

Entrevista En Diferido 9: Rodolfo Pilas.

16 de Mayo de 2019 a las 15:43

Hoy comenzamos una entrevista a *Rodolfo Pilas*, administrador de sistemas y creador del podcast *deployandome*.

Antes de todo, gracias por participar y apoyar este proyecto. La primera pregunta es una presentación por parte del entrevistado, para que los lectores te puedan conocer un poco.

Entrevista En Diferido: ¿Te podrías presentar en unas líneas?

Rodolfo Pilas: Difícil en unas líneas cuándo empecé mi vida laboral en los inicios de la computación personal, monté y gestioné un BBS (previo a Internet), fui activista y difusor del Software Libre en mi País y la región, dicté conferencias en la mayoría de los países de América Latina y actualmente estoy subido a la cultura DevOps pero vamos a ver cuántas líneas queda mi presentación:

En tecnologías de la información me desempeño como DevOps en la empresa Moove-It, que es una moderna y divertida (me divierto mucho trabajando!) empresa de desarrollo con equipos en Uruguay, EE.UU., Colombia y Argentina. Además soy Fundador y Director de Rootway Internet Ltd. una empresa de servicios sobre Internet con oficinas en Uruguay, Argentina y EE.UU.

En el ámbito académico soy Profesor a cargo de las materias Administración Linux y Cloud Computing en la Facultad de Ingeniería y Tecnologías de la Universidad Católica de Montevideo. Soy RedHat Certified Engineer (RHCE) y entrenador y RedHat Certified Examiner (RHCE) y durante muchos años me desempeñé como entrenador de técnicos en distintos institutos de Uruguay y Argentina.

Por último, mi actividad más longeva, es la de Agente de la Propiedad Industrial donde comencé de adolescente junto a mi padre y actualmente soy en Director de un Estudio de Propiedad Intelectual con oficinas en Uruguay y Argentina.

Además mantengo mi blog personal desde 2004 <http://pilas.guru>, actualmente estoy incursionando (desde hace un par de años) en mi proyecto de podcasting <http://deployando.me> que me ha resultado todo un desafío de auto-descubrimiento personal. Y, cuando hago deporte, me emociona y relaja la arquería.

Además tengo la dicha junto a mi esposa mantener una familia con mi hija de 12, mi hijo de 10 y una gata gorda.

Bueno, creo que no quedó en pocas líneas.

Blog de J.A. Jimenez Toro, rooteando.com

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

EED: Para empezar , unas cuantas preguntas cortas para conocerte mejor tu entorno tecnológico.

¿Ordenador principal?

¿Sistema operativo?

¿Ordenador de escritorio o portátil?

¿Consola de terminal que utilizas?

¿Herramienta de monitorización de red?

¿Docker, Vagrant o Wmware?

Dulce de leche ¿argentino o uruguayo?

RP: Hace 6 años tengo mi MacBook Pro (es mi tercer MacBook Pro) con MacOSX *echo \$HOSTNAME: emilia* por Emilia Clarke.

Leer correo, navegar y escribir este texto lo hago desde MacOS X pero la mayoría de mi trabajo es en GNU/Linux al que accedo siempre por SSH, la mayoría en servicios de Cloud o en VM en mi MacBook. GNU/Linux prefiero Debian, pero uso también CentOS o AmazonLinux.

En mi casa no hay ordenador de escritorio, todos portátiles incluso los de mis hijos. En mi trabajo cada compañero tiene su portátil. Creo que los ordenadores de escritorio tienen casos de uso cada vez más acotados.

Utilizo iTerm como consola, aunque no soy usuario avanzado, y como shell utilizo FiSH, pero mi lenguaje de programación es BaSH.

Mis requisitos de monitoreo de redes son básicos: con mtr, dig y curl suelo tener suficiente. Para monitoreo de servidores y servicios uso Xymon Monitor.

Me encanta Vagrant. Mis cursos de Linux y Cloud Computing están basados sobre Vagrant, mis alumnos aprenden Vagrant en mis cursos. Utilizo Vagrant para prototipar todo pues mi MacOS no tiene nada extra instalado. Hay que hacer algo en Ansible: vagrant up. Hay armar algo con Docker: aprovisiono una VM con Vagrant y dentro de ella armo el docker-compose.yml. Probar nuevo software: vagrant up. Armar una solución de respaldo: vagrant up. Tres ediciones del podcast DeployandoMe las dediqué a Vagrant.

Contenedores también es una tecnología maravillosa. Tengo muchos servidores con LXC y OpenVZ. Docker es la implementación de contenedores "de moda" y no se si seguirá siéndolo; pero "contenedores" tiene un largo futuro.

VMWare es "software privativo" y evito comprometer proyectos con infraestructura privativa, antes dispongo de KVM, VirtualBox, LXC, Docker u OpenStack. Además discrepo con el sistema de licenciamiento extorsivo de VMWare.

Esa última pregunta es la leche (\o/): en Argentina vas a encontrar las mejores medialunas (croissants), pero si quieres dulce de leche busca el uruguayo.

EED: Dentro de tu variada actividad , me voy a centrar en tu trabajo como DevOps.

Blog de J.A. Jimenez Toro, rooteando.com

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

Una actividad relativamente nuevo y desconocida para mucha gente, en lo relativo a esa cultura de DevOps.

¿Qué mejorarías?

¿Qué añadirías?

¿Qué eliminarías?

RP: DevOps es una cultura, una forma de ver las cosas que favorece la integración de los profesionales en tecnologías de la información (Dev y Ops), mediante el uso de los mismos procesos, estrategias, herramientas de coordinación y colaboración.

Lo que mejoraría sería dotar a los cursos formales de Dev más información sobre los problemas y soluciones de Ops. Es frecuente encontrarme con desarrolladores que no entienden cómo funciona un servicio de nombre (DNS).

En la misma línea educativa, añadiría sólidos cursos de infraestructura, administración de sistemas y operaciones. Si bien cursos de estos temas existen desde hace tiempo, suelen carecer de la visión Dev; son cursos que no explican como usar un repositorios de versiones git (los egresados siguen generando archivos .bak o ` .old`). Los cursos existentes carecen de conceptos como "infraestructura cómo código", "orquestación", "microservicios", "natural computing", etc. y ni hablemos de conceptos como desarrollo ágil aplicado a operaciones.

Eliminaría algo que no es eliminable: el hype. DevOps es emergente y tenemos desde gente que con buenas intenciones que entiende a medias, hasta gente que lo entiende y no tiene oportunidad de aplicarlo; pero lo peor son los "vende humo", los productores de hype, que te llevan por caminos "de moda". La solución es la madurez tecnológica, pero tecnologías de la información en general es un campo nuevo e inmaduro (comparado con tecnologías centenarias como la fotografía o la aviación; o la de miles de años como la arquitectura), todos los días encontramos cosas nuevas y todos los días compramos humo.

EED: La empresa que estoy mantiene múltiples clientes, con múltiples proyectos, múltiples tecnologías, múltiples proveedores (Cloud, SaaS, data analytics, etc.) y distintos requisitos; aunque se centra principalmente en las áreas de `_health_` y `_banking_`. Eso lleva a que mi rol es ayudar a múltiples equipos y suelo cambiar de camiseta varias veces por jornada.

Actualmente estoy trabajando en un proyecto que me tiene muy entusiasmado: junto con un grupo de desarrolladores (somos cuatro ingenieros) tenemos como objetivo generar patrones automatizados y sustentables para el desarrollo de aplicaciones que involucran desde el nacimiento del proyecto (de una aplicación) hasta su puesta en producción.

La idea que aspiramos es que un desarrollador que inicia una aplicación X lenguaje y se clona un repositorio git. Llena un archivo con "variables" y ejecuta `docker-compose up` para poder empezar a desarrollar la aplicación ya en ese lenguaje. Allí debería tener todo lo básico que necesita (base de datos, estructura y librerías para generar un API de

Blog de J.A. Jimenez Toro, rooteando.com

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/4.0/)

backend y un frontend). Cuando el desarrollo madure, se crea un entorno de staging en forma automática en la cloud que se decida publicarlo, se clona también automáticamente el mismo repositorio, se llenan las variables correspondientes y docker-compose up pero ahora para tener todo funcionando con su proxy reverso, balanceo de carga, gestión de logs, políticas de seguridad, etc. como si fuera el ambiente de producción. Para esto estamos utilizando cloud-init, docker y ansible.

El desafío es grande, pues no somos una empresa de un problema acotado, sino que se atienden múltiples realidades y requisitos según los clientes y los proyectos. Además tenemos los sistemas heredados o que ya son proyectos maduros de la cartera, que seguirán con sus metodologías asentadas. Por todo ello, los patrones tienen que ayudar a la normalización, pero no deben representar un obstáculo para entregar la mejor aplicación que se pueda desarrollar.

En un futuro, luego de generalizar el uso de estos patrones dentro de la empresa apuntaremos a tecnologías que nos abstraigan más y nos permitan ser agnósticos de proveedores, como Kubernetes o Mesos.

Mi flujo de trabajo es muy parecido al de un desarrollador que esté en dos o tres proyectos en forma concurrente. Mis tareas nacen de un backlog del proyecto y se ejecutan dentro de un sprint, accedo al tablero para mover mis tarjetitas y al git del para disponer del código, genero mis aportes en infraestructura y automatización en una rama del repositorio y hago mis Pull Request, que pasan por peer review antes del merge. A veces participo de las daily meetings y generalmente suelo estar en las weekly meetings. A veces es tan puntual mi aporte (automatizar un backup, certificados SSL, actualizar) que solo recibo una petición de ayuda y la confirmación de la inclusión de mi certificado SSH para acceder al ambiente de la aplicación.

Vengo de operaciones, visualizo DevOps como el desarrollador que programa en esos intérpretes raros como bash, ansible, terraform, cloudformation y que le piden desarrollos de módulos para los proyectos de las aplicaciones; me siento parte del equipo, comulgo con el objetivo del proyecto y hago aportes desde mi área de experiencia.

EED: Tienes perfil académico como profesor universitario, respecto a ese trabajo.

¿Cómo enseñas?

¿Cómo te gustaría enseñar?

RP: Toda mi actividad académica ha sido enseñar tal cual he aprendido, aplicando las mejores estrategias, pero siempre siguiendo con lo que conocía: como me enseñaron a mi. El docente explica la teoría y luego los alumnos ejecutan ejercicios y rinden pruebas al final del semestre para verificar que resuelven ejercicios similares.

Blog de J.A. Jimenez Toro, rooteando.com

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

Pero la Universidad Católica del Uruguay hace ya algunos años está en un proceso continuo de mejora de la docencia universitaria y mediante distintas estrategias de enseñanza y evaluación ahora educamos por competencias, es decir los cursos los diseño basado en qué competencias debe tener quién egresa del curso.

Como docente no es una tarea fácil, no solo me sacó de mi zona de confort mientras estoy en clase, sino que los mecanismos de evaluación y los objetivos de la evaluación son otros, a veces más sutiles o menos objetivos.

Así que ahora mi enseñanza es muy práctica: planteo desafíos y los alumnos deben buscar resolverlos y mi tarea es hacer una "tutoría" del proceso de resolución. Cuando encuentran una solución en Stackoverflow ahí intervengo para abordar el tema y explicarlo más en profundidad, para que sepan qué están haciendo.

Y cómo más me gusta enseñar es en base al error, me gusta que mis alumnos se equivoquen. Muchas veces veo que están siguiendo un tutorial de Youtube que los va a complicar y los dejo continuar, a veces hasta con un comentario de apoyo "interesante solución!, siguan por ahí", en un esfuerzo de llevarlos a donde todo se descarrila, pierden tiempo e invierten esfuerzo en vano. Y más me gusta cuando ellos mismos se dan cuenta que deberían buscar nuevos enfoques al problema pero si no, comienzo a explicarles teoría y fundamentos hasta que se dan cuenta donde cometieron los errores. Esas son las clases que más me gustan, porque se que el error es mucho mejor docente que yo y suelen acordarse después de mucho tiempo de sus metidas de pata.

EED: Esta pregunta es de parte de un lector de las entrevistas.

Vivimos en una época donde los jóvenes nacen con la tecnología al alcance de la mano desde su nacimiento. Respecto a estos jóvenes.

¿Crees que saben más o menos de tecnología que otras generaciones?

RP: No creo que sea un análisis adecuado comparar la cantidad o calidad de conocimiento de una generación con otra, pues hablamos de contextos y situaciones diferentes.

Cuando era joven, la computación estaba en sus inicios y necesitabas conocer el throughput del bus de datos si querías hacer cosas divertidas con tu computador. O te gustaba el cacharreo o no sabías de computación.

Hoy en día puedes hacer cosas mucho más divertidas sin siquiera tener el computador, no necesitas ni entender el cacharreo para análisis de BigData, Redes Neuronales y AI (de las mate no zafás, pero ese es otro tema).

Soy de la idea que tecnologías de la información es un área para estudiar toda tu vida. Un técnico que consigue un trabajo por sus conocimientos y durante dos o tres años

Blog de J.A. Jimenez Toro, rooteando.com

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

no se actualiza (con cursos, lecturas permanentes, pruebas nuevas) queda en la categoría de "ignorante", o por lo menos el siguiente puesto no lo tendrá por lo que sabe.

Además es tanta la afluencia de nuevos conocimientos que el estudio se produce "por descarte": tengo una biblioteca de libros con temas que me gustaría abordar, que es más grande que lo que podría aprender si solo me dedicara a estudiar.

En este contexto y para el joven de hoy, el conocimiento en si mismo tiene importancia pero no es fundamental, el conocimiento te hace ir más rápido pero nada más. Es como saber las noticias del diario de hoy o el último chiste político.

Actualmente importa más llegar a soluciones, saber trabajar en equipos (porque solo no se resuelven las cosas), poder colaborar. Importa si eres curioso, si te agrada auto-superarte, estudiar más o abordar nuevos temas. Importa también cómo vives tu propia ignorancia: si te gusta sentirte ignorante y enfrentarte a tus límites o prefieres tu zona de confort. Es importante saber dar la bienvenida a los fracasos y errores para aprender de ellos, cómo manejas la resiliencia para seguir luego de un porrazo que eventualmente te dejó mal parado.

Por supuesto que siempre van a estar los cargos que requieren un nivel de conocimiento experto, donde es importante casi exclusivamente el conocimiento. Pero las tecnologías de la información se han abierto en un abanico amplio y cambiante de opciones que permite a los jóvenes desarrollarse profesionalmente con conocimientos menos profundos y más flexibles (a esto agregar el detalle que ni con los chinos y los indúes hemos colmado la demanda de puestos de trabajo).

Si tengo que elegir un compañero de equipo prefiero un joven con estas características que comento (por encima de otro con profundos conocimientos del tema que hay que solucionar), pues juntos tenemos mejor oportunidad de obtener un mejor resultado.

EED: Ahora mismo hay múltiples tecnologías, servicio o productos disponibles, algunos mas útiles que otros. Si alguien viene y te pide consejo sobre que aprender para el futuro no muy lejano.

¿Qué crees que debería aprender?

¿Qué crees que no se debería aprender?

RP: Si alguien me dice que quiere dedicarse a lo mismo que yo y consulta qué le sugeriría aprender, le diría que necesita algunos conocimientos profundos de:

* al menos un lenguaje de programación y que ese lenguaje sirva en al área donde piensa desarrollar actividad. Para administrar sistemas Python o Go, o como mínimo Bash.

* contenedores (docker, pakman, lxc) es algo que también se debe saber y entender.

Orquestar contenedores en clusters, mejor aún (docker swarm, kubernetes).

* Un administrador de configuración como Ansible, Chef, Puppet, o Salt.

* OpenSSH y entender todo su sistema de certificados.

Blog de J.A. Jimenez Toro, rooteando.com

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

Y que va a ser muy útil tener conocimientos de:

- * Herramientas y metodologías vinculadas al desarrollo ágil y al desarrollo en equipos (MVP, Scrum, Vision Board, etc.)
- * Infraestructura de redes (red, sub-redes, ruteo, firewall)
- * Pipelines, automatización y DevOps.

Y por último, mucha gente habla de las 10.000 horas de práctica para dominar un arte y en tecnologías de la información es igual: requiere de práctica. Siempre sugiero auto-imponerse desafíos de casos prácticos (cosas que te sean útiles), enfrentarlos y resolverlos. Personalmente los considero atajos de aprendizaje a los cursos para dominar los un temas. Los cursos te van a mostrar el bosque y las herramientas, pero con un caso práctico vas a talar árboles o a construir un camino a través del bosque.

Sobre qué no aprender pues, al software privativo no le dedico mis horas o mi esfuerzo personal y me abstengo de promocionarlo o sugerirlo. Es decir, que cuando he hecho cursos de herramientas o sistemas privativos ha sido porque el proyecto lo requería y lo aplicaba inmediatamente, pero no es algo a lo que le dedicaría la regla de las 10.000 horas y, por consecuencia, no soy experto en software privativo. (claro que este ha sido MI path profesional, mucha gente se desarrolla a pleno en el camino del software privativo).

Pero cursos de administración GNU/Linux que te lleven por las opciones del man, dedicando tiempo a cada comando ya no lo considero actual, pues para algo está el man. Lo mismo si los cursos se detienen en configurar LDAP o crear un Raid o configurar un kernel para compilar, no porque sea algo obsoleto ni mucho menos, sino porque el día que lo necesite aplicar, lo buscaré. Así que prefiero los cursos que abordan los temas que indiqué arriba, en detrimento de estos últimos.

EED: En el ámbito tecnológico cada vez se automatizan mas tareas, el proyecto en el que estas trabajando esta enfocado a eso. Pero pensemos en un futuro algo mas lejano, las tareas que realiza un administrador de sistemas o un DevOps.

¿Pueden ser totalmente sustituidos por software?

¿En que grado crees que el factor humano será necesario para esas funciones?

RP: No me gusta hacer futurología, pero se puede reflexionar basado en la experiencia pasada. No creo en el futuro gobernado por software y automatismos que adquieren conciencia para dominarnos o que nos transforman en "gordos en reposera con gafas de realidad aumentada", por el contrario, soy de los que adhieren la visión pesimista: un mundo altamente contaminado, o un virus que nos diezma y sobreviven cucarachas, o una sociedad fracturada basada en profundas desigualdades.

El otro día escuchaba un podcast comentando que en años estamos más cerca de Cleopatra que lo que ella está del antiguo Egipto. Así que el futuro lejano para un Faraón, para Cleopatra o para mí mismo, seguro no será el futuro lejano que

Blog de J.A. Jimenez Toro, rooteando.com

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

Pero el futuro cercano será cómo el pasado cercano pero más acelerado (al menos mientras se pueda respirar la atmósfera o no aparezca el virus). Es decir, en tecnologías de la información va a ser necesario actualizarse continuamente y cada vez centrarnos más en una disciplina y descartar conocimiento.

Entonces, la posibilidad de ser sustituido solo será cuando dejes de estudiar, de perfeccionarte (como lo es actualmente, pero más rápido).

Sustituciones se producen ya, la necesidad de perfeccionarse o re-convertirse existe en este 2019. Existieron en el pasado, pero ahora se han acelerado (más disciplinas y más rápido).

El cambio, los procesos de obsoletización, la necesidad de re-convertirse y estar actualizado son parte natural e inseparable de las tecnologías de la información. Cuando daba cursos en 1er semestre les comentaba a los alumnos: "si no te gusta estudiar hasta que te mueras, busca otra profesión" (tal vez por esos comentarios ahora mis cursos son de 8vo semestre).

Por ejemplo, mi dentista trabaja el cuerpo humano y los cambios y mejoras siempre estarán vinculados a hueso y carne, que no cambia. Nosotros actuamos principalmente con intangibles, con construcciones intelectuales y a medida que seamos más, cambiaremos más rápido todo el stack de conocimiento, obsoletizando otro tanto, y aumentaremos exponencialmente la dimensión de las áreas en las que podemos desarrollarnos como profesionales.

EED: En el ámbito tecnológico cada vez se automatizan mas tareas, el proyecto en el que estas trabajando esta enfocado a eso. Pero pensemos en un futuro algo mas lejano, las tareas que realiza un administrador de sistemas o un DevOps.

¿Pueden ser totalmente sustituidos por software?

¿En que grado crees que el factor humano será necesario para esas funciones?

RP: No me gusta hacer futurología, pero se puede reflexionar basado en la experiencia pasada. No creo en el futuro gobernado por software y automatismos que adquieren conciencia para dominarnos o que nos transforman en "gordos en reposera con gafas de realidad aumentada", por el contrario, soy de los que adhieren la visión pesimista: un mundo altamente contaminado, o un virus que nos diezma y sobreviven cucarachas, o una sociedad fracturada basada en profundas desigualdades.

El otro día escuchaba un podcast comentando que en años estamos más cerca de Cleopatra que lo que ella está del antiguo Egipto. Así que el futuro lejano para un

Blog de J.A. Jimenez Toro, rooteando.com

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Faraón, para Cleopatra o para mí mismo, seguro no será el futuro lejano que tendremos.

Pero el futuro cercano será cómo el pasado cercano pero más acelerado (al menos mientras se pueda respirar la atmósfera o no aparezca el virus). Es decir, en tecnologías de la información va a ser necesario actualizarse continuamente y cada vez centrarnos más en una disciplina y descartar conocimiento.

Entonces, la posibilidad de ser sustituido solo será cuando dejes de estudiar, de perfeccionarte (como lo es actualmente, pero más rápido).

Sustituciones se producen ya, la necesidad de perfeccionarse o re-convertirse existe en este 2019. Existieron en el pasado, pero ahora se han acelerado (más disciplinas y más rápido).

El cambio, los procesos de obsoletización, la necesidad de re-convertirse y estar actualizado son parte natural y inseparable de las tecnologías de la información. Cuando daba cursos en 1er semestre les comentaba a los alumnos: "si no te gusta estudiar hasta que te mueras, busca otra profesión" (tal vez por esos comentarios ahora mis cursos son de 8vo semestre).

Por ejemplo, mi dentista trabaja el cuerpo humano y los cambios y mejoras siempre estarán vinculados a hueso y carne, que no cambia. Nosotros actuamos principalmente con intangibles, con construcciones intelectuales y a medida que seamos más, cambiaremos más rápido todo el stack de conocimiento, obsoletizando otro tanto, y aumentaremos exponencialmente la dimensión de las áreas en las que podemos desarrollarnos como profesionales.

EED: En la actualidad hay multitud de servicios Cloud, empresas como Amazon, Google o Microsoft están constantemente desplegando nuevos servicios para su Cloud, que se unen a una larga lista de servicios disponibles. Pero desde tu experiencia usando estos servicios..

¿Cuál crees que será la siguiente evolución, o revolución, de los servicios en Cloud?

RP: Tal cual comentas, constantemente se están agregando nuevos servicios y eso es lo primero que hay que entender: hay grupos de técnicos y gente altamente capacitada y que ve el mundo redondito (entienden las realidades de distintas sociedades, culturas y mercados) que están pensando qué puedes necesitar y cómo te lo pueden vender As A Service, es decir que lo uses mediante un API y que se pueda contabilizar el consumo para facturar por hora o por minuto.

Esas personas (que piensan el As A Service) están despegadas de nuestras realidades: suelo comentarles a mis alumnos, por ejemplo, de servicios como Amazon Snowball que representa el mayor ancho de banda que se puede lograr en transferencia de datos (que es un array de discos enviado por avión o con semirremolque); o el servicio

Blog de J.A. Jimenez Toro, rooteando.com

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

de ejecución de instrucciones de programas a demanda, que abrió el camino tecnológico del Serverless; o servicios tan particulares como "auditores automatizados" que revisan reglas de conformidad de certificaciones a lo largo y ancho de tus despliegues en la nube para que las auditorías no tengan nada para decir. Y esto solo por nombrar algunas cosas que YA ha pensado esta gente.

Imposible que este humilde entrevistado pueda decirte cuál será la siguiente evolución/revolución, ¿acaso mejoras en interfaces de voz? ¿implantes 5G para estar siempre conectados? Hay cosas en la que se están jugando partidos muy interesantes: Realidad Aumentada, Game As A Service, Auto-conducción, Internet of Medical Things, que cuando salgan del ámbito de los "precursores" y lleguen a la sociedad van a producir revoluciones profundas.

EED: Hay multitud de soluciones de virtualización ,de diferentes tipos y desde la llegada de Docker, todo se ha vuelto mas complejo por la cantidad de opciones disponibles.

**¿Podrías explicarme las diferencias entre Docker,VirtualBox y Vagrant?
¿En que escenarios utilizar cada opción?**

RP: Hay dos ramas: *virtualización* y *contenedores* y son fáciles de distinguir. En VIRTUALIZACION tienes software (instrucciones) que están "dibujando" hardware para que otro software crea que está corriendo con recursos reales. Hay varios paradigmas de virtualización: emulación (Mame), paravirtualización (Xen), virtualización completa (QEmu), de procesador (JVM), de instrucciones (Wine), y nativa (KVM, Xen); pero todos suponen un código que "engaña" a otro. En CONTENEDORES (que también lo llaman "virtualización de sistema operativo" y complica un poco...) lo que hay es aislamiento, es decir hay un software (el kernel) que se dedica a "aislar" otro para que crea que está ahí solito, dueño y señor de todo, pero solo. En contenedores no se dibuja hardware (bueno, la red sí, pero es un funcionamiento nativo del kernel), el hardware es el que hay, el real (bueno, si estás corriendo contenedores en máquina virtual, ese que te dibujan).

Por esto en virtualización vas a encontrar conceptos como recursos del host y sobre venta, que van a definir cuántas máquinas virtuales puedes levantar con qué recursos. Y en contenedores vas a encontrar conceptos como cuotas y etiquetas, que te van a dar una densidad mayor de contenedores para el mismo hardware.

Por esto es difícil y costoso virtualizar dentro de una máquina virtual (nested virtualization) y necesitas la cooperacion del procesador real (eso que habilitas en el EFI). Pero es trivial poner a correr un contenedor dentro de un contenedor y si quieres dentro de éste pones a correr otro contenedor.

En virtualización KVM y Xen son hipervisores (administradores de VM) nativos, es decir

Blog de J.A. Jimenez Toro, rooteando.com

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

corren directamente sobre el hardware real. Openstack es un gestor de cloud (API + validación + accounting) que utiliza KVM, Xen y otros hipervisores para entregar instancias (VM). Virtualbox es un hipervisor hosteado, es decir corre sobre un SO que es el que corre contra el hw real.

Vagrant es un software escrito en Ruby para gestionar el ciclo de vida (up, halt, destroy) de máquinas virtuales, que utiliza como proveedor virtualbox, vmware e hyper-v en forma nativa o aws, azure, kvm, openstack, parallels, etc. mediante plugins. Además, utiliza como proveedor nativo docker o, con plugin, lxc para gestionar contenedores.

Docker no es la única implementación de contenedores. Contenedores es una funcionalidad del kernel Linux (que NO fue inventada en Linux). Docker es a contenedores, como iptables es a netfilter. También se pueden tener contenedores de tipo LXC, y si se quiere con una arquitectura semejante a docker con LXD.

Existe un movimiento que busca tener una definición de consenso para contenedores: Open Container Initiative (OCI), que busca coordinar los esfuerzos hacia objetivos comunes y que el mundo de los contenedores no sea una batalla por ver quién es el estándar.

¿qué aplicar en qué caso? Como primer respuesta podemos decir que si vamos a usar distintos sistemas operativos en el mismo hardware: virtualización. Si vamos a usar solo GNU/Linux y aplicaciones que corren sobre él, pues contenedores. A partir de ahí habrá ponderar disponibilidad, conocimientos, oportunidad, tiempo de entrega y todas esas cosas que se deben analizar para una solución.

EED: Como última pregunta de la entrevista , una pregunta un poco diferente

¿Qué te hubiera gustado que te preguntase? Evidentemente debes responderla.

RP: Como contaba en la presentación, estoy en esto desde antes de Internet, como adolescente tuve un BBS donde usuarios se conectaban telefónicamente por modem para chatear e intercambiar archivos; toda una _movida_ que hoy parece prehistoria.

Así que preguntas como: ¿qué te hubiera gustado que existiera actualmente que ya no existe?, ¿qué te hubiera gustado que se siga desarrollando que ha quedado obsoleto? o ¿qué extrañas de aquellos tiempos? (o sea una pregunta para los amantes del Steampunk \o/).

Pues sí, hay algo que extraño mucho, tal vez porque en mis tiempos juveniles lo viví con mucho entusiasmo y hoy lo veo como "en colores pastel", sin decisión, sin aquella fuerza.

Blog de J.A. Jimenez Toro, rooteando.com

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

Me refiero a la "militancia tecnológica".

Hubo un tiempo en que la tecnología se militaba, existían los grupos de usuarios, las discusiones de boliche, los foros de discusión con preguntas que despertaban pasiones ("¿cuál es la mejor distribución Linux?"), cuando se veía que la tecnología era parte del cambio social y las personas se comprometían con ello y se sentían parte de un movimiento.

Por ejemplo: hace unos meses llegó a mis manos el manifiesto del movimiento BAARF (Battle Against Any Raid [Five]) de 2003, y me emocioné! pues ponía en blanco y negro esta militancia que ya no se ve (o está atenuada, o mantenida por unos viejos pelados con panza, como yo). Este movimiento BAARF eran unos jóvenes técnicos que, cervezas por medio, se dieron cuenta que el Raid no era una solución "sana" y se comprometían a concurrir a las Usenix a manifestar su desconformidad y a organizar BAARF Party para difundir sus ideas. (loquitos, si, pero con corazón!)

Ahora todo es mas light, el cambio social se canaliza mediante la huella de CO2, si fumo electrónico o uso bolsas de papel. La tecnología produce cambios sociales como siempre, pero ya no se ven movimientos y compromisos a favor o en contra de ella.

Converso con mis alumnos sobre los valores éticos del Software Libre, sobre las ventajas sociales del desarrollo Open Source y veo sus ojos que me miran entre asombrados y con sospecha de que soy un tipo raro.

Encuentro muchos motivos para que ahora las cosas sean como son y no me parecen mal, pero también extraño esas reuniones, esos eventos, esos movimientos donde se militaba la tecnología.

EED: Hoy es el último día de la entrevista y es el momento de la despedida, pero antes me gustaría agradecerle tu participación, espero que haya sido una experiencia interesante y entretenida para tí.

También si quieres, puedes poner tus métodos de contacto y si quieres publicitar algún podcast,web,evento u otro proyecto tienes este espacio para hacerlo.

Por último, me gustaría que me propusieras a una persona que tú creas que estaría dispuesto a participar en una futura entrevista.

Ha sido un placer , hasta la próxima.

RP: El agradecido soy yo. La experiencia ha sido muy interesante, me he divertido mucho. Espero haber aportado al pensamiento crítico de quienes siguen este canal.

Invito a todos a escuchar mi podcast, aunque también entiendo que es muy de nicho y no a todos les va a interesar. Mi podcast es DeployandoMe, cada 15 días, planteo un Blog de J.A. Jimenez Toro, rooteando.com
Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](http://creativecommons.org/licenses/by-nc-sa/3.0/)

tema vinculado a las tareas de sysadmin o devops. En diez minutos abordo un problema común y una solución, siempre con software libre. La forma que sugiero para escuchar podcast es mediante una aplicación que permita suscribirse y mi podcast lo encuentran en la mayoría de los catálogos que revisan esas aplicaciones (iTunes, Google Podcast, iVoox, Sticher, TuneIN), también en el sitio web <https://deployando.me>, en Youtube y en Spotify. Siguen el podcast en Twitter @deployandome, en Instagram en @deployandome y pueden contactarnos por correo a podcast@deployando.me.

Mantengo también un blog <https://pilas.guru> donde cada tanto publico algún artículo, pero ¿vieron que ultimamente los blogs están un poco abandonados? ¿será por tema de moda?

De forma más personal me pueden seguir y contactar por twitter en @pilasguru o por el formulario de contacto del blog que llega directo a mi correo personal.

Para una posible futura entrevista @JoseAJimenez, te propongo a mi amigo Anahuac de Paula, @anahuac aquí en Telegram y @anahuacpg en Twitter. El es del nordeste de Brasil, de la ciudad de Joao Pessoa (ahí justo en la punta de América del Sur contra Africa). Anahuac es autor de los libros OpenLDAP* <http://www.openldap.com.br> entre otros. Mantiene ese un fuerte compromiso social con el software libre, la privacidad y la vigilancia masiva. Además es de esas personas alegres y positivas con las cuales da gusto compartir y dialogar casi de cualquier tema de la vida.

El honor y el placer ha sido todo mío. Gracias y acá seguiré leyendo con interés todos los próximos entrevistados.