

Impacto de los Sistemas Operativos Ubicuos en la Actualidad

Henry Peña
Escuela de Ingeniería de Sistemas e
Informática
Universidad Industrial de Santander
Bucaramanga, Santander
line 5: email address

Lizeth Parra
Escuela de Ingeniería de Sistemas e
Informática
Universidad Industrial de Santander
Bucaramanga, Santander
Lparra2597@gmail.com

Jorge Perea
Escuela de Ingeniería de Sistemas e
Informática
Universidad Industrial de Santander
Bucaramanga, Santander
bassbeat120@hotmail.com

Abstract— Currently there is a great interest in the use of mobile devices to causing a great impact on the people enviroment,so that computers appear anywhere and anytime, with any device, without matter the location and the format that is being used [1]. This is defined as ubiquitous computing which allowed a redesign of operating systems, guaranteeing service quality in terms of security, availability or efficiency[2].

Resumen— Actualmente existe un gran interés por el uso de dispositivos móviles los cuales causan que la informática genere gran impacto en el entorno de la persona, de forma que los ordenadores aparezcan en cualquier lugar y en cualquier momento, con cualquier dispositivo, sin importar la ubicación y el formato que se esté usando.[1] Lo anterior se define como computación ubicua la cual permitió un rediseño o en los sistemas operativos, garantizando calidad de servicios en términos de seguridad, disponibilidad o eficiencia[2].

Palabras Clave—sistemas operativos ubicuos, sistemas operativos, sistemas definidos por software, arquitectura de sistemas operativos.

I. INTRODUCCIÓN

Resulta cada vez más interesante ver avances realmente impresionantes. No hay más que pensar en nuevas formas de interacción más naturales y gestuales como la Wii o el iPhone, zapatillas que registran el ejercicio realizado, juguetes que reaccionan al tacto y al habla (sencillos por fuera, pero complejos por dentro), brazaletes que controlan la posición de personas y activan alarmas, paraguas que avisan cuando lo vas a necesitar, y un largo etcétera. Esto es gracias a la computación ubicua la cual se denomina la tercera era de la computación. La primera corresponde a los mainframes (una gran computadora compartida de una compañía), la segunda a la computadora personal y la tercera corresponde a la computación ubicua caracterizada por productos de cómputos portables conectados.

La característica principal de los sistemas ubicuos es establecer un modelo de interacción en el que el procesamiento de información se integra fuertemente en las actividades y objetos cotidianos. En lugar de interactuar intencionadamente con un solo dispositivo como sucede hasta ahora, se interactúa con muchos dispositivos simultáneamente, incluso para las tareas cotidianas y en muchas ocasiones sin que la persona sea consciente de ello. La computación ubicua toca una vasta gama de tópicos, incluyendo el cómputo distribuido, computación móvil, redes móviles, computación contextual y semántica, redes de sensores e interacción hombre-máquina.

Desde hace diez años se vienen introduciendo constantes mejoras en los teléfonos llamados inteligentes, hasta superar el hecho de ser sólo celulares al integrar poder de cómputo y comunicación en un equipo que entra en un bolsillo. Otros ejemplos son las computadoras embebidas incorporadas en los SmartTV que permiten el desarrollo de la televisión interactiva; en los automóviles que proveen seguridad activa y pasiva; en los artefactos del hogar, y en los edificios inteligentes.

A fin de 2012, según estimaciones del Pronóstico Global de Tráfico Móvil de Datos de Cisco, habrá más dispositivos inteligentes que personas [1]. Morgan Stanley, en un informe de 2009 llamado “The Mobile Internet Report”, indica “... La Internet móvil está aumentando más rápido que lo que lo hizo Internet de escritorio y creemos que en cinco años, más usuarios podrán conectarse a Internet a través de dispositivos móviles que desde computadoras de escritorio.

Para analizar los alcances que puede tener esta nueva era de la computación se deben tener en cuenta cinco productos basados en IP que están en continuo crecimiento y convergiendo para el uso de internet móvil: 3G + redes sociales + video + VoIP + dispositivos móviles, además de eso las plataformas de Apple y Facebook las cuales sirven para descubrir como los usuarios se conectan y comunican y los

mercados emergentes tienen un potencial increíble dado el crecimiento de usuarios de internet móvil. La baja penetración de telefonía fija y el valor agregado que se le está dando a los servicios móviles indican que internet será móvil.

Relacionando esta información con el desarrollo de aplicaciones y buscando lograr la integración (característica fundamental de los entornos de computación ubicua) las aplicaciones deben implementar mecanismos para descubrir las necesidades de los usuarios móviles con el fin de presentar la información pertinente en el lugar correcto y en el momento adecuado.

II. ESTADO DEL ARTE

COMPUTACIÓN UBICUA

La computación ubicua o también llamada “UbiComp” [3] es la tercera onda de la computación, la cual tiene dos visiones: reducir la necesidad de los usuarios de concentrarse cuando interactúan con dispositivos de cómputo y proporcionar potencial de cómputo en cualquier momento, en cualquier lugar. En otras palabras, es la integración de la informática en el entorno de una persona, de forma que las computadoras no se perciben como objetos diferenciables, la ubicuidad hace énfasis en tener acceso a la información donde sea en cualquier momento [1].

El objetivo fundamental de la computación ubicua es enviar a las computadoras a un segundo plano, esto permitirá que los usuarios usen las computadoras inconscientemente en apoyo a sus actividades diarias. [3] Esto es lo que Rusell se refiere como “Desaparecer Computadoras” [4].

La computación ubicua hace referencia a una nueva tendencia en la computación en la cual las computadoras tienen una presencia permanente en la vida del usuario. En la computación ubicua las computadoras son teóricamente invisibles para el usuario, a pesar de que siempre están presentes en las actividades diarias de un usuario para apoyar sus necesidades sin que el usuario deba preocuparse por buscar una.

A inicios de 1991 Mark Weiser [2] el científico director de Xerox Palo Alto Research Center ideó la visión de la computación ubicua como una infraestructura omnipresente de información y tecnologías de información. Su investigación en computación ubicua aun continúa para definir las consideraciones socio políticas y tecnologías para lograr el ideal de querer hacer a las computadoras tan naturales que las personas las usarán sin siquiera pensar en hacerlo.

De acuerdo con Weiser, podemos hablar de computación ubicua teniendo en cuenta 4 criterios que son [2]:

- Las microcomputadoras están integradas en objetos físicos de cualquier tipo o forma. Podrían remplazar los servicios presentados por las computadoras de escritorio.
- Estos sistemas integrados se caracterizan por su pequeño tamaño y por su invisibilidad para el usuario.
- Las microcomputadoras incorporadas en objetos aumentan el valor de uso del objeto con un nuevo conjunto de aplicaciones digitales.
- La disponibilidad de servicios ubicuos yace en el centro de la comunicación entre el dispositivo y la aplicación, no del dispositivo en si.

El último criterio es el que distingue a la computación ubicua de las redes móviles familiares de hoy en día. La computación ubicua se distingue por la disponibilidad omnipresente de servicios, independientemente de la plataforma de destino. Los servicios se adaptan a la capacidad física de un dispositivo específico, ya sea un teléfono móvil, una PDA u otro dispositivo.

Los avances en microelectrónica y en las tecnologías de comunicación han trasladado la visión técnica de la computación ubicua en el dominio de lo posible; esto quiere decir que hoy en día contamos con el hardware y software suficiente para lograr hacer computación ubicua. Los primeros ejemplos de computación ubicua incluyen la integración de un módulo de procesamiento en la identificación de documentos y la integración de transpondedores en paletas de carga que envían el ID numérico a un lector de forma automática [2].

Los UOS constituyen un nuevo tipo de sistema operativo para un mundo definido por software en el que el software se utilizará para gestionar todos los aspectos de nuestras vidas. Para comprender el enorme impacto que tendrán los UOS, considere estos ejemplos:

• **SO web** . Los sistemas operativos web, también conocidos como escritorios web o webtops, proporcionan un entorno similar a Linux dentro de un navegador para que los usuarios puedan ejecutar aplicaciones y administrar todos sus datos y almacenamiento. También incluyen API para que los desarrolladores creen aplicaciones que puedan ejecutarse dentro del navegador. Los SO web de ejemplo incluyen Firefox OS, Chrome OS, eyeOS, YouOS y G.ho.st.

• **El sistema operativo Robot** . ROS es un meta-OS que proporciona soporte de desarrollo y tiempo de ejecución para aplicaciones robóticas complejas y robustas [9]. Su extensa colección de herramientas, bibliotecas, abstracciones y API de código abierto se puede utilizar en una amplia variedad de plataformas.

• **HomeOS** . Una iniciativa de Microsoft para habilitar "hogares más inteligentes para todos", HomeOS tiene como objetivo simplificar la creación y gestión de la tecnología de domótica[10] Proporciona controles de usuario intuitivos y

abstracciones de mayor nivel para la orquestación de dispositivos. Los prototipos de investigación de HomeOS se han desplegado en más de una docena de hogares.

•**Sistemas operativos de la ciudad** . Hay muchas iniciativas para crear sistemas operativos para facilitar el crecimiento, el uso de la energía y la sostenibilidad ambiental. Por ejemplo, el Sistema Operativo Urban Living PlanIT (living-planit.com) ofrece abstracciones e interfaces de administración para energía, agua, gestión de desechos, transporte, telecomunicaciones y sistemas de atención médica, así como programación de API para garantizar la interoperabilidad entre las diferentes plataformas.

•**Sistemas operativos en la nube** . Conceptualmente, un sistema operativo en la nube hace lo que hace un sistema operativo tradicional: administrar aplicaciones y hardware, pero a escala del cómputo en la nube, reemplazar los sistemas de archivos por almacenamiento de objetos y permitir una capacidad de almacenamiento y rendimiento de E / S casi ilimitados. En lugar de administrar procesos en máquinas físicas, un sistema operativo en la nube administra tareas en máquinas virtuales. Más importante aún, ofrece varias API para que las aplicaciones en la nube utilicen los recursos de la nube. Muchos proveedores de servicios en la nube han creado su propio sistema operativo en la nube, que incluye Microsoft Azure, Amazon Web Services (AWS) y Huawei FusionSphere. También hay populares OS de nube de código abierto como Open-Stack y Apache CloudStack.

•**SO de IoT** . Cosas de Android de Google (Brillo) es una plataforma de SO integrada diseñada para dispositivos de IoT con poca memoria y con poca memoria que usa las API de Android y los servicios de Google.

Principios De los sistemas operativos Ubicuos

Los UOS pueden escalar a cualquier sistema de tamaño

PRINCIPIOS DE LOS SISTEMAS OPERATIVOS UBICUOS (UOS)

Los UOS pueden escalar a cualquier sistema de tamaño

Los sistemas operativos ya se han creado para pequeños sistemas integrados y dispositivos móviles, como teléfonos inteligentes y tabletas, computadoras de escritorio y portátiles tradicionales, estaciones de trabajo independientes y servidores en red, y clústeres y nubes de servidores. Prevemos que los sistemas operativos se ampliarán para incluir casi todos los sistemas heredados y de próxima generación, desde diminutos dispositivos informáticos de borde hasta enormes entornos informáticos distribuidos que abarcan continentes. Los UOS también se pueden construir para dominios de aplicaciones emergentes como big data e inteligencia artificial.

Se puede construir un UOS para cada objeto (o colección de objetos) en el mundo físico

El objetivo de la informática ubicua es expandir las capacidades de computación más allá de los sistemas de TI tradicionales para hacer que todos los objetos sean más inteligentes. Eso eventualmente significará hacer que estas entidades sean programables, lo que requerirá un sistema operativo. Los robots (incluso los robots Lego) ya tienen sistemas operativos. En una casa inteligente, todos los electrodomésticos, incluidos televisores, lavadoras, refrigeradores, luces, hornos de microondas y cafeteras, necesitarán un sistema operativo para poder programarse. Todos los objetos en movimiento, incluidos vehículos, drones, bicicletas, sillas de ruedas e incluso cochecitos necesitarán también un UOS.

Se puede construir un UOS para cada entidad en el mundo virtual

Además de los objetos y sistemas físicos, los sistemas operativos también podrían crearse para entidades en dominios de aplicación específicos. Por ejemplo, las organizaciones de diversos tipos y tamaños, incluidas las familias, las empresas, las instituciones y las agencias gubernamentales, podrían estar equipadas con capacidades definidas por software para administrar el personal, la información, los cronogramas y el inventario. Los sistemas operativos proporcionarían abstracciones para administrar los recursos, así como también soporte para el desarrollo y la ejecución de nuevas aplicaciones.

CATEGORIAS DE LOS SISTEMAS OPERATIVOS UBICUOS (UOS)

Dados estos principios, podemos esperar ver muchas categorías diferentes de UOSs, por ejemplo:

- **Big Data OS** . Las aplicaciones de Big Data se han creado para una amplia variedad de dominios. Un sistema operativo de big data podría proporcionar funciones especiales para la abstracción y administración de datos, API de acceso y administración de datos, y modelos de programación e idiomas para aplicaciones de big data.
- **SO empresarial** . Es posible que las empresas u organizaciones futuras necesiten un sistema operativo para respaldar la administración eficiente de los procesos, así como de los recursos, incluidas las personas, los fondos y las máquinas. Los sistemas operativos empresariales podrían crearse a partir de sistemas empresariales existentes, como sistemas de información de gestión (MIS) o sistemas de planificación de recursos empresariales (ERP), al agregar API de

programación para permitir el desarrollo flexible de aplicaciones empresariales.

- SO industrial / de fabricación . Muchos fabricantes ya han implementado sistemas automatizados de producción y control robótico. Aunque muchos de estos sistemas se han administrado con sistemas integrados simples, las nuevas abstracciones definidas por software y las capacidades de comunicación mejorarán la eficiencia e inteligencia de los sistemas.
- Sistema operativo humano-ciberfísico . Una tendencia emergente en la informática es la convergencia de tres dominios anteriormente aislados: los seres humanos, los sistemas cibernéticos y el mundo físico. Esto traerá muchas aplicaciones interesantes más allá de los actuales sistemas ciberfísicos y el IoT. Sin embargo, se requerirán nuevas abstracciones y capacidades definidas por software para respaldar la administración del sistema físico-ciberfísico, el desarrollo de aplicaciones y las comunicaciones.
- Inteligencia artificial OS . Se necesitará un sistema operativo para proporcionar abstracciones para aprendizaje automático o capacidades de aprendizaje profundo, así como también soporte de programación para aplicaciones de inteligencia artificial. El cofundador de Android, Andy Rubin, predijo recientemente que la inteligencia artificial sería el próximo gran avance del sistema operativo. [11] Un AIOS se convertirá en una infraestructura esencial para el éxito de los nuevos tipos de aplicaciones inteligentes.

DESAFÍOS TÉCNICOS

A pesar de su promesa, los UOS presentan numerosos desafíos técnicos.

- Modelos y arquitecturas UOS . Es probable que un modelo y arquitectura genérica de UOS no sea adecuado para todos los UOS. El factor más importante es la granularidad de las abstracciones y las interfaces de programación. Una granularidad más pequeña permite más flexibilidad, pero a un costo potencial del rendimiento del tiempo de ejecución de la aplicación. La determinación de este intercambio será fundamental para el diseño de la arquitectura de UOS.
- Virtualización de recursos . La virtualización es la tecnología clave que habilita todos los SO y SDX. Con los UOS, la informática pasará de la nube central al borde, como en los teléfonos inteligentes y los dispositivos de IoT. Por lo tanto, debemos investigar las tecnologías de virtualización livianas para proporcionar abstracciones de sistema operativo eficientes y admitir la informática de borde definida por software.

- La optimización del rendimiento . En los UOS para sistemas informáticos de pequeña escala u objetos con capacidades informáticas débiles, mejorar el rendimiento de ejecución de la aplicación se volverá crítico. A medida que surjan más tipos de hardware, recursos y aplicaciones, será un reto proporcionar servicios eficientes, especialmente para escenarios de alto rendimiento y paralelos masivos.
- La seguridad y la privacidad . El software es más vulnerable a las amenazas de seguridad que el hardware. Con un UOS en su lugar, el software se convierte en el centro de control de un sistema o entorno, convirtiéndolo en el objetivo principal de los atacantes. Además, para los UOS de sistemas que manejan datos personales confidenciales o información crítica, la privacidad también se convertirá en una consideración de primer orden.
- Lenguajes de programación específicos del dominio . Los lenguajes de programación actuales de alto nivel como C / C ++ y Java están diseñados con computadoras en mente. Se necesitarán nuevos lenguajes específicos de dominio para desarrollar aplicaciones más eficientes para UOS particulares, por ejemplo, para un SO empresarial.
- Alcanzar la verdadera inteligencia . El software es la base de todas las aplicaciones inteligentes. Para lograr una verdadera inteligencia, tanto los UOS como las aplicaciones deben ser capaces de "pensar", administrar y ejecutar inteligentemente.

Un aspecto fundamental a tener en cuenta al hablar de computación ubicua es el entorno. En un entorno típico de computación ubicua hay tres (3) componentes fundamentales:

1. DISPOSITIVOS

Dentro de la computación ubicua se hablan de diferentes tipos de dispositivos, se abarca desde los clásicos computadores de escritorio hasta los dispositivos móviles, pasando por las tabletas, servidores y dispositivos móviles que por ejemplo se pueden encontrar en automóviles.

Los dispositivos usados en computación ubicua pueden ser compactos, móviles y proveer funcionalidades que antiguamente eran difíciles de ofrecer con los paradigmas de computación de escritorio y mainframes; un ejemplo de estos dispositivos es IBM BlueBoard; este fue creado por el director de ciencias de usuario e investigación de experiencia (USER) del laboratorio de IBM Dan Russell; los BlueBoards se han diseñado como una forma sencilla pero eficaz para colaborar fácilmente en proyectos dentro de una habitación, un edificio o incluso entre sitios que se encuentran a miles de kilómetros de distancia[6]. Adicionalmente, los sensores wireless son requeridos para percibir la información del entorno; generalmente esta información incluye la hora, identidad y ubicación del dispositivo o del usuario. Por último, la

infraestructura de red facilita la conectividad de los dispositivos y los sensores [3].

2. SENSORES WIRELESS

Dado que la movilidad es una parte integral de nuestra vida cotidiana los ambientes de cómputo ubicuo deben ser compatibles con la movilidad. Por movilidad se entiende la capacidad de apoyar las necesidades de cómputo de los usuarios mientras se mueven de un punto a otro en un sistema de cómputo ubicuo. Estas necesidades abarcan tanto la movilidad de las tareas de cálculo como la disponibilidad de recursos.

La mayoría de los sistemas de computación ubicua incluyen dispositivos que están conectados entre sí a través de enlaces inalámbricos. Por lo tanto, el uso de las redes inalámbricas crea mucha vulnerabilidad e introduce numerosas amenazas. Dentro de las amenazas más comunes se encuentran [7]:

- El control de acceso.
- Denegación de servicios.
- Manipulación de datos.
- Uso no autorizado de identidad.

Para mitigar este conjunto de amenazas se deben tener en cuenta los requerimientos de seguridad. Éstos pueden ser clasificados en generales y específicos; en generales se encuentran los requerimientos de seguridad principales de una red ubicua como son la confidencialidad, integridad, autenticación, autorización y el no repudio. Los requerimientos generales incluyen algunos requerimientos basados en aplicaciones como la interoperabilidad, la disponibilidad ubicua, credenciales de red, la plataforma y la protección de contenido [7]

3. INFRAESTRUCTURA DE RED

Como la computación ubicua se centra en minimizar las interacciones de los usuarios on los dispositivos informáticos, se debe asegurar una buena conectividad; la capacidad de un dispositivo móvil para automáticamente conectarse, desconectarse y reconectarse a diferentes antenas de celular para garantizar un servicio continuo es un buen ejemplo de requerimientos de sistemas ubicuos. De esta forma en un entorno de computación ubica se debe brindar un continuo seguimiento al usuario para poder brindarle los servicios y la capacidad de computo; además la capacidad del entorno de enviar cualquier cosa que un usuario estuviera trabajando en su computador de escritorio a su dispositivo móvil demuestra el soporte de la movilidad del usuario.

III. METODOLOGÍA

A. Estado del arte de los sistemas operativos oblicuos

Los sistemas operativos ubicuos podrían verse realmente como una representación mas general y abstracta de los mismos principios sobre los cuales se fundamenta un sistema

operativo tradicional: virtualización de los recursos de un sistema de hardware y la programabilidad de funciones sobre dicho sistema de hardware.

Para hablar más en detalle de lo que es virtualización de los recursos de un sistema de hardware, hay que hablar de dos puntos de vistas claves: punto de vista físico del sistema de hardware el cual indica el estado real del sistema en cuestión y el punto de vista lógica el cual indica los estados del sistema de hardware tal que favorezca la programabilidad de funciones sobre dicho sistema. Dicho lo anterior podríamos definir cualquier mecanismo de virtualización como un mapeo de los estados reales del sistema hacia un conjunto de estados ficticios que permiten llevar a cabo operaciones sobre procesos lógicos tales como la conmutación de procesos y la partición recursos.

Dicho lo anterior podríamos construir el estado del arte de los sistemas operativos ubicuos en términos de como se modelan y se implementan los mecanismos de virtualización de tal forma que estos sean lo más robusto posible con el fin de que sean fácilmente adaptable a cualquier sistema de hardware. Actualmante los modelamientos de los mecanismos de virtualización antes mencionados se dan bajo una arquitectura que coincide conceptualmente con la de un sistema operativo tradicional como se muestra en la figura 1.

De la anterior imagen mostrada se puede observar que la abstracción o el mapeo del punto de vista físico hacía de características ficticias de un sistema de hardware es robusta que la de un sistema operativo tradicional, ya que esta se puede adaptar, en teoría, a muchos de los sistemas de

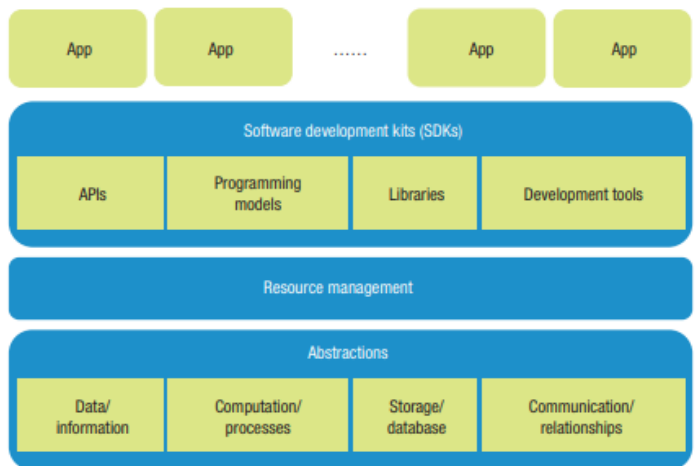


Fig. 1. Arquitectura general de un sistema operativo ubicuo

hardware específico sobre los que actúan normalmente un sistema operativo tradicional. Bajo la anterior abstracción se puede administrar los recursos de cualquier sistema de hardware específico con el fin de desarrollar app sin la preocupación que estas no funcionen en uno de los dispositivos que administra el sistema operativo ubicuo.

Dado lo anterior mostraremos un prototipo de sistema operativo ubicuo el cual se llama internetware el cual consiste en un conjunto de entidades de software distribuidos en internet, dichas entidades poseen la capacidad de comunicarse entre sí. Lo anterior supone entonces un paradigma para el diseño de software o aplicaciones que requieren del uso tanto de recursos de hardware de bordes (pc, celulares, sistemas embebidos, televisores, tablets, etc.) y los recursos de hardware de la nube, con el fin de dotar con cierta capacidad de computo al hardware de borde. Lo anterior planteado se ve esquematizado en la figura 2 en donde se muestra la arquitectura de cualquier sistema operativo internetware. A forma de ejemplos específicos de los sistemas operativos internetware podemos mencionar los siguientes: YanCloud, CampusOS y YanDaaS.

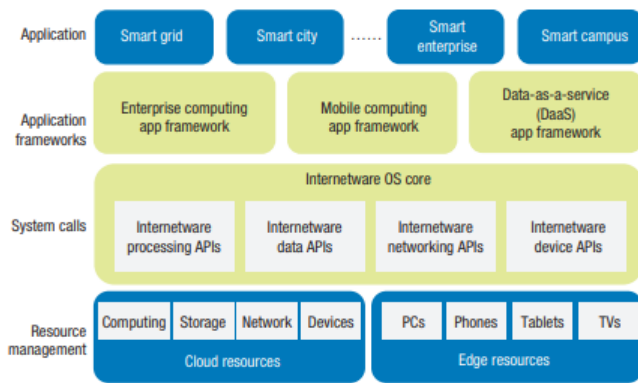


Fig. 2. Arquitectura de un internetware OS

B. Importancia de los sistemas definidos por software

Para abarcar la importancia de los sistemas definidos por software, es pertinente definir lo que es un sistema operativo o el por qué el surgimiento de este; ya que en las primeras computadoras, o más bien, en los sistemas de cómputo antiguos no existía lo que hoy en día llamamos sistemas operativos (Windows, MacOS, Linux) esto es debido a que en esos sistemas computacionales las aplicaciones o programas que permitían al usuario realizar distintas tareas se ejecutaban directamente en las máquinas, esto porque su complejidad lo permitía, pero con el paso del tiempo y con el desarrollo de lo que hoy llamamos software, se hizo más complicado administrar los recursos de manera directa mediante las mismas aplicaciones que se ejecutaban en la máquina. Debido a lo mencionado anteriormente se hizo necesario sintetizar funcionalidades y herramientas comunes, las cuales llamamos hoy en día controladores y bibliotecas, creando así una manera de abstraer los recursos de un sistema de hardware para ser compartidas y usadas por diferentes aplicaciones que están pensadas para ejecutarse en dicho sistema de hardware. Esta capa de software que inicialmente era la encargada de contener los controladores y bibliotecas constituiría lo que actualmente se conoce como un sistema operativo, cabe resaltar que los sistemas operativos actuales no se centran en cómo el sistema de hardware opera, sino que más bien busca

dar soporte al mismo tiempo que el sistema computacional está en marcha. el cómo el sistema de hardware opera, sino que más bien busca dar soporte al mismo tiempo que el sistema computacional está en marcha.

La figura 3 muestra lo que fueron los sistemas operativos principales y compara sus características, la diferencia entre los sistemas operativos actuales se da más en lo que quieren contrastar, por mencionarlo de otra forma, se enfocan en características diferentes para cumplir múltiples propósitos más que todo ‘comerciales’, pero emplean principalmente una arquitectura basada en Unix.

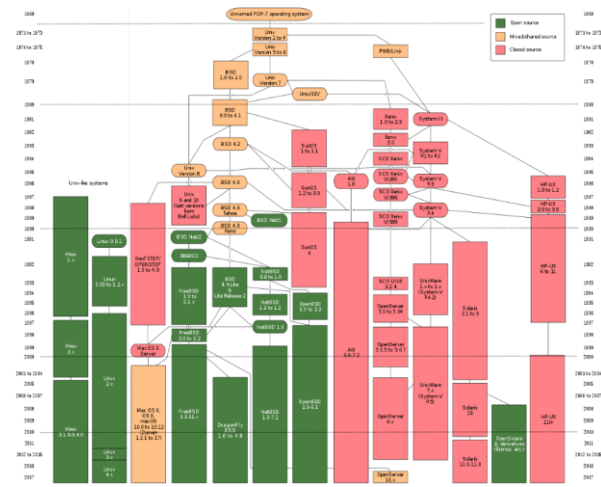


Fig. 3. Breve evolución de los sistemas operativos

Con el paso del tiempo, el ansia por compartir la información y facilitar las actividades del diario vivir se implementó la conexión entre sistemas computacionales proporcionándoles a estos la capacidad de unirse a una red lo cual condujo al desarrollo e implementación de sistemas operativos de red (NOS). El primer NOS fue Novell Network, el cual se centro en conectar sistemas computacionales dentro de una red local (LAN), pero fue discontinuado y retirado más tarde cuando la necesidad por la conexión a la red aumento junto al escaso marketing, además de que los sistemas operativos de escritorio empezaron a implementar capacidades de red y a incorporar funcionalidades de administración de datos relacionadas con internet.

Después de lo anteriormente dicho apareció el termino de “sistemas definidos por software” el cual hace referencia a una familia de tecnologías, las cuales pueden ser realizadas o automatizadas por software, un ejemplo de esto pueden ser aplicaciones tales como Uber o aplicaciones para dispositivos móviles con acceso a internet como las cámaras GoPro, drones hasta automóviles sin conductor. La infraestructura física de sistemas computacionales definidas por software es la próxima generación de diseño de infraestructura necesaria para conectar correctamente los dispositivos anteriormente mencionados. Esta interconexión entre dispositivos que hacen una computación sencilla es lo que hace que se empiece a

acuñar el término computación ubicua que tiene que ver con dotar e interconectar dispositivos con baja, mediana y alta capacidad de computación.

Ahora bien, ¿Qué diferencia lo que llamamos sistema definido por software de un sistema operativo?, la respuesta corta a esta pregunta es no mucho; ya que en un sistema definido por software los recursos de hardware pueden ser virtualizados y administrados mediante rutinas, ya sean del sistema operativo o del plano de control con el fin de que los usuarios puedan acceder y administrar servicios proporcionados por estos recursos virtualizados lo cual es algo muy similar a las capacidades y funciones de un sistema operativo como por ejemplo Linux, MacOS o Windows, que proporcionan virtualización de recursos de hardware a través de controladores de hardware, además de un soporte en tiempo de ejecución por medio de kits de desarrollo de software y bibliotecas, pero esto no solo lo ofrecen los sistemas operativos de escritorio, también lo hacen los sistemas operativos de los dispositivos móviles o dispositivos pequeños como los denominados TinyOS incluso grandes clustres por medio de sistemas operativos en la nube, por lo que se puede llegar a la conclusión de que las tecnologías definidas por software se pueden relacionar como un sistema operativo ya que estos dos están basados en los mismos principios.

Con todo lo anterior mencionado podemos denotar claramente cual es la importancia de los sistemas definidos por software, y es que si tomamos como enfoque o meta principal llegar a la ubicuidad o sistemas operativos ubicuos, el software definido es de gran importancia e interés ya que el software desde muchos puntos de vista moldea nuestra forma de vivir ya sea desde la forma en que trabajamos hasta la forma en que jugamos y su infraestructura es la próxima generación de infraestructura para poder conectar dispositivos y aplicaciones a la red entre sí y en general a todos nosotros.

C. Retos de la implementación de sistemas operativos ubicuos

Antes de abarcar lo que son los retos técnicos que pueden surgir del desarrollo e implementación de un sistema ubicuo, cabe recalcar cuales son los principios básicos y estos son tres:

- **Escalabilidad:** los sistemas operativos ubicuos deberán escalarse a cualquier sistema de cualquier tamaño, todo esto previendo que los sistemas operativos se amplían para incluir casi todos los sistemas heredados y de próxima generación abarcando desde diminutos dispositivos hasta grandes entornos informáticos.
- **Interoperabilidad:** los sistemas operativos ubicuos serán construidos para cada objeto (o colección de objetos) en el mundo físico, esto con el fin de expandir las capacidades de computación más allá de los sistemas tradicionales para hacer la vida cotidiana o sus componentes tecnológicos un poco más “inteligentes”, lo cual implica que dichos objetos puedan ser programados lo cual intrínsecamente requeriría un sistema operativo.

- **Virtualización:** además de objetos y sistemas físicos, los sistemas operativos ubicuos también tendrán la capacidad de establecerse en dominios de aplicación específicos proporcionando abstracciones para administrar los recursos, así como también soporta para el desarrollo y la ejecución de nuevas aplicaciones.

A partir de estos principios básicos surgen entonces categorías para estos sistemas operativos ubicuos con sus respectivos desafíos técnicos ya que se querrá obviamente enfatizar en cierta área para cubrir y mejorar su aplicación; entre estas categorías se encuentran los sistemas operativos para big data, sistemas operativos empresariales, industriales o de fabricación, sistemas operativos humano-ciberfísicos y de inteligencia artificial entre otros.

Como fue mencionado anteriormente estos sistemas operativos presentarán desafíos técnicos, gracias a que siempre existirán limitantes presentes en el desarrollo humano, entre estos se encuentran:

Los modelos y arquitecturas: El encontrar una arquitectura genérica para un sistema operativo será un desafío ya que puede que este no sea el adecuado para todos los sistemas operativos Ubicuos, un factor importante a tratar en este problema, es la granularidad de las abstracciones y las interfaces de programación. ya que el elegir entre una granularidad baja o alta implicaría entre mayor o menor flexibilidad y un bajo o alto costo potencial del rendimiento del tiempo de ejecución de la aplicación.

La virtualización de recursos: Se sabe que la virtualización es una característica clave de un sistema operativo, y esta misma es la tecnología más importante, por lo que se tendrá que buscar una manera de hacer estas más “livianas” para proporcionar abstracciones de sistema operativo eficientes y admitir la computación en el borde (Edge Computing).

La seguridad y privacidad: Como muy bien se sabe, en cuestiones de seguridad, el software siempre ha sido más “delicado” o más bien vulnerable que el hardware, por lo que con un sistema operativo ubicuo este problema aumentaría, ya que con un sistema operativo ubicuo, el software se convierte en el centro de control de un sistema o entorno, por lo que este mismo pasaría a ser objetivo principal de cualquier atacante, sumándole que si este sistema operativo maneja datos personales o confidenciales u información crítica, la privacidad también se convertiría en un factor peligro a considerar.

La optimización del Rendimiento: Los sistemas operativos Ubicuos tendrán también problemas a la hora de optimizar el rendimiento de ejecución, ya que, a pesar de su tendencia de segregación frente al hardware, va a necesitar de él para proporcionar servicios eficientes, de alto rendimiento, hasta de paralelismo.

Lenguajes de programación: Lenguajes actuales de alto nivel como por ejemplo C++ o Java están diseñados para cierto tipo de computadoras, por lo que se necesitarán nuevos lenguajes

específicos de dominio para desarrollar aplicaciones más eficientes para determinado Sistema Operativo Ubicuo.

Sistemas cada vez más Inteligente: Se sabe que el software es la base de todo sistema o aplicación inteligente. y para lograr aumentar la “inteligencia”, tanto los sistemas operativos como las aplicaciones, deben ser capaces de Interactuar con el entorno, en otras palabras, administrar y ejecutar inteligentemente.

Además de estos retos técnicos también se podrían presentar problemas en las compañías de infraestructuras, ya que este cambio, las haría enfrentar a una crisis de identidad inesperada, reformulándoles preguntas como ¿Quiénes somos?, y ¿qué y dónde entregamos valor?

Con el rápido desarrollo y despliegue de sistemas ubicuos en el mundo, ¿qué retos debemos afrontar como futuros ingenieros de sistemas? De un ejemplo de una posible aplicación en algún sector productivo en Colombia (agroindustria, energía, salud).

IV. CONCLUSIONES

- Los sistemas ubicuos, se pueden enriquecer, haciéndolos más eficientes y más viables, adicionándoles una parte social y participativa, con el fin de dar solución a diferentes problemas con los que convivimos en el diario vivir como es: la contaminación ambiental y la presencia de alérgenos, analizando aspectos como la calidad del aire, agua y estado del suelo
- Este tipo de sistemas, debe considerar un gran número de características que son heredadas de la computación ubicua, como de los sistemas participativos (crowdsourcing). El objetivo de un sistema de información ubicuo y participativo es la generación de conocimiento con base a la recopilación y uso de los datos en cualquier momento y en cualquier lugar (característica principal de ubicuidad de la información). Gracias a los análisis comparativos realizados a los frameworks de computación ubicua, se identificaron una serie de oportunidades de investigación, dentro de estas se encuentran: i) Plantear estrategias y técnicas para prevenir los ataques en los posibles escenarios de amenaza, en estos escenarios, terceros malintencionados podrían afectar la integridad del sistema. ii) Representar el contexto mediante un modelo unificado y general.
- Con todo el estudio realizado se puede esperar que el internet móvil este aumentando cada vez más rápido, permitiendo a cualquier usuario conectarse a internet a través de dispositivos móviles teniendo en cuenta puntos clave como son los dispositivos IP que están en continuo crecimiento para el uso de internet móvil. También las plataformas de Apple y Facebook ayudan a descubrir como los usuarios se conectan y

comunican, otro aspecto importante son los mercados emergentes los cuales tienen un potencial increíble dado el crecimiento de usuarios de internet móvil. Así que todo lo anterior se puede relacionar con el desarrollo de aplicaciones pero para esto se debe lograr la integración (característica fundamental de la computación ubicua) con mecanismos que permitan descubrir las necesidades de los usuarios móviles con el fin de presentar información pertinente en el lugar correcto y en el momento adecuado.

V. REFERENCIAS

- [1]. R. Jason Weiss, J. Philip Craiger. Ubiquitous Computing.
- [2] Bundesamt für Sicherheit in der Informationstechnik–BSI. Pervasive Computing: Trends and Impacts.. 2006.
- [3] Dennis Lupiana, Ciaran O’Driscoll, Fredrick Mtenzi. Taxonomy for Ubiquitous Computing Environments.
- [4]. D. Russell, N. Streitz, and T. Winograd, "Building disappearing computers," Communications of the ACM, vol. 48, pp. 42-48, 2005
- [5]. B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," presented at Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on, 1994.
- [6].IBM Blue Board. <http://www.richgossweiler.com/projects/BlueBoard/>.Accedido el 16 de Septiembre de 2012.
- [7]. Dr. Dionisis X. Adamopoulos. Meeting the Needs of Sophisticated Applications with Ubiquitous Computing Systems.
- [8]. Hong Mei and Yao Guo, Peking University. 0018-9162/18/\$33.00 © 2018 IEEE. Toward Ubiquitous Operating Systems: A Software-Defined Perspective
- [9].M. Quigley et al., “ROS: An OpenSource Robot Operating System,” ICRA Workshop Open Source Software, vol. 3, no. 3.2, 2009; www.willowgarage.com/sites/default/files/icraoss09-ROS.pdf
- [10]. C. Dixon et al. “An Operating System for the Home,” Proc. 9th USENIX Symp. Networked Systems Design and Implementation (NSDI), 2012, pp. 25–25.
- [11]. T. Haselton, “The Man behind Android Says A.I. Is the Next Major Operating System,” CNBC, 18 Aug. 2017; www.cnbc.com/2017/08/18/andy-rubin-says-ai-is-next-big

