

Entrevista En Diferido 31: Sergio Infante

25 de Agosto de 2020 a las 01:06

Empezamos una nueva entrevista con Sergio Infante, programador de Python, podcaster y mucho mas.
Pero mejor que se presente el mismo...

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

Sergio Infante: Hola. Me llamo Sergio Infante, tengo 35 años y nací en Perú. Pase gran parte de mi vida en Huancayo, lugar donde nací, y donde encontré en la tecnología y el internet una pasión por crear cosas y también por enseñar a otras personas, estudié Ingeniería Informática y estuve involucrado en proyectos de software libre y open source desde estudiante, eso me dio una fascinante perspectiva del mundo de la tecnología, el poder interactuar y compartir la pasión por la tecnología con otras personas con objetivos similares a los míos.

Llegue a ser profesor de un instituto de tecnología y también profesor universitario, pero por perseguir otras oportunidades laborales me mude a Lima hace ya hace varios años, al ser la capital las oportunidades de seguir trabajando en tecnología se incrementaron, y es así como he tenido el privilegio de haber trabajado con empresas de todo el mundo, y haber desarrollado proyectos que van desde el desarrollo móvil, la automatización hasta proyectos de inteligencia artificial y ciencia de datos.

Por temas laborales deje de enseñar presencialmente hace 7 años, y esa inquietud que aún la tengo se tradujo en crear contenido para enseñar a través de medio digitales, empecé con un blog donde publicaba tutoriales, y luego colaboré con otros blogs, con artículos, guías y contenidos, grave varios podcasts pero no encontraba aún la formula que funcione.

El año pasado decidí afrontar el reto con otras estrategias, así que me animé nuevamente a grabar un podcast e intentar cosas distintas como también publicar videos en youtube, al día de hoy tengo decenas de oyentes en mi podcast y cientos de seguidores en mi canal de youtube, siento que estoy encontrando nuevamente esa manera de juntar mis dos pasiones, enseñar y programar.

EDD: Para empezar , unas cuantas preguntas cortas para conocerte mejor a tí y tu entorno tecnológico.

¿Qué ordenador utilizas habitualmente?

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/)

¿Sistema operativo?**¿IDE o editor para programar?****¿Lenguaje de programación?****¿Cámara de fotos?****¿Micrófono que utilizas?****¿Libro de programación?****SI: ¿Qué ordenador utilizas habitualmente?**

Actualmente uso tres ordenadores, para el trabajo uso una macbook del 2015, core i7 con 16gb de ram, para mis proyectos personales, iniciativas y emprendimientos propios uso una macbook del 2017, core i7 16gb de ram y para editar audios y videos uso una desktop, mainboard ASUS TUF B365 plus gaming con un procesador intel core i5, 16 gb de ram doble disco duro, una tarjeta gráfica NVIDIA GTX 1660 super.

¿Sistema operativo?

Uso Linux, macOS y Windows, cada uno en los aspectos en que los considero fuertes, para mi la fortaleza de Linux radica en los servidores, todos mis proyectos personales y emprendimientos están desplegados en Linux, macOS me funciona muy bien para programar y no tener que estar preocupándome por drivers, ni incompatibilidad de formatos, me permite enfocarme en el desarrollo y windows lo uso para editar videos, audios, imágenes y también para jugar.

¿IDE o editor para programar?

Prefiero una IDE, me permite ahorrar mucho tiempo automatizando varios procesos y en este aspecto PyCharm para mi es una maravilla, ha sido mi IDE favorito desde hace varios años y creo que lo seguirá siendo en los próximos, por lo general cada vez que tengo que tocar algún lenguaje de programación que no sea python, le echo un vistazo a los IDE que tiene JetBrains, son muy buenos, a excepción de desarrollar con swift en el que prefiero Xcode

¿Lenguaje de programación?

Mi favorito es Python, por la versatilidad que me da, de poder usarlo para varios propósitos, desde backend hasta automatización, entre otros. Si tuviera que elegir un segundo lenguaje favorito sería Swift, y si hay uno que quiero darle mas tiempo para aprender ese sería Rust.

¿Cámara de fotos?

Es algo que no conté en esta entrevista aún y es que una pasión paralela, para mi es la fotografía, sino hubiera sido programador, definitivamente hubiera sido mi profesión, he probado varias cámaras, tengo en mente en algún momento tener una Fujifilm o una Sony, pero mientras eso no sucede mi favorita es la canon EOS M5, es una mirrorless bastante funcional, versátil, la estoy usando con diversos objetivos, algunos que no son mirrorless y también un par de otros fabricantes, la verdad que estoy muy contento con los resultados.

¿Micrófono que utilizas?

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/)

Actualmente tengo tres, y los uso por diversas razones, tengo un Shure Rode smartLav+ que lo uso con una grabadora portátil Tascam DR-07X, me esta sirviendo mucho sobre todo cuando necesito movilizarme. También uso un micrófono HyperX QuadCast cuando estoy en mi pequeño estudio, y tengo que grabar tutoriales y videos cortos y por último un Tonor Q9 que fue el micrófono con el que retome el podcasting y que tiene excelentes resultados a pesar de su bajo precio, lo uso para grabar contenidos extensos. Tengo en mente probar el Shure SM58 así que tal vez en unos meses tenga un nuevo micrófono favorito.

¿Libro de programación?

The Pragmatic Programmer, es el libro que siempre recomiendo leer, me gusta como enfoca los temas que son importantes en el desarrollo de software, donde no solamente es programar, sino ser eficientes, productivos, generando valor para el proyecto constantemente, es un excelente libro.

EDD: Un pregunta fija para todos los entrevistados que solo cambia el contexto, como eres programador de Python...

¿Qué añadirías a Python?

¿Qué modificarías de Python?

¿Qué eliminarías de Python?

SI: ¿Qué añadirías a Python?

Una mejor integración con el manejo de GPU, y multiprocesamiento para que sea mucho más solido en computación paralela, tener soporte nativo para el procesamiento paralelo de GPU haría que Python se consolide como el lenguaje de programación ideal para machine learning.

¿Qué modificarías de Python?

La manera como se escriben los decorators, para hacerlos mas fáciles de leer, en lugar de lo que actualmente se puede encontrar, expresiones complicadas que muchas veces el mismo autor del código le dificulta explicarlo.

¿Qué eliminarías de Python?

El weakly typed provoca muchos errores sobre todo con programas extensos, para evitar que suceda hay que implementar controles, que a veces hacen tedioso el lidiar capturando errores por tipado, si fuera un poco mas estricto, sería más robusto.

EED: Tienes un podcast llamado neosergio.fm (https://www.ivoox.com/podcast-neosergio-fm_sq_f1763203_1.html) donde hablas de programación , tecnologías, fotografía y otros temas.

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/)

¿Qué te ha aportado como programador grabar un podcast?

SI: En el libro *The Pragmatic Programmer* existe una historia que hace referencia a un patito de goma y un programador que siempre lo llevaba consigo, y lo usaba para depurar su código, de forma tal que tenía que explicarle al patito línea por línea que es lo que hacía el código, a esta técnica se le conoce como la depuración del patito de goma o en inglés como Rubber Duck Debugging.

Existen cientos de artículos al respecto, explicando sus resultados, yo siento que para mí el podcasting es una forma de Rubber Duck Debugging, ya que no se que audiencia voy a tener en cada episodio, si son muy técnicos o no, por lo que me veo obligado a buscar la manera mas entendible de explicar las cosas, y eso me ha ayudado en la manera que escribo un componente o un modulo, trato de pensar que desconozco la audiencia que leerá lo que escribo y lo trato de hacer muy entendible siempre.

EDD: Hace un tiempo leí un artículo interesante, sobre la pregunta que le hicieron en un foro a un reconocido fotógrafo. Le preguntaron que cámara recomendaba para empezar en el mundo de la fotografía , él recomendó un equipo de cámara y objetivos valorados en unos 2000\$, cuando varios usuarios le pidieron explicaciones por esa exageración , él contesto con un completo recorrido de cámaras y objetivos que te comprarías conforme ibas avanzando en el mundo de la fotografía y mejorando tu habilidad.

¿Crées que para alguien que quiere empezar en el mundo de la fotografía, con una cámara barata es suficiente o con una buena cámara y objetivo es mejor?

¿Qué cámara recomiendas para empezar?

SI: ¿Crées que para alguien que quiere empezar en el mundo de la fotografía, con una cámara barata es suficiente o con una buena cámara y objetivo es mejor?

Es una buena pregunta, de hecho, todo el que empieza a tomar la fotografía un poco mas serio de lo normal se hace esa pregunta, en mi humilde opinión para mí es más importante el saber los fundamentos de la fotografía, el porqué se debe cuidar la luz, ya que es eso, el arte de dibujar con la luz (foto, luz - grafía, dibujar). Si tuviera que empezar de nuevo, lo haría como lo hice una cámara barata, y estudiar composición fotográfica y es donde uno empieza a disfrutar de la fotografía y a dejar fluir la creatividad. Hay muchos fotógrafos que con la cámara del móvil logran unas fotos alucinantes. En el transcurso uno va descubriendo que tipo de fotografía le gusta: callejera, retratos, eventos, macrofotografía, astrofotografía, paisajes, documental, publicitaria, y muchas otras categorías.

¿Qué cámara recomiendas para empezar?

Con la que tengas a la mano, si tienes un móvil basta con eso, puedes empezar con la fotografía de manera seria. Si tienes un poco más de presupuesto y puedes comprarte

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 cámara con lentes intercambiables sería una buena inversión así sea la más barata. La ventaja de intercambiar los lentes es que te va a servir para probar con diversos objetivos (lentes), si tuviera q ué recomendar objetivos, mi recomendación sería empezar por un 50mm, luego un gran angular que puede ser de 10mm, y luego un teleobjetivo, con esos tres objetivos tienes la suficiente versatilidad para poder probar diversos estilos de fotografía mientras vas aprendiendo. Si hablamos de marcas, y no sabes mucho, busca una de estas cinco: Canon, Nikon, Sony, Fujifilm o Panasonic, que en cámaras de aficionado tienes las mejores entre relación costo / beneficio.

EDD: Dentro del mundo de la fotografía , tenemos los dos componentes principales , que son la cámara(cuerpo) y los objetivos, también existen un variedad de accesorios que dependiendo del uso serán muy útiles.

¿Qué accesorios consideras que son mas útiles para un fotógrafo?

SI: Para mi algo fundamental es un trípode, te permite probar con fotografía de larga exposición, con fotografía de la naturaleza, e incluso con fotografía callejera con un teleobjetivo para no espantar a los sujetos, baterías adicionales, si tomas muchas fotos, la batería te va a faltar, muchas memorias, por la misma razón, y buenas correas, y enganches versátiles del tipo peak design, son muy buenas para transportar la cámara con seguridad y proteger el equipo, me olvidaba, filtros y tapasoles son muy necesarios para proteger los lentes y también para probar otros estilos de fotografía

EDD: Dentro del desarrollo no todo es escribir código , hay muchas herramientas que te ayudan en el proceso de desarrollo .

A continuación pondré una lista de herramientas, tareas o conceptos que pueden ser utilizados en el desarrollo de una aplicación. Responder ,de forma breve, con el grado de importancia y conocimiento que debe tener para un programador . Por ejemplo, si es muy importante conocerlas, secundario o no es necesario .

Entornos virtuales de programación(virtualenv,conda,venv..)

Git y similares.

Programación Orientada a Objetos.

Linting.

Base de datos.

Jupyter

Depurador.

Programación asíncrona

Metodologías ágiles.

Testing.

Servicios como AWS, Azure y similares.

SI:**Entornos virtuales de programación(virtualenv,conda,venv..)**

Muy importante, es una de las mejores formas de mantener tus dependencias ordenadas y sin generar retrasos en tus proyectos por una librería o una configuración compartida.

Git y similares.

Super importante, versionar el código para mi debe ser obligatorio en cualquier proyecto.

Programación Orientada a Objetos.

Es importante conocer la POO, es un paradigma bastante usado y popular.

Linting.

Es muy importante, si se esta buscando incrementar la calidad de código.

Base de datos.

Esto puede ser secundario, si bien es cierto sería bueno conocer al menos su fundamentos básicos, no es imprescindible.

Jupyter

Es secundario, es una herramienta que te ayuda a explicar código, algoritmos, o a visualizar resultados, de las muchas que existen.

Depurador.

Super importante, es la manera más eficiente de saber cómo esta ejecutándose un programa.

Programación asincróna.

Esto puede ser secundario, ya que depende mucho del proyecto que estes construyendo

Metodologías ágiles.

Importante conocer al menos, los principios ágiles. Y si me preguntarían mi favorita sería Extreme Programming

Testing.

Importante, y al automatizarlo o buscar testing de performance sería lo mejor.

Servicios como AWS, Azure y similares.

Super importante aprenderlos, la mayoría de proyectos terminará viviendo en algún servicio de ese tipo.

EDD: En ocasiones, he leído y escuchado bastante frases o mitos sobre Python ,en relación a varias de sus características . A continuación te pondré varias de esas frases, como eres desarrollador de Python , me gustaría saber tu opinión y que grado

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 veracidad le darías a cada una de ellas.

Python es terriblemente lento.

Es lenguaje muy versátil , hace muchas cosas bien , pero ninguna de ella es la mejor opción.

Python es un lenguaje de tipado débil y eso tiene muchas desventajas.

Aunque es un lenguaje muy versátil , donde destaca en es el ámbito del data science.

Django, framework web de Python, es muy complejo de usar y no es recomendable.

Si:

Python es terriblemente lento.

Si, Python es lento, es un lenguaje interpretado, comparado con un compilado definitivamente es lento a ese nivel de comparación, sin embargo hace varios años esto ha dejado de ser una preocupación, no que deje de importar pero hay otras cosas que considerar, antiguamente el hardware era costoso, y agregar mas hardware para incrementar la velocidad pues era doloroso, pero ahora es realmente mucho más barato, y lo que ha pasado a tomar importancia es el tiempo del desarrollador, es decir lo mas importante ahora es el tiempo en que se demoren en tener el producto listo, es MUY importante, por lo que Python es una muy buena opción si se quiere tener las cosas resueltas rápidamente, y es por eso que empresas como Google lo consideran entre sus tecnologías importantes.

Es lenguaje muy versátil , hace muchas cosas bien , pero ninguna de ella es la mejor opción.

Esto es falso y se responde con evidencia de los proyectos en donde usan Python, ya que se puede usar en casi toda la industria del software, basta con solo mirar a organizaciones como Facebook, Spotify, Google, Netflix, NASA, Reddit, Uber, Dropbox, Lyft, entre otras, usándolo para diversos propósitos. No creo que estas organizaciones no hayan elegido la mejor opción para sus proyectos, no digo que sea la única tecnología que usan, pero es una de las principales dentro de sus proyectos y tareas.

Python es un lenguaje de tipado débil y eso tiene muchas desventajas.

Admito haberme confundido también con esta clasificación, entre tipado débil y fuerte versus tipado dinámico o estático. Así que siendo muy objetivo creo que mejor es usar tipado dinámico y estático, en este caso Python es dinámico porque te permite cambiar una variable en su declaración de cadena a entero, pero a la vez sería un tipado fuerte, porque no puedes mezclar variables de dos tipos distintos sin convertir alguna al mismo tipo de la otra. En la pregunta anterior sobre qué eliminaría de Python cometí ese error debí decir tipado dinámico. Y bueno su desventaja es claramente si existen malas prácticas en un proyecto el que se pueda declarar nuevamente una variable con otro tipo definitivamente nos lleva a cometer errores.

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/)

Aunque es un lenguaje muy versátil , donde destaca en es el ámbito del data science.

Esto es cierto, últimamente se hace muy fuerte en data science y en los campos de la inteligencia artificial, esto es gracias a las librerías y herramientas que están siendo usadas por la industria, sumado a que es fácil de aprender y extender, se va consolidando, sin embargo esto puede cambiar en los próximos años, así que no diría que es una verdad escrita en roca, como todo en la tecnología.

Django, framework web de Python, es muy complejo de usar y no es recomendable.

Esto lo he escuchado frecuentemente en los principiantes y es que la falta de experiencia los hace querer saltar de haber aprendido lo básico del lenguaje, a saltar al framework, y esto claramente es un error, porque al hacer ese salto sin conocer el lenguaje, no llegan a entender como funciona, y sin mencionar que Django tiene muchas cosas preconstruídas, sin conocer el lenguaje es como entrar a una jungla desconocida sin armas. Sin embargo si se tienen claros los fundamentos del lenguaje y se conocen los paradigmas de desarrollo, el usar Django es una de las mejores opciones , es muy fácil de aprender y es muy potente, y aquí nuevamente me apoyo en la evidencia de los grandes que lo usan: Instagram, Spotify, Dropbox, The Washington Post, NASA, Pinterest, Mozilla, Reddit, entre otras.

EDD: Has comentado que has tenido experiencia como profesor, con la aparición de Internet han surgido múltiples plataformas para la enseñanza de contenido , algunas plataformas mas específicas como Udemy u otras plataformas mas genéricas como Youtube. Este tipo de plataformas basan ,principalmente, la enseñanza en la visualización de vídeo.

¿Qué opinas de este tipo de plataformas para aprender?**¿Qué opinas de una enseñanza basada en vídeos?****SI: ¿Qué opinas de este tipo de plataformas para aprender?**

Son plataformas ideales para el aprendizaje, y si a estas las completamos con libros y documentación de proyectos, tenemos la formula perfecta para aprender prácticamente cualquier cosa en tecnología, en los tiempos que vivimos de trabajo remoto, distanciamiento social, y muchos países en cuarentena, a mi parecer estas plataformas no han hecho mas que mostrar su enorme potencial como medios de aprendizaje efectivo.

¿Que opinas de una enseñanza basada en vídeos?

Para mi son a una excelente manera de aprender, mucho de lo que aprendemos en el día a día lo aprendemos viendo, en todo aspecto, no sólo desde tecnología, incluso los programas de aprendizaje mas famosos como los nanodegree de Udacity, basa el gran porcentaje de sus contenidos en videos, así que definitivamente es el presente y el futuro de la educación.

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/)

EDD: Si alguien te pidiera consejo para aprender a programar con la intención de obtener, en el futuro, una carrera laboral como desarrollador. Con la diversas opciones que hay disponibles; estudios oficiales, libros, videos, master, bootcamp...etc.

¿Qué camino le recomendarías escoger?

SI: Muchas veces se considera que aprender programación es difícil de aprender, y esto se debe no a la falta de capacidad de las personas, sino a que se tienen que aprender muchos temas no solamente una sintaxis de un lenguaje de programación en particular.

Para mi lo primero es conocerse uno mismo, tratando de identificar cuál es la mejor manera en la que uno aprende, muchas personas aprenden con libros a otras les gusta mas aprender viendo, a otras con un mentor al lado y algunas con un proyecto personal en construcción, así que en base a como uno mismo aprende, se puede elegir la estrategia adecuada, no existe una formula mágica que funcione para todos.

Sin embargo independientemente de la estrategia que uno elija para aprender, siempre existirá cierta estructura que debemos de cubrir, primero independientemente de la tecnología que uno elija para aprender, uno debe enfocarse en el pensamiento computacional, es decir, entender como resolver problemas en la manera que una computadora entiende, esto se logra a través del entendimiento de algoritmos y patrones, para poder lograr este entendimiento los fundamentos de la ciencia de la computación son ideales.

Luego de entender los fundamentos de la ciencia de la computación, lo importante es conseguir libros que ayuden al desarrollo de una carrera, como el que mencionaba al principio de esta entrevista: The Pragmatic Programmer, o libros como Code Complete o Clean Code, que son libros mundialmente famosos y que te ayudan a desarrollar el pensamiento fuera de la caja como muchos lo llaman, luego de ello ya vendrán libros más técnicos.

Como tercer aspecto, busca divertirte con retos y juegos, hay plataformas como Codingame, Minecraft Education, Scratch, code.org, exercism.io entre otras donde encontraras retos interesantes que te ayudarán a mejorar tu lógica y agilizar tu manera de resolver problemas.

Suscríbete a varios canales en Youtube que tengan que ver con programación, estos canales publican información interesante relevante a la carrera, nuevas herramientas, tutoriales, guías entre otras cosas.

Y por último participa en comunidades de programadores, ya sea como colaborador de algún proyecto open source, o como un miembro de un foro de programadores, o incluso en grupos de facebook, es importante generar comunidad, muchas veces se

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/)

consiguen excelente oportunidades laborales a través de ellas.

EDD: El Sábado es el entrevistado quien realiza la última pregunta de la entrevista, pero he decidido realizar un pequeño cambio y añadido una nueva pregunta. El entrevistado realizará dos preguntas.

La primera es la misma pregunta de todas las entrevistas.

¿Qué te hubiera gustado que te preguntase? Evidentemente , debes responder a tu propia pregunta.

La segunda pregunta es el nuevo añadido, pensando en el perfil del entrevistado como desarrollador de Python.

¿Qué le hubieras preguntado a un desarrollador?

SI: ¿Qué te hubiera gustado que te preguntase?

Me parece que una buena pregunta es; ¿Qué programadores famosos admiro?, Y el detalle es que esta respuesta sería muy extensa :) , así que mencionaré los nombres y tratare de poner el porque en una sola frase: Dennis Ritchie (Creo el lenguaje C), Linus Torvalds (creador del kernel de linux y de git) y Guido Van Rossum (creador de Python)

¿Qué le hubieras preguntado a un desarrollador?

Siempre trato de averiguar que perspectivas tienen respondiendo a la pregunta: ¿Cómo ves que tu carrera esta cambiando y cambiará con el dominio cada vez mas sólido de la inteligencia artificial?, me da un panorama de como aprecian la tecnología y como esta seguirá cambiando la humanidad.

EDD: Muchas gracias por la invitación a participar en la entrevista, ha sido una experiencia entretenida y nueva, espero que sigas creciendo en audiencia, cada vez más.

Me pueden contactar por twitter como @neosergio, suscribirse a mi canal de youtube <https://www.youtube.com/neosergio> o escuchar mi podcast personal <https://anchor.fm/neosergio>, dejarme sus comentarios y sugerencias si así lo desean.

Yo propongo a una amiga mia UXer llamada Fiorella Vercelli, es muy entretenida y será una entrevista divertida.

Nuevamente muchas gracias por la invitación y estaré atento a mas entrevistados.