Carlos de la Torre-Verdejo Sola

DISEÑO DE INTERFACES WEB  14/02/2021

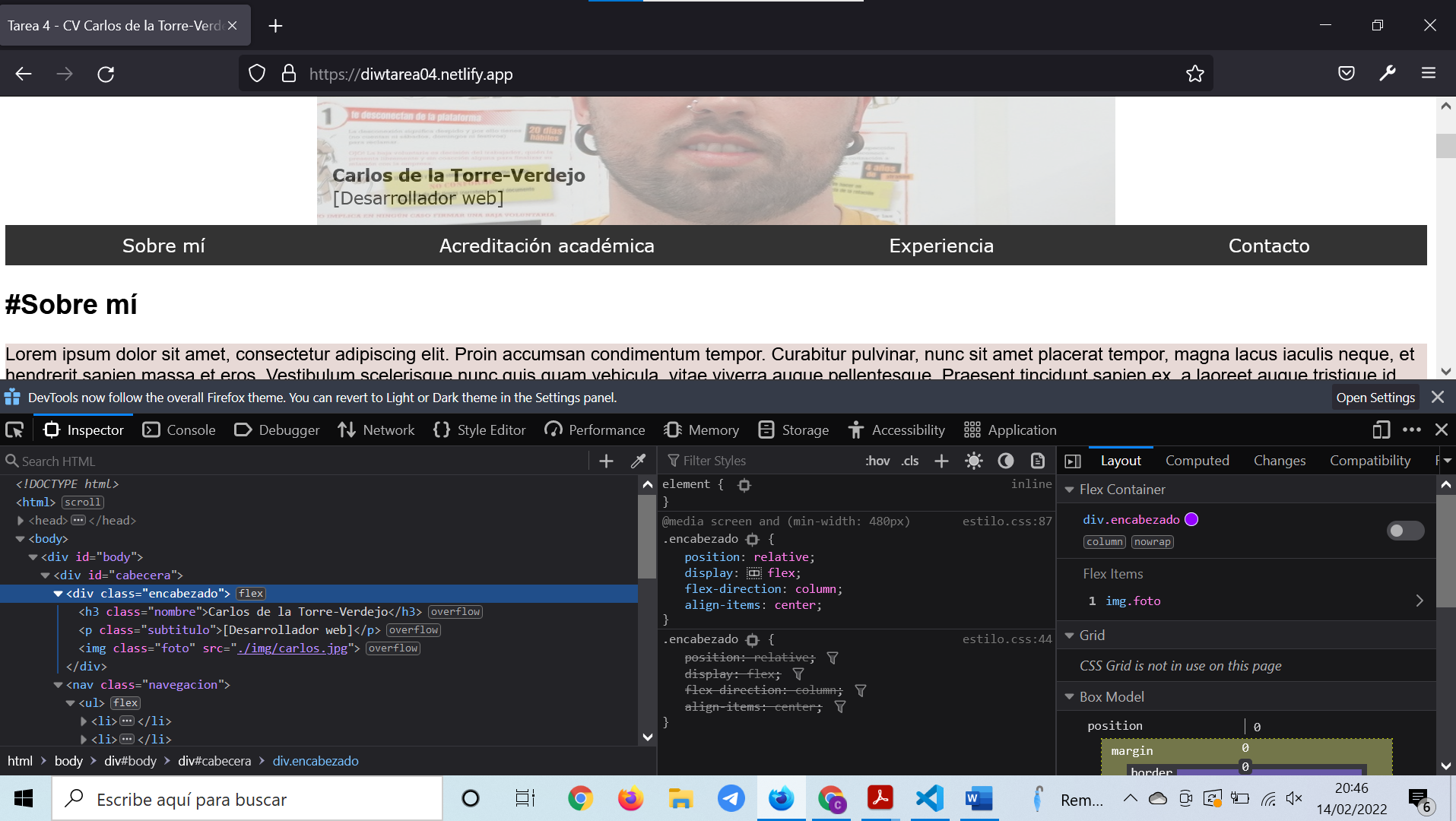
TAREA 3 DIW

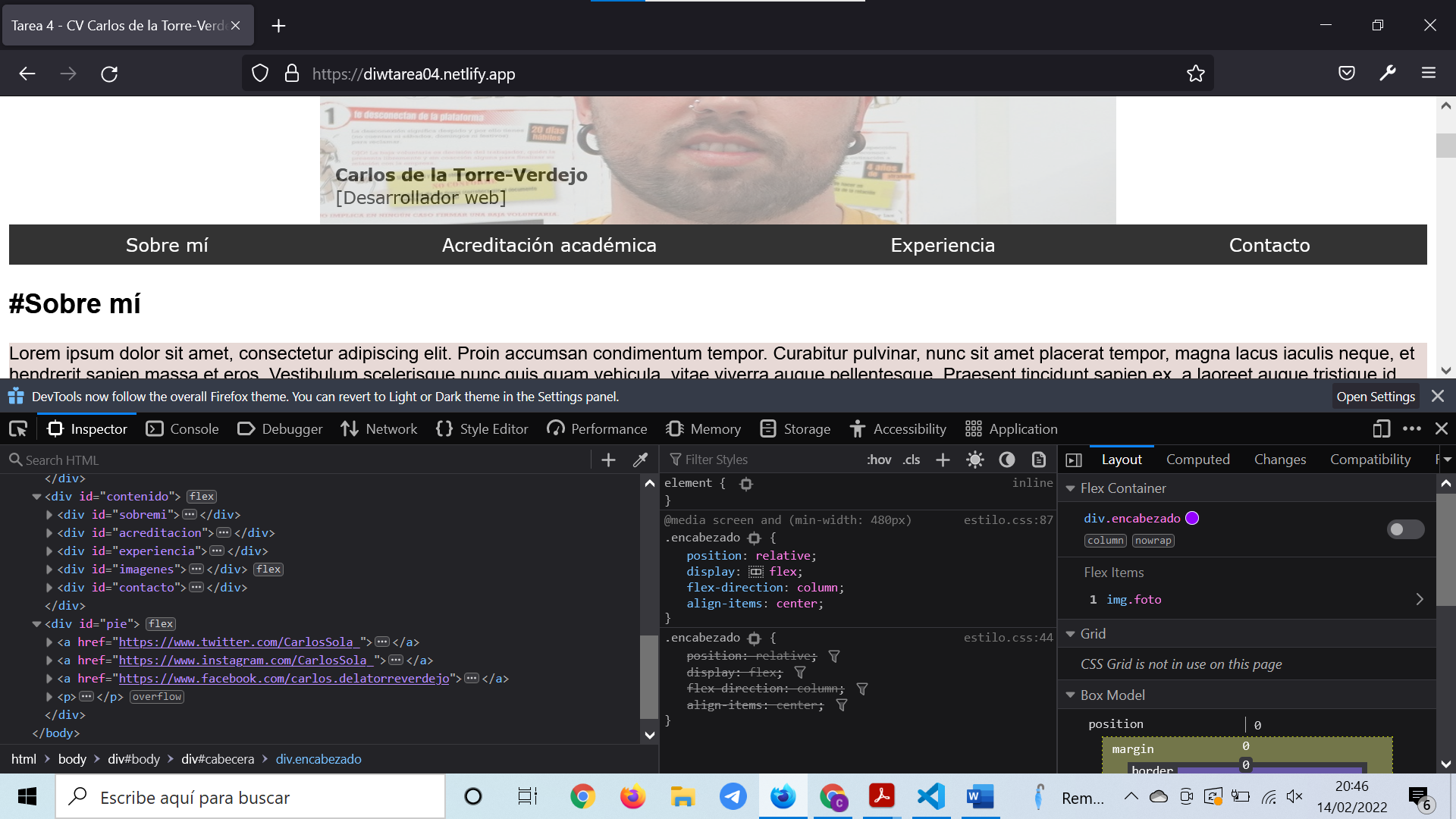
**MEMORIA DEL PROYECTO DE CONSTRUCCIÓN DE SITE WEB PARA PORTFOLIO HTML+CSS RESPONSIVE**

1. **Diseño escogido y partes del mismo**

En primer lugar, es importante señalar que se ha seguido el paradigma “mobile first” teniendo en cuenta que el requisito esencial de esta tarea es que nuestro proyecto tuviera un diseño responsive. Por tanto, se pensó la interfaz comenzando por una pantalla pequeña, de un smartphone, para posteriormente trasladar el diseño a todos los tamaños de pantalla.

Para ello, se tomó la decisión de utilizar Flexbox como módulo de diseño de CSS3 por su facilidad y practicidad a lo hora de diseñar páginas web responsivas. Cabe señalar que el aprendizaje adquirido tras el estudio de los distintos contenidos virtuales que se fueron exponiendo en el foro, han ayudado mucho al entendimiento de este módulo de diseño y su posterior uso en el diseño de esta página web. Facilita enormemente el uso de CSS y mejora sustancialmente el uso de float de CSS2, fundamentalmente. Aún así, debido al poco manejo sobre este módulo de diseño, se ha seguido utilizando características de CSS2 para el posicionamiento de algunos elementos, como se explicará a continuación. Si examinamos con Firefox Developer Edition nuestra página web, veremos los contenedores y elementos concretos en los que se ha usado Flexbox y veremos como ha sido en la mayoría de ellos:





Pasando a describir brevemente cada una de las partes y el por qué de los diseños escogidos para cada una de ellas, veremos cómo se ha seguido de forma natural la lógica de “contenedor flex” 🡪 “elementos flex” y, finalmente, aunque no se ha visto necesario su uso en algunos de los elementos, se ha mantenido la definición de su id o clase por si en un futuro fuese necesario adaptar dichos elementos concretos a un uso flex.

En primer lugar, se define el uso de una hoja de estilo guardada en la carpeta /css dentro del directorio raíz “Tarea 4”, de nombre “estilo.css”.

Posteriormente, se engloba dentro de un div con id “body” de cara a futuro, ya que de inicio no se ha planteado dar un estilo global a la página (como pudiera ser un color o imagen de fondo, por ejemplo), pero, entendemos que en un futuro podría ser conveniente y de ahí que se haya definido este contenedor.

El segundo contenedor ya se define con la idea de que sea utilizando flex, ya que dentro de él se incluyen múltiples elementos que deben ser responsivos y con los que tendremos que “jugar” para colocar de una u otra manera. Aún así, entendemos que dentro de ese contenedor “cabecera”, hay 2 subelementos principales: “encabezado” y “navegación”, que debemos separar para poder trabajar en el diseño, posicionamiento y tamaño de cada uno de ellos. De esta forma se define el encabezado como la imagen junto al nombre y puesto profesional superpuestos a esta imagen. Y por el otro lado, tenemos que el elemento de nombre de clase “navegación” será el menú de navegación que deberemos adaptar de forma vertical u horizontal en función del tamaño de la pantalla.

Si bien es cierto que esto se planteó así, el contenedor principal “cabecera” no tiene ningún diseño para el conjunto de los elementos debido a que no se vio necesario, pero igual que se apunta anteriormente en el caso del contenedor del elemento body, en este caso podríamos darle la misma explicación.

Por tanto, se les da un diseño flex a los elementos de los dos subcontenedores por separado. El encabezado lo diseñamos en columnas para posteriormente mediante CSS2 poder superponer el texto en la imagen, además de transparentar la imagen para hacer que el texto se vea de una forma clara. Igualmente, se centra dicho encabezado para que quede cuadrada la imagen y por último, se adapta tanto el tamaño de la imagen como los textos para que por un lado, no se sobredimensione y pixele la imagen y por el otro, sea el texto suficientemente grande para su lectura. En cuanto al subcontenedor de navegación, se diseña de una forma para las pantallas pequeñas y (en filas) y de otra para pantallas grandes (en una única fila y en columnas con espacio igual a su alrededor). Esto lo definimos mediante la sentencia “@media screen and (min-width: 480px)”. En este caso, en lugar de coger el contenedor en su conjunto, hemos aplicado el diseño a sus elementos concretos. Sin que exista una necesidad de explicar uno a uno los diseños de cada elemento, se van definiendo unos colores de fondo, un color del texto, su fuente y un color cuando pasamos el ratón por encima para todos los casos, sea en pantalla pequeña o grande. Y posteriormente, se define el diseño flex de los elementos de la barra de navegación para el caso de pantallas grandes.

La segunda parte principal del diseño es el contenedor del contenido de la página web, donde irán los datos que recogen nuestro CV. Aquí, también siguiendo la lógica de párrafos anteriores, se han definido subcontenedores para cada uno de los contenidos en concreto, aunque finalmente no se ha visto la necesidad de dar un diseño diferente a cada uno de ellos, pero es altamente probable que, en caso de retomar el diseño en un futuro, fuera recomendable e incluso necesario.

Por tanto, lo que se ha realizado es tomar el contenedor en su conjunto y aplicarle un diseño flex para hacerlo responsivo y posicionarlo en forma de columnas para así colocar cada una de las secciones de forma separada. Igualmente, se define la fuente del texto y se da un color igual para cada elemento “p” dentro del contenedor “contenido”.

Donde sí que ha sido necesario dar un diseño propio ha sido en el subcontenedor “imágenes”, ya que en este caso debíamos ocupar el espacio disponible de forma igual por cada uno de los elementos y ser responsivo, para que en el caso de las pantallas pequeñas pudiésemos ver las imágenes de forma continua frente al caso de las pantallas grandes, donde pudiéramos encajar las imágenes en varias filas, y no en una sola columna.

La última parte principal es el contenedor del pie. En este caso, además de dar un diseño flex al contenedor en su conjunto para ser responsivo y adaptar el contenido de dicho pie de página a la pantalla en concreto, se han determinado otros diseños sobre todo para cambiar el color a los logotipos de las RRSS, adaptar su posición para que estuvieran a la misma altura o adaptar su tamaño debido a la diferencia de tamaños y píxeles de cada uno de los PNG. Además, se alinean los elementos y se justifican dejando espacio igual alrededor de cada uno de ellos, ya que, bajo mi punto de vista, es la forma más limpia de colocar este tipo de pies de página o como se ha explicado con anterioridad, en el caso de menús de navegación poco cargados.

1. **Dificultades encontradas**

Las dificultades en un inicio parecían ser muchas, pero conforme se ha ido avanzando en el diseño del proyecto se han ido subsanando. Esto es, en un primer lugar resulta complicado pensar en “lógica CSS” debido a ser un lenguaje poco intuitivo y no controlarlo por el momento de una forma avanzada, pero gracias a la documentación, videotutoriales y ejercicios prácticos colgados en el foro de la UT y búsquedas mediante la web, se han ido resolviendo todas las dificultades.

La dificultad fundamental radicaba en el uso de flexbox, ya que con el resto no había ninguna duda, más allá de dificultades propias del uso de CSS con posicionamientos de elementos, adaptación de tamaños y cuestiones por el estilo, que finalmente se resuelven mediante la prueba y numerosos cambios y comprobaciones. Pero como se comenta, estas dificultades han sido resueltas y sobre el papel, aunque queda mucho camino por recorrer y mucho más por aprender y mejorar sobre el diseño tanto flexbox como CSS en general, se podría decir que el reto que imponía el desarrollo de esta tarea, ha sido superado satisfactoriamente.

No obstante, cabe señalar para finalizar que, debido a cuestiones fundamentalmente laborales y militantes, mi tiempo para la realización de esta tarea ha sido muy reducido y por ello, los contenidos de lo que debía ser mi CV, no han sido incluidos. Esto, de cara a futuro, debe ser corregido mediante la correcta priorización de tareas y organización propia de, en general, los trabajos de los distintos módulos encajando con el resto de tareas vitales.

1. **Enlaces al sitio web en despliegue continuo y repositorio con el código**

Enlace al sitio web en despliegue continuo Netlify: [diwtarea04.netlify.app](https://diwtarea04.netlify.app/)

Enlace al repositorio de GitHub: <https://github.com/carlosdtv/DIW_TAREA4>