



## WAI-ARIA

© 2017, ACTIBYTI PROJECT SLU, Barcelona  
Autor: Ricardo Ahumada



red.es



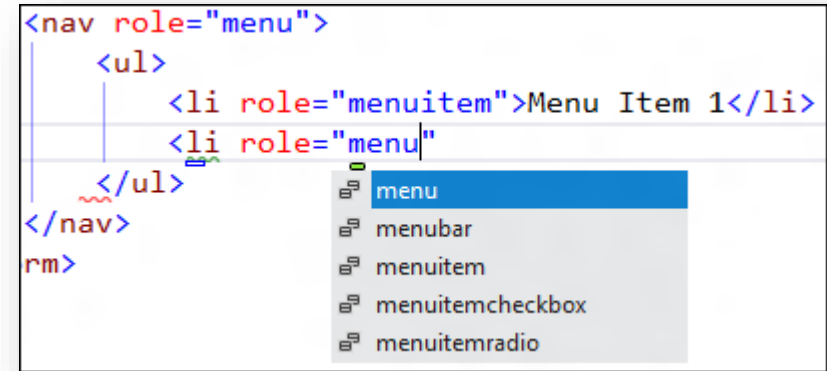
**UNIÓN EUROPEA**

Fondo Social Europeo  
"El FSE invierte en tu futuro"

# WAI-ARIA

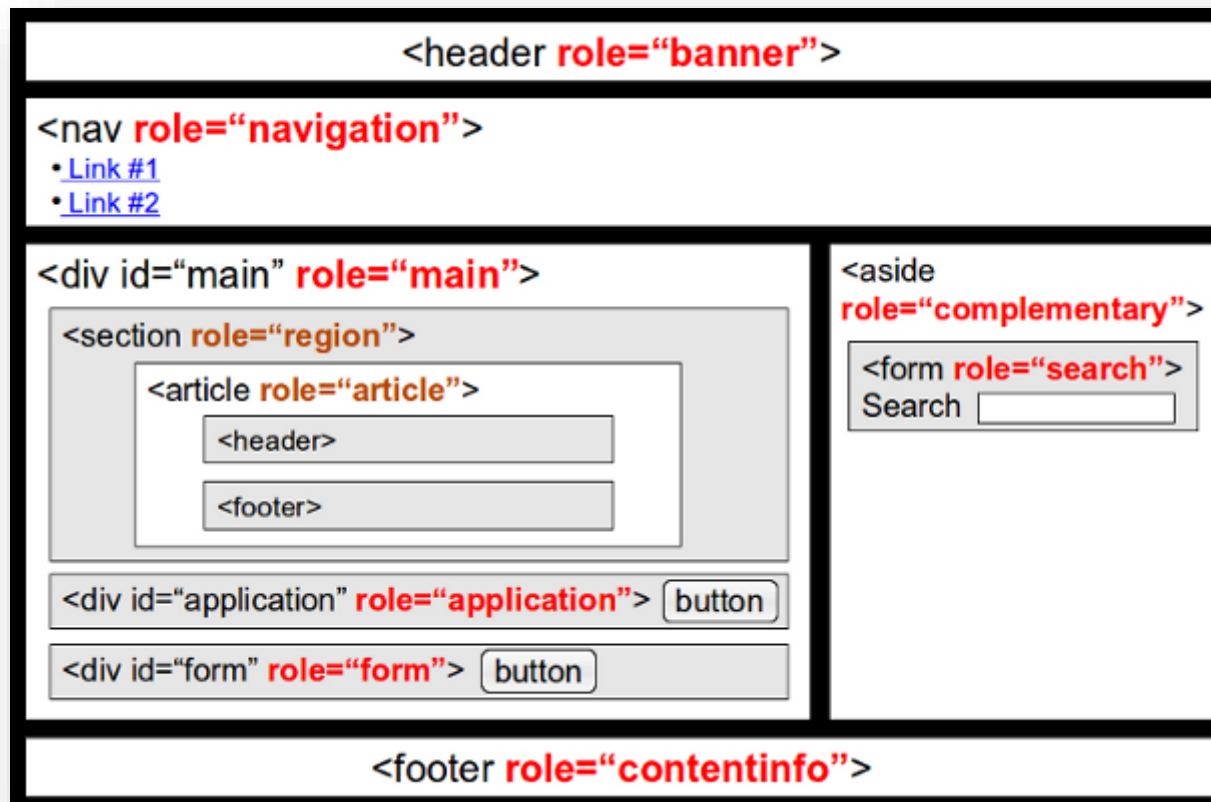
- Es una tecnología que se encarga de aportar información acerca de las diferentes partes que constituyen los contenidos dinámicos y los controles avanzados de interfaz generados por scripts.
- Esta es usada por productos de apoyo de interacción con el usuario final.
- Es decir, es una suite que determina la manera de elaborar contenidos y aplicaciones de internet más accesibles.
- Para cumplir su objetivo, se basa en un conjunto de atributos que se comportan como identificadores de las diferentes partes de la aplicación que interactúa con el usuario.

```
<nav role="menu">
  <ul>
    <li role="menuitem">Menu Item 1</li>
    <li role="menu"
  </ul>
</nav>
rm>
```



# Roles

- Describe el tipo de widget. Estos pueden ser de varios estilos, desde menús desplegables en forma de árbol, hasta sliders.
- Describe la estructura de la página web. Pueden definir elementos de encabezado o incluso, tablas y grids.



# Lista de roles

- La definición de los roles se encuentra en <https://www.w3.org/TR/wai-aria/roles>
- **Widgets**
  - alert, alertdialog, button, checkbox, dialog, gridcell, link, log, marquee, menuitem, menuitemcheckbox, menuitemradio, option, progressbar, radio, scrollbar, slider, spinbutton, status, tab, tabpanel, textbox, timer, tooltip, treeitem
- **User interface widgets**
  - combobox, grid, listbox, menu, menubar, radiogroup, tablist, tree, treegrid
- **De organización**
  - article, columnheader, definition, directory, document, group, heading, img, list, listitem, math, note, presentation, region, row, rowheader, separator, toolbar
- **Navegación**
  - application, banner, complementary, contentinfo, form, main, navigation, search

# Atributos ARIA

- Un atributo ARIA proporciona información específica sobre un objeto concreto y forma parte de la definición de la naturaleza de los roles.
- Tiene un comportamiento dinámico (no como el rol que no cambia), permitiendo definir las propiedades y estados de los elementos.

```
<span role="checkbox" aria-checked="true">  
</span>
```

- Realmente estos atributos se dividen en dos tipos: **propiedades y estados**. Sin embargo su diferencia es muy sutil (como tantas otras a las que tan acostumbrados nos tiene la W3C).
- La mayor diferencia es que las propiedades cambian menos que los estados.
- Para profundizar un poco más: WAI-ARIA 1.0 (Propiedades y Estados Soportados).
  - [https://www.w3.org/TR/wai-aria/states\\_and\\_properties](https://www.w3.org/TR/wai-aria/states_and_properties)

# Estados y propiedades

- **Atributos Widget:** orientados a componentes en los que el usuario ha de introducir una entrada de datos para procesarla.
- **Atributos de regiones activas:** son los destinados a las regiones activas, aquellas susceptibles de incorporar contenido dinámico actualizable. Pueden ser aplicados a cualquier elemento.
- **Atributos de Drag and Drop:** proporcionan información sobre los efectos Drag and Drop (arrastrar y soltar) que se encuentran en algunas aplicaciones.
- **Atributos de relaciones:** tienen la función de relacionar elementos que no se puede establecer mediante la estructura del documento.

Lo más importante, y que no debes olvidar, es que no solo tienes que indicar el estado y propiedades iniciales, sino que debes **modificarlos dinámicamente por javascript** cuando estos cambien, de modo que el usuario de productos de apoyo conozca siempre su estado actual. Los roles no se actualizan, los estados y propiedades sí.

# Algunos estados y propiedades Importantes

## ➤ **aria-live**

- permite identificar una zona dinámica de nuestro contenido que se actualiza automáticamente, de esta manera los cambios se anunciarán al usuario de los productos de apoyo.
- En función de su valor (off, polite, assertive) indicaremos cuándo queremos que se anuncie la actualización.
- Se combina con aria-atomic para indicar si queremos que se anuncie toda la región o solo las partes que cambian, y con aria-relevant para indicar el tipo de actualización que queremos que se anuncie.

## ➤ **aria-label, aria-labelledby y aria-describedby**

- Tanto aria-label como aria-labelledby permiten etiquetar un elemento.
  - La diferencia es que con aria-label indicas directamente el texto y con aria-labelledby el id o ids de los elementos de la página que actúan como etiqueta.
- aria-describedby permite asociar una descripción larga a un elemento indicando el id del elemento que proporciona la descripción. Uno de las novedades previstas en la versión WAI-ARIA 1.1 será la inclusión de aria-describedby que permitirá incluir la URL de la página que describe al elemento.

# Ejemplos aria-label

```
<div id="leftnav" role="navigation" aria-label="Menú principal">  
<ul><li>...una lista de enlaces ...</li></ul> </div>  
<div id="rightnav" role="navigation" aria-label="Menú secundario">  
<ul><li>...una lista de enlaces ...</li> </ul></div>
```

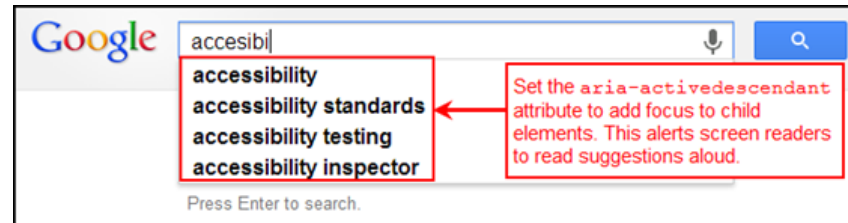
```
<p id="report-title">Download 2012 Sales Report:  
<a aria-labelledby="report-title pdf" href="pdf.pdf" id="pdf">PDF</a> |  
<a aria-labelledby="report-title doc" href="word.doc" id="doc">Word</a> |  
<a aria-labelledby="report-title ppt" href="ppt.ppt" id="ppt">Powerpoint</a></p>
```



# Algunos estados y propiedades Importantes

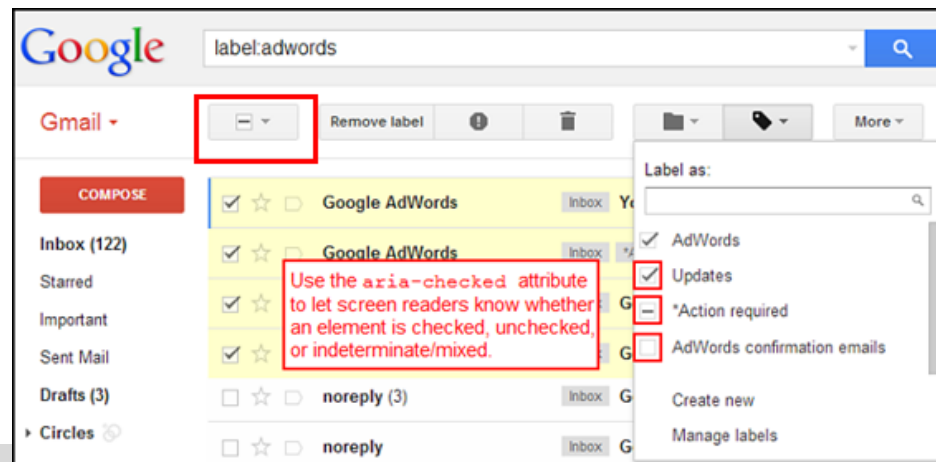
## ➤ aria-autocomplete y aria-activedescendant

- permiten mejorar la accesibilidad de las sugerencias en los campos de formularios.



## ➤ aria-checked

- permite definir un estado nuevo para los checkbox: aria-checked = "mixed".
- Es útil para los checks que permiten seleccionar o deseleccionar todos los checks de una lista, de esta manera se puede indicar cuándo su estado es misto (ni todos seleccionados ni todos deseleccionados).



# Paso a paso para el diseño con WAI-ARIA

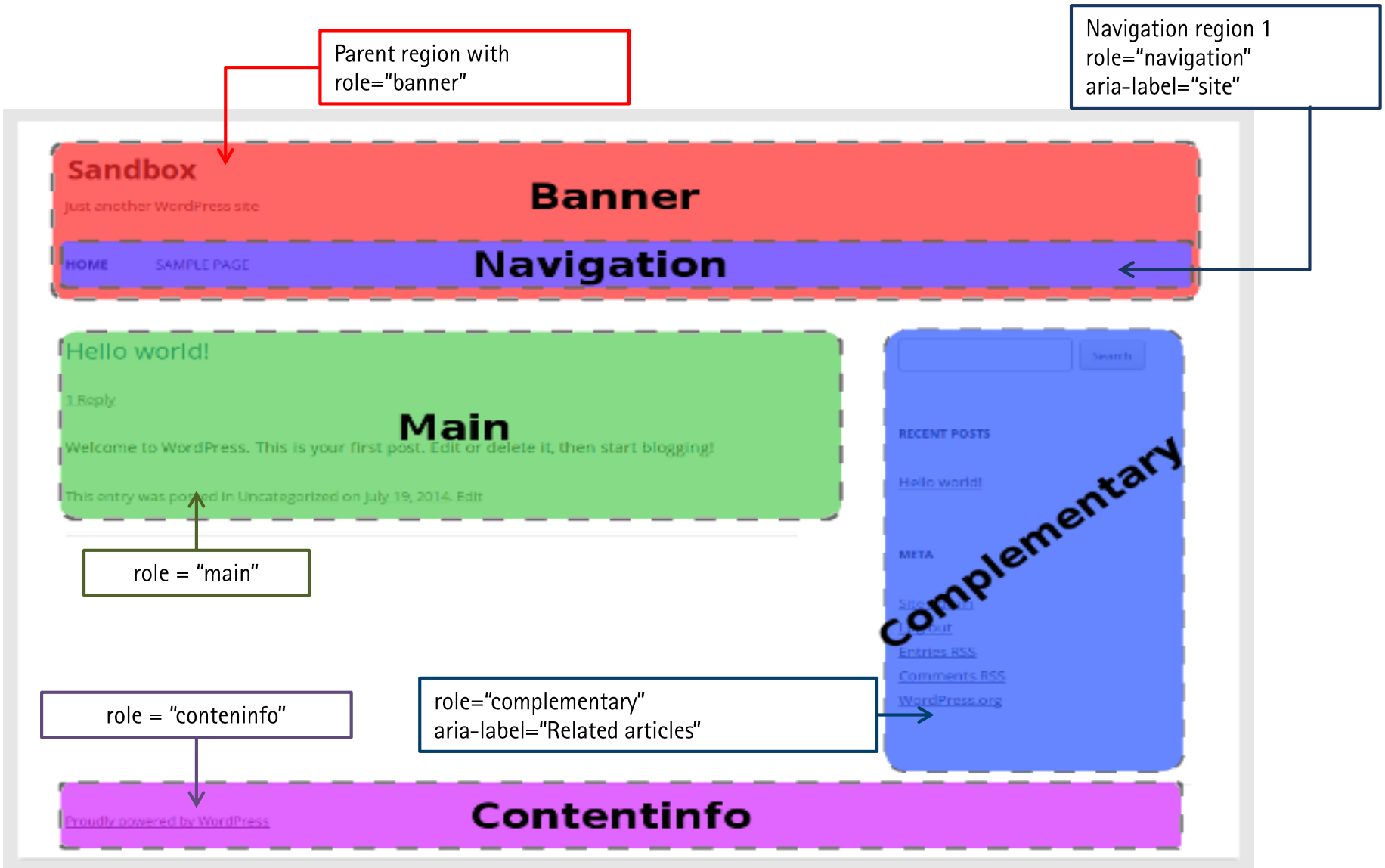
1. Mapea el diseño visual a regiones
2. Diseña la experiencia de teclado
3. Mapea de experiencia de teclado a widgets.
4. Define las relaciones de etiquetado y descripción
5. Completa el estado y propiedades de widgets.

# 1. Mapea el diseño visual a regiones

## Regiones de Referencia WAI-ARIA

- **Banner** – Región de cabecera
- **Búsqueda** – Contiene la herramienta de búsqueda.
- **Navegación** – Contiene links y otros elementos de navegación.
- **Principal** – Contiene el contenido principal de la pagina
- **Formulario** – Contiene una colección de elementos de tipo formulario
- **Complementario** – Contiene información que se vincula con el contenido principal
- **Contentinfo** – Contiene notas al pie, derechos de autor, normas de privacidad, etc.
- **Región** – Colección de contenido que no es aplicable a otra función.
- **Aplicación** – Unidad de software incrustada en la pagina (no recomendable).

# Regiones de Referencia WAI-ARIA



## 2. Diseña la Experiencia de Teclado

- El objetivo es "Hace que la navegación por teclado sea igual a la experiencia de una aplicación de escritorio"
- Para elaborar una buena experiencia de teclado, se debe evaluar y estudiar el diseño visual y **agrupar, agrupar, agrupar...**

The screenshot shows a web application interface for the Accessibility Architecture Review Board (AARB). Red boxes highlight several key areas for keyboard navigation:

- Top Navigation Bar:** Includes links like 'Home', 'Profiles', 'Communities', and 'Apps'.
- Secondary Navigation Bar:** Includes 'Activities', 'To Do List', 'Activity Templates', and 'Recent Updates'.
- Activity Header:** Includes 'This Activity' and a search bar.
- Activity Goal:** 'Document twice monthly meetings for 2013'.
- Activity Outline:** A sidebar menu with 'Recent Updates', 'To Do Items', 'Tasks', 'Members', and 'Sections'.
- Activity Content:** A main area with sections like 'Logistics', 'February 18, 2013', 'January 21, 2013', 'January 7, 2013', 'March 4, 2013', and 'March 18, 2013'.

The interface is designed to be navigable via keyboard, with clear visual cues for each section and item.

# Interacción con el teclado

## ➤ Grupos verticales

- Con la tecla de tabulación es posible desplazarse por los grupos
- Dentro de cada grupo navegar por ellos con las teclas de flecha arriba o flecha abajo.



## ➤ Grupos horizontales

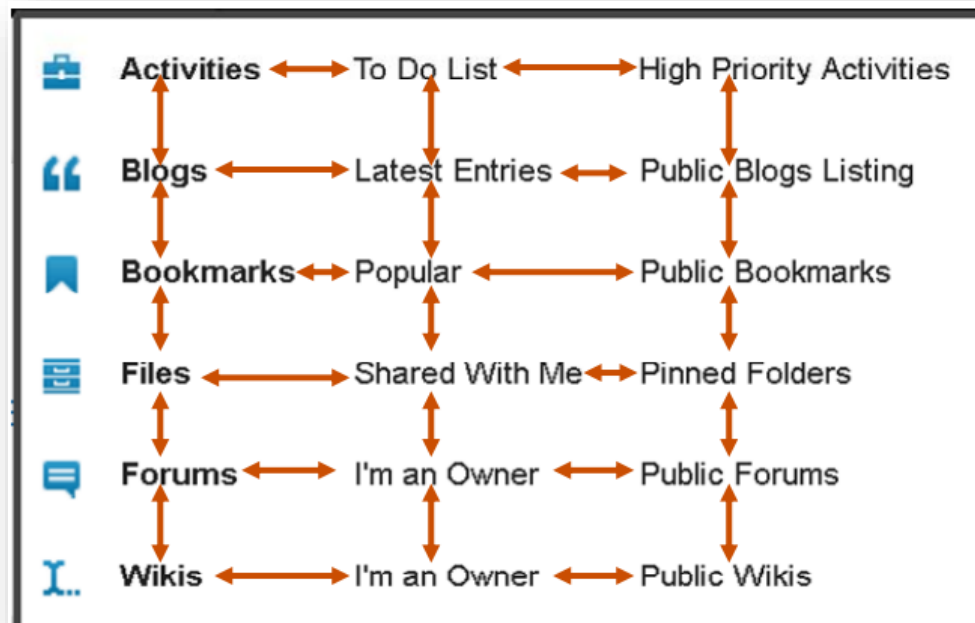
- Con la tecla de tabulación es posible desplazarse por los grupos y navegar por ellos con las teclas de flecha derecha o flecha izquierda.



# Interacción con el teclado

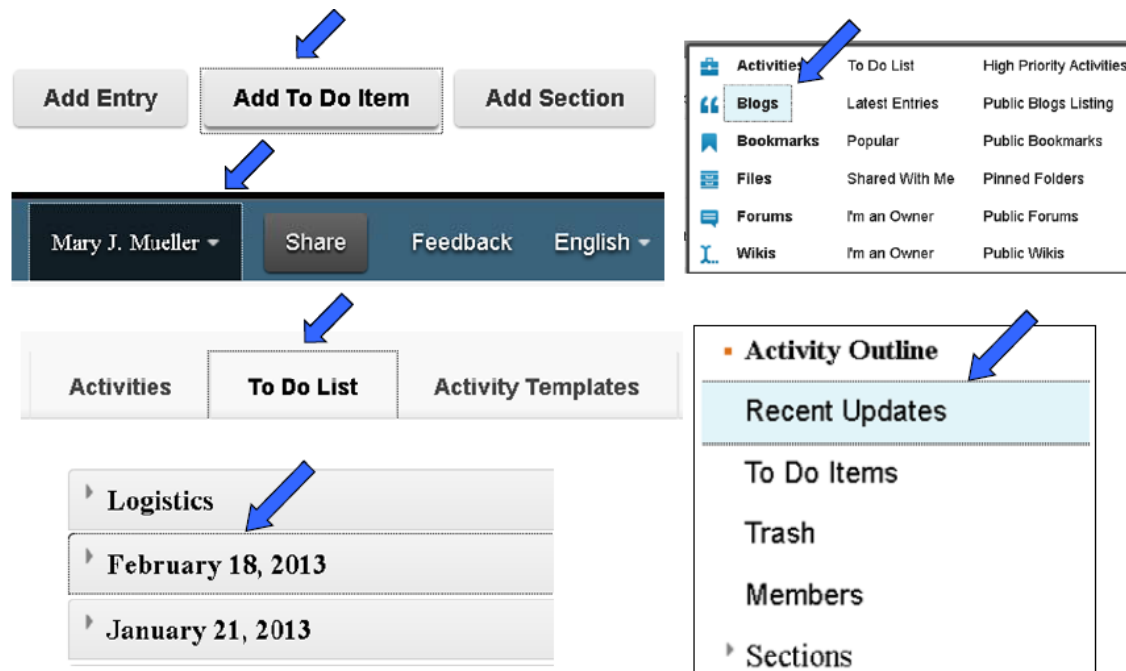
## ➤ Grupos tabulares

- Cuando aparecen bloques de elementos tabulares (tablas) se puede soportar las 4 teclas de flechas.



# Indicador visual

- Es importante no olvidar el indicador de selección visual, que señala en que posición se encuentra el teclado.





### 3. Mapea de experiencia de teclado a widgets

- El 99% de los diseños de navegación de una pagina moderna puede hacerse maravillosamente accesible con WAI-ARIA.
- Hay que conocer los patrones de teclado estándar para:
  - Barra de menú con submenús.
  - Menú y botón de menú.
  - Árbol.
  - Barra de herramientas.
  - Grid.
  - Diálogos que contienen menús y otros.
  - No utilizar nombres de rol en etiquetas, por ejemplo, “Sitio de Navegación” para región de navegación.
- Con una elección apropiada de un widget:
  - Se puede mapear perfectamente casi cualquier esquema de navegación.
  - Hacerlo igualmente eficaces para todos (teclado o ratón, visual o no visual).

# Escogiendo un widget

Para una adecuada elección:

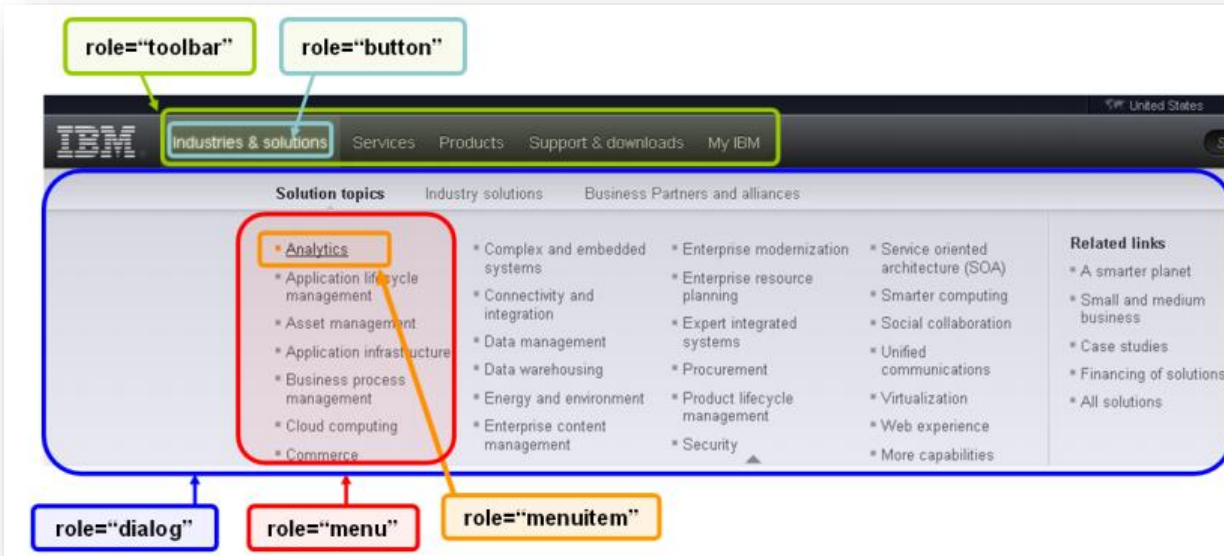
- ¿Hay flay-outs, ventanas emergentes o superposiciones?
  - Entender las características distintivas de menús, menús de botones y diálogos.
- ¿Hay indicación visual de la página actualmente activa?
  - Entender los patrones que pueden explotar aria-pressed, aria-selected, o ariachecked:
    - Árbol de navegación
    - Barra de herramientas con botones de palanca

# Ejemplo – Barra de menú



- Buena opción cuando:
  - Los menús son listas simples
  - La interacción del teclado puede coincidir con el patrón de menús

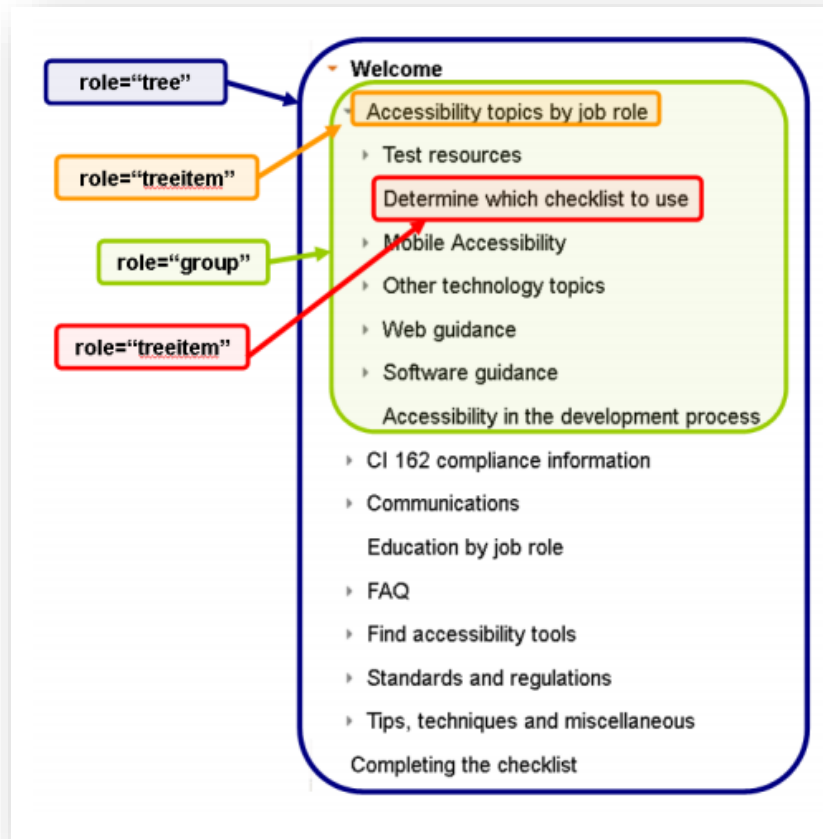
# Ejemplo – Mega menú



- Mega menu fly-outs o superposiciones:
- Tiene varios grupos de enlaces categorizados
- Puede tener búsqueda
- Puede incluir otros widgets

- Por lo tanto, requieren un patrón de diálogo:
  - Hacer modal desde la perspectiva del teclado (trap focus)
  - Incluye enfoque de icono cercano y soporte de escape
- ¿Qué widgets deben contener enlaces a un mega menú?
  - Los elementos de la barra de menús pueden abrir menús, no diálogos.
  - Los cuadros de diálogo deben abrirse con botones
  - Por tanto, los enlaces que abren mega menús pertenecen a la barra de herramientas de botones

# Ejemplo – Árbol de navegación



- Buena opción cuando:
  - Tenemos una estructura de navegación jerárquica
  - La interacción del teclado puede coincidir con el patrón de árbol

### 3. Etiquetado y descripción

- Después de identificar los elementos de referencia o widget, TODOS, con pocas excepciones, debe ser etiquetados y relacionados con cualquier elemento descriptivo.
- Utilizar **aria-labelledby** donde sea posible.
- La reutilización de textos facilita el mantenimiento, traducción, etc.
- La clave es ser breve:
  - Usar 1 o 2 palabras cortas, 3 máximo.
  - No utilizar nombres de rol en etiquetas, por ejemplo, “Sitio de Navegación” para región de navegación.
  - No colocar instrucciones en las etiquetas, por ejemplo, “Presione enter para...”

### 3. Etiquetado y descripción - Ejemplos

Elemento	Bien	Mal
Región de navegación	“Sitio”, “IBM”	“Navegación del Sitio Principal”
Barra de menú principal de navegación	“Mapa del Sitio”	“Menú de Navegación”
Árbol de navegación izquierdo	“Sitio”, “Wiki”	“Navegador del sitio – Utilice las flechas hacia arriba o hacia abajo para seleccionar un enlace y pulse enter”
Región de búsqueda para búsqueda de productos	“Productos”	“Buscar un Producto”

# Etiquetado de referencia

- Usar etiquetas en varias regiones del mismo tipo





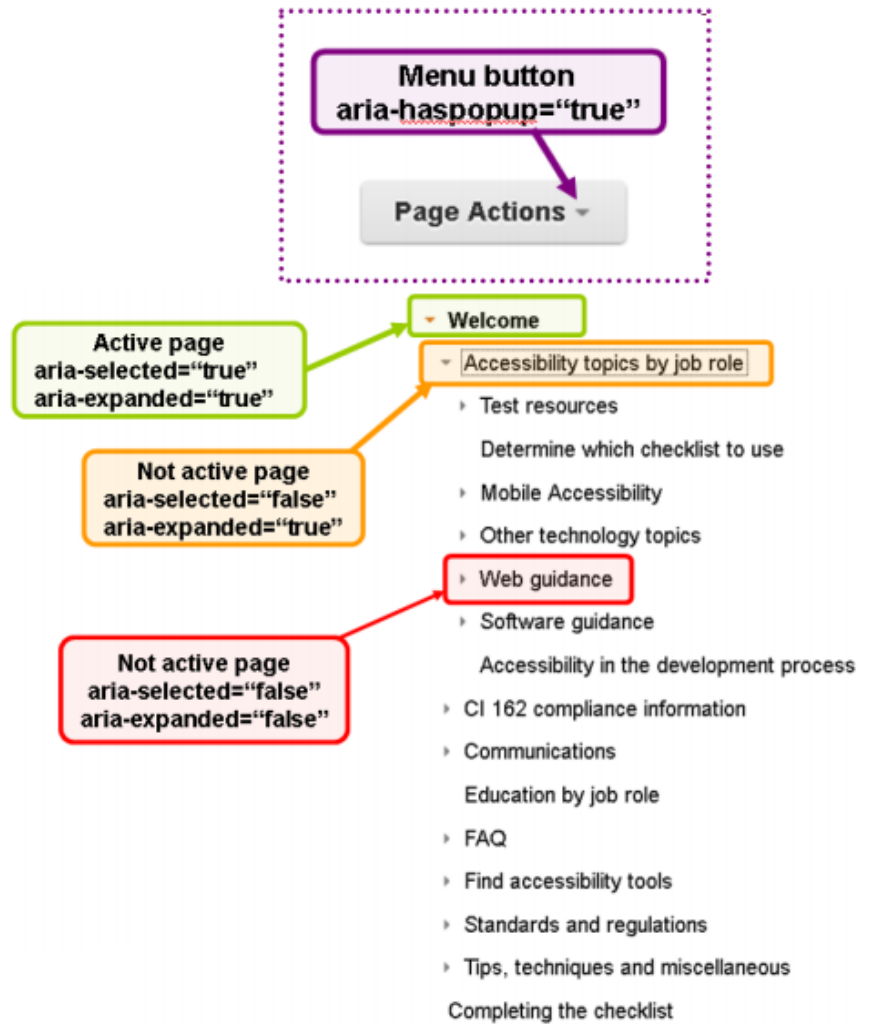
# Etiquetas compuestas

- Creación de etiquetas compuestas a partir de múltiples elementos
  - Dar al aria-labelledby múltiples IDs
  - Puede auto-referenciarse mediante la inclusión del id del elemento que se etiqueta
  - Puede tener aria-label y aria-labelledby (ambos) en el mismo elemento
  - El navegador concatena automáticamente en el orden especificado
- Muy útil cuando varios elementos tienen la misma etiqueta visual
  - Su contexto proporciona distinción para los usuarios visuales
  - La etiqueta compuesta agrega dinámicamente el contexto para los usuarios de lectores de pantalla
- Ejemplo: 5 botones de acción en una página
  - etiquetas aria-label en cada botón
  - aria-labelledby concatena la etiqueta del botón con la etiqueta del contexto para dar la etiqueta completa

```
<h2 id="title1">Wiki portlet</h2>
<a id="button1" role="button" aria-label="Actions for" aria-labelledby="button1 title1" href="...">
    Actions
</a>
```

## 5. Completa el estado y propiedades de widgets.

- Además del comportamiento del teclado, rol del widget, etiqueta y descripción:
- Implementar las propiedades WAI-ARIA por patrones de diseño:
  - Representar indicadores visuales de página activa con propiedades WAI-ARIA
  - Comunique la presencia de menús emergentes y submenús con aria-haspopup
- Proporcionar una indicación visual clara del enfoque del teclado





## Pongámoslo en práctica

- Haz tu proyecto compatible con WAI-ARIA



 **netmind**

**WeKnowIT**

## Barcelona

C. Almogàvers, 123  
08018 Barcelona  
Tel. 93 304.17.20  
Fax. 93 304.17.22

## Madrid

Plaza Carlos Trías Bertrán, 7  
28020 Madrid  
Tel. 91 442.77.03  
Fax. 91 442.77.07

[www.netmind.es](http://www.netmind.es)



MINISTERIO  
DE ENERGÍA, TURISMO  
Y AGENDA DIGITAL

**red.es**



**UNIÓN EUROPEA**

Fondo Social Europeo  
*"El FSE invierte en tu futuro"*