**Proyecto de cuestionario de autodiagnóstico para ONAC encargado por Jinneth Hernadez**

**Consideraciones logísticas para la realización del proyecto**

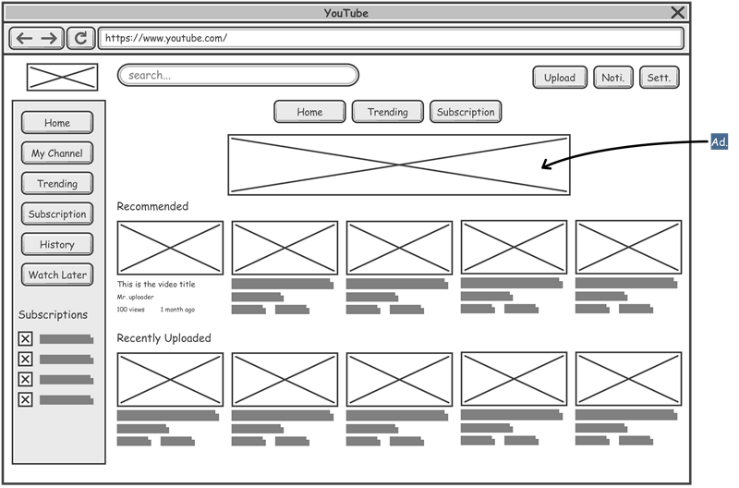
Primero y mas importante, el salario de un programados junior ronda mas de los dos millones por lo cual cobrar el trabajo en solo 200 mil fue bastante por debajo del precio del mercado, el desarrollo de un cuestionario así ronda un precio de entre 300 y 400 mil pesos.

Es importante considerar si el cliente tiene o no bien estructurada la idea del proyecto y lo que quiere hacer. hacer un mapa mental para organizar ideas sueltas del cliente conlleva mas trabajo, lo cual incrementa el precio del proyecto porque el trabajo inicia no desde la fase de desarrollo si no desde la fase de planeación.

Es importante tener en cuenta que hay trabajos que necesitan conocimientos básicos de redacción y en ocasiones los clientes carecen de estas habilidades. Esto merece varias consideraciones dependiendo el caso en aras de ser eficientes con la entrega:

* Si los textos son pequeños puede ser mejor corregirlos uno mismo en vez de detenerse en pequeñeces para no estancar el proceso de desarrollo
* Si hay mucho texto es valido rechazar el texto y pedirle al cliente que lo corrija puesto que es un insumo necesario pero el trabajo de un desarrollados no es el de un corrector de estilo
* Si el cliente requiere en que sea uno quien corrija los textos largos, esto implica que el plazo de entrega debe ampliarse y el costo del producto debe incrementar

Saber bosquejar la estructura es también importante, es decir tener conocimientos de wireframe (estructura de alambres) ayuda a concretizar y darle forma a la idea del cliente pues muchas veces el cliente solo tiene una idea y quiere un programa que la haga realidad, pero no tiene conocimientos sobre como debe ser estructurada una web

  
*estructura wireframe de una web*

En resumen, hay 3 habilidades que ayudan a la planeación logista y realización del proyecto: saber estructurar una web a través de wireframe, poder desglosar y aterrizar la idea del cliente con mapas mentales y de ser necesario una redacción correcta

**Consideraciones técnicas para la realización del proyecto**

Hubo varios comandos y varas formas de trabajo que fueron necesario estudiarlas para completar el proyecto. Así que se hará mención de todos los aspectos técnicos que fueron necesario estudiar, repasar o consultar por más básicos que sean.

Una de las primeras consideraciones fue como incluir correctamente un archivo de CSS en el HTML, lo cual implica poner las dependencias correctamente en head del documento y además escribirlas bien

|  |
| --- |
| <link rel="stylesheet"  href="(nombre del archive de CSS a incluir)" /> |
| *(sintaxis correcta de la dependencia de CSS)* |

Siguiendo con las dependencias, la correcta inclusión de Bootstrap es indispensable a la hora de diseñar front dada la increíble versatilidad que brinda. En clases pasadas se observó que las dependencias usadas por los profesores eran mucho mas largas y completas que las brindadas por la pagina misma, así que es importante consignarlas y tenerlas a la mano

|  |
| --- |
| <link          href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"          rel="stylesheet"          integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx"          crossorigin="anonymous"      />      <script          src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js"          integrity="sha384-A3rJD856KowSb7dwlZdYEkO39Gagi7vIsF0jrRAoQmDKKtQBHUuLZ9AsSv4jD4Xa"          crossorigin="anonymous"      ></script>      <script          src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.5/dist/umd/popper.min.js"          integrity="sha384-Xe+8cL9oJa6tN/veChSP7q+mnSPaj5Bcu9mPX5F5xIGE0DVittaqT5lorf0EI7Vk"          crossorigin="anonymous"      ></script>      <script          src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.min.js"          integrity="sha384-ODmDIVzN+pFdexxHEHFBQH3/9/vQ9uori45z4JjnFsRydbmQbmL5t1tQ0culUzyK"          crossorigin="anonymous"      ></script> |
| *(sintaxis correcta de la dependencia de Bootstrap)* |

A nivel técnico también fue importante como insertar lo que retorna una función, ya sea texto o una imagen, desde el archivo de JavaScript en el HTML fue también importante

Al insertar una desde un archivo JS en la etiqueta de imagen Se deja el campo de src que sería donde va la ruta de la imagen que debe mostrarse solo con un #

|  |
| --- |
| <img id="img" class="img-responsive" src="#" alt="" width="25%” > |

Posteriormente en vez del return clásico en la función de JS se referencia el campo donde se quiere insertar la imagen y se le da la ruta

|  |
| --- |
| document.getElementById("img").src = "(ruta de la imagen)"; |

Debe resaltarse que el argumento del método document.getElementById, que en este caso es img, debe ser el id de la etiqueta de <img> donde se quiere insertar la imagen para que asi el programa entienda que es lo que debe insertar y en donde

El proceso de insertar un texto es muy similar. Hay que definir una etiqueta y darle un id para poderla referenciar en la función de JS, la etiqueta más fácil para este propósito sería un <div>

|  |
| --- |
| <div class="message-Modal" id="mensajeModal"></div> |

Luego desde la función se referencia dicho id y con el método .innerHTML se le indica que debe insertar el texto definido después del igual en donde se encuentre el id indicado (que para este caso es mensajeModal)

|  |
| --- |
| document.getElementById("mensajeModal").innerHTML = (texto que se debe insertar); |

Cabe resaltar que no es necesario indicar expresamente un texto, puede ser una variable tipo Sting que contenga el texto deseado o en general cualquier tipo de variable (Inr, float, boolean, etc)

Las propiedades de CSS que permitieron dar estilo al proyecto también vale la pena recordarlas:

* width="25%"(define el tamaño de una imagen) (puede ser en porcentaje [%] o pixeles [px])
* text-align : center; (define si el testo estará hacia la izquierda, derecha, centrado o justificado )
* text-align: justify;
* color: #FF0000; (define el color del contenido de la etiqueta, en el caso de ser texto cambia el color de la letra)

A la hora de crear estilos en el archivo de CSS es importante recordar como hacer referencias a los componen en el HTML

* Si se pone el nombre solo del componente como titulo del estilo se cambiarán todas las etiquetas de ese tipo
* Si el titulo esta precedido por un punto, hace referencia a la clase y todas las etiquetas con esa misma clase se verán afectadas, de esta forma se afectan menos componentes que al referenciar una etiqueta
* Si el titulo esta precedido por un #, hace referencia al id y al ser un id algo que por buenas prácticas debería ser único, solo se vería afectado ese componente

Fue importante averiguar ciertas funciones como las de recargar la pagina o la de devolver la pagina a la parte superior

|  |
| --- |
| window.location.reload(); |
| *(función para recargar la página)* |

|  |
| --- |
| window.scrollTo(0,0) |
| *(función para llevar la página a la parte superior)* |

Tener presentes los tipos de eventos ayuda mucho a la interacción fluida con la página, en este caso se usaron dos:

* onclick="(acción que se debe iniciar con el evento)" – este tipo de evento se desencadena con un clic del mouse
* onblur="(acción a inicar)" – este evento se desencadena cuando el componente donde se encuentra pierde el foco

**aspectos a mejorar**

el proyecto tiene varios puntos donde podría mejorarse los cuales no se tocaron porque no eran parte del pedido ni relevantes para el cliente. Aun así, como programados hay ciertos puntos donde me gustaría poder hacer mejoras.

En la función de JS hay muchos condicionales de tipo if en la primera parte que seguramente se deben poner cambiar por un ciclo for, pero dado que la parte que debería cambiar es el argumento del método document.getElementById me fue imposible lograr hacer ese cambio

|  |
| --- |
| function validacion(){      let warning=[]      var bandera=false      if(document.getElementById('p11').checked==false && document.getElementById('p12').checked==false) {          warning.push("1")          bandera=true      }      if(document.getElementById('p21').checked==false && document.getElementById('p22').checked==false) {          warning.push("2")          bandera=true      }      if(document.getElementById('p31').checked==false && document.getElementById('p32').checked==false) {          warning.push("3")          bandera=true      }      if(document.getElementById('p41').checked==false && document.getElementById('p42').checked==false) {          warning.push("4")          bandera=true      }      if(document.getElementById('p51').checked==false && document.getElementById('p52').checked==false) {          warning.push("5")          bandera=true      }      if(document.getElementById('p61').checked==false && document.getElementById('p62').checked==false) {          warning.push("6")          bandera=true      }      if(document.getElementById('p71').checked==false && document.getElementById('p72').checked==false) {          warning.push("7")          bandera=true      }      if(document.getElementById('p81').checked==false && document.getElementById('p82').checked==false) {          warning.push("8")          bandera=true      }      if(document.getElementById('p91').checked==false && document.getElementById('p92').checked==false) {          warning.push("9")          bandera=true      }      if(document.getElementById('p101').checked==false && document.getElementById('p102').checked==false) {          warning.push("10")          bandera=true      }      if(bandera){          document.getElementById("advertenciaModal").innerHTML = "no has respondido las siguientes preguntas: " + warning          document.getElementById("mensajeModal").innerHTML = "Hay preguntas sin responder"          document.getElementById("img").src = "imagenes/advertencia.png";      } |
| *(función de JS sujeta a mejoras, debe ser posible cambiar tantos if por un for)* |

En segundo lugar, dado que la pagina debía recargarse después de salir del modal, esto hacía que las respuestas del formulario se borraran. hubiera sido excelente lograr que al mismo tiempo volviera al comienzo del formulario para diligenciarlo nuevamente en vez de que se quedara en la parte inferior y que tocara subir manualmente.

Este objetivo no se logro porque no se hayo una manera correcta de hacer una función con el window.location.reload y el window.scrollTo(0,0), pues en el momento de intentar implementarlo entraban en conflicto y alguna de las dos no se ejecutaba.