

Accesibilidad

En este documento revisaremos todos los elementos de nuestro código para dotar a nuestra aplicación de una buena accesibilidad. Destacar que los colores, estilos, etc... están bastante estudiados e intentamos desplegar una interfaz lo más sencilla y fácil de usar posible.

- **html**: Introducimos la etiqueta `lang="en"` para que el asistente de voz ponga el acento correspondiente e indicar que nuestra página está en inglés.
- **title**: Cambiaremos el contenido dinámicamente dependiendo de la pantalla en la que se encuentre. `PokeApi` para la primera pantalla y `Resultados` para la segunda, para aclarar en qué pantalla se encuentra.
- **div (filters)**: Tiene un `lang="es"` dado que está en español. Destacar también que cambiamos los `spans` por `labels` y añadimos 2 botones. Como se oculta y muestra, debemos jugar con el `aria-expanded` (en vez de con el `aria-hidden` dado que no nos funcionaba y nos sugirieron que probásemos con este atributo), le añadimos el role `toolbar` también. Y finalmente el `aria-controls="contenido"`. También hemos decidido añadirle un `aria-label="Filtros de Pokemons"` dado que no está de más tenerlo. En cuanto al orden del `tabindex`, nos daría igual dado que de primeras va a estar colapsado.
- **label**: Dado que antes teníamos `spans`, los reemplazamos por `labels` que asociamos a los diferentes `selects` y que son de ayuda para comprender el significado de los `inputs`.
- **selects**: Estos elementos actualizaban la lista de pokemons con que solo cambiase su estado, por lo que decidimos cambiar este comportamiento por 2 botones (limpiar y aplicar).
- **button (filtros)**: Finalmente, decidimos añadir los dos botones que mencionamos anteriormente para facilitar el uso al usuario.
- **header**: Antes teníamos un `nav` y un `div` y lo cambiamos por la etiqueta `html` nativa con el rol correspondiente. Dentro de este header se muestran y ocultan diversos elementos por lo que tendremos que añadir las etiquetas correspondientes (las veremos a continuación).
- **button (header)**: Ambos botones cumplen la función de desplegar los elementos necesarios para realizar las tareas de o bien buscar o bien filtrar. No nos olvidemos del botón de volver de los resultados. Pues bueno, todos se pueden ocultar por lo que añadiremos los `aria-hidden` correspondientes. Y dado que todos hacen que aparezcan elementos tenemos que introducir los atributos `aria-haspopup` necesarios. Destacar también que como tenemos iconos en vez de texto deberíamos incluir un `aria-label` para aclarar todo.

- **i:** Al usar font awesome para poner iconos como si de una fuente se tratase y siempre estar contenidos dentro de un botón, será dicho botón el que se encargue de aclarar su funcionamiento.
- **a (título):** Se oculta y muestra en función de search, por lo tanto jugaremos con el **aria-hidden**.
- **input (search):** De la misma forma que los filtros, este input modificará el contenido del elemento contenido, consecuentemente, añadimos **aria-controls="contenido"** y dado que se muestra y oculta en función del botón de búsqueda, jugaremos con **aria-hidden**. Por último, añadimos **aria-label** dado que con el placeholder no sería suficiente y no tenemos ningún label que nos ayude a saber que debemos a introducir.
- **main:** Añadimos un main para indicar que es uno de los contenidos principales de la página, antes simplemente teníamos 2 divs hijos de body.
- **div(contenido):** Añadimos un **role list** ya que contendrá la lista de pokemons. Declaramos este elemento como una región activa mediante **aria-live="polite"** ya que cambiará dinámicamente. Este atributo lo asignamos como polite para que no interrumpa a la usuaria en caso de que esté escribiendo o leyendo. Relacionado con esto también añadimos un **aria-atomic="true"** y un **aria-relevant="additions"**. Finalmente, añadimos un **aria-hidden**, porque se ocultará cuando se lance la segunda pantalla. (está en el fichero js)
- **div(pokeitem):** Añadimos un **role listitem**, porque son los ítems que mostrará el main(contenido) con role list. También añadimos un atributo **aria-haspopup** ya que si se clicca aparecerá la pantalla de resultados. Se añadió el atributo **tabindex** para hacerlo accesible mediante tabulador. Y un **onkeypress** para acceder al popup resultados.
- **img(pokeitem):** Ya tenemos el atributo **alt** que dota de una alternativa textual.
- **span(pokeitem):** Habrá un span para el nombre, el id y los tipos del pokemon. Destacar que cuando no hay una imagen tenemos un span que la substituye y de esta forma este span tendrá el **role="img"**.
- **button(go-up):** Como en la segunda pantalla no estará presente, añadimos un **aria-hidden**.
- **div(resultados):** Este div tendría un comportamiento muy similar al **div contenido** y de este modo, tendrá los mismos atributos.
- **img(resultados):** Tenemos un atributo **alt** que dota de un texto alternativo para ayudar al usuario y en caso de que no se pueda cargar la imagen, un span tendrá el **role de imagen**.

- **span(resultados):** No hay mucho que comentar, al nombre del pokemon se le añade un `tabindex` para que sea accesible desde el teclado lo único.
- **divs (tarjetas resultados):** Le añadimos un `tabindex="0"` para que se puedan leer los elementos accediendo a ellos desde el teclado.
- **span(fact):** Representa el título que aclara de qué va el texto que viene a continuación. No se le añade nada.
- **p(fact):** Le añadimos un `aria-describedby="fact"` para que quede claro que este texto es un dato curioso del pokemon.
- **span(base stats):** Representa el título que aclara que se van a presentar los estadísticas base.
- **table:** En cuanto a las tablas, sería interesante añadir un **caption**, pero hemos decidido no hacerlo en el caso de las base stats y adaptar lo que tenemos con un `aria-describedby = "stats"`. En el caso de la tabla de **peso, altura y base** metimos un **caption** oculto. Consideramos que se ve perfectamente en qué consiste la tabla, pero bueno para una persona con capacidades visuales limitadas sería interesante que el asistente le concretase en qué consiste.
- **tbody:** Como es una etiqueta nativa no hace falta añadirle un role.
- **tr:** Como es una etiqueta nativa no hace falta añadirle un role de fila.
- **th:** Como es una etiqueta nativa no hace falta añadirle un role.
- **td:** Como es una etiqueta nativa no hace falta añadirle un role.
- **span(types título):** Mismo caso que con **fact**.
- **span(types):** Mismo caso que con **fact** con los diferentes tipos.
- **div(base stats):** Con dos divs y estilos, se representa una especie de progressbar que indica el porcentaje del pokemon sobre una característica. De esta forma le pondremos un `role="progressbar"` y consecuentemente los `aria-valuenow="x"` `aria-valuemin="0"` `aria-valuemax="255"`