



Secretaría  
de Educación



## **UNIVERSIDAD TECNOLÓGICA DE TEHUACÁN**

### **INGENIERÍA EN DESARROLLO Y GESTIÓN DE SOFTWARE**

**Alumna:  
Concepción Guadalupe Paniagua González**

**Asignatura:  
Desarrollo Web Profesional**

**Docente:  
José Miguel Carrera Pacheco**

**Octavo Cuatrimestre Grupo "A"**

**S - 4. Navegación avanzada y manejo de  
errores**

## UX de páginas de error (404 y 500)

### Error 404 Not found:

Los errores 404 son la respuesta a una petición que no se puede encontrar. Esto, en la práctica viene a ser que el usuario quiere acceder a una página que no existe ya sea porque se ha borrado, se ha movido a otra dirección sin comunicárselo al servidor o que quizá que nunca existido.

Algunos de los factores que pueden causar un error 404 son:

- Página movida o eliminada
- URL mal escrita
- Problema de configuración DNS
- El navegador puede almacenar en caché la página de error 404 en lugar del contenido real. Por lo tanto, sigues viendo el error, aunque el sitio web funciona para todos los demás.
- Normalmente, cualquier activo que falle, como una imagen, un archivo CSS o JavaScript, puede generar un error 404.

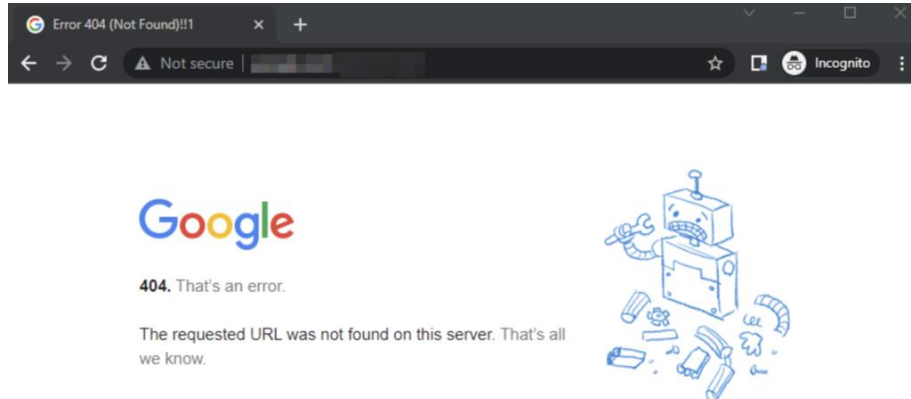
### ¿Cómo diseñar una página de error 404?

Los errores pueden molestar y estresar a los usuarios, por lo que es importante ofrecerles una posible solución.

1. No uses mensajes poco claros: si has movido el contenido pon un enlace al nuevo sitio, si lo has borrado para siempre indícalo claramente. Así el usuario deja de dar vueltas y no pierde el tiempo buscando algo que ya no existe.
2. Dar siempre una salida: no dejes a nadie atrapado en una pantalla de error. Lo mínimo es poner un botón para volver atrás (al punto exacto antes del fallo) o un acceso rápido al inicio para que puedan seguir con lo suyo sin tener que cerrar la pestaña.
3. Explicar qué hacer ahora: Hay que ser precisos pero cercanos. Si una búsqueda sale vacía, no te límites a decir "sin resultados"; sugiere probar con otras palabras o dales una ruta alternativa para que encuentren lo que necesitan por otro lado.
4. Una vez resuelta la parte técnica, se puede mejorar la experiencia con un buen diseño. Un toque de personalidad o un detalle creativo pueden transformar un error en una oportunidad para conectar con el usuario y reforzar la imagen de marca.

Ejemplos:

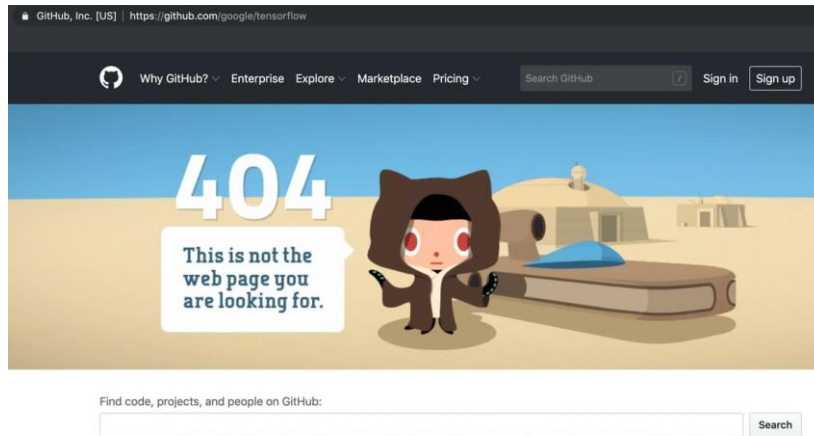
- **Google Chrome:** si te encuentras con una página de error 404 en Chrome, el navegador muestra un mensaje descriptivo bajo el código de estado HTTP.



- **Pixar:** utiliza personajes de sus películas, como Tristeza de "Inside Out", para comunicar el error. Esto no solo aporta una carga emocional positiva, sino que también refuerza su identidad visual y conecta con los fans a través del contenido.



- **GitHub:** esta herramienta para desarrolladores muestra una ilustración de un gato (el icono de la marca) en otro planeta, transmitiendo la idea de “pérdida” de forma simpática.



## Error 500 Internal Server Error:

**Error 500 (Fallo interno del servidor)**, ocurre cuando el servidor donde se aloja la web encuentra un problema inesperado que le impide procesar la solicitud. Es un error genérico del lado del sistema, lo que significa que el usuario no tiene la culpa, pero se queda sin acceso al contenido.

¿Qué causa un error 500?

- Límite de memoria PHP agotado
- Caché del navegador
- Servidor sobrecargado
- Archivos dañados
- Permisos incorrectos de archivos y carpetas
- Archivos .htaccess con nombres incorrectos
- Conflicto con un plugin o tema de WordPress
- Conflicto con diferentes configuraciones del servidor
- Código obsoleto
- Error de programación

¿Cómo diseñar una página de error 500?

Cuando un usuario se encuentra con un error 500, lo más honesto es avisarle de inmediato para que no pierda el tiempo. A veces el fallo es algo momentáneo y se

arregla simplemente con recargar la página, por lo que sugerir un "reintento" suele ser la solución más rápida y efectiva.

Pero la cosa se complica si, por ejemplo, el error salta en pleno proceso de compra o justo después de pagar algo. Aunque la mayoría de los servidores tienen filtros para evitar desastres, siempre cabe la posibilidad de que algo falle. Si el usuario recarga por su cuenta, se arriesga a duplicar los productos en el carrito o a que se le pase el mismo cobro dos veces.

Para evitar este caos, lo ideal es:

- Tener un sistema de confirmación externo que valide rápido qué se ha pagado y qué no.
- Darle al cliente una forma directa de contactar si cree que ha habido un error con su dinero. No hay nada que estrese más que un fallo técnico durante un pago.
- Quitando estos casos críticos, el diseño de una pantalla 500 tiene que seguir la misma lógica que la del 404. Hay que ser claros, explicar qué ha pasado sin rodeos y dar siempre una opción para que el usuario pueda volver atrás o ir a otro sitio en lugar de quedarse bloqueado.

Ejemplo:

- **El error 500 estándar:**

## Internal Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

Please contact the server administrator at [admin@example.com](mailto:admin@example.com) to inform them of the time this error occurred, and the actions you performed just before this error.

More information about this error may be available in the server error log.

- **Google Chrome:**



**500.** That's an error.

There was an error. Please try again later. That's all we know.



## Propuesta de mensajes

- **Página 404 – Página no encontrada**

- ✓ El título principal: “Esta página no existe”
- ✓ Mensaje descriptivo: El enlace puede estar mal escrito o la página ya no está disponible.
- ✓ Acciones disponibles:
  1. Botón primario (CTA): Volver al inicio
  2. Botón secundario: Ver otras opciones

**Objetivo:** Evitar frustración y permitir que el usuario continúe navegando sin quedar bloqueado en un solo lugar.

- **Página 500 – Error del sistema**

- ✓ Título principal: “Algo salió mal”
- ✓ Mensaje descriptivo: Tuvimos un problema al cargar esta sección. Puedes intentar nuevamente o regresar al inicio.
- ✓ Acciones disponibles:
  1. Botón primario (CTA): Reintentar
  2. Enlace secundario: Volver al inicio

**Objetivo:** Evitar culpar al usuario por el error y ofrecer una solución inmediata.

## Diseño de navegación accesible

### Menú persistente y breadcrumbs

La navegación en Rapiti debe permitir que el usuario se desplace por la página de forma intuitiva, consistente y accesible, evitando la desorientación y reduciendo la carga cognitiva.

Esta sección define cómo se estructura el menú persistente y los breadcrumbs, alineados con una arquitectura SPA en React y con criterios de accesibilidad.

### Menú persistente

Objetivo: Garantizar que el usuario tenga siempre un punto de referencia, pueda volver al inicio en cualquier momento y acceda rápidamente a las acciones principales de la aplicación.

El menú persistente responde a escenarios de uso reales de Rapiti: consultas rápidas de precios, navegación desde dispositivos móviles y usuarios con tiempo limitado.

- Estructura del menú

El menú se presenta como una barra fija visible en todas las vistas públicas de la página.

Elementos del menú:

1. Inicio
2. Buscar productos
3. Tiendas
4. Soy tienda (Login)

Los textos son claros y no técnicos, evitando términos como *dashboard* o *panel*, para facilitar la comprensión inmediata.

- Experiencia de uso

- ✓ El menú permanece visible durante la navegación, sin recargarse al cambiar de vista.
- ✓ Permite al usuario regresar al inicio o cambiar de sección en cualquier momento.
- ✓ Reduce la necesidad de utilizar el botón “atrás” del navegador.

- Accesibilidad aplicada

- ✓ Todos los elementos del menú son accesibles mediante navegación por teclado (TAB).
- ✓ El foco es visible en cada elemento interactivo.
- ✓ El orden de navegación es lógico y predecible.

- ✓ Las áreas de interacción cumplen con un tamaño mínimo adecuado para dispositivos táctiles.
- Alineación con la arquitectura React  
El menú se implementa como un componente global dentro de un layout principal, garantizando su persistencia entre rutas y su compatibilidad con una SPA.

### **Breadcrumbs (migas de navegación)**

Un elemento clave en la navegación web porque señalan en todo momento en qué lugar de la web está el usuario dentro de la jerarquía del sitio, de forma que facilitan que este no se pierda.

Objetivo:

1. Mejorar la navegación del usuario, permitiendo que este se mueva entre páginas y categorías con facilidad, porque sabe en todo momento dónde está.
2. Marcar una jerarquía y ruta de navegación más clara, que los motores de búsqueda entienden y agradecen, lo cual repercute positivamente en temas de posicionamiento [SEO](#).
3. Mejorar la experiencia de usuario.

- Estructura de los breadcrumbs:

Los breadcrumbs reflejan la jerarquía real de navegación.

Ejemplo de ruta:

*Inicio > Buscar productos > Producto > Tienda*

El último elemento representa la vista actual y no es interactivo.

- Experiencia de uso
  - ✓ Se muestran únicamente en vistas internas (no en la página de inicio).
  - ✓ Permiten regresar a secciones previas con un solo clic.
  - ✓ Refuerzan la orientación del usuario dentro del sistema.
- Accesibilidad aplicada
  - ✓ Los breadcrumbs se presentan como un elemento de navegación semántico.
  - ✓ Son completamente navegables mediante teclado.
  - ✓ El lector de pantalla anuncia correctamente la ruta y la página actual.
  - ✓ El último elemento se identifica claramente como la vista activa.



## Alineación con la arquitectura React

Los breadcrumbs se generan dinámicamente a partir de la ruta activa y se muestran debajo del menú persistente, manteniendo coherencia visual y funcional entre vistas.

- Navegación por teclado y validación  
La navegación entre el menú, los breadcrumbs y el contenido principal sigue un flujo lógico y continuo.
- Se validará que:
  - ✓ El usuario pueda recorrer el menú y los breadcrumbs únicamente con teclado.
  - ✓ El foco sea visible en todo momento.
  - ✓ No existan bloqueos o saltos inesperados en la navegación.
  - ✓ La estructura de navegación sea consistente en todas las vistas.

## Aria-live y lectores de pantalla

Las regiones en vivo de ARIA le permiten informar a los usuarios cuando algo ha cambiado en su sitio web, evitando que se pierdan actualizaciones importantes.

Hay dos formas distintas de crear regiones en vivo y es importante comprender la diferencia:

1. Uso del *aria-live* atributo: su enfoque directo es agregar el atributo a cualquier elemento que desee monitorear para detectar cambios. Esto le brinda control total sobre cómo y cuándo se anuncian las actualizaciones.
2. Uso de roles especializados de regiones activas: Algunos roles ARIA crean automáticamente regiones activas sin necesidad del *aria-live* atributo. Estos roles (como "estado" o " alerta ") tienen comportamientos integrados que los hacen ideales para situaciones específicas.

Su uso es especialmente relevante en:

- Mensajes de error
- Cambios de estado
- Contenido dinámico sin recarga de página

En Rapiti, el uso de aria-live evita que el usuario tenga que buscar manualmente el mensaje de error, reduciendo confusión y carga cognitiva.

## **Anuncio de errores en la aplicación**

Estrategia definida:

- El mensaje principal de error se presenta como el primer contenido de la vista.
- Se utiliza una región aria-live para anunciar el error al cargar la página.
- El foco inicial se posiciona sobre el mensaje principal.

Esta estrategia aplica para:

- Errores de sistema (404, 500)
- Errores críticos que bloquean la interacción del usuario

## **Lenguaje del mensaje**

Los mensajes anunciados:

- Evitan lenguaje técnico.
- Explican claramente qué ocurrió.
- Ofrecen al menos una acción para poder continuar.

## **Navegación por teclado (TAB)**

La navegación por teclado es fundamental para:

- Usuarios con discapacidad motriz
- Usuarios de lector de pantalla
- Ocasiones donde el mouse no está disponible

Rapiti debe ser completamente usable mediante la tecla TAB.

Comportamiento esperado:

- El recorrido con TAB sigue un orden lógico y predecible.
- No existen saltos inesperados de foco.
- No se generan bloqueos de navegación (*focus traps*).

Orden de navegación:

1. Menú persistente
2. Breadcrumbs
3. Contenido principal
4. Acciones secundarias

Indicador de foco visible:

El foco visible permite identificar claramente el elemento activo durante la navegación por teclado.

Criterios definidosg

- Todos los elementos interactivos muestran un indicador de foco visible.
- El foco tiene suficiente contraste con el fondo oscuro de la interfaz.
- No se utiliza únicamente el color para indicar el foco.

## Bibliografía

Boada, D. (2024, abril 5). *Error 404: qué es, por qué ocurre y cómo solucionarlo rápido*. Tutoriales Hostinger; Hostinger.

[https://www.hostinger.com/mx/tutoriales/error-404?utm\\_campaign=Generic-Tutorials-DSA-t2|NT:Se|LO:MX&utm\\_medium=ppc&gad\\_source=1&gad\\_campaignid=20926284428&gbraid=oAAAAADMy-hbw1oCHGz5isgaOtKeElbwLo&gclid=CjoKCQiAkPzLBhD4ARIsAGfah8j7JSNB5W-vadPi5NGKT5ueHPNXTRA8ocRnVbnt54NRI8TfMYz4tDoaAgvKEALw\\_wcB](https://www.hostinger.com/mx/tutoriales/error-404?utm_campaign=Generic-Tutorials-DSA-t2|NT:Se|LO:MX&utm_medium=ppc&gad_source=1&gad_campaignid=20926284428&gbraid=oAAAAADMy-hbw1oCHGz5isgaOtKeElbwLo&gclid=CjoKCQiAkPzLBhD4ARIsAGfah8j7JSNB5W-vadPi5NGKT5ueHPNXTRA8ocRnVbnt54NRI8TfMYz4tDoaAgvKEALw_wcB)

Pablo. (2023, abril 11). *Error 404 Not found y error 500 Internal Server Error*. Aprender UX UI. <https://aprenderuxui.com/error-404-not-found-y-error-500-internal-server-error/aprender/uidesign/>

Product Auditors. (2025, marzo 4). *Del problema a la solución en UX: Cómo definir un planteamiento de problema e hipótesis con ejemplos prácticos*. Productauditors.com. <https://www.productauditors.com/ux-ui/problemas-en-ux/>

*¿Qué es error 500?* (s/f). One.com. Recuperado el 2 de febrero de 2026, de <https://www.one.com/es-es/hosting/que-es-error-500/>

Facchin, J. (2021, abril 21). *¿Qué son las breadcrumbs o migas de pan y para qué sirven?* En el blog de SE Ranking hablamos sobre SEO y el Marketing Digital:

tutoriales, casos prácticos, y tendencias; SE Ranking.  
<https://seranking.com/es/blog/guia-sobre-breadcrumbs/>

Schroiff, F. (2024, diciembre 5). *The complete guide to ARIA live regions for developers*. The A11Y Collective. <https://www.a11y-collective.com/blog/aria-live/>