

Pausa y repaso

"Lo que más admiro en los demás es la ironía, la capacidad de verse desde lejos y no tomarse en serio." -Jorge Luis Borges.

Cuando se aprende tanto y a esta velocidad, viene bien tomarse un momento para integrar las nociones en la práctica, validar conocimientos poniendo manos a la obra. Por eso mismo, ¡durante el próximo encuentro participarás de un workshop! Para esto te prepara esta bitácora. Pero antes de empezar, conoce una herramienta fundamental que facilitará tu trabajo como developer: las librerías.

Librerías

Tanto en los sitios webs como en aplicaciones móviles o software de escritorio existen un montón de funcionalidades que son idénticas o muy parecidas. Las más comunes –aunque claramente no las únicas– en un sitio web podrían ser un registro de usuario, login, menú hamburguesa, slider de imágenes o gráficos, entre otras.

Cada una de ellas requieren muchas horas de desarrollo, entonces la pregunta que te debes estar haciendo en este momento es: ¿tengo que programar todo esto cada vez que debo realizar alguna de estas funcionalidades? La respuesta es no. Y para esto las librerías son la respuesta al ahorro de muchas horas de desarrollo.

Una **librería** es una colección de funciones con el fin de resolver un propósito específico.

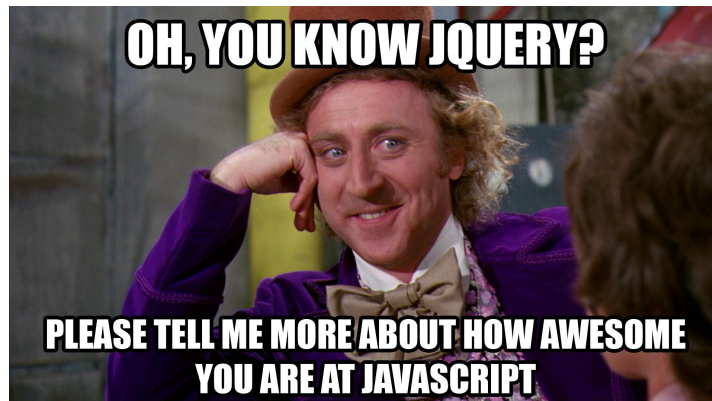
Existen miles de librerías en Javascript que puedes incluir en tus desarrollos, solo debes seguir las instrucciones del developer que la ha programado para adoptarla en tu proyecto.

[jQuery](#) es la librería que desarrolló John Resig cuando solo tenía 20 años. Publicó su primera versión a principios de 2006 y rápidamente se convirtió en un éxito que aún hoy continúa.

Su comunidad creció tanto que developers del mundo (si, un/a como nosotros/as) hicieron cientos de plugins para extender su funcionalidad y



hacerla más popular. Hoy en día existen proyectos contruidos alrededor de [jQuery](#) como [jQuery UI](#), [jQuery Mobile](#), [jQuery Unit](#) o [Sizzle](#).



El mercado de las librerías siguió evolucionando y nacieron nuevas librerías cómo [React JS](#) (Facebook) o [Angular JS](#) (Google) que tomaron los errores que venían arrastrando otras librerías, aplicaron nuevos conceptos y aprovecharon el crecimiento de la tecnología hacer sus librerías más robustas y potentes.

En conclusión, puedes tomar un problema común que debas resolver, buscar una librería que lo haga y analizar si puedes adaptar la librería a las funcionalidades que buscas, o simplemente tomar algunos conceptos de la librería para aplicar en tu desarrollo.

Ahora sí... ¡workshop time!

Nuestro próximo encuentro será un workshop integrador para poner en práctica todo lo visto en el Bloque 2 (¡que es un montón!).

Hemos visto distintas formas de capturar cada comportamiento que sucede en un sitio web, definir acciones y comunicarse con servicios externos:

- Variables, Arrays y Objetos
- Acceder al DOM y modificarlo
- Capturar eventos y aplicar funciones.
- Asincronismo
- Promesas (then, catch)
- Funciones asíncronas
- Estructura de paquetes HTTP
- Request methods



Pero antes de empezar, hay una pregunta de suma importancia que debemos hacernos, ¿qué podemos mejorar de la última experiencia de workshop que tuvimos?

Autocrítica

No hace mucho estuvimos resolviendo un workshop, para ser específicos nuestro segundo workshop. En esta nueva instancia queremos que identifiques qué aspectos de tu desarrollo puedes pulir o mejorar para atacar de nuevo en el próximo. Algunos de los criterios sobre los que puedes pensar son:

- **Calidad del código:** ¿me preocupé más de que mi código fuera legible y mantenible, o sólo programé para resolver objetivos específicos?
- **Diseño de algoritmos:** ¿me tomé el tiempo de plantear un flujo para las diferentes funcionalidades de mi aplicación?
- **Identificación de sub problemas:** ¿dividí las diferentes tareas en partes más pequeñas y realizables?
- **Administración del tiempo:** ¿consideré el tiempo otorgado para el alcance de mis objetivos?
- **Planificación:** ¿listé las tareas a realizar?

Estas son solo algunas preguntas disparadoras que te ayudarán a entender más sobre tu propia forma de trabajar, y aquellos aspectos en los que quieres seguir desarrollándote como developer. El objetivo de un workshop no sólo consiste en tener una aplicación web interactiva, sino también en plantearse una serie de objetivos enmarcados en un espacio temporal para alcanzarlos. Este último es un aspecto fundamental para cualquier desarrollador de software.




¡Prepárate para el próximo encuentro!

Profundiza


 En [este video](#) Wes Bos y Scott Tolinski te cuentan los 11 hábitos de un/a developer eficaz.

 [Josh Kaufman reflexiona sobre las primeras 20 horas de aprender algo.](#)

Comunidad

 [Programar es una mierda](#) es un podcast sobre desarrollo de software. Programar es muy divertido, pero a veces es un infierno. Mejor tomárselo con humor.

Challenge

 Crea tu primer HTML incluyendo JQuery desde la CDN y muestra el famoso "Hola Mundo".