Primero de todo he hecho un git clone con el directorio del proyecto UOC Boilerplate. Seguidamente he borrado la carpeta .git. Despues con el comando git init la he creado de nuevo y con git commit -m "mensaje" he hecho mi primer commit. Con el comando git remote add origin seguido de la URL de mi directorio de github he vinculado mi trabajo en local con el repositorio online. Seguidamente con git push -u origin master he actualizado mi repositorio online con el local.

En este proyecto ya el packaga.json por ese motivo no teníamos que crearlo con npm start, pero si que debiamos de hacer npm install para instalar todas las dependencias. Con el comando npm run dev nos elimina el historial del ejecutado anteriormente y nos abre un puerto en local. Con el comando npm run build nos prepara toda la página minificada en la carpeta dist preparada para subirla a producción.

Para la publicación en Netlify le das al botón ADD A NEW PROJECT, seguidamente lo linkeas con github en mi caso y luego sigues las instrucciones hasta llegar a donde tienes que decidir la branch que en mi caso es la master y el Dir que es donde npm run build te crea la página es decir en dist. Y, por último, en build commando el comando de producción es decir npm run build. Despues cada vez que hagas un push se actualizará automaticamente.

Luego para instalar el paquete de tailwind he escrito por línea de comandos: npm install tailwindcss --save-dev y seguidamente con el comando npx tailwind init he generado el documento base de configuración. En este documento he añadido las media queries.

module.exports = {

theme: {

screens: {

lg: { max: "975px" },

// => @media (max-width: 1023px) { ... }

md: { max: "700px" },

// => @media (max-width: 767px) { ... }

sm: { max: "480px" },

// => @media (max-width: 639px) { ... }

},

},

variants: {},

plugins: [require("@tailwindcss/custom-forms")],

};

En el documento css colocamos

@tailwind base;

@tailwind components;

//Todo nuestros componentes con @apply

@tailwind utilities;

// seguido de nuestras utilidades

* ¿Qué diferencias hay entre el enfoque de tipo CSS semántico (el que usaste en las otras PEC) al respecto del CSS de utilidades? ¿Cómo afectó esto a tu proceso de desarrollo? ¿Y a tu código?

El semántico su nombre te informa de su procedencia y de la estructura a la que forma parte. Esto te ayuda a identificar con facilidad de que forma parte un componente. El de utilidades se centra exclusivamente en describir una funcionalidad lo más específicamente posible siguiendo un patrón.

* ¿Qué diferencias encontraste entre usar una libería de componentes y una libería de utilidades?

Una de componentes como Bootstrap te da todo más hecho por consecuente es más fácil de usar mientras que una librería de utilidades como tailwind es como escribir css dando clases al html.

* ¿Qué componentes decidiste extraer y por qué?

He extraído los myNavbar\_\_item—selected porque en el hover hago una transición y he considerado que era algo muy especifico que iba a ser difícil de estandarizar con un nombre adecuado con el cual crear una clase. Con grayImg me ha pasado lo mismo que con myNavbar\_\_item—selected. La última, firstBigLetter-w-300px, la cual cambia el tamaño de la primera letra de cada párrafo, me ha parecido fuera de las opciones que ofrece tailwind y por ese motivo la he extraído.

He intentado extraer el mínimo creando nuevas clases cuando lo necesitaba siguiendo el criterio de nombres de tailwind

Git: <https://github.com/netraluis/Herramientas-HTML-y-CSS-II-PEC-Repetida3>

Netlify: <https://sad-mahavira-efbcd8.netlify.app/>