

 Grabación Presentación

Con un pie afuera

"If you want to be a good Archeologist, you gotta get out of the library!" —Indiana Jones.

Indiana Jones and the Kingdom of the Crystal Skull.

Como developer llevarás a cabo una serie de actividades, algunas más técnicas que otras. Durante tu carrera has aprendido y desarrollado las habilidades necesarias para hacerlo: desde construir un botón, hasta diseñar e implementar arquitecturas de componentes, manejando estados y eventos. Y en este último Sprint has aprendido a trabajar del lado del Backend, utilizando Javascript como lenguaje para generar tus APIs, documentarlas, probarlas, para conectarlas a una base de datos y escribir datos en el disco rígido.

Además has puesto en práctica las competencias necesarias para la comunicación, la creatividad y la organización propias de un espacio de trabajo profesional. Estas aptitudes te servirán a la hora de participar en un equipo de desarrollo, ya sea dentro de una empresa, un emprendimiento, e ¡incluso en un hackaton con tus pares!

¿Quién abre la caja?

Estás a pocos encuentros de finalizar la carrera y queremos prepararte lo mejor posible para tu próxima aventura en la vida real. Por eso, en el próximo encuentro realizaremos un workshop en el que podrás poner a prueba todos los conocimientos adquiridos hasta ahora. ¡Aprovéchalo para resolver cualquier duda —o problema— que tengas!

En el mundo del software, los problemas están aún por descubrirse. Si nadie los ve, o los detecta, decimos que en realidad no son un problema! Es como [la paradoja de Schrödinger](#): si nadie abre la caja, no sabremos el resultado.

De todos modos, los problemas en el desarrollo de software emergen tarde o temprano: nuevos requerimientos se asoman o nuevos desafíos técnicos serán necesarios para completar las partes.

También decimos que un problema no se resuelve, sino que se enfrenta. En ocasiones nos enfocamos tanto en llegar a la solución, que nos olvidamos de realmente encarar el conflicto, entenderlo, y empezar a dilucidar las diferentes partes que lo constituyen.

Llevando esta idea a nuestra línea, nuestros proyectos son un problema que requieren un abordaje, y este abordaje empezará a arrojar problemas que necesitan entendimiento y ajustes, llevándolo a su solución. La solución de un problema nos es otra cosa que el proceso que te llevó, en definitiva, a pasar de un estado caótico al estado de balance que necesitas.

Cordones sueltos

Pensemos en un problema sencillo, que nos puede pasar a todas y a todos en cualquier momento. Alguna vez hemos querido desatar los nudos incómodos que a veces se forman en los cordones de nuestros zapatos. Podríamos pensar que nuestra solución es que estén desenlazados pero ese es el estado ideal, es decir, el balance que deseamos para que nuestro problema no esté ocurriendo.

La solución a nuestro problema, en cambio, consistirá en tomar los cordones, empezar a ver donde están los nudos más enredados, y aplicar la presión necesaria para poder liberarlos. Y deberemos repetir esto tantas veces como sea necesario, en tantos nudos existan, para finalmente llegar a nuestro estado ideal: un par de cordones sueltos.

Refinamiento del backlog

Lamentablemente, para resolver la complejidad de requerimientos a los que nos enfrentamos

los requerimientos pueden ser tan abstractos como “*necesitamos hacer un portal de registro de usuarias/os*”, lo cual dispara un infinito de preguntas como:

1. ¿Cuántos datos se requerirán para registrarse?
2. ¿Se necesita un password?
3. De necesitar un password:
 1. ¿Se debe confirmar el password en otro campo?
 2. ¿Se puede permitir copiar/pegar sobre el campo?
 3. ¿Tiene un patrón el password? ¿6 letras y un número?
4. De necesitar un patrón:
 1. ¿Qué pasa si no se cumple? ¿Se emite un error?
 2. ¿La validación se hace tecla a tecla o al presionar el botón de registro?

Y así la lista puede seguir hasta llegar a un requerimiento mínimo que la/el developer pueda realizar. Generalmente las respuestas a estos ítems pueden venir resueltas por parte de la figura del *manager del producto*; sin embargo, parte del trabajo de un/a developer también consiste en levantar la mano sobre cualquier ítem que no queda claro, especialmente si afecta el comportamiento funcional del sistema. Aquí va un consejo: ¡no tomes decisiones por tu cuenta sin antes debatirlas con tu equipo! Haz propuestas a tus respectivos managers, o miembros del equipo y asegúrate de saber lo que piensan.

El proceso de descomponer una tarea en múltiples ‘pedacitos’ accionables se le conoce como **backlog refinement**, y es una práctica muy común en los equipos de desarrollo. Consiste en tomar una pieza de requerimiento y desmembrarla tantas veces hasta tener una tarea realizable. El

La identificación de tareas a realizar en un proceso de desarrollo de software es una actividad tan importante como programar dichas tareas. El backlog es un artefacto de las metodologías ágiles para centralizar todas las tareas de un proyecto, y visibilizarlo ante quienes deberán tomar decisiones y ejecutar.

Cada tarea deberá estar refinada a un nivel de ejecución factible acorde a la experiencia del/la developer. El proceso de refinamiento propone un cuestionamiento sobre la tarea en sí, desencadenando una serie de preguntas que le permite al developer entender lo que debe y cómo lo debe hacer, creando un esquema mental de los pasos a seguir.

Un conjunto de tareas refinadas, listas para ejecutar, pueden minimizar también la aparición sorpresiva de errores, o bugs, en nuestro código. Sin embargo, alcanzar un nivel “bug free” ocurre de la mano de muchas herramientas y posturas, como testing, el seguimiento de buenas prácticas y la minimización y control de efectos secundarios en funciones. Todos estos criterios son esenciales para la práctica avanzada de un desarrollo.



¡Primer bug encontrado!

¡Prepárate para el próximo encuentro!



Profundiza

Te invitamos a conocer más sobre el tema de esta bitácora.



Challenge

