

El día de ayer estuve trabajando en el curso de Frontend Masters: Design for Developers. Ha estado bastante interesante, ya que he visto cosas que no conocía y he repasado temas que tengo muy bien aprendidos. Entre ellos, están los layouts, como crearlos, trabajar con simetría para que el diseño del sitio web se vea atractivo, las diferentes técnicas que se pueden utilizar al aplicar simetría y hacer volar la imaginación con tantas cosas que se pueden implementar. La persona encargada del curso estaba utilizando Photoshop, y estuve creando unos layouts de como podrían ser creados para los desarrolladores y nos dió bastantes ejemplos. Luego estuvo ya implementando CSS para la creación de layouts. Lo bueno de esta persona del curso es que tenía DEMASIADA información y sources para compartir y me eso es lo que me gusta porque a la misma vez que explica, yo también voy descubriendo y aprendiendo para que se utiliza esto o para que se utiliza aquello y eso es super valioso. Después de pasar la parte de CSS Layout, entramos a la que yo pienso que ha sido mi favorita del curso, ya que estuvo enseñándonos demasiadas cosas sobre los colores y toodaa una gran gama de como se trabajan con ellos y como crear paletas de colores. Vimos los colores aditivos y sustractivos (los cuales no tenía ni idea que existieran), las ventajas y desventajas de trabajar con colores rgba, hexadecimales y hsla (el cual la persona del curso comentó que era la mejor manera de implementar colores en sitios web por su fácil manera de comprensión), las diferentes técnicas que se utilizan para mezclarlos y obtener una buena paleta de colores, como por ejemplo: analogos, monocromáticos, triad, complementarios, compuestos y muchos más. Después de esa parte entramos a Tipografías, la importancia de utilizar tipografías en texto y en títulos, los colores que debemos implementar al trabajar con diferentes tipografías, el utilizar o no ligaduras para mejor comprensión o mejor estilo y muchas cosas más para que nuestro sitio se vea bonito y elegante a la vez. También nos enseñó a hacer lettering solo con HTML y CSS lo cual fue excelente. El día de hoy terminé el curso e iniciaré el siguiente de diseño de SVG, así que a darle con todos los powerade, potentísimos y activadísimos.

El día de ayer estuve trabajando en los cursos de diseño. Del primer curso, logré terminar las partes de Inspiration, Images and Performance, Prototyping y Wrapping Up, en la parte de Inspiration aprendí que para tomar inspiración de crear, inventar e innovar, se pueden tomar piezas de proyectos de otras personas y con ellos hacer un solo proyecto, propio y único, lo cual esta bien, no hay ningun problema y es una tecnica llamada Remixing. Luego en la parte de Images and Performance, conocí muchos sitios web y sources para optimizar los kilobytes de las imagenes o svg que muchas veces utilizamos en nuestros sitios web y no nos damos cuenta que pesan más que nuestro HTML y CSS juntos. Luego en la parte de Prototyping, estuvo más tranquila, por que fué más que todo consejos de como implementar animaciones en nuestros proyectos, el flujo de entradas y salidas de elementos para que se vea elegante y no brusco, el diseñar una pantalla en modo UI pero en papel, para despues pasarla a Photoshop y luego al Frontend Developer, y como tomar cosas simples para un sitio y poco a poco irlo perfeccionando, añadirle algun framework como ReactJS, o VueJS, o Angular y jugar con el codigo, vivir dentro del codigo y convivir con el codigo para ver como evoluciona y mejora en sí. Para finalizar el curso, estuve viendo unas transitions dentro de sitios web, para ver como se puede implementar en proyectos futuros y hacer explotar la imaginacion con el asombroso poder que CSS tiene. Luego de terminar Design for Developers, empecé el nuevo curso de SVG Animations. En este curso, en la parte introductoria conocimos la web Codepen bastante bien, para que se utiliza y como se puede ocupar dentro del ambiente del desarrollo de software. Luego, ya entrando en materia, en la parte de SVG Anatomy, recuerdo que el año pasado tome un curso de HTML en SoloLearn, y ahi fue la primera vez que cree algo en SVG. No estaba tan perdido porque tambien es como trabajar con map y encontré rapidamente la logica que se implementa al crear con este tipo de tecnología. Luego en la parte de CSS Animation, aprendí a como estilizar un SVG y como animarlo con CSS. De igual manera aprendí a como aplicar un responsive tomado de la mano con elementos SVG a la hora de implementarlos en un sitio web. Estuve viendo algo llamado storyboard, y ayuda bastante a los diseñadores a crear animaciones o elementos a utilizar, y se empieza desde crear algo en papel, un lápiz, borrador e imaginación a pasarlo a una plataforma digital para utilizar. Luego de ello estuve viendo algo llamado sprites, la cual fue la parte más interesante que ví, en la que era una animación super bonita, con muchos elementos, ambientes de la animación y todo ello en un solo SVG. El día de hoy finalizaré el curso de SVG Animations, y empezaré ya con el path de ReactJS de nuevo, tomando el curso Intermediate React. Así que a seguirle dando con todo que ya casi estamos por finalizar. Esta es mi semana 11, me quedan 4 aún, así que hay que meterle para sacar todo el jugo posible.

El día de ayer estuve trabajando en los cursos de SVG Animation y Intermediate React. En el curso de SVG Animation, aprendí muchas cosas INCREÍBLES de lo que se puede hacer al trabajar con SVG en volar la cabeza con tanta imaginación, herramientas y recursos que existen para crear todo un mundo posible solamente con CSS, JavaScript y elementos de HTML. Estuve trabajando en las partes GreenSock, en la cual empecé a ver la utilización de JavaScript para manejo y animación de SVG, la persona encargada del curso nos compartió bastantes sources para animar SVG, como D3, Timeline, GSAP, Stagger, entre otras. Luego pasamos a la parte de UI/UX Animation, en la que nos centramos bastante en algo llamado "Draggable" (la cual fue la que más me gustó). Pienso que para entrar más a esos niveles tengo que hacerlo con tiempo y con paciencia, para entenderle bien bien y poner a trabajar y manipular muy bien el DOM para hacer cosas bonitas y funcionales y no desordenadas y catastróficas jeje. Después entrando a la parte de GSAP Extras estuve viendo animaciones de curvaturas y rotación (la animación de una luciernaga hice con SVG y JavaScript), animaciones con texto utilizando SplitText, animaciones con colores y trabajos con la herramienta Greensock. Ya para finalizar, en la parte Advanced SVGs, estuve viendo algo llamado Clipping and Masking al trabajar con animaciones y que éstas se vean elegantes, llamativas y reales, no como algo robótico o sin flujo, y otras cosas llamadas viewBox y svgOrigin que de igual manera ayudan a crear animaciones, con good performance, otras features y más. Finalizando el último curso de diseño, empecé el path de ReactJS. El path contenía 3 cursos pero 1 de ellos ya lo había realizado en el path de Frontend Professionals, así que empecé en el segundo curso llamado Intermediate React, v3. Lo que más me gustó de esta vez al trabajar con ReactJS es que me sentía seguro. Seguro de lo que estaba programando, de lo que estaba modificando y me volvió esa satisfacción de decir yo puedo programar. En el curso empecé la parte de Hooks, en la que solo abarqué useState, useEffect y useContext. El día de hoy seguiré trabajando en la parte de Hooks, la siguiente parte engloba TailwindCSS (el cual ya estoy muy bien empapado de la librería) y a seguir aprendiendo que andamos con todos los power super activados.

El día de ayer, estuve trabajando en el curso de Intermediate React, v3. En la parte de Hooks, conocí muchos de ellos, los cuales ni tenía idea que existían ni para que servían. Los únicos que sí recordaba eran el `useState` y el `useEffect`. Estuve viendo el `useContext`, `useRef`, `useReducer`, `useMemo`, `useCallback`, `useLayoutEffect`, `useImperativeHandle` y el `useDebugValue`. Lo bueno de la persona que está impartiendo el curso, es que a la misma vez que nos están enseñando para que sirve cada hook, nos manda el link del hook en la documentación de ReactJS, para así después revisarla y estudiarla de mejor manera. Luego en la parte de TailwindCSS, aprendí a como fusionar tailwindCSS con ReactJS, descargando la librería, aplicándole tailwind a cada componente del proyecto y jugando con los diseños y estilos que tailwind ofrece. Dentro del proyecto creado, se le aplicó responsive y grid para el manejo del sitio y de igual manera se hizo un gradient en el header del proyecto. Tailwind lo tengo en un 70% aprendido. Se me hizo super fácil entenderle a todo lo que se le estaba aplicando a cada componente, pero algo que sí debo seguir perfeccionando es trabajar con el postCSS y el `tailwind.config.json`, porque cuando me veo en la situación de añadir un plugin o una feature personalizada, todavía me quedo perdido, entonces debo mejorar esa parte (con la práctica me voy con todo). El día de hoy estaré trabajando en la parte de Code Splitting & Server Side Rendering, TypeScript, Redux y Testing. Ayer por la noche estuve leyendo la documentación de React en cuanto a la lógica de la librería y repasando algunas cosillas que se me habían olvidado. Así que hoy espero terminar el curso para el día lunes empezar el siguiente curso de React: State Management in Pure React, v2 y así culminar el path de React. A seguir dándole.

El día de viernes, trabajando en la parte de Code Splitting, aprendí algo super interesante para un good performance al trabajar con componentes que son demasiado pesados y quiero que carguen rapido en mi proyecto. Las herramientas se llaman lazy y Suspense. Lazy es una funcion y Suspense es un componente de React. Lazy se trabaja como closure, y se le agrega una callback que contiene un import, lo que hace que el archivo pese mucho menos, crea un archivo .map y este carga super rapido a la hora de abrir el sitio. Luego en el archivo App.js, se importan lazy y Suspense y todo el cuerpo del sitio web en el return, se introduce dentro del componente Suspense y este hace que indique si el componente esta cargando si aun no ha sido renderizado. De igual manera estuvimos creando un servidor con Node.js junto con Express.js. Entendía y no entendía a la vez lo que el tipo del curso estaba haciendo, pero lo bueno es que sigo aprendiendo muchas cosas que se pueden implementar, solo con un simple proyecto. Luego al terminar esa parte del curso, pase a el gran TypeScript. En TypeScript estuve viendo por primera vez los archivos .tsx, en el cual era basicamente migrar archivos React a TypeScript. Al principio me costó entender lo que el tipo del curso estaba haciendo, pero ya después fui leyendo el codigo, le di una repasada a TypeScript y le capté de mejor manera a todo el procesos que se estaba haciendo con React. Luego en la parte de Redux, estuve viendo que antes Facebook utilizaba una tecnología llamada Flux, en la cual tenía que utilizar bastantes archivos Flux para trabajar con el estado de la aplicación, pero se utilizaban muchos archivos, y es cuando Redux llegó a reducir esa cantidad de archivos necesarios y con el mismo resultado. Estuvo muy interesante esa parte. Al principio no comprendía que era o en que consistía. Investigué un poquito más después y ya le agarre. Luego vi como se implementaba Redux en React, conocí las Redux Dev Tools rapidito y pasamos a la parte de Testing. Ya en testing, estuve viendo por primera vez, frameworks directamente creados para et testing (no conocía ni tenía idea que existía esa rama del desarrollo tampoco). Me gusto mucho la manera de pensar de la persona encargada del curso, y esa era que si uno como developer no esta 100% seguro y confiado del codigo que ha escrito, entonces que no se haga el test, porque ya se sabe que saldrá mal. En cambio, realizar un codigo legible, que se entienda porque se hace esto, porque se hace aquello, que pasa si quito esto, que pasa si modifico aquello hago el test y estoy 100% seguro que va a ser un success seguro, entonces ahi si es buena manera de hacer un test, porque tambien existe la posibilidad de que ese test nos de un error (lo cual es lo más importante), y como ya sabemos como trabaja el codigo, captamos rapidamente lo que tiene mal y rapidamente lo solucionamos. Luego estuvimos viendo ejemplos de como hacer tests de los componentes anteriormente creados (component'sName.test.js), se le aplica la metodología que explique sobre la implementación de tests al conocer todo el codigo, y asi obtener todos los componentes passed y listos para production. Luego de ello estuve viendo el proceso de los tests, algo llamado snapshot y una herramienta de Jest (framework de testing) llamada Istanbul, la cual nos da un reporte más llamativo de un testing realizado, dentro de nuestro proyecto (super

interesante y useful, ya a la hora de trabajar). El día de hoy estaré trabajando en el curso: State Management in Pure React. Este curso tarda 4 horas y media, trataré de terminarlo el día de hoy y meterle con todo para terminar unos 4 cursos esta semana para las últimas 3 de mi trainee, meterle feeling al final project, así que a darle con todo.