# DOCUMENTACIÓN DE FALLOS CON RESPECTO A LAS NORMAS WAI-ARIA

## DEFINICIÓN DE WAI-ARIA

El W3C* define WAI-ARIA (Web Accessibility Initiative - Accesible Rich Internet Applications) como "**La forma para crear contenido Web y aplicaciones Web que sean accesibles para las personas con discapacidades**."

WAI-ARIA es una especificación del W3C, pensada para hacer más accesible el contenido dinámico y los controles desarrollados con Ajax, HTML, Javascript y sus tecnologías relacionadas, permitiendo transmitir información sobre el comportamiento de la interfaz y su estructura a las APIs de accesibilidad para que sea utilizada por los productos de apoyo para la interacción con el usuario final.

WAI-ARIA proporciona una ontología de roles, estados y propiedades que definen los elementos de la interfaz. En el documento del W3C se indican los pasos y buenas prácticas para aplicar esta especificación.

(*) Nota: W3C (World Wide Web Consortium) es una comunidad internacional que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo.

## APLICACIÓN DE WAI-ARIA

### -MARCADO NATIVO

Se recomienda utilizar marcado nativo siempre que sea posible. Esto consiste usar elementos nativos en vez de sus roles equivalentes. Por ejemplo, utilizar <button> en lugar de .

En nuestro código:

```
<section>
    <h2>LIGA SENIOR MASCULINA</h2>
    <a href="HockeyLinea.html" id="volver">Volver Competiciones</a>
        <div id="botones">
    <button class="boton">CLASIFICACIONES</button>
    <button class="boton">CALENDARIO Y RESULTADOS</button>
    <button class="boton">DESIGNACIONES ARBITRALES</button>
    <button class="boton">GOLES</button>
    <button class="boton">ASISTENCIAS</button>
    <button class="boton">TARJETAS</button>
    <button class="boton">PLANTILLAS</button>
    <button class="boton">PUNTOS</button>
    </div>

    <div class="imagen">
    <img src="imagenes/LOS.png" style="width:100%" alt="LOS">
    <div class="contenedor">
      <a href="">LOS</a>
    </div>
    </div>
```

En este fragmento de código perteneciente a LigaSeniorMasculina.html se puede observar que se utiliza el elemento <button> en vez de un .

**-ROLES ADECUADOS**

Se recomienda utilizar los roles según su especificación, y no cambiarlos dinámicamente. En nuestro código no se utilizar roles.

**-ESTRUCTURA SEMÁNTICA**

Se recomienda conservar la estructura semántica; para ello, se pueden formar grupos lógicos, incluir *landmark roles* que faciliten la navegación por teclado, y definir las *live regions* o zonas que cambian dinámicamente sin intervención del usuario. Como se ha mencionado en el punto anterior, en nuestro código no se utilizan roles. Tampoco existen zonas que cambien dinámicamente.

**-RELACIONES**

Se recomienda construir relaciones entre los elementos, marcándolas con el atributo más apropiado.

**-ESTADOS Y PROPIEDADES**

Se recomienda cambiar (por javascript) los estados y propiedades que se introduzcan durante el ciclo de vida del elemento, normalmente en respuesta a los eventos de entrada del usuario. Así, los agentes de usuario notificarán a los productos de apoyo los cambios de estado. Además, se deben usar solo atributos soportados por el rol o el elemento elegido. En nuestro código no se utiliza javascript.

**-ACCESIBILIDAD POR TECLADO**

El usuario debe poder interactuar con los controles mediante el teclado, accediendo a ellos mediante el tabulador y las teclas de flechas en el caso de controles complejos. Además, se debe seguir un orden de tabulación lógico. Nuestro código permite el acceso por tabulador y sigue un orden de izquierda a derecha y de arriba abajo.

**-SINCRONIZACIÓN DE INTERFACES**

La interfaz visual y la interfaz accesible deben estar sincronizadas. Para ello, se puede recurrir a los CSS attribute selectors. En nuestro código se ha decidido no usarlos, ya que no los vemos necesarios.