



MEDIWEB

Antón Vilas Pazos
Hugo Viqueira Ans
Sofía Pazos Pazos

1. Actualizaciones y cambios de estado por los que transita la interfaz:

La interfaz de nuestra aplicación muestra un campo de entrada de texto correspondientes al buscador del código de paciente y 2 campos de entrada de tipo fechas correspondientes a las fechas de Inicio y Fin del intervalo temporal a consultar. Cuenta también con dos listas (una de medicamentos y otra de tomas de medicamentos, en un principio vacías) y 5 botones con un texto breve explicativo, que al pulsarse modifican los campos de fechas Inicio y Fin del intervalo (de lado del último de ellos existe un campo que permite al usuario indicar el número de días a abarcar).

Al introducir un código de paciente correcto (es decir, que sigue el formato y existe en la base de datos) y seleccionar el periodo de consulta (Fecha inicio y fin) y presionar el botón de confirmar, la pantalla, aunque sigue siendo la misma, cambia de estado y pasa a mostrar además de lo que ya había en ella la lista de medicamentos asociados a ese paciente durante esas fechas y sus tomas.

2. WAI-ARIA:

Accesibilidad:

Buscamos conseguir que los contenidos de la web tengan las siguientes características para hacerlos accesibles.

Entendibles:

- Indicamos el idioma de la página para hacer el contenido textual entendible y legible (<html lang="es">)
- Asistencia de entrada, para ayudar a los usuarios a corregir errores (proporcionando etiquetas, instrucciones y explicación del error).

Operables:

- Es navegable, las cabeceras y etiquetas describen los propósitos de las mismas.
- Todas las funcionalidades son operables desde el teclado.
- Las funcionalidades que implican gestos son accesibles con un tap o click.
- Tamaño mínimo de elemento pulsable (24*24px)

Perceptibles:

- Identificamos el propósito de los campos de entrada, botones y regiones de la interfaz.
- Facilitamos la visualización y escucha del contenido: Las acciones y la información no dependen del color (aunque se use algún color para resaltar la importancia de lo mostrado, todo está respaldado por explicaciones con texto).
- Se usa un contraste por encima del mínimo y no se usan imágenes con texto.

- Se emplean tonalidades azules para la interfaz teniendo en cuenta que este es uno de los colores que mejor distinguen las personas con cualquier tipo de vista.

Robustos:

- Para compatibilidad con tecnologías asistivas, se usa html semántico y se realizan adecuaciones para que, ante problemas de visión, se pueda con una herramienta de lectura de pantalla interactuar con la interfaz de la forma más sencilla y entendible posible.
- Proporcionamos nombres, estados y roles correctos a los componentes creados.

Normas de accesibilidad WAI-ARIA:

Para ajustar la implementación de nuestra aplicación web a las normas de accesibilidad en aplicaciones Web WAI-ARIA, hemos realizado las siguientes acciones:

- Uso de roles WAI-ARIA:
Siguiendo las reglas de uso de WAI-ARIA incluimos tags de rol únicamente cuando no haya tag html semántico.
Como las etiquetas necesarias para las necesidades de nuestra aplicación existen en html semántico (<main>, <form>) no empleamos Landmark roles para identificar las regiones existentes en el documento html. Usamos html semántico para permitir una correcta navegación para herramientas de lectura de pantalla.
Tampoco usamos Document structure roles, ya que usamos el tag de html semántico correspondiente a las listas , para describir estructuras como en este caso una lista, que organiza el contenido de la página.
Ni roles de widget, ya que gracias al html semántico podemos hacer uso de los tags <grid> y <button>.
- Atributos de widget: (propiedades y estados)
Propiedades:
Empleamos aria-label (para poner más información en los elementos, necesario para correcto funcionamiento de la aplicación con uso de lector de pantalla)
Usamos aria-required (define campo obligatorio) Lo combinamos con required (html 5) para compatibilidad de navegadores y herramientas de lectura de pantalla.
- Live región:
Hacen referencia a la posibilidad de que el contenido de una región cambie de forma dinámica sin que el usuario interacciones directamente con ellas (cuando se hace click en un botón establecer y las fechas cambian el usuario es notificado tanto visualmente como a través del lector de pantalla).

Usamos el siguiente atributo de WAI-ARIA:

aria-live [assertive], que permite que se detecten los cambios en la pantalla, haciendo posible que el lector de pantalla pare de leer lo que estaba leyendo y lea los cambios producidos.

- Uso del atributo de WAI-ARIA de tabindex, para informar a los navegadores web de que elementos como las listas y las capas div pueden recibir el foco.
- Para facilitar la navegación realizamos lo siguiente:
Si no se introduce código y se le da a confirmar, salta error y llevamos el foco al campo input de código de paciente. (facilidad para que el usuario sepa qué hacer), hacemos lo mismo con el botón de establecer y el campo input de N días, para esto usamos la función focus().
Cuando se hace click en cualquier botón de establecer salta una notificación que indica que el cambio se ha realizado correctamente (para mayor claridad visual y facilidad para que el usuario saber lo que desencadena la acción que ha realizado.
- También se ha tenido en cuenta que no se debe cambiar la semántica de los elementos y que todos los controles nativos de ARIA tienen que poder usarse a través del teclado.