

20/9/2020

LSN133. Lo que he aprendido de la liberación de código de Radar Covid

Hace unos días la palabra "github" fue *trending* en twitter España.

Esta vez no era una explosión informativa sobre una compra millonaria o un **ataque combinado** de hackers.

Github estaba en la cresta de la ola porque era el contenedor de la sensación del momento:

La liberación del código del ecosistema de software RadarCovid (aplicaciones móviles y de gestión).

Ríos de tinta y lenguas de píxeles se han vertido sobre este tema desde entonces.

Lo que no sabe nadie es lo que pasó unos días antes.

(Ojo, tampoco lo sé yo.)

Una conversación telefónica entre dos personas poderosas. Una en un edificio ultramoderno de hormigón y cristal. La otra en un despacho con tapices flamencos y olor a naftalina.

—¡Señor Subsecretario! Precisamente iba a llamarle ahora...

—Ya, ¿qué hay de lo mío? —preguntó interrumpiéndole.

—Estábamos justo terminando los primeros despliegues en beta. Los resultados son...

—En Alemania ya llevan semanas con esto preparado. Están volando. Nosotros dices que todavía estamos desplegando las alas.

—Es que los alemanes tienen más financiación, ya sabe, el rodillo...

Hubo un silencio de unos segundos. Parecieron horas.

—¿Me vas a empezar a hablar de dinero? ¿En serio? ¿Tú sabes lo que nos jugamos en esto?

—Claro que sí señor Subsecretario, pero es que...

—Es que llaman de arriba, ¿sabe?. Día y noche. De tan arriba que solo ven nubes. ¡¡¿Cuándo demonios vais a publicar el código?!!

Minutos después alguien creó una copia de la carpeta de trabajo y la subió a otro sitio.

Un *"git push origin master"* puso fin a otro día de presión máxima.

✓ Antes de seguir te cuento que si te gusta este envío, puedes [compartirlo en twitter](#).

El código se ha convertido en noticia

Avisos a navegantes

No programo ni en Kotlin, ni Swift, ni Java, que son los tres lenguajes principales que hacen funcionar el ecosistema de [RadarCovid](#).

No vamos a analizar la funcionalidad específica de la app. Otros lo han hecho y muy bien.

(Ver enlaces del final.)

No conocemos la realidad del entorno vital del equipo que ha estado detrás de este desarrollo. La conversación del principio es una *chanza*, pero estoy convencido que muchas muy parecidas se han producido antes y después de publicar la app y liberar el código.

Esto no es nuevo, es raro, pero cada vez será más codiciado.

El código, y lo que lo acompaña, es información.

Es oro.

Aquí no vamos a hacer un análisis del código en profundidad.

No va de eso.

Se trata de traer lo que han hecho en la liberación del código al mundo a real.

A lo que podemos hacer tú y yo en nuestros proyectos para seguir buenas prácticas.

Repositorio sin memoria

El hecho más destacado a primera vista es que el código no tiene histórico de registros anterior a la publicación del mismo hace 11 días.

En el repositorio de la app para Android hay un *commit* con el mensaje: "[Initial release](#)"

Cientos de ficheros e inserciones de golpe.

Así no podemos conocer lo que se hizo antes de publicar el software en abierto.

Parece como si hubieran **copiado y pegado** una carpeta con todo, eliminaran lo que no gustara y añadir los archivos con un "git add .".

Esto podría ser fruto de la improvisación, de no contar con git como sistema de control de versiones o de las prisas por la llamada de algún subsecretario.

El motivo quizás fuera **más calculado**:

O bien eliminar el rastro de los nombres de las personas que han participado en la ejecución.

O bien no dejar pistas sobre cómo ha sido el proceso de desarrollo, para no ver meteduras de pata.

Si te pasa lo primero hay soluciones que podrían funcionar como [GitMask](#), [git-anonymize](#) o [este artículo](#) sobre *code reviews* sin sesgo.

El valor de los errores

Respecto a lo segundo del párrafo anterior.

¿Quién no ha cometido errores?

Cuando el desarrollo del software está en su época más febril, con cambios locos en *master*, sin ramas auxiliares, tests que fallan, código que no compila y líneas comentadas.

En mis *repos* habrá burradas de esas.

Claro, no hay miles de ojos examinando cada cambio, pero, ¿para qué tomarse la molestia de ocultarlo?

¿No es precisamente ese proceso donde se demuestra de lo que somos capaces de llegar a hacer?

Deberíamos tener una **cajita de los horrores** con nuestros *commits* más desgraciados y hacernos camisetas con las aberraciones cometidas.

Así quedaría para lo posteridad y nunca volveríamos a cometer el mismo fallo.

Claves privadas en abierto

Me centro en la app de Android, porque es la que más "estrellas" tiene.

Es curioso ver como el equipo de desarrollo tiene [4 de los 8 estándares de comunidad](#) ejecutados. Licencia, forma de contribuir, *readme* y descripción.

En mis *repos* seguro que flojeo en esto más que ellos.

Un vistazo a las *issues* nos muestra como uno de los mayores fallos si se ha cometido: [subir claves privadas a repositorios públicos](#).

En los últimos comentarios de ese hilo se dice que "no pasa nada", porque son los restos de las pruebas de la app en la isla de La Gomera.

En los *realities* cuando expulsan a un participante siempre le dicen que "no pasa nada". Pero si pasa: se va del juego.

Uno de los mayores agujeros de seguridad cuando se comparte el código es que se *escurren* claves SSH, "api keys", tokens...

Ese copia y pega que era tan rápido en un principio, luego se olvida y el fichero "*config*" de turno acaba viajando por la red sin querer.

Nuestras claves deben estar en ficheros ignorados por git, para empezar.

Si se coló algo importante, podemos eliminar datos confidenciales de un repositorio con algo más de esfuerzo.

Más allá del inglés

Una *issue* ya cerrada preguntaba si se podían abrir en castellano.

La respuesta fue afirmativa.

Github es un ecosistema en inglés.

Por tanto lo habitual es trabajar en este idioma, además de ser el más universal en el mundo del software, nos viene bien practicarlo.

No obstante hay algo que más allá: **la etiqueta**.

Esta igual de feo ir contra el código de conducta en la lengua de Shakespeare o la de Cervantes.

Publicar una *issue* en ruso en un proyecto en el que nadie tiene que ver con Rusia es hacer trabajar al que lo mantiene.

Recordemos que es un código público y sus autores están disponibles para ti.

¿Por qué no hacer tú el esfuerzo de hacerte entender?

Una buena práctica es escribir nuestro software en inglés.

También es bueno hacer el esfuerzo de convivir con este idioma. Pero no podemos obstaculizar al que, si el contexto es el adecuado, quiere participar en otra lengua.

Las cosas buenas

Estoy de acuerdo con tweets como los de [@lferna](#) o [@meri_minimeri](#).

Hay que agradecer el esfuerzo al equipo que ha desarrollado este ecosistema porque su propósito final es salvar vidas.

No es lo habitual, en mi caso, crear código que sirva para ese propósito.

También a todos los involucrados en el trabajo de [@carmelatroncoso](#) para diseñar el protocolo en el que se basa la capacidad de detección de contactos infeccioso de RadarCovid: [DP3T](#) (Decentralized Privacy-Preserving Proximity Tracing).

Lo que han generado se integra con 17 sistemas de salud, uno por cada Comunidad Autónoma.

Estoy convencido que, aunque el software se base en trabajo ya hecho por otros, gran parte de los 300.000 euros de la licitación se han ido en esa parte, en la integración.

Algunas de las *pull request* ya cerradas son participaciones no precisamente pequeñas para añadir traducciones al [gallego](#) o el [euskera](#).

Y que los árboles no nos impidan ver el bosque: **han liberado el código**.

Es algo que no pasa todos los días en la administración.

Enlaces sobre el tema

- [RadarCovid en Github](#)
- [Análisis de la app en Palabra de hacker](#)
- [Informe de análisis funcional y de código de @linuxct](#)
- [Hilo de @jjmerelo](#)
- [Así es Radar COVID por dentro: análisis de las aplicación de notificación de exposición](#)
- [Qué ha pasado con Radar Covid y Firebase: para qué sirve esta herramienta y qué significa su presencia en el código](#)
- [El Control de Versiones con Git](#)
- [Curso Empezar con Git en la Zona Premium](#)

¡Nos leemos el próximo domingo!

PD: Ahora, a descargarse la app.

Publicado esta semana:

- [WRP 98. Cómo ampliar tu red de contactos online](#)

- [WR 157: React y testing en frontend con Dani de la Cruz](#)
- [El fuego se aviva en el ring de Jamstack vs WordPress #6](#)

dani.g

Tu email

¡Quiero suscribirme!

He leído y acepto la política de privacidad.
