Descargué las fuentes desde Google Fonts en formato .ttf y las transformé a través de <https://www.fontsquirrel.com/tools/webfont-generator> y <https://transfonter.org/> en formato .woff y .woff2, ya que son compatibles con la mayoría de navegadores y ofrecen un buen equilibrio entre calidad y tamaño de archivo. Finalmente, escogí .woff2 al tratarse de la versión más reciente y ofrecer una mejor comprensión con respecto a .woff. Los archivos .woff2 son más eficientes en términos de rendimiento, al ser más pequeños, pueden ayudar a reducir el tiempo de carga de la web. A partir de este punto, hice lo mismo que en la PEC anterior, crear una carpeta con el nombre de la fuente, y guardar los archivos .woff2 en la carpeta local, para garantizar que mi web siempre se pueda leer con estas letras y no corra el riesgo de falta de disponibilidad de la fuente en caso de importarla directamente desde la web, ya que si en el futuro esta fuente dejara de tener soporte o se eliminara del sitio original, mi web ya no mostraría esa letra. Posteriormente, en el CSS, a través de @font-face, importo y creo la letra para cada tamaño con los archivos correspondientes y ya puedo usarlas en todo el proyecto. Lo ideal sería tener también en el proyecto tanto el formato .woff como .ttf para garantizar mayor compatibilidad con más navegadores y con navegadores más antiguos.

Para saber cómo se vería en el móvil o Tablet, dentro del navegador, le doy click derecho e inspeccionar. Desde ahí, le doy a Activar/Desactivar barra de herramientas del dispositivo, y escojo las dimensiones del dispositivo que me interesan para poder ver cómo quedaría, directamente desde la pantalla del pc. He usado las dimensiones del iPhone 12 Pro (390x844), iPad Air (820x1180).

En index.html para ubicar la imagen de fondo hero-home en versión móvil, he tenido que añadir padding top y bottom, así como jugar con background-size y background position hasta lograr el efecto parecido al que se pide. También, tuve que usar background-attachment:scroll para que la imagen se mueva junto al resto de la página cuando vamos bajando por el contenido.

Para crear el enlace vacío con aspecto de botón, he usado href=”#” en lugar de href=” “; no todos los navegadores pueden manejar el espacio en blanco de la misma manera, por lo que # es más correcto y ampliamente aceptado.

He usado las etiquetas <article> para crear secciones ya que al final son temas separados e independientes. He usado <span Lang=”en”> para añadir accesibilidad a la página y marcar cada cambio de idioma. He usado etiquetas <b> sin significado semántico, para marcar las palabras en negrita. Para convertir un email en enlace con protocolo mailto hice <a href="mailto:info@vital360.com">[info@vital360.com</a](mailto:info@vital360.com%3c/a)>.

Para ocultar el borde del formulario y el texto legend sin perder accesibilidad, he dispuesto el siguiente código en css: fieldset{ border: none; } legend{ opacity:0 }. Al parecer, si se oculta legend con display: none, dejaría de ser accesible para los programas lectores, por lo que opté por la opacidad.