tiene principalmente las siguientes ventajas:

- Mayor cantidad de direcciones: Mientras IPv4 tenía 232 direcciones IPv6 cuenta con 2128. “Se estima que si se repartiesen en toda la superficie de la Tierra habría  $6,67 \times 10^{23}$  IPs por  $m^2$ ” [INN12].
- Flexibilidad: Formato de cabecera más flexible que en IPv4 para agilizar el encaminamiento.
- Extensibilidad: IPv6 ha sido diseñado para ser extensible y ofrece soporte optimizado para nuevas opciones y agregados, permitiendo introducir mejoras en el futuro.
- Identificación de Flujo de Paquetes: Nueva etiqueta de flujo para identificar paquetes de un mismo flujo.
- Fragmentación en nodos: La fragmentación se realiza en el nodo origen y el re-ensamblado se realiza en los nodos finales, y no en los routers como en IPv4.
- Movilidad: incluida en el estándar, que permitirá cambiar de red sin perder la conectividad. IPv6 incluye mecanismos de movilidad más eficientes y robustos lo cual beneficiará no sólo a los usuarios de telefonía y dispositivos móviles, sino también (por ejemplo) tener buenas conexiones a internet durante los vuelos de avión.
- Multicast: Además de Unicast, Anycast y Broadcast. IPv6 incorpora Multicast (posibilidad de envío a un grupo de receptores interesados).



- Auto-configuración de los nodos finales, que permite a un equipo aprender automáticamente una dirección IPv6 al conectarse a la red.
- Aplicaciones: IPv6 permite el uso de jumbogramas (paquetes de datos de mayor tamaño 64K). Para dar mejor soporte a tráfico en tiempo real (ej. videoconferencia), IPv6 incluye etiquetado de flujos en sus especificaciones. Con este mecanismo los encaminadores o routers pueden reconocer a qué flujo extremo a extremo pertenecen los paquetes que se transmiten.
- Plug and Play: IPv6 incluye en su estándar el mecanismo "plug and play", lo cual facilita a los usuarios la conexión de sus equipos a la red. La configuración se realiza automáticamente. Esto permite que al conectar una máquina a una red IPv6, se le asigne automáticamente una (ó varias) direcciones IPv6.
- VOIP: Dos de los problemas actuales de los servicios de Voz sobre IP (VoIP) son QoS y NAT. Las comunicaciones pueden resultar en baja calidad de voz (QoS), y presentar dificultad para atravesar firewalls (NAT). Al incorporar IPv6 una gran cantidad de direcciones, no será necesario utilizar NAT, y sus nuevas capacidades de Plug and Play, seguridad, y QoS implicarán mejores conexiones de voz.

LACNIC (Latin American and Caribbean Internet Addresses Registry) indica que “la adopción temprana de IPv6 por la comunidad académica ha tenido como fin, por un lado la experimentación e investigación y por otro la formación de recursos humanos en el tema. A su vez, algunas necesidades propias de este sector se ven beneficiadas con características disponibles en este protocolo” [LACne]. Este organismo señala algunos ejemplos a nivel aplicación:

- La necesidad de contar con direcciones públicamente alcanzables, que permitan la interacción entre pares (en aplicaciones "peer to peer" como videoconferencia, operación remota de instrumentos, GRIDs, etc.)
- Características como multicast, necesario en aplicaciones como access grid y otras que requieren optimizar el uso del ancho de banda.
- Disponibilidad de IPSec como parte del stack, lo que facilita el despliegue de aplicaciones que requieren seguridad de extremo a extremo, como disponibilidad de recursos en malla (grids).



- Las nuevas posibilidades que brindan las características de QoS (Calidad de Servicio) incorporadas al protocolo.

### 3.3.1.2. Tareas 2 y 3: Proyectos Actuales en el Área y Trabajos Relacionados

Diversas empresas y entidades públicas están interesadas en implantar el protocolo IPv6 realizando un periodo de transición desde IPv4 a IPv6. Por ejemplo, en España el gobierno ha publicado las bases del “plan de fomento para la incorporación de IPv6” y la implantación del protocolo en el Ministerio de Industria, Energía y Turismo [ESP11]. Se prevé que para el 2013 los diversos países hayan completado gran parte de las etapas de pruebas del protocolo IPv6 y comiencen a ver como extender el ámbito de alcance, así como poner el foco en Servicios y Aplicaciones. En particular el grupo GRIDTICs (UTN - FRM) [GRI12], se plantea la importancia de comenzar a aprovechar las ventajas del nuevo protocolo (seguridad, multicast, etc), en aplicaciones y el 6 de Junio del 2012 dejó planteada esta necesidad “Ya lo probamos, ahora llegó la hora de usarlo”.

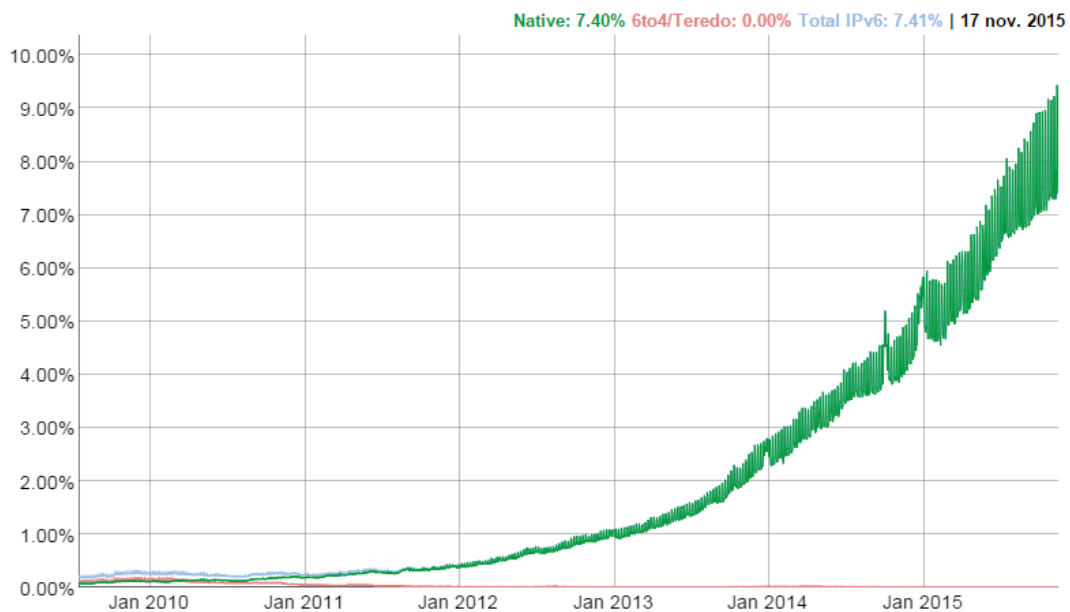
GT IPv6 [CLA15], tiene un listado de proyectos realizados, vigentes y planificados, en el área de IPv6, entre los planificados se encuentran:

- Desarrollo de aplicaciones con soporte IPv6 (Programación de Sockets)
  - Universidad Nacional Autónoma de México (México)
- VoIPv6 con SIP para IPv6
  - Universidad Nacional Autónoma de México (México)
  - Universidad Nacional de Río Cuarto (Córdoba)
- Multicast IPv6 (Opera Oberta, Open Student Television Network (OSTN), etc.)
  - Universidad Nacional Autónoma de México (México)
  - Universidad Nacional del Sur (Argentina)
- Firewalls con soporte IPv6
  - Universidad Nacional Autónoma de México (México)
  - Universidad Nacional de Río Cuarto (Córdoba)

A nivel académico IPv6 es un tema en constante desarrollo. Algunos artículos han tratado la temática de transición de IPv4 a IPv6 mediante el estudio de métodos de acceso transparente como es el caso del trabajo [MER11]. Se continúa trabajando en la implementación y despliegue de IPv6 en redes académicas [CIC14] [MER14]. También el estudio de características puntuales como por ejemplo: multicast, calidad de servicio resulta de interés para diversos autores entre ellos [FAC13], [DIA10]; así como cuestiones de

movilidad sobre IPv6 [CEN12], [TAF14], [TAF15]. Actualmente el trabajo con redes de sensores de IPv6 en el marco de IoT (Internet de la Cosas) es un área en auge [GUA15], [MER13].

A nivel mundial es notable la adopción de IPv6 y su crecimiento marcado, ver **figura 3**. Un 7,41% de los accesos se realizan por IPv6 siendo prácticamente en su totalidad (7,40%) en forma nativa.



**Figura 3.** Crecimiento de Accesos Mediante IPv6 – Estadística Google

A nivel mundial los 10 países con mayor porcentaje de adopción de IPv6 son: Estados Unidos (22,82%), Alemania (19,57%), Portugal (19,46%), Grecia (17,31%), Perú (15,48%), Estonia (9,96%), Japón (8,66%), Malasia (8,31%), Noruega (7,92%), Ecuador (7,34%). No obstante en Argentina el porcentaje de adopción de IPv6 es bajo, representando tan solo un 0,03%. En la **tabla 3** se muestra el grado de adopción los países de América. Como era de esperarse en base a la referencia mundial Estados Unidos, Perú y Ecuador son los que tienen los mejores porcentajes en el continente americano, seguidos por Brasil (5,98%), Canadá (5,3%), Bolivia (2,8%). Siendo estos 6 países los que destacan del resto quienes no alcanzan el 1%. Como puede observarse en la tabla 3, hay 8 países sin adopción de IPv6. La tabla está ordenada por porcentaje lo cual permite generar un ranking en el cual pueden verse los 6 primeros puestos conformados por los países mencionados previamente, el puesto 9, 13 y 16 son compartidos por dos ó más países; quedando en el puesto 17 quedan aquellos países que no tienen adopción a IPv6 siendo el porcentaje 0%. Quedando Argentina junto a Uruguay en el puesto 13.

**Tabla 3.** Porcentaje de Adopción de Ipv6 en América – Estadística Google

	País	Porcentaje
1	Estados Unidos	22,82
2	Perú	15,48
3	Ecuador	7,38
4	Brasil	5,98
5	Canadá	5,30
6	Bolivia	2,80
7	Puerto Rico	0,37
8	Cuba	0,34
9	Guatemala	0,10
	Guyana	0,10
10	Belice	0,06
11	Honduras	0,05
12	México	0,04
13	Argentina	0,03
	Uruguay	0,03
14	Chile	0,02
15	Colombia	0,02
16	Costa Rica	0,01
	Haití	0,01
	Surinam	0,01
	Venezuela	0,01
17	Bahamas	0,00
	El Salvador	0,00
	Groenlandia	0,00
	Guayana Francesa	0,00
	Jamaica	0,00
	Nicaragua	0,00
	Panamá	0,00
	Paraguay	0,00
	República Dominicana	0,00
	Martinica	0,00

### 3.3.1.3. Tarea 4: Software Existente en el Mercado - Tabla Comparativa

Actualmente existen diversos programas que permiten hacer streaming de video, siendo en su mayoría pagos y algunos de ellos son adquiridos al comprar un equipo determinado. En este último caso el costo del equipamiento incluye tanto al hardware como al software, siendo muy elevado el costo que se tendría en comparación con el desarrollo de un prototipo que incluya la motorización de la cámara. Esto es uno de los fundamentos que da valor a la presente iniciativa de construir una solución autónoma. En el **anexo A** se presenta una tabla comparativa con diversas soluciones existentes.

También en el campo de las aplicaciones y dentro del streaming de video se destacan Skype y Hangouts que han adoptado IPv6. Estadísticas a nivel mundial, destacan que para



la franja etaria de 25 a 34 un 34% usa de uso de los servicios de VoIP y Chat, por medio de estos aplicativos [STA15a], [STA15b].

### 3.3.2. ETAPA 2: STREAMING DE VIDEO SOBRE IPV6

#### 3.3.2.1. Tarea 5 y 6: Análisis de Protocolos para Streaming y Compresión de Video

El streaming de video puede realizarse de dos maneras, tal como se señala en [CHA14] la **tabla 4** muestra dichas maneras.

**Tabla 4.** Alternativas para streaming de video

	En Directo o Tiempo Real	Sobre Demanda
Manejo del Contenido Multimedia	Tiempo Real (el contenido no está guardado)	Previamente grabado y almacenado en el disco rígido
Forma de Visualizar el contenido multimedia.	El usuario puede ver el inicio, medio o ya al final de la transmisión.	El usuario tiene acceso para ver todo el contenido en cualquier momento.
Interactividad	No posee interactividad	Amplia interactividad

Existen diversos protocolos a analizar, así como CODEC (Codificador-Decodificador) de video. Los cuales se sintetizan a continuación.

Protocolos<sup>3</sup>:

- a) En directo o Tiempo Real: Para transmitir videos en tiempo real es necesario utilizar protocolos que no estén basados en el protocolo TCP (Protocolo de Control de Transmisión), puesto que éste protocolo está orientado a la conexión y en caso que se produzca un error o se pierda un dato la información se vuelve a reenviar. Debido a ello es preciso utilizar protocolos basados en UDP, el protocolo más importante para realizar la transmisión de video y sonido en tiempo real es RTP (Protocolo de Transporte en Tiempo Real).
- b) Sobre Demanda: Cuando se realiza la transmisión de videos sobre demanda, el cliente es el que controla la recepción de los datos, por lo tanto se pueden utilizar protocolos basados en TCP, como HTTP (Protocolo de Transferencia de Archivos)

<sup>3</sup> Este apartado ha sido sintetizado de la tesis para la obtención del título de grado de los autores: Evelyn Chavez Paredes y Diego Armando Castro Arrobo. Titulada "Evaluación de codecs de video sobre tráfico multicast en IPV6 para el desarrollo de un prototipo en el laboratorio LIRSI-FIE." (2014).



y FTP (Protocolo de Transferencia de Hipertexto). Debido a que los protocolos HTTP y FTP son fiables y se construyen en la capa más alta de TCP, aseguran que los paquetes lleguen a su destino de manera secuencial. Cuando se esté utilizando protocolos basados en TCP existe la ventaja de reenviar los paquetes en caso de que éstos se pierdan en el camino. Para el presente trabajo se va hacer uso del protocolo RTP /MPEG (Grupo de Expertos en Imágenes en Movimiento) Transport Protocol para transmisiones sobre demanda y transmisiones en vivo, debido a que este protocolo trabaja conjuntamente con el protocolo UDP, permitiendo una comunicación de manera mucho más eficiente.

## CODEC

Todo archivo multimedia y en especial los videos, requieren gran capacidad de almacenamiento en la computadora, por ello el utilizar un CODEC permitirá poder almacenarlo con un tamaño menor al que debería ocupar originalmente dichos archivos. “CODEC es un CODificador- DECodificador, es decir, un programa que es capaz de comprimir y codificar video en su propio formato y también lo decodifica y descomprime” [CEN97].

## TIPOS DE COMPRESIÓN DE VIDEO

Dependiendo del método de compresión que se utiliza existen diferentes tipos de CODEC, por ende la compresión de video se lo hace de dos maneras, explicadas a continuación:

- a. **COMPRESIÓN CON PÉRDIDAS:** “El sistema de compresión elimina aquella información de las imágenes que es prácticamente inapreciable por el ojo humano. La calidad de imagen depende directamente de la cantidad de información que se haya eliminado, esto se conoce con el nombre de grado de compresión. A mayor grado de compresión menor calidad de imagen”. [UNE10].
- b. **COMPRESIÓN SIN PERDIDAS** Este tipo de compresión se refiere a que conserva todos los datos de la señal original para cuando llega el momento de la descompresión, se obtenga la imagen tal y como era en un principio. La compresión sin pérdida de datos, es utilizada para comprimir archivos o información que contienen datos que no pueden ser degradados o perdidos, como pueden ser documentos de texto, archivos ejecutables, etc.





La **tabla 5** (la cual ha sido resumida de la tabla II.2 de [CHA14]) muestra una interesante comparativa entre 3 CODEC de video que corresponde al ítem a de la calificación previa (Comprensión sin Pérdidas).

**Tabla 5.** Comparativa de CODEC de video

	<b>H.264</b>	<b>Xvid</b>	<b>Theora</b>
<b>Características y Ventajas</b>	Reduce el flujo de bits. Proporciona buena calidad de imagen. Mayor flexibilidad. Alta robustez de errores.	Factor de compresión alto. Compatible con diferentes sistemas operativos. Garantiza mejor calidad de audio y video.	Multiplataforma, adecuado para internet sobre http, transmite a una tasa de bits variable, usa DCT
<b>Usabilidad</b>	Videoconferencia Televisión por cable y HD Video streaming Almacenamiento Bluerays	Video streaming Comprimir una película DVD	Reproduce videos con formato Ogv y Ogg Streaming de video
<b>CODEC similar y contenedores</b>	MPEG-4 parte 10 MP4	MPEG-4 parte 2 AVI	MPEG-4 parte 2 OGG Vorbis
<b>Algoritmos de comprensión predecesores</b>	MPEG-1 MPEG-2 H.261 H.263	Basado en las librerías que utilizan los CODEC's DivX 4 y OpenDivX	VP3 MPEG 1 MPEG 2 H.263

### 3.3.2.2. Tarea 7: Calidad de Servicio y Disminución de Retardo

La calidad de Servicio (QoS) está contemplada de forma intrínseca dentro del protocolo de IPv6.

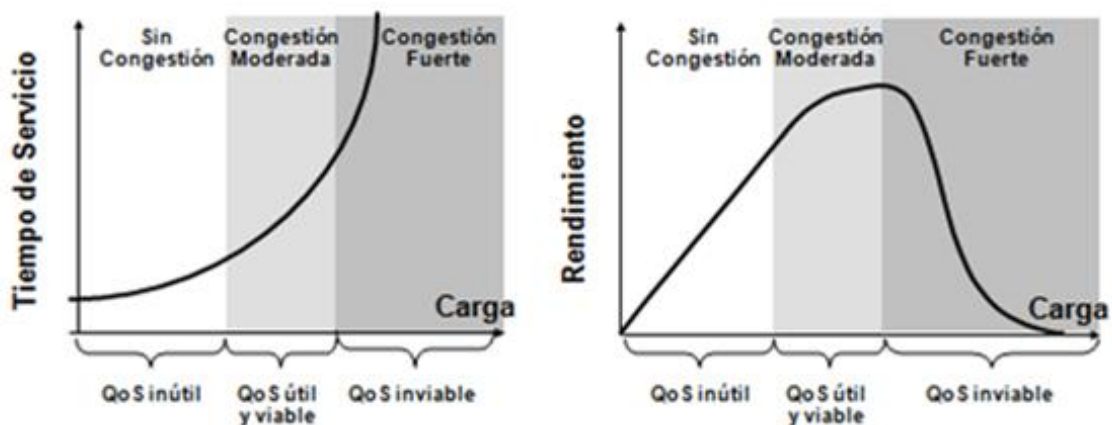
Existen tres niveles de servicio: mejor esfuerzo, servicio diferenciado y servicio garantizado/integrados.

- a) Mejor esfuerzo o best-effort: Es cuando la red hace todo lo posible para entregar el paquete a su destino, pero no hay garantía de que esto ocurra. Este es el modelo utilizado por las aplicaciones de FTP y HTTP.
- b) Servicios integrados: El modelo de Servicios Integrados provee a las aplicaciones de un nivel garantizado de servicio, negociando parámetros de red de extremo a

extremo. La aplicación solicita el nivel de servicio necesario con el fin de operar apropiadamente y se basa en la QoS para que se reserven los recursos de red necesarios antes de que la aplicación comience a operar.

- c) Servicios diferenciados: Este incluye un conjunto de herramientas de clasificación y de mecanismos de cola que proveen a ciertas aplicaciones o protocolos con determinadas prioridades sobre el resto del tráfico en la red.

Con ancho de banda suficiente se resuelven “casi” todos los problemas. Sería muy fácil dar Calidad de Servicio (QoS) si las redes nunca se congestionaran. Para ello habría que sobredimensionar todos los enlaces, cosa no siempre posible o conveniente. Para dar QoS con congestión es preciso tener mecanismos que permitan dar un trato distinto al tráfico preferente y cumplir el SLA (Service Level Agreement). La **figura 4** plantea los efectos de la congestión en el tiempo de servicio y rendimiento y ha sido extraída de [3CU13].



**Figura 4.** Efectos de la congestión en el tiempo de servicio y rendimiento

Es bien sabido que incluso desde una perspectiva de optimizar el uso global de los recursos no es deseable una excesiva carga en los enlaces. Cuando la carga aumenta el tiempo de servicio crece de forma exponencial y como consecuencia de esto las aplicaciones no pueden funcionar o retransmiten la información que creían perdida. Por tanto a partir de un cierto nivel de carga no solo crece el tiempo de servicio, sino que disminuye el rendimiento obtenido del enlace debido a las retransmisiones.

El objetivo de la Calidad de Servicio es asegurar que en casos de carga relativamente elevada (la zona marcada como de ‘congestión moderada’ en la gráfica) las aplicaciones que lo requieran podrán disfrutar de un tiempo de servicio reducido. Si la red tiene siempre niveles de carga inferiores el funcionamiento se complica y no se obtiene beneficio al aplicar mecanismos de Calidad de Servicio. Si la red tiene normalmente niveles fuertes de



gestión los mecanismos de Calidad de Servicio difícilmente serán capaces de asegurar el nivel de calidad pedido a las aplicaciones que así lo requieran.

**Tabla 6.** Parámetros de Calidad de Servicio

Parámetro	Unidades	Significado
Ancho de Banda (bandwidth)	Kb/s	Indica el caudal máximo que se puede transmitir
Retardo (delay) o latencia (latency)	Ms	El tiempo medio que tardan en llegar los paquetes
Jitter	Ms	La fluctuación que se puede producir en el Retardo
Tasa de Pérdidas (loss rate)	%	Proporción de paquetes perdidos respecto a los enviados

El campo ToS fue implementado dentro del grupo de diseño de IPv4 como un campo de ocho bits, compuesto por un valor de precedencia IP de tres bits y cuatro bits indicadores. Su función es especificar parámetros de prioridad, retardo, rendimiento y fiabilidad. De este modo, los paquetes con diversas opciones de ToS se pueden manejar con diferentes niveles de servicio dentro de la red. De acuerdo a la recomendación RFC 791 [IET81] el ToS proporciona una indicación de los parámetros de calidad de servicio deseados. Éstos se utilizan para especificar el tratamiento del datagrama durante su transmisión en una red en particular. Algunas redes ofrecen prioridad de servicio, lo cual consiste en considerar el tráfico de alta prioridad como más importante que el resto (generalmente aceptando sólo tráfico por encima de cierta prioridad en momentos de sobrecarga). La elección más común es un compromiso de tres factores: bajo retardo, alta fiabilidad y alto rendimiento.

El campo ToS está compuesto por un campo de precedencia, tres indicadores D, T, R y dos bits no utilizados; el campo de precedencia utiliza ocho niveles, de cero (rutinaria) a siete (paquete de control de red); los tres bits indicadores permiten especificar qué es lo que más interesa, el retardo, el rendimiento o la fiabilidad. A continuación, se presenta un resumen del campo ToS:

- Bits 0-2 = prioridad.
- Bit 3:0 = retardo normal, 1 = bajo retardo.
- Bit 4:0 = rendimiento normal, 1 = alto rendimiento.
- Bit 5: 0 = fiabilidad normal, 1= alta fiabilidad.
- Bits 6-7 = reservado para uso futuro.

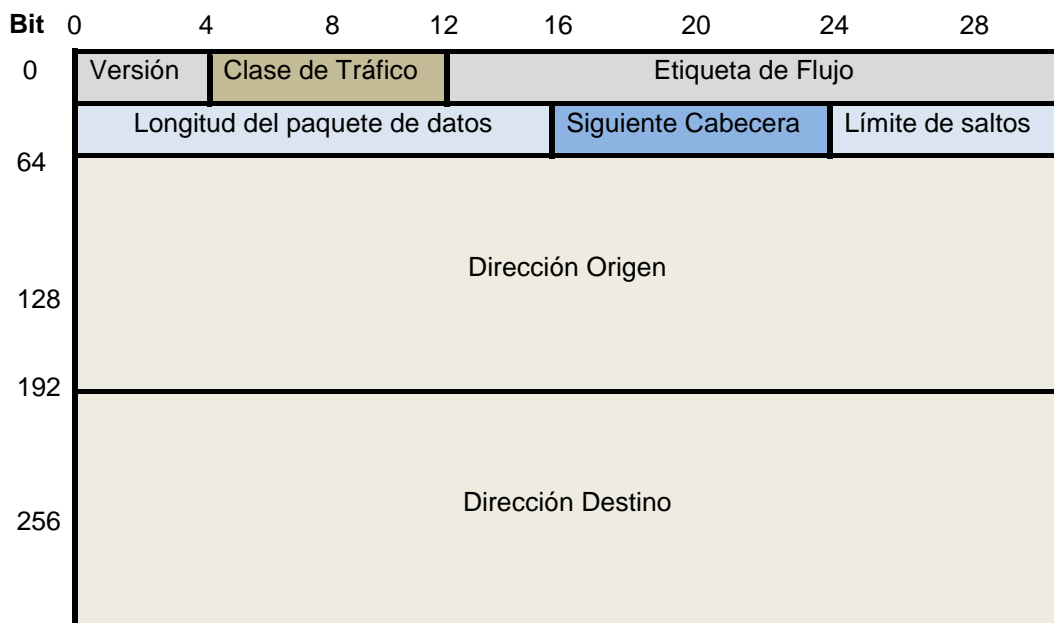


El subcampo de precedencia es una medida de importancia relativa al datagrama. Se utilizan ocho-niveles de precedencia. IP tratará de proporcionar un tratamiento preferencial a los diagramas con precedencias superiores.

- 111-Control de Red
- 110-Control Entre Redes
- 101 - CRITICO/ECP
- 100 - Muy urgente (Flash Override)
- 011-Urgente (Flash)
- 010-Inmediato
- 001-Prioridad
- 000 - Rutina

El uso de los indicadores de retardo, rendimiento y Habilidad puede incrementar el coste (en cierto sentido) del servicio. En muchas redes un mejor desempeño de uno de estos parámetros significa un peor desempeño de otro. Por lo tanto, excepto para casos excepcionales, no se deben establecer más de dos indicadores.

Por su parte, el protocolo IPv6 tiene dos campos que pueden ser utilizados como herramientas para implementar QoS: Etiqueta de Flujo y Clase de Tráfico (ver **figura 5**).



**Figura 5.** Estructura del Encabezado de un paquete IPv6

El campo Etiqueta de Flujo de 20 bits en la cabecera IPv6 se agrega para permitir el etiquetado de paquetes que pertenecen a flujos de tráfico particulares y puede ser usado por el origen para etiquetar secuencias de paquetes para las cuales solicita un manejo



especial por parte de los enrutadores IPv6, tal como la calidad de servicio no estándar o el servicio en tiempo real. Se exige a los hosts o a los enrutadores, que no dan soporte al campo Etiqueta de Flujo, poner el campo en cero al enviar un paquete, pasar el campo inalterado al reenviar un paquete e ignorar el campo al recibir un paquete.

El campo de ocho bits Clase de Tráfico en la cabecera IPv6 es utilizado por los nodos origen y/o enrutadores intermedios para identificar y distinguir entre las diferentes clases o prioridades de paquetes IPv6; su fondón es similar al campo ToS de IPv4.

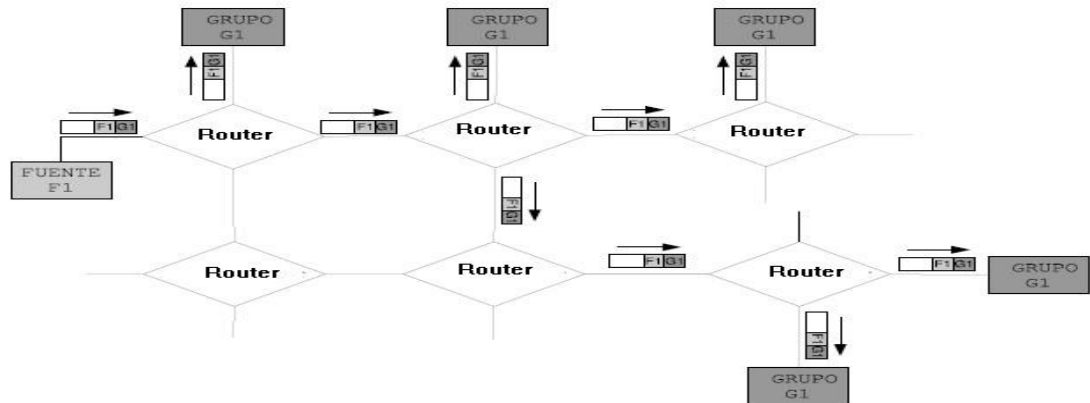
Los siguientes requisitos generales se aplican al campo Clase de Tráfico:

- La interface de servicio para el servicio IPv6 dentro de un nodo debe proporcionar un medio para que un protocolo de capa superior proporcione el valor de los bits Clase de Tráfico en los paquetes originados por éste. El valor por defecto debe ser cero para todos los ocho bits.
- Los nodos que soportan un uso específico de algunos o todos los bits Clase de Tráfico se les permite cambiarlos en los paquetes que los nodos originan, reenvían o reciben, como sea requerido para ese uso específico. Los nodos deben ignorar y dejar sin alterar los bits del campo Clase de Tráfico para los cuales no dan soporte a un uso específico.
- Un protocolo de capa superior no debe asumir que el valor de los bits Clase de Tráfico en un paquete recibido es el mismo que el valor enviado por el origen del paquete.

### 3.3.2.3. Tarea 8: Multicast en IPv6

Multicast, se basa en el concepto de grupo. Un grupo arbitrario de receptores que esperan recibir un particular flujo de datos. Dichos grupos no tienen sentido físico de ubicación, pueden encontrarse en cualquier parte de Internet.

En la comunicación multicast existe una fuente y un grupo de destinos (ver **figura 6**), por lo tanto, es una relación de uno a muchos. En este tipo de comunicación, la dirección de origen es una dirección unicast, pero la dirección de destino es la dirección de un grupo, formado por uno o más receptores.



**Figura 6.** Comunicación Multicast

En la **figura 6** se puede observar que la estructura de los paquetes es la misma: la dirección de origen es F1 (dirección del emisor) y la de destino es G1 (dirección del grupo).

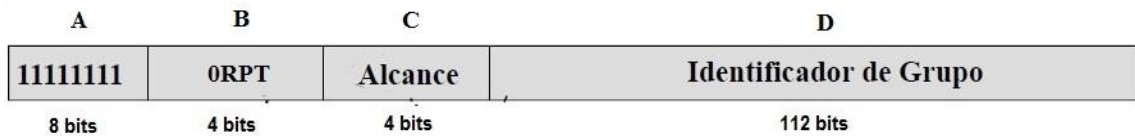
Entre las características Generales se destacan:

- Se realiza un único envío desde el origen, independientemente del número de destinos. Cada router se encarga de realizar las copias, si fuera necesario.
- Menor consumo de recursos, ya que cada enlace solo transporta una copia del paquete.
- A diferencia de la unidifusión múltiple<sup>4</sup>, en multicast no hay retardo entre los paquetes ya que no se transmiten copias desde el emisor. Esto es una ventaja en ciertas aplicaciones de tiempo real.
- Las direcciones multicast solo puede aparecer como dirección IP de destino.
- Los sistemas finales pueden ser origen y/o destino de los paquetes IP de multidifusión.
- Los routers que manejan tráfico multicast son los routers de multidifusión de Internet: estos manejan las direcciones multicast y además se encargan de realizar la copia de la información cuando sea necesario. Usan algoritmos de enrutamiento dinámico de multidifusión.

<sup>4</sup> La unidifusión múltiple es un método de comunicación en el cual el emisor envía N paquetes a N destinos. Presenta desventajas frente a la multidifusión, principalmente por el retardo que se genera en el origen al copiar los paquetes.



La **figura 7** muestra el formato de una dirección Multicast. Las partes se encuentran identificadas con letras las cuales son descriptas a continuación:



**Figura 7.** Formato de una dirección Multicast

**A)** Los primeros 8 bits de una dirección multicast siempre son “1” (11111111), que en hexadecimal sería FF. Sirve para diferenciarlas de las demás direcciones.

**B) Banderas RPT:**

Cada uno de estos *flags* tiene un significado. A continuación se detallan:

R = 1 (la dirección multicast incorpora la dirección de un Punto de Reunión)

P = 1 (la dirección multicast está basada en un prefijo unicast)

T = 0 es una dirección asignada permanentemente por la IANA/ICANN<sup>5</sup> (no es transitoria)

T = 1 no es una dirección asignada permanentemente (es transitoria)

**C) Alcance del grupo de multidifusión** (también llamado ámbito o límite):

Está representado por un número entero de 4 bits.

0 (0000): reservado.

1 (0001): alcance de nodo local.

2 (0010): alcance de enlace local.

5 (0101): alcance de sitio local (varios enlaces).

8 (1000): alcance de organización local (compuestas de varios centros o sitios).

E (1110): alcance global (en Internet).

F (1111): reservada.

**D)** El **identificador de grupo** permite identificar al grupo multicast al que nos referimos dentro de un determinado ámbito.

<sup>5</sup> ICANN (Internet Corporation for Assigned Names and Numbers) es una organización mundial que se encarga de la asignación de direcciones IP, identificadores de protocolos, entre otras tareas. IANA (Internet Assigned Numbers Authority) es actualmente un departamento de la ICANN.



Los identificadores de grupo permanentemente asignados son independientes del ámbito. Los identificadores de los grupos temporales, en cambio, solo son relevantes para un ámbito determinado.

En las **tablas 7 y 8** se muestran las direcciones multicast reservadas.

**Tabla 7.** Direcciones Reservadas en alcance nodo-local

Dirección IPv6	Descripción
FF01::1	Dirección de multidifusión a todos los nodos
FF01::2	Dirección de multidifusión a todos los routers

**Tabla 8.** Direcciones Reservadas en alcance enlace-local

Dirección IPv6	Descripción
FF02::1	Dirección de multidifusión a todos los nodos
FF02::2	Dirección de multidifusión a todos los routers
FF02::4	Todos los routers DVMRP <sup>6</sup>
FF02::5	Todos los routers OSPFIGP <sup>7</sup>
FF02::6	Todos los routers designados OSPFIGP
FF02::9	Todos los routers RIP <sup>8</sup>
FF02::D	Todos los routers PIM <sup>9</sup> .
FF02::1:FFXX:XXXX	Solicited-Node Address. Permite calcular la dirección multicast a partir de la unicast o anycast de un determinado nodo. Se sustituyen los 24 bits de menor peso ("X") por los mismos bits de la dirección original.

### **Correspondencia IP-MAC en Multicast:**

Existe una correspondencia entre las direcciones IP y las MAC en unidifusión, difusión y multidifusión.

En el caso de la multidifusión, la dirección MAC se obtiene a partir de la dirección del grupo. Las tramas de multidifusión tienen que poder ser transmitidas o recibidas por la capa de enlace de datos. El prefijo de las tramas Ethernet de multidifusión en IPv6 es 33-33.

Las tarjetas de red "captan" todas las tramas que comiencen con 33-33 y se pasan al nivel de IP. Las tarjetas más actuales pueden analizar los octetos restantes para saber si pertenece a dicho grupo de multidifusión, y de esa manera evitar enviar información innecesaria al nivel IP.

<sup>6</sup> Protocolo de enrutamiento multicast basado en vector distancia, se desarrolla más adelante.

<sup>7</sup> Protocolo de enrutamiento basado en estado del enlace.

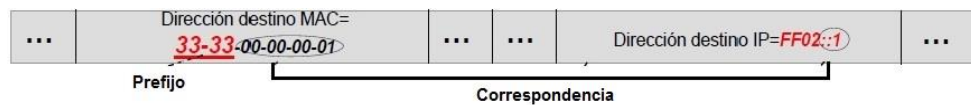
<sup>8</sup> Protocolo de Información de Enrutamiento basado en vector distancia, presenta ciertas limitaciones.

<sup>9</sup> Es una familia de protocolos de enrutamiento multicast, se desarrolla más adelante.



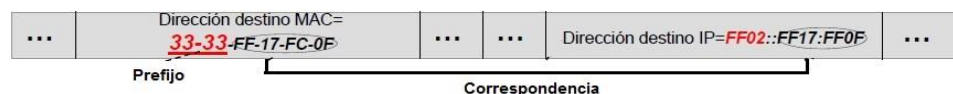
Cuando un usuario se agrega a un grupo de multicast, se le informa a la tarjeta de red que filtre las tramas multicast de ese grupo. Existen dos casos en los que hay que realizar una conversión de la IP a la MAC:

- A. Multidifusión a todos los nodos de la red (ver **figura 8**): En este caso, el objetivo es difundir un mensaje a todos los dispositivos de una red. Por lo tanto, se utiliza la dirección IP de multidifusión a todos los nodos vecinos (FF02::1). Al realizar la correspondencia entre dicha dirección IPv6 y la MAC, se utiliza el prefijo 33-33 seguido de “00-00-00-01”, esto último está relacionado con el “1” de la dirección IPv6.



**Figura 8.** Equivalente a la difusión limitada en IPv4

- B. Multidifusión a un grupo específico (ver **figura 9**): En este ejemplo, la dirección IPv6 del grupo multicast es: FF02::FF17:FF0F. Para construir la dirección MAC, se toma el “FF-17-FC-0F” de la dirección IPv6, que representa al grupo multicast. El protocolo MLD que administra los grupos multicast en una red se llama MLD (Multicast Listener Discovery Protocol). La versión actual es la MLDv2. No es un protocolo de enrutamiento dinámico, sino que se encarga de administrar la pertenencia de “procesos activos” a grupos de multidifusión. Los hosts usan este protocolo para indicarle a un router de multidifusión local (RML) que posee un proceso activo en un grupo de multidifusión de Internet. Los RML mantienen, usando el protocolo MLD, una lista con los grupos que tienen al menos un proceso activo en la red.



**Figura 9.** Ejemplo IPv6 – Multidifusión a un grupo específico

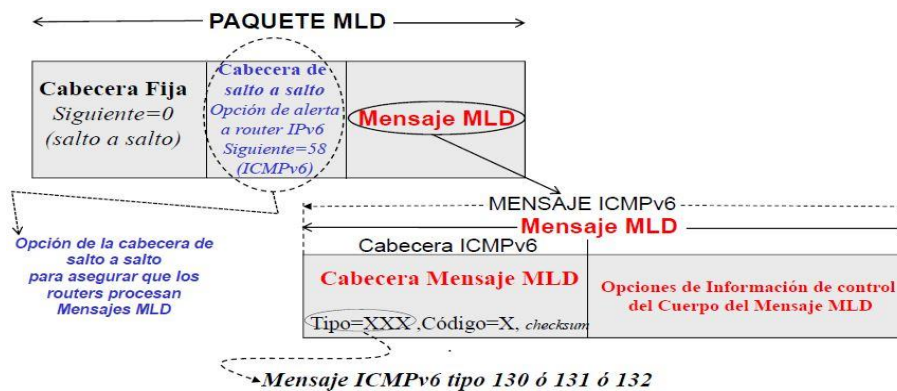
**Protocol Independent Multicast (MDL):** El protocolo MLD está compuesto por 3 mensajes ICMPv6<sup>10</sup>:

- Multicast Listener Query (ICMPv6 Type 130): es el mensaje de sondeo de pertenencia o consulta. Es enviado periódicamente por los routers. Hay dos tipos:

<sup>10</sup> Es el Protocolo de Mensajes de Control de Internet Versión 6.

- General Query: el campo de dirección multicast tiene valor “0”. De esa forma el router pregunta qué grupos multicast tienen participantes en la red local.
  - Group Specific: el campo de dirección multicast tiene una dirección específica. De esta forma el router pregunta si un grupo específico tiene participantes en la red.
- Multicast Listener Report (ICMPv6 Type 131): es el mensaje de informe de pertenencia al grupo. Un equipo deberá responder a cualquier consulta del RML con un mensaje de informe por cada grupo al que pertenece o desea pertenecer.
  - Multicast Listener Done (ICMPv6 Type 132): es el mensaje de informe de abandono de grupo.

Cuando un equipo observa que ningún proceso local está activo envía un informe de abandono a dicho grupo. El RML, al recibirlo, envía un mensaje de consulta especial que incluye la dirección de dicho grupo multicast y espera 10 segundos a que otro equipo le responda. En caso que ningún equipo responda con el informe de pertenecía, asume que el grupo está vacío y lo elimina de la lista.



**Figura 10.** Ejemplo IPv6 – Multidifusión a un grupo específico

El enrutamiento es un proceso por el cual se determina la ruta que van a tomar los paquetes para llegar a su destino. En el caso del tráfico multicast, existen ciertas funciones que se requieren para llevar a cabo el enrutamiento:

- a) Establecer una convención para identificar direcciones multicast: las direcciones IPv6 utilizan un prefijo de 8 bits con valor “1”.
- b) Cada router debe traducir una dirección multicast IPv6 a una lista de redes que contengan miembros del grupo multicast. Esta información permite construir un



- árbol de distribución de camino más corto (SPT) para llegar hacia todas las redes que tengan miembros del grupo.
- c) El router debe poder traducir una dirección IP multicast a una dirección MAC para poder entregar en la red de destino el datagrama multicast.
  - d) Las direcciones multicast pueden ser estáticas, pero generalmente son dinámicas y los hosts se pueden unir o abandonar grupos multicast dinámicamente.
    - Por eso se necesita un mecanismo que permita informar la incorporación o el abandono de un miembro al grupo.
    - Los routers deben intercambiar dos tipos de información:
      - Qué redes contienen miembros de un grupo en particular.
      - Información para calcular los caminos más cortos a cada red que contenga miembros del grupo.
        - a) Se necesita un algoritmo de enrutamiento para calcular los caminos más cortos a todos los miembros del grupo.
        - b) Cada router debe determinar la ruta de distribución multicast basándose en las direcciones destino y fuente.

En el enrutamiento multicast, un paquete recibido por un router puede ser reenviado a distintos destinos de distintas redes.

Para reenviar un paquete multicast se requiere de un árbol SPT.

Existen dos enfoques para construir dicho árbol:

- **Árbol source-based:** cada router tiene un SPT por cada grupo multicast que existe. Este SPT define el próximo salto para cada red que tenga miembros registrados para ese grupo.
- **Árbol group-shared:** en este caso, solo un router designado, llamado Rendezvous Point (RP) se encarga de la distribución del tráfico multicast. Cuando un router recibe un paquete multicast, lo encapsula en uno unicast y lo envía al RP, que luego consulta su tabla de enrutamiento para dirigir el paquete. En este enfoque, solo el RP, que tiene un SPT por cada grupo, está involucrado en la transmisión multicast.

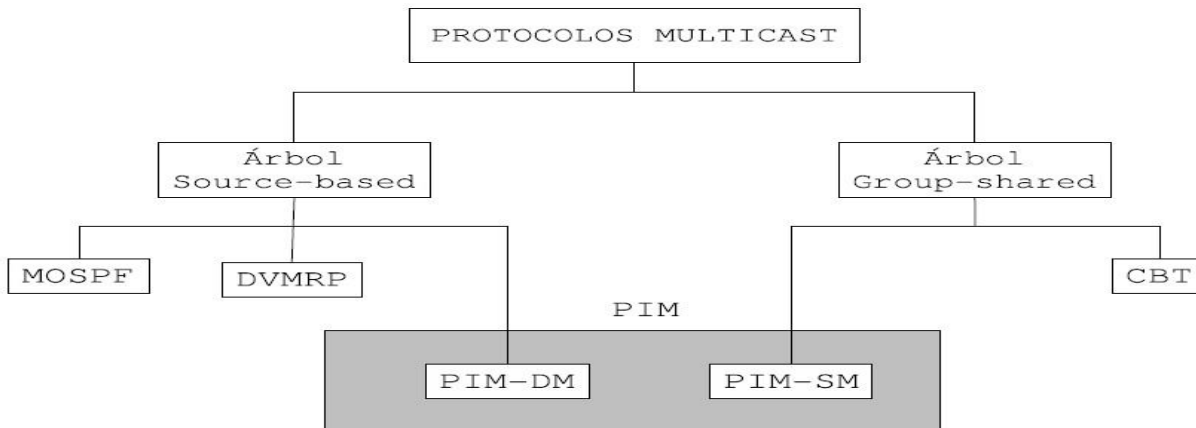


Figura 11. Modelo de Protocolo de Enrutamiento Multicast

- **Protocol Independent Multicast (PIM)**: Es una familia de protocolos que se desarrolló a fines de la década de los 90. Se caracterizan por no depender de ningún protocolo de enrutamiento específico, sino que aprovechan las tablas de enrutamiento existentes para reenviar los datos multicast.
- **PIM Dense Mode (PIM-DM)**: Este protocolo se recomienda cuando hay probabilidades que todos los routers participen en la distribución multicast y también para entornos multicast numerosos (por ejemplo, redes LAN). Este protocolo implementa el enfoque de árbol source-based, es decir, que cada router tiene una tabla con la interfaz de salida que tiene el camino más óptimo a cierto destino. PIM-DM utiliza un mecanismo push (empuje), es decir, empujar los datos multicast hacia los receptores. El router que implemente este protocolo envía tráfico multicast por todas sus interfaces. Si los routers downstream<sup>11</sup> no tienen conectados ningún receptor del tráfico multicast, envía un mensaje stop hacia el router upstream<sup>12</sup>. Estos mensajes se conocen como prune (poda) porque el router upstream podará su árbol de reenvío para eliminar esa rama en particular. El tráfico multicast es enviado hasta que un router downstream lo rechace.
- **PIM Sparse Mode (PIM-SM)**: Este protocolo se recomienda cuando existe alguna posibilidad que el router participe en la distribución multicast y es apropiado para entornos multicast dispersos, tales como redes WAN. Implementa el enfoque de árbol group-shared, es decir que tiene un RP como fuente del árbol. En caso que exista un área numerosa de actividad lejos del RP, ésta

<sup>11</sup> Son los routers que se encuentran conectados por “debajo” del router que envía el tráfico, son los que lo reciben.

<sup>12</sup> Es el router que se encuentra por “arriba”, el que envía los datos.



puede ser administrada más eficientemente con una estrategia de árbol source-based. PIM-SM utiliza una estrategia pull (solicitud), contrastando con la técnica push de PIM-DM. Esto significa que los routers deben hacer una solicitud específica de tráfico multicast antes de que los datos sean reenviados a ellos. Este protocolo se adapta a Internet, ya que reduce la sobrecarga y el ancho de banda utilizado para el tráfico multicast. Los routers generan periódicamente un mensaje hello para descubrir y mantener sesiones de estado con los vecinos. El router downstream puede enviar un mensaje join al upstream para unirse a un grupo multicast, indicando el grupo y la fuente a los que el router desea unirse. A partir de ese momento el router downstream empieza a recibir el tráfico multicast. Esto demuestra la gran diferencia que tiene con PIM-DM, donde el router upstream siempre está enviando tráfico multicast hasta que el downstream le indique que deje de hacerlo.

- **Distance Vector Multicast Routing Protocol (DVMRP):** Es el más antiguo de los protocolos para enrutamiento multicast. Es análogo a RIP, es decir, que es un protocolo por vector distancia que proporciona una limitada flexibilidad y funcionalidad. Es adecuado para redes pequeñas (no para Internet) debido a sus problemas de escalabilidad. El funcionamiento es muy similar al del protocolo PIM-DM, ya que ambos son protocolos del tipo denso. La principal diferencia es que DVMRP genera sus propias tablas de enrutamiento basándose en el vector distancia, mientras que PIM-DM aprovecha las tablas existentes en el router.
- **Multicast Open Shortest Path First (MOSPF):** Es una extensión del protocolo OSPF, por lo tanto requiere su utilización para poder funcionar. Esto hace que su aplicación esté limitada a universidades, empresas u organizaciones, donde se pueda implementar el protocolo OSPF en su totalidad. Es un protocolo basado en el estado de enlace. Esto significa que cada router realiza una serie de pruebas sobre sus enlaces para poder determinar el “costo” de cada uno de ellos. La principal diferencia con los basados en vector distancia (como DVMRP), es la velocidad de convergencia, que es mayor con MOSPF. Esto quiere decir que tarda menos en realizar una actualización del “mapa” de la red.
- **Core-Based Trees (CBT):** Cada grupo de multicast en la red posee un único árbol, con centro en un nodo llamado core o nodo central. Todos los mensajes a un grupo en particular son enviados como mensajes unicast hacia el core hasta que alcanzan un router que pertenezca al árbol de destino. Cuando un nodo del



árbol de destino recibe el paquete, lo envía hacia todos sus enlaces, excepto por el que vino.

- **Implementación de Multicast:** Las ventajas de implementar Multicast:
  - Existe un considerable ahorro de ancho de banda de la red y, por lo tanto, de la utilización de la red en recursos económicos. Este ahorro se obtiene principalmente porque un mensaje multicast transmitido a N receptores no se envía N veces, sino que se realiza un solo envío y cada router realiza las copias que fueran necesarias. Además, en cada enlace solo se transmite una copia del paquete, reduciendo la carga de la red.
  - Otro beneficio es que los receptores reciben los paquetes al mismo tiempo (sin tener en cuenta los retardos o los factores externos). Esto es muy útil para las aplicaciones time-sensitive<sup>13</sup>, como por ejemplo los grids<sup>14</sup>.
  - Las aplicaciones de gran escala y con gran cantidad de receptores, como la transmisión de video por Internet, son técnica y económicamente viables utilizando multicast.
  - La seguridad en las transmisiones multicast suele ser muy compleja. Sin embargo, como IPv6 implementa en forma nativa IPSEC, no tendría que haber problemas.

Las desventajas encontradas son:

- Al utilizar el mecanismo de transporte UDP, no se garantiza la entrega de paquete y el orden en que son recibidos.
- Requiere el soporte y la cooperación entre los routers para poder funcionar correctamente.

Existen distintos tipos de aplicaciones que pueden verse beneficiadas por multicast entre ellas: Acceso a bases de datos distribuidas; Distribución de software y de información; Servicios de tiempo; Servicios de resolución de nombre, como DNS; Replicación de base de datos; Video y audio streaming; Computación distribuida; Educación a distancia; Videoconferencias; Video bajo demanda.

---

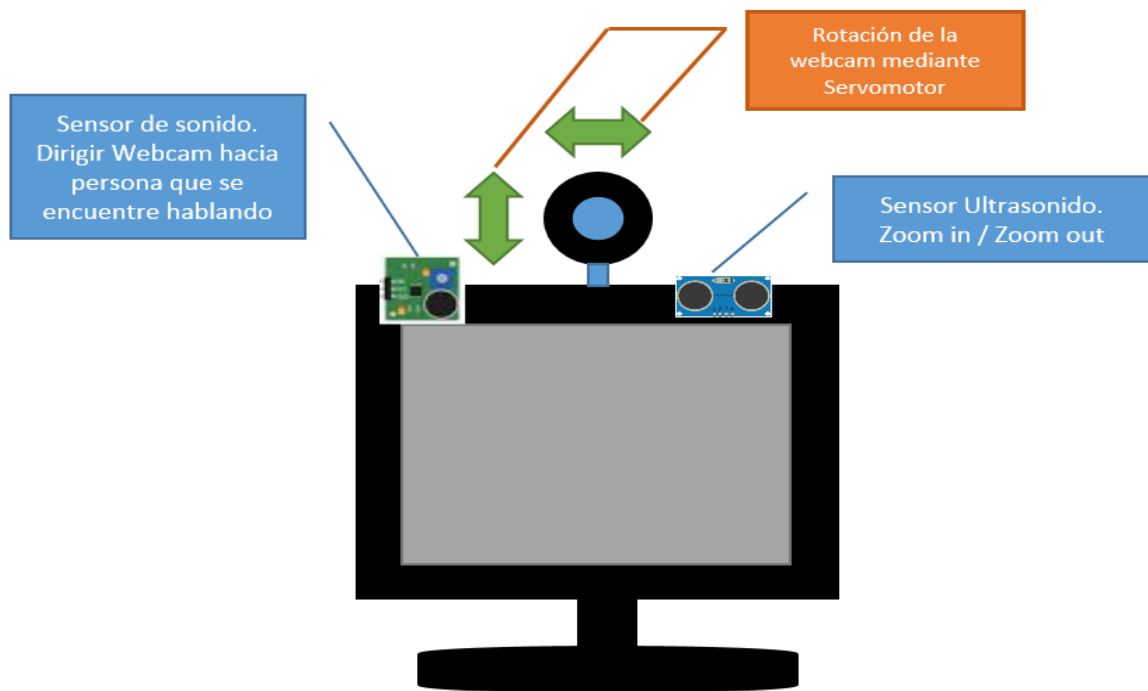
<sup>13</sup> Son aplicaciones en tiempo real, donde es importante que no existan retrasos en la comunicación.

<sup>14</sup> Son sistemas que permiten compartir recursos que se encuentran distribuidos por todo el mundo.

### 3.3.3. ETAPA 3: ENTORNO DE DESARROLLO

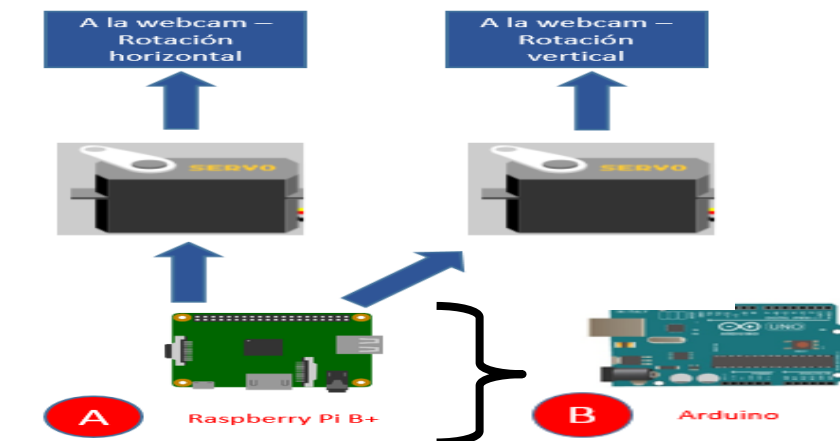
#### 3.3.3.1. Tarea 9: Análisis de Entorno de Desarrollo y Versionado

La elección del entorno de desarrollo estuvo condicionada en el hardware de trabajo para la generación del futuro prototipo. En cuanto a la solución de hardware, la propuesta diseñada prevé la motorización de una cámara, lo cual puede observarse en la **figura 12**.



**Figura 12.** Solución Propuesta Motorización de Cámara Web

Para implementar esta solución se requerirá de dos servomotores que puedan darle movilidad a la cámara tanto en forma horizontal como vertical y además sensores vinculados al sonido, que permitirán detectar de este modo al orador dentro de una sala y enfocarlo con la cámara en forma automática. La solución se puede implementar de dos formas: (a) Raspberry Pi B+; (b) Arduino, tal como se muestra en la **figura 13**.



**Figura 13.** Solución Posible que incluye la motorización de Cámara Web



La opción (a) consiste en una “minicomputadora” con sistema operativo basado en Linux, mientras que la opción (b) consiste en un microcontrolador.

Según la envergadura del proyecto y las necesidades particulares es posible seleccionar una u otra opción, si bien tienen algunas características que la diferencian ambas podrían cubrir las necesidades de este proyecto (algunos trabajos de referencia que pueden ser de interés son [BAL14], [GAR15], [VAL14]).

Para seleccionar una de las dos opciones se realizó una comparativa entre Arduino y Raspberry (ver **tabla 9**), seleccionándose finalmente a Raspberry. Lo cual trabajo aparejado la selección del lenguaje de programación.

**Tabla 9.** Comparativa entre Arduino y Raspberry

	<b>Raspberry Pi B+</b>	<b>Arduino UNO</b>
<b>RAM</b>	512 MB	2 KB
<b>Almacenamiento</b>	Micro SD	32 KB (flash), 1 KB (EEPROM)
<b>Procesador</b>	ARM1176	ATmega328
<b>Velocidad</b>	700 Mhz	16 Mhz
<b>Ethernet</b>	Sí	No
<b>Salida HDMI (audio y video)</b>	Sí	No
<b>Nº puertos USB</b>	4	1 (Conexión con PC)
<b>Soporta instalación de S.O.</b>	Sí (Linux)	No
<b>Cantidad de GPIO (puertos útiles)</b>	40	23
<b>Lenguajes de programación soportados</b>	Python, C, C++, Java, Perl, Ruby, etc	Arduino (basado en Wiring)
<b>Alimentación</b>	5v	5v
<b>Tamaño</b>	85 x 56 x 17mm	68.6 x 53.4 mm

Sobre Raspberry Pi (RPI) corren sistemas operativos basados en Linux. Existen varias distribuciones, la que fue desarrollada específicamente para la RPi se llama Raspbian. Con lo cual cualquier entorno de desarrollo (IDE) que corra bajo Linux se podría instalar y utilizar en la RPi. Pero dado que la RPi es un poco limitada en cuanto a recursos de hardware (procesador y memoria) comparada con una PC de escritorio, es recomendable utilizar IDEs que sean livianos, esto tiene la desventaja de que la mayoría no ofrece un set completo de funcionalidades como otros "IDEs de escritorio" como puede ser Visual Studio en Windows.

En la **tabla 10** se listan a modo de ejemplo algunos IDEs livianos que podrían funcionar bien en la RPi.



**Tabla 10.** Ejemplos de IDEs para utilizar en la Raspberry Pi

Nombre del IDE	Lenguaje que soporta
BlueJ	Java
DrJava	Java
IDLE	Python
Adafruit WebIDE	C, Python
Geany	C, C++, Java, Python
QtCreator	C++
CodeLite	C, C++, PHP

Para manejar dispositivos conectados a las RPi se usan bibliotecas. Existen bibliotecas para los lenguajes: C, C++, Python, Java.

La elección del lenguaje y del IDE va a depender del proyecto que se quiera realizar. En caso que se quiera realizar un proyecto de mayor envergadura y se requiera un IDE "potente", existe la posibilidad de programar en la PC de escritorio y compilar-ejecutar en la RPi. Esta técnica se conoce como "cross-compiling".

Para poder desarrollar con un IDE potente y por otra parte comenzar a analizar cuestiones de desarrollo sin contar con el equipamiento físico el cual se adquirirá a comienzo del segundo año del proyecto, se optó por utilizar la técnica "cross-compiling". El lenguaje elegido ha sido Java por tener buen soporte para hardware.

**Software Utilizado:** A continuación se detalla el software utilizado y sus versiones correspondientes. Se recomienda utilizar las mismas versiones en caso que se encuentren disponibles, o versiones posteriores en caso que las indicadas ya no estén disponibles.

- Sistema operativo host: Windows 7 64 bits.
- Sistema operativo guest: Ubuntu 14.04.1 LTS 32 bits.
- Software de virtualización: Oracle VM VirtualBox 4.3.28.100309.
- Toolchain: gcc-linaro-arm-linux-gnueabi-hf-raspbian.
- Java: JDK/JRE 8u45.
- Eclipse: Mars 4.5 32 bits.

Sistemas Operativos involucrados:

- Sistema operativo host: es el SO que se encuentra instalado en forma nativa en la PC. La máquina utilizada para desarrollar este instructivo tiene Windows 7 64 bits, pero no es necesario que sea esa versión e incluso podría ser otro SO que no sea Windows (Mac OS X, por ejemplo), ya que el VirtualBox puede ser instalado en una gran cantidad de sistemas operativos. Obviamente si se cuenta con una PC con cualquier distribución de Linux, no

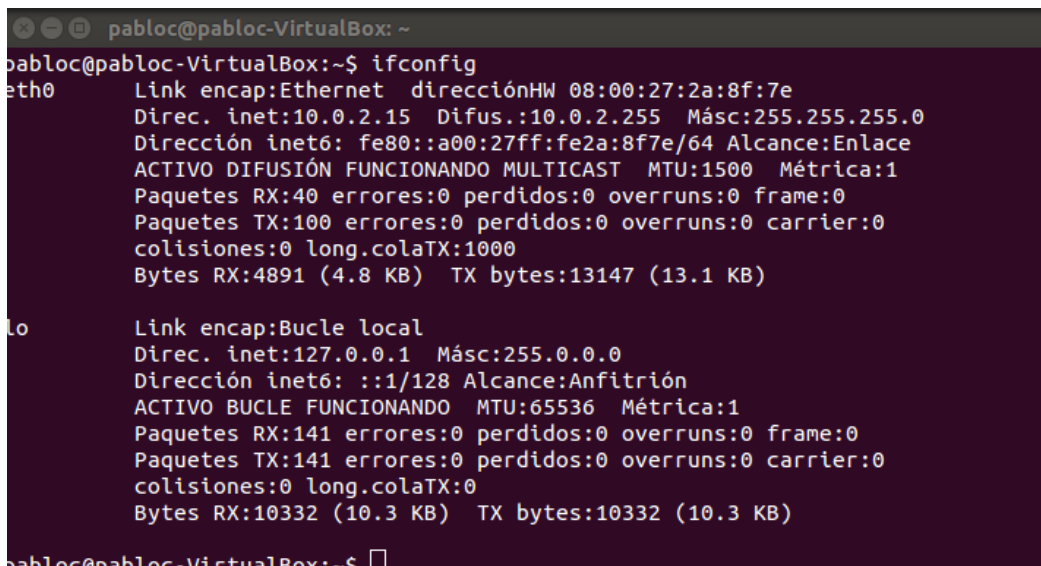
es necesaria la instalación del VirtualBox y directamente se puede pasar a la sección de instalación del entorno de desarrollo.

- Sistema operativo guest: es el SO que se instala en la máquina virtual. A pesar de que este instructivo fue desarrollado en 2015, se optó por la versión 14.04 LTS de Ubuntu ya que tiene soporte extendido por 5 años, a diferencia de los 6 meses de las versiones no-LTS. Se optó por 32 bits, pero se podría haber elegido la versión de 64 bits, aunque habría que verificar que las herramientas a instalar después (Eclipse, toolchain, etc.) sean compatibles con 64 bits. No va a haber grandes diferencias de rendimiento salvo que se le asignen 4 GB o más de memoria RAM a la máquina virtual, en cuyo es conveniente utilizar la versión 64 bits de Ubuntu.

En el **anexo A** se presenta el instructivo realizado para la instalación del entorno de desarrollo para la Raspberry PI.

### 3.3.3.2. Tarea 10: “Manejo de IPv6 en el entorno de Desarrollo”

Se habilitó IPv6 en la máquina virtual de linux con el comando: **\$ sudo modprobe ipv6** quedando configurada la dirección IPv6 tal como puede observarse en la **figura 14**.



```
pabloc@pabloc-VirtualBox: ~  
pabloc@pabloc-VirtualBox:~$ ifconfig  
eth0      Link encap:Ethernet direcciónHW 08:00:27:2a:8f:7e  
          Direc. inet:10.0.2.15 Difus.:10.0.2.255 Másc:255.255.255.0  
          Dirección inet6: fe80::a00:27ff:fe2a:8f7e/64 Alcance:Enlace  
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1  
          Paquetes RX:40 errores:0 perdidos:0 overruns:0 frame:0  
          Paquetes TX:100 errores:0 perdidos:0 overruns:0 carrier:0  
          colisiones:0 long.colaTX:1000  
          Bytes RX:4891 (4.8 KB) TX bytes:13147 (13.1 KB)  
  
lo        Link encap:Bucle local  
          Direc. inet:127.0.0.1 Másc:255.0.0.0  
          Dirección inet6: ::1/128 Alcance:Anfitrión  
          ACTIVO BUCLE FUNCIONANDO MTU:65536 Métrica:1  
          Paquetes RX:141 errores:0 perdidos:0 overruns:0 frame:0  
          Paquetes TX:141 errores:0 perdidos:0 overruns:0 carrier:0  
          colisiones:0 long.colaTX:0  
          Bytes RX:10332 (10.3 KB) TX bytes:10332 (10.3 KB)  
pabloc@pabloc-VirtualBox:~$
```

**Figura 14.** Configuración de IPv6

Se realizaron algunas pruebas las que derivaron a la instalación de una biblioteca open-source en java denominada jNetPcap<sup>15</sup> que tiene soporte para IPv6.

<sup>15</sup> La página con documentación sobre esta biblioteca puede encontrarse en: <http://jnetpcap.com/>

### 3.3.3.3. Tarea 11: “Manejo de Sensores”

La Raspberry Pi incorpora un puerto denominado GPIO (General Purpose input/out) especialmente diseñado para operaciones de entrada salida de propósito general. Este puerto tiene distintos pines que pueden utilizarse para controlar fácilmente tanto sensores como actuadores (ver **figura 15**). Además existen distintas bibliotecas de programación que facilitan la comunicación con el hardware, por ejemplo si se selecciona Java como lenguaje de programación, puede utilizarse la api Device I/O [CRU15]. Esta biblioteca simplifica el manejo del puerto GPIO de las raspberry Pi.

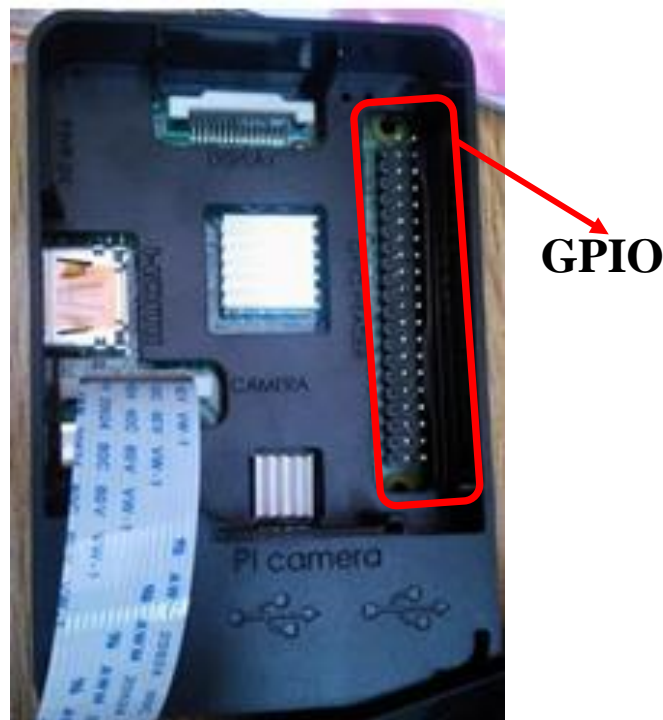


Figura 15. Puerto GPIO

### 3.3.4. ETAPA 4: ANÁLISIS DE QoS BASADO EN AUDIO/VIDEO STREAMING EN TIEMPO REAL

El protocolo de IPv6 ya trae integrado en su núcleo la Calidad de Servicio (QoS), esto ha sido explicado previamente en este informe (tarea 7 y tarea 8 resumidas previamente en este informe). Algunos autores como [SAL11] abordan la temática de la QoS en streaming en tiempo real. No obstante es un tema importante atender la QoS en redes multicast, para lo cual principalmente se consideró el capítulo 6 de la tesis de maestría de [PER05] que está destinada a analizar profundamente este tema.

La Calidad de Servicio (QoS), provee a la comunicación de métodos para contrarrestar los problemas que afectan a la transmisión como escaso ancho de banda, latencia o retardo (delay), jitter y pérdida de paquetes (packet loss).

A su vez, las distintas aplicaciones muestran distinto nivel de tolerancia a errores en la transmisión en general. La **Tabla 1** muestra en qué grado afectan a distintos tipos de aplicaciones, los problemas de transmisión [LIAnE]. Esta característica denominada tolerancia al error, posee distintos grados. El más leve es Tolerante, donde la aplicación se ve poco perjudicada; Sensible, correspondiente a una aplicación que se ve muy perjudicada por los problemas e Intolerante, máxima graduación, donde es imposible reconocer la transmisión.

**Tabla 1: Aplicaciones y Problemas en la transmisión**

APLICACIÓN	PROBLEMA				
	Ancho de banda	Latencia Delay	Jitter	Pérdida de Paquetes Packet loss	Tolerancia al error
<b>Voip</b>	Variación baja	Requerimientos estrictos	Sensible	Muy sensible	Sensible
<b>Streaming de video y audio</b>	Variación media y baja	Requerimientos Elásticos	Requerimientos Elásticos	Muy sensible	Sensible
<b>Juegos interactivos</b>	Variación media	Intolerante	Sensible	Muy sensible	Intolerante
<b>Señalización Tele-inmersión</b>	Variación media y baja	Requerimientos altos	Sensible	Muy sensible	Sensible
<b>Video-conferencia Distribución de video de alta calidad</b>	Alta variación	Requerimientos estrictos	Sensible	Sensible	Sensible
<b>Aplicaciones interactivas transaccionales. (ej: e-commerce)</b>	Variación media	Requerimientos altos	Poco sensible	Poco sensible	Sensible
<b>Aplicaciones IP elásticas</b>	Alta variación	Tolerante	Tolerante	Poco sensible	Tolerante
<b>Entornos colaborativos</b>	Alta variación	Tolerante	Tolerante	Tolerante	Tolerante

Como es posible observar en la **Tabla 1**, la mayoría de las aplicaciones son Tolerantes o Sensibles a los problemas. Sólo los Juegos Interactivos se presenta como Intolerante.

Dentro de los problemas de transmisión, el ancho de banda ocupa un lugar prominente, por lo tanto se han buscado soluciones para paliar los problemas que emergen cuando, la demanda supera al ancho de banda produciéndose “congestiones”.

Dentro de las posibles soluciones, se encuentra el uso de colas de diferentes tipos para organizar la congestión y proporcionar a cada transmisión la oportunidad de ser enviada.

Tipos de encolamiento (queuing):

1. FIFO (Fist In First Out. (Primero que Entra Primero que Sale)
2. Cola de Prioridad (PQ. Priority Queuing)
3. Cola Personalizada (CQ. Custom Queuing))
4. Weighed Fair Queuing (WFQ)
5. Modified Deficit Round Robin

Estos tipos se detallan a continuación:

- 1) FIFO (Fist In First Out) (PEPS: Primero que Entra Primero que Sale):** Este método almacena los paquetes salientes dentro de un buffer en el orden que van llegando. Cuando la interfaz de transmisión puede enviarlos los va enviando en el mismo orden en que fueron llegando al buffer, como muestra la **figura 1**, la cual ha sido extraída de [HOH13].

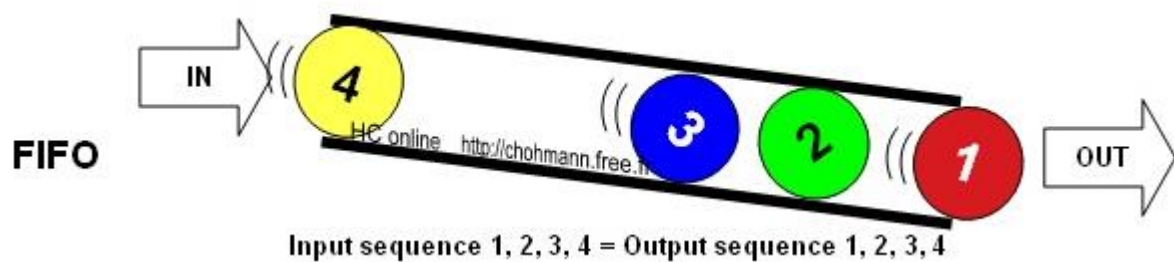


Fig 1: Cola. FIFO. Fist In First Out.

- 2) Cola de Prioridad (Priority Queuing. PQ):** Este método recibe paquetes identificados según su prioridad (alta, media, normal y baja), separándolos en 4 colas, una para cada nivel de prioridad. Estas 4 colas alimentan un buffer de la interfaz de transmisión. Se envían primero los paquetes de la cola de alta prioridad, luego los de media, etc. La **figura 2** muestra un esquema de

funcionamiento de este método, extraído de [CIS07]. Este método, mostrado en **figura 2** tiene como inconveniente la falta total de oportunidad de transmisión de las colas de baja prioridad.

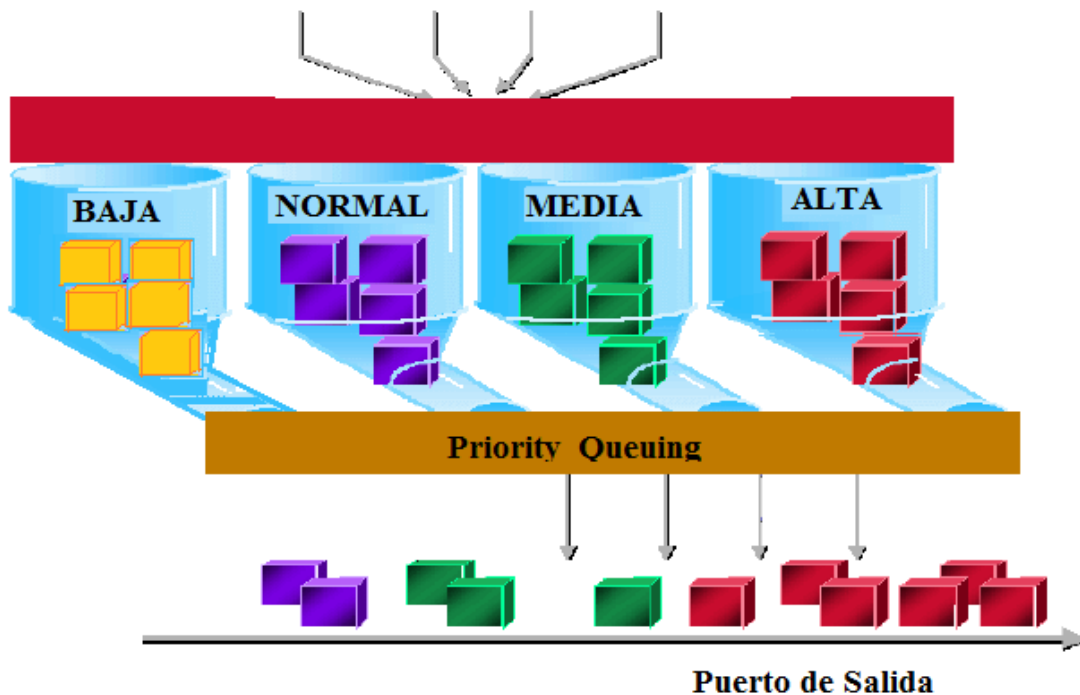


Fig. 2: Cola de Prioridad. (Priority Queuing).

- 3) **Cola Personalizada (Custom Queuing. CQ):** Las Colas de Prioridad presentan el inconveniente de que los paquetes de baja prioridad, tal vez nunca sean enviados. Las Colas Personalizadas (Custom Queuing) son una posible respuesta a ese problema. Este método trata de compartir el ancho de banda entre paquetes utilizando la visión máximo-mínimo. Se reserva un porcentaje del ancho de banda para cada tipo de tráfico. Cada clase de paquete obtiene la parte proporcional a su peso del ancho de banda más la posibilidad de reclamar el ancho de banda de la interfaz que no está siendo utilizado. Consiste en 16 colas estáticas con criterio de clasificación configurable. Este método le asigna un contador de bytes a cada cola con un valor de 1500 bytes por defecto y sirve las colas con el método de Round Robin, proporcional a los contadores. Cada paquete enviado, decrementa el contador de bytes, hasta llegar a cero. El protocolo APPN posee el concepto de Clase de Servicio (CoS) que determina la prioridad de transmisión de cada mensaje. La **figura 3** muestra un ejemplo de este método y fue extraída de [CIS09]. Como puede observarse en la **figura 3**, cada tipo de tráfico es enviado, aún los de baja prioridad.

Una de sus ventajas es todas las colas serán enviadas, aún aquellas de la más baja prioridad. Si bien, no asegura una prioridad absoluta, asegura una porción fija del ancho de banda en lugares de congestión y proporciona los recursos disponibles al resto del tráfico.

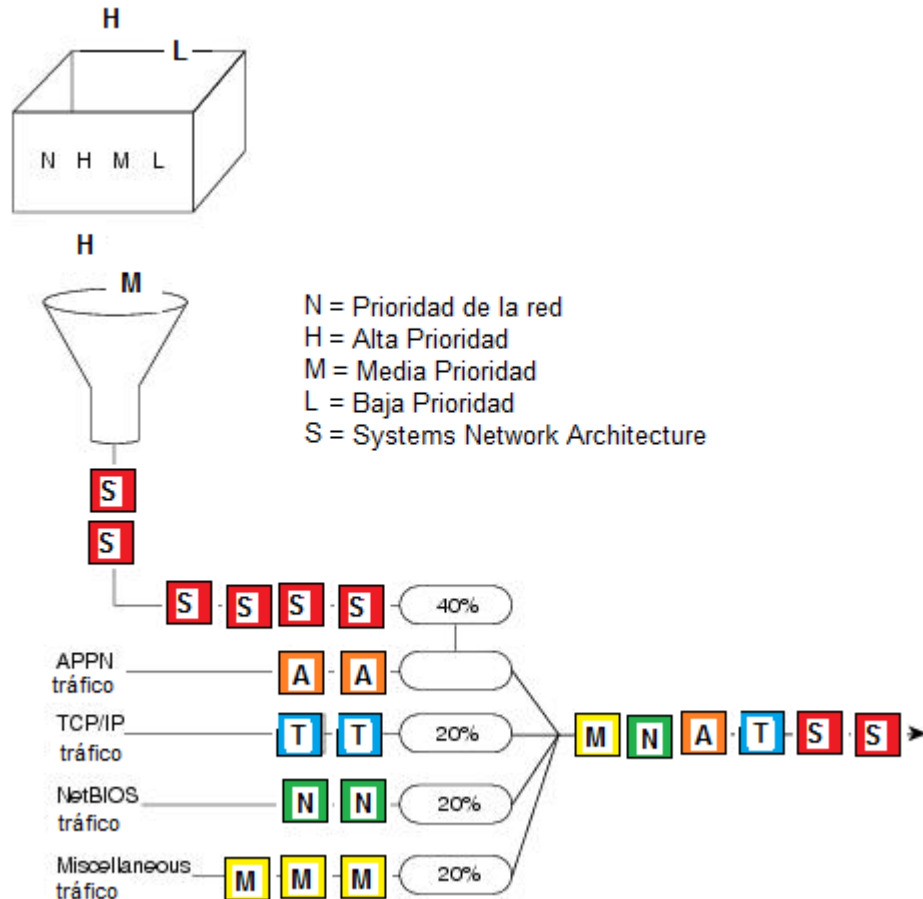


Fig. 3: Cola Personalizada (Custom Queuing).

**4) Espera equitativa ponderada. Weighted Fair Queuing (WFQ):** Este método asigna a cada paquete un peso o ponderación que determina su número de orden dentro de la cola de paquetes. El peso o ponderación se asigna de acuerdo a discriminadores de la suite de protocolos TCP/IP (Dirección de origen y destino y tipo de protocolo en IP, número de socket (puerto de TCP/UDP) o ToS en IP.

De acuerdo a estos discriminadores, se crea una cola distinta para cada flujo de datos activos, utilizándose un valor predeterminado para la profundidad de la cola. La **figura 4** muestra el funcionamiento de este método y fue extraída de [KUR99].

Este método, esquematizado en la **figura 4**, tiene la característica de que dado que cada flujo de datos tiene su propia cola, si un flujo ha enviado los paquetes

más grandes o más paquetes por segundo que los otros flujos desde que está activo, sólo se perjudicará a sí mismo y no a otras sesiones.

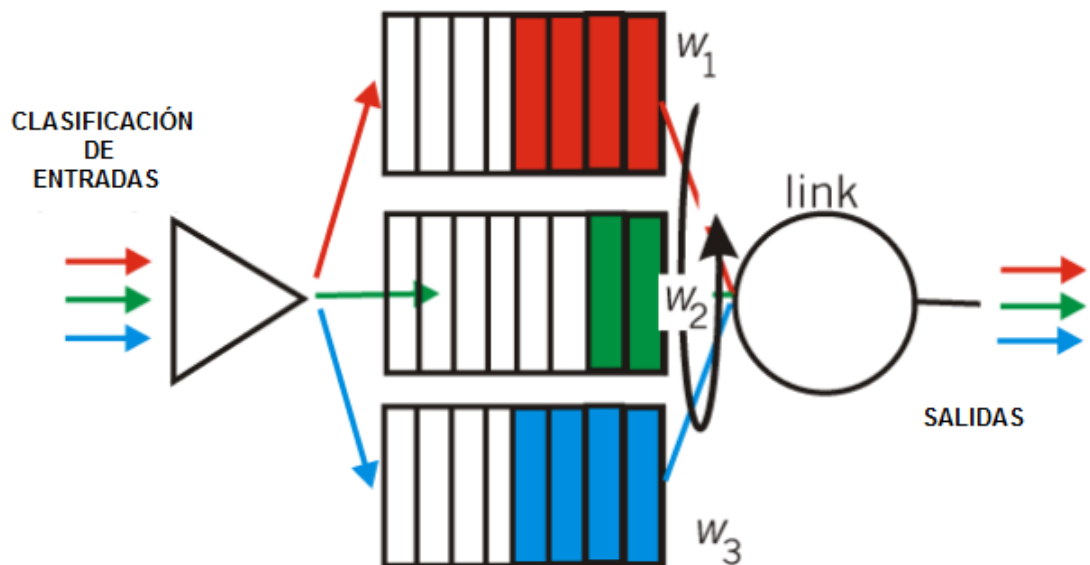


Fig 4: Espera equitativa ponderada. Weighted Fair Queuing (WFQ)

**5) Modified Deficit Round Robin (MDRR):** Es una técnica utilizada para la gestión de congestión. Con MDRR cada cola no vacía es servida una tras otra con el método Round Robin. Este método de planificación consiste en un algoritmo donde cada flujo se le asigna una porción de tiempo equitativa y ordenada, tratando a todos los flujos con la misma prioridad.

Cada vez que es servida una cola se desencola una cantidad fija de datos. El algoritmo sigue sirviendo la siguiente cola.

Se trabaja con dos variables asociadas a cada cola:

- (1) Quantum: Cantidad de bytes que una cola puede servir en cada vuelta.
- (2) Contador de Déficit (Deficit Counter): Cantidad de bytes servidos por una cola en cada vuelta. El Contador de Déficit se inicializa con el valor del Quantum. Mientras los paquetes de una cola son transmitidos, sus largos se restan del Contador de Déficit. Cuando el Contador de Déficit llega a 0 (cero), la transmisión se pasa a la siguiente cola. Si la cola anterior no se vació, su contador de Déficit se incrementa nuevamente con el Quantum.

La **figura 5** muestra este método y fue extraída de [SHA15]



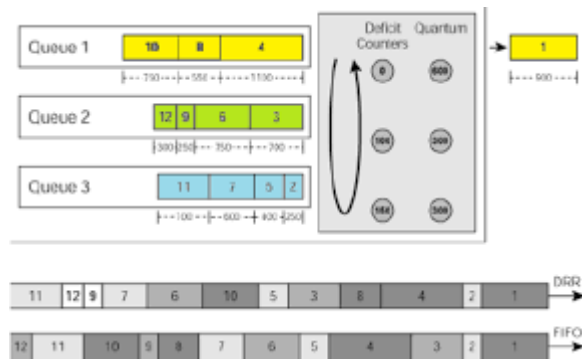


Fig. 5: Modified Deficit Round Robin.

La **figura 5** permite observar que MDRR registra la cantidad de datos servidos en exceso por cada cola en el Contador de Déficit (Deficit Counter) , de acuerdo a un valor preconfigurado denominado Quantum.

En el siguiente paso, cuando la cola es nuevamente servida, se desencolará menos datos para compensar el exceso de datos servidos en la vuelta anterior.

Como resultado, el promedio de datos desencolados por cola, será cercano al valor configurado.

Adicionalmente MDRR mantiene una cola prioritaria que es servida en forma preferencial.

**3.3.5. ETAPA 5 – EVALUACIÓN DE IMPLEMENTACIÓN DE HARDWARE**

En base a lo definido en la ETAPA 3, la elección final es comprar una Raspberry PI, la cual en forma autónoma conectada a un monitor o televisor permita utilizar la solución integral de video teleconferencia.

Al momento de planificar el presente proyecto se había considerado comprar dos Raspberry PI B+ existiendo actualmente un modelo nuevo denominado Raspberry PI 2 B. Lo cual se denominará a continuación por simplicidad Raspberry 1 y Raspberry 2. Las diferencias entre ambas soluciones radican en el hardware en cuanto a memoria RAM y procesador (ver **tabla 11**). Tomándose la primera decisión en cuanto a hardware que es comprar la Raspberry 2, a pesar de que ello implique un incremento de dinero.

**Tabla 11.** Características del modelo previo y actual de la Raspberry PI

	Memoria RAM	Procesador
Raspberry 1	512 MB	700 MHz
Raspberry 2	1 GB	900 MHz QUAD CORE (ARM cortex A7)

Estaba previsto en los fondos asignados al proyecto la compra de 2 Raspberry PI 1 + Kit, para contar con el equipamiento en el segundo año del proyecto. Para poder hacer pruebas se compraron Raspberry PI 2 + Kit (ver **Figura 16**) lo cual permitió comenzar a familiarizarse con el equipamiento. Surgió la necesidad de pensar no sólo en un emisor y receptor representado cada uno de ellos mediante una Raspberry sino también poder realizar a futuro pruebas de multicast sobre ipv6, para lo cual dos nodos no eran suficientes motivo por el cual se invirtió en una tercer Raspberry.

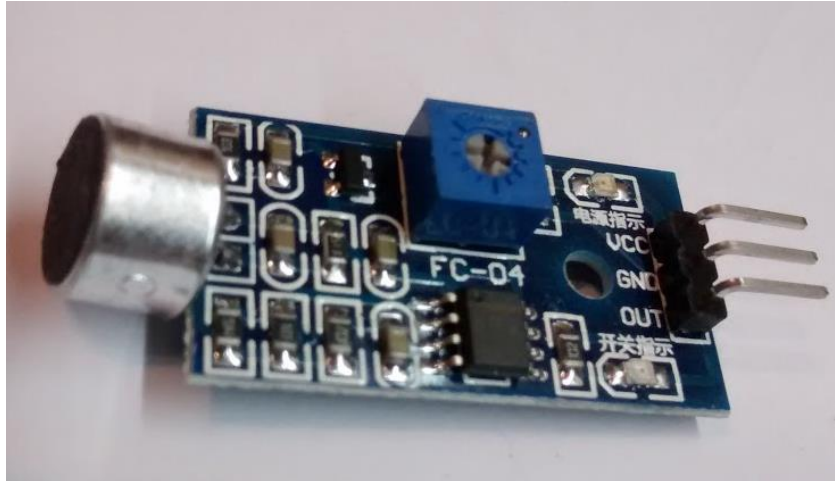


**Figura 16.** KIT Raspberry PI 2

### 3.3.5.1. Tarea 12: Análisis de los sensores

En uno de los prototipos se incorporarán 2 sensores con el objetivo de poder desplazar la cámara automáticamente en la dirección en la que se encuentra la fuente que emite el sonido. De este modo un orador podrá desplazarse por la sala y la cámara se desplazará también acompañando sus explicaciones. Esto permitirá también poder manipular elementos y que la persona que esté ofreciendo una video-conferencia pueda libremente moverse sin tomar en cuenta si está dentro del área de visión de la cámara. Con este fin se incorpora el sensor de sonido y ultrasonido.

El sensor de sonido posee sensibilidad ajustable, su respuesta será mediante dos niveles lógicos nivel bajo o alto (0 ó 1), que permitirá indicar si hay sonido detectable o no. En la **figura 17** se muestra el sensor de Ultrasonido.



**Figura 17.** Sensor de Sonido

El sensor de ultrasonido HC-SR04 para Raspberry Pi (ver **figura 18**) permite medir distancias. “Funciona enviando un pulso de ultrasonidos (inaudible para el oído humano por su alta frecuencia) a través de uno de los cilindros que componen el sensor y esperando a que dicho sonido rebote sobre un objeto y vuelva. El retorno es captado por el otro cilindro del sensor... Sabemos que la velocidad del sonido en el aire es 340 m/segundo así que calculando el tiempo transcurrido entre el envío del pulso y la recepción de la señal de retorno y luego aplicando una sencilla fórmula matemática obtenemos la distancia entre el sensor y el objeto que hay delante”<sup>16</sup>. También es posible realizar 2 mediciones consecutivas y de este modo se podría detectar si el objeto en cuestión está en movimiento y la velocidad de dicho movimiento.



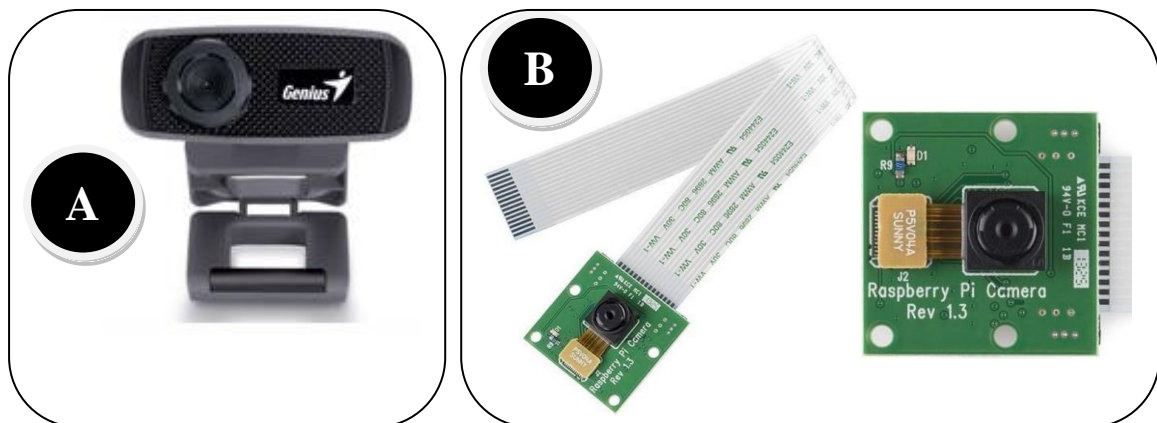
**Figura 18.** Sensor de UltraSonido HC-SR04 para Raspberry Pi

Se han adquirido ambos sensores para realizar una solución motorizada, lo cual está vinculado con la tarea que sigue a continuación en este informe.

<sup>16</sup> Frase tomada de: <http://fpaez.com/sensor-ultrasonico-hc-sr04-para-raspberry-pi/>

### 3.3.5.2. Tarea 13: Evaluación Integración de un controlador de motorización para cámara USB

Existe la posibilidad de conectar una cámara web estándar, por USB, contándose actualmente en el laboratorio de investigación con dos webcam modelo: GENIUS FACECAM 1000x HD como puede verse en la parte A de la **figura 19**; siendo una segunda opción conectar directamente a la Raspberry un módulo conteniendo una cámara web lo cual puede verse en la parte B de la **figura 19**.



**Figura 19.** Cámara Web y Cámara Integrada

La segunda opción indicada como B en la **figura 19** tiene un peso de 3 gramos, siendo una placa muy pequeña (25x20x9 milímetros) para generando un prototipo más reducido. La conexión en este caso es directamente a la Raspberry Pi al SCI<sup>17</sup> por medio del cable ribbon. Este módulo tiene un costo aproximado de 79 dólares<sup>18</sup>. La elección de la solución impactará en como conectar los servomotores horizontales y verticales para poder motorizar la cámara (lo cual ha sido planteado previamente en este informe considerándose inicialmente la opción A en la Etapa 3 Tarea 9 del presente informe). De todos modos, ambas soluciones son viables.

Como se mencionó anteriormente, existen dos opciones de cámara para conectar a la Raspberry Pi: (A) a través del puerto USB ó (B) usando el puerto SCI de la Raspberry Pi. En el caso de la cámara USB, la ventaja que ofrece es la de poder programar una aplicación desde la PC de escritorio utilizando la cámara, y luego conectar esa cámara y llevar la aplicación a la Raspberry sin tener que realizar modificaciones. Mientras que la ventaja de la cámara integrada mediante el puerto SCI es la velocidad y la calidad de captura de imagen y video.

<sup>17</sup> SCI (Camera Serial Interfaz)

<sup>18</sup>Link de Referencia: <http://www.ocstore.com.ar/productos.asp?opt=1>

Habiéndose elegido la opción B se muestra en la **figura 20** la Raspberry con la cámara conectada.



**Figura 20.** Cámara por puerto SCI

La Raspberry trae un logo perforado, al conectar la cámara a través de él la lente queda descubierta de forma de poder tomar imágenes o grabar sin dificultad alguna.

Una vez instalada la cámara por medio de línea de comandos, se ejecutó para testear la cámara, se tipea: **raspivid -o vid.h264**. Esta línea tiene el parámetro **-o** que indica la salida a un archivo que es el que se indica a continuación de dicho parámetro **vid.h264**, esto permite grabar un video durante unos segundos. Utilizando el parámetro **-t** se puede definir el tiempo de duración del video.

El entorno de programación configurado en el laboratorio se encuentra instalado sobre una máquina virtual. El software de virtualización utilizado, VirtualBox, ofrece la posibilidad de conectar periféricos USB a la máquina virtual y ser utilizados desde el sistema operativo guest (Ubuntu en este caso). Esto nos permitirá desarrollar una aplicación que utilice la cámara desde el entorno de programación que habíamos configurado.

Se realizó la búsqueda de una API que nos permita capturar imágenes fácilmente desde código Java, sin tener que tener en cuenta detalles de la cámara utilizada. Y que

además sea multiplataforma, para poder utilizarla en una PC de escritorio o en la Raspberry Pi.

La API elegida se denomina “Webcam Capture API<sup>19</sup>”. Es una biblioteca que permite utilizar cámaras web integradas al equipo (como en el caso de una notebook) o externas (como es nuestro caso) directamente desde Java. La API presenta un diseño simple, que nos permite hacer uso de las diferentes características de la cámara, abstrayéndonos de los detalles de implementación y además soporta múltiples frameworks de captura de imágenes.

Las características de la API:

- Es simple y se puede utilizar en un entorno multi-threading.
- No se requiere de software adicional.
- Soporta múltiples plataformas (Windows, Linux, Mac OS) y diferentes arquitecturas (32-bit, 64-bit, ARM). Esto es muy importante para nosotros ya que la Raspberry tiene un procesador de arquitectura ARM.
- Permite capturar imágenes desde cámaras USB o cámaras integradas.
- Los archivos que componen la biblioteca se encuentran almacenados en el repositorio central de Maven<sup>20</sup>, lo que hace más fácil la utilización por parte del programador. O también se pueden descargar como archivo ZIP.
- Ofrece componentes gráficos para mostrar el video de la cámara.
- Soporta los siguientes frameworks de captura: OpenIMAJ, LTI CIVIL, Java Media Framework (JMF), Freedom for Media in Java (FMJ), OpenCV, VLC, GStreamer (solo 0.10.x).

A continuación se presenta un código de ejemplo que demuestra cómo se utiliza la API para capturar una imagen con la cámara web. Primero se debe definir algunas variables que representan el tamaño de la captura a realizar y el tipo y nombre de la imagen que se va a capturar (ver **figura 21**).

```
final int anchoImagen = 1280;  
final int altoImagen = 720;  
final String tipoFoto = "JPG";  
final String nombreFoto = "fotoEjemplo."+tipoFoto.toLowerCase();
```

**Figura 21.** Variables para el tamaño de la captura, nombre y tipo de imágenes

<sup>19</sup> Web oficial: <http://www.webcam-capture.sarxos.pl>

<sup>20</sup> Maven es un gestor de dependencias, que se encarga de resolver las dependencias de un proyecto, que pueden ser bibliotecas u otros módulos de software que son utilizados dentro de dicho proyecto.

La API permite detectar que cámaras se encuentran conectadas al equipo, y seleccionar cual utilizar. Para ello se almacenan en una lista las cámaras detectadas por la API (ver **figura 22**).

```
List<Webcam> listaCamaras = new ArrayList<Webcam>();  
Webcam camaraSeleccionada = null;  
  
System.out.println("Obteniendo lista de cámaras...");  
listaCamaras = Webcam.getWebcams();  
for (Webcam camara : listaCamaras){  
    System.out.println("Cámara detectada:\n" + camara.getName());  
}
```

**Figura 22.** Lista de Cámara Detectadas por la API

En nuestro caso seleccionamos la primera cámara de la lista, que es la única que tenemos conectada a la PC (ver **figura 23**). Luego lo que se debe hacer es configurar el tamaño de la captura, en este caso utilizamos la máxima resolución que soporta la cámara web, resolución 1280x720 (ver **figura 24**).

```
System.out.println("Seleccionando cámara...");  
if(listaCamaras.isEmpty()){  
    System.out.println("No hay cámaras conectadas");  
}  
else{  
    camaraSeleccionada = listaCamaras.get(0);  
}
```

**Figura 23.** Selección de la cámara conectada

```
System.out.println("Seteando tamaño de captura a: " + anchoImagen + " x " + altoImagen);  
Dimension[] resNoEstandar = new Dimension[] {  
    WebcamResolution.HD720.getSize(),  
    new Dimension(anchoImagen, altoImagen),  
};  
  
camaraSeleccionada.setCustomViewSizes(resNoEstandar);  
camaraSeleccionada.setViewSize(WebcamResolution.HD720.getSize());
```

**Figura 24.** Selección de la cámara conectada

Por último se abre la cámara seleccionada y se toma la foto (ver **figura 25**).

```
System.out.println("Abriendo cámara...");
camaraSeleccionada.open();
.....
System.out.println("Capturando foto...");
try {
    BufferedImage imagen = camaraSeleccionada.getImage();
    ImageIO.write(imagen, tipoFoto, new File(nombreFoto));
    System.out.println("Imagen OK: " + nombreFoto);
} catch (IOException e) {
    e.printStackTrace();
    System.out.println("Error al capturar imagen");
}
```

**Figura 25.** Se abre la cámara y se toma la foto

Este es un ejemplo sencillo, sin interfaz gráfica, pero permite ver lo fácil que es utilizar la API elegida para capturar imágenes.

También se están analizando una API que permite el manejo de la cámara integrada que es la opción por la que finalmente se ha optado. En la **figura 26** se muestra la solución que permitirá empotrar el gabinete de la raspberry el cual tiene la cámara integrada mostrando previamente en la **figura 20** y por medio de 2 servomotores uno horizontal y otro vertical podrá moverse según lo sensado en cuanto a sonido (sensor de sonido y ultrasonido – mostrados previamente en la **figura 17** y la **figura 18**) en base a la ubicación del orador.



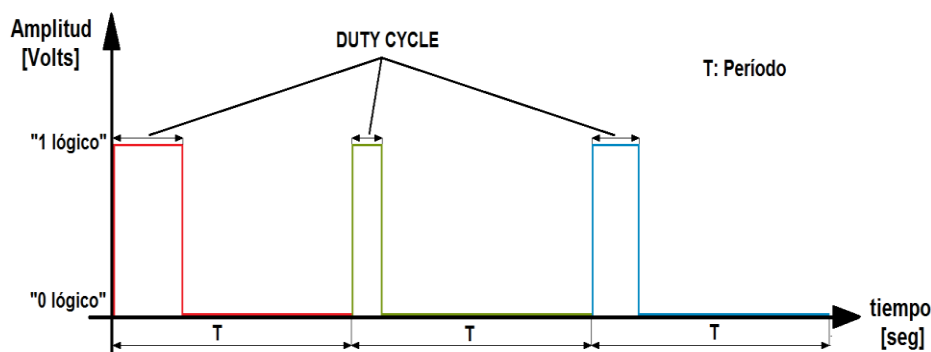
**Figura 26.** Solución con Servomotores

Los servos son motores de corriente continua (CC), pero no pueden lograr un giro continuo de 360°, están preparados para moverse a un ángulo fijo en respuesta a una señal de control, y mantenerse fijos en dicha posición. “Un servo principalmente está formado por un conjunto reductor (engranajes), un motor de CC y por último por un circuito



de control, aunque en la práctica se comporta como un bloque funcional que posiciona su eje en un ángulo preciso en función de la señal de control” [DEL16].

Normalmente estos pequeños servos funcionan con una tensión de alimentación de 5V y a su vez la señal digital de control puede ser de un nivel de tensión entre 3V y 7,2V, pero el valor de voltaje recomendable para evitar errores de detección del “1 lógico” es de 5V. El control se realiza mediante una señal modulada por ancho de pulso, en la que el ancho el pulso indica el ángulo que se desea que adopte el eje. La modulación por ancho de pulso, PWM (Pulse Width Modulation), es uno de los sistemas más empleados para el control de motores. Este sistema consiste en generar una onda cuadrada en la que se varía el tiempo que el pulso está en el nivel alto, manteniendo el mismo período (normalmente), con el objetivo de modificar la posición del servo según se desee (ver **figura 27**).

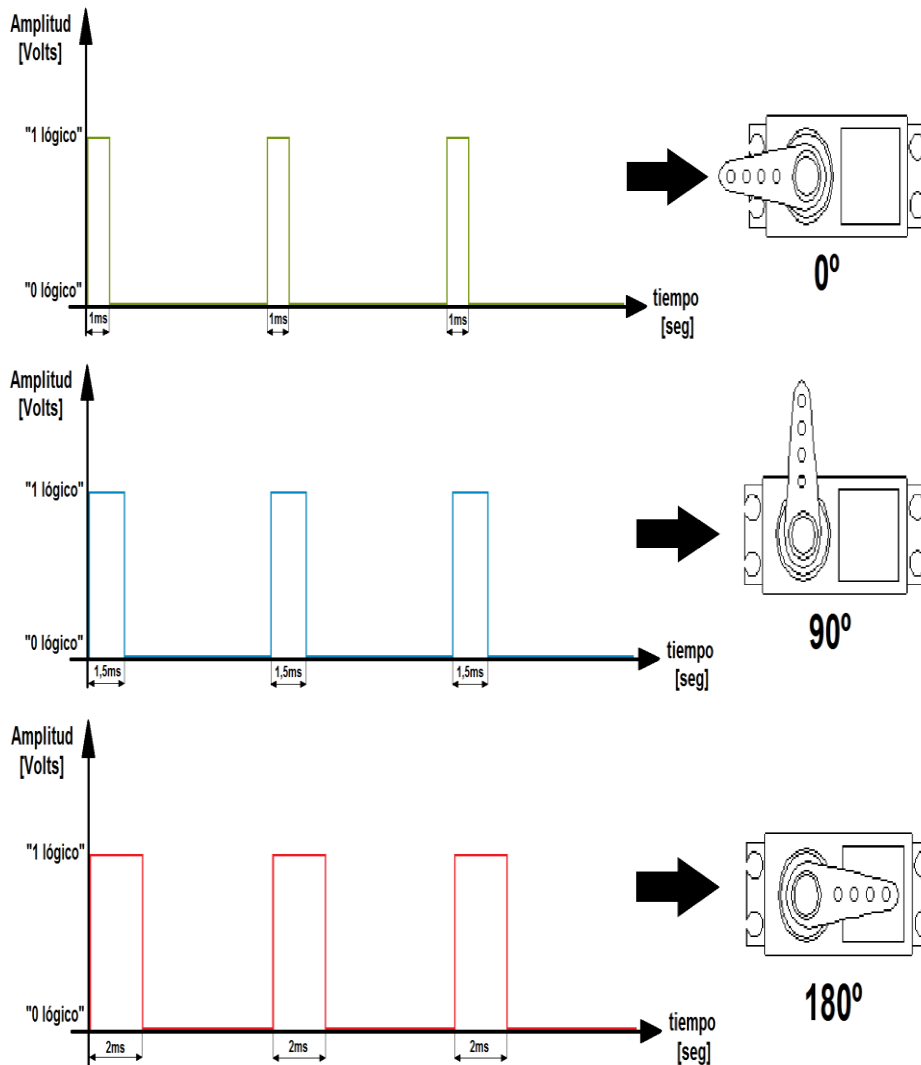


**Figura 27:** Ejemplo de modulación por ancho de pulso (PWM)

Por lo tanto, se enviará un tren de pulsos por la entrada de control del servomotor tal que irá variando la duración (anchura) del pulso activo en función al ángulo de giro deseado del motor. Los pulsos donde la señal de PWM se mantiene en “1 lógico” se denomina “Duty Cycle”.

Se necesitan valores específicos de frecuencia en la señal de control. Los valores más generales se corresponden con períodos de la señal cada 20 ms, y pulsos en nivel alto de entre 1 ms y 2 ms de anchura. En otras palabras, se trataría de una señal de 50Hz de frecuencia con un duty cycle entre el 5% y el 10% del período.

Habitualmente cada servo tiene sus márgenes de operación, que se corresponden con el ancho del pulso máximo y mínimo con el que el mismo funciona. Mayormente pueden moverse entre un ángulo de 0° y 180°, aunque existen algunos modelos comerciales que permiten un giro de 360°. En el caso del modelo SG90 permite un ángulo de giro de 180°, con lo cual con un duty cycle de 1.5 ms indicaría la posición central o neutra (90°), con otro de 2 ms giraría 180°, con otro de 1 ms se movería a la posición de 0°, y mientras que otros valores del pulso lo dejarían en posiciones intermedias. En la **figura 28**, se demuestra gráficamente lo explicado.



**Figura 28:** Ángulos de Giro del ServoMotor

Estos valores suelen ser los recomendados, sin embargo, es posible emplear pulsos menores de 1 ms o mayores de 2 ms, para poder conseguir ángulos mayores de 180°. Pero en la práctica esto no se corresponde con la teoría, ya que existen límites mecánicos propios del servomotor, que si se sobrepasan el servo comenzará a emitir un zumbido ya que estará ejerciendo una fuerza para intentar girar un ángulo mayor a 180° que el factor limitante mecánico no se lo permitirá. Es por ello que se debe evitar enviar señales de control de estas características para evitar problemas tales como: pérdida de precisión de giro o desgaste mecánico producido más rápidamente de lo estimado.

La duración entre pulso y pulso en que la señal de PWM se encuentra en nivel bajo ("0 lógico") no es crítica, e incluso puede ser distinto entre uno y otro período de la señal. Generalmente se suelen emplear valores de período de la señal de aproximadamente 20 ms (entre 10 ms y 30 ms). Si el tiempo de duración del pulso en nivel bajo es inferior al



mínimo, puede interferir con la temporización interna del servo, causando un zumbido, y la vibración del eje de salida. Si es mayor que el máximo, entonces el servo pasará a estado dormido entre pulsos. Esto provoca que se mueva con intervalos pequeños.

Es importante destacar que para que un servo se mantenga en la misma posición durante un cierto tiempo, es necesario enviarle continuamente el pulso correspondiente. De este modo, si existe alguna fuerza que le obligue a abandonar dicha posición, este intentará resistirse. De igual manera, si se deja de enviar pulsos (o el intervalo entre pulsos es mayor que el máximo) entonces el servo perderá fuerza y dejará de intentar mantener su posición, de modo que cualquier fuerza externa podría desplazarlo.

Un servo tiene un conector de 3 hilos, Alimentación de 5V (rojo), GND (negro o marrón) y el otro Control (amarillo o blanco). Además como se mencionó anteriormente funcionan con niveles de tensión de la señal de control entre 3V y 7,2V, por lo tanto esto nos permite utilizarlos fácilmente en placas de Arduino o Raspberry, ya que la primera trabaja con niveles de tensión de 5V y la segunda con 3,3V.

A continuación se presenta un fragmento de código en python que permite mover los servomotores cambiando la posición de la cámara.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
GPIO.setup(21, GPIO.OUT)
p = GPIO.PWM(21, 50)
p.start(7.5)
try:
    while True:

        p.ChangeDutyCycle(4.5)
        time.sleep(0.5)
        p.ChangeDutyCycle(10.5)
        time.sleep(0.5)
        p.ChangeDutyCycle(7.5)
        time.sleep(0.5)
except KeyboardInterrupt:
    p.stop()
GPIO.cleanup()
```

El servomotor utilizado es el modelo SG90, en la hoja de especificación de datos del fabricante puede observarse sus características, algunas de ellas se resumen en la **tabla 12**.

**Tabla 12:** Características del ServoMotor seleccionado (SG90)

<b>Peso</b>	9 g
<b>Dimensiones</b>	22.2 x 11.8 x 31 mm aprox.
<b>Longitud del cable de</b>	24.5cm



<b>Conector</b>	
<b>Torque</b>	1.8 kg / cm (a un voltaje de operación de 4.8V)
<b>Velocidad de Operación</b>	0.1 seg / 60° (a un voltaje de operación de 4.8V)
<b>Voltaje de Operación</b>	3.0V ~ 7.2V (valor recomendado 5V)
<b>“Dead Band Width”</b>	10 $\mu$ s
<b>Rango de Temperatura</b>	-30°C ~ 60°C
<b>Ángulo de Rotación</b>	180°
<b>Ancho de pulso</b>	500-2400 $\mu$ s

“Dead Band Width” significa Ancho de Banda Muerto y es el tiempo máximo hasta el cual el servomotor no detecta si hay una señal activa en su entrada de control. Por ejemplo, si se enviase una señal de control con un pulso activo de 1,5ms, entonces el servo se movería a su posición central (90°). Si ahora el pulso activo fuese de 1,5ms + 5 $\mu$ s, el servo seguiría en la misma posición que antes, es decir que no se moverá hasta que el pulso activo sea de 1,5ms  $\pm$  10 $\mu$ s.

Dead Band Width es un parámetro utilizado para evitar que el servo detecte señales de muy bajo duty cycle, ya que las mismas podrían tratarse de ruido y esto provocaría que el mismo se moviese sin haber enviado ninguna señal de control.

### 3.3.6. ETAPA 7: DESARROLLO DE HARDWARE DEDICADO

#### 3.3.6.1. Tarea 14: Desarrollo con Sensores

En esta etapa se presta atención a la motorización de la cámara por medio de servomotores, dividada en dos momentos: (1) Motorización en base a acciones de un usuario por medio de un teclado (por ejemplo a la espera de un enter, o pulsar una flecha señalando una dirección); (2) Motorización en base a la información de posición del orador recibida por medio de sensores de sonido.

Inicialmente se arma un esquema circuital para conectar el servomotor, que es lo que puede observarse en la **Figura 29**.

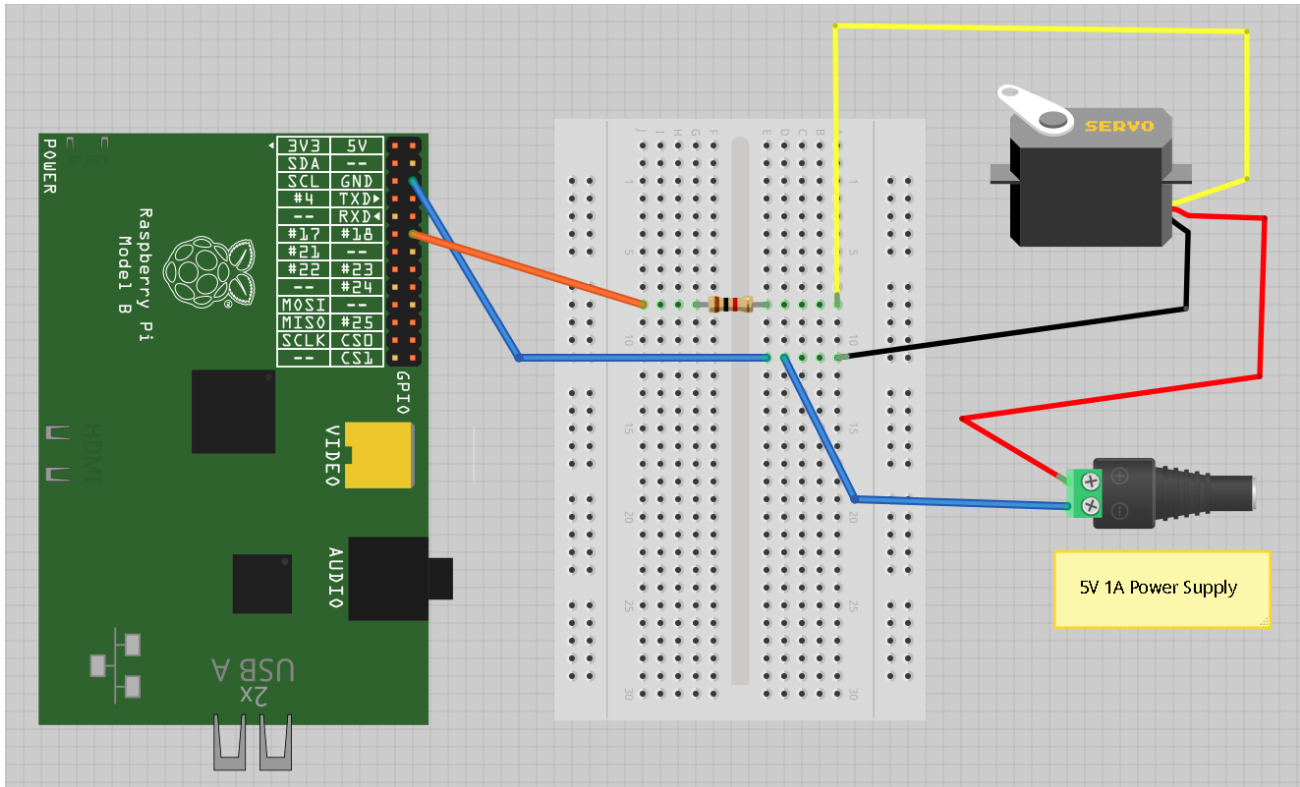


Figura 29. Modelo del Esquema Circuitual

### 3.3.6.2. Tarea 15: Implementación

Se utilizó la biblioteca Pi4J que permite configurar un pin de la raspberry como salida, entrada, PWM, o la funcionalidad que se quiera, es decir el “GPIO 4”, el número que hay que indicarle a la biblioteca Pi4J es distinto en cada caso. Los valores que hay que indicarle a la biblioteca son los que pueden observarse en la **figura 30**, donde:

- **BCM:** Es el número de GPIO. Por ej: 13 = GPIO 13 en el esquema de la raspberry
- **wPi:** Es el número que maneja internamente la biblioteca Pi4J para referirse al número de GPIO especificado al lado en la columna BCM. Por ejemplo para el GPIO 13 de la raspberry, cuando se quiera usar dicho pin con la biblioteca Pi4J habrá que indicar el número 23 y no 13



```

pi@raspberrypi ~/wiringPi $ gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | 8 | 3.3v | | | 1 | 2 | | | 5v | | |
| 3 | 9 | SDA.1 | IN | 1 | 3 | 4 | | | 5V | | |
| 4 | 7 | SCL.1 | IN | 1 | 5 | 6 | | | 0v | | |
| 4 | 7 | GPIO.7 | IN | 1 | 7 | 8 | 1 | ALT0 | TxD | 15 | 14 |
| | | 0v | | | 9 | 10 | 1 | ALT0 | RxD | 16 | 15 |
| 17 | 0 | GPIO.0 | IN | 0 | 11 | 12 | 0 | IN | GPIO.1 | 1 | 18 |
| 27 | 2 | GPIO.2 | IN | 0 | 13 | 14 | | | 0v | | |
| 22 | 3 | GPIO.3 | IN | 0 | 15 | 16 | 0 | IN | GPIO.4 | 4 | 23 |
| | | 3.3v | | | 17 | 18 | 0 | IN | GPIO.5 | 5 | 24 |
| 10 | 12 | MOSI | IN | 0 | 19 | 20 | | | 0v | | |
| 9 | 13 | MISO | IN | 0 | 21 | 22 | 0 | IN | GPIO.6 | 6 | 25 |
| 11 | 14 | SCLK | IN | 0 | 23 | 24 | 1 | IN | CE0 | 10 | 8 |
| | | 0v | | | 25 | 26 | 1 | IN | CE1 | 11 | 7 |
| 0 | 30 | SDA.0 | IN | 1 | 27 | 28 | 1 | IN | SCL.0 | 31 | 1 |
| 5 | 21 | GPIO.21 | IN | 1 | 29 | 30 | | | 0v | | |
| 6 | 22 | GPIO.22 | IN | 1 | 31 | 32 | 0 | IN | GPIO.26 | 26 | 12 |
| 13 | 23 | GPIO.23 | IN | 0 | 33 | 34 | | | 0v | | |
| 19 | 24 | GPIO.24 | IN | 0 | 35 | 36 | 0 | IN | GPIO.27 | 27 | 16 |
| 26 | 25 | GPIO.25 | IN | 0 | 37 | 38 | 0 | IN | GPIO.28 | 28 | 20 |
| | | 0v | | | 39 | 40 | 0 | IN | GPIO.29 | 29 | 21 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figura 30. Captura pantalla de parámetros

En base a lo explicado en la sección anterior, la primer etapa consistió en motorizar la solución mediante acciones de teclado. Lo cual se muestra en el código Java que se detalla a continuación. El código resultante (no consignado en este informe por su extensión), toma los valores arrojados por los sensores.

```

import com.pi4j.io.gpio.*;
import com.pi4j.util.CommandArgumentParser;
import com.pi4j.util.Console;

public class PwmExample {

    /**
     * Se puede especificar el pin que se quiere usar como pwm para hacer girar
     * a un servomotor con:
     * [ARGUMENT/OPTION "--pin (#)" | "-p (#)" ]
     * -- Ej: "--pin 4" o "-p 4"
     *
     * Aclaracion: los unicos pines de la raspberry 2 que pueden usarse como pwm
     * son:
     *
     *      GPIO 18 (pin 12 fisico) en la raspy (GPIO 1 para Pi4J)
     *      GPIO 13 (pin 33 fisico) en la raspy (GPIO 23 para Pi4J)
     *      GPIO 19 (pin 35 fisico) en la raspy (GPIO 24 para Pi4J)
     *      GPIO 12 (pin 32 fisico) en la raspy (GPIO 26 para Pi4J)
     *      pinoutPi4J.txt ubicado en /home/pi/javaFiles
     */
    public static void main(String[] args) throws InterruptedException {
        final Console console = new Console();
        console.title("<-- The Pi4J Project -->", "PWM Example");
        console.promptForExit();
        GpioController gpio = GpioFactory.getInstance();
        Pin pin = CommandArgumentParser.getPin(
            RaspiPin.class,
            RaspiPin.GPIO_01,
            args);
        GpioPinPwmOutput pwm = gpio.provisionPwmOutputPin(pin);
    }
}

```



```
com.pi4j.wiringpi.Gpio.pwmSetMode(com.pi4j.wiringpi.Gpio.PWM_MODE_MS);
com.pi4j.wiringpi.Gpio.pwmSetRange(1000);
com.pi4j.wiringpi.Gpio.pwmSetClock(250);

//Con el rango y clock configurado la frecuencia del PWM sera -> (19.2MHz / 1000 /
250) = 76.8 Hz = 13.2 mS

    // seteo angulo de servo en -90 grados
    pwm.setPwm(80);
    console.println("PWM angle is: -90 degrees -> " + pwm.getPwm() + " rate");

    console.println("\n\nPress ENTER to set the PWM angle to -45 degrees");
    System.console().readLine();

    // seteo angulo de servo en -45 grados
    pwm.setPwm(115);
    console.println("PWM angle is: -45 degrees -> " + pwm.getPwm() + " rate");

    console.println("\n\nPress ENTER to set the PWM angle to 0 degrees ");
    System.console().readLine();

    // seteo angulo de servo en 0 grados
    pwm.setPwm(150);
    console.println("PWM rate is: 0 degrees -> " + pwm.getPwm() + " rate");

    console.println("\n\nPress ENTER to set the PWM angle to +45 degrees");
    System.console().readLine();

    // seteo angulo de servo en +45 grados
    pwm.setPwm(185);
    console.println("PWM rate is: +45 degrees -> " + pwm.getPwm() + " rate");

    //NO LLEGA A +90 GRADOS. ES POR COMO ESTA COLOCADO EN LA BASE
    console.println("\n\nPress ENTER to set the PWM angle to 60 degrees");
    System.console().readLine();

    // seteo angulo de servo en 60 grados
    pwm.setPwm(200);
    console.println("PWM rate is: 60 degrees -> " + pwm.getPwm() + " rate");

    gpio.shutdown();
}
}
```

### 3.3.6.3. Tarea 16: Pruebas

Las pruebas también se realizaron en dos etapas primero sin la implementación de sensores para realizar manualmente por medio del teclado el movimiento de los servos en los distintos ángulos posibles y analizar cuestiones como la estabilidad del mismo, la fluidez del movimiento y el impacto de ello en el streaming de video. Luego se prueba la solución que incorpora a los sensores de sonido, probandose otros factores como por ejemplo la distancia hasta la que se reconoce el sonido del orador y se mueve la cámara como efecto de esta detección.

Queda por fuera de la actual solución el tema de ruido externo que pueda superar al propio orador y generar inconvenientes en el desplazamiento de la cámara. Es decir está pensado para que el orador pueda moverse por la sala libremente y que la cámara lo siga siempre y cuando exista un nivel de ruido ambiente bajo.

### 3.3.7. ETAPA 8: DESARROLLO DE APLICACIÓN DE STREAMING SOBRE UNA LAN

La **figura 31** plantea la estructura prevista donde dos o más Raspberry Pi (RPi) pueden conectarse y realizarse en tiempo real streaming de video, para lo cual se ya ha explicado en el presente informe las técnicas posibles y el uso del comando raspivid. El planteamiento es imaginar que en un determinado nodo A habrá un orador dando una conferencia (servidor) y pueden existir N nodos (clientes) conectados a esa conferencia. Todos los nodos son RPi que tienen implementada una solución que fue desarrollada en Java.



**Figura 31. Conectividad entre dos RPi – Streaming de Video**

La solución fue desarrollada en Java utilizando como base Eneter Messaging Framework para la transmisión de datos.

#### 3.3.7.1. Tarea 17: Generación de servidor de streaming sobre IPv6

El servidor constantemente está escuchando conexiones entrantes, o sea que soporta varios clientes. Cuando se conecta un cliente, el servidor le transmite los bytes resultantes de la captura del video con el comando raspivid. La cámara integrada de la Raspberry sólo captura video pero no sonido. Por lo cual se adquirieron unas placas económicas de sonido que se conectan por USB (ver figura 32).





**Figura 32. Placa de Sonido USB**

Esta pequeña placa resuelve el tema, con una entrada para micrófono y una salida para parlantes.

Como el video y audio no están integrados fue necesario hacer pruebas para medir los retrasos de ambos y trabajar en partición de paquetes para evitar que el audio llegue a destiempo con respecto al video. Esta fue la primer complejidad.

### **3.3.7.2. Tarea 18: Generación de cliente/player de streaming IPv6**

Se utilizó la biblioteca VLCJ en primera instancia. Existiendo problemas de procesamiento esto debido a que el VLC no soporta aceleración por Hardware en la Raspberry cuando se incrementa la resolución del video, se utiliza el 90% del procesamiento. Por lo cual se adaptó el cliente en Java para reproducir a través de mplayer ya que este reproductor sí soporta aceleración por Hardware.

Fue posible lograr generar una solución propia desarrollada en Java haciendo uso de bibliotecas, luego de probar diversas técnicas de streaming, tema que se aborda en el apartado a continuación. Producto de las pruebas también fue posible enviar a un congreso internacional resultados de las mediciones y la compartiva de dichas técnicas.

### **3.3.8. ETAPA 9: PRUEBAS DE APLICACIÓN DE STREAMING**

El streaming de video puede realizarse de dos maneras: (1) En tiempo real ó (2) Bajo Demanda. El interés en nuestro caso recae sobre el video en tiempo real.

La raspberry pi tiene una aplicación de consola que permite capturar fácilmente el video de la cámara incorporada. Esta aplicación se denomina raspivid<sup>21</sup> y soporta numerosos parámetros de configuración entre los cuales se encuentra la calidad del video a capturar, los cuadros por segundo y la posibilidad de enviar el resultado a un archivo o a la salida estándar.

---

<sup>21</sup> <https://www.raspberrypi.org/documentation/usage/camera/raspicam/raspivid.md>



Para realizar streaming existen diferentes alternativas:

- 1) Utilización de software especializado para streaming de video como es el caso de VLC o GStreamer. Ambos permiten tomar el video directamente desde raspivid y enviarlo utilizando distintos protocolos por ejemplo: http, rtp, rtsp o mms.
- 2) Utilizar el comando NC de Linux que corresponde a la herramienta NetCat. Esta herramienta permite enviar información directamente a otro equipo a un puerto específico. De esta forma la salida del video de raspivid puede ser enviada directamente por la red hacia otro host, que la toma y puede mostrarla con cualquier reproductor de video como por ejemplo el mplayer.
- 3) Utilizando el proyecto “picam”<sup>22</sup> que no permite hacer streaming por si solo, pero utilizando un servidor “nginx”<sup>23</sup> y el módulo RTMP (Real-Time Messaging Protocol)<sup>24</sup> se puede hacer streaming del video.
- 4) Desarrollo de una aplicación: Aprovechando la posibilidad de tomar el video mediante el comando raspivid y enviándolo a la salida estándar es posible capturarlo mediante una aplicación. Por ejemplo desde Java se puede utilizar la clase Process y asignarle al mismo la ejecución del comando raspivid con los parámetros deseados:

```
Process      procesoRaspivid      =      Runtime.
Runtime.getRuntime().exec("raspivid -n -b 15000000 -w 1920
-h 1080 -fps 30 -t 0 -o -");
```

Donde se invoca al comando raspivid con los parámetros presentados en la **tabla 13**.

**Tabla 13.** Comandos del raspivid

-w 1920	Para establecer el ancho del video en 1920 pixel
-h 1080	Para establecer el alto del video en 1080 pixel
-b 15000000	Para establecer el bit rate a 15Mbit/s
-n	Evita que muestre una ventana con el video capturado
-fps 30	Establece la captura a 30 cuadros por segundo
-t 0	Captura el video indefinidamente
-o -	Envía el resultado a la salida estándar

<sup>22</sup> <https://github.com/iizukanao/picam>

<sup>23</sup> Es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3)

<sup>24</sup> <https://github.com/arut/nginx-rtmp-module>



Al flujo de video capturado se lo divide en paquetes de información para que puedan ser enviados por la red:

```
byte[] datosVideo = new byte[4096];
while (flujoVideo.read(datosVideo) != -1)
{
//envío de los paquetes por la red
}
```

Para el manejo de las comunicaciones pueden utilizarse las bibliotecas estándar de java o puede facilitarse el proceso incorporando bibliotecas externas por ejemplo Eneter Messaging Framework [ENE10] que es un framework para comunicación entre procesos multiplataforma gratuito para uso no comercial. En [ENE10] Ondrej Uzovic ofrece código desarrollado para el Cliente en .NET y el Servidor en Java, su objetivo es hacer streaming entre un cliente computadora y un Servidor RPi. En nuestro caso para formar una red P2P de igual prioridad se utilizaron dos Raspberrys Pi lo que requirió desarrollar el servidor también en Java. Una vez realizado esto se pudo comparar esta alternativa con las anteriores, obteniéndose los resultados que se presentan a continuación.

Considerándose en todos los casos una cantidad de frames por segundo fija (24 fps), se realizaron las pruebas en 3 resoluciones distintas, para ver que sucedía con streaming a baja calidad inicialmente y luego cuando se va mejorando dicha calidad: 320 x 240, 640 x 480 y 800 x 600. Lo primero a destacar, es que en prácticamente todos los casos siempre se obtuvo menor retardo con la alternativa 4 (A4), salvo contra la A2 en baja resolución en donde existe una ínfima diferencia. Para poder sintetizar los resultados obtenidos se ha tomado en cuenta el retardo de dicha transmisión (Retardo 4, R4) en proporción con los otros retardos originados, por ejemplo, si se toma en consideración una resolución de 800 x 600 casi se disminuye a la tercera parte el retardo con respecto a la alternativa 1, esto se calculó de la siguiente manera:  $R1/R4=2,71$ . En la **tabla 14** se muestra la mejora que se obtiene en la disminución de retardo tomando los resultados de la alternativa 4 (A4) frente a las otras alternativas planteadas.

**Tabla 14.** Comparativas de Disminución de retardo tomando como Base la Alternativa 4(A4)

Resoluciones	(A1) VLC	(A2) NC Linux	(A3) PICAM
800 x 600	2,71	38,89	3,66
640 x 480	4,61	48,20	8,89
320 x 240	4,14	0,97	10,04

Cabe aclarar que a pesar de los retardos, no se pierde fluidez en el streaming transmitido. Tomando VLC que fue la alternativa mejor entre las dos primeras comparadas

(A1), se hizo la comparativa de retardo con respecto a A4 en full hd (ya que la cámara integrada soporta este tipo de resolución), dando por resultado: 1,26 a favor de la alternativa 4. Es decir salvo la excepción ínfima a baja resolución (representado en la **tabla 14**, siendo el único valor menor a 1), en todos los casos restantes la mejor alternativa en cuanto a retardos es la solución propia desarrollada.

En el **Anexo C** se encuentra una comparativa de los tiempos de retardo que se obtuvieron con cada una de las técnicas de streaming.

### 3.3.9. ETAPA 10: IMPLEMENTACIÓN

Luego de realizarse diversas pruebas se implementaron los prototipos para ser utilizados en la Universidad. En la **figura 33** se muestra en funcionamiento la solución.



**Figura 33.** Conectividad entre dos nodos de la Universidad

## 4. Producción Científico-Tecnológica

En el transcurso del proyecto, se logró planificar el prototipo a realizar, decidiendo la tecnología a utilizar en cuanto a hardware y producto de esto se realizaron las siguientes publicaciones.

1. Congreso: WICC 2015 – (XVII Workshop de Investigadores en Ciencias de la Computación), Universidad Nacional de Salta, Argentina



Título del artículo: Desarrollo de un sistema de video conferencia autónomo sobre IPv6

Link de acceso: <http://hdl.handle.net/10915/46080>

2. Revista: CYTA (Ciencia y Técnica Administrativa)

Título del artículo: Utilización del Protocolo IPv6 para la realización de Video-Conferencias

Volumen: 14 - Número: 2 - Artículo: 1

Link de Acceso: <http://www.cyta.com.ar/ta1402/v14n2a1.htm>

3. Revista: CYTA (Ciencia y Técnica Administrativa)

Título del artículo: Herramienta de apoyo para las clases impartidas en laboratorios de informática: construcción de una solución en red basada en IPv6

Volumen: 15 - Número: 3 - Artículo: 1

Link de Acceso: <http://www.cyta.com.ar/ta1503/v15n3a1.htm>

4. Congreso: CACIC 2016 – (XXII Congreso Argentino de Ciencias de la Computación), Universidad Nacional de San Luis, Argentina

Título del artículo: Sistema autónomo de video conferencia motorizado - Implementación sobre Raspberry Pi

Link de acceso: <http://hdl.handle.net/10915/56533>

5. Congreso: CIACA 2016 – (IV Conferencia Iberoamericana de Computación Aplicada), Lisboa, Portugal

Título del artículo: Análisis de técnicas de streaming de video usando Raspberry Pi

Link de acceso: <http://ciaca-conf.org/es/>

## 5. Conclusiones

Cabe destacar primeramente que el equipo de trabajo es interdisciplinario formado en su mayoría por personas vinculadas a la ingeniería en informática y electrónica, con 5 alumnos que participan en él lo que conforma al 29% de los integrantes del equipo, los cuales se encuentran formándose en actividades de I+D, dándose de este modo una alta importancia a la formación de recursos humanos.

El proyecto propuso desafíos no sólo en cuanto a desarrollo en Java para poder generar la solución sino también en cuanto a hardware. La Raspberry permite generar una gran cantidad de aplicaciones que requieran sensores o hardware dedicado, a un bajo costo. En este artículo se presenta el análisis de técnicas de streaming utilizando el módulo de



cámara integrado por medio del puerto SCI. Seleccionar la mejor técnica es fundamental al momento de realizar streaming de videos ya que existe una notable disminución de los retardos, resultando la mejor opción hacer uso del comando raspivid y enviarlo en la salida estándar; para luego capturarlo mediante una aplicación desarrollada.

Con esta técnica es posible desarrollar un sistema de video conferencia autónomo en HD que se pueda conectar directamente a un televisor o monitor con HDMI. Además la Raspberry permite conectar sensores de sonido/ultrasonido los cuales pueden detectar sonido y medir distancia, de este modo pueden ser un buen complemento para que a través de servomotores se pueda mover la RPi haciendo de este modo que la cámara pueda seguir al orador. Quedando como resultado implementado un prototipo que permite realizar videoconferencias en HD.



## 6. Bibliografía

- [AZA07] Azael Fernandez Alcantara. "Direcciones IPv4 ¿Recurso de Internet en Agotamiento?". México, (2007)  
<http://www.enterate.unam.mx/Articulos/2007/junio/art1.html>
- [BAL14] Ballesta Torre F. "Arduino+Blaspberry Pi"; Universidad de Catalunya, España; (2014)
- [CEN12] CETIN, G.; BILISIM ENSTITUSU, ELEKTRON. VE BILGISAYAR EGITIMI, GAZI. "Development of an mobile agent platform for IPv6 network environment: Mobile-C", Universidad de Ankara, Turkía, (2012).  
[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6204507](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6204507)
- [CEN97] Fernández Centeno, Miguel Ángel. "Compresión de Video – Los CODECs de Video". Extractado de la revista 3D Magazine Numero 11. Septiembre 1997 (1997). <http://ingenieria.udea.edu.co/~marthac/multimedia/codecvideo.html>
- [CHA14] Chavez Paredes, Evelyn Pamela, and Diego Armando Castro Arrobo. "Evaluación de codecs de video sobre tráfico multicast en IPV6 para el desarrollo de un prototipo en el laboratorio LIRSI-FIE." (2014).
- [CIC14] Cicileo, Guillermo, and Mariela Rocha. "Evolución de la red de interconexión universitaria de Argentina." (2014).  
<http://hdl.handle.net/10786/778>
- [CIS07] CISCO, "QoS Output Scheduling on Catalyst 6500/6000 Series Switches Running CatOS System Software" (2007)  
<http://www.cisco.com/c/en/us/support/docs/switches/catalyst-6000-series-switches/10582-60.html>
- [CIS09] CISCO. "Internetwork Design Guide -- Internetworking Design Basics" (2009)  
[http://docwiki.cisco.com/wiki/Internetwork\\_Design\\_Guide\\_-\\_Internetworking\\_Design\\_Basics](http://docwiki.cisco.com/wiki/Internetwork_Design_Guide_-_Internetworking_Design_Basics)
- [CLA15] RED CLARA. "Proyectos" (2015)  
[https://wiki2.redclara.net/wiki/masred/index.php/INFORMACION\\_GENERAL\\_DEL\\_GT\\_Servicios\\_con\\_IPv6\\_en\\_CLARA](https://wiki2.redclara.net/wiki/masred/index.php/INFORMACION_GENERAL_DEL_GT_Servicios_con_IPv6_en_CLARA)
- [CRU15] Cruz José. Oracle. "Java ME 8 + Raspberry Pi + Sensores = Un Mundo IoT - Part 1". (2015)  
<http://www.oracle.com/technetwork/es/articles/java/javame8-raspberry-pi-sensores-2687098-esa.html>
- [DEL16] Del Campo García Mariano, "Programar posiciones en un Micro Servo Tower Pro SG90 9G". (2016)  
<http://miarduinounotieneunblog.blogspot.com.ar/2016/01/programar-posiciones-en-un-micro-servo.html>
- [DIA10] Díaz, Javier F., et al. "Modelos de QoS en redes IPv6, integración con otras redes." XII Workshop de Investigadores en Ciencias de la Computación. 2010.  
<http://hdl.handle.net/10915/19420>
- [ENE10] ENETER Messaging Framework. Cross-Plataform Framework for Interprocess Communication <http://www.eneter.net/>



- [ESP11] GOBIERNO DE ESPAÑA, Ministerio de Industria, Energía y Turismo. "IPv6 - Más direcciones para Comunicarnos Mejor", España. (2011). <http://www.ipv6.es>
- [FAC13] Facchini, Higinio Alberto, et al. "Análisis de prestaciones de tráfico multicast en redes mixtas IPv4 e IPv6." XV Workshop de Investigadores en Ciencias de la Computación. (2013). <http://hdl.handle.net/10915/27085>
- [GAR15] García Muelas C, "Integración de redes telemáticas, IoT con Blaspberry Pi"; Universidad de Catalunya, España (2015)
- [GRI12] GRIDTICS (grupo de investigación y desarrollo en tecnologías de la información y comunicación / utn – frm). "IPV6 -Ya lo probamos, luego la hora de usarlo". Argentina, Junio 2012. (2012) <http://gridtics.frm.utn.edu.ar/site/?p=150>
- [GUA15] Gualotuña, Espinosa, and Cristian Fabricio. "Adaptación del algoritmo Omega para ser utilizado en un prototipo de red de sensores inalámbrica 6LowPAN (IPv6 over Low Power Wireless Personal Area Networks)". Diss. Quito, (2015). <http://bibdigital.epn.edu.ec/handle/15000/11775>
- [HOH13] Christian Hohmann. "What is FIFO (flow management)". (2013) <https://hohmannchris.wordpress.com/2014/01/18/fifo-lifo-explained-in-a-simple-video/>
- [IET81] IETF. "RFC 791". (1981) <https://tools.ietf.org/html/rfc791>
- [INN12] INNOVA RED (Red Nacional de Investigación y Educación de Argentina). "Acceso IPV6" (2012) <http://www.innova-red.net/node/38>
- [KUR99] Kurose James, Ross Keith. "Scheduling and Policing Mechanisms" (1999) [http://www2.ic.uff.br/~michael/kr1999/6-multimedia/6\\_06-scheduling\\_and\\_policing.htm](http://www2.ic.uff.br/~michael/kr1999/6-multimedia/6_06-scheduling_and_policing.htm)
- [LACne] LACNIC (Latin America and Caribbean Network Information Centre). CICLEO GUILLERMO. "IPv6 en el Ambiente Académico" <http://portalipv6.lacnic.net/es/ipv6/ipv6-en/ambiente-acad-mico-0>
- [LIAne] Liakopoulos Athanassios, Kalogeras Dimitrios, Primpas Dimitrios. "QoS support in IPv6 environments". Greece. <https://www.6diss.org/tutorials/qos.pdf>
- [MER11] Mercado, Gustavo, et al. "Implementación y Evaluación de métodos de Traslación de Protocolos para la transición IPv4-IPv6." XVII Congreso Argentino de Ciencias de la Computación. (2011). <http://hdl.handle.net/10915/18769>
- [MER13] Mercado, Gustavo, et al. "Aplicaciones de Internet de las Cosas SIPIA6-Red de Sensores Inalámbricos con IPv6." XV Workshop de Investigadores en





- Ciencias de la Computación. (2013).  
<http://hdl.handle.net/10915/27097>
- [MER14] Mercado, Gustavo, et al. "ACyT-Net Red Académica Científica y Tecnológica Experimental de Mendoza Argentina." XVI Workshop de Investigadores en Ciencias de la Computación. (2014).  
<http://hdl.handle.net/10915/41702>
- [PER05] Perez, David A. "Análisis de requerimiento de QoS sobre redes IP Multicast. Tesis de Maestría UNLP" (2005)  
<http://hdl.handle.net/10915/4104>
- [SAL11] Salcedo Parra Octavio José, López Danilo, Ríos Ángela Patricia, "Desempeño de la calidad del servicio (QoS) sobre IPv6". (2011)  
[http://www.scielo.org.co/scielo.php?pid=S0123-921X2011000100004&script=sci\\_arttext](http://www.scielo.org.co/scielo.php?pid=S0123-921X2011000100004&script=sci_arttext)
- [SHA15] Sharma Rajeev, Sehra Sukhjit Singh, Sehra Sumeet Kaur. "Review of Different Queuing Disciplines in VOIP", Video Conferencing and File Transfer. (2015)  
[https://www.researchgate.net/figure/277615820\\_fig7\\_Figure-8-Modified-Deficit-Round-Robin](https://www.researchgate.net/figure/277615820_fig7_Figure-8-Modified-Deficit-Round-Robin)
- [STA15a] STATISTA. "Global Google Hangouts usage penetration among internet users worldwide as of 2nd quarter 2015" (2015)  
<http://www.statista.com/statistics/477047/google-hangouts-reach-by-age-worldwide/>
- [STA15b] STATISTA. "Global Skype usage penetration among internet users worldwide as of 2nd quarter 2015" (2015)  
<http://www.statista.com/statistics/476998/skype-reach-by-age-worldwide/>
- [TAF14] Taffernaberry, Juan Carlos, et al. "PMIP6: Análisis, Evaluación y Comparación de ambientes Proxy Mobile IP en versión 6, aplicado a Redes de Avanzada." XVI Workshop de Investigadores en Ciencias de la Computación. (2014).  
<http://hdl.handle.net/10915/41658>
- [TAF15] Taffernaberry C, Tobar S, Mercado G, Noguera J, Perez Montes "Aplicaciones del RFC 5213 – Proxy Mobile IPv6 – Comparación de ambientes móviles en redes académicas". (2015)  
<http://44jaiio.sadio.org.ar/sites/default/files/ietfday17-21.pdf>
- [UNE10] Universidad Nacional de Educación a Distancia. "Más multimedia para explicar Multimedia - Conversión entre Formatos de video y audio". (2010)  
[http://ocw.innova.uned.es/mmm3/video\\_digital/contenidos/pdf/Conversion\\_entre\\_formatos.pdf](http://ocw.innova.uned.es/mmm3/video_digital/contenidos/pdf/Conversion_entre_formatos.pdf)
- [VAL14] Valera A., Soriano A., Vallés M (2014) "Plataformas de Bajo Coste para la Realización de Trabajos Prácticos de Mecatrónica y Robótica"; Instituto Universitario de Automática e Informática Industrial, Universidad Politécnica de Valencia, España
- [3CU13] Cu Electrónica. "Calidad de Servicio". (2013).  
<https://sites.google.com/site/3cuelelectronica/home/ethernet/calidad-de-servicio>

**Anexo A – Tabla Comparativa Software Existente**

En este anexo se presentan soluciones para videoconferencia que tienen soporte para IPv6.

MARCA	¿PAGO?	CARACTERÍSTICAS
POLYCOM	SI	<p>Polycom HDX 4000 Series Telepresencia Personal Premier<sup>25</sup></p> <ul style="list-style-type: none"> <li>• IPv6 (DISA. Defense Information Systems Agency)</li> <li>• Autenticación de contraseña de seguridad.</li> <li>• Software MPPlus hasta para 4 sitios.</li> <li>• Polycom Touch Control (solo HDX 4500).</li> <li>• Polycom HDX 4500</li> </ul> <p>Polycom RealPresence Distributed Media Application (DMA)<sup>26</sup></p> <ul style="list-style-type: none"> <li>• Pv4/IPv6 - H.323 y SIP.</li> <li>• Administración del ancho de banda de la red.</li> <li>• Enrutamiento de acuerdo a prioridad.</li> <li>• Clase de servicio con Juniper Networks.</li> </ul> <p><u>Polycom RealPresence Group 310<sup>27</sup></u></p> <ul style="list-style-type: none"> <li>• <b>Compatible con IPv4 e Ipv6.</b></li> <li>• <b>1 puerto Ethernet 10/100/1G.</b></li> <li>• <b>Auto-MDIX.</b></li> <li>• <b>H.323 y/o SIP hasta 3 Mbps.</b></li> <li>• <b>Tecnología Polycom Lost Packet. Recovery: La tecnología LPR protege a las video llamadas del impacto de la pérdida de paquetes destinando temporariamente una pequeña porción del ancho de banda para ser usada para corrección de errores posteriores (Forward error correction (FEC)).</b></li> </ul>
PROYECCIONES DIGITALES S.A. <sup>28</sup>	SI	<ul style="list-style-type: none"> <li>• End Points Full HD para cualquier tipo de sala y aplicación.</li> <li>• Soluciones para la comunicación de usuarios de PC, notebooks o dispositivos móviles.</li> <li>• Infraestructura de MCU, Transit, Gatekeeper, administración de redes y video. Integración Smart Video.</li> <li>• MCU y soluciones para grandes grupos de usuario en la nube.</li> </ul>
LIFESIZE <sup>29,30</sup>	SI	<ul style="list-style-type: none"> <li>• Soportan configuración dual de direccionamiento IPV4 e IPV6 impidiendo deshabilitar el direccionamiento IPv4 en su sistema.</li> <li>• Las llamadas que usan direccionamiento IPv6 utilizan el protocolo H.323.</li> <li>• Además de la dirección IPv4 local especificada, se puede asignar una dirección IPv6 a su sistema si se está usando direccionamiento IPv6 en su red de la siguiente manera: Preferencias de Acceso del Administrador: Red: General y elegir Habilitada si se prefiere IPv6.</li> <li>• Habilitar o deshabilitar IPv6, causa el rebooting del sistema después de dejar la pantalla de preferencias.</li> </ul>
IPV6 & VIDEO CONFERENCING/ TELEPRESENCE <sup>31</sup>	SI	<ul style="list-style-type: none"> <li>• El sistema soporta interoperatividad de cualquiera con cualquiera, con endpoints de video conferencia de definición alta y estándar, y colaboración con video aplicaciones de escritorio como Cisco WebEx meeting applications y Cisco Digital Media System.</li> <li>• El sistema incluye un dispositivo de control de 12 pulgadas Cisco TelePresence Touch que provee una interfaz de usuario intuitiva</li> </ul>

<sup>25</sup> <http://latinamerica.polycom.com/content/dam/polycom/common/documents/data-sheets/hdx-4000-ds-esla.pdf>

<sup>26</sup> <http://latinamerica.polycom.com/content/dam/polycom/common/documents/data-sheets/realpresence-dma-data-sheet-esla.pdf>

<sup>27</sup> <http://latinamerica.polycom.com/content/dam/polycom/common/documents/data-sheets/realpresence-group-310-data-sheet-esla.pdf>

<sup>28</sup> <http://proyecciones.net/videoconferencia?gclid=CMPx7KvY9sUCFYXHWodCCYAeQ>

<sup>29</sup> <http://www.lifesize.com/en/solutions/industry/state-and-local-government>

<sup>30</sup> <http://www.lifesize.com/~media/Documents/Product%20Documentation/Video%20Systems/Guides%20and%20Reference/Video%20Administrator%20Guide%2045%20EN.ashx>

<sup>31</sup> <http://ipv6.com/articles/applications/IPv6-Video-Conferencing-Telepresence.htm>



		<p>para llamadas agendadas y acceso a contactos y directorios.</p> <ul style="list-style-type: none"> <li>La resolución 1080p y 720p en una pantalla premium de 32 pulgadas ayuda a asegurar la telepresencia clara y natural.</li> </ul>
CISCO TELEPRESENCE SYSTEM (TELEPRESENCE 1000) CISCO UNIFIED COMMUNICATIONS MANAGER <sup>32</sup>	SI	<ul style="list-style-type: none"> <li>IPv6 Video Interoperability Cisco Unified Communications Manager</li> <li>Soporta la transmisión de streams de video y aplicaciones en líneas SIP y trunks SIP en modo IPv6 así como en modo de doble pila.</li> <li>Soporta interworking con otros clusters de Unified Communications y Servers de Video Comunicaciones (VCS) sobre líneas SIP y trunks SIP que hayan sido configurados para usar IPv4 o IPv6.</li> </ul> <p>Precio: Desde U\$S 79.000.- Alrededor de U\$S 299.000 para una sala con un sistema de 3 pantallas (TelePresence 3000).</p>
JITSI <sup>33</sup>	NO	<p>Jitsi (antes SIP Communicator) es una aplicación de videoconferencia, VoIP, y mensajería instantánea que soporta varios sistemas operativos, incluyendo Windows, así como sistemas de tipo UNIX, como Linux, Mac OS X y BSD siendo distribuido bajo los términos de la GNU Lesser General Public License, Jitsi es software libre y de código abierto.</p> <p>Jitsi maneja correctamente IPv6 permitiéndole la comunicación directa de PC a PC, en el caso, por ejemplo en que ambas partes están "atrapadas" detrás de routers NAT, pero se puede obtener una dirección IPv6 accesible a través de un túnel-corredor.</p> <p>La comunidad Jitsi ha completado también una aplicación llamada ice4j.org, que se utiliza para proporcionar capacidades de NAT transversal, y ayudar a la transición de IPv4 a IPv6.</p> <p>Jitsi incluye facilidades como: Transferencia de llamadas atendidas y/o ciegas; Cambio a "ausente" automático; Autorreconexión; Grabación de llamadas; Cifrado con protocolos SRTP y ZRTP; Llamadas de conferencia; Establecimiento de conexión de medios directa mediante protocolo ICE; Streaming de escritorio; Almacenamiento de contraseñas cifradas con una contraseña maestra; Transferencia de archivos para los servicios XMPP; AIM/ICQ; Windows Live Messenger Service, Yahoo!; Cifrado de mensajería instantánea con Off-the-Record Messaging; Soporte IPv6 para SIP y XMPP; Relevo (relaying) de medios con protocolo TURN; Indicador de mensaje en espera (RFC 3842); Llamadas de voz y video mediante protocolos SIP y XMPP, con H.264, H.263, VP8 para codificación de video; Telefonía de banda ancha con G.722 y Speex; Usa lookup combinado de IPv4 e IPv6 de NetworkUtils.</p>
PANOPTO <sup>34</sup>	SI	<p>Es la única plataforma que provee grabación de video integrada, live streaming, video management, y búsqueda dentro del video a empresas y universidades.</p> <p>Permite realizar:</p> <p>Empresas: Entrenamiento On Line; You Tube Cooperativo; Eventos en Vivo, Ventas; Comunicaciones corporativas, etc.</p> <p>Educación: Clases Filmadas; Captura de Conferencias; Clases Abiertas Masivas en Línea (MOOCs); Eventos; Demostraciones; Campus en You Tube; Registros de Estudiantes; etc.</p>

<sup>32</sup>[http://www.cisco.com/c/en/us/td/docs/voice\\_ip\\_comm/cucm/admin/10\\_0\\_1/ccmfeat/CUCM\\_BK\\_F3AC1C0F\\_00\\_cucm-features-services-guide-100/CUCM\\_BK\\_F3AC1C0F\\_00\\_cucm-features-services-guide-100\\_chapter\\_0100100.pdf](http://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cucm/admin/10_0_1/ccmfeat/CUCM_BK_F3AC1C0F_00_cucm-features-services-guide-100/CUCM_BK_F3AC1C0F_00_cucm-features-services-guide-100_chapter_0100100.pdf)

<sup>33</sup><https://jitsi.org/Main/HomePage>

<sup>34</sup><http://panopto.com/>

## 7. Anexo B – Instructivo de Instalación del Entorno de Desarrollo para Raspberry PI

### 7.1. Software utilizado

#### Actualizaciones (18-09-15):

- Ubuntu 14.04.3 LTS 32 bits
- Oracle VM VirtualBox 5.0.4
- Java: JDK/JRE 8u60

### 7.2. Descarga de software

Lo primero que hay que hacer es descargar la imagen de Ubuntu y el software Oracle VM VirtualBox. Estas descargas hay que hacerlas desde el sistema operativo host, Windows 7 en este caso.

### 7.3. Descarga de Ubuntu

Para descargar Ubuntu hay que dirigirse a su página web oficial:

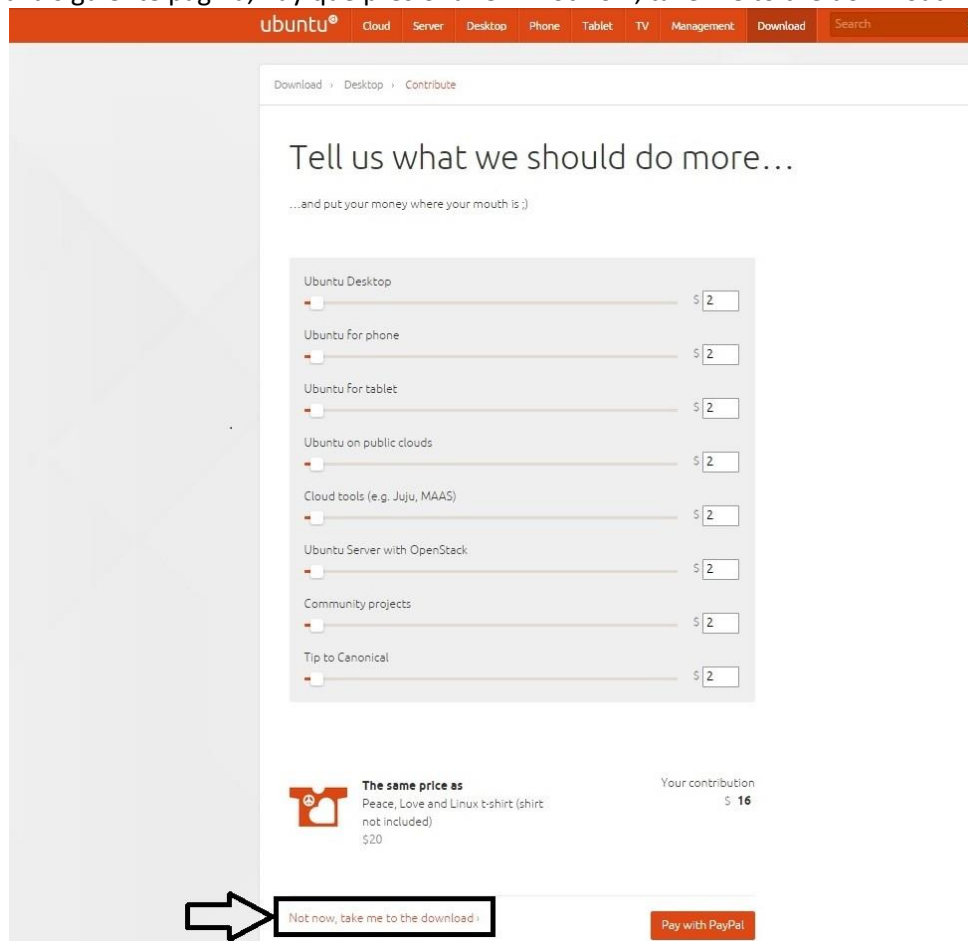
[www.ubuntu.com/download/desktop](http://www.ubuntu.com/download/desktop).

Aparecerá la siguiente página, hay que elegir la versión de 32 bits y apretar en “Download”.

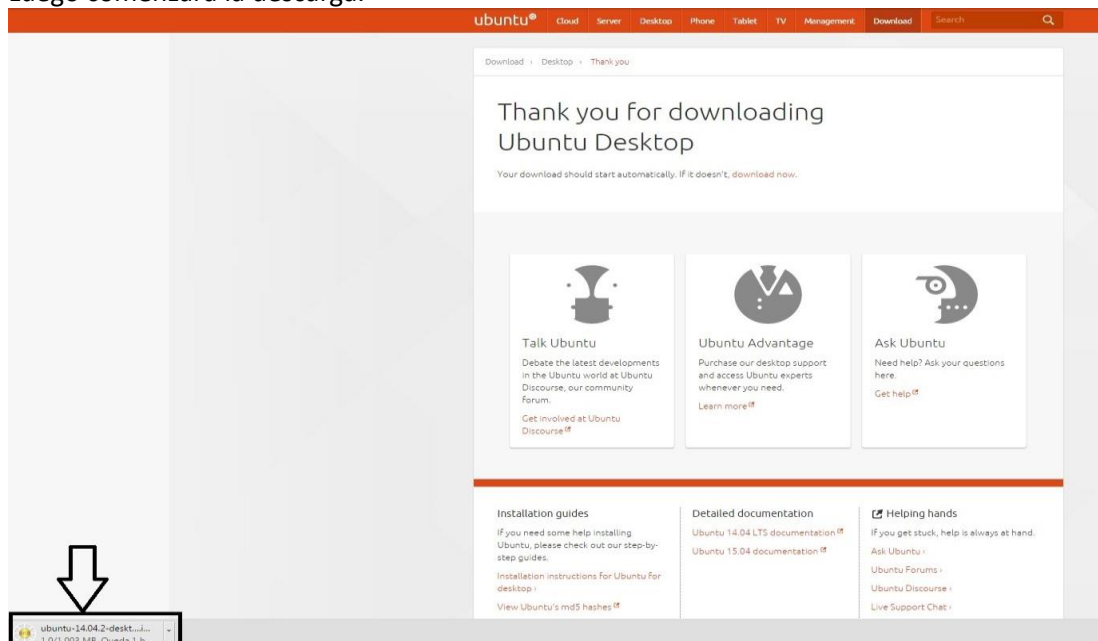


The screenshot shows the Ubuntu website's download page. The main heading is "Download Ubuntu Desktop". Underneath, it lists "Ubuntu 14.04.2 LTS" as the "Recommended for most users" version. A dropdown menu labeled "Choose your flavour" is set to "32-bit — for machines with less than 4GB RAM". A white arrow points to this option. Below the dropdown is a "Download" button, also pointed to by a white arrow. Below the 14.04.2 LTS section, the "Ubuntu 15.04" section is visible, with its "Choose your flavour" dropdown set to "64-bit — recommended".

**Nota:** la versión utilizada para el tutorial fue la 14.04.1. La que está disponible en el sitio (14.04.2) al día de la fecha de la captura de pantalla realizada también puede ser utilizada sin problemas. Aparecerá la siguiente página, hay que presionar en “Not now, take me to the download”.



Luego comenzará la descarga.

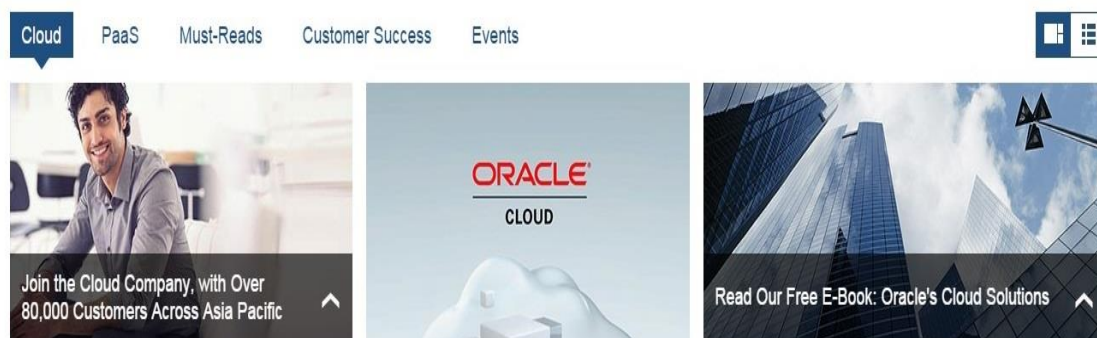
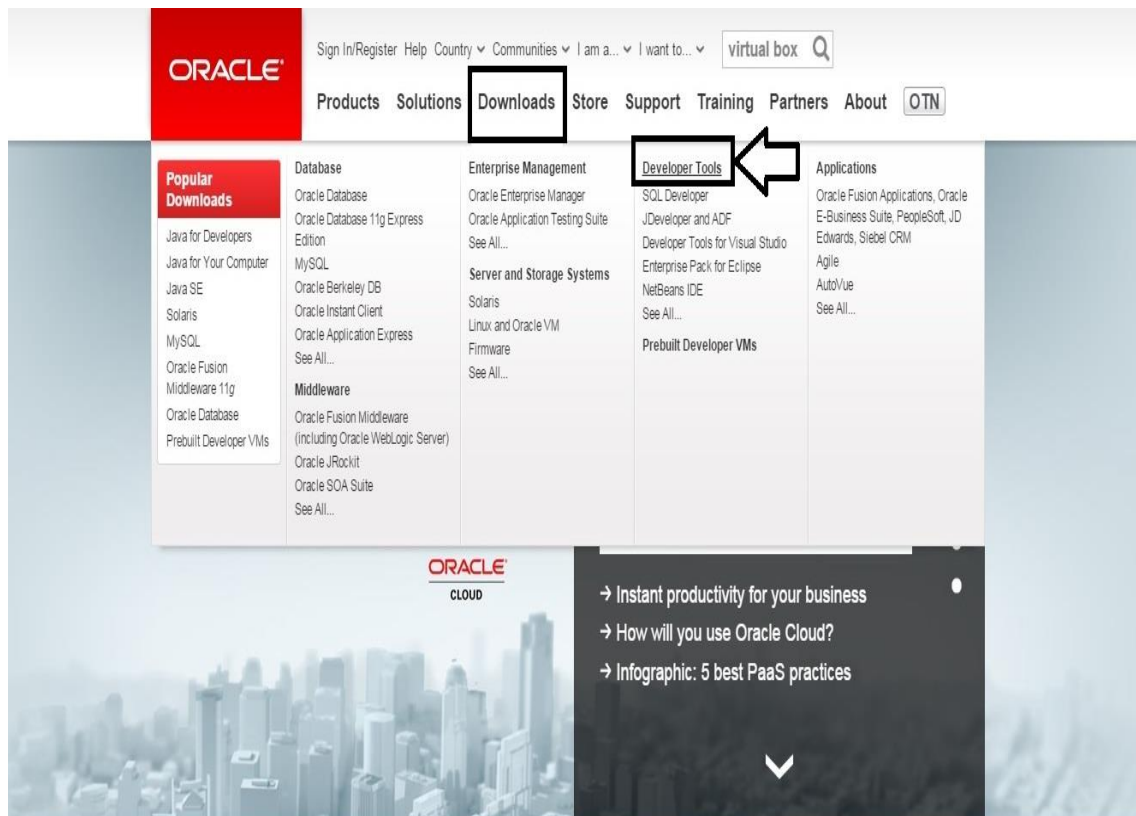


Con esto termina la descarga de Ubuntu, localice el directorio donde se guardará la imagen descargada ya que habrá que utilizarla en los siguientes pasos.

#### 7.4. Descarga de Oracle VM VirtualBox

Ahora hay que descargar el software de virtualización Oracle VM VirtualBox, para ello nos dirigimos a la página oficial de Oracle: [www.oracle.com](http://www.oracle.com).

Allí, dentro de la sección “Downloads”, seleccionamos “Developer Tools”.



Luego, en la pantalla que aparece, vamos hasta la sección “Servers and Storage Systems” y seleccionamos “VM VirtualBox”.



- Business Process Management
- Coherence
- Content Management SDK
- Crystal Ball
- Data Integrator and Suite
- Discoverer
- Endeca Information Discovery
- Endeca Server
- Enterprise Repository
- Entitlements Server
- Event Processing
- Event-Driven Architecture Suite
- Forms & Reports Services
- Real-Time Decisions
- Service Bus
- Service Bus for Financial Services
- Service Registry
- SOA Suite
- Tuxedo
- Virtual Assembly Builder
- Web Services Manager
- Web Tier
- WebCenter Content
- WebCenter Portal
- WebCenter Sites
- WebLogic Integration
- WebLogic Server 12c

**Servers and Storage Systems**

- Oracle Linux and Oracle Enterprise Kernel
- Solaris 10
- Solaris 11
- Solaris Cluster
- Solaris Legacy Containers
- Grid Infrastructure
- Oracle VM Server
- Oracle Flash Storage System and Pillar Axiom Plug-ins
- Secure Global Desktop
- Sun ZFS Storage Appliance Plug-ins
- Vdbench
- VM VirtualBox** ←
- Virtual Desktop Infrastructure

**Developer Tools**

- ADF Faces
- Application Express Standalone
- BI Publisher
- BI Spreadsheet Add-in
- BPEL Process Manager
- Data Miner
- Developer Suite 10g (includes Designer)
- Developer Tools for Visual Studio
- Enterprise Pack for Eclipse
- Forms & Reports Services
- JDeveloper & ADF
- NetBeans IDE
- SOA Suite
- Solaris Studio
- SQL Developer
- SQL Developer Data Modeler
- StorageTek
- Team Productivity Center
- TopLink
- Warehouse Builder
- XML Developer Kit 10g

**Applications**

Luego, en la siguiente pantalla, pulsamos sobre el link de descarga “Windows Installer”.

Oracle Technology Network > Server and Storage Systems > Oracle VM VirtualBox > Downloads

Overview Downloads Documentation Community Support Learn More

### Oracle VM VirtualBox

The latest release is version 4.3.28.

On this page you can download:

- Oracle VM VirtualBox
- Oracle VM VirtualBox Extension Pack
- Oracle VM VirtualBox Source Code
- Oracle VM VirtualBox Pre-built Appliances

### Oracle VM VirtualBox

Freely available for Windows, Mac OS X, Linux and Solaris x-86 platforms:

Platform	File
Windows (32-bit/64-bit)	<a href="#">.Windows Installer</a>
Mac OS X	<a href="#">dmg Image</a>
Solaris 10 5/08 and later (64-bit)	<a href="#">Solaris Package</a>
<b>Linux 32-bit Platforms</b>	
Ubuntu 13.04 (Raring Ringtail), 13.10 (Saucy Salamander), 14.04 (Trusty Tahr), 14.10 (Utopic Unicorn)	<a href="#">deb Package</a>
Ubuntu 12.10 (Quantal Quetzal)	<a href="#">deb Package</a>
Ubuntu 12.04 (Precise Pangolin)	<a href="#">deb Package</a>
Ubuntu 10.04 LTS (Lucid Lynx)	<a href="#">deb Package</a>
Debian 8.0 (Jessie)	<a href="#">deb Package</a>

**Nota:** la versión utilizada para este tutorial es la última provista por Oracle a la fecha de realización de este instructivo (4.3.28).

La descarga comenzará.

Descargas

Hoy  
5 de jun. de 2015

VirtualBox-4.3.28-100309-Win.exe 1.471 KB/s - 10,1 MB de 106 MB, Queda 1 minuto  
<http://dlc-cdn.sun.com/virtualbox/4.3.28/VirtualBox-4.3.28-100309-Win.exe>  
Detener Cancelar

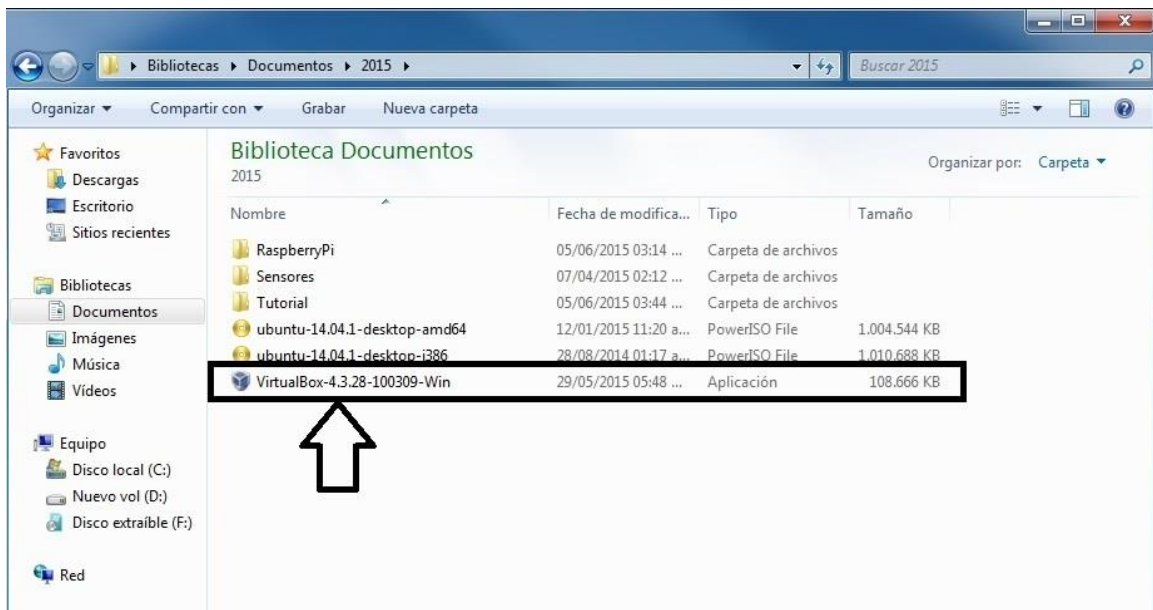


## 7.5. Instalación y configuración de software

En esta sección se instalará todo el software necesario para obtener el entorno de desarrollo para Raspberry Pi. Primero se instalará el software de virtualización Oracle VM VirtualBox, después el sistema operativo Ubuntu. Luego, dentro de Ubuntu, se instalará la toolchain, Java y el IDE Eclipse.

### 7.5.1. Instalación de Oracle VM VirtualBox

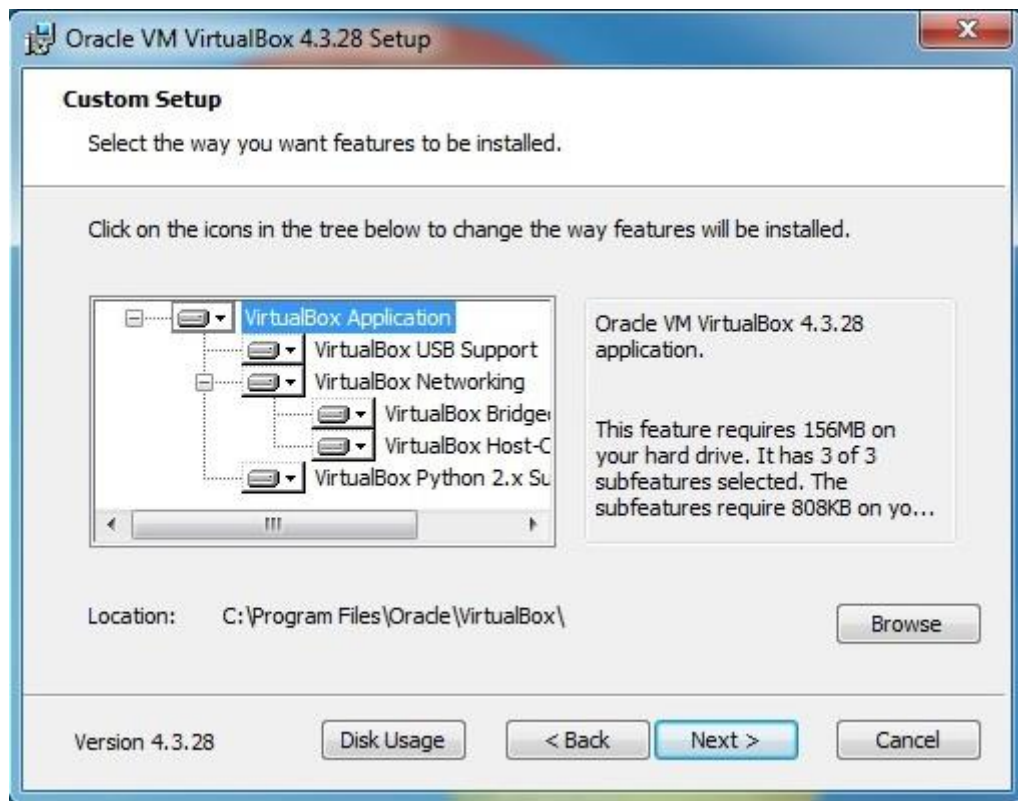
Una vez finalizada la descarga del Oracle VM VirtualBox, nos dirigimos al directorio donde se encuentra el archivo descargado y lo abrimos.



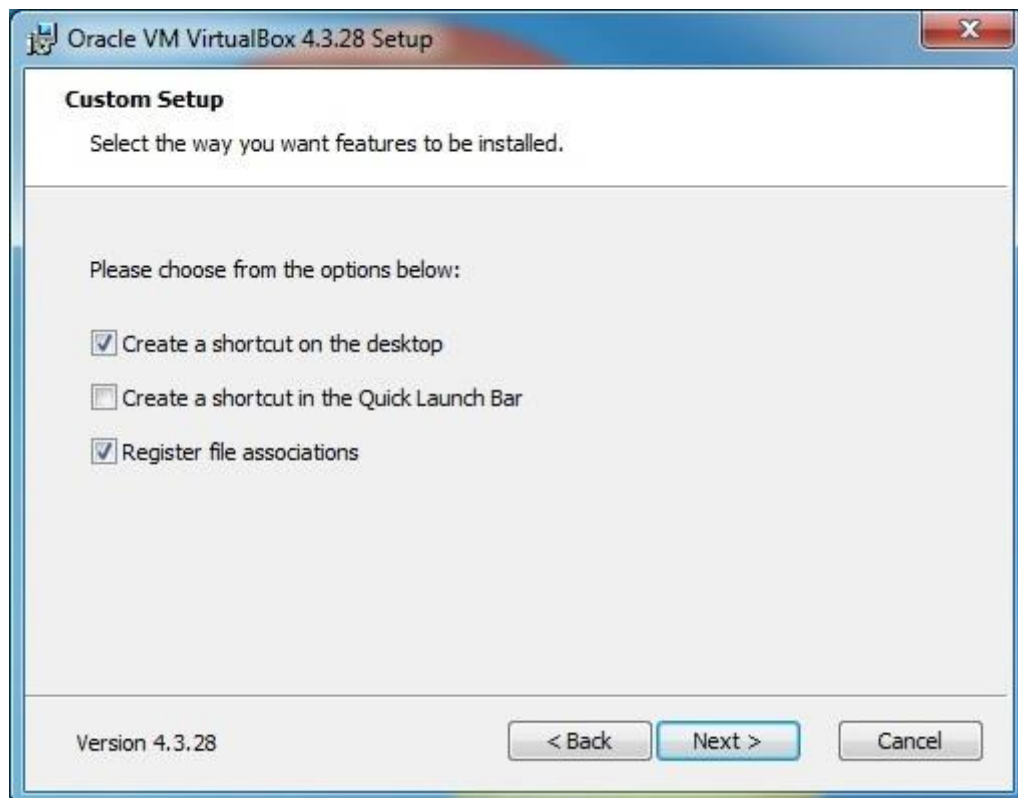
Aparecerá el asistente de instalación, pulsamos “Next”.



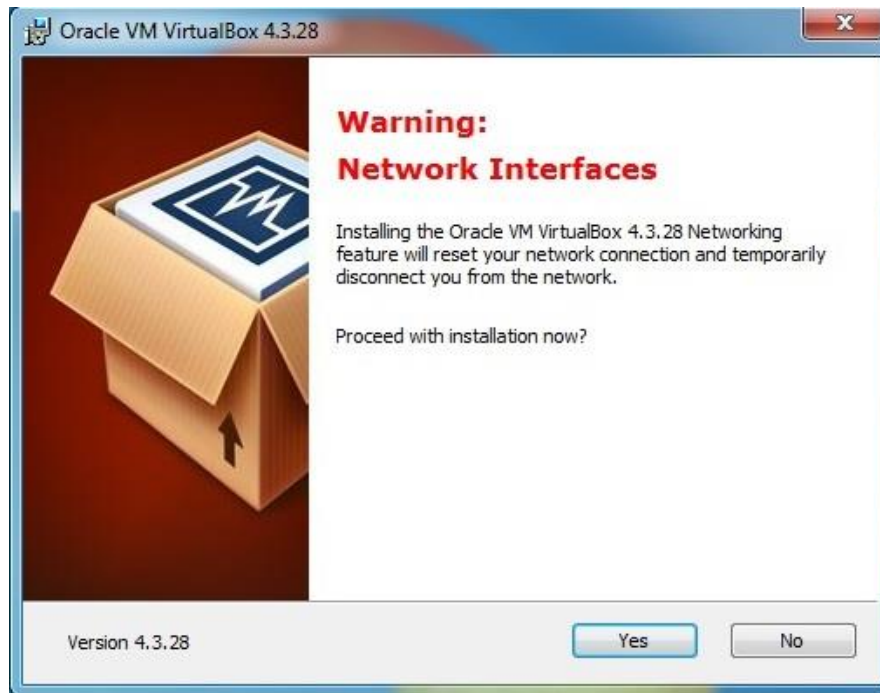
Luego, sin modificar nada, pulsamos “Next”.



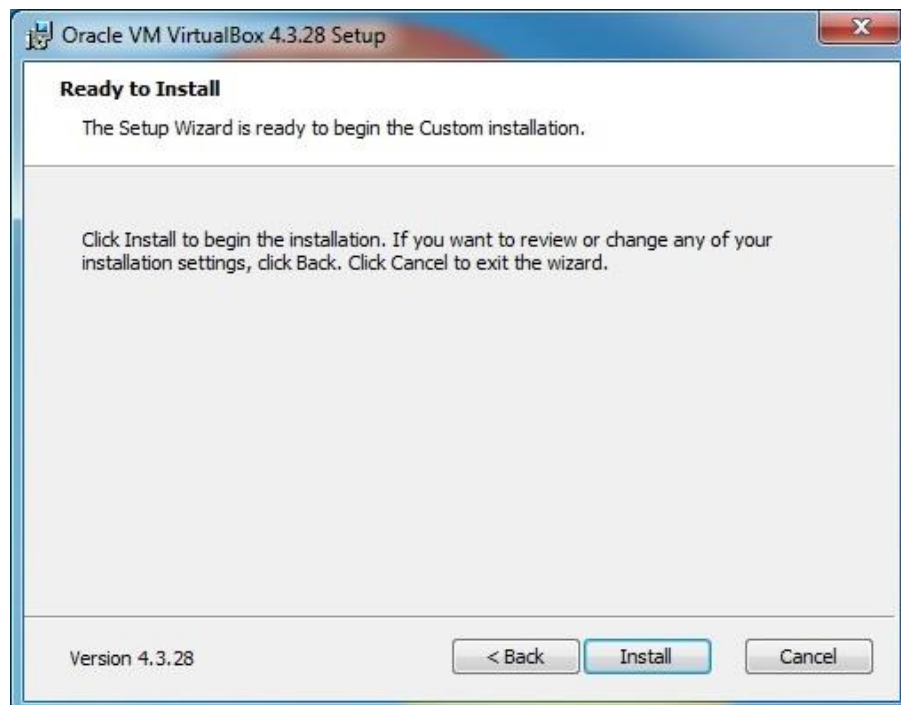
Seleccionamos para que se cree un acceso directo en el escritorio y para que se asocien los tipos de archivos conocidos. Pulsamos “Next”.



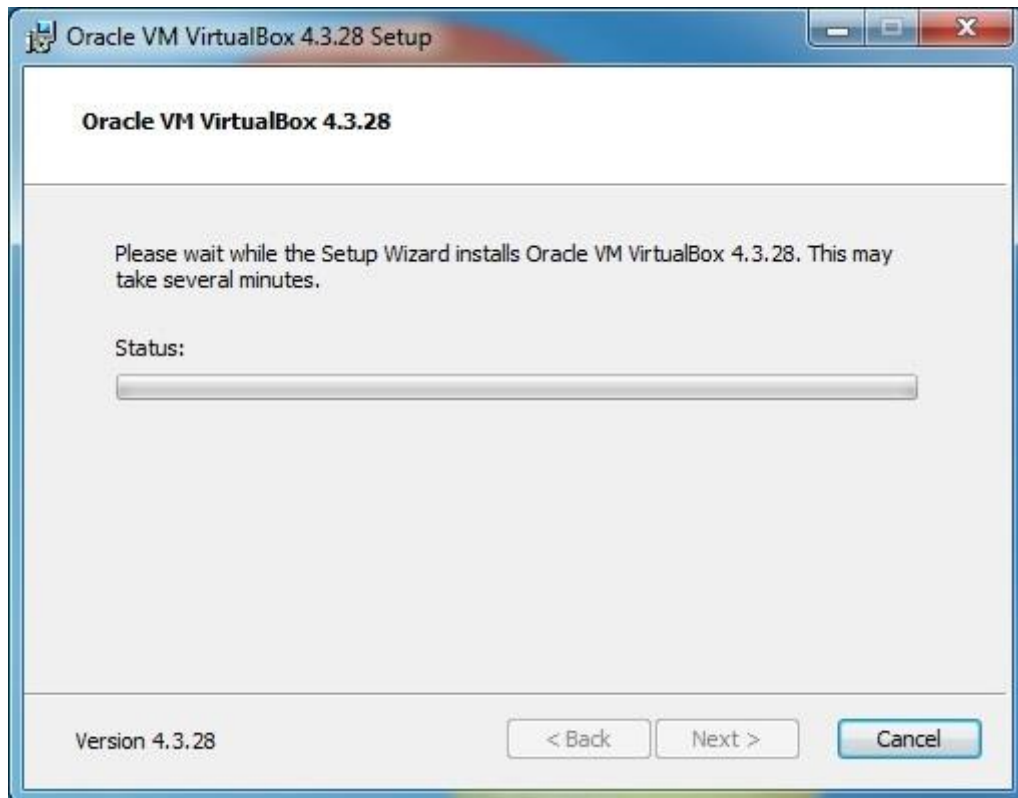
Nos advierte que se puede perder la conexión a Internet durante el proceso de instalación, es recomendable guardar los trabajos que tengamos en el navegador y cerrarlo. Luego pulsamos “Yes”.



Pulsamos “Install”.



El proceso de instalación comenzará, se puede demorar algunos minutos.

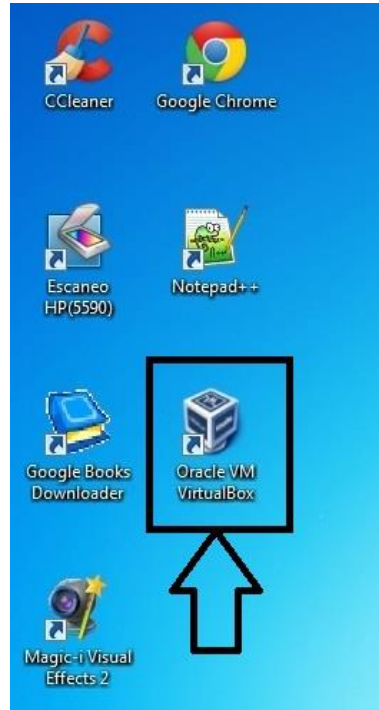


Una vez que finalice, pulsamos en "Finish".

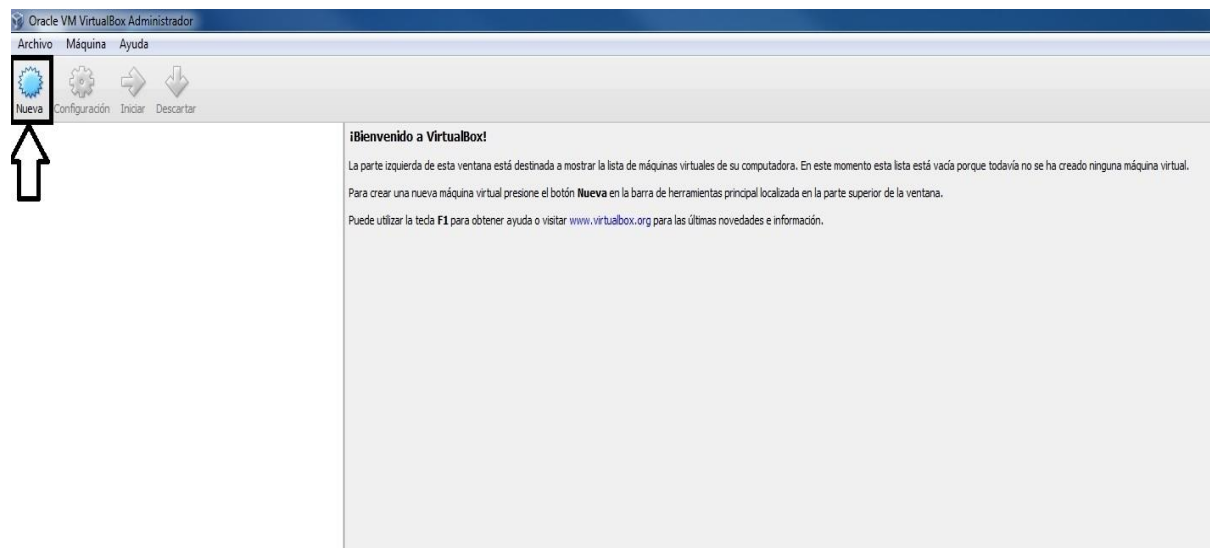


## 7.6. Configuración de Oracle VM VirtualBox

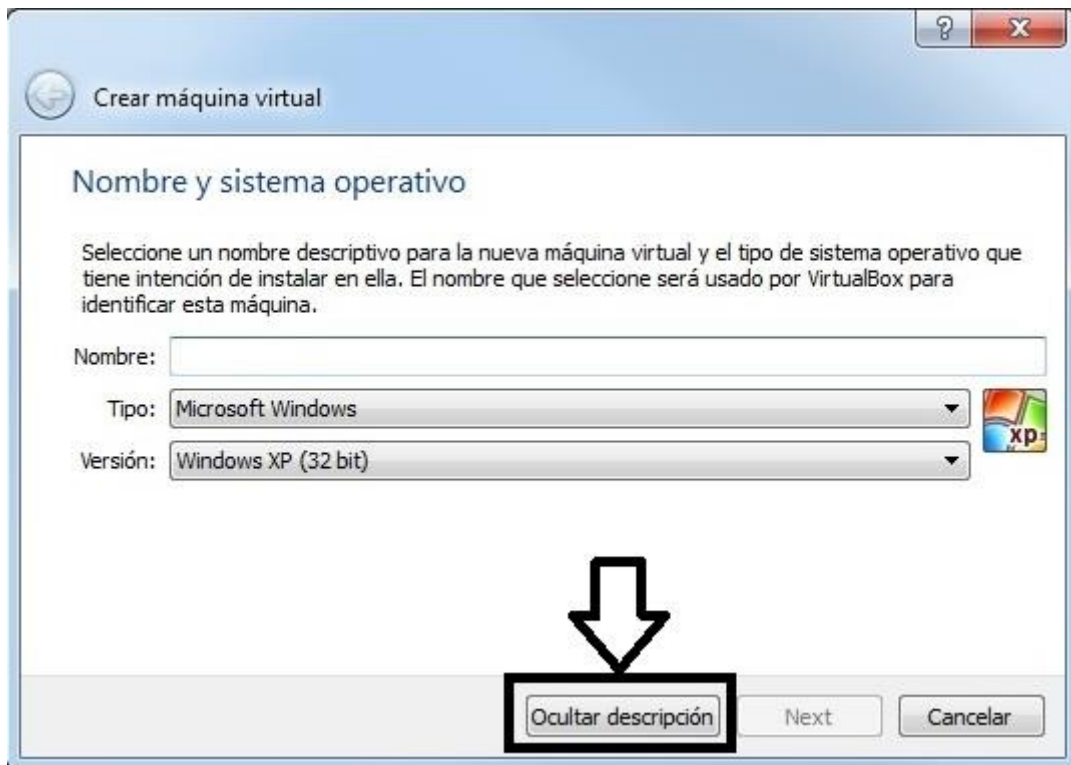
Ahora vamos a configurar la máquina virtual sobre la cual luego se instalará Ubuntu. Localizamos el acceso directo de Oracle VM VirtualBox en el escritorio y abrimos el programa.



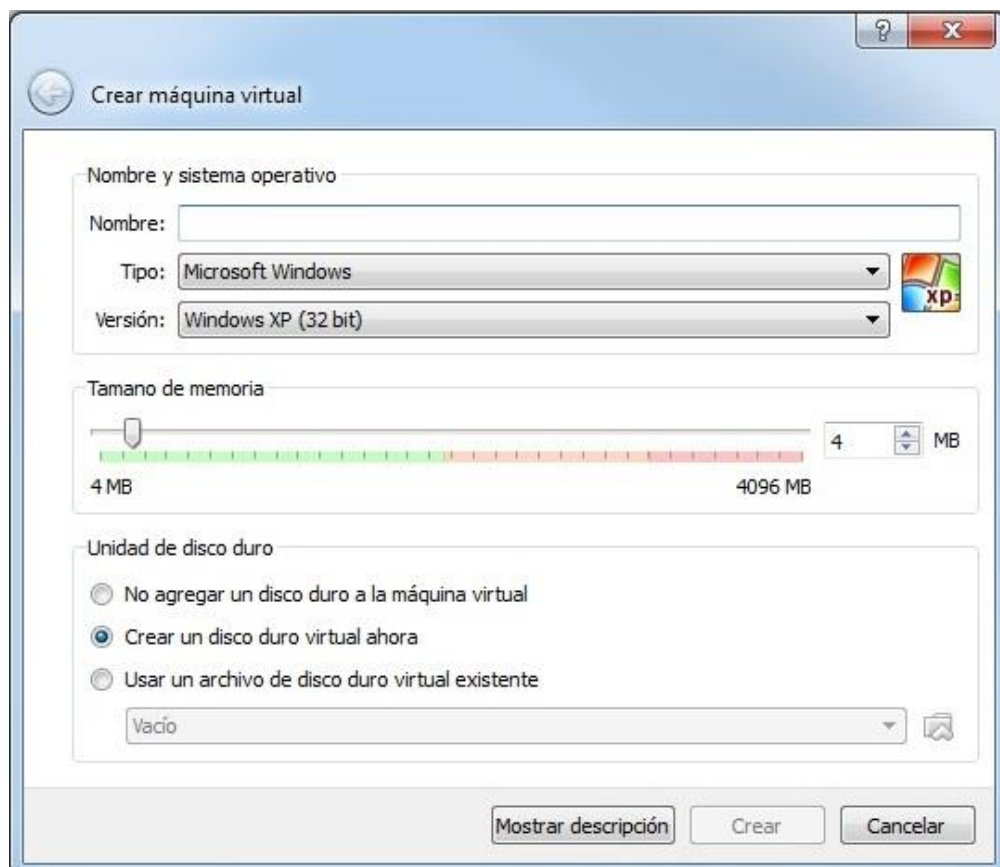
Aparecerá la siguiente pantalla, pulsamos en “Nueva”.



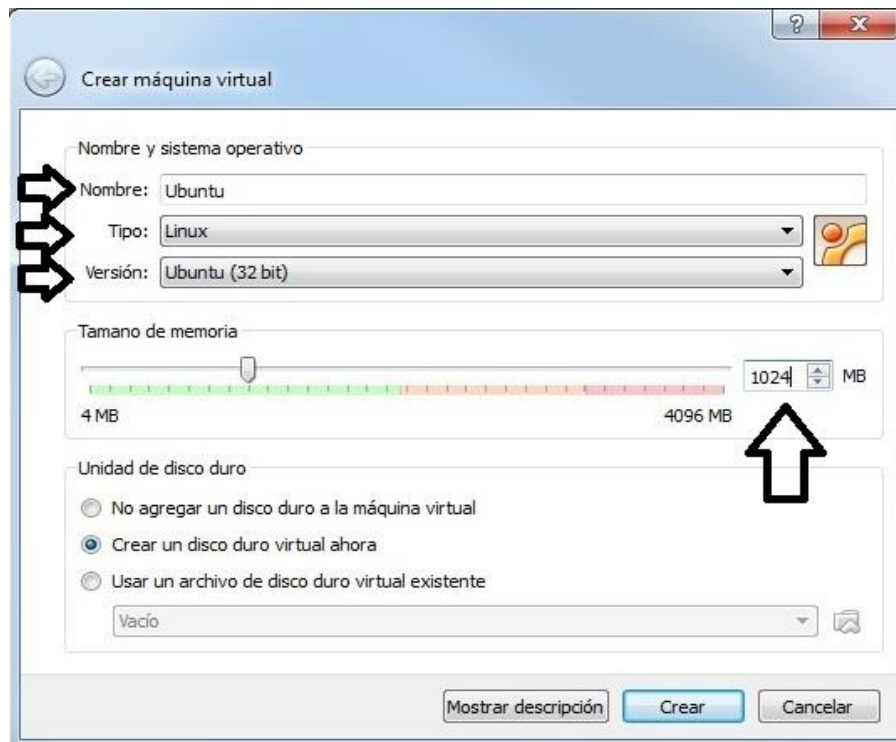
Pulsamos en “Ocultar descripción”.



Aparecerá la siguiente pantalla.

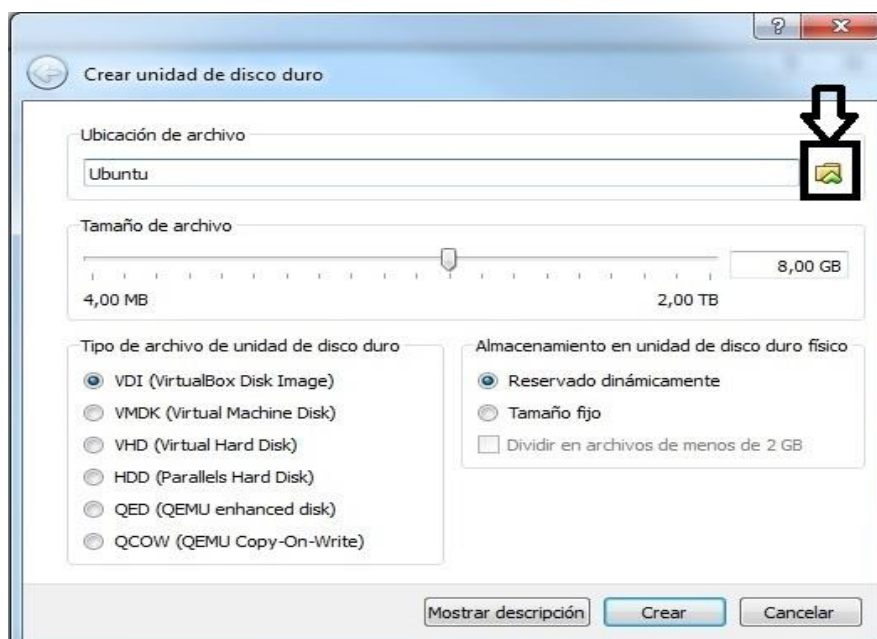


Modificamos el nombre, cambiamos el tipo y versión del sistema operativo, y el tamaño de memoria como indica la siguiente figura.

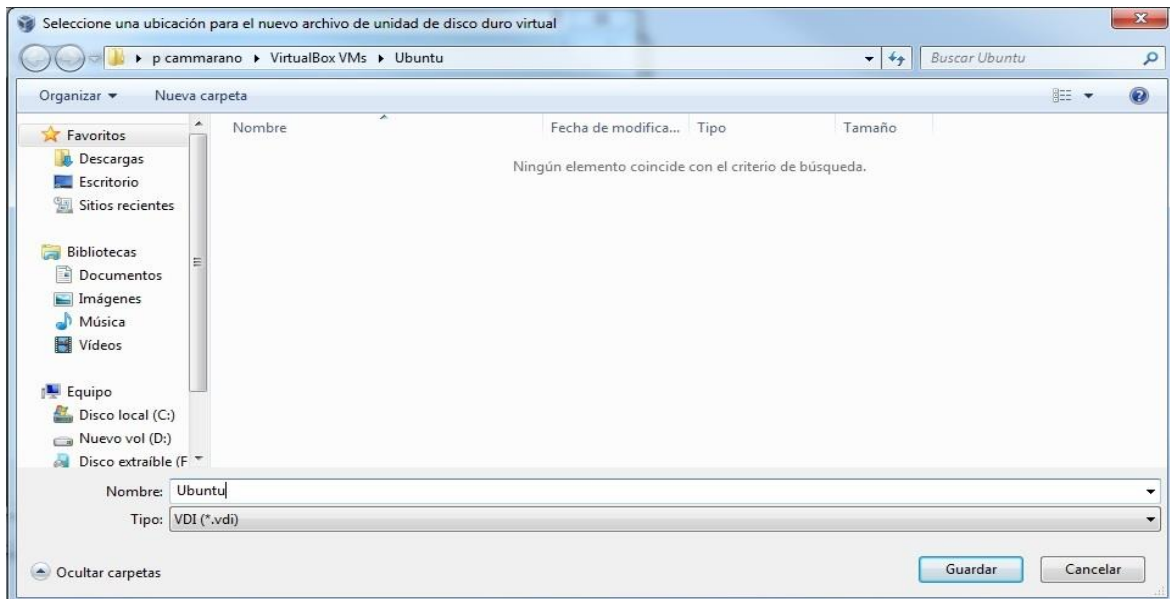


**Nota:** se recomienda como mínimo 1GB de memoria, pero se le puede asignar más en caso de disponerla.

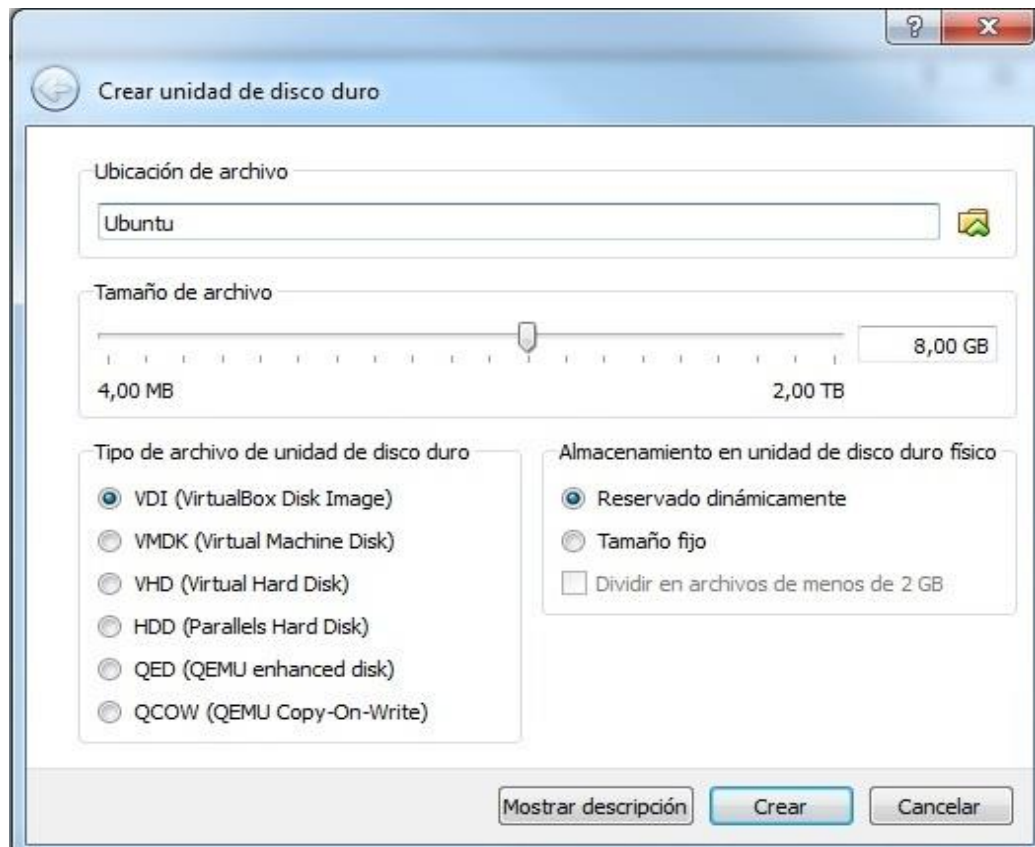
Luego presionamos en “Crear”, y aparecerá la siguiente pantalla. Pulsamos en el ícono que tiene una flecha verde.



Aparecerá la siguiente pantalla, donde elegimos la ubicación del archivo que representa al disco virtual. Es recomendable dejar la ubicación por defecto, que es la que aparece, seleccionamos “Guardar”.

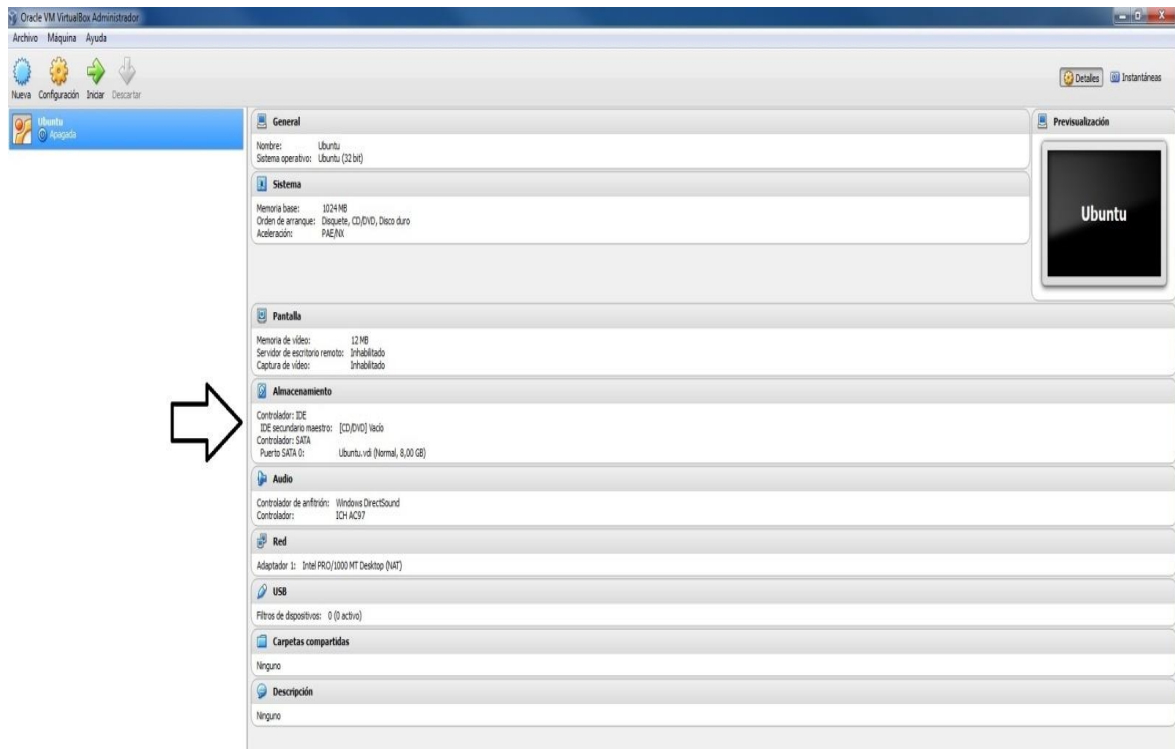


Se volverá a la pantalla anterior, seleccionamos 15 GB de tamaño de archivo (es el tamaño del disco virtual), el resto se deja como indica la siguiente figura, pulsamos “Crear”. En la figura se ven 8 GB, **pero se recomienda como mínimo 15 GB**.

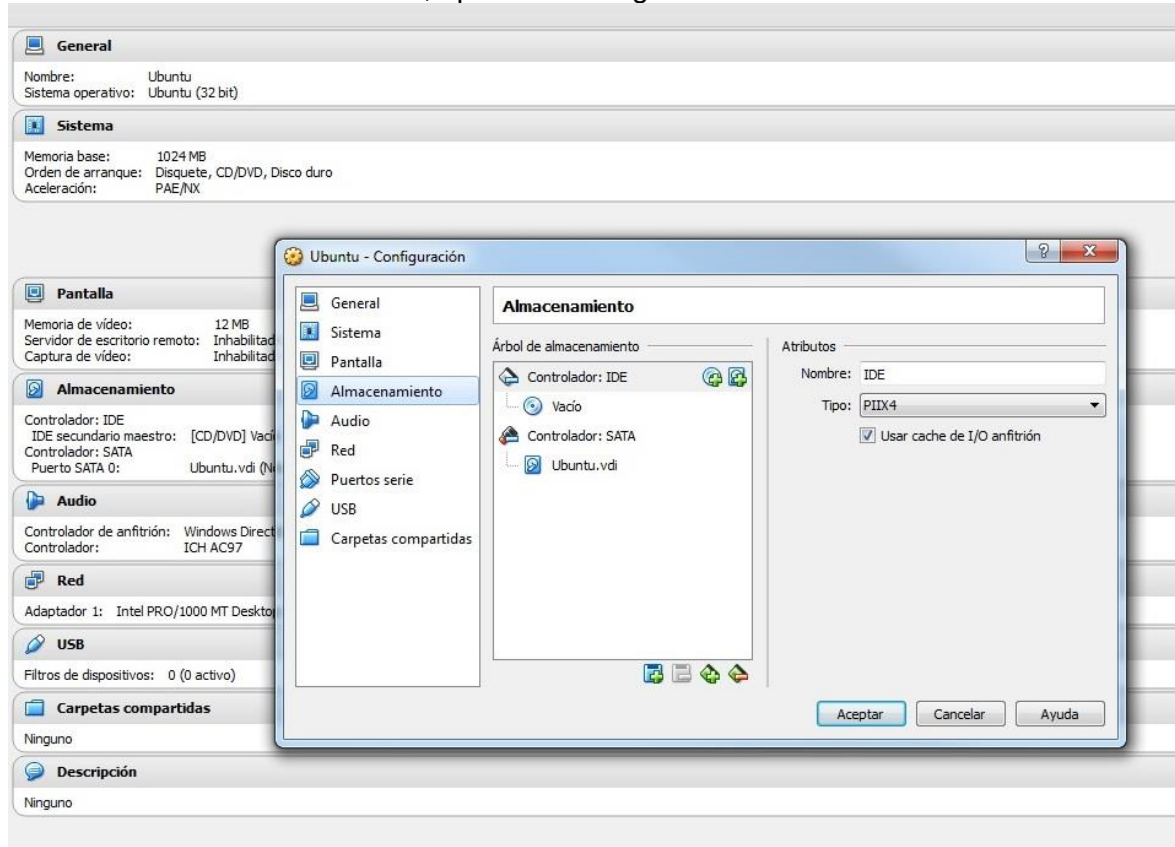




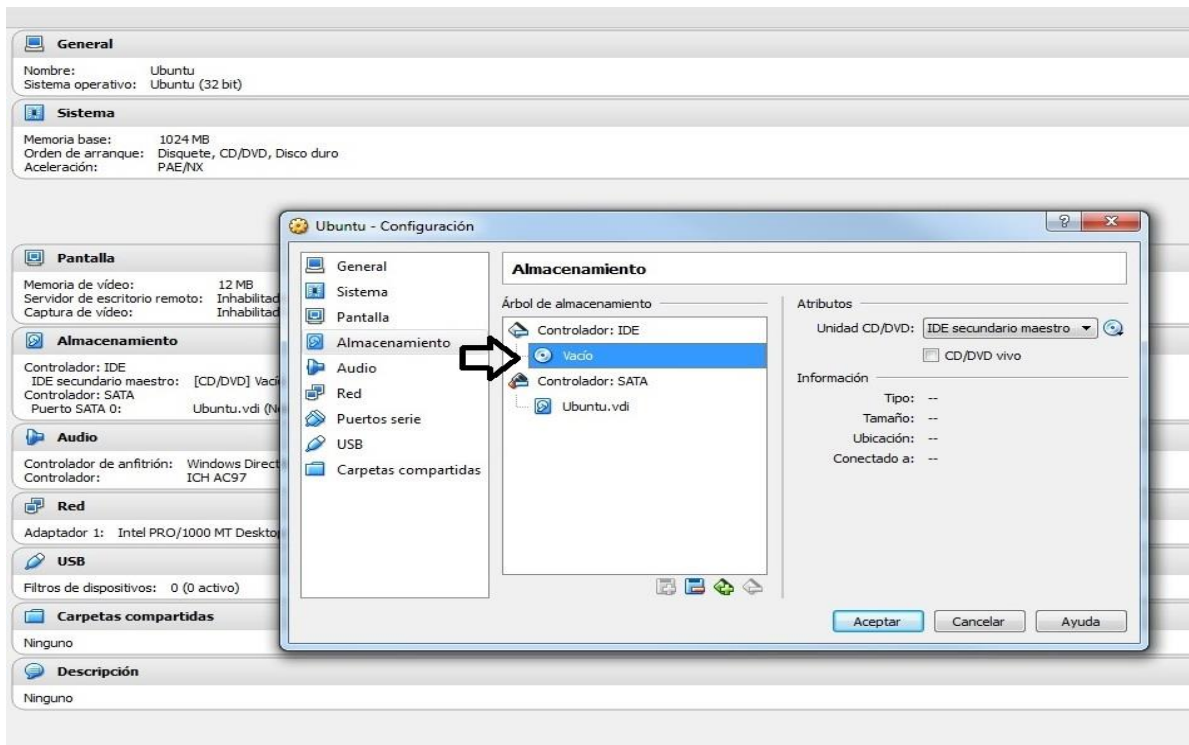
Luego se creará la máquina virtual, aparecerá la siguiente pantalla donde se ven las configuraciones de la máquina virtual recientemente creada. Por ahora solo modificaremos la que dice “Almacenamiento”.



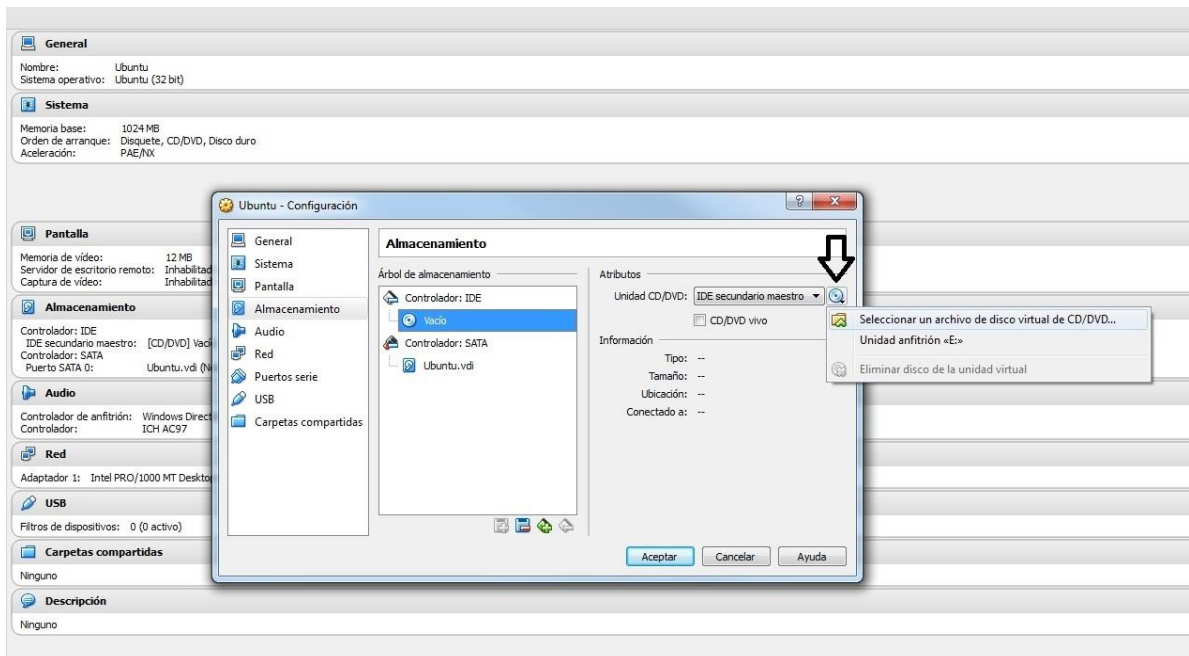
Pulsamos en “Almacenamiento”, aparecerá la siguiente ventana.



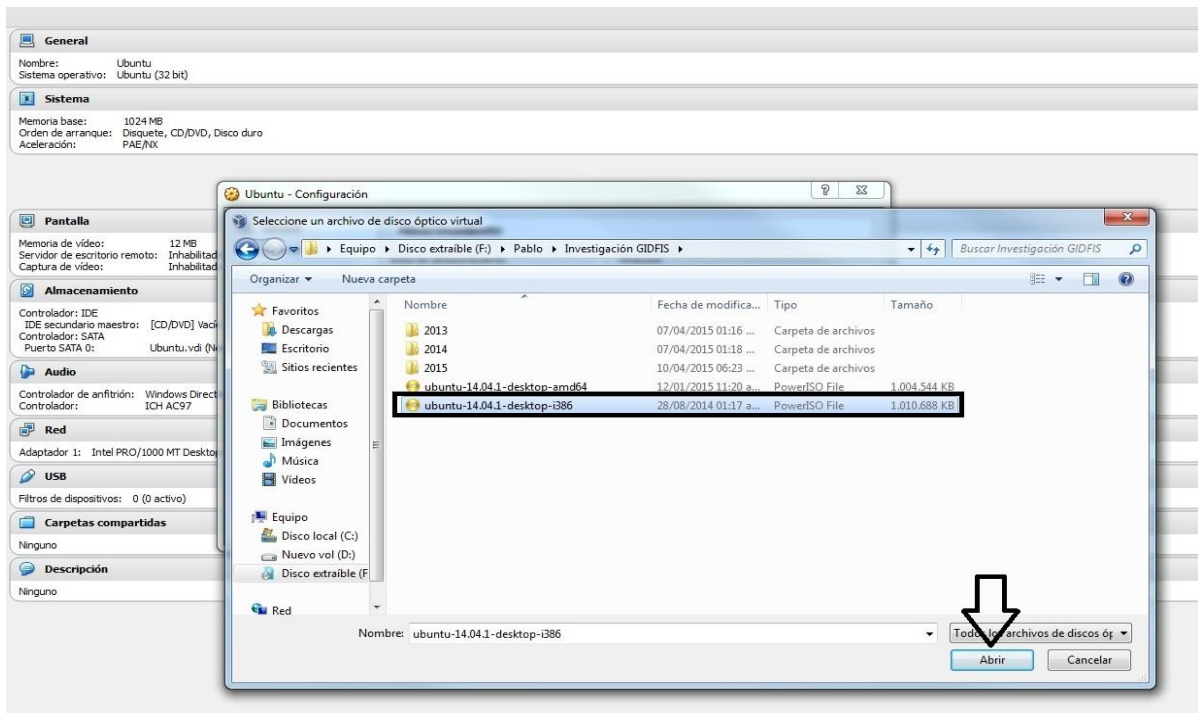
Lo que vamos a hacer es indicar que queremos que utilice la imagen de Ubuntu que descargamos al principio como unidad de CD/DVD. Esto simularía la acción de introducir el DVD de Ubuntu en la lectora. Pulsamos donde dice “Vacío”.



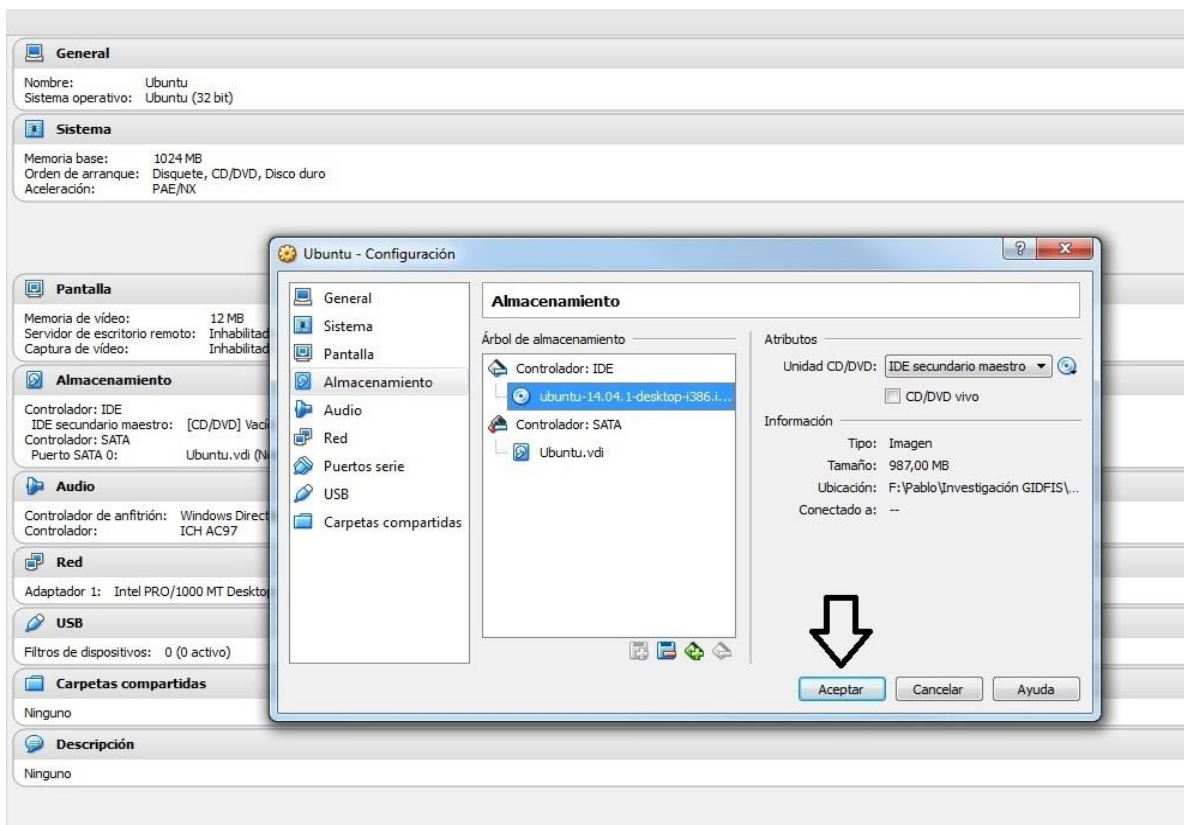
Luego presionamos el ícono del CD y elegimos la opción “Seleccionar un archivo...”.



Aparecerá una ventana para que busquemos la imagen de Ubuntu que descargamos al principio. Nos dirigimos al directorio donde está la descarga, seleccionamos la imagen y pulsamos “Abrir”.

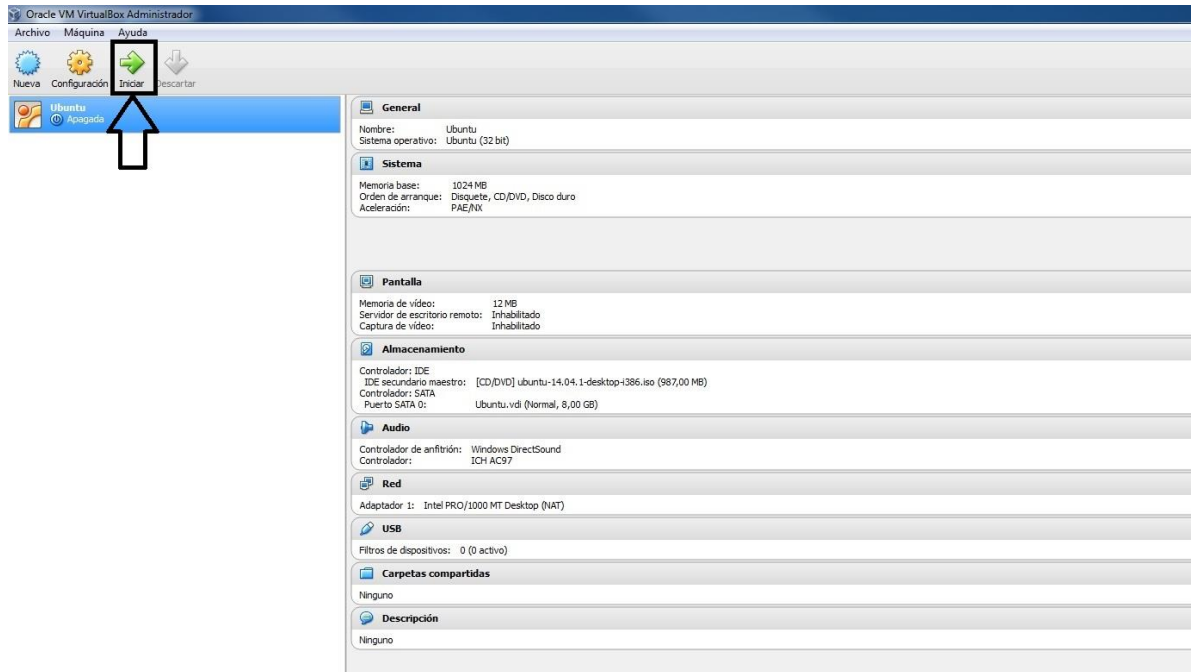


Volveremos a la otra ventana, pero ahora aparece cargada la imagen de Ubuntu, presionamos “Aceptar”.

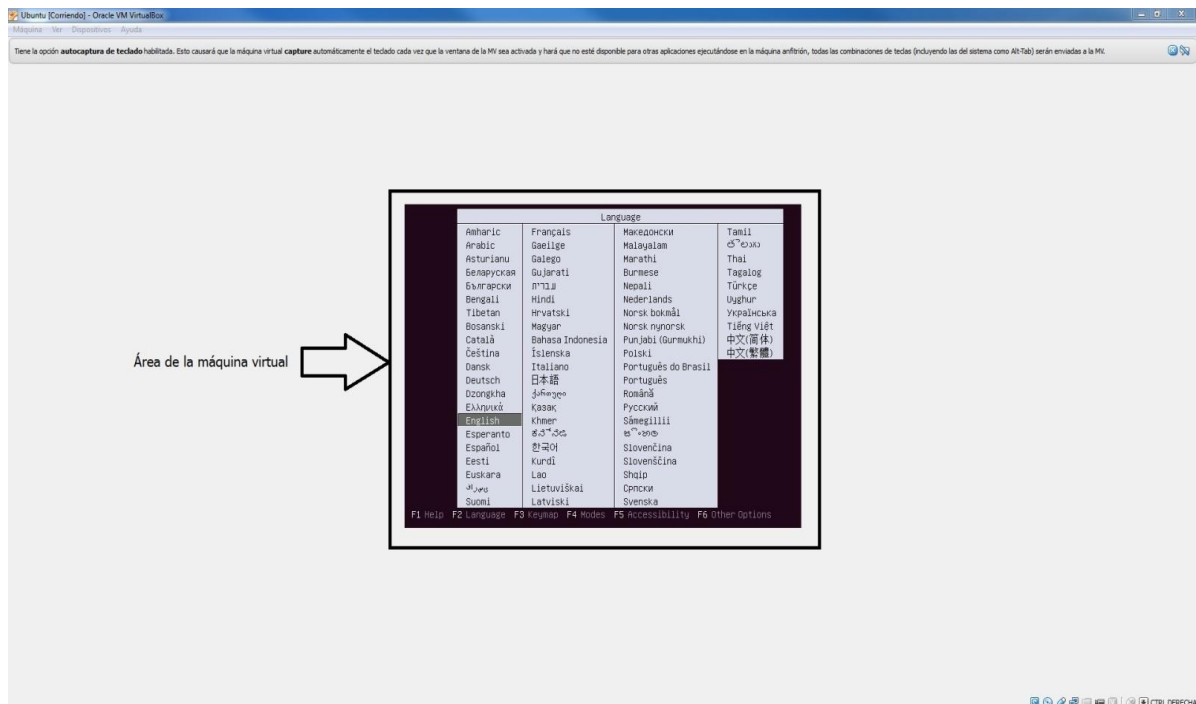


## 7.7. Instalación de Ubuntu

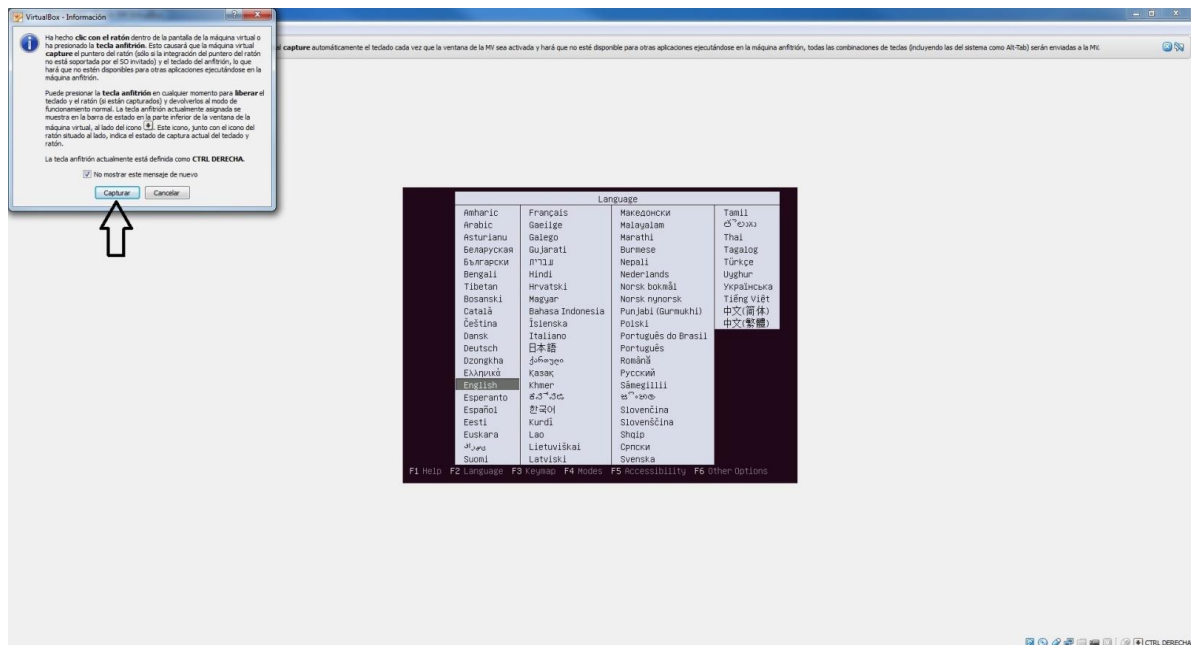
Ya terminamos con la configuración básica de la máquina virtual, ahora vamos a instalar Ubuntu en ella. Estando en la pantalla principal, pulsamos en “Iniciar”.



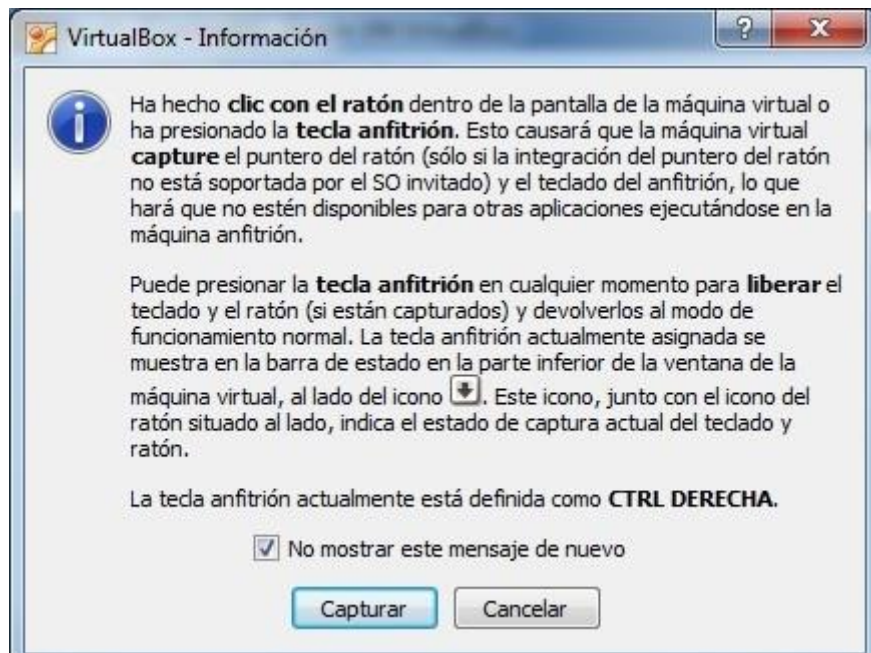
Se abrirá otra ventana donde se iniciará la máquina virtual. Aparecerá una pantalla de selección de idioma.



Para que la máquina virtual capture los eventos del mouse y teclado debemos hacer click sobre el área correspondiente a la máquina virtual. Aparecerá un cartel informándonos de que la máquina virtual va a capturar los eventos del teclado y mouse. Seleccionamos el “check box” para que no vuelva a mostrar el mensaje y pulsamos en “Capturar”.



Detalle del cartel.





Ahora la máquina virtual capturará los eventos del teclado, entonces utilizando las flechas seleccionamos el idioma español y apretamos “Enter”.



Aparecerá la siguiente pantalla, utilizando las flechas seleccionamos “Instalar Ubuntu” y apretamos “Enter”.



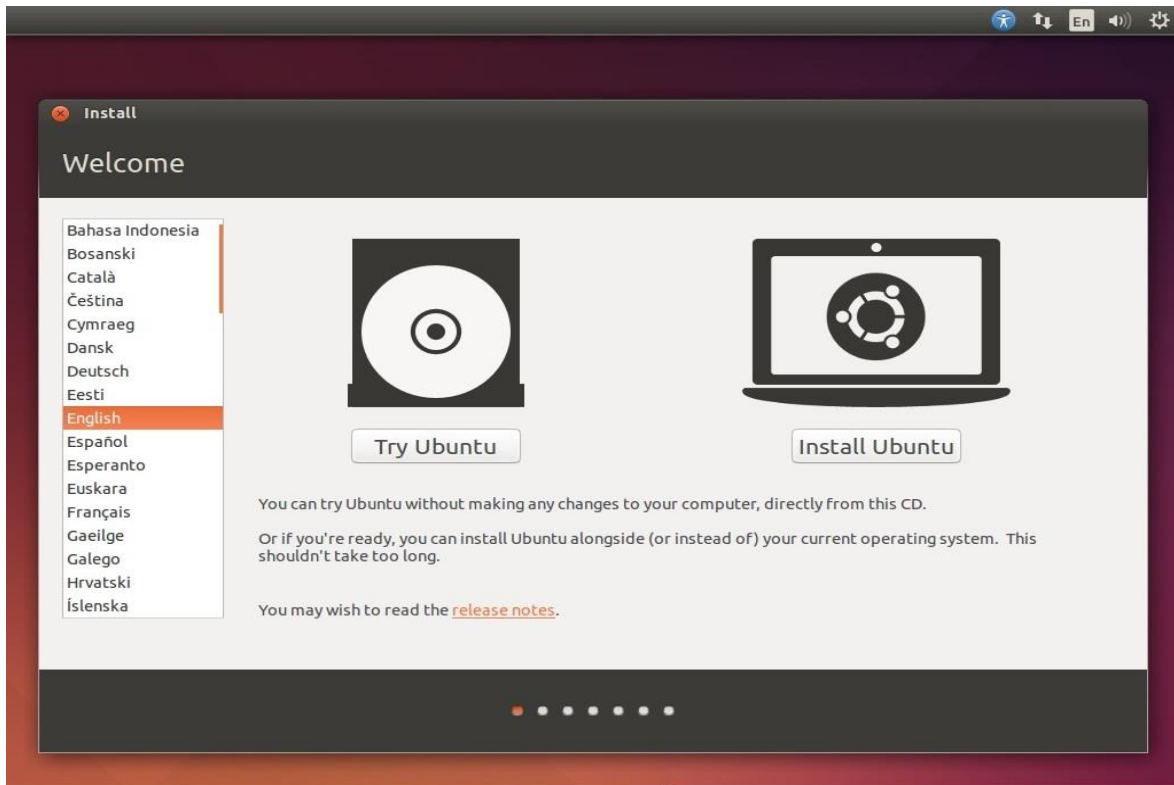
Luego aparecerá una pantalla de carga.



Si aparece algún mensaje como el que muestra la siguiente figura, no pasa nada, luego de unos segundos terminará la carga.



Aparecerá la pantalla del instalador.

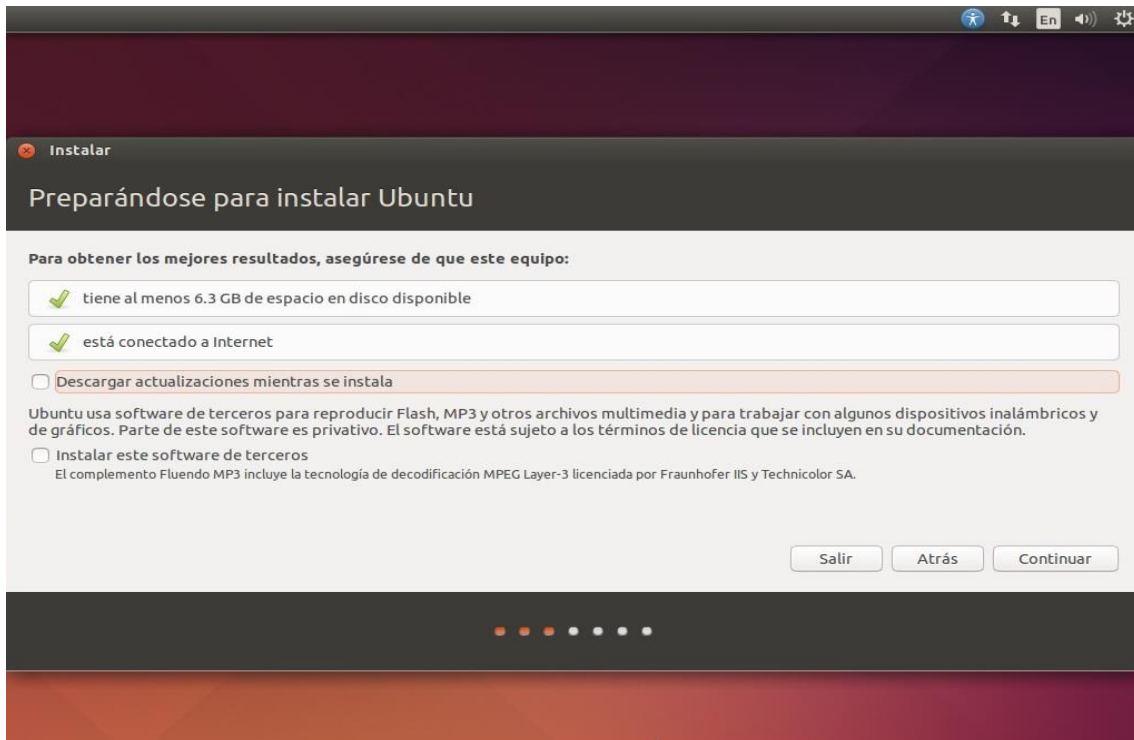


Ahora ya podemos utilizar el mouse, seleccionamos “Español” y luego en “Instalar Ubuntu”.

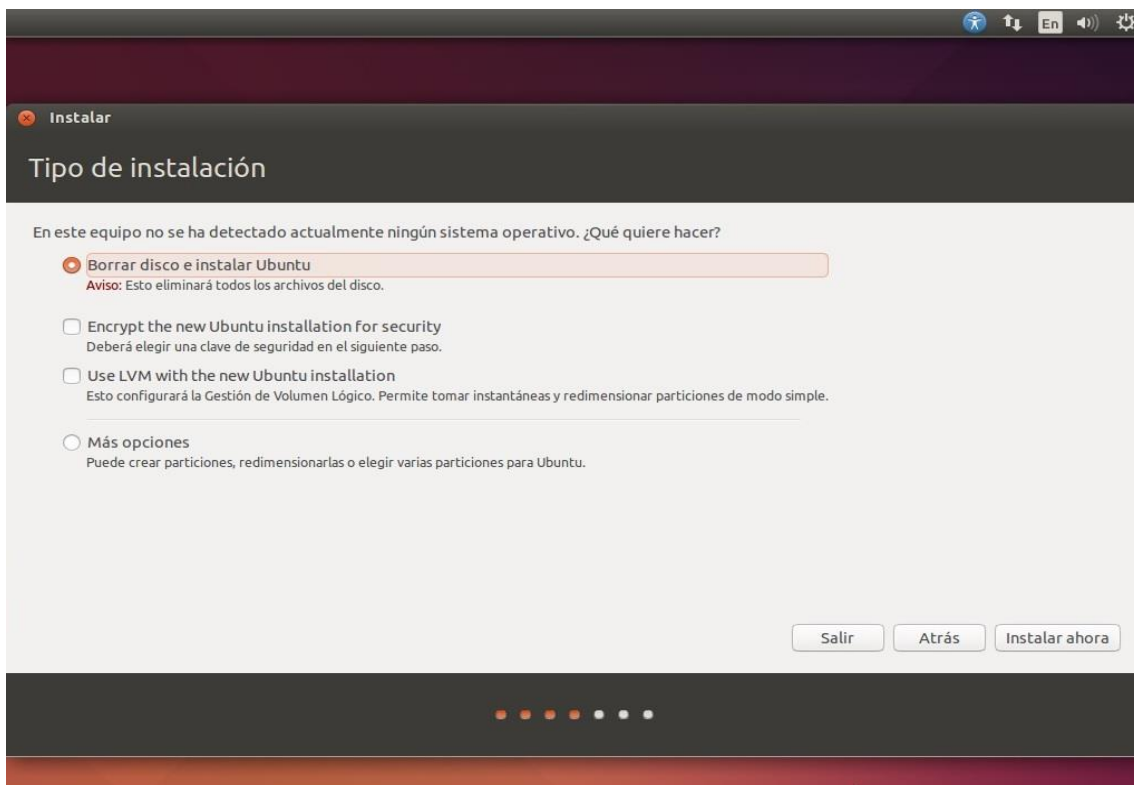


No tildamos ninguna de las dos opciones y seleccionamos “Continuar”.



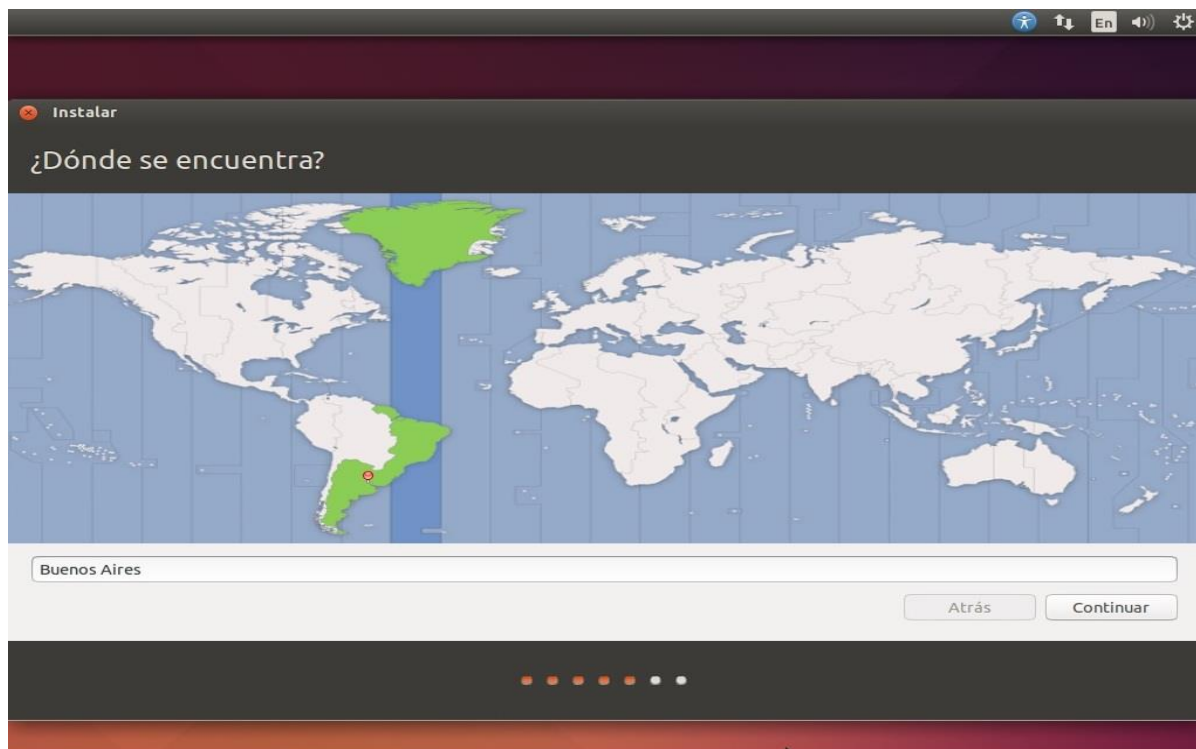


Seleccionamos “Borrar disco e instalar Ubuntu” y pulsamos en “Instalar ahora”:

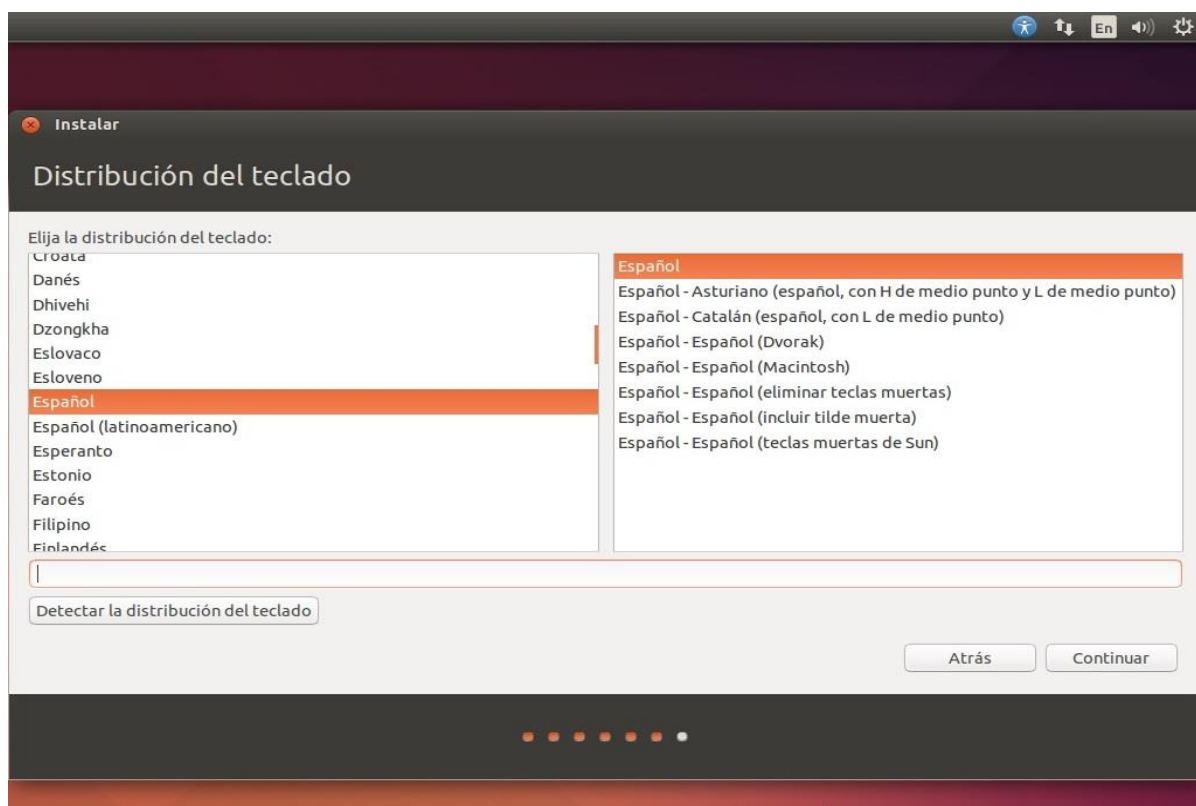


**Nota:** cuando dice “Borrar disco” se refiere al disco de la máquina virtual, no al de nuestro equipo.

Seleccionamos nuestra ubicación.

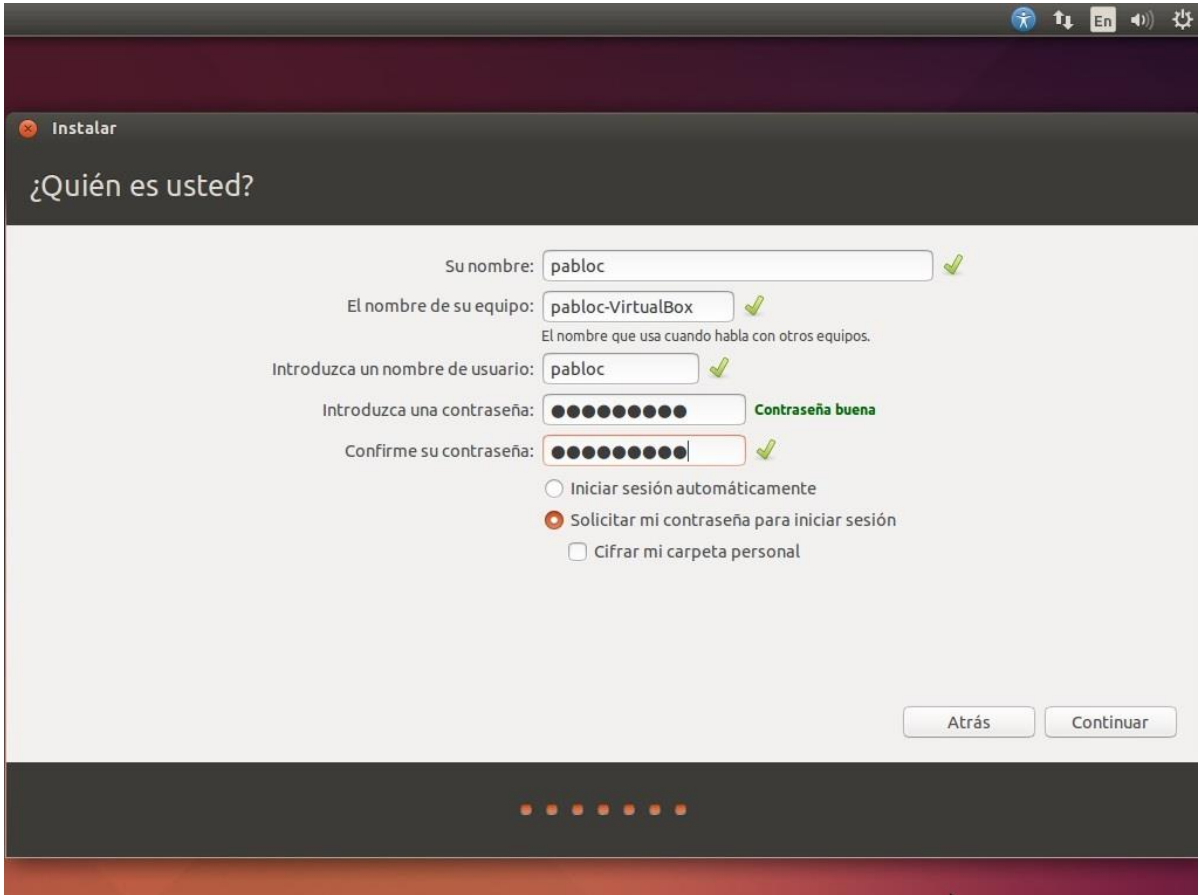


Seleccionamos el tipo de teclado, podemos probar que funcionen todos los caracteres escribiendo un texto de prueba, luego pulsamos en “Continuar”.

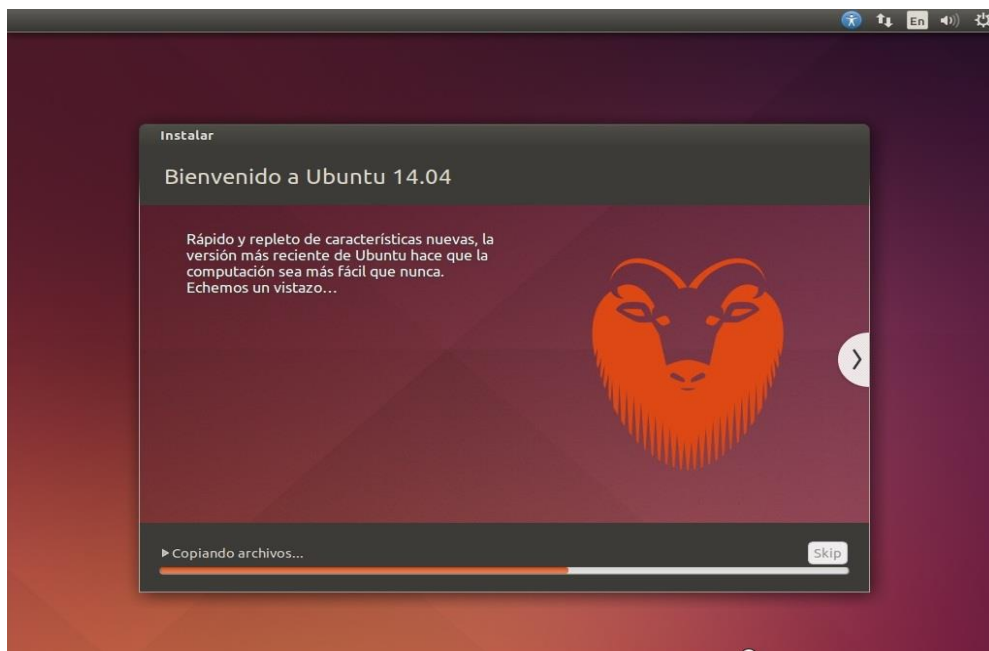




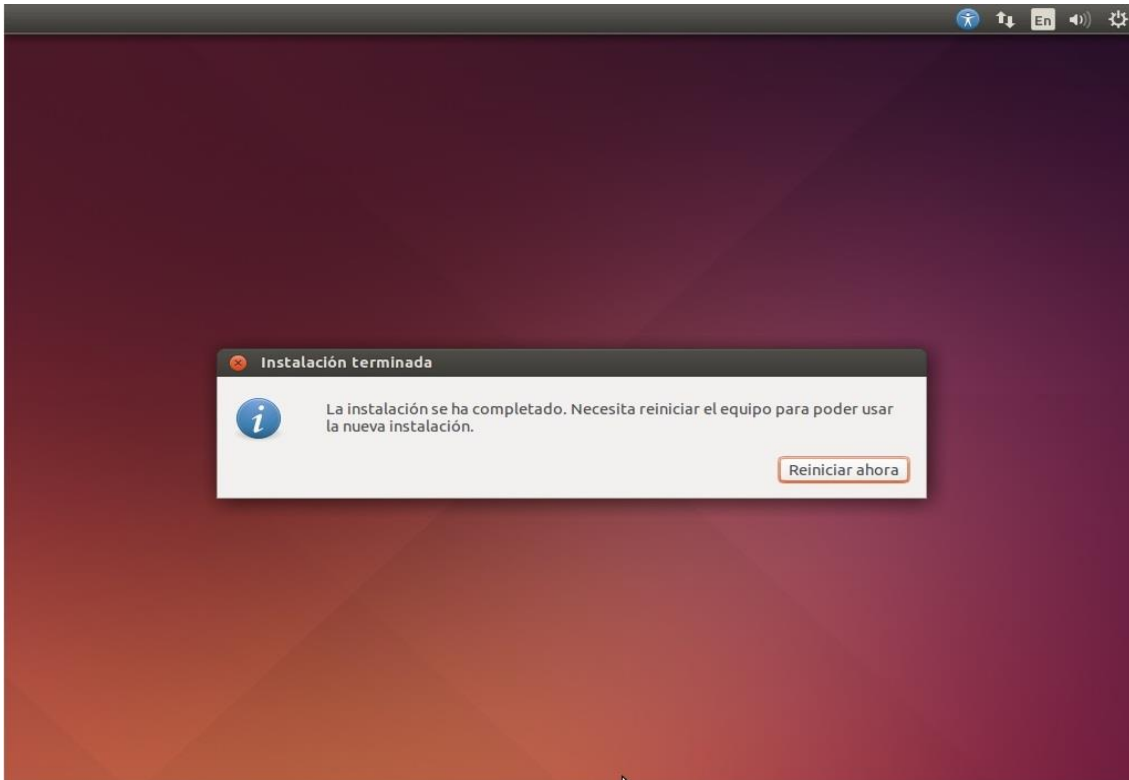
Ingresamos nuestro nombre, se completará automáticamente el nombre de equipo y de usuario. Luego ingresamos una contraseña, es importante recordarla porque se usará bastante, no es solo para iniciar sesión en el equipo. Podemos optar por iniciar sesión automáticamente o por solicitar una contraseña para ello.



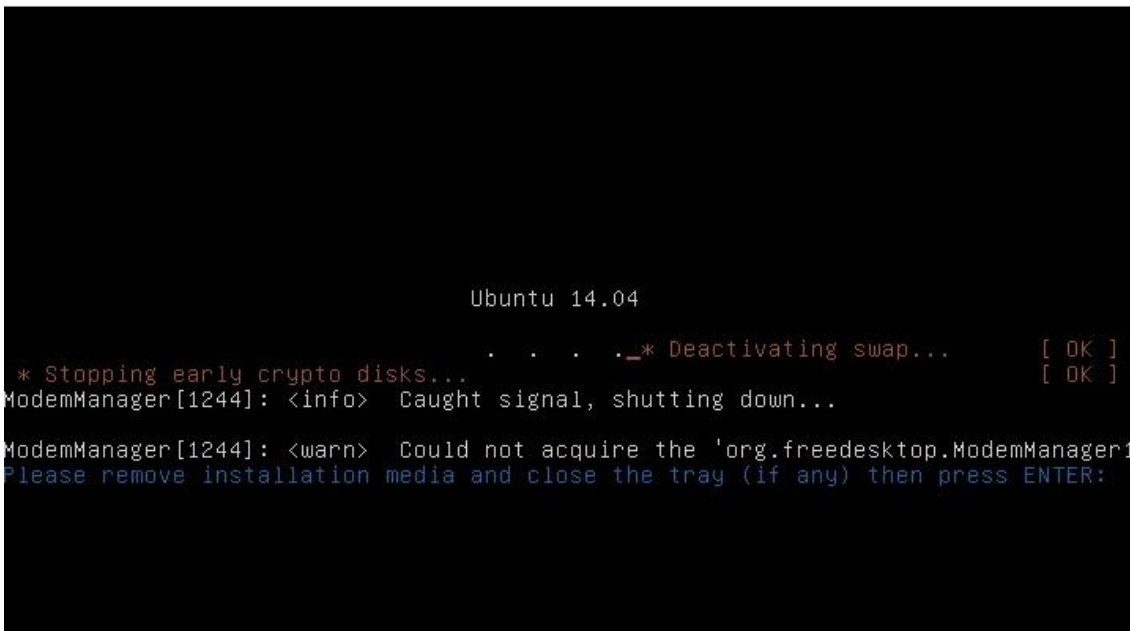
Luego comenzará la instalación, puede demorar unos minutos.



Cuando finalice, pedirá reiniciar el equipo.



Aparecerá la siguiente pantalla, presionamos “Enter”.

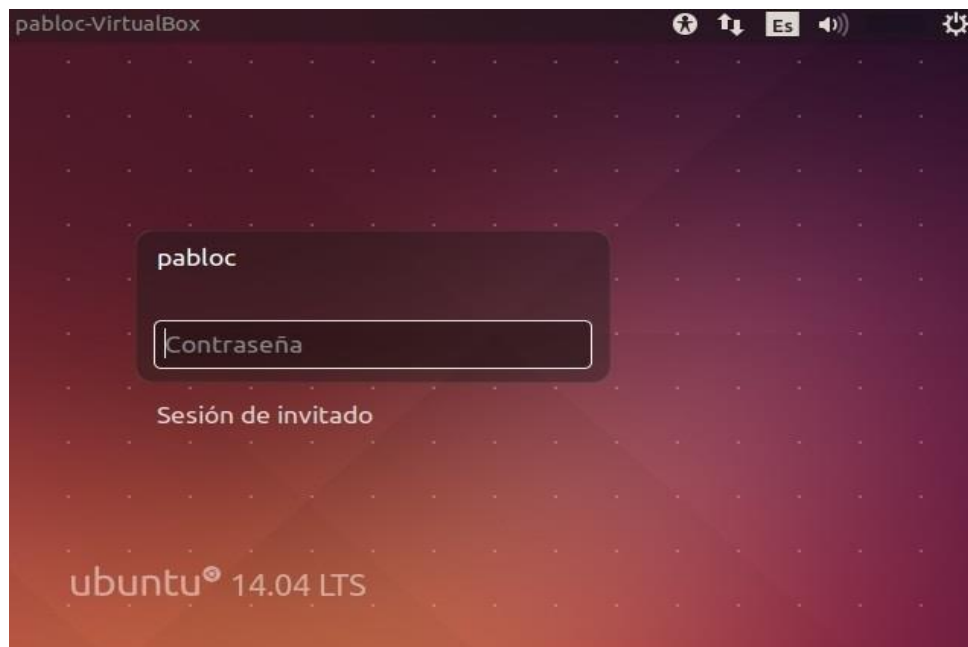


Luego se reiniciará, pueden aparecer algunos mensajes como los que se muestran en la siguiente figura pero no pasa nada. Puede tardar unos segundos en iniciar.



```
[ 11.024176] piix4_smbus 0000:00:07.0: SMBus base address uninitialized - upgr  
ade BIOS or use force_addr=0xaddr  
[ 12.305757] intel_rapl: no valid rapl domains found in package 0
```

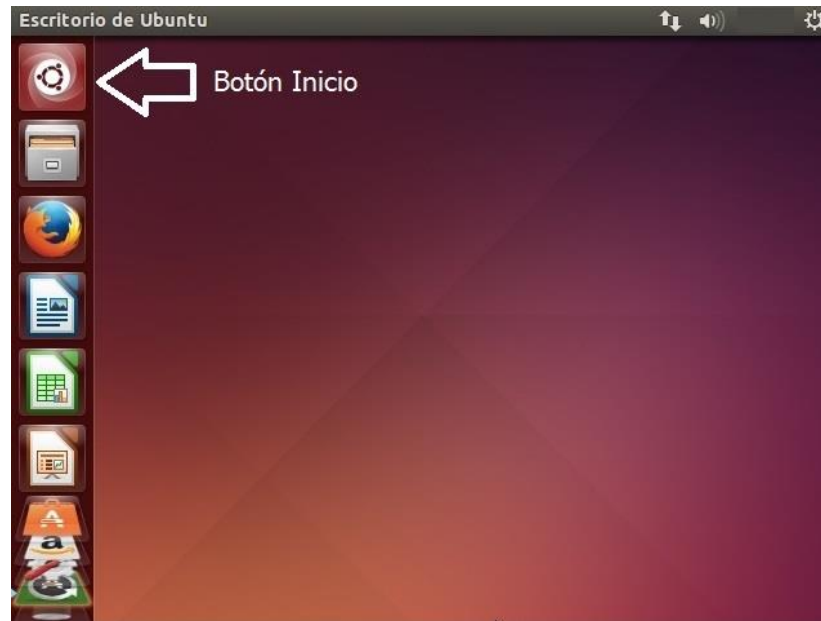
Cuando inicie, aparecerá la pantalla de inicio de sesión, ingresamos la contraseña y pulsamos “Enter”.



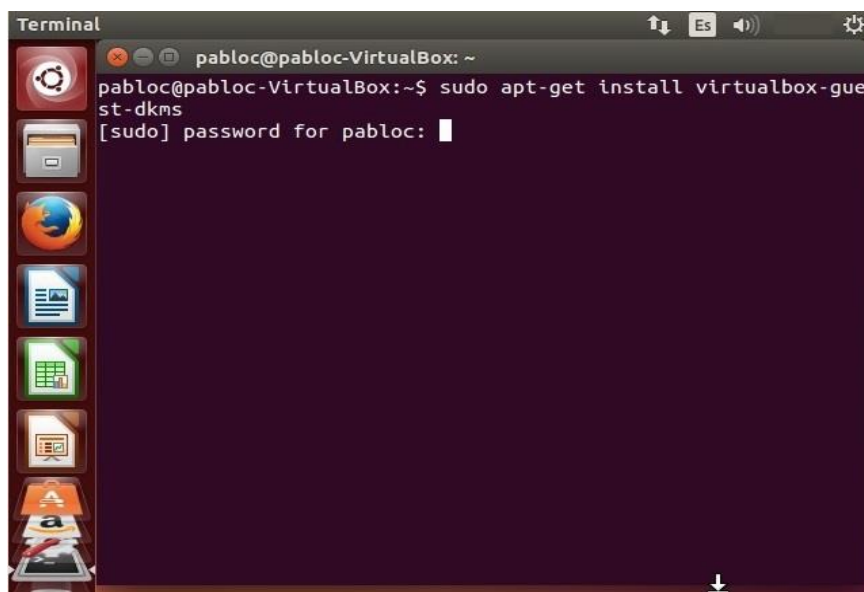
Luego se mostrará el escritorio. Con esto finaliza la instalación, ahora se procederá a la configuración de Ubuntu.

## 7.8. Configuración de Ubuntu

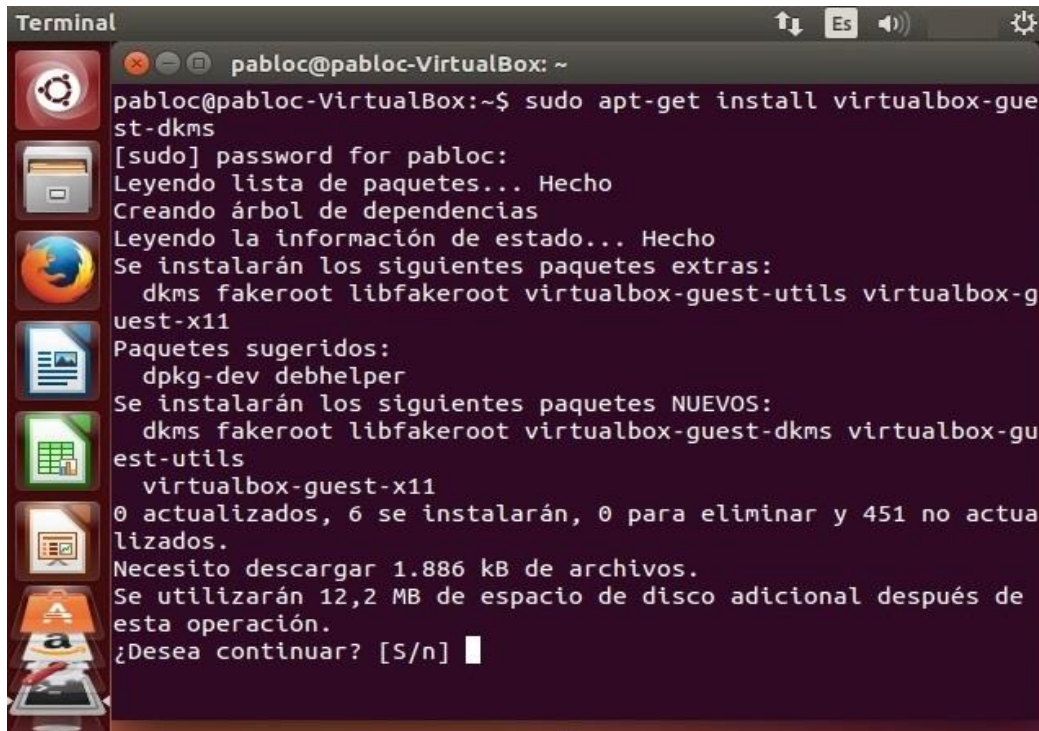
Finalizamos con la instalación, ahora estamos en el escritorio de Ubuntu, como podemos observar en la figura, el tamaño de la pantalla es realmente chico, y si queremos cambiar la resolución por una mayor no podremos. Para ello hay que instalar: “VirtualBox Guest Tools”, que es un conjunto de herramientas que permite “integrar” el sistema operativo guest con la máquina virtual.



Presionamos “CTRL+ALT+T” para abrir la consola (también llamada Terminal). Otra manera es presionar el botón de inicio, escribir “Terminal” y ejecutarla desde allí. Una vez en la Terminal, escribimos “sudo apt-get install virtualbox-guest-dkms” y pulsamos “Enter”. Nos pedirá la contraseña, la ingresamos y pulsamos “Enter” nuevamente.

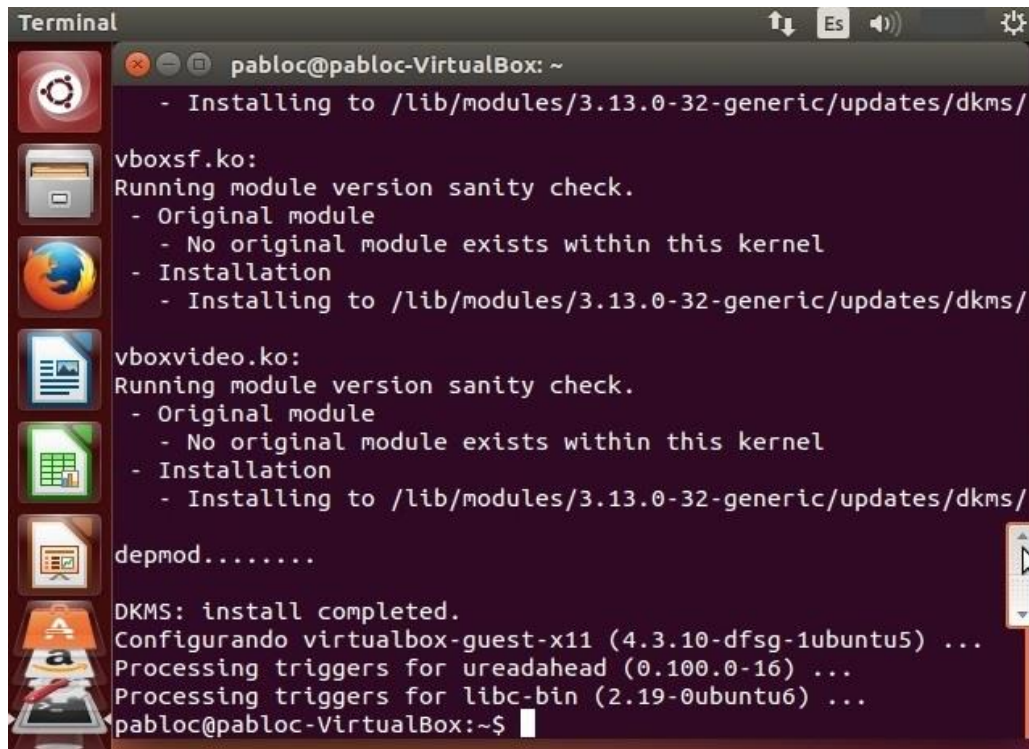


Aparecerán una serie de mensajes, luego ingresamos “S” y pulsamos “Enter” para continuar con la instalación.



```
Terminal
pabloc@pabloc-VirtualBox: ~
pabloc@pabloc-VirtualBox:~$ sudo apt-get install virtualbox-guest-dkms
[sudo] password for pabloc:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  dkms fakeroot libfakeroot virtualbox-guest-utils virtualbox-guest-x11
Paquetes sugeridos:
  dpkg-dev debhelper
Se instalarán los siguientes paquetes NUEVOS:
  dkms fakeroot libfakeroot virtualbox-guest-dkms virtualbox-guest-utils
  virtualbox-guest-x11
0 actualizados, 6 se instalarán, 0 para eliminar y 451 no actualizados.
Necesito descargar 1.886 kB de archivos.
Se utilizarán 12,2 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

La instalación puede tardar unos segundos, una vez finalizada hay que apagar la máquina virtual.



```
Terminal
pabloc@pabloc-VirtualBox: ~
- Installing to /lib/modules/3.13.0-32-generic/updates/dkms/
vboxsf.ko:
Running module version sanity check.
- Original module
- No original module exists within this kernel
- Installation
- Installing to /lib/modules/3.13.0-32-generic/updates/dkms/
vboxvideo.ko:
Running module version sanity check.
- Original module
- No original module exists within this kernel
- Installation
- Installing to /lib/modules/3.13.0-32-generic/updates/dkms/
depmod.....
DKMS: install completed.
Configurando virtualbox-guest-x11 (4.3.10-dfsg-1ubuntu5) ...
Processing triggers for ureadahead (0.100.0-16) ...
Processing triggers for libc-bin (2.19-0ubuntu6) ...
pabloc@pabloc-VirtualBox:~$
```

Cerramos la terminal, pulsamos en el ícono del “engranaje” que está arriba a la derecha y se desplegará un menú, seleccionamos “Apagar”.



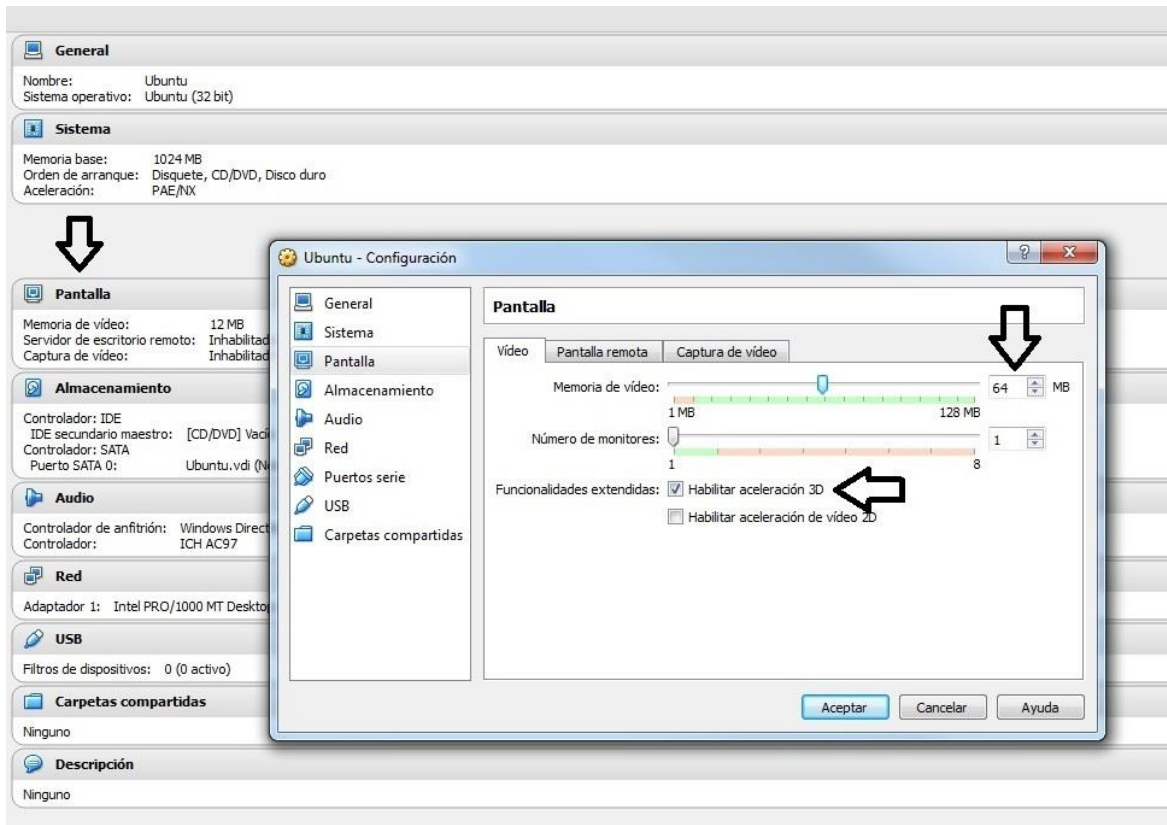
Aparecerá otra ventana, pulsamos en “Apagar”.



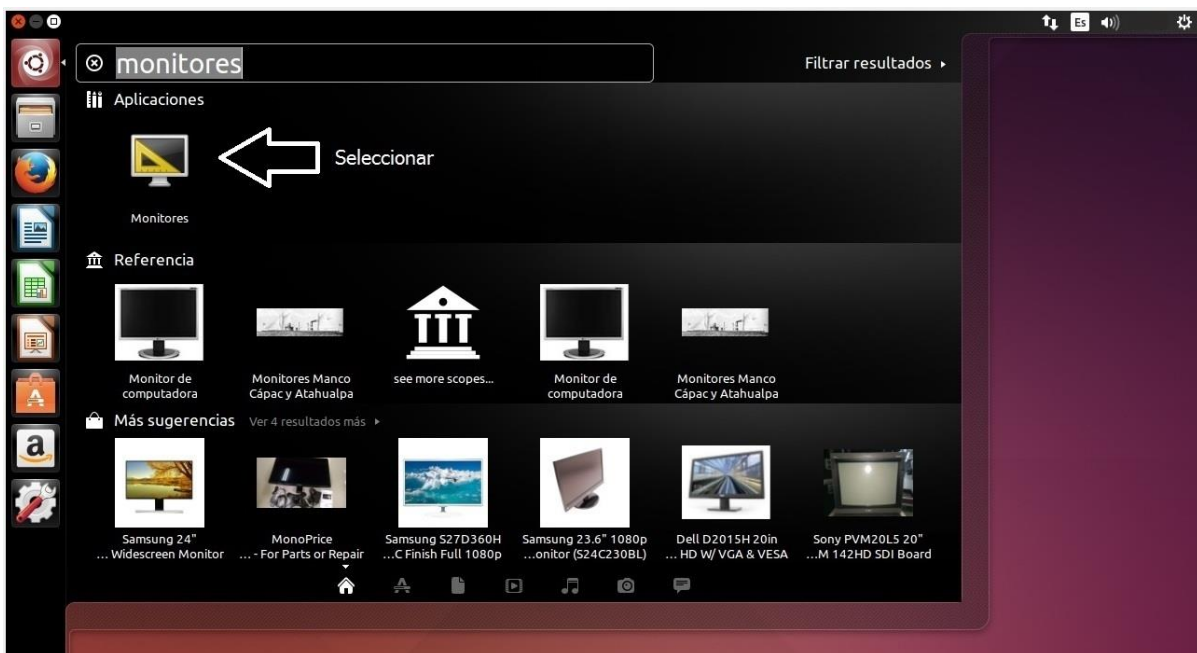
Volveremos a la pantalla principal del “VirtualBox”, seleccionamos “Pantalla” para editar la configuración. En la ventana que aparece aumentamos la cantidad de memoria de video a



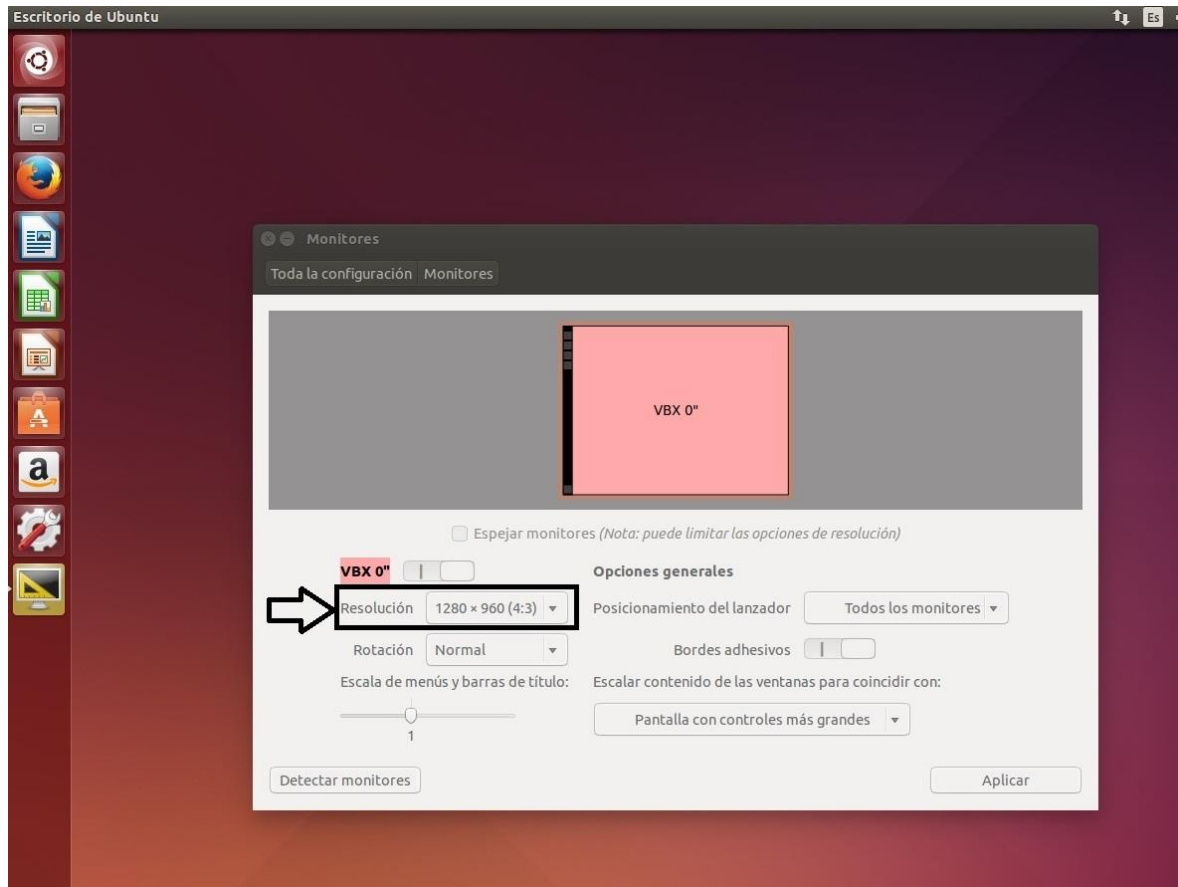
64 MB y tildamos la opción “Habilitar aceleración 3D”. Luego presionamos aceptar y volvemos a iniciar la máquina virtual.



Una vez iniciado Ubuntu, ya podríamos cambiar la resolución de la pantalla. Para ello presionamos el botón de inicio, escribimos “monitores” y seleccionamos la aplicación.



Allí elegimos una resolución que nos resulte cómoda, eso va a depender del monitor que estemos usando. En este caso, elegí 1280x960. Luego presionamos “Aplicar”.

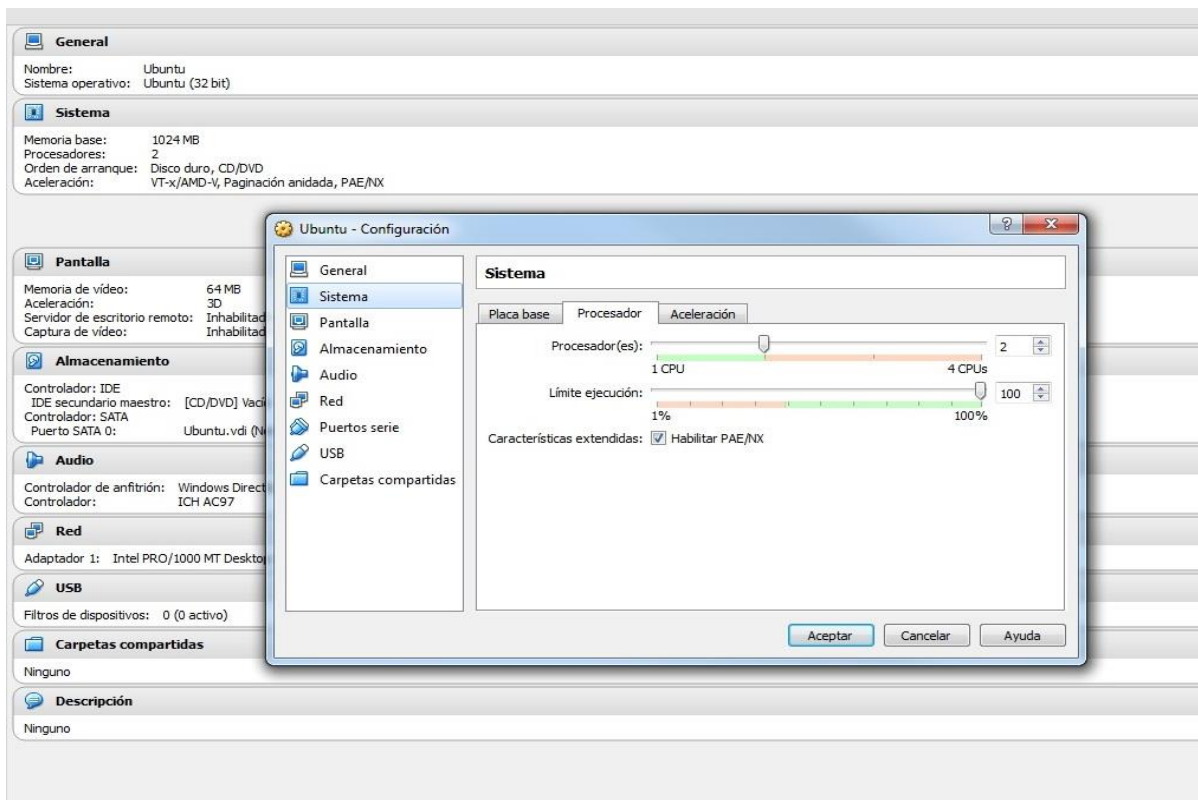
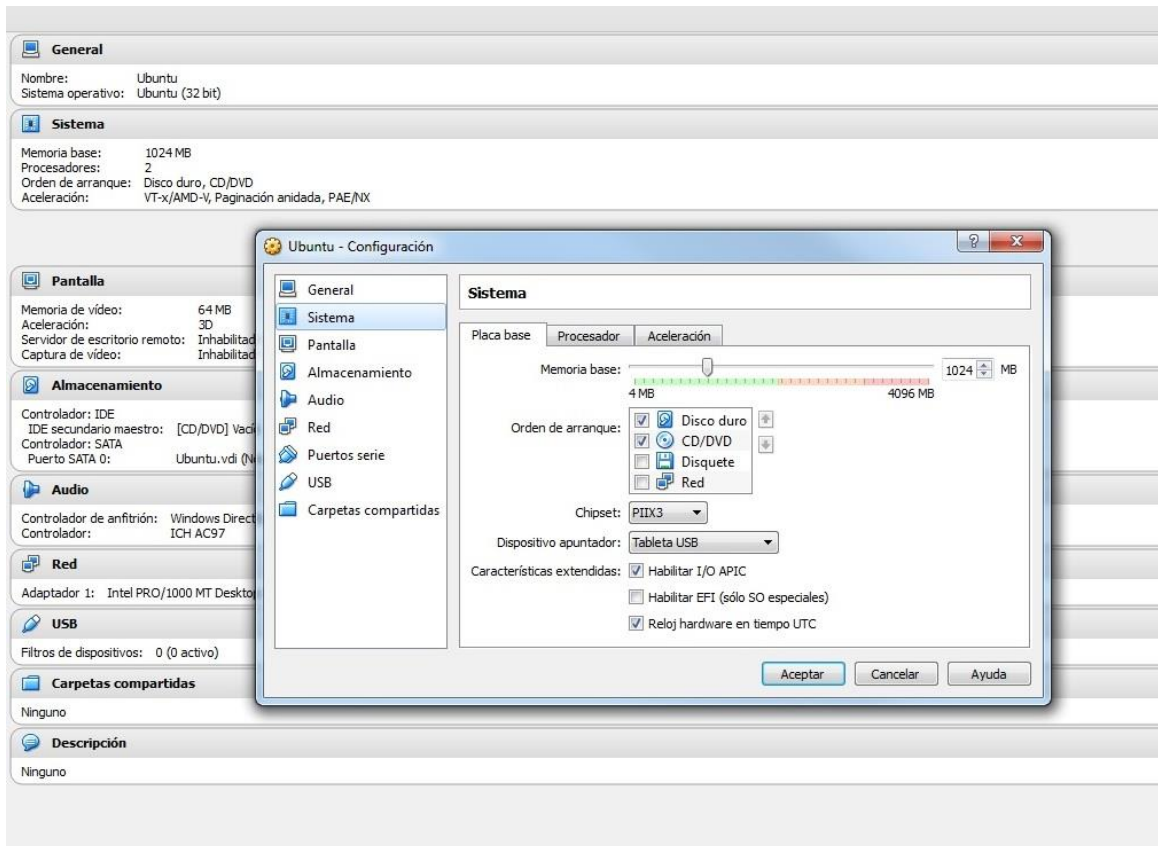


Ahora restaría configurar algunas cuestiones relacionadas con el rendimiento de la máquina virtual. Para ello necesitamos que nuestro equipo soporte “Virtualización por Hardware”. Esta característica depende de nuestro procesador, podemos consultar en la web del fabricante para saber si cuenta con esa característica. Para procesadores Intel se suele llamar “VT-x” y para los AMD, “AMD-V”.

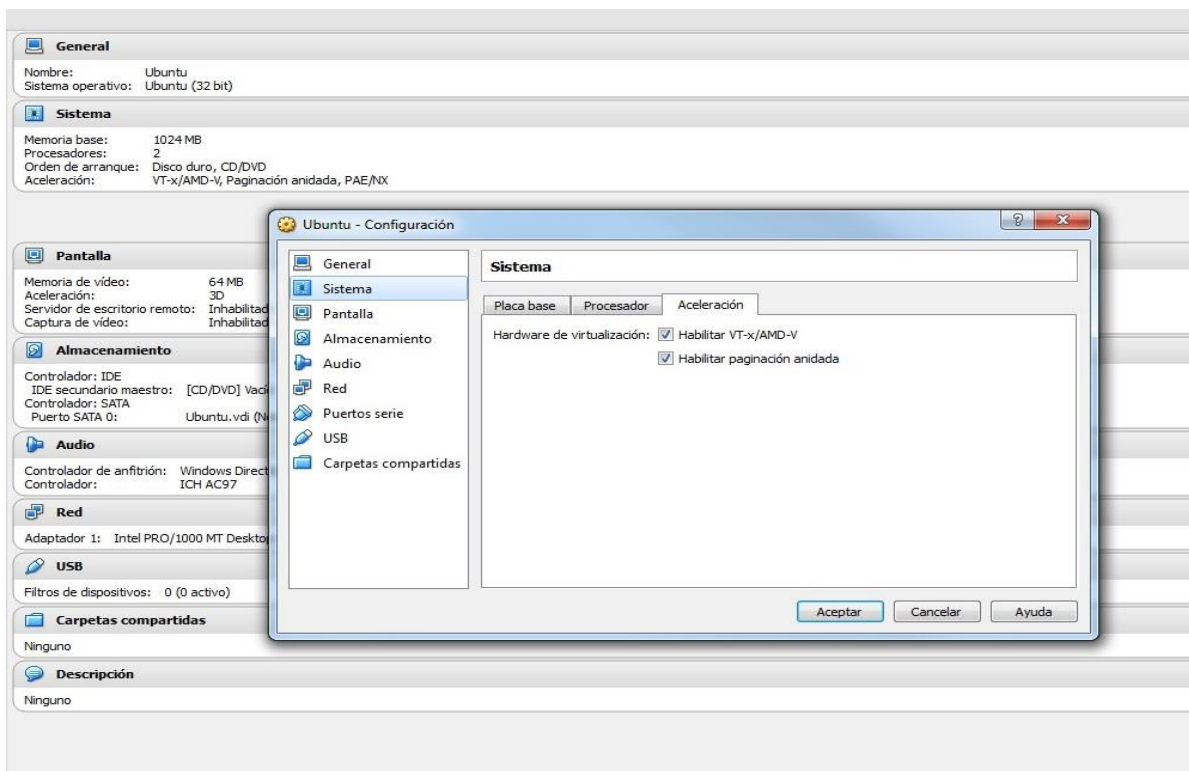
Si nuestro procesador no soporta esta característica no podremos hacer mucho para mejorar el rendimiento de la máquina virtual. En caso de tener la característica, es muy probable que se encuentre deshabilitada por defecto. Para habilitarla hay que acceder al BIOS y activarla desde allí. Este procedimiento depende del motherboard que tengamos, por lo tanto se recomienda buscar en Internet cómo activar la virtualización por hardware, según nuestro motherboard.

Una vez que activamos esta característica en BIOS, hay que configurar la máquina virtual para que la utilice. Para ello abrimos el VirtualBox, seleccionamos “Sistema” en la

configuración de la máquina virtual y aplicamos la configuración que se muestra en las siguientes figuras.



**Nota:** en este caso el equipo utilizado es un Dual-Core, por eso solo seleccionamos dos procesadores (núcleos). Es recomendable seleccionar la cantidad máxima de procesadores permitida, de acuerdo a la cantidad de núcleos de nuestro procesador.



## 7.9. Configuración de entorno de desarrollo en Ubuntu

Hasta ahora hemos instalado Ubuntu dentro de la máquina virtual. A continuación se descargará, instalará y configurará el software necesario para desarrollar para la Raspberry Pi. Todo lo que se hace en esta sección hay que hacerlo dentro de Ubuntu. Por ejemplo, si se requiere descargar un archivo hay que hacerlo desde el navegador de Ubuntu.

### 7.9.1. Java

Primero configuraremos un entorno de desarrollo para programar en el lenguaje Java. Para ello es necesario instalar Java y el Eclipse.

Cuando hablamos de “instalar Java” nos referimos a la instalación del JDK y el JRE. El JDK (Java Development Kit) contiene todos los recursos necesarios para programar en Java, incluyendo el núcleo (core) del lenguaje y las principales bibliotecas. El JRE (Java Runtime Environment) tiene principalmente la máquina virtual de Java (JVM), que permite la ejecución de las aplicaciones desarrolladas en dicho lenguaje.



En cuanto al IDE, la decisión de cual utilizar depende de preferencias personales. En este caso se va a usar Eclipse, pero también podría optarse por otro, como por ejemplo NetBeans o IntelliJ IDEA.

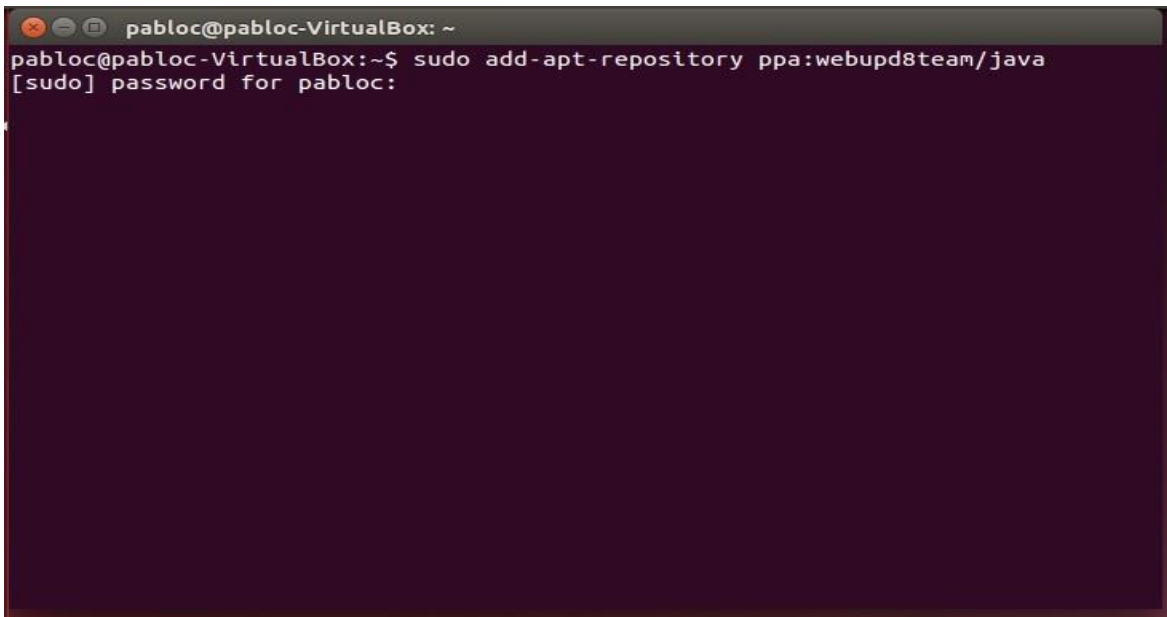
### 7.9.2. Instalación de Java

Utilizaremos la última versión de Java disponible, en mi caso utilicé la versión 8u45. Lo recomendable es utilizar siempre la última versión.

Hay varias maneras de instalar Java en Ubuntu, nosotros utilizaremos el repositorio mantenido por “Webupd8 Team” (<http://www.webupd8.org/>). Agregaremos un repositorio que contiene Oracle Java versión 8.

Al realizar este tipo de instalación (a través del repositorio) siempre se instalará la última versión disponible en el repositorio.

Para agregar el repositorio utilizamos el comando “sudo add-apt-repository ppa:webupd8team/java”. Nos pedirá nuestra contraseña porque estamos ejecutando como “superusuario” (sudo).



```
pabloc@pabloc-VirtualBox: ~  
pabloc@pabloc-VirtualBox:~$ sudo add-apt-repository ppa:webupd8team/java  
[sudo] password for pabloc:
```

Aparecerá lo siguiente, pulsamos “Enter”.



```
pabloc@pabloc-VirtualBox: ~
pabloc@pabloc-VirtualBox:~$ sudo add-apt-repository ppa:webupd8team/java
[sudo] password for pabloc:
Oracle Java (JDK) Installer (automatically downloads and installs Oracle JDK7 /
JDK8 / JDK9). There are no actual Java files in this PPA.

More info (and Ubuntu installation instructions):
- for Oracle Java 7: http://www.webupd8.org/2012/01/install-oracle-java-jdk-7-in
-ubuntu-via.html
- for Oracle Java 8: http://www.webupd8.org/2012/09/install-oracle-java-8-in-ubu
ntu-via-ppa.html

Debian installation instructions:
- Oracle Java 7: http://www.webupd8.org/2012/06/how-to-install-oracle-java-7-in-
debian.html
- Oracle Java 8: http://www.webupd8.org/2014/03/how-to-install-oracle-java-8-in-
debian.html

Important!!! For now, you should continue to use Java 8 because Oracle Java 9 is
available as an early access release (it should be released in 2016)! You shoul
d only use Oracle Java 9 if you explicitly need it, because it may contain bugs
and it might not include the latest security patches! Also, some Java options we
re removed in JDK9, so you may encounter issues with various Java apps. More inf
ormation and installation instructions (Ubuntu / Linux Mint / Debian): http://ww
w.webupd8.org/2015/02/install-oracle-java-9-in-ubuntu-linux.html
Más información: https://launchpad.net/~webupd8team/+archive/ubuntu/java
Pulse [Intro] para continuar o ctrl-c para cancelar
█
```

Una vez que termine, actualizamos los repositorios con “sudo apt-get update”.

```
pabloc@pabloc-VirtualBox: ~
- for Oracle Java 8: http://www.webupd8.org/2012/09/install-oracle-java-8-in-ubu
ntu-via-ppa.html

Debian installation instructions:
- Oracle Java 7: http://www.webupd8.org/2012/06/how-to-install-oracle-java-7-in-
debian.html
- Oracle Java 8: http://www.webupd8.org/2014/03/how-to-install-oracle-java-8-in-
debian.html

Important!!! For now, you should continue to use Java 8 because Oracle Java 9 is
available as an early access release (it should be released in 2016)! You shoul
d only use Oracle Java 9 if you explicitly need it, because it may contain bugs
and it might not include the latest security patches! Also, some Java options we
re removed in JDK9, so you may encounter issues with various Java apps. More inf
ormation and installation instructions (Ubuntu / Linux Mint / Debian): http://ww
w.webupd8.org/2015/02/install-oracle-java-9-in-ubuntu-linux.html
Más información: https://launchpad.net/~webupd8team/+archive/ubuntu/java
Pulse [Intro] para continuar o ctrl-c para cancelar

gpg: anillo «/tmp/tmpi3anhr76/secring.gpg» creado
gpg: anillo «/tmp/tmpi3anhr76/pubring.gpg» creado
gpg: solicitando clave EEA14886 de hkp servidor keyserver.ubuntu.com
gpg: /tmp/tmpi3anhr76/trustdb.gpg: se ha creado base de datos de confianza
gpg: clave EEA14886: clave pública "Launchpad VLC" importada
gpg: Cantidad total procesada: 1
gpg:             importadas: 1 (RSA: 1)
OK
pabloc@pabloc-VirtualBox:~$ sudo apt-get update█
```

Luego, instalamos Java con el comando “sudo apt-get install oracle-java8-installer”. Esto instalará tanto el kit de desarrollo (JDK) como el entorno de ejecución (JRE).



```
pabloc@pabloc-VirtualBox: ~
Obj http://ar.archive.ubuntu.com trusty-backports/main Sources
Ign http://extras.ubuntu.com trusty/main Translation-en
Obj http://ar.archive.ubuntu.com trusty-backports/restricted Sources
Obj http://ar.archive.ubuntu.com trusty-backports/universe Sources
Des:13 http://security.ubuntu.com trusty-security/universe i386 Packages [109 kB
]
Obj http://ar.archive.ubuntu.com trusty-backports/multiverse Sources
Obj http://ar.archive.ubuntu.com trusty-backports/main i386 Packages
Des:14 http://security.ubuntu.com trusty-security/multiverse i386 Packages [3.85
4 B]
Obj http://ar.archive.ubuntu.com trusty-backports/restricted i386 Packages
Obj http://ar.archive.ubuntu.com trusty-backports/universe i386 Packages
Obj http://ar.archive.ubuntu.com trusty-backports/multiverse i386 Packages
Obj http://security.ubuntu.com trusty-security/main Translation-en
Obj http://ar.archive.ubuntu.com trusty-backports/main Translation-en
Obj http://ar.archive.ubuntu.com trusty-backports/multiverse Translation-en
Obj http://security.ubuntu.com trusty-security/multiverse Translation-en
Obj http://ar.archive.ubuntu.com trusty-backports/restricted Translation-en
Obj http://ar.archive.ubuntu.com trusty-backports/universe Translation-en
Obj http://security.ubuntu.com trusty-security/restricted Translation-en
Obj http://security.ubuntu.com trusty-security/universe Translation-en
Ign http://ar.archive.ubuntu.com trusty/main Translation-es_AR
Ign http://ar.archive.ubuntu.com trusty/multiverse Translation-es_AR
Ign http://ar.archive.ubuntu.com trusty/restricted Translation-es_AR
Ign http://ar.archive.ubuntu.com trusty/universe Translation-es_AR
Descargados 614 kB en 8seg. (73,0 kB/s)
Leyendo lista de paquetes... Hecho
pabloc@pabloc-VirtualBox:~$ sudo apt-get install oracle-java8-installer
```

Presionamos “s”.

```
pabloc@pabloc-VirtualBox: ~
Obj http://ar.archive.ubuntu.com trusty-backports/multiverse Translation-en
Obj http://security.ubuntu.com trusty-security/multiverse Translation-en
Obj http://ar.archive.ubuntu.com trusty-backports/restricted Translation-en
Obj http://ar.archive.ubuntu.com trusty-backports/universe Translation-en
Obj http://security.ubuntu.com trusty-security/restricted Translation-en
Obj http://security.ubuntu.com trusty-security/universe Translation-en
Ign http://ar.archive.ubuntu.com trusty/main Translation-es_AR
Ign http://ar.archive.ubuntu.com trusty/multiverse Translation-es_AR
Ign http://ar.archive.ubuntu.com trusty/restricted Translation-es_AR
Ign http://ar.archive.ubuntu.com trusty/universe Translation-es_AR
Descargados 614 kB en 8seg. (73,0 kB/s)
Leyendo lista de paquetes... Hecho
pabloc@pabloc-VirtualBox:~$ sudo apt-get install oracle-java8-installer
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  gsfonTS-x11 java-common
Paquetes sugeridos:
  default-jre equivs binfmt-support visualvm ttf-baekmuk ttf-unfonts
  ttf-unfonts-core ttf-kochi-gothic ttf-sazanami-gothic ttf-kochi-mincho
  ttf-sazanami-mincho ttf-arphic-uming
Se instalarán los siguientes paquetes NUEVOS:
  gsfonTS-x11 java-common oracle-java8-installer
0 actualizados, 3 se instalarán, 0 para eliminar y 6 no actualizados.
Necesito descargar 162 kB de archivos.
Se utilizarán 547 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

Nos pedirá aceptar los términos de licencia, los cuales pueden consultarse desde una URL señalada. Pulsamos “Aceptar” y luego “Si”.

```
pabloc@pabloc-VirtualBox: ~  
Configuración de paquetes  
  
Configuración de oracle-java8-installer  
  
Oracle Binary Code License Agreement for the Java SE Platform Products  
and JavaFX  
  
You MUST agree to the license available in http://java.com/license if  
you want to use Oracle JDK.  
  
<Aceptar>
```

```
pabloc@pabloc-VirtualBox: ~  
Configuración de paquetes  
  
Configuración de oracle-java8-installer  
  
In order to install this package, you must accept the license terms, the  
"Oracle Binary Code License Agreement for the Java SE Platform Products  
and JavaFX ". Not accepting will cancel the installation.  
  
Do you accept the Oracle Binary Code license terms?  
  
<Sí> <No>
```

Comenzará con la descarga y la instalación, puede tardar unos minutos.





```
pabloc@pabloc-VirtualBox: ~
conectado.
Petición HTTP enviada, esperando respuesta... 302 Moved Temporarily
Ubicación: http://download.oracle.com/otn-pub/java/jdk/8u45-b14/jdk-8u45-linux-i
586.tar.gz?AuthParam=1435344048_9c9abe835989f4531dba4460e8e42d20 [siguiente]
--2015-06-26 15:38:49-- http://download.oracle.com/otn-pub/java/jdk/8u45-b14/jd
k-8u45-linux-i586.tar.gz?AuthParam=1435344048_9c9abe835989f4531dba4460e8e42d20
Conectando con download.oracle.com (download.oracle.com)[200.110.216.122]:80...
conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 174985642 (167M) [application/x-gzip]
Grabando a: "jdk-8u45-linux-i586.tar.gz"

  0K ..... 1% 187K 14m55s
 3072K ..... 3% 174K 15m11s
 6144K ..... 5% 174K 15m5s
 9216K ..... 7% 174K 14m54s
12288K ..... 8% 174K 14m39s
15360K ..... 10% 174K 14m24s
18432K ..... 12% 174K 14m8s
21504K ..... 14% 174K 13m52s
24576K ..... 16% 174K 13m35s
27648K ..... 17% 174K 13m18s
30720K ..... 19% 174K 13m1s
33792K ..... 21% 174K 12m44s
36864K ..... 23% 174K 12m27s
39936K ..... 25% 153K 12m17s
43008K ..... 26% 203K 11m53s
46080K .....
```

Cuando finalice veremos algo así.

```
pabloc@pabloc-VirtualBox: ~
er /usr/bin/jstack (jstack) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-8-oracle/bin/jstat para provee
r /usr/bin/jstat (jstat) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-8-oracle/bin/jstatd para prove
er /usr/bin/jstatd (jstatd) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-8-oracle/bin/jvisualvm para pr
over /usr/bin/jvisualvm (jvisualvm) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-8-oracle/bin/native2ascii para
proveer /usr/bin/native2ascii (native2ascii) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-8-oracle/bin/rmic para proveer
/usr/bin/rmic (rmic) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-8-oracle/bin/schemagen para pr
over /usr/bin/schemagen (schemagen) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-8-oracle/bin/serialver para pr
over /usr/bin/serialver (serialver) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-8-oracle/bin/wsgen para provee
r /usr/bin/wsgen (wsgen) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-8-oracle/bin/wsimport para pro
veer /usr/bin/wsimport (wsimport) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-8-oracle/bin/xjc para proveer
/usr/bin/xjc (xjc) en modo automático
Oracle JDK 8 installed
update-alternatives: utilizando /usr/lib/jvm/java-8-oracle/jre/lib/i386/libnprp2
.so para proveer /usr/lib/mozilla/plugins/libjavaplugin.so (mozilla-javaplugin.s
o) en modo automático
Oracle JRE 8 browser plugin installed
Configurando gsfontr-x11 (0.22) ...
pabloc@pabloc-VirtualBox:~$
```

Ahora verificamos que la instalación se realizó correctamente con el comando “java -version”.

```
pabloc@pabloc-VirtualBox: ~  
pabloc@pabloc-VirtualBox:~$ java -version  
java version "1.8.0_45"  
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)  
Java HotSpot(TM) Client VM (build 25.45-b02, mixed mode)  
pabloc@pabloc-VirtualBox:~$
```

**Nota:** en este caso la versión instalada es la 1.8.0\_45, pero esto puede variar dependiendo de cuando se realizó la instalación, ya que se instalará la última versión disponible.

Luego instalaremos un paquete que configura las variables de entorno. Usamos el comando “sudo apt-get install oracle-java8-set-default”.

```
pabloc@pabloc-VirtualBox: ~  
pabloc@pabloc-VirtualBox:~$ java -version  
java version "1.8.0_45"  
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)  
Java HotSpot(TM) Client VM (build 25.45-b02, mixed mode)  
pabloc@pabloc-VirtualBox:~$ sudo apt-get install oracle-java8-set-default  
[sudo] password for pabloc:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes NUEVOS:  
  oracle-java8-set-default  
0 actualizados, 1 se instalarán, 0 para eliminar y 6 no actualizados.  
Necesito descargar 4.666 B de archivos.  
Se utilizarán 37,9 kB de espacio de disco adicional después de esta operación.  
Des:1 http://ppa.launchpad.net/webupd8team/java/ubuntu/ trusty/main oracle-java8  
-set-default all 8u45+8u33arm-1~webupd8~1 [4.666 B]  
Descargados 4.666 B en 0seg. (7.799 B/s)  
Seleccionando el paquete oracle-java8-set-default previamente no seleccionado.  
(Leyendo la base de datos ... 169123 ficheros o directorios instalados actualmen  
te.)  
Preparing to unpack .../oracle-java8-set-default_8u45+8u33arm-1~webupd8~1_all.de  
b ...  
Unpacking oracle-java8-set-default (8u45+8u33arm-1~webupd8~1) ...  
Configurando oracle-java8-set-default (8u45+8u33arm-1~webupd8~1) ...  
pabloc@pabloc-VirtualBox:~$
```

Con esto finaliza la instalación de Java.

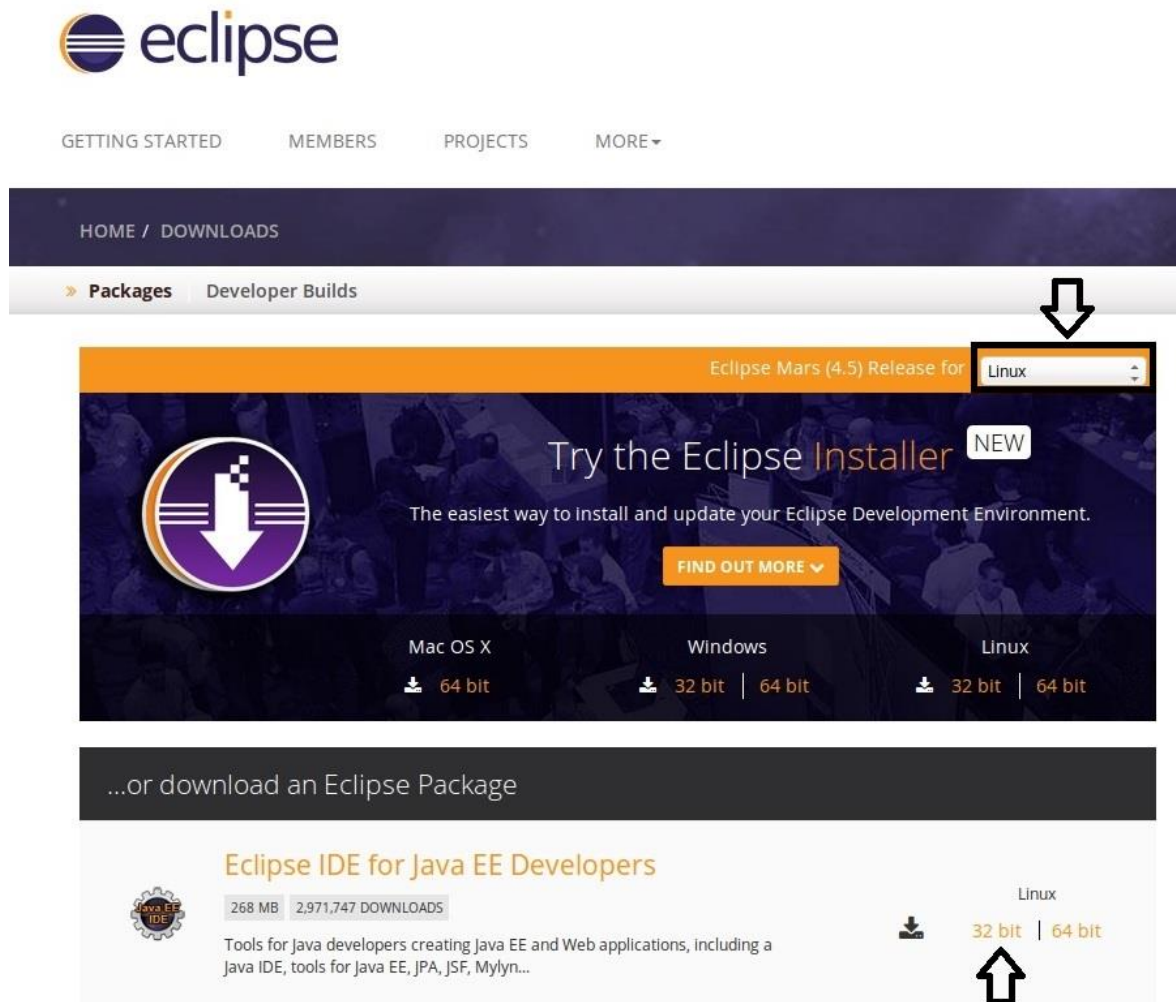
### 7.9.3. Instalación de Eclipse

Ahora que instalamos Java, ya podemos instalar el Eclipse.

Para este tutorial se utilizó la versión Mars 4.5 (lanzada en Junio 2015).

Nos dirigimos a la web oficial de Eclipse: [www.eclipse.org/downloads](http://www.eclipse.org/downloads) y seleccionamos Linux.

Vamos a instalar la versión de Eclipse para Java EE, ya que es la más completa. Seleccionamos la versión de 32 bits. Si se encuentra en una versión de Ubuntu u otro sistema operativo de 64 bits, seleccione el Eclipse de 64 bits.



The screenshot shows the Eclipse website's download page. At the top, the Eclipse logo is visible, followed by navigation links: GETTING STARTED, MEMBERS, PROJECTS, and MORE. Below this is a breadcrumb trail: HOME / DOWNLOADS. The main content area features a large banner for the Eclipse Mars (4.5) Release for Linux. The banner includes the Eclipse logo, the text "Try the Eclipse Installer NEW", and the description "The easiest way to install and update your Eclipse Development Environment." Below the banner are three columns for different operating systems: Mac OS X (64 bit), Windows (32 bit and 64 bit), and Linux (32 bit and 64 bit). A dropdown menu is open, showing "Linux" selected. Below the banner is a section titled "...or download an Eclipse Package" which lists "Eclipse IDE for Java EE Developers" with a download icon and options for Linux (32 bit and 64 bit). Two arrows point to the "Linux" dropdown menu and the "Linux" download options.

Pulsamos "Download".



GETTING STARTED

MEMBERS

PROJECTS

MORE ▾

HOME / DOWNLOADS / ECLIPSE DOWNLOADS - SELECT A MIRROR

## Eclipse downloads - Select a mirror

All downloads are provided under the terms and conditions of the **Eclipse Foundation Software User Agreement** unless otherwise specified.

Download from: **Brazil - C3SL - Federal University of Parana (http)** **OR >**

File: eclipse-jee-mars-R-linux-gtk.tar.gz

Checksums:



Choose a mirror close to you

North America

Canada - **Rafal Rzczkowski**

United States - **Columbia University**

United States - **Indiana University**

### Get It Faster from our Members

**IBM**

Blazingly fast downloads hosted by IBM Bluemix.

**Obeo**

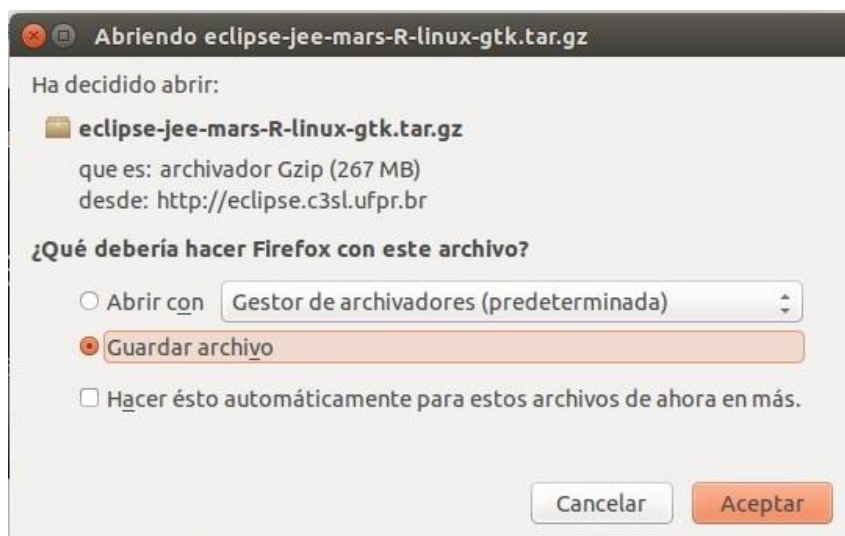
Download the latest Eclipse easily and discover how to create your OWN modeling solutions.

**EclipseSource**

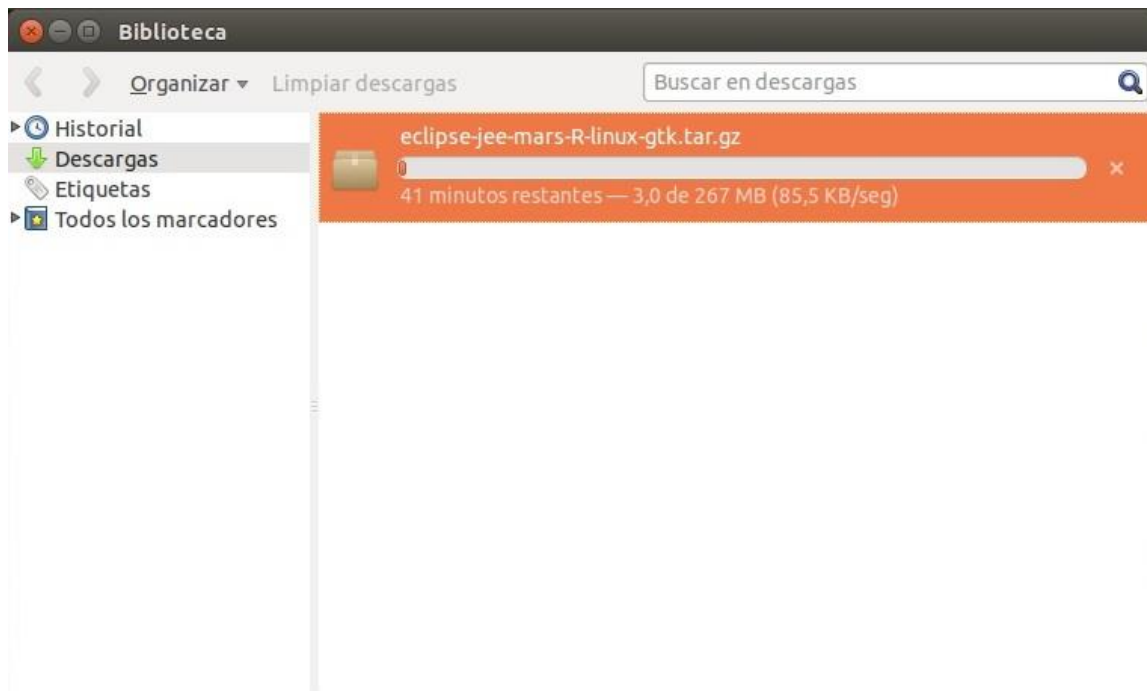
Warp speed from the Amazon cloud plus a choice of hundreds of plug-ins with managed dependencies.

Seleccionamos “Guardar archivo” y pulsamos “Aceptar”.

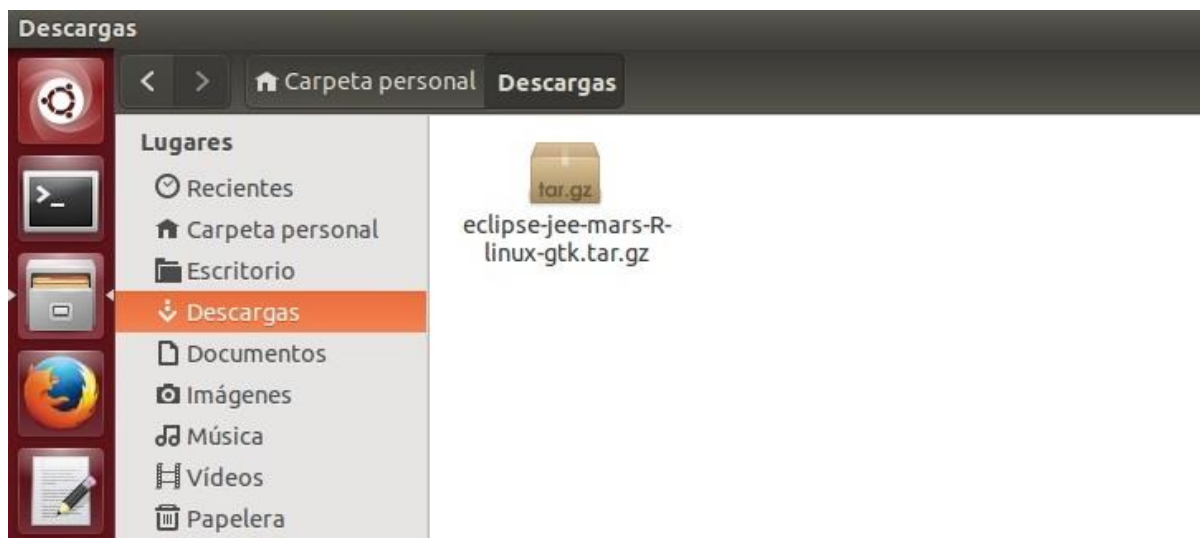
Esto guardará el archivo en la ubicación por defecto (Descargas).



La descarga comenzará, puede tardar varios minutos.



Si nos dirigimos a la carpeta “Descargas” podremos observar el archivo descargado.

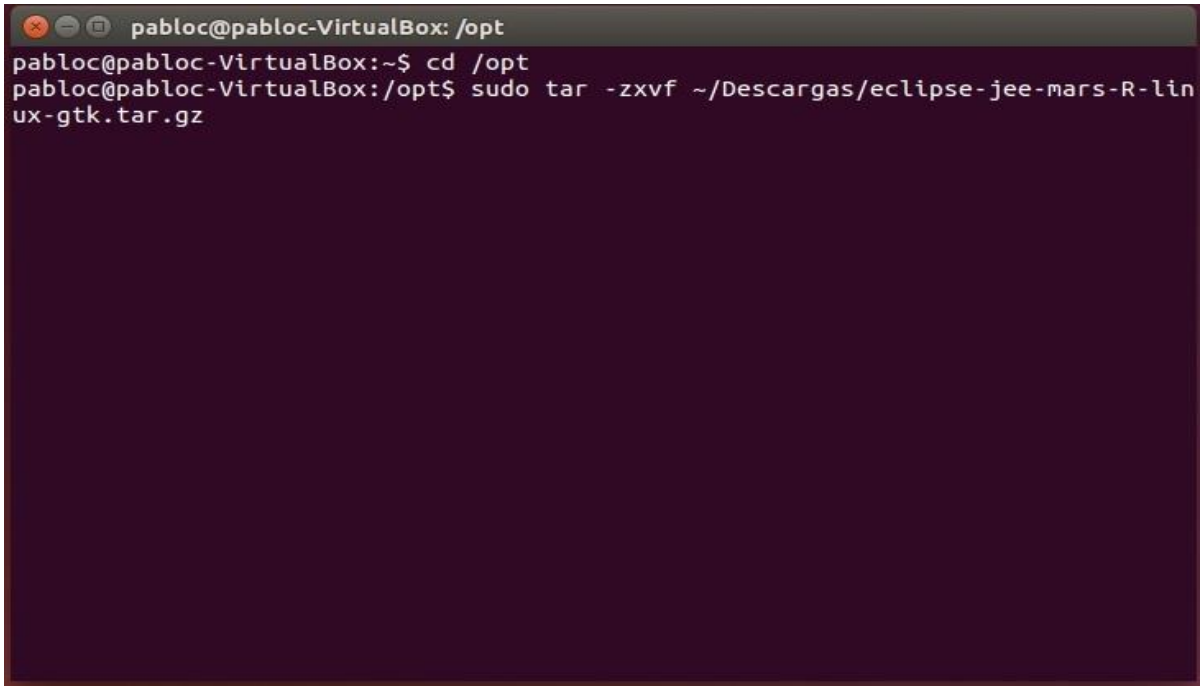


**Nota:** en este caso, el archivo está en la carpeta “Descargas” porque es la ubicación por defecto y no fue modificada. En caso de haber modificado la ubicación de la descarga, dirigirse al directorio correspondiente.



Abrimos la consola, ingresamos el comando “cd /opt”, que nos dirigirá al directorio “/opt”, donde instalaremos el Eclipse. Luego ingresamos el comando para descomprimir el archivo descargado: “sudo tar -zxvf ~/Descargas/eclipse-jee-mars-R-linux-gtk.tar.gz”.

**Nota:** lo que sigue a “sudo tar -zxvf” es la ruta del archivo descargado. Cámbiela en caso que sea necesario.



```
pabloc@pabloc-VirtualBox: /opt
pabloc@pabloc-VirtualBox:~$ cd /opt
pabloc@pabloc-VirtualBox:/opt$ sudo tar -zxvf ~/Descargas/eclipse-jee-mars-R-linux-gtk.tar.gz
```

Cuando finalice la operación veremos algo así.



```
pabloc@pabloc-VirtualBox: /opt
eclipse/plugins/javafx.xml.rpc_1.1.0.v201209140446/META-INF/ECLIPSE_SF
eclipse/plugins/javafx.xml.rpc_1.1.0.v201209140446/META-INF/MANIFEST.MF
eclipse/plugins/javafx.xml.rpc_1.1.0.v201209140446/META-INF/eclipse.inf
eclipse/plugins/javafx.xml.rpc_1.1.0.v201209140446/about.html
eclipse/plugins/javafx.xml.rpc_1.1.0.v201209140446/lib/
eclipse/plugins/javafx.xml.rpc_1.1.0.v201209140446/lib/jaxrpc.jar
eclipse/plugins/javafx.xml.rpc_1.1.0.v201209140446/plugin.properties
eclipse/plugins/javafx.xml.rpc_1.1.0.v201209140446/about_files/
eclipse/plugins/javafx.xml.rpc_1.1.0.v201209140446/about_files/LICENSE
eclipse/plugins/javafx.xml.rpc_1.1.0.v201209140446/about_files/NOTICE
eclipse/plugins/org.eclipse.wst.common.frameworks.ui_1.2.400.v201504292002.jar
eclipse/plugins/org.eclipse.mylyn.wikitext.context.ui_2.5.0.v20150420-1908.jar
eclipse/plugins/org.eclipse.ui.navigator_3.6.0.v20150422-0725.jar
eclipse/plugins/org.eclipse.wst.server.ui.infopop_1.1.200.v201309182117.jar
eclipse/plugins/org.eclipse.equinox.concurrent_1.1.0.v20130327-1442.jar
eclipse/plugins/org.eclipse.epp.mpc.help.ui_1.4.0.v20150603-1603.jar
eclipse/plugins/org.eclipse.datatools.connectivity.dbdefinition.genericJDBC_1.0.2.v201310181001.jar
eclipse/plugins/org.eclipse.oomph.extractor.lib_1.1.0.v20150609-0914.jar
eclipse/plugins/org.eclipse.egit.core_4.0.0.201506090130-r.jar
eclipse/icon.xpm
eclipse/dropins/
eclipse/.eclipseproduct
pabloc@pabloc-VirtualBox: /opt$
```

Ahora crearemos el acceso al Eclipse. Ingresamos el comando:

“sudo -i gedit /usr/share/applications/eclipse.desktop”.

```
pabloc@pabloc-VirtualBox: /opt
pabloc@pabloc-VirtualBox: /opt$ sudo -i gedit /usr/share/applications/eclipse.desktop
*

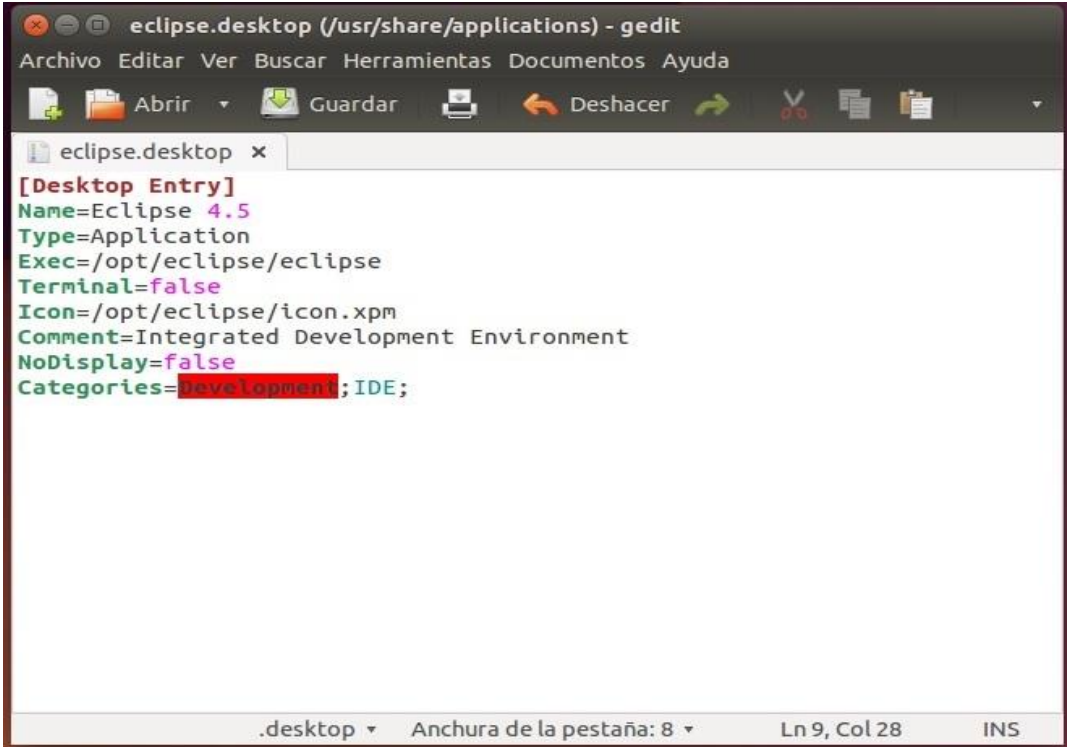
```



Se abrirá el editor de texto y copiamos lo siguiente:

```
[Desktop Entry]
Name=Eclipse 4.5
Type=Application
Exec=/opt/eclipse/eclipse
Terminal=false
Icon=/opt/eclipse/icon.xpm
Comment=Integrated Development Environment
NoDisplay=false
Categories=Development;IDE;
```

Guardamos el archivo y cerramos el editor.

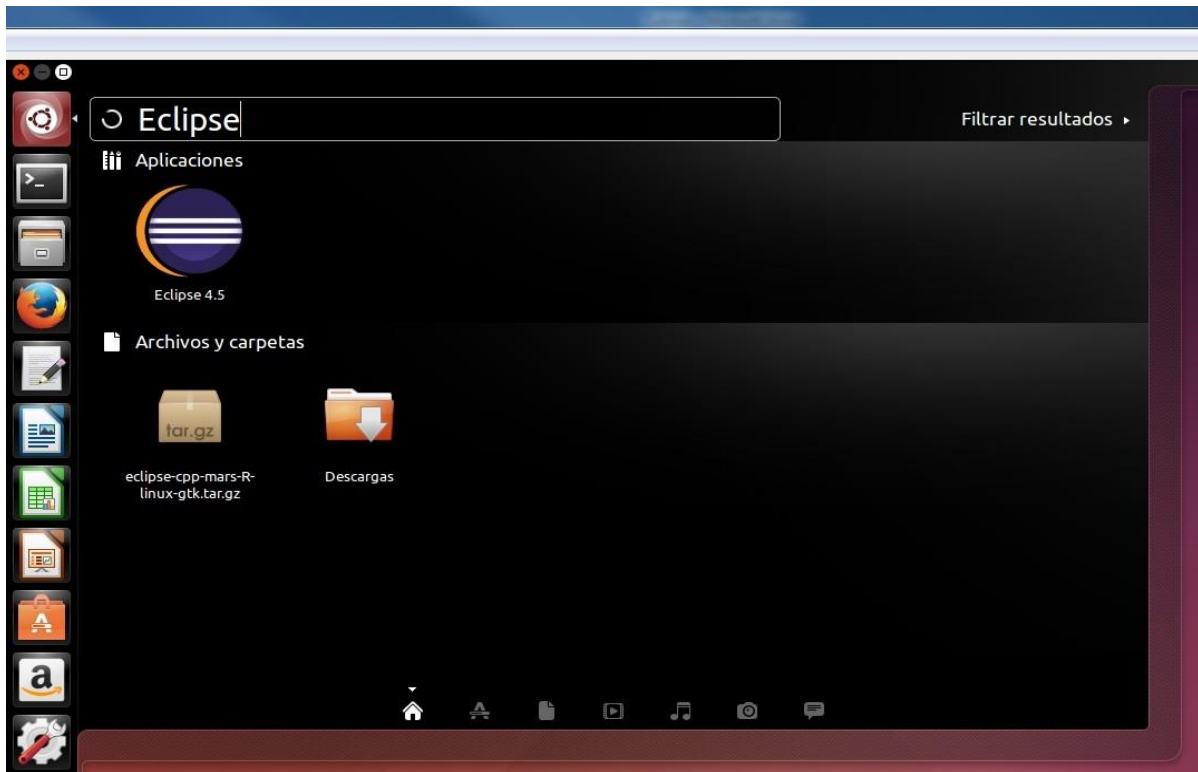


```
eclipse.desktop (/usr/share/applications) - gedit
Archivo Editar Ver Buscar Herramientas Documentos Ayuda
Abrir Guardar Deshacer
eclipse.desktop x
[Desktop Entry]
Name=Eclipse 4.5
Type=Application
Exec=/opt/eclipse/eclipse
Terminal=false
Icon=/opt/eclipse/icon.xpm
Comment=Integrated Development Environment
NoDisplay=false
Categories=Development;IDE;
```

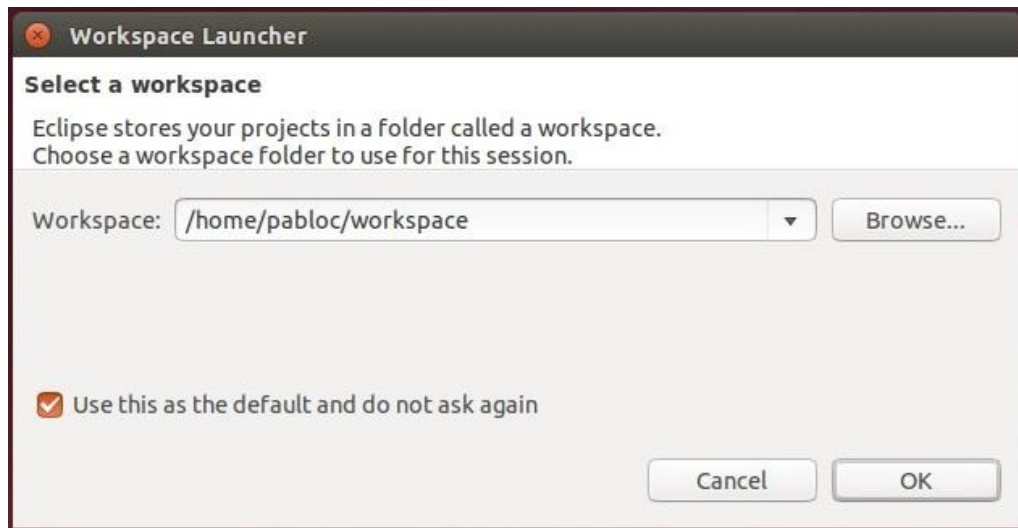
Si buscamos Eclipse, ya debería aparecer. Lo abrimos.

**Nota:** en caso que no aparezca, abrir una Terminal e ingresar “cd /opt/eclipse”. Luego “./eclipse”. Se debería abrir el programa.

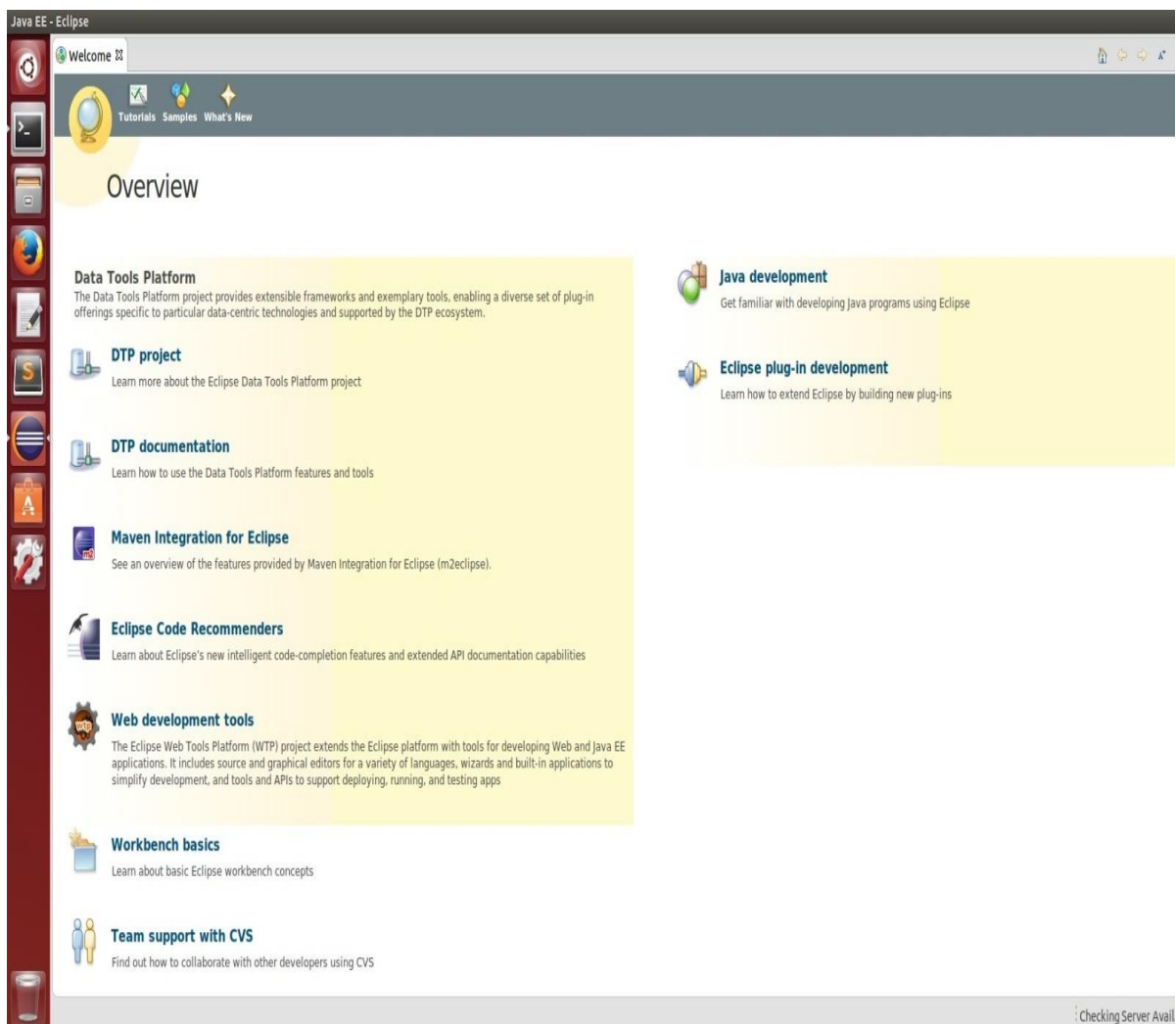




Seleccionamos la ubicación del “Workspace”. Allí se guardarán todos nuestros proyectos. Podemos dejar la ubicación por defecto o seleccionar otra. Tildamos la opción para que siempre use ese Workspace y presionamos “OK”.



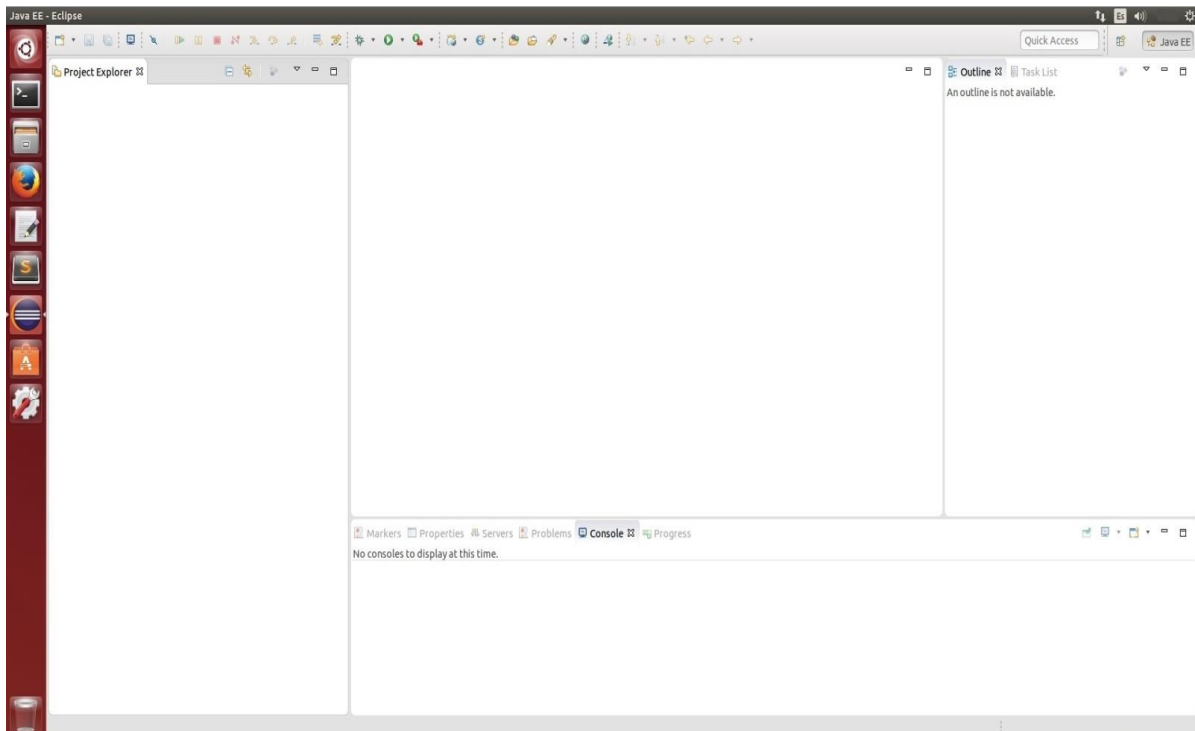
Aparecerá la pantalla de inicio.



Si hacemos click derecho en el ícono del Eclipse y seleccionamos “Mantener en el lanzador”, el ícono quedará allí y podremos ejecutar el programa directo desde la barra lateral la próxima vez que lo necesitemos. Es recomendable hacer esto, sobre todo si el Eclipse no fue encontrado por el buscador.



Cerramos la pantalla de Welcome y aparecerá lo siguiente.



Con esto concluye la instalación del Eclipse para programar en Java. Ya se puede empezar a trabajar con un nuevo proyecto o con proyectos existentes. Cuando se desee ejecutar un programa hecho en Java, solo habrá que generar el archivo .jar desde el Eclipse, copiarlo a la Raspberry y ejecutarlo.



## 8. Anexo C - Análisis y comparativa de retardos en técnicas de streaming

### Raspivid y NC

**Tabla C1:** Tiempos obtenidos en Raspivid y NC

Resolución	Tiempos obtenidos
800 x 600	28 seg
640 x 480	18,8 seg
320 x 240	0,4 seg

### VLC

**Tabla C2:** Tiempos obtenidos en VLC

Resolución	Tiempos obtenidos
800 x 600	1,95 seg
640 x 480	1,8 seg
320 x 240	1,7 seg
1920 x 1080	1,91 seg

### PICAM

**Tabla C3:** Tiempos obtenidos en PICAM

Resolución	Tiempos obtenidos
800 x 600	2,64 seg
640 x 480	3,47 seg
320 x 240	4,12 seg

### Solución Propia

**Tabla C3:** Tiempos obtenidos en la solución propia

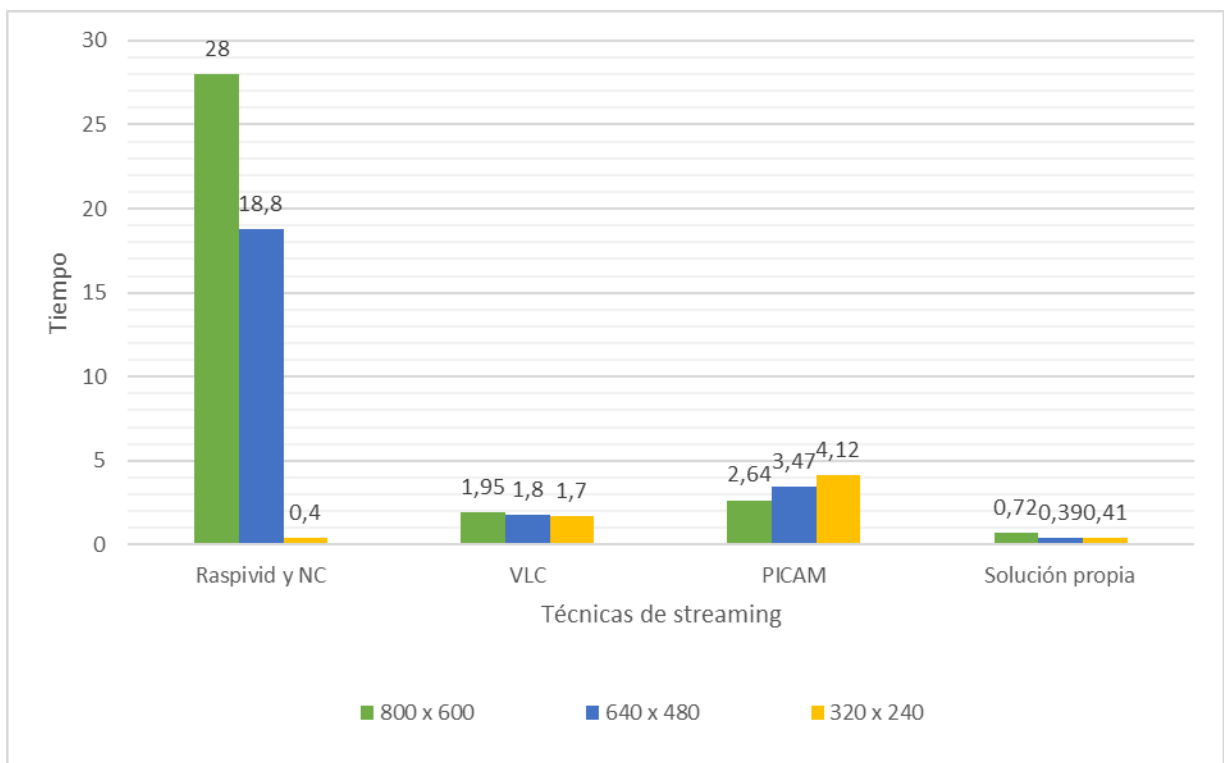
Resolución	Tiempos obtenidos
800 x 600	0,72 seg
640 x 480	0,39 seg
320 x 240	0,41 seg

**Comparación de los resultados obtenidos**

En la tabla 4 se compara los resultados obtenidos en cada caso y se muestran a continuación:

**Tabla C4:** Comparación de los tiempos

Resolución	Tiempo en Raspivid y NC	Tiempo en VLC	Tiempo en PICAM	Tiempo en solución propia
<b>800 x 600</b>	28 seg	1,95 seg	2,64 seg	0,72 seg
<b>640 x 480</b>	18,8 seg	1,8 seg	3,47 seg	0,39 seg
<b>320 x 240</b>	0,4 seg	1,7 seg	4,12 seg	0,41 seg



**Gráfico C1:** Comparación de tiempos de retardo