**Control de semántica**

La semántica, en los archivos parece correcta, en caso de cambio de posiciones de algunos elementos o cuando se utilicen ngFor de angular para generar elementos a partir de consumos del back-end, podrían implementarse además de la reducción de declaraciones, se podría hacer un mayor uso de etiquetas como <article> o <section>, aunque su correcta implementación dependerá de las necesidades del proyecto.

Con respecto al uso de <div> y las implementaciones de Bootstrap, es posible que a veces se use bastante , podría reducirse cambiando alguno de ellos por una etiqueta semántica, por ejemplo en Blog.html además de poder aplicar ng For (consumo del back o de un mock dentro de la app), podría cambiar <div class="container"> por <section class=”container”> , realmente son cambios menores y no creo que tengamos problemas debido a que en la primer evaluación que hicieron los profesores del trabajo la estructura semántica , pasó como aprobada.

Angular, por otro lado obligará a cambiar algunas declaraciones y/o usar ng Class, One way binding o Two way binding,

Otro cambio importante será en la navegación, ya que habrá que implementar angular routing lo cual cambiará la forma en la que navegamos dentro de la página ejemplos:

**En app routing module:** const routes: Routes = [

{ path: 'first-component', component: FirstComponent },

{ path: 'second-component', component: SecondComponent },

];

**En el html del componente:**

<h1>Angular Router App</h1>

<nav>

<li><a routerLink="/first-component" routerLinkActive="active" ariaCurrentWhenActive="page">First Component</a></li>

<li><a routerLink="/second-component" routerLinkActive="active" ariaCurrentWhenActive="page">Second Component</a></li>

</nav>

<router-outlet></router-outlet> (Aquí es donde se renderiza el componente)

Por último, en Angular es posible uso de condicionales dentro del HTML haciendo uso de la directiva ng If , con lo cual a un elemento del HTML le daremos una condicional , se escribe de diferentes formas, un ejemplo extraído de angular docs:

Si la condición del ternario es true(es una forma de usar una estructura condicional similar a un operador ternario en Angular):  
 <div \*ngIf="condition; else elseBlock">Content to render when condition is true.</div>

Si la condición es false renderiza esto:

<ng-template #elseBlock>Content to render when condition is false.</ng-template>

Importante:

Esta evaluación la hice basándome en los escasos conocimientos de Angular que tengo y el proyecto de Mountain Hike realizado hasta el momento, y para nada quiere decir que todo tenga que aplicarse de las maneras que aquí presento, pero cabe decir que son características básicas de Angular y deberían ser aplicadas en cualquier proyecto que se realice en este Framework.

**Control de responsive web design**

Importante: Estas correcciones están pensadas para el proyecto existente, aunque, al implementar Angular, se reducirá en gran medida el HTML necesario y como consecuencia de esto se podrá simplificar bastante aplicar diseño responsive y/o modificar diseños existentes.

-Corregir sección de reseñas para tamaño de IPhone SE , IPhoneXR, IPhone 12 Pro, Pixel 5, S8, S 20, y menores

-Tamaño menor a 1224 px x 937 px y hasta 822 px x 937px : Corregir posición de imágenes en la página nosotros

-Tamaño menor a 821 px x 937px: Corregir posición de imágenes en la página nosotros (quedaría una línea centrada o bien reducir el tamaño de los elementos para entrar de una forma más visual)

-Responsive Galaxy Fold ¿Será un requisito?, en caso contrario , habrá que rediseñar que se mostrará y como en todas las páginas del sitio, ya que es un tamaño de pantalla muy angosto y por por ello se vuelve poco intuitivo, difícil de leer y de utilizar en general.

**Posibles mejoras**

Propongo algunas mejoras (no se encuentran ordenadas por importancia y/o prioridad) :

- La página de Login, que sea un elemento flotante (como un modal window) de esta forma el usuario no tendría que acceder a una nueva página y si desea no loguear por el motivo que fuera después de ingresar al formulario, puede salir del mismo simplemente clickeando fuera del modal. Bootstrap posee modales que podrían ser utilizados para generar el modal del login de forma sencilla.

- Las imágenes utilizadas por el sitio subirlas a algún servidor, y luego que se consuma desde el back las rutas a las imágenes desde la data base. Esto permitiría aprovechar el uso de ng for para iterar sobre los datos recibidos provenientes de la base de datos y generar los elementos web de una forma mucho más sencilla y reduciendo el código HTML necesario.

- Convertir todas las imágenes al mismo tipo. Si son webp, jpg, etc. que todas sean del mismo tipo.

- Crear un modo dark o light. Esto permite a los usuarios elegir entre un fondo oscuro con texto claro o un fondo claro con texto oscuro, según sus preferencias personales o las condiciones de iluminación ambiental. Hacer esto requerirá la implementación de un botón (dark-light), además de los cambios CSS y de imágenes (como por ejemplo el logo de la empresa).

- Crear algo de movimiento en la web, con alguna animación sencilla y agradable, por ejemplo, modificar la sección Hero para que exista una animación que muestre imágenes con diferentes paisajes de personas practicando Trekking cada cuatro segundos por ejemplo.

- Dar más y mejor información al usuario en la sección Nosotros de la web, podrían darse datos más específicos sobre la historia de la empresa, fecha de fundación de la misma, por ejemplo. También agregar a las fotos de los guías los nombres de los mismos (también podría agregarse su apodo), o su experiencia laboral referida a esta actividad (simplificada), a fin de mostrar mejor al usuario quienes son los guías y generarle confianza. En este punto se puede modificar la configuración actual de “grilla” y pasarlo a mostrar de otra forma y con imágenes de mayor tamaño.

Crear una “llamada a la acción” para el usuario desde esta sección, ya que si los guías le generan la confianza necesaria, es muy probable que quiera contratar una guía, podría ser redirigido a la sección de recorridos o login desde este punto.

Mejorar la sección de envía tu curriculum, haciéndola más visual, agregarle alguna imagen.

- Mejora general de estilos en la sección de recorridos, posible búsqueda de una api, que informe el estado del tiempo en ese lugar, y que pueda ser consumida en cada elemento generado. De esta forma se evita que el usuario salga de la web para revisar el estado del clima en esa región, una variante podría ser que en la sección donde se harán las reservas, se muestre el clima y de estas dos posibles formas se puede eliminar el uso del botón “consultá el clima acá”

A continuación se puede observar un diagrama donde se muestra a modo de ejemplo una posible nueva estructura de tablas (sig página):

