

# Instituto Tecnológico de Pabellón de Arteaga

## Desarrollo, Aplicación y Consultoría de Sistemas de Información

Tics 9no.

Lectura 10

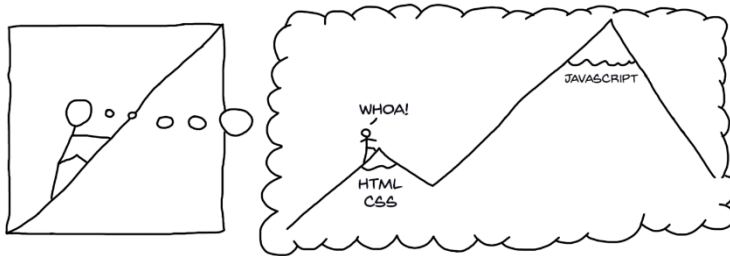
Prof. Eduardo Flores Gallegos

Denise Kirstie Martínez Santana

7 de Diciembre del 2019

## LECTURA 10

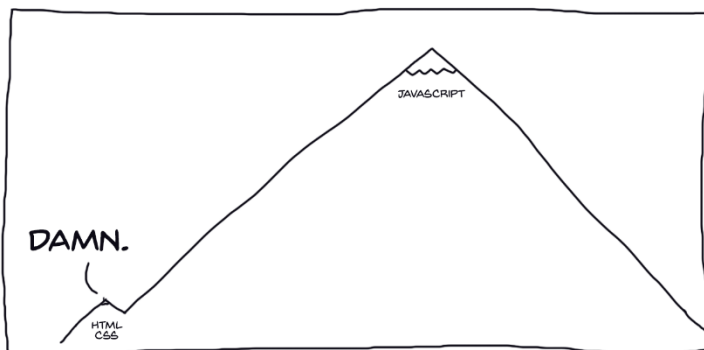
Todo comienza cuando te encuentras en uno de los puntos más emocionantes de tu viaje de desarrollo web: un momento para cambiar tu enfoque de estilos, estética y sistemas de grilla a lógica, marcos y escritura de código JavaScript.



Empieza así ...

En este mismo momento, empiezas a ver que cuando se trata de JS, es mucho más que unos simples trucos y efectos visuales de jQuery. Usted ve una gran imagen de sí mismo creando aplicaciones web, no solo meras páginas web.

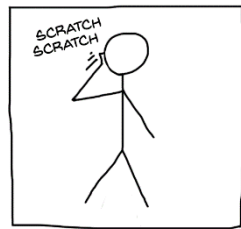
Mientras pones más y más esfuerzo en crear código JS, empiezas a pensar en la interactividad, tus subsistemas y lógica. Las cosas comienzan a funcionar y, por fin, sientes la vida en tus aplicaciones. Un mundo completamente nuevo y emocionante aparece frente a ti. Y con eso, un montón de nuevos grandes problemas.



... y terminar en esto. ¡Y no es el final!

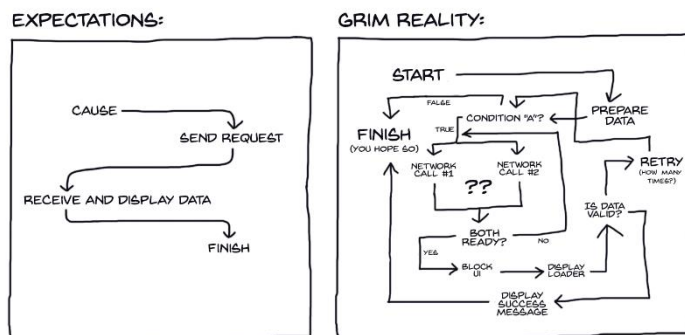
Pero no estás desanimado, las nuevas ideas surgen constantemente, escribes cada vez más código. Se están probando varias técnicas de esa publicación de blog, y se están probando diversos enfoques para resolver problemas.

Entonces, empiezas a sentir una pequeña picazón.



Su script.js archivo crece. Hace una hora tenía 200 líneas de código, ahora supera las 500 líneas. "Oye", piensas, "no es un gran problema". Has leído sobre código limpio y mantenerle, y para lograrlo, comienzas a separar tu lógica en archivos, bloques y componentes. Las cosas comienzan a verse hermosas otra vez. Todo está cuidadosamente catalogado como en una biblioteca meticulosamente organizada. Te sientes bien, porque varios archivos se nombran correctamente y se colocan en un directorio determinado. El código se vuelve modular y más mantenerle.

De la nada, empiezas a sentir otra picazón, pero la causa aún no está clara. Las aplicaciones web rara vez son lineales en su comportamiento. De hecho, se supone que muchas acciones en cualquier aplicación web ocurren repentinamente (a veces incluso de forma inesperada o espontánea). Las aplicaciones deben responder adecuadamente a los eventos de la red, las interacciones del usuario, los mecanismos de sincronización y varias acciones diferidas. De la nada, monstruos feos llamados "asíncronicidad" y "condición de raza" están llamando a sus puertas.



Debes casar tu hermosa estructura modular con una novia fea: código asíncrono. Y entonces el picor se vuelve obvio. Una pregunta difícil comienza a surgir: ¿dónde demonios debería poner este código? Usted podría tener su aplicación bellamente dividida en bloques de construcción. Los componentes de navegación y contenido se pueden colocar ordenadamente en directorios apropiados, los archivos de script de ayuda más pequeños podrían contener código repetitivo que realiza tareas mundanas. Todo puede ser administrado por un solo app.js archivo de código de entrada, donde todo comienza. Ordenado.

Pero su objetivo es invocar código asíncrono en una parte de la aplicación, procesarlo y enviarlo a otra parte.

¿Debería colocarse el código asíncrono en el componente UI? ¿O en el archivo principal? ¿Qué bloque de construcción de tu aplicación debería ser responsable de manejar la reacción? ¿Cuál para procesar? ¿Qué pasa con el manejo de errores? Usted prueba varios enfoques en su mente, pero su inquietud no desaparecerá, es consciente del hecho de que si desea expandir y escalar este código,



INSTITUTO TECNOLÓGICO  
de Pabellón de Arteaga

**ITEC**

se volverá más y más difícil. La picazón no se había ido todavía. Necesitas encontrar alguna solución ideal y versátil.

Comienza a leer sobre cómo tratar este problema y busca soluciones listas para usar. En el primer momento, lee sobre las ventajas de las promesas sobre las devoluciones de llamada. La próxima hora, te encuentras tratando de comprender qué es RxJS (y por qué un individuo al azar en Internet dice que es la única salvación legítima para la humanidad en desarrollo web). Después de leer un poco más, intentas entender, por qué un blogger escribió eso `reduxsin redux-thunksentido`, pero otro piensa lo mismo `redux-saga`.