



ALUMNO:

María Fernanda Fuentes Abascal

DOCENTE:

José Miguel Carrera Pacheco

ACTIVIDAD:

DOM y componentes dinámicos accesibles

MATERIA:

Desarrollo Web Profesional

FECHA:

08/02/2026

Este documento tiene como finalidad establecer de manera detallada y estructurada las decisiones de diseño, arquitectura e implementación relacionadas con el manejo del DOM y la creación de componentes dinámicos accesibles dentro de la Plataforma Web de Apoyo Académico.

Su propósito no es únicamente describir qué se va a desarrollar, sino justificar técnicamente por qué se toman dichas decisiones, alineándolas con estándares profesionales, buenas prácticas de accesibilidad (WCAG 2.1) y flujos reales de trabajo en equipos fullstack.

El documento funciona como una referencia común para todos los roles del proyecto, asegurando que cada área comprenda cómo sus acciones impactan directamente en la experiencia del usuario, la accesibilidad, la calidad del sistema y su mantenibilidad a largo plazo.

ARQUITECTURA GENERAL DE COMPONENTES

La arquitectura de componentes se diseña bajo un enfoque modular, escalable y accesible, permitiendo que el sistema crezca de forma controlada y sostenible.

Los componentes se conciben como unidades reutilizables, capaces de responder a eventos del usuario, actualizar el contenido de forma dinámica y mantener la accesibilidad como parte intrínseca de su diseño. Cada componente es responsable de su comportamiento interactivo, su semántica y su relación con el foco.

Esta arquitectura prioriza:

- Claridad en los flujos de interacción
- Consistencia en el comportamiento de la interfaz
- Separación entre datos, presentación e interacción

DISEÑO DE COMPONENTES DINÁMICOS

El diseño de los componentes dinámicos parte del principio de que el contenido puede cambiar sin recargar la página, pero dichos cambios no deben ser invisibles ni confusos para el usuario.

3.1 Principios de diseño aplicados

Cada componente dinámico cumple con los siguientes principios:

- No requiere recarga de página para mostrar información nueva
- Mantiene una jerarquía semántica clara
- Permite navegación completa mediante teclado
- Mantiene el foco visible y predecible
- Anuncia cambios relevantes a tecnologías asistivas

Estos principios garantizan una experiencia usable tanto para usuarios estándar como para personas que navegan sin mouse o utilizan lectores de pantalla.

3.2 Componentes dinámicos definidos

Dentro de la plataforma se identifican los siguientes componentes clave:

- Listado dinámico de cursos, que se actualiza según filtros o búsquedas
- Buscador con resultados en tiempo real, manteniendo el foco en el campo de entrada
- Mensajes de estado (carga, error, éxito), visibles y correctamente anunciados
- Modales informativos accesibles, que no rompen el flujo de navegación

Cada componente se diseña pensando en flujos de interacción, no únicamente en su apariencia visual.

MANIPULACIÓN DEL DOM Y GESTIÓN DEL FOCO

La manipulación del DOM es uno de los puntos más críticos del sistema, ya que cambios mal gestionados pueden provocar pérdida de foco, desorientación o información inaccesible.

4.1 Estrategia de manipulación del DOM

Se establecen las siguientes prácticas obligatorias:

- Uso de actualizaciones selectivas del DOM
- Evitar reemplazos completos de secciones cuando no son necesarios
- Separación clara entre lógica de datos y lógica de renderizado

Estas decisiones permiten mantener el control sobre el comportamiento dinámico de la interfaz y reducen el riesgo de errores de accesibilidad.

4.2 Gestión del foco

La gestión del foco se considera una responsabilidad explícita del sistema, no un comportamiento automático del navegador.

Reglas definidas:

- El foco nunca debe desaparecer tras una actualización dinámica
- El foco solo se mueve cuando la acción del usuario lo justifica
- Al cargar contenido nuevo, el foco se dirige al encabezado principal del área actualizada
- El foco visual debe ser claramente perceptible

En modales accesibles:

- El foco se mueve al interior del modal al abrirse
- Se mantiene dentro del diálogo
- Se devuelve al elemento original al cerrarse

Este enfoque garantiza continuidad, previsibilidad y comprensión del sistema.

USO DE ARIA EN COMPONENTES DINÁMICOS

ARIA se utiliza como complemento del HTML semántico para comunicar cambios dinámicos a tecnologías asistivas.

5.1 Atributos utilizados

- `aria-live="polite"` para actualizaciones informativas
- `aria-live="assertive"` para errores críticos
- `role="alert"` para mensajes importantes
- `role="dialog"` en modales accesibles
- `aria-labelledby` para asociar regiones con encabezados

5.2 Criterios de uso

ARIA no sustituye al HTML semántico. Su uso se limita a los casos donde la semántica nativa no es suficiente, evitando sobrecargar la interfaz y generar confusión en lectores de pantalla.

FLUJOS DE INTERACCIÓN (NO PANTALLAS)

El sistema se concibe en términos de flujos de interacción y no como un conjunto de pantallas aisladas.

Ejemplo de flujo:

1. El usuario accede al buscador de cursos
2. Introduce texto mediante teclado
3. Se dispara un evento dinámico
4. Se consulta la información correspondiente
5. El listado se actualiza sin recargar
6. El cambio se anuncia mediante `aria-live`
7. El foco permanece en el campo de búsqueda

Este enfoque evita desorientación, interrupciones y pérdida de contexto.

PRUEBAS FRONTEND Y ASEGURAMIENTO DE CALIDAD

Las pruebas frontend son fundamentales para garantizar que los componentes dinámicos funcionen correctamente en distintos escenarios y no comprometan la accesibilidad.

7.1 Tipos de pruebas realizadas

- Pruebas unitarias de componentes dinámicos
- Pruebas de navegación por teclado
- Pruebas de foco visible
- Pruebas de anuncios accesibles

7.2 Casos de prueba clave

- Verificar que el foco no se pierda tras una actualización
- Confirmar que los mensajes dinámicos sean anunciados correctamente
- Validar que toda la interfaz sea usable sin mouse

Estas pruebas permiten detectar errores que no son evidentes visualmente pero afectan gravemente la experiencia del usuario.

INTEGRACIÓN CON CI/CD

El proyecto integra un flujo de Integración Continua que incluye:

- Ejecución automática de pruebas frontend
- Validación de rutas y componentes dinámicos
- Verificación de errores críticos de accesibilidad
- Bloqueo de merges si los tests fallan

Este enfoque garantiza que la accesibilidad, el foco y la calidad del sistema se mantengan de forma constante a lo largo del desarrollo.

CONCLUSIÓN

Este documento establece cómo la teoría sobre DOM, accesibilidad y componentes dinámicos se traduce en decisiones técnicas reales. La correcta gestión del foco, la validación mediante pruebas y la integración con CI/CD permiten construir una plataforma profesional, accesible, mantenible y alineada con prácticas reales de la industria.