

1-Contenido: Contar con al menos 4 páginas HTML, con temáticas de libre elección. Si el diseño es un estilo de página única (One Page), se requiere que contenga al menos 4 secciones.

Para el proyecto se creo un pequeño pero interesante modulo de reactividad el cual lo desarrolle personalmente y bautice con el nombre de **Js Reactor Tag v0.0.3 y también los documento .**

La documentación se creo específicamente para poder comprender el flujo de trabajo del proyecto. (esta adjunta en el directorio raíz del proyecto “Js Reactor Tags v0.0.3.pdf”

-----**INDICE DE DOCUMENTACIÓN.**-----

¿Que es JS Reactor Tags v0.0.3 ?

Prologo

1. Creación y utilización.

- 1.1 Enlazando Js Reactor Tags.*
- 1.2 Enlazando componentes.*
- 1.3 Creación de cuerpo de componentes.*
- 1.4 Uso de componentes.*
- 1.5 Reutilización de componentes.*
- 1.6 Reutilización de componentes por iteración de propiedad.*

2. Tipos de componentes y caracticas importantes.

- 2.1 Tipos de componentes.*
- 2.2 Características importantes de los componentes.*

3 Profundizando en los elementos Artesanos o Shape.

- 3.1 Creando un Resource en nuestro componentes*
- 3.2 Utilizando la característica Resource en nuestros componentes de tipo Shapes.*
- 3.3 Shapes dinámicas con datos calculados fijados y no fijados.*
- 3.4 Utilizando la característica Api Rest en nuestros componentes de tipo Shapes.*
- 3.5 Utilizando la característica de Contextos en nuestros componentes de tipo Shapes.*

4. Conociendo los componentes del tipo Constructores o Builders.

- 4.1 La potencia de los Builder sobre las Api Rest.*
- 4.2 Los Builder build_by_context y build_by_resource.*
- 4.3 Herencias de api sobre builders para construccion recursiva.*

4. Ciclo de vida de un componente y sus estados.

- 4.1 El ciclo de vida de un componente.*
- 4.2 Estado de un componente.*
- 4.3 Composición de un ciclo de vida por estados.*
- 4.4 Utilizando los estados de un ciclo de vida.*
- 4.5 Accediendo a la abstracción del componente desde sus estados.*

Epilogo

<i>2- Etiquetas semánticas: Utilizar etiquetas semánticas para estructurar el contenido del sitio.</i>
--

El proyecto se abordo en un enfoque de utilización de etiquetas y propiedades estructurales de índole semántica para mejorar :

La claridad como también el significado de los diferentes segmentos de código permitiendo de esta manera la correcta interiorizacion del integrantes del proyecto a la estructura general , así

mismo en caso hipotético de futuros ingresos de nuevos desarrolladores al proyecto permitirá que la integración sea mas amigable.

Minimizan de costos , aunque nuestro proyecto no requiere de costos económicos , la integración correcta de las etiquetas semánticas ayudo y ayudara a minimizar los costos de tiempo del integrante actual y de futuros integrantes frente a futuras actualizaciones.

Segmentación del flujo de trabajo para permitir al integrante , pese a la pequeña dimensión del proyecto a trabajar en paralelo en diferentes enfoques de la estructura.

Facilitan de la toma de decisiones sobre la responsividad y adaptabilidad , la diferenciacion semántico permitió facilitar la toma de decisiones de redimensionamiento de los diferentes segmento compuestos en el sitio, permitiendo llegar a la creación de un sitio web responsivo hasta con un mínimo de 200px de ancho.

Cumplimiento de los estándares de la WWW , html5 incorpore dichas etiquetas semanticas por que vio en ella una mejora de recursos actuales y futuros , permitiendo que el proyecto sea mas propenso a permitir la incorporación de actualizaciones y mejoras futuras de dicho lenguaje de maquetación.

3. Formulario: Incluir un formulario de contacto con validación mediante JavaScript. Como opción adicional, el formulario puede enviar un correo electrónico utilizando un servicio externo de correo.

Para la creación del formulario se utilizo el modulo reactivo propio Js Retag , interceptando la finalización de creación del componente `<section_contact></section_contact>` para la adición de lógica de verificación donde la misma tomo un enfoque para cumplir con las siguientes características :

- Organización de código
- Reutilización de código
- Separación de responsabilidades
- Delegación de responsabilidades
- Manejo de errores
- Interacción dinámica con el DOM
- Validación de campos de formulario
- Facilidad en la mantencion
- Facilidad en la legibilidad
- Facilidad en la aplicación de actualizaciones
- Separación de la lógica de la maquetacion
- Respetar la convenciones de nomenclatura

4-Diseño estético: Incorporar al menos un Iframe, íconos de FontAwesome, Flaticon o similar, y fuentes locales, de Google Fonts u otro similar.

El Diseño estético se enfoco en crear una identidad visual coherente para que se alinee intrínsecamente con la temática del sitio web , para producir como resultado un concepto distintivo , atractivo y modular que forme parte de un todo abarcando y cubriendo los siguiente conceptos :

- Poseer una tipografía acorde y legible.
- Poseer una paleta de colores atractiva , consistente y que evitar a toda costa la sobre carga cromática.
- Utilizar espacios en blanco de manera estrategica para separar semanticamente los subconjuntos perteneciente a un conjunto superior de elementos de sus peres adyacentes.
- Permitir una navegación clara e intuitiva
- Consistencia visual y estética a los largo de todas las paginas del sitio web.

Se utilizaron las siguientes tipografías provistas de *Google Fonts* de manera estratégica:

YsabeauSC-VariableFont_wght

IndieFlower-Regular

Mina-Regular

AlfaSlabOne-Regular

Goldman-Regular

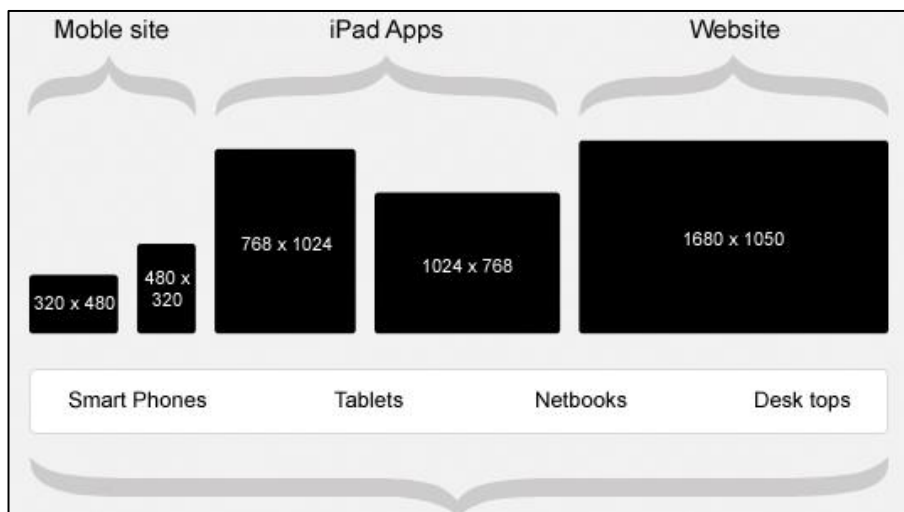
Se utilizaron recursos de iconos de *Google* de manera estratégica:

Material Symbols.

5-Diseño responsive: Incluir al menos una página completamente adaptable a tres tamaños de dispositivos diferentes (Mobile, Tablet y Escritorio). Esto garantizará que tu proyecto se vea correctamente en diversos dispositivos, brindando una experiencia óptima a los usuarios.

El proyecto tomo un enfoque receptivo de dispositivos bajo el contexto de uso de las tecnologías de usuario actuales abordando de esta manera los estándares de mobile tablets y desktop llegando a dimensionar la responsividad y adaptabilidad del sitio a un mínimo de 200px de ancho donde el usuario final podrá recibir la mejor experiencia posible dentro de los rangos elegidos por el mismo a lo largo de todas las páginas del sitio web completo.

Se contemplo la siguiente tabla de medidas para la toma de decisiones para cubrir las necesidades de adaptabilidad y responsividad :



6. Dinamismo visual: Introducir al menos una animación, transformación o transición en algún punto del sitio web (deberás indicar dónde se implementó la animación). Pueden utilizar librerías externas si lo desean.

Abordando las buenas prácticas del correcto diseño para mejorar la experiencia de usuario las animaciones y transiciones cumplen las siguientes características :

La no invasión entre contextos dados de conjuntos adyacentes , superiores y recursivos , una animación pertenece a una estructura semántica dada , evitando así , confundir al usuario en donde comienza y termina cada sección.

Respetar la estilización contextual , las animaciones respetan su ecosistema , como ser las dimensiones y la paletas de colores en otras.

Se estableció una duración y velocidad adecuada , ya sea en concepto de tomar la decisión de que la misma sea instantánea u ocupe un espacio en la línea de tiempo.

No interrumpen el objetivo del usuario , la existencia de las animaciones aplicadas consisten en evitar ser un paso a paso mas agregado en el camino hacia el objetivo del usuario, es decir , que el usuario podrá cumplir su cometido sin necesidad de esperar la finalización de ningún segmento de animación.

No ser un excedente , este punto se abordó desde un concepto de corte y prueba, donde se hace la siguiente citación :

Marucci Mauro - integrante:

“El primer sitio maquetado y estilizado que hice para el proyecto comenzó con 3 animaciones de renderizado de tamaño de los elementos y algo efectos de movimiento, todo muy sencillo , acorde seguí el desarrollo del mismo, me sumergí en la idea de querer crear y ver ‘vivo’ lo que acababa de hacer, comencé a colocar animaciones con cambios en los colores, agregue ciclos de animaciones infinitos, Una animación cruzaba por “encima” de otras estructuras invadiendo su espacio, algunas de ellas no necesitaban de la interacción del usuario para comenzar o finalizar , lo hacían de manera arbitraria y solo por la razón de que existían, las cosas se movían y se movían por todas partes y visualizándolas individualmente , se veían preciosas , pero siendo el proyecto un desarrollo bajo la condición de temática de empresa de viajes, luego de dos días de dedicarme a el , observe que el sitio web se parecía mas ...a una jugueteria o un circo ... había creado un monstruo y yo era el responsable de ello , dándome cuenta que cada elemento de un sitio pertenece a un conjunto mayor que lo contiene y que el elemento consumido y el cual esta destinado para los usuarios finales es el sitio web completo y no solo una parte de El.

”

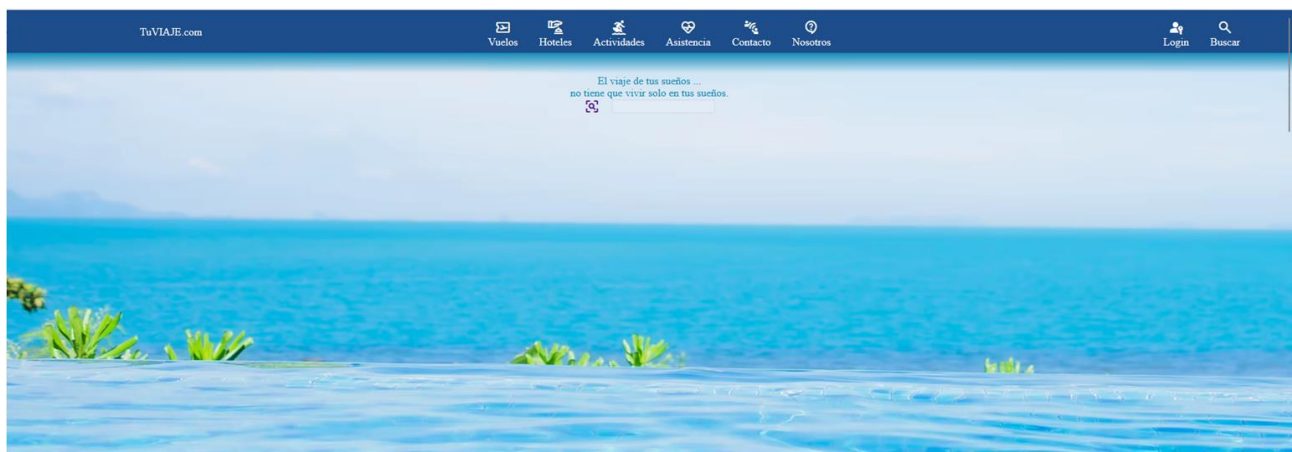
los segmentos que contienen las animaciones están delimitadas dentro del código por la siguiente línea de comentario:

```
/* PREPARE KEYFRAMES SUB SECTION *****/
```

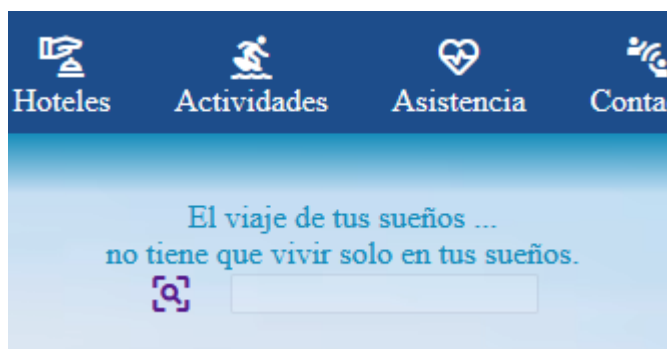
Algunos de los lugares donde se emplearon transiciones y animaciones fueron:

Header

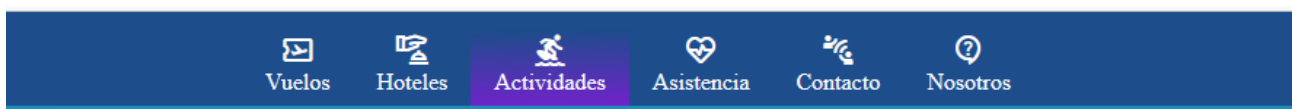
primer bloque receptivo preparado para el usuario , consiste en un segmento dado que promueven la inmersión del usuario en la primer instancia al ingresar al sitio web, mostrando una imagen de fondo con un leve movimiento horizontal, para esta animación se selecciono una imagen de resolución y muy alta calidad.



Dentro de este mismo bloque se puede observar el apartado de búsqueda rápida del middle-header el cual contiene una animación para mejorar la accesibilidad de escritura una vez confirmada la intención del usuario de generar la misma al recibir el evento flotante del mouse.



Navbar para medidas de desktop, generando una transición hacia un gradiente en el evento flotante del mouse sobre el elemento.



Navbar para medidas de tablets y phones , generando una transición hacia un color solido en el evento flotante del mouse sobre el elemento.



Sección logo de header esquina superior izquierda, se genero una animación de un avión despegando donde se evito crear innecesariamente un nodo de imagen , en su contraparte se utilizo y manipulo los aspectos relacionados a la propiedad de “back-groundimage” del contenedor de todos los segmentos del logo para manejar desde la imagen de fondo dicha animación.



Sección logo de header esquina superior izquierda, se genero una animación sobre el elemento contenedor del logotipo tipográfico asignado para el proyecto.



7-Maquetación: Utilizar una estructura HTML diseñada con Flexbox y/o Grid.

Para la correcta maquetación y estilización de la estructura del proyecto el enfoque se abordó sobre una composición de contenedores padres dotados de las propiedades `display: flex` o `display: grid`, permitiendo que la estructura en un futuro pueda recibir nuevas actualizaciones y/o incorporaciones de módulos nuevos o duplicados.

Los siguientes selectores (entre otros más) utilizan dichas propiedades:

- .pseudobody utiliza `display: flex`.
- .draggable utiliza `display: grid`.
- .nav-item-anchor utiliza `display: flex`.
- .middle-header-effect utiliza `display: flex`.

.article-flights utiliza display: flex.
.article-flights-body utiliza display: flex.
.article-flights-body-galery utiliza display: flex.
.article-flights-body-card utiliza display: flex.
.article-standard-header utiliza display: flex.
.footer-header utiliza display: flex
.footer-body utiliza display: grid.
.footer-widget-branch-list utiliza display: flex
.footer-widget-web-link utiliza display: flex
.footer-widget-contact-list utiliza display: flex
.footer-widget-1 utiliza display: flex
.footer-widget-2 utiliza display: flex
.footer-widget-3 utiliza display: flex
.footer-widget-list-material-symbols utiliza display: flex

8. API: Realizar una solicitud a una API Rest desde JavaScript. Debes indicar la API que usaste al momento de presentar el TPO.

Frente a la situación de que las apis gratuitas disponibles no se adecuaban a la información que requería el proyecto , se procedió a desarrollar archivos propios de JavaScript Object Notation para cubrir la necesidad con información personalizada.



Las llamadas a las Apis se realizan utilizando el catalizador de Js Reactor TAGS 0.0.3 , gran parte del sitio utiliza información obtenida desde estas apis.

9-Framework de CSS: En caso de querer usar Bootstrap (no obligatorio), solo podrán emplearlo en la página del formulario del sitio. No se admitirán proyectos desarrollados enteramente con Frameworks de CSS (como Bootstrap, Materialize, Tailwind, etc.) ni se podrán emplear estilos obtenidos de estos frameworks. Si estas pautas no se respetan, el trabajo práctico será considerado no válido.

Para la confección del proyecto se opto por la no utilización de framework de maquetación y estilizacion, para asbordar el mismo desde un enfoque que permita visualizar correctamente los conocimientos adquiridos en el transcurso del tramo de capacitación.

10-Publicación y GIT: La página web debe ser cargada en un servidor en línea para que el Instructor pueda navegar por ella. Pueden utilizar GitHub Pages, Netlify u otras opciones similares. Nota: El TPO es un trabajo en equipo. Deberán sincronizar el código del proyecto en un repositorio en GitHub (u otra plataforma similar) y enviar el enlace del repositorio al

Instructor al momento de entregar el proyecto. Este requisito es obligatorio. Para más información sobre GIT, consulta el material en el Aula Virtual bajo la sección "GIT".

El proyecto se desarrollo bajo las tecnologías de versionamiento de git en el contexto de repositorio local utilizando su consola gitbash , y github en el contexto de repositorio remoto utilizando su interfaz web.

12-Diseño Integral del Proyecto: Además de los aspectos técnicos previamente mencionados, se evaluará la parte de diseño del proyecto. Esto incluye: maquetación del sitio, alineación, orden y márgenes entre elementos, elección de tipografía acorde a la temática, uso de fotografías de calidad y optimizadas para la web, selección de paleta de colores adecuada, uso de logo e íconos, y favicon. Para asegurarte de cumplir con los criterios de diseño, te recomendamos revisar el material disponible en el Aula Virtual. Pueden encontrar un Seminario de Diseño Web en el enlace: <https://youtu.be/fVkJt2tYIj0>. Además, se proporcionará material adicional sobre diseño en el Aula Virtual bajo la sección "Diseño", que estará disponible después de la Unidad de Bootstrap.

Cubierto.

13-Documentación: Incluir en la entrega final del proyecto web un archivo de documentación preliminar en formato PDF. Esta documentación deberá contener la planificación, organización, objetivo y descripción del proyecto, detallando cómo se han abordado los diferentes aspectos del desarrollo. Las pautas y especificaciones detalladas para este punto serán proporcionadas en un documento aparte para asegurar una presentación consistente y adecuada de la documentación.

Documentación incluida en el mismo directorio del presente documento.