

Las 10 Peores Pesadillas Para Los Desarrolladores – Parte 1

Muchos piensan que el **trabajo como desarrollador ó programador es fácil**, que es solo estar en un PC casi todo el día, meneando los dedos y ellos solitos harán el trabajo. Lo que pueden llegar a ver no es ni el principio de la odisea que puede pasar un desarrollador, y para no desmeritar, hablo de todo tipo de desarrollador de software o programador. Puedes vernos tranquilos, ó maldiciendo de un momento a otro, pueden ver una tasa de té ó un 1 litro de café ¿Exagero? Lo que no ven, es lo que pasa en la maquina, todas esas **pesadillas para los desarrolladores** surgiendo en el momento menos esperado.



Para aquellos de ustedes que están pensando en entrar de lleno ~~ó ya lo estarán~~ en el mundo de desarrollo web ó de aplicaciones, estas son algunas de las situaciones que se encontrara eventualmente. Esté preparado para hacer frente a ellos, y no digas que nunca fuiste advertido. Estos son las 10 peores pesadillas para los desarrolladores ó programadores.

1. «Mejorar» El Trabajo De Alguien Mas

Imagina el siguiente panorama: Acabas de unirte a una empresa, lo más probable es que tengas que limpiar y solucionar una cantidad de errores en el proyecto de alguien mas. Te encuentras que el código es largo, realmente complejo, ilegible, lleno de bugs ... y aun así, y de una forma casi mística, aún funciona! Por supuesto, podrías tener



un 5,0003% de suerte al solucionar los errores, pero francamente la solución de estos problemas es casi imposible, por cierto, el porcentaje (5,0003%) fue sacado por un análisis intensivo al momento de la redacción ;)

El verdadero problema surge cuando no se tiene una documentación. Sin la documentación adecuada, como el nuevo desarrollador tienes que escanear línea por línea el funcionamiento de código, y esto de alguna manera, es conocer como piensa el desarrollador ó programador que has reemplazado.

2. Error Crítico A La Vista !

Después de un buen tiempo de trabajo y toneladas de cafeína, finalmente publicas tu aplicación. Sin duda alguna estarás contento, emocionado y habrás visto la luz al final de ese largo túnel, que has recorrido durante varias noches.

Luego, después de una calma interior oyes alguien «tocando a tu puerta». **Un error crítico ocurre durante la demostración**, ó provoca miles de molestias, avisandote a través de los usuarios. Debes saber que esto puede pasarle a cualquiera. Para aquellos que han estado allí, ya conocen lo frustrante que puede ser esta situación.

Para esta situación, mantén la calma. Corrige los errores lo antes posible y mientras, pon un [temporizador para el lanzamiento de tu proyecto](#). Con esto mantendrás informado a tus usuarios o clientes si tu aplicación o sitio web esta de forma offline.

3. Corrígese 1 Error...

Aparecen 100

Sin duda alguna, la **corrección de un error es un mal necesario**. Después de horas de tocar el teclado, finalmente arreglas el fallo original, sólo para encontrar que esa «solución» ha traído muchos más errores. Puede ser que has actualizado una biblioteca que no es compatible con otra biblioteca. Mientras solucionas los nuevos errores, parece que escucharas el **tic toc** en tu cabeza, llamadas entran para realizar pruebas...

Esto lo puede evitar de una gran manera. Utiliza Git para controlar sus revisiones, ya que le permite volver a las revisiones anteriores si la nueva modificación no funciona correctamente. Además, recuerda documentar cuidadosamente cada revisión.

4. No Confíes

En Esa Librería !

Cuando el **error que has encontrado en el código no existe en realidad en el código**, y buscas y buscas sin éxito. A menudo nos basamos en diferentes librerías para desarrollar los sitios web, pero de seguro, no desconfías en las librerías, ya que estas las usas diariamente en tus proyectos.

¿Qué puedes hacer? Es un dilema, ¿no es así? Vamos a considerar las opciones.

- Es posible que desee solucionar el error de la biblioteca por su cuenta, pero, ¿Conoces la forma en la que se desarrollo la librería?
- No se puede Solucionar? Entonces, debe presentar una solicitud al desarrollador para que él lo solucione. Eso va a tomar algún tiempo, y recuerda, ellos no saldrán corriendo a solucionar el problema que te atormenta, el de la entrega eres tu, no ellos.
- ¿Qué pasa si cambiamos esa librería por otra? Eso ayudaría a solventar el fallo del sistema. Pero entonces tendrá que cambiar bloques de códigos. ¿Estas dispuesto a esto?

Sólo reza a los dioses de programación para que nunca tenga que estar en esta situación.

5. Un Bug Desconocido Es La Causa

No, esto no puede ser! Has estado buscando durante días el error, has creado varias ramas en Git para las pruebas, pero el error sigue existiendo. Recurras a StackOverflow, sólo para encontrar una pregunta con el mismo problema con el envío hace 2 años y cero respuestas.

Tal vez no creas, pero la solución puede estar en dejar en paz el computador unas cuantas horas. Algo increíble, no? Claro, la solución no es que sea algo de magia, solo que muchas veces, los errores que no podemos solucionar provienen de fallos en programas, librerías que usamos, para esto muchas veces es necesario revisar los cambios, el blog oficial o alguna fuente de información de las herramientas que usemos.

Bueno, estos son solo algunas de las pesadillas que todo desarrollador ó programador podría enfrentar. En nuestro proximo articulo compartiremos las otras 5 pesadillas para los desarrolladores y programadores. ¿Con cual te identificas?